

TRAFFIC SIGN RECOGNITION

A THESIS SUBMITTED TO  
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES  
OF  
MIDDLE EAST TECHNICAL UNIVERSITY

BY

UFUK SUAT AYDIN

IN PARTIAL FULFILMENT OF THE REQUIREMENTS  
FOR  
THE DEGREE OF MASTER OF SCIENCE  
IN  
ELECTRICAL AND ELECTRONICS ENGINEERING

MAY 2009

Approval of the thesis:

**TRAFFIC SIGN RECOGNITION**

Submitted by **UFUK SUAT AYDIN** in partial fulfillment of the requirements for the degree of **Master of Science in Electrical and Electronics Engineering Department, Middle East Technical University** by,

Prof. Dr. Canan ÖZGEN  
Dean, Graduate School of **Natural and Applied Sciences** \_\_\_\_\_

Prof. Dr. İsmet ERKMEN  
Head of Department, **Electrical and Electronics Eng.** \_\_\_\_\_

Assoc. Prof. Mehmet Mete BULUT  
Advisor, **Electrical and Electronics Eng.** \_\_\_\_\_

Prof. Dr. Gözde BOZDAĞI AKAR  
Co-Advisor, **Electrical and Electronics Eng.** \_\_\_\_\_

**Examining Committee Members**

Assoc. Prof. Mehmet Mete BULUT  
Electrical and Electronics Eng. Dept., METU \_\_\_\_\_

Prof. Dr. Gözde BOZDAĞI AKAR  
Electrical and Electronics Eng. Dept., METU \_\_\_\_\_

Assoc. Prof. Aydın ALATAN  
Electrical and Electronics Eng. Dept., METU \_\_\_\_\_

Assis. Prof. Çağatay CANDAN  
Electrical and Electronics Eng. Dept., METU \_\_\_\_\_

Msc. Emre ULAY  
MGEO, ASELSAN \_\_\_\_\_

**Date:** **12.05.2009**

**I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.**

**Name, Last name: U. Suat AYDIN**

**Signature :**

## **ABSTRACT**

### **TRAFFIC SIGN RECOGNITION**

AYDIN, Ufuk Suat

M.Sc., Department of Electrical and Electronics Engineering

Advisor: Assoc. Prof. Mehmet Mete BULUT

Co-Advisor: Prof. Dr. GÖZDE BOZDAĞI AKAR

May 2009, 96 pages

Designing smarter vehicles, aiming to minimize the number of driver-based wrong decisions or accidents, which can be faced with during the drive, is one of hot topics of today's automotive technology. In the design of smarter vehicles, several research issues can be addressed; one of which is Traffic Sign Recognition (TSR). In TSR systems, the aim is to remind or warn drivers about the restrictions, dangers or other information imparted by traffic signs, beforehand. Since the existing signs are designed to draw drivers' attention by their colors and shapes, processing of these features is one of the crucial parts in these systems. In this thesis, a Traffic Sign Recognition System, having ability of detection and identification of traffic signs even with bad visual artifacts those originate from some weather conditions or other circumstances, is developed.

The developed algorithm in this thesis, segments the required color influenced by the illumination of the environment, then reconstructs the shape of partially occluded traffic sign by its remaining segments and finally, identifies it. These three stages are called as “Segmentation”, “Reconstruction” and “Identification” respectively, within this thesis. Due to the difficulty of analyzing partial segments to construct the main frame (a whole sign), the main complexity of the algorithm takes place in the “Reconstruction” stage.

**Keywords:** Traffic sign, pattern recognition, segmentation, identification, neural network

## ÖZ

### TRAFİK İŞARETİ TANIMA

AYDIN, Ufuk Suat

Yüksek Lisans, Elektrik-Elektronik Mühendisliği Bölümü

Tez Danışmanı: Doç. Dr. Mehmet Mete BULUT

Ortak Tez Danışmanı: Prof. Dr. Güzde BOZDAĞI AKAR

Mayıs 2009, 96 sayfa

Sürüş sırasında meydana gelebilecek sürücü temelli yanlış kararları veya kazaların sayısını en aza indirmek amacıyla daha zeki araçlar tasarlamak, günümüzün otomotiv teknolojisinin en yoğun uğraşı konularından biridir. Bu akıllı araç tasarımına uygun olan çeşitli araştırma konuları gösterilebilir; bu konulardan biri de Trafik İşareti Tanıma konusudur. Trafik İşareti Tanıma sistemlerinde amaç, sürücülere, trafik işaretleriyle belirtilmiş kısıtlamaları, tehlikeleri ve diğer bilgileri önceden hatırlatmak veya onları uyarmaktır. Halihazırda bulunan işaretler, renkleri ve şekilleriyle sürücülerin dikkatini çekmek amacıyla dizayn edildiği için, bu sistemlerin en önemli bölümünü, bu özellikleri işleme oluşturmaktadır. Bu tezde, bazı hava koşulları veya diğer çevresel durumlardan kaynaklı, kötü görsel faktörlerle dahi, trafik işaretlerini algılama ve tanımlama yetisine sahip bir Trafik İşareti Tanıma Sistemi geliştirilecektir.

Bu tezde geliştirilen algoritma, çevresel aydınlanmadan etkilenmiş, aranan rengi ayırır, daha sonra parçalı olarak kapanmış trafik işaretinin şeklini, kalan parçasından tekrar oluşturur ve son olarak da onu tanımlar. Bu üç safha, bu tez içinde, sırasıyla “Bölümleme”, “Tekrar Oluşturma” ve “Tanımlama” olarak anılır. Ana yapıyı (bütün bir işareti) oluşturmak için kısmi bölümlerin analiz edilmesinin zorluğu nedeniyle, algoritmanın asıl kompleksliği “Tekrar Oluşturma” safhasında yer alır.

**Anahtar Kelimeler:** Trafik işareti, örüntü tanıma, bölümleme, tanımlama, sinir ağıları

**To My Life**

# TABLE OF CONTENTS

ABSTRACT.....	iv
ÖZ.....	vi
TABLE OF CONTENTS.....	ix
LIST OF FIGURES.....	xi
LIST OF TABLES .....	xvi
LIST OF ABBREVIATIONS.....	xvii
CHAPTERS	
1 INTRODUCTION.....	1
1.1 SCOPE OF THE THESIS.....	2
1.2 OUTLINE OF THE THESIS.....	3
2 LITERATURE SURVEY .....	4
2.1 SENSING .....	5
2.2 DETECTION OF TRAFFIC SIGNS .....	6
2.2.1 COLOR ANALYSIS .....	6
2.2.2 SHAPE ANALYSIS.....	16
2.2.3 ANALYSIS OF INDIRECTLY OBTAINED FEATURES ..	26
2.3 IDENTIFICATION.....	30
3 THE PROPOSED ALGORITHM.....	39
3.1 SEGMENTATION.....	40
3.2 CANDIDATE SELECTION .....	42
3.2.1 LINE HISTOGRAM ANALYSIS .....	43
3.2.1.1 LINE SEGMENTS.....	45
3.2.1.2 ARCS .....	48
3.2.1.3 COMBINED COMPONENTS .....	51

3.2.1.4	NOISY COMPONENTS .....	52
3.2.2	DECISION OF TRIANGULAR AND CIRCULAR CANDIDATES .....	53
3.2.3	RESIZING AND SKEWING THE COMPONENTS .....	55
3.3	IDENTIFICATION .....	56
3.3.1	FEATURE EXTRACTION .....	56
3.3.2	TEMPLATES AND TRAINING SETS .....	59
3.3.3	CLASSIFICATION .....	61
3.3.3.1	TEMPLATE MATCHING .....	61
3.3.3.2	NEURAL NETWORKS .....	61
4	EXPERIMENTS AND RESULTS .....	67
4.1	DETECTION .....	70
4.2	CLASSIFICATION .....	78
5	CONCLUSIONS AND FUTURE WORK .....	86
5.1	CONCLUSION .....	86
5.2	FUTURE WORKS .....	88
	REFERENCES .....	89
	APPENDIX A .....	94

## LIST OF FIGURES

Figure 2.1 Color classification LUTs [5] .....	7
Figure 2.2 Original image and result of chromatic equalization [16]. ...	10
Figure 2.3 Gamma correction curve [16] .....	11
Figure 2.4 Hue membership function [17].....	12
Figure 2.5 Saturation membership function [17] .....	12
Figure 2.6 The output function [17].....	13
Figure 2.7 The fuzzy system surface [17].....	13
Figure 2.8 a) Detection function on hue value for red and blue pixel detections, b) detection function on saturation value.....	16
Figure 2.9 (a) is the input image, (b) shows the region of interest, (c) is the output of Canny's edge detection algorithm, (d) is the resulting line segments after algorithm by Pavlidis, (e) shows the selected edges with suitable slopes, (f) is showing the found triangular sign [6].....	17
Figure 2.10 [8] Used masks to detect central upper and lower left corners respectively.....	18
Figure 2.11 [8] Search regions for a triangular sign.....	19
Figure 2.12 a) Search regions for a rectangular sign, b) application of the corner masks to a circular sign [8] .....	19
Figure 2.13 The normal parameters for a line [10].....	20
Figure 2.14 Mapping of 3 points on x-y plane to 3 circles on a-b plane [11].....	21
Figure 2.15 An example of accumulator of the CHT with real world data [12].....	22

Figure 2.16 a) The $R_1$ membership functions, b) The $R_2$ membership functions, c) The T membership functions, d) The E membership functions, e) The O membership functions, f) The output membership functions [17] .....	25
Figure 2.17 Different choices of rectangular regions a) [19] for Haar-like feature, b) [14] as disassociated dipoles. The black region corresponds to the inhibitory dipole, whereas the white region corresponds to the excitatory dipole [14], c) [19] Calculation of features by integral image. ....	28
Figure 2.18 Cascaded structure of the designed classifier [19] .....	29
Figure 2.19 a)[19] selected Haar-like features for Speed-Limit sign, ...	30
Figure 2.20 Example of variations, rotations and translations, of each model included in the training set. The original model is drawn in the image at the top [16]. ....	32
Figure 2.21 Result of distance transform. a) Original image, b) Edge map, c) Distance transform of b) [19].....	33
Figure 2.22 a) Network topology of the SOM [3], b) Threshold neighborhood, narrowing as training process [3] .....	35
Figure 2.23 Gabor wavelets representations [21] .....	35
Figure 2.24 a) Intermediate images with one example of Gabor features generated [13] b) Two-tier self organized map with 4 Gabor wavelets [21].....	36
Figure 2.25 Decision boundary obtained by a) an ordinary classifier and b) SVM [22].....	36
Figure 3.1 Sample input to the algorithm .....	39
Figure 3.2 saturation/value slice of a specific hue in the HSV model [2] .....	41
Figure 3.3 Thresholding graphs for each component of HSV (TH1=30, TH2=330, TH3=0.4).....	41

Figure 3.4 a) Sample input, b) output of HSV Thresholding, c) output of morphological opening.....	42
Figure 3.5 Sample of a partially occluded traffic signs.....	43
Figure 3.6 Line Histogram of a sample connected component.....	44
Figure 3.7 Sample connected component, Comp1.....	46
Figure 3.8 Line Histogram of Comp1.....	46
Figure 3.9 The reason of degree range on the line histogram.....	46
Figure 3.10 Sample connected component Comp2.....	48
Figure 3.11 Line histogram of Comp2.....	48
Figure 3.12 a) Sample connected component Comp3 containing an arc, b) Line histogram of Comp3, c) Local maximum region of line histogram.....	50
Figure 3.13 Sample for a combined component and its line histogram	51
Figure 3.14 Sample of a connected component that belongs to a triangular traffic sign and its line histogram.....	52
Figure 3.15 Two samples of traffic signs with bad visual conditions, their segmented major components and their line histograms.....	53
Figure 3.16 a) Occluded traffic signs, b) corresponding connected component, c) local maximum regions on their line histograms, d) extracted line segments and circle containing the arc. ....	54
Figure 3.17 a) Sample circular sign, b) two local maximum regions, c) two different circles found by each local maximum regions. ....	55
Figure 3.18 a) Captured traffic signs, observed as rotated and compressed, b) the found triangular sign, c) resulting 50x50-pixel candidate after transformation. ....	56
Figure 3.19 Sample segmentation of the informative part of the sign, a) Found candidate on the frame, b) affined candidate, c) default mask for triangular signs, d) the resulting input for the classification stage.....	57

Figure 3.20 a) Affined candidate, b) First feature set: red component of RGB image, c) Second feature set: RGB image itself, d) Third feature set: Gray-scaled image, e) Forth feature set: Distance transform of binary image. .... 58

Figure 3.21 a) -6 degrees rotated version, b) -3 degrees rotated version, c) original version, d) +3 degrees rotated version, e) +6 degrees rotated version of a sample sign. .... 60

Figure 3.22 original version at the center and shifted versions to 8 directions. .... 60

Figure 3.23 a) the original version, b) noisy version with standard deviation 0.1, c) with standard deviation 0.2. .... 60

Figure 3.24 50 by 50 pixel triangular mask image. .... 62

Figure 3.25 a) Similarity matrix  $A_{DT}^0$  for distance transform images, b) Similarity matrix  $A_{DT}^{11}$  for the first group after the first segmentation, c) Similarity matrix  $A_{DT}^{12}$  for the second group after the first segmentation. .... 64

Figure 3.26 a) Similarity matrix  $A_{RGB}^0$  for feature sets, red, gray and RGB, b) Similarity matrix  $A_{RGB}^{11}$  for the first group after the first segmentation, c) Similarity matrix  $A_{RGB}^{12}$  for the second group after the first segmentation. .... 64

Figure 4.1 a) Sample image from Set-1, b) another sample from Set-1, sample images from Set-2 c) Cluster group, d) Deformed group, e) Mixed group, f) Different Shapes group, g) Illumination group, h) Occlusion group, i) Rotation group, j) Shadows group, k) Sizes group, l) Translation group..... 69

Figure 4.2 Sample misdetection of a circular sign for a merged sign... 74

Figure 4.3 a) Sample merged signs, b) one of the connected components (with different color) to be analyzed, c) line histogram result, indicating 4 line segments, d) resulting line segments. ....	75
Figure 4.4 a) Recovered triangular sign, b) recovered circular sign, c) unrecovered triangular sign. ....	76
Figure 4.5 a) Sample circular sign, b) detected arc (different colored), c) recovered circle. ....	77
Figure 4.6 a) Input image, b) result of color segmentation, c) result after morphological opening.....	77
Figure 4.7 Two samples of Set-1 .....	83
Figure 4.8 a) Sample input candidates from Set-2, b) equivalent signs in the used database, c) common misclassification results. ....	84
Figure 4.9 a) Sample candidate without a response on the database, b) wrong classification result.....	84
Figure A.1 Stop and Parking Traffic Signs [1].....	94
Figure A.2 Danger Warning Traffic Signs [1] .....	95
Figure A.3 Prohibitory and Restrictive Traffic Signs [1].....	96

## LIST OF TABLES

Table 2.1 Shape measures of stop sign [17] .....	25
Table 2.2 Shape measures of yield sign [17].....	26
Table 2.3 Shape measures of no entry sign [17] .....	26
Table 2.4 Shape measures of rectangular signs [17].....	26
Table 4.1 Set-1 Detection Results .....	71
Table 4.2 Set-2 Detection Results (Proposed Algorithm) .....	71
Table 4.3 Set-2 Detection Results ( Proposed Algorithm with cvHoughLines2() & cvHoughCircles() ).....	72
Table 4.4 Set-2 Detection results of [27].....	73
Table 4.5 Set-1 Classification Results .....	79
Table 4.6 Set-2 Classification Results (Template Matching: Distance Transform) .....	79
Table 4.7 Set-2 Classification Results (Template Matching: RGB Image) .....	80
Table 4.8 Set-2 Classification Results (Neural Networks: Distance Transform) .....	80
Table 4.9 Set-2 Classification Results (Neural Networks: Red component).....	81
Table 4.10 Set-2 Classification Results (Neural Networks: Gray Scaled Image) .....	81
Table 4.11 Set-2 Classification Results (Neural Networks: RGB Image) .....	82

## **LIST OF ABBREVIATIONS**

CCD: Charge-Coupled Device  
HSV: Hue Saturation Value  
HSI: Hue Saturation Intensity  
LUT: Look-Up Table  
RGB: Red Green Blue  
TSR: Traffic Sign Recognition  
CHT: Circular Hough Transform  
MLP: Multi Layer Perceptron

# CHAPTER 1

## INTRODUCTION

Since safety becomes more important for customers, Traffic Sign Recognition (TSR) becomes one of today's research subjects aiming to improve safety of driving. While, they are presently developed just to warn drivers about some important traffic signs, in the future, these systems may take control of the vehicle under some circumstances.

The input is mainly consists of visual data during a drive. According to those inputs, the action of driving is performed by the driver. Although, today's technology cannot process all visual inputs as a human being, by a system focusing on some specified portion of this process, the workload of drivers can be decreased. Moreover, the resulting system is not influenced by some human related factors, such as tiredness, sleeplessness or so. As a result, the safety of driving is improved. For this purpose, TSR systems are developed, mainly to decrease the probability of missing some important traffic signs on the road.

The problem may seem to be easy to handle, at the first glance. However, since the process on a visual data is performed by human brain, based on all of his experiences, it cannot be easy for a computer

to perform the same process. Instead of those experiences, just some knowledge about distinctive features of traffic signs can be used. These features mainly consist of color and shape information. Although, this knowledge is not enough to separate traffic signs from other objects, the segmentation can be improved by the help of some intelligence while using the knowledge. For example, since the illumination is changing from time to time or from scene to scene, the color identification shall be performed accordingly. Additionally, since some traffic signs may be partially observed because of some conditions, the shape information shall be extracted despite of those conditions.

## **1.1 SCOPE OF THE THESIS**

In this thesis, detection and classification of traffic signs on captured frames are studied. In detection stage, color segmentation is used. Then, by shape analysis, candidates of traffic signs to be identified are selected within the connected components found by color segmentation. At the last step, classification, the candidates are identified using two different approaches, "Template Matching" and "Neural Networks".

The algorithm has to be applicable to the video stream while capturing. Hence, the implementation should work as fast as the video stream. Therefore, C++ is preferred as the programming language, and because of its processing speed, Open Computer Vision Library (OpenCV) is used for the implementation. On the other hand, MATLAB is used for some design calculations.

## **1.2 OUTLINE OF THE THESIS**

In Chapter 2, some of the major algorithms, used for Traffic Sign Recognition Systems in previous researches, are mentioned.

Chapter 3 gives the details about proposed algorithms in this thesis and the theory behind them.

Finally, Chapter 4 gives the experimental results.

Finally, Chapter 5 includes the conclusion and the future work.

## CHAPTER 2

### LITERATURE SURVEY

Traffic Sign Recognition (TSR) is a pattern-recognition problem based on image-processing concepts. Therefore, the stages of the problem consist of the five pattern-recognition steps applied with a guidance of image-processing methodology. Those are, "Sensing", "Segmentation", "Feature Extraction", "Classification" and "Post Processing". Shortly, in "Sensing" stage, the data is collected from sensors. In this case, this stage consists of capturing an image by a camera. Here the sensor is the CCD sensor in the camera and the data is the image. Next, in "Segmentation" stage, the aim is to isolate the required part of the whole data sensed previously. This stands for the image segmentation for this problem. "Feature Extraction" is the stage, gathering the distinctive features of the required pattern. In "Classification" stage, the assignment of the segmented data to one of the known classes is made, according to the extracted features. For this problem, this stage assigns the segmented image to a known traffic sign. Finally, in "Post Processing" stage, the final decision is made according to the costs.

For a Traffic Sign Recognition system, those five stages can be considered in three main steps, "Sensing", "Detection of Traffic Signs"

and “Identification”, accordingly. These steps are generally common for most of the TSR systems with some intersections between each other. Therefore, literature will be studied under the titles accordingly.

## **2.1 SENSING**

As its name indicates, this step stands for the “Sensing” stage of the Pattern Recognition problem. In this step, data is collected by one or several sensors. Most of the times, video cameras are used as sensors to capture sequential images. Since, the input data passed by this stage, the validity, reliability and usability of the data is highly dependent to this part of the solution. Therefore, additional sensors can be used to get higher quality of data. For this purpose, in [4], two cameras are used. One is equipped with a wide-angle lens (called wide-camera) and the other with a telephoto lens (called tele-camera) [4]. Wide-camera is directed to the direction of motion. On the other hand, tele-camera can change its direction according to the commands from next layer of the algorithm. While, wide-angle camera is used to detect candidates of traffic signs, tele-camera is directed to each candidate, by a prediction of the motion, to get more detailed image of the target signs. While this system is catching sign candidates with higher resolution, the probability of missing a candidate on a scene including more than one sign increases with such a moving camera.

Another two-camera system is used in [14]. In this case, two identical cameras are used to get stereo pairs of images. Using the epipolar geometry and the predefined calibration parameters, the region of a detected object on one source is estimated on its stereo objector. With a similarity criterion based on normalized correlation, the position of the

target is found. Using this dual redundancy, only the object that is not observable by either of the cameras is lost.

## **2.2 DETECTION OF TRAFFIC SIGNS**

In this step, the aim is to locate a road sign candidate on the captured image. To perform this mission, researchers tried to use the performance, offered by that day's computer technology, as efficient as possible. The less processing time of computers, the more complicated processes are used to get better results. Therefore, the researches, published within closer time periods, generally include similar techniques.

In detection step, the distinctive features of traffic signs shall be considered. Since, traffic signs are regulated to be in specific color and shape; it is convenient to use those features to decide the candidates. Additionally, in recent years, several feature extraction techniques are preferred to get more distinctive features those cannot be observed directly, like color or shape. Therefore, the detection algorithms can be studied under three subsections "Color Analysis" , "Shape Analysis" and "Analysis of Indirectly Obtained Features".

### **2.2.1 COLOR ANALYSIS**

On captured images of real world scenes, it is not always possible to obtain required colored regions by applying thresholds directly to RGB color space, because of the changing illumination of the environment. Therefore, this uncontrollable factor shall be separated from color

information. For this purpose, preferred approach is to work on different color spaces. In [5], HSI (Hue, Saturation and Intensity) space is used to segment red colored regions on the image. In this color map, hue component of a pixel stands for the color information. On the other hand, saturation refers to the dominance of hue in the color [2]. In other words, saturation describes the pureness of the color. The last component, intensity, is the illumination factor separated from color information. As shown on Figure 2.1, two Look-Up Tables (LUT) are constructed, where the used Hue and Saturation components are normalized to 255.

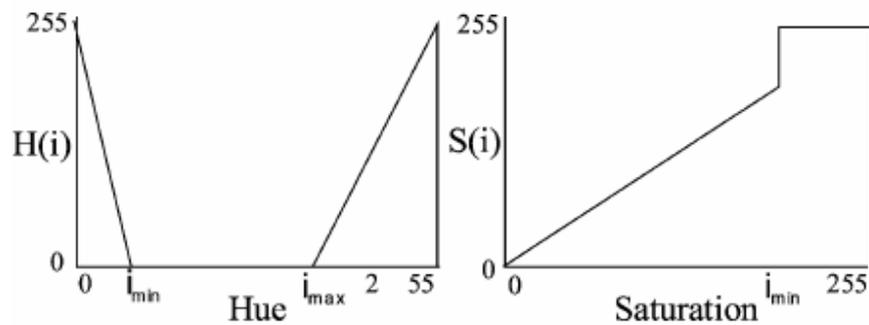


Figure 2.1 Color classification LUTs [5]

The formulations of those LUTs are as follows:

$$H(i) = \begin{cases} 255 \frac{i_{\min} - i}{i_{\min}} & 0 \leq i \leq i_{\min} \\ 0 & i_{\min} < i < i_{\max} \\ 255 \frac{i - i_{\max}}{i_{\max}} & i_{\max} \leq i \leq 255 \end{cases} \quad (2.1)$$

and

$$S(i) = \begin{cases} i & 0 \leq i \leq i_{\min} \\ 255 & i_{\min} \leq i \leq 255 \end{cases} \quad (2.2)$$

The aim of using these two LUTs, is to suppress the errors of color classification, resulting from each LUT. After the LUTs are applied, both masks are multiplied and hence, the resulting mask has only the red regions.

Also in [15], HSI color space is used to find red regions on the scene. However, this time binary thresholding is applied to each component as follows:

$$\left\{ \begin{array}{l} Hue < 0.027 \quad \text{or} \quad Hue > 0.97 \\ Saturation > 0.6 \\ Intensity > 0.02 \end{array} \right\} \quad (2.3)$$

Where, each component is normalized to one. The output of this thresholding is a binary image. The binary image is next labeled

according to the pixel's 8-connectivity [15]. Each connected component is then eliminated according to the following criterion:

$$\left\{ \begin{array}{l} 200 \text{ pixels} < \text{Area} < 5000 \text{ pixels} \\ 0.4 < \text{AspectRatio} < 1.0 \end{array} \right\} \quad (2.4)$$

A 3-layer Multi Layer Perceptron (MLP) network is used as the next step of the detection module to exclude non-road sign objects. The extracted object is resized to 30x30 pixels and converted to YCbCr color format [15]. Therefore, the network has 30x30x3 (2700) input neurons. 6 neurons are used for hidden layer and 2 for output layer which determines if the input object is a traffic sign or not. Using a classifier at detection step actually decreases some load on classification stage and this decreases the misclassification ratio of the system.

In [16], RGB values are used to detect desired colored regions. The following expression shows the thresholding method, used:

$$\left\{ \begin{array}{l} \alpha_{\min} * G < R < \alpha_{\max} * G \\ \beta_{\min} * B < R < \beta_{\max} * B \\ \gamma_{\min} * B < G < \gamma_{\max} * B \end{array} \right\} \quad (2.5)$$

The values of the parameters involved have been obtained tuning the algorithm on real images in different conditions [16]. This method is preferred because of its simplicity of application. Before the application however, chromatic equalization is applied to suppress the effect of color changes based on the color of light source. For example, during sunset, the dominance of red color increases as shown on Figure 2.2.

To overcome this problem, researchers get the light source color from the temporal window enclosing a predefined region that most likely shows the pave. By integrating the difference between the color inside the window and the color gray (that is the actual color of pave), the light source color is obtained without affected by the fast changes on the obtained color. According to the found color, a correction curve, shown on Figure 2.3, is applied to each channel. Where, parameter  $\alpha$  is the ratio of intermediate channel value over considered channel value. Additionally,  $k$  is set by some empiric tests on sample images, to avoid saturations,  $\beta$  and  $\gamma$  are obtained by making the curve continue and end on point D.



Figure 2.2 Original image and result of chromatic equalization [16].

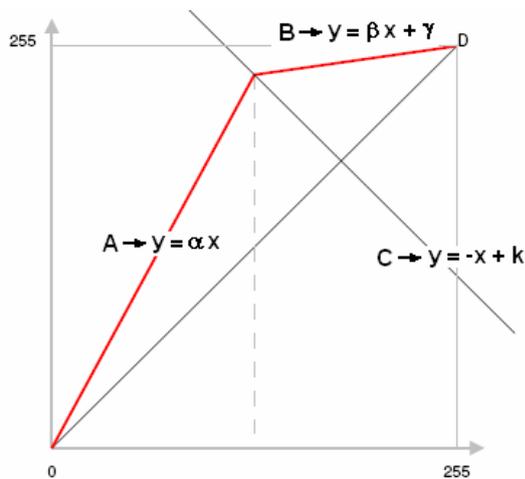


Figure 2.3 Gamma correction curve [16]

In [17], HSV color space is used in detection stage. But this time the color of each pixel on the image is decided by a designed fuzzy system. Figure 2.4 and Figure 2.5 show the used Hue and Saturation membership functions. Using these functions, following fuzzy rules are defined in [17]:

1. If (Hue is Red1) and (Saturation is Red) then (Result is Red)
2. If (Hue is Red2) and (Saturation is Red) then (Result is Red)
3. If (Hue is Yellow) and (Saturation is Yellow) then (Result is Yellow)
4. If (Hue is Green) and (Saturation is Green) then (Result is Green)
5. If (Hue is Blue) and (Saturation is Blue) then (Result is Blue)
6. If (Hue is Noise1) then (Result is Black)
7. If (Hue is Noise2) then (Result is Black)

Figure 2.6 shows the “Result” output variable which consists of five member functions, one for each color [17]. The combination of these three functions, the fuzzy surface is constructed as shown on Figure 2.7. Using this surface like a lookup table, a gray-scale image is constructed from Hue and Saturation values of the input image, where each colored region is defined by a certain level of gray value.

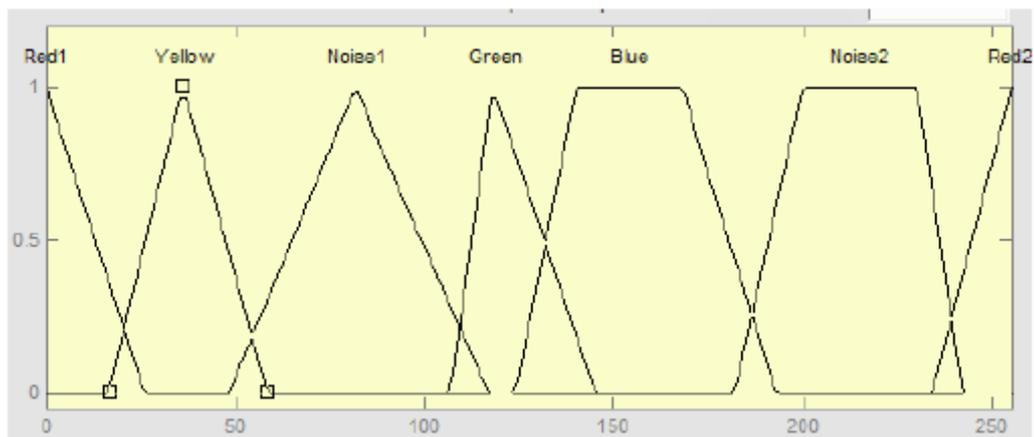


Figure 2.4 Hue membership function [17]

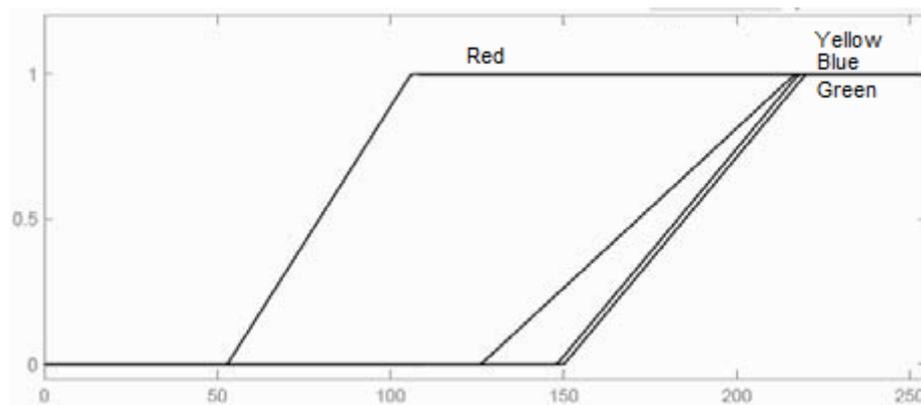


Figure 2.5 Saturation membership function [17]

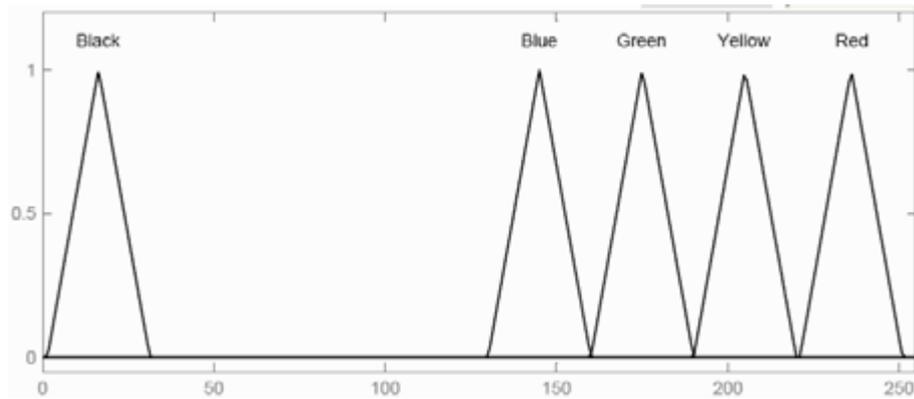


Figure 2.6 The output function [17]

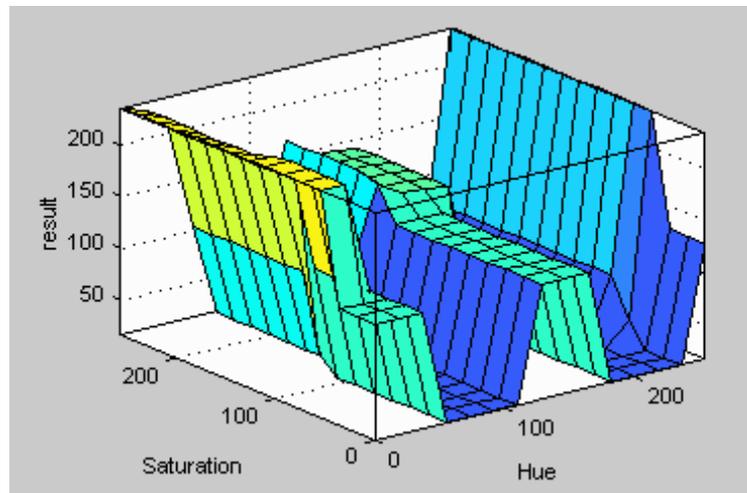


Figure 2.7 The fuzzy system surface [17]

In [18], a region growing technique, based on local thresholds, is used. In this work, three different regions in HSV color space are defined as:

1. Achromatic area :  $S \leq 0.25$  or  $V \leq 0.2$  or  $V \geq 0.9$
2. Unstable chromatic area :  $0.25 < S < 0.5$  and  $0.2 < V \leq 0.9$
3. Chromatic area :  $S \geq 0.5$  and  $0.2 < V \leq 0.9$

where, S and V values are between 0 and 1. In the proposed segmentation method, the chromatic area is fully used, the achromatic area is fully neglected and the unstable chromatic area is considered for region growing within a sub region. Image is filtered for desired color, using HSV color thresholding within chromatic area. Using the found pixels within each predefined sub region, having 34x24 pixels dimension, seed points are defined. Then the filtered pixels are expanded by checking the similarity of each unstable chromatic pixel with the seed pixel defined on that sub region. A pixel is defined as similar to the related seed if the Euclidian distance in the HSV color space is less than a threshold value, defined as:

$$a = k - \sin(S_{seed}) \quad (2.6)$$

Where,  $a$  is the threshold,  $k$  is a parameter to normalize the threshold and  $S_{seed}$  is the seed pixel saturation value. Therefore, for small saturation values the similarity is restricted about the instability effect on hue value.

In [27], two different color segmentation approaches are used with color spaces RGB and HSV. For the first approach, named as “Color 1”, following equations and thresholds are used for detecting red pixels with RGB color space.

$$k = \frac{255}{\max(I_{i,j}^R, I_{i,j}^G, I_{i,j}^B)} \quad (2.7)$$

$$I_{i,j}^{*R} = I_{i,j}^R \cdot k \quad (2.8)$$

$$I'_{i,j}{}^G = I_{i,j}^G \cdot k \quad (2.9)$$

$$I'_{i,j}{}^B = I_{i,j}^B \cdot k \quad (2.10)$$

$$(I'_{i,j}{}^R - I'_{i,j}{}^G \geq 10) \& (I'_{i,j}{}^R - I'_{i,j}{}^B \geq 10) \quad (2.11)$$

The same linear thresholding approach is used on HSV color space to detect red pixels, as follows:

$$((I_{Hue} > 150) \parallel (I_{Hue} < 10)) \& (I_{Sat} > 120) \quad (2.12)$$

Where the range of hue is [0, 180] and of saturation is [0,255].

The second approach, called “Color2”, is based on two look up tables (or detection functions) are used for hue and saturation values of the pixels respectively. Those LUTs are shown on Figure 2.8. Equations of these functions are as follows:

$$hd_{blue} = e^{\frac{-(x-170)^2}{30^2}} \quad (2.13)$$

$$hd_{red} = e^{\frac{-x^2}{20^2}} + e^{\frac{-(x-255)^2}{20^2}} \quad (2.14)$$

$$sd = e^{\frac{-(x-255)^2}{115^2}} \quad (2.15)$$

For each pixel, detection values of hue and saturation are multiplied (the result is named as “hs”) and passed through the following filter:

$$hsn = \begin{cases} 0 & hs < 0.3 \\ \left(\frac{hs - 0.3}{0.7 - 0.3}\right)^2 & 0.3 \leq hs < 0.7 \\ 1 & hs \geq 0.7 \end{cases} \quad (2.16)$$

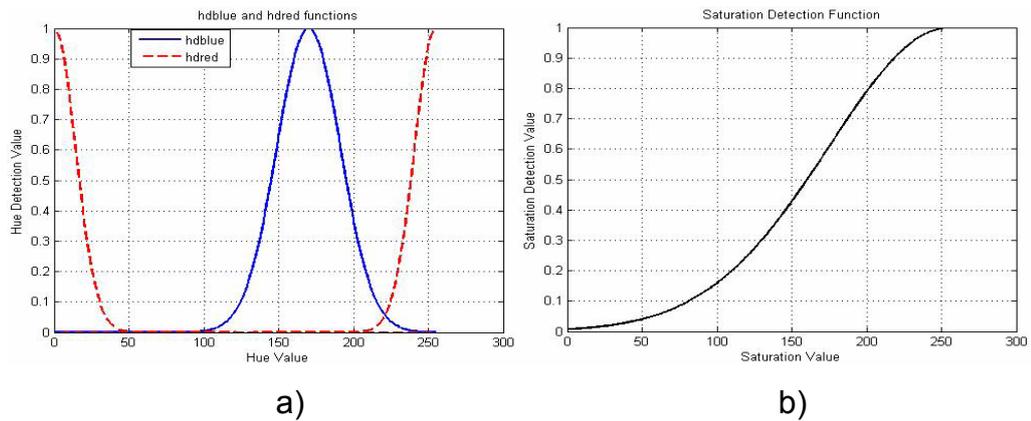


Figure 2.8 a) Detection function on hue value for red and blue pixel detections, b) detection function on saturation value.

## 2.2.2 SHAPE ANALYSIS

Shape information is also a preferred feature to identify traffic signs in an image, because of the instability of color information under bad lighting condition changing during the day time.

[6] includes Canny's edge detection as the first step of its detection stage. Next, found edges are passed thru edge chaining and polygonal approximation algorithm by Pavlidis [7], to obtain straight edges of components. It is assumed that, the vehicle is moving thru the road direction. Since, the traffic signs are orthogonal to that direction; the shape of the sign is not influenced by perspective visualization. Therefore, for triangular traffic signs, by selecting the edges having slopes in the range of  $[-e, +e]$  or  $[60-e, 60+e]$  or  $[-60-e, -60+e]$  degrees (where  $e$  is a predefined error term), the probable traffic sign

edges are found. Finally, edges are checked if an equilateral triangle can be produced by the combination of them. The outputs of each step, shown on Figure 2.9, describe the process better.

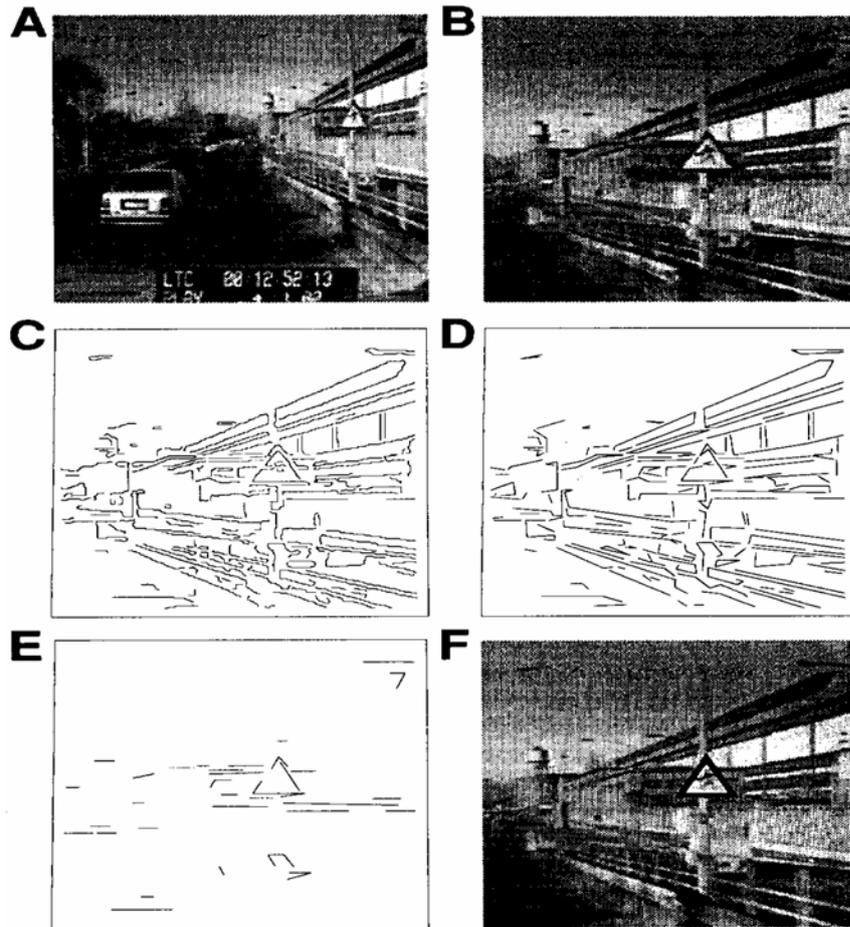


Figure 2.9 (a) is the input image, (b) shows the region of interest, (c) is the output of Canny's edge detection algorithm, (d) is the resulting line segments after algorithm by Pavlidis, (e) shows the selected edges with suitable slopes, (f) is showing the found triangular sign [6]

In [8] corner detection is performed on the mask, which is found by previous color segmentation. This corner detection is based on the convolution of binary image with special masks prepared to find specified corners. Figure 2.10 shows two specified masks to detect upper corner and lower left corner of a triangular traffic sign. The one for lower right corner is the symmetric version of the lower left one, with respect to the vertical axis. In the algorithm, first, the image is scanned until an upper corner is found [8]. Then, a search region, shown on Figure 2.11-a, is defined. That is the most likely region to include the lower left corner. If a corner is found in that region, another search region, shown on Figure 2.11-b, is defined for the lower right one. Same principle is applied for rectangular signs, with masks defined for 90-degree corners. The search regions for the corners of rectangular sign are shown on Figure 2.12-a. For circular signs, masks, defined for 90-degree corners, are used again as shown on Figure 2.12-b. However, this time, after 3 corners are detected the circumference is calculated. If most of the points belong to that circumference, the component is decided to be a circular sign.

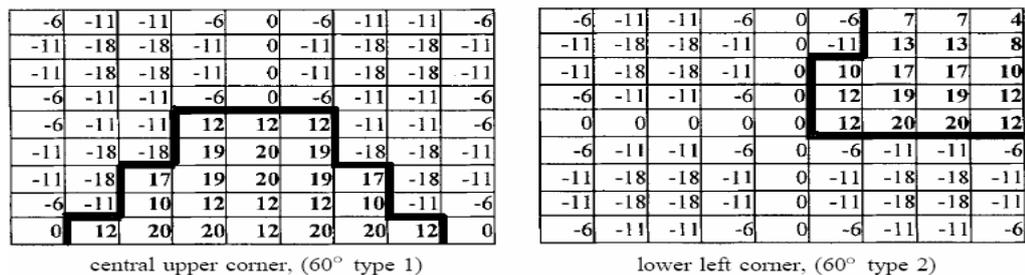


Figure 2.10 [8] Used masks to detect central upper and lower left corners respectively

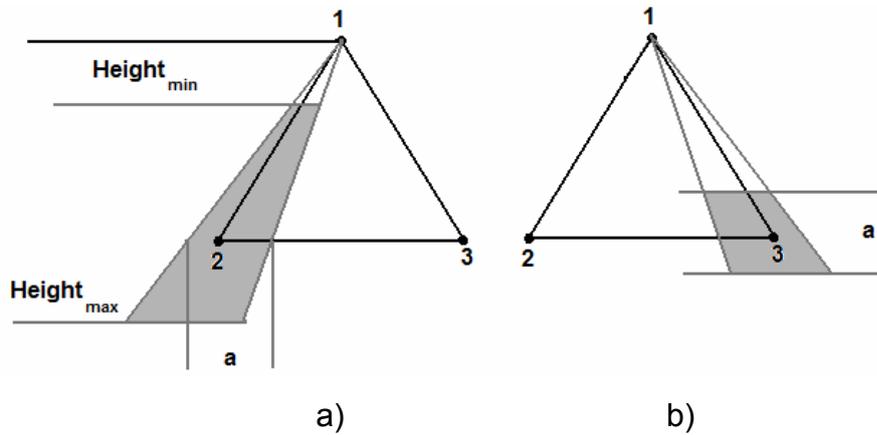


Figure 2.11 [8] Search regions for a triangular sign

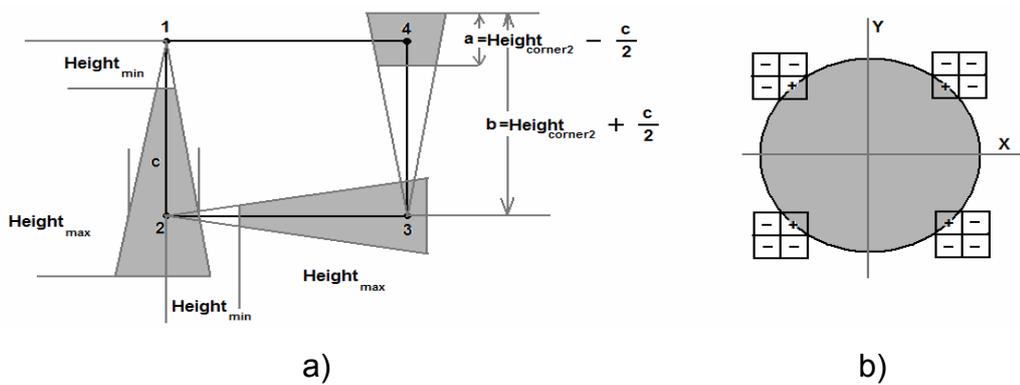


Figure 2.12 a) Search regions for a rectangular sign, b) application of the corner masks to a circular sign [8]

Another method for shape analysis is using Hough Transformation, introduced in [10], to detect lines or curves on the image. Hough transform maps the required points on x-y coordinate, to a parameterized space. For example, a line equation can be written as:

$$x \cdot \cos(\theta) + y \cdot \sin(\theta) = \rho \quad (2.17)$$

Figure 2.13 shows a line and its parameters on x-y plane. By this mapping, each line on x-y plane corresponds to a point on  $\theta$ - $\rho$  plane. In [9], a thinning algorithm is applied to the image after color segmentation, in order to reduce edge thicknesses to 1-pixel. Then, these edges are passed through Hough Transformation to catch line segments those correspond to a triangular sign. Actually, this transformation can be used to find not only line segments, but also other curves, the geometry of which can be parameterized.

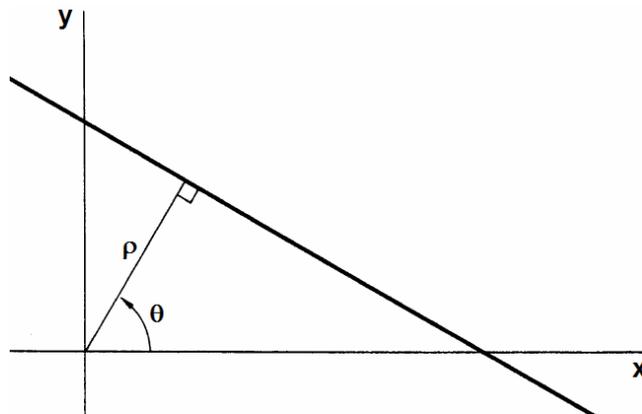


Figure 2.13 The normal parameters for a line [10]

Since, some traffic signs have circular shapes; Circular Hough Transform (CHT) is preferred to detect them on the binary image. A circle equation can be written as:

$$(x - a)^2 + (y - b)^2 = r \quad (2.18)$$

Assuming that the radius is constant, a point at  $(x_0, y_0)$  can be represented as a circle centered at  $(x_0, y_0)$  on a-b plane. This is illustrated on Figure 2.14. The intersection point of each circle gives the center point  $(a, b)$ . Therefore, steps of the algorithm that uses CHT to find circular objects on an image can be summarized as:

- Obtain edges of required parts on the image.
- Draw a circle (with a predefined radius) on the accumulation matrix, for each edge point. Each drawn circle shall be added on top of the previous accumulation matrix, in order to get higher values at intersection points.
- Find the center point of circles by localizing the local maxima of accumulator.

In [12] CHT is implemented on some examples. One simple illustration is shown on Figure 2.15. For details about generalized version of Hough Transform for arbitrary shapes, see [13].

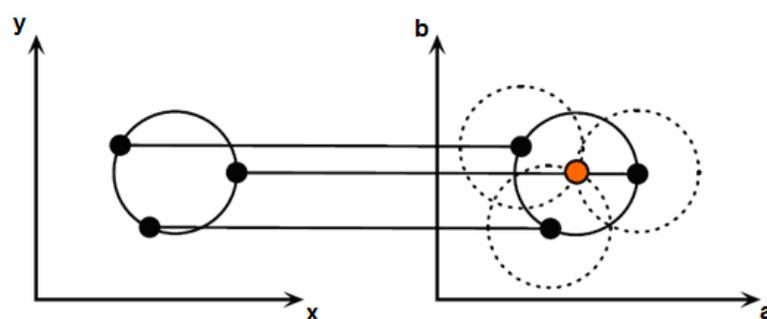


Figure 2.14 Mapping of 3 points on x-y plane to 3 circles on a-b plane  
[11]

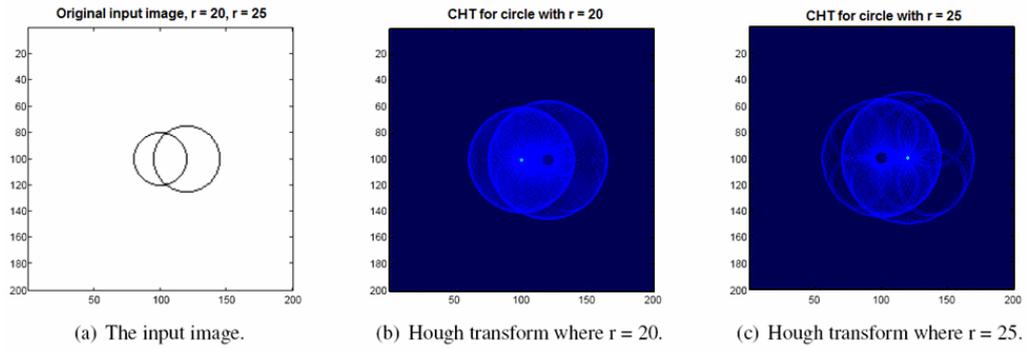


Figure 2.15 An example of accumulator of the CHT with real world data [12]

Besides these shape searching algorithms on the image, in [18], shape of the candidates, found by a color analysis method, are decided by a similarity measure with sample traffic sign shapes. In their research, first the found binary candidate is resized to 36x36 pixels and Tanimoto distance measure is used for similarity measure between the candidate and the sample binary image. To find the maximum similarity measure, sample binary shape image is compared with the candidate image as rotated from -5 to +5 degrees with a step of 1 degree. This measure is defined as follows:

$$d = \frac{\|X \cap Y\|}{\|X \cup Y\|} \quad (2.19)$$

Where, X and Y are set of pixels of the compared binary image and d is the similarity measure with a value between 0 and 1.

In [17], the candidates, found by a previous color segmentation process, are analyzed by a set of fuzzy variables to decide their shapes. These shape measures are, ellipticity, triangularity, rectangularity and octagonality. To define those variables, another variable  $I_1$  is used.  $I_1$  can be written as follows:

$$I_1 = (\mu_{20}\mu_{02} - \mu_{11}^2) / \mu_{00}^4 \quad (2.20)$$

Where,  $\mu_{20}, \mu_{02}$  and  $\mu_{11}$  are the second order central moments, and  $\mu_{00}$  is the area of the object [17]. The ellipticity measure is found as follows:

$$E = \begin{cases} 16\pi^2 I_1 & \text{if } I_1 \leq 1/(16\pi^2) \\ 1/(16\pi^2 I_1) & \text{otherwise} \end{cases} \quad (2.21)$$

Where, E gets 1 if the shape is a perfect ellipse. The triangularity measure T is given by [17]:

$$T = \begin{cases} 108I_1 & \text{if } I_1 \leq 1/108 \\ 1/(108I_1) & \text{otherwise} \end{cases} \quad (2.22)$$

Where, T ranges over [0,1] like ellipticity. Octagonality measure is defined likewise:

$$O = \begin{cases} 15.932\pi^2 I_1 & \text{if } I_1 \leq 1/(15.932\pi^2) \\ 1/(15.932\pi^2 I_1) & \text{otherwise} \end{cases} \quad (2.23)$$

Finally, the rectangularity R is defined as the ratio of the area of the candidate shape to the area of its minimum bounding rectangle.  $R_1$  is

defined as the rectangularity of horizontal aligned objects, while,  $R_2$  is defined as the rectangularity of objects oriented in any other direction. Using these five fuzzy variables, the following rules are constructed:

1. If (R1 is Low) and (R2 is Low ) and (T is One) and (E is Low) and (O is High) then (Shape is Triangle)
2. If (R1 is One) or (R2 is One ) then (Shape is Rectangle)
3. If (R1 is Low) and (R2 is Low ) and (T is High) and (E is Low) and (O is One) then (Shape is Octagon)
4. If (R1 is Low) and (R2 is Low ) and (T is High) and (E is One) and (O is Low) then (Shape is Circle)
5. If (R1 is not One) and (R2 is not One) and (T is not One) and (E is not One) and (O is not One) then (Shape is Undefined)

The membership functions of each fuzzy variable and the output membership function are shown on Figure 2.16 a-f. The sample shape measures are given on Tables 2.1, 2.2, 2.3 and 2.4.

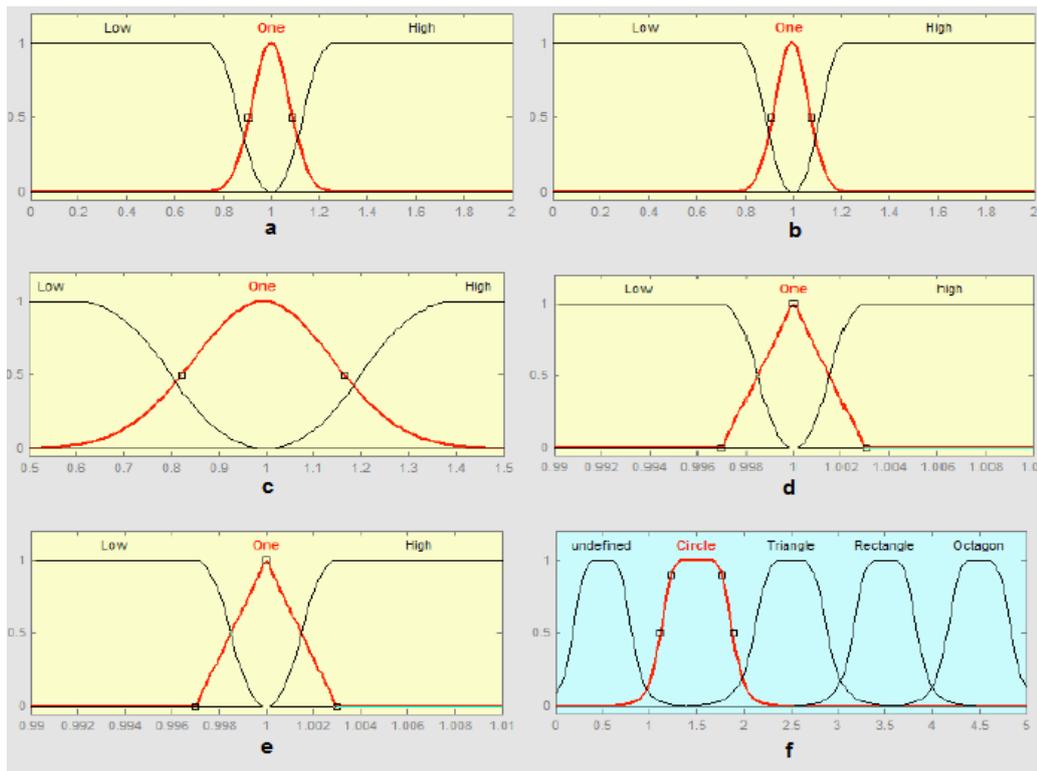


Figure 2.16 a) The  $R_1$  membership functions, b) The  $R_2$  membership functions, c) The  $T$  membership functions, d) The  $E$  membership functions, e) The  $O$  membership functions, f) The output membership functions [17]

Table 2.1 Shape measures of stop sign [17]

R1	R2	T	E	O
0.80208	0.60937	1.455	0.9953	1.0004
0.81985	0.724	1.457	0.99708	0.9986
0.8272	0.8272	1.455	0.99559	1.0002
0.82068	0.75818	1.455	0.99569	1.0001
0.82291	0.5598	1.456	0.99631	0.9994

Table 2.2 Shape measures of yield sign [17]

R1	R2	T	E	O
0.62788	0.59325	1.119	0.76562	1.3006
0.63364	0.58888	1.139	0.77836	1.2793
0.62946	0.58419	1.107	0.75756	1.3144
0.5977	0.59549	1.142	0.78133	1.2744
0.67105	0.66459	1.21	0.82784	1.2028

Table 2.3 Shape measures of no entry sign [17]

R1	R2	T	E	O
0.80252	0.80252	1.46	0.99883	0.9969
0.78971	0.77293	1.462	0.99988	0.9959
0.77659	0.77661	1.461	0.99976	0.996
0.79097	0.76044	1.461	0.9997	0.996
0.78246	0.7755	1.462	0.99993	0.9958

Table 2.4 Shape measures of rectangular signs [17]

R1	R2	T	E	O
0.91632	0.93806	1.331	0.91085	1.0932
0.92316	0.95918	1.335	0.91302	1.0906
0.94446	0.53557	1.336	0.91424	1.0891
0.9566	0.83917	1.339	0.91584	1.0872
0.92637	0.80277	1.337	0.91498	1.0883

### 2.2.3 ANALYSIS OF INDIRECTLY OBTAINED FEATURES

Selection of distinctive features is the major step of any recognition system and it mainly determines the success of the system. In recent years, recognition techniques are preferred for not only classification

stage, but also the detection stage of TSR systems, as proposed methods. Therefore, besides color and shape features, other indirectly found features are used to find existing traffic signs on the image.

In [19], Haar-like features are used to find traffic signs on the image. These features are constructed by subtracting the sum of pixel intensities of two different rectangular regions on the focused sub-region of the gray-scaled image. Sample choices of rectangular regions are shown on Figure 2.17-a. To calculate the features with a little computational cost, integral images are used. For a given image  $I$ , the integral image  $S(x,y)$  can be written as:

$$S(x, y) = \sum_{i=0}^x \sum_{j=0}^y I(i, j) \quad 2.24 [19]$$

Like Figure 2.17-c, the sum  $F(R)$  of pixel intensities in  $R$  can be easily calculated with the form [19]:

$$\begin{aligned} F(R) &= S(r, b) + S(l, t) - S(l, b) - S(r, t) \\ &= (A + B + C + R) + A - (A + B) - (A + C) \end{aligned} \quad 2.25 [19]$$

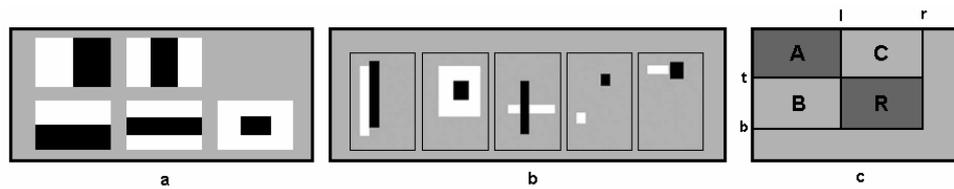


Figure 2.17 Different choices of rectangular regions a) [19] for Haar-like feature, b) [14] as disassociated dipoles. The black region corresponds to the inhibitory dipole, whereas the white region corresponds to the excitatory dipole [14], c) [19] Calculation of features by integral image.

Using the calculated features, several weak classifiers can be constructed to decide if the focused region is a traffic sign or not. Since these classifiers use previously described Haar-like simple features, each has poor performance on classification. However, one strong classifier can be constructed by boosting several good weak classifiers. By using Adaboost algorithm, at each iteration, a good classifier is selected within the set of weak classifier and added to constructed strong classifier with a weight according to its performance. In their research, cascaded structure of weak classifiers is used. In this type of structure, each sub-window passed thru each weak classifier one by one, and at each step if it is decided as a true object, it is sent to the next classifier. If the sub-window is decided to be rejected at any of the classifiers, it is rejected. The structure is shown on Figure 2.18. As a result, while each weak classifier must have very small false negative rate, the false positive rate can be higher.

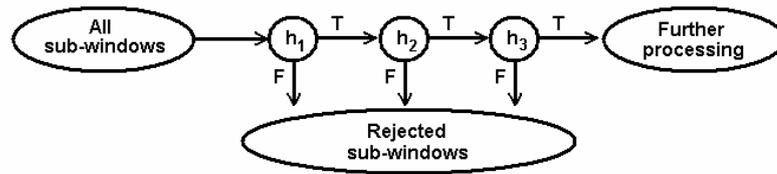


Figure 2.18 Cascaded structure of the designed classifier [19]

Likewise in [14], sub-regions of the captured image are classified if it refers to a traffic sign or not. For this purpose, disassociated dipoles are used to extract wide range of features of sub-regions. As shown on Figure 2.17-b, disassociated dipoles are composed of more generally aligned rectangular regions than the ones for Haar-like features. This fact widens the features those can be extracted from the input sub-window. In [20], Lienhart and Maydt show that the accuracy of the detector increases with the number of available features [14]. Moreover, instead of taking the difference of sum of pixel values, each feature of the sample region is calculated by taking the difference of mean values of the pixel intensities within the inhibitory dipole and the excitatory dipole. Therefore, disassociated dipoles are not affected but the scale of selected rectangles. Figure 2.19 shows some sample feature selections.

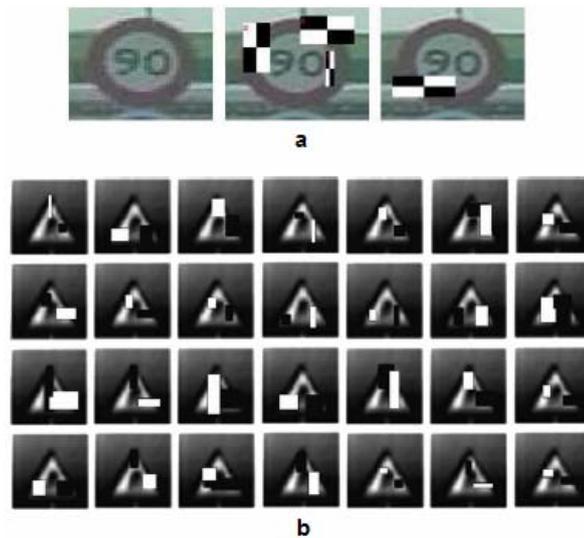


Figure 2.19 a)[19] selected Haar-like features for Speed-Limit sign,  
 b)[14] selected dipoles for danger sign

## 2.3 IDENTIFICATION

In [15], MLP networks are used to classify detected object as one of six traffic signs. The preferred classification module is composed of six separate neural networks, each of which handles one traffic sign. Therefore, each has 2 outputs to decide if the detected object belongs to the particular class or not. This structure makes the classifier expandable. YCbCr color space is used for identification and while training the network, 30x30 images are used.

In [16], traffic signs are considered in 9 categories:

- obligation signs (blue circles),

- prohibition signs (red circles),
- danger signs (red triangles),
- indication signs (blue squares),
- no parking, no waiting signs (red and blue circles),
- stop, no access (red filled circles),
- work in progress signs (red and yellow triangles)
- priority signs (yellow rhombs),
- yield signs (red reverse triangles).

Since the last two categories have one sign each, with the results of detection stage about the shape and color information, these signs are identified without an identification stage. Therefore, 7 neural networks are designed for classification within the first 7 traffic sign categories. Each network composed by 2500 input neurons corresponding to pixels of 50x50 input images. Only one hidden layer is used and the number of neurons vary between 30 and 100 according to experiments on each categories. For training the network, 50x50 synthetic images for each sign are used. Training set is composed of gray-scaled sample images with their rotated and translated versions as shown on Figure 2.20. Before the detected sub-region image is sent to the network, the detected border of fixed-border signs are deleted, then it is converted to gray scale, next it is resized to 50x50, and finally a contrast stretching is applied.



Figure 2.20 Example of variations, rotations and translations, of each model included in the training set. The original model is drawn in the image at the top [16].

In [19], a template matching approach is preferred. Distance transform is used to extract the features of detected sign image for the similarity check. The distance transform of a binary image shows the distance of each pixel to the closest edge pixel on the image. Then, the distance transform of a pixel  $p$  on the image  $R$  can be written as:

$$DT_R(p) = \min_{q \in B_R} d(p, q) \quad (2.26) [19]$$

Where,  $B_R$  is the boundary pixel set and  $d(p, q)$  is the Euclidian distance between pixels  $p$  and  $q$ . In their research, the equation 2.26 is modified to enhance the strength of the equation to distance changes as follows:

$$\overline{DT}_R(p) = \min_{q \in B_R} d(p, q) * \exp(d(p, q)/10) \quad (2.27) [19]$$

Figure 2.21 shows a sample distance transform. Since the informative part of a traffic sign is located at its center, different weights are applied to each pixel of distance transform according to its distance from the center. These weights are found as follows:

$$w_i = \begin{cases} \exp(-|r_i|^2) & \text{if } r_i \leq z \\ 0 & \text{otherwise} \end{cases} \quad (2.28) [19]$$

Where  $z$  is the radius of circumcircle of region  $R$  and  $r_i$  is the distance between the  $i^{\text{th}}$  pixel and the center of region. The similarity can be found using the equation 2.29.

$$S(H, R_i) = \exp(-(\overline{F_H} - \mu_i)\sigma_i^{-1}(\overline{F_H} - \mu_i)^t) \quad (2.29) [19]$$

Where,  $H$  is the candidate image,  $R_i$  is the  $i^{\text{th}}$  training element,  $\overline{F_H}$  is the vectorial representation of the weighted distance transform of sample image  $H$ ,  $\mu_i$  is the mean of vectorial representation of the weighted distance transform of sample training images of  $i^{\text{th}}$  class and  $\sigma_i$  is their variance. The class  $i$ , giving the minimum  $S(H, R_i)$  value, is then decided to be the type of the candidate  $H$ .

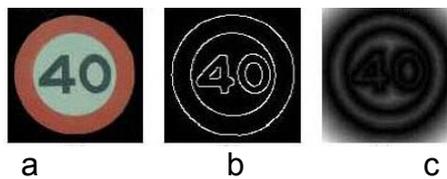


Figure 2.21 Result of distance transform. a) Original image, b) Edge map, c) Distance transform of b) [19]

In [21], Self-Organized Map (SOM) is used as neural network for classification stage. The structure of SOM is shown on Figure 2.22-a. While training SOM networks, according to the winning neuron, which has the highest output for the related input, the weights of neighboring neurons are updated using a neighborhood function. In other words, if a neuron is closer to the winning neuron, its weight update has higher coefficient. The related formula is as follows (from [3] chapter-8):

$$w_{ij}(t+1) = w_{ij}(t) + \eta(t)(u_i - w_{ij})\|u_i - w_{ij}\|N(j,t) \quad (2.30) [3]$$

Where,  $w_{ij}$  is the  $i^{\text{th}}$  weight of the neuron  $n_j$ ,  $u_i$  is the  $i^{\text{th}}$  input value of the neuron  $n_j$ ,  $\eta(t)$  is the learning rate and  $N(j,t)$  is the neighborhood function. The resulting weight update coefficients are illustrated on Figure 2.22-b, for rectangular neighborhood function. In their research, the feature extraction is performed using Gabor wavelets. Sample Gabor wavelets are shown on Figure 2.23 for different orientations and frequencies. In the classification stage, first the input image is converted to HSI color space. Then using a threshold, binary images are obtained for each sign candidate. The next step is convolving the binary image with selected wavelet. The result of the convolution than passed thru two SOM networks, first one is to decide if the candidate is a traffic sign or not and second one is to decide the type of the sign. The structure of designed network is shown on Figure 2.24-b. In their research six types of traffic signs are used. These signs and produced convolution results are shown on Figure 2.24-a (from [3] chapter-8).

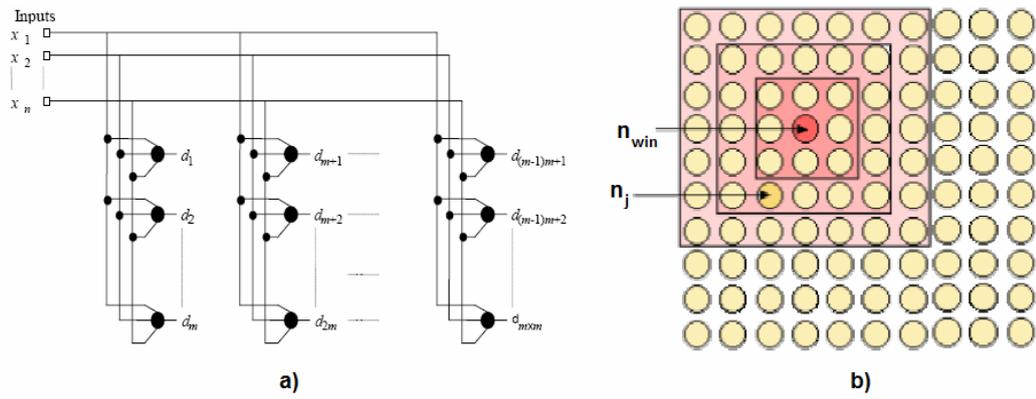


Figure 2.22 a) Network topology of the SOM [3], b) Threshold neighborhood, narrowing as training process [3]

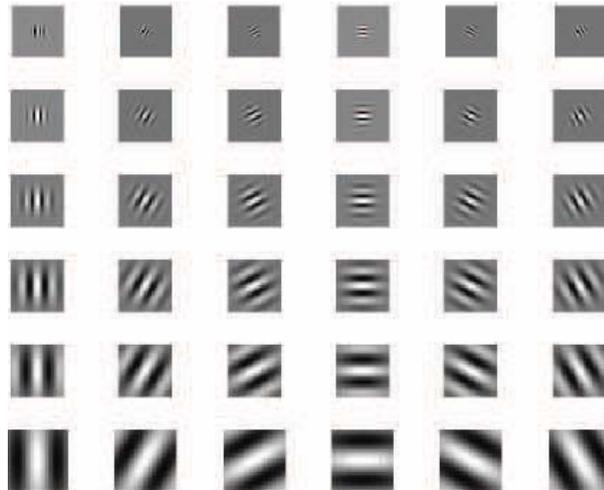


Figure 2.23 Gabor wavelets representations [21]

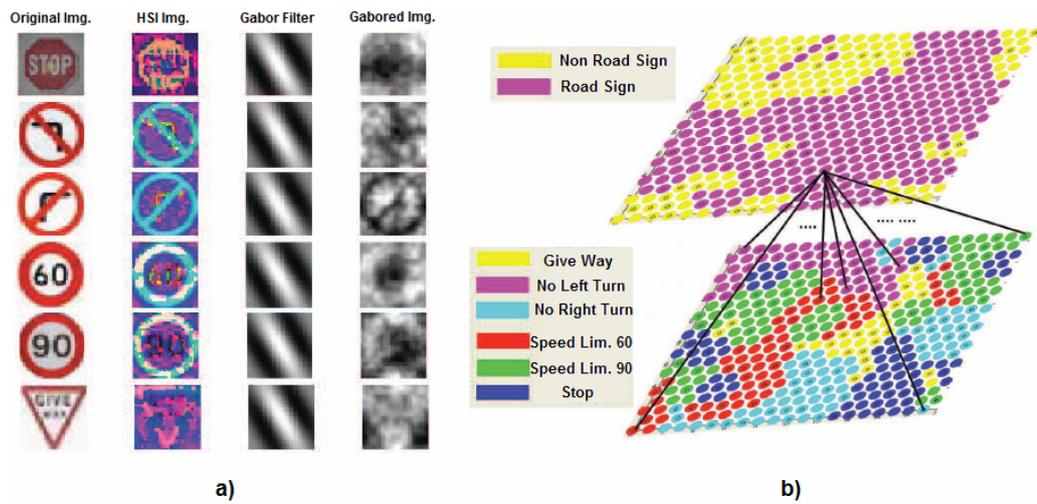


Figure 2.24 a) Intermediate images with one example of Gabor features generated [13] b) Two-tier self organized map with 4 Gabor wavelets [21]

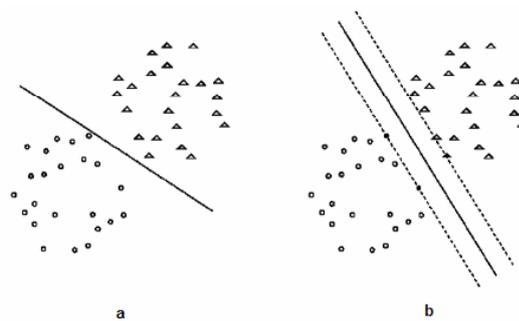


Figure 2.25 Decision boundary obtained by a) an ordinary classifier and b) SVM [23]

In [22], Support Vector Machine (SVM) is used at identification stage. SVM is a classification method that is used to find a decision boundary while trying to maximize the margin between two classes on feature space with minimum risk. A simple illustration of difference between an

ordinary classifier and SVM on a 2D feature space is shown on Figure 2.25. For higher dimensions a hyperplane is defined for discrimination as follows:

$$f(x) = \sum_{i=1}^l \alpha_j y_j K(x_i, x) + b \quad (2.31) [22]$$

Where,  $x$  is the input vector to be classified,  $l$  is the number of training samples, and  $K()$  is known as kernel function [22]. In this research, four different kernels are tested. Those are:

- linear:  $K(x_i, x_j) = x_i^T x_j$
- polynomial:  $K(x_i, x_j) = (\gamma x_i^T x_j + r)^d, \gamma > 0$
- radial basis function (RBF):  $K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2), \gamma > 0$
- sigmoid:  $K(x_i, x_j) = \tanh(\gamma x_i^T x_j + r)$

Where,  $\gamma$ ,  $r$  and  $d$  are kernel parameters [22]. Direct binary image and Zernike moments are used as extracted features separately. 2D Zernike moment of order  $p$  and repetition  $q$  is expressed as follows:

$$Z_{pq} = \frac{p+1}{\pi} \sum_x \sum_y f(x,y) V_{pq}^*(x,y) \quad (2.32) [22]$$

Where,  $x^2 + y^2 \leq 1$ ,  $V_{pq}^*(x,y)$  is circular Zernike polynomials for the region in a unit circle and it is expressed as:

$$V_{pq}(x,y) = R_{pq}(r_{xy}) e^{jq\theta_{xy}} \quad (2.33) [22]$$

Where  $r_{xy} = \sqrt{x^2 + y^2}$  and  $\theta_{xy} = \tan^{-1}(y/x)$

$$R_{pq}(r) = \sum_{k=0}^{(p-|q|/2)} (-1)^k \frac{(p-k)!}{k! \left(\frac{p+|q|}{2} - k\right)! \left(\frac{p-|q|}{2} - k\right)!} r^{p-2k} \quad (2.34) [22]$$

While calculating Zernike moments, first a minimum circle containing the candidate sign is defined, than the pixel coordinates are remapped as if that circle is a unit circle. Found coordinates and binary value is used to calculate the moment. Defining (p,q) parameters from (5,1) to (12,12) gives a 40 dimensional feature vector for each image.

## CHAPTER 3

### THE PROPOSED ALGORITHM

The algorithm actually starts after an RGB image sequence is obtained from an appropriate video source. Therefore, this chapter excludes the sensing stage of the typical pattern recognition problem under the assumption that an RGB image sequence has already been obtained. Figure 3.1 shows a sample input frame to the algorithm.



Figure 3.1 Sample input to the algorithm

In this thesis, red framed traffic signs are used for detection stage and these specific signs are given at Appendix A. For classification stage, only the triangular signs are examined.

### 3.1 SEGMENTATION

Considered traffic signs have two major distinctive properties; those are their color and shape. Therefore, two main approaches exist for segmentation step. Either use color segmentation or segment the image by extracting shape information. Since the considered signs have a main color, red, color segmentation eliminates most of the background information more easily. On the other hand, since the environmental illumination is not controllable for this problem, HSV color space is preferred. Figure 3.2 shows a slice of HSV model with constant hue value. For more information about HSV color map, see [2]. Since, hue and saturation values include the color information, binary thresholding each component with proper thresholds, gives red pixels on the image. Figure 3.3 shows these thresholding graphs. Since each connected component will be examined next, it is necessary to eliminate noises after the thresholding to minimize processing time. For this purpose, the resulting binary image is passed through morphological operation, opening, by a 3 by 3 diamond shaped operator defined as:

$$DiamondOperator = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad (3.1)$$

A sample input image, and its resulting binary images after thresholding and opening operations are shown on Figure 3.4 a, b and c respectively.

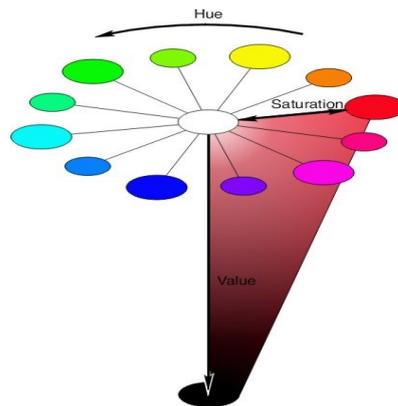


Figure 3.2 saturation/value slice of a specific hue in the HSV model [2]

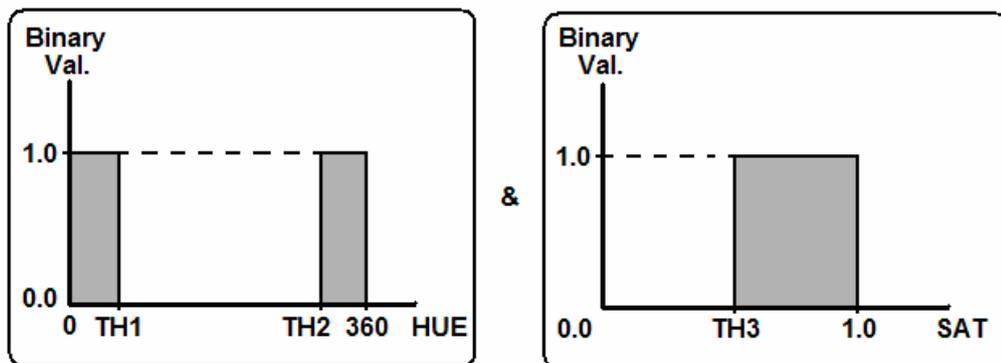


Figure 3.3 Thresholding graphs for each component of HSV (TH1=30, TH2=330, TH3=0.4)



(a)

(b)

(c)

Figure 3.4 a) Sample input, b) output of HSV Thresholding, c) output of morphological opening

### 3.2 CANDIDATE SELECTION

Because of some conditions (such as foggy air, rusty metal of the sign, bad color segmentation, etc...) each connected component found may have a partially occluded shape, defined by one traffic sign. Figure 3.5 shows an example of two, partially observed traffic signs because of a tree branch in front of them. At this point, the shape information shall be constructed with the observed part of the traffic sign. Some methods are based on corner detection to catch three corners of a triangular sign, such as the one in [8]. Then, combining those corners will give a candidate to be identified. However, as for the triangular sign on Figure 3.5, it is more likely to observe edges of the signs without corner.

Moreover, just missing a single corner will make that method fail. Therefore, catching the edges and combining them accordingly, seem to be a more convenient method. “Line Histogram” method is developed for this purpose.



Figure 3.5 Sample of a partially occluded traffic signs.

### 3.2.1 LINE HISTOGRAM ANALYSIS

Line Histogram analysis is based on the idea of linear Hough Transform described in [10]. It is a matrix,  $N^{\text{th}}$  row of which is containing the normalized sum of each pixel values on the row of a binary image that is rotated by an angle of  $N$ . Consider a connected component  $\text{Comp1}$  having a bounding rectangle  $\text{Rect1}$ . The construction of that

component's line histogram is shown in Figure 3.6. As a short description, each row of line histogram shows the accumulation of non-zero pixel values, forming a line segment that has a specific slope for that row.

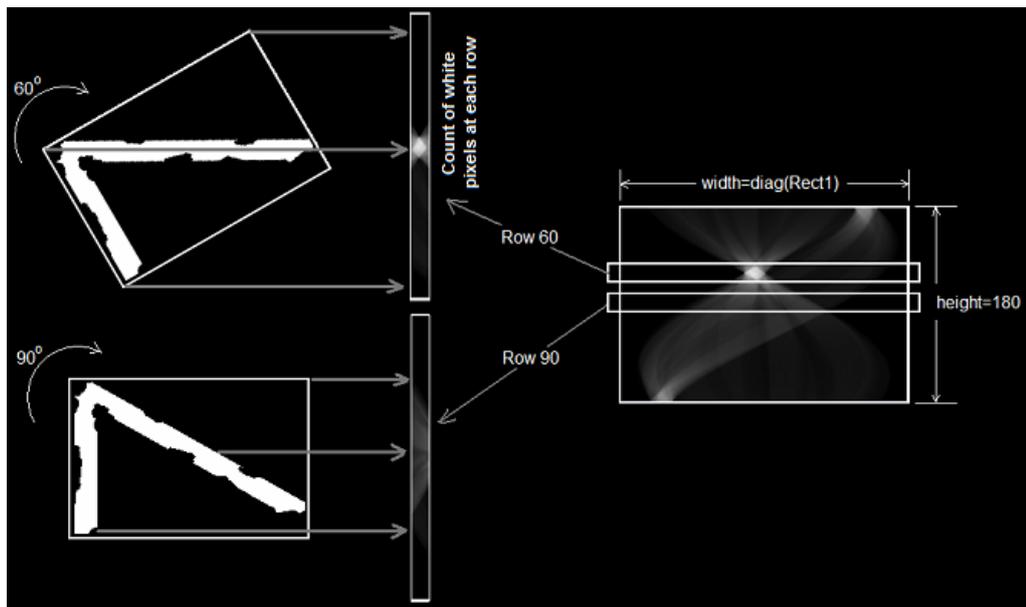


Figure 3.6 Line Histogram of a sample connected component

Let  $Diag1$  be the diagonal length of rectangle  $Rect1$ , which is the bounding rectangle of the connected component. Using the fact that, sum of white pixels cannot exceed the diagonal length of  $Rect1$  for any row and rotation angle; line histogram is constructed to contain values normalized by  $Diag1$ . Therefore, a pixel value of 0.4 on line histogram at coordinates  $(H_x, H_y)$  shows that, after a rotation of connected

component Comp1 with a degree of  $H_y$ , there are  $(0.4 \times \text{Diag1})$  white pixels at row  $H_x$ .

### **3.2.1.1 LINE SEGMENTS**

Assume that, Comp1 has a line segment, Line1, with a length of  $L \times \text{Diag1}$ . Let this line segment has a thickness of  $T$  and an angle of  $45^\circ$ . Figure 3.7 displays Comp1 in white and the information in yellow. As shown on Figure 3.8, the information about line segments of the connected component can be obtained, by an analysis on the local maximum regions of line histogram. For Comp1, the value of local maximum value, which is at  $45^{\text{th}}$  row, is  $L$ . With this information, the existence of a line segment, at degree 45 and of length  $(L \times \text{Diag1})$ , is detected. Moreover, the horizontal range of the local maxima, within a defined value range, gives the thickness of that segment of the component. Because of the thickness of line segment is not 1, a degree range is observed on its line histogram. For line segments, this range is dependent to the thickness and the length of the line. The relation is shown on Figure 3.9.

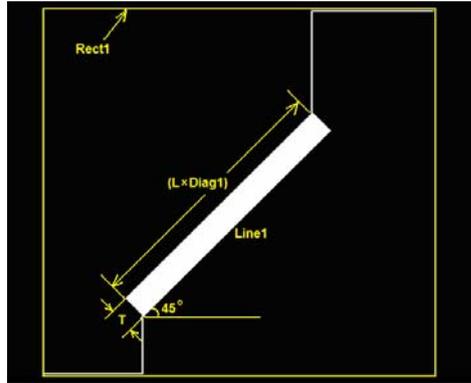


Figure 3.7 Sample connected component, Comp1

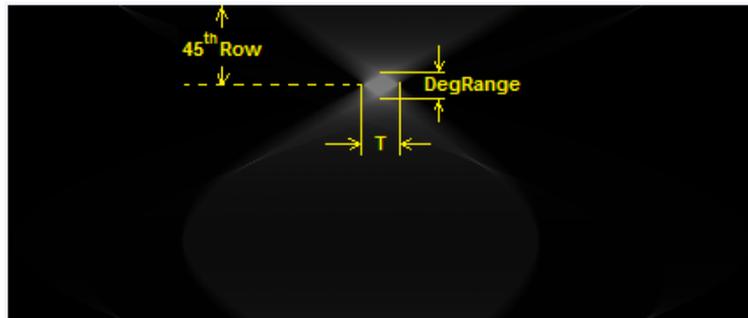


Figure 3.8 Line Histogram of Comp1

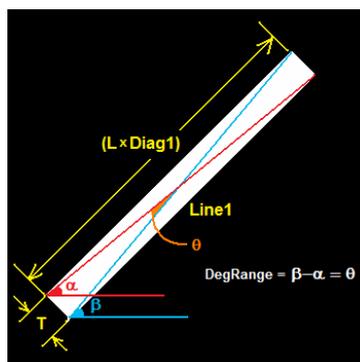


Figure 3.9 The reason of degree range on the line histogram

where,

$$\theta = 2 \times \arctan\left(\frac{T}{L \times \text{Diag1}}\right) \quad (3.2)$$

$$\Rightarrow L = \left( \frac{T}{\text{Diag1} \times \tan\left(\frac{\theta}{2}\right)} \right) \quad (3.3)$$

Assume that, the value at local maxima of the line histogram is  $L'$ . If the found length  $L$ , calculated by equation (3.3), is greater than the local maximum value  $L'$ , the line segment is decided to be broken or have some black spots on it.

The degree of line segment was found previously. To identify that line, an additional information is required. Now, let  $(C_x, C_y)$  be the rotation center coordinates of the connected component. This center is chosen as the center of bounding rectangle  $\text{Rect1}$ . As shown on Figure 3.10, after the component rotated clockwise by  $\phi$  degrees, the point  $(X_1, Y_1)$  on the line will transformed to a new point  $(X'_1, Y'_1) = (C_x, C_y - D)$ . Figure 3.11 shows the centerline, which always include the contribution of rotation center  $(C_x, C_y)$ , and the distance of the local maxima to that line. This distance is equal to  $D$ , which is previously defined as the distance between the rotation center and the line segment. Since  $D$  is known, directly the equation of found line can be written (referring to Equation 2.17) as:

$$x \cdot \cos(90 + \phi) + y \cdot \sin(90 + \phi) = D \quad (3.4)$$

Equivalently,

$$-x \cdot \sin(\phi) + y \cdot \cos(\phi) = D \quad (3.5)$$

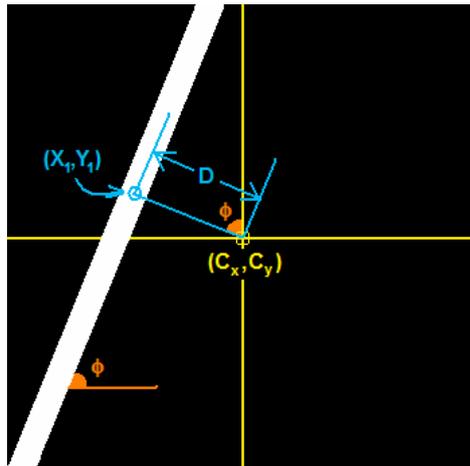


Figure 3.10 Sample connected component Comp2

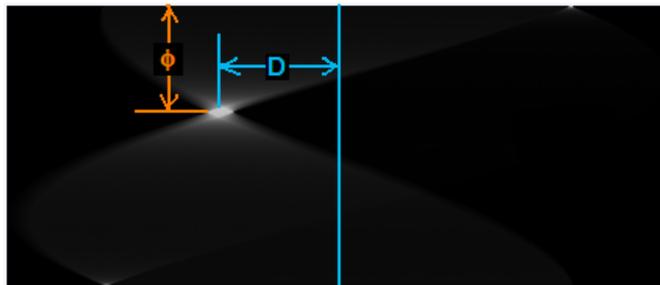


Figure 3.11 Line histogram of Comp2

### 3.2.1.2 ARCS

Let the connected component Comp3 has an arc with an angle of 135 degrees. As shown on Figure 3.12, the arc, having a reasonable thickness, produces local maximum region, having nearly the same value with a much higher degree range, on its line histogram. Three

samples of angles,  $\alpha$ ,  $\beta$  and  $\phi$ , are shown on Figure 3.12. While the line histogram is constructed, after the component is rotated by each of those angles separately, the total count of white pixels at corresponding lines will be nearly the same maximum value for that row. Actually, this fact makes the local maximum region spread through a higher range of angle on the histogram. Therefore, by examining the histogram, the component can be decided to have an arc that covers that degree range  $|\beta - \alpha|$ . Moreover, the center and the radius of the circle, containing that arc, can be found by those local maximum regions. There are 3 unknowns, x and y components of circle center ( $CC_x, CC_y$ ) and inner radius R. Therefore, 3 equations are necessary. The equations of each line at the three sample angles,  $\alpha$ ,  $\beta$  and  $\phi$ , can be used. The equation of distance of a point  $(X_p, Y_p)$  to a line,  $ax+by-c=0$ , is as follows:

$$Dist = aX_p + bY_p - c \quad (3.6)$$

Therefore, in order to find  $CC_x$ ,  $CC_y$  and R1, by using equations (3.5) and (3.6), required 3 equations can be written, according to Figure 3.12, as follows:

$$D_\alpha = -CC_x \sin(\alpha) + CC_y \cos(\alpha) - R \quad (3.7)$$

$$D_\beta = -CC_x \sin(\beta) + CC_y \cos(\beta) - R \quad (3.8)$$

$$D_\phi = -CC_x \sin(\phi) + CC_y \cos(\phi) - R \quad (3.9)$$

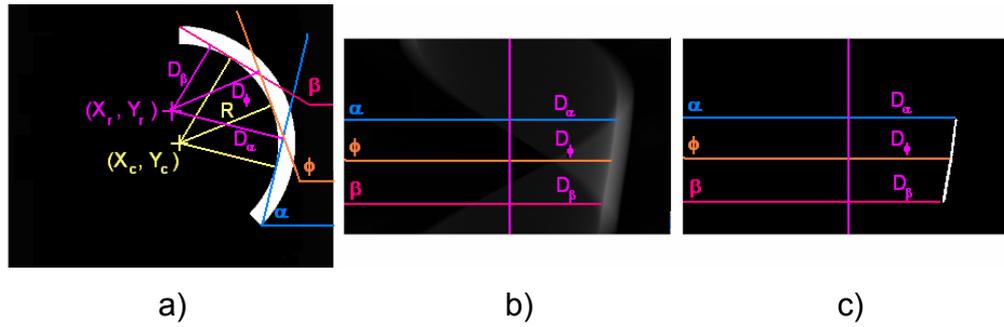


Figure 3.12 a) Sample connected component Comp3 containing an arc, b) Line histogram of Comp3, c) Local maximum region of line histogram

Three equations, 3.7, 3.8 and 3.9 can be represented as a matrix equation form ( $AX=B$ ) as follows:

$$\begin{bmatrix} -\sin(\alpha) & \cos(\alpha) & -1 \\ -\sin(\beta) & \cos(\beta) & -1 \\ -\sin(\phi) & \cos(\phi) & -1 \end{bmatrix} \begin{bmatrix} CC_x \\ CC_y \\ R \end{bmatrix} = \begin{bmatrix} D_\alpha \\ D_\beta \\ D_\phi \end{bmatrix} \quad (3.10)$$

Therefore, our unknown vector  $X$  can be found as:

$$\begin{bmatrix} CC_x \\ CC_y \\ R \end{bmatrix} = \begin{bmatrix} -\sin(\alpha) & \cos(\alpha) & -1 \\ -\sin(\beta) & \cos(\beta) & -1 \\ -\sin(\phi) & \cos(\phi) & -1 \end{bmatrix}^{-1} \begin{bmatrix} D_\alpha \\ D_\beta \\ D_\phi \end{bmatrix} \quad (3.11)$$

Not to have a singularity, the differences between any two of the angles  $\alpha$ ,  $\beta$  and  $\phi$  shall not be equal to 0 or 180 degrees.

### 3.2.1.3 COMBINED COMPONENTS

In this section, connected components those include several line segments or arcs, are analyzed. If a traffic sign is fully observed, it will generally consist of several parts, such as, circular edges, internal cross lines or external linear edges. Even for combined components, the segments can be separately analyzed by the help of line histogram. Figure 3.13 shows a component consisting of a ring shaped boundary and two inner line segments. On its line histogram, the ring boundary produces two local maximum regions with 180 degrees of range at both sides. Thinner line segment is observed at 45<sup>th</sup> row and the thicker one at 135<sup>th</sup> row of the histogram. By just analyzing line histogram, those three segments of the connected component can be identified. On Figure 3.14, a triangular component is shown with its line histogram. The local maximum regions implies the existence of three line segments those at 0° (or 180°), 60° and 120° degree respectively.

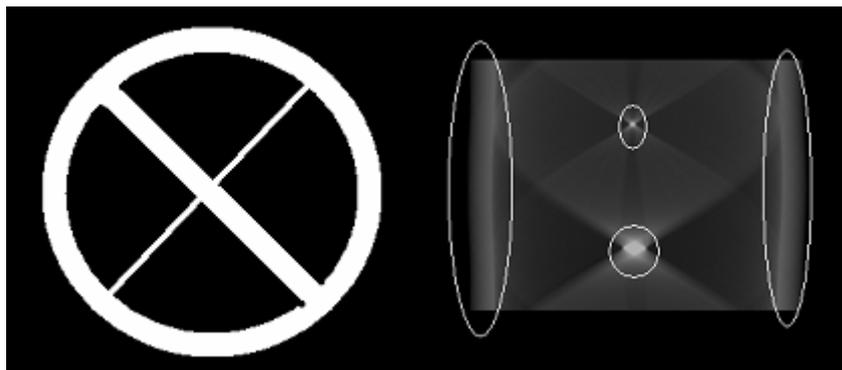


Figure 3.13 Sample for a combined component and its line histogram

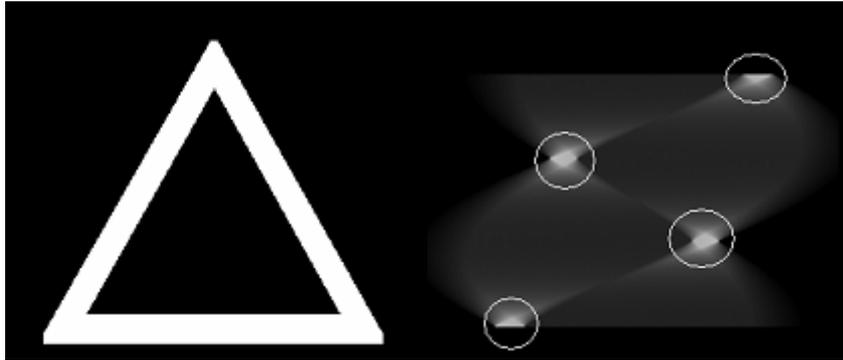


Figure 3.14 Sample of a connected component that belongs to a triangular traffic sign and its line histogram

#### 3.2.1.4 NOISY COMPONENTS

Since the visual conditions are not good enough to get perfectly segmented components, the developed method shall also apply to noisy components. As shown on Figure 3.15, line histograms can still identify the edges of the triangular sign at 0, 60 and 120 degrees, and the circular boundary of the speed limit sign.



Figure 3.15 Two samples of traffic signs with bad visual conditions, their segmented major components and their line histograms

### 3.2.2 DECISION OF TRIANGULAR AND CIRCULAR CANDIDATES

For a triangular sign, three line segments, with slopes 0 or 180, 60 and 120 degrees, are necessary. Since the captured sign may have different orientation, the slopes cannot be observed exactly at those degrees. Therefore, if a connected component has:

- Line segment with slope in the range  $[-25,25]$ , then assume it is the lower edge of a triangular sign.
- Line segment with slope in the range  $[35,85]$ , then assume it is the left edge of a triangular sign.
- Line segment with slope in the range  $[145,95]$ , then assume it is the right edge of a triangular sign.

Using these 3 edges, each corner can be found by computing the intersection points of each. As an exception, If the corner defined by left and the right edges of triangle, lies above the other two corners, then the component is decided to be a yield sign candidate.

On the other hand, if a component has at least one local maximum region with a degree range of above 30 degrees on its line histogram, then it is decided that the component has an arc and it is considered as a circular traffic sign candidate. Calculating the circle defined by the arc gives the region of the candidate.

The results of the algorithm on two occluded traffic signs are shown on Figure 3.16.

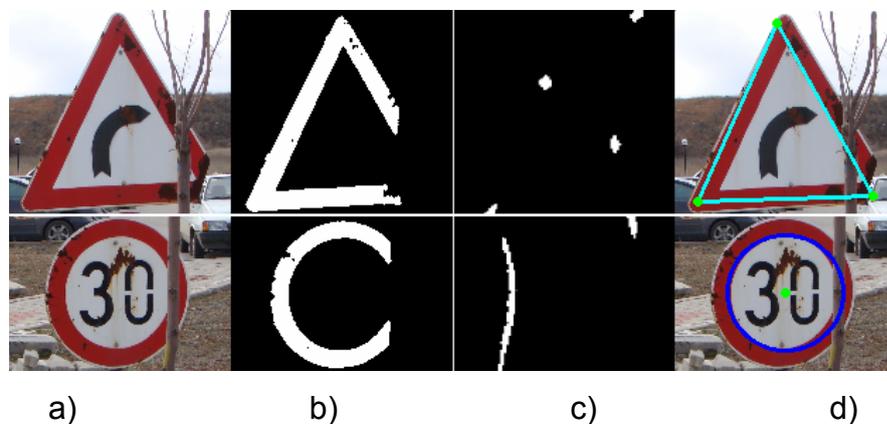


Figure 3.16 a) Occluded traffic signs, b) corresponding connected component, c) local maximum regions on their line histograms, d) extracted line segments and circle containing the arc.

Additionally, if a circular sign is observed as an ellipse on the image because of the camera position, then its line histogram gives different arcs defining different circles as shown on Figure 3.17.

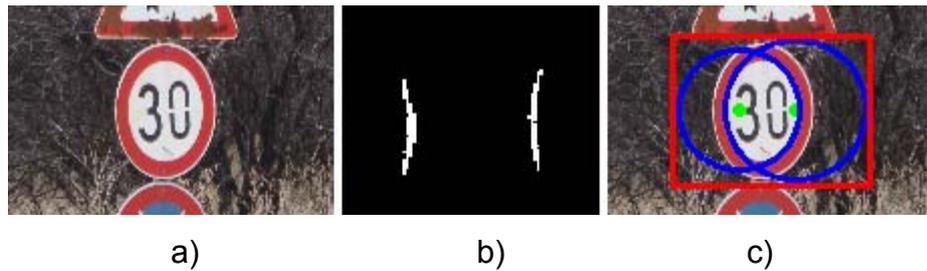


Figure 3.17 a) Sample circular sign, b) two local maximum regions, c) two different circles found by each local maximum regions.

### 3.2.3 RESIZING AND SKEWING THE COMPONENTS

If a found component is decided to be a candidate, it shall be transformed to a predefined size and a template pattern according to its shape. Used traffic sign set includes 50 by 50 pixels sample images. Therefore, for example, a found triangle shall be transformed such that the upper corner of it shall coincide with the mid point of the upper edge, and the lower two corners shall coincide with the lower two corners of 50x50-pixel image. For this purpose, `cvGetAffineTransform()` function of OpenCV Library is used. This function gets 3 source points (which are the found corner points of the triangle) and 3 destination points (which are the mid point of the upper edge and two lower corners

of the 50x50-pixel image). Then the function returns a transformation matrix. The application of this matrix to the bounding rectangle of the found triangular sign gives a well-positioned bilateral triangle with the predefined size. Figure 3.18 illustrates this transformation for a triangular traffic sign.

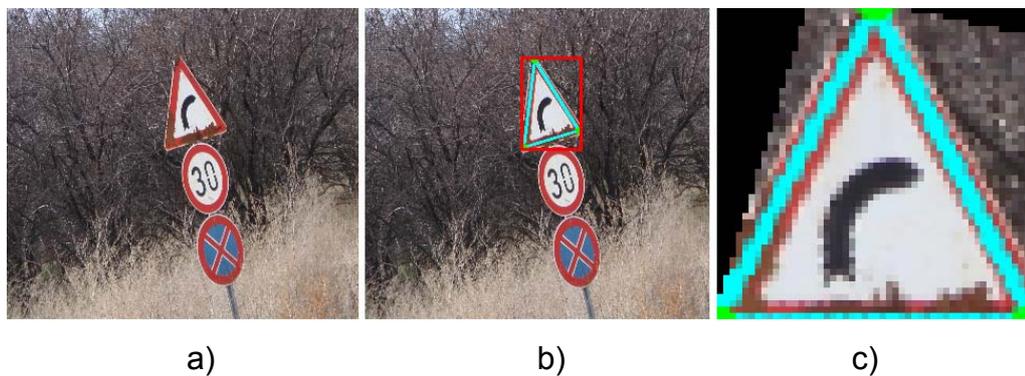


Figure 3.18 a) Captured traffic signs, observed as rotated and compressed, b) the found triangular sign, c) resulting 50x50-pixel candidate after transformation.

### 3.3 IDENTIFICATION

#### 3.3.1 FEATURE EXTRACTION

At the previous stage, the candidates are found including the shape information. Since, the shapes are known, the informative part of the traffic signs remains to be identified. Those parts are defined by the

internal drawings of the signs. Since the boundaries are obtained by the previous stage, internal drawings can be segmented by masking the real frame with the filled component boundary. A sample result of this process is shown on Figure 3.19. Since, pixel values are used at classification step, the input image shall be normalized according to the white and the black regions. This normalization is done by examining the maximum and minimum pixel intensities of gray-scaled image within the triangular mask. After this procedure, intensity of the black region (the pictogram of the traffic sign) becomes nearly 0 while the one of white region becomes nearly 1.

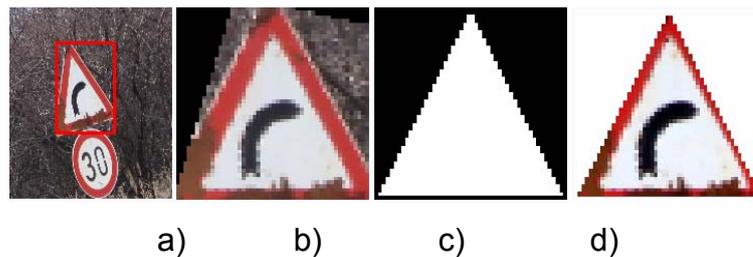


Figure 3.19 Sample segmentation of the informative part of the sign, a) Found candidate on the frame, b) affined candidate, c) default mask for triangular signs, d) the resulting input for the classification stage.

For identification stage, two different approaches, template matching and neural networks, are used with following 4 different feature sets:

1. Binary image constructed by thresholding RED pixel values of the 50x50 RGB candidate image, for neural networks approach,

2. RGB pixel values of the 50x50 RGB candidate image, for both template matching and neural networks approaches,
3. Gray-scaled pixel values of the 50x50 RGB candidate image, for neural networks approach,
4. Distance Transform of binarized candidate image, for both template matching and neural networks approaches.

The first feature set is constructed by simply taking only the red component of the candidate and thresholding it by a threshold value 0.5. The second set is just the RGB image itself. The third one is simply defined by the intensity values of each pixel, which are the mean of red, green and blue components. On the other hand, the last feature set, Distance Transform of a binary image, is a gray-scaled image with intensity values showing the distance of the corresponding pixel on the binary image, to the closest non-zero pixel. These four sets of extracted features are shown on Figure 3.20 b, c, d and e respectively.

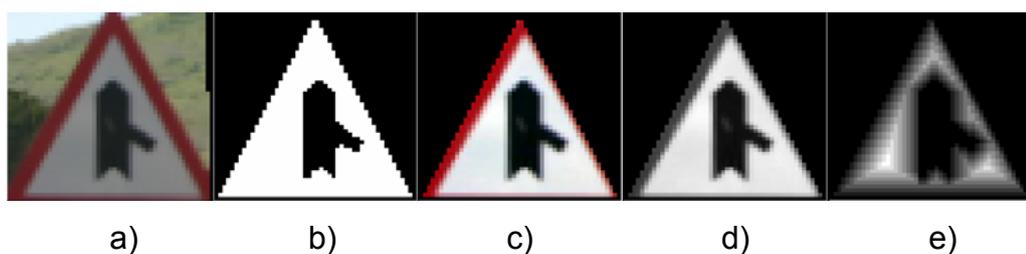


Figure 3.20 a) Affined candidate, b) First feature set: red component of RGB image, c) Second feature set: RGB image itself, d) Third feature set: Gray-scaled image, e) Forth feature set: Distance transform of binary image.

### 3.3.2 TEMPLATES AND TRAINING SETS

For template matching approach, templates consist of 36 triangular traffic signs defined at [25]. Since the processing time is very high while comparing each pixel on the candidate with the one on lots of templates, each template image is preferred as the perfect copy of a triangular sign. On the other hand, for neural networks, the computation time is proportional to the number of neurons. It is not directly proportional to the number of the members of the training set. Therefore, the training set also includes deformed versions of each triangular sign in addition to their perfect copies.

The candidates are passed to identification state as affined to a known size and orientation. However, because of some error at previous stages, which are the errors while calculating the corner points of the triangular sign, the candidate may oriented as a little shifted or rotated with respect to the original triangular shape. Moreover, the noise also affects the input of identification stage. Therefore, the training set includes the following versions of 36 triangular signs and all of their combinations:

1. Perfect copies of 36 triangular signs,
2. Rotated versions, by  $\pm 3, \pm 6$  degrees,
3. Shifted versions to 8 directions, by 3 pixels,
4. Noisy version with additional random noise with a normal distribution with mean=0.0 and standard deviation=0.1 and 0.2.

As a result, total training set includes  $(4+1)*(8+1)*(2+1) = 135$  versions for each of 36 triangular signs. This yields 4860 total images in

training set. The sample rotated, shifted and noisy versions are shown on Figure 3.21, Figure 3.22 and Figure 3.23 respectively.

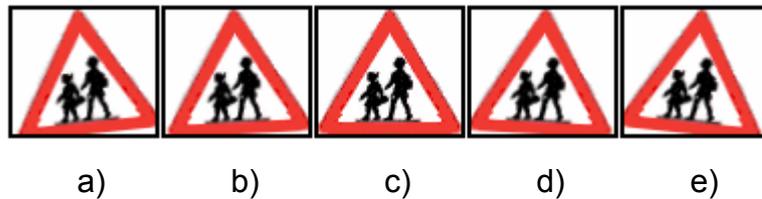


Figure 3.21 a) -6 degrees rotated version, b) -3 degrees rotated version, c) original version, d) +3 degrees rotated version, e) +6 degrees rotated version of a sample sign.



Figure 3.22 original version at the center and shifted versions to 8 directions.

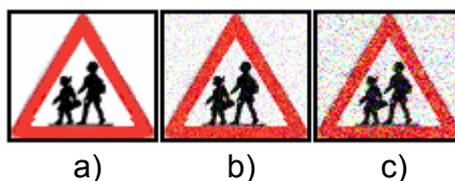


Figure 3.23 a) the original version, b) noisy version with standard deviation 0.1, c) with standard deviation 0.2.

### **3.3.3 CLASSIFICATION**

#### **3.3.3.1 TEMPLATE MATCHING**

In template matching approach, the input candidate is compared by each of 36 traffic signs. This comparison is done by normalized cross correlation of the candidate and the template. Since the templates are the original versions of the signs, candidates are shifted by 1 to 5 pixels to all 8 directions and the maximum correlation value is selected as the similarity value, in order to minimize the errors on calculations of the exact location of triangular sign. For this purpose, `cvMatchTemplate()` function is used. The candidate is sent to the function as centered on a bigger (expanded 5 pixels to each directions) image. Therefore, the output includes 11x11 pixel normalized correlation results for each of the shifted versions. Selecting the best (maximum) result gives the appropriate similarity. Full match gives 1.0, while a full mismatch results a 0.0. In the algorithm, with a maximum similarity measure, less than 0.9, is decided as an unknown traffic sign.

#### **3.3.3.2 NEURAL NETWORKS**

There are 36 classes. Therefore, the designed neural network shall have the capability of classifying each of 36 classes. The first network, tried, was designed with 2 hidden layers with 100 and 50 neurons respectively. It had 1152 neurons at its input layer. This number is the number of white pixels on 50x50-pixel triangular mask image shown on Figure 3.24. Since the background information is unnecessary and differs from time to time, each candidate is considered only with the pixels coinciding with the white pixels of the mask image. On the other

hand, the number of neurons at the output layer was 36. However, training one neural network having 36 output neurons, with 4860 images was a time consuming process. Additionally, without increasing number of neurons at the hidden layers, it was impossible to train the network with all images in the training set. Therefore, 36 classes are separated into 4 class sets, by “Normalized Cuts” method detailed in [26].

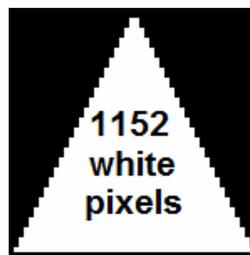


Figure 3.24 50 by 50 pixel triangular mask image.

“Normalized Cuts” is simply a method to separate a group of graphs defined in a feature space, into two sets, those include the most similar members within the same set, while separating the most different members. In order to make the separation, a similarity measure shall be defined. Therefore, normalized cross correlation is selected as the similarity measure. The feature space is 1152 dimensional space showing each pixel’s intensity at its each dimension. Exceptionally, if the extracted features of the candidate image are selected as RGB values, than the feature space becomes a  $3 * 1152 = 3456$  dimensional space.

Assume a 36x36 matrix A is defined by the similarities of each graph with another. In other words, the similarity of graph i and graph j is written at the i<sup>th</sup> row and j<sup>th</sup> column of the similarity matrix A. Now, assume another 36x36 matrix D, containing the sum of each row of similarity matrix A, as its diagonal entries, and 0 as the other entries. Another 36x36 matrix W is the same as A except its all diagonal entries are replaced by 0. Now according to the normalized-cuts method, the eigen-vector of the 2<sup>nd</sup> minimum eigen-value of the generalized eigen-value system Equation 3.12, separates the graphs.

$$(D - W)y = \lambda Dy \quad (3.12)$$

Applying this method to both groups one more time gives four separate groups of traffic signs.

For the distance-transform images, the similarity matrix is shown as an image on Figure 3.25 – a. The first cut, divides 36 classes into two groups of 23 and 13 classes respectively. For the second application of normalized-cut, the similarity matrices are shown of Figure 3.25 – b and c respectively. As a result, the whole set is divided into 4 groups having 12, 11, 7 and 6 members.

On the other hand, for other feature sets (red, gray and RGB), the similarity matrices are the same, with scaled versions of each other. These matrices are shown on Figure 3.26. Therefore, the separated groups for these three feature sets are the same. This time 36 classes are separated into 4 groups having 10, 10, 9 and 7 members respectively.

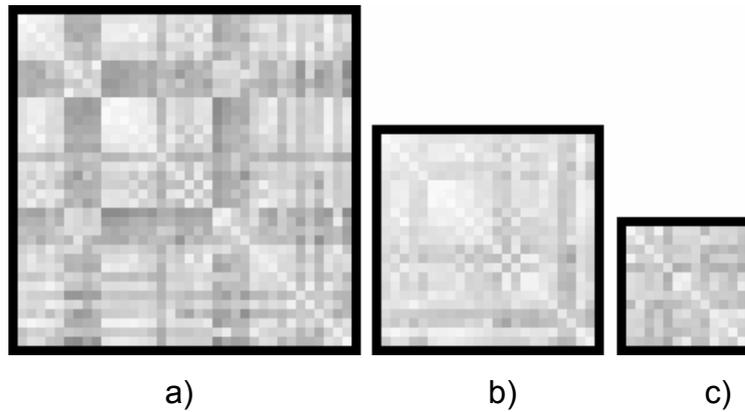


Figure 3.25 a) Similarity matrix  $A_{DT}^0$  for distance transform images, b) Similarity matrix  $A_{DT}^{11}$  for the first group after the first segmentation, c) Similarity matrix  $A_{DT}^{12}$  for the second group after the first segmentation.

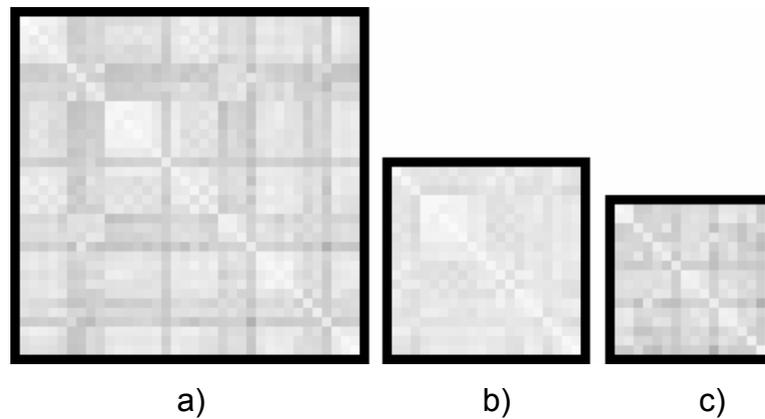


Figure 3.26 a) Similarity matrix  $A_{RGB}^0$  for feature sets, red, gray and RGB, b) Similarity matrix  $A_{RGB}^{11}$  for the first group after the first segmentation, c) Similarity matrix  $A_{RGB}^{12}$  for the second group after the first segmentation.

According to these groups, 7 neural networks are constructed for each feature sets as follows:

- The first network  $Net_0$  has (1152, 50, 25, 2) neurons at each of its layers, in order to classify the candidate according to the first normalized-cut.
- The second one,  $Net_{11}$ , has (1152, 50, 25, 2) neurons at each of its layers and decides if the candidate is a member of the 1<sup>st</sup> or the 2<sup>nd</sup> group ( $Grp_{11}$  and  $Grp_{12}$ ).
- The third one,  $Net_{12}$ , has (1152, 50, 25, 2) neurons at each of its layers and decides if the candidate is a member of the 3<sup>rd</sup> or the 4<sup>th</sup> group ( $Grp_{21}$  and  $Grp_{22}$ ).
- The last four networks,  $Net_{21}$ ,  $Net_{22}$ ,  $Net_{23}$  and  $Net_{24}$ , have (1152, 50, 25, X) neurons at each of their layers, where X is the number of members for each 4 groups. And each classifies the candidate within its group.

As an exception, for RGB feature set, the created networks have  $3 \times 1152$  neurons at their input layers.

According to the output of  $Net_0$ , the next level network is decided as  $Net_{11}$  or  $Net_{12}$ . In other words, if the two outputs of  $Net_0$  is [1,-1], then the candidate is a member of one of the groups  $Grp_{11}$  or  $Grp_{12}$ . Therefore, the next network will be  $Net_{11}$ , in order to decide the group. If the outputs of  $Net_0$  is [-1,1], then the candidate is a member of one of the groups  $Grp_{21}$  or  $Grp_{22}$ . Therefore, the next network will be  $Net_{12}$ . As the last level, according to the previous network's output, the last

network is selected as one of the networks  $Net_{21}$ ,  $Net_{22}$ ,  $Net_{23}$  and  $Net_{24}$ . Therefore, for each candidate 3 neural networks are used to classify it.

The transfer function of each neuron is selected as  $Tanh(x)$ . For training, "Back Propagation Algorithm", described in [3] chapter-6, is used.

## CHAPTER 4

### EXPERIMENTS AND RESULTS

The algorithm is tested on two image sets, Set-1 and Set-2. Set-1 includes 20 images showing 21 red-framed triangular signs. All samples in that set are well illuminated. On the other hand, Set-2 includes 309 images including red-framed triangular and circular traffic signs. Some of these images are duplicate of others. This set is separated into 10 groups, described as follows:

- **Cluster:** This group includes 38 images. The samples on this set include more than one traffic sign, either on top of or next to each other.
- **Deformed:** In this group, 20 images show several vandalized or broken signs.
- **Mixed:** 33 images in this set show samples of different sign types and distortions in each image.
- **Different Shapes:** 41 images in this set show samples of signs having circular and triangular shapes.
- **Illumination:** 34 images in this set include samples of bad illuminated traffic signs.

- **Occlusion:** 34 images show samples of partially occluded traffic signs.
- **Rotation:** 11 images in this set include signs, rotated to different directions according to the viewing direction.
- **Shadows:** 30 images show samples of traffic signs, which are mostly in shade.
- **Sizes:** 35 images in this set include samples of scenes captured from different distances to the signs.
- **Translation:** 33 images show samples of signs observed on different parts of the scene.

Set-2 is obtained from [24]. Each traffic sign on the images is assumed as a candidate. In other words, there are no constraints on counting the traffic signs. Sample images of Set-1 and Set-2 are shown on Figure 4.1.



Figure 4.1 a) Sample image from Set-1, b) another sample from Set-1, sample images from Set-2 c) Cluster group, d) Deformed group, e) Mixed group, f) Different Shapes group, g) Illumination group, h) Occlusion group, i) Rotation group, j) Shadows group, k) Sizes group, l) Translation group

## 4.1 DETECTION

The detection results for Set-1 are shown on Table 4.1. First row shows the result of Line Histogram algorithm described in this thesis. Second row shows the result of detection by using OpenCv `cvHoughLines2()` function, which implements line detection by Hough Transform. Last two rows give the results of [27] for the same set.

For Set-2, the detection results of the proposed algorithm are shown on Table 4.2. Table 4.3 shows the detection results by using OpenCv `cvHoughLines2()` and `cvHoughCircles()` function on detection stage. In [27], Set-2 is also used for several detection algorithms and the results are shown on Table 4.4.

The sign types, considered in this thesis, are only the red-framed circular and triangular ones, which can exist everywhere in the picture. Since in [27], only the signs on the main road are considered, the total number of signs shown on the tables can be different. The listed detection algorithms of [27] on Table 4.1 and Table 4.4 are mentioned at 2.2.2. Additionally, in [27], the results are optimized by setting the algorithm parameters according to the specific set of data. These optimized results are declared as “(optimized)” on the tables.

Table 4.1 Set-1 Detection Results

	TOTAL	DETECTED	% of DETECTION	FALSE DETECTION
Line Histogram	21	21	100	2
Hough Lines	21	19	90.5	2
Shape Algorithms [27]	19	14	73.7	1
Shape Algorithms (Optimized) [27]	19	14	73.7	1

Table 4.2 Set-2 Detection Results (Proposed Algorithm)

CLUSTER		TOTAL	DETECTED	% of DETECTION	FALSE DETECTION
	TRIANGULAR	45	22	48.9	0
CIRCULAR	63	9	14.3	0	
DEFORMED		TOTAL	DETECTED	% of DETECTION	FALSE DETECTION
	TRIANGULAR	16	12	75.0	0
CIRCULAR	8	6	75.0	1	
MIXED		TOTAL	DETECTED	% of DETECTION	FALSE DETECTION
	TRIANGULAR	29	19	65.5	0
CIRCULAR	21	13	61.9	0	
DIFFERENT SHAPES		TOTAL	DETECTED	% of DETECTION	FALSE DETECTION
	TRIANGULAR	41	33	80.5	0
CIRCULAR	19	9	47.4	0	
ILLUMINATION		TOTAL	DETECTED	% of DETECTION	FALSE DETECTION
	TRIANGULAR	31	14	45.2	0
CIRCULAR	29	11	37.9	1	
OCCLUSION		TOTAL	DETECTED	% of DETECTION	FALSE DETECTION
	TRIANGULAR	33	5	15.2	0
CIRCULAR	33	16	48.5	2	
ROTATION		TOTAL	DETECTED	% of DETECTION	FALSE DETECTION
	TRIANGULAR	12	3	25.0	0
CIRCULAR	9	4	44.4	1	
SHADOW		TOTAL	DETECTED	% of DETECTION	FALSE DETECTION
	TRIANGULAR	37	10	27.0	0
CIRCULAR	27	16	59.3	0	
SIZED		TOTAL	DETECTED	% of DETECTION	FALSE DETECTION
	TRIANGULAR	51	15	29.4	0
CIRCULAR	36	6	16.7	1	
TRANSLATION		TOTAL	DETECTED	% of DETECTION	FALSE DETECTION
	TRIANGULAR	21	9	42.9	0
CIRCULAR	34	14	41.2	0	
TOTAL		TOTAL	DETECTED	% of DETECTION	FALSE DETECTION
	TRIANGULAR	316	142	44.9	0
	CIRCULAR	279	104	37.3	6

Table 4.3 Set-2 Detection Results ( Proposed Algorithm with cvHoughLines2() & cvHoughCircles() )

CLUSTER		TOTAL	DETECTED	% of DETECTION	FALSE DETECTION	
		TRIANGULAR	45	21	46.7	3
	CIRCULAR	63	19	30.2	0	
DEFORMED		TOTAL	DETECTED	% of DETECTION	FALSE DETECTION	
		TRIANGULAR	16	12	75.0	2
	CIRCULAR	8	1	12.5	1	
MIXED		TOTAL	DETECTED	% of DETECTION	FALSE DETECTION	
		TRIANGULAR	29	18	62.1	7
	CIRCULAR	21	6	28.6	0	
DIFFERENT SHAPES		TOTAL	DETECTED	% of DETECTION	FALSE DETECTION	
		TRIANGULAR	41	33	80.5	7
	CIRCULAR	19	2	10.5	0	
ILLUMINATION		TOTAL	DETECTED	% of DETECTION	FALSE DETECTION	
		TRIANGULAR	31	14	45.2	4
	CIRCULAR	29	2	6.9	0	
OCCLUSION		TOTAL	DETECTED	% of DETECTION	FALSE DETECTION	
		TRIANGULAR	33	5	15.2	3
	CIRCULAR	33	3	9.1	0	
ROTATION		TOTAL	DETECTED	% of DETECTION	FALSE DETECTION	
		TRIANGULAR	12	3	25.0	2
	CIRCULAR	9	0	0.0	1	
SHADOW		TOTAL	DETECTED	% of DETECTION	FALSE DETECTION	
		TRIANGULAR	37	10	27.0	8
	CIRCULAR	27	6	22.2	0	
SIZED		TOTAL	DETECTED	% of DETECTION	FALSE DETECTION	
		TRIANGULAR	51	19	37.3	2
	CIRCULAR	36	5	13.9	0	
TRANSLATION		TOTAL	DETECTED	% of DETECTION	FALSE DETECTION	
		TRIANGULAR	21	9	42.9	5
	CIRCULAR	34	1	2.9	0	
TOTAL		TOTAL	DETECTED	% of DETECTION	FALSE DETECTION	
		TRIANGULAR	316	144	45.6	43
		CIRCULAR	279	45	16.1	2

Table 4.4 Set-2 Detection results of [27]

	Algorithm	Sign Type	Signs to be Detected	Detected Traffic Signs	% of Detection	False Positive
	DEFORMED	Color1, RGB	RED	24	13	54,17
Color1, RGB (optimized)		RED	24	21	87,50	9
Color1,HSV		RED	24	14	58,33	2
Color1,HSV (optimized)		RED	24	22	91,67	2
Color2		RED	24	15	62,50	2
Color2 (optimized)		RED	24	21	87,50	2
Shape		TRIANGULAR	15	7	46,67	0
		CIRCULAR	17	4	23,53	5
Shape (optimized)	TRIANGULAR	15	14	93,33	0	
	CIRCULAR	17	15	88,24	5	
ILLUMINATION	Algorithm	Sign Type	Signs to be Detected	Detected Traffic Signs	% of Detection	False Positive
	Color1, RGB	RED	50	23	46,00	16
	Color1, RGB (optimized)	RED	50	29	58,00	16
	Color1,HSV	RED	50	26	52,00	1
	Color1,HSV (optimized)	RED	50	30	60,00	1
	Color2	RED	50	27	54,00	6
	Color2 (optimized)	RED	50	38	76,00	6
	Shape	TRIANGULAR	23	5	21,74	2
		CIRCULAR	29	10	34,48	8
	Shape (optimized)	TRIANGULAR	23	22	95,65	2
CIRCULAR		29	26	89,66	8	
OCCLUSION	Algorithm	Sign Type	Signs to be Detected	Detected Traffic Signs	% of Detection	False Positive
	Color1, RGB	RED	62	43	69,35	16
	Color1, RGB (optimized)	RED	62	50	80,65	16
	Color1,HSV	RED	62	38	61,29	6
	Color1,HSV (optimized)	RED	62	46	74,19	6
	Color2	RED	62	37	59,68	6
	Color2 (optimized)	RED	62	45	72,58	6
	Shape	TRIANGULAR	28	1	3,57	1
		CIRCULAR	38	16	42,11	3
	Shape (optimized)	TRIANGULAR	28	24	85,71	1
CIRCULAR		38	35	92,11	3	
ROTATION	Algorithm	Sign Type	Signs to be Detected	Detected Traffic Signs	% of Detection	False Positive
	Color1, RGB	RED	30	21	70,00	25
	Color1, RGB (optimized)	RED	30	26	86,67	25
	Color1,HSV	RED	30	18	60,00	2
	Color1,HSV (optimized)	RED	30	22	73,33	2
	Color2	RED	30	16	53,33	2
	Color2 (optimized)	RED	30	20	66,67	2
	Shape	TRIANGULAR	11	2	18,18	1
		CIRCULAR	14	4	28,57	5
	Shape (optimized)	TRIANGULAR	11	9	81,82	1
CIRCULAR		14	11	78,57	5	
SHADOW	Algorithm	Sign Type	Signs to be Detected	Detected Traffic Signs	% of Detection	False Positive
	Color1, RGB	RED	64	39	60,94	23
	Color1, RGB (optimized)	RED	64	50	78,13	23
	Color1,HSV	RED	64	45	70,31	1
	Color1,HSV (optimized)	RED	64	61	95,31	1
	Color2	RED	64	39	60,94	4
	Color2 (optimized)	RED	64	55	85,94	4
	Shape	TRIANGULAR	36	1	2,78	1
		CIRCULAR	29	18	62,07	3
	Shape (optimized)	TRIANGULAR	36	31	86,11	1
CIRCULAR		29	28	96,55	3	
TRANSLATION	Algorithm	Sign Type	Signs to be Detected	Detected Traffic Signs	% of Detection	False Positive
	Color1, RGB	RED	48	29	60,42	33
	Color1, RGB (optimized)	RED	48	38	79,17	33
	Color1,HSV	RED	48	33	68,75	4
	Color1,HSV (optimized)	RED	48	43	89,58	4
	Color2	RED	48	31	64,58	5
	Color2 (optimized)	RED	48	41	85,42	5
	Shape	TRIANGULAR	16	1	6,25	2
		CIRCULAR	33	6	18,18	5
	Shape (optimized)	TRIANGULAR	16	15	93,75	2
CIRCULAR		33	31	93,94	5	

Set-1 includes well-illuminated samples of traffic signs. Therefore, the algorithm gives successful results with 100%. However, the success of the algorithm shall be measured on several images including difficulties on detection. For this purpose, Set-2 is used.

The detection algorithm decides a connected component as a triangular sign if it has three edges, as described in 3.2.2. Otherwise, the circularity is checked. Therefore, for a merged sign of a triangular and a circular sign, the priority of the detection of triangular signs is higher. A sample example is shown on Figure 4.2. On the other hand, for a merged sign of two or more circular signs, local maximum regions on the line histogram indicate several line segments instead of arcs. A related misdetection sample is given on Figure 4.3.



Figure 4.2 Sample misdetection of a circular sign for a merged sign

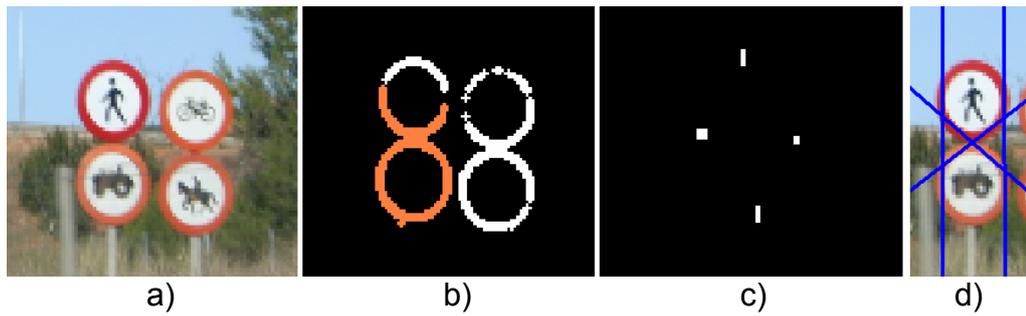


Figure 4.3 a) Sample merged signs, b) one of the connected components (with different color) to be analyzed, c) line histogram result, indicating 4 line segments, d) resulting line segments.

The color segmentation at detection stage includes a threshold on the saturation value of the image, which eliminates red regions with low pureness of color. Therefore, this thresholding may cause misdetections of traffic signs. However, omitting it may result higher false detection rate on the background region. As a result, the detection rates on illumination and shadow group are lower, while keeping the false detection rate low. Adjusting the thresholds for the specific group of images, increases the detection rates as shown on the results of [27] as optimized versions of methods, shown on Table 4.4.

A triangular sign can be detected, only if all of its three edges are obtained with one connected component. This means, if one corner region is occluded, the triangle can be recovered as shown on Figure 4.4 – a. However, if one edge is totally occluded or separated from the rest of the connected component, the triangular shape cannot be recovered as shown on Figure 4.4 – c. On the other hand, one arc can

give the whole information of the circle it belongs, for the circular signs as shown on Figure 4.4 – b. Therefore, the result on occlusion group for triangular signs is worse than the one for circular signs.

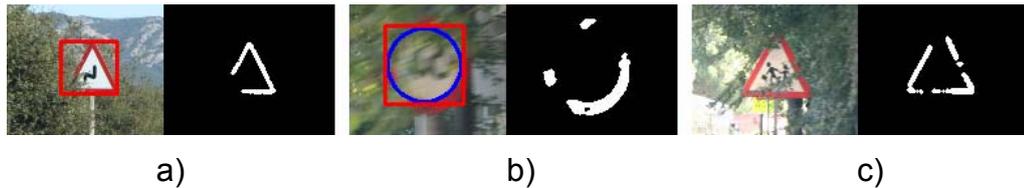


Figure 4.4 a) Recovered triangular sign, b) recovered circular sign, c) unrecovered triangular sign.

If the arc length of a circular sign is very short or the ends of the arc give wrong information about the rest of the circle, the recovered circle may not totally match with the rest of the sign. A sample result of reconstruction of a circular shape is shown on Figure 4.5.

In order to eliminate noise on the color segmentation part, morphological opening is used after HSV thresholding. However, if the obtained frame of a sign is very thin, the opening operation will yield broken or erased edges as shown on Figure 4.6 – c. Therefore, the sized group has lower detection rate.

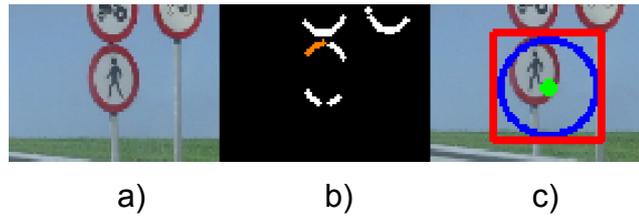


Figure 4.5 a) Sample circular sign, b) detected arc (different colored), c) recovered circle.

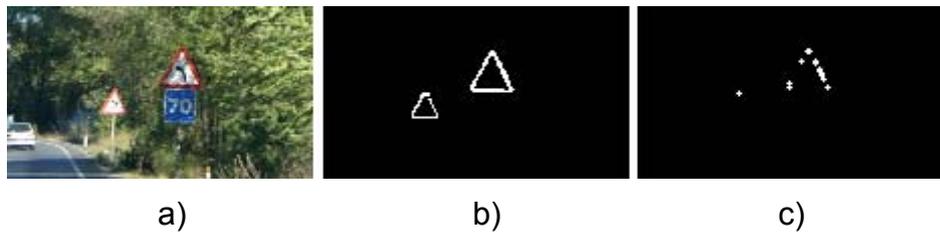


Figure 4.6 a) Input image, b) result of color segmentation, c) result after morphological opening.

Since the success of Circular Hough Transform does not depend on the number of circles on the input, combination of connected components gives the same detection results with the separated ones. However, for Line Histogram algorithm, if the input component is defined by a combination of several shapes, the extraction of shape information becomes more difficult. Therefore, for circular signs in Cluster group, the success rate of Circular Hough Transform is higher, as shown on Table 4.3. On the other hand, for other groups, the success rates of detection of circular signs are very low. This is the result of properties of `cvHoughCircles()` function. Since this function, first passes the input through a Canny Edge Detection algorithm, if the observed sign has a

defect on its edge or it is observed as an ellipse, the function cannot detect that circular sign. The success rate can be increased by trimming the parameters of the function. However, this may yield more false detection rate.

Optimizing the threshold values may yield more hit on detection. Moreover, different selections of morphological operator for triangular and circular signs respectively may result higher detection rate. Used detection stage starts with color segmentation and then the shape information of found segments are analyzed. Therefore, if the color information cannot be extracted from a scene, the signs are omitted even if their shapes are detectable. As a result, if a shape analysis is added to color analysis as a parallel sense, the detection rate may be increased.

## **4.2 CLASSIFICATION**

The classification results for Set-1 are shown on Table 4.6. For Set-2, the results are on Table 4.6 to 4.11. In detection stage, Yield sign <sup>1</sup> can be identified, because of its unique shape. Due to this, the classification percentages are calculated by excluding yield signs. “Classified” column, on the tables, shows the number of successful classification (including Yield signs). “False Positive” column shows the number of confusions on classification. “False Negative” column shows the number of candidates those cannot be classified.

---

<sup>1</sup> Figure A.3 “Yol Ver” Sign

Table 4.5 Set-1 Classification Results

Template Matching	CANDIDATES	CLASSIFIED	% of CLASSIFICATION	FALSE POSITIVE	FALSE NEGATIVE
DT	18	4	11.11	15	1
Template Matching	CANDIDATES	CLASSIFIED	% of CLASSIFICATION	FALSE POSITIVE	FALSE NEGATIVE
RGB	18	7	27.78	13	0
Neural Networks	CANDIDATES	CLASSIFIED	% of CLASSIFICATION	FALSE POSITIVE	FALSE NEGATIVE
DT	18	14	66.67	4	2
Neural Networks	CANDIDATES	CLASSIFIED	% of CLASSIFICATION	FALSE POSITIVE	FALSE NEGATIVE
Red	18	14	66.67	3	3
Neural Networks	CANDIDATES	CLASSIFIED	% of CLASSIFICATION	FALSE POSITIVE	FALSE NEGATIVE
RGB	18	3	5.56	7	10
Neural Networks	CANDIDATES	CLASSIFIED	% of CLASSIFICATION	FALSE POSITIVE	FALSE NEGATIVE
Gray	18	5	16.67	13	2

Table 4.6 Set-2 Classification Results (Template Matching: Distance Transform)

CLUSTER	CANDIDATES	CLASSIFIED	% of CLASSIFICATION	FALSE POSITIVE	FALSE NEGATIVE
	19	16	84,21	3	0
DEFORMED	CANDIDATES	CLASSIFIED	% of CLASSIFICATION	FALSE POSITIVE	FALSE NEGATIVE
	2	1	50,00	1	0
MIXED	CANDIDATES	CLASSIFIED	% of CLASSIFICATION	FALSE POSITIVE	FALSE NEGATIVE
	18	10	55,56	8	0
DIFFERENT SHAPES	CANDIDATES	CLASSIFIED	% of CLASSIFICATION	FALSE POSITIVE	FALSE NEGATIVE
	25	14	56,00	11	0
ILLUMINATION	CANDIDATES	CLASSIFIED	% of CLASSIFICATION	FALSE POSITIVE	FALSE NEGATIVE
	13	8	61,54	5	0
OCCCLUSION	CANDIDATES	CLASSIFIED	% of CLASSIFICATION	FALSE POSITIVE	FALSE NEGATIVE
	5	2	40,00	3	0
ROTATION	CANDIDATES	CLASSIFIED	% of CLASSIFICATION	FALSE POSITIVE	FALSE NEGATIVE
	3	0	0,00	3	0
SHADOW	CANDIDATES	CLASSIFIED	% of CLASSIFICATION	FALSE POSITIVE	FALSE NEGATIVE
	10	4	40,00	3	3
SIZED	CANDIDATES	CLASSIFIED	% of CLASSIFICATION	FALSE POSITIVE	FALSE NEGATIVE
	12	4	33,33	8	0
TRANSLATION	CANDIDATES	CLASSIFIED	% of CLASSIFICATION	FALSE POSITIVE	FALSE NEGATIVE
	6	0	0,00	6	0
TOTAL	113	59	52,21	51	3

Table 4.7 Set-2 Classification Results (Template Matching: RGB Image)

CLUSTER	CANDIDATES	CLASSIFIED	% of CLASSIFICATION	FALSE POSITIVE	FALSE NEGATIVE
	19	15	78,95	4	0
DEFORMED	CANDIDATES	CLASSIFIED	% of CLASSIFICATION	FALSE POSITIVE	FALSE NEGATIVE
	2	1	50,00	1	0
MIXED	CANDIDATES	CLASSIFIED	% of CLASSIFICATION	FALSE POSITIVE	FALSE NEGATIVE
	18	10	55,56	8	0
DIFFERENT SHAPES	CANDIDATES	CLASSIFIED	% of CLASSIFICATION	FALSE POSITIVE	FALSE NEGATIVE
	25	16	64,00	9	2
ILLUMINATION	CANDIDATES	CLASSIFIED	% of CLASSIFICATION	FALSE POSITIVE	FALSE NEGATIVE
	13	8	61,54	4	1
OCCCLUSION	CANDIDATES	CLASSIFIED	% of CLASSIFICATION	FALSE POSITIVE	FALSE NEGATIVE
	5	3	60,00	1	1
ROTATION	CANDIDATES	CLASSIFIED	% of CLASSIFICATION	FALSE POSITIVE	FALSE NEGATIVE
	3	1	33,33	2	0
SHADOW	CANDIDATES	CLASSIFIED	% of CLASSIFICATION	FALSE POSITIVE	FALSE NEGATIVE
	10	2	20,00	8	0
SIZED	CANDIDATES	CLASSIFIED	% of CLASSIFICATION	FALSE POSITIVE	FALSE NEGATIVE
	12	7	58,33	5	0
TRANSLATION	CANDIDATES	CLASSIFIED	% of CLASSIFICATION	FALSE POSITIVE	FALSE NEGATIVE
	6	4	66,67	2	0
<b>TOTAL</b>	113	67	59,29	44	4

Table 4.8 Set-2 Classification Results (Neural Networks: Distance Transform)

CLUSTER	CANDIDATES	CLASSIFIED	% of CLASSIFICATION	FALSE POSITIVE	FALSE NEGATIVE
	19	16	84,21	2	1
DEFORMED	CANDIDATES	CLASSIFIED	% of CLASSIFICATION	FALSE POSITIVE	FALSE NEGATIVE
	2	1	50,00	1	0
MIXED	CANDIDATES	CLASSIFIED	% of CLASSIFICATION	FALSE POSITIVE	FALSE NEGATIVE
	18	8	44,44	5	5
DIFFERENT SHAPES	CANDIDATES	CLASSIFIED	% of CLASSIFICATION	FALSE POSITIVE	FALSE NEGATIVE
	25	14	56,00	5	6
ILLUMINATION	CANDIDATES	CLASSIFIED	% of CLASSIFICATION	FALSE POSITIVE	FALSE NEGATIVE
	13	9	69,23	3	1
OCCCLUSION	CANDIDATES	CLASSIFIED	% of CLASSIFICATION	FALSE POSITIVE	FALSE NEGATIVE
	5	3	60,00	1	1
ROTATION	CANDIDATES	CLASSIFIED	% of CLASSIFICATION	FALSE POSITIVE	FALSE NEGATIVE
	3	1	33,33	1	1
SHADOW	CANDIDATES	CLASSIFIED	% of CLASSIFICATION	FALSE POSITIVE	FALSE NEGATIVE
	10	3	30,00	6	1
SIZED	CANDIDATES	CLASSIFIED	% of CLASSIFICATION	FALSE POSITIVE	FALSE NEGATIVE
	12	5	41,67	6	1
TRANSLATION	CANDIDATES	CLASSIFIED	% of CLASSIFICATION	FALSE POSITIVE	FALSE NEGATIVE
	6	2	33,33	2	2
<b>TOTAL</b>	113	62	54,87	32	19

Table 4.9 Set-2 Classification Results (Neural Networks: Red component)

CLUSTER	CANDIDATES	CLASSIFIED	% of CLASSIFICATION	FALSE POSITIVE	FALSE NEGATIVE
	19	15	78,95	1	3
DEFORMED	CANDIDATES	CLASSIFIED	% of CLASSIFICATION	FALSE POSITIVE	FALSE NEGATIVE
	2	1	50,00	0	1
MIXED	CANDIDATES	CLASSIFIED	% of CLASSIFICATION	FALSE POSITIVE	FALSE NEGATIVE
	18	7	38,89	7	4
DIFFERENT SHAPES	CANDIDATES	CLASSIFIED	% of CLASSIFICATION	FALSE POSITIVE	FALSE NEGATIVE
	25	12	48,00	9	4
ILLUMINATION	CANDIDATES	CLASSIFIED	% of CLASSIFICATION	FALSE POSITIVE	FALSE NEGATIVE
	13	10	76,92	2	1
OCCCLUSION	CANDIDATES	CLASSIFIED	% of CLASSIFICATION	FALSE POSITIVE	FALSE NEGATIVE
	5	2	40,00	3	0
ROTATION	CANDIDATES	CLASSIFIED	% of CLASSIFICATION	FALSE POSITIVE	FALSE NEGATIVE
	3	0	0,00	1	2
SHADOW	CANDIDATES	CLASSIFIED	% of CLASSIFICATION	FALSE POSITIVE	FALSE NEGATIVE
	10	2	20,00	6	2
SIZED	CANDIDATES	CLASSIFIED	% of CLASSIFICATION	FALSE POSITIVE	FALSE NEGATIVE
	12	5	41,67	6	1
TRANSLATION	CANDIDATES	CLASSIFIED	% of CLASSIFICATION	FALSE POSITIVE	FALSE NEGATIVE
	6	1	16,67	3	2
<b>TOTAL</b>	<b>113</b>	<b>55</b>	<b>48,67</b>	<b>38</b>	<b>20</b>

Table 4.10 Set-2 Classification Results (Neural Networks: Gray Scaled Image)

CLUSTER	CANDIDATES	CLASSIFIED	% of CLASSIFICATION	FALSE POSITIVE	FALSE NEGATIVE
	19	10	52,63	6	3
DEFORMED	CANDIDATES	CLASSIFIED	% of CLASSIFICATION	FALSE POSITIVE	FALSE NEGATIVE
	2	1	50,00	1	0
MIXED	CANDIDATES	CLASSIFIED	% of CLASSIFICATION	FALSE POSITIVE	FALSE NEGATIVE
	18	8	44,44	7	3
DIFFERENT SHAPES	CANDIDATES	CLASSIFIED	% of CLASSIFICATION	FALSE POSITIVE	FALSE NEGATIVE
	25	10	40,00	9	6
ILLUMINATION	CANDIDATES	CLASSIFIED	% of CLASSIFICATION	FALSE POSITIVE	FALSE NEGATIVE
	13	2	15,38	11	0
OCCCLUSION	CANDIDATES	CLASSIFIED	% of CLASSIFICATION	FALSE POSITIVE	FALSE NEGATIVE
	5	2	40,00	3	0
ROTATION	CANDIDATES	CLASSIFIED	% of CLASSIFICATION	FALSE POSITIVE	FALSE NEGATIVE
	3	0	0,00	1	2
SHADOW	CANDIDATES	CLASSIFIED	% of CLASSIFICATION	FALSE POSITIVE	FALSE NEGATIVE
	10	0	0,00	8	2
SIZED	CANDIDATES	CLASSIFIED	% of CLASSIFICATION	FALSE POSITIVE	FALSE NEGATIVE
	12	5	41,67	5	2
TRANSLATION	CANDIDATES	CLASSIFIED	% of CLASSIFICATION	FALSE POSITIVE	FALSE NEGATIVE
	6	0	0,00	6	0
<b>TOTAL</b>	<b>113</b>	<b>38</b>	<b>33,63</b>	<b>57</b>	<b>18</b>

Table 4.11 Set-2 Classification Results (Neural Networks: RGB Image)

CLUSTER	CANDIDATES	CLASSIFIED	% of CLASSIFICATION	FALSE POSITIVE	FALSE NEGATIVE
	19	15	78,95	3	1
DEFORMED	CANDIDATES	CLASSIFIED	% of CLASSIFICATION	FALSE POSITIVE	FALSE NEGATIVE
	2	0	0,00	1	1
MIXED	CANDIDATES	CLASSIFIED	% of CLASSIFICATION	FALSE POSITIVE	FALSE NEGATIVE
	18	4	22,22	13	1
DIFFERENT SHAPES	CANDIDATES	CLASSIFIED	% of CLASSIFICATION	FALSE POSITIVE	FALSE NEGATIVE
	25	11	44,00	7	7
ILLUMINATION	CANDIDATES	CLASSIFIED	% of CLASSIFICATION	FALSE POSITIVE	FALSE NEGATIVE
	13	3	23,08	8	2
OCCCLUSION	CANDIDATES	CLASSIFIED	% of CLASSIFICATION	FALSE POSITIVE	FALSE NEGATIVE
	5	2	40,00	2	1
ROTATION	CANDIDATES	CLASSIFIED	% of CLASSIFICATION	FALSE POSITIVE	FALSE NEGATIVE
	3	0	0,00	2	1
SHADOW	CANDIDATES	CLASSIFIED	% of CLASSIFICATION	FALSE POSITIVE	FALSE NEGATIVE
	10	2	20,00	6	2
SIZED	CANDIDATES	CLASSIFIED	% of CLASSIFICATION	FALSE POSITIVE	FALSE NEGATIVE
	12	5	41,67	5	2
TRANSLATION	CANDIDATES	CLASSIFIED	% of CLASSIFICATION	FALSE POSITIVE	FALSE NEGATIVE
	6	4	66,67	2	0
<b>TOTAL</b>	<b>113</b>	<b>46</b>	<b>40,71</b>	<b>49</b>	<b>18</b>

For Set-1, since all of the images include traffic signs having the same pictogram group, shown on Figure 4.7, the identification results depend mostly on the performance of the identification method on that specific traffic sign images. Therefore, it is reasonable to evaluate identification stage by the results of Set-2. The rest of the comments on identification results are done under consideration of the results on Set-2.



Figure 4.7 Two samples of Set-1

For Set-2 images, common classification error occurs if a traffic sign has some pictogram differences according to the used template of that sign. These common classification errors are shown on Figure 4.8. Moreover, because some traffic signs are not defined in Turkey, the database does not include them. Therefore, most of the times, the results of the classification for those candidates, are the closest database image. Sample candidate signs, which have no equivalent template in the database, and their classification results are shown on Figure 4.9.



Figure 4.8 a) Sample input candidates from Set-2, b) equivalent signs in the used database, c) common misclassification results.



Figure 4.9 a) Sample candidate without a response on the database, b) wrong classification result

Results of the template matching algorithms and the neural networks approaches are nearly the same. However, template matching is a more time consuming process, because each time the candidate shall be compared by each template, including its shifted versions. Additionally, in this thesis, template-matching approaches cannot handle the rotational errors on the candidate image. On the other hand, training neural networks with lots of deformed versions of training set is possible. Moreover, the processing time of the neural network does not directly depend on the number of the members of training set.

In addition to the approach-related differences, some other classification errors, related to the selected feature set, are observed. For example, using gray values as the feature set makes the classifier more sensitive to illumination differences (such as shadows) on the candidates. Since the extraction of red feature set includes a thresholding stage, the influence of illumination on classification is decreased. However, this time errors on the actual pictogram boundaries, after thresholding, become the major factor on misclassification. Therefore, using the distance transformation of the binarized image, distributes the pictogram information (including its position) over the empty regions on the sign. As a result, the influence of illumination becomes negligible.

## **CHAPTER 5**

### **CONCLUSIONS AND FUTURE WORK**

#### **5.1 CONCLUSION**

The difficulty of the Traffic Sign Recognition concept is the most common difficulty for other pattern recognition problems. As experienced, one more time with this thesis work, the segmentation of required data within a captured frame of a scene is the most difficult step. This difficulty is the result of image processing methodology, which handles the data pixel by pixel. While, a human being is observing the visual data by his eyes, the image is rendered by experiences, learned facts, laws of physics, known words or others. Therefore, all objects on the scene are separated and many extra features are added according to the previous experiences. With such a segmentation stage, it would be very easy to identify a required object. Since, the data for this problem consists of RGB pixel values without a smart renderer as a human brain, the segmentation step was implemented by color and shape analysis. In order to be adaptive to bad visual factors, based on some weather conditions or other circumstances, the algorithm was developed under the assumption that

the captured frame includes occluded traffic signs. By this assumption, “Line Histogram” method was designed. That method provides the ability of identifying the shape information of connected components, those are parts of traffic signs. Since, each segment of the connected component can be separately detected by this method; recovering the shape of the traffic signs becomes easier. Hence, the influence of visual disadvantages, are tried to be minimized. At the final stage, two different approaches, feed-forward neural networks and template matching, are used to identify the candidates.

Line histogram method seems to be applicable to detect geometrical shapes on binary images. Although the detection of line segments is shown to be successful, the analysis for circular shapes shall be improved. As classification approaches, both template matching and neural networks give close results. However, neural networks can cover more range of errors on used training set with less processing time. On the other hand, each additional template of traffic signs increases the computation time for template matching approach.

Besides the classification approaches, the result is surely dependent on the selected feature set. For neural networks, distance transformation leads the rest of feature sets, because of its ability of distributing the informative part (pictogram) over the area of the sign. This increases the comparison area of the sign. On the other hand, for template matching method, normalized cross correlation provides stable results for illumination changes. Therefore, RGB feature set also results high detection rates, for template matching approach.

## 5.2 FUTURE WORKS

The detection algorithm can be improved by an additional stage that can separate the clustered traffic signs, in order not to lose one or all of those signs. Moreover, instead of working on only one connected component, considering other neighboring components can improve the result on occluded signs.

For classification stage, the training set of neural networks may be improved by additional members, defining the same traffic sign with different pictograms. Besides the rotated, shifted and noisy samples on training set, additional scaled versions of sample signs can also improve the results of neural networks.

## REFERENCES

- [1] "Trafik İşaretleri", <http://www.olcaysurucukursu.com/>,  
last accessed on 31.01.2009 at 21:37
- [2] "Color Principles - Hue, Saturation, and Value" NC State University  
College of Education - Graphic Communications Program,  
[http://www.ncsu.edu/scivis/lessons/colormodels/color\\_models2.html](http://www.ncsu.edu/scivis/lessons/colormodels/color_models2.html)  
last accessed on 01.02.2009 at 00:25
- [3] Uğur HALICI, "Artificial Neural Networks – Lecture Notes", Middle  
East Technical University, Electrical and Electronics Engineering Dept.,  
<http://vision1.eee.metu.edu.tr/~halici/courses/543LectureNotes/543index.html>  
last accessed on 02.02.2009 at 01:00
- [4] J. Miura, T. Kanda, Y. Shirai, "An Active Vision System for Real-  
Time Traffic Sign Recognition", 2000 IEEE Intelligent Transportation  
Systems Conference Proceedings, Dearborn (MI), USA, October 1-3,  
2000.
- [5] A. de la Escalera, J.M Armingol, and M. Mata, "Traffic sign  
recognition and analysis for intelligent vehicles", Image and Vision  
Computing, vol.21, pp.247–258, 2003.

[6] G. Piccioli, E. De Micheli, P. Parodi, and M. Campani, "Robust method for road sign detection and recognition", *Image and Vision Computing*, vol.14(3) , pp.209–223, April 1996.

[7] T. Pavlidis, "Algorithms for Graphics and Image Processing", Computer Science Press, Rockville, MD, 1982.

[8] A. de la Escalera, L.E. Moreno, M.A. Salichs and J.M. Armingol, "Road Traffic Sign Detection and Classification", *Industrial Electronics, IEEE Transactions on* , vol.44, no.6, pp.848-859, Dec 1997.

[9] J. Hatzidimos, "Automatic Traffic Sign Recognition In Digital Images", *Proceedings of the International Conference on Theory and Applications of Mathematics and Informatics - ICTAMI 2004*, Thessaloniki, Greece, 2004.

[10] R. Duda. "Use Of The Hough Transformation To Detect Lines And Curves In Pictures", Peter E. Hart Artificial Intelligence Center, January 1972.

[11] C. F. Carlson, "Lecture 10: Hough Circle Transform", Rochester Institute of Technology: Lecture Notes, October 11, 2005.

[12] S. J. K. Pedersen, "Circular Hough Transform", Aalborg University, Vision, Graphics, and Interactive Systems, November 2007, [http://www.cvmt.dk/education/teaching/e07/MED3/IP/Simon\\_Pedersen\\_CircularHoughTransform.pdf](http://www.cvmt.dk/education/teaching/e07/MED3/IP/Simon_Pedersen_CircularHoughTransform.pdf),

last accessed on 04.02.2009 at 17:34

[13] D. H. Ballard, "Generalizing the Hough Transform To Detect Arbitrary Shapes", Computer Science Department, University of Rochester, Rochester, NY 14627, USA, 23 September 1980.

[14] X. Baro, S. Escalera, J. Vitria, O. Pujol, P. Radeva, "Traffic Sign Recognition Using Evolutionary Adaboost Detection and Forest-ECOC Classification," Intelligent Transportation Systems, IEEE Transactions on , vol.10, no.1, pp.113-126, March 2009

[15] Nguwi Yok-Yen, A.Z. Kouzani, "Automatic Road Sign Recognition Using Neural Networks," Neural Networks, 2006. IJCNN '06. International Joint Conference, pp.3955-3962,

[16] A. Broggi, P. Cerri, P. Medici, P.P. Porta, "Real Time Road Signs Recognition", Proceedings of the 2007 IEEE Intelligent Vehicles Symposium Istanbul, Turkey, June 13-15, 2007.

[17] H. Fleyeh, "Traffic Sign Recognition By Fuzzy Sets" Intelligent Vehicles Symposium, 2008 IEEE , pp.422-427, 4-6 June 2008.

[18] S. Vitabile, G. Pollaccia, G. Pilato, E. Sorbello, "Road Signs Recognition Using A Dynamic Pixel Aggregation Technique In The Hsv Color Space" Image Analysis and Processing, 2001. Proceedings. 11th International Conference on , pp.572-577, 26-28 Sep 2001.

[19] Sin-Yu Chen, Jun-Wei Hsieh, "Boosted Road Sign Detection And Recognition" Machine Learning and Cybernetics, 2008 International Conference on , vol.7, pp.3823-3826, 12-15 July 2008.

[20] R. Lienhart, J. Maydt, "An Extended Set Of Haar-Like Features For Rapid Object Detection" Image Processing. 2002. Proceedings. 2002 International Conference on , vol.1, pp. I-900-I-903 vol.1, 2002.

[21] Yok-Yen Nguwi; Siu-Yeung Cho, "Two-Tier Self-Organizing Visual Model For Road Sign Recognition" Neural Networks, 2008. IJCNN 2008. (IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on , pp.794-799, 1-8 June 2008.

[22] M. Shi, H. Wu, H. Fleyeh, "A Robust Model for Traffic Signs Recognition Based on Support Vector Machines," Image and Signal Processing, 2008. CISP '08. Congress on , vol.4, pp.516-524, 27-30 May 2008.

[23] A. A. ALATAN, "Pattern Recognition – Lecture Notes", Middle East Technical University, Electrical and Electronics Engineering Dept., <http://www.eee.metu.edu.tr/~alatan/>  
last accessed on 08.04.2009 at 17:40

[24] Alcalà de Henares (Madrid, Spain) Universidad de Alcalà, <http://roadanalysis.uah.es>  
last accessed on 24.04.2009 at 12:20

[25] "Tehlike Uyarı İşaretleri", <http://www.sinyalsurucukursu.com/?p=25>  
last accessed on 24.04.2009 at 14:35

[26] Jianbo Shi, J. Malik, "Normalized Cuts And Image Segmentation" Pattern Analysis and Machine Intelligence, IEEE Transactions on , vol.22, no.8, pp.888-905, Aug 2000.

[27] Emre Ulay, "Color and Shape Based Traffic Sign Detection", Thesis submitted to the Graduate School of Natural and Applied Sciences of Middle East Technical University, November 2008.

## APPENDIX A

### LIST OF SPECIFIED TRAFFIC SIGNS USED FOR THIS THESIS

The training sets and templates are prepared by the images from [25]. The following list - Figure A.1, Figure A.2 and Figure A.3 - just shows the considered types and members of traffic signs.



Figure A.1 Stop and Parking Traffic Signs [1]

			
Sağa Tehlikeli Viraj	Sola Tehlikeli Viraj	Sağa Tehlikeli Devamlı Virajlar	Sola Tehlikeli Devamlı Virajlar
			
Yaya Geçidi	Okul Geçidi	Bisiklet Geçebilir	Ana Yol - Tali Yol Kavşağı
			
Ana Yol - Tali Yol Kavşağı	Ana Yol - Tali Yol Kavşağı	Sağda Ana Yola Giriş	Soldan Ana Yola Giriş
			
Deniz veya Nehir Kıyısında Biten Yol	Kasisli Yol	Kasisli Köprü Yaklaşımı	Gevşek Şev
			
Kaygan Yol	Gevşek Malzemeli Zemin	İki Yönlü Trafik	Dikkat
			
KontROLSÜZ Kavşak	Dönel Kavşak Yaklaşımı	Tehlikeli Eğim (İniş)	Tehlikeli Eğim (Çıkış)
			
İki Taraftan Daralan Kaplama	Sağdan Daralan Kaplama	Soldan Daralan Kaplama	Açılan Köprü
			
Yolda Çalışma Var	Işıklı İşaret Cihazı	Hava Alanı (Alçak Uçuş)	Yandan Rüzgar
			
Ehli Hayvanlar Geçebilir	Vahşi Hayvanlar Geçebilir	Kontrollü Demiryolu Geçidi	KontROLSÜZ Demiryolu Geçidi

Figure A.2 Danger Warning Traffic Signs [1]

Yol Ver	Azami Hız Sınırlaması	Taşıt Trafiğine Kapalı Yol	Karşıdan Gelene Yol Ver
Taşıt Giremez	Motorlu Taşıt Giremez	Otobüs Giremez	Treyler Giremez
Yaya Giremez	At Arabası Giremez	El Arabası Giremez	Tarım Traktörü Giremez
Belirli Miktardan Fazla Patlayıcı ve Parlayıcı Madde Taşıyan Taşıt Giremez	Belirli Miktardan Fazla Su Kirlenici Madde Taşıyan Taşıt Giremez	Motosiklet Hariç Motorlu Taşıt Trafiğine Kapalı Yol	Motosiklet Giremez
Bisiklet Giremez	Mopet Giremez	Kamyon Giremez	Genişliği ... Metreden Fazla Olan Taşıt Giremez
Yüksekliği ... Metreden Fazla Olan Taşıt Giremez	Uzunluğu ... Metreden Fazla Olan Araç Giremez	Dingil Başına ... Tondan Fazla Yük Düşen Taşıt Giremez	Yüklü Ağırlığı ... Tondan Fazla Olan Taşıt Giremez
Öndeki Taşıt ... Metreden Daha Yakın Takip Edilmez	Sağa Dönülmez	Sola Dönülmez	U Dönüşü Yapılmaz
Öndeki Taşıtı Geçmek Yasaktır	Kamyonlar İçin Öndeki Taşıtı Geçmek Yasaktır	Sesli İkaz Cihazlarının Kullanımı Yasaktır	Gümrük - Durmadan Geçmek Yasaktır

Figure A.3 Prohibitory and Restrictive Traffic Signs [1]