Face Recognition Using Canonical Correlation, Discrimination Power, and Fractional Multiple Exemplar Discriminant Analyses

Submitted: 14th August 2015; accepted: 17th September 2015

Mohammadreza Hajiarbabi, Arvin Agah

DOI: 10.14313/JAMRIS_4-2015/29

Abstract:

Face recognition is a biometric identification method which compared to other methods, such as finger print identification, speech, signature, hand written and iris recognition is shown to be more noteworthy both theoretically and practically. Biometric identification methods have various applications such as in film processing, control access networks, among many. The automatic recognition of a human face has become an important problem in pattern recognition, due to (1) the structural similarity of human faces, and (2) great impact of factors such as illumination conditions, facial expression and face orientation. These have made face recognition one of the most challenging problems in pattern recognition. Appearance-based methods are one of the most common methods in face recognition, which can be categorized into linear and nonlinear methods. In this paper face recognition using Canonical Correlation Analysis is introduced, along with the review of the linear and nonlinear appearance-based methods. Canonical Correlation Analysis finds the linear combinations between two sets of variables which have maximum correlation with one another. Discriminant Power analysis and Fractional Multiple Discriminant Analysis has been used to extract features from the image. The results provided in this paper show the advantage of this method compared to other methods in this field.

Keywords: *face recognition, Canonical Correlation Analysis, Discrimination Power Analysis, Multiple Exemplar Discriminant Analysis, and Radial Basis Function neural networks*

1. Introduction

Recognizing the identity of humans is of great importance. Humans recognize each other based on physical characteristic such as face, voice, gait and etc. In the past centuries, the first systematic methods for recognizing were invented and used in police stations for recognizing the villains. This method measured the different parts of the body. After discovering that the finger print is unique for each person, this method became the best method for recognizing humans. In the last decades and because of inventing high speed computers, a good opportunity has been provided for the researches to work on different methods and to find certain methods for recognizing humans based on unique patterns.

A biometric system is a system which has an automated measuring component that is robust and can distinguish physical characteristics that can be used to identify a person. By robust it is meant that the features should not change significantly with the passing of years. For example iris recognition is more robust than other biometric systems because it does not change a lot over time. Due to matters of security, the budget for implementing biometric systems has increased [25]. A face biometric system can use both visual images and infra-red images, which have their own properties [19]. Face biometric systems can be divided into three categories based on the utilized implementation:

1. Appearance-based methods: These methods use statistical approaches to extract the most important information from the image.
2. Model-based methods: These use a model and then the model is placed on the test images and by computing some parameters, the person can be recognized. Elastic bunch graph [34], Active Appearance Model (AAM) [6] and 3D morphable model are some examples of model-based methods [1, 18].
3. Template-based methods: these methods first find the location of each part of the face for example eyes, nose etc. and then by computing the correlation between parts of the training images and the test images the face can be recognized [4].

All the face biometric systems should also include a face detection part in order to find the place of the face in the image. Viola used Adaboost algorithm to find faces in an image [33]. Rowley used neural networks [24]. In Viola and Rowley method a window was moved over the image in order to find a face. New methods use color images. Hsu [17] first used color images and skin detection in order to find faces in the image. In [14] faces were detected by using correlation and skin segmentation [15].

2. Appearance-Based Methods

Appearance-based methods start with the concept of image space. A two dimensional image can be shown as a point or vector in a high dimensional space which is called image space. In this image space, each dimension is compatible with a pixel of an image. In general, an image with m rows and n columns shows a point in a N dimensional space where $N = m \times n$. For example, an image with 20 rows and 20 columns describes a point in a 400 dimensional space. One important characteristic of image space is that changing the pixels of one image with each other does not

change the image space. Also image space can show the connection between a set of images [31]. The image space is a space with high dimensions. The appearance-based methods extract the most important information from the image and lower the dimension of the image space. The produced subspace under this situation is called feature space or face space [31].

The origin of appearance-based methods dates back to 1991 when Turk and Pentland introduced the Eigen face algorithm which is based on a famous mathematical method, namely, Principal Component Analysis [32]. This was the start of appearance-based methods. In 2000, Scholkopf by introducing kernel principal component analysis (Kernel Eigenface) expanded the concept of appearance-based method into non-linear fields. Appearance-based methods are robust to noise, defocusing, and similar issues [10]. Appearance-based methods have been classified into two categories of linear and non-linear methods. In the following sections these methods are described.

2.1. Linear Discriminant Analysis

In face space which is of $m \times n$ dimension with m, n as the image dimensions, $X = (X_1, X_2, ..., X_n) \subset \Re^{m \times n}$ is a matrix containing the images in the training set. X_i is an image that has been converted to a column vector. LDA maximize the between class scatter matrix to the within class scatter matrix [8]. The between class scatter matrix is calculated as:

$$S_B = \sum_{i=1}^{c} n^i \left(\overline{X}^i - \overline{X}\right)\left(\overline{X}^i - \overline{X}\right)^T$$

Where $\overline{X} = \left(1/n\right)\sum_{j=1}^{n} X_j$ the mean of the images in the training is set and $\overline{X}^i = \left(1/n^i\right)\sum_{j=1}^{n^i} X_j^i$ is the mean of class i, and c is the number of the classes (total images that belong to one person). The within class scatter matrix is calculated as:

$$S_W = \sum_{i=1}^{c} \sum_{X_i \in n^i} \left(X_i - \overline{X}^i\right)\left(X_i - \overline{X}^i\right)^T$$

The optimal subspace is calculated by:

$$E_{optimal} = \arg\max_E \frac{\left\|E^T S_B E\right\|}{\left\|E^T S_W E\right\|} = \left[c_1, c_2, ..., c_{c-1}\right]$$

Where $\left[c_1, c_2, ..., c_{c-1}\right]$ is the set of Eigen vectors of S_B and S_W corresponding to $c-1$ greatest generalized Eigen value λ_i and $i = 1,2,...,c-1$

$$S_B E_i = \lambda_i S_W E_i \quad i = 1,2,...,c-1$$

Thus, the most discriminant response for face images X would be [8]:

$$P = E_{optimal}^T \cdot X$$

In order to avoid the singularity problem first one has to reduce the dimension of the problem and then apply LDA. Principal component analysis (PCA) is the most common method which is used for dimension reduction. In this paper we applied principal component analysis on the images prior to other methods have been discussed. In addition to PCA, there are other effective methods that can be used as dimension reduction prior to LDA, such as Discrete Cosine Transform (DCT) [12].

Some researchers have observed that applying PCA to reduce the dimension of the space can cause another problem which is the elimination of some useful information from the null space. The 2FLD algorithm was introduced to address this problem and also the computational problem that applying PCA produces. But 2FLD algorithm introduces other problems such that the output of 2FLD method is a matrix and its dimension for an $m \times n$ image could be $n \times n$. This high dimension when a neural network is used for classification causes issues. A two and high dimension matrix cannot be applied to a neural network. If the matrix is changed into a vector, a vector with size n^2 is produced and because of low sample test of each face the network cannot be trained well. A direct LDA method that does not need to use PCA method before applying LDA has been proposed, but this method is time inefficient [36]. A fuzzy version of the LDA has also been proposed [20]. Shu *et al.* designed a linear discriminant analysis method that also preserved the local geometric structures [29]. In [9] the discriminant information was added into sparse neighborhood.

2.2. Fractional Multiple Exemplar Discriminant Analysis

The problem of face recognition differs from other pattern recognition problems and therefore it requires different discriminant methods rather than LDA. In LDA the classification of each class is based on just one sample and that is the mean of each class. Because of shortage of samples in face recognition applications, it is better to use all the samples instead of the mean of each class for classification. Rather than minimizing within class distance while maximizing the between class distance, multiple exemplar discriminant analysis (MEDA) finds the projection directions along which the within class exemplar distance (i.e., the distances between exemplars belonging to the same class) is minimized while the between-class exemplar distance (i.e., the distances between exemplars belonging to different classes) is maximized [37].

In MEDA the within class scatter matrix is calculated by:

$$S_W = \sum_{i=1}^{C} \frac{1}{n^{i^2}} \sum_{j=1}^{n^i} \sum_{k=1}^{n^i} \left(X_j^i - X_k^i\right)\left(X_j^i - X_k^i\right)^T$$

Where X_j^i is the *j*th image of *i*th class. Through comparison with the within class scatter matrix of LDA, it can be seen that in this method all the images in a class have participated in making the within class scatter matrix instead of using just the mean of the class, as in the LDA method. The between class scatter matrix is computed by:

$$S_B = \sum_{i=1}^{C} \sum_{j=1, j\neq i}^{C} \frac{1}{n^i n^j} \sum_{k=1}^{n^i} \sum_{l=1}^{n^j} \left(X_k^i - X_l^j\right)\left(X_k^i - X_l^j\right)^T$$

Dissimilarly to LDA in which the means of each class and means of all samples made the between class scatter matrix, in MEDA all the samples in one class are compared to all samples of the other class. The computation of $E_{optimal}$ is the same as LDA.

There is a drawback which is common in both LDA and MEDA. In between class scatter matrix (S_B) there will be no difference if the samples are closer or far from each other. However, it is clear that for the classes which are closer to each other the probability of collision is more than the other classes.

When the idea was first proposed [21], it was used for LDA and was not applied to face recognition databases. Later [13] the algorithm was combined with MEDA and was applied to face recognition. This algorithm suggests reducing the dimension of the problem step by step and in each iteration the samples which are closer are made to become far from each other. For this purpose a weight function has been introduced:

$$w\left(d_{X_1X_2}\right) = \left(d_{X_1X_2}\right)^{-p} \quad , \quad p = 3,4,...$$

Where $d_{X_1X_2}$ denotes the distance of the center of each class from each other [21] but for MEDA it should be considered as the distance of each two samples [13]. The between class scatter matrix in fractional MEDA is defined as:

$$S_B = \sum_{i=1}^{C} \sum_{j=1, j\neq i} \frac{1}{n^i n^j} \sum_{k=1}^{n^i} \sum_{l=1}^{n^i} w\left(d_{x_k^i x_l^j}\right) \times \left(X_k^i - X_l^j\right)\left(X_k^i - X_l^j\right)^T$$

The within class scatter matrix is the same as MEDA.

The fractional algorithm is shown in Table 1 [21]:

In the pseudo code r is the number of fractional steps used to reduce the dimensionality by 1 [21].

***Table 1. Fractional algorithm* [21]**

Set $W = I_{n\times n}$ (the identity matrix)
for $k = n$ to $(m+1)$ step (-1)
  for $\ell = 0$ to $(r-1)$ to step 1
  Project the data using W as $y = W^T x$
  Apply the scaling transformation to obtain $z = \varphi\left(y, \alpha^\ell\right)$
  For the z patterns, compute the $k \times k$ between class scatter matrix S_b
  Compute the ordered eigenvalues $\lambda_1, \lambda_2, ..., \lambda_k$ and corresponding eigenvectors $\phi_1, \phi_2, ...\phi_k$ of S_b
  Set $W = W\tilde{\Phi}$, where $\tilde{\Phi} = \left[\phi_1, \phi_2, ..., \phi_k\right]$
  end for
Discard the last (kth) column of W.
end for

The scaling transformation compresses the last component of y by a factor α^ℓ with $\alpha < 1$, *i.e*, $\Psi\left(y; \alpha^\ell\right): y \in \Re^k \rightarrow z \in \Re^k$ such that:

$$z_i = \begin{cases} \alpha^\ell y_i, & i = k \\ y_i, & i = 1,2,...,(k-1) \end{cases}$$

Some explanations about this algorithm are [21]:

- In the *rth* step, the reduction factor is α^{r-1}. It stipulates that a dimension is removed by $1, \alpha, \alpha^2, ...\alpha^{r-1}$ scales.
- When the number of steps is smaller, then α should be chosen larger and vice versa.
- The weighting functions should be chosen, as d^{-3}, d^{-4} and so on.

The FMEDA algorithm is shown in Table 2 [13].

***Table 2. FMEDA algorithm* [12]**

1. Applying PCA on the training set.
2. Computing within class scatter matrix using

$$S_W = \sum_{i=1}^{C} \frac{1}{n^{i2}} \sum_{j=1}^{n^i} \sum_{k=1}^{n^i} \left(X_j^i - X_k^i\right)\left(X_j^i - X_k^i\right)^T$$

3. Computing between class scatter matrixes using

$$S_B = \sum_{i=1}^{C} \sum_{j=1, j\neq i} \frac{1}{n^i n^j} \sum_{k=1}^{n^i} \sum_{l=1}^{n^i} w\left(d_{x_k^i x_l^j}\right) \times \left(X_k^i - X_l^j\right)\left(X_k^i - X_l^j\right)^T$$

4. Applying fractional step dimensionally reduction algorithm.
5. Computing optimal subspace using

$$E_{optimal} = \arg\max_E \frac{\left\|E^T S_B E\right\|}{\left\|E^T S_W E\right\|} = \left[c_1, c_2, ..., c_{c-1}\right]$$

6. Computing most discriminant vectors using

$$P = E_{optimal}^T \cdot X$$

2.3. Kernel Methods

Kernel methods are more recent methods, as compared to linear algorithms [3]. A kernel method finds the higher order correlations between instances and the algorithm, as described in this section. It is considered that patterns $x \in \Re^N$ are available, and that the most information lies in the *d*th dimension of pattern x.

$$\left[x\right]_{j_1} ... \left[x\right]_{jd}$$

One manner to extract all the features from data is to extract the relations between all the elements of a vector. In computer vision applications where all the images are converted to vectors, this feature extraction shows the relations between all pixels of the image. For example in $\Re^2$ (an image) all the second order relations can be mapped into a non-linear space:

$$\Phi: \Re^2 \rightarrow F = \Re^3$$
$$\left(\left[x\right]_1, \left[x\right]_2\right) \mapsto \left(\left[x\right]_1^2, \left[x\right]_2^2, \left[x\right]_1\left[x\right]_2\right)$$

This method is useful for low dimensional data but can cause problems for high dimensional data. For N dimensional data there are

$$N_F = \frac{(N+d-1)!}{d!(N-1)!}$$

different combinations that make a feature space with N_F dimension. For example, a 16*16 image with $d=5$ has a feature space of moment 10^{10}. By using kernel methods there is no need to compute these relations explicitly.

For computing dot products $\left(\Phi(x).\Phi(x')\right)$ the kernel method is defined as follows:

$$k(x,x') = \left(\Phi(x).\Phi(x')\right)$$

Which allows the dot product F to be computed without any need to map Φ. In this method, first used in [3], if x is an image then the kernel $(x.x')^d$ (or any other kernels) can be used to map onto a new feature space. This feature space is called the Hilbert space. In Hilbert space all the relations between any vectors can be shown using dot products. The input space is denoted as χ and the feature space is denoted by F, and the map by $\phi: \chi \to F$. Any function that returns the inner product of two points $x_i \in \chi$ and $x_j \in \chi$ in the F space is called a kernel function.

Some of the popular kernels include [21]:

Polynomial kernel: $k(x,y) = (x.y)^d$

RBF kernel: $k(x,y) = \exp\left(-\|x-y\|^2 / 2\sigma^2\right)$

Sigmoid kernel:

$$k(x,y) = \tanh\left(\kappa(x.y)^d + \theta\right) \; d \in \aleph, \kappa > 0, \theta < 0$$

Also kernels can be combined using these methods in order to produce new kernels:

$$\alpha k_1(x,y) + \beta k_2(x,y) = k(x,y)$$

$$k_1(x,y)k_2(x,y) = k(x,y)$$

2.3.1. Kernel Methods

By having m instances x_k with zero mean and $x_k = \left[x_{k1}, x_{k2}, ..., x_{kn}\right]^T \in \Re^n$, principal component analysis method finds the new axis in the direction of the maximum variances of the data and this is equivalent to finding the eigenvalues of the covariance matrix C:

$$\lambda w = Cw$$

For eigenvalues $\lambda \geq 0$ and eigenvectors $w \in \Re^n$ in kernel principal component analysis, each vector x from the input space $\Re^n$ to the high dimensional feature space $\Re^f$ is mapped using a nonlinear mapping function $\Phi: \Re^n \to \Re^f, f > n$. In $\Re^f$ the eigenvalue problem is as follows:

$$\lambda w^\Phi = C^\Phi w^\Phi$$

$$C^\Phi = \frac{1}{m}\sum_{j=1}^{m} \Phi(x_j)\Phi(x_j)^T$$

Where C^Φ is the covariance matrix. The eigenvalues $\lambda \geq 0$ and eigenvectors $w^\Phi \in F \setminus \{0\}$ (the eigenvectors with eigenvalues that are not zero) must be determined in a manner that qualifies $\lambda w^\Phi = C^\Phi w^\Phi$. By using it:

$$\lambda\left(\Phi(x_k)\cdot w\right) = \left(\Phi(x_k)\cdot C^\Phi w\right) \; k = 1,2,...,m$$

Also the coefficient α_i exists such that:

$$w^\Phi = \sum_{i=1}^{m} \alpha_i \Phi(x_i)$$

By combining the last three and by introducing K a $m \times m$ matrix:

$$K_{ij} = k(x_i, x_j) = \Phi(x_i)\cdot\Phi(x_j)$$

The equation:

$$\lambda\sum_{i=1}^{m}\alpha_i\left(\Phi(x_k)\cdot\Phi(x_i)\right) = \frac{1}{m}\sum_{i=1}^{m}\alpha_i\left(\Phi(x_k)\cdot\sum_{j=1}^{m}\Phi(x_j)\right)\times$$
$$\left(\Phi(x_j)\cdot\Phi(x_i)\right) \Rightarrow m\lambda K\alpha = K^2\alpha$$

is reached, the kernel principal component analysis becomes:

$$m\lambda K\alpha = K^2\alpha \equiv m\lambda\alpha = K\alpha$$

Where α is a column vector with values $\alpha_1, ..., \alpha_m$ [27].

For normalizing the eigenvectors in F that is $\left(w^k \cdot w^k\right) = 1$ the equation used is:

$$1 = \sum_{i,j=1}^{m}\alpha_i^k\alpha_j^k\left(\Phi(x_i)\cdot\Phi(x_j)\right) = \left(\alpha^k\cdot K\alpha^k\right) = \lambda_k\left(\alpha^k\cdot\alpha^k\right)$$

For extracting the principal components from the test instance x, and its projection in the $\Re^f$ space is $\Phi(x)$, only the projection of $\Phi(x)$ must be computed on the eigenvectors w^k in the feature subspace F by [27]:

$$\left(w^k\cdot\Phi(x)\right) = \sum_{i=1}^{m}\alpha_i^k\left(\Phi(x_i)\cdot\Phi(x)\right) = \sum_{i=1}^{m}\alpha_i^k k(x_i, x)$$

It should be noted that none of equations need $\Phi(x_i)$ in an explicit way. The dot products must only be calculated using the kernel function without the need to apply the map Φ. In face recognition, each vector x shows a face image and this is why the non-linear principal component is called kernel eigenface in the face recognition domain.

The kernel principal component analysis is shown is Table 3 [27].

***Table 3. KPCA algorithm* [27]**

1. Calculate the gram matrix by using:

$$K_{training}=\begin{bmatrix} k(x_1,x_1) & k(x_1,x_2) & \dots & k(x_1,x_m) \\ k(x_2,x_1) & k(x_2,x_{21}) & \dots & k(x_2,x_m) \\ \dots & \dots & \dots & \dots \\ k(x_m,x_1) & k(x_m,x_2) & \dots & k(x_m,x_m) \end{bmatrix}$$

2. Calculate $m\lambda\alpha = K\alpha$ and compute α
3. Normalize α^n using:

$$1=\sum_{i,j=1}^{m}\alpha_i^k\alpha_j^k\left(\Phi(x_i)\cdot\Phi(x_j)\right)=\left(\alpha^k\cdot K\alpha^k\right)=\lambda_k\left(\alpha^k\cdot\alpha^k\right)$$

4. Calculate the principal component coefficients for test data x using:

$$\left(w^k\cdot\phi(x)\right)=\sum_{i=1}^{m}\alpha_i^k k(x_i,x)$$

The classical principal component analysis is also a special version of kernel principal component analysis in which the kernel function is a first order polynomial. Therefore, the kernel principal analysis is a generalized form of principal component analysis that has used different kernels for nonlinear mapping.

Another important matter is using data with zero mean in the new subspace, which can be accomplished using:

$$\tilde{\Phi}(x_i)=\Phi(x_i)-\left(1/m\right)\sum_{i=1}^{m}\Phi(x_i)$$

As there are no data in explicit form in the new space, the following method is used [26]. By considering that for each *i* and *j* $1_{ij}=1$.

$$\tilde{K}_{ij}=\tilde{\Phi}(x_i)^T\tilde{\Phi}(x_j)=\left(\Phi(x_i)-\sum_{l=1}^{m}\Phi(x_l)\right)^T\left(\Phi(x_j)-\sum_{n=1}^{m}\Phi(x_n)\right)=$$

$$\Phi(x_i)^T\Phi(x_j)-\frac{1}{m}\sum_{l=1}^{m}\Phi(x_l)^T\Phi(x_j)-\frac{1}{m}\sum_{n=1}^{m}\Phi(x_i)^T\Phi(x_n)+$$

$$\frac{1}{m^2}\sum_{l,n=1}^{m}\Phi(x_l)^T\Phi(x_n)=K_{ij}-\frac{1}{m}\sum_{l=1}^{m}1_{il}K_{lj}-\sum_{n=1}^{m}K_{in}1_{nj}+\frac{1}{m^2}\sum_{l,n=1}^{m}1_{il}K_{\ln}1_{nj}$$

The above formula can be rewritten as [26]:

$$\tilde{K}_{ij}=K-1_mK-k1_m+1_mK1_m\ (1_m)_{ij}=1/m$$

For Kernel Fisher face first principal component analysis is applied on the image, and then LDA is applied on the new vector [35].

2.4. Canonical Correlation Analysis

Canonical Correlation Analysis (CCA) is one mechanism for measuring the linear relationship between two multi-dimensional relationships. This method was first introduced by [16], and although it has been known as a standard tool in pattern recognition, it has been used rarely in signal processing and biometric identification systems. CCA has had various applications in economics, medical studies and metrology.

It is assumed that X is a matrix with $m\times n$ dimension that consists of m array of a n dimensional vector from a random variable x. The correlation coefficient ρ_{ij} that shows the correlation between the x_i and x_j is defined by:

$$\rho_{ij}=\frac{C_{ij}}{\sqrt{C_{ii}C_{jj}}}$$

Where C_{ij} shows the covariance matrix between x_i and x_j, and is computed by:

$$C_{ij}=\frac{1}{m-1}\sum_{k=1}^{m}\left(X_{ki}-\mu_i\right)\left(X_{kj}-\mu_j\right)$$

μ_i is the average of x_i's. A_x is the centered matrix of X that its elements are

$$a_{ij}=X_{ij}-\mu_j$$

Therefore the covariance matrix is defined by:

$$C==\frac{1}{m-1}A_x^TA_x$$

It has to be considered that correlation coefficients demonstrate a measurement of linear intersection between two variables. When two variables are uncorrelated (i.e., their correlation coefficients are zero) it states that there is no linear function that could describe the connection between the two variables.

The aim of CCA is to determine the correlation between two sets of variables. CCA attempts to find the basis vectors for two sets of multidimensional variables in such a way that the linear correlation between the projected vectors on these basis vectors are maximized mutually.

The CCA method attempts to find the basis vectors for two sets of vectors, one for x and one for y in such a way that the correlation between the projection of these variables on the basis vectors are maximized. Assuming that the zero mean vectors are X and *Y*, the CCA method finds the vectors α and β such that the correlations between the projections of $a_1=\alpha^TX$ and $b_1=\beta^TY$ are maximized. The projections a_1 and b_1 are called the first canonical variables. Then the second dual canonical variables a_2 and b_2 are computed which are uncorrelated with the canonical variables a_1 and b_1, and this process is continued.

Considering $\omega_1,\omega_2,...,\omega_c$ as features belonging to class c and the training data space being defined as $\Omega=\left\{\xi\,|\,\xi\in\Re^N\right\}$, defining $A=\left\{x\,|\,x\in\Re^P\right\}$ and $B=\left\{y\,|\,y\in\Re^q\right\}$ then x and y are feature vectors from one instance ξ which have been extracted using two different feature extracting method. The goal is to calculate the canonical correlations between x and y α_1^Tx and β_1^Ty are the two first vectors, α_2^Tx and β_2^Ty are second dual vectors and can be written as:

$$X^*=\left(\alpha_1^Tx,\alpha_2^Tx,...,\alpha_d^Tx\right)^T=\left(\alpha_1,\alpha_2,...,\alpha_d\right)^Tx=W_x^Tx$$

$$Y^*=\left(\beta_1^Ty,\beta_2^Ty,...,\beta_d^Ty\right)^T=\left(\beta_1,\beta_2,...,\beta_d\right)^Ty=W_y^Ty$$

$$Z_1 = \begin{pmatrix} X^* \\ Y^* \end{pmatrix} = \begin{pmatrix} W_x^T x \\ W_y^T y \end{pmatrix} = \begin{pmatrix} W_x & 0 \\ 0 & W_y \end{pmatrix}^T \begin{pmatrix} x \\ y \end{pmatrix}$$

And the transform matrix is:

$$W_1 = \begin{pmatrix} W_x & 0 \\ 0 & W_y \end{pmatrix}$$
$$W_x = (\alpha_1, \alpha_2, ..., \alpha_d), W_y = (\beta_1, \beta_2, ..., \beta_d)$$

The directions α_i and β_i are called the *i*th Canonical Projective Vectors (CPV) and x, y, $\alpha_i^T x$ and $\beta_i^T y$ are the *i*th features of canonical correlations. Also W_1 and W_2 are called Canonical Projective Matrix (CPM) and Z_1 is Canonical Correlation Discriminant Feature (CCDF) and the method is called Feature Fusion Strategy (FFS) [2, 30].

For determining the CCA coefficients, it is assumed that x and y are two random variables with zero means. The total covariance matrix is defined by:

$$C = \begin{bmatrix} C_{xx} & C_{xy} \\ C_{yx} & C_{yy} \end{bmatrix} = E\left[\begin{pmatrix} x \\ y \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}^T \right]$$

Where C_{xx} and C_{yy} are the inner set covariance matrix of x and y, and $C_{xy} = C_{yx}^T$ is the between set covariance matrix. The correlation between x and y is defined as [30]:

$$\begin{cases} C_{xx}^{-1} C_{xy} C_{yy}^{-1} C_{yx} \alpha = \rho^2 \alpha \\ C_{yy}^{-1} C_{yx} C_{xx}^{-1} C_{xy} \beta = \rho^2 \beta \end{cases}$$

Where ρ^2 is the square correlation and the eigenvectors α and β are normalized basis correlation vectors.

3. Experimental Results

In order to test the described algorithms, the Sheffield (UMIST) [26] and ORL [23] databases were utilized in the experiments. The Sheffield database contains 575 images that belong to 20 people with variety of head pose from front view to profile. For training, 10 images were used from each person and the rest were used as test set. Figure 1 shows a sample of this database.

***Fig. 1. Sheffield database* [28]**

ORL database contains 400 images. This database contains 40 people with variety in scale and pose of the head. From every person, five images were used as training set and the rest as test set. Figure 2 shows a sample of this database.

***Fig. 2. ORL database* [23]**

3.1. Linear Methods

In order to establish a baseline, the linear algorithms were utilized. Matlab was used for the simulation [22]. For the neural network, the network inputs are equal to the features vector's dimensions. For the output two approaches can be used. The first one is the bit method in which the class number is shown by using bits. Each output neuron is equivalent to one bit. For instance, 000110 shows class 6 and 001001 shows class 9. The output of an RBF network is a real number between 0 and 1. The other method is considering a neuron for each class. If there are 40 classes, then there are also 40 nodes in the output layer. The second method produced better results and in all simulations the second method has been used. However in cases with large number of classes, the first method may be preferred. Also a neuron can be considered for images that do not belong to any classes.

It should be noted that the two other important neural networks classifiers, back propagation neural network and probabilistic neural network, have lower performances than RBF neural networks in these experiments. Back propagation neural network needs significant time for training compared to the RBF neural network. The memory needed for back propagation neural network is also much larger than the RBF neural network. The experiment also shows that the results using back propagation neural network is of lesser quality than RBF neural network. Probabilistic neural network performance is equivalent to distance based classifiers performance.

For linear methods, principal component analysis, linear discriminant analysis, fuzzy linear discriminant analysis [20] and multiple exemplar linear discriminant analysis have been used. Results are shown based on the number of extracted features. Figure 3 illustrates the results for linear methods using RBF neural network [5, 10]. For all algorithms in this paper the distance based classifier was also used as a classifier and in most cases the RBF neural networks outperformed distance based classifiers. When the number of the extracted features was low, distance based had better results.

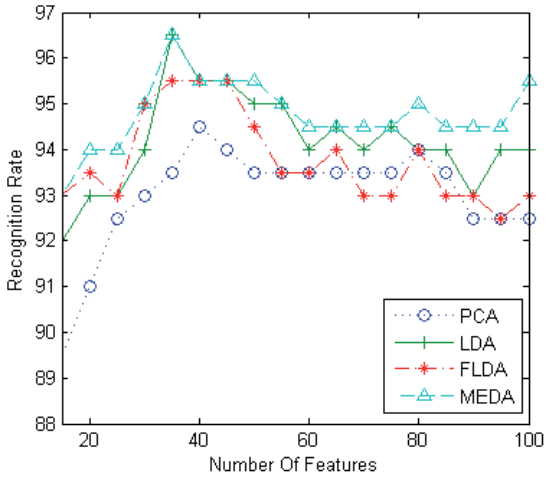

Fig. 3. Linear based algorithms using RBF classifier on ORL database

As the figures show multiple exemplar discriminant analysis has stronger discriminant capabilities compared with the other methods. Figure 4 shows the results for the Sheffield database.

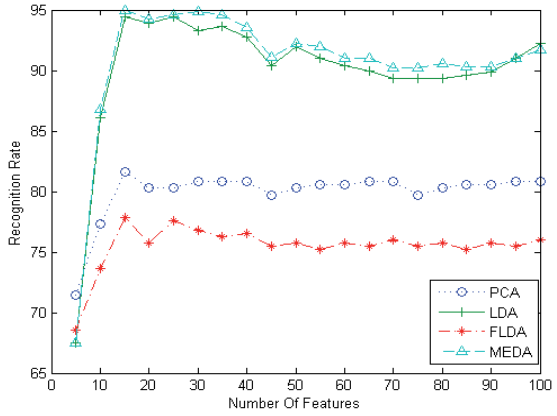

Fig. 4. Linear based algorithms using RBF classifier on Sheffield database

Figure 5 and Figure 6 show the results of FMEDA algorithm compared with LDA and MEDA algorithm. The results indicate that FMEDA algorithm has better recognition rate compared to LDA and MEDA methods and other linear methods.

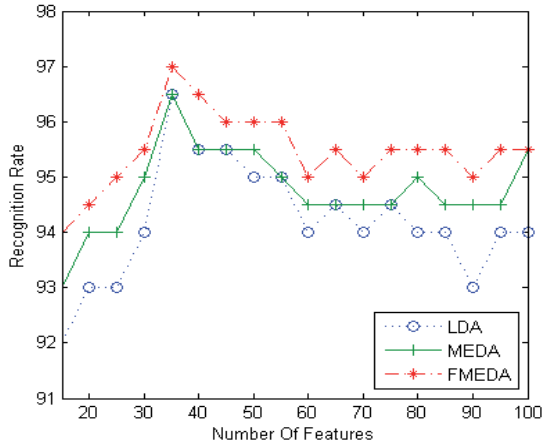

Fig. 5. FMEDA algorithm using RBF classifier on ORL database

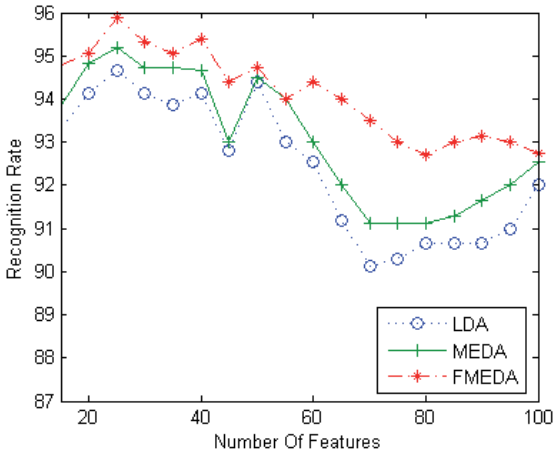

Fig. 6. FMEDA algorithm using RBF classifier on Sheffield database

3.2. Non-Linear Methods

For nonlinear methods kernel principal component analysis and kernel linear discriminant analysis have been used. For kernel linear discriminant analysis first kernel principal component analysis has been applied to the images and then the linear discriminant analysis is applied to the new vector. Second order polynomial is used as kernel function. Figures 7 and 8 display the results.

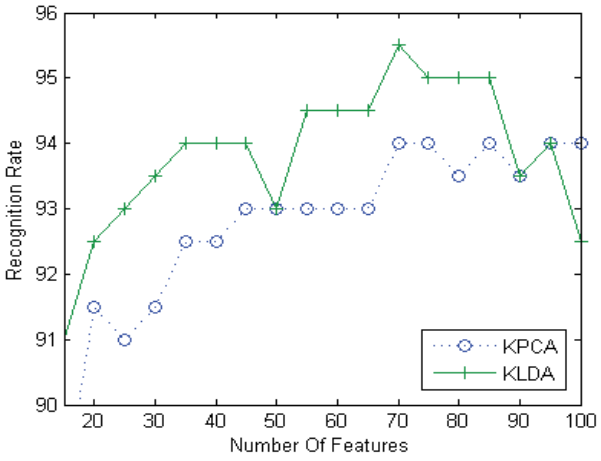

Fig. 7. Non-linear based algorithms using RBF classifier on ORL database

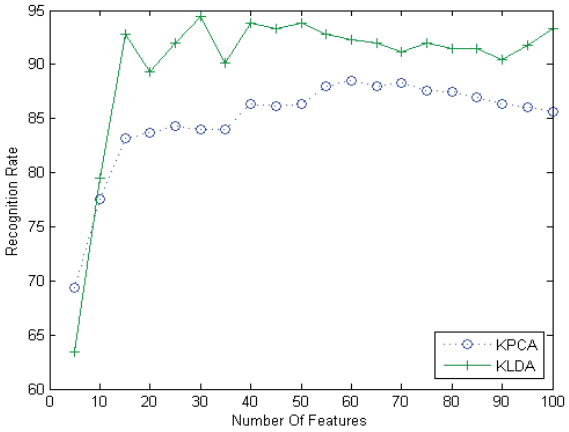

Fig. 8. Non-linear based algorithms using RBF classifier on Sheffield database

As the figures show kernel linear discriminant analysis has better results compared to kernel principal component analysis. Also kernel principal component analysis has better results compared to Eigen face method.

Comparing the results with the linear algorithms confirms that the non-linear method is not that much better than the linear methods. The reason can be that the between class distances have not become more when the space is changed to a higher dimensional space.

3.3. Evaluating CCA

Combining the information is a powerful technique that is being used in data processing. This combining can be done at three levels of pixel level, feature level, and decision level - Similar to combining the classifiers. CCA combines the information in the feature level.

One of the advantages of combining the features is that the features (vectors which have been calculated using different methods) contain different characteristic from the pattern. By combining these two methods not only the useful discriminant information from the vectors are kept, but also the redundant information is omitted.

For this experiment, CCA was applied to two different feature vectors. In this case two different methods should be used where each extract features from the image using different technique. One of the methods that are used is FMEDA which had better results compared to other linear and non-linear methods in appearance-based methods. The other method that we used is Discrimination Power Analysis (DPA). CCA is applied to the extracted features using these two methods.

A method has been introduced based on DCT that extract features that have better capability to discriminate faces [7]. As mentioned before in conventional DCT the coefficients are chosen using a zigzag manner, where some of the low frequency coefficients are discarded because they contain the illumination information. The low frequency coefficients are in the upper left part of the image. Some of the coefficients have more discrimination power compared to other coefficients, and therefore by extracting these features a higher true recognition rate can be achieved. So, instead of choosing the coefficients in a zigzag manner [7] searched for coefficients which have more power to discriminate between images. Unlike other methods such as PCA and LDA which use between and within class scatter matrices and try to maximize the discrimination in the transformed domain, DPA searches for the best discrimination features in the original domain.

The DPA algorithm is as follows [7]:

Considering DCT has been applied to an image and coefficients are *X:*

$$X=\begin{bmatrix} x_{11} & x_{12} & \dots & x_{1N} \\ x_{21} & x_{22} & \dots & x_{2N} \\ \dots & \dots & \dots & \dots \\ x_{M1} & x_{M2} & \dots & x_{MN} \end{bmatrix}_{M\times N}$$

Where the number of people in the database is C (The number of classes), and for each person there are S images (Training images). There are total $C*S$ training images. Table 4 shows how to calculate the DPA of each coefficient x_{ij} :

***Table 4. DPA algorithm* [7]**

1. Construct a large matrix containing all the DCT from the training images.

$$A_{ij}=\begin{bmatrix} x_{ij}(1,1) & x_{ij}(1,2) & \dots & x_{ij}(1,C) \\ x_{ij}(2,1) & x_{ij}(2,2) & \dots & x_{ij}(2,C) \\ \dots & \dots & \dots & \dots \\ x_{ij}(S,1) & x_{ij}(S,2) & \dots & x_{ij}(S,C) \end{bmatrix}_{S\times C}$$

2. Calculate the mean and variance of each class:

$$M_{ij}^{c}=\frac{1}{S}\sum_{s=1}^{S}A_{ij}(s,c)$$

$$V_{ij}^{c}=\sum_{s=1}^{S}\left(A_{ij}(s,c)-M_{ij}^{c}\right)^{2} \quad c=1,2,\dots,C$$

3. Calculate variance of all classes

$$V_{ij}^{W}=\frac{1}{C}\sum_{c=1}^{C}V_{ij}^{c}$$

4. Calculate the mean and variance of all training samples:

$$M_{ij}^{c}=\frac{1}{S\times C}\sum_{c=1}^{C}\sum_{s=1}^{S}A_{ij}(s,c)$$

$$V_{ij}^{B}=\sum_{c=1}^{C}\sum_{s=1}^{S}\left(A_{ij}(s,c)-M_{ij}\right)^{2}$$

5. For location *(i, j)* calculate the DP:

$$D(i,j)=\frac{V_{ij}^{B}}{V_{ij}^{W}} \quad 1\le i\le M,\ 1\le j\le N$$

The higher values in D show the higher discrimination ability it has. Table 5 shows the procedure for recognizing faces:

***Table 5. Procedure for recognizing faces* [7]**

1. Compute the DCT of the training images, and normalize the results.
2. Use a mask, to discard some of the low and high frequencies.
3. Calculate DPA for the coefficients inside the mask.
4. The n largest coefficients are found and marked. Set the remaining coefficients to zero. The resulting matrix is an $M*N$ matrix having n elements that are not zero.
5. Multiply the DCT coefficients by the matrix which was calculated in the previous step. Convert the resulting matrix into a vector.
6. Train a classifier using the training vectors. Apply the same process for the test images.

Figure 9 shows the comparison between FMEDA, DPA and CCA. The results illustrate that applying CCA to the features can increase the recognition rate for human faces.

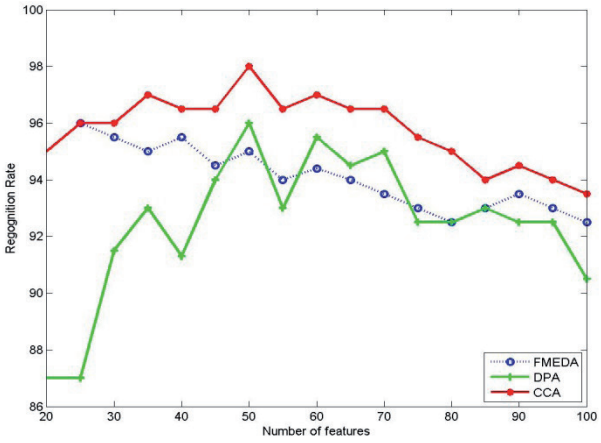

Fig. 9. Comparing CCA with FMEDA and DPA using RBF classifier on ORL database

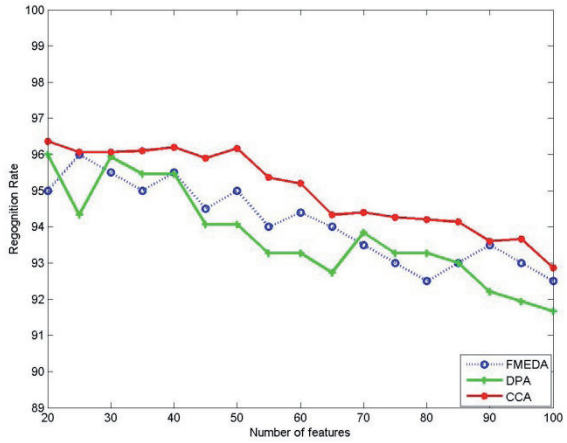

Fig. 10. Comparing CCA with FMEDA and DPA using RBF classifier on Sheffield database

4. Conclusion

In this paper several linear and non-linear appearance based method were discussed and the methods were applied on two popular face recognition database. In linear methods FMEDA had better results compared to other linear methods and in non-linear methods KLDA outperforms KPCA. Also the experiments show that the linear method has similar recognition rate compared to non-linear methods. Also a new method for face recognition was introduced that outperforms existing linear and non-linear methods. Canonical Correlation Analysis (CCA) is a strong tool in combining the information at feature level. Fractional Multiple Exemplar Analysis (FMEA) and Discriminant Power Analysis (DPA) were used as feature extraction techniques. This paper's experimental results show that CCA using DPA and FMEDA exhibits improved results compared to other related methods.

AUTHORS

Mohammadreza Hajiarbabi* – Department of Electrical Engineering and Computer Science, University of Kansas, Lawrence, Kansas, USA.
E-mail: mehrdad.hajiarbabi@ku.edu

Arvin Agah – Department of Electrical Engineering and Computer Science, University of Kansas, Lawrence, Kansas, USA.
E-mail: agah@ku.edu

*Corresponding author

REFERENCES

[1] Blanz V. S., Vetter T., "Face identification across different poses and illuminations with a 3D morphable model". In: *IEEE International Conference on Automatic Face and Gesture Recognition*, 2002, 202–207. DOI: 10.1109/AFGR.2002.1004155.

[2] Borga M., *Learning multidimensional signal processing*, Department of Electrical Engineering, Linköping University, Linköping Studies in Science and Technology Dissertations, no. 531, 1998.

[3] Boser B. E., Guyon I. M., Vapnik V. N., "A training algorithm for optimal margin classifiers." In: D. Haussler, editor, *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory*, 1992, 144–152. DOI: 10.1145/130385.130401.

[4] Brunelli R., Poggio T., "Face recognition: Features versus templates", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 10, 1993, 1042–1053. DOI: 10.1109/34.254061.

[5] Chen S., Cowan P.M., "Orthogonal least squares learning algorithms for radial basis function networks", *IEEE Transaction on Neural Networks*, vol. 2, no. 2, 1991, 302–309. DOI: 10.1109/72.80341.

[6] Cootes T.F., Edwards G.J., Taylor C.J., "Active appearance models", *IEEE Transactions Pattern Analysis and Machine Intelligence*, vol. 23, no. 6, 2001, 681–685. DOI: 10.1109/34.927467.

[7] Dabbaghchian S., Ghaemmaghami M., Aghagolzadeh A., "Feature extraction using discrete cosine transform and discrimination power analysis with a face recognition technology", *Pattern Recognition*, vol. 43, no. 4, 2010, 1431–1440. DOI: 10.1016/j.patcog.2009.11.001.

[8] Fukunaga K., *Introduction to statistical pattern recognition*, 2nd ed., San Diego, CA: Academic Press, 1990, 445-450.

[9] Gui J., Sun Z., Jia W., Hu R., Lei Y., Ji S., "Discriminant sparse neighborhood preserving embedding for face recognition", *Pattern Recognition*, vol. 45, no. 8, 2012, 2884–2893. DOI: 10.1016/j.patcog.2012.02.005.

[10] Gupta J. L., Homma N., *Static and dynamic neural networks from fundamentals to advanced theory*, John Wiley & Sons, 2003.

[11] Hajiarbabi M., Askari J., Sadri S., Saraee M., "The Evaluation of Camera Motion, Defocusing and

Noise Immunity for Linear Appearance Based Methods in Face Recognition". In: *IEEE Conference WCE 2007/ICSIE 2007*, vol. 1, 2007, 656–661.

[12] Hajiarbabi M., Askari J., Sadri S., Saraee M., "Face Recognition Using Discrete Cosine Transform plus Linear Discriminant Analysis". In: *IEEE Conference WCE 2007/ICSIE 2007*, vol. 1, 2007, 652–655.

[13] Hajiarbabi M., Askari J., Sadri S., "A New Linear Appearance-based Method in face Recognition", *Advances in Communication Systems and Electrical Engineering. Lecture Notes in Electrical Engineering*, vol. 4, Springer, 2008, 579–587. DOI: 10.1007/978-0-387-74938-9_39.

[14] Hajiarbabi M., Agah A., "Face Detection in color images using skin segmentation", *Journal of Automation, Mobile Robotics and Intelligent Systems*, vol. 8, no. 3, 2014, 41–51.

[15] Hajiarbabi M., Agah A., "Human Skin Color Detection using Neural Networks", *Journal of Intelligent Systems*, under review, 2014.

[16] Hotelling H., "Relations between two sets of variates", Biometrika, vol. 28, no. 3–4, 1936, 321–377. DOI: 10.2307/2333955.

[17] R., Hsu, M., and Abdel-Mottaleb, A., Jain, "Face Detection in Color images", IEEE Transactions on pattern analysis and machine intelligence, vol. 24, no. 5, 2002, 696-706.

[18] Huang J., Heisele B., Blanz V., "Component-based Face Recognition with 3D Morphable Models". In: *Proceedings of the 4th International Conference on Audio- and Video-based Biometric Person Authentication*, chapter 4, Surrey, UK, 2003. DOI: 10.1007/3-540-44887-X_4.

[19] Kong S., Heo J., Abidi B., Pik P., M., Abidi, "Recent Advances in Visual and Infrared Face Recognition – A Review", *Journal of Computer Vision and Image Understanding*, vol. 97, no. 1, 2005, 103–135. DOI: 10.1016/j.cviu.2004.04.001.

[20] Kwak K.C., Pedrycz W., "Face recognition using a fuzzy Fisher face classifier", *Pattern Recognition*, vol. 38, 2005, 1717–1732.

[21] Lotlikar R., Kothari R., "Fractional-step dimensionality reduction", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 6, 2000, 623–627. DOI: 10.1109/34.862200.

[22] Math Works, 2015: www.mathworks.com.

[23] ORL Database, 2015: http://www.camorl.co.uk.

[24] Rowley H., Baluja S., Kanade T., "Neural network-based face detection", *IEEE Pattern Analysis and Machine Intelligence*, vol. 20, 1998, 22–38.

[25] Sarfraz M., Computer Aided Intelligent Recognition Techniques and Applications, John Wiley & Sons, 2005, 1–10.

[26] Scholkopf B., *Statistical learning and kernel methods*, Microsoft Research Limited, February 29, 2000.

[27] Scholkopf B., Smola A., Muller K.R., "Non-linear component analysis as a kernel eigenvalue problem", *Neural Computation*, vol. 10, no. 5, 1998, 1299–1319.

[28] Sheffield (UMIST) Database, 2015: http://www.sheffield.ac.uk/eee/research/iel/research/face.

[29] Shu X., Gao Y., Lu H., "Efficient linear discriminant analysis with locality *preserving for face recognition", Pattern* Recognition, vol. 45, no. 5, 2012, 1892–1898.

[30] Sun Q.S., Zeng S.G., Liu Y., Heng P.A., Xia D.S., "A new method of feature fusion and its application in image recognition", *Pattern Recognition*, vol. 38, no. 12, 2005. DOI: 10.1016/j.patcog.2004.12.013.

[31] Turk M., "A Random Walk through Eigen space", *IEICE Transactions on Information and System*, vol. 84, no. 12, 2001.

[32] Turk M., Pentland A.,, "Eigen faces for recognition", *Journal of Cognitive Neuroscience*, vol. 3, 1991, 71–86.

[33] Viola P., Jones M.J., "Robust real-time object detection". In: *Proceedings of IEEE Workshop on Statistical and Computational Theories of Vision*, 2001.

[34] Wiskott L., Fellous J.M., Kruger N., Malsburg C., "Face recognition by elastic bunch graph matching", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, 1997, 775–779.

[35] Yang J., Jin Z., Yang J., Zhang D., Frangi F., "Essence of Kernel Fisher discriminant: KPCA plus LDA", *Elsevier Pattern Recognition*, vol. 37, no. 10, 2004, 2097–2100. DOI: 10.1016/j.patcog.2003.10.015.

[36] Yu H., Yang J., "A Direct LDA algorithm for high dimensional data with application to face recognition", *Pattern Recognition*, vol. 34, no. 10, 2001, 2067–2070.

[37] Zhou Sh. K., Chellappa R., "Multiple-Exemplar discriminant analysis for face recognition", Center for Automation Research and Department of Electrical and Computer Engineering University of Maryland, College Park, MD 20742, 2003.