Fisherfaces Vs Eigenfaces

Eigenfaces (based on PCA)

- Many features of the human face are heavily correlated and therefore provide only redundant information.
- PCA acts as a dimensionality reduction algorithm by creating a new space of features which can provide the maximum discrimination amongst images.
- It does so by projecting data points along the axis of the maximum variation (PC1), followed by the second maximum variation axis (PC2) and so on.
- Only the top k-axes are chosen keeping in mind the accuracy required and resource constraints.

Mathematics behind PCA

Transformation matrix W is given by :

$$\mathbf{y}_k = W^T \mathbf{x}_k \qquad k = 1, 2, \ldots, N$$

Covariance matrix is given by :

$$S_T = \sum_{k=1}^{N} (\mathbf{x}_k - \boldsymbol{\mu})(\mathbf{x}_k - \boldsymbol{\mu})^T$$

We need to choose the transformation matrix so as to maximize the variance , which is given by :

$$\begin{aligned} W_{opt} &= \arg\max_{W} \left| W^T S_T W \right| \\ &= \begin{bmatrix} \mathbf{w}_1 & \mathbf{w}_2 & \dots & \mathbf{w}_m \end{bmatrix} \end{aligned}$$

Each column of the transformation matrix turns out to be an eigenvector of the total scatter matrix

Drawbacks of Eigenfaces

- We work with the total scatter matrix in this case. This contains the between class scatter (useful for classification) but also the within class scatter (unwanted information).
- Much of variation from one image to next is due to illumination changes.
- As a result, variation in lighting/shadowing creeps into the outcome if only eigenfaces are used.

Linear Discriminant Analysis

- LDA algorithm aims to maximize separability between classes and minimize variance within classes
- Leads to the concept of within class and between class scatter matrix
- LDA does this by maximizing the ratio of the determinant of between class scatter matrix to that of the within class scatter matrix

Mathematics behind LDA

Between class scatter matrix is given by:

$$S_B = \sum_{i=1}^{c} N_i(\boldsymbol{\mu}_i - \boldsymbol{\mu})(\boldsymbol{\mu}_i - \boldsymbol{\mu})^T$$

Within class scatter matrix is given by:

$$S_W = \sum_{i=1}^{c} \sum_{\mathbf{x}_k \in X_i} (\mathbf{x}_k - \boldsymbol{\mu}_i)(\mathbf{x}_k - \boldsymbol{\mu}_i)^T$$

Therefore transformation matrix is given by:

$$
\begin{aligned}
W_{\text{opt}} &= \arg\max_{W} \frac{\left|W^T S_B W\right|}{\left|W^T S_W W\right|} \\
&= \begin{bmatrix} \mathbf{w}_1 & \mathbf{w}_2 & \ldots & \mathbf{w}_m \end{bmatrix}
\end{aligned}
$$

This reduces into solving the following problem:

$$
S_B \mathbf{w}_i = \lambda_i S_W \mathbf{w}_i, \qquad i = 1, 2, \ldots, m
$$

- Generally Sw matrix (n x n) tends to be singular as the rank is at most N-c because of the dimension of an image being much larger than the number of test images.
- In this case PCA is used for dimensionality reduction
- It throws away the least useful c-1 principal components
- Then LDA is applied to further reduce dimensionality to c-1.

The new equation is given by:

$$W_{opt}^T = W_{fld}^T W_{pca}^T$$

where

$$W_{pca} = \arg\max_W \left| W^T S_T W \right|$$

$$W_{fld} = \arg\max_W \frac{\left| W^T W_{pca}^T S_B W_{pca} W \right|}{\left| W^T W_{pca}^T S_W W_{pca} W \right|}$$

Therefore PCA smears out class differences while LDA retains it .

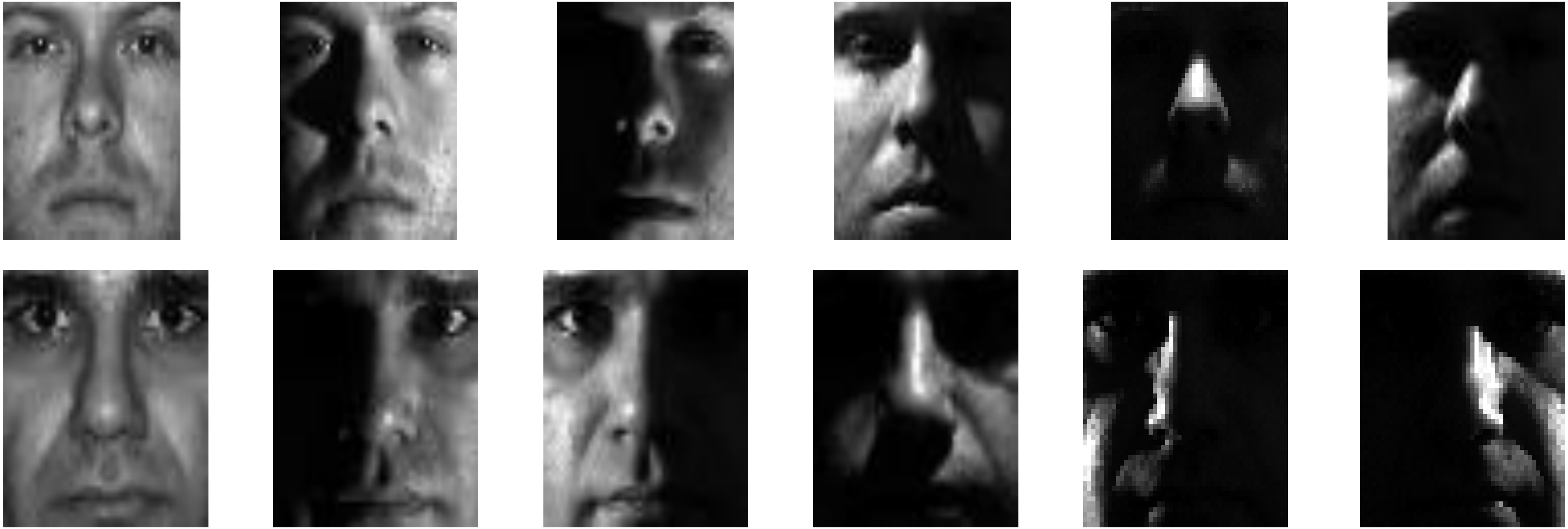
PCA maximizes overall scatter while LDA maximizes between class scatter for better classification

Implementation

Dataset

- **We have used a dataset with 10 classes (persons) each having 64 images**
- **Within each class the subject is placed under different lighting/shadowing conditions**
- **Hence this dataset is ideal to show difference between eigenfaces and fisherfaces**

Example Faces in Dataset



Accuracy Comparison

Using Eigen Faces

"Number of eigenvectors = 10"

R =

6×6 string array

"Train/test "	"Subset 1"	"Subset 2"	"Subset 3"	"Subset 4"	"Subset 5"
"Subset 1"	"100"	"74.1667"	"27.5"	"12.1429"	"10"
"Subset 2"	"61.4286"	"100"	"28.3333"	"15"	"13.6842"
"Subset 3"	"8.5714"	"14.1667"	"100"	"16.4286"	"14.7368"
"Subset 4"	"10"	"8.3333"	"13.3333"	"100"	"10.5263"
"Subset 5"	"4.2857"	"9.1667"	"19.1667"	"19.2857"	"100"

H2 =

"Number of eigenvectors = 31"

S =

6×6 string array

"Train/test "	"Subset 1"	"Subset 2"	"Subset 3"	"Subset 4"	"Subset 5"
"Subset 1"	"100"	"74.1667"	"27.5"	"12.1429"	"10"
"Subset 2"	"61.4286"	"100"	"30.8333"	"15.7143"	"13.1579"
"Subset 3"	"8.5714"	"14.1667"	"100"	"16.4286"	"14.7368"
"Subset 4"	"10"	"8.3333"	"15"	"100"	"11.5789"
"Subset 5"	"4.2857"	"9.1667"	"19.1667"	"22.1429"	"100"

Accuracy Comparison

Using Fisher Faces

"Number of eigenvectors = 10"

R =

6×6 string array

"Train/test "	"Subset 1"	"Subset 2"	"Subset 3"	"Subset 4"	"Subset 5"
"Subset 1"	"100"	"95.8333"	"43.3333"	"15"	"12.6316"
"Subset 2"	"88.5714"	"100"	"67.5"	"25.7143"	"22.6316"
"Subset 3"	"87.1429"	"89.1667"	"100"	"82.1429"	"20"
"Subset 4"	"20"	"26.6667"	"60"	"100"	"22.1053"
"Subset 5"	"8.5714"	"10.8333"	"20"	"35.7143"	"100"

H2 =

"Number of eigenvectors = 31"

S =

6×6 string array

"Train/test "	"Subset 1"	"Subset 2"	"Subset 3"	"Subset 4"	"Subset 5"
"Subset 1"	"100"	"98.3333"	"45.8333"	"16.4286"	"10.5263"
"Subset 2"	"92.8571"	"100"	"65"	"17.1429"	"12.6316"
"Subset 3"	"61.4286"	"74.1667"	"100"	"77.1429"	"28.4211"
"Subset 4"	"14.2857"	"17.5"	"61.6667"	"100"	"27.3684"
"Subset 5"	"15.7143"	"19.1667"	"35.8333"	"68.5714"	"100"

References

- https://cseweb.ucsd.edu/classes/wi14/cse152-a/fisherface-pami97.pdf
- https://iopscience.iop.org/article/10.1088/1742-6596/1028/1/012119/pdf