A Face Recognition System using Neural Networks with Incremental Learning Ability

Youness Aliyari Ghassabeh, Hamid Abrishami Moghaddam

***Abstract*—In this paper, we present a new incremental face recognition (IFR) system based on new adaptive learning algorithms and networks. We introduce new adaptive linear discriminant analysis (LDA) algorithm and related network for optimal facial feature extraction and use them to construct a new IFR system. Convergence proof of all algorithms is given using an appropriate cost function and discussing about its initial conditions. Application of the new IFR on feature extraction from facial image sequences is given in two steps: i) image preprocessing, which include** s **normalization, histogram equalization, mean centering and background omission** removal**, ii) adaptive LDA feature estimation. In preprocessing stage, all input images are cropped and prepared for next step. Outputs of preprocessing stage are used as a sequence of inputs for IFR system. The proposed system was tested on YALE face database. Experimental results on this database demonstrated the effectiveness of the proposed system for adaptive estimation of feature space for online face recognition.**

I. Introduction

Choosing an appropriate set of features is critical when designing pattern classification systems under the framework of supervised learning. Ideally, it is desirable to use only features having high separability power while ignoring the rest. There has been an increased interest on deploying feature selection in applications such as face and gesture recognition [1]. Most effort in the literature has been focused mainly on developing feature extraction methods [2-5]. Feature extraction for face representation is one of the central issues in face recognition (FR) systems. FR is a high dimensional pattern recognition problem. Even low resolution images generate huge dimensional feature space. Among various solutions to the problem [6], the most successful seems to be those appearance-based approaches, which generally operate directly on images or appearances of face objects and process the image as a two-dimensional pattern. The main trend in feature extraction has been representing the data in a lower dimensional space computed through a linear or non-linear transformation satisfying certain properties. Statistical techniques have been widely used for face recognition and in facial analysis to extract the abstract features of the face patterns. Principal component analysis (PCA) [2] and linear discriminant analysis (LDA) [3] are two main techniques used for dimensionality reduction and feature extraction in the appearance-based approaches.

Typical implementation of these two techniques assumes that a complete dataset of training samples is given in advance, and learning is carried out in one batch. However, when we conduct PCA/LDA learning over datasets in real-world applications, we often confront difficult situations where a complete set of training samples is not given in advance. Even if a large amount of face images are available when constructing a FR system, all the variations that will happen in the future can not be considered in advance. Thus high recognition performance in practical situations can hardly be expected with only a static data set. A solution to this problem is to make face recognition systems learn continuously to adapt to incoming training samples. This can be done by embedding an adaptive learning ability into a face recognition system. For this purpose, different adaptive versions of PCA and LDA have been introduced by researchers and some of them are used to construct adaptive FR systems [7-10].

In this paper, we introduce a new adaptive learning algorithm for estimation of LDA feature during incremental training process. Convergence proof of the proposed algorithm is given by introducing an appropriate cost function and discussing about its properties and initial conditions. Based on the proposed learning algorithms, we present the related neural networks and construct our new incremental face recognition (IFR) system. Proposed networks and IFR system use a sequence of data for training stage. Therefore the need to keep a large amount of sample data for training phase is reduced. Memory size and complexity reduction provided by the new adaptive feature extraction networks make them appropriate for different real time pattern recognition applications. Simulation results approved ability of the proposed IFR system for effective face classification during online training process.

Organization of the paper is as follows: In the next section, we present our new adaptive algorithms for dimension reduction and feature extraction; furthermore, we prove convergence of the proposed algorithms by introducing the related cost function. In section III, we implement networks based on the introduced learning algorithms and then in section IV, we present our IFR system by cascading the $\mathbf{\Sigma}^{-1/2}$ network and adaptive PCA (APCA) network. Section V, is devoted to simulations and experimental results. Finally, concluding remarks are given in section VI.

Authors are with the Electrical Engineering Department, K. N. Toosi University of Technology, Tehran, Iran.
E-mails: y_aliyari@ee.kntu.ac.ir, moghadam@saba.kntu.ac.ir.

II. NEW ADAPTIVE LEARNING ALGORITHM

A. Linear Discriminant Analysis

Let $\{\mathbf{x_1}, \mathbf{x_2}, ..., \mathbf{x_N}\}$, $\mathbf{x} \in \Re^n$ be N samples belonged to L classes $\{\omega_1, \omega_2, ..., \omega_L\}$. Consider $\mathbf{m}$ and $\mathbf{\Sigma}$ denote the mean vector and covariance matrix of samples, respectively. LDA algorithm searches the directions for maximum discrimination of classes in addition to dimensionality reduction. To achieve this goal, within-class matrix is defined. A within-class scatter matrix is the scatter of the samples around their respective class means $\mathbf{m}_i$ and denoted by $\mathbf{\Sigma}_w$. The mixture scatter matrix is the covariance of all samples regardless of class assignments, and represented by $\mathbf{\Sigma}$. In LDA, the optimum linear transform is composed of $p(\leq n)$ eigenvectors of $\mathbf{\Sigma}_w^{-1}\mathbf{\Sigma}$ corresponding to its p largest eigenvalues [11]. The computation of the eigenvector matrix $\mathbf{\Phi}_{LDA}$ of $\mathbf{\Sigma}_w^{-1}\mathbf{\Sigma}$ is equivalent to the solution of the generalized eigenvalue problem $\mathbf{\Sigma\Phi}_{LDA} = \mathbf{\Sigma}_w\mathbf{\Phi}_{LDA}\mathbf{\Lambda}$, where $\mathbf{\Lambda}$ is the generalized eigenvalue matrix. Under assumption of positive definite matrix $\mathbf{\Sigma}_w$, if we consider: $\mathbf{\Psi} = \mathbf{\Sigma}_w^{1/2}\mathbf{\Phi}_{LDA}$, there exists a symmetric $\mathbf{\Sigma}_w^{-1/2}$ such that the problem can be reduced to following symmetric eigenvalue problem [11]:

$$\mathbf{\Sigma}_w^{-1/2}\mathbf{\Sigma}\mathbf{\Sigma}_w^{-1/2}\mathbf{\Psi} = \mathbf{\Psi\Lambda} \tag{1}$$

B. New $\mathbf{\Sigma}^{-1/2}$ Algorithm

Consider cost function $J(\mathbf{W})$ with parameter $\mathbf{W}$, $J: \Re^{n\times n} \to \Re$ as follows:

$$J(\mathbf{W}) = \frac{trace(\mathbf{W}^3\mathbf{\Sigma})}{3} - trace(\mathbf{W}) \tag{2}$$

Where $\mathbf{W}$ is a positive definite, symmetric $n \times n$ matrix, $\mathbf{\Sigma}$ is the covariance matrix associated to incoming random vectors (it is assumed that sample vectors have zero mean vector) and *trace(.)* function computes sum of diagonal elements of its input matrix. The cost function $J(\mathbf{W})$ is a continuous function with respect to $\mathbf{W}$. The first derivative of (2) is computed as follows [12]:

$$\frac{\partial E(J(\mathbf{W}))}{\partial \mathbf{W}} = (\mathbf{W}^2\mathbf{\Sigma} + \mathbf{W\Sigma}\,\mathbf{W} + \mathbf{\Sigma W}^2)/3 - \mathbf{I} \tag{3}$$

Where $\mathbf{I}$ is identity matrix. If $\mathbf{W}$ is selected such that $\mathbf{W\Sigma} = \mathbf{\Sigma W}$, equating (3) to zero will result in $\mathbf{W} = \mathbf{\Sigma}^{-1/2}$. Therefore, $\mathbf{\Sigma}^{-1/2}$ is a critical point (matrix) of (2). The Hessian matrix of equation (2) with respect to $\mathbf{W}$ is computed as follows [12]:

$$H(J(\mathbf{W})) = 2(\mathbf{I} \otimes \mathbf{\Sigma W}) + 2(\mathbf{\Sigma W} \otimes \mathbf{I}) + \mathbf{W} \otimes \mathbf{\Sigma} + \mathbf{\Sigma} \otimes \mathbf{W} \tag{4}$$

Where $\otimes$ in (4) is denoted the kronecker product of two matrix. Substituting $\mathbf{W} = \mathbf{\Sigma}^{-1/2}$ in (4) will result a positive definite matrix (summation of some positive definite matrix is again a positive definite matrix and kronecker product of two positive definite matrix is positive definite, therefore (5) is a positive definite matrix).

$$H(J(\mathbf{W}))|_{\mathbf{W}=\mathbf{\Sigma}^{-1/2}} = 2(\mathbf{I} \otimes \mathbf{\Sigma}^{1/2}) + 2(\mathbf{\Sigma}^{1/2} \otimes \mathbf{I}) + \mathbf{\Sigma}^{-1/2} \otimes \mathbf{\Sigma} + \mathbf{\Sigma} \otimes \mathbf{\Sigma}^{-1/2} \tag{5}$$

Where H denotes the Hessian matrix. The above analysis implies that if $\mathbf{W}$ is a symmetric matrix satisfying $\mathbf{W\Sigma} = \mathbf{\Sigma W}$, the cost function $J(\mathbf{W})$ will have a minimum that occurs at $\mathbf{W} = \mathbf{\Sigma}^{-1/2}$ [12]. Using the gradient descent optimization method [13, 14] we obtained the following adaptive equation for the computation of $\mathbf{\Sigma}^{-1/2}$:

$$\mathbf{W}_{k+1} = \mathbf{W}_k + \eta_k\left(-\frac{\partial J(\mathbf{W})}{\partial \mathbf{W}}\right) = \mathbf{W}_k + \eta_k(\mathbf{I} - (\mathbf{W}_k{}^2\mathbf{x}_{k+1}\mathbf{x}_{k+1}{}^t + \mathbf{x}_{k+1}\mathbf{x}_{k+1}{}^t\mathbf{W}_k{}^2 + \mathbf{W}_k\mathbf{x}_{k+1}\mathbf{x}_{k+1}{}^t\mathbf{W}_k)/3) \tag{6}$$

where $\mathbf{W}_{k+1}$ is the estimation of $\mathbf{\Sigma}^{-1/2}$ in *k+1-th* iteration, η_k is the step size where satisfies Ljung assumptions [15] and $\mathbf{x}_{k+1}$ is the input vector at iteration *k+1*. The only constraint on (6) is its initial conditions, that is $\mathbf{W}_0$ must be a symmetric and positive definite matrix satisfying $\mathbf{W_0\Sigma} = \mathbf{\Sigma W_0}$. It is quite easy to prove that if $\mathbf{W_0\Sigma} = \mathbf{\Sigma W_0}$, then we will obtain:

$$\begin{aligned} E(\mathbf{W}_{k+1}) &= \mathbf{W}_k + \eta_k(\mathbf{I} - \mathbf{W}_k^2\mathbf{\Sigma}) \\ &= \mathbf{W}_k + \eta_k(\mathbf{I} - \mathbf{W}_k\mathbf{\Sigma W}_k) = \mathbf{W}_k + \eta_k(\mathbf{I} - \mathbf{\Sigma W}_k^2) \end{aligned} \tag{7}$$

Therefore (6) is simplified to three more efficient forms as follows:

$$\mathbf{W}_{k+1} = \mathbf{W}_k + \eta_k(\mathbf{I} - \mathbf{W}_k^2\mathbf{x}_{k+1}\mathbf{x}_{k+1}^t) \tag{8}$$

$$\mathbf{W}_{k+1} = \mathbf{W}_k + \eta_k(\mathbf{I} - \mathbf{W}_k\mathbf{x}_{k+1}\mathbf{x}_{k+1}^t\mathbf{W}_k) \tag{9}$$

$$\mathbf{W}_{k+1} = \mathbf{W}_k + \eta_k(\mathbf{I} - \mathbf{x}_{k+1}\mathbf{x}_{k+1}^t\mathbf{W}_k^2) \tag{10}$$

Equations (8-10) have less computational cost with respect to (6). Obviously, the expected values of $\mathbf{W}_k$ as $k \to \infty$ in (6) and (8-10) are equal to $\mathbf{\Sigma}^{-1/2}$, provided that $\mathbf{W_0\Sigma} = \mathbf{\Sigma W_0}$.

C. Incremental Estimation of the Covariance Matrix

Let $\mathbf{W}$ be a symmetric positive definite matrix. Consider the cost function $J: \Re^{n\times n} \to \Re$ defined as follows:

$$J(\mathbf{W}) = \frac{J(\mathbf{W}^2)}{2} - tr(\mathbf{W\Sigma}) \quad (11)$$

Considering $E(\mathbf{x}) = 0$, the first derivative of the (11) with respect to $\mathbf{W}$ is computed as follows [12]:

$$\partial E(J(\mathbf{W}))/\partial(\mathbf{W}) = \mathbf{W} - \mathbf{\Sigma} \quad (12)$$

The only critical point (matrix) of the cost function (zeros of (12)) is equal to the covariance matrix. The Hessian matrix of the cost function in (11) with respect to $\mathbf{W}$ is a positive definite matrix computed as follows [12]:

$$H(J(\mathbf{W})) = \mathbf{I} \otimes \mathbf{I} \quad (13)$$

Therefore, $\mathbf{W} = \mathbf{\Sigma}$ is a minimum point (matrix) for the cost function defined in (11). Using gradient descent optimization theory, the following adaptive equation is used to estimate covariance matrix:

$$\mathbf{\Sigma}_{k+1} = \mathbf{\Sigma}_k + \gamma_k(\mathbf{x}_{k+1}\mathbf{x}_{k+1}^t - \mathbf{\Sigma}_k) \quad (14)$$

Where $\mathbf{\Sigma}_k$ is the estimation of covariance matrix at *k-th* iteration, γ_k is learning rate and $\mathbf{x}_{k+1}$ is the random input vector at iteration *k+1*.

Sanger [16] introduced the following algorithm for adaptive computation of eigenvectors:

$$\mathbf{T}_{k+1} = \mathbf{T}_k + \gamma_k(\mathbf{y}_k\mathbf{x}_k^t - LT[\mathbf{y}_k\mathbf{y}_k^t]\mathbf{T}_k) \quad (15)$$

where $\mathbf{y}_k = \mathbf{T}_k\mathbf{x}_k$ and $\mathbf{T}_k$ is a $p \times n$ matrix that converges to a matrix $\mathbf{T}$ whose rows are the first p eigenvectors of $\mathbf{\Sigma}$. $LT[.]$ sets all entries of its matrix argument which are above the diagonal to zero and γ_k is learning rate where meets Ljung assumptions [15].

The following equation is used for adaptive estimation of the mean vector [17-19]:

$$\mathbf{m}_{k+1} = \mathbf{m}_k + \eta_k(\mathbf{x}_{k+1} - \mathbf{m}_k) \quad (16)$$

Where $\mathbf{m}_k$ is the estimation of the mean vector at *k-th* iteration, $\mathbf{x}_{k+1}$ is the random input vector at iteration *k+1* and η_k denotes the learning rate.

D. New Incremental LDA Algorithm

As discussed in the previous section, the LDA features are significant eigenvectors of $\mathbf{\Sigma}_\mathbf{w}^{-1}\mathbf{\Sigma}$. For adaptive computation of them, we combine two adaptive algorithms discussed in the previous sub-sections in cascade and show that this architecture asymptotically computes LDA features. Let $\mathbf{m}_k^i$ denote the estimated mean vector of class $i (i = 1,2,...,L)$ at *k-th* iteration and $\omega(\mathbf{x}_k)$ denote the class of $\mathbf{x}_k$. The training sequence $\{\mathbf{y}_k\}$ for $\mathbf{\Sigma}^{-1/2}$ algorithm is defined by $\mathbf{y}_k = \mathbf{x}_k - \mathbf{m}_k^{\omega(\mathbf{x}_k)}$. With the arrival of every training sample $\mathbf{x}_k$, $\mathbf{m}_k^i$ is updated according to its class using (16). It is easy to show that the correlation of the sequence $\{\mathbf{y}_k\}$ is the within-class scatter matrix $\mathbf{\Sigma}_w$. Therefore, we have the following equation:

$$\lim_{k\to\infty} E[(\mathbf{x}_k - \mathbf{m}_k^{\omega(x_k)})(\mathbf{x}_k - \mathbf{m}_k^{\omega(x_k)})^t = \lim_{k\to\infty} E[\mathbf{y}_k\mathbf{y}_k^t] = \mathbf{\Sigma}_w \quad (17)$$

Suppose that the sequence $\{\mathbf{z}_k\}$ is defined by, $\mathbf{z}_k = \mathbf{x}_k - \mathbf{m}_k$. Where $\mathbf{m}_k$ is the estimated mixture mean value in *k-th* iteration. We train the $\mathbf{\Sigma}^{-1/2}$ algorithm by the sequence $\{\mathbf{y}_k\}$ and use $\mathbf{W}_k$ in (8-10) to create the new sequence $\{\mathbf{u}_k\}$ as follows, $\mathbf{u}_k = \mathbf{W}_k\mathbf{z}_k$. The sequence $\{\mathbf{u}_k\}$ is used to train the algorithm (15). As mentioned before, the matrix $\mathbf{T}$ in the algorithm (15) will converges to the eigenvectors of the covariance matrix of its input, ordered by decreasing eigenvalues. Hence, is (15) is trained using the sequence $\{\mathbf{u}_k\}$, the matrix $\mathbf{T}$ will converge to the eigenvectors of $E(\mathbf{u}_k\mathbf{u}_k^t)$. It is quite easy to show:

$$\lim_{k\to\infty} E(\mathbf{u}_k\mathbf{u}_k^t) = \mathbf{\Sigma}_w^{-1/2}\mathbf{\Sigma}\mathbf{\Sigma}_w^{-1/2} \quad (18)$$

Our aim is to estimate the eigenvectors of $\mathbf{\Sigma}_\mathbf{w}^{-1}\mathbf{\Sigma}$. Suppose $\mathbf{\Phi}$ and Λ denote the eigenvector and eigenvalue matrices correspond to $\mathbf{\Sigma}_\mathbf{w}^{-1}\mathbf{\Sigma}$. Following equations are held [11]:

$$\mathbf{\Sigma}_w^{-1}\mathbf{\Sigma\Phi} = \mathbf{\Phi\Lambda}, \ \mathbf{\Sigma}_w^{-1/2}\mathbf{\Sigma}\mathbf{\Sigma}_w^{-1/2}\mathbf{\Psi} = \mathbf{\Psi\Lambda}. \quad (19)$$

where $\mathbf{\Psi} = \mathbf{\Sigma}_w^{1/2}\mathbf{\Phi}$. From (19), it is concluded that the eigenvector matrix of $\mathbf{\Sigma}_w^{-1/2}\mathbf{\Sigma}\mathbf{\Sigma}_w^{-1/2}$ is equal to $\mathbf{\Psi}$. In the other words, if (15) is trained using the sequence $\{\mathbf{u}_k\}$, according to (18, 19), the matrix $\mathbf{T}^t$ in the (15) converges to $\mathbf{\Psi}$ and the following equation is held:

$$\lim_{k\to\infty} \mathbf{T}_k^t = \mathbf{\Sigma}_w^{1/2}\mathbf{\Phi} = \mathbf{\Psi} \quad (20)$$

Where $\mathbf{\Psi} = \mathbf{\Sigma}_w^{1/2}\mathbf{\Phi}$. Therefore, if (8-10) are trained using the sequence $\{\mathbf{y}_k\}$ and (15) is trained using the sequence $\{\mathbf{u}_k\}$, by multiplying the outputs of the equations (8-10) and (15) as $k \to \infty$, we will have:

$$\lim_{k\to\infty} \mathbf{W}_k \mathbf{T}_k^t = \mathbf{\Sigma}_w^{-1/2}\mathbf{\Sigma}_w^{1/2}\mathbf{\Phi}^t = \mathbf{\Phi}^t . \qquad (21)$$

Where $\mathbf{\Phi}$ is a matrix whose columns are eigenvectors of $\mathbf{\Sigma}_\mathbf{w}^{-1}\mathbf{\Sigma}$.

III. NETWORK IMPLEMENTATION

Consider a single layer network with the training input vector $\mathbf{x}_k$ and output vector $\mathbf{o}_k$, and let $\mathbf{o}_k = \mathbf{W}_k \mathbf{x}_k$, where $\mathbf{W}_k$ is a weight matrix updated by the sequential update rule presented in (8) (we can construct the related networks for equations (9) and (10) in a same way). Let $\mathbf{x}_k^j$ denote the *j*-th component of $\mathbf{x}_k$, $\mathbf{W}_k(i,j)$ denote the *ij*-th element of $\mathbf{W}_k$, $\mathbf{o}_k^i$ denote *i-th* component of $\mathbf{o}_k$ and $\mathbf{W}_k^i$ correspond to *i-th* row of $\mathbf{W}_k$, respectively. We use adaptive learning algorithm described in (8), in order to train the weight matrix. With arrival of each training Gaussian sample, the weight matrix is updated in order to take into consideration the effect of the new training Gaussian data. The weight matrix update can be written as follow:

$$\begin{aligned}\Delta\mathbf{W}_k &= \eta(\mathbf{I} - \mathbf{W}_k \mathbf{W}_k \mathbf{x}_k \mathbf{x}_k^t) \\ &= \eta(\mathbf{I} - \mathbf{W}_k \mathbf{o}_k \mathbf{x}_k^t)\end{aligned} \qquad (22)$$

The element update equation for the weight matrix can be written as:

$$\begin{aligned}&\Delta\mathbf{W}_k(i,j) = \eta(\delta_{ij} - \mathbf{W}_k^i \mathbf{o}_k \mathbf{x}_k^j) = \\ &\eta(\delta_{ij} - [\mathbf{W}_k(i,1)\mathbf{o}_k^1 + \mathbf{W}_k(i,2)\mathbf{o}_k^2 + \ldots + \mathbf{W}_k(i,n)\mathbf{o}_k^n]\ \mathbf{x}_k^j\end{aligned} \qquad (23)$$

where δ_{ij} is the Kronecker's delta function (when *i=j* then $\delta_{ij}=1$, otherwise $\delta_{ij} = 0$).

Figure 1 shows a network implementation for updating $\mathbf{W}_k(i,j)$ $(i,j=1,2,\ldots,n)$ according to (22) and (23). In Figure 1, $\mathbf{o}$ is a $n\times 1$ vector denoted output, $\mathbf{W}_k^i$ is *i-th* row of the estimated $\mathbf{W}_k$ matrix in *k-th* iteration, z^{-1} denotes one unit delay and $\mathbf{x}_k^j$ denote the *j*-th component of $\mathbf{x}_k$. Network implementation of equations (9) and (10) are same as (8). For construction LDA network, we cascade two independent networks related to algorithms (8) ((9) or (10)) and (15). The first network is for the computation of the $\mathbf{\Sigma}^{-1/2}$ and the second network is based on the Sanger's algorithm [16]. Figure 2 shows block diagram of cascade architecture. It is clear that, multiplication of weight matrix in the second layer of LDA network (Figure 2, $\mathbf{\Sigma}_w^{1/2}\mathbf{\Phi}$) and weight matrix of $\mathbf{\Sigma}^{-1/2}$ network ($\mathbf{\Sigma}^{-1/2}$) will converges to $\mathbf{\Phi}$, whose columns represent the LDA features.

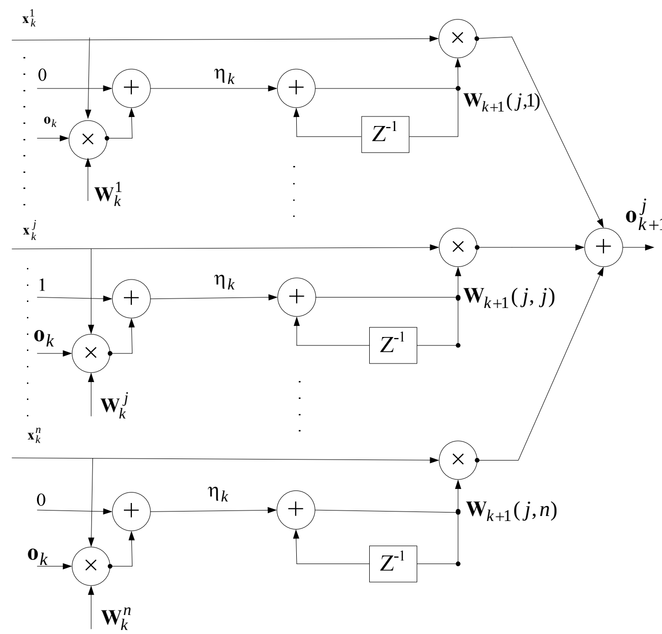

Fig. 1 Network implementation of $\mathbf{\Sigma}^{-1/2}$ learning algorithm according to (8).

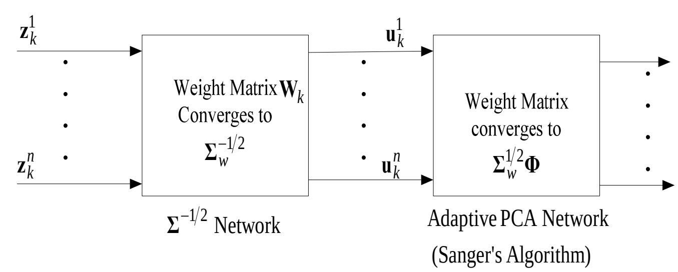

Fig. 2 Network implementation for LDA feature estimation

IV. INCREMENTAL FACE RECOGNITION SYSTEM

In this section, we introduce a new IFR system constructed based on the proposed neural networks. To prepare the face images, we reduced the background information and cropped all face images into size of 40×40. In addition, all face images, histogram equalized, normalized and mean centered (equation (16) can be used for adaptive computation of mean image). Then using an APCA algorithm we reduced dimensionality of all face images to 60. Equation (14) can be used for adaptive estimation of covariance matrix, and then by projection of input images on the eigenvectors of covariance matrix, it is possible to reduce the image dimensionality. Preprocessing and dimension reduction are necessary before applying the face images into the proposed IFR system. Figure 3 shows the proposed IFR system constructed based on the introduced neural network. Every face images (converted to a 60 ×1 vector) enters the $\mathbf{\Sigma}^{-1/2}$ network (Figure 1) and output of $\mathbf{\Sigma}^{-1/2}$ network goes through an APCA network (Figure 2). Finally, output of $\mathbf{\Sigma}^{-1/2}$ network and APCA network are multiplied to make the LDA projection matrix. By projecting every input face image into this matrix, we can find LDA significant features and represent the image in a lower dimensional LDA feature space in addition to receive the high separability among different face images. It is important that inputs of this system are preprocessed (histogram equalized, mean centered), dimension reduced face vectors.

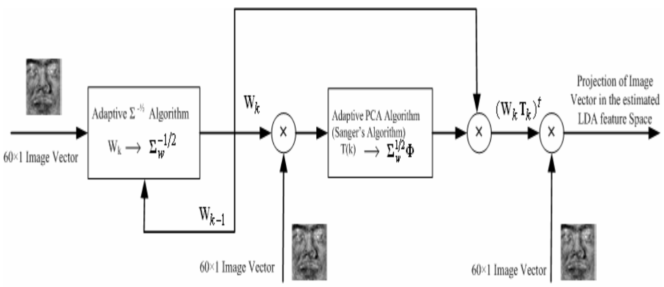

Fig. 3 Block diagram of proposed IFR system.

V. Simulation Results

We applied the proposed new IFR system on YALE face dataset for online face classification. Preprocessing that includes: background omission, histogram equalization, normalization and mean centering is done on all input faces. All input images vectorized and their size is reduced to 60×1 using APCA algorithm. In experiment described in this section, we considered input images as a sequence of random data and simultaneously trained the proposed IFR system.

Fig. 4 Sample images from 5 subjects in different posses and illuminations

YALE database contains grayscale images of 15 subjects in GIF format. In this experiment, we chose 5 individual subjects and considered 64 images per each subject (total 320 images) containing different illumination and different poses. Figure 4 shows some of selected subjects in different position and illumination. We trained the proposed IFR system using these images as a stream. In this experiment, we considered 5 subjects; therefore there exist 4 LDA features. Using the introduced IFR system, we estimated the three significant LDA features and by projection of face images onto them, we reduced the data dimension into three. Figure 5 shows convergence of three estimated LDA features used for dimension reduction during the training process. In figure 5 we computed the normalized error between estimated LDA feature and real LDA feature (computed using scatter matrices) in each iterations. It is clear from figure 5, as the number of training images increases, LDA features converges to their real values. At the end of experiment we get the normalized errors equal to 0.121, 0.232 and 0.305 for the first, second and third LDA significant features. Figure 6 shows distribution of face images in the estimated three dimensional feature space during the training phase. Face images related to each subject verified by a different color and symbol. The top left figure shows distribution of face images after 80 iteration on the estimated LDA feature space and other three figures demonstrated the distribution of subject images in feature space after 120, 220 and 320 iteration, respectively. it is clear from figure 6 that face images at first iterations are not clearly separable but gradually by adaptive training of IFR system, each subjects separate from others and at the end of process (after 320 iteration) all of the subjects are linearly separable (although overlapping) in three dimensional estimated feature space.

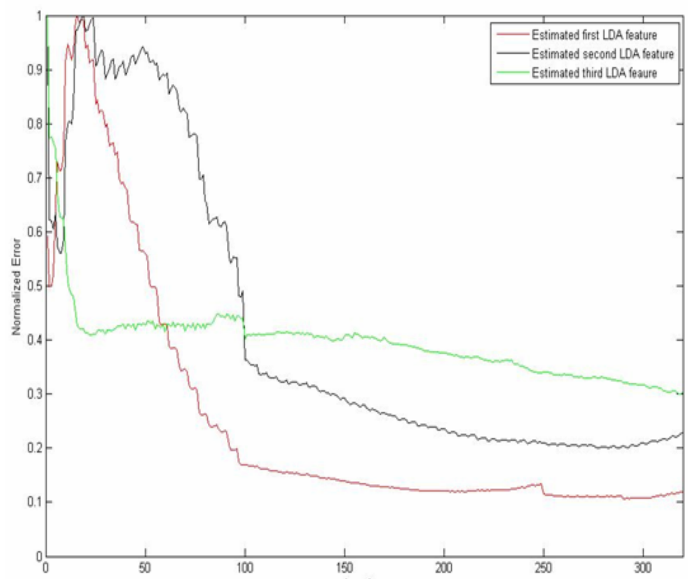

Fig. 5 Convergence of three significant LDA features, estimated using proposed IFR system.

VI. Concluding Remarks

In this paper, a new IFR system based on adaptive LDA is introduced. The new IFR system was considered as a combination of a new adaptive $\mathbf{\Sigma}^{-1/2}$ network in cascade with APCA network. Convergence proof of the new adaptive learning algorithms and networks is given by introducing an appropriate cost function and discussing about its initial conditions. Simulation results on YALE face dataset demonstrated the ability of the proposed algorithm and network for adaptive optimal feature extraction. The new adaptive method can be used in many fields of online machine learning and pattern recognition applications such as face and gesture recognition and mobile robotics.

ACKNOWLEDGMENT

This project was supported partially by Iranian Telecommunication Research Center (ITRC).

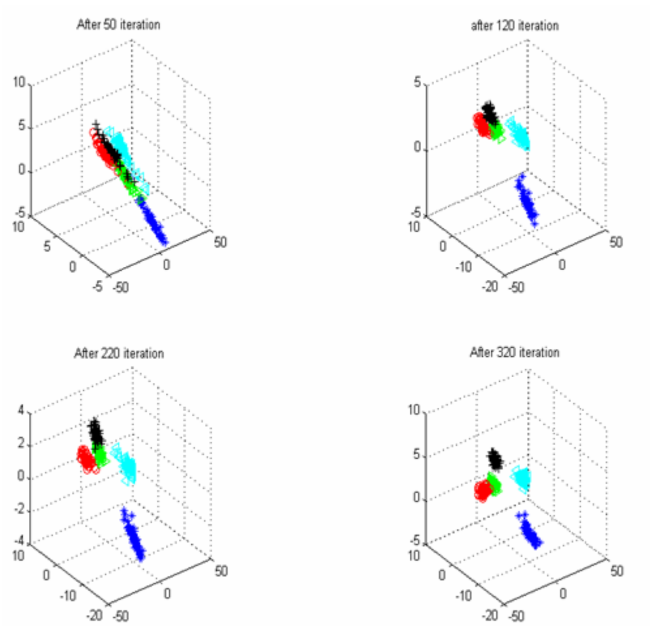

Fig. 6. Distribution of face images in the estimated three dimensional feature space, after 50, 120, 220 and 320.

REFERENCES

[1] Z. sun, G. Bebis, R. miller, 2004, object detection using feature selection. Pattern recognition, Elsevier Vol. 37, no 11, pp 2165-3176.

[2] M. Turk, A. Pentland, "Eigenfaces for face recognition", journal cognitive neuroscience, Vol. 3, No.1, pp. 71-80, 1991.

[3] P. Belhumeur, J. Hespanha, D. Kregman, "Eigenfaces vs. Fisherfaces: recognition using class specific linear projection".IEEE Trans. Pattern Anal. Machine Intell.,Vol 19, pp711-720,1997.

[4] O.Deniz, M. Castrill_on, M. Hernandez, " Face recognition using independent component analysis and support vector machines" , Pattern Recognition letters, Vol. 24, pp. 2153-2157, 2003.

[5] J. Lu, K. N. Plataniotis and A.N. Venetsanopoulos," Face recognition using LDA –based algorithms", IEEE Trans. Neural Networks, Vol. 14, No. 1, 2003.

[6] R. Chellappa, C.L. Wilson, and S.Sirohey, "Human and machine recognition of faces: A survey", Proc. IEEE0, Vol. 83, pp. 705-740, May 1995.

[7] O. Deniz, M. Castrllon, J. Lorenzo, N. Hernandez, "an incremental learning algorithm for face recognition," in M. Tistarelli, J. Bigun, A.K. Jain (Eds), Biometric Authentication , pp. 1-9, Speringer-Verlag, 2002.

[8] S.L. Toh, S. Ozawa, "A face recognition system using neural networks with incremental learning ability", Proc. 8^{th} Australian and New Zealand Conf. on Intelligent Information Systems, pp. 389-394, 2003.

[9] J. Weng, C.H. Evans, W.S. Hwang, "An incremental learning method for face recognition under continuous video stream", Proc. 4^{th} IEEE Int. Conf. Automatic face and gesture recognition, pp. 251-256, 2000.

[10] S. Ozawa, S.L. Toh, S. Abe, S. Pang, N. Kasabov, "Incremental learning of feature space and classifier for face recognition", Neural Networks, Vol. 18, pp. 575-584, 2005.

[11] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, *2^{nd}* Edition, Academic Press, New York, 1990.

[12] J.R. Magnus, H. Neudecker, *Matrix Differential Calculus*, John Wiley, 1999.

[13] B.Widrow, S. Stearns, *Adaptive Signal Processing*, Prentice-Hall, 1985.

[14] M. Hagan, H. Demuth, 2002, *Neural Network Design*, PWS Publishing Company.

[15] L. Ljung, 1977, Analysis of recursive stochastic algorithms , *IEEE Trans. Automat Control*, Vol. 22, pp. 551-575, Aug. 1977.

[16] T.D. Sanger,"optimal unsupervised learning in a single-layer linear feed forward neural network", Neural Networks, Vol. 2, pp. 459-473, 1989.

[17] C. Chatterjee, V. P. Roychowdhurry, "On self-organizing algorithm and networks for class separability features", IEEE Trans. Neural Network, Vol. 8, No. 3, pp. 663-678, 1997.

[18] Y. Aliyari Ghassabeh, H. Abrishami Moghaddam, "New Adaptive Algorithms for Optimal Feature Extraction from Gaussian Data", Proc. 2nd International Conference on Machine Vision Theory and Applications (VISAPP 2007), Spain, March 2007.

[19] H. Abrishami Moghaddam, M.Matinfar, S.M. Sajad Sadough, Kh. Amiri Zadeh, "Algorithms and networks for accelerated convergence of adaptive LDA", Pattern Recog , Vol. 38, No. 4, pp. 473-483, 2005.