Neural Comput & Applic (1994)2:227–233

Neural
Computing
& Applications

Fingerprint Classification using a Fuzzy Multilayer Perceptron

Sushmita Mitra, Sankar K. Pal and Malay K. Kundu

Machine Intelligence Unit, Indian Statistical Institute, Calcutta, India

A fuzzy multilayer perceptron is used for the classification of fingerprint patterns. The input vector consists of texture-based features along with some directional features. The output vector is defined in terms of membership values to the three classes, viz. Whorl, Left Loop and Right Loop. Perturbation is produced randomly at pixel locations to generate noisy patterns. This helps to demonstrate the ability of the model in handling distorted fingerprint images. A study is made on the effect of reducing the number of input features while increasing the size of the network on its recognition performance.

Keywords: Fuzzy multilayer perceptron; Fingerprint classification; Texture

1. Introduction

Artificial neural networks [1–3] are found to be proficient in solving various pattern recognition problems. An advantage of neural nets lies in the high computation rate provided by their massive parallelism, so that real-time processing of huge data sets becomes feasible with proper hardware. On the other hand, the utility of fuzzy sets [4, 5] is inherent in their ability to model the uncertain or ambiguous data so often encountered in real life. Therefore, fuzzy neural networks [6, 7] are designed to utilize a synthesis of the computational power of the neural networks along with the uncertainty handling capabilities of fuzzy logic. The fuzzy multilayer perceptron, reported in refs [8, 9], has been found to be capable of efficient classification of fuzzy as well as linearly nonseparable nonconvex and disjoint patterns.

Texture is one of the important characteristics used in identifying objects or regions of interest in an image [10–12]. It is often described as a set of statistical measures of the spatial distribution of grey levels in an image. The grey level co-occurrence matrix method [10] assumes that the texture information in an image is contained in the overall or 'average' spatial relationships which the grey tones have to one another. This second-order statistics scheme has been found to supply a powerful input feature representation. It has been applied to cloud classification from satellite images [13] and in the identification of human faces [14], using neural network models.

Automated fingerprint classification constitutes a complex problem in the pattern recognition domain. As the size of the database increases, problems may arise in the classification efficiency of the method. Neural networks may be proposed as a solution in tackling such huge sets of complicated patterns. A layered neural network using an extracted feature ridge pattern as input and different subnetworks for each fingerprint category has been reported [15]. Besides, use of moment invariants has been made in ref. [16] for fingerprint matching.

The present work demonstrates the capability of a fuzzy neural network model [8, 9] in classifying fingerprint patterns. Noise is incorporated at randomly chosen pixel locations to test the efficiency of the model in recognising inaccurate and distorted patterns. The input vector consists of features extracted from texture and some directional properties. The output is expressed in terms of class membership values to the various fingerprint categories. A study is made over the effect on the model's performance, when the number of input

Correspondence and offprint requests to: Sushmita Mitra, Machine Intelligence Unit, Indian Statistical Institute, 203, B. T. Road, Calcutta 700035, India.

features is decreased and the size of the network is increased.

2. Feature Extraction from Fingerprint Image

The choice of an appropriate data representation is a crucial point when solving a classification task. If a relatively low-level data representation is used, one might need a very large training set to get satisfactory results, because in general, the performance of the classifier is quite sensitive to the peculiarities of the training set. Therefore, the original input representation is usually transformed into a higher-level one by using human expertise for designing appropriate preprocessing operations. However, besides mere recognition rates, there are several other factors which may influence the choice of the data representation, *viz.* the need for high computation speed, or hardware limitations, favouring data representations which do not require complicated and time-consuming preprocessing. In many cases, this precludes the use of structured data representations. It is also to be noted that the number of samples needed to achieve a certain degree of accuracy grows exponentially with the dimension of the input space.

Fingerprints are recognised as a basic tool for positive identification of individuals, be it for criminals in law enforcement, for security clearance in the armed services, or for normal civilian identification purposes. However, it also becomes necessary to maintain large files of print records for this process. Automated computer processing promises a fast and accurate alternative in this sphere. In this paper, we utilise a neural network-based technique for fingerprint classification.

First, the scanner is used to take a picture of the fingerprint image. This is followed by the digitiser for translating the image of a pixel array of grey values. The pictorial information of the fingerprint pattern is represented as a two-dimensional array of pixels $N_x \times N_y$. The digital image I is a function which assigns some grey-tone value $G \in \{1,2, \ldots, N_g\}$ to each and every resolution cell. Let $L_x = \{1,2, \ldots, N_x\}$ and $L_y = \{1,2, \ldots, N_y\}$ be the horizontal and vertical spatial domains, respectively. Then $I : L_x \times L_y \rightarrow G$. We extract a set of textural and direction features (as explained below) to serve as the set of inputs to the neural network model. The preprocessing serves to reduce considerably the dimension of the input space, while also capturing the inherent details of the fingerprint patterns. This is evident from the results described in Sect. 5.

2.1. The Second-Order Statistics Method

The textural features are computed from a set of angular nearest-neighbour grey-tone spatial-dependence matrices [10]. The contextual texture information is specified by the matrix of relative frequencies P_{ij} with which two neighbouring resolution cells, having grey levels i and j and separated by a distance δ, occur in the image.

Co-occurrence Matrix. The unnormalized frequencies are defined by the elements $P(i,j,\delta;\theta)$ of a set of co-occurrence matrices, where θ is 0°, 45°, 90° and 135° for horizontal, right-diagonal, vertical and left-diagonal neighbour pairs, respectively. Formally, for angles quantized to 45° intervals, the unnormalised frequencies are defined by

$$P(i,j,\delta,\theta) = \#\{((k,l),(m,n)) \in (L_x \times L_y) \times (L_x \times L_y)\}$$

such that

$$|k - m| = \delta, l - n = 0, I(k,l) = i, I(m,n) = j \text{ for } \theta = 0^\circ$$

$$(k - m = -\delta, l - n = \delta) \text{ or } (k - m = \delta, l - n = -\delta), I(k,l) = i, I(m,n) = j \text{ for } \theta = 45^\circ$$

$$k - m = 0, |l - n| = \delta, I(k,l) = i, I(m,n) = j \text{ for } \theta = 90^\circ$$

$$(k - m = \delta, l - n = \delta) \text{ or } (k - m = -\delta, l - n = -\delta), I(k,l) = i, I(m,n) = j \text{ for } \theta = 135^\circ$$

where # denotes the number of elements in the set. Note that these matrices are symmetric, i.e. $P(i,j;\delta,\theta) = P(j,i;\delta,\theta)$.

Let us consider a simple 4 × 4 image with four grey levels, ranging from 0 to 3,

$$\begin{matrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 2 & 2 & 2 \\ 2 & 2 & 3 & 3 \end{matrix}$$

to illustrate the process of generating the co-occurrence matrix. Let P_H, P_{RD}, P_V and P_{LD} denote the co-occurrence matrices obtained from this pixel array along the four angular specifications, for $\delta = 1$. Then we obtain

$$P_H = \begin{matrix} 4 & 2 & 1 & 0 \\ 2 & 4 & 0 & 0 \\ 1 & 0 & 6 & 1 \\ 0 & 0 & 1 & 2 \end{matrix} \quad \text{for } \theta = 0^\circ,$$

$$P_{RD} = \begin{matrix} 4 & 1 & 0 & 0 \\ 1 & 2 & 2 & 0 \\ 0 & 2 & 4 & 1 \\ 0 & 0 & 1 & 0 \end{matrix} \quad \text{for } \theta = 45^\circ,$$

$$P_V = \begin{matrix} 6 & 0 & 2 & 0 \\ 0 & 4 & 2 & 0 \\ 2 & 2 & 2 & 2 \\ 0 & 0 & 2 & 0 \end{matrix} \quad \text{for } \theta = 90° \quad \text{and}$$

$$P_{LD} = \begin{matrix} 2 & 1 & 3 & 0 \\ 1 & 2 & 1 & 0 \\ 3 & 1 & 0 & 2 \\ 0 & 0 & 2 & 0 \end{matrix} \quad \text{for } \theta = 135° \; .$$

For example, the element in the position (2,1) of the horizontal P_H matrix denotes the total number of times the two grey levels with values 2 and 1 occurred horizontally adjacent to each other.

For nearest neighbour pairs, we have $\delta = 1$. Then the number of neighbouring resolution cell pairs R is given by

$$R = \begin{cases} 2N_y(N_x - 1) & \text{for } \theta = 0° \\ 2N_x(N_y - 1) & \text{for } \theta = 90° \\ 2(\mathrm{N}_x - 1)(\mathrm{N}_y - 1) & \text{otherwise} \end{cases} \tag{1}$$

The Extracted Features. The four matrices P_H, P_{RD}, P_V and P_{LD}, as described above, are used to generate the textural features. These correspond to the four angles $\theta = 0°, 45°, 90°$ and $135°$ respectively. For the sake of clarity of definition let us now consider the matrix $P(i,j)$ (without any specification about the angle θ).

Angular Second Moment (A) gives a measure of the homogeneity of the texture and is defined as

$$A = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} \left(\frac{P(i,j)}{R} \right)^2 \tag{2}$$

Note that R, from Eq. (1), is used as the normalising constant.

Contrast (C) provides a measure of the local variation in the texture, and is evaluated as

$$C = \sum_{n=0}^{N_g - 1} n^2 \left\{ \sum_{|i-j|=n} \frac{P(i,j)}{R} \right\} \tag{3}$$

Entropy (E) indicates the amount of randomness in the texture, and is computed as

$$E = - \sum_i \sum_j \frac{P(i,j)}{R} \log \frac{P(i,j)}{R} \tag{4}$$

The measure Homog (H) also provides an indication of the amount of homogeneity [17] in the texture. It is expressed as

$$H = \sum_{n=0}^{N_g - 1} \frac{1}{1 + n^2} \left\{ \sum_{|i-j|=n} \frac{P(i,j)}{R} \right\} \tag{5}$$

Note that the notation θ was omitted in Eqs. (2)–(5) to avoid clutter. Each measure may be calculated four times, corresponding to each of the four directional co-occurrence matrices. The average values A_I, C_I, E_I and H_I provide a non-directional (rotation-invariant) texture representation. We have

$$A_I = \frac{1}{4}(A_0 + A_{45} + A_{90} + A_{135})$$

$$C_I = \frac{1}{4}(C_0 + C_{45} + C_{90} + C_{135})$$

$$E_I = \frac{1}{4}(E_0 + E_{45} + E_{90} + E_{135})$$

$$H_I = \frac{1}{4}(H_0 + H_{45} + H_{90} + H_{135})$$

2.2. Some Directional Features

Let us consider the $N_x \times N_y$ image to be traversed along the right diagonal (a), vertically (b,c,d), along the left diagonal (e) and horizontally (f,g,h), such that each directional traversal encompasses a band of w pixels. This is depicted in Fig. 1.

Frequency is defined as the number of times one encounters humps or local maxima (valleys or local minima) among the grey tone values in the course of the traversal. We designate these as F_a, F_b, F_c, F_d, F_e, F_f, F_g, F_h, respectively, along the eight directions chosen. An average value is computed along each direction, considering the group of w pixels. For the right diagonal, we have

$$F_a = \frac{1}{w} \sum_w (\mathit{No.\ of\ local\ maxima/minima}) \tag{6}$$

Difference is evaluated as the square of the

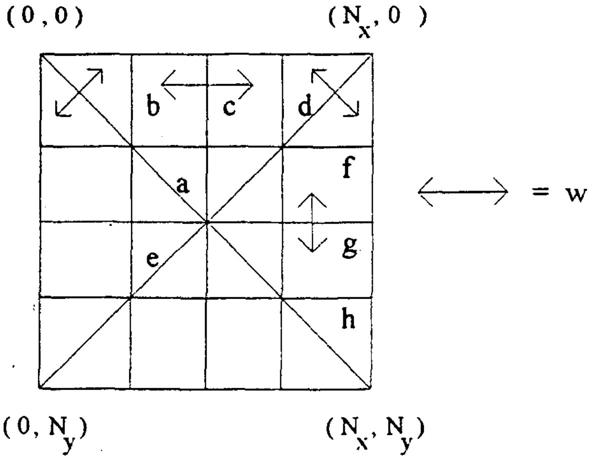

Fig. 1. The eight directions of traversal (each of width w) on the $N_x \times N_y$ image.

difference in the grey level values, between successive pixels, along the direction of traversal. The eight corresponding features are D_a, D_b, D_c, D_d, D_e, D_f, D_g, D_h and an average is computed over the group of w pixels. We define (along the right diagonal)

$$D_a = \frac{1}{w} \sum_w \sum_p (G_p - G_{p+1})^2 \tag{7}$$

where p and $p + 1$ refer to consecutive pixels along the chosen direction.

Height is computed as the normalised sum of the maximum grey tone value (among the band of w pixels) along the direction of traversal. It is expressed (along the right diagonal) as

$$T_a = \frac{1}{N_g} \sum_p \max_w \{G_p\} \tag{8}$$

where the summation over p refers to the set of pixels along the direction of traversal. The eight corresponding features (along the eight directions) are given as T_a, T_b, T_c, T_d, T_e, T_f, T_g and T_h.

Using a procedure analogous to Eq. (3), the directional contrast (along direction c and orientation θ) is computed as

$$K_{c_\theta} = \frac{1}{2w} \sum_{n=0}^{N_g-1} n^2 \left\{ \sum_{|i-j|=n} \frac{P'_{c_\theta}(i,j)}{N_y - 1} \right\} \tag{9}$$

where $P'(i,j)$ refers to the relative frequency with which two nearest neighbour cells, having grey levels i and j occurring along the vertical c band of w pixels in the image. Here, the normalising constant is $2w(N_y - 1)$. Note that only the direction c is traversed in this case.

3. Fuzzy Multilayer Perceptron

The multilayer perceptron (MLP) [1, 2] consists of multiple layers of sigmoid processing elements or neurons that interact using weighted connections. Consider the network given in Fig. 2. The output of a neuron in any layer other than the input layer ($h > 0$) is given as

$$y_j^{h+1} = \frac{1}{1 + e^{-\sum_i y_i^h w_{ji}^h}} \tag{10}$$

where y_i^h is the state of the ith neuron in the preceding hth layer and w_{ji}^h is the weight of the connection from the ith neuron in layer h to the jth neuron in layer $h + 1$. For nodes in the input layer we have $y_j^0 = x_j^0$, where x_j^0 is the jth component of the input vector.

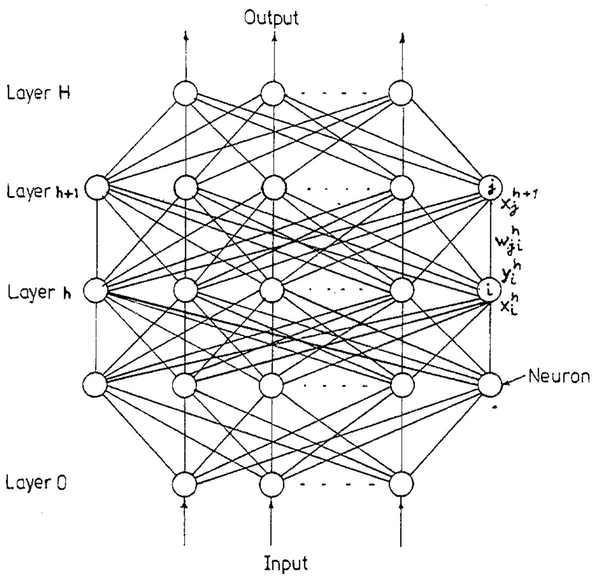

Fig. 2. The multilayer perceptron.

The Least Mean Square error in output vectors, for a given network weight vector $\mathbf{w}$, is defined as

$$E(\mathbf{w}) = \frac{1}{2} \sum_{j,c} (y_{j,c}^H(\mathbf{w}) - d_{j,c})^2 \tag{11}$$

where $y_{j,c}^H(\mathbf{w})$ is the state obtained for output node j in layer H in input-output case c and $d_{j,c}$ is its desired state specified by the *teacher*. One method for the minimisation of E to apply the method of gradient-descent by starting with any set of weights and repeatedly updating each weight by an amount

$$\Delta w_{ji}^h(t) = -\epsilon \frac{\partial E}{\partial w_{ji}} + \alpha \Delta w_{ji}^h(t-1) \tag{12}$$

where the positive constant ϵ controls the descent, $0 \leq \alpha \leq 1$ is the momentum coefficient and t denotes the number of the iteration currently in progress. After a number of sweeps through the training set, the error E in Eq. (11) may be minimized.

The fuzzy version of the MLP, discussed here, is based on the model reported in ref. [8], and is capable of the classification of fuzzy patterns. To model real-life data with finite belongingness to more than one class, we clamp the desired membership values (lying in the range [0,1]) at the output nodes during training. For the ith input pattern we define the desired output of the jth output node as d_j, where $0 \leq d_j \leq 1$ for all j.

The ϵ of Eq. (12) is gradually decreased in discrete steps, taking values from the chosen set {2, 1, 0.5, 0.3, 0.1, 0.05, 0.01, 0.005, 0.001}, while the momentum factor α is also decreased [8].

4. The Output Vector for the Fingerprint Pattern

Fingerprint images essentially consist of two types of characteristic regions, *viz.* ridges and valleys. These ridges run somewhat parallelly and slowly over the finger. The ridge structure and the skin texture provide the uniqueness to the fingerprint, and this remains unchanged during one's lifetime. A fingerprint consists of three regions, *viz.* core area, marginal area and base area. The ridges from these three areas meet at a triangular formation called the 'delta region'. The centroid of this region is identified as the 'delta point'.

Depending upon the ridge flow on the core area and the number of delta points, fingerprints can be broadly classified (according to Henry) [18] as

Plain Arch: ridges enter from the left side, rise in the middle and leave on the right side.

Tented Arch: same as in Plain Arch, but the amount of rise in the middle is more here.

Loop: this is the most common type. Ridges enter from one side, proceed towards the centre and then turn to leave from the same side. There are two categories, *viz.* Left Loop and Right Loop, depending on the direction of the loop formed.

Whorl: ridge flow in the core area is circular, and two delta points are defined.

Twin Loop: the core area consists of ridges from two distinct loop patterns.

Accidental: this type consists of those patterns that cannot be classified under any of the above categories.

In this work, we studied the feasibility of our method on three common classes, *viz.* Whorl, Left Loop and Right Loop. Figure 3 shows some typical images of these three different fingerprint categories. The images were first digitised and then the input features extracted, as described in Sect. 2.

The output vector **d** has components

$$\mathrm{d}_j = \mu_j \tag{13}$$

for $j = 1, \ldots, 3$, where μ_j indicates the membership of the pattern to the jth category.

Initially, we had nine patterns, three each belonging to the three above-mentioned categories. With an objective of creating more patterns and also to test the performance of the model in the presence of distorted images, we introduced noise. Perturbation was made randomly at *perc*% of the $N_x \times N_y$ pixels (for each pattern). Let pixel p with grey value G_p be randomly chosen to be perturbed. Then we have

$$G_p = \begin{cases} G_p + 2 & \text{if } G_p \leq N_g - 2 \\ N_g & \text{otherwise} \end{cases} \tag{14}$$

for $p = 1, 2, \ldots, (perc^*N_x{}^*N_y)/100$.

The output membership μ_k (for a fingerprint belonging to category k) is computed as

$$\mu_k = 1 - 0.05 * perc \tag{15}$$

such that $\mu_j = 0$ if $j \neq k$.

5. Implementation and Results

There were initially nine fingerprint patterns, three each belonging to the three categories Whorl, Left Loop and Right Loop. Grey values in the range of [1,16] were present in the 256 × 256 image array. Perturbation was made randomly at *perc*% of the pixel positions for each pattern. The percentage *perc* of pixels affected by noise were varied from 0, 0.5, 1, 1.5, ..., 9, 9.5, 10. A total of 189 images

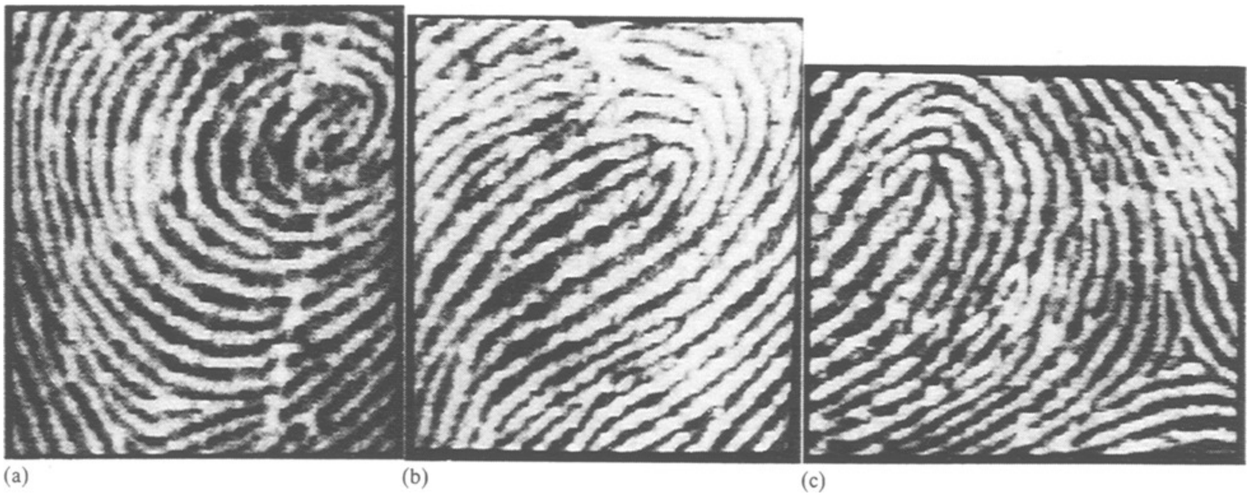

Fig. 3. The different categories of fingerprint patterns. (a) Whorl; (b) Left Loop; and (c) Right Loop.

were generated in the process, of which 66 were randomly selected for training the network, and the remaining 123 were kept aside for the testing phase. Note that the application of noise at random locations by Eq. (14) altered a number of grey values of the pixel arrays representing the fingerprint images. This had a corresponding effect on the values of the input features and the output vector, generating different input-output combinations (in each case) to be used for training and/or testing the fuzzy MLP under consideration. The input features were extracted as described in Sect. 2 and the output of the neural network clamped by the output vector defined in Eq. (15), in cases of both the training as well as test sets. A width of $w = 5$ was chosen for the directional features of Eqs. (6)–(9).

The fuzzy MLP had three output nodes corresponding to the three fingerprint categories. We used various numbers of layers as well as hidden nodes m. Note that in case of networks having more than one hidden layer, the number of hidden nodes m was kept constant in each layer. The perfect match p, best match b and mean square error *mse* were computed for the training set, while the individual classwise recognition scores, the overall score t and the mean square error mse_t were computed for the test set. The network was trained in the batch mode and all inputs were normalised to the range [0,1].

Table 1 demonstrates the classification results using 44 input features. These were A_0, A_{45}, A_{90}, A_{135} from Eq. (2), C_0, C_{45}, C_{90}, C_{135} from Eq. (3), E_0, E_{45}, E_{90}, E_{135} from Eq. (4), H_0, H_{45}, H_{90}, H_{135} from Eq. (5), F_a, F_b, F_c, F_d, F_e, F_f, F_g, F_h from Eq. (6), D_a, D_b, D_c, D_d, D_e, D_f, D_g, D_h from Eq. (7), T_a, T_b, T_c, T_d, T_e, T_f, T_g, T_h from Eq. (8) and K_{c_0}, $K_{c_{45}}$, $K_{c_{90}}$, $K_{c_{135}}$ from Eq. (9). It can be observed that the three-layered net was quite efficient in recognising the fingerprint patterns. Increasing the number of layers lead to a deterioration in performance. The best results were obtained with a single hidden layer having $m = 15$ nodes.

Table 2 illustrates the effects of reducing the number of input features on the classification efficiency of the model. Those features that had nearly similar values corresponding to all the patterns (from the three classes) were chosen to be removed. By eliminating E_0, E_{45}, E_{90} and E_{135}, we had 40 features. Next, we removed C_0, C_{45}, C_{90} and C_{135} to generate 36 input features. Omission of F_b, F_d, F_f, D_b, D_d, D_f, T_b, T_d and T_f resulted in 27 features at the input. As the number of features was decreased, the performance of the network deteriorated. This was especially noticeable over the test set. However, on increasing the number of hidden layers (in the case involving 27 input features), the performance improved again. This indicates that a balance has to be struck between the number of features at the input and the number of hidden layers (and nodes).

6. Conclusions

The fuzzy multilayer perceptron was used for the classification of fingerprint images. The input vector consisted of features extracted from texture and some directional properties. The output was provided as membership values to the three fingerprint categories, *viz*. Whorl, Left Loop and Right Loop. Random perturbation of pixel grey values was undertaken to obtain noisy patterns. A study was

Table 1. Recognition scores (%) for fingerprint data with 44 input features.

Layers =		3			4		5	
m =		10	15	20	10	15	10	15
perfect p		97.0	98.5	100.0	51.6	71.3	77.3	57.6
best b		100.0	100.0	100.0	100.0	100.0	100.0	100.0
mse		0.0006	0.0006	0.0009	0.006	0.004	0.003	0.005
T	1	68.3	82.9	63.4	48.7	63.4	63.4	48.7
e	2	56.1	70.7	92.6	48.7	48.7	64.1	48.7
s	3	100.0	97.5	87.8	100.0	100.0	100.0	100.0
t	Overall t	74.8	83.7	81.3	65.8	70.7	74.8	65.8
	mse_t	0.074	0.06	0.099	0.133	0.105	0.088	0.147

Table 2. Effect of reducing the number of input features on recognition score.

inpt	40	36	27										
layr	3	3	3				4				5		
m	15	15	5	10	15	20	5	9	10	15	5	6	10
perf	100.0	100.0	86.4	98.5	100.0	100.0	100.0	100.0	100.0	66.7	40.9	66.7	80.3
best	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
mse	0.0007	0.0007	0.002	0.0008	0.0003	0.0006	0.0006	0.001	0.0007	0.002	0.005	0.003	0.003
1	48.7	82.9	48.7	48.7	48.7	48.7	48.7	48.7	60.9	48.7	48.7	48.7	48.7
2	100.0	87.8	48.7	48.7	48.7	48.7	48.7	100.0	80.4	48.7	75.6	48.7	48.7
3	100.0	68.3	100.0	100.0	100.0	100.0	100.0	53.6	80.4	100.0	100.0	65.8	100.0
Net	82.9	79.6	65.8	65.8	65.8	65.8	65.8	67.4	73.9	65.8	74.8	54.4	65.8
mse_t	0.095	0.071	0.114	0.107	0.12	0.088	0.135	0.101	0.089	0.101	0.079	0.153	0.121

made on the impact over the classification efficiency of the model by reducing the number of input features while increasing the size of the network.

The results demonstrate that the fuzzy neural model is capable of recognizing distorted patterns. This is a positive indication of the strength of the neural net-based approach in classifying fingerprints that may possess distortion. The noise could be in the form of cut marks, blurs or perhaps be due to insufficient inking or smearing of the fingerprint images. Limitations regarding the availability of fingerprint data restricted us to some extent. The ability to handle more fingerprint classes will be investigated. Although only synthetic distortion has been used in the present work, the results hold promise for further investigation with naturally distorted fingerprint patterns.

We observed that the fuzzy MLP was translation invariant in classifying the fingerprint patterns with respect to the input window. We plan to extend the model in future to be rotation invariant as well.

Acknowledgement. This work was done while Prof. S.K. Pal held the Jawaharlal Nehru Fellowship.

References

1. Lippman RP. An introduction to computing with neural nets. IEEE Acoustics, Speech and Signal Processing Mag 1987; 4: 4–22
2. Hinton GE. Connectionist learning procedures. Artif Intell 1989; 40: 185–234
3. Carpenter GA. Neural network models for pattern recognition and associative memory. Neural Networks 1989; 2: 243–257
4. Klir GJ, Folger T. Fuzzy Sets, Uncertainty and Information. Addison-Wesley, Reading, MA, 1989
5. Zimmermann H-J. Fuzzy Set Theory–and its Applications. Kluwer, Boston, MA, 1991
6. Pao YH. Adaptive Pattern Recognition and Neural Networks. Addison-Wesley, Reading, MA, 1989
7. Bezdek JC, Pal SK. Editors. Fuzzy Models for Pattern Recognition: Methods that Search for Structures in Data. IEEE Press, New York, 1992
8. Pal SK, Mitra S. Multi-layer perceptron, fuzzy sets and classification. IEEE Trans Neural Networks 1992; 3:683–697
9. Pal SK, Mitra S. Fuzzy versions of Kohonen's net and MLP-based classification: Performance evaluation for certain nonconvex decision regions. Infor Sci 1994; 76: 297–337
10. Haralick RM, Shanmugam K and Dinstein I. Textural features for image classification. IEEE Trans Systems, Man Cybernetics 1973; 3: 610–621
11. Weszka JS, Dyer CR, Rosenfeld A. A comparative study of texture measures for terrain classification. IEEE Trans Systems, Man Cybernetics 1976; 6: 269–285
12. Tamura H, Mori S, Yamawaki T. Textural features corresponding to visual perception. IEEE Trans Systems, Man Cybernetics 1978; 8: 460–473
13. Lee J, Weger RC, Sengupta SK, Welch RM. A neural network approach to cloud classification. IEEE Trans Geoscience Remote Sensing 1990; 28: 846–855
14. Augusteijn MF, Skufca TL. Identification of human faces through texture-based feature recognition and neural network technology. Proc IEEE Int Joint Conf Neural Networks 1993; 392–398
15. Kamijo M. Classifying fingerprint images using neural network: Deriving the classification state. Proc IEEE Int Joint Conf Neural Networks 1993; 1932–1937
16. Newton SC, Pemmaraju S, Mitra S. Adaptive fuzzy leader clustering of complex data sets in pattern recognition. IEEE Trans Neural Networks 1992; 3: 794–800
17. Krishnapuram R, Lee J. Fuzzy-set-based hierarchical networks for information fusion in computer vision. Neural Networks 1992; 5: 335–350
18. Chapel CE. Fingerprinting: A Manual of Identification. Coward McCunn, 1941