TRAFFIC SIGN RECOGNITION
FOR UNMANNED VEHICLE CONTROL

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

BY

MEHMET BÜLENT HAVUR

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

NOVEMBER 2006

Approval of the Graduate School of Natural and Applied Sciences.

Prof. Dr. Canan ÖZGEN
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Prof. Dr. İsmet ERKMEN
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Assoc. Prof. Dr. Gözde Bozdağı Akar	Assoc. Prof. Dr. M. Mete BULUT
Co-Supervisor	Supervisor

Examining Committee Members

Assoc. Prof. Dr. A. Aydın Alatan	(METU, EE)	
Assoc. Prof. Dr. M. Mete Bulut	(METU, EE)	
Assoc. Prof. Dr. Gözde Bozdağı Akar	(METU, EE)	
Assist. Prof. Dr. Çağatay Candan	(METU, EE)	
Burcu Kepenekçi, M.S.	(TUBITAK)	

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: Mehmet Bülent Havur

Signature:

ABSTRACT

TRAFFIC SIGN RECOGNITION FOR UNMANNED VEHICLE CONTROL

Havur, Mehmet Bülent
M.S. in Electrical - Electronics Engineering
Supervisor: Assoc. Prof. Dr. Mehmet Mete Bulut
Co-Supervisor: Assoc. Prof. Dr. Gözde Bozdağı Akar

November 2006, 151 pages

In this thesis, video frames acquired by a camera in a moving car are processed for detection of candidates of triangular, rectangular and circular traffic/road signs based on mainly shape information by performing contour analysis. Color information is utilized as an auxiliary method to improve detection. Then recognition based on template matching is realized on detected traffic/road sign candidates. Detection and recognition results of traffic/road signs in video frames taken in different time intervals of day for these methods are compared.

After implementation, results show that the video scene taken in a sunny day in the afternoon gives better results than others. Binary threshold plays a great role in detection with respect to Canny edge detector especially for triangular and rectangular traffic signs. Higher number of binary threshold levels improves detection in general. In addition, the recognition rate for triangular and rectangular traffic/road signs is higher than that of circular sings in general by the methods used in this study.

Keywords: Autonomous Vehicle, Road Sign Detection, Road Sign Recognition.

ÖZ

İNSANSIZ ARAÇ KONTROLU İÇİN TRAFİK İŞARETİ TANIMA

Havur, Mehmet Bülent
Yüksek Lisans, Elektrik - Elektronik Mühendisliği Ana Bilim Dalı
Tez Yöneticisi: Doç. Dr. Mehmet Mete Bulut
Ortak Tez Yöneticisi: Doç. Dr. Gözde Bozdağı Akar

Kasım 2006, 151 sayfa

Bu tezde, hareket halindeki bir aracın içindeki kameradan elde edilen video görüntüleri üzerinde, kenar analizi ile şekil bilgisine dayalı üçgen, diktörtgen ve daire şekilli trafik/yol işaretlerinin tespiti yapılmıştır. Bunun yanında renk bilgisi trafik/yol işaretinin tespit oranını geliştirmek için kullanılmıştır. Tespit edilen trafik/yol işaret adaylarını tanıma işlemi, şablon eşleme yöntemi kullanılarak gerçekleştirilmiştir. Günün değişik saatlerinde çekilen videolardan elde edilen görüntülerden tespit ve tanıma sonuçları karşılaştırılmıştır.

Uygulama sonuçlarından, güneşli bir günde öğleden sonra alınan görüntüler üzerindeki tespit işleminin diğerlerine oranla daha iyi olduğu sonucu elde edilmiştir. İkili eşik fonsiyonu ile yapılan denemelerde Canny kenar yakalayıcısına oranla daha yüksek tespit oranları elde edilmiştir. Yüksek sayıdaki ikili eşik seviylerinin kullanılmasıyla daha iyi tespit oranları gerçekleşmiştir. Bu çalışmada kullanılan yöntemlerle genel olarak üçgen ve dikdörtgen şekilli trafik/yol işaretlerini tanıma oranlarının, dairesel olanları tanıma oranlarından daha yüksek olduğu sonucuna varılmıştır.

Anahtar Kelimeler : Otomatik Araç, Trafik İşareti Tespiti, Trafik İşareti Tanıma.

ACKNOWLEDGMENTS

I would like to thank, first of all, to my supervisors Assoc. Prof. Dr. Mehmet Mete Bulut and Assoc. Prof. Dr. Gözde Bozdağı Akar for their critics, guidance and support.

I would like to express my great gratitude to my family for their endless support and great love. Moreover, I would like to thank to my friends at ASELSAN Inc. where special thanks to Ahmet Bahtiyar Gül, Özkan Gönder, and Gökhan Özzaman for their recommendations. I would also like to thank to Burcu Kayar for her great support and her help especially in acquiring the video scene including traffic signs and in grabbing them. Finally I would like to thank to my company Aselsan for their support and understanding during this study.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

GA	: Genetic Algorithm
HLS	: Hue Luminance Saturation
HSI	: Hue Saturation intensity
HSV	: Hue Saturation Value
IHLS	: Improved Hue Luminance Saturation
PCA	: Principle Component Analysis
RGB	: Red Green Blue
ROI	: Region of Interest
SVF	: Simple Vector Filter

CHAPTER 1

INTRODUCTION

When considering the 20th century, the invention of motor vehicle is one of the greatest technological advances affecting human-being's economical activities and life. In the 21st century, it is expected that the advances in information and computer science will be driving technology. The main curiosity is how computer science and automotive industry are integrated to create intelligent vehicles that can advance safety by helping driver even that can make unmanned/autonomous vehicle possible [3]. Current studies and future trends show us that, recent support systems will take important role as driver-aid system evaluating towards fully autonomous vehicle in the future [2].

Today, drivers maintain full control over their vehicles and driving is a task which is based fully on visual information. Traffic signs constitute a visual language which has a vital importance for driver to navigate vehicle successfully in safe [13]. They give information about current traffic situation, warn about risky factors, prohibit or permit for certain directions thus provide useful and necessary information that makes driving safe and convenient [13]. According to one research, 60 percent of crashes at intersections and 30 percent of head-on collisions could be avoided if the driver had an additional half-second to react and about 75 percent of vehicular crashes are caused by inattentive drivers [3]. The reason of many accidents is the failure of a driver to notice traffic sign, either because of insufficient attention at a critical moment or due to adverse conditions that reduce visibility [3]. At night, drivers are easily blinded by the headlights of oncoming vehicles. In bad weather (e.g., rain, snow, and fog) road signs are less likely than normal to attract a driver's attention [3]. All these situations make driving harder and may cause more traffic accidents [3]. Therefore it motivates that if some warning systems providing driver with being aware of situation and risk even a few hundred milliseconds earlier by notifying to the presence of traffic sign may prevent most of the accidents. As a part of driver-aid system, traffic sign detection and recognition module can serve in this mission.

Studies on a vision-based vehicle guidance system for road vehicles focus on three main missions: 1) road detection; 2) obstacle detection; and 3) sign recognition. The first two have been studied for many years and with many good results, and traffic sign recognition is still an open search area [7].

In the future, as a field of applied computer vision area, traffic sign detection and recognition system is one of the important parts of future autonomous vehicle in collaboration with GPS based navigation system or system with magnetic sensors or communicating system with other cars or direction center etc.

The studies have been carried out extensively for over 20 years and many algorithms have been introduced since then. Most of the studies have been carried out with video streams acquired from car moving along the roadway by some speed. After considering these studies, the problem of detection and recognition traffic signs can be stated as follows:

-The traffic scene images often suffer from vibrations [2].

-Color and contour information is affected by varying illumination [2], [5], [13].

-Environmental influences such as bad weather, dirt and achromatic traffic signs [5].

-Traffic signs are occluded partially by other objects [2], [13].

-Many objects are present in traffic scenes which make the traffic sign detection hard, because other vehicles, buildings and billboards may confuse the detection system [2], [13].

-Moreover the algorithms should be suitable for the real-time implementation [2].

Traffic sign detection and recognition system can work as follows: First traffic sign candidates are detected all over the video frame using color, shape and other possible features, then ROI for detected candidates are pre-processed (cropped, rotated if necessary, resized, normalized) and fed to trained recognition module. If recognition module does not classify input as a traffic sign then detected candidate is discarded.

1.1. Scope of the Thesis

In this thesis, the detection and recognition of traffic signs were performed by using video streams which are taken in different time intervals of day (day time, evening time, night time) and in different environments (rural area, urban area) where these factors include considerable variation of illumination and background. Then video frames were acquired offline by grabber to convert into digital format. After obtaining certain video frame, two detection techniques are applied in detection module. First one is mainly based on segmentation according to shape after which other features and heuristics are considered whereas second one is mainly based on color segmentation after which shape feature is utilized [1], [5], and [7]. Next step of detection is recognition. After utilizing some pre-processing techniques, the traffic sign candidates found in detection module are fed into recognition module. In recognition module, the candidates found in detection module are pre-classified according to their shapes. These shapes are triangle, rectangle and ellipse, which was perspective form of circle. After that, Template Matching based methods are to be applied for classification in recognition module. The results are compared with respect to time of day which affects mostly illumination and the area which affects background of traffic sign causing contours to be less detecTable.

1.2. Thesis Outline

The outline of the thesis is summarized as follows:

In Chapter 2, a brief survey of traffic sign detection and recognition techniques is given. In this chapter, different detection and recognition techniques introduced so far are presented. The algorithms used to detect traffic signs in this thesis and their theoretical backgrounds will be summarized. These algorithms which are part of the detection system are used for detection of triangle, rectangle and ellipse (which is perspective form of circle) where these shapes are the template shapes of traffic signs by two different approaches. First is mainly shape detection and second is mainly color segmentation.

In Chapter 3, the application of algorithms which are the part of the detection system used for detection of triangle, rectangle and ellipse (which is perspective form of circle) will be explained.

In Chapter 4, recognition of traffic signs will be explained.

In Chapter 5, system developed for traffic sign detection and recognition is described, and results drawn will be drawn and discussed.

Finally in Chapter 6, conclusion will be drawn and some possible future studies will be discussed.

CHAPTER 2

SURVEY ON DETECTION AND RECOGNITION

2.1 Survey on Traffic Sign Detection and Recognition

2.1.1 Introduction

The traffic sign detection and recognition is a computer vision field which serves as a driver-aid system by first helping driver and then by fully replacing driver in the future as being one of the components of autonomous vehicle. Computer vision has mainly three tasks to make road vehicle fully autonomous [7]. First one is the road detection and following, which has been studied for a long time and many results have been drawn. Second one is obstacle detection and avoidance, which has been contributed extensively since 1980's and is still an open research area [7]. The last one is traffic sign detection and recognition which is the latest to start among others [7].

The purpose and structure of traffic sign recognition system is well defined and quite clear. The system platform consists of video camera seen of which is adjusted to catch traffic scene in front of driver, some processor unit to evaluate video frames coming from video camera altogether mounted onto vehicle. The main purpose is to recognize the traffic signs in seen. At the first sight, solving this problem seems to be quite simple because shapes, colors, and pictograms of traffic signs are known and the position of them in traffic scene usually occurs at certain places in seen [2].

However the problem is not as simple, the adverse effects discussed in Chapter 1 make traffic sign detection and recognition problem more complex. Moreover traffic sign detection and recognition algorithms should be suitable for real-time implementation. Considering them, the approach to recognizing traffic signs is first making the search area to be recognized small which is formally called detection and then candidate objects can be fed into recognition module. Recognition module has two functions: It classifies the objects if it is a traffic sign, and if not, it discards the candidate object. The tracker module can be added to take advantage of video stream, which provides less effort for redundant information.

Many algorithms for detection and recognition of traffic signs have been introduced up to now. The summary of these studies will be given in this chapter.

2.1.2 Detection

Traffic signs play a major role to keep flow on roads and highways safe, fast and reliable. Their mission is critical in many cases. However there are many objects in seen and traffic signs should be distinguished among them as easily as possible. For this reason they have some features to attract driver's attention to improve distinguishability of traffic signs. These features are mainly color and shape. There are international standards regulating shape and color of traffic signs according to their classes. Below are the Tables which give feature value and its meaning for color and shape [13].

Table 2.1 Standard traffic sign background colors and their meanings [13]

Color	Meaning
Red	Prohibition and warning
Blue	Directive
Green	Guidance and mileage
Orange	Construction and maintenance
Brown	Recreation
Yellow	Warning
White	Auxiliary

Table 2.2 Standard traffic sign shapes and their meanings [13]

Shape	Meaning
Circle	Prohibition
Equilateral triangle, pointing up	Warning
Equilateral triangle, pointing down	Yield
Octagon	Stop
Rectangle	Regulation and guidance

As seen from the Tables, traffic signs are designed in pre-defined, fixed 2-D shapes such as triangle, rectangle, octagon and circle [1], [37], [38]. The information on the sign has one color and the rest has another color.

The philosophy behind the idea is that the features to ease human driver perception of traffic sign can make the computer perception of traffic sign in the worst case with cluttered background and under adverse illumination condition. It is the prior purpose of traffic sign detection systems that traffic sign detecTable by human-driver is also detecTable by computer. In other words, computer-aided detection systems should be as good as at least human-driver. By this thought many researches have been done and many results have been reported on detection of traffic signs.

Most of the studies carried out up to now start with describing model of traffic signs with color feature [1], [7], [14], [15], [16], [17], [18], [20], [21], [22], [23], [24], [25], [28], [30], [33], [34], [35], [36], [42]. After extracting color feature, shape feature is utilized which is considered as complementary feature to detect traffic sign candidates in these studies. Bahlmann et al. [29] states the drawback of this sequential approach of color and shape detection since regions falsely rejected by the color segmentation can not be recovered in the further processing. However there are some studies using color and shape features jointly [13], [29]. Moreover some researchers have analyzed the use of temporal information by taking advantage of successive frames and usage of tracking module to increase performance of detection module [1], [13].

As a result traffic sign detection can be classified in the three following approaches:

- Segmentation through color threshold, region detection and shape analysis on color segmentation.
- Segmentation through the border detection in a monochrome image and their analysis.
- Segmentation on border detection and color threshold jointly.

2.1.2.1 Color Feature Extraction

As shown in Table 2.1, there are dominant colors used on traffic signs which are red, blue, green, orange, brown, yellow and white. These colors can be set with certain thresholds and their inter-relationships can be used to localize traffic sign in the seen [13].

The outline described below is generally followed as a method of detection with color. First a suitable color space or spaces are selected then some segmentation algorithm is applied for desired color. After that the binary image obtained is processed and the algorithms are utilized to provide group of pixels with meaningful localization of traffic signs hopefully [13].

There are several color spaces to localize a traffic sign. While color segmentation in RGB space is performed in some studies by applying algorithms on basic relationship between R, G and B components, in others some more complicated linear and non-linear transformations of RGB space are applied. Among them, the most popular ones are using HSV, HIS, HLS and IHLS (which is improved HLS by Hanbury and Serra [39]) color spaces, and extracting color information based on hue component.

HSI model is considered as most suitable for traffic sign detection by Fang et al. [13] because it represents human color perception [25] where colors of traffic sign are originally chosen to attract human attention. Piccioli et al. [1] reported their algorithms developed mainly based on hue component to localize traffic signs. Moreover, they set minimum threshold of saturation component as %20 to ignore unsaturated pixels where hue value may not correspond to true color value. There are some further color connection algorithms to improve performance of color feature extraction. Piccioli et al. [1] suggested subdividing image 16x16 pixels region and classifying each region as '1' if the number of labeled pixels for certain color exceeded certain threshold (~1/3 of total number of pixels in 16x16 region which is

80 in their cases) and otherwise '0'. Then search region was associated with every cluster of '1' regions.

Pacheco et al. [11] also used HSI color space to segment probable colors of traffic sign candidates and moreover described their special hardware to convert RGB color space to HSI color space. Shape feature is not utilized for detection, instead color coding is used that after finding the number of primary color pixels given in Table 2.1 above threshold, another primary colors used in the traffic signs next to segmented pixels are checked. If enough number of such pixels is found, the region is decided as location of traffic sign.

Liu, Liu and Xin [12] make use of hue component together with Simple Vector Filter (SVF) and edge detection jointly. They apply the distinction of chromatic and achromatic colors. The SVF is defined as a method which is used for removing contour by using achromatic color. This is achieved due to fact that achromatic color often appears in the contour part. HSI Table is used and when the direction of vectors is same the elementary colors are also same. So, achromatic color is expressed by the same direction vector along with chromatic one. However they reported that this method is greatly influenced by the change of illumination. So the method can only be applied for traffic scene taken in a day and under good weather conditions.

In their later studies in 2001 and 2003, Escalera et al [35], [43] also start detection phase with color segmentation by using HSI color space. Only hue and saturation components are taken into account and value component is discarded. Then hue value for reference color and saturation are applied by threshold function. Figure 2.1 illustrates threshold functions for hue, red and saturation from which two lookup Tables are constructed. Correspondence for 255-level image is 0 to red, 85 to green, 170 to blue and 255 to red again.

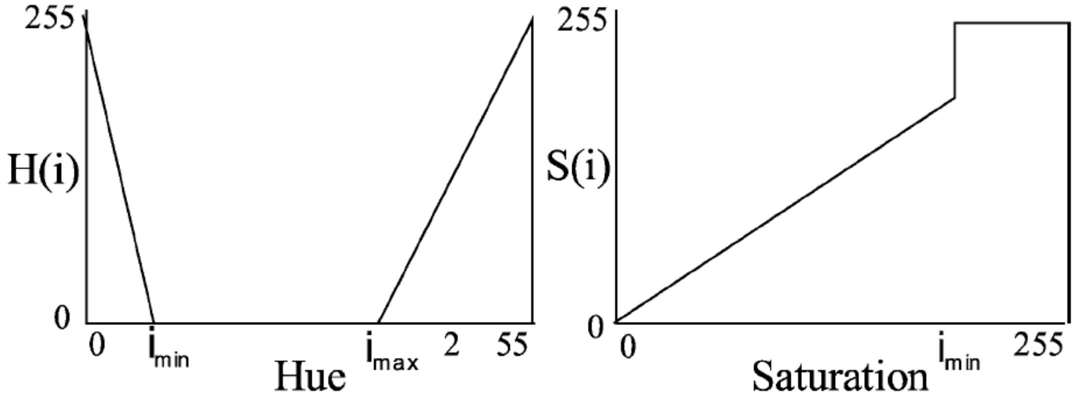


Figure 2.1: Hue for red and saturation transfer functions [35] [43]

The formula for threshold H(i) is given in formula 2.1:

$$H(i) = \begin{cases} 255\dfrac{i_{\min} - i}{i_{\min}} & 0 \le i \le i_{\min} \\ 0 & i_{\min} \le i \le i_{\max} \\ 255\dfrac{i - i_{\max}}{i_{\max}} & i_{\max} \le i \le 255 \end{cases} \tag{2.1}$$

The formula for threshold S(i) is given in formula 2.2:

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

In his study, Fleyeh [28] also used hue component. While saturation and intensity are influenced by illumination (light or shadow) hue component is largely invariant to such changes in daylight [25]. This is due to fact that it is invariant to the variations in light conditions since it is multiplicative/scale invariant, additive shift invariant and it is invariant to saturations change [28]. For this reason hue is selected by most of the researchers in traffic sign detection systems. However there are some drawbacks:

- Small changes in the RGB can cause large variation in hue.

- When the intensity is very low or very high, hue does not correspond to any color.
- When the saturation is too low, hue does not correspond to true color.
- When the saturation is less than a certain threshold, hue becomes unsTable.

Moreover Aoyagi and Asakura [44] note that hue component changes with distance, weather and sign age.

Vitabile et al. [34] defined three regions in HSV color space.

- Achromatic area, where $S \leq 0.25$ OR $V \leq 0.2$ OR $V \geq 0.9$.
- UnsTable chromatic area, where $0.25 \leq S \leq 0.5$ AND $0.2 \leq V \leq 0.9$.
- Chromatic area, where $S \geq 0.5$ AND $0.2 \leq V \leq 0.9$.

Then Vitabile et al. [34] applied subdivision on image after segmentation. In each sub-region, local luminance was taken in account to avoid global image illumination differences. After applying dynamic shareholding for Euclidian distance in the cylindrical HSV color space, color segmentation for traffic signs was completed.

Fleyeh [28] applied the method of Hanbury and Serra [39] which is called IHLS (Improved Hue Luminance Saturation) to localize traffic signs. It is claimed that IHLS avoids the inconveniences of the other color spaces designed for computer graphics rather than image processing [28]. The color space ensures the independence of chromatic and achromatic components [40]. The formula to calculate IHSL components from RGB is as follow:

$$H = \theta \qquad \text{if } B \leq G$$

$$H = 360 - \theta \qquad \text{if } B \geq G$$

Where:

$$\theta = \cos^{-1}\left(\frac{\left[R - \frac{G}{2} - \frac{B}{2}\right]}{\sqrt{R^2 + G^2 + B^2 - RG - RB - GB}}\right) \tag{2.3}$$

The Other two parameters are calculated as follows:

$$S = \max(R, G, B) - \min(R, G, B) \tag{2.4}$$

$$L = 0.212R + 0.715G + 0.072B \tag{2.5}$$

Color is affected by chromatic variations [28]. This is due to:

- Intensity and position of incident light onto object.
- The reflectance properties of the object.

Fleyeh [28] applied some methods based on IHLS to detect traffic signs:
First method make use of luminance such that, dynamic threshold value related to average luminance, namely thresh, is calculated for each image sequence by the formula below:

$$\begin{aligned} mean &= \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1} L(i, j) \\ Nmean &= mean/256 \\ thresh &= e^{-Nmean} \end{aligned} \tag{2.6}$$

The reference color (such as red for obligatory traffic signs) and unknown color are represented by hue and saturation colors as shown in Figure 2.2.

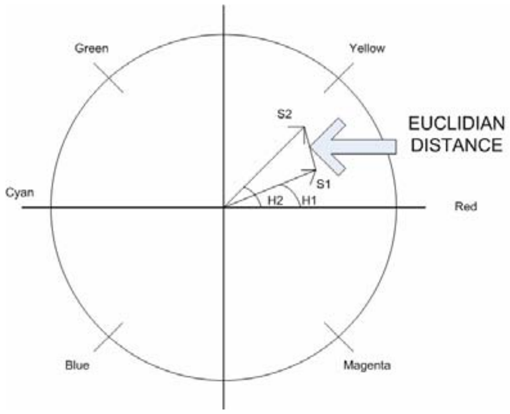


Figure 2.2: The vector model of hue and saturation [28]

Then, the Euclidian distance, d between two vectors is calculated as follow:

$$d = \sqrt{\left(S_2 \cos H_2 - S_1 \cos H_1\right)^2 + \left(S_2 \sin H_2 - S_1 \sin H_1\right)^2} \qquad (2.7)$$

Pixel is considered as an object pixel if $d \le thresh$. The key idea of Fleyeh [28] in this method is that brightness has a dynamic control over the relation between reference pixel and unknown pixel.

Another method that Fleyeh [28] applied is a modified version of Escalera et al [35]. In this method the RGB image is converted to IHLS instead of HSI color space for the case of Escalera et al. [35]. Both hue and saturation are normalized between 0 and 255. To avoid achromatic area that Vitabile et al. [34] defined, Fleyeh [28] choose $S_{min} = 51$, $S_{max} = 170$. After that the saturation is formulated as follows:

$$S_{out} = \begin{cases} 0 & 0 \le S_{in} \le S_{\min} \\ S_{in} & S_{\min} \le S_{in} \le S_{\max} \\ 255 & S_{\max} \le S_{in} \le S_{\max} \end{cases} \qquad (2.8)$$

Figure 2.3 illustrates (2.8).

The hue is formulated by:

$$H_{out} = \begin{cases} 255 & H_{min} \le H_{in} \le H_{max} \\ 0 & otherwise \end{cases} \tag{2.9}$$

Figure 2.4, 2.5, 2.6 illustrates formula (2.9) the extraction of red, green and blue respectively. The difference from Escalera et al. [35] is that while Escalera et al. [35] prefers ramp function, Fleyeh [28] prefers unit function for hue and saturation thresholds.

The logical AND operator between S_{out} and H_{out} give a binary image which represents a candidate of traffic sign.

Figure 2.3: Saturation transfer function [34]

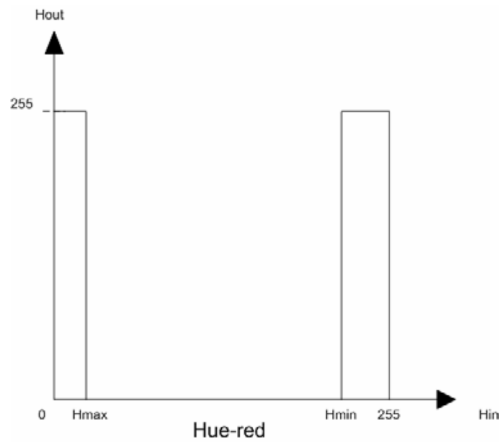


Figure 2.4: Hue transfer function for red [28]

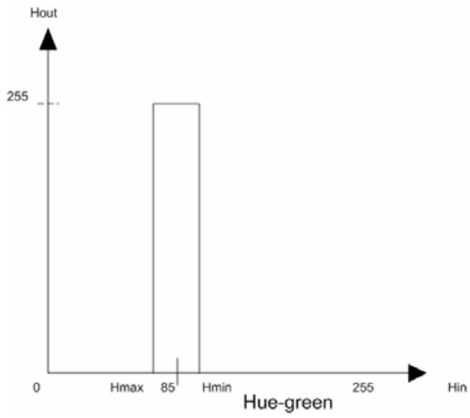


Figure 2.5: Hue transfer function for green [28]

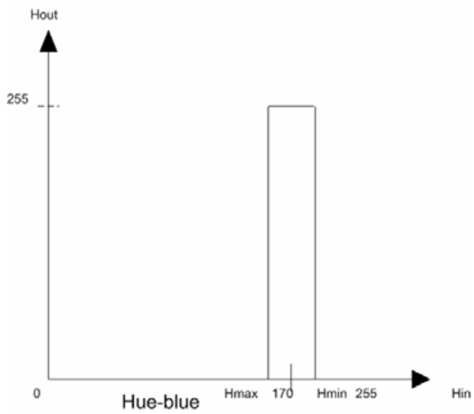


Figure 2.6: Hue transfer function for blue [28]

Here the key idea of Fleyeh [28] is selecting true color by avoiding achromatic and unsTable chromatic areas defined by Vitabile et al. [34]. Fleyeh [28] reported that after applying methods described above on more than hundred images with various illumination, first method gave a better result with respect to second described method. Fleyeh [28] commented this result that since first method made use of dynamic threshold with respect to luminance, it was able to distinguish wide range of color under different illumination whereas other method detected similar colors.

There are different approaches. In their earlier study in 1997, Escalera et al. [7] state that computation cost of calculating hue component is too much if special hardware is not used and they developed RGB color threshold algorithm extending the idea suggested by Kamada and Yoshida [26]. They used relationship between RGB color components. They choose a reference color according to type of traffic sign and apply threshold, red for their example. They defined a threshold function g(x,y) for detecting red component:

$$g(x,y) = k_1 \begin{cases} Ra \leq fr(x,y) \leq Rb \\ TGa \leq \dfrac{fg(x,y)}{fr(x,y)} \leq TGb \\ TBa \leq \dfrac{fb(x,y)}{fr(x,y)} \leq TBb \end{cases} \tag{2.10}$$

$$g(x,y) = k_2 \text{ otherwise}$$

where,

- g(x,y) is the decision output of color threshold,
- fr(x,y), fg(x,y), fb(x,y) are the red, green, blue components of each point of image respectively,
- TGa, TGb, TBa, TBb are the threshold values corrected for denominator fr(x,y) where constant T is added with denominator.

In order to reduce computational cost and achieve real-time process, Escalera et al. [7] used a 16-bit lookup Table instead of direct each time-calculation for threshold.

Ghica et al. [18] used the vector distance based threshold in RGB color space. They defined Red, Green and Blue as basis vectors and formulated color c as:

$$c = c1 * Red + c2 * Green + c3 * Blue, \tag{2.11}$$

where $0 \leq c1, c2, c3 \leq 1$

Given a reference color, for example red for stop sign, r=(r1,r2,r3), the distance d between an unknown color c and reference color r is as follow:

$$d = \sqrt{(r1 - c1)^2 + (r2 - c2)^2 + (r3 - c3)^2} \tag{2.12}$$

The threshold function F is given by:

$$F(c) = \begin{cases} (0,0,0) & if \quad \|r-c\| \le t \\ c & if \quad \|r-c\| > t \end{cases} \tag{2.13}$$

Where, t is the suitable threshold.

Their detection module relied on color segmentation. They did not use shape information, instead they applied morphological filter as they named. The idea was based on color connection. That is, relationship between color outputs was utilized to remove non-sense pixels (such as object pixel with value ‘c’ surrounded by many non-object pixels ‘(0,0,0)’) of function F which is binary image.

Another basic approach by Janssen et al. [5] was implemented which was a color conversion algorithm from RGB. To gain robustness against illumination variations, they used normalized color space Irg:

$$I = \frac{R+G+B}{3} \qquad r = \frac{R}{3I} \qquad g = \frac{G}{3I} \tag{2.14}$$

Where ‘I’ corresponds to intensity, r corresponds the normalized red, g corresponds to normalized green and R, G, B correspond to red, green, blue signal of camera respectively. They use a lookup Table like Escalera et al. [7] instead of direct each time-calculation for threshold.

Another color space, YIQ was used by Kehtarnavaz and Ahmad [14]. They reported that YIQ donated with unsupervised artificial neural network gave the best results of segmentation with respect to other possible color spaces they considered (RGB, HSI, YUV, XYZ, UVH) on their data set which is composed of red (warning) and yellow (caution) traffic signs.

Miura, Kanda and Shirai [30] use YUV color spaces to extract traffic sign candidates. Shaded, Nadi and Mismar [8] also uses YUV but with HSV. That is, their approach to traffic sign detection problem on color feature is using two color spaces instead of one. They stated, by incorporating two color spaces they obtained better segmentation by overcoming some deficiencies of single one. They used YUV color space incorporated with HSV. In their algorithm, first they converts RGB image to YUV color space. Then they make use of the fact that since chrominance components, U and V of YUV color space are independent of luminance, U and V can be used to represent color information quite well. They convert RGB to YUV as follows:

$$Y = 0.229R + 0.587G + 0.114B \tag{2.15}$$

$$U = 0.492(B - Y) \tag{2.16}$$

$$V = 0.877(R - Y) \tag{2.17}$$

Then they equalize the histogram on Y component corresponding luminance. After that, in order to improve image luminance without changing chrominance, three thresholds in the histogram of Y image which are the average value of Y, upper average value of Y histogram and lower average of Y histogram are set and some mapping are performed. After all, the YUV image is converted back to RGB color space. Shaded, Nadi and Mismar [8] states that the process above can be repeated more than once to reach more sTable condition. After completing these processes, RGB image is converted to both YUV and HSV color spaces. H represents color as discussed before. U and V also represent color information where U is positive if the blue is greater than certain percentage of red and green, and V is positive if the red is greater than certain percentage of green and blue. For example to segment red pixels from others, V value is used and the logical AND is applied to combine V value of YUV and H value of HSV, where H value is obtained in a similar manner to Vitabile et al. [34]. Then Shaded, Nadi and Mismar [8] discussed on two examples in their paper that while one of them is segmented in YUV color space well, the other is segmented in HSV color space. By using two of them can provide extra benefit to segment and localize traffic sign.

2.1.2.2 Shape Feature Extraction

As seen in Table 2.1, the shapes used for traffic signs are some primitive 2-D geometrical shapes which are triangle, rectangle, circle and octagon.

Generally shape feature extraction may be classified in two approaches:

i. Segmentation on border detection in a monochrome image
ii. Shape detection after color segmented image

For both of the approaches, monochrome image to be segmented on border detection, or feature image after color segmentation is manipulated to extract shape feature and detect traffic sign. There are several ways to do it:

Piccioli et al. [1] represent edge image by Canny's algorithm [4] applied to color segmented image (ii). Then different approaches are applied to detect different geometrical shapes. After obtaining edge image, to detect triangle shapes, a polygonal approximation of the edge chains inside the search region (color segmented region coming from previous step) to eliminate part of the chains strongly departing from a straight segment by Piccioli et al. [1]. Angles between line segments are extracted to decide triangles finally. Detecting circles with a similar method described above is more difficult because results are unsTable and thus there occurs poor rate of success [1]. Piccioli et al. [1] use a different method to detect circle, which is mainly based on radial and angular distribution of the edges. They [1] compare their methods to detect circles with that of Etemadi [45], Hough [46] and Masciangelo [47] of whom algorithms are generally used to detect elliptic curves in image processing.

Garcia et al [6] also have an approach similar to that of second group (ii). Although horizontal and vertical edge images are calculated first, color information is used to validate and segment the candidate areas. After that candidate locations are applied blob shape analysis and circular ring template matching to detect circular traffic signs.

The earlier study of Escalera et al. [7] for shape detection in 1997 is utilizing corner detector. The detector works directly on image obtained after color segmentation step discussed in 2.2.1. Corners are detected from the convolution of the image with a mask. Some 3x3 and 9x9 masks are used as a mask according to the shape to be detected. If the triangle shape is to be detected, the mask emphases triangle corners, if the rectangular shape is to be detected, the mask emphases rectangular corners. After convolution, the regions exceeding some threshold value are accepted as corners and labeled according to the type of mask detecting it. Candidate circles are detected by using same mask of rectangular one. Then considering geometrical relationship between labeled corners, geometrical shapes are detected. However in their later studies in 2003, Escalera et al. [35] use Genetic Algorithm to detect traffic signs.

Yabuki et al. [9] experiments active net to detect traffic signs. Active net is a deformable model minimizing energy function to detect target region. They have two different approaches to localize traffic signs. First one is an approach with segmentation on border detection in a monochrome image (i). Second one is an approach applied on color image (ii). In the first approach active net is applied in two steps. The first active net roughly estimates a position of target region; the second one is applied around the estimated region. This procedure is carried out on monochrome image in the first approach. However in the second approach it is applied for color image. First color distribution function is calculated and color distribution image is constructed (i.e. intensity of pixel corresponds color similarity). Then two active net applied procedures in the first approach are applied to color distribution image.

Liu, Liu, Xin [12] applies edge detection with color extraction jointly as mentioned in chapter 2.2.1. Edge filter given by (2.18) is used. After that Genetic Algorithm (GA) based technique is applied to detect a circular traffic signs by treating both the position and size of traffic sign as gene information.

$$L_g(i, j) = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} \tag{2.18}$$

Fang et al. [13] approach is to detect traffic sign based on color segmentation (ii). Color segmented image (mentioned in chapter 2.2.1) is used to extract shape feature by using two-layer neural network. For the shape to be detected, weights are arranged so that neural network emphasis related shape and discard others.

EsTables et al. [20] method is a bit different and has an approach based on learning. Contours are found from knowledge sources and system increases knowledge about undefined shapes. Knowledge sources have to be finely tuned to discriminate shape classes.

Estevez and Kehtarnavaz [21] apply edge detection based on RGB differencing. Edge is decided by following formula:

$$(Edge)p = abs(R - Rp) + abs(G - Gp) + abs(B - Bp) \tag{2.19}$$

Where Rp, Gp, Bp represent values associated with preceding pixel.

Then local maxima operator is applied to find exact location of edge where the operator can be defined as following algorithm:

IF Difference[x] GREATER THAN Difference[x-2] AND

Difference[x] GREATER THAN Difference[x+2] (2.20)
Difference[x] = 1, ELSE Difference[x] = 0

Miura, Kanda and Shirai [30] try to find circular region by using color segmentation as mentioned in chapter 2.2.1 and then by using shape extraction (ii). After color segmentation by using YUV color space, edges in the search region found in color segmentation are extracted. Their approach to detect circular shapes is based on image gradient direction. The idea is that the line in the same direction with gradient of edge passing through that edge point also passes through center of the circle. To detect rectangular signs four line segments are detected by threshold strongest edges.

Vitabile et al. [33], [34] applies similarity analysis to color segmented traffic scene images for detection of specified shapes. Their method assumes that sample image and segmented image are the same size, and in order to eliminate rotation errors, segmented image is rotated between -5 and +5 degrees by 1 degree step. The similarity function between two binary images is found by the Tanimoto distance measure [48]. Circular prohibitory and regulating road signs (also speed limit signs), triangular warning signs and circular mandatory direction signs are tested.

The approach for detection in the study of Loy and Barnes [41] is similar to that of Miura, Kanda and Shirai [30] such that image gradient direction is utilized. Each gradient element votes for circle center for circular shape detection.

The shape detection part of Mo and Aoki’s study [36] is based on axial symetry feature of all traffic signs. First all symetry axisis in image are found then color information is used to test regions if traffic sign or not (i).

As a result there are several approaches utilizing Genetic Algorithm [12] [35] [44], contours by polygonal approximation [1] for polygonal shapes, Hough Transformation [46], Etemadi [45] and Masciangelo [47] algorithms to find elliptic arcs [1], template matching [6], corner detection with mask [7], active nets [9], neural networks [13], similarity analysis with distance measure [33], [34] after edge detection for shape extraction.

2.1.3 Recognition

There are a variety of traffic signs and although international standard exists they differ from country to country and even region to region. That makes the cost of the system expensive and recognition process difficult because the number of signs to be recognized is huge. There are some methods developed generally for recognition in computer vision and they were used in recognition process for traffic signs.

Piccioli et al. [1] use the normalized cross-correlation between the road sign candidate found in detection phase and templates in the road sign data base to recognize. They have three assumptions such that the orientation of road sign candidate found in detection phase is almost constant, the lighting and resolution range of candidate sign can be large in scale and process should be real-time. They apply some pre-processing. First, image is converted to grayscale if it is color image then normalized to a size of 50x50 pixels by linear interpolation. Piccioli et al. [1] reports that the correct classification rate is about 98% even for as small as 25x25 size road sign candidates. They also report that they have a 60 circular signs and 47 triangular signs. However they do not report under what kind of weather conditions and at which times of day they experimented.

Miura et al. [30] also use normalized cross-correlation like Piccioli et al. [1] and they reported 100% recognition success rate for their experiment.

Janssen et al. [5] uses weighted distance metrics however they do not report details of what feature space they apply. Nearest Neighbor (NN) Algorithm is utilized. Since most of the pictograms of traffic signs are black and white, they make use of this fact and consider monochrome data on Region of Interest (ROI).

The earlier study of Escalera et al. [7] for recognition in 1997 is utilizing neural networks with multilayer perceptrons. Two different neural networks for each of triangle and circular signs are structured and three different topologies for hidden layers are applied in their studies. The number of perceptrons in input layer corresponds to number of an image pixels (30x30 in their case) and number of output perceptrons is the number of road signs in training set and plus one which is for rejecting as non-road sign.

Si Wei Lu [15] reports in 1994 that neural network composed of three subnetworks is used for traffic sign recognition. These subnetworks are organized in hierarchical manner such that output of the previous subnetwork is input of successor subnetwork. The first subnetwork whose elements are sigmoid neurons is a square array to extract the features of size normalized traffic sign. Second subnetwork is used to get rid of rotation and noise corruption. The third subnetwork is used for classification.

Ghica et al. [18] in 1994 use neural network to recognize traffic signs. Some preprocessing techniques such as morphological filters are applied to improve recognition. Then the image is normalized and fed into neural network which is composed of classification subnetwork, Hopfield network, and validation network. They report that the speed of the system is suitable for real-time process. It is stated that under good illumination good results are reported however under poor illumination identification rate is low.

Kehtarnavaz and Ahmad [14] also use neural networks for recognition. Their neural network consists of 16 input nodes, 48 hidden layer nodes, and 5 output nodes. They train their network according to back-propagation algorithm. They have four output nodes for four different traffic signs and one output for no-road sign. That is, they experimented to recognize among four different road signs in 1995 by their studies. The approach is to feed network with a Fourier transform of image with log-polarexponential grid. It is stated that scale invariance and rotation invariance are provided.

Vitabile et al. [33] in 2001 are another to use neural classifier to recognize traffic signs. Experiments were carried out both at day and night time and they report 100% identification success rate for blue circular signs, 88% for red triangular signs, and 84% for red circular signs.

Priese et al. [16] utilize four modules to classify road signs. Their approach is based on ideograms as they name. The first module finds position and direction of arrows. The second module identifies numerical symbols. The third module is a nearest neighbor classifier which is applied on prohibition sign ideograms, speed limits, arrows on mandatory signs. The fourth module is neural network.

The approach of Bahlmann et al. [29] is probabilistic modeling with unimodal Gaussian probability densities after Linear Discriminant Analysis (LDA). They experimented 30 minutes video with resolution of 384x288 at day time. %6 classification error was reported and the reason they stated is mostly confusions between similar signs.

Paclik [32] in 2000 make use of Laplace Kernel Classifier for recognition. Error rate between %0 and %14.2 were reported for nine different groups of traffic signs.

2.2 The Theoretical Background of Tools Applied for Detection

In this section, the algorithms applied to detect traffic signs and theoretical background of the implemented methods will be explained. The applied algorithms used for detection in this thesis are widely being used in computer vision and they are not novel techniques. Moreover most of the techniques mentioned in this chapter either itself or equivalent were used in traffic sign detection as explained in the chapter 2.

The Section 2.2.1 is dedicated to detection via shape feature extraction, whereas the section 2.2.2 is dedicated to detection via color feature extraction.

2.2.1 Detection via Shape Feature Extraction

In this section, basics of detection; filtering, morphological operators, edge detection, contour establishment, segmentation to be used for traffic sign detection will be explained and discussed.

2.2.1.1 Gaussian Pyramid

Gaussian pyramid is a filter i.e. convolution kernel which is used for low-pass filtering to filter out noises. It can be used further as image encoder with Laplacian Pyramid however it is irrelevant and out of scope for this study. In this study, its use is to filter out unnecessary details and noise by low-pass effect it has.

It was first proposed by Burt and Adelson [49] to serve for the purpose of image encoding. While Gaussian Pyramid is applied, the image size is reduced by the factor of 1/2 for each dimension, and thus by factor of 1/4 in two dimensions.

2.2.1.1.1 Definition

Burt and Adelson [49] define the image to be filtered out as $I_0[i, j]$ or shortly I_0 which corresponds to image intensity value at i and j. Then I_0 contain C columns and R rows of pixels, where $0 \le i < C$ and $0 \le j < R$. This image, I_0 is the basement or zero level of the Gaussian pyramid [49]. Pyramid level-1 represents image, which is obtained by low-pass filtering and 1/2 resampling I_0. While calculating the values of I_1, weighted average values I_0 within a 5x5 window is calculated [49]. Actually window size can be different but 5x5 window is chosen since it is enough to obtain accepTable results as experimented [49]. The same procedure is applied to obtain pyramid level-2 representing image, I_2 from pyramid level-2 representing image, I_1

as applied to get level-1 representing image, I_1 from pyramid level-0 representing image, I_0. Graphical representation of this process in one dimension is given in Figure 2.7 [49].

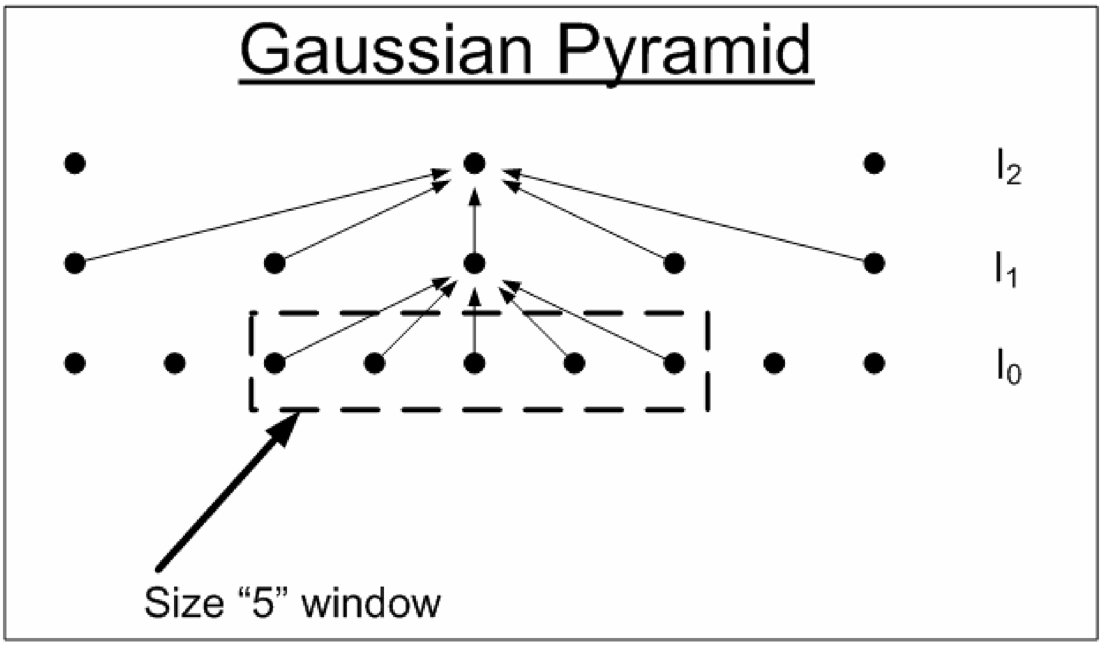


Figure 2.7: 1-D representation of Gaussian Pyramid [49]

2.2.1.1.2 Gaussian Pyramid Generation

The low-pass filtering is done step by step with function defined by Burt and Adelson [49] as they named REDUCE.

$$I_k = REDUCE(I_{k-1}) \tag{2.21}$$

The number of columns of pixels and the number of rows of pixels are halved by each REDUCE step. So number of the columns and number of rows depend on the level. Define C_l and R_l as number of columns and rows respectively for the given level of l [49].

For levels $0 \le l < N$ and for the node coordinates i, j: $0 \le i < C_l$, $0 \le j < R_l$.

$$I_l(i,j) = \sum_{m=-2}^{2} \sum_{n=-2}^{2} w(m,n) I_{l-1}(2i+m, 2j+n) \qquad (2.22)$$

Where w(m,n) is a generating kernel, Gaussian filter, N refers to the number of levels in the pyramid, C_l and R_l are the number of the columns and rows pixel of lth level respectively [49].

As seen in Figure 3.1, density is reduced by half in one dimension, or by a fourth in two dimensions by each application of REDUCE. Burt and Adelson [49] state that, the dimensions of the original image are suitable for pyramid construction if integers M_C, M_R, and N exist such that

$$C = M_C 2^N + 1 \qquad (2.23)$$

$$R = M_R 2^N + 1 \qquad (2.24)$$

Then the dimensions of I_l occur $C_l = M_C 2^{N-1} + 1$ and $R_l = M_R 2^{N-1} + 1$.

As Burt and Adelson [49] mentioned, the general properties of generating kernel, w(m,n) is chosen such that:

- It is separable.

$$w(m,n) = \hat{w}(m)\hat{w}(n) \qquad (2.25)$$

- The one dimensional, length 5 function ŵ is normalized.

$$\sum_{m=-2}^{2} \hat{w}(m) = 1 \qquad (2.26)$$

- And it is symmetric.

$$\hat{w}(i) = \hat{w}(-i) \text{ for } i=0,1,2 \qquad (2.27)$$

Generating kernel can be chosen as (3.8) where ŵ(m)=ŵ(n)=[1,4,6,4,1]/16 which satisfies conditions given by (3.5), (3.6) and (3.7) as stated by Burt and Adelson [49].

$$w(m,n)=\begin{bmatrix}1&4&6&4&1\\4&16&24&16&4\\6&24&36&24&6\\4&16&24&16&4\\1&4&6&4&1\end{bmatrix}/256 \tag{2.28}$$

When the image is convolved with Gausian Pyramids, it is low-pass filtered and output image is blured [49]. Burt and Adelson compare the speed of Gaussian Pyramids with low-pass filtering in Fourier domain. They state that Gaussian filtering requires less computation than low-pass filtering in Fourier domain and for this reason low-pass filtering an image with Gaussian filters is much faster than that of Fourier domain calculation [49].

2.2.1.1.3 Gaussian Pyramid Interpolation

Burt and Adelson [49] defined a function and named it EXPAND which had a reverse function of REDUCE. While REDUCE shrinks an image half, EXPAND enlarges an image double at each application in one dimension. The expansion is achieved by interpolation of new nodes between existing nodes. After interpolation, the array expands from (M + 1)x(N + 1) to (2M + 1)x(2N + 1). Then

$$I_{l,n}=EXPAND(I_l,n-1) \tag{2.29}$$

Which means, for levels $0<l\le N$, $0\le n$ and nodes i, j: $0\le i<C_{l-n}$, $0\le j<R_{l-n}$,

$$I_l(i,j)=4\sum_{m=-2}^{2}\sum_{n=-2}^{2}w(m,n)I_{l,n-l}\left(\frac{i-m}{2},\frac{j-n}{2}\right) \tag{2.30}$$

Where, w(m,n) is a generating kernel, Gaussian filter and the terms are included only for the integer values of (i-m)/2 and (j-n)/2 in the sum[49].

2.2.1.1.4 Low Pass Filtering with Gaussian Pyramids

If REDUCE and EXPAND functions are applied equal number of times, the low-pass filtered less noisy image with the same size as original image is obtained. Figure 2.8 is the examples of applying REDUCE and EXPAND successively.

Figure 2.8: Gaussian Pyramid REDUCE and EXPAND function result on traffic scene [49]

2.2.1.2 Canny Edge Detector

Canny [4] introduced his edge detection algorithm as his MSc thesis, and it was published in 1986. Canny's edge detector might be the most popular edge detector in literature. It may be caused from the approach of hysteresis threshold. Actually Canny edge detector is not a simple single operator.

2.2.1.2.1 Smoothing

First, the image on which Canny edge detector is to be applied is smoothed by low-pass filter in order to eliminate unnecessary details and noise. This step eases the job of next steps to get edge points. Canny proposes Gaussian filter to smooth image [4]. In this study, Gaussian Pyramids REDUCE (2.21) and EXPAND (2.29) by Burt and

Adelson [49] are applied successively once to smooth images. If the image to be smoothed is defined as $I[i, j]$ and smoothed image as $S[i, j]$, then:

$$S[i, j] = EXPAND(REDUCE(I[i, j])) \tag{2.31}$$

2.2.1.2.2 Gradient Calculation

Canny [4] uses the gradient of the smoothed array $S[i, j]$ to obtain partial derivatives $S_X[i, j]$ in x direction and $S_Y[i, j]$ in y direction as follows:

$$S_X[i, j] \approx (S[i, j+1] - S[i, j] + S[i+1, j+1] - S[i+1, j])/2 \tag{2.32}$$

$$S_Y[i, j] \approx (S[i, j] - S[i+1, j] + S[i, j+1] - S[i+1, j+1])/2 \tag{2.33}$$

The magnitude and orientation of gradient can be calculated as follows [4]:

$$M[i, j] = \sqrt{S_X[i, j]^2 + S_Y[i, j]^2} \tag{2.34}$$

$$\theta[i, j] = \arctan(S_Y[i, j], S_X[i, j]) \tag{2.35}$$

2.2.1.2.3 Non-maximum Suppression

The next step of gradient calculation which is proposed by Canny [4] is called non-maximum suppression. In this step, instead of selecting edge points according to magnitude greater than single threshold, orientation is considered. The maximum likelihood of edge is in the direction of gradient according to orientation as stated by Canny [4]. So, the points perpendicular to the edge direction are discarded or in other word suppressed. The process ending with one pixel wide ridge is called "Non-maximum Suppression" [4].

If non-maximum suppressed image is defined as $N[i, j]$, then:

$$N[i, j] = nms(M[i, j]) \tag{2.36}$$

where $N[i, j]$ points are zero except for the maximum points. At the maximum points, the value of $M[i, j]$ is kept for $N[i, j]$.

2.2.1.2.4 Hysteresis Threshold

Canny [4] states that although significant improvements are realized by non-maximum suppression, there exist still many errors in $N[i, j]$ due to noise, that is, there are many non-edge segments on non-maximum suppressed image. However the magnitude difference of segments is smaller than that of edge points.

At that point Canny [4] proposes a usage of two threshold value instead of one in order to overcome problems of broken edges. Two threshold values, τ_1 and τ_2 are chosen such that τ_2 is several times grater than τ_1.Then two threshold images $T_1[i, j]$ and $T_2[i, j]$ are produced. When the end of the edge contour segment is reached in $T_2[i, j]$, 8 neighbour pixel points are checked in $T_1[i, j]$ and if any edge contour is detected then it is linked to $T_2[i, j]$ edge contour segment.

As a final note, the performance of the detector mostly depends on the threshold values τ_1 and τ_2.

2.2.1.3 Binary Threshold

2.2.1.3.1 Introduction

In this thesis, the video frame processed is either monochrome single channel 8-bit image or 3-channel each for red, green, blue 8-bit image. In order to extract contour information, the image should be binary, which is to be explained in section 3.1.4. The image can be binary either because of being the output of Canny Edge Detector as explained in section 3.1.2 or because of being the output of binary threshold.

2.2.1.3.2 Definition

Define $I[i, j]$ as n-bit grey level image array, $0 \leq i < C$ and $0 \leq j < R$ where C is the number of the columns and R is the number of the rows. In this study, n is equal to 8. Then $I[i, j]$ is 2^8, which is 256 grey level image from 0 to 255 where 0 corresponds to maximum black level while 255 corresponds to maximum white level for monochrome image. Define $I''[i, j]$ as 1-bit image, that is, its element is either 0 or 1. Define T as a threshold value where $0 \leq T \leq 255$.

2.2.1.3.3 Application

The algorithm for binary threshold is straightforward. Any pixel point in $I[i, j]$ with value grater than T is marked as 1, otherwise it is marked as zero in $I''[i, j]$.

2.2.1.4 Contours

2.2.1.4.1 Introduction

The edge map image usually has a binary format to represent borders of objects to be detected. However edge maps usually have redundant information to represent features of object. In order the reduce redundancy and to ease process of edge-map image, contour retrieving algorithms are utilized which are used for storing contours in the chain format. While storing contours in chain format, some polygonal approximations can be done.

2.2.1.4.2 Definition

The output of Canny's edge detection algorithm and binary threshold is binary image which consists of only 0 valued and 1 valued pixels. According to definition stated by Suzuki and Abe [50], the set of connected "0" valued pixels makes 0-component, whereas the set of connected "1" valued pixels makes 1-component. Then there are two connectivity types: the 4-connectivity and 8-connectivity. If (2.37) is applicable then two pixels are defined as 4-connected. (2.37) implies that the connected pixel is neighbor and it is not on diagonal [50].

$$|x_1 - x_2| + |y_1 - y_2| = 1 \tag{2.37}$$

If (2.38) is applicable, then two pixels are defined as 8-connected [50]. (2.38) implies that the connected pixel is neighbor and it can be anywhere in the neighbored [50].

$$\max(|x_1 - x_2|, |y_1 - y_2|) = 1 \tag{2.38}$$

Figure 2.9 shows these relations:

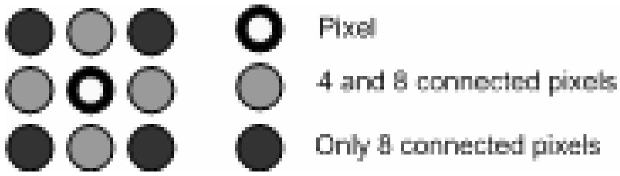


Figure 2.9: Pixel connectivity to neighbor pixels

Using 4 and 8 connectivity, the image can be separated into several non-overlapped 1 valued 4-connected and 0 valued 8-connected components [50]. Each set is composed of either 0 valued or 1 valued pixels, however one group of pixels are linked by a sequence of 4-connection and others are linked by 8-connection [50].

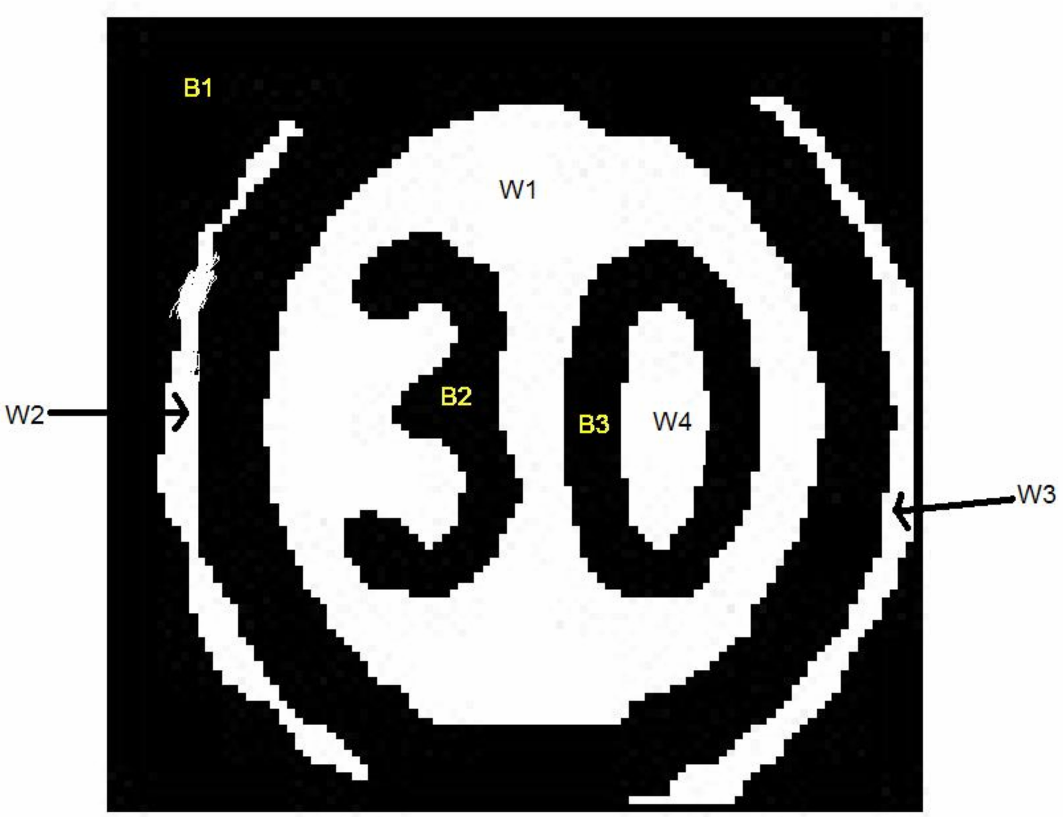


Figure 2.10: contour relationships according to 4- or 8-connected pixels

According to the Figure 2.10, 1-components W1, W2 and W3 are inside 0-component B1. So, W1, W2 and W3 are directly surrounded by B1. 1-component W4 are inside B3, which is inside W1, therefore W4 component is indirectly inside W1 component. W1, W2 and W3 are not surrounded by each other, for this reason they are on the same level. One of the topology should be chosen to prevent a topological contradiction. For this reason either “0” valued pixels must be regarded as 8-connected pixels where “1” valued pixels are dealt with as 4-connected or vice a versa [50]. In this thesis, 8-connectivity is assumed to be used with 1-pixels and 4-connectivity with 0-pixels.

Since 0-components are complementary to 1-components, only 0-components or 1-components are enough to represent whole structure [50]. In this study, for convention, 1-components have been chosen as topological structure to be studied; therefore, 0-pixels make up the background. By this approach, a 0-component directly surrounded by a 1-component is called the hole of the 1-component. The border point of a 1-component can be any pixel which has a 4-connected 0-pixel. A connected set of border points is called the border [50].

Each 1-component has a single outer border that separates it from the surrounding 0-component and zero or more hole borders that separate the 1-component from the 0-components it surrounds [50]. The outer border and hole borders give a full structure of the component and as a result, contour which is the lowered representation of all the borders of all components constitutes a compressed representation of the source binary image [50].

2.2.1.4.3 Representation

Contour may be represented in different coding techniques. In this study, since it is suitable for geometrical approximations to be described in Section 2.2.1.4.5, polygonal representation was preferred. Polygonal representation codes the chain as a sequence of points which represent the vertices of a polyline.

2.2.1.4.4 Contour Retrieving Algorithm

There are four different algorithms to retrieve contours described by Suzuki and Abe [50], however two of them were are utilized in this study:

1. Only the extreme outer contours are found and coded as a sequence of points [50]. The examples of external boundaries are shown in Figure 2.10 as W1, W2, and W3.

2. All contours are found and coded as a sequence of points [50]. The examples of boundaries are shown in Figure 2.10 where the total of 6 such contours exist.

For both of the algorithm, except for some cases, single pass through the image to retrieve contours is made, so it is stated as time efficient by Suzuki and Abe [50].

2.2.1.4.5 Polyline Approximation, Douglas-Peucker Algorithm

In this study, Douglas-Peucker Algorithm [52] was used to simplify contour representation of borders and to achieve geometrical analysis of polygonal shapes, which are rectangle and triangle.

In the Douglas-Peucker (DP) algorithm, according to closeness of a vertex to a line segment, new line segments generated if vertex is further from line more than some threshold value "ε". In this algorithm, from some set of points (vertexes) constituting polyline, fewer set of points (vertexes) constituting approximated polyline is produced with required accuracy:

In this study, the contours retrieved are closed. Two points on the contour are chosen. These initial chosen two points are generally the furthest points (Figure 2.12).

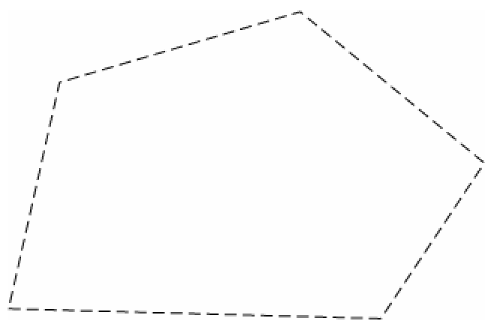


Figure 2.11: Initial polygonal contour representation

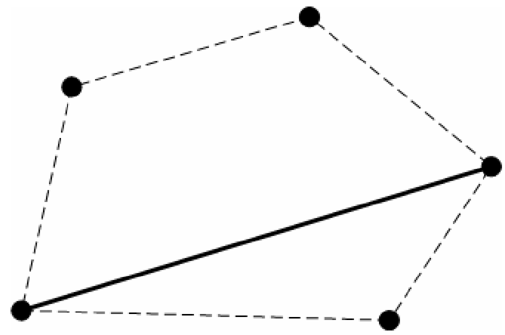


Figure 2.12: Initial two furthest points and line segment connecting them

Then the furthest point from the line is found as shown in Figure 2.13. If the distance between furthest point and line segment is greater than ε (ε > 0), then initial line segment is replaced with two line segments as shown in Figure 2.14.

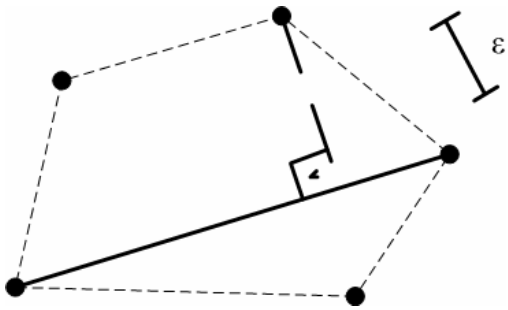


Figure 2.13: Line segment and furthest point from line segment

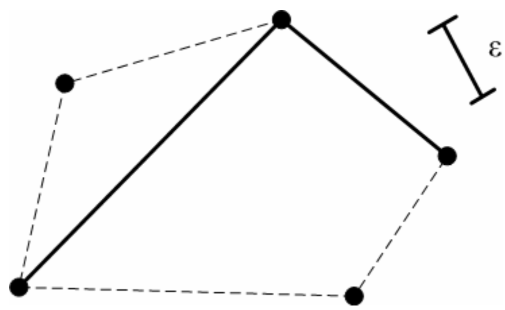


Figure 2.14: Two line segments which is approximated

Same procedure is applied to lower part of the polygonal shape independently. This procedure goes on step by step until there remains no line segment where the points are further than threshold value "ε" to any line segment. After algorithm finishes, the polyline approximation occurs as shown in Figure 2.15.

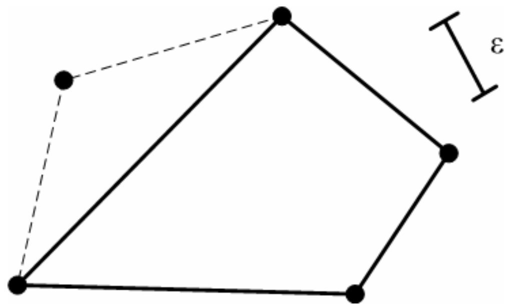


Figure 2.15: Polyline approximation

2.2.1.4.6 Shape Detection with Geometrical Features

Three shapes are to be detected by utilizing geometrical features, they are: Triangle, rectangle and ellipse (which is perspective distortion of circle traffic sign shape).

2.2.1.4.6.1 Detection by Rectangle and Triangle Geometrical Features

After polyline approximation step as described in Section 2.2.1.4.5, the traffic signs are detected by using geometrical features that they have (i.e. triangle and rectangle). The contours are represented only by vertices value after polyline approximation and these points can be investigated as if they are the corner points. If they are not corner values they are going to be discarded.

The corner points should be three and the angle between them should be 60 degrees with some epsilon for triangle traffic signs. Similarly the corner points should be four and the angle between them should be 90 degrees with some epsilon for rectangle traffic signs.

These points are taken in sequence such that c_0, c_1, c_2 are the successor corner points. Then two vectors, $a^1 = c_0 - c_1$ and $a^2 = c_2 - c_1$ calculated. According to formula (3.19), the cosine of angle between these two vectors are calculated.

$$\cos\left(a^1, a^2\right) = \frac{a_x^1 a_x^2 + a_y^1 a_y^2}{\left\|a^1\right\|\left\|a^2\right\|} \tag{2.39}$$

For triangle, $\cos\left(a^1, a^2\right) \approx 1/2$,which corresponds to 60 degrees . For rectangle, $\cos\left(a^1, a^2\right) \approx 0$, which corresponds to 90 degrees. Then for triangle the cosine check is done for successive two corners and for rectangle, this check is done for successive three corners. After checking, passed contours are evaluated as candidate for further processing such as size check, angle check with horizontal.

2.2.1.4.6.2 Detection by Ellipse Geometrical Feature

In order to detect ellipses, since it has no corner, approach in Section 2.2.1.4.6.1 can not be used. Instead, the algorithm proposed by Fitzgibbon et al. [53] can be used to detect ellipses.

The representation of general conic by the second order polynomial is as shown in formula 3.20 [53]:

$$F(\vec{a}, \vec{x}) = \vec{a}^T . \vec{x}^T = ax^2 + bxy + cy^2 + dx + ey + f = 0 \tag{2.40}$$

Where $\vec{a} = \left[a, b, c, d, e, f\right]^T, \vec{x} = \left[x^2, xy, y^2, x, y, 1\right]^T$

Fitzgibbon et al. [53] call $F(\vec{a}, \vec{x})$ as the "algebraic distance between point $\left[(x_0, y_0)\right]$ and conic $F(a, x)$. The sum of squared algebraic distances $\sum_{i=1}^{n} F(\vec{x}_0)^2$ are minimized to achive the fitting of conic by Fitzgibbon et al. [53]. In order to get ellipse-specific fitting polynomial coefficients, (2.41) must be satisfed:

$$b^2 - 4ac < 0 \tag{2.41}$$

Fitzgibbon et al. [53] apply the equality constraint $4ac - b^2 = 1$ on (2.41) and coefficients scaling is incorporated in constraint as they stated.

This constraint can be written as a matrix [53]:

$$\vec{a}^T C \vec{a} = 1 \tag{2.42}$$

After that, the problem could be formulated as minimizing $\left\| D\vec{a} \right\|^2$ with constraint $\vec{a}^T C \vec{a} = 1$, where $D = \left[\vec{x}_1, \vec{x}_2, \vec{x}_3, ..., \vec{x}_n \right]^T$. Then introducing the Lagrange multiplier results in the system $2D^T D \vec{a} - 2\lambda C \vec{a} = 0$ and $\vec{a}^T C \vec{a} = 1$, which can be re-written as follows [53]:

$$\begin{aligned} S\vec{a} &= 2\lambda C \vec{a} \\ \vec{a}^T C \vec{a} &= 1 \end{aligned} \tag{2.44}$$

After the system is solved, ellipse center and axis can be extracted [53].

2.2.2 Detection via Color Feature Extraction

The method which is similar to one proposed by Escalera et al. [35], and Fleyeh [28] was used to extract red color feature of circular traffic signs by using HSV color-space.

$$H_{out} = \begin{cases} 0 & H_{\min} \leq H_{in} \leq H_{\max} \\ 255 & otherwise \end{cases} \tag{2.45}$$

Using the formula (2.25), red colored pixels are masked and all other pixels are left as it is. This function gives an enhancement on red, which improves the performance of detector based on shape feature described in section 2.2.1. However this function is mostly meaningless for the scene taken at night, because the saturation value is too low to decide on hue component, i.e. saturation value below %20.

CHAPTER 3

DETECTION VIA SHAPE AND COLOR FEATURE EXTRACTION

In this chapter, how the traffic signs are localized and detected will be explained.

Mainly there are two different approaches to localize traffic signs in literature as discussed in Chapter 2. One is based on shape feature extraction; other is based on color feature extraction. In this thesis, the main approach to localize traffic signs is shape feature extraction. Color feature extraction is an auxiliary method to localize and detect traffic signs for this study. The reason of choosing shape based localization as main method instead of color is due to two facts:

i. Since the objects to be localized are certain predefined geometrical primitives, it is not very difficult and so costly to make shape feature based localization.
ii. Realizing night time recognition of traffic signs is one of the objectives of this thesis. Although traffic signs have certain colors i.e. red, blue, yellow, orange, for most of the traffic scene, the illumination and saturation values are very low and for this reason it is not possible to localize most of the traffic sign in seen by color segmentation at night time.

Although simulations are mainly based on shape feature extraction, in order to detect circular traffic signs (actually ellipse due to perspective distortion) in day time, the hue component for red color segmentation is used as an auxiliary method described in section 3.2. At night time color based segmentation is almost meaningless for most of the traffic image scene.

3.1 Shape Based Detection

For all of the shapes, experiment is carried out by first extraction of edges. Initially the color image acquired from video stream is separated into 8-bit color channels of red, green and blue as shown in Figure 3.1:

Figure 3.1: Blue, green and red channels of RGB image

At the first step, Gaussian pyramid [49] is applied as explained in Section 2.2.1.1.4 for low-pass filtering of all 3 channels of image.

Then Canny edge detection algorithm [4] as described in section 2.2.1.2 is applied to all three color-channel images. At the same level, binary threshold as described in section 2.2.1.3 is applied to all three channels for N different threshold values where N to be experimented as explained in Chapter 5. Moreover for day-time images and for night time images different upper and lower limits can be chosen. The reason of using different threshold values is caused from the need of detection over wide dynamic ranges of intensity. After this step all the output images are binary.

Contour retrieving algorithm proposed by Suzuki and Abe [50] as explained in Section 2.2.1.4.4 is applied and contours that images include are found. For example, Canny [4] and ten binary threshold level images with their contour information are obtained for each color channel as shown in Figure 3.2:

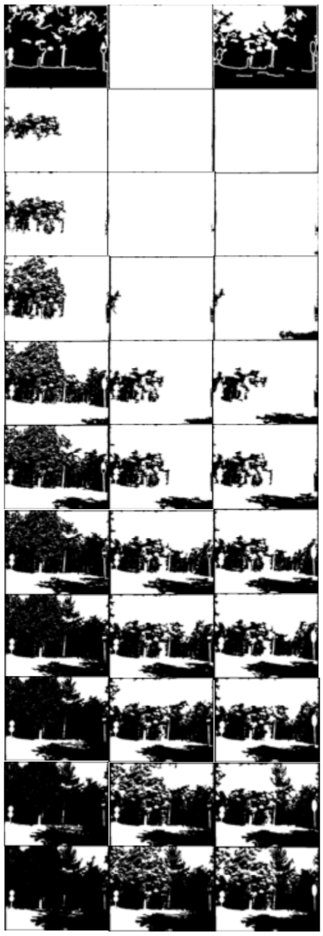


Figure 3.2: Binary image outputs of Canny and binary threshold, first column corresponds to blue component, second one corresponds to green component and last one corresponds to red component and first row corresponds to canny output, second row to last row correspond to 1^{st} threshold to 10^{th} threshold.

After finding the contours different algorithms are carried out to detect different shaped traffic signs.

3.1.1 Detection of Triangular Objects

In order to detect triangular traffic signs, polynomial approximation namely Douglas-Peucker algorithm [52] is applied as explained in Section 2.2.1.4.5. The important parameter for Douglas-Peucker algorithm is the threshold value "ε". The threshold value is chosen considering noise of border values i.e. contours. If the threshold value gets bigger then more polylines are candidate to be erased.

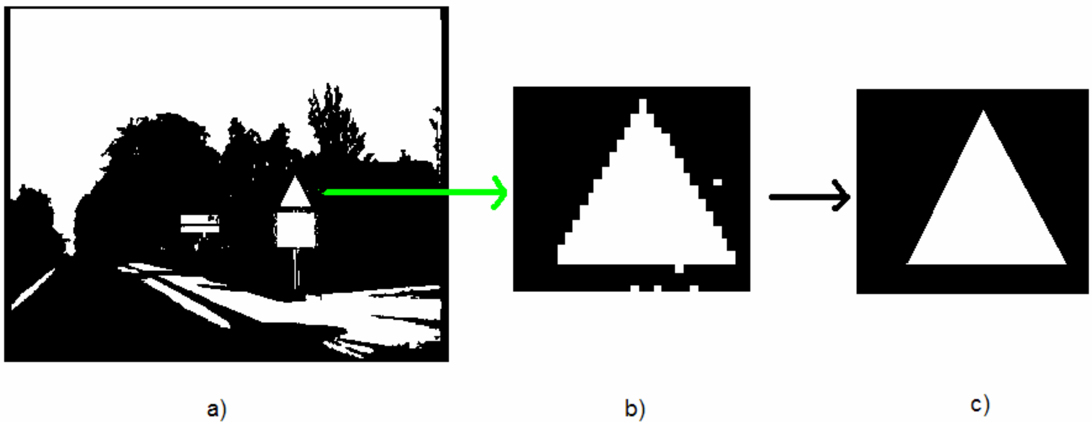


Figure 3.3: a) Level-9 binary threshold contour representation, b) enlarged view of triangular object which is to be checked as a candidate of traffic sign, c) illustration of polyline approximation output by Douglas-Peucker algorithm [52]

After polynomial approximation, the resultant contours are expected to be three vertex points if they correspond to triangle. If so, the points are taken in sequence such that c_0, c_1, c_2 are the successor vertex points and then two vectors, $a^1 = c_0 - c_1$ and $a^2 = c_2 - c_1$ are calculated as explained in Section 2.2.1.4.6.1 and as shown in Figure 3.4.

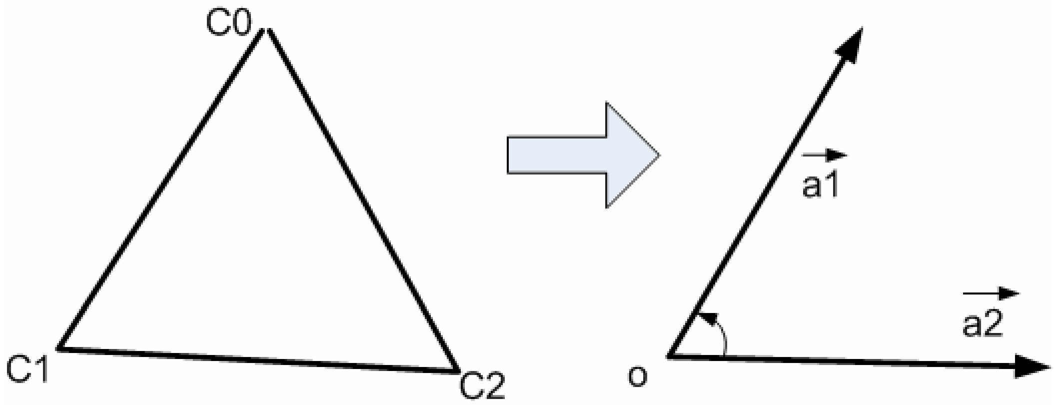


Figure 3.4: Vertex points and vector representation for triangle sign candidates

Then according to formula 2.19, the cosine of angle between these two vectors are calculated. For triangle, $\cos\left(a^1, a^2\right) \approx 0.5$, which corresponds to 60 degrees. If the $\cos\left(a^1, a^2\right)$ is between 0.3 and 0.7 then the vertex point of c_1 is accepted as correct triangle vertex. The same procedure is applied for c_1, c_2, c_0 successor vertex points. If c_2 is accepted as correct triangle vertex as c_1 then the contour is accepted as triangle.

Then the angle between one line segment of triangle and horizontal of image is checked. It is assumed that the angle between any one of line segment of triangle and horizontal is between 48 and 72 degrees or 0 and 12 degrees. If not, the contour is discarded.

Then last check is performed for the size of the triangle candidate. If it is smaller than 18 pixels or greater than 96 pixels in any of the horizontal or vertical dimensions then the contour is discarded.

If the contour passes all the steps above, then it is validated as triangle sign candidate and the coordinates values of vertex points are passed to the recognition module.

There are some example images in which some triangles are detected shown in Figure 3.5, and in Figure 3.6:

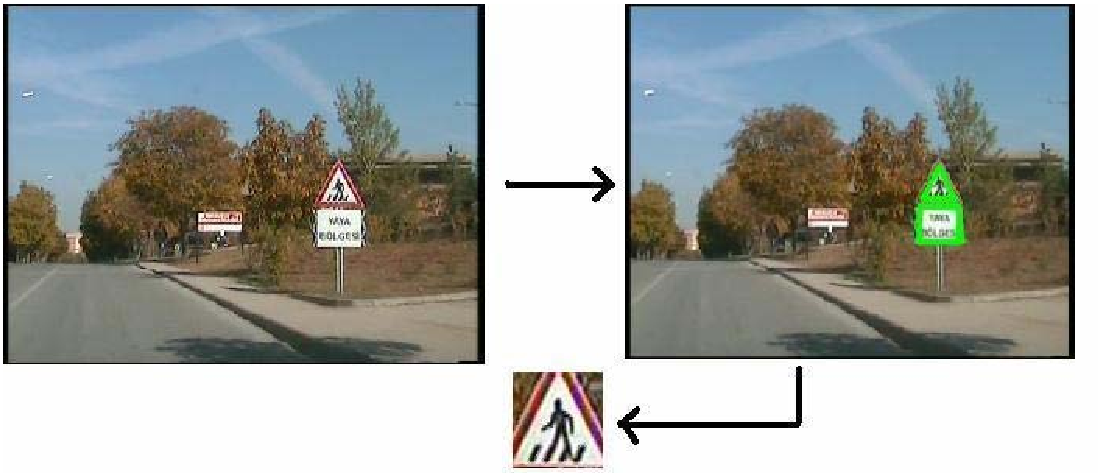


Figure 3.5: Traffic scene with a rectangular, triangular signs with detected and cropped triangle sign taken in afternoon

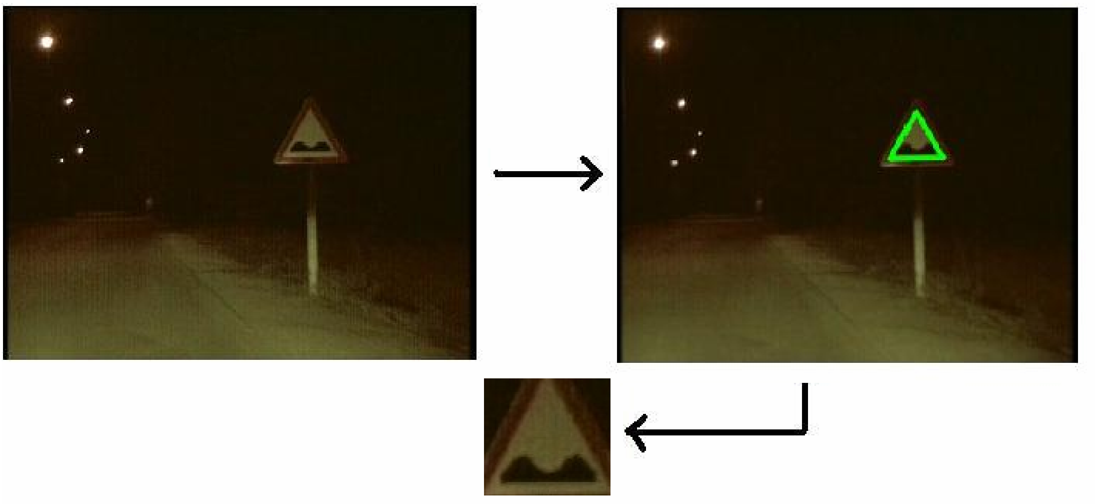


Figure 3.6: Traffic scene with a triangular sign with detected and cropped triangle sign taken at night

Figure 3.7: Some examples of detected and cropped triangular traffic signs

3.1.2 Detection of Rectangular Objects

Detection method of rectangle objects is similar to detection of triangle objects. To detect rectangular traffic signs, polynomial approximation namely Douglas-Peucker algorithm [52] is applied as realized for triangular signs. The threshold value "ε" can be set for different test values for each experiment.

After polynomial approximation, the contours are expected to be four vertex points if they correspond to rectangle. If so, the points are taken in sequence such that c_0, c_1, c_2, c_4 are the successor vertex points and then two vectors, $a^1 = c_0 - c_1$ and $a^2 = c_2 - c_1$ are calculated as explained in Section 2.2.1.4.6.1 and as shown in Figure 3.8.

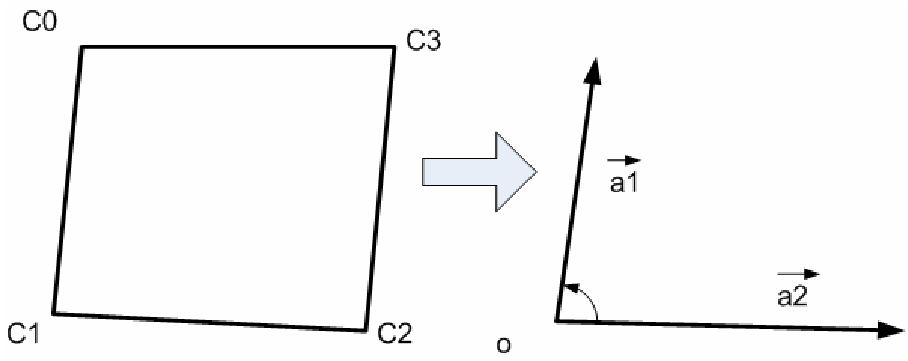


Figure 3.8: Vertex points and vector representation for rectangle sign candidates

Then according to formula 2.19, the cosine of angle between these two vectors are calculated. For rectangle, $\cos(a^1, a^2) \approx 0$ constraint corresponds to 90 degrees. If the $\cos(a^1, a^2)$ is smaller than 0.2 then the vertex point of c_1 is accepted as true rectangle vertex. The same procedure is applied for c_1, c_2, c_3 and c_2, c_3, c_0 successor vertex points. If both c_2 and c_3 are accepted as correct rectangle vertex as c_1 then the contour is accepted as triangle.

Then the angle between one line segment of triangle and horizontal of image is checked. It is assumed that the angle between any one of line segment of rectangle and horizontal is between 0 and 8 degrees or between 82 and 98 degrees. If not, the contour is discarded.

If the contour passes all the steps above, then it is validated as rectangle sign candidate and the coordinates values of vertex points are passed to recognition module.

Figure 3.9 and Figure 3.10 illustrate the detected rectangular traffic signs:

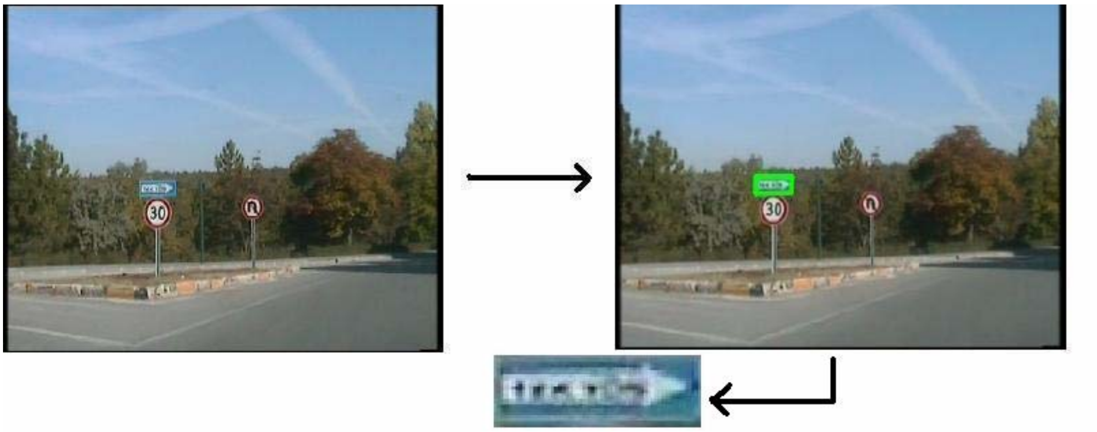


Figure 3.9: Detected rectangular sign taken in afternoon with cropped rectangular sign

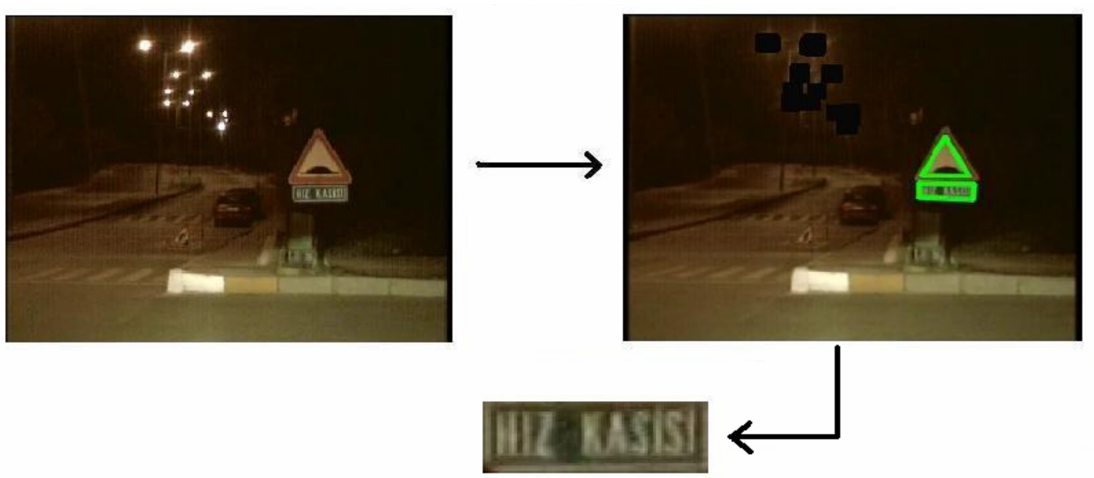


Figure 3.10: Detected rectangular and triangular signs taken at night with cropped rectangular sign

Figure 3.11 illustrates some examples of detected and cropped rectangular signs:

Figure 3.11: Some examples of detected and cropped rectangular traffic signs

Figure 3.12 illustrates some false positives in detection of rectangular traffic signs

Figure 3.12: Some examples of false positives in detection of rectangular traffic signs

3.1.3 Detection of Elliptical Objects

Circular traffic signs usually appear in the seen as elliptical objects due to perspective distortion. In order to detect them, the method named as ellipse fitting proposed by Fitzgibbon et al. [53] is used as explained in Section 2.2.1.4.6.2. The contour points should be at least six in order to solve six unknown linear system. First of all the contour is checked whether it contains more than or equal to six points. If so, ellipse fitting algorithm is applied [53] together with contour filled correlation to ensure founded ellipses by ellipse fitting algorithm really a close-contour. By eliminating non-closed contour curves, number of false positive detected ellipses is reduced.

Then the size of elliptical object is checked. If it is smaller than 18 pixels or greater than 96 pixels in any dimension then the contour again discarded. The elliptical box parameters are passed if it passes the check.

There are some example images in which some triangles are detected shown in Figure 3.13 and 3.14:

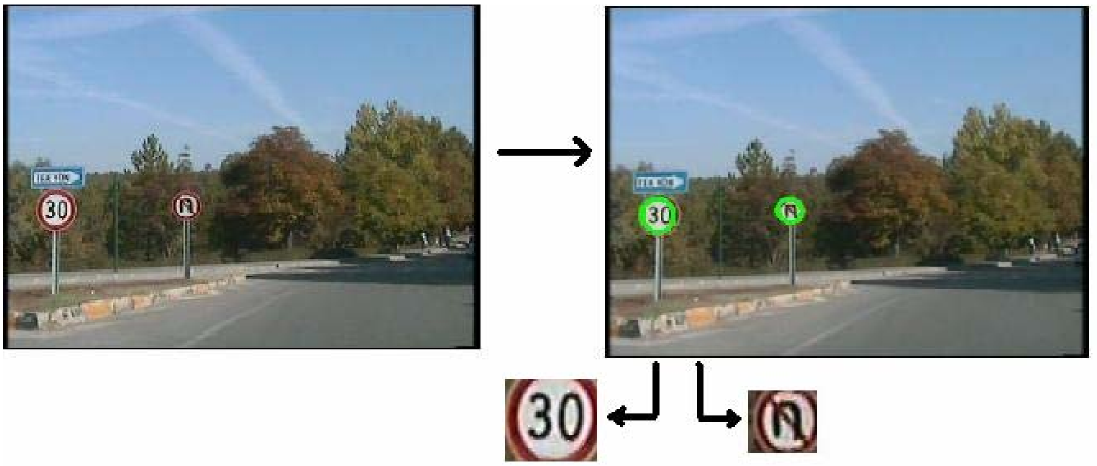


Figure 3.13: Traffic scene with circular signs taken in a sunny day with detected and cropped circular signs

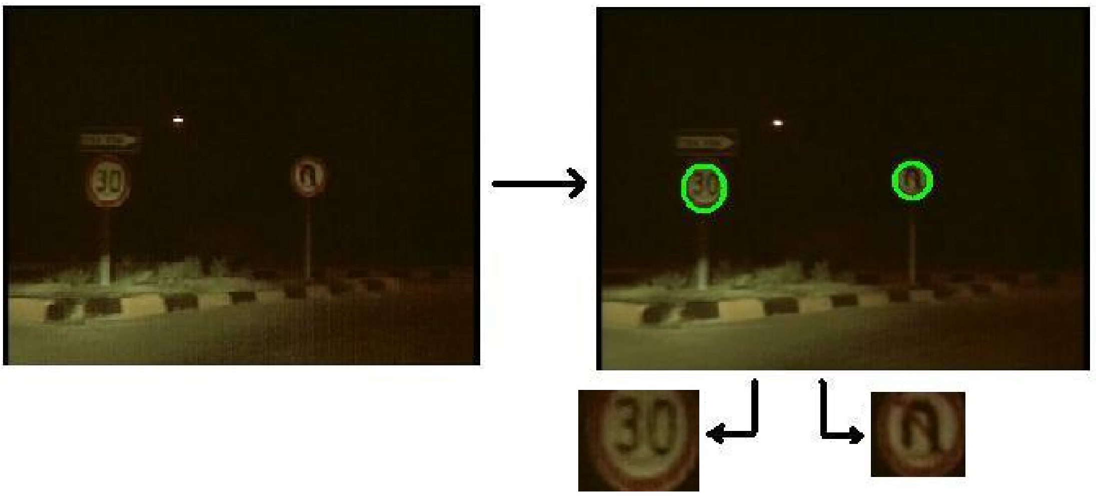


Figure 3.14: Traffic scene with circular signs taken at night with detected and cropped circular signs

Figure 3.15: Some examples of detected and cropped circular traffic signs

Figure 3.16: Some examples of false positives in detection of circular traffic signs

3.2 Detection via Color Feature

While detecting elliptical objects, in order to help ellipse-fitter [53], color feature is utilized as described in Section 2.2.2. Using the formula 2.45, red colored pixels are emphasized and all other pixels are left as it is. The hue chart is mapped between the values of 0 and 180 degrees instead of the values of 0 and 360 degrees. Red color value corresponds to degrees of between 160 and 180, and degrees of 0 and 10 as shown in Figure 3.17. By the way, in the formula 3.25, the threshold values H_{min} and H_{max} are piecewise defined for red color.

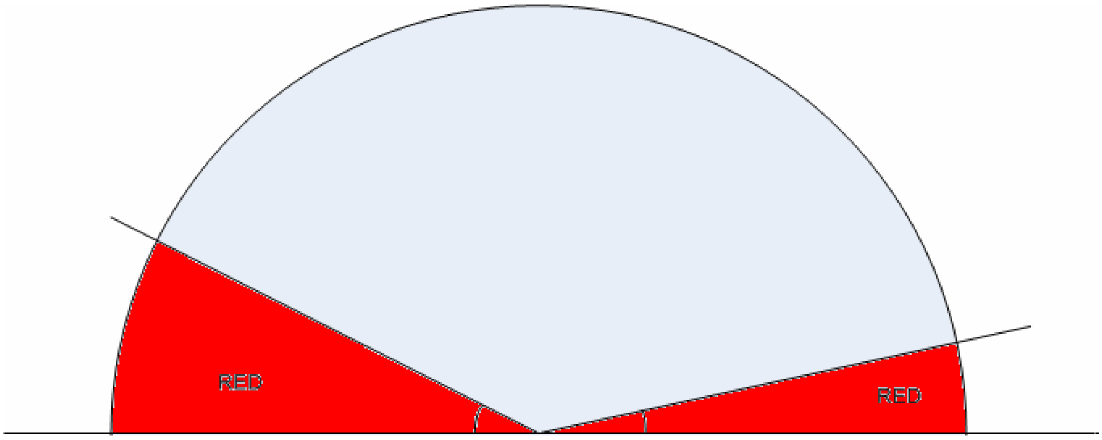


Figure 3.17: Hue chart mapped between 0 to 180 degrees

The formula 2.45 gives an enhancement on red, which improves the performance of detector based on shape feature described in section 3.1. In Figure 3.17, a traffic scene is seen on left and red color-marked image with green by using formula 3.25 is seen as green on right. The color feature extraction for red on the frame shown in Figure 3.18 enhances shape feature extraction, where the input image to shape feature extractor is color enhanced image instead of original image. The other steps are the same for both of the color emphasized and color-non emphasized image frames.

Figure 3.18: A traffic scene on left and red-color emphasized image on right

However this function is mostly meaningless for the scene taken at night, because the saturation value is too low to decide on hue component, i.e. saturation value is below %20.

Figure 3.19: A traffic scene on left and red-color emphasized image on right

As seen in Figure 3.19 which is one of the frames of video taken at night, there is almost no emphasizing on color red. So, color feature extraction does not contribute detection process for dark i.e. low illuminated and low saturated scenes.

CHAPTER 4

RECOGNITION OF TRAFFIC SIGNS

4.1 Introduction

In order to recognize traffic signs, template matching based methods can be utilized. Since the informative part of traffic signs is mainly composed of ideograms, recognition process can be performed simply calculating mean square error between the template ideogram and the candidate traffic sign to be recognized. However the method is very sensitive to the place of the ideogram in frame. Moreover illumination changes over wide range. For this reason preprocessing is important and essential for recognition. After preprocessing, search window in a certain frame can be utilized to improve recognition process.

4.2 Preprocessing

Preprocessing is the essential step before recognition because the candidate acquired from detection module is rotated; perspective-distorted and needs to be resized to match with template. It is a fortune that vertex points of triangular and rectangular sign candidates are known parameters from detection module. However, since ellipse does not have any vertex point, and elliptical box mentioned in Section 3.1.3 can not give exact coordinates, the transformations such as rotation and resizing of ellipses do not provide perfect preprocessing.

4.2.1 Image Resizing with Respect to Corner Points

The output of the detection module is the vertex points of polygonal shapes of triangle, rectangle and rectangular box for the ellipse as shown in Figure 4.1.

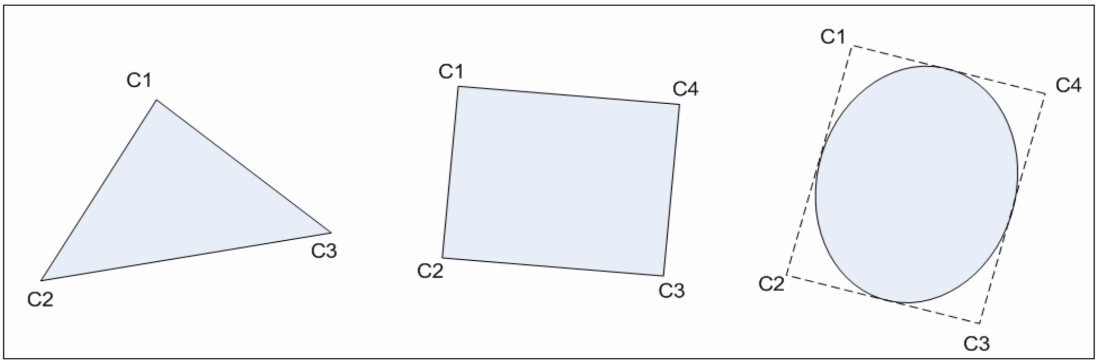


Figure 4.1: The output of the detection module as vertex points

In general, detected traffic sign is perspective-distorted and rotated. Moreover it should be resized to be equal to template image in size to realize template matching. Since the vertex points of triangular and rectangular traffic sign candidate are known parameters, it is possible to translate the traffic sign candidate into template size as shown in Figure 4.2:

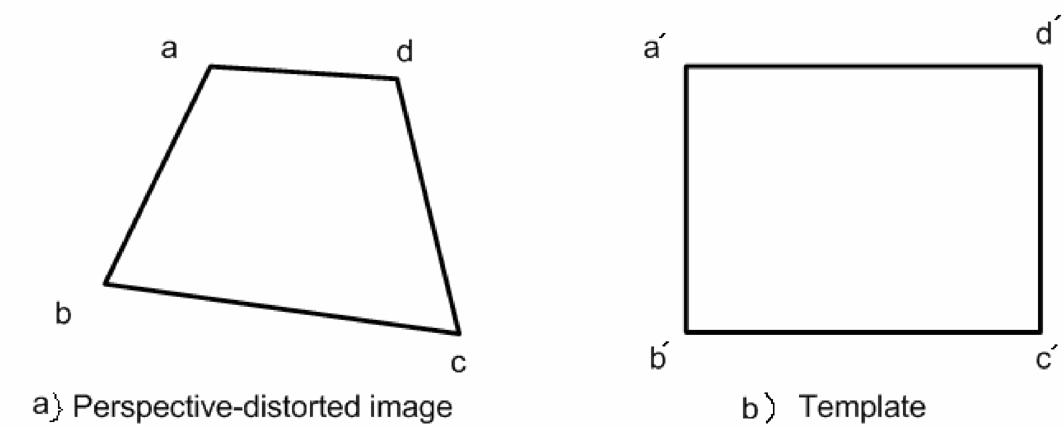


Figure 4.2: (a) Perspective-distorted object, **(b)** transformed counterpart

Transformation matrix can be written as follows:

$$X^{l} = TX$$

$$\begin{pmatrix} wx^{l} \\ wy^{l} \\ w \end{pmatrix} = \begin{bmatrix} t_{11} & t_{12} & t_{13} \\ t_{21} & t_{22} & t_{23} \\ t_{31} & t_{32} & t_{33} \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \tag{4.1}$$

Where *x, y* is the coordinate points of perspective-distorted image and *x', y'* is the coordinate points of template.

The entity t_{33} can be set as 1, because it only affects scaling. Then the T matrix can be solved for 8 unknown entities. For this reason, at least 4 linearly independent point pairs are sufficient to solve T matrix. Then for the given point pairs $(x_1, y_1) - (x_1^l, y_1^l)$, $(x_2, y_2) - (x_2^l, y_2^l)$, $(x_3, y_3) - (x_3^l, y_3^l)$, $(x_4, y_4) - (x_4^l, y_4^l)$, the entities of the T matrix can be calculated by solving the homogeneous system:

$$\begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -x_1x_1^l & -y_1x_1^l & -x_1^l \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -x_1y_1^l & -y_1y_1^l & -y_1^l \\ x_2 & y_2 & 1 & 0 & 0 & 0 & -x_2x_2^l & -y_2x_2^l & -x_2^l \\ 0 & 0 & 0 & x_2 & y_2 & 1 & -x_2y_2^l & -y_2y_2^l & -y_2^l \\ x_3 & y_3 & 1 & 0 & 0 & 0 & -x_3x_3^l & -y_3x_3^l & -x_3^l \\ 0 & 0 & 0 & x_3 & y_3 & 1 & -x_3y_3^l & -y_3y_3^l & -y_3^l \\ x_4 & y_4 & 1 & 0 & 0 & 0 & -x_4x_4^l & -y_4x_4^l & -x_4^l \\ 0 & 0 & 0 & x_4 & y_4 & 1 & -x_4y_4^l & -y_4y_4^l & -y_4^l \end{bmatrix} \begin{pmatrix} t_{11} \\ t_{12} \\ t_{13} \\ t_{21} \\ t_{22} \\ t_{23} \\ t_{31} \\ t_{32} \\ 1 \end{pmatrix} = 0 \tag{4.2}$$

Once the transformation matrix is calculated, it can be used to transform arbitrary size triangular, rectangular and circular sign candidates to certain size 2-D image plane. Triangular and rectangular sign candidates are transformed as 104x120 size images, whereas circular sign candidates are transformed as 120x120 size images in this study. The examples of them are shown in Figure 4.3 for triangular sings, in Figure 4.4 for rectangular signs, and in Figure 4.5 for circular signs:

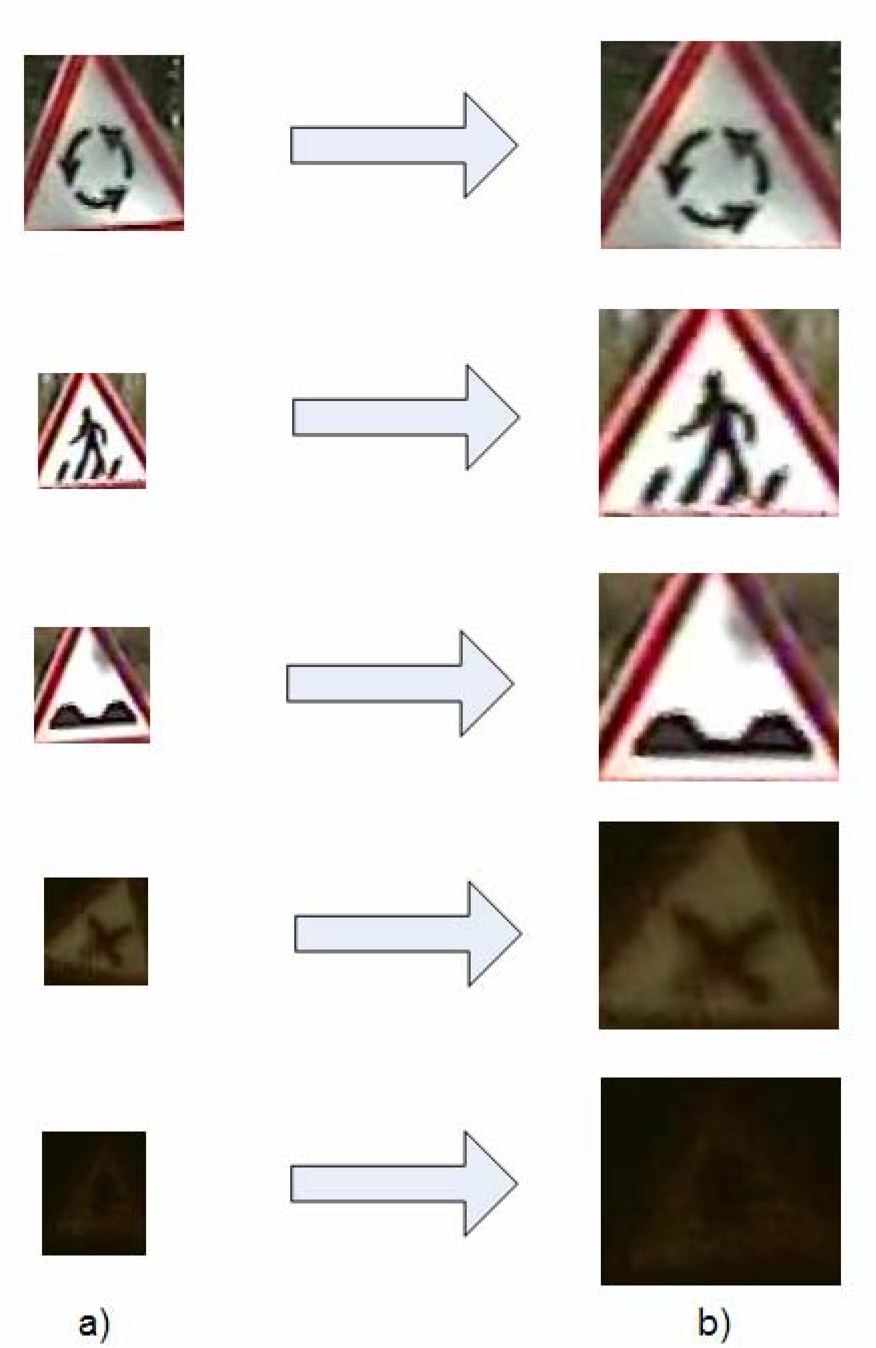


Figure 4.3: (a) Examples of triangular detected and cropped signs, (b) transformed triangular detected signs

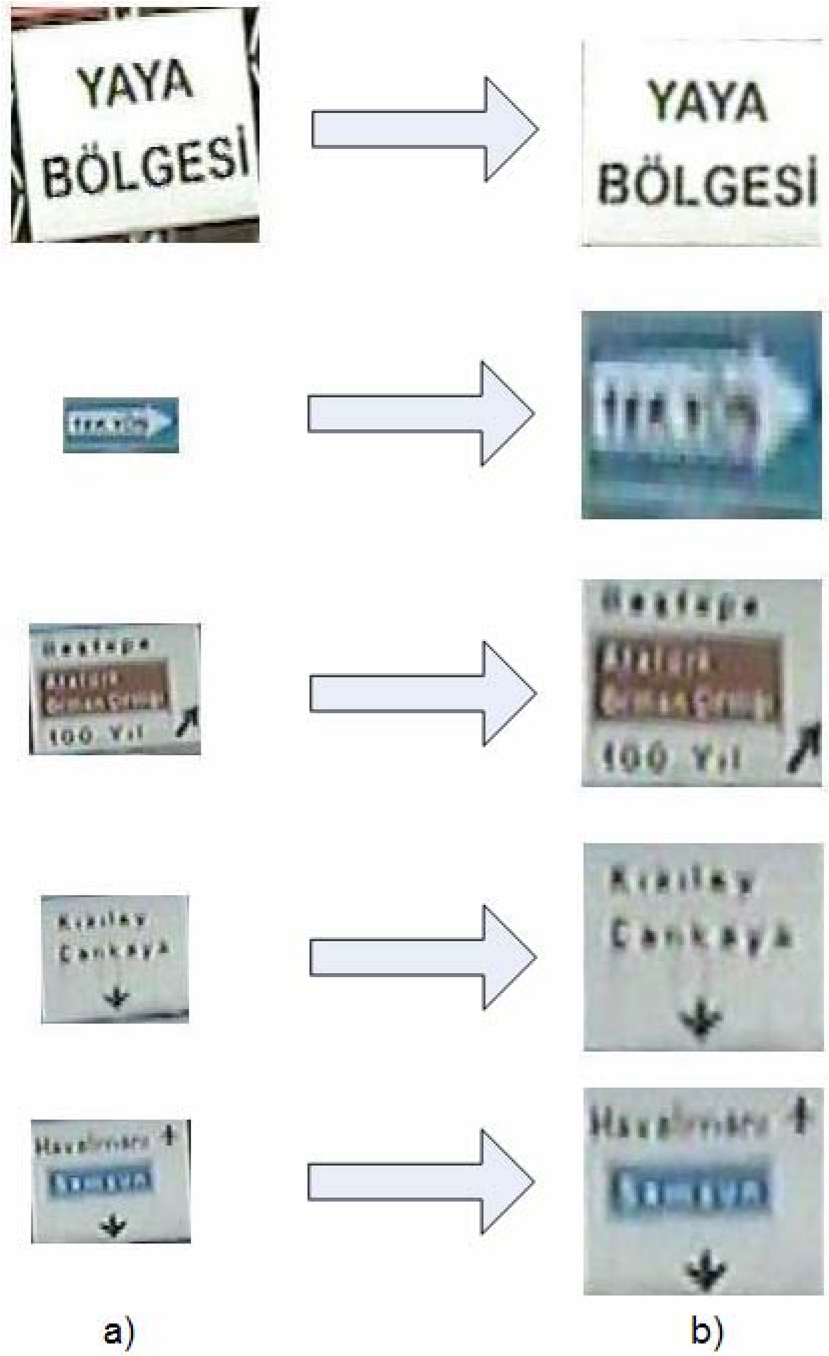


Figure 4.4: (a) Examples of rectangular detected and cropped signs, (b) transformed rectangular detected signs

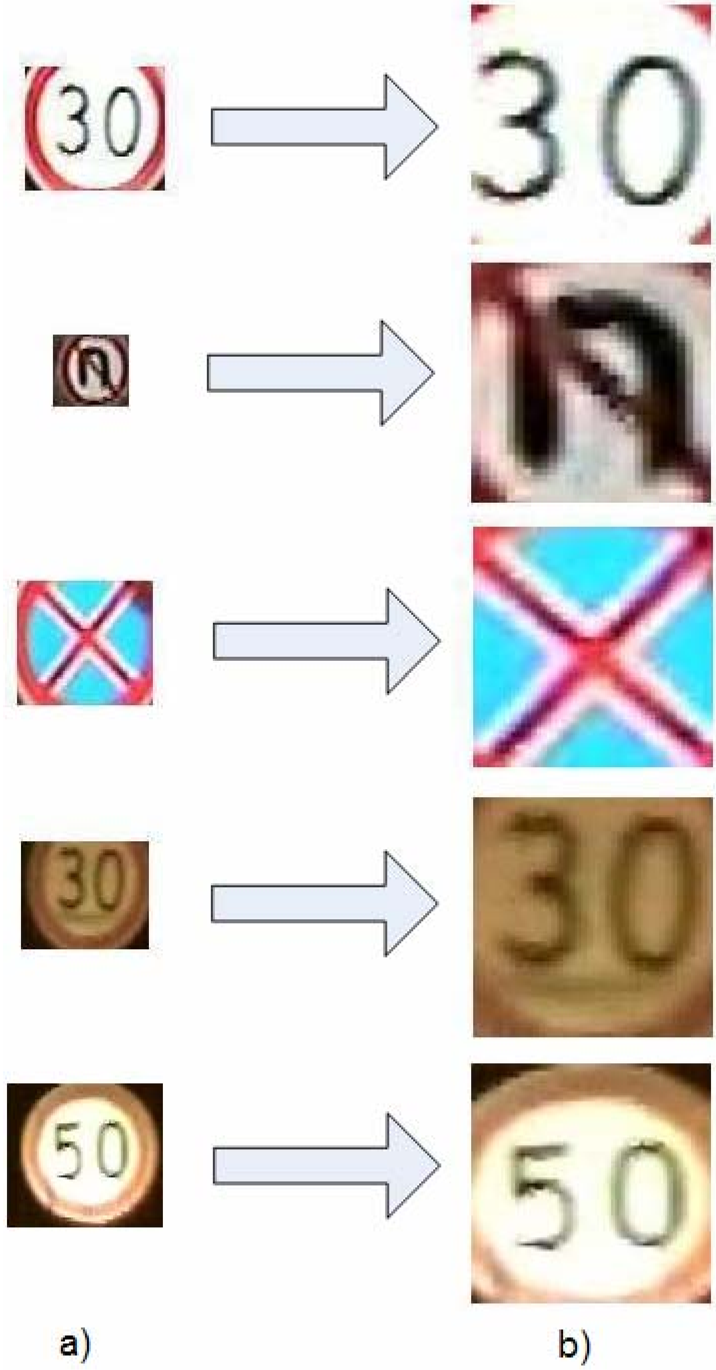


Figure 4.5: (a) Examples of circular detected and cropped signs, (b) transformed circular detected signs

Size transformation is realized pixel by pixel as given:

$$\begin{pmatrix} wx^{l} \\ wy^{l} \\ w \end{pmatrix} = \begin{bmatrix} t_{11} & t_{12} & t_{13} \\ t_{21} & t_{22} & t_{23} \\ t_{31} & t_{32} & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \tag{4.3}$$

(x,y) is a coordinate for any pixel of image which is the output of size transformation, and (x',y') is a coordinate for any pixel of image which is to be transformed. Then the pixel value at a coordinate of (x',y') is copied to the pixel which has a coordinate of (x,y). This process is performed for all of the pixels of output image with Cartesian coordinate system of x and y. Then "REDUCE" process is applied as defined in Section 2.2.1.1.2 [49] onto size-transformed image in order to reduce some artifacts such as aliasing.

4.2.2 Histogram Equalization

The detected traffic sign candidates are generally under wide variation of illumination. They can be illuminated by bright sunlight, they can be illuminated by ongoing light of car in a dark traffic scene, or they can be illuminated by a very weak light source. Wide variation of illumination adversely affects the performance of recognition module based on template matching. For this reason, histogram equalization is to be realized in order to reduce adverse effects of wide variation of illumination.

Before applying histogram equalization; detected and size-transformed traffic sign candidate image in RGB color space is transformed into another color space of HSV. Then the V component is taken which corresponds to monochrome gray-scale counterpart of color one as shown in Figure 4.6.

Figure 4.6: (a) Detected and cropped traffic sign candidate, (b) transformed, RGB color image, (c) V channel of HSV equivalent transformed image

After monochrome equivalent of color RGB traffic sign candidate image is obtained, its histogram is calculated. Then, histogram equalization is carried out:

STEP 1: Two extreme histogram bins which are lowest and highest gray-level bins are deleted. This step reduces the sensitivity to extreme illumination value.
STEP 2: The histogram is equalized by using algorithm [55] in C code described as follow:

```
while ((j<256)&&(i<256)) {
 while ((cumsum[i]<ideal[j]+1)&&(i<256)) {
 map[i]= j;
 i++;
 }
 j++;
 }
```

Where “cumsum” corresponds to histogram of image to be applied for histogram equalization, “ideal” is a histogram of equally spread bins, and “map” is the aimed transform function which is used to transform original image with small dynamic range i.e. low contrast into image with wider dynamic range. Then the image transformed by function “map”:

$$hist\_eq\_img[x,y] = map(img[x,y]) \tag{4.4}$$

Where img[x,y] corresponds to image intensity value , hist_eq_img[x,y] corresponds to image intensity value of histogram equalized image.

Example images for histogram equalization for triangular traffic sign candidate are shown in Figure 4.7, and their histogram graphs are shown in Figure 4.8.

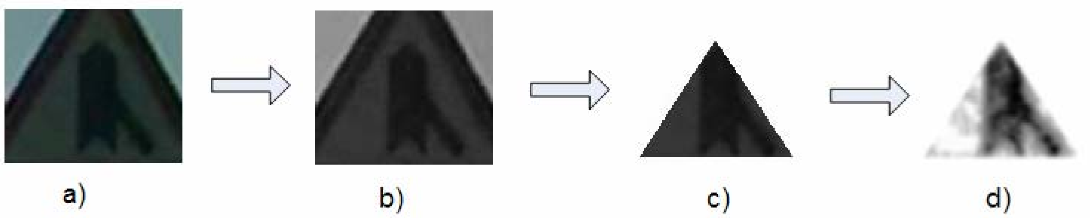


Figure 4.7: (a) Transformed, RGB, traffic sign candidate image, (b) V channel of HSV equivalent transformed image, (c) masked image where the region around traffic sign is cut out, (d) traffic sign candidate after histogram equalization

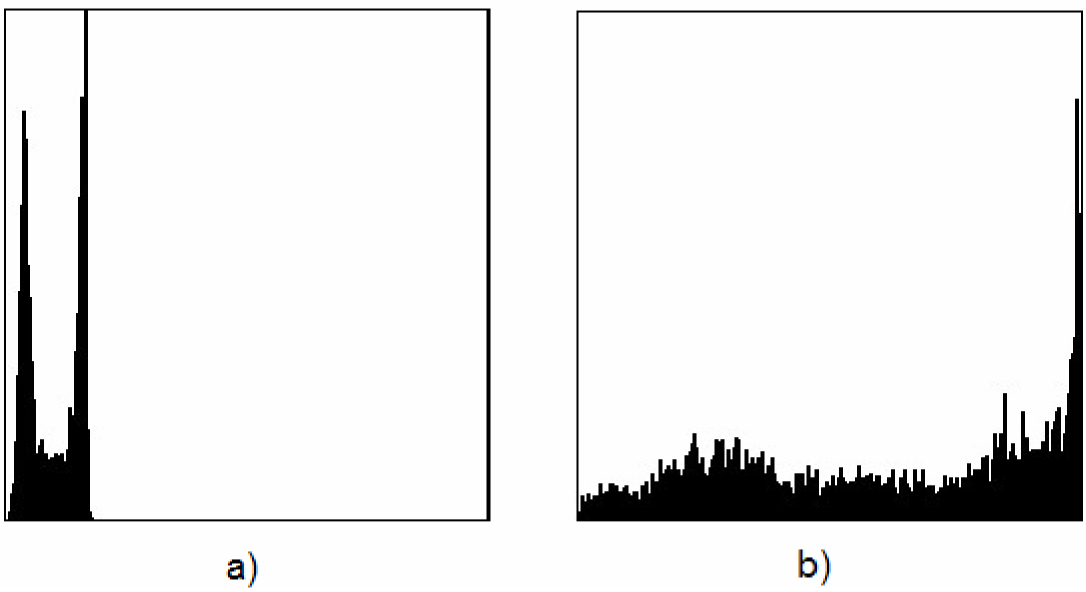


Figure 4.8: (a) Histogram graph of traffic sign candidate image in Figure 4.7-c before applying histogram equalization, (b) histogram graph of traffic sign candidate image in Figure 4.7-d after applying histogram equalization

4.2.3 Binary Threshold

The output of histogram equalization from Section 4.2.2 is 8-bit gray-level monochrome traffic sign candidate image. In order to prepare this image for template matching with minimum square error, binary threshold function is applied as explained in Section 3.1.3. The threshold value is optimized as 95 in between [0,255] gray levels. The examples of binary threshold are shown in Figure 4.9, and in Figure 4.10.

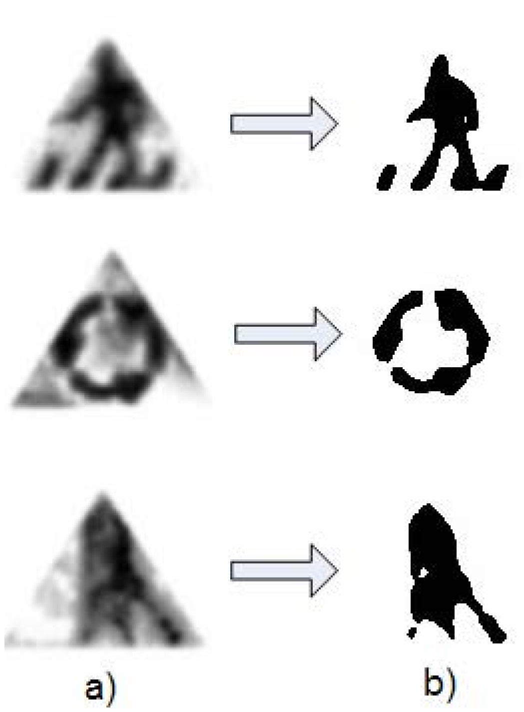


Figure 4.9: (a) Examples of triangular traffic sign candidates after histogram equalization, (b) ideograms extracted after binary threshold

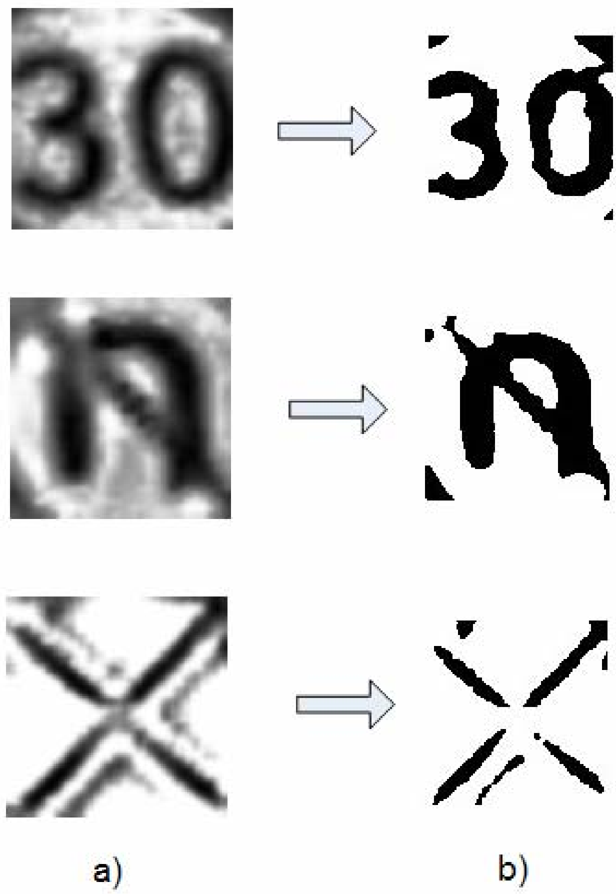


Figure 4.10: (a) Examples of circular traffic sign candidates after histogram equalization, (b) ideograms extracted after shrinking and binary threshold

4.3 The Nearest Neighbor Rule

The Nearest Neighbor Rule is a statistical pattern recognition method. It is a nonparametric method such that the form of density function is unknown. That is, complete distribution of observation **x** (representative vectors of training samples) is not given [56]. Instead, correctly pre-classified number of samples and their classes, ($\mathbf{x_i}$, $\mathbf{\Theta_j}$) are given as a set of pairs. Here, $\mathbf{x_i}$ is correctly pre-classified sample and $\mathbf{\Theta_j}$ is a class where $\mathbf{x_i}$ belongs to. Then set of pre-classified samples can be written as follows [56]:

$$D^n = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, ..., \mathbf{x}_n\} \tag{4.5}$$

In the Nearest Neighbor Rule, the nearest observation **x'** (where **x'** $\in D^n$**)** to test vector **x** according to some metrics is chosen. If more observations (k nearest observations) are chosen and labeling is realized taking a vote, this method is called k-Nearest-Neighbor Rule [56]. In the Nearest Neighbor Rule, after deciding nearest sample **x'**, **x** is labeled with category **Θ_j** which is the category class of nearest neighbor **x'**. The Nearest Neighbor Rule can be considered as "suboptimal" [56]. Because error rate is greater than the most optimal Bayes rate however given the infinite number of samples, error rate is not greater than twice of the Bayes rate [56].

$$P^* \le P \le 2P^* \tag{4.6}$$

Where P^* is Bayes error rate and P is the error rate by Nearest Neighbor Rule [56].

Some metrics can be utilized to carry out the process of deciding nearest sample. Some possible "distance functions" [56] are Euclidean metric, Manhattan distance and Tanimato metric. Euclidean metric is the second order of Minkowski metric (formula 4.7) and Manhattan distance is the first order of Minkowski metric [56].

$$L_k(\mathbf{a},\mathbf{b}) = \left(\sum_{i=1}^{d} |a_i - b_i|^k \right)^{1/k} \tag{4.7}$$

In this study, Manhattan distance was used as a metric to decide nearest sample to test vector. For Manhattan distance, k is set as 1.

4.4 Template Matching within Search Window

After preprocessing as described in Section 4.2, the image of traffic sign candidate is to be compared with templates in database to perform recognition process. This process is carried out as follow:

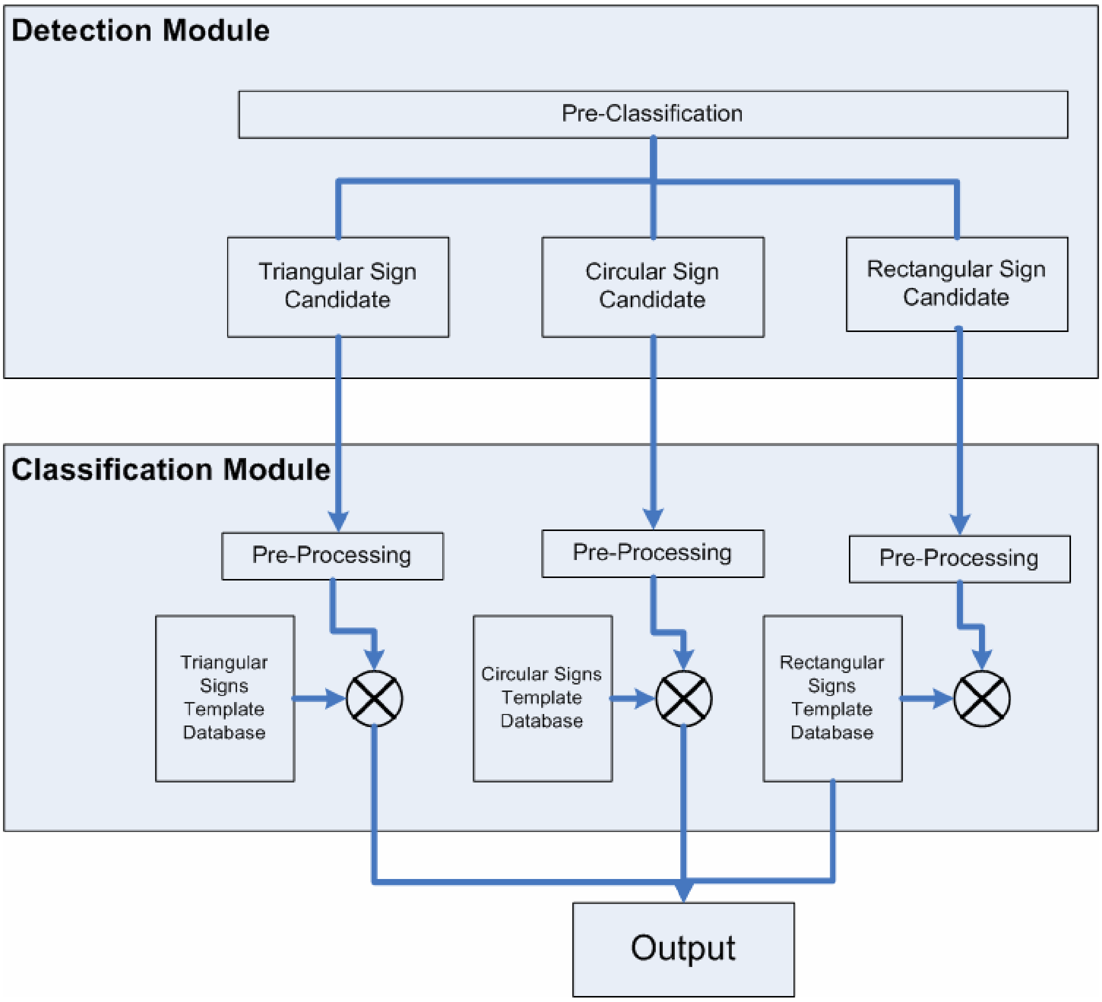


Figure 4.11: Simplified block diagram of recognition scheme

While the traffic sign candidates are detected according to their shapes, pre-classification is realized because the shape information of detected candidate sign is obtained in detection stage at the same time. Then detected and pre-classified traffic sign candidate images are fed into classification module. After preprocessing, size-normalized, histogram-equalized traffic sign candidate image is ready for recognition process.

4.4.1 Normalized Cross Correlation

Cross correlation can be realized in order to measure the similarity of two images where one image is a template whereas other is a test image. A shifting window is utilized to search maximum possible match. However while shifting search window and comparing cross correlations, similarity measurement usually fail [57]. Because image cross-correlated with some bright image give a higher similarity than cross-correlation with itself [57]. In order to solve this problem, normalized cross correlation can be utilized [57]. In normalized cross correlation, cross correlation is normalized with average brightness of template image and test image within search window.

$$\gamma(u,v)=\frac{\sum_{X,Y}\left[f(x,y)-\bar{f}_{U,V}\right]\left[t(x-u,y-v)-\bar{t}\right]}{\left\{\sum_{X,Y}\left[f(x,y)-\bar{f}_{U,V}\right]^2\sum_{X,Y}\left[t(x-u,y-v)-\bar{t}\right]^2\right\}^{0.5}} \tag{4.8}$$

Where $\gamma(u,v)$ is correlation coefficient for each shift in search window, $t(x,y)$ is the test image, $f(x,y)$ is the template image, $\bar{t}$ is the mean of test image, $\bar{f}_{U,V}$ is the mean of the $f(x,y)$.

4.4.2 Recognition Process of Triangular Traffic Signs

The input image such as Figure 4.12 is to be compared with triangular template images shown in Figure 4.13 one by one for recognition.

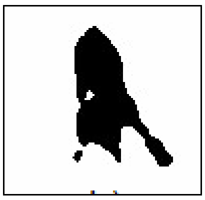


Figure 4.12: Preprocessed triangular sign candidate to be recognized

Figure 4.13: Examples of template images in triangular template database.

Both of the triangular candidate image and template images are the same size such that they are composed of 104 rows and 120 columns. The main approach to recognize them is calculating Manhattan distance between two images. The Algorithm runs as follow:

```
GLOBAL_ERROR = 65535
for i=1 TO N
if MSE( TEST[x,y], TEMPLATE[x,y] ) < GLOBAL_ERROR
then
GLOBAL_ERROR = MSE( TEST[x,y], TEMPLATE[x,y] )
```

IMAGE_INDEX = i
end (if)
end (for)
if GLOBAL_ERROR >NON_RECOGNITION_THRESHOLD
then IMAGE_INDEX = 0

GLOBAL_ERROR is a variable to find maximum matching template with tested input image. N is the number of images in template database. MSE is a function which simply XOR the pixel values of binary images of template and tested images and then sums XOR results up. IMAGE_INDEX is the variable corresponding index of template image which best matches with test image. For example, if Figure 4.12 was taken as a test image and the recognition algorithm was applied as described, then the correct result would be 4 for IMAGE_INDEX which corresponds to t4 as shown in Figure 4.13. However if GLOBAL_ERROR is greater than certain threshold value (NON_RECOGNITION_THRESHOLD) then test image is labeled as non-recognizable and refused.

However, the algorithm has a deficiency such that, if the place of the ideogram on test image is not the same as that of template image, then recognition results are poor. It is illustrated in Figure 4.14:

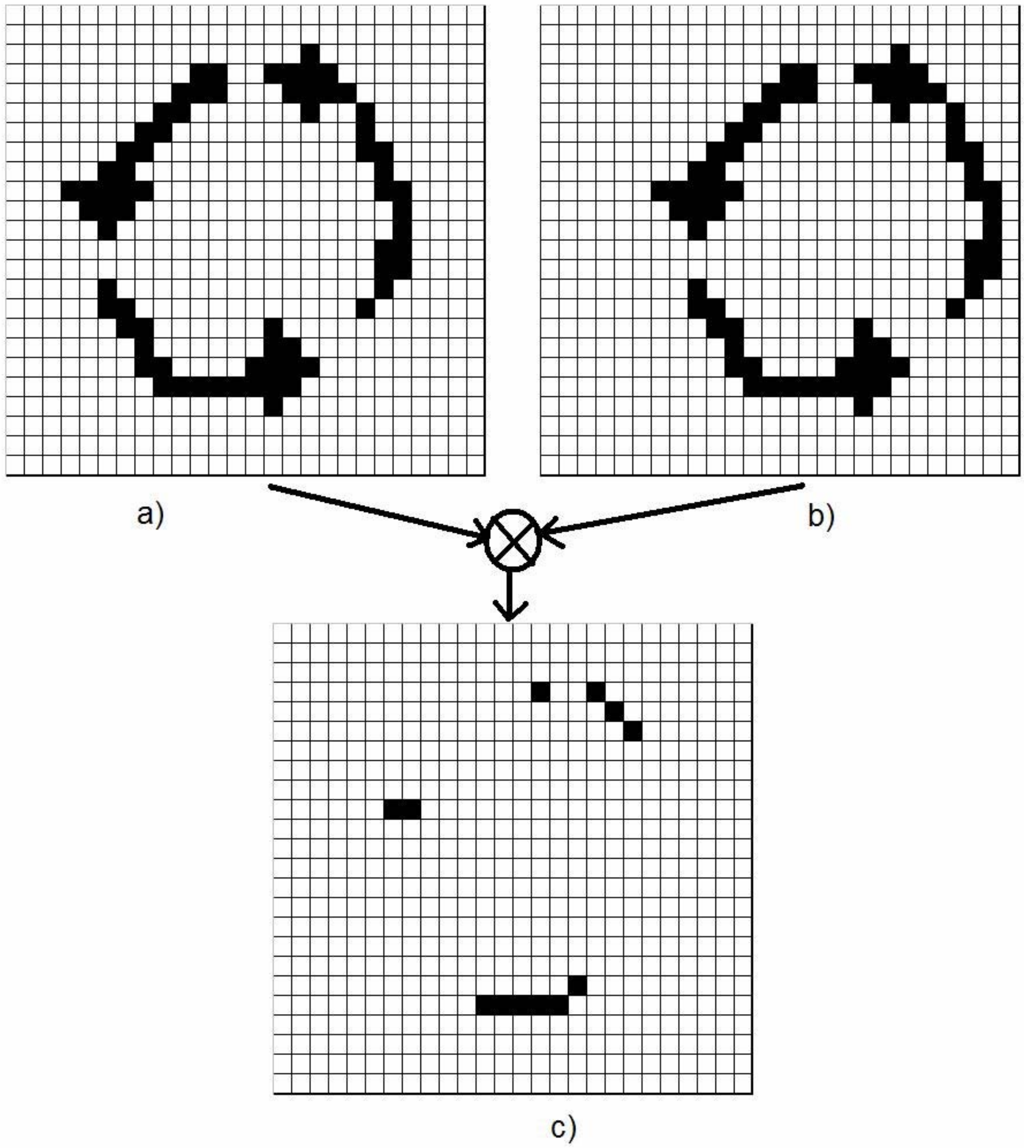


Figure 4.14: (a) Template image in database, (b) test image to be recognized, (c) output image with direct matching where black pixels illustrates the matching pixels of ideograms

As seen in Figure 4.14, although the ideograms of template image and test image are exactly the same, due to difference of placement, the matching pixel numbers are few. The solution of this problem is utilizing search window as illustrated in Figure 4.15.

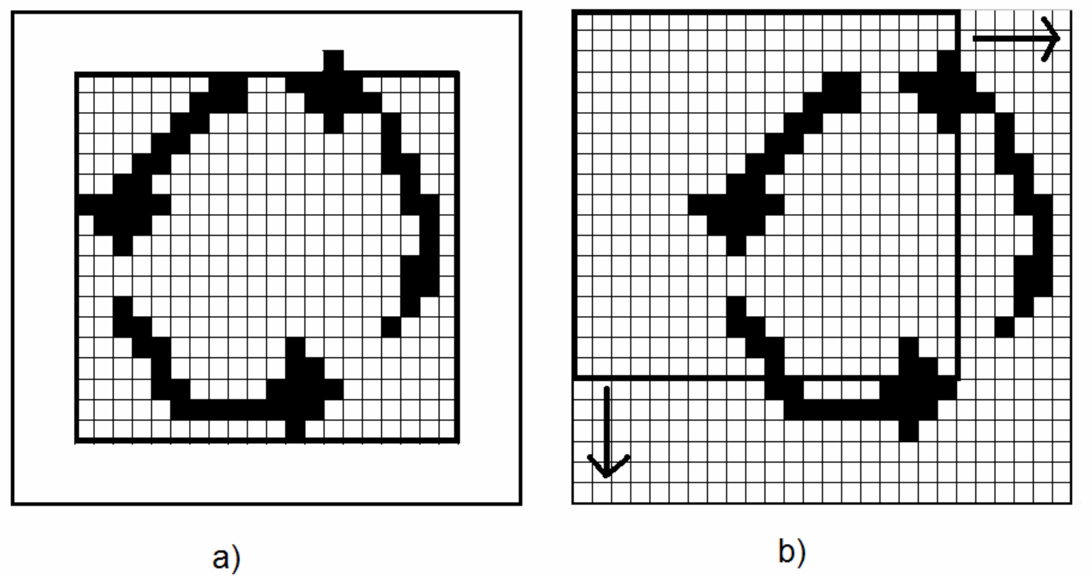


Figure 4.15: (a) Template image in database, (b) test image to be recognized

In spite of matching whole images, smaller frames are matched. At each step the frame window is shifted to cover all the images. For an example image of Figure 4.15, the number of shifting process would be 36, or in other words, the complexity increases 36 times.

The matching algorithm can be modified as below:

```
GLOBAL_ERROR = 65535
for SHIFT_INDEX=1 to MAX_SHIFT_VALUE
for i=1 to N
SET_ROI_TEST(SHIFT_INDEX)
if MSE( TEST[x,y], TEMPLATE[x,y] ) < GLOBAL_ERROR
then
GLOBAL_ERROR = MSE(TEST[x,y], TEMPLATE[x,y] )
IMAGE_INDEX = i
end (if)
end (for)
end (for)
```

if GLOBAL_ERROR >NON_RECOGNITION_THRESHOLD
then IMAGE_INDEX = 0

The added variables to algorithm are SHIFT_INDEX, MAX_SHIFT_VALUE, and added function is SET_ROI_TEST. SET_ROI_TEST modifies the ROI of test image according to its input parameter, SHIFT_INDEX. Then the algorithm would yield perfect matching for shift value of "3" in x direction, and shift value of "0" in y direction.

4.4.3 Recognition Process of Circular Traffic Signs

The recognition algorithm of circular traffic sign is similar to that of triangle but there are some minor differences. Since circular traffic signs do not have any corner point, size transformation can not correct perspective distortion effects wholly; Rotation faults can not be corrected by size transformation. Moreover due to detection algorithm which is ellipse fitting algorithm by Fitzgibbon et al. [53], the location of circular traffic sign candidate can not be as certain as that of triangular one as illustrated in Figure 4.16.

Figure 4.16: Localization of circular traffic sign

The localization of circular traffic sign may not be exactly correct because some noisy additive points or lack of some points on contour may cause fitting algorithm to fit slightly wrong. For this reason, the templates of single circular sign in template database are generally more than one. Some examples of circular sign templates obtained from detected signs are shown in Figure 4.17.

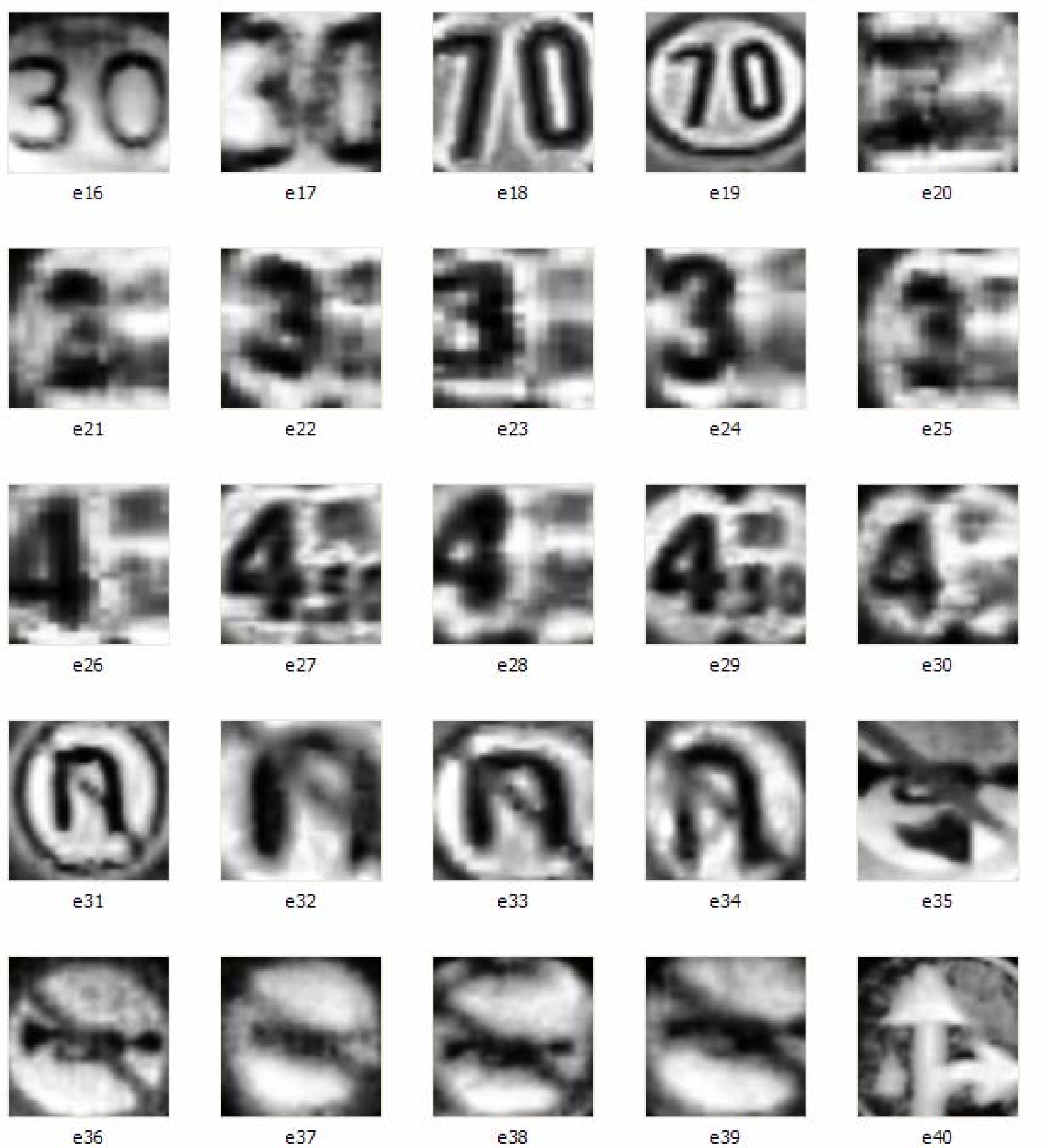


Figure 4.17: Examples of circular sign template database

4.4.4 Recognition Process of Rectangular Traffic Signs

Rectangular traffic signs are used for the purpose of regulation and guidance. The main feature of rectangular traffic signs is that they do not contain basic ideograms. Instead, they contain some more complex information in general as shown in Figure 4.18:

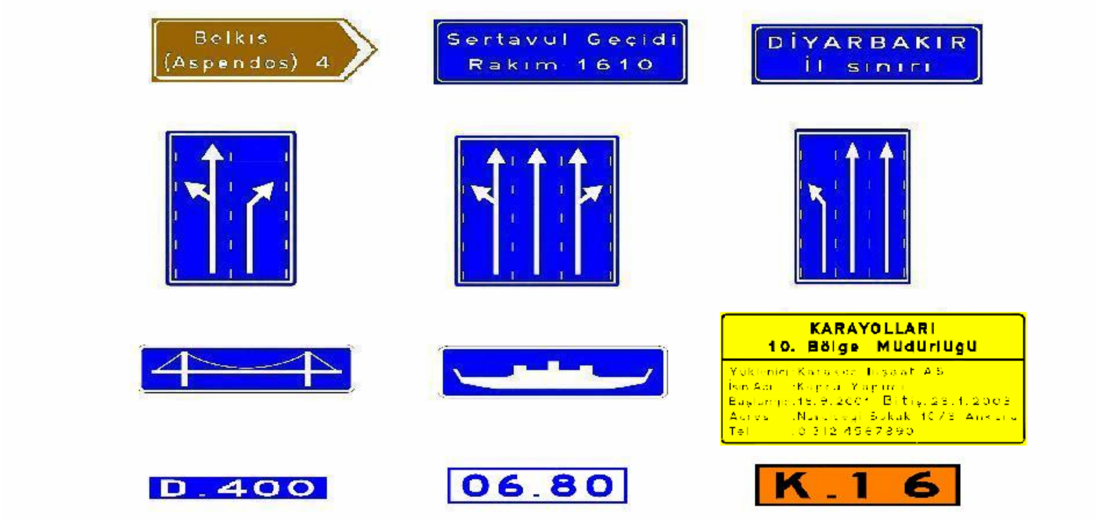


Figure 4.18: Examples of rectangular traffic signs

So ideogramatic recognition approach does not hold for rectangular signs in general. Instead detected, homographic transformed and histogram equalized rectangular image is directly applied for normalized template matching in this study.

CHAPTER 5

IMPLEMENTATIONS ON TRAFFIC SIGN DETECTION AND RECOGNITION

5.1 Properties of the System

In order to detect and to localize traffic signs, Borland C++ has been used as a code developing platform with utilizing OpenCV library. The reason to choose OpenCV library is that it is suitable for real-time video process. The video streams were taken with an 800K pixel resolution-camera in front-right of the car cabinet and the data then was grabbed with resolution of 720x576. The frames were processed by one by independent of each other, so the correlation between the video frames was not used.

5.2 Data and Parameter Set

The video streams are to be experimented by observing effect of parameters on detection and recognition of traffic signs.

5.2.1 Data Set

Four sets of video data were experimented for detection and recognition:

SET 1: Video taken starting at 14:30 in a sunny weather and lasting 10 minutes.

SET 2: Video taken at night starting at 20:45 and lasting 11 minutes.

SET 3: Video taken at night starting at 21:14 and lasting 12 minutes.

SET 4: Video taken in the evening before sunset starting at 20:10 and lasting 14 minutes.

SET 5: Video taken in the evening starting at 19:00 and lasting 24 minutes.

SET-1 is a video-stream which is composed of approximately 14000 frames. The video taken in the afternoon in a sunny day, and contains 12 triangular, 25 circular and 12 rectangular traffic signs to be recognized.

SET-2 is a video-stream which is composed of approximately 17000 frames. The video taken at night, and contains 20 triangular, 21 circular and 17 rectangular traffic signs to be recognized.

SET-3 is a video-stream which is composed of approximately 18000 frames. The video also taken at night, and contains 24 triangular, 24 circular and 26 rectangular traffic signs to be recognized.

SET-4 is a video-stream which is composed of approximately 23000 frames. The video taken before sunset contains 2 triangular, 25 circular and 9 rectangular traffic signs to be recognized.

SET-5 is a video-stream which is composed of approximately 45000 frames. The video taken evening contains 43 triangular, 58 circular and 31 rectangular traffic signs to be recognized.

5.2.2 Parameter Set

Four sets of parameters were tested on video data sets:

Parameter Set-1: Douglas-Peucker [52] threshold value "ε" are set as 0.02 times of contour perimeter for detection of triangular and rectangular sign candidates as explained in Section 2.2.1.4.5. Canny's [4] lower threshold was set as 0, upper threshold was set as 210 as explained in Section 2.2.1.2.4.

Parameter Set-2: Douglas-Peucker [52] threshold values "ε" explained in Section 2.2.1.4.5 are set as 0.2 times of contour perimeter for detection of triangular and 0.08 times of contour perimeter for detection of rectangular sign candidates. Moreover Canny's [4] lower threshold value was set as 0 while upper threshold value was set as 30 as explained in Section 2.2.1.2.4.

Parameter Set-3: Douglas-Peucker [52] threshold values "ε" explained in Section 2.2.1.4.5 are set as 0.02 times of contour perimeter for detection of triangular and 0.02 times of contour perimeter for detection of rectangular sign candidates. In addition, Canny's [4] lower threshold value was set as 0 while upper threshold value was set as 30 as explained in Section 2.2.1.2.4.

Parameter Set-4: Douglas-Peucker [52] threshold values "ε" explained in Section 2.2.1.4.5 are set as 0.2 times of contour perimeter for detection of triangular and 0.08 times of contour perimeter for detection of rectangular sign candidates. Moreover Canny's [4] lower threshold value was set as 0 while upper threshold value was set as 210 as explained in Section 2.2.1.2.4.

With Canny[4] edge detector, 3 different threshold-level sets were tested on video data sets as defined in Section 2.2.1.3:

Threshold Set-1: The number of binary threshold levels is chosen as 10. That is, binary threshold is to be applied for ten equally spread levels.

Threshold Set-2: The number of binary threshold levels is chosen as 5. That is, binary threshold is to be applied for five equally spread levels.

Threshold Set-3: The number of binary threshold levels is chosen as 3. That is, binary threshold is to be applied for three equally spread levels.

5.3 Implementation Results

In this study, SET-1 and SET-2 were experimented with all of the parameter sets and threshold sets. SET-3 and SET-4 were experimented with all of the threshold sets. SET-1 was also experimented with Canny[4] without binary threshold.

5.3.1 Implementations on SET-1

SET-1 was experimented with all of the threshold set values and for Parameter Set-1 and Parameter Set-4 values. Number of detected objects, number of detected traffic signs and false positives for each of the shape are given from Table 5.1 to 5.6 for each of the parameters set and thresholds set.

The best result is given for triangular traffic signs such that there is no false detection. Worst result is for rectangular signs, because successfully detected real rectangular signs are much less than false positives.

Number of detected objects with Parameter Set-1 is less than number of detected objects with Parameter Set-4. That is not surprise, because Douglas-Peucker [52] threshold values "ε" in Parameter Set-4 is higher than that of Parameter Set-1, which causes more objects to be extracted. However, number of false positives increase with number of detected objects sharply with Parameter Set-4.

As number of threshold values decreases, the number of detected objects also decreases in general.

Detection rates are given from Table 5.7 to 5.12:

Table 5.1 Detection with Parameter Set-1 and Threshold Set-1 (10 level)

Shape	# of traffic signs to be detected	# of detected objects	# of traffic signs among the detected objects	# of false positives
Triangular	12	12	12	0
Circular	25	33	20	11
Rectangular	12	32	7	25

Table 5.2 Detection with Parameter Set-1 and Threshold Set-2 (5 level)

Shape	# of traffic signs to be detected	# of detected objects	# of traffic signs among the detected objects	# of false positives
Triangular	12	12	12	0
Circular	25	30	19	11
Rectangular	12	22	7	15

Table 5.3 Detection with Parameter Set-1 and Threshold Set-3 (3 level)

Shape	# of traffic signs to be detected	# of detected objects	# of traffic signs among the detected objects	# of false positives
Triangular	12	12	12	0
Circular	25	26	18	8
Rectangular	12	22	7	15

Table 5.4 Detection with Parameter Set-4 and Threshold Set-1 (10 level)

Shape	# of traffic signs to be detected	# of detected objects	# of traffic signs among the detected objects	# of false positives
Triangular	12	58	12	46
Circular	25	31	20	11
Rectangular	12	131	12	119

Table 5.5 Detection with Parameter Set-4 and Threshold Set-2 (5 level)

Shape	# of traffic signs to be detected	# of detected objects	# of traffic signs among the detected objects	# of false positives
Triangular	12	40	12	28
Circular	25	30	20	10
Rectangular	12	120	12	108

Table 5.6 Detection with Parameter Set-4 and Threshold Set-3 (3 level)

Shape	# of traffic signs to be detected	# of detected objects	# of traffic signs among the detected objects	# of false positives
Triangular	12	12	12	0
Circular	25	28	20	8
Rectangular	12	21	6	15

Detection rates for triangular signs are %100 for all of the parameter sets and threshold values. Detection rates for circular signs slightly changes with parameters and threshold levels, so detection rate of circular signs mostly independent of parameter sets and threshold value. However detection rates of rectangular signs change considerably for different parameter sets and threshold values.

Recognition rates with respect to detected signs are given from Table 5.13 to 5.18:

Table 5.7 Recognition Rates with Parameter Set-1 and Threshold Set-1 (10 level)

Shape	# of traffic signs among the detected objects	# of Recognizied Traffic Signs	Recognition Rate (%)
Triangular	12	12	100,0
Circular	20	20	100,0
Rectangular	7	5	71,4

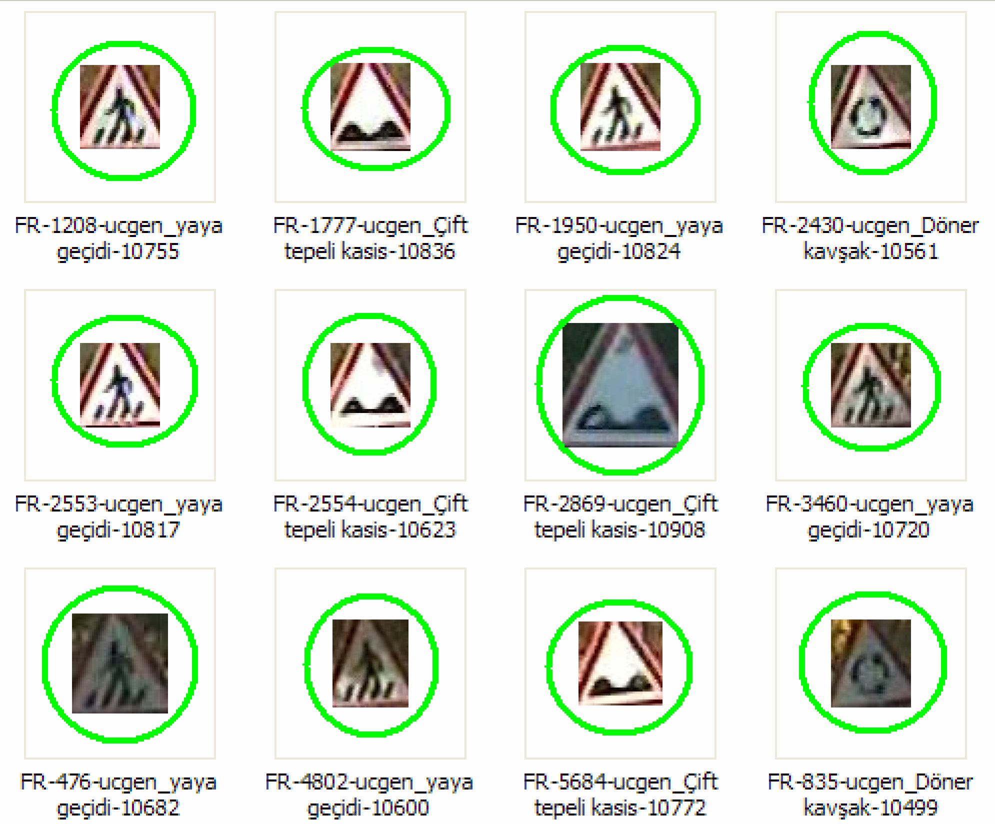


Figure 5.1: Detected and recognized triangular signs for Parameter Set-1 and Threshold Set-1 (10 levels) for Table 5.7

As seen in Figure 5.1, all of the truly detected triangular traffic signs for parameter

Figure 5.2: Detected and recognized circular signs for Parameter Set-1 and Threshold Set-1 (10 levels) for Table 5.7

Figure 5.3: Detected and recognized rectangular signs for Parameter Set-1 and Threshold Set-1 (10 levels) for Table 5.7

Table 5.8 Recognition Rates with Parameter Set-1 and Threshold Set-2 (5 level)

Shape	# of traffic signs among the detected objects	# of Recognizied Traffic Signs	Recognition Rate (%)
Triangular	12	12	100,0
Circular	19	19	100,0
Rectangular	7	5	71,4

Figure 5.4: Detected and recognized triangular signs for Parameter Set-1 and Threshold Set-2 (5 levels) for Table 5.8

Figure 5.5: Detected and recognized circular signs for Parameter Set-1 and Threshold Set-2 (5 levels) for Table 5.8

Figure 5.6: Detected and recognized rectangular signs for Parameter Set-1 and Threshold Set-2 (5 levels) for Table 5.8

Table 5.9 Recognition Rates with Parameter Set-1 and Threshold Set-3 (3 level)

Shape	# of traffic signs among the detected objects	# of Recognizied Traffic Signs	Recognition Rate (%)
Triangular	12	12	100,0
Circular	18	18	100,0
Rectangular	7	5	71,4

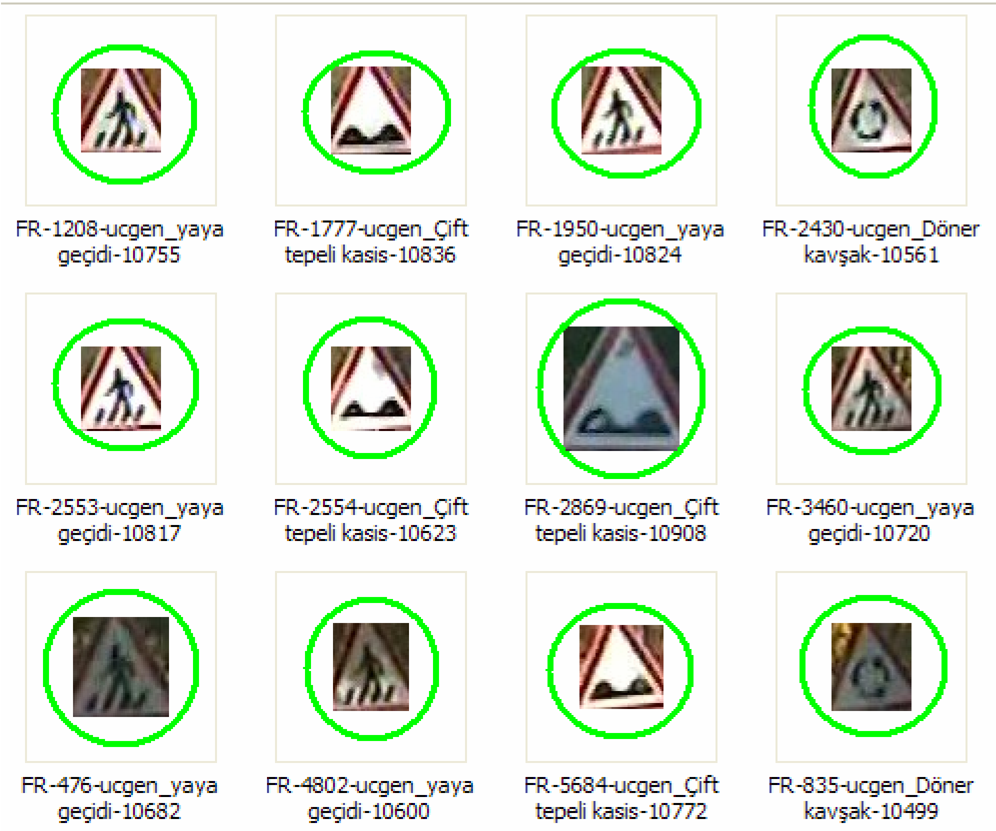


Figure 5.7: Detected and recognized triangular signs for Parameter Set-1 and Threshold Set-3 (3 levels) for Table 5.9

Figure 5.8: Detected and recognized circular signs for Parameter Set-1 and Threshold Set-3 (3 levels) for Table 5.9

Figure 5.9: Detected and recognized rectangular signs for Parameter Set-1 and Threshold Set-3 (3 levels) for Table 5.9

Table 5.10 Recognition Rates with Parameter Set-4 and Threshold Set-1 (10 level)

Shape	# of traffic signs among the detected objects	# of Recognizied Traffic Signs	Recognition Rate (%)
Triangular	12	12	100,0
Circular	20	20	100,0
Rectangular	12	7	58,3

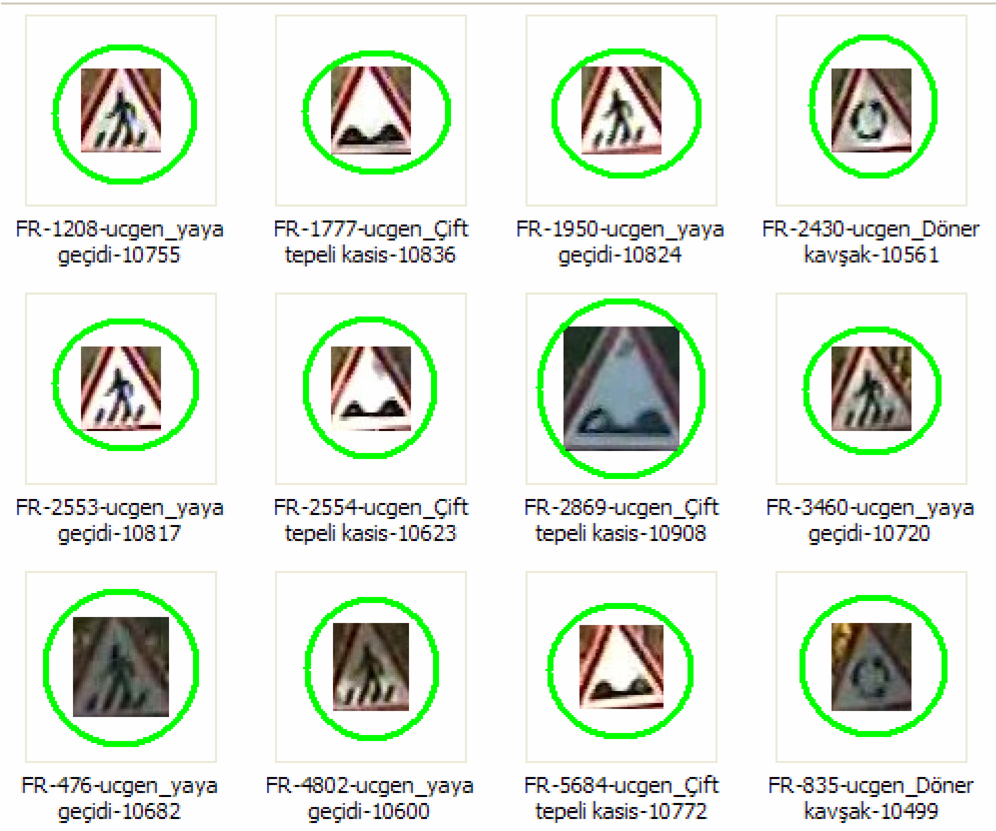


Figure 5.10: Detected and recognized triangular signs for Parameter Set-4 and Threshold Set-1 (10 levels) for Table 5.10

Figure 5.11: Detected and recognized circular signs for Parameter Set-4 and Threshold Set-1 (10 levels) for Table 5.10

Figure 5.12: Detected and recognized rectangular signs for Parameter Set-4 and Threshold Set-1 (10 levels) for Table 5.10

Table 5.11 Recognition Rates with Parameter Set-4 and Threshold Set-2 (5 level)

Shape	# of traffic signs among the detected objects	# of Recognizied Traffic Signs	Recognition Rate (%)
Triangular	12	12	100,0
Circular	20	20	100,0
Rectangular	12	7	58,3

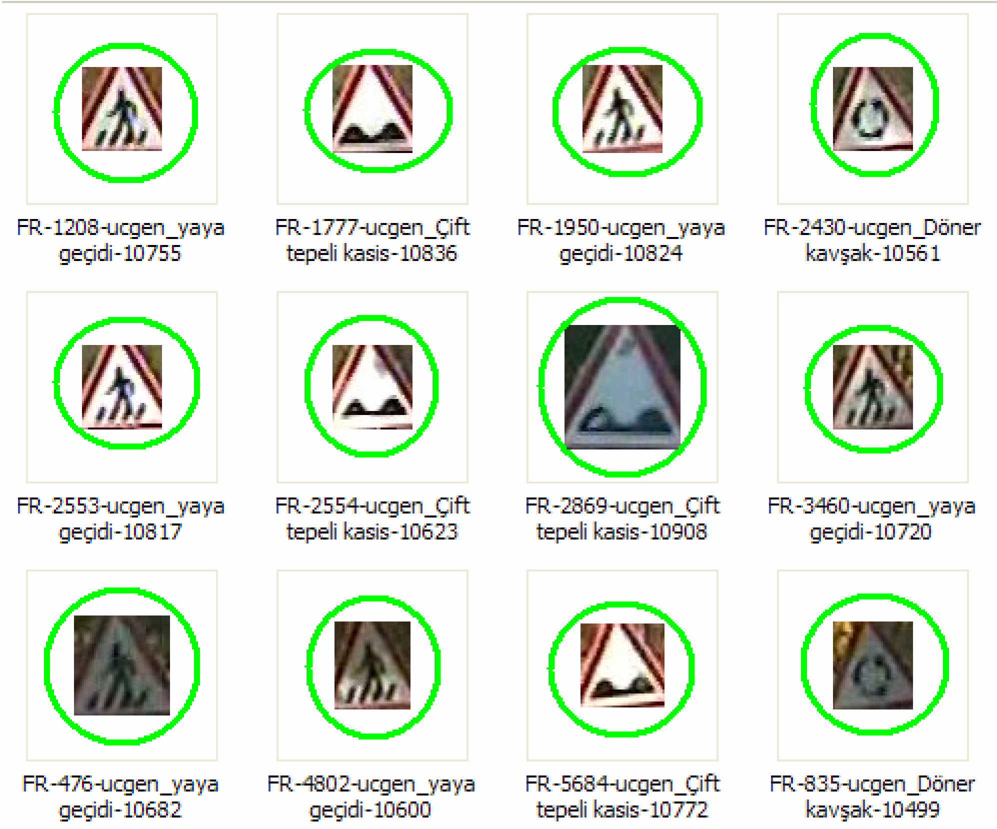


Figure 5.13: Detected and recognized triangular signs for Parameter Set-4 and Threshold Set-2 (5 levels) for Table 5.11

Figure 5.14: Detected and recognized circular signs for Parameter Set-4 and Threshold Set-2 (5 levels) for Table 5.11

Figure 5.15: Detected and recognized rectangular signs for Parameter Set-4 and Threshold Set-2 (5 levels) for Table 5.11

Table 5.12 Recognition Rates with Parameter Set-4 and Threshold Set-3 (3 level)

Shape	**# of traffic signs among the detected objects**	**# of Recognizied Traffic Signs**	**Recognition Rate (%)**
Triangular	12	12	100,0
Circular	20	20	100,0
Rectangular	6	4	66,6

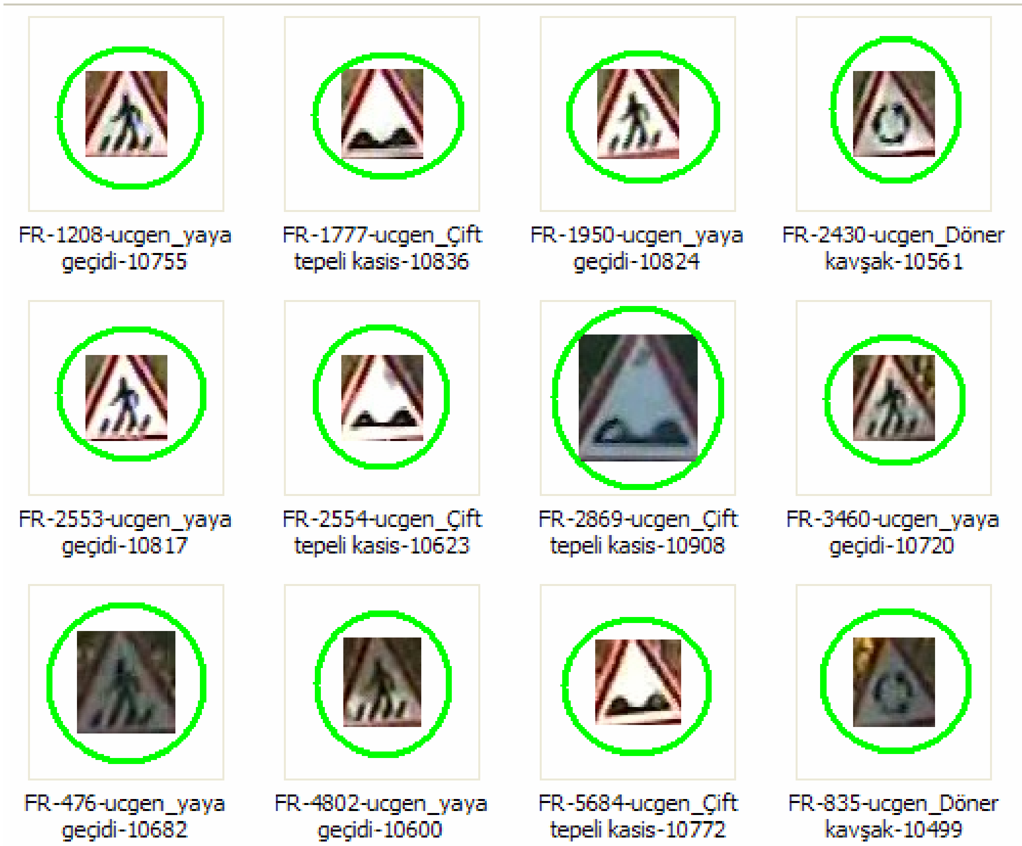


Figure 5.16: Detected and recognized triangular signs for Parameter Set-4 and Threshold Set-3 (3 levels) for Table 5.12

As seen in Figure 5.16 all of the detected signs are truly recognized.

Figure 5.17: Detected and recognized circular signs for Parameter Set-4 and Threshold Set-3 (3 levels) for Table 5.12

Figure 5.18: Detected and recognized rectangular signs for Parameter Set-4 and Threshold Set-3 (3 levels) for Table 5.12

Recognition rates are high for triangular and circular signs (%100 for all set of parameters and threshold values). The candidate objects are recognized if matching rate is higher than %92 for triangular signs where recognition is based on template matching with Manhattan distance [56], %45 for circular and rectangular signs where recognitions are based on template matching with normalized cross correlation. All of the rectangular signs which were not recognized were labeled as non-recognized for SET-1. There is a slight reduce in detection rate for circular and rectangular signs when threshold levels are decreased. There is a huge increase in detected objects when Douglas-Peucker [52] threshold value is increased for triangular and rectangular sign detection. However rise in Douglas-Peucker [52] threshold value increases detection of false positives as well as detection of true traffic signs.

In addition to experiment with using both binary threshold and Canny[4] edge detector, Canny[4] edge detector alone was experimented.

In the first experiment, Canny[4] lower threshold was set 0, and upper one was set 30 as explained in Section 2.2.1.2.4. None of the triangular and rectangular signs could be detected. One circular sign could be detected and recognized. In the second experiment, Canny[4] lower threshold was set 0, and upper one was set 210 as explained in Section 2.2.1.2.4. Two triangular signs could be detected and recognized. 17 circular signs could be detected and recognized. Three rectangular signs could be detected and one of them could be recognized.

After obtaining these results, it can be deduced that, triangular and rectangular signs are mostly detected by binary threshold in this study. Moreover Canny's [4] edge detector is very sensitive to the value of hysteresis thresholds. Lower threshold values causes more edge to be extracted. Extracting more edges results in contours mixing other objects which reduces detection rate.

5.3.2 Implementations on SET-2

SET-2 was experimented with all of the threshold set values and for Parameter Set-2 and Parameter Set-3 values. Detection rates are given from Table 5.13 to 5.18:

Table 5.13 Detection with Parameter Set-3 and Threshold Set-1 (10 level)

Shape	# of traffic signs to be detected	# of detected objects	# of traffic signs among the detected objects	# of false positives
Triangular	20	8	8	0
Circular	21	45	18	27
Rectangular	17	9	5	4

Table 5.14 Detection with Parameter Set-3 and Threshold Set-2 (5 level)

Shape	# of traffic signs to be detected	# of detected objects	# of traffic signs among the detected objects	# of false positives
Triangular	20	7	7	0
Circular	21	25	16	9
Rectangular	17	4	2	2

Table 5.15 Detection with Parameter Set-3 and Threshold Set-3 (3 level)

Shape	# of traffic signs to be detected	# of detected objects	# of traffic signs among the detected objects	# of false positives
Triangular	20	0	0	0
Circular	21	21	14	7
Rectangular	17	4	2	2

Table 5.16 Detection with Parameter Set-2 and Threshold Set-1 (10 level)

Shape	# of traffic signs to be detected	# of detected objects	# of traffic signs among the detected objects	# of false positives
Triangular	20	70	20	50
Circular	21	52	25	27
Rectangular	17	149	11	138

Table 5.17 Detection with Parameter Set-2 and Threshold Set-2 (5 level)

Shape	# of traffic signs to be detected	# of detected objects	# of traffic signs among the detected objects	# of false positives
Triangular	20	30	15	15
Circular	21	23	14	9
Rectangular	17	29	5	24

Table 5.18 Detection with Parameter Set-2 and Threshold Set-3 (3 level)

Shape	# of traffic signs to be detected	# of detected objects	# of traffic signs among the detected objects	# of false positives
Triangular	20	26	14	12
Circular	21	23	16	7
Rectangular	17	23	5	18

For Parameter Set-3, detection rates for triangular and rectangular signs are very low. However increasing the Douglas-Peucker [53] threshold value using Parameter Set-2 also increases the detection rate sharply with a cost of increasing number of false positives.

Recognition rates with respect to detected signs are given from Table 5.19 to 5.24:

Table 5.19 Recognition Rates with Parameter Set-3 and Threshold Set-1 (10 level)

Shape	# of traffic signs among the detected objects	# of Recognizied Traffic Signs	Recognition Rate (%)
Triangular	8	8	100,0
Circular	18	3	16,7
Rectangular	5	5	100,0

Figure 5.19: Detected and recognized triangular signs for Parameter Set-3 and Threshold Set-1 (10 levels) for Table 5.19

Figure 5.20: Detected and recognized circular signs for Parameter Set-3 and Threshold Set-1 (10 levels) for Table 5.19

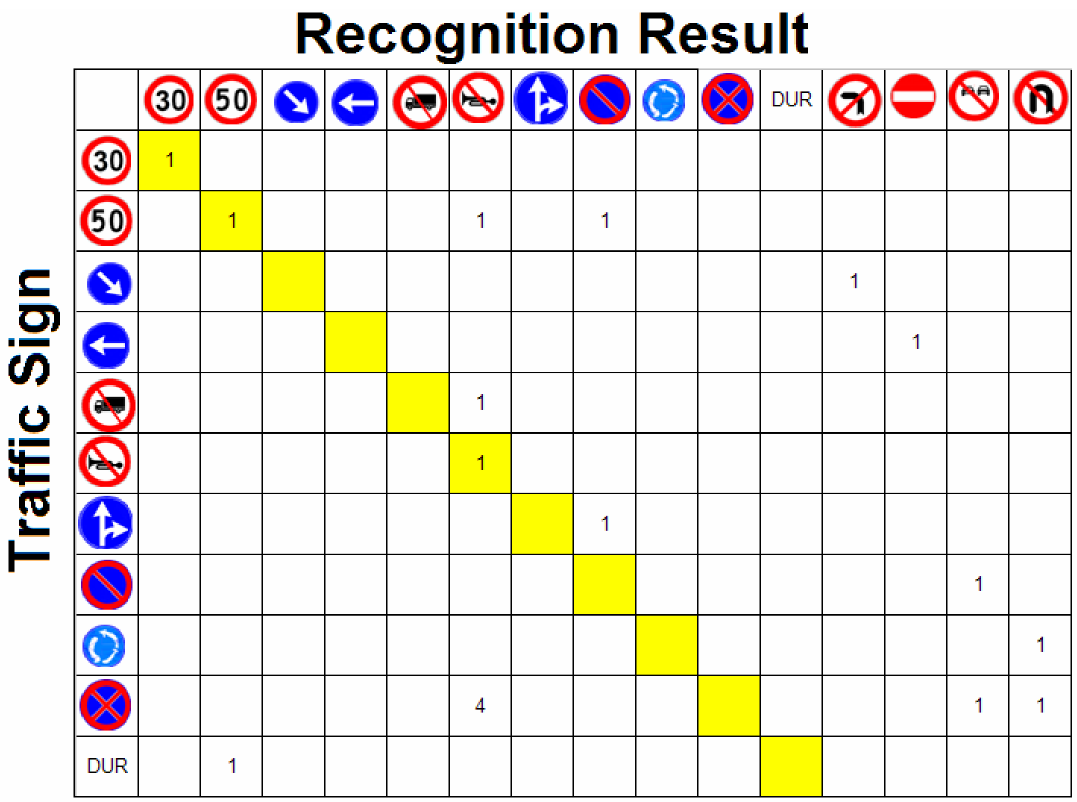


Figure 5.21: Confusion matrix for Table 5.19, for circular signs in Figure 5.20

Figure 5.22: Detected and recognized rectangular signs for Parameter Set-3 and Threshold Set-1 (10 levels) for Table 5.19

As seen in Figure 5.1, 8 circular signs are confused with other signs. Total of 15 circular signs can not be recognized for Parameter Set-3 and Threshold Set-1 (10 levels). 2 circular signs identified as unrecognizable because their recognition match rate are below %45.

Table 5.20 Recognition Rates with Parameter Set-3 and Threshold Set-2 (5 level)

Shape	# of traffic signs among the detected objects	# of Recognized Traffic Signs	Recognition Rate (%)
Triangular	7	7	100,0
Circular	16	3	19,8
Rectangular	2	2	100,0

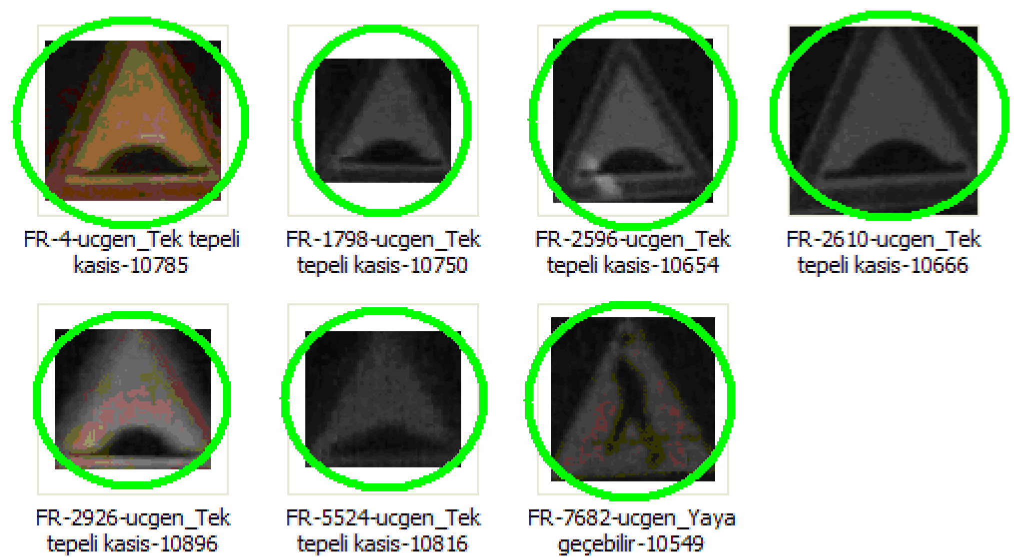


Figure 5.23: Detected and recognized triangular signs for Parameter Set-3 and Threshold Set-2 (5 levels) for Table 5.20

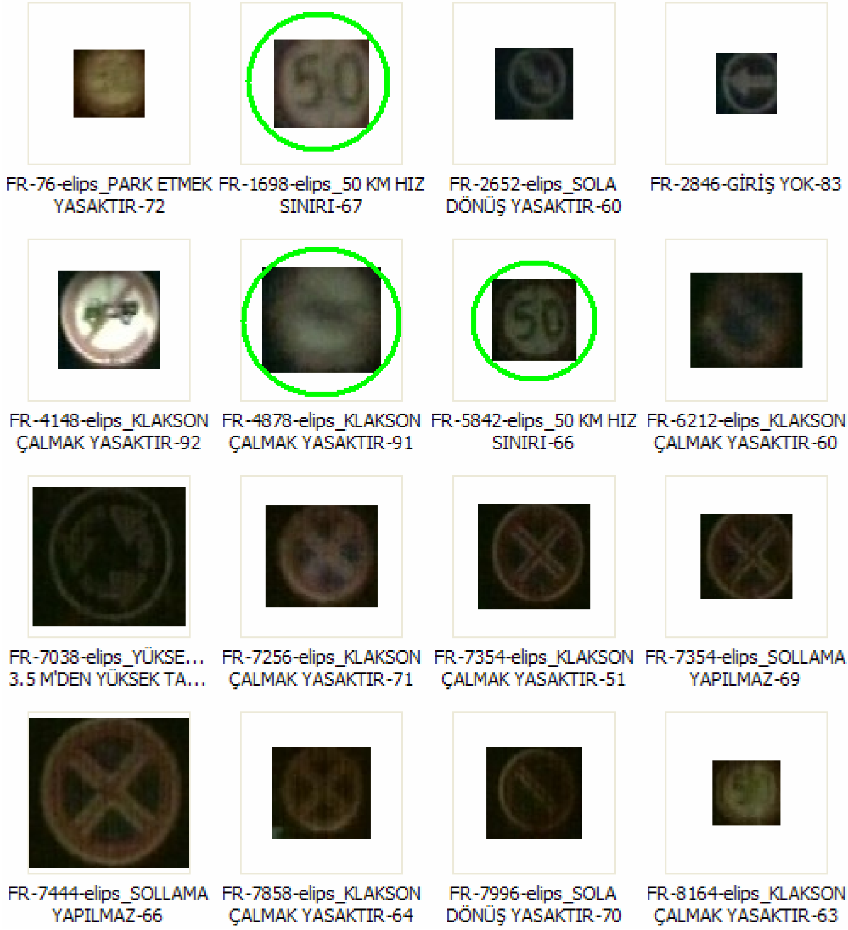


Figure 5.24: Detected and recognized circular signs for Parameter Set-3 and Threshold Set-2 (5 levels) for Table 5.20

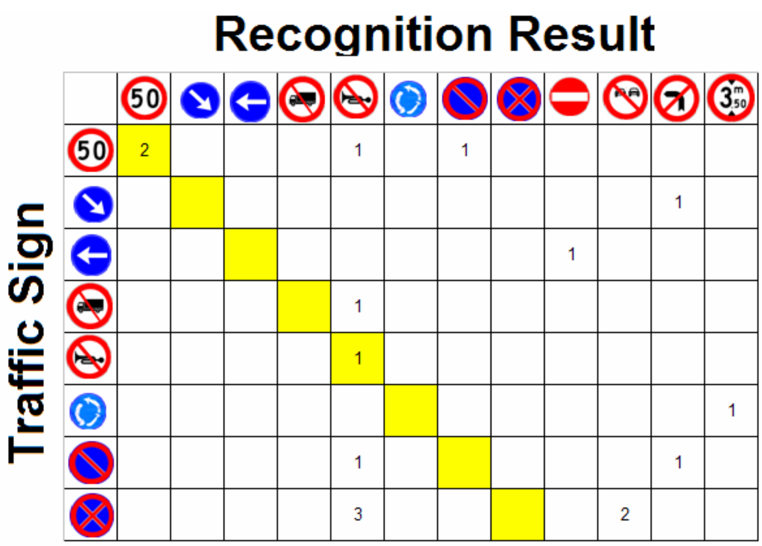


Figure 5.25: Confusion matrix for Table 5.20, for circular signs in Figure 5.24

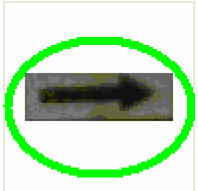


Figure 5.26: Detected and recognized rectangular signs for Parameter Set-3 and Threshold Set-2 (5 levels) for Table 5.20

As seen in Figure 5.2, 6 circular signs are confused with other signs. Total of 13 circular signs can not be recognized for Parameter Set-3 and Threshold Set-2 (5 levels). 2 circular signs identified as unrecognizable.

Table 5.21 Recognition Rates with Parameter Set-3 and Threshold Set-3 (3 level)

Shape	**# of traffic signs among the detected objects**	**# of Recognizied Traffic Signs**	**Recognition Rate (%)**
Triangular	0	0	-
Circular	14	2	10,7
Rectangular	2	2	100,0

Figure 5.27: Detected and recognized circular signs for Parameter Set-3 and Threshold Set-3 (3 levels) for Table 5.21

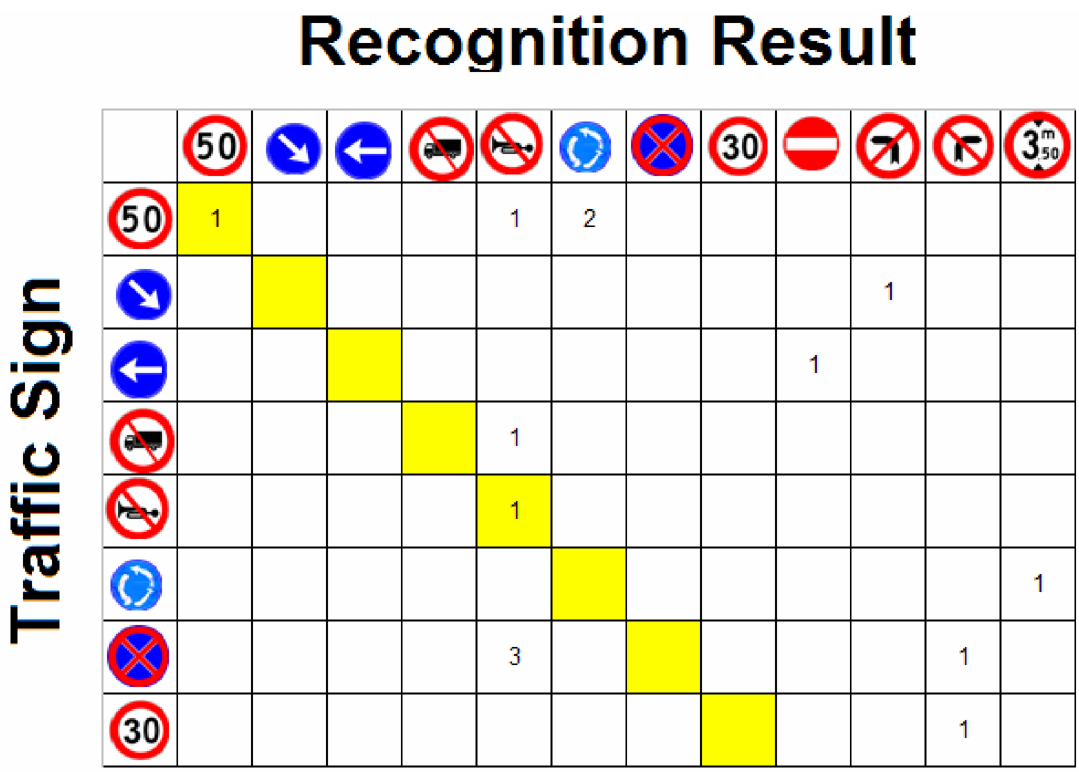


Figure 5.28: Confusion matrix for Table 5.21, for circular signs in Figure 5.27

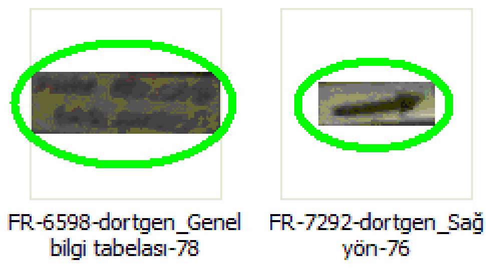


Figure 5.29: Detected and recognized rectangular signs for Parameter Set-3 and Threshold Set-3 (3 levels) for Table 5.21

Table 5.22 Recognition Rates with Parameter Set-2 and Threshold Set-1 (10 level)

Shape	# of traffic signs among the detected objects	# of Recognizied Traffic Signs	Recognition Rate (%)
Triangular	20	17	85,0
Circular	21	4	19,0
Rectangular	11	9	81,8

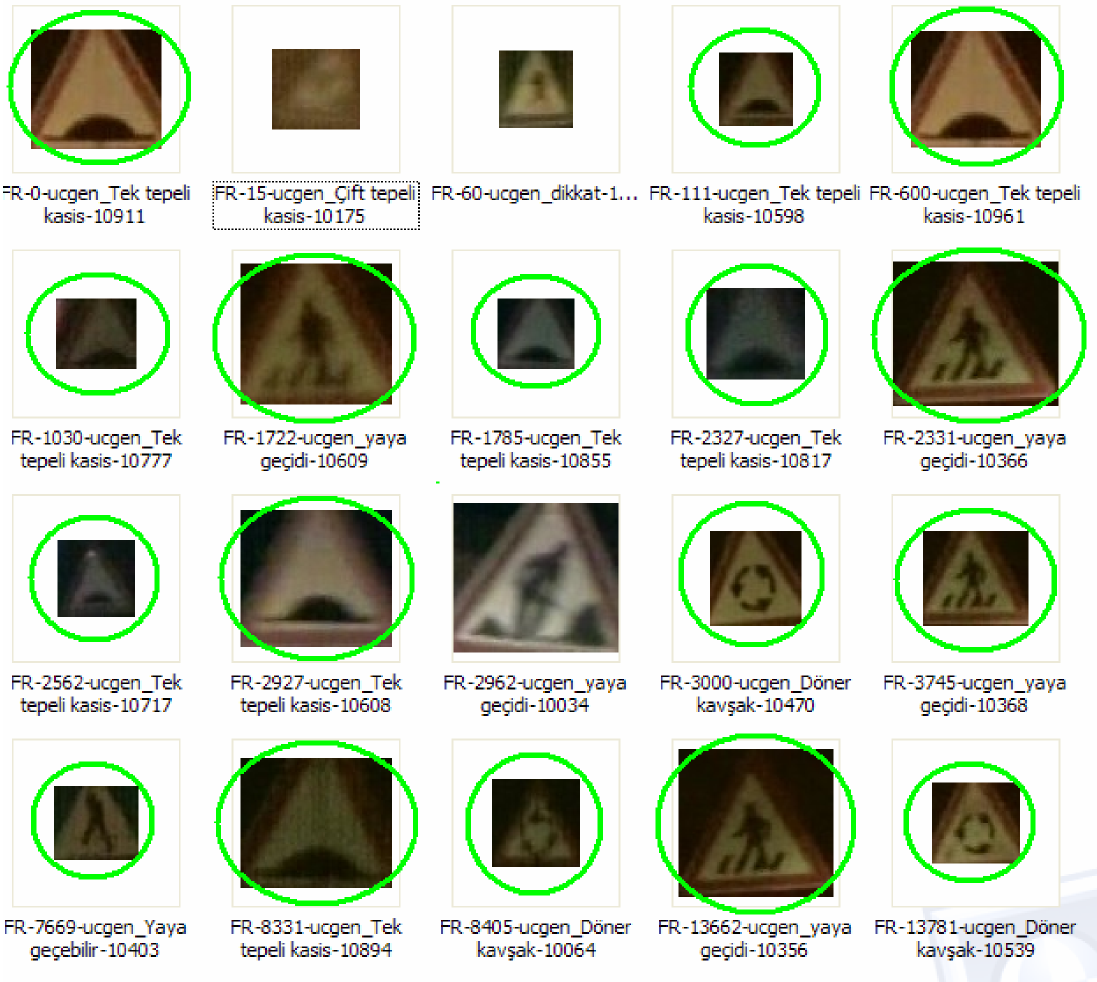


Figure 5.30: Detected and recognized triangular signs for parameter Set-2 and Threshold Set-1 (10 levels) for Table 5.22

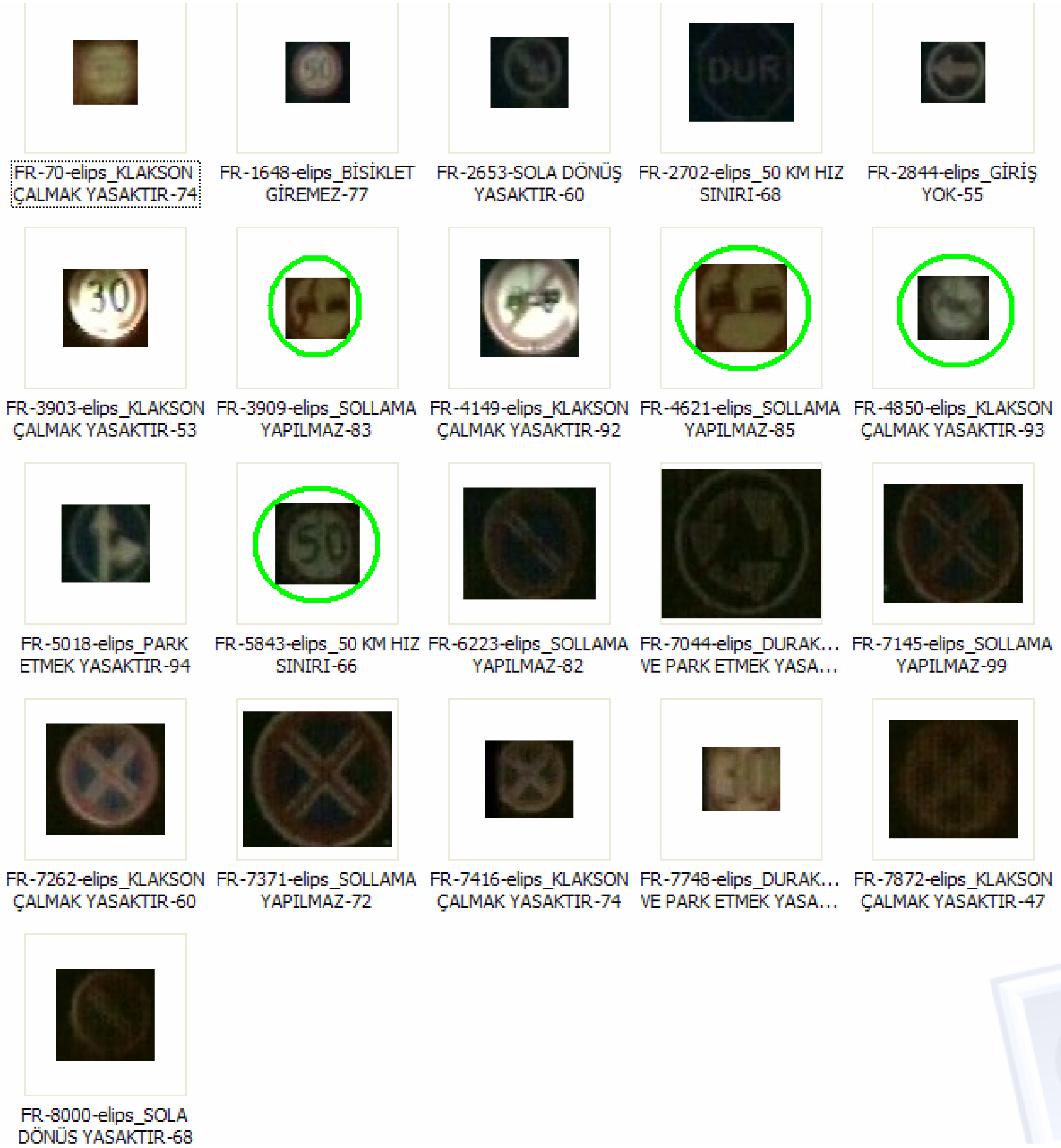


Figure 5.31: Detected and recognized circular signs for Parameter Set-2 and Threshold Set-1 (10 levels) for Table 5.22

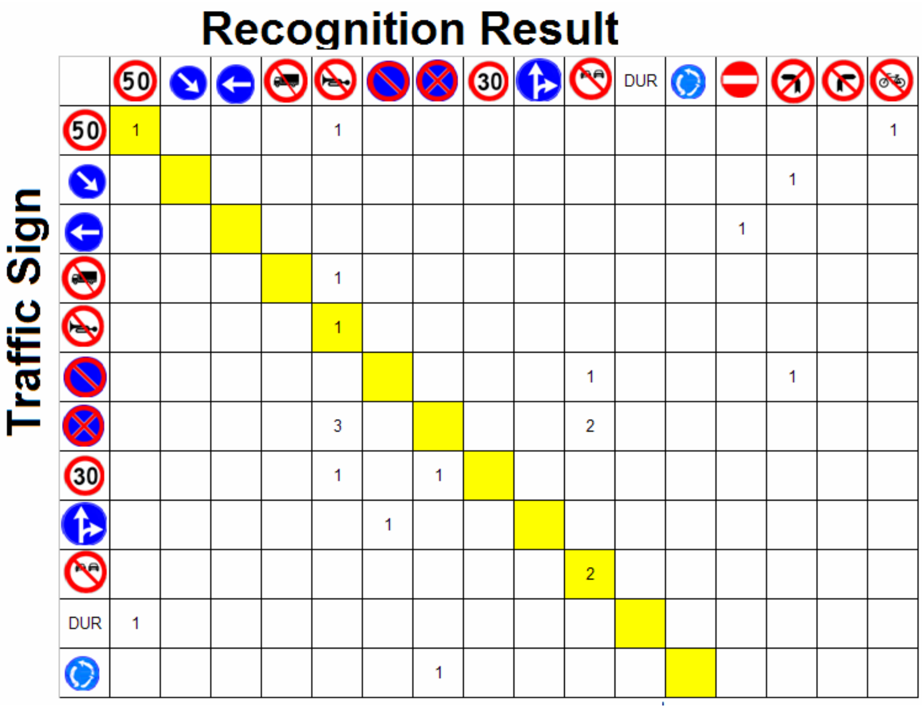


Figure 5.32: Confusion matrix for Table 5.22, for circular signs in Figure 5.31

Figure 5.33: Detected and recognized rectangular signs for Parameter Set-2 and Threshold Set-1 (10 levels) for Table 5.22

Table 5.23 Recognition Rates with Parameter Set-2 and Threshold Set-2 (5 level)

Shape	# of traffic signs among the detected objects	# of Recognizied Traffic Signs	Recognition Rate (%)
Triangular	15	12	80,0
Circular	14	5	35,7
Rectangular	5	5	100,0

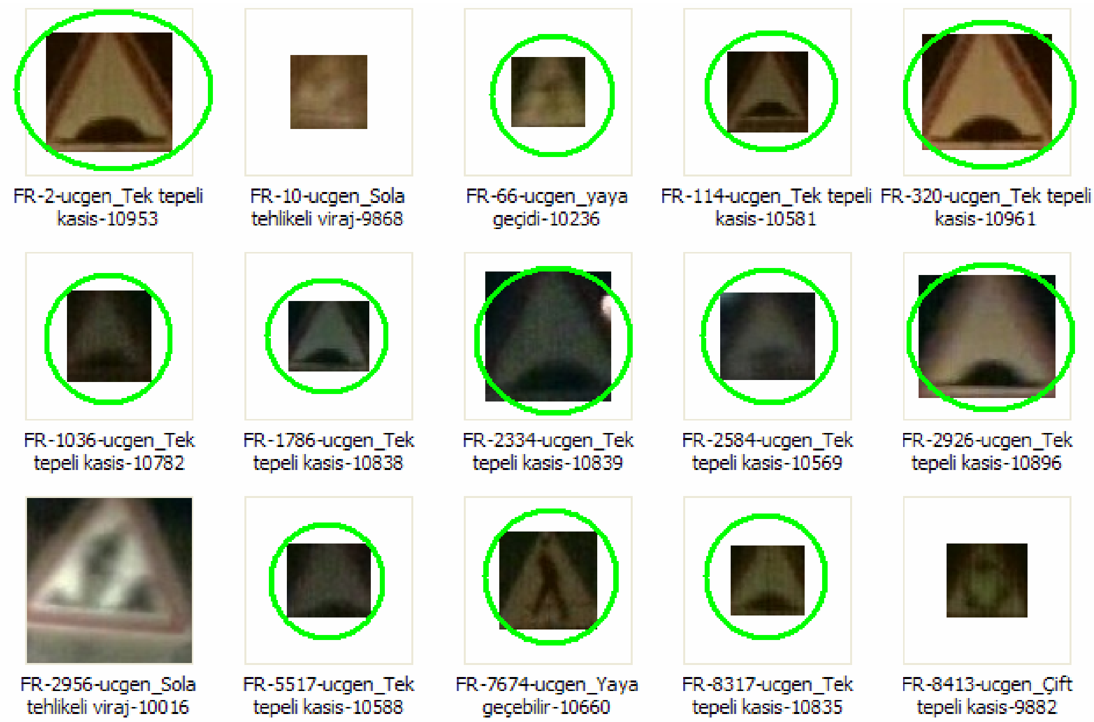


Figure 5.34: Detected and recognized triangular signs for parameter Set-2 and Threshold Set-2 (5 levels) for Table 5.23

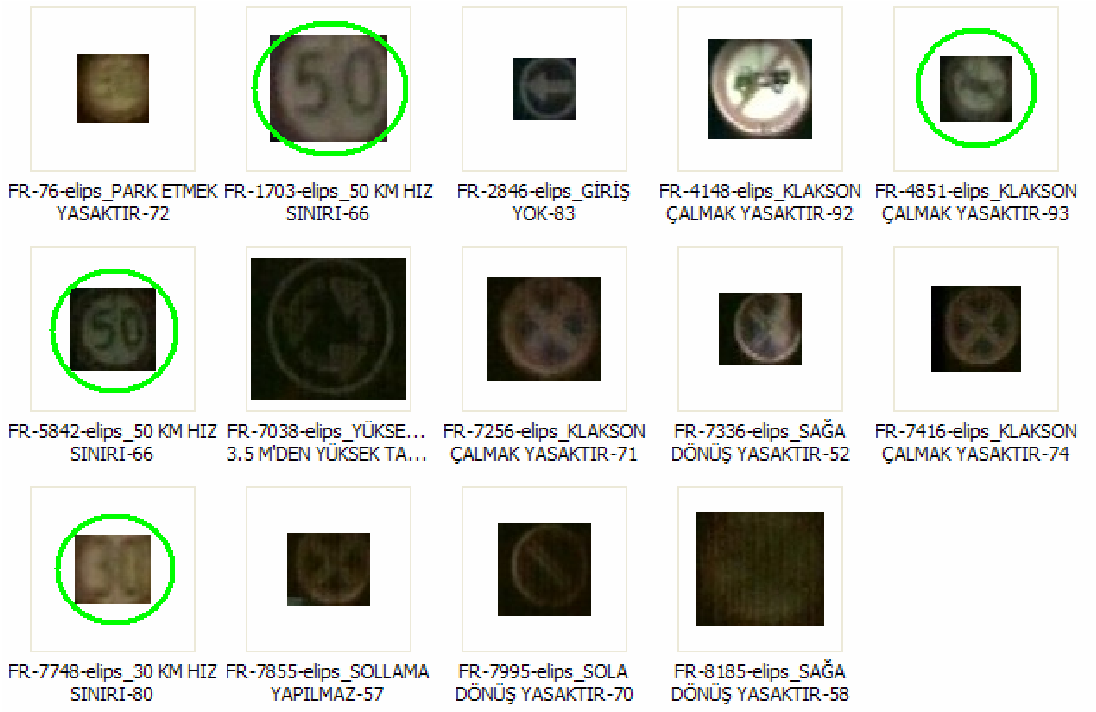


Figure 5.35: Detected and recognized circular signs for parameter Set-2 and Threshold Set-2 (5 levels) for Table 5.23

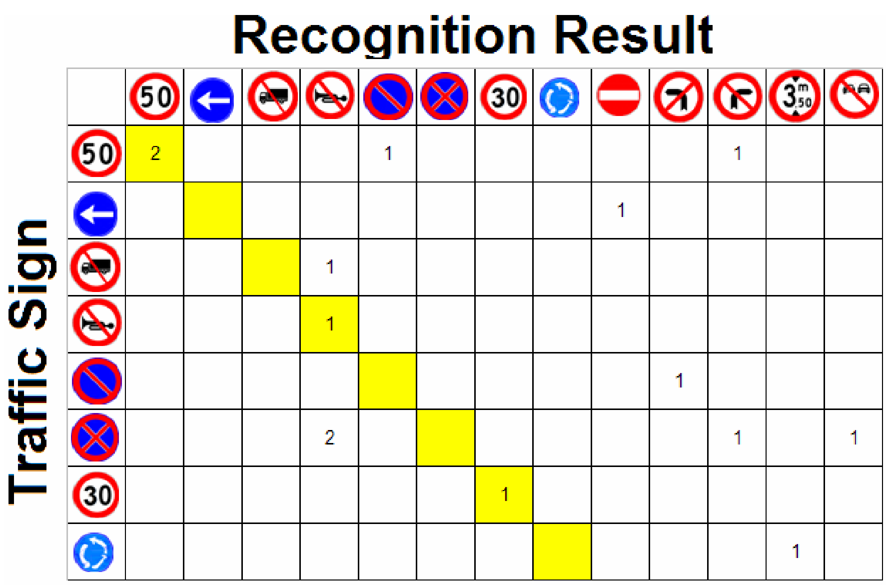


Figure 5.36: Confusion matrix for Table 5.23, for circular signs in Figure 5.35

Figure 5.37: Detected and recognized rectangular signs for parameter Set-2 and Threshold Set-2 (5 levels) for Table 5.23

Table 5.24 Recognition Rates with Parameter Set-2 and Threshold Set-3 (3 level)

Shape	# of traffic signs among the detected objects	# of Recognizied Traffic Signs	Recognition Rate (%)
Triangular	14	12	85,7
Circular	16	1	6,3
Rectangular	4	4	100,0

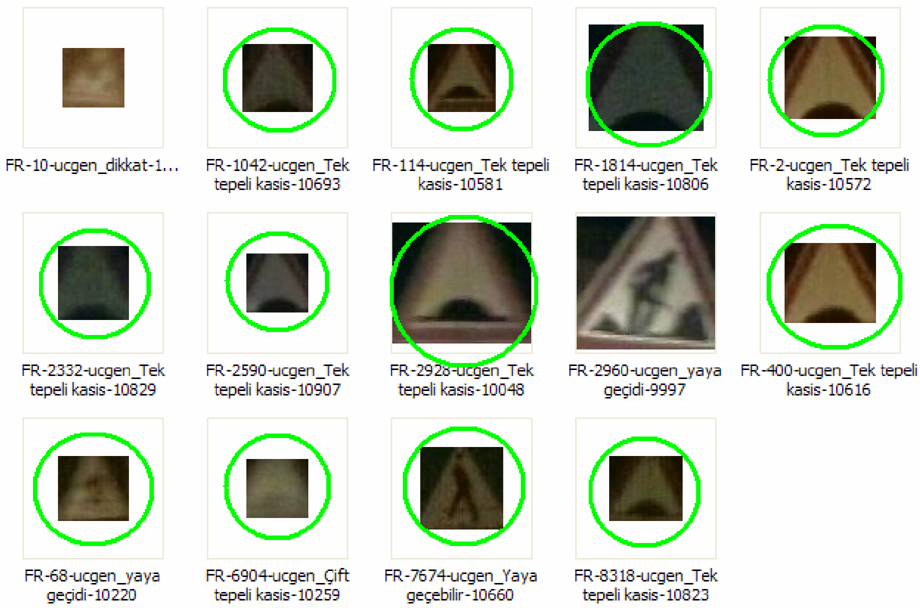


Figure 5.38: Detected and recognized triangular signs for parameter Set-2 and Threshold Set-3 (3 levels) for Table 5.24

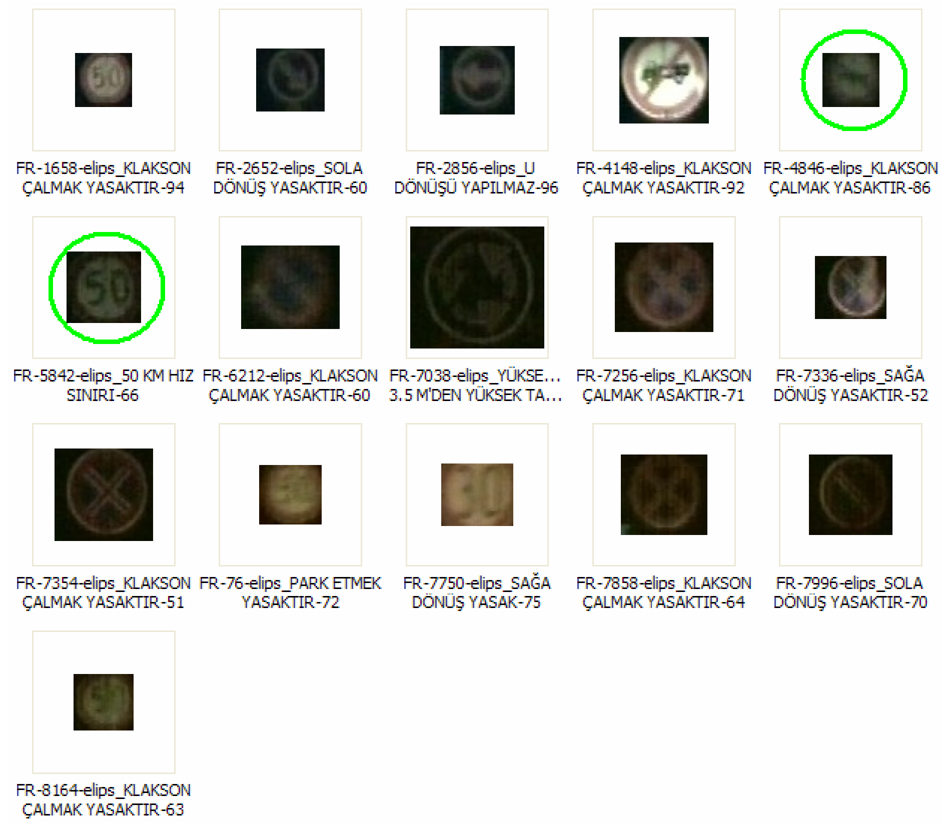


Figure 5.39: Detected and recognized circular signs for parameter Set-2 and Threshold Set-3 (3 levels) for Table 5.24

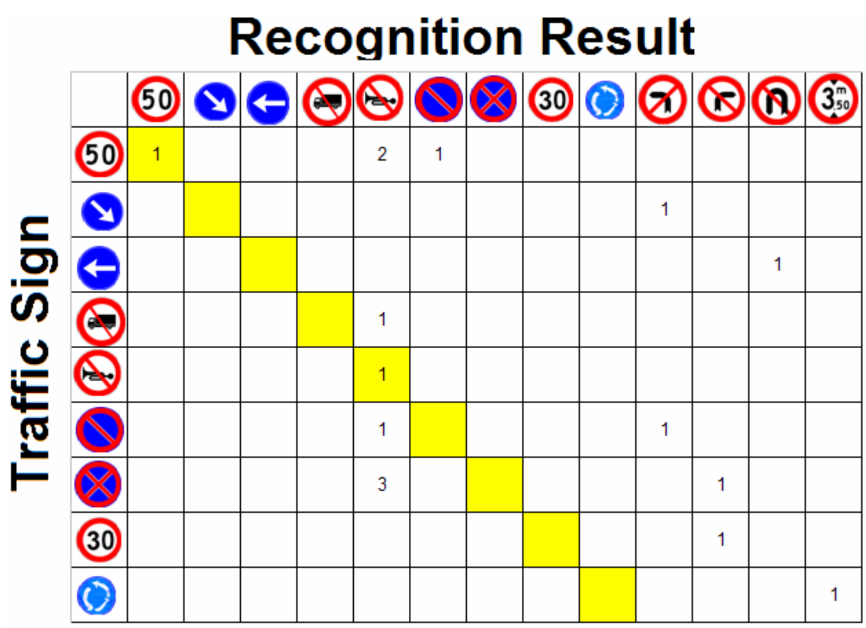


Figure 5.40: Confusion matrix for Table 5.24, for circular signs in Figure 5.39

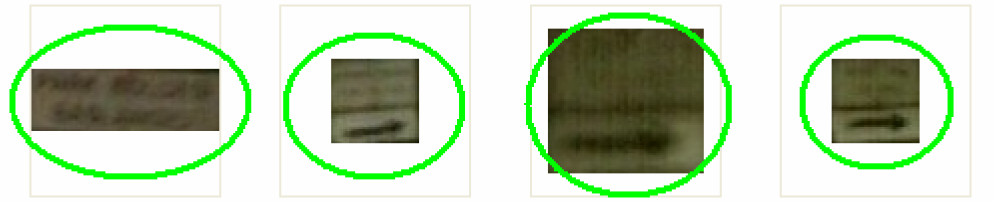


Figure 5.41: Detected and recognized rectangular signs for parameter Set-2 and Threshold Set-3 (3 levels) for Table 5.24

Dislike for SET-1, for SET-2, there is a great reduce in detection rate for circular and rectangular signs when threshold levels are decreased. Like SET-1, There is a huge increase in detected objects when Douglas-Peucker [52] threshold value is increased for triangular and rectangular sign detection. However rise in Douglas-Peucker [52] threshold value increases false positives as well as true traffic signs for detection.

The recognition rates are low especially for circular signs. Remembering that template matching is sensitive to illumination, this is probably caused from the fact that illumination of SET-2 is very low and circular sign recognizer based on normalized cross-correlation method fails. However triangular sign-recognition based on binary threshold template matching and it is more immune to illumination. Rectangular sign-recognition is similar to circular sign recognition such that both of them are based on normalized cross-correlation. However recognition rate of rectangular signs are higher. This may be caused from the fact that, at detection stage, dark rectangular signs can not be detected as well as circular and for this reason most of the rectangular signs which are dark are eliminated at detection stage, and rectangular signs have luminance high enough at recognition stage.

5.3.3 Implementations on SET-3

SET-3 was experimented with all of the threshold set values and for Parameter Set-4 values. Detection rates are given from Table 5.25 to 5.27:

Table 5.25 Detection Rates with Parameter Set-4 and Threshold Set-1 (10 level)

Shape	# of traffic signs to be detected	# of detected objects	# of traffic signs among the detected objects	# of false positives
Triangular	24	74	18	56
Circular	24	52	16	34
Rectangular	26	42	6	36

Table 5.26 Detection Rates with Parameter Set-4 and Threshold Set-2 (5 level)

Shape	# of traffic signs to be detected	# of detected objects	# of traffic signs among the detected objects	# of false positives
Triangular	24	28	18	10
Circular	24	25	16	7
Rectangular	26	9	0	9

Table 5.27 Detection Rates with Parameter Set-4 and Threshold Set-3 (3 level)

Shape	# of traffic signs to be detected	# of detected objects	# of traffic signs among the detected objects	# of false positives
Triangular	24	20	18	2
Circular	24	14	10	4
Rectangular	26	1	0	1

Detection rates of rectangular signs are %0 for threshold levels of 5 and 3. The reason may be the fact that SET-3 is a video stream taken at night and the light reflectance of rectangular signs are too low in a stream, thus rectangular signs are too dark. As the threshold levels decrease, the detection rates also decrease in general. Rising threshold levels does not contribute detection rate much, however increases the false positives.

Recognition rates with respect to detected signs are given from Table 5.28 to 5.31:

Table 5.28 Recognition Rates with Parameter Set-4 and Threshold Set-1 (10 level)

Shape	# of traffic signs among the detected objects	# of Recognizied Traffic Signs	Recognition Rate (%)
Triangular	18	14	77,8
Circular	16	12	75,0
Rectangular	6	2	33,3

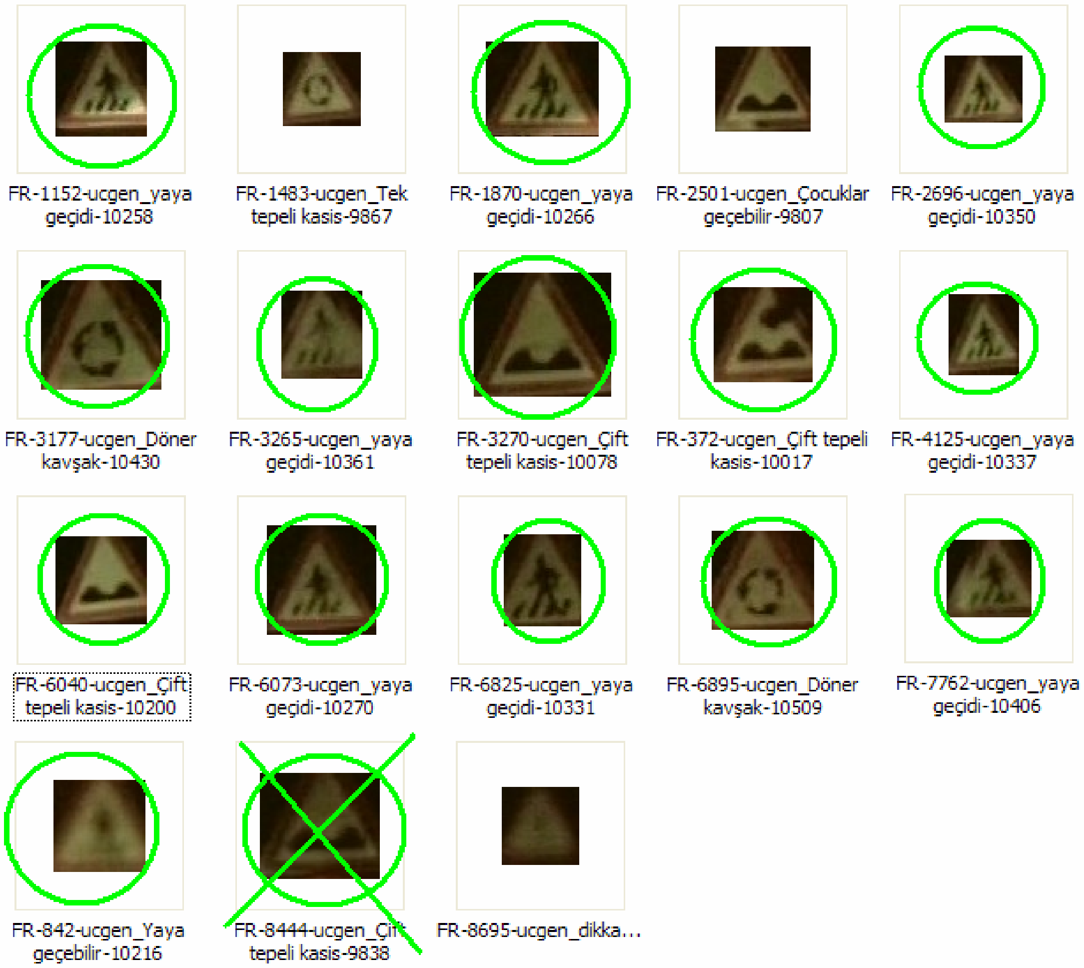


Figure 5.42: Detected and recognized triangular signs for parameter Set-4 and Threshold Set-1 (10 levels) for Table 5.28.

Although FR-8444 in Figure 5.42 is recognized and true labeled, since matching rate is under recognition threshold of 10200 (which is %92 of matching rate), it is rejected. Since triangular signs were not misclassified, confusion matrix was not given.

Figure 5.43: Detected and recognized circular signs for parameter Set-4 and Threshold Set-1 (10 levels) for Table 5.28

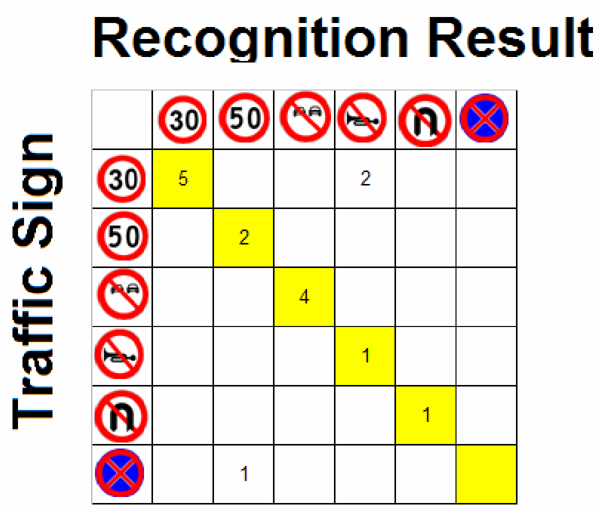


Figure 5.44: Confusion matrixes for Table 5.28, for circular signs in Figure 5.43

Table 5.29 Recognition Rates with Parameter Set-4 and Threshold Set-2 (5 level)

Shape	# of traffic signs among the detected objects	# of Recognizied Traffic Signs	Recognition Rate (%)
Triangular	18	12	66,7
Circular	16	10	62,5
Rectangular	0	0	-

Figure 5.45: Detected and recognized circular signs for parameter Set-4 and Threshold Set-2 (5 levels) for Table 5.29

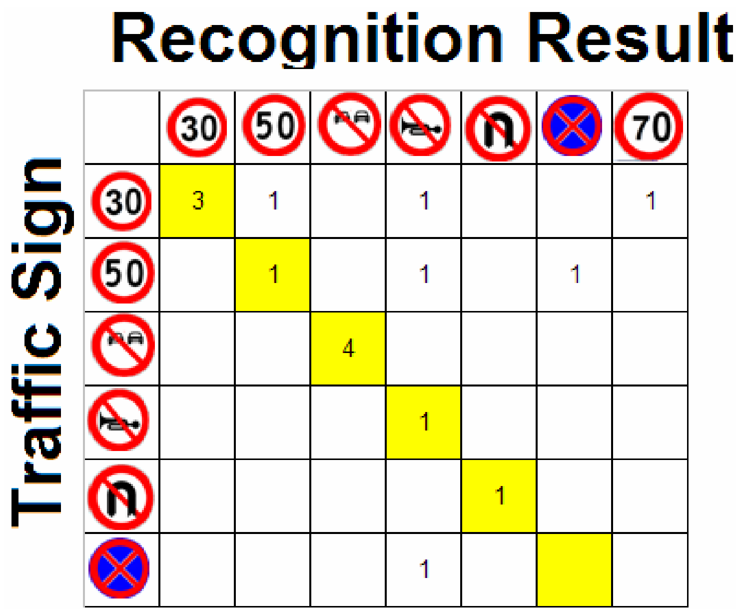


Figure 5.46: Confusion matrix for Table 5.29, for circular signs in Figure 5.45

Table 5.30 Recognition Rates with Parameter Set-4 and Threshold Set-3 (3 level)

Shape	# of traffic signs among the detected objects	# of Recognizied Traffic Signs	Recognition Rate (%)
Triangular	18	14	77,8
Circular	10	6	60,0
Rectangular	0	0	-

When SET-1 (day time) and SET-3 (night time) are compared it is seen that recognition rates are much lower for SET-3 than that for SET-1.

5.3.4 Implementations on SET-4

SET-4 was experimented with all of the threshold set values and for Parameter Set-1 values. Detection rates are given from Table 5.31 to 5.33:

Table 5.31 Detection Rates with Parameter Set-1 and Threshold Set-1 (10 level)

Shape	# of traffic signs to be detected	# of detected objects	# of traffic signs among the detected objects	# of false positives
Triangular	2	1	1	0
Circular	25	67	16	51
Rectangular	9	127	6	121

Table 5.32 Detection Rates with Parameter Set-1 and Threshold Set-2 (5 level)

Shape	# of traffic signs to be detected	# of detected objects	# of traffic signs among the detected objects	# of false positives
Triangular	2	1	1	0
Circular	25	39	10	29
Rectangular	9	49	5	44

Table 5.33 Detection Rates with Parameter Set-1 and Threshold Set-3 (3 level)

Shape	# of traffic signs to be detected	# of detected objects	# of traffic signs among the detected objects	# of false positives
Triangular	2	1	1	0
Circular	25	27	10	17
Rectangular	9	33	3	30

Recognition rates with respect to detected signs are given from Table 5.34 to 5.36:

Table 5.34 Recognition Rates with Parameter Set-1 and Threshold Set-1 (10 level)

Shape	# of traffic signs among the detected objects	# of Recognizied Traffic Signs	Recognition Rate (%)
Triangular	1	1	100,0
Circular	16	12	75
Rectangular	6	4	66,6

Figure 5.47: Detected and recognized triangular signs for parameter Set-1 and Threshold Set-1 (10 levels) for Table 5.34.

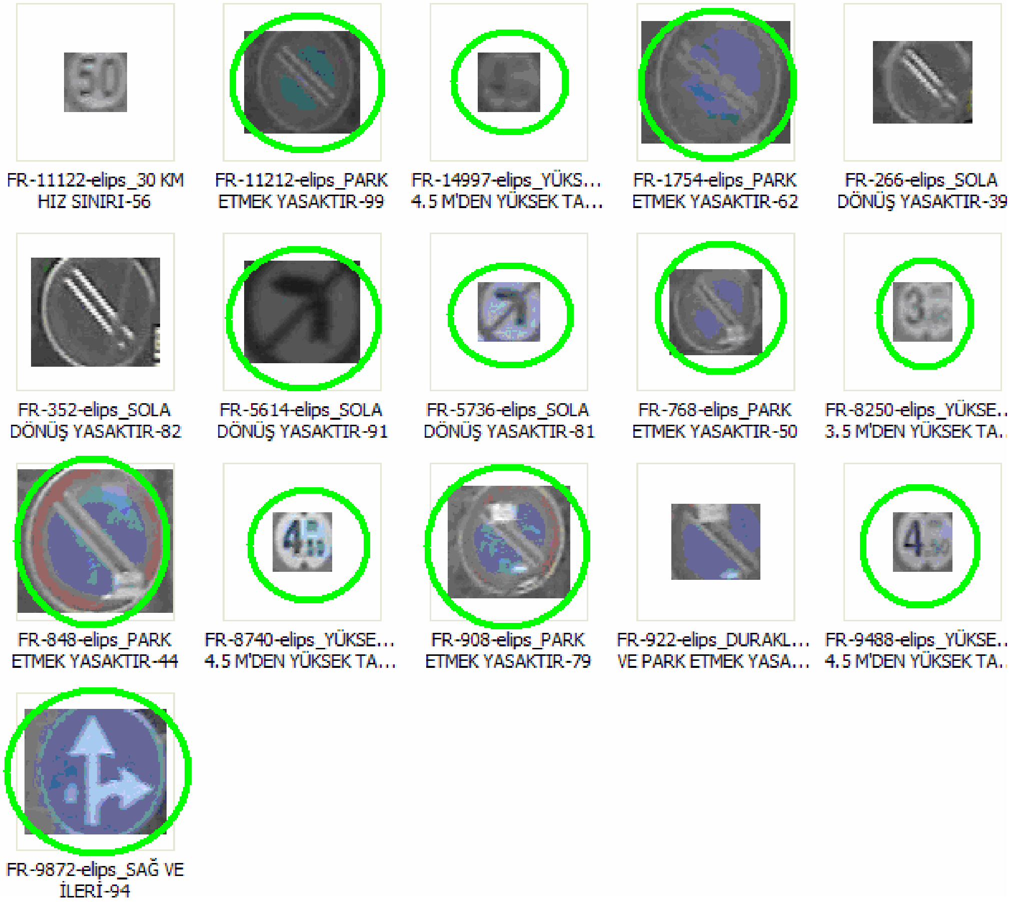


Figure 5.48: Detected and recognized circular signs for parameter Set-1 and Threshold Set-1 (10 levels) for Table 5.34.

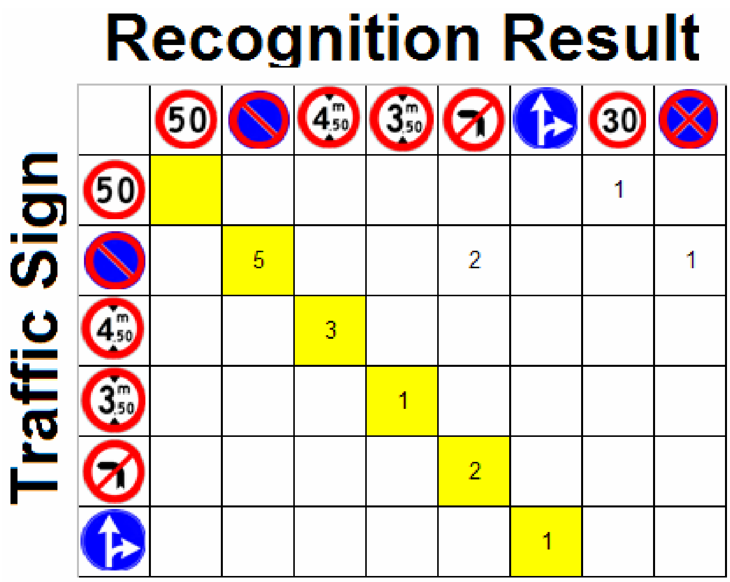


Figure 5.49: Confusion matrix for Table 5.34, for circular signs in Figure 5.48

Figure 5.50: Detected and recognized rectangular signs for parameter Set-1 and Threshold Set-1 (10 levels) for Table 5.34.

Table 5.35 Recognition Rates with Parameter Set-1 and Threshold Set-2 (5 level)

Shape	# of traffic signs among the detected objects	# of Recognizied Traffic Signs	Recognition Rate (%)
Triangular	1	1	100,0
Circular	10	7	70,0
Rectangular	5	3	60,0

Figure 5.51: Detected and recognized triangular signs for parameter Set-1 and Threshold Set-2 (5 levels) for Table 5.35.

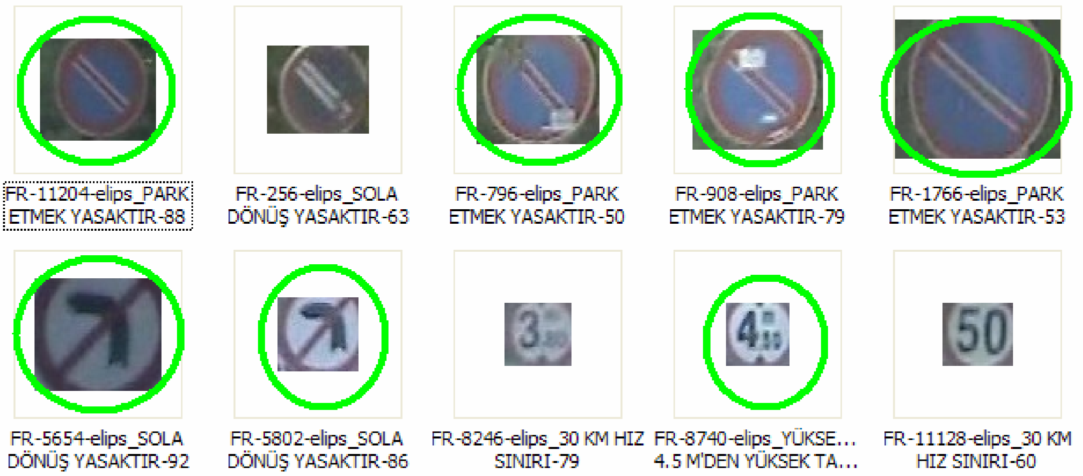


Figure 5.52: Detected and recognized circular signs for parameter Set-1 and Threshold Set-2 (5 levels) for Table 5.35.

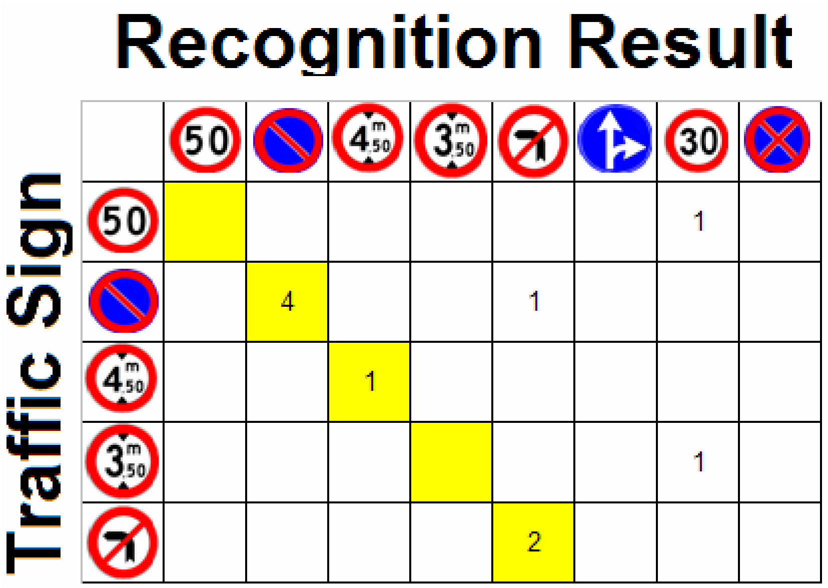


Figure 5.53: Confusion matrix for Table 5.35, for circular signs in Figure 5.52

Figure 5.54: Detected and recognized rectangular signs for parameter Set-1 and Threshold Set-2 (5 levels) for Table 5.35.

Table 5.36 Recognition Rates with Parameter Set-1 and Threshold Set-3 (3 level)

Shape	# of traffic signs among the detected objects	# of Recognizied Traffic Signs	Recognition Rate (%)
Triangular	1	1	100,0
Circular	10	6	60,0
Rectangular	3	2	66,6

Figure 5.55: Detected and recognized triangular signs for parameter Set-1 and Threshold Set-3 (3 levels) for Table 5.36.

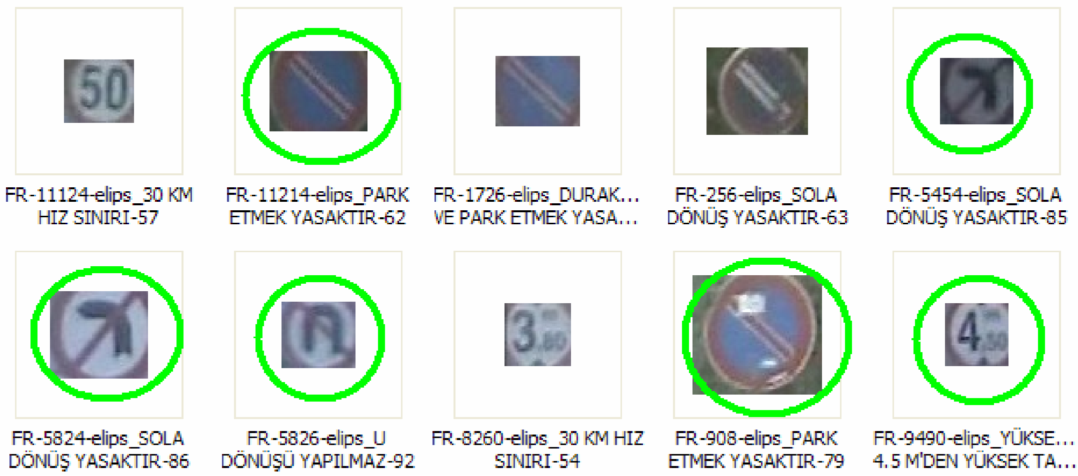


Figure 5.56: Detected and recognized circular signs for parameter Set-1 and Threshold Set-3 (3 levels) for Table 5.36.

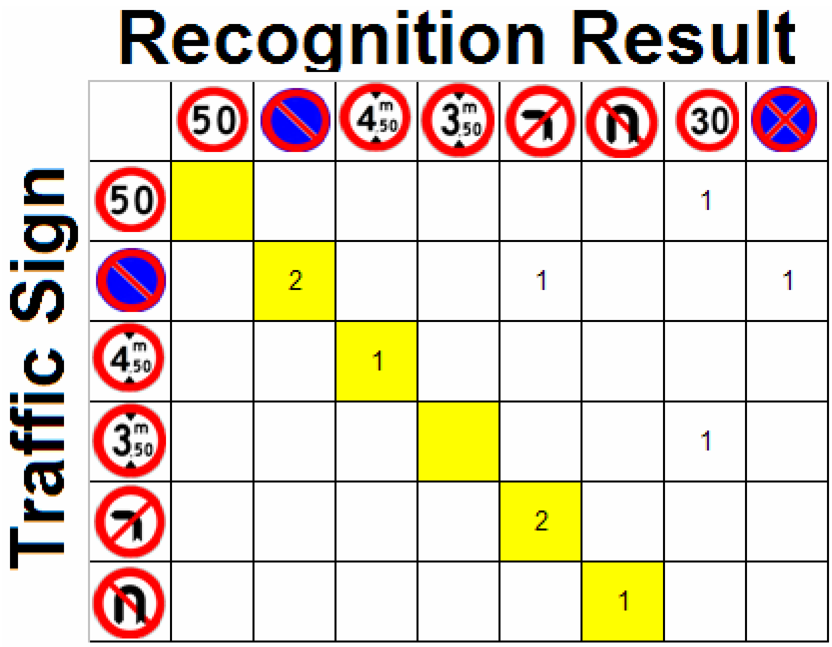


Figure 5.57: Confusion matrix for Table 5.36, for circular signs in Figure 5.56

Figure 5.58: Detected and recognized rectangular signs for parameter Set-1 and Threshold Set-3 (3 levels) for Table 5.36.

Since the number of triangular signs in SET-4 is low, it may not be considered as dataset large enough but circular and rectangular signs can be compared with those of other SETs.

5.3.5 Implementations on SET-5

SET-5 was experimented with all of the threshold set values and for Parameter Set-1 values. Detection rates are given from Table 5.37 to 5.39:

Table 5.37 Detection Rates with Parameter Set-1 and Threshold Set-1 (10 level)

Shape	# of traffic signs to be detected	# of detected objects	# of traffic signs among the detected objects	# of false positives
Triangular	43	40	40	0
Circular	58	76	40	36
Rectangular	31	82	17	65

Table 5.38 Detection Rates with Parameter Set-1 and Threshold Set-2 (5 level)

Shape	# of traffic signs to be detected	# of detected objects	# of traffic signs among the detected objects	# of false positives
Triangular	43	37	37	0
Circular	58	62	37	25
Rectangular	31	58	15	43

Table 5.39 Detection Rates with Parameter Set-1 and Threshold Set-3 (3 level)

Shape	# of traffic signs to be detected	# of detected objects	# of traffic signs among the detected objects	# of false positives
Triangular	43	31	31	0
Circular	58	32	10	22
Rectangular	31	44	13	31

As similar to other video Sets, for SET-5 the number of detected objects and traffic signs reduces considerably by the reducing number of threshold levels.

Recognition rates with respect to detected signs are given from Table 5.40 to 5.42:

Table 5.40 Recognition Rates with Parameter Set-1 and Threshold Set-1 (10 level)

Shape	# of traffic signs among the detected objects	# of Recognizied Traffic Signs	Recognition Rate (%)
Triangular	40	40	100,0
Circular	40	25	62,5
Rectangular	17	13	76,4

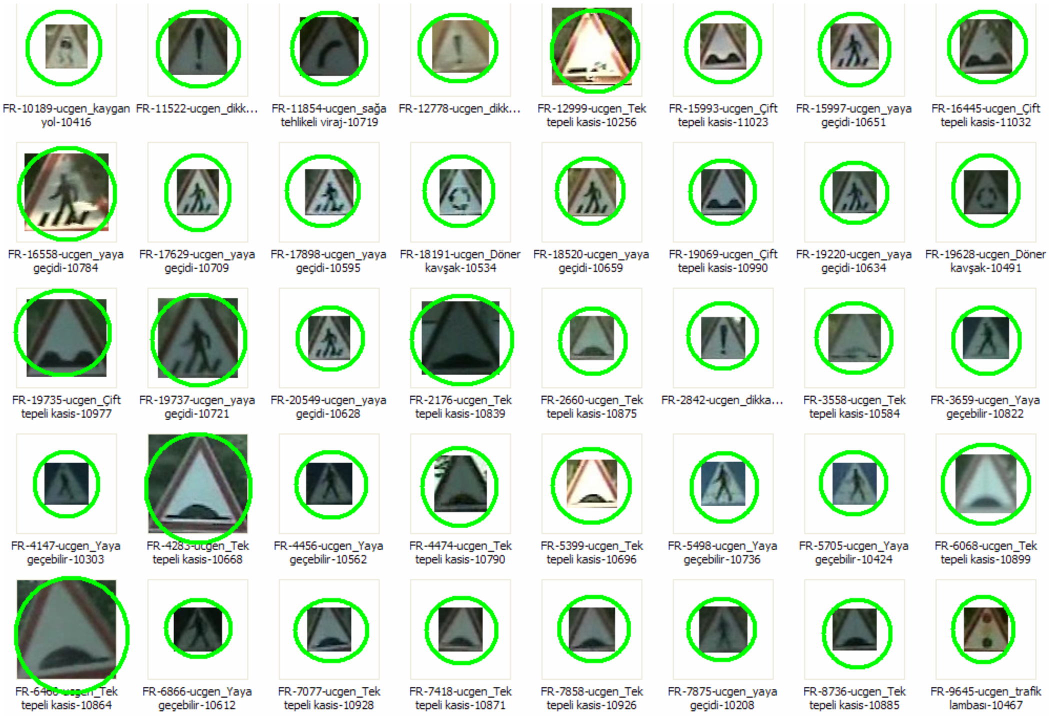


Figure 5.59: Detected and recognized triangular signs for parameter Set-1 and Threshold Set-1 (10 levels) for Table 5.40.

Figure 5.60: Detected and recognized circular signs for parameter Set-1 and Threshold Set-1 (10 levels) for Table 5.40.

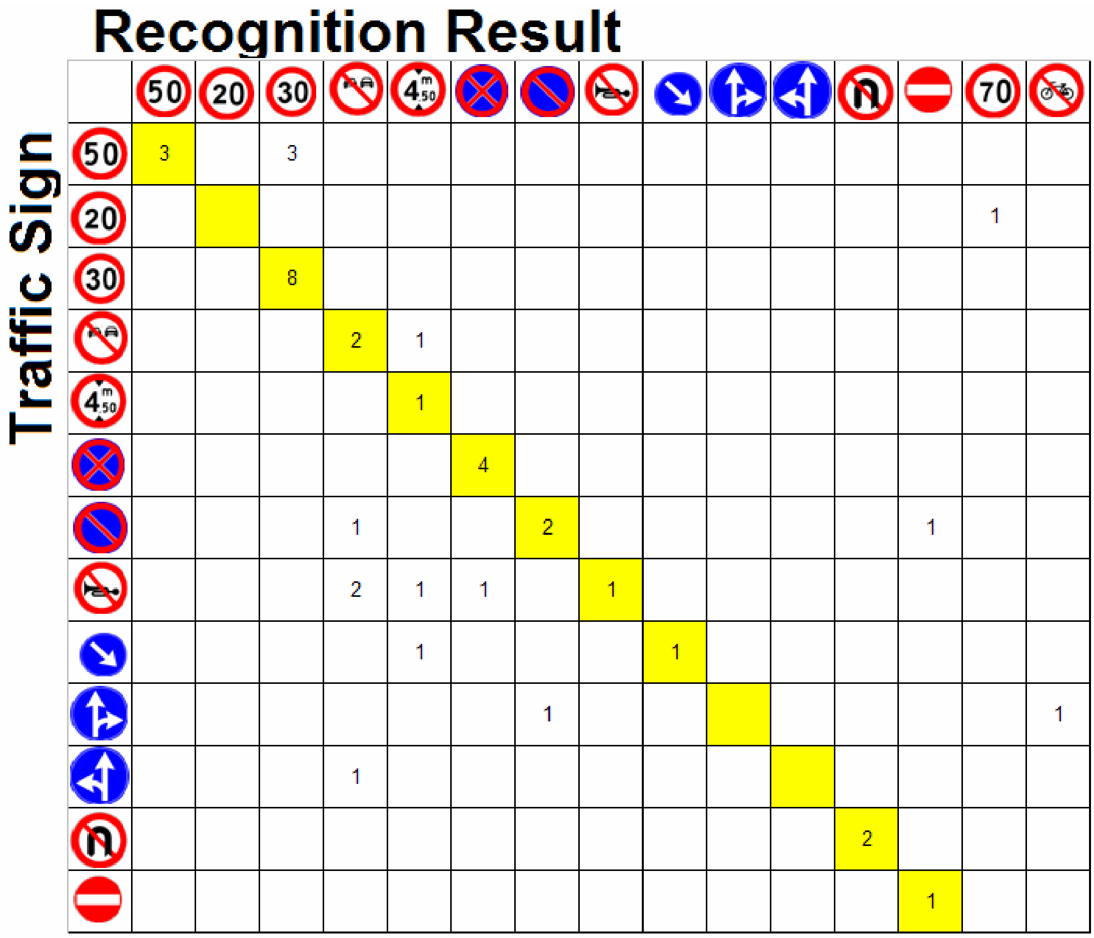


Figure 5.61: Confusion matrix for Table 5.40, for circular signs in Figure 5.60

Figure 5.62: Detected and recognized rectangular signs for parameter Set-1 and Threshold Set-1 (10 levels) for Table 5.40.

Table 5.41 Recognition Rates with Parameter Set-1 and Threshold Set-2 (5 level)

Shape	# of traffic signs among the detected objects	# of Recognizied Traffic Signs	Recognition Rate (%)
Triangular	37	36	97,2
Circular	37	21	56,8
Rectangular	15	11	73,3

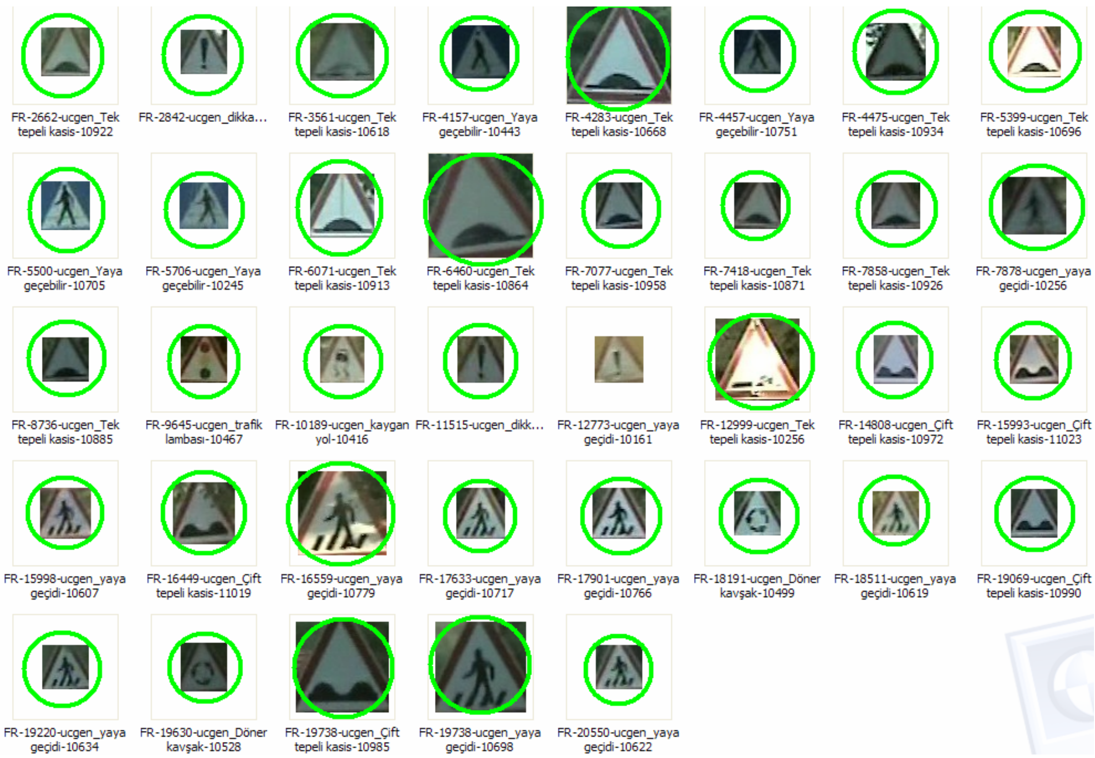


Figure 5.63: Detected and recognized triangular signs for parameter Set-1 and Threshold Set-2 (5 levels) for Table 5.41.

Figure 5.64: Detected and recognized circular signs for parameter Set-1 and Threshold Set-2 (5 levels) for Table 5.41.

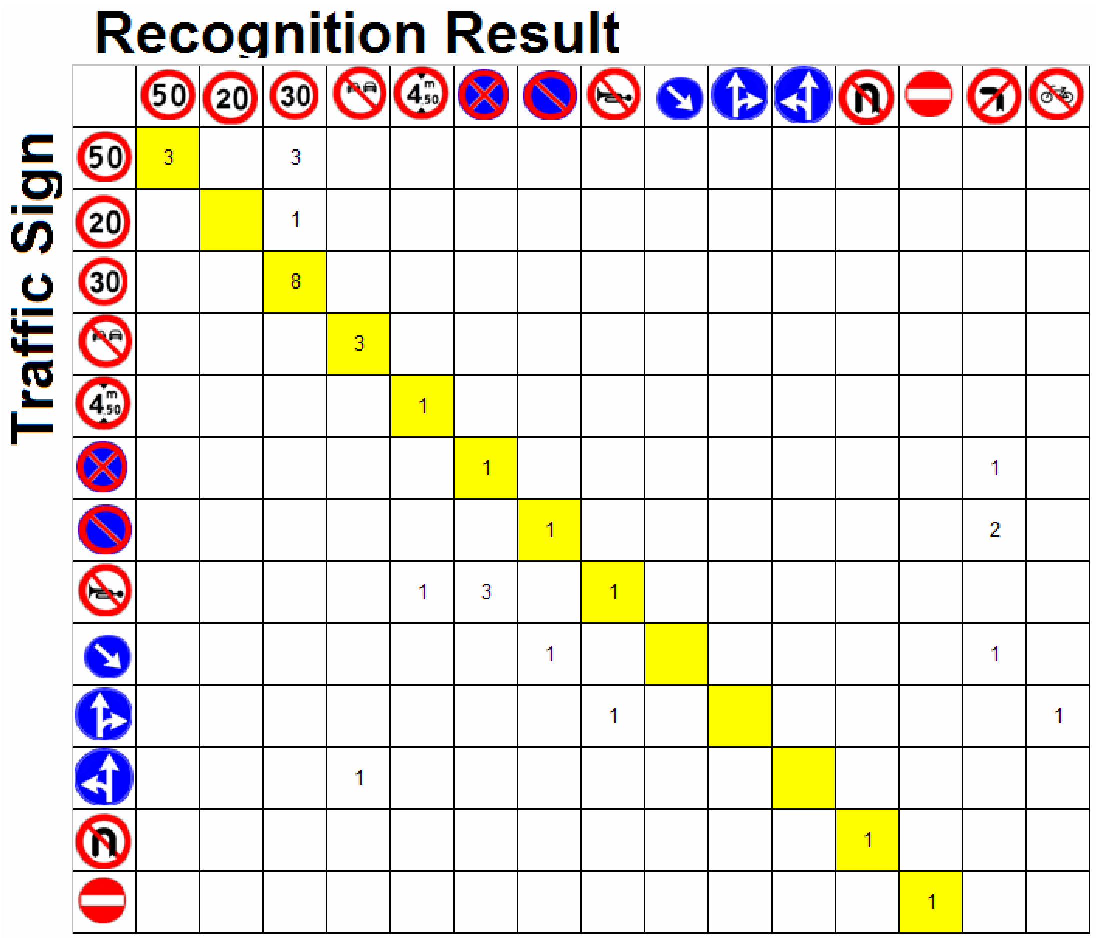


Figure 5.65: Confusion matrix for Table 5.41, for circular signs in Figure 5.64

Figure 5.66: Detected and recognized rectangular signs for parameter Set-1 and Threshold Set-2 (5 levels) for Table 5.41.

Table 5.42 Recognition Rates with Parameter Set-1 and Threshold Set-3 (3 level)

Shape	# of traffic signs among the detected objects	# of Recognized Traffic Signs	Recognition Rate (%)
Triangular	31	31	100,0
Circular	10	8	80,0
Rectangular	13	10	76,9

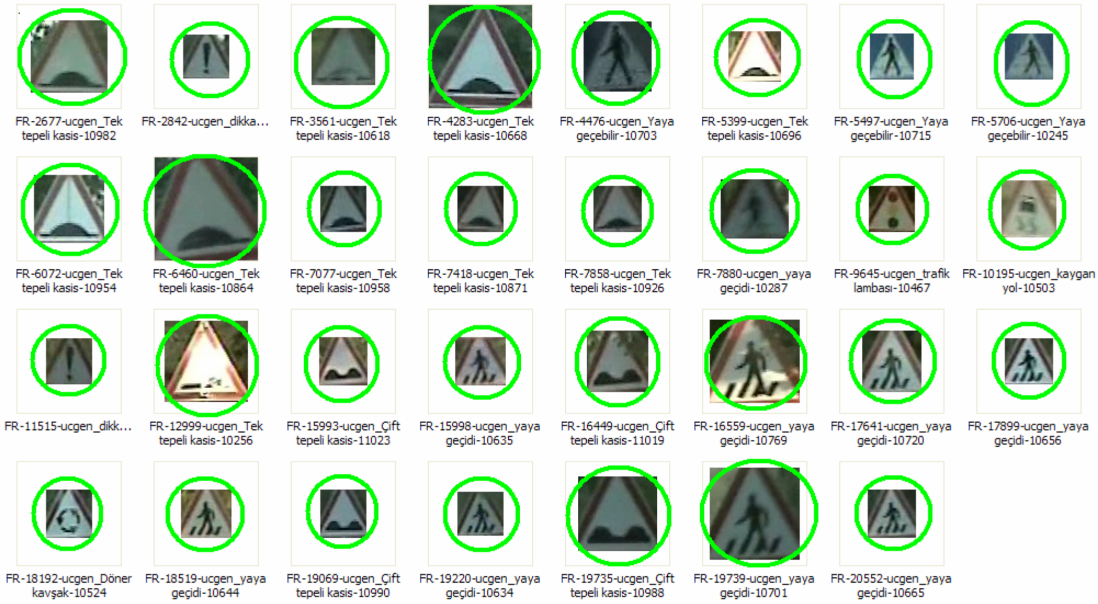


Figure 5.67: Detected and recognized triangular signs for parameter Set-1 and Threshold Set-3 (3 levels) for Table 5.42.

Figure 5.68: Detected and recognized circular signs for parameter Set-1 and Threshold Set-3 (3 levels) for Table 5.42.

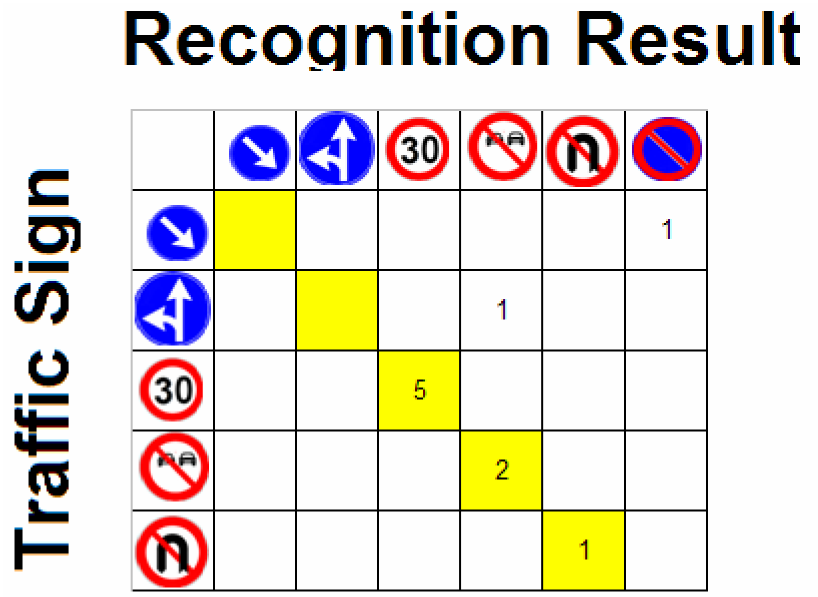


Figure 5.69: Confusion matrix for Table 5.42, for circular signs in Figure 5.68

Figure 5.70: Detected and recognized rectangular signs for parameter Set-1 and Threshold Set-3 (3 levels) for Table 5.42.

CHAPTER 6

CONCLUSION

In this thesis, detection and recognition of traffic signs were studied. The detection and recognition processes were carried out in different time intervals of day, and while detecting, mainly shape features were utilized. Color information was used as an auxiliary method for circular traffic signs, which improved the detection rate on day-time taken video streams.

For five set of video streams with different set of parameters were experimented for detection in order to compare which improve results best.

In detection stage, Canny's edge detector [4] and binary threshold were used to extract edges. Contour retrieving algorithm of Suzuki and Abe [50] were used to get contour and Douglas-Peucker [53] algorithm was utilized to realize polyline approximation. After detection stage, the most successful detection rate occurred for triangular signs for most of the parameters according to other signs. The main reason for success of triangular sign detection might be caused from two facts: First, triangular shapes except for traffic signs can be seen more rarely in nature and more constraint can be imposed to detect triangular signs without increasing number of false positives. Second, backgrounds of the triangular signs are white and that means that there are more triangular signs with sharper contrast than rectangular and circular traffic signs, which enhances the detection rate of triangular signs based on shape by contour retrieving. False positive detection rate for rectangular signs is the highest among others. The main reason might be the fact that, especially in urban areas, there are many rectangular objects such as advertisement Tables, windows, doors etc.

When comparing the performance of detection according to Douglas-Peucker [53] threshold and binary threshold, generally, lower Douglas-Peucker [53] threshold causes less detection, and higher Douglas-Peucker [53] threshold results in high detection of traffic signs (rectangular and triangular) but increases false detection as well. Moreover higher Douglas-Peucker [53] threshold values increase detection of traffic signs especially night time with false detections. However in day time there is no great improvement for detection of traffic signs, but there is a great increase in false detection, which reduces the performance of detection.

In recognition stage, template based methods were used to classify traffic signs. Recognition rate was the highest for triangular signs. This may be caused from two facts: First, compared with rectangular signs, triangular signs include certain ideogram as an informative part and ideogram can be extracted quite well. So recognition rate of triangular signs is higher than that of rectangular one. Second, compared with circular signs, triangular signs have certain corner points and thus fits the template more exactly than circular one, which increases the recognition rate of triangular signs. Generally speaking, recognition with template matching performance is sensitive to illumination. For this reason recognition performance for night time is less than that of day time.

6.1 Future Works

In order to improve the performance of detection phase, some tracking algorithms can be implemented. This improvement includes reducing detection of false positives, detection of more rectangular and circular signs. The detection is based on shape feature extraction by contour retrieving, and contours are acquired by Canny

[4] and threshold functions. Threshold fails to detect traffic/road sign candidate in front of bright background. If Canny [4] edge detector can not extract the border in bright background, then traffic/road sign can not be detected. This problem may be solved by suppressing bright background. Moreover, to improve recognition rate of circular sign, and speed up the system, circular sign can be localized more certain to fit template well. Another improvement can be realized on the speed of the detection. As the number of binary threshold levels increase, the speed of detection tends to slow down. Although increasing the number of binary threshold contributes the detection in general, it sometimes does not contribute the true detection much especially for low bright images but contributes cost. In order to overcome the cost of computation for unnecessary binary levels, histogram may be utilized and threshold levels can be selected dynamically.

REFERENCES

[1] G. Piccioli, E. De Michelli, P. Parodi, M. Campani, "A Robust Method for Road Sign Detection and Recognition," Image and Vision Computing, vol. 14, pp.209-223, 1996.

[2] "Road Sign Recognition Survey" [On-line] Available: http://euler.fd.cvut.cz/research/rs2/files/skoda-rs-survey.html

[3] C. Little, "The Intelligent Vehicle Initiative: Advancing Human-Centered Smart Vehicle," US Department of Transportation Federal Highway Administration, [On-line] Available: http://www.tfhrc.gov/pubrds/pr97-10/p18.htm

[4] J. F. Canny, "A computational Approach to Edge Detection," IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. 8, No. 6, pp. 679-698, Nov. 1986.

[5] R. Janssen, W. Ritter, F. Stein, and S. Ott. "Hybrid Approach for Traffic Sign Recognition," In Proc. of Intelligent Vehicles Conference, pp 390-395, 1993.

[6] M. A. Garcia, M. A. Sotelo, E. M. Gorostiza. "Emerging Technologies and Factory Automation," ETFA '03. IEEE Conference Volume 2, pp.212–215,16-19 Sept. 2003.

[7] A. De la Escalera, L. E. Moreno, M. A. Salichs, J. M. Armigol, "Road Traffic Sign Detection and Classification," Industrial Electronics, IEEE Transactions on," Vol. 44, No. 6., pp. 848-859, 1997.

[8] W. G. Shadeed, D. I. Abu-Al-Nadi, and M. J. Mismar, "Road Traffic Sign Detection in Color Images," 10th IEEE International Conference on Vol. 2, pp. 890-893, 2003.

[9] N. Yabuki, Y. Matsuda, Y. Sumi, Y. Fukui, S. Miki, “Active Net Control for Traffic Signs Detection,” Vehicle Electronics Conference, Proceedings of the IEEE International, pp. 151-155, Sept. 2001.

[10] W. Ritter, “Traffic Sign Recognition in Color Image Sequences,” Proceedings of the Intelligent Vehicles '92 Symposium, pp. 12-17, 1992.

[11] L. Pacheco, J. Batlle, X. Cufi, “A New Approach to Real Time Traffic Sign Recognition Based on Colour Information,” Proceedings of the Intelligent Vehicles '94 Symposium, pp. 339-344, Oct. 1994.

[12] H. Liu, D. Liu, J. Xin, “Real-time Recognition of Road Traffic Sign in Motion Image Based on Genetic Algorithm,” Machine Learning and Cybernetics Proceedings, Volume 1, pp. 83-86, Nov. 2002.

[13] C.Y. Fang, S. W. Chen, C. S. Fuh, “Road-Sign Detection and Tracking,” Vehicular Technology, IEEE Transactions on Volume 52, Issue 5, pp. 1329-1341, Sept. 2003.

[14] N. Kehtarnavaz, A. Ahmad, “Traffic Sign Recognition in Noisy Outdoor Scenes,” Proceedings of the Intelligent Vehicles '95 Symposium, pp. 460-465, Sept. 1995.

[15] S. W. Lu, “Recognition of Traffic Signs Using a Multilayer Neural Network,” Electrical and Computer Engineering, 1994. Conference Proceedings, 1994 Canadian Conference on, Volume 2, pp. 833-834, 1994.

[16] L. Priese, R. Lakmann, V. Rehrmann, “Ideogram Identification in a Real-time Traffic Sign Recognition System,” In Proc. Intelligent Vehicles Symp., pp. 310-314, 1995.

[17] Y.J. Zheng, W. Ritter, and R. Janssen, "An Adaptive System for Traffic Sign Recognition," Intelligent Vehicles Symposium, Paris, 1994.

[18] D. Ghica, S.W. Lu, X. Yuan ”Recognition of Traffic Signs by Artificial Neural Network,” 1995 IEEE International Conf. on Neural Networks, Perth, Australia, vol.3, pp.1444-1449, Nov. 1995.

[19] Y. Aoyagi, T. Asakura, “A Study on Traffic Sign Recognition in Scene Image Using Genetic Algorithms and Neural Networks,” Proc. IEEE

IECON Int. Conf. on Industrial Electronics, Control, and Instrumentation, Taipei, Taiwan, vol. 3, pp. 1838-1843, 1996.

[20] S**.** EsTable, J. Schick, F. Stein, R. Janssen, R. Ott, W. Ritter, Y. J. Zheng, "A Real Time Traffic Sign Recognition System," in proc. Intelligent Vehicles'94, 1994.

[21] L. Estevez, N. Kehtarnavaz, "A Real-Time Histographic Approach to Road Sign Recognition," Proceedings of the IEEE Southwest Symposium on Image Analysis and Interpretation, pp. 95-100, 1996.

[22] J. A. Janet, M. W. White, T. A. Chase, R. C. Luo, J. C. Sutto, "Pattern Analysis for Autonomous Vehicles with the Region- and Feature-Based Neural Network: Global Self-Localization and Traffic Sign Recognition," Proc. IEEE Int. Conf. on Robotics and Automation, Minneapolis, Minnesota, vol. 4, pp. 3598-3604, 1996.

[23] D. S. Kang, N. C. Griswold, N. Kehtarnavaz, "Image Analysis and Interpretation," Proceedings of the IEEE Southwest Symposium on Volume , pp. 88–93, Apr. 1994.

[24] M. Lalonde, Y. Li, "Detection of Road Signs Using Color Indexing," Technical Report CRIM-IT-95/ 12-49, Centre de Recherche Informatique de Montreal, 1995.

[25] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, Addison-Wesley, 1993.

[26] H. Kamada and M. Yoshida, "A Visual Control System Using Image Processing and Fuzzy Theory," in Vision Based Vehicle Guidance, I. Masaki, Ed. Berlin, Germany: Springer-Verlag, 1992, pp. 111–128.

[27] B. Mertsching, H. Austermeier, U. Bilker, G. Hartmann, "Interpretation of Traffic Scenes Using A Hierarchical Data Structure," Intelligent Vehicles '93 Symposium, pp. 163-168, 1993.

[28] H. Fleyeh, "Color Detection and Segmentation for Road and Traffic Signs," Cybernetics and Intelligent Systems, 2004 IEEE Conference on, vol. 2, pp. 809-814, 2004.

[29] C. Bahlmann, Y. Zhu, V. Ramesh; M. Pellkofer, T. Koehler, “A System for Traffic Sign Detection, Tracking, and Recognition Using Color, Shape, and Motion Information,” Intelligent Vehicles Symposium, 2005 Proceedings, pp. 255-260, 2005.

[30] J. Miura, T. Kanda, Y. Shirai, “An Active Vision System for Real-Time Traffic Sign Recognition,” 2000 IEEE Intelligent Transportation Systems Conference Proceedings, pp. 52-57, Oct. 2000.

[31] S. Vitabile, and F. Sorbello, "Pietogram Road Signs Detection and Understanding in Outdoor Scenes," presented at Proc. Conf. Enhanced and Synthetic Vision, Orlando, Florida, 1998.

[32] P. Paclik, J. Novovicova, P. Pudil, P. Somol, “Road Sign Classification Using Laplace Kernel Classifier,” selected papers from the 11th Scandinavian conference on image analysis, pp. 1165-1173, 2000.

[33] S. Vitabile, G. Pollaccia, G. Pilato, F. Sorbello, “Road Signs Recognition Using a Dynamic Pixel Aggregation Technique in the HSV Color Space,” Proceedings of the 11th International Conference on Image Analysis and Processing, pp. 572-577, 2001.

[34] S. Vitabile, S. Gentile, F. Sorbello, “A Neural Network Based Automatic Road Signs Recognizer,” Neural Networks, 2002. IJCNN '02. Proceedings of the 2002 International Joint Conference on, Vol. 3, pp. 2315-2320, 2002.

[35] A. de la Escalera, J. Armingol, and M. Mata, "Traffic Sign Recognition and Analysis for Intelligent Vehicles," lmage and Vision Comput., Vol. 21, pp. 247-258, 2003.

[36] G. Mo, Y. Aoki, “Recognition of Traffic Signs in Color Images,” 2004 IEEE Region 10 Conference, Volume B, pp. 100-103, 2004.

[37] J. Plane, Traffic Engineering Handbook, Prentice Hall, 1992.

[38] Trafik İşaretleri El Kitabı, Bakım Dairesi Başkanlığı, Trafik Şubesi Müdürlüğü, 2004.

[39] A. Hanbury and J. Serra, “A 3D-polar Coordinate Colour Representation Suitable for Image Analysis,” Technical Report PRIP-TR-77, PRIP, T.U. Wien, 2002.

[40] J. Angula, J. Serra, "Color Segmentation by Ordered Merging," IEEE Image Processing, International Conference on, Vol. 2, pp. II-125-128, 2003.

[41] G. Loy, N. Barnes, "Fast Shape-Based Road Sign Detection for a Driver Assistance System," Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on," Vol. 1, pp. 70-75, 2004.

[42] M. Benallal, J. Meunier, "Real-Time Color Segmentation of Road Signs," Electrical and Computer Engineering, 2003. IEEE CCECE 2003. Canadian Conference on, Vol. 3, pp. 1823-1826, 2003.

[43] A. de la Escalera, J. M. Armingol, M. A. Salichs, "Traffic Sign Detection for Driver Support Systems," International Conference on Field and Service Robotics, 2001.

[44] Y. Aoyagi, T. Asakura, "A Study on Traffic Sign Recognition in Scene Image Using Genetic Algorithms and Neural Networks," Industrial Electronics, Control, and Instrumentation, 1996., Proceedings of the 1996 IEEE IECON 22nd International Conference on, Vol. 3, pp. 1838-1843, 1996.

[45] A. Etemadi, "Robust Segmentation of Edge Data," IEEE Image Processing Conference, Maastricht, 1992.

[46] P. Hough, "A Method and Means for Recognizing Complex Patterns". US Patent, 3069654, 1962.

[47] S. Masciangelo, "3-D cues from a Single View: Detection of Elliptical Arcs and Model Based Perspective Back Projection," Proc. British Machine Vision Conference '90, pp. 223-228, Oxford, 1990.

[48] K. S . Jr, S. L. Tanimoto, "Progressive Refinement of Raster Images," IEEE Transactions on Computers, pp. 871-874, Nov. 1979.

[49] P. J. Burt, E. H. Adelson, "The Laplacian Pyramid as a Compact Image Code," IEEE Transactions on Communications, VOL. COM-3l, NO. 4, April 1983.

[50] S. Suzuki and K. Abe. "Topological Structural Analysis of Digital Binary Images by Border Following," CVGIP, Volume 30(1), pp:32-46, 1985.

[51] Andrew W. Fitzgibbon, R.B.Fisher. “A Buyer’s Guide to ConicFitting,” Proc.5th British Machine Vision Conference, Birmingham, pp. 513-522, 1995.

[52] D. H. Douglas and T. K. Peucker. “Algorithms for the Reduction of the Number of Points Required to Represent a Digitized Line or its Caricature”, Canadian Cartographer, 10, 112-122, 1973.

[53] A. Fitzgibbon, M. Pilu and R. B. Fisher. “Direct Least Square of Ellipses,” Pattern Analysis and Machine Intelligence, Vol. 21, No. 5, May 1999.

[54] E. Y. Kang, I. Cohen, G. Medioni. “Non-Iterative Approach to Multiple 2D Motion Estimation,” Proceedings of the 17^{th} International Conference on Pattern Recognition, 2004.

[55] J. S. Lim, “Two Dimensional Signal & Image processing,” Prentice Hall, 1990.

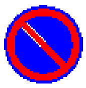
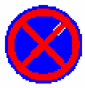
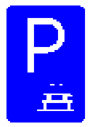
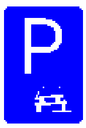
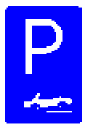
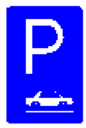
[56] R. O. Duda, P. E. Hart, D. G. Stork, “Pattern Classification (2^{nd} edition),” Wiley, New York 2001.

[57] A. Goshtasby, S. H. Gage, J. F. Bartholic, “A Two-Stage Cross-Correlation Approach to Template Matching,” IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. 6, no. 3, pp. 374-378, 1984.

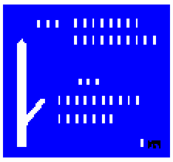
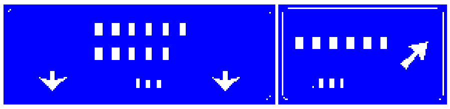
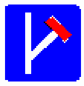
APPENDIX A

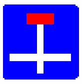
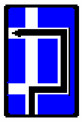
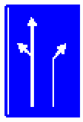
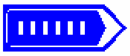
Appendix A contains some examples of traffic signs used in Turkey [38]. First group consists of examples of parking signs. Second group consists of examples of information signs. Third group consists of examples of danger warning signs. Fourth group consists of examples of regulation signs.

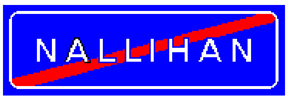
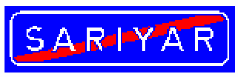
DURMA VE PARKETME İŞARETLERİ

 (P-1) Parketmek Yasaktır	 (P-2) Duraklama ve Parketme Yasaktır	 (P-7a) Park Yeri	 (P-7b) Park Yeri
 (P-7c) Park Yeri	 (P-7d) Park Yeri	 (P-7e) Park Yeri	

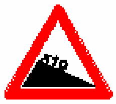
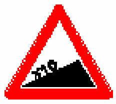
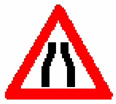
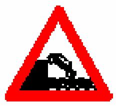
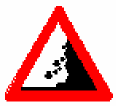
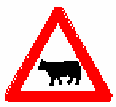
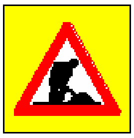
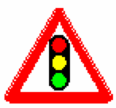
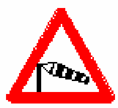
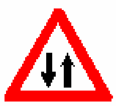
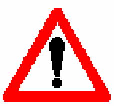
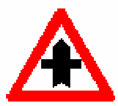
BİLGİ İŞARETLERİ

 (B-1a) Kavşak Öncesi Yön Levhası	 (B-1b) Kaplama Üstü Yön Levhası	 (B-2a) Girişi Olmayan Yol Kavşağı

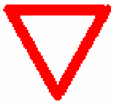
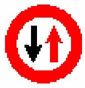
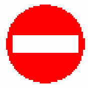
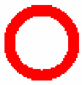
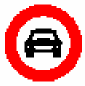
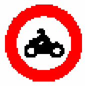
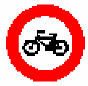
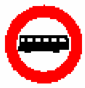
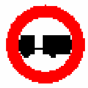
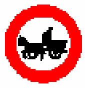
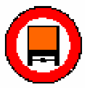
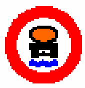
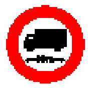
 (B-2b) Girişi Olmayan Yol Kavşağı	 (B-2c) Girişi Olmayan Yol Kavşağı	 (B-3) İlerideki Kavşakta Sola Dönüş Yasağını Gösteren Yön Levhası	 (B-4) Kavşak Öncesi Şerit Seçimi Levhası
 (B-5a) Kavşak İçi Yön Levhası	 (B-5b) Kavşak İçi Yön Levhası (Turistik Mahal)	 (B-6) Kavşak İçi Yön Levhası (Havaalanı-Havalimanı)	 (B-7) Kavşak İçi Yön Levhası (Kamp Yeri)

 (B-8a) Türkiye Devlet Sınırı Levhası	 (B-8b) İl Sınırı Levhası	 (B-8c) Türkiye Hız Sınırları Levhası
 (B-9a) Meskun Mahal (İl Merkezi) Levhası	 (B-9b) Meskun Mahal (İlçe Merkezi) Levhası	 (B-9c) Meskun Mahal (Köy) Levhası
 (B-10a) Meskun Mahal Sonu İşareti Levhası (İl Merkezi)	 (B-10b) Meskun Mahal Sonu İşareti Levhası (İlçe Merkezi)	 (B-10c) Meskun Mahal Sonu (Köy)

TEHLİKE UYARI İŞARETLERİ

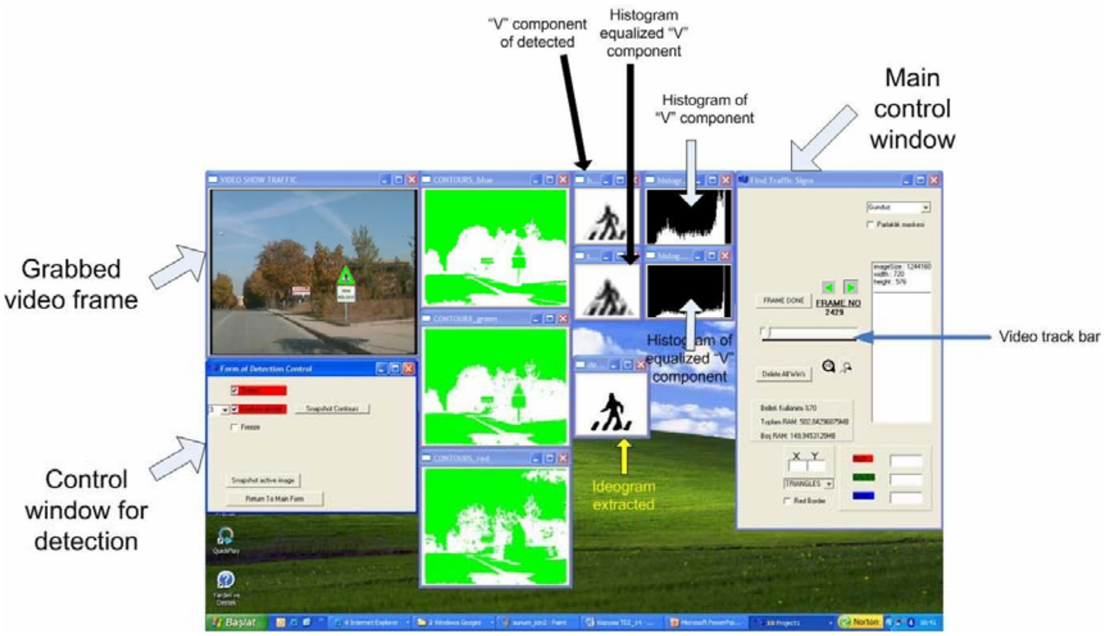
 (T-1a) Sağa Tehlikeli Viraj	 (T-1b) Sola Tehlikeli Viraj	 (T-2a) Sağa Teh. Devamlı Virajlar	 (T-2b) Sola Teh. Devamlı Virajlar
 (T-3a) Tehlikeli Eğim (İniş)	 (T-3b) Tehlikeli Eğim (Çıkış)	 (T-4a) İki Taraf Daralan Kaplama	 (T-4b) Sağdan Daralan Kaplama
 (T-4c) Soldan Daralan Kaplama	 (T-5) Açılan Köprü	 (T-6) Deniz-Nehir Ken. Biten Yol	 (T-7) Kasisli Yol
 (T-8) Kaygan Yol	 (T-9) Gevşek Malzemeli Zemin	 (T-10) Gevşek Şev	 (T-11) Yaya Geçidi
 (T-12) Çocuklar Geçebilir	 (T-13) Bisiklet Geçebilir	 (T-14a) Ehli Hayvanlar Geçebilir	 (T-14b) Vahşi Hayvanlar Geçebilir
 (T-15) Yolda Çalışma	 (T-16) Işıklı İşaret Cihazı	 (T-17) Havaalanı (Alçak Uçuş)	 (T-18) Yandan Rüzgar
 (T-19) İki Yönlü Yol	 (T-20) Dikkat	 (T-21) Kontrolsüz Kavşak	 (T-22a) Anayol-Tali Yol Kavşağı

TRAFİK TANZİM İŞARETLERİ

 (TT-1) Yol Ver	 (TT-2) Dur	 (TT-3) Karşıdan Gelene Yol Ver	 (TT-4) Taşıt Giremez
 (TT-5) Taşıt Trafiğine Kapalı Yol	 (TT-6) Motosiklet Hariç Taşıt Trafiğine Kapalı Yol	 (TT-7) Motosiklet Giremez	 (TT-8) Bisiklet Giremez
 (TT-9) Mopet Giremez	 (TT-10a) Kamyon Giremez	 (TT-10b) Otobüs Giremez	 (TT-11) Treyler Giremez
 (TT-12) Yaya Giremez	 (TT-13) At Arabası Giremez	 (TT-14) El Arabası Giremez	 (TT-15) Tarım Traktörü Giremez
 (TT-16a) Belirli Miktarlardan Fazla Patlayıcı ve Parlayıcı Madde Taşıyan Taşıt Giremez	 (TT-16b) Tehlikeli Madde Taşıyan Taşıt Giremez	 (TT-17) Belirli Miktarlardan Fazla Su Kirletici Madde Taşıyan Taşıt Giremez	 (TT-18) Motorlu Taşıt Giremez
 (TT-19) Taşıt Giremez	 (TT-20) Genişliği ..... Metreden Fazla Olan Taşıtlar Giremez	 (TT-21) Yüksekliği .... Metreden Fazla Olan Taşıt Giremez	 (TT-22) Uzunluğu ..... Metreden Fazla Olan Taşıt Veya Katar Giremez

APPENDIX B

The platform used for traffic sign detection and recognition was developed on Borland C++ 6.0 with utilizing openCV library.



Graphical user interface can be used to command software to acquire a frame of video either one by one or as a stream. Once the frame is retrieved, detection and recognition process can be started by simply selecting "detect" check-box. The detected signs are marked by green color on "Video Show Traffic" window. Found contours can be seen by selecting "contour show" check-box. Histograms of image before and after histogram equalization, ideogram extracted (if any) are shown automatically upon detection. Recognition results are reported on the screen.