PROJECT REPORT

ON

"AUTOMATIC NUMBER PLATE RECOGNITION SYSTEM"

Submitted in partial fulfilment of the requirements for **the partial completion of**

PROJECT FOR COMMUNITY SERVICE [16EC7DCPW1]

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

VISVESVARAYA TECHNOLOGICAL UNIVERSITY, BELGAUM

SUBMITTED BY:

Nikhil Anand B	**1BM16EC065**
Prajwal V Holla	**1BM16EC079**
Pranava Y N	**1BM16EC005**
Ranjith D	**1BM16EC087**

Under the Guidance of

Dr. Sudhindra K R

(Associate Professor, ECE, BMSCE)

August - December 2019

Department of Electronics and Communication Engineering

B.M.S COLLEGE OF ENGINEERING

(Autonomous College Affiliated to Visvesvaraya Technological University, Belgaum)

Bull Temple Road, Basavanagudi, Bangalore-560019

DECLARATION

We undersigned students of pre-final semester B.E in Electronics and Communication Engineering, BMS College of Engineering, Bangalore, hereby declare that the dissertation entitled "AUTOMATIC NUMBER PLATE RECOGNITION SYSTEM", embodies the report of our project work carried out independently by us under the guidance of Dr. Sudhindra K R, Associate Professor, E&C Department, BMSCE, Bangalore in partial fulfilment for the award of Bachelor of Engineering in Electronics and Communication from Visvesvaraya Technological University, Belgaum during the academic year 2019-2020.

We also declare that to the best of our knowledge and belief, this project has not been submitted for the award of any other degree on earlier occasion by any student.

Place: Bangalore

Date: 29th November 2019

Nikhil Anand B	1BM16EC065
Prajwal V Holla	1BM16EC079
Pranava Y N	1BM16EC005
Ranjith D	1BM16EC087

B.M.S COLLEGE OF ENGINEERING

(Autonomous College under VTU)

Department of Electronics and Communication Engineering

CERTIFICATE

This is to certify that the project entitled " **AUTOMATIC NUMBER PLATE RECOGNITION SYSTEM"** is a bonafide work carried out by **Nikhil Anand B** (USN:1BM16EC065), **Prajwal V Holla** (USN:1BM16EC079), **Pranava YN** (USN:1BM16EC005) and **Ranjith D** (USN:1BM16EC087) in partial fulfillment for the partial completion of PROJECT FOR COMMUNITY SERVICE [16EC7DCPW1] during the academic year 2019-2020.

Dr. Sudhindra K R	**Dr. Arathi R Shankar**	**Dr. B. V. Ravishankar**
Associate Prof., ECE, BMSCE	HOD, ECE, BMSCE	Principal, BMSCE

External Examination: **Signature with date:**

1.

2.

ABSTRACT

Information technology is rapidly advancing with respect to automated systems. In such systems, people utilize computer based expert systems to analyse and handle real-life problems such as intelligent transportation systems. By observing the current trends and technological advancements in automobile industries, one can easily expect it to boom in the short future. With ever increase in automobiles on road it has indeed become a great task to manage vehicular traffic. Thus there is a need for intervention of technology in traffic management. This trend can already be noticed from the fact that there are smart traffic signals being installed to monitor traffic. Smart ambulance systems are being installed for convenience of ambulances in case of emergency.

With latest advancements in IoT, development of various sensors, camera technologies and the resources for computation available for use, the visual data from these sensor devices(cameras) can now easily collected, stored and analysed. Using Artificial intelligence and machine learning techniques all the analyses of data can be made with less or almost no human intervention and decision making.

In this project we utilise techniques such as image processing, AI, neural networks and cloud data management to detect and recognise vehicle number plates from the camera images.

ACKNOWLEDGEMENT

Any achievement, be it scholastic or otherwise does not depend solely on the individual efforts but on the guidance, encouragement and cooperation of intellectuals, elders and friends. A number of personalities, in their own capacities have helped us in carrying out this project work. We would like to take this opportunity to thank them all.

We express profound gratitude to respected principal **Dr. B. V. Ravishankar,** BMS College of Engineering for providing a congenial environment to work in. Our sincere gratitude to **Dr. Arathi R Shankar,** Head of the Department, Electronics and Communication Engineering for encouraging and providing this opportunity to carry out the project in the department.

We would like to thank our guide **Dr. Sudhindra K R,** Associate Professor, Department of ECE who helped us in all the ways to carry out the project work. He stood beside and guided us in every step.

We thank all our professors for providing the basic knowledge without which this project wouldn't have been possible. Last but not the least we thank our family and friends, who made their valuable support compelled us to maintain a standard throughout our endeavour.

-

Nikhil Anand B

Prajwal V Holla

Pranava YN

Ranjith D

LIST OF FIGURES

LIST OF TABLES

CONTENTS

CHAPTER 1: INTRODUCTION

1.1 Introduction:

To extract the number from vehicle number plate it is necessary to capture the image, process it using Digital image processing, recognise the numbers and characters as well as store data to the cloud.

Digital image processing:

Digital image processing is a subcategory in Digital Signal Processing which deals with the usage of computer algorithms to perform image processing tasks. Digital counterpart is much more effective and reliable than the Analog Image processing as it offers wider variety of algorithms to be utilized. Digital image processing also avoids the problems such as noise build-up and signal distortion.

With the advent of Digital signal processors, processing is being performed in a real-time and other time critical scenario. Thereby Digital image processing has found wide range of applications of which some significant ones are noted below,

- Pattern recognition
- Medical field
- Machine/robot vision
- Remote sensing
- Military equipment etc.

Artificial intelligence:

Artificial intelligence is defined as the study of "intelligent agents": any device that perceives its environment and takes actions that maximize its chance of successfully achieving its goals. A more elaborate definition characterizes AI as “a system’s ability to correctly interpret external data, to learn from such data, and to use those learning to achieve specific goals and tasks through flexible adaptation.”

Machine learning (ML) is the scientific study of algorithms and statistical models that computer systems use to perform a specific task without using explicit instructions, relying on patterns and inference instead. It is seen as a subset of artificial intelligence. Machine learning algorithms build a mathematical model based on sample data, known as "training data", in order to make calculations or decisions without being explicitly programmed to perform the task.

Recently many Machine learning algorithms are being used to create smart and intelligent systems. Machine learning directly utilises statistics to get “trained” to perform certain operations and decision making. It is being called as the Technology of the future.

Neural networks:

A neural network is a series of algorithms that endeavours to recognize underlying relationships in a set of data through a process that imitate the way the human brain operates. In this sense, neural networks refer to systems of neurons, either organic or artificial in nature. Neural networks can adapt to changing input, so the network generates the best possible result without needing to redesign the output criteria. The concept of neural networks, which has its origins in artificial intelligence, is swiftly gaining popularity.

Hidden layers fine-tune the input weightings until the neural network’s margin of error is negligible. It is conjectured that hidden layers extrapolate salient features in the input data that have predictive power regarding the outputs. This describes feature extraction, which accomplishes a utility similar to statistical techniques such as principal component analysis.

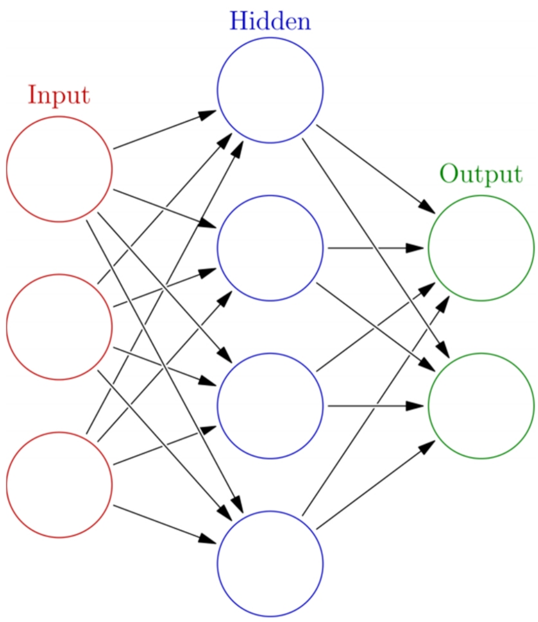


Fig 1: A simple Neural network

Some basic types of Neural networks being used currently are,

- Multilayer perceptron (MLP)
- Convolutional neural network (CNN)
- Recursive neural network (RNN)
- Recurrent neural network (RNN)
- Long short-term memory (LSTM)
- Sequence-to-sequence models
- Shallow neural networks

Any neural network is characterised by the following,

Weights (Parameters):

A weight represents the strength of the connection between units. If the weight from node 1 to node 2 has greater magnitude, it means that neuron 1 has greater influence over neuron 2. A weight brings down the importance of the input value. Weights near zero means changing this input will not change the output. Negative weights means increasing this input will decrease the output. A weight decides how much influence the input will have on the output.

Input Layer—This is the first layer in the neural network. It takes input values and passes them on to the next layer. It doesn't apply any operations on the input values & has no weights and biases values associated.

Hidden Layers:

Hidden layers have nodes which apply different transformations to the input data. One hidden layer is a collection of neurons arranged vertically. Last hidden layer passes on values to the output layer. All the neurons in a hidden layer are connected to each and every neuron in the next layer; hence we have fully linked hidden layers.

Output Layer:

This layer is the last layer in the network & receives input from the last hidden layer. With this layer we can get anticipated number of values and in a desired range.

Training of Neural Networks:

Training a neural network involves figuring out the weights of that network such that the defined cost function comes down to a minimum. Many iterative algorithms are used to train a neural network.

Some of them are,

- Gradient descent
- Newton method
- Conjugate gradient
- Quasi-Newton method
- Levenberg-Marquardt algorithm

Gradient descent is most widely used among them.

Gradient descent algorithm:

Gradient descent is a first-order iterative optimization algorithm for finding the local minimum of a function. To find a local minimum of a function using gradient descent, one takes steps proportional to the negative of the gradient (or approximate gradient) of the function at the current point. If, instead, one takes steps proportional to the positive of the gradient, one approaches a local maximum of that function; the procedure is then known as gradient ascent. Gradient descent was originally proposed by Cauchy in 1847.

Gradient descent is also known as steepest descent. However, gradient descent should not be confused with the method of steepest descent for approximating integrals.

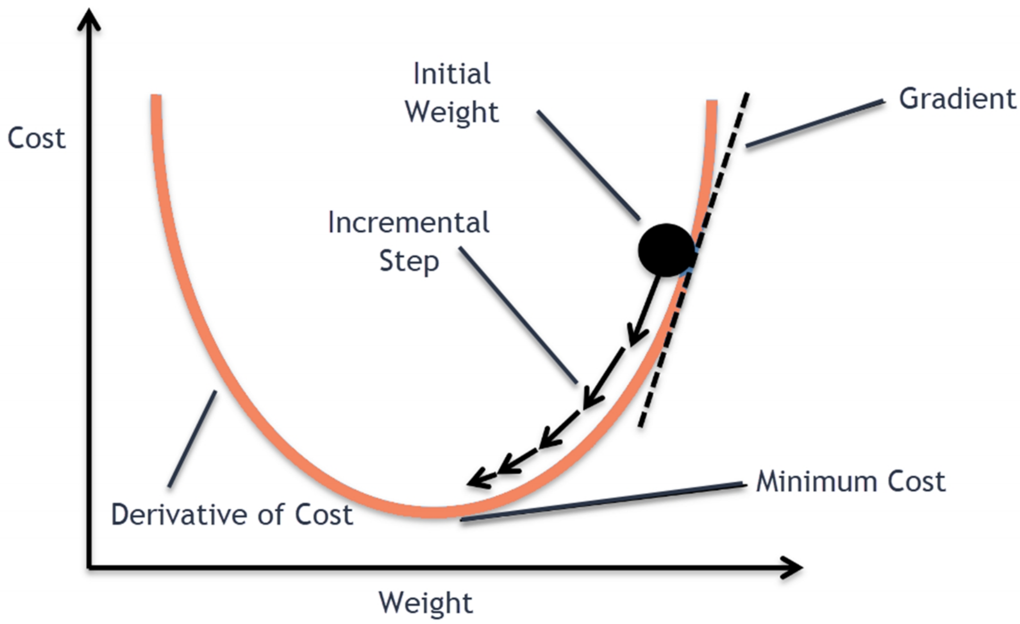


Fig 2.1: Illustration of working of Gradient descent algorithm

Types of gradient Descent:

- *Batch Gradient Descent:*

  This is a type of gradient descent which processes all the training examples for each iteration of gradient descent. But if the number of training examples is large, then batch gradient descent is computationally very expensive. Hence if the number of training examples is large, then batch gradient descent is not preferred. Instead, we prefer to use stochastic gradient descent or mini-batch gradient descent.

- *Stochastic Gradient Descent:*

  This is a type of gradient descent which processes 1 training example per iteration. Hence, the parameters are being updated even after one iteration in which only a single example has been processed. Hence this is quite faster than batch gradient descent. But again, when the number of training examples is large, even then it processes only one example which can be additional overhead for the system as the number of iterations will be quite large.

- *Mini Batch gradient descent:*

  This is a type of gradient descent which works faster than both batch gradient descent and stochastic gradient descent. Here b examples where b<m are processed per iteration. So even if the number of training examples is large, it is processed in batches of b training examples in one go. Thus, it works for larger training examples and that too with lesser number of iterations.

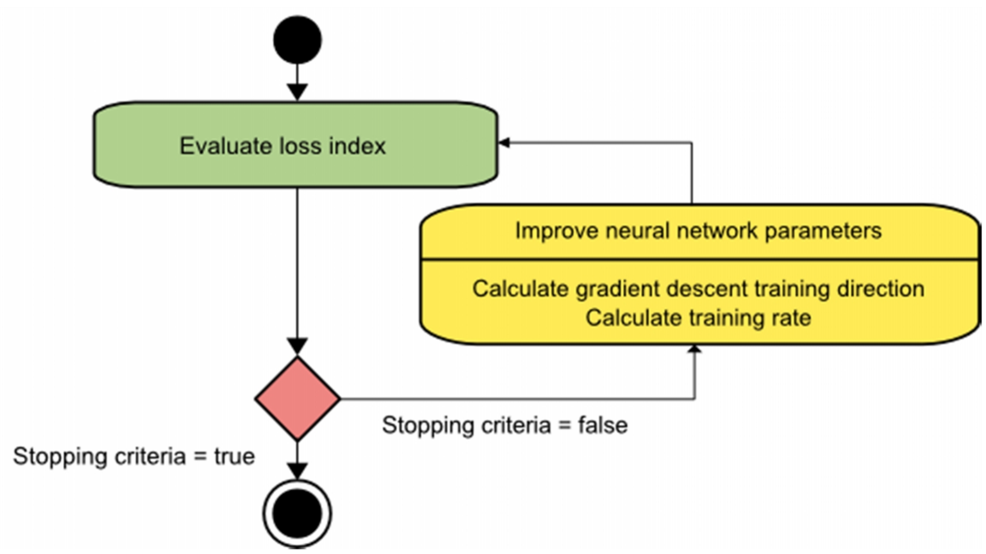


Figure 2.2: Gradient descent algorithm

Cloud data storage and computing:

Cloud storage is a model of computer data storage in which the digital data is stored in logical pools. The physical storage involves multiple servers (sometimes in multiple locations), and the physical environment is typically owned and managed by a hosting company. These cloud storage providers are responsible for keeping the data available and accessible, and the physical environment protected and running. People and organizations buy or lease storage capacity from the providers to store user, organization, or application data. Security and accessibility are two major concerns in using cloud storages.

Some widely used cloud storages are,

- Google Drive
- OneDrive
- DropBox
- ThingSpeak cloud for Internet of Things etc.

Character recognition and extraction from an image has always been a challenging task. Many resource consuming and tedious algorithms which use image processing techniques have been developed for this job but most of them fail to deliver at some circumstances thereby paving a way for utilization of modern analysis techniques such as Machine learning and AI.

1.2 Problem definition:

With an increase in population there is an increase in traffic density which begs the need for a more efficient system for the recognition of traffic violations. There is also a need to reduce manual intervention in traffic management so as to allow seamless traffic flow.

1.3 Objective of the project:

The main objective of the project is to reduce manual intervention in vehicle traffic management as well as recognition of traffic rules violation such jumping signal lights, over speeding of vehicles etc. This requirement of traffic police cop manually being present there to recognise the vehicle can be reduced by automatic detection and recognition of number plates.

CHAPTER 2: LITERATURE SURVEY

In the paper [1], authors provide an efficient approach for number plate extraction from vehicles image under image processing. Authors of the paper present a proficient approach for number plate extraction from pre-processed vehicle input image using morphological operations, thresholding, sobel vertical edge recognition and connected component analysis. The input image is firstly pre-processed using iterative bilateral filter and adaptive histogram equalization. They use several image processing techniques to detect the part of image which is a number plate and extract various features to recognise the characters. The success rate as claimed by this paper is around 90 percentages.

Paper [2] describes the Smart Vehicle Screening System, which can be installed into a tollbooth for automated recognition of vehicle license plate information using a photograph of a vehicle. There are considered an approach to identify vehicle through recognizing of it license plate using image fusion, neural networks and threshold techniques as well as some experimental results to recognize the license plate successfully.

Papers [3] [4] and [5] use many variants of image processing algorithms integrated with automated learning algorithms for the effective recognition of number plate. Author in paper [3] in particular recognises the difficulties faced in plate detections of Indian vehicles as they are very versatile and of different sizes. The success rate of recognition in almost all papers is around 90 - 95 percentage.

Book [6] provides a very knowledgeable insight on various techniques and algorithms used for digital image processing.

Books [7] and [8] help in learning the fundamentals of neural networks and machine learning.

Paper [9] provides the description of the dataset used to train the neural network.

CHAPTER 3: METHODOLOGY AND IMPLEMENTATION

3.1 Block diagram:

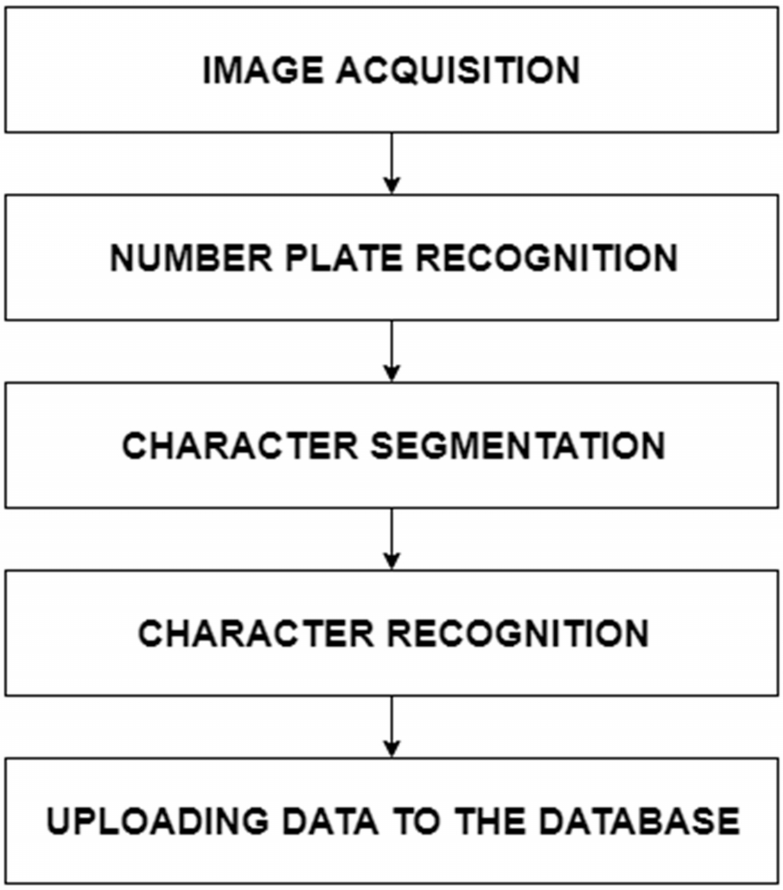


Fig 3: Block diagram of the plate recognition system

3.2 Project flow:

The project consists of 5 major subparts as represented in the block diagram (Fig 3.)

Image acquisition:

Number plate information in the project is extracted from the image data. Therefore as a first step a high resolution image is captured. This image is further analysed and number is extracted.

Number plate recognition:

Next step is to recognise the area of image that corresponds to a number plate. This is achieved using Image processing techniques such as contour recognition, image cleansing and de-noising, smoothening etc. This step is further discussed in detail in Section 3.3.

Character segmentation:

Once the image of desired number plate is extracted from the image, it is required to recognize the area that corresponds to individual characters. Even this part of the system is realised using image processing algorithms.

Character recognition:

In this step each sub-image (also called as Region of Interest) is analysed and the character in it is recognised. This is achieved using Neural networks and machine learning. A neural network pre-trained with the required dataset is used to classify the characters present in the image. Detailed process of training, samples of dataset used is presented in Section 3.3.

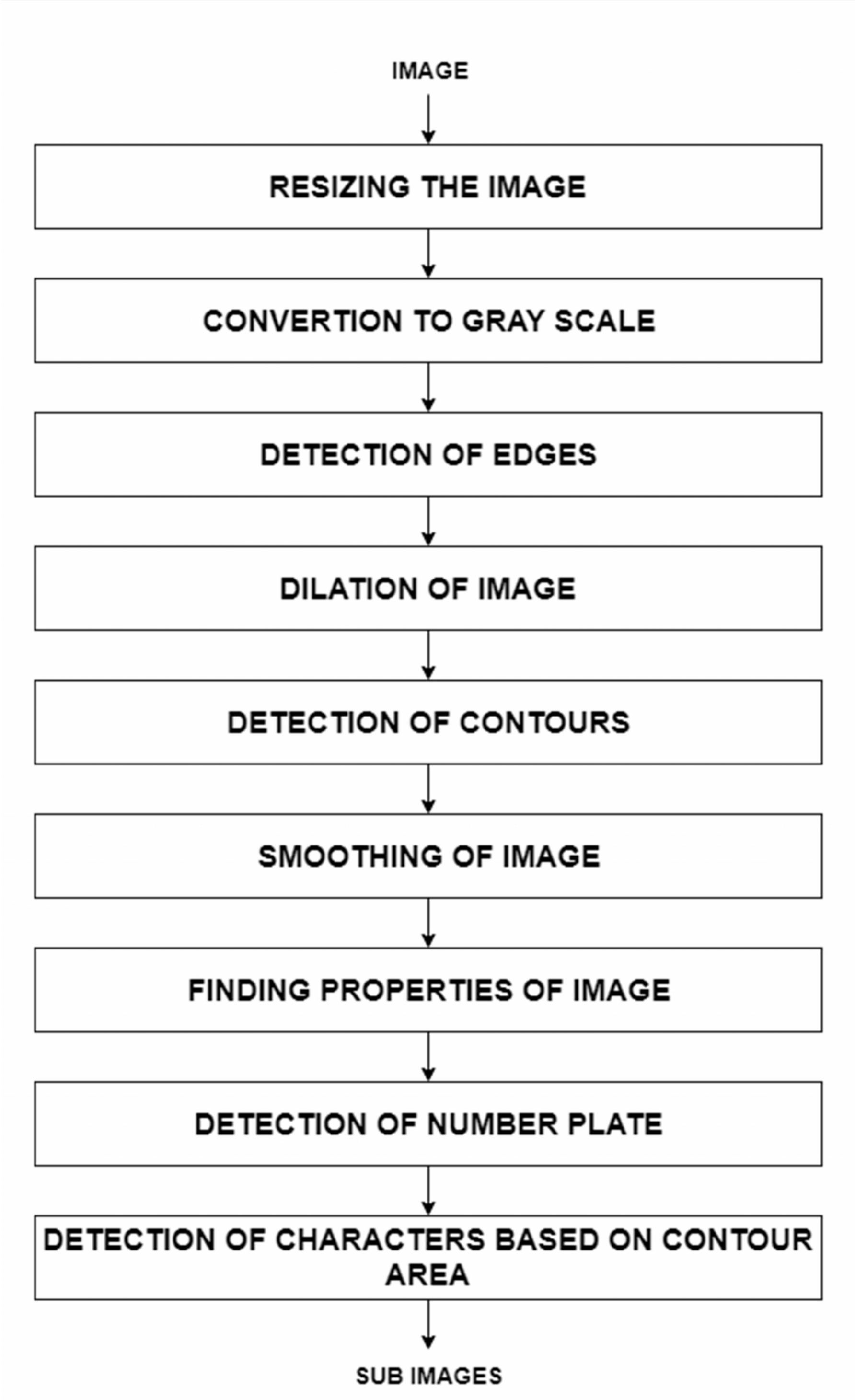
Uploading data to the database:

The number extracted from the image is uploaded to a cloud storage. For demonstration purpose the cloud storage used is ThingSpeak (by MATHWORKS). This cloud storage is specialized for use in IoT applications providing easy access and simple API for usage.

The software used for the Project is Matlab

3.3 Software architecture:

3.3.1 Image processing:



Step 1:

We first read the image using the Matlab function imread. The syntax for this is

A=imread('address of the image from the path of including image name ') where A is some variable. The example image is as follows:

Fig 4: Actual image

<u>Step 2:</u>

After reading the image we resize the image to contain 480 rows and any number of columns. This is done so that all images have the same resolution of 480p irrespective of its original resolution.

<u>Step 3:</u>

The resized image is now in RGB (Red Green Blue) format, that is the image is coloured. Our main objective is to find contours in the image but the **hue** and **saturation** information of the RGB image can deviate us from finding the contours. In order to remove **hue** and **saturation** content and retain the **luminance** we use the conversion of the image from RGB to grayscale. This is done by the Matlab function **rgb2gray**. Its syntax is as follows:

imgray = rgb2gray(im)

Here im is the image in RGB format and imgray is the grayscale image from the function rgb2gray. The grayscale image is as follows,

Fig 5: Gray converted image

<u>Step 4:</u>

Now the grayscale image contains shades of grey as it's colour. The next objective is to binarize the image to just white and black. This is done by the inbuilt Matlab function **imbinarize**. The syntax for the imbinarize is as follows:

Imbinary=imbinarize(imgray)

Where imgray is the input and imbinary is the resulting image after binarization.

The imbinarize by default uses Otsu's method in order to calculate a threshold based on the grayscale image. After the threshold is calculate all the values above the threshold are made 1(complete black) and all the values below threshold are made 0(complete white). The binarized image is as follows:

Fig 6: Binarised image

Step 5:

Now we try to find the edges in the grayscale images (before binarized) and this is done by the inbuilt Matlab function **edge**. The syntax is as follows:

imedge=edge(imgray,method)

Here the inputs are the grayscale image and method string. The method string is used to specify the method with which the edges have to be detected.

Some of the methods commonly used are:

- **Sobel**-Finds edges at those points where the gradient of the image I is maximum, using the Sobel approximation to the derivative.
- **Prewitt**-Finds edges at those points where the gradient of I is maximum, using the Prewitt approximation to the derivative.
- **Roberts**-Finds edges at those points where the gradient of I is maximum, using the Roberts approximation to the derivative.

Here we use the Sobel method to find the edges in the grayscale image.

The edge detected image is as follows:

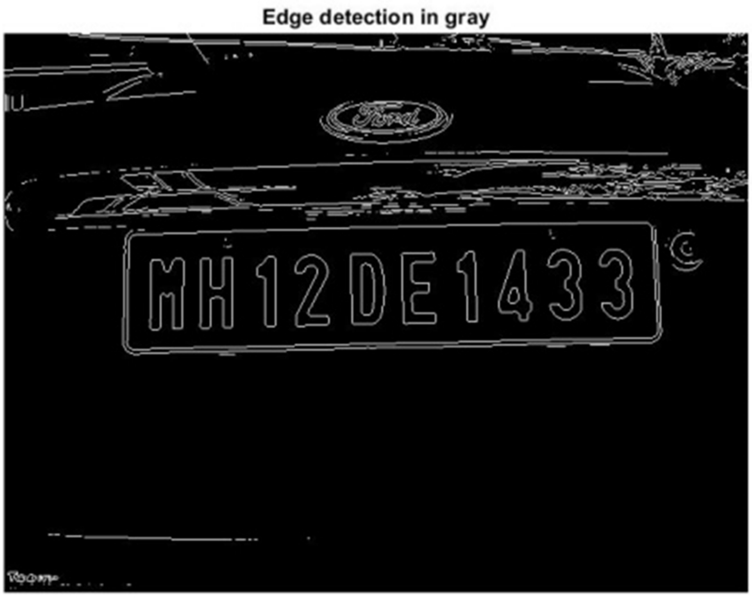


Fig 7: Edge detection in image

Step 6:

Then we create a structural element using the inbuilt Matlab function **strel**. The strel creates a flat structural element where the pixels within the morphological computation are true (or 1) and the others are false (or 0). These structures are very useful in dilation and erosion. The syntax to create a diamond structural element is as follows:

se=strel(‘diamond’,r)

where r is the distance from the origin to the vertices of the diamond.

Using this structural element, we dilate the edge image to make the edges broader. This is done by the inbuilt Matlab function **imdilate**. Its syntax is as follows:

Imdil=imdilate(imedge,se)

where imedge (edge image) and se (structural element) are the inputs and the output image is a dilated image. The dilated image is as follows:

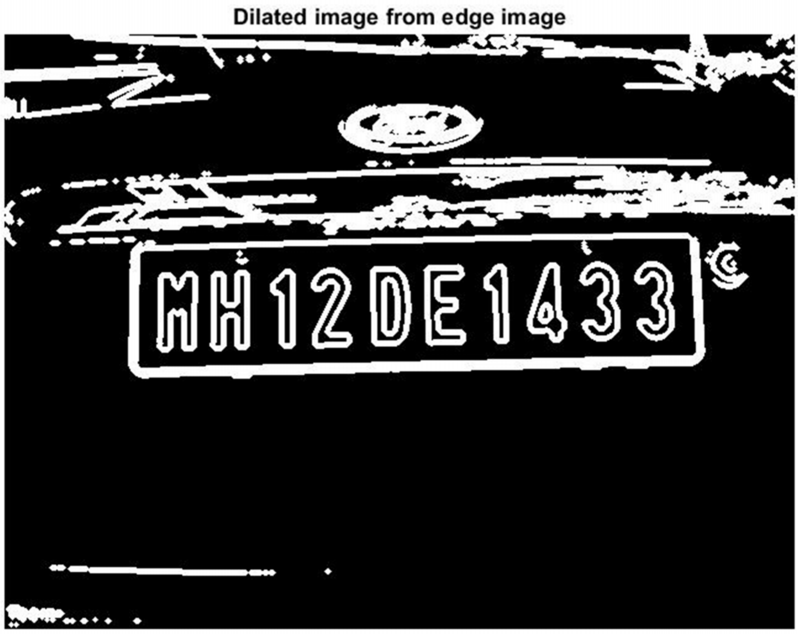


Fig 8: Creation of structural element from edges

<u>Step 7:</u>

In the dilated image some of the pixels are not uniform and have holes in them (random 0s in a region of 1s and vice versa). In order to make the image more uniform the holes are filled using the inbuilt Matlab function **imfill**. The syntax is as follows:

imFill=imfill(imdill,'holes')

where imdill is the dilated image and imFill is the holes filled image.

The filled image is as follows:

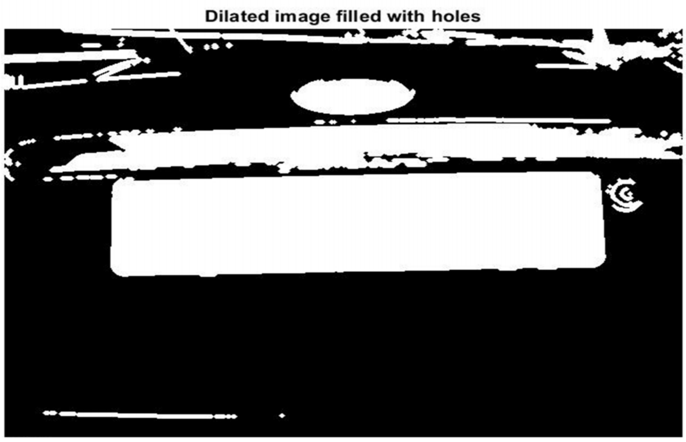


Fig 9: Filled image

Step 8:

We now create another structural element of type diamond with a bigger length for erosion using the same syntax as given above. Then we erode the hole filled image by using the inbuilt Matlab function **imerode**. This returns the eroded image as desired. The syntax is as follows:

imErode=imerode(imFill,se1)

where imErode is the eroded image, imFill is the holes filled image and se1 is the second structural element defined. The eroded image is as follows:

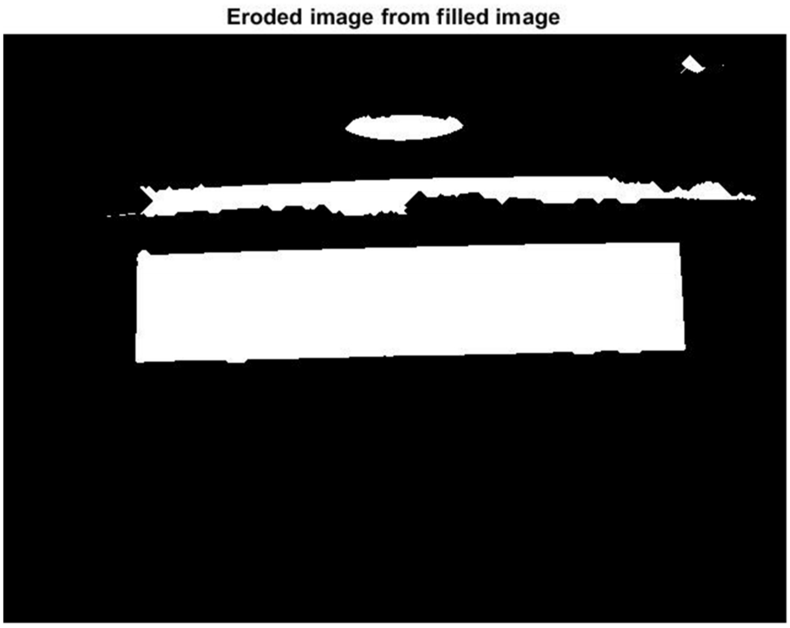


Fig 10: Erroded image

Step 9:

Now using the inbuilt Matlab function **regionprops** we extract the image properties of imErode (eroded image) such as the area, bounding box and sub image into a single structure.

The Syntax for this is as follows:

Iprops=regionprops(imErode,'Area','BoundingBox','Image')

After the extraction of properties, we find the bounding box of the sub image with the largest area (which is the required licence plate region).

Step 10:

Using the BoundingBox found from the previous step above we crop out that region of image from the binary image. This is done by the inbuilt Matlab function **imcrop**. The syntax for imcrop is as follows:

ImCrop=imcrop(imbin,BoundingBox)

where ImCrop is the cropped out binary image and imbin is the binary image.

The cropped out binary image is as follows:

Fig 11: Number plate extracted

Step 11:

After the above step we resize the required licence plate image to 240p as follows

imres=imresize (imCrop, [240, NaN])

Step 12:

We now use the inbuilt function **imopen** to morphologically open (which is erosion followed by dilation) to clear the dusts in the image imres. We declare a structuring element of type rectangle for this operation (The same structuring element is used for both erosion and dilation).

imclear=imopen (imres, strel (‘rectangle’, [4,4]))

Step 13:

We now clear the image again using the inbuilt Matlab function **bwareaopen**. The syntax of this function is as follows:

imAlter=bwareaopen(~imclear,500)

Here the ~imclear is the inverted imclear image (all white as black pixels and all black as white pixels) and the 500 pixels is the area threshold. If the objects in the image do not have a minimum of 500 pixels area the object is removed from the image.

Fig 12: Pixel invertion on image

Step 14:

We now take the image properties of the imAlter by using **regionprops** into another structure. Now in the sub-images if the width and height is greater than the half of imAlter height and one-third of imAlter height then that image is segmented as a character of the licence plate.

Fig 13: Segmented characters

3.1.2 Neural networks:

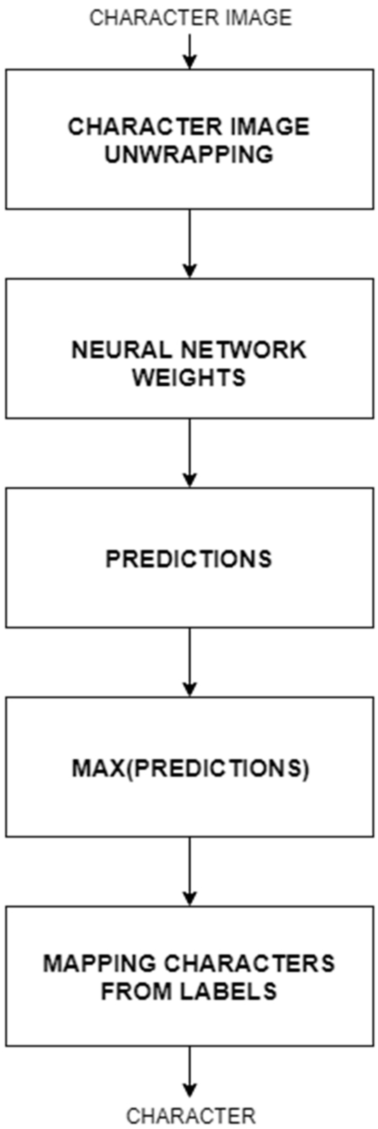


Fig 14: Flow of character recognition

The neural network used for the project is a simple shallow network with only one hidden layer. The processed character-image is resized to an image of 28pixels X 28pixels. Then it is unwrapped and introduced as input to the Neural Network. The output of the neural network is the confidence in prediction ranging between 0 and 1. The character with maximum confidence value is chosen to be the output for particular character image.

Number of nodes in each layer of neural network is as below:

Input layer – 784

Hidden layer – 150

Output layer – 36 (26 alphabets + 10 digits)

Neural network is trained using Gradient descent algorithm (Fig 15.)

Repeat until convergence

{

$$\theta_j := \theta_j - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)}$$

}

Fig 15: Gradient descent algorithm

The dataset used for the training is a subset of standard dataset described in [9]

Samples from the dataset are shown in Fig .

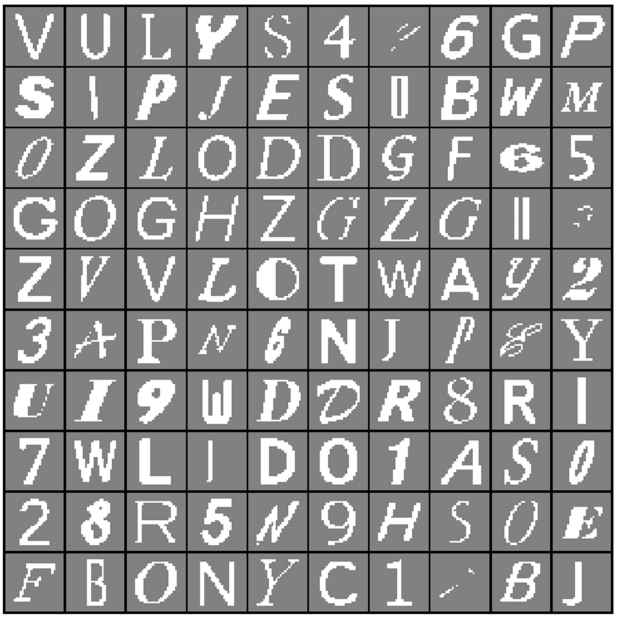


Fig 16: Dataset used for training the Neural Network

Basic flow of character recognition using Neural Network is shown in the block diagram (Fig 14.)

CHAPTER 4: RESULTS

Some input images and corresponding outputs are listed in the table below,

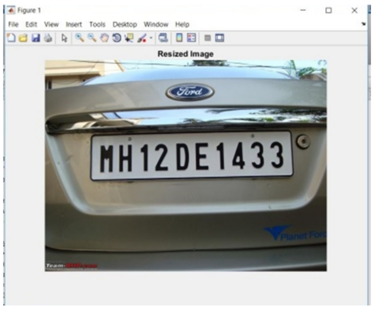
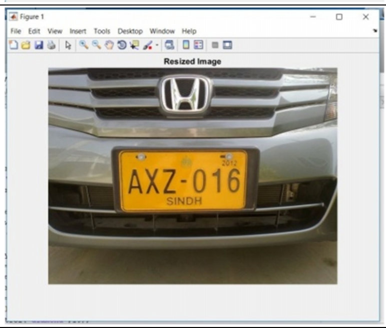
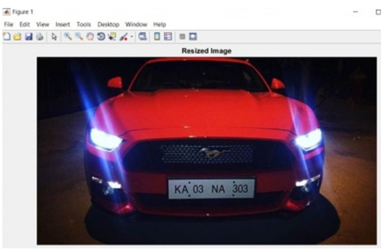
Input images	Outputs
	Loading Saved Neural Network Parameters ... The updated plate number is: M H 1 2 D E 1 4 3 3 fx >>
	Loading Saved Neural Network Parameters ... The updated plate number is: A W R 3 1 0 fx >>
	Loading Saved Neural Network Parameters ... The updated plate number is: A X Z 0 1 6 fx >>
	Loading Saved Neural Network Parameters ... The updated plate number is: K A 0 3 N A 3 0 3 fx >>

Table 1: Outputs for some sample input images

CHAPTER 5: CONCLUSION AND FUTURE WORK

5.1 CONCLUSION:

In this project an attempt is being made to develop computationally efficient Automatic number plate detection, recognition system. Outputs for various input images are documented in Chapter 4. This system is believed to be producing comparable success rate to that of other systems [2] [3].

5.2 FUTURE WORK:

Main improvement that can be made to this project is to try increase the accuracy of the prediction from neural networks. Many more sophisticated datasets can be used to train the model for higher accuracy. Also many developments in terms of image processing can be done as advancement to the current software. For example current version of project would require the image to be taken at a particular angle. This constraint can be removed by employing various image corrections and de-noising techniques. This system can also be made to work under extreme environmental conditions such as fog and rain which would be very much necessary when deployed as a part of smart traffic signal monitoring system. This system will also be useful in automation of toll collection.

This project produces the basis for the entire smart transport control and management system. The entire fining process for traffic rules violations can be easily automated because all we need to recognise a vehicle is its number. This would require generation and maintenance of a detailed database. The project can further be extended into development of a complete mobile application (App) which will be useful for traffic police personnel.

REFERENCES

[1] Kaur, Sarbjit & Kaur, Sukhvir. (2014). *An Efficient Approach for Number Plate Extraction from Vehicles Image under Image Processing.*

[2] Koval, Vasyl, et al. "*Smart license plate recognition system based on image processing using neural network.*" Second IEEE International Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, 2003. Proceedings. IEEE, 2003.

[3] Wu Wei, Yuzhi Li, Mingjun Wang and Zhongxiang Huang, "*Research on number-plate recognition based on neural networks,*" Neural Networks for Signal Processing XI: Proceedings of the 2001 IEEE Signal Processing Society Workshop (IEEE Cat. No.01TH8584), North Falmouth, MA, USA, 2001, pp. 529-538.

[4] J. Sharma, A. Mishra, K. Saxena and S. Kumar, "*A hybrid technique for License Plate Recognition based on feature selection of wavelet transform and artificial neural network*," 2014 International Conference on Reliability Optimization and Information Technology (ICROIT), Faridabad, 2014, pp. 347-352.

[5] Bhushan, Bharat et al. "*License Plate Recognition System using Neural Networks and Multithresholding Technique.*" (2013).

[6] Gonzales, Rafael C., and Richard E. Woods. "*Digital image processing*." (2002).

[7] Haykin, Simon. *Neural networks: a comprehensive foundation*. Prentice Hall PTR, 1994.

[8] Kumar, Satish. *Neural networks: a classroom approach*. Tata McGraw-Hill Education, 2004.

[9] T. E. de Campos, B. R. Babu and M. Varma. *Character recognition in natural images.* In Proceedings of the International Conference on Computer Vision Theory and Applications (VISAPP), Lisbon, Portugal, February 2009.