ELSEVIER

FUZZY sets and systems

Fuzzy Sets and Systems 80 (1996) 121–132

Noisy fingerprint classification using multilayer perceptron with fuzzy geometrical and textural features

Sankar K. Pal, Sushmita Mitra*

Machine Intelligence Unit, Indian Statistical Institute, 203, B.T. Road, Calcutta – 700035, India

Received December 1994; revised May 1995

Abstract

A multilayer perceptron is used for the classification of noisy fingerprint patterns. In the first phase the input vector consists of some fuzzy geometrical features. In the second phase, we use some texture-based and directional features. The output vector is defined in terms of five classes, viz., whorl, left loop, right loop, twin loop and plain arch. Perturbation is produced randomly at pixel locations to generate noisy patterns. Cut marks and loss of information in certain random locations are also simulated. The investigation helps to demonstrate the generalization ability of the model in handling distorted fingerprint images.

Keywords: Multilayer perceptron; Fingerprint classification; Fuzzy geometry; Texture

1. Introduction

Artificial neural networks [5, 9] 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. The networks are also found to be robust to input noise and generally degrade gracefully to loss of components. On the other hand, the utility of fuzzy sets [7, 19] is inherent in their ability to model the uncertain or ambiguous data so often encountered in real life. Therefore, fuzzy neural networks [1, 14] are designed to utilize a synthesis of the computational power of the neural networks along with the uncertainty handling capabilities of fuzzy logic.

In pattern recognition and image analysis we often want to measure geometrical properties of regions in an image that are not crisply defined. Many of the standard geometrical properties of and relationship among regions can be generalized to fuzzy subsets. There has been a great deal of work in this regard by Rosenfeld [15, 16] who made these generalizations and extended the concept of digital picture geometry to fuzzy subsets. Such an extension is called fuzzy geometry of image subsets. Some more work has been done in this regard by Pal et al. [3, 11–13]. Fuzzy geometrical measures have also been found to reflect the spatial (geometrical) ambiguity of an image. Thus they seem to be useful in computing image properties and providing soft decision for image description and analysis by not allowing one to commit hard decisions.

Texture is one of the important characteristics used in identifying objects or regions of interest in an image [4, 17, 18]. It is often described as a set of statistical

* Corresponding author. E-mail: sushmita@isical.ernet.in.

SSDI 0165-0114(95)00192-1

measures of the spatial distribution of gray levels in an image. The method based on second-order statistical features, obtained from the gray level co-occurrence matrix [4], assumes that the texture information in an image is contained in the overall or "average" spatial relationships which the gray tones have to one another. This scheme has been found to provide a powerful input feature representation for various recognition problems.

Automated fingerprint classification constitutes a complex problem in the pattern recognition domain. Conventional approaches for fingerprint classification/recognition involve various tasks such as noise cleaning/enhancement of the images, thinning of ridges, feature extraction and (then) matching. As the regions are not always well defined (particularly because of the presence of noise, cut marks, blurs, excess ink or loss of information), any hard decision made at an operation would have an impact on the higher-level tasks; thereby introducing/enhancing the uncertainty in the final decision. Moreover, as the size of the database increases, the overall recognition task may become computationally more intensive.

A connectionist approach, with the input features being directly computed from the raw fingerprints without doing low-level operations, may be proposed as a solution for efficiently tackling such huge sets of complicated data and in handling uncertainties in the decision making process. Note that other connectionist approaches, using low-level operations for preprocessing the fingerprint images, include the work reported in [6] (using extracted feature ridge pattern as input and different subnetworks for each fingerprint category) and [10] (using moment invariants for fingerprint matching).

The present work is an attempt to demonstrate the capability of a multilayer perceptron (MLP) for classifying fingerprints in the aforesaid framework, where fuzzy geometrical features, and textural and directional features of the fingerprint images are considered as input. The present investigation also demonstrates the generalization capability of the MLP in identifying noisy, incomplete, blurred, distorted and cut marked fingerprints, particularly when it is trained only with unambiguous (correct) samples. The work is carried out in two stages. In the first part, we use fuzzy geometrical features as the input vector. In the second part, the input vector consists of features extracted from texture and some directional properties. The output is expressed in terms of the various fingerprint categories. Note that in both cases the training is done with unambiguous (noise-free) data.

2. Feature extraction from fingerprint-image

2.1. Fuzzy geometrical features

A fuzzy subset of a set S is a mapping μ from S into $[0, 1]$. For any $p \in S$, $\mu(p)$ is called the degree of membership of p in μ. A crisp (ordinary or non-fuzzy) subset of S can be regarded as a special case of a fuzzy subset in which the mapping μ is into $\{0, 1\}$. Some of the fuzzy geometrical properties of μ, relevant to the present work, are described below.

Let $\mu(I)$ denote a fuzzy representation of an $N_x \times N_y$ image I, i.e., a mapping μ from $I \in \{1, \ldots, N_g\}$ into $[0, 1]$ representing a fuzzy subset of I. For convenience, we shall use μ only to denote $\mu(I)$ in this work.

Area: The area of a fuzzy subset μ is defined as

$$a(\mu) = \int \mu \tag{1}$$

where the integration is taken over a region outside which $\mu = 0$. For μ being piece-wise constant (in case of digital image) the area is

$$a(\mu) = \sum \mu, \tag{2}$$

the summation being considered over a region outside which $\mu = 0$. The area is therefore the weighted sum of the regions on which μ has constant value weighted by these values.

Perimeter: If μ is piece-wise constant, the perimeter of μ is defined as

$$p(\mu) = \sum_{i,j,k} |\mu(i) - \mu(j)| \times |A(i, j, k)|. \tag{3}$$

This is just the weighted sum of the lengths of the arcs $A(i, j, k)$ along which the regions having μ values $\mu(i)$ and $\mu(j)$ meet, weighted by the absolute difference of these values. In case of an image if we consider the pixels as the piece-wise constant regions, and the common arc length for adjacent pixels as unity, then the perimeter of an image is defined by

$$p(\mu) = \sum_{i,j} |\mu(i) - \mu(j)| \tag{4}$$

where $\mu(i)$ and $\mu(j)$ are the membership values of two adjacent pixels.

Compactness: The compactness of a fuzzy set μ having area $a(\mu)$ and perimeter $p(\mu)$ is defined as

$$comp(\mu) = a(\mu)/p^2(\mu). \tag{5}$$

Physically, compactness means the fraction of maximum area (that can be encircled by the perimeter) actually occupied by the fuzzy region/concept represented by μ.

Height and width: The height $h(\mu)$ and width $w(\mu)$ of a fuzzy set μ are defined by

$$h(\mu) = \int \max_x \{\mu(x, y)\}\, dy \tag{6}$$

and

$$w(\mu) = \int \max_y \{\mu(x, y)\}\, dx, \tag{7}$$

where the integration is taken over a region outside which $\mu(x, y) = 0$. For a digital picture the definitions take the form

$$h(\mu) = \sum_y \max_x \{\mu(x, y)\} \tag{8}$$

and

$$w(\mu) = \sum_x \max_y \{\mu(x, y)\}. \tag{9}$$

So, height (width) of a digital picture is the sum of the maximum membership values of each row (column).

Length: The length of a fuzzy set μ may be defined as

$$l(\mu) = \max_x \left\{ \int \mu(x, y)\, dy \right\} \tag{10}$$

where the integration is taken over the region outside which $\mu(x, y) = 0$. In case of a digital picture the expression takes the form

$$l(\mu) = \max_x \left\{ \sum_y \mu(x, y) \right\}. \tag{11}$$

Breadth: The breadth of a fuzzy set μ may be defined as

$$b(\mu) = \max_y \left\{ \int \mu(x, y)\, dx \right\} \tag{12}$$

where the integration is taken over the region outside which $\mu(x, y) = 0$. For a digital image

$$b(\mu) = \max_y \left\{ \sum_x \mu(x, y) \right\}. \tag{13}$$

The length (breath) of an image fuzzy subset gives its longest expansion in the y direction (x direction). If μ is crisp, $\mu(x, y) = 0$ or 1; then length (breadth) is the maximum number of pixels in a column (row).

Index of area coverage (IOAC): The index of area coverage of a fuzzy set may be defined as

$$IOAC(\mu) = a(\mu)/l(\mu) \times b(\mu). \tag{14}$$

IOAC of a fuzzy image subset represents the fraction (which may be improper also) of the maximum area (that can be covered by the length and breadth of the image) actually occupied by the image.

2.2. Textural and directional features

The textural features are computed from a set of angular nearest-neighbour gray-tone spatial-dependence matrices [4]. The contextual texture information is specified by the matrix of relative frequencies P_{ij} with which two neighbouring resolution cells, having gray levels i and j and separated by a distance δ, occur in the image.

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

$$R = \begin{cases} 2N_y(N_x - 1) & \text{for } \theta = 0°, \\ 2N_x(N_y - 1) & \text{for } \theta = 90°, \\ 2(N_x - 1)(N_y - 1) & \text{otherwise.} \end{cases} \tag{15}$$

Angular second moment (A) gives a measure of the homogeneity of the texture and is defined as

$$A = \sum_{i=1}^{N_x} \sum_{j=1}^{N_y} \left(\frac{P(i, j)}{R} \right)^2. \tag{16}$$

Note that R, from Eq. (15), is used as the normalizing constant.

The measure *Homog*(H) also provides an indication of the amount of homogeneity [8] in the texture. It is expressed as

$$H = \sum_{n=0}^{N_s-1} \frac{1}{1+n^2} \left\{ \sum_{|i-j|=n} \frac{P(i,j)}{R} \right\}. \tag{17}$$

Note that the notation θ was omitted in Eqs. (16) and (17) to avoid clutter. Each measure may be calculated four times, corresponding to each of the four directional co-occurrence matrices. The average values A_1 and H_1 provide a non-directional (rotation-invariant) texture representation. We have

$$\begin{aligned} A_1 &= \tfrac{1}{4}(A_0 + A_{45} + A_{90} + A_{135}), \\ H_1 &= \tfrac{1}{4}(H_0 + H_{45} + H_{90} + H_{135}). \end{aligned} \tag{18}$$

Next, let us consider the $N_x \times N_y$ image to be traversed along the right diagonal, vertically (across the middle), along the left diagonal and horizontally (along the middle and also the lower region), such that each of the five directional traversals encompasses a band of w pixels.

Frequency is defined as the number of times one encounters humps or local maxima (valleys or local minima) among the gray tone values in the course of the traversal. An average value is computed along each direction, considering the group of w pixels. We have

$$F = \frac{1}{w} \sum_{w} (No\ of\ local\ maxima/minima) \tag{19}$$

Difference is evaluated as the square of the *difference* in the gray level values, between successive pixels, along the direction of traversal. We define

$$D = \frac{1}{w} \sum_{w} \sum_{p} (G_p - G_{p+1})^2 \tag{20}$$

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

Directional height is computed as the normalized sum of the maximum gray tone value (among the band of w pixels) along the direction of traversal. It is expressed as

$$T = \frac{1}{N_u} \sum_{p} \max_{w} \{G_p\} \tag{21}$$

where the summation over p refers to the set of pixels along the direction of traversal.

Directional contrast (for vertical traversal with orientation θ) is computed as

$$K_\theta = \frac{1}{2w} \sum_{n=0}^{N_s-1} n^2 \left\{ \sum_{|i-j|=n} \frac{P'_\theta(i,j)}{N_s - 1} \right\} \tag{22}$$

where $P'(i,j)$ refers to the relative frequency with which two nearest neighbour cells, having gray levels i and j occur along the vertical band of w pixels in the image. Here, the normalizing constant is $2w(N_s - 1)$.

3. The multilayer perceptron

The multilayer perceptron (MLP) [5, 9] consists of multiple layers of sigmoid processing elements or neurons that interact using weighted connections. Consider the network given in Fig. 1. The output of a neuron in any layer other than the input layer ($h > 0$) is given by

$$y_j^{h+1} = \frac{1}{1 + e^{-\sum_i y_i^h w_{ji}^h}} \tag{23}$$

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.

The least mean square error in output vectors, for a given network weight vector $\boldsymbol{w}$, is defined by

$$E(\boldsymbol{w}) = \frac{1}{2} \sum_{j,c} \left(y_{j,c}^H(\boldsymbol{w}) - d_{j,c} \right)^2 \tag{24}$$

where $y_{j,c}^H(\boldsymbol{w})$ 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 minimization of E is to apply the method of gradient-descent by starting with any set of weights and repeatedly updating each weight by an amount

$$\Delta w_{ji}^h(t) = -\varepsilon \frac{\partial E}{\partial w_{ji}} + \alpha \Delta w_{ji}^h(t-1) \tag{25}$$

where the positive constant ε controls the descent, $0 \leqslant \alpha \leqslant 1$ is the momentum coefficient and t denotes

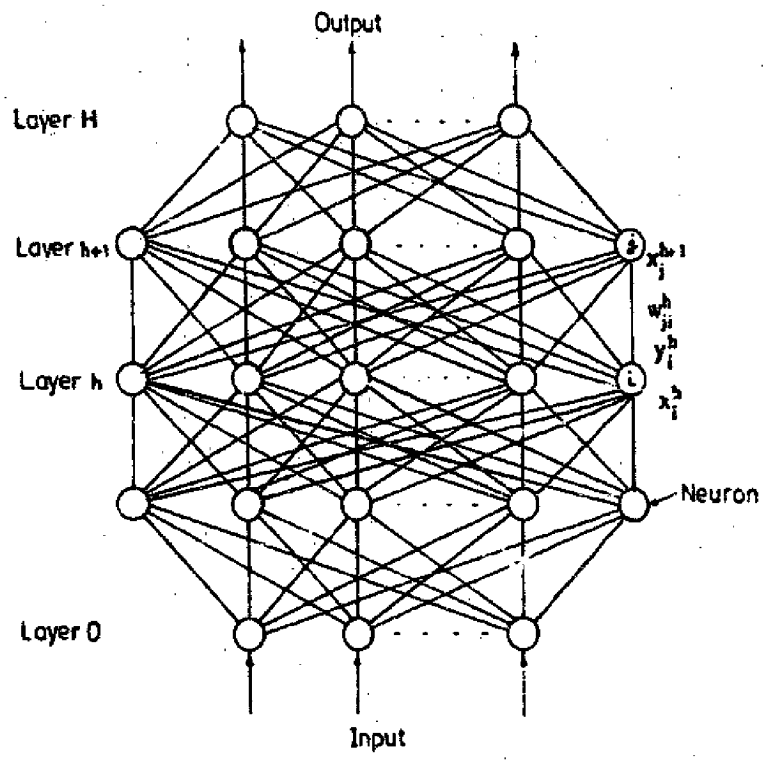


Fig. 1. The multilayer perceptron.

the number of the iteration currently in progress. After a number of sweeps through the training set, the error E in Eq. (24) may be minimized.

To model real-life data with finite belongingness to more than one class, we can 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 $0 \leqslant d_j \leqslant 1$ for all j. In the crisp case this reduces to $d_j \in \{0, 1\}$.

4. The output vector for the fingerprint pattern

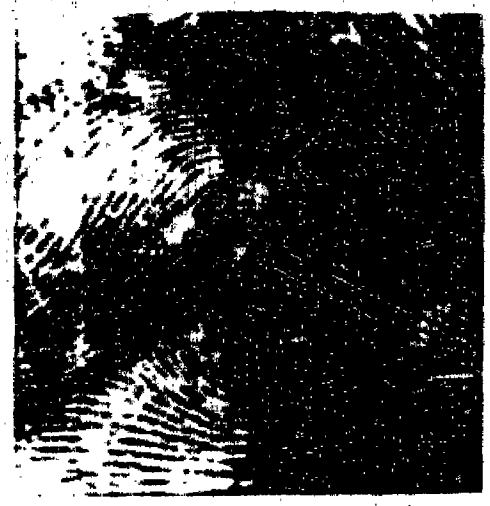
Fingerprint images essentially consist of ridges and valleys. The ridges run somewhat parallelly and slowly over the finger. The ridge structure and the skin texture provide the uniqueness to the fingerprint, and this remains unchanged during one's lifetime. A fingerprint consists of three regions, viz., 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.

4.1. Fingerprint categories

Depending upon the ridge flow on the core area and the number of delta points, fingerprints can be broadly classified (according to Henry) [2] 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 article, we have studied the classification ability of our method on five common classes, viz., whorl, left loop, right loop, twin loop and plain arch. These correspond to the five output nodes with desired output d_j for the MLP used. Fig. 2 shows some



(a)

(b)

(c)

(d)

(e)

Fig. 2. The different categories of fingerprint patterns. (a) Whorl, (b) left loop, (c) right loop, (d) twin loop and (e) plain arch.

typical images of these five different fingerprint categories.

5. Noisy pattern generation

In practice, we get fingerprints which are noisy. Noise may be of different types. There may be one or more cut mark(s) in the fingerprint, some portion of the fingerprint image may be missing due to an improperly taken impression, or noise may be distributed throughout the image. To model such situations, we have generated noisy fingerprint data using the following techniques.

5.1. Random distribution of constant noise

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 gray value G_p be randomly chosen to be perturbed. Then we have

$$G_p = \begin{cases} G_p + nt & \text{if } G_p \leqslant N_g - nt, \\ N_g & \text{otherwise} \end{cases} \tag{26}$$

for $p = 1, 2, \ldots, perc * N_x * N_y/100$, where nt represents the magnitude of noise introduced.

5.2. Random distribution of random noise

Next, we randomly selected a predefined percentage of pixels and injected random noise in the corresponding gray values. Let the magnitude of noise so added be represented by $X = x$, where X is normally distributed. We use $X \sim N(m, \sigma)$, where m is the mean and the σ is the standard deviation of the normal distribution. Thus, if a pixel p with gray value G_p is selected randomly, its new gray value becomes

$$G_p = G_p + x \tag{27}$$

such that $0 < G_p \leqslant N_g$.

5.3. Cut mark

Any two points in the fingerprint image were selected randomly and the pixels lying on a line of width b_w joining these two points were set to the highest gray value, N_g. In other words, we used

$$G_p = N_g \tag{28}$$

for all pixels p lying along the generated line (of width b_w), to simulate a cut mark on the fingerprint image. The cut marks were generated in two different orientations (along the left and right diagonals through the image), such that they are 90° apart. These are termed as the *forward* and *reverse* directions, respectively, for all later references.

5.4. Missing information

To model the occurrence of loss of information in a certain portion of a fingerprint image, we selected a portion of the image randomly. Setting all the pixels within this portion to the highest (N_g) or lowest (1) gray value simulates the loss of information in that region. So we have $G_p = N_g(1)$ for all pixels p lying within the randomly selected portion of the image. Note that setting $G_p = N_g$ models the case for insufficient inking of the fingerprint in the said region, whereas setting $G_p = 1$ simulates the condition of excess inking or blotches.

5.5. Other noisy versions

We randomly selected several seed points and generated boxes of size $b_s \times b_s$ around these points in each case. Then the gray values of the pixels within these regions were replaced by the average of all pixels within the respective boxes.

6. Implementation and results

There were initially thirty two noise-free fingerprint patterns, belonging to the five categories whorl, left loop, right loop, twin loop and plain arch. In procedure (i), we generated a total of 45 noisy samples, randomly perturbing 10% of the pixel locations with constant magnitude ($nt = 2$) of noise by Eq. (26), while the original 32 patterns were used for training the network. The input features were extracted as described in Section 2, in cases of both the training as well as test sets. A width of $w = 5$ was chosen for the directional features of Eqs. (19)–(22).

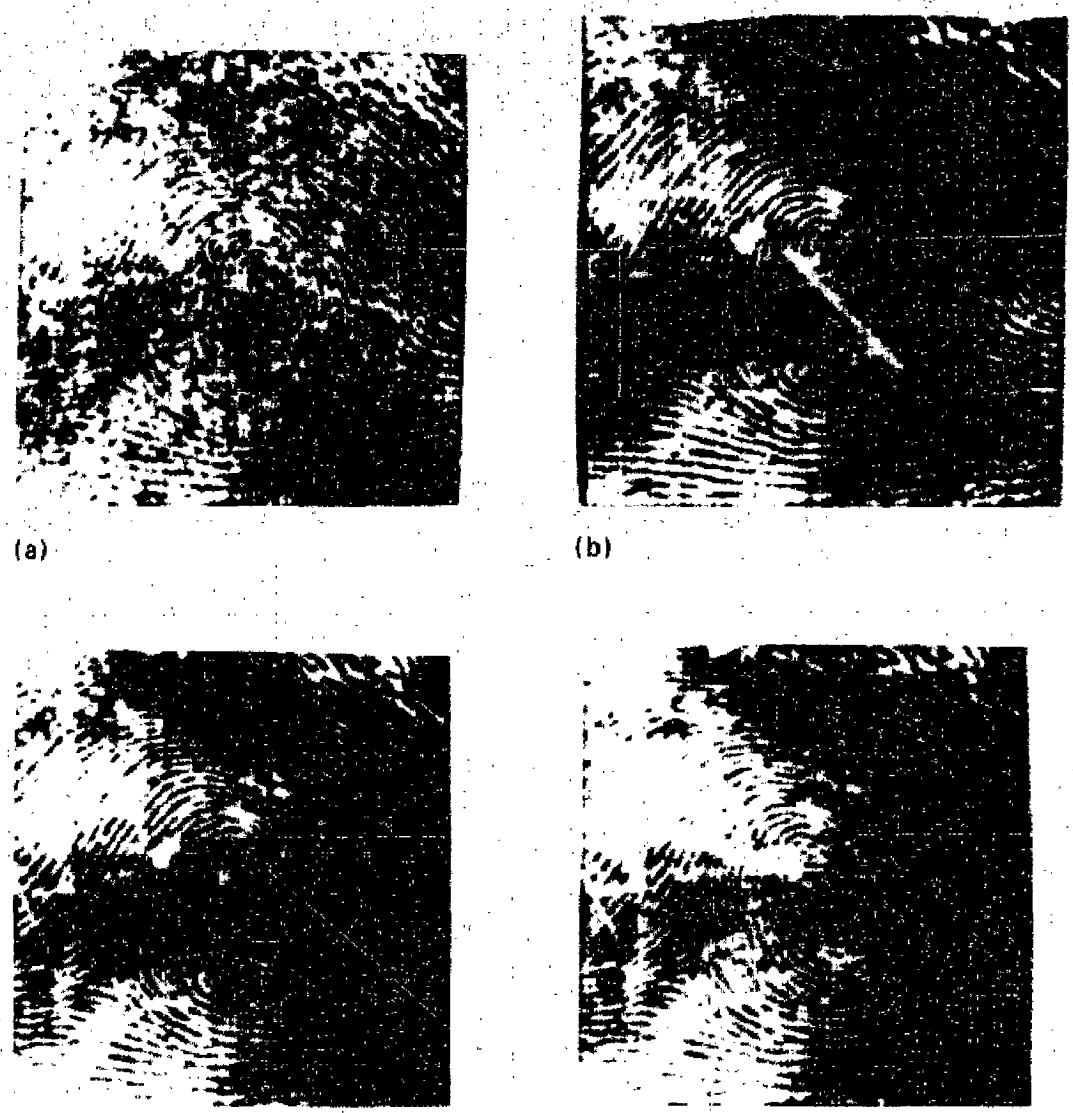


Fig. 3. The different types of noise modelled. (a) Random noise, (b) cut mark, (c) missing information (black), and (d) missing information (white).

Procedure (ii) involved simulating random noise, cut marks and loss of information in certain regions, as explained in Sections 5.2–5.5. The various types of noise injected are illustrated in Fig. 3. For inserting random noise into an image we used a normal distribution with mean 2 and standard deviation 10.0. A total of 10% of the pixels from the whole image were selected for this purpose. In the case of cut marks a band of width $b_w = 5$ was chosen in Eq. (28), while, for the averaging of gray values of Section 5.5 we used boxes of length $b_s = 21$. Once again the 32 unambiguous (noise-free) fingerprint images constituted the training set.

The multilayer perceptron had five output nodes corresponding to the five fingerprint categories. We used various numbers of layers as well as hidden nodes m. Note that in the case of networks having two hidden layers, the number of hidden nodes m was indicated as $m_1 : m_2$ corresponding to the two layers respectively. The best match b was computed for the training set while the individual classwise recognition scores along with the overall score t were computed for the test set. The network was trained in the batch mode and all input features were normalized to the range [0,1].

6.1. *Using fuzzy geometrical features*

The whole image was first divided into 16 blocks, each of size 64×64. Then we calculated the eight fuzzy geometrical features of Eqs. (1)–(14) for each

Table 1
Performance of fuzzy geometrical features using three-layered net

Hidden nodes m	25	30	35	40	45	50
Best match b	100.0	100.0	100.0	100.0	100.0	100.0
Test						
Whorl	77.78	77.78	77.78	77.78	77.78	77.78
L. loop	66.67	55.56	66.67	55.56	55.56	66.67
R. loop	66.67	66.67	66.67	66.67	66.67	66.67
T. loop	100.0	100.0	100.0	100.0	100.0	88.89
P. arch	100.0	100.0	100.0	100.0	100.0	100.0
Overall t	82.22	80.0	82.22	80.0	80.0	80.0

such subimages and generated a total of 128 features for a fingerprint. An MLP with a single hidden layer consisting of m hidden nodes was used for classifying the images. A total of around 1000 sweeps were required to reduce the error sufficiently during training.

6.1.1. Procedure (i)

Here we used the entire data set of 128 input features. The results in Table 1 demonstrate 100% learning over the training set (consisting of the original 32 unambiguous images) and a good performance ($\geqslant 80\%$) for the test set. It was observed that the model had relative difficulty in classifying patterns falling under the categories left loop and right loop. The test set corresponded to the patterns generated using random noise of constant magnitude as described in Section 5.1.

6.1.2. Procedure (ii)

In this part we tested the effect of reducing the rather large number of input features (used in Procedure (i)) on a *different* set of noisy patterns. For this purpose we selected five features, viz., *perimeter*, *compactness*, *width*, *breadth* and *IOAC* (from the eight initial fuzzy geometrical features) intuitively. It can be observed from Eqs. (1) and (2) that area is merely the sum of all the pixels in an image. Therefore, we ignored the feature *area* as this does not consider any neighbourhood information of a pixel. Since the features *height* and *length* are analogous to the features *width* and *breadth*, respectively (the only difference being that the computation is row-wise/column-wise), we selected only the features *width* and *breadth* from this set. In addition, we retained the two extracted features, viz., *compactness* and *IOAC*. These five features were calculated for each of the subimages (obtained as above), generating a total of 80 features. A multilayer perceptron with one hidden layer containing 25 nodes was trained for 1000 sweeps during training and 100% classification accuracy was obtained for the training set (consisting of the 32 unambiguous images). Finally the network was tested with the different types of noisy data (simulated as described in Sections 5.2–5.5), consisting of random noise with random magnitude, cut marks, information loss and averaging of gray values.

Table 2 shows the results of the testing phase using both 128 and 80 features. (During training the classification accuracy was 100% in both cases.) The results reveal that the data are very much sensitive to random noise as compared to other types of noises. This may be explained by the fact that the total number of pixels perturbed in this case, i.e., 10%, was much larger than that involved in the other cases (like cut mark, information loss or averaging of information). The performance was found to be the best in the case of the patterns where the gray values had been averaged over small regions. As expected, the performance deteriorated (on the whole) with a reduction in the number of input features.

Next we computed all the eight fuzzy geometrical features of Eqs. (1)–(14) globally, on the whole image, without dividing it into blocks. The network was trained with these eight input features (with seven hidden nodes) for 50,000 sweeps when 96.8% classification accuracy was obtained. Testing was carried out on the noisy data as above. Results of Table 3

Table 2
Recognition score (%) of noisy data with fuzzy geometrical features

Noise type	Random noise		Cut mark				Information loss				Averaging gray values	
			Forward		Reverse		Black		White			
Features	128	80	128	80	128	80	128	80	128	80	128	80
Whorl	60.0	60.0	100.0	100.0	100.0	100.0	100.0	100.0	60.0	60.0	100.0	100.0
L. loop	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
R. loop	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
T. loop	100.0	100.0	100.0	85.7	100.0	85.7	100.0	85.7	71.4	42.9	100.0	100.0
P. arch	71.4	57.2	100.0	100.0	100.0	100.0	100.0	100.0	57.1	66.7	100.0	100.0
Overall t	81.3	75.0	100.0	96.9	100.0	96.9	100.0	96.9	68.8	65.6	100.0	100.0

Table 3
Recognition score (%) of noisy data with 8 fuzzy geometrical features

Noise type	Random noise	Cut mark		Information loss		Averaging gray values
		Forward	Reverse	Black	White	
Whorl	40.0	60.0	60.0	60.0	40.0	100.0
L. loop	0.0	66.7	66.7	66.7	66.7	66.7
R. loop	33.3	100.0	100.0	66.7	66.7	100.0
T. loop	100.0	71.4	71.4	57.1	100.0	100.0
P. arch	28.6	100.0	100.0	100.0	64.3	100.0
Overall t	43.8	84.4	84.4	78.1	71.9	96.9

demonstrate that the performance is again very much sensitive to the presence of random noise, and least sensitive to the case of the averaging of gray values. Note that the recognition ability of the network is poorer in all cases with this reduced input feature set.

6.2. *Using textural and directional input features*

A total of 27 textural and directional features were generated from the whole image using Eqs. (16)–(22). The training and test sets were the same as used in case of the fuzzy geometrical features described in Section 6.1 (for the two corresponding *procedures*). The number of sweeps required during training were of the order of 150 to 200. It may be mentioned that here each input feature was computed along the four directions, viz., vertical, horizontal, right diagonal and left diagonal. This served to capture the directional properties of the image pattern. Note that this technique is different from the division into blocks as described for the fuzzy geometrical features. Thus both techniques serve to capture more information about the input space, albeit in different ways.

6.2.1. *Procedure (i)*

Table 4 depicts the results obtained with the textural and directional features with the test set being generated using random noise of constant magnitude as described in Section 5.1. During training, the network classified with 100% accuracy. The testing phase was also reasonably good, considering the much smaller number of sweeps required for training and the smaller number of input features involved (as compared to the fuzzy geometrical features of Table 1).

6.2.2. *Procedure (ii)*

In this part we used a three-layered MLP with 15 hidden nodes. The performance on the noisy data was found to be poorer in Table 5, as compared to that obtained with the fuzzy geometrical features of Tables 2 and 3. Here we have tested the effectiveness of the model for distortion under different types of noise simulated as described in Sections 5.2–5.5.

Table 4
Performance of textural and directional features

Layers	3				4	
Hidden nodes m	10	15	20	25	10:10	12:9
Best match b	100.0	100.0	100.0	100.0	100.0	100.0
Test						
Whorl	66.7	66.7	66.7	66.7	55.6	66.7
L. Loop	66.7	100.0	100.0	66.7	66.7	100.0
R. Loop	66.7	66.7	66.7	66.7	44.4	44.4
T. Loop	55.6	88.9	88.9	100.	77.8	77.8
P. Arch	100.0	100.0	100.0	100.0	100.0	100.0
Overall t	71.1	84.4	84.4	80.0	68.9	77.8

Table 5
Recognition score (%) of noisy data with textural and directional features

Noise type	Random noise	Cut mark		Information loss		Averaging gray values
		Forward	Reverse	Black	White	
Whorl	20.0	60.0	100.0	20.0	20.0	100.0
L. loop	100.0	100.0	100.0	66.7	33.3	100.0
R. loop	0.0	66.7	0.0	100.0	100.0	100.0
T. loop	0.0	71.4	28.6	40.0	40.0	85.7
P. arch	57.2	100.0	90.3	100.0	100.0	100.0
Overall t	37.5	84.4	71.9	68.8	65.6	96.9

This work serves to bring out the utility of the fuzzy geometrical features in classifying fingerprint images under various types of distortion, viz., random noise, cut marks, information loss and averaging of gray values. It is observed that the more conventional texture-based features are less effective in modelling such cases. Note that, once again, the network generated the worst results in the case of random noise and best results (comparable to that of Tables 1–3) in the case of the averaging of gray values.

7. Conclusions

The multilayer perceptron was used for the classification of noisy fingerprint images. In the first phase, fuzzy geometrical features were used as the input vector. In the second case, the input vector consisted of features extracted from texture and some directional properties. The output was provided in terms of the five fingerprint categories, viz., whorl, left loop, right loop, twin loop and plain arch. Random perturbation of pixel gray values was undertaken to obtain noisy patterns. Cut marks and loss of information in certain regions were also simulated to model damaged or distorted patterns. Note that in both cases the training set consisted of the unambiguous (noise-free) fingerprint images.

The results demonstrate that the use of fuzzy geometrical features helped the neural network in recognizing distorted patterns, to an appreciable extent. This is a positive indication of the generalization ability of the neural net based approach and the choice of the fuzzy input features selected, for the purpose of classifying distorted fingerprint patterns. The noise could be in the form of cut marks, blurs or perhaps be due to insufficient inking or smearing of the fingerprint images. Although only synthetic

distortion has been used in the present work, the results hold promise for further investigation with naturally distorted fingerprint patterns.

The fact that the *cut marks* or *loss of information*, could be better classified by the neural network brings out an interesting point for furtner investigation. Perhaps the 10% random noise, which visually damaged the pattern very little, caused some major changes in the feature values computed (both in cases of fuzzy geometrical and textural/ directional features). This calls for the selection of some new features (probably with a different approach) that may be able to overcome this problem. However, it should also be noted that the 10% locations of a 256×256 image consist of much more pixels than a cut mark of size (say) 21×5 or a region of information loss of size (say) 21×21. This probably accounts for the better performance in the latter case even though the distortion may have been in a sensitive region of the image that is relevant for the classification.

Acknowledgements

The authors gratefully acknowledge Mr. Rajat K De and Mr. Suman Mitra for their valuable assistance in programming and data generation. The helpful discussions with Dr. M.K. Kundu, Dr. N.R. Pal, Dr. A. Ghosh and Dr. M. Mitra is also acknowledged. This work was supported by the CSIR under the project grant No. 22(235)/93-EMR-ii. The work was done while Prof. S.K. Pal held the Jawaharlal Nehru Fellowship.

References

[1] J.C. Bezdek and S.K. Pal, Eds., *Fuzzy Models for Pattern Recognition: Methods that Search for Structures in Data* (IEEE Press, New York, 1992).

[2] C.E. Chapel, *Fingerprinting: A Manual of Identification* (Coward McCann, 1941).

[3] A. Ghosh, On image segmentation using neural networks and fuzzy sets. Ph.D. Thesis, Indian Statistical Institute, Calcutta, India (1992).

[4] R.M. Haralick, K. Shanmugam and I. Dinstein, Textural features for image classification, *IEEE Trans. Systems Man Cybernet.* **3** (1973) 610–621.

[5] G.E. Hinton, Connectionist learning procedures, *Artificial Intelligence* **40** (1989) 185–234.

[6] M. Kamijo, Classifying fingerprint images using neural network: Deriving the classification state, *Proc. IEEE Internat. Joint Conf. on Neural Networks*, San Francisco (1993) 1932–1937.

[7] G.J. Klir and T. Folger, *Fuzzy Sets, Uncertainty and Information* (Addison-Wesley, Reading, MA, 1989).

[8] R. Krishnapuram and J. Lee, Fuzzy-set-based hierarchical networks for information fusion in computer vision, *Neural Networks* **5** (1992) 335–350.

[9] R.P. Lippmann, An introduction to computing with neural nets, *IEEE Acoust. Speech Signal Process. Mag.* **4** (1987) 4–22.

[10] S.C. Newton, S. Pemmaraju and S. Mitra, Adaptive fuzzy leader clustering of complex data sets in pattern recognition, *IEEE Trans. on Neural Networks* **3** (1992) 794–800.

[11] S.K. Pal and A. Ghosh,Fuzzy geometry in image analysis, *Fuzzy Sets and Systems* **48** (1992) 23–40.

[12] S.K. Pal and A. Rosenfeld, A fuzzy medial axis transformation based on fuzzy disk, *Pattern Recognition Lett.* **12** (1991) 585–590.

[13] S.K. Pal and L. Wang, Fuzzy medial axis transformation (FMAT): Practical feasibility, *Fuzzy Sets and Systems* **50** (1992) 15–34.

[14] Y.H. Pao, *Adaptive Pattern Recognition and Neural Networks* (Addison-Wesley, Reading, MA, 1989).

[15] A. Rosenfeld, Fuzzy geometry of image subsets, *Pattern Recognition Lett.* **2** (1984) 311–317.

[16] A. Rosenfeld, Fuzzy geometry: an overview, *Proc. 1st IEEE Internat. Conf. on Fuzzy Systems*, San Diego, USA (1992) 113–117.

[17] H. Tamura, S. Mori and T. Yamawaki, Textural features corresponding to visual perception, *IEEE Trans. Systems Man Cybernet.* **8** (1978) 460–473.

[18] J.S. Weszka, C.R. Dyer and A. Rosenfeld, A comparative study of texture measures for terrain classification, *IEEE Trans. Systems Man Cybernet.* **6** (1976) 269–285.

[19] H.J. Zimmermann, *Fuzzy Set Theory and its Applications* (Kluwer, Boston, MA, 1991).