Lecture 9.
Face Recognition

Pattern Recognition and Computer Vision

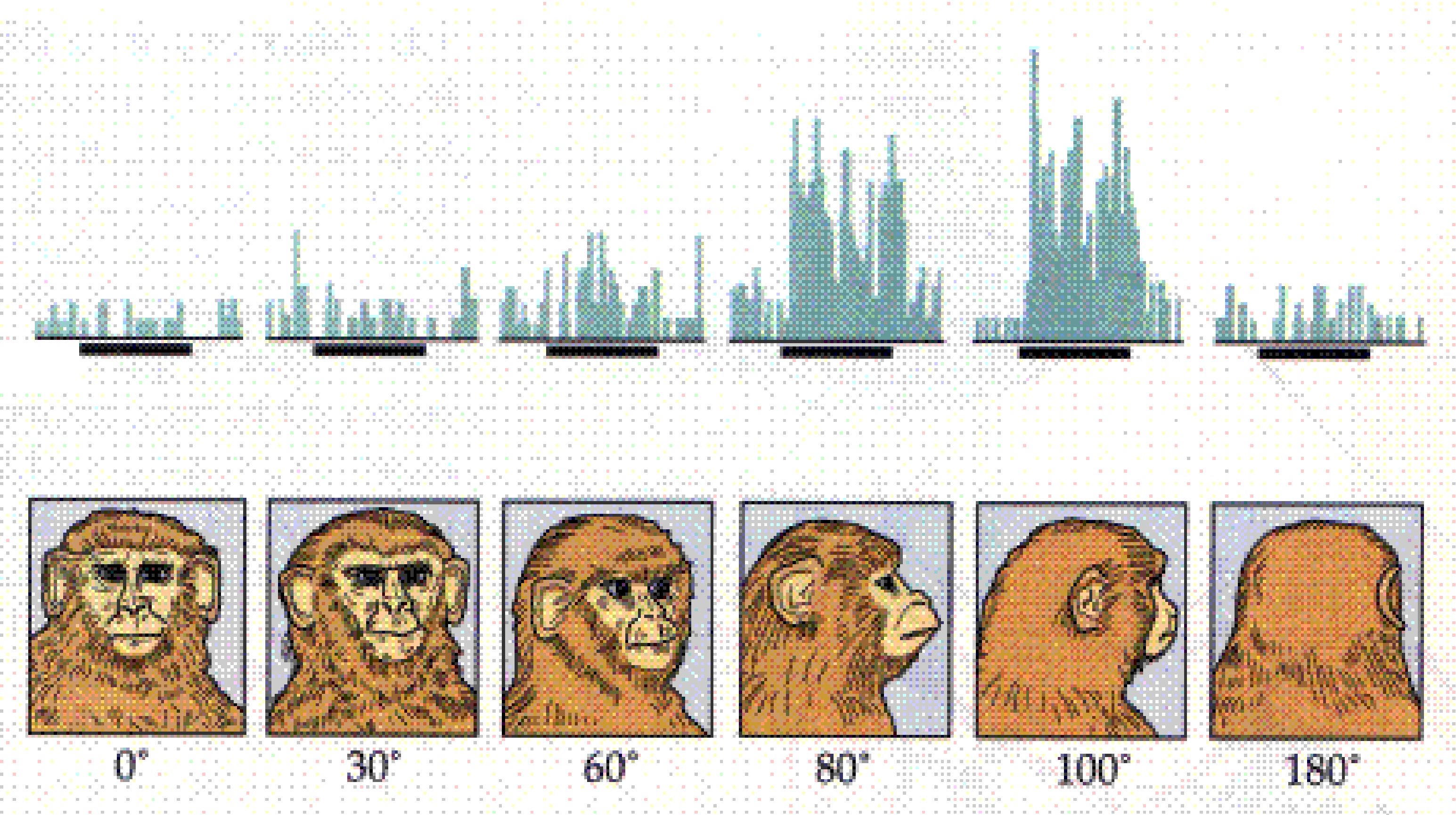
Guanbin Li,
School of Data and Computer Science, Sun Yat-Sen University

扫码签到

What we will learn today?

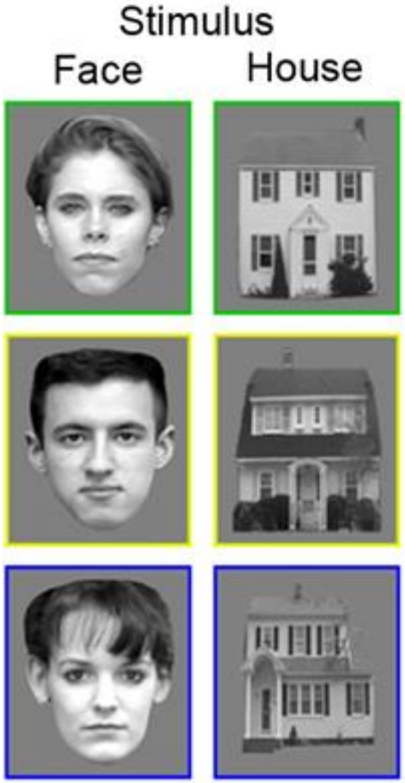
- Introduction to face recognition
- The Eigenfaces Algorithm
- Linear Discriminant Analysis (LDA)

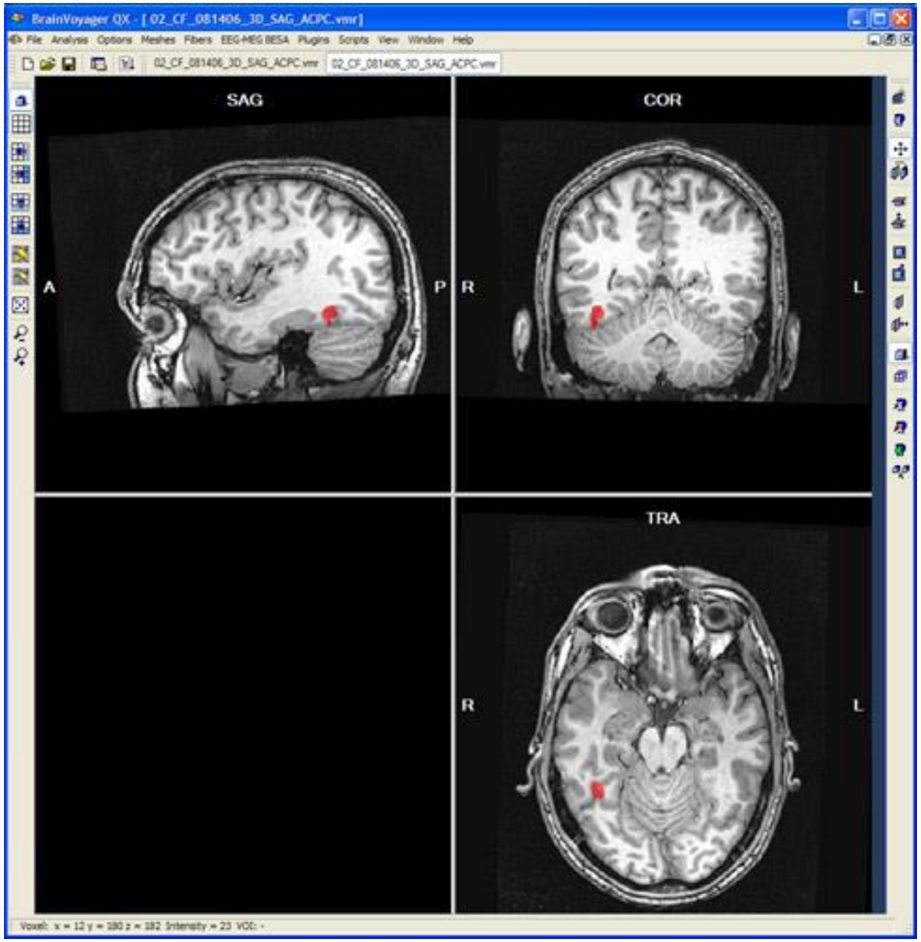
“Faces” in the brain



Courtesy of Johannes M. Zanker

“Faces” in the fusiform face area



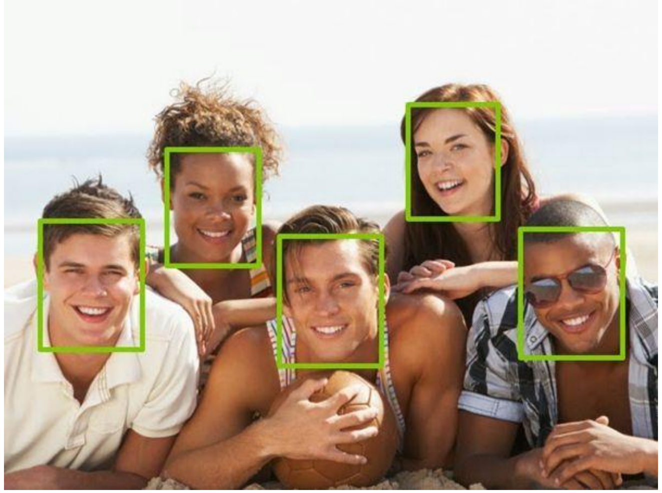

BrainVoyager QX - [02_CF_081406_3D_SAG_ACPC.vmr]
File
Analysis
Options
Meshes
Fibers
EEG-MEG BESA
Plugins
Scripts
View
Window
Help
02_CF_081406_3D_SAG_ACPC.vmr
02_CF_081406_3D_SAG_ACPC.vmr
SAG
COR
TRA
A
P
R
L
R
L

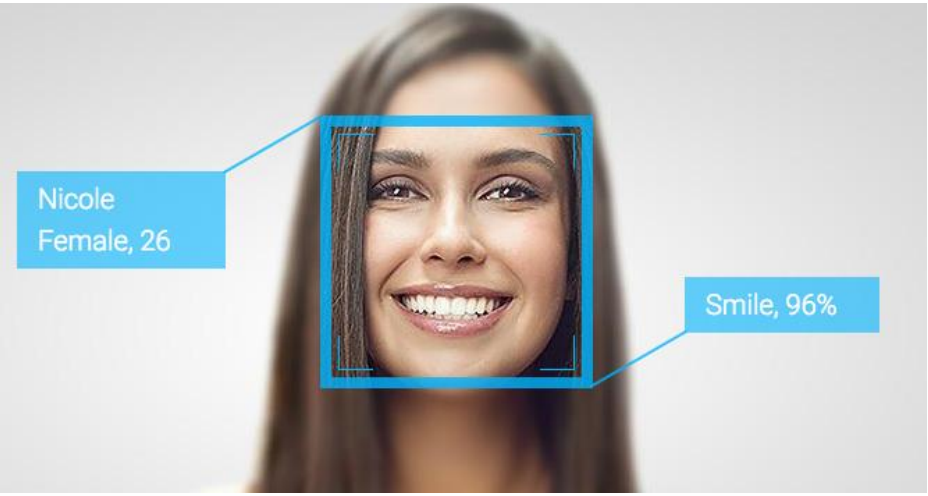
Voxel: x = 12 y = 180 z = 182 Intensity = 23 VOI: -

梭形面部区域

Kanwisher, et al. 1997

Detection vs. Recognition



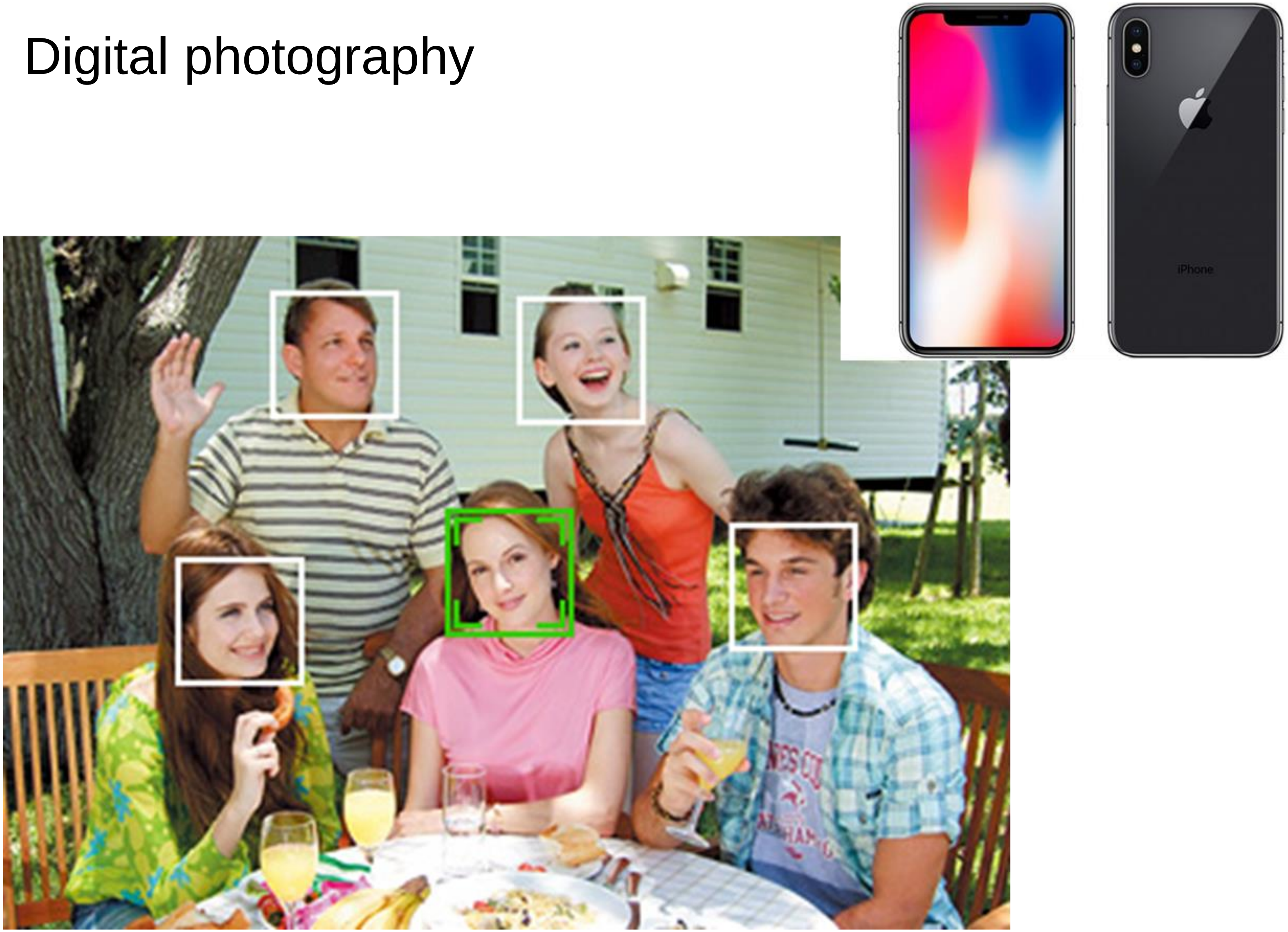


Detection finds the faces in images

Recognition recognizes WHO the person is

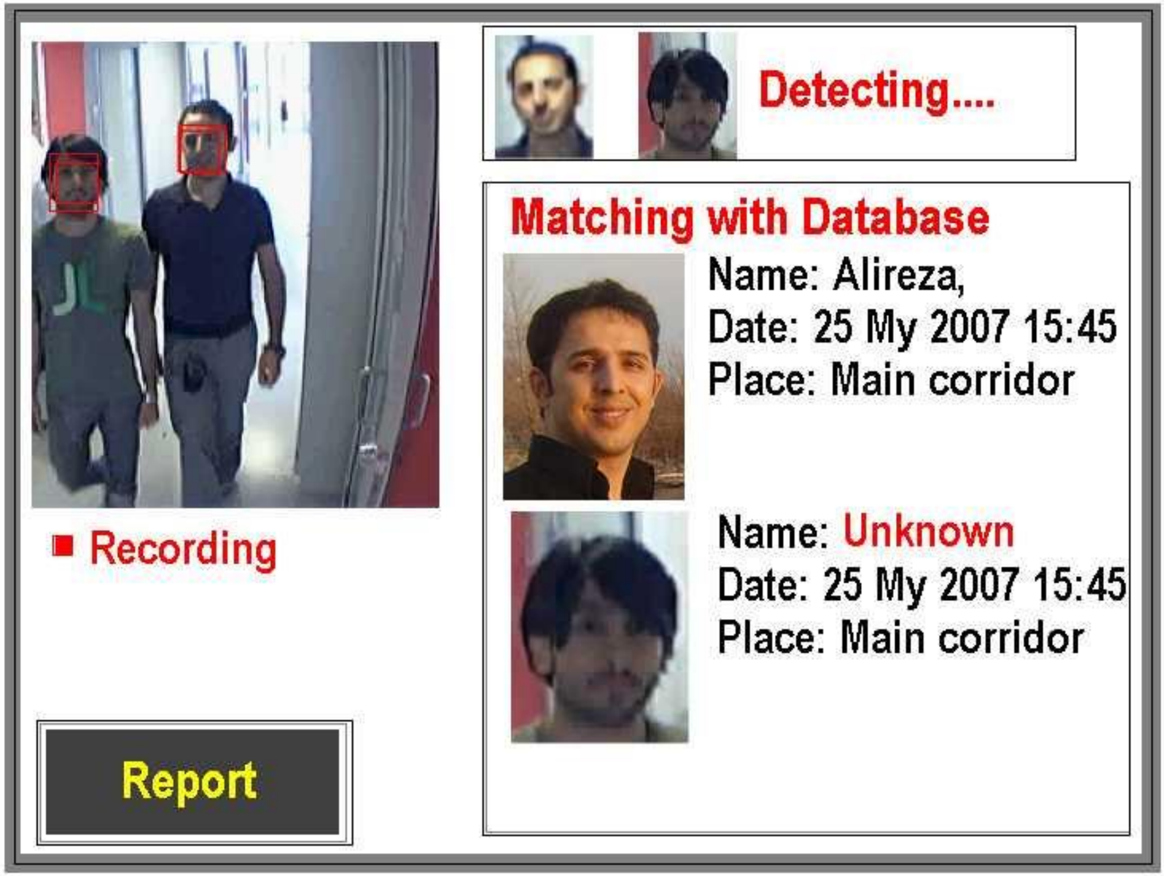
Applications of Face Recognition

- Digital photography



Applications of Face Recognition

- Digital photography
- Surveillance



Name: Alireza,
Date: 25 My 2007 15:45
Place: Main corridor

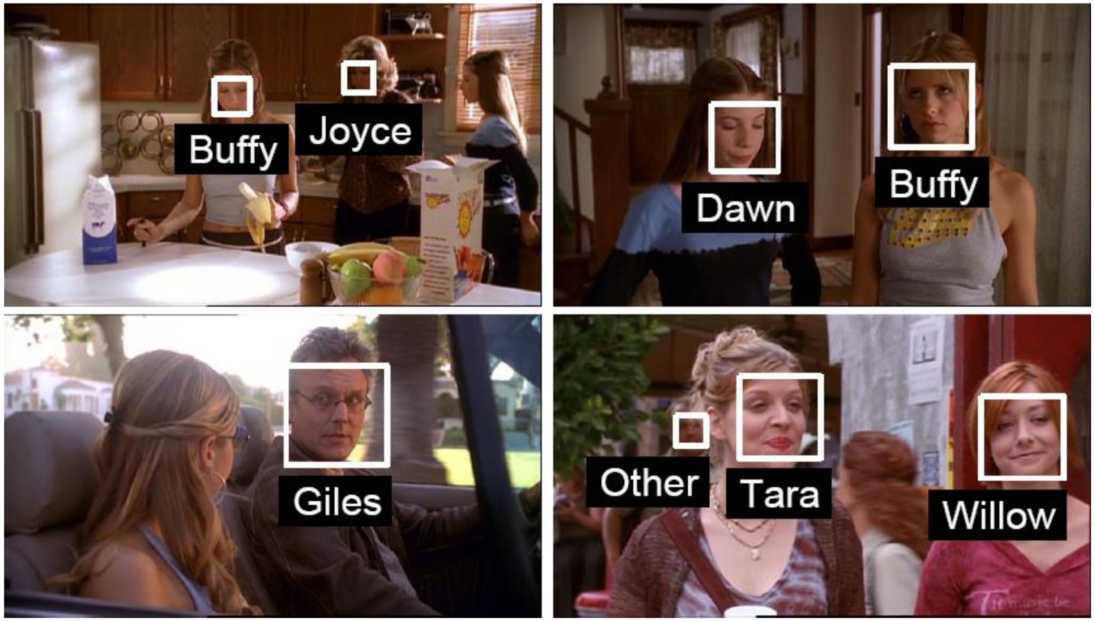
Name: Unknown
Date: 25 My 2007 15:45
Place: Main corridor

Applications of Face Recognition

- Digital photography
- Surveillance
- Album organization

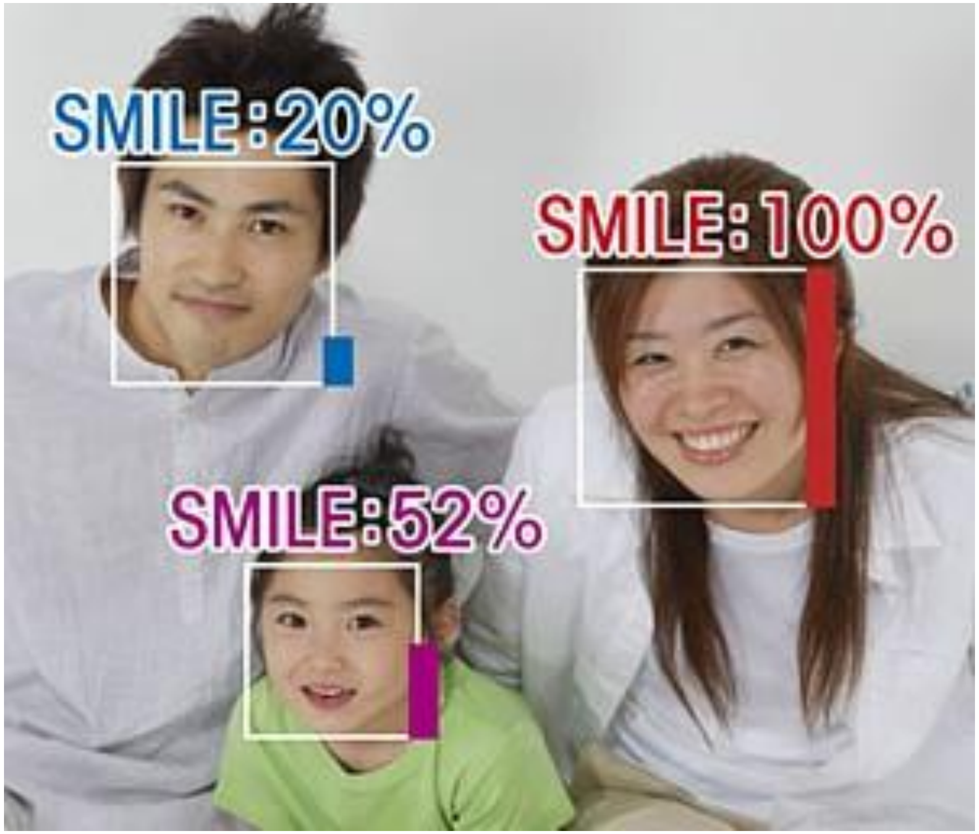
Applications of Face Recognition

- Digital photography
- Surveillance
- Album organization
- Person tracking/id.



Applications of Face Recognition

- Digital photography
- Surveillance
- Album organization
- Person tracking/id.
- Emotions and expressions

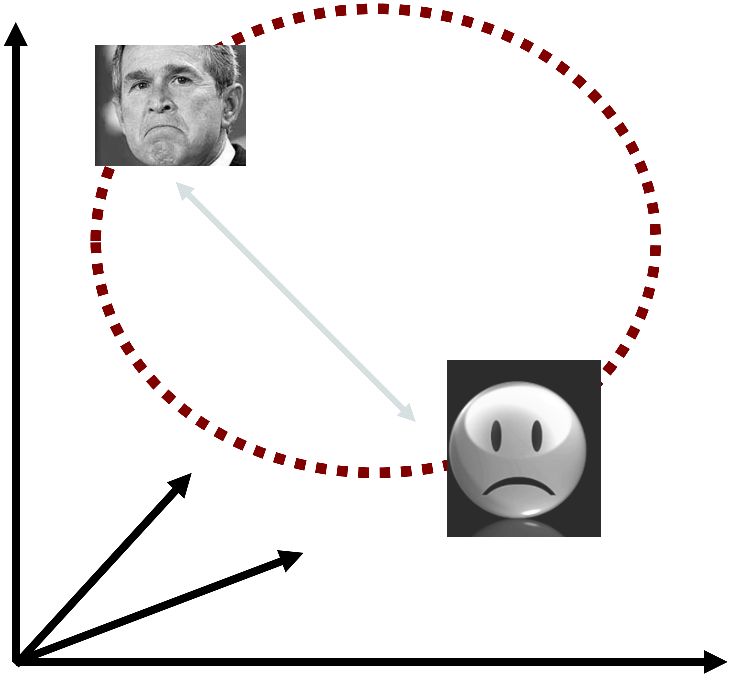


Applications of Face Recognition

- Digital photography
- Surveillance
- Album organization
- Person tracking/id.
- Emotions and expressions
- Security/warfare
- Tele-conferencing
- Etc.

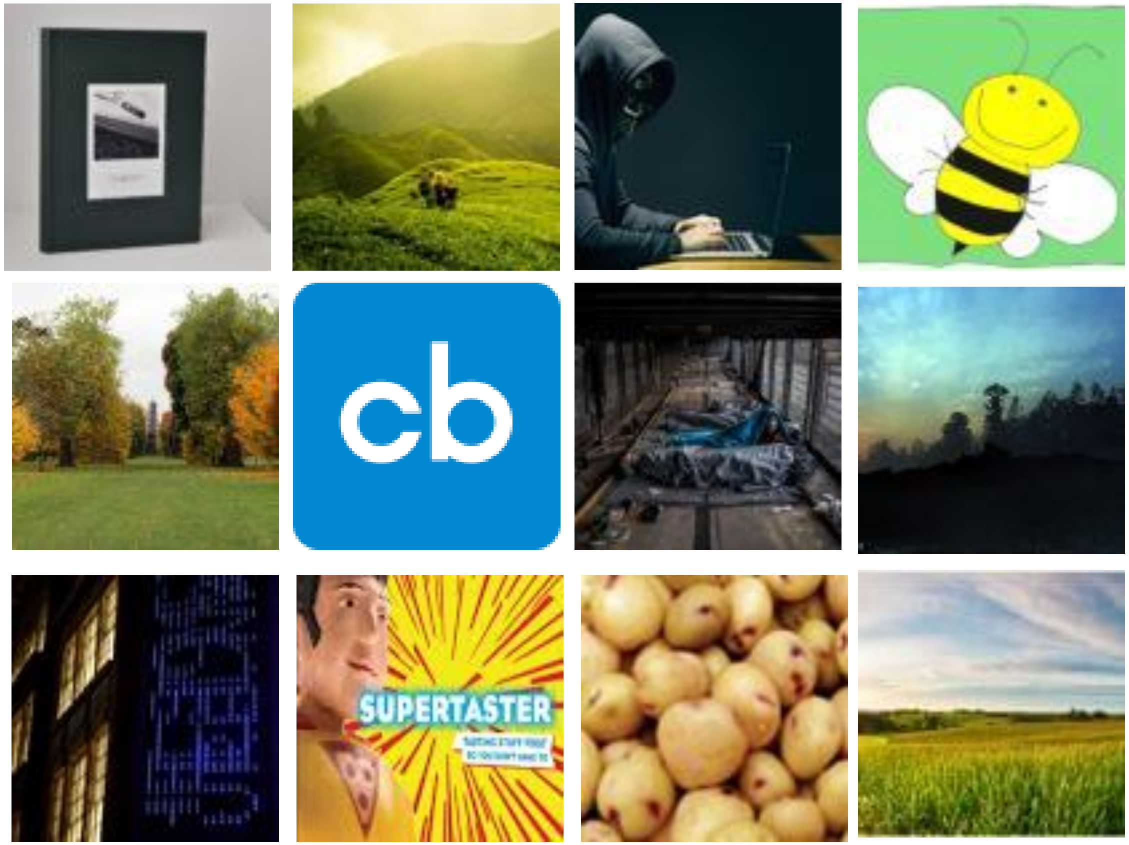
The Space of Faces

- An image is a point in a high dimensional space
  - If represented in grayscale intensity, an N x M image is a point in R^{NM}
  - E.g. 100x100 image = 10,000 dim



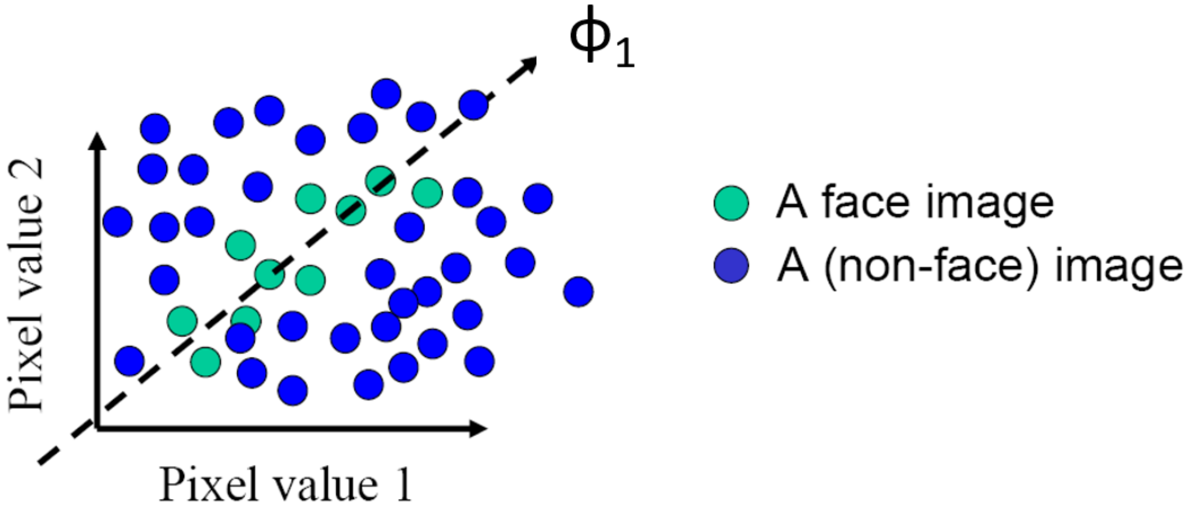
The Space of Faces

100x100 images can contain many things other than faces!



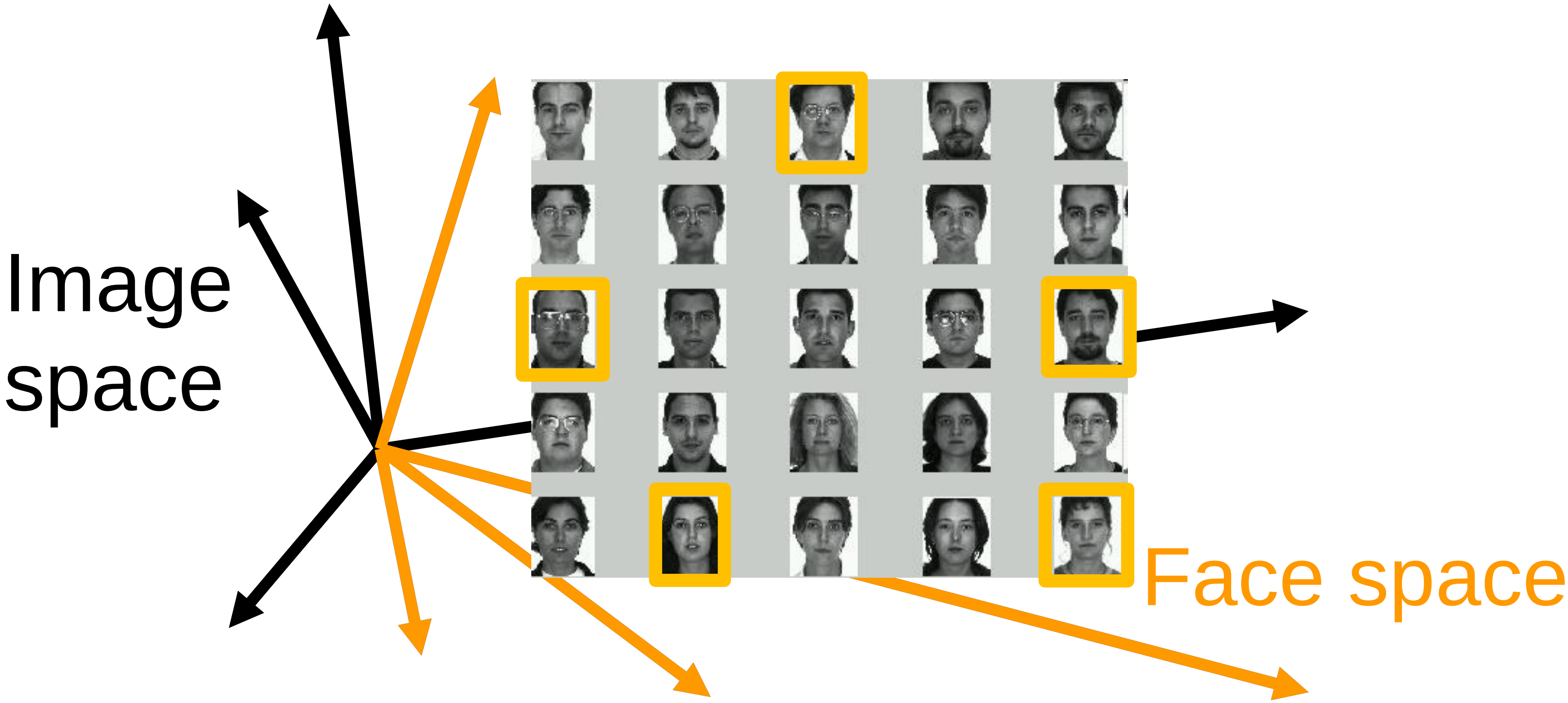
The Space of Faces

- An image is a point in a high dimensional space
  - If represented in grayscale intensity, an N x M image is a point in R^{NM}
  - E.g. 100x100 image = 10,000 dim
- However, relatively few high dimensional vectors correspond to valid face images
- We want to effectively model the subspace of face images



The Space of Faces

- Compute n-dim subspace such that the projection of the data points onto the subspace has the largest variance among all n-dim subspaces.
- Maximize the scatter of the training images in face space.



What we will learn today?

- Introduction to face recognition
- The Eigenfaces Algorithm
- Linear Discriminant Analysis (LDA)

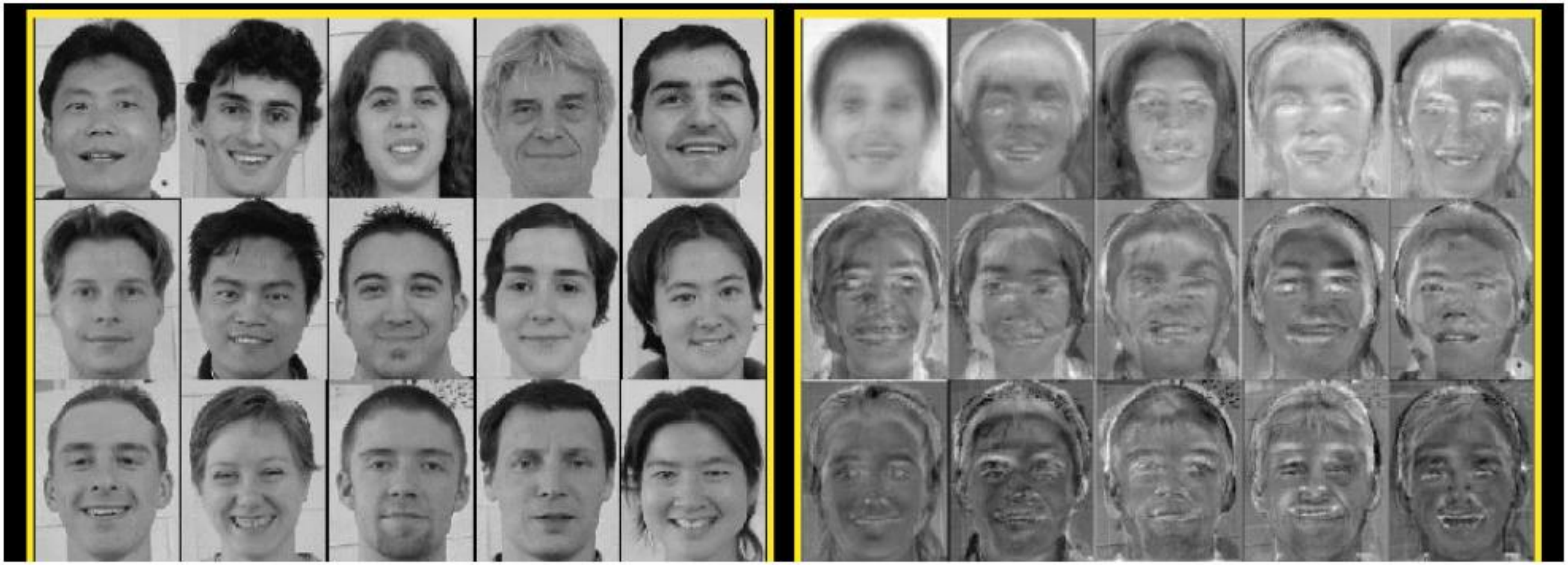
Eigenfaces: key idea

- Assume that most face images lie on a low-dimensional subspace determined by the first k (k<<d) directions of maximum variance
- Use PCA to determine the vectors or “eigenfaces” that span that subspace
- Represent all face images in the dataset as linear combinations of eigenfaces

Turk and Pentland, Eigenfaces for Recognition, *Journal of Cognitive Neuroscience* **3** (1): 71–86.

What are eigenfaces?

- "Eigenfaces" are the visual representations of the eigenvectors in the directions of maximum variance. They often resemble generic-looking faces.



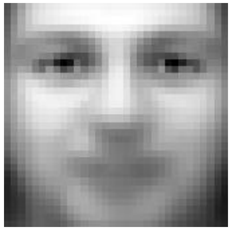
Visualization of eigenfaces

Eigenface algorithm

- Training
  1. Align training images $x_1, x_2, \ldots, x_N$

Note that each image is formulated into a long vector!

  2. Compute average face $m=\frac{1}{N}\mathring{a}\, x_i$



μ

  3. Compute the difference image (the centered data matrix):

$$X_c = \begin{bmatrix} | & & | \\ x_1 & \ldots & x_N \\ | & & | \end{bmatrix} - \begin{bmatrix} | & & | \\ \mu & \ldots & \mu \\ | & & | \end{bmatrix}$$

$$= X - \mu 1^T = X - \frac{1}{N} X 1 1^T = X(I - \frac{1}{N} 1 1^T)$$

Eigenface algorithm

4. Compute the covariance matrix

$$\Sigma = \begin{bmatrix} | & & | \\ x_1^c & \cdots & x_N^c \\ | & & | \end{bmatrix} - \begin{bmatrix} - & x_1^c & - \\ & \cdots & \\ - & x_N^c & - \end{bmatrix} = \frac{1}{N} X_c X_c^T$$

5. Compute the eigenvectors of the covariance matrix Σ using PCA
6. Compute each training image x_i's projections as

$$x_i \rightarrow \left(x_i^c \cdot f_1,\, x_i^c \cdot f_2, \ldots,\ x_i^c \cdot f_K \right) \equiv \left(a_1, a_2, \ldots, a_K \right)$$

where ϕ_i the i'th-highest ranked eigenvector

7. Visualize the estimated training face x_i

$$x_i \text{ » } m + a_1 f_1 + a_2 f_2 + \ldots + a_K f_K$$

Eigenface algorithm

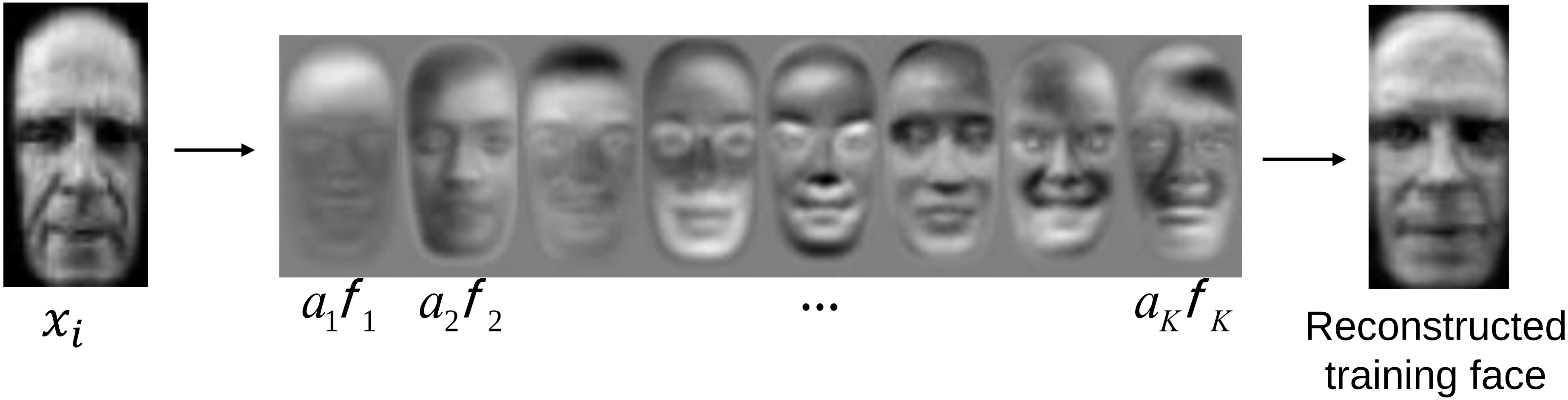
6. Compute each training image x_i's projections as

$$x_i \rightarrow \left(x_i^c \cdot f_1,\, x_i^c \cdot f_2, \ldots,\ x_i^c \cdot f_K\right) \equiv \left(a_1, a_2, \ldots, a_K\right)$$

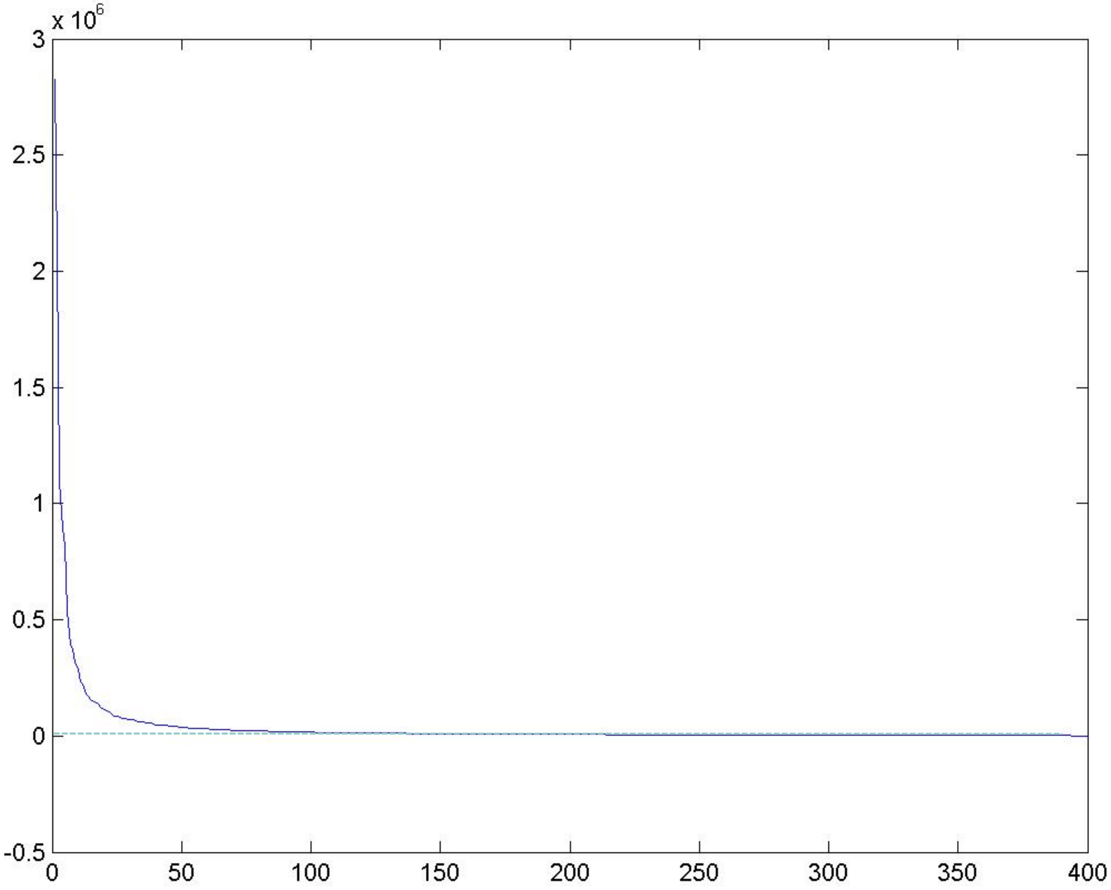
where ϕ_i the i'th-highest ranked eigenvector

7. Visualize the estimated training face x_i

$$x_i \,\text{»}\, m + a_1 f_1 + a_2 f_2 + \ldots + a_K f_K$$

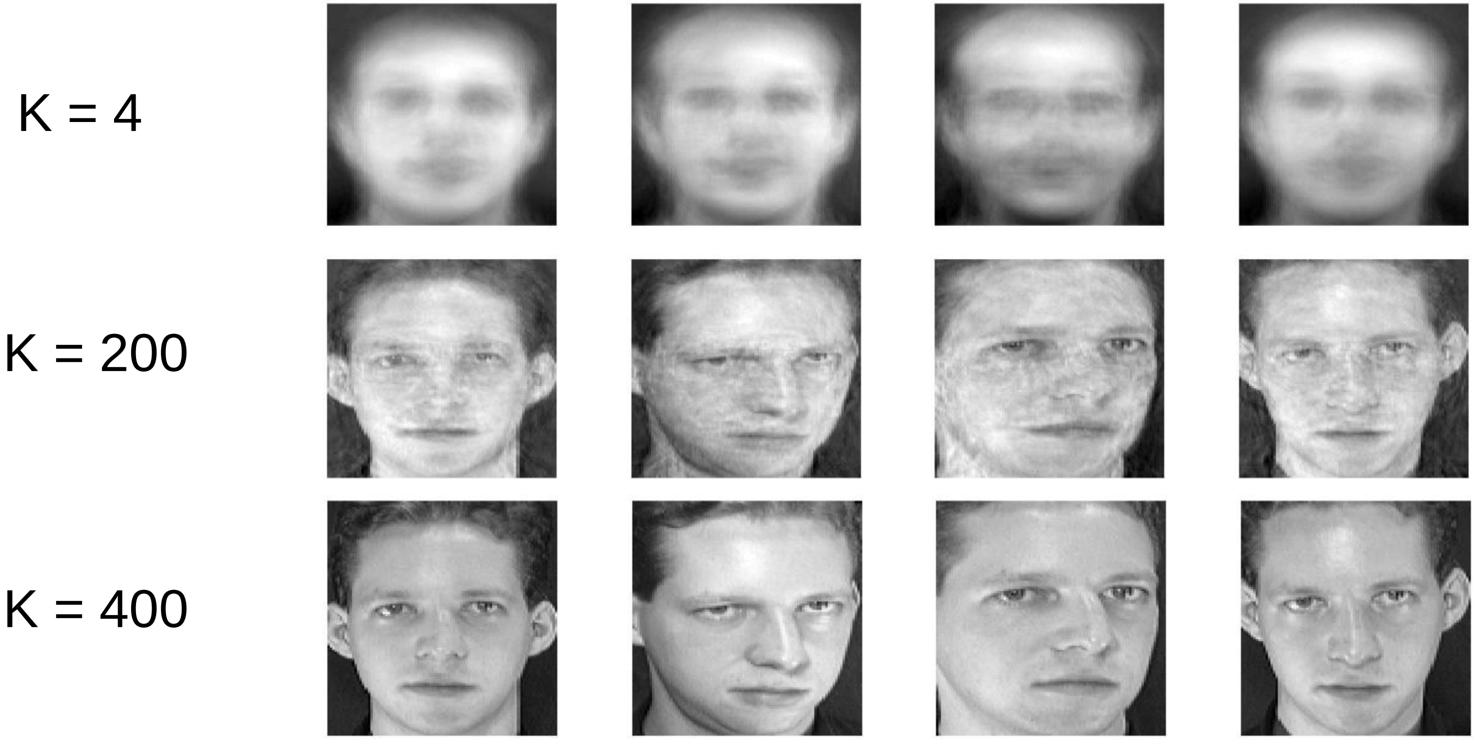


Why can we do this?



Eigenvalues (variance along eigenvectors)

Reconstruction and Errors



- Only selecting the top K eigenfaces → reduces the dimensionality.
- Fewer eigenfaces result in more information loss, and hence less discrimination between faces.

Eigenface algorithm

- Testing
  1. Take query image t
  2. Project into eigenface space and compute projection

$$t \rightarrow \left((t - m)\cdot f_1, (t - m)\cdot f_2, \ldots, (t - m)\cdot f_K\right) \equiv \left(w_1, w_2, \ldots, w_K\right)$$

  3. Compare projection w with all N training projections
     - Simple comparison metric: Euclidean
     - Simple decision: K-Nearest Neighbor

     (note: this “K” refers to the k-NN algorithm, is different from the previous K’s referring to the # of principal components)

Advantages

- Method is completely knowledge free:
  - Doesn’t know anything about faces, expressions, etc.
- Non-iterative (fast), globally optimal solution

Disadvantages

- Requires carefully controlled data:
  - All faces centered in frame
  - Same size
  - Some sensitivity to angle
- Alternative:
  - “Learn” one set of PCA vectors for each angle
  - Use the one with lowest error
- Method is completely knowledge free：
  - Doesn’t take into account the labels associated with the faces
  - Makes no effort to preserve class distinctions

Expressions and Emotions

- This technique can be used to detect expressions and emotions.
- The subspaces would therefore represent happiness, disgust, or other potential expressions.

Happiness subspace

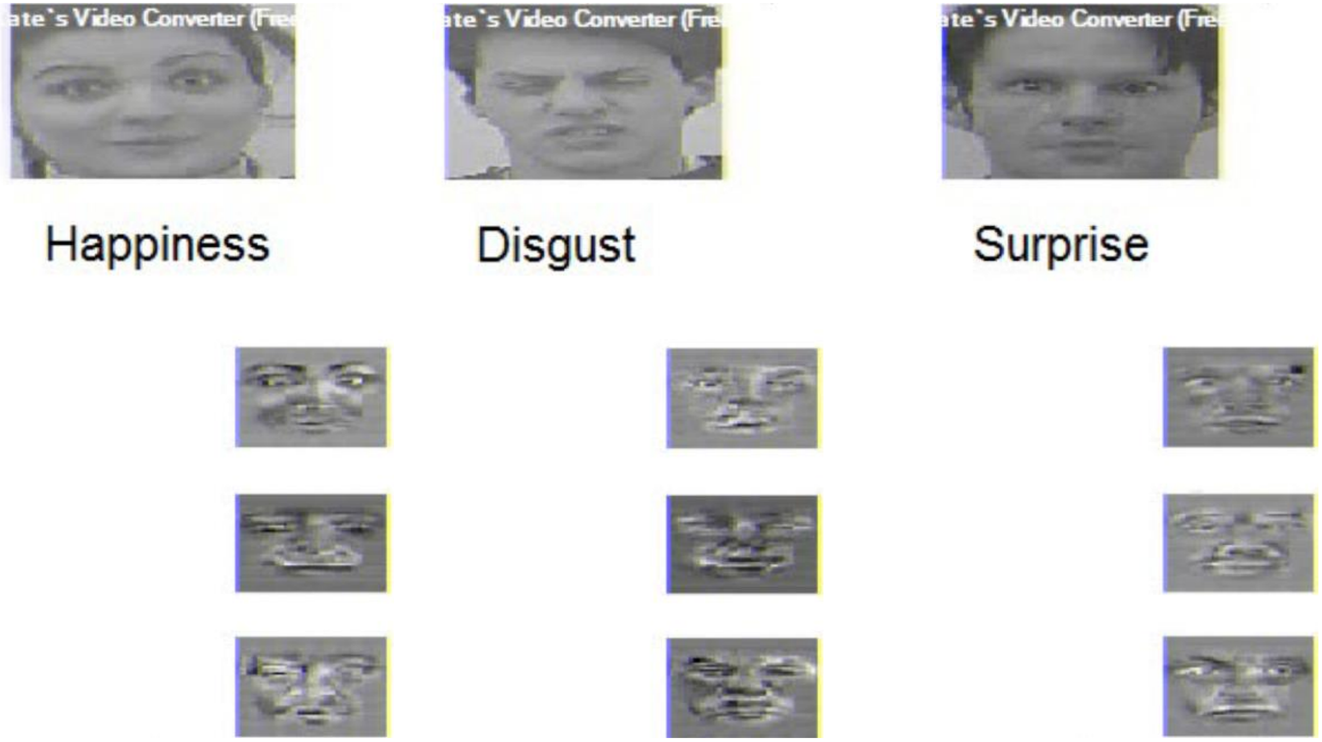
Expressions and Emotions

- This technique can be used to detect expressions and emotions.
- The subspaces would therefore represent happiness, disgust, or other potential expressions.

Disgust subspace

Expressions and Emotions

- This technique can be used to detect expressions and emotions.
- The subspaces would therefore represent happiness, disgust, or other potential expressions.


ate`s Video Converter (Fre
ate`s Video Converter (Fre
ate`s Video Converter (Fre
Happiness
Disgust
Surprise

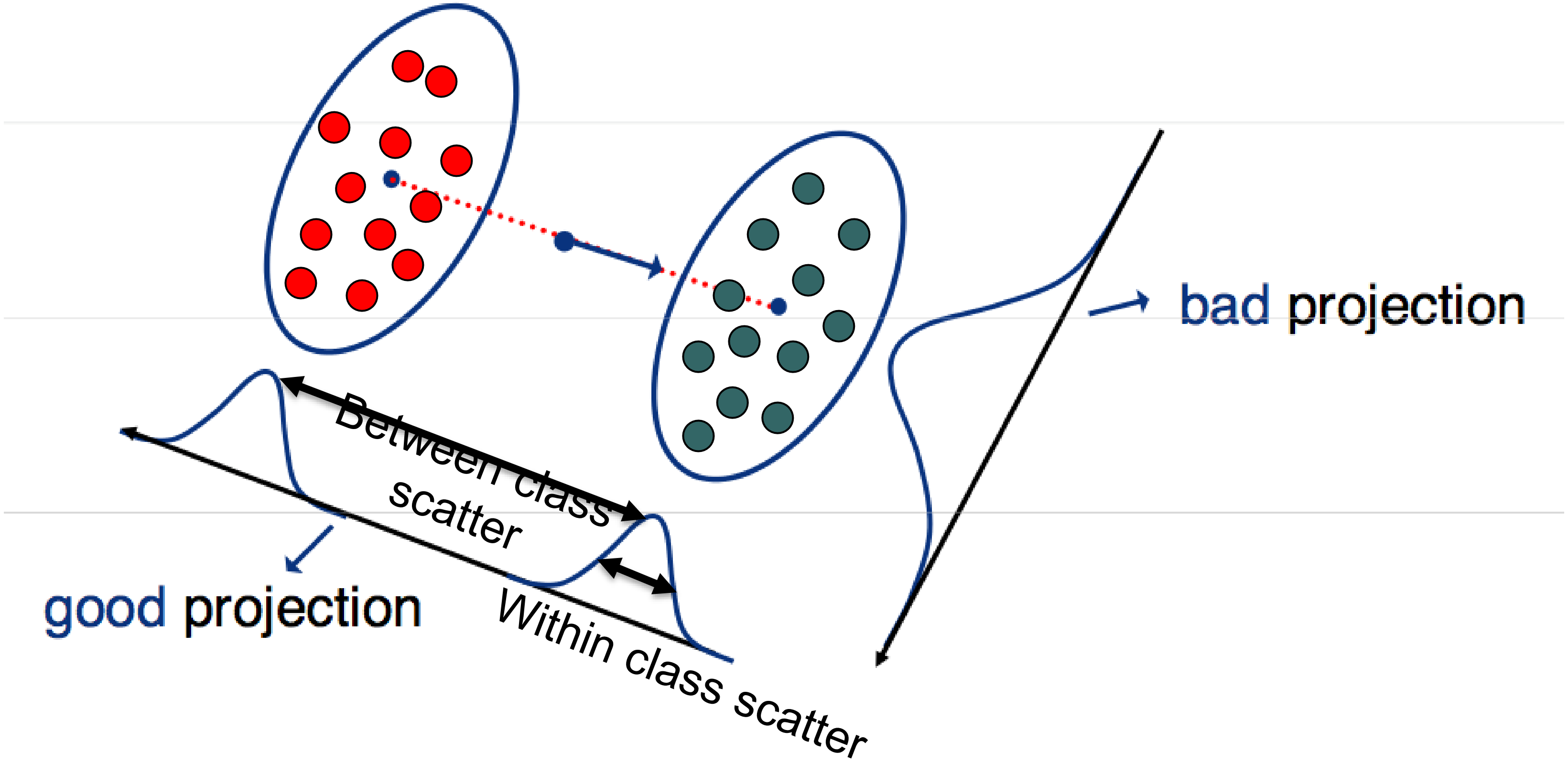
Facial Expression Recognition Movies

What we will learn today?

- Introduction to face recognition
- The Eigenfaces Algorithm
- Linear Discriminant Analysis (LDA)

Linear Discriminant Analysis (LDA)

- Goal: find the best separation between two classes



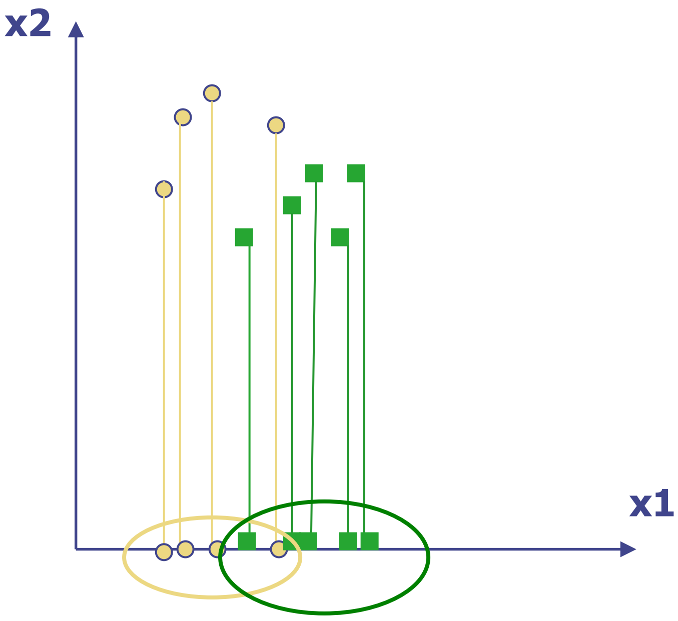
P. Belhumeur, J. Hespanha, and D. Kriegman. "Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection". *IEEE Transactions on pattern analysis and machine intelligence* **19** (7): 711. 1997.

Difference between PCA and LDA

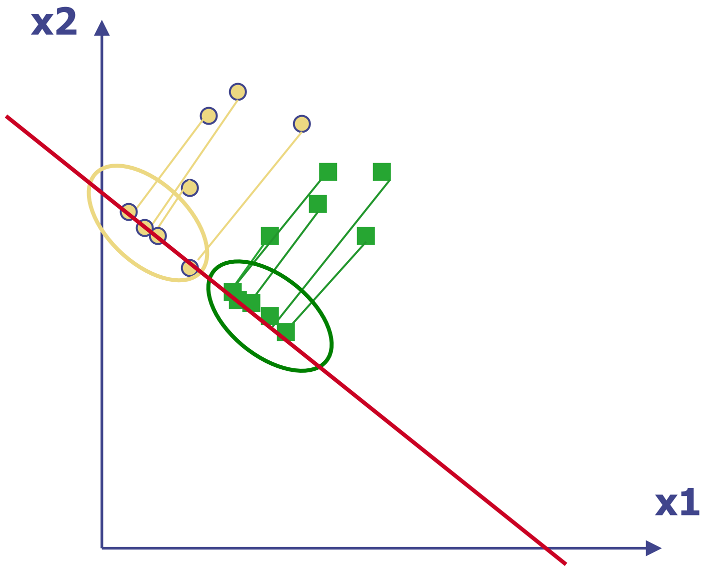
- PCA preserves maximum variance
- LDA preserves discrimination
  - Find projection that maximizes scatter between classes and minimizes scatter within classes

Illustration of the Projection

- Using two classes as example:

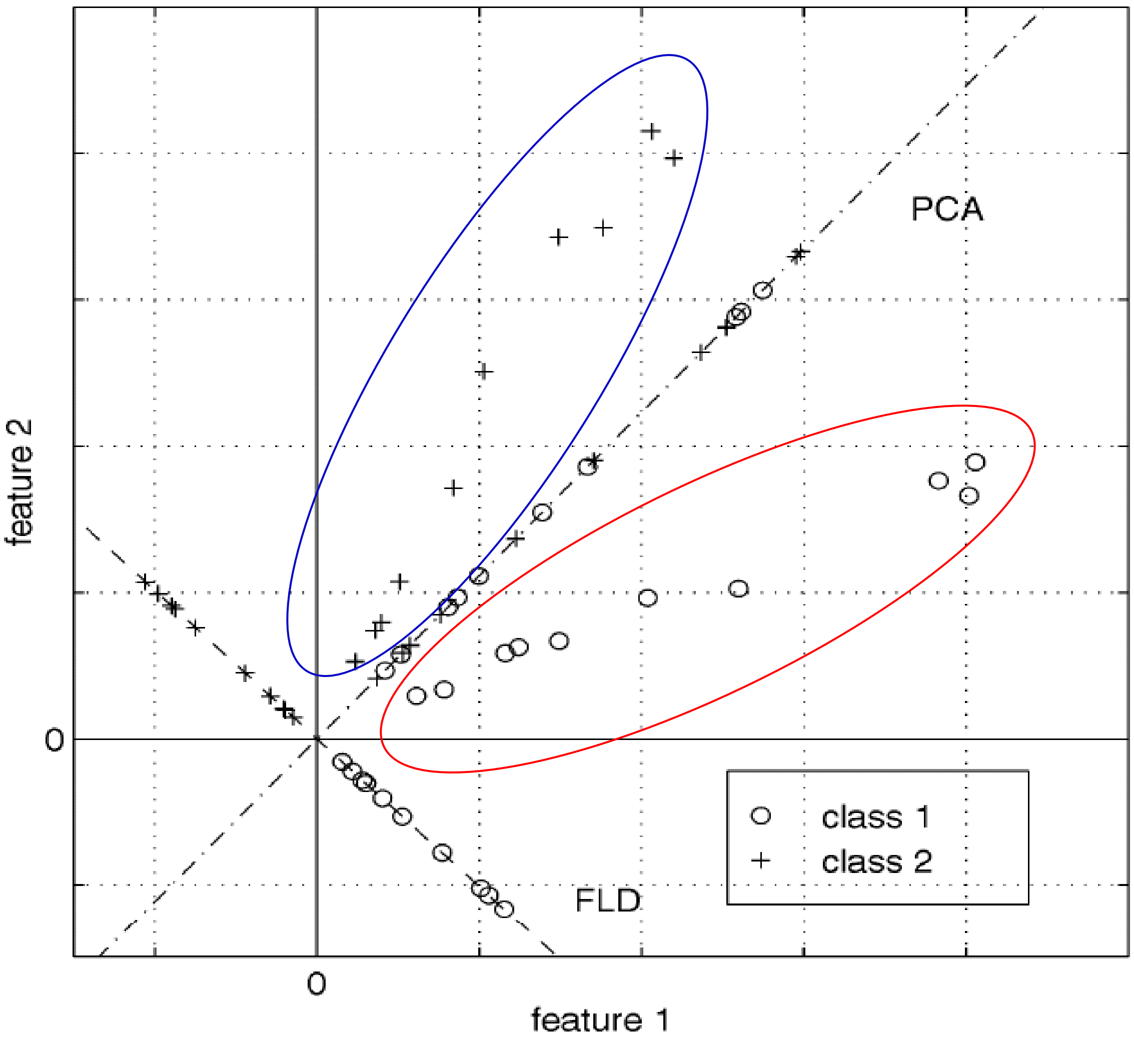


Poor Projection



Good

Basic intuition: PCA vs. LDA



LDA with 2 classes

- We want to learn a projection w such that the projection converts all the points from x to a new space (For this example, assume m == 1):

$$z = w^T x \qquad z \in \boldsymbol{R}^m \qquad x \in \boldsymbol{R}^n$$

- Let the **per class** means be:

$$E_{X|Y}[X|Y = i] = \mu_i$$

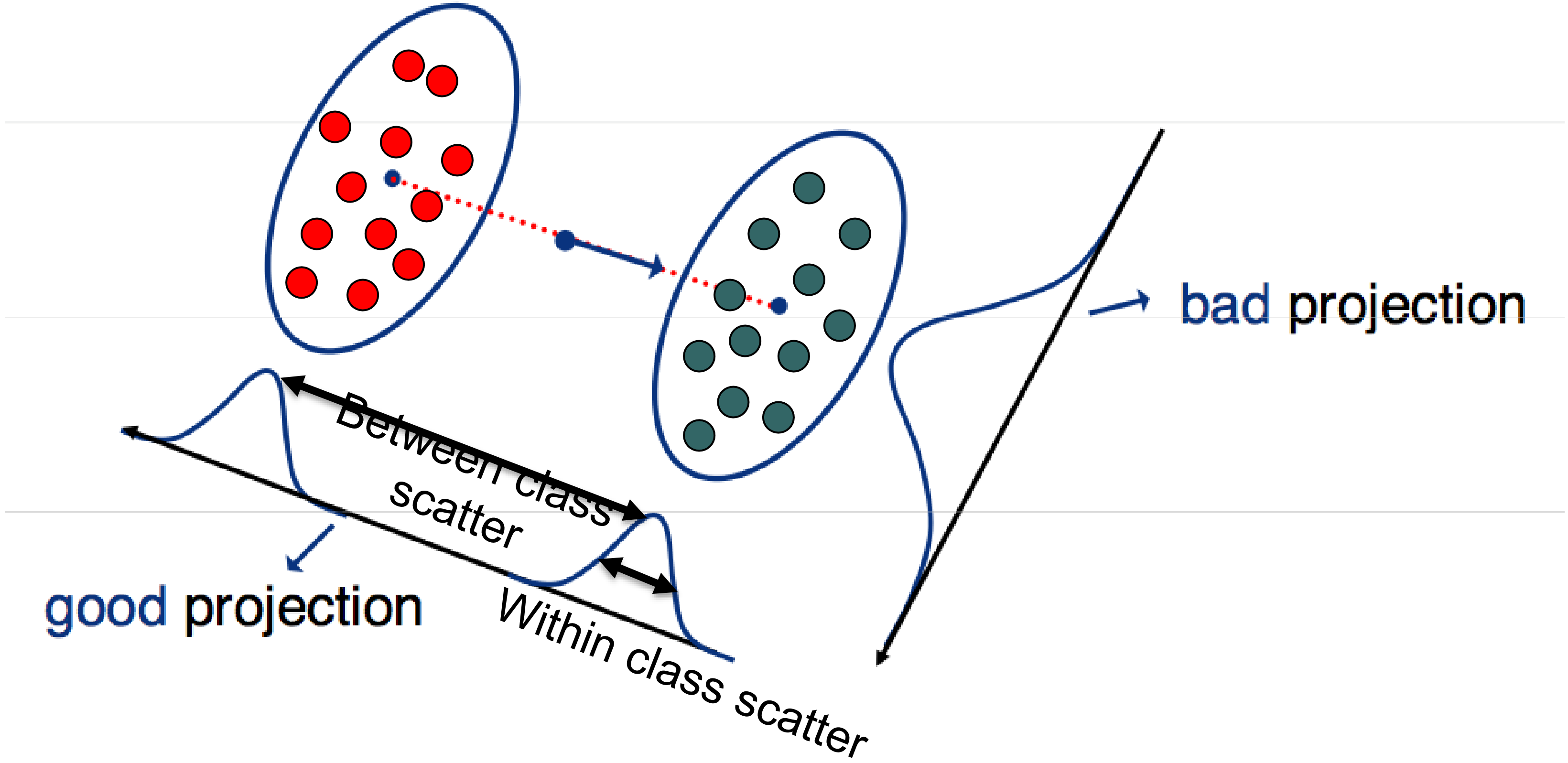
- And the **per class** covariance matrices be:

$$E_{X|Y}[(X - \mu_i)(X - \mu_i)^T \big| Y = i] = \sum_i$$

- We want a projection that maximizes:

$$J(w) = \frac{between\ class\ scatter}{within\ class\ scatter}$$

Linear Discriminant Analysis (LDA)



LDA with 2 classes

- The following objective function:

$$J(w) = \frac{between\ class\ scatter}{within\ class\ scatter}$$

- Can be written as

$$J(w) = \frac{(E_{Z|Y}[Z|Y = 1] - E_{Z|Y}[Z|Y = 0])^2}{var[Z|Y = 1] + var[Z|Y = 0]}$$

LDA with 2 classes

- We can write the between class scatter as:

$$(E_{Z|Y}[Z|Y=1]-E_{Z|Y}[Z|Y=0])^2 = (w^T[\mu_1-\mu_0])^2$$
$$= w^T[\mu_1-\mu_0][\mu_1-\mu_0]^T w$$

- Also, the within class scatter becomes:

$$var[Z|Y=i]$$
$$= E_{Z|Y}\left\{\left(z-E_{Z|Y}[Z|Y=i]\right)^2\middle|Y=i\right\}$$
$$= E_{Z|Y}\{(w^T[x-\mu_i])^2|Y=i\}$$
$$= E_{Z|Y}\{w^T[x-\mu_i][x-\mu_i]^T w|Y=i\}$$
$$= w^T\Sigma_i w$$

LDA with 2 classes

- We can plug in these scatter values to our objective function:

$$J(w) = \frac{w^T S_B w}{w^T S_W w}$$

Between class scatter

$$S_B = (\mu_1 - \mu_0)(\mu_1 - \mu_0)^T$$

$$S_W = (\Sigma_1 + \Sigma_0)$$

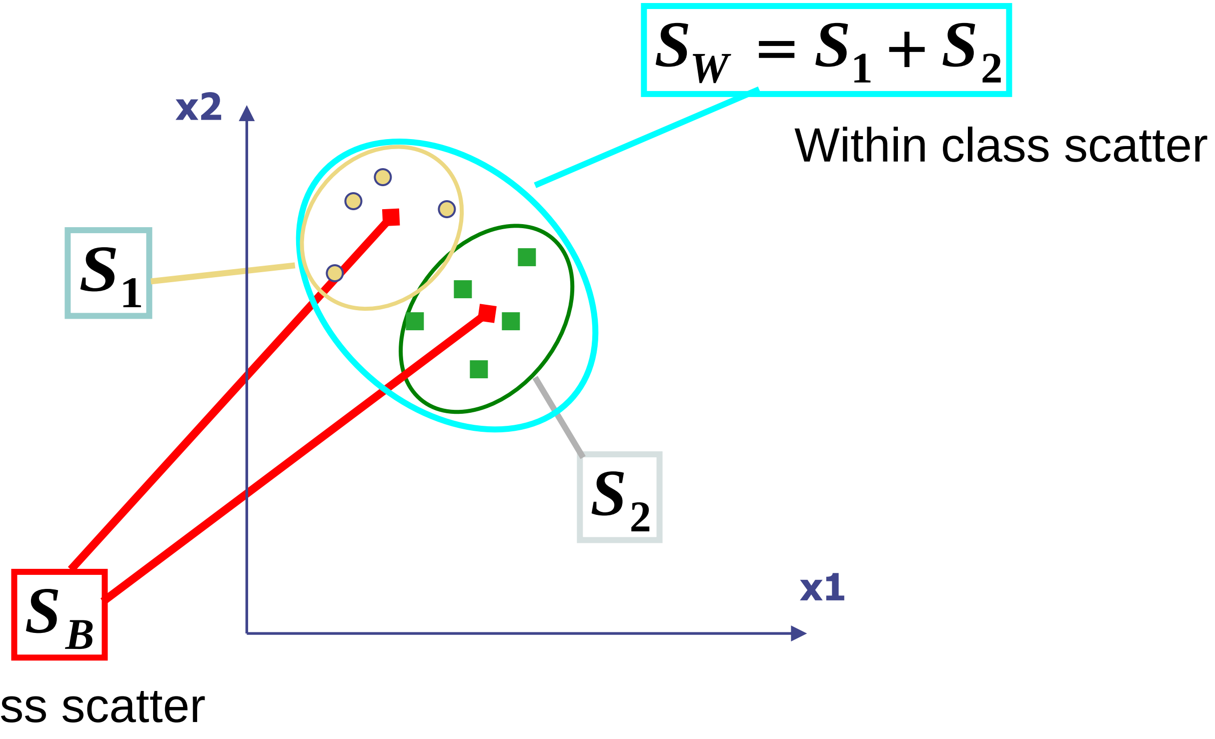
Within class scatter

- And our objective becomes:

$$J(w) = \frac{(E_{Z|Y}[Z|Y=1] - E_{Z|Y}[Z|Y=0])^2}{var[Z|Y=1] + var[Z|Y=0]}$$

$$= \frac{w^T(\mu_1 - \mu_0)(\mu_1 - \mu_0)^T w}{w^T(\Sigma_1 + \Sigma_0)w}$$

Visualization



Linear Discriminant Analysis (LDA)

- Maximizing the ratio

$$J(w) = \frac{w^T S_B w}{w^T S_W w}$$

- Is equivalent to maximizing the numerator while keeping the denominator constant, i.e.

$$\max_w w^T S_B w \quad subject\ to \quad w^T S_W w = K$$

- And can be accomplished using Lagrange multipliers, where we define the Lagrangian as

$$L = w^T S_B w - \lambda(w^T S_W w - K)$$

- And maximize with respect to both w and λ

Linear Discriminant Analysis (LDA)

- Setting the gradient of

$$L = w^T(S_B - \lambda S_W)w + \lambda K$$

With respect to w to zeros we get

$$\nabla_w L = 2(S_B - \lambda S_W)w = 0$$

or

$$S_B w = \lambda S_W w$$

- This is a generalized eigenvalue problem

- The solution is easy when $S_W{}^{-1} = (\Sigma_1 + \Sigma_0)^{-1}$

Linear Discriminant Analysis (LDA)

- In this case

$$S_W^{-1} S_B w = \lambda w$$

- And using the definition of S_B

$$S_W^{-1}(\mu_1 - \mu_0)(\mu_1 - \mu_0)^T w = \lambda w$$

- Noting that $(\mu_1 - \mu_0)^T w = \alpha$ is a scalar this can be written as

$$S_W^{-1}(\mu_1 - \mu_0) = \frac{\lambda}{\alpha} w$$

- and since we don’t care about the magnitude of w

$$w^* = S_W^{-1}(\mu_1 - \mu_0) = (\Sigma_1 + \Sigma_0)^{-1}(\mu_1 - \mu_0)$$

LDA with N variables and C classes

- Variables
  - N Sample images: $\{x_1, x_2, \dots, x_N\}$
  - C classes: $\{Y_1, Y_2, \dots, Y_C\}$
  - Average of each class: $\mu_i = \frac{1}{N_i} \sum_{x_k \in Y_i} x_k$
  - Average of all data: $\mu = \frac{1}{N} \sum_{k=1}^{N} x_k$

LDA with N variables and C classes

- Scatter Matrices
  - Scatter of class i: $$S_i = \sum_{x_k \in Y_i} (x_k - \mu_i)(x_k - \mu_i)^T$$
  - Within class scatter: $$S_w = \sum_{i=1}^{\mathrm{C}} S_i$$
  - Between class scatter: $$S_B = \sum_{i=1}^{C} N_i (\mu_i - \mu)(\mu_i - \mu)^T$$

Mathematical Formulation

- Recall that we want to learn a projection W such that the projection converts all the points from x to a new space z:

$$z = W^T x \qquad z \in \boldsymbol{R}^m \qquad x \in \boldsymbol{R}^n$$

- After projection:
  - Between class scatter $\tilde{S}_B = W^T S_B W$
  - Within class scatter $\tilde{S}_W = W^T S_W W$
- So, the objective becomes:

$$\mathrm{W}_{opt} = \arg\max_W \frac{\left|\tilde{S}_B\right|}{\left|\tilde{S}_W\right|} = \arg\max_W \frac{\left|W^T S_B W\right|}{\left|W^T S_W W\right|}$$

Mathematical Formulation

- Solution: Generalized Eigenvectors

$$S_B w_i = \lambda_i S_W w_i \qquad\qquad i = 1, \ldots, m$$

- Rank of w_{opt} is limited
  - Rank(S_B) ≤ C-1
  - Rank(S_W) ≤ N-C

PCA vs. LDA

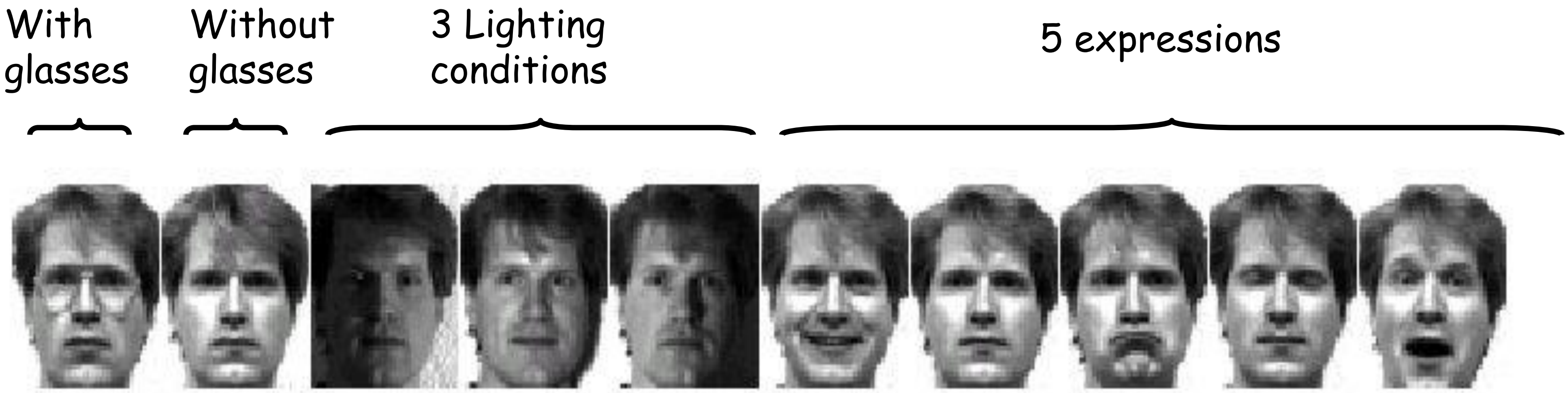
- Eigenfaces exploit the **max scatter** of the training images in face space

- Fisherfaces attempt to **maximize the between class** scatter, while **minimizing the within class scatter**.

Results: Eigenface vs. Fisherface

- Dataset: 160 images of 16 people
- Train: 159 images
- Test: 1 image

- Variation in Facial Expression, Eye Wear, and Lighting



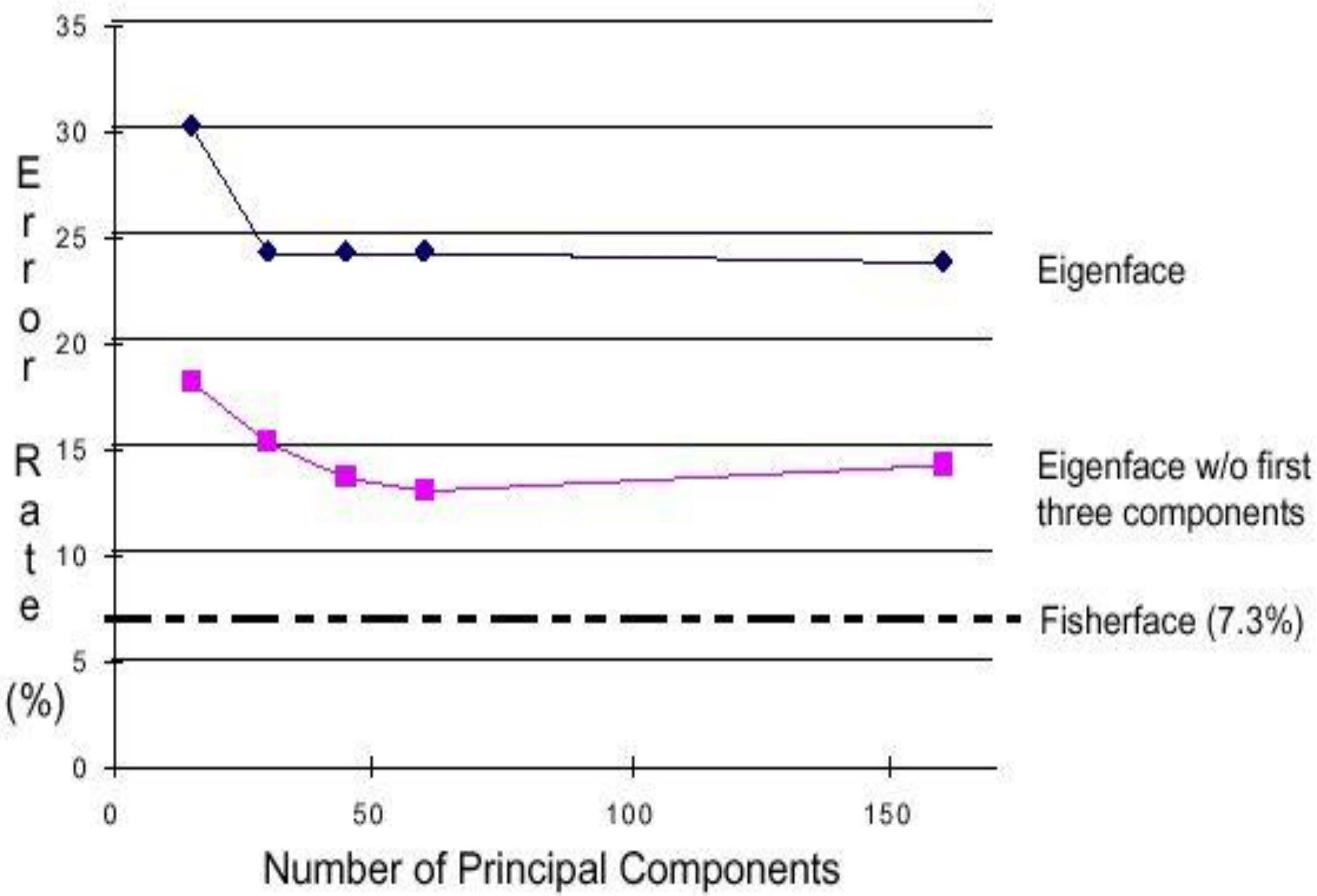
With glasses

Without glasses

3 Lighting conditions

5 expressions

Results: Eigenface vs. Fisherface



What we have learned today?

- Introduction to face recognition
- The Eigenfaces Algorithm
- Linear Discriminant Analysis (LDA)