

TRAFFIC SIGNS SEGMENTATION ON VIDEO FRAMES

**A THESIS SUBMITTED TO THE GRADUATE
SCHOOL OF APPLIED SCIENCES
OF
NEAR EAST UNIVERSITY**

**By
DEHA DOĞAN**

**In Partial Fulfillment of the Requirements for
the Degree of Master of Science
in
Information Systems Engineering**

NICOSIA, 2018

**TRAFFIC SIGNS SEGMENTATION ON VIDEO
FRAMES**

**A THESIS SUBMITTED TO THE GRADUTE
SCHOOL OF APPLIED SCIENCES
OF
NEAR EAST UNIVERSITY**

**By
DEHA DOĞAN**

**In Partial Fulfillment of the Requirements for the
Degree of Master of Science
in
Information Systems Engineering**

NICOSIA, 2018

DEHA DOĞAN: TRAFFIC SIGNS SEGMENTATION ON VIDEO FRAMES

**Approval of Director of Graduate
School of Applied Sciences**

Prof. Dr. Nadire ÇAVUŞ

**We certify this thesis is satisfactory for the award of the degree of Masters of
Science in Information Systems Engineering**

Examining Committee in Charge:

Assoc. Prof. Dr. Kamil Dimililer

Department of Automotive Engineering,
NEU

Assist. Prof. Dr. Yöney K. Ever

Department of Software Engineering, NEU

Assist. Prof. Dr. Boran Şekeroğlu

Supervisor, Department of Information
Systems Engineering, NEU

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

Name, Last name:

Signature:

Date:

ACKNOWLEDGMENTS

First of all, I would like to give thanks to my supervisor, Assist. Prof. Dr. Boran Şekeroğlu for his continuous and unceasing support, help and knowledge.

Also, I would like to thank all the engineering staff at Near East University Department of Information Systems Engineering.

I especially would like to give my thanks to my family and friends, E. Mete Yalçın, Ezgi Özyalçın, Şerife Kaba, Oğuz Ekrem, and Musa Sani Musa for their constant and unceasing support during the preparation of this work. Their patience, guidance and vast knowledge was extremely valuable for the completion of this work. Sincerely, Thank you all.

To my parents...

ABSTRACT

Traffic signs provide a system which organized the stream for a safe road trip and also give information to drivers and pedestrians. Traffic signs are crucial and valuable because these signs warn to drivers about the traffic rules such as speed limits, crosswalk and stop sign.

Some drivers can lose their attention for analysis the traffic signs and these can cause a traffic accident Situation such as these leads to loss of life and property. In this thesis a system is proposed which provides the detection and recognition of the traffic signs to prevent such as these situations.

In this thesis image processing techniques were used on video frames. These techniques were object capturing and point feature matching, applied to several videos recorded on the moving car to capture and recognize traffic signs videos for the driver's vision.

The system is tested by using the of six (6) traffic signs for each of the video frames. On the analysed data proposed system obtained minimum success rate % 63.09 and maximum success rate % 97.56 of the six traffic signs that were detected.

Keywords: Traffic signs; Image processing; Point Feature Matching; Detection of Futures.

ÖZET

Trafik işaretleri, güvenli bir yol gezisi için akışı düzenleyen ve ayrıca sürücülere ve yayalara bilgi veren bir sistem sağlar. Trafik işaretleri önemli ve değerlidir çünkü bu işaretler, sürücülere hız sınırları, yaya geçidi ve dur işareti gibi trafik kuralları hakkında uyarır.

Bazı sürücüler, trafik işaretleri ve analizleri için dikkatlerini kaybedebilir ve bunlar bir trafik kazasına yol açabilir. Bunlar gibi durumlar yaşam ve mal kaybına yol açar. Bu tezde, bu gibi durumları önlemek için trafik işaretlerinin tespit ve tanınmasını sağlayan bir sistem önerilmiştir.

Bu tezde video karakterlerinde görüntü işleme teknikleri kullanılmıştır. Bu teknikler, nesne yakalama ve nokta özellik eşlemesidir. Sürücü görüş açısına yönelik hareket halindeki arabada birkaç video kaydedilmiş ve bu videolar üzerinde trafik işaretleri yakalama ve tanımlama işlemi yapılmıştır.

Sistem video çerçevelerinin her biri için altı (6) trafik işaretleri kullanılarak test edilir. Analiz edilen veriler üzerinde önerilen sistem, tespit edilen altı trafik işaretinin asgari başarı oranı % 63.09 ve maksimum başarı oranı % 97.56 elde etmiştir.

Keywords: Trafik işaretleri; Görüntü İşleme; Nokta Özelliği Eşleştirme;

TABLE OF CONTENTS

ACKNOWLEDGMENTS	viii
ABSTRACT	x
ÖZET	xi
TABLE OF CONTENTS	xii
LIST OF FIGURES	xiv
CHAPTER 1 INTRODUCTION	1
1.1 Introduction	1
1.2 Advanced Driver Assistance Systems	1
1.3 The Aim of the Thesis	2
1.4 Thesis Overview	3
CHAPTER 2 TRAFFIC SIGNS AND LITERATURE REVIEW	4
2.1 Traffic Signs	4
2.2 Properties of Traffic Signs.....	4
2.3 United Kingdom Road and Traffic Signs	5
2.4 Literature Review	8
CHAPTER 3 COMPUTER IMAGING	12
3.1 Computer Imaging	12
3.2 Image Processing	12
3.3 Imaging Representation	15
3.3.1 Gray-scale images.....	15
3.3.2 Colour images.....	16
3.4 Pattern Classification.....	18
3.5 Digital Video	19
3.6 Frame concept.....	20
CHAPTER 4 PROPOSED SYSTEM	22

4.1 Database.....	22
4.2 Proposed System.....	23
4.2.1 Video to Frames.....	24
4.3 Detect and Extraction Features.....	25
4.4 Outline and Algorithm Overview	25
4.4.1 Speeded-Up Robust Features.....	27
4.4.2 Scale- Invariant feature transform	30
4.4.3 Scale-space local extrema detection	31
4.4.4 Keypoint localization.....	34
4.4.5 Orientation appointment	34
4.4.6 Key point descriptor	35
CHAPTER 5 RESULTS AND DISCUSSIONS	37
5.1 Results	37
5.2 Discussions	41
CHAPTER 6 CONCLUSION AND SUGGESTIONS	42
6.1 Conclusion	42
6.2 Suggestions	42
REFERENCES	43
APPENDICES.....	46
APPENDIX 1 Program Interface	47
APPENDIX 2 Program Listing	48

LIST OF FIGURES

Figure 2.1: Warning signs	5
Figure 2.2: Prohibitory signs	6
Figure 2.3: Mandatory signs	7
Figure 2.4: Indicatory and Supplementary signs	8
Figure 3.1: Image restoration (Umbaugh, 2005)	13
Figure 3.2: Noise removal	14
Figure 3.4: Schematic of the RGB color cube. (Gonzalez and Woods, 2002).....	17
Figure 3.5: RGB 24-bit color cube. (Gonzalez and Woods, 2002)	18
Figure 3.6: Video is composed a series of still image. Each image is composed of individual lines of data. (Jack, 2005).....	20
Figure 3.7: Progressive Display “Paint” the lines of an Image Consecutively (Jack, 2005).....	21
Figure 4.1: Frame sample.	22
Figure 4.2: Proposed system represented by Block Diagram.....	23
Figure 4.3: RGB to Grayscale	25
Figure 4.4: Box filter representations of Gaussian second order partial derivative.	27
Figure 4.5: SURF key points detected in a traffic sign image.....	28
Figure 4.6: The demonstration of descriptor building (Bay et Al., 2006).....	29
Figure 4.7: Matching of the fast index.	29
Figure 4.8: Example of point matching result.	30
Figure 4.9: This extraction process done by SIFT.	31
Figure 4.10: Gaussian pyramid (Lowe, 2004).....	33
Figure 4.11: Scale space for keypoint (Lowe,2004).....	34
Figure 4.12: SIFT keypoint descriptor (Lowe,2004).....	35
Figure 4.13: A traffic sign image including SIFT keypoints detected by the program.	36
Figure 5.1: Example frames for video 4	39

Figure 5.2: Video 4 timeline.....	40
Figure 5.3: Video 1 timeline.....	40

CHAPTER 1

INTRODUCTION

1.1 Introduction

In the latter years, the vehicle use by the population is increased thus, the amount of vehicles on the roadways has been increased. For this reason, road traffic accidents rise dramatically in all over the world (Swashi and Suresh 2017).

Traffic signs help road users to give information about the road directions, traffic rules and provide a safe road trip for drivers (Swathi and Suresh, 2017). Drivers must obey the traffic rules for a safe driving experience for this. They must follow the traffic signs. There are four essential types of traffic signs which are prohibition, warning, informative and obligation. Each type of traffic sign has different properties depending on their pattern and the colour. Patterns of the signs demonstrate a circle. A border of the circle has a red colour and a white or blue background. An equilateral triangle shows the warning signs with a peak upwards. The warning signs are coloured by a white background and red borders. Both prohibition signs and warning signs have a background with a yellow colour when they are located in a public road. Informative signs have the similar colours with a blue background in circles that illustrate an obligation. There are 2 cases which does not follow the previous pattern of traffic signs. One of them is the stop signs and other is yield signs. The stop signs have a hexagonal shape. Yield signs have an inverted triangle shape with using these are properties, you can easily detect and recognize a sign.

1.2 Advanced Driver Assistance Systems

National Highway Traffic Safety Administration (NHTSA) has a research on relation of human error in vehicle crashes and found the percentage as 94% concluding, the main reason and cause of vehicle crashes are because of human error. If we want to concentrate on most urgently important driver-related errors, these are, decision errors, recognition errors and performance errors that can end in a disastrous accident. ADAS is a system that can provide valuable and urgently important information regarding the environment, road and situation of the traffic to the driver and it is called as ADAS (Advanced Driver Assistance Systems) and this system can increase the overall safety of drivers and pedestrians by taking control

of some repetitive and complex tasks. By the research that is done by NHTSA we can come to conclusion on developing and implementing technologies to prevent and avoid these disastrous accidents is crucial. In the recent years it is common to come across with vehicles that are equipped with ADAS and there are many examples that can be given for the last 20 years, as a very well-known and popular example we can give is the GPS (Global Positioning Systems). If we look at the more recent years we can give other popular examples and these can be named as, adaptive cruise control, automatic braking systems (ABS), automatic parking, adaptive light-beam and headlight control, collision avoidance systems, driver drowsiness detection, blind spot detection, night vision, lane departure warning systems, hill descent control and many more, the aim and purpose of these systems is to make the traffic and road that is used by the vehicle to be safer for both drivers and pedestrians. Nonetheless these systems in particular that we mention pay little to none attention to the behavioral modes that are presented by the driver. The purpose in this thesis concentrates on the driver-gaze behavior and the implementation of TSDR (Traffic Sign Detection and Recognition) that is capable of identifying traffic and road signs and informing the driver if they have not seen the specific traffic signs.

1.3 The Aim of the Thesis

Traffic and road signs help the drivers by giving information about the rules of road for a safe and sound journey both for pedestrians and the people in the vehicle and as mentioned in the topic above there are plenty of Advanced Driver Assistance Systems to help the drivers for a safe trip. For this thesis I used a system called Traffic Sign Detection and Recognition (TSDR) and the purpose of this thesis is to improve and advance in a system that is utilized for traffic and road signs.

Image processing methods have been implemented to execute this system. At first, videos were recorded using a digital camera in a moving vehicle. Then, these videos were converted into frame by frame images where pre-processing techniques are performed for each of the frames. These frames were RGB (red, green, blue) images but for this system to work, they are compressed into grayscale images. This method was applied to create each of feature points according to given traffic sign features. In the next step, the traffic signs were compared according to given feature points. In this thesis, feature point matching method

and Speeded up Robust Feature (SURF) were used to detect and recognize the traffic signs in various classes of the traffic signs.

1.4 Thesis Overview

Main parts of the thesis are as shown below:

- Chapter 1 presents the introduction to the thesis and gives information about the advanced driver assistance systems.
- Chapter 2 explains general information about and traffic signs types and the literature review related to traffic sign detection and identification system.
- Chapter 3 gives information related to computer imaging and image processing methods.
- Chapter 4 demonstrates traffic sign detection and identification system that were applied in this thesis.
- Chapter 5 is the results and discussion.
- Chapter 6 is the research part of this thesis that represented with an insight into the suggestions for future work and improvement.

CHAPTER 2

TRAFFIC SIGNS AND LITERATURE REVIEW

2.1 Traffic Signs

Road users (vehicles and pedestrians) are informed and instructed by specific signs placed at the above or side of the road and these signs are called as “Traffic Signs” or “Road Signs”. The primitive road signs were made of stones (milestones) or simply made of wood. By time, the directional arms were designed for these signs, we can still see examples of these in United Kingdom as fingerposts.

As mentioned in the paragraph above, traffic and road signs are used for various reasons such as, controlling, guiding, warning or informing the users of the roads (vehicles and pedestrians). Traffic signs, provide and accomplish an appropriate level of traffic quality and safety of the road for an orderly flow of traffic, both for pedestrians and vehicles (Fang,C.). Road and traffic signs are created to recognised by drivers essentially because their colors and shapes are easily perceptible from their regions Hoose.N). According to the European signs, all signs are formed to have a layer capable of reflecting light combined with a selective section of the sign. Most United Kingdom road signs utilize pictograms to express the information and message given by the traffic sign.

2.2 Properties of Traffic Signs

Traffic signs and road signs are defined by a number of features that can make them distinguishable according to the environment around them. In all over the country, the road signs are designed by utilizing standard fonts of text, and magnitude of characters. Road signs are created, constructed and established with strict regulations. They are formed in 2-D shapes for instance octagons, circles, triangles, or rectangles. The surrounding environment is considered when choosing colours that are used in the traffic signs because the colours should be contrasting. Therefore, drivers can effortlessly distinguish and recognise the signs [(Jiang, G. Y., & Choi,). The colours are organised according to the sign category.

2.3 United Kingdom Road and Traffic Signs

Traffic signs and road signs in the United Kingdom can be ideogram-based (pictogram-based) involving basic ideographs to represent the meaning of a sign, or text-based whereas the properties of the sign might be arrows, text or the different types of characters and symbols. United Kingdom traffic signs are observed and classified in four different groups:

- Warning signs: Figure 1.1 illustrates a group of traffic signs that defines a hazard or a threat ahead over on the road. It is symbolized by an equal-sided triangle with a thick and prominent red border and a white background. A pictogram (ideogram) is utilized to indicate in different types of warnings. Another signs like the YIELD sign and track level crossing sign also belong in this class and distance to level crossing signs. Yield signs have an inverted triangle shape as shown in Figure 1.1 section (e).



a) General caution



b) Road work sign.



c) Bicycles crossing ahead sign



d) Zebra crossing ahead sign



e) Yield sign

Figure 2.1: Warning signs

- Prohibitory signs: These signs are used to prohibit specific kind of maneuvers or some traffic types. The speed limit, no parking, and no entry signs pertain to these categories that are illustrated in Figure 2.2. Generally, the prohibitory signs are made in a circular pattern with a thick and prominent red rim around them with a white interior but, there are some various exceptions such as, stop, no parking and no standing signs. the stop sign is a prominent red colored octagon with a white rim around. the no standing and the no parking signs are blue with a red prominent rim around.



Figure 2.2: Prohibitory signs

- Mandatory signs: These signs can be described by a completely blue circle and a white colored arrow or a pictogram, Figure 2.3 shows the types of mandatory signs. They check the performance of road users and drivers.

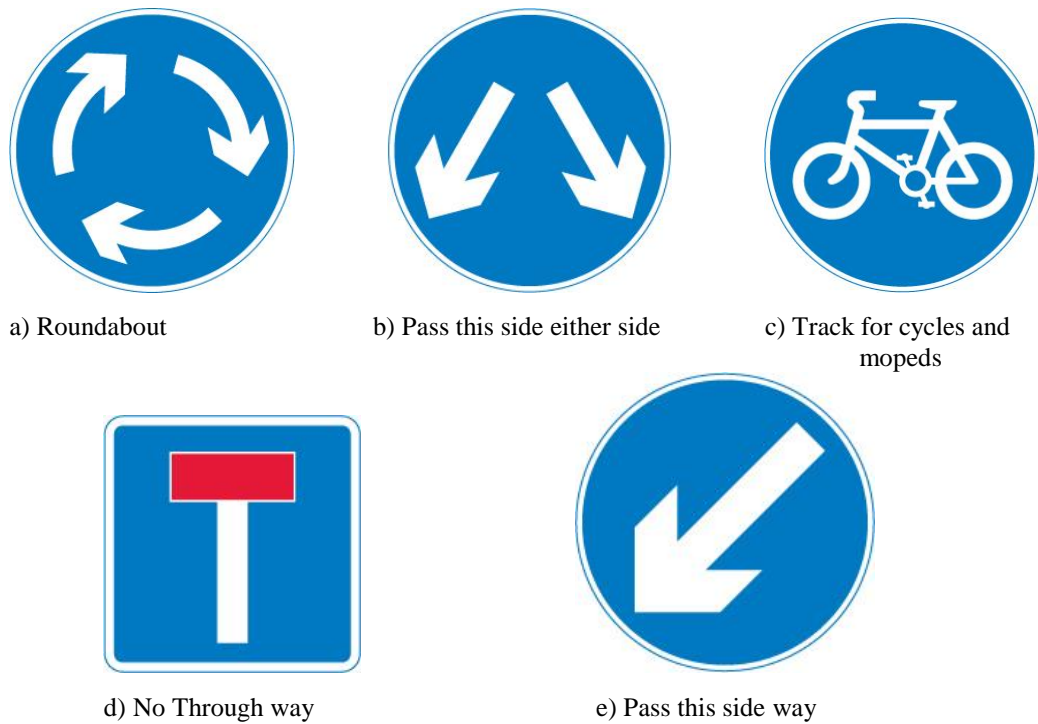


Figure 2.3: Mandatory signs

- **Indicatory and Supplementary signs:** These signs can be described by using rectangles with various colors as their backgrounds, such as blue, yellow or white. Figure 2.4 shows some of the traffic signs that are related to this category. The pictograms are either black or white. This category can contain diamond shaped quadrangles and the signs presenting road priority information.

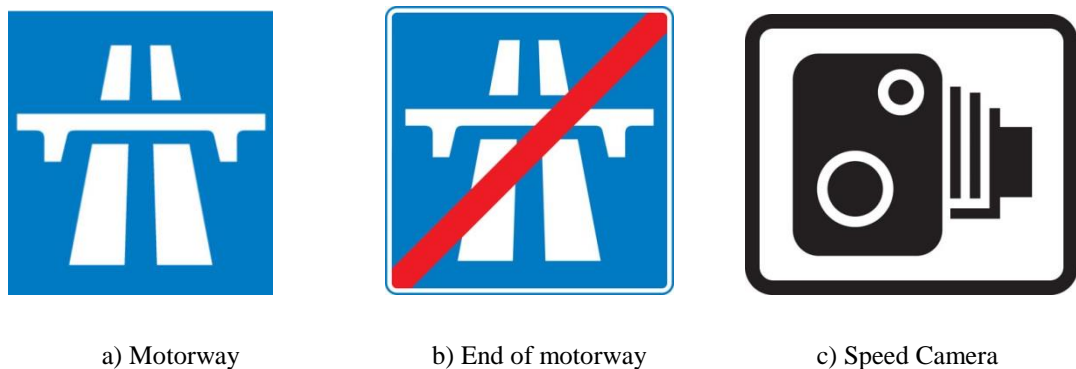




Figure 2.4: Indicatory and Supplementary signs

- As mentioned so far, the colours, shapes and patterns of the traffic and road signs identifies that specific traffic signs category. The colors which are used on traffic and road signs are in specific wavelengths in the visible electromagnetic spectrum. They are chosen to be recognizable from the natural and manmade surrounding environment, so that the users of the roads (vehicles and pedestrians) can easily identify these traffic signs.

2.4 Literature Review

There has been a study on identification and detection of the traffic signs published by Yoon and Lee in 2012. The identification of traffic sign is one the most critical issue for the Driver Assistant Systems (DAS). DAS gives critical knowledge and data about a safe driving. The first stage of detection involves color segmentation for a given color region. A color threshold was applied to segment the potential traffic signs. Then the Speeded up robust features (SURF) matching technique was used to compare the potential signs and template signs from the database that was utilized 1280 images with 5 different videos were used for the proposed algorithm. The second stage of the study which was the identification stage, the SURF algorithm was applied for the General purpose computing on graphics processing units (GPGPU) system. Experimental results demonstrated the technique that was developed in this research, performs effectively and robustly for the chosen data.

Drivers who have deficiency to follow the traffic signs mainly the stop signs, has led to severe traffic accidents. The detection of video-based traffic sign plays a significant role of

driver assistance systems. In the previous systems, shape based, and colour detection techniques were applied. In recent years, feature based traffic sign detection algorithms are preferred to get more scientific results. The study of by Zhu and Huang (2013) emphasized a real-time detection of traffic sign using SURF features in Field-programmable gate array (FPGA). A Xilinx KC705 FPGA platform was used in this study. The hardware section of the study was created for processing video streams with 800 x 600 resolutions in every passing second for 60 frames. The results of this study presented real time Super video graphics array (SVGA) video at 60 frames per second with applying SURF on a Field-programmable gate array for detecting the traffic sign.

Kushal and Oruklu (2015) have created a system that recognizes the robust traffic sign. This system was developed for the applications of driver assistance and self-driving cars. Detection of the traffic sign and classification were combined by two main steps for this system. The sign detection depends on colour segmentation and includes hue detection, morphological filtering and labelling. For the classification of the traffic sign, a nearest classifier was used. Speeded up robust features (SURF) algorithm was applied for removing the training features. For training, there were three feature extraction approaches which were compared to obtain the optimal feature database. This system shows the detection of robust for signs which were rotated. The results of the experiments were done by the system, represented 97.54% detection accuracy.

Jin et al. (2010) have investigated a study to describe the detection of traffic signs, recognition and tracking. Shape and color are integrated to identify and detect signs. The red part of the sign is detected by the application of saturation and hue. Circle shapes are detected and identified over the developed round-degree technique of removing region feature parameters. Kalman filter is proposed to track multiple targets for the following frames. For the recognition of traffic sign, feature extraction technique was used through the depends on 2DPCA (Two-Dimensional Principal Component Analysis). Experiments are carried out for two traffic sign databases with Euler distance and the proximate neighbour classifier. One of these databases is the image library which images are acquired from a multiple array of simulation transformation after applying image binarization method. Other data base is composed of images shot of real scenes with choosing different types of location scenes. This method has a good impact for recognition of the both images. The highest recognition

percentage might be 100% for the database which were taken from a series of transformation. For real scenes images which were shot in various environment, the highest recognition rate might be 98%.

Most of the traffic sign recognition algorithms use Template matching that contrast detected signs and its templates. In the research of Fitriyah et al. (2017), edge detection and Eigen-face methods were used for traffic sign recognition. A remarkable accuracy of recognition has been achieved by the surveys of this method. However, the matching of the template has trouble in the usage of memory as it needs to store various templates. For the recognition of the faces, Eigen-face is an essential technique. It is significant and useful as system only requires storing an Eigen-face image and Weights. In this study, traffic sign recognition system was developed with using Eigen-face algorithm. Alternatively, the using of RGB images, edge-detected images were used. This is a more specific feature for the comparison of the intensity of colour that varies from red, blue, yellow in addition to the black symbols and signs. First of all, the templates signs were converted from RGB to grayscale intensity. Then, by using Sobel Operator, edges were detected and organized into a single matrix. The next step was to determine the Eigen-vectors and Eigen-values of the covariance of the matrix. The highest Eigen vector was chosen and designed as an Eigenface image. Every traffic sign has a specific Weight related to the Eigen-face image which can be applied for recognition. The pre-classification test is a kind of test used in Hue demonstration better weights' differentiation yet lower in weights 'distribution. This differentiation is significant to decrease the false-recognition to other type of traffic signs, even though the diffusion within types needs to be increased. However, whether or not using Hue pre-classification, the Eigen-face that utilizes the edge information was able to yield different weights for each traffic signs. So that, the traffic signs will be able to be recognized by the algorithm.

The study of Fu et al. (2016) has developed an automatic traffic sign recognition method by using salient region segmentation based on geodesic distance transform. First of all, Visual-based traffic sign recognition (TSR) needs detection and then the signs should be classified from the image which is captured. Similarly, for a cascade system, the accuracy of classification is afflicted from the detected results. This study introduces a technique for removing a salient area of traffic sign with a detection window for the purpose of a more accurate description and feature extraction, to increase the well execution of this

classification. In the first stage, the proposed method, a super-pixel based distance map is created by using a sign geodesic distance transform from a group of chosen background seeds and foreground seeds. In the next stage, an effective technique which derives a final segmentation through the distance map. The distance map is converted into segmentation proposals by choosing the optimal proposals which is most appropriate to get a certain shape. With applying these two stages, this method can remove salient sign regions of different types of shapes. The proposed method is analysed and verified with a completed traffic sign recognition(TSR) system. For this system, 30 Chinese prohibit or signs and 420 street images were trained by the classifier whereas 693 signs were detected accurately. The experimental results presented that the proposed technique gives a 97.11% accuracy rate for the dataset including street images.

CHAPTER 3

COMPUTER IMAGING

3.1 Computer Imaging

Devices and machinery, developed with innovative mind-set and scientific knowledge, simply put “technology” is the ultimate convenience that human life needs in our age, especially computers. Computers are being used in every field in human life, in the field of economy, health, governmental services even in the field of art.

One of the many ways that computers are used is image processing and it is one of the most substantial way that we use computers because it addresses our primary sensory ability, “Vision”. Computer imaging helps us understand and evaluate complex data by using our ability of vision.

Computer imaging helps us deliver digital data through images by collecting and processing a digital data or visual data into a version of image that we see fit for the work we are doing by using methods like compression and segmentation.

The results we get from computer imaging can be from a tomography device in a hospital or the camera footage with face recognition from a security camera.

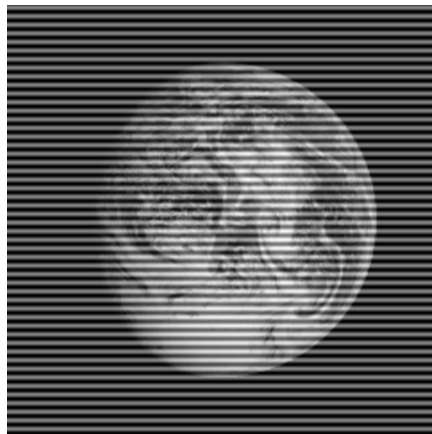
3.2 Image Processing

Image processing is manipulation of digital images with analysis by specialized software but this software is operated by people. Other than knowing how human visual perception works, analysing and manipulation methods of these software should be well understood by the people who operate this software. Image processing system works by compression, restoration and enhancement of the raw image data.

Digital image is understood by the computer by a 2 dimensional function $f(x, y)$ - “f” is representing “function” and “x, y” are representing the extent of this function at each sequent of the plane while vitality rates of the function staying limitless.

A digital image is composed of several components, each with a value and area specific to them. These components are pixels, pels, picture and image elements, where pixels are the components that are shown to the person on a digital screen.

Image restoration is a technique often mostly used in a range of photography and is a process of obtaining an image with an estimated degradation and reconstructing it to its original form. It is simply improving a degraded image before printing. These types of applications require a knowledge about degradation for developing a model for the distortion. When a model of distortion is developed for degradation a process of inversing the image to reconstruct it to its original appearance can easily be applied. Normally, these types of images are used in space surveys such as equalizing the flaws in the optical system of a telescope or selecting artefacts formed through mechanical pulsation in a spacecraft. Moreover, restoration process could be used in adjusting geometric distortion or in removing noise. (Umbaugh, 2005).

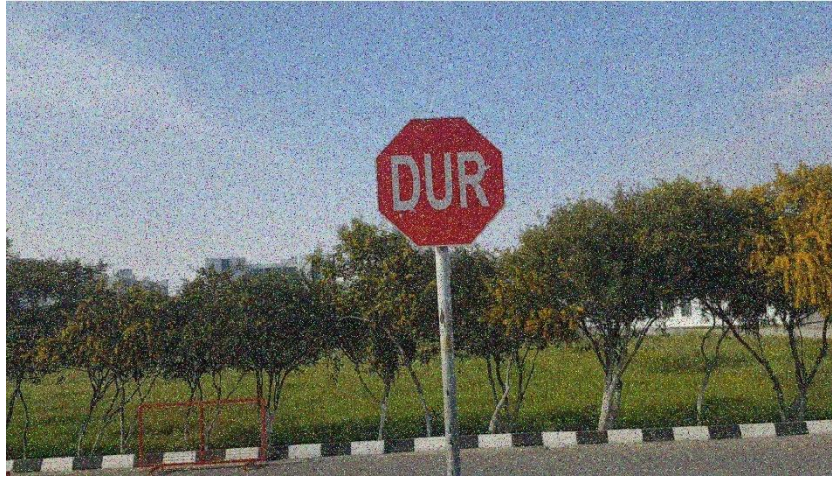


(a) image with distortion



(b) image with restoration

Figure 3.1: Image restoration (Umbaugh, 2005)



(a) Image with noise



(b) Noise removed by using image restoration

Figure 3.2: Noise removal

Image enhancement is a technique of improving an image visually through using the human visual system's reaction. One of the basic and effective technique of image enhancement is known as stretching the contrast of an image also sharpening an image. However, these methods sometimes could be problematic since a technique used for enhancing satellite images man not be convenient for images used in medical field. Both techniques restoration and enhancement is primarily used for improving an image but there is a difference of approaching a problem between each. Restoration method is used model the distortion to the image and corrects this degradation whereas enhancement uses the information of human visual system reactions to develop images in terms of visually. (Umbaugh, 2005)

Image compression is a process to reduce the amount of data required to restore an image. It is applied by selecting unnecessary visual data and using the option of redundancy which typically exists in many images. This system is applied in computer vision systems but it is undertaken as an image processing technique since the process is being held by people who actually study the images. Therefore, it is required to understand the most important part of image data for comprehension. With the help of psychology and physiologic aspects of human visual system image data can be reduced 10 to 50 times, and videos can be reduced 100 or 200. Figure 3.2 presents several degrees of compression applied to an image. It should be considered that the amount of compression and the quality of the image compressed is depending on an image used and will differ externally. (Umbaugh, 2005)

3.3 Imaging Representation

The digital image ($I(r, c)$) is made by 2D cluster of data which each and every pixel adjusts the brightness of the picture at the point (r, c) . The digital image model utilized here -which is a 2D data cluster- $I(r, c)$ is considered as a matrix and one section is named as a vector in direct polynomial terms. The data of this digital image is in gray-scale which could be depicted as monochrome. Nonetheless, there are different kinds of digital image data which should have been broadened and altered for this model. Most of the time, these image data are multi-band images with multi spectrum or in color, which can be demonstrated by $I(r, c)$ function, this function adjusts each and every data by the band of brightness. The images assessed here are, multispectral, colour, binary and gray-scale.

3.3.1 Gray-scale images

In Gray-scale images, there is only brightness information and the image is formed without the colour information, by forming the image by pixels of different brightness levels, in our perspective of vision they are only formed by 2 colours, white and different shades of black/gray that's why they are called monochrome images.

Basic Gray-scale image consists of 8-bit per-pixel data, which shows different brightness levels between "0" and "255" and in our perspective of vision as mentioned above, this type of image formation provides a more efficient resolution of brightness with an increased noise margin by giving more gray levels than a colour image.

3.3.2 Colour images

To know how an image with colour is formed we have to understand how the bands and band patterns work together to form the digital image. Colour images can be thought as three-band monochrome formation, where each data of red, green and blue combines and forms different colours. At the end, the information gained by the computer through the digital image and what is shown to the screen is the information of brightness of each colour as red, green and blue formed in each spectrum of colour and by combination of all.

All this information that are mentioned are compared to the 8-bit monochrome digital image model but, the digital image should be 24-bits/per-pixel where it demonstrates 8-bits for each colour, which are, red, green and blue in the monochrome formation that are forming the colour band. This whole process is named as the colour model.

After this process has been completed, the digital image process creates a matching, two dimensional colour space with one dimensional illumination which helps us to identify different colours in the image.

In the RGB color model, each colour represents its initial spectral ingredients of blue, green and red. The model relies on Cartesian coordinate system. The colour subspace related areas are shown in the cube in figure 3.4 where RGB basic values are black at the starting point, white at the corner of the starting point. In this demonstration, the gray scale which shows the equal values of RGB drags on from black to white through the line that connects these two parts. Other colours are shown inside or on the cube and are described by vectors stretching from the starting point. For relevance, the hypothesis is that value of each colour has been standardized where the cube forms a unit cube in Figure 3.4 where all the values of R, G, and B are considered to be in the range $[0,1]$. (Gonzalez and Woods, 2002)

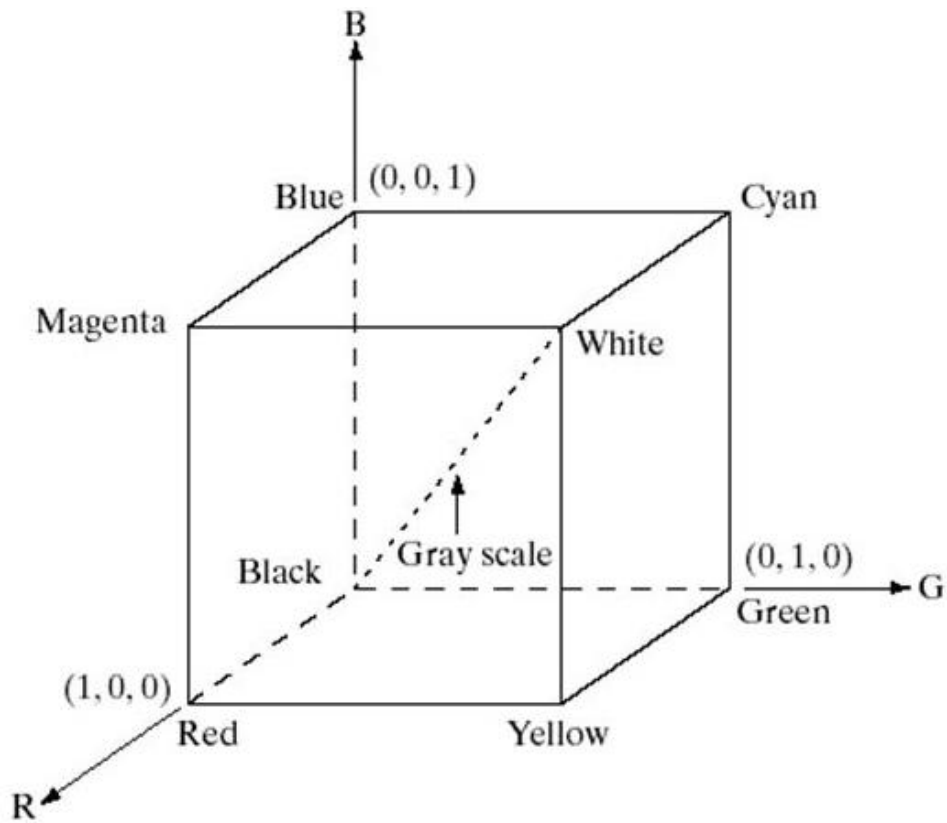


Figure 3.4: Schematic of the RGB color cube. (Gonzalez and Woods, 2002)

Images shown in the RGB colour model are formed by three supplementary images where each represents one primary colour. On RGB monitor these three images originate a composite colour image on the screen. The amount of bits included in each pixel in RGB space is named as the pixel depth. Each colour, red, green and blue in RGB image is an 8-bit image. Therefore, every single RGB colour pixel have 24-bits depth where values for R, G, and B are multiplied by 3. The expression of full-colour image is usually used to identify a 24-bit RGB colour image. The calculation of number of colours in a 24-bit RGB image is $(2^8)^3 = 16,777,216$. The cube represented in Figure 3.5 is formed by the calculation of $(2^8)^3 = 16,777,216$. The most suitable way of displaying these colours is to create planes for each colour; sides or intersections of the cube. This is finalised through adjusting one of three colours and enabling other two to differ. For example; a cross-sectional plane within the centre of the cube and parallel to the GB-plane in the figure shows the lane $(127, G, B)$ for $G, B = 0, 1, 2, \dots, 255$. Instead of mathematically

suitable standardized values in the range $[0,1]$ current pixel values are used here since previous values are the ones used to form colour in computer. (Gonzalez and Woods, 2002)

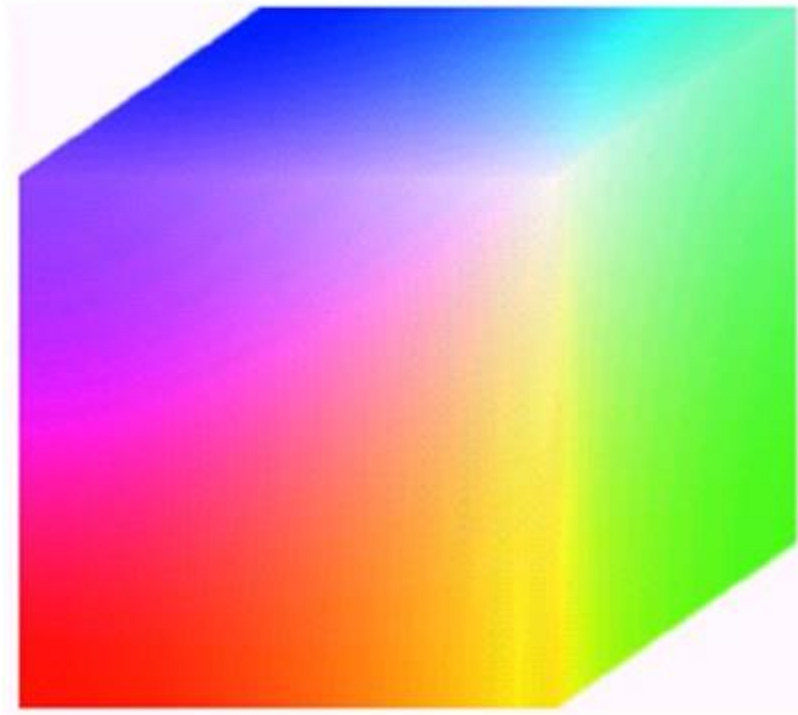


Figure 3.5: RGB 24-bit color cube. (Gonzalez and Woods, 2002)

3.4 Pattern Classification

For pattern classification, we use image analysis of a digital image, using features that are extracted from the original digital image and using those features to automatically classify and describe the objects from the image that are in a certain pattern. For the automatic classification we need to build an algorithm to use the data that belongs to our desired feature and for the algorithm there are some important specifications that should exist, such as distances, shapes, similar forms and measures so that the algorithm can distinguish and exclude or include specific features that we want to be analysed. In a more basic way of explanation, pattern classification is kind of a post-processing stage to future extract and analyses certain data from a specific digital image.

In this section we will examine some of the basic algorithms and widely used methods in detail. As mentioned before, pattern classification is primarily used in image analysis which can be used for image compression, computer vision, data analysis and development of applications that feature these specifications and it is the final step in computer vision algorithm development so that the computer can perform this vision-related task.

These tasks can range from robotic control and all the way to the analysis of medical images for the diagnosis of otherwise complicated cases. After this step, all of the useless data from the image are removed and useful information are compressed. Compressions should include high pixel-value representations of the data we want to analyses.

3.5 Digital Video

Images of moving subjects in fact, arise from a series of still images in a way that creates the perception of continuity in the human visual system changes quickly. Still image sequence is obtained by the spatial and temporal sampling of the natural scene. For example, to produce a motion video signal (e.g. 1/25, 1/30 or 1/60 seconds) repeated at intervals.

A natural scene is always spatial and temporal. Transformation of such a scene into digital format is performed by spatial and temporal sampling. A digital video is a numerical representation of a sampled visual scene.

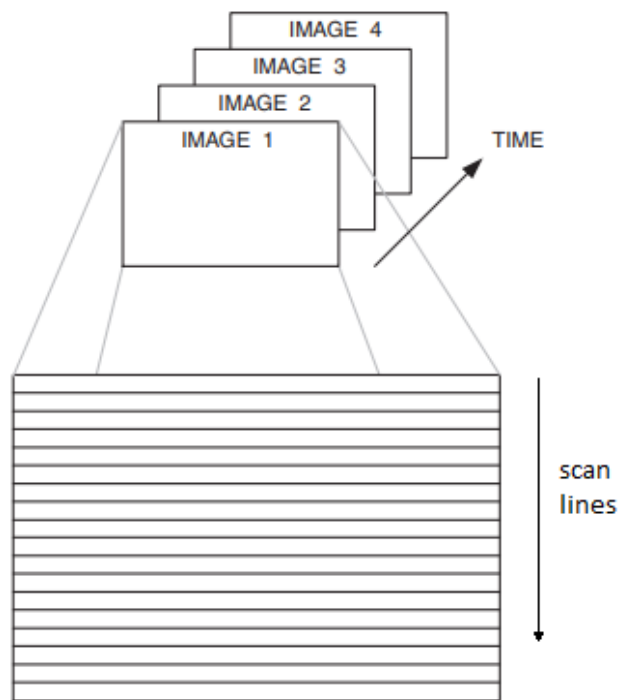


Figure 3.6: Video is composed a series of still image. Each image is composed of individual lines of data. (Jack, 2005)

A natural video scenes of the video processing and distinguished related to the compression characteristics of the scene in which tissue exchange, the number of objects and shapes, colors and similar positional edge and subject movement, changes in lighting, motion or perspectives of the camera, and are similar temporal edge. (Jack, 2005)

3.6 Frame concept

Each of still images creates a video that is called a frame. Frames consist of scan lines that follow all of the frames are in sequence from top to bottom (Figure 3.6). There are two types of schedules used in the video display. Vertical synchronization shows a new frame and horizontal synchronization indicates the starting point of a new scan line.

The horizontal and vertical synchronization information is transmitted in one of three approaches; these are separate vertical and horizontal synchronization signals, composite synchronization signal and composite synchronization signal inserted in the video signal.

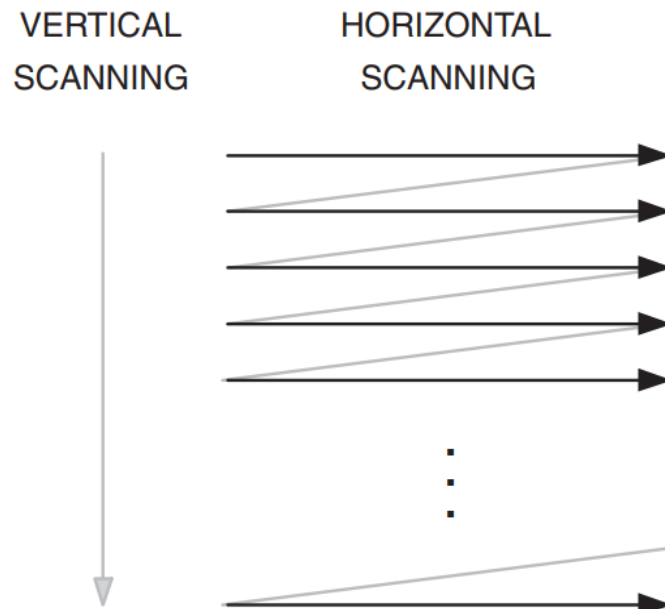


Figure 3.7: Progressive Display “Paint” the lines of an Image Consecutively (Jack, 2005)

The composite synchronization signal is composed of the horizontal and vertical synchronization. Consumer and computer device which uses analog RGB video generally utilizes vertical and horizontal synchronization signals, composite synchronization signal methods. Consumer device which provides composite video or analogous YPbPr video uses composite synchronization signal inserted in the video signal method. A separate vertical and horizontal synchronization signal is mainly used for digital video and also timing code can be embedded in the digital video stream. (Richardson, 2003)

CHAPTER 4

PROPOSED SYSTEM

4.1 Database

In this thesis, 25 videos and 6 different traffic signs were used to test the performance of the system. These videos were collected from North Cyprus. The videos were divided into frames and the numbers of frames are changing depends on duration of the videos.



Figure 4.1: Frame sample.

4.2 Proposed System

A block diagram representation of the steps used in the proposed system is represented in the Figure 4.2.

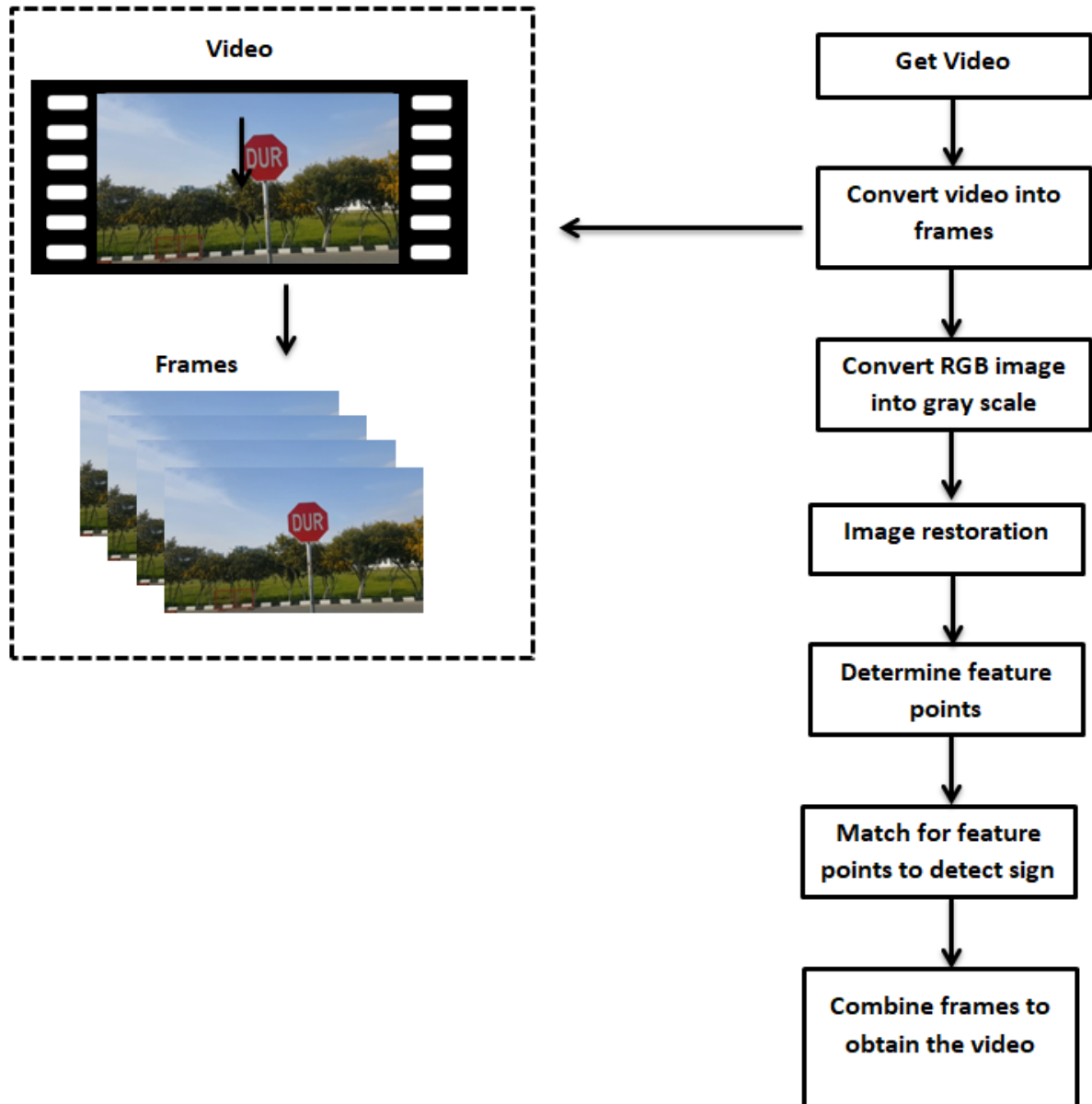


Figure 4.2: Proposed system represented by Block Diagram

4.2.1 Video to Frames

A video is consisting of many images which are captured in quick succession. Each image that makes the video is called a frame, rate at which images are captured specifies video's frame rate.

In order to perform computations on a video file, it must first have to be converted to its primal stage using the metadata information of the video such as rate at which images are captured the primal stage of the video is series of frames. After the conversion, frames then stored in a database with their index in the video sequence. The system proposed in this thesis computes the amount of frames per second using video data gathered. (Idris and Panchanathan,1997)

$$\text{Number of frames} = \text{Frame rate} \times \text{Video duration} \quad (4.1)$$

4.2.2 RGB to Grayscale

In order to convert the colour space of the images from RGB to grayscale the weighted sum for each pixel in each palate of the RGB layers calculated. Weights suggested by ITU (International Telecommunication Union) which are;

$$G = 0.299 \times R + 0.587 \times G + 0.114 \times B \quad (4.2)$$

Formula given above is used to obtain the transformation from colour image to grayscale image. (ITU-R BT.601-7, 2011)



Figure 4.3: RGB to Grayscale

4.3 Detect and Extraction Features

There are certain characteristics exist within the image, and these characteristics such as shape and/or colour are used by identifying these characteristics to describe the object for Feature Extraction. The extracted features of the image are utile extractable aspects of parts, regions or shapes within a digital image. (Gose et al., 2008).

4.4 Outline and Algorithm Overview

The Speed-Up Robust Features (SURF) algorithm is a local feature detector and descriptor and it is based on the same principles and steps as Scale-Invariant Feature Transform (SIFT) nevertheless the details in each and every individual step are disparate. SURF algorithm is a more superior and sophisticated adaptation of SIFT when it comes to identifying key points and local features in a traffic sign image. Both the SIFT and the SURF algorithms have three fundamental components; interest point detection, local neighborhood description and lastly, matching. SIFT algorithm approach employs cascaded filters to distinguish scale-invariant characteristic points, where the DoG (Difference of Gaussians) is imposed on rescaled images increasingly. In the other hand, SURF algorithm approach uses square-shaped filters as an approximation of Gaussian smoothing.

If we want to look at Multi-scale analysis methods such as the SIFT method, the identification and detection of features in Speed-Up Robust Features (SURF) depends on a scale-space representation, combined with 1st and 2nd order deferential operators. The inventiveness and brilliance of the SURF algorithm (Speeded Up Robust Features) is that these tasks are sped up by the operation and use of box filter methods. Thus, we will use a

terminology called “box-space” to distinguish and differentiate in distinction to the normal Gaussian scale-space. In the other hand the Gaussian scale space is achieved by involution of the original images alongside Gaussian kernels, the discrete box-space can also be achieved by convoluting the initial image with box filters at assorted scales.

In the detection step, feature detection is achieved by selecting interest point candidates with the use of the local maxima in the box-space of the determinant of Hessian operator. The selected candidates are then substantiated if the feedback is exceeding a certain accustomed threshold. After these steps, by using quadratic fitting method the location and scale of these candidates are refined. At the end, a few hundred interest points are detected occasionally in a megapixel image.

The objective of feature description phase is to construct a descriptor of the neighborhood of each POI (point of interest) which are proportional to the view-point changes. By the many advantages of multi-scale analysis methods, the selection of the particular points in our box-space caters transition and scale invariance. From the estimation of Haar wavelets, a dominant orientation is determined by taking into account of all the local gradient orientation distributions to enact rotation invariance. Based on 1st order statistics of Haar wavelets coefficients, a 64-dimensional descriptor is built by employing a spatial localization grid.

As the last phase, an image matching task is needed to be done, and this phase is called as, Feature matching. Feature matching can be used in various different situations, such as, object detection, image registration and image indexing. The aim and goal of feature matching is to match local descriptors from various distinct images. In this phase a comprehensive comparison is executed by computing the Euclidean distance between all potential pairs which are matching. To reduce the mismatches and false results, a nearest-neighbor distance-ratio matching criterion combined with an optional RANSAC (Random Sample Consensus)-based technique is employed for geometric consistency checking.

4.4.1 Speeded-Up Robust Features

Speed-Up Robust Features(SURF) is a local feature detector and descriptor, developed by Bay, Tuytelaars, and Van Gool and presented at the 2006 European Conference on Computer Vision (et al. 2006). The SURF algorithm happens to be among the best is for identifying key points of all local features in traffic and road sign images. It is a more sophisticated adaptation of Scale-Invariant Feature Transform (SIFT).

As we specified in Scale-Invariant Feature Transform (SIFT) calculations, for scale-space step rather than Laplacian of Gaussian (LoG), Difference of Gaussian (DoG) was utilized. SURF has a step ahead by estimating Laplacian of Gaussian (LoG) with Box filters. Estimation demonstration is shown in Figure 4.4. The estimation greatest pros are that, with the help of vital pictures the box filter convolution will be simple ascertained, and parallel figuring should be possible for various scales. Additionally, SURF relies on Hessian matrix for both position and scale.

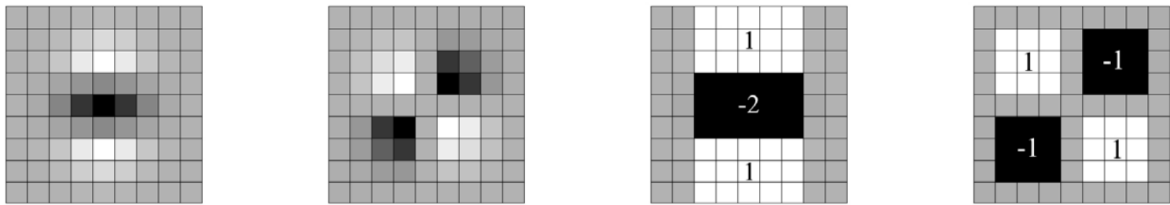


Figure 4.4: Box filter representations of Gaussian second order partial derivative.

Wavelet reactions are utilized as a part of vertical and horizontal dimensions for a neighborhood with a size of 6 increased through the scale in which the key point is recognized so as to get orientation task. later on, Proper Gaussian weights will be used. At that point, the estimation of the fundamental orientation is acquired through summing of all reactions inside a sliding orientation windows of 60 degrees. An important point is, essentially at any pivot invariance, there is no need to look for the orientation, as needs be, the speed of processing is increased. SURF likewise gives one more strategy termed U-SURF or Upright-SURF that is speedier and it is strong up to plus or minus 15 degrees. Figure presented below, (Figure 4.5) displays particular SURF key points of a traffic/road sign image.



Figure 4.5: SURF key points detected in a traffic sign image.

Feature portrayal is done by employing a $16s \times 16s$ neighborhood measure around each key point where S represents the size. Division is performed, furthermore there is formation of 4×4 sub regions as presented in Figure 4.6. For every one of the sub regions a vertical and horizontal wavelet reactions are picked and vector is created. Consequently, there are aggregate of 64 measurements of SURF features from the vector formed. This will prompt higher the matching and computation speed, and bring down the measurement, so signs of improvement of feature uniqueness is shown.

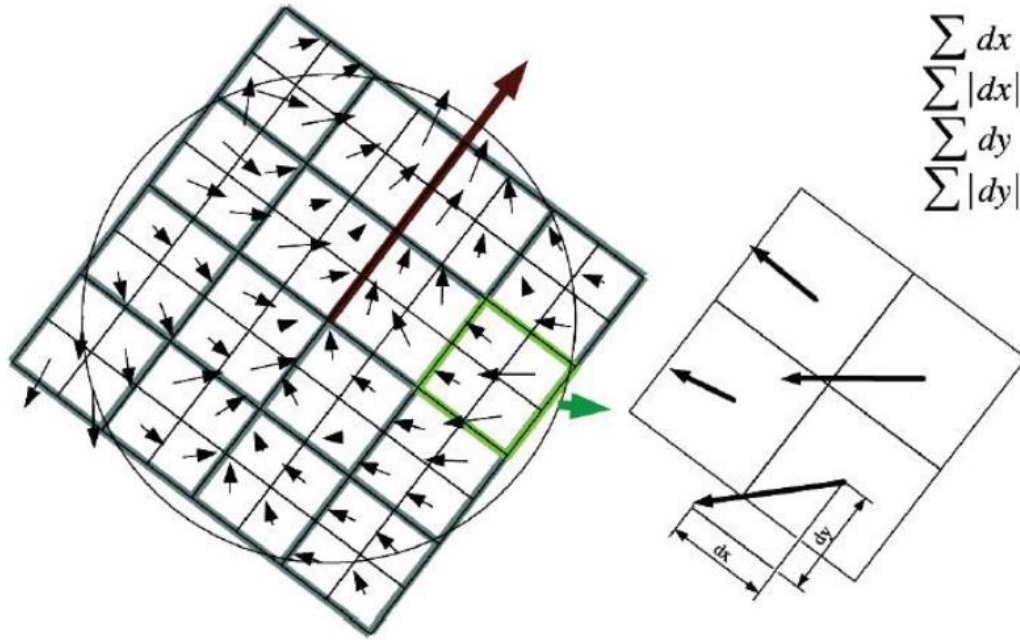


Figure 4.6: The demonstration of descriptor building (Bay et Al., 2006)

For unmistakable basic key points, another advancement that was done is utilizing sign of Laplacian (trace of Hessian Matrix). For this method, no extra calculations cost is required, in light of the fact that it has been done amid identification. In the revers circumstance, Laplacian sign separates the blobs made up of bright on dark background. Amid matching of the features, only those with the same background are compared as shown in Figure 4.7. thus, the matching process is speed-up without influencing the descriptor's execution adversely.

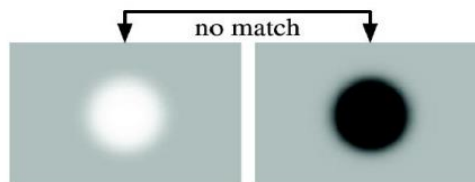


Figure 4.7: Matching of the fast index.

And so forth, in each progression, a great deal of highlights is combined with the Speeded-URF and this enhances the speed of the procedure in Speeded-URF. We need to specify that, in taking care of pictures with obscuring and rotation SURF is great, yet at taking care of brightening and perspective variety, it isn't so good.

Both in SIFT and SURF, the matching procedure betwixt the key points of two pictures are by distinguishing their nearest neighbors (k-NN, where “k” represents the value and NN represents nearest neighbors) which as represented in Figure 4.8. Be that as it may, at times, the first and second nearest coordinates are near each other, which results from noise or possibly different reasons. At these circumstances, the principal nearest distances to the following nearest distance is considered.



Figure 4.8: Example of point matching result.

4.4.2 Scale- Invariant feature transform

This algorithm developed by Prof. Dr. David G. Lowe in the year 2004 and it's called Scale Invariant Feature Transform (SIFT). Its main purpose is to extract even features from traffic

sign images, and then use this features to match and identify any traffic sign within a database of corresponding traffic sign images. factors such as images scale, rotations, changing illuminations and noise do not affect the features extracted from the image, simply put, the changes don't interrupt the features. The steps involved in SIFT algorithm for feature extraction and key point description are given in Figure 4.9.

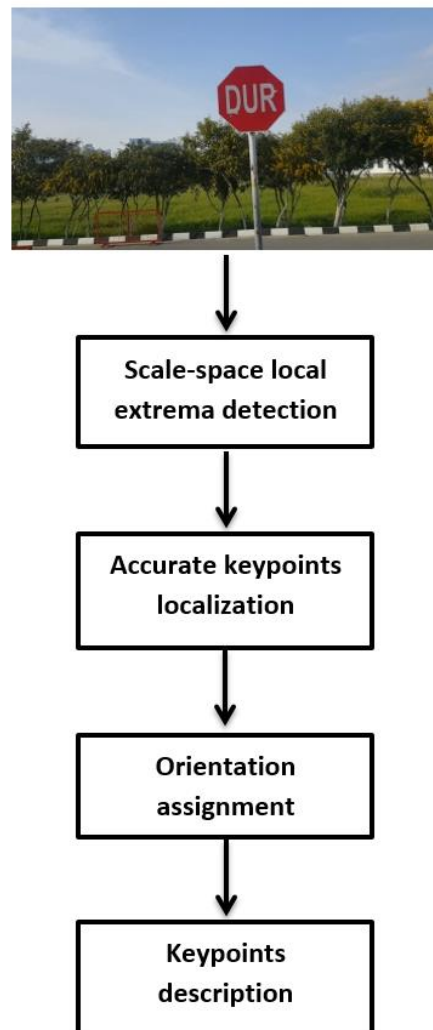


Figure 4.9: This extraction process done by SIFT.

4.4.3 Scale-space local extrema detection

In SIFT algorithm, key points can be acquired by identifying different image scale with varying window sizes. key points identification can be done by detecting the image's larger

corners with larger windows, the process is less demanding for distinguishing little corners. for this reason, scale-space kernels are utilized here. these kernels give diverse σ esteems. In this way, LoG is essentially a blob identifier which works as indicated by varying of σ on various sizes of the traffic sign image. As needs be, σ is the scaling parameter. Gaussian kernels yield an increasing value for little corners with low σ esteems, but perfectly adapts to bigger corners that have high σ esteems. A conclusion was reached that over the scale and space, a local maxima can be discovered, which gives us an arrangement of (x, y, σ) values that demonstrates the potentials of having key points at (x, y) at σ scale. The cost nature of LoG is the reason why it is not adopted in SIFT algorithm, instead Difference of Gaussians (DoG) was used. DoG refers to the DoG blurring of the traffic sign image with two different σ . The DoG algorithm is given below:

$$L(x, y, \sigma) = G(x, y, \sigma) \times I(x, y), \quad (4.3)$$

With the Gaussian kernel:

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2} \quad (4.4)$$

K separates the Gaussian difference, which results in the interpretation below:

$$\begin{aligned} D(x, y, \sigma) &= L(x, y, k\sigma) - L(x, y, \sigma) \\ &= (G(x, y, k\sigma) - G(x, y, \sigma)) \times I(x, y) \end{aligned} \quad (4.5)$$

Consequently, several octaves(scales) of the traffic sign images as it is observed in the Gaussian Pyramid, went through the process shown in Figure 4.9 above.

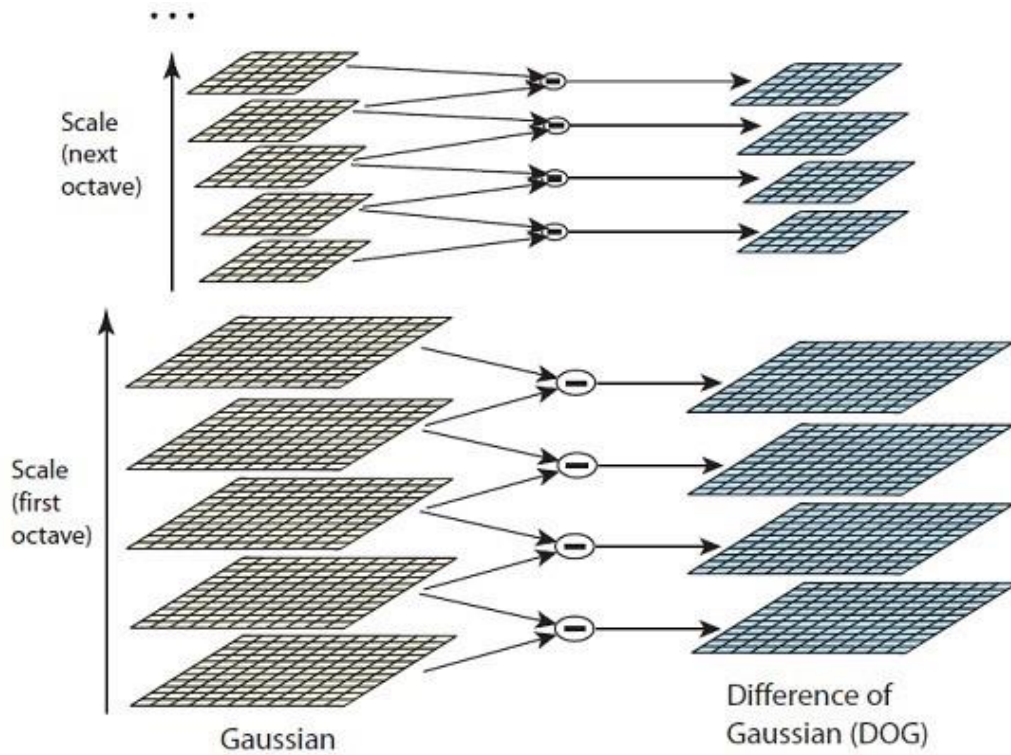


Figure 4.10: Gaussian pyramid (Lowe, 2004).

Availability of DoG gave the feasibility of observing traffic sign images for local maximum and minimum over scale and space. For instance, a comparison can be made between a simple input point on the traffic sign image and its neighbors which happens to be 26 pixels or point i.e. 8 neighbors, 9 pixels from past scale and 9 pixels in the following scale. Appropriately, local maximum or local minimum key points will be obtained. Local maximum pixel refers to a pixel whose value is greater than those of its neighbor, whereas local minimum is when the scenario is reversed. So the discovered key point is the most ideal in the scale as seen in Figure 4.11.

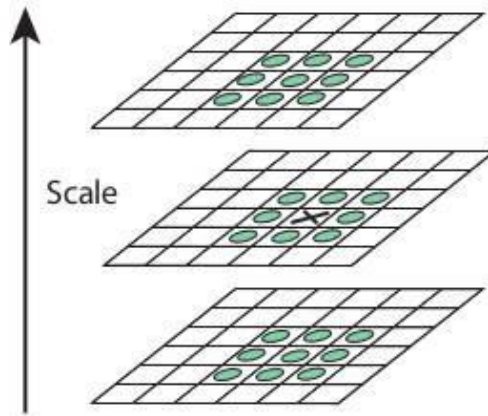


Figure 4.11: Scale space for keypoint (Lowe,2004).

4.4.4 Keypoint localization

After finding the conceivable key points, it needs some thorough observation in order to make it more realistic, refining is done to make a key point have precise outcomes. Taylor Series Approximation is utilized for key points refining procedure to locate the exact position of the key point in the scale space. Key points has fall under the Taylor Series limit level will be eliminated. Difference of Gaussians is utilized for the edge issue, in the same manner the edges should be eliminated, for this reason, a methodology same as Harris corner detector that discusses about a main curvature that processed employing a 2x2 Hessian Framework (H) is applied. As indicated by Harris corner locator one characteristic value for any edge is greater than the other characteristic value. So a notion is brought that, if a key point proportion was bigger than the decided edge esteem, that key point will be rejected. Along these lines hearty and solid key points are obtained, while low differing key points or edge key points are disposed.

4.4.5 Orientation appointment

With giving picture revolution invariance to each and every individual key point we get introduction arrangement. The scale is the foundation for deciding the slope size and bearing of the area. Thus, the 360-degree scope of orientation is secured by 36 bins of orientation histogram. The weight of this 36 bins is calculated using Gaussian-weighted circular windows and its angle size with σ equivalent to 1.5 times the size of key point. The

histogram's highest point and every other peaks that are more than 80% are a piece of the orientation. key points of varying directions are created however having a similar scale and position. A matching stability will be achieved.

4.4.6 Key point descriptor

Pixels of 16x16 scale are acquired around each key point neighborhood. It is separated into 16 sub-areas of 4x4 size. A histogram of 8 bin if created for individual sub-blocks. In general, a 128 bin are accomplished. Likewise, values are allocated as vectors in order to make key point descriptors as indicated is Figure 4.12.

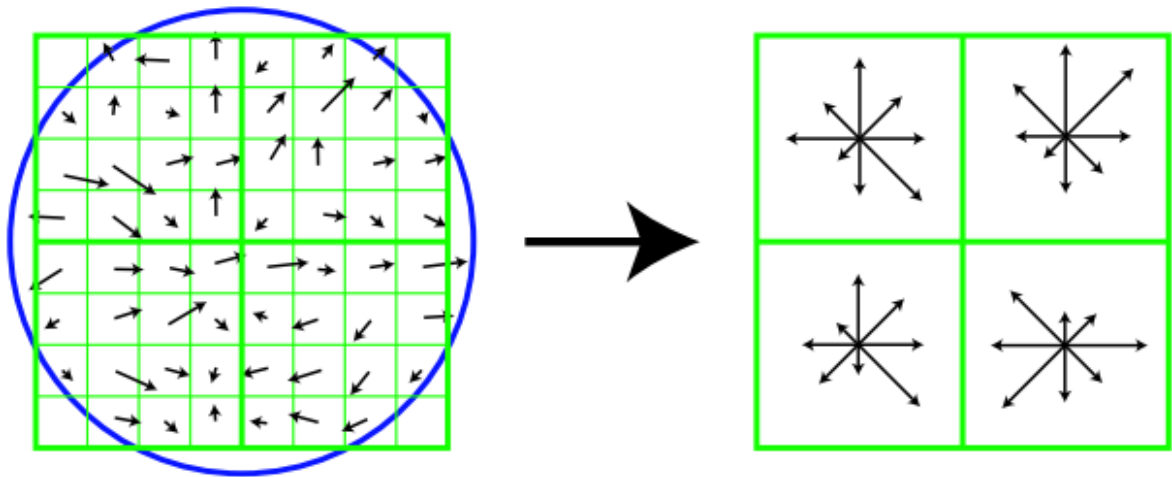


Figure 4.12 SIFT keypoint descriptor (Lowe,2004)

To pick up stability against light changes, pivot and so on a few methods are used.

Figure 4.13 shows unmistakable key points identified in traffic picture.

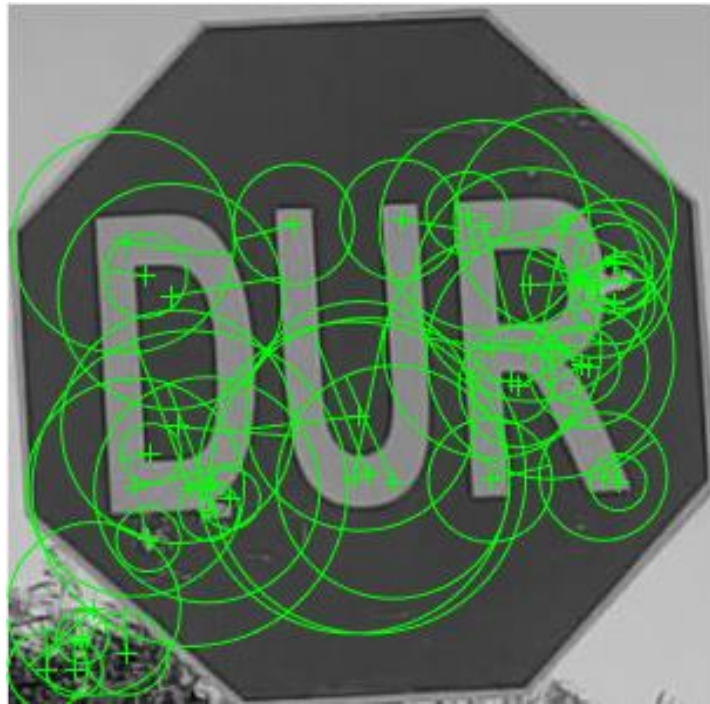


Figure 4.13: A traffic sign image including SIFT keypoints detected by the program.

CHAPTER 5

RESULTS AND DISCUSSIONS

5.1 Results

In this thesis, 6 different types of traffic signs were analysed. The traffic signs were automatically detected via the image processing methods. The automatic and manual detection were compared and the success rate was determined as shown in the Table 5.1.

Table 5.1: Comparison of automatically and manually detected signs values and their success rate.

Sign	Video name	Detected	Total	Success rate
Stop sign	Video 1	85	105	80.95
Roundabout sign	Video 2	128	142	90.14
Pedestrian sign	Video 3	80	82	97.56
Pedestrian sign	Video 4	75	85	88.23
Park Area	Video 5	73	98	74.48
Roundabout sign	Video 6	53	84	63.09
Pedestrian sign	Video 6	70	80	87.5
Left way sign	Video 7	87	133	65.41
Left way sign	Video 8	57	78	73.07
Park Area	Video 9	82	103	79.61
Stop sign	Video 10	97	127	76.37
			Ave. =	80.54

Table 5.1 shows the list of traffic signs that were detected using the software and their corresponding success rate. It shows that pedestrian sign from video 3 has the highest success rate of 97.56 percent followed by pedestrian sign from video 4 video. Round sign from Video 6 has the lowest success rate.

The % success rate between these detection was calculated with the formula given in Equation 5.1.

$$\% \text{ Success rate} = \left| \frac{\text{Automatically detected value}}{\text{Manually detected value}} \right| \times 100 \quad (5.1)$$



a) Video 4 frame 34.



b) Video 4 frame 53.



c) Video 4 frame 70.



d) Video 1 frame 88.

Figure 5.1: Example frames for video 4

Figure 5.1 shows some of the video frames used the feature extraction. The images were taken from the driver's view; they contain several traffic signs that were detected by image processing software.

Figure 5.2 shows light blue colour which represents Roundabout sign, yellow colour represents pedestrian sign and navy blue demonstrates that the system does not detect and match the traffic signs that are included in the database. In Figure 5.3, green colour indicates the stop sign. The values on the horizontal axis represent the video frame size

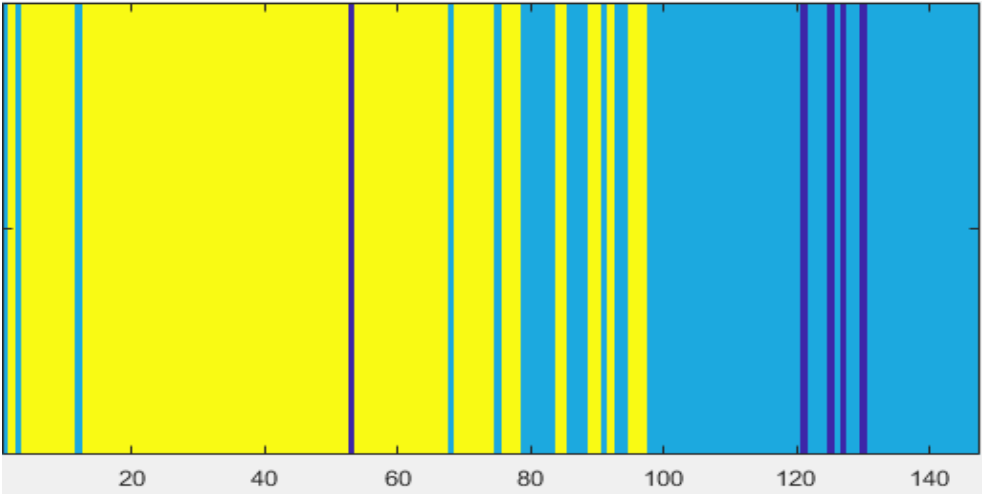


Figure 5.2: Video 4 timeline

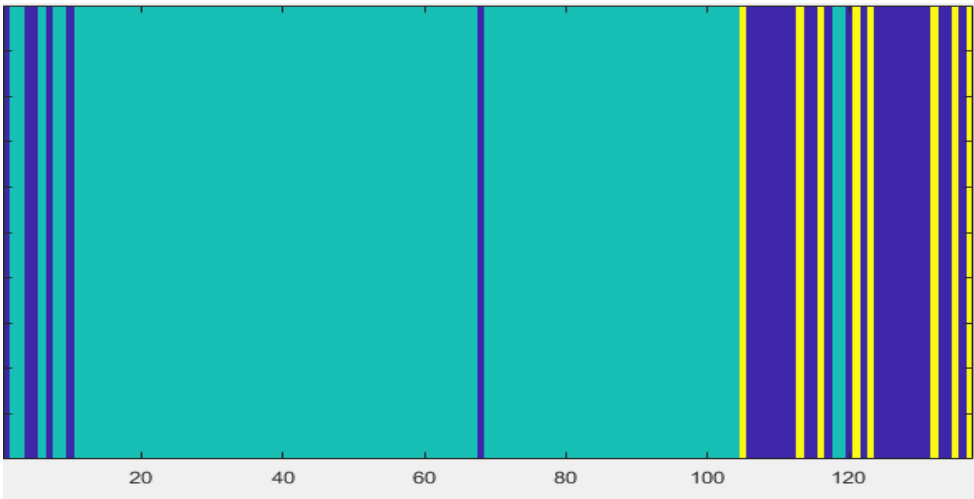


Figure 5.3: Video 1 timeline

5.2 Discussions

Several studies were conducted in the past with the aim of traffic sign detection using SIRF algorithms, fuzzy app approach etc. This study uses a different method to detect the traffic signs in a particular environment. This location was chosen because it has access ways used by drivers and pedestrians of which mostly are students. Traffic signs like parking area, zebra crossing, stop sign, roundabout sign etc. were detected in the video frame. The major findings of the study are high success rate of pedestrian sign detection whereas park way and left way signs showed low success rate. This result is good because the drivers using the roads will be aware of all the signs and they will perform the necessary action. Furthermore, actions such as stopping when they observe the STOP sign or Zebra crossing will help prevent minor and major accidents. The pedestrians will also be safe due to the active utilization of the software by the car owners.

There are no major limitations regarding this study because it presents us with accurate results based on our inputs. Nevertheless, limitations could occur with constant use of the software. The videos used as inputs were taken during the day time and under clear weather condition, none were taken during cloudy weather or at night therefore these conditions can be put into consideration in future in order to determine the overall performance of the software. We can compare our result to the available literatures because the outcome is quite good even though some studies have superior success rates to ours. This software can be classified as very good for traffic sign detection.

CHAPTER 6

CONCLUSION AND SUGGESTIONS

6.1 Conclusion

Using SURF method for traffic sign detection has proved to be very effective at detecting traffic signs in a particular environment. The result from this thesis are obtained from feature extraction using image processing in MATLAB. Videos containing several traffic signs were uploaded to the MATLAB program, which were then converted to video frames and used for the image processing. The software was successful at detecting the traffic signs in the video frames with the closest sign having the highest matching point. The software was tested in high ways and within a school environment, and it successfully identified the traffic signs within the driver's view. This software was tested in rigorous weather condition or night time, which prevent us from reporting the overall performance. Based on the output data available from the given inputs, the software is very good for its purpose. A table containing the list of the individual video frames with their corresponding traffic sign and success rate is given in the result section of this thesis.

6.2 Suggestions

For future related studies, augmented reality methods can be employed for similar processing. Also, mobile applications can be developed so that it can be available to several users. Furthermore, voice assistances can be used in the system to tell the drivers about the traffic signs on the road. More road signs can be added because we use only six sign in this study, real time system can be developed, the data sets can be increased, images taken at night or other weather conditions can be added.

REFERENCES

- Ding, D., Yoon, J., & Lee, C. (2012, November). Traffic sign detection and identification using SURF algorithm and GPGPU. In *SoC Design Conference (ISOCC), 2012 International* (pp. 506-508). IEEE.
- Fang, C. Y., Fuh, C. S., Chen, S. W., & Yen, P. S. (2003, June). A road sign recognition system based on dynamic visual model. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on* (Vol. 1, pp. I-I). IEEE.
- Fang, C. Y., Chen, S. W., & Fuh, C. S. (2003). Road-sign detection and tracking. *IEEE transactions on vehicular technology*, 52(5), 1329-1341.
- Fitriyah, H., Widasari, E. R., & Setyawan, G. E. (2017, November). Traffic sign recognition using edge detection and Eigen-face: Comparison between with and without color pre-classification based on Hue. In *Sustainable Information Engineering and Technology (SIET), 2017 International Conference on* (pp. 155-158). IEEE.
- Fu, K., Gu, I. Y., Ödöblom, A., & Liu, F. (2016, June). Geodesic distance transform-based salient region segmentation for automatic traffic sign recognition. In *Intelligent Vehicles Symposium (IV), 2016 IEEE* (pp. 948-953). IEEE.
- Gonzalez, R., & Richard, E. (2002). C., and Woods, R., E., 2002. Digital image processing.
- Han, Y., Virupakshappa, K., & Oruklu, E. (2015, May). Robust traffic sign recognition with feature extraction and k-NN classification methods. In *Electro/Information Technology (EIT), 2015 IEEE International Conference on* (pp. 484-488). IEEE.
- Hoose, N. (1991). Computer image processing in traffic engineering.
- Idris, F., & Panchanathan, S. (1997). Review of image and video indexing techniques. *Journal of visual communication and image representation*, 8(2), 146-166.

Jack, K., (2005), Video Demystified, Elsevier Inc.

Jiang, G. Y., & Choi, T. Y. (1998). Robust detection of landmarks in color image based on fuzzy set theory. In *Signal Processing Proceedings, 1998. ICSP'98. 1998 Fourth International Conference on* (Vol. 2, pp. 968-971). IEEE.

Jin, T., Xiong, L., Bin, X., Fangyan, C., & Bo, L. (2010, August). A method for traffic signs detection, tracking and recognition. In *Computer Science and Education (ICCSE), 2010 5th International Conference on* (pp. 189-194). IEEE.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2), 91-110.

Richardson, Iain E. G., (2003), H.264 and MPEG-4 Video Compression, John Wiley and Sons Ltd.

Studio encoding parameters of digital television for standard 4:3 and wide-screen 16:9 aspect ratios 2011, Grayscale Retrieved April 10, 2018 from https://www.itu.int/dms_pubrec/itu-r/rec/bt/R-REC-BT.601-7-201103-I!!PDF-E.pdf

Swathi, M., & Suresh, K. V. (2017, May). Automatic traffic sign detection and recognition in video sequences. In *Recent Trends in Electronics, Information & Communication Technology (RTEICT), 2017 2nd IEEE International Conference on* (pp. 476-480). IEEE.

Traffic Signs Manual Introduction 1982, Traffic signs Retrieved March 30, 2018 from https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/203662/traffic-signs-manual-chapter-01.pdf

Umbaugh, S. E. (2005). *Computer imaging: digital image analysis and processing*. CRC press.

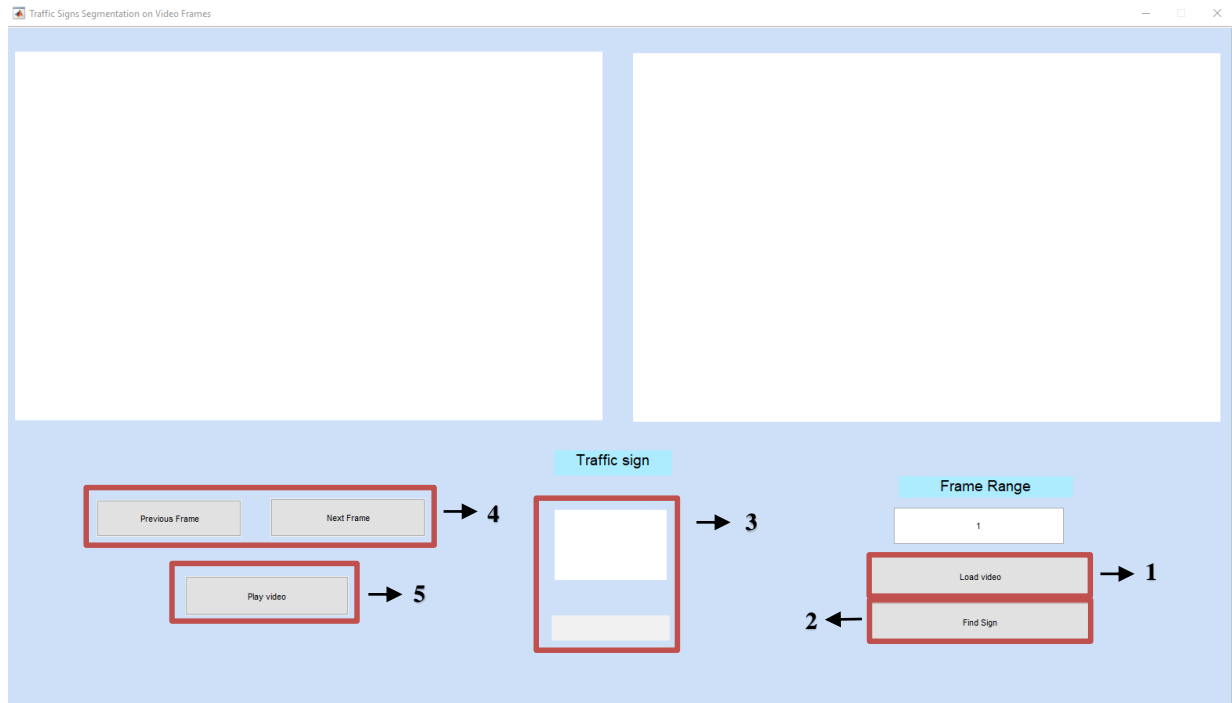
United Kingdom Government, Traffic signs Retrieved March 30, 2018 from <https://www.gov.uk/guidance/the-highway-code/traffic-signs>

Zhao, J., Zhu, S., & Huang, X. (2013, September). Real-time traffic sign detection using SURF features on FPGA. In *High Performance Extreme Computing Conference (HPEC), 2013 IEEE* (pp. 1-6). IEEE.

APPENDICES

APPENDIX 1

PROGRAM INTERFACE



1. Get video and convert to frames and saves those images.
2. This part start to process to detect and identification traffic signs.
3. Section 3 show if any traffic sign match in the system show in axis and demonstrate the detected traffic sign.
4. Allow to move between frames and show above the axis.
5. Collect the processed frames and combine those frames converts to video.

APPENDIX 2

PROGRAM LISTING

Convert video to frames part function:

```
handles = guidata(hObject);
[filename, pathname] = ...
    uigetfile({'video/*. *'}, 'Select Image File');
T=strcat(pathname, filename);
vid=VideoReader(T);
numFrames = vid.NumberOfFrames;
handles.n=numFrames;

vn= get(handles.edit1, 'String');
handles.v=str2num(vn);
for i = 1:handles.v:handles.n
    frames =read(vid,i);
    frames= imresize(frames, 0.01);
    imwrite(frames, ['image/' int2str(i), '.jpg']);
```

Main Function:

```
vn= get(handles.edit1, 'String');
handles.v=str2num(vn);
set(handles.text4, 'String', '');

StopImage = imread('ornek\durl.jpg');
StopImagex=StopImage;
StopImage=rgb2gray(StopImage);

CemberImage = imread('ornek\ceber.jpg');
CemberImagex=CemberImage;
CemberImage=rgb2gray(CemberImage);

parkImage= imread('ornek\parkarea3.png');
parkImagex=parkImage;
parkImage=rgb2gray(parkImage);

leftImage= imread('ornek\leftway.png');
leftImagex=leftImage;
leftImage=rgb2gray(leftImage);

yayaImage= imread('ornek\yaya2.png');
yayaImagex=yayaImage;
yayaImage=rgb2gray(yayaImage);

stopPoints = detectSURFFeatures(StopImage);
[stopFeatures, stopPoints] = extractFeatures(StopImage, stopPoints);

CemberPoints = detectSURFFeatures(CemberImage);
[cemberFeatures, CemberPoints] = extractFeatures(CemberImage,
CemberPoints);
```

```

parkPoints = detectSURFFeatures(parkImage);
[parkFeatures, parkPoints] = extractFeatures(parkImage, parkPoints);

leftPoints = detectSURFFeatures(leftImage);
[leftFeatures, leftPoints] = extractFeatures(leftImage, leftPoints);

yayaPoints = detectSURFFeatures(yayaImage);
[yayaFeatures, yayaPoints] = extractFeatures(yayaImage, yayaPoints);

for xa = 1:handles.v:handles.n
    sceneImage= imread(sprintf('image/%d.jpg',xa));

    t=sceneImage;
    sceneImage=rgb2gray(sceneImage);

    scenePoints = detectSURFFeatures(sceneImage);

    [sceneFeatures, scenePoints] = extractFeatures(sceneImage, scenePoints);

    StopPairs = matchFeatures(stopFeatures, sceneFeatures);
    cemberPairs = matchFeatures(cemberFeatures, sceneFeatures);
    parkPairs = matchFeatures(parkFeatures, sceneFeatures);
    leftPairs = matchFeatures(leftFeatures, sceneFeatures);
    yayaPairs = matchFeatures(yayaFeatures, sceneFeatures);

    matchedStopPoints = stopPoints(StopPairs(:, 1));
    matchedScenePoints = scenePoints(StopPairs(:, 2));

    matchedCemberPoints2 = CemberPoints(cemberPairs(:, 1));
    matchedScenePoints2 = scenePoints(cemberPairs(:, 2),:);

    matchednoparkPoints3 = parkPoints(parkPairs(:, 1));
    matchedScenePoints3 = scenePoints(parkPairs(:, 2),:);

    matchedleftPoints4 = leftPoints(leftPairs(:, 1));
    matchedScenePoints4 = scenePoints(leftPairs(:, 2),:);

    matchedyayaPoints5 = yayaPoints(yayaPairs(:, 1));
    matchedScenePoints5 = scenePoints(yayaPairs(:, 2),:);

    numstopPairs = length(StopPairs); %the number of pairs
    numCemberPairs = length(cemberPairs); %the number of pairs
    numparkPairs = length(parkPairs); %the number of pairs
    numleftPairs = length(leftPairs); %the number of pairs
    numyayaPairs = length(yayaPairs); %the number of pairs

    son(1,1)=numstopPairs;
    son(2,1)=numCemberPairs;
    son(3,1)=numparkPairs;
    son(4,1)=numleftPairs;
    son(5,1)=numyayaPairs;
    satir = find(son == (max(son)));

```

```

satir =satir (1,1);
switch satir

    case 1

        [tform, inlierStopPoints, inlierScenePoints,status] = ...
            estimateGeometricTransform(matchedStopPoints,
matchedScenePoints, 'affine');
        if (status >2)
            disp('0');
            listson(xa,1)=0;
            myString = sprintf('No match');
            set(handles.text6, 'String', myString);
            axes(handles.axes2);
            imshow(t);
            wh= imread('white.jpg');
            axes(handles.axes3);
            imshow(wh);

        else
            disp('1');
            listson(xa,1)=1;
            stop = [1, 1;...
                size(StopImage, 2), 1;...
                size(StopImage, 2), size(StopImage, 1);...
                1, size(StopImage, 1);...
                1, 1];
            newstop = transformPointsForward(tform, stop);
            myString = sprintf('Stop sign');
            set(handles.text6, 'String', myString);
            axes(handles.axes2);
            imshow(t);
            line(newstop(:, 1), newstop(:, 2), 'Color', 'g');
            axes(handles.axes3);
            imshow(StopImagex);

        end
    case 2

        [tform, inlierStopPoints, inlierScenePoints,status] = ...
            estimateGeometricTransform(matchedCemberPoints2,
matchedScenePoints2, 'affine');
        if (status >2)
            disp('0');
            listson(xa,1)=0;
            myString = sprintf('No match');
            set(handles.text6, 'String', myString);
            axes(handles.axes2);
            imshow(t);
            wh= imread('white.jpg');
            axes(handles.axes3);
            imshow(wh);

        else
            disp('2');
            listson(xa,1)=2;

```

```

        cember = [1, 1;...
                  size(CemberImage, 2), 1;...
                  size(CemberImage, 2), size(CemberImage, 1);...
                  1, size(CemberImage, 1);...
                  1, 1];

        newcember = transformPointsForward(tform, cember);
        myString = sprintf('Roundabout sign');
        set(handles.text6, 'String', myString);
        axes(handles.axes2);
        imshow(t);
        line(newcember(:, 1), newcember(:, 2), 'Color', 'b');
        axes(handles.axes3);
        imshow(CemberImagex);

    end
    case 3
        [tform, inlierStopPoints, inlierScenePoints,status] = ...
            estimateGeometricTransform(matchednoparkPoints3,
matchedScenePoints3, 'affine');
        if (status >2)
            disp('0');
            listson(xa,1)=0;
            myString = sprintf('No match');
            set(handles.text6, 'String', myString);
            axes(handles.axes2);
            imshow(t);
            wh= imread('white.jpg');
            axes(handles.axes3);
            imshow(wh);

        else
            disp('3');
            listson(xa,1)=3;
            park = [1, 1;...
                    size(parkImage, 2), 1;...
                    size(parkImage, 2), size(parkImage, 1);...
                    1, size(parkImage, 1);...
                    1, 1];

            newpark = transformPointsForward(tform, park);
            myString = sprintf('Park area sign');
            set(handles.text6, 'String', myString);
            axes(handles.axes2);
            imshow(t);
            line(newpark(:, 1), newpark(:, 2), 'Color', 'r');
            axes(handles.axes3);
            imshow(parkImagex);

        end
    case 4

        [tform, inlierStopPoints, inlierScenePoints,status] = ...
            estimateGeometricTransform(matchedleftPoints4,
matchedScenePoints4, 'affine');

```



```

if (status >2)
    disp('0');
listson(xa,1)=0;
myString = sprintf('No match');
set(handles.text6, 'String', myString);
axes(handles.axes2);
imshow(t);
wh= imread('white.jpg');
axes(handles.axes3);
imshow(wh);

else
    disp('4');
    listson(xa,1)=4;
left = [1, 1;...
        size(leftImage, 2), 1;...
        size(leftImage, 2), size(leftImage, 1);...
        1, size(leftImage, 1);...
        1, 1];

    newleft = transformPointsForward(tform, left);
myString = sprintf('Leftway sign');
set(handles.text6, 'String', myString);
axes(handles.axes2);
imshow(t);
line(newleft(:, 1), newleft(:, 2), 'Color', 'r');
axes(handles.axes3);
imshow(leftImage);

end

case 5

    [tform, inlierStopPoints, inlierScenePoints,status] = ...
        estimateGeometricTransform(matchedyayaPoints5,
matchedScenePoints5, 'affine');
    if (status >2)
        listson(xa,1)=0;
        disp('0');
        myString = sprintf('No match');
        set(handles.text6, 'String', myString);
        axes(handles.axes2);
        imshow(t);
        wh= imread('white.jpg');
        axes(handles.axes3);
        imshow(wh);
    else
        listson(xa,1)=5;
        disp('5');
        yaya = [1, 1;...
                size(yayaImage, 2), 1;...
                size(yayaImage, 2), size(yayaImage, 1);...
                1, size(yayaImage, 1);...
                1, 1];
        newyaya = transformPointsForward(tform, yaya);
        myString = sprintf('Pedestrian sign');
        set(handles.text6, 'String', myString);
    end
end

```

```

        axes(handles.axes2);
        imshow(t);
        line(newyaya(:, 1), newyaya(:, 2), 'Color', 'b');
        axes(handles.axes3);
        imshow(yayaImage);

    end

    F = getframe(handles.axes2);
    Image = frame2im(F);
    imwrite(Image, sprintf('pic/%d.png', xa));
end

listson = ( listson );
save('listson.mat', 'listson');

```

Frames converts to video function:

```

video = VideoWriter('video.avi');
open(video);
for xx = 1:handles.v:handles.n
    I = imread(sprintf('pic/%d.png', xx));
    writeVideo(video, I);
end
close(video);
set(handles.text4, 'String', 'done');

guidata(hObject, handles);

```

Play video function:

```

xyloObj = VideoReader('video.avi');

nFrames = xyloObj.NumberOfFrames;
vidHeight = xyloObj.Height;
vidWidth = xyloObj.Width;

mov(1:nFrames) = ...
    struct('cdata', zeros(vidHeight, vidWidth, 1, 'uint8'), ...
        'colormap', []);
for k = 1 : nFrames
    mov(k).cdata = read(xyloObj, k);
end
h = figure ('units', 'normalized', 'outerposition', [0 0 0.50 0.56]);
hf = axes('Parent' , h);

movie(hf, mov, 1, xyloObj.FrameRate);

```