A Project report on

NUMBER PLATE RECOGNITION FOR USE IN DIFFERENT COUNTRIES USING IMPROVED SEGMENTATION

Submitted in partial fulfilment of the Academic requirements for the award of the degree of

Bachelor of Technology

In

ELECTRONICS & COMMUNICATION ENGINEERING

by

S. SANJAYGANESH	**17H51A04B4**
T. SRAVANI	**17H51A04C0**
CH. REVANTH	**17H51A04D0**

Under the guidance of

Mrs. Y. Aruna Suhasini Devi
Associate Professor, ECE Department

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

CMR COLLEGE OF ENGINEERING & TECHNOLOGY

(AUTONOMOUS)

(NAAC Accredited with 'A+' Grade & NBA Accredited)

(Approved by AICTE, Permanently Affiliated to JNTU Hyderabad)

KANDLAKOYA, MEDCHAL ROAD, HYDERABAD-501401

2020-2021

CMR COLLEGE OF ENGINEERING & TECHNOLOGY

(AUTONOMOUS)

(NAAC Accredited with 'A+' Grade & NBA Accredited)

(Approved by AICTE, Permanently Affiliated to JNTU Hyderabad)

KANDLAKOYA, MEDCHAL ROAD, HYDERABAD-501401

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

CERTIFICATE

This is to certify that the dissertation entitled **"NUMBER PLATE RECOGNITION FOR USE IN DIFFERENT COUNTRIES USING IMPROVED SEGMENTATION"** is a bonafide work done by **S. SANJAYGANESH(17H51A04B4), T. SRAVANI(17H51A04C0), CH. REVANTH (17H51A04D0)** in partial fulfilment of requirements for the award of degree of Bachelor of Technology in Electronics & Communication Engineering, submitted to the Department of Electronics & Communication Engineering, CMR College of Engineering & Technology, Hyderabad during the Academic Year 2020-2021

Mrs. Y. ARUNA SUHASINI DEVI
Associate Professor, ECE Department
(Project Guide)

Prof. E.N.V PURNA CHANDRA RAO
HOD, ECE Department

ACKNOWLEDGEMENT

We are highly indebted and grateful to our guide Mrs. Y. ARUNA SUHASINI DEVI, Associate Professor, Department of ECE, CMRCET for her excellent guidance and constant encouragement throughout for the successful completion of the Project.

We are obliged and grateful to thank, Prof. E.N.V. PURNA CHANDRA RAO HOD**,** Department of ECE, CMRCET**,** for his cooperation in all respects.

We are obliged and grateful to thank, Dr. B. LOKESHWAR RAO, Dean (Academics), CMRCET, for his cooperation in all aspects.

We would like to thank Major Dr. V. A. NARAYANA, Principal, CMRCET, for his support in the course of this project work.

We would like to thank Sri Ch. GOPAL REDDY garu, Secretary& Correspondent of CMRCET, for his cooperation in all respects during the course.

It gives immense pleasure in expressing our deep sense of gratitude to project In-charge, Mr. P. RAVEENDRABABU, Associate Professor, Department of ECE, & project Coordinator Mr. V. PANDURANGA, Associate Professor, Department of ECE CMRCET for their valuable suggestions in each and every review during the course of my project.

Finally, we would like to thank all teaching & non- teaching staff members of the department, for their cooperation and support throughout the duration of our course.

Ultimately we own all our success to our beloved parents, whose vision, love and inspiration has made us to reach out for these glories.

SIGNATURE

SEELAM SANJAYGANESH (17H51A04B4)

TUMMEPALLY SRAVANI (17H51A04C0)

CHALLA REVANTH (17H51A04D0)

DECLARATION

We hereby declare that results embodied in this report of project on "**NUMBER PLATE RECOGNITION FOR USE IN DIFERENT COUNTRIES USING IMPROVED SEGMENTATION**" are from work carried out by using partial fulfilment of the requirements for the award of B. Tech degree. We have not submitted this report to any other university for the award of the other degree.

SIGNATURE

SEELAM SANJAYGANESH (17H51A04B4)

TUMMEPALLY SRAVANI (17H51A04C0)

CHALLA REVANTH (17H51A04D0)

DATE:

ABSTRACT

Number Plate Recognition (NPR) is a real time embedded system which identifies the characters directly from the image of the license plate. It is an active area of research. NPR systems are very useful to the law enforcement agencies as the need for Radio Frequency Identification tags and similar equipment's are minimized. Since number plate guidelines are not strictly practiced everywhere, it often becomes difficult to correctly identify the non-standard number plate characters. In this project we try to address this problem of NPR by using a pixel based segmentation algorithm of the alphanumeric characters in the license plate. The non-adherence of the system to any particular country-specific standard & fonts effectively means that this system can be used in many different countries – a feature which can be especially useful for trans-border traffic e.g. use in country borders etc. Additionally, there is an option available to the end-user for retraining the Artificial Neural Network (ANN) by building a new sample font database. This can improve the system performance and make the system more efficient by taking relevant samples.

INDEX

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

Number plate acknowledgment is a type of programmed vehicle recognizable proof. A number plate is the one of a kind recognizable proof of vehicle. It is a picture preparing innovation used to distinguish vehicles by their own particular number plates Programmed Number Plate Recognition is a picture handling innovation used to distinguish vehicles by their number plates. The Vehicle Number Plate is a special recognizable proof number for each vehicle made. Dissimilar to enrolment number, number plate does not change all through the lifetime. Since the 21st century, with social advancement and change of expectations for everyday comforts, the quantity of vehicles is constantly expanded. The movement conditions are exacerbating, which conveyed immense weights to the general public and condition. Number plate acknowledgment framework can take care of the different street issues produced by the activity clog, in this manner getting increasingly consideration.

It is an image processing technology used to identify vehicles by their own number plates. Real time number plate recognition plays an important role in maintaining law enforcement and maintaining traffic rules. It has wide applications areas such as toll plaza, parking area, highly security areas, boarder's areas etc. Number plate recognition is designed to identify the number plate and then recognize the vehicle number plate from a vehicle. Number plate recognition has three major parts: vehicle number plate extraction, character segmentation and Optical Character Recognition (OCR). Number plate extraction is that stage where vehicle number plate is detected. The detected number plate is pre-processed to remove the noise and then the result is passed to the segmentation part to segment the individually characters from the extracted number plate. The segmented characters are normalized and passed to an OCR. At last the optical character information will be converted into encoded text. The characters are recognized using Template matching. The final output must be in the form of string of characters.

The goals achieved by this are lowering of costs, enhance competitive position and makes the stake holders more comfortable. Software systems designed to automate data conversion processes are often referred to as data capture systems. A typical data capture system incorporates a number of subordinate processes that range from document imaging and image enhancement to document interpretation, data extraction, data validation and, finally, publication of the data to application systems. Document imaging process also includes the storage, indexing, retrieval, processing, transmission and printing of the documents. Now a days, in general documents are created and stored electronically. Nearly every organization, however, has a large volume of documents that currently exist on paper media. Document imaging or imaging, allows organizations to capture paper-based information and convert it to electronic images that are stored in a computer electronically. This capture of paper based information can be done through scanning of the paper documents. Optical Character Recognition (OCR) technology enables us to convert different types of documents, such as scanned paper documents, PDF files or images captured by a digital camera into editable and searchable data. OCR systems perform the translation of scanned images of handwritten, typed or printed text into machine-encoded or computer understandable form. This enables us to save storage space, edit the text and search/index it. OCR systems play a very significant role in document image processing as it aids in converting electronic images into electronic text documents which can be edited, searched, indexed and stored effectively and efficiently. The generic OCR systems are capable of converting text from an image into machine readable text with limitations. Due to limitations, the OCR systems fails to read text from document images in majority of applications and requires the document images to be prepared suitably in order to enable the OCR systems to read the same. The objective of this research is to develop few pre-processing stages to document images for better readability by OCR systems

CHAPTER 2

LITERATURE SURVEY

2.1 LITERATURE SURVEY

Optical Character Recognition (OCR) is broadly utilized innovation which changes over filtered pictures of printed content, written by hand message characters into machine encoded content data, for example, ASCII. It can be perceived printed characters and written by hand characters yet the execution is straightforwardly reliant from the nature of information records. The OCR performed disconnected.

K. K. Kim et. al. is entirely intended for Korean plates. It has planned a framework executing for Support Vector Machines and report amazing normal character acknowledgment.

M.A. Ko et. al., T. Naito et. al.displays most optical character recognizers presented in a 2D-plane can keep up high achievement rate just inside a constrained scope of visual edge and shooting separation. X. Dish et. al. proposed a two phase cross breed acknowledgment framework joining measurable and auxiliary acknowledgment technique. This work incorporates Distinguishing comparable characters by nearby auxiliary elements and building up a framework engineering consolidating measurable and basic acknowledgment techniques. To begin with, the four subclassifiers autonomously perceive the character and after that acknowledgment results are consolidated utilizing the Bayes technique. Furthermore, if the perceived characters have a place with the arrangements of vague characters the auxiliary stage is utilized for further choice.

Y. Huang et. al. presents to identify the number plate in the procured picture caught from camcorder. The angle administrator is utilized to find the plausible number plate range, the otsu strategy used to binarize the picture and the layout coordinating for acknowledgment. The root-mean squared-mistake (RMSE) utilized for figuring closeness of a model and double picture.

Parul Shah et. al. presents a novel calculation for vehicle undercarriage number recognizable proof in view of OCR utilizing manufactured neural system. This technique gives extensively high incentive for right distinguishing proof rate alongside zero wrong recognizable proof rate. The caught picture is contrasted and all the database pictures of alphanumeric characters and after that the most encouraging character is picked. Along these lines the acknowledgment procedure is finished. This technique is very quicker than the element coordinating methodology however we need to trade off with the precision of acknowledgment.

Muhammad Tahir Qadri et. al. presents programmed vehicle distinguishing proof framework. The OCR techniques in this are touchy to misalignment and to various sizes. Along these lines, the fundamental contrast between Template coordinating and Machine Learning methodology is that Template coordinating is a Shape-Matching methodology however machine learning methodology is a Feature-Matching methodology. Along these lines, the time required to prepare any framework for highlight coordinating methodology is very long. In this venture we accepted a controlled situation, the format coordinating methodology is use to lessen the aggregate calculation of the LPR framework.

S. Hamidreza kasaei et. al. presents a continuous and hearty technique for auto number plate discovery and acknowledgment. Morphological administrator is utilized to find the number plate and format coordinating for character acknowledgment. This strategy sets aside much opportunity to assess yet the exactness of acknowledgment is high. The second technique, format coordinating strategy essentially measures the straight connection between the caught pictures and the database pictures . This technique absolutely depends on the nature of the caught picture.

The Number Plate Recognition was invented in 1976 at the Police Scientific Development Branch in the UK. The review process was adopted by surveying the research in last 5 years (2010-2015) for collection of information about number plate recognition issues. In the Existing system extensive research has been done in the area of number plate recognition since its invention in the year 1976.This is the topic of recent research attracting several papers around the world. By taking an overview on studies of Number Plate Recognition from the past few years it is still a challenging task to detect characters from number plate.

CHAPTER 3
SYSTEM REQUIREMENTS

3.1 SYSTEM REQUIREMENTS

HARDWARE REQUIREMENTS:

TABLE 1 Hardware Requirements

PC	WINDOWS 7 OR HIGHER
PROCESSOR	DUAL CORE, CORE2DUO, INTEL i3
RAM	2 GB
DISK SPACE	5-8 GB
GRAPHICS ADAPTER	8-bit graphics adapter

SOFTWARE REQUIREMENTS:

MATLAB V.13 (R2013a)

3.2 High Level Specifications:

- MATLABV.13(R2013a)
- Intel Dual core or core2duo, Intel i3, i5, XP based personal computer.
- 4 GB RAM recommended
- 8-bit graphics adapter and display (for 256 simultaneous colours). A 32-bit or 64-bit OpenGL capable graphics adapter is strongly recommended.

3.3 Low Level Specifications:

- Microsoft Windows supported graphics accelerator card, printer, and soundcard.
- Microsoft Word 8.0 (Office 97), Office 2000.
- TCP/IP is required on all platforms when using a license server.
- Some license types require a license server running FLEX lm 8.0d, which is provided by the Math Works installer.

3.4 Functional Requirements:

- System should detect the images.
- System should recognize the input image and should display the output if it is matches

3.5 Non Functional Requirements:

- **Performance**: input image will be recognized with an accuracy of about 90% and more.
- **Functionality:** This software will deliver on the functional requirements mentioned in this document.
- **Availability:** This system will recognize only those images by which it was trained.
- **Flexibility:** It provides the users to load the image easily.
- **Learn ability:** The software is very easy to use and reduces the learning work.

CHAPTER 4

METHODS

4.1 EXISTING METHOD

Learning-based approach for license plate recognition

This project presents Adaboost learning-based method for license plate detection in unconstrained environment (cluttered scenes, changing illumination, in-plane and out-plane rotation of license plates). Our approach is motivated by the idea that learning-based method can implicitly derive a robust object model through training using large set of positive and negative samples. In addition, edge rather than intensity information is used to train license plate detector (LPD) since edge information – using canny edge detector – has shown better representation than intensity for license plate problem. We present comparative results of our approach against intensity, selection of different number of stages as well as our LPD detection speed. Our approach achieves true positive rate of ~70%, with detection speed ~80 ms for image size of 320 x 240.

As the number of vehicles is growing, license plate detection is becoming more important. It can be applied to applications such as traffic control, security system, vehicle verification, car park payment system and etc. A license plate detector (LPD) locates the position of the license plate from a given image. It is challenging to detect a license plate from a cluttered background, and image with a lot of noises such as illumination, rotation, and etc. The rest of the paper is presented in the following structure. Section 2 presents background information related to license plate detection. The architecture of our algorithm is introduced in section 3, and section 4 is introduces Haar-like features with AdaBoost learning algorithm. Experimental results and discussion are presented in section 9. We conclude This project in section

Viola et al. presented a framework for face detection that achieves high detection rate and yet with extremely rapid image processing. Motivated by , they introduced a new image representation known as the integral image that allows the features used in the detection to be calculated very rapidly. They use the Haar-like features to classify the patterns for an image, and the Haar-like features used in their framework are reminiscent of Haar wavelet used by Papageorgiou and Poggio .Then, they used

AdaBoost learning algorithm to select a simple Haar-like features from the over-complete features set. Each feature is known as a weak classifier, and the weak classifiers will be combined to become a strong classifier. Chen and Yuille demonstrated an algorithm for detecting text in natural images, also based on AdaBoost. They claimed that the set of features used for face detection by Viola and Jones might not be suitable for detecting text. This is because there is less spatial similarity for the text compare to face; a face can be regarded as spatial similar object since it consists of facial features such as eyes, nose, and mouth that are approximately the same spatial position for any face. Some of the algorithms are designed specifically for a particular object detection problem such as the adaptive algorithm for text detection from natural scenes by Gao and Yang. They developed a prototype system that can recognize Chinese sign inputs. The algorithm is designed in hierarchical structure with different conditions regulated in each layer. The algorithm can be applied to other languages text by modifying the layout constraints. Nevertheless, it is difficult to design an algorithm that can be used to detect different object without changing the architecture of the system.

Robust License-Plate Recognition Method for Passing Vehicle Under Outside Environment

In recent years, many researches on Intelligent Transportation Systems (ITS) have been reported. License Plate Recognition (LPR) is one form of ITS technology that not only recognizes and counts vehicles, but distinguishes each as unique by recognizing and recording the license plate's characters. This project discusses the main techniques of LPR. Several open problems are proposed at the end of the paper for future research.

In recent years, many researches on Intelligent Transportation Systems (ITS) have been reported. As one form of ITS technology, License plate Recognition (LPR) not only recognizes and counts vehicles, but distinguishes each as unique by recognizing the characters in the license plates. In the approach, a camera captures the vehicle images and a computer processes the captured images and recognizes the information on the license plate by applying various image processing and optical pattern recognition techniques . The significant advantage of the LPR system is that it assumes that all vehicles already have the unique identification (the license plate). So, no additional transmitter or responder is required to be installed on the vehicle. Computer-based license plate recognition emerged in the 1980's. Currently, there are multiple commercial license plate recognition systems available . LPR plays an important role in numerous applications. Examples are : - Border crossing control - Identification of stolen vehicles - parking attendant - Petrol station forecourt surveillance - Red light camera - Speed enforcement - Security - Customer identification - Indoor parking lots/Underground parking One of the key points for most of the traffic related applications, such as road traffic monitoring or parking lots access control, is the possibility to ally detect and recognize vehicle license plate in uncontrolled open environments. Due to different working environments, LPR techniques vary from application to application . Although LPR is a developed technology, there are, however, still many problems needed to be solved.

License Plate Surveillance System Using Weighted Template Matching

In this research work main aim is to focus on to pattern matching and analysis of image for vehicle's number plate recognition using template matching and neural network as hybrid method. In this research work car images was tested by applying template matching algorithm neural network as hybrid method out of many images of vehicle's number plate recognition done correctly. Some of images are wrong result due to distance, image size, angle of view, condition of weather or illumination condition etc. pattern matching and analysis of number plate recognition system have many applications like payment of parking fees; toll fee on the highway; traffic monitoring system; border security system; signal system etc

4.2 PROPOSED METHOD

OPTICAL CHARACTER RECOGNITION

Number plate recognition is a technology that uses optical character recognition on images to read vehicle registration plates to create vehicle location data. It can use existing closed-circuit television, road-rule enforcement cameras, or cameras specifically designed for the task. NPR is used by police forces around the world for law enforcement purposes, including to check if a vehicle is registered or licensed. It is also used for electronic toll collection on pay-per-use roads and as a method of cataloguing the movements of traffic, for example by highways agencies.

Number plate recognition can be used to store the images captured by the cameras as well as the text from the license plate, with some configurable to store a photograph of the driver. Systems commonly use infrared lighting to allow the camera to take the picture at any time of day or night. NPR technology must take into account plate variations from place to place.

Concerns about these systems have centered on privacy fears of government tracking citizens' movements, misidentification, high error rates, and increased government spending. Critics have described it as a form of mass surveillance.

4.2.1 COMPONENTS

The software aspect of the system runs on standard home computer hardware and can be linked to other applications or databases. It first uses a series of image manipulation techniques to detect, normalize and enhance the image of the number plate, and then optical character recognition (OCR) to extract the alpha-numeric of the license plate. NPR systems are generally deployed in one of two basic approaches: one allows for the entire process to be performed at the lane location in real-time, and the other transmits all the images from many lanes to a remote computer location and performs the OCR process there at some later point in time. When done at the lane site, the information captured of the plate alphanumeric, date-time, lane identification, and any other information required is completed in approximately 250 milliseconds

This information can easily be transmitted to a remote computer for further processing if necessary, or stored at the lane for later retrieval. In the other arrangement, there are typically large numbers of PCs used in a server farm to handle high workloads, such as those found in the London congestion charge project. Often in such systems, there is a requirement to forward images to the remote server, and this can require larger bandwidth transmission media.

NPR uses optical character recognition (OCR) on images taken by cameras. When Dutch vehicle registration plates switched to a different style in 2002, one of the changes made was to the font, introducing small gaps in some letters (such as *P* and *R*) to make them more distinct and therefore more legible to such systems. Some license plate arrangements use variations in font sizes and positioning—NPR systems must be able to cope with such differences in order to be truly effective. More complicated systems can cope with international variants, though many programs are individually tailored to each country.

4.2.2 IN MOBILE SYSTEMS

During the 1990s, significant advances in technology took number plate recognition (NPR) systems from limited expensive, hard to set up, fixed based applications to simple "point and shoot" mobile ones. This was made possible by the creation of software that ran on cheaper PC based, non-specialist hardware that also no longer needed to be given the pre-defined angles, direction, size and speed in which the plates would be passing the camera's field of view. Further scaled-down components at more cost-effective price points led to a record number of deployments by law enforcement agencies around the world. Smaller cameras with the ability to read license plates at higher speeds, along with smaller, more durable processors that fit in the trunks of police vehicles, allowed law enforcement officers to patrol daily with the benefit of license plate reading in real time, when they can interdict immediately.

Despite their effectiveness, there are noteworthy challenges related with mobile NPRs. One of the biggest is that the processor and the cameras must work fast enough to accommodate relative speeds of more than 100 mph (160 km/h), a likely scenario in the case of oncoming traffic. This equipment must also be very efficient since the power source is the vehicle battery, and equipment must be small to minimize the space it requires.

Relative speed is only one issue that affects the camera's ability to actually read a license plate. Algorithms must be able to compensate for all the variables that can affect the NPR's ability to produce an accurate read, such as time of day, weather and angles between the cameras and the license plates. A system's illumination wavelengths can also have a direct impact on the resolution and accuracy of a read in these conditions.

Installing NPR cameras on law enforcement vehicles requires careful consideration of the juxtaposition of the cameras to the license plates they are to read. Using the right number of cameras and positioning them accurately for optimal results can prove challenging, given the various missions and environments at hand. Highway patrol requires forward-looking cameras that span multiple lanes and are able to read license plates at very high speeds. City patrol needs shorter range, lower focal length cameras for capturing plates on parked cars. Parking lots with perpendicularly parked cars often require a specialized camera with a very short focal length. Most technically advanced systems are flexible and can be configured with a number of cameras ranging from one to four which can easily be repositioned as needed. States with rear-only license plates have an additional challenge since a forward-looking camera is ineffective with oncoming traffic. In this case one camera may be turned backwards

4.2.3 STEPS

There are seven primary steps that the software requires for identifying a license plate:

1. Plate localization – responsible for finding and isolating the plate on the picture.
2. Plate orientation and sizing – compensates for the skew of the plate and adjusts the dimensions to the required size.

3. Normalization – adjusts the brightness and contrast of the image.
4. Character segmentation – finds the individual characters on the plates.
5. Optical character recognition.
6. Syntactical/Geometrical analysis – check characters and positions against country-specific rules.
7. The averaging of the recognised value over multiple fields/images to produce a more reliable or confident result. Especially since any single image may contain a reflected light flare, be partially obscured or other temporary effect.

The complexity of each of these subsections of the program determines the accuracy of the system. During the third phase (normalization), some systems use edge detection techniques to increase the picture difference between the letters and the plate backing. A median filter may also be used to reduce the visual noise on the image.

4.2.4 Difficulties

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

- Poor file resolution, usually because the plate is too far away but sometimes 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.
- Read license plates that are different at the front and the back because of towed campers, etc.
- Vehicle lane change in the camera's angle of view during license plate reading.
- A different font, popular for vanity plates (some countries do not allow such plates, eliminating the problem).
- Circumvention techniques.

- Lack of coordination between countries or states. Two cars from different countries states can have the same number but different design of the plate.

While some of these problems can be corrected within the software, it is primarily left to the hardware side of the system to work out solutions to these difficulties. Increasing the height of the camera may avoid problems with objects (such as other vehicles) obscuring the plate but introduces and increases other problems, such as the adjusting for the increased skew of the plate.

On some cars, tow bars may obscure one or two characters of the license plate. Bikes on bike racks can also obscure the number plate, though in some countries and jurisdictions, such as Victoria, Australia, "bike plates" are supposed to be fitted. Some small-scale systems allow for some errors in the license plate. When used for giving specific vehicles access to a barricaded area, the decision may be made to have an acceptable error rate of one character. This is because the likelihood of an unauthorized car having such a similar license plate is seen as quite small. However, this level of inaccuracy would not be acceptable in most applications of an NPR system

4.2.5 IMAGING HARDWARE

At the front end of any NPR system is the imaging hardware which captures the image of the license plates. The initial image capture forms a critically important part of the NPR system which, in accordance to the garbage in, garbage out principle of computing, will often determine the overall performance.

License plate capture is typically performed by specialized cameras designed specifically for the task, although new software techniques are being implemented that support any IP-based surveillance camera and increase the utility of NPR for perimeter security applications. Factors which pose difficulty for license plate imaging cameras include the speed of the vehicles being recorded, varying level of ambient light, headlight glare and harsh environmental conditions. Most dedicated license plate capture cameras will incorporate infrared illumination in order to solve the problems of lighting and plate reflective.

Many countries now use license plates that are retroreflective. This returns the light back to the source and thus improves the contrast of the image. In some countries, the characters on the plate are not reflective, giving a high level of contrast with the reflective background in any lighting conditions. A camera that makes use of active infrared imaging (with a normal colour filter over the lens and an infrared illuminator next to it) benefits greatly from this as the infrared waves are reflected back from the plate. This is only possible on dedicated NPR cameras, however, and so cameras used for other purposes must rely more heavily on the software capabilities. Further, when a full-colour image is required as well as use of the NPR-retrieved details, it is necessary to have one infrared-enabled camera and one normal (colour) camera working together.

To avoid blurring it is ideal to have the shutter speed of a dedicated camera set to 1/1000 of a second. It is also important that the camera uses a global shutter, as opposed to rolling shutter, to assure that the taken images are distortion-free. Because the car is moving, slower shutter speeds could result in an image which is too blurred to read using the OCR software, especially if the camera is much higher up than the vehicle. In slow-moving traffic, or when the camera is at a lower level and the vehicle is at an angle approaching the camera, the shutter speed does not need to be so fast. Shutter speeds of 1/500 of a second can cope with traffic moving up to 40 mph (64 km/h) and 1/250 of a second up to 5 mph (8 km/h). License plate capture cameras can produce usable images from vehicles traveling at 120 mph (190 km/h).

4.2.6 LICENSE PLATE RECOGNITION (LPR)

License Plate Recognition (LPR) is a technology that uses optical character recognition (OCR) to ally read license plate characters. There are two types of LPR: stationary, which uses infrared (IR) cameras at high fixed points, and mobile, which uses vehicle-mounted IR cameras.

Stationary cameras can be mounted on signs, street lights, highway overpasses or buildings as a cost-effective way to monitor moving and parked vehicles twenty-four hours a day. Camera software is able to identify the pixel patterns that make up a license plate and translate the letters and numbers on the plate to a digital format. The plate data is then sent to a database where it is compared in real-time to a list of plate numbers that belong to "vehicles of interest". If the system detects a match, it sends an alert to the dispatcher or other designated personnel.

Mobile LPR software suites use multiple cameras mounted on a vehicle. As the vehicle moves, it photographs license plates and transmits plate data to a database. The database may be a national database or it may be created at the local level and downloaded into the vehicle's onboard computer at the beginning of each shift. If the system detects a match, the officer receives an alert on his computer. A mobile LPR can read up to 1,000 plates per hour and cover two or more lanes of traffic at once. License Plate Recognition has many uses including:

- Recovering stolen cars.
- Identifying drivers with an open warrant for arrest.
- Catching speeders by comparing the average time it takes to get from stationary camera A to stationary camera B.
- Determining what cars do and do not belong in a parking garage.
- Expediting parking by eliminating the need for human confirmation of parking passes.

CHAPTER 5

SYSTEM DESIGN

5.1 BLOCK DIAGRAM

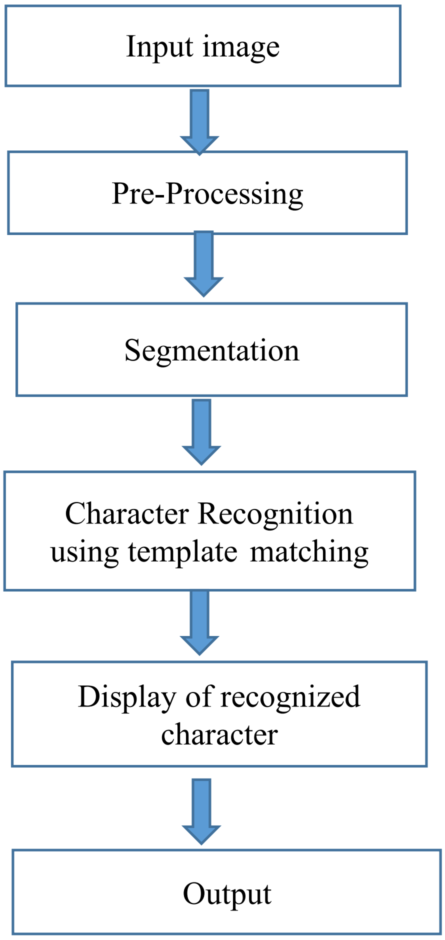

Fig 5.1. Block Diagram

The block diagram shows that, firstly an image is given to the system which is captured by camera. This image is before segmentation is given to pre-processing unit in which it will undergo filtering process, detection of license plate area and collecting the information about the boundary area of the license plate. After the pre-processing the image is given to the segmentation unit where the license plate is divided or segmented into images of pixels containing single character of from the license plate. These images are given to the OCR unit in which the number plate is detected.

5.2 Pre-Processing

- RGB to Gray Image Conversion
- Noise Removal
- Binary Image
- Median Filters
- Bounding Box of License plate

5.3 Flow Chart

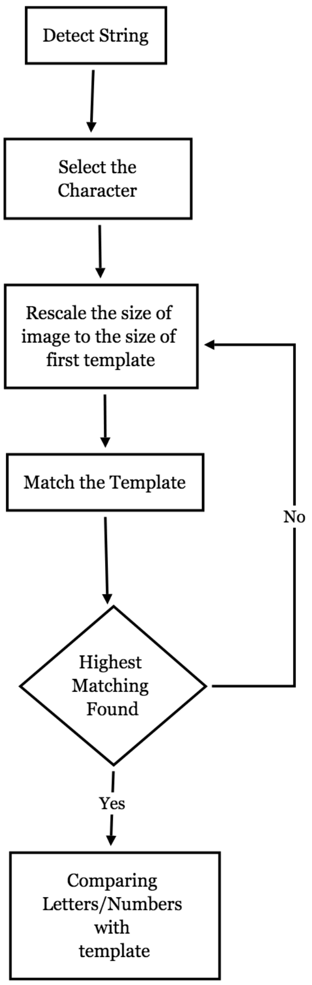

Fig 5.2 Template Flow Chart

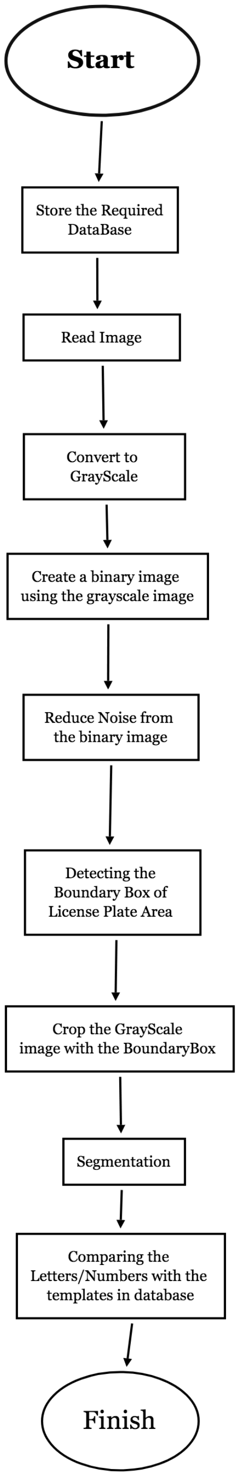

fig 5.3 Flow Chart

5.4 Design of the User Interface

GUIDE, the MATLAB graphical user interface development environment, provides a set of tools for creating graphical user interfaces (GUIs). These tools simplify the process of lying out and programming GUIs.

The GUIDE Layout Editor enables you to populate a GUI by clicking and dragging GUI components — such as buttons, text fields, sliders, axes, and so on — into the layout area. It also enables you to create menus and context menus for the GUI. The three main principle elements required in the creation of Graphical User Interface are

Components: Each item on the MATLAB GUI is a graphical component. The types of components include graphical controls (pushbuttons, edit boxes, lists, sliders, etc, static elements (frames and text strings), menus and axes. Axes, which are used to display the graphical data, are created using function axes. Graphical control and static elements are created by the function ui-control, and menus are created by function submenu and up context menu. Axes which are used to display graphical data are created by the function axes.

Figures: The components of the GUI must be arranged within the figure, which is a window on the computer screen. In the post figure have been created ally whenever we have plotted data. However empty figure can be created with the function figure and can be used to hold any combination of components.

Call back: Finally, there must be some way to perform an action if a user clicks a mouse on a button or type information on a keyboard. A mouse click or a key press is an event, and the MATALAB program must respond to each event if the program is to perform its function. The code executed in response to an event is called a call-back. There must be a call-back to implement the function of each graphical user component on the GUI.

5.5 Risks Identified

MATLAB is an interpreted language. The main disadvantage of interpreted languages is execution speed. When a language is compiled, all of the code is analysed and processed efficiently, before the programmer distributes the application. With an interpreted language, the computer running the program has to analyse and interpret the code (through the interpreter) before it can be executed (each and every time), resulting in slower processing performance.

The values of 39 and 35 hidden neurons for gradient features and character geometry respectively are chosen based on experiments conducted on several different images and are used by classifiers to produce correct classification results. The variations in the hidden neuron values might tend to produce wrong result. Hence these values should be carefully chosen.

CHAPTER 6
IMPLEMENTATION

6.1 Pre-Processing

6.1.1 Gray Image

Convert RGB image or colormap to grayscale.

I = rgb2gray(RGB) converts the TrueColor image RGB to the grayscale image I. The rgb2gray function converts RGB images to grayscale by eliminating the hue and saturation information while retaining the luminance. If you have Parallel Computing Toolbox installed, rgb2gray can perform this conversion on a GPU.

newmap = rgb2gray(map) returns a grayscale colormap equivalent to map.

Read and display an RGB image, and then convert it to grayscale.

Read the sample file, peppers.png, and display the RGB image.

```
RGB = imread('input.jpg');
imshow(RGB);
```

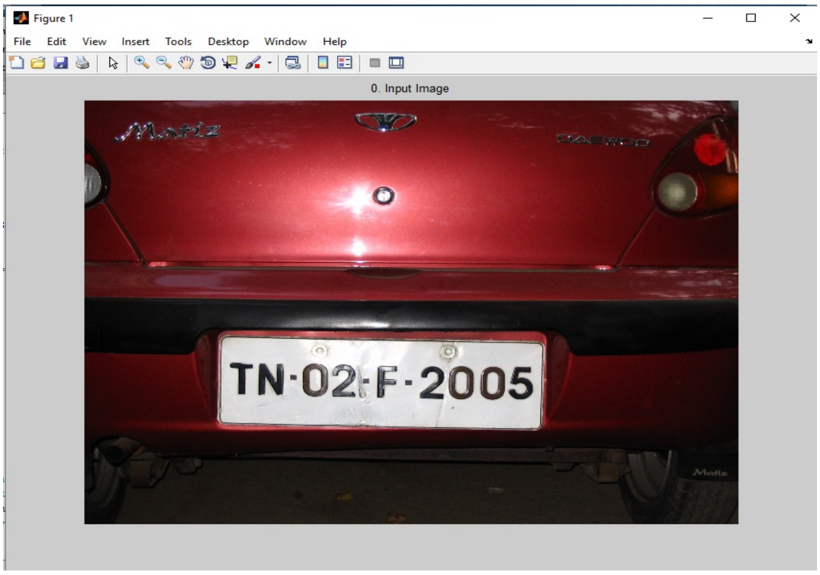

Fig 6.1 RGB Image

Convert the RGB image to a grayscale image and display it.

```
I = rgb2gray(RGB);
figure
imshow(I)
```

Fig 6.2 Gray Image

6.1.2 Noise Removal

Digital images are prone to various types of noise. Noise is the result of errors in the image acquisition process that result in pixel values that do not reflect the true intensities of the real scene. There are several ways that noise can be introduced into an image, depending on how the image is created. For example:

- If the image is scanned from a photograph made on film, the film grain is a source of noise. Noise can also be the result of damage to the film, or be introduced by the scanner itself.
- If the image is acquired directly in a digital format, the mechanism for gathering the data (such as a CCD detector) can introduce noise.
- Electronic transmission of image data can introduce noise.

To simulate the effects of some of the problems listed above, the toolbox provides the imnoise function, which you can use to *add* various types of noise to an image. The examples in this section use this function.

6.1.3 Remove Noise by Linear Filtering

You can use linear filtering to remove certain types of noise. Certain filters, such as averaging or Gaussian filters, are appropriate for this purpose. For example, an averaging filter is useful for removing grain noise from a photograph. Because each pixel gets set to the average of the pixels in its neighbourhood, local variations caused by grain are reduced.

Remove Noise using an Average Filter and Median Filter

Removing noise from an image using an averaging filter and a median filter to allow comparison of the results. These two types of filtering both set the value of the output pixel to the average of the pixel values in the neighbourhood around the corresponding input pixel. However, with median filtering, the value of an output pixel is determined by the median of the neighbourhood pixels, rather than the mean. The median is much less sensitive than the mean to extreme values (called outliers). Median filtering is therefore better able to remove these outliers without reducing the sharpness of the image.

Note: Median filtering is a specific case of order-statistic filtering, also known as rank filtering. For information about order-statistic filtering, see the reference page for the ordfilt2 function.

Read image into the workspace and display it.

```
I = imread('input.jpg');
figure
imshow(I)
```

Fig 6.3 Input Image

Filter the noisy image, J, with an averaging filter and display the results. The example uses a 3-by-3 neighbourhood.

Now use a median filter to filter the noisy image, J. The example also uses a 3-by-3 neighbourhood. Display the two filtered images side-by-side for comparison. Notice that medfilt2 does a better job of removing noise, with less blurring of edges of the coins.

Fig 6.4 Pair Images, a) Image with Noise b) Image with reduced noise

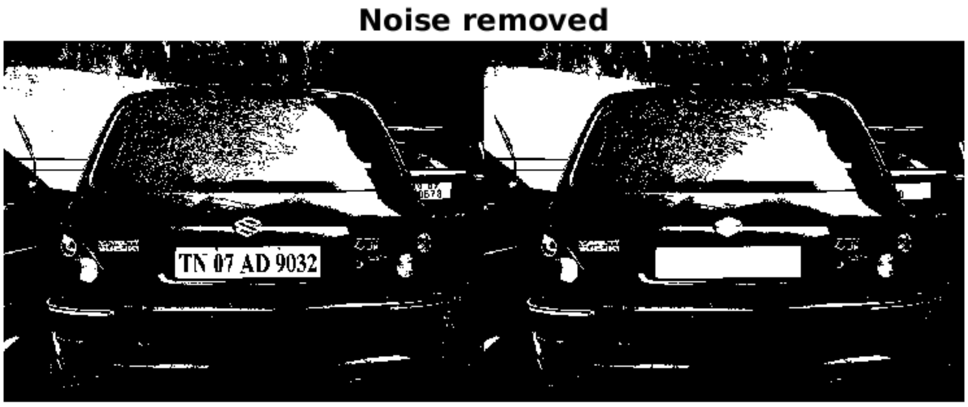

Fig 6.5 Changes observed by reducing Noise

6.1.4 Detecting the boundary of license plate

The Gray Scale image is converted to binary image (which contains only 0's and 1's. 0 representing black and 1 representing white). Noise is removed from the binary image and is further processed for location the bounding box of the license plate. Using the regionprops of the binary image the image is further filtered by replacing the unwanted area of pixels with 0 and the boundary box representing the license plate are taken into consideration. After removing the unwanted area, the binary image is will be as fig 6.6. From those values the boundary box values are taken for segmentation.

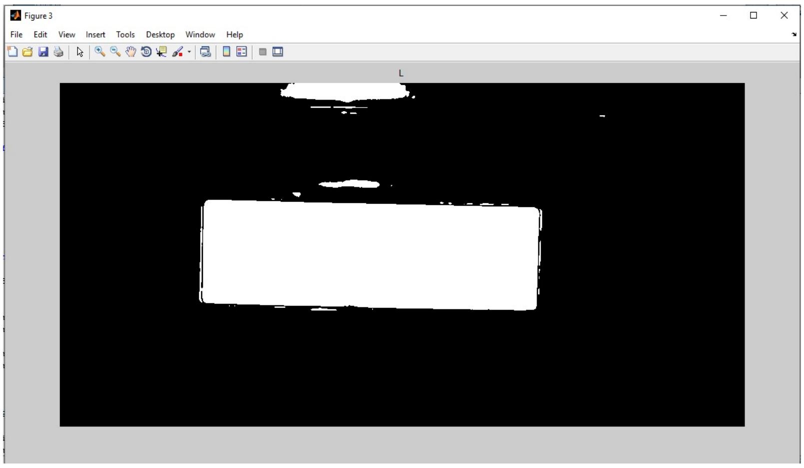

Fig 6.6 Image after Noise Reduction.

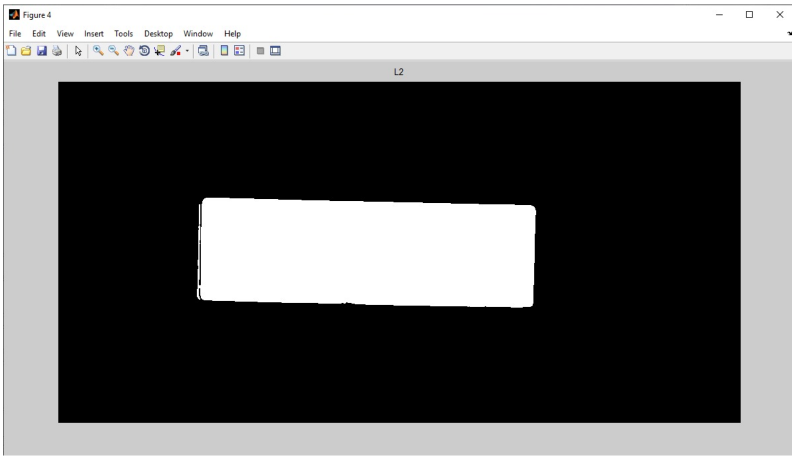

Fig 6.7 regionprops filtered License Plate Area

6.2 Image Segmentation

Segmentation is one of the most important processes in the automatic number plate recognition, because all further steps rely on it. In MATLAB, the function region props (for "region properties") provides a shortcut for determining many properties of a black and white or labelled image. Measure properties of image regions (blob analysis) the regionprops syntax is STATS = regionprops (L, properties), it measures a set of properties for each labelled region in the label matrix L. Positive integer elements of L correspond to different regions. For example, the set of elements of L equal to 1 corresponds to region 1; the set of elements of L equal to 2 corresponds to region 2; and so on. The return value STATS is a structure array of length max (L (:)). The fields of the structure array denote different measurements for each region, as specified by properties. Properties can be a comma-separated list of strings, a cell array containing strings, the single string 'all', or the string 'basic'. If properties are the string 'all', regionprops computes all the preceding measurements. If properties are not specified or if it is the string 'basic', regionprops computes only the 'Area', 'Centroid', and 'Bounding Box' measurements. Fig 6.7 shows the segmented section of the cropped image by using regionprops.

The Gray Scale image is cropped to the boundary box generated from the fig. 6.6 and is processed for segmentation. The segmented images are shown in fig. 6.7.

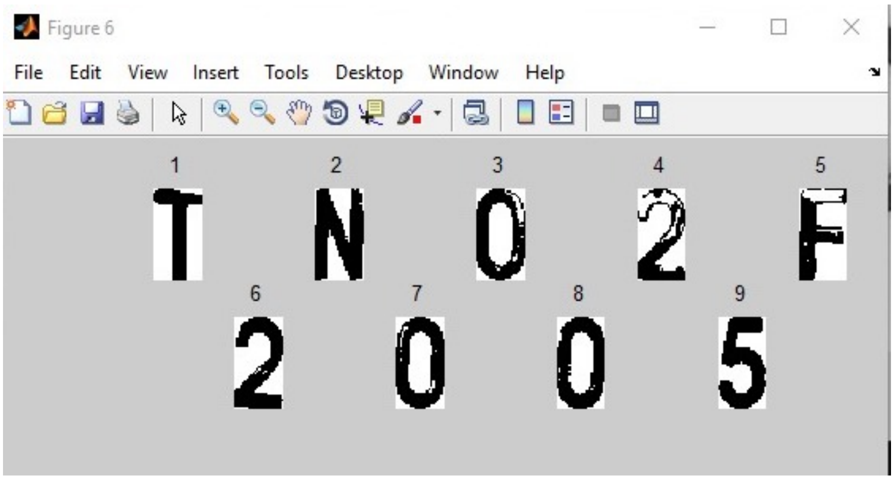

Fig 6.8 Segmented images

6.3 Character Recognition

After segmentation, the images are given to the character recognition section. I this section each image that had been segmented is compared with each images or templates of the required characters/numbers in the database and among those images, the image which results maximum correlation is taken into consideration. The correlation coefficient (r) measures the linear dependence between the two variables or datasets. There are different types of correlation coefficients. The research uses the Pearson's correlation coefficient in the proposed algorithm. The mathematical formula for computing correlation coefficient between two datasets x and y is as follows,

$$r = \frac{\sum_m \sum_n (A_{mn} - \bar{A})(B_{mn} - \bar{B})}{\sqrt{\left(\sum_m \sum_n (A_{mn} - \bar{A})^2\right)\left(\sum_m \sum_n (B_{mn} - \bar{B})^2\right)}}$$

where, {A} and {B} are two datasets containing n values and x and y are mean values of these datasets respectively. As explained earlier, the Correlation Coefficient indicates the strength and the direction of the linear relationship between two datasets. It may take any value from -1 and +1. A value of +1 shows total positive linear correlation, 0 shows no linear correlation at all and -1 shows total negative linear correlation between two datasets. A perfect Correlation of ±1 occurs only when all the points of the dataset fall on a single straight line. A value greater than 0.8 generally signifies a strong correlation between the datasets whereas a value less than 0.5 generally signifies a weak correlation. Every correlation coefficient of the segmented image is stored in a variable and the value representing the maximum correlation value which refers to maximum relationship between images is taken into consideration. This process is repeated for every image from the segmentation results and are stored in a variable. Finally, these values are used for generating the license plate string and is displayed in the command window.

CHAPTER 7

RESULTS

Results:

Step 1: Input image is read and displayed on the screen.

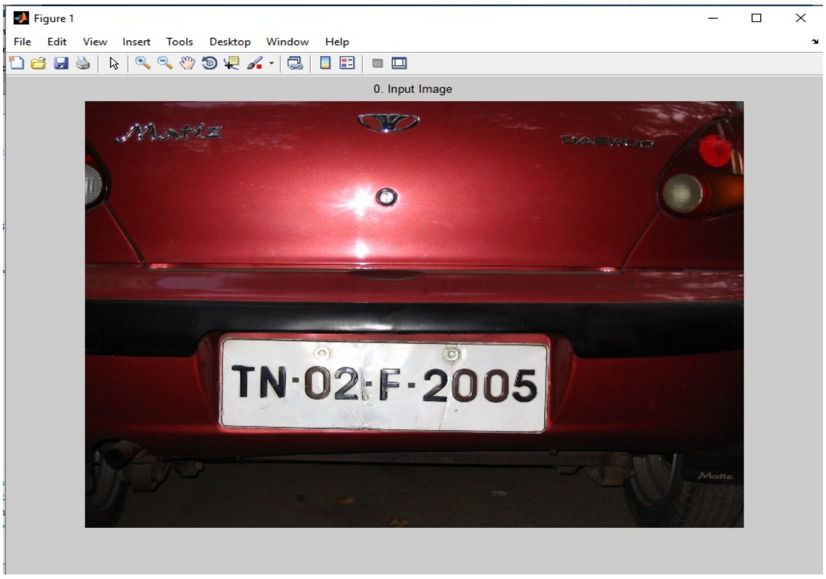

Fig 7.1 Input Image

Step 2: The input RGB image is converted to Gray Image

Fig 7.2 Image Converted from RGB to Gray.

Step 3: Creating a binary image of the grayscale image

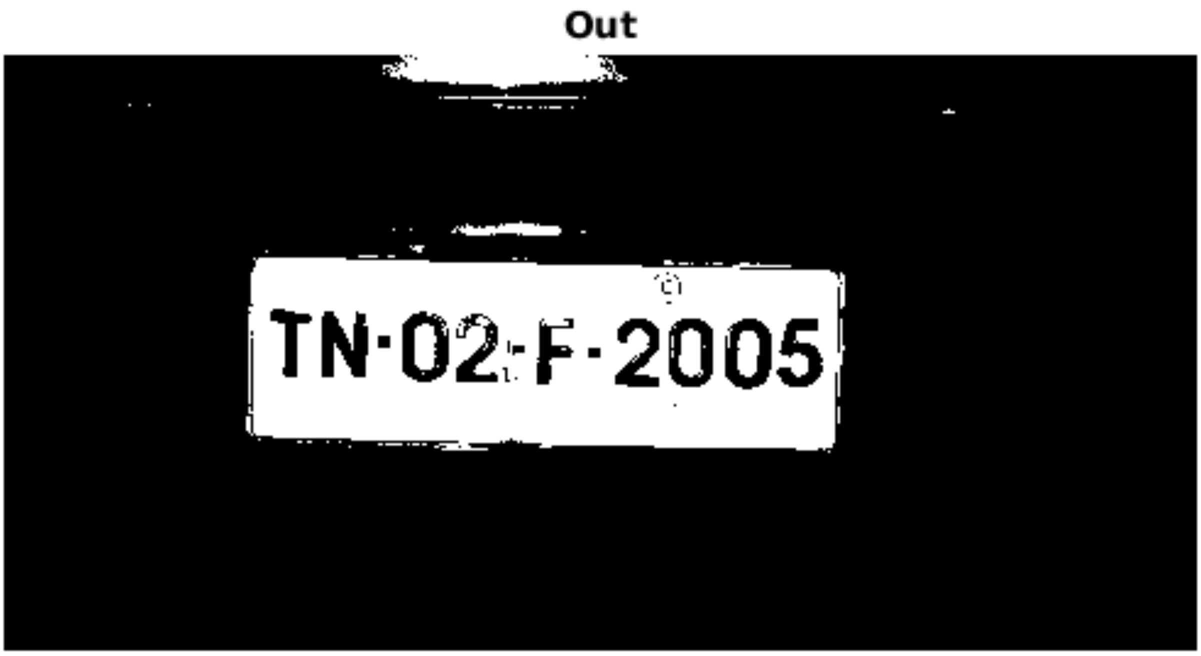

Fig. 7.3 Binary Image

Step 4: Reducing the Noise in the Binary Image.

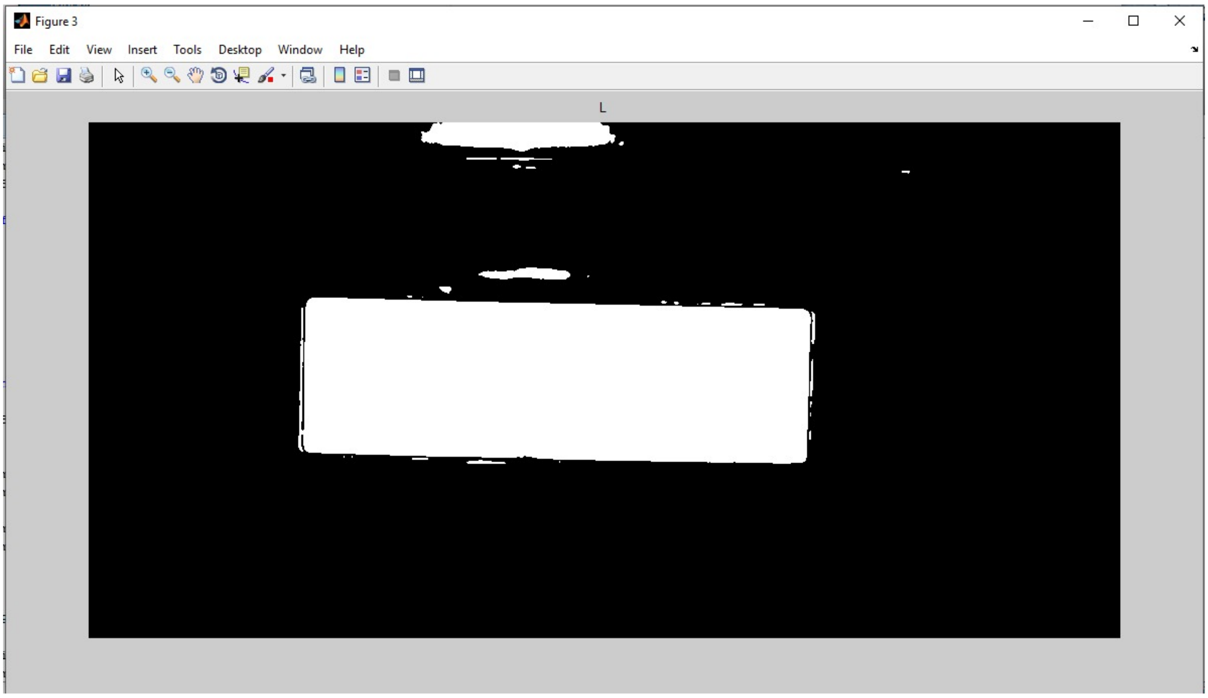

Fig 7.4 Median Filtered Image

Step 5: Removing the extra noise present in the image 7.4 using image regionprops.

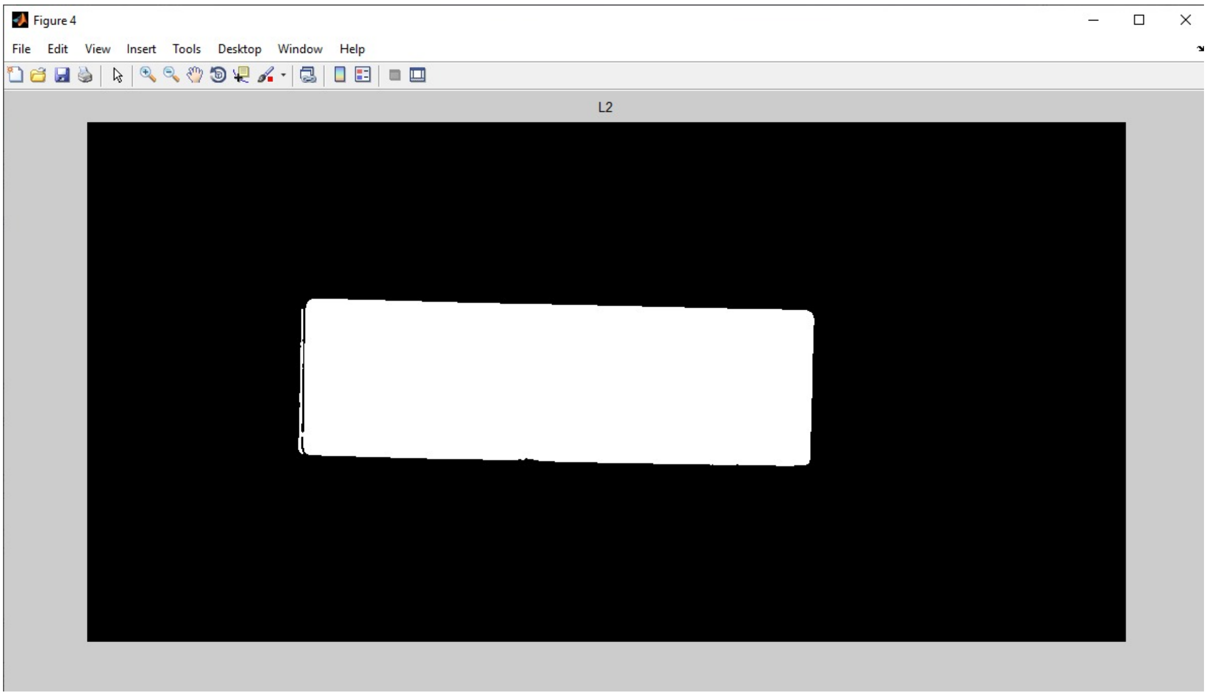

Fig 7.5 Filtered Image Used for detecting the Boundary of license plate

Step 6: Cropping the Gray image (fig 7.2) with the boundary values obtained from fig 7.4.

Fig 7.6 Gray Image Cropped to License plate from the values achieved from the above image

Step 7: The resulting image from the fig 7.6 is given to the split function where segmentation of the image is done based on the image regionprops and segmented character images are displayed.

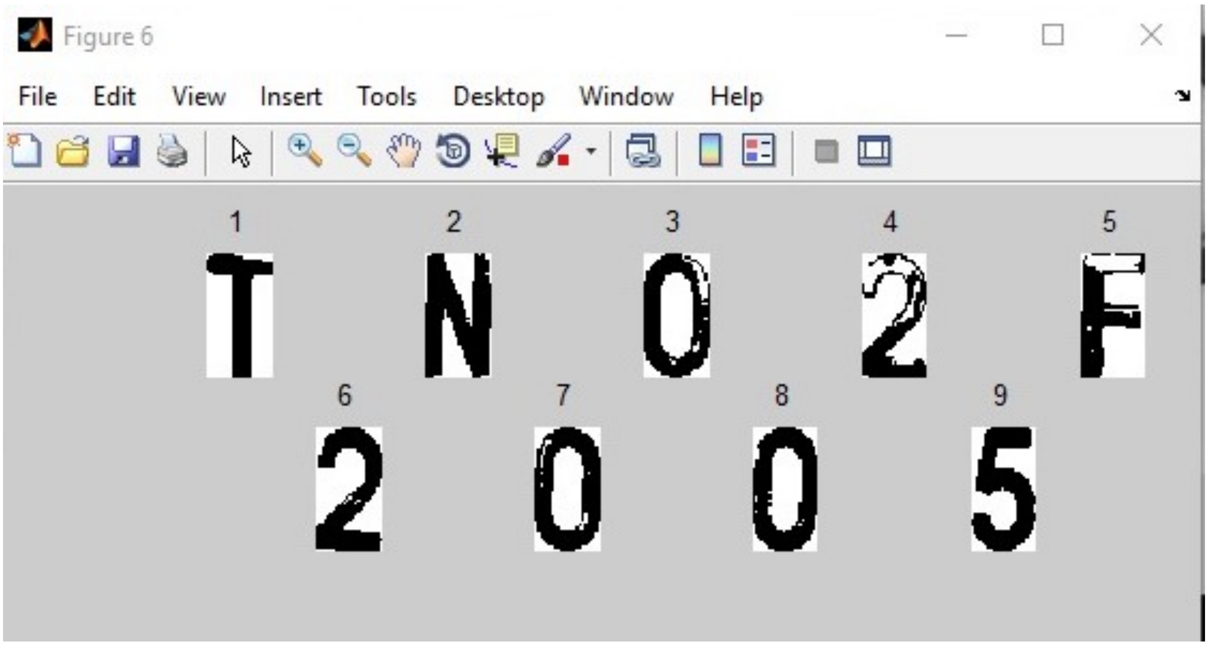

Fig 7.7 Segmented Images

Step 8: The segmented images are compared or matched with the image properties or correlated with the image properties in the database and the resulting values are displayed.

Fig 7.8 Final Results

CHAPTER 8
CONCLUSION

8.1 ADVANTAGES

- Flexibility.
- Invariant to license plate position.
- High performance Accuracy.
- Less time consumption compared to existed system

8.2 APPLICATIONS

- In Government Sector
- At Toll Gates
- In Parking Areas
- In Hospitals
- In Schools And Colleges Etc.…

8.3 CONCLUSION

In This project we have checked and evaluated the accuracy of the OCR technique. The Template matching affects the accuracy of number plate recognition. We have found that there are some factors which affect the effectiveness of template matching based on OCR technique i.e. font type, noise in image, tilting etc. In future the work can be done on these factors and efficiency may be increased further for better results.

8.4 FUTURE SCOPE

The future scope is that the vehicle recognition system Plays major role in detecting threats to defence. Also it can Improve the security related to the women's as they can easily detect the number plate before using cab or other services. The system Robustness can be increase, if bright and sharp camera is used

REFERENCES

[1] K. K. Kim, K. I. Kim, J. B. Kim and H. J. Kim, Learning-based approach for license plate recognition", in Proc. IEEE Signal Process. Soc. Workshop, Neural Network Signal Process. 2000, vol. 2, pp. 614-623.

[2] T. Naito, T. Tsukada, K. Yamada, K. Kozuka, and S. Yamamoto, "Robust License-Plate Recognition Method for Passing Vehicle Under Outside Environment", IEEE Trans. On Vehicular Technology, Vol. 49, No.6, November 2000, pp. 2309-2319.

[3] M.A. Ko and Y.M. Kim, "License Plate Surveillance System Using Weighted Template Matching", in Proc. of the 32nd Applied Imagery Pattern Recognition Workshop, October 2003, pp. 269-274.

[4] X. Pan, X. Ye and S. Zhang, "A hybrid method for robust car plate character recognition", presented at the IEEE International Conference on Systems, Man and Cybernetics,2004, pp. 4733-4737.

[5] Y. Huang, S. Lai and W. Chuang, "A Template-Based Model for License Plate Recognition", in The 2004 IEEE International Conference on Networking, Sensing & Control, Taipei,2004,pp. 737-742.

[6] Parul Shah, Sunil Karamchandani, Taskeen Nadkar, Nikita Gulechha, Kaushik Koli, Ketan Lad, "OCR-based Chassis- Number Recognition using Artificial Neural Networks", presented at the IEEE ICVES 2009, pp. 31-34.

[7] Muhammad Tahir Qadri, Muhammad Asif, " Number Plate Recognition System for Vehicle Identification using Optical Character Recognition", in IEEE 2009 International Conference on Education Technology and Computer, 2009, pp. 335-338.

[8]S. Hamidreza Kasaei, S. Mohammadreza Kasaei, S. Alireza Kasaei, "New Morphology-Based Method for Robust Iranian Car Plate Detection and Recognition", International Journal of Computer Theory and Engineering, Vol. 2, No. 2 April, 2010 1793-8201. pp. 264-268.

[9] Xiaojun Zhai, Faycal Bensaali and Reza Sotudeh, " OCR Based Neural Network for NPR", in IEEE 2012.

APPENDIX
MATLAB

Introduction

MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. MATLAB stands for matrix laboratory, and 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 array that does not require pre dimensioning which to solve many technical computing problems, especially those with matrix and vector formulations, in a fraction of time.

MATLAB features a family of applications specific solutions called toolboxes. Very important to most users of MATLAB, toolboxes allow learning and applying specialized technology. These are comprehensive collections of MATLAB functions (M- files) that extend the MATLAB environment to solve particular classes of problems. Areas in which toolboxes are available include signal processing, control system, neural networks, fuzzy logic, wavelets, simulation and many others. Typical uses of MATLAB include: Math and computation, Algorithm development, Data acquisition, Modelling, simulation, prototyping, Data analysis, exploration, visualization, Scientific and engineering graphics, Application development, including graphical user interface building.

Basic building blocks of MATLAB:

The basic building block of MATLAB is MATRIX. The fundamental data type is the array. Vectors, scalars, real matrices and complex matrix are handled as specific class of this basic data type. The built in functions are optimized for vector operations.

MATLAB Window

The MATLAB works based on five windows: Command window, Workspace window, Current directory window, Command history window, Editor Window, Graphics window and Online-help window.

Command Window

The command window is where the user types MATLAB commands and expressions at the prompt (>>) and where the output of those commands is displayed. It is opened when the application program is launched. All commands including user-written programs are typed in this window at MATLAB prompt for execution.

Workspace Window

MATLAB defines the workspace as the set of variables that the user creates in a work session. The workspace browser shows these variables and some information about them. Double clicking on a variable in the workspace browser launches the Array Editor, which can be used to obtain information.

Current Directory Window

The current Directory tab shows the contents of the current directory, whose path is shown in the current directory window. For example, in the windows operating system the path might be as follows: C:\MATLAB\Work, indicating that directory "work" is a subdirectory of the main directory "MATLAB"; which is installed in drive C. Clicking on the arrow in the current directory window shows a list of recently used paths. MATLAB uses a search path to find M-files and other MATLAB related files. Any file run in MATLAB must reside in the current directory or in a directory that is on search path.

Command History Window

The Command History Window contains a record of the commands a user has entered in the command window, including both current and previous MATLAB sessions.

Editor Window

The MATLAB editor is both a text editor specialized for creating M-files and a graphical MATLAB debugger. The editor can appear in a window by itself, or it can be as u window in the desktop. In this window one can write, edit, create and save programs in files called M-files. MATLAB edit or window has numerous pull-down menus for tasks such as saving, viewing, and debugging files. Because it performs some simple check sandals colour to differentiate between various elements of code, this text editor is recommended as the tool of choice for writing and editing M- functions.

Graphics or Figure Window

The output of all graphic commands typed in the command window is seen in this window.

Online Help Window

MATLAB provides online help for all it's built in functions and programming language constructs. The principal way to get help online is to use the MATLAB help browser, opened as a separate window either by clicking on the question mark symbol (?) on the desktop toolbar, or by typing help browser at the prompt in the command window. The help Browser is a web browser integrated into the MATLAB desktop that displays a Hypertext Mark-up Language (HTML) documents. The Help Browser consists of two panes, the help navigator pane, used to find information, and the display pane, used to view the information. Self-explanatory tabs other than navigator pane are used to perform a search.

M-Files

These are standard ASCII text file with 'm' extension to the file name and creating own matrices using M-files, which are text files containing MATLAB code. MATLAB editor or another text editor is used to create a file containing the same statements which are typed at the MATLAB command line and save the file under a name that ends in .m. There are two types of M-files:

Script Files

It is an M-file with a set of MATLAB commands in it and is executed by typing name of file on the command line. These files work on global variables currently present in that environment.

Function Files

A function file is also an M-file except that the variables in a function file are all local. This type of files begins with a function definition line.

MAT-Files

These are binary data files with '.mat' extension to the file that are created by MATLABwhenthedataissaved.ThedatawritteninaspecialformatthatonlyMATLAB can read. These are located into MATLAB with 'load' command.

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) MATLAB Mathematical Function:

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) 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 complete large and complex 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)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.
Whose list the variables and describes their matrix size clear erases variables and functions from memory clear x erases the matrix 'x' from your work space close by itself, closes the current figure window figure creates an empty figure window hold on holds the current plot and all axis properties so that subsequent graphing commands add to the existing graph hold off sets the next plot property of the current axes to "replace" find find indices of nonzero elements. X label (‘’): Allows you to label x-axis Y label (‘‘): Allows you to label y-axis Title (‘‘): Allows you to give title for plot Subplot (): Allows you to create multiple plots in the same window.

SOME BASIC PLOTCOMMANDS:

Kinds of plots: plot(x,y) creates a Cartesian plot of the vectors x &y plot(y) creates a plot of y vs. the numerical values of the elements in they-vector loglog(x,y) plots log(x) vs log(y) polar (theta,r) creates a polar plot of the vectors r & theta where theta is in radians bar(x) creates a bar graph of the vector x. (Note also the command stairs(x)) bar(x,y) creates a bar-graph of the elements of the vector y, locating thebars according to the vector elements of 'x' .

Plot description:

Grid: creates a grid on the graphics plot.

title('text'): places a title at top of graphics plot.

X label('text'): writes 'text' beneath the x-axis of a plot.

Y label('text'): writes 'text' beside the y-axis of a plot.

Text (x,y,'text'): writes 'text' at the location (x,y).

text(x,y,'text','sc'): writes 'text' at point x,y assuming lower left corner is (0,0) and upper right corner is (1,1)

axis([xmin xmax ymin ymax]): sets scaling for the x- and y-axes on the current plot.

ALGEBRIC OPERATIONS IN MATLAB:

Scalar Calculations:

+ Addition

- Subtraction

* Multiplication

/ Right division (a/b means a ÷b)

For example: 3*4 executed in 'matlab' gives ans=124/5 gives ans=0.8

Array products: Recall that addition and subtraction of matrices involved addition or subtraction of the individual elements of the matrices. Sometimes it is desired to simply multiply or divide each element of an matrix by the corresponding element of another matrix 'array operations". Array or element-by-element operations are executed when the operator is preceded by a '.' (Period): a.*b multiplies each element of a by the respective element of b a./b respective element of b.

a.\b divides each element of b by the respective element of a a.^ b respective b element

MATLAB WORKINGENVIRONMENT:

divides each element raise each element
of a by the of a by the
MATLABDESKTOP

Matlab Desktop is the main Matlab application window. The desktop contains five sub windows, the command window, the workspace browser, the current directory window, the command history window, and one or more figure windows, which are shown only when the user displays a graphic.

The command window is where the user types MATLAB commands and expressions at the prompt (>>) and where the output of those commands is displayed. MATLAB defines the workspace as the set of variables that the user creates in a work session. can be used to obtain information and income instances edit certain properties of the variable. The current Directory tab above the workspace tab shows the contents of the current directory, whose path is shown in the current directory window. For example, in the windows operating system the path might be as follows: C:\MATLAB\Work, indicating that directory "work" is a subdirectory of the main directory “MATLAB”; WHICH IS INSTALLED IN DRIVE C. clicking on the arrow in the current directory window shows a list of recently used paths. Clicking on the button to the right of the window allows the user to change the current directory.

MATLAB uses a search path to find M-files and other MATLAB related files, which are organize in directories in the computer file system. Any file run in MATLAB must reside in the current directory or in a directory that is on search path. By default, the files supplied with MATLAB and math works toolboxes are included in the search path. The easiest way to see which directories are soon the search path, or to add or modify a search path, is to select set path from the File menu the desktop, and then use the set path dialog box. It is good practice to add any commonly used directories to the search path to avoid repeatedly having the change the current directory.

The Command History Window contains a record of the commands a user has entered in the command window, including both current and previous MATLAB sessions. Previously entered MATLAB commands can be selected and re-executed from the command history window by right clicking on a command or sequence of commands.

This action launches a menu from which to select various options in addition to executing the commands. This is useful to select various options in addition to executing the commands. This is a useful feature when experimenting with various commands in a work session.

SOURCE CODE

MAIN.M

```
clc
clear all
close all
[filename, pathname] = uigetfile('*.jpg', 'Pick an Image');
    if isequal(filename,0) | isequal(pathname,0)
       warndlg('User pressed cancel')
    else
                    a=imread(filename);
                    figure,
                    imshow(a);
    end
a=rgb2gray(a);
[r c p]=size(a);
b=a(r/3:r,1:c);
figure,
imshow(b);title('LP AREA')
[r c p]=size(b);
Out=zeros(r,c);
for i=1:r
    for j=1:c
      if b(i,j)>150
          Out(i,j)=1;
        else
          Out(i,j)=0;
      end
    end
end
```

```
BW3 = bwfill(Out,'holes');
BW3=medfilt2(BW3,[3 3]);
BW3=medfilt2(BW3,[3 3]);
BW3=medfilt2(BW3,[3 3]);
BW3=medfilt2(BW3,[5 5]);
BW3=medfilt2(BW3,[5 5]);
[L num]=bwlabel(BW3);
STATS=regionprops(L,'all');
removed=0;
for i=1:num
dd=STATS(i).Area;
   if (dd < 50000)
         L(L==i)=0;
                     removed = removed + 1;
         num=num-1;
               else
   end
end
[L2 num2]=bwlabel(L);
 STATS = regionprops(L2,'All');
if num2>2
    for i=1:num2
              aa=  STATS(i).Orientation;
            if aa > 0
              imshow(L==i);
            end
   end
end
```

```
 B=STATS.BoundingBox;
Xmin=B(2);
Xmax=B(2)+B(4);
Ymin=B(1)
Ymax=B(1)+B(3);
LP=b(Xmin+25:Xmax-20,Ymin+10:Ymax-10);
figure,
imshow(LP,[]);
I=split(LP);
disp(I);
Len=length(I);
LEN=length(I);
load DataBase;
RECOG=[];
Dlen=540
k=1;
for i=1:LEN
   Test=I{i};
   for j=1:Dlen
                  Test2=DataBase{j};
                  X(j)=corr2(Test,Test2);
   end
   [Res INDEX]=max(X);
   RECOG(k)=INDEX;
   k=k+1;
                  disp(INDEX);
end
disp('Exit');
```

```
Len=length(RECOG);
Output=[]
for i=1:Len
a=    RECOG(i);
            if (a>=1)&(a<=15)
                        Output=[Output '0']
    elseif (a>=16)&(a<=30)
                        Output=[Output '1']
            elseif (a>=31)&(a<=45)
                        Output=[Output '2']
     elseif (a>=46)&(a<=60)
            Output=[Output '3']
    elseif (a>=61)&(a<=75)
                        Output=[Output '4']
    elseif (a>=76)&(a<=90)
                        Output=[Output '5']
    elseif  (a>=91)&(a<=105)
                        Output=[Output '6']
     elseif  (a>=106)&(a<=120)
                        Output=[Output '7']
     elseif (a>=121)&(a<=135)
                        Output=[Output '8']
    elseif  (a>=136)&(a<=150)
                        Output=[Output '9']
    elseif  (a>=151)&(a<=165)
                        Output=[Output 'A']
            elseif  (a>=166)&(a<=180)
                        Output=[Output 'B']
```

```
elseif  (a>=181)&(a<=195)
            Output=[Output 'C']
    elseif  (a>=196)&(a<=210)
            Output=[Output 'D']
    elseif  (a>=211)&(a<=225)
            Output=[Output 'E']
elseif  (a>=226)&(a<=240)
            Output=[Output 'F']
elseif  (a>=241)&(a<=255)
            Output=[Output 'G']
elseif  (a>=256)&(a<=270)
            Output=[Output 'H']
elseif  (a>=271)&(a<=285)
            Output=[Output 'I']
elseif  (a>=286)&(a<=300)
            Output=[Output 'J']
elseif  (a>=301)&(a<=315)
            Output=[Output 'K']
elseif  (a>=316)&(a<=330)
            Output=[Output 'L']
elseif  (a>=331)&(a<=345)
            Output=[Output 'M']
elseif  (a>=346)&(a<=360)
            Output=[Output 'N']
elseif  (a>=361)&(a<=375)
            Output=[Output 'O']
elseif  (a>=376)&(a<=390)
            Output=[Output 'P']
```

```
    elseif  (a>=391)&(a<=405)
                    Output=[Output 'Q']
    elseif  (a>=406)&(a<=420)
                    Output=[Output 'R']
    elseif  (a>=421)&(a<=435)
                    Output=[Output 'S']
    elseif  (a>=436)&(a<=450)
                    Output=[Output 'T']
    elseif  (a>=451)&(a<=465)
                    Output=[Output 'U']
    elseif  (a>=466)&(a<=480)
                    Output=[Output 'V']
    elseif  (a>=481)&(a<=495)
                    Output=[Output 'W']
    elseif  (a>=496)&(a<=510)
                    Output=[Output 'X']
    elseif  (a>=511)&(a<=525)
                    Output=[Output 'Y']
    elseif  (a>=526)&(a<=540)
                    Output=[Output 'Z']
    end
end
disp('Letters');
disp(Output);
```

SPLIT.M

```
function I=split(a);
% a=imread('real2.bmp');
% imshow(a);
[r c p]=size(a);
if p==3
   a=rgb2gray(a);
end
% imshow(a);
% pixval on;
aaa=a;
% a=im2double(a);
for i=1:r % rows
   for j=1:c %
      dd=a(i,j);
      if dd >125
      Out(i,j)=255;
   else
      Out(i,j)=0;
   end
   end
end
a=Out;
% imshow(a);
% pixval on;
b=im2bw(a);
c=imcomplement(b);
% imshow(b);
[L num]=bwlabel(c);
```

```
STATS = regionprops(L,'all');
disp(num);
Data=[];
for i=1:num %. number of characters in number plate
   Data1=STATS(i).Area
   Data=[Data Data1];
end
Data2=-sort(-Data);
Data3=Data2(1:9);
%         L = bwlabel(BW,4);
%         [r,c] = find(L == 2);
AreaData=min(Data3);
removed=0;
for i=1:num
   dd=STATS(i).Area;
   if dd < AreaData
            L(L==i)=0;
            removed = removed + 1;
      num=num-1;
   % else
   % imshow(L==i);
   end
end
[L2, num2]=bwlabel(L);
disp(num);
stats1 = IMFEATURE(L2, 'Image'); % get image features
C = [];
str='.bmp';
for j=1:1:(num2)
```

```
c = stats1(j);
C{j} = [c.Image]; % sepreate diffrent objects into C cell array.
%     C{j} = imresize(C{j}, [45 24], 'bilinear'); % resize objects for SVM
 EE = imresize(C{j}, [45 24]);
%     figure;
%     imshow(EE,[]);
EE=imcomplement(EE);
imshow(EE,[]);
 C{j}=EE;
 %     ee=uint8(EE);
 [r c]=size(EE);
 dd=zeros(r,c);
 end
 I=C;
```

****