ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 3, Issue 1, pp: (256-265), Month: January - March 2015, Available at: www.researchpublish.com

TOLL PLAZA AUTOMATION

Keerthi.R[1], Dhivya.M[2], Divya.K [3], Mausami Gowsami[4]

[1,2,3,4] M.Tech students, CSE Dept, Christ University, Bangalore, India

***Abstract*: Recognizing number plate, using image processing techniques, is indicating new ways of capturing traffic data. Recent advances in computer vision technology and the falling prices of the related devices encourage the use of video/ closed circuit television cameras by making it practical to automatically identify vehicles visually on-line or off-line. Automatic vehicle identification system could be achieved using automated number-plate reading with high accuracy and low processing time. The objective of this project is to explore the potential of using the neural network for numberplate reading. Image processing techniques are used to extract the location of each included character, and the neural network is applied to the character recognition process.**

1. INTRODUCTION

In this paper we are trying to make the automatically pay of the tax at Toll-Plaza by using the Digital Image Processing to recognize vehicle number. After recognizing the number plate, money will be directly deducted from the account of the owner of the vehicle depending on the database. So we have to reduce traffic at near Toll Plaza and it also save time of the customer.

MATLAB (matrix laboratory) is a numerical computing environment. In that we are using mostly the Image Processing Toolbox. The Image Processing Toolbox software is a collection of functions that extend the capability of the MATLAB numeric computing environment. The toolbox supports a wide range of image processing operations, including Spatial image transformations, Morphological operations, Neighbour hood and block operations, Linear filtering and filter design, Transforms, Image analysis and enhancement, Image registration, Deblurring, Region of interest operations.

An artificial neural network (ANN), usually called neural network (NN), is a mathematical model or computational model that is inspired by the structure and/or functional aspects of biological neural networks. A neural network consists of an interconnected group of artificial neurons, and it processes information using a connectionist approach to computation. In most cases an ANN is an adaptive that changes its structure based on external or internal information that flows through the network during the learning phase. Modern neural networks are non-linear statistical data modelling tools. They are usually used to model complex relationships between inputs and outputs or to find patterns in data.

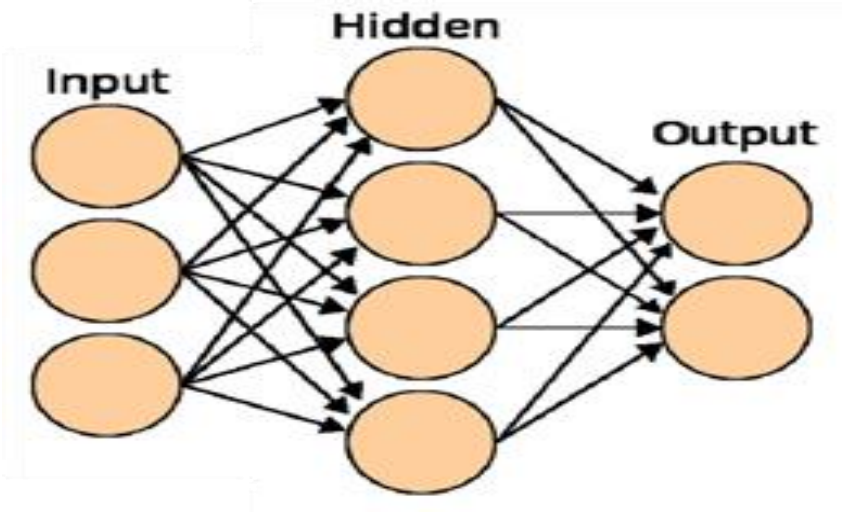


Fig. 1:- Neural Network

• Why use neural networks?

Neural networks, with their remarkable ability to derive meaning from complicated or imprecise data, can be used to extract patterns and detect trends that are too complex to be noticed by either humans or other computer techniques. A trained neural network can be thought of as an "expert" in the category of information it has been given to analyze. This expert can then be used to provide projections given new situations of interest and answer "what if" questions.

Other advantages include:

1. Adaptive learning: An ability to learn how to do tasks based on the data given for training or initial experience.
2. Self-Organization: An ANN can create its own organization or representation of the information it receives during learning time.
3. Real Time Operation: ANN computations may be carried out in parallel, and special hardware devices are being designed and manufactured which take advantage of this capability.
4. Fault Tolerance via Redundant Information Coding: Partial destruction of a network leads to the corresponding degradation of performance. However, some network capabilities may be retained even with major network damage.

• Neural networks versus conventional computers:

Neural networks take a different approach to problem solving than that of conventional computers. Conventional computers use an algorithmic approach i.e. the computer follows a set of instructions in order to solve a problem. Unless the specific steps that the computer needs to follow are known the computer cannot solve the problem. That restricts the problem solving capability of conventional computers to problems that we already understand and know how to solve. But computers would be so much more useful if they could do things that we don't exactly know how to do.

Neural networks process information in a similar way the human brain does. The network is composed of a large number of highly interconnected processing elements(neurones) working in parallel to solve a specific problem. Neural networks learn by example. They cannot be programmed to perform a specific task. The examples must be selected carefully otherwise useful time is wasted or even worse the network might be functioning incorrectly. The disadvantage is that because the network finds out how to solve the problem by itself, its operation can be unpredictable.

On the other hand, conventional computers use a cognitive approach to problem solving; the way the problem is to solved must be known and stated in small unambiguous instructions. These instructions are then converted to a high level language program and then into machine code that the computer can understand. These machines are totally predictable; if anything goes wrong is due to a software or hardware fault.

Neural networks and conventional algorithmic computers are not in competition but complement each other. There are tasks are more suited to an algorithmic approach like arithmetic operations and tasks that are more suited to neural networks. Even more, a large number of tasks, require systems that use a combination of the two approaches (normally a conventional computer is used to supervise the neural network) in order to perform at maximum efficiency.

2. EXTRACTING NUMBER PLATE

```
I=imread (image_name);
I=imresize (I,[1024 1024],'Method',
'bicubic','Colormap','optimized');
I=rgb2gray (I);
level = graythresh(I);
I = im2bw (I, level);
H = fspecial ('gaussian');
I2 = filter2 (H,I);
I3 = imclearborder (I2);
= graythresh (I3);
I3 = im2bw (I3, level);
I4 = imfill (I3,'holes');
I6=I4-I3;
I6=bwareaopen (I6,50);
```

2.1 Reading the image and resizing it.

ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 3, Issue 1, pp: (256-265), Month: January - March 2015, Available at: www.researchpublish.com

Read image from graphics file and storing the image in form of matrix in variable I using the command I=imread(image_name); Resize the every image to same size 1024x1024 using the command, I=imresize (I,[1024 1024],'Method','bicubic','Colormap','optimized');

Fig 2:- Main Image

Where method used to resize is Bicubic interpolation in which the output pixel value is a weighted average of pixels in the nearest 4-by-4 neighbourhood and the Colormap is 'optimized'. After executing above command fig (2) is appeared.

2.1 Image Binarazation

I = rgb2gray(I) converts the truecolor image I to the grayscale intensity image I. rgb2gray

Fig: 3

Converts I images to grayscale by eliminating the hue and saturation information while retaining the luminance.
level = 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].

ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 3, Issue 1, pp: (256-265), Month: January - March 2015, Available at: www.researchpublish.com

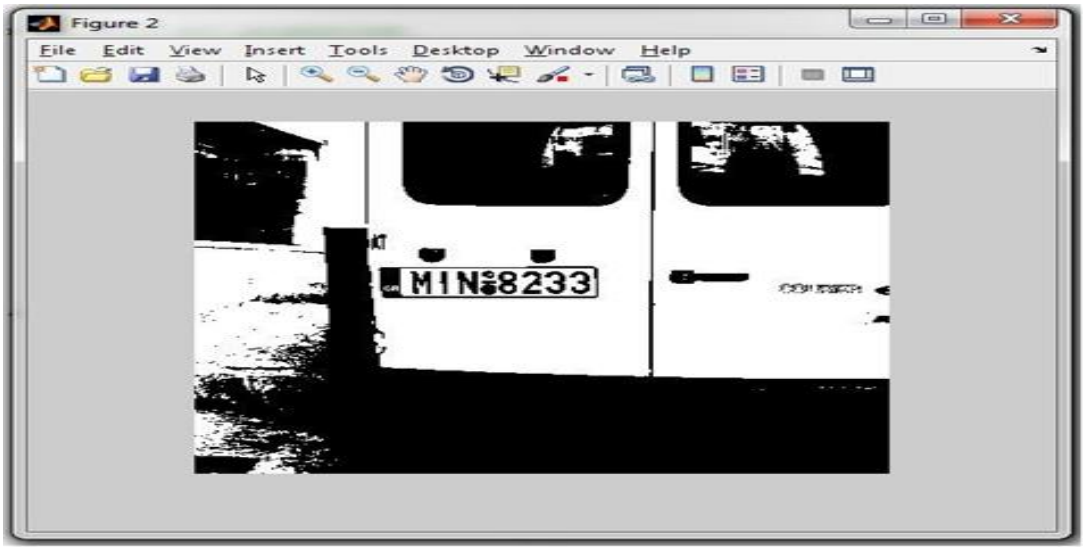


Fig.4

i = im2bw(i, level) converts the greyscale 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).

2.2 Remove unwanted Objects

The next step removes any object contiguous to the border of the image. Thus, we can get rid of unnecessary objects, while the plate characters will not be affected because they are surrounded by a black background. After removing the unwanted objects, a specific filter is used for illuminating the very small objects based on the size of each one.

H = fspecial ('gaussian');

I2 = filter2 (H,I);
I3 = imclearborder (I2);

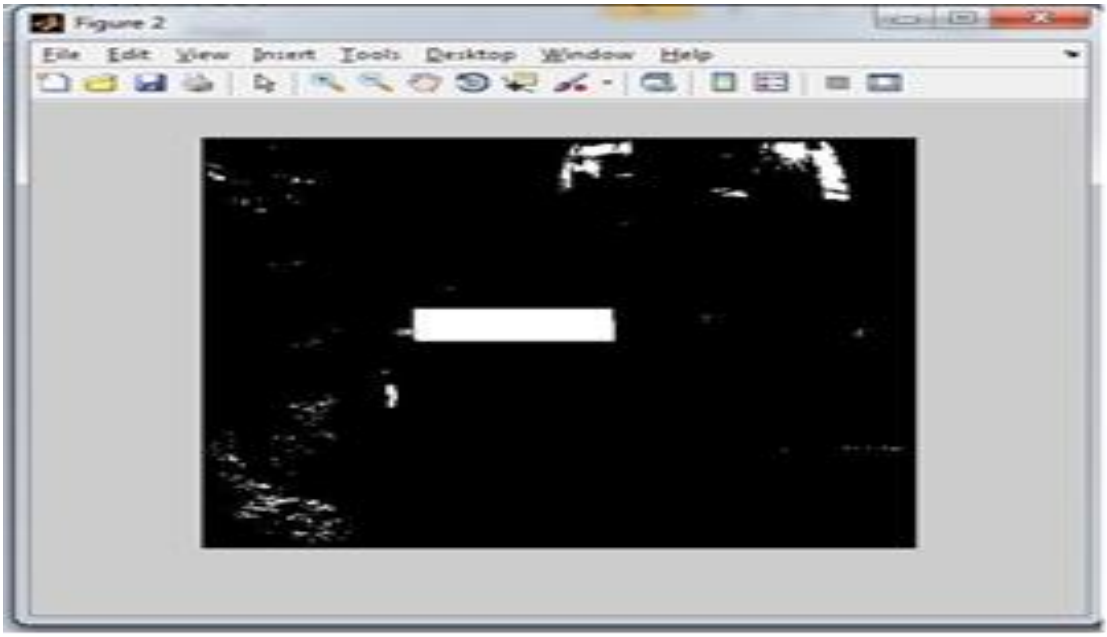


Fig: 5

ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 3, Issue 1, pp: (256-265), Month: January - March 2015, Available at: www.researchpublish.com

Where I4 = imfill (I4, 'holes') fills holes in the binary image I. A hole is a set of background pixels that cannot be reached by filling in the background from the edge of the image.

2.3 Subtraction

I6=I4-I3; will subtract the I3 from I4 which will give the Number Plate. I6=bwareaopen (I6,50); removes from a binary image all connected components (objects) that have fewer than 50 pixels, producing another binary image.

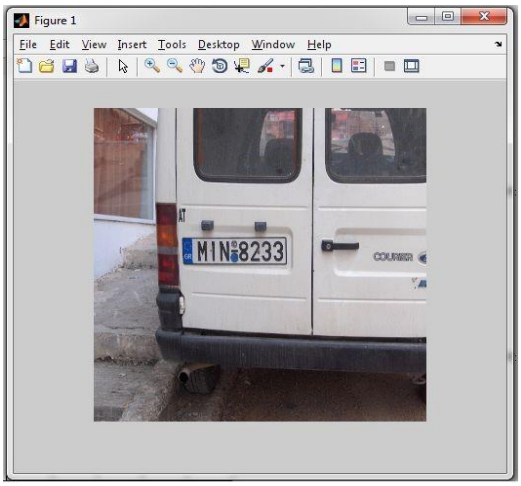


Fig.6

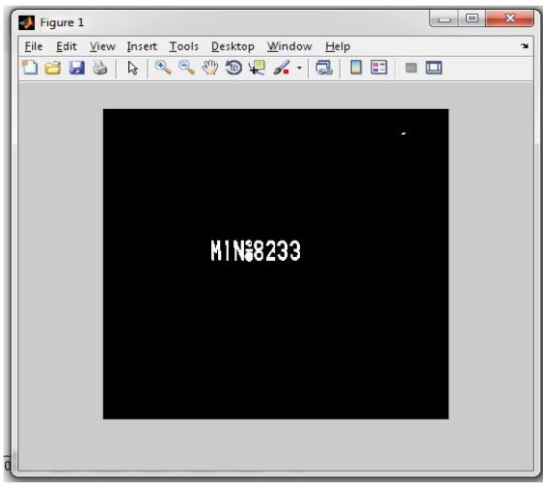


Fig.7

Fig. 8

This algorithm is used to divide the lines. First we use "find" function locates all nonzero elements of image given into find function, and returns the linear indices of those elements in output vector than crop the image according to output of findfunction.

3. EXTRACTING THE CHARACTER (GUI2.m)

We design GUI for showing what happen image is coming from input device. From the output of above line program we extract the one character and string them in one image variable. For extracting the character we use function [L num] = bwlabel(I7) which will give the connected component and from we are taking last four component as we are focusing on last four digits of the number plate. These components are competes with the neural network to find the corresponding match. These characters are resized to 10x8 as we have designed the neural network using 10x8 matrix representations of digits.

ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 3, Issue 1, pp: (256-265), Month: January - March 2015, Available at: www.researchpublish.com

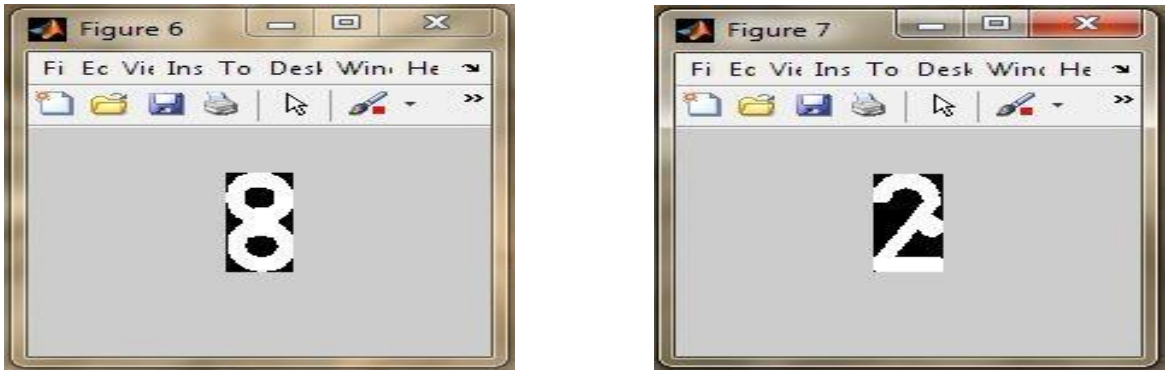


Fig. 10

We can use different function for filtering the image, remove noise, for image threshold and for converting image into binary representation. So after extracting the one character from given input image we use neural network to identify which character is this.

4. NEURAL NETWORKS (EXTRACT. m FILE)

We are using artificial neural network in following two ways.

(1) Manual database

In this technique we create manual database of digits. For this we use 10x8 matrixes in which we store 1 according to respective digit. It is show in following figure.

The disadvantage of this technique is that we can't create all possible neural networks of single digits. Because one digits has many possibilities. For example digit 5 can be represented as following three ways but there are so many ways. For that we are using automatic data base.

					1	1	1
			1	1	1	1	1
	1	1	1			1	1
1	1					1	1
						1	1
						1	1
						1	1
						1	1
						1	1
						1	1

1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1						
1	1		1	1	1		
1	1	1	1	1	1	1	
1	1	1	1	1	1	1	1
						1	1
						1	1
1	1	1	1	1	1	1	1
		1	1	1	1	1	

			1	1	1	1	1
		1	1	1	1	1	1
	1	1					
1	1						
1	1	1	1	1	1	1	
1	1	1	1	1	1	1	1
						1	1
						1	1
1	1	1	1	1	1	1	
	1	1	1	1	1		

ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 3, Issue 1, pp: (256-265), Month: January - March 2015, Available at: www.researchpublish.com

1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1						
1	1						
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
						1	1
						1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1

Fig. 11

Automatic database:

In this technique we are taking real images of existing number plates and recognise character from that sample image and convert them into 10x8 matrix. As we can't create all possible ways of how one digit looks instead of that this program will create automatic database. For example digit 1 can have different font in different number plates, we are storing all this patterns in matrix and use it for neural network. Similarly for all other digits we are making patterns.

4.1 Function used for creating network:

For making network we are using NEWPR and for training we are using TRAINPR function. NEWPR Create a pattern recognition network. NEWPR returns a network exactly as NEWFF would, but with an output layer transfer function of 'TANSIG' and additional plotting functions included in the network's net.plotFcn property. NEWFF Create a feedforward backpropagation network. TANSIG is a neural transfer function. Transfer functions calculate a layer's output from its net input. TANSIG Hyperbolic tangent sigmoid transfer function.

Syntax,

A = tansig(N,FP) dA_dN = tansig('dn',N,A,FP)

INFO = tansig(CODE)

An important application of neural networks is pattern recognition. Pattern recognition can be implemented by using a feed-forward (figure 1) neural network that has been trained accordingly. During training, the network is trained to associate outputs with input patterns. When the network is used, it identifies the input pattern and tries to output the associated output pattern. The power of neural networks comes to life when a pattern that has no output associated with it, is given as an input. In this case, the network gives the output that corresponds to a taught input pattern that is least different from the given pattern

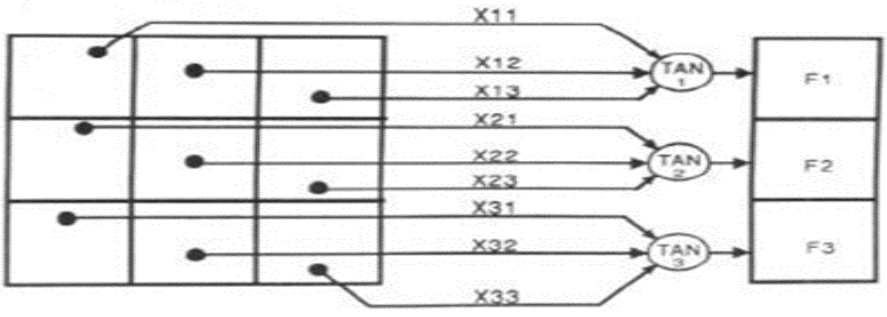
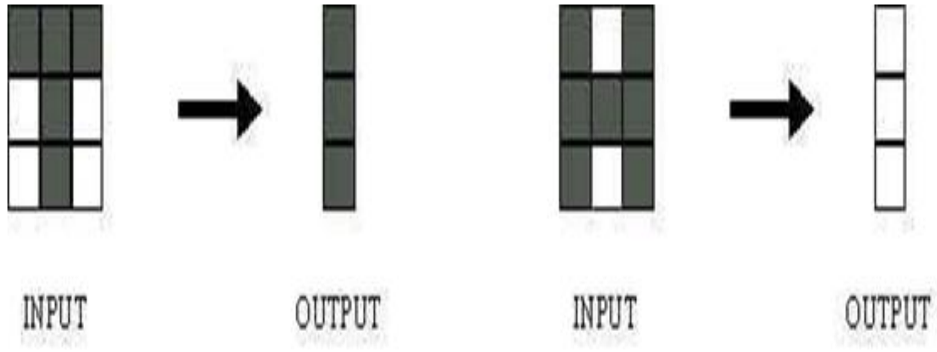


Fig. 12

ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 3, Issue 1, pp: (256-265), Month: January - March 2015, Available at: www.researchpublish.com

For example: The network of figure 1 is trained to recognise the patterns T and H. The associated patterns are all black and all white respectively as shown below.



X21:	0	0	0	0	1	1	1	1
X22:	0	0	1	1	0	0	1	1
X23:	0	1	0	1	0	1	0	1
OUT:	1	0/1	1	0/1	0/1	0	0/1	0

Fig: 13

X11:	0	0	0	0	1	1	1	1
X12:	0	0	1	1	0	0	1	1
X13:	0	1	0	1	0	1	0	1
OUT:	0	0	1	1	0	0	1	1

Fig: 14

From the tables it can be seen the following associasions can be extracted:

Fig: 15

In this case, it is obvious that the output should be all blacks since the input pattern is almost the same as the 'T' pattern.

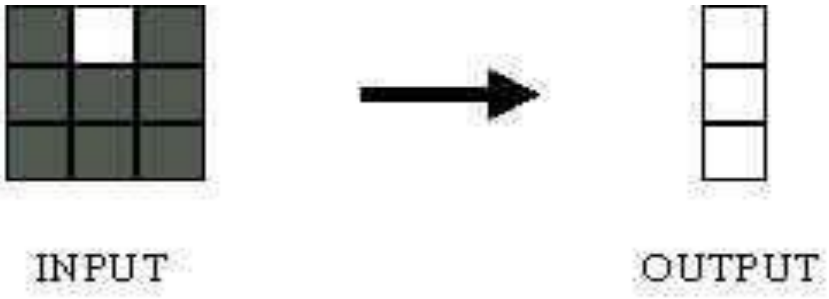


Fig:16

Here also, it is obvious that the output should be all whites since the input pattern is almost the same as the 'H' pattern. Figure.

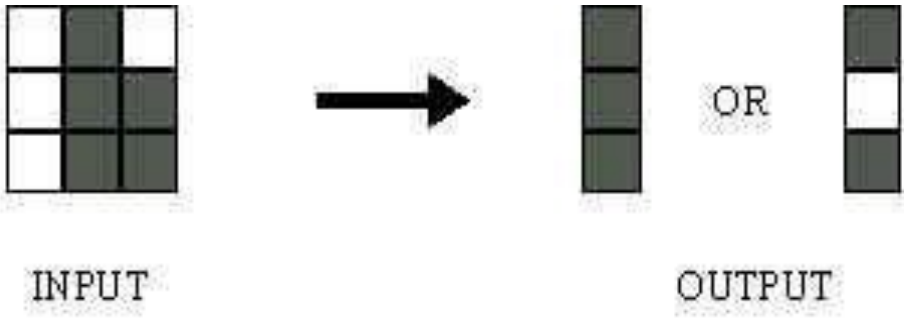


Fig: 17

Here, the top row is 2 errors away from the a T and 3 from an H. So the top output is black. The middle row is 1 error away from both T and H so the output is random. The bottom row is 1 error away from T and 2 away from H. Therefore the output is black. The total output of the network is still in favour of the T shape.

Resilient Back propagation (trainpr)

Multilayer networks typically use sigmoid transfer functions in the hidden layers. These functions are often called "squashing" functions, since they compress an infinite input range into a finite output range. Sigmoid functions are characterized by the fact that their slope must approach zero as the input gets large. This causes a problem when using steepest descent to train a multilayer network with sigmoid functions, since the gradient can have a very small magnitude; and therefore, cause small changes in the weights and biases, even though the weights and biases are far from their optimal values.

Algorithms

trainrp can train any network as long as its weight, net input, and transfer functions have derivative functions.

Backpropagation is used to calculate derivatives of performance perf with respect to the weight and bias variables X. Each variable is adjusted according to the following:

dX = deltaX.*sign(gX);

Where the elements of deltaX are all initialized to delta0, and gX is the gradient. At each iteration the elements of deltaX are modified.

If an element of gXchanges sign from one iteration to the next, then the corresponding element of deltaX is decreased by delta_dec. If an element of gX maintains the same sign from one iteration to the next, then the corresponding element of deltaX is increased by delta_inc.

Training stops when any of these conditions occurs:

• The maximum number of epochs (repetitions) is reached.

• The maximum amount of time is exceeded.

• Performance is minimized to the goal.

• The performance gradient falls below min_grad.

• Validation performance has increased more than max_fail times since the last time it decreased (when using validation).

5. CONCLUSION

Currently we complete following things. .Extract Number Plate from Image. .Recognize digits of number plate using neural network from extracting image. The Future Implementation. .Recognize alphabets of number plate. .Store recognizes number in database. .Extract image from video itself and maintain live working of project with high resolution camera. .According to database of customer money is automatic deducted from the account of the customer.

REFERENCES

[1] Jensen, John R., and Kalmesh Lulla. "Introductory digital image processing: a remote sensing perspective." (1987): 65-65.

[2] Lee, J. S. (1983). Digital image smoothing and the sigma filter. Computer Vision, Graphics, and Image Processing, 24(2), 255-269.

[3] Cuche, Etienne, Pierre Marquet, and Christian Depeursinge. "Spatial filtering for zero-order and twin-image elimination in digital off-axis holography." Applied optics 39.23 (2000): 4070-4075.

[4] Hornik, Kurt, Maxwell Stinchcombe, and Halbert White. "Multilayer feedforward networks are universal approximators." Neural networks 2.5 (1989): 359-366.

[5] Specht, Donald F. "Probabilistic neural networks." Neural networks 3.1 (1990): 109-118.