CAR PLATE RECOGNITION

A MASTER'S THESIS

in

Computer Engineering

Atilim University

by

KAYHAN BORA

JUNE 2009

Approval of the Graduate School of Natural and Applied Sciences, Atılım University.

Prof. Dr. Abdurrahim ÖZGENOĞLU

Director

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

Prof. Dr. İbrahim AKMAN

Head of Department

This is to certify that we have read the thesis "Car Plate Recognition" submitted by "Kayhan BORA" and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Asst. Prof. Dr. Hakan TORA   Asst. Prof. Dr. Nergiz Ercil ÇAĞILTAY

Co-Supervisor   Supervisor

Examining Committee Members

Asst. Prof. Dr. Murat KOYUNCU

Asst. Prof. Dr. Mehmet CANTÜRK

Dr. Ruşen ÖKTEM

Date: (29/06/2009)

I declare and guarantee that all data, knowledge and information in this document has been obtained, processed and presented in accordance with academic rules and ethical conduct. Based on these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name: Kayhan BORA

Signature:

ABSTRACT

CAR PLATE RECOGNITION

Bora, Kayhan

M.S., Computer Engineering Department

Supervisor: Asst. Prof. Dr. Nergiz Ercil ÇAĞILTAY

Co-Supervisor: Asst. Prof. Dr. Hakan TORA

June 2009, 46 pages

In rapidly developing countries, the number of vehicles has increased in accordance with the technology. The need to recognize the vehicles has increased in parallel to the number of the vehicles. Need of vehicle recognition was emerged for the cases of security, automatic switching systems, highway speed detection, light violation. License plate recognition system consists of three main topics: finding plate location from digital images, character segmentation from the plate images and character recognition from segmented characters. In this thesis, a study is conducted on character segmentation from plate image and character recognition from segmented characters. Character recognition work has been examined through artificial neural networks in the previously conducted studies. Instead of Artificial Neuronal Networks or complex mathematical formula, this thesis takes the features (of the character) the human eyes perceive during recognition of the character into account. Plate image was reduced to gray level, the threshold value was calculated, and the plate image was converted into the binary system. The boundary of characters in the vertical and horizontal lines were scanned after plate image was converted into the binary system. For each character, the feature classes were created for the characters scanned from left to right, right to left, top to bottom, bottom to top. The character feature classes were compared to the previously prepared feature character class's database. The rate which is similar to other characters of the character has been displayed to the user. T.C. license plate is used during working.

Keywords: Car plate recognition

ÖZ

ARABA PLAKASI TANIMA

Bora, Kayhan

Yüksek Lisans, Bilgisayar Mühendisliği Bölümü

Tez Danışmanı : Yrd. Doç. Dr. Nergiz Çağıltay

Yardımcı Tez Danışmanı : Yrd. Doç. Dr. Hakan Tora

Haziran 2009, 46 sayfa

Hızlı gelişen teknoloji ile beraber ülkelerdeki araç sayısı artmıştır. Araç sayısının artışına paralel olarak araçların tanınması gereksinimi de artmıştır. Güvenlik, otomatik geçiş sistemleri, otoyollarda hız tespiti, ışık ihlali gibi durumlarda araçların tanınması ihtiyacı doğmuştur. Araç plaka tanıma sistemi üç ana konudan oluşmaktadır. Sayısal bir görüntüden plakanın bulunması, bulunan plaka görüntüsünden karakterlerin ayrıştırılması, ayrıştırılan karakterin tanınmasıdır. Bu tez çalışmasında ikinci ve üçüncü konular üzerine bir çalışma yapılmıştır. Önceki yapılan çalışmalar incelendiğinde daha çok yapay sinir ağları ile karakter tanınmaya çalışıldığı görülmüştür. Bu çalışmada yapay sinir ağları veya karmaşık matematiksel işlemler yerine insan gözü ile karakterin nasıl algılandığına dikkat edilmiştir. Plaka görüntüsü gri seviyeye indirgenmiş, eşik değeri hesaplanmış ve ikili sisteme çevrilmiştir. Plaka görüntüsü ikili sisteme çevrildikten sonra dikey ve yatay doğrultularda taranarak karakterlerin sınırları bulunmuştur. Bulunan karakterler soldan sağa, sağdan sola, yukarıdan aşağıya, aşağıdan yukarı taranarak her bir karakter için özellik sınıfları oluşturulmuştur. Daha önceden karakterler için hazırlanmış olan özellik sınıfları veritabanı ile karakterin özellik sınıfı karşılaştırılmıştır. Karakterin diğer karakterlere benzeme oranı kullanıcıya gösterilmiştir. Çalışma esnasında T.C. araç plakaları kullanılmıştır.

Anahtar Kelimeler : Araba Plakası Tanıma

To My Son and Wife

ACKNOWLEDGMENTS

I express sincere appreciation to my supervisor, Asst Assoc. Prof. Hakan TORA and . Asst. Prof. Dr. Nergiz Ercil ÇAĞILTAY for their guidance, constructive proofreading, and many fruitful discussions. I thank them for helping me during writing this thesis.

I also wish to acknowledge my parents for always supporting my choices in life and my colleague at HAVELSAN for their support and encouragement.

TABLE OF CONTENTS

LIST OF TABLES TABLE

LIST OF FIGURES FIGURES

LIST OF ABBREVIATIONS

MSPaint	: Microsoft Paint

CHAPTER 1

INTRODUCTION

The requirement of the licence plate number recognition has emerged many years ago. Accordingly, many studies have been established on this subject. It is possible to discuss these studies under two main headings:

- Active Systems
- Passive Systems

Active systems use laser, radio frequency and other technologies for recognizition of the vehicle. A barkot is placed on the car in laser systems. The vehicle has a tag in the radio frequency system. Into this tag, it is also possible to install some other information along with the licence plate of the vehicle. The RFID reader is placed where we would like the vehicle to be identified. RFID reader can read the tag information of the vehicle from a certain distance. Thus, the recognition of the vehicle is provided. The cost of this system was high initially. RFID technology has become cheaper with the developing technology. The cost has come quite down since 2000. The image of the vehicle is not needed in this system. The system is not fallible. This is why it is a reliable system. In recent years, RFID systems can be offered where a serious solution is absent and the risk of error is not accepted in environment. For instance, the highway transition system is also used in Turkey.

In passive systems, the vehicle license plate image is processed for recognition. Therefore, license plate recognition depends on the image quality. Noises in the image plate make the recognition difficult. Accordingly, incorrect results may occur.

In this thesis an approach for the license plate recognized by using a passive approach is proposed. The proposed algorithm is simple and more understandable according to the current approached found in the literature Which use comlex mathematical algorthms.

The rest of this thesis is organized as follows. In Chapter 2, a literature review is provided. In Chapter 3, concepts of the basic image processing is introduced. In Chapter 4, the case study is presented. In chapter 5, the software application usage is shown. In Chapter 6, tests and results are given. Finally, the discussion and conclusion is given in Chapter 7.

1.1 Statement of the Problem

This thesis regards the automatic license plate recognition as the scope of the problem. The license plate recognition has been accepted mandatory due to too many numbers of cars. Active systems are easier to process the license plate recognition. Sun light, shadow, mud and other factors in the plate image make it difficult to recognize at passive plate recognition systems. A portion of the license plate characters which is deleted in the plate makes it difficult to recognize it in the passive recognition system. The plate image is not used in the active systems. Thus, these factors do not affect the license plate recognition.

1.2 Scope and Outline of the Thesis

In this thesis, the study is to be recognized as passive license plate. This thesis tries to identify the license plates in Turkey in a passive way.

License plate recognition system consists of three main topics: finding plate location from digital image, character segmentation from the plate image and character recognition from segmented characters. In this thesis, a study is conducted on character segmentation from plate image and character recognition from segmented characters. Plate image was converted into the binary system. The boundaries of each character were found. For each character, four feature classes were created in these ways: from left to right, from right to left, from top to bottom and from bottom to top. This feature classes were compared to the database feature classes. Similarity rate of the character of each one is shown to the user.

Information about how much and which direction the character change is available in features classes.

With Turkish licence plates, the first two and last two characters are numbers. The third one is a letter. This rule is applied in this thesis.

CHAPTER 2

LITERATURE REVIEW

In general, it is possible to divide previous passive studies into 2 parts. The one using Artificial Neural Networks and others using other methods. In Neural network systems, each character is introduced as an input data to the system. The system learns these characters. After that, it compares the input data characters to the existing characters. As a result, the character which has very similar rate is activated.

In this study, [1] is used for Inductive Learning Based Method and SVM method. Inductive Learning Based Method is used to divide all classes into smaller groups. SVM method is used for classification of these groups. All these methods are applied on the plate characters. Then, a training process is used for each character.

This system [2] is a web based system. This system consists of three modules:

- Creating digital image form the video signals, using hardware for this;
- Using pattern recognition and artificial intelligence for plate recognition;
- Accessing to the database from web browser and run query for vehicles

In this system [3] morphological operators are used for preprocessing. After preprocessing, Template Matching is used for character recognition. This system is used for Macao-style license plates.

In this study, it is the first time that the plate image is normalized [4]. Scaling and cross-validation are applied for remove the outliers and find clear parameters for SVM method. Then use SVM method for character recognition. Correct recognition rate is higher then neural network systems.

This method [5] is applicable in camera-in-motion applications. Images are acquired via a webcam. The light conditions, background and position of the vehicle are not important for character recognition. This method can localize different sizes of the plate from the image. After localization of the plate, the characters are segmented. Multiple neural networks are used for character recognition. The correction rate is %95 in this method. This application was used in University of Málaga (Spain) in the entrance of the Computer Science building.

Sobel color edge detector is used for detecting vertical edges in this study [6]. Then, the invalid edge is eliminated. The license plate region was searched by using template matching. Mathematical morphology and connected component analysis was used for segmentation. Radial basis function neural network was used for character recognition. This system is also successful in night hours and daytime real conditions.

Plate location is found by using plate background and the character's color in this study [7]. For segmentation, the column sum vector is obtained. The Artificial Neural Network is used for character recognition.

This system is designed for Islamabad Computerized Number Plates [8]. SCAN_NP algorithm is developed for plate extract. This algorithm can find candidate plates. The algorithm brings out these candidate objects which would turn out to be the plate characters. The objects have horizontal and vertical lines. This algorithm scans the image to remove the noises from the image. Neural Network and template matching is used for character recognition. The correction rate is %90 in this method.

This method [9] is used in off-line Thai license plate recognition. Hausdorff Distance technique is used for recognition. The correction rate in this method is %92.

This method [10] is used for Chine license plate recognition. The plate image is converted into a binary image. Then the noises are removed from the image. The skeleton is used for generating the feature of the character. Then the character is normalized to size 8*16 pixels. The plate image is processed in the Back-Propagation Neuronal Network for recognition after being normalized. Back-Propagation Neuronal Network is used for character recognition.

This method [11] is not used to preprocess for recognition. Image transformation is applied for original license plate picture. After transformation process, in the database, the number of the input and the data increase. Convolution neural network is used for character recognition. The correction rate in this method is %98.

This method [12] has two modules: plate locating and plate segmentation modules. Fuzzy geometry is used for the first module. Fuzzy C mean is used as the second module. The correction rate for segmentation is %94.24

In this method [13] blob labeling and clustering are used for segmentation. The studies of Kirsch, Sobel, Laplacian, Wallis, Prewitt, Frei Chen on edge detectors are compared and contrasted, and Kirsch's edge detector is regarded as the most appropriate one among others. This method is used neural network for classification and recognition character.

Field-programmable gate array (FPGA) is used in this study [14]. Gabor filter, threshold and connected component labeling algorithms are used for finding plate location. After segmentation, a self-organizing map (SOM) neural network is used for character recognition. Hardware is used in this system. Then, the system spends less time than computer-based recognition system for does character recognition. Moreover, the system is mobile.

Two-layer Markow network is used for segmentation and character recognition in this study [15]. In this method "8" and "B" can be mixed. This study is made synthesising with Housdorff and Shape context methods.

A median filter is used for removing noises from plate image in this method [16]. Hough transform is used for rotating the plate image when it is necessary. Adoptive threshold method is used for binarization. Segmentation is applied after binarization. FFM (Filled Function Method)-BPNN (BP Neural Network) algorithm is used for character recognition. This method is used for Chine character. And can be use for fingerprint and retina recognition.

In this paper [17], they use a different algorithm which is developed for Italian Highway Company to control the trafic flow. Thier classification algorithm has two stages. One of them is parameters setting phase and the other is character classification by embedded Generative Models with using covariance matrix. They search for 7.000 diffent images of license plates. In this research, correct classification is 98.1% with using their algorithm which is called Generative Models.

In this study [18] Robert edge detector and morphology operator is used for finding plate edge from the picture. Horizontal and vertical projections are used for rotating plate image when it is needed. Least squares support vector machines are used for character recognition.

This method [19] is used different method for finding plate location from the picture. White and black pixels have different weight according to this method. Hence, this method uses this feature for finding plate location. Hybrid neural network is used for character recognition. The correction rate in this method is %97.

This method uses is [20] Ostu Threshold algorithm for converting into binary level for plate image. Character coordinates are used for character segmentation. Improved pattern matching algorithm is used for character recognition. The correction rate is %98 in this method. Scanline checking is used for plate localization in this study [21]. Dynamic projection warping is used for character recognition.

This method [22] is about plate localization and character segmentation. AdaBoost algorithm is used for finding plate localization from the image. Vertical edge detection and horizontal projection histogram are used for upper and lower boundary disposal. Image binarization and vertical projection histogram are used for character segmentation. The correction rate is %96.4 in this method.

This method [23] uses composite colors for detecting the plate area of the image. Horizontal and vertical histograms are used for character segmentation. Artificial hippocampus algorithm is used for character recognition. The correction rate is %95 in this method.

In this thesis, the image of the plate is converted to the binary level by a written application program. The feature classes were created with plate image from left to right, from right to left, from top to bottom, from bottom to top scanning. Recognition of the characters was maintained by making use of these feature classes.

CHAPTER 3

CONCEPTS OF THE BASIC IMAGE PROCESSING

3.1 Image Description

Data formats should be converted into computer readable for interpretation of the images. This process is called Digitizing. It is possible to use different methods for converting an image into digital form. One of the methods is using a scanner. The other method is a system in which the image is converted into digital format with using analog to digital transformation. The other method is using multi-channel scanners placed on the aircraft or satellites for remote sensing.

The digital image is formed as a result of process which is converts analog signals into digital signals. This process is possible with the subject of energy spread (analog signal) predicted by an electromagnetic sensor to the digital signal into the detection range. Figure 3.1 is a related example for image digitializing. [35]

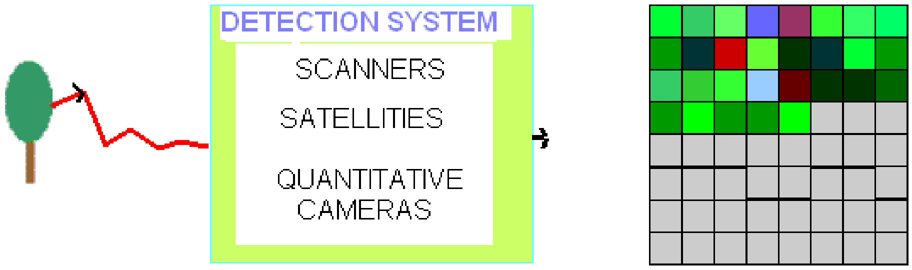


Figure 3.1: Image Digitialization

3.2 Image Types

The smallest component at the image which are called pixels. Digital images are formed trough a matrix of pixels whose dimension is MxN. Figure 3.2 is an example of an image which is formed by pixels.

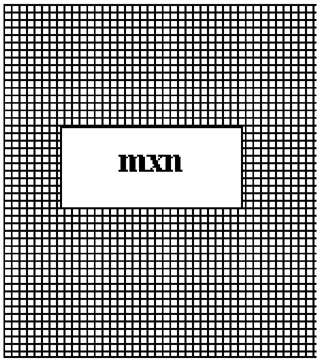


Figure 3.2: The Image is formed by Pixels.

There are two basic features of creating a pixel.

- Radiometric feature: It is a gray value which creates pixel detection on the electromagnetic spectrum.
- Geometric feature: It is a matrix coordinate which indicates the position of the image matrix.

Firstly, there is an example of the black-white image for explaining image digitialization easily. Black-white image consists of two gray values. In this image, each pixel is either black or white. In Figure 3.3 white pixels are shown with a value of "1", black pixels are shown with a value of "0".

Expressing the image in this way is called binary image. [35]

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	0	0	0	0	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	0	0	0	0	0	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	0	0	0	0	0	1	1	1	1	1	1	1	1
1	1	1	1	1	1	0	0	0	0	0	0	1	1	1	1	1	1	1	1
1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
1	1	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
1	1	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
1	1	0	0	0	0	1	1	0	0	0	0	1	1	1	1	1	1	1	1
1	1	0	0	1	1	1	1	0	0	0	0	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	0	0	0	0	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	0	0	0	0	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	0	0	0	0	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	0	0	0	0	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	0	0	0	0	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	0	0	0	0	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	0	0	0	0	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	0	0	0	0	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	0	0	0	0	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	0	0	0	0	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	0	0	0	0	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	0	0	0	0	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	0	0	0	0	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Figure 3.3: The Binary Image Level of Character“1”

In Figure 3.4 the image coordinate system is shown on the image matrix.

	1	**2**	**3**	**……..**		**M**
1						
2		**(x-1,y-1)**	**(x-1,y)**	**(x-1,y+1)**		
3		**(x,y-1)**	**(x,y)**	**(x,y+1)**		
⋮		**(x+1,y-1)**	**(x+1,y)**	**(x+1,y+1)**		
N						

Figure 3.4: The Image Coordinate System for any Picture

Display color is expressed with 24 - bit data on computer screens. Three basic colors are used to generate the image. Red (R), green (G) and blue (B) are these basic colors. Images occur while the three main colors transfer on the screen. This situation is shown on Figure 3.5 with an example. 0,4 - 0,5 mm wavelength corresponds to the color blue, 0,5 - 0,6 mm wavelength corresponds to the color green, and 0,6 - 0,7 mm wavelength corresponds to the color red in electro-magnetic spectrum. [35]

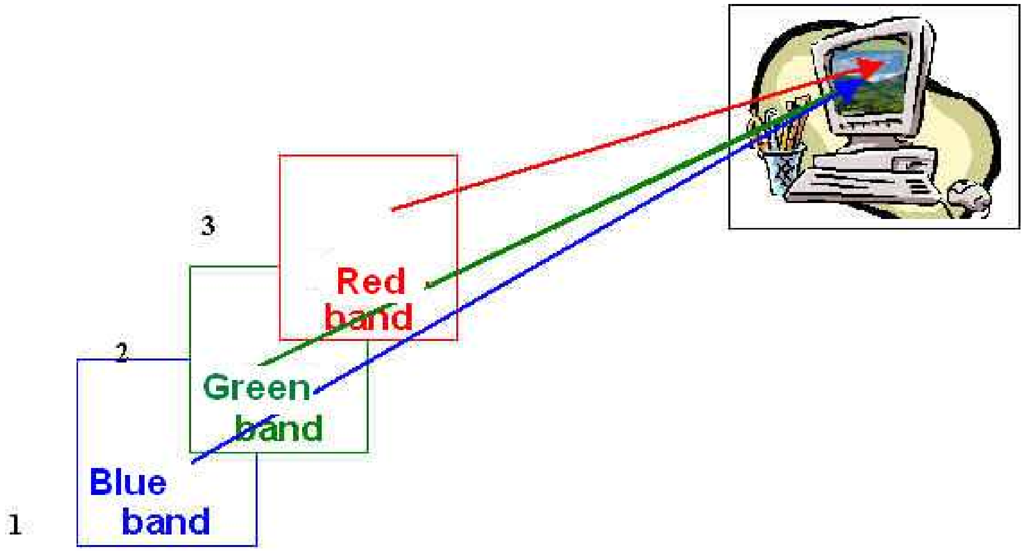


Figure 3.5: Color Image Concepts in Computer

3.3 RGB Color Model

Color distributions are shown in Figure 3.6. Values of the Red, Green and Blue are between 0 - 255 the. Colors upgrade to darker towards to 0 and upgrade to lighter when towards to 255. This situation is explained in the Cartesian coordinate system. The (0, 0, 0) origin point is black, and (1, 1, 1) point is white.

Any color occurs as a result of the merger red, green, blue color with certain coefficients in the coordinate system. Gray color is above the white and black level, combining diagonal corners. [36]

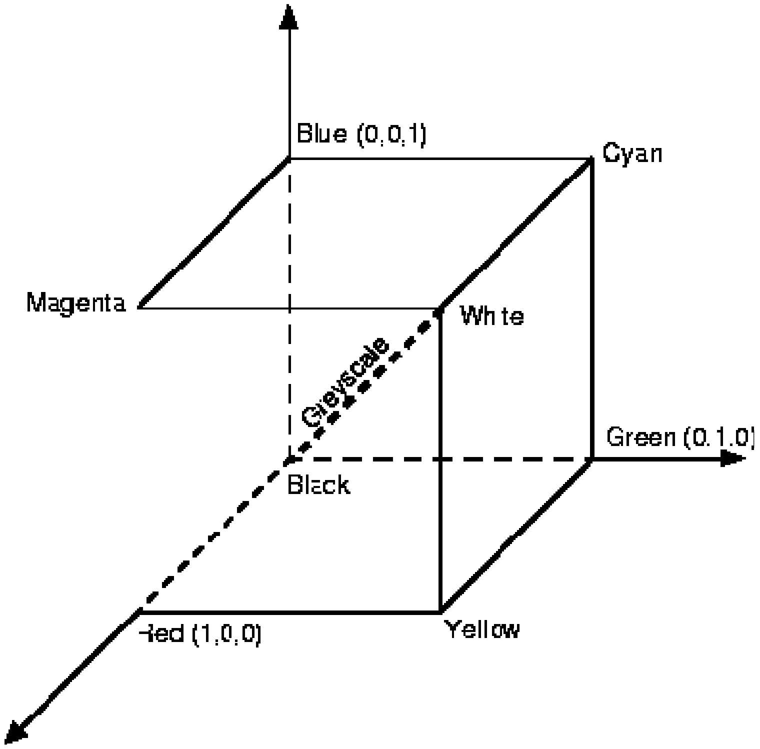


Figure 3.6: RGB Color Module

3.4 Gray Level

The image is converted to gray level to accelerate the image processing. Thus, in the picture, there will remain only black, white and gray values. The process needs to be done before the image is converted into binary level.

3.5 Binary Level

An image consists of numeric values between 0 - 255. The numerical value of the picture is reduced to two values with binary level. Thus, a 8 - bit image is converted into 2 - bit format. The threshold value must be determined for this conversion. Using a fixed threshold value is not correct because of external factors such as sunlight, shadows at real-plate images. A distribution histogram is useful for calculating threshold value. If the pixel value in the image is greater then threshold value, then the pixel value is shown as "0"; and if the image pixel' value is less then threshold value, the pixel value is shown as "1". In this way the image is converted to the binary level.

3.5.1 Histogram Distribution

Expressing values on the image pixels graphically is called image histogram. In each point, pixels on the image histogram and the number of pixels can be seen by using the image histogram. This image histogram does not show positons of the pixels on the image, but shows distribution of light-dark areas on the image. The histogram distribution is required for the calculation of the threshold value. Histogram can be expressed mathematically with the following formula. [37]

P (rk) = nk / n
Where rk: k'nd gray level,
nk: number of total pixels that have gray level,
n: Number of total pixels on the display,
is defined as

Figure 3.7 is a sample for plate image and Figure 3.8 indicates the license plate image distribution.

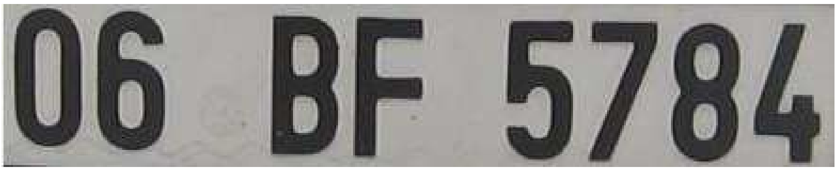


Figure 3.7: Sample Plate Image

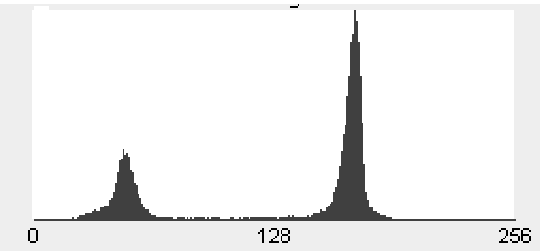


Figure 3.8: Histogram Distribution

3.5.2 Histogram Equalization

Histogram equalization is used to improve color distribution where the image is not clear. In an image, if pixel values are clustered into a specific number, this clustering would be reduced by using histogram equalization.

A histogram equalization is done for the plate image of Figure 3.7; and its histogram distribution is shown in Figure 3.9. A histogram equalization result in a license plate image is shown as figure 3.10. [38]

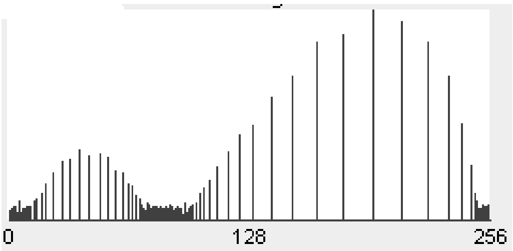


Figure 3.9: The Histogram Distribution After Histogram Equalization

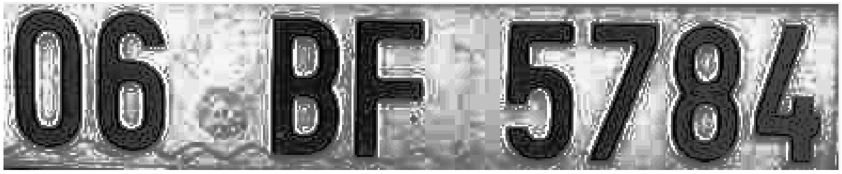


Figure 3:10: The Plate Image after the Histogram Equalization

3.5.3 The Threshold Value

Threshold process is the process of changing and/or disposing the image pixels for a specific value. In this way, the object on the image will be separated from the background. The data of histogram distribution is used for calculating the threshold value.

3.5.4 Otsu Method

The purpose of the Otsu method is to calculate the iterative value of the threshold value. This value is obtained by regarding the in-class value or interclass value as the smallest. Otsu method has been shown as a mathematical expression by figure 3.11. [39].

$$q_1(1) = P(1)$$
$$\mu_1(0) = 0.$$

$$q_1(t+1) = q_1(t) + P(t+1)$$
$$\mu_1(t+1) = \frac{q_1(t)\mu_1(t) + (t+1)P(t+1)}{q_1(t+1)}$$
$$\mu_2(t+1) = \frac{\mu - q_1(t+1)\mu_1(t+1)}{1 - q_1(t+1)}$$
$$\mu = \sum_{i=1}^{I} i\,P(i)$$

$$\sigma_B^2(t) = q_1(t)[1 - q_1(t)][\mu_1(t) - \mu_2(t)]^2$$

Figure 3:11: Otsu Threshold Formula

3.5.5 Calculation of Adaptive Threshold Value

Getting of the threshold value as middle value T = 128 will not be enough to us most of the time to convert to binary level. This situation can be seen in Figure 3.12.

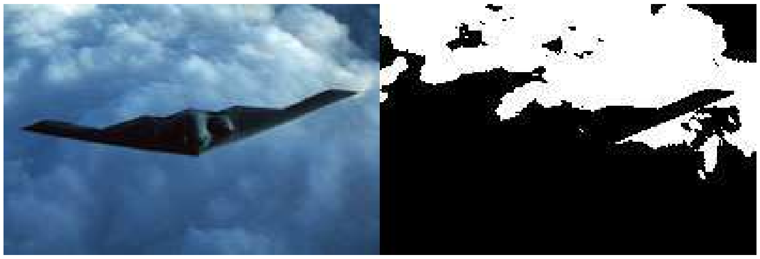


Figure 3:12: Threshold value of 128 for the Image Results

The iterative approach can be used instead of using this one. According to the iterative approach, the algorithm is:

1. An initial treshold value is determined, t (say 128).
2. Two different mean values are calculated for below (m1) and above (m2).
3. The new treshold value is calculated. *tnew = (m1 + m2) / 2.*
4. If the treshold is stabilized *(t = tnew)*, then this would indicate that your new treshold level is this one. Elsewise, t would become *tnew* and it would also reiterate from step2.

When this algorithm runs for the example picture in Figure 3.12, the threshold value is found as t=74. The image in the figure 3.12 is like as figure 3.13 for T=74. [40]

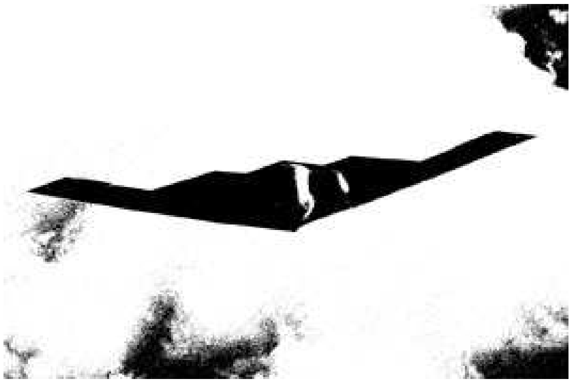


Figure 3:13: Threshold value of 74 for the Image Result

3.6 Morphological Operators

Morphological operators are used to explicit image objects and/or distinguish objects in the image. Morphological operators can be applied to gray-level image and binary level image. [41]

3.6.1 Erosion

Odd matrixes are used for erosion processes such as 3x3, 5x5, and 7x7. This matrix will be moved from left to right on the image. No processing will be done if all pixels of the matrix is at the background. If any pixel of the matrix is on the object, the area under the matrix pixel and object intersection point is included to the background. In Figure 3.14, an example of 3x3 size matrix is given [41].

1	1	1
1	1	1
1	1	1

Figure 3.14: An Example of a 3x3 Matrix

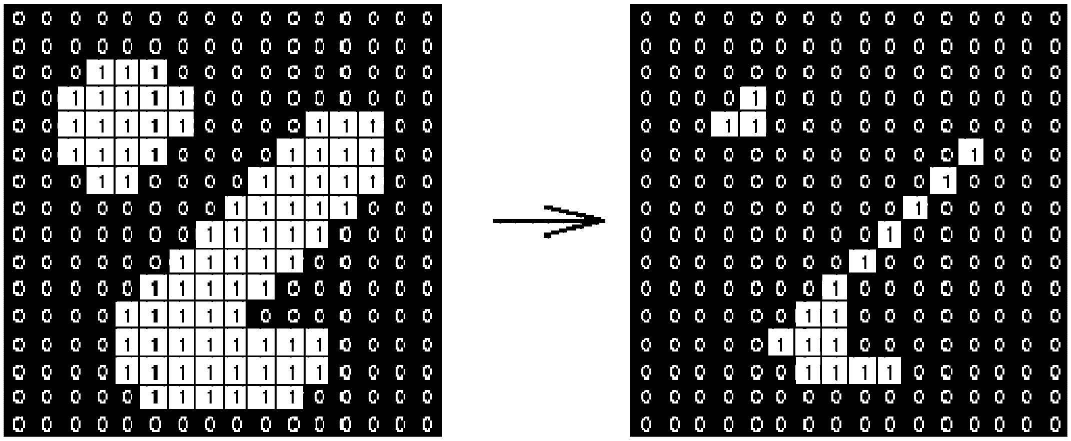


Figure 3.15: A 3x3 Matrix Using Erosion Sample Configuration

3.6.2 Dilation

Odd matrixes are used for dilation processes such as 3x3, 5x5, and 7x7. This matrix will be moved from left to right on the image. If the objects in the image intersect the center of the matrix coordinate, the remaning space objects are combined at the bottom of the matrix.

In figure 3.16, the dilation process has been shown by using a 3x3 size matrix. [41]

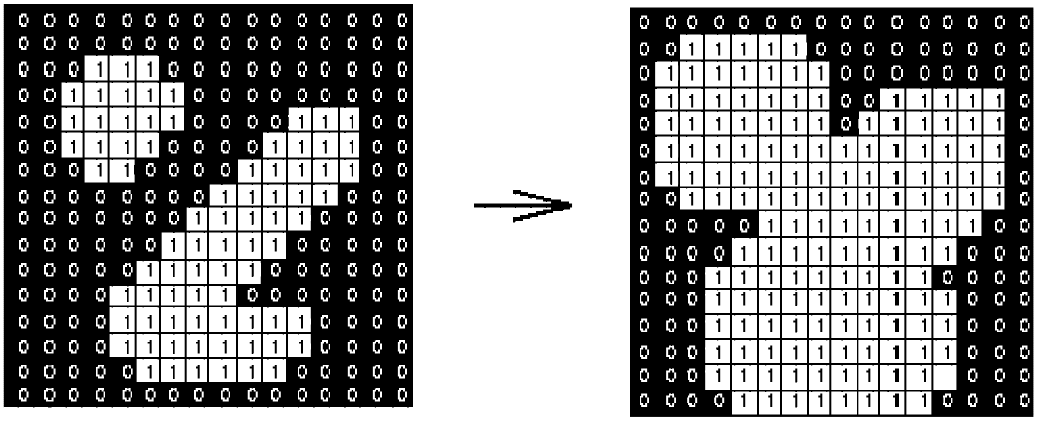
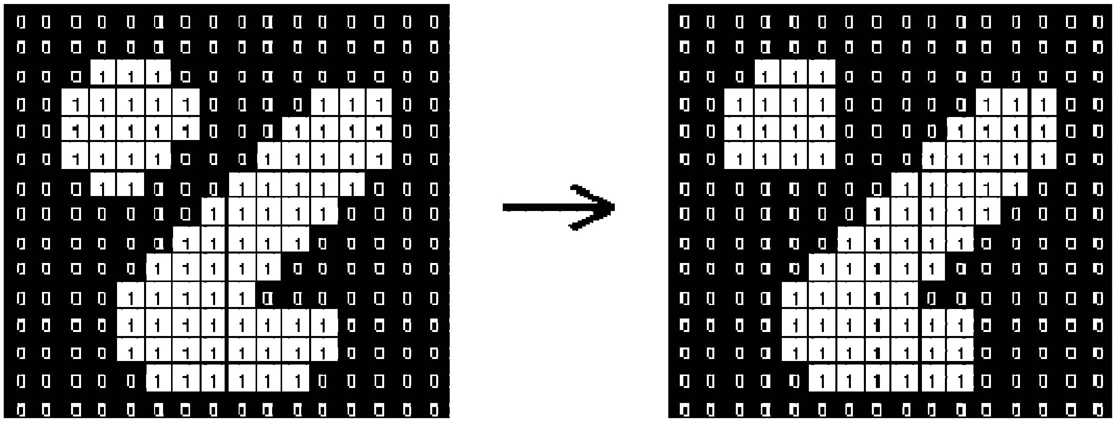


Figure 3:16: A Dilation Example by a 3x3 Matrix

3.6.3 Opening

Firstly erosion and then dilation operators are applied. In figure 3.17 the result of the opening process has been shown by using 3x3 size of matrix. [41]



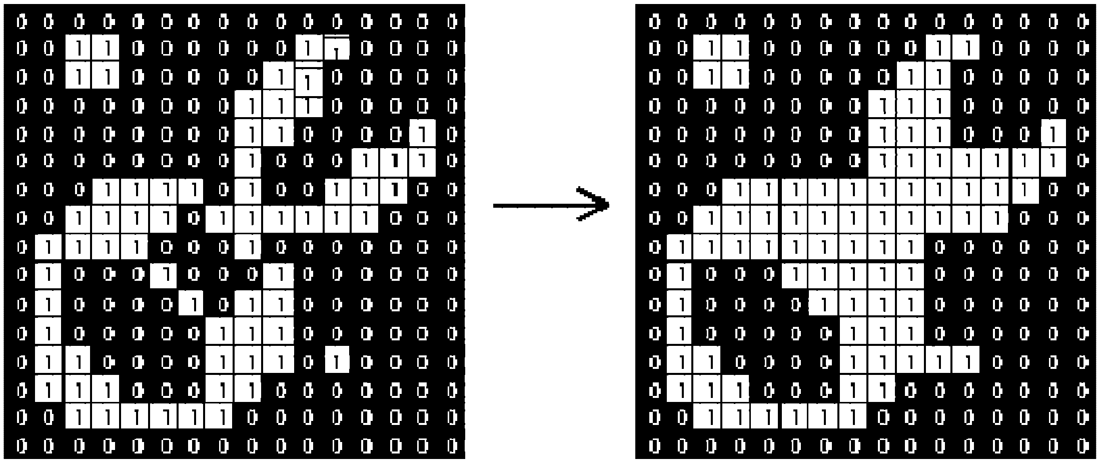
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0
0	0	1	1	1	1	1	0	0	0	0	1	1	1	0	0
0	0	1	1	1	1	1	0	0	0	1	1	1	1	0	0
0	0	1	1	1	1	0	0	0	1	1	1	1	1	0	0
0	0	0	1	1	0	0	0	1	1	1	1	1	0	0	0
0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0
0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0
0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0
0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0
0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0
0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0
0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0
0	0	1	1	1	1	0	0	0	0	0	1	1	1	0	0
0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0
0	0	1	1	1	1	0	0	0	1	1	1	1	1	0	0
0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0
0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0
0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0
0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0
0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0
0	0	0	0	1	1	1	1	1	1	1	0	0	0	0	0
0	0	0	0	1	1	1	1	1	1	1	0	0	0	0	0
0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure **3.17: An Opening Example by a 3x3 Matrix**

3.6.4 Closing

Firstly dilation and then erosion operators are applied. In figure 3.18 the result of the closing process has been shown by using 3x3 size of matrix. [41]



0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	0	0	0	1	1	0	0	0	0
0	0	1	1	0	0	0	0	0	1	1	0	0	0	0	0
0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0
0	0	0	0	0	0	0	0	1	1	0	0	0	0	1	0
0	0	0	0	0	0	0	0	1	0	0	0	1	1	1	0
0	0	0	1	1	1	1	0	1	0	0	1	1	1	0	0
0	0	1	1	1	1	0	1	1	1	1	1	1	0	0	0
0	1	1	1	1	0	0	0	1	0	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	1	0	0	0	0	0	0
0	1	0	0	0	0	1	0	1	1	0	0	0	0	0	0
0	1	0	0	0	0	0	1	1	1	0	0	0	0	0	0
0	1	1	0	0	0	0	1	1	1	0	1	0	0	0	0
0	1	1	1	0	0	0	1	1	0	0	0	0	0	0	0
0	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	0	0	0	1	1	0	0	0	0
0	0	1	1	0	0	0	0	0	1	1	0	0	0	0	0
0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0
0	0	0	0	0	0	0	0	1	1	1	0	0	0	1	0
0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	0
0	0	0	1	1	1	1	1	1	1	1	1	1	1	0	0
0	0	1	1	1	1	1	1	1	1	1	1	1	0	0	0
0	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0
0	1	0	0	0	1	1	1	1	1	0	0	0	0	0	0
0	1	0	0	0	0	1	1	1	1	0	0	0	0	0	0
0	1	0	0	0	0	0	1	1	1	0	0	0	0	0	0
0	1	1	0	0	0	0	1	1	1	1	1	0	0	0	0
0	1	1	1	0	0	0	1	1	0	0	0	0	0	0	0
0	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3:18: A Closing Example by a 3x3 Matrix

CHAPTER 4

A CASE STUDY (TC REGISTERED VEHICLE PLATE RECOGNITION)

The Application program has been tested on ideal characters before being tested on the actual plate images. The program, MSPaint (Microsoft Paint), is used for creating the ideal character in the Windows Operating System. All characters which are in plate image are generated by using the MSPaint program. Firstly, the recognition of these characters is checked. After this trial successfully fulfills the desired task, then the recognition of the real licence plate image is worked upon.

In MSPaint, the characters were generated as similar to real plate character sizes. Sizes of the characters which are generated are 36, 48, 78 punto.

The success rate of character recognition of car license plates is lower than the one which is generated by using MSPaint. The reason for this is because the characters of license plate images do not have as sharp lines as the ones which are generated by using MSPaint. Additionally, shadow and sunlight on the images might be the reason of this low recognition rate.

Firstly pre-processing steps were done on the plate image for the recognition of the licence plate characters.By doing this, the plate image noises were reduced as much as possible. The characters which are prepared by MSPaint do not need image processing steps. These stages are not required since characters are ideal in the image.

4.1 Converting to the Gray Level

Firstly, a grey – level process is applied upon the plate image. In this way color information in the image is removed. Plate image was converted to the gray level.

4.2 Converting to the Binary Level

The threshold value must be calculated to convert the real plate image to the binary level. In the consequence of the calculating the threshold value; if the value of the pixels in the plate image is greater than the threshold value, it is expressed as “0”; if it is less than the threshold value, it is expressed as “1”. In this way, the image plate is converted into the binary level.

Threshold value is difficult to be accepted since 128 characters are created by using MSPaint. This value does not give successful results in the plate images. Otsu Thresholding method is used for plate characters. Although this method is successful in general-including the shadow-, the intense sunlight in the plate would not be successful. Therefore this method was abandoned. Then calculating the value of the dynamic threshold method is used. Thus, noises such as shadow, sun light, "TR" logo were removed from the image plate. The image plate is converted to the binary system more clearly. In Figure 4.1, a sample plate image and an image of plate which is converted into the binary system by using the Dynamic Treshold Value are given.

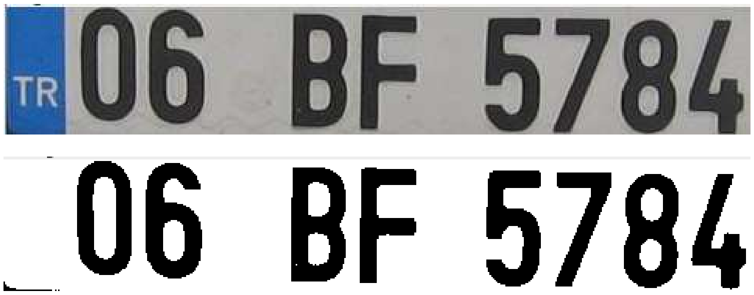


Figure 4.1: The conversion of the plate image into the Binary System by using the Dynamic Treshold Value

4.3 Erozion and Dilation Operators

The Erosion and the Dilation processes are applied on the plate image in order to destroy the little noise and to make the characters more specific. This operation is not required for recognition of characters prepared by using MSPaint.

4.4 Finding the Character Boundaries

Each character boundaries must be found in plate image which is converted into the binary system. For this, the image plate which was converted to binary system is scanned in vertical and horizontal directions. Left and right borders of the each character are identified after vertical scanning, upper and lower borders of the each character are identified after horizontal scanning. The upper and lower boundaries of license plate image boundaries may not be the same for each plate character. Upper and lower boundaries are found again for each plate character. The upper and lower boundaries are checked for each character after finding the character boundaries. Horizontal line is scanned again. The upper and lower boundaries of the characters may be changed after the scan result.

Noises can be perceived as characters in the image plate. This character is interpreted as noise if the size of the character’s boundaries is less than half of the plate size. The noise is deleted from the set of characters.

4.5 Character Recognition Algorithm

Character recognition algorithm is executed for the specified bounds of the character. Character recognition algorithm scans character line by line not only from left to right and from right to left, but also it scans column by column from top to bottom and bottom to top. During scanning, the character direction is detected from change of the coordinates of the characters “1”.

For example, change in the character may be in left, right, straight or jumping (expressing sudden changes) during scanning line by line characters from left to right, from right to left. The change in the character may be up, down, straight or jumping (expressing sudden changes) during scanning column by column characters from top to bottom, from bottom to top. The size of the characters will be similar in the same plate. The license plate of the size of the characters will be different depending on the license plate image's display closeness. The size of characters must be matched up with the percentage slice to evaluate the percentage change in the character. The width of the character is divided into hundred steps. On each step "1" is rated. While the width of the character is scanned step by step, the total value of these steps value will be "100". Thus, change direction of the character has been found in the percentage. The same procedure is applied to the character's size. In table 4.1, the information about the direction of character change and the amount in percentages have been displayed for the characters "A", "L", "H", "2" and "7" .

Table 4.1: "A", "L", "H", "2" and "7" – Character Changes in Percentages

	Left to right	**Right to left**	**Top to bottom**	**Bottom to top**
A	%100 left	%100 right	%50 up %50-100 down	%15 direct %15-30 up %30-70 direct %70-85 down %85-100 direct
L	%100 left	%89 direct %89-90 jump right %90-100 direct	%20 direct %20-21 jump down jump %21-100 direct	%100 direct
H	%100 direct	%100 direct	%15 direct %15-16 jump down %16-82 direct %82-83 jump down %83-100 direct	%15 direct %15-16 jump down %16-82 direct %82-83 jump down %83-100 direct
2	%70 left %70-71 jump right %71-100 left	%30 right %30-90 left %90-91 jump right %91-100 direct	%40 up %41-65 direct %65-89 down %89-90 jump down %90-100 direct	%100 direct
7	%12 direct %12-13 jump right %13-100 left	%10 direct %10-100 left	%100 direct	%19 direct %19-20 jump down %20-38 jump up %38-100 up

This change of characters can easily be detected for the generated characters by using MSPaint. Thus, characters can be recognized. Characters which change in sharp lines are not clear in the license plate image. (There may be noise, changes in the shape of a ladder and recess ledges in the plate image.) Therefore, problems have been experienced in character recognition.

"A" character of the boundary change is very similar to the information in table 4.1 which is generated in MSPaint. However, for the character “A”, change of the boundaries "is different in the image plate. "A" characters which are created using MSPaint and converted into the state of binary level has been displayed in figure 4.2. "A" characters which are taken from plate image were converted into the state of binary level has also been displayed in figure 4.3.

```
0000000000000000000000111111110000000000000000000000
0000000000000000000000111111110000000000000000000000
0000000000000000000001111111110000000000000000000000
0000000000000000000001111111111000000000000000000000
0000000000000000000001111111111000000000000000000000
0000000000000000000011111111111100000000000000000000
0000000000000000000011111111111100000000000000000000
0000000000000000000011111011111100000000000000000000
0000000000000000000111111011111110000000000000000000
0000000000000000000111111001111110000000000000000000
0000000000000000001111111001111111000000000000000000
0000000000000000001111110001111111000000000000000000
0000000000000000001111110000111111000000000000000000
0000000000000000011111110000111111100000000000000000
0000000000000000011111110000111111100000000000000000
0000000000000000011111110000011111110000000000000000
0000000000000000111111100000011111110000000000000000
0000000000000000111111100000011111110000000000000000
0000000000000001111111000000001111111000000000000000
0000000000000001111111000000001111111000000000000000
0000000000000011111110000000000111111100000000000000
0000000000000011111110000000000111111100000000000000
0000000000000011111110000000000111111110000000000000
0000000000000111111100000000000011111110000000000000
0000000000000111111100000000000011111110000000000000
0000000000000111111000000000000001111111000000000000
0000000000001111111000000000000001111111000000000000
0000000000001111111000000000000001111111100000000000
0000000000011111110000000000000000111111100000000000
0000000000011111110000000000000000111111100000000000
0000000000011111110000000000000000111111110000000000
0000000000111111100000000000000000011111110000000000
0000000000111111100000000000000000011111110000000000
0000000000111111111111111111111111111111111000000000
0000000001111111111111111111111111111111111000000000
0000000001111111111111111111111111111111111100000000
0000000011111111111111111111111111111111111100000000
0000000011111111111111111111111111111111111110000000
0000000011111111111111111111111111111111111110000000
0000000111111111111111111111111111111111111110000000|
0000000111111100000000000000000000000011111111000000
0000000111111100000000000000000000000011111111000000
0000001111111000000000000000000000000001111111100000
0000001111111000000000000000000000000001111111100000
0000011111111000000000000000000000000000111111100000
0000011111110000000000000000000000000000111111110000
0000011111110000000000000000000000000000111111110000
0000111111100000000000000000000000000000011111111000
0000111111100000000000000000000000000000011111111000
0000111111110000000000000000000000000000001111111000
0001111111100000000000000000000000000000001111111100
0001111111100000000000000000000000000000001111111100
0011111111100000000000000000000000000000000111111110
0011111111000000000000000000000000000000000111111110
0111111111000000000000000000000000000000000111111111
0111111111000000000000000000000000000000000011111111
0111111100000000000000000000000000000000000011111111
```

Figure 4.2: A Sample was prepared in MSPaint for a Medium "A" Character of the Binary Level Impressions

```
-----------------------------------------
00000000111111111111111111111111110000000
00000000111111111111111111111111110000000
00000000111111111111111111111111110000000
00000000111111111111111111111111110000000
00000000111111111111111111111111110000000
00000000111111111111111111111111110000000
00000000111111111111111111111111110000000
00000000111111111111111111111111111000000
00000001111111111111111111111111111000000
00000001111111111111111111111111111000000
00000001111111111111111111111111111000000
00000011111111111111111111111111111000000
00000011111111111111111111111111111100000
00000011111111111111111111111111111100000
00000011111111111111111111111111111110000
00000011111111111111111111111111111110000
00000111111111111111111111111111111110000
00000111111111111111111111111111111110000
00000111111111111111111111111111111110000
00000111111111111111111111111111111110000
00000111111111111111111111111111111111000
00000111111111111111111111111111111111000
00000111111111111111111111111111111111000
00001111111111111111001111111111111111000
00001111111111111100000111111111111111000
00001111111111111100000011111111111111000
00000111111111111000000001111111111111100
00001111111111111000000001111111111111100
00001111111111111000000001111111111111100
00011111111111111000000001111111111111100
00011111111111111000000001111111111111100
00011111111111110000000001111111111111110
00011111111111110000000000111111111111110
00011111111111110000000000011111111111111
00011111111111110000000000111111111111111
00011111111111110000000000111111111111111
00011111111111110000000000111111111111111
00011111111111110000000000111111111111111
00011111111111110000000000011111111111111
00011111111111110000000000011111111111111
00111111111111110000000000011111111111111
00111111111111111000000000111111111111111
00111111111111111000000000111111111111111
00111111111111111000000000111111111111111
00111111111111111110000001111111111111111
01111111111111111111111111111111111111111
01111111111111111111111111111111111111111
01111111111111111111111111111111111111111
01111111111111111111111111111111111111111
11111111111111111111111111111111111111111
11111111111111111111111111111111111111111
11111111111111111111111111111111111111111
11111111111111111111111111111111111111111
11111111111111111111111111111111111111111
11111111111111111111111111111111111111111
11111111111111111111111111111111111111111
11111111111111111110000001111111111111111
11111111111111110000000000011111111111111
11111111111111100000000000000111111111111
11111111111111000000000000000111111111111
11111111111110000000000000000111111111111
11111111111110000000000000000011111111111
11111111111110000000000000000011111111111
11111111111110000000000000000011111111111
```

Figure 4.3: Plate Images Received From the Real Character "A" of the Binary Level Impressions

The changing edge of the image as in figure 4.2 is shown in figure 4.4. The changing edge of the image in figure 4.3 is also shown in figure 4.5.

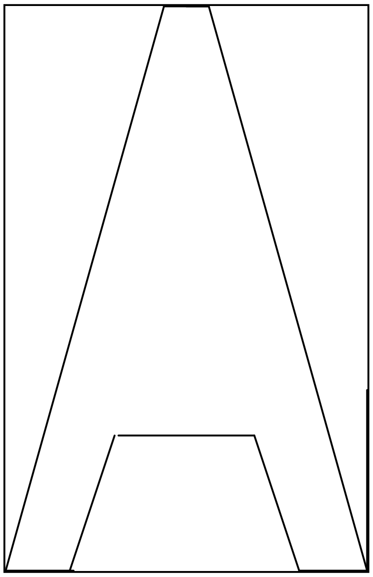


Figure 4.4: The Edge Change of the image in the Figure 4.2

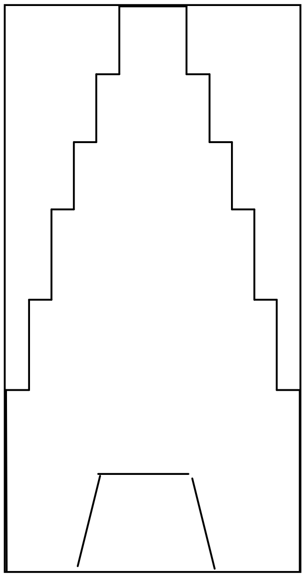
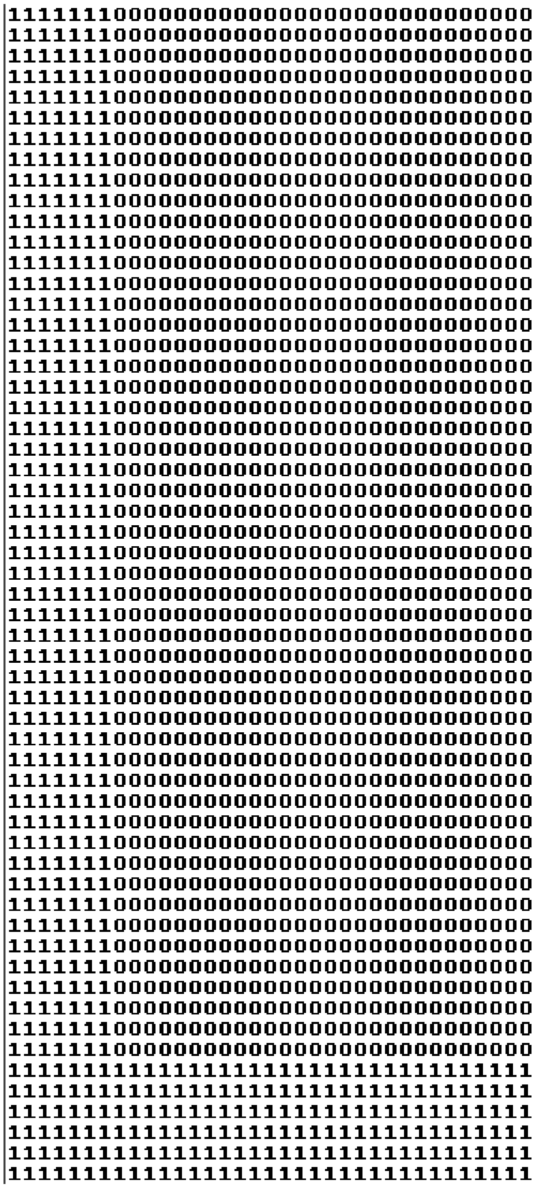


Figure 4.5: The Edge Change of the image in the Figure 4.3

"L" characters which are created using MSPaint and which were converted into the state of binary level has been displayed in figure 4.6. "L" characters which are taken from plate image and were converted into the state of binary level has also been displayed in Figure 4.7.



```
11111110000000000000000000000000000
11111110000000000000000000000000000
11111110000000000000000000000000000
11111110000000000000000000000000000
11111110000000000000000000000000000
11111110000000000000000000000000000
11111110000000000000000000000000000
11111110000000000000000000000000000
11111110000000000000000000000000000
11111110000000000000000000000000000
11111110000000000000000000000000000
11111110000000000000000000000000000
11111110000000000000000000000000000
11111110000000000000000000000000000
11111110000000000000000000000000000
11111110000000000000000000000000000
11111110000000000000000000000000000
11111110000000000000000000000000000
11111110000000000000000000000000000
11111110000000000000000000000000000
11111110000000000000000000000000000
11111110000000000000000000000000000
11111110000000000000000000000000000
11111110000000000000000000000000000
11111110000000000000000000000000000
11111110000000000000000000000000000
11111110000000000000000000000000000
11111110000000000000000000000000000
11111110000000000000000000000000000
11111110000000000000000000000000000
11111110000000000000000000000000000
11111110000000000000000000000000000
11111110000000000000000000000000000
11111110000000000000000000000000000
11111110000000000000000000000000000
11111110000000000000000000000000000
11111110000000000000000000000000000
11111110000000000000000000000000000
11111110000000000000000000000000000
11111110000000000000000000000000000
11111110000000000000000000000000000
11111110000000000000000000000000000
11111110000000000000000000000000000
11111110000000000000000000000000000
11111110000000000000000000000000000
11111110000000000000000000000000000
11111110000000000000000000000000000
11111110000000000000000000000000000
11111110000000000000000000000000000
11111110000000000000000000000000000
11111110000000000000000000000000000
11111111111111111111111111111111111
11111111111111111111111111111111111
11111111111111111111111111111111111
11111111111111111111111111111111111
11111111111111111111111111111111111
11111111111111111111111111111111111
```

Figure 4.6: A Sample Prepared in MSPaint Media for Character "L"

```
11111111111110000000000000000000000
11111111111110000000000000000000000
11111111111110000000000000000000000
11111111111110000000000000000000000
11111111100000000000000000000000000
11111111100000000000000000000000000
11111111100000000000000000000000000
11111111100000000000000000000000000
11111111100000000000000000000000000
11111111100000000000000000000000000
11111111100000000000000000000000000
11111111100000000000000000000000000
11111111111110000000000000000000000
11111111111110000000000000000000000
11111111111110000000000000000000000
11111111111110000000000000000000000
11111111111110000000000000000000000
11111111111110000000000000000000000
11111111111110000000000000000000000
11111111111110000000000000000000000
11111111111000000000000000000000000
11111111111000000000000000000000000
11111111111000000000000000000000000
11111111111000000000000000000000000
11111111110000000000000000000000000
11111111111000000000000000000000000
11111111111000000000000000000000000
11111111111000000000000000000000000
11111111111000000000000000000000000
11111111111000000000000000000000000
11111111111000000000000000000000000
11111111110000000000000000000000000
11111111110000000000000000000000000
11111111111000000000000000000000000
11111111111000000000000000000000000
11111111111000000000000000000000000
11111111110000000000000000000000000
11111111110000000000000000000000000
11111111110000000000000000000000000
11111111110000000000000000000000000
11111111110000000000000000000000000
11111111111000000000000000000000000
11111111111000000000000000000000000
11111111110000000000000000000000000
11111111110000000000000000000000000
11111111110000000000000000000000000
11111111111000000000000000000000000
11111111111000000000000000000000000
11111111111000000000000000000000000
11111111111000000000000000000000000
11111111111000000000000000000000000
11111111111100000000000000000000000
11111111111110000000000000000000000
11111111111111000000000000000000000
11111111111111000010000000000000000
11111111111111111111111111111111111
11111111111111111111111111111111111
11111111111111111111111111111111111
11111111111111111111111111111111111
11111111111111111111111111111111111
11111111111111111111111111111111111
11111111111111111111111111111111111
11111111111111111111111111111111111
11111111111111111111111111111111111
```

Figure 4.7: Real-plate Images Received from the Character "L"

The changing edge of the image in Figure 4.6 is shown in figure 4.8. The changing edge of the image in figure 4.7 is also shown in figure 4.9.

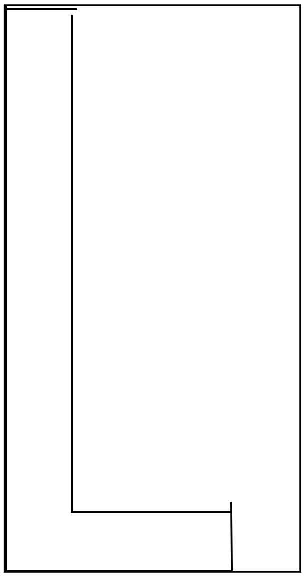


Figure 4.8: The Edge Change of the image in the Figure 4.6

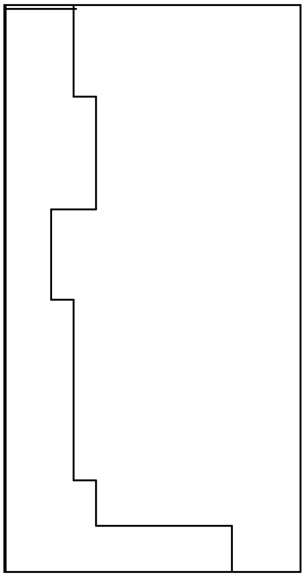


Figure 4.9: The Edge Change of the image in the Figure 4.7

Character's lines, which were prepared using MSPaint, are straight and clear. However, the lines might not be clear when the character which was generated from the license plate images is converted into binary system. This situation is shown as an example in figure 4.5 and 4.9. In this case the threshold value is used for understanding the direction of change in character.

For example, character "A" in figure 4.5 has emerged in the form of stairs when you look at the picture in the left side. This illustration of the direction (left, straight, left, straight) of the character; left is proceeding as ordered in a way. These values are very different from the value of the character "A" which is stored in the database.Hence, character recognition is very difficult. The character "A" which changes in figure 4.5 could not be recognized because it is not similar to character "A" which changes in the database. To resolve this situation difference value between the coordinates is used. Movement in the direction of the characters were not immediately decided if this value is smaller than the threshold value. These unstable changes in the character are throwen into a building as a stack. The difference between the coordinates or characters till the end of the stack has continued to take action until it exceeds the threshold value. Coordinates for the difference between the values exceeding the threshold value or coming to the end of the character must be decided for the cases which put in the stack. The difference between the first element coordinate of the stack and the last element coordinate of the stack is considered. If the scanning process is from top to bottom and from bottom to top, then the difference between the line coordinates is to be obtained; and if the scanning process is from left to right and from right to left, then the difference between the column coordinates is to be obtained. If difference value exceeds the threshold value, then the movement direction is determined depending on the difference sign. (If the sign of the difference is positive the movement of direction is scanned from top to bottom and from bottom to top. If Difference of the sign is negative then the movement direction is down. If the sign of the difference is positive, the movement of direction is right scanned from left to right and from right to left. Movement direction is towars left if it is a negative sign) By doing this, change of the edge of the "A" character in figure 4.5 can be made similar to change of the edge of the character "A" as in figure 4.4. As a result, the character's movement is close to 100% in the left direction decision. In the same way the unstable situation for the character "L" in figure 4.9 is decided according to the direction of the motion.

Another problem encountered in this step is about how the threshold value is calculated here. When the threshold value is selected as fixed, the results were not successful. Therefore, the threshold value is calculated dynamically depending on the length and width of the character. Thus, more successful results have been obtained.

A different threshold value is used to for a sudden change in the direction of characters. For example, a sudden movement towards left is observed while scanning from right to left as in figure 4.10 for the character “6”. In order to see if the movement direction is towards left, the difference between the columns of the coordinates which are in between two points is obtained.The direction of movement is determined when the difference is greater than the threshold value depending on the sign of the difference. In Figure 4.11 jumping points of the character "T" are shown.

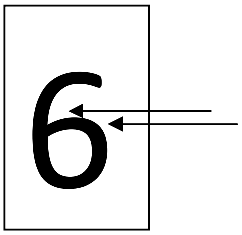


Figure 4.10: Jumping Points for Character "6"

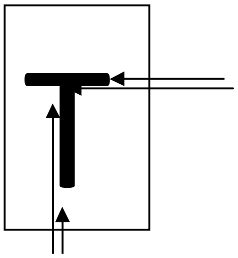


Figure 4.11: Jumping Points for Character "T"

Points are given for each simulation step for character recognition. Jumping points are important for their being distinguishing features for recognition of the character. Therefore, these points are given more points. However, the jumping points of the characters to be recognized and the jumping points of the same characters which are in the database may not be the same. Hence, the score is increased depending on closeness of jump coordinates to the coordinates of the database. The further it gets from the jumping point, the less score it would receive. After a certain distance, a low score is given. The following are some examples for this code.

```
if (tramp[0]-trampValue<position && tramp[0]+trampValue > position)
        matchedTramp = new BigDecimal(25);
else if (tramp[0]-trampValue*2<position &&
        tramp[0]+trampValue*2 > position)
        matchedTramp = new BigDecimal(15);
else
        matchedTramp = new BigDecimal(10);
```

4.6 Creating Properties of Character Classes

A data set of characters has been created by using plate images in Turkey. The feature classes were created for each character for this data set. This feature classes

were developed by using data-base classes Feature classes were created for the trial to recognize the characters. Created feature classes are compared to previously created database class. By doing so, each the character's rate is calculated for trying in order to recognize a character. Similar rates are shown on the screen to the user.

Characters are scanned from left to right, right to left, top to bottom, bottom to top. The characters' jumping points, left and right changes are determined when scanning from left to right and from right to left. The characters' jumping points, left and right changes are determined when scanning from bottom to top and from top to bottom. Each of the change rate is calculated for these changes and were evaluated over the score of a hundred. This method is used in character recognition because this change will be different for each character in mind. Below, a sample feature class code is shown for character "7".

```
ch = new Char();
      ch.setCharName('7');
      motionMap = new HashMap();
      motionMap.put(Direction.LEFT, new int [][]
            {{12, 100}});
      motionMap.put(Direction.DIRECT, new int [][]
            {{0, 12}});
      motionMap.put(Direction.RIGHT, null);
      motionMap.put(Direction.NONE, null);
      motionMap.put(Direction.TRAMP, new int [][]
            {{12, Direction.RIGHT}});
      ch.setMotionPercentage(motionMap);
      leftToRightCharMap.put('7', ch);
      ch = new Char();
      ch.setCharName('7');
      motionMap = new HashMap();
      motionMap.put(Direction.LEFT, new int [][]
            {{10, 100}});
      motionMap.put(Direction.DIRECT, new int [][]
            {{0, 10}});
      motionMap.put(Direction.RIGHT, null);
      motionMap.put(Direction.NONE, null);
      motionMap.put(Direction.TRAMP, null);
      ch.setMotionPercentage(motionMap);
      rightToLeftCharMap.put('7', ch);

      ch = new Char();
      ch.setCharName('7');
      motionMap = new HashMap();
      motionMap.put(Direction.UP, null);
      motionMap.put(Direction.DIRECT, new int [][]
            {{0, 100}});
      motionMap.put(Direction.DOWN, null);
      motionMap.put(Direction.NONE, null);
      motionMap.put(Direction.TRAMP, null);
      ch.setMotionPercentage(motionMap);
      upToDownCharMap.put('7', ch);

      ch = new Char();
      ch.setCharName('7');
      motionMap = new HashMap();
      motionMap.put(Direction.UP, new int [][]
            {{38, 100}});
      motionMap.put(Direction.DIRECT, new int [][]
            {{0, 19}, {19, 38}});
      motionMap.put(Direction.DOWN, null);
      motionMap.put(Direction.NONE, null);
      motionMap.put(Direction.TRAMP, new int [][]
            {{19, Direction.DOWN}, {38, Direction.UP}});
      ch.setMotionPercentage(motionMap);
      downToUpCharMap.put('7', ch);
```

Below, sample feature class code is shown for “P” character.

```
ch = new Char();
ch.setCharName('P');
motionMap = new HashMap();
motionMap.put(Direction.LEFT, null);
motionMap.put(Direction.DIRECT, new int [][]
      {{0, 100}});
motionMap.put(Direction.RIGHT, null);
motionMap.put(Direction.NONE, null);
motionMap.put(Direction.TRAMP, null);
ch.setMotionPercentage(motionMap);
leftToRightCharMap.put('P', ch);

ch = new Char();
ch.setCharName('P');
motionMap = new HashMap();
motionMap.put(Direction.LEFT, new int [][]
      {{36, 60}});
motionMap.put(Direction.DIRECT, new int [][]
      {{23, 36}, {61, 100}});
motionMap.put(Direction.RIGHT, new int [][]
      {{0, 23}});
motionMap.put(Direction.NONE, null);
motionMap.put(Direction.TRAMP, new int [][]
      {{61, Direction.LEFT}});
ch.setMotionPercentage(motionMap);
rightToLeftCharMap.put('P', ch);

ch = new Char();
ch.setCharName('P');
motionMap = new HashMap();
motionMap.put(Direction.UP, null);
motionMap.put(Direction.DIRECT, new int [][]
      {{0, 66}});
motionMap.put(Direction.DOWN, new int [][]
      {{66, 100}});
motionMap.put(Direction.NONE, null);
motionMap.put(Direction.TRAMP, null);
ch.setMotionPercentage(motionMap);
upToDownCharMap.put('P', ch);

ch = new Char();
ch.setCharName('P');
motionMap = new HashMap();
motionMap.put(Direction.UP, new int [][]
      {{63, 100}});
motionMap.put(Direction.DIRECT, new int [][]
      {{0, 16}, {17, 63}});
motionMap.put(Direction.DOWN, null);
motionMap.put(Direction.NONE, null);
motionMap.put(Direction.TRAMP, new int [][]
      {{17, Direction.UP}});
ch.setMotionPercentage(motionMap);
downToUpCharMap.put('P', ch);
```

The recognition algorithm is executed for each character.

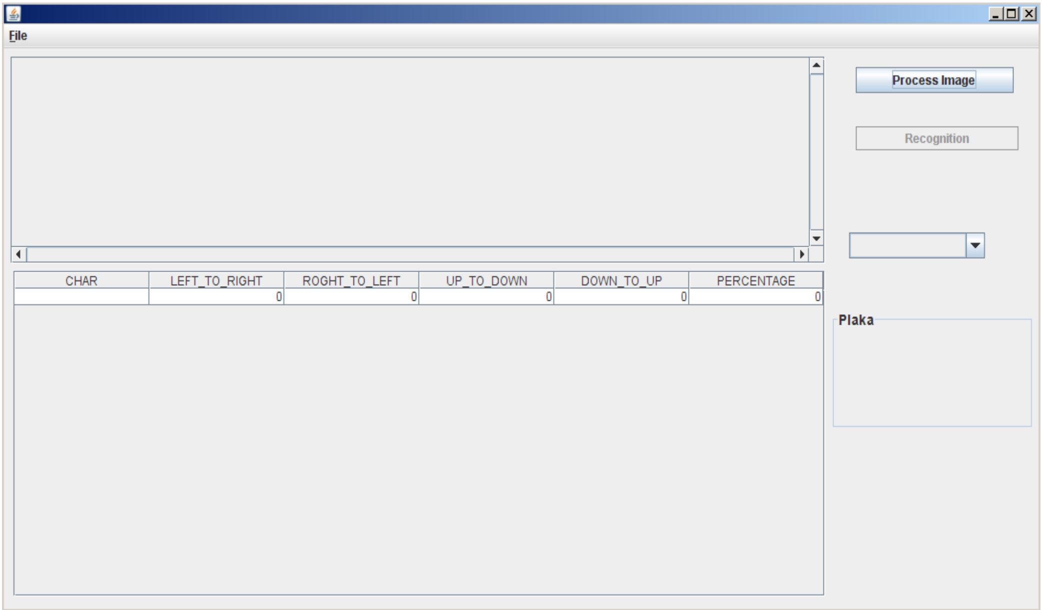
4.7 The Constraints in the Character Recognition

The plate image for the system needs to be obtained through fixed instruments, from a distance of 3-6 meters, and of a steep angle.

CHAPTER 5

SOFTWARE APPLICATION USAGE

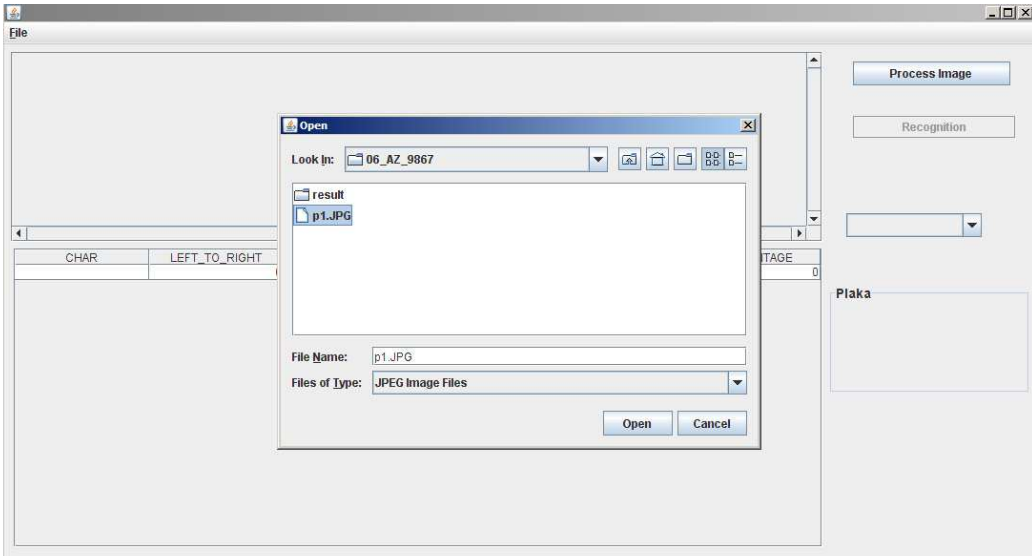
Figure 5.1 is given in the application’s main screen.



CHAR	LEFT_TO_RIGHT	ROGHT_TO_LEFT	UP_TO_DOWN	DOWN_TO_UP	PERCENTAGE
	0	0	0	0	0

Figure 5.1: Application’s Main Screen

The display opens in figure 5.2 when the "Open" option under the "File" menu is selected. The plate image is selected from this screen.



Look In: 06_AZ_9867

File Name: p1.JPG

Files of Type: JPEG Image Files

Figure 5.2: Application's "Open" Dialog Screen

A pre-processing is done for the selected plate image by clicking the button “Process Image”.Processing results are shown in figure 5.3.

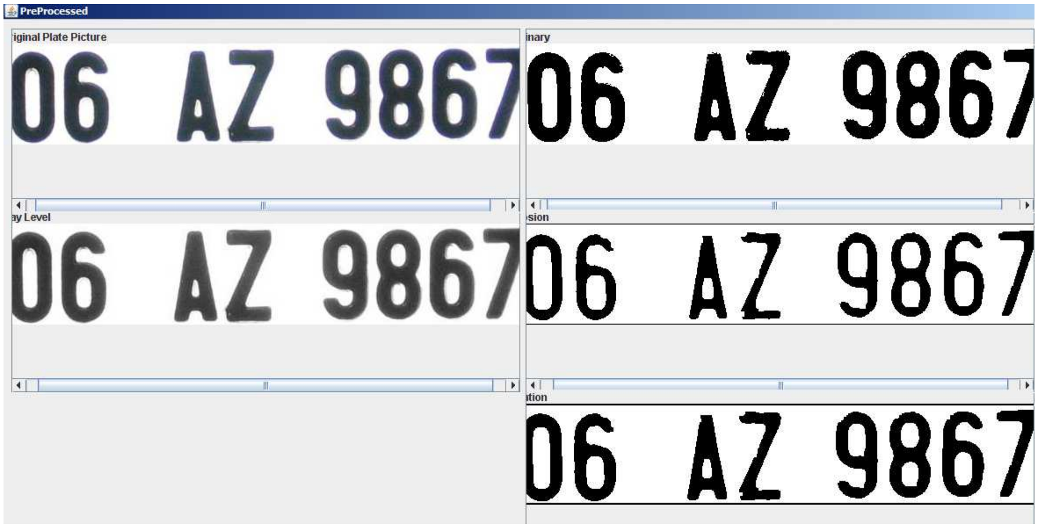
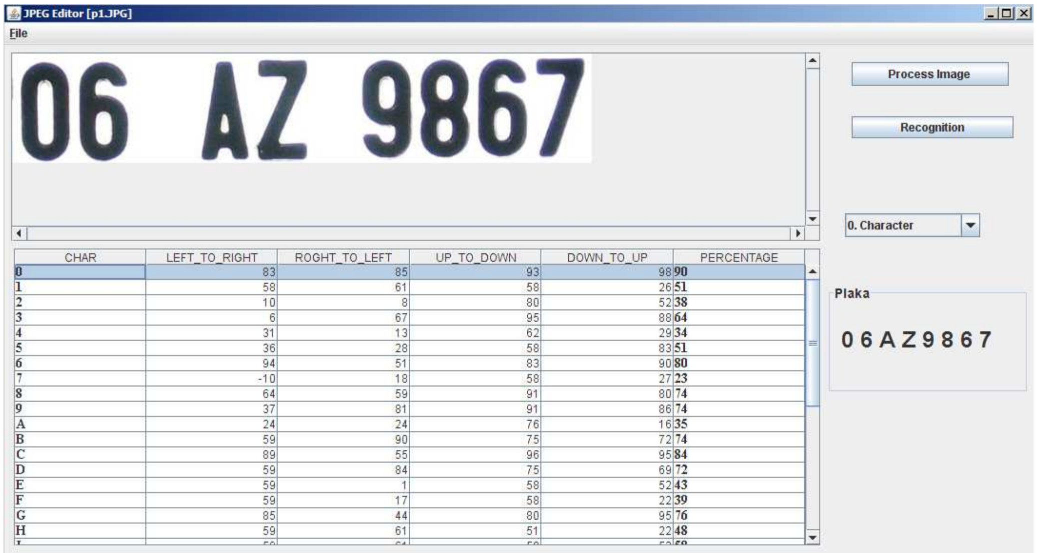


Figure 5.3: Results of Pre-Screening Process

The character recognition algorithm is executed by clicking the "Recognition" button. The character which is desired to be displayed from the combo is selected. The results are listed on the screen as in figure 5.4 for the selected character.

The most similar character set of the plate is displayed on the screen as in figure 5.4.



CHAR	LEFT_TO_RIGHT	ROGHT_TO_LEFT	UP_TO_DOWN	DOWN_TO_UP	PERCENTAGE
0	83	85	93	98	**90**
1	58	61	58	26	**51**
2	10	8	80	52	**38**
3	6	67	95	88	**64**
4	31	13	62	29	**34**
5	36	28	58	83	**51**
6	94	51	83	90	**80**
7	-10	18	58	27	**23**
8	64	59	91	80	**74**
9	37	81	91	86	**74**
A	24	24	76	16	**35**
B	59	90	75	72	**74**
C	89	55	96	95	**84**
D	59	84	75	69	**72**
E	59	1	58	52	**43**
F	59	17	58	22	**39**
G	85	44	80	95	**76**
H	59	61	51	22	**48**

Plaka

06AZ9867

Figure 5.4: Character Recognition Results on the Display Screen

Application software was developed in Java platform. Developing software version 1.6 JRE, Eclipse version 3.1 is used. Developing application software IBM brand a desktop computer with Pentium 4 3.0 GHz, 1.00 GB RAM, and a laptop computer HP Pavilion brand and which have Ghz 1.66 GB of RAM 1.00 were used.

CHAPTER 6

THE TESTS AND RESULTS

Application software has been tested on a desktop computer and laptop with the configurations, respectively, IBM Pentium 4 3.0 GHz, 1.00 GB RAM, HP Pavilion 1.66 GHz, 1.00 GB RAM. 30 plates images are used to test the software application. Test results are depicted in Table 6.1

Table 6.1: Detailed Plate Test Results

Tested Plates	**1.**	**2.**	**3.**	**4.**	**5.**	**6.**	**7.**	**8.**
06_AF_0686	0	6	A	F	0	6	8	6

06_AS_9912	0	6	A	6(%74)
				S(%56)
					9	9	1	2

06_AZ_9867	0	6	A	Z	9	8	6	7
06_BF_5784	0	6	B	F	5	7	8	4
06_BH_6831	0	6	B	H	6	8	3	1
06_BJ_1296	0	6	B	J	1	2	9	6
06_BK_9103	0	6	B	K	9	1	0	3

06_BL_4654	0	0(%62)
		6(%54)
			B	L	4	6	5	4

06_BM_1599	0	6	B	M	1	5	9	9

06_BM_2984	0	6	B	M	2	9	0(%59)
							8(%56)
								4

06_BV_3753	0	6	B	V	3	7	5	3

06_GHB_36	0	6	C(%75)
			G(%72)
				H	B	9(%68)
						3 (%66)
							6	

06_GPE_79	0	6	G	P	E	7	9	

06_GUU_16	0	6	C(%75)
			G(%74)
				U	U	E(%59)
						1(%31)
							6	

06_MVZ_78	6(%67)
	0(%66)
		6	M	V	Z	7	8	

06_S_7151	0	6	C(%69)
			S(%60)
				7	1	5	1	

06_VLE_99	0	6	V	L	F(%88)
					E(%87)
						9	9	

34_EP_0921	9(%89)
	3(%77)
		4	E	P	0	9	2	1

34_ET_6596	9(%)
	3(%74)
		4	F(%94)
			E(%91)
				T	6	5	9	6

34_FF_1597	9(%82)
	3(%80)
		4	F	F	1	5	9	7

06_BJ_9425	0	0(%78)
		6(%75)
			B	J	9	4	2	5

06_BS_092	0	6	B	6(%63)
				S(%58)
					0	9	7(%43)
							2(%40)	

06_E_9229	0	6	E	9	2	2	0(%75)
							9(%74)	

06_HR_860	0	6	H	R	8	6	0	

06_M_9293	0	6	M	9	2	9	9(%72)
							3(%67)	

06_RGS_81	0	6	R	6(%88)
				G(%87)
					6(%75)
					S(%64)
						8	1	

06_UB_258	0	6	U	B	2	5	8	

06_ZS_877	0	6	Z	6(%77)
				S(%64)
					8	7	7	

35_SZ_012	9(%68)
	3(%65)
		5	S	Z	0	1	2	

71_DR_278	7	1	D	R	2	7	2(%55)
							8(%54)	

Test results indicate that results for the characters “B” and “D” are very similar. It may be stated that character recognition algorithm ignores the curve in the middle while processing the character from the left side. The reason of this ignoring is this curve’s being very tiny in real plate screens. What is more, algorithm assumes this curve as a noise and handle it as being straight.

Due to small differences between the upper and lower parts of the “D” character, it couldn’t be recognized and this caused misidentification of the character. Very similar results were encountered for the characters “6”, “G”, “0” and occasionally these characters were mixed with each other. Identification rate of the characters “5” and “S”, “2” and “Z”, “I” and “1”, “U” and “H” were found very close. Due to this, in some occasions, they couldn’t be distinguished from one another.

Plates in Turkey should follow the rule that first and the second two characters must be numbers. The last two characters have to be elements of the numeric system. Accordingly, third character has to be one of the characters between A-Z. This rule has also been implemented in the plate recognition program.By doing so, conditions in which cause the confusion of “I” and “1”, “2” and “Z”, “6” and “G”, “5” and “S” could be diminished.

In situations where changes on the characters “B” and “D” are not realized, a property class which is constructed for those characters seems to be similar to that of the character “I”. For this reason, those characters were identified as being “I”. In order be able to resolve this problem, length and width properties of the character “I” were utilized. It has been observed that while heigth/width ratio of the “I” character is 3, this figure is aproximately 2 for the “B”, “D” and other characters. Thus, depending on that property, in situations where the character “I” looked very similar to the characters “B” and “D” , similarity rate has been decreased.

As surface changes of the characters were considered in the algorithm, it was observed that noises within the character had no main effect on the overall identification. Figure 6.1 examplifies this issue through the characters “H” and “I” which have noise within the character borders.

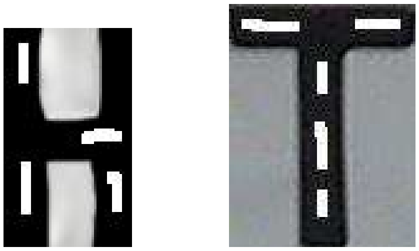


Figure 6.1: Noises on the Characters “H” and “T”

In cases where small changes related to the characters were ignored, particular characters were misidentified. However, conversely, considering small changes as important did also cause misidentification. (due to the blurredness of the characters, stair-shaped changes.) In Table 6.2, correct identification rates for the characters on the tested plate are presented.

Table 6.2: Plate Test Results

	Number of Tests	**Number of Correvt Recognition**	**Number of Wrong Recognition**	**Percentage Rate**
Character Segmentation	225	225	0	%100
Character Recognition	225	201	24	%89.33

True recognition rates for the tested characters and numbers are presented in tables 6.3 and 6.4, respectively.

Table 6.3: Results of the Characters

Character	**Trial Number**	**Correct Recognition**	**False Recognition**	**Percentage**
A	33	28	5	%84.84
B	19	15	4	%78.94
C	22	17	5	%77.27
D	12	8	4	%50
E	19	14	5	%73.68
F	35	27	8	%77.14
G	22	16	6	%72.72
H	32	29	3	%90.06
I				
J	19	15	4	%78.94
K	33	30	3	%90
L	33	27	6	%75
M	22	16	6	%72.72
N	24	19	5	%79.16
O				
P	23	17	6	%73.91
R	27	20	7	%74.07
S	16	11	5	%68.75

T	35	30	5	%85.71
U	10	8	2	%80
V	34	29	5	%85.29
Y	25	20	5	%80
Z	31	29	2	%93.54

Table 6.4: Results of the Numbers

Number	**Trial Number**	**Correct Recognition**	**False Recognition**	**Percentage**
0	36	32	4	%88.88
1	26	20	6	%76.92
2	31	25	6	%83.87
3	18	15	3	%83.33
4	33	31	2	%93.93
5	40	35	5	%87.5
6	48	43	5	%89.58
7	37	34	3	%91.89
8				
9	28	24	4	%85.71

CHAPTER 7

DISCUSSION AND CONCLUSION

This study improves different and simple algorithms among the neural network, support vector and the other systems. This algorithm can be improved in the future since the algorithm can easily be understood. If this algorithm is improved, then the correction rate will be higher. Neural network, support vector machine and other systems have complex mathematical algorithms. Then improving and understanding is difficult for these algorithms.

We can create three types character database. First is for small size charater, second is for normal size character and last one is for big size character. Then when the program detects the character, program can be shift to database automaticly which one is appropriate for detected character size. Then we can improve the correction rate for this study.

In this study, some problematic features like distance, light and corner are restricted. In future study, can be make solution for those problems. This study is interested only Turkish plate recognition. In future study can be interesting with international plate recognition. Finding plate location from digital image does not exist in this study. In future study it can be implemented also.

REFERENCES

[1] A hierarchically combined classifier for license plate recognition Lihong Zheng; Xiangjian He; Qiang Wu; Wenjing Jia; Samali, B.; Palaniswami, M.; Computer and Information Technology, 2008. CIT 2008. 8th IEEE International Conference on 8-11 July 2008 Page(s):372 - 377 Digital Object Identifier 10.1109/CIT.2008.4594704

[2] A high performance license plate recognition system based on the web technique Dai Yan; Ma Hongqing; Liu Jilin; Li Langang; Intelligent Transportation Systems, 2001. Proceedings. 2001 IEEE 25-29 Aug. 2001 Page(s):325 - 329 Digital Object Identifier 10.1109/ITSC.2001.948677

[3] A Macao license plate recognition system Cheokman Wu; Lei Chan On; Chan Hon Weng; Tong Sio Kuan; Kengchung Ng; Machine Learning and Cybernetics, 2005. Proceedings of 2005 International Conference on Volume 7, 18-21 Aug. 2005 Page(s):4506 - 4510 Vol. 7 Digital Object Identifier 10.1109/ICMLC.2005.1527732

[4] A Simple Method for Chinese License Plate Recognition Based on Support Vector Machine Xiaojun Chi; Junyu Dong; Aihua Liu; Huiyu Zhou; Communications, Circuits and Systems Proceedings, 2006 International Conference on Volume 3, 25-28 June 2006 Page(s):2141 - 2145 Digital Object Identifier 10.1109/ICCCAS.2006.284922

[5] A versatile low-cost car plate recognition system Lopez, J.M.; Gonzalez, J.; Galindo, C.; Cabello, J.; Signal Processing and Its Applications, 2007. ISSPA 2007. 9th International Symposium on 12-15 Feb. 2007 Page(s):1 - 4 Digital Object Identifier 10.1109/ISSPA.2007.4555412

[6] An Algorithm for License Plate Recognition Using Radial Basis Function Neural Network Bo Li; Zhi-yuan Zeng; Jian-zhong Zhou; Hua-li Dong; Computer Science and Computational Technology, 2008. ISCSCT '08. International Symposium on

Volume 1, 20-22 Dec. 2008 Page(s):569 - 572 Digital Object Identifier 10.1109/ISCSCT.2008.272

[7] Plate detection and recognition by using color information and ANN, Ziya TELATAR, Emre ÇAMAŞIRCIOĞLU, Electronic Engineering Department, Ankara University, ANKARA, 2007

[8] Distance and Color Invariant Automatic License Plate Recognition System Khan, N.Y.; Imran, A.S.; Ali, N.; Emerging Technologies, 2007. ICET 2007. International Conference on 12-13 Nov. 2007 Page(s):232 - 237 Digital Object Identifier 10.1109/ICET.2007.4516349

[9] Car license plate recognition through Hausdorff distance technique Juntanasub, R.; Sureerattanan, N.; Tools with Artificial Intelligence, 2005. ICTAI 05. 17th IEEE International Conference on 16-16 Nov. 2005 Page(s):5 pp. Digital Object Identifier 10.1109/ICTAI.2005.46

[10] Character recognition using parallel BP neural network Feng Yang; Fan Yang; Audio, Language and Image Processing, 2008. ICALIP 2008. International Conference on 7-9 July 2008 Page(s):1595 - 1599 Digital Object Identifier 10.1109/ICALIP.2008.4590169

[11] Chinese License Plate Recognition Using a Convolutional Neural Network Zhihong Zhao; Shaopu Yang; Xinna Ma; Computational Intelligence and Industrial Application, 2008. PACIIA '08. Pacific-Asia Workshop on Volume 1, 19-20 Dec. 2008 Page(s):27 - 30 Digital Object Identifier 10.1109/PACIIA.2008.196

[12] Color edge enhancement based fuzzy segmentation of license plates Syed, Y.A.; Sarfraz, M.; Information Visualisation, 2005. Proceedings. Ninth International Conference on 6-8 July 2005 Page(s):227 - 232 Digital Object Identifier 10.1109/IV.2005.33

[13] Comparison of Feature Extractors in License Plate Recognition Abdullah, Siti Norul Huda Sheikh; Khalid, Marzuki; Yusof, Rubiyah; Omar, Khairuddin;

Modelling & Simulation, 2007. AMS '07. First Asia International Conference on 27-30 March 2007 Page(s):502 - 506 Digital Object Identifier 10.1109/AMS.2007.25

[14] Efficient Embedded Neural-Network-Based License Plate Recognition System Caner, H.; Gecim, H. S.; Alkar, A. Z.; Vehicular Technology, IEEE Transactions on Volume 57, Issue 5, Sept. 2008 Page(s):2675 – 2683 Digital Object Identifier 10.1109/TVT.2008.915524

[15] Graphical Models for Joint Segmentation and Recognition of License Plate Characters Xin Fan; Guoliang Fan; Signal Processing Letters, IEEEVolume 16, Issue 1, Jan. 2009 Page(s):10 – 13 Digital Object Identifier 10.1109/LSP.2008.2008486

[16] License plate character recognition algorithm based on filled function method training bp neural network Ying Zhang; Yingtao Xu; Gejian Ding; Control and Decision Conference, 2008. CCDC 2008. Chinese 2-4 July 2008 Page(s):3886 - 3891 Digital Object Identifier 10.1109/CCDC.2008.4598060

[17] Generative Models for License Plate Recognition by using a Limited Number of Training Samples Mecocci, A.; Tommaso, C.; Image Processing, 2006 IEEE International Conference on 8-11 Oct. 2006 Page(s):2769 - 2772 Digital Object Identifier 10.1109/ICIP.2006.313121

[18] License plate recognition system based on morphology and LS-SVM Haiyan Zhao; Chuyi Song; Haili Zhao; Shizheng Zhang; Granular Computing, 2008. GrC 2008. IEEE International Conference on 26-28 Aug. 2008 Page(s):826 - 829 Digital Object Identifier 10.1109/GRC.2008.4664738

[19] New approaches for license plate recognition system Nathan, V.S.L.; Ramkumar, J.; Kamakshi Priya, S.; Intelligent Sensing and Information Processing, 2004. Proceedings of International Conference on 2004 Page(s):149 - 152 Digital Object Identifier 10.1109/ICISIP.2004.1287642

[20] Research and Realization of Improved Pattern Matching in License Plate Recognition Ping Wang; Wei Zhang; Intelligent Information Technology

Application Workshops, 2008. IITAW '08. International Symposium on 21-22 Dec. 2008 Page(s):1089 – 1092 Digital Object Identifier 10.1109/IITA.Workshops.2008.108

[21] Robust license plate recognition based on dynamic projection warping Tsang-Hong Wang; Feng-Chou Ni; Keh-Tsong Li; Yon-Ping Chen; Networking, Sensing and Control, 2004 IEEE International Conference on Volume 2, 2004 Page(s):784 - 788 Vol.2 Digital Object Identifier 10.1109/ICNSC.2004.1297046

[22] Segmentation of characters on car license plates Xiangjian He; Lihong Zheng; Qiang Wu; Wenjing Jia; Samali, B.; Palaniswami, M.; Multimedia Signal Processing, 2008 IEEE 10th Workshop on 8-10 Oct. 2008 Page(s):399 - 402 Digital Object Identifier 10.1109/MMSP.2008.4665111

[23] The Performance Improvement of License Plate Recognition System Using Composite Color and Artificial Hippocampus Algorithm Hye-Youn Lim; Young-Do Lim; Dae-Seong Kang; Future Generation Communication and Networking, 2008. FGCN '08. Second International Conference on Volume 2, 13-15 Dec. 2008 Page(s):82 - 85 Digital Object Identifier 10.1109/FGCN.2008.135

[24] (http://www.yildiz.edu.tr/~bayram/sgi/saygi.htm (26/05/2009))

[25] (http://www.emo.org.tr/ekler/fbce6c74ff376d1_ek.pdf) (27/05/2009)

[26] (http://4uzbk.sdu.edu.tr/4UZBK/HYB/4UZBK_025.pdf)(27/05/2009)

[27] (http://www.csharpnedir.com/makalegoster.asp?MId=595) (29/05/2009)

[28](http://mennan.kagitkalem.com/ImageProcessingThresholdingOtsuMethod.aspx) (29/05/2009)

[29] (http://www.generation5.org/content/2003/segmentation.asp) (30/05/2009)

[30] (coursemain.ee.ukzn.ac.za/enel4iph2/notes/Lec_11.ppt)(28/05/2009)