VISVESVARAYA TECHNOLOGICAL UNIVERSITY

BELAGAVI-590018

A Project Report
on

"Number Plate Recognition"

Submitted in partial fulfillment of the requirements for the award of the degree of
Bachelor of Engineering in Computer Science and Engineering
of Visvesvaraya Technological University, Belagavi

by

Vishal U	**1RN12CS125**
Sachin P	**1RN12CS079**
Shreyas R	**1RN12CS100**
Kruti B	**1RN12CS037**

Under the Guidance of:

Mr. Bhavanishankar K
Asst. Professor
Dept. of CSE, RNSIT

Department of Computer Science and Engineering

RNS Institute of Technology

Channasandra, Dr. Vishnuvardana Road, Bengaluru-560 098

2015-2016

RNS Institute of Technology

Channasandra, Dr. Vishnuvardana Road, Bengaluru-560 098

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

Certified that the project work entitled "***Number Plate Recognition***" has been successfully carried out by **Vishal U, Sachin P, Shreyas R** and **Kruti B** bearing USN 1RN12CS125, 1RN12CS079, 1RN12CS100 and 1RN12CS037 respectively**,** bonafide students of **RNS Institute of Technology** in partial fulfillment of the requirements for the award of degree in **Bachelor of Engineering in Computer Science and Engineering** of **Visvesvaraya Technological University, Belagavi** during academic year 2015-2016. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report deposited in the departmental library. The project report has been approved as it satisfies the academic requirements in respect of project work for the said degree.

Mr. Bhavanishankar K **Asst. Prof., Dept. of CSE,** **Internal Guide**	**Dr. G T Raju** **Dean, Prof. and HOD,** **Dept. of CSE**	**Dr. M K Venkatesha** **Principal**

<u>External Viva:</u>

Name of the Examiners	**Signature with Date**
1.	
2.	

ABSTRACT

Number Plate Recognition (NPR) is an application of Digital Image Processing (DIP). In which, number plate of vehicle is used to recognize the vehicle. The objective is to design an efficient vehicle identification system by using the vehicle number plate and to implement it for various applications such as Toll tax collection, Parking system, Border crossings, Traffic control, Stolen cars etc. The system has color image inputs of a vehicle and the output has the registration number of that vehicle. The system first senses the vehicle and then gets an image of front or back view of the vehicle. The system focuses on processing the image in Matlab software for number plates in India. The system has four main steps to get the required information. They are Image Acquisition, Plate Localization, Character Segmentation and Character Recognition. This system mainly introduces a Number Plate Recognition System (NPR) using Morphological operations, Histogram manipulation and Edge detection Techniques for Plate Localization and Characters Segmentation. Characters are identified using Optical Character Recognition (OCR) with correlation approach. The algorithm is successfully constructed with sample of images correctly identified.

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of the people who made it possible, whose constant guidance and encouragement crowned the efforts with success.

We would like to profoundly thank **Management** of **RNS Institute of Technology** for providing such a healthy environment for the successful completion of Project work.

We would like to express my thanks to the Director **Dr. H N Shivashankar** and the Principal **Dr. M K Venkatesha** for their encouragement that motivated me for the successful completion of Project work.

It gives me immense pleasure to thank **Dr. G T Raju,** Dean, Professor and Head of Department for his constant support and encouragement. Also, we would like to thank **Mr. Bhavanishankar K,** Assistant Professor, Department of Computer Science & Engineering for being the guide of our project and guiding us in executing our ideas.

We would also like to thank the Coordinator **Mr. Devaraju B M,** Assistant Professor, Department of Computer Science & Engineering and all other teaching and non-teaching staff of Computer Science Department who has directly or indirectly helped me in the completion of the Project work.

Last, but not the least, we would hereby acknowledge and thank our parents who have been a source of inspiration and also instrumental in the successful completion of the Project work.

Sachin P **[1RN12CS079]**
Vishal U **[1RN12CS125]**
Shreyas R **[1RN12CS100]**
Kruti B **[1RN12CS037]**

TABLE OF CONTENTS

LIST OF FIGURES

Chapter 1

INTRODUCTION

1.1 Overview

"Number Plate Recognition" system uses Optical Character Recognition (OCR) software to convert images of vehicle registration numbers into information for real time or retrospective matching with law enforcement and other databases. This system is an integrated hardware-software device that reads the vehicles number plate and outputs the number plate number in ASCII - to some data processing system. This system is also known as License Plate Recognition (LPR) system.

It is believed that there are currently more than half a billion cars on the roads worldwide. All those vehicles have their Vehicle Identification Number as their primary identifier. The vehicle identification number is actually a license number which states a legal license to participate in the public traffic. All vehicle world-wide should have its license number - written on a license plate - mounted onto its body (at least at the back side) and no vehicle without properly mounted, well visible and well readable license plate should run on the roads. The license number is the most important identification data a computer system should treat when dealing with vehicles. If the data is already in the computer most of these tasks are rather easy to be carried out.

Suppose a company's security manager would require a system that precisely tells at every moment where the cars of the company are: in the garage or out on roads. By registering every single drive-out from and drive-in to the garage, the system could always tell which car is out and which is in. The key issue of this task is that the registration of the movement of the vehicles should be done automatically by the system, otherwise it would require manpower.

Concisely, Car Number Plate Recognition is automation of data input, where data equals the registration number of the vehicle. This system replaces, redeems the task of manually typing the plate number of the bypassing vehicle into the computer system. Fig 1.1 shows the major steps involved in the system integration.

This project is an offline application. Car images are taken by digital camera or traditional camera. Then a program written in MatLab is used to identify the Number Plate.

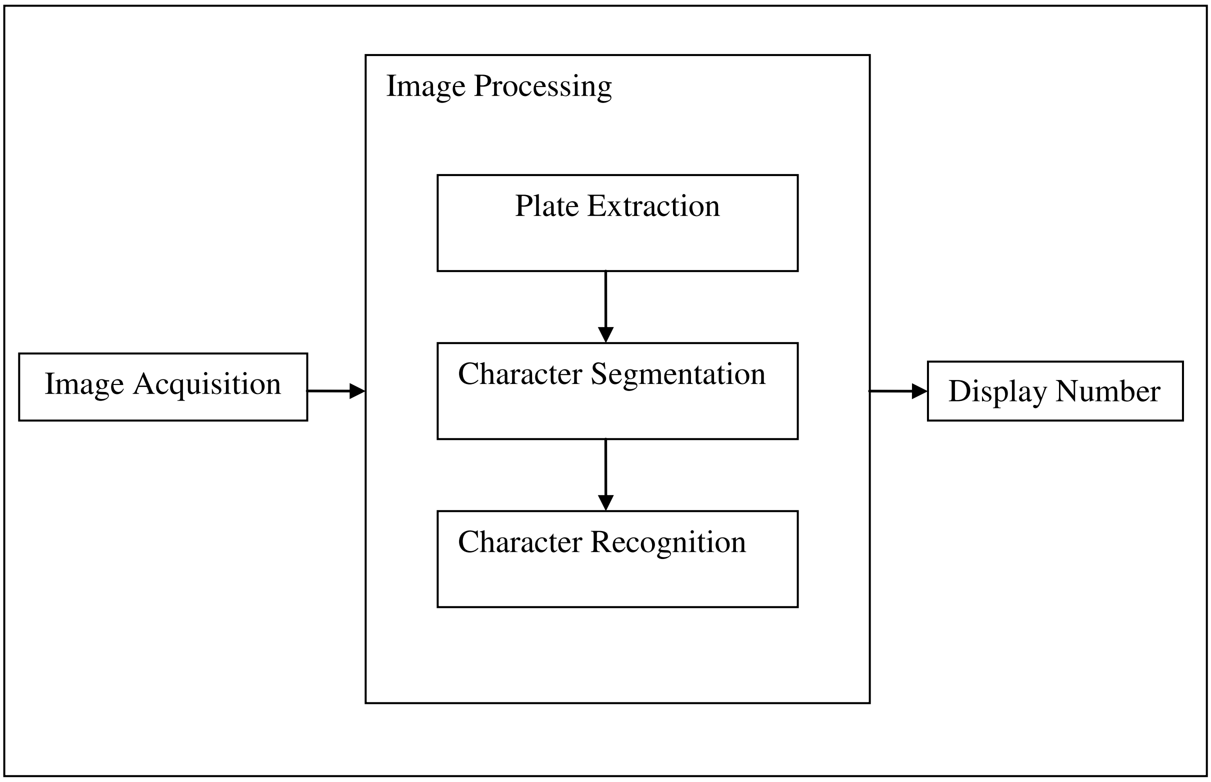

Fig. 1.1. Steps involved in NPR system

1.2 History of NPR

The Number Plate Recognition System was invented in 1976 at the Police Scientific Development Branch in the UK. Prototype systems were working by 1979 and contracts were let to produce industrial systems, first at EMI Electronics then at Computer Recognition Systems (CRS) in Wokingham, UK. Early trial systems were first used in Britain in 1979 with trial units placed on the A1 road and the Dartford tunnel. To minimize errors, license plate's character font, positioning and sizes were optimized for the character recognition systems used. The first arrest due to a detected stolen car was made in 1981. Early NPR systems were unable to read white or silver lettering on black background, as permitted on UK vehicles built prior to 1973.

There are a number of other possible difficulties that the software must be able to cope with. These include:

- Poor image resolution, usually because the plate is too far away but sometimes resulting from the use of a low-quality camera.
- Blurry images, particularly motion blur
- Poor lighting and low contrast due to overexposure, reflection or shadows
- An object obscuring (part of) the plate, quite often a tow bar, or dirt on the plate

- A different font, popular for vanity plates (some countries do not allow such plates, eliminating the problem)

1.3 Objectives

The overall objective of the project is to develop a system to recognize vehicle license plate from a car at a gate entrance of a parking lot. 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 captured. Once the vehicle license plate is captured, the characters will be recognized and displayed in a text file. 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, there is scope to improve or rewrite the algorithm to improve the accuracy. It is necessary to address a set of constraints and focus on the design of the algorithm to extract the vehicle license plate in order to improve the accuracy. There is definitely a lot more room for further improvement on this project.

1.4 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 vehicle license 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 front 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.
- Deal with only Indian License Plate.

1.5 Introduction to Digital Image Processing

Information carrying function of time is called signal. Real time signals can be audio or video (image) signals. Still video is called an image. Moving image is called a video. Difference between digital image processing and signals and systems is that time graph is not there in DIP. X and Y coordinates in Digital Image Processing (DIP) are spatial coordinates. Time graph is not there because photo doesn't change with time.

- **Image:** An image is defined as a two dimensional function f(x, y) where x and y are spatial coordinates and the amplitude 'f' at any point (x, y) is known as the intensity of image at that point.
- **Pixel:** A pixel (short for picture element) is a single point in a graphic image. Each such information element is not really a dot, nor a square but an abstract sample. Each element of the above matrix is known as a pixel where dark = 0 and light = 1. A pixel with only 1 bit will represent a black and white image. If the numbers of bits are increased then the number of gray levels will increase and a better picture quality is achieved.

All naturally occurring images are analog in nature. If the number of pixels is more, then the clarity is more. An image is represented as a matrix in DIP. In DIP we use only row matrices. Naturally occurring images should be sampled and quantized to get a digital image. A good image should have 1024*1024 pixels which is known as

1k * 1k = 1M pixel.

1.5.1 Fundamental Steps in DIP:

1. **Image Acquisition:** Digital image acquisition is the creation of digital images typically from a physical object. A digital image may be created directly from a physical scene by a camera or similar device. Alternatively it can be obtained from another image in an analog medium such as photographs, photographic film, or printed paper by a scanner or similar device. Many technical images acquired with tomographic equipment, side-looking radar, or radio telescopes are actually obtained by complex processing of non-image data.
2. **Image Enhancement:** The process of image acquisition frequently leads to image degradation due to mechanical problems, out-of-focus blur, motion, inappropriate illumination and noise. The goal of image enhancement is to start from a recorded image and to produce the most visually pleasing image.

3. **Image Restoration:** The goal of image restoration is to start from a recorded image and to produce the most visually pleasing image. The goal of enhancement is beauty. The goal of restoration is truth. The measure of success in restoration is usually an error measure between the original and the estimate image. No mathematical error function is known that corresponds to human perceptual assessment of error.
4. **Colour Image Processing:** Colour image processing is based on that any colour can be obtained by mixing 3 basic colours red, green and blue. Hence 3 matrices are necessary each one representing each colour.
5. **Wavelet and Multiresolution Processing:** Many times a particular spectral component occurring at any instant can be of particular interest. In these cases it may be very beneficial to know the time intervals these particular spectral components occur. For example, in EEGs the latency of an event-related potential is of particular interest. Wavelet transform is capable of providing the time and frequency information simultaneously, hence giving a time-frequency representation of the signal. Although the time and frequency resolution problems are results of a physical phenomenon (the Heisenberg uncertainty principle) and exist regardless of the transform used, it is possible to any signal by using an alternative approach called the Multi Resolution Analysis (MRA). MRA analyzes the signal at different frequencies with different resolutions. MRA is designed to give good time resolution and poor frequency resolution at high frequencies and good frequency resolution and poor time resolution at low frequencies.
6. **Compression:** Image compression is the application of data compression on digital images. Its objective is to reduce redundancy of the image data in order to be able to store or transmit data in an efficient form.
7. **Morphological Processing:** Morphological processing is a collection of techniques for DIP based on mathematical morphology. Since these techniques rely only on the relative ordering of pixel values not on their numerical values they are especially suited to the processing of binary images and grayscale images.
8. **Segmentation:** In the analysis of the objects in images it is essential that we can distinguish between the objects of interest and “the rest”. This latter group is also referred to as the background. The techniques that are used to find the objects of interest are usually referred to as segmentation techniques.
9. **Object Recognition:** The visual perception of familiar objects. In Computer vision, it is the task of finding a given object in an image or video sequence. Humans recognize

a multitude of objects in images with little effort, despite the fact that the image of the objects may vary somewhat in different viewpoints, in many different sizes / scale or even when they are translated or rotated. Objects can even be recognized when they are partially obstructed from view. This task is still a challenge for computer vision systems in general.

1.5.2. Image Sensing

The main parts of image sensing are:

1. Sensor (converts optical to electrical energy)
2. Digitizer (It converts analog signal to digital signal by sampling and quantization)

- **Interesting phenomena:** Our eye is capable of differentiating between various levels of intensity. The range of human eye is from scotopic threshold to glare limit which is about 10^10 lambards. It accomplishes this variation by changes in overall sensitivity and this phenomenon is called "brightness adaptation". Subjective brightness (intensity as perceived by human eye) is a log function of light intensity incident on the eye. A visual system cannot operate over such a large range simultaneously. Hence our eye cannot see very bright and very dim images simultaneously.
- **Light and electromagnetic spectrum:** Light is a part of electromagnetic spectrum that can be seen and sensed with the human eye. Light travels at a speed of 3*10^8 m/s. If anyone can cross this speed they can go into past or future. The visible light can be split up into VIBGYOR from range of violet (0.43 micrometers) to red (0.79micrometres). A substance which absorbs all the colours appears as black, no colour as white. A substance which reflects blue will appear as blue. Colour is the part of light spectrum which is not absorbed by human eye. Light that is void of colour is called monochromatic or achromatic light. The only property of this light is intensity or gray level.

1.5.3. Properties of Light:

1. **Radiance**: The total energy that flows from a light source is called radiance. Units are watts. Example of sources are sun, bulb etc.
2. **Luminance**: The amount of energy that an observer perceives from the source is called luminance. Units are lumens. Example is seeing the sun with black glasses.
3. **Brightness**: Brightness is an attribute of visual perception in which a source appears to emit a given amount of light. It has no units as it is practically impossible to measure.

1.5.4. Image Sensing and Acquisition

In image sensing light energy is converted into voltage. Image acquisition can be done using 3 principle sensor arrangements:

1. Single sensor
2. Line sensor/strip sensor
3. Array sensor

If something is changing more than 17 times per second or if the frequency is greater than 17 then we cannot differentiate it.

1.5.5. Simple Image Formation Model

Images are represented as 2-D functions f(x, y). The value of f(x, y) at any point (x, y) at any point (x, y) is a positive quantity.

$$0 < f(x, y) < \text{infinity}$$

f(x, y) depends on two parameters :

1. Amount of light incident on the scene = i(x, y)
2. Amount of light reflected from the scene = r(x, y)

$$f(x, y) = i(x, y) * r(x, y)$$

$$0 < i(x, y) < \text{infinity}$$

$$0 < r(x, y) < 1$$

In the last inequality 0 represents total absorption and no reflection (black) and 1 represents total reflection and no absorption (white).

$$0 < f(x, y) < \text{infinity}$$

$$imin < i(x, y) < imax$$

$$rmin < r(x, y) < rmax$$

$$imin * rmin < f(x, y) < imax * rmax$$

$$Lmin < l < Lmax$$

If you consider monochromatic light L = number of gray levels. If L = 30 then n = 5. The interval (Lmin – Lmax) is called grayscale which is shifted to (0, L-1) where l = 0 (black) & l = L-1 (is white). All intermediate stages are shades of gray.

All real time signals are analog in nature. But we need a digital image. But we acquire an analog image and digitize it. For this we need an A/D converter. To convert naturally occurring images into digital form we must digitize both coordinates and amplitudes. Digitizing coordinate values is called sampling and digitizing amplitude values is called quantization.

Hence the quality of digital image will depend on the number of samples (sampling) & the number of gray levels (quantization). More the samples better will be the quality.

1.5.6. Representing Digital Images

The result of sampling and quantization of an image is an array of numbers according to x and y coordinates which is nothing but a matrix of m rows and n columns. Columns and rows depend on number of samples. Brightness depends on number of bits used to represent each pixel. For an image of size m rows and n columns total number of pixels = m*n

If each pixel is represented using k bits then total amount of memory (in bits) required to store the image = m*n*k.

A 1M pixel size image requires 1024 rows and 1024 columns. If each pixel is represented using 8 bits then total amount of memory required to store the image = 1k*1k*1 bytes

1.5.7. Resolution:

Resolution is classified into 2 types:

- **Spatial Resolution:**
  It is the smallest discernable detail in the image. Consider vertical lines of width ‘w’ with spaces between also having width ‘w’. A line pair consists of one such line & its adjacent space. Hence width of line pair = 2w and there are 1/ (2*w) line pairs per unit distance. Thus spatial resolution is the number of discernable line pairs per unit distance. Example: 100 line pairs/unit distance.
- **Graylevel Resolution:**
  It is the smallest discernable change in gray level. The number of gray levels should be a power of 2. The commonly used number of gray levels is 256.

$$L = 256$$

$$2^k = 256$$

$$k = 8 \text{ (8 bits per pixel)}$$

1.6 Introduction to MATLAB

MATLAB is a high performance language for technical computing. It integrates computation visualization and programming in an easy to use environment

MatLab stands for MATrix LABoratory. It was written originally to provide easy access to matrix software developed by LINPACK (Linear System Package) and EISPACK (Eigen System Package) projects. MATLAB is therefore built on a foundation of sophisticated matrix software in which the basic element is matrix that does not require pre dimensioning

1.6.1 Applications of MATLAB

1. Mathematical Computation
2. Algorithm Development
3. Data Acquisition
4. Data Analysis, Exploration and Visualization
5. Scientific and Engineering Graphics

1.6.2 Features of MATLAB

1. Advance algorithm for high performance numerical computation ,especially in the Field matrix algebra.
2. A large collection of predefined mathematical functions and the ability to define one's own functions.
3. Two-and three dimensional graphics for plotting and displaying data.
4. A complete online help system.
5. Powerful, matrix or vector oriented high level programming language for individual applications.
6. Toolboxes available for solving advanced problems in several application areas.

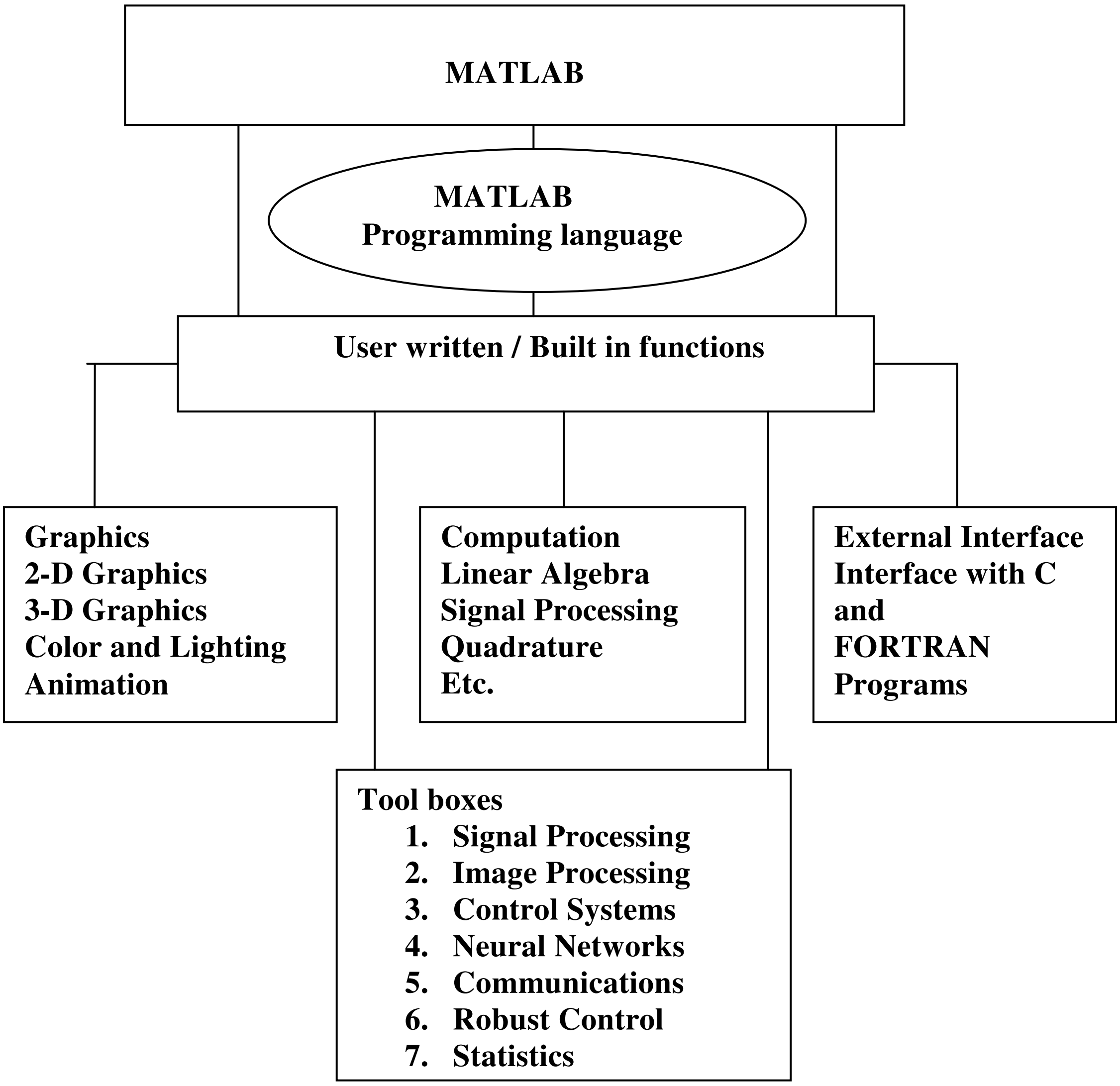

Fig.1.2. Features and capabilities of MATLAB

1.6.3. The MATLAB System

The MATLAB system consists of five main parts:

1. **Development Environment.**

   This is the set of tools and facilities that help you use MATLAB functions and files. Many of these tools are graphical user interfaces. It includes the MATLAB desktop and Command Window, a command history, an editor and debugger, and browsers for viewing help, the workspace, files, and the search path.

2. **The MATLAB Mathematical Function Library.**

   This is a vast collection of computational algorithms ranging from elementary functions, like sum, sine, cosine, and complex arithmetic, to more sophisticated functions like matrix inverse, matrix Eigen values, Bessel functions, and fast Fourier transforms.

3. **The MATLAB Language.**
   This is a high-level matrix/array language with control flow statements, functions, data structures, input/output, and object-oriented programming features. It allows both "programming in the small" to rapidly create quick and dirty throw-away programs, and "programming in the large" to create large and complex application programs.
4. **Graphics.**
   MATLAB has extensive facilities for displaying vectors and matrices as graphs, as well as annotating and printing these graphs. It includes high-level functions for two-dimensional and three-dimensional data visualization, image processing, animation, and presentation graphics. It also includes low-level functions that allow you to fully customize the appearance of graphics as well as to build complete graphical user interfaces on your MATLAB applications.
5. **The MATLAB Application Program Interface (API).**
   This is a library that allows you to write C and FORTRAN programs that interact with MATLAB. It includes facilities for calling routines from MATLAB (dynamic linking), calling MATLAB as a computational engine, and for reading and writing MAT-files.

1.6.4. Starting MATLAB

On Windows platforms, start MATLAB by double-clicking the MATLAB shortcut icon on your Windows desktop. On UNIX platforms, start MATLAB by typing mat lab at the operating system prompt. You can customize MATLAB startup. For example, you can change the directory in which MATLAB starts or automatically execute MATLAB statements in a script file named startup.m.

1.6.5. MATLAB Desktop

When you start MATLAB, the MATLAB desktop appears, containing tools (graphical user interfaces) for managing files, variables, and applications associated with MATLAB. The following illustration in Fig 1.3., shows the default desktop. You can customize the arrangement of tools and documents to suit your needs and for or more information about the desktop tools.

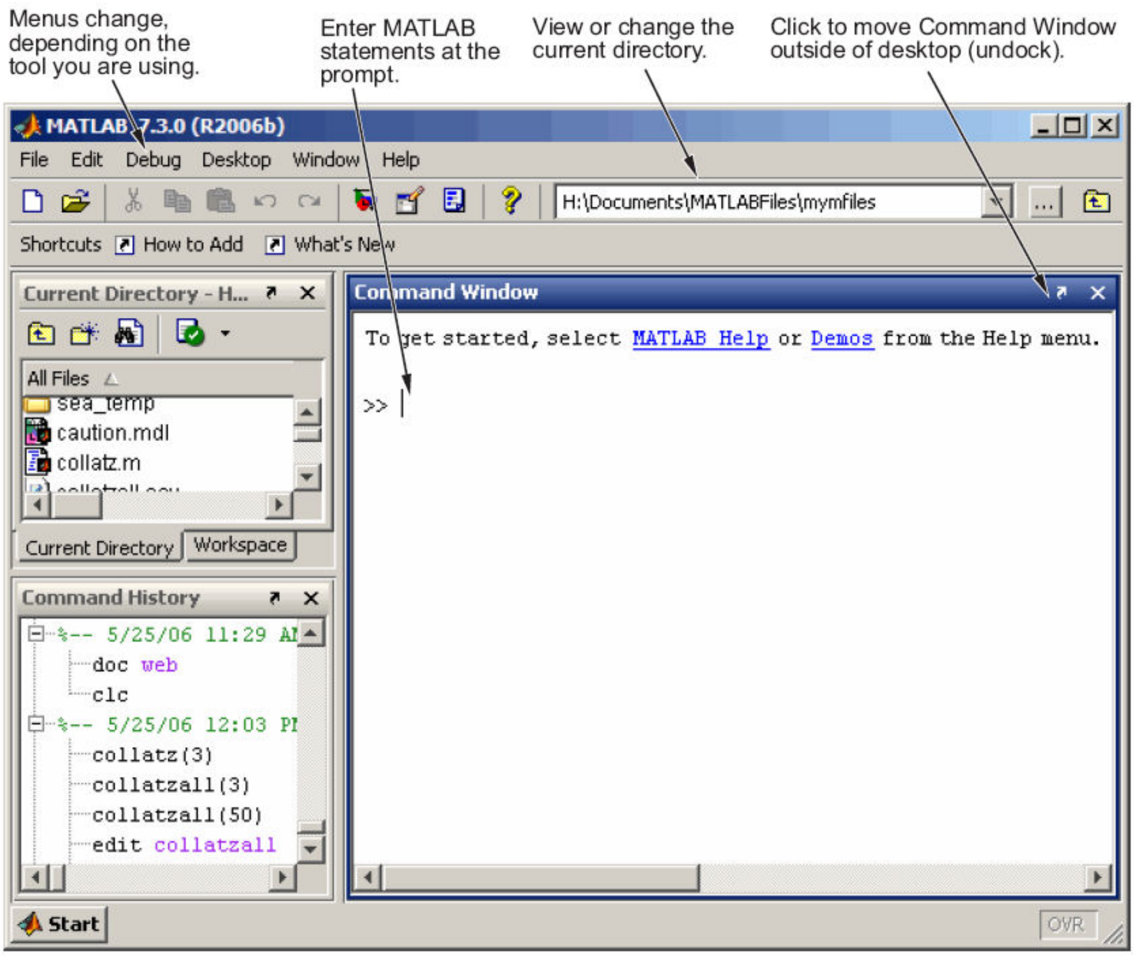

Fig 1.3. MATLAB desktop

1.7 Applications of NPR

- **Car Parking**: Car parks in some countries like Spain or Greece, the law binds the recording of the number plate on the ticket. In this way, the number plate and the ticket number are linked. Thus, car parks (Fig. 1.4) improve their management.

Fig. 1.4: Car Parking

These are some direct benefits: Number plate register: The ticket number, day and the time, and the number plate are linked and registered in the management application. Finding a lost ticket: Thanks to the registry of the number plates it is possible to find a lost ticket and to receive the correct amount.

Ticket interchange is avoided: It is possible to block the exit of a vehicle, if the number plate of the vehicle does not match with the number plate in the entrance ticket.

- **Access Control**: The NPR equipment has been used for the access control of vehicles, it was thought as one more tool that allows increasing the security (Fig. 1.5). The client application could control users through personal cards, and the NPR allows vehicles control.

Fig. 1.5: Access Control

Nowadays, the NPR equipment’s are used to automatize the accesses of vehicles.

- NPR access control: These are the main advantages to incorporate NPR equipment in access control: Security increased: Integrating the NPR technology in access control applications together with the traditional control devices, allows vehicle and people control. Thanks to this convention the security is increased.
- Dynamic access of vehicles: Automatizing vehicle access is possible through NPR equipment. If the data base knows the vehicle, the client application will open the barrier automatically. By contrast, if the data base does not know the vehicle, it will not open the barrier.
- Vehicle images: It is possible to store the image used for the NPR equipment during the recognition process. It allows having more information about the vehicle in the client application.

- **Traffic Control**: Some NPR equipment are able to recognize the number plate of vehicles that circulate up to 200 km/h with a reliability of 95%. Thanks to this, the use of NPR equipment for traffic control has increased significantly in the last years.
  These are some examples of NPR application on traffic control (Fig. 1.6): Detecting vehicles in a black list it is possible to control vehicles that are in search and capture, through the installation of the NPR equipment in the main accesses to cities, such as highways and roads.

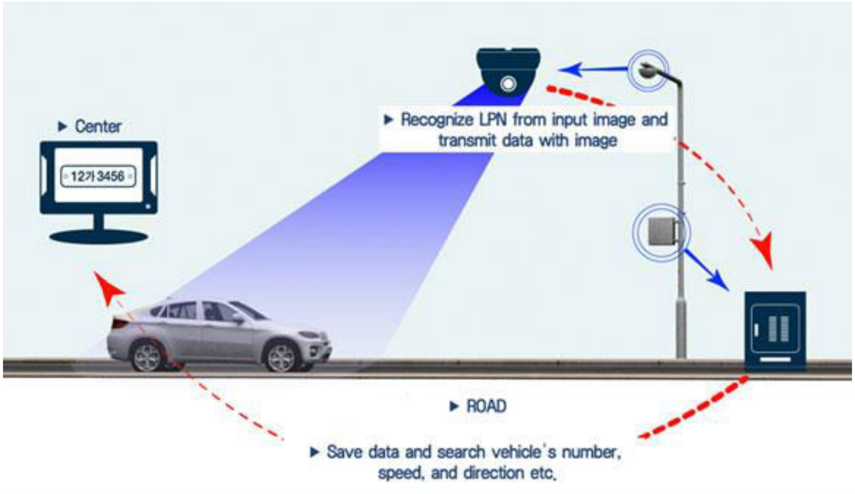

Fig. 1.6: NPR Control Point

- **Average Speed Control**: The majority of speed control devices, such as radars or speed traps, control the vehicle instantaneous speed. With the NPR equipment it is possible to control the average speed during an itinerary. By means of the installation of two NPR equipment's in different points in the same lane, it is possible to make two consecutive recognitions of the number plate and to calculate the average speed of the vehicle (Fig. 1.7).

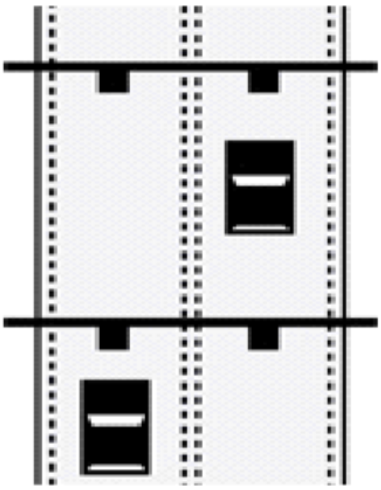

Fig. 1.7: Average Speed Control

- **Traffic Optimization**: It is important to improve the vehicles mobility during rush hours and traffic jam. The installation of NPR equipment allows to know how many time a vehicle spends to cross an itinerary (Fig. 1.8). This way, the average time can be informed.

Fig. 1.8: Traffic Optimization

- **Toll Enforcement**: Although the use of NPR equipment in tolls is a logical application, today, the NPR equipment are only used to control the vehicles that circulate in the toll (Fig 1.9), as a data base that increases the security

Fig. 1.9: Toll Enforcement

- **Red Light Control**: To jump a red light is a really dangerous infraction with consequences for others drivers and pedestrians. Nowadays, it is possible to control when a vehicle jumps a red light (Fig 1.10). This system combines sensors and image capture equipment, such as cameras or video recorders. The NPR systems reinforce the red light control systems. Thanks to NPR technology the fine process can be done automatically.

Fig. 1.10: Red Light Control

Chapter 2

LITERATURE SURVEY

In [6], license plate recognition methods are: (1) Image Acquisition: By digital camera (2) License Plate Extraction: vertical edge detection by Sobel algorithm, filtering by seed filling algorithm, vertical edge matching (3) Segmentation (4) Character Recognition: Normalization, Template matching using hamming distance approach.

Result: License Plate Extraction: 587/610, overall system efficiency: 95%

Limitation: Detection only for white, black, red, and green colour plate or numbers.

In [7], (1) Extraction of plate region: edge detection algorithms and smearing algorithms (2) segmentation of Characters: smearing algorithms, filtering and some morphological algorithms (3) recognition of plate characters: template matching.

Result: extraction of plate region: 97.6% segmentation of the characters: 96%, recognition unit: 98.8%, Overall system performance: 92.57% recognition rate.

Limitation: Limited to recognition of car license plate only, and this system is designed for the identification of Turkish license plates.

In [8], recognition steps are as follow: (1) Image Enhancement: by histogram equalization method. (2) Structuring Elements : by thickening. (3) Hat transformations: which is use for contrast, enhancement (top has & bottom has) setting. (4) Morphological Operations like dilation and erosion. (5) Plate region confirmation. (6) Character Segmentation and Recognition by neuron implementation model.

Result: 95%. Work very well in the real time environment.

Limitation: For the overall system, while the some more work is to be done to make the technique more efficient.

In[9], recognition by (1) Target recognition: by using feature-salience theory, features of license plates. (2) License plate locating by Hough transform (HT). (3) Recognizing license characters by different steps like binarization, noise removal, and orientation adjustment, Optical Character Recognition.

Result: in this paper, the success rate for the identification with the set of 1144 license plates is 95.7%. Combining this rate with the location success (97.3%), the overall rate of success for our LPR algorithm is 93.1%.

Limitation: This system is intended for the recognition of Chinese license plates only.

In[10], mainly focused on Edge Detection(Sobel Edge Detection)technique and then filtering of noise by Median Filter, Smoothing, Connector, Masking, ,and then Colour Conversation is done.

Result: All over system result is not mention in this paper.

Limitation: we can see that the detection is not that clear and proper, which we find, is due to improper light segment or varying illumination effects.

In [11], (1) Extracting the Plate region, edge detection algorithm and vertical projection method are used. (2) In segmentation part filtering, thinning and vertical and horizontal projection are used. And finally, (3) Chain code concept with different parameter is used for recognition of the characters.

Result: Final system Efficiency: 98%.

Limitation: The proposed method is mainly designed for real-time Malaysian license plate.

In [12], involves three approaches: (1) In plate Localization Noise alleviation, Changing color space, Intensity dynamic range modification, Edge detection, Separating objects from background, Finding connected component, Candidate selection, all above process are used (2) In segmentation part multistage model are used. (3) For the recognition Artificial Feed forward neural network is used.

Result: The method achieved accuracy over 91% for localizing plates.

Limitation: Detection only for English and Parisian number plate.

In [13], For the Number plate recognition first image conversion in binary and apply to neural network, and apply mpl algorithm, then detection individual symbol, by matrix mapping.

Result: Training by this approach obtained 96.53% average recognition rate using double hidden layer and 94% using single hidden layer.

Limitation: The captured image 2-3 meters taken away from the cameras.

In[14], (1) Preprocessing of Image by histogram equalization. (2) Extraction of plate region by edge detection algorithm (canny operator) and Plate Area Detection by various morphological operations. (3) Segmentation of characters by connected component, bounding box method, Median filter, all above methods.

Result: Observed final result as Extraction: 71/78 which gives 91.02% efficiency, Segmentation 69/78 which gives 88.46%efficiency. Overall accuracy of our system is 89.74%.

Limitation: Proposed method is sensitive to the angle of view, physical appearance and environment conditions.

In [15], Given All 3 process by 2D Hear after the discrete wavelet Transform technique: (1) Locate and extract the license-plate. (2) Train of the license-plate. (3) Real time scan recognize of the license-plate.

Result: Recognition rates: 93.0%. Advantage of this approach is Hear Discrete Wavelet Transform are that it each time transform only needs 1/4 of the original image. Hence, this method can have fast execution speed.

Limitation: Disadvantage is that, in this system only specified camera are used like, using the Casio Exilim, 10.1 Mega Pixels Digital Camera Ex-S10, adjusting the resolution 480 x 640 for photography vehicle license plates.

In [16], (1) PVW approach is used in this orientation, ratio of scale to character height, and relative position in the character region are done by clustering. (2) Visual word matching by comparing the extracted SIFT features and histogram approach is used. (3) For license plate location.

Result: This technique achieves a 93.2% “true” detection rate.

Limitation: The weakness of this approach is that it may fail when the license plate resolution is too low, or when the distortion from the observation angle is too severe.

In [17], (1) Detect a license plate region by vertical or a horizontal edge based method (2) Pre-processing: is also needed in this approach, so first converted in to binary image then eliminate noise using morphological operation. (3) Character segmentation by thresholding method. (4) Feature extraction and character recognition by Euler number formation.

Result: In this approach, skewness is not present in the detected vehicle number plate compare to other methods.

Limitation: It limits the efficiency of the total system.

In [18], (1) Plate extraction by Mathematical Morphology approach and then Dilation and Erosion is applied to image. (2) Segmentation by Structuring Elements approach, Median Filter technique and Edge Detection Methods are used. (3) Character extraction by Preprocessing, Text/non-text classification approach .result derived by this paper is as given, Real Time Data: 100 Images, Correctly Detected: 93.

Results: Accuracy 93%.

Limitation: It says that very much damaged plate cannot be recognized.

Chapter 3

MATHEMATICAL FRAMEWORK

3.1. RGB to Grayscale Conversion

Humans perceive color through wavelength-sensitive sensory cells called cones. There are three different types of cones, each with a different sensitivity to electromagnetic radiation (light) of different wavelength. One type of cone is mainly sensitive to red light, one to green light, and one to blue light. By emitting a controlled combination of these three basic colors (red, green and blue), and hence stimulate the three types of cones, we are able to generate almost any perceivable color. This is the reasoning behind why color images are often stored as three separate image matrices; one storing the amount of Red (R) in each pixel, next the amount of Green (G) and finally the amount of Blue (B). We call such color images as stored in an RGB format. Fig.3.1 represents the human perception of color Image.

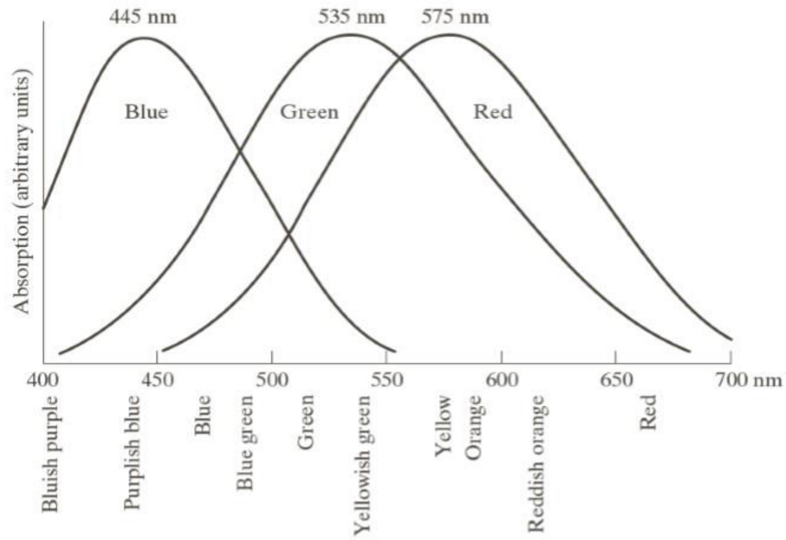

Fig.3.1 Absorption of light by the red, green and blue cones in the human eye as a function of wavelength

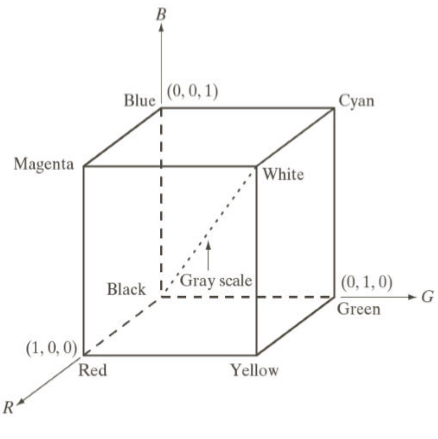

Fig.3.2. Schematic of the RGB color cube. Points along the main diagonal have gray values, from black at the origin to white at the point (1, 1, 1)

3.1.1. Acquiring an RGB Color Image

A color image can be acquired by using three filters, sensitive to red, green, and blue respectively. When we view a color scene with a monochrome camera equipped with one of these filters, the result is a monochrome image whose intensity is proportional to the response of that filter. Repeating this process with each filter produces three monochrome images that are the RGB component images of the color scene. In practice, RGB color image sensors usually integrate this process into a single device.

3.1.2. Normalization

Normalization transforms an n-dimensional grayscale image with intensity values in the range (Min,Max), into a new image with intensity values in the range (newMin,newMax).

$$I : \{\mathbb{X} \subseteq \mathbb{R}^n\} \to \{\mathrm{Min}, .., \mathrm{Max}\} \quad ..(1)$$

$$I_N : \{\mathbb{X} \subseteq \mathbb{R}^n\} \to \{\mathrm{newMin}, .., \mathrm{newMax}\} \quad ..(2)$$

The linear normalization of a grayscale digital image is performed according to the formula

$$I_N = (I - \mathrm{Min})\frac{\mathrm{newMax} - \mathrm{newMin}}{\mathrm{Max} - \mathrm{Min}} + \mathrm{newMin} \quad ..(3)$$

3.1.3. Pseudo-color (False color) Image Processing

It consists of assigning colors to gray values based on a specified criterion. The term pseudo or false color is used to differentiate the process of assigning colors to monochrome images from the processes associated with the true color images. The principle use of pseudo-color is for human visualization and interpretation of gray scale events in an image or sequence of images.

In grayscale images, however, we do not differentiate how much we emit of the different colors, we emit the same amount in each channel. What we can differentiate is the total amount of emitted light for each pixel; little light gives dark pixels and much light is perceived as bright pixels. Intensity level slicing is one of the simplest examples of pseudo color image processing.

- **Intensity (Density) level slicing technique**

  It is one of the simplest example of pseudo-color image processing. If an image is interpreted as a 3D function, the method can be viewed as one of placing planes parallel to the coordinate plane of the image; each plane then slices the function in the area of intersection. If a different color is assigned to each side of the plane as shown in the figure, any pixel whose intensity level is above the plane will be coded with one color, and any pixel below the plane will be coded with the other. Levels that lie on the plane itself may be arbitrarily assigned one of the two colors. The result is a two color image whose relative appearance can be controlled by moving the slicing plane up and down the intensity axis. Fig.3.3 (a) shows geometrical representation of allotting gray levels for an RGB image.

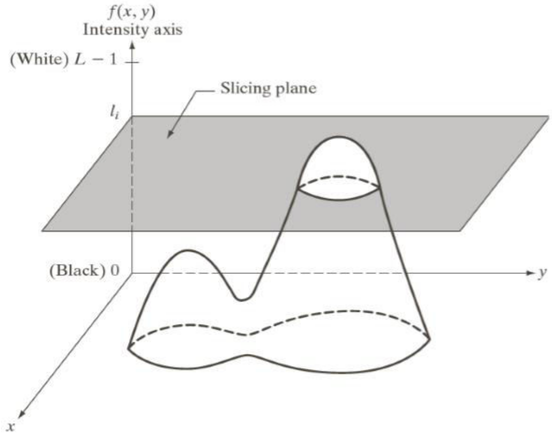

Fig.3.3 (a) Geometrical representation of the Intensity level slicing technique

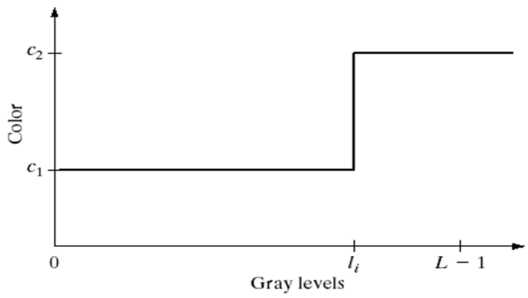

Fig.3.3 (b) an alternative representation of the Intensity level slicing technique

Where C_1 and C_2 are the color levels and [0, L-1] represents the gray levels.

Fig.3.3 (b) shows an alternative representation of the intensity slicing technique. According to the mapping function shown in the figure, any input intensity level is assigned one of two colors, depending on whether it is above or below the value of li. When more levels are used, the mapping function takes on a staircase form.

3.1.4. Methods of Conversion

The RGB to Grayscale conversion can be done three methods. They are

1. The lightness method: The lightness method averages the most prominent and least prominent colors: (max (R, G, B) + min (R, G, B)) / 2.
2. The average method simply averages the values: (R + G + B) / 3.
3. The luminosity method is a more sophisticated version of the average method. It also averages the values, but it forms a weighted average to account for human perception.

Humans are more sensitive to green than other colors, so green is weighted most heavily. The formula for luminosity is 0.2126 R + 0.7252 G + 0.0722 B.

The lightness method tends to reduce contrast. The luminosity method works best overall and is the default method used to change an image from RGB to grayscale as shown in Fig. 3.4. However, some images look better using one of the other algorithms. And sometimes the three methods produce very similar results.

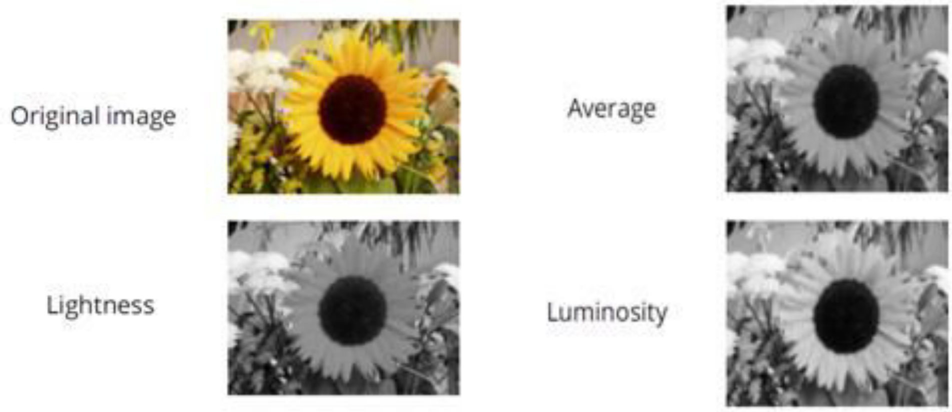

Fig.3.4 Simulation results of RGB to Gray scale conversion

3.2. Median filtering

Median filtering is a nonlinear process useful in reducing impulsive, or salt-and-pepper noise. It is also useful in preserving edges in an image while reducing random noise. Impulsive or salt-and pepper noise can occur due to a random bit error in a communication channel as shown in Fig. 3.5. In a median filter, a window slides along the image, and the median intensity value of the pixels within the window becomes the output intensity of the pixel being processed. Mathematically,

$$\hat{f}(x,y) = \underset{(s,t)\in S_{xy}}{\text{median}} \{ g(s,t)\} \qquad ..(4)$$

Like low pass filter, median filter smoothens the image and is thus useful in reducing noise. Unlike low pass filter, median filter can preserve discontinuities in a step function and can smoothen few pixels whose values differ significantly from their surroundings without affecting the other pixels.

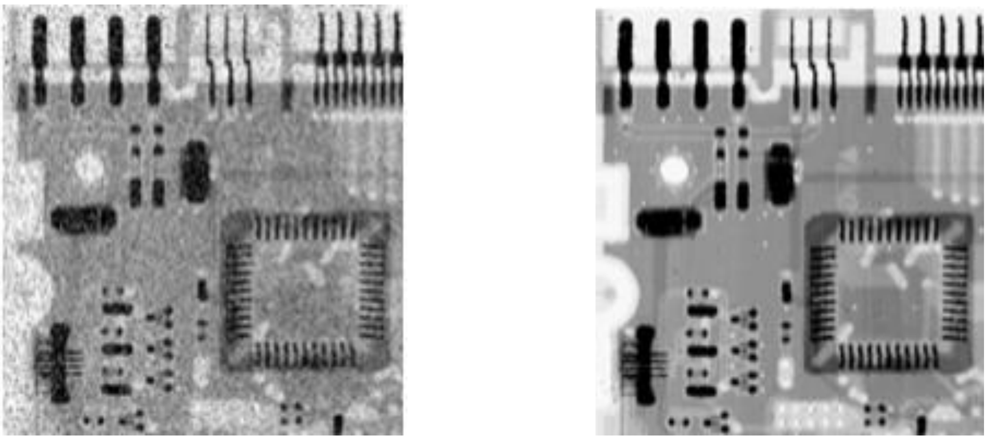

Fig.3.5. (a) X-ray image of circuit board corrupted by salt-and-pepper noise.
(b) Noise reduction with a 3*3 median filter

In order to perform median filtering at a point in an image, we first sort the values of the pixel under consideration and its neighbors, determine their median, and replaces the pixel under consideration by the median value.

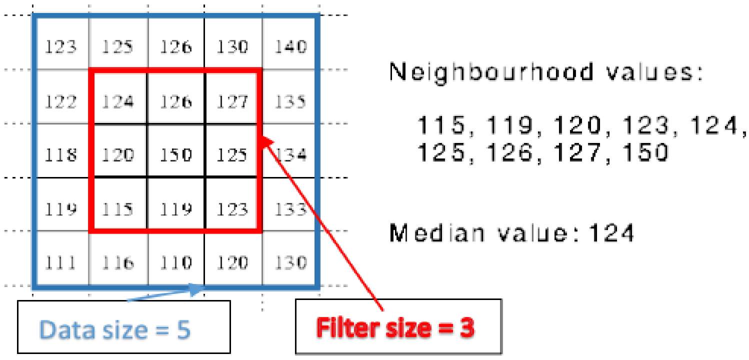

Fig.3.6. Median filtering using 3×3 neighborhood.

Consider the example shown in Fig.3.6, suppose that a 3×3 neighborhood has values 124, 120, 115, 126, 150, 119, 127, 125, and 123. These values are sorted as (115, 119, 120, 123, 124,125,126,127 and 150), which results in a median of 124. The median filter replaces the center pixel 150 by 124 which is the median of the nine values.

If the pixel under consideration is at the edges, then zeros are padded to form a 3×3 neighborhood. Then the median value is determined which replaces the center pixel.

Thus, the principal function of median filters is to force points with distinct gray levels to be more like their neighbors. In fact, isolated clusters of pixels that are light or dark with respect to their neighbors, and whose area is less than $n^2/2$ (one-half the filter area), are eliminated by an n×n median filter. In this case “eliminated” means forced to the median intensity of the neighbors. Larger clusters are affected considerably less.

3.3. Thresholding

This technique is based upon a simple concept. A parameter Θ called the brightness threshold is chosen and applied to the image *a* [*m, n*] as follows:

If a [m, n] ≥ Θ	a [m, n] = object = 1
else	a[m, n] = background = 0

This version of the algorithm assumes that we are interested in light objects on a dark background. For dark objects on a light background we would use:

If a [m, n] < Θ	a [m, n] = object = 1
else	a [m, n] = background = 0

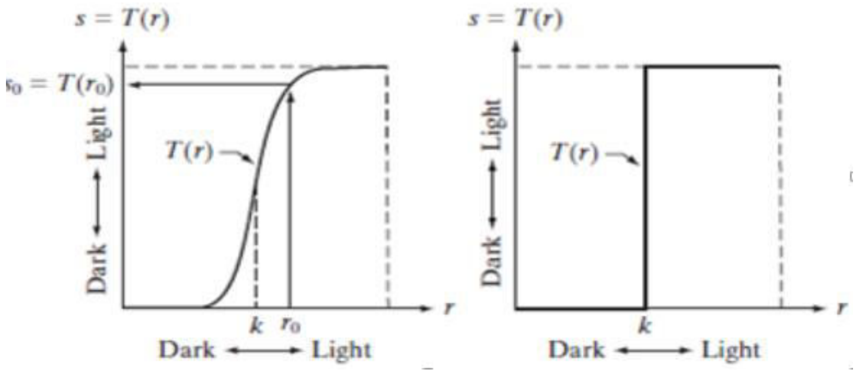

Fig. 3.7. Intensity transformation function. (a) Contrast stretching function

(b) Thresholding function

The central question in thresholding then becomes: how do we choose the threshold Θ? While there is no universal procedure for threshold selection that is guaranteed to work on all images, there are a variety of alternatives.

3.3.1. Fixed threshold

One alternative is to use a threshold that is chosen independently of the image data. If it is known that one is dealing with very high-contrast images where the objects are very dark and the background is homogeneous and very light, then a constant threshold of 128 on a scale of 0 to 255 might be sufficiently accurate. By accuracy we mean that the number of falsely-classified pixels should be kept to a minimum. Thresholding does not have to be applied to entire images but can be used on a region by region basis.

In each region a threshold is calculated and the resulting threshold values are put together (interpolated) to form a thresholding surface for the entire image. The regions should be of "reasonable" size so that there are a sufficient number of pixels in each region to make n estimate of the histogram and the threshold. The utility of this procedure-like so many others-depends on the application at hand.

Fig.3.8. (a) Original image (b) Result of Thresholding.

3.4. Segmentation

Image segmentation is the process of partitioning a digital image into multiple segments (sets of pixels, also known as super pixels). The goal of segmentation is to simplify and/or change the representation of an image into something that is more meaningful and easier to analyze.

The following are different categories of segmentation:

- **Threshold based segmentation**: Histogram thresholding and slicing techniques are used to segment the image. They may be applied directly to an image, but can also be combined with pre- and post-processing techniques.
- **Edge based segmentation**: With this technique, detected edges in an image are assumed to represent object boundaries, and used to identify these objects.
- **Region based segmentation.** Where an edge based technique may attempt to find the object boundaries and then locate the object itself by filling them in, a region based technique takes the opposite approach, by (e.g.) starting in the middle of an object and then "growing" outward until it meets the object boundaries.

3.4.1. Threshold based segmentation

Thresholding is probably the most frequently used technique to segment an image. The thresholding operation is a grey value remapping operation *g* defined by (5):

$$g(v) = \begin{cases} 0\,; if\ v < t \\ 1; if\ v \geq t \end{cases} \qquad ..(5)$$

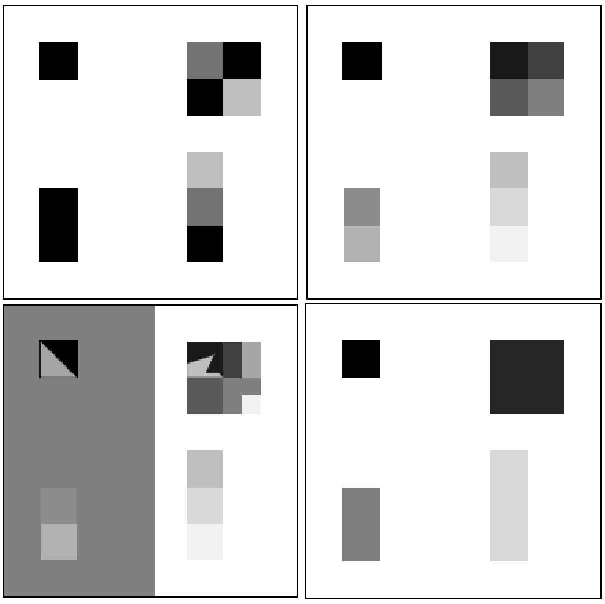

Fig.3.9. An original image is shown at the top left. If it is known that this image contains only uniformly sized squares, then the image on the top right shows the correct segmentation. Each segment has been indicated by a unique grey value here. The bottom left and right images show examples of over segmentation and under segmentation respectively.

Where *v* represents a grey value, and t is the threshold value. Thresholding maps a grey-valued image to a binary image. After the thresholding operation, the image has been segmented into two segments, identified by the pixel values 0 and 1 respectively.

3.4.2. Threshold selection

Many methods exist to find a suitable threshold for segmentation. The simplest method is the interactive selection of a threshold by the user –possibly with the aid of the image

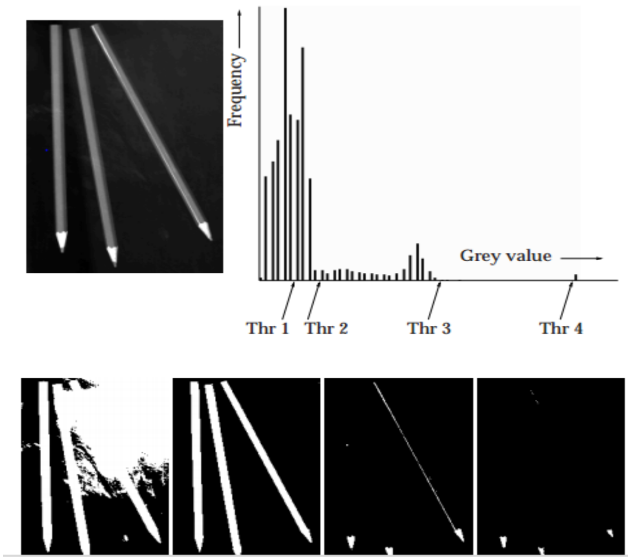

Fig 3.10: Example of threshold selection from the histogram.

Fig.3.10 represents the different threshold selection from the histogram. Top row: original image and histogram. Four special grey values (indicated by Thr 1, 2, 3, and 4) have been chosen. The bottom row shows the respective thresholding results at each of the values. At a first glance, the original image appears to have only three grey values. But the histogram shows that the grey value distribution is more diffuse; that the three basic values are in fact spread out over a certain range. Because the background grey values occur most frequently, we expect all of the large values in the left part of the histogram to correspond to the background. This is indeed the case. The result of threshold 1 shows that the peak between thresholds 1 and 2 also corresponds to background values. The result of threshold 2 shows the desired segmentation; every grey value to the right of threshold 2 corresponds to the pencils. Threshold 4 shows that the right-most little peak corresponds to the bright grey value in the tips of the pencils.

We will give an example using one threshold, i.e., two segments. For simplicity, we normalize the histogram H (v) of the image to a proper distribution h (v), i.e., %h (v) =1. This can be achieved by setting $h(v)$ = H (v)/n for all v, where n is the total number of pixels in the image. The variance of the grey values σ^2 in the image then (by definition) equals

$$\sigma^2 = \sum_v (v - \mu)^2 h(v) \qquad ..(6)$$

Where $\mu = \sum_v v\, h(v)$is the mean grey value of the image. If we segment the image with threshold value t into a background segment 0 and an object segment 1, then the variance of grey values within each segment (respectively σ0 and σ1) is

$$\sigma_0^2 = \sum_{v<t} (v - \mu_0)^2 h(v) \qquad ..(7)$$

$$\sigma_1^2 = \sum_{v\geq t} (v - \mu_1)^2 h(v) \qquad ..(8)$$

Where $\mu_0 = \frac{1}{h_0}\sum_{v<t} vh(v)$ and $\mu_1 = \frac{1}{h_1}\sum_{v\geq t} vh(v)$are the mean grey values of the respective segments 0 and 1. The probabilities h0 and h1 that a randomly selected pixel belongs to segment 0 or 1 are

$$h_0 = \sum_{v<t} h(v) \quad , \quad h_1 = \sum_{v\geq t} h(v) \qquad ..(9)$$

Note that $h_0 + h_1 = 1$. The total variance within segments σ^2_w is

$$\sigma_w^2 = h_0\sigma_0^2 + h_1\sigma_1^2 \qquad ..(10)$$

This variance only depends on the threshold value t: $\sigma^2_w = \sigma^2_w(t)$. This means that we can find the value of t that minimizes the variance within segments by minimizing $\sigma^2_w(t)$. The variance between the segments 0 and 1, σ^2_b, is

$$\sigma_b^2 = h_0(\mu_0 - \mu)^2 + h_1(\mu_1 - \mu)^2 \qquad ..(11)$$

Again, this variance is only dependent on the threshold value t. finding the t that maximizes σ^2_b maximizes the variance between segments. A hybrid approach that attempts to maximize σ^2_b while minimizing σ^2_w is to find the threshold t that maximizes the ratio σ^2_b / σ^2_w. If more than two segments are required, the method described above can be extended to use multiple thresholds. The variances σ^2_w and σ^2_b will then be functions of more than one threshold, so we need multi-dimensional optimization to find the set of optimal thresholds. Since this is especially cumbersome if the number of segments is large, a more practical algorithm that minimizes the variances within segments is often used, an iterative algorithm known as K-means clustering.

3.4.3. Pixel based Segmentation

In order to perform pixel based segmentation, the original image has to be converted to black and white image. Here we perform segmentation based on the pixel transition from 0 to 1 and vice versa.

Fig.3.11.Vehicle number plate.

From the Fig.3.11, it can be observed that by adding the pixels along the columns, the width of the character can be estimated. Further, by adding the pixels along the rows, the height of the character can be estimated. In this work, the row sum and the column sum are plotted and by observing the transitions, each character is extracted from the number plate. The simulation results and further explained.

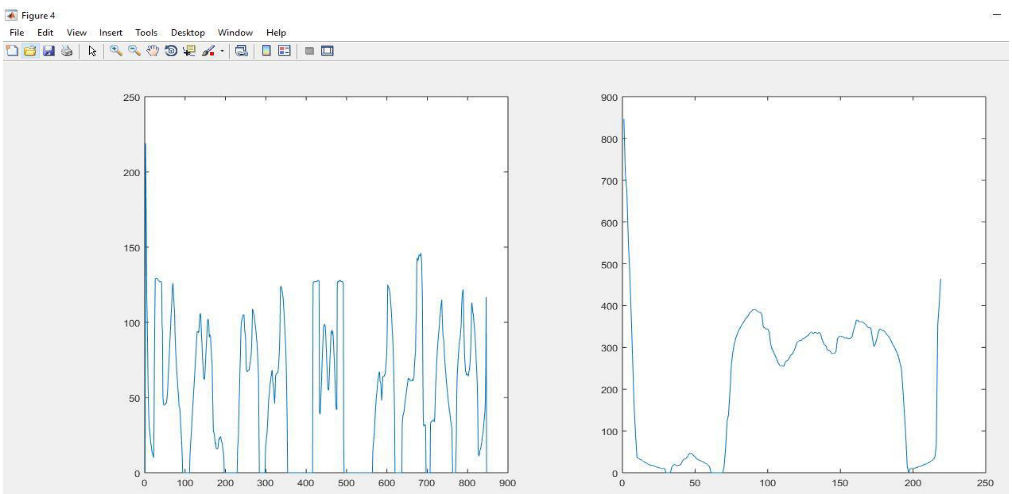

Fig.3.12. Simulation result of adding pixels (a) columns. (b) Rows.

Fig.3.12. represents the total sum of columns of the image and total sum of rows of that image. Based on this plot we can notice that there is a huge variation in the amplitudes of the pixels of the image, which helps in segmentation process. When the image is traversed vertically there is a continuous variation from white to black and vice versa whereas during horizontal traversing, initially image is dark and later there is a variation of pixels of the image from black to white and vice versa until the presence of character and after which the pixel values are zero which representing dark. Therefore the initial dark pixels indices and the present pixel indices helps in extracting the character from the image and the same logic is applicable in extracting other characters in the image . The pixel transition indices can be found out by the plotting a sum of rows vs sum of columns for a black and white image as shown in Fig.3.12.

Fig.3.13. Simulation result of character segmentation.

The segmented character image is as shown in above Fig.3.13. Therefore by using the above technique pixel based segmentation is one of the most important techniques that can be used to extract any features of the black and white image.

3.5. Correlation

When two sets of data are strongly linked together we say they have a High Correlation. Correlation is a technique that can show how strongly a pair of signals is related. The mathematical equation for Correlation is mentioned below:

$$\mathrm{Corr}(A,B) = \frac{\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} A(x,y)*B(x,y)}{\sqrt{(\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} A(x,y)*A(x,y))*(\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} B(x,y)*B(x,y))}} \quad ..(12)$$

Example 1:

$$A = \begin{pmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \qquad \text{AND} \qquad B = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

$$A.B = \begin{pmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} * \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} = 1+0+1+1+1+1+1+1+1 = 8$$

$$A.A = \begin{pmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} * \begin{pmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} = 1+0+1+1+1+1+1+1=8$$

B.B = $\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} * \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$ =1+1+1+1+1+1+1+1+1=9

$Corr(A,B) = 8/\sqrt{8*9} = 0.9428$ ..(13)

The above two matrices indicate that they are highly correlate.

Example 2:

A = $\begin{pmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$ AND B = $\begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$

A.B = $\begin{pmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} * \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$ = 0+0+1+0+0+0+0+0+0= 1

A.A = $\begin{pmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} * \begin{pmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$ = 1+0+1+1+1+1+1+1+1=8

B.B = $\begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} * \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$ = 0+0+1+0+0+0+0+0+0=1

$Corr(A,B)\ 2/\sqrt{8*2} = 0.5$ ..(14)

The two matrices indicate that they are less correlated.

Chapter 4

SYSTEM REQUIRMENTS AND SPECIFICATIONS

4.1 Functional Requirements

Functional Requirement defines a function of a software system and how the system must behave when presented with specific inputs or conditions. These may include calculations, data manipulation and processing and other specific functionality. In this system following are the functional requirements:-

- The system should be able to detect the number plate in the image.
- The system must recognize any Indian standard number plate.
- The system must support the addition of new templates for recognition.
- The system must improve accuracy on addition of new templates.

4.2 Non-Functional Requirements

1. **Performance**: The response time (time between input given and output recognized) should not exceed 3 seconds.
2. **Portability**: The system should work in varying operating environments.
3. **Maintainability**: It should be easy to maintain the system.
4. **Usability**: It should be easy for a user to use the system and it must take less time for a new user to learn about the usage of the system.
5. **Robustness**: It is the ability of a computer system to cope with errors during execution or the ability of an algorithm to continue to operate despite abnormalities in input, calculations, etc.
6. **Reliability**: The system should be reliable and must not degrade the performance of the existing system and should not lead to the hanging of the system.

4.3 Software Requirements

- Programming Language: MatLab
- Operating System : Windows XP or above

4.4 Hardware Requirements

- Processor: x86 or x64 compatible processor.
- RAM: 1GB or higher
- Hard Drive: 40GB or higher
- Imaging Hardware: Digital Camera or Higher resolution web camera
- Graphics card supporting OpenGL 3.3 with 1GB GPU memory recommended

Chapter 5

PROPOSED SYSTEM ARCHITECTURE

The proposed system is focused to solve the two main problems prevalent in educational institutions, namely keeping a track of number and type of vehicles currently in the premise while also aiding owners with exact time their vehicle had left premise in case of thefts. The System consists of two major components,

1) An image capturing source.

2) The application that is developed using MATLAB.

The block diagram for the proposed system is as shown in the Fig 5.1.

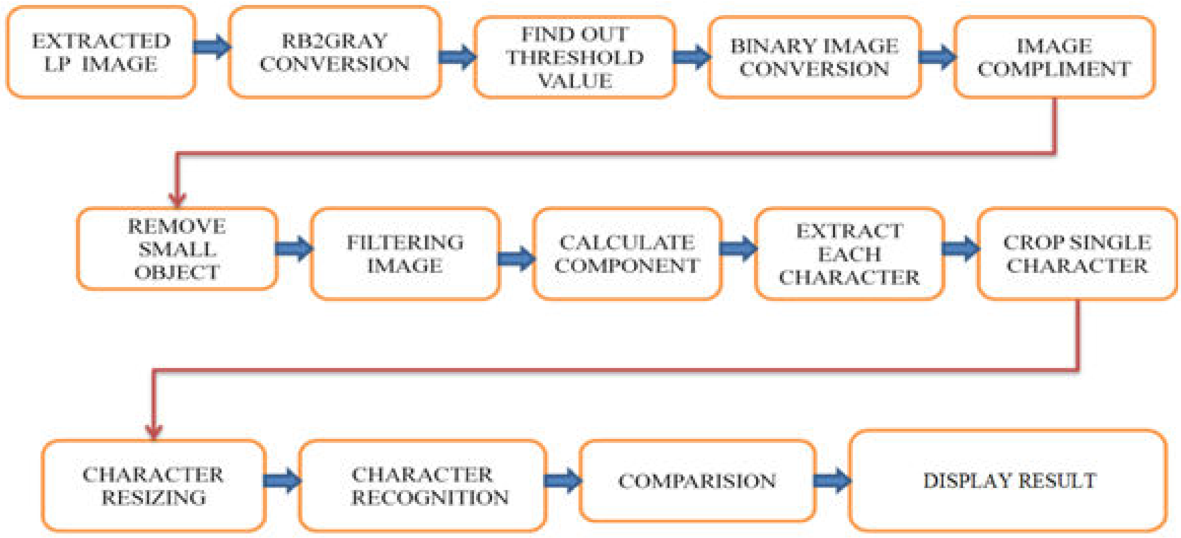

Fig 5.1. Block diagram of system for vehicle number plate recognition

5.1 Localization of number plate region and Character Segmentation

The selected image is pre-processed by passing it over gray scale filter and edge detection is applied to isolate the region of interest, which is the number plate itself as discussed in Chapter 3.1. A gray scale digital image is an image in which each pixel is quantized exclusively the shades of neutral gray, varying from black at the weakest intensity to white at the strongest intensity. The obtained gray image is then binarized, that is, it is converted to logical matrix by giving the pixel values of 1 for white shade and 0 for black shade. Possible number plate areas in images are identified by observing sudden changes in contrast. Remaining areas are filtered out. The best possible number plate location is found out by comparing width by height factor of actual Indian number plates to the same factor of plate like areas found by this method. The system shows maximum efficiency when the width by height factor is set between 3 and 7. The gray level plate images are enhanced by applying contrast extension and median filtering techniques. So, the contrast differences between images and the noises such as dirty regions in white background of the plate can be eliminated.

5.2 Contrast Extension

To extend the contrast of an image means equalization of the histogram of that image will be used. In other words, the contrast extension makes the image sharpen. The gray-level histogram of an image is the distribution of the gray level values in an image. The histogram equalization is a popular technique to improve the appearance of a poor contrasted image. The process of equalizing the histogram of an image consists of 4 steps: (i) find the sum of the histogram values. (ii) Normalize these values dividing by the total number of pixels. (iii) Multiply these normalized values by the maximum gray-level value. (iv) Map the new gray level values.

5.3 Median Filtering

Median filter is used for eliminating the unwanted noisy regions. As discussed in Chapter 3.2, in this filtering method, the 3x3 matrices is passed around the image. The dimension of these matrices can be adjusted according to the noise level. The process of working is (i) one pixel is chosen as centre pixel of the 3x3 matrices, (ii) the surrounding pixels are assigned as neighborhood pixels, (iii) the sorting process are employed between these nine pixels from smaller to the bigger, (iv) the fifth element is assigned as median element, (v) these procedures are implemented to the all pixels in plate image.

5.4 Character Segmentation

The characters of the identified number plate region are segmented using Region props function of MATLAB to obtain bounding boxes for each of the characters as discussed in Chapter 3.4. Region props function returns the smallest bounding box that contains a character. This method is used to obtain the bounding boxes of all characters in the number plate.

5.5 OCR using Template Matching

Template matching is one of the Character Recognition techniques. It is the process of finding the location of a subimage called a template, inside an image. Template matching involves determining similarities between a given template and windows of the same size in an image and identifying the window that produces the highest similarity measure. It works by pixel-by-pixel comparison of the image and the template for each possible displacement of the template. This process involves the use of a database of characters or templates. There exists a template for all possible input characters. Templates are created for each of the alphanumeric characters (from A-Z and 0-9) using 'Regular' font style. Fig 5.2 shows the templates for few of the alphanumeric characters.

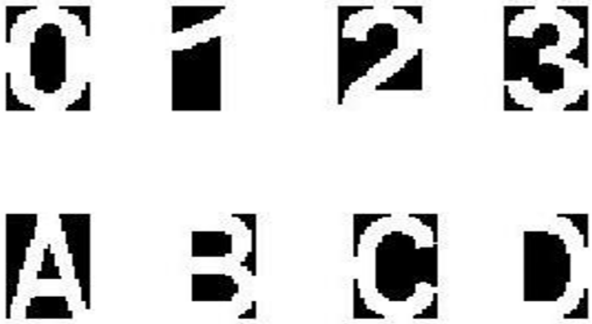

Fig 5.2. Template creation

For recognition to occur, the current input character is compared to each template to find either an exact match, or the template with the closest representation of the input character. It can capture the best position where the character is by moving standard template, thereby carry out the exact match. Moving template matching method is based on the template of target character, using the template of standard character to match the target character from eight directions of up, down, left, right, upper left, lower left, upper right, lower right.

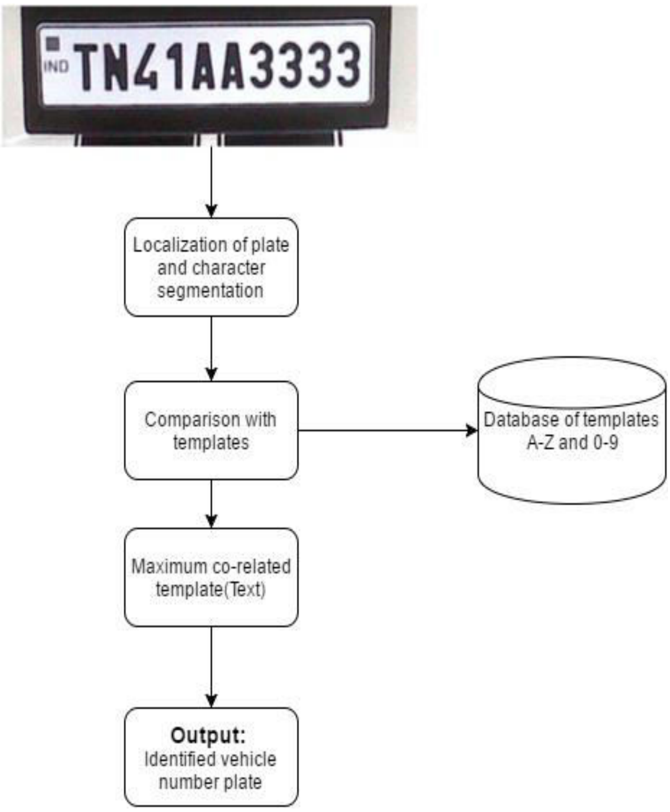

Fig 5.3. Block diagram of system for vehicle number plate recognition using Template Matching

Chapter 6

SYSTEM DESIGN

The process of Number Plate Recognition consists of four major stages:

6.1. Image Acquisition and Pre-processing

Image to be acquired using any available hardware. These captured images are in RGB format. Pre-processing is essential to enhance the input image which in turn reduces time complexity for localization and segmentation of characters. It mainly involves series of filtering and saturating the image to make the required region prominent. Finally, it involves converting the image into gray scale and increasing the contrast. Fig.6.1 shows the steps involved. Image pre-processing can significantly increase the reliability of an optical inspection. Several filter operations which intensify or reduce certain image details enable an easier or faster evaluation. Users are able to optimize a camera image with just a few clicks.

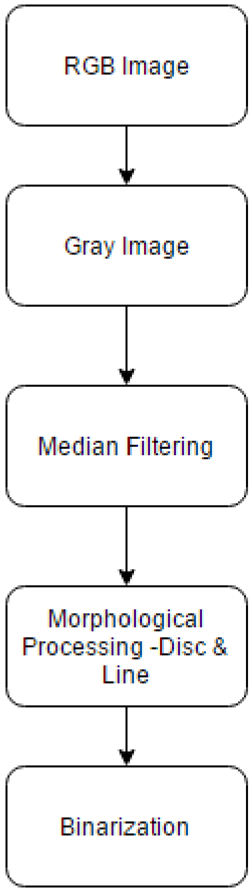

Fig 6.1 Block Diagram of Pre-processing stage

The intensity of a pixel is expressed within a given range between a minimum and a maximum, inclusive. This range is represented in an abstract way as a range from 0 (total absence, black) and 1 (total presence, white), with any fractional values in between. In image processing, normalization is a process that changes the range of pixel intensity values.

The main idea of the median filter is to run through the signal entry by entry, replacing each entry with the median of neighbouring entries. The pattern of neighbours is called the "window", which slides, entry by entry, over the entire signal. For 1D signals, the most obvious window is just the first few preceding and following entries, whereas for 2D (or higher-dimensional) signals such as images, more complex window patterns are possible (such as "box" or "cross" patterns). Note that if the window has an odd number of entries, then the median is simple to define: it is just the middle value after all the entries in the window are sorted numerically. For an even number of entries, there is more than one possible median.

Morphology is a broad set of image processing operations that process images based on shapes. Morphological operations apply a structuring element to an input image, creating an output image of the same size. In a morphological operation, the value of each pixel in the output image is based on a comparison of the corresponding pixel in the input image with its neighbours. By choosing the size and shape of the neighbourhood, you can construct a morphological operation that is sensitive to specific shapes in the input image.

The most basic morphological operations are dilation and erosion. Dilation adds pixels to the boundaries of objects in an image, while erosion removes pixels on object boundaries. The number of pixels added or removed from the objects in an image depends on the size and shape of the *structuring element* used to process the image. In the morphological dilation and erosion operations, the state of any given pixel in the output image is determined by applying a rule to the corresponding pixel and its neighbours in the input image. The rule used to process the pixels defines the operation as a dilation or an erosion.

Flood-fill operation on binary and grayscale images. For binary images, flood fill changes connected background pixels (0s) to foreground pixels (1s), stopping when it reaches object boundaries. For grayscale images, floodfill brings the intensity values of dark areas that are surrounded by lighter areas up to the same intensity level as surrounding pixels. In effect, floodfill removes regional minima that are not connected to the image border.

- Image binarization converts an image of up to 256 gray levels to a black and white image. Frequently, binarization is used as a pre-processor before OCR. In fact, most OCR packages on the market work only on bi-level (black & white) images.
- The simplest way to use image binarization is to choose a threshold value, and classify all pixels with values above this threshold as white, and all other pixels as black. The

problem then is how to select the correct threshold. In many cases, finding one threshold compatible to the entire image is very difficult, and in many cases even impossible.

6.2. Localization of Number Plate and Segmentation of Characters

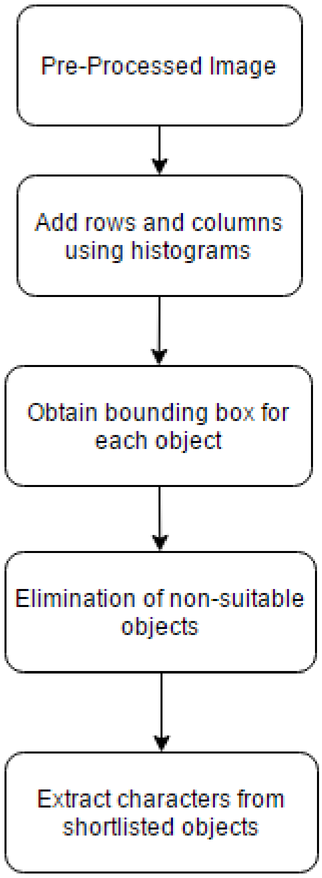

Fig 6.2. Steps involved for Localization and Segmentation

Localization and segmentation are processes in the number plate recognition. Fig 6.2 shows the sequence involved. Here we introduce segmentation based on sum of pixel values.

- **Add rows and column values**

Since the number plate contains black background and spaces in between the numbers, one can easily find out the variations in the graph of sum of rows and column values.

The extraction and segmentation procedures have slight differences. In extraction, large differences are identified and using that information number plate region was extracted, where as in segmentation process, a continuous monitoring of 0 to 1 and 1 to 0 transition is employed. These transition points are helpful in identifying the length and width of each character in the number plate. Hence each character is extracted using the graphical information.

Sometimes, it requires a multi-level segmentation process in order to get the accurate results. Under that condition, the above procedure is repeated such that each character is cropped exactly to the character limits.

- **Obtaining bounding box for each object**

Using the information obtained by analyzing the histogram, a bounding box for each of the object is obtained. Bounding box is the smallest box possible enclosing each of the individual objects.

The bounding boxes are stored as an array containing the coordinates of each of the objects in the image. The coordinates are specified as a 1-by-Q*2 vector, where Q is the number of image dimensions, for example, [ul_corner width]. ul_corner specifies the upper-left corner of the bounding box in the form [x y z ...]. width specifies the width of the bounding box along each dimension in the form [x_width y_width ...].

- **Elimination of non-suitable Objects**
  - Size Thresholding:

    In almost every NPR system, distance between car and camera is approximately fixed. Hence, certain range for character size can be fixed. For Indian number plates, this range needed to be quite wide. Many false character locations are removed in this step.

    In this step, each shortlisted character is scanned vertically and horizontally to calculate its size. If it is less than half of maximum possible character size that location is not considered as probable character location.
  - White-pixel Thresholding:

    Every true character location will have certain white pixel density. Hence, this was found to be reliable feature to discard false character locations. White pixel density at each probable character location is calculated. If it is above 40% of total area occupied by character, location is considered as probable character location.
- **Extract characters from shortlisted objects**

The remaining objects contain characters, these characters are extracted and passed on to the next step i.e Charater Recognition

.

6.3. Character Recognition

The system performs character recognition by quantification of the character into a mathematical vector entity using the geometrical properties of the character image. The scope of the proposed system is limited to the recognition of a single character.

Character recognition include a number of problems which make mandatory the development of an automatic process of classifying input information according to the specific requirements imposed on such a classification. The problem of character recognition results in the automatic making of the decision on the basis of data which does not directly indicate the best of all possible decisions. In general form the problem may be formulated as "There exists a set M of some objects which are divided into n nonintersecting subsets called object classes or characters. To each character there corresponds a specific character description x which, without restriction may be considered as multidimensional vector. The description of objects are not necessarily unique i.e. identical description may sometimes correspond to different object classes."

In the most general case, a reader which recognize characters must solve a problem equivalent to the calculation of the several logic function(depending on the character being distinguished), such that each function is equal to unity when and only when there is a character corresponding to this function in the field of vision of the character reader. These functions must be invariant with respect to all shifts and changes in the outlines of the characters which are considered permissible for this character reader. It is possible that in the calculation of functions corresponding to various characters several actions or sequences of action will be repeated. These sequences of action yield values of elementary functions that are common to all the characters. In order to recognize a character it is necessary to determine which value the elementary function has for this character. It is natural to call these elementary function features.

The features must be invariant with respect to the permissible changes in the characters. The direction of the stroke, which makes up the character, can be taken as indicators, which are invariant with respect to the forward movement and changes in the dimensions and in some cases with respect to the change in the proportion in the shape of the character. If the direction of the strokes is determined approximately, similar directions being taken as identical, then it is possible in addition to obtain variance with respect to slight rotations of a character or part of it. The directions of the stroke alone do not provide exhaustive information about the character e.g. "Γ" "L" may be characterized by identical stroke directions. Additional information can be obtained about the character if the direction of is determined , not of the stroke themselves but also of the their boundaries i.e. the boundaries between the black and white fields, and if in this process account is taken as to which side of the boundary the black field is on. It is most convenient to analyze characters by means of such indicators moving along the boundary of a stroke and recording the direction of movement in the sequence in which it occurs in the character.

Character recognition step will be identifying the characteristics of the character input image. In this stage, the segmented characters are rescaled to match the characters into a standard size. Different methods can used for character recognition. One of the method for character recognition is the optical character recognition (OCR) is used to compare the each individual character against the complete alphanumeric database. The OCR actually uses correlation method to match individual character and finally the number/alphabet is identified and stored .The character is then compared with the database for the vehicle authorization. The resultant signals are given according to the result of comparison.

The template-matching algorithm implements the following steps:

1. Firstly, the character image from the detected string is selected.
2. After that, the image to the size of the first template is rescaled.
3. After rescale the image to the size of the first template (original) image, the matching metric is computed.
4. Then the highest match found is stored. If the image is not match repeat again the third step.
5. The index of the best match is stored as the recognized character.

The value of the data that was entered will be extracted from the images, comprising letters. Each character was automatically selected and threshold using methods previously described. This process involves the use of a database of characters or templates. There exists a template for all possible input characters. For recognition to occur, the current input character is compared to each template to find either an exact match, or the template with the closest representation of the input character. Fig. 6.3 shows the templates used.

Fig. 6.3. Template used for OCR

The steps involved in Character Recognition is as shown in Fig.6.4

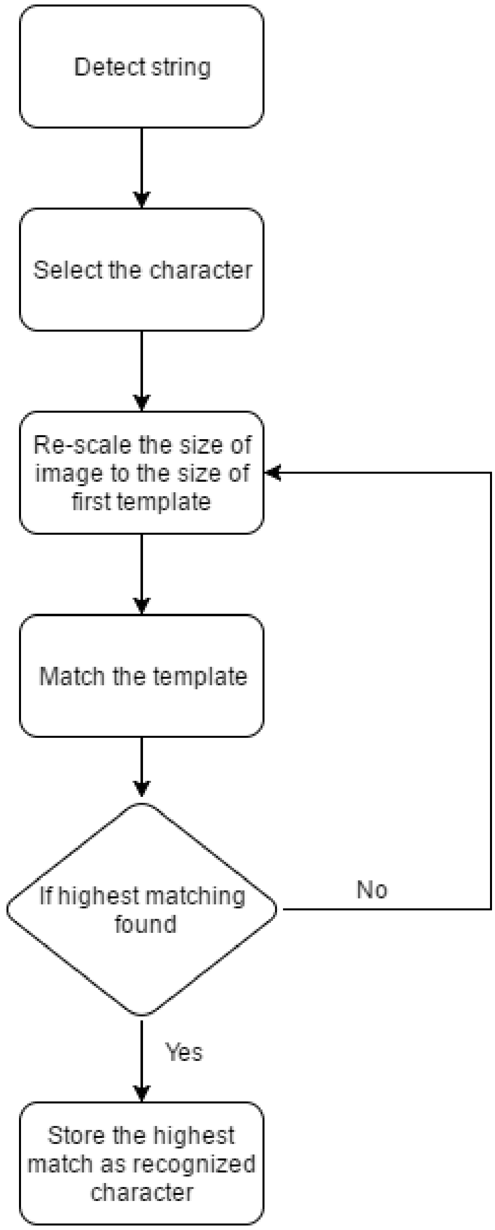

Fig. 6.4. Flow chart of OCR

6.3.1 Feature Extraction

Feature extraction is the process of getting information about an object or a group of object in order to facilitate classification. This is an important part in our system.

The input document may contain several lines of text that needs to be categorized into single character for recognition. For this purpose the following steps are to be applied:

1. The document is to be scanned for the initial darker pixel to be named as top of the row.

2. Now for bottom the next blank line is detected. The area between this is row of characters in image.

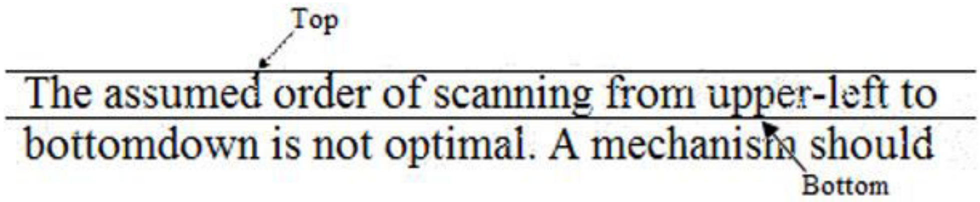

Fig. 6.5. Row Detection

3. Now each character is to be identified for the row obtained earlier. This is done by scanning the row vertically from top to bottom, the first darker pixel detected is the leftmost (left) pixel of character.
4. Now if all pixel are found to be blank then this is right of character.

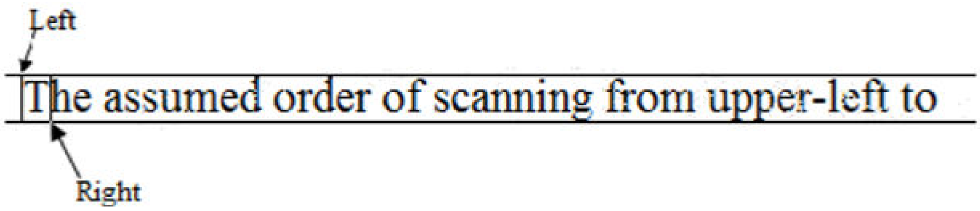

Fig. 6.6. Boundary

5. The character from the scanned image is normalised from any pixel size to 15 X 15 pixel. It cropped the image by using top, left, right, and bottom boundaries as in fig. 6.7

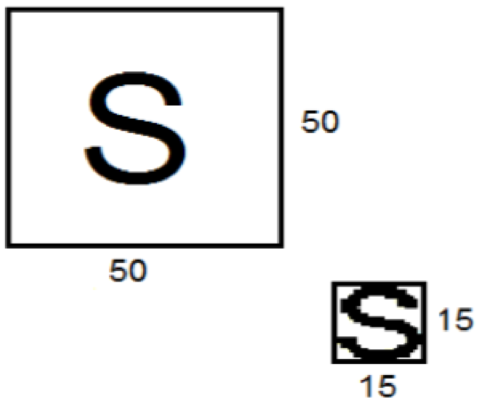

Fig. 6.7. Scaling

6. Now the cropped image of 15 X 15 can be binarized into array of 15 X 15, where black representing 1 and white representing 0 as shown in fig. 6.8.

Fig. 6.8. Binarization of character

Recognition of Pattern: Pattern based recognition require matching of generated binary format with the existing template for this purpose the binary has been divided into 5 tracks and each track subdivided into 8 sectors. A corresponding track-sector matrix is to be generated, identifying number of pixels in each region. This procedure is shown in fig. 6.9.

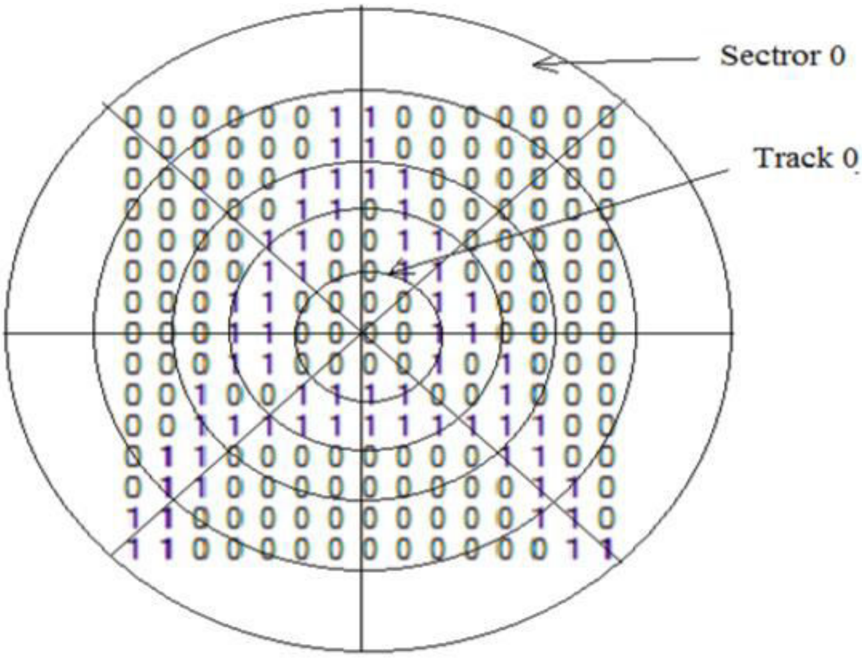

Fig. 6.9. Division into tracks and sectors

This can be done using following procedure 1. Identify center of matrix 2. Calculate radius say rad by finding pixel with maximum distance from center using distance formulae. Dist= ((2−1)+(2−1)) 3. Perform (rad ÷ 5) to identify size of each imaginary track. 4. Identify imaginary sectors. 5. Generate track- sector matrix by calculating number of 1's in each intersection of sector and track.

Recognition of Output: The track-sector matrix generated above is then matched with existing template. The existing template consists of each track-sector intersection value, each track value and each sector value. If all these parameters are found to match with the template values then the resultant is the character identified. The resultant matrix contains unique value for each font and thus makes it easy to identify each font separately.

Chapter 7

RESULTS AND DISCUSSION

NPR solution has been tested on static snapshots of vehicles of varying sizes, which has been divided into several sets according to difficulty. Sets of blurry and skewed snapshots give worse recognition rates than a set of snapshots which has been captured clearly. The objective of the tests was not to find a one hundred percent recognizable set of snapshots, but to test the invariance of the algorithms on random snapshots systematically classified to the sets according to their properties. Fig. 7.1 shows the number plate is detected and identified

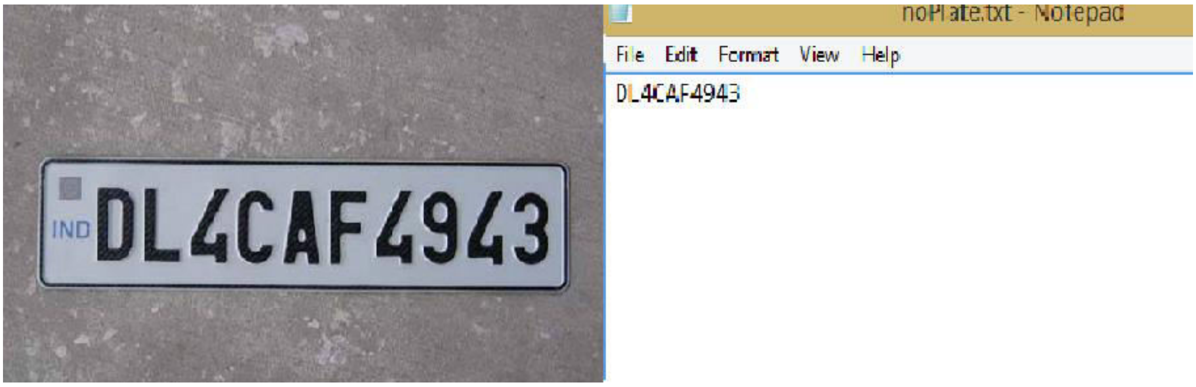

Fig. 7.1. Input and Corresponding Output

The number plate area detection algorithm is simulated with several pictures and the result is recorded in Table 1.

TABLE 1 Data collected for number plate area segmentation

Total samples	15
Identified samples	10
Incorrectly identified samples	5
Accuracy percentage	66.67%

7.1 SNAPSHOTS

7.1.2 License Plate No. 1

Fig. 7.2. Input Image (1)

Fig. 7.3. Pre-Processed Image (1)

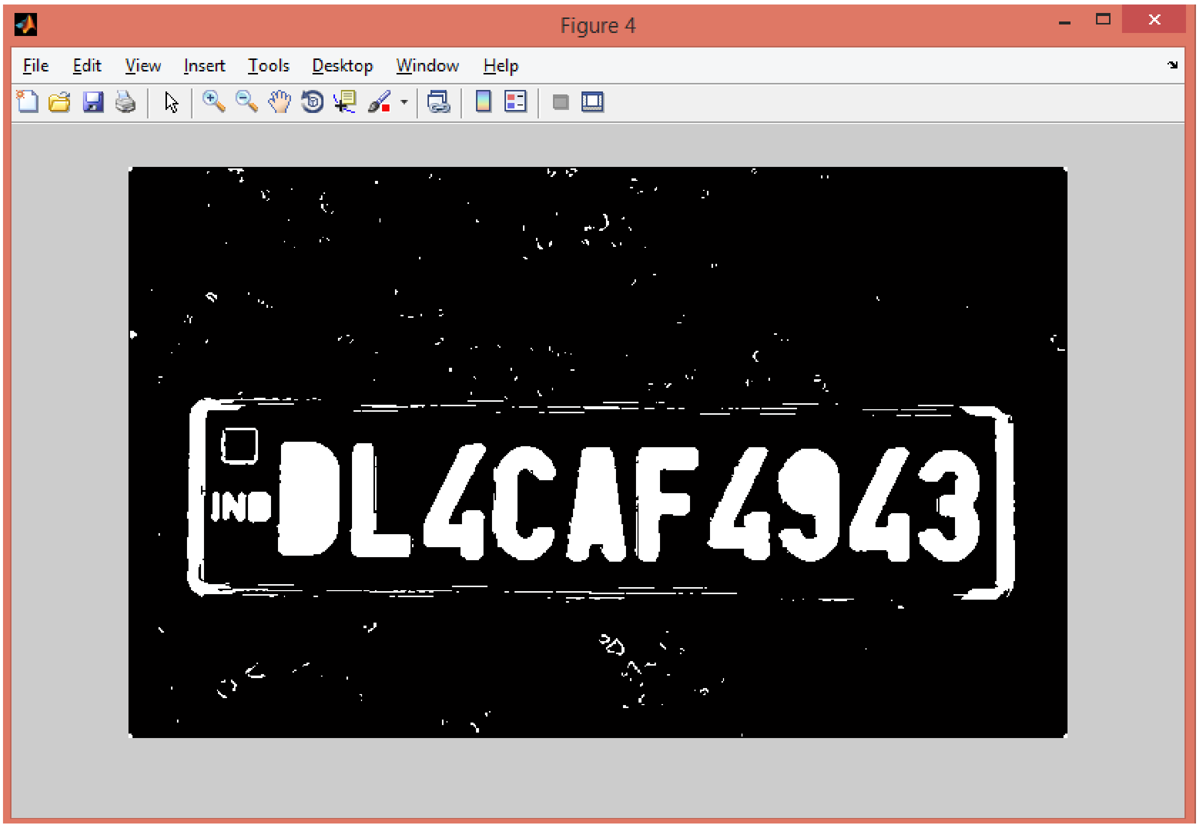

Fig. 7.4. Localised Image (1)

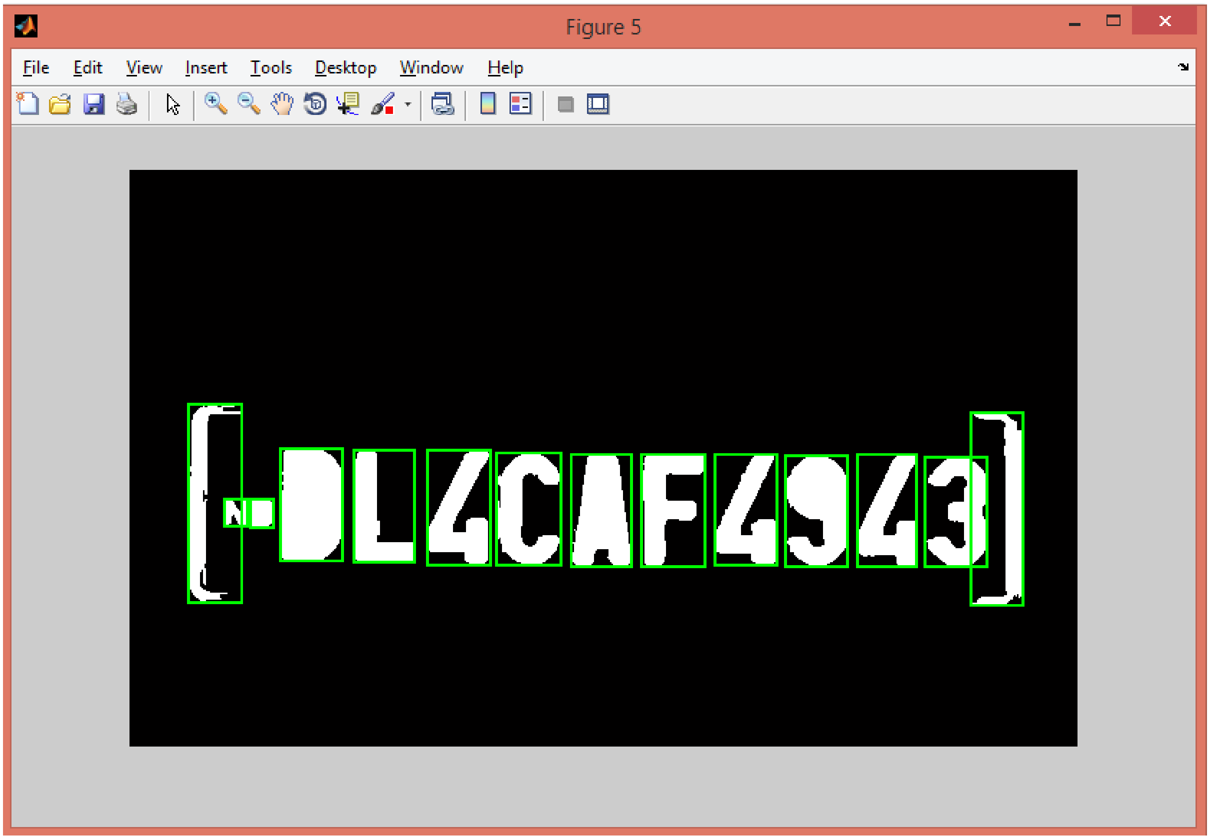

Fig. 7.5. Segmented Image (1)

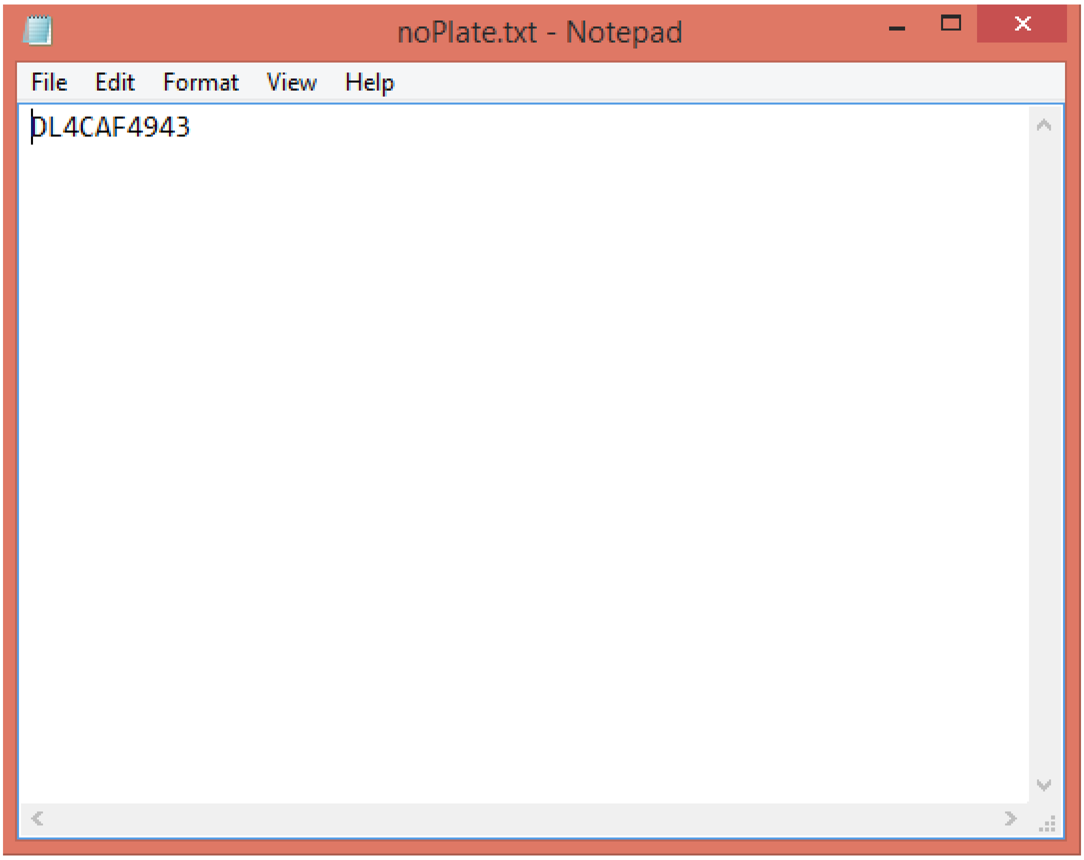

Fig. 7.6. Recognized Output (1)

7.1.2 License Plate No. 1

Fig. 7.7. Input Image (2)

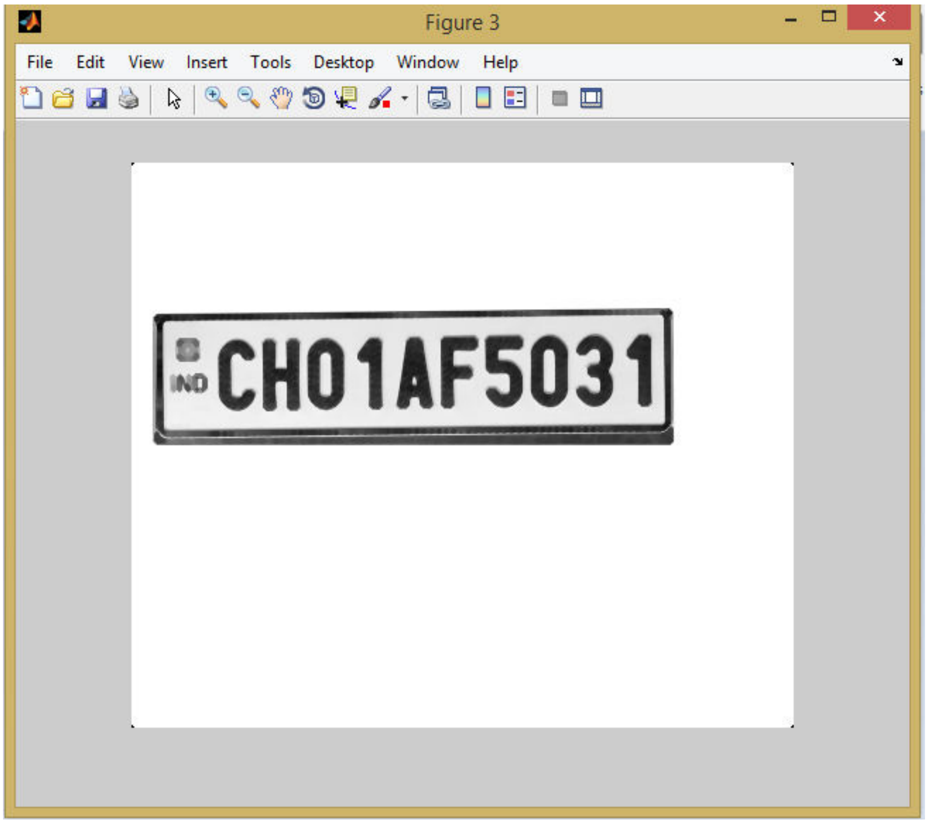

Fig. 7.8. Pre-Processed Image (2)

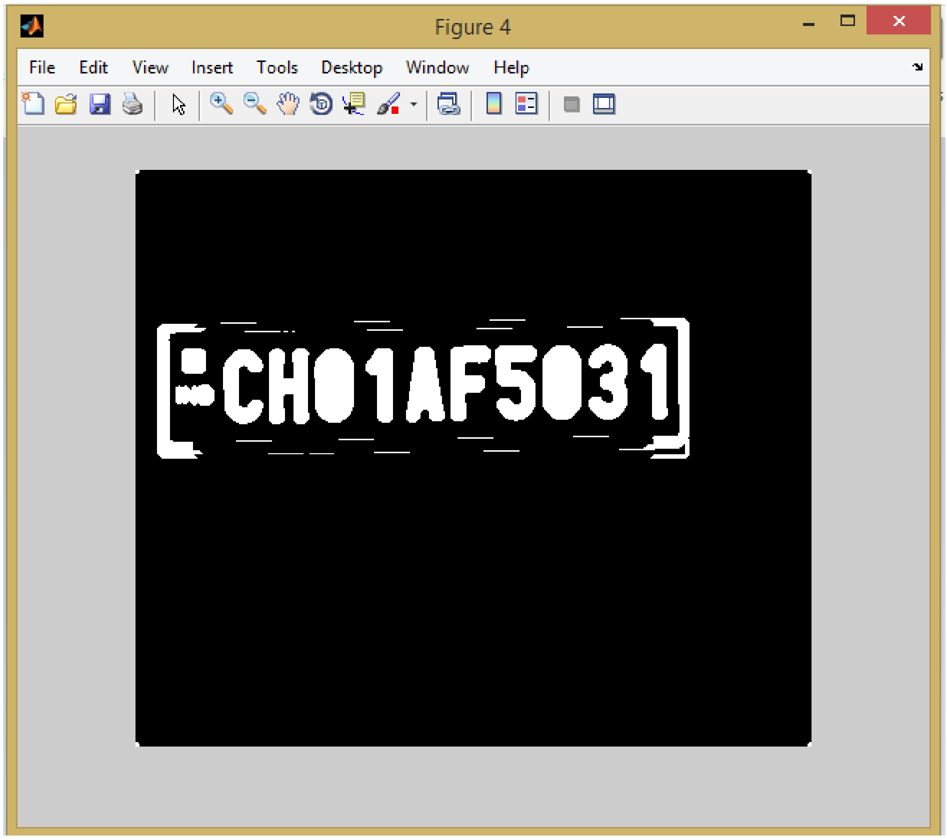

Fig. 7.9. Localized Image (2)

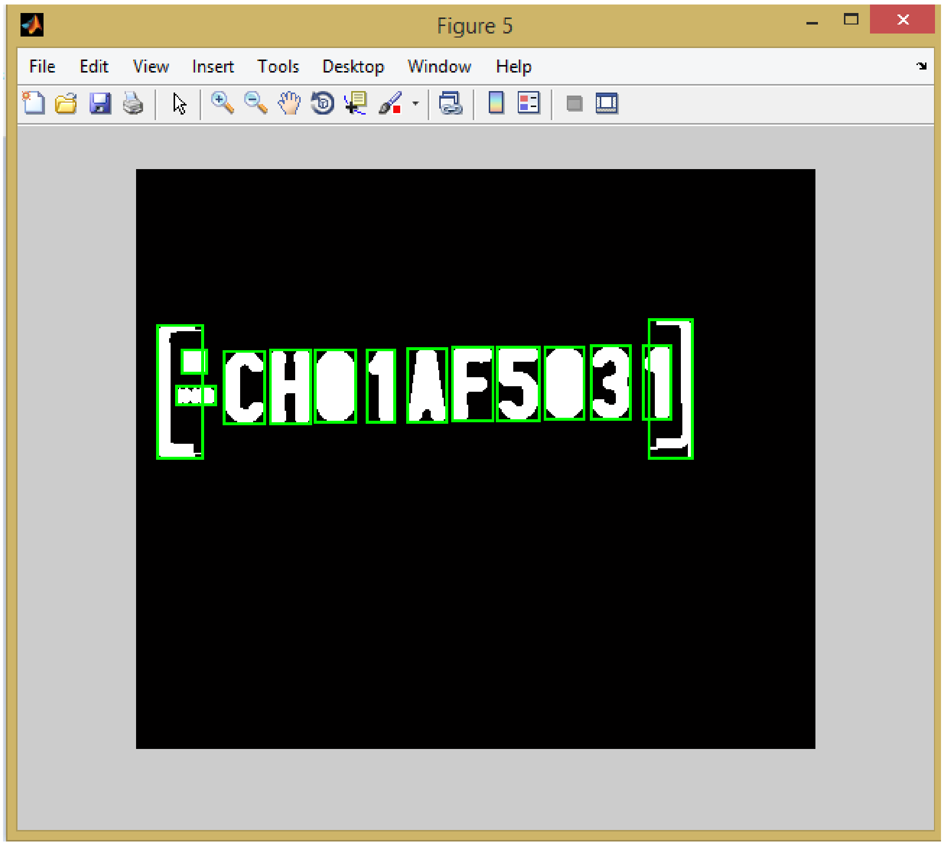

Fig. 7.10. Segmented Image (2)

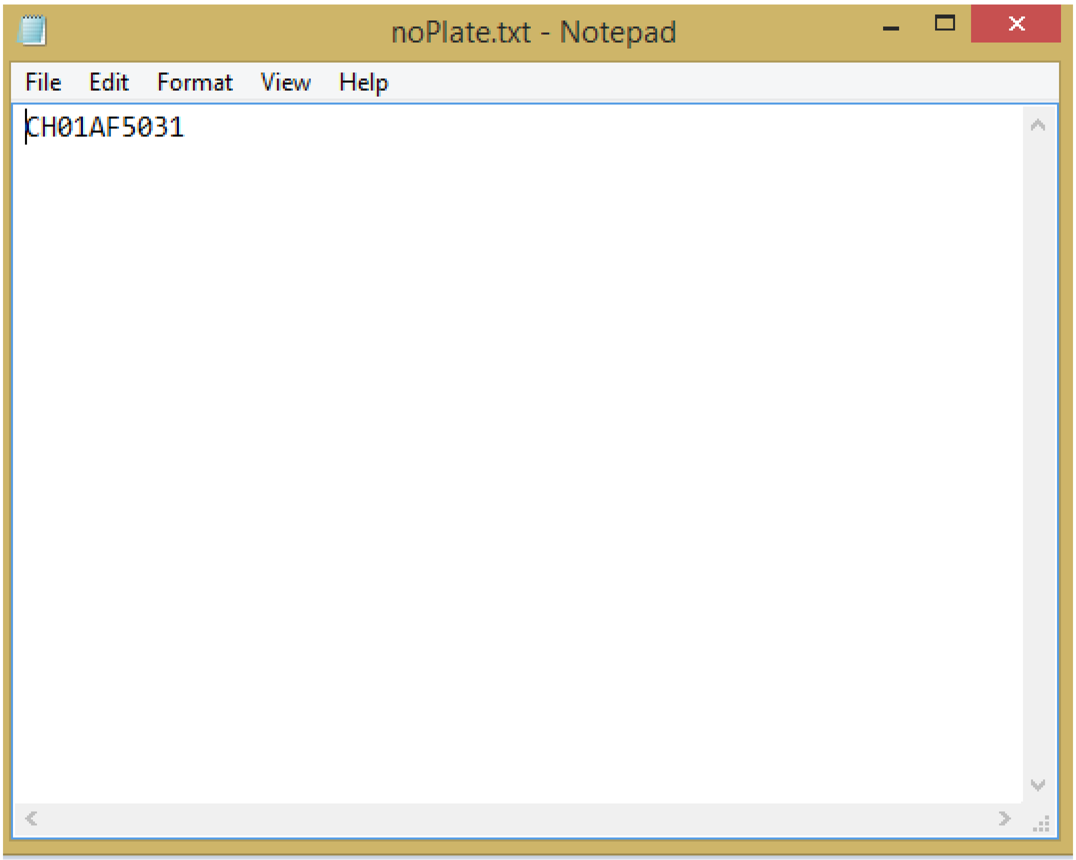

Fig. 7.11. Recognized Output

Chapter 8

CONCLUSION

In this work, existing methodologies and algorithms proposed in literature for Vehicle and Number Plate recognition were reviewed. Template matching was implemented on number plates obtained from static images and an average accuracy of 80.8% was obtained. This accuracy can be improved greatly by positioning the camera suitably to capture the best frame and using larger data set.

NPR can be further oppressed for vehicle owner identification, vehicle model identification traffic control, vehicle speed control and vehicle location tracking. It can provide various benefits like traffic safety enforcement, security- in case of suspicious activity by vehicle, easy to use, immediate information availability- as compare to searching vehicle owner registration details manually and cost effective. NPR solution has been tested on static snapshots of vehicles, which has been divided into several sets according to difficulty. The objective of the tests was not to find a one hundred percent recognizable set of snapshots, but to test the invariance of the algorithms on random snapshots systematically classified to the sets according to their properties.

At the current stage of development, the software does perform well either in terms of speed or accuracy but not better. It is unlikely to replace existing OCR methods, especially for alphanumeric text. Artificial neural networks are commonly used to perform character Recognition due to their high noise tolerance. The systems have the ability to yield excellent results. The implementation of the proposed system can be extended for the recognition of number plates of multiple vehicles in a single image frame by using multi-level genetic algorithms. Also, a more sophisticated version of this system can be implemented by taking inputs from live video feed and selecting the best vehicle frame for classification of vehicle types and recognizing the number plates using neural networks.

REFERENCES

[1] Patel, Ronak P., Narendra M. Patel, and Keyur Brahmbhatt. "Automatic licenses plate recognition." *International Journal of Computer Science and Mobile Computing (IJCSMC)* 2.4, 2013: 285-294.

[2] Du, Shan, et al. "Automatic license plate recognition (ALPR): A state-of-the-art review." *Circuits and Systems for Video Technology, IEEE Transactions on* 23.2, 2013: 311-325.

[3] Friedrich, Markus, Prokop Jehlicka, and Johannes Schlaich. "Automatic number plate recognition for the observance of travel behavior." *Processding of 8th International Conference on Survey Methods in Transport*. 2008.

[4] Zheng, Lihong, Xiangjian He, and Yuanzong Li. "A comparison of methods for character recognition of car number plates." *Verified OK,* 2005.

[5] Patel, Chirag, Dipti Shah, and Atul Patel. "Automatic number plate recognition system (anpr): A survey." *International Journal of Computer Applications* 69.9, 2013.

[6] Sarfraz, Muhammad, Mohammed Jameel Ahmed, and Syed A. Ghazi. "Saudi Arabian license plate recognition system." *Geometric Modeling and Graphics, 2003. Proceedings. 2003 International Conference on*. IEEE, 2003.

[7] Ozbay, Serkan, and Ergun Ercelebi. "Automatic vehicle identification by plate recognition." *World Academy of Science, Engineering and Technology*9.41, 2005: 222-225.

[8] Sulehria, Humayun Karim, Danish Irfan Ye Zhang, and Atif Karim Sulehria. "Vehicle number plate recognition using mathematical morphology and neural networks." *WSEAS Transactions on Computers* 7, 2008.

[9] Chen, Zhen-Xue, et al. "Automatic license-plate location and recognition based on feature salience." *Vehicular Technology, IEEE Transactions on*58.7, 2009: 3781-3785.

[10] Suri, Dr PK, Dr Ekta Walia, and Er Amit Verma. "Vehicle number plate detection using sobel edge detection technique." *International Journal of Computer Science and Technology, ISSN*, 2010: 2229-4333.

[11] Parasuraman, Kumar, and P. Vasantha Kumar. "An efficient method for indian vehicle license plate extraction and character segmentation." *IEEE International conference on computational intelligence and computing research*. 2010.

[12] Dashtban, Muhammad H., Zahra Dashtban, and Hassan Bevrani. "A novel approach for vehicle license plate localization and recognition." *International Journal of Computer Applications (0975–8887)* 26.11, 2011.

[13] Asthana, Stuti, Niresh Sharma, and Rajdeep Singh. "Vehicle number plate recognition using multiple layer back propagation neural networks."*International Journal of Computer Technology and Electronics Engineering (IJCTEE)* 1.1, 2011.

[14] Sharma, Chetan, and Amandeep Kaur. "Indian vehicle license plate extraction and segmentation." *International Journal of Computer Science and Communication* 2.2, 2011: 593-599.

[15] Lee, R. T., K. C. Hung, and H. S. Wang. "Real Time Vehicle License Plate Recognition Based on 2D Haar Discrete Wavelet Transform." *International Journal of Scientific & Engineering Research* 3.4, 2012: 1-6.

[16] Lekhana, G. C., and R. Srikantaswamy. "Real time license plate recognition system." *International Journal of Advanced Technology & Engineering Research* 2.4, 2012: 5-9.

[17] Zhou, Wengang, et al. "Principal visual word discovery for automatic license plate detection." *Image Processing, IEEE Transactions on* 21.9, 2012: 4269-4279.

[18] Rani, P. Sandhya, and Vara Prasad. "License plate character segmentation based on pixel distribution density." *IJESAT] International journal of engineering science and advanced technology* 2, 2012: 1539-1542.

[19] Khalifa, Othman, et al. "Malaysian Vehicle License Plate Recognition." *Int. Arab J. Inf. Technol.* 4.4, 2007: 359-364.

[20] Patel, Ronak P., Narendra M. Patel, and Keyur Brahmbhatt. "Automatic licenses plate recognition." *International Journal of Computer Science and Mobile Computing (IJCSMC)* 2.4, 2013: 285-294.

[21] Liao, Yu. "Research on Edge Detection in License Plate Recognition."*Proceedings of the 2012 International Conference on Computer Application and System Modeling*. Atlantis Press, 2012.

[22] Saha, Satadal, et al. "License Plate localization from vehicle images: An edge based multi-stage approach." *International Journal of Recent Trends in Engineering* 1.1, 2009: 284-289.

[23] Kranthi, S., K. Pranathi, and A. Srisaila. "Automatic number plate recognition." *Int. J. Adv. Tech* 2.3, 2011: 408-422.