

REPUBLIC OF TURKEY
FIRAT UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES



TRAFFIC SIGN DETECTION AND RECOGNITION

MASTER THESIS

SARDAR OMAR RAMADHAN

Supervisor: Assoc. Prof. Dr. Burhan ERGEN

May - 2017

REPUBLIC OF TURKEY
FIRAT UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

TRAFFIC SIGN DETECTION AND RECOGNITION

MASTER THESIS
SARDAR OMAR RAMADHAN
151129120

Submission Date to the Institute:
Thesis Presentation Date:

(Title of Juror, Name and L/Name and institution):

Sign

Supervisor: **Assoc. Prof. Dr. Burhan ERGEN**

/ **Firat Uni. / Comp. Eng.**



Other Jury Members: **Prof. Dr. Ali KARCI**

/ **İnönü Uni. / Comp. Eng.**



Other Jury Members: **Assoc. Prof. Dr. Taner TUNCER**

/ **Firat Uni. / Comp. Eng.**



May - 2017

ACKNOWLEDGEMENT

I would like to express my highest gratitude to my project supervisor [Assoc. Prof. Dr. Burhan ERGEN] for the right guidance and courage given from the start of my MSc. Project work to the last. My great appreciation goes to my family who has supported me all these year. Their love and motivation provides me the spirit to finish this thesis successfully.

I also would like to express my gratitude to all academic staff of Computer Engineering department in the University of Firat and my special thanks to my friends that were always beside me and never stopped giving the motivation specially my dear friend [Dilovan Muhsin HAJI].

JUNE-2017

SARDAR OMAR
RAMADHAN

LIST OF CONTENTS

ACKNOWLEDGEMENT	I
LIST OF CONTENTS	II
LIST OF FIGURES.....	IV
LIST OF TABLES.....	VI
ABBREVIATIONS.....	VII
OZET	VIII
ABSTRACT	IX
1. INTRODUCTION	1
1.1. Problem Formulation and Goals.....	1
1.2. Thesis Objectives	3
2. LITERATURE REVIEW	4
2.1. Visual Object Detection	4
2.2. Object Recognition.....	6
2.3. Background of Traffic Sign Recognition	7
2.4. Evaluation of Previous Studies.....	8
3. METHODOLOGY	10
3.1. Detection	10
3.1.1. Color Segmentation.....	12
3.1.2. Image Binarization and Region Labeling.....	17
3.1.3. Region Analysis	19
3.1.4. RIO Extraction	20
3.2. Classification.....	23
3.2.1. Shape Identification for Square and Triangle.....	24
3.2.2. Circle Shape Identification	25
3.2.3. Classification of Traffic Sign	28
3.3. Recognition	29
3.3.1. Pictogram Extraction.....	32
3.3.2. Connect Regions.....	34
3.3.3. Curvature Scale Space (CSS).....	35
4. EXPERIMENTAL RESULTS	39
4.1. Detection Results.....	40
4.2. Recognition Results.....	41

4.3.	Overall Results	43
5.	CONCLUSIONS	45
	REFERENCES	46
	CURRICULUM VITA	51



LIST OF FIGURES

Figure 1.1:	Usage scenario of a recognition of traffic sign system: a) schematic depiction of a vehicle approaching a traffic sign, b) an example way of presenting the information about a detected and recognized sign to the driver. The right illustration by courtesy of Siemens AG.....	2
Figure 3.1:	Flowchart of proposed system	10
Figure 3.2:	Blue and red color used on: (a) – Informative plates; (b) – Buildings; (c) – Advertising; (d) – Traffic lights; (e) – Car body paint and car headlamps; (f) – Clothes.	11
Figure 3.3:	Flowchart of Detection	12
Figure 3.4:	(a) Functions for detection of red and blue image areas; (b) Function for detection of color saturated areas	13
Figure 3.5:	Example of the color segmentation process	14
Figure 3.6:	Range of colors selected from the input image, after color segmentation, (a) for red color; (b) for blue color	15
Figure 3.7:	Color segmentation for a real photo situation.....	15
Figure 3.8:	Example of the new color segmentation process	17
Figure 3.9:	Range of colors selected from the input image, after color segmentation, (a) For red color; (b) for blue color	17
Figure 3.10:	Binarization.....	18
Figure 3.11:	Binarization.....	18
Figure 3.12:	(a) Input image containing two red signs; (b) Respective binarized image.....	19
Figure 3.13:	Sign basic shape illustration.....	20
Figure 3.14:	ROI extraction example	22
Figure 3.15:	Binarization Example.....	22
Figure 3.16:	Problems on filling ‘holes’	22
Figure 3.17:	Circular and elliptical shapes	23
Figure 3.18:	Shapes with vertices.....	24
Figure 3.19:	For corner occurrence regions tested	24
Figure 3.20:	Circle rotation invariance.....	26
Figure 3.21:	Circle classification process.....	26
Figure 3.22:	Correctly classified signs except <i>STOP</i> sign	29
Figure 3.23:	Signs with similar content.....	30
Figure 3.24:	Similar pictograms with distinguishable contours	31
Figure 3.25:	Flowchart of recognition.....	31

Figure 3.26:	Pictogram color. (a) – Red signs: pictogram contains black and red colors; (b, c and d) – Blue signs: pictograms contain black, red and white colors.....	32
Figure 3.27:	Red detected regions	33
Figure 3.28:	Blue detected regions	33
Figure 3.29:	Red component of RGB color space.....	33
Figure 3.30:	Five real photo signs	33
Figure 3.31:	Red detected regions	34
Figure 3.32:	Blue detected regions.....	34
Figure 3.33:	Region connection idea (a) – A roundabout sign (b) – Roundabout extracted pictogram (c) – Curve encasing the pictogram (d) – Curve deformed according to pictogram shape	35
Figure 3.34:	The influence of the starting point on the CSS image [35]. (a) Re-sampled outer contour with ‘x’ marking the starting point, (b) CSS image, (c) CSS peaks	38
Figure 4.1:	Original image with its corresponding grayscale and binary image.....	39
Figure 4.2:	Detected sign using blob detection method	40
Figure 4.3:	Correctly detected example.....	41
Figure 4.4:	Signs database	42
Figure 4.5:	Influence of sign definition in the recognition result.....	43

LIST OF TABLES

Table 3.1:	Square and triangle classification results	25
Table 3.2:	Circle classification results.....	28
Table 3.3:	Traffic sign classification into the considered classes	29
Table 4.1:	Detection sign results	41
Table 4.2:	Recognition results for database signs	42
Table 4.3:	Recognition results for real signs using successfully detected and recognized signs	43
Table 4.4:	Recognition results for real signs	44



ABBREVIATIONS

CSS	: Curvature Scale Space
FRS	: Fast Radial Symmetry
GPS	: Global Positioning System
GVF	: Gradient Vector Flow
HSV	: Hue-Saturation-Value color space
LRC	: Lower Right Corner
PROMETHEUS	: Programme for a European Traffic with Highest Efficiency and Unprecedented Safety
RGB	: Red-Green-Blue color space
ROI	: Regions of Interest
TLC	: Top Left Corner

OZET

Trafik İşaretlerini Bulma ve Tanıma Sistemi

Sürücülere navigasyon, güvenlik ve uyarı işaretleri gibi yol işaretleri ile önemli bilgiler verilebilir, bu nedenle bu günlerde otomatik işaretleme ve algılama için bilgisayar tabanlı sistem popülerdir. Trafik işareti algılama ve tanıma tasarımı, ulaşım sistemlerine yardımcı olmak için bilgisayar sistemleri kullanılabilir. Bir yolda trafik işaretlerinin bir otomatik algılama sistemi ile donatılmış kameralar gibi robotik yardım eklemek çok yararlı olacaktır. Bu araştırma tezi, trafik işaretini tanımak ve tespit etmek için bir genel bakış ve bazı teknikler sunar. Bu tezde, trafik işaretlerini trafik akışına özerk bir şekilde algılayıp algılayabilen bir Blob analizi yöntemi kullanan bilgisayarlı bir sistem oluşturulmuştur. Bu tez çalışmasında, uygulanan yaklaşımın güvenilirliği gösterilmiş ve deneysel sonuçları ile başarımı sunulmuştur.

Sonuç olarak aktif olarak kullanılan trafik işaretleri için mükemmel olmasa bile, daha az benzerlik gösteren farklı işaretler dizisi kullanmak tanıma oranını büyük ölçüde artırabilir. Gelecekteki taşımanın özerk araçlarla yapıldığını varsayarsak, kullanılan işaretler kesinlikle insan tarafından algılanabilmesi için değil kolay tespit edilebilir ve tanınabilir olacaktır

Anahtar kelimeler:Trafik İşareti Tanıma, Trafik işareti bulma, Renk sengemtasyonu, HSV, Blob analysis.

ABSTRACT

Important information can be provided to the drivers by the signs on the road like navigation, safety and warning thus computer based system for automatic recognition and detection of road sign grow popular these days. Designing a traffic sign detection and recognition use computer capabilities to aid the transportation systems can be very useful it is also possible to add robotic assistance like cameras that are equipped with an auto detection system for traffic signs on the road. This research thesis presents an overview and some technique for recognition and detection of traffic sign, in practice we have implemented Blob Analysis as our method for constructing a computerized system that can traffic signs recognize and detect in an autonomous manner from images that are fed to the system. In this paper we will explain how the implemented approach can be reliable and we will demonstrate the process and experimental results found in the process of implementation.

Even if the results are not perfect for the present traffic signs, using a different set of signs with fewer similarities between could greatly improve the recognition rate. Supposing that the future transportation is done with autonomous vehicles, certainly the used signs will be made to be easily detected computationally and not specifically for the human vision.

Keywords: Recognition of traffic sign, detection of traffic sign, color segmentation, HSV, blob analysis.

1. INTRODUCTION

Signs placed along the relative of roads to let drivers know significant information, this signs are called traffic signs or road signs. Traffic signs can be divided into four main types, type one: prohibition, type two: warning, type three: informative, type four: obligation. Depending on the color and the form, Signs of Prohibition are circles with a red border and a blue or white background. An equilateral triangle with one vertex upwards that is the signs of warning. They have are surrounded by a red border and a white background. Each one of the prohibition signs and warning signs, have a yellow background while they are placed in a side of public road. Signs of informative have the same colors which are blue background with the signs in circles, to indicate obligation. Finally, there are two exceptions: the stop sign, a hexagon and they are yield sign, and their shape is an inverted triangle; to detect and recognition a sign, two properties should been known which they are the shape and color of it.

Detection method of road sign have been increased in recent years researches. Mostly, colors are segmented by using shapes. For example, to limit the possible signs of an image the shape and color is used [1]. Then, by applying the cross-correlation technique, the look and edges of the triangular regions or circular beings are extracted and recognized the signs type. Redness is used in [2] to find and detect yield signs, they used tow signs: “do not enter” and "stop". Besides, they also used shape and edge analysis for detection and identifying the sign. At the same time, the author in [3] have used another technique, color matching starts with signs and corner detection, where the triangles, rectangles, or circles correspond to the specific corresponding to the previously identifying tables. A neural network used for classification by Yuille and colleagues [4], to detecting stop signs in their approach, corrected the brightness of the ambient light, matched the sign to a pre-parallel place before sign reading, and placed boundaries of signs.

1.1. Problem Formulation and Goals

The aim of the recognition of traffic sign system operating on board of a vehicle is to detect and recognition the sign instances over time and to correctly interpret their pictograms, so that the driver can react properly to the encountered traffic situation. The input to a TSR system is a live video stream captured by an in-vehicle camera/cameras and its output are the desired-form signals providing a human- understandable interpretation of

the detected and recognized signs. Such a system can be conceptually visualized using a block diagram with three main components. The arrows between each pair of components are drawn by default in both directions. However, depending on the actual system architecture, certain interactions may be unidirectional, or may not exist at all.

Think of a single front-view camera mounted on the front of a vehicle. Whenever a relevant traffic sign is detected within the field of view of the camera, as shown in Figure 1.1a, the TSR system should analyze the sign's pictogram over time, classify it, possibly before the sign is passed by, and present the outcome to the driver, for instance in a way shown in Figure 1.1b.

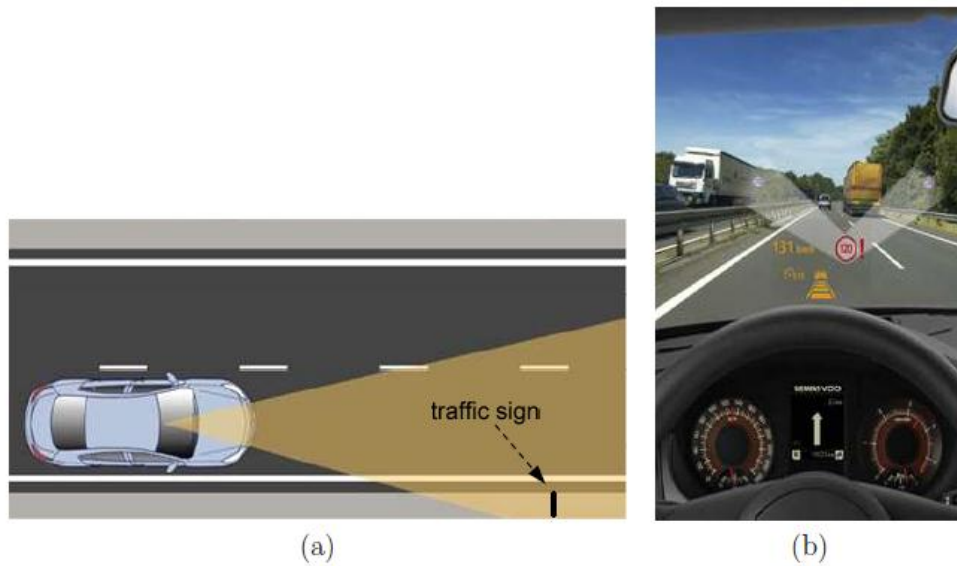


Figure 1.1: Usage scenario of a recognition of traffic sign system: a) schematic depiction of a vehicle approaching a traffic sign, b) an example way of presenting the information about a detected and recognized sign to the driver. The right illustration by courtesy of Siemens AG.

The goal of the road sign tracker is to maintain the track of an initially detected sign over time, until it disappears from the field of view of the camera. We want our tracker to go beyond the commonly adopted scheme in which it is only used to reduce the local search region for the sign detector. First, we would like to model the evolution of a tracked sign, or at least its boundary, on a feature or pixel level. Secondly, our goal is to develop a framework for modeling the full structure of the affine apparent transformations of the target in the image plane so that its frontal view can always be retrieved, regardless of the actual camera's viewpoint. Secondly, the desired tracker should be able to operate

efficiently in real time, in natural, possibly cluttered urban street scenes. To certain extent it should also be robust to illumination changes and ought to have an ability to retrieve the geometry of the target even when for several frame it remains occluded.

1.2. Thesis Objectives

An automated system of traffic sign recognition helps reduce the number of traffic accidents and is required for any autonomous vehicle project. Signs of traffic are designed to easily contrast with the background, so by the drivers can be detected. Many of the signs have red or blue Color tone with reflective attributes and also highly saturated properties, because they must be detected at various weather conditions. Traffic signs have also distinct shapes like circles, triangles, rectangles and octagons.

Although traffic sign recognition is an easy task for most of humans, it is still a challenge to perform in an automatic system, especially when low processing time is essential. Even supposing that a computer system able to correctly recognize 100% of the traffics signs exists, searching each possible sign along an image, would probably take more than the desired time to have the expected result, even using the fastest technology available today. If the exact sign location on images is available, it needs to be compared to the sign database and thus still be a time consuming process, because image comparison is usually a lengthy process. Any strategy that allows reducing the list of candidate signs, without taking too much time, helps improving the overall system performance.

Since the goal it is to known which signs appear in photos or video frames, as a first step it is required to know where the sign appears in the image. Most of the work done in this area relies on the color information to successfully detect signs on images. However, objects with the same color of the signs will also be identified as possible signs. This is one of the reasons why shape is also usually taken into account. Combining color and shape sign features it is possible to reduce the number of regions that could correspond to signs. Actually, with some exceptions, each combination of color and shape corresponds to a traffic sign class like prohibition, obligation, information and danger. This means that is possible to greatly reduce the number of possible sign comparisons, if color and shape is known. The possible signs are then compared to the database templates and a final recognition result is obtained.

2. LITERATURE REVIEW

Humans are faced with the problem of recognition in their everyday lives. Whether it concerns as intangible matter as disease diagnosis, or more perceivable one, like face identification, there seems to be no universal recipe for an always-working solution. In other words, every recognition problem is different. Regardless of the actual entity to be recognized, a natural approach involves identifying some number of features that make the object of interest distinguishable. It is usually possible to determine such distinctive features straightaway, based on the domain-level knowledge. However, sometimes the feature space is too large or the discriminative patterns in the data are not easy to spot. This requires aid of the advanced data analysis techniques from an area of data mining, statistical pattern recognition, machine learning, artificial intelligence and the related fields.

Pattern Recognition is a central term to the substance of our work. In this research a small area of pattern recognition, visual object detection, and recognition, is explored. This restricted area of interest is further reduced as we focus on a specific family of automotive machine vision applications those related to the detection and recognition of traffic signs from a moving vehicle. Certain methods presented in this work also address several related issues like human detection and car model classification, or are devised in a sufficiently generic way to enable applying them to a much broader class of machine vision problems. Due to the latter fact we find it necessary to set this research against the existing approaches to a broadly defined object detection and recognition.

2.1. Visual Object Detection

Visual object recognition is perhaps what comes to our minds most often when thinking about recognition in general. We recognize objects when trying to spot a colleague in a crowd, when reading a newspaper, or while driving a car. A similar sequence of actions is performed hundreds and thousands of times every day, when a human is trying to catch sight of a previously seen object in the environment, not necessarily on purpose, sometimes even subconsciously. When image and video processing became attainable on a computer machine, automation of these tasks was found a natural direction to follow. However, automatic object detection and recognition is still a difficult undertaking and only a limited progress has been made for over 30 years of

research in this area. It seems that the way humans do it is still far more efficient and effortless compared to most of the existing computer algorithms.

A robust object detector must be able to cope with the diversity of the imagery constituting the scene. Normally, unless further restricted by the problem constraints, actual image may depict the object of interest in a non-uniform, possibly cluttered background, in varying pose, scale and illumination, all of it affecting its appearance. In certain applications not only the target object has to be detected, but also its specific identity should be determined. The latter task, classification, is frequently even more of a challenge than the detection, especially when a large number of unique but similar instances of the same object category exist. An example of such a challenging problem is face recognition, e.g. [5, 6, 7] or fingerprint recognition [8]. Frequently, there is no clear distinction between the detection and the recognition, but even if they are performed sequentially, the latter may strongly depend on the former. In a video context, where a motion comes into play and the scene changes over time, also the third component becomes necessary, the target tracker. Depending on the actual problem characteristics, the tracking can be implemented at different levels to model solely the motion of the target, its temporal appearance variability, or both.

One popular approach to object detection is by first preprocessing the raw image in order to locate the salient regions likely containing the target objects. Further analysis is performed only in the found regions of interest (ROI). Discovery of such regions is typically performed by simple image filtering aimed at extracting from a raw image certain low-level features, like independent colour channel values, edges, or gradient directions/magnitudes. However, if this is insufficient, (ROI) finding can also be driven by a more advanced image segmentation technique, e.g. region growing [9, 10], clustering [11, 12], watersheds [13], background subtraction [14, 15] and many other. Actual choice of the segmentation method depends on the nature of the problem and the input feature space. For example in the case of traffic sign detection, a natural approach is to concentrate on the highly-contrasting regions of characteristic colours. In most surveillance applications focus is primarily put on those regions of the scene where temporal appearance changes are observed. In many cases, however, probability of the target object existence is relatively regularly distributed over the entire image, which renders an early segmentation void. Furthermore, in many complex systems such a sequential approach to

object detection may be a cause of problems. A failure in (ROI) detection at an early stage of the processing pipeline is irreversible, which makes the following components fail too. In such cases using dedicated, problem-specific focus operators is usually a workaround.

2.2. Object Recognition

Detection can be considered a special case of a classification problem with only two classes: the target and the background. In certain applications the problem to solve is defined such that only the anonymous instances of the target class need to be captured. A good example is a traffic monitoring system for vehicle counting or a human detector. In such systems the determination of the exact type or identity of the object is explicitly considered irrelevant, or for objective or technical reasons it cannot be found, e.g. because all instances of the target class look the same or the target is too far away from the camera. However, other problems clearly separate the detection from the classification. For example, a traffic sign recognition system operating on board of a moving vehicle must not only capture the sign instances in the scene, but also their exact types have to be determined in order to issue correct signals to the driver. Similarly, the contemporary visual access control systems not only detect human faces, but further, for verification, also match them to the database-stored pictures of the individuals with granted access to the guarded premises.

In general, visual object recognition approaches can be divided into discriminative and generative methods. In brief, the latter family of methods can be described as those that model the distribution of the image features, while the discriminative methods do not. More formally, the discriminative recognition models are a class of methods used for modeling the dependence of a target class variable c on an observed vector of features x . Within a statistical framework, this is done by parametrical modeling of the conditional probability distribution $p(c|x)$, which can be used for predicting class c from the observation x . The values of the unknown parameters are usually inferred from a set of labeled training data. This may be done by making maximum likelihood estimates of the parameters, or by computing distributions over the parameters in a Bayesian setting. Generative models on the other hand model the joint distribution $p(c, x)$ of image features and class labels. This is typically done by learning the class-conditional densities $p(x|c)$ and prior class probabilities $p(c)$, if they are not explicitly available. The required posterior probabilities required for classification are obtained using the Bayes' theorem:

$$p(c|x) = \frac{p(x|c)p(c)}{\sum_i p(x|ci)p(ci)} \quad (2.1)$$

The generative and the discriminative approaches to object recognition have different properties and complementary strengths and weaknesses. A good discussion of those is given by Ulusoy and Bishop [16]. Let us first introduce the state-of-the-art discriminative recognition methods.

One of the most common techniques of discriminative visual object recognition is template matching. This approach associates a real-valued function with each class and assigns such a label to the unknown observation that maximizes value of this function. Naturally, this function approximates the posterior probability $p(c|x)$ required for classification and the equal class prior's $p(c)$ are implicitly assumed. Matching is a very generic term, and in practice it may proceed in many ways, depending on the chosen matching criterion, image representation, and matching control strategy. The most popular criterion used is a normalized cross-correlation between the tested image and the template, understood as some function (e.g. inverse or exponential) of the image distance. The latter may be expressed with different metrics e.g. Euclidean, Manhattan, or Chamfer, and over different image representations, e.g. raw RGB pixels, gray-level values, or binaries values. In certain applications other, more complex image representations and distance metrics may be suitable for matching. For example distance transforms maps are convenient for contour matching. Hausdorff distance is particularly useful for shape comparison. Also Fourier matching in frequency space is possible and sometimes very useful, e.g. for image mosaicking. Unfortunately, pixel-wise image comparison is very sensitive to pose changes and computationally expensive. One possible solution to this problem is via a block-wise comparison with content averaging or histogramming. A more robust approach involves comparing the images in a pyramidal fashion. Such a hierarchical, multi-scale image analysis dramatically reduces the computational load as the unlikely class templates can be discarded at a very early stage. Their algorithms are reported to be fast and insensitive to noise and other disturbances.

2.3. Background of Traffic Sign Recognition

Road signs are an inherent part of the traffic infrastructure. They are designed to regulate flow of the vehicles, give specific information to the traffic participants, or warn against unexpected road circumstances. Road signs are always mounted in places that are

easy to spot by the drivers without distracting them from maneuvering the vehicle, e.g. on posts by the roadside or over the motorway lanes. Besides, their pictograms are designed in a way that admits easy discrimination between multiple signs, even from the considerable distance and under poor lightning and weather conditions. These properties of traffic signs have not changed for decades, with the exception of better, more durable and more reflective materials being used in the production process. In addition, new sign types are constantly introduced to reflect an inevitable technological advance in the traffic infrastructure and road safety standards.

Significant advances in the area of DSS were made in late 1980's and 1990's when numerous large-scale projects were developed in the USA, e.g. DARPA ALV PROgraMme 1, IVHS project 2, Europe, e.g. PROMETHEUS project 3, and Japan. The main contribution of these projects was popularization of the intelligent vehicle concept, which gave birth to the numerous academic and industrial automotive research groups. It would have probably never happened if it had not been for the technological progress of the 1990's which made it possible to operate a visual driver assistance system on board of a vehicle with the use of the off-the-shelf video cameras and cheap mobile computers. In parallel, numerous research papers were published and many international discussions on DSS were initiated. One of such discussions in Summer 1990 in Tokyo not only resulted in a valuable book by Masaki et al. [17] on vision-based vehicle guidance, but also started a series of annual Intelligent Vehicles Symposia' sponsored by the IEEE Industrial Electronics Society.

2.4. Evaluation of Previous Studies

Although the issue of traffic sign detection and recognition has been addressed for more than three decades, the number of recognized publications in this area is small compared to the number of those addressing many other computer vision problems like human detection or face recognition. This difference becomes even more striking when the numbers of commercial applications are to be compared. Whereas we can think about numerous worldwide operating access control systems based on face or fingerprint analysis, visual driver assistance only very recently went beyond the stage of a prototype, e.g. [18, 19]. Moreover, the existing industry-scale applications expose certain limitations, e.g. only restricted categories of traffic signs are handled.

In a vast majority of the existing road sign detection algorithms a sequential processing is adopted. The scene is typically segmented in order to identify the regions of interest (ROI) and further analysis is carried out within the detected ROI. In such a sequential processing scheme the primary cues, colour and shape, are extracted and analysed independently. Usually, colour is utilised for preliminary scene segmentation. It is reasonable as road signs contain distinctively bright colours compared to the background, unless the adverse illumination destroys this valuable information. However, as the colour analysis does not take into account other image features like edges or gradient orientations; in certain situations it may yield suboptimal, inaccurate or even incorrect results. For example, a blue circular road sign will usually stand out from the background blue sky in a feature space spanned by joint colour-gradient features, but may appear completely indistinguishable when the colour information is analysed alone. A real-time processing requirement, which is common in a majority of practical applications, imposes further constraints on the design of the sign detectors. In the absence of computationally efficient multi-feature extraction methods, the sequential scheme may appear a decent solution.

Sequential processing often excuses the failures of inaccurate segmentation because it incrementally reduces the region of attention; each new feature type analyzed refining the estimation obtained through the previous feature type analysis. This principle underlies for example the approach of Ritter [21] who first used neural network for identification of the characteristic colour patches in the scene, but then ran a connectivity analysis to filter out the patches not being part of the traffic signs. In a similar fashion Piccioli et al. [22] reduced the search area based on the a priori known ranges of image coordinates defining the region where the new traffic signs must occur in the scene. To identify the potential sign regions at a finer level, geometrical analysis of the edges extracted in the initially reduced image region was performed. Oftentimes, sequential analysis is done over time for consistency checking of the already detected candidate road signs. For example, in [20] a sign-like shape must appear in the scene for at least several concurrent frames, its radius must not have changed greatly during that time, and it must not move far in the image in order for the sign hypothesis to be accepted.

3. METHODOLOGY

In this Thesis can be divided methodology into three stages: detection, classification and recognition (see Figure 3.1). In the detection stage, color information is exploited to detect regions of interest (ROI) that may correspond to traffic signs. The shape of these regions is tested in the classification stage, allowing rejecting many of the initial candidates and grouping traffic signs into classes. Finally, the pictogram contained on each ROI (if exist) is extracted, analyzed and compared with the pictogram database. The best match between the ROI and database pictogram, if high enough, is considered the sign that is more likely to appear in that ROI. Each recognized sign is part of the output result of the recognition stage.

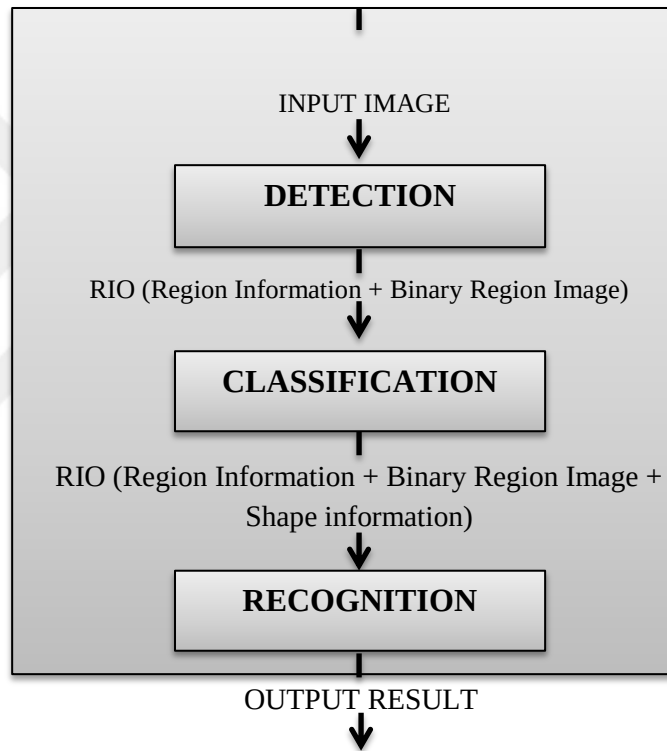


Figure 3.1: Flowchart of proposed system

3.1. Detection

Traffic sign detection, it undertakes a very important role in any application of traffic sign recognition. Actually, if a sign is not perceived as correct, to inform the driver it cannot be classified and recognized. For example, there is a possibility of poor classification and recognition when the sign area cannot be precisely determined.

In the case of traffic signs, the choice of the detection strategy largely depends on how broad class of signs is targeted. When a single type of sign is focused on, it is

relatively easy to develop a fast and robust detector as the target object is uniquely or nearly uniquely defined. On account of this fact, the detector can operate on a dedicated discriminative feature representation that can be derived from real-life sign images or their Highway Code prototypes. The task becomes much more complicated when multiple diverse road signs are to be detected, which is the case in most practical applications. Usually, taking the real-time performance requirement into account, the existing TSR systems of this kind cannot offer a general-purpose detection algorithm capable of handling the entire traffic sign diversity in a uniform way. They must strike a balance between the flexibility of the solution and its computational efficiency. Therefore, they usually specialize in detecting only narrow sign categories that exhibit many common appearance characteristics. For example detection on the European prohibition signs is facilitated by the fact that all of them have a circular shape with red rim, white interior and possibly some black symbols in it.

Unfortunately, the colors present on signs are not exclusively used by them, also appearing on several other objects. They are likely to appear on informative plates (see Figure 3.2(a)), buildings (see Figure 3.2(b)) and also on advertisements (see Figure 3.2(c)), for instance. Even if it is possible to find roads where the previously examples would not appear, traffic lights often appear in the roads, especially in the cities, for traffic regulation. If the red color is lit, it will be obviously detected as a red colored region (see Figure 3.2(d)). Additionally, cars appear on any road, having the most varied body paint colors, including red and blue color and also red colored headlamps (see Figure 3.2(e)). In city roads, people on the sidewalks, may be wearing clothes and/or carrying objects with similar colors to those used on signs (Figure 3.2(f)).

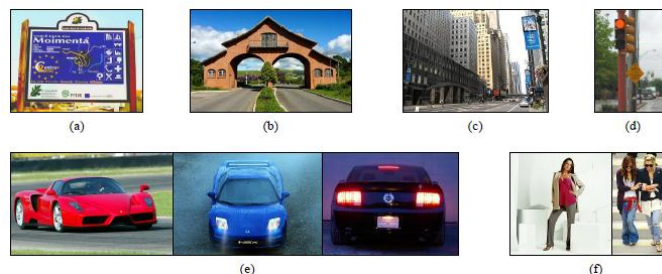


Figure 3.2: Blue and red color used on: (a) – Informative plates; (b) – Buildings; (c) – Advertising; (d) – Traffic lights; (e) – Car body paint and car headlamps; (f) – Clothes.

This means that, although using color is a great advantage to detect the regions where traffic signs appear on images, it will eventually additional detect non-sign regions. To minimize the number of wrongly detected regions, inherent region features (aspect ratio, area, centroid and orientation) are compared with sign features and only regions conforming to these features are considered as ROIs.

For an easy understanding of the procedure, it is shown in Figure 3.3 the four steps used for detection.

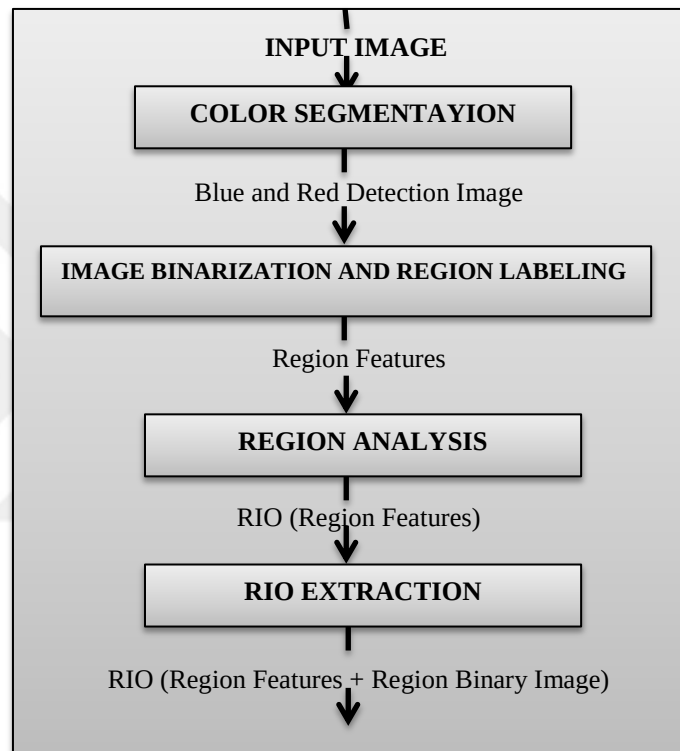


Figure 3.3: Flowchart of Detection

3.1.1. Color Segmentation

The purpose of the color segmentation step is to separate the color of interest from the others present in an image, allowing to locate signs on images by searching for their color. This could be a flawless detection method, knowing that sign colors are standardized for each country, however it is likely to find signs that don't have exactly the same original color and this difference tends to grow for older signs where the color has changed due to environmental conditions like sun exposure.

The HSV color space allows decoupling the color, saturation and intensity information, which, without being flawless, can be very useful to find sign colors at this

stage. Converting the input image to the HSV color space, it is possible to identify the colors by analyzing the hue (H) component. The saturation (S) component is also used, as for very low saturation values the color is no longer reliable. The intensity value (V) does not give any valuable information about the color and it is not used for color detection.

The detection of a color is made using a fuzzy detection of the relevant H and S values. For each pixel, a hue-based detection (hd) of the blue and red colors is done, according to equations (3.1) [29], where hd_{blue} gives a value close to one for blue regions and hd_{red} has a similar behavior for red regions. As H has values in the [0-255] range, values for blue are close to 170, while for red the values of interest are close to 0 or 255.

A saturation detection (sd) value is found by analyzing the S channel (equation 3.2) [29], the values of interest correspond to high color saturation. The curves corresponding to the detection of the hue values corresponding to blue and red, as well as for saturation detection, are illustrated in Figure 3.4.

$$hd_{blue} = e^{\frac{-(H-170)^2}{30}}, hd_{red} = e^{\frac{-H^2}{20^2}} + e^{\frac{-(H-255)^2}{20^2}} \quad (3.1)$$

$$sd = e^{\frac{-(S-255)^2}{115^2}} \quad (3.2)$$

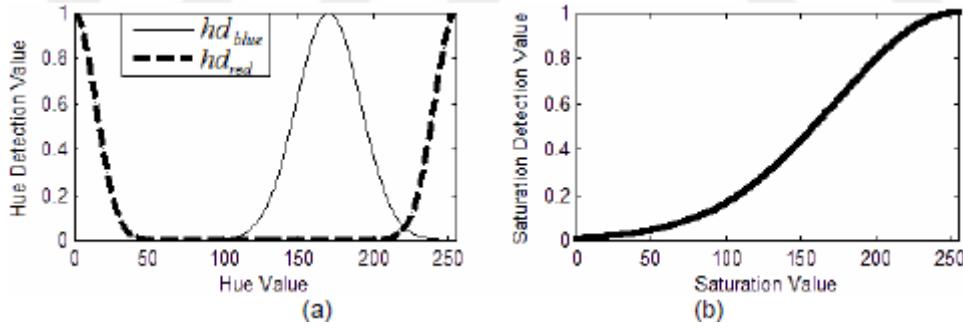


Figure 3.4: (a) Functions for detection of red and blue image areas; (b) Function for detection of color saturated areas

The output of the hd_{blue} or hd_{red} detection functions, with values between 0 and 1, is multiplied with the (sd) output value, yielding an initial detection value (hs) for each pixel. By experimentation, values lower than 0.33 are considered non-sign regions, being discarded by setting their value to 0.

In [29] the values used on equations (3.1) and (3.2) were tuned by experimentation using the input image shown in Figure 3.5, where an example of the color segmentation process is presented. By analyzing the results of the color segmentation procedure it is

possible to see the range of values that are filtered as red and blue sign candidate regions in Figure 3.6 (a) and (b), respectively.

The performance of the proposed color detection strategy is exemplified on a real photography in Figure 3.7. The input image has one big red sign, a medium sized blue sign and two blue signs that appear with a very small dimension, one of them being partially occluded by the red sign post. Applying the color segmentation to the input image, results in the two images hs_{red} and hs_{blue} , containing regions of red and blue color, respectively. By inspection, it is possible to see that the red and blue sign areas are detected successfully, even when the sign appear with very low dimensions.

However, not only the sign areas were detected, but also the red back lights from the car appear as red detected regions. This was expected to happen, since the car lights have red color, similar to the traffic signs. These regions need to be discarded later, to avoid non-sign detections. Another particularity of the color segmentation on real photos is that isolated or low agglomerated pixels are likely to appear with a high color probability value and can be interpreted as noise in the result images.

This issue does not reveal to be problematic, since traffic signs will be regions containing a minimum number of connected pixels, corresponding to the minimum average size of traffic signs that the detection system is expected to detect and all the noisy regions will be ignored.

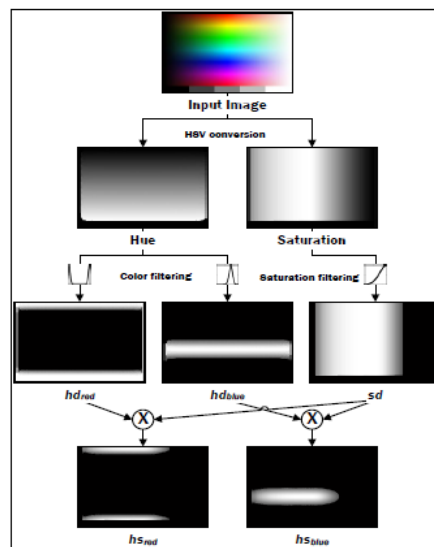


Figure 3.5: Example of the color segmentation process [29]

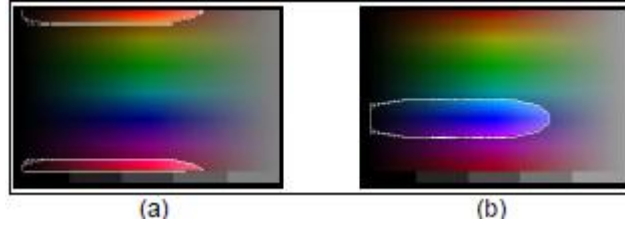


Figure 3.6: Range of colors selected from the input image, after color segmentation, (a) for red color; (b) for blue color [29]

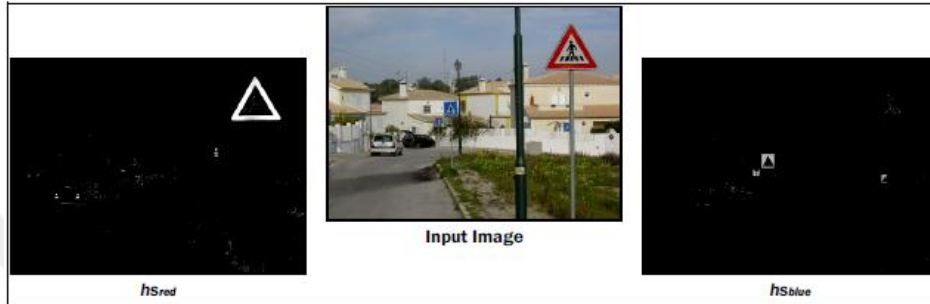


Figure 3.7: Color segmentation for a real photo situation

The color segmentation previously presented works very well for a large variety of images with good color definition. However, the color of signs with high brightness is not always detected and a lot of dark areas were often misclassified as a being of sign color. Having said that and looking at Figure 3.6, probably the color tolerance initially considered was not ideal. The dark color pixels should not be detected, since they don't have enough color information to allow a confident decision about their color and probably increasing the tolerance for handling brighter images would permit detection of the previously missed detections.

Instead of taking advantage of a typical HSV conversion, a new conversion based on the HSV has been adopted, as described in the following. To distinguish the method that will be further presented, from the previous one, they will be referred as method #2 and method #1, respectively. For an input image in RGB format, the initial conversion to HSV is done according to equations (3.3), (3.4) and (3.5) [29], where MAX and MIN are equal to the maximum and minimum of (R, G, B) pixel values, respectively.

$$H = \begin{cases} \text{undefined}, & MAX = MIN \\ 60 \times \frac{G-B}{MAX-MIN}, & (MAX = R) \wedge (G \geq B) \\ 60 \times \frac{G-B}{MAX-MIN} + 360, & (MAX = R) \wedge (G < B) \\ 60 \times \frac{B-R}{MAX-MIN} + 120, & MAX = G \\ 60 \times \frac{R-G}{MAX-MIN} + 240, & MAX = B \end{cases} \quad (3.3)$$

$$S = \begin{cases} 0, & MAX = 0 \\ 1 - \text{otherwise} \end{cases} \quad (3.4)$$

$$V = MAX \quad (3.5)$$

Since for traffic sign detection only the red and blue colors are of interest, method #2 proposes the usage of modified hue-based detection functions for red and blue, according to equations (3.6) and (3.7) [29].

These functions, as before, give the color probability similarity for blue and red color, with values ranging from 0 to 1, where a higher value corresponds to a higher color probability. However, no sd function for analyzing the color saturation is calculated this time, the S channel being directly used instead (equation (3.4)). The option, to avoid pixels where the color is not well defined, was to set to 0 all values where the difference between MAX and MIN is below a threshold value. A value of 0.1, which was found by experimentation, proved to yield better results, being possible to eliminate the areas that presented good saturation values but for which the color was not well defined.

$$hd_{blue} = \begin{cases} 1 - \frac{|R-G|}{MAX-MIN}, & (MAX = B) \wedge (MAX - MIN \geq th) \\ 0, & \text{otherwise} \end{cases} \quad (3.6)$$

$$hd_{red} = \begin{cases} 1 - \frac{|G-B|}{MAX-MIN}, & (MAX = R) \wedge (MAX - MIN \geq th) \\ 0, & \text{otherwise} \end{cases} \quad (3.7)$$

This strategy not only reduces the computation time, but also improves the color detection of the algorithm as shown on Figure 3.8 and 3.9.

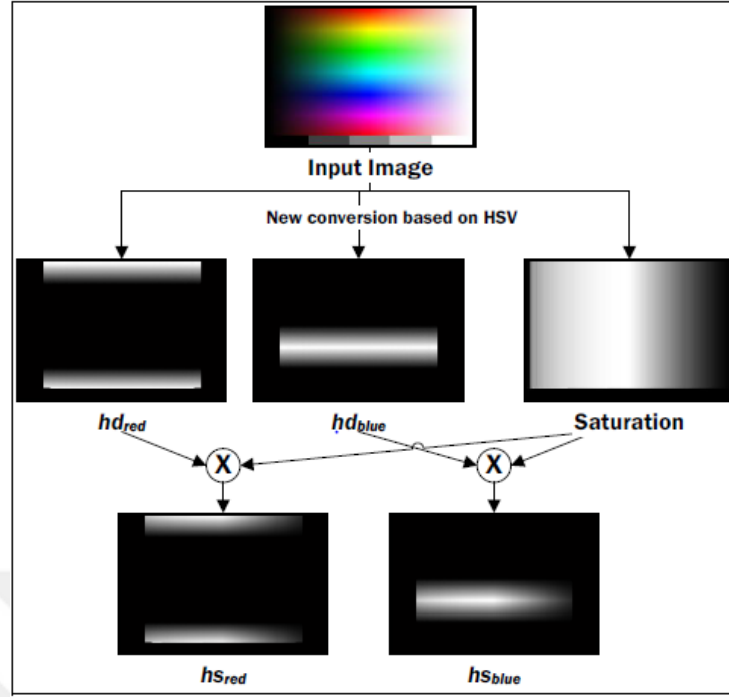


Figure 3.8: Example of the new color segmentation process [29]

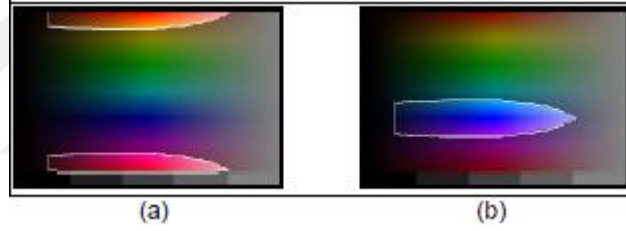


Figure 3.9: Range of colors selected from the input image, after color segmentation, (a) For red color; (b) for blue color [29]

3.1.2. Image Binarization and Region Labeling

Color segmentation detected sign color similarity in images with values ranging from ‘0’ to ‘1’, where a higher value corresponds to a higher probability that the color is contained on traffic sign. However, the main goal of the detection is to find regions where signs are likely to appear. Each color segmentation image (hs_{red} and hs_{blue}) is binaries (i.e., thresholded), so that the resulting ‘1’ valued pixels correspond to sign color and other pixels take value ‘0’. The resulting binary image usually contains more than one detected region, where a region is considered to be any group of 8-connected ‘1’ valued pixels. To easily identify each region a label is attributed to each one, resulting in a labeled image. Finally, a set of important region features is acquired for further processing.

Given a threshold value, all values below that threshold will be set to '0' and those above are set to '1'. The used threshold value is '0.3' and was obtained by experimentation. Figure 3.10 shows an example where the image is binaries according to this threshold, being possible to see that sign regions are well detected.

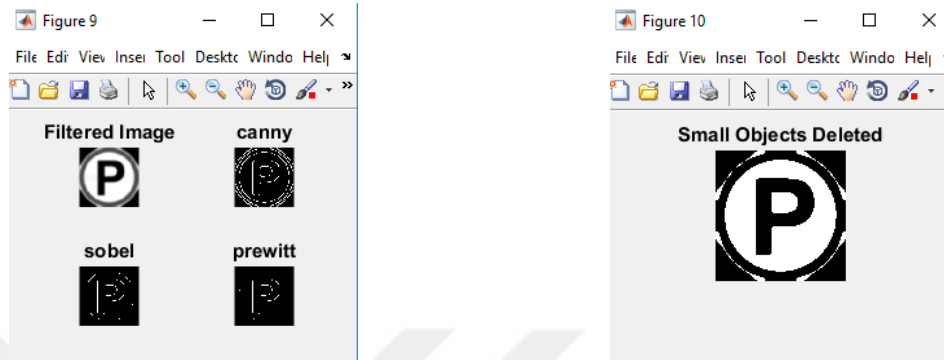


Figure 3.10: Binarization

Again, as the color segmentation, also the binarization detects regions that are not sign related. In Figure 3.11 it is possible to see that the sign region is well binarized, but also many other regions survive the binarization step. This is not problematic, since regions are going to be tested further.

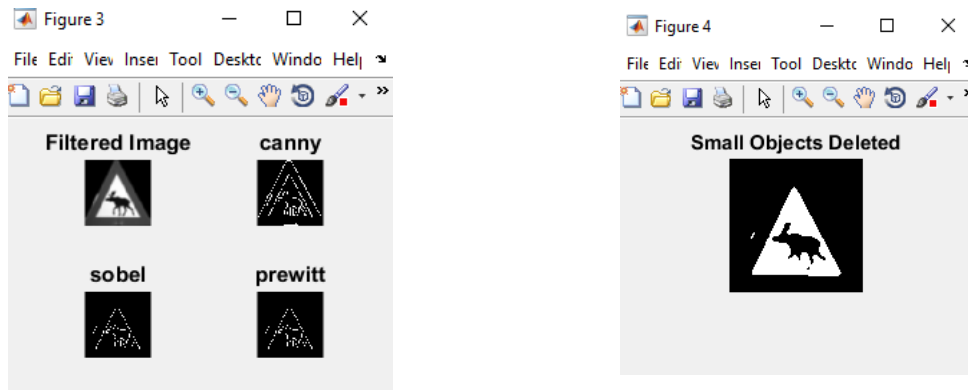


Figure 3.11: Binarization

It is usual that after the color segmentation and binarization, more than one region is found in an image. All the identified regions of the binarized image are labeled, resulting in labeled image (li). Since there are two binarized images, one for red and another for blue color, there are also two labeled images, named li_{red} and li_{blue} , respectively.

The result after the color segmentation and binarization steps are shown at Figure 3.12 (b) corresponding to an input image containing two red signs, see Figure 3.12 (a). The blue color results are not shown they do not contribute with extra information about the labeling process.

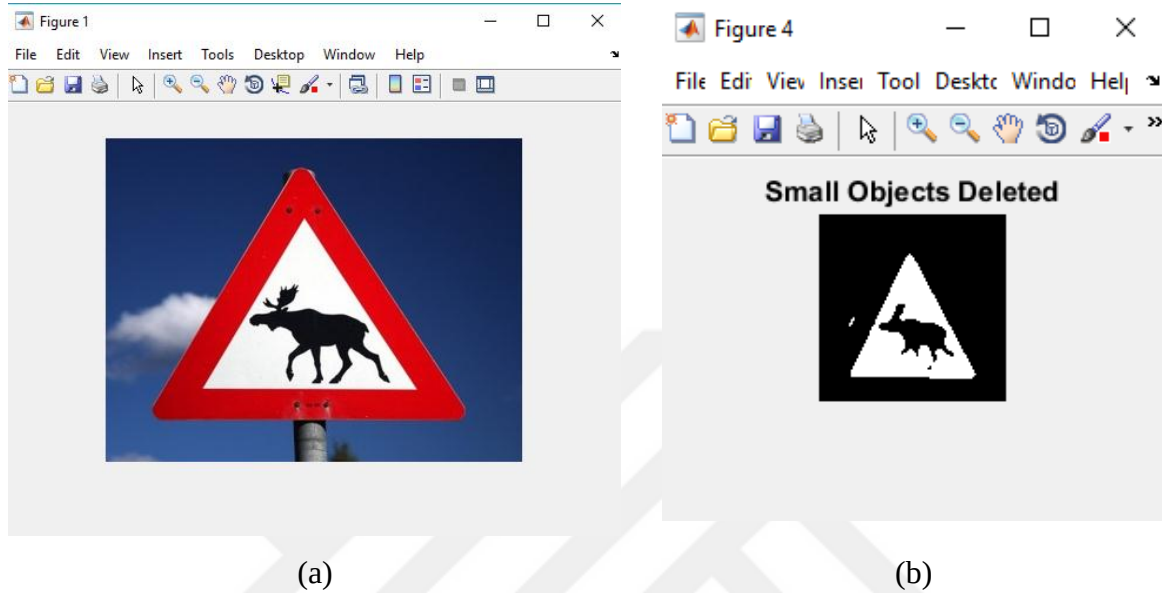


Figure 3.12: (a) Input image containing two red signs; (b) Respective binarized image.

3.1.3. Region Analysis

The regions detected after the color segmentation and binarization exhibit a color very similar to the one expected to be found on traffic signs. However, the sign color may appear on other objects and so, those objects would be detected too.

In this step, the detected regions are tested for sign features including the region area, aspect ratio, respective centroid and orientation. Only regions with features conforming to the ones expected to be found on traffic signs will be considered as ROI.

The region area is useful to discard regions that appear too small, or too big, knowing that there are expected sign sizes. Another factor that can be taken into account is that there is a minimum and maximum area percentage of the region bounding box, which must have the desired color to characterize that region as a sign. If a region is below this minimum or above the maximum, it is rejected as a ROI. The relation between the region area and the bounding box area is named as the region fulfillment (Rf).

Although traffic signs are triangle, square, octagon and circle shaped, their bounding box is approximately squared for red and the majority of blue signs (see Figure 3.13). Concerning blue signs, there are some exceptions, since there are also some signs that are rectangular. Again, the relation between width and height of these signs, referred as the region aspect ratio (Ar), is well known.



Figure 3.13: Sign basic shape illustration

Taking advantage of the previously described sign features, ROIs will be found according to the procedure. Regions not having the expected region area and fulfillment values are immediately discarded. The remaining ones have their aspect ratio computed based on the width and height values of the respective bounding box.

All the regions having aspect ratio values approximately equal to the aspect ratio of a square have a high probability of being a sign. In some cases, also testing the region centroid allows discarding some non-sign regions. Regions that conform to the expected sign features are added to the ROI vector.

3.1.4. RIO Extraction

Even though some sign features were already taken into account, allowing finding regions with high probability to contain a sign, there is still no indication as to the class to which the sign belongs, or a guarantee that it really is a traffic sign.

Here, sign classification is done by combining shape and color information, where the color is already known to be blue or red. The next step is check if the ROI shape is a triangle, square, octagon or a circle. Instead of doing shape classification by analysis of the

full image and since ROIs were already identified, each one can be classified independently, providing more robustness and improving the classification performance.

Each ROI contains information about the region coordinates in the labeled image (li). The region image can thus be cropped from the li and binarized, where pixels with the desired region label are set to '1' and the remaining to '0'. To improve the future shape detection, all the areas with '0' valued pixels, that have pixels labeled '1' as 8-neighbors, are also set to '1', being these areas are denominated 'holes'.

An example of ROI extraction is presented in Figure 3.14. The input image contains only the blue sign at the right side of the image. Due to that, only the li for blue color is shown, where the sign region is represented with a bright red color in the labeled image (li_{blue}). The cropped image from li_{blue} , that contains the sign region, may still include regions that do not belong to the detected sign, corresponding to noise, as in this example. In this case, those regions are not problematic, but there might be cases where those additional regions, such as in Figure 3.15, where the sign shape could be misunderstood. To prevent unwanted changes to the region shape, only the labeled pixels corresponding to the tested region are set to '1', the remaining ones being set to '0'.

The next operation is to fill all the 'holes' of the cropped image with '1' valued pixels. This ensures that only the sign shape will be further tested, and not the pictogram shape inside the sign. The squared sign of Figure 3.14 might be mistaken with a triangular sign, if it's 'holes' weren't previously filled. Removing the pictogram triangle from the cropped image clearly ensures that only one shape can be found, which the real sign shape is.

However, signs that appears with low resolution or vandalized like the one shown in Figure 3.16, may not contain 'holes', meaning that the filling operation does not modify the labeled image. Nevertheless, even when part of the shape information is missing and the filling operation does not produce the desired results, if enough sign information is detected, then the sign shape can still be correctly identified. Looking at the sign of Figure 3.16, it is possible to see that the lower left side of the sign was not detected, and thus no 'holes' were found. This could appear problematic, but most of the shape was detected and it is not likely to be mistaken with any other shape.

The classification stage will need to find shapes like triangles, squares, octagons and circles. These are basic shapes and it is not required to describe these shapes with a large amount of resolution. Instead, a maximum resolution is defined, which not only allows reducing the memory used but also the computational cost.

Finally, the canvas size of the image is adjusted, increasing 1 pixel for each side. Like said before, this is required because the shape classification requires contrast information between pixels. If the region pixels are included on the border of the image, no contrast will be found at those pixels, degrading the shape information.

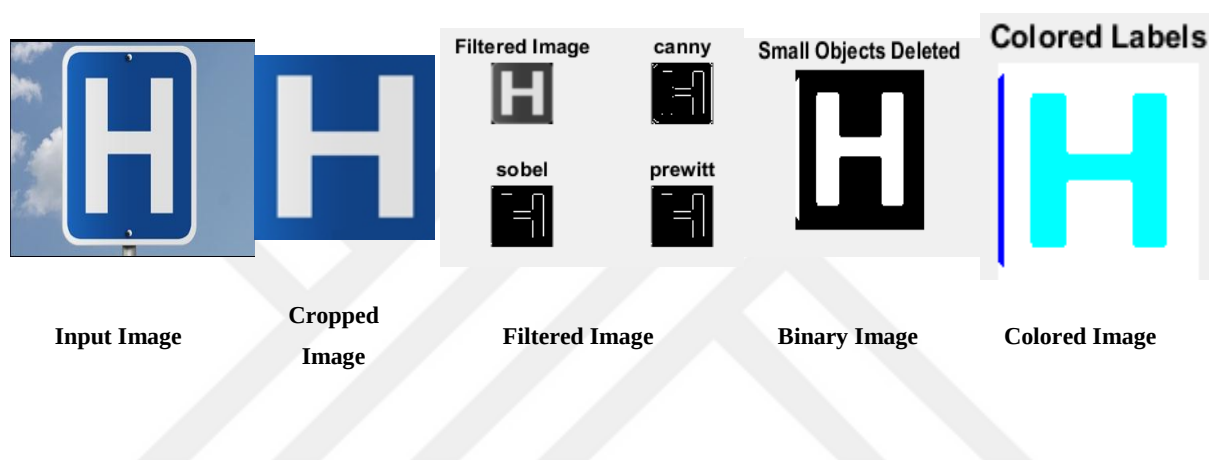


Figure 3.14: ROI extraction example

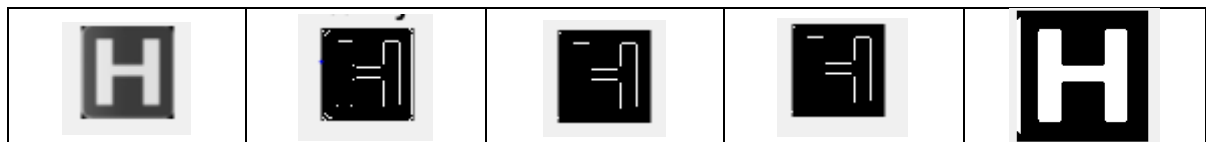


Figure 3.15: Binarization Example



Figure 3.16: Problems on filling 'holes'

3.2. Classification

The classification module takes the detected ROIs and classifies them into one of the considered classes: information, danger, prohibition or obligation, or as a non-sign. In addition, Yield, Wrong Way and STOP signs are recognized as special cases.

The binary maps of each of the ROIs are evaluated separately according to the shape at this stage and a probability value is assigned which includes triangular, square or circular shapes. If there is at least one high probability (above 75%), the most valuable shape is assumed for this sign. Otherwise, if the region has at least 50% of red color, there are two possible signs to be tested: Wrong Way and STOP signs.

Differences of these signs are detected in the center region, where the Wrong Way presents a white band and the STOP sign also contain parts of red color in the same center region location. These differences can and are used to distinguish between a Wrong Way and STOP sign. If it is not considered as belonging to any of the tested classes, that ROI is classified as a non-sign region and is discarded.

Classification from one of the noted classes is done taking into account both information of shape and color. In the next sub-sections are discussed the methods for shape classification, where methods for identifying circles as well as triangles and squares are presented.

Traffic signs have two dimensional outside geometric shapes and, like any shape, are formed by a closed line. This line can be a smooth line without any peaks in its extension, resulting on circular or elliptical shapes (Figure 3.17), or may contain points with abrupt line direction changes, i.e., vertices or corners, resulting on the most varied shapes (Figure 3.18).

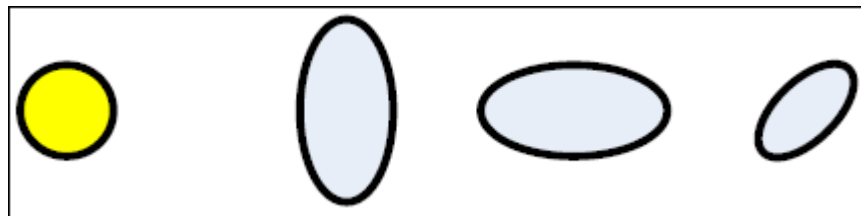


Figure 3.17: Circular and elliptical shapes

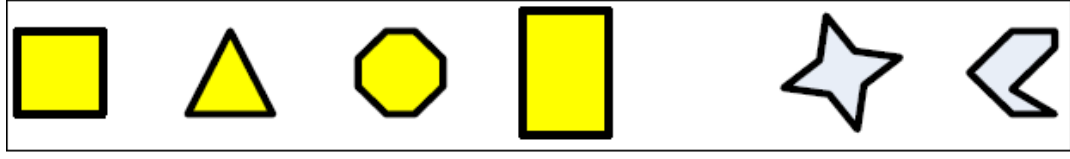


Figure 3.18: Shapes with vertices

Despite the infinite number of possible shapes, traffic signs have regular shapes, with symmetry along the vertical axis, such as the ones represented with yellow color in the previous figures.

3.2.1. Shape Identification for Square and Triangle

Shapes of squared and triangular are identified by finding the corners of each ROI, using the Harris corner detection algorithm [30]. The existence of corners is then tested in six different control areas of the ROI, as illustrated in Figure 3.19. Each control area value (tl, tc, tr, bl, bc, br) is initialized to zero. When a corner is found inside a control area, the respective value (0.25 for vertices and 0.34 for central control areas) is assigned to that control area value.

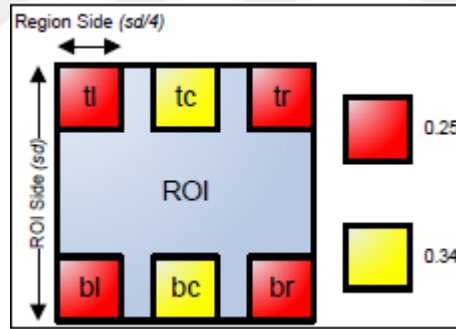


Figure 3.19: For corner occurrence regions tested

The probabilities that a given ROI contains a triangle pointing up (tup), a triangle pointing down (tdp), and square (sqp) are computed according to equations (3.8, 3.9, and 3.10) [29].

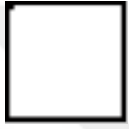
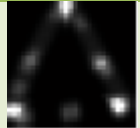
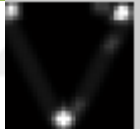
$$sqp = tl + tr + bl + br \quad (3.8)$$

$$tup = 1.32 \times (bl + br) + tc - 1.1 \times (tl + tr) \quad (3.9)$$

$$tdp = 1.32 \times (tl + tr) + bc - 1.1 \times (bl + br) \quad (3.10)$$

In figure 3.21 for the example, sqp scored 0% and both tdp and tup had a value of 0.34%. In this case, only the circle probability scored at least 75%, and the sign was correctly identified as a circle, more results are shown in Table 3.1 with the respective values for (sqp), (tup) and (tdp).

Table 3.1: Square and triangle classification results

Example Number	Input ROI	Corner detector result	Sqp	tup	Tdp
1			0%	0%	0%
2			100%	11%	11%
3			50%	100%	0%
4			50%	0%	100%
5			0%	0%	0%

3.2.2. Circle Shape Identification

The circle is probably the simplest shape available. In contrast to the triangle and square shapes discussed previously, where a rotation modifies the perception of the shape, a circle can be rotated maintaining the appearance whatever the rotation angle.

This is shown in Figure 3.20, where a circle and square are rotated 45 degrees, resulting on the red shapes. Superimposing the two different colored shapes it is possible to see that the circles maintain its shape properties, while the squares apparently results on a new shape with eight vertices, instead of four.

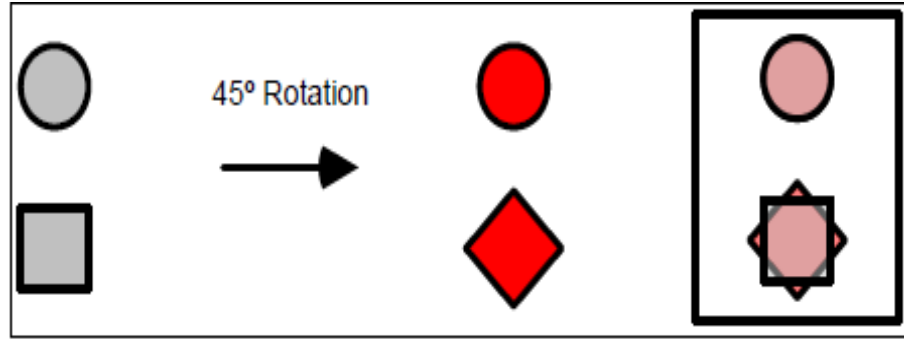


Figure 3.20: Circle rotation invariance

This may not seem a big advantage, since signs are supposed to be always vertically aligned. However, this property can be exploited for circle identification using the detection method Fast Radial Symmetry (FRS) [31].

In the present case, the shape is always contained in a binary image with known dimensions. This means that if a circle shape is present on the binary image; its radius will be approximately half the width of the binary image. Each ROI is tested for the probability of containing a circle using the FRS method.

If a shape of circular is present, the Fast Radial Symmetry detection (FRS) output will contain high values on the circle's central area. In ideal conditions, only the center pixel of the output image would need to be tested, but for real images all pixel values inside a square region, around the output center, are analyzed. The size (sz) of the squared region used is 20% of the largest dimension (width, ow , or height, oh) of the output image, as shown in Figure 3.21.

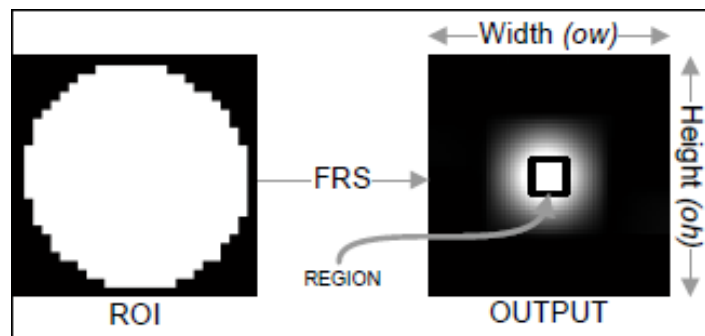


Figure 3.21: Circle classification process

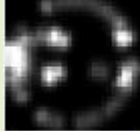
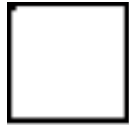
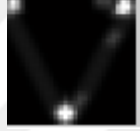
Within this squared central region, all pixel values are averaged (*avg*). The average value and the maximum (*max*) output value are used to find the resulting circle probability (*cp*) according to equation (3.11).

$$cp = \begin{cases} 0 & max < \\ avg & 0 \leq max \leq 1 \\ \frac{avg}{max} & max > 1 \end{cases} \quad (3.11)$$

On Figure 3.21's a *cp* value example of 88.5% was obtained, the next table shows some circle classification results for some detected ROIs.



Table 3.2: Circle classification results

Example number	Input ROI	Corner detector result	Sqp
1			94.2%
2			25.8%
3			4.9%
4			7.2%
5			94.1%

3.2.3. Classification of Traffic Sign

Signs can be classified into the considered classes, after color and shape information is known, as shown in Figure 3.22. Like said before, the Yield, Wrong Way and STOP signs are recognized at this stage.

The sign of Yield is recognized as the only red colored sign with triangular pointing down shape. As for the Wrong Way and the STOP sign, it has been said that the red color contained on the center region is used to distinguish them. The center region is defined by two parameters: the width (valued has 60% of the ROI) and height (valued has 10% of the ROI) of the region. Both values are centered on the ROI, and all parameters were taken by experimentation, to fit correctly the white band of Wrong Way signs. Finally, if more than 25% of pixels are red, the sign is considered as a STOP sign. Otherwise, the sign is considered to be a Wrong Way sign.

While the Yield and STOP signs appear on Table 3.3, the Wrong Way sign is grouped in the prohibition sign, since there are more signs presenting a circular shape and a red color.

Table 3.3: Traffic sign classification into the considered classes

SHAPE COLOR					
BLUE			INFORMATION	OBLIGATION	
RED	DANGER	YEILD SIGN		PROHIBITION	STOP SIGN

Almost all the classified signs (see example in Figure 3.22), had their shape correctly classified. The only exception is usually the STOP sign, which is often mistaken for a circle shaped sign (Figure 3.22(b)). However, with the previously described is possible to detect that it is a STOP sign, instead of a Wrong Way sign, ensuring that even with a bad shape classification, the sign is correctly classified.



Figure 3.22: Correctly classified signs except *STOP* sign

3.3. Recognition

Traffic signs recognition is a hard multi-class problem. In practice, handling the entire gamut of pictograms is never considered in TSR. This would be impractical as the total number of signs is huge, they differ from country to country, and some of them are extremely rare. Therefore, the common approach adopted is to focus on a relatively narrow category of the most relevant signs within one country. This reduces the complexity of the classification task and is hence more suitable for in-vehicle application. Besides, many traffic signs are not standardized in terms of colour, shape and the symbols contained. A

good example is the signs giving direction, which may vary in terms of size, shape, and the background colour of the plate. Such signs also contain various symbols, like arrows, as well as variable-font characters. With such great appearance variability, only very coarse class definitions make sense, but this kind of categorization is of little practical use. Alternatively, the contained textual information, if present, may be focused on. However, fast and reliable recognition of these kinds of patterns is not possible in low-resolution and noisy imagery captured with a wide-angle camera and therefore requires more advanced hardware, e.g. an additional telephoto lens.

Road sign recognition involves two critical tasks, each requiring special attention: feature extraction and classification. In particular, the sign interpreter will certainly be allowed to trigger certain mechanical actions to increase the driver's safety or to prevent a vehicle from breaking the traffic regulations.

Problems could arise for different signs that contain the same pictographic information – as the ones shown in Figure 3.23. However, this similarity only occurs for signs of different classes. And, since ROIs were previously classified into classes, the problem is easily bypassed.



Figure 3.23: Signs with similar content

Regarding each class, although it is possible to see signs having similar pictograms, the outer contours of each pictogram are unique. The example of Figure 3.24 shows three different signs with similar pictograms, whose outer contours are enough to make them distinguishable.

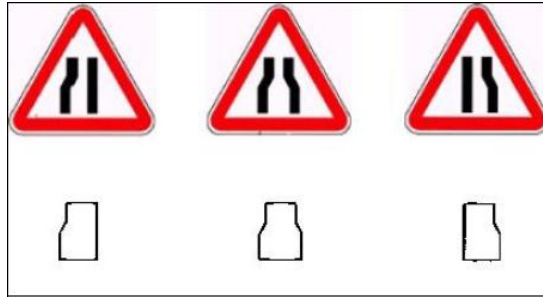


Figure 3.24: Similar pictograms with distinguishable contours [29]

A block diagram illustrating the procedure used to recognize a sign is presented in Figure 3.24. It starts by extracting the pictogram information of each ROI. If the resulting pictograms have two or more disconnected regions they are connected together, to obtain a representation consisting of a single (and unique) contour for each sign. Then, the contour information is transformed into contour peak based information, using the curvature scale space (CSS) representation, being matched with the database to find the best candidate.

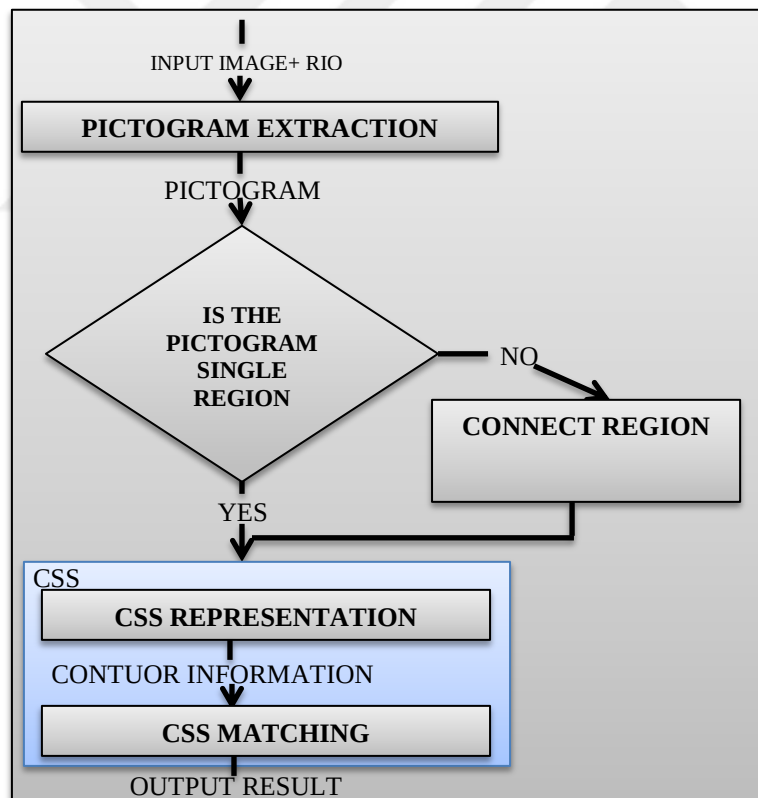


Figure 3.25: Flowchart of recognition

3.3.1. Pictogram Extraction

Pictograms contained on signs are usually displayed using a black color over a white background area (see example in Figure 3.26 (d)). There are however many exceptions to this general case. In the case of red signs there are exceptions where part of the pictograms is displayed with a red color (see example in Figure 3.26 (a)). Also for blue signs, pictograms can contain parts in red color (see example in Figure 3.26 (b)), but some blue signs also contain pictograms using the white color (see example in Figure 3.26 (b and c)).

There is also an exceptional case, where a blue sign contains a blue pictogram over a white background area, whose pictogram is not correctly extracted by the method proposed here, since the majority of blue signs contain blue areas inside white pictograms (see example in Figure 3.26 (c)).

Signs shown on Figure 3.26 were chosen to display the most common pictogram types contained on the global signs. Also, these examples correspond to ideal cases, since these are database/template signs, without color or shape distortion. The letter indexes used on Figure 3.26 will be used throughout this section, to represent each sign.



Figure 3.26: Pictogram color. (a) – Red signs: pictogram contains black and red colors; (b, c and d) – Blue signs: pictograms contain black, red and white colors.

Due to the above mentioned characteristics of some signs, and despite the similarity of the overall pictogram extraction method, slight differences in extraction for each sign color (red or blue) had to be implemented.

Taking advantage of the previously collected information, including the red and blue segmented regions (Figure 3.27 and Figure 3.28 refer to the following examples) and using the red component of the RGB color space (see example in Figure 3.29), Pictograms It is also possible to identify signs for white areas in the parts (red or blue) that share the sign color as well as these white areas.



Figure 3.27: Red detected regions



Figure 3.28: Blue detected regions

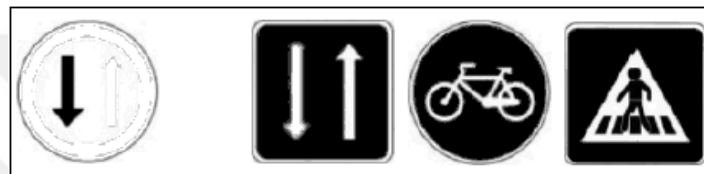


Figure 3.29: Red component of RGB color space

- **Pictogram Extraction Examples**

The figures of the previous example contained signs extracted from the Global sign database, with near optimum color values. An example for real photo signs is now presented, showing signs where the pictogram is correctly extracted (see Figure 3.30 (a), (b) and (c)), a problematic case where the sign appears partially occluded (see Figure 3.30 (d)) and an exceptional case where the pictogram of a blue sign is represented also in blue, consisting in the single exception in the complete Global sign database. This case is not correctly handled by the previously explained algorithm (see Figure 3.30 (e)). The red and blue detected regions are shown in Figure 3.31, Figure 3.32.

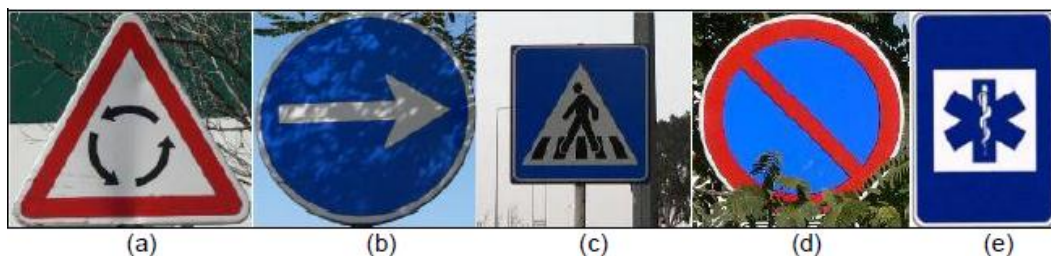


Figure 3.30: Five real photo signs



Figure 3.31: Red detected regions



Figure 3.32: Blue detected regions

3.3.2. Connect Regions

After a successful pictogram extraction, it is possible to test if it has a single region, whose outer contour will be used for recognition purposes. Pictograms represented by a single region, can have their outer contour described using a CSS representation that will be used for the recognition task.

If the sign's pictogram is represented by two or more regions, do not convert outer contours directly into a CSS representative. In this case it is then necessary to find a single contour enclosing the pictogram, connecting all the independent regions into a single one. This is the purpose of the second module represented in Figure 3.25. "Connect Regions".

Taking as example a roundabout sign, the resulting pictogram has three disconnected regions that need to be represented by a single contour (see Figure 3.33 (a) and (b)). A way to do this is by using an active contour that starting from the bounding box of the pictogram (see Figure 3.33 (c)) is deformed until it conforms to the outer boundary of the various regions composing the pictogram (see Figure 3.33 d).

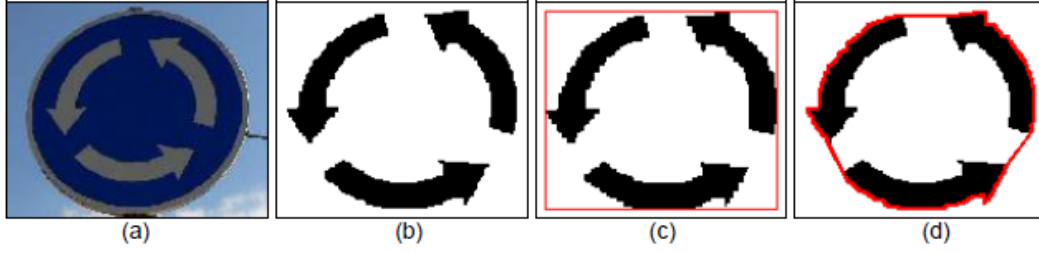


Figure 3.33: Region connection idea (a) – A roundabout sign (b) – Roundabout extracted pictogram (c) – Curve encasing the pictogram (d) – Curve deformed according to pictogram shape

3.3.3. Curvature Scale Space (CSS)

Is one of the central problems in computer vision that object recognition and representation. The Curvature Scale Space (CSS) [32] transforms an object contour shape into a very compact representation that is robust in respect to noise, orientation and scale. It is a powerful descriptor for shape matching and it is used here for pictogram recognition.

A brief description of CSS is presented in the following two subsections. The first one describes the principles behind the CSS representation. The second explains how CSS information can be used for matching purposes.

This is the purpose of the third module represented in Figure 3.25. “CSS”, which is divided in the two subsections previously referred.

- **CSS Representation**

Is a multi-scale representation that describes object shapes by analyzing a closed contour’s shape inflection points this is called Curvature Scale Space. Inflection points are the points where the second derivative changes sign, i.e., the zero-crossing points of the second derivative, or in other words, the points where the contour changes from being concave upwards (positive curvature) to concave downwards (negative curvature), or vice versa.

According to the definition, the curvature at a given point is measured by the derivative of the tangent angle to the curve.

Considering a closed planar curve Γ (i.e. a non-self-intersecting contour) defined as:

$$\Gamma(u) = (x(u), y(u)) \quad (3.12)$$

With $x(u)$ and $y(u)$ representing the parametric coordinates of the curve and u as arc length parameter, the curve curvature can be expressed as:

$$k(u) = \frac{x_1(u)y_2(u) - x_2(u)y_1(u)}{(x_1^2(u) + y_1^2(u))^{3/2}} \quad (3.13)$$

Where $x_1(u)$, $y_1(u)$ and $x_2(u)$, $y_2(u)$ are the components of the first and second derivatives to the curve, respectively [33].

Gradually smoothing the curve allows computing the curvature at various levels of detail, also known as an evolved curve. For the evolved curve defined by:

$$\Gamma_\sigma(u) = (X(u, \sigma), Y(u, \sigma)) \quad (3.14)$$

Smoothing can be achieved with a Gaussian filter with a progressively larger width σ . Considering the Gaussian filter $g(u, \sigma)$, the respective components of the curve can be computed as follows:

$$X(u, \sigma) = x(u) * g(u, \sigma) \quad (3.15)$$

$$Y(u, \sigma) = y(u) * g(u, \sigma) \quad (3.16)$$

And, according to the properties of convolution, the first and second derivatives of each component are:

$$X_u(u, \sigma) = x(u) * g_u(u, \sigma) \quad (3.17)$$

$$Y_u(u, \sigma) = y(u) * g_u(u, \sigma) \quad (3.18)$$

$$X_{uu}(u, \sigma) = x(u) * g_{uu}(u, \sigma) \quad (3.19)$$

$$Y_{uu}(u, \sigma) = y(u) * g_{uu}(u, \sigma) \quad (3.20)$$

And the curvature of each level of detail can be computed by:

$$k(u, \sigma) = \frac{X_u(u, \sigma)Y_{uu}(u, \sigma) - X_{uu}(u, \sigma)Y_u(u, \sigma)}{(X_u(u, \sigma)^2 + Y_u(u, \sigma)^2)^{3/2}} \quad (3.21)$$

The curvature inflection points, from now on named as zero-crossing points, are defined as the solution to $k(u, \sigma) = 0$, being the arc position of a Gaussian smoothed curve of width σ . As the result of the curve smoothing, the number of zero-crossings decreases,

reaching zero when the curve becomes convex. Determining the locations of the curvature zero crossings for each σ , it is possible to create a binary image called CSS image of the curve, having characteristic peaks (CSS peaks). These peaks are described in terms of u and σ , the highest ones composing the CSS representation.

The CSS peaks of the contour along with circularity, eccentricity and aspect ratio of the pictogram, compose the CSS information used in this Thesis. Also, the used method is conveniently adopted to fit computational data, since any shape needs to be discretized in a number of points.

- **CSS Matching**

CSS matching takes a central role for traffic sign pictogram recognition, since it evaluates the similarity between a specific sign and the database signs. CSS matching is done on a database that contains all the sign pictogram information converted to CSS information.

The matching of two CSS images consists of finding the optimal horizontal shift of the maxima in one of the CSS images that would yield the best possible overlap with the maxima of the other CSS image [34]. The sum of pairwise distances between corresponding pairs of maxima is the resulting matching cost.

The implemented algorithm starts by comparing the circularity, eccentricity and aspect ratio parameters, greatly reducing the number of database sign pictograms that need to be further tested, as well as reducing the computational cost of the matching algorithm. The matching cost will be computed for the database pictograms that pass this first test.

In Figure 3.34 it is possible to see why an optimal horizontal shift needs to be found for CSS matching. Using the same contour with a different starting point (pictogram rotation results on different starting points) results on shifted a CSS image and consequently on shifted CSS peaks. A direct comparison between the peaks would give an unwanted matching result. Shifting the peaks along the x-axis makes the method invariant to rotations.

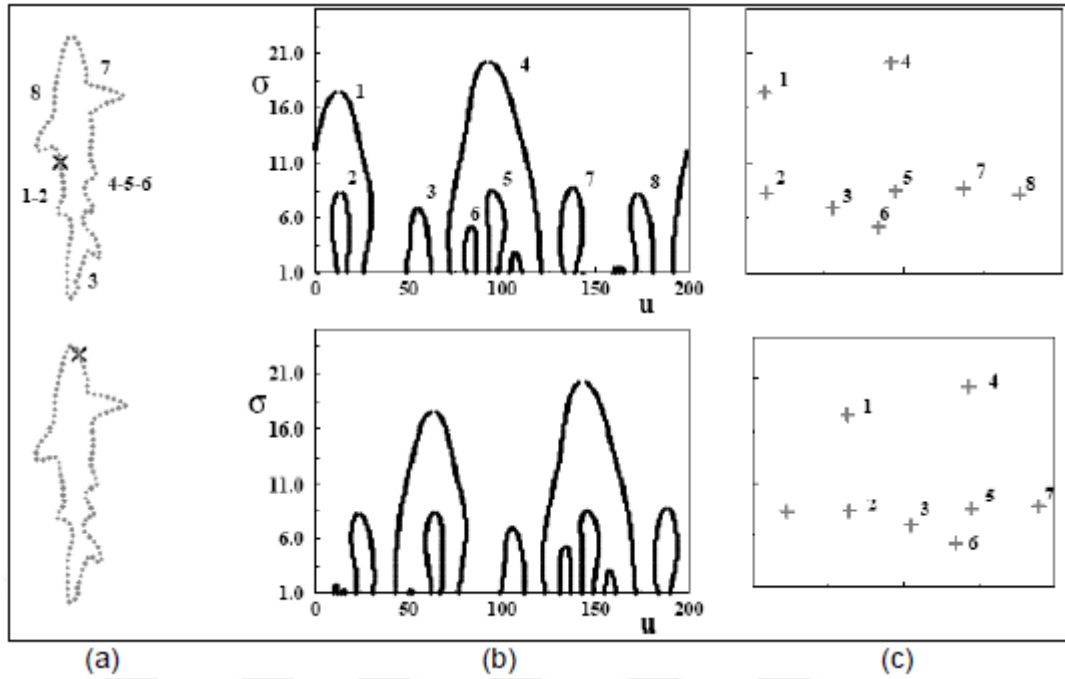


Figure 3.34: The influence of the starting point on the CSS image [35]. (a) Re-sampled outer contour with 'x' marking the starting point, (b) CSS image, (c) CSS peaks

Comparing peaks for each shifting possibility available would require a large computation effort. Instead, to reduce the peak comparison effort, only the two highest peaks are compared. Actually, shapes with different values for the highest peaks, have a very high probability of not being similar.

The two highest peaks of each input and database pictogram pair are combined in the four possible ways, and the distance between each corresponding peak pair is summed, resulting on a matching cost. Highest similarity values are obtained for the lowest matching costs, where a zero matching cost only occurs for identical contours.

4. EXPERIMENTAL RESULTS

This chapter discusses the implementation results; First, the sign database is tested, where good results are expected, since the method was based on these signs. Afterwards, photos containing real signs are tested to see if the overall results are maintained.

In this thesis a total of 50 real traffic signs that has been selected from a set of test pictures and considered for detection, of which 20 good, and 8 excessive luminosity, and 22 presented low luminosity. Since the recognition testing was not done at the same time of detection and classification and the recognition has proved to be the most time consuming routine, a smaller set of images was tested. Also, the recognition results are based only on successfully detected and classified signs.

This thesis presented to traffic sign detection and recognition we used blob analysis method. Road traffic signs usually classified depending on two main basics, shape and color. They are positioned and designed in such a way to be noticed easily while driving. Firstly the input video should be converting to frames to be processed. Then in the first step of blob detection which called blob extraction, the original image is converted to grayscale image then to binary image using quantization process.

Quantization of color images is a process that used to reduce the number of distinct colors used in an image, but it should be considered that the new processed image visually should be similar as it could be to the original one. The original images with its corresponding gray scaled and binary image are shown in figure 4.1.



Figure 4.1: Original image with its corresponding grayscale and binary image

Then then the binary image is processed, first it will be inverted then all connected components (objects) that have fewer than specific number of pixels will be removed. Then the binary image is scanned, the scan process start from upper-left and continued to the right and from top to bottom until the entire image is scanned. This process help tracing region boundaries in binary image, each shape is saved in a unique RGB colored image called blob. Figure 4.2 shows the detected sign in both colored and processed binary image with extracted blob of a traffic sign.

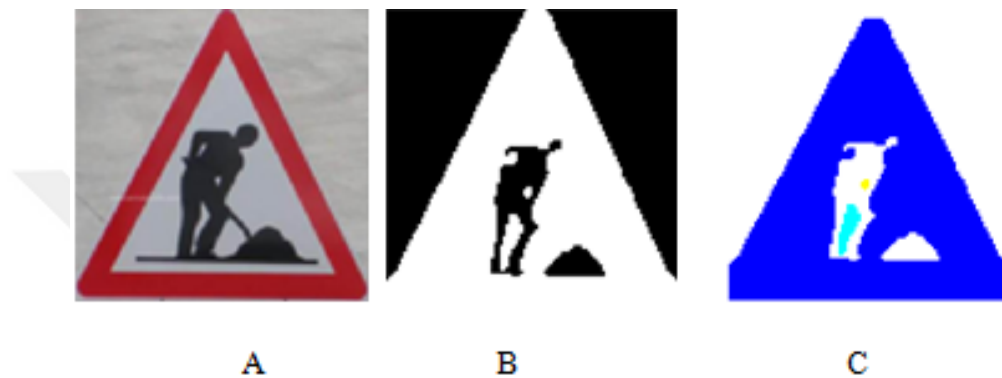


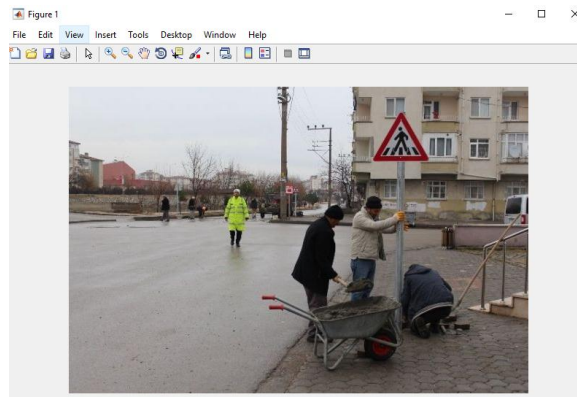
Figure 4.2: detected sign using blob detection method

Finally, after the presentation of the detection, classification and recognition results, another set of randomly selected photos are tested for the overall results.

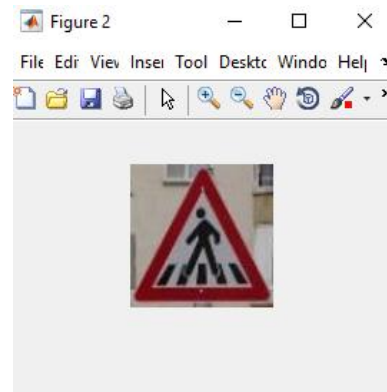
4.1. Detection Results

In chapter three detailed the sign detection method is Applied, on the database signs has proved that the detection works well for correctly colored signs, since all were detected resulting in a 100% detection rate.

Correct detection rates for the real signs of 90.0%, 100.0%, and 50.0% were achieved for each lighting condition (low, normal and excessive luminosity respectively), correspondent to an overall of 88.0% detected signs correctly (see Table 4.1). The lost detections occur for partially of signs close or presenting unforeseen color characteristics due to conditions of environmental. As showing in figure 4.3.



Original Image



Detection Image

Figure 4.3: correctly detected example

Table 4.1: Detection sign results

Luminosity	Present Signs	Detected Signs	Detection Rate	Overall Detection
Low	22	20	90.0%	88.0%
Normal	20	20	100.0%	
Excessive	8	4	50.0%	

The detection is one of the most important phases of any automatic image recognition purpose, since it allows focusing the processing only on the important image areas.

4.2. Recognition Results

Finally, the recognition of traffics signs is tested. Again, when using the recognition method for database photos a good result is expected, since the detection and classification proved to be highly effective, and all the work done had these database photos as a model.

Like said before, all the database signs are fully detected and well classified into the respective classes. Even if the 100% detection result is achieved for database signs, it is already an excellent result. Any detection mistake for database signs would mean that real signs would be also mistaken.

However, even using database signs, the used recognition method presented some incorrect results, mistaking some of the excepted signs for other ones. These results are displayed in the following figure 4.4.



Figure 4.4: Signs database

Table 4.2: Recognition results for database signs

	Danger signs	Prohibition signs	Obligation signs	Information signs	Non-classified signs	All Signs
Presented Signs	17	11	13	6	3	50
Recognized Signs	15	10	10	4	3	42
Recognition Rate	88.5%	90.0%	77.0%	82.5%	100%	84.0%

The recognition result is not fantastic, since almost 16% of signs were incorrectly or not recognized. It is expected to have higher error values for real sign pictures, for the most varied reasons including: noisy pictures, bad weather conditions, rotated signs and more.

A set of real pictures is tested for sign recognition purposes. This will be done in a small set of pictures, but capturing varied sign types. However, bad weather condition and night pictures are not part of the set.

After collecting some detection and recognition output results it was possible to see that, although the detection and classification performs well in the most of the situations, the recognition tends to give incorrect values for low definition signs (see Figure 4.5).

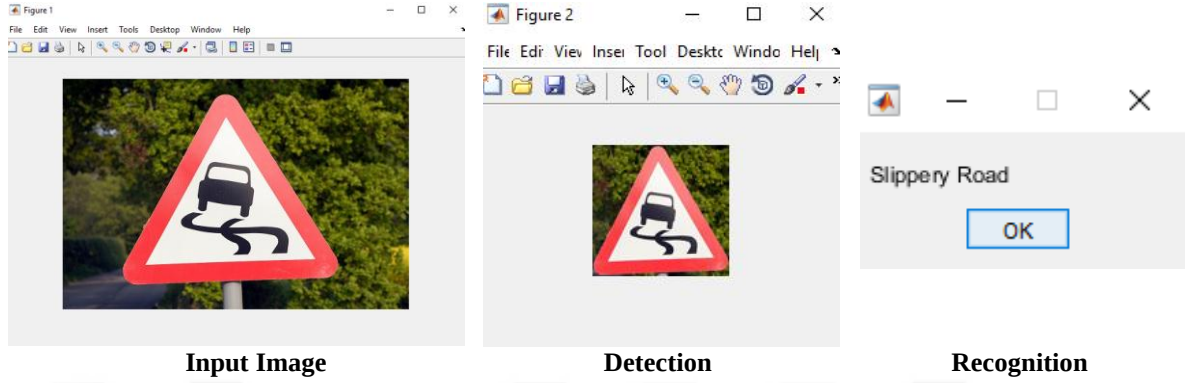


Figure 4.5: Influence of sign definition in the recognition result

Recognition results for real signs are presented in Table 4.3. For a comparison purpose, since the entire model signs were successfully detected, these results were also based on successfully detected signs.

Table 4.3: Recognition results for real signs using successfully detected and recognized signs

	Danger Signs	Prohibition Signs	Obligation Signs	Information Signs	Non-Classified Signs	All Signs
Presented Signs	8	6	5	4	22	45
Recognized Signs	6	4	3	2	22	37
Recognition Rate	75.0%	66.6%	60.0%	50.0%	100%	81.4%

4.3. Overall Results

For the overall results, a randomly set of pictures is chosen and tested for detection and recognition purposes altogether. Detailed information about this test is presented on Table 4.4.

As expected, the overall recognition rate is below 80%, since this was the value obtained considering that the signs were successfully detected. Taking the detection and classification errors into account, the overall recognition rate is of 67.5%, which is far

from the desired values. Nevertheless, the set of non-classified signs are almost always correctly recognized and a decent rate of obligation signs is well recognized. The danger and prohibition signs appear as the most problematic ones to be recognized, having recognition rates below 60%.

Table 4.4: Recognition results for real signs

	Danger Signs	Prohibition signs	Obligation signs	Information signs	Non-classified signs	All Signs
Presented Signs	17	11	13	6	3	50
Detected Signs	15	10	11	5	3	44
Recognized Signs	15	10	10	4	3	42
Overall Recognition Rate	53.9%	55.5%	82.5%	60.2%	100.0%	67.5%

5. CONCLUSIONS

This thesis presented to traffic sign detection and recognition we used blob analysis method and based on the (MATLAB) program platform. Blob analysis algorithm depended on the color and shape for detection. Then in the first step of blob detection which called blob extraction, the original image is converted to grayscale image then to binary image using quantization process. This process help tracing region boundaries in binary image, each shape is saved in a unique RGB colored image called blob.

After many trying, we have run our program many times, have tried with different inputs, we are working on a complete quail program. Depending on simulation result this method can successfully detect and recognize traffic sings accurately recognize traffic signs with fast response and it can be used for real time detection.

Detection of Correct traffic sign is essential for classification accurate, as expected. Classifying each sign into a class allows not only to reduce the error probability, but also reduces the overall method computation time, since the most time consuming routines are done into a smaller set of signs.

Recognition of traffic sign is one of the critical tasks for the intelligent vehicles which are they a technology of the future. Solving the TSR problem will have deep implications on our safety while driving and should therefore reduce the number of road traffic fatalities. Computer-aided recognition of traffic signs has now a long record of related academic and industrial research projects and the first industry-scale applications are now emerging on the market.

Even if the results are not perfect for the present traffic signs, using a different set of signs with fewer similarities between could greatly improve the recognition rate. Supposing that the future transportation is done with autonomous vehicles, certainly the used signs will be made to be easily detected computationally and not specifically for the human eye.

REFERENCES

- [1] **Piccioli, G., De Micheli, E., Parodi, P., and Campani, M.,** 1996 "A Robust Method for Road Sign Detection and Recognition," *Image and Vision Computing*, vol. 14, no. 3, pp. 209- 223.
- [2] **Estevez, L. and Kehtarnavaz, N. A.,** 1996 Real-Time Histogrammic Approach to Road Sign Recognition. *Proceedings of the IEEE Southwest Symposium on Image Analysis and Interpretation*, 95- 100.. San Antonio, TX.
- [3] **de la Escalera, A., Moreno, L. E., Salichs, M. A., and Armingol, J. M.,** "Road Traffic Sign Detection and Classification," *IEEE Transactions Industrial Electronics*, vol. 44, no. 6, pp. 848-859, 1997.
- [4] **Yuille, A. L., Snow, D., and Nitzberg, M.** 1998 Signfinder: Using Color to Detect, Localize and Identify Informational Signs. *International Conference on Computer Vision (ICCV98)*, 628-633. Bombay, India.
- [5] **G. Guo, S. Z. Li and K. Chan,** 2000 "Face recognition by support vector machines", In *Proceedings of the 4th IEEE International Conference on Automatic Face and Gesture Recognition*, 196-201.
- [6] **B. Heisele, P. Ho and T. Poggio,** 2001 "Face recognition with support vector machines: global versus component-based approach", In *Proceedings of the 8th International Conference on Computer Vision*, 688-694.
- [7] **G.-D. Guo, H.-J. Zhang and S. Z. Li.** 2001 "Pairwise face recognition", In *Proceedings of the 2001 International Conference on Computer Vision*, 2:282-287.
- [8] **D. Maltoni, D. Maio, A. K. Jain, S. Prabhakar,** 2003 "Handbook of Fingerprint Recognition", Springer,.
- [9] **M. Sonka, V. Hlavac, R. Boyle,** 1998 "Image Processing, Analysis, and Machine Vision", 2nd Edition, Thomson-Engineering,.
- [10] **M. Sato, S. Lakare, M. Wan and A. Kaufman:** 2000 "A Gradient Magnitude Based Region Growing Algorithm for Accurate Segmentation", In *Proceedings of the International Conference on Image Processing*, 3:448-451.
- [11] **M. Mirmehdi and M. Petrou,** 2000 "Segmentation of Color Textures", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(2):142-159.
- [12] **Y. Xia, D. Feng, T. Wang, R. Zhao and Y. Zhang,** 2007 "Image segmentation by clustering of spatial patterns", *Pattern Recognition Letters*, 28(12):1548-1555.
- [13] **L. Vincent and P. Soille,** 1991 "Watersheds in digital spaces: An efficient algorithm based on immersion simulations", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(6):583-598.
- [14] **C. Stauffer and W. E. L. Grimson,** 1999 "Adaptive background mixture models for real-time tracking", In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, 2:246-252.
- [15] **M. D. Fairchild,** 2005 "Color Appearance Models", 2nd Edition, Wiley-IS&T, Chichester, UK,.

- [16] **I. Ulusoy and C. M. Bishop**, 2005 “Generative versus discriminative methods for object recognition”, In Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition, 2:258-265,.
- [17] **I. Masaki**, “Vision-based vehicle guidance”, Springer-Verlag New York, New York, USA, 1992.
- [18] Mobileye vision system for driver assistance, <http://www.mobileye.com/default.asp?PageID=202>, [last accessed February 16, 2009].
- [19] **Opel Cars Can See**: 2009 Opel Eye camera reads signs, improves safety", 18 June 2008, [last accessed: February 16.
- [20] **C. Bahlmann, Y. Zhu, V. Ramesh, M. Pellkofer, T. Koehler**, 2005 “A System for Traffic Sign Detection, Tracking, and Recognition Using Color, Shape, and Motion Information”, In Proceedings of the IEEE Intelligent Vehicles Symposium, 255-260.
- [21] **W. Ritter**, 1992 “Traffic Sign Recognition in Color Image Sequences”, In Proceeding of the IEEE Intelligent Vehicles Symposium, 12-77,.
- [22] **G. Piccioli, E. De Micheli, P. Parodi and M. Campani**, 1997 “A Robust Method for Road Sign Detection and Recognition”, Image and Vision Computing, 14:209- 223,.
- [23] **K. Crammer and Y. Singer**, 2002 “On the algorithmic implementation of multiclass kernel-based vector machines”, Journal of Machine Learning Research, 2:265- 292,.
- [24] **W. Hao and J. Luo**, “Generalized Multiclass AdaBoost and its Applications to Multimedia Classification”, In Proceedings of the 2006 Computer Vision and Pattern Recognition Workshop, 113, 2006.
- [25] **L. Breiman, J. H. Friedman, R. A. Olshen and C. J. Stone**, 1984 “Classification and Regression Trees”, Wadsworth International Group,.
- [26] **L. Breiman**, 1996 “Bagging predictors”, Machine Learning, 24(2):123-140,.
- [27] **Y. Freund and R. E. Schapire**, 1999 “A short introduction to boosting”, Journal of Japanese Society for Artificial Intelligence, 14(5):771-780,.
- [28] **N. Otsu**, 1994 “A threshold selection method from gray-level histogram”, IEEE Transactions on System Man Cybernetics, vol. SMC-9, n° 1, pp. 62-66, April.
- [29] Carlos Filipe Moura Paulo, 2007 “Detection and Recognition of Traffic Signs”, September.
- [30] **C. Harris and M.J. Stephens**, 1988 “A combined corner and edge detector”, Alvey Vision Conference, pp 147-152,.
- [31] **G. Loy, and A. Zelinsky**, 2003. “Fast Radial Symmetry for Detecting Points of Interest”, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 25, n° 8, pp 959-973, August 2003.
- [32] **Farzin Mokhtarian**, 1995 “Silhouette-Based Isolated Object Recognition through Curvature Scale Space”, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 17, n° 5, May.
- [33] **Farzin Mokhtarian, Sadegh Abbasi and Josef Kittler**, “Robust and Efficient Shape Indexing through Curvature Scale Space”, Department of Electronic & Electrical Engineering, University of Surrey, England.

- [34] **B. S. Manjunath, Philippe Salembier, Thomas Sikora**, 2002 “Introduction to MPEG-7: Multimedia Content Description Interface”, ISBN: 978-0-471-48678-7, Hardcover, 396 pages, April.
- [35] **Farzin Mokhtarian, Sadegh Abbasi and Josef Kittler**, 1996 “Efficient and Robust Retrieval by Shape Content through Curvature Scale Space”, Proceedings of International Workshop on Image Databases and Multimedia Search, pp 35-42, Amsterdam, the Netherlands.





CURRICULUM VITA

SARDAR OMAR RAMADHAN

Zakho – Iraq

Serdaromer1990@gmail.com

PERSONAL DETAILS:

Birth Date:	15/03/1990
Birth Place:	Zakho- Iraq
Gender:	Male
Marital Status:	Married
Telephone number:	00964-750-4517163

EDUCATION:

2009 - 2013	B.Sc. at University Of Zakho – Computer Science Department
2015 - 2017	MSc at University Of Firat - Computer Engineering Department

EXPERIENCE:

2013 - 2015	Lecturer and Research Assistant Computer Science Department University Of Zakho
-------------	---