On the Power of Feature Analyzer for Signature Verification

Jalal Uddin Mahmud* and Chowdhury Mofizur Rahman**
**Department of Computer Science*
Stony Brook University, NY 11794 USA.
*** Department of Computer Science*
United International University, Dhaka Bangladesh

*E-mail: *jmahmud@cs.sunysb.edu,**cmr@uiu.ac.bd*

Abstract

This paper is concerned with verification of signatures using feature analysis and non linear classifier. Signatures are collected and scanned to obtain input image. Preprocessing involves removal of noise and making the input image size invariant. Feature analyzer can reduce the large domain of feature space and extract invariable information. Because of non linearity present in the input, a non linear classifier is used. Instead of using feed forward neural network, multiple feed forward neural networks are used which are trained in the form of ensemble. Using such ensemble makes the system more general than a regular single neural network based system. Resilient back propagation algorithm has been used for each neural network training to achieve faster recognition. Significant amount of training and testing has been performed using 10 fold cross validation and resultant impressive recognition accuracy (More than 90%) proves the effectiveness of the system.

1. Introduction

Pattern recognition is the interest of scientists and researchers for several decades for its applicability in many practical problems like character recognition, image classification and recognition, speech recognition, signature verification etc. In this paper we are interested to extract such useful information (features) from signature. From the viewpoint of pattern recognition, the task of signature verification is to judge whether an input signature is a genuine signature or a forgery by comparing it with collected signature samples. Automated signature verification is a particular interest among researchers for few years and significant works have been done in this area. Researchers like T.Hastie et al [1], I.Evett et al [2], F.Leclerc et al [3] studied and developed a model for signature verification.

The researchers applied the art of signature verification to the signatures during whose productions no measure is taken about pen trajectory or dynamics and to the signatures during whose productions pen trajectory or dynamics is captured. The first version is called as *offline signature verification system* and the second version is called as *online signature verification system* .The verification accuracy of the present signature verification systems are not very impressive and the systems are not very efficient in the presence of noise for offline signature verification. The proposed system of signature verification works only for offline signatures to reduce the complexity of the system architecture and to avoid the need of specialized hardware for collection of online signatures. For the development of signature verification system, useful and efficient feature extraction [4,5] is a crucial step. A signature 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 [6,7] is the most challenging part for pattern recognition problems. Selection of a particular class of feature vector varies from problem to problem. The choice of features to represent the patterns affects several aspects of the pattern recognition problem such as accuracy, required learning time and necessary number of samples.

For feature extraction from signature images, quad tree representation has been incorporated with density analysis, moment analysis and structural analysis. For classification purposes, artificial neural network ensemble [8,9] have been used. Artificial neural network ensemble is a learning technique where multiple artificial neural networks are trained to solve the same problem. The reason for using artificial neural network ensemble [10] is to enhance generalization ability of the system. In order to get a fast learner, resilient back propagation algorithm has been incorporated. Extensive training and testing have been performed using 10 fold cross validation technique [11] that also proves the effectiveness of the system.

2. Preprocessing

In the present system signature images have been obtained by optical scanning of the signature images on the plain paper. Preprocessing includes scaling, noise removal and elimination of redundant information as far as possible.

2.1. Digitization

For digitization purpose, signatures are written on white papers and acquired as binary images. The scanner scans the image that may be saved as a bitmap file. Further processing may be carried on that bitmap image.

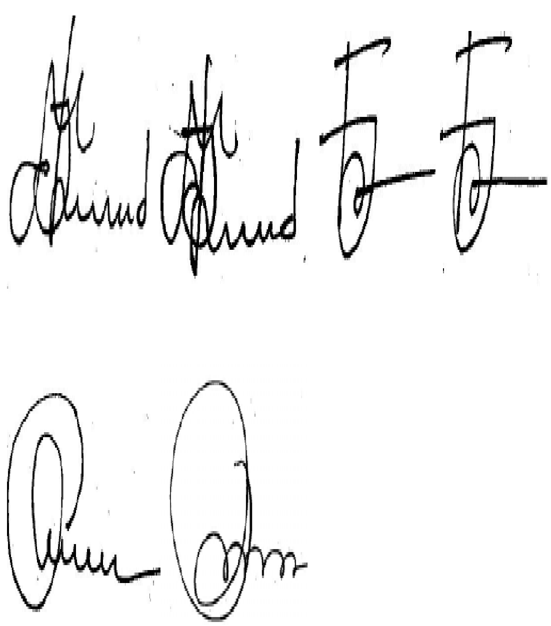

Fig 1: Some sample signatures captured by the scanner

2.2. Scaling

Scaling of the signature images has been performed so that size invariant recognition can be possible. The signatures that are scaled using an efficient scaling algorithm [12] converted to standard size, which is 64 x 64 for the system.

2.3. Noise removal

Dirt on camera or scanner lens, imperfections in the scanner lighting, etc introduces noises in the scanned signature images. A filtering function is used to remove the noises in the image. Filtering function works like a majority function that replaces each pixel by its majority function.

3. Feature Analysis

The size of the feature set is important in order to avoid a phenomenon called the dimensionality problem. A feature vector obtained from the coefficients of the expansion base function has global as well as detailed description at the same time. The resolution in symbolic features depends on the area from which the feature extractions are performed. In the proposed method, several types of features are extracted from the scaled input signature image, these includes density features, moment features and structural features. The overall feature extraction may be divided in some phases which are described below.

3.1. Quad tree formation

Each image is divided into four regions depending on the center of mass and bounded rectangle of the image. Such a division yields quad tree of depth one. Similar procedure may be applied to get quad tree of higher depth.

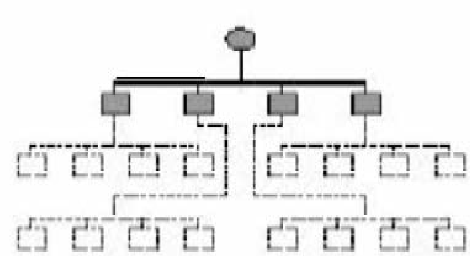

Fig 2: Quad tree representation

The mathematical definition of center of mass:

$$X_{CM} = \sum_{j=1}^{N} X_j / N \ \ldots\ldots\ldots\ldots\ldots\ldots (1)$$

$$Y_{CM} = \sum_{j=1}^{N} Y_j / N \ldots\ldots\ldots\ldots\ldots\ldots (2)$$

Here N = Number of Black pixels in the image,
X_j = x Coordinate of Black pixel in the image,
Y_j = y Coordinate of Black pixel in the image.

The mathematical definition of R_1, R_2, R_3 and R_4 are,

$R_1 = (X_{min} \ldots\ldots X_{CM}, Y_{min} \ldots\ldots Y_{CM})$
$R_2 = (X_{min} \ldots\ldots X_{CM}, Y_{CM+1} \ldots\ldots Y_{max})$
$R_3 = (X_{CM+1} \ldots\ldots X_{max}, Y_{CM+1} \ldots\ldots Y_{max})$
$R_4 = (X_{CM+1} \ldots\ldots X_{max}, Y_{min} \ldots\ldots Y_{CM})$

3.2. Density features

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

$$D_i = \sum_j (I(\beta_i)) / (N_i) \ \ldots\ldots\ldots\ldots\ldots\ldots (3)$$

Here, β_j is the jth black pixel,
I is an intensity function.
N_i is the total number of pixels in region i.

Density features are 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 are collected. Normalized density features generate 20 feature vectors.

3.3. Moment features

Using the histogram of the contour [13,14], moment features may be effectively calculated. 8 moment features are obtained, using the signature image histogram. Ratios of the length of the sum of contour segments, which are present within a sub-image to the total contour perimeter, generate the Moment features.

Moment feature for contour component j is denoted by μ_j, where

$$\mu_j = \left(\sum_{i=1}^{n} \prod_{k=1}^{n_i} (D_i, \beta(k))\right) / L_j \ \ldots\ldots\ldots\ldots\ldots\ldots (4)$$

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 are 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.

3.4. Structural features

Structural features from the signature image have been calculated for gaining the local information from the regions. Structural features conveys the following information,

3.4.1. Coarse structural features. A count of the number pixels in the thinned segment is obtained during segment tracing. This is normalized with respect to the area of the bounding box of the signature image. With the information of the number of pixels in the contour in a particular region, it is possible to gain an insight into the space filling property. For 4 Quad tree regions, there are 4 Space filling features that may be denoted as S_i,

$$\text{Where } S_i = \sum_j F(\alpha_{j,i})/A \ \ldots\ldots\ldots\ldots\ldots\ldots (5)$$

$\alpha_{j,i}$ = Number of active pixels in the boundary of the image in region i, F is an intensity function.

3.4.2. Directional features. Stated previously, critical points are calculated from the contour analysis of the image. The length of each segment is detected and the directivity of each segment indicates directional property of the image. Further analysis on directional property yields directional strength measure that effectively generates 4 new structural features denoted as θ_1, θ_2, θ_3 and θ_4.

The mathematical definition of the directional features denoted by θ are defined here:

$$\theta_i = \sum_j (\Psi_{j,i}) / m_j , \; m \le n \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(6)$$

$\Psi_{j,i}$ denotes the directivity of jth segment in the ith sub image. Here m_j denotes the total number of segments in sub image i. n_j denotes the total number of chaining elements in region i.

From the definition of these structural features, a set of modified structural features may be computed, which actually measure the directional strength of a region. Directional strength measures the weight value of a piece wise linear contour segment. Mathematical definition of directional strength of a particular contour segment can be given as:

$$\partial_i = \mu \prod_{j=1}^{m_i} L_i f(L_i, \lambda_{i,,j}) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(7)$$

∂i is the directional strength of segment i.
m_i is the number of neighbors of segment i.

Here, μ is the momentum constant whose value ranges between 0.001 to 0.999.For this particular experiment the value of μ was set to 0.5.

L_i is the length of particular segment i, measured in number of pixels. $\lambda_{i,,j}$ is the strength of jth element in ith segment. For this specific experiment $\lambda_{i,,j}$ equals the number of active neighbors of jth element of ith segment.
f is an importance function which is defined as :
$f(x,0) = 1$,
$f(x,y) = y$, when $x > Z_0$, where Z_0 is a threshold value.
$f(x,y) = y^2 - y + Z_0$, otherwise.

From the directional strength of contour segment, weighted directivity of a region is defined as:

$$W(\theta, i) = \prod_{j=1}^{m_i} \theta_i \partial_j \ldots\ldots\ldots\ldots\ldots\ldots(8)$$

For the present analysis, 4 coarse structural features and 4 directional features are cal calculated for quad tree region one. Their normalized values generate 8 structural feature vectors.

4. Classification and Recognition

Signature data is highly non-linear in nature as a result of varying styles. For the present analysis, it is required to use an efficient non-linear classifier [14]. The general architecture of the neural network is shown in the following figure. The network is arranged in multiple layers, each layer containing fixed number of nodes for the specific problem. The nodes of the input layer are called input nodes, output layers are called output nodes and the 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. There is also a weighted bias feeding into each hidden and output node. Therefore the input to each hidden node is the sum of all on the input nodes times the weight along the path plus the weighted bias to that node. The output from that hidden node is then determined by passing the input through an activation function. In the figure shown above the activation function is a bipolar (or tan) sigmoid function. 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. For each training iteration (epoch), the error is calculated by taking the difference between the output and the expected output.

4.1. Use of artificial neural network ensemble

Instead of using single neural network, an artificial neural network ensemble [5,6] is used for training and testing purposes. Say the training data set is $S = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$, where x_i and y_i are the feature vector and the expected class label of the *i*-th training instance, respectively. An artificial neural network ensemble is trained with *S*. An artificial neural network ensemble is built in two steps, that is, generating component artificial neural networks and then combining their predictions. As for generating component networks, *Bagging* [15] and *Boosting* [16] are prevailing approaches. Bagging generates multiple

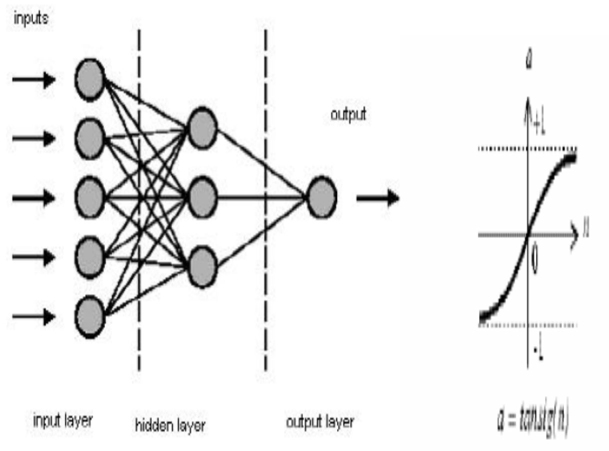

Fig 3: General neural network architecture and activation function

training data sets from the original training data set and then trains a component network using each of those training data sets. Boosting generates a series of component networks whose training data sets are determined by the performance of the former networks. Training instances that are wrongly predicted by the former networks will play more important roles in the training of the later networks. As for combining component predictions, *voting* [8] is prevailing for classification. Voting regards the class label receiving the most number of votes as the final output of the ensemble. For the present system, voting is used for component prediction combination. T bootstrap samples $S_1, S_2, \ldots, S_T$ are generated from the original training data set and a component artificial neural network N_t is trained using each S_t, an ensemble N^* is built from $N_1, N_2, \ldots, N_T$ whose output is the class label receiving the most number of votes. Since artificial neural network ensembles usually have strong generalization ability, some noise is depressed by the process of N^*.

4.2. Use of resilient back propagation

In the usual back propagation algorithm, the gradient 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 will be a very small change in the weights. In resilient back propagation [17], 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.

5. Experimental Result

For the signature verification system, signatures are collected from different persons. For each person, 10 sample signatures are collected. 30 persons participated in the experiment. The system was trained using 10 original signatures and 5 forgeries generated for each person. Original signature served as positive examples and forgeries as well as other signatures in the training data acted as negative examples. The number of persons taking part in the experiment has been varied to observe the verification accuracy at various levels. 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 [11], at any one time, 90% of the data is used for training and the performance is tested on the remaining 10%. 10-fold cross validation is performed in each case study. In each fold, an artificial neural network ensemble comprising sixteen individual networks is generated via Bagging [15]. The training data sets of the neural networks are bootstrap [18] sampled from the training data set of the fold. During the training process, the generalization error of the network is estimated in each epoch on its validation data set. If the validation error does not change in consecutive six epochs, the training of the network is terminated in order to avoid over fitting. Following figures display experimental results of the present system. From the experimental results, it is seen that recognition accuracy was above 90%. The execution speed of the system has been measured. The average execution speed was approximately 300 ms.

Table 1: Recognition accuracy for different training data sizes.

No of Person	No of patterns for each person	Rate of Correct classifi cation(%)	Rate of False accep-tance(%)	Rate of Rejec-tion(%)
10	10	96	2	2
20	10	94	4	2
25	15	92	3	5
30	15	90	5	5

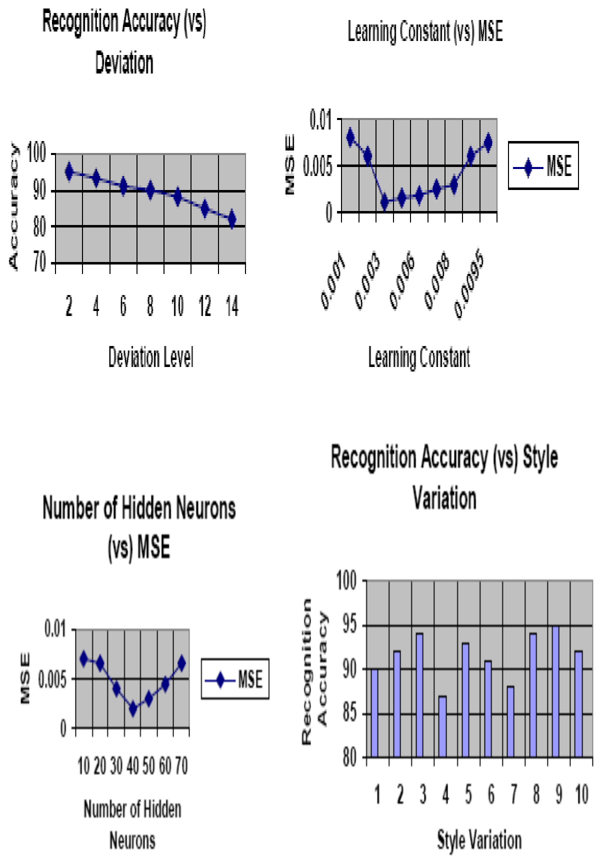

Fig 4: Graphical representation of experimental results

6. Conclusion

A new approach for signature verification system has been presented and implemented. Analysis of features from different dimension enables the system to have a set of feature vectors which are noise invariant. Present system of signature verification also employs efficient non linear 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 also performed and cross validation approach during training and testing enables the system to have robust recognition which is impressive as seen from the experimental result presented. Signature of same person is varied highly for various factors like pen pressure, emotion, environmental factors. This is why an offline signature verification system may not be able to verify signatures in all circumstances. To improve the rate of verification, the system should be online that involves extraction of stylistic features that are changed while writing on the paper. Even for the offline signature verification system, to accelerate the verification rate, features that are capable of describing the individual strokes, junctions and holes should be extracted. There are potential scopes for further research with these feature vectors extracted from signature image.

7. References

[1] T.Hastie, E.Kishon, M.Clark, and J.Fan, "A model for signature verification," in *Proc. of IEEE Int Conf. on Syst,ManCybern.*,Vol.1,Charlottesville,VA,Oct.1991,pp.191-196, 1991.

[2] I. Evett and R. N. Totty, "A study of the variation in the dimensions of genuine signatures," *J. Forensic Sci. Soc.*, Vol. 25, pp. 207–215, 1985.

[3] F. Leclerc and R. Plamondon, "Automatic signature verification: The state of the art 1989-1993," *Int. J.Patt. Recognit. And Artificial Intell.*,Vol. 8, No. 3, pp. 643–660, June 1994.

[4] T. Pavlidis, "*A Vectorizer and Feature Extraction for Document Recognition*", Computer Vision, Graphics and Image Processing, Vol. 35, pp. 111-127, (1986).

[5] D.Rutovitz, "*Pattern Recognition*", Journal of Royal Statistical Society, Vol. 129,Series A, pp. 504-530, (1966).

[6] S. Yokoi, J.I. Toriwaki, T. Fukumura, "*An Analysis of Topological Properties of Digitized Binary Pictures Using Local Features*", Journal of Computer Graphics and Image Processing, Vol. 4, pp. 63-73, (1975).

[7] T. Pavlidis, "*Algorithms for Graphics and Image Processing*", Springer-Verlag,Berlin, (1982).

[8] L. K. Hansen and P. Salamon, "*Neural network ensembles*," *IEEE Trans.Pattern Analysis and Machine Intelligence*, vol.12, no.10, pp.993-1001,1990.

[9] 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.

[10] 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.

[11] M. Stone, Cross-Validatory Choice and Assessment of Statistical Predictions, Journal of the Royal Statistical Society. **36** (1), 111-147, 1974.

[12] Suman Kumar Nath and Muhammad Mashroor Ali, An Efficient Object Scaling Algorithm for raster device", raphics and Image Processing, NCCIS , 1997.

[13] 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.

[14] C. M. Bishop, Neural networks for pattern recognition, Clarendon Press, Oxford, 1995.

[15] L. Breiman, "Bagging predictors," *Machine Learning*, vol.24, no.2, pp.123-140, 1996.

[16] H. Drucker, R. Schapire, and P. Simard, "Improving performance in neural networks using a boosting algorithm," in *Advances in Neural Information Processing Systems 5*, S.J. Hanson, J. D. Cowan, and C. L.Giles, Eds. San Mateo, CA: Morgan Kaufmann, 1993, pp.42-49.

[17] T. P. Vogl, J. K. Mangis, A. K. Rigler, W. T. Zink, and D. L. Alkon, Accelerating the convergence of the back-propagation method, Biological Cybernetics, **59**, 257-263, 1988.

[18] B. Efron and R. Tibshirani, *An Introduction to the Bootstrap*. New York: Chapman & Hall, 1993.