NEURAL NETWORK BASED APPROACH FOR RECOGNITION FOR DEVANAGIRI CHARACTERS

Neha Sahu[1],Shipra Gupta[2],ShilpaSrivastava Khare[3]

[1,2]ECE Department, Amity School of Engineering and Technology,AMITY University, Noida, U.P (India)
[3]Assistant professor, ECE Department, AMITY University, Noida, U.P, (India)

ABSTRACT

The development of a Character recognition system for Devnagri is difficult because (i) there are about 350 basic, modified ("*matra"*) and compound character shapes in the script and (ii) the characters in a words are topologically connected. Here focus is on the recognition of offline handwritten Hindi characters that can be used in common applications like bank cheques, commercial forms, government records, bill processing systems, Postcode Recognition, Signature Verification, passport readers, offline document recognition generated by the expanding technological society. Challenges in handwritten characters recognition lie in the variation and distortion of offline handwritten An approach using Artificial Neural Network is considered for recognition of Handwritten Hindi Character Recognition.

Keywords: *Devanagari script, Neural Network, Feature Extraction, Online Character Recognition, Offline Character recognition.*

I. INTRODUCTION

Hindi handwritten character recognition is the one of the major problem in today's world. Typed Hindi characters can be easily recognized by computer machine. But Hindi handwritten characters are not recognized efficiently and accurately by computer machine. Many researches have been done to recognize these characters and many algorithms have been proposed to recognize characters. Many types of software are in the market for optical Hindi character recognition. For recognizing characters, many processes have to be performed. No single process or single machine can perform that recognition. Artificial neural networks can be used for recognition of characters due to the simplicity of their design and their universality. Hindi character recognition is becoming more and more important in the modern world. It helps human ease their jobs and solve more complex problems. The problem of recognition of hand-printed characters is still an active area of research. With ever increasing requirement for office automation, it is imperative to provide practical and effective solutions. It has been observed that all sorts of structural, topological and statistical information about the characters does not lend a helping hand in the recognition process due to different writing styles and moods of persons at the time of writing[2][4][5]. Mainly, attention is focused on recognition of hand-printed Hindi characters. Limited variations in shapes of character are considered.

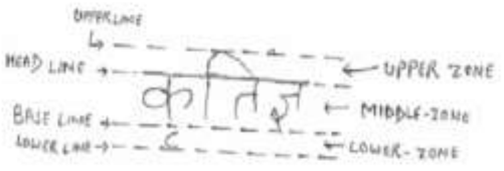
For more than 30 years, researchers have been working on handwritten recognition. Over the few past years, the numbers of companies involved in research on handwritten recognition are increasing continually. Handwritten recognition is not a new technology, but it has not gained public attention until recently. The ultimate goal of designing a handwritten recognition system with an accuracy rate of 100% is quit illusionary, because even human beings are not able to recognize every handwritten text without any confusion. It can be seen that most of the people can not even read their own notes. Therefore there is an obligation for a writer to write clearly[5][6].

1.1 DEVANAGARI SCRIPT

Hindi is world's third most commonly used language after Chinese and English, and there are approximately 500 billion people all over the world who speak and write in Hindi. It is the basic script of many languages in India, such as Hindi and Sanskrit.Many other languages use close variants of this script. Although Sanskrit is an ancient language and no longer spoken, written material still exist. It is very expressive language, which has been influenced and enriched by Dravidian, Turkish, Farsi, Arabic, Portuguese and English. Thus research on Devanagari script mainly Hindi language attracts a lot of interest[8][9].

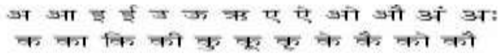
1.1.1 DEVANAGARI SCRIPT IDENTIFICATION

Instead of describing what type of features can be used to identify Devanagari script words from document images, we examine the appearance of Devanagari script. Regular Hindi word can be divided into three Zones i.e. Upper, Middle, Lower Zone. Example is shown below where three Zones are illustrated. The Upper Zone and middle Zone are always separated by the header line called shirorekha. The Upper Zone contains the modifiers, and Lower Zone contains lower modifiers. In Hindi word, Upper and Lower Zone are not always necessary, but depend on Upper and lower modifiers[11].



1.1.2 DEVANAGARI SCRIPT OVERVIEW

Its basic set of symbols consists of 34 consonants and 18 vowels, and though Devanagari has a native set of symbols for numerals, Arabic numbers are now commonly used. Character Recognition for Devanagari is highly complex due to its rich set of conjuncts. Devanagari is written from left to right along a horizontal line. Its basic set of symbols consists of 34 consonants or ('vyanjan') and 18 vowels ('svar'). Characters are joined by a horizontal bar that creates an imaginary line by which Devanagari text is suspended, and no spaces are used between words. A single or double vertical line called 'Purn Viram' was traditionally used to indicate the end of phrase or sentence. Devanagari also has a native set of symbols for numerals, though Arabic numbers are typically used [12][13]. In part, Devanagari owes its complexity to its rich set of conjuncts. The language is partly phonetic in that a word written in Devanagari can only be pronounced in one way, but not all possible pronunciations can be written perfectly. A syllable ("akshar") is formed by a vowel alone or any combination of consonants with a vowel. Each vowel exceptcorrespond to a modifier symbol. In Hindi when consonant are combined with other consonant, the consonant with vertical bar may appear as a half form. Except for the character, the half forms of consonant are the left part of original consonant with vertical bar. Here is a sample set of non-compound Devanagari characters. [15].



अ आ इ ई उ ऊ ऋ ए ऐ ओ औ अं अः
क का कि की कु कू कृ के कै को कौ

Figure 1.1.2.1 : Non compound Devanagari Charters

Some characters have upper and lower modifiers. Here is a sample of Devanagari modifiers.

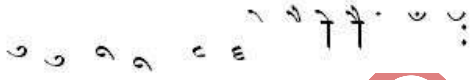
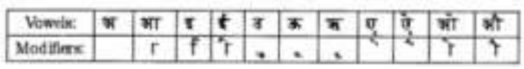


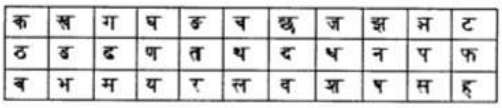
Figure 1.1.2.2: Devanagari Modifiers

Obviously, these modifiers make Character Recognition with Devanagari script very challenging. HDCR system is further complicated by compound characters that make character separation and identification very difficult.



Vowels:	अ	आ	इ	ई	उ	ऊ	ऋ	ए	ऐ	ओ	औ
Modifiers:		ा	ि	ी	ु	ू	ृ	े	ै	ो	ौ

Figure 1.1.2.3 : Vowels and corresponding modifiers



क	ख	ग	घ	ङ	च	छ	ज	झ	ञ	ट
ठ	ड	ढ	ण	त	थ	द	ध	न	प	फ
ब	भ	म	य	र	ल	व	श	ष	स	ह

Figure 1.1.2.4: Consonant

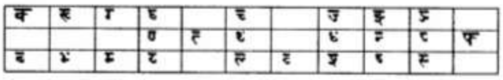


Figure 1.1.2.5 : Half form of Consonant with vertical bar

1.2 TYPES OF CHARACTER RECOGNITION SYSTEM

The constant development of computer tools leads to a requirement of easier interfaces between the man and the computer. Character Recognition is one way of achieving this. A Character Recognition deal with the problem of reading offline handwritten character i.e. at some point in time (in mins, sec, hrs) after it has been written[6][7]. However recognition of unconstrained handwritten text can be very difficult because characters cannot be reliably isolated especially when the text is cursive handwriting. They are classified as the following two types

i) Online Character Recognition
ii) Offline Character Recognition

1.2.1 ONLINE CHARACTER RECOGNITION

In case of online character recognition, there is real time recognition of characters. Online systems have better information for doing recognition since they have timing information and since they avoid the initial search step of locating the character as in the case of their offline counterpart. Online systems obtain the position of the pen as a function of time directly from the interface. Offline recognition of characters is known as a challenging problem because of the complex character shapes and great variation of character symbols written in different modes[8].

1.2.2 OFFLINE CHARACTER RECOGNITION

In case of offline character recognition, the typewritten/handwritten character is typically scanned in form of a paper document and made available in the form of a binary or gray scale image to the recognition algorithm. Offline character recognition is a more challenging and difficult task as there is no control over the medium and instrument used. The artifacts of the complex interaction between the instrument medium and subsequent operations such as scanning and binarization present additional challenges to the algorithm for the offline character recognition. Therefore offline character recognition is considered as a more challenging task then its online counterpart.

The steps involved in character recognition after an image scanner optically captures text images to be recognized is given to the recognition algorithm.

The major difference between Online and Offline Character Recognition is that Online Character Recognition has real time contextual information but offline data does not [38].This difference generates a significant divergence in processing architectures and methods.

II IMPLEMENTATION

2.1 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.

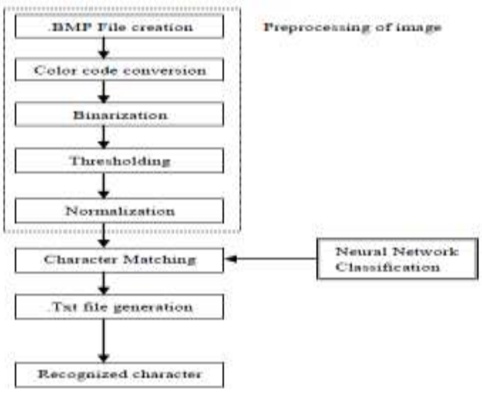


Figure 1: Block diagram for handwritten character recognition

2.2 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:

2.2.1 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

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[9][11].

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.

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[12].

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[14].

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].

III NEURAL NETWORK

Neural networks are composed of simple elements operating in parallel. These elements are inspired by biological nervous systems. As in nature, the network function is determined largely by the connections between elements. We can train a neural network to perform a particular function by adjusting the values of the connections (weights) between elements. Commonly neural networks are adjusted, or trained, so that a particular input leads to a specific target output. Such a situation is shown in fig 2.1.1. There, the network is adjusted, based on a comparison of the output and the target, until the network output matches the target. Typically many such input/target pairs are used, in this supervised learning, to train a network[10].

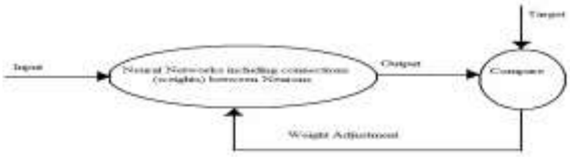

Target
Input
Neural Networks including connections (weights) between Neurons
Output
Compare
Weight Adjustment

Figure 3.1: Block Diagram of Neural Network

Neural networks have been trained to perform complex functions in various fields of application including pattern recognition, identification, classification, speech, vision and control systems. Today neural networks can be trained to solve problems that are difficult for conventional computers or human beings. The supervised training methods are commonly used, but other networks can be obtained from unsupervised training techniques or from direct design methods. Unsupervised networks can be used, for instance, to identify groups of data. Certain kinds of linear networks and Hopfield networks are designed directly. In summary, there are a variety of kinds of design and learning techniques that enrich the choices that a user can make[13][15].

The field of neural networks has a history of some five decades but it has been found that solid application only in the past fifteen years and the field is still developing rapidly. Thus, it is distinctly different from the fields of control systems or optimization where the terminology, basic mathematics, and design procedures have been firmly established and applied for many years.

3.2 NEURAL NETWORK ARCHITECTURE

Basically the neural network consists of:

- Neurons
- Interconnection among neurons

Neurons: Similar to the human brain neuro

transports the incoming information on their outgoing connections to the other neurons.

Weights: are called weights. The following information is simulated with specific values stored in those weights.

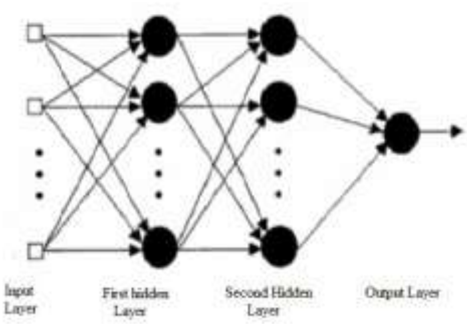


Figure 3.2 : **Architecture of Neural networkshowing different Layers**

3.3 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: dPresent 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[12[13].

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 vj(n) for neuron j in layer l is:

$$\mathbf{v_j(n)=\sum w_{ji}(n)\ y_i(n)} \qquad \mathbf{3.3.1}$$

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

$$\mathbf{y_j = \Phi_(v_j\ \ (n))} \qquad \mathbf{3.3.2}$$

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

$$\mathbf{y_i(n) = o_i\,(n)} \qquad \mathbf{3.3.3}$$

Compute the error signal:

$$\mathbf{e_j(n) = d_j(n) – o_j(n)} \qquad \mathbf{3.3.4}$$

where dj(n) is the jth element of desired response vector d(n).

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

$$\mathbf{\delta_j\ (n) = e_j(n)\ \Phi'(v_j(n))} \qquad \mathbf{3.3.5}$$

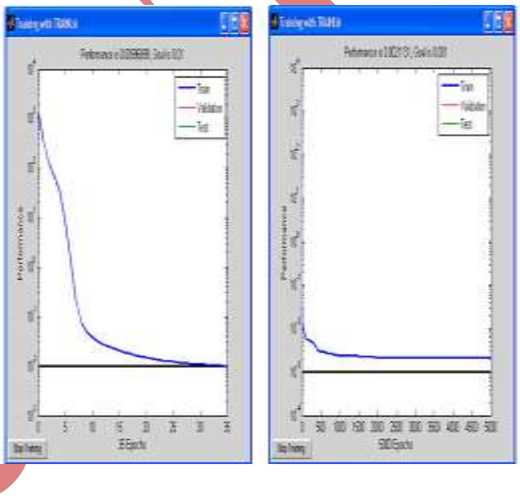
$$\mathbf{\delta_j(n) = \varphi'(v_{j(n)})\ \sum \delta_k(n) w_{kj}(n)} \qquad \mathbf{3.3.6}$$

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

$$\mathbf{w_{ji}(n+1) = w_{ji}(n) + \alpha[w_{ji}(\ n\text{-}1) + \eta\delta_j(n)y_j(n)]} \qquad \mathbf{3.3.7}$$

where η is the learning rate parameter and α 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[19].



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[18]. 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:

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

Experiment	Data
Convergence Objective	0.01
Learning rate	0.01
Training method used	trainlm (Levenberg-Marquardt)
No. of training data	8 x 26 = 208
No. of testing data	3 x 26 = 78
No. of epoch taken to converge	236
Time taken to execute	8.85275 seconds
Gradient	0.153241
Total no. of patterns taken for testing	3 x 26 = 78
No. of characters found correctly	71
Accuracy%	91.02%

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

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.

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

[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 Thedore 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)

[21] Wikipedia on 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.