

## Recognition of Numerals Using Neural Network

Ms. Shraddha S. Talele<sup>1</sup>, Mr. P. M. Mahajan<sup>2</sup>

<sup>1</sup>(P. G. Student Extc Department, J. T. M. C.O.E. Faizpur, India)

<sup>2</sup>(Extc Department, J. T. M. C.O.E. Faizpur, India)

---

**Abstract :** Recognition of handwritten numerals is a challenging problem researchers had been research into this area for so long especially in the recent years. The goal of optical digit recognition is to classify optical patterns contained in a digital image corresponding to numerals. In our study there are many fields concern with numbers, for example, checks in banks or recognizing numbers in car plates, the subject of digit recognition appears. A system for recognizing isolated digits may be as an approach for dealing with such application. In other words, to let the computer understand the English numbers that is written manually by users and views them according to the computer process. Image processing is simply the processing of the given image. The input is just an image that may be from any source, and the output may be an image or a set of parameters that are related to that particular image. The main objective for our system was to recognize isolated digits exist in different applications. For example, different users had their own handwriting styles where here the main challenge falls to let computer system understand these different handwriting styles and recognize them as standard writing. The process involves three phases namely pre-processing, training and recognition. Pre-processing stage performs noise removal, binarization, labelling, rescaling and segmentation operations. Training stage adopts back propagation with feed forward technique. Recognition stage recognizes input images of numerals.

**Keywords:** - Artificial Neural network, segmentation, geometrical feature extraction, OTSU's method, feed forward back propagation algorithm.

---

### I. INTRODUCTION

Character recognition has a great potential in data and word processing for instance, automated postal address and ZIP code reading, data acquisition in bank checks, processing of archived institutional records, etc. This paper represents system for recognized digits is, which may benefit in various fields, the system concerning on isolated digits, the input is considered to be an image of specific size and format, the image is processed and then recognized to result of an edited digits. A neural network method is proposed to recognize handwritten numerals. Scanned images of handwritten numerals are input for recognition. The recognition operation comprises 3 stages: i) Pre-processing ii) Training iii) Recognition. Pre-processing stage does noise removal, binarisation, labeling, rescaling and segmentation operation. Training and recognition employs feed forward back propagation algorithm [1], [2]. Feature selection is also important to enhance classifier performance. The pixel counting technique is used to extract structural features [3].

### II. BACKPROPAGATION NEURAL NETWORK

A back propagation network generally consists of three or fewer layers of neurons: the input layer, the hidden layer, and the output layer. The back propagation algorithm works by supervised training. It first submits an input for a forward pass through the network. The network output is compared to the desired output, which is specified by a supervisor and the error for all the neurons in the output layer is calculated. In back propagation the error is propagated backward to earlier layers so that a gradient descent algorithm can be applied [4],[5]. This assumes that the network can run backwards, that means for any two connected neurons, the backward weights must be the same as the forward weights. Inputs arrive from other neurons, or from the environment. By taking the inner product, the neuron multiplies each input it receives by a weight which is specific to that input. This depends on the neurons that are connected by synapses and vary in strength. Any given neuron in the brain receives inputs from as many as thousands of other neurons, and it pays more attention to some than to others [6].The neurons add all of these weighted activations together, along with any bias that it might have, to form the inner product. The inner product is then subjected to a transfer function which is mostly non-linear results in the neuron's output.

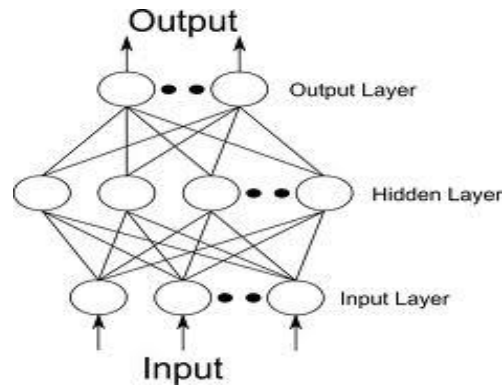


Fig1: Feed forward back-propagation neural network

Back propagation networks consist of either two or three layers of neurons and all the neurons behave as shown in figure. Any given neuron receives input from all neurons in the preceding layer, and it sends its output to all neurons in the layer below.

### III. FEATURE EXTRACTION

We have used geometrical feature set is a structural feature set. The pixel counting technique is used to extract structural features. In this technique, the pixels are detected in each row or column of the portion of the image where feature is to be extracted. The pixel counting technique is very useful to find any structural feature from the image and can tolerate the variances in the position of pixels, which form a particular feature. Structural features are extracted from each image. The digits are divided into 5 classes depending upon the similarity in features:

**Class 1-** It includes digits 0 and 8.

**Class 2-** It includes only digit 1.

**Class 3-** It includes digits 2 and 3.

**Class 4-** It includes digits 7, 4 and 9.

**Class 5-** It includes digits 5 and 6.

Class 2 has one further subdivision called class 6. The properties of this class are similar to class 3. Similarly, class 4 has two subdivisions, one for digit 4 whose properties are similar to digit 9 and second for digit 9 that has lower bar. There are total 17 features included to recognize the digits. The handwritten digits by one writer have roughly the same orientation, size and slant; although some variations exist that vary over the whole image. These variations are large enough to affect the features considerably, if slant and size variant features are used. However, by using features invariant to size, rotation and skew, a large variability is allowed. It is possible to find size and slant directions in the feature space using statistical features. However, digits with unusual size, rotation will not be correctly classified. An appropriate solution is therefore to use a mix of variant and invariant features. After a careful analysis of shape of digits for different sizes, fonts and slants, two sets of features are developed. The Features are divided into two sets:

(1) *Set S1 - Size and slant invariant features.*

(2) *Set S2 - Size and slant variant features.*

The Set S1 contains 11 features. The slant up to 45 degree in clockwise or anti clockwise direction of vertical image is taken. The user writes up to this slant in routine life. The slant is different from rotation. In rotation whole of the image is shifted but slant is just a slope of the image. Also size is different from scaling. So two assumptions are made about data:

(1) The data can have up to 45 degree slant either in clockwise or anticlockwise direction. This assumption is made after analysing data of 50 users.

(2) The digits are complete and have no missing parts. The broken digits up to 3-4 pixels are considered.

The size of a single digit has been taken of size varying from 16 by 16 to 300 by 400 pixels. The pixels of the digit images are stored in two dimensional array. Then all features are extracted by manipulation on this two-dimensional array. The following features are extracted from the digits:

(1) *F1 - Presence of side bars on Left and right side of the image* - If 90% part of the image has side bar on both sides then it is taken as 1 otherwise 0. It distinguishes class 1 from other classes. This feature is extracted by pixel counting method.

(2) *F2- Presence of one Lower sidebar (in 20% part of the image) on either side of the image* - If lower 20% part of the image has one side bar then it is taken as 1 otherwise 0. It distinguishes class 2 and class 4 from others.

(3) *F3 - The whole digit is vertical straight line only* This feature is taken by pixel counting in vertical direction. If whole digit is vertical straight line it is taken as 1 otherwise 0. This feature distinguishes class 2 from other classes.

(4) *F4 - Upper right edge* - If edge is there it is taken as 1 otherwise 0. This feature distinguishes class 5 from other classes.

(5) *F5 - Angle on left side in upper 80% part of the digit* This feature is taken by following contour profile. The 45 degree angle is taken as a feature to distinguish digit 4 from digit 9.

(6) *F6- Convergence* - This feature is taken as 1 if there is anywhere convergence in middle 80% portion of the image otherwise 0. This feature is taken to distinguish digit 0 from digit 8.

(7) *F7- Double loop with center single line* - This feature helps in separating certain cases of digit 2 from others.

(8) *F8 - Absence of middle left side bar (20%portion missing in middle)* - This feature distinguishes digits 2 and 3 from others. This portion is missing in digits 2 and 3.

(9) *F9 - Second topmost periphery* - This feature is very useful to differentiate digit 7 from digits 4 and 9.

(10) *F10 - Upper and lower right converge* - This feature is very important to separate digit 3 from digit 2 and other digits.

(11) *F11 - Left side bar (in 90% portion)* - This feature distinguishes digit 6 from digit 5. This feature is taken by following contour profile starting from upper left side pixel. The Set S2 contains 6 features. These features are not slant invariant but are size invariant. The feature is taken as 1 if the feature exist otherwise 0.

Set S2 contains following features:

(1) *F12 - Lower line* - This feature is taken by counting number of pixels in lower line. This feature distinguishes digits (4, 7, 9) from other digits.

(2) *F13 - Upper line* - This feature is also taken by counting number of pixels in upper line. This feature distinguishes digit 4 from digit 9.

(3) *F14 - Upper left edge* - This feature is taken by following contour profile. This feature helps in distinguishing digit 7 from other digits.

(4) *F15 - Right lower edge* - This feature is taken by following contour profile. This feature helps in separating digit 2 from digit 3. But it is not slant towards left invariant feature but it is slant towards right invariant feature.

(5) *F16 - Left lower side bar (in 20% portion)* - This feature distinguishes digit 5 from digit 6.

(6) *F17 - Right upper side bar (in 20% portion)* - This feature distinguishes digits 5 and 6 from other digits. All these features are extracted from the digits and stored in a file. The features are extracted from left and right contour profiles. Then classifier is used to recognize the digits based on these feature.

## 1. Proposed Research Work

There are three stages in the network are (i) Pre-processing, (ii) Training and (iii) Recognition.

### 1.1 Pre-processing Stage

*Step 1* Gaussian filter is used to remove noise from image.

$$g(x,y) = 1.e^{-(x^2+y^2)/2\sigma^2}/2\pi\sigma^2 \quad (1)$$

where  $x$  is the distance from the origin in the horizontal axis,  $y$  is the distance from the origin in the vertical axis, and  $\sigma$  is the standard deviation.

The standard deviation is calculated using,

$$\sigma = \sqrt{1/(n-1)\sum_1^n (x_i - \bar{x})^2} \quad (2)$$

*Step 2* OTSU's method is used to convert filtered colour image into binary image [7].

$$\sigma_b^2(t) - [\sigma^2 - \sigma_w^2(t)] = \omega_1(t)\omega_2(t)[\mu_1(t) - \mu_2(t)] \quad (3)$$

$\sigma_b^2(t)$  is the desired maximum threshold. Where  $\omega_i$  is the class probability and  $\mu_i$  the class means. The class probability  $\omega_i(t)$  is calculated as

$$\omega_i(t) = \sum_0^1 p(i)$$

and the class means by

$$\mu_i(t) = \sum_0^1 p(i) x(i)$$

Step 3 A label number is then provided for each binary-connected components, and those labels that share similar characteristics are segmented. The shortest path between each binary component is then find by,

$$\begin{aligned} \text{shortestpath}(i,j,k) &= \min(\text{shortest path}(i,j,k)) \\ \text{shortestpath}(I,k,k+1) &+ \text{shortestpath}(k,j,k-1) \end{aligned} \quad (4)$$

It first finds the shortest path (i, j, k) for all (i, j) pairs for k = 1, and then k = 2, etc. This continues until k = n.

Step 4 These binary images are rescaled by bilinear interpolation method [8] and then extracted. To find the value of the unknown function f at the point P = (x, y). It is assumed that the value of f at the four points Q11 = (x1, y1), Q12 = (x1, y2), Q21 = (x2, y1), and Q22 = (x2, y2). First linear interpolation is applied in the x-direction by,

$$f(R_1) \approx [(x_2-x)f(Q_{11})/(x_2-x_1)] + [(x-x_1)f(Q_{21})/(x_2-x_1)] \quad (5)$$

where  $R_1 = (x, y_1)$

$$f(R_2) \approx [(x_2-x)f(Q_{11})/(x_2-x_1)] + [(x-x_1)f(Q_{21})/(x_2-x_1)] \quad (6)$$

Where  $R_2 = (x, y_2)$

Interpolation in y-direction is found by,

$$f(P) \approx [(y_2-y)f(R_1)/(y_2-y_1)] + [(y-y_1)f(R_2)/(y_2-y_1)] \quad (7)$$

After calculating the interpolation value in the x and y direction, the desired interpolation value f(x, y) is found by using equations 5, 6 and 7 as,

$$f(x, y) \approx [f(Q_{11})(x_2-x)(y_1-y)/(x_2-x_1)(y_2-y_1)] + [f(Q_{21})(x_2-x)(y_1-y)/(x_2-x_1)(y_2-y_1)] + [f(Q_{12})(x_2-x)(y_1-y)/(x_2-x_1)(y_2-y_1)] + [f(Q_{22})(x_2-x)(y_1-y)/(x_2-x_1)(y_2-y_1)] \quad (8)$$

Step 5 The extracted binary images are fed into a multilayer feed-forward back propagation neural network to recognize digits.

### 1.2 Training the Network

In this paper Three layers network is used.

The simplest among the associative memory model is the feed-forward type of neural network. A multi-layer feed-forward back propagation neural network is used for recognizing handwritten numerals.

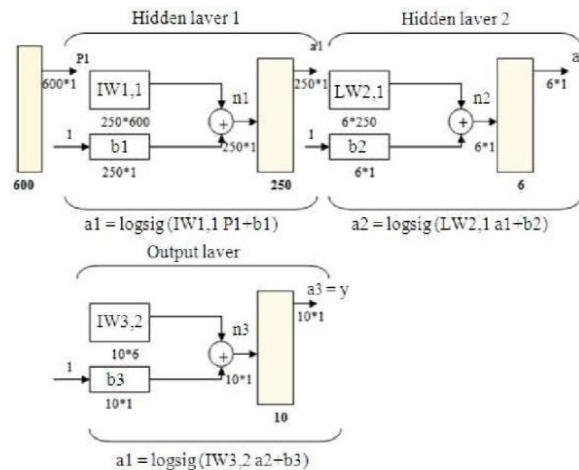


Fig 2: Three layers network, two hidden and one output, with 250, 6, 10 neurons respectively for each layer.

The training network consists of three layers the input layer, hidden layer and the output layer. X is the input vector applied to the network through the input layer and the output Y is produced across the output layer. In the case of multilayer network, the NET value is calculated for each neuron layer by layer using,

$$\text{NET} = XW \quad (9)$$

Where X is the input and W is the weight value. An activation function is applied to the NET value to process the NET signal to further layers, and the OUT value is calculated by,

$$\text{OUT} = F(XW) \quad (10)$$

For example the target value is set to 1. If the calculated OUT value is not greater than or equal to 1, the desired target value, the back propagation algorithm is applied and the associated weights are adjusted using the Delta rule.[6]

The Delta value is calculated using the formula,

$$\delta = \text{OUT}(1-\text{OUT})(\text{Target}-\text{OUT}) \quad (11)$$

Where (Target-OUT) gives the error signal, it is then multiplied by the squashing function OUT(1-OUT). After the Delta value is calculated, it is multiplied with the OUT value of the desired layers neuron, which is then multiplied with the training rate coefficient  $\eta$  as,

$$\Delta W = \eta \cdot \delta \cdot \text{OUT} \quad (12)$$

After that the result obtained is added to the weight value of the corresponding neuron in the hidden layer to the output layer by,

$$W(n+1) = W(n) + \Delta W \quad (13)$$

Where  $W(n)$  is the value of weight from neuron in the hidden layer to the neuron in the output layer, and  $W(n+1)$  is the weight value after adjustment. After adjustment of weight value, it is multiplied with the input vector  $X$  in the hidden layer and the OUT value is calculated.

### 1.3 Recognition

After completion of training, the pre-processed images are fed into the network for recognition. If the given input is same as compared with the trained data stored in memory, the proper image of the digit is extracted and displayed. But if the given input is deformed or distorted, and also the calculated OUT value not the proper then the back-propagation algorithm is used since it propagates backwards and the weight value from the hidden layer to the output layer is adjusted to recalculate the OUT value.

## IV. RESULTS AND CONCLUSION

This paper shows implementation of a multi-layer feed-forward back propagation network that recognizes handwritten English numerals. The input image of numerals undergoes three stages namely pre-processing, training and recognition. pre-processing stage plays a vital role and influences the accuracy of recognition. Images of various sizes, shapes undergo pre-processing stage first. The pixel counting technique is used to extract structural features. In this technique, the pixels are detected in each row or column of the portion of the image where feature is to be extracted. The multi-layered feed-forward back propagation network is trained with pre-processed images. The final recognition stage recognizes images of any size and shape as input to the method. We have used three layers network, two hidden and one output, with 250, 6, 10 neurons respectively for each layer.

Finally we can conclude that back propagation neural network seems to be better than other techniques used for recognition and gives accuracy 98%. Still the drawback is similar strokes of two numerals are recognized as one for the other, further research is going on to solve this problem.

Table I: Network performance

Number of layers	3
Input vector	600
No. of neurons in hidden layer 1	250
No. of neurons in hidden layer 2	6
Neurons in output layer	10
Learning rate	1.21
Training performance	0.095
No. of epochs	107
Accuracy	98%

## REFERENCES

- [1] Y. Le Cu, B. Boser, J. S. Denker, D. Henderson, R.E. Howard, W. Hubbard and L. D. Jackel, "Handwritten Digit Recognition with a Back-Propagation Network", AT&T Bell Laboratories, Holmdel, N.J.07733 1990.
- [2] Jurgen F, 1997 "Isolated Handprinted Digit Recognition. In: Handbook of Character Recognition and Document Image Analysis," Bunke, H. and P.S.P. Wang (Eds.). World Scientific Publishing Company. ISBN: 10: 981022270X, pp.: 103-121.

- [3] Calin Enachescu, Cristian-Dumitru Miron, "Handwritten Digits Recognition Using Neural Computing", Scientific Bulletin of the Petru Maior University of Tirgu Mures, Vol.6(XXIII), 2009, ISSN 1841-9267.
- [4] Huang Hanmin, Huang Xiyue, Zhang Ping, Chai Yi, Shi Weiren, "ANN-Based Handwritten Character Recognition", in Proc. of the 38th SICE Annual Conference at Morioka, Aug 1999, pp. 1177-1180.
- [5] Aurene Fausett, "Fundamentals of Neural Networks: Architectures, Algorithms and applications", Prentice Hall, Englewood Cliffs, NJ 07632, pp.5-7, 1994.
- [6] Christopher L. Scofield, Lannie Kento, Jung-Chou Chang, "Multiple Neural Net Architectures for Character Recognition", IEEE Trans on Comcon Spring 91, Digest of Papers at San Francisco, CA, 25 Feb-1 Mar 1991, pp. 487-491.
- [7] N. Otsu "A Threshold Selection Method from Gray-Level Histograms". IEEE Transactions on Systems, Man, and Cybernetics. 1979, 9(1): 62-66
- [8] T.M. Lehmann, C. Gonner, and K. Spitzer, "Survey: Interpolations Methods In Medical Image Processing, "IEEE Transactions on Medical Imaging, vol. 18, no. 11, pp. 1049-1075, 1999.