

Research Article

Combining Biometric Fractal Pattern and Particle Swarm Optimization-Based Classifier for Fingerprint Recognition

Chia-Hung Lin, Jian-Liung Chen, and Zwe-Lee Gaing

Department of Electrical Engineering, Kao-Yuan University, No. 1821, Jhongshan Rd., Lujhu Township, Kaohsiung County 82151, Taiwan

Correspondence should be addressed to Jian-Liung Chen, clchen@cc.kyu.edu.tw

Received 22 December 2009; Revised 11 May 2010; Accepted 12 May 2010

Academic Editor: Mohammad I. Younis

Copyright © 2010 Chia-Hung Lin et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper proposes combining the biometric fractal pattern and particle swarm optimization (PSO)-based classifier for fingerprint recognition. Fingerprints have arch, loop, whorl, and accidental morphologies, and embed singular points, resulting in the establishment of fingerprint individuality. An automatic fingerprint identification system consists of two stages: digital image processing (DIP) and pattern recognition. DIP is used to convert to binary images, refine out noise, and locate the reference point. For binary images, Katz's algorithm is employed to estimate the fractal dimension (FD) from a two-dimensional (2D) image. Biometric features are extracted as fractal patterns using different FDs. Probabilistic neural network (PNN) as a classifier performs to compare the fractal patterns among the small-scale database. A PSO algorithm is used to tune the optimal parameters and heighten the accuracy. For 30 subjects in the laboratory, the proposed classifier demonstrates greater efficiency and higher accuracy in fingerprint recognition.

1. Introduction

Physiological biometrics for uniquely recognizing humans include fingerprints, facial features, hand and palm geometry, deoxyribonucleic acid (DNA), retina, and vein authentication. These information are permanent and unique, and distinguish individuals from one another, and are also used to identify individuals in groups in home or office buildings, industrial networks, and other controlled systems [1–4]. The fingerprint-based technique is widely accepted due to its easy collection, low cost, and lack of change with age. Its techniques have become popular for biometric security systems, including the fingerprint-based security system, identity (ID) card, and smart-gate system. The identification function authenticates users from the fingerprint alone without the smartcards, usernames, or ID numbers. The template is compared to all records within the database and the closest match

decision is returned. The closest match within the allowed specific threshold is deemed to be individual and authenticated. The fingerprint identification problem can be considered a pattern recognition problem on a one-to-one basis. The goal of this study is to create an efficient and automatic technique as well as a default security level (the safest mode) for fingerprint matching.

In the literatures, many automatic techniques have been proposed to solve this problem. Fingerprints can be represented by the ridge flow and direction, morphologies (arch, loop, whorl, and accidental), and the position of the singular points [5, 6]. These features result in fingerprint individuality. Fingerprint orientation models, such as the gradient-based method and 2D Fourier expansion and Taylor expansion with sine and cosine nonlinear functions, are used to describe overall ridge topology, which can be extracted from the gradients of image gray intensity changes [2, 5, 6]. However, the first method is very sensitive to noise in poor-quality images. For an $N \times N$ image size, the second method results in $2 \times (2N + 1)^2$ model coefficients needed to estimate a series of basis functions and memory requirements. Then, significant features are identified using the multilayer artificial neural networks (MLNN), fuzzy logic and neural networks, principal component analysis (PCA), and support vector machines (SVM) [3, 7–10]. The above-mentioned methods provide promising results for fingerprint recognition.

A fingerprint image conceals several spiral-segment schemes alternating along the anatomical structure of human thumbs and fingers. To reduce a large amount of parameters requirement, Katz's algorithm [11, 12] is used to estimate the fractal dimensions (FDs) from a 2D image. An attractor with different FDs and scaling factors is used to extract features and form various fractal patterns whose can enhance the confidentiality and uniqueness. Then, a classifier performs to compare the fractal patterns among the current database and decides the identity of the user's fingerprint. In real-world applications, the adaptive classifier should be applied in a dynamical modeling system with current database changes. It also has abilities of machine learning and pattern matching, and has effect in real-time detection. The PNN classifier is identified as a nonlinear dynamic system in pattern recognition and allows add-in and delete-off training patterns. Its advantages have simple algorithm and less parameters adjustment. In this study, PSO algorithm is used to automatically estimate the best network parameters and adjust the desired targets. It is a population-based method to search an optimal solution in the high-dimensional problem space [13–18]. It can be suitable to apply in a dynamical modeling environment and can overcome the problems of over learning, learning speed, and local minimum in the large-scale training patterns. For a laboratory scale database of 30 subjects, the testing results will show computational efficiency and accurate recognition.

2. Method Description

2.1. Digital Image Preprocessing (DIP)

DIP can be divided into the image processing and reference point location. Fingerprint images are captured from subjects using an optical fingerprint reader (OFR), as shown in Figure 1. It is a complementary metal-oxide-semiconductor (CMOS) sensor (StarTex Engineering Inc., FU320U, PIV-071006) with a 500×500 pixel-sized window, 8 bites/pixel, maximum 256 gray scales, 500 dpi image resolution, and 1 : 1 image ratio of length to width, capturing a right or left thumbprint and translating the information into the computer. Many popular techniques

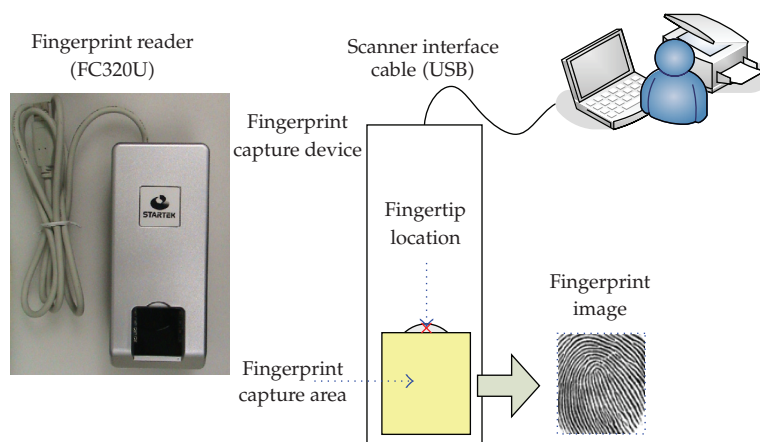


Figure 1: Optical fingerprint reader.

have been presented for image processing and reference point location (RPL), and are not described here [19–22]. By the way, RPL is used to locate a unique reference point, providing a stable sampling window for fingerprint analysis. A reference point can quickly be detected in the capture area. Central to the detected reference point, the captured template is defined as a 256×256 matrix, in which each element represents a binary value at the n th row and n th column. The key features can be further extracted using this specific window.

2.2. Fractal Dimension (FD)-Based Features

A fingerprint is a pattern of interleaved ridges (dark lines) and valleys (bright lines) that are often run in parallel, and are sometimes bifurcate and terminative, as shown in Figure 2. We can consider a ridged curve to which it is impossible to attribute a length. This curve could be locally rectifiable. Let construct a sequence Φ of graph Γ so the maximal length of the segments of Φ is less than ε . We know the Hausdorff distance $\text{dist}(\Gamma, \Phi)$ tends to zero. So graph Γ is of infinite length if the length $L(\Phi)$ tends to infinity [23]. Suppose Φ is a non-constant continuous function with a limited interval, and curve Γ is its graph, then

$$\Delta(\Gamma) = \limsup_{\varepsilon \rightarrow 0} \left(2 - \frac{\log \text{Var}_{\varepsilon}(\Phi)}{\log \varepsilon} \right) \quad (2.1)$$

$\text{Var}_{\varepsilon}(\Phi)$ is the variation of a function Φ on a segmented graph. If Φ is a constant function, the variation is zero for all ε . In particular, if Φ is differentiable, then $\text{Var}_{\varepsilon}(\Phi) \approx \varepsilon$ and the limit is one, corresponding to the dimension of a graph of finite length. We define the dimension as

$$1 \leq \Delta(\Gamma) \leq 2. \quad (2.2)$$

For a 2D binary image with $N \times N$ pixels (image size = 256×256 and resolution = 500 dpi in this study), the $f(n, m)$ has binary value of 1 at each row, $n = 1, 2, 3, \dots, N, m = 1, 2, 3, \dots, N,$

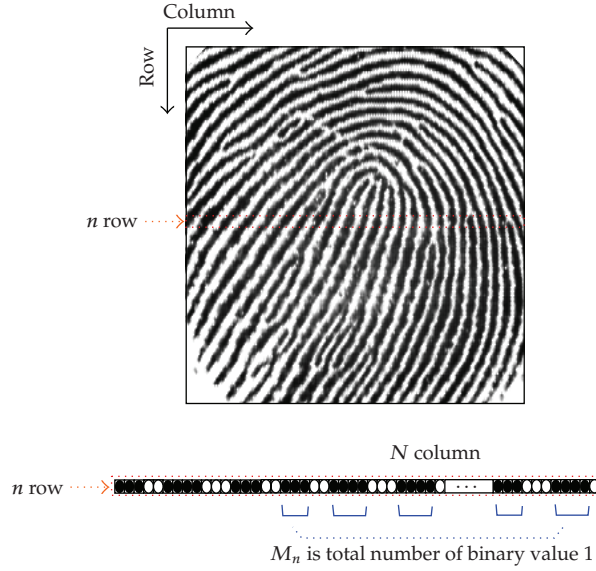


Figure 2: 2D fingerprint image (image size = 256×256 and resolution = 500 dpi in this study).

and M_n is the total number of the binary value 1 (dark portion). Katz's algorithm [11] is used to estimate the fractal dimension d_n , which can be directly estimated as

$$d_n = \frac{\log_{10}(N)}{\log_{10}(M_n)}, \quad M_n \neq 0, \quad n = 1, 2, 3, \dots, N. \quad (2.3)$$

A set of N rows, each FD is generated using (2.3), and each one can construct the fractal pattern Φ as

$$\Phi = \bigcup_{n=1}^N (n', d_n), \quad (2.4)$$

$$n' = \left[\frac{(d_n - 1)}{(n + 1) - n} \right] \times n = (d_n - 1) \times n, \quad (2.5)$$

where n' is the horizontal scaling factor (HSF) satisfying $0 < n' < N$. Equation (2.4) is utilized to construct the various fractal patterns of subjects' thumbs in the laboratory. The stored data can be reduced from $N \times N$ to $1 \times N$ for each subject. Fractal features as codes are not binary or integer values, and they can enhance confidentiality and uniqueness.

2.3. Adaptive Probabilistic Neural Network

The radial basis network (RBN) has a family network that can be designed in two different ways, including the generalized regression neural network (GRNN) and probabilistic neural

network (PNN) [24–30]. The first network is often used for function approximation. It gives a sufficient number of hidden neurons and can approximate a linear or nonlinear function to an arbitrary accuracy. Radial basis functions (RBFs) create RBN with as many radial basis neurons as there are input vectors in the training data. The second network can be used for control system, identification, and classification applications [24, 25, 29]. Its design is straightforward and does not depend on training. A PNN is guaranteed to converge to a Bayesian classifier providing it is given enough training data, and the number of radial basis neurons is proportional to the complexity of the problem. RBNs may require more neurons than standard feed-forward backpropagation networks, but they can be designed in a fraction of the time it takes to train standard feed-forward networks. They work best when many training data are available. The assignments of the network parameters are usually based on the overall statistical calculation from a precollected batch of training data. In a continuous modeling system, an adaptive PNN is used to design the fractal pattern classifier for fingerprint identification, as shown in Figure 3.

A general structure of the PNN consists of four layers, including the input layer, pattern layer, summation layer, and output layer, that have the parallelism distributed process, learning, and pattern recognition ability. Let input pattern $\Phi = [\varphi_1, \varphi_2, \varphi_3, \dots, \varphi_n, \dots, \varphi_N]$ connect to the PNN. The input layer transmits pattern Φ to the PNN, and its Euclidean distance (ED) to the recorded patterns are computed in each pattern node (also called hidden node) at pattern layer as

$$ED(k) = \sqrt{\sum_{n=1}^N (\varphi_n - w_{kn}^{IH})^2}, \quad (2.6)$$

where the weights w_{kn}^{IH} are created by the training patterns $\Phi(k) = [\varphi_1(k), \varphi_2(k), \varphi_3(k), \dots, \varphi_N(k)]$, $k = 1, 2, 3, \dots, K$. The output of pattern node H_k can be computed by [30]

$$H_k = \frac{\xi}{1 + \exp[ED(k)^2 / \sigma_k^2]}, \quad (2.7)$$

where ξ is the recognition coefficient with an interval of $(0, \infty)$; σ_k is the smoothing parameter and $\sigma_1 = \sigma_2 = \dots = \sigma_k = \dots = \sigma_K = \sigma$. The activation function H_k is applied to the $ED(k)$ between the input pattern and K training patterns. It is inversely proportional to the $ED(k)$. If $ED(k)$ approaches zero, the H_k will be a maximum value, meaning the input pattern is similar to the k th training pattern. This concept can be used for analyzing pattern relationships. Equation (2.7) can enhance the contrast, and its intensity adjustment is a technique for mapping an original intensity value to a specific range. The range of the H_k is in the interval $[0, \xi/2]$. The coefficient ξ is selected to $\gg 1$ to make H_k more distinguishable. The output O_j , $j = 1, 2, 3, \dots, K$, can be computed by

$$O_j = \frac{\sum_{j=1}^K w_{jk}^{HS} H_k}{\sum_{k=1}^K H_k} = \frac{S_k}{\sum_{k=1}^K H_k}. \quad (2.8)$$

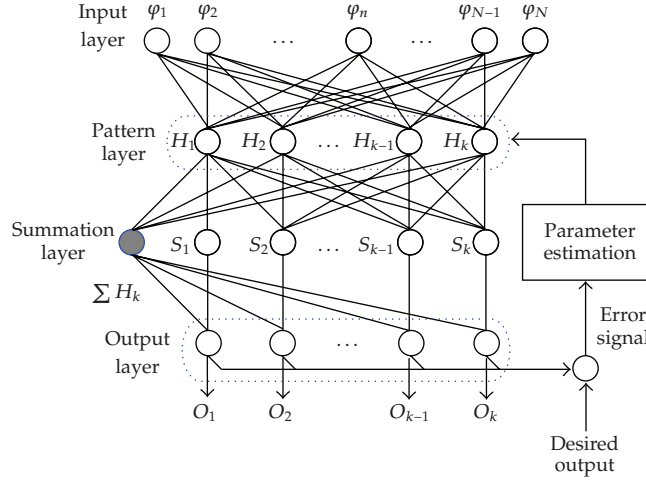


Figure 3: The structure of adaptive PNN.

Equation (2.8) expresses each predicted output component w_{jk}^{HS} is a function of the corresponding output O_j associated with each stored pattern $\Phi(k)$. The output O_j always lies between the minimum and maximum value ($0 \leq O_j \leq 1$).

For an online application, the adaptive mechanism may be suitable to apply in a dynamical modeling system, such as a smoothing parameter assignment and automatic targets adjustment. In the adaptation strategy, the choice of smoothing parameter significantly affects the network outcome and further refines the accuracy. Literatures have proposed optimum methods to automatically tune the parameter σ and desired targets, such as the traditional genetic algorithm (GA) [13, 14], gradient/steepest descent method [24, 25], and least mean square (LMS) method [30]. The optimization techniques are used to minimize the object function, which is defined as the mean squared error function (MSEF)

$$\text{MSEF} = \frac{1}{K} \sum_{k=1}^K \sum_{j=1}^K [T_j(k) - O_j(k)]^2, \quad (2.9)$$

where $T_j(k)$ is the desired target for the k th training pattern. The optimal parameter σ is intended to minimize the MSEF. However, MSEF is a nonlinear function and its partial differential equation is difficult to obtain. Traditional GA provides a promising result for optimizing a multivariate function, but it cannot guarantee convergence to a global optimal solution and is time-consuming.

The particle swarm optimization (PSO) algorithm is one of the evolutionary optimization techniques, which has been proven to be efficient in solving optimization problems, such as featuring nonlinearity, multiple optima, nondifferentiability, and high dimensionality [15–18, 28]. It guides searches using multiple particles rather than individuals and can avoid trapping at a local minimum. Each particle represents a candidate solution to the optimization problem. Particles modify their search points around a multidimensional search space until unchanged positions have been achieved. Each position is adjusted by dynamically altering the velocity of each particle, according to its flying experience and the flying experiences of the other particles in the search space. The PSO algorithm can handle

the nonlinear optimization problem with only a small program. The best σ as a position associated with the best object function is called the global best σ_{best} . The σ_{best} is determined and updated during the search. The modification of parameter σ can be represented by the concept of velocity. Velocity $\Delta\sigma_g^t$ of each agent can be represented as

$$\text{Velocity : } \Delta\sigma_g^{t+1} = \omega\Delta\sigma_g^t + c_1\text{rand}_1(\sigma_{\text{best}_g} - \sigma_g^t) + c_2\text{rand}_2(\sigma_{\text{best}} - \sigma_g^t), \quad (2.10)$$

$$\text{Position : } \sigma_g^{t+1} = \sigma_g^t + \Delta\sigma_g^{t+1}, \quad (2.11)$$

where ω is the inertia weight controlling the impact of the current velocity on the next velocity, c_1 and c_2 are the positive acceleration parameters pulling each particle toward the best positions, rand_1 and rand_2 are the uniformly random numbers between 0 and 1, σ_i^t is the current position of agent g th at iteration number t , $g = 1, 2, 3, \dots, G$, G is the population size, σ_{best_g} is the individual best.

The second part of (2.10) is the “*cognitive component*”, representing the personal thinking of each particle. It encourages the particles to move toward their own best positions found so far. The third part is the “*social component*”, representing the collaborative effect of the particles to find the global optimal solution. To efficiently convergence to the global optimal solution, PSO with time-varying acceleration coefficients (TVAC) is used to improve the performance of PSO. Parameters c_1 and c_2 can be modified as [17, 18]

$$\begin{aligned} c_1 &= (b_1 - a_1)\frac{t}{t_{\text{max}}} + a_1, \\ c_2 &= (b_2 - a_2)\frac{t}{t_{\text{max}}} + a_2, \end{aligned} \quad (2.12)$$

where a_1 , b_1 , a_2 , and b_2 are constant, t_{max} is the maximum number of allowable iterations, $t = 1, 2, 3, \dots, t_{\text{max}}$. With a large cognitive component and small social component, particles are allowed to move around the search space at the beginning stage. Through reducing the cognitive component and increasing the social component, a small cognitive component and a large social component allows the particles to converge to the global optimal at the end of search. The selected values are used in this study with changing c_1 from 2.5 to 0.5 and changing c_2 from 0.5 to 2.5. These experience values find the best ranges with numerical experiments [17].

There are two convergent conditions to terminate the PSO algorithm: (a) the objective function MSEF is less than the prespecified value; (b) the number of iteration achieves the maximum allowable number t_{max} . Thus, the optimal parameter σ can be found to minimize the MSEF.

3. PSO-Based Classifier Construction

3.1. Experimental Setup

In this study, an optical sensor is capable of digitizing the fingerprint on contact. Live-scanned images are directly acquired by sensing the tip of the fingers. A fixed fingertip location and a specific sampling window are used to capture a reliable fingerprint image. Each image

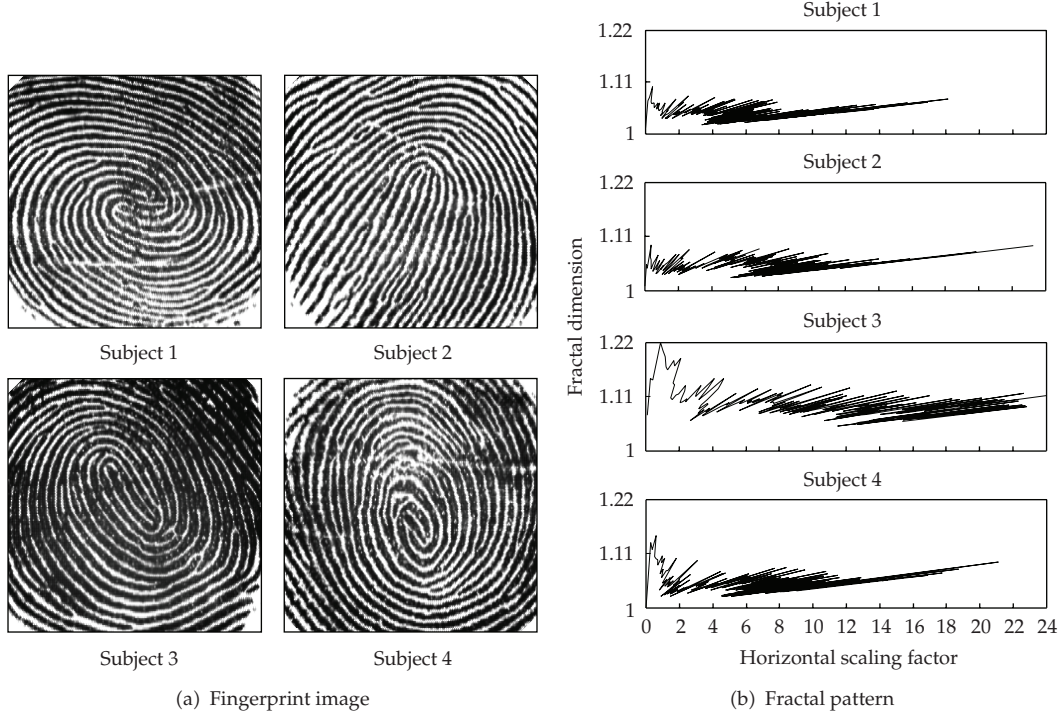


Figure 4: Original fingerprint image and their fractal patterns for four subjects.

is binarized using image adjustment with a binarization threshold associating bright pixels with binary value 0 and the dark pixels with binary value 1, translating the information into the computer. The RPL process is used to specify a $256(\text{horizontal}) \times 256(\text{vertical})$ sampling window for key features extraction. Let $f(n, m)$, $n = 1, 2, 3, \dots, 256$, $m = 1, 2, 3, \dots, 256$, the sampling window is defined as a 256×256 matrix, where each element represents a binary value at the n th row and m th column. Equation (2.3) is used to extract the fractal dimensions in the row-direction, and the number of fingerprint features can be reduced from 256×256 to 256×1 . Then (2.4) and (2.5) are used to form the various fractal patterns. For 30 subjects in the laboratory, a small-scale database was used to validate the fingerprint identification. These fingerprints can be classified into three typologies such as loop, whorl, and arch, and the major typology is loop. For example, Subject 1~Subject 4, various fractal patterns are shown in Figure 4, where horizontal scaling factors (HSFs) are chaotic orbit satisfying $0 < n' < 24$, and fractal dimensions (FDs) obey $1.00 < d_n < 1.22$. According to the various fractal patterns, we can systematically create training data. We have 30-set training patterns for the proposed classifier.

We proposed a PSO-based classifier with 256 input nodes, 30 hidden nodes, 31 summation nodes, and 30 output nodes. The overall nodes are fully connected among the four layers. In the learning stage, the classifier structure was determined based on the number of input-output training data. According to the various fractal patterns, we can systematically create training pattern $\Phi(k)$ using (2.4) and (2.5). The associated fractal patterns could be expressed as weights w_{kn}^H , $k = 1, 2, 3, \dots, 30$, $n = 1, 2, 3, \dots, 256$, between the input and hidden layer. The desired targets could also be expressed as weights w_{jk}^{HS} , $j = 1, 2, 3, \dots, 30$, between the hidden and summation layer. These are encoded as binary values with signal "1"

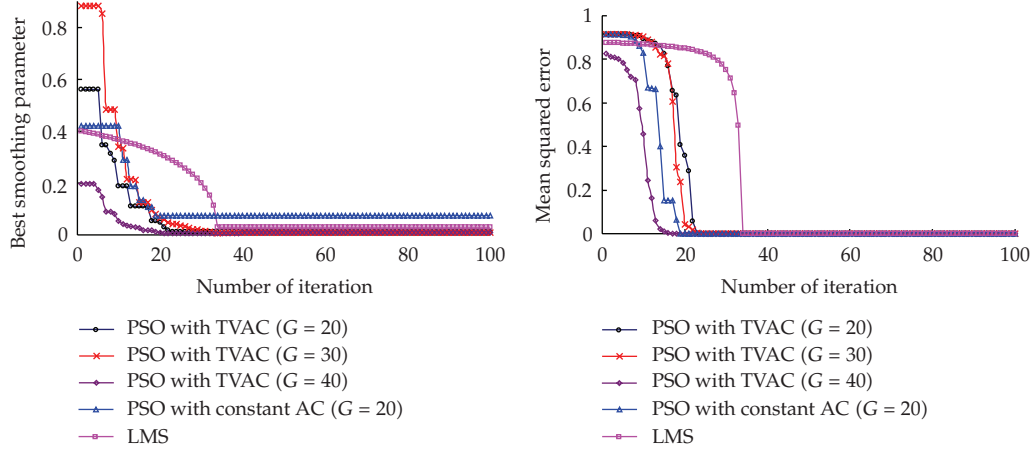


Figure 5: The best smoothing parameters and mean squared errors versus the number of iterations.

denoting the 30 identities while the rest of the weights are zero. For 30 subjects classification, each output automatically adjusts to approach its desired target $w_{jj}(w_{jj} = T_j(j) = 1)$ after the training stage. The final output $O_j(j)$ of each subject will be equal or close to value 1 using (2.8). Finally, a threshold of rejection is used to confirm the decision, can be defined as

$$O_j > \text{threshold } \theta = 0.95 \times \frac{1}{30} \sum_{j=1}^{30} O_j(j), \quad (3.1)$$

where O_j is the output of PNN in the recalling stage. That means O_j is maximum one and is up to the threshold $\theta = 0.95 \times$ average output, we have a security criterion to judge the identity of the user's fingerprint. A threshold θ is as high as possible and approaches to 1.

3.2. PSO-Based Classifier Training

The related data of the PSO-based classifier are shown in Table 1. The related parameters of the PSO algorithm with TVAC are given by population size $G = 20$ for each iteration, inertia weight $\omega = 0$, coefficients $a_1 = 2.5$, $b_1 = 0.5$ (acceleration parameter $c_1 = 2.5 \sim 0.5$), $a_2 = 0.5$, $b_2 = 2.5$ (acceleration parameter $c_2 = 0.5 \sim 2.5$) [17], and number of iterations $t_{\max} = 100$ for the optimal parameter estimation. The convergent condition is set a predefined number of iterations t_{\max} to terminate the evolution computations. The optimal parameter $\sigma_{\text{best}} = 0.0129$ minimizes the classification error. These promising results could be obtained after at least 5 successive learning stages. Figure 5 shows the best smoothing parameters and mean squared errors versus the number of iterations, respectively. It can rapidly converge for less than 22 learning iterations and takes 2.5790 seconds to classify the 30 training data. The performance of the traditional PSO algorithm is poor with constant acceleration coefficients $c_1 = 2.5$ and $c_2 = 0.5$ or $c_1 = 0.5$ and $c_2 = 2.5$. It can quickly search an optimum solution, but it is easy to trap local minimum ($\sigma_{\text{best}} = 0.0733$, CPU time = 2.3130 seconds). In PSO optimization research, PSO can find a good solution for considerable two control factors, such as the number of iterations and population sizes. However, only slight improvement

Table 1: Related data of the PSO-based classifier and traditional classifier.

Task	Method	
	Proposed PSO-based classifier	Traditional classifier
Training data	30	30
Activation function	(2.7) $\xi = 5,$ $k = 1, 2, 3, \dots, K,$ $\sigma_1 = \sigma_2 = \dots = \sigma_K = \sigma.$	Gaussian function: $H_k = \exp\left[-\frac{ED(k)^2}{2\sigma_k^2}\right]$ $k = 1, 2, 3, \dots, K,$ $\sigma_1 = \sigma_2 = \dots = \sigma_K = \sigma.$
Parameter assignment	$\omega \in [0, 1]$ $\text{rand}_1 \in [0, 1]$ $\text{rand}_2 \in [0, 1]$ $a_1 = 2.5, b_1 = 0.5$ $c_1 = 2.5 \sim 0.5$ $a_2 = 0.5, b_2 = 2.5$ $c_2 = 0.5 \sim 2.5$ $t_{\max} = 100$	$\sigma(t=0) = 0.4 \sim 1.0$ $\eta(t=0) = 1.0$ $\eta(t \rightarrow \infty) = 0.1$ $\eta = 0.1 \sim 1.0$
Parameter estimation	PSO algorithm with TVAC	Least mean square (LMS) Algorithm
Convergent condition	The number of iterations achieve the maximum allowable number $t_{\max} = 100$	MSEF is less than the prespecified value 10^{-3}
Training iteration	Figure 5	Figure 5
Optimal parameter	0.0129	0.0131~0.0302
Training time (sec)	2.5790	2.4210

Note: Time-Varying Learning Rate: $\eta(t) = \eta_0 \exp(-t/\tau)$.

of optimal parameters was obtained by increasing above two control factors ($G = 20 \sim 40$ and $t_{\max} = 50 \sim 100$). Its training stage increases the number of evolution computations to minimize the MSEF, and average CPU time also increases from 2.50 seconds to 20.00 seconds. For experimental analysis, the global optimum solution can be achieved with time-varying cognitive components and social component, whose are the sensitivities on the global search.

For comparison with learning performance, the least mean square (LMS) algorithm is also considered. However, it is difficult to derive the partial derivative of the nonlinear MSEF as (2.9). To simplify the learning procedure, the first partial derivatives of MSEF are used in this study [24–26]. The best smoothing parameters and mean squared errors versus the number of iterations are also shown in Figure 5. For the convergent condition $\text{MSEF} \leq 10^{-3}$, the traditional classifier converges to the optimal solution for less than 40 learning iteration. It takes 2.4210 seconds to classify the 30 training data. The optimal parameter $\sigma_{\text{best}} = 0.0302$ minimizes the MSEF. The choice of initial smoothing parameter $\sigma(t=0)$, learning rate, and convergent condition will affect the learning performance, which are determined by the experience formulas or trial-and-error procedure. In addition, its can rapidly approach to near optimal solution with time-varying learning rates from $\eta(t=0) = 1.0$ to $\eta(t \rightarrow \infty) = 0.1$, the so-called search-convergence rule. However, its solution is monotone decreasing and is easy to trap the local minimum. As the number of training data increases, the training process and classification efficiency become the main problem. These comparisons show the proposed PSO-based

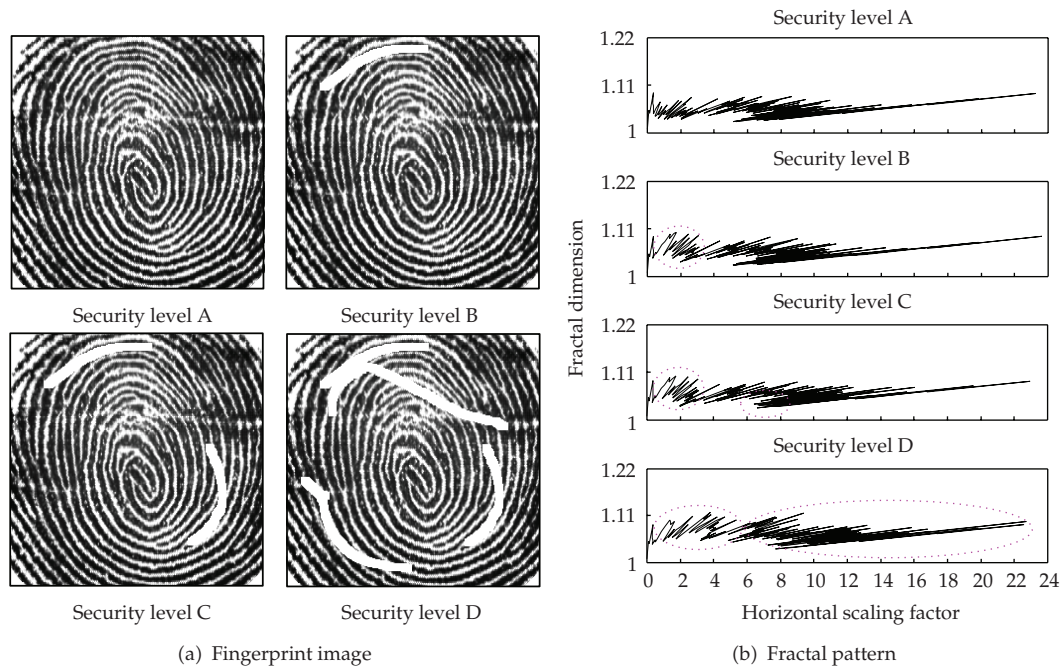


Figure 6: Fingerprint images with/without attrition lines and their fractal patterns.

classifier has better performance than other methods. Therefore, we can reduce the stress of the operator waiting for the convergence of the learning and construction procedure.

4. Experimental Results

The proposed PSO-based classifier was developed on a PC Pentium-IV 3.0GHz with 480MB RAM and Matlab software (MathWorks Inc.), which is a well-known image processing workspace to examine the proposed PSO-based classifier. The definition of fingerprint identification is to verify a live-scanned fingerprint against the previously enrolled fingerprint, checking if they came from the same finger. It requires a security criterion to judge the identity of the fingerprint. The specific threshold is set via the experimental results using (3.1). To set the “threshold value” and test the “security level”, the testing images were produced with attrition lines to evaluate the security criterion. The performances of the proposed classifier were tested as detailed below.

A laboratory-scale database of 30 subjects is used to validate the fingerprint identification system. A live-scanned fingerprint could be captured with noise, including naturally occurring dirt, bruises, and moisture on the image. This means these noises will affect the security level. Therefore, a critical threshold is important in further fingerprint recognition. To test the robustness of the proposed classifier, attrition lines were randomly generated on the scanned image. Twelve randomly selected subjects’ fingerprints involving attrition lines were used to examine the security level. Synthetic attrition lines are gradually increased, and are randomly added to the original fingerprint image, as shown in Figure 6. It can be seen that the fractal dimensions of original image are between 1.00 and 1.11, and the fractal dimensions increase as the attrition lines increase, as shown in dash-line portion. If

Table 2: Compares the results of the proposed method and other methods.

Security level		A	B	C	D	E
The number of attrition lines		0	1	2~3	4	≥ 5
Proposed PSO-based classifier	Accuracy (%)	100%	100%	100%	16.7%	0.00%
	Average output	1.0000	1.0000	0.9583	0.1667	0.0000
	Threshold value	0.95	0.95	0.91	—	—
LMS-based classifier	Accuracy (%)	100%	100%	100%	16.7%	0.00%
	Average output	0.9998	0.9920	0.9044	0.2981	0.0000
	Threshold value	0.95	0.95	0.86	—	—
Traditional PSO-based classifier	Accuracy (%)	100%	100%	75%	16.7%	0.00%
	Average output	0.9998	0.9920	0.8310	0.3187	0.0000
	Threshold value	0.95	0.95	0.79	—	—

Note: (1) Accuracy(%) = $(N_r/N_t) \times 100\%$, N_r : the number of correctly discriminated fingerprints; N_t : total number of fingerprints.

(2) Average Output = $(1/30) \sum_{j=1}^{30} O_j(j)$.

(3) Symbol—means that overall outputs of the classifier are close, and the threshold values is unable to determine.

the fingerprint is clear and average, the fractal pattern will not corrupt the previous features. To ensure the identification quality, a critical threshold of rejection is important in further fingerprint recognition. The average outputs decay as the attrition lines increase. Security levels A to C are the safest mode with less than three attrition lines, providing 100% accuracy and an average output = 0.9583. The experimental results show the proposed PSO-based classifier has high identification confidence using the threshold $\theta = 0.91$, as shown in Table 2.

To develop an automatic fingerprint identification system, a designer can set the “security level” according to the enrolled quality. The following five conditions may occur [31].

- (i) Level A. Enrollment of user’s fingerprint is successful and very clear, has stable features, and is suitable for further identification.
- (ii) Level B. Enrollment of user’s fingerprint is successful and clear, has stable features, and is suitable for further identification.
- (iii) Level C. Enrollment of user’s fingerprint is average with enough stable features for further identification.
- (iv) Level D. Fingerprint may not be very clear, or may not have very good features. At this level, some subjects may not be able to be identified.
- (v) Level E. Fingerprint is not clear. The false rejection for this level is higher than other levels.

Fingerprint enrollment is the key stage in the biometric security system. If a user does not place her/his finger on the OFR (FU320U, PIV-071006) correctly, the RPL program will prompt user with a warning message (at least three times and check the enrolled quality). The enrollment will be finished when a fingerprint enough stable features for further identification. The proposed PSO-based classifier could tolerate minor noise. The default security level could be set to “level C” for a real world application. It provides reliable and accurate results, and the matching compares a live-scanned fingerprint with the specific threshold enhances the authentication.

5. Conclusion

A PSO-based classifier using fractal patterns and an adaptive PNN to recognize fingerprints has been proposed. Live-scanned images are captured using an optical fingerprint reader. The digital image processing is used to enhance the image, convert to a binary image, and locate the reference point. Katz's algorithm is used to estimate the fractal dimensions from a two-dimensional image. The fractal patterns are not binary or integer values, whose can avoid becoming duplicated and decoded templates. They can retain the main particular features with FDs and HSFs. The proposed PSO-based classifier can also tolerate minor noise. The default security level C could be set for automatic fingerprint identification with average and enough stable features. For a small-scale database, the examination results show efficiency and higher accuracy. The proposed method has been verified in a personal computer-based biometric security system. The framework could implement in a portable device or hardware device, such as digital signal processor (DSP) and field-programmable gate array (FPGA). It can be further integrated into the identity access management system, industrial security network, and smartgate system.

References

- [1] A. Jain and S. Pankanti, "A tough of money," *IEEE Spectrum*, pp. 14–19, 2006.
- [2] Y. Wang, J. Hu, and D. Phillips, "A fingerprint orientation model based on 2D fourier expansion (FOMFE) and its application to singular-point detection and fingerprint indexing," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 4, pp. 573–585, 2007.
- [3] J. P. S. Medeiros, A. C. da Cunha, A. M. Brito Jr., and P. S. M. Pires, "Automating security tests for industrial automation devices using neural networks," in *Proceedings of IEEE Symposium on Emerging Technologies and Factory Automation (ETFA '07)*, pp. 772–775, September 2007.
- [4] D. Bouchaffra and A. Amira, "Structural hidden Markov models for biometrics: fusion of face and fingerprint," *Pattern Recognition*, vol. 41, no. 3, pp. 852–867, 2008.
- [5] A. M. Bazen and S. H. Gerez, "Systematic methods for the computation of the directional fields and singular points of fingerprints," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 905–919, 2002.
- [6] W.-Y. Yau, J. Li, and H. Wang, "Nonlinear phase portrait modeling of fingerprint orientation," in *Proceedings of the 8th International Conference on Control, Automation, Robotics and Vision (ICARCV '04)*, vol. 2, pp. 1262–1267, December 2004.
- [7] M. Kamijo, "Classifying fingerprint images using neural network: deriving the classification state," in *Proceedings of IEEE International Conference on Neural Networks*, vol. 3, pp. 1932–1937, March-April 1993.
- [8] K. R. Crouse and L. O. Chua, "Methods for image processing and pattern formation in cellular neural networks: a tutorial," *IEEE Transactions on Circuits and Systems*, vol. 42, no. 10, pp. 583–601, 1995.
- [9] K.-A. Toh and W.-Y. Yau, "Combination of hyperbolic functions for multimodal biometrics data fusion," *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 34, no. 2, pp. 1196–1209, 2004.
- [10] J.-H. Hong, J.-K. Min, U.-K. Cho, and S.-B. Cho, "Fingerprint classification using one-vs-all support vector machines dynamically ordered with naïve Bayes classifiers," *Pattern Recognition*, vol. 41, no. 2, pp. 662–671, 2008.
- [11] M. J. Katz, "Fractals and the analysis of waveforms," *Computers in Biology and Medicine*, vol. 18, no. 3, pp. 145–156, 1988.
- [12] R. Esteller, G. Vachtsevanos, J. Echauz, and B. Litt, "A Comparison of waveform fractal dimension algorithms," *IEEE Transactions on Circuits and Systems I*, vol. 48, no. 2, pp. 177–183, 2001.
- [13] C. T. Cheng, C. P. Ou, and K. W. Chau, "Combining a fuzzy optimal model with a genetic algorithm to solve multi-objective rainfall-runoff model calibration," *Journal of Hydrology*, vol. 268, no. 1–4, pp. 72–86, 2002.
- [14] N. Muttill and K.-W. Chau, "Neural network and genetic programming for modelling coastal algal blooms," *International Journal of Environment and Pollution*, vol. 28, no. 3–4, pp. 223–238, 2006.

- [15] S. H. Ling, H. H. C. Iu, K. Y. Chan, H. K. Lam, B. C. W. Yeung, and F. H. Leung, "Hybrid particle swarm optimization with wavelet mutation and its industrial applications," *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 38, no. 3, pp. 743–763, 2008.
- [16] J. Zhang and K.-W. Chau, "Multilayer ensemble pruning via novel multi-sub-swarm particle swarm optimization," *Journal of Universal Computer Science*, vol. 15, no. 4, pp. 840–858, 2009.
- [17] A. Ratnaweera, S. K. Halgamuge, and H. C. Watson, "Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 240–255, 2004.
- [18] P. N. Suganthan, "Particle swarm optimizer with neighborhood operator," in *Proceedings of IEEE International Conference on Evolutionary Computation*, vol. 3, pp. 1958–1962, 1999.
- [19] D. Maltoni, D. Maio, A. K. Jain, and S. Prabhkar, *Handbook of Fingerprint Recognition*, Springer, New York, NY, USA, 2003.
- [20] L. Hong, Y. Wan, and A. Jain, "Fingerprint image enhancement: algorithm and performance evaluation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 8, pp. 777–789, 1998.
- [21] P. Porwik and L. Wieclaw, "A new approach to reference point location in fingerprint recognition," *IEICE Electronics Express*, vol. 1, no. 18, pp. 575–581, 2004.
- [22] H. Guo, "A Hidden Markov Model fingerprint matching approach," in *Proceedings of the International Conference on Machine Learning and Cybernetics (ICMLC '05)*, pp. 5055–5059, August 2005.
- [23] M. Barnsley, *Fractals Everywhere*, Academic Press, Boston, Mass, USA, 1988.
- [24] C.-H. Lin and C.-H. Wang, "Adaptive wavelet networks for power-quality detection and discrimination in a power system," *IEEE Transactions on Power Delivery*, vol. 21, no. 3, pp. 1106–1113, 2006.
- [25] C.-H. Lin, Y.-C. Du, and T. Chen, "Adaptive wavelet network for multiple cardiac arrhythmias recognition," *Expert Systems with Applications*, vol. 34, no. 4, pp. 2601–2611, 2008.
- [26] T. L. Seng, M. Khalid, and R. Yusof, "Adaptive general regression neural network for modelling of dynamic plants," in *Proceedings of IEEE International Symposium on Intelligent Control*, pp. 217–222, Vancouver, Canada, October 2002.
- [27] Z.-K. Huang and K.-W. Chau, "A new image thresholding method based on Gaussian mixture model," *Applied Mathematics and Computation*, vol. 205, no. 2, pp. 899–907, 2008.
- [28] Z. H. Cui, X. J. Cai, J. C. Zeng, and G. J. Sun, "Particle swarm optimization with FUSS and RWS for high dimensional functions," *Applied Mathematics and Computation*, vol. 205, no. 1, pp. 98–108, 2008.
- [29] S. Chen, X. Hong, B. L. Luk, and C. J. Harris, "Non-linear system identification using particle swarm optimization tuned radial basis function models," *International Journal of Bio-Inspired Computation*, vol. 1, no. 4, pp. 246–258, 2009.
- [30] S. Chen, E. S. Chng, and K. Alkadhimi, "Regularized orthogonal least squares algorithm for constructing radial basis function networks," *International Journal of Control*, vol. 64, no. 5, pp. 829–837, 1996.
- [31] StarTek FC320 demo Program User's Guide, StarTek Engineering, Incorporated.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

