Neural Comput & Applic (1998)7:180–191

Neural
Computing
& Applications

Noisy Fingerprints Classification with Directional FFT Based Features Using MLP

S.N. Sarbadhikari, J. Basak, S.K. Pal and M.K. Kundu

Machine Intelligence Unit, Indian Statistical Institute, Calcutta, India

A methodology is described for classifying noisy fingerprints directly from raw unprocessed images. The directional properties of fingerprints are exploited as input features by computing one-dimensional fast Fourier transform (FFT) of the images over some selected bands in four and eight directions. The ability of the multilayer perceptron (MLP) for generating complex boundaries is utilised for the purpose of classification. The superiority of the method over some existing ones is established for fingerprints corrupted with various types of distortions, especially random noise.

Keywords: Fast Fourier Transform (FFT); Multilayer Perceptron (MLP); (Noisy) fingerprint classification

1. Introduction

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.

Automated fingerprint classification poses an interesting problem in pattern recognition, especially for forensic applications. The computer based identification of fingerprints involves two major steps: [1] (a) 'preprocessing' like enhancement of images, thinning of ridges and extraction of features; and (b) analysis of the processed (quality enhanced) image. Conventional fingerprint analysis systems involve 'thresholding', converting a multilevel intensity data into two level intensity data (black and white) using some heuristics. This is followed by thinning of the black concentric lines. Despite all these time consuming operations, the resultant fingerprint may not be of the desired quality. Fingerprints are identified in a hierarchical manner. The preprocessed (binary) images are classified by determining different micro characteristic features like ridge flow and minutiae type, number and position. A multilevel classifier using a syntactic approach [2], graph matching [3], detecting the number and locations of singular points [4] and using minutiae features after smoothing the binary image [1] have all been tried for the classification of fingerprints. Recently, neural networks have been used for this purpose. Artificial Neural Networks (ANN) [5] can be formally defined as *a massively parallel interconnected network of simple (usually adaptive) processing elements that interact with objects of the real world in a manner similar to biological systems*. The benefit of neural nets lies in the high computation rate provided by their inherent massive parallelism, thereby enabling real-time processing of huge data sets with proper hardware backing. The networks are also found to be robust to input noise, and generally degrade gracefully to loss of components. Various methods using networks for classification of binary ridge patterns for each fingerprint category have been developed [6–8].

One may note that using binary images in all the above techniques often leads to information loss. Moreover, it is not appropriate to commit oneself to a specific thresholding for binarisation, particularly when the ridges are not well defined. For example, in the case of forensic applications, the

Correspondence and offprint requests to: Dr M.K. Kundu, Machine Intelligence Unit, Indian Statistical Institute, 203, B.T. Road, Calcutta 700035, India. Email: malay@isical.ac.in

quality of fingerprint data is often found to be very poor because of the faint nature, noise and incompleteness. Conventional preprocessing techniques based on heuristic logic are usually incapable of handling such situations, often leading to erroneous processed results at the cost of expensive computer time. So it is desirable to have a system where such time consuming and error prone preprocessing techniques could be avoided altogether.

By computing input features directly from the raw fingerprints, the uncertainties in reaching a decision, and also the overall computational burden, are greatly reduced [9,10]. Based on this realisation, Pal and Mitra [11] computed fuzzy geometrical features and probabilistic entropy measures directly from unprocessed fingerprint images for their classification using the multilayer perceptron (MLP). Another investigation on fingerprint classification [12] uses a fuzzy MLP, [12] which exploits the nonlinear boundary generating capability of MLP and the uncertainty handling capacity of fuzzy sets to provide a more intelligent system. However, in earlier work [12,11], the performance of the neural nets in the presence of random noise was not very satisfactory. Moreover, it has been observed that obtaining fuzzy geometrical features is computationally expensive.

An important characteristic feature of an image is the texture [13]. By using textural features instead of geometric features, one can make computations easily, and also preserve the directional (semiglobal) properties. The power spectral (Fast Fourier transform, or FFT) approach for estimating the texture of an image [14,15] is an established method. In the case of fingerprints, where there is a definite periodicity (of ridges/valleys) and directionality, FFT could be a suitable quantifier of the texture in different directions. For the various fingerprint types, the FFT components are likely to be different. Moreover, since these features are global in nature, they are likely to be less sensitive to random noise.

The present article aims at developing an MLP-based methodology for fingerprint classification, exploiting the characteristics of FFT-based textural features, derived from the grey images, as input. Here the FFT is computed over only a few directional bands. This allows us to extract the specific directional properties of the various fingerprint types, and also reduces the computation time. The performance of the network, in the presence of different types of noise, is studied. The network's performance is also compared with some existing methods and the KNN classifier. Apart from that, whether the fractal dimensions of the FFT coefficients' curves provide distinguishing characteristics for the various fingerprint types or not is verified.

Section 2 describes the various fingerprint classes and the method of FFT feature extraction, followed by an outline of the multilayer perceptron-based classification scheme in Section 3. Section 4 presents the implementation method and results. The conclusions are drawn in Section 5.

2. Fingerprint Processing

2.1. Fingerprint Categories

Fingerprint images essentially consist of two types of characteristic regions, ridges and valleys. These ridges run parallel 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, 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) [16] as:

- *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 common categories, Left Loop and Right Loop, depending on the direction of the loop formed. In a third variety, Twin or Double Loop, the core area consists of ridges from two distinct loop patterns.
- *Arch*: in a Plain Arch, ridges enter from one side, rise in the middle and leave on the other side. The Tented Arch is the same as the Plain Arch, but the amount of rise in the middle is more here.
- *Whorl*: ridge flow in the core area is circular, and two delta points are defined. However, there may be two subtypes, Central Pocket and Elliptical Whorl. In the first subtype, there are circular ridges in the core, but it becomes asymmetric towards the base, i.e. between a whorl and loop. In the other subtype, the ridges are stretched in the direction of the finger. The distance between the end points of these stretched ridges are distinct.
- *Accidental, Mixed or Composite*: this type consists of those patterns that cannot be classified under any of the above categories.

In this work, we study the feasibility of our method

on five common classes: Left Loop, Right Loop, Twin Loop, Plain Arch and Whorl. Figure 1 shows some typical images of these five different fingerprint categories. The images are first digitised, and then the input features extracted, as described in Section 4. In our database, the images are at the same scale and have approximately the same orientation. They are 256×256 in size, with 8 bits per pixel and labelled manually.

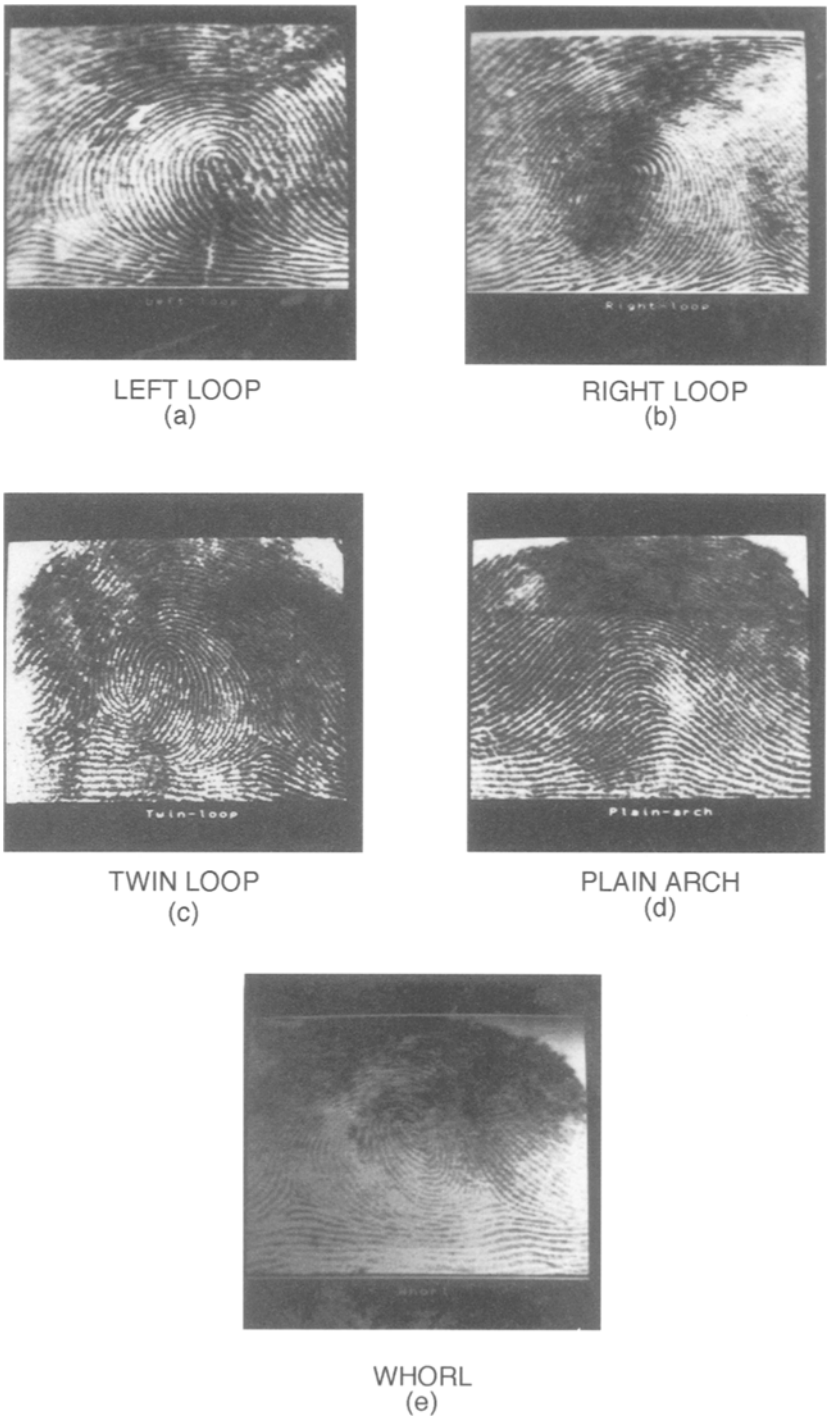


Fig. 1. The five varieties of fingerprint patterns used: Left Loop, Right Loop, Twin Loop, Plain Arch and Whorl.

2.2. Feature Extraction

Bands (horizontal, vertical and diagonal) of some fixed width (Fig. 2), passing through the core regions of the fingerprints, are considered for attributing the directionality properties. For example, in the case of a 'whorl', the nature of greytone variation will be approximately the same in all directions, whereas, for a 'loop' (left or right), the nature of greytone variations will be widely different for the two different diagonals. Thus, the features for classifying the different fingerprint categories can be obtained by suitably quantifying the nature of greytone variations in the directional bands. The greytone variations are quantified by computing the Fourier spectrum in these bands.

The Fourier spectrum (FFT) is computed for each line in each band as

$$F(u) = \frac{1}{N} \sum_{x=0}^{N-1} f(x) \exp[-j2\pi x/N] \tag{1}$$

for

$$f(x) = f(x_0 + x\Delta x)$$

where x assumes a discrete value $0, 1, 2, \ldots, N-1$ and $u = 0, 1, \ldots, N-1$. The average Fourier components are defined as

$$F(u) = \frac{1}{L} \sum_{i=1}^{L} F(u_i) \tag{2}$$

The components of one-dimensional Fourier transform (power spectrum) are obtained along the four major directions, i.e. horizontal (0°), first diagonal (45°), vertical (90°) and second diagonal (135°). For verifying whether increasing the number of directions helps in producing a better classification, FFT is also performed on four more directions: 22.5°, 67.5°, 112.5° and 157.5° to the horizontal, respectively.

Some Remarks. If the FFT components are represented graphically, then the locus of the power spectrum appears as a continuous curve. Apparently, the curve has a fractal nature (Fig. 3). Note that the fractal dimensions [17,18] of the locii of the FFT coefficients have been calculated by the box counting method. The fractal dimension is the slope obtained by performing a least square fit to the data set $\{\ln(L), -\ln(N(L))\}$, with $N(L) = \sum_{m=1}^{N} (1/m)P(m,L)$, where N is the number of possible points in the box (square of side L) containing m points with probability $P(m,L)$. However, the fractal dimensions did not reveal many distinguishing characteristics between the different classes of fingerprint patterns.

3. MLP Based Classification

A four layered MLP (multi-layer perceptron) [5,19] (Fig. 5) with suitably chosen architecture is used for classifying the fingerprint patterns. It accepts the FFT components along all the different bands as input, and produces output signifying the presence of some particular class. The number of output nodes is equal to the number of fingerprint categories to be classified, i.e. five in the present case. The size of the input layer of the network is equal to the product of the number of the frequency components and the number of directional bands. For example, if there are four bands, each having 64 components, then the number of input nodes is equal to 4×64. The input nodes of the network are arranged in the form of a two-dimensional array, the number of columns representing the number of frequency components and the number of rows representing the number of directional bands. The contiguous activation pattern of the input nodes along a particular row represents the power spectrum of the input fingerprint for the corresponding directional band. The higher layers of the network incrementally group the features to coarser level descriptions from finer level details.

The first hidden layer integrates the frequency components within certain frequency bands in all directions. To achieve this, the connections from each node in the first hidden layer are restricted over a certain neighbourhood of the input layer representing the band of frequency over which the FFT components are grouped. The zones of attention (i.e. the neighbourhood zone of input layer over which the connections are restricted) of the first layer hidden nodes are overlapped. The shape of neighbourhoods is chosen to be rectangular. Let each neighbourhood be of size $m \times n$. Let p_x and p_y be the fraction of overlaps in the two orthogonal directions, respectively. Then $p_x m$ and $p_y n$ neurons (of the second layer) send activations to two neighbouring neurons in the third layer. Let the size of the second layer be $M \times N$. In that case, each pair of two neighbouring neurons in the third layer corresponds to a gap of $m(1-p_x)$ neurons in the second layer in one direction and $n(1-p_y)$ neurons in the other direction. Therefore, the size of the third layer (say, $M' \times N'$) is given by

$$M' = \frac{M}{m(1 - p_x)}$$

and

$$N' = \frac{N}{n(1 - p_y)}$$

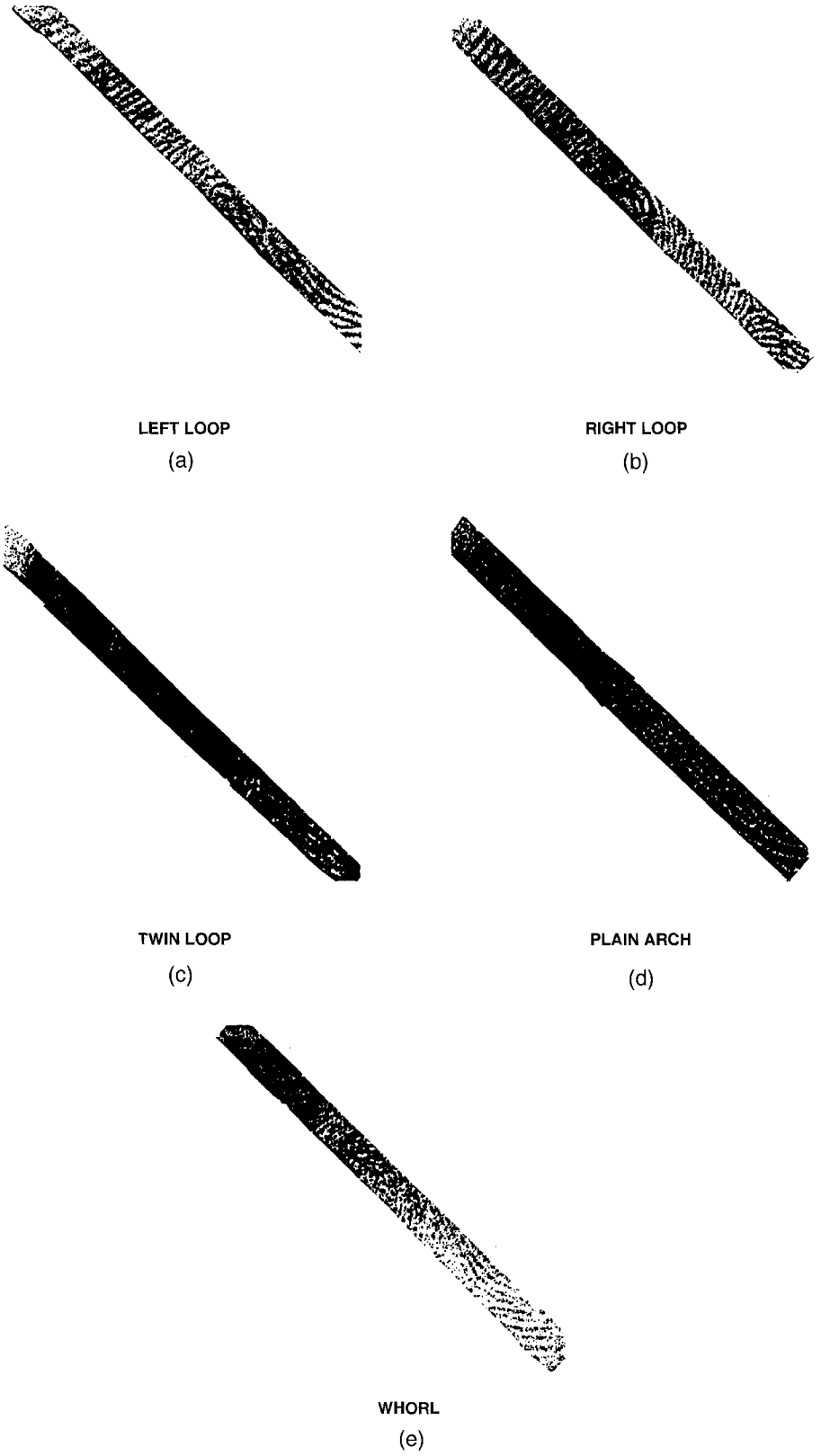


Fig. 2. (a–e) The five fingerprint classes along the left diagonal band.

Similarly, the entities represented by the first hidden layer (i.e. collective effects of frequency components over certain bands) are grouped in the second hidden layer to derive more complex and informative macrofeatures. The connectivity pattern between the first and second hidden layers is similar to that between the input and first hidden layer with different p_x and p_y. Therefore,

$$M'' = \frac{M'}{m'(1 - p_x')}$$

and

$$N'' = \frac{N'}{n'(1 - p_y')}$$

Finally, in the output layer, the complex and informative entities in the third layer (i.e. in the second hidden layer) are grouped collectively in order to make the final decision. Each output node, representative of a particular fingerprint category, is fully connected to the third layer in order to extract the global characteristics of the corresponding fingerprint patterns.

4. Implementation and Results

4.1. Obtaining the Pure and Noisy Fingerprint Images

We originally had 50 samples comprising five different types of patterns. With the objective of testing the effectiveness of the method in the presence of distorted images, we generated 220 more patterns by introducing noise into the pure fingerprint patterns. The methods of introducing various types of noise are described below.

4.1.1. Random Noise Generation. A predefined percentage (5, 10, 15, 20, 30, 40 and 50) of pixels were selected and random noise was injected in the corresponding grey 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 σ is the standard deviation of the normal distribution. Thus, if a pixel p with grey value G_p is selected randomly, its new grey value becomes $G_p = G_p + x$, such that $0 < G_p \leq N_g$ (Fig. 6).

4.1.2. Cut Mark. Any two points in the fingerprint image were selected randomly, and the pixels lying on a width b_w joining these two points were set to the highest grey value, N_g. That is, $G_p = N_g$ was used for all pixels p lying along the generated line (of width b_w), to simulate a cut mark on the fingerprint image. The cutmarks were generated in two different orientations (along the first and second diagonals of the image), at 90° difference. These are termed as the *forward* and *reverse* directions respectively (Fig. 6).

4.1.3. Missing Information. To model the occurrence of loss of information in some portion of the fingerprint image, we selected a portion of the image randomly. Setting all pixels within this portion to the highest (N_g) or lowest (1) grey value simulates the loss of information in that region. So, $G_p = N_g(1)$ for all pixels p lying within the randomly selected part of the image. Note that, setting $G_p = N_g$ models the case for insufficient inking (information loss – white) of the fingerprint in the region concerned, while, setting $G_p = 1$ simulates the condition of excess inking (information loss – black) or blotches or smudging (Fig. 6).

4.2. FFT Coefficients and Feature Selection

The frequency components up to 256 Hz, in the four directions, are found to be distinct for each of the five classes. Figure 3 shows the FFT coefficients from one representative sample of each class. In accordance with the direction of the ridges in the fingerprints, the FFT peaks show prominence in a particular orientation for a specific class. For example, for the left loop fingerprint pattern, the peak prominence is more corresponding to the left diagonal and vertical directions. On the contrary, for the right loop pattern, the maximum peak prominence is found in the right diagonal and horizontal directions. For the twin loop pattern, peak prominences are therefore present in all four directions. In the cases of plain arch and whorl patterns, the peak prominences are visible in all four directions, but are more or less distinct for the classes.

These 256 features in four directions (256×4) i.e. 1024 features are used as input vector to the MLP. To simplify the representation, histograms are drawn as the overall (computed for all the 50 training patterns) maximum number of the peak frequencies in the four directions. For this, eight successive FFT coefficients are averaged. Therefore, in Fig. 4, the first 32 points on the abscissa represent the horizontal direction, the next 32 points denote the left diagonal, the third 32 points depict the right diagonal and the final 32 points are plotting the vertical direction.

As described in Section 2.2, 256 features are also

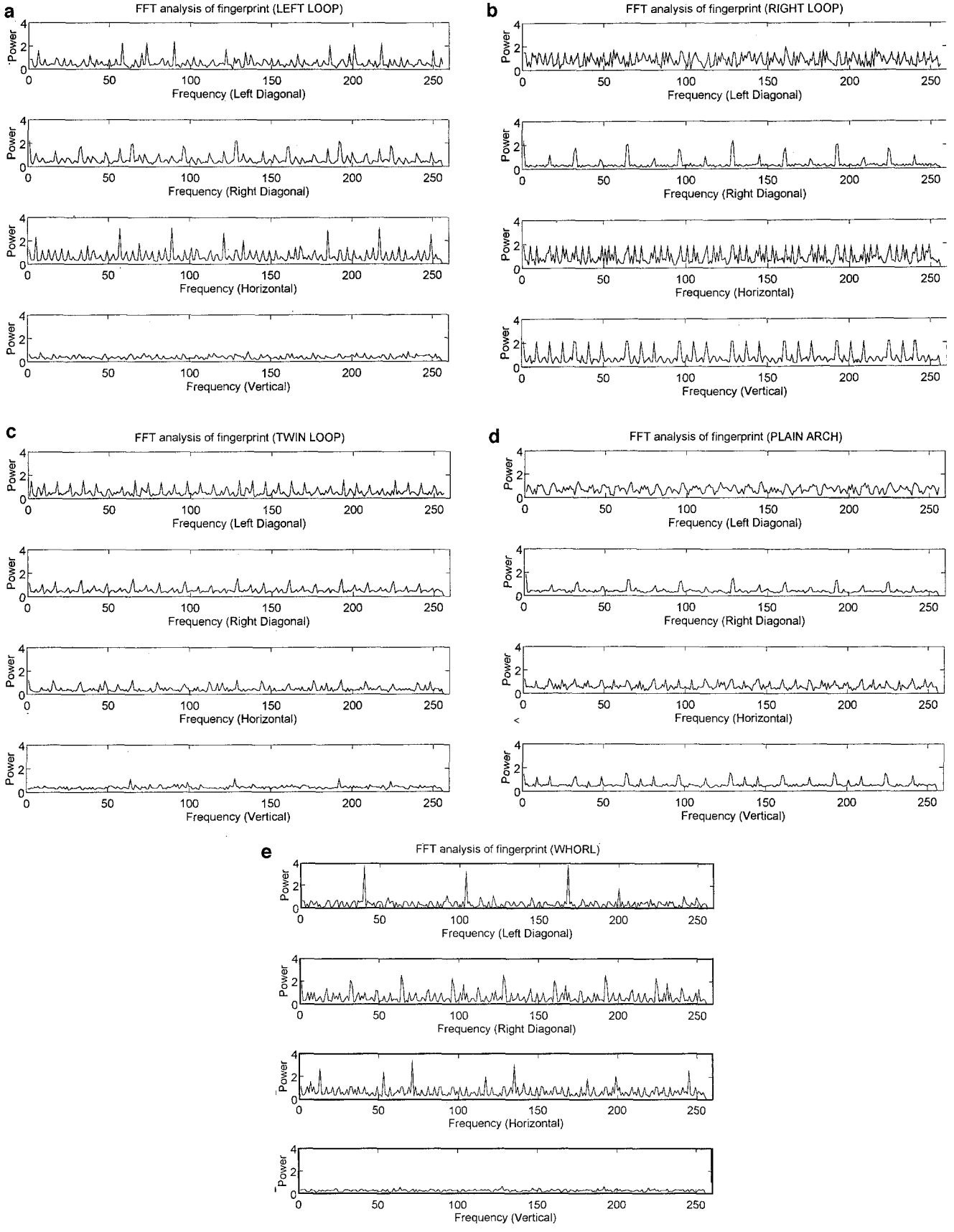


Fig. 3. (a–e) The FFT coefficients of the various types of fingerprints, in four directions.

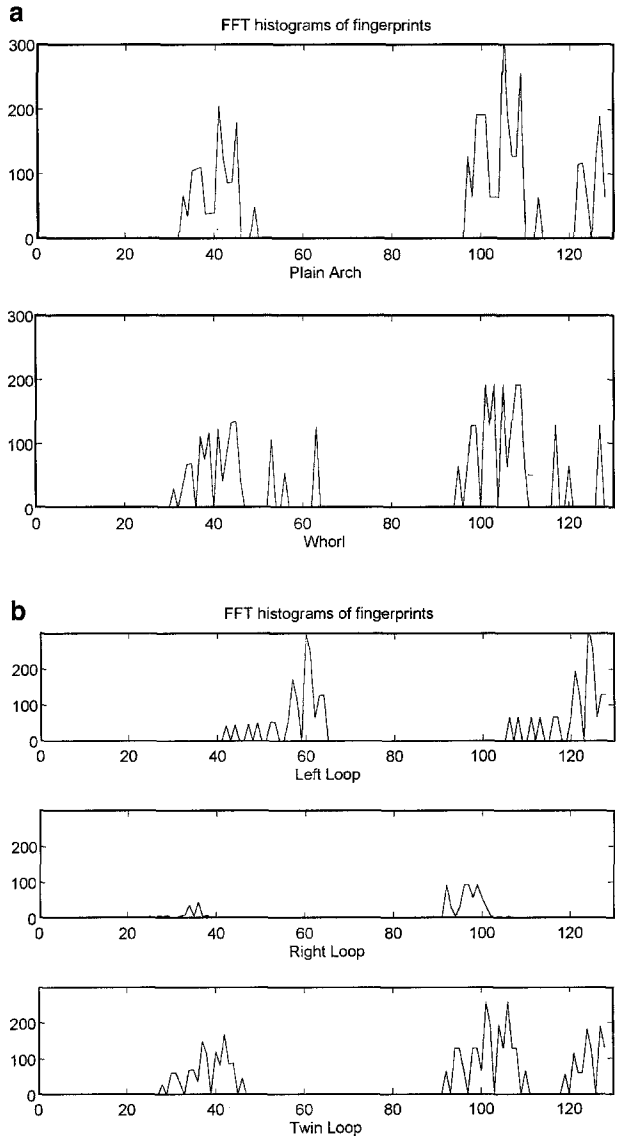


Fig. 4. (a–b) The histogram of the FFT coefficients of the different types of fingerprints.

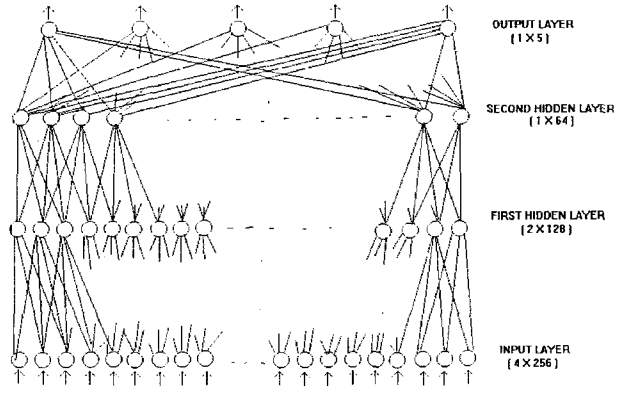


Fig. 5. Schematic diagram of Multilayer Perceptron (MLP).

taken in eight directions, 0°, 22.5°, 45°, 67.5°, 90°, 112.5°, 135° and 157.5° to the horizontal. However, as the number of features increases tremendously, the average of four successive FFT coefficients are taken in four and eight directions, i.e. 64×4 (256) and 64×8 (512) input features, respectively. The average of eight successive FFT coefficients are also computed in the described four and eight directions, i.e. 32×4 (128) and 32×8 (256) input features, respectively.

4.3. Net Parameters and Training

The MLP-based classification scheme is subjected to online learning using a standard back propagation algorithm. The thresholds of all neurons are set to zero. The following architecture is selected empirically. There are altogether four layers (two hidden layers). For 4×256 features, the input layer has 4×256 nodes. The first hidden layer consists of 2×128 nodes, and each node in this layer is connected to 2×32 nodes in the input layer. The second hidden layer consists of 1×64 nodes, and each node is connected to 2×16 nodes in the first hidden layer. The output layer, consisting of five nodes (corresponding to five different classes) is fully connected to the second hidden layer. For 8×256 features, 4×64 features, 8×64 features, 4×32 features and 8×32 features, the number of nodes in each layer and the connectivity pattern is adjusted proportionately.

For training the network, ten (pure) images for each of the five fingerprint classes are selected and used as input patterns. For testing, a set of 220 noisy data generated (as described in Section 4.1) from the five classes is used.

4.4. Results

After training with 500 iterations, zero misclassification is obtained for the training data set. The learning rate is initially considered to be 0.1. It is then changed after every 100 iterations according to the schedule 0.05, 0.02, 0.01 and 0.001, respectively.

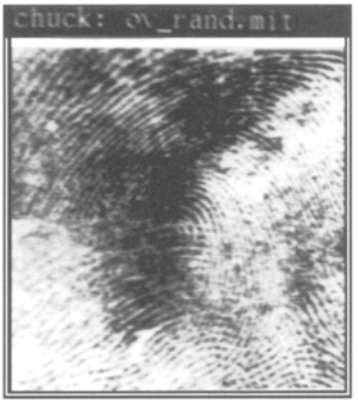
During the testing phase, the fingerprint images are corrupted with different types of noise, as described in Section 4.1. Tables 1 and 2 show the performance of the method for different types of noisy data. Table 1 shows the classwise recognition score of the noisy test data, with 4×256 input features. The score is seen to be maximum for the left-loop class and minimum for the right-loop class.

The variation of overall performance with different types of noise is shown in Table 2. It can be

cutmark (forward)
(a)

under inking
(b)

smudging
(c)



20% random noise
(d)

50% random noise
(e)

Fig. 6. Some of the varieties of noise introduced into the data, cutmark (forward), under inking, smudging, 20% random noise and 50% random noise.

Table 1. Classwise recognition score of noisy data in number and percent.

Fingerprint type	Total number of patterns	Patterns recognised	
		Number	(%)
Left Loop	44	40	90.9
Right Loop	44	34	77.27
Twin Loop	44	37	84.09
Plain Arch	44	37	84.09
Whorl	44	37	84.09
Overall	220	185	84.09

seen from the table that the best results are obtained with 256 features in the 4 and 8 directions. Averaging four successive FFT coefficients (in the 4 and 8 directions) marginally reduces the recognition score. The performance of the MLP is worst in the cases of 32 features (i.e. when averaging eight successive coefficients) in the 4 and 8 directions. This indicates that a balance has to be struck between the number of features and the number of directions of the input features for arriving at a good classification.

The patterns corrupted with random noise (even up to 20%) are recognised correctly with 256×4 features. Increasing the noise worsens the performance of the net. As an illustration, the confusion matrix corresponding to 30% random noise is shown in Table 3. Here the diagonal entries signify the number of patterns correctly classified and the other entries indicate the classes to which misclassifications (confusion) occur.

Interestingly, excellent recognition is obtained even with cutmarks (both forward and reverse) and with smudging. However, with under inking, which introduces more background (white) pixels, the results are comparatively inferior. These findings are similar to the earlier investigation [11] using fuzzy geometrical features.

4.4.1. Comparative Study. The results are compared with those of the k-nearest neighbours (k-NN) classifier [20], with $k = 1,3,5$, and also with two MLP based methods [12,11] using fuzzy and directional features, reported recently.

The k-NN classifier is reputed to be able to generate piecewise linear decision boundaries, and is thus quite efficient in handling concave and linearly nonseparable pattern classes. Therefore, a comparison of the performance of the neural net model with that of the k-NN classifier is highly appropriate. The k-NN classifier is practical when large amounts of memory and sufficient computation power are available for a rapid single trial learning. For good generalisation, the distance between the stored exemplar and input patterns is computed by Euclidean distance metrics E as

$$E_{i,j} = \sum_{d=1}^{D} |T_d^i - U_d^j|$$

Table 2. Variation of overall recognition score (%) with noise.

Noise type features	Random noise							Cut marks		Information loss	
	5%	10%	15%	20%	30%	40%	50%	forward	reverse	black	white
4×256	100	100	100	100	70	10	20	100	100	100	85
8×256	100	100	100	100	60	10	10	100	100	100	80
4×64	100	100	90	70	50	10	10	100	100	80	80
8×64	100	100	100	75	50	10	10	100	100	100	80
4×32	100	80	70	70	50	10	10	80	75	70	65
8×32	100	80	80	70	50	10	10	80	70	60	60

Table 3. Confusion matrix for 30% noise (four images in each class).

	Type	Recognised as				
		Left loop	Right loop	Twin loop	Plain arch	Whorl
Actual class	Left Loop	4	0	0	0	0
	Right Loop	1	2	0	0	1
	Twin Loop	0	0	4	0	0
	Plain Arch	1	1	0	2	0
	Whorl	1	0	1	0	2

Table 4. Comparison of overall recognition score (%) with k-NN taking 4 × 64 features.

Noise type classifier	Random noise							Cut marks		Information loss	
	5%	10%	15%	20%	30%	40%	50%	forward	reverse	black	white
MLP	100	100	90	70	50	10	10	100	100	80	80
k-NN ($k=1$)	100	100	80	60	40	10	10	100	100	80	80
k-NN ($k=3$)	65	65	60	50	30	10	10	80	80	70	60
k-NN ($k=5$)	55	50	50	50	30	10	10	80	80	60	60

where D is the dimension of the feature vectors, T^i is the ith test vector and U^j is the jth training vector. The results obtained by k-NN are not as good as those obtained by MLP, though they are comparable. Both types of classifier prove that the classes are well separable in the feature space selected here (Table 4).

The results of the proposed method, on fingerprints corrupted with 10% random noise, are compared with those of the earlier investigations [12,11] in Table 5. In an earlier investigation [12] on the classification of only three classes of fingerprint patterns with fuzzy MLP, the recognition scores varied between 54.4% and 82.9% in images corrupted with random noise (up to 10%) only. This establishes the better noise sensitivity of the present method.

In another recent investigation [11] of fingerprint classification with 128 fuzzy geometrical and 27 textural features using an MLP, 1000 sweeps were required to recognise 100% of the training set. Reducing the features to 80 required 50,000 iterations for 96.8% correct recognition of the training set. The overall performance with the test set was about 80%. In the case of textural features, 27 features required 150–200 sweeps for 70–84% correct recognition. This indicates the superiority of the present method in terms of the number of required sweeps and recognition score. In that investigation also [11], the worst recognition scores were obtained from the data corrupted with random noise (up to 10%). This further strengthens the noise sensitivity of the proposed method.

5. Conclusions

A multilayer perceptron is used for the classification of fingerprint images using directional FFT components as input features. Five fingerprint categories (Left Loop, Right Loop, Twin Loop, Plain Arch and Whorl) are considered. The results demonstrate that taking FFT directly from the unprocessed grey level images can be helpful in automated classification of the various fingerprint types. Computing the FFT over a set of only four directional bands is adequate for classification. Increasing the number of directions (e.g. to eight) does not necessarily give better performance. The FFT-based textural features are much less sensitive to random noise than those used in earlier approaches [12,11]. It was also found that MLP-based classifier is better than the k-NN classifier for this problem. As has been pointed out elsewhere [15], the computation of a two-dimensional FFT for each training and each test image is time consuming. From the present investigation, it is apparent that computing the one-dimensional FFT can be much faster, as well as effective. However,

Table 5. Classwise comparative recognition score (%) for 10% noise.

	Investigation features	Present study			Ref. [12]				Ref. [12]			
		FFT Coeff. X directions			for Fuzzy MLP				Fuzzy geometrical			Tex. & Dir.
		256×4	64×4	64×8	44	40	36	27	128	80	8	27
Finger Print Types	Left Loop	100	100	100	70.7	100	87.8	48.7	100	100	0	100
	Right Loop	100	100	100	97.5	100	68.3	100	100	100	33.3	0
	Twin Loop	100	100	100	–	–	–	–	100	100	100	0
	Plain Arch	100	100	100	–	–	–	–	71.4	57.2	28.6	57.2
	Whorl	100	100	100	82.9	48.7	82.9	48.7	60	60	40	20
	Overall	100	100	100	83.7	82.9	79.6	65.8	81.3	75	43.8	37.5

the quality of the input data plays a very crucial role in the classification rates.

The fractal dimensions of the FFT coefficients' curves could not be very discriminatory for classifying the fingerprints. Although the FFT features used here reflect the global information of an image, the size of the windows for the FFT can be reduced easily to obtain local features. Further investigations with naturally distorted and overlapping fingerprints are being contemplated.

Acknowledgements. This work is supported by a Project Grant (No. 22(235)/93/EMR-II) of the Council for Scientific and Industrial Research (CSIR), New Delhi, India. Dr. S. N. Sarbadhikari is a Research Associate in this project.

References

1. Malleswara Rao TCL. Feature extraction for fingerprint classification. Pattern Recognition 1976; 8: 181–192
2. Moayer B, Fu KS. A syntactic approach to fingerprint pattern recognition. Pattern Recognition 1975; 7: 1–23
3. Isensor DK, Zaky SG. Fingerprint identification using graph matching. Pattern Recognition 1986; 19: 113–122
4. Karu K, Jain AK. Fingerprint classification. Pattern Recognition 1996; 29: 389–404
5. Rumelhart DE, McClelland JL (eds). Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Vol. 1. 1986. MIT Press, Cambridge, MA
6. Kamijo M. Classifying fingerprint images using neural network: Deriving the classification state. Proceedings of IEEE International Joint Conference on Neural Networks, San Francisco, CA, 1993; 1932–1937
7. Baldi P, Chauvin Y. Neural networks for fingerprint recognition. Neural Computation 1993; 5: 402–418
8. Wilson CL, Candela GT, Watson CI. Neural network fingerprint classification. J Artif Neural Networks 1993; 1: 1–25
9. Pal SK, Ghosh A. Fuzzy geometry in image analysis. Fuzzy Sets and Systems 1992; 48: 23–40
10. Pal SK, Leigh AB. Motion frame analysis and scene abstraction: Discrimination ability of fuzziness measures. J Intelligent and Fuzzy Systems 1995; 3: 247–256
11. Pal SK, Mitra S. Noisy fingerprint classification using multilayer perceptron with fuzzy geometrical and textural features. Fuzzy Sets and Systems 1996; (in press)
12. Mitra S, Pal SK, Kundu MK. Fingerprint classification using fuzzy multilayer perceptron. Neural Computing and Applications 1994; 2: 227–233
13. Haralick RM, Shanmugam K, Dinstein I. Textural features for image classification. IEEE Trans Systems, Man Cybernetics 1973; 3: 610–621
14. Gonzalez RC, Woods RE. Digital Image Processing. Addison-Wesley, Reading, MA, 1992
15. Coetzee L, Botha EC. Fingerprint recognition in low quality images. Pattern Recognition 1993; 26: 1441–1460
16. Henry ER. Classification and uses of fingerprints. Routledge & Sons, London, 1900
17. Voss R. Random fractals: characterization and measurement. In: Pynn R, Skjeltorp A (eds), Scaling Phenomena in Disordered Systems. Plenum Press, New York, 1986
18. Keller JM, Chen S, Crownover RM. Texture description and segmentation through fractal geometry. Computer Vision, Graphics and Image Processing 1989; 45: 150–166
19. Basak J, Pal NR, Pal SK. A connectionist system for learning and recognition of structures. Neural Networks 1995; 8: 643–657
20. Duda RO, Hart PE. Pattern Classification and Scene Analysis. John Wiley, New York, 1973