

Face Recognition Based on Improved Fuzzy RBF Neural Network for Smart Device

Eung-Joo Lee[†]

ABSTRACT

Face recognition is a science of automatically identifying individuals based their unique facial features. In order to avoid overfitting and reduce the computational burden, a new face recognition algorithm using PCA-fisher linear discriminant (PCA-FLD) and fuzzy radial basis function neural network (RBFNN) is proposed in this paper. First, face features are extracted by the principal component analysis (PCA) method. Then, the extracted features are further processed by the Fisher's linear discriminant technique to acquire lower-dimensional discriminant patterns, the processed features will be considered as the input of the fuzzy RBFNN. As a widely applied algorithm in fuzzy RBF neural network, BP learning algorithm has the low rate of convergence, therefore, an improved learning algorithm based on Levenberg-Marquart (L-M) for fuzzy RBF neural network is introduced in this paper, which combined the Gradient Descent algorithm with the Gauss-Newton algorithm. Experimental results on the ORL face database demonstrate that the proposed algorithm has satisfactory performance and high recognition rate.

Key words: face recognition, PCA-FLD, fuzzy RBFNN, L-M learning algorithm

1. INTRODUCTION

Human face recognition from still and video images has become an active research area in the communities of image processing, pattern recognition, neural networks and computer vision. This interest is motivated by wide applications ranging from static matching of controlled format photographs such as passports, credit cards, driving licenses, and mug shots to real-time matching of surveillance video images presenting different constraints in terms of processing requirements [1].

Face recognition involves the extraction of different features of the human face from the face image for discrimination it from other persons, it has evolved as a popular identification technique to perform verification of human identity.

During the past 30 years, many different face-recognition techniques have been proposed, motivated by the increased number of real-world applications requiring the recognition of human faces. PCA algorithm is known as Eigen face method; In PCA method, the images are projected onto the facial value and called "eigenspace" [2, 3]. PCA approach reduces the dimension of the data by means of basic data compression method [4] and reveals the most effective low dimensional structure of facial patterns [5]. LFA method of recognition is based on the analysis the face in terms of local features e.g. eye, nose etc. by what is referred LFA kernels. Recognition by Neural Network [6] and [7] are based on learning of the faces in an "Example Set" by the machine in the "Training Phase" and carrying out recognition in

※ Corresponding Author : Eung-Joo Lee, Address : Dept. of Information and Communications Engineerings, 535 Yongdang-Dong, Nam-Gu, Busan 608-711, Korea, TEL : +82-51-629-1143, FAX : +82-51-629-1129, E-mail : ejlee@tu.ac.kr

Receipt date : Sep. 12, 2013, Revision date : Oct. 15, 2013
Approval date : Oct. 17, 2013

[†] Department of Information & Communications Engineering, Tongmyong University, Busan, Korean

※ This research was supported by the Tongmyong University Research Grants 2012 (2012-A001).

the "Generalization Phase". Support Vector Machines (SVM) technique is in fact one of the binary classification methods. The support vectors consist of a small subset of training data extracted by the algorithm given in [8]. Face recognition based on template matching represents a face in terms of a template consisting of several enclosing masks the projecting features e.g. the mouth, the eyes and the nose [9]. In [10], a face detection method based on half face-template is discussed.

Although researchers in psychology, neural sciences and engineering, image processing and computer vision have investigated a number of issues related to face recognition by human beings and machines, it is still difficult to design an automatic system for this task, especially when real-time identification is required. The reasons for this difficulty are two-fold: 1) Face images are highly variable and 2) Sources of variability include individual appearance, three dimensional (3-D) pose, facial expression, facial hair, makeup, and so on and these factors change from time to time. Furthermore, the lighting, back-ground, scale, and parameters of the acquisition are all variables in facial images acquired under real-world scenarios [1]. This makes face recognition a great challenging problem.

In recent work, researchers always use a hybrid method, which combines some linear and nonlinear projection methods for obtaining better recognition results. In many of the hybrid methods, the combination of Neural Network and Fuzzy System has become a hot research area in recent years. Reference [11] applied the theory of fuzzy to designing of RBF Neural Network, combined the respective advantage of Neural Network and fuzzy function and derived satisfactory results. But since the problem of learning speed with Fuzzy Neural Network, the optimal procedure is easily stacked into the local minimal value and it causes slow convergence. Generally speaking, multi-layer networks usually coupled with the backpropagation (BP) algorithm, are most widely used in face rec-

ognition [12]. Yet, two major criticisms are commonly raised against the BP algorithm: 1) It is computationally intensive because of its slow convergence speed and 2) there is no guarantee at all that the absolute minima can be achieved. On the other hand, RBF neural networks have recently attracted extensive interests in the community of neural networks for a wide range of applications [13,14,22].

In order to avoid these cases, we propose an improved RBF neural network for face recognition in this paper. The whole recognition procedure includes three stages, feature extraction and classification. Firstly, dimension reduction and face feature extraction using PCA-FLD algorithm is presented; and then the structure and L-M learning algorithm of the fuzzy RBF neural network (RBFNN) are introduced for face classifier. Fig. 1 shows the flow chart of proposed algorithm in this paper.

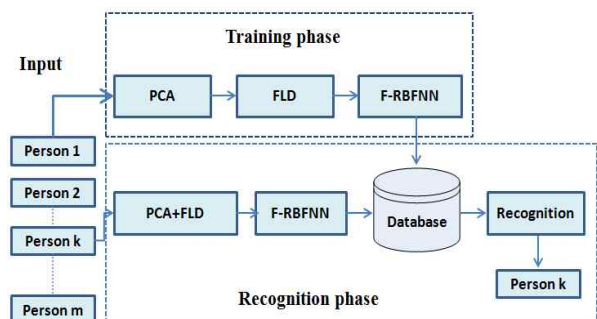


Fig. 1. Flowchart of proposed recognition algorithm.

2. Dimension Reduction and Feature Extraction

As the problem of face recognition, face image data are usually high-dimensional and large-scale, recognition has to be performed in a high-dimensional space. So it is necessary to find a dimensional reduction technique to cope the problem in a lower-dimension space. Researchers have presented many linear and nonlinear projection algorithms such as the eigenfaces [15], Principal

Component Analysis (PCA) [15], Linear Discriminant Analysis (LDA) [16,17], Fisher faces [6], Direct LDA (DLDA) [16,18], Discriminant Common Vector (DCV) [19] and Independent Component Analysis (ICA) [20], etc.

In this paper, we use a hybrid algorithm, which combine PCA algorithm with FLD algorithm, for feature extraction. The number of input variables is reduced through feature selection, i.e., a set of the most expressive features is first generated by the PCA and FLD is then implemented to generate a set of the most discriminant features so that different classes of training data can be separated as far as possible and the same classes of patterns are compacted as close as possible. The procedures are as follows:

(1) Obtaining face images I_1, I_2, \dots, I_M , each image can be expressed as $I(x, y)$. Let the training set of face images be $\Gamma_1, \Gamma_2, \dots, \Gamma_M$.

(2) Calculating the average face Ψ :

$$\Psi = \frac{1}{M} \sum_{i=1}^M \Gamma_i \quad (1)$$

where M presents the number of training face images and Γ_i denotes i^{th} face image vector. (3) Calculating the mean subtracted face Φ_i by

$$\Phi_i = \Gamma_i - \Psi, i = 1, 2, \dots, M \quad (2)$$

The objective of this formula is to project Γ_i to a lower dimension space.

(4) Calculating the covariance matrix C

$$C = \frac{1}{M} \sum_{i=1}^M \Phi_i \Phi_i^T = AA^T \quad (3)$$

where $A = (\Phi_1, \Phi_2, \dots, \Phi_M)$.

(5) Calculating the v_k and λ_k , which indicate eigenvectors and eigenvalues of C matrix. Where v_k determines linear combinations of the M training set of face images from the eigenface u [15]:

$$u_l = \sum_{k=1}^M v_{lk} \Phi_k, k = 1, 2, \dots, M \quad (4)$$

Here, we obtain a set of eigenface vectors $U = [u_1, u_2, \dots, u_M]$. Fig. 2 shows the first 10 ei-



Fig. 2. First 10 eigenfaces with highest eigenvalues.

genfaces on ORL face database.

(6) In order to find a best subspace for classification, and maximize the ratio of between-class scatter and within-class scatter, so computing between-class scatter matrix S_b and within-class scatter matrix S_w [16]:

$$S_b = \sum_{i=1}^c n^i (\bar{u}_i - \bar{U})(\bar{u}_i - \bar{U})^T \quad (5)$$

$$S_w = \sum_{i=1}^c \sum (u_i - \bar{u}_i)(u_i - \bar{u}_i)^T \quad (6)$$

where $\bar{U} = \frac{1}{M} \sum_{i=1}^M u_i$ ($i = 1, 2, \dots, M$) is the mean of eigenface images and $\bar{u}_i = \frac{1}{n^i} \sum_{i=1}^M u_i$ is the mean of i^{th} set of eigenface. c is the number of the classes and n^i represents the number of i^{th} class.

(7) The optimal subspace, $E_{optimal}$ by the FLD is determined as

$$E_{optimal} = \underset{W}{\operatorname{argmax}} \frac{|W^T S_b W|}{|W^T S_w W|} = [w_1, w_2, \dots, w_{c-1}] \quad (7)$$

Where $[w_1, w_2, \dots, w_{c-1}]$ is the set of generalized eigenvectors of S_b and S_w corresponding to the $c-1$ largest generalized eigenvalues $\lambda_i, i = 1, 2, \dots, c-1$. Thus, the feature vectors Ω for any query face images I in the most discriminant sense can be calculated as follows:

$$\Omega = E^{T_{optimal}} \cdot u^T \cdot \Phi \quad (8)$$

The best subspace $E_{optimal}$ is calculated by Lagrange multiplier. Meanwhile, the Ω is considered as the input unit of the fuzzy RBF neural network.

3. The Structure of Fuzzy RBF Neural Network

3.1 Instruction of RBF Neural Network

An RBF neural network can be considered as a mapping [21]: $R^r \rightarrow R^s$.

Let $P \in R^r$ be the input vector and $C_i \in R^r (1 \leq i \leq u)$ be the prototype of the input vectors. The output of each RBF unit is as follows:

$$R_i(P) = R_i(\|P - C_i\|) \quad i = 1, 2, \dots, u \quad (9)$$

where $\|\cdot\|$ indicates the Euclidean norm on the input space. Usually, the Gaussian function is preferred among all possible radial basis functions due to the fact that it is factorizable. Hence

$$R_i(P) = \exp\left[-\frac{\|P - C_i\|^2}{2\sigma_i^2}\right] \quad (10)$$

where σ_i is the width of the i^{th} RBF unit. The j^{th} output $y_j(P)$ of an RBF neural network is

$$y_j(P) = \sum_{i=1}^u R_i(P) \times w(j, i) \quad (11)$$

where $R_0 = 1$, $w(j, i)$ is the weight or strength of the i^{th} receptive field to the j^{th} output and $w(j, 0)$ is the bias of the j^{th} output.

We can see from (10) and (11) that the outputs of an RBF neural classifier are characterized by a linear discriminant function. They generate linear decision boundaries (hyperplanes) in the output space. Consequently, the performance of an RBF neural classifier strongly depends on the separability of classes in the u -dimensional space generated by the nonlinear transformation carried out by the u RBF units.

Geometrically, the key idea of an RBF neural network is to partition the input space into a number of subspaces which are in the form of hyperspheres.

3.2 Fuzzy RBF Neural Network

Based on the RBF neural network above, we

construct the environment parameters of network to give the fuzzy inference ability to it, fuzzy characteristics improved the learning generalization ability of neural networks and made a better approximation to the actual models.

We assume that there are fuzzy rules as:

$$R^i: \quad \text{If } x_1 = A_1^i, x_2 = A_2^i, \dots, x_n = A_n^i, \quad \text{then } y_1^i = \omega_{i1}, y_2^i = \omega_{i2}, \dots, y_o^i = \omega_{io}.$$

Where x_i is input vector, y is output vector; $u_{A_j^i}(x_i)$ is the membership function (Gaussian function) of x_i for A_j^i , $i = 1, 2, \dots, n, j = 1, 2, \dots, m$; ω_{ik} corresponds to k^{th} output of i^{th} rule, there are m rules totally and o output vectors.

The output of fuzzy system can be expressed as

$$y_k = \frac{\sum_{i=1}^m \omega_{ik} \pi_i}{\sum_{i=1}^m \pi_i} \quad (12)$$

Where π_i indicates the incentive intensity of i^{th} rule, namely

$$\pi_i = \mu_{1i} \mu_{2i} \dots \mu_{ki} = \prod_{j=1}^k \mu_{ji}, \quad 1 \leq i \leq m \quad (13)$$

Due to the function equivalence between RBF neural networks and fuzzy inference system, and these two systems can be unified by functions. We correspond the number of pattern clustering in RBF neural network to the number of fuzzy rules. Thus, the parameters are with fuzzy inference ability, which structure the fuzzy RBF neural network.

3.3 The Structure of Fuzzy RBF Neural Network

The structure of fuzzy RBF neural network we proposed in this paper consists of four layers: input layer, fuzzification layer, fuzzy inference layer and output layer. The topology of the fuzzy RBF neural network is shown as Fig. 3.

Suppose there are a k dimensional characteristic

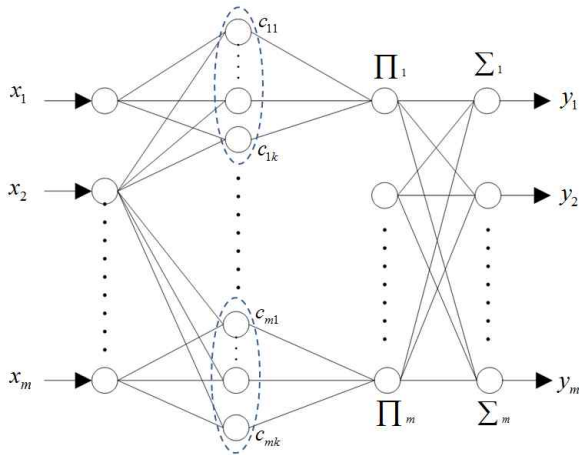


Fig. 3. The topology of fuzzy RBFNN.

space $F = \{f_1, f_2, \dots, f_k\}$ and m classes of patterns c_1, c_2, \dots, c_m , which follow normal distribution. N samples are selected as training set from total samples, we write them as training samples space: $X = \{X_1, X_2, \dots, X_N\}$. If the number of samples belong to i^{th} pattern c_i is n_i , then $N = \sum_{i=1}^m n_i$. Through statistical calculation to these training samples, the mean vector $\theta_i = (\theta_{i1}, \theta_{i2}, \dots, \theta_{ik})^k$ and the variance vector $\delta_i = (\delta_{i1}, \delta_{i2}, \dots, \delta_{ik})$ of each pattern can be obtained, where θ_{ij} and δ_{ij} denote the mean value and variance value of j^{th} feature in i^{th} pattern, respectively.

Based on the analysis above, all of the samples can be processed by fuzzification when they are inputted RBF neural network, multiple input-one output system will be transformed to a multiple input-multiple outputs fuzzy neural classifier.

The structure of fuzzy RBF neural network consists of four layers as follows:

The first layer (Input layer): each node denotes an input linguistic variable and they are transferred to the next layer directly.

The second layer (Fuzzification layer): in this layer, the Gaussian radial basis function is adopted as a fuzzy membership function of neuron. This layer compose by n neurons, which is divided into m groups, each group contains k neurons, then

$n = m \times k$. Thus, the input-output relationship of j^{th} neuron in i^{th} group is

$$y_{ij} = \frac{1}{\sqrt{2\pi}\delta_{ij}} \exp\left[-\frac{(x_i - \theta_{ij})^2}{2\sigma_{ij}^2}\right] \quad (14)$$

where y_{ij} denotes the probability density of j^{th} feature belong to the membership of pattern c_i , the output of i^{th} ($i = 1, 2, \dots, m$) group of input neuron: $(y_{i1}, y_{i2}, \dots, y_{ik})^T$ constitutes the membership vector of pattern c_i for input samples. The k -dimensional eigenvector from the input layer was translated into the membership of each feature to reach pattern by processing of fuzzification.

The third layer (Fuzzy inference layer): each node corresponds to a fuzzy rule and this layer implements the mapping from fuzzy rules to output space. In this paper, we defined the product of all input signals as the output of each node as follow.

$$y'_i = \prod_{j=1}^k w_j y_{ij}, \quad i = 1, 2, \dots, m \quad (15)$$

where w_i is the weight function of fuzzy rule, y_{ij} is the output in last layer.

The fourth layer (Output layer): this layer is a linear combination of outputs in last layer for defuzzification computation.

$$y = \frac{\sum_{i=1}^m \mu_i y'_i}{\sum_{i=1}^m \mu_i} \quad (16)$$

where $\mu_i = \mu_{1i} \mu_{2i} \dots \mu_{ki} = \prod_{j=1}^k \mu_{ji}, 1 \leq i \leq m$.

4. Learning Algorithm of Fuzzy RBF Neural Network

The learning process of fuzzy RBF neural network mainly based on updating the central value c_i and the radius δ_i of basis function. Usually, the adjustment algorithm of c_i and δ_i takes K-NN clustering algorithm under unsupervised learning. The weight w_i mostly used BP algorithm for ad-

justing, which is a supervised learning algorithm. Though we can apply the gradient paradigm to find the entire set of optimal parameters, the paradigm is generally slow and likely to become trapped in local minima. For solving this problem, Levenberg-Marquart (L-M) training algorithm was used to adjust w_i in this paper. L-M algorithm is an improved training algorithm of BP, which combined the advantages of gradient paradigm and newton method, it has both local performance and entire performance. The L-M training process can be illustrated as follow.

Assume that there are N samples $\{p_1, p_2, \dots, p_N\}$, the expected output of network are d_1, d_2, \dots, d_N and the actual output are y_1, y_2, \dots, y_N . When input the i^{th} sample, the output $y_{ij}(j = 1, 2, \dots, m)$ can be received, the error is the sum of each output error, that can be expressed by

$$E_i(x) = \frac{1}{2} e_i^2(x) = \frac{1}{2} \sum_{j=1}^m (d_{ij} - y_{ij})^2 \tag{17}$$

where $e(x)$ is error, its gradient $\nabla E(x)$ and Hessian matrix $\nabla^2 E(x)$ are

$$\nabla E(x) = J^T(x)e(x) \tag{18}$$

$$\nabla^2 E(x) = J^T(x)e(x) + S(x) \tag{19}$$

where

$$S(x) = \sum_{i=1}^N e_i(x) \nabla^2 e_i(x) \tag{20}$$

J is Jacobian Matrix, as

$$J(x) = \begin{bmatrix} \frac{\partial e_1(x)}{\partial x_1} & \frac{\partial e_1(x)}{\partial x_2} & \dots & \frac{\partial e_1(x)}{\partial x_n} \\ \frac{\partial e_2(x)}{\partial x_1} & \frac{\partial e_2(x)}{\partial x_2} & \dots & \frac{\partial e_2(x)}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial e_n(x)}{\partial x_1} & \frac{\partial e_n(x)}{\partial x_2} & \dots & \frac{\partial e_n(x)}{\partial x_n} \end{bmatrix} \tag{21}$$

Assume that $x^{(k)}, x^{(k+1)}$ denote the vector consists of weight and threshold by k^{th} and $(k+1)^{th}$ iteration. Thus

$$x^{(k+1)} = x^{(k)} + \Delta x \tag{22}$$

For L-M training algorithm,

$$\Delta x = -[J^T(x)J(x) + \mu I]^{-1} J(x)e(x) \tag{23}$$

where proportion coefficient $\mu (\mu > 0)$ is constant, I denotes unit matrix. When $\mu = 0$, L-M algorithm equals to Gaussian-Newton method; when μ is very large, it is approximate to gradient paradigm. In practical application, μ is a tentative parameter, it should be adjusted based on Δx .

The process of using L-M algorithm for training fuzzy RBF neural network can be described as below.

- (1) Normalizing the training samples;
- (2) Setting up predetermined training error ε , β , μ_0 and initialize weight and threshold vector, let $k = 0, \mu = \mu_0$;
- (3) Calculating the output of network and error function $E(x^{(k)})$;
- (4) Calculating Jacobian matrix $J(x)$ using E. q. (21);
- (5) Calculating Δx using E. q. (23);
- (6) If $E(x^{(k)}) < \varepsilon$, jump to step (8);
- (7) Calculating $E(x^{(k+1)})$ based on weight and threshold $x^{(k+1)}$, if $E(x^{(k+1)}) < E(x^{(k)})$, then weight and threshold will be updated, namely, let $x^{(k)} = x^{(k+1)}$ and $\mu = \frac{\mu}{\beta}$, return to step (3); otherwise, keep the weight and threshold, let $\mu = \mu \times \beta$ and return to step (5);
- (8) Stop.

The error function was defined as

$$E_p = \frac{1}{2} \sum_{k=1}^m (y_k^* - \bar{y}_k)^2, k = 1, 2, \dots, m. \tag{24}$$

where E_p also be called as learning objective function, in another words, the objective of training is to minimize E_p by adjusting the parameters; y_k^* means the actual output of k^{th} unit in output layer; \bar{y}_k denotes the expected output of k^{th} unit in output layer.

5. Experimental Results

The proposed face recognition system is running

on the hardware environment of Intel(R) Core(TM) 2 (2.93GHz) and the software environment of Windows 7 and Matlab R2009a.

The experiment uses the ORL Database of faces. Their Database of Faces, formerly "The ORL Database of Faces", contains a set of face images taken between April 1992 and April 1994 at the lab. There are ten different images of each of 40 distinct subjects. For some subjects, the images were taken at different times, varying the lighting, facial expressions (open/closed eyes, smiling/not smiling) and facial details (glasses/no glasses). All the images were taken against a dark homogeneous background with the subjects in an upright, frontal position, with tilting and rotation tolerance up to 20 degree, and tolerance of up to about 10% scaly. The files are in BMP format. The size of each image is 92*112 pixels, with 256 grey levels per pixel. Fig. 4 shows part samples in ORL face database.

In our experiments, we select 20 persons and 5 images each person in random. The selected 100 images are implemented as the set of training. Meanwhile, we select the other 100 images from 20 persons for testing. The original image with size of 92*112 can be translated into a 10304*1 matrix by PCA algorithm, and we select first 60 eigenfaces for experiments. Then, the resulting features are further projected into the Fisher's optimal subspace in which the ratio of the between-class scatter and the within-class scatter is maximized.



Fig. 4. Samples of ORL face database.

So, the number of input units to fuzzy RBF neural network is 60, namely $k=60$; the number of classes is 20, $m=20$; thus, the number of nodes in fuzzification layer is $60 \times 20 = 1200$, which includes 20 groups and each group contains 60 neurons. The number of fuzzy inference layer is $20 \times 5 = 100$ (select 20 persons and 5 images each person).

The expected output i^{th} output-node's value is 1, the others are 0, but the actual outputs are around the expected value range. Based on competitive choice rule, the category of the input samples is determined by the maximum value of actual output in fuzzy RBF neural network's output layer. If there are not only one maximum value, the network will refuse to make a judgment. The experimental results are shown as Table 1, Table 2 and Table 3.

From Table 1 and Table 2 we can find that: There is no "Reject" and with low recognition error rate, the learning ability of proposed method is strong; When the number of learning times is small, the recognition rate could be improved by increasing learning coefficient; When the number of learning times is beyond a certain range (very large), the recognition rate will not be enhanced, but tends to stabilize.

Table 3. indicates that when the number of classes is low, fuzzy RBF neural network classifier has good performance. the recognition rate will re-

Table 1. Experimental results of fuzzy RBF neural network using BP learning algorithm

Test	1	2	3	4	5	6
Learning Coefficient	0.25	0.5	0.9	0.9	0.99	0.99
Learning Times (*1000)	1	1	1	5	10	50
Correct	85	87	89	92	96	96
Error	15	13	11	8	4	4
Reject	0	0	0	0	0	0
Recognition Rate	85%	87%	89%	92%	96%	96%

Table 2. Experimental results of fuzzy RBF neural network using L-M learning algorithm

Test	1	2	3	4	5	6
Learning Coefficient	0.25	0.5	0.9	0.9	0.99	0.99
Learning Times (*1000)	1	1	1	5	10	50
Correct	87	89	90	94	98	98
Error	13	11	10	6	2	2
Reject	0	0	0	0	0	0
Recognition Rate	87%	89%	90%	94%	98%	98%

Table 3. Performance of network training algorithm (BP and L-M)

m	10	15	20	25	30	35	40
BP	100	97	96	93	88	80	74
L-M	100	99	98	95	91	84	80

duce as the number increase. From Table 1 to Table 3, result data indicate that L-M training algorithm is better than BP algorithm, the performance of fuzzy RBF neural network trained by L-M is more stable and with faster convergence ability than the one trained by BP algorithm.

6. CONCLUSION

In this paper, a general design approach using a fuzzy RBF neural network classifier for face recognition to cope with small training sets of high-dimensional problem is presented. Firstly, face features are first extracted by PCA algorithm. Then, the resulting features are further projected into the Fisher's optimal subspace in which the ratio of the between-class scatter and the within-class scatter is maximized. As a widely applied algorithm in fuzzy RBF neural network, BP learning algorithm has the low rate of convergence, therefore, an improved learning algorithm based on L-M for fuzzy RBF neural network is introduced in this paper, which combined the Gradient Descent algorithm with the Gauss-Newton algorithm, it

has both local performance and entire performance.

The experimental results shown that the proposed algorithm works well on ORL face database with different expressions, poses and illumination conditions. So, this algorithm has good capability of generation, and it can effectively reduce the dimensional of classification. Meanwhile, this algorithm can also reduce the computational complexity. The feature vectors are only extracted from gray-scale images, more features extracted from both gray-scale and spatial texture information and a real-time face recognition system will be studied in the future work.

REFERENCE

- [1] R. Chellappa, C.L. Wilson, and S. Sirohey, "Human and Machine Recognition of Faces: A Survey," *Proc. the IEEE*, Vol. 83, No. 5, pp. 705-740, 1995.
- [2] M. Turk and A. Pentland, "Eigenfaces for Recognition," *Journal of Cognitive Neuroscience*, Vol. 3, No. 1, pp. 71-86, 1991.
- [3] Tat-Jun Chin and D. Suter, *A Study of the Eigenface Approach for Face Recognition*, Technical Report of Monash University, Dept. of Elect & Computer Systems Engineering, pp. 1-18, 2004.
- [4] D. Blackburn, M. Bone, and P. Phillips, *Face Recognition Vendor Test 2000: Evaluation Report*, National Institute of Science and Technology, 2000.
- [5] P.J. Phillips, H. Moon, S. Rizvi, and P. Rauss, "FERET Evaluation Methodology for Face Recognition Algorithms," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, Vol. 22, Issue 10, pp. 1090-1103, 2000.
- [6] D. Bryliuk and V. Starovoitov, "Access Control by Face Recognition using Neural Networks and Negative Examples," *Proc. 2nd International Conference on Artificial Intelligence*, pp. 428-436, 2002.

- [7] S.A. Nazeer, N. Omar, and M. Khalid, "Face Recognition System using Artificial Neural Networks Approach," *Proc. IEEE International Conference on Signal Processing, Communications and Networking*, pp. 420-425, 2007.
- [8] Huang, X. Shao, and H. Wechsler, *Face Pose Discrimination Using Support Vector Machines*, Technical report of George Mason University and University of Minnesota, Minneapolis Minnesota, Vol. 1, pp. 154-156, 1998.
- [9] R. Brunelli and T. Poggio, "Face Recognition: Features versus Templates," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 15, No.10, pp. 1042-1052, 1993.
- [10] W. Chen, T. Sun, X. Yang, and L. Wang, "Face Detection based on Half Face Template," *Proc. the IEEE Conference on Electronic Measurement and Instrumentation*, pp. 54-58, 2009.
- [11] Dai Yang-chun and Xie Fang, "Face Recognition Based on the Fuzzy RBF Neural Network," *Techniques of Automation and Applications*, Vol. 25, No. 6, pp. 112-119, 2006.
- [12] Wenkai Xu and Eung-Joo Lee, "A Combinational Algorithm for Multi Face Recognition," *International Journal of Advancements in Computing Technology*, Vol. 4, No. 13, pp. 146-154, 2012.
- [13] A. Esposito, M. Marinaro, D. Oricchio, and S. Scarpetta, "Approximation of Continuous and Discontinuous Mappings by a Growing Neural RBF-based Algorithm," *Neural Networks*, Vol. 12, No. 1, pp. 651 - 665, 2000.
- [14] E.D. Virginia, "Biometric Identification System using a Radial Basis Network," *Proc. IEEE International Conference of Security Technology*, pp. 47 - 51, 2000.
- [15] M. Kirby and L. Sirovich, "Application of the Karhunen-Loeve Procedure for The Characterization of Human Faces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 12, No. 1, pp.103-108, 1990.
- [16] J. Lu, N. Plataniotis Kostantinos, and N. Venetsanopoulos Anastasios, "Face Recognition using LDA-based Algorithm," *IEEE Transactions on Neural Networks*, Vol. 14, No. 1, pp. 195-200, 2003.
- [17] A.M. Martinez and A.C. Kak., "PCA versus LDA," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 23, No. 2, pp. 228- 233, 2001.
- [18] Z. Liang and P.F. Shi, "Uncorrelated Discriminant Vectors using a Kernel Method," *Pattern Recognition*, Vol. 38, No. 2, pp. 307-310, 2005.
- [19] N. Peter Belhumeur, P. Joao Hespanha, and David J. Kriegman, "Eigenfaces vs. Fisherfaces: Using Class Specific Linear Projection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 14, No. 2, pp. 239-256, 1997.
- [20] B. Marian, Stewart, R. Movellan Javier, and J. Sejnowski Terrence, "Face Recognition by Independent Component Analysis," *IEEE Transactions on Neural Networks*, Vol. 13, No. 6, pp.1450-1464, 2002.
- [21] Meng Joo Er, Shiqian Wu, Juwei Lu, and Hock Lye Toh, "Face Recognition with Radial Basis Function Neural Networks", *IEEE Transactions on Neural Networks*, Vol. 13, No. 3, pp. 697-710, 2002.
- [22] Wenkai Xu and Eung-Joo Lee, "Dynamic Human Activity Recognition Based on Improved FNN Model," *Journal of Korea Multimedia Society*, Vol. 15, No. 4, pp. 417-424, 2012.



Eung-Joo Lee

received his B. S. , M. S. and Ph. D. in Electronic Engineering from Kyungpook National University, Korea, in 1990, 1992, and Aug. 1996, respectively. Since 1997 he has been with the Department of Information &

Communications Engineering, Tongmyong University, Korea, where he is currently a professor. From 2000 to July 2002, he was a president of Digital Net Bank Inc.. From 2005 to July 2006, he was a visiting professor in the Department of Computer and Information Engineering, Dalian Polytechnic University, China. His main research interests includes biometrics, image processing, and computer vision.