*Singh et al., International Journal of Advanced Research in Computer Science and Software Engineering 4(6), June - 2014, pp. 866-871*
**Volume 4, Issue 6, June 2014** **ISSN: 2277 128X**
**International Journal of Advanced Research in Computer Science and Software Engineering**
**Research Paper**
**Available online at: www.ijarcsse.com**

# Handwritten Character Recognition Using Classifier

**Neha Singh***
CSE Department,
Azad Institute of Engg. India

**Mr. Sanjay Sachan**
CSE Department,
Azad Institute of Engg., India

***Abstract— Character recognition comes into play when we want to recognize the hand written characters in a perticular natural language. It can be done using various types of method and algorithm that are already defined.Character recognition is essentially a pattern recognition problem and has been around for years now. Although there are implementation of hand written characters in many natural languages and with many different algorithms but here we are doing the hindi charater recognition.***

***Hand written charter recognition finds its use in todays handheld devices like phones, pda, tablet computers where user can input characters through his or her fingers or a stylus and the system recognizes the character.***

***We will consider here the neural network and the classifier based recognition methods for our character recognition problem.***

***Keywords— Hindi character recognition, neural networks, multilayer perceptron, error backpropagation, radial basis function.***

## I. INTRODUCTION

Pattern recognition is assigning a label to a given input value. This generally lies in the field of Machine Learning. Pattern recognition can be like classification that generally breaks or divides the input into a set of classes. It can be applied to a vast variety of problems that include facial recognition, object recognition, Character recognition etc.

Optical character recognition (OCR) is computer software designed to translate images of typewritten or handwritten text into machine-editable text encoded in a standard encoding scheme (e.g. ASCII or Unicode) [1]. In the 1950s, the early solutions for recognizing typewritten English text started to appear. And now, there are many solutions for accurately recognizing typewritten Latin text. Moreover, modern operating systems include built-in support for OCR.

Although the progress in developing OCR solutions for the Arabic language is slower than the progress in developing solutions for Latin and Asian languages, there has been some success in developing solutions for recognizing typewritten Arabic text [2]. There are even some commercial products for this application that offer more than 99% accuracy in some cases.

Over the years several algorithms have been devised or developed for pattern recognition some of which are more recent like the neural network approach and deep learning approach. Pattern recognition algorithms generally aim to provide a reasonable answer for all possible inputs and to perform "most likely" matching of the inputs, taking into account their statistical variation. This is opposed to pattern matching algorithms, which look for exact matches in the input with pre-existing patterns. Examples of which are hand written character recognition and optical character recognition respectively.

Pattern recognition is generally categorized according to the type of learning procedure used to generate the output value. Supervised learning assumes that a set of training data (the training set) has been provided, consisting of a set of instances that have been properly labelled by hand with the correct output. A learning procedure then generates a model that attempts to meet two sometimes conflicting objectives: Perform as well as possible on the training data, and generalize as well as possible to new data (usually, this means being as simple as possible, for some technical definition of "simple", in accordance with Occam's Razor, discussed below). Unsupervised learning, on the other hand, assumes training data that has not been hand-labelled, and attempts to find inherent patterns in the data that can then be used to determine the correct output value for new data instances. A combination of the two that has recently been explored is semi-supervised learning, which uses a combination of labelled and unlabeled data (typically a small set of labelled data combined with a large amount of unlabeled data). Note that in cases of unsupervised learning, there may be no training data at all to speak of; in other words, the data to be labelled is the training data.

### *A. Probabilistic classifiers*

Many common pattern recognition algorithms are probabilistic in nature, in that they use statistical inference to find the best label for a given instance. Unlike other algorithms, which simply output a "best" label, often probabilistic algorithms also output a probability of the instance being described by the given label. In addition, many probabilistic algorithms output a list of the N-best labels with associated probabilities, for some value of N, instead of simply a single best label. When the number of possible labels is fairly small (e.g., in the case of classification), N may be set so that the probability of all possible labels is output. Probabilistic algorithms have many advantages over non-probabilistic algorithms:

They output a confidence value associated with their choice. (Note that some other algorithms may also output confidence values, but in general, only for a probabilistic algorithm is this value mathematically grounded in probability theory. Non-probabilistic confidence values can in general not be given any specific meaning, and only used to compare against other confidence values output by the same algorithm.)

Correspondingly, they can abstain when the confidence of choosing any particular output is too low.Because of the probabilities output, probabilistic pattern-recognition algorithms can be more effectively incorporated into larger machine-learning tasks, in a way that partially or completely avoids the problem of error propagation.

### *B. Feature Variables*

Feature selection algorithms, attempt to directly prune out redundant or irrelevant features. A general introduction to feature selection which summarizes approaches and challenges has been given. The complexity of feature-selection is, because of its non-monotonous character, an optimization problem where given a total of n features the power set consisting of all 2n -1 subsets of features need to be explored. The Branch-and-Bound algorithm does reduce this complexity but is intractable for medium to large values of the number of available features n.

Techniques to transform the raw feature vectors (feature extraction) are sometimes used prior to application of the pattern-matching algorithm. For example, feature extraction algorithms attempt to reduce a large-dimensionality feature vector into a smaller-dimensionality vector that is easier to work with and encodes less redundancy, using mathematical techniques such as principal components analysis (PCA). The distinction between feature selection and feature extraction is that the resulting features after feature extraction has taken place are of a different sort than the original features and may not easily be interpretable, while the features left after feature selection are simply a subset of the original features.

### *C. Approaches*

The first pattern classifier – the linear discriminant presented by Fisher – was developed in the Frequentist tradition. The frequentist approach entails that the model parameters are considered unknown, but objective. The parameters are then computed (estimated) from the collected data. For the linear discriminant, these parameters are precisely the mean vectors and the Covariance matrix. Also the probability of each class is estimated from the collected dataset. Note that the usage of 'Bayes rule' in a pattern classifier does not make the classification approach Bayesian.

Bayesian statistics has its origin in Greek philosophy where a distinction was already made between the 'a priori' and the 'a posteriori' knowledge. Later Kant defined his distinction between what is a priori known – before observation – and the empirical knowledge gained from observations. In a Bayesian pattern classifier, the class probabilities can be chosen by the user, which are then a priori. Moreover, experience quantified as a priori parameter values can be weighted with empirical observations – using e.g., the Beta- (conjugate prior) and Dirichlet-distributions. The Bayesian approach facilitates a seamless intermixing between expert knowledge in the form of subjective probabilities, and objective observations.

Building or training a classifier is essentially statistical inference. This means that an attempt is made to identify stochastic (often unknown) relations between feature variables and the categories to be predicted. For example, the influence of increased cholesterol on the risk of a heart attacks for a patient, within the next year. Which other variables besides the current cholesterol level determine this risk? The two categories to 'predict' by a classifier are 'heart attack likely', or 'heart attack unlikely'.

The theoretically optimal classifier is called the Bayes classifier. It minimizes the loss-function or risk. When all types of misclassifications are associated with equal losses (outcome A becomes B is as undesired as when outcome B becomes A), the Bayes classifier with the minimal error rate (on a test set) is the optimal one for the classification task. In general, it is unknown what is the optimal classifier type and true parameters. However, bounds on the optimal Bayes error rate have been derived. For, for example, the K-nearest neighbour classifier theoretical results are derived that bound the error rate, in relation to the optimal Bayes error rate.

## II. NEURAL NETWORK

We have demonstrated the application of MLP networks to the handwritten Hindi character recognition problem. The skeletonised and normalized binary pixels of Hindi characters as well as features of these characters were used as the inputs of the MLP networks. The networks with binary pixel inputs were better than the networks with features inputs, for generalization i.e., the performance on test set. The results show that the RBF networks offer shorter training time at the expense of greater memory and longer response time for testing examples. The RBF networks were slightly poorer in recognition accuracy than the MLP networks. In our further research work, we would like to improve the recognition accuracy of the both networks for Hindi character recognition by using more training samples written by one person and by improving our feature extraction system. Finally, we would like to develop an automatic system for handwritten Hindi text recognition.

The design of artificial neural network (ANN) has been inspired by the biological research on how the human's brain works.A neural network is an interconnected network of simple processing elements, e.g. scaling and filtering. The processing elements interact along paths of variable connection strengths which when suitably adapted can collectively produce complex overall desired behaviour.

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 (neurons) 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 solve 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.

The most basic components of neural networks are modeled after the structure of the brain. Some neural network structures are not closely to the brain and some does not have a biological counterpart in the brain. However, neural networks have a strong similarity to the brain and therefore a great deal of the terminology is borrowed from neuroscience.

The Hindi language consists of 49 characters (13 vowels, 36 consonants) and is written from left to right. A set of handwritten Hindi characters is shown in Figure 1.

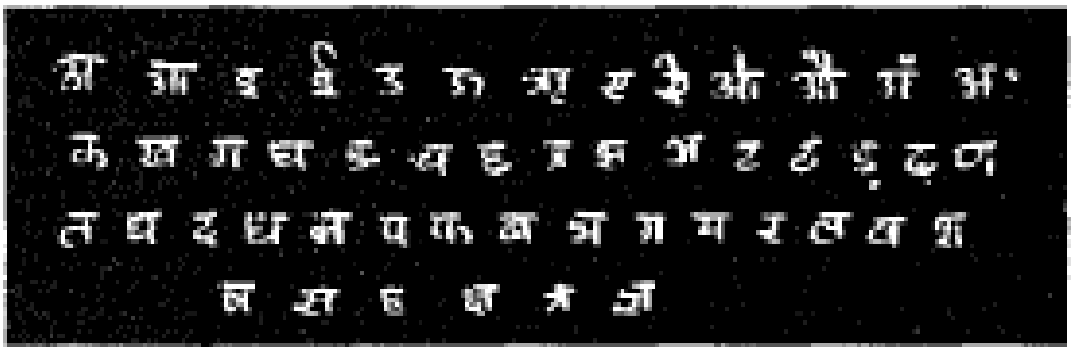

Fig.1 A set of handwritten Hindi characters

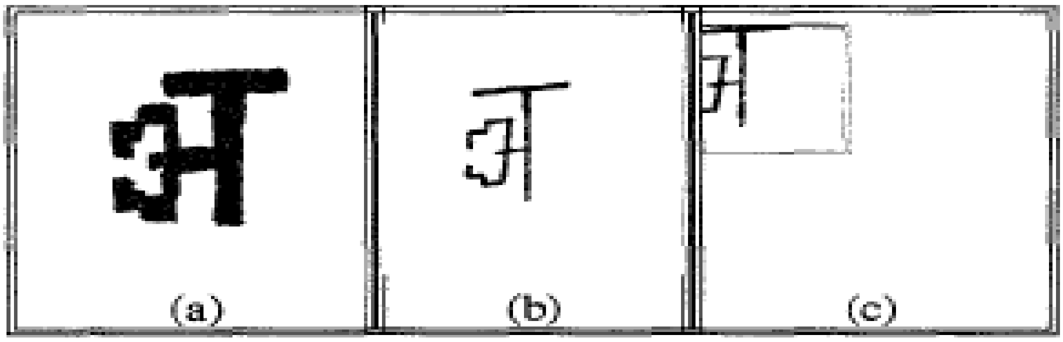


Fig.2 Skeletonization and normalization of a Hindi character.

**Scanning and skeletonization**

Each character was scanned at 300 pixel per inch using Scanner HP-Scan Jet 11. The scanned character was converted into 4096 (64x64) binary pixels. The skeletonization process was used to binary pixel image and the extra pixels which were not belong to the backbone of the character, were deleted and the broad strokes were reduced to thin lines.

Skeletonization is illustrated in Figure 2. A character before and after skeletonization is shown in Figure 2a and 2b respectively. After skeletonization of a character, we used a normalization process, which normalized 64x64 pixel skeletonized character into 32x32 pixel character and it was shifted to the left and upper comer of 64x64 pixel window. The final skeletonized and normalized character is shown in Figure 2c, which was used as an input of the neural networks. The skeletonization and normalization processes were used for each character.

**Feature Extraction**

Hindi characters have special features; some of them are shown in Figure 3. Hindi character is formed by a number of components of simple structures. Some of the compoients are formed by basic components. Eventually, those components are disassembled into a number of strokes (dots, lines, curves, loops). Hence a Hindi character can be described in a three-level representation hierarchy. An example of this representation is given in Figure 4. We developed a separate system for Hindi character feature extraction.

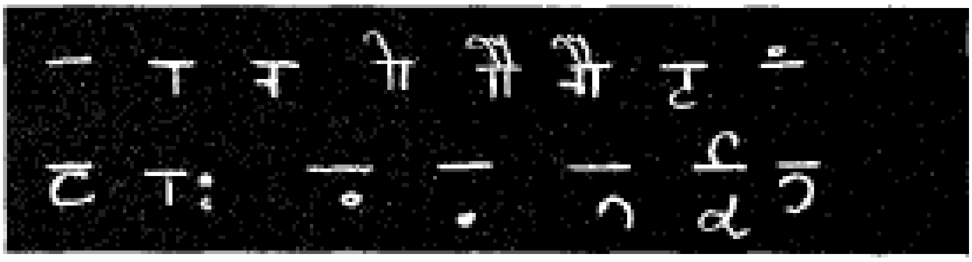

Fig.3 Some characteristics of Hindi characters

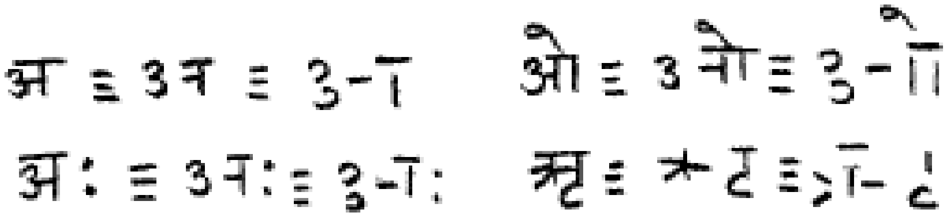

Fig.4 Hindi character representation 3-level hierarchy

## III. THE MLP CLASSIFIER

The MLP is a special kind of Artificial Neural Network (ANN). ANNs are developed to replicate learning and generalization abilities of human’s behavior with an attempt to model the functions of biological neural networks of the human brain.

Architecturally, an MLP is a feed-forward layered network of artificial neurons. Each artificial neuron in the MLP computes a sigmoid function of the weighted sum of all its inputs. An MLP consists of one input layer, one output layer and a number of hidden or intermediate layers, as shown in Fig 5. The output from every neuron in a layer of the MLP is connected to all inputs of each neuron in the immediate next layer of the same. Neurons in the input layer of the MLP are all basically dummy neurons as they are used simply to pass on the input to the next layer just by computing an identity function each. The multilayer perceptrons neural networks with the EBP algorithm have been applied to the wide variety of problems [ 15]-[19]. We used a two-layer perceptron ie., single hidden layer and an output layer. A structure of MLP network for Hindi character recognition is shown in Figure

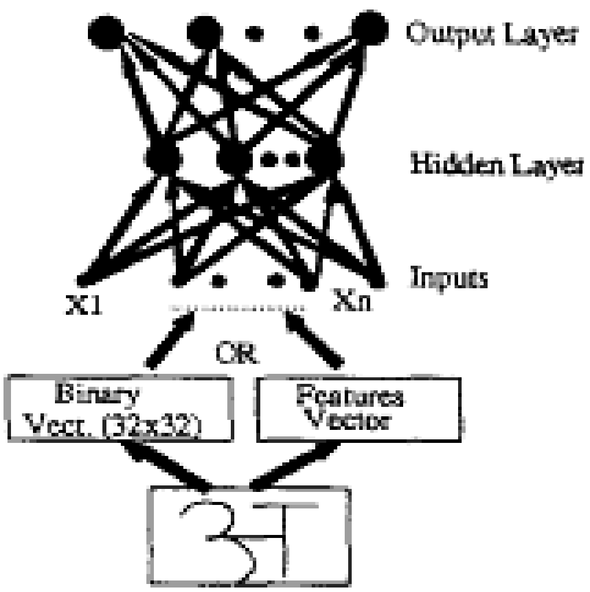


Fig.5 . Multilayer perceptron network

The activation function of a neuron j can expressed as :

$f_j(x) = \frac{1}{1+e^{-net}}$ and $net = \sum_i w_{ij} o_i$

where oi is the output of unit i, wd is the weight from unit i to unit j. The generalized delta rule algorithm [5] was used to update the weights of the neural network in order to minimize the cost function:

$w_{ij}(n+l) = w_{ij}(n) + \Delta w_{ij}(n)$
$\Delta w_{ij}(n) = \eta\delta x_j+\alpha(w_{ij}(n)-w_{ij}(n-l))$

where η is the learning rate, α is the momentum, wij(n) is the weight from hidden node i or from an input to node j at nth iteration, xi is either the output of unit i or is an input, and δj is an error term for unit j. If unit j is an output unit, then $\delta_j=o_j( l-o_j)(d_j-o_j)$

If unit j is an internal hidden unit, then

$$\delta_j = o_j(1-o_j)\sum_k \delta_k w_{kj}$$

## IV. IMPLEMENTATION AND RESULTS

Each MLP network uses two-layer feedforward network [2] with nonlinear signoidal functions. Many experiments with the various number of hidden units for each network were carried out. The output layer contained 49 neuron (one for each character). Each network was trained using the EBP algorithm as described and was trained until mean square error between the network output and desired output falls below 0.05. The weights were updated after each pattern presentation. The learning rate and momentum were 0.2 and 0.1 respectively. The results are shown in Table 1. The Root-Mean-Square (RMS) [2, 51 errors vs. the number of iterations, for MLP networks with two types of inputs (global character, features of a character).

TABLE 1 RESULTS FOR HANDWRITTEN HINDI CHARACTER RECOGNITION USING MLP

| **Input of MLP** | **No of hidden units** | **No of iterations** | **Training time** | **Classification time** | **Performance on training set (%)** | **Performance on test set** |
|---|---|---|---|---|---|---|
| 32x32 pixel global input | 12 | 200 | 5071 | 57 | 100.0 | 75.8 |
| | 24 | 200 | 6564 | 114 | 100.0 | 85.0 |
| | 36 | 200 | 8190 | 171 | 100.0 | 84.1 |
| Feature input | 12 | 350 | 597 | 8 | 100.0 | 54.1 |
| | 24 | 350 | 1332 | 17 | 100.0 | 70.0 |
| | 36 | 350 | 2155 | 25 | 100.0 | 70.0 |

## V. CONCLUSION AND FUTURE WORK

We have demonstrated the application of MLP networks to the handwritten Hindi character recognition problem. The skeletonised and normalized binary pixels of Hindi characters as well as features of these characters were used as the inputs of the MLP networks. The networks with binary pixel inputs were better than the networks with features inputs, for generalization i.e., the performance on test set. The results show that the RBF networks offer shorter training time at the expense of greater memory and longer response time for testing examples. The RBF networks were slightly poorer in recognition accuracy than the MLP networks. In our further research work, we would like to improve the recognition accuracy of the both networks for Hindi character recognition by using more training samples written by one person and by improving our feature extraction system. Finally, we would like to develop an automatic system for handwritten Hindi text recognition.

## REFERENCES

[1] R. K. Young, Wavelet Theory and Its Applications, Kluwer Acadamic Publishers, Boston, MA, 1993
[2] P. Goupillard, A. Grossmann, and J. Morlet, “Cycle-Octave and Related Transforms in Seismic Signal Analysis,’ Geoexploration, v23, pp. 85-102, 1984.
[3] J. M. Combes, A. Grossmann, and Ph. Tchamitchian, Eds., Wavelets: Time-Frequency Methods and Phase Space, 2nd edition, Springer-Verlag, New York, NY, 1989.
[4] A. Grossmann and J. Morlet, “Decomposition of Hardy Functions into Square Integrable Wavelets of Constant Shape,” SIAM J. Math. Anal., v15, n4, pp. 723-36, 1984.
[5] L. G. Weiss, “Wavelets and Wideband Correlation Processing,” IEEE Signal Processing Mag., v11, n1, pp 13-32, Jan. 1994.
[6] Rafael C. Gonzalez and Richard E. Woods. Digital Image Processing, Second Edition. Prentice Hall,2002.
[7] St´ephane Mallat. A theory for multiresolution signal decomposition: The wavelet representation. IEEE Trans. Pattern Anal. Mach. Intell., 11(7):674–693,1989.

 

[8] Meng Shi, Yoshiharu Fujisawa, Tetsushi Wakabayashi, and Fumitaka Kimura. Handwritten numeral recognition using gradient and curvature of gray scale image. Pattern Recognition, 35(10):2051–2059, 2002.

[9] Cheng-Lin Liu and Masaki Nakagawa. Evaluation of prototype learning algorithms for nearestneighbor classifier in application to handwritten character recognition. Pattern Recognition, 34(3):601– 615, 2001.

[10] AssCheng-Lin Liu, Kazuki Nakashima, Hiroshi Sako, and Hiromichi Fujisawa. Handwritten digit recognition: benchmarking of state-of-the-art techniques. Pattern Recognition, 36(10):2271–2285, 2003.

[11] Weipeng Zhang,Yuan Yan Tang, Yun Xue Handwritten Character Recognition Using Combined Gradient and Wavelet Feature, 1-4244-0605-6/06/$20.00 ©2006 IEEE.

[12] R. Lippmann, “An Introduction to Computing with Neural Networks,” IEEE ASSP Mag., pp 4-22, April 1987.

[13] W. S. McCulloch and W. Pitts. A logical calculus of ideas immanent in nervous activity. Bulletin of Mathematical Biophysics, 5:115-133, 1943. Reprinted in Anderson and Rosenfeld, 1988

[14] D. R. Hush and B. G. Horne, “Progress in Supervised Neural Networks: What’s New Since Lippmann?,” IEEE Signal Proc. Mag., pp 8-39, Jan. 1993.

[15] J. Hertz, A. Krogh and R. Palmer, "Introduction to the theory of neural computation," Addison-Wesley Publishing Company, USA, 1991.

[16] K. Yamada and H. Kami, "Handwritten numeral recognition by multilayered neural network with improved learning algorithm," IJCNN Washington DC, vol. 2, pp. 259-266, 1989.

[17] P. Morasso, "Neural models of cursive script handwriting," IJCNN, WA, vol. 2, pp. 539-542, June 1989.

[18] R.O. Duda and P.E. Hart, "Pattern classification and scene analysis," Wiley, New York, 1973.

[19] S.J. Smith and M.O. Baurgoin, "Handwritten character classification using nearest neighbor in large database," IEEE Trans. On Pattem and Machine Intelligence, vol. 16, no 10, pp. 915-919, Oct. 1994.

 