EEE-6561 Fundamentals of Biometric Identification

February 9th, 2018

Lecture #8: Appearance-Based Face Recognition (PCA/LDA)

Damon L. Woodard, Ph.D.
Dept. of Electrical and Computer Engineering
dwoodard@ece.ufl.edu

Principal Component Analysis (PCA)

- Pattern recognition in high-dimensional spaces
  - Problems arise when performing recognition in a high-dimensional space (curse of dimensionality).
  - Significant improvements can be achieved by first mapping the data into a *lower-dimensional sub*-space.

$$x = \begin{bmatrix} a_1 \\ a_2 \\ \dots \\ a_N \end{bmatrix} --> reduce\ dimensionality --> y = \begin{bmatrix} b_1 \\ b_2 \\ \dots \\ b_K \end{bmatrix} \quad (K << N)$$

  - The goal of PCA is to reduce the dimensionality of the data while retaining as much as possible of the variation present in the original dataset.

Principal Component Analysis (PCA)

- Dimensionality reduction
  - PCA allows us to compute a linear transformation that maps data from a high dimensional space to a lower dimensional sub-space.

Principal Component Analysis (PCA)

- Lower dimensionality basis
  - Approximate vectors by finding a basis in an appropriate lower dimensional space.

(1) Higher-dimensional space representation:

$$x = a_1 v_1 + a_2 v_2 + \cdots + a_N v_N$$

$v_1, v_2, \ldots, v_N$ is a basis of the N-dimensional space

(2) Lower-dimensional space representation:

$$\hat{x} = b_1 u_1 + b_2 u_2 + \cdots + b_K u_K$$

$u_1, u_2, \ldots, u_K$ is a basis of the K-dimensional space

- *Note:* if both bases have the same size ($N = K$), then $x = \hat{x}$)

Principal Component Analysis (PCA)

- Information loss
  - Dimensionality reduction implies information loss !!
  - Want to preserve as much information as possible, that is:

$$\textbf{minimize} \; ||x - \hat{x}|| \; \text{(error)}$$

- How to determine the best lower dimensional sub-space?

The best low-dimensional space can be determined by the "best" eigenvectors of the covariance matrix of x (i.e., the eigenvectors corresponding to the "largest" eigenvalues -- also called "principal components").

Principal Component Analysis (PCA)

- Methodology
  - Suppose $x_1, x_2, ..., x_M$ are N x 1 vectors

Step 1: $\bar{x} = \frac{1}{M}\sum_{i=1}^{M} x_i$

Step 2: subtract the mean: $\Phi_i = x_i - \bar{x}$

Step 3: form the matrix $A = [\Phi_1\ \Phi_2 \cdots \Phi_M]$ (NxM matrix), then compute:

$$C = \frac{1}{M}\sum_{n=1}^{M} \Phi_n \Phi_n^T = AA^T$$

(sample **covariance** matrix, NxN, characterizes the *scatter* of the data)

Step 4: compute the eigenvalues of C: $\lambda_1 > \lambda_2 > \cdots > \lambda_N$

Step 5: compute the eigenvectors of C: $u_1, u_2, \ldots, u_N$

Principal Component Analysis (PCA)

- Methodology – cont.

- Since C is symmetric, $u_1, u_2, \ldots, u_N$ form a basis, (i.e., any vector x or actually $(x - \bar{x})$, can be written as a linear combination of the eigenvectors):

$$x - \bar{x} = b_1 u_1 + b_2 u_2 + \cdots + b_N u_N = \sum_{i=1}^{N} b_i u_i$$

Step 6: (**dimensionality reduction step**) keep only the terms corresponding to the K largest eigenvalues:

$$\hat{x} - \bar{x} = \sum_{i=1}^{K} b_i u_i \text{ where } K << N$$

- The representation of $\hat{x} - \bar{x}$ into the basis $u_1, u_2, \ldots, u_K$ is thus

$$\begin{bmatrix} b_1 \\ b_2 \\ \ldots \\ b_K \end{bmatrix}$$

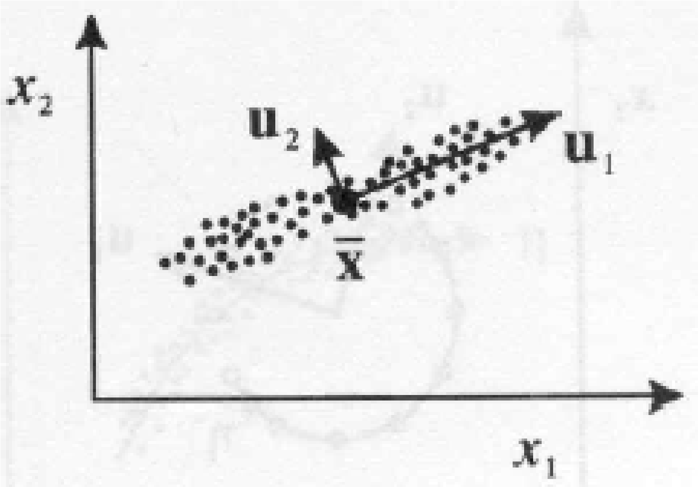
Principal Component Analysis (PCA)

- Linear transformation implied by PCA
  - The linear transformation $R^N \rightarrow R^K$ that performs the dimensionality reduction is:

$$\begin{bmatrix} b_1 \\ b_2 \\ \ldots \\ b_K \end{bmatrix} = \begin{bmatrix} u_1^T \\ u_2^T \\ \ldots \\ u_K^T \end{bmatrix} (x - \bar{x}) = U^T (x - \bar{x})$$

Principal Component Analysis (PCA)

- Geometric interpretation
  - PCA projects the data along the directions where the data varies the most.
  - These directions are determined by the eigenvectors of the covariance matrix corresponding to the largest eigenvalues.
  - The magnitude of the eigenvalues corresponds to the variance of the data along the eigenvector directions.



Principal Component Analysis (PCA)

- How to choose the principal components?
  - To choose K, use the following criterion:

$$\frac{\sum_{i=1}^{K} \lambda_i}{\sum_{i=1}^{N} \lambda_i} > Threshold \quad \text{(e.g., 0.9 or 0.95)}$$

Principal Component Analysis (PCA)

- What is the error due to dimensionality reduction?
  - We saw above that an original vector *x* can be reconstructed using its principal components:

$$\hat{x} - \bar{x} = \sum_{i=1}^{K} b_i u_i \text{ or } \hat{x} = \sum_{i=1}^{K} b_i u_i + \bar{x}$$

  - It can be shown that the low-dimensional basis based on principal components minimizes the reconstruction error:

$$e = \left\| x - \hat{x} \right\|$$

  - It can be shown that the error is equal to:

$$e = 1/2 \sum_{i=K+1}^{N} \lambda_i$$

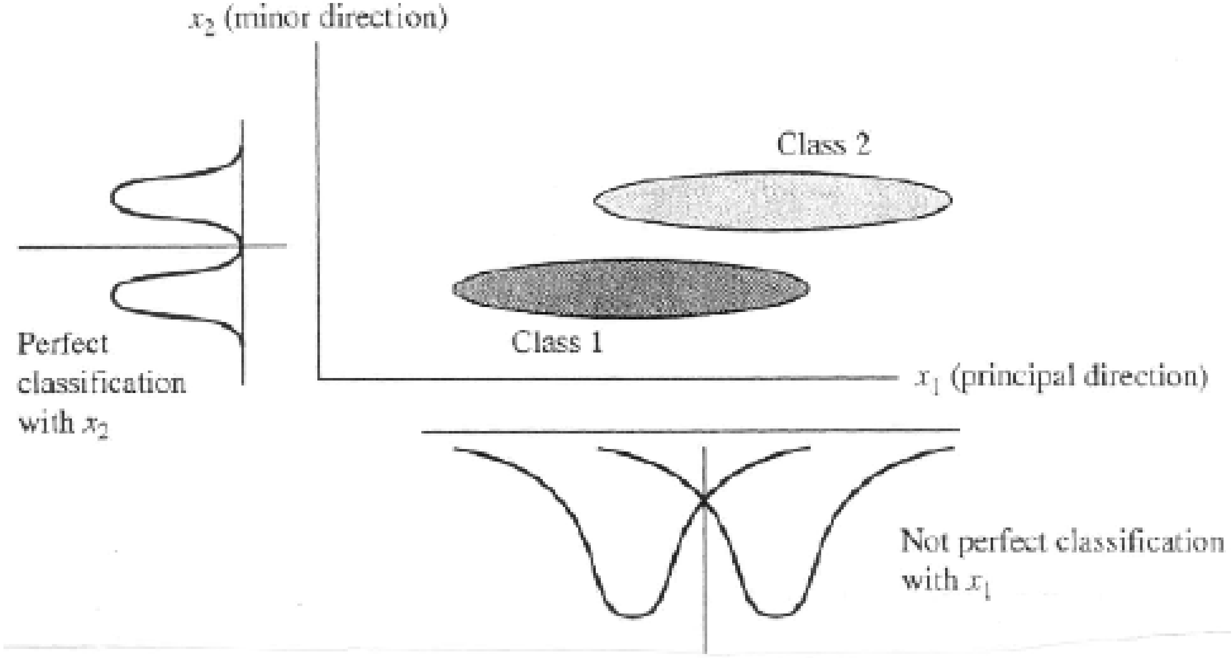
Principal Component Analysis (PCA)

- Standardization
  - The principal components are dependent on the *units* used to measure the original variables as well as on the *range* of values they assume.
  - We should always standardize the data prior to using PCA.
  - A common standardization method is to transform all the data to have zero mean and unit standard deviation:

$$\frac{x_i - \mu}{\sigma} \quad (\mu \text{ and } \sigma \text{ are the mean and standard deviation of } x_i\text{'s})$$

Principal Component Analysis (PCA)

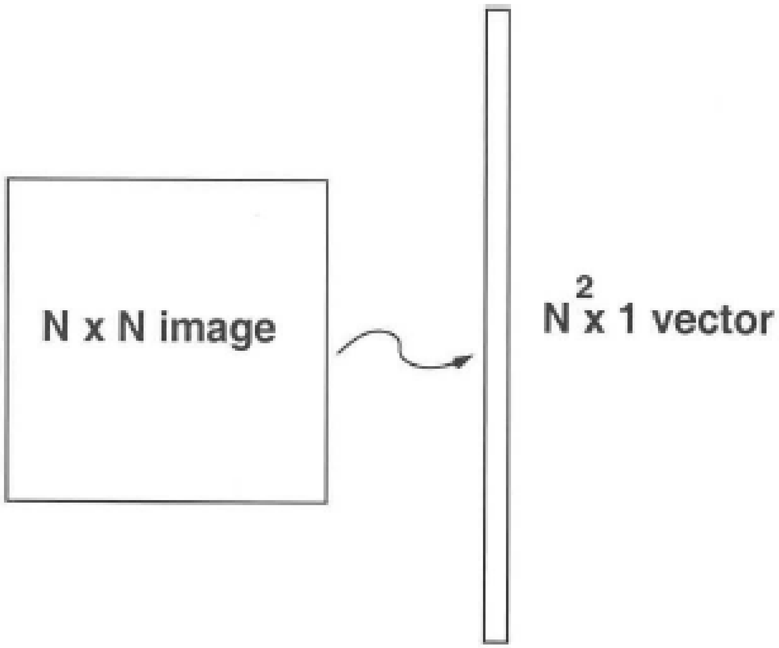
- PCA and classification
  - PCA is **not** always an optimal dimensionality-reduction procedure for classification purposes:



Principal Component Analysis (PCA)

- Case Study: Eigenfaces for Face Detection/Recognition
  - M. Turk, A. Pentland, "Eigenfaces for Recognition", *Journal of Cognitive Neuroscience*, vol. 3, no. 1, pp. 71-86, 1991.

- Face Recognition
  - The simplest approach is to think of it as a template matching problem
  - Problems arise when performing recognition in a high-dimensional space.
  - Significant improvements can be achieved by first mapping the data into a *lower dimensionality* space.
  - How to find this lower-dimensional space?



Principal Component Analysis (PCA)

- Main idea behind eigenfaces

- Suppose Γ is an N^2x1 vector, corresponding to an NxN face image I.

- The idea is to represent Γ (Φ=Γ - mean face) into a low-dimensional space:

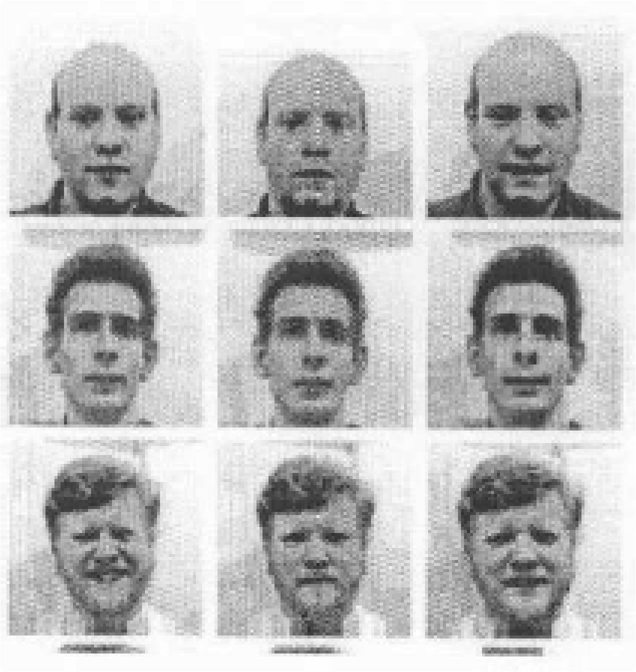
$$\hat{\Phi} - mean = w_1 u_1 + w_2 u_2 + \cdots \; w_K u_K \; (K << N^2)$$

Principal Component Analysis (PCA)

- Computation of the eigenfaces

Step 1: obtain face images $I_1, I_2, ..., I_M$ (training faces)

(**very important:** the face images must be *centered* and of the same *size*)



Step 2: represent every image I_i as a vector Γ_i

Principal Component Analysis (PCA)

- Computation of the eigenfaces – cont.

Step 3: compute the average face vector Ψ:

$$\Psi = \frac{1}{M}\sum_{i=1}^{M}\Gamma_i$$

Step 4: subtract the mean face:

$$\Phi_i = \Gamma_i - \Psi$$

Step 5: compute the covariance matrix C:

$$C = \frac{1}{M}\sum_{n=1}^{M}\Phi_n\Phi_n^T = AA^T \quad (N^2 \text{x} N^2 \text{ matrix})$$

where $A = [\Phi_1 \; \Phi_2 \cdots \Phi_M] \quad (N^2 \text{x} M \text{ matrix})$

Principal Component Analysis (PCA)

- Computation of the eigenfaces – cont.

Step 6: compute the eigenvectors u_i of AA^T

The matrix AA^T is very large --> not practical !!

Step 6.1: consider the matrix $A^T A$ (M x M matrix)

Step 6.2: compute the eigenvectors v_i of $A^T A$

$$A^T A v_i = \mu_i v_i$$

What is the relationship between us_i and v_i?

$A^T A v_i = \mu_i v_i \Rightarrow AA^T A v_i = \mu_i A v_i \Rightarrow$

$CAv_i = \mu_i A v_i$ or $Cu_i = \mu_i u_i$ where $u_i = Av_i$

Thus, AA^T and $A^T A$ have the same eigenvalues and their eigenvectors are related as follows: $u_i = Av_i$!!

Principal Component Analysis (PCA)

- Computation of the eigenfaces – cont.

Note 1: AA^T can have up to N^2 eigenvalues and eigenvectors.

Note 2: $A^T A$ can have up to M eigenvalues and eigenvectors.

Note 3: The M eigenvalues of $A^T A$ (along with their corresponding eigenvectors) correspond to the M *largest* eigenvalues of AA^T (along with their corresponding eigenvectors).

Step 6.3: compute the M best eigenvectors of AA^T: $u_i = Av_i$

(**important:** normalize u_i such that $\|u_i\| = 1$)

Step 7: keep only K eigenvectors (corresponding to the K largest eigenvalues)

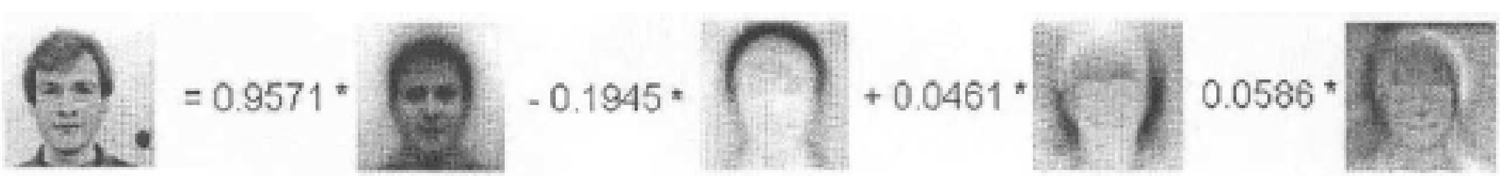
Principal Component Analysis (PCA)

- Representing faces onto this basis

- Each face (minus the mean) Φ_i in the training set can be represented as a linear combination of the best K eigenvectors:

$$\hat{\Phi}_i - mean = \sum_{j=1}^{K} w_j u_j, \quad (w_j = u_j^T \Phi_i)$$

(we call the u_j's *eigenfaces*)



Principal Component Analysis (PCA)

- Representing faces onto this basis – cont.

- Each normalized training face Φ_i is represented in this basis by a vector:

$$\Omega_i = \begin{bmatrix} w_1^i \\ w_2^i \\ \cdots \\ w_K^i \end{bmatrix}, \quad i = 1, 2, \ldots, M$$

Principal Component Analysis (PCA)

- Face Recognition Using Eigenfaces

- Given an unknown face image Γ (centered and of the same size like the training faces) follow these steps:

Step 1: normalize Γ: $\Phi = \Gamma - \Psi$

Step 2: project on the eigenspace

$$\hat{\Phi} = \sum_{i=1}^{K} w_i u_i \quad (w_i = u_i^T \Phi)$$

Step 3: represent Φ as: $\Omega = \begin{bmatrix} w_1 \\ w_2 \\ \cdots \\ w_K \end{bmatrix}$

Step 4: find $e_r = \min_l \|\Omega - \Omega^l\|$

Step 5: if $e_r < T_r$, then Γ is recognized as face l from the training set.

Principal Component Analysis (PCA)

- Face Recognition Using Eigenfaces – cont.
  - The distance e_r is called *distance within the face space* (*difs*)
  - Comment: we can use the common Euclidean distance to compute e_r, however, it has been reported that the *Mahalanobis distance* performs better:

$$\|\Omega - \Omega^k\| = \sum_{i=1}^{K} \frac{1}{\lambda_i} (w_i - w_i^k)^2$$

Principal Component Analysis (PCA)

- Problems
  - Background (de-emphasize the outside of the face – e.g., by multiplying the input image by a 2D Gaussian window centered on the face)
  - Lighting conditions (performance degrades with light changes)
  - Scale (performance decreases quickly with changes to head size)
    - multi-scale eigenspaces
    - scale input image to multiple sizes
  - Orientation (performance decreases but not as fast as with scale changes)
    - plane rotations can be handled
    - out-of-plane rotations are more difficult to handle

Linear Discriminant Analysis (LDA)

- Multiple classes and PCA
  - Suppose there are *C* classes in the training data.
  - PCA is based on the sample covariance which characterizes the scatter of the entire data set, *irrespective of class-membership*.
  - The projection axes chosen by PCA might not provide good discrimination power.

- What is the goal of LDA?
  - Perform dimensionality reduction while preserving as much of the class discriminatory information as possible.
  - Seeks to find directions along which the classes are best separated.
  - Takes into consideration the scatter *within-classes* but also the scatter *between-classes*.
  - More capable of distinguishing image variation due to identity from variation due to other sources such as illumination and expression.

Linear Discriminant Analysis (LDA)

- Methodology

- Suppose there are C classes
- Let $\boldsymbol{\mu}_i$ be the mean vector of class i, $i = 1, 2, .., C$
- Let M_i be the number of samples within class i, $i = 1, 2, .., C$,
- Let $M = \sum_{i=0}^{C} M_i$ be the total number of samples. and

Within-class scatter matrix:

$$S_w = \sum_{i=1}^{C} \sum_{j=1}^{M_i} (y_j - \boldsymbol{\mu}_i)(y_j - \boldsymbol{\mu}_i)^T$$

Between-class scatter matrix:

$$S_b = \sum_{i=1}^{C} (\boldsymbol{\mu}_i - \boldsymbol{\mu})(\boldsymbol{\mu}_i - \boldsymbol{\mu})^T$$

$$\boldsymbol{\mu} = 1/C \sum_{i=1}^{C} \boldsymbol{\mu}_i \quad \text{(mean of entire data set)}$$

Linear Discriminant Analysis (LDA)

- Methodology – cont.
  - LDA computes a transformation that maximizes the between-class scatter while minimizing the within-class scatter:

$$\text{maximize } \frac{det(S_b)}{det(S_w)}$$

  - Such a transformation should retain class separability while reducing the variation due to sources other than identity (e.g., illumination).

Linear Discriminant Analysis (LDA)

- Linear transformation implied by LDA
  - The linear transformation is given by a matrix *U* whose columns are the eigenvectors of $S_w^{-1} S_b$ (called *Fisherfaces*).

$$\begin{bmatrix} b_1 \\ b_2 \\ \dots \\ b_K \end{bmatrix} = \begin{bmatrix} u_1^T \\ u_2^T \\ \dots \\ u_K^T \end{bmatrix} (x - \boldsymbol{\mu}) = U^T (x - \boldsymbol{\mu})$$

  - The eigenvectors are solutions of the *generalized eigenvector problem*:

$$S_B u_k = \boldsymbol{\lambda}_k S_w u_k$$

Linear Discriminant Analysis (LDA)

- Does S_w^{-1} always exist?
  - If S_w is non-singular, we can obtain a conventional eigenvalue problem by writing:

$$S_w^{-1} S_B u_k = \lambda_k u_k$$

  - In practice, S_w is often singular since the data are image vectors with large dimensionality while the size of the data set is much smaller ($M << N$)

Linear Discriminant Analysis (LDA)

- Does S_w^{-1} always exist? – cont.
  - To alleviate this problem, we can perform two projections:

1) PCA is first applied to the data set to reduce its dimensionality.

$$\begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_N \end{bmatrix} --> PCA --> \begin{bmatrix} y_1 \\ y_2 \\ \dots \\ y_K \end{bmatrix}$$

2) LDA is then applied to further reduce the dimensionality.

$$\begin{bmatrix} y_1 \\ y_2 \\ \dots \\ y_K \end{bmatrix} --> LDA --> \begin{bmatrix} z_1 \\ z_2 \\ \dots \\ z_{C-1} \end{bmatrix}$$

Linear Discriminant Analysis (LDA)

- Assumptions
  - "Well-framed" images are required as input for training and query-by-example test probes.
  - Only a small variation in the size, position, and orientation of the objects in the images is allowed.

Linear Discriminant Analysis (LDA)

- Some terminology
  - Most Expressive Features (MEF): the features (projections) obtained using PCA.
  - Most Discriminating Features (MDF): the features (projections) obtained using LDA.

- Computational considerations
  - When computing the eigenvalues/eigenvectors of $S_w^{-1} S_B u_k = \lambda_k u_k$ numerically, the computations can be unstable since $S_w^{-1} S_B$ is not always symmetric.
  - See paper for a way to find the eigenvalues/eigenvectors in a stable way.
  - Important: Dimensionality of LDA is bounded by C-1 --- this is the rank of $S_w^{-1} S_B$

Linear Discriminant Analysis (LDA)

- Factors unrelated to classification
  - MEF vectors show the tendency of PCA to capture major variations in the training set such as lighting direction.
  - MDF vectors discount those factors unrelated to classification.

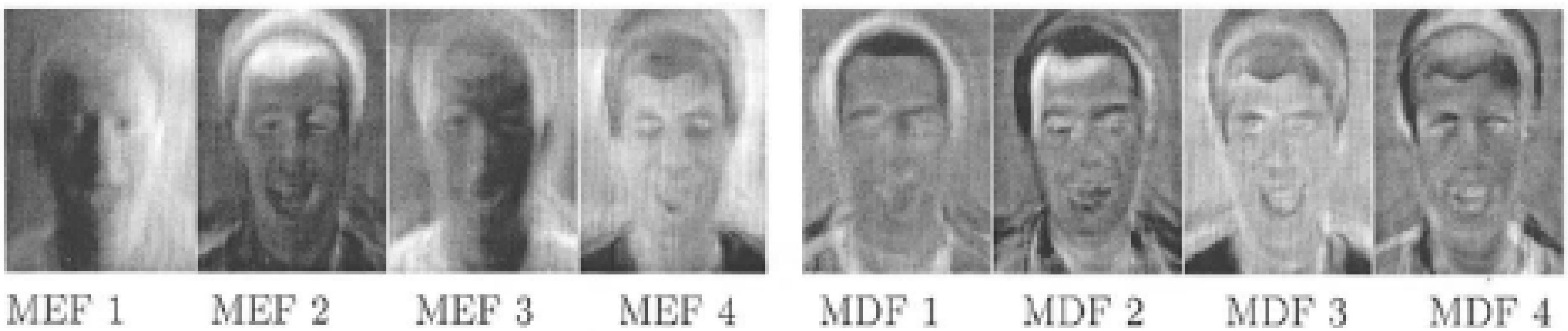


Figure 2. A sample of MEF and MDF vectors treated as images. The MEF vectors show the tendency of the principal components to capture major variations in the training set, such as lighting direction. The MDF vectors show the ability of the MDFs to discount those factors unrelated to classification. The training images used to produce these vectors are courtesy of the Weizmann Institute.

Linear Discriminant Analysis (LDA)

- Case Study: PCA versus LDA
  - A. Martinez, A. Kak, "PCA versus LDA", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 2, pp. 228-233, 2001.
- Is LDA always better than PCA?
  - There has been a tendency in the computer vision community to prefer LDA over PCA.
  - This is mainly because LDA deals directly with discrimination between classes while PCA does not pay attention to the underlying class structure.
  - This paper shows that when the training set is small, PCA can outperform LDA.
  - When the number of samples is large and representative for each class, LDA outperforms PCA.

Linear Discriminant Analysis (LDA)

- Critique of LDA
  - Only linearly separable classes will remain separable after applying LDA.
  - It does not seem to be superior to PCA when the training data set is small.

Questions

References

[1] M. Turk, A. Pentland, “Face Recognition Using Eigenfaces”

[2] J. Ashbourn, Avanti, V. Bruce, A. Young, ”Face Recognition Based on Symmetrization and Eigenfaces”

[3] http://www.markus-hofmann.de/eigen.html

[4] P. Belhumeur, J. Hespanha, D. Kriegman, “Eigenfaces vs Fisherfaces: Recognition using Class Specific Linear Projection”

[5] R. Duda, P. Hart, D. Stork, “Pattern Classification”, ISBN 0-471-05669-3, pp. 121-124

[6] F. Perronin, J.-L. Dugelay, “Deformable Face Mapping For Person Identification”, ICIP 2003, Barcelona

Credits

Slides adapted from:

- George Bebis
- Alexander Roth