A

DISSERTATION

ON

AUTOMATED NUMBER PLATE RECOGNITION SYSTEM

SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF ENGINEERING

(ELECTRONICS & COMMUNICATION ENGINEERING)

SUBMITTED BY:

URVASHI B. JAIN`

College Roll No: 04/EC/PT/2009

University Roll No: 13923

UNDER THE GUIDANCE OF:

MS. N. JAYANTHI

(ASST. PROFESSOR)

DEPARTMENT OF

ELECTRONICS & COMMUNICATION ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

DEPARTMENT OF

ELECTRONICS & COMMUNICATION ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

DELHI

2010-2012

A

DISSERTATION

ON

AUTOMATED NUMBER PLATE RECOGNITION SYSTEM

SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF ENGINEERING

(ELECTRONICS & COMMUNICATION ENGINEERING)

SUBMITTED BY:

URVASHI B. JAIN`

College Roll No: 04/EC/PT/2009

University Roll No: 13923

UNDER THE GUIDANCE OF:

MS. N. JAYANTHI

(ASST. PROFESSOR)

DEPARTMENT OF

ELECTRONICS & COMMUNICATION ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

DEPARTMENT OF

ELECTRONICS & COMMUNICATION ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

DELHI

2010-2012

CERTIFICATE

Certified that the thesis work entitled ‘AUTOMATED NUMBER PLATE RECOGNITION SYSTEM’ is a bonafide work carried out by Urvashi B. Jain (University Roll No. 13923) in partial fulfillment for the award of degree of Master of Engineering Degree in Electronics & Communication Engineering of Delhi Technological University during the session 2009-12. The project has been approved as it satisfied the academic requirements in respect of thesis work prescribed for the Master of Engineering degree.

Signature of the Guide
Ms. N. JAYANTHI
(Asst. Professor)
Department of Electronics & Communication Engineering
Delhi Technological University

ACKNOWLEDGEMENT

It gives me a great pleasure to express my profound gratitude to my project guide Ms. N. Jayanthi (Asst. Professor), Department of Electronics & Communication Engineering, Delhi Technological University, for her valuable and inspiring guidance throughout the progress of this project.

At the same time, I would like to extend my heartfelt thanks to Mr. Rajeev Kapoor, Head of the department, Department of Electronics & Communication Engineering, Delhi Technological University, for keeping the spirits high and clearing the visions to work on the project.

Urvashi B. Jain
Roll No: 13923

ABSTRACT

The strategy of the project is to design automatic system for recognizing the number plate of the car from image taken by the traffic surveillance system. Thus, the project is to investigate and construct an application whereby the system will recognize the car number plate from the image captured of the car. The system will be based on a personal computer and software packages available, MATLAB program and a digital camera, which help to capture the images of the car.

The general algorithm involves the following steps:

- Ø Image Processing: The image captured is preprocessed and reduction in the contrast.
- Ø Conversion of RGB image to Gray image: Processed image is converted into a gray image.
- Ø Erosion and Dilation: Unwanted noise is removed.
- Ø Localization of number plate using Histogram approach: Thresholding is applied to both the row and column difference vectors obtained.
- Ø Smoothening and Filtering: Unwanted rows and columns are removed using averaging method.
- Ø Final extraction of number plate: Final number plate is extracted by selecting the area having maximum intensity variation between pixels and having maximum row and column sum of pixel intensity.
- Ø Number plate recognition based on OCR method.

Table of Content

List of Figures

1 Introduction

1.1 Project Background

Automatic Number Plate Recognition is an image processing system whereby it is used to recognize number plate of the car from the image of the car. It is basically used for traffic and security purposes.

<u>How the Vehicle License Plate System works</u>

Firstly, the image of the car would be captured with the help of the camera. Secondly, the RGB image of the car would be converted into gray image to enhance the image by performing processes of erosion and dilation on it to remove any noise present. The system will read the pixel information of the car and run the recognition process. Thirdly, the system will apply certain algorithm to analyses the car image like histogram based approach. Localization of the car number plate position would be implemented by plotting both the vertical and horizontal histograms and reading the difference in the pixels value and storing them in a row and column vector. Lastly, concepts of thresholding and averaging would be applied to filter out the unwanted rows and columns and extracting out the number plate.

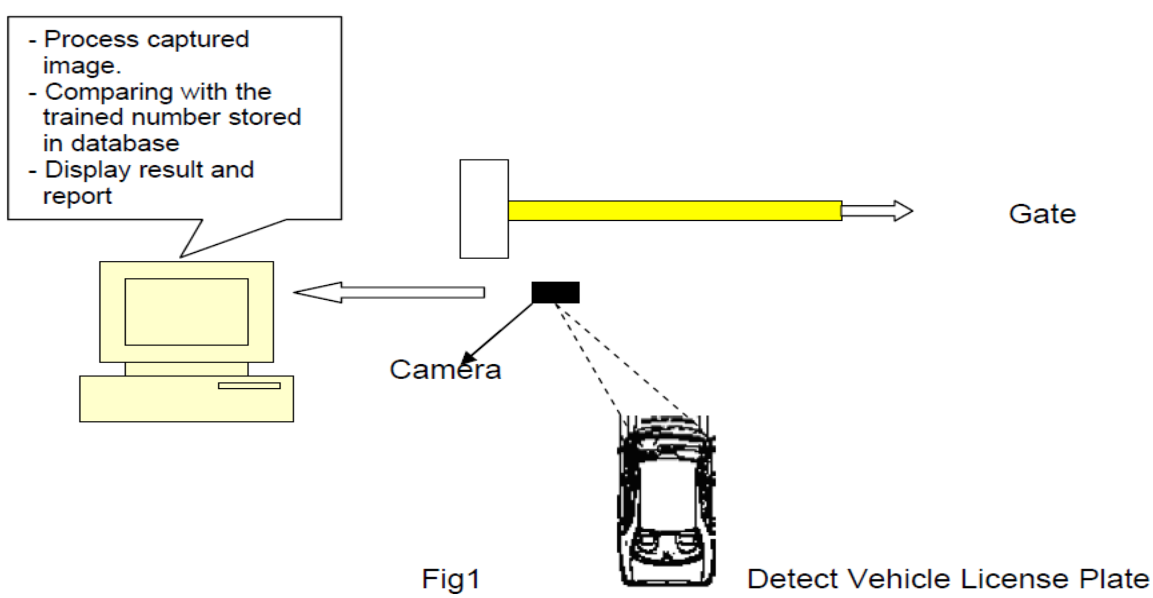


Figure 1 How the Vehicle License Plate System works

Besides, the Automatic Number Plate Recognition also provides an advantage by keeping the image of the car in which it will be useful for crime fighting. There are difficulties for Automatic Number Plate Recognition in which it will affect the efficiency and accuracy of the system. It is essential and important to determine the facts which will able to influence the operations and recognition proficiency. Next, we also need to look into other facts of variables that are not constant. Below are the non-constant variables which will affect the accuracy of recognition:

- Surroundings
- Distance between car and the camera
- Type of plate(rectangular, bent type)
- Number plate orientation
- Type of car font characters

1.2 Objectives

The overall objective of the project is to develop a system to recognize the number plate of the car from the image of the car. The software could lead to a cheaper and faster way of enhancing and determined the performance of the recognition system. The system will be based on a Personal Computer such that it will generate report on the vehicle license plate it has extracted. Besides, the system can also serve as a security purpose whereby it can spot on any wanted or stolen vehicles. In the past, there has been similar project implemented but had poor accuracy. Thus, we would need to improve or rewrite the algorithm to improve the accuracy. We will set of constraints and focus on the design of the algorithm to extract the car number plate in order to improve the accuracy. Following were the main objectives of the project:

1.2.1 Recognition of only characters present in the number plate

The input image should contain only the image of the car. The reason for this constraint is that the algorithm is based on maximum intensity variation present in only the number plate area of the car and the presence of surroundings should not disturb the effective computation of the algorithm.

1.2.2 Characters should not be overlapping

The characters should not be touching each other .Though segmentation of touching characters has been one of the toughest jobs in text recognition and this alone has remained one of the toughest area of research.

1.2.3 Image should be noise free

Another assumption about the input image of the car should be noise free. This assumption does not reduce the complexity of the problem as this is just a part of preprocessing module.

A noisy image can be made noise free by applying standard functions and techniques. These were skipped due to time constraints involved

There is definitely a lot more room for further improvement on this project. However, due to the limited time frame given, it is not advisable for me to cover all aspects in this project. Thus, in discussion with my tutor, we managed to come up with the progress guideline as to what need to be included in this project.

2 Main text and discussion

2.1 Introduction to Matlab

MATLAB is chosen as the main development for the license plate recognition application. The task is to construct the algorithm and recognizes can be done using MATLAB.

Besides, MATLAB is also very efficient as it have built-in-function tools for neural network and image processing.

The advantages of MATLAB can be classified as follows: Platform independence, predefined function and device-independent plotting, Graphical User interface and MATLAB compiler.

MATLAB reads images using the function “imread”. The table below show images/graphs format that is supported by “imread” with MATLAB.

Format Name	Description	Extension
TIFF	Tagged Image File Format	.tif ; tiff
JPEG	Joint Photograph Experts Group	.Jpg ; .jpeg
GIF	Graphic Interchange Format	.gif
BMP	Windows Bitmap	.bmp
PNG	Portable Network Graphics	.png
XWD	X Window dump	.xwd

<u>Advantage of Matlab</u>

- Data can be represented in terms of matrices or vectors.
- MATLAB coding is shorter and simpler
- Algorithm used in MATLAB can be converted to use in hardware.
- The system will perform faster when using “Matrix Approach”.

- MATLAB possesses power graphic visualization tools.

2.2 Overview of the Number Plate Recognition

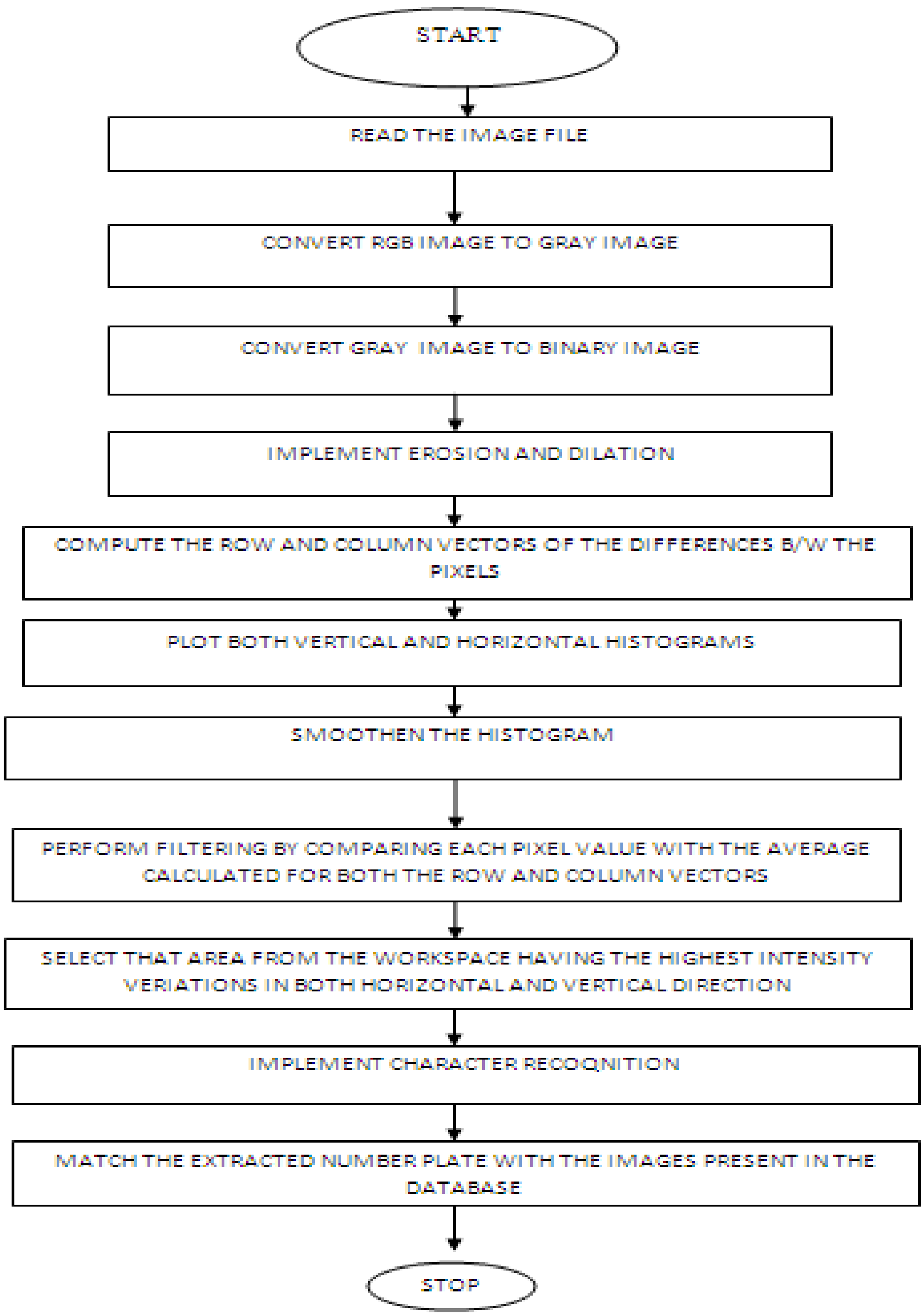


Figure 2 Overview of the Number Plate Recognition

2.3 Existing systems

ANPR systems have been implemented in many countries like Australia, Korea and few others .Strict implementation of license plate standards in these countries has helped the early development of ANPR systems. These systems use standard features of the license plates such as: dimensions of plate, border for the plate, color and font of characters, etc. help to localize the number plate easily and identify the license number of the vehicle. In India, number plate standards are rarely followed. Wide variations are found in terms of font types, script, size, placement and color of the number plates. In few cases, other unwanted decorations are present on the number plate. Also, unlike other countries, no special features are available on Indian number plates to ease their recognition process. Hence, currently most of the systems are manual recording systems in India.

<u>Format of the Indian License Plate</u>

Format of the registration is as shown below:

AA 11 BB 1111

Where AA is the two letter state code; 11 is the two digit district code; 1111 is the unique license plate number and BB are the optional alphabets if the 9999 numbers are used up. An example would be:

TN 01 CA 1003

The first two alphabets TN indicate that the vehicle is from the Tamil Nadu state. The next two are representing the district (In this case the capital Chennai).

CA 1003 is the unique license plate number. In some states (such as the union territory of Delhi) the initial 0 of the district code is omitted; thus Delhi district 2 numbers appear as DL 2 not DL 02. The National Capital Territory of Delhi has an additional code in the registration code:

DL 11 C AA 1111

Where DL is the two letter code for Delhi (DL), the additional C (for category of vehicle) is the letter 'S' for two-wheelers, 'C' for cars, 'P' for public passenger vehicles such as buses, 'R' for three-wheeled rickshaws, 'T' for tourist licensed vehicles and taxis, 'V' for pick-up trucks and vans and 'Y' for hire vehicles. Thus a Delhi-specific example is:

DL 5 S AB 9876

<u>License Plate Specifications</u>

The License plate specifications in India include the dimension of the plate (Length*Breadth), size of characters with its height and width and the spacing between the characters present in it. The specifications differ for various kinds of vehicles like cars, buses, two wheelers etc. All those driving motor vehicles with registration number plates not conforming to the specifications will be penalized.

So vehicle's registration number plate should be checked and ensured that they are in accordance with the specifications (see Figure 3).

Figure 3 The Right Indian License plate

In case of two lines, the state code and registering authority code will form the first line and the rest will form the second line, one below the other. All the letters should be in English and numerals / numbers should be in Arabic e.g. DL2CA 1234.

3 Digital Image Processing

3.1 What is an image?

An image may be defined as a two-dimensional function, f(x, y), where x and y are spatial (plane) coordinates, and the amplitude of f at any pair of coordinates (x, y) is called the intensity or gray level of the image at that point. When x, y, and the amplitude values of f are all finite, discrete quantities, we call the image a digital image. The field of digital image processing refers to processing digital images by means of a digital computer. Note that a digital image is composed of a finite number of elements, each of which has a particular location and value. These elements are referred to as picture elements, image elements, pels, and pixels. Pixel is the term most widely used to denote the elements of a digital image.

- An image refers to a 2D light intensity function f(x,y), where (x,y) denote spatial coordinates and the value of f at any point (x,y) is proportional to the brightness or gray levels of the image at that point.
- A digital image is an image f(x,y) that has been discretized both in spatial coordinates and brightness
- The elements of such a digital array are called image elements or pixels.

The Digital image processing is the use of computer algorithms to perform image processing on digital images. As a subcategory or field of digital signal processing, digital image processing has many advantages over analog image processing. It allows a much wider range of algorithms to

3.2 Area of applications of DIP

Image processing has an enormous range of applications; almost every area of science and technology can make use of image processing methods. Here is a short list just to give some indication of the range of image processing applications.

i. Medicine
   - Inspection and interpretation of images obtained from X-rays, MRI or CAT scans
   - Analysis of cell images, of chromosome karyotypes.

ii. Agriculture
   - Satellite/aerial views of land, for example to determine how much land is being used for different purposes, or to investigate the suitability of different regions for different crops
   - Inspection of fruit and vegetables distinguishing good and fresh produce from old.

iii. Industry
   - Automatic inspection of items on a production line
   - Inspection of paper samples.

iv. Law enforcement
   - Fingerprint analysis
   - Sharpening or de-blurring of speed-camera images.

v. X-Ray Imaging
   - X-rays are among the oldest sources of EM radiation used for imaging. The best known use of X-rays is medical diagnostics, but they also are used extensively in industry and other areas, like astronomy.

vi. Film
   - Westworld (1973) was the first feature film to use digital image processing to pixilated photography to simulate an android's point of view.

vii. Intelligent Transportation System

- Digital image processing has wide applications in intelligent transportation systems, such as automatic number plate recognition and traffic sign recognition.

3.3 Aspects of image processing

It is convenient to subdivide different image processing algorithms into broad subclasses. There are different algorithms for different tasks and problems, and often we would like to distinguish the nature of the task at hand.

1. Image enhancement. This refers to processing an image so that the result is more suitable for a particular application. Example includes:
    - sharpening or de-blurring an out of focus image,
    - highlighting edges,
    - improving image contrast, or brightening an image,
    - Removing noise.
2. Image restoration. This may be considered as reversing the damage done to an image by a known cause, for example:
    - removing of blur caused by linear motion,
    - removal of optical distortions,
    - Removing periodic interference.
3. Image segmentation. This involves subdividing an image into constituent parts, or isolating certain aspects of an image:
    - finding lines, circles, or particular shapes in an image,
    - In an aerial photograph, identifying cars, trees, buildings, or roads.

These classes are not disjoint; a given algorithm may be used for both image enhancement or for image restoration. However, we should be able to decide what it is that we are trying to do with our image: simply make it look better (enhancement), or removing damage (restoration).

3.4 Image Processing Task

We will look in some detail at a particular real-world task, and see how the above classes may be used to describe the various stages in performing this task. The job is to obtain, by an automatic process, the postcodes from envelopes. Here is how this may be accomplished:

1. Acquiring the image. First we need to produce a digital image from a paper envelope. This can be done using either a CCD camera, or a scanner.
2. Preprocessing. This is the step taken before the _major_ image processing task. The problem here is to perform some basic tasks in order to render the resulting image more suitable for the job to follow. In this case it may involve enhancing the contrast, removing noise, or identifying regions likely to contain the postcode.
3. Segmentation. Here is where we actually “get” the postcode; in other words we extract from the image that part of it which contains just the postcode.
4. Representation and description. These terms refer to extracting the particular features which allow us to differentiate between objects. Here we will be looking for curves, holes and corners which allow us to distinguish the different digits which constitute a postcode.
5. Recognition and interpretation. This means assigning labels to objects based on their descriptors (from the previous step), and assigning meanings to those labels. So we identify particular digits, and we interpret a string of four digits at the end of the address as the postcode.

4 Working with images in Matlab

4.1 Types of images

1. Binary: It is an image which quantatised into two values representing 0 and 1 or in pixel values of 0 and 255 representing the color black and white. Binary image is the simplest process and has applied to many other applications. It is useful as the information we need can be obtained from the silhouette of the object.
2. Grayscale: Each pixel is a shade of grey, normally from 0(black) to 256 White. This range means that each pixel can be represented by eight bits, or exactly one byte. This is a very natural range for image file handling. Other grayscale ranges are used, but generally they are a power of 2. Such images arise in medicine (X-rays), images of printed works, and indeed 256 different grey levels are sufficient for the recognition of most natural objects.
3. True color or RGB: Here each pixel has a particular color; that color being described by the amount of red, green and blue in it. If each of these components has a range 0-255, this gives a total of 2553= 16,777,216 different possible colors in the image. This is enough colors for any image. Since the total number of bits required for each pixel is 24, such images are also called 24-bit colour images. Such an image may be considered as consisting of a “stack” of three matrices; representing the red, green and blue values for each pixel. This means that for every pixel there correspond three values.
4. Indexed: Most colour images only have a small subset of the more than sixteen million possible colours. For convenience of storage and file handling, the image has an associated colour map, or colour palette, which is simply a list of all the colours used in that image. Each pixel has a value which does not give its colour (as for an RGB image), but an index to the colour in the map. It is convenient if an image has 256 colours or less, for then the index values will only require one byte each to store. Some image files formats (for

example, Compuserve GIF), allow only 256 colours or fewer in each image, for precisely this reason.

4.2 Data types

Elements in Matlab matrices may have a number of different numeric data types.

Data Type	Description	Range
int8	8-bit integer	(-128) – (+127)
uint8	8-bit unsigned integer	0 – (+255)
int16	16-bit integer	(-32768) – (+32767)
uint16	16-bit unsigned integer	0 – (65535)
double	Double precision real number	Machine specific

4.3 Conversion between different types of images

4.3.1 RGB to Gray Conversion

The RGB image is made of color pixels of an M x N x 3 array. Every pixel has three color information associated with it for every primary color. We may convert RGB images to grayscale by eliminating the hue and saturation information while retaining the luminance.

The luminance information at every pixel point can be obtained from its three primary color information

$$Y = 0.2989 * R + 0.5870 * G + 0.1140 * B$$

Luminance gives us the information about the brightness level at that point in the picture. Brightness level in a grayscale image obtained will vary from 0 (for complete black) to 255 (for complete white).

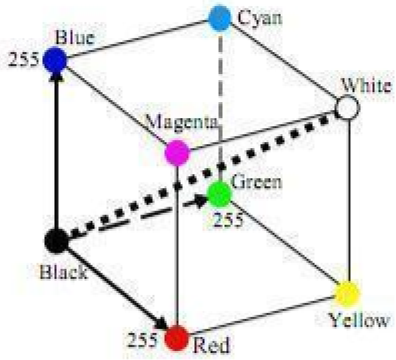


Figure 4 RGB model

4.3.2 Gray to Binary Conversion

Binary image has only two levels associated with every pixel value, i.e. Zero (0) for black and one for white. Any gray images can be converted into binary image with the help of Threshold method.

Thresholding is one of the oldest segmentation methods. The value of threshold (T) is being selected and compare with the pixel of the image. It also transforms the input image (K) into an output binary image (F) which is being segmented.

$$F(x,y) = 1 \text{ if } K(x,y) >= T;$$
$$= 0 \text{ if } K(x,y) < T$$

We can use the graythresh toolbox function to calculate the threshold value by using Otsu's method. Otsu's method works out based on the discriminate analysis. The threshold is chosen by partitioning the image pixel into two classes: C0 and C1, (Foreground and Background).

5 Dilation and erosion

5.1 Dilation

Dilation is an operation that grows or thickens objects in a binary image. The specific manner and the extent of this thickening are controlled by a shape referred to as a structuring element. Dilation is a process that translates the origin of the structuring element throughout the domain of the image and checks to see whether it overlaps with 1-valued pixels. The output image is 1 at each location of the origin of the structuring element if the structuring element overlaps at least one 1-valued pixel in the input image.

The Matlab image processing toolbox function imdilate performs dilation. The syntax is

C = imdilate (A, B)

Where A and C are binary images and B is a matrix of 0s and 1s that specify the structuring element.

5.1.1 Characteristics of Dilation:

a. Dilation generally increases the sizes of objects, filling in holes and broken areas, and connecting areas that are separated by spaces smaller than the size of the structuring element.
b. With grayscale images, dilation increases the brightness of objects by taking the neighborhood maximum when passing the structuring element over the image.
c. With binary images, dilation connects areas that are separated by spaces smaller than the structuring element and adds pixels to the perimeter of each image object.

5.1.2 The strel() Function

Matlab image processing toolbox function strel constructs structuring elements with a variety of shapes and sizes. Its basic syntax is

B = s t r e l (shape, parameters)

This returns a special quantity called an strel object. The function imdilate is able to use the decomposed from of the structuring element.

5.1.3 Example

B = s t r e l (‘diamond’, 5)

creates a flat diamond shaped structuring element such that the distance from the structuring element’s origin to the extreme points of the diamond is 5. There are other shapes such as disk, line, rectangle, etc. The following code shows how to use the structuring element generated by the strel function is used to carry out dilation.

```
A = imread (image. t i f);
B = s t r e l (‘diamond’, 5)
C = imdilate (A, B);
imshow(C)
```

5.2 Erosion

Erosion shrinks or thins in a binary image. As in dilation, the manner and extent of shrinking is controlled by the structuring element. The output of erosion has a value 1 at each location of the origin of the structuring element, such that the structuring element overlaps only 1-valued pixels of the input image.

The Matlab image processing toolbox function imerode performs erosion. The syntax is

C = imerode (A, B)

where A and C are binary images and B is a matrix of 0s and 1s that specify the structuring element.

5.2.1 Characteristics

a. Erosion generally decreases the sizes of objects and removes small anomalies by subtracting objects with a radius smaller than the structuring element.
b. With grayscale images, erosion reduces the brightness (and therefore the size) of bright objects on a dark background by taking the neighborhood minimum when passing the structuring element over the image.
c. With binary images, erosion completely removes objects smaller than the structuring element and removes perimeter pixels from larger image object.

6 Histogram analysis over image

6.1 Histogram processing/equalization

The Histogram image equalization is the process whereby the preceding transformation will generate an image that will show a histogram at each intensity level. The net result of the process will yield an image with an increased sparse range of intensity and higher contrast compare to the original image. The processed image from the histogram is not uniform due to the discrete nature.

6.2 Histogram-Based Image Search

The color histogram for an image is constructed by counting the number of pixels of each color. Retrieval from image databases using color histograms has been investigated in [tools, fully, automated]. In these studies the developments of the extraction algorithms follow a similar progression:

- Selection of a color space.
- Quantization of the color space.
- Computation of histograms.
- Derivation of the histogram distance function.
- Identification of indexing shortcuts.

6.3 Images showing different kind of Histogram:

6.3.1 High contrast image

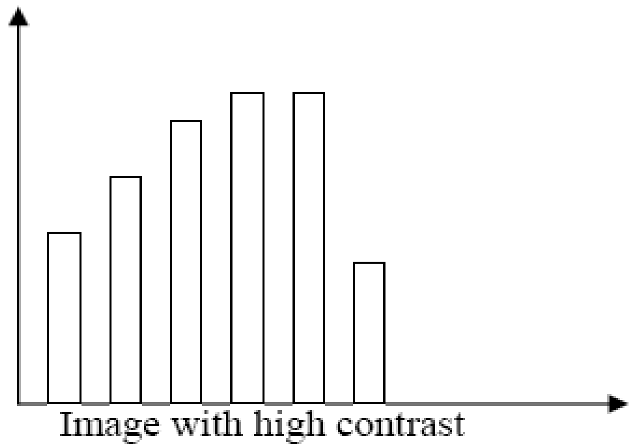


Figure 5 High contrast image

6.3.2 Low contrast image

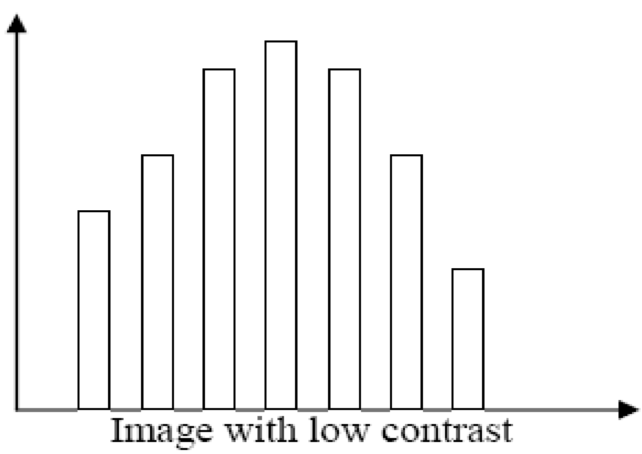


Figure 6 Low contrast image

6.3.3 Dark Image Histogram

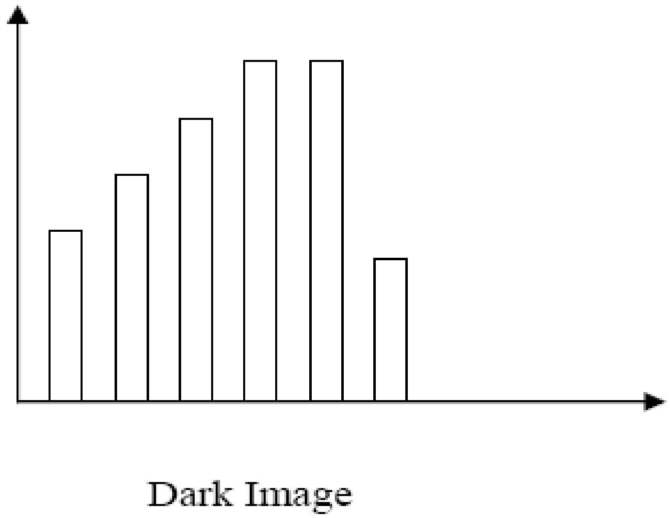


Figure 7 Dark Image Histogram

6.3.4 Bright Image Histogram

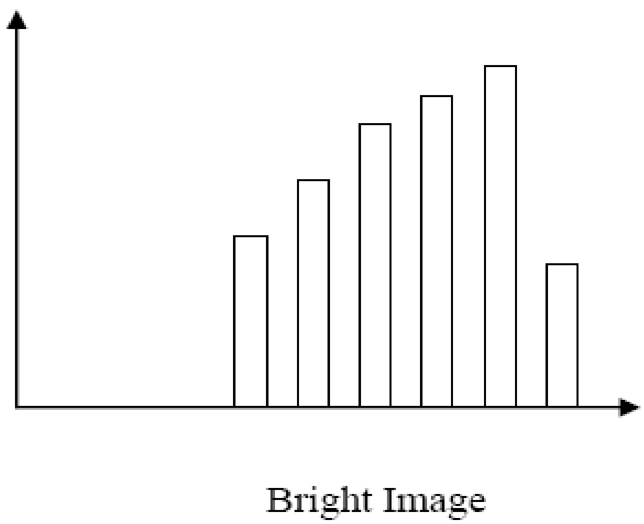


Figure 8 Bright Image Histogram

6.4 Horizontal and Vertical Histogram of Image

In Number plate extraction system we have used horizontal and vertical histograms representing column wise and row wise histogram respectively. The histogram represents the sum of the differences of gray values between the neighboring pixels of an image column wise and row wise.

6.4.1 Horizontal Histogram

To find horizontal histogram the algorithm traverses through each column of an image. In each column it starts with the second pixel from the top. The difference between the second and the first is calculated. If the difference exceeds certain threshold, it is added to total sum of differences. Then the algorithm moves down to calculate the difference between the third and the second pixel and so on. At the end an array containing column wise sum of difference is created.

6.4.2 Vertical Histogram

To find vertical histogram same process is repeated. Here rows are processed instead of column. In each row it starts with the second pixel from the left. The difference between the second and the first is calculated. If the difference exceeds certain threshold, it is added to total sum of differences. Then the algorithm moves right to calculate the difference between the third and the second pixel and so on. At the end an array containing row wise sum of difference is created.

6.5 Smoothening

The vertical and horizontal histogram so obtained contains various peaks. This must be smoothened out before any further processing. Matlab provides us with smooth() command in its toolbox to smoothen any response data.
Here the data in any column vector is smoothen with the help of a moving average filter.

6.6 Filtering out unwanted region in the image

Once the histogram has been smoothened then all the unwanted rows and columns (not containing the number plate) must be filtered out of the image. Here the unwanted areas are the rows and the columns with low histogram values. A low histogram value indicates that the region has low intensity variation among the neighboring pixels. But the region containing the number plate with has intensity variation among neighboring pixels especially at the edges of characters. Thus areas with low value are not required anymore and they will b removed by applying dynamic threshold. In this algorithm the dynamic threshold is the average value of the histogram.

The horizontal and vertical histogram will be passed through a filter with this dynamic threshold and the resulting histogram will contain only areas with highest probability of containing the number plate.

7 Character recognition

Optical Character Recognition (OCR) is a process that translates images of typewritten scanned text into machine-editable text, or pictures of characters into a standard encoding scheme representing them in ASCII or Unicode. An OCR system enable us to feed a book or a magazine article directly into a electronic computer file, and edit the file using a word processor. Though academic research in the field continues, the focus on OCR has shifted to implementation of proven techniques.

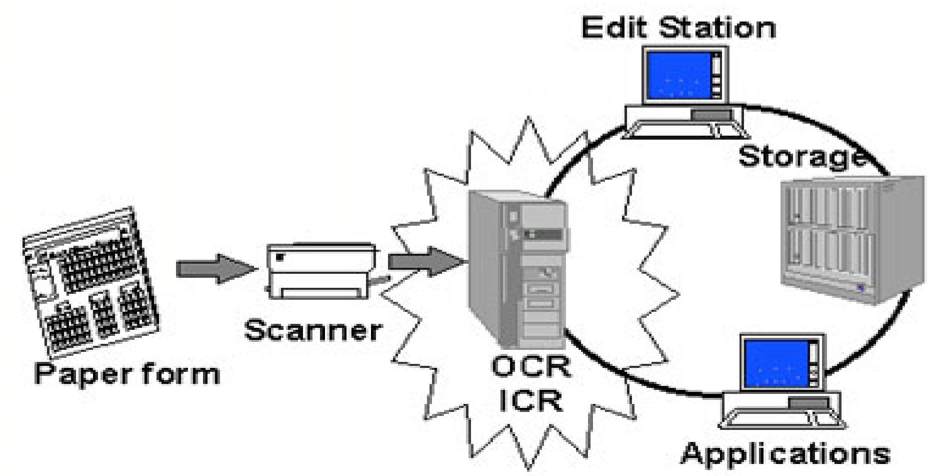


Figure 9 Character Recognition

Optical character recognition (using optical techniques such as mirrors and lenses) and digital character recognition (using scanners and computer algorithms) were originally considered separate fields. Because very few applications survive that use true optical techniques, the OCR term has now been broadened to include digital image processing as well. Early systems required training (the provision of known samples of each character) to read a specific font. "Intelligent" systems with a high degree of recognition accuracy for most fonts are now common. Some systems are even capable of reproducing formatted output that closely approximates the original scanned page including images, columns and other non-textual components. However, this approach is sensitive to the size of the fonts and the font type. For handwritten input, the task becomes even more formidable. Soft computing has been adopted into the process of character recognition for its ability to create input output mapping with good approximation. The alternative for input/output mapping may be the use of a lookup table that is totally rigid with no room for input variations. A few examples of OCR applications are listed here:

- People wish to scan in a document and have the text of that document available in a word processor.
- Recognizing license plate numbers
- Post Office needs to recognize zip-codes
- I have used OCR application for number plate recognition as well as for printed text recognition.

The character recognition algorithm has two essential components:

- Feature extractor
- The classifier

7.1 Feature Extractor

Feature analysis determines the descriptors, or the feature set used to describe all characters. Given a character image, the feature extractor derives the features that the character possesses. The derived features are then used as input to the character classifier.

Template matching or matrix matching, is one of the most common classification methods. Here individual image pixels are used as features. Classification is performed by comparing an input character with a set of templates (or prototypes) from each character class. Each comparison results in a similarity measure between the input characters with a set of templates. One measure increases the amount of similarity when a pixel in the observed character is identical to the same pixel in the template image. If the pixels differ the measure of similarity may be decreased. After all templates have been compared with the observed character image, the character's identity is assigned the identity of the most similar template.

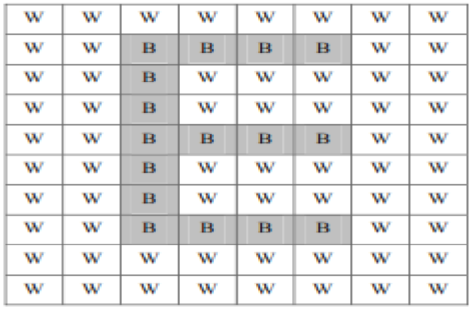


Figure 10 Identification of background and character

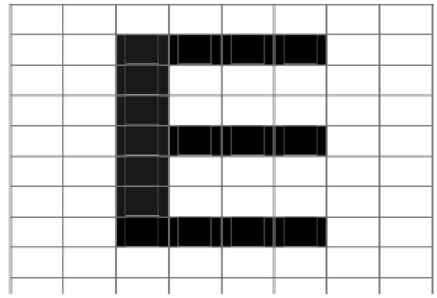
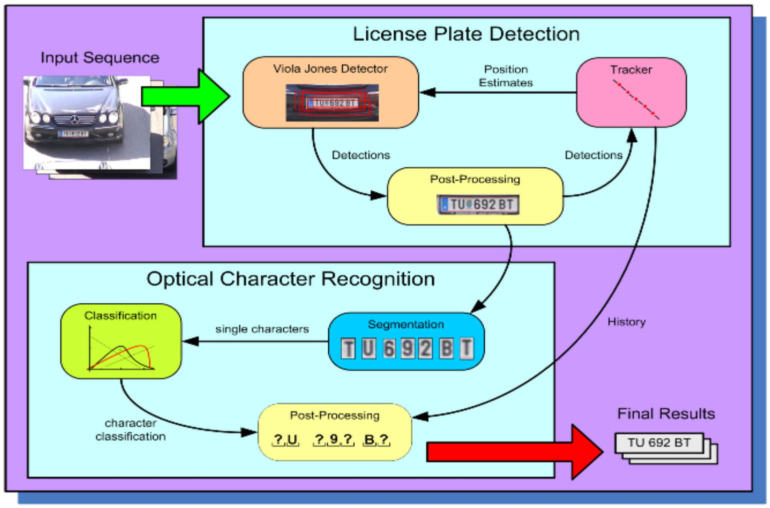


Figure 11 Extraction of a character in a matrix

Template matching is a trainable process as template characters can be changed. This method considers the handwritten character to be a directed abstract graph of node sets consisting of tips, corners and functions and the branch set consists of line segments connecting pair of adjacent nodes. The segments (branches) are fuzzily classified to branch types (features) such as straight lines, circles or portions of circles. Since the features under consideration are fuzzy in nature the fuzzy set is utilized and the features are treated as fuzzy variables. As handwritten characters are considered to be ill-defined objects, with the art of fuzzy functions such objects can be objectively defined and studied. A handwritten character is represented by fuzzy functions which relates its fuzzy variables and by the node pair involved in each fuzzy variable. The recognition involves two steps:

- First the unknown character is preprocessed to produce its representation.
- The classification of the unknown character is reduced to finding a character (previously learned of which the representation is isomorphic to the representation of the unknown character).

As there is lack of precision in the definition of the elements of the feature set, the fuzzy concept has been used. The characterization is position, size and distortion variant.



7.2 The Classification Process

(Classification in general for any type of classifier) There are two steps in building a classifier:

1. Training
   a) Pre-processing – Processes the data so it is in a suitable form
   b) Feature extraction – Reduce the amount of data by extracting relevant information - Usually results in a vector of scalar values. (We also need to NORMALIZE the features for distance measurements!)
   c) Model Estimation – from the finite set of feature vectors, need to estimate a model (usually statistical) for each class of the training data
2. Testing

   a) Pre-processing
   b) Feature extraction – (both same as above)
   c) Classification – Compare feature vectors to the various models and find the closest match. One can use a distance measure.

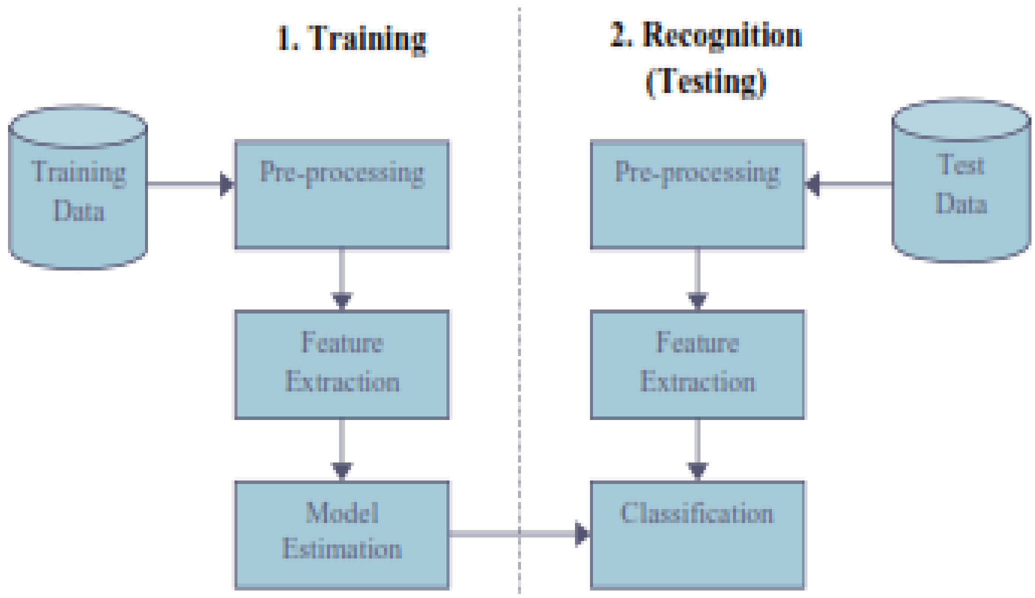


Figure 12 The pattern classification process

7.3 OCR – Pre-processing

These are the pre-processing steps often performed in OCR

a) Binarization – Usually presented with a grayscale image, binarization is then simply a matter of choosing a threshold value.

b) Morphological Operators – Remove isolated specks and holes in characters, can use the majority operator.

c) Segmentation – Check connectivity of shapes, label, and isolate. Can use Matlab 6.1's bwlabel and regionprops functions. Difficulties with characters that aren't connected, e.g. the letter i, a semicolon, or a colon (; or :).

Segmentation is by far the most important aspect of the pre-processing stage. It allows the recognizer to extract features from each individual character. In the more complicated case of handwritten text, the segmentation problem becomes much more difficult as letters tend to be connected to each other.

7.4 OCR – Feature extraction

Given a segmented (isolated) character, what are useful features for recognition?

a) Moment based features

Think of each character as a pdf. The 2-D moments of the character are:

$$m_{pq} = \sum_{x=0}^{W-1} \sum_{y=0}^{H-1} x^p y^q f(x, y)$$

From the moments we can compute features like:

1. Total mass (number of pixels in a binarized character)
2. Centroid - Center of mass
3. Elliptical parameters
   i. Eccentricity (ratio of major to minor axis)
   ii. Orientation (angle of major axis)
4. Skewness
5. Kurtosis
6. Higher order moments

b) Hough and Chain code transform

c) Fourier transform and series

7.5 OCR - Model Estimation

Given labeled sets of features for many characters, where the labels correspond to the particular classes that the characters belong to, we wish to estimate a statistical model for each character class. For example, suppose we compute two features for each realization of the characters 0 through 9. Plotting each character class as a function of the two features we have:

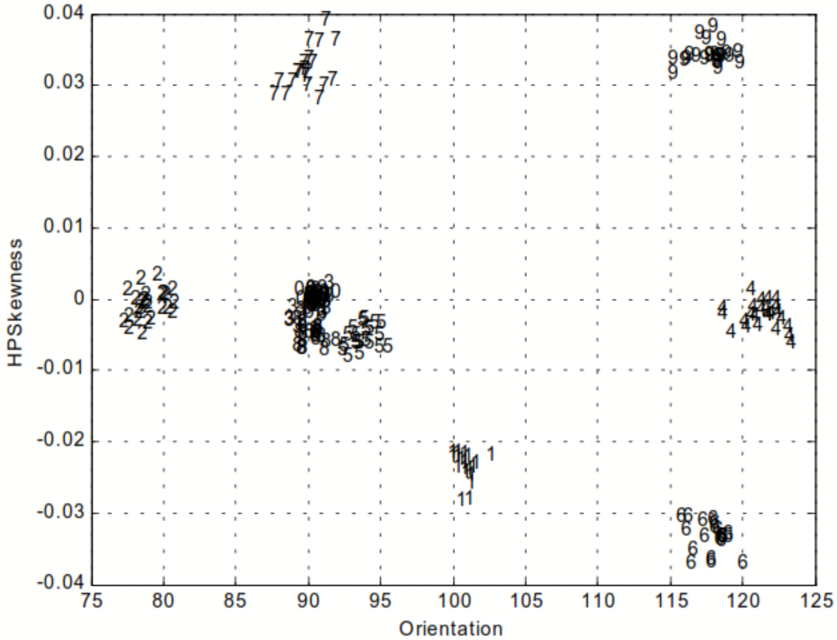


Figure 13 Character classes plotted as a function of two features

According to Tou and Gonzalez, “The principal function of a pattern recognition system is to yield decisions concerning the class membership of the patterns with which it is confronted.” In the context of an OCR system, the recognizer is confronted with a sequence feature patterns from which it must determine the character classes.

Each character class tends to cluster together. This makes sense; a given number should look about the same for each realization (provided we use the size font type and size). We might try to estimate a pdf (or pdf parameters such as mean and variance) for each character class. For example, in Figure 3, we can see that the 7’s have a mean Orientation of 90 and HP Skewness of 0.033.

8 Method

- Firstly the image of the car is captured with help of a camera from which we want to extract the number plate and the image is read and displayed using imread ( ) and imshow ( ) commands respectively.

Figure 14 Original image of car

- <u>RGB image</u> obtained is converted into gray image using the command rgb2gray( ) by using the formula:

$$Y = 0.30R + 0.59G + 0.11B$$

to obtain the luminance signal

Figure 15 Grayscale image obtained from rgb image

- Then the process of erosion is applied to the grayscale image by using the command imerode( ) by using the structuring element rectangle of size = [2 3].

Figure 16 Eroded grayscale image

Ø Dilation is performed on the eroded image to remove any unwanted noise by using the command imdilate( ) by using the same structuring element rectangle of size =[2 3].

Figure 17 Dilated grayscale image

Ø Using the histogram-based approach, horizontal histogram is plotted by calculating the difference between adjacent pixels in each adjacent column pair in each row and if they are greater than the threshold value set, then the differences is stored in a row vector.

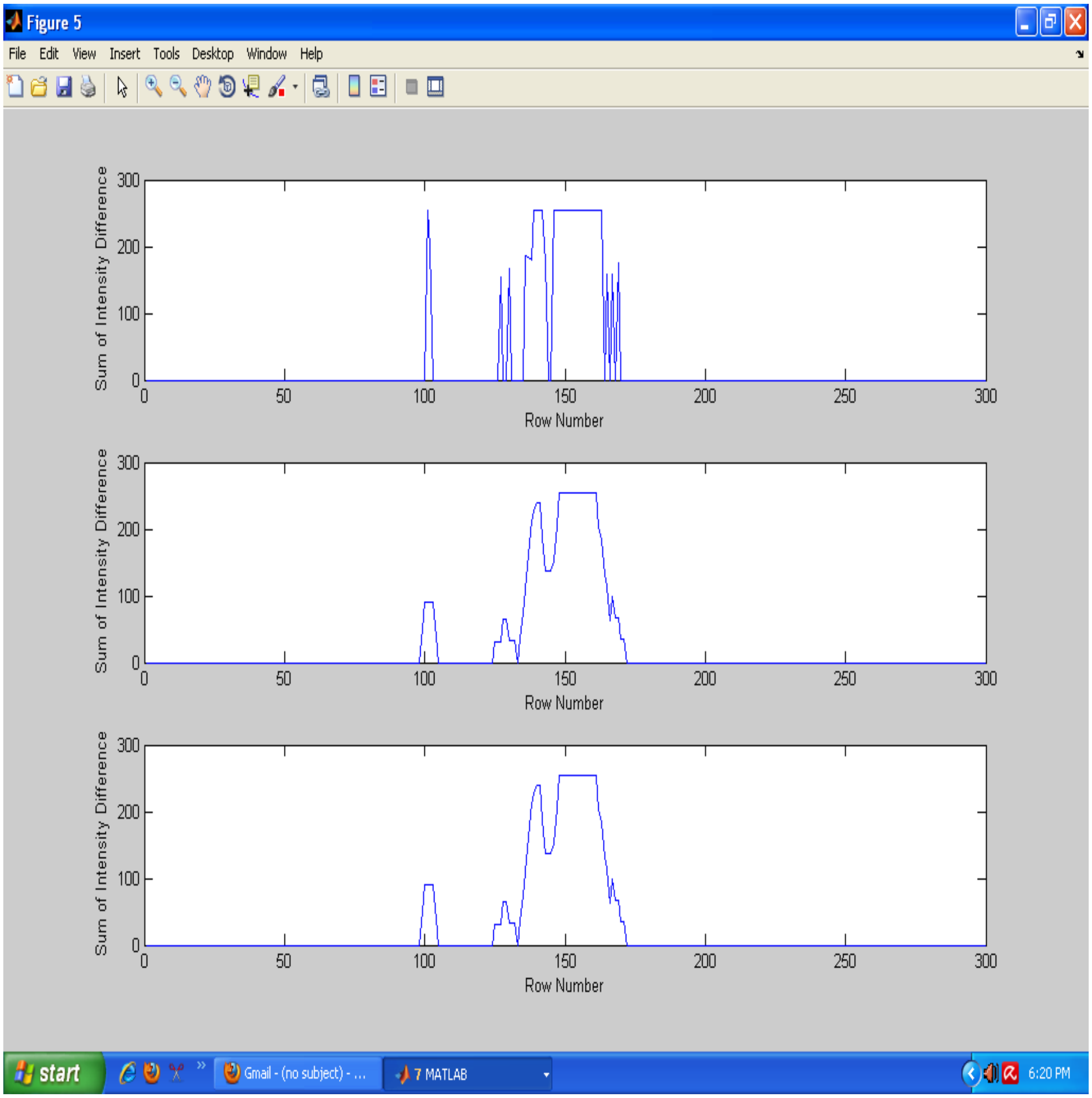


Figure 18 Row difference vector

- Using the histogram-based approach, vertical histogram is plotted by calculating the difference between adjacent pixels in each adjacent row pair in each column and if they are greater than the threshold value set ,then the differences are stored in a column vector.

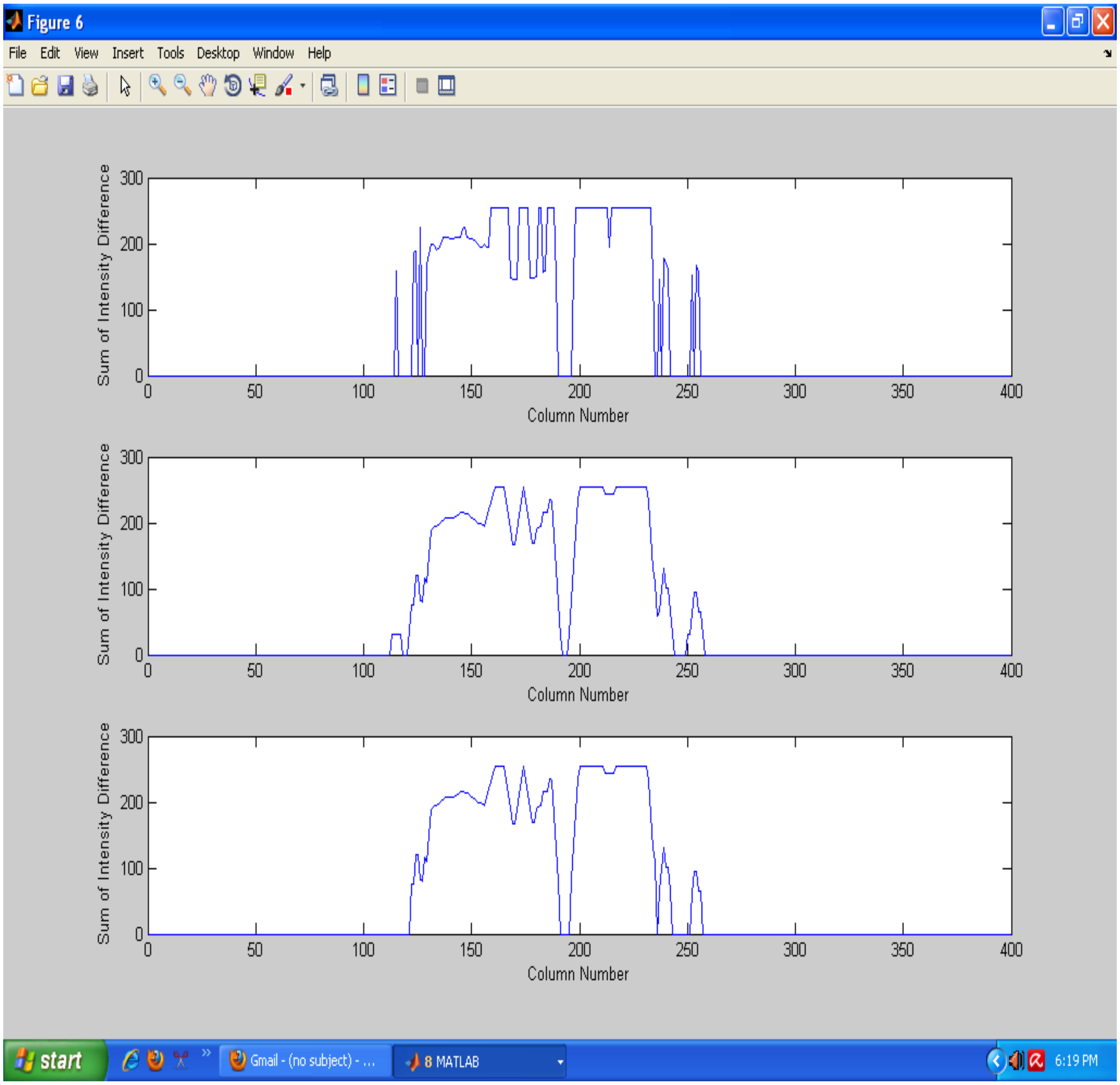


Figure 19 Column difference vector

Ø Then the Segmentation of the image is performed. Obtained histograms are smoothened and then filtering is performed using the averaging method i.e.

by calculating the average of both row and column vector and setting elements having value less than average value equal to zero in both row and column. And thus the rows and columns having highest probability of containing the number plate is segmented out.

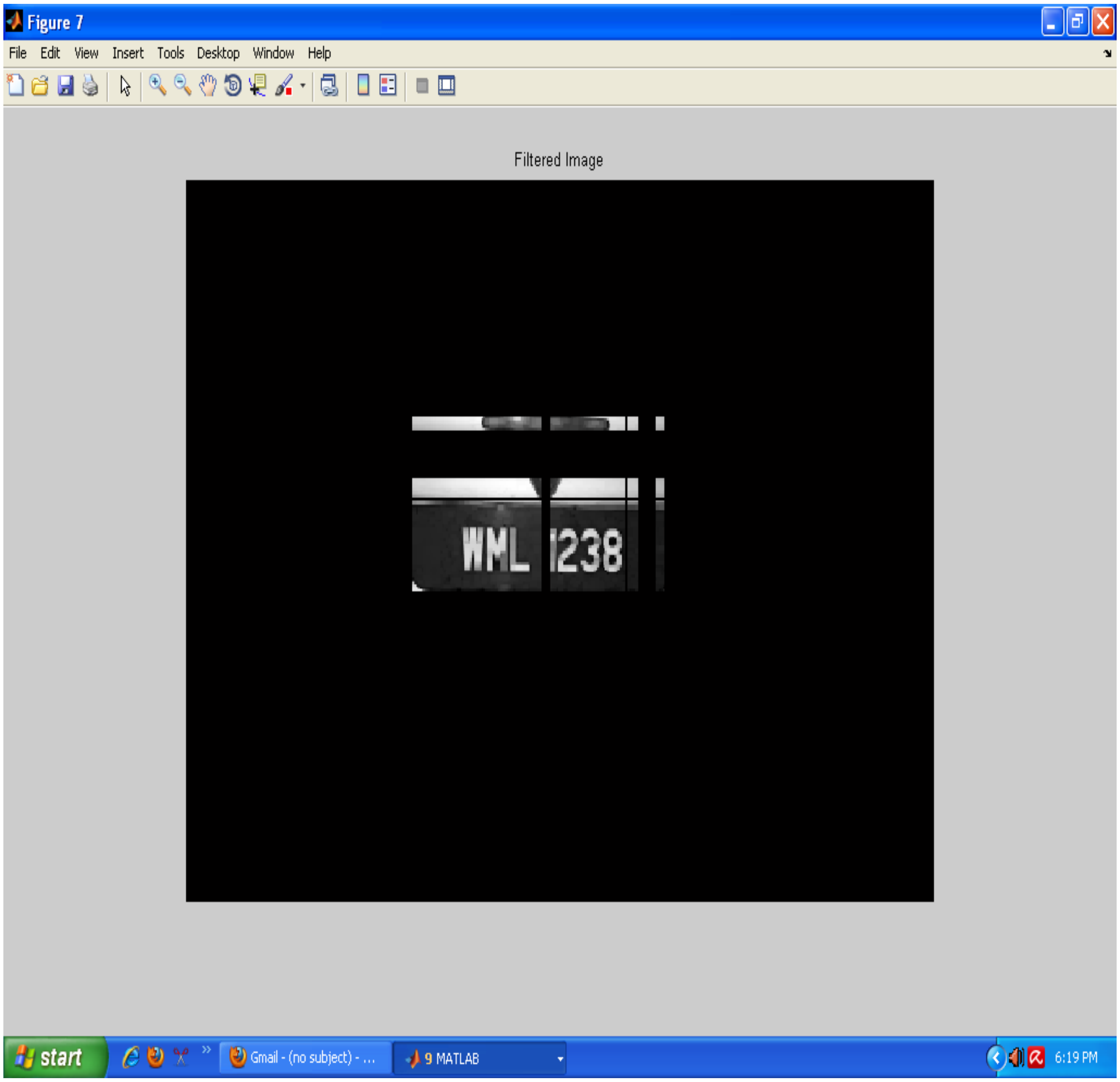


Figure 20 Filtered image

- After this region of interest is extracted out. It is done by removing unwanted rows and columns and by selecting that area from the filtered image having maximum intensity variation vertically and horizontally.

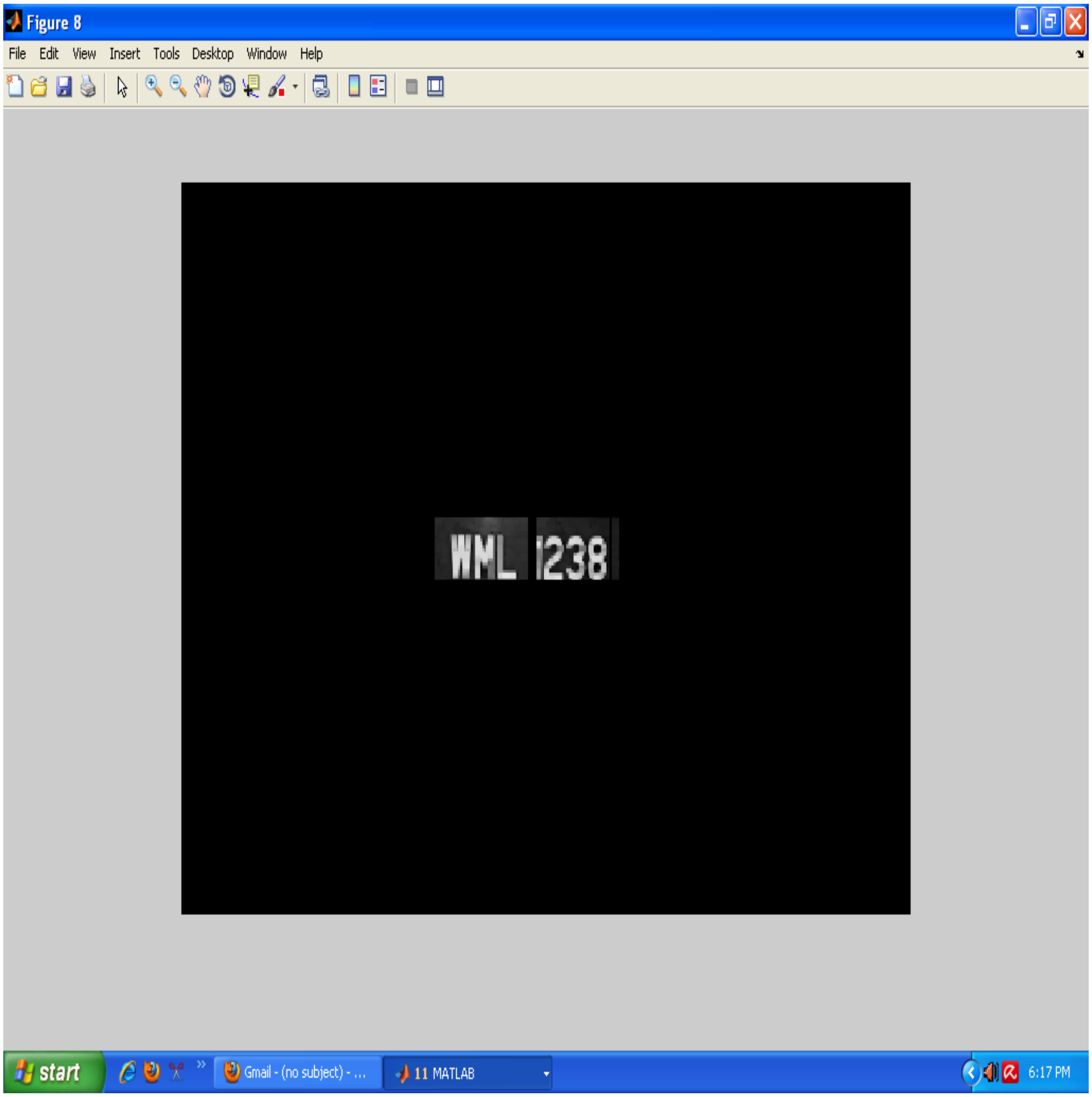


Figure 21 Removing unwanted rows and columns

Ø Final number plate is obtained by converting the grayscale image into binary image using the command im2bw( ) by setting a threshold level.

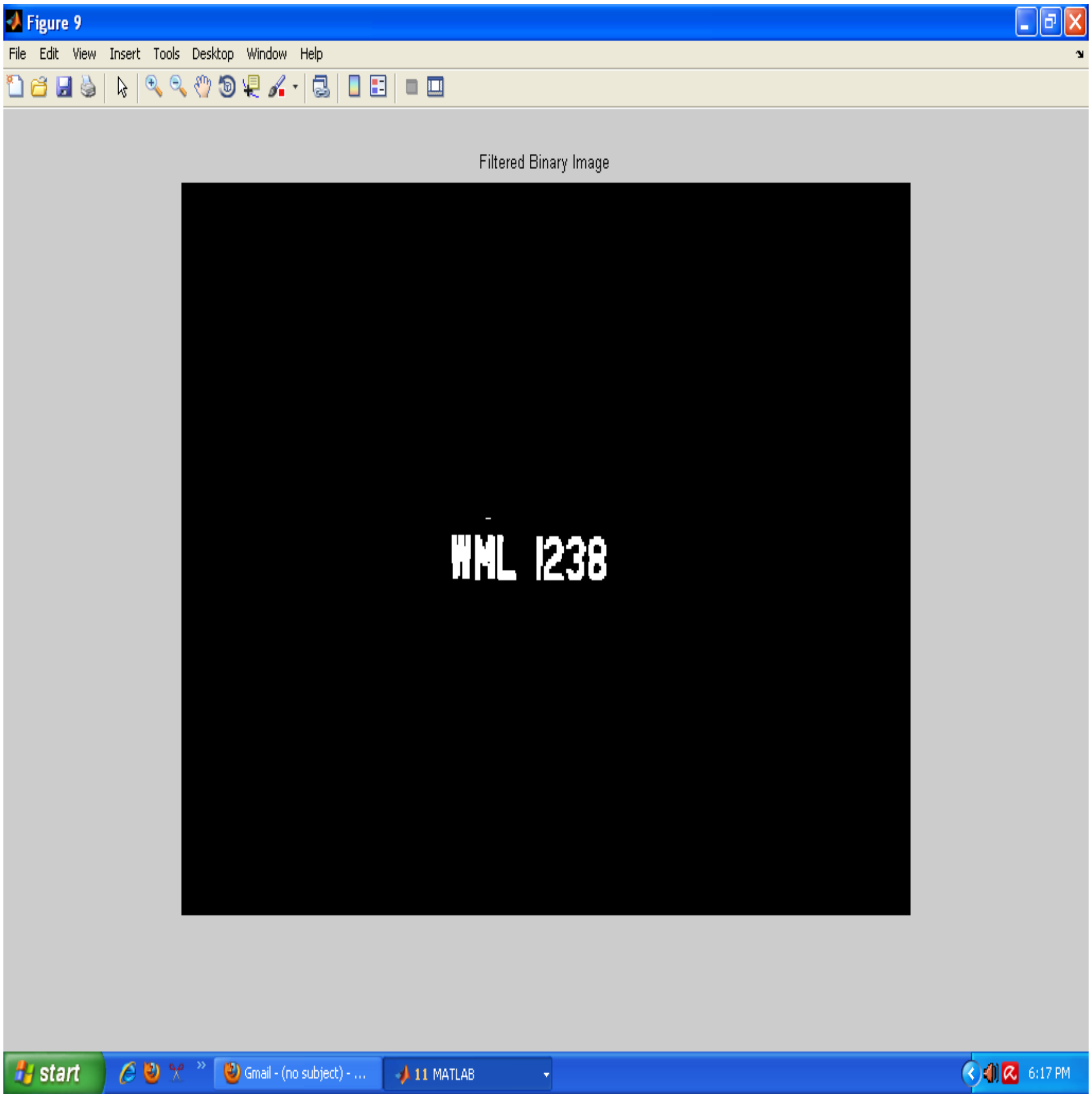


Figure 22 Binary image of number plate

Ø Finally number plate is extracted from the image.

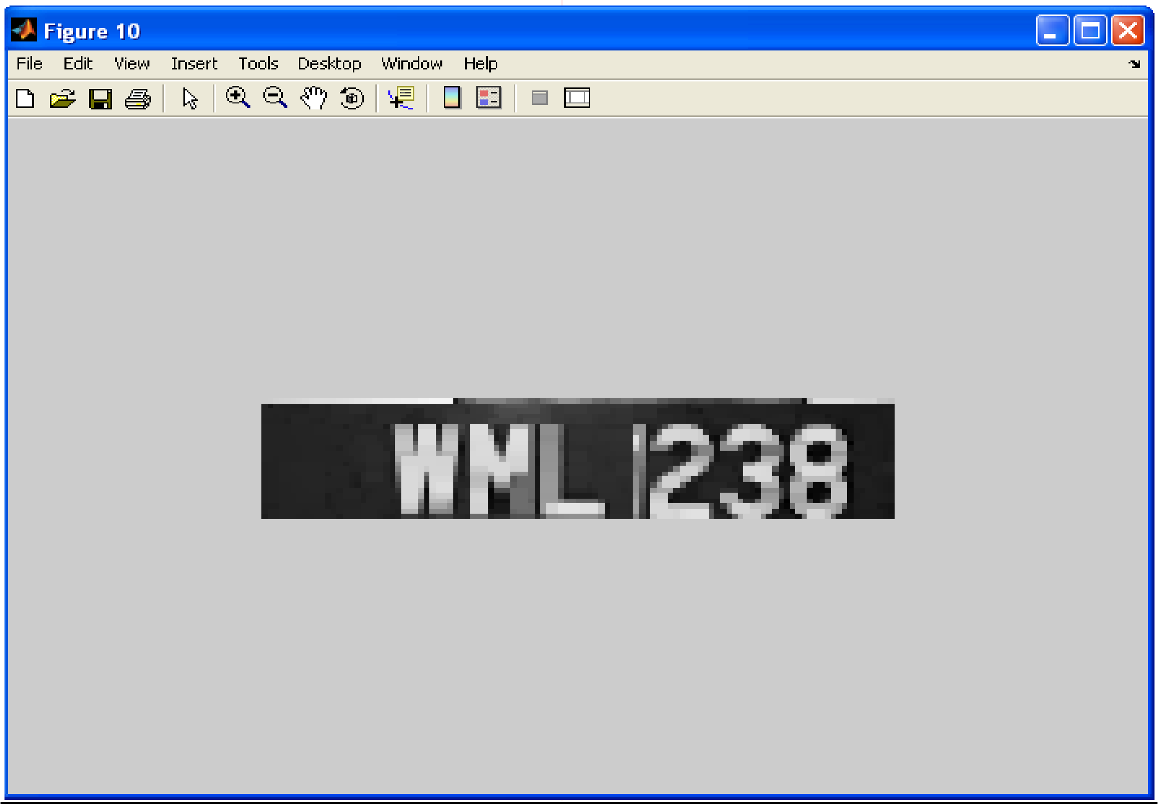


Figure 23 Final extracted number plate

8.1 Results

The number plates of 10 cars have been successfully extracted from the respective images of the cars and they are matched with the original number plates and the accuracy of the results obtained is as shown in the table below:-

S.No	Original Number Plate	Extracted Number plate	Accuracy
1.			9/10

2.			10 / 10
3.			10 / 10
4.			8 / 10
5.			7 / 10
6.			7 / 10
7.			8 / 10
8.			8 / 10
9.			7 / 10
10.			7 / 10

Final accuracy of the project= ((accuracy of each image)/100)*100%

= 93 %

8.2 Character Segmentation

Image segmentation plays an important and critical step that lead to the analysis of the processed image data. In order to extract and analyze the object characteristic, the process need to partition the image into different parts that will have a strong correlation with the objects.

8.2.1 Connected Component Analysis

- Once region boundaries have been detected, it is often useful to extract regions which are not separated by a boundary.
- Any set of pixels which is not separated by a boundary is call connected.
- Each maximal region of connected pixels is called a connected component.
- The set of connected components partition an image into segments.
- Image segmentation is a useful operation in many image processing applications.

8.2.2 Connected Neighbors

- Let •s be a neighborhood system.
  – 4-point neighborhood system
  – 8-point neighborhood system

- Let c(s) be the set of neighbors that are connected to the point s. For all s and r, the set c(s) must have the properties that
  – c (s) • •s

– r • c(s) • s • c(r)

- Example:

  c (s) = {r • •s : Xr = Xs}

  c (s) = {r • •s : |Xr • Xs| < Threshold}

- In general, computation of c(s) might be very difficult, but we won't worry about that now.

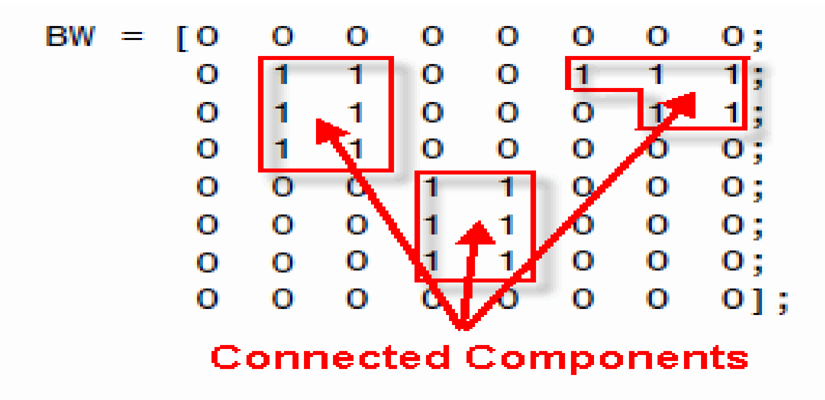


Figure 24 Connected Components

STATS = regionprops (BW, properties) measures a set of properties for each connected component (object) in the binary image, BW. The image BW is a logical array; it can have any dimension. The output is as shown below:

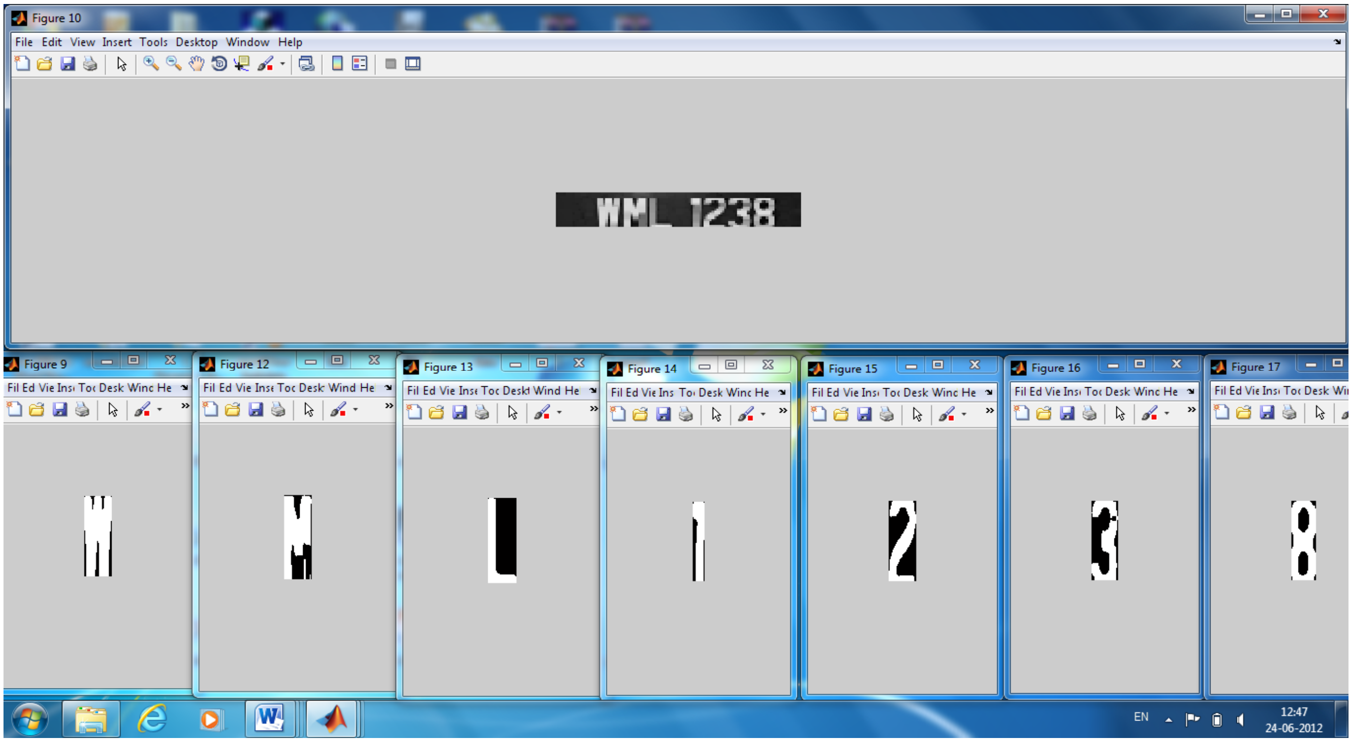


Figure 25 Output after segmentation

8.3 The Matlab Implementation

The Character Classifier Graphical User Interface (GUI)

A Matlab GUI was written to encapsulate the steps involved with training an OCR system. This GUI permits the user to load images, binarize and segment them, compute and plot features, and save these features for future analysis. The file is called train.m and has been implemented.

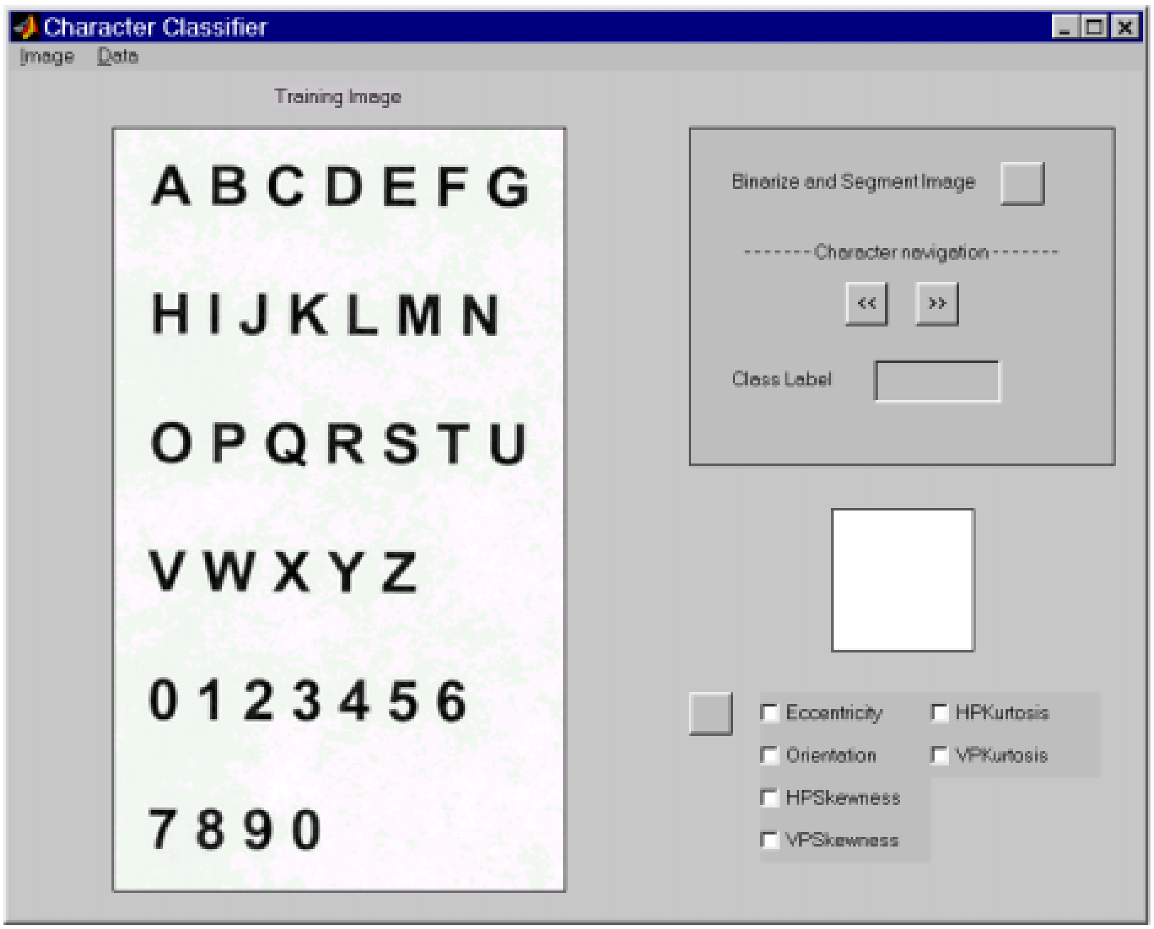


Figure 26 The Character Classifier Graphical User Interface

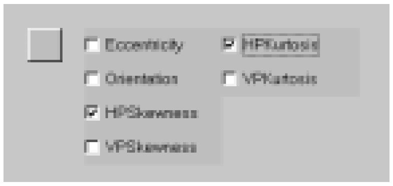


Figure 27 Select features to plot

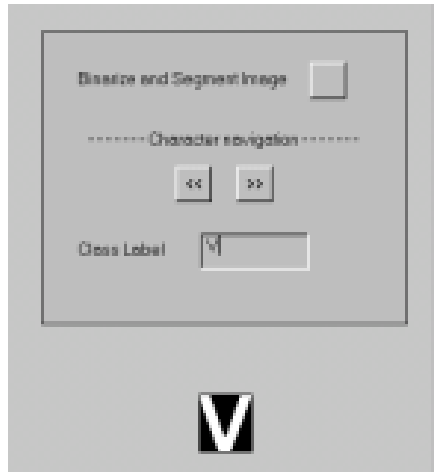


Figure 28 Labeling the characters

Comparison between various methodologies

Approach	Representation	Recognition Function	Typical Criterion
Template matching	Samples, pixels, curves	Correlation, distance measure	Classification error
Statistical	Features	Discriminant function	Classification error
Syntactic or structural	Primitives	Rules, grammar	Acceptance error
Neural networks	Samples, pixels, features	Network function	Mean square error

Figure 29 Comparison between various methodologies

8.4 Templates

8.5 Final output after OCR implementation

9 Conclusions

The development of the Automatic Number Plate Recognition System is quite successfully implemented using MATLAB version 7.

For this project, I have discussed with my mentor and have listed out the process report that needed to be done.

- Read the color image in Matlab.
- Analyze the RGB image.
- Convert the image from RGB to gray and to remove hue, saturation.
- Image enhancement by using Histogram Equalization
- Noise Reduction using erosion and dilation.
- Plotting both horizontal and vertical histograms and analyzing them.
- Performing smoothening and filtering using averaging method.
- Extracting the final number plate
- Repeating the process for 10 images of cars
- Matching the extracted images with the original images and determining the accuracy.

After much research, reading and mastering the skill of MATLAB, I am able to complete the above task and achieve good results on the recognition. When running the main .m file, it is able to detect and recognize some of the zoom in car number plate but fail on the image of cars with headlights.

For the future works and suggestion on improvements, these are the steps which are recommended:

- Modification is needed to be done on the offset of detecting the rectangular plate or by applying other technique to the system.

- There are quite a number of possible case and due to lack of time, I did not tackle cases which capture image from the camera. We can incorporate by using GUI to MATLAB which will in line with a video camera.
- There should be improvement on the decision of the algorithm and ways to detect error. When the probability of recognition guess is correct but falls below the threshold, the recognition system should refuse to make the decision.

10 Applications

- Speed Checks
- Park Management:-

  Lost tickets

  Lost vehicles

  Vehicle theft

  Duration of stay monitoring

  Revenue auditing
- Traffic Management:-

  Bus & taxi lane compliance

  Congestion charge compliance and payment

  Persistent evader searches
- Other:-

  On-street compliance cameras in P&D machines

  Access control and monitoring

11 Constraints

Due to limited time that we possess and dealing with image vision software, it is not advisable to include all of the possible cases. Thus, we have to set a list of constraints to make the project more systematic and manageable. The constraint is listed as below:

- Image taken only when vehicle is stationary.
- Captured image of vehicle at fixed distance.
- Captured image of vehicle at fixed angle
- There will be no motion capture image
- The number plate position should be captured centered
- The image should be taken with the height of 50cm to 70 cm above the ground level.
- Take only the back view image of the car.
- Try on zoom in image of the car and image consists of headlamp
- Captured images on location where light is proportional

12 Bibliography

1. Jun Wu, Zhitao Xiao ''New License Plate Character Recognition Algorithm based on ICM'' Control, Automation and Systems Engineering (CASE), 2011 International Conference.
2. P. V. Suryanarayana, Suman K. Mitra, Asim Baneree and Anil K Royhalina Kwasnicka and Bartosz Wawrzyniak, "A Morphology Based Approach for Car License plate extraction ''IEEE Indicon 200 conference, Chennai , India, I I - 1 3 Dec. 2005.
3. Tarun Kumar and Karun Kumar "A theory based on conversion of RGB Image to Grayscale image", (International Journal of Computer Applications (0975 – 8887) Volume 7– No.2, September 2010).
4. R.C. Gonzalez, R.E. Woods, Digital Image Processing, Addison-Wesley.
5. Prathamesh Kulkarni (Student Member, IEEE), Ashish Khatri, Prateek Banga, Kushal Shah"Automatic Number Plate Recognition (ANPR) System''.
6. Nafiz Arica, Student Member, IEEE and Fatos T. Yarman-Vural (Senior Member, IEEE),''An overview of character recognition focussed on off-line handwriting''.
7. Clemens Arth, Florian Limberger, Horst Bischof, "Real time license plate recognition on an embedded DSP-Platform".
8. Yungang Jhang, Changshui Jhang, ''A new algorithm for character segmentation of license plate''.

13 Appendix

13.1 Matlab code

13.1.1 Algorithm for Importing and Processing the Image

```
A=imread('car1.jpg');            %importing JPG image in matlab
figure(1);
imshow(A)                        %displaying the original image
title('Original Image')

I=rgb2gray(A);                   %converting RGB image into Gray image
figure(2);
imshow(I)                        %displaying gray image
title('Gray Image')

SE=strel('rectangle',MN);        %defining the structuring element

BW2=imerode(I,SE);               %performing erosion
figure(3);
imshow(BW2)                      %displaying eroded image
title('Image After Erosion')

BW3=imdilate(BW2,SE);            %performing dilation
figure(4);
imshow(BW3)                      %displaying dilated image
title('Image After Dilation')
```

13.1.2 Algorithm for Vertical Histogram

```
%------------ VERTICAL HISTOGRAM-----------


c= zeros(p,1);        %row vector for sum of difference of rows
rand1=0;

row = [ 1:1:p] ;        % row vector
```

```
%Loop for calculating vertical histogram

for i=1:1:p                          % varying i for rows
   for j=1:1:q-1                       % varying j for column pixel within the row
      if (BW3(i,j+1)>BW3(i,j))              %comparing neighboring pixel and
          rand1= BW3(i,j+1) - BW3(i,j);     % finding difference among them.
      else
        rand1= BW3(i,j)-BW3(i,j+1);
      end
       if( rand1>=155)                   % if difference is greater than threshold
          c(i)= c(i)- rand1;              % then included in sum of difference

       end
   end
end

figure(5);
title('Vertical Histogram')
subplot(3,1,1)
plot(row,c)                             % plotting vertical histogram
title('Row Difference Vector')
xlabel('Row Number')
ylabel('Sum of Intensity Difference')

s=smooth(c);                           % smoothening histogram
subplot(3,1,2)
plot(row,s)                          %  plotting smoothened vertical histogram
% title('Smoothen Row Difference Vector')
xlabel('Row Number')
ylabel('Sum of Intensity Difference')


% calculating average for the histogram
y=0;

 for i=1 :1 :p
   y=y+s(i);

 end
```

```
avg=y / p;

% filtering histogram with average as threshold
for i=1:1:p
   if(s(i)<avg)
       s(i)=0;
   end
end

subplot(3,1,3)
plot(row,s)            % plotting vertical histogram after filtering
% title('Row Difference Vector After Filtering')
xlabel('Row Number')
ylabel('Sum of Intensity Difference')
```

13.1.3 Algorithm for Horizontal Histogram

```
%------------ HORIZONTAL HISTOGRAM-----------

d= zeros(1,q);        % column vector for sum of difference of columns

rand=0;

col=[1:1:q];          % column vector


%Loop for calculating horizontal histogram

for j=1:1:q                    % varying j for columns
   for i=1:1:p-1                    % varying i for row pixel within the column
       if (BW3(i+1,j)>BW3(i,j))             %comparing neighboring pixel and
              rand= BW3(i+1,j) - BW3(i,j);    % finding difference among them.
       else
             rand= BW3(i,j)-BW3(i+1,j);
       end
       if( rand>=145)                  % if difference is greater than threshold
            d(j)= d(j)- rand;           % then included in sum of difference
       end
   end
```

```
end

figure(6);
title('Horizontal Histogram')
subplot(3,1,1)
plot(col,d)                              % plotting horizontal histogram
% title('Column Difference Vector')
xlabel('Column Number')
ylabel('Sum of Intensity Difference')

r=smooth(d);                             % smoothening histogram
subplot(3,1,2)
plot(col,r)                              %  plotting smoothened horizontal histogram
% title('Smoothen Column Difference Vector')
xlabel('Column Number')
ylabel('Sum of Intensity Difference')


% calculating average for the histogram
 x=0;
 for i=1 :1 :q
    x=x+d(i);
 end

 avg1=x/q;

% filtering histogram with average as threshold
 for i=1:1:q
    if(r(i)<avg1)
        r(i)=0;
    end
end

subplot(3,1,3)
plot(col,r)                   % plotting horizontal histogram after filtering
% title('Column Difference Vector After Filtering')
xlabel('Column Number')
ylabel('Sum of Intensity Difference')
```

13.1.4 Algorithm for Segmentation of Image:

```
% removing unwanted rows and columns from image using filtered histogram

for i=1:1:p
   for j=1:1:q
      if((s(i)==0)||(r(j)==0))
            BW3(i,j)=0;
      end
   end
end
figure(7)
imshow(BW3)
title('Filtered Image')      % displaying filtered image


% further extracting the region of interest

% loop for extracting unwanted rows with low histogram value
for i= 1 : 1 : p
   if(c(i)<255)
      s(i)=0;
   end
end

% loop for extracting unwanted columns with low histogram value
for j= 1 : 1: q
   if(d(j)<140)
      r(j)=0;
   end
end

% removing unwanted region from the image
for i=1:1:p
   for j=1:1:q
      if((s(i)==0)||(r(j)==0))
            BW3(i,j)=0;
      end
   end
end
figure(8);
```

```
imshow(BW3);    % displaying image after final filtering of unwanted region

%  converting image into binary
level=graythresh(BW3)
BW=im2bw(BW3,level);
figure(9);
imshow(BW)
title('Filtered Binary Image')
```

13.1.5 Algorithm for Extracting the Number Plate

```
% Calculating row sum

rsum = zeros(p,1);      % row sum vector

rowcount=0;

for i= 1 : 1 : p
   for j=1 : 1 :q
      rsum(i)= rsum(i) + BW3(i,j);
   end

    if (rsum(i)>0)
          rowcount=rowcount+1;    % calculating number of rows with number plate

    end

end

% Calculating column sum

csum= zeros(1,q);

columncount=0;

for j= 1 : 1 : q
   for i=1 : 1 :p
      csum(j)= csum(j) + BW3(i,j);
   end
   if(csum(j)>0)
```

```
        columncount=columncount+1;    % calculating number of column with number plate

    end
end

X1= zeros  (rowcount,columncount,'uint8');  % defining new image matrix for saving the extracted number plate

% loop for extracting the number plat and save it in new image matrix

m=0;
for i=1 :1 : p
    if(rsum(i)>0)
        m=m+1;
    end
    n=0;
    for j= 1 : 1:q
        if(csum(j)>0)
            n=n+1;
        end
        if(BW3(i,j)>0)
           X1(m,n )= BW3(i,j);
        end
    end
end

figure(10);
imshow(X1);    % displaying extracted number plate
```

13.2 MATLAB FUNCTIONS USED

Imread( ):- Read an image.(Within the parenthesis you type the name of the image file you wish to read. Put the file name within single quotes ' ' followed by extension of the format used of the image.)

Imshow( ):-Display an image represented as the matrix X.(Within the parenthesis you type the name of the image file you wish to read)

Figure( ):-Show the image (with several figures within the same program you can number the figures in this command)

Rgb2gray( ):-Converts RGB image to gray image by using the formula:

$$Y = 0.30R + 0.59G + 0.11B$$

to obtain the luminance signal

.

Im2bw( ) :- To convert any grayscale image into a binary image by setting a threshold level.

Imerode( ):- Used to erode the grayscale image to remove any unwanted noise (Within the parenthesis you type the name of the image file you wish to erode)

Imdilate( ):- Used to dilate the eroded image to remove any unwanted noise (Within the parenthesis you type the name of the image file you wish to erode)

Smooth( ):-Smooths any response data having various peaks. Here the data in any column vector is smoothened with the help of a moving average filter.