Neural Comput & Applic (2013) 22:435–445
DOI 10.1007/s00521-011-0806-0

Fingerprint matching based on extreme learning machine

Jucheng Yang · Shanjuan Xie · Sook Yoon · Dongsun Park · Zhijun Fang · Shouyuan Yang

Received: 9 August 2011 / Accepted: 29 December 2011 / Published online: 14 January 2012

Abstract Considering fingerprint matching as a classification problem, the extreme learning machine (ELM) is a powerful classifier for assigning inputs to their corresponding classes, which offers better generalization performance, much faster learning speed, and minimal human intervention, and is therefore able to overcome the disadvantages of other gradient-based, standard optimization-based, and least squares-based learning techniques, such as high computational complexity, difficult parameter tuning, and so on. This paper proposes a novel fingerprint recognition system by first applying the ELM and Regularized ELM (R-ELM) to fingerprint matching to overcome the demerits of traditional learning methods. The proposed method includes the following steps: effective preprocessing, extraction of invariant moment features, and PCA for feature selection. Finally, ELM and R-ELM are used for fingerprint matching. Experimental results show that the proposed methods have a higher matching accuracy and are less time-consuming; thus, they are suitable for real-time processing. Other comparative studies involving traditional methods also show that the proposed methods with ELM and R-ELM outperform the traditional ones.

Keywords Extreme learning machine · Fingerprint matching · Invariant moments · Regularized

J. Yang (✉) · Z. Fang · S. Yang
School of Information Technology,
Jiangxi University of Finance and Economics,
Nanchang, China
e-mail: yangjucheng@hotmail.com; jcyang@jxufe.edu.cn

Z. Fang
e-mail: zhijunfang@jxufe.edu.cn

S. Yang
e-mail: syyang@jxufe.edu.cn

S. Xie · D. Park (✉)
School of Electronics and Information Engineering,
Chonbuk National University, Jeonbuk, South Korea
e-mail: dspark@jbnu.ac.kr

S. Xie
e-mail: xieshanjuan@jbnu.ac.kr

S. Yoon
Department of Multimedia Engineering,
Mokpo National University, Jeonnam, South Korea
e-mail: syoon@mokpo.ac.kr

1 Introduction

Biometrics, described as the science of recognizing an individual based on his or her physical or behavioral traits, is beginning to gain acceptance as a legitimate method for determining an individual's identity. Among all the biometric indicators, fingerprints have one of the highest levels of reliability and have been extensively used by forensic experts in criminal investigations [1]. Fingerprint recognition has emerged as one of the most reliable means of biometric authentication because of its universality, distinctiveness, permanence, and accuracy.

Usually, traditional fingerprint recognition approaches are divided into two main categories: minutiae-based [2, 3] and image-based [4–10]. The minutiae-based methods use feature vectors extracted from fingerprints and stored as sets of points in a multidimensional plane. The feature vectors may contain features of minutia points such as their positions, orientations, and types. It is essentially required to find the best alignment between the features of minutiae in templates and those in the fingerprints to be verified. The disadvantages of minutiae-based methods are that they may not use the rich discriminatory information available in the

fingerprints and may have high computational complexity [1]. On the other hand, image-based methods use different types of features obtained from fingerprint ridge patterns, such as local orientations and frequencies, ridge shapes, and texture information. These features can be extracted more reliably than the features used for the minutiae-based methods.

In the last two decades, learning methods have been popularly used for fingerprint recognition systems, especially for fingerprint image preprocessing [11, 12], feature extraction [13], classification [14], and matching [8, 9, 15–18]. For preprocessing, Ji et al. [11] presented a coarse-to-fine binary image-thinning algorithm with a template-based pulse-coupled neural-network model and obtained good performance for image thinning. Kang and Zhang [12] proposed an approach for fingerprint segmentation based on a cellular neural network (CNN). For feature extraction and classification, Yang et al. [13] proposed to use a fuzzy neural network for direct minutiae feature extraction. Wilson et al. [14] proposed to use neural networks for fingerprint classification to boost the classification rate. For fingerprint matching, Yang et al. [8] proposed to apply a learning vector quantization neural network for fingerprint matching. Yang and Park [9] proposed a fingerprint verification method based on invariant moment features and nonlinear backpropagation neural networks (BPNN). Yang et al. [15] proposed to use assembling invariant moments and a support vector machine (SVM) for fingerprint matching to improve performance. Jia et al. [16] proposed to use weighted minutiae features with SVM for fingerprint matching. In addition, other neural networks, such as CNN [17] and adaptive neuro-fuzzy inference systems (ANFIS) networks [18], have been proposed for fingerprint matching. However, because machine learning methods require iterative learning and tuning of the parameters, they suffer from the demerits of high time consumption and difficult parameter tuning.

The learning approaches for feedforward networks are usually classified into three approaches [19]: gradient-based (e.g., the backpropagation method), standard optimization method–based (e.g., single-hidden-layer SVM feedforward networks), and least squares-based (e.g., radial basis function (RBF) networks). The most popular gradient-based learning algorithms update each learnable parameter of the networks according to the error derivatives with respect to all parameters, and the gradients required in feedforward neural networks can be computed efficiently by propagation from the output to the input [20]. However, these learning algorithms have several demerits: (1) the learning rate is not easy to determine. If it is too slow, the algorithm converges very slowly; if too fast, the algorithm becomes unstable and diverges; (2) the presence of local minima impacts the performance of the algorithm. It is undesirable to have the learning algorithm stop at a local minimum which is located far away from the global minimum; (3) the network may become overtrained, in which case generalization performance will worsen; and (4) gradient-based learning is very time-consuming in most applications. Similarly to the gradient methods, learning algorithms that use standard optimization networks (e.g., SVM) and least squares–based methods (e.g., RBF networks) also have the demerits of high computational complexity and difficult parameter tuning.

Recently, Huang et al. [21, 22] proposed a novel machine learning algorithm called the extreme learning machine (ELM) that has significantly faster learning speed and requires less human intervention than other learning methods. It has been proven that the hidden nodes of the "generalized" single-hidden-layer feedforward networks (SLFNs) can be randomly generated and that the universal approximation capability of such SLFNs can be guaranteed. The ELM can determine analytically all the parameters of SLFNs instead of adjusting parameters iteratively. Thus, it can overcome the demerits of the gradient-based method and of most other learning methods. Compared to the most effective SVM-based method, the latest research [22] also shows that the ELM tends to achieve better generalization performance, less sensitivity to user-specified parameters, and easier implementation than a traditional SVM. Besides, it has been noted that the ELM has extremely fast learning speed and produces good performance in many cases [23, 24]. However, the random input weights may make the SLFN trained with the ELM sensitive to the disturbances. For instance, as the input weights and the hidden layer biases are randomly assigned in an SLFN, the changes of the hidden layer output matrix sometimes are very large because of the effects of the input disturbances, which also result in the big changes of the output weight matrix of the SLFN. Recently, many modified ELMs have been developed, such as Regularized Extreme Learning Machine (R-ELM)[25] and Finite Impulse Response Extreme Learning Machine (FIR-ELM) [26], and so on, for improving the robustness of the SLFN with respect to the input disturbances and the randomly selected input weights.

So, this paper proposes a novel fingerprint matching algorithm based on the ELM and its improved version R-ELM. The ELM is a powerful classifier for assigning inputs to their corresponding classes. An ELM is used to verify the identification of test fingerprint inputs by matching the feature vectors of test inputs with those of template images. The ELM is trained to determine whether an input matches a template in a database. Using ELM as a verifier may provide a flexible scheme with high matching accuracy and fast speed. The R-ELM is used to further improve the effectiveness of the fingerprint matching. For a

comparison study on the matching accuracy of recognition schemes, certain traditional learning methods such as BPNN and SVM are compared with the proposed method. The contribution of this paper is to use the ELM and R-ELM algorithm first for fingerprint matching combined with some effective preprocessing as well as feature extraction and feature selection techniques. Experimental results show that the proposed method with ELM and R-ELM achieves better accuracy and speed than traditional learning methods; thus, they are suitable for real-time processing.

The paper is organized as follows. In Sect. 2, the theory of ELM and R-ELM is briefly reviewed, and the moment features used in this paper are introduced. In Sect. 3, the proposed method with its three main steps of preprocessing, feature extraction, and ELM- and R-ELM-based matching is explained in detail. Section 4 gives experimental results. Finally, concluding remarks and directions for future work are stated in Sect. 5.

2 Theory

2.1 Brief overview of the ELM

Based on the structure of SLFN, an ELM is a unified SLFN with randomly generated hidden nodes independent of the training data [21, 22, 27]. For N arbitrary distinct samples $(\mathbf{x}_i, \mathbf{t}_i)$, where $\mathbf{x}_i = [x_{i1}, x_{i2}, \ldots, x_{in}]^T \in R^n$ and $\mathbf{t}_i = [t_{i1}, t_{i2}, \ldots, t_{im}]^T \in R^m$ (n is the number of dimensions of input $\mathbf{x}$ and m is the number of classes of data), a given set of training samples $\{(\mathbf{x}_i, \mathbf{t}_i)\}_{i=1}^N \subset R^n \times R^m$, the output of a SLFN with L hidden nodes, can be represented by:

$$f_L(\mathbf{x}_j) = \sum_{i=1}^{L} \beta_i K(\mathbf{a}_i, b_i, \mathbf{x}_j) = \mathbf{t}_j \quad j = 1, \ldots, N \tag{1}$$

where $\mathbf{a}_i$ and b_i are the parameters of the ith hidden node, which could be randomly generated, $K(\mathbf{a}_i, b_i, \mathbf{x}_j)$ is the output of the ith hidden node with respect to the input $\mathbf{x}_j$, and $\boldsymbol{\beta}_i$ is the weight connecting the ith hidden node to the output nodes.

Equation 1 can be written compactly as:

$$\mathbf{H}\beta = \mathbf{T} \tag{2}$$

where

$$H(\mathbf{a}_1, \ldots, \mathbf{a}_L, b_1, \ldots b_L, \mathbf{x}_1, \ldots, \mathbf{x}_N) = \begin{bmatrix} K(\mathbf{a}_1, b_1, \mathbf{x}_1) \ldots & K(\mathbf{a}_L, b_L, \mathbf{x}_1) \\ \vdots & \vdots \\ K(\mathbf{a}_1, b_1, \mathbf{x}_N) \ldots & K(\mathbf{a}_L, b_L, \mathbf{x}_N) \end{bmatrix}_{N \times L} \tag{3}$$

$$\boldsymbol{\beta} = \begin{bmatrix} \boldsymbol{\beta}_1^T \\ \vdots \\ \boldsymbol{\beta}_L^T \end{bmatrix}_{L \times m} \quad \text{and} \quad \mathbf{T} = \begin{bmatrix} \mathbf{t}_1^T \\ \vdots \\ \mathbf{t}_L^T \end{bmatrix}_{N \times m} \tag{4}$$

$\boldsymbol{\beta}^T$ is the transpose of a matrix or vector $\boldsymbol{\beta}$. $\mathbf{H}$ is called the hidden layer output matrix of the network [21, 22]; the ith column of $\mathbf{H}$ is the ith hidden node's output vector with respect to the inputs $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$, and the jth row of $\mathbf{H}$ is the output vector of the hidden layer with respect to the input $\mathbf{x}_j$. It has been proved in theory [22, 27] that SLFNs with random hidden nodes have the universal approximation capability and that the hidden nodes can be randomly generated independently of the training data.

After the hidden nodes have been randomly generated and the training data are given, the hidden-layer output matrix $\mathbf{H}$ is known and need not be tuned. Therefore, training a SLFN simply amounts to obtaining the solution of a linear system (2) of output weights $\boldsymbol{\beta}$.

According to Bartlett's theory [28] for feedforward neural networks, to obtain better generalization performance, the ELM tries to achieve the smallest training error:

$$\begin{aligned} &\text{Minimize} \ \ ||\varepsilon|| \\ &\text{subject to} \ \ \varepsilon = \mathbf{H}\beta - \mathbf{T} \end{aligned} \tag{5}$$

Under the constraint of Eq. 5, a simple representation of the solution of system (2) is given explicitly by Huang et al. [22] as:

$$\boldsymbol{\beta} = \mathbf{H}^{\dagger}\mathbf{T} \tag{6}$$

where $\mathbf{H}^{\dagger}$ is the Moore–Penrose generalized inverse of the hidden-layer output matrix $\mathbf{H}$.

If the N training data points are distinct, $\mathbf{H}$ is of full column rank with probability one when $L \leq N$. In real applications, the number of hidden nodes is always less than the number of training data points, $L < N$. Therefore,

$$\mathbf{H}^{\dagger} = (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T \tag{7}$$

Huang et al. [22, 27] have proved Eq. 8 for SLFNs with a wide type of random computational hidden nodes. Additive hidden nodes are often used in applications. For example, an additive hidden node with the activation function $k(\mathbf{x}) : R \to R$ (e.g., sigmoid, threshold, sin/cos, etc.), $K(\mathbf{a}_i, b_i, \mathbf{x})$ is given by:

$$K(\mathbf{a}_i, b_i, \mathbf{x}) = K(\mathbf{a}_i \cdot \mathbf{x} + b_i) \tag{8}$$

where $\mathbf{a}_i$ is the weight vector connecting the input layer to the ith hidden node and b_i is the bias of the ith hidden node. $\mathbf{a}_i \cdot \mathbf{x}$ denotes the inner product of vectors $\mathbf{a}_i$ and $\mathbf{x}$ in R^n. The three-step ELM simple learning algorithm can be summarized as follows:

Algorithm I: learning of ELM

Given a training set $\{(\mathbf{x}_i,\mathbf{t}_i)\}_{i=1}^{N} \subset R^n \times R^m$, the hidden-node output function $K(\mathbf{a}_i,b_i,\mathbf{x})$, and the hidden-node number L.

1. Assign hidden-node parameters randomly $(\mathbf{a}_i,b_i)$, i =1, . . . , L .
2. Calculate the hidden-layer output matrix **H**.
3. Calculate the output weight vector $\boldsymbol{\beta} = \mathbf{H}^{\dagger}\mathbf{T}$.

2.2 Brief overview of the R-ELM

According to statistical learning theory, the real prediction risk of learning is consisted of empirical risk and structural risk [25]. A model with good generalization ability should have the best trade-off between the two risks. Therefore, the real risk can be represented by the weighted sum of these two kinds of risk. The proportion of them can be regularized by introducing a weight factor γ for empirical risk. The empirical risk is represented by sum error square, i.e., $||\varepsilon||^2$, and the structural risk is represented with $||\boldsymbol{\beta}||^2$, which is derived from maximizing the distance of the Margin separating classes [25]. Thus, the mathematic model of Regularized ELM algorithm can be described as:

$$\begin{aligned} &\text{Minimize} \quad \left\{\frac{\gamma}{2}||\varepsilon||^2 + \frac{1}{2}||\boldsymbol{\beta}||^2\right\} \\ &\text{subject to} \quad \varepsilon = \mathbf{H}\beta - \mathbf{T} \end{aligned} \tag{9}$$

where $\varepsilon = [\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_N]$ are the errors of N samples. γ is a constant parameter used to adjust the balance of the structural risk and the empirical risk. By regulating γ, we can adjust the proportion of empirical risk and structural risk. The Lagrangian multiplier for expression (9) can be written as follows [25, 26]:

$$\begin{aligned} L(\boldsymbol{\beta},\varepsilon,\alpha) &= \frac{\gamma}{2}||\mathbf{H}\beta - \mathbf{T}||^2 + \frac{1}{2}||\boldsymbol{\beta}||^2 \\ &\quad - \sum_{j=1}^{N}\alpha_j\left(\sum_{i=1}^{\widetilde{\mathrm{N}}}\boldsymbol{\beta}_i K(\mathbf{a}_i x_j + b_i) - t_j - \varepsilon_j\right) \\ &= \frac{\gamma}{2}||\mathbf{H}\beta - \mathbf{T}||^2 + \frac{1}{2}||\boldsymbol{\beta}||^2 - \alpha(\mathbf{H}\beta - \mathbf{T} - \varepsilon) \end{aligned} \tag{10}$$

where $\alpha_j \in R(j = 1, 2, \ldots, N)$ is the Lagrangian multiplier. Setting the gradients of this Lagrangian with respect to $(\boldsymbol{\beta}, \varepsilon, \alpha)$ equal to zero gives the following optimality conditions:

$$\begin{cases} \frac{\partial L}{\partial \beta} \rightarrow \beta^T = \alpha\mathbf{H} \\ \frac{\partial L}{\partial \varepsilon} \rightarrow \gamma\varepsilon^T + \alpha = 0 \\ \frac{\partial L}{\partial \alpha} \rightarrow \mathbf{H}\beta - \mathbf{T} - \varepsilon = 0 \end{cases} \tag{11}$$

Then, the solution of expression (11) is:

$$\boldsymbol{\beta} = \left(\frac{\mathrm{I}}{\gamma} + \mathbf{H}^T\mathbf{H}\right)^{-1}\mathbf{H}^T\mathbf{T} \tag{12}$$

The expression (12) only involves the inversion of a matrix of order $\widetilde{\mathrm{N}} \times \widetilde{\mathrm{N}}$, where $\widetilde{\mathrm{N}}$ is the number of hidden nodes with $\widetilde{\mathrm{N}} \ll \mathrm{N}$. Therefore, the speed of calculating $\boldsymbol{\beta}$ is very fast. Thus, the R-ELM learning algorithm can be summarized as follows:

Algorithm II: learning of R-ELM

Given a training set $\{(\mathbf{x}_i,\mathbf{t}_i)\}_{i=1}^{N} \subset R^n \times R^m$, the hidden-node output function $K(\mathbf{a}_i,b_i,\mathbf{x})$, and the hidden-node number L.

1. Assign hidden-node parameters randomly $(\mathbf{a}_i,b_i)$, i =1, . . . , L .
2. Calculate the hidden-layer output matrix **H**.
3. Calculate the output weight vector $\boldsymbol{\beta} = (\frac{\mathrm{I}}{\gamma} + \mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T\mathbf{T}$.

2.3 Geometric moment analysis

Geometric moment (Hu moment) features can provide the properties of invariance to scale, position, and rotation [29]. In this research, geometric moment analysis is used to extract invariant features from tessellated cells in a region of interest (ROI). This section gives a brief description of geometric moment analysis.

For a 2-D continuous function $f(x, y)$, the moment of order $(p + q)$ is defined as:

$$m_{pq} = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} x^p y^q f(x,y)\,\mathrm{d}x\,\mathrm{d}y \quad \text{for} \quad p,q = 0,1,2\ldots \tag{13}$$

A uniqueness theorem states that if $f(x, y)$ is piecewise continuous and has nonzero values only in a finite part of the xy-plane, then moments of all orders exist, and the moment sequence (m_{pq}) is uniquely determined by $f(x, y)$. Conversely, (m_{pq}) is uniquely determined by $f(x, y)$.

The central moments are defined as:

$$\mu_{pq} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} (x-\bar{x})^p (y-\bar{y})^q f(x,y)\mathrm{d}x\mathrm{d}y \tag{14}$$

where $\bar{x} = m_{10}/m_{00}$ and $\bar{y} = m_{01}/m_{00}$.

If $f(x,y)$ is a digital image, then Eq. 14 becomes:

$$\mu_{pq} = \sum_x\sum_y (x-\bar{x})^p (y-\bar{y})^q f(x,y) \tag{15}$$

and the normalized central moments, denoted by η_{pq}, can be defined as:

$$\eta_{pq} = \mu_{pq}/\mu_{00}^{\gamma}, \quad \text{where} \quad \gamma = (p+q)/2 + 1 \quad \text{for} \quad p+q = 2, 3\ldots \tag{16}$$

A set of seven invariant moments can be derived from the second and third moments [29]. The set consists of groups of nonlinear centralized moment expressions and is a set of absolute orthogonal moment invariants that can be used for pattern identification invariant to scale, position, and rotation as follows:

$$\begin{aligned}
\phi_1 &= \eta_{20} + \eta_{02} \\
\phi_2 &= (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 \\
\phi_3 &= (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - 3\eta_{03})^2 \\
\phi_4 &= (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \\
\phi_5 &= (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})\left[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2\right] \\
&\quad + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})\left[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\right] \\
\phi_6 &= (\eta_{20} - \eta_{02})(\eta_{30} + \eta_{12})\left[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\right] \\
&\quad + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03}) \\
\phi_7 &= (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})\left[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2\right] \\
&\quad + (3\eta_{12} - \eta_{30})(\eta_{21} + \eta_{03})\left[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\right]
\end{aligned} \tag{17}$$

2.4 Zernike moment analysis

Zernike moment features can also provide the properties of invariance to scale, position, and rotation [30]. In this research, Zernike moment analysis is used to extract invariant features from multiple ROIs. This section gives a brief description of Zernike moment analysis.

The magnitudes of Zernike moments have been treated as rotation-invariant features. It has also been shown that Zernike moments can obtain translation- and scale-invariant properties by simple geometric transformations [30].

The Zernike radial polynomials of order n with repetition m, $V_{nm}(x, y)$ can be written as:

$$V_{nm}(x, y) = R_{nm}(x, y)e^{jm\theta} \tag{18}$$

where

$$j = \sqrt{-1}, \quad \theta = \tan^{-1}(y/x) \tag{19}$$

and

$$R_{nm}(x,y) = \sum_{s=0}^{(n-|m|)/2} \times \frac{(-1)^s (x^2+y^2)^{(n-2s)/2} (n-s)!}{s!((n+|m|-2s)/2)!((n-|m|-2s)/2)!} \tag{20}$$

where $s = 0, 1, \ldots, (n - |m|)/2$, $n \geq 0$, $|m| < n$, and $n - |m|$ is even.

The angle θ lies between 0 and 2π and is measured with respect to the x-axis in a counterclockwise direction. The origin of the coordinate scheme is at the center of the image.

For a digital image, the Zernike moments of order n and repetition m are given by:

$$A_{nm} = \frac{n+1}{\pi} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) V_{nm}^*(x, y) \tag{21}$$

where $V_{nm}^*(x, y)$ is the complex conjugate of $V_{nm}(x, y)$.

One of the major properties of Zernike moments is that the image can be reconstructed using the inverse transformation:

$$\hat{f}(x, y) = \sum_{n=0}^{n_{\max}} \sum_{m=-n}^{n} A_{nm} V_{nm}(x, y) \tag{22}$$

where $n_{\max}$ is the maximum order of the Zernike moments considered for a particular application.

The magnitudes of the Zernike moments $|A_{nm}|$ are rotation-invariant. They may also be invariant to translation and scale.

The Zernike moment $|A_{nm}|$ is of order n with repetition m, where n is a nonnegative integer and m is an integer subject to the constraints $n - |m| =$ even, $|m| \leq n$. It has been found that low-order Zernike moments are stable under linear transformations, while high-order moments have large variations. Therefore, the order n was chosen to be less than five, and the first ten Zernike moments ($n \leq 5$) can be defined as: $A_{0,0}$, $A_{1,1}$, $A_{2,0}$, $A_{2,2}$, $A_{3,1}$, $A_{3,3}$, $A_{4,0}$, $A_{4,2}$, $A_{4,4}$, $A_{5,1}$.

3 Proposed method

The proposed fingerprint recognition system consists of three stages: fingerprint image preprocessing, feature extraction, and ELM (R-ELM) matching, as shown in Fig. 1.

The preprocessing module consists of three main steps: image enhancement, determination of reference points, and determination of the ROI. The feature extraction module consists of three main steps: invariant moment analysis, combination of invariant moment analyses, and principal component analysis (PCA). The techniques used for preprocessing and feature extraction are briefly described below. To speed up the overall feature extraction process, only a predefined area (ROI) around the reference point of the fingerprint image is used instead of the entire fingerprint. The center of the cropped ROI image is the position of the reference point which was determined in the

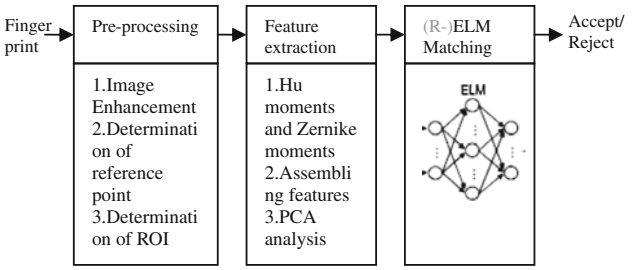

Fig. 1 Overview of the proposed fingerprint recognition system

previous step. To account for both global and local information and to reduce the effects of noise and nonlinear distortions, multiple ROIs of different sizes are used to extract features to represent a fingerprint. To locate a unique reference point consistently for all types of fingerprints and partial fingerprints, the reference point is defined as, "the point of the maximum curvature on the convex ridge" [10], which is usually located in the central area of the fingerprint. The feature vector has 51 elements and consists of three sets of moments derived from three different sub-ROIs. Because the three sub-ROIs are defined using similar ridge structures in a fingerprint image, the features obtained from these three sub-ROIs may be strongly correlated. Therefore, the dimension of the feature vector is reduced by PCA, which examines the feature covariance matrix and then selects the most distinct features.

The last part of the system, the matching module, was implemented using the ELM (R-ELM) for training and testing stages. In the ELM training stage, fingerprint images of the different individuals to be verified are first processed by the preprocessing module and the feature extraction module, and then their extracted features are stored as templates in the database for later use and for the ELM to learn the SLFN in the matching module. In the test stage, a fingerprint image of an individual to be verified is first processed by the preprocessing module and the feature extraction module, and then its extracted features are fed to the matching module with the individual's identity ID (the identity ID is associated with the testing samples to check the classification rate), which matches them against the corresponding templates in the database. Both stages contain the same preprocessing, feature extraction, and matching modules. The proposed fingerprint matching system is explained in detail below.

3.1 Preprocessing

3.1.1 Step 1: Image enhancement

The input fingerprint image is enhanced using a two-stage enhancement algorithm [31]. The performance of a fingerprint matching algorithm depends critically on the quality of the input fingerprint image. Although the quality of a fingerprint image cannot be objectively measured, it roughly corresponds to the clarity of the ridge structure in the image, and hence it is necessary to enhance the image. The enhancement algorithm enhances the image in both the spatial and frequency domains using two-stage filtering, making use of the information from the first enhanced image to estimate the parameters for the second filter. Thus, the algorithm will enhance the fingerprint image completely and effectively.

3.1.2 Step 2: Determination of the reference point

In this step, a reference point is determined from the enhanced image instead of directly from the original image. The ridge orientation field is obtained from the enhanced image, which increases the reliability and accuracy of the determination. The robust and reliable determination of a reference point can be accomplished by detecting the maximum curvature using complex filtering methods with multiple resolutions [32] applied to ridge-orientation field images. The multiple-resolution method will help to overcome potential orientation-field estimation error from poor-quality original images. Figure 2 shows an example of an original fingerprint image and its reference point displayed on its enhanced counterpart. An original fingerprint image in Fig. 2a is enhanced to yield the image in Fig. 2b in the "image enhancement" step, and the position and orientation of its reference point are determined in the "reference point determination" step, as shown in Fig. 2b.

3.1.3 Step 3: Determination of the ROI

In this step, three ROIs of size 64 × 64, 32 × 32, and 16 × 16 are used for feature extraction to represent a

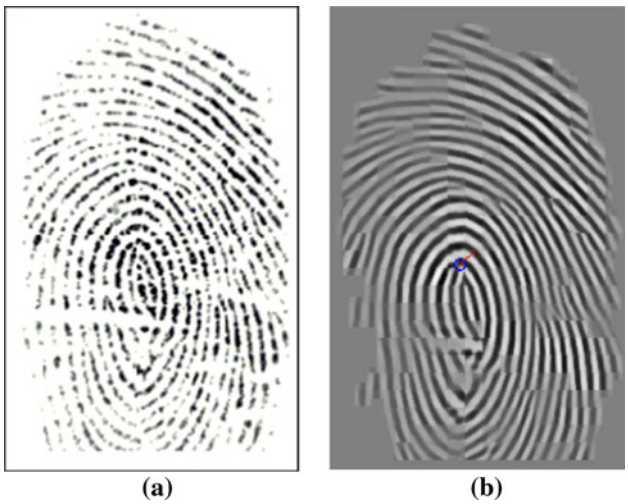

Fig. 2 Example of an original fingerprint image and its reference point determination: **a** an original fingerprint (size 296 × 560); **b** a reference point displayed on the enhanced image of **a**

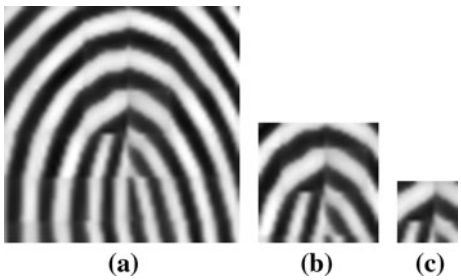

Fig. 3 Three ROIs used for feature extraction: **a** 64 × 64, **b** 32 × 32, **c** 16 × 16

fingerprint, as shown in Fig. 3. The sizes of the ROIs are determined by experiments; if the ROIs are too large, they may easily extend outside of the fingerprint foreground. In this experiment, 64 × 64 is the optimal size for the biggest ROI. The set of invariant moments introduced in Sect. 2 is computed for each ROI, and multiple sets of geometric moment and Zernike moment features are then extracted.

3.2 Feature extraction

3.2.1 Step 1: Hu moments and Zernike moments

In this step, the geometric moments and Zernike moments introduced in Sect. 2 are computed on each sub-ROI. To obtain stronger regional representation abilities, the geometric moment and Zernike moment features are extracted from the binary image instead of directly from the gray-level image. Each gray-level sub-ROI is first binarized by applying local thresholds on the regions according to the local variance of the subimages.

3.2.2 Step 2: Assembling invariant moment features

In this step, three sets of invariant moments consisting of both geometric moments and Zernike moments are extracted as features to represent a fingerprint. Let $\phi_{k,l}$, $k = 1, 2, 3$, $l = 1, 2, 3, \cdots, 17$, be an element of the three sets of features, where $\phi_{k,l}$ for $l = 1, 2, 3, \cdots, 7$ consists of geometric moments and $\phi_{k,l}$ for $l = 8, 9, 10, \ldots, 17$ consists of Zernike moments, and k is the index of sub-ROIs. The total feature vectors $V = [\phi_{1,1}, \phi_{1,2}\ldots, \phi_{1,17}, \phi_{2,1}, \phi_{2,2}\ldots, \phi_{2,17}, \phi_{3,1}, \phi_{3,2}\ldots, \phi_{3,17}]$ are combinations of the three sets of geometric moments and Zernike moments. The length of the total feature vectors is $3 \times 17 = 51$.

3.2.3 Step 3: Feature selection using PCA

PCA is an effective technique for reducing the dimension of a data set while retaining as much as possible of the variation in the data set [33]. Assume that a set $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\} \in R^{u \times N}$ of N training samples is given, where $\mathbf{x}_i$ is a column vector corresponding to one image and u is the length of the feature vector.

The average feature value of the training set can be expressed as follows:

$$\bar{\mathbf{M}} = \frac{1}{N} \times \sum_{i=1}^{N} \mathbf{x}_i \tag{23}$$

The distance from each feature to the average feature can be expressed as follows:

$$\mathbf{M}_i = \mathbf{x}_i - \bar{\mathbf{M}}, \quad i = 1, 2, \ldots N \tag{24}$$

Assume a matrix $\mathbf{M} = [\mathbf{M}_1, \mathbf{M}_2, \ldots \mathbf{M}_N]$ and a corresponding covariance matrix $\mathbf{S} = \mathbf{M} \times \mathbf{M}^T$. The next task is to select the v largest eigenvalues with corresponding eigenvectors and to form a projection space $\mathbf{W}_{opt}$. Let $\mathbf{g}_1$, $\mathbf{g}_2$,…, $\mathbf{g}_n$ and λ_1, λ_2,…, λ_n be the eigenvectors and eigenvalues respectively of $\mathbf{S}$, $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$. Then, PCA factors $\mathbf{S}$ into $\mathbf{S} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^{\mathrm{T}}$, with $\mathbf{U} = [\mathbf{g}_1, \mathbf{g}_2, \ldots, \mathbf{g}_n]$ and $\mathbf{\Lambda} = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n)$. It follows that once the PCA of $\mathbf{S}$ is available, the best rank-v approximation of $\mathbf{S}$ can be readily computed. Then the original image vector of dimension u is projected onto the space of dimension v, where $u > v$. Let $\mathbf{W}_{\mathrm{opt}} = [\mathbf{g}_1, \mathbf{g}_2, \ldots, \mathbf{g}_n]$. The dimensionally reduced feature vector after projection is as follows:

$$\mathbf{y} = \mathbf{W}_{\mathrm{opt}}^T \times \mathbf{x}_k, \quad k = 1, 2, \ldots, n \tag{25}$$

3.3 ELM (R-ELM) matching

Because the purpose of fingerprint matching is to judge whether the input fingerprint matches or does not match a particular fingerprint template according to the querying ID, the matching process can be viewed as a two-class classification problem.

An ELM or R-ELM as introduced in Sect. 2 can be used as a two-class classification tool. There are two phases for an ELM- or R-ELM-based fingerprint matching process: training phase and test phase. In the training phase, for each input fingerprint, its feature vectors are obtained through the preprocessing module and the feature extraction module and are fed to the ELM or R-ELM to get the ELM or R-ELM model parameters. In the test phase, an ELM or R-ELM with the trained parameters is used to verify a match between the feature vectors of the test fingerprint image and those of the template image, and the output of the ELM or R-ELM is the closest output class. The ELM (R-ELM)-based fingerprint matching process is summarized as below:

(a) Training phase:

1. First, we select the training set $TrainSet = \{x_i, y_i\}, i = 1, \ldots, N$ from the fingerprint database, where N is the

total training samples, and x_i, y_i identifies the feature vectors as ELM or R-ELM classifier's input and target, respectively;

2. Randomly set weights a_i between input nodes and hidden layer, and the bias of hidden layer nodes b_i;
3. For each sample of training set, after learning, the actual output value for the entire training set is $Y = \sum_{i=1}^{L} \beta_i K(a_i \times X_j + b_i)$, where L is number of hidden layer nodes;
4. Select the appropriate activation function $f(x)$, options could be: Sine, Sigmoidal, Hardlim, Radial Basis;
5. The ideal target output value matrix T equals Y; according to the expression (3), we may get the hidden layer matrix H;
6. According to the expression (6) or (12), the connection weights β for ELM or R-ELM are determined;
7. Get the optimal parameters $W = (a_i, b_i, \beta_i)$, after the ELM or R-ELM model was trained, these parameters are recorded.

(b) Test phase:

1. As the same steps in the training phase, using the trained parameters obtained from the training model, we can obtain the actual output through the test image set by $Y = \sum_{i=1}^{L} \beta_i K(a_i \times X_j + b_i)$, where L is number of hidden layer nodes.
2. Compared with the desired target, we can determine the test image to be matched or not.

4 Experimental results

The fingerprint image database used in this experiment is the FVC2002 fingerprint database set [34, 35], which contains four distinct databases: DB1_a, DB2_a, DB3_a, and DB4_a. The resolution of DB1_a, DB3_a, and DB4_a is 500 dpi, and that of DB2_a is 569 dpi. Each database consists of 800 fingerprint images at 256 grayscale levels (100 persons, 8 fingerprints per person). Because the databases include some cases in which the reference points of the fingerprint images are near to or on the border of the fingerprint image due to small sensor size, which would lead to failure to acquire ROIs for computing features, in these experiments, the average of two feature vectors obtained from the previous and next impressions was used as the feature vector of the current impressions; similar process can be extended to data sets containing more rejected impressions.

For ELM-based fingerprint matching (ELM source code can be download from [36]), in each subdatabase of FVC2002, six out of eight fingerprints per person were used for training, while testing was performed on all patterns. This means that the training data and testing data contained 600 and 800 samples, respectively. The training data and testing data were normalized by scaling the maximum values of each column into (0,1] to avoid non-convergence. The feature vector with $17 \times 3 = 51$ elements, which consists of three sets of moments derived from three different sets of ROIs, was analyzed by PCA for feature selection. Figure 4 shows the spectrum of eigenvalues with different elements obtained using PCA for feature selection. From Fig. 4, it is apparent that the entire spectrum of eigenvalues is concentrated below 20. To keep 98% of the variation, fifteen eigenvectors ($k = 15$) were chosen. The new 15-dimensional features obtained from the PCA are uncorrelated. Then, the preprocessed training and testing data are fed to the ELM or R-ELM for matching.

Figure 5 shows the variation in recognition rate with respect to the number of hidden neurons and different types of ELM using the 15-dimensional features for fingerprint matching. It is clear that the recognition rate increases with the number of hidden neurons. The sigmoidal and sine types of ELM can achieve high recognition rates in cases with fewer hidden neurons and achieve nearly the same curves, while the Hardlim ELM has a lower recognition rate for the same number of hidden neurons as the other types. Therefore, the sigmoidal type of ELM is used for the following experiments. It should be noted that if the number of hidden neurons is large enough (for example, 150), then the sigmoidal type of ELM can achieve a high recognition rate. Because it needs no tuning of ELM parameters, it is easier to use than other learning methods. In the experiments of R-ELM, sigmoidal type was

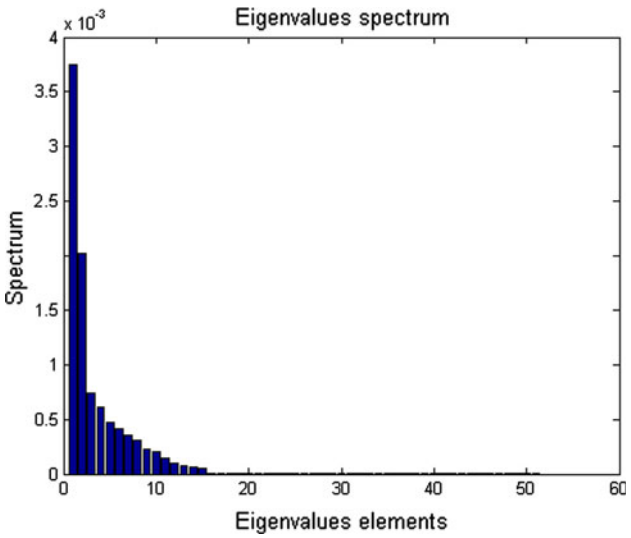

Fig. 4 Eigenvalue spectrum with different elements obtained using PCA for feature selection

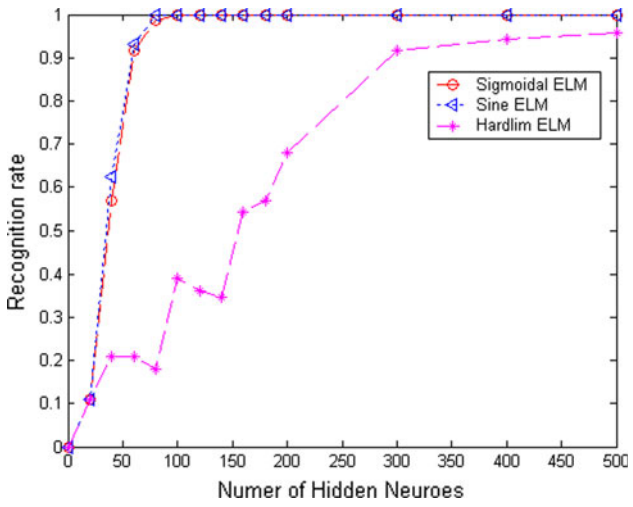

Fig. 5 Recognition rate with respect to the number of hidden neurons and different types of ELM using the 15-dimensional features for fingerprint matching

also chosen as the activation function and the parameter $\gamma = 2^5$.

The performance evaluation protocol used in FVC2002 was selected [35]. To compute the false acceptance rate (FAR) and the false reject rate (FRR), a genuine match and an impostor match were performed. For the genuine match, each test impression of each person was compared with the template of the same person. The FRR and FAR are defined as follows:

$$\text{FRR} = \frac{\text{Number of rejected genuine claims}}{\text{Total number of genuine accesses}} \times 100\% \quad (26)$$

$$\text{FAR} = \frac{\text{Number of accepted imposter claims}}{\text{Total number of imposter accesses}} \times 100\% \quad (27)$$

To evaluate the recognition rate of the proposed ELM-based matching method, the receiver operating characteristic (ROC), which is a plot of the genuine acceptance rate (GAR = 1 − FRR) against the FAR, was used. The tests were executed on Pentium 2 GHz machines with MATLAB software. The equal error rate (EER) was used as a performance indicator. The EER indicates the point where the FRR and FAR are equal. The smaller the value of the EER, the better is the performance of the system. To compute the FAR and the FRR, the genuine match and impostor match were performed on the four subdatabases of FVC2002 database. For genuine match, each fingerprint pattern of each person was compared with other fingerprint patterns of the same person. And for impostor match, each test fingerprint pattern was compared with one of fingerprint patterns belonged to other persons. Since there were 600 training (one person 6 patterns) and 800 test patterns for an experiment, the total number of matches for genuine and impostor was $800 \times 6 = 4{,}800$ and $800 \times 99/2 = 39{,}600$ for each subdatabase.

To test the performances of the proposed R-ELM- and ELM-based matching approaches, a statistics analysis of the experimental results was conducted. That is, ten repeated experiments with the same training and testing data were fed to the ELM or R-ELM for matching on FVC2002 DB1_a database, and then, the average and standard division of EER value of these ten experiments were recorded. The average EER value of these experiments for ELM and R-ELM is 1.21 and 1.10, respectively, and with a standard division of 0.1 and 0.07. That is to say that, comparing with R-ELM, the output matrix of ELM has a relative larger variations and is more sensitive to the disturbances of inputs because of random selection of the input weights.

Figure 6 shows a comparison of the ROC curves obtained from the BPNN-based method [9], the SVM-based method [15], and the proposed ELM- or R-ELM-based method on the FVC2002 DB1_a database. Because the ROC curve of our proposed method (red line) is higher than those of the BPNN-based method and the SVM-based method (other lines), it can be concluded that the proposed method achieves better accuracy than the other two methods on this database. For example, with a FAR of 2%, the GAR values of the proposed method with R-ELM, the proposed method with ELM, the SVM-based method, and the BPNN-based method are 98.7, 98.6, 97.3, and 95.2%, respectively, on the FVC2002 DB1_a database. This means that the genuine acceptance rate of the proposed method with R-ELM is the highest among those tested at the same false acceptance rate and it has the best performance.

Table 1 shows the EER values of the proposed ELM and R-ELM matching methods with BPNN-based and SVM-

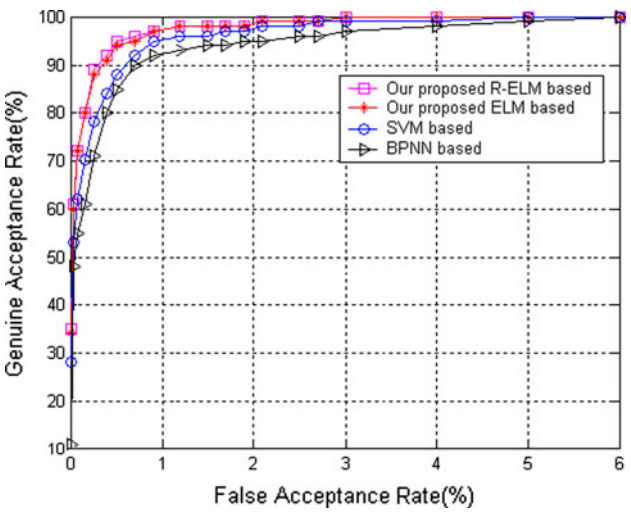

Fig. 6 ROC curves indicating the recognition rate of the proposed method with ELM and R-ELM, BPNN-based method, and SVM-based method on database FVC2002 DB1_a

Table 1 EER (%) of the proposed ELM- or R-ELM-based methods, BPNN-based method, and SVM-based method over the four FVC2002 databases

	DB1_a	DB2_a	DB3_a	DB4_a	Average
BPNN	2.17	3.13	4.25	3.63	3.29
SVM	1.32	2.18	2.38	2.21	2.02
Proposed ELM	1.21	2.13	2.24	2.12	1.93
Proposed R-ELM	1.11	2.08	2.16	2.09	1.86

Table 2 Comparison on average training time, average testing time, and average total time of the proposed ELM- or R-ELM-based methods with those of the BPNN-based method and the SVM-based method over the four FVC2002 databases

	Average training time (s)	Average testing time (s)	Average total time (s)
BPNN	40.94	0.82	41.76
SVM	0.45	0.15	0.60
Proposed ELM	0.25	0.13	0.38
Proposed R-ELM	0.27	0.13	0.40

based methods applied to the four FVC2002 databases. Different databases have different EERs according to the quality of their members and the methods applied. The average EER value for ELM and R-ELM matching over the four databases was 1.93 and 1.86%. The results show that the proposed ELM and R-ELM methods have the best accuracy and achieve significant improvements of 0.09 and 0.16% in performance over the better of the other two prominent methods.

In Table 2, the average training time, average testing time, and average total time of the proposed ELM-based method are compared with those of the BPNN-based method and the SVM-based method for the four FVC2002 databases. It can be seen that the average training time, average testing time, and average total time of the proposed ELM- or R-ELM-based method are 0.25 or 0.27 s, both 0.13 s, and 0.38 or 0.40 s, respectively, which are satisfactory values for real-time processing. The simulation results indicate that the proposed method outperforms the other methods. The average total time for the proposed ELM- or R-ELM-based method is less than those of the BPNN-based method and the SVM-based method.

5 Conclusions and further works

This paper proposes a novel fingerprint recognition system which is the first to apply ELM or R-ELM to fingerprint matching. It uses uncorrelated 15-dimensional feature vectors extracted from fingerprint images by the proposed preprocessing module and feature extraction module. The proposed ELM- or R-ELM-based fingerprint matching method overcomes the disadvantages of traditional learning methods, such as high computational complexity and difficult parameter tuning. An ELM or R-ELM is used to verify the identification of test fingerprint inputs by matching feature vectors of the test inputs with those of template images. The proposed scheme is based on effective preprocessing, local and global invariant moment feature extraction, PCA feature selection, and powerful classification. A preprocessing enhancement with two-stage analysis makes the algorithm highly robust to poor-quality fingerprint images and improves matching accuracy. With the benefit of image enhancement, the reference point can be reliably and accurately determined with complex filtering methods, invariant moment feature extraction, and PCA feature selection, resulting in better features for matching. Experimental results show that the proposed method with ELM or R-ELM has higher matching accuracy and greater speed compared with two traditional methods; thus, they are suitable for real-time processing. Further research should aim to improve the robustness and reliability of the proposed method. Further works with other improved versions of the ELM, such as FIR-ELM, to promote the quality of the fingerprint matching work will be conducted.

Acknowledgment This work is supported by the National Natural Science Foundation of China (No. 61063035), and it is also supported by National Research Foundation of Korean Grant funded by the Korean Government (2009-0077772).

References

1. Maltoni D, Maio D, Jain AK, Prabhakar S (2003) Handbook of fingerprint recognition. Springer, Berlin, pp 164–165
2. Jang X, Yau WY (2000) Fingerprint minutiae matching based on the local and global structures. In: Proceedings of international conference on pattern recognition, vol 2, pp 1024–1045
3. Liu J, Huang Z, Chan K (2000) Direct minutiae extraction from gray-level fingerprint image by relationship examination. In: International conference on image processing, vol 2, pp 427–430
4. Jain AK, Prabhakar S, Hong L, Pankanti S (2000) Filterbank-based fingerprint matching. IEEE Trans Image Process 9:846–859
5. Sha LF, Zhao F, Tang XO (2003) Improved fingercode for filterbank-based fingerprint matching. Int Conf Image Process 2:895–898
6. Tico M, Kuosmanen P, Saarinen J (2001) Wavelet domain features for fingerprint recognition. Electron Lett 37(1):21–22
7. Amornraksa T, Tachaphetpiboon S (2006) Fingerprint recognition using DCT features. Electron Lett 42(9):522–523
8. Yang JC, Yoon S, Park DS (2006) Applying learning vector quantization neural network for fingerprint matching. Lecture Notes Artif Intell (LNAI) 4304:500–509
9. Yang JC, Park DS (2008) Fingerprint verification based on invariant moment features and nonlinear BPNN. Int J Control Automat Syst 6(6):800–808

10. Yang JC, Park DS (2008) A fingerprint verification algorithm using tessellated invariant moment features. Neurocomputing 71(10–12):1939–1946
11. Ji L, Yi Z, Shang L, Pu X (2007) Binary fingerprint image thinning using template-based PCNNs. IEEE Trans Syst Man Cybern B Cybern 37(5):1407–1413
12. Kang J, Zhang W (2009) Fingerprint segmentation using cellular neural network. Int Conf Comput Intell Nat Comput 2:11–14
13. Yang G, Shi D, Quek C (2005) Fingerprint minutia recognition with fuzzy neural network. In: Proceedings of ISNN, vol 2, pp 165–170
14. Wilson CL, Candela GT, Watson CI (1993) Neural network fingerprint classification. J Artif Neural Netw 1(2):203–228
15. Yang JC, Xiong N, Vasilakos AV, Fang Z, Park DS, Xu H, Yoon S, Xie SJ, Yang Y (2011) A fingerprint recognition scheme based on assembling invariant moments for cloud computing communications. IEEE Syst J
16. Jia J, Cai L, Lu P, Liu X (2007) Fingerprint matching based on weighting method and the SVM. Neurocomputing 70(4–60): 849–858
17. Su TJ, Du YY, Cheng YJ, Su YH (2005) A fingerprint recognition system using cellular neural networks. In: 9th International workshop on cellular neural networks and their applications, pp 170–173
18. Hong H, Hua LJ (2003) Fingerprint matching using ANFIS. IEEE Int Conf Syst Man Cybern 1:217–222
19. Huang GB, Wang D, Lan Y (2011) Extreme learning machines: a survey. Int J Mach Learn Cybern 2(2):107–122
20. Cao FL, Xie TF, Xu ZB (2008) The estimate for approximation error of neural networks: a constructive approach. Neurocomputing 71(4–6):626–630
21. Feng GR, Huang G-B, Lin QP et al (2009) Error minimized extreme learning machine with hidden nodes and incremental learning. IEEE Trans Neural Networks 20(8):1352–1357
22. Huang G-B, Zhu Q-Y, Siew C-K (2006) Extreme learning machine: theory and applications. Neurocomputing 70(6):489–501
23. Wu J, Wang S, Chung FL (2011) Positive and negative fuzzy rule system, extreme learning machine and image classification. Int J Mach Learn Cybern 16(8):1408–1417
24. Wang X, Chen A, Feng H (2011) Upper integral network with extreme learning mechanism. Neurocomputing 74(16):2520–2525
25. Deng W, Zheng Q, Chen L (2009) Regularized extreme learning machine. In: IEEE symposium on computational intelligence and data mining, pp 389–395
26. Man Z, Lee K, Wang D, Cao Z, Miao C (2011) A new robust training algorithm for a class of single-hidden layer feedforward neural networks. Neurocomputing 74(16):2491–2501
27. Huang G-B, Ding X, Zhou HM (2010) Optimization method based extreme learning machine for classification. Neurocomputing 74(12):155–163
28. Barlett PL (1998) The sample complexity of pattern classification with neural networks: the size of the weights is more important than the size of the network. IEEE Trans Inf Theory 44(2): 525–536
29. Hu MK (1962) Visual pattern recognition by moment invariants. IRE Trans Inf Theory 8:179–187
30. Gonzalez RC, Woods RE (2002) Digital image processing, 2nd edn. Prentice Hall, Englewood Cliffs
31. Yang JC, Xiong N, Vasilakos AV (2012) Two-stage enhancement scheme for low-quality fingerprint images by learning from the image. IEEE Trans Syst Man Cybern C (in press)
32. Kenneth N, Josef B (2003) Localization of corresponding points in fingerprints by complex filtering. Pattern Recogn Lett 24:2135–2144
33. Fausett L (1994) Fundamentals of neural networks: architecture, algorithm, and application. Prentice Hall, Englewood Cliffs, pp 289–304
34. http://bias.csr.unibo.it/fvc2002/
35. Maio D, Maltoni D, Cappelli R, Wayman JL, Jain AK (2002) FVC2002: second fingerprint verification competition. In: Proceedings of 16th international conference on pattern recognition, vol 3, pp 811–814
36. http://www.ntu.edu.sg/home/egbhuang/