

# Face Recognition using Radial Basis Function Network based on LDA

Byung-Joo Oh

**Abstract**—This paper describes a method to improve the robustness of a face recognition system based on the combination of two compensating classifiers. The face images are preprocessed by the appearance-based statistical approaches such as Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA). LDA features of the face image are taken as the input of the Radial Basis Function Network (RBFN). The proposed approach has been tested on the ORL database. The experimental results show that the LDA+RBFN algorithm has achieved a recognition rate of 93.5%.

**Keywords**—Face recognition, linear discriminant analysis, radial basis function network.

## I. INTRODUCTION

FACE recognition has drawn considerable interest and attention from many researchers in the pattern recognition field for the last two decades. The recognition of faces is very important because of its potential commercial applications, such as in the area of video surveillance, access control systems, retrieval of an identity from a data base for criminal investigations and user authentication.

One of typical procedures can be described for video-surveillance applications. A system that automatically recognizes a face in a video stream first detects the location of face and normalizes it with respect to the pose, lighting and scale. Then, the system tries to extract some pertinent features and to associate the face to one or more faces stored in its database, and gives the set of faces that are considered as nearest to the detected face. Usually, each of these stages for detection, normalization, feature extraction and recognition is so complex that it must be studied separately [1].

Although there are a number of face recognition algorithms which work well in constrained environments, face recognition is still an open and very challenging problem in real applications. Many problems arise because of the variability of many parameters: face expression, pose, scale, lighting, and other environmental parameters [2], [3].

Among face recognition algorithms, appearance-based approaches have been successfully developed and tested as a reference. These approaches utilize the pixel intensity or intensity-derived features. Because of these characteristics,

these methods may not perform well when the test face data is significantly different from the training face data, due to variations in pose, illumination and expression. While a robust classifier could be designed to handle any one of these variations, it is extremely difficult for an approach to deal with all of these variations. Each individual classifier has different performance to different variations in the environmental parameters. This situation suggests that different classifiers contribute complementary information to the classification task [4]. Therefore, it is expected that combined classifiers which integrate different information sources and various face features are likely to improve the overall system performance.

A face appearance in a computer is considered as a map of pixels of different gray levels. Thus in order to recognize an individual using face image it has to represent a human face in an intelligent way, that is, to represent a face image as a feature vector of reasonably low dimension and high discriminating power. Developing this representation is one of main challenges for this face recognition problem [5].

In face recognition, the 2-dimensional face image is considered as a vector, by concatenating each row or column of the image. That is, we usually represent an image of size  $p \times q$  pixels by a vector in a  $p \cdot q$  dimensional space. In practice, however, these  $(p \cdot q)$ -dimensional spaces are too large to allow robust and fast object recognition. A common way to attempt to resolve this problem is to use dimensionality reduction techniques [6]. Two of the most popular techniques for this purpose are: Principal Components Analysis (PCA) and Linear Discriminant Analysis (LDA).

In PCA and LDA approaches, each classifier has its own representation of the basis vectors of a high dimensional face vector space. By projecting the face vector to the basis vectors, the projection coefficients are used as the feature representation of each face images [4]-[7]. This feature representation vectors are then used to train the weighting factors in the combined neural networks.

Recently neural networks have been employed and compared to conventional classifiers for a number of classification problems. The neural network method is capable of rapid classification. A feedforward multi-layer neural networks (MLNN) is used as a classifier instead of the classical mean square error classifier.

A Radial Basis Function (RBF) network is a two layer network that has different types of neurons in the hidden layer and the output layer. The RBF network performs similar

Manuscript received July 9, 2005. This work was supported in part by the Korea Ministry of Commerce, Industry and Energy under Grant No. R12-2003-004-03001-0.

Byung-Joo Oh is with the Hannam University, Daejeon, Korea (phone: 82-42-629-7397; fax: 82-42-629-7397; e-mail: bjoh@hannam.ac.kr).

function mapping with the MLNN, however its structure and function is much different. A RBF is a local network that is trained in a supervised manner. This contrasts with a MLNN that is a global network. The distinction between local and global is made through the extent of input surface covered by the function approximation. An MLNN performs a global mapping, meaning all inputs cause an output, while an RBF performs a local mapping, meaning only inputs near a receptive field produce an activation [8]-[11].

In this work, we propose an algorithm for face recognition based on the combination of multiple classifiers for improving the performance of the best individual one. The face recognition system that we propose consists of two main stages: First, the feature extraction and dimension reduction techniques such as PCA or LDA are applied to the input face image. Then classification techniques such as RBFN are applied to the produced feature vectors.

This paper is structured as follows. In section 2 we briefly describe the PCA and LDA as a preprocessing technique. In section 3 we describe neural networks as classifiers. In section 4 we present experimental results and in section 5 we draw some preliminary conclusions and we point out further investigations.

## II. PREPROCESSING WITH PCA AND LDA

### A. PCA Processing

Principal Components Analysis (PCA) technique is used for dimensionality reduction to find the vectors which best account for the distribution of face images within the entire image space. The basic approach is to compute the eigenvectors of the covariance matrix of the training data, and approximate the original data by a linear combination of the leading eigenvectors. These vectors define the subspace of face images and the subspace is called eigenface space. All faces in the training set are projected onto the eigenface space to find a set of weights that describes the contribution of each vector in the eigenface space.

The key procedure in PCA is based on Karhunen-Loeve transformation. It is an orthogonal linear transform of the signal that concentrates the maximum information of the signal with the maximum number of parameters using the minimum square error.

By using PCA procedure, the test image can be identified by first, projecting the image onto the eigenface space to obtain the corresponding set of weights, and then comparing with the set of weights of the faces in the training set. The distance measure used in the matching could be a simple Euclidean, or a weighted Euclidean distance. The problem of low-dimensional feature representation can be stated as follows [1], [4]:

Let  $X = (x_1, x_2, \dots, x_i, \dots, x_N)$  represent the  $n \times N$  data matrix, where each  $x_i$  is a face vector of dimension  $n$ , concatenated from a  $p \times q$  face image. Here  $n (= p \cdot q)$  represents the total number of pixels in the face image and  $N$

is the number of face images in the training set. The PCA can be considered as a linear transformation (1) from the original image vector to a projection feature vector, i.e.,

$$Y = W^T X \quad (1)$$

where  $Y$  is the  $m \times N$  feature vector matrix,  $m$  is the dimension of the feature vector, and transformation matrix  $W$  is an  $n \times m$  transformation matrix whose columns are the eigenvectors corresponding to the  $m$  largest eigenvalues computed according to the formula:

$$\lambda e_i = S e_i. \quad (2)$$

Here the total scatter matrix  $S$  and mean image of all samples are defined as

$$S = \sum_{i=1}^N (x_i - \mu)(x_i - \mu)^T, \quad \mu = \frac{1}{N} \sum_{i=1}^N x_i. \quad (3)$$

After applying the linear transformation  $W^T$ , the scatter of the transformed feature vectors  $\{y_1, y_2, \dots, y_N\}$  is  $W^T S W$ . In PCA, the projection  $W_{opt}$  is chosen to maximize the determinant of the total scatter matrix of the projected samples, i.e.,

$$W_{opt} = \arg \max_W |W^T S W| \\ = [w_1 w_2 \dots w_m] \quad (4)$$

where  $\{w_i | i = 1, 2, \dots, m\}$  is the set of  $n$ -dimensional eigenvectors of  $S$  corresponding to the  $m$  largest eigenvalues. In other words, the input vector (face) in an  $n$ -dimensional space is reduced to a feature vector in an  $m$ -dimensional subspace. We can see that the dimension of the reduced feature vector  $m$  is much less than the dimension of the input face vector  $n$ .

Some authors [8] presented that a drawback of this approach is that the scatter being maximized is due not only to the between-class scatter that is useful for classification, but also to the within-class scatter that is due to unwanted illumination changes. Thus if PCA is presented with images of faces under varying illumination, the projection matrix  $W_{opt}$  will contain principal components which retain the variation due lighting in the projected feature space.

### B. LDA Processing

The goal of the Linear Discriminant Analysis (LDA) is to find an efficient way to represent the face vector space. PCA constructs the face space using the whole face training data as a whole, and not using the face class information. On the other hand, LDA uses class specific information which best discriminates among classes. LDA produces an optimal linear discriminant function which maps the input into the classification space in which the class identification of this sample is decided based on some metric such as Euclidean distance. LDA takes into account the different variables of an object and works out which group the object most likely belongs to.

Exploiting the class information can be helpful to the identification tasks, whether or not two or more groups are significantly different from each other with respect to the mean of particular variables. By defining different classes with different statistics, the images in the training set are divided into the corresponding classes[3]-[8].

The principle of the LDA algorithm is described in many papers[1-8], however their descriptions are so much similar in their algorithms. The algorithm employed here is mainly based on [8].

Let a training set of  $N$  face images represent  $c$  different subjects. The face images in the training set are two-dimensional intensity arrays, represented as a vectors of dimension  $n$ . Different instances of the same person are defined to belong to the same class while faces of different subjects should belong to different classes. LDA selects  $W$  in (5) in such a way that the ratio of the between-class scatter and the within-class scatter is maximized:

$$z_i = W^T y_i. \quad (5)$$

Assuming that  $S_w$  is non-singular, the basis vectors in  $W$  correspond to the first  $l$  eigenvectors with the largest eigenvalues of  $S_w^{-1}S_B$ . Here  $S_B$  is the between-class scatter matrix and  $S_w$  is the within-class scatter matrix, of the training image set, defined as

$$S_w = \sum_{i=1}^c \sum_{y_k \in Y_i} (y_k - \mu_i)(y_k - \mu_i)^T, \quad \mu_i = \frac{1}{N_i} \sum_{k=1}^{N_i} y_k \quad (6)$$

$$S_B = \sum_{i=1}^c N_i (\mu_i - \mu)(\mu_i - \mu)^T. \quad (7)$$

In the above expression,  $N_i$  is the number of training samples in class  $Y_i$ ,  $c$  is the number of distinct classes,  $\mu_i$  is the mean vector of samples belonging to class  $Y_i$ ,  $y_k$  represents the samples belonging to class  $Y_i$ . The optimal projection  $W_{opt}$  is chosen as,

$$W_{opt} = \arg \max_w \frac{W^T S_B W}{W^T S_w W} = [w_1 w_2 \dots w_m] \quad (8)$$

where  $\{w_i | i = 1, 2, \dots, l\}$  is the set of generalized eigenvector of  $S_B$  and  $S_w$  corresponding to the largest generalized eigenvalues  $\{\lambda_i | i = 1, 2, \dots, l\}$ , i.e.,

$$S_B w_i = \lambda_i S_w w_i, \quad i = 1, 2, \dots, l \quad (9)$$

If we assume that the number of classes is  $c$ , then there are at most  $c - 1$  nonzero generalized eigenvalues, and so an upper bound on  $l$  is  $c - 1$ . The  $l$ -dimensional representation is then obtained by projecting the original face images onto the subspace spanned by the  $l$  eigenvectors. The representation  $z_i$  should enhance the separability of the different face objects

under consideration.

In most practical face recognition problem, the within-class scatter matrix  $S_w$  is singular because of the rank of  $S_w$  is at most  $N - c$ . In practical problems,  $N$ , the number of images in the training set is much smaller than  $n$ , the number of pixels in each image. This problem can be avoided by projecting the image set to a lower dimensional space so that the resulting  $S_w$  is nonsingular. This is first achieved by using PCA to reduce the dimension of the feature space to  $N - c$ , and then applying the standard LDA to reduce the dimension to  $c - 1$  [6],[8].

LDA transformation is strongly dependent on the number of classes, the number of samples, and the original space dimensionality.

### III. RADIAL BASIS FUNCTION NETWORKS

The Radial Basis Function (RBF) networks performs similar function mapping with the multi-layer neural network, however its structure and function are much different. A RBF is a local network that is trained in a supervised manner. This contrast with a MLNN is a global network. The distinction between local and global is made through the extent of input surface covered by the function approximation. RBF performs a local mapping, meaning only inputs near a receptive field produce an activation [10],[11].

Note that some of the symbols used in this section may not be identical to those used in the above sections.

A typical RBF neural network structure is shown in Fig. 1.

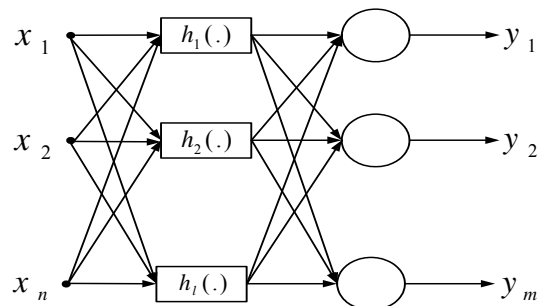


Fig. 1 RBF network structure

The input layer of this network is a set of  $n$  units, which accept the elements of an  $n$ -dimensional input feature vector.  $n$  elements of the input vector  $x_n$  is input to the  $l$  hidden function, the output of the hidden function, which is multiplied by the weighting factor  $w_{ij}$ , is input to the output layer of the network  $y_j(x)$ .

For each RBF unit  $k$ ,  $k = 1, 2, 3, \dots, l$ , the center is selected as the mean value of the sample patterns belong to class  $k$ , i.e.,

$$\mu_k = \frac{1}{N_k} \sum_{i=1}^{N_k} x_k^i, \quad k = 1, 2, 3, \dots, m \quad (10)$$

where  $x_k^i$  is the eigenvector of the  $i$ th image in the class  $k$ ,

and  $N_k$  is the total number of trained images in class  $k$ . For any class  $k$ , the Euclidean distance  $d_k$  from the mean value  $\mu_k$  to the farthest sample pattern  $x_k^f$  belong to class  $k$ :

$$d_k^f = \left\| x_k^f - \mu_k \right\|, \quad k = 1, 2, \dots, m \quad (11)$$

Only the neurons in the bounded distance of  $d_k$  in RBF are activated, and from them the optimized output is found

Since the RBF neural network is a class of neural networks, the activation function of the hidden units is determined by the distance between the input vector and a prototype vector. Typically the activation function of the RBF units(hidden layer unit) is chosen as a Gaussian function with mean vector  $\mu_i$  and variance vector  $\sigma_i$  as follows[10],[11]:

$$h_i(x) = \exp \left[ -\frac{\|x - \mu_i\|^2}{\sigma_i^2} \right], \quad i = 1, 2, \dots, l \quad (12)$$

where  $\|\cdot\|$  indicates the Euclidean norm on the input space. Note that  $x$  is an  $n$ -dimensional input feature vector,  $\mu_i$  is an  $n$ -dimensional vector called the center of the RBF unit,  $\sigma_i$  is the width of the  $i$ th RBF unit and  $l$  is the number of the RBF units. The response of the  $j$ th output unit for input  $x$  is given as:

$$y_j(x) = \sum_{i=1}^l h_i(x)w_{ij} \quad (13)$$

where  $w_{ij}$  is the connection weight of the  $i$ -th RBF unit to the  $j$ -th output node.

For designing a RBF neural network classifier, the number of the input data  $x_i$  is equal to the number of the feature vector elements, which is produced by the LDA process, and the number of the output is equal to the number of the class number in the training face image database.

The learning algorithm for radial basis function network is implemented in two phases. During the first phase of learning, the numbers of radial basis functions and their mean and standard deviation values are obtained. During the second phase of learning, weights connecting layers L1 and L2 are determined via a gradient descent or a least-square method.

The learning algorithm using the gradient descent method is as follows[13],[14]:

1. Initialize  $w_{ij}$  with small random values, and define the radial basis function  $h_j$ , using the mean  $\mu_j$ , and standard deviation  $\sigma_j$ .
2. Obtain the output vectors  $h$  and  $y$  using (12) and (13).
3. Calculate  $\Delta w_{ij} = \alpha(t_i - y_i)h_j$ .

where  $\alpha$  is a constant,  $t_i$  is the target output, and  $o_i$  is the actual output.

4. Update the weights  $w_{ij}(k+1) = w_{ij}(k) + \Delta w_{ij}$ .

5. Calculate the mean square error  $E = \frac{1}{2} \sum_{i=1}^m (t_i - y_i)^2$

6. Repeat steps 2-5 until  $E \leq E_{\min}$ .

7. Repeat steps 2-6 for all training samples.

The output layer is a layer of standard linear neurons and performs a linear transformation of the hidden node outputs. This layer is equivalent to a linear output layer in a MLNN, but the weights are usually solved for using a gradient descent algorithm. The output layer may, or may not, contain biases; the examples in this supplement do not use biases.

Receptive fields center on areas of the input space where input vectors lie, and serve to cluster similar input vectors. If an input vector ( $x$ ) lies near the center of a receptive field ( $\mu$ ), then that hidden node will be activated. If an input vector lies between two receptive field centers, but inside the receptive field width ( $\sigma$ ) then the hidden nodes will both be partially activated. When input vectors that lie far from all receptive fields there is no hidden layer activation and the RBF output is equal to the output layer bias values [15].

A RBF is a local network that is trained in a supervised manner. This contrasts with a MLNN network that is a global network. The distinction between local and global is the made though the extent of input surface covered by the function approximation. An MLNN performs a global mapping, meaning all inputs cause an output, while an RBF performs a local mapping, meaning only inputs near a receptive field produce an activation.

The following Fig. 2 shows the flow of LDA+RBF algorithm.

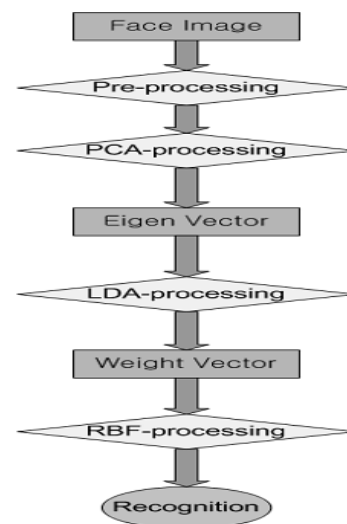


Fig. 2 The flow of LDA+RBF algorithm

#### IV. EXPERIMENTS

Our experiments were performed on the ORL database, which contains 400 face images of 40 individuals. In our

experiments, a total of 200 images were randomly selected as the training set. That is, since each class has 10 images, five of them are selected as the training images, and the other five as the testing images. The methods used are LDA, and RBF with LDA. The original images are preprocessed using histogram equalization before they are applied to the PCA process.

The table 1 shows the performance of two different algorithms applied to the face recognition. The recognition rate for the LDA+MLNN shows a better recognition rate. However this does not mean that this method is always better than that of LDA only.

TABLE I  
 RECOGNITION RATE OF THE LDA+MLNN AND LDA+RBF

Method	LDA	LDA+RBF
Error(#of images)	13	11
Unrecognizable images	•	•
Recognition rate (Recog./Try)	92.35% (157/170)	93.53% (159/170)

## V. CONCLUSION

This paper presented the results of the performance test of face recognition algorithms on the ORL database. The well known PCA algorithm is first tried as a preprocessing step. The next processing step was by LDA algorithm. This LDA was known as a more robust algorithm for the illumination variance. Finally LDA+RBF algorithm has been tested. The performance was by LDA+RBF resulted in 93.5% recognition rate. The introduction of RBF enhances the classification performance and provides relatively robust performance for the variation of light. To compare the performance of the LDA and LDA+RBF, more experiments on the various databases should be performed on diverse environments.

## ACKNOWLEDGMENT

And the author wishes to thank G. Yang for providing simulation results and preparing plots used in this paper.

## REFERENCES

- [1] G. L. Marcialis and F. Roli, "Fusion of appearance-based face recognition algorithms," *Pattern Analysis & Applications*, V.7. No. 2, Springer-Verlag London (2004) 151-163
- [2] W. Zhao, R. Chellappa, A. Rosenfeld, P. J. Phillips, "Face recognition: A literature survey," UMD CfAR Technical Report CAR-TR-948, (2000)
- [3] X. Lu, "Image analysis for face recognition," available: [http://www.cse.msu.edu/~lvxiaogu/publications/ImAna4FacRcg\\_Lu.pdf](http://www.cse.msu.edu/~lvxiaogu/publications/ImAna4FacRcg_Lu.pdf) (2003).
- [4] X. Lu, Y. Wang, and A. K. Jain, "Combining classifier for face recognition," *Proc. of IEEE 2003 Intern. Conf. on Multimedia and Expo*. V.3. (2003) 13-16.
- [5] V. Espinosa-Duro, M. Faundez-Zanuy, "Face identification by means of a neural net classifier," *Proc. of IEEE 33<sup>rd</sup> Annual 1999 International Carnahan Conf. on Security Technology*, (1999) 182-186
- [6] A. M. Martinez and A. C. Kak, "PCA versus LDA," *IEEE Trans. On pattern Analysis and Machine Intelligence*, Vol. 23, No. 2, (2001) 228-233.
- [7] W. Zhao, A. Krishnaswamy, R. Chellappa, "Discriminant analysis of principle components for face recognition," *3<sup>rd</sup> Intern. Conf. on Face & amp; Gesture Recognition*, pp. 336-341, April 14-16, (1998), Nara, Japan.

- [8] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. Fishfaces: Recognition using class specific linear projection," *IEEE Trans. On Pattern Analysis and Machine Intelligence*, Vol. 19, No. 7 (1997) 711-720
- [9] S. Lawrence, C. L. Giles, A. C. Tsoi, and A. D. Back, "Face recognition: A convolutional neural-network approach," *IEEE Trans. On Neural Networks*, Vol. 8, No. 1 (1997) 98-112
- [10] M. J. Er, S. Wu, J. Lu, and H. L. Toh, "Face recognition with radial basis function(RBF) neural networks," *IEEE Trans. On Neural Networks*, Vol. 13, No. 3 (2002) 697-710
- [11] J. Haddadnia, K. Faez, M. Ahmadi, "N-feature neural network human face recognition," *Proc. Of the 15<sup>th</sup> Intern. Conf. on Vision Interface*, Vo. 22, Issue 12, (2004) 1071-1082
- [12] Lu, Juwei, K.N. Plataniotis, and A.N. Venetsanopoulos, "Face recognition using LDA-based algorithms," *IEEE Transactions on Neural Networks*, Vol. 14, No. 1 pp. 195-200, 2003
- [13] A. D. Kulkarni, *Computer Vision and Fuzzy-Neural Systems*, Prentice-Hall, 2001.
- [14] D. A. Forsyth and J. Ponce, *Computer Vision: A Modern Approach*, Prentice-Hall, 2003, pp. 512-514.
- [15] J.W. Hines, *Fuzzy and Neural Approaches in Engineering*, John Wiley & Sons (1997).