Available online at www.sciencedirect.com
SCIENCE DIRECT®
NEUROCOMPUTING
Neurocomputing 69 (2006) 1782–1786
www.elsevier.com/locate/neucom

Letters

Palmprint recognition using FastICA algorithm and radial basis probabilistic neural network ☆

Li Shang[a,b], De-Shuang Huang[b,*], Ji-Xiang Du[a,b], Chun-Hou Zheng[a,b]

[a]*Department of Automation, University of Science and Technology of China, Hefei, Anhui 230026, China*
[b]*Institute of Intelligent Machines, Chinese Academy of Sciences, P.O. Box 1130, Hefei, Anhui 230031, China*

Received 6 September 2005; received in revised form 15 November 2005; accepted 16 November 2005
Available online 24 February 2006
Communicated by R.W. Newcomb

Abstract

This paper proposes a novel and successful method for recognizing palmprint based on radial basis probabilistic neural network (RBPNN) proposed by us. The RBPNN is trained by the orthogonal least square (OLS) algorithm and its structure is optimized by the recursive OLS algorithm (ROLSA). The Hong Kong Polytechnic University (PolyU) palmprint database, which is pre-processed by a fast fixed-point algorithm for independent component analysis (FastICA), is exploited to test our approach. The experimental results show that the RBPNN achieves higher recognition rate and better classification efficiency than other usual classifiers.

Keywords: FastICA algorithm; Palmprint recognition; Radial basis probabilistic neural network (RBPNN); Recognition rate

1. Introduction

Recently, biometric personal identification is emerging as a powerful means for automatically recognizing a person's identity with a higher confidence. Biometric palmprint verification is such a technology, which recognizes a person based on unique features in his palm, such as the principal lines, wrinkles, ridges, minutiae points, singular points and texture, etc. For the designers of pattern recognition algorithms, palmprint recognition is a very challenging problem. Many recognition methods, such as the nearest feature line method [5], the Cosine measure [1], the Fisher classifier [7] and neural networks (NN) method [8], etc., have been proposed. This paper focuses on using a novel radial basis probabilistic neural network (RBPNN) model [8] to perform the palmprint recognition task. Features of palmprint images can be extracted by certain transforms, such as Fourier transform [6], wavelets-based transform [4], principal component analysis (PCA) and independent component analysis (ICA) [2], etc. Fourier and wavelet transforms have strong mathematical foundations and fast implementations, but they are not of adaptive ability to particular data. While the significant advantage of the PCA and ICA is that they only depend on the statistic properties of image data. However, the PCA technique is usually suitable for the second order accumulation variant, whereas the ICA method can be used for multi-dimensional data. Here, we use a fast fixed-point algorithm for independent component analysis (FastICA) to extract successfully features of palmprint images since it is a neural algorithm particularly efficient and light from the point of view of computational effort [3].

2. The FastICA algorithm

Independent component analysis (ICA) of observed n-dimensional random vectors $X(X=(x_1,x_2,\ldots,x_n)^{\mathrm{T}})$ is defined as the process of finding an m-dimensional linear transform $S=WX(S=(s_1,s_2,\ldots,s_m)^{\mathrm{T}})$ such that the separation matrix W is a linear transformation and the hidden components s_i are as independent as possible.

☆This work was supported by the National Science Foundation of China (Nos. 60472111 and 0405002), and partially supported by Hong Kong Research Grants Council (Nos. CUHK4205/043E).
*Corresponding author.
E-mail addresses: shangli@iim.ac.cn (L. Shang), dshuang@iim.ac.cn (D.-S. Huang), du_jx@iim.ac.cn (J.-X. Du), zhengch@iim.ac.cn (C.-H. Zheng).

0925-2312/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2005.11.004

Usually, the dimension of S is less than that of $X(m \leqslant n)$. In this paper, note that only the case of $m = n$ is considered. Here, the matrix W is assumed to be invertible. FastICA is just such an algorithm, which is based on a fixed-point iteration scheme for finding a maximum nongaussianity of WX. There are different measures of nongaussianity, such as kurtosis (fourth-order cumulant) and negentropy. In this paper, we make use of the kurtosis as the maximum nongaussianity measure (i.e., the score function), which is defined for a zero-mean random variable s as [3]

$$kurt(s) = E\{s^4\} - 3(E\{s^2\})^2. \tag{1}$$

Using the natural gradient method under the constraint $||w_j||_2 = 1$ (where w_j is the jth column of W), the updating rule of w_j is written as

$$\begin{aligned} w_j(t+1) = w_j(t) + \mu(t)\big\{ & v_i(t)[(w_j(t))^{\mathrm{T}} v_i(t)]^3 \\ & -3||w_j(t)||^2 w_j(t) + f(||w_j(t)||^2) w_j(t)\big\}, \end{aligned} \tag{2}$$

where v_i is the ith row of whitened results of observed vectors X, $(w_j(t))^{\mathrm{T}}$ is the transpose of $w_j(t)$, $\mu(t)$ is the learning rate, and $f(\cdot)$ is a penalty term due to the constraint $||w_j|| = 1$. The learning rule will stop at a fixed point for which $|w_j^{\mathrm{T}}(t) - w_j(t-1)|$ is sufficiently close to unity. The linear combination WX will be one of the required independent components in accordance with the formula of $S = WX$.

3. Two architectures of performing FastICA

According to the literature [1], there are two types of implementation architectures for FastICA in the image recognition task. Architecture I treats palmprint images as random variables and pixels as observations, i.e., the palmprint images are in rows and the pixels in columns. This goal in this approach is to find a $n \times m$ matrix D such that the rows of $U = DX$ are as statistically independent as possible. The source images estimated by the row of U are then used as basis images to represent palmprints. Palmprint image representations consist of the coordinates of these palmprint images with respect to the image basis defined by the rows of U (i.e., coefficients of basis images), as shown in Fig. 1.

On the contrary, architecture II utilizes pixels as random variables and palmprint images as observations, i.e., the pixels are in rows and the palmprint images in columns. This goal in this approach is to find a representation in which all coefficients are as statistically independent as possible. The FastICA representations are in columns of $U = DX$. Each column of U contains the coefficients of the basis images in A (A is the inverse or pseudoinverse of D) for reconstructing each palmprint image in X, see Fig. 2.

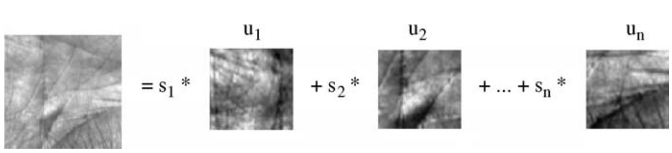

Fig. 1. The independent basis image representation consisted of the coefficients, S, for the linear combination of independent basis images, $U = (u_1, u_2, \ldots, u_n)$, that comprised each palmprint image. FastICA representation is in S ($S = (s_1, s_2, \ldots, s_n)$).

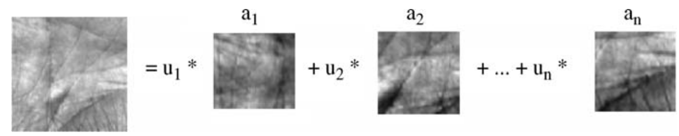

Fig. 2. The factorial representation consisted of the independent coefficients, U, for the linear combination of basis images in $A = (a_1, a_2, \ldots, a_n)$, that comprised each palmprint image. FastICA factorial representation is in U ($U = (u_1, u_2, \ldots, u_n)$).

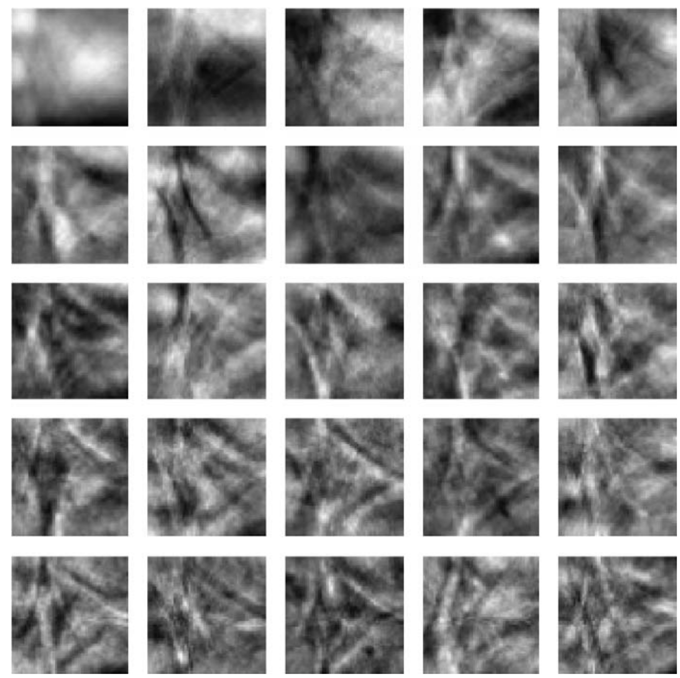

Fig. 3. First 25 PC axes of the palmprint image set (columns of V), ordered left to right, top to bottom, by the magnitude of the corresponding eigenvalues.

Convenient for calculating, we use PCA to realize data whitening and a dimensional reduction before performing the FastICA algorithm. Let V_k denote a matrix with the size of $p \times k$, namely it contains the first k principal components (PC) axes in its columns (see Fig. 3, where $k = 25$), where p is the number of pixels in a training image. In architecture I let X_I denote the n-dimensional set of zero-mean images (image data is contained in each row), the independent basis vector U is computed as follows:

$$U = W_I * V_k^{\mathrm{T}}, \tag{3}$$

where $W_I = W * [2(COV(X_I^{\mathrm{T}}))^{-1/2}]$, W is the matrix learned by ICA, X_I^{T} is the transpose of X_I with the size of $n \times p$, and $COV(X_I^{\mathrm{T}})$ is the covariance matrix of X_I^{T}. Note that the ICA basis images from this architecture show more localized features [1], see Fig. 4. Furthermore, by

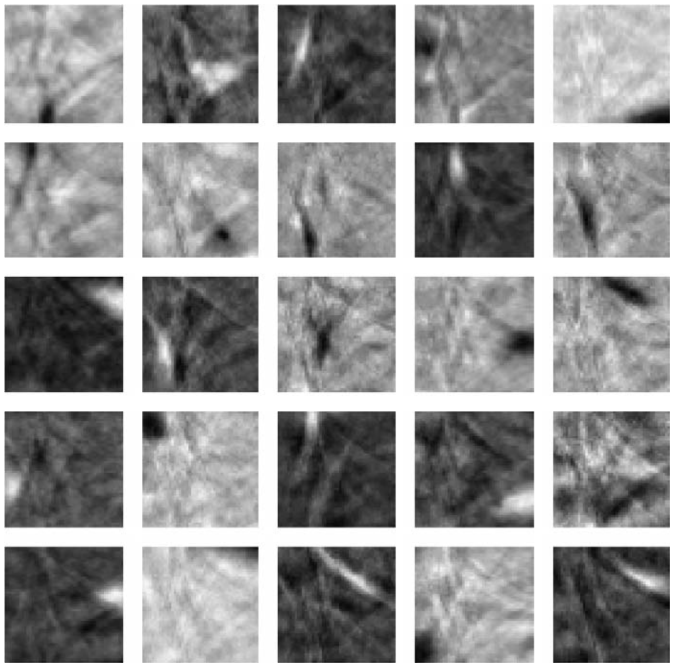

Fig. 4. The first 25 basis images (rows of $W_I V_k^{\mathrm{T}}$) that are obtained by FastICA architecture I. In this approach, the basis images are statistically independent.

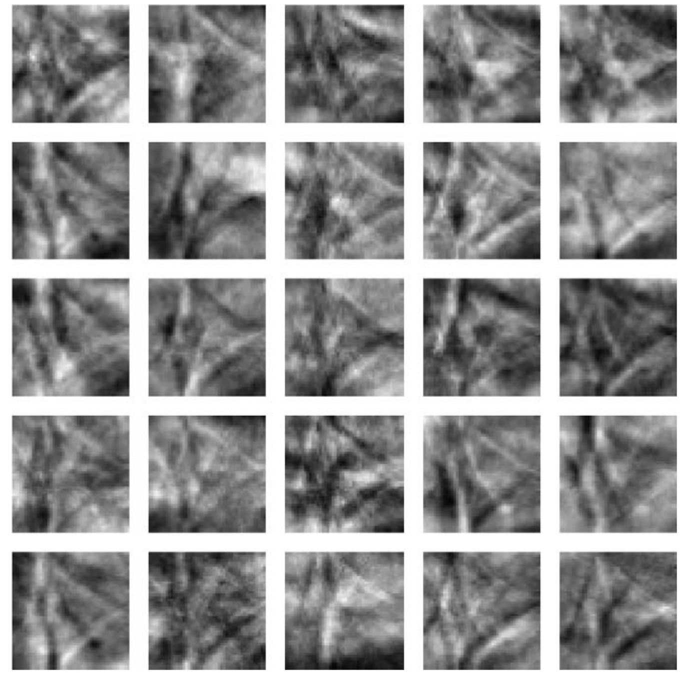

Fig. 5. The first 25 basis images (columns of $V_k W_I^{-1}$) obtained by architecture II. In this approach, the coefficients are statistically independent.

taking R_k as the PCA coefficients, where $R_k = X_I V_k$, the coefficient matrix can be calculated as $S = R_k * W_I^{-1}$. But, in architecture II, the source separation is performed on pixels. Here, FastICA is performed on the PCA coefficients rather than directly on the input images. The statistically independent coefficients are computed as $W_I R_k^{\mathrm{T}}$. The basis vectors are obtained from the columns of $V_k W_I^{-1}$. Note that the basis generated by this architecture shows more globalized features, see Fig. 5.

4. The RBPNN model and training algorithm

The radial basis probabilistic neural network (RBPNN) model [8] proposed by us, as shown in Fig. 6, was derived from the radial basis function neural network (RBFNN) and the probabilistic neural networks (PNN). Hence, the RBPNN possesses the characters of the above two networks, i.e., the signal is concurrently feed-forwarded from the input layer to the output layer without any feedback connections within the three layers network models. On the other hand, to some extent, it lowers the demerits of the two original models. It can be seen that this network consists of four layers. The first hidden layer is a nonlinear processing layer, generally consisting of the centers selected from training samples. The second hidden layer selectively sums the outputs of the first hidden layer according to the categories, to which the hidden centers belong. Namely, the connection weights between the first hidden layer and the second hidden layer are 1's or 0's. For pattern recognition problems, the outputs in the second hidden layer need to be normalized. The last layer for the RBPNN is just the output layer.

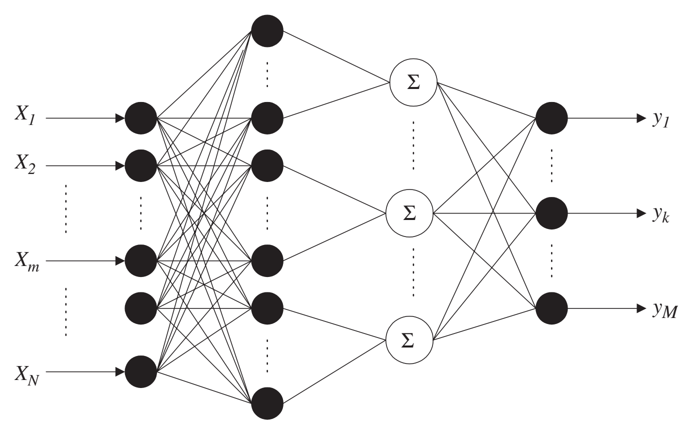

Fig. 6. The structure of radial basis probabilistic neural network.

Mathematically, for an input vector x, the actual output value of the ith output neuron of the RBPNN, y_i^a, is expressed as

$$y_i^a = \sum_{k=1}^{M} w_{ik} h_k(x), \tag{4}$$

$$h_k(x) = \sum_{i=1}^{n_k} \phi_i(||x - c_{ki}||_2), \quad k = 1, 2, 3, \ldots, M \tag{5}$$

where $h_k(x)$ is the kth output value of the second hidden layer of the RBPNN; w_{ik} is the synaptic weight between the kth neuron of the second hidden layer and the ith neuron of the output layer of the RBPNN; c_{ki} is the ith hidden center vector for the kth pattern class of the first hidden layer; n_k represents the number of hidden center vectors for the kth pattern class of the first hidden layer; $||\cdot||_2$ is Euclidean norm; and M denotes the number of the neurons of the output layer and the second hidden layer, or the pattern class number for the training samples set; $\phi_i(\cdot)$ is the kernel function, which is generally the Gaussian kernel function. $\phi_i(||x-c_{ki}||_2)$ is written as

$$\phi_i(||x-c_{ki}||_2) = \exp\left[-\frac{||x-c_{ki}||_2^2}{\sigma_i^2}\right], \tag{6}$$

where σ_i is the shape parameter for the Gaussian kernel function.

Generally, the training algorithms for the RBPNN include the orthogonal least square algorithm (OLSA) and recursive least square algorithms (RLSA) [8], etc. These two methods have the common advantages of fast convergence and good convergence accuracy. The RLSA, which requires good initial conditions, however, is fit for those problems with a large training samples set. As the OLSA makes full use of matrix computation, such as the orthogonal decomposition algorithm of matrices, its training speed and convergence accuracy is faster and higher than the ones of the RLSA. Therefore, the OLSA is preferred to train the RBPNN in this paper. For N training samples corresponding to M pattern classes, considering the form of the matrix, Eq. (4) can be written as [8]

$$Y^a = HW, \tag{7}$$

where Y^a and H are both an $N \times M$ matrix, W is a square $M \times M$ matrix. According to the literature [8], it can be known that the synaptic weight matrix W between the output layer and the second layer of the RBPNN can be solved for as follows:

$$W = R^{-1}\hat{Y}, \tag{8}$$

where R is an $M \times M$ upper triangular matrix with the same rank as H, and $\hat{Y}$ is an $M \times M$ matrix. Both of them can be respectively obtained as follows:

$$H = Q \times \begin{bmatrix} R \\ \cdots \\ 0 \end{bmatrix}, \quad Q^{\mathrm{T}} \times Y = \begin{bmatrix} \hat{Y} \\ \tilde{Y} \end{bmatrix}, \tag{9}$$

where Q is an $N \times N$ orthogonal matrix with orthogonal columns satisfying $QQ^{\mathrm{T}} = Q^{\mathrm{T}}Q = 1$, and $\tilde{Y}$ is an $(N-M) \times M$matrix. Eq. (9) expresses the orthogonal decomposition of the output matrix H of the second hidden layer of the RBPNN.

5. Experimental results and conclusions

We make use of the Hong Kong Polytechnic University (PolyU) palmprint database, available from http://www.comp.polyu.edu.hk/~biometrics, to verify our RBPNN algorithm. This database includes 600 palmprint images with the size of 128×128 from 100 individuals, with 6 images from each. For each person, the first three images are used as training data while the remaining three are treated as testing data. The interval captured between training samples and test samples is two months. To reduce the computational cost, each image is scaled to the size of 64×64. Thus, the training set is a matrix with 300×4096. By PCA, the dimension of the training set is reduced to 214, i.e., the number of the fist k PC is 214.

Using FastICA architectures, basis vectors (features) of palmprint images are extracted. Then we select all the 214 training samples as the hidden centers of the first hidden layer. The number of the second hidden neurons and the output layer neurons is set as 100 (classes of patterns), respectively. According to [8], the shape parameter σ_i is set as 650. Using the OLSA to train the RBPNN, the recognition rate of the testing samples corresponding to two FastICA architectures is respectively 97.53% and 98.67%. In order to optimize and prune the RBPNN, we use ROLSA to optimize the structure of RBPNN. As a result, the number selected of hidden centers of the first hidden layer is reduced from 214 to 60 and the recognition rate of testing samples corresponding to each FastICA architecture is still 97.53% and 98.67%. Clearly, FastICA architecture II outperforms FastICA architecture I in classification.

Compared with the RBPNN, with the same training and testing data, by using all the training samples as the hidden centers of RBFNN, the maximum recognition rate of RBFNN corresponding to each FastICA architecture is respectively 95.53% and 96.76%, where the shape parameter of the Gaussian kernel function of the RBFNN is about 8900. Likewise, the recognition rate of BPNN corresponding to each FastICA architecture is respectively 95.21% and 97.31%. Thus, it can be clearly seen that the recognition rate of the RBPNN is higher than both that of the RBFNN and that of the BPNN. On the other hand, in experiment, it was found that the training speed and testing speed with the RBPNN are also very fast. The algorithm was programmed with MATLAB 6.5, and it was run on a

Table 1
Training and classification CPU time for the PolyU palmprint database

Methods	FastICA architecture I CPU time (s)		FastICA architecture II CPU time (s)	
	Training	Classification	Training	Classification
RBPNN	2.9	0.020	2.7	0.018
RBFNN	3.2	0.017	3.1	0.015
BPNN	216.65	0.26	214.95	0.22

Pentium IV with 2.6 GHz clock and 256 Mb RAM under the Microsoft Windows XP environment. For each FastICA architecture, the CPU time needed to recognize one palmprint image with the RBPNN is respectively about 0.020 s and 0.018 s, as well as the training CPU time needed is about 2.9 and 2.7 s, respectively. Distinctly, in FastICA architectures, the difference in training and testing speed is very small. As far as the palmprint recognition based on BPNN being concerned, the corresponding concerning training CPU time and classification CPU time are much longer than the ones of the RBPNN, as listed in Table 1.

Therefore, from the above experimental results, it can be concluded that our palmprint recognition method based on FastICA architectures and the RBPNN not only achieves higher statistical recognition rate, but also behaves faster training speed and testing speed than other classifiers used usually in practice. This method is indeed effective and efficient, which greatly support the claim that the RBPNN proposed by us is a very promising neural network model in practical applications.

References

[1] M.S. Bartlett, J.R. Movellan, T.J. Sejnowski, Face recognition by independent component analysis, IEEE Trans. Neural Networks 13 (6) (2002) 1450–1464.

[2] T. Connie, A. Teoh, M. Goh, D. Ngo, Palmprint recognition with PCA and ICA, Image and Vision Computing New Zealand 2003, Palmerston North, New Zealand, November 2003, pp. 232–227.

[3] A. Hyvärinen, Fast and robust fixed-point algorithms for independent component analysis, IEEE Trans. Neural Networks 10 (3) (1999) 626–634.

[4] A. Kumar, H.C. Shen, Recognition of palmprints using wavelet-based features, in: Proceedings of the International Conference on Systems and Cybernetics, SCI-2002, Orlando, Florida, July 2002, pp. 371–376.

[5] S.Z. Li, J. Lu, Face recognition using the nearest feature line method, IEEE Trans. Neural Networks 10 (6) (1999) 439–443.

[6] W. Li, Z. David, Z. Xu, Palmprint identification by Fourier transform, Int. J. Pattern Recognition Art. Intell. 16 (4) (2002) 417–432.

[7] X. Wu, D. Zhang, K. Wang, Fisherpalms based on palmprint recognition, Pattern Recognition Lett. 24 (15) (2003) 2829–2938.

[8] W.B. Zhao, D.S. Huang, Application of recursive orthogonal least squares algorithm to the structure optimization of radial basis probabilistic neural networks. In: Proceedings of the Sixth International Conference on Signal Processing (ICSP2002), Beijing, China, August 2002, pp. 1211–1214.

Li Shang received the B.Sc. degree and M.Sc. degree in Xi'an Mine University in June 1996 and June 1999, respectively. From September 2003, she is in pursuit for Doctor's degree in Pattern Recognition & Intelligent System in University of Science & Technology of China (USTC), Hefei, China. Now, she works at department of automation of USTC. Her research interests include Image processing, Artificial Neural Networks and Intelligent Computing.

De-Shuang Huang (SM'98) Professor and Ph.D. Advisor in the University of Science and Technology of China (USTC), Hefei, China, and Professor of Graduate School of the Institute of Intelligent Machines, Chinese Academy of Sciences (CAS). From September 2000 to March 2001, he worked as Research Associate in Hong Kong Polytechnic University. From April 2002 to June 2003, he worked as Research Fellow in City University of Hong Kong. From October to December 2003, he worked as Research Fellow in Hong Kong Polytechnic University. From July to December 2004, he worked as the University Fellow in Hong Kong Baptist University. Dr. Huang is currently a senior member of the IEEE. He has published over 190 papers.

Ji-Xiang Du From September 1995 to July 1999, took courses as B.Sc. degree candidate in Vehicle Engineering, Hefei University of Technology, and obtained B.Sc. degree in July 1999. From September 1999 to July 2002, took courses as M.Sc. degree candidate in Vehicle Engineering, Hefei University of Technology, and obtained B.Sc. degree in July 2002. From February 2003 on, in pursuit for Ph.D. degree in Pattern Recognition & Intelligent System in University of Science and Technology of China (USTC), Hefei, China, and in December 2005, he received Ph.D. degree. Now, he is engaged in the postdoctoral study in Department of Automation of USTC.

Chun-Hou Zheng was born in Shandong province, China, in 1973. He received M.Sc. degree in Control Theory & Control Engineering in 2001, from Qu Fu Normal University, China. He is now in pursuit for Ph.D. degree in Pattern Recognition & Intelligent System in University of Science and Technology of China. His research interests include Artificial Neural Networks, Intelligent Computing, and Intelligent Information Processing.