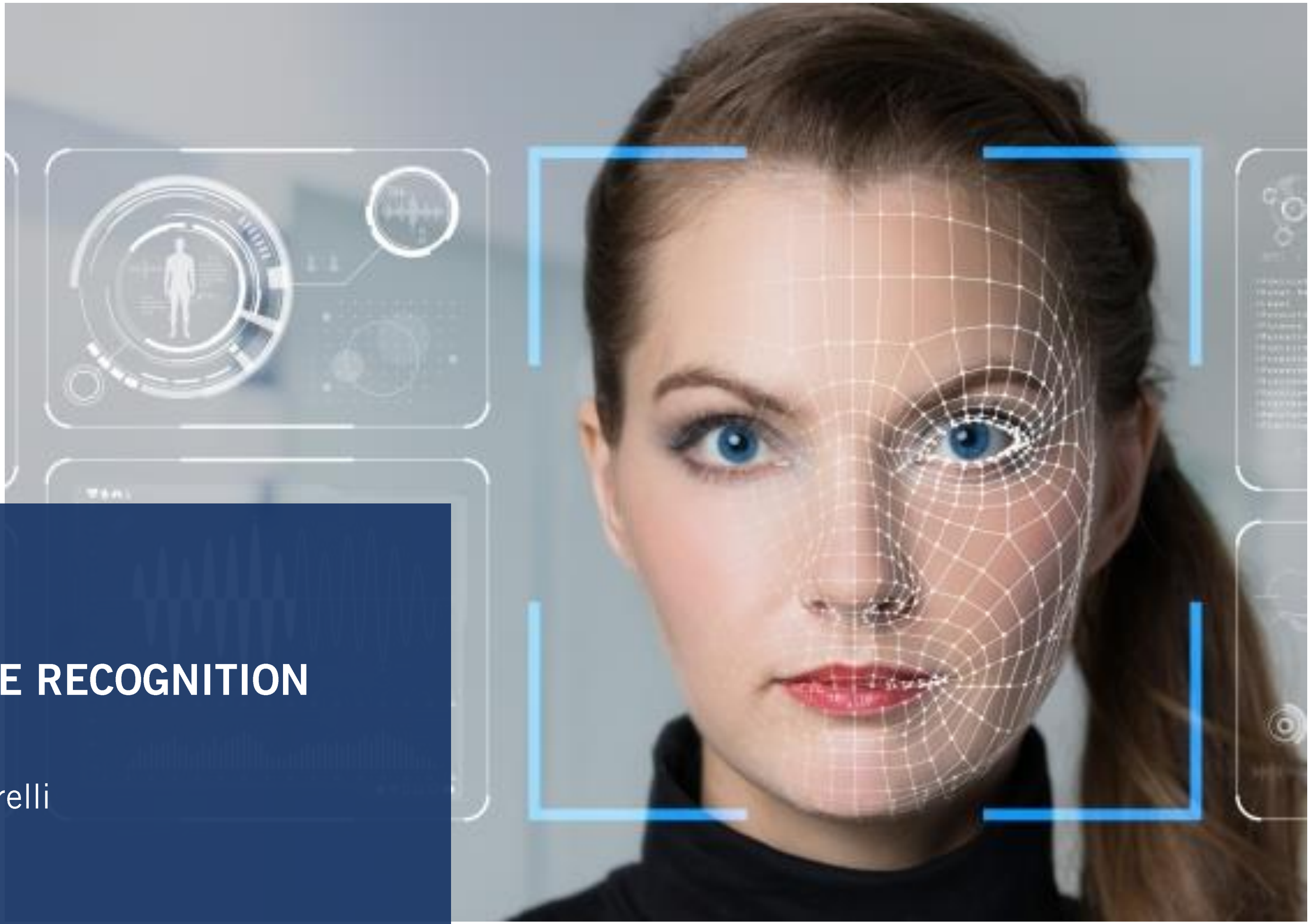


COMPUTER VISION
LECTURE 23 – FACE RECOGNITION

Prof. Dr. Francesco Maurelli

2019-11-19

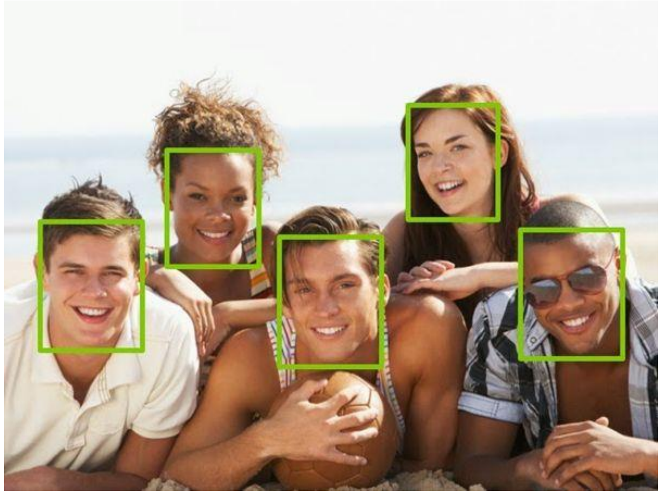
What we will learn today

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

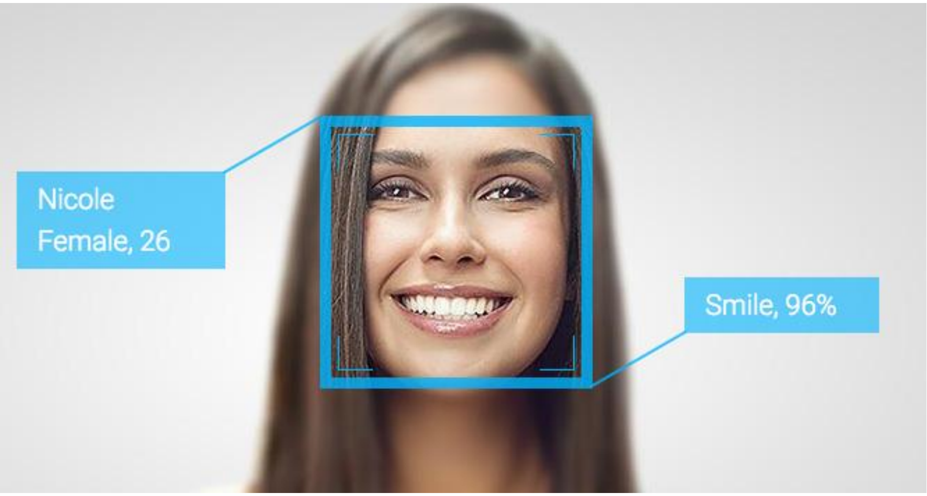
Turk and Pentland, Eigenfaces for Recognition, *Journal of Cognitive Neuroscience* **3** (1): 71–86.

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.

Detection versus Recognition



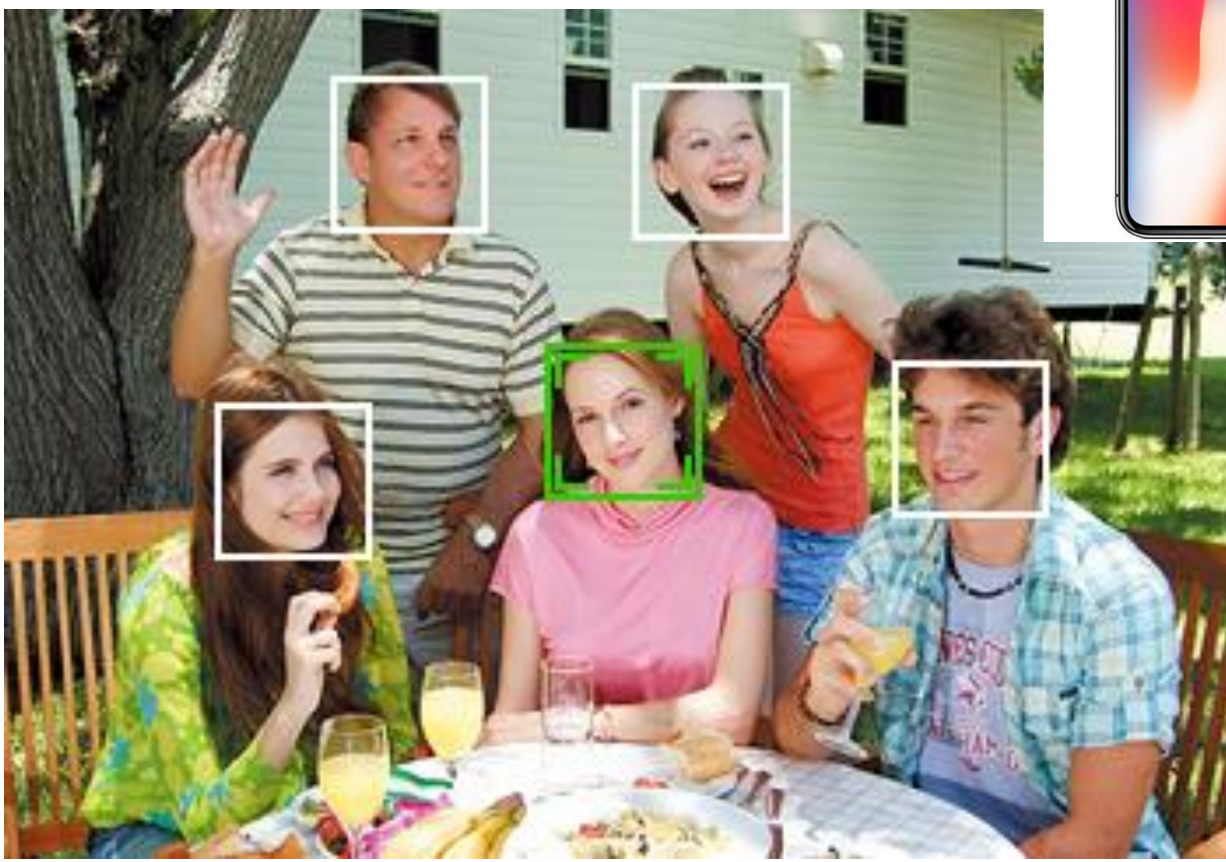
Detection finds the faces in images



Recognition recognizes WHO the person is

Face Recognition

- Digital photography



Face Recognition

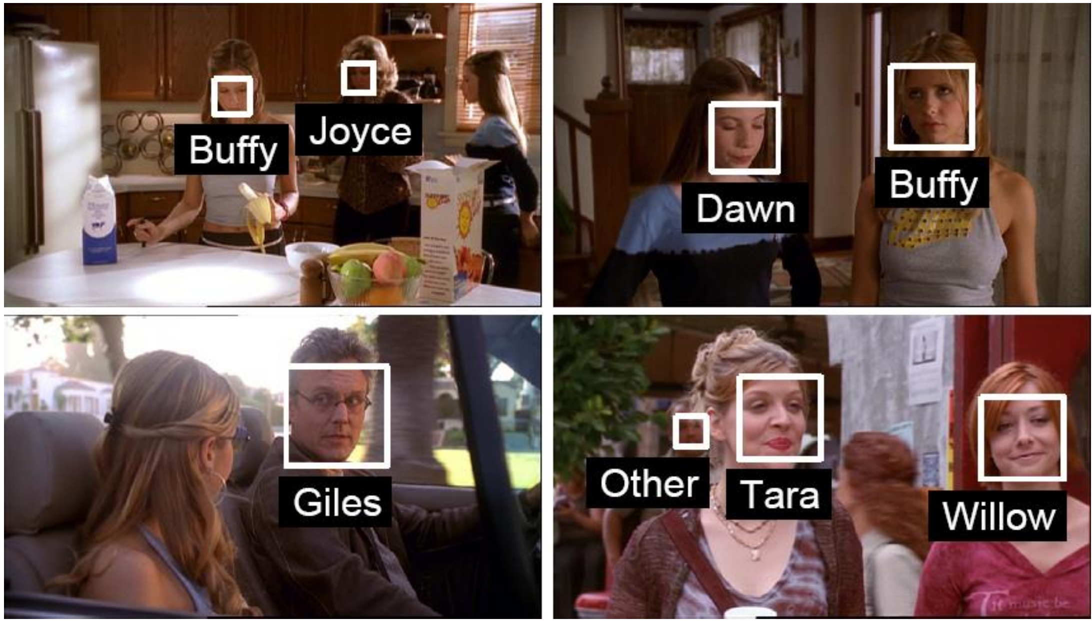
- Digital photography
- Surveillance

Face Recognition

- Digital photography
- Surveillance
- Album organization

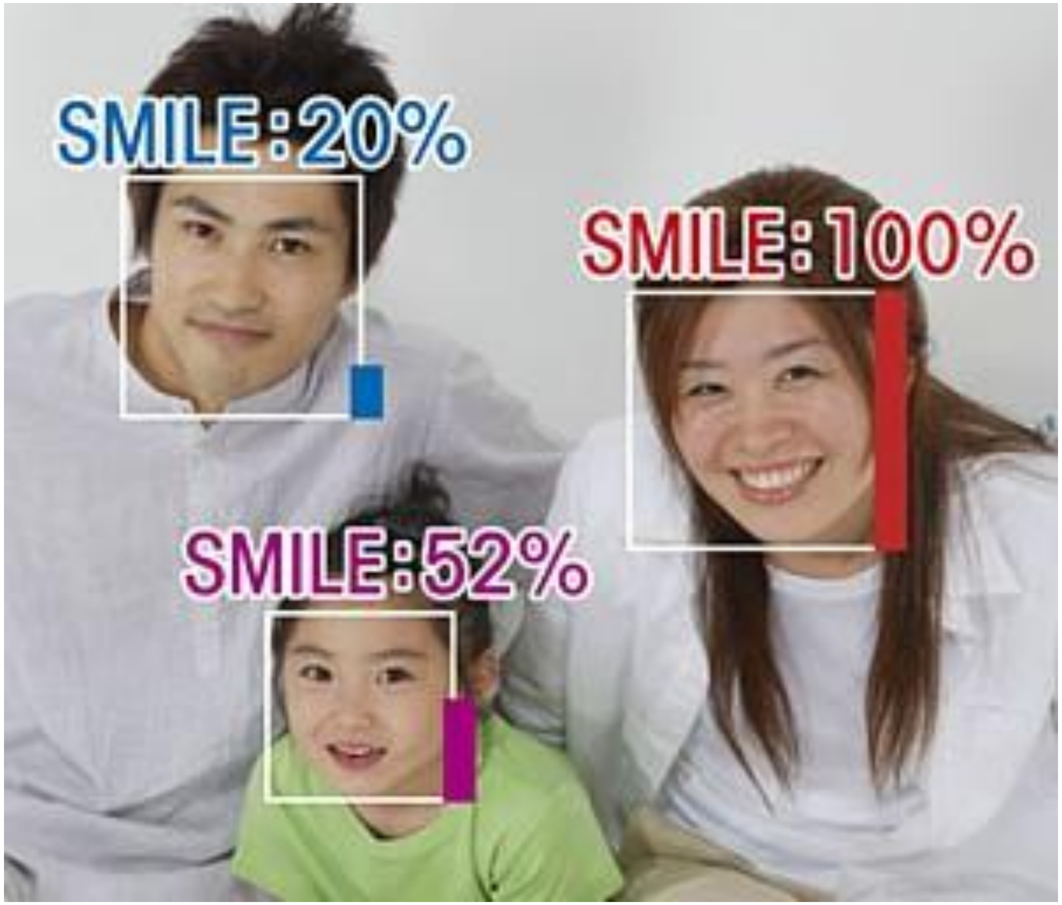
Face Recognition

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



Face Recognition

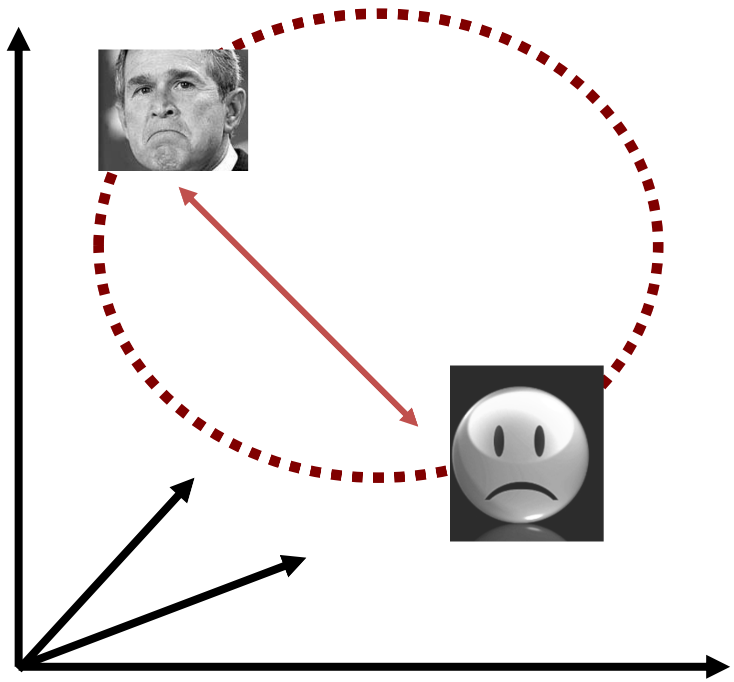
- Digital photography
- Surveillance
- Album organization
- Person tracking/id.
- Emotions and expressions



Face Recognition

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

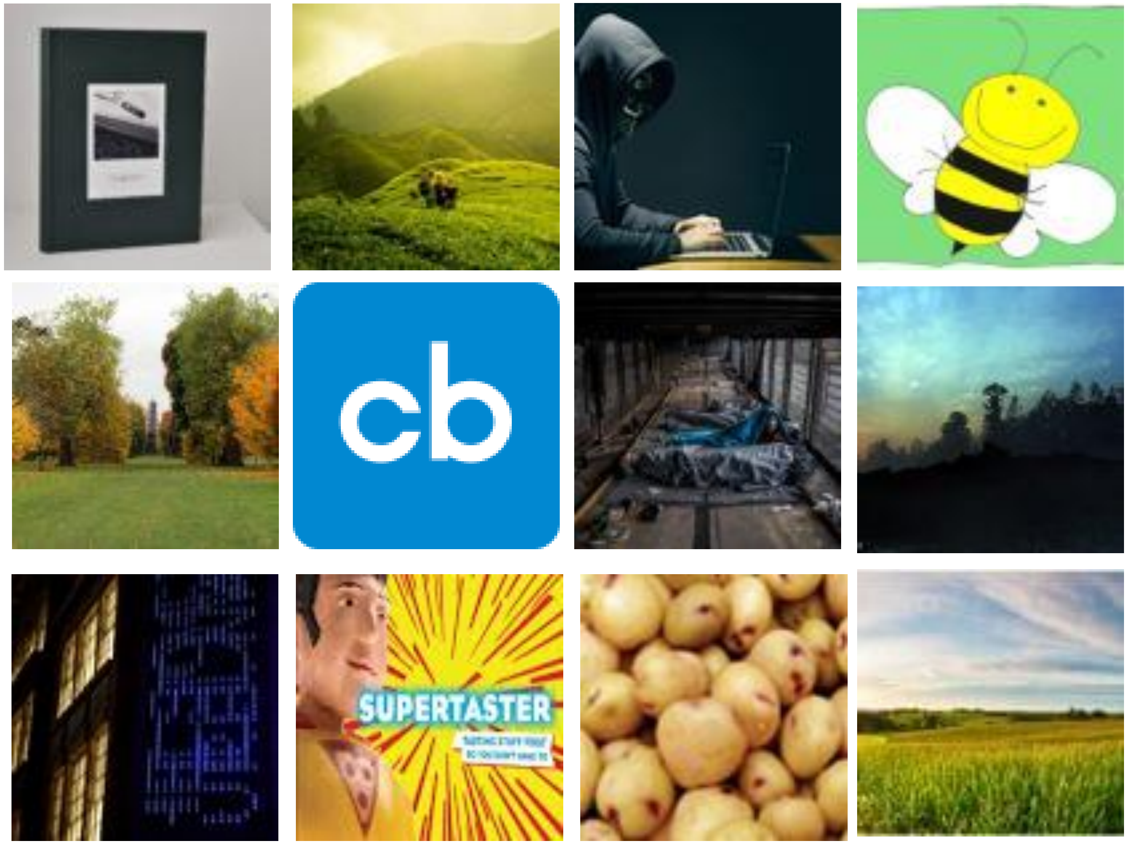
The Space of Faces



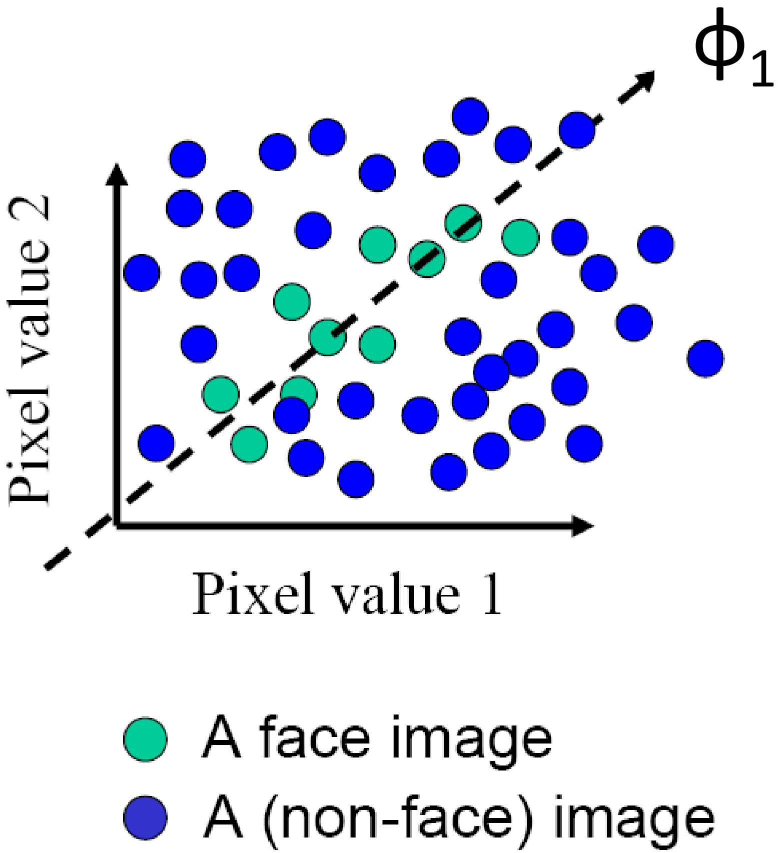
- An image is a point in a high dimensional space
  - If represented in grayscale intensity, an N x M image is a point in R^{NM}
  - E.g. 100x100 image = 10,000 dim

Slide credit: Chuck Dyer, Steve Seitz, Nishino

100x100 images can contain many things other than faces!



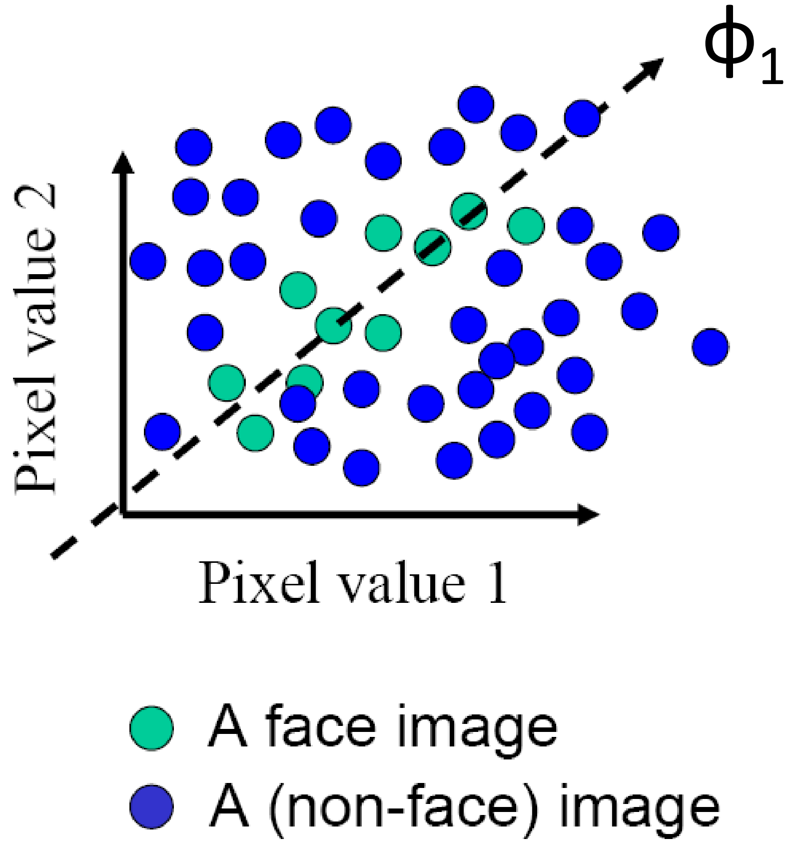
The Space of Faces

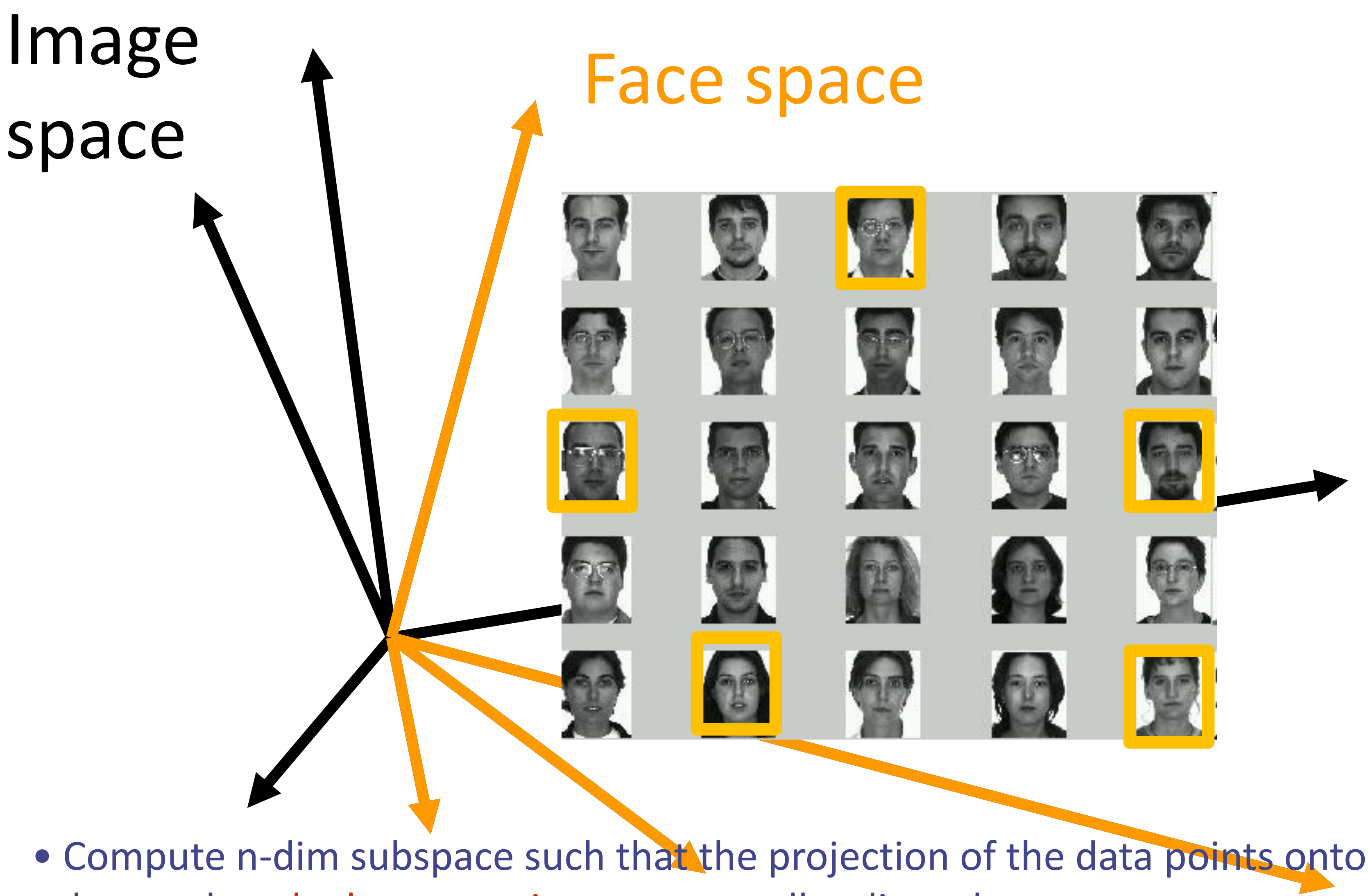


- An image is a point in a high dimensional space
  - If represented in grayscale intensity, an N x M image is a point in R^{NM}
  - E.g. 100x100 image = 10,000 dim
- However, relatively few high dimensional vectors correspond to valid face images
- We want to effectively model the subspace of face images

Slide credit: Chuck Dyer, Steve Seitz, Nishino

Where have we seen something like this before?





- 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

Key Idea

- So, compress them to a low-dimensional subspace that captures key appearance characteristics of the visual DOFs.

- USE PCA for estimating the sub-space (dimensionality reduction)

- Compare two faces by projecting the images into the subspace and measuring the EUCLIDEAN distance between them.

What we will learn today

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

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

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.

Eigenfaces: key idea

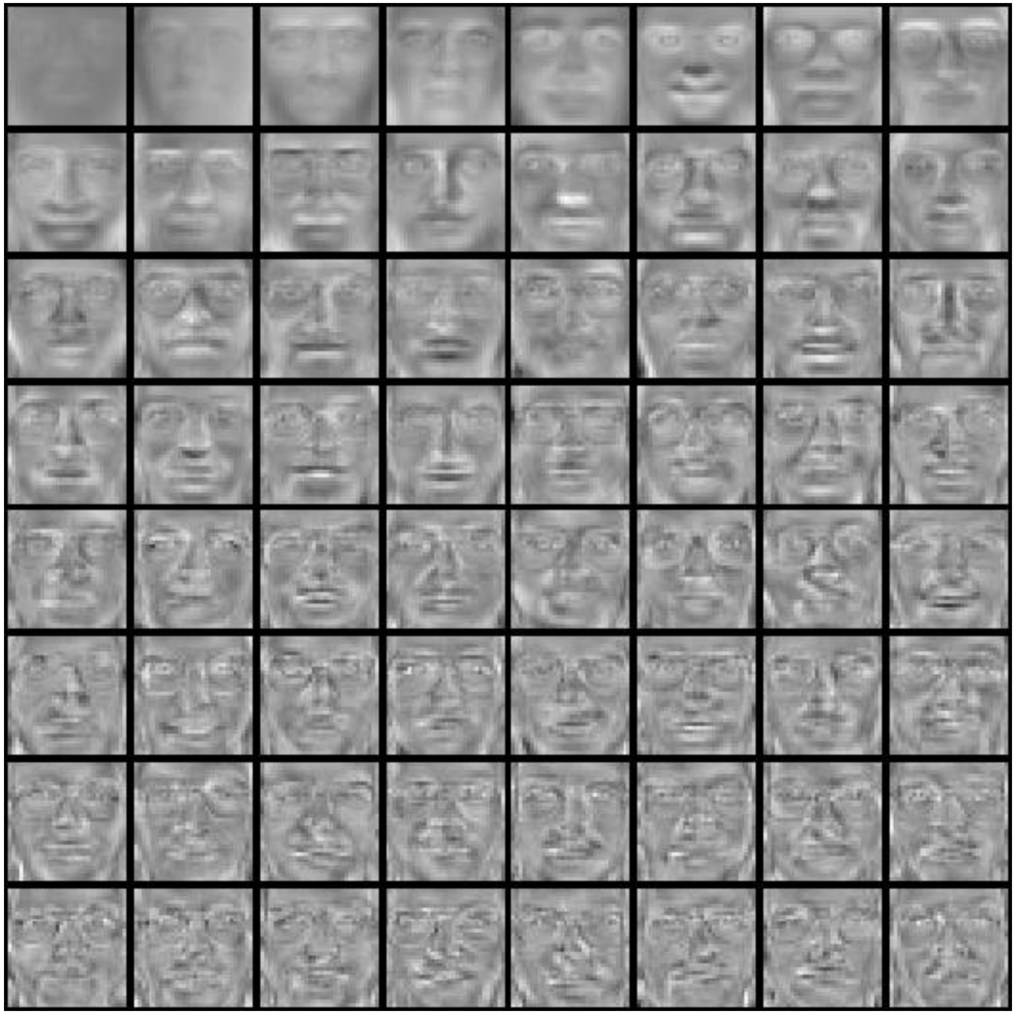
- Assume that most face images lie on a low-dimensional subspace determined by the first k (k<<d) directions of maximum variance
- Use PCA to determine the vectors or "eigenfaces" that span that subspace
- Represent all face images in the dataset as linear combinations of eigenfaces

M. Turk and A. Pentland, [Face Recognition using Eigenfaces](), CVPR 1991

Training images: $\mathbf{x}_1,\ldots,\mathbf{x}_N$

Top eigenvectors: $\phi_1,\ldots,\phi_k$

Mean: μ



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 $\mu = \frac{1}{N} \sum x_i$

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

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

Eigenface algorithm

4. Compute the covariance matrix

$$\boxed{\Sigma = \frac{1}{n}\begin{bmatrix} | & & | \\ x_1^c & \dots & x_n^c \\ | & & | \end{bmatrix}\begin{bmatrix} - & x_1^c & - \\ & \vdots & \\ - & x_n^c & - \end{bmatrix} = \frac{1}{n} X_c X_c^T}$$

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

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

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

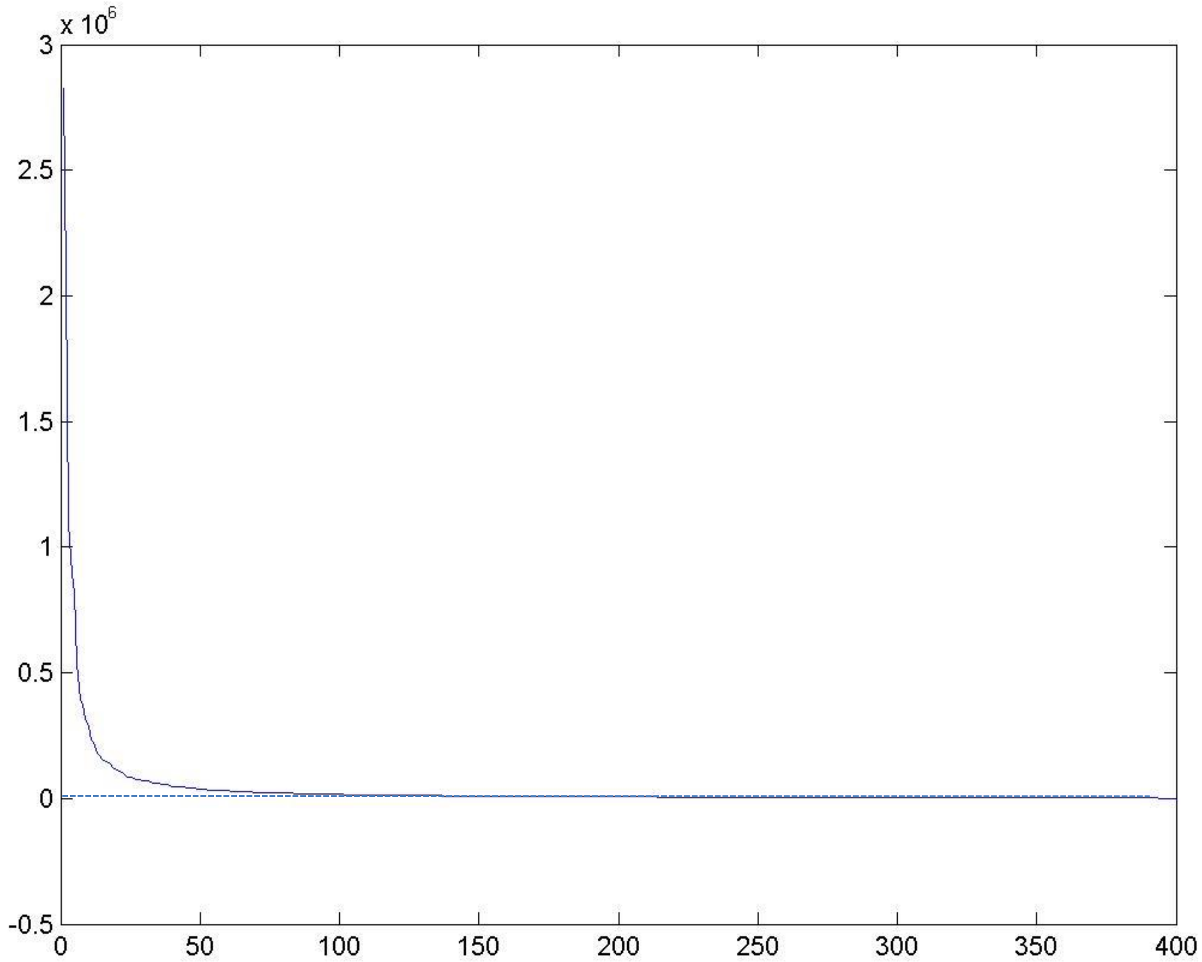
6. Compute each training image x_i ‘s projections as

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

7. Visualize the reconstructed training face x_i

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

Eigenvalues (variance along eigenvectors)



Reconstruction and Errors

K = 4

K = 200

K = 400

- 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)

Shortcomings

- 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
  - (sometimes this is good!)
  - Doesn’t know that faces are wrapped around 3D objects (heads)

Summary for Eigenface

Pros

- Non-iterative, globally optimal solution

Limitations

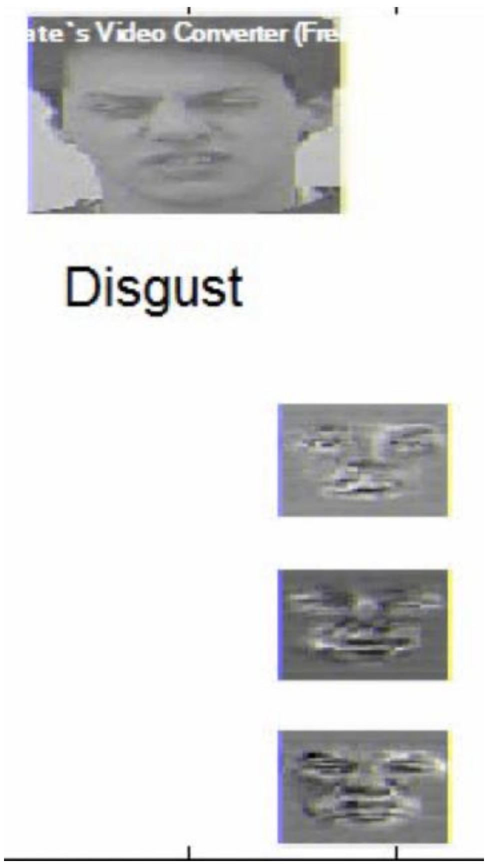
- PCA projection is **optimal for reconstruction** from a low dimensional basis, but **may NOT be optimal for discrimination…**

Besides face recognitions,
we can also do
Facial expression recognition

Happiness subspace (method A)

Disgust subspace (method A)

Facial Expression Recognition Movies (method A)



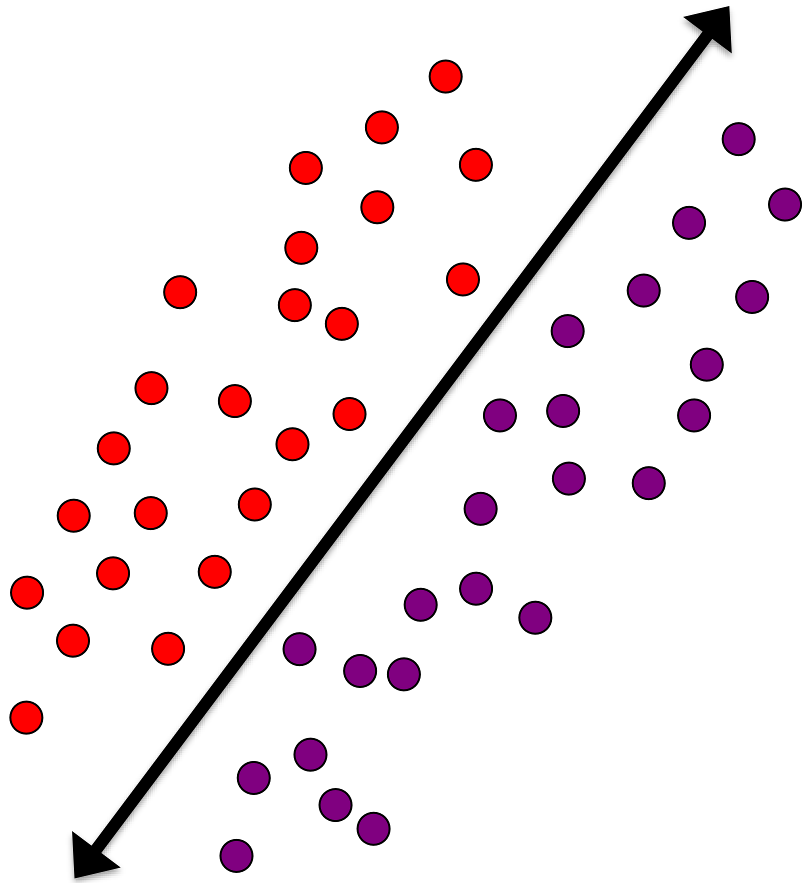
What we will learn today

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

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

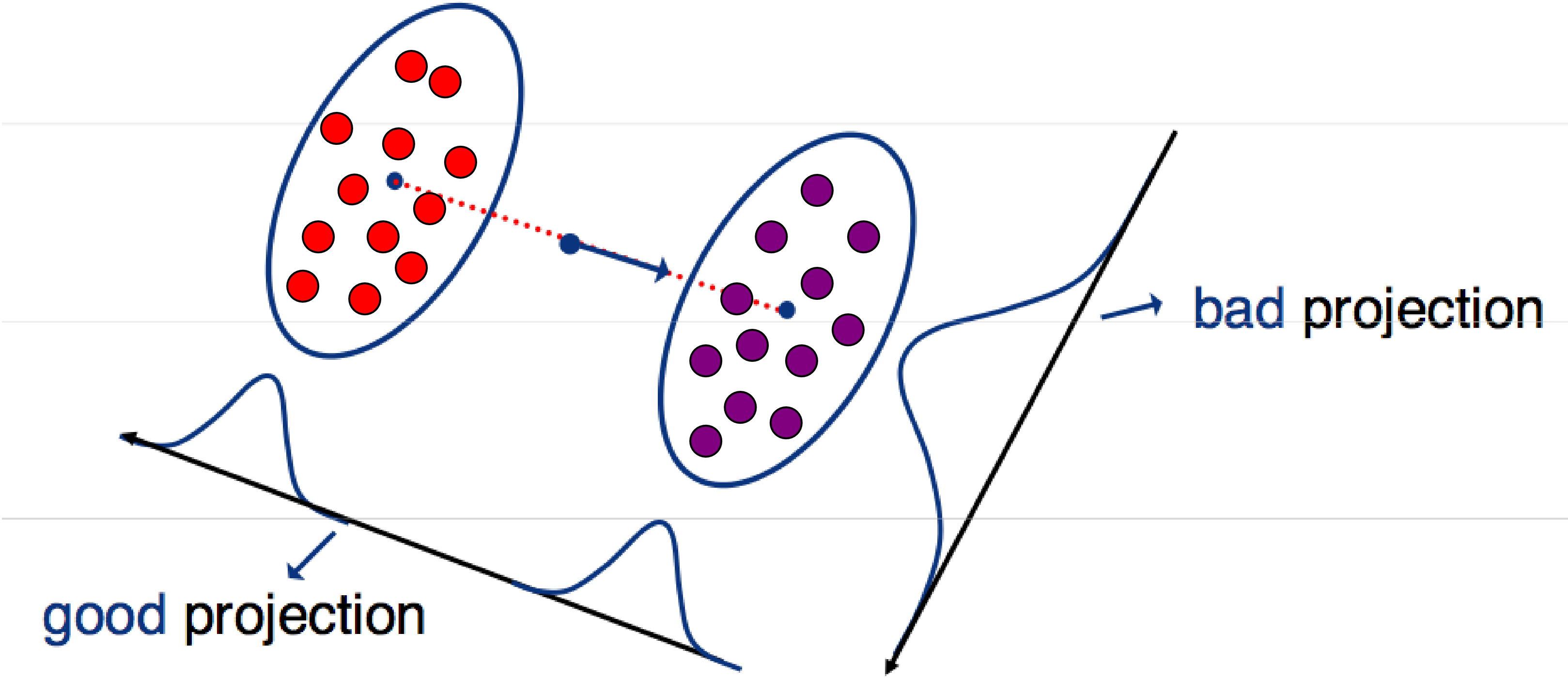
P. Belhumeur, J. Hespanha, and D. Kriegman. "Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection". *IEEE Transactions on pattern analysis and machine intelligence* **19** (7): 711. 1997.

Which direction is the first principle component?



Fischer's Linear Discriminant Analysis

- Goal: find the best separation between two classes



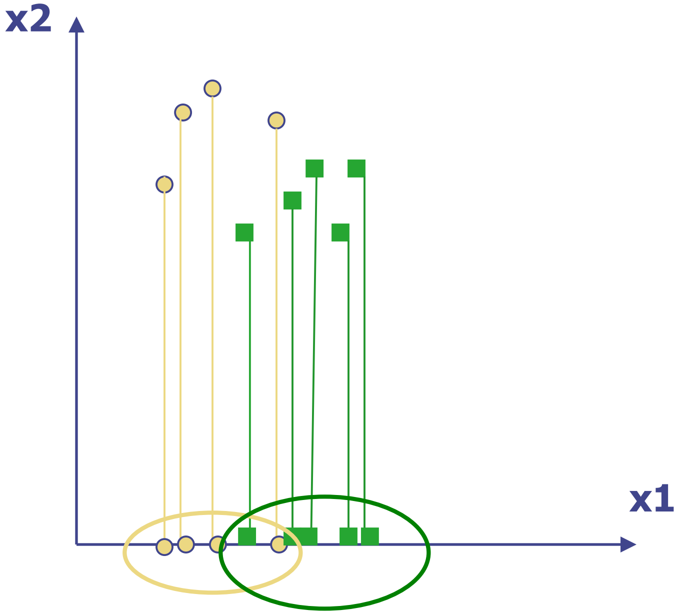
Slide inspired by N. Vasconcelos

Difference between PCA and LDA

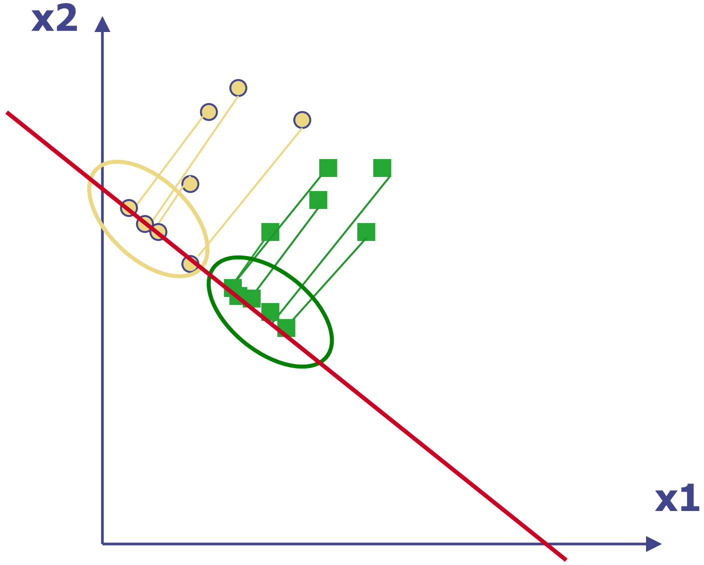
- PCA preserves maximum variance
- LDA preserves discrimination
  - Find projection that maximizes scatter between classes and minimizes scatter within classes

Illustration of the Projection

- Using two classes as example:



Poor Projection



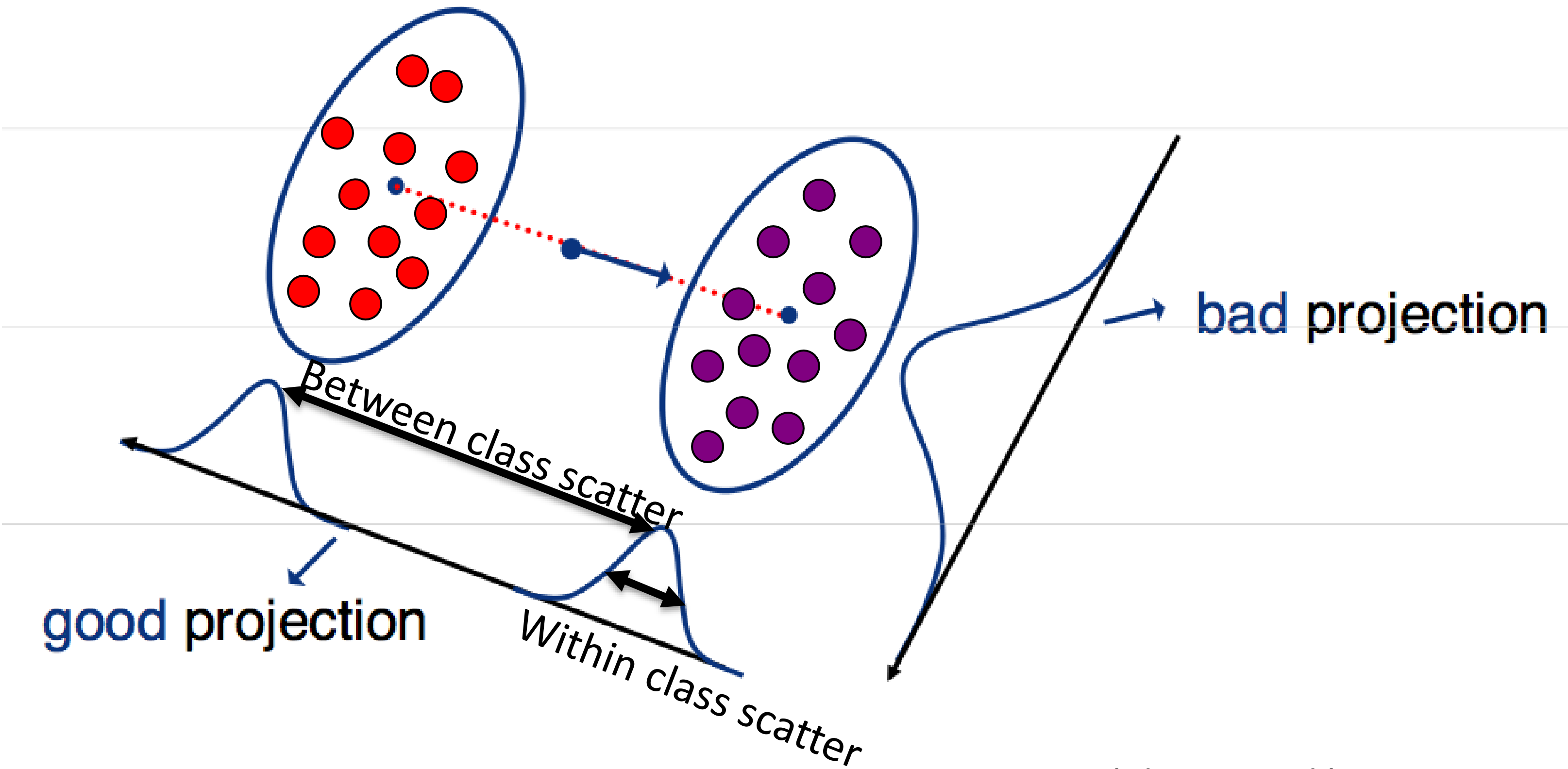
Good

LDA

We want a projection that maximizes:

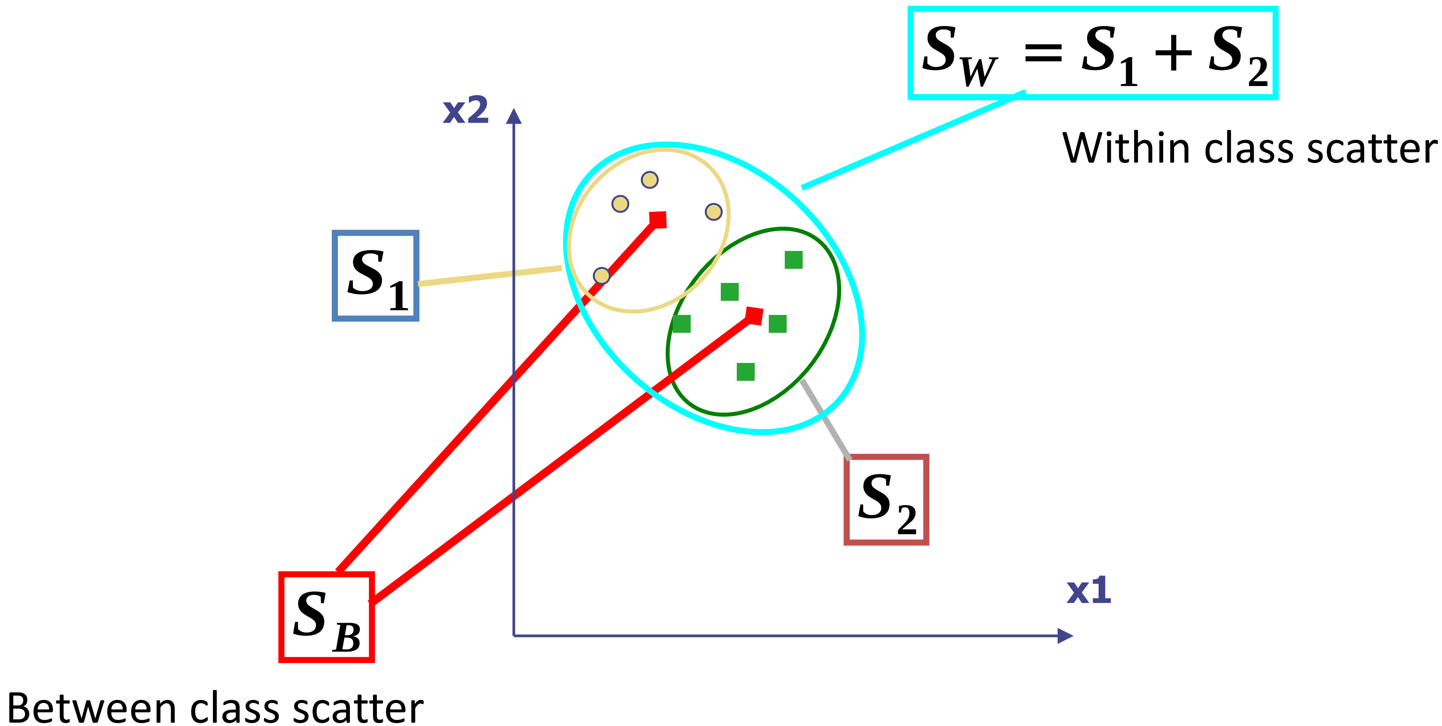
$$J(w) = \max \frac{between\ \ class\ \ scatter}{within\ \ class\ \ scatter}$$

Fischer's Linear Discriminant Analysis



Slide inspired by N. Vasconcelos

Visualization



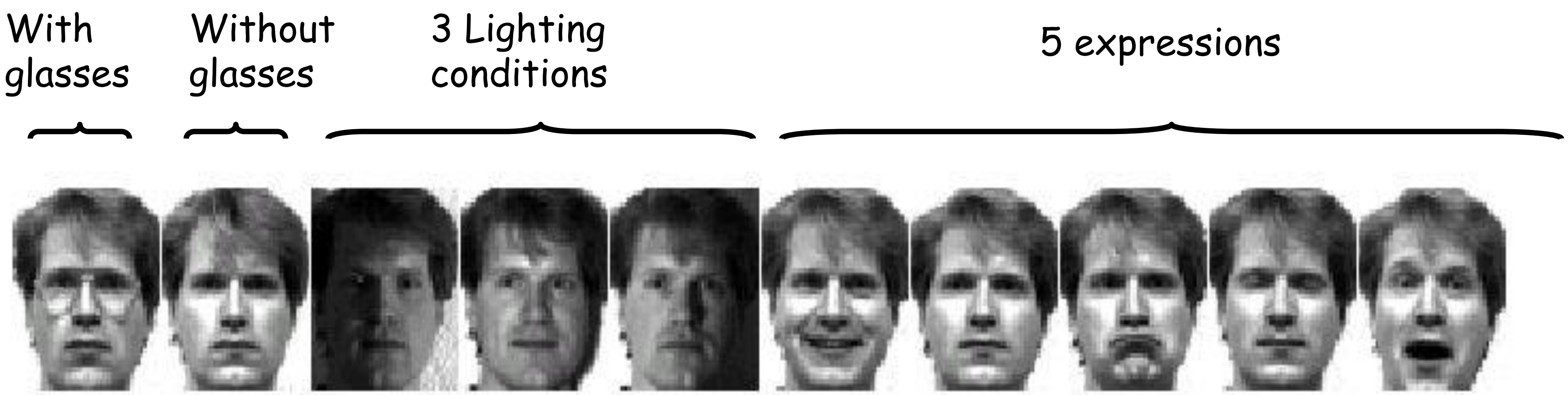
PCA vs. LDA

- Eigenfaces exploit the max scatter of the training images in face space
- Fisherfaces attempt to maximise the **between class scatter**, while minimising the **within class scatter**.

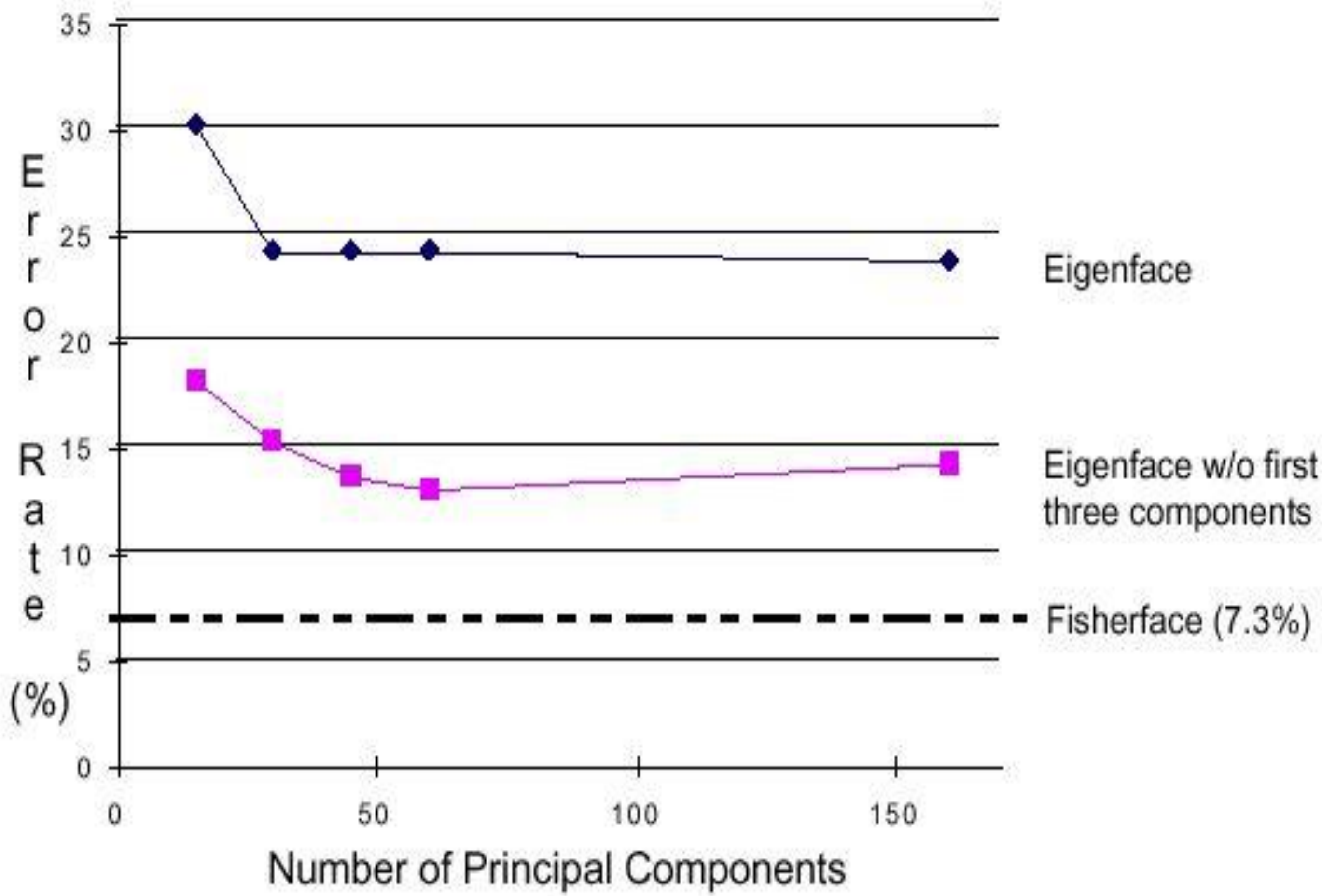
Results: Eigenface vs. Fisherface

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

- Variation in Facial Expression, Eyewear, and Lighting



Eigenface vs. Fisherface



What we have learned today

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

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

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.