J. Basic. Appl. Sci. Res., 3(3)1110-1117, 2013
© 2013, TextRoad Publication
ISSN 2090-4304
Journal of Basic and Applied Scientific Research
www.textroad.com

Face Recognition with the Mixture of MDA and MPCA

Ali Akbar Shams-Baboli[*1], Mohsen Kaffashpour-Yazdi[2], Aref Shams-Baboli[3], Samad Araghi[4]

[1]Msc student at Dept. of Electrical Engineering Iran University of Science and Technology
Tehran, Iran
[2] NAOC Dispatching Center, Mazandaran Regional Electrisity Co. Sari, Iran
[3]BSc Student at Dept. of Electrical and computer Engineering Noshirvani University of Technology
Tehran, Iran
[4]Telecommunication Department of North Power Transmission Maintenance Co. (TANESH) Sari, Iran

ABSTRACT

Many algorithms had been proposed for face recognition problems in the last few years, but none of them could defeat the enormous changeability of some environmental parameters such as: lighting, scale and pose; and the SSS problem at the same time. In this paper, we propose a method for improving the robustness of a face recognition algorithm with tensors representation and fusion of the MDA and MPCA. a multilinear principal component analysis(MPCA) for tensor object feature extraction and a multilinear discriminant analysis(MDA), to find the best subspaces have been proposed. It should be noted that both of them work with tensor objects so the structure of the objects has been never broken. Therefore we achieve a better performance. For the final decision, these two criterion are combined according to a given combination rule. We propose two algorithms for the fusion phase: the K-Nearest Neighbors and the Nearest Mean. Finally, a comprehensive experiment has been provided to demonstrate that this fusion on tensor base algorithm has the potential to outperform the traditional vector based subspace learning methods, especially in the cases with small sample sizes and curse of dimensionality and variability of many environmental parameters such as lighting, pose and scale. Experimental results approve that fusing of these two tensor based approaches gives us better recognition accuracy.
KEYWORDS- Fusion; Tensor objects; Multilinear principal component analysis; Multilinear discriminant analysis; subspace learning.

I. INTRODUCTION

Because of Face Recognition wide applications, it has become one of the most important tasks in the pattern recognition problems in the last few years [1]. These applications are: forensic applications, video supervision and finding criminals for investigations. The word tensor itself is actually used for a multidimensional object. The numbers of indices will define the order of our tensor object and each of them refers to one modes. Each image and video data is naturally a tensor object. For example, grayscale image is a second order tensor object(2-D object) or color images are 3-D (third-order tensor) objects with column, row, and color modes and color video sequences are fourth-order tensors with a color mode [1]. For a face image the acquisition process is non-invading, simpler and cheaper than fingerprint and the iris and does not need to be collaborated, because of these reasons face is a good criterion for recognitions. From the other point of view, because most of the parameters for a face image (such as: expressions, scale, pose …) are variable many problems will be shine.

A typical tensor object in machine vision or pattern recognition applications is actually in a high-dimensional tensor space. In reality, the extracted feature of an object has specific structures. Some of them are in second order tensor or third or even higher order tensors [2]. Feature extraction or dimensionality reduction is transforming a high-dimensional data set into a lower space while keeping most of the information regarding the underlying structure. Linear discriminant analysis (LDA) [3], [4] has been one of the most popular techniques employed in the face recognition. The basic idea of the Fisher Linear discriminant is to calculate the Fisher optimal discriminant vectors to maximizing the ratio of the between class scatter and the within class scatter. For that purpose, in the process of finding the optimal vectors some restrictions will be added to reduce the error rate for face recognition.

Most of the previous algorithms for classification and dimensionality reduction transform the input image pixel into a vector, which disregard the underlying data structure so these methods suffer from curse of dimensionality [5], [6], [7] and often leads us to the small sample size problem. Computationally, handling high dimensional samples is so expensive and many classifiers have a very bad performance in high dimensional spaces when the training set is small. Some recent works have started to consider an object as a 2-D matrix (second order tensors) rather than vectors (first order tensors) for subspace learning. A 2-D PCA algorithm is proposed in [8], where gets the input images as a matrix and compute a covariance matrix. To solve these problems, we proposed a method that is based on tensor representation, so the structure of the objects will not break so it gives us the better

***Corresponding Author:** Ali Akbar Shams-Baboli, Msc student at Dept. of Electrical Engineering Iran University of Science and Technology, Tehran, Iran .ali.shams2222@gmail.com

performance and also we use 2 different algorithms for classification step. We fuse 2 multilinear algorithms to cover their defects and increasing the performance of the system.

In this paper the MDA and MPCA algorithms have been proposed, and because those algorithms work with tensor objects, those algorithms could give us the better results. We use each of them separately and for the final decision, these two algorithms are combined according to a given combination rule. We propose two algorithms for the fusion phase: 1) the K-Nearest Neighbors (KNN) and 2) the Nearest Mean (NM). The important part is using multiliner algorithm for fusion phase. MPCA and MDA can prevented the curse of dimensionality dilemma by applying higher order tensor for objects and n-mode optimization approach. Due to using the MDA [9], [10], after applying the MPCA [11], [12], this method is managed in a much lower dimension feature space than MDA and the traditional vector based methods, such as PCA and LDA do. Because of the structure of MDA, it can overcome the small sample size problem. Theoretically, the maximum feature dimension of LDA is limited by the number of classes in the data set but in our algorithm it is not limited. So it could give us the better recognition accuracy. By using tensor the algorithm will be more effective and also learn much more easily. we expect this novel method to be a better choice than LDA and PCA algorithms and a more general algorithm than MDA and MPCA for the pattern classification problems in image analysis and also overcome the small sample sizes and curse of dimensionality dilemma and also more robust on variation of scale, lighting and pose [13].

The rest of this paper is organized as follows. Section II introduces basic multilinear algebra notations and concepts. In Section III, the Initialization procedures of MPCA and introducing the DTC and n-mode optimization that are used in MDA is discussed. Then in section IV the fusion rule and its methology will be presented and after that in Section V, we show the face recognition experiments by coding the image objects as third order tensors and compare them with traditional subspace learning methods and MDA and MPCA algorithm and finally in section VI conclusion will be presented.

II. MULTILINEAR NOTATIONS AND BASIC MULTILINEAR ALGEBRA

This section briefly will be reviewed some basic multilinear concepts used in our framework and see an example for n–mode unfolding of a tensor. Here, vectors are marked by lowercase boldface letters, such as, x, y. The bold uppercase symbols are used for representing matrices, such as U, S, and tensors will be used by calligraphic letters, e.g. $\mathcal{A}$. An Nth-order tensor is denoted as $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$. It is addressed by N indices In, n = 1, … , N and each In refers the n-mode of $\mathcal{A}$. The n-mode product of a tensor $\mathcal{A}$ by a matrix U, denoted by $\mathcal{A} \times_n U$ is

$$(\mathcal{A} \times_n U)(i_1, \dots, i_{n-1}, j_n, i_{n+1}, \dots, i_N) = \sum_{i_n} \mathcal{A}(i_1, \dots, i_N) . U(j_n, i_n) \qquad (1)$$

The scalar product of two tensors $\mathcal{A}, \mathcal{B} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ is defined as $\prec \mathcal{A}, \mathcal{B} \succ = \sum_{i_1} \sum_{i_2} \dots \sum_{i_N} \mathcal{A}(i_1, \dots, i_N) . \mathcal{B}(i_1, \dots, i_N)$ and the Frobenius norm of $\mathcal{B}$ is defined as $\|\mathcal{B}\|_F = \sqrt{\prec \mathcal{B}, \mathcal{B} \succ}$ [14], [15].

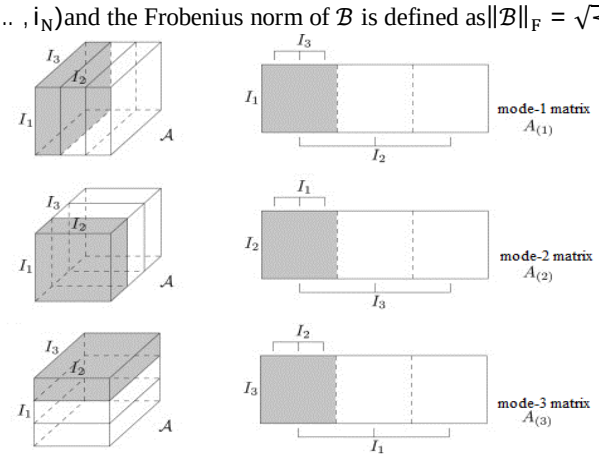

Figure 1. Illustration of the *n-mode unfolding* of a third –tensor.

The "n mode vectors" of $\mathcal{A}$ are determined as the In-dimensional vectors achieved from $\mathcal{A}$ by varying the index i_n while all the other indices are fixed. A rank-1 tensor $\mathcal{A}$ equals to the outer product of N vectors $\mathcal{A} = u^{(1)} \circ u^{(2)} \circ \dots \circ u^{(N)}$, which means that $\mathcal{A}(i_1, \dots, i_N) = u^{(1)}(i_1) \circ u^{(2)}(i_2) \circ \dots \circ u^{(N)}(i_N)$ for all values of indices. Spreading along the n mode is marked as $A_{(n)} \in \mathbb{R}^{I_n \times (I_1 \times \dots \times i_{n-1} \times i_{n+1} \times \dots \times i_N)}$. The column vectors of $A_{(n)}$ are the n mode vectors of $\mathcal{A}$. "Fig. 1" illustrates three ways to unfold a third-order tensor. For unfolding along the first-mode, a tensor is unfolded into a matrix along the I1 axis, and the matrix width

direction is indexed by searching index I2 and I3 index iteratively. In the second-mode, the tensor is unfolded along the I2 axis and the same trend afterwards. Tensor $\mathcal{A}$ can be spread as the product:

$$\mathcal{A} = \mathcal{S} \times_1 U^{(1)} \times_2 U^{(2)} \times \ldots \times_N U^{(N)} \qquad (2)$$

Where $\mathcal{S} = \mathcal{A} \times_1 U^{(1)^T} \times_2 U^{(2)^T} \times \ldots \times_N U^{(N)^T}$ and we call $\mathcal{S}$ core tensor that will be used for HOSVD and $U^{(n)} = \left(u_1^{(n)} u_2^{(n)} \ldots u_{I_n}^{(n)}\right)$ is an orthogonal $I_n \times I_n$ matrix. Since $U^{(n)}$ have orthonormal columns, $\|\mathcal{A}\|_F{}^2 = \|\mathcal{S}\|_F{}^2$ [9]. The relationship between unfolded tensor $A_{(n)}$ and its decomposition core tensor $\mathcal{S}_{(n)}$ is

$$A_{(n)} = U^{(n)} \cdot \mathcal{S}_{(n)} \cdot (U^{(n+1)} \otimes U^{(n+2)} \otimes \ldots \otimes U^{(N)} \otimes U^{(1)} \otimes U^{(2)} \otimes \ldots \otimes U^{(n-1)})^T \qquad (3)$$

Where $\otimes$ means the Kronecker product[14]. The decomposition can also be written as

$$\mathcal{A} = \sum_{i_1=1}^{I_1} \sum_{i_2=2}^{I_2} \ldots \sum_{i_N=1}^{I_N} \mathcal{S}(i_1, i_2 \ldots, i_N) \times u_{i_1}^{(1)} \circ u_{i_2}^{(2)} \circ \ldots \circ u_{i_N}^{(N)} \qquad (4)$$

The projection of an n-mode vector of $\mathcal{A}$ by $U^{(n)^T}$ is calculated as the inner product between the rows of $U^{(n)^T}$ and the n-mode vector.

For example in Fig. 2, a third order tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ is projected in its one mode vector space by a projection matrix $\mathcal{B}^{(1)^T} \in \mathbb{R}^{m_1 \times I_1}$, the projected tensor is $\mathcal{A} \times_1 \mathcal{B}^{(1)^T} \in \mathbb{R}^{m_1 \times I_2 \times I_3}$. In the one mode projection, each one mode vector of length I_1 is projected by $\mathcal{B}^{(1)^T}$ to obtain a vector with the length of m_1 that is lower than the original dimension. As you can see the dimension of tensor in that mode should be the same as the dimension of projection matrix column. With this product we project our tensor in that specific mode into a lower space and it will help us to separate different classes more.

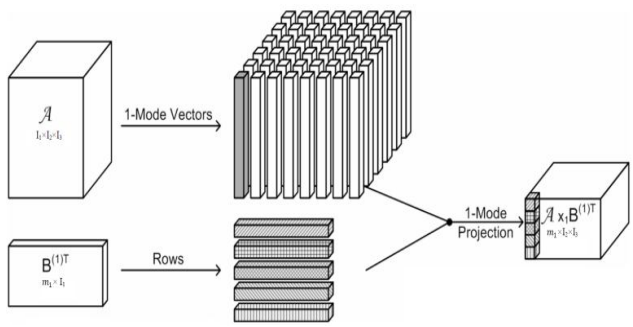

Figure 2. Illustration of multilinear projection in the 1 –mode vector space

III. MULTILINEAR PRINCIPAL COMPONENT ANALYSIS AND MULTILINEAR DISCRIMINANT ANALYSIS

Most of the previous algorithms to dimension reduction, such as LDA and PCA, treated the object as one dimension vector so the learning algorithms should apply on a very high dimension feature space. Often it makes these methods to suffer from the problem of curse of dimensionality. Most of the objects in computer vision are more naturally represented as second or higher order tensors. For example, the image matrix in Fig. 3(a) is a second order tensor and the Fig. 3(b) shows a third-order tensor.

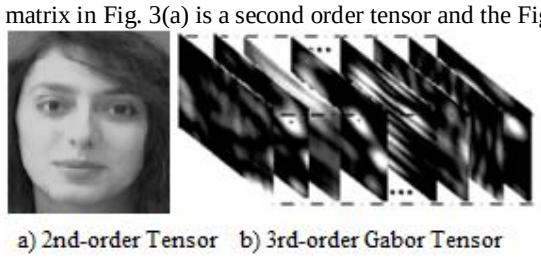

Figure 3. Second and third-order Tensor representations samples

In this section, first we see, how the MPCA solution for tensor objects is working and then we will see the DTC and n-mode optimization that is used in MDA for tensor objects. A set of M tensor objects $\{\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_M\}$ is attainable for training. Each tensor object $\mathcal{X}_m \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ surmises values in a tensor space $\mathbb{R}^{I_1} \otimes \mathbb{R}^{I_2} \ldots \otimes \mathbb{R}^{I_N}$, where I_n is the n-mode dimension of the

tensor. The MPCA defines a multilinear transformation that maps the original tensor space into a tensor subspace. In other words, the MPCA objective is the determination of the projection matrices $\{U^{(n)} \in \mathbb{R}^{I_n \times P_n}, n = 1, \dots, N\}$ that maximize the total tensor scatter, Ψ_y

$$\{U^{(n)}, n = 1, \dots, N\} = \arg \max_{U^{(1)}, U^{(2)}, \dots, U^{(N)}} \Psi_y \qquad (5)$$

Where

$$\Psi_y = \sum_{m=1}^{M} \|\mathcal{A}_m - \bar{\mathcal{A}}\|_F^2 \ , \ \bar{\mathcal{A}} = (1/m) \sum_{m=1}^{M} \mathcal{A}_m \qquad (6)$$

Here, the dimensionality P_n for each mode $U^{(n)}$, is assumed to be known or predetermined.

A. MPCA Algorithm

There is no optimal solution for optimizing the N projection matrices simultaneously. An Nth-order tensor consists of N projections with N matrix, so N optimization subproblems can be solved by finding the $U^{(n)}$ that maximizes the scatter in the n-mode vector subspace. If $\{U^{(n)}, n = 1, \dots, N\}$ be the answer of (5) and $U^{(1)}, \dots, U^{(n-1)}, U^{(n+1)}, \dots, U^{(N)}$ be all the other known projection matrices, the matrix $U^{(n)}$ consists of the P_n eigenvectors corresponding to the largest eigenvalues of the matrix $\Phi^{(n)}$

$$\Phi^{(n)} = \sum_{m=1}^{M} \left(X_{m(n)} - \bar{X}_{(n)}\right) . U_{\Phi^{(n)}} . U_{\Phi^{(n)}}^{T} . \left(X_{m(n)} - \bar{X}_{(n)}\right)^{T} \qquad (7)$$

Where

$$U_{\Phi^{(n)}} = U^{(n+1)} \otimes U^{(n+2)} \otimes \dots \otimes U^{(N)} \otimes U^{(1)} \otimes U^{(2)} \otimes \dots U^{(n-1)} \qquad (8)$$

The proof of (7) is given in [11].

Since $\Phi^{(n)}$ depends on all the other projection matrices, there is no closed-form solution to this maximization problem. Instead, reference [11] introduce an iterative procedure that can be utilized to solve (5). For initialization, MPCA used full projection. The term full projection refers to the multilinear projection for MPCA with Pn=In for n= 1, … , N. There is no dimensionality reduction through this full projection. The optimal is obtained without any iteration, and the total scatter in the original data is fully captured. As we know, if all eigenvalues (per mode) are distinct, the full projection matrices are also distinct. Therefore, the full projection is unique [15]. This FPT initialization is equivalent to the HOSVD[9] solution that is proofed in [18]. Although this FPT is not the optimal solution to (4), but it is a good starting point for the iterative procedure. After finding the projection matrices, $U^{(n)}, n = 1, \dots, N$, we applied those matrices to the training set. At this point, we provide a set of tensors with the new dimension that would be the new training set for MDA algorithm.

B. Multilinear Discriminant Analysis

Here, the DTC is introduced which is used in MDA algorithm. The DTC is prepared to set multiple interrelated projection matrices, which maximize the scatter between classes and at the same time minimize the scatter of each class. That is

$$U^{(n)*}\Big|_{n=1}^{N} = \arg \max_{U^{(n)}\big|_{n=1}^{N}} \frac{\sum_c n_c \left\|\bar{\mathcal{X}}_c \times_1 U^{(1)} \dots \times_n U^{(n)} - \bar{\mathcal{X}} \times_1 U^{(1)} \dots \times_n U^{(n)}\right\|^2}{\sum_i \left\|\mathcal{X}_i \times_1 U^{(1)} \dots \times_n U^{(n)} - \bar{\mathcal{X}}_{c_i} \times_1 U^{(1)} \dots \times_n U^{(n)}\right\|^2}. \qquad (9)$$

Where the average tensor of a class c is showing with $\bar{\mathcal{X}}_c$, $\bar{\mathcal{X}}$ is the average tensor of all the available samples, and n_c is number of sample in class c. Equation (9) is equivalent to a higher-order nonlinear optimization problem with a higher-order nonlinear restriction; Therefore, it is difficult to find a closed form solution. We could optimize that function by using n-mode optimization approach from only one direction of the tensor with equation below:

$$U^{(n)*} = \arg \max_{U^{(n)}} \frac{\sum_c n_c \left\|\bar{\mathcal{X}}_c \times_n U^{(n)} - \bar{\mathcal{X}} \times_n U^{(n)}\right\|^2}{\sum_i \left\|\mathcal{X}_i \times_n U^{(n)} - \bar{\mathcal{X}}_{c_i} \times_n U^{(n)}\right\|^2} \qquad (10)$$

The optimization problem in (10) can be reformulated as a special discriminant analysis problem as follows:

$$U^{(n)*} = \arg \max_{U^{(n)}} \frac{Tr(U^{(n)T} S_B U^{(n)})}{Tr(U^{(n)T} S_W U^{(n)})}$$

$$S_B = \sum_{j=1}^{\prod_{o\neq n} m_o} S_B^j, S_B^j = \sum_{c=1}^{N_c} n_c(\bar{X}_{c_{(n)}}^{\ j} - \bar{X}_{(n)}^j)(\bar{X}_{c_{(n)}}^{\ j} - \bar{X}_{(n)}^j)^T$$

$$S_W = \sum_{j=1}^{\prod_{o\neq n} m_o} S_W^j, S_W^j$$

$$= \sum_{i=1}^{N_c} (\bar{X}_{i_{(n)}}^{\ j} - \bar{X}_{c_{i_{(n)}}}^{\ j})(\bar{X}_{i_{(n)}}^{\ j} - \bar{X}_{c_{i_{(n)}}}^{\ j})^T \quad (11)$$

Where, $X_{i_{(n)}}^j$ represents the jth column vector of matrix $X_{i_{(n)}}$ which is the n-mode spread matrix from sample tensor $\mathcal{X}_i$. $\bar{X}_{c_{(n)}}^{\ j}$ and $\bar{X}_{(n)}^j$ are defined in the same way as $X_{i_{(n)}}$ with respect to tensors $\bar{\mathcal{X}}_c$ and $\bar{\mathcal{X}}$ and the proofs are given in [10]. To utilizing n-mode optimization, first the input tensors (that are the outputs of MPCA) should be projected with all the other modes matrices and then all the new tensors are unfolded into a matrix along the nth-mode so the optimization problem in (10) could be recompose as a special discriminant analysis problem, and it could be solved in the same way for the LDA algorithm [8]. Since DTC has no closed form the projection matrices should be iteratively optimized.

IV. FUSION OF MDA AND MPCA FOR FACE RECOGNITION

Here, we will present our new method that is fusing the MPCA and the MDA for recognition rules. We already saw the fusion of LDA and PCA for face recognition problems with good results [15].the procedure of our approach will be shown here: first, we applied the MPCA and the MDA algorithms separately on target database and train these algorithms and find the projection matrix for each of them in each mode. After that we project each of the training set to MPCA and MDA area, for recognition the new given image of face we also should project that into those area and compute the distance between this image to all of the other faces (d^{MPCA} and d^{MDA}), for the final decision, we combine these two vector with two proposed algorithms. We use the K-Nearest Neighbors (KNN) and the Nearest Mean (NM) for this fusion step. In the following, we briefly describe that how this method works.

All the reduction algorithms that had been proposed for recognition such as LDA and PCA had been applied separately to project the image to the lower space for face recognition problems and never used together. Because of the correlation between these two algorithms we inspired to mixture the multi linear form of these two. And because of the difference at their projection space we interested to try mixing them and get the better recognition accuracy.

In our propose methods first, we normalize the distance vectors d^{MPCA} and d^{MDA} between zero and one for reducing the computation complexity. After that at the second step, we compute a vector d that is the combination distance and of course includes information of MPCA and MDA. For finding this vector we propose two methods that are presented here:

First method; we compute the vector of distance for both algorithms and combined them with the eq (12). It means, we using the average distance for classifying:

$$d = \{\frac{d_1^{MPCA} + d_1^{MDA}}{2}, \dots, \frac{d_n^{MPCA} + d_n^{MDA}}{2}\} \quad (12)$$

The second method is combining distance vector by appending the dMPCA vector and dMDA vector:

$$d = \{d_1^{MPCA}, \dots, d_n^{MPCA}, d_1^{MDA}, \dots, d_n^{MDA}\} \quad (13)$$

Where N is the number of train face in our data set. All the d_j^{MPCA}, d_j^{MDA} s, j = 1,… N are for one class.

For the KNN method, we are computing the distance vector d and arranging that. Then for this decision we should find the most similar first K components for selecting. If we use the eq. (12), we named this method Mean KNN (M-KNN); if we use eq. (13), it would be marked as Append KNN (A-KNN).

For the NM approach, first, we find the average image for each class from the training set then we compute the distance from each of them with MPCA and MDA. Of course, in this case we should compute our image from C other image. Here again we use eq. (12), (13) for fusing the distance vector and finally the class will be selected to the image that has smallest distance. For more illustration, fig 4 shows the flowchart of our proposed method. As it can be seen, we use two multilinear algorithms for learning steps and also proposing two rules for classification as explained before.

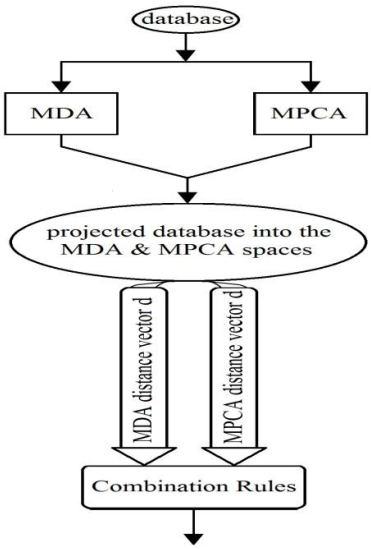

Figure 4. Overview of our fusion method

V. EXPERIMENTAL RESULTS

We used four standard face databases, AT&T [16], FERET[17], CMU-PIE[18] and Yale[19] to evaluate the effectiveness of our proposed algorithm. These algorithms were contrasted with other popular algorithms. In this work, we report the best result on different test and on different dimensions for the fisherface. The performances on the problems with low number of training samples were also appraised to demonstrate their robustness in the small sample size cases.

A. AT&T Database

AT&T database includes 400 images of 40 persons. These images were taken at different times with different expression such as smiling or nonsmiling, open or closed eyes and with glasses or without glasses. The database was divided into a training set, with 5 images for each class and 5 images for each one to be recognized. Fig 5 shows the result of face recognition accuracies.

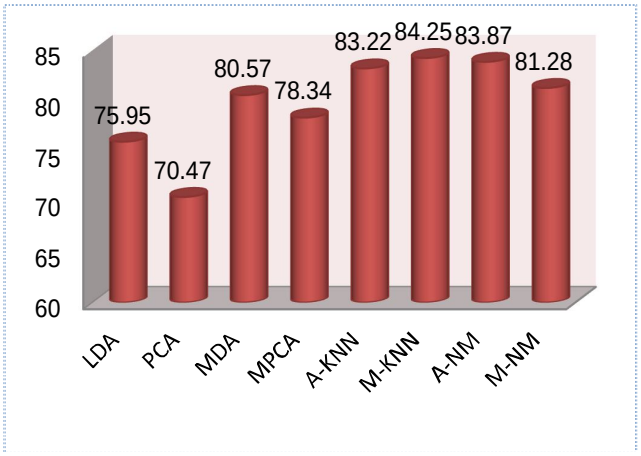

Figure 5. Recognition Accuracy(%) Comparsion on AT&T Database

As result shows, it is worth trying our method for face recognition and our results are worth comparing to the traditional algorithms like LDA and PCA. These results presented our proposed procedure as a multi classifier for face recognition on the AT&T dataset. The accuracy that is achieved by K-NN is comparable with other previous algorithms. The combination of MPCA and MDA gives us the better performance, and a better robustness.

B. Yale Database

The Yale database is concluded of 11 images for each person and 15 individual people. Each face is characterized by different facial expressions like happy, sad and surprised and configurations suck as with/without glasses, left, normal and right

light, and etc. The database was divided into a training set, with 5 images for each person and a test set, with 6 images for each class. We are evaluating our test for ten times randomly and reported the average performance.

Table 1. Recognition Accuracy(%) Comparsion on Yale Database

Single classifiers				Combined classifiers			
LDA	**PCA**	**MDA**	**MPCA**	**A-KNN**	**M-KNN**	**A-NM**	**M-NM**
82.73	83.24	84.5	83.26	85.32	84.67	84.86	82.46

The challenging problem of The Yale database is intense changes at expression and lighting so this will become our experiment with more complexity and much challenging for our method so the results will be much worse. Table 1 shows the accuracy of our experiment on Yale database.

Even in this case, the fusing of MPCA and MDA gives the best result rather than the other algorithms for this difficult database.

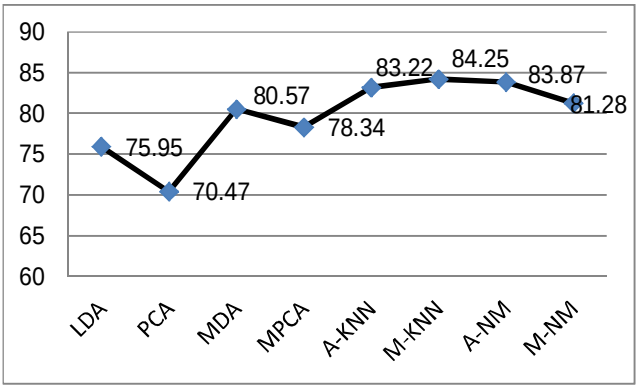

Figure 6. Recognition Accuracy(%) Comparision on CMU PIE Database

C. CMU PIE Database

This database has more than 40,000 images of face for 68 individual people. All faces were achieved under different light and pose and expressions. In our experiment, five almost frontal poses (C5, C07, C29, C09 and C27) and illumination indexed as 08 and 11 were used. The data set was divided into training and test sets randomly and two samples per person were used for training. We take out 40 Gabor features. Fig 6 shows the detailed face recognition system performances.

Figure 7. Six samples of one person in the FERET face database

D. FERET Database

This database is applied on 70 people of the FERET data set with six images for each person. Fig 7 shows six image of one person under different situations from this database. The data set was accidently divided into training and test set; two images were for training and the other four for testing will be used. We compared all the above mentioned algorithms with and without Gabor features on the FERET database. Table 2 shows the comparison of face recognition accuracies. As we see, it shows that the M-KNN has the best performance.

Table 2. Recognition Accuracy(%) Comparision on FERET Database

Single classifiers				Combined classifiers			
LDA	**PCA**	**MDA**	**MPCA**	**A-KNN**	**M-KNN**	**A-NM**	**M-NM**
76.07	75.37	83.57	82.21	85.56	83.29	82.12	81.11

VI. CONCLUSION

In this paper, we present MPCA and MDA algorithms and the fusions of these two statistical approaches for face representation. A MPCA framework is used for analysis of tensor objects and determines a multilinear projection onto a tensor subspace with lower dimensional that captures most of the covariance of the original tensor objects, and MDA has been applied for supervised dimensionality reduction. Compared with common algorithms, such as PCA and LDA, our proposed algorithm effectively prevents the curse of dimensionality dilemma and has no problem with the small sample size.

We combined MPCA and MDA with two rules: the KNN the NM. Overall, the KNN gives us the better recognition accuracy as we expected because it compare the image with all the training images. The recognition accuracy is dependent on the dataset: the difficult one such as Yale database or CMU PIE gives use the worse accuracy at recognition. But as we can see from the results our fusion method gives us the better recognition than the other algorithms.

REFERENCES

[1] A. Jain, R. Bolle, S. Pankanti Eds, "BIOMETRIC – Personal Identification in Networked Society", Kluwer, Academic Publishers, Boston, London, 1999.

[2] M. Turk, A. Pentland, "Eigenfaces for Face Recognition", Journal of Cognitive Neuroscience, Vol. 3, pp. 71-86, No.1, 1991.

[3] P. Belhumeur, J. Hespanha, and D. Kriegman, "Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection," IEEE Trans. Pattern Anal. Mach. Intell., Vol. 19, No. 7, pp. 711–720, Jul. 1997.

[4] W. Zhao, R. Chellappa, A. Rosenfeld, and P. J. Phillips, "Face recognition: A literature survey," ACM Comput. Surveys, pp. 399–458, 2003.

[5] J. Duchene and S. Leclercq, "An optimal transformation for discriminant and principal component analysis," IEEE Trans. Pattern Anal. Machine Intell., Vol. 10, No. 6, pp. 978–983, Jun. 1988.

[6] X. He, D. Cai, and P. Niyogi, "Tensor subspace analysis," in Advances in Neural Information Processing Systems 18 (NIPS). Cambridge, MA: MIT Press, 2005.

[7] G. Shakhnarovich and B. Moghaddam, "Face recognition in subspaces," in Handbook of Face Recognition, S. Z. Li and A. K. Jain, Eds. New York: Springer-Verlag, pp. 141–168, 2004.

[8] J. Yang, D. Zhang, A. Frangi, and J. Yang, "Two-dimensional PCA: A new approach to appearance-based face representation and recognition," IEEE Trans. Pattern Anal. Mach. Intell., vol. 26, no. 1, pp. 131–137, Jan. 2004.

[9] L. D. Lathauwer, B. D. Moor, and J. Vandewalle, "A multilinear singular value decomposition," SIAM J. Matrix Anal. Appl., Vol. 21, No. 4, pp. 1253–1278, 2000.

[10] S. Yan, D. Xu, Q. Yang, L. Zhang, X. Tang, and H.-J. Zhang, "Multilinear discriminant analysis for face recognition," IEEE Trans. Image Process., Vol. 16, No. 1, pp. 212–220, Jan. 2009.

[11] H. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos, "Multilinear principal component analysis of tensor objects for recognition," in Proc Int. Conf. Pattern Recognit., pp. 776–779, Aug. 2010.

[12] H. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos, "MPCA: Multilinear Principal Component Analysis of Tensor Objects," IEEE Trans. Neural Networks, Vol. 19, No. 1, pp. 18-39 Aug. 2011.

[13] L. Haiping, K.N. Plataniotis, and A.N. Venetsanopoulos, "Uncorrelated Multilinear Discriminant Analysis With Regularization and Aggregation for Tensor Object Recognition" Trans on Neural Networks, Vol. 20, No. 1 , pp. 103-123, 2011.

[14] T. Kolda, "Orthogonal tensor decompositions," SIAM J. Matrix Anal. Appl., Vol. 23, No. 1, pp. 243–255, 2000.

[15] G.L. Marcialis ,and F. Roli, "Fusion of LDA and PCA for Face Verification, Proceedings of the Workshop on Biometric Authentication", M. Tistarelli and J. Bigun Eds., Springer LNCS 2359, Copenhagen Denmark, 2002.

[16] F. Samaria and A. Harter (1994), Parameterization of a Stochastic Model for Human face Identification. 2nd IEEE Workshop on Applications of Computer Vision, Saratosa FL. pp 138-142, December, 1994, Available at: http://www.uk.research.att.co/faced databased.html.

[17] I. Philips, H.Wechsler, J. Huang, and P. Rauss, "The FERET database and evaluation procedure for face recognition algorithms," Image Vis. Computer, Vol. 16, pp. 295–306, 1998.

[18] T. Sim, S. Baker, and M. Bsat. "The CMU Pose, Illumination, and Expression (PIE) Database", Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition, May, 2002.

[19] P. N. Bellhumer, J. Hespanha, and D. Kriegman. "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection", IEEE Trans on Pattern Analysis and Machine Intelligence, pp. 711-720, May, 1997, Available at: http://cvc.yale.edu/projects/yalefaces/yalefaces.html.