Multi-Class Discriminant Analysis Based on Linear Classifiers and AdaBoost.M2 for Ranking Principal Components in Face Spaces

Tiene A. Filisbino
National Laboratory for Scientific Computing
Petropolis, Brazil
Email: tiene@lncc.com
Gilson A. Giraldi
National Laboratory for Scientific Computing
Petropolis, Brazil
Email: gilson@lncc.br
Carlos Eduardo Thomaz
Department of Electrical Engineering
FEI, São Paulo, Brazil
Email: cet@fei.edu.br

Abstract

Despite the success of Principal Component Analysis (PCA) for dimensionality reduction, it is known that its most expressive components do not necessarily represent important discriminant features for pattern recognition. In this paper, the problem of ranking PCA components, computed from multi-class databases, is addressed by building multiple linear learners that are combined through the AdaBoost.M2 in order to determine the discriminant contribution of each PCA feature. In our implementation, each learner is a weakened version of a linear support vector machine (SVM). The strong learner built by the ensemble technique is processed following a strategy to get the global discriminant vector to sort PCA components according to their relevance for classification tasks. Also, we show how the proposed methodology to compute the global discriminant vector can be applied to other multi-class approaches, like the linear discriminant analysis (LDA). In the computational experiments we compare the obtained approaches with counterpart ones using facial expression experiments. Our experimental results have shown that the principal components selected by the proposed technique allows higher recognition rates using less linear features.

I. Introduction

Nowadays, increasingly large amount of high dimensional image databases are being generated, leading to a strong demand for dimensionality reduction for discarding redundancy, and features selection techniques to reduce the feature space for discriminating sample groups before executing classification tasks [1].

In this avenue, we follow statistical learning approaches whose basic pipeline can be described as follows [2]: (a) Linear subspace learning for dimensionality reduction; (b) Among the linear components obtained, select the most discriminant ones; (c) Solve the classification problem; (c) Reconstruction problem, that is, visualize the information captured by the discriminant linear components.

The step (a) can be accomplished through classical works on linear dimensionality reduction including the principal component analysis (PCA), factor analysis (FA) [3], multidimensional scaling (MDS) [1] and projection pursuit (PP)[4], [3]. The determination of discriminant features (step (b) above) is very known in the context of PCA. In this case, it was observed that, since PCA explains the covariance structure

of all the data its most expressive components, that is, the first principal components with the largest eigenvalues, do not necessarily represent the most important discriminant directions to separate sample groups [5], [6]. This observation motivates the development of specific techniques to compute discriminant subspaces which, in general, depend on the incorporation of prior information based on labeled data. The Fisher's linear discriminant analysis (LDA) [1], discriminant principal components analysis (DPCA) [5] and its extension to multi-class problems, named Multi-Class DPCA [7], Zhu and Martinez [8] criterion, are techniques reported in the literature for discriminant features section.

The discriminant principal components analysis (DPCA), based on the idea of using the discriminant weights obtained by separating hyperplanes to select among the principal components the most discriminant ones. So, by assuming that there are only two classes to separate the DPCA addresses the problem of selecting the discriminant principal components by using a linear classifier framework. The DPCA methodology presented in [5] focuses on the LDA and linear support vectors machine (SVM) [9] methods although any other separating hyperplane could be used. The efficiency of the discriminant principal components selected by DPCA has been reported for two-group separation tasks in gender and facial expression experiments using frontal face images [5].

In this work we focus on discriminant analysis on multi-class problems. In this case, given an N-class database, the Multi-Class DPCA builds a linear support vector machine (SVM) ensemble, composed of N SVM machines, to get the discriminant weights that are combined through an AdaBoost technique in order to determine the discriminant contribution of each feature.

Ensemble methods find an accurate classifier by combining many moderately accurate learners [10]. The main loop of AdaBoost generates a collection of weak component classifiers and their corresponding AdaBoost weights. In the last step of the AdaBoost procedure, the component classifiers are linearly combined through the computed AdaBoost weights.

In this work we keep the Multi-Class DPCA methodology, but we replace the AdaBoost by the AdaBoost.M2 algorithm and combine the separating SVM hyperplanes through a simple strategy to compute the global discriminant weights. In this way, we get a new ranking method for the principal components, called Multi-Class.M2 DPCA algorithm, given by the group-differences extracted by a linear ensemble and the AdaBoost.M2 technique. The computational experiments demonstrate that the new discriminant technique improves the Multi-Class DPCA for both reconstruction and recognition. Also, we show that the proposed methodology to compute discriminant weights can be applied to other multi-class approaches, like the linear discriminant analysis (LDA).

It is important to highlight that we do not deal with the problem of computing general discriminant directions that are not principal components. Rather, we apply the idea of using a set of linear classifiers and an ensemble method (AdaBoost.M2, in this case) to compute a matrix of discriminant weights that is processed to select among the principal components the most discriminant ones. We have focused here on the SVM [9] method but any other separating hyperplane could be used.

To evaluate the Multi-Class.M2 DPCA algorithm, we perform group separation tasks in facial expression experiments involving neutral, happiness, sad, fear, and anger face images. The experiments show that the SVM can be used as an effective component classifier to generate the discriminant weights for the multi-class discriminant principal components analysis. Furthermore, the computational experiments demonstrate the benefits of sorting principal components using the Multi-Class.M2 DPCA if compared with the traditional PCA, and the Multi-Class DPCA methodologies for selecting PCA components.

The paper is organized as follows. In Section II we revise the PCA methodology. The linear classifiers are presented in Section III and in Section IV. Next, the approaches for ranking components are described in Section V. In Section VI we present the methodology behind of DPCA. The Section VII presents the Multi-Class.M2 DPCA algorithm. The computational experiments are described in section VIII. Finally, in Section IX, we conclude the paper, summarizing its main contributions and describing further developments.

II. Principal Components Analysis (PCA)

PCA is a feature extraction procedure concerned with explaining the covariance structure of a set of variables through a small number of linear combinations of these variables. Let an $M \times n$ training set matrix $\widetilde{\Theta}$ be composed of M input samples (or face images) with n variables (or pixels) centered respect to the global mean, computed by:

$$\widehat{\mathbf{x}} = \frac{1}{M} \sum_{i=1}^{M} \mathbf{x}_i. \tag{1}$$

This means that each column of matrix $\widetilde{\Theta}$ represents the values of a particular variable observed all over the M samples centered respect to the corresponding mean. Let this data matrix $\widetilde{\Theta}$ have covariance matrix Ω with respectively P and Λ eigenvector and eigenvalue matrices, that is:

$$P^T \Omega P = \Lambda. \tag{2}$$

It is a proven result that the set of m' ($m' \leq n$) eigenvectors of Ω, which corresponds to the m' largest eigenvalues, minimizes the mean square reconstruction error over all choices of m' orthonormal basis vectors [11]. Such a set of eigenvectors that defines a new uncorrelated coordinate system for the training set matrix $\widetilde{\Theta}$ is known as the principal components. In the context of face recognition, those $P_{pca} = [p_1, p_2, ..., p_{m'}]$ components are frequently called eigenfaces [12]. The PCA methodology is summarized in the Algorithm 1.

Algorithm 1: Procedure to compute PCA matrix.

1: Compute the global mean $\widehat{\mathbf{x}}$ of the input data through expression (1);
2: Centering input samples: $\widetilde{\mathbf{x}}_i = \mathbf{x}_i - \widehat{\mathbf{x}}$;
3: Build the centered data matrix $\widetilde{\Theta} = [\widetilde{\mathbf{x}}_1^T \widetilde{\mathbf{x}}_2^T \cdots \widetilde{\mathbf{x}}_M^T]$;
4: Determine $P_{pca} = [\mathbf{p}_1, \mathbf{p}_2, ..., \mathbf{p}_{m'}]$.

III. LINEAR SVM

SVM [9] is primarily a two-class classifier that maximizes the width of the margin between classes, that is, the empty area around the separating hyperplane defined by the distance to the nearest training samples. As a consequence, SVM is more robust to outliers, zooming into the subtleties of group differences [13]. It can be extended to multi-class tasks by solving essentially several two-class problems [14]. Given a labeled training set:

$$X = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2) \ldots (\mathbf{x}_M, y_M)\}; \tag{3}$$

where $\mathbf{x}_i$ denote the n-dimensional training observations and $y_i \in \{-1, 1\}$ the corresponding classification labels, the SVM method [9] seeks to find the hyperplane defined by

$$f(\mathbf{x}) = (\mathbf{x} \cdot \phi) + b = 0, \tag{4}$$

which separates positive and negative observations with the maximum margin. It can be shown that the solution vector ϕ_{svm} is defined in terms of a linear combination of the training observations, that is,

$$\phi_{svm} = \sum_{i=1}^{N} \alpha_i y_i \mathbf{x}_i, \tag{5}$$

where α_i are non-negative coefficients obtained by solving a quadratic optimization problem with linear inequality constraints [15], [9]. Those training observations $\mathbf{x}_i$ with non-zero α_i lie on the boundary of the margin and are called support vectors.

Behind the solution given by expression (5) is the assumption that there is a unit vector $\phi \in \Re^n$ and a constant $c \in \Re$ such that the inequalities:

$$\begin{cases} \mathbf{x}_i * \phi > c, \quad if \quad y_i = +1 \quad (a) \\ \mathbf{x}_j * \phi < c, \quad if \quad y_j = -1 \quad (b), \end{cases} \tag{6}$$

hold true ("$*$" denotes the usual inner product in $\Re^n$).

However, sometimes the classes may be nonseparables in the sense that we can not find an hyperplane $x * \phi = c$ such that conditions in expression (6) hold true. In this case, one solution is to work with a more convenient SVM formulation based on minimize the functional [9]:

$$L = \frac{1}{2}(\phi * \phi) + C \sum_{i=1}^{n} \xi_i \tag{7}$$

subject to:

$$\begin{cases} \xi_i \geq 0, \quad i = 1, 2, ..., n, \\ y_i \left((\mathbf{x}_i * \phi) + b \right) \geq 1 - \xi_i, \quad i = 1, 2, ..., n., \end{cases} \tag{8}$$

Nonseparable SVMs allow the decision boundary to misclassify some examples. The constant C must be given in advance and controls the cost for the number of violated constraints. Once solved the optimization problem defined by expressions (7)-(8), the ϕ_{svm} solution is computed by expression (5).

IV. Linear Discriminant Analysis (LDA)

In general, Fisher's Linear Discriminant Analysis (LDA) [16], [11] is used to identify the most important linear directions for separating sample groups [16], [11] rather than PCA. This method, as well as the weighted pairwise variant of the well-known multi-class Fisher criterion introduced in [17] has the limitation of finding number of groups - 1 meaningful discriminant directions.

The main purpose of LDA, represented in the Figure 1, is to separate samples of distinct groups by maximizing their between-class separability while minimizing their within-class variability. Its main objective is to find a projection matrix W^{lda} that maximizes the Fisher criterion:

$$W^{lda} = \arg\max_{W} \frac{\left| W^T S_b W \right|}{\left| W^T S_w W \right|} \tag{9}$$

where S_b and S_w are the between-class and within-class matrices, respectively, which are defined as:

$$S_b = \sum_{i=1}^{g} N_i (\overline{x}_i - \overline{x})(\overline{x}_i - \overline{x})^T \quad and \quad S_w = \sum_{i=1}^{g} \sum_{j=1}^{N_i} (x_{i,j} - \overline{x}_i)(x_{i,j} - \overline{x}_i)^T \tag{10}$$

with N_i representing the number of training pattern from class 'i', $x_{i,j}$ is the n-dimensional pattern j from class 'i', g is the total number of classes. Each sample group 'i' has a class mean, which is denote as $\overline{x}_i$, where:

$$\overline{x}_i = \frac{1}{N_i} \sum_{j=1}^{N_i} x_{i,j}, \tag{11}$$

and the grand mean vector $\overline{x}$, equivalent to expression (1), is given by:

$$\overline{x} = \frac{1}{N} \sum_{i=1}^{g} N_i \overline{x}_i = \frac{1}{N} \sum_{i=1}^{g} \sum_{j=1}^{N_i} x_{i,j}, \tag{12}$$

where N is total number of samples, that is, $N = N_1 + N_2 + ... + N_g$.

It can be demonstrated that the Fisher criterion is maximized when the projection matrix W_{lda} is the solution of the eigensystem problem [18], [16]:

$$S_b W - S_w W \Lambda = 0. \tag{13}$$

So, by multiplying both sides of equation (13) by S_w^{-1}, we obtain:

$$S_w^{-1} S_b W - S_w^{-1} S_w W \Lambda = 0, \tag{14}$$

$$S_w^{-1} S_b W - W \Lambda = 0, \tag{15}$$

$$(S_w^{-1} S_b) W = W \Lambda. \tag{16}$$

and, consequently, W_{lda} is composed the $g-1$ eigenvectors of $S_w^{-1}S_b$ with nonzero eigenvalues [16]. In the case of a two-class problem, the LDA projection matrix is in fact the leading eigenvector w^{lda} of $S_w^{-1}S_b$.

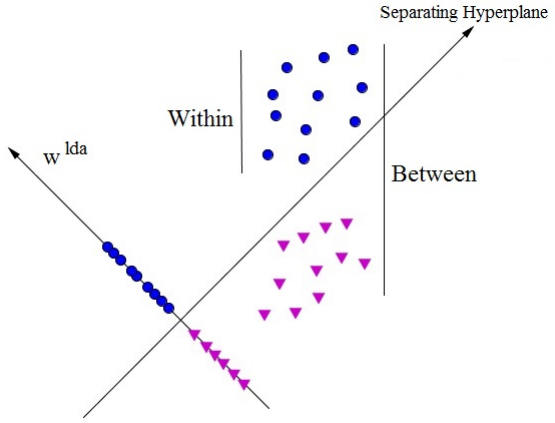


Figure 1. Representation of separating hyperplane generated by LDA

The matrix S_w may be singular when there is a limited number of total training observations N compared to the dimension of the feature space m, [18]. So, the performance of the standard LDA can be seriously degraded due to the fact that it is necessary to invert the S_w matrix to find the LDA subspace. In order to deal with such situations we have used a regularized version of the LDA approach called Maximum Uncertainty LDA (MLDA), [18] which replaces S_w by the matrix:

$$S_w^* = S_p^*(N-g) \tag{17}$$

with $S_p^* = \Phi \Lambda^* \Phi^T$, where Φ is composed by the eigenvectors of matrix:

$$S_p = \frac{S_w}{(N-g)}, \tag{18}$$

and the diagonal matrix Λ^* is formed by:

$$\Lambda^* = \begin{bmatrix} \max\{\lambda_1, \overline{\lambda}\} & 0 & \dots & 0 \\ 0 & \max\{\lambda_2, \overline{\lambda}\} & \dots & 0 \\ \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & \dots & \max\{\lambda_m, \overline{\lambda}\} \end{bmatrix} \tag{19}$$

where, each λ_i, is an eigenvalue of S_p and $\overline{\lambda}$ is computed by:

$$\overline{\lambda} = \frac{1}{m} \sum_{i=1}^{m} \lambda_i. \tag{20}$$

The output of the MLDA is the projection matrix W^{mlda} computed by replacing S_w with S_w^* in the Fisher criterion and by solving the corresponding optimization problem:

$$W^{mlda} = \arg\max_{W} \frac{|W^T S_b W|}{|W^T S_w^* W|} \tag{21}$$

The Algorithm 2, proposed in [18], summarizes the MLDA procedure.

Algorithm 2: Procedure for MLDA.

1: Find the Φ eigenvectors and Λ eigenvalues of S_p, where $S_p = \frac{S_w}{(N-g)}$
2: Calculate the average eigenvalue $\overline{\lambda}$ using expression (20);
3: Form a new matrix of eigenvalues through:
$\Lambda^* = diag[max\{\lambda_1, \overline{\lambda}\}, ..., max\{\lambda_n, \overline{\lambda}\}]$;
4: Compute the modified within-class scatter matrix

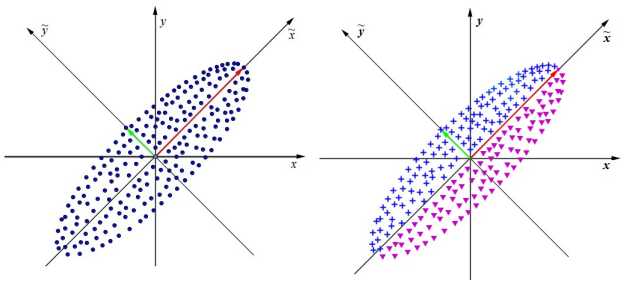
$$S_w^* = S_p^*(N-g) = (\Phi \Lambda^* \Phi^T)(N-g). \tag{22}$$

5: Compute W^{mlda} by solving the problem (IV).

V. PCA and Mult-Class Discriminant Analysis

Given a feature space, a key question is "how can we determine (or compute) the most important discriminant features for a pattern recognition task, like classification?" Discriminant analysis techniques address this question, which is very known in the context of PCA.

The Figure 2 is a simple example that pictures the limitation of PCA for discriminant features extraction. Both Figures 2.(a) and 2.(b) represent the same data set. Figure 2.(a) just shows the PCA directions ($\widetilde{\mathbf{x}}$ and $\widetilde{\mathbf{y}}$) and the distribution of the samples over the space. However, in Figure 2.(b) we distinguish two patterns: plus $(+)$ and triangle $(\blacktriangledown)$. We observe that the principal PCA direction $\widetilde{\mathbf{x}}$ can not discriminate samples of the considered groups because the projection of the data points over direction $\widetilde{\mathbf{x}}$ will mix the patterns in the corresponding one-dimensional subspace.



(a) (b)

Figure 2. (a) Scatter plot and PCA directions. (b) The same population but distinguishing patterns plus $(+)$ and triangle $(\blacktriangledown)$.

In general, Fisher's linear discriminant analysis (LDA) is used to identify the most important linear directions for separating sample groups rather than PCA [1]. This method, as well as the weighted pairwise variant of the well-known multi-class Fisher criterion introduced in [17] has the limitation of finding number of groups - 1 meaningful discriminant directions.

In [5] it is proposed the DPCA technique, based on the idea of using the discriminant weights obtained by separating hyperplanes to select among the principal components the most discriminant ones. In [7]

the DPCA was extended for discriminant principal components selection in multi-class problems. The so called multi-class discriminant principal components analysis (Multi-Class DPCA)[7] consists of the following steps: (a) apply PCA technique for dimensionality reduction in order to eliminate redundancy. (b) Compute a linear ensemble, based on the "one-against-all" SVM multi-class approach. (c) Combine the discriminant weights computed through the separating SVM hyperplanes in order to determine the discriminant contribution of each feature. So, given a N-class database, the step (b) builds N SVM machines in the PCA space. The step (c) is implemented by adapting an ensemble technique, the AdaBoost one [10], to yield a global discriminant vector. The proposed solution was evaluated in group separation tasks involving facial expression experiments and achieves higher recognition rates using less PCA features. However, the Multi-Class DPCA is not efficient for reconstruction. Also, the number of iterations (step (b) above) is equal to the number of classes in the main loop of Multi-Class DPCA. Such characteristic may limits the ability of the method to select discriminant features. These drawbacks have motivated the current work that is described next.

A. Technique overview

The whole Multi-Class.M2 DPCA methodology is schematized in Figure 3. We follow [7] and keep the application of PCA technique for dimensionality reduction in the step (1) of the pipeline. Then, in step (2), we compute a set of linear SVM hyperplanes, based on the "one-against-all" SVM multi-class approach. We also apply an ensemble technique, the AdaBoost.M2 algorithm, to combine the linear classifiers in order to compute the global discriminant vector. The key idea of this step is based on the fact that AdaBoost.M2 linearly combines weak classifiers to get the strong hypothesis. So, it is straightforward to obtain the global discriminant weights from the expression that defines the strong classifier by using a simple scheme, that corresponds to step (3) of Figure 3. This strategy can be also used to combine discriminant directions computed by other multi-class approaches, like linear discriminant analysis (LDA).

However, it is known that a strong learner like SVM does not work well as the base component for Adaboost [19]. Therefore, we follow [19] and implement a strategy to compute a weakened version of SVM that is useful as an and Adaboost.M2 component [20].

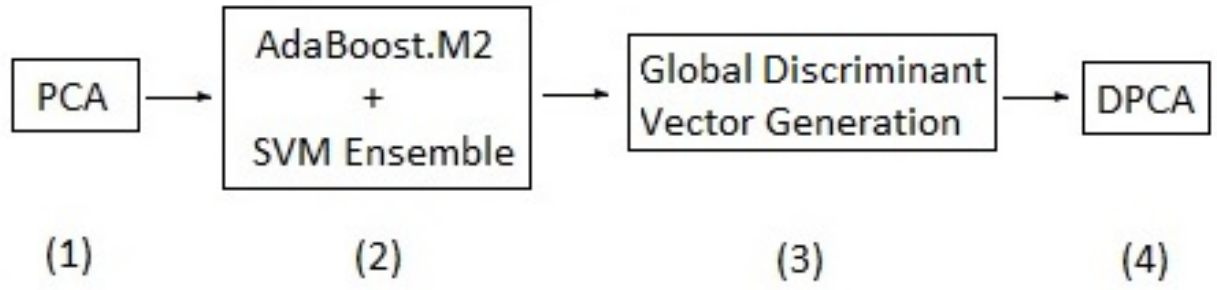


Figure 3. Flowchart with main steps of the proposed technique.

Finally, in the stage (4), we follow the traditional DPCA proposal and sort PCA components in the decreasing order of the global discriminant weights. The method is not restricted to any particular probability density function of the sample groups because it can be based on either a parametric or non-parametric separating hyperplane approaches.

VI. Technical background

The Multi-Class.M2 DPCA technique is based on the DPCA [5], the weakened SVM proposed in [19], the AdaBoost.M2 algorithm described in [20], and the nonseparable linear SVM [9]. Following, we describe the DPCA methodology.

Let the training observations $\mathbf{x}_i \in \Re^n$, $i = 1, \cdots, M$ that generates an $M \times n$ training set matrix $\widetilde{\Theta}$ centered respect to the global mean $\widehat{\mathbf{x}}$. Hence, the PCA algorithm computes a transformation matrix $P_{pca} = [\mathbf{p}_1, \mathbf{p}_2, ..., \mathbf{p}_{m'}]$ whose collumns $\mathbf{p}_i$, $i = 1, \ldots, m'$ minimize the mean square reconstruction error being the m' ($m' \leq n$) eigenvectors of the covariance Ω of $\widetilde{\Theta}$, which corresponds to the m' largest eigenvalues [12].

If to each training sample $\mathbf{x}_i$ it is associated a label $y_i \in \{-1, 1\}$, then we have a labeled training set:

$$X = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2) \ldots (\mathbf{x}_M, y_M)\}, \tag{23}$$

and, we can apply the DPCA technique to select the most discriminante principal components to separate sample groups.

The original DPCA is implemented taking as input a training set X, like in expression (23), to construct the separating hyperplane [5]. Firstly, for discarding redundancies, the PCA transformation matrix $P_{pca} = [\mathbf{p}_1, \mathbf{p}_2, ..., \mathbf{p}_{m'}]$ is computed and each zero mean data vector $\widetilde{\mathbf{x}}_i$ is projected generating a vector $\overline{\mathbf{x}}_i = (P_{pca})^T \widetilde{\mathbf{x}}_i$. Afterwards, the obtained $M \times m'$ data matrix and their corresponding labels are used as input to calculate the separating hyperplanes. In the following we focus on the SVM technique, although any other linear classifier could be used.

Since DPCA assumes only two classes to separate, there are only one discriminant vector $\phi_{svm} = (w_1, w_2, \cdots, w_{m'})$ given by the SVM hyperplane. The most discriminant feature of each one of the m'-dimensional vectors $\overline{\mathbf{x}}_i$ is obtained by multiplying the $M \times m'$ most expressive features matrix by the $m' \times 1$ discriminant SVM vector:

$$\begin{aligned} c_1 &= \overline{\mathbf{x}}_{11} w_1 + \overline{\mathbf{x}}_{12} w_2 + ... + \overline{\mathbf{x}}_{1m'} w_{m'}, \\ c_2 &= \overline{\mathbf{x}}_{21} w_1 + \overline{\mathbf{x}}_{22} w_2 + ... + \overline{\mathbf{x}}_{2m'} w_{m'}, \\ &\ldots \\ c_M &= \overline{\mathbf{x}}_{N1} w_1 + \overline{\mathbf{x}}_{N2} w_2 + ... + \overline{\mathbf{x}}_{Nm'} w_{m'}. \end{aligned} \tag{24}$$

We can determine the discriminant contribution of each feature by investigating the weights $[w_1, w_2, ..., w_{m'}]$. In fact, weights that are estimated to be 0 or approximately 0 have negligible contribution on the discriminant scores c_i described in equation (24), indicating that the corresponding features are not significant to separate the sample groups. In contrast, largest weights (in absolute values) indicate that the corresponding features contribute more to the discriminant score and consequently are important to characterize the differences between the groups.

Therefore, instead of sorting these features by selecting the corresponding principal components in decreasing order of eigenvalues, as PCA does, DPCA selects as the most important features for classification the ones with the highest discriminant weights, that is, $|w_1| \geq |w_2| \geq ... \geq |w_{m'}|$.

VII. Multi-Class.M2 discriminant analysis

The Multi-Class.M2 DPCA procedure is described by the Algorithm 3. At the input of the procedure, the training instances in the database $X \subset \Re^n$ are supposed independently and identically distributed from an uniform distribution D. Following the pipeline in Figure 3, the first stage of Multi-Class.M2 DPCA applies the PCA, for dimensionality reduction in order to eliminate redundancy (line 2).

The labeled projected data set is built in line 3 and composes the input to generate the weak SVM classifiers. In line 9 of Multi-Class.M2 DPCA algorithm, each weak learner generates an hypotheses, which has the form $h : X \times Y \rightarrow [0, 1]$, and can be interpreted as the probability that y is the correct label associated with instance $\mathbf{x}$. So, given a sample $\mathbf{x}_i$, the probability of choosing an incorrect label y is [20]: $Pr = \frac{1}{2}(1 - h(\mathbf{x}_i, y_i) + h(\mathbf{x}_i, y))$.

However, we have $|Y| - 1$ possibilities to obtain the incorrect answer. So, we can define the loss of the hypothesis through a weighted average according to some $q_{i,y}$, called the label weighting function, that assigns to each example i in the training set a load, with $\sum_{y \neq y_i} q_{i,y} = 1$. The resulting formula is called the pseudo-loss of h on training instance i with respect to q [20]:

$$ploss_q\left(h,i\right)=\frac{1}{2}\left(1-h(\mathbf{x}_i,y_i)+\sum_{y\neq y_i}q_{i,y}h(\mathbf{x}_i,y)\right). \tag{25}$$

So, following the AdaBoost.M2 strategy [20], in each iteration t of the Algorithm 3, the weak learner's goal is to minimize the expected pseudo-loss, computed in line 10 of the Algorithm 3, for a distribution D^t and weighting function q^t. The algorithm uses a second weight vector whose values at time t are denoted by $w_{i,y}^t$, $i=1,\cdots,M$, $y\in Y-\{y_i\}$, which is initialized in line 1, based on the initial distribution D. The main loop of the algorithm aims to update these weights in order to minimize the expected pseudo-loss. So, the weighting function q^t and the distribution D^t are computed using the $w_{i,y}^t$ (line 5 of procedure 3).

Next, the Multi-Class.M2 DPCA computes a set of SVM hyperplanes, based on the one-against-all SVM multi-class approach presented in [14]. Hence, as we have N classes, the internal loop in the Algorithm 3 (line 6 to 9) constructs N weakened SVMs, in the PCA subspace, using the Algorithm 4. To do this, in line 7 of Algorithm 3 we buid the $\overline{\Theta}^y$ set by taking all k_y projected samples from class y and label them as 1. Then, using random sampling we choose $(2k_y)/(N-1)$ projected samples from classes other than y and label them as -1. The obtained set of feature vectors $\overline{\mathbf{x}}_m^y\in\Re^{m'}$ and corresponding labels $y_m\in\{-1,1\}$:

$$\overline{\Theta}^y=\left\{\left(\overline{\mathbf{x}}_1^y,l_1\right),\left(\overline{\mathbf{x}}_2^y,l_2\right),...\left(\overline{\mathbf{x}}_{3k_y}^y,l_{3k_y}\right)\right\}, \tag{26}$$

are the input to call the Algorithm 4 which construct the weak SVM (WSVM) model y, represented by a hyperplane direction $\left(\phi_y^t\right)$ and a linear coefficient $\left(b_y^t\right)$. Each hypothesis h^t, in line 9 of Algorithm 3, is generated through a WSVM and the following normalization function:

$$f\left(z\right)=\frac{z-z_{min,y}^t}{z_{max,y}^t-z_{min,y}^t}, \tag{27}$$

where $f:[z_{min,y}^t,z_{max,y}^t]\rightarrow[0,1]$, with $z_{min,y}^t$ and $z_{max,y}^t$ being the minimum and maximum values, respectively, of the set $\left\{<x_i,\phi_y^t>+b_y^t,\quad i=1,2,\ldots,M\right\}$.

The lines 16-18 of the Algorithm 3 are based on the AdaBoost.M2 idea of deriving a strong learner h_f by using the linear combination of weak (WSVM, in our case) learners $h^1,h^2,\cdots,h^T$:

$$h_f\left(\mathbf{x}\right)=\arg\max_{y\in Y}\sum_{t=1}^{T}\widetilde{\alpha}^t h^t(\mathbf{x},y), \tag{29}$$

where $\widetilde{\alpha}^t$ is computed in line 16. This expression offers the possibility of extending the DPCA methodology to multi-class problems using the Adaboost.M2 result. To see this, we shall remember that $h^t(x,y)$ in line 9 is computed through the function f, in expression (27), and rewrite expression (29) as:

$$h_f\left(\mathbf{x}\right)=\arg\max_{y\in Y}\left[\sum_{t=1}^{T}\widetilde{\alpha}^t f\left(<\mathbf{x},\phi_y^t>+b_y^t\right)\right]. \tag{30}$$

But, from equation (27), we get:

$$f\left(<\mathbf{x},\phi_y^t>+b_y^t\right)=\frac{<\mathbf{x},\phi_y^t>+b_y^t-z_{min,y}^t}{z_{max,y}^t-z_{min,y}^t}. \tag{31}$$

Therefore, by substituting this expression into equation (30), and using the linearity of the inner product, we can show that:

$$h_f\left(\mathbf{x}\right)=\arg\max_{y\in Y}\left[<\mathbf{x},\Phi_y>+\psi_y\right], \tag{32}$$

where:

$$\Phi_y=\sum_{t=1}^{T}\widetilde{\alpha}^t\frac{\phi_y^t}{z_{max,y}^t-z_{min,y}^t},\psi_y=\sum_{t=1}^{T}\widetilde{\alpha}^t\frac{\left(b_y^t-z_{min,y}^t\right)}{z_{max,y}^t-z_{min,y}^t},$$

Algorithm 3: Multi-Class.M2 DPCA Procedure

Input: Samples: $X = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2) \dots (\mathbf{x}_M, y_M)\}$; where $y_i \in Y$ and $Y = \{1, 2, 3, .., N\}$; Distribution D over the M examples; Percentage μ;

Initialize the weight vector: $w^1_{i,y} = \frac{D(i)}{|Y|-1}$, $for \quad i = 1, \cdots, M$; $y \in Y - \{y_i\}$

Calculate P_{pca} and the projected data $\overline{\mathbf{x}}_i = (P_{pca})^T \widetilde{\mathbf{x}}_i$ where $\widetilde{\mathbf{x}}_i = x_i - \widehat{x}$, such as, $\widehat{x} = \frac{1}{M} \sum_{i=1}^{M} x_i$

Build the labeled projected data set $\overline{\Theta} = \{(\overline{\mathbf{x}}_1, y_1), (\overline{\mathbf{x}}_2, y_2) \dots (\overline{\mathbf{x}}_M, y_M)\}$

for $t = 1, ...$ ***to*** T **do**

  for $y \neq y_i$: $q^t_{i,y} = \frac{w^t_{i,y}}{W^t_i}$; and set $D^t(i) = \frac{W^t_i}{\sum_{i=1}^{N} W^t_i}$

  for $y = 1, ...$ ***to*** N **do**

    Build the subset $\overline{\Theta}^y$, given by expression (26);

    $(\phi^t_y, b^t_y) = WSVM(\overline{\Theta}^y, \mathcal{Y}, D^t, \mu)$ where $\mathcal{Y} = \{-1, 1\}$;

    Get hypothesis $h^t : X \times Y \longrightarrow [0, 1]$, given by $h^t(\mathbf{x}, y) = f\left(< \mathbf{x}, \phi^t_y > + b^t_y\right)$

  Compute:

$$e^t = \frac{1}{2} \sum_{i=1}^{N} D^t(i) \left(1 - h^t(\mathbf{x}_i, y_i) + \sum_{y \neq y_i} q^t_{i,y} h^t(\mathbf{x}_i, y)\right)$$

  if $e_t > 0.5$ **then**

    break;

  Calculate AdaBoost.M2 weights: $\alpha^t = \frac{1}{2} \ln\left(\frac{1-e^t}{e^t}\right)$;

  for $i = 1, ...N$ *and* $y \in Y - \{y_i\}$ **do**

    Update: $w^{t+1}_{i,y} = w^t_{i,y} exp(-\alpha^t(1 - h^t(x_i, y_i) + h^t(x_i, y)))$;

Normalize $\widetilde{\alpha}^t = \alpha^t / \sum_{j=1}^{T} \alpha^j$, $t = 1, 2, \dots, T$

for $i = 1, ...$ ***to*** m' **do**

$$|\Phi_{i,y}| = \left| \sum_{t=1}^{T} \widetilde{\alpha}^t \frac{\phi^t_{i,y}}{z^t_{max,y} - z^t_{min,y}} \right|, y \in Y \tag{28}$$

Compute $v(i) = \max_{y \in Y} \{|\Phi_{i,y}|\}$, $i = 1, 2, \dots, m'$

Sort discriminant weights: $v(1) \geq v(2) \geq \dots v(m')$

Select the principal components following $v(i)$

Output: Discriminant principal components: $\mathbf{q}_1, \mathbf{q}_2, \cdots, \mathbf{q}_{m'}$

Algorithm 4: WSVM Procedure: Build a Weakened version of SVM.

Input: Labeled samples:$X = \{(\mathbf{x}_i, y_i), i = 1, 2, \cdots, n'\}$ where $y_i \in Y$ is the label of the sample $\mathbf{x}_i$; Samples probability distribution $D(\mathbf{x}_i)$;
Percentage μ;
Select $\mathcal{J}$ so that, $\sum_{j \in \mathcal{J}} D(x_j) \leq (1 - \mu)$;
Select $(\mathbf{x}_i, y_i); i \in \mathcal{J}$, and define $D^* = D_{\mathcal{J}}$;
Compute the weighted data $X^* = \{(D^*_i \cdot \mathbf{x}_i, y_i), i \in \mathcal{J}\}$
Compute the (weak) SVM hyperplane ϕ_{svm} using X^*;
Output: WSVM hyperplane ϕ_{svm}, b.

with $\Phi_y \in \mathbb{R}^{m'}$ and $\psi_y \in \mathbb{R}$. The bias ψ_y can be incorporated in the inner product through a translation $\overline{T}_y$ such that $< \overline{T}_y, \Phi_y >= \psi_y$, which renders:

$$h_f(x) = \arg\max_{y \in Y} \left[\sum_{i=1}^{n} \left(x_i + \overline{T}_{i,y} \right) \Phi_{i,y} \right]. \tag{33}$$

This expression is the key to generalize the DPCA technique for multi-class problems. Specifically, each feature i has a vector of weights $\Phi_{i,y}$ with size $|Y|$. So, for each feature i we need to seek for the most important weight, in absolute value $|\Phi_{i,y_i}|$, which can be interpreted as a measure of the discriminant contribution of the corresponding feature. These values are used to generate the vector $\mathbf{v}$ in line 19 of Algorithm 3. Next, we shall sort the obtained array in decreasing order, as performed in line 20 of the Algorithm 3, to get the global discriminant weights. The output of the Multi-Class.M2 DPCA procedure is the discriminant principal components $\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_{m'}$ where $\mathbf{q}_i$ is a PCA component selected according to its discriminant weight $v(i)$.

On the other hand, if we compute the LDA in the PCA space, we get $N-1$ hyperplane directions $\phi^i_{lda} \in \mathbb{R}^{m'}$, $i = 1, 2, \ldots (N-1)$. Consequently, we obtain in this case a LDA weight matrix $\phi^{i,j}_{lda}$, which can be processed according to lines 19-20 of Algorithm 3, by just replacing $\Phi_{i,y}$ by $\phi^{i,j}_{lda}$. The obtained global discriminant weights are named Multi-Class LDA-DPCA in the following sections.

We also aim to study the influence of the denominator $z^t_{max,y} - z^t_{min,y}$ in expression (28) of line 18 of Multi-Class.M2 DPCA algorithm . To perform this task, we test a version of the Algorithm 3 with equation (28) replaced by $|\Phi_{i,y}| = \left| \sum_{t=1}^{T} \widetilde{\alpha}^t \phi^t_{i,y} \right|$. We call the obtained (non normalized) algorithm as the Multi-Class.M2 DPCA-NN.

VIII. Computational Experiments

In this section we perform facial expression experiments using the Radboud (RaFD) [21] and the Japanese Female Facial Expression (JAFFE) image databases [22].

In order to save memory allocation along the Algorithm 3 execution, we convert each pose to gray scale and resize it to 50×50 before computation. To compare the Multi-Class.M2 DPCA not only to the standard PCA but also to other methods, we consider the Multi-Class DPCA [7], as well as the Multi-Class LDA-DPCA and Multi-Class.M2 DPCA-NN, both explained in the last paragraph of section VII. For evaluation of the discriminant principal components, the following separation tasks have been performed using frontal face images of the mentioned databases.

- **Three-Class** experiment: neutral, happiness, and sad samples;
- **Five-Class** experiment: neutral, happiness, sad, fear, and anger classes.

The recognition tasks experiments are carried out using the full rank PCA subspace with all non-zero eigenvalues. In these experiments we have assumed equal prior probabilities and misclassification costs for all the classes. On the PCA subspace, the mean of each class i has been calculated from the corresponding training images and the Mahalanobis distance from each class mean $\widehat{\mathbf{x}}_i$ has been used to assign a test observation $\mathbf{x}_r$ to either the different facial expressions. That is, we have assigned $\mathbf{x}_r$ to class i that minimizes:

$$d_i(\mathbf{x}_r) = \sum_{j=1}^{k} \frac{1}{\lambda_j} (x_{rj} - \widehat{x}_{ij})^2, \tag{34}$$

where λ_j is the corresponding eigenvalue, k is the number of principal components retained, x_{rj} and $\widehat{x}_{ij}$ are the projections of the sample $\mathbf{x}_r$ and of the mean $\widehat{\mathbf{x}}_i$, respectively, in the jth component considered.

The Figure 4 shows the average recognition rates of the 10-fold cross validation experiments for PCA and the discriminant techniques for the three and five-class classification problems above mentioned. When

analysing the Figures 4.(a)-(d) we notice that the Multi-Class.M2 DPCA and Multi-Class.M2 DPCA-NN achieve highest recognition rates or perform closer to the best one. Specifically, let us highlight the intervals where the highest recognition rates are not achieved by Multi-Class.M2 DPCA or Multi-Class.M2 DPCA-NN. This happens in the Figure 4.(a), for $25 < k < 29$, where the Multi-Class DPCA is the best technique. Also, in Figure 4.(b), the Multi-Class LDA-DPCA outperforms both Multi-Class.M2 DPCA and Multi-Class.M2 DPCA-NN in the range $10 \leq k \leq 17$. For $k > 110$ all the methods, except the Multi-Class DPCA, achieves the same accuracy. The Figure 4.(d) shows that in the range $8 \leq k \leq 20$ the Multi-Class LDA-DPCA technique is the best method. However, in all these cases, if we take the absolute value of the difference between the minimum classification rate obtained by Multi-Class.M2 DPCA and Multi-Class.M2 DPCA-NN and the maximum accuracy of the other ones, we get the values $1,4\%$ for $k = 17$ and 3% for $k = 10$, for Figure 4.(b) and Figure 4.(d) respectively , which are not an expressive values.

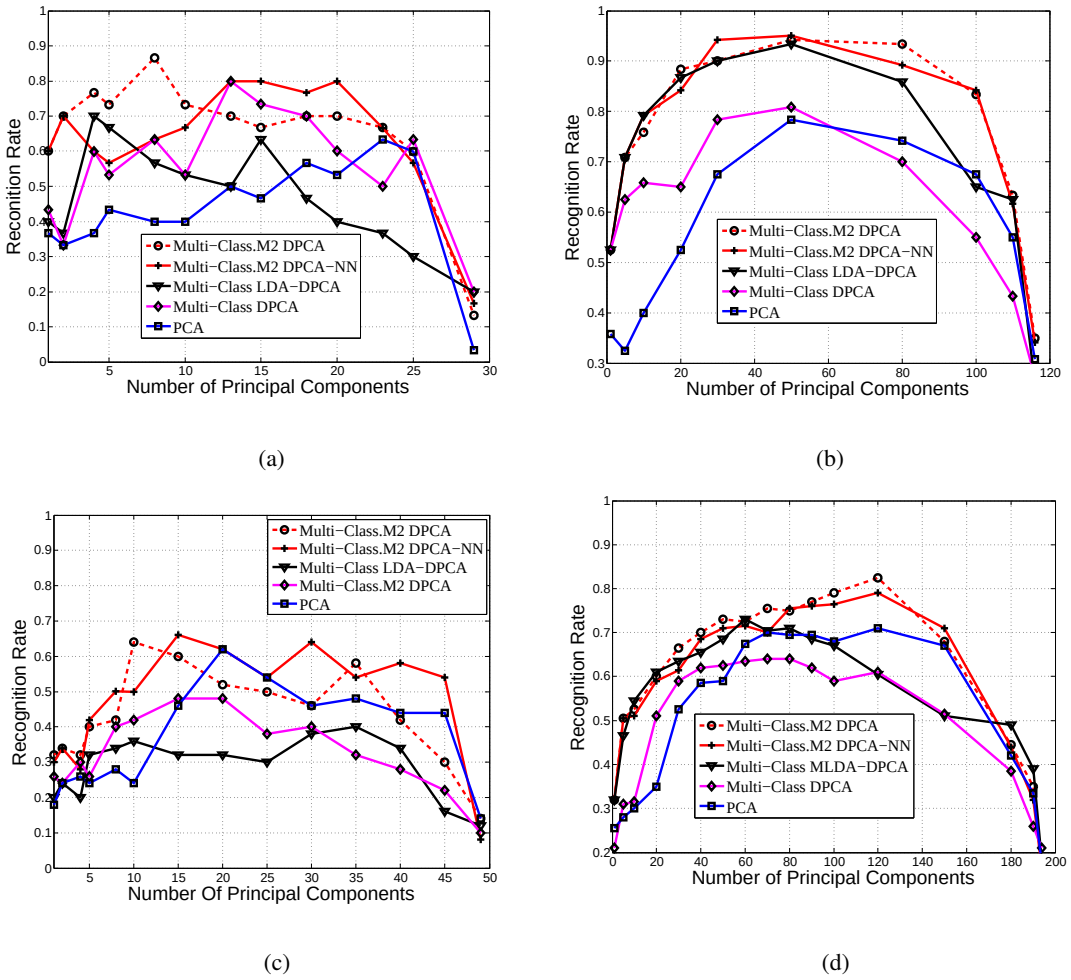


Figure 4. Expression experiments using using the JAFFE and Radboud databases. Average recognition rates of PCA components selected by the focused techniques:(a) Three-class tasks with JAFFE. (b) Three-class experiments using Radboud. (c) Five-class tasks with JAFFE. (d) Five-class experiments using Radboud.

Moreover, although Multi-Class LDA-DPCA and PCA classification rates are equal to the highest accuracy in some intervals of Figures 4.(b)-(d), we shall notice that the maxima of the recognition rates are obtained by Multi-Class.M2 DPCA (Figure 4.(a): 86% in $k = 8$; Figure 4.(d): 82% in $k = 120$) and Multi-Class.M2 DPCA-NN (Figure 4.(b): 94% in $k = 30$; Figure 4.(c): 66% in $k = 15$). These values offers an objective way to compare Multi-Class.M2 DPCA and Multi-Class.M2 DPCA-NN. In fact, the Figure 4.(a) shows that the Multi-Class.M2 DPCA obtain the maximum recognition rate using just 8 of the principal PCA components. In the case of Figure 4.(d), the maximum is also obtained by Multi-Class.M2 DPCA. In the Figure 4.(b) the Multi-Class.M2 DPCA-NN gets the maximum recognition rate with $k = 30$.

Finally, the Figure 4.(c) shows that the maximum (66%) is achieved by Multi-Class.M2 DPCA-NN with 15 PCA componentes. However, we shall observe also that Multi-Class.M2 DPCA gets an accuracy very close to the maximum (64%) but using just 10 components. These facts indicates a slight superiority of Multi-Class.M2 DPCA subspaces against the Multi-Class.M2 DPCA-NN ones for expression recognition tasks. Also, it is important to notice in Figures 4.(a)-(d) a degradation in the accuracy of all techniques for higher subspace dimensions, probably due to overfitting.

The tables I-IV lists the 10 principal components with the highest discriminant weights given by the Multi-Class.M2 DPCA algorithm and the counterparts, for discriminating the expression samples. When considering the three-class experiment with the Radboud database, we can observe that all the considered techniques have selected some distant PCA components among the first 10 most discriminant principal components. In the specific case of the of Multi-Class.M2 DPCA, it selected the $42th$, $38th$ and $81th$ PCA components among its first 10 discriminant principal components. Since principal components with lower variances describe particular information related to few samples, these results confirm the ability of Multi-Class.M2 DPCA of zooming into the details of group differences. However, components with lower variances should count less for the global reconstruction than PCA components with higher variances. We expect some consequences of this fact in the reconstruction experiments, as we will see next.

Multi-Class.M2 DPCA	15	17	21	10	11	12	13	19	18	9
Multi-Class.M2 DPCA-NN	15	17	13	12	18	2	21	9	19	5
Multi-Class MLDA DPCA	21	17	15	28	29	18	19	26	12	24
Multi-Class DPCA	11	7	10	21	12	5	8	9	1	22

Table I
TOP 10 (FROM TOP TO BOTTOM AND LEFT TO RIGHT) DISCRIMINANT PRINCIPAL COMPONENTS, RANKED BY THE DISCRIMINANT TECHNIQUES, USING THE JAFFE DATABASE FOR THREE-CLASS TASKS.

Multi-Class.M2 DPCA	24	36	26	32	20	38	81	42	29	9
Multi-Class.M2 DPCA-NN	24	26	36	20	32	8	9	38	17	19
Multi-Class MLDA DPCA	24	26	20	71	81	36	56	19	59	21
Multi-Class DPCA	24	26	20	22	19	17	34	40	8	44

Table II
TOP 10 (FROM TOP TO BOTTOM AND LEFT TO RIGHT) DISCRIMINANT PRINCIPAL COMPONENTS, RANKED BY THE DISCRIMINANT TECHNIQUES, USING THE RADBOUD DATABASE FOR THREE-CLASS TASKS.

Multi-Class.M2 DPCA	11	6	13	31	19	14	12	9	16	18
Multi-Class.M2 DPCA-NN	6	11	13	12	18	16	9	10	31	14
Multi-Class MLDA DPCA	44	31	48	30	19	32	39	43	41	28
Multi-Class DPCA	4	7	8	3	2	39	16	18	6	11

Table III
TOP 10 (FROM TOP TO BOTTOM AND LEFT TO RIGHT) DISCRIMINANT PRINCIPAL COMPONENTS, RANKED BY THE DISCRIMINANT TECHNIQUES, USING THE JAFFE DATABASE FOR FIVE-CLASS TASKS.

Multi-Class.M2 DPCA	27	30	25	26	1	31	9	57	34	20
Multi-Class.M2 DPCA-NN	27	30	25	26	1	9	5	2	34	31
Multi-Class MLDA DPCA	27	30	25	26	31	61	86	57	70	81
Multi-Class DPCA	6	11	7	33	2	10	57	23	20	4

Table IV
TOP 10 (FROM TOP TO BOTTOM AND LEFT TO RIGHT) DISCRIMINANT PRINCIPAL COMPONENTS, RANKED BY THE DISCRIMINANT TECHNIQUES, USING THE RADBOUD DATABASE FOR FIVE-CLASS TASKS.

To understand the changes described by the principal components for the three-class and five-class separation task with the Radboud data set, we reconstruct the most expressive features by varying each principal component $\mathbf{p}_i$ separately using the equation:

$$I = \widehat{\mathbf{x}} + \beta \cdot \mathbf{p}_i, \tag{35}$$

where $\widehat{\mathbf{x}}$ is the global mean, $\beta \in \{\pm j \cdot \bar{\lambda}^{0.5}, \quad j = 0, \pm 3\}$, and $\bar{\lambda}$ is the average eigenvalue of the total covariance matrix of PCA. We choose $\bar{\lambda}$ instead of λ_i because some λ_i can be very small (or big) in this case, showing no changes (or color saturation) between the samples when we move along the corresponding principal components.

Figure 5 illustrates the transformations on the forth PCA most expressive component contrasted with the forth discriminant principal component selected by the discriminant techniques to separate facial expressions. In Figure 6 we apply the same process but it was used the sixth PCA most expressive component contrasted with the sixth discriminant principal component selected by the discriminant techniques to separate facial expressions.

In Figures 5.(m)-(o), it can be seen that the forth PCA most expressive direction captures essentially the changes in gender, which are the major variations of all the training samples. Owing to the fact that changes in facial expression are much less significant than the gender ones, the standard PCA is unable to capture such minor variations in its most expressive components. However, when we compare these results with the ones reconstructed by the forth most discriminant principal component selected by the other techniques, illustrated by Figures 5.(a)-(l), we can see that more distant principal component (see Table II) carry more information about expression variations then the first PCA ones. That is why the discriminant methods achieves, in general, higher classification rates than PCA. On the other hand, we can notice in Figure 5.(j)-(l) that the Multi-Class.M2 DPCA and Multi-Class.M2 DPCA-NN forth discriminant principal component capture more clearly the facial expression with less artifacts in the reconstruction then the other discriminant techniques which agrees with the observed superiority of Multi-Class.M2 DPCA and Multi-Class.M2 DPCA-NN in the recognition experiments.

In Figure 6.(a)-(l), we also can see that the sixth discriminant principal component selected by discriminants techniques carry more information about expression variations then the first PCA ones. Moreover, as also observed in Figure 5.(m)-(o), PCA subspace presents changes in gender that can be seen in Figure 6.(m)-(o).

Now, it is worthwhile to consider the accumulated variance, for Radboud experiment, explained by the selected subspaces which is computed by:

$$Vacc^{l,i}(k) = \frac{\sum_{j=1}^{k} \lambda_j^l}{\sum_{j=1}^{m'} \lambda_j^l}, \tag{36}$$

where $i \in \{3, 5\}$, $l \in \{1, 2, 3, 4, 5\}$ with $l = 1$ corresponds to Multi-Class.M2 DPCA, $l = 2$ to Multi-Class.M2 DPCA-NN, $l = 3$ to Multi-Class LDA-DPCA, and $l = 4, 5$ correspond to Multi-Class DPCA and PCA, respectively. Also, λ_j^l is the variance associated to the jth component selected by each the discriminant techniques l. The expression (36) is important for this discussions because we expect some correlation between the performance for reconstruction and the accumulated variance in expression (36)

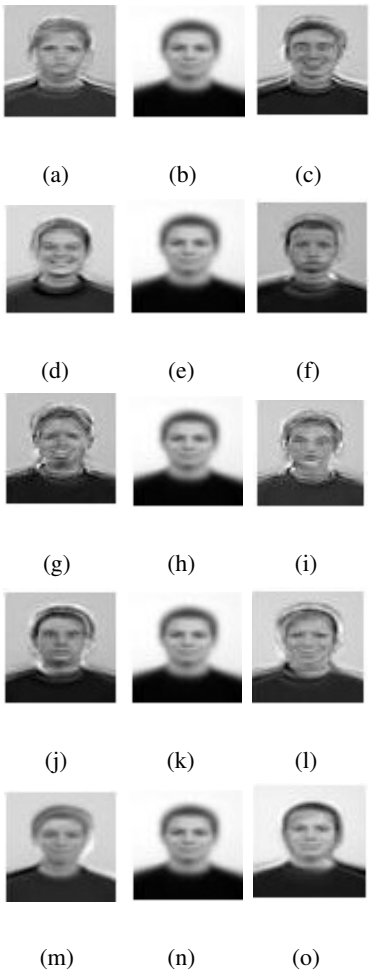


Figure 5. Visualization of the changes described by the forth principal direction, using the Radboud face database for three-class experiments, selected by: (a)-(c) Multi-Class.M2 DPCA. (d)-(f) Multi-Class.M2 DPCA-NN. (g)-(i) Multi-Class LDA-DPCA. (j)-(l) Multi-Class DPCA. (m)-(o) PCA.

due to the known fact that the components with larger variances keep global information related to features that most vary in the samples [6]. In Figure 7 we plot the result of expression (36).

We observe that the Multi-Class LDA-DPCA gives the lowest results for the $Vacc$ while PCA technique gives the largest accumulated variances for both three and five-class JAFFE and Radboud experiments. The $Vacc$ of the other considered techniques fall between Multi-Class LDA-DPCA and PCA in all Figures 7.(a)-(d). Although PCA gives the largest values for $Vacc$ its recognition rates are, in general, outperformed by the discriminant principal components. Since PCA explains features that most vary in the samples the principal subspaces do not necessarily represent important discriminant directions to separate sample groups. However, the reconstruction results are expected to give lower errors if we take components with higher variances, like PCA does. Besides, in the case of five-classes, we expect a better reconstruction performance for Multi-Class.M2 DPCA and Multi-Class.M2 DPCA-NN once their accumulated variances are closer to the PCA ones in this case. To make clear these observations, let us quantify the reconstruction quality through the root mean squared error (RMSE), computed as follows:

$$RMSE^{l,i}(k) = \sqrt{\frac{\sum_{j=1}^{N} ||P.I_k^l.P^T x_j - x_j||^2}{N}}, \tag{37}$$

where the index l and i follows the same map used in expression (36), I_k^l is a truncated identity matrix that keeps the selected subspace with dimension k, and $\|\cdot\|$ is the usual 2-norm.

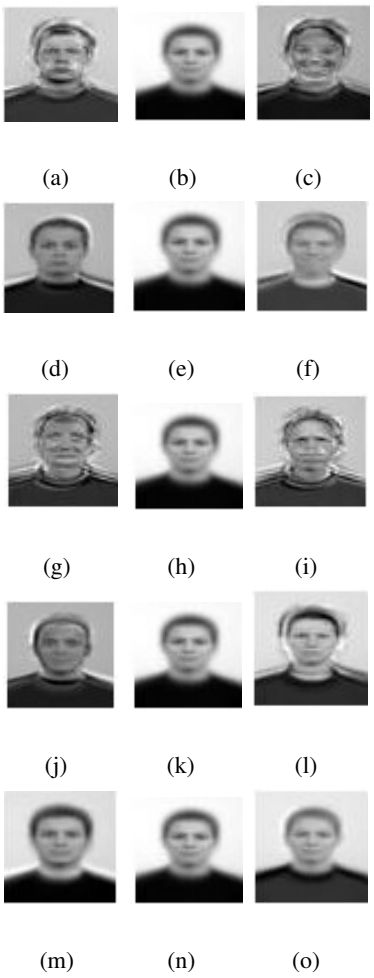
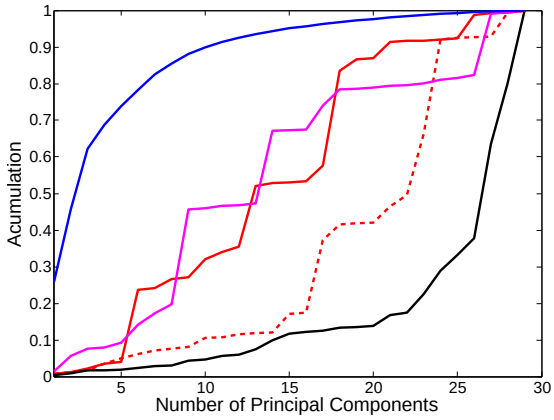


Figure 6. Visualization of the changes described by the sixth principal direction, using the Radboud face database for three-class experiments, selected by: (a)-(c) Multi-Class.M2 DPCA. (d)-(f) Multi-Class.M2 DPCA-NN. (g)-(i) Multi-Class LDA-DPCA. (j)-(l) Multi-Class DPCA. (m)-(o) PCA.

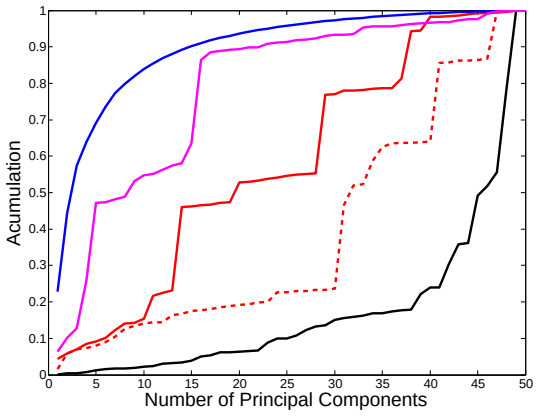
The Figure 8 shows the RMSE for the reconstruction process for the subspaces given by the focused techniques. It is noticeable that for all experiments, PCA reconstruction performs equal or better than the multi-class discriminant components for all the simulated values of k. This observation agrees with the accumulated variance reported in Figure 7 which is such that $Vacc^{5,i}(k) \geq Vacc^{l,i}(k)$ for all values of k, l and $i = 3, 5$. Therefore, while in the classification tasks the PCA method is, in general, outperformed by the Multi-Class DPCA method, in the reconstruction experiments the PCA subspaces become more efficient for almost all the simulated values.

Let us compare the reconstruction performance of the discriminant aproaches. In Figure 8(a)-(b), $RMSE^{3,3}(k)$, it is, the RMSE for Multi-Class LDA-DPCA is the larger one. Figure 8(a) for $1 \leq k \leq 17$, $RMSE^{4,3}(k)$ is smaller than other discriminant techniques. So, for $17 < k \leq 29$, an inversion occurs, $RMSE^{2,3}(k)$ it becomes the smallest or equal the all. The same performace can be see in Figure 8(b) but in different interval. In this case,$1 \leq k \leq 39$, $RMSE^{4,3}(k)$ is smaller than other discriminant techniques. In $39 \leq k \leq 49$, $RMSE^{2,3}(k)$ it becomes the smallest or equal the all.

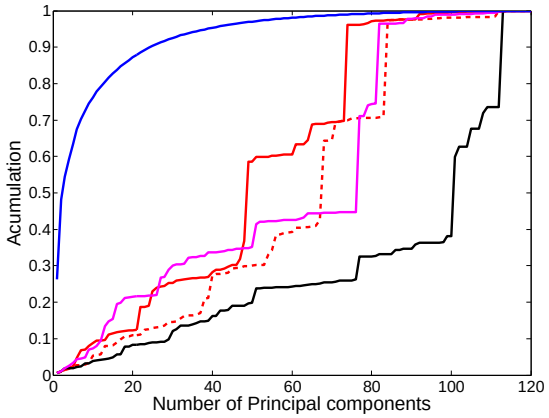
From Figure 8.(c), for $1 \leq k \leq 5$ the discriminant techniques performs equal to each other. For $5 \leq k \leq 116$ the RMSE for Multi-Class LDA-DPCA is the larger one. The other discriminant technique performs as: (a) For $5 \leq k \leq 42$, $RMSE^{1,3}(k) > RMSE^{2,3}(k) > RMSE^{4,3}(k)$; (b) For $42 \leq k \leq 116$, $RMSE^{1,3}(k) \geq RMSE^{4,3}(k) \geq RMSE^{2,3}(k)$. Therefore, the Multi-Class.M2 DPCA is worst or equal the Multi-Class DPCA and Multi-Class.M2 DPCA-NN in terms of RMSE results. On the other hand, the Multi-Class.M2 DPCA-NN performs better than the Multi-Class DPCA and Multi-Class.M2 DPCA for



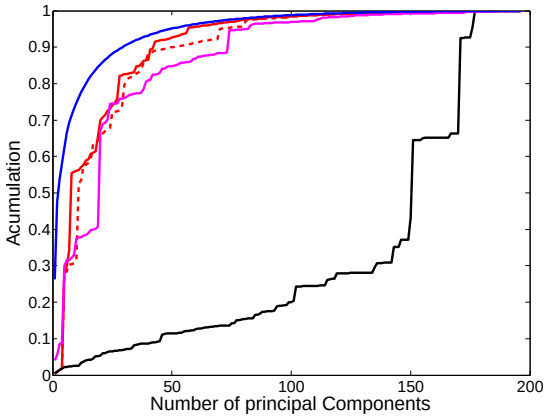
(a) 3 classes(JAFFE)



(b) 5 classes(JAFFE)



(c) 3 classes(Radboud)

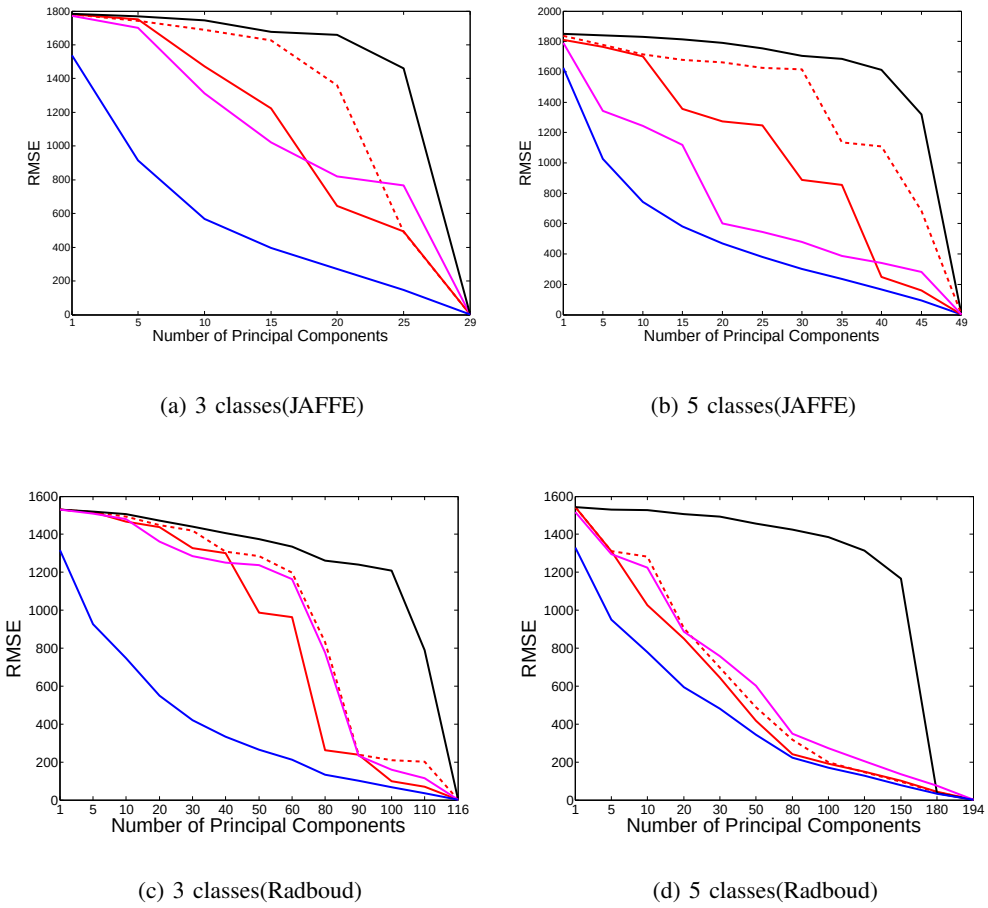


(d) 5 classes(radboud)

Figure 7. Total variance computed by expression (36), explained by discriminant principal components selected by Multi-Class.M2-DPCA (dashed red line); Multi-Class.M2-DPCA NN (solid red line); Multi-Class-LDA-DPCA (black line); Multi-Class DPCA (magenta line); and PCA (blue line).

$42 \leq k \leq 116$. This observations are in accordance with the corresponding accumulated variances.

In the case of the RMSE for the five-class experiments with the Radboud, shown in Figure 8.(d), we observe an analogous behaviour for PCA and Multi-Class LDA-DPCA for $1 \leq k \leq 180$. The Multi-Class.M2 DPCA-NN is better than Multi-Class DPCA for $5 < k < 194$. Besides $RMSE^{2,5}(k) \leq RMSE^{1,5}(k)$ for all the subspace dimensions and $RMSE^{1,5}(k) \geqslant RMSE^{4,5}(k)$ in $5 \leq k \leq 21$ and $RMSE^{1,5}(k) \leq RMSE^{4,5}(k)$ in $21 \leq k \leq 194$.



(a) 3 classes(JAFFE)

(b) 5 classes(JAFFE)

(c) 3 classes(Radboud)

(d) 5 classes(Radboud)

Figure 8. RMSE computed by equation 37 for: Multi-Class.M2 DPCA (dash line); Multi-Class.M2-DPCA-NN (solid red line); Multi-Class LDA-DPCA (black line); Multi-Class DPCA (magenta line); and PCA (blue line).

IX. Conclusion and Future Works

This paper introduces the Multi-Class.M2 DPCA and Multi-Class.M2 DPCA-NN algorithms for ranking PCA components computed from multi-class facial expression databases. The basic methodology has a computational complexity dominated by the AdaBoost.M2 algorithm plus PCA computation. The facial expressions experiments show that, in general, the PCA components selected by Multi-Class.M2 DPCA and Multi-Class.M2 DPCA-NN allow higher recognition rates using less linear features than Multi-Class LDA-DPCA, Multi-Class DPCA and the standard PCA.

Further work is being undertaken to test the algorithm for more than 5 classes as well as with other applications. We shall replace the AdaBoost.M2 technique by the bagging one [10] as a direction to improve the classification performance when increasing the number of classes. Moreover, we need to improve its reconstruction in low dimensional subspaces.

References

[1] T. Hastie, R. Tibshirani, and J. Friedman, "The elements of statistical learning," *Springer*, 2001.

[2] G. A. Giraldi, P. S. Rodrigues, E. C. Kitani, and C. E. Thomaz, "Dimensionality reduction, classification and reconstruction problems in statistical learning approaches," *Revista de Informatica Teorica e Aplicada (RITA)*, vol. 15, no. 1, pp. 141–173, 2008.

[3] J. P. Cunningham and Z. Ghahramani, "Linear dimensionality reduction: Survey, insights, and generalizations," *J. of Mach. Learn. Research*, vol. 16, pp. 2859–2900, 2015.

[4] H. Safavi and C.-I. Chang, "Projection pursuit-based dimensionality reduction," *Proc. SPIE*, vol. 6966, pp. 69 661H–69 661H–11, 2008.

[5] C. E. Thomaz and G. A. Giraldi, "A new ranking method for principal components analysis and its application to face image analysis," *Image Vision Comput.*, vol. 28, no. 6, pp. 902–913, June 2010.

[6] D. Swets and J. Weng, "Using discriminants eigenfeatures for image retrieval," *IEEE Trans. Patterns Anal. Mach Intell.*, vol. 18(8), pp. 831–836, 1996.

[7] T. Filisbino, D. Leite, G. Giraldi, and C. Thomaz, "Multi-class discriminant analysis based on svm ensembles for ranking principal components," in *36th Ibero-Latin Am. Cong. on Comp. Meth. in Eng. (CILAMCE)*, Nov 2015.

[8] M. Zhu and A. M. Martinez, "Selecting principal components in a two-stage lda algorithm," in *CVPR'06*, June 2006, pp. 132–137.

[9] V. N. Vapnik, *Statistical Learning Theory*. John Wiley & Sons, INC., 1998.

[10] Z.-H. Zhou, *Ensemble Methods: Foundations and Algorithms*, 1st ed. Chapman & Hall/CRC, 2012.

[11] K. Fukunaga, *Introduction to Statistical Pattern Recognition*. New York: Academic Press, 1990.

[12] M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of Cognitive Neuroscience*, vol. 3, pp. 71–86, 1991.

[13] C. Davatzikos, "Why voxel-based morphometric analysis should be used with great caution when characterizing group differences," *NeuroImage*, vol. 23, pp. 17–20, 2004.

[14] E. Yildizer, A. M. Balci, M. Hassan, and R. Alhajj, "Efficient content-based image retrieval using multiple support vector machines ensemble," *Expert Syst. Appl.*, vol. 39, pp. 2385–2396, Feb. 2012.

[15] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 121–167, 1998.

[16] P. Devijver and J. Kittler, *Pattern Classification: A Statistical Approach*. Prentice-Hall, 1982.

[17] M. Loog, R. P. W. Duin, and R. Haeb-Umbach, "Multiclass linear dimension reduction by weighted pairwise fisher criteria," *IEEE Trans. Patterns Anal. Mach Intell.*, vol. 23(7), pp. 762–766, 2001.

[18] C. Thomaz, E. Kitani, and D. Gillies, "A maximum uncertainty lda-based approach for limited sample size problems - with application to face recognition," *Journal of the Brazilian Computer Society*, vol. 12, no. 2, pp. 7–18, 2006.

[19] E. Garcia and F. Lozano, "Boosting Support Vector Machines," in *Proceedings of International Conference of Machine Learning and Data Mining (MLDM'2007)*. Leipzig, Germany: IBal publishing, Jul. 2007, pp. 153–167.

[20] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of computer and system sciences*, vol. 55, no. 1, pp. 119–139, 1997.

[21] O. Langner, R. Dotsch, G. Bijlstra, D. H. J. Wigboldus, S. T. Hawk, and A. van Knippenberg, "Presentation and validation of the radboud faces database," *Cognition & Emotion*, vol. 24, no. 8, pp. 1377–1388, 2010.

[22] M. J. Lyons, S. Akamatsu, M. Kamachi, J. Gyoba, and J. Budynek, "The japanese female facial expression (jaffe) database," 1998.