**ACKNOWLEDGEMENT**

The successful completion of this project work was made possible with the help and guidance received by me from quarters. We would like to avail this opportunity to express our sincere thanks and gratitude to all of them.

We express our sincere gratitude to **Dr. Narendra Vishwanath**, Principal, GAT, for providing us a technically sound environment in college.

We also take this opportunity to express our heartfelt thanks to **Prof. H. S. Manjunatha Reddy,** Head of the Department of Electronics and Communication, GAT, who was a constant source of inspiration.

We are humbly thankful to **Mr. Tilak Kumar L.** and other lecturers for their support and encouragement.

We also thank our teaching and non-teaching staffs who have helped us to steer pure path to success indirectly or directly.

**UDAY WALVEKAR**

**MARTIN J.A**

**RAVI S.P**

**VENKATA REDDY**

**ABSTRACT**

A novel approach for number plate recognition is presented based on artificial neural network principles.

The images of number plate in the image database have been processed for various image processing techniques to obtain number plate using matlab software. The fundamental processes of image processing like image acquisition and morphological processing are used to perform the number plate extraction. The individual characters are then extracted using the segmentation process in which the position of each character pixel wise in rows and columns is stored in some variable. The variables are then used to extract the character by cropping it from the main image and creating a sub image.

The sub images of characters are then trained by artificial neural networks for the process of authentication. Now the trained characters are stored as a database which is used as a reference for authenticating the number plate which appears at toll gate. The valid number plates are allowed and others are not allowed.

As Neural Network is an intelligence engine, it ensures greater accuracy and better recognition speed. The recognized characters are sent over serial communication to microcontroller for display as well as the status of authentication whether valid or not.

**Contents**

**1. INTRODUCTION**

1.1 WHAT IS NPR………………………………………………………………6

1.2 A TYPICAL NPR SYSTEM………………………………………………...6

1.3 PROPOSED NPR SYSTEM………………………………………………...6

1.4 APPLICATION OF NPR SYSTEM………………………………………...7

1.5 PROBLEM STATEMENT…………………………………………………..8

1.6 OBJECTIVE…………………………………………………………………9

1.7 IMAGE PROCESSING……………………………………………………..9

1.8 NEURAL NETWORK……………………………………………………...14

1.9 MICROCONTROLLER 8051……………………………………………...15

1.10 MAX 232…………………………………………………………………..15

1.11 LCD DISPLAY……………………………………………………………15

1.12 ULN 2803………………………………………………………………….16

1.13 STEPPER MOTOR………………………………………………………..16

**2. Literature survey**

2.1 IMAGE ACQUISITION…………………………………………………...18

2.2 NUMBER PLATE EXTRACTION………………………………………..18

2.3 SEGMENTATION………………………………………………………….18

2.4 IMAGE RESTORATION…………………………………………………..19

2.5 NEURAL NETWORK TRAINING STYLES……………………………...19

2.6 PROBABLISTIC NEURAL NETWORK………………………………….21

**3. BLOCK DIAGRAM**

3.1 PROJECT BLOCK DIAGRAM……………………………………………24

3.2 BLOCK DIAGRAM FOR MICROCONTROLLER SECTION…………...26

**4. DESIGN AND IMPLEMENTATION**

4.1 FLOWCHART FOR IMAGE PROCESSING SECTION…………………..28

4.2 FLOWCHART FOR MICROCONTROLLER SECTION………….……….29

4.3 PROJECT IMPLEMENTATION STEPS……………………………………32

4.4 MICROCONTROLLER DATA FLOW STEPS…………………………….41

**5. RESULT**

5.1 SETTINGS……………………………………………………………………43

5.2 RESULT………………………………………………………………………43

5.3 CONCLUSION………………………………………………………………44

5.4 FUTURE WORK……………………………………………………………..44

**6. BIBLIOGRAPHY**……………………………………...……………….45

**LIST OF FIGURES AND TABLES**

1. NEURAL NETWORK TRAINING DIAGRAM.....................................................................................14
2. PROBABLISTIC NEURAL NETWORK BLOCK DIAGRAM.....................................................................................22
3. PROJECT BLOCK DIAGRAM.....................................................................................24
4. BLOCK DIAGRAM FOR MICROCONTROLLER SECTION.......................................................................................27
5. FLOW CHART FOR IMAGE PROCESSING SECTION..................................................................................28-29
6. FLOW CHART FOR MICROCONTROLLER SECTION..................................................................................30-31
7. PROJECT IMPLEMENTATION FIGURES (15FIGURES)...........................................................................32-36
8. PLATE LOCALISATION AND CARACTER SEGMENTATION........................................................................43

**1. INTRODUCTION**

**1.1 What is NPR?**

Number plate recognition (NPR) is an image-processing technology used to identify vehicles by their number plates. This technology is gaining popularity in security and traffic installations. Much research has already been done for the recognition of number plates. This thesis presents a number plate recognition system as an application of computer vision. Computer vision is a process of using a computer to extract high level information from a digital image.

NPR systems normally consist of an image database, illumination, frame grabber, computer, software and hardware.

**1.2 A Typical NPR System**

When the vehicle approaches the secured area, the NPR unit senses the car and activates the illumination. The NPR unit takes the pictures from the front plates from the NPR camera. The image of the vehicle contains the number plate. The NPR unit feeds the input image to the system. The system then enhances the image, detects the plate position, extracts the plate, segments the characters on the plate and recognizes the segmented characters.

After that recognized character is send to LCD, after displaying in LCD signals to open the gate. After passing the gate closed. Now the system waits for the next vehicle to approach the secured area.

**1.3 Structure of the Proposed NPR System**

The system presented is designed to recognize number plates from the front of the vehicle. Input to the system is an image sequence acquired by a digital camera that consists of a number plate and its output is the recognition of characters on the number plate. The system consists of the standard four main modules in an NPR system, viz. Image acquisition, number plate extraction, and number plate segmentation and number plate recognition. The first task acquires the image. The second task extracts the region that contains the number plate. The third task isolates the characters, letters and

numerals, as in the case of Indian Number Plates. The last task identifies or recognizes the segmented characters.

**1.3.1 Image Acquisition** This is the first phase in an NPR system. This phase deals with acquiring an image by an acquisition method. In our proposed system, we use a high resolution digital camera to acquire the input image. The input image is 3648 x 2056 pixels.

**1.3.2 Number Plate Extraction** Number Plate Extraction is a key step in an NPR system, which influences the accuracy of the system significantly. This phase extracts the region of interest, i.e., the license plate, from the acquired image. The proposed approach involves “defined region of plate”.

**1.3.3 Number Plate Segmentation** Number Plate Segmentation, which is sometimes referred to as Character Isolation takes the region of interest and attempts to divide it into individual characters.

**1.3.4 Number Plate Recognition** The last phase in NPR system is to recognize the isolated characters. After splitting the extracted number plate into individual character images, the character in each image can be identified. There is method used to recognize isolated characters Probabilistic Neural Networks.

**1.4 Applications of NPR Systems**

Vehicle number plate recognition is one form of automatic vehicle identification system. NPR systems are of considerable interest, because of their potential applications to areas such as highway electronic toll collection, automatic parking attendant, petrol station forecourt surveillance, speed limit enforcement, security, customer identification enabling personalized services, etc. Real time NPR plays a major role in automatic monitoring of traffic rules and maintaining law enforcement on public roads.

**Law Enforcement: -** The plate number is used to produce a violation fine on speeding vehicles, illegal use of bus lanes, and detection of stolen or wanted vehicles. The main advantage is that the system can store the image record for future references. The front part of the vehicle is extracted off the filmed image and is given to the system for processing. The processed result is fed into the database as input. The violators can pay the fine online and can be presented with the image of the car as a proof along with the speeding information.

**Parking: -** The NPR system is used to automatically enter pre-paid members and calculate parking fee for non-members. The car plate is recognized and stored upon its exit the car plate is read again and the driver is charged for the duration of parking.

**Automatic Toll Gates: -** Manual toll gates require the vehicle to stop and the driver to pay an appropriate tariff. In an automatic system the vehicle would no longer need to stop. As it passes the toll gate, it would be automatically classified in order to calculate the correct tariff.

**Border Crossing: -** This application assists the registry of entry or exits to a country, and can be used to monitor the border crossings. Each vehicle’s information is registered into a central database and can be linked to additional information.

**1.5 Problem Statement**

**1.5.1 Software implantation of character recognition in vehicle number plates using Neural Network by**

* Image capture.
* Image processing.
* Segmentation and recognition.

**1.5.2. Development of hardware and software for character display in LCD and control of stepper using microcontroller.**

* Power supply board 5 and 12 volts
* 89C51 microcontroller board
* LCD board
* Max 232 board
* ULN 2804 board

**1.5.3. Testing of the hardware and software developed for the system.**

**1.6 Objective**

The work presented here aims at the following aspects.

* Study of MATLAB Toolbox like Image Processing and Neural Network toolbox.
* Study the existing license plate recognition systems.
* Develop a car license plate recognition system using Image Processing Toolbox and Neural Network Toolbox.
* Integrate between Image Processing and Neural Network.
* Integrating MATLAB and microcontroller 8051.
* Compare the various techniques at hand with the proposed system.

**1.7 Image processing**

**1.7.1 Image Pre-processing**

The only inputs to the system were the images of vehicles captured by a digital camera. The captured images were taken from approximately 3-5 meters away from the front of the vehicle so that the number plates were clearly visible in the view. Since there are pattern recognition problems arising due to poor image quality caused by varying ambient lighting conditions, number plates are often difficult to detect accurately and efficiently in real situations. Based on the pilot tests carried out with several test runs, for fixed camera, we must maintain car distance constant from fixed camera.

**1.7.2 Image Resize**

Interpolation is basic tool used extensively in task such as zooming, shrinking, rotating and zoometric correction. Interpolation mainly using here for image resize (zooming and shrinking), which is basically image re-sampling methods. Interpolation is the process of using known data to estimate values at unknown locations.

Different types of interpolation methods

* **Nearest neighbour interpolation**

It is called nearest neighbor interpolation because it assigns to each new location the intensity of its nearest neighbor in the original image.

* **Bilinear interpolation**

Bilinear interpolation is in which we use the four nearest neighbors to estimate the intensity at a given location.

* **Bi-cubic interpolation**

Bi-cubic interpolation is in which it involves the sixteen nearest neighbors of a point. Thus for doing image resize bi-cubic interpolation method is used.

**1.7.3 Image Cropping**

Cropping the number plate from car image .by trial error method taking some approximate pixels for all four co-ordinate values where number plate located using that pixel.

**1.7.4 Color Image Processing**

Color is a power full descriptor that often simplifies object identification and extraction from scene. And humans can discern thousands of color shades and intensities, compared to about only two dozen shades of gray. Second factor is particularly important in manual image analysis. Basically, the colors that humans and some other animals perceive in an object are determined by the nature of the light reflected from object. Visible light is composed of a relatively narrow band of frequencies in the electromagnetic spectrum. Cones are the sensors in the responsible for color vision. The 6 to 7 million cones in the human eye can divided in to three principal sensing categories, corresponding roughly to red, green and blue . Approximately 65% of all cones are sensitive to red light,33% cones are sensitive to green light and 2% cones are sensitive to blue light ,due to this colours are seen as variable combination of the so called “primary colours red(R),green(G),and blue(B).

An RGB image, sometimes referred to as a true color image, is stored as an m-by-n-by-3 data array that defines red, green, and blue color components for each individual pixel. The color of each pixel is determined by the combination of the red, green, and blue intensities stored in each color plane at the pixel's location. Graphics file formats store RGB images as 24-bit images, where the red, green, and blue components are 8 bits each. This yields a potential of 16 million colors. The precision with which a real-life image can be replicated has led to the nickname "true color image".

M-by-n-by-3 array of class uint8, uint16, single, or double whose pixel values specify intensity values. For single or double arrays, values range from [0, 1]. For uint8, values range from [0, 255]. For uint16, values range from [0, 65535].

Gray scale =0.2989 \* R + 0.5870 \* G + 0.1140 \* B

**1.7.5 Gray Scale Images**

A grayscale image (also called gray-scale, gray scale, or gray-level) is a data matrix whose values represent intensities within some range. MATLAB stores a grayscale image as a individual matrix, with each element of the matrix corresponding to one image pixel.

Array of class uint8, uint16, int16, single, or double whose pixel values specify intensity values. For single or double arrays, values range from [0, 1]. For uint8, values range from [0,255]. For uint16, values range from [0, 65535]. For int16, values range from [-32768, 32767].

Graythresh(I) computes a global threshold (level) that can be used to convert an intensity image to a binary image with im2bw. level is a normalized intensity value that lies in the range [0, 1].The graythresh function uses Otsu's method, which chooses the threshold to minimize the intraclass variance of the black and white pixels.

Im2bw(I, level) converts the grayscale image I to a binary image. The output image BW replaces all pixels in the input image with luminance greater than level with the value 1 (white) and replaces all other pixels with the value 0 (black). You specify level in the range [0, 1], regardless of the class of the input image.

**1.7.6 Binary Images**

In a binary image, each pixel assumes one of only two discrete values: 1 or 0. A binary image is stored as a logical array. Binary images are easily to process further processing like removing noise, filtering, segmentation, recognition etc.

**1.7.7 Intensity Transformation**

Intensity transformation operates on single pixels of an image, principally for the purpose of contrast manipulation and image threshold. Intensity transformation is among the simplest of all image processing techniques. The values of pixels, before and after processing, will be denoted by r and s respectively. s=T(r), where T is a transformation that maps a pixel value r in to a pixel values.

**Three basic Intensity Transformations:**

* **Linear (negative and identity transformation).**

The negative of an image with intensity levels in the range [0,L-1] is obtained by using the negative transformation

s=L-1-r

Reversing the intensity levels of an image in this manner produces the equivalent of a photographic negative. By binary image we can use this formula reversing the image

s=1-r

* **Logarithmic (log and inverse log transformation).**

The general form of the log transformation

s=c log (1+r)

Where c is a constant, and it is assumed that r ≥ 0 .The narrow range of low intensity values in put in to a wider range of output levels and higher range values become narrow. We use a transformation of this type to expand the values of dark pixels in an image while compressing the higher –level values. The opposite is true of the inverse log transformation.

* **Power –law (nth power and nth  root transformation).**

Power-Law transformation has the basic form

s=crγ

where c and γ are positive constants. Plot of s and r for various values of γ. power law curves with fractional values of γ map a narrow range of dark input values in wider range of output values, with opposite being true for higher values of input levels the curves generated with values of γ>1 have exactly the opposite effect as those generated with values of γ<1.c= γ=1 we getting identity transformation. The identification of objects within an image can be a very difficult task. One way to simplify the problem is to change the gray scale image into a **binary image**, in which each pixel is restricted to a value of either 0 or 1. The techniques used on these binary images go by such names as: **blob analysis**, **connectivity analysis**, and **morphological image processing**. The foundation of morphological processing is in the mathematically rigorous field of set theory; however, this level of sophistication is seldom needed. Most morphological algorithms are simple logic operation.

**1.7.8 Pixel Connectivity**

Morphological processing starts at the peaks in the marker image and spread throughout the rest of the image based on the connectivity of the pixels. Connectivity defines which pixels are connected to other pixels. A set of pixels in a binary image that form a connected group is called an object or a connected component.

Estimate the area of all of the pixels in an image by summing the areas of each pixel in the image. The area of an individual pixel is determined by looking at its 2-by-2 neighborhood. There are six different patterns, each representing a different area:

* Patterns with zero on pixels (area = 0)
* Patterns with one on pixel (area = 1/4)
* Patterns with two adjacent on pixels (area = 1/2)
* Patterns with two diagonal on pixels (area = 3/4)
* Patterns with three on pixels (area = 7/8)
* Patterns with all four on pixels (area = 1)

Each pixel is part of four different 2-by-2 neighborhoods. This means, for example, that a single on pixel surrounded by off pixels has a total area of 1.

**1.8 Neural Network Introduction**

Neural networks are composed of simple elements operating in parallel. These elements are inspired by biological nervous systems. As in nature, the connections between elements largely determine the network function. Train a neural network to perform a particular function by adjusting the values of the connections (weights) between elements. Typically, neural networks are adjusted, or trained, so that a particular input leads to a specific target output. There, the network is adjusted, based on a comparison of the output and the target, until the network output matches the target. Typically, many such input/target pairs are needed to train a network. Neural networks have been trained to perform complex functions in various fields, including pattern recognition, identification, classification, and speech, vision, and control systems.

Neural Network

Including Connections

(Called Weights)

Between neurons

Input

**Adjust Weights**

Target

Compare

Output

**Figure 4.1 Neural Network Training Diagram**

**1.9 Microcontroller introduction**

A micro-controller can be compared to a small stand alone computer, it is a very powerful device, which is capable of executing a series of pre-programmed tasks and interacting with other hardware devices. Being packed in a tiny integrated circuit (IC) whose size and weight is usually negligible, it is becoming the perfect controller for robots or any machines requiring some kind of intelligent automation. A single microcontroller can be sufficient to control a small mobile robot, an automatic washer machine or a security system. Any microcontroller contains a memory to store the program to be executed, and a number of input/output lines that can be used to interact with other devices, like reading the state of a sensor or controlling a motor.

Nowadays, microcontrollers are so cheap and easily available that it is common to use them instead of simple logic circuits like counters for the sole purpose of gaining some design flexibility and saving some space. Some machines and robots will even rely on a multitude of microcontrollers, each one dedicated to a certain task. Most recent microcontrollers are **'In System Programmable'**, meaning that we can modify the program being executed, without removing the microcontroller from its place.

**1.10 Max 232**

Serial RS-232 communication works with voltages “1” represent -3 to -25 and “0” represent +3 to +25, making -3 to 3 undefined. For this reason .to connect any RS-232 to microcontroller system. MAX 232 to convert the TTL logic level to the RS 232 voltage level, and vice versa. MAX232 IC chips are commonly referred to as line drivers.

**1.11 LCD Display**

LCD or liquid Crystal Display has different types of LCD available. In this project 16x2 lines and 5x7 matrix is uesd. LCD contains logic analyzer to controlling display and also communication with microcontroller. It contains 16 pins for different purpose like data transmission controlling display.

**1.12 Introduction ULN 2803**

Since the digital outputs of the some circuits cannot sink much current, they are not capable of driving relays directly. So, high-voltage high-current Darlington arrays are designed for interfacing low-level logic circuitry and multiple peripheral power loads. The series ULN2803 ICs can drive 8 stepper motor At an appropriate duty cycle depending on ambient temperature and number of drivers turned ON simultaneously, typical power loads total over 260W [400mA x 7, 95V] can be controlled. Typical loads include relays, solenoids, stepping motors, magnetic print hammers, multiplexed LED and incandescent displays, and heaters. These Darlington arrays are furnished in 16-pin dual in-line plastic packages (suffix A) and 16-lead surface-mountable SOICs (suffix L). All devices are pinned with outputs opposite inputs to facilitate ease of circuit board layout.

**1.13 Stepper Motor**

Of all motors, step motor is the easiest to control. Its handling simplicity is really hard to deny - all there is to do is to bring the sequence of rectangle impulses to one input of step controller and direction information to another input. Direction information is very simple and comes down to "left" for logical one on that pin and "right" for logical zero. Motor control is also very simple - every impulse makes the motor operating for one step and if there is no impulse the motor won't start. Pause between impulses can be shorter or longer and it defines revolution rate. This rate cannot be infinite because the motor won't be able to "catch up" with all the impulses.

The key to driving a stepper is realizing how the motor is constructed. A diagram shows the representation of a 4 coil motor, so named because 4 coils are used to cause the revolution of the drive shaft. Each coil must be energized in the correct order for the motor to spin. Responsible for providing the control signals to open and close the switches at the appropriate times in order to spin the motors. The control unit is commonly a computer or programmable interface controller, with software directly generating the outputs needed to control the switches. As with drive circuitry for variable reluctance motors, the inductive kick produced when each of these switches is turned off.

**Step angle**

It is angle through which motor shaft rotates in one step. The step angle is different for different motors. Selection of motor according to step angle depends on the application, simply if we require small increments in rotation choose motor having smaller step angle.

No of steps require to rotate one complete rotation = 360 deg. / step angle in deg.

**Steps/second**

The relation between RPM and steps per sec. is given by,

Steps or impulses /sec. =(RPM X Steps /revolution ) /60

Pause between impulses can be shorter or longer and it defines revolution rate. This rate cannot be infinite because the motor won't be able to "catch up" with all the impulses (documentation on specific motor should contain such information). So referring to RPM value in datasheet we can calculate steps/sec and from it delay or pause between impulses.

**2. LITERATURE SURVEY**

**2.1 Image Acquisition**

Image Acquisition is the first step in an NPR system and there are a number of ways to acquire images, the current literature discusses different image acquisition methods used by various authors.

Referring to bibliography 13 in which Naito et. al. developed a sensing system, which uses two CCDs (Charge Coupled Devices) and a prism to split an incident ray into two lights with different intensities. The main feature of this sensing system is that it covers wide illumination conditions from twilight to noon under sunshine, and this system is capable of capturing images of fast moving vehicles without blurring. Referring to the Yan et. al. used an image acquisition card that converts video signals to digital images based on some hardware-based image pre-processing.

Referring to bibliography 8 where Comelli et. al. used a TV camera and a frame grabber card to acquire the image for the developed vehicle NPR system. Salgado et. al. used a Sensor subsystem having a high resolution CCD camera supplemented with a number of new digital operation capabilities. Kim et. al. uses a video camera to acquire the image.

**2.2 Number plate extraction**

Number plate extraction is the most important phase in an NPR system. This section discusses some of the previous work done during the extraction phase.

Referring to bibliography 14 Hontani et. al. proposed a method for extracting characters without prior knowledge of their position and size in the image. The technique is based on scale shape analysis, which in turn is based on the assumption that, characters have line-type shapes locally and blob-type shapes globally.

**2.3 Segmentation**

Referring to bibliography 10 in which Nieuwoudt et. al. used region growing for segmentation of characters. The basic idea behind region growing is to identify one or more criteria that are characteristic for the desired region. After establishing the criteria, the image is searched for any pixels that fulfill the requirements. Whenever such a pixel is encountered, its neighbors are checked, and if any of the neighbors also match the criteria, both the pixels are considered as belonging to the same region.

Referring to bibliography 5 Morel et. al. used partial differential equations (PDE) based technique, Neural network and fuzzy logic were adopted in for segmentation into individual characters.

**2.4 Recognition**

This section presents the methods that were used to classify and then recognize the individual characters. The classification is based on the extracted features. These features are then classified using either the statistical, syntactic or neural approaches. Some of the previous work in the classification and recognition of characters is as follows.

Hasen et. al. discusses a statistical pattern recognition approach for recognition but their technique found to be inefficient. This approach is based on the probabilistic model and uses statistical pattern recognition approach.

Referring to bibliography 13 in which Mei Yu et. al. and Naito et. al. used template matching. Template matching involves the use of a database of characters or templates. There is a separate template for each possible input character. Recognition is achieved by comparing the current input character to each of template in order to find the one which matches the best. If I(x ,y) is the input character, T(x ,y) is template n, then the matching function s(l , Tn) will return a value indicating how well template n matches the input.

Zemike moments have also been used by several authors for recognition of characters. Using zernike moments both the rotation variant and rotation invariant features can be extracted. These features then uses neural network for the recognition phase. Neural network accepts any set of distinguishable features of a pattern as input. It then trains the network using the input data and the training algorithms to recognize the input pattern.

**2.5 Neural Network Training Styles**

Two different styles of training are:

* In incremental training the weights and biases of the network are updated each time an input is presented to the network.
* In batch training the weights and biases are only updated after all the inputs are presented.

**2.5.1 Incremental Training**

Referring to bibliography 14 Incremental training can be applied to both static and dynamic networks, although it is more commonly used with dynamic networks, such as adaptive filters.

**Incremental Training with Static Networks**  To train it incrementally, so that the weights and biases are updated after each input is presented. The function adapt and the inputs and targets are presented as sequences. The inputs are presented as a cell array of sequential vectors, then the weights are updated as each input is presented (incremental mode)

**Incremental Training with Dynamic Networks**

It can be also train dynamic networks incrementally here also used adapt functions.

**2.5.2 Batch Training**

Batch training, in which weights and biases are only updated after all the inputs and targets are presented, can be applied to both static and dynamic networks.

**Batch Training with Static Networks**

Referring to bibliography 14 batch training can be done using either adapt or train, although train is generally the best option, because it typically has access to more efficient training algorithms. Incremental training can only be done with adapt; train can only perform batch training. For batch training of a static network with adapt, the input vectors must be placed in one matrix of concurrent vectors.

**Batch Training with Dynamic Networks**

Training static networks is relatively straightforward. train function used to network is trained in batch mode and the inputs are converted to concurrent vectors (columns of a matrix), even if they are originally passed as a sequence (elements of a cell array).It adapt function , the format of the input determines the method of training. If the inputs are passed as a sequence, then the network is trained in incremental mode. If the inputs are passed as concurrent vectors, then batch mode training is used. With dynamic networks, batch mode training is typically done with train only, especially if only one training sequence exists.

**2.6 Probabilistic Neural Networks**

Referring to bibliography 14 Probabilistic neural networks can be used for classification problems. When an input is presented, the first layer computes distances from the input vector to the training input vectors and produces a vector whose elements indicate how close the input is to a training input. The second layer sums these contributions for each class of inputs to produce as its net output a vector of probabilities. Finally, a compete transfer function on the output of the second layer picks the maximum of these probabilities, and produces a 1 for that class and a 0 for the other classes. The architecture for this system is shown below.

P

**:**

**:**

**:**

**:**

PR

||dist||

b1=1

**^**

LW 2,1

C

Q x 1

Rx1

**IW 1,1**

Q X R

R

Q X 1

-

**+**

n 1

a1

Qx1

Q

Qx1

KxQ

n 2

Kx1

a2

K

Input

Radial Basis Layer

Compet layer

**Figure 4.14 Probabilistic Neural Networks Block Diagram**

R=Number of elements in input vector

Q=Number of input/target pairs =Number of neurons layer 1

K=Number of classes of input vector = Number of neurons layer 2

**Radial basis layer output**

a 1 i= radbas(|| iIW 1,1-P||b1 i )

**Compet layer output**

a2 i=compet( LW 2,1 a1i)

Referring to bibliography 14 it is assumed that there are **Q** input vector/target vector pairs. Each target vector has K elements. One of these elements is **1** and the rest are **0**. Thus, each input vector is associated with one of K classes.

The first-layer input weights, **IW1,1 (**net .IW{1,1}**),** are set to the transpose of the matrix formed from the Q training pairs, **P**'. When an input is presented, the ||dist|| box produces a vector whose elements indicate how close the input is to the vectors of the training set. These elements are multiplied, element by element, by the bias and sent to the **radbas** transfer function. An input vector close to a training vector is represented by a number close to **1** in the output vector **a**1. If an input is close to several training vectors of a single class, it is represented by several elements of **a**1 that are close to **1**.

The second-layer weights, **LW1,2 (**net .LW{2,1}**),** are set to the matrix **T** of target vectors. Each vector has a **1** only in the row associated with that particular class of input, and **0's** elsewhere. The multiplication **Ta**1 sums the elements of **a**1 due to each of the K input classes. Finally, the second-layer transfer function, **compete**, produces a **1** corresponding to the largest element of **n**2, and **0's** elsewhere. Thus, the network classifies the input vector into a specific K class because that class has the maximum probability of being correct.

**3. MODEL OR BLOCK DIAGRAM**

**3.1 Project Block Diagram**

The image captured is stored in database. The image read from the database is converted to standard size; noise present in the image is removed. Then the preprocessed image is given as input to segmentation. Then find the height and width of each character of the image. After this it is given as input to neural network, the character recognized is sent as input to microcontroller. The Microcontroller receives data from the PC, and sends it to LCD, for displaying. After the Character is displayed on LCD, Microcontroller rotates stepper motor 90 degrees clock wise and anticlockwise direction.

**TRAIN BY**

**NUMBER PLATE IMAGE AS THE INPUT**

**COMPUTER PROCESSING USING MATLAB OUTPUT KA 04 ME 6152**

**COMPARE WITH IMAGE DATABASE TO AUTHENTICATEU**

**MUC**

**8051**

**KA 04 ME 6152**

**LCD NUMBER PLATE DISPLAY**

**ULN**

**2803**

**STEPPER MOTOR ROTATES TO OPEN TOLLGATE**

The explanation for individual parts is given below:

* **Number plate image**

The image of the number plate to be identified is given as input to the computer for processing. The image is taken by a high resolution camera of around 10.1 Mega pixels. The image may be in the form of .jpg, .png, .gif, .jpge, .bmp which is to be processed by the computer.

* **Computer processing by matlab**

The image given as input is processed by matlab software for various fundamental image processing steps that includes acquisition which accepts the high resolution input image and converts it into matrix form. Image pre processing helps us to improve the image in ways that increase the chances for success of the other processes. Image segmentation helps to partitions an input image into its constituent parts or objects. Image representation helps to convert the input data to a form suitable for computer processing. Image description helps us to extract features that result in some quantitative information of interest or features that are basic for differentiating one class of objects from another. Image recognition helps us to assign a label to an object based on the information provided by its descriptors. Image interpretation helps us to assign meaning to an ensemble of recognized objects.

* **Training images by neural network**

The recognized characters in the number plate are stored as separate images by considering 90 x 40 images. The characters are separated and the separated characters are arranged in their respective databases, for e.g. all k’s are grouped in a separate ‘K’ database. Thereafter the neural network code is trained so as to recognize those characters in the real world applications.

* **Compare with image database to authenticate**

The number plate recognized has ten characters which are compared with number plate in database and only those in the database are authenticated.

* **Microcontroller 8051**

The 8051 is used to interface stepper motor and lcd for display and tollgate operation. It is also used for serial communication interface between the system and 8051 via max232.

* **LCD display**

The lcd is used to display the number plate and also whether the number plate is “valid” or “not valid” and accordingly the gate opened message is displayed.

* **ULN 2803**

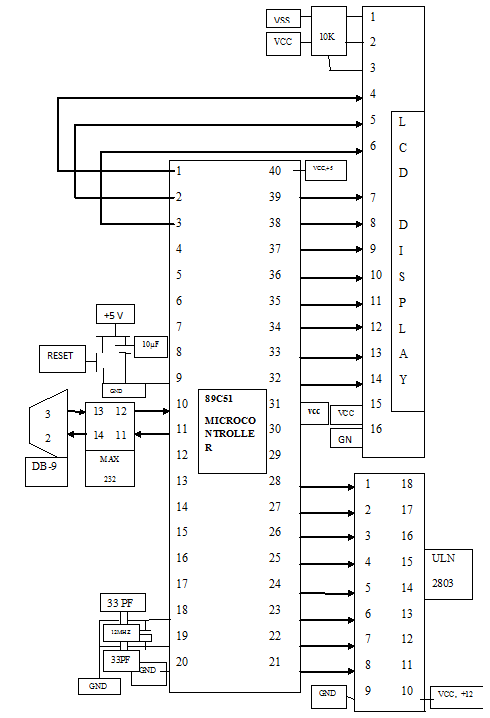
This is the driver used to run the stepper motor by interfacing it with 8051 and the stepper motor.

* **Stepper motor**

The stepper motor is rotated by 90 degrees clockwise to open the toll gate when the number plate is valid.

**3.2 Block Diagram for microcontroller section**

* Designing of Microcontroller part is shown with the connecting diagram. Microcontroller used is Atmel 8051.
* It is a 40 pin IC. It requires 5V power supply. Port 0 is connected to LCD of DB0 – DB7 and in pin P1.0 is connected to pin no. 4 of LCD display.
* Pin P1.1 is connected to pin no. 5 of LCD display and similarly in pin P1.2 is connected to pin no. 6 of LCD display.
* Microcontroller port P2 is is connected to ULN 2803 from pin no.1 to pin no.8. Microcontroller pin No. 10 and 11 are connected to MAX 232 with pin no.12 and 11 respectively.
* Microcontroller pin no. 18 and 19 is connected to Crystal oscillator with a frequency of 11.0952MHz. Pin no.31 of microcontroller is connected to VCC because internal Flash memory is used.
* Max 232 converts TTL to RS232 level and ULN 2803 using for driver for Stepper Motor.
* The block diagram for microcontroller section is shown below:-

****

**4. DESIGN AND IMPLEMENTATION**

**4.1 Project Flow chart for image processing section:**

Start

Image capturing

Images Stored in database

Images read from data base

RGB to gray conversion

Image cropping

Image resize

A

Gray to binary conversion

Pre –processed image

Inversing of Image

Inverse of Image

Getting ROI part

Removing noise

Image filling

Finding the width and height if each character

Extracting region of objects

Resizing of extracted images

Converting images to column vector then sending I/P Neural Network

B

If image is identified?

Respective character identified and stored in variable **result**

It is consider blank

Stepper motor rotating 90̊ clock wise and 90 ̊ anti clock wise

Sending recognized characters serially to microcontroller

Microcontroller received characters from PC displaying In LCD

Yes

No

B

Stop

**4.2 Program flow Chart for Microcontroller section:**

Initialization of serial communication, baud rate, data bits, start and stop bits will be done at the beginning. Initialize LCD, then display message on LCD. After the characters received from PC, the received characters are displayed on LCD. Then the Microcontroller controls the stepper motor clockwise and anticlockwise direction.

Start

Enable the interrupt serial, into and int1 using IE register.IE=#95H

Initialize of TMOD register Timer 1, mode 2

Set the timer 1

Selecting the data, stop bit using SCON register

Initialize the LCD

X

x

Stepper motor rotates clock wise direction 90̊ and displays In LCD “GATE OPENED”.

Stepper motor rotates anti clock wise direction 90̊ and displays In LCD “GATE CLOSED”.

Clear LCD screen

Clear LCD screen and waiting from next data from PC

End

Writing In LCD “WEL COME TO NUMBER PLATE” message

Clear LCD screen

Waiting for receiving characters from personal computer

If first character received from personal computer displaying first line in LCD “NUMBER PLATE” and second line character received from PC

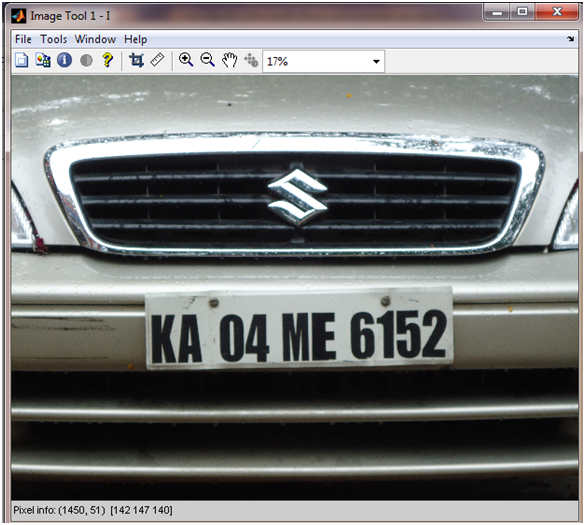
**4.3 Project Implementation steps**

**4.3.1 Project Implementation steps for image processing section**

**Step 1**

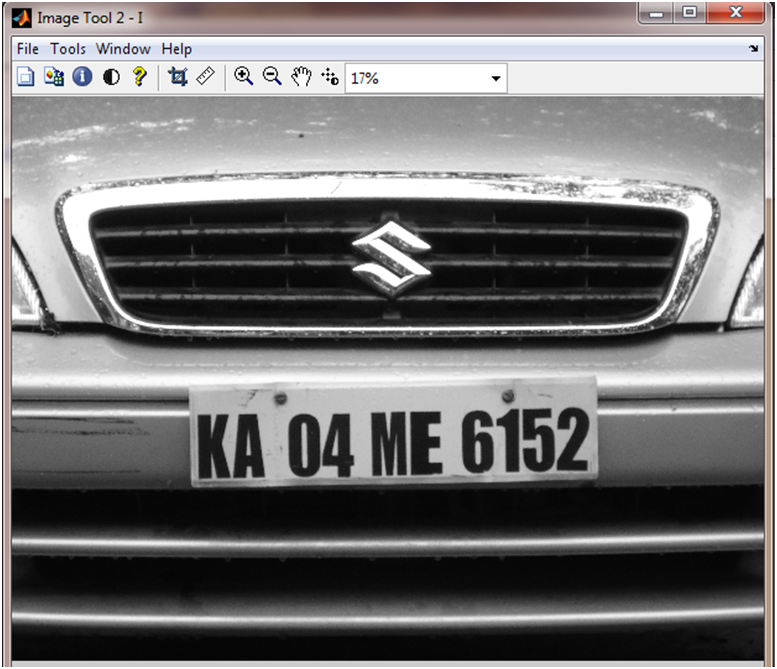
Captured RGB image stored in data base. With size 3648 columns and 2057 rows

Captured Image



**Step 2**

Convert RGB image to Gray scale image using function **rgb2gray**.



RGB to Gray scale image

**Step 3**

Resize the “gray scale” image using function **imresize.**



Image resized 1000Rand 2000C

2000

**Step 4**

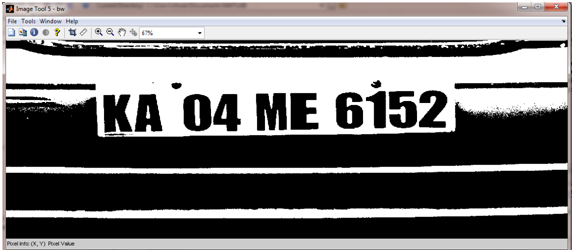
Crop the number plate location image 601 rows and 1701 columns.



Cropped Image 601R and 1701C

**Step 5**

Convert the gray scale image to binary image using function **“im2bw”.**



Gray to binary image

**Step 6**

Image fill by holes, it fills the surround by back ground white pixels using function “imfill”.

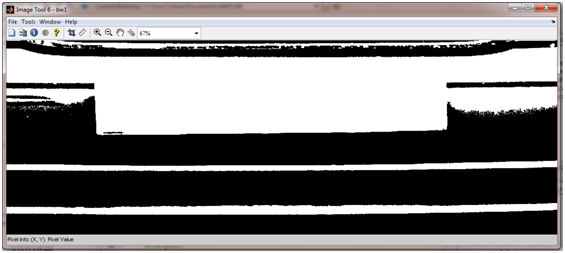
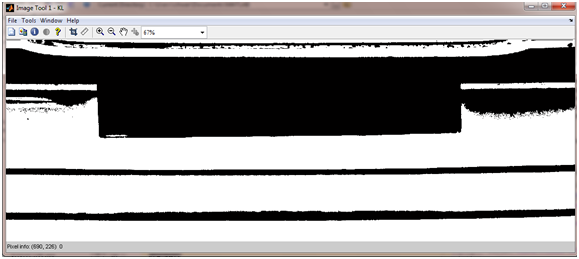


Image fill

**Step 7**

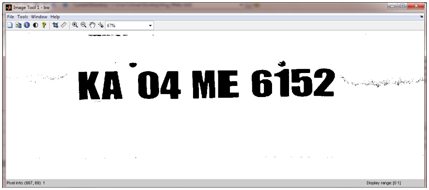
Inversed image from step 6



Inverse image

**Step 8**

Adding the images in step and step 7 to et the final image.



**Step 9**

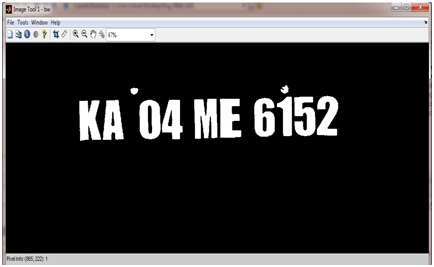
Now inversed the image from step 7 now object become white and background become black.



Inversed image

**Step 10**

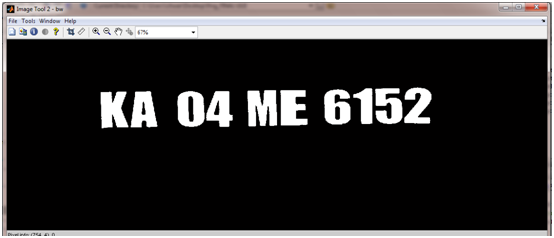
Remove the small connected white pixels using function “bwareaopen”.



Removed small noise

**Step 11**

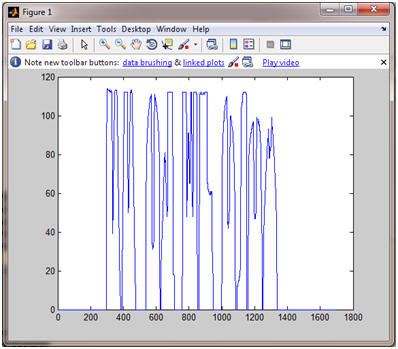
Finally noise removed image, this input image to segmentation.



Pre-processed image

**Step 12**

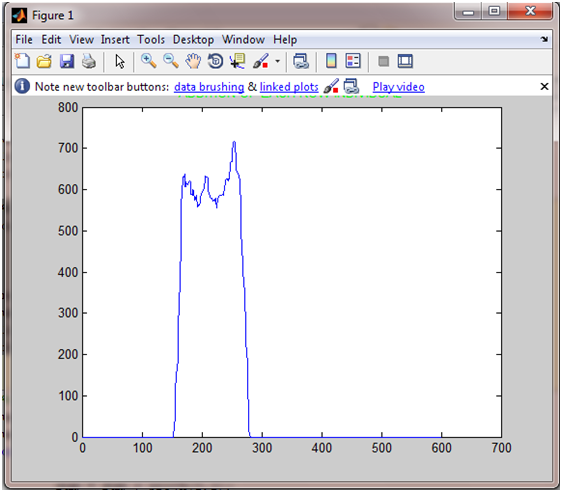
Adding the each column individually, add value available in “projh”.



Sum individually each column

**Step 13**

Adding the each row individually, add value available in “projv”.



Sum individually each row

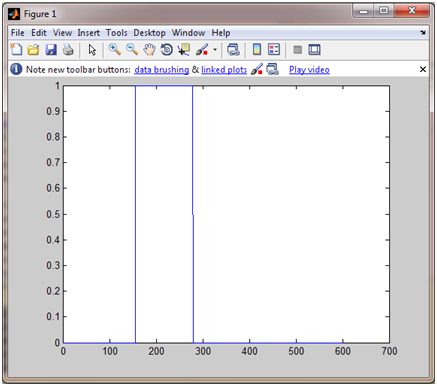
**Step 14**

Find some threshold value.

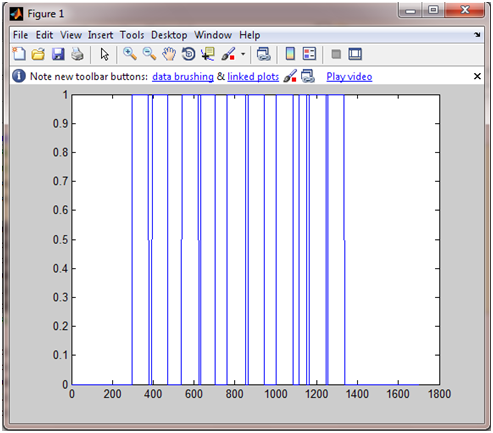
**Step 15**

Plot the vertical and horizontal graphs using threshold value.

Plotting using threshold



Plotting using threshold



**4.3.2 Finding width and height of each character in the image**

**Step 16**

Register the potions where values change from 0 to 1 and from 1 to 0 in xloc (d) are respectively.

**xloc**

**Columns 1 through 12**

**297 377 394 471 539 622 631 702 759 851 863 942**

**Columns 13 through 21**

**1000 1083 1114 1151 1162 1246 1254 1335 1701**

**Xloc**

**Columns 1 through 12**

**297 377 394 471 539 622 631 702 759 851 863 942**

**Columns 13 through 20**

**1000 1083 1114 1151 1162 1246 1254 1335**

**Step 17**

Find the height of each character.

**yloc1**

166 280

165 279

165 277

165 277

**162 275**

161 273

160 270

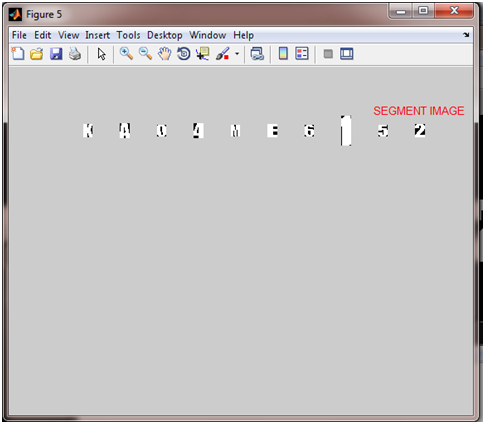
156 268

155 268

154 266

**Step 18**

Dividing height by width if **height by width less <18** it indicates candidate is there, then extracting that region. As shown in the result.



Extracted objects

**Step 19**

Resize all the candidates with 90 rows and 40 columns.



**Step 20**

Converting 90x40 matrix into row vector.

**4.3.3 Project Implementation steps for Character recognition using Neural Network**

**Step 21**

Row is converted in to column giving input to neural network.

**Step 22**

Step 18 and 19 repeat at the end of candidate.

**Step 23**

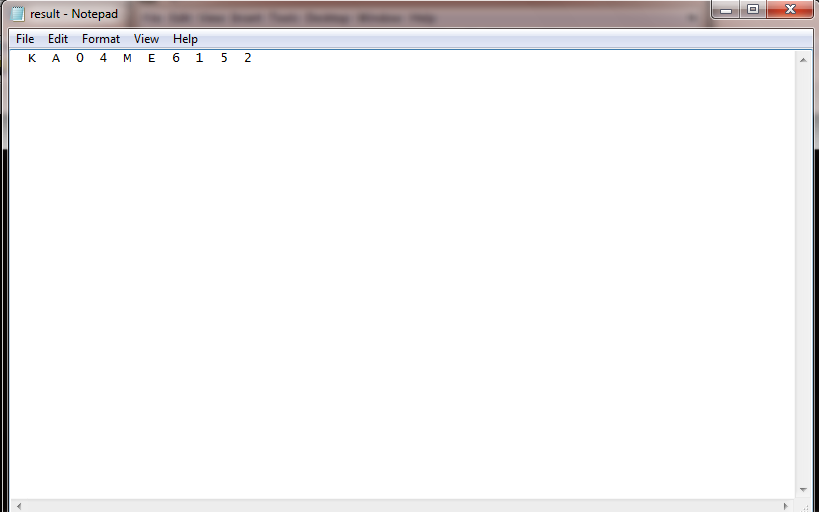
Recognized characters located variable result **and str**

**Step 24**

Send each character serially at the baud rate of 9600.

**Step 25**

Display result in notepad using function “winopen”.



**4.3.4 Recognized Character Displaying in LCD and controlling Stepper Motor using Microcontroller**

**Step 26**

Microcontroller receives all character from computer at the baud rate of 9600.

**Step 27**

Receive all character display in LCD.

**Step 28**

After receiving all the character stepper motor rotates 90 ̊ clock wise direction and then 90 ̊ anti clock wise direction.

**4.4 Microcontroller data flow steps**

**4.4.1 Initialization**

**Step 1**

Enable the interrupt serial, into and int1 using IE register.

**Step 2**

Select timer 1 mode 2 for serial communication using TMOD register.

**Step 3**

Calculating baud rate using following formula and loaded in to TH1 register.

XTAL=11.0592

Baud rate=11.0592 MHZ/12=921.6 KHZ/32=28,800HZ/3=9600

-3 or FD(hex) is loaded in to TH1

**Step 4**

Select of data bits, start and stop bit using SCON register.

**Step 5**

Set the timer TR1 register.

**Step 6**

By sending the commands to LCD display to initialize the LCD 16x2, 5X7 matrix and cursor on

**4.4.2 Displaying Message on LCD**

**Step 7**

Writing data in to the LCD “WEL COME TO NUMBER PLATE”

**Step 8**

Clearing the LCD screen by sending the command #01H.

**Step 9**

Waiting for characters receive from Personal Computer (PC).

**Step 10**

After receiving the first character from PC, display first line of LCD “NUMBER PLATE”. Then in second line displaying all characters receive from PC (Recognized characters).

**Step 11**

Clear the screen of LCD.

**9.4.3 Controlling Stepper Motor**

**Step 12**

Stepper motor rotates clock wise direction 90̊ and displays In LCD “GATE OPENED”.

**Step 13**

Stepper motor rotates anti clock wise direction 90̊ and displaying In LCD “GATE CLOSED”.

**Step 14**

Clear the screen of LCD.

**5. Results and Conclusion**

**5.1 Settings**

Due to the limited availability of data at the time of experiment, the NPR system was tested by using a reduced set of data. A total of 17 images with vehicle license plates taken under different lighting conditions, different distance and different exposure were used as input to the system. The images were originally taken using a 10.1Mega-pixel digital camera and resized to 1000x2000 for quicker processing.

**5.2 Result**

**5.2.1. Plate Localization**

All images were submitted to the plate localization process, candidate regions are marked with a red rectangle after processing. The images are visually inspected to verify its correctness.

**Plate Localization Success Rate**

|  |  |  |  |
| --- | --- | --- | --- |
| **Total Images** | **Plate located** | **Failed to locate** | **Success Rate** |
| **17** | **17** | **0** | **100** |

**5.2.2 Character Segmentation**

Only those images with candidate regions identified are submitted to Character Segmentation. After processing, the plate area is extracted from the input image.

**Character Segmentation Success Rate**

|  |  |  |  |
| --- | --- | --- | --- |
| **Images Character** | **Character Segmented** | **Failed** | **Success rate** |
| **17** | **17** | **0** | **100** |

**5.2.3 Character Recognition**

It recognizes all trained sample but it may be recognize other than trained samples. For this the training of probabilistic Neural Network is necessary.

**5.3 Conclusion**

The process of vehicle number plate recognition requires a very high degree of accuracy when we are working on a very busy road or parking which may not be possible manually as a human being tends to get fatigued due to monotonous nature of the job and they cannot keep track of the vehicles when there are multiple vehicles are passing in a very short time .To overcome this problem, many efforts have been made by the researchers across the globe for last many years. A similar effort has been made in this work to develop an accurate and automatic number plate recognition system. MATLAB has been used to obtain the desired results. The setup has been tested for 17 vehicles containing different number plates. In the process of final evaluation after optimizing the parameters like brightness, contrast and gamma, adjustments, optimum values for lightening and the angle from which the image is to be taken. We get an overall efficiency of 100% for this system. Though this accuracy is not acceptable in general, but still the system can be used for vehicle identification. It can give us a relative advantage of data acquisition and online warning in case of stolen vehicles which is not possible by traditional man handled check posts.

The Neural Network trained with 17 vehicle boards for character recognition. The same 17 vehicle boards have been used for testing the software. It recognized all the boards. However other boards which are not trained could not be recognized, because capturing image different angle, distance and also size of fonts different in number plates. Some of the number plates have not recognized due to localization and segmentation errors.

**5.4 Future work**

From this project it has been clear that it is easy to implement Character Recognition in number of applications. This project is detecting particular number plate position. Further designing an algorithm for detecting number plate in any position of the vehicle is to be done. In this project it is recognizing trained samples, further designing an algorithm for recognition with less number of samples and capturing the number plates with different angles, algorithm has to be developed. So that it gives more accurate and reliable.

**5. Bibliography**

1. Digital image processing by Rafael C.Gonzalez and Richard E. Wood.

2. Neural network tool box and Image processing tool box from MATLAB software.

3. The 8051 microcontroller architecture, programming, and applications by Kenneth J. Ayala.

4. The 8051 Microcontroller and Embedded Systems by Muhammad Ali Mazidi and Janice Gillispie Mazidi.

5. Hu, M. K., "Visual Pattern Recognition by Moment Invariant", IRE Transaction on Information Theory, volume IT- 8, pp. 179-187, 1962. [2] Khotanzad, A., and Hong, Y.H., "Invariant image recognition by zernike moments," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 12, no. 5, pp. 489-497, 1990.

6. Khotanzad, A., and Hong, Y.H., "Rotation in-variant image recognition using features selected via a systematic method," Pattern Recognition, vol. 23, no. 10, pp. 1089-1101, 1990.

7.Lee, E. R., Earn, P. K., and Kim, H. J., "Automatic recognition of a car license plate using color image processing", IEEE International Conference on Image Processing 1994, vol. 2, pp.301-305, 1994.

8. Comelli, P., Ferragina, P., Granieri. M. N., and Stabile, F., "Optical recognition of motor vehicle license plates", IEEE Transactions on Vehicular Technology, vol. 44, no. 4, pp: 790-799, 1995.

9. Morel J., and Solemini S., "Variational Methods in Image Segmentation", Birkhauser, Boston, 1995.

10. Nieuwoudt, C, and van Heerden, R., "Automatic number plate segmentation and recognition", Seventh annual South African workshop on Pattern Recognition, pp. 88-93, IAPR, 1996.

11. Kim, G. M., "The automatic recognition of the plate of vehicle using the correlation coefficient and Hough transform", Journal of Control, Automation and System Engineering, vol. 3, no.5, pp. 511-519, 1997. 75.

12. Cho, D. U., and Cho, Y. Ft., "Implementation of pre-processing independent of

environment and recognition and template matching ", The Journal of the Korean Institute of Communication Sciences, vol. 23, no. 1, pp. 94-100, 1998. [11] Park, S. FL, Kim, K. I., Jung, K., and Kim, H. J., "Locating car license plates using neural network", IEE Electronics Letters, vol.35, no. 17, pp. 1475-1477, 1999.

13. Naito, T. Tsukada, T. Yamada, K. Kozuka, K. and Yamamoto, S., "Robust recognition methods for inclined license plates under various illumination conditions outdoors", Proceedings IEEE/IEEJ/JSAI International Conference on Intelligent Transport Systems.

14. Kim, K. K., Kim, K. I., Kim, J.B., and Kim, H. J., "Learning based approach for license plate recognition", Proceedings of IEEE Processing Society Workshop on Neural Networks for Signal Processing, vol. 2, pp: 614-623, 2000.

15. Yu, M., and Kim, Y. D., "An approach to Korean license plate recognition based on vertical edge matching", IEEE International Conference on Systems, Man, and Cybernetics, vol. 4, pp. 2975-2980, 2000.

16. Efficient Embedded Neural-Network-Based License Plate Recognition System Hakan Caner, H. Selcuk Gecim, and Ali Ziya Alkar, Member, IEEE.

17. License Plate Character Recognition System using Neural Network Anuja P. Nagare Thadomal Shahani Engineering College, University of Mumbai, Mumbai-400 032, India.

18. Recognition of car license plates using neocognitron type of artificial neural network B.Th.M cornet and L.J.M Rothkrantz.

19. “Adaptive Boosting of Neural Networks for Character Recognition” Holger Schwenk Yoshua Bengio Dept. Informatique et Recherche Op´erationnelle Universit´e de Montr´eal, Montreal, Qc H3C-3J7, Canada fschwenk,bengioyg@iro.umontreal.ca May 29, 1997.

20. A neural network based character recognition system for Sinhala script Rohana K. Rajapakse, A. Ruvan Weerasinghe And E. Kevin Seneviratne Department of Statistics and Computer Science, University of Colombo