

VEHICLE NUMBER PLATE DETECTION (IRAQI PLATES)

**A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCE
MATHEMATICS AND COMPUTER SCIENCE**

ÇANKAYA UNIVERSITY

BY

OTHMAN SUBHI SIDDIK

**IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE
DEGREE OF**

MASTER OF SCIENCE

IN

THE DEPARTMENT OF

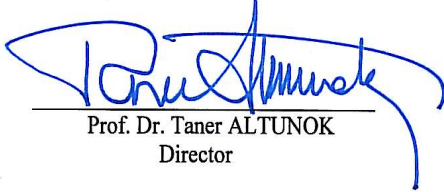
MATHEMATICS AND COMPUTER SCIENCE

JANUARY 2014


Title of Thesis: **Vehicle Number Plate Detection (Iraqi Plates)**

Submitted By **Othman SIDDIK**


Approval of the Graduate School of Natural And Applied Science, Çankaya University.


Prof. Dr. Taner ALTUNOK
Director

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


Prof. Dr. Bülür KAYMAKÇALAN
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis of degree Master of Science (M.Sc.) in Mathematics and Computer Science.


Asst. Prof. Dr. Abdül Kadir GÖRÜR
Supervisor

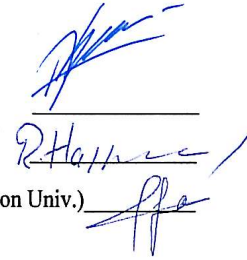
Examination Date: 14.01.2014

Examining Committee Members

Asst. Prof. Dr Abdül Kadir GÖRÜR (Çankaya Univ.)

Asst. Prof. Dr. Reza HASSANPOUR (Çankaya Univ.)

Assoc.Prof. Dr. Fahd JARAD (Turkish Aeronautical Association Univ.)



STATEMENT OF NON PLAGIARISM PAGE

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 : Othman SIDDIK

Signature



Date

: 14.01.2014

ABSTRACT

VEHICLE NUMBER PLATE DETECTION (IRAQI PLATES)

SIDDIK, Othman Subhi

M.Sc., Department of Mathematics and Computer Science

Supervisor: Assist. Prof. Dr. Abdül Kadir GÖRÜR

January 2014, 64 pages

This thesis presents a system for Iraqi vehicle number plate detection. It is one of the most interesting and challenging research topics in the past few years. The system is designed to perform detection for Iraqi vehicle number plates under any environmental conditions, it is shown that the number plates have different size and shape and also have different colors. Vehicle number plate detection is mainly used to detect an object for traffic management. There is a need for intelligent system. Vehicle number plate detection is widely used for security control, detecting speeding cars, electronic toll collection and traffic law enforcement. In Iraq the most common vehicle number plates uses white color as background and black color as character color, also there are different colors like yellow color which is used as background and black color as for character color for the trucks. In this thesis we propose a system for the detection of number plates mainly for the vehicles in Iraq. This thesis presents an approach based on simple and efficient horizontal and vertical histogram operation and Robert edge detection method. After reducing noise from the input image, we try to enhance the contrast of the binarized image. In this thesis, examples of correct and incorrect results, as well as possible, practical applications

of proposed method are presented, and the system was implemented on 50 car plate, and the system has successfully detected all car plates.

Keywords: Extraction Plate Region, Binrizing, Edge Detection, Robert Edge Detector.

ÖZ

ARAÇ PLAKASI TESPİTİ (IRAK PLAKALARI)

SIDDIK, Othman Subhi

Yüksek Lisans, Matematik - Bilgisayar Anabilim Dalı
Tez Yöneticisi: Yrd. Doç. Dr. Abdül Kadir GÖRÜR

Ocak 2014, 64 sayfa

Bu tez Irak'taki araç plakalarının tespiti için bir sistem sunmaktadır. Bu, son yılların en ilginç ve en zor araştırma konularından biridir. Sistem her türlü hava ve yol koşulunda Irak'taki araç plakalarını tespit etmek üzere tasarlanmıştır ve Irak'taki araç plakaları değişik renkte, tipte ve boyuttadır. Araç plakası tespit sistemi en çok trafik yönetiminde kullanılmakta ve akıllı bir sisteme ihtiyaç duyulmaktadır. Araç plakası tespit sistemi genelde güvenlik kontrolleri, hız yapan araçların tespiti, yol ücretlerinin tahsili ve trafik kanunlarının uygulanmasında kullanılmaktadır. Irak'ta görülen en yaygın araç plakası tipi beyaz zemin üzerine siyah yazılı plakalardır, bununla birlikte kamyonlarda sarı zemin üzerine siyah yazılı plakalar bulunmaktadır. Bu tezde, esasen Irak'taki araçların plakalarının tespiti için bir sistem önermekteyiz. Bu tez, Robert ayrıt saptama metodu ve basit ve etkili yatay ve dikey histogram operasyonu yaklaşımına dayanmaktadır. Girdi görüntüsünden parazit azaltıldıktan sonra ikilileştirilmiş görüntünün kontrastını iyileştirmeye çalıştık. Bu tezde doğru ve yanlış sonuçların örnekleriyle beraber önerilen metodun uygulamaları da

bulunmaktadır ve sistem 50 araç plakasına uygulanmıştır ve sistem tüm araç plakalarını başarılı bir şekilde tespit etmiştir.

Anahtar Kelimeler: Bölge Plakası Çıkarımı, İkileştirme, Ayrıt Saptama, Robert Ayrıt Saptayıcı.

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to **Assist. Prof. Dr. Abdül Kadir GÖRÜR** for his supervision, special guidance, suggestions, and encouragement through the development.

It is a pleasure to express my special thanks to my family for their valuable support.

TABLE OF CONTENTS

STATEMENT OF NONPLAGIARISM.....	III
ABSTRACT.....	IV
ÖZ.....	VI
ACKNOWLEDGEMENTS.....	VIII
TABLE OF CONTENTS.....	IX
LIST OF FIGURES.....	XI
LIST OF ABBREVIATIONS.....	XV

CHAPTERS:

1. INTRODUCTION.....	1
1.1. Background	1
1.2. Related Work	3
2. FUNDAMENTALS OF IMAGE PROCESSING	4
2.1. Image Processing Basics.....	4
2.2. Image Processing Toolbox.....	4
2.3. Image Formats Supported by MATLAB	5
2.4. Working Formats in MATLAB.....	5
2.5. MATLAB Image Types.....	5
2.5.1. RGB Format	5
2.5.2. Intensity Image (Gray Scale Image) GRAY Format.....	7
2.5.3. Binary Image.....	8
2.5.4. Indexed Image.....	9
2.6. Image Enhancement	10
2.7. Illustration Preserving in the Illustrating Chain.....	10
2.8. Contrast Enhancements.....	11
2.9. Gray-Level Histogram	11

2.10.1.	Histogram Equalization	13
2.10.2.	Comparing Two Histograms	14
2.11.	Edge Detection	15
2.11.1.	Gradient Based Edge Detection.....	16
2.11.2.	Laplacian Based Edge Detection.....	16
2.11.3.	Sobel Operator.....	16
2.11.4.	Robert's Cross Operator.....	17
2.11.5.	Prewitt's Operator	17
2.11.6.	Laplacian of Gaussian	18
2.12.	Visual Comparison of Various Edge Detection Algorithms.....	18
3.	DESIGN AND APPLICATION OF PREPOSED WORK	21
3.1.	Overview.....	21
3.2.	Converting the RGB Image Into Gray Image	21
3.3.	Binrizing	22
3.4.	Edge Detection	23
3.5.	Edge Processing of Horizontal and Vertical Histogram	24
3.6.	Investigation Output Image	26
4.	EXPERIMENTAL AND RESULTS.....	32
4.1.	Implementation.....	32
4.2.	Results and Discussions	39
5.	CONCLUSION AND FUTURE WORKS.....	63
5.1.	Conclusion	63
5.2.	Future Work.....	64
	REFERENCES.....	R1
	APPENDICES.....	A1
A.	CURRICULUM VITAE.....	A1
B.	PROPOSED WORK IN MATLAB.....	B1

LIST OF FIGURES

FIGURES

Figure 1	Flow Diagram for the Algorithm Steps.....	2
Figure 2	RGB Format	6
Figure 3	Representation of Pixels in RGB Format	6
Figure 4	RGB Image.....	7
Figure 5	Gray Scale Image.....	8
Figure 6	Binary Image.....	9
Figure 7	Indexed Image.....	9
Figure 8	Camera's Potential Image Quality for Modeling.....	11
Figure 9	High Contrast and Low Contrast of Gray-Level Histogram.....	12
Figure 10	Example of Histogram Equalization	13
Figure 11	Masks Used by Sobel Operator	16
Figure 12	Masks Used for Robert Operator	17
Figure 13	Masks for The Prewitt Gradient Edge Detector	17
Figure 14	Discrete Approximations to the Laplacian Filter	18
Figure 15	Image used for Edge Detection analysis (Wheel.Gif).....	19
Figure 16	Results of Edge Detection Roberts had the best results.....	19
Figure 17	Comparison of Edge Detection Techniques Original Image.....	20
Figure 18	Comparison of Edge Detection Techniques on Lena Image	20
Figure 19	Comparison of Edge Detection Technique on Noisy Image	20
Figure 20	Original Colored Image (Car.Jpg).....	22
Figure 21	Gray Scale Image	22
Figure 22	Binary Threshold Image	23
Figure 23	Roberts Edge Detection	24
Figure 24	Row and Column Content	24
Figure 25	Horizontal Histogram	25
Figure 26	Vertical Histogram.....	26
Figure 27	Output Image.....	26
Figure 28	Car Plate Test 1	27

FIGURES

Figure 29	Car Plate Test 2	27
Figure 30	Car Plate Test 3	28
Figure 31	Car Plate Test 4	28
Figure 32	Contrast Check Diagram	29
Figure 33	Output the Car Plate.....	29
Figure 34	Histogram of the Car Plate	30
Figure 35	Car Parts with High Histogram	30
Figure 36	Histogram of Car Parts	31
Figure 37	Iraqi Car Plate Detection	32
Figure 38	Iraqi Plates Parts of High Horizontal and Vertical Histograms.....	33
Figure 39	Result of High Horizontal and Vertical Histogram.....	34
Figure 40	High Histogram Output Image	34
Figure 41	Image with High Parts of Histogram Converted to Black.....	35
Figure 42	Final Output of the Program.....	35
Figure 43	Histogram of the Car Plate Final Output	36
Figure 44	Input Image with Yellow Car Plate.....	37
Figure 45	European Car Plates.....	38
Figure 46	Netherland Car Plates	39
Figure 47	Iraqi Plate Test 1	40
Figure 48	Iraqi Plate Test 2	40
Figure 49	Iraqi Plate Test 3	41
Figure 50	Iraqi Plate Test 4	41
Figure 51	Iraqi Plate Test 5	42
Figure 52	Iraqi Plate Test 6	42
Figure 53	Iraqi Plate Test 7	43
Figure 54	Iraqi Plate Test 8	43
Figure 55	Iraqi Plate Test 9	44
Figure 56	Iraqi Plate Test 10	44
Figure 57	Iraqi Plate Test 11	45

FIGURES

Figure 58	Iraqi Plate Test 12	45
Figure 59	Iraqi Plate Test 13	46
Figure 60	Iraqi Plate Test 14	46
Figure 61	Iraqi Plate Test 15	47
Figure 62	Iraqi Plate Test 16	47
Figure 63	Iraqi Plate Test 17	48
Figure 64	Iraqi Plate Test 18	48
Figure 65	Iraqi Plate Test 19	49
Figure 66	Iraqi Plate Test 20	49
Figure 67	Iraqi Plate Test 21	50
Figure 68	Iraqi Plate Test 22	50
Figure 69	Iraqi Plate Test 23	51
Figure 70	Europe Plate Test 1	51
Figure 71	Europe Plate Test 2	52
Figure 72	Europe Plate Test 3	52
Figure 73	Europe Plate Test 4	53
Figure 74	Europe Plate Test 5	53
Figure 75	Europe Plate Test 6	54
Figure 76	Europe Plate Test 7	54
Figure 77	Europe Plate Test 8	55
Figure 78	Europe Plate Test 9	55
Figure 79	Europe Plate Test 10	56
Figure 80	Europe Plate Test 11	56
Figure 81	Europe Plate Test 12	57
Figure 82	Europe Plate Test 13	57
Figure 83	Europe Plate Test 14	58
Figure 84	Europe Plate Test 15	58
Figure 85	Europe Plate Test 16	59
Figure 86	Europe Plate Test 17	59

FIGURES

Figure 87	Europe Plate Test 18	60
Figure 88	Europe Plate Test 19	60
Figure 89	Input Image with the Environment Effects.....	62
Figure 90	Input Image without Environment Effects	62

LIST OF ABBREVIATIONS

MATLAB	Matrix Laboratory (Software)
OCR	Optical Character Recognition
RGB	Red-Green-Blue
M-files	MATLAB Files
GUI	Graphical User Interface
XWB	Extra Wide Body
HDF	Hierarchical Data Format
BMP	Bone Morphogenetic Protein
TIFF	Tag Image File Format
PCX	Planoconvex (Optische Linsen/Optical Lenses)
JPEG	Joint Photographic Expert Group
2D	Two-Dimensional
3D	Three-Dimensional
LPD	Lisance Plate Detection
LOG	Laplac of Gaussian
G _x	Gradient Mask of X
G _y	Gradient Mask of Y
T	Transformation
r	Gray Values
F	Input Image
S	New Pixels with Gray Levels
G _x	Gradient Mask One
G _y	Gradient Mask Two
H	Histogram
d_{correl}	Distance Metric of Correlation
$d_{chi-square}$	Distance Metric of Chi-Square
$d_{cintersection}$	Distance Metric of Cintersection
$d_{Bhattacharyya}$	Distance Metric of Bhattacharyya

CHAPTER 1

INTRODUCTION

1.1. Background

Currently in the worldwide, there are more than half a billion vehicles [1]. As a primary identifier, all those vehicles have their identification numbers. The vehicle number plate detection in fact is a vehicle number that means a legal license to participate in the public traffic. It should be fixed onto its body. In fact, the manual methods for dealing with more than half a billion vehicles are much difficult. Therefore, an automatic system called vehicle plate detection is strongly needed (including detection of plate from the vehicle image). In this thesis, we present automatic algorithm system for detecting Iraqi vehicle plates. In our thesis, we are going to detect the car plates as shown below.

Basically, the system of car plate detection consists of four main stages:

- 1) Binrizing.
- 2) Edge processing.
- 3) Extracting using horizontal and vertical projection profiles.
- 4) Checking contrasts.

Regarding the procedure used here, first, we convert image into gray scale image to make the image easier to read off several plates. Second, dilation is to take place is a clearer image reached by filling holes. In this regard, the edges are sharpened and the noise is removed and the differences between dark and light are evident. In addition, we work with horizontal and vertical process by checking each column and row of the image is which traversed by the algorithm. The difference in pixel value between adjacent pixels is calculated if the difference in gray scale value between the successive pixels exceeds a certain limit. Finally, contrast of the output image if checked whether it is a car plate histogram contrast or not by checking the difference in whiteness and blackness.

Here is the chart diagram of full implementation discussed within MATLAB code; the block diagram is shown in figure1.

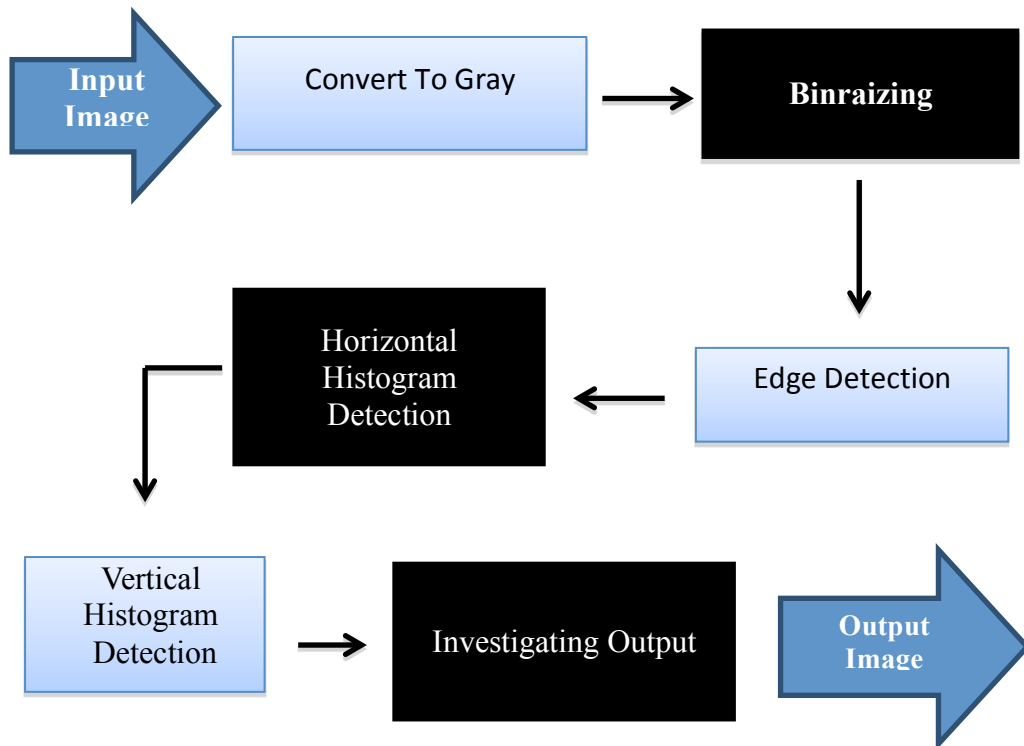


Figure 1 Blog Diagram for the Algorithm Steps.

In the following sections description of each block given in figure 1.

Convert Image to Gray: First get the image converted to a gray scale image. Thus, it is easy to read off various types of plates.

Binrizing: This is the clearer image, got by means of filling the holes, sharpening the edges and removing the noise. Differences between dark and light are also to be evident.

Horizontal and Vertical Histogram Processing: Each of column and row of the image are traversed by the algorithm, the difference in pixel value between adjacent pixels is calculated. If the difference in the gray scale value between the successive pixels exceeds a certain limit, then the count on the total for that particular column.

Here, the Edge processing is used. The region with the highest value of vertical and horizontal histogram will contain the number plate.

Investigation Output Image: Here, the histogram processing is used, and the region with the highest value of white pixels histogram will contain the car plate.

1.2. Related Work

Many algorithms have been proposed to implement Vehicle Number Plate detection, the algorithms can be classified as: filtering, edge based, color base and texture based. The Edge detection and color information method to extract car plate and location are proposed by and Runping Han and Hulli Hsan [2], for Indian car plate recognition system there is a novel approach proposed by Dr.A.Jhansi Rani, Ch. Jaya Lakshmi, Dr K.Sri Ramakrishna, M. kanti Kiran and V.R Siddharthea and V.R Siddhartha [3]. Morphology and edge detection algorithms for license plate location described in [4] [5] [6] [7]. For Iraqi car plate recognition the algorithm is proposed by Abdulhussein Mohsin¹, Abbas H. Hassin^{1,2}, Iman Qais Abdul Jaleel¹[8]. The most use algorithm for car plate identification a system for car plate recognition is proposed by Tulu Muluneh, Dening Jiang, Ashenafi Gebrehiwot, Tiruneh Embiale [9]. To detect a car plate first it have to delete the noise and the noise may it is from the environment effects so there is method is used to recognize and extract the car plate by passing under outside environment the algorithm proposed by Seyed Hamidreza Mohades Kasaei and Sayed mohammadreza Mohades Kasaei proposed [10]. Honglei Zhu and Zhigang Xu they have proposed an efficient method for detecting the car plate [11]. There are several methods to segmentation a car plate [12]. For more efficient recognition we need to more efficient detection and many number plate localization algorithms have been developed in this thesis an efficient and improved approach is identified with high detection rate based on Roberts's edge detection the system detect 25 Iraqi car plate successfully and 25 car plate for other countries.

CHAPTER 2

FUNDAMENTALS OF IMAGE PROCESSING

2.1. Image Processing Basics

Image is used to deliver useful information in a visible format, and it is the order of small elements in a two-dimensional plane. Also the pixels are consisting of these small elements. The main element that is used to describe an image is a pixel. Mostly, each pixel of an image is represented in RGB (Red Green Blue) format. In the case of an RGB image, the three components called R, G and B are collected together to order information about the brightness and color of a single pixel. Each component consumes certain memory space during image processing [13]. An image form is consisting of a large number of pixels collected together, whether they are small or large. The relevant information of image, like color, luminance and light intensity, is represented by each pixel. A large number of these pixels are collected together to form an image.

2.2. Image Processing Toolbox

The Image Processing Toolboxes, a set of functions, provide numeric computing environment of MATLAB capability. The toolbox backing large number operations on image processing contains:

- I-Transforms
- II-Geometric operations
- III-Neighborhood and block operations
- IV-Binary image operations
- V-Filter design and linear filtering
- VI-Image analysis and enhancement
- VII-Region of interest operations

MATLAB M-files are the main toolbox functions. The code performs experts in image processing algorithms. Depending on the declaration type function name, we

view the MATLAB code functions by writing our own M-files We provide the Toolbox of the Image Processing capabilities, or by depending on toolbox collections with other toolboxes, like Wavelet Toolbox and the Signal Processing Toolbox [14].

2.3. Image Formats Supported by MATLAB

The formats of image corroborated by MATLAB are:

I-XWB

II-HDF

III-BMP

IV-TIFF

V-PCX

VI-JPEG

Most used compression standards for images are called JPEG-images. Therefore, by suffixing we can see what format we have stored when we store an image. For instance, when we store a JPEG format, the image is called myimage.jpg [15].

2.4. Working Formats in MATLAB

In order to start working with an image we firstly read it into MATLAB. Then we store the image on our disc as Jpeg image [15].

2.5. MATLAB Image Types

Intensity format images: [0, 1] or uint8

Indexed format images: m-by-3 color map

RGB format images: m-by-n-by-3

Binary format images: [0, 1].

2.5.1. RGB Format

Human beings use colors to recognize objects, time, food, quality, their material, texture etc. Colored image processing needs an understanding of the light as well as perception of the color in the image [16]. Color images take much more space as compared to that taken by their identical gray scale or binary images. Moreover,

through image processing techniques and activities, the gray scale images are usually used due to many other factors that make them simpler and better in terms of calculation and usage. In the RGB color type, there are three basal colors; they are Red, Green, and Blue. From here basal colors can derive further secondary colors like:

GREEN+ BLUE =CYAN

GREEN+RED=YELLOW

BLUE+RED=MAGENTA

These secondary colors are shown in figure2 [16].

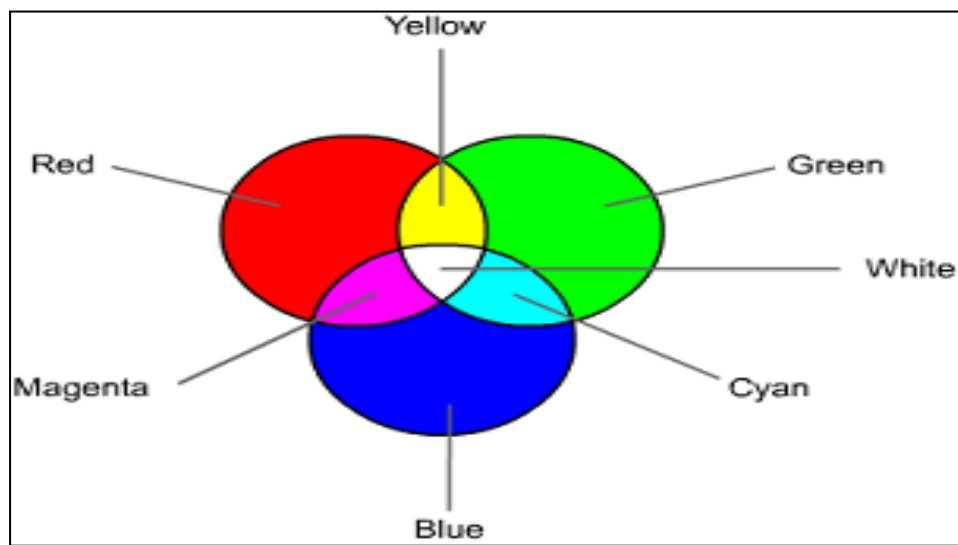


Figure 2 RGB Format.

Color mixtures subtract magenta; yellow and cyan for fabricate RGB image red, blue, and green. Each pixel is consisting of three different components R, G and B. Each of these components requires at least 8 bits for their storage. In general, a single pixel may require up to $8 * 3$ bits for its storage. An example of a representation of single pixel in RGB format is shown in figure 3. [17].

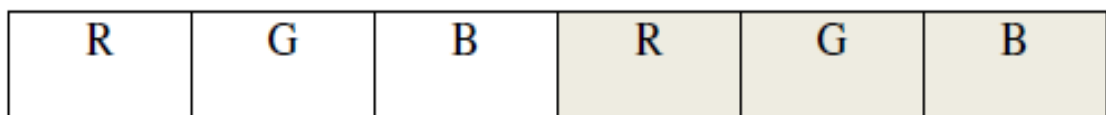


Figure 3 Representation of Pixels In RGB Format [2].

The value of R, G and B, is that each ranges from 0-255. A value of (255, 0, 0) is represented. The pixels of red and (0, 255, 0) represents the pixels of green and (0, 0, 0) represents the black pixel. Also to store a value for a single component 8 bits are required. All levels of red, green, and blow are stored like matrix as shown in figure4. [18].

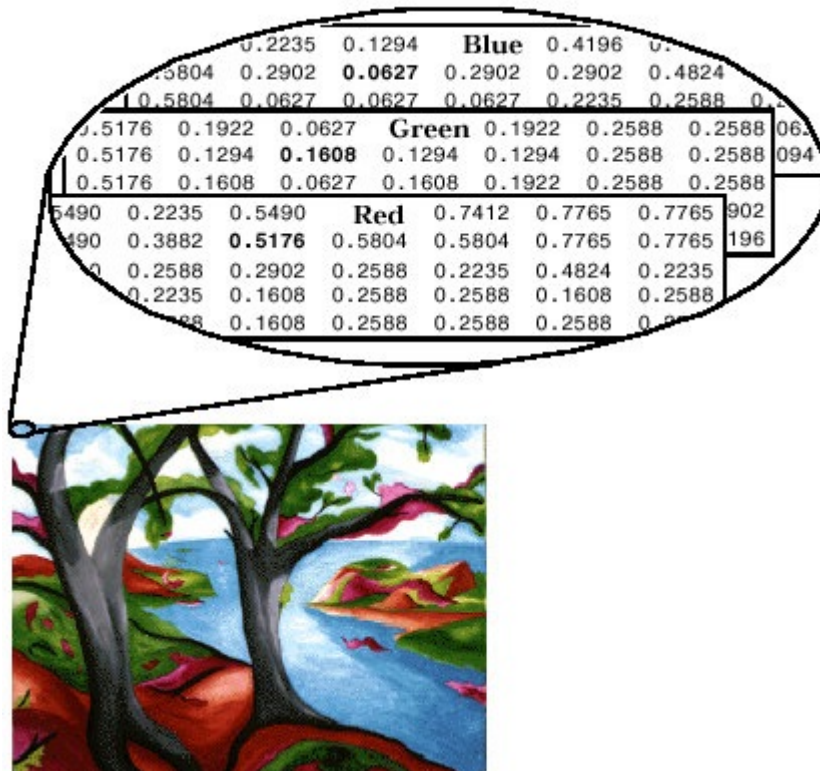


Figure 4 RGB Image.

2.5.2. Intensity Image (Gray Scale Image) GRAY Format

Matrices where every element of the matrix determines the density of the corresponding pixel represent intensity image tantamount to a gray scale image. For appropriateness, every element contains a value identical for the bright/dark and colors every pixel at the matching position. The maximum of the current digital files uses the integer number between 0 (the color of minimal intensity to indicate black) and 255 (maximum intensity to indicate white). Numbers that show the brightness of the pixel are represented in two ways: The data type or double class. In addition, the floating number specifies decimal numbers amidst 0 and 1 to every pixel. The value 1 identifies white and the value 0 identifies black and the other class is named unit 8

which specifies an integer amidst the 0 and 255 to be clarified as the brightness of a pixel. The value 255 identifies white and the value 0 identifies black as shown in figure 5. Therefore, to store the class uint8 only values almost 1/8 form the store compared to double class. Finally, the double class can be available of many mathematical functions [13].

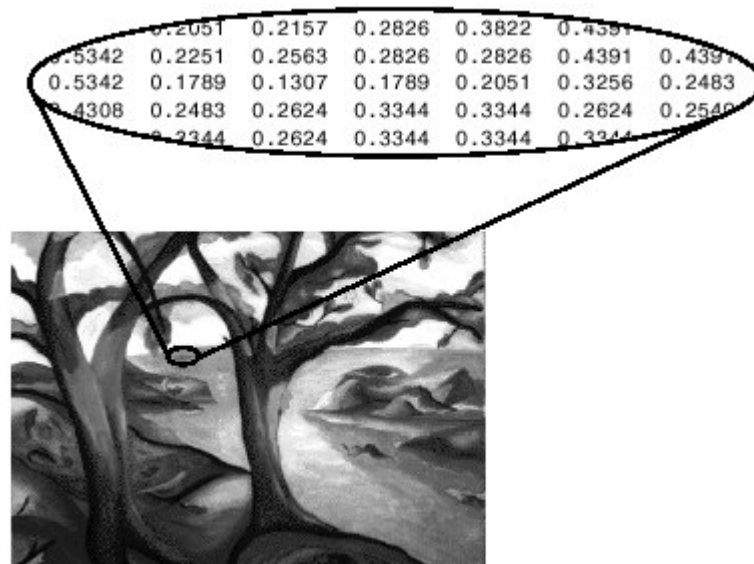


Figure 5 Gray Scale Image.

2.5.3. Binary image

To store binary image format, it is stored like matrix. However we can colore binary image pixels as black or white. So it specifies 1 for the white and 0 for the black. There are only two probable values for each pixel (0, 1), binary image requires only one bit per pixel as shown in figure 6. [19].

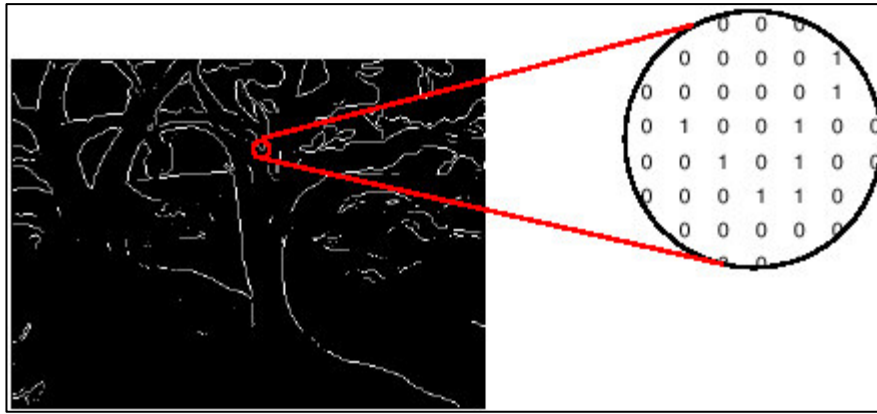


Figure 6 Binary Image.

2.5.4. Indexed image

It is a workable way for performing the color images and the first matrix is to store an indexed images. It has the quasi size as the one number for each pixel. Then, the color map is named of the second matrix and it is may be different in size from the image, The directive of what number to use in the second matrix, the color map matrix, is from the numbers in the first matrix as shown in figure 7. [15].

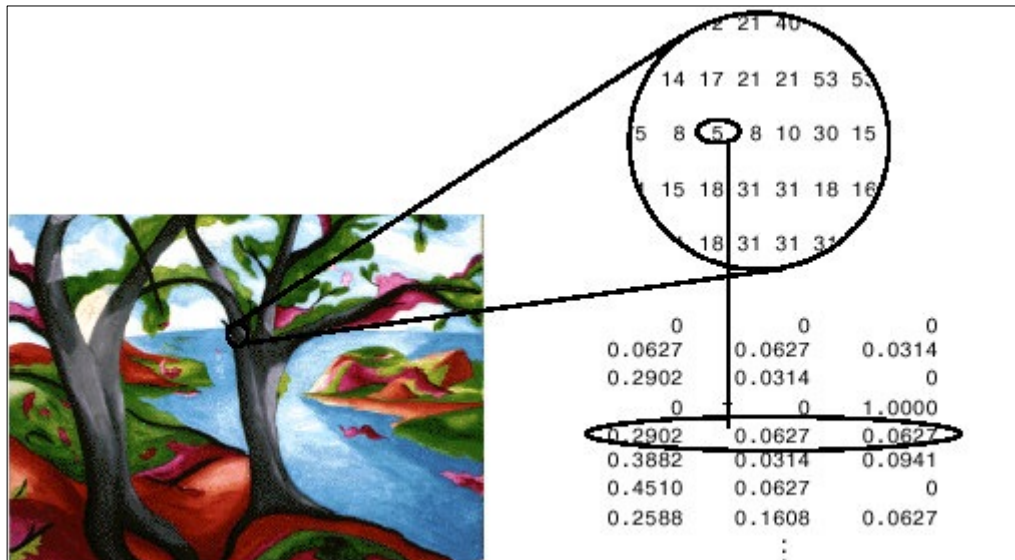


Figure 7 Indexed Image.

2.6. Image Enhancement

Combination of techniques plays a role in the formation of the image enhancement processes that claim to ameliorate the visual occurrence of an image or to convert the image to a better form convenient for analysis by a machine or a human. Also in regulation of an image enhancement, improving the fidelity of an image transcribed has no sentient stress with regard to several quixotic form of the image, and the same is done in recon quest image. Completely, there is some directory reference that oftentimes a deformed image, like an image of object edges with space undershoot and overshoot, is more individually joyfully than a completely original copy. For the analysis of image and the analysis of objects, information extraction stops short of the definition of image enhancement. For example, in the system of an image hatchment, the high-frequency filtering confirms the outline edges of objects in an image this image edge enhanced would then benefit as a machine input that would affect the edges outlines, and perhaps make measurements of the outline form and size. Also in this implementation, the original image confirms prominent features in the image enhancement processors and unwrap the processing duty of a data machine extraction. At present, in image enhancement, there is no theory of general unifying because the quality of the image, there is no general standard that can avail, like image enhancement processor layout standard. Assortment of techniques is a consideration that is given here to confirm usefulness for image analysis and supervision refinement of human [20].

2.7. Illustration Preserving in the Illustrating Chain

The electronic sensor product is a “new” electronic illustration that contains a collection of electronic count values. Each one of these values shows the gray level, or brightness, in the illustration of pixel. Generally, the illustration preserving is applied on the illustrating chain to develop the image data effectiveness Figure 8. In this aspect, the illustration preserving is a much-extended area that contains condensation, specialty determining and grouping. It is emphasized in the discussion that the general preserving ways are useful to improve the visual quality of the illustration. Particularly, it will be looked at contrast enhancement ways, in addition to, spatial filtering ways that sharpen edges and clear away considerable amount of

the illustration blur. (Generally the detector calibration is the first part of the chain illustration improvement; before it has been stated that it is part of the modeling sensor). Simply, it is thought that the dynamic range of an image has eight bits; for example, possible gray levels are $2^8=256$, and the range of gray levels in the image will be 0-255, with 255 being whit and zero being black. Three collections of numbers that the colored image has, are commonly displaying the red, blue and green illustrations and get the full spectrums of colors by bring them together [20].

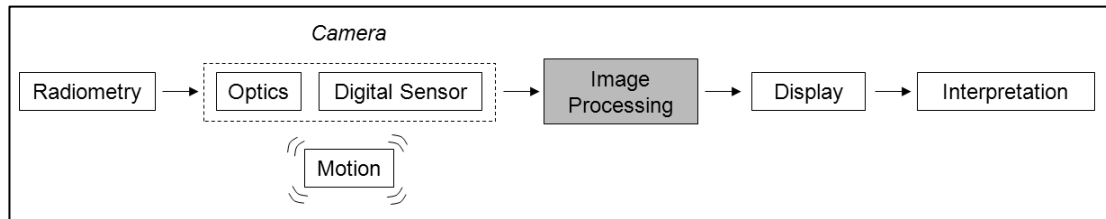


Figure 8 Camera's Potential Image Quality for Modeling.

2.8. Contrast Enhancements

Contrast enhancement develops the visibility of the items in the scene between items and their backgrounds by enhancing the brightness variety. They are commonly functioned the contrast enhancements as a contrast expansion are followed by a soft embellishing. Even though, these enhancements can both function in one action. A contrast expansion develops the brightness varieties regularly with the active range of the illustration, while soft enhancing develops the brightness varieties in the use of brightness varieties in the other parts of highlight parts in the shadow [21].

2.9. Gray-Level Histogram

The gray-level histogram benefits the most contrast embellishing ways Gray-level histogram is almost created by contrast enhancement methods. Gray-level is made by histogram enumerations. The times of every gray value happening in the image are counted by dividing every gray level in the image by the aggregate number of pixels in the image to produce an allocation of the percentage of every gray level in the image Figure 9. Declaration of the gray-level histogram without spatial information about the image is the statistical allocation of the gray levels in the image. Therefore, it optimizes the contrast of the full dynamic domain by regulating the exposition of the camera to span. This makes the risk of saturated mode of the detector with any

irradiation value that would override 255 counts. Thus, it leads to pruning these values into 255 counts and missing out any view of information above this irradiation level. Also, exposures are usually set to gather lower contrast images that do not span the dynamic range in order that the images can be processed later and preserving control over the amount of clipping that occurs to enhance the contrast [21].

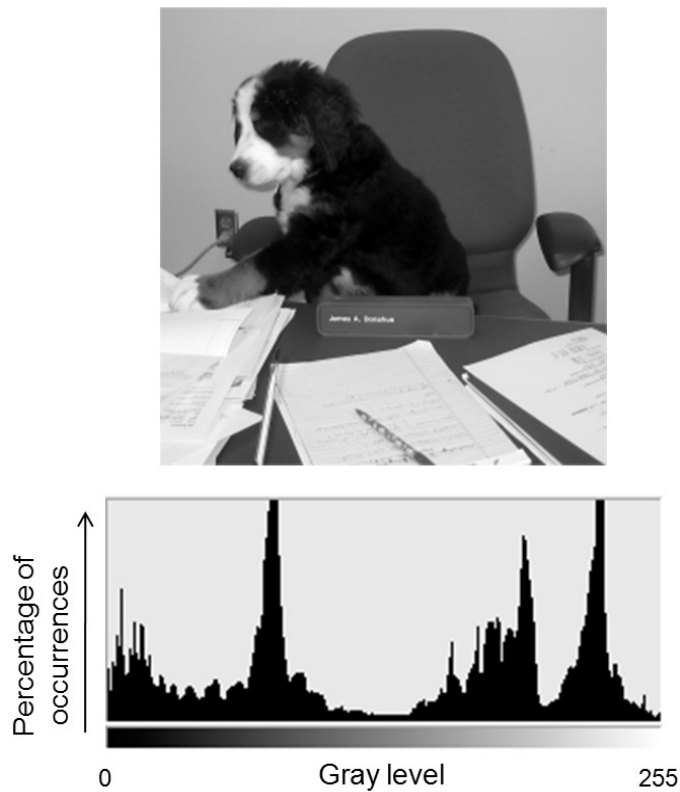


Figure 9 Iraqi Plates Parts of High Horizontal and Vertical Histograms.

Contrast enhancement process is to improve vision by regulating the proportional darkness of objects and brightness of objects. A gray-level transforming the tone and contrast of the image can be changed while mapping the gray levels in the image to new values.

2.10.1. Histogram Equalization

A concerted technique for image enhancement is histogram equalization. The function of the histogram transformation makes a uniform histogram for the output image. T is transformation. It maps r which is grey values into F which is input image to s grey values, $s = T(r)$ for transformed image. Transformation T is monotonically growing with single value; therefore,

$$0 \leq T(r) \leq 1 \text{ for } 0 \leq r \leq 1 \dots(2.1)$$

The transformation remapping function should be the accumulative allocation function. A Gaussian distribution as an example of histogram equalization is shown in Figure 10. [6].

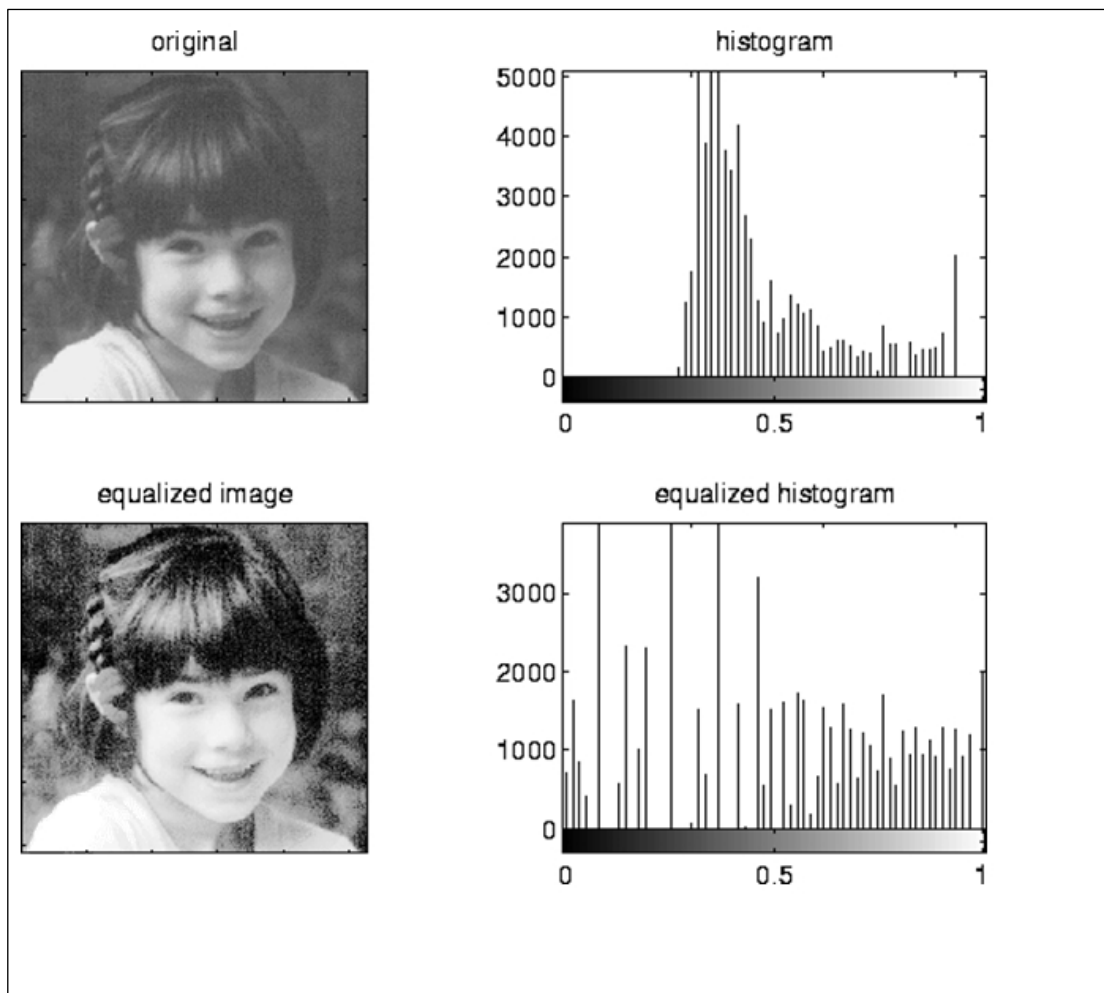


Figure 10 Example of Histogram Equalization.

2.10.2. Comparing Two Histograms

The compare two histograms capability in terms of resemblance specified criteria. (First presented by Swain and Ballard and further propagate by Schiele and Crowley [22]) distance metric:

Correlation

$$d_{correl}(H_1, H_2) = \frac{\sum_i H_1(i).H_2(i)}{\sqrt{\sum_i h_1^2(i).H_2^2(i)}} \dots(2.2)$$

Where

$$H_k(i) = H_k(i) - (1/n)(\sum_j H_k(j)) \dots(2.3)$$

And N assign to the number of bins in the histogram. For engagement, a high score appears a better match than a low score. A good match is 1 and an upper mismatch is -1; a value of 0 indicates no engagement (random association) [22].

Chi-square

$$d_{chi-square}(H_1, H_2) = \sum_i \frac{(H_1(i)-H_2(i))^2}{H_1(i)+H_2(i)} \dots(2.4)$$

For chi-square, a low score appears a better match than a high score. A perfect match is 0 and a total unconformity is unlimited (reckon on the size of the histogram) [22].

Intersection

$$d_{intersection}(H_1, H_2) = \sum_i \min (H_1(i), H_2(i)) \dots(2.5)$$

For histogram crossing, high scores point perfect matches and low scores point invalid matches. So if both of histograms are normalized to 1, then a ideal match is 1 and a total mismatch is 0 [22].

Bhattacharyya Distance

$$d_{Bhattacharyya}(H_1, H_2) = \sqrt{1 - \sum_i \frac{\sqrt{H_1(i).H_2(i)}}{\sqrt{\sum_i H_1(i).\sum_i H_2(i)}}} \dots(2.6)$$

For Bhattacharyya matching, low results points perfect matches and high results point's worst matches. A perfect match is 0 and a total mismatch is a 1[22].

2.11. Edge Detection

Edge detection indicates the determination sharp abrupt and the process of identifying an image. It describes the intensity of the pixel and it describes the limits of objects in a view abrupts are abrupt alterations in pixel intensity. The convolve image with the factor (a 2-D filter) are the traditional methods of edge detection. They are used for the big gradients in the image. They are structured to be sensitive for it, whereas values of zero are reverting in regular regions. Also, there are considerably a lot of numbers of edge detection operators; every one of it is intended to become sensitive to specific edge types and noise environment, edge structure and edge direction. These variables are implicated in the option operators for the edge detection. Characteristic direction is determined by the geometry of the operator. It is prompt in edges with most sensitivity. These factors are optimized to look for diagonal edges. They also look for vertical edges and horizontal edges. The edge detection is severe at the noisy images, since both of the noise and the edges include high frequency content. It endeavors to minimize the noise result in distorted edges and blurred edges. In domain, noisy images have exemplary larger numbers of operators. Therefore, we can get total of sufficient data to adversary localized noisy pixels. The result has detected edges with less precise localization. And the stride alteration in intensity is not covered by all edges. Result in objects with boundaries through poor focus or effects as refraction and the boundaries are represented by a piecemeal change in intensity [23]. Necessarily, the operator must be selected to be interactive to such a piecemeal change in those cases. Therefore, a lot of problems like false edge detection, problems of missing true edges, edge localization, high computational time and other problems cause of noise etc. In addition, the objective is to do the arbitrage of different edge detection techniques and analyze the effectiveness of the various techniques in various conditions [24]. Also, there are many ways to implement edge detection. Anywise, various methods majority may be gathered into two categories:

2.11.1. Gradient Based Edge Detection

It is directional change in image and intensity color. The method of gradient based edge detection is detection of the edges by researching the first derivative of the image for the maximum and minimum edges [25].

2.11.2. Laplacian Based Edge Detection

The Laplacian method is to find edges by checking for zero crossings in the second derivative of the image, so an edge has the one-dimensional form of a ramp and counting the derivative of the image can highlight of its location [26].

2.11.3. Sobel Operator

The operator contains a pair of 3×3 detour kernels as displayed in Figure 11. one kernel is merely the other rotated by 90° [26].

-1	0	+1		+1	+2	+1
-2	0	+2		0	0	0
-1	0	+1		-1	-2	-1
G _x				G _y		

Figure 11 Masks Used by Sobel Operator.

These kernels are contagious to react maximally to edges operation horizontally and vertically proportional to the pixel network, one kernel for each of the two perpendicular orientations. The kernels Able to add separately to the input image, to harvest independent measurements of the tendency component in every orientation (named G_x and G_y). These can combine simultaneously to find the unlimited size of the direction at every point and the orientation of that direction [27].

2.11.4. Robert's Cross Operator

The Roberts Cross operator implements a quick to compute a modest and 2-D spatial tendency measurement on an image. Pixel values at every point in the output shows the estimated absolute size of the spatial tendency of the input image at that point [5]. The operator contains a pair of 2×2 detour kernels as displayed in Figure 12. one kernel is merely the other rotated by 90° . This is very identical to the Sobel operator

+1	0	0	+1
0	-1	-1	0
G_x		G_y	

Figure 12 Masks Used for Robert Operator .

These kernels are contagious to comply maximally with edges running at 45° to the pixel net, one kernel for each of the two vertical directions. So in the input image the kernels can be utilized separately [28]. To produce independent measurements of the tendency ingredient in each orientation (recall these G_x and G_y). These can then be collective jointly to find the absolute size of the tendency at every point and the direction of that tendency [29].

2.11.5. Prewitt's Operator

Prewitt operator is comparable to the Sobel operator and is applied for detecting horizontal edges and vertical edges in images as shown in Figure 13. [30].

-1	0	+1	+1	+1	+1
-1	0	+1	0	0	0
-1	0	+1	-1	-1	-1
G_x			G_y		

Figure 13 Masks for the Prewitt Gradient Edge Detector.

2.11.6. Laplacian of Gaussian

The Laplacian of Gaussian is a 2-D isotropic gauge of the 2nd spatial derivative of an image. The Laplacian of an image highlights areas of fast density change and is subsequently often used for edge detection [31]. The Laplacian is often utilized to an image that has first been polished with object sacrificial a Gaussian polished filter so as to minimize its sensitivity to noise [32]. The operator ordinary picks a single gray level image as input and makes another gray level image accordingly output.

So we have to be obliged to discover a discrete detour kernel purpose of the input image that is explains as a set of discrete pixels. That can convergent the second derivatives in the introduction of the Laplacian [33]. Three usually used small kernels are shown in figure 14.

1	1	1	-1	2	-1
1	-8	1	2	-4	2
1	1	1	-1	2	-1

Figure 14 Discrete Approximations to the Laplacian Filter.

2.12. Visual Comparison of Various Edge Detection Algorithms

Here in figure 15. We have an image of wheel in gray scale to check all four kinds of edge detection performed as shown in figure 16. From here we can see the results. For detect Iraq car plate with best results we use Roberts's edge detection. This was foreseeable with regard to Roberts's edge detection calculation for regions in an image Roberts works well with binary images. And we check all operators for other three images first for comparison of Edge Detection Techniques in figure 17. And for comparison of Edge Detection Techniques on Lena Image Original Image are shown in figure 18. Finally we are showing the comparison of Edge Detection technique on Noisy Image in figure 19. [34].

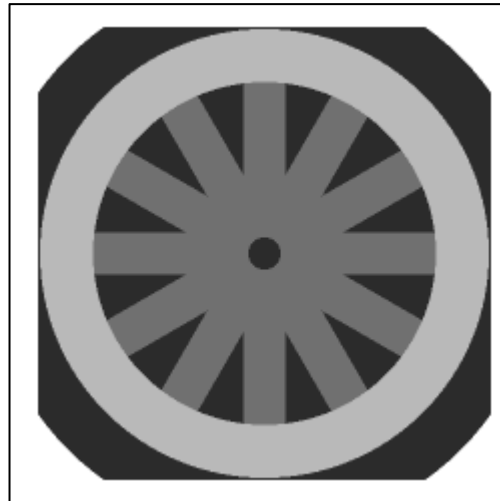


Figure 15 Image Used for Edge Detection Analysis (Wheel.Gif).

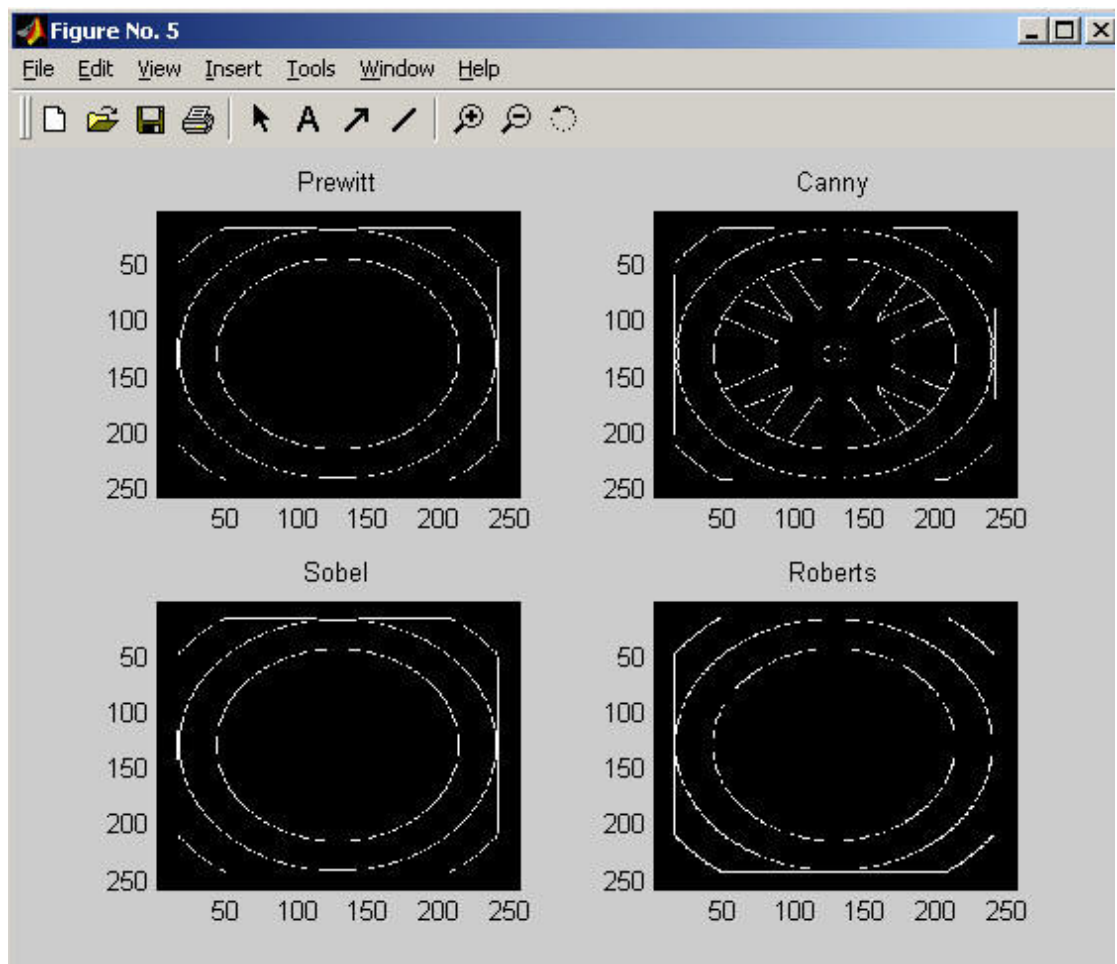


Figure 16 Results of Edge Detection Roberts Had the Best Results.

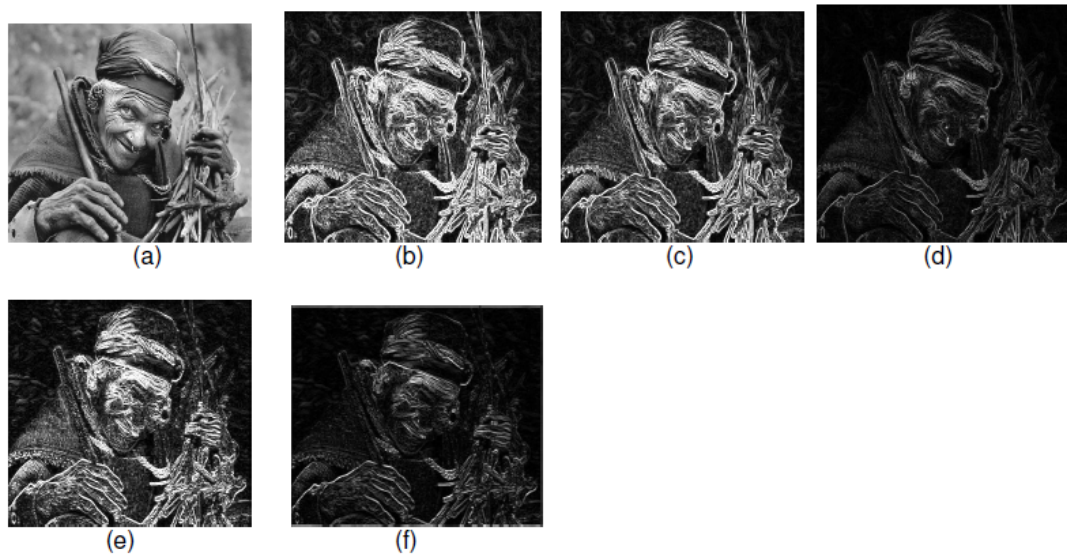


Figure 17 Comparison of Edge Detection Techniques (a) Original Image (b) Sobel (c) Prewitt (d) Robert (e) Laplacian (f) Laplacian of Gaussian.

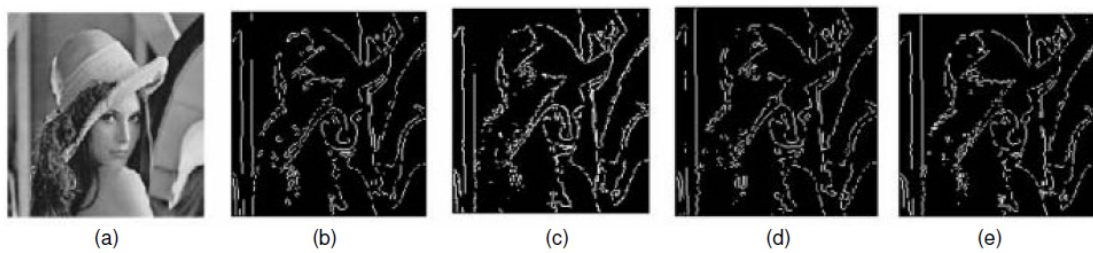


Figure 18 Comparison of Edge Detection Techniques on (a) Lena Image (b) Canny Method (c) Roberts Edges (d) LOG edges (e) Sobel.

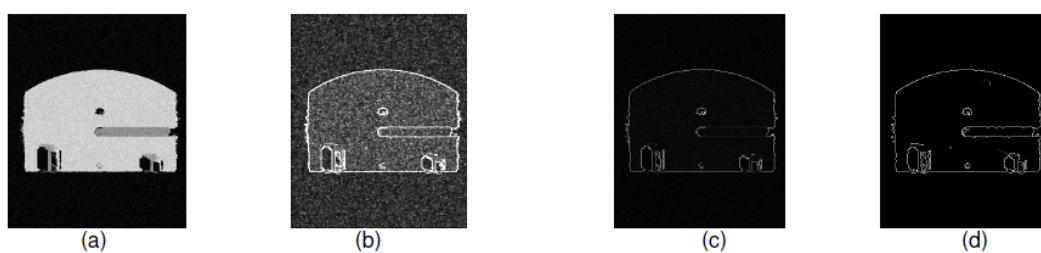


Figure 19 Comparison of Edge Detection technique on Noisy Image (a) Original image with Noise (b) Sobel (c) Robert (d) Canny.

Classical (Sobel, Prewitt, Kirsch...)

Advantages: Simplicity, Detection of edges and their orientations

Disadvantages: Sensitivity to noise, Inaccurate [24].

CHAPTER 3

DESIGN AND APPLICATION OF PREPOSED WORK

3.1. Overview

In this chapter we are going to describe the implementation the algorithm of Car Plate Detection, we are using very powerful software tool which is MATLAB in our algorithm because it provides easy and quicker implementation compared to C and C++ and it contains library functions of image processing that makes easy dealing with our algorithm to make horizontal and vertical histogram counting and get histogram differences. Once the algorithm was developed it was verified with multiple input images containing Iraqi car plates. A flow-chart showing chapter one Figure 1 and each implementation of algorithm is shown below

3.2. Converting the RGB Image into Gray Image

First step we convert image into gray scale for better and easy processing of pixels, the content of colored images are Red, Green and Blue values these values are not used for this algorithm. If the input image is RGB image in MATLAB represented as an array of 3-dimensional RGB, then before further processing in MATLAB it is converted into an array of 2-dimensional gray image. The original input image RGB is shown in figure 20 and gray image is shown in figure 21.



Figure 20 Original Colored Image (Car.jpg).



Figure 21 Gray Scale Image.

3.3. Binrizing

Mean convert gray scale into binary image, here we multiply the binary image to 1.3 we have checked all probabilities then we get best results with 1.3 as a threshold for our image to get 30% better results to clarify Iraqi car plates and increase the plate details information in an image we need first binary image as shown in figure 22. The image will contain 1(Wight pixels) and 0(Black pixels).



Figure 22 Binary Threshold Image.

3.4. Edge Detection

In edge detection we use Robert edge over our image to mark edge points, one of reasons we have use Robert that it works best with binary images. To delete the unnecessary region in an image we need only to work with the edges of the image. For the detection of the edges, we make use of a built-in MATLAB function. But first we put the original image into a grayscale image and binrize it, we can calculate MATLAB for different intensities in the binary image, anything above a certain threshold and that will be white and everything under threshold black. We have thus as it were a binary image. Finally, the edge detection algorithm is applied to binrized gray scale image as in figure 23 we can clearly see the result.

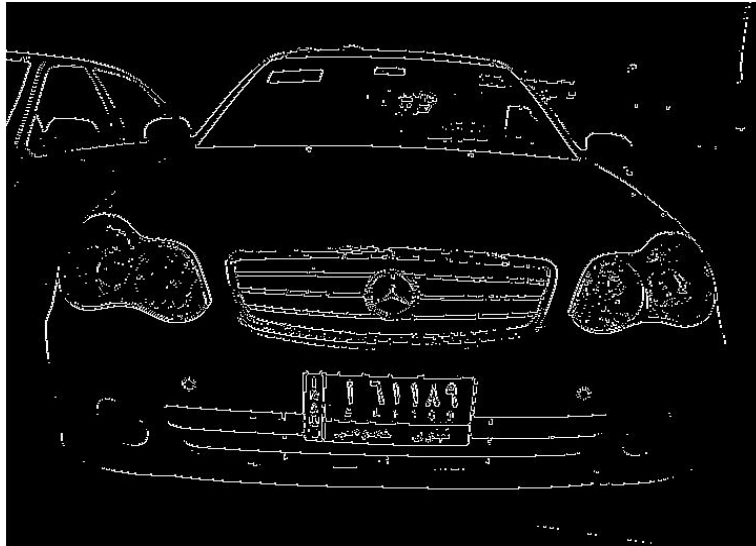


Figure 23 Roberts Edge Detection.

3.5. Edge Processing of Horizontal and Vertical Histogram

Before vertical and horizontal histogram process we first check image tool box to show that every pixel content as white pixels and black in every column and row by using MATLAB function as shown in figure 24.

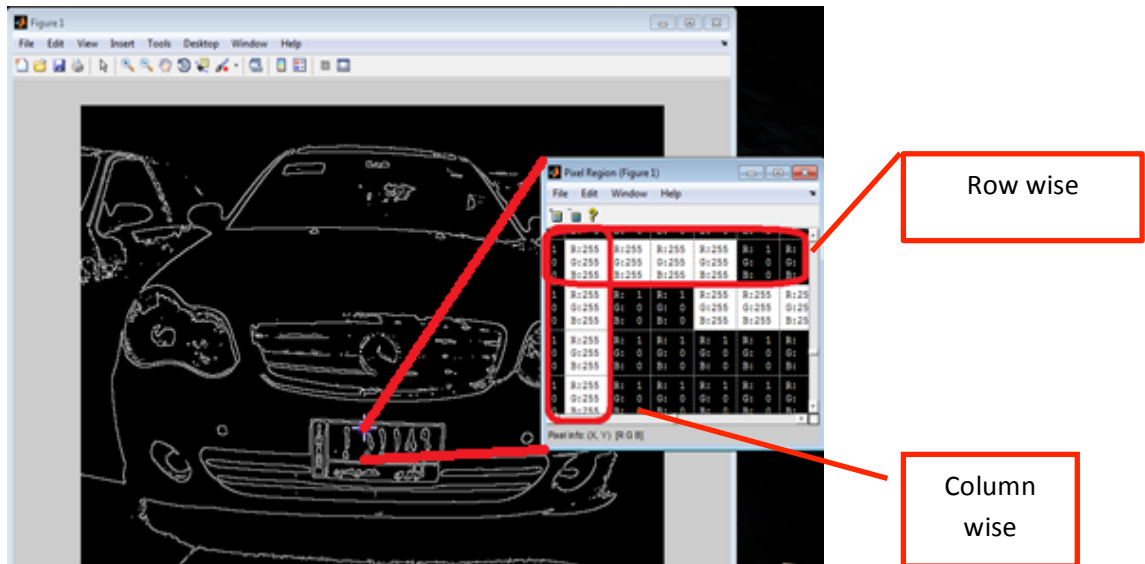


Figure 24 Row and Column Content.

We are going to make horizontal and vertical histogram processing, for horizontal operation first we collocate every pixel in each column and we get the maximum of column average of containing white pixels we set the image to grayscale as the difference in gray value between successive pixels exceeds a certain limit, we add on to the total of that particular column. Finally, all column totals are stored in a matrix. If we were to plot this matrix we get a result as in Figure 25. Showing horizontal histogram from this it can be derived the horizontal position of the plate in many cases.

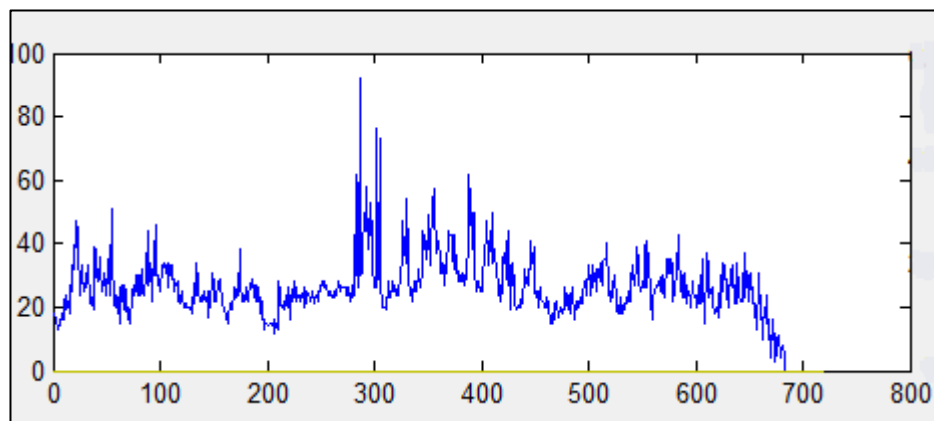


Figure 25 Horizontal Histogram (X-Axis: Position of Column of Pixels on the Image, Y-Axis: Number of Containing White Pixels Close Exceeds A Threshold).

Our application to be the best, detecting the correct region in horizontal edge processing as follows: if the number of white pixels (one edge is white) per column is above a calculated beforehand threshold, we assume that some regions the horizontal position of the indicating plate. In case there are several possible regions selected, we will continue to work with the widest. For vertical edge processing is the method is to not look at the number of white pixels per column, but the number of times a pixel differences to it is neighbor in a row. In the rows of containing a license plate there will be several differences pixels. The results are plotted in vertical histogram in the application as shown in Figure 26. From this it can be derived the vertical position of the plate in many cases the highlighted region is colored in red.

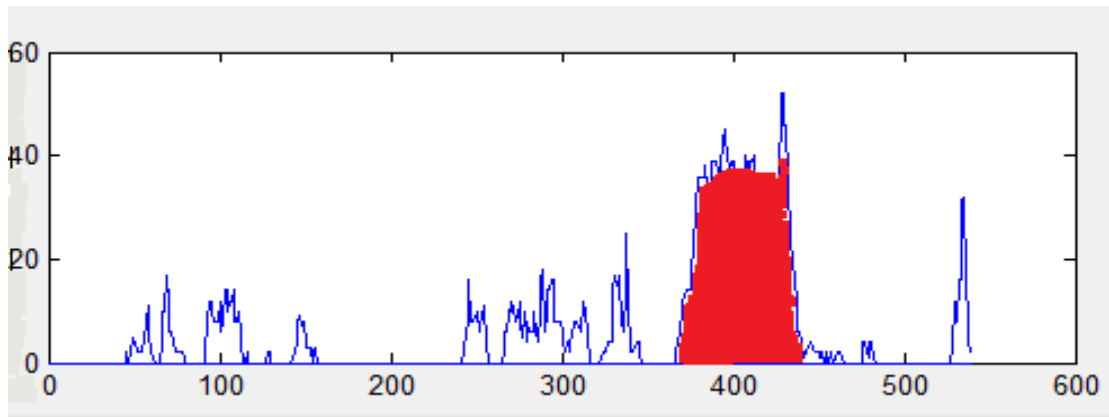


Figure 26 Vertical Histogram (X-Axis: Line Position on the Image, X-Axis: Number of Counted Transitions Zeros and Ones Per Row).

After the detect highest horizontal histogram and vertical histogram we can get the position of the car plate and extract the region for it the output car plate image is shown in Figure 27.

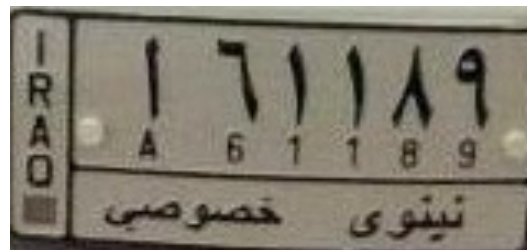


Figure 27 Output Image.

3.6. Investigation Output Image

To start investigation the car plate we first tested some samples of plate images to check the contrast of the car plates as shown in figures below all car plates has high contrast with Wight pixels more than black pixels we can clearly see the histograms of plates in figures below and we are showing the curtain threshold in red line as below:

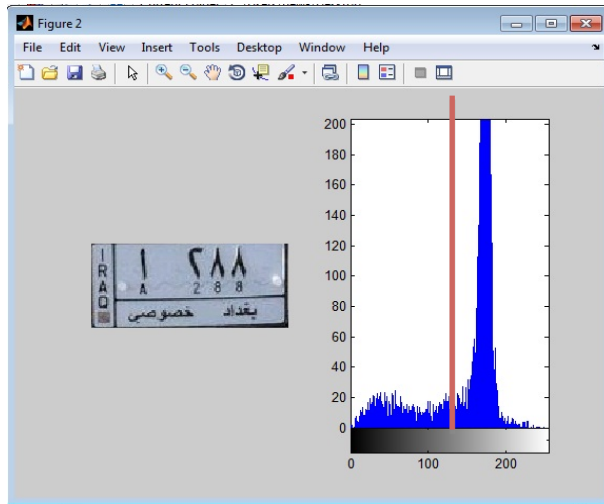


Figure 28 Car Plate Test 1.

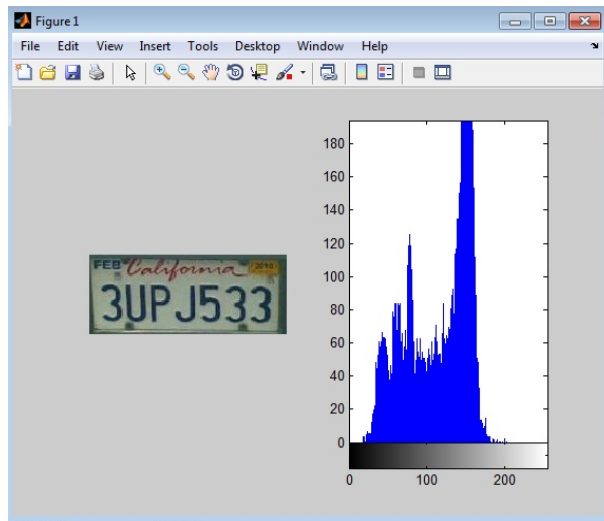


Figure 29 Car Plate Test 2.

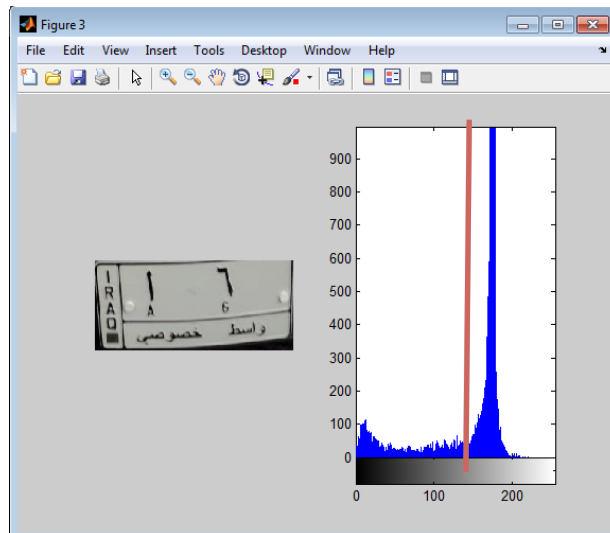


Figure 30 Car Plate Test 3.

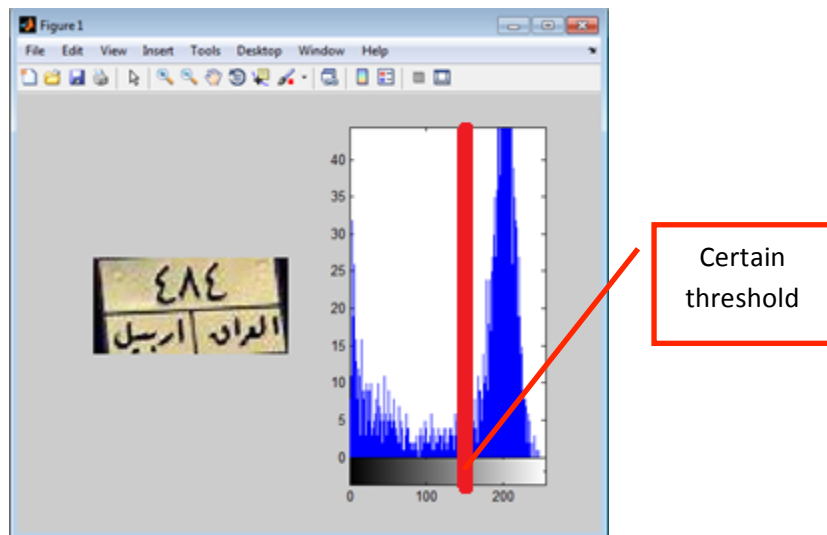


Figure 31 Car Plate Test 4.

All sample images have been processed when (x-axis is number pixels on the image and y-axis: pixels rate of contain gray scale from black to white).

From here we can assume a contrast threshold 150 as middle of the gray scale level if the whiteness contains high histogram than the black histogram. Then the final step is check the output image of our program is it a car plate or not, by checking the contrast of the image and comparing to the plate contrasts as we tested plates as we see in Figure 32 the diagram of algorithm that checks the contrast of the car plate.

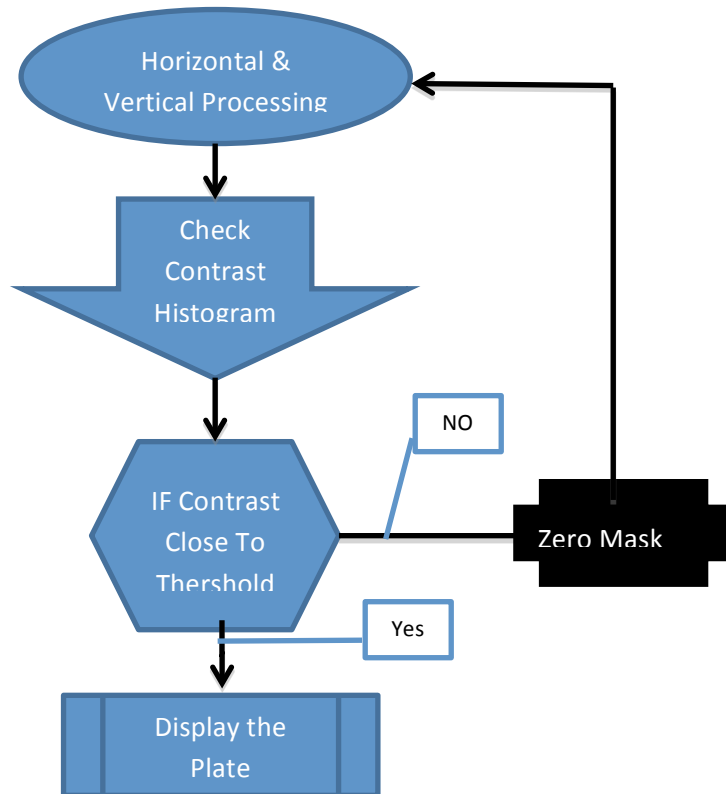


Figure 32 Contrast Check Diagram.

Here the output image of our program is shown below in figure 33. And the histogram of the output is shown in figure 34.

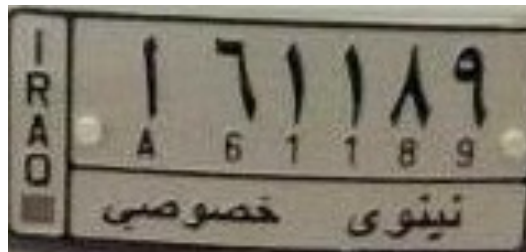


Figure 33 Output the Car Plate.

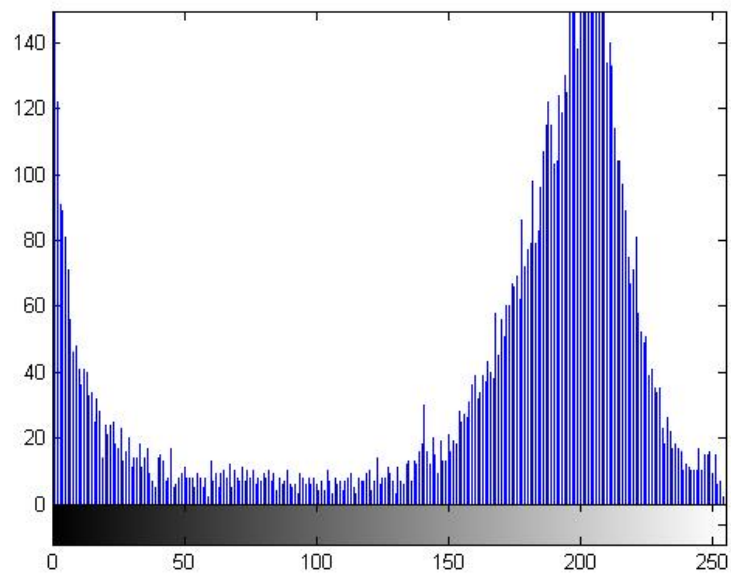


Figure 34 Histogram of the Car Plate.

Here we can see the plate is contains a car plate contrast as we threshold .In other conditions we are showing part of images have high horizontal and vertical histograms and it is not a car plate as showing in figure 35. And the histogram of it is shown in figure 36 as below:



Figure 35 Car Parts with High Histogram.

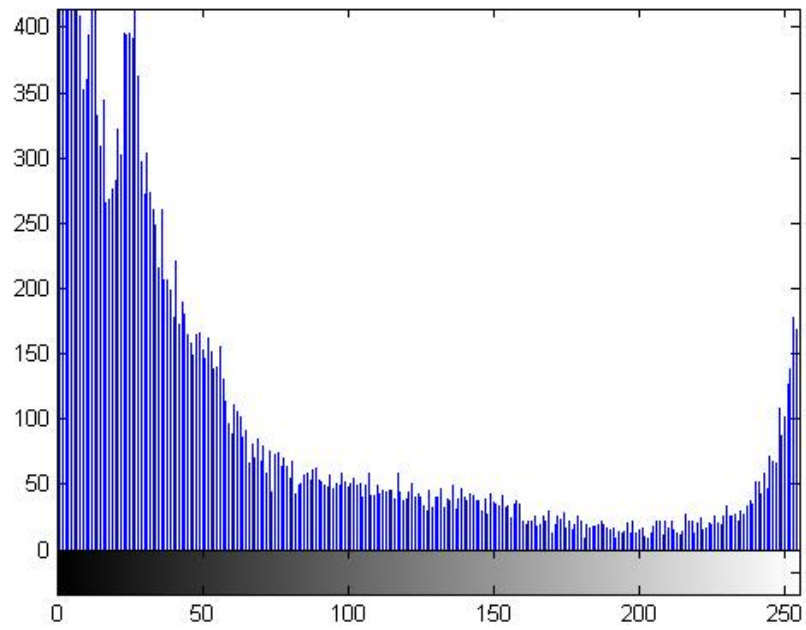


Figure 36 Histogram of Car Parts.

As we see the car plate image has high white contrast histogram by checking every pixel in the output image we can know the difference of pixels by making threshold if it contains white more than black pixels.

CHAPTER 4 EXPERIMENTAL AND RESULTS

4.1. Implementation

In this method, we used the MATLAB R2011a as performing program. The code is applied with GUI. This thesis applied to many car plates with different types, numbers, size and colors and plate of different Countries.

The figures below show car plates of three different countries. At first the input image is Iraqi car plate that has special size and types and with Arabic text inside it. The program extracts the highest horizontal and vertical histogram after converting image into gray scale and adding Robert edge over it and binarizing the image then the program extracts the plate with black and white color as shown in figure 37.

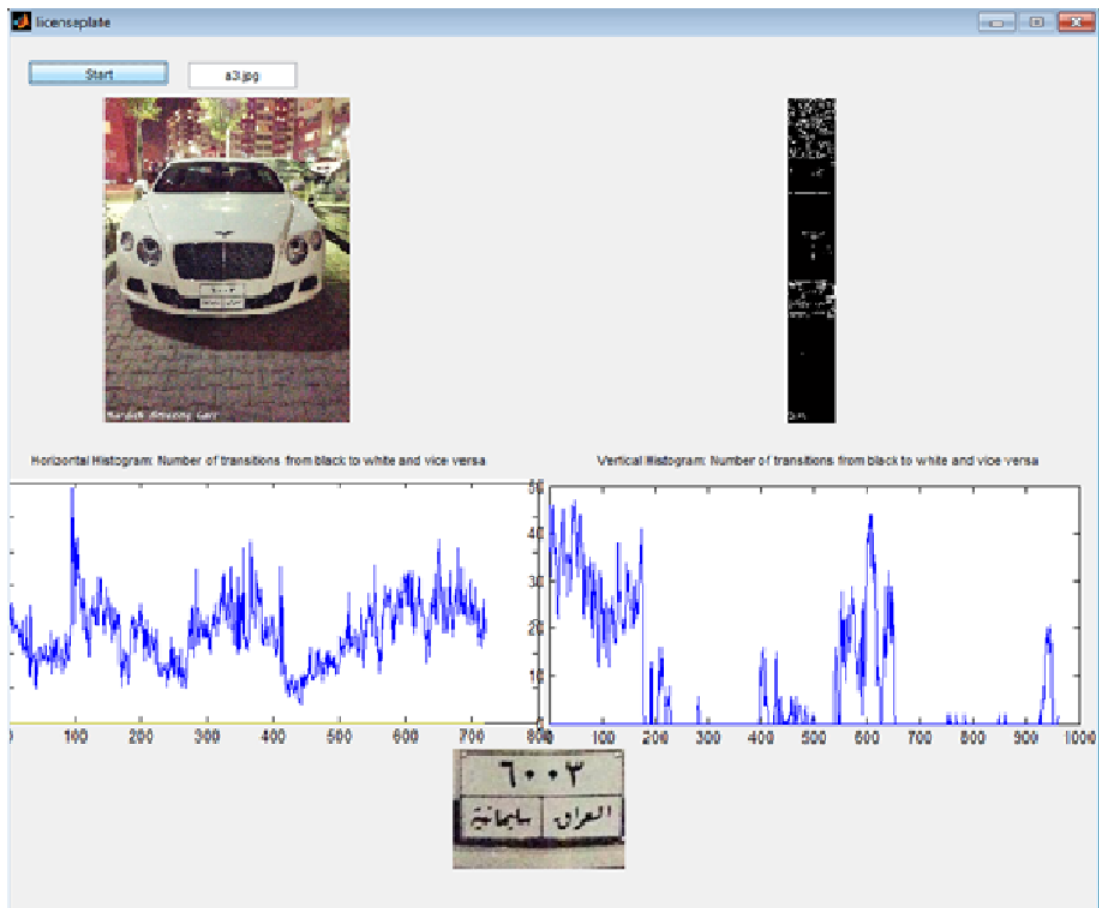


Figure 37 Iraqi Car Plate Detection.

There are some problems of image for a car that has many parts of high horizontal and vertical histograms the output of the program for such image is shown in figure38.

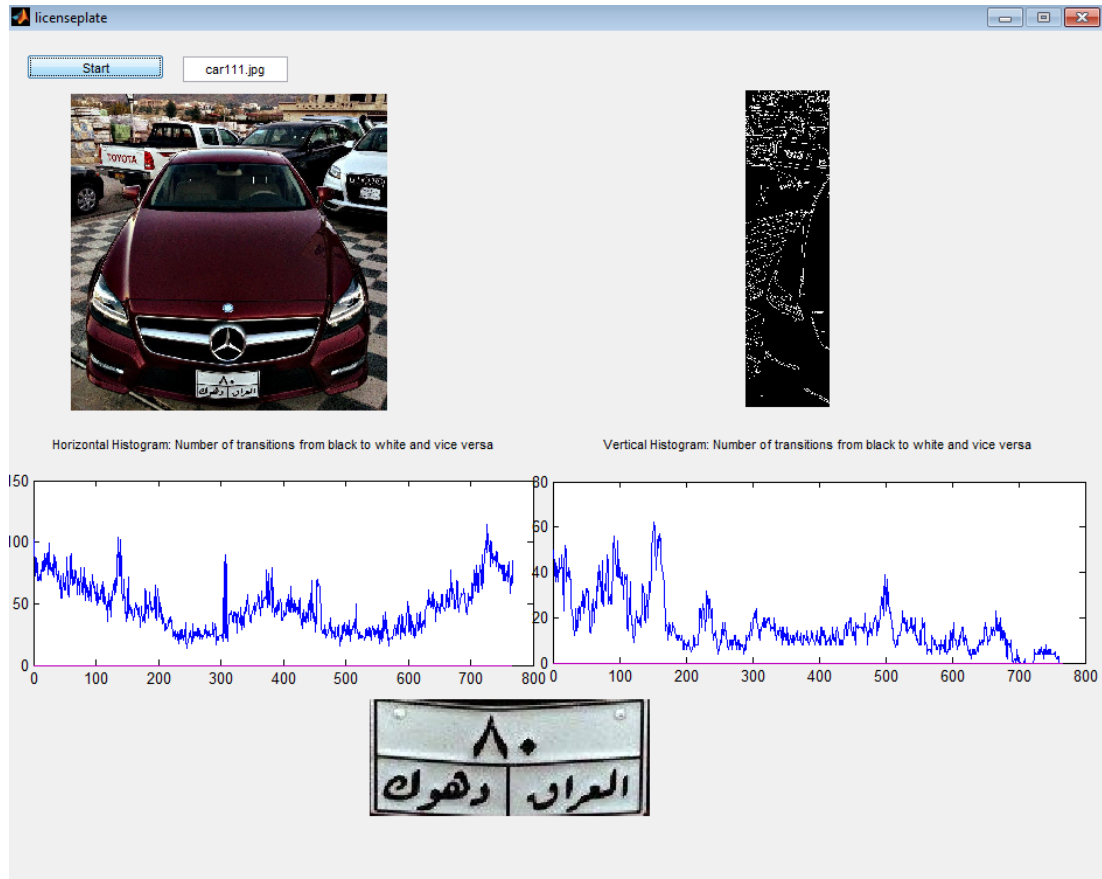


Figure 38 Iraqi Plates Parts of High Horizontal and Vertical Histograms.

The program gets the highest horizontal histogram than the highest vertical histogram, and there are many parts in one image with high histograms of its edges and it is not a plate. By checking the output of the horizontal and vertical histograms, the contrast of the output image in gray scale if white pixels are more than black pixels it is a car plate and by dividing the average of white pixels over black pixels the results will be 3 and we make it as a threshold if any result near to 3 means white pixels over black pixels and if the output is close to the threshold number it is a car plate. In figure 2 the process of the program gets many parts with high horizontal and vertical histograms as shown in figure 39.



Figure 39 Result of High Horizontal and Vertical Histogram.

And these parts with high horizontal and vertical histograms have different contrasts which contains more black pixels than a plate as is shown in figure 40. The histogram is very high in black space.

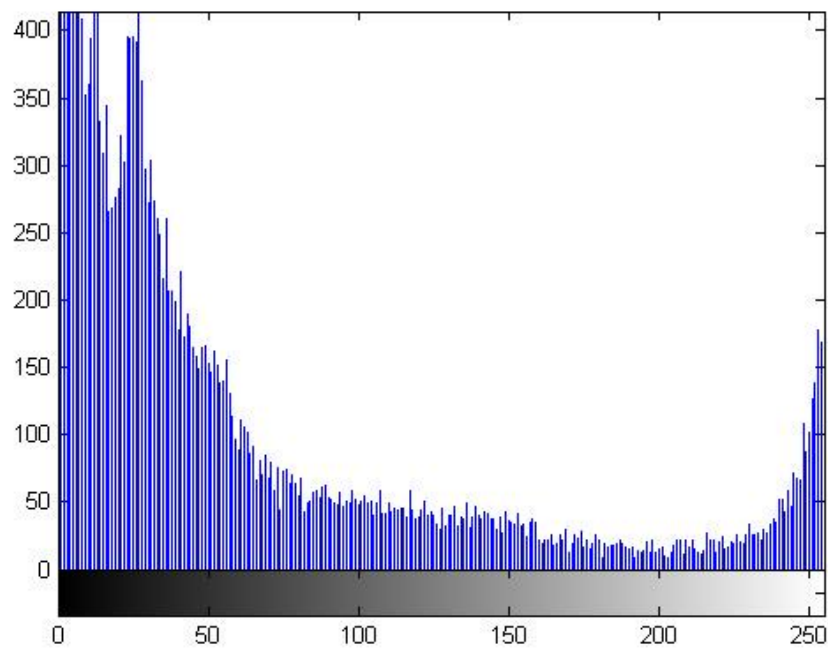


Figure 40 High Histogram Output Image.

After checking the contrast of the image if it is not close to the contrast threshold it will be converted into black by multiplying it to zeroes matrix, as shown in figure41. The image of car has two parts with high vertical and horizontal histogram and with contrast that it is not close to the threshold therefore it converted to black as shown below:



Figure 41 Image with High Parts of Histogram Converted to Black.

By eliminating the parts with high histogram and it is not a plate. While processing, the program gradually detects the car plate as shown in figure 42.

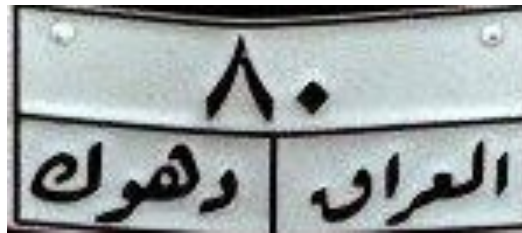


Figure 42 Final Output of the Program.

Here we check the contrast of the final output as shown below in figure 43.

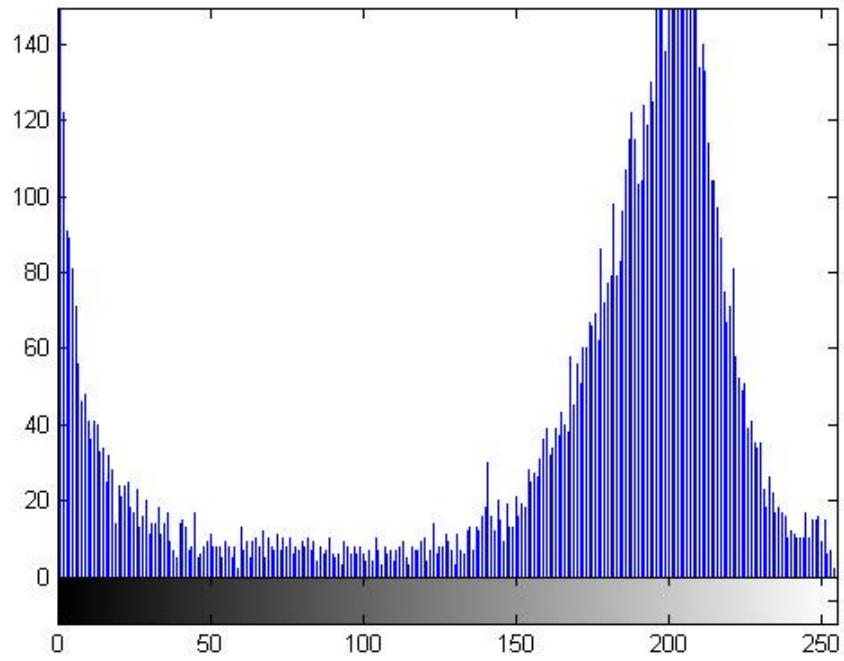


Figure 43 Histogram of the Car Plate Final Output.

After checking the histogram of contrast image and it is close to the contrast threshold the program shows the number plate as final output.

And our algorithm works with colored plates also as shown in figure 44.

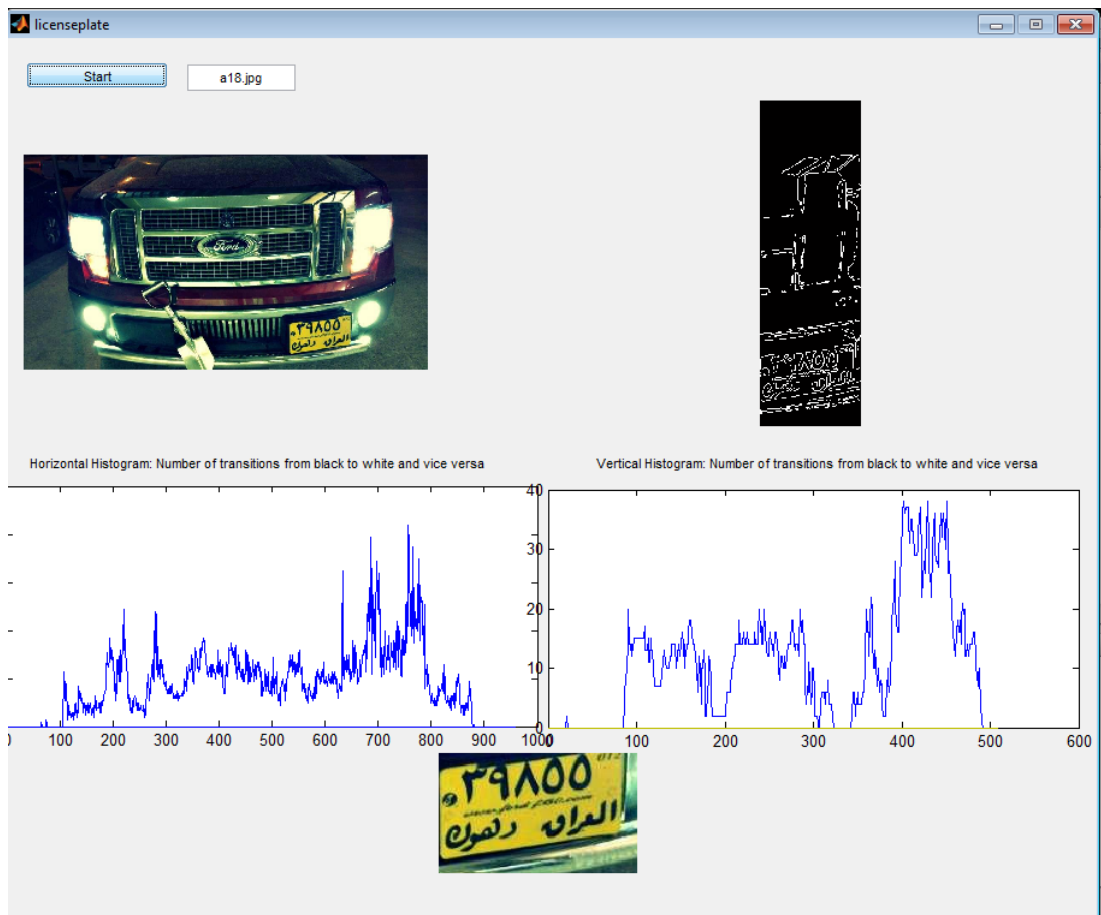


Figure 44 Input Image with Yellow Car Plate.

The program converts colored image into gray scale, in a gray scale the intensity of a pixel is expressed within a given range between a minimum and a maximum, this range is represented as 0 (black) and 1 (white).

Also the program works with different plate size and types like shown in figure 45. The input image is European car plate.

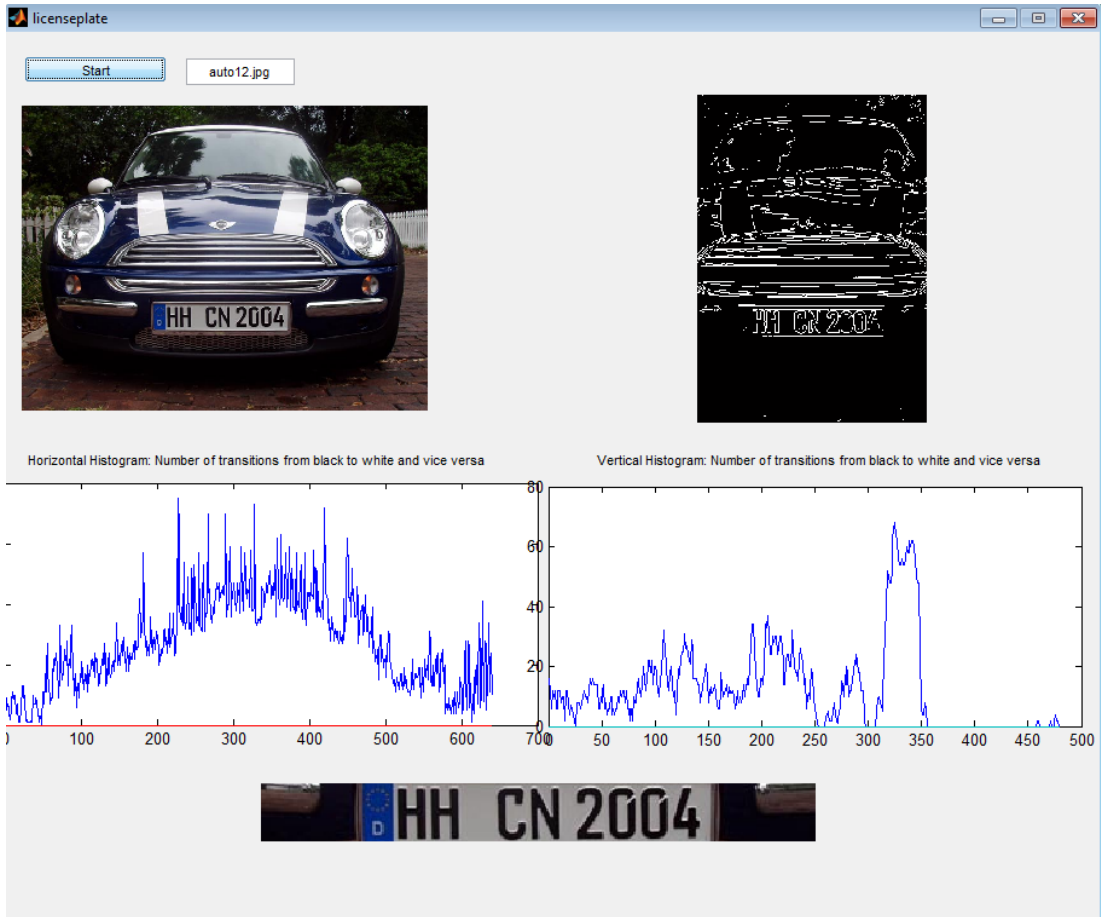


Figure 45 European Car Plates.

The program checked on Netherlands car plates as shown in figure 46.

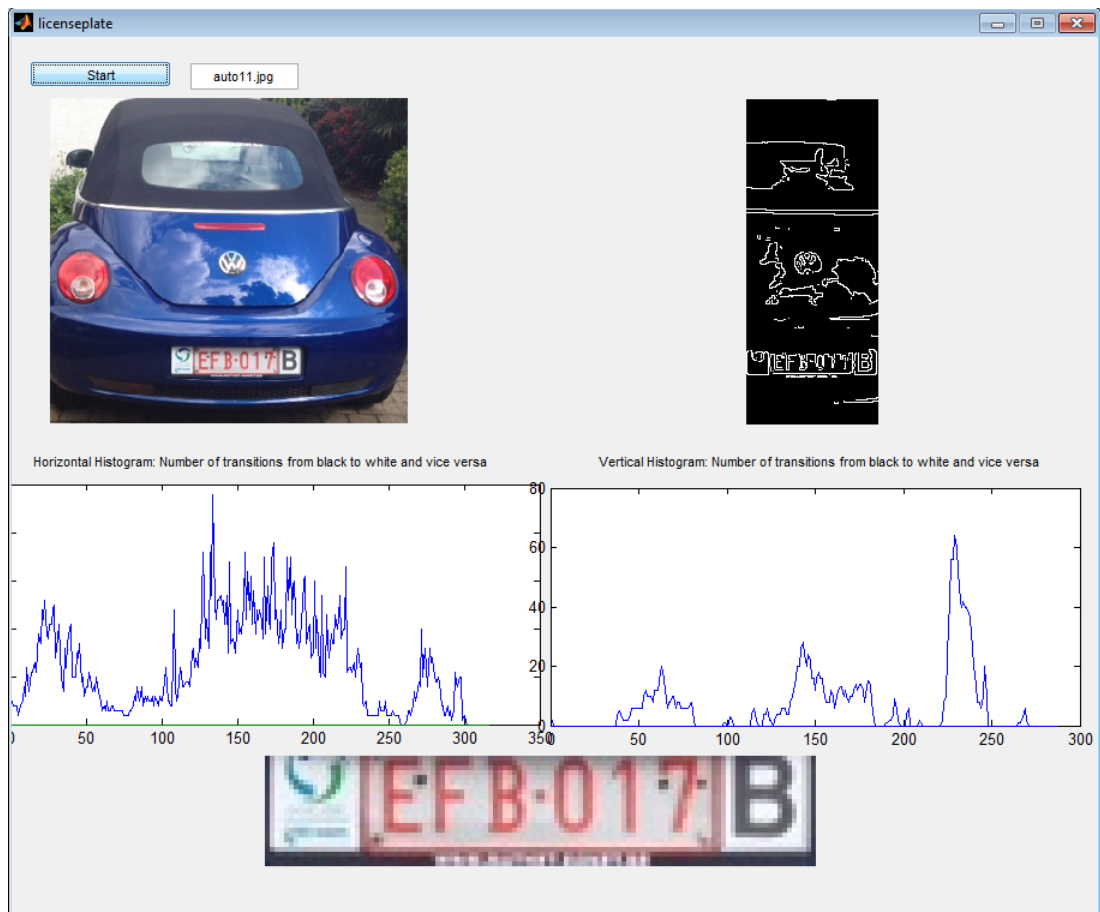


Figure 46 Netherland Car Plate.

4.2. Results and Discussions

The algorithm correctly detected more than 50 car plates the 25 of it was Iraqi number plates and other 25 for different European car plates. The images contained vehicles of different colors and varying intensity of light and the results are

1-First for Iraqi car plate we have successfully detected 25 Iraqi car plates for different countries with different colors of plate. Some of these images the algorithm has detected the plate without adding a mask of parts with error rate in high vertical and horizontal histogram these images have the correct and appropriate conditions to find the edges. The other images the algorithm has replayed with masking error parts for three to five times till founding the car plate. Examples for detection of the plate regions tested on Iraqi car plate images are shown between figure (47-69):

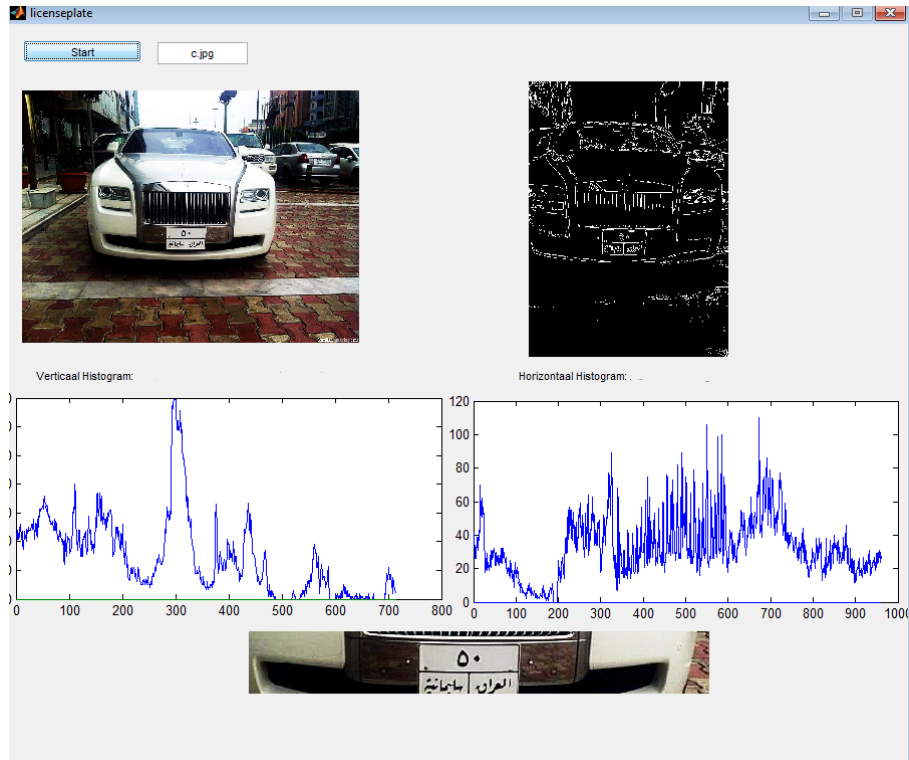


Figure 47 Iraqi Plate Test 1.

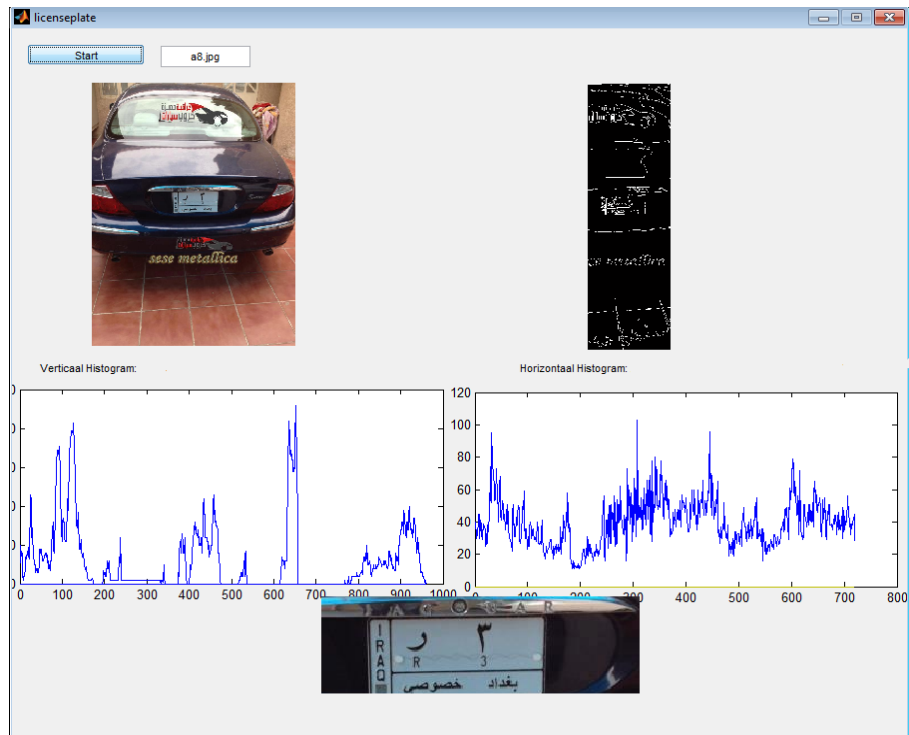


Figure 48 Iraqi Plate Test 2.

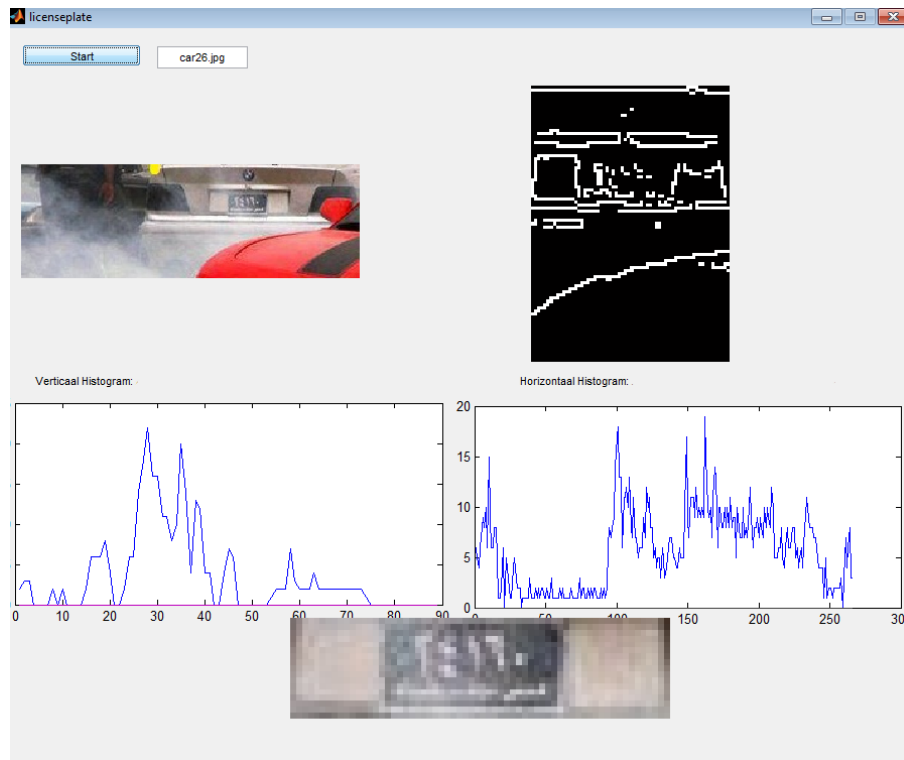


Figure 49 Iraqi Plate Test 3.

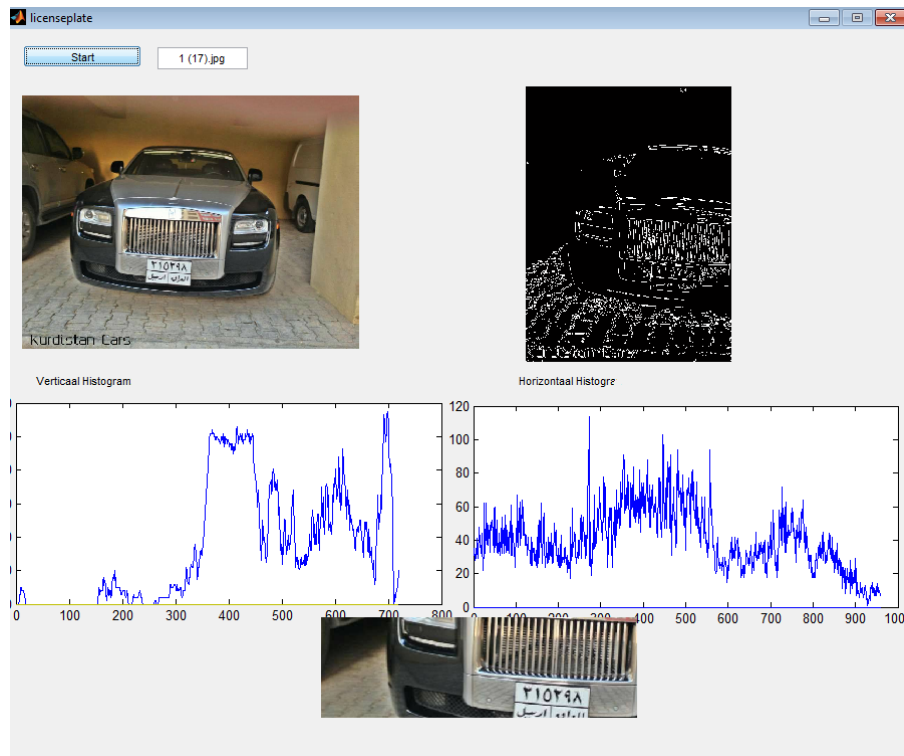


Figure 50 Iraqi Plate Test 4.

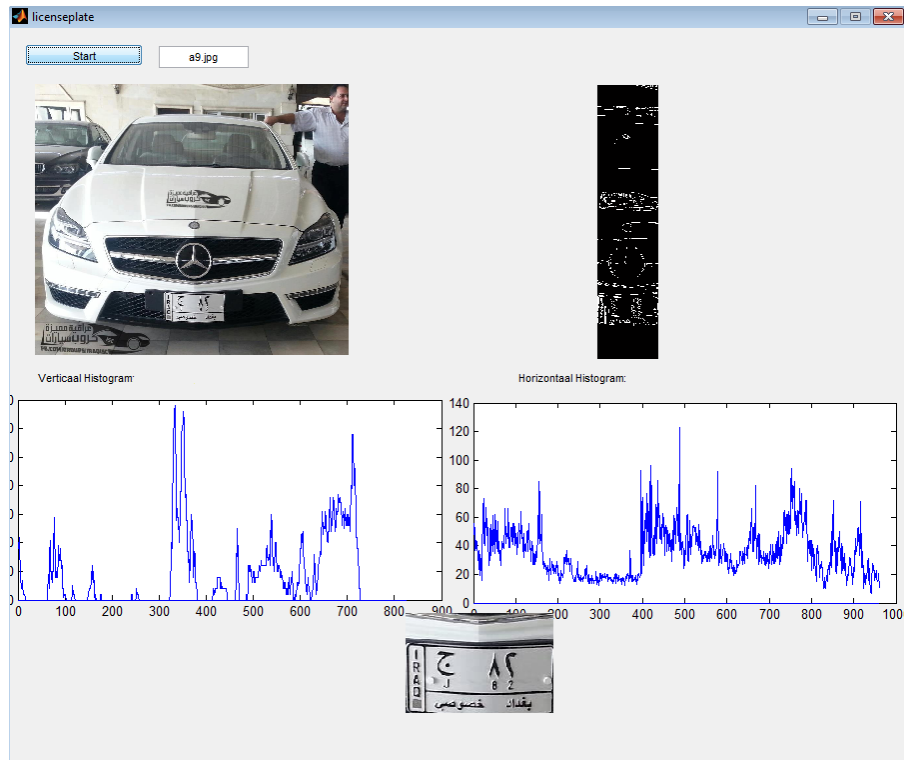


Figure 51 Iraqi Plate Test 5.

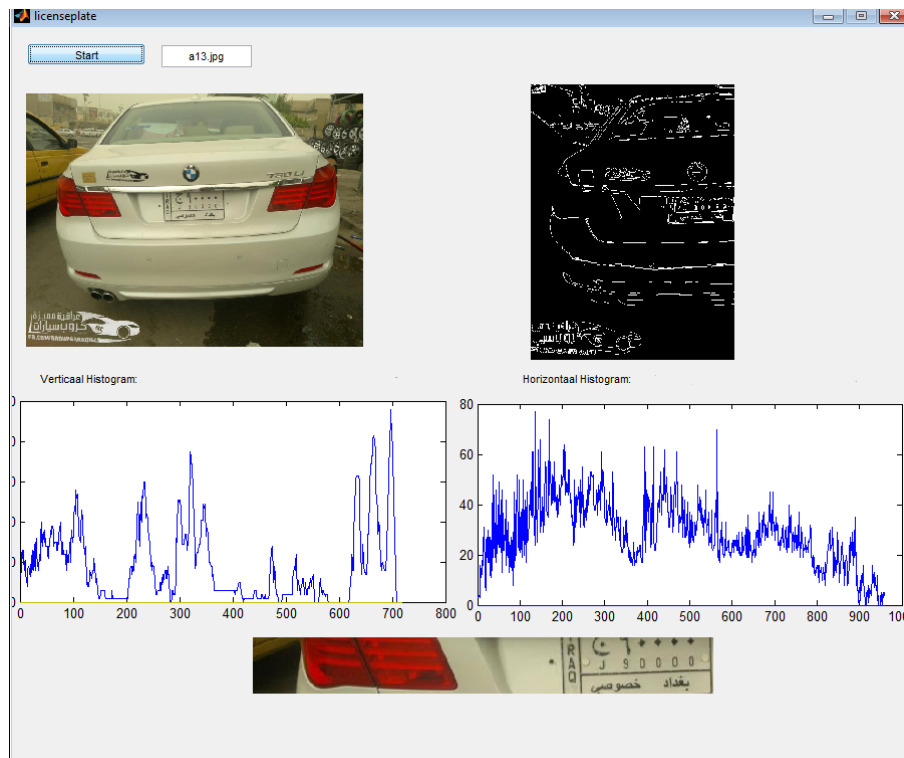


Figure 52 Iraqi Plate Test 6.

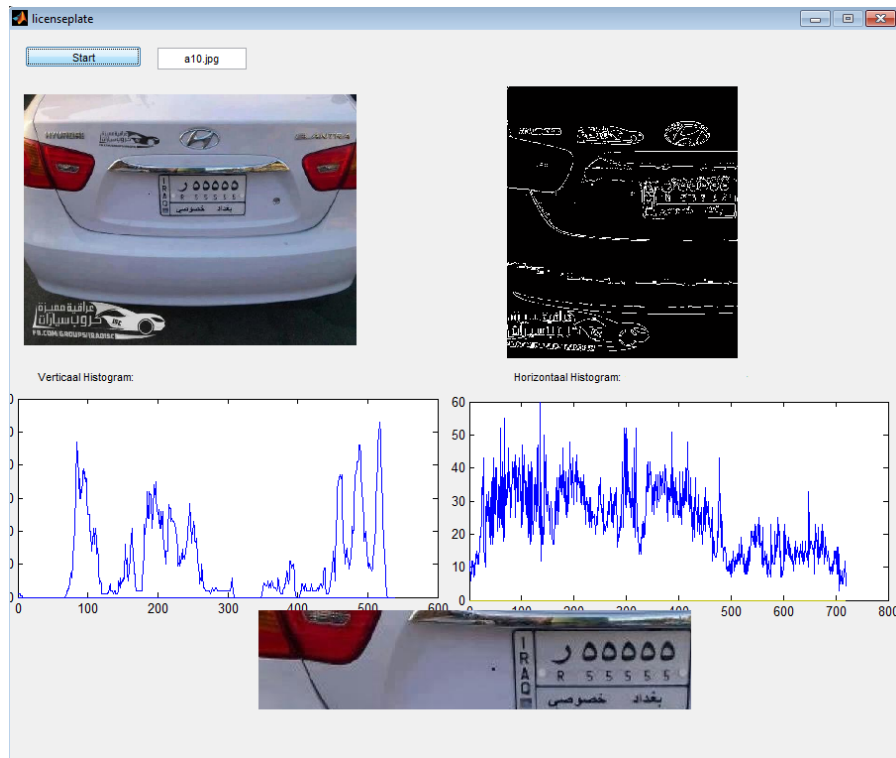


Figure 53 Iraqi Plate Test 7.

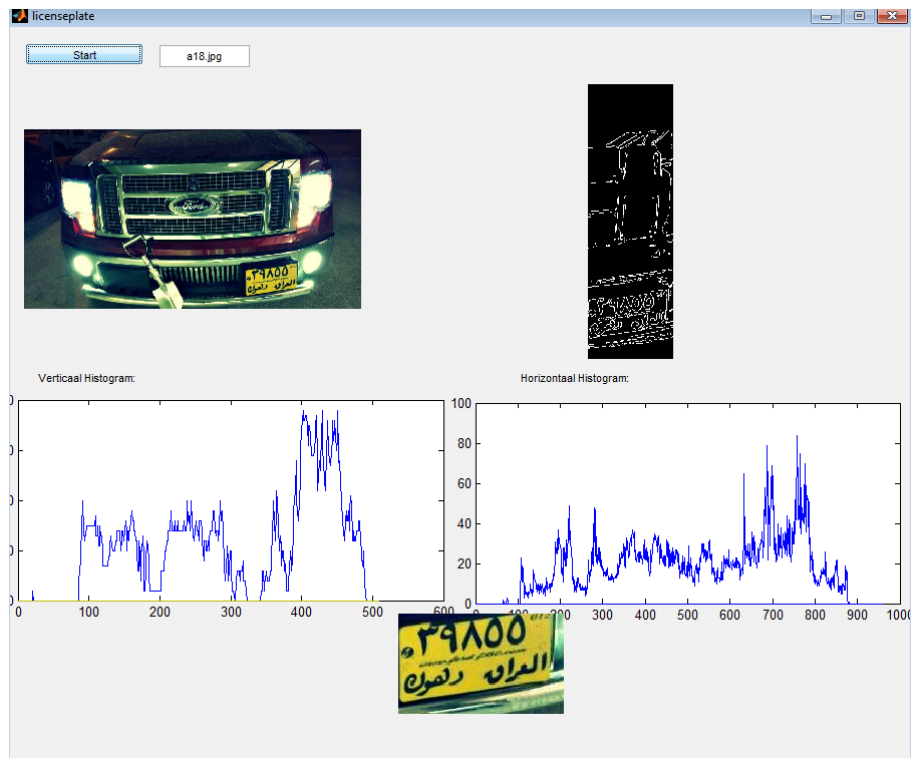


Figure 54 Iraqi Plate Test 8.

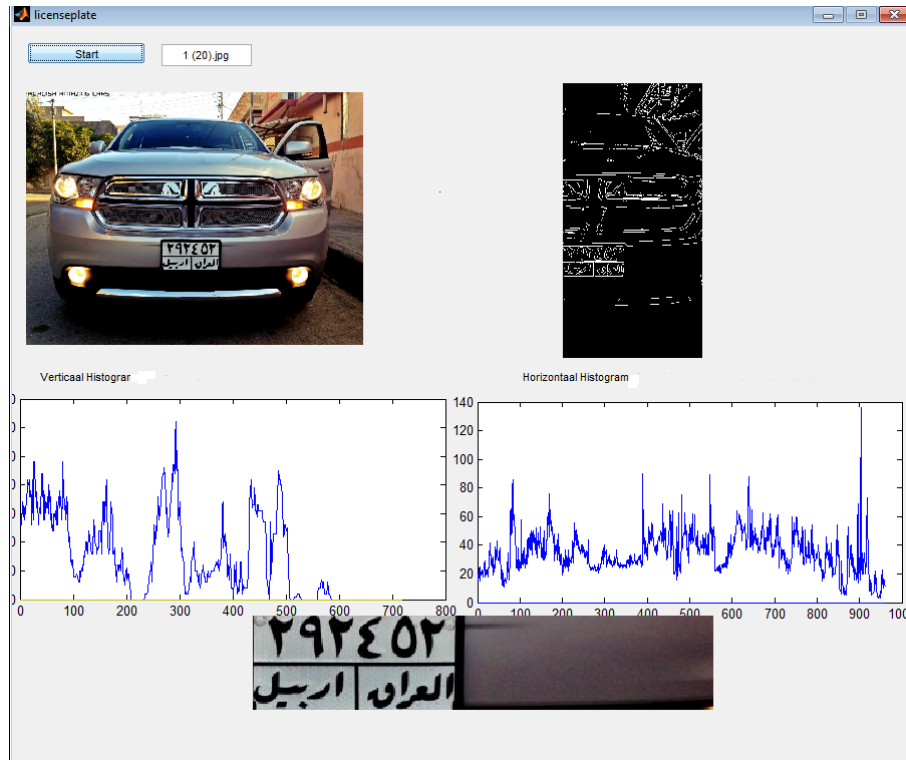


Figure 55 Iraqi Plate Test 9.

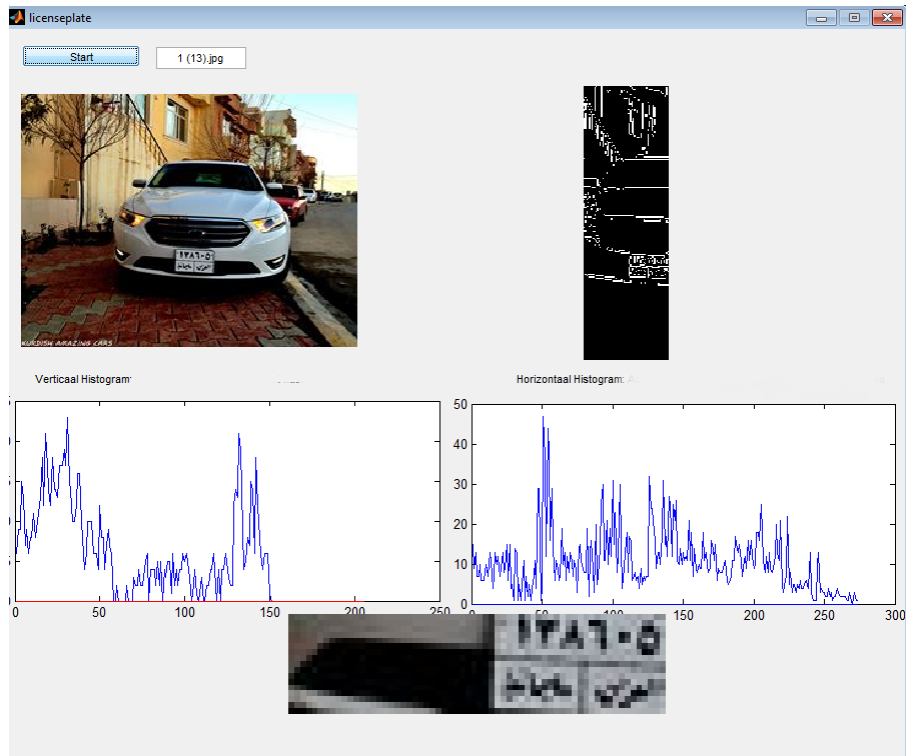


Figure 56 Iraqi Plate Test 10.

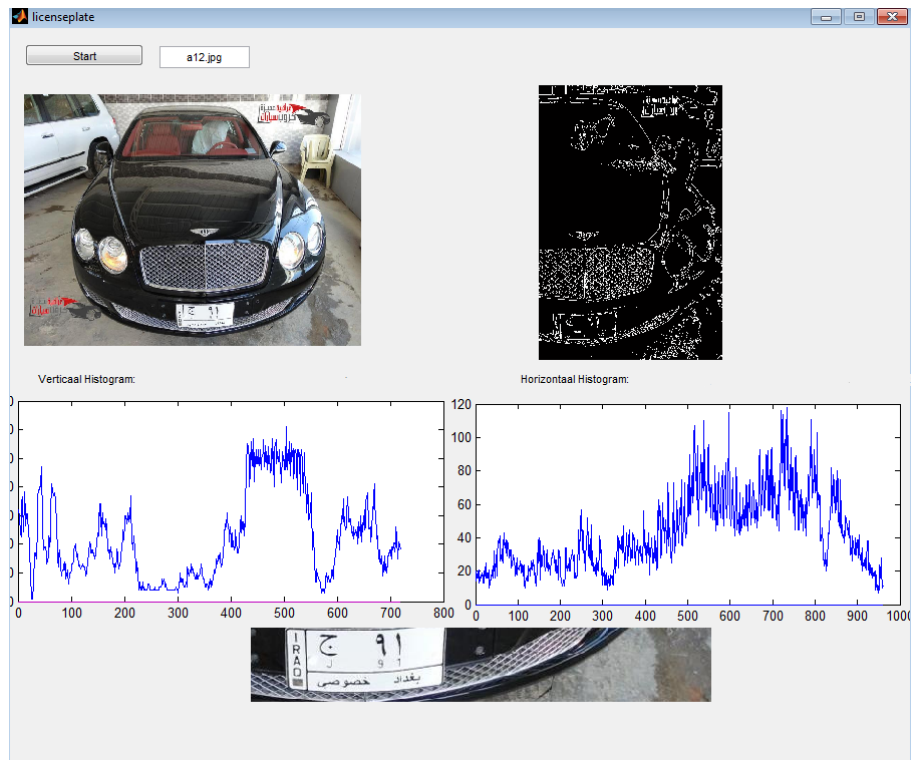


Figure 57 Iraqi Plate Test 11.

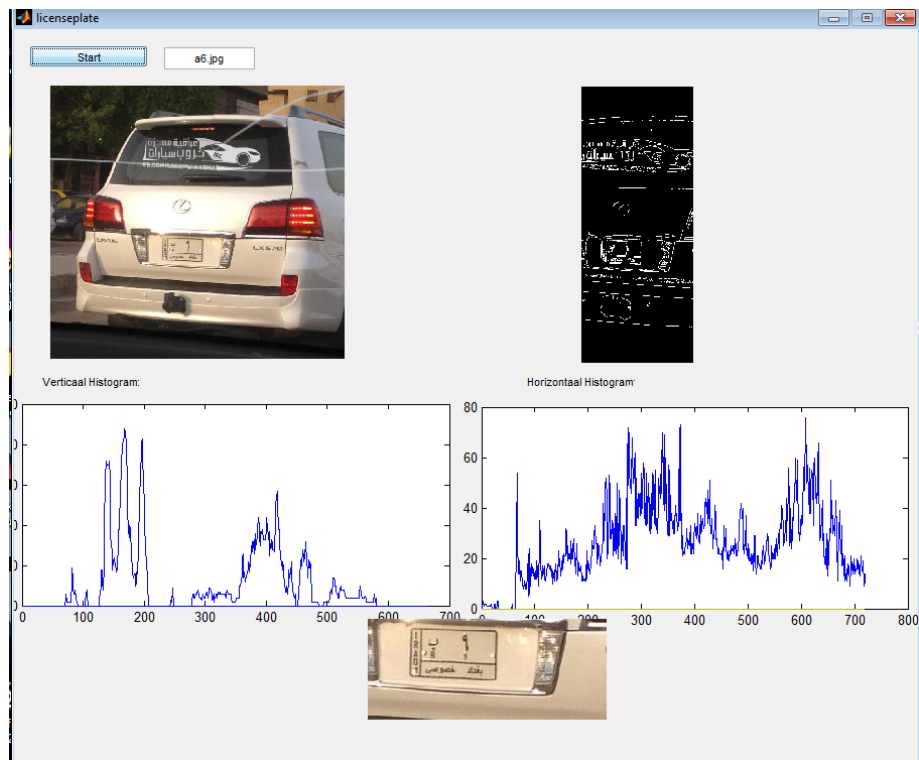


Figure 58 Iraqi Plate Test 12.

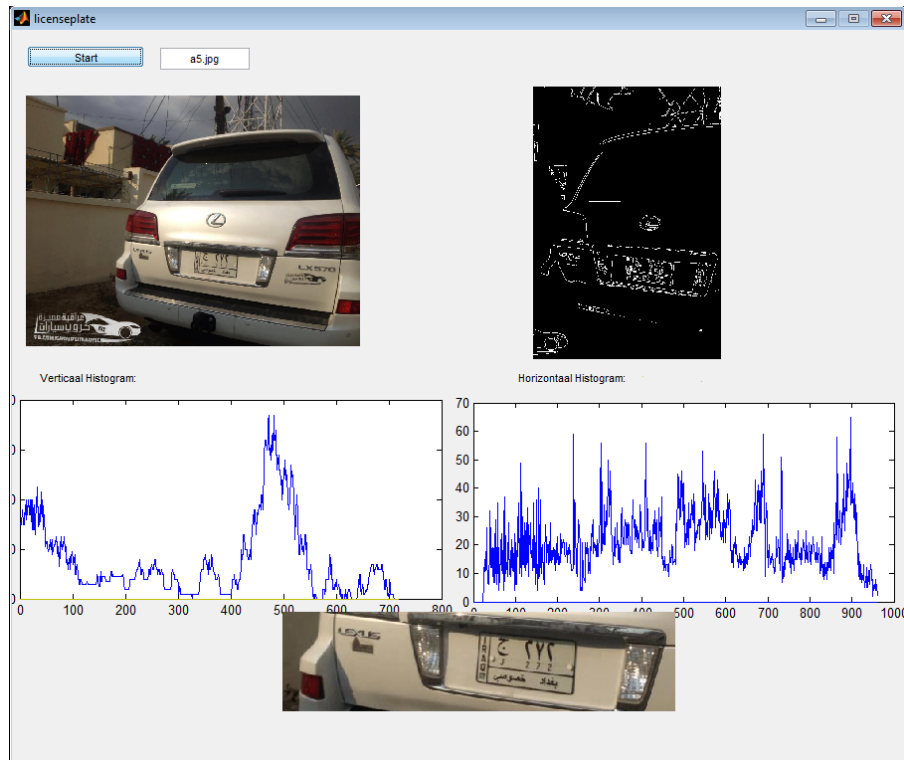


Figure 59 Iraqi Plate Test 13.

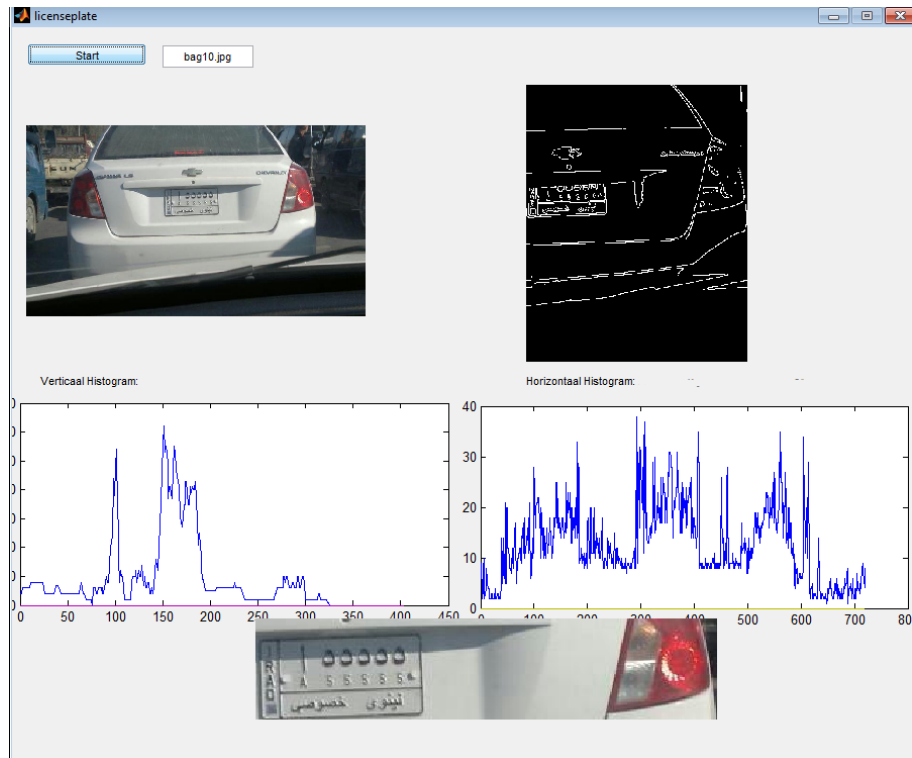


Figure 60 Iraqi Plate Test 14.

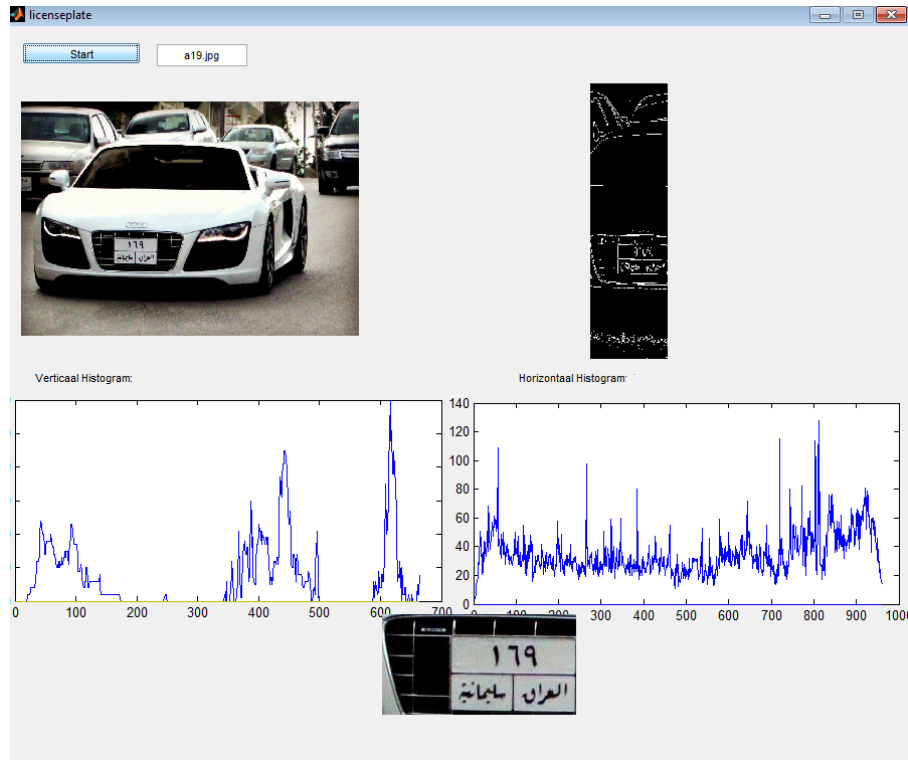


Figure 61 Iraqi Plate Test 15.

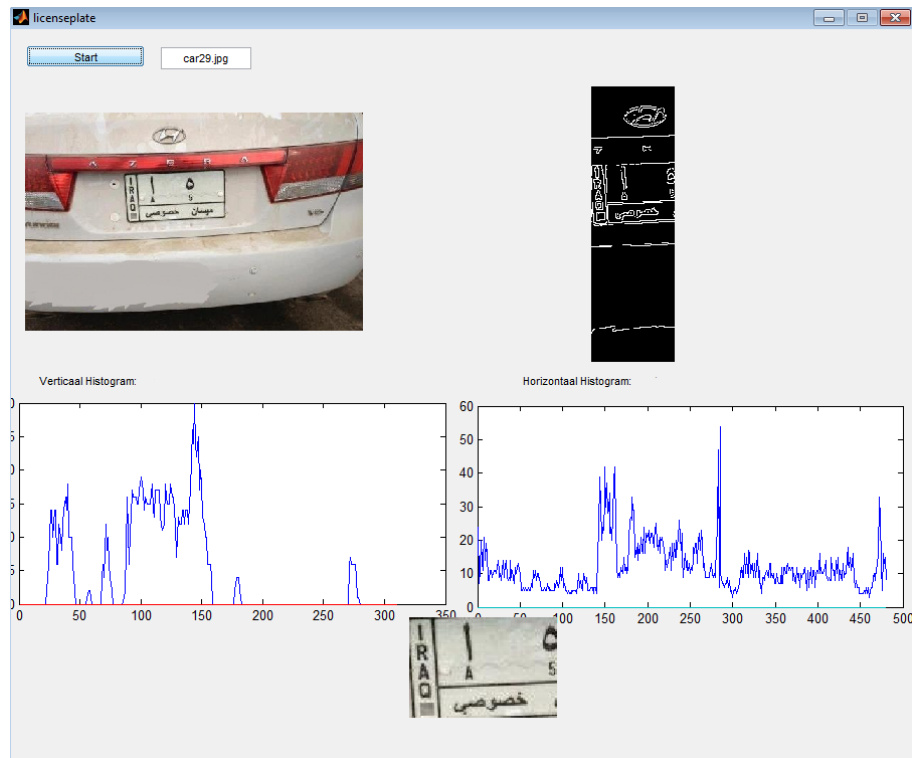


Figure 62 Iraqi Plate Test 16.

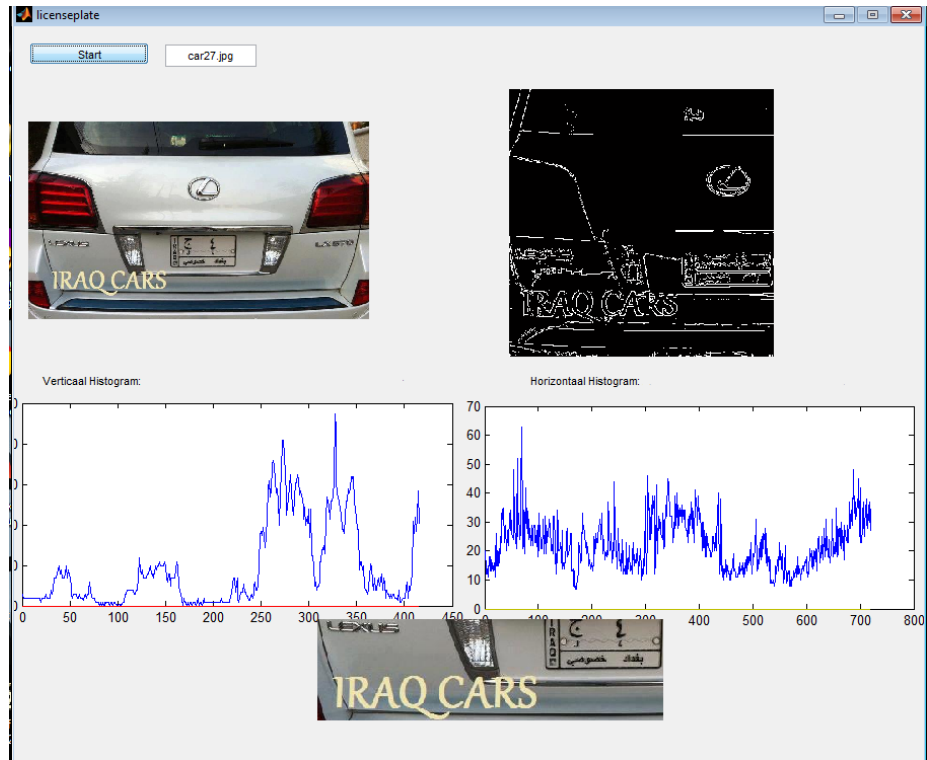


Figure 63 Iraqi Plate Test 17.

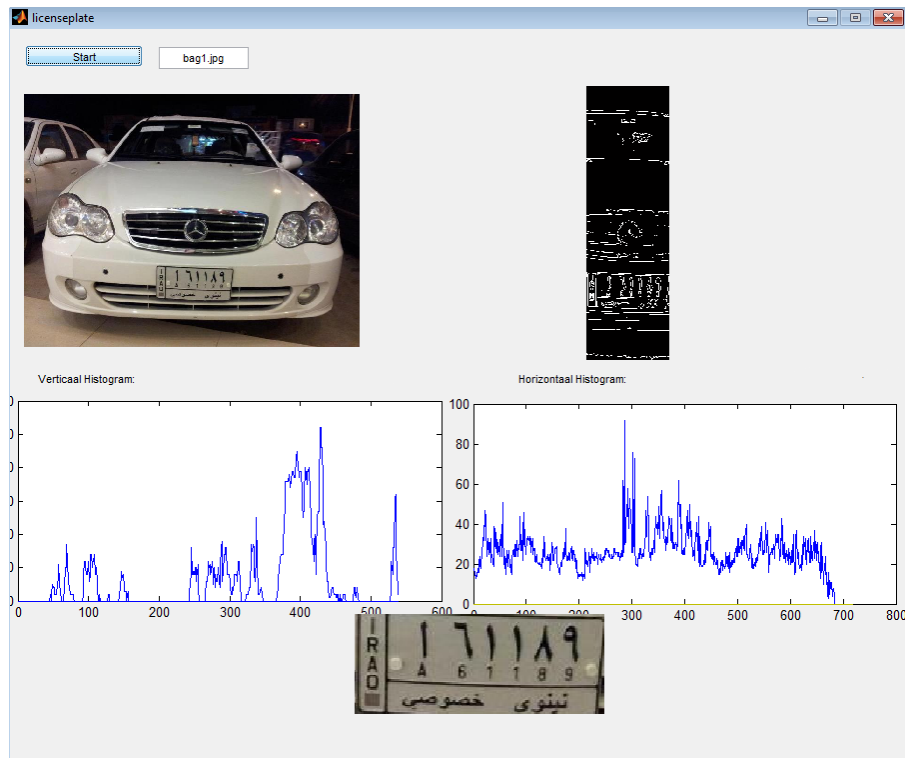


Figure 64 Iraqi Plate Test 18.

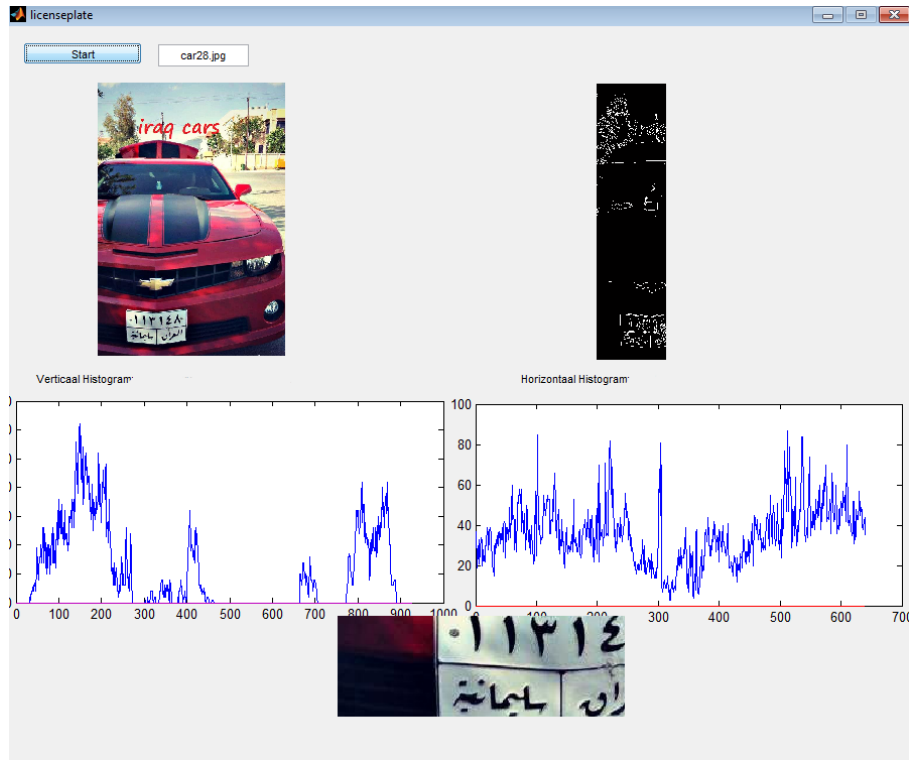


Figure 65 Iraqi Plate Test 19.

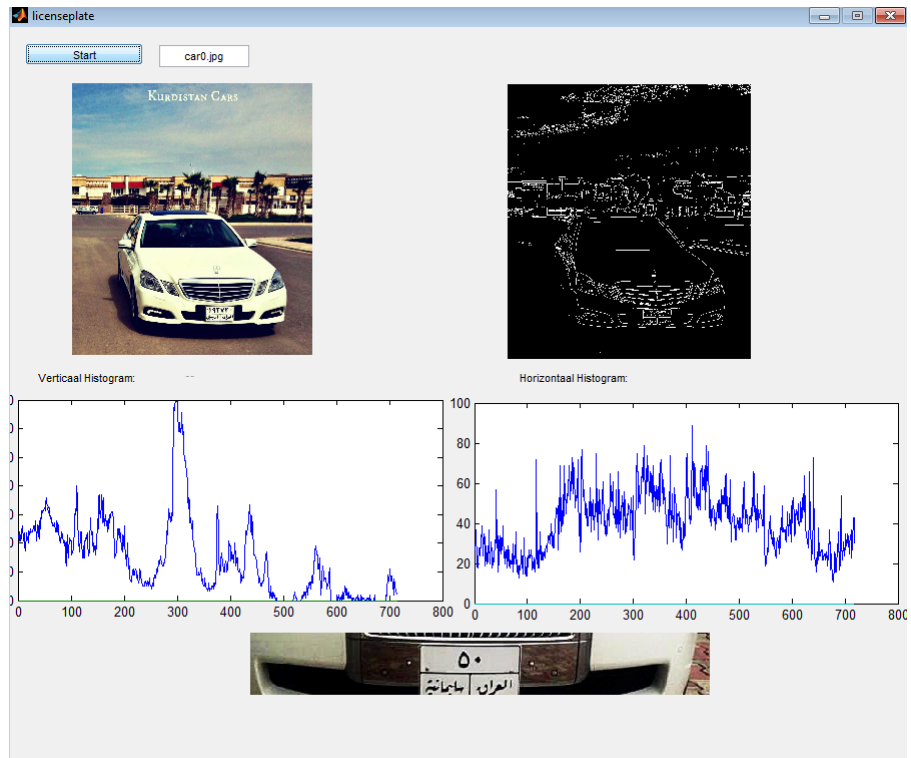


Figure 66 Iraqi Plate Test 20.

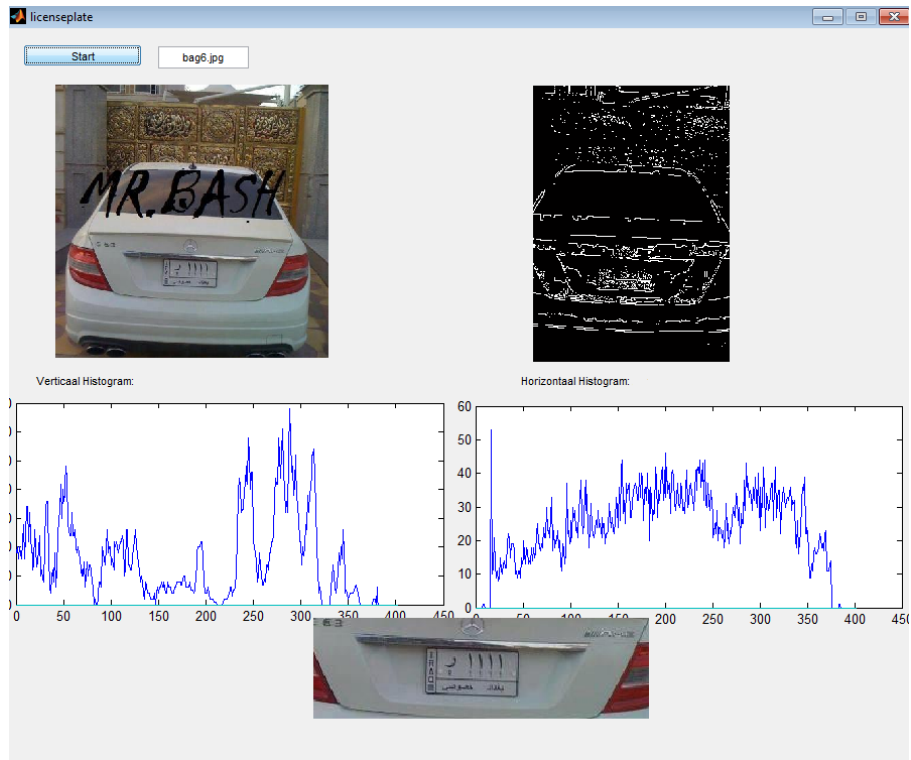


Figure 67 Iraqi Plate Test 21.

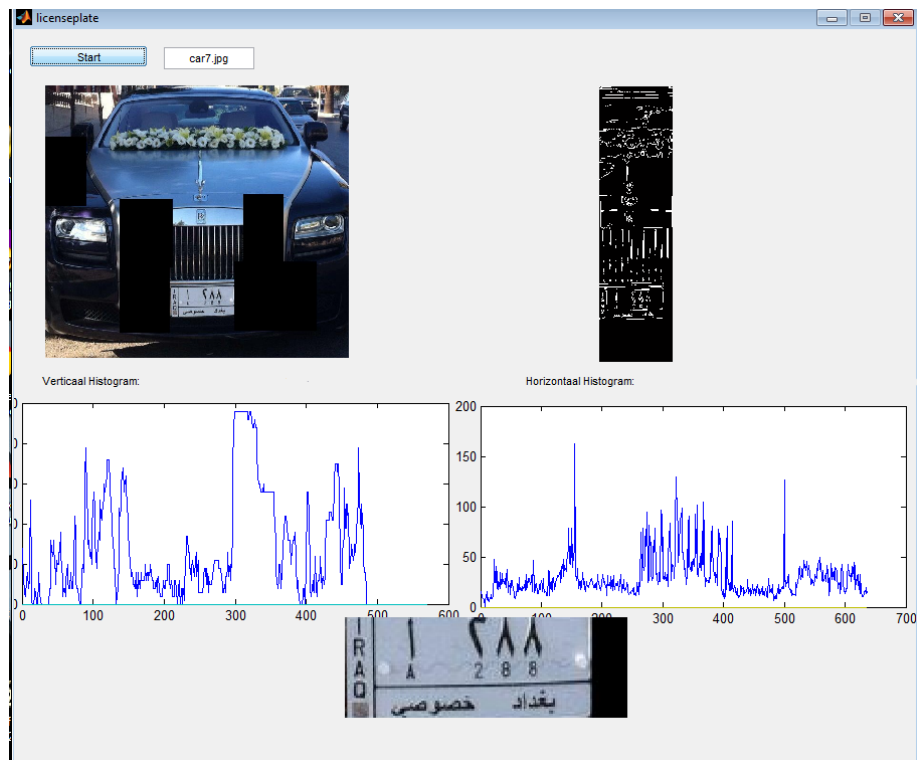


Figure 68 Iraqi Plate Test 22.

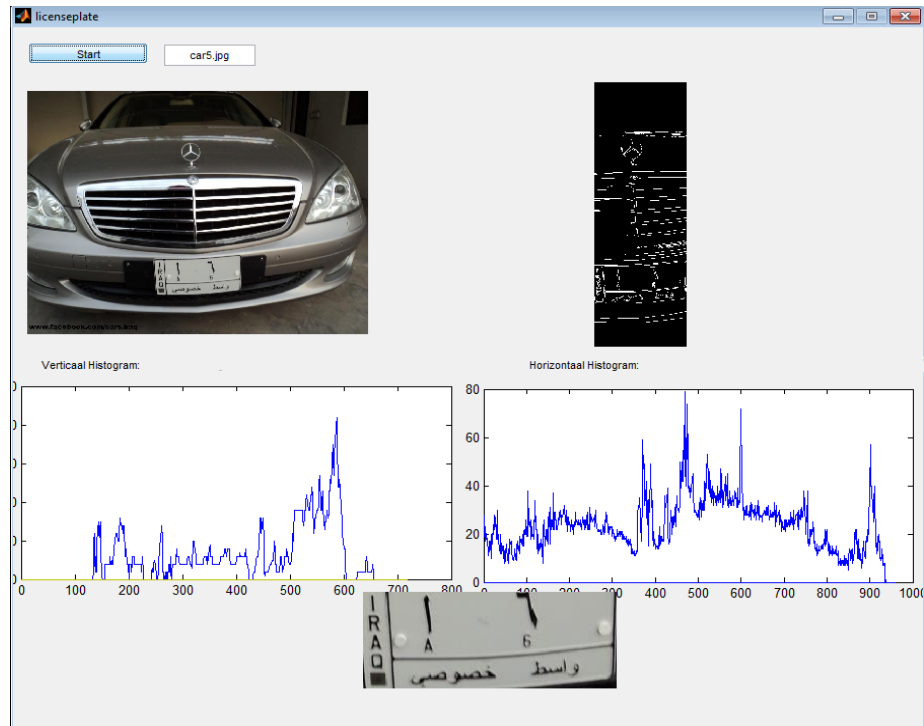


Figure 69 Iraqi Plate Test 23.

2- For other countries car plates we have successfully detected 25 plates they are European car plates the algorithm almost detected car plates with masking parts till founding the plates. Examples for detection of the plate regions tested on different countries of Europe plate images are shown in figure (70-88):

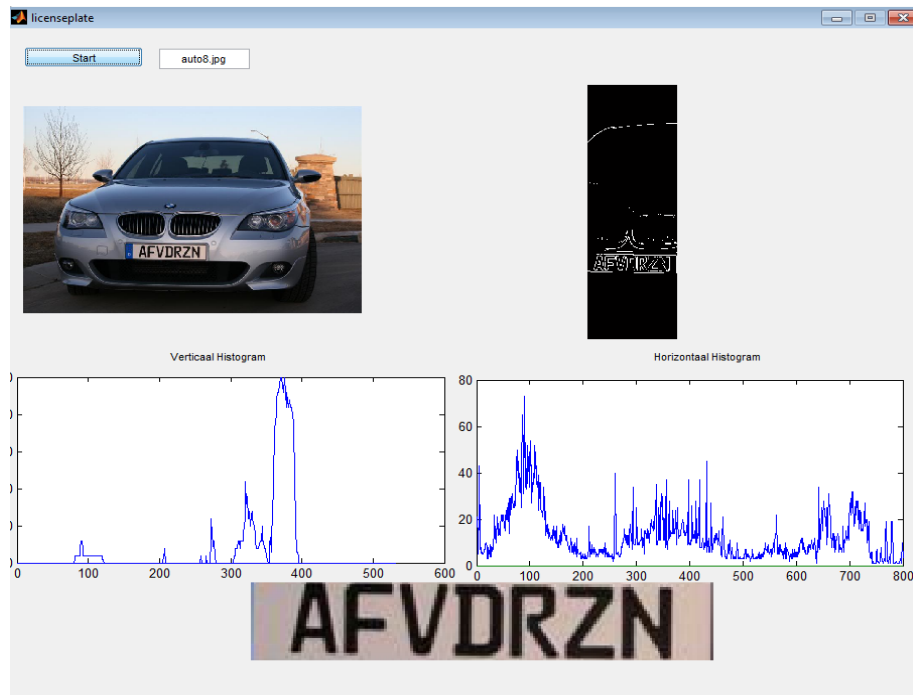


Figure 70 Europe Plate Test 1.

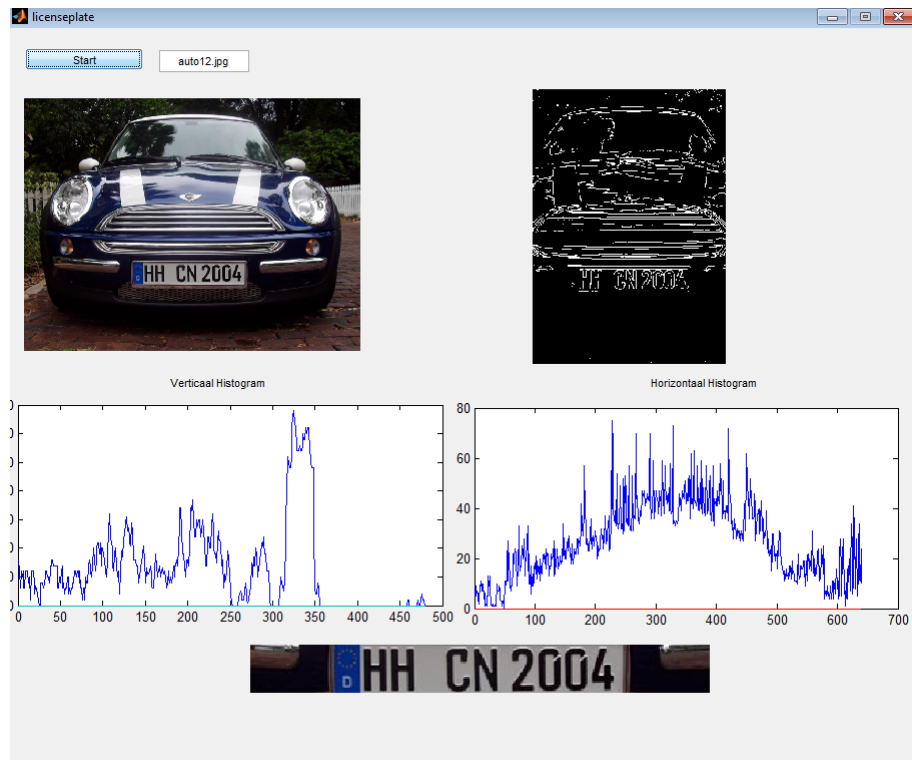


Figure 71 Europe Plate Test 2.

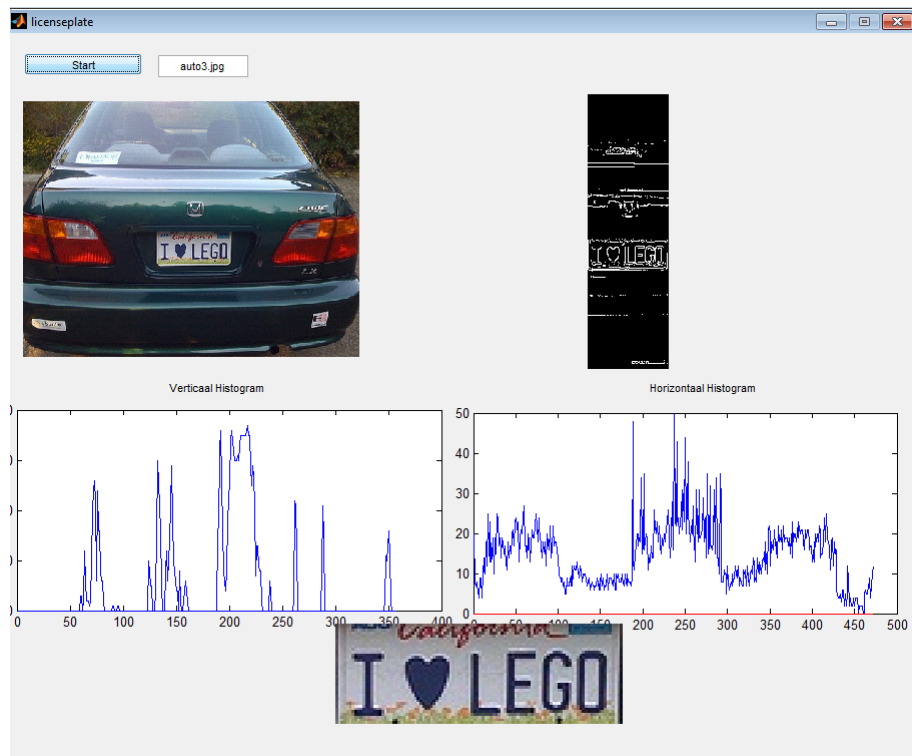


Figure 72 Europe Plate Test 3.

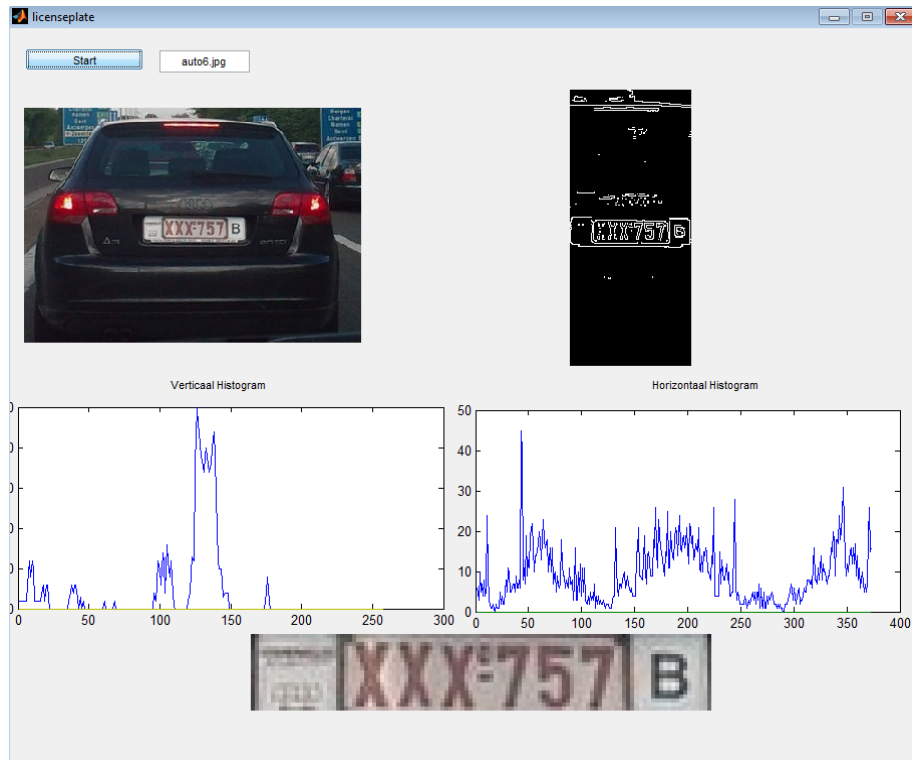


Figure 73 Europe Plate Test 4.

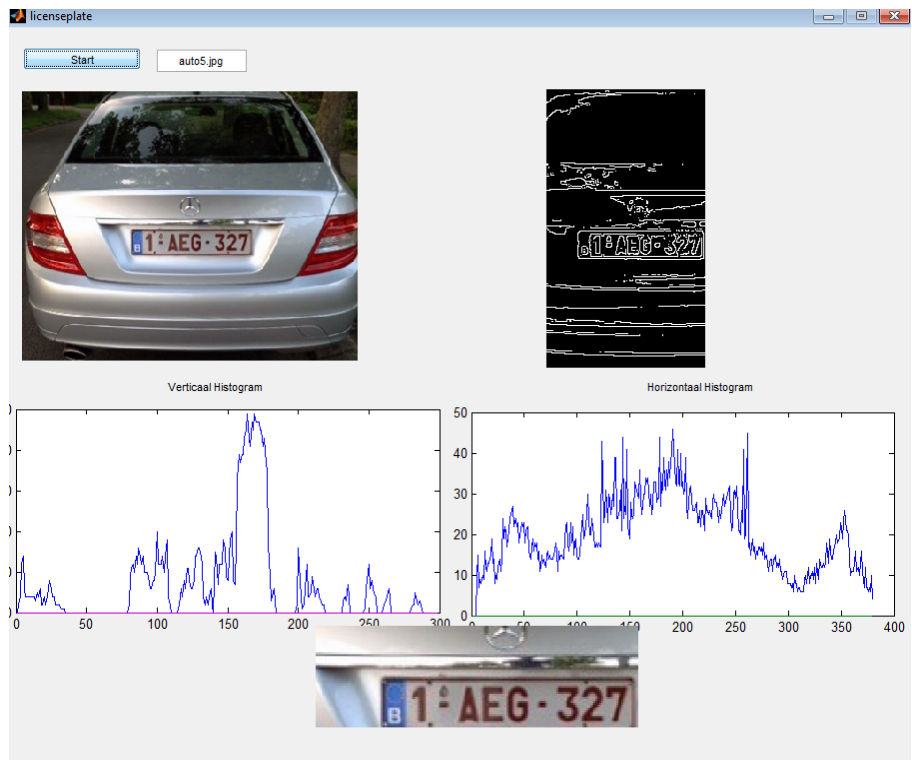


Figure 74 Europe Plate Test 5.

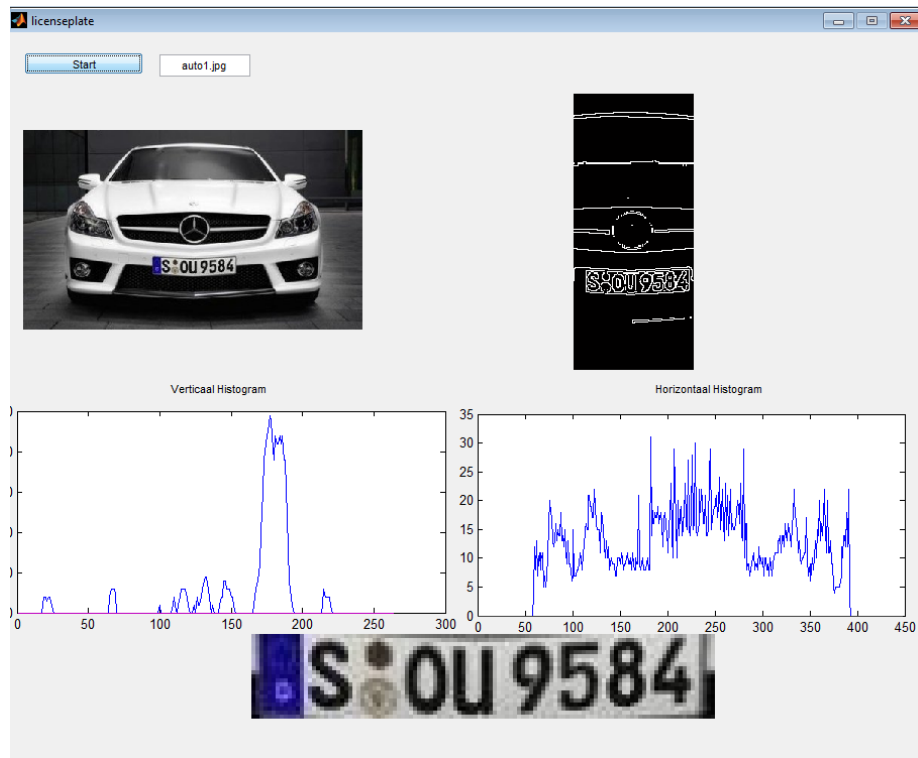


Figure 75 Europe Plate Test 6.

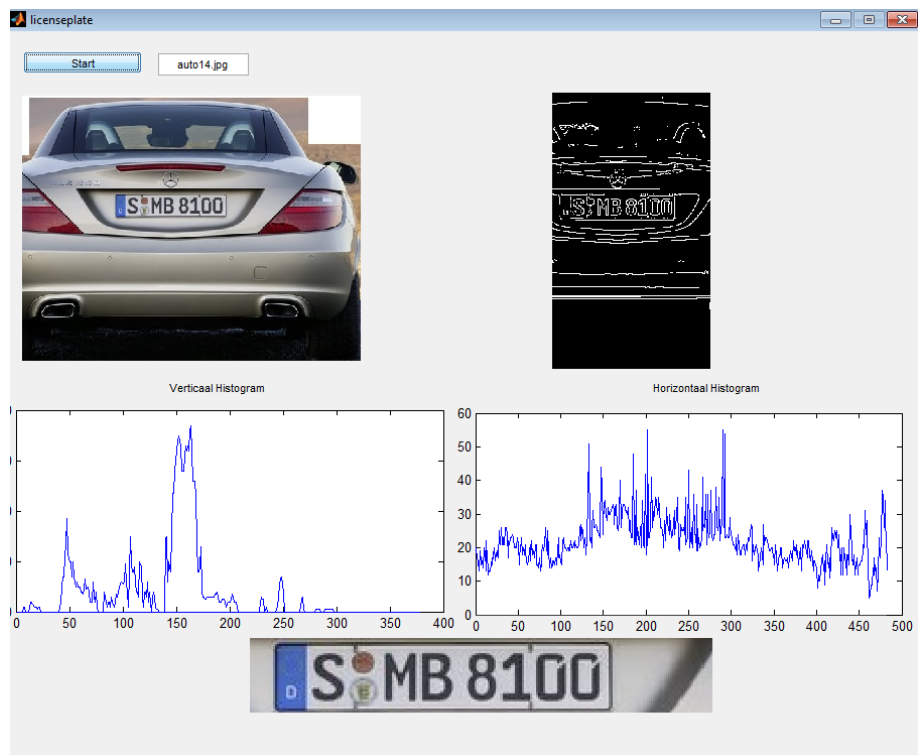


Figure 76 Europe Plate Test 7.

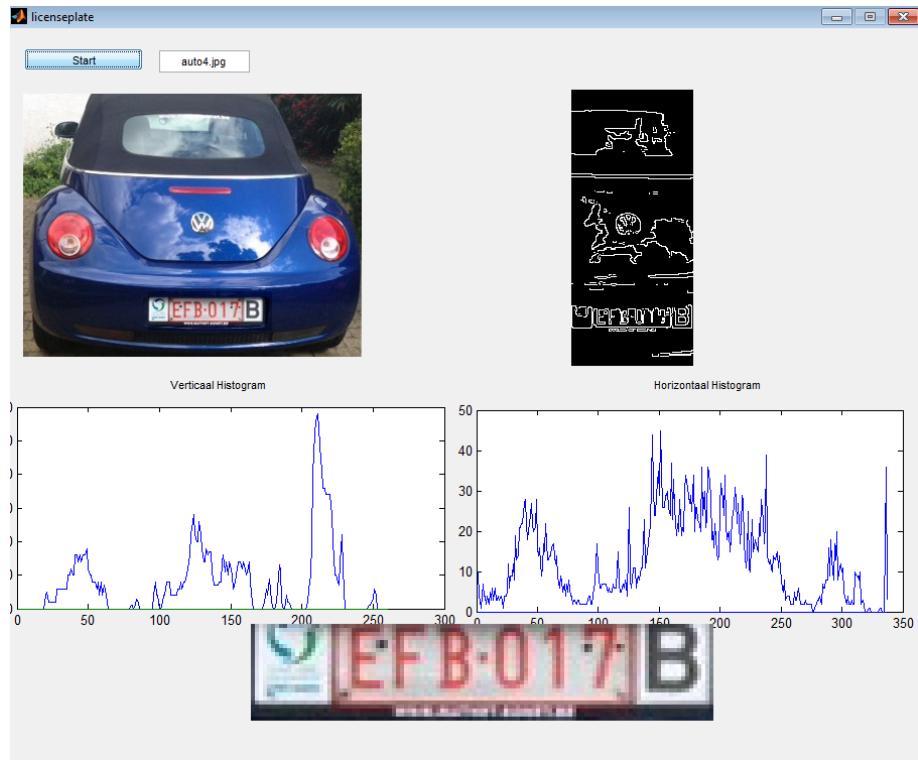


Figure 77 Europe Plate Test 8.

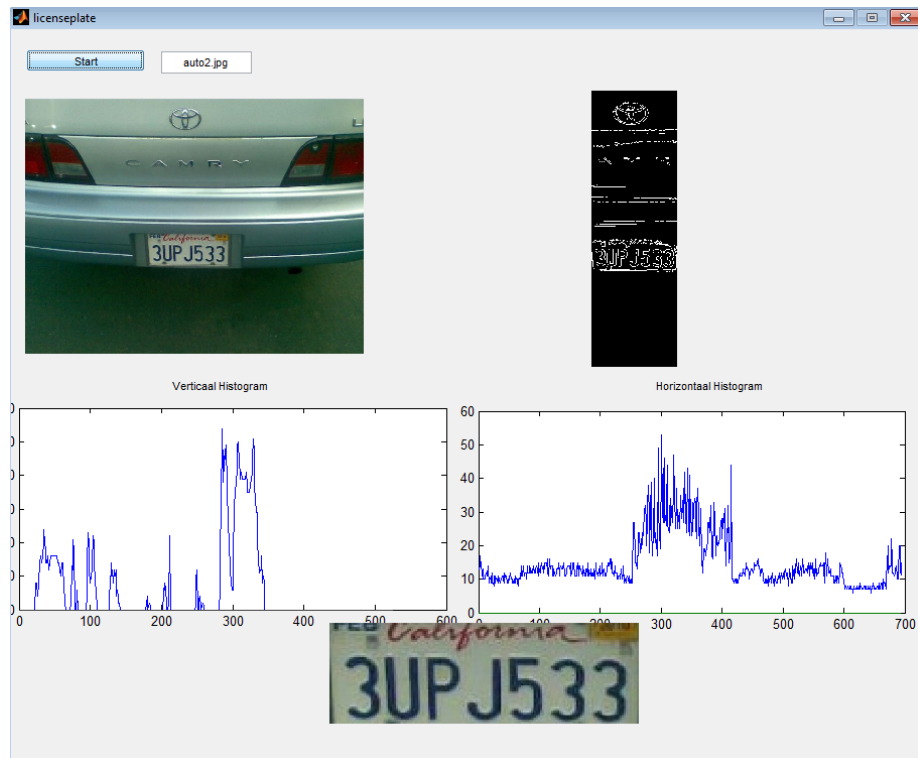


Figure 78 Europe Plate Test 9.

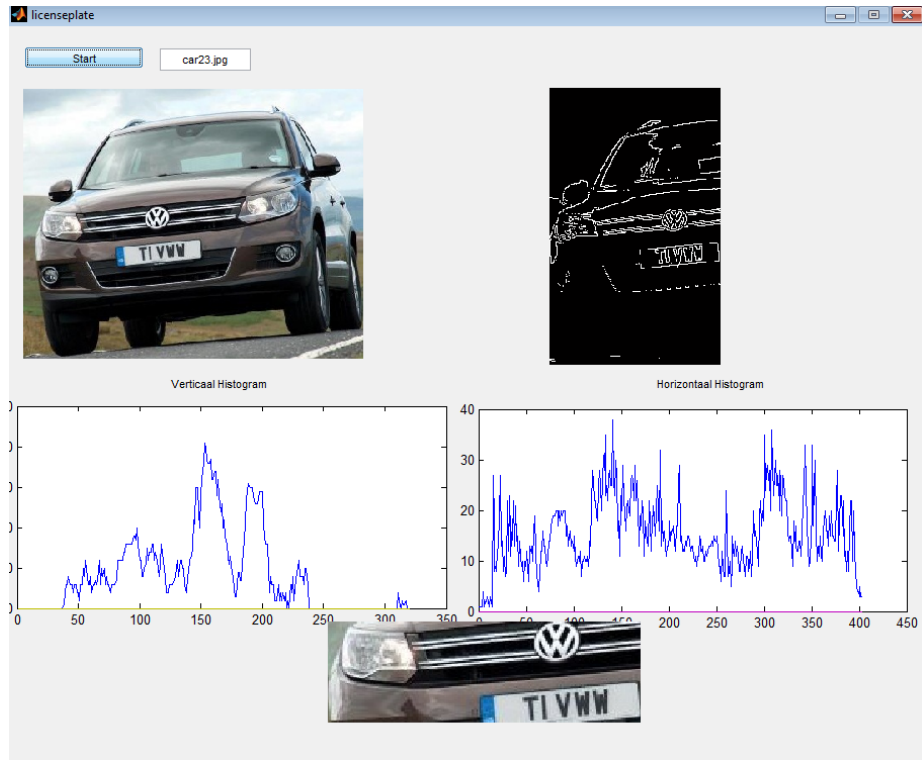


Figure 79 Europe Plate Test 10.



Figure 80 Europe Plate Test 11

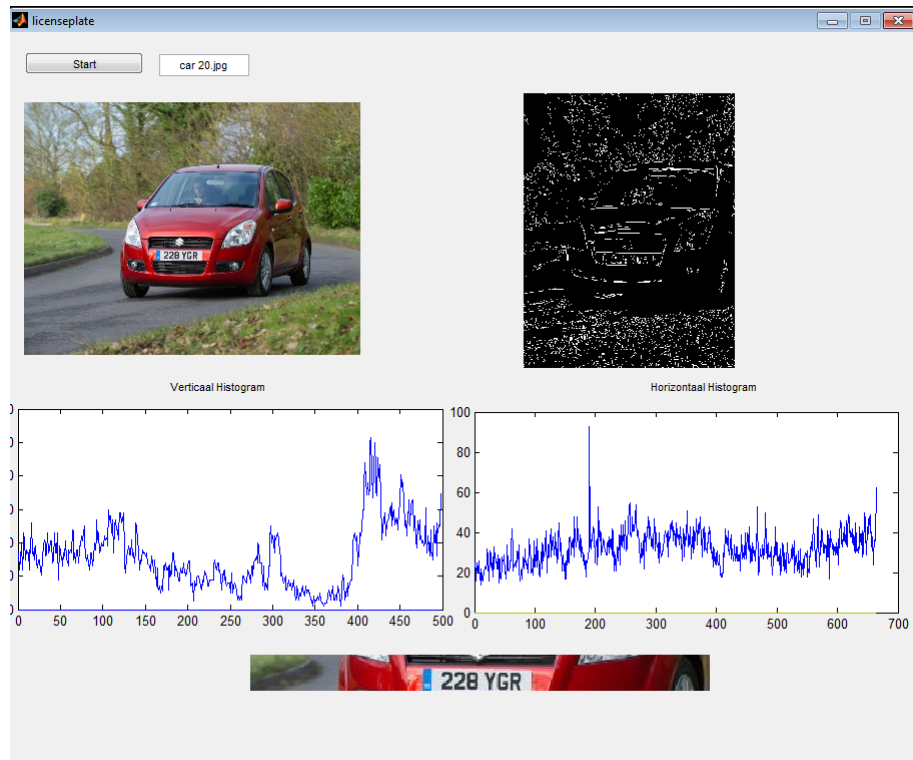


Figure 81 Europe Plate Test 12.

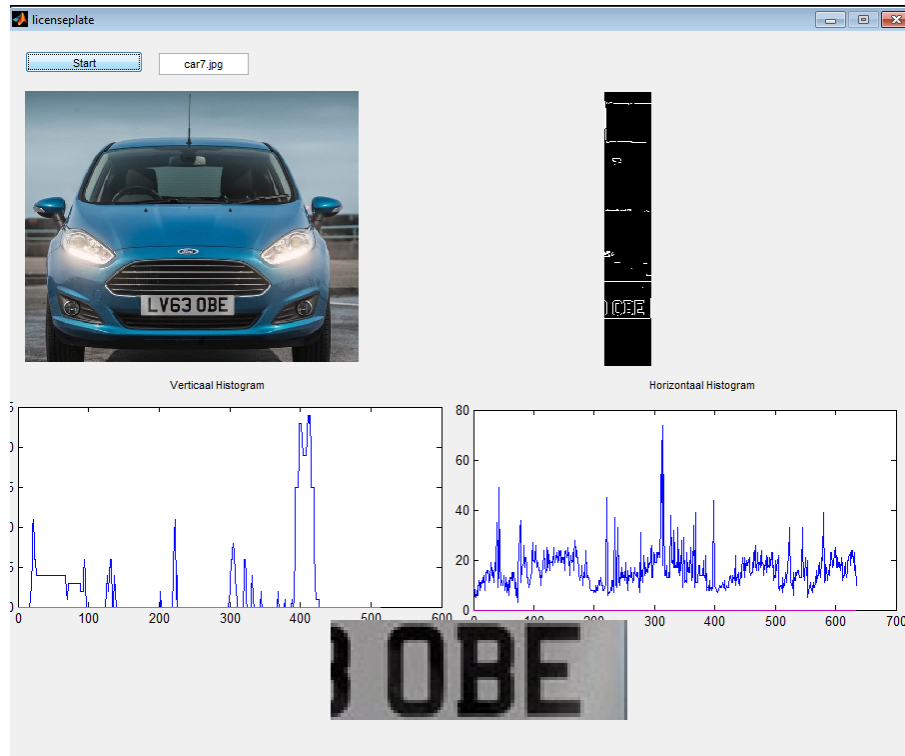


Figure 82 Europe Plate Test 13.

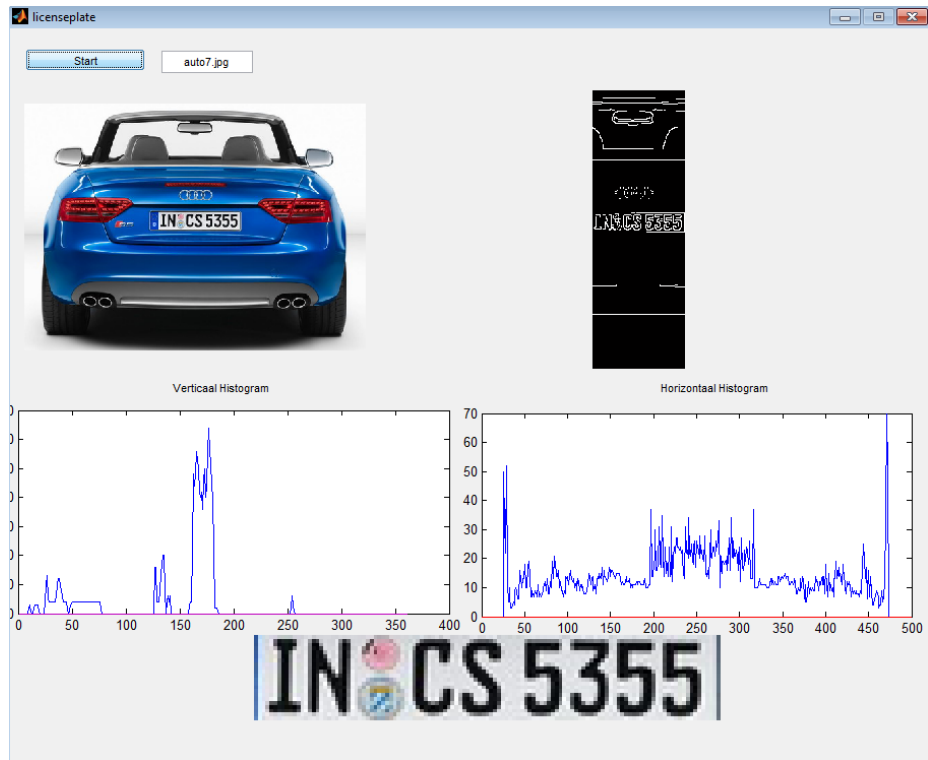


Figure 83 Europe Plate Test 14.

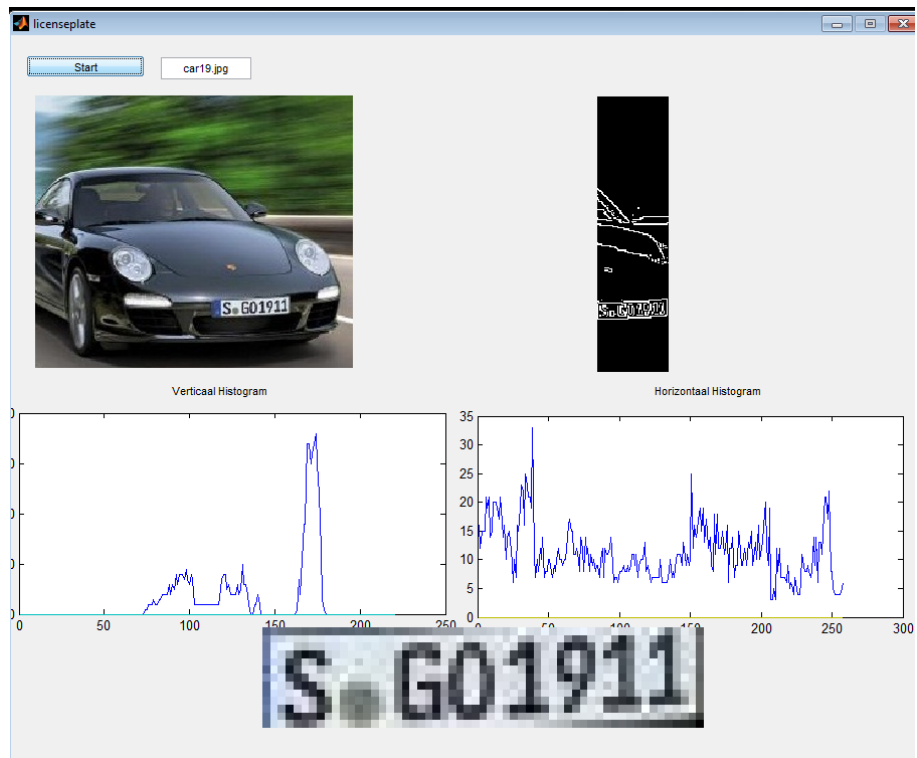


Figure 84 Europe Plate Test 15.

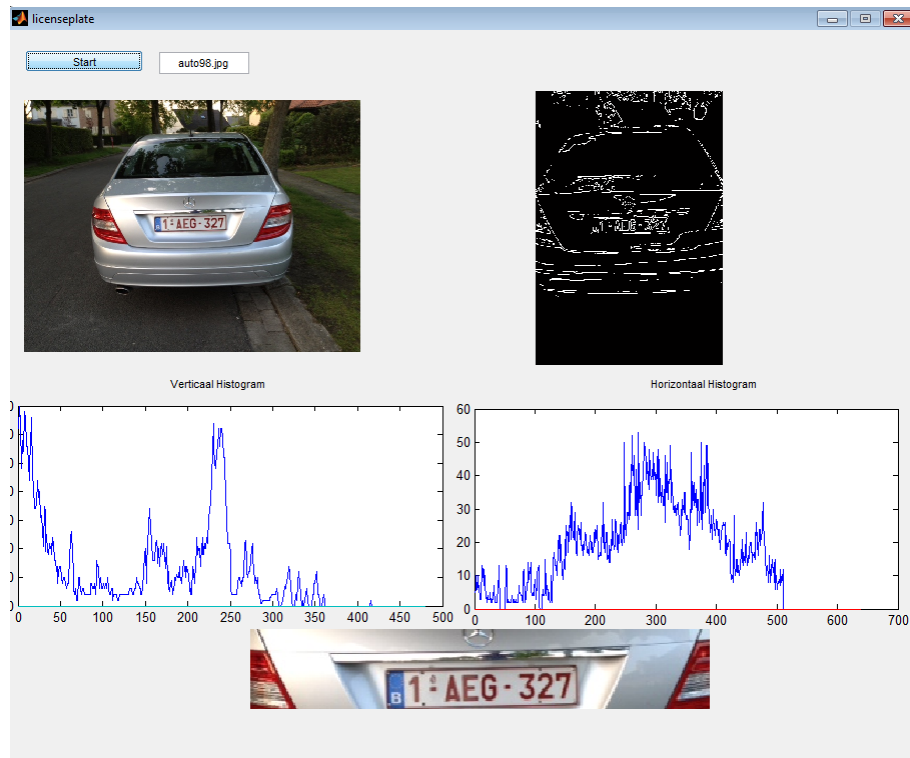


Figure 85 Europe Plate Test 16.

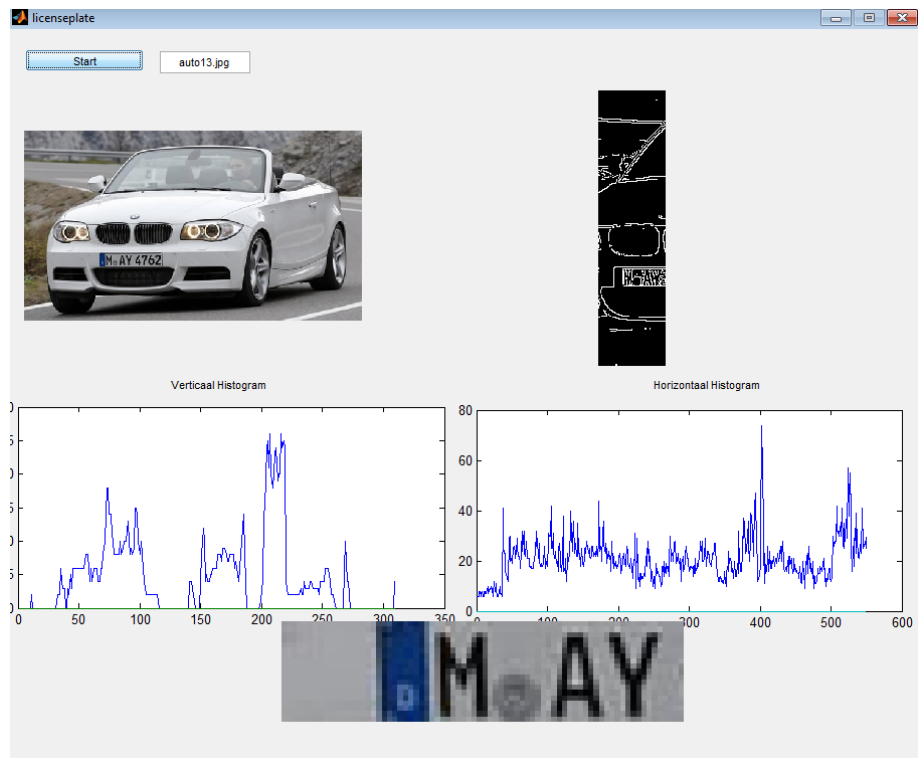


Figure 86 Europe Plate Test 17.

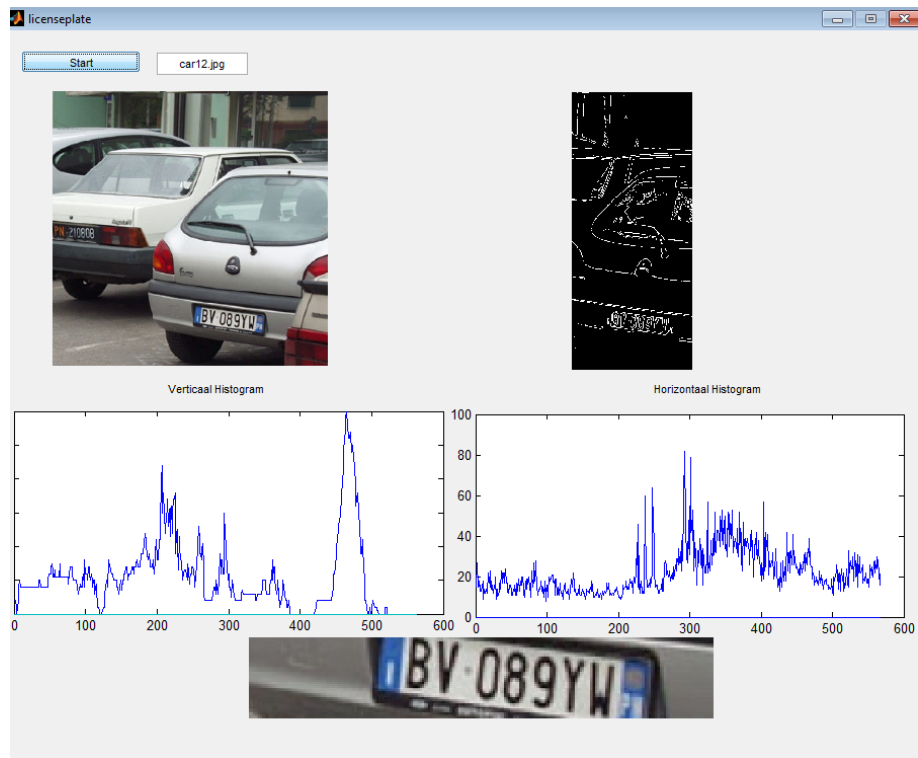


Figure 87 Europe Plate Test 18.

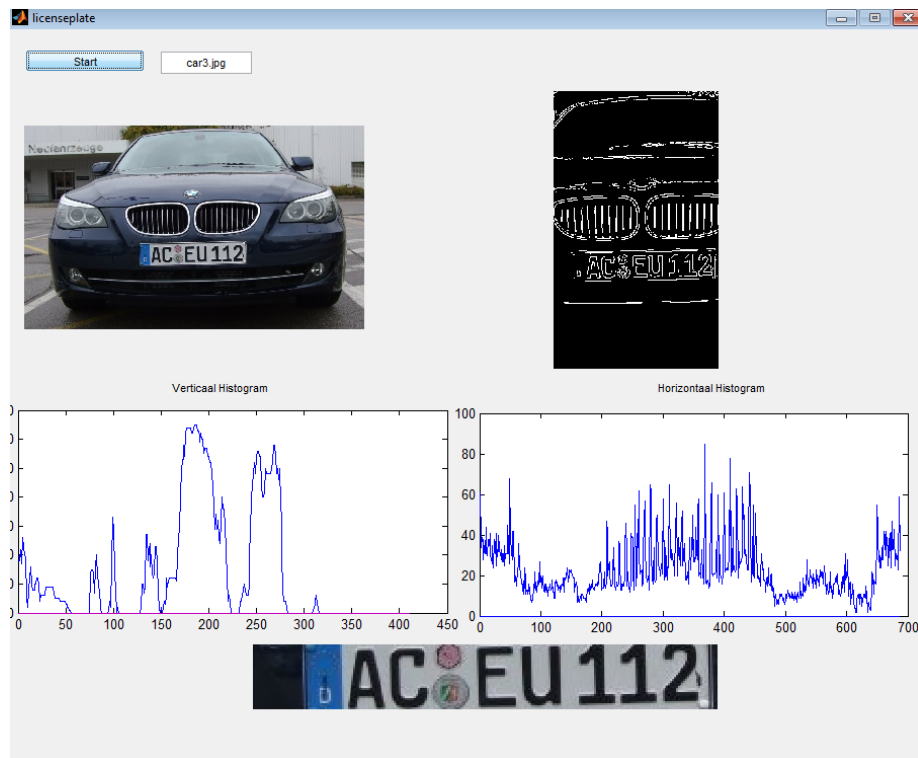


Figure 88 Europe Plate Test 19.

3- Problem and Effects on the environment:

The Edge Detection method used brings several problems with it. Edges of the environment outside the car also displayed therefore the entire algorithm no longer works. Often an example is shown in Figuur 89. and in Figure 90, below in Figure 89. we use image of Mercedes car, and around the Mercedes are distinctly different environmental factors visible. This has as a result that too many edges of the environment can be displayed in order. The edge detection algorithm in addition to the edges of the car in the histogram it is very difficult to adjust the position of the license plate can be found, and the result of the detection of the license plate in this case is therefore not good. In Figuur 90, however, we use the same photo but without the environmental factors, the result of the detection in this case, as well as perfect.

Solution

The solution should not be performed in this case in the program, but rather in the application of the program. When applying an LPD license plate detection system the camera can be positioned. Few photos of certain processed lane the angle of the camera can also ensure that the horizon is not visible. This will have as a result that the only environmental factor next to the car itself, the lane will be. And that lane will be completely filtered out in most cases by the Edge-detection algorithm.

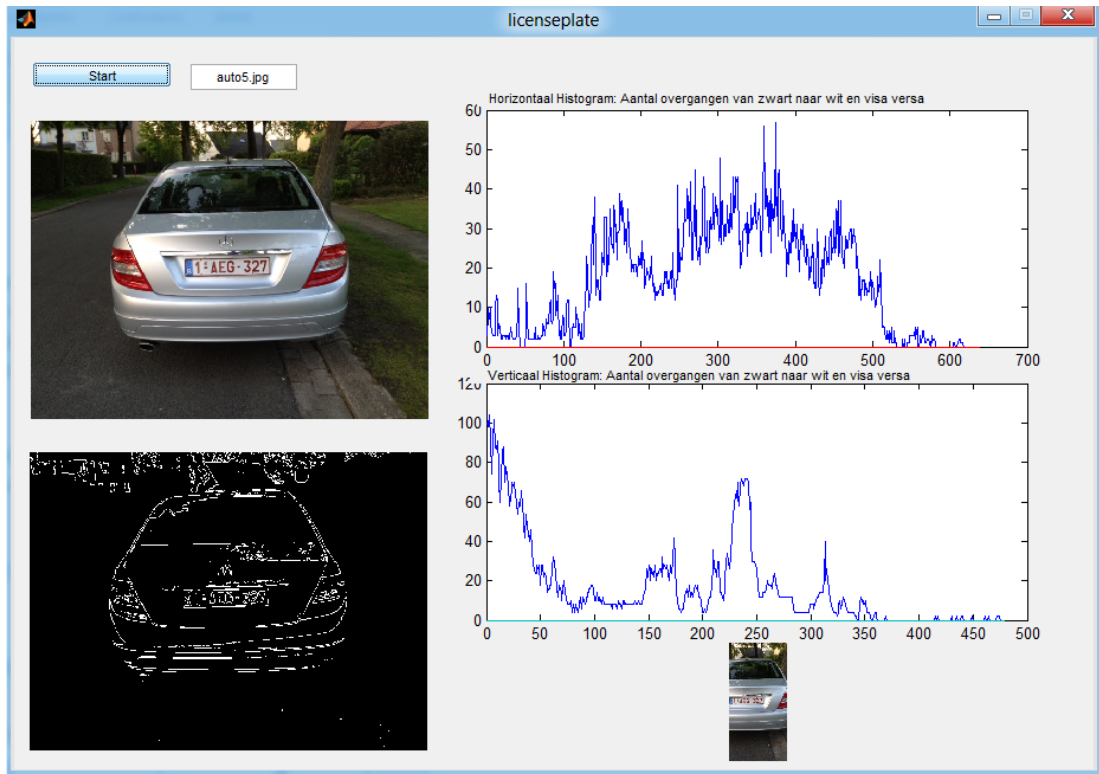


Figure 89 Input Image with the Environment Effects.

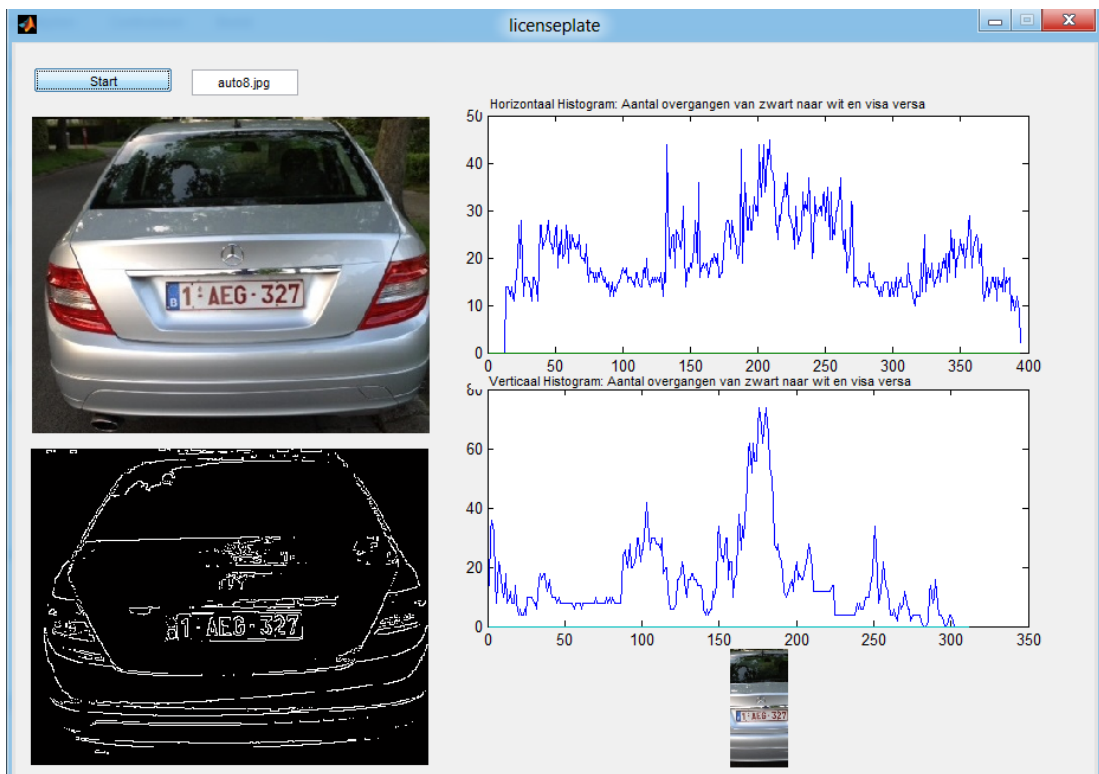


Figure 90 Input Image without Environment Effects.

CHAPTER 5

CONCLUSION AND FUTURE WORKS

5.1. Conclusion

In this thesis, we have introduced an algorithm for Iraqi license number plate detection. The test images were taken under various illuminations, size and types of license plate conditions and different countries. The experimental results for the car plate detection system were performed under different illuminations, different distances and various types of license plate. The program converts input images into gray scale because the pixels intensity consists of a given range, these ranges are 1 and 0. We used binary images and we increased the threshold to get 30% for better results. We used Robert Edge Detection method here for detecting edges. We have to make horizontal and vertical edge processing, while we have been setting the image to gray scale, taking into consideration the difference in gray value between successive pixels what exceed a certain limit. Then we added to the total of that particular column. Finally, all column totals have been stored in a matrix; we have to plot this matrix. From this the position of the plate off it can be derived, in many cases and that the plate is to be the highlighted region.

For vertical edge processing, we do not just look at the number of white pixels per column, but the number of times a pixel relative to its neighbor in a row when the rows of a license plate include several adjacent pixels.

. In addition, there are many parts in one image with many high histograms of edges and it is not plate, by checking horizontal and vertical histogram of the output image, the contrast of the output image in gray scale if it is close to the certain threshold means our program is detected successfully. If it is not the algorithm multiply final region to zero matrixes then replay the program till finds car plate.

In our algorithm we have checked some samples of car plate and we have proved that all contrast of the car plats all have high contrast histogram in white phase so we add a threshold to check our output. Further, we have tested 50 car plate in our algorithm

25 of them are Iraqi car plates and 25 of them are plates of different countries also we have some error rate because of environment effect.

5.2. Future Work

There is room for the algorithm to be optimized and it can be further improved by considering more parameters and introducing pre-processing and post processing steps that can be applied to increase the accuracy of detection. The scope of the project can be extended to consider independent orientation.

REFERENCES

1. **Bamler G., Drane T., Schomburgbamler P., (2010)**, “*Number Plates Worldwide: Car Registration Plates, Codes and Many Facts About Road Traffic in All Countries*”, ISBN-10: 3981386221.
2. **Han H., Han R., (2011)**, “*Method of License Plate Location Based on Edge Detection and Color Information*”, IEEE, 978-1-4577-1701-7/11, 2011, pp1477-1480.
3. **Lakshmi J., Rani J., Ramakrishna S., Lantikiran M., Siddhartha V., (2011)**, “*A Novel Approach for Indian License Plate Recognition System*”, IJAEST, vol 2 Issue I, 2011, pp 010-014.
4. **Gendy S., Clifton L., Lachowicz S., (1997)**, “*Automatic Car Registration Plate Using Fast Hough Transform*”, IEEE, 0-7803-3913-4-9/97,1997, PP 209-218.
5. **Anishiya P., Joans A.S., (2011)**, “*Number Plate Recognition for Indian Cars Using Morphological Dilation and Erosion with the Aid Of OCRs*”, IACSIT, vol.4, 2011, pp. 115-119.
6. **Pandya P., Singh M., (2011)**, “*Morphology Based Approach to Recognize Number Plates in India*”, IJSCE, vol.1, issue 3, 2011, pp107-112.
7. **Sridevi T., Sindhu C., Kumar P., Sagar P., (2012)**, “*Morphology Based Number Plate Localization for Vehicular Surveillance System*”, IJMER, vol.2, ISSUE.2, 2012, pp 334-337.
8. **Mohsin A., Hassin A., AbdulJaleel I., (2010)**, “*An Automatic Recognizer for Iraqi License Plates Using ELMAN Neural Network*”, Doi:10.4236/Jsea.2010.312136.
9. **Jiang D., Mekonnen T., Embiale T., Gebrehiwot A., (2012)**, “*Car Plate Recognition System in Fifth International Conference on Intelligent Networks and Intelligent Systems*”, IEEE, 978-0-7695-4855-5/12 2012.
10. **Seyad H., Seyad M., (2013)**, “*Extraction and Recognition of the Vehicle License Plate for Passing Under Outside Environment*”, IEEE, pp. 234-237.
11. **Zhigang X., Honglei Z., (2007)**, “*An Efficient Method of Locating Vehicle License Plate, 3rd International Conference on Natural Computation*”, IEEE, 0-7695-2875-9/07, 2007.

12. **Peng H., Long F., Zheru C., (2003)**, “*Document Image Recognition Based on Template Matching of Component Block Projection*”, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol 25, No9 September 2003.
13. **Ian T.Y., Jan J.G., Lucas J., Van V., (1998)**, “*Fundamentals of Image Processing*”, ISBN 90–75691–01–7, NUGI 841, Version 2.2, by I.T. Young, J.J. Gerbrands and L.J. Van Vliet.
14. **Russ J.C., (1994)**, “*The Image Processing Handbook, Second Edition, CRC Press, Boca Raton*”, ISBN 0-8493-2516-1.
15. **Kristian S., (2011)**, “*Introduction to Image Processing in MATLAB*” Department of Applied Mathematics, University of Colorado at Boulder <http://webpage.pace.edu/kt52695n/improcessing/matlabimpr.html> (Data Download Date: 25.6.2013).
16. **Pratt W., (2001)**, “*Digital Image Processing*” Copyright © 2001 John Wiley & Sons, Inc, ISBNs: 0-471-37407-5 (Hardback); 0-471-22132-5.
17. **Suri P.K., Walia E., Verma A., (2010)**, “*Vehicle Number Plate Detection Using Sobel Edge Detection Technique*”, I S S N : 2 2 2 9 - 4 3 3 3 | I S S N : 0 9 7 6 - 8 4 9 1 (On l i n e) IJCST Vol. 1, Issue 2, December 2010.
18. **Kalloniatis M., Luu C., (2007)**, “*The Organization of the Retina and Visual System*”, Created: May 1, 2005; Last Update: July 9, 2007. Copyright Notice Bookshelf ID: NBK11538PMID: 21413396.
19. **Gomes J., (2009)**, “*Image Processing for Computer Graphics and Vision*”, Springer-Verlag, ISBN: 978-1-84800-192-3.
20. **Sattar F., Floreby L., Salomonsson G., Lovstrom B., (1997)**, “*Image Enhancement Based on a Nonlinear Multiscale Method*”, IEEE trans, vol 6 Issue 6, pp 888-895.
21. **Kaur M., Kaur J., (2011)**, “*Survey of Contrast Enhancement Techniques Based on Histogram Equalization*”, (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 2, No. 7, 2011.
22. **Schiele B., Crowley J., (1996)**, “*Object Recognition Using Multidimensional Receptive Field Histograms*”, ECCV’96, Springer LNCS 1064, I:610–619.
23. **Torre V., Poggio T., (1986)**, “*Edge Detection*”, IEEE Trans. Pattern Anal. Machine Intell, vol. PAMI-8, no. 2, pp. 187-163.
24. **Maini R., Aggarwal H., (2014)**, “*Study and Comparison of Various Image Edge Detection Techniques*” International Journal of Image Processing (IJIP), Volume (3) : Issue (1).

25. **Narendra V., Hareesh K., (2011)**, “*Study and Comparison of Various Image Edge Detection Techniques Used in Quality Inspection and Evaluation of Agricultural and Food Products by Computer Vision*” Int J Agric & Biol Eng Open Access at <http://www.ijabe.org> Vol. 4 No.2 83. DOI: 10.3965/j.issn.1934-6344.2011.02.083-090.
26. **Abel J., (2011)**, “*Edge Detectors*” Copyright All Rights Reserved. Based on: www.uweb.ucsb.edu/~shahnam/AfED.doc. (Data Download Date: 15.10.2013).
27. **Matthews J., (2002)**, “*An Introduction to Edge Detection: The Sobel Edge Detector*”, Available at <http://www.generation5.org/content/2002/im01.asp>. (Data Download Date: 18.10.2013).
28. **Roberts L., (1965)**, “*Machine Perception of 3-D Solids Ser. Optical and Electro-Optical Information Processing.MITP ress*”. International Journal of Image Processing (IJIP), Volume (3) : Issue (1).
29. **Roborealm, (2014)**, “*Roberts Edge*”,All Rights Reserved Based on: http://www.roborealm.com/help/Roberts_Edge.php (Data Download Date: 20.11.2013).
30. **Juneja M., Sandhu P., (2009)**, “*Performance Evaluation of Edge Detection Techniques for Images in Spatial Domain*” International Journal of Computer Theory and Engineering, Vol. 1, No. 51793-8201.
31. **Fisher R., Perkins S., Walker A., Wolfart E., (2003)**, All Rights Reserved Based on: <http://homepages.inf.ed.ac.uk/rbf/HIPR2/log.htm> (Data Download Date: 25.11.2013).
32. **Gonzalez R., Woods R., (2002)**, “*Digital Image Processing*”, 2nd ed. Prentice Hall.
33. **Torre V., TPoggio V., (1986)**, “*Edge Detection*”. IEEE Trans. Pattern Anal. Machine Intell, vol. PAMI-8, no. 2, pp. 187-163.
34. **Kirsch R., (1971)**, “*Computer Determination of the Constituent Structure of Biomedical Images*”. Comput. Eiorned. Res., vol. 4, pp. 315-328.

APPENDIX A
CURRICULUM VITAE

PERSONAL INFORMATION

Surname, name: SIDDIK, Othman

Nationality: Iraq

Date and place of birth: 17/09/1989, IRAQ / KIRKUK

Marital state: Single

Phone: +905340208431, +9647701002858

Email: Osman_subhi@yahoo.com, c1174555@student.cankaya.edu.tr

Year of Graduation	Eeducation Degree	Institution
2011	Cihan University Collage	BSC
2006	Kirkuk	High School

FOREIGN LANGUAGES

Turkish, Arabic, Kurdish, English

HOBBIES

Image Processing, Mathematic, Cars and Football

APPENDIX B

Proposed Work in MATLAB

```
function varargout = licenseplate(varargin)

gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @licenseplate_OpeningFcn, ...
                  'gui_OutputFcn',  @licenseplate_OutputFcn, ...
                  'gui_LayoutFcn',  [] , ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

function licenseplate_OpeningFcn(hObject, eventdata, handles,
varargin)

handles.output = hObject;

guidata(hObject, handles);

function varargout = licenseplate_OutputFcn(hObject, eventdata,
handles)

varargout{1} = handles.output;

function StartKnop_Callback(hObject, eventdata, handles)

naam=get(handles.naam, 'String');
A=imread(strcat('images/',naam));

org=A;
axes(handles.axes1);
[h,w,f]=size(A);
%original image is shown
imshow(A);
%we convert the image to grayscale to a treshhold

A = rgb2gray(A);

level = graythresh(A);
%the image is converted to a binary image, we increase the
%threshold still get 30% for better results
A = im2bw(A,level*1.3);
```

```

axes(handles.axes2);
bb=A
save bb
%Then we use a built-in matlab function for
%edge-detection
A=edge(A,'roberts');
imshow(A);

horHist=zeros(w);
%The number of white pixels in each column are added and stored
% تضاف? عدد بكسل أبيض في كل عمود وتخزينها

for i=1:w
    tot=0;
    for j=1:h
        if (A(j,i)==1)
            tot=tot+1;
        end
    end
    horHist(i)=tot;
end
axes(handles.axes3);
%berekende treshold
gem=max(horHist)/2.3;
plot(horHist);
hstart=0;
heinde=0;
width=0;
hcounter=0;
arc=0;
hcoor=zeros(1,2);

for i=1:w
    if horHist(i)>gem(1)
        if(hstart==0)
            hstart=i;
        end
        hcounter=0;
    else
        if hstart>0
            if hcounter>(w*0.07)
                heinde=i-hcounter;
                width=heinde-hstart;
                if(width>(w*0.1))
                    arc=arc+1;
                    hcoor(arc,1)=hstart;
                    hcoor(arc,2)=width;
                end
                hstart=0;
                hcounter=0;
                heinde=0;
                width=0;
            end
            hcounter=hcounter+1;
        end
    end
end
[ww,f]=size(hcoor);

```

```

hstart=0;
hwidth=0;

for i=1:ww
    if(hcoor(i,2)>hwidth)
        hwidth=hcoor(i,2);
        hstart=hcoor(i,1);
    end
end

A=A(:,hstart:(hstart+hwidth),:);
axes(handles.axes2);
imshow(A);
ss=hstart
dd=hwidth
verHist=zeros(h);

for j=1:h
    tot=0;
    for i=2:hwidth
        if (A(j,i-1)==1 && A(j,i)==0) || (A(j,i-1)==0 && A(j,i)==1)
            tot=tot+1;
        end
    end
    verHist(j)=tot;
end
axes(handles.axes4);
verh=zeros(1);
coun=1;

for i=1:h
    if(verHist(i)>0)
        verh(coun)=verHist(i);
        coun=coun+1;
    end
end
gem=mean(verh)
plot(verHist);
vstart=0;
veinde=0;
height=0;
vcounter=0;
arc=0;
vcoor=zeros(1,2);
h*0.07

for(i=1:h)
    if verHist(i)>gem(1)
        if(vstart==0)
            vstart=i;
        end
        vcounter=0;
    else
        if vstart>0
            if vcounter>(h*0.03)
                veinde=i-vcounter;
                height=veinde-vstart;
                if(height>(h*0.05))
                    arc=arc+1;
                end
            end
        end
    end
end

```

```

                vcoor(arc,1)=vstart;
                vcoor(arc,2)=height;
            end
            vstart=0;
            vcounter=0;
            veinde=0;
            height=0;
        end
        vcounter=vcounter+1;
    end
end
end
[1,f]=size(vcoor);
axes(handles.axes5);

A=org(vcoor(1,1):vcoor(1,1)+vcoor(1,2),hstart:(hstart+hwidth),:);
imshow(A);
xx=vcoor(1,1)
yy=vcoor(1,2)
oa=rgb2gray(org)
bbb=0
www=0
%oas=size(oa)
%oax=oas(1)
%oay=oay(2)
for x=xx:xx+yy
    for y=ss:ss+dd
        if oa(xx,yy)<=150
            bbb=bbb+1;
        end
        if oa(xx,yy)>=150
            www=www+1;
        end
    end
end

end
if www>bbb
    white='this is car plate'
else
    black='this is not a car plate'
    if size(vcoor)==[1,2]
        xx=vcoor(1,1)
        yy=vcoor(1,2)
        for x=xx:xx+yy
            for y=ss:ss+dd
                oa(x,y)=0;
            end
        end
    elseif size(vcoor)==[2,2]
        xx=vcoor(2,1)
        yy=vcoor(2,2)
        for x=xx:xx+yy
            for y=ss:ss+dd
                oa(x,y)=0;
            end
        end
    end
end
end
end

```

```

end
imwrite(oa, 'oa.jpg')

save

%-----
-----
-----
    if bbb>www
for p=1:5

A=imread('oa.jpg');

A = im2bw(A, level*1.3);

A=edge(A, 'roberts');
imshow(A);

horHist=zeros(w);

for i=1:w
    tot=0;
    for j=1:h
        if (A(j,i)==1)
            tot=tot+1;
        end
    end
    horHist(i)=tot;
end
%axes(handles.axes3);
%berekende treshold
gem=max(horHist)/2.3;
plot(horHist);
hstart=0;
heinde=0;
width=0;
hcounter=0;
arc=0;
hcoor=zeros(1,2);

for i=1:w
    if horHist(i)>gem(1)
        if(hstart==0)
            hstart=i;
        end
        hcounter=0;
    else
        if hstart>0
            if hcounter>(w*0.07)
                heinde=i-hcounter;
                width=heinde-hstart;
                if(width>(w*0.1))
                    arc=arc+1;
                end
            end
        end
    end
end

```



```

                hcoor(arc,1)=hstart;
                hcoor(arc,2)=width;
            end
            hstart=0;
            hcounter=0;
            heinde=0;
            width=0;
        end
        hcounter=hcounter+1;
    end
end
end
[ww,f]=size(hcoor);
hstart=0;
hwidth=0;

for i=1:ww
    if(hcoor(i,2)>hwidth)
        hwidth=hcoor(i,2);
        hstart=hcoor(i,1);
    end
end

A=A(:,hstart:(hstart+hwidth),:);
%axes(handles.axes2);
imshow(A);
ss=hstart
dd=hwidth
verHist=zeros(h);

for j=1:h
    tot=0;
    for i=2:hwidth
        if (A(j,i-1)==1 && A(j,i)==0) || (A(j,i-1)==0 && A(j,i)==1)
            tot=tot+1;
        end
    end
    verHist(j)=tot;
end
%axes(handles.axes4);
verh=zeros(1);
coun=1;

for i=1:h
    if(verHist(i)>0)
        verh(coun)=verHist(i);
        coun=coun+1;
    end
end
gem=mean(verh)
plot(verHist);
vstart=0;
veinde=0;
height=0;
vcounter=0;
arc=0;
vcoor=zeros(1,2);
h*0.07

```

```

for(i=1:h)
    if verHist(i)>gem(1)
        if(vstart==0)
            vstart=i;
        end
        vcounter=0;
    else
        if vstart>0
            if vcounter>(h*0.03)
                veinde=i-vcounter;
                height=veinde-vstart;
                if(height>(h*0.05))
                    arc=arc+1;
                    vcoor(arc,1)=vstart;
                    vcoor(arc,2)=height;
                end
                vstart=0;
                vcounter=0;
                veinde=0;
                height=0;
            end
            vcounter=vcounter+1;
        end
    end
end
[1,f]=size(vcoor);
%axes(handles.axes5);

A=org(vcoor(1,1):vcoor(1,1)+vcoor(1,2),hstart:(hstart+hwidth),:);
imshow(A);
%axes(handles.axes6);

%-----
%-----%

oa1=rgb2gray(org)

xx=vcoor(1,1)
yy=vcoor(1,2)
oa1=rgb2gray(org)
bbb=0
www=0
for x=xx:xx+yy
    for y=ss:ss+dd
        if oa1(xx,yy)<=150
            bbb=bbb+1;
        end
        if oa1(xx,yy)>=150
            www=www+1;
        end
    end
end

end
if www>bbb

```

```

        white='this is car plate'
        else
            %black='this is not a car plate'
            blk=1
        end
        if bbb>www
            black='this is not a car plate'
            if size(vcoor)==[1,2]
                xx=vcoor(1,1)
                yy=vcoor(1,2)
                for x=xx:xx+yy
                    for y=ss:ss+dd
                        oa(x,y)=0;
                    end
                end
            elseif size(vcoor)==[2,2]
                xx=vcoor(2,1)
                yy=vcoor(2,2)
                for x=xx:xx+yy
                    for y=ss:ss+dd
                        oa1(x,y)=0;
                    end
                end
            end
            end
            if size(vcoor)==[1,2]
                xx=vcoor(1,1)
                yy=vcoor(1,2)
                for x=xx:xx+yy
                    for y=ss:ss+dd
                        oa1(x,y)=0;
                    end
                end
            end
            end
            else
                white='this is car plate'
            end
        end
    end
end

% size(vcoor)==[4,2]
%xx=vcoor(4,1)
%yy=vcoor(4,2)
%for x=xx:xx+yy
%    for y=ss:ss+dd
%        oa(x,y)=0;
%    end
%end

imwrite(oa,'oa.jpg');

end
end
%-----

```

```
function naam_Callback(hObject, eventdata, handles)
```

```
function pushbutton2_Callback(hObject, eventdata, handles)
```

```
function ot_Callback(hObject, eventdata, handles)
```

```
function ot_CreateFcn(hObject, eventdata, handles)
```

```
if ispc && isequal(get(hObject,'BackgroundColor'),  
get(0,'defaultUicontrolBackgroundColor'))  
    set(hObject,'BackgroundColor','white');  
end
```