The CSI Journal on
Computer Science and Engineering
Vol. 3, No. 4&5, 2005
Pages 19-26
Short Paper

Fingerprint Classification Based on Spectral Features

Hossein Pourghassem Hassan Ghassemian

Department of Electrical Engineering, Tarbiat Modarres University, Tehran, Iran.

Abstract

Fingerprint is one of the most important indexes that can be applied to human verification and identification. In the recent decade, with the population explosion and tremendous increase in fingerprint databases, automation of identification has been unavoidable. Fingerprint classification decreases the time of search for an unknown image in large databases. In this research, translation and rotation invariant features are extracted from spectrum of the fingerprint image, and a new tessellation on the frequency spectrum based on an appropriate definition of frequency concept in the fingerprint application, has been designed. The extracted features obtain not only information from frequency of ridges but also valuable information from direction of ridges in the fingerprint images. Features are classified with Probabilistic Neural Network. FVC2000 and FVC2002 databases are used to assess the proposed algorithm. The proposed algorithm provides an accuracy and speed of classification more than previously reported in the literature. We have obtained an accuracy of 93.4 percent with a rejection ratio 2.8 percent for the seven-class task and for the six-class classification task an accuracy of 95.1 percent with a rejection ratio 2.4 percent is achieved.

Keywords*:* fingerprint, classification, probabilistic neural network, feature extraction, spectral features

1. Introduction

Fingerprint; have been widely used for personal identification for several centuries. Many organizations are using fingerprints for criminal investigation, access control of restricted areas, driving license applicants, and personnel electronic passports [1]. An automatic recognition of people based on fingerprints requires that the input fingerprint be matched with a large number of fingerprints in a database. To reduce the search time and computational complexity, it is desirable to classify fingerprints in an accurate and consistent manner such that the input fingerprint needs to be matched only with a subset of the fingerprints in the database. Automatic fingerprint classification is a technique used to assign a fingerprint into one of the several prespecified classes of fingerprints.

The most widely used classification method is based on Henry's classification, which consists of eight classes: Arch, Tented arch, Left loop, Right loop, Whorl, Central packet loop, Twin loop and Accidental. Figure 1 illustrates the fingerprint images that belong to these different classes. Here, the Accidental class defines the same conditions as rejected fingerprints. Thus, only seven classes are considered in practical fingerprint classification systems.

Fingerprints are the characteristic structure of flow lines of ridges and furrow that are present on the skin on one's finger. There are two main types of features in a fingerprint: 1) global ridge and furrow structures, which from special patterns (singular points), and 2) local ridge and furrow structures (on such is the ridge endings and bifurcations, also known as minutiae).

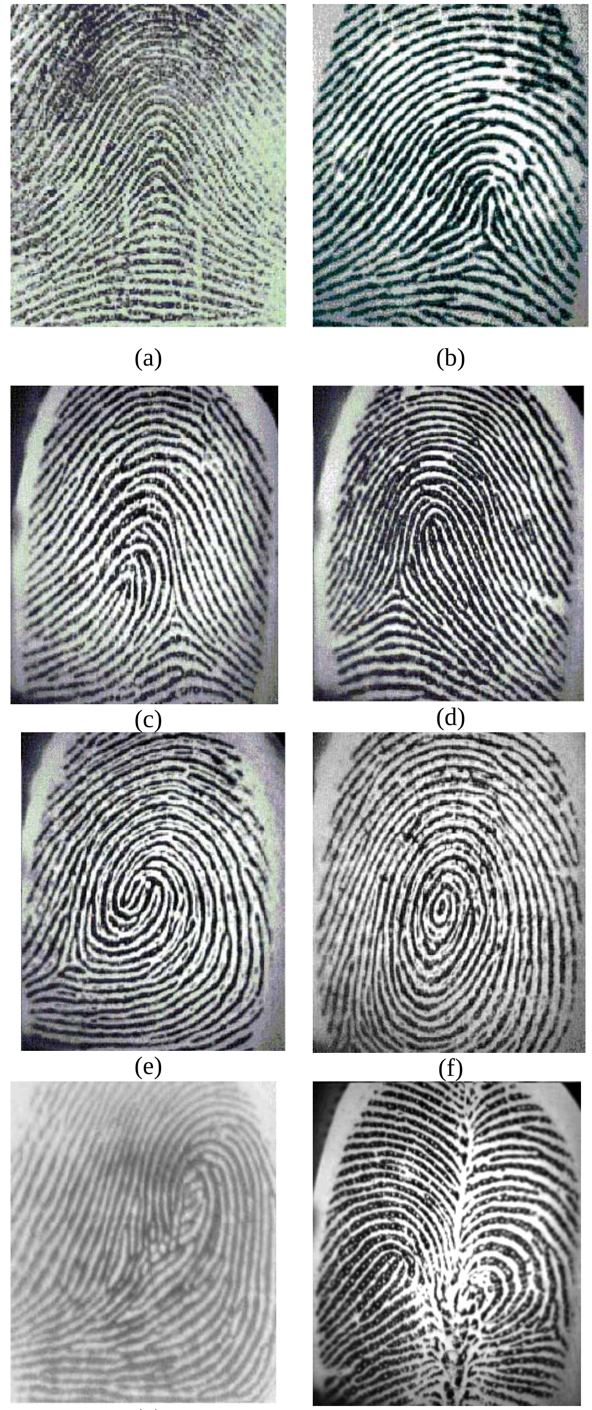
Singular points (cores and deltas) are points of discontinuity of the flow field. The two types of singular points are defined in terms of the ridge structures; the core is the end point of the innermost curving ridge while the delta is the confluence point of three different flow directions (See Figure 2) [2].

Several approaches have been developed for automatic fingerprint classification. They are the model-based, structure-based, frequency-based and syntactic approaches [3]. The model-based fingerprint classification technique uses the locations of singular points to classify a fingerprint [4, 5]. Zhang and Yan [5] introduced an approach to fingerprint classification based on extraction and analysis of both singular points and traced pseudo ridges relating to singular points. In this article, with the help of pseudo ridge tracing and an analysis of the traced curves, their method didn't rely on the extraction of the exact number and positions of the true singular points, thus improving the classification accuracy. They achieved 92.7 percent classification accuracy for a four-class classification problem. A structure-based approach uses the estimated orientation field in a fingerprint image to classify the fingerprint [3, 6-8]. An accuracy of 94.8 percent with 4.2 percent rejection for five-class classification problem is reported [3]. Shah and Sastry [7] used feature vectors obtained from the output of the oriented line detector. Their line detector gives the ridgelines along with the orientation information at each point. Their feature vector is based on dividing the line-detected image into zones and clustering the distribution of orientations in each zone. Jain and Prabhakar [8] achieved 90 percent classification accuracy (with 1.8 percent reject at feature extraction stage) for a five-class classification problem. Their classification accuracy was further increased to 96 percent after a total 32.5 percent of the images were rejected. A syntactic approach uses a formal grammar to represent and classify fingerprints [9-12]. Nagaty [9] extracted a string of symbols using the block directional image of the fingerprint, which represented the set of structural features for this string and its Euclidean distance measures were computed by using this moment. An accuracy of 95.6 percent with 20 percent rejection for five-class classification problem is reported [11]. Frequency-based approaches use the frequency spectrum of the fingerprints for classification [13]. A Hexagonal Fourier Transform is applied in [13] that classify fingerprints into whorls, loops and arches. Fitz and Green [13] have proposed a Fourier transform method to locate the core points. The Hexagonal Fourier Transform allows the utilization of hexagonally sampled data and the extention of output data in a rectangular scheme.

The frequency domain features are used for classification through nearest-neighbor method.

In this paper, we propose a fingerprint classification algorithm based on a novel representation scheme, which is derived from the frequency spectrum of the fingerprint. The representation does not use the core, delta and orientation field. The main steps in our classification algorithm are as follows:

1. Extract the frequency spectrum of the fingerprint image.
2. Apply the band pass filter on the frequency spectrum.
3. Extract the translation and rotation invariant feature from the frequency spectrum.
4. Feed the feature vector in to a Probabilistic Neural Network.



(a) (b) (c) (d) (e) (f) (g) (h)

Figure 1. Major fingerprint classes. (a) Arch, (b) Tented arch, (c) Left loop, (d) Right loop, (e) Twin loop, (f) Whorl, (g) Central packet loop,(h) Accidental

Section 2 explains the feature extraction scheme. Section 3 introduces classifier. Section 4 shows experimental results on the FVC2000 and FVC2002 databases [14]. The concluding remarks and feature research directions are presented in section 5.

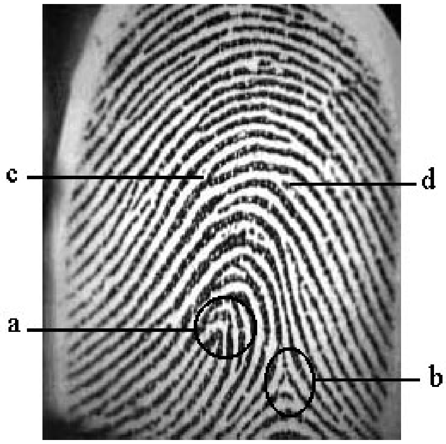


Figure 2. Ridges, minutiae, and singular points.(a) core, (b) delta, (c) bifurcation, (d) ridge ending.

2. Fingerprint Feature Extraction

A structure-based approach uses the estimated orientation field in s fingerprint image to classify the fingerprint, which is directly derived from local ridge structures. In the spatial domain, feature extraction is a complex and time consuming task. However, features can be easily extracted from spatial spectrum.

The Discrete Fourier Transform of fingerprint image $f(x,y)$ of size $M \times N$ is given by the following equation.

$$F(u,v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) e^{-j2\pi(\frac{ux}{M} + \frac{vy}{N})} \quad (1)$$

It is usually convenient to express in polar rather than Cartesian form,

$$F(u,v) = |F(u,v)| e^{-j\phi(u,v)} \quad (2)$$

where $|F(u,v)|$ is the magnitude of the frequency spectrum and $\phi(u,v)$ is the phase angle.

$$|F(u,v)| = \sqrt{R^2(u,v) + I^2(u,v)} \quad (3)$$

$$\phi(u,v) = \tan^{-1}\left[\frac{I(u,v)}{R(u,v)}\right] \quad (4)$$

where $R(u,v), I(u,v)$ are real part and imaginary part of the $F(u,v)$. Figure 3 shows three example fingerprints and their corresponding special spectrums.

Each pixel in the spectrum represents a change in the spatial frequency of one cycle per image width. The origin (at the centered of the ordered image) is the DC term of $f(x,y)$.

Consider an image $f_2(x,y)$ that is a rotated and translated replica of $f_1(x.y)$,

$$f_2(x,y) = f_1[(x\cos\alpha + y\sin\alpha) - x_0, (-x\sin\alpha + y\cos\alpha) - y_0] \quad (5)$$

where α is rotation angle and x_0 and y_0 are translational offsets. The Fourier transforms of $f_1(x,y)$ and $f_2(x,y)$ are related by,

$$F_2(u,v) = e^{-j\phi(u,v)} F_1[(u\cos\alpha + v\sin\alpha), (-u\sin\alpha + v\cos\alpha)] \quad (6)$$

where $\phi(u,v)$ is the spectral phase of the image $f_2(x,y)$. The phase depends on the translation and rotation, but the spectral magnitude,

$$|F_2(u,v)| = |F_1[(u\cos\alpha + v\sin\alpha), (-u\sin\alpha + v\cos\alpha)]| \quad (7)$$

is translation invariant. Equation (7) shows that a rotation of the image rotates the spectral magnitude by the same angle, therefore if $f^0(i,j)$ is the feature vector for an image I then for image I^θ obtained by rotation of I by angle θ, the feature vector $f^\theta(i,j)$ is related to $f^0(i,j)$ by the expression $f^\theta(i,j) = f^0(i, j-\theta)$. Thus the feature vector for a particular image will have circular shift proportional to the rotation angle.

2.1 Filter Bank Structure

Features are extracted from the upper half of special spectrum, because it has the central symmetrical property. On the other hand the second half of special spectrum is redundant (Figure 3). Applying the band pass filter on the special spectrum not only cancels the image's noise but also reduces the number of features. The lower cut off frequency of the filter is determined empirically to f=4/15(1/mm) (correspond to 4th sample) and the upper cut off frequency of the filter is determined by $f_{\max}$ in the image. The frequency resolution is defined as:

$$\Delta f = \frac{1}{N\Delta x} \quad (8)$$

where N is image length (e.g., N=300), Δx is image resolution (e.g., 500dpi or $\Delta x = 0.05mm$) and $f_{\max}$ is defined as:

$$f_{\max} = \frac{n_v}{L_i} \quad (9)$$

where n_v is the maximum variation of intensity in the fingerprint image (e.g., $n_v = 60$) and L_i is image width (e.g., $L_i = 15mm$), thus $n_{\max}$ is:

$$n_{\max} = \frac{f_{\max}}{\Delta f} \quad (10)$$

therefore in our experimental, the upper cut off frequency equals 60th sample.

Features are extracted from the filtered image by a novel tessellation (Figure 4). This tessellation is defined as:

$$S_{ij} = \{(u,v) \mid R_i < r < R_{i+1}, \theta_j < \theta < \theta_{j+1}, 1 \le u \le N, 1 \le v \le M\} \quad (11)$$

where

$$r = \sqrt{(u-u_c)^2 + (v-v_c)^2} \quad (12)$$

$$R_i \in \{R_1, R_2, .., R_n\} \quad (13)$$

$$\theta = \tan^{-1}\left(\frac{v-v_c}{u-u_c}\right) \quad (14)$$

$$\theta_j \in \{0, \pi/4, \pi/2, 3\pi/4, \pi\} \quad (15)$$

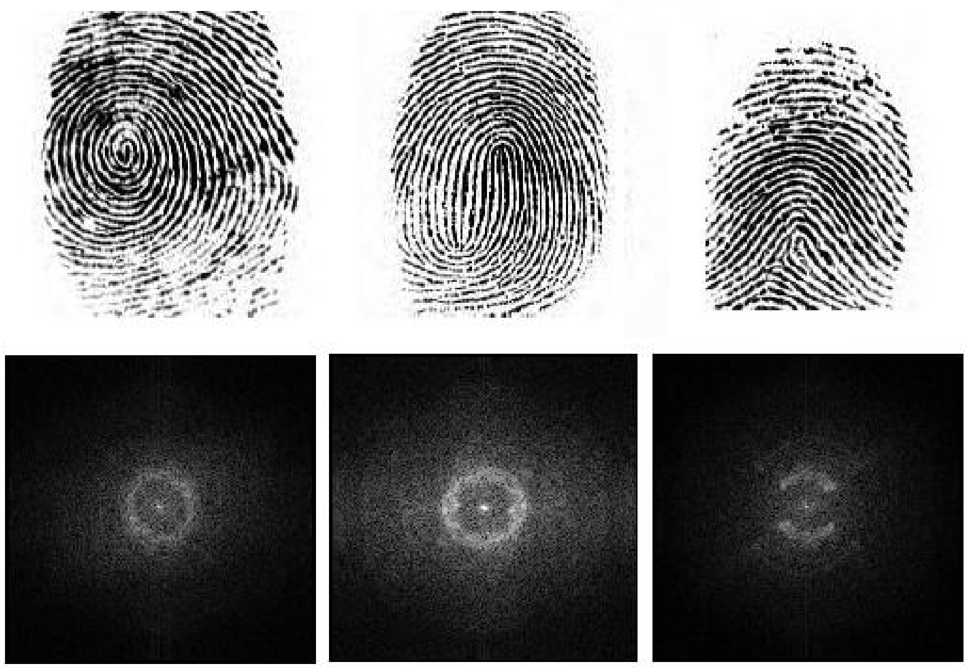


Figure 3. Shows three example fingerprints and their corresponding special spectrums

$N \times M$ is the image size, (u_c, v_c) is the center of frequency spectrum. R_i $(i = 1,2,..,n)$ is determined in a manner that the total number of pixels in each sector is equal to the others. The numbers of features vary for the different values of the n in (13).

The spatial distribution of the variations in neighborhoods of the component spectral images constitutes a characterization of the global ridge structures and is well captured by the standard deviation of gray scale values. In our algorithm the standard deviation within the sectors defines the feature vector. The standard deviation is defined as:

$$f_{ij} = \sqrt{\sum_{K_{ij}} (I(u,v) - M_{ij})^2} \tag{16}$$

where K_{ij} is the number of pixels in S_{ij} and M_{ij} is the mean of the pixel values S_{ij}. The extracted feature vector for an image is a $(n-1)$ by 4 matrix.

A shift in the time domain by m points corresponds in the frequency domain to multiplication of the Fourier Transform by the linear phase factor e^{-jwm}. Using this we can write

$$f^{\theta}(i, j) \xrightarrow{DFT} e^{-j(2\pi k/N)m} F(i,k) \tag{17}$$

where

$$f(i, j) \xrightarrow{DFT} F(i,k) \tag{18}$$

and N is total length of sequence. This magnitude of the DFT will remain constant for any image orientation only the phase will change. On the other hand the DFT magnitude is independent of the orientation of the image in input sample. The block diagram of our feature extraction algorithm is shown in Figure 4.

3. Fingerprint Classifier

For fingerprint classification problem, we used from Neural Network that Probabilistic Neural Network is more compatible to the extracted features of the fingerprint. Probabilistic Neural Network is a Bayes-Parzen classifier [14]. The PNN was first introduced by Specht [15], who showed how the Bayes-Parzen classifier could be broken up in to a large number of simple processes implemented in a multilayer neural network each of which could be run independently in parallel.

In the Bayes classifier, the density function $f_k(x)$ corresponds to the concentration of class k examples around the unknown example is an important parameter that should be determined. Since the probability density function does usually not known in practice, it is often assumed that they are members of normal distribution. The training set is then used to estimate the parameters of the distribution. However, it is more appropriate to use a nonparametric estimation method such as Parzen windows. In order to classify unknown samples, most common classifiers separate the unknown sample from each know member of the training set using the Euclidean distance. The unknown member is then classified into the population of its nearest neighbor. The Parzen windows technique goes step further in a way that it takes into account more distant neighbors. Parzen technique estimates a bell-shaped Gaussian function for separating an unknown point from the known training sample point. Such function has a higher value if the distance is close and converges to zero if the distance becomes large. Taking the sum of this function for all known training set members, and classifying the unknown point into the population with the largest sum is the main idea of the probabilistic algorithm. Parzen's estimated density function is,

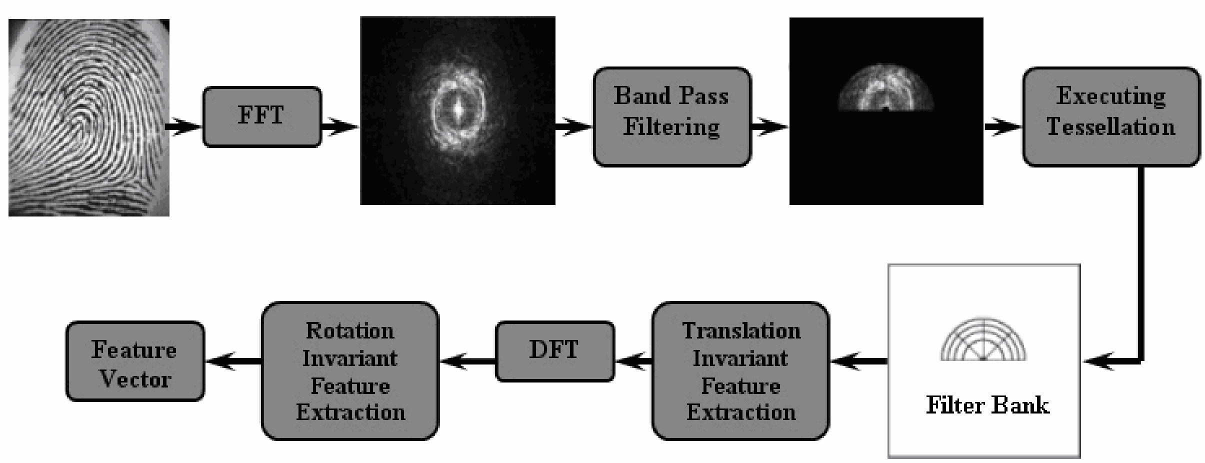


Figure 4. The block diagram of feature extraction

$$g(x) = \frac{1}{n\sigma^p (2\pi)^{p/2}} \sum_{i=0}^{n-1} e^{\frac{|x-x_i|^2}{2\sigma^2}} \quad (19)$$

where n is the total number of training examples σ is scaling parameter that controls the width of the area of influence of the distance.

Although the value of σ is an important smoothing parameter in the probabilistic network since it affects the estimation error; there is no mathematical way of determining it. A too small value of σ gives the same effect as the nearest neighbor technique and a too large σ does not give clear separation of classes and classification cannot be made.

Consider the simple network architecture shown in Figure 5 with four input nodes (p=4) in the input layer, two population classes (class 1 and class 2), five training examples belonging to class 1 (n_1=5), and three examples in class 2 (n_2=3). The pattern layer is designed to contain one neuron (node) for each training case available and the neurons are split into the two classes. The summation layer contains one neuron for each class. The output layer contains one neuron that operates trivial threshold discrimination; it simply retains the maximum of the two summation neurons. The PNN executes a training case by first presenting it to all pattern layer neurons. Each neuron in the pattern layer computes a distance measure between the presented input vector and the training example represented by that pattern neuron. The PNN then subjects this distance measure to the Parzen window and yield the activation of each neuron in the pattern layer. Subsequently, the activation form each class is fed to the corresponding summation layer neuron, which adds all the results in a particular class together. The activation of each summation neuron is executed by applying the remaining part of the Parzen's estimator equation to obtain the estimated probability density function value of population of a particular class. The results from the two summation neurons are then compared and the largest is fed forward to the output neuron to yield the computed class and the probability that this example will belong to that class.

The most important parameter that needs to be determined to obtain an optimal PNN is the smoothing parameter of the random variables. A straightforward procedure involves selecting an arbitrary value of σ's, training the network, and testing it on a test set of examples. This procedure is repeated for other σ's and the set of σ's that produces the least misclassification rate is chosen.

4. Experimental Results

We have reported the results of our fingerprint classification algorithm on the FVC2000 and FVC2002 databases for the seven-class fingerprint classification problem. The FVC2000 and FVC2002 databases consist of 600 fingerprint images. We form our training set with the first 350 images and the test set contains the remaining 250 images.

We trained a Probabilistic Neural Network with training set. The PNN has three layers with n_p (12, 16, 20) input neurons, and seven output neurons corresponding to the seven classes. The smoothing parameter is determined to use trial and error way (Figure 6). The result of the fingerprint classification into seven-class (Arch (A), Tented arch (TA), Left loop (LL), Right loop (RL), Whorl (W), Central packet loop (CL), Twin loop (TL)) with different numbers of features is shown in the Table 1. As it is shown in the Table 1, the best accuracy is achieved for seven-class classification problem, if the number of features is equal to 16. We obtain an accuracy of 93.4 percent with a rejection ratio 2.8 percent for the seven-class task with 16 features. For the six-class classification task (where classes Whorl and Central Packet Loop were merged into one class) an accuracy of 95.1 percent with a rejection ratio 2.4 percent is achieved with 16 features. The confusion matrix for the seven-class classification problem is shown in the Table 2. The confusion matrix for the six-class classification problem is shown in the Table 3.

We also compare our fingerprint classification results with previously reported in the literature. Fingerprint classification algorithm presented by Jain et al. [8], classifies fingerprints in to five categories using a two-stage classifier.
For a five-class fingerprint classification problem, they achieved 90% classification accuracy (with 1.8% reject at feature extraction stage).
Their classification accuracy was further increased to 96%

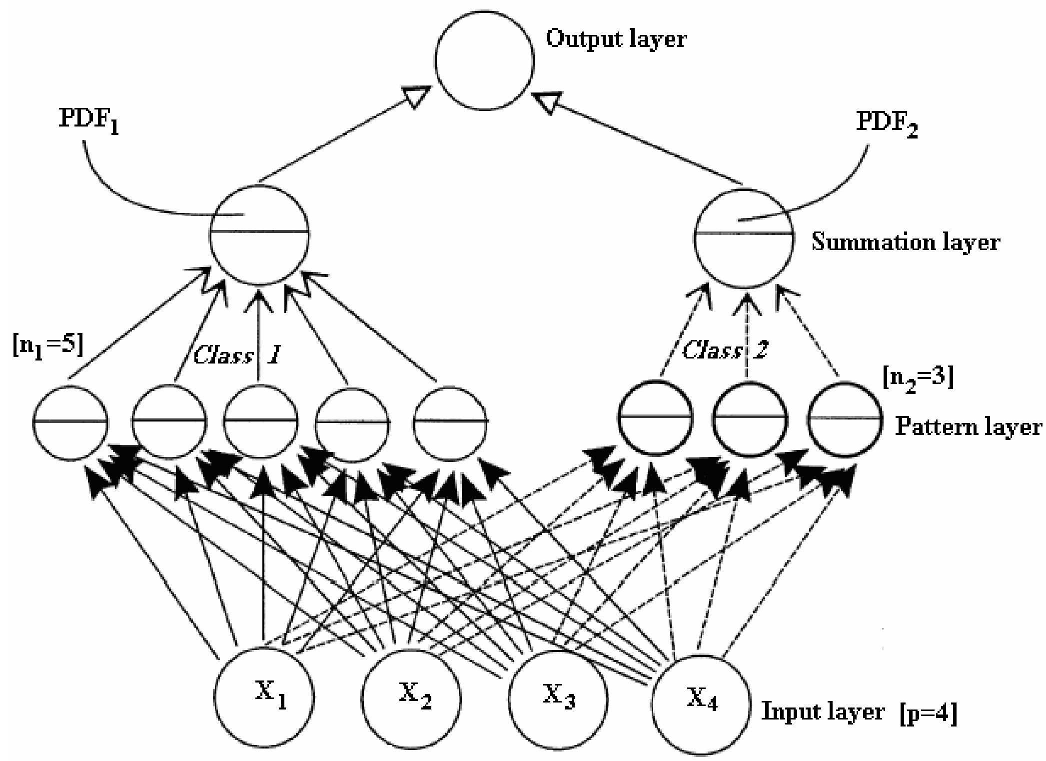


Figure 5. A simple Probabilistic neural network with four input variables, two classes, and eight training exam

after a total of 32.5% images were rejected. The best reported result in [7], [11], [3] and [17], for five-class classification problem is 95.6% accuracy, 95.6% accuracy with 20% rejection rate, 94.8% accuracy with 5.1% rejection rate, and 94% accuracy, respectively. In [5] and [12], for four-class classification problem, 92.7% and 93.9% accuracy is achieved, respectively.

Our classification algorithm has computational complexity less than previously reported in [7-9]. Our classification algorithm classifies one fingerprint image about 0.4 second on a Pentium 1GHz PC. While, the time needed for classifying a fingerprint, in [7] is 13.1 seconds on a Pentium 300MHz PC, in [8] is about 10 seconds on a Sun Ultra-1 machine, and in [9] is 11.5 seconds on a Pentium 100MHz PC. Table 4 shows comparison between the results achieved by a number of fingerprint classification systems that have previously been published and our classification algorithm.

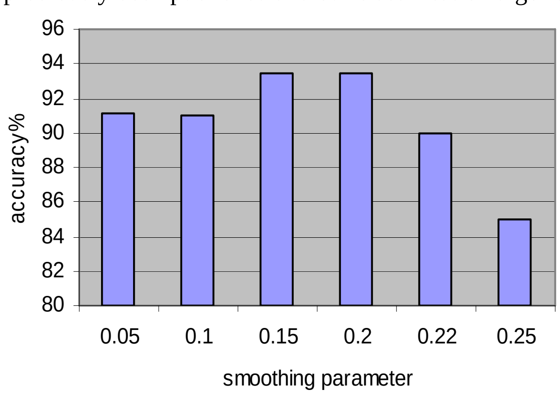


Figure 6. Performance of PNN(with n_p =16) for various values of the smoothing parameter in the Pattern layer

In Table 4, the number of features in each classification algorithm is measure of computational complexity and time consuming.

Table 1. Fingerprint classification into seven-class with different numbers of features

No. of features	Accuracy%	Rejection %
12	88.9	3.1
16	93.4	2.8
20	91.2	2.5

5. Concluding Remarks

In this paper, a new scheme for the fingerprint feature extraction is presented. The extracted feature vector is not only translation and rotation invariant but also robust to noise. Features are classified with Probabilistic Neural Network. We have tested our algorithm on the standard FVC2000 and FVC2002 databases and the achieved performance has been better than previously reported in the literature (93.4% accuracy with 2.8% rejection rate for the seven-class classification task and 95.1% accuracy with 2.4% rejection rate for the six-class classification problem). However, performance of this algorithm will be improved if poor quality images are rejected and incorrect classified images are retrained to the classifier.

Table 2. Confusion Matrix for seven-class problem with 16 features

True Class		Assigned Class TA	LL	RL	W	CL	TL	A	Reject	Accuracy%
TA	**58**	55	0	0	1	1	0	0	1	96.5
LL	**54**	0	49	0	2	2	1	0	0	90.7
RL	**53**	0	0	48	0	1	1	1	2	94.1
W	**24**	0	0	0	21	2	0	0	1	91.3
CL	**20**	0	0	0	2	17	0	0	1	89.4
TL	**22**	0	0	0	0	0	21	0	1	100
A	**19**	0	0	0	1	0	1	16	1	88.8

Table 3. Confusion Matrix for six-class problem with 16 features

True Class		Assigned Class TA	LL	RL	W/CL	TL	A	Reject	Accuracy%
TA	**58**	55	0	0	2	0	0	1	96.5
LL	**54**	0	49	0	4	1	0	0	90.7
RL	**53**	0	0	48	1	1	1	2	94.1
W/CL	**44**	0	0	0	42	0	0	1	100
TL	**22**	0	0	0	0	21	0	1	100
A	**19**	0	0	0	1	1	16	1	88.8

Table 4. A comparison between of published fingerprint classification methods and our method

Authors and Year	Classes	Accuracy %	Reject rate%	Number of features
Chang et al. 2002 [3]	5	94.8	5.1	-
Zhang et al. 2004 [5]	4	92.7	-	-
Shah et al. 2004 [7]	5	95.6	-	224
Jain et al. 1999 [8]	5 5	90 96	1.8 32.5	192 192
Nagaty 2001 [9]	5	95	15.1	186
Yao et al. 2003 [11]	5	95.6	20	192
Senior 2001 [12]	4	93.9	-	256
Fitz et al. 1996 [13]	3	85	-	-
Allah 2005 [17]	5	94	-	-
Our method	6 7	95.1 93.4	2.8 2.4	16 16

References

[1] H. Ghassemian, "Structural Fingerprint Restoration," Lecture Notes in Computer Science, Image Analysis and Processing, Vol. 2, pp. 544-551, 1997.

[2] C.D. Sarat, "Markov Random Field Models for Directional Field and Singularity Extraction in Fingerprint Images," IEEE Trans. on Image Processing, Vol.13, No.10, pp. 1358-1367, 2004.

[3] J.H. Chang and K. Fan, "A new model for fingerprint classification by ridge distribution sequences," *Pattern Recognition*, Vol. 35, No. 6, pp. 1209-1223, 2002.

[4] H. Ghassemian, "An automatic fingerprint classification algorithm," *Esteghlal Journal of Engineering*, Vol.18, No.1, pp.1-11, 1999.

[5] O. Zhang and H. Yan, "Fingerprint classification based on extraction and analysis of singularities and pseudo ridges," *Pattern Recognition*, Vol. 37, No.11, pp.2233-2243, 2004.

[6] S. Wang and W. Zhang, "Fingerprint Classification by Directional Fields," *Proc. 4th IEEE Int. Conf. Multimodal Interface*, pp. 395-398, 2002.

[7] Sh. Shah and P.S. Sastry, "Fingerprint Classification using a Feedback-Based Line Detector," *IEEE Trans. on Systems-Man and Cybernetics*, Vol. 34, No. 1, pp. 85-94, 2004.

[8] A.K. Jain and S. Prabhakar, and L. Hong, "A Multichannel Approach to Fingerprint Classification," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 21, No.4, pp. 348-358, 1999.

[9] K.A. Nagaty, "Fingerprint Classification using Artificial Neural Network: A Combination Structural and Statistical Approach," *Neural Networks*, Vol. 14, pp. 1293-1305, 2001.

[10] J. Gao and H.M. Dong, "Research on Syntactic Fingerprint Classification and Matching," *Proc. of the second Int. Conf. on Machine Learning and Cybernetics*, Vol. 3, pp. 3066-3071, 2003.

[11] Y. Yao and G. Marcialis, and M. Pontil, "Combining Flat and Structured Representations for Fingerprint Classification with Recursive Neural Networks and Support Vector Machines," *Pattern Recognition*, Vol. 36, pp. 397-406, 2003.

[12] A. Senior, "A Combination Fingerprint Classifier," *IEEE Trans. on Pattern Analysis and Machine Intelligence*,

Vol. 23, No.10, pp. 1165-1175, 2001.

[13] A.P Fitz and R.J. Green, "Fingerprint classification using Hexagonal Fast Fourier transform," *Pattern Recognition*, Vol. 29, No.10, pp. 1587-1597, 1996.

[14] Fingerprint Group of Bologna university, Italy, http://bias.csr.it/FVC2000/databases.asp and http://bias.csr.it/FVC2002/databases.asp, May 2004.

[15] T. Masters, *Advanced algorithms for Neural Networks*, New York, John Wiley & Sons, 1995.

[16] D.F. Specht, "Probabilistic neural networks," *Neural Networks*, Vol. 3, pp. 109-118, 1990.

[17] M.M.A. Allah, "A novel approach for fingerprint classification system based on new feature area search," *Int. Journal of Machine Graphics and Vision*, Vol. 14, No. 2, pp. 201-212, 2005.

Hossein Pourghassem was born in Iran in 1980. He received his B.S. degree in Biomedical engineering from Shahed University in 2002 and his M.S. degree in Biomedical Engineering from Tarbiat Modares University in 2004 both in Tehran, Iran. He is currently Ph.D. student in Biomedical Engineering at Tarbiat Modares University. His major fields of interest are pattern recognition, image processing, neural networks, machine vision, biometrics and content-based image retrieval.
E-mail: h_poorghasem@modares.ac.ir

Hassan Ghassemian was born in Iran in 1956. He received the B.S.E.E. degree from Tehran College of Telecommunication in 1980 and the M.S.E.E. and Ph.D. degree from Purdue University, West Lafayette, USA, in 1984 and 1988, respectively. He is a Professor of Electrical Engineering at Tarbiat Modares University in Tehran, Iran. His research interests include Multi-Source Image Processing and Analysis, Information Processing and Pattern Recognition in Remote Sensing and Biomedical Engineering. Dr. Ghassemian is an IEEE Senior member.
E-mail: ghassemi@modares.ac.ir