سیزدهمین کنفرانس مهندسی برق ایران
۲۰-۲۲ اردیبهشت ۱۳۸۴

Fingerprint Classification Based on Spectral Features

Hossein Pourghassem
Tarbiat Modares University
h_poorghasem@modares.ac.ir

Hassan Ghassemian
Tarbiat Modares University
ghassemi@modares.ac.ir

Abstract: *Fingerprint is one of the most important indexes that can be applied for verification and identification. In the recent decade, with the development of societies and databases of fingerprint, automation of identification has been unavoidable. Fingerprint classification decreases the time of search for an unknown image in large databases. The purpose of this research is to increase the number of classes and improvement the accuracy of classification. Fingerprint images are classified into seven classes: Right loop, left loop, Twin loop, Arch, Tented arch, Whorl and Central packet loop. In this research, translation invariant features are extracted from spectrum of the fingerprint image. 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 better than previously reported in the literature.*

Keywords: Fingerprint, Classification, PNN, Feature Extraction.

1. Introduction

Fingerprint; have been widely used for personal identification for several centuries. The main reason for this is that every person is believed to have unique fingerprints, which remain invariant with the age [1]. Presently, several organizations are using fingerprints for not only criminal investigation but also to access control of restricted areas, driving license applicants, and employee identification passports. 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 these 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. Hence, 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 for examples of ridges, minutiae, and singular points in a fingerprint image.

Several approaches have been developed for automatic fingerprint classification. They are the model-based, structure-based, frequency-based and syntactic approaches [1]. The model-based fingerprint classification technique uses the locations of singular points to classify a fingerprint [2]. A structure-based approach uses the estimated orientation field in a fingerprint image to classify the fingerprint [1], [3-5]. A syntactic approach uses a formal grammar to represent and classify fingerprints [6-9]. Frequency-based approaches use the frequency spectrum of the fingerprints for classification [10].

In this paper, we propose a fingerprint classification algorithm (Fig.3) 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 of 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 invariant feature from the frequency spectrum.
4) Feed the feature vector in to a Probabilistic Neural Network.

In the following sections, we will present the details of fingerprint classification algorithm. Section 2 presents our feature extraction scheme. In section 3, we introduce our classifier (PNN). In section 4, we present our experimental results on the FVC2000 and FVC2002 databases. The conclusions and feature research directions are presented in section 5.

2. Feature extraction

2.1 Two-dimensional Fourier Transform

The discrete Fourier Transform of a two-dimensional function (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

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

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.

where the magnitude function $|F(u,v)|$ is called the frequency spectrum of image and $\phi(u,v)$ is phase angle. The frequency spectrum and phase spectrum are defined as,

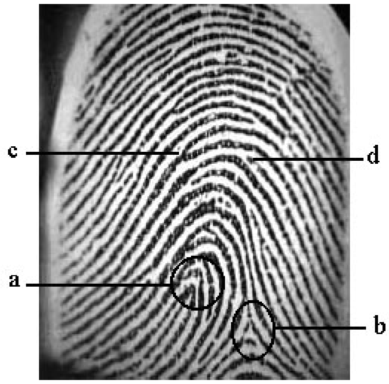

Figure 2: Ridges, minutiae, and singular points.(a) core, (b) delta, (c) bifurcation, (d) ridge ending.

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

where $R(u,v), I(u,v)$ are real part and imaginary part of the $F(u,v)$. In the frequency domain, u represents the spatial frequency along the original image's x-axis and v represents the spatial frequency along the y-axis. In the center of the image u and v have their origin. The spatial frequency of an image refers to the rate at which the pixel intensities change.

Features are extracted from the upper half of frequency spectrum of the fingerprint image, because it has the central symmetrical property (Figure 3). Applying the band pass filter on the frequency 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 to $f_{\max}$ in the image. The frequency resolution is defined as:

$$\Delta f = \frac{1}{N\Delta x} \quad (4)$$

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

$$f_{\max} = \frac{n_v}{L_i} \quad (5)$$

where n_v ($n_v = 60$) is the maximum variation of intensity in the image ($L_i = 15mm$), so $n_{\max}$ (correspond to the $f_{\max}$) is:

$$n_{\max} = \frac{f_{\max}}{\Delta f} = 60 \quad (6)$$

therefore the upper cut off frequency equals 60th sample.

2.2. Translation invariant features

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

$$f_2(x,y) = f_1((x-x_0),(y-y_0)) \quad (7)$$

where (x_0, y_0) is translational offset. The Fourier transforms of $f_1(x,y)$ and $f_2(x,y)$ is related by

$$F_2(u,v) = e^{-j\phi(u,v)} F_1(u,v) \quad (8)$$

where $\phi(u,v)$ is the spectra phase of $f_2(x,y)$. This phase depends on the translation, but the spectral magnitude,

$$|F_1(u,v)| = |F_2(u,v)| \quad (9)$$

is translation invariant.

Features are extracted from the filtered image by a novel tessellation (Fig. 3). This tessellation is defined as:

$$S_i = \{(u,v) \mid R_j < r < R_{j+1}, \theta_k < \theta < \theta_{k+1} \quad (10)$$
$$1 \le u \le N, 1 \le v \le N\} \quad \text{where}$$

$$r = \sqrt{(u-u_c)^2 + (v-v_c)^2} \quad (11)$$

$$\theta = \tan^{-1}\left(\frac{v-v_c}{u-u_c}\right), \theta_k \in \{0, \pi/4, \pi/2, 3\pi/4, \pi\} \quad (12)$$

N is the image length, (u_c, v_c) is the center of frequency spectrum. R_j ($R_j \in \{R_1, R_2, .., R_n\}$, $R_1 = 4$, $R_n = 60$) is determined in a manner that the total number of pixels S_i is equal to S_j for all i, j. The number of features vary for the different values of the n (n=4, 5, 6).

The spatial distribution of the variations in neighborhoods of the component 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 defined as:

$$F_i = \sqrt{\sum_{K_i} (I(u,v) - M_i)^2} \quad (13)$$

where K_i is the number of pixels in S_i and M_i is the mean of the pixel values in each sector.

The block diagram of our feature extraction algorithm is shown in Figure 3.

3. Probabilistic Neural Network (PNN)

3.1. Basics

Probabilistic Neural Network is a Bayes-Parzen classifier. The PNN was first introduced by Specht, who showed how the Bayes-Parzen classifier could

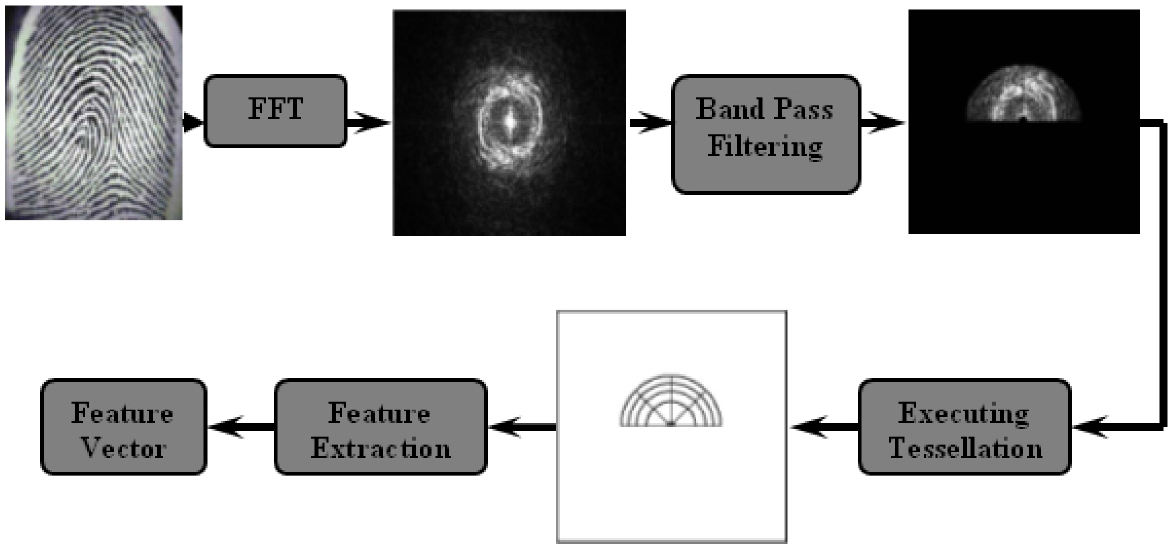

Figure 3: The block diagram of our feature extraction is shown.

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 general, the classification problem can be stated as sampling the p-component multivariate random vector $X = [x_1, x_2, .., x_p]$, where the samples are indexed by k ($k = 1,2,.., K$). The Probability that a sample belongs to the kth population (class) is h_k, the cost associated with misclassifying that sample is c_k, and that the true probability density function of all population $f_1(x), f_2(x),.., f_k(x),.., f_K(x)$ are know, Bayes theorem classifies an unknown sample into the ith population if,

$$h_i c_i f_i(x) > h_j c_j f_j(x) \quad (14)$$

for all population $j \neq i$. The density function $f_k(x)$ corresponds to the concentration of class k examples around the unknown example. Since the probability density function does usually not know 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 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

$$g(x) = \frac{1}{n\sigma^p (2\pi)^{p/2}} \sum_{i=0}^{n-1} e^{\frac{|x-x_i|^2}{2\sigma^2}} \quad (15)$$

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 can not be made.

3.2. Network operation

Consider the simple network architecture shown in Figure 4 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),

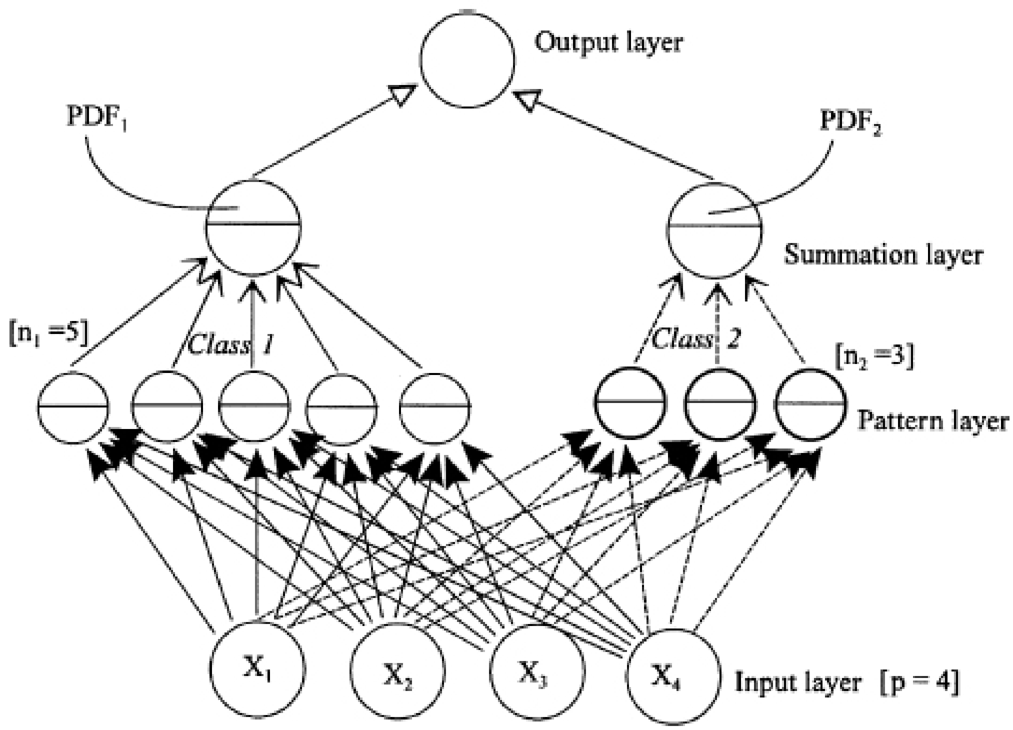

Figure 4: A simple Probabilistic neural network with four input variables, two classes, and eight training examples.

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. If the misclassification cost and prior probabilities are equal between the two classes, and the classes are mutually exclusive (i.e., no case can be classified into more than one class) and exhaustive (i.e., the training set covers all classes fairly), the activation of the summation neurons will be equal to the probability of each 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 report 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

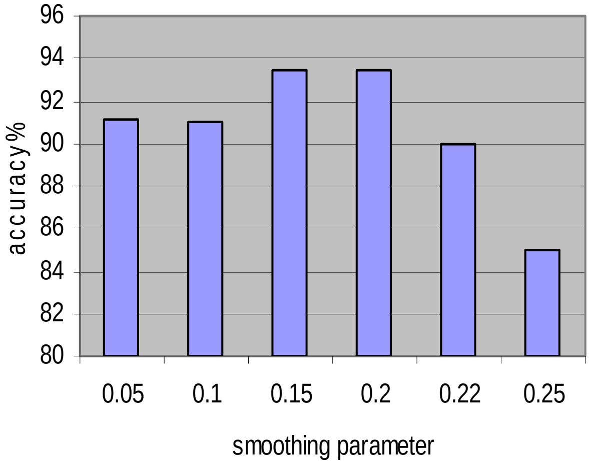

Figure 5: Performance of PNN for various values of the smoothing parameter in the Pattern layer.

corresponding to the seven classes. The smoothing parameter is determined to use try and error way (Figure 5). The result of the fingerprint classification into seven-class with different numbers of features is shown in the Table 1. We obtain an accuracy of 93.4 percent with a Rejection Ratio (RR) 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 collapsed into one class) an accuracy of 95.6 percent is achieved with 16 features. The confusion matrix for the seven-class classification problem is shown in the Table 2.

5. Conclusions

In this paper, a new scheme for the feature extraction of the fingerprint image is proposed. The extracted feature vector is not only translation invariant but also robust to noise which is reflected in the classification accuracy. We have tested our algorithm on the standard FVC2000 and FVC2002 databases and a very good performance has been achieved (93.4% for the seven-class classification task and 95.6% for the six-class classification problem). However, performance of this algorithm is improved if poor quality images are rejected and incorrect classified images are retrained to the classifier.

Table 1: Fingerprint classification with various numbers of features

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

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

True Class	Assigned Class 1	2	3	4	5	6	7	Reject
1	55	0	0	1	1	0	0	1
2	0	49	0	2	2	1	0	0
3	0	0	48	0	1	1	1	2
4	0	0	0	21	1	1	0	1
5	0	0	0	2	17	0	0	1
6	0	0	0	0	0	21	0	1
7	0	0	1	0	0	1	16	1

References

[1] J.H.Chang and K.Fan,"A new model for fingerprint classification by ridge distribution sequences ",Pattern Recoqnition,vol.35,Issue 6,pp.1209-1223,June 2002.

[2] H.Ghassemian, "An automatic fingerprint algorithm",Esteghlal Journal of Engineering ,vol.18,no.1,pp.1-11,1999.

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

[4] Sh.Shah and P.S.Sastry, "Fingerprint Classification using a Feedback-Based Line Detector ", IEEE Trans. on Systems-Man and Cybernetics ,vol.34, pp. 1-10,Februry2004.

[5] A.K.Jain and S.Prabhakar and L.Hong, "A Multichannel Approach to Fingerprint Classification", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.21, no.4, pp.348-358,1999.

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

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

[8] 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.

[9] A.Senior, "A Combination Fingerprint Classifier", IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 23, no.10, pp.1165-1175, 2001.

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