

An Intelligent Feature Analyzer for Handwritten Character Recognition

Jalal Mahmud

*Department of Computer Science
Stony Brook University, NY 11794 USA.*

E-mail: jmahmud@cs.sunysb.edu

Abstract

This paper is concerned with the development of an efficient feature analyzer for handwritten character recognition. Feature analyzer presented in this paper can reduce the large domain of feature space and extract invariable information. Feature extraction has been viewed from multi dimensional perspective. To cope with the fuzziness of the recognition problem, a nonlinear classifier based on back propagation algorithm was used for classification. Generalizing capability of the system was increased by using ensemble of neural networks instead of using regular neural network. Training and testing using 10 fold cross validation and resultant impressive recognition accuracy (More than 90%) proves the effectiveness of the scheme.

1. Introduction

Any pattern recognition system typically consists of selection and extraction of useful features [1,2,3,5] from a pattern and use of a classifier to distinguish it from a set of similar looking patterns. A pattern can have a large number of measurable attributes, all of which may not be necessary for uniquely identifying it from other patterns in a particular domain of classification problem using a chosen classifier. Good features enhance within-class pattern similarity and between class pattern dissimilarity. Therefore feature extraction is the most challenging part for pattern recognition problems. For pattern recognition problems, that involve image analysis, different feature extraction scheme have been proposed and used by the researchers [2,3,5,6,7]. Selection of a particular class of feature vector varies from problem to problem. For character recognition problem, researchers have employed different feature extraction methods. Chen & Lieh [5] proposed a two-layer random graph-based scheme, which used components and strokes as primitives. Govindan et Al [6], Jain et al [7] developed character recognition system using pattern matching.

Researchers [10] also used Moment method for handwritten character recognition. Pure template based approach using neural network was also used for such purpose. Use of a specific type of feature vector limits the nature of variation of the input domain system can incorporate. None of the approaches used by the previous work could deal with fuzziness of the problem space and recognition accuracy was also not very high for this reason. Presence of noise and variability of the environment introduce fuzzy ness in the problem space. To cope with these variations, some intelligent approaches of feature detection are required. In this paper, feature extraction is viewed from different perspective of a character image, which is a novel approach in the sense that no relevant literature unified such broad feature dimensional perspective. Quad tree representation and contour based [4,8,9] representation have been used in order to facilitate boundary value analysis which also motivates critical point analysis. Development of profile curve from character image, enables projection profile analysis. Density analysis and structural analysis are also performed in order to extract structural information from character images. Images used for this particular research work are limited to handwritten characters of a specific alphabet, which is Bengali alphabet for the present study. But the pattern of feature analysis is not limited to some specific alphabet. For classification purposes, connectionist networks have been used because of their, inherent capability to learn and generalize. But instead of using regular feed forward neural network, artificial neural network ensemble [12,13,14,15] was used that also helped to increase the generalizing ability. Extensive training and testing has been performed using cross validation technique [16]. There are two processing phases: i) Training and ii) Recognition. Input images go through noise removal, smoothing and feature extraction in both phases.

In the next section, some image acquisition and processing steps are presented. Section 3 is based on feature analysis techniques. The next section discusses on classification and recognition. Experimental results are shown in section 5 and the conclusion is drawn in the final section.

2. Image Acquisition and Processing

In the present system, handwritten character images were obtained by optical scanning on the plain paper. For digitization purpose, input image was scanned and a digital image was formed. Following figure shows scanned handwritten image.



Fig 1: Some Sample Characters Captured by the Scanner

Characters were scaled using an efficient scaling algorithm [11] and converted to standard size, which was 128 x 128 for the system. Scaling is necessary to have a size invariant recognition. A filtering function was used to remove the noises in the image. Filtering function replaced each pixel by its majority function. Preprocessed image was divided into regions to have a quad tree representation. Each image was divided into four regions depending on the center of mass and bounded rectangle. Such a division yields quad tree of depth one. Similar procedures were applied to get quad tree of higher depth.

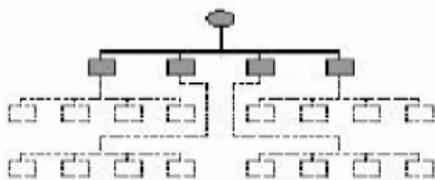


Fig 2. Quad Tree Representation

3. Feature Analysis

Feature extraction is an important step to achieve high performance for a character recognizer. To have a robust character recognition system, it is required to analyze and extract useful features [6,7,8,9], from character images. Extracted features must be invariant to the distortions and variations that can be expected in a specific application. Size of the feature set is important in order to avoid a phenomenon called the dimensionality problem. Following subsections

describes component of feature vectors, which were used in current analysis.

3.1. Density Feature Analysis

Pixel density in each of the regions from quad tree of depth one and higher were calculated to get density features. Density feature in quad tree region i is denoted by D_i where,

$$D_i = \frac{\sum_j (I(\beta_j))}{(N_i)} \dots\dots\dots (1)$$

Here, β_j is the j th black pixel, I is an intensity function. N_i is the total number of pixels in region i . Density features were calculated from quad tree of depth one and two. So 4 density features from quad tree depth one and 16 density features from quad tree of depth two were collected and their normalized values generated 20 feature vectors.

3.2 Boundary Value Analysis

Boundary of the image was traversed to find a string of code, which are also known as chain code [4,8,9].

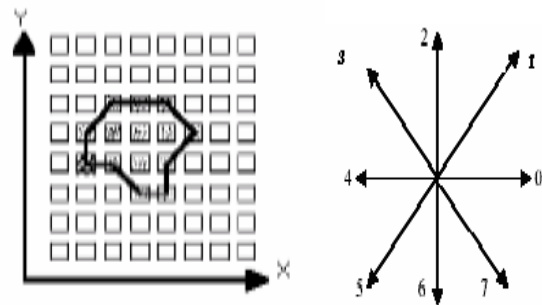


Fig 3: Chain Code Generation

Boundary tracing in depth zero, generated eight global chain coded features, denoted as F_{21} to F_{28} . Boundary tracing at depth one generated 32 chain coded features which were denoted as F_{29} to F_{60} . It is worth mentioning that these features were the distribution density in each region.

3.3 Critical Point Analysis

During tracing of the boundary of the image, critical points are noted for further analysis and processing. Here critical points are defined as the pixel within the contour where the change of directional

slope is notable comparing with its adjacent pixels. Distribution of critical points in quad tree of depth one and two yielded another set of meaningful features. For the present system, critical point distribution generated 20 such features denoted as: F_{61} to F_{80} .

3.4. Moment Feature Analysis

Using the histogram of the contour [8,9], moment features were calculated. 8 moment features were obtained, using the image histogram. Ratios of the length of the sum of contour segments, present within a sub-image to the total contour perimeter, generated the moment features. Moment feature for contour component j is denoted by μ_j , where

$$\mu_j = \frac{\sum_{i=1}^{n_i} \sum_{k=1}^{n_i} (D_i \beta(k))}{L_j} \dots \dots \dots (2)$$

Here, L_j = Total length of contour component j . D_i = Pixel density of region i . n_i = Number of segments in region i . $\beta(k)$ is a function that gives the chain coded value of contour segment k .

For the present analysis, moment features were calculated from quad tree of depth one and two, thus generating 8 feature vectors from depth one and 32 feature vectors from depth two regions. Moment features were denoted as F_{81} to F_{120} .

3.5 Gradient Feature Analysis

The ratio of the total contour length corresponding to the contour segments whose slope angle falls in division i and the total contour perimeter gives 4 gradient features. The mathematical definition of gradient feature is following:

$$\Omega_j = \frac{\sum_{i=1}^N f(S_{i,j})}{L} \dots \dots \dots (3)$$

Ω_j is the gradient feature of division j for the character image. Here L is the total contour perimeter. N is the total number of segments present in the contour. S_i is the i th contour segment. f is a gradient function that returns the value 1 if segment S_i falls in division j , otherwise it returns 0. For present analysis, gradient features are calculated from quad tree depth 1 thus yielding 4 gradient features denoted as F_{121} to F_{124} .

3.6 Projection Profile Analysis

Projections from four major directions were made to the image of the character. Each of the projection generated a profile curve, which was further processed.



Fig 4. Projection Profile for a Character

3.6.1 Length of the Profile Curve. Length of each of the profile curve was calculated in depth 1 quad tree region. Each length value was divided by the total length of boundary of the image. Thus four new features F_{125} to F_{128} were found from profile curve length.

3.6.2 Directional Strength of the Profile Curve. Directivity of the profile curve in each of the depth 1 quad tree region was computed. Frequency of directional slope of each of the profile curve was taken and multiplied by a constant to have directional strength in each of the directions. Here direction (0,4), (1,5), (2,6) and (3,7) were taken as equivalent and therefore four frequency values were computed for a profile curve. After multiplying by a strength constant, frequency values were converted to directional strength of the profile curve. These values were normalized to obtain fractional values, which were used as features by the system. Therefore 16 new features denoted as F_{129} to F_{144} were found from directional strength of the profile curve.

4. Classification and Recognition

For the present analysis, we used an efficient non-linear classifier [14]. The general architecture of the neural network is shown in figure 6. Our non linear classifier is essentially a neural network. The network is arranged in multiple layers, each layer containing fixed number of nodes for specific problem.

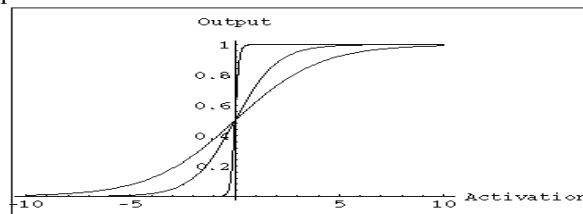


Fig 5. A Sigmoid Activation Function

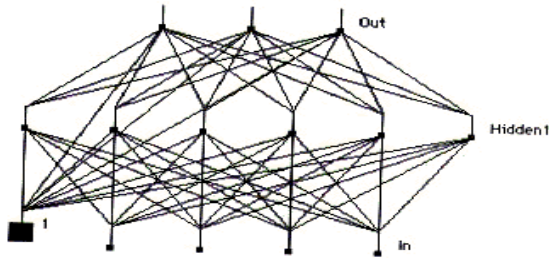


Fig 6. An Example Neural Network

The nodes of the input layer are called input nodes and output layers are called output nodes. Nodes at intermediate layers are called hidden nodes where all the intermediate layers are designated as hidden layers. Each input node is connected to each hidden node, and each hidden node is connected to each output node. There is a weight associated with each path between nodes. The input to a hidden node is the sum of all the input nodes times the weight along the path plus the bias to that node. The output from that hidden node is then determined by passing the input through an activation function. In figure 5, the activation function is a bipolar (or tan) sigmoid function.

4.1. Network Size

There is no rigorous rule about choice regarding the size of the network. The size for which convergence speed is higher can be used for training and recognition purpose. The learning constant of the neural network is usually kept small for faster convergence. Generally size of the input layer is made equal to the dimension of feature vectors and size of output layer is made equal to the dimension of output class. For the present application, size of the input layer was fixed at 144, which was equal to dimension of feature vectors but we varied the number of hidden neuron in hidden layer. Size of output layer was made equal to 50, which was equal to alphabet size for the present application.

4.2 Training

The feed-forward back propagation network undergoes supervised training, with a finite number of pattern pairs consisting of an input pattern and a desired output pattern. The network is trained by modifying the weights between the layers. Running the network consists of forward pass when the outputs are calculated and the error at the output units calculated and backward pass when the output unit error is used

to alter weights on the output units. Then the error at the hidden nodes is calculated and the weights on the hidden nodes altered using these values. For each data pair to be learned a forward pass and backwards pass is performed. This is repeated over and over again until the error is at a low enough level. In the usual back propagation algorithm, the gradient (calculated using the derivative of the activation function) is used to determine the change in the weights. However, especially in the second layer of back propagation, the result of the derivative of the activation function can produce a very small number, so there is a very small change in the weights, even though the net may be far from ideal. In resilient back propagation, only the sign of the gradient is used. The weight is then changed by one of two constant values depending on the sign of the gradient. This allows a net to learn much more quickly. We used resilient back propagation for training the network. Moreover, instead of using single neural network, an artificial neural network ensemble [13,14,15] was used for training and testing purposes. An artificial neural network ensemble was built in two steps, that is, generating component artificial neural networks and then combining their predictions. T bootstrap samples S_1, S_2, \dots, S_T were generated from the original training data set and a component artificial neural network N_t was trained using each S_t , an ensemble N^* was built from N_1, N_2, \dots, N_T whose output was the class label receiving the most number of votes. Since artificial neural network ensembles usually have strong generalization ability, some noise was depressed.

4.3 Recognition

In the recognition phase of the network a single iteration was enough to give the confidence value for each class of the character set. The value obtained from the output layer of the neural network, which closed to 1, implied the presence of that character class. Except the confidence value of the recognized character, other confidence values were closes to 0.

4.4 Addition of Intelligent Learning

Whenever some test patterns showed significant variation of feature values, but the network successfully recognized the pattern with a confidence value greater than a predetermined threshold, the feature value of that test pattern was stored in a separate database. After a particular number of times the network sees such significant and cognitive change

of input test patterns, the network retrained itself to adapt with such changing environment.

5. Evaluation

This section describes the performance of the feature analysis scheme for handwritten character recognition. Quantitative assessment of the recognition system has been illustrated in this section.

5.1 Experimental Setup

We collected handwritten samples from 12 individuals. Each person was asked to write each character of the alphabet 5 times. Thus 60 samples were collected for each character of the alphabet. Total 3000 sample handwritten characters were used for this work. The samples were divided into two parts, one for training phase and one for recognition phase to ensure cross validation. For the present system, 10 fold cross validation has been used. For 10 fold cross validation [16], 90% of the data was used for training and performance was tested on the remaining 10%. We divided characters in 10 major groups and measured recognition accuracy for each group separately. Recognition accuracy for each subject was also measured.

5.2 Results

Our experimentation showed that recognition accuracy ranged from 84% to 96% for each subject who participated in the experimentation. Recognition accuracy for each character group ranged from 85% to 95%. Training size was varied and resultant recognition accuracy was measured. Highest recognition accuracy occurred when size of training was 90% and size of test data was 10%. Number of hidden neuron of each of the neural network was varied and corresponding mean squared error was measured. Minimum error was found when number of hidden neuron was 50. Value of learning constant was also varied during training process and resultant accuracy was observed.

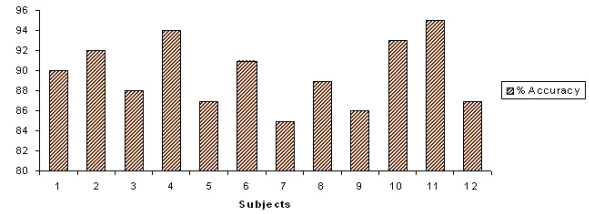


Fig 7. Accuracy Variation for Different Subjects

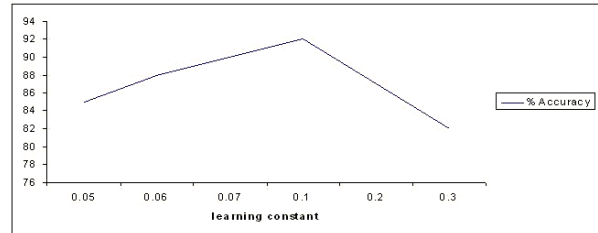


Fig 8. Accuracy Variation with Learning Constant

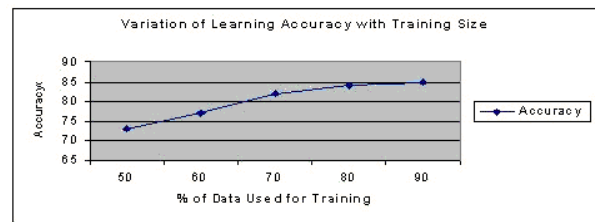


Fig 9. Accuracy Variation with Training Size

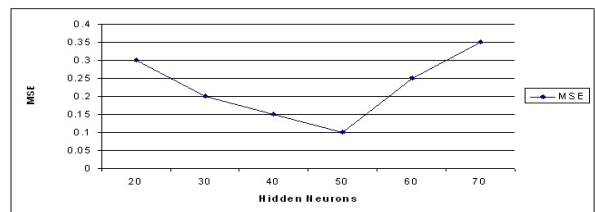


Fig 10. Mean Error Variation with Hidden Neurons

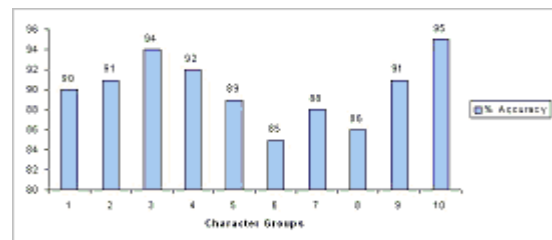


Fig 11. Accuracy Variation for Different Character Groups

6. Conclusion

Proposed intelligent feature analysis scheme has been successfully implemented in the area of handwritten character recognition. Analysis of features from different dimension enables the system to have a set of feature vectors which are noise invariant. Present system of handwritten character recognition also uses efficient classifier like neural network. To improve the generalization ability, ensemble based approach has been used. Speed of convergence has been effectively enhanced by resilient back propagation algorithm. Significant amount of training and testing was also performed and cross validation approach during training and testing enabled the system to have robust recognition accuracy. To limit the problem within a manageable size, only character images from a particular alphabet were used for training and recognition. But the feature analysis model is not language dependent, it is entirely possible to train and test using characters from other alphabets. This feature analysis model is capable of dealing with large image databases. The system may be enhanced to incorporate other image types like signatures and fingerprints. Application of such feature based analysis to signature images and fingerprints is a work in progress.

7. References

- [1] T. Pavlidis, "A Vectorizer and Feature Extraction for Document Recognition", *Computer Vision, Graphics and Image Processing*, Vol. 35, pp. 111-127, (1986).
- [2] B.Zhang and S.N.Srihari. Analysis of handwriting individuality using handwritten words. *Proceedings of the 7th International Conference on Document Analysis and Pattern Recognition*, 2003.
- [3] T. Pavlidis, "Algorithms for Graphics and Image Processing", Springer-Verlag,Berlin, (1982).
- [4] H.Freeman , "Computer Processing of Line Drawing Images, Computing Surveys",Vol. 6, No. 1,pp. 57- 97,1974.
- [5] Chen L-H,Lieh J R, "Handwritten character recognition using a two layer random graph model by relaxation matching". *Pattern Recogn.* 23: pp. 1189–1205, 1990.
- [6] Govindan V K, Shivaprasad A P, "Character recognition - a review",*Pattern Recogn.*23: pp. 671–683, 1990.
- [7] Jain A K,Zongkar D, "Representation and recognition of handwritten digits using deformable templates". *IEEE Trans. Pattern Anal. Machine Intell.* PAMI-19: pp. 1386–1391, 1997.
- [8] S. Madhvanath,G. Kim and V.Govindaraju, "Chain Code Contour Processing for Handwritten Word Recognition", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol-21, No-9, 1998.
- [9] C.C.Lue and J.G Dunhaum, "Highly Efficient Coding schemes for contour lines based on Chain Code Representation", *IEEE Trans, Comm Vol.* 39, No. 10,pp. 1511-1514., 1991.
- [10] G.L. Cash and M. Hatamian,"Optical character recognition by the method of moments", *Comput. Vision Graphics Image Processing*, vol. 39, pp. 291-310,1987.
- [11] Suman Kumar Nath and Muhammad Mashroor Ali, "An Efficient Object Scaling Algorithm raster device",*Graphics and Image Processing*, NCCIS, 1997.
- [12] B. Gosselin, "Neural Networks Combination for Improved Handwritten Characters Recognition", *Proc. of Int. Conf. on Signal and Image Processing*, pp 144-146, Las Vegas, Nevada,November 1995.
- [13] L. K. Hansen and P. Salamon, "Neural network ensembles," *IEEE Trans.Pattern Analysis and Machine Intelligence*, vol.12, no.10, pp.993-1001,1990.
- [14] L. K. Hansen, L. Liisberg, and P. Salamon, "Ensemble methods for handwritten digit recognition," in *Proc. IEEE Workshop on Neural Networks for Signal Processing*, Helsingoer, Denmark, 1992, pp.333-342.
- [15] Cunningham, J. Crney, and S. Jacob, "Stability problems with artificial neural networks and the ensemble solution," *Artificial Intelligence in Medicine*, vol. 20, no.3, pp.217-225, 2000.
- [16] M. Stone, Cross-Validatory Choice and Assessment of Statistical Predictions, *Journal of the Royal Statistical Society.* **36** (1), 111-147, 1974.