Fingerprint Type Classification Using Learning Vector Quantization

Aye Su Phyo, Khin Sandar

University of Computer studies, Mawlamyine

ayesu.p@gmail.com, drkhinsandar@gmail.com

Abstract

This paper proposes a fingerprint types classification algorithm using Learning Vector Quantization (LVQ) with FingerCode features. This algorithm assigns each fingerprint image to one of the five subclasses, according to the Henry system: Arch(A), Tented Arch(T), Left Loop(L), Right Loop(R), and Whorl Loop(W). The search for a specific fingerprint can therefore be performed only on specific subclasses containing a small portion of a large database, which will save enormous computational time. We use the feature vectors from FingerCode generation process to train with the LVQ classifiers. In our feature extraction process, the oriented components are extracted from a fingerprint image using a bank of Gabor filters, and a feature vector is computed for each oriented component. The feature vectors from the input image are classified using LVQ classifier. This algorithm has been tested the fingerprint database. For the 100 fingerprint images, the classification accuracy is 93 %, with 7 % error rate for 5-classes.

1. Introduction

Biometrics is the measurement of biological data. The term biometrics is commonly used today to refer to the authentication of a person by analyzing physical characteristics, such as fingerprints, handprints, eyes, and voice, or behavioral characteristics, such as signatures. Fingerprints are the ridges and furrow patterns on the tip of the finger. The well-known fact is that each person has a unique fingerprint.

Fingerprint analysis is probably one of the oldest and most commonly used identification technologies in biometrics. If fingerprints can be properly pre-assigned into classes, then the searching process can be entire database, which will then significantly reduce computational time.

Fingerprint classification is an important part of Automatic Fingerprint Identification System (AFIS). Classification is necessary to reduce the number of one-to-one comparisons performed when executing a fingerprint query.

In the early 20th century, Henry (1900) had classified fingerprints into three basics types: loop, arch and whorl. The loop type was then sub-classified into left loop and right loop, and the arch type was subdivided into plain arch (or arch) and tented arch. This 5-class classification system has been adopted by the National Institute of Standards and Technology (NIST) special database-4, NIST-4 (Watson and Wilson, 1992) [5]. A typical fingerprint for each class is shown in Figure 1.

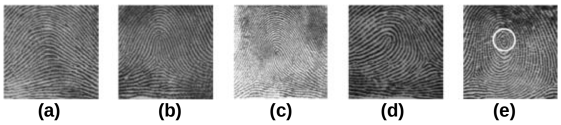

Figure1. The Henry patterns: (a) arch; (b) tented arch; (c) right loop; (d) left loop and (e) whorl

Another well-known and the most complicated classification system is the National Crime Information Center (NCIC) classification system (Federal Bureau of Investigation) which includes 19 classes. The NCIC classification system has been adopted by the NIST-9 and NIST-14 databases. The NCIC defines two arch types, Arch and Tented Arch, and two loop types, Radial and Ulnar. Whorl type is subdivided into four types: Plain Whorl, Central Pocket Whorl, Double Loop Whorl, and Accidental Whorl. Each whorl type has three subclasses depending on the ridge tracing: Inner, Outer, and Meeting. The last three types are Approximate Class, Amputation, and Scar [5]. Figure 2 shows some examples of the NCIC classification system.

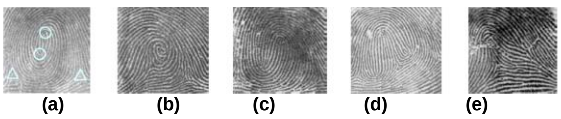

Figure 2. Some of the NCIC classes: (a) plain whorl;(b) central pocket whorl; (c) double loop whorl; (d) accidental whorl; and (e) scar;

2. Related Work

Most of the existing works aim to classify the fingerprint database on the minutiae sets and singular points. In this section, we report some of these in brief:

Masayoshi Kamijo's approach [9]: It is an ANN based approach, where a neural network for the classification of fingerprint images is constructed. It uses a two-step learning method to train the four-layered neural network which has one sub-network for each category of fingerprint images.

Cho, Kim, Bae et al's approach [2]: They described an effective fingerprint classification algorithm that uses only the information related to the core points.

Shah and Sastry's approach [10]: It can be found to be useful over low-dimensional feature vector obtained from the output of a feedback based line detector. Three types of classifiers were used here: support vector machines, nearest-neighbor classifier, and neural network classifier.

Ballan and Ayhan Sakarya's approach [8]: Here, a fast, automated, feature-based technique for classifying fingerprints is presented. The technique extracts the singular points (*deltas* and *cores*) in the fingerprints based on the directional histograms.

Karu and Jain's approach [6]: This approach first finds the ridge direction at each pixel and then extracts Global features such as singular points (*cores* and *deltas*) in the fingerprint image and performs the classification based on the number and locations of the detected singular points.

Jain, Prabhakar and Hong's Multi-channel approach [1]: The algorithm uses a representation (FingerCode) and is based on a two-stage classifier. A *k*-nearest neighbor classifier is used in its first stage and a set of neural network classifiers is used in its second stage to classify a feature vector into one of the five fingerprint classes.

This paper presents the system overview design in section 3. Feature extraction is discussed in section 4. Learning Vector Quantization algorithm is presented in section 5. Experimental result is shown in section 6. Conclusion of this paper is presented finally.

3. System Overview Design

The system overview design is shown in Figure 3. The fingerprint type classification system includes loading image, converting grayscale image, extracting the feature vectors, LVQ classification, and show the output result.

The fingerprint image takes from the fingerprint database in loading image step. In converting grayscale image, if the input fingerprint image is an RGB value (24-bit), it is converted into gray value (8-bit).Then, feature vectors are extracted using FingerCode generation process in feature extraction step. These feature vectors are used to put in the LVQ classifier. Finally, the output of this classifier is the result of the system.

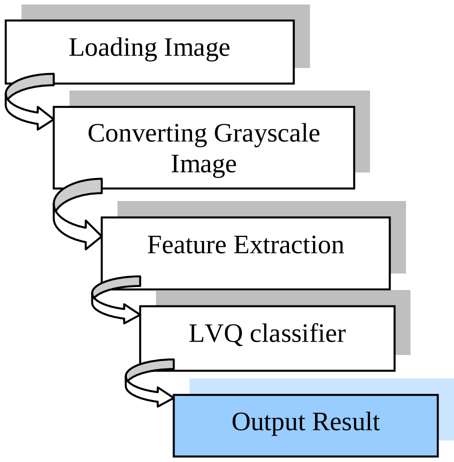

Figure 3. System Overview Design

4. Feature Extraction

The feature extraction using FingerCode uses a grayscale image as input. Therefore, we convert input image to grayscale image.

4.1 Converting Grayscale Image

If the input image is a color image, we use the process of converting grayscale image. The result of converting grayscale image is shown in Figure 4.

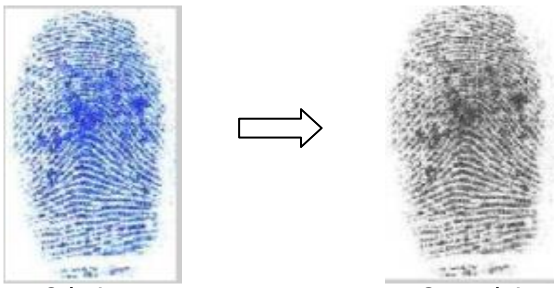

Figure 4. Converting Grayscale Image

An image is a color image, in which each pixel is specified by three values, one each for the red, green, and blue components of the pixel's color. It has 24-bit color depth: 8-bit for each R, G and B color and contain 8×8×8 bits = 256×256×256 colors = 16 million colors.

The grayscale image contains the pixels only with different gray values. In a typical gray image, a pixel's gray value is represented by 8-bits and therefore, there are 256 different gray values [3].

4.2 Feature Vector Extraction

The *FingerCode* method is a correlation based technique, where a small circular area around the core point is tessellated in an arc fashion and filtered by an oriented Gabor filter bank. For each cell in the oriented components, a value is computed that constitutes the texture feature vector.

The feature extraction algorithm can be split into four main steps as shown in Figure 5. By referring to such a figure, given the fingerprint grayscale image, the feature vectors are extracted as follow:

1. determine the reference point and crop the image centered the reference point (more details in [7]);
2. tessellate the region of interest around the reference point ;
3. filter region of interest in eight different directions using a bank of Gabor filters (more details in [7]);
4. compute the average absolute deviation from the mean of gray values in individual sectors in filtered images to define the feature vector or the FingerCode.

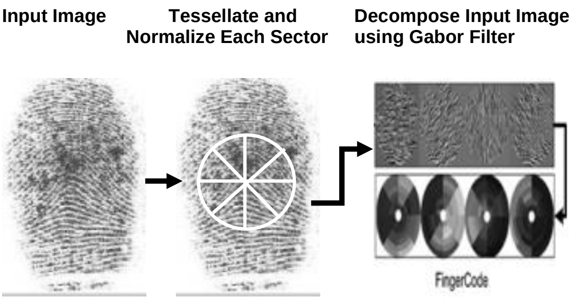

Figure 5. Flow diagram of the FingerCode feature vector

The output of the FingerCode generation process is floating points: eg., [2.4421 2.4073 2.3788 2.4264 2.4546 2.3977 2.3297 2.3807]. These values are converted into binary values. Each floating point value produces 8-bit binary values: eg. , 2.4421=0100111. Therefore, the output of the FingerCode generation process is 64-bit binary values.

4.2.1 Reference Point Location

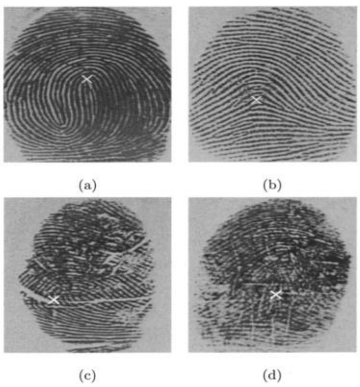

Figure 6. Example of the results of the reference point location algorithm

The reference point is the center of the region of the interest. Locating the reference point is the essential step that can influence the classification accuracy. Figure 6 shows the results of the reference point location for four different images.

4.2.2 Tessellation

Let $I(x,y)$ denote the gray level at pixel (x,y) in an $M\times N$ fingerprint image and let (x_c,y_c) denote the reference point. The region of interest in the fingerprint is defined as the collection of all the sectors S_i, where the i^{th} sector S_i is computed in terms of parameters (r,θ) as follows:

$$S_i = \left\{ (x,y) \middle|_{\theta_i \leq \theta < \theta_{i+1},\, 1 \leq x < N,\, 1 \leq y < M}^{b(T_i+1) \leq r < b(T_i+2),} \right\} \quad \textbf{(1)}$$

where,

$$T_i = i \; div \; k \quad \textbf{(2)}$$

$$\theta_i = (i \; mod \; k) \times \left(\frac{2\pi}{k}\right) \quad \textbf{(3)}$$

$$r = \sqrt{(x - x_c)^2 + (y - y_c)^2} \quad \textbf{(4)}$$

$$\theta = \tan^{-1}\left(\frac{y - y_c}{x - x_c}\right) \quad \textbf{(5)}$$

b is the width of each band, k is the number of sectors considered in each band, and i=0,…, $(B\times k-1)$, where B is the number of concentric bands considered around the reference point for feature extraction.

4.2.3 Filtering

Before filtering the fingerprint image, we normalize the gray level intensities in the region of interest in each sector separately to a constant mean and variance. Normalization is performed to remove the effects of sensor noise and gray level background due to finger pressure differences. Let $I(x,y)$ denote the gray value at pixel (x,y) , M_i and V_i, the estimated mean and variance of gray levels in sector S_i, respectively, and $N_i(x,y)$, the normalized gray-level value at pixel (x,y). For all the pixels in sector S_i, the normalized image is defined as:

$$N_i(x,y) = \begin{cases} M_0 + \sqrt{\dfrac{V_0 \times (I(x,y) - M_i)^2}{V_i}}, & if \; I(x,y) > M_0 \\ M_0 - \sqrt{\dfrac{V_0 \times (I(x,y) - M_i)^2}{V_i}}, & otherwise \end{cases} \quad \textbf{(6)}$$

where M_0 and V_0 are the desired mean and variance values, respectively.

An even symmetric Gabor filter has the following general form in the spatial domain:

$$G(x, y; f, \theta) = \exp\left\{-\frac{1}{2}\left[\frac{x'^2}{\delta_x^2} + \frac{y'^2}{\delta_y^2}\right]\right\}\cos(2\pi f x') \quad (7)$$

$$x' = x\sin\theta + y\cos\theta \quad (8)$$

$$y' = x\cos\theta - y\sin\theta \quad (9)$$

where f is the frequency of the sinusoidal plane wave along the direction from the x-axis, and δ_x and δ_y are the space constants of the Gaussian envelope along x and y axes, respectively.

4.2.4 Feature Vector

Let $F_{i\theta}(x, y)$ be the θ-direction filtered image for sector S_i. Now, the feature value, $V_{i\theta}$, is the average absolute deviation from the mean defined as :

$$V_{i\theta} = \frac{1}{n_i}\left(\sum_{n_i}\left|F_{i\theta}(x, y) - P_{i\theta}V_{i\theta}\right|\right) \quad (10)$$

where n_i is the number of pixels in S_i and $P_{i\theta}$ is the mean of pixel values of $F_{i\theta}(x, y)$ in sector S_i.

5. Learning Vector Quantization

Learning Vector Quantization (LVQ) is a supervised version of vector quantization that can be used when we have labeled input data. This is particularly useful for pattern classification problems. The first step is feature selection – a reasonably small set of features in which the essential information content of the input data is concentrated. The second step is the classification where the feature domains are assigned to individual classes.

LVQ network has no hidden layer [4]. There are one input layer and one output layer. But, there are one or more input nodes in each layer as in Figure 6.

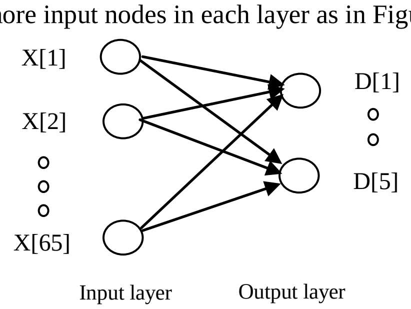

Figure 7. LVQ Neural Network

In Figure 7, the input vector is X[1] to X[65] and the desired output vector is D[1] to D[5]. The output may be 1 to 5.If the input fingerprint is “Arch” type, the output is 1. As 64-bit output of the FingerCode generation process and the 1-bit bias, this system proposes 65 input nodes in the input layer and 5 output nodes in the output layer as five classes of fingerprint images. Each output node is assigned as Arch, Tented Arch, Left Loop, Right Loop, and Whorl Loop respectively.

Actual Learning Vector Quantization Algorithm is following:

BEGIN

η is the learning rate and n is the time step.

Initialize weights to random values.

while not HALT

For each input vector

calculate the distance from the training vector

$$d(j) = \sum (w_{ij} - x_i)^2 \quad (11)$$

Find unit j with the minimum distance

update all weights vectors for units

$$w_{ij}(n+1) = w_{ij}(n) + \eta(n)\,[x_i - w_{ij}(n)] \quad (12)$$

check to see if the learning rate or radius need updating

check HALT

END

where, d(j) is the distance between input node and output node, w_{ij} is the weight between i and j and x_i is the value of input node i.

6. Experimental Result

The system is trained using 50 input images for each class. Therefore, we used the 250 fingerprint images in the training data set.

The experimental results of 5 classes with 20 samples for each are shown in Table 1.

The overall result of the Fingerprint Type Classification system is in Table 2.

The result of the system depends on the reference point. If the reference point is accurate, the result is very good. If the reference point is not accurate, the result is not so good.

The fingerprint classification system accuracy is computed using the following formula.

$$\text{Accuracy} = 100 \times \frac{x}{y} \quad (13)$$

where, x = number of accurate result images,

y = number of input images.

Table 1. The result of the system for each class

Class	Classification results					Correct rate	Error Rate
	A	*T*	*L*	*R*	*W*		
A	19	1	0	0	0	95%	5%
T	1	18	0	1	0	90%	10%
L	0	1	19	0	0	95%	5%
R	0	1	0	19	0	95%	5%
W	1	1	0	0	18	90%	10%

A= Arch, *T*= Tented Arch, *L*= Left Loop, *R*= Right Loop, *W*= Whorl Loop

Table 2. The result of 100 fingerprint images

Number of Samples	Percentage of Correct Rate	Percentage of Error Rate
100	93%	7%

6.1 Implementation of the system

The fingerprint image takes from the fingerprint database in loading image step. In converting grayscale image, if the input fingerprint image has got an RGB value (24-bit), it is converted into gray value (8-bit).Then, feature vectors are extracted using FingerCode generation process in feature extraction step. These feature vectors are used to put in the LVQ classifier. Finally, the output of this classifier is the result of the system.

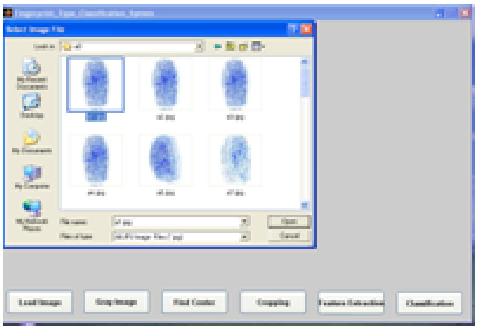

Figure 8. Loading a fingerprint image

Figure 9.Converting Grayscale Image

Figure 10.Locating the Reference Point

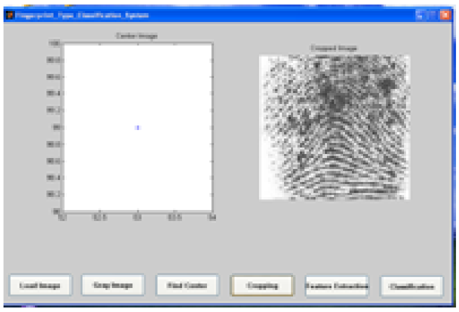

Figure 11. Cropping the image

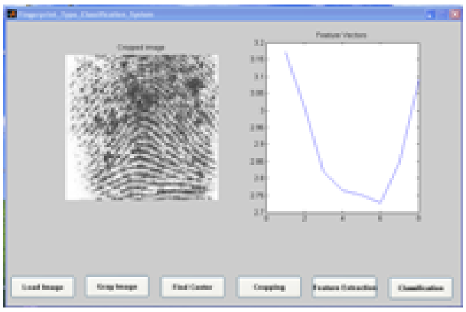

Figure 12. Feature Vector Extraction using FingerCode

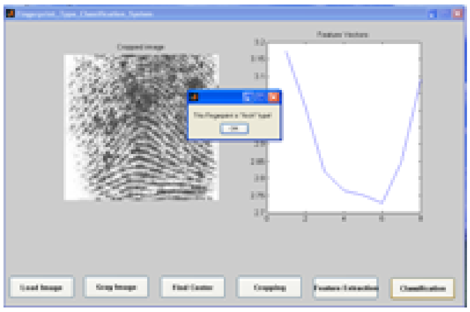

Figure 13. Fingerprint Type Classification using LVQ and the final output of the system

6. Conclusion

This paper has proposed the fingerprint type classification using Learning Vector Quantization.

The input of LVQ mainly depends on the FingerCode feature extraction process. The feature vectors also mainly depend on locating the reference point.

If the location of the reference point is accurate, the result of the system is very good. If not, this system cannot produce the good result. If the knowledge of locating the reference point is much, the system is very good.

This paper used 100 fingerprint input images to test the system. Therefore, the system cannot apply as mind as to be a good system. The system could get good performance if the system uses many good quality input fingerprint images and many enhancement methods before feature extraction step.

Therefore, this paper can be extended with very good enhancement method before feature extraction step and the knowledge of locating the reference point. The fingerprint type classification system can be performed by many fields and many methods in neural network. Other fields are image processing, DSP, fuzzy logic, and so on. Many features can be extracted that are singularities, location, angle, midpoint ridge contour, polygon, and so on. We used MATLAB 7.7.0.471 (R2008b).

References

[1] A. Jain, S. Prabhakar, and L. Hong, "A multichannel approach to fingerprint classification," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*,Volume: 21, Number: 4, Apr 1999, pp. 348-359.

[2] B. Cho, J. Kim, J. Bae, IN. Bae, and K. Yoo, "Fingerprint Image Classification by Core Analysis," *Proc. Of ICSP*, 2000.

[3] R. C. Gonzalez, R. E. Woods, *Digital Image Processing – Second Edition*, Addison-Wesley, 1992.

[4] http://highered.mcgraw-hill.com/sites/0070591121.

[5] http://en.wikipedia.org/wiki/Fingerprint.

[6] K. Karu and A.K. Jain, "Fingerprint Classification", *Pattern Recognition*, Volume: 29, Number 3, 1996, pp.389–404.

[7] L. Sha, F.Zhao, X.Tang, "Improved FingerCode for Filterbank-based fingerprint matching," *IEEE ICIP_2003*, 2003.

[8] M. Ballan, F. A. Sakarya and B. L. Evans, "A Fingerprint Classification Technique Using Directional Images," *IEEE Mathematical and Computational Applications*, 1997, pp.91-97.

[9] M. Kamijo, "Classifying Fingerprint Images using Neural Network: Deriving the Classification State", *IEEE International Conference on Neural network*, Volume: 3, 1993, pp. 1932-1937.

[10] S. Shah and P. S. Sastry, "Fingerprint Classification Using a Feedback-Based Line Detector," *IEEE Transactions on Systems, Man, and Cybernetics*, Part B: Cybernetics, Volume: 34, Number 1, February 2004, pp. 85- 94.