

# Neural Network based Approach for Recognition of Text Images

Gaurav Kumar

Department of Information Technology,  
Panipat Institute of Engineering & Technology,  
Panipat, Haryana, India

Pradeep Kumar Bhatia

Department of Computer Science & Engineering,  
Guru Jambheshwar University of Science & Tech.,  
Hisar, Haryana, India

## ABSTRACT

Handwritten character recognition is a difficult problem due to the great variations of writing styles, different size of the characters. Multiple types of handwriting styles from different persons are considered in this work. An image with higher resolution will certainly take much longer time to compute than a lower resolution image. In the practical image acquisition systems and conditions, shape distortion is common processes because different people's handwriting has different shape of characters. The process of recognizing character recognition in this work has been divided into 2 phases. In the first phase, Image preprocessing is done in which image is firstly converted into binary form based on some threshold value obtained through Otsu's method. After that removal of noise is done using median filter. After that feature extraction takes place that is done here through Fourier descriptor method using Fourier transform and correlation between template made through training data and test data is obtained. A multilayer feed forward neural network is created and trained through Back Propagation algorithm. After the training, testing is done to match the pattern with test data. Results for various convergence objective of neural network are obtained and analyzed.

## General Terms

Binarization, Multilayer Perceptron, Skeletonization, Thresholding.

## Keywords

Character Recognition, Image Processing, MatLab, Neural Network.

## 1. INTRODUCTION

Character recognition is one of the most successful applications of neural network technology. In character recognition, printed documents are transformed into ASCII files for the purpose of editing, compact storage, fast retrieval through the use of computer. The recognition of character in a document becomes difficult due to noise, distortion, various character fonts and size, writing styles as handwriting of different persons is different.

Handwritten character recognition can be differentiated into two categories i.e. Online Handwritten character recognition and Offline Handwritten character recognition. On-line handwritten character recognition deals with automatic conversion of characters that may be written using a special digitizer; tablet PC in which a sensor picks up the pen-tip movements and also the pen-up/pen-down switching. Off-line handwritten character recognition deals with a data set, which is obtained from a scanned handwritten document.

Neural network has a wide application in the field of pattern recognition. In this work, English handwritten characters are

recognized through Feed Forward Multi-Layer Perceptron Network (MLPN) with one hidden layer. For training, back-propagation algorithm has been implemented. The network can be used to learn the character in the format of patterns and then generalizing from the trained network and recognizing the character that is presented in the form of image.

Following are the main objectives of character recognition system that can recognize English Characters are to classify a given input in character image into a binary pattern image.

- To develop/implement algorithms and data structures for handwritten English characters.
- To develop/implement image processing and neural network based implementation that will be used to convert any scanned and pre-processed document of handwritten English character to machine readable form.

Multilayer perceptron and one hidden layer [1], for handwritten English character recognition, the features are extracted from the handwritten character through boundary tracing along with Fourier descriptor. Here the shape of the character is analyzed and features are compared to distinguish each character. The skeletonised and normalized binary pixels of English characters were used as the inputs of the network.

RCS algorithm uses Back propagation algorithm [2] where Java Neural Network is used for implementation which provides a complete tool bar for training, recognizing and editing options for handwritten Tamil character recognition. Fourier descriptor feature as used in [1] is independent of position, size and orientation. Here the accuracy achieved is 97% and recognition speed of each character is 0.1 sec.

For handwritten Arabic character recognition, a stochastic gradient technique with back propagation algorithm [3] is used where updation is made after each presentation of a single pattern. Implementation was divided into two phases. The first is to recognize the main shape of the character, and the second stage is for dots recognition. Also, the characteristics, structure, and the training algorithm for the network are presented. The stochastic gradient converges much faster than true gradient especially on large redundant databases. The MSE calculated for training set was 0.087 and for test data was 0.42.

To recognize handwritten Kannada numeral recognition, a technique based on Crack Code and Fourier Descriptors [4] in which Handwritten printed Kannada digits are scan converted in to binary images and then normalized to a size of 40 x 40 pixels. The crack belongs to the boundary lies between a foreground and background pixel. Encoding this line (a sequence of horizontal and vertical pixel edges) yields the crack code of the digitized object boundary.

A technique based on Normalized Chain code and Fourier Descriptor [5] to recognize handwritten Marathi Numeral

recognition is presented. Chain code and Fourier Descriptors that capture the information about the shape of the numeral are used as features. After pre-processing the numeral image, the normalized chain code and the Fourier descriptors of the contour of the numeral are extracted. Freeman chain codes are generated by locating a boundary pixel, also called as starting pixel, and then moving along the boundary either clockwise or anticlockwise, finding next boundary pixel and assign this new pixel a code depending upon its location from the previous pixel. The process of finding the next pixel is terminated when starting pixel is encountered.

Multilayer perceptron with back propagation algorithm [6] for recognizing handwritten characters input through mouse. Algorithm and procedure to implement back propagation algorithm for character recognition is discussed in detail. The BPN network designed proposed has the ability to recognize stimulus patterns without affecting by shift in position not by a small distortion in shape of input pattern.

Normalization cooperated gradient feature [13] for handwritten Japanese and Chinese character recognition, which reduces the recognition error rate by factors ranging from 8.63 percent to 14.97 percent with high confidence of significance when combined with pseudo-two-dimensional normalization methods.

For recognizing handwritten character with different font sizes and multi font type, to achieve high accuracy results, centroid dithering training process with a low noise-sensitivity normalization procedure [14] is used. Here firstly, the two dimensional pixel array of the input character is pre-processed, normalized and decomposed into a vector, secondly, the vector is processed by the neural network to yield an output of ASCII 94 numbers. Thirdly, the neuron in the output layer with the highest threshold value is declared the winner, identifying the input character image. Fourth, a simple post processing algorithm is used to detect invalid characters and to discriminate between characters whose images become indistinguishable during pre-processing.

A technique based on Gaussian Hermite Moments with back propagation algorithm [15], for identifying vehicle through number plate; a classifier is designed to recognize character. Gaussian Hermite Moments are less sensitive to noise as compared to other moment technique used for pattern recognition. For Gaussian-Hermite moments, standard deviation  $\sigma$  of Gaussian function is used for recognition results. The larger standard deviation is helpful for removal of noise, but when the characters are too crowded together (or the characters too small); it will bring greater errors of recognition.

A hypercube approach and distance field approach for feature extraction and then Feed forward network with Clustering to recognize character [17], that can be used that cover the entire region of the vector space using a minimum number of clusters. The clusters are barred to contain the alien patterns, which results in a better classification performance. The number of clusters generated depends on the inter-class variance encountered in the data.

## 2. ALGORITHM USED FOR IMPLEMENTING CHARACTER RECOGNITION SYSTEM

Firstly, Reading an image file in binary format is done. Then threshold limits, skeletonization operations and Normalization operations are applied on the image

containing text and then extracting the features of normalized binary image through Fourier Descriptor method. The process of handwritten character recognition can be divided into phases as shown in the figure 1. Stepwise implementation of the algorithm implemented is as follows.

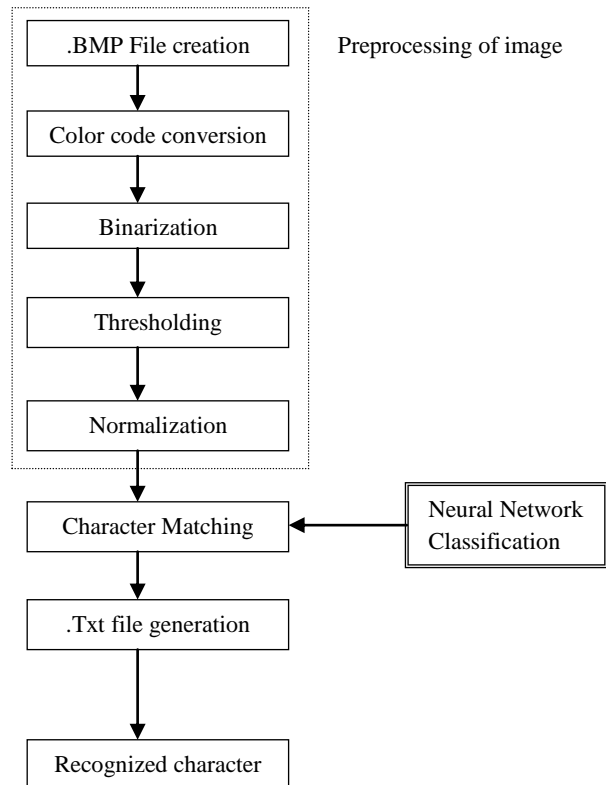


Figure 1: Block diagram for handwritten character recognition

### 2.1 Step-1: Pre-Processing

Pre-processing of the image means applying a number of procedures for thresholding, smoothing, filtering, resizing, and normalizing so that successive algorithm to final classification can be made simple and more accurate. Various Pre-processing Methods are explained below:

#### 2.1.1 Binarization & Thresholding

Document image Binarization (thresholding) refers to the conversion of a gray-scale image into a binary image. Two categories of thresholding:

- ❖ In the Global thresholding, one threshold value is picked for the entire document image that is basically based on an estimation of the background level from the intensity histogram of the image.
- ❖ Adaptive (local) thresholding is a method used for images in which different regions of the image may require different threshold values.

Now the first task is to convert this image to binary image (black and white). The first step in preprocessing is to binarize the numeral image (printed and hand written) so that the character image is converted into binary image (black and white) i.e. having pixel values 0 and 1. Generally the scanning image is true color (RGB image) and this has to be converted into a binary image, based on a threshold value. Here in this work, firstly checking the dimension of test image by size(). If

image is RGB image, then it is converted into gray image by `rgb2gray ()` i.e.

```
if size(testImg,3)==3
    testImg=rgb2gray(testImg);
```

To get a binary image, this RGB format image has to be converted to grayscale format, and then by using the threshold value (found by Otsu's method) this grayscale image is converted to binary image.

A thresholding application has to be performed on scanned gray scale images. Otsu's method is used in this work for the purpose of selecting the threshold and binarizing the gray scale images, so that resulting image has 0 as background pixels and 1 as foreground pixels.

#### Otsu's Method:

**Step 1:** count the number of pixel according to color (256 color) and save it to matrix count.

**Step 2:** calculate probability matrix P of each color,  $P_i = \text{count}_i / \text{sum of count}$ , where  $i=1, 2, \dots, 256$ .

**Step 3:** find matrix omega,  $\text{omegai} = \text{cumulative sum of } P_i$ , where  $i=1, 2, \dots, 256$ .

**Step 4:** find matrix mu,  $\text{mu}_i = \text{cumulative sum of } P_i * i$ , where  $i=1, 2, \dots, 256$  and  $\text{mu}_t = \text{cumulative sum of } P_{256} * 256$

**Step 5:** calculate matrix sigma\_b\_squared where,

$$\text{sigma\_b\_squared}_i = \frac{(\text{mu}_t * \text{omegai} - \text{mu}_i)^2}{\text{Omega}_i - (1 - \text{Omega}_i)}$$

**Step 6:** Find the location,  $\text{idx}$ , of the maximum value of  $\text{sigma\_b\_squared}$ . The maximum may extend over several bins, so average together the locations.

**Step 7:** If maximum is not a number, meaning that  $\text{sigma\_b\_squared}$  is all not a number, and then threshold is 0.

**Step 8:** If maximum is a finite number,

$$\text{threshold} = \frac{(\text{idx} - 1)}{(256 - 1)}$$

In a grayscale image there are 256 combinations of black and white colors where 0 means pure black and 255 means pure white. This image is converted to binary image by checking whether or not each pixel value is greater than 255-level (level, found by Otsu's Method). If the pixel value is greater than or equal to 255-level then the value is set to 1 i.e. white otherwise 0 i.e. black.

#### 2.1.2 Noise Removal

Noise removal is used to remove any type of unwanted bit-patterns, that may garbled the nature of output. Various filtering operations can be applied to remove noise e.g. Median Filter, Weiner filter etc.

#### 2.1.3 Smoothing

The objective of smoothing is to smooth shape of broken and/or noisy input characters. Some pixels are added in the image so that a smooth shape may be obtained.

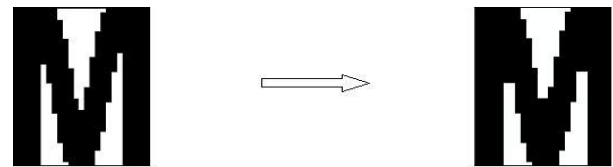


Figure 2: Applying thresholding & filtering operations on image.

#### 2.1.4 Skeletonization

Skeletonization is also called thinning. Skeletonization refers to the process of reducing the width of a line like object from many pixels wide to just single pixel. This process can remove irregularities in letters and in turn, makes the recognition algorithm simpler because they only have to operate on a character stroke, which is only one pixel wide. It also reduces the memory space required for storing the information about the input characters and no doubt, this process reduces the processing time too.

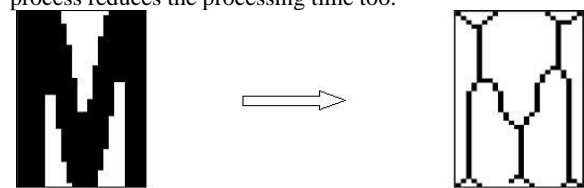


Figure 3: Applying Skeletonization on the filtered image

#### 2.1.5 Normalization

Normalization is a linear process. Consider an example that if the intensity range of the image is 100 to 230 and the required range is 0 to 150 then subtract 100 from each of pixel intensity that will create range 0 to 130. Then multiply each pixel intensity by 150/130 so that the range of 0 to 150 can be obtained. The process of Auto-normalization normalizes an image to the full dynamic range of the number system specified in the image file format. The regions produced by the normalization process will have the same constant dimensions [19]. Normalization methods aim to remove all types of variations during the writing and standardized data is obtained [23]. For example Size normalization is used to adjust the character size to a certain standard. Methods of character recognition may apply both horizontal and vertical size normalizations.

## 2.2 Step-2: Feature Extraction

Each character has some features, which play an important role in pattern recognition. Feature extraction describes the relevant shape information contained in a pattern so that the task of classifying the pattern is made easy by a formal procedure. In the area of image processing and pattern recognition, feature extraction is used for dimensionality reduction. The main goal of feature extraction is to obtain the most relevant information from the original data and represent that information in a lower dimensionality space. When the input data to an algorithm is too large and also may be redundant (much data, but not much information) then the input data will be transformed into a reduced representation set of features (also named features vector). A term Feature Extraction is termed that transforms the input data into the set of features. Features extracted should be chosen carefully. The features set will be used to extract the relevant information from the input data in order to perform the desired task using this reduced representation instead of the full size input [11, 12].

Feature extraction method used in this work is Fourier descriptor. Feature extraction is the name given to a family of procedures for measuring the relevant shape information contained in a pattern so that the task of classifying the pattern is made easy by a formal procedure. Among the different design issues involved in building a recognizing system, perhaps the most significant one is the selection of set of features. Feature extraction for exploratory data projection enables high-dimensional data visualization for better data structure understanding and for cluster analysis. In feature extraction for classification, it is desirable to extract high discriminative reduced-dimensionality features, which reduce the classification computational requirements.

Discrete Fourier transform (DFT) is itself a sequence rather than a function of continuous variable and it corresponds to equally spaced frequency samples of discrete time Fourier transform of a signal [23]. Fourier series representation of the periodic sequence corresponds to discrete Fourier transform of finite length sequence. So we can say that DFT is used for transforming discrete time sequence  $x(n)$  of finite length into discrete frequency sequence  $X[k]$  of finite length. This means that by using DFT, the discrete time sequence  $x(n)$  is transformed into corresponding discrete frequency sequence  $X[k]$ .

Function used in this work for fourier descriptor is `fft2()`.

$Y = \text{fft2}(X, m, n)$  truncates  $X$ , or pads  $X$  with zeros to create an  $m$ -by- $n$  array before doing the transform. The result is  $m$ -by- $n$ . `fft2(X)` can be simply computed as `fft(fft(X),.)`. This computes the one-dimensional DFT of each column  $X$ , then of each row of the result. The execution time for `fft` depends on the length of the transform. It is fastest for powers of two.

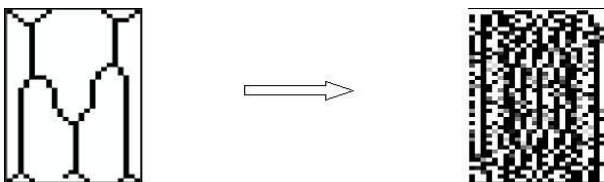


Figure 4: Feature extraction of Skeletonized image.

## 2.3 Step-3: Neural Network Implementation

### 2.3.1 Network Creation

A feed-forward backpropagation network is created. To create a feedforward network, `newff` function is used that takes three arguments and network object is returned. The first argument is a matrix of sample  $R$ -element input vectors. The second argument is a matrix of sample  $S$ -element target vectors. The sample inputs and outputs are used to set up network input and output dimensions and parameters. The third argument is an array containing the sizes of each hidden layer. (The output layer size is determined from the targets). More optional arguments can be provided. The fourth argument is a cell array that contains names of the transfer functions that may be used in each layer. The fifth argument is the name of the training function. If only three arguments are supplied, the default transfer function for hidden layers is `tansig` and the default for the output layer is `purelin`. The default training function is `trainlm`. `newff` command creates the network object and also initializes the weights and biases of the network; therefore the network is ready for training. The command used in this work is given as under:

```
net = newff(input, target, 26, { }, 'trainlm')
```

Initialization of the weights and biases has to be done before training a feedforward network. The `newff` command automatically initializes the weights. To reinitialize the weights, `init` command is used. This function takes a network object as input and returns a network object with all weights and biases initialized. A network can be initialized (or reinitialized) by using:

```
net = init(net);
```

Following parameters are used for training of Neural Networks:

No. of neurons in Input Layer: 26

No. of Layers taken: 2 (Input layer + hidden layer+ Output Layer)

No of epochs: 2000

Learning Rate Parameter: 0.01

Goal defined to achieve: 0.1

Transfer Function Used for Layer 1: "Logsig"

Transfer Function Used for Layer 2: "Tansig"

Adaption Learning Function: "Learnngd"

### 2.3.2 Training

The first part is the training phase, where we manually identify the correct class. The features extracted from the second step would serve as the data to train the neural network. Recognition of handwritten characters is a very complex problem. The capability of neural network to generalize and insensitive to the missing data would be very beneficial in recognizing handwritten characters. In this paper, for English handwritten character recognition in Feed Forward Multi-Layer Perceptron network (MLPN) with one hidden layer has been used. For training, back-propagation algorithm has been implemented [1].

#### 2.3.2.1 Back Propagation Algorithm

The algorithm for back propagation is given as under:

*Step 1: Initialization:* Assuming that no prior information is available, pick the synaptic weights and thresholds from a uniform distribution whose mean is zero and whose variance is chosen to make the standard deviation of the induced local fields of the neurons lie at the transition between the linear and saturated parts of the sigmoid activation function.

*Step 2: Presentation of Training examples:* Present the network with an epoch of training examples. For such example in the set, ordered in some fashion, perform the sequence of forward and backward computations.

*Step 3: Forward Computation:* Let a training example in the epoch be denoted by  $(x(n), d(n))$ , with the input vector  $x(n)$  applied to the input layer of sensory nodes and the desired response vector  $d(n)$  presented to the output layer of computation nodes. Compute the induced local fields and function signals of the network by proceeding forward through the network, layer by layer. The induced local field  $v_j(n)$  for neuron  $j$  in layer  $l$  is:

$$v_j(n) = \sum w_{ji}(n)y_i(n)$$

where  $y_i(n)$  is the output (function) signal of neuron  $i$  in the previous layer  $l-1$  at iteration  $n$  and  $w_{ji}(n)$  is the synaptic

weight of neuron  $j$  in layer  $l$  that is fed from neuron  $i$  in layer  $l-1$ . For  $i=0$ , we have  $y_0(n) = +1$  and  $w_{j0}(n) = b_j(n)$  is the bias applied to neuron  $j$  in layer  $l$ .

Assuming the use of a sigmoid function, the output signal of neuron  $j$  in layer  $l$  is

$$y_j = \varphi(v_j(n))$$

If neuron  $j$  is in the first hidden layer (i.e.  $l=1$ ), set

$$y_j(n) = o_j(n)$$

Compute the error signal:

$$e_j(n) = d_j(n) - o_j(n)$$

where  $d_j(n)$  is the  $j^{\text{th}}$  element of desired response vector  $d(n)$ .

**Step 4: Backward Computation:** Compute the  $\delta_j$  (i.e. local gradients) of the network, defined by

$$\delta_j(n) = e_j(n)\varphi'(v_j(n)) \text{ for neuron } j \text{ in output layer } L$$

$\delta_j(n) = \varphi'(v_j(n)) \sum \delta_k(n)w_{kj}(n)$  for neuron  $j$  in hidden layer  $l$ .

where the prime in  $\varphi'(\cdot)$  denotes differentiation with respect to the argument. Adjust the synaptic weights of the network in layer  $l$  according to the generalized delta rule:

$$w_{ji}(n+1) = w_{ji}(n) + \alpha [w_{ji}(n-1) + \eta \delta_j(n)y_i(n)]$$

where  $\eta$  is the learning rate parameter and  $\alpha$  is the momentum constant.

**Step 5: Iteration:** Iterate the forward and backward computations under step3 and 4 by presenting new epochs of training examples to the network until the stopping criterion is met.

For training purpose 11 sets of all 26 alphabets are taken with different handwriting styles. No. of epochs taken are 35 and 5000 to achieve the goal of 0.01 and 0.001 respectively (may be more than 5000 as convergence is not achieved) as shown in figure 5. As goal is 0.001, more epochs as well as more time is taken to achieve the goal.

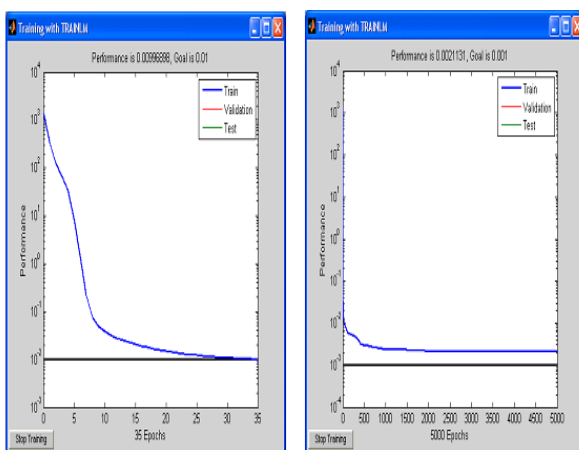


Figure 5: Neural Network Training

### 3. RESULT ANALYSIS

Offline handwritten English character sets are taken for recognition. The steps that are followed to obtain best accuracy and efficiency of input handwritten English character image from the recognition system. First of all,

training of system is done by using different training sample with different handwriting styles. And then system is tested for few of the given sample, and then result is measured. The data set was partitioned into two parts. The first part is used for training the system and the second was for testing purpose. For each character, feature were computed and stored for training the network. The table 1 displays the results obtained from the program. The convergence of the network is set at 0.01 for first experiment and 0.001 for second experiment, No. of training data is 11, and 1 bitmap file is taken for testing in which letters are arranged in the form of words. After recognition a text file is generated that gives the character in text format in a notepad file. The experiment results for training of neural network in the Matlab that includes Convergence objective, Learning Rate, Training method used, No. of training data, No. of testing data, number of epoch, time taken to execution of program and gradient, No. of characters found matched and accuracy in % are shown in table 1.

To get the accuracy % results are compared with the desired result. Difference between the two values gives the error. Percentage accuracy is found as follows:



$$\%Accuracy = \frac{\text{No. of characters found correctly}}{\text{Total No. of patterns}} * 100$$

Table 1. The experimental results using neural network analysis for recognition

| Experiment                              | First Experiment              | Second Experiment             |
|---|-------------------------------|-------------------------------|
| Convergence Objective                   | 0.01                          | 0.001                         |
| Learning rate                           | 0.01                          | 0.05                          |
| Training method used                    | trainlm (Levenberg-Marquardt) | trainlm (Levenberg-Marquardt) |
| No. of training data                    | 8 x 26 = 208                  | 8 x 26 = 208                  |
| No. of testing data                     | 3 x 26 = 78                   | 3 x 26 = 78                   |
| No. of epoch taken to converge          | 236                           | >5000                         |
| Time taken to execute                   | 8.85275 seconds               | 1024.06 seconds               |
| Gradient                                | 0.153241                      | 2.00145e-005                  |
| Total no. of patterns taken for testing | 3 x 26 = 78                   | 3 x 26 = 78                   |
| No. of characters found correctly       | 71                            | 73                            |
| Accuracy%                               | 91.02%                        | 93.58%                        |

### 4. CONCLUSION AND FUTURE SCOPE

Recognition approaches heavily depend on the nature of the data to be recognized. As neural network is used here for recognition of offline English character images and it has been seen that recognition increases, although at a slow rate.

Also some characters like I & J are similar, so the recognition system gives sometimes bad results for similar character. Also it is based on the handwriting style e.g. G may be written as  or . This may also create problem sometimes. There is sometimes result variation may be due to the number of character set used for training was reasonably low. As the network is trained with more number of sets, the accuracy of recognition of characters will increase definitely. It can be concluded that the work successfully does the character recognition. It has the limitation that it performs the training as well as testing at a slow rate.

The algorithm implemented is well suited for recognition of Hindi, Punjabi or any other language characters. It can be extended for the recognition of words, sentence and documents. Also further research may be done on the character images degraded or blurred due to noise or various reasons. More image processing techniques with better results can be applied for good results. We can improve the recognition accuracy of our system by tuning the features drawn through feature extraction method. One way to achieve this is to normalize the binary pattern obtained after pre processing of image and before computing the Fourier transform. Also feature extraction through PCA or ICA may perform better.

## 5. REFERENCES

- [1] Anita Pal, Dayashankar Singh, "Handwritten English Character Recognition Using Neural Network", *International Journal of Computer Science & Communication* Vol. 1, No. 2, pp. 141-144, July-December 2010.
- [2] C. Suresh kumar, Dr. T. Ravichandran, "Handwritten Tamil Character Recognition Using RCS Algorithm", *International Journal of Computer Applications (0975 – 8887)*, Volume 8, No. 8, October 2010.
- [3] Rashad Al-Jawfi, "Handwriting Arabic Character Recognition LeNet Using Neural Network", *The International Arab Journal of Information Technology*, Vol. 6, No. 3, July 2009
- [4] G. G. Rajput, Rajeswari Horakeri, Sidramappa Chandrakant, "Printed and Handwritten Kannada Numeral Recognition Using Crack Codes and Fourier Descriptors Plate", *International Journal of Computer Application (IJCA) on Recent Trends in Image Processing and Pattern Recognition (RTIPPR)*, pp 53-58, 2010.
- [5] G. G. Rajput, S. M. Mali, "Marathi Handwritten Numeral Recognition using Fourier Descriptors and Normalized Chain Code" in *International Journal of Computer Application(IJCA) on Recent Trends in Image Processing and Pattern Recognition (RTIPPR)*, pp 141-145, 2010.
- [6] Srinivasa Kumar DeviReddy, Settipalli Appa Rai, "Hand written character recognition using back propagation network", *Journal of Theoretical and Applied Information Technology*, 2005-2009.
- [7] R. Plamondon and S. N. Srihari, "On-line and off-line handwritten recognition: a comprehensive survey", *IEEE Transactions on PAMI*, Vol. 22(1), pp. 63–84, 2000.
- [8] Jean R. Ward and Theodore Kuklinski, "A Model for Variability Effects in Hand-writing Character Recognition Systems" in *IEEE Trans. Sys. Man. Cybernetics*, Vol.18, No. 3, pp 438-451, 1988.
- [9] Nafiz Arica and Fatos T. Yarman-Vural, "An Overview of Character Recognition Focused on Off-Line Handwriting", in *IEEE Trans. Sys. Man. Cybernetics*, Vol. 31, No. 2, pp.216-238, 2001.
- [10] Simon Haykin, *Neural Networks, A Comprehensive Foundation*, Pearson Education, Inc., 2004.
- [11] Rafael C. Gonzalez, Richard E. Woods, *Digital Image Processing*, 2nd edition, Pearson Education.
- [12] Mark S. Nixon, Alberto S. Aguado, *Feature Extraction and Image Processing*, Newnes Publisher, 2002 ISBN 0 7506 5078 8.
- [13] Cheng-Lin Liu, "Normalization-Cooperated Gradient Feature Extraction for Handwritten Character Recognition", *IEEE Transaction on pattern analysis and machine intelligence*, vol. 29, no. 8, Aug. 2007.
- [14] Hadar I. Avi-Itzhak, Thanh A. Diep, and Harry Garland, "High Accuracy Optical Character Recognition Using Neural Networks with Centroid Dithering", *IEEE Transactions on pattern analysis and machine intelligence*, vol. 17, no. 2, Feb.1995.
- [15] Youfu Wu, Yongwu Wu, Gang Zhou, Jing Wu, "Recognizing Characters Based on Gaussian-Hermite Moments and BP Neural Networks", *International Conference on Intelligent Computation Technology and Automation*, ISBN 978-0-7695-4077-1, 2010.
- [16] Birijesh K. Verma, "Handwritten Hindi Character Recognition Using Multilayer Perceptron and: Radial Basis Function Neural Networks", *IEEE International conference on neural networks*, Vol. 4, pp. 2111-2115, Nov.1995.
- [17] Janusz A. Starzyk and Nasser Ansari, "Feedforward Neural Network for Handwritten Character Recognition", *IEEE symposium on circuit and systems*, 1992.
- [18] J. SUTHA, N. RAMARAJ," Neural Network Based Offline Tamil Handwritten Character Recognition System", *International Conference on Computational Intelligence and Multimedia Applications*, pp. 446-450, IEEE 2007.
- [19] Deepayan Sarkar, Report on Optical Character Recognition using Neural Network, <http://homepages.cae.wisc.edu/~ece539/project/f03/sarkar-rpt.pdf>
- [20] Wikipedia on Normalization (Image Processing), [http://en.wikipedia.org/wiki/Normalization\\_\(image\\_processing\)](http://en.wikipedia.org/wiki/Normalization_(image_processing))
- [21] Wikipedia on Optical Character Recognition, [http://en.wikipedia.org/wiki/Optical\\_character\\_recognition](http://en.wikipedia.org/wiki/Optical_character_recognition)
- [22] Diego Orlando, Optical Character Recognition, <http://www.mathworks.com/matlabcentral/fileexchange/18169-optical-character-recognition-ocr>
- [23] W. Guerfaii and R. Plamondon, "Normalizing and Restoring On-line Handwriting", *Pattern Recognition*, Vol. 26, No. 3, pp. 418-431, 1993.
- [24] Image Processing and Neural Network Toolbox help of MatLab R2010.
- [25] S. K. Hasnain, Muhammad Samiullah Awan, "Recognizing Spoken Urdu Numbers Using Fourier Descriptor and Neural Networks with Matlab", *Second International Conference on Electrical Engineering*, IEEE ISBN 978-1-4244-2293-7, March 2008.