Lecture: Face Recognition

Juan Carlos Niebles and Ranjay Krishna
Stanford Vision and Learning Lab

CS 131 Roadmap

Pixels	Segments	Images	Videos	Web
Convolutions Edges Descriptors	Resizing Segmentation Clustering	Recognition Detection Machine learning	Motion Tracking	Neural networks Convolutional neural networks

Let’s recap

- A simple object recognition pipeline with kNN
- PCA

Object recognition: a classification framework

- Apply a prediction function to a feature representation of the image to get the desired output:

f() = “apple”

f() = “tomato”

f() = “cow”

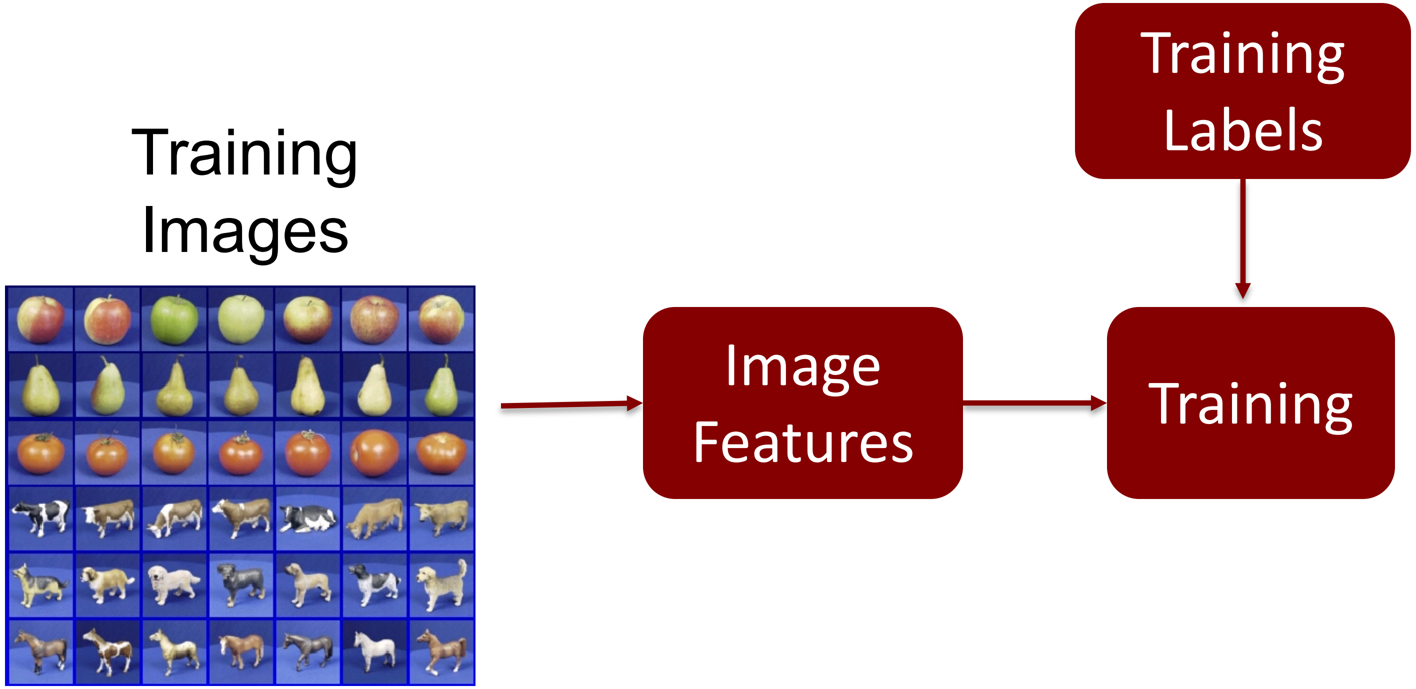
Dataset: ETH-80, by B. Leibe   Slide credit: L. Lazebnik

A simple pipeline - Training

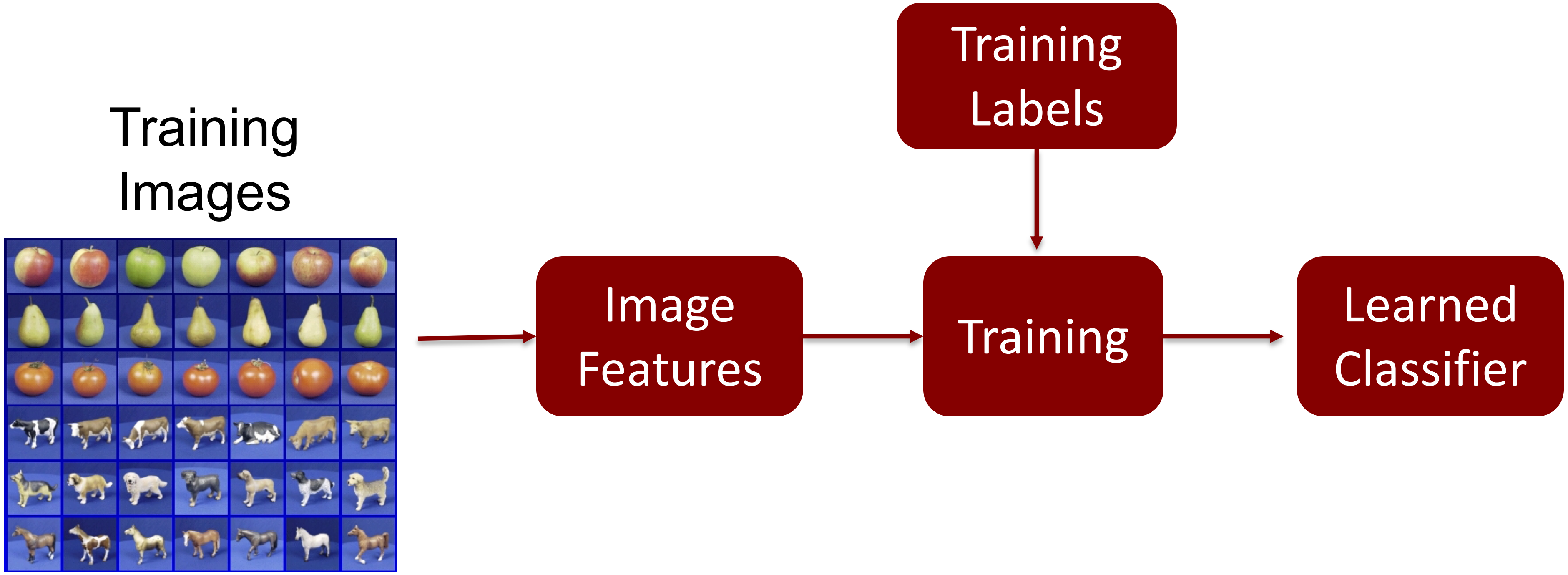
Training Images

→ Image Features

A simple pipeline - Training



A simple pipeline - Training



A simple pipeline - Training

Training Images

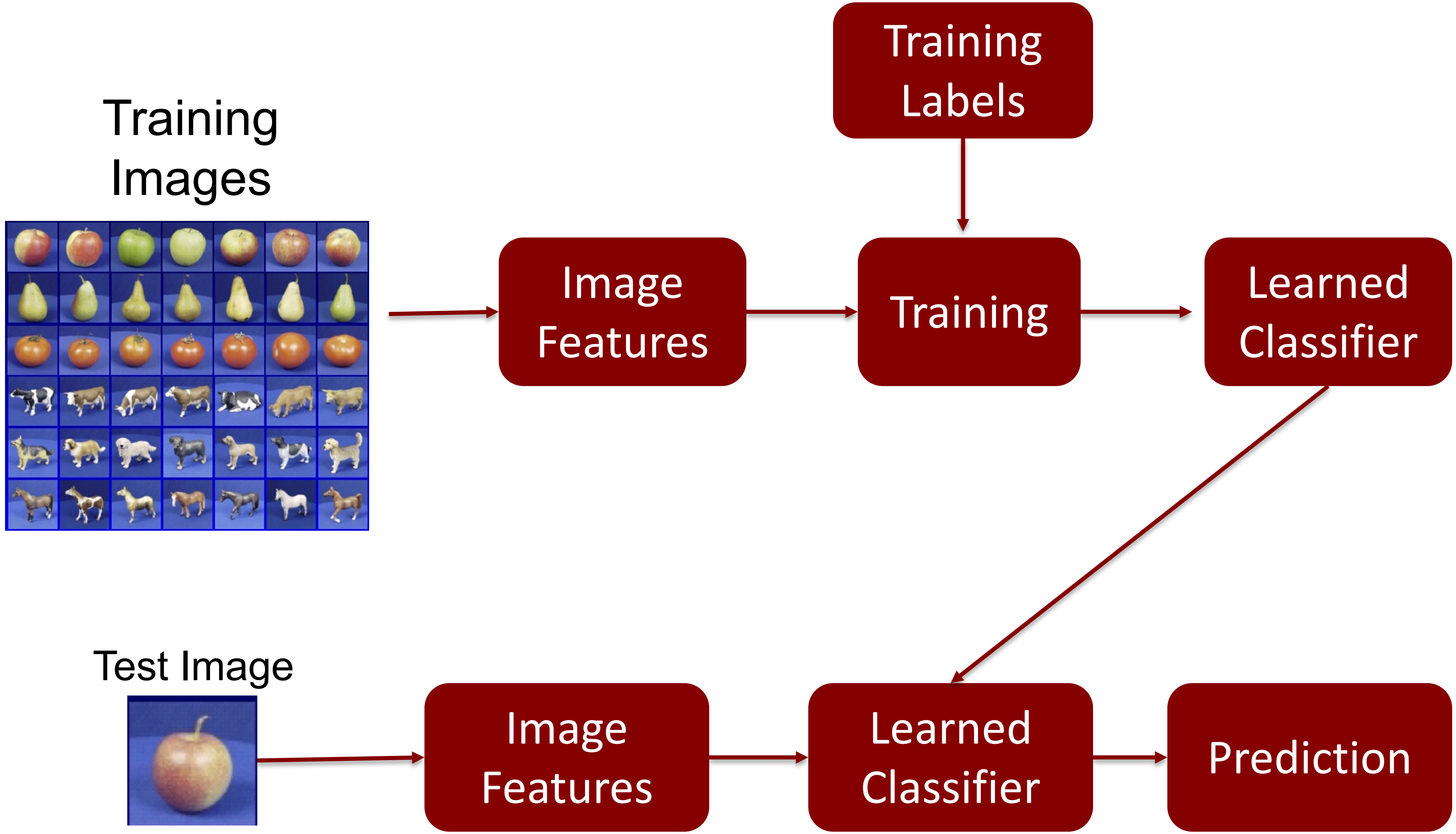
Image Features → Training → Learned Classifier

Training Labels → Training

Test Image

Image Features

A simple pipeline - Training



A simple pipeline - Training

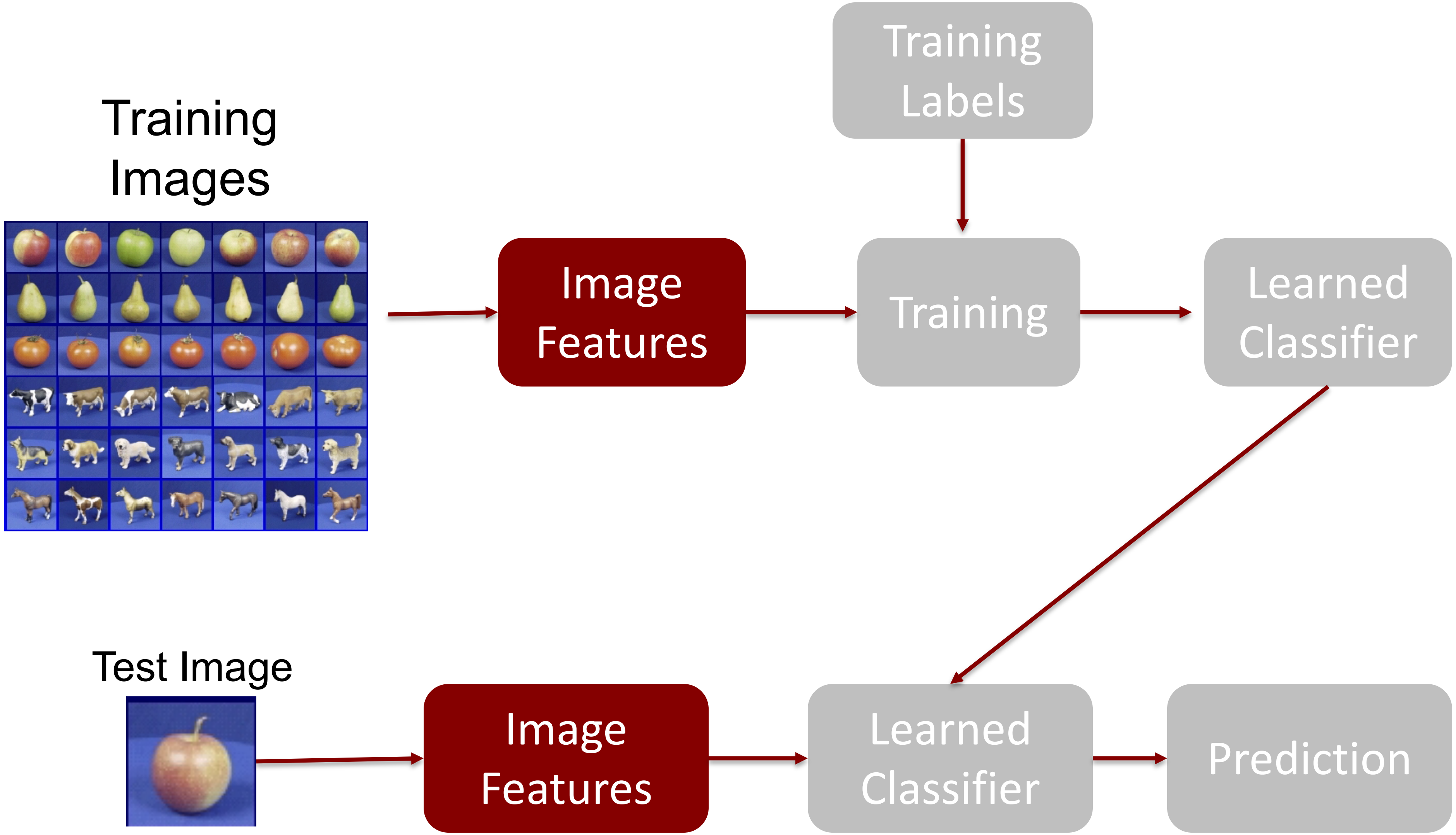
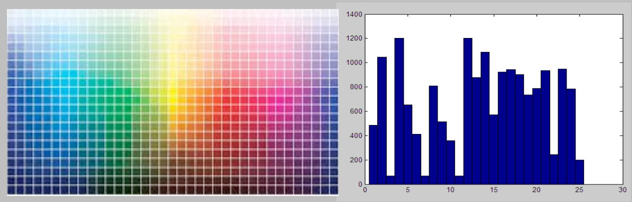


Image features

Input image

Color: Quantize RGB values

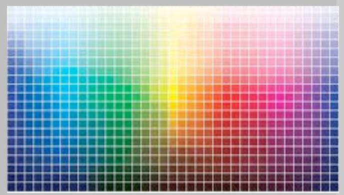


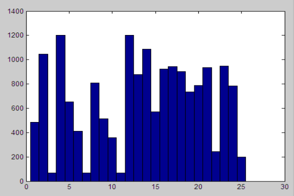
Invariance?
- ? Translation
- ? Scale
- ? Rotation
- ? Occlusion

Image features

Input image

Color: Quantize RGB values





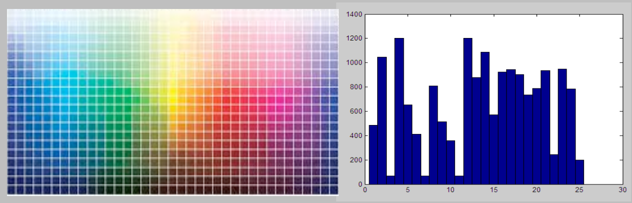
Invariance?

- 🙂 Translation
- 🙁 Scale
- 🙂 Rotation (in-planar)
- 🙁 Occlusion

Image features

Input image

<u>Color:</u> Quantize RGB values



<u>Invariance?</u>
- 🙂 Translation
- 🙁 Scale
- 🙂 Rotation (in-planar)
- 🙁 Occlusion

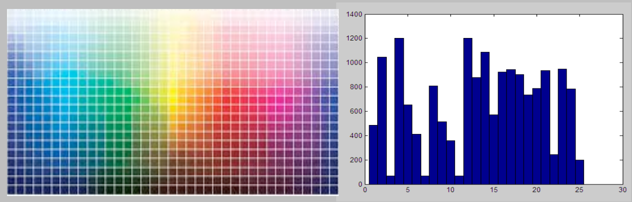
<u>Global shape:</u> PCA space

<u>Invariance?</u>
- ? Translation
- ? Scale
- ? Rotation (in-planar)
- ? Occlusion

Image features

Input image

Color: Quantize RGB values



Invariance?
- 🙂 Translation
- 🙁 Scale
- 🙂 Rotation (in-planar)
- 🙁 Occlusion

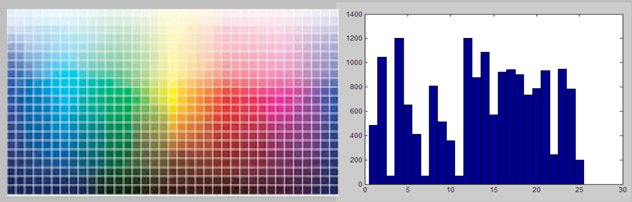
Global shape: PCA space

Invariance?
- 🙂 Translation
- 🙁 Scale
- 🙂 Rotation (in-planar)
- 🙁 Occlusion

Image features

Input image

Color: Quantize RGB values

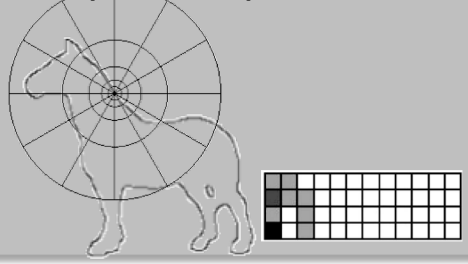


Invariance?
- Translation
- Scale
- Rotation
- Occlusion

Global shape: PCA space

Invariance?
- Translation
- Scale
- Rotation
- Occlusion

Local shape: shape context

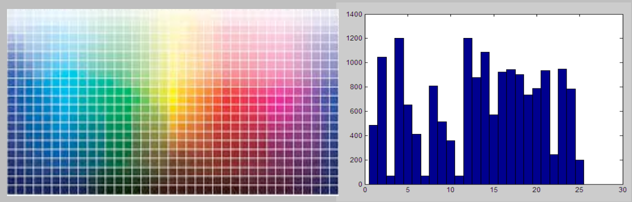


Invariance?
- ? Translation
- ? Scale
- ? Rotation (in-planar)
- ? Occlusion

Image features

Input image

Color: Quantize RGB values

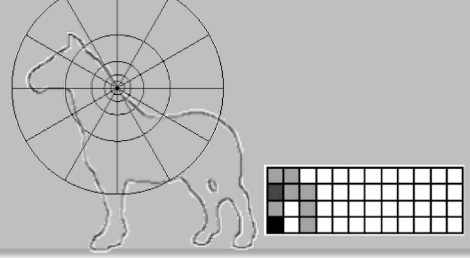


Invariance?
- 🙂 Translation
- 🙂 Scale
- 🙂 Rotation
- 🙁 Occlusion

Global shape: PCA space

Invariance?
- 🙂 Translation
- 🙁 Scale
- 🙂 Rotation
- 🙁 Occlusion

Local shape: shape context

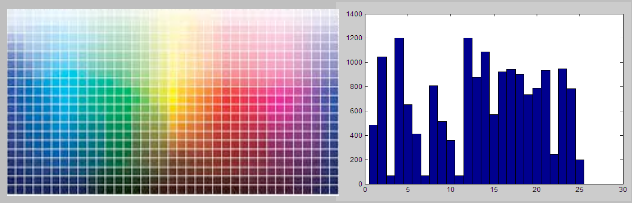


Invariance?
- 🙂 Translation
- 🙂 Scale
- 🙂 Rotation (in-planar)
- 🙁 Occlusion

Image features

Input image

Color: Quantize RGB values

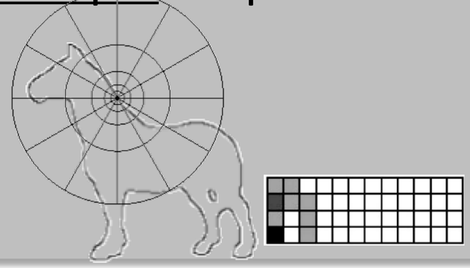


Invariance?
- 🙂 Translation
- 🙂 Scale
- 🙂 Rotation
- 🙁 Occlusion

Global shape: PCA space

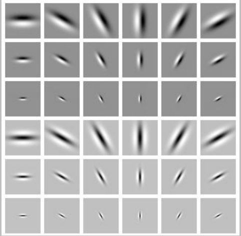
Invariance?
- 🙂 Translation
- 🙁 Scale
- 🙂 Rotation
- 🙁 Occlusion

Local shape: shape context



Invariance?
- 🙂 Translation
- 🙂 Scale
- 🙂 Rotation (in-planar)
- 🙁 Occlusion

Texture: Filter banks

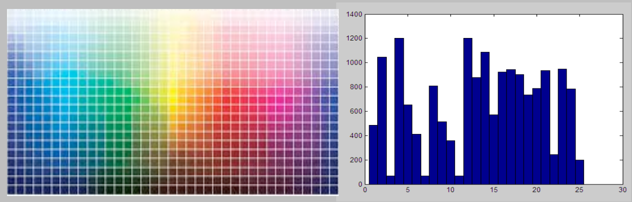


Invariance?
- ? Translation
- ? Scale
- ? Rotation (in-planar)
- ? Occlusion

Image features

Input image

Color: Quantize RGB values

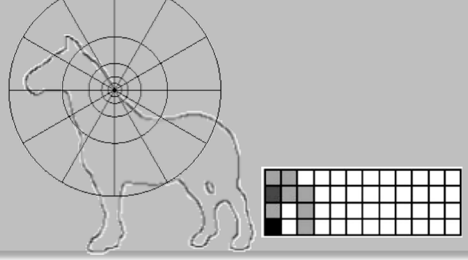


Invariance?
- 🙂 Translation
- 🙂 Scale
- 🙂 Rotation
- 🙁 Occlusion

Global shape: PCA space

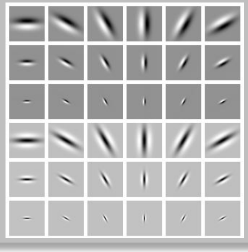
Invariance?
- 🙂 Translation
- 🙁 Scale
- 🙂 Rotation
- 🙁 Occlusion

Local shape: shape context



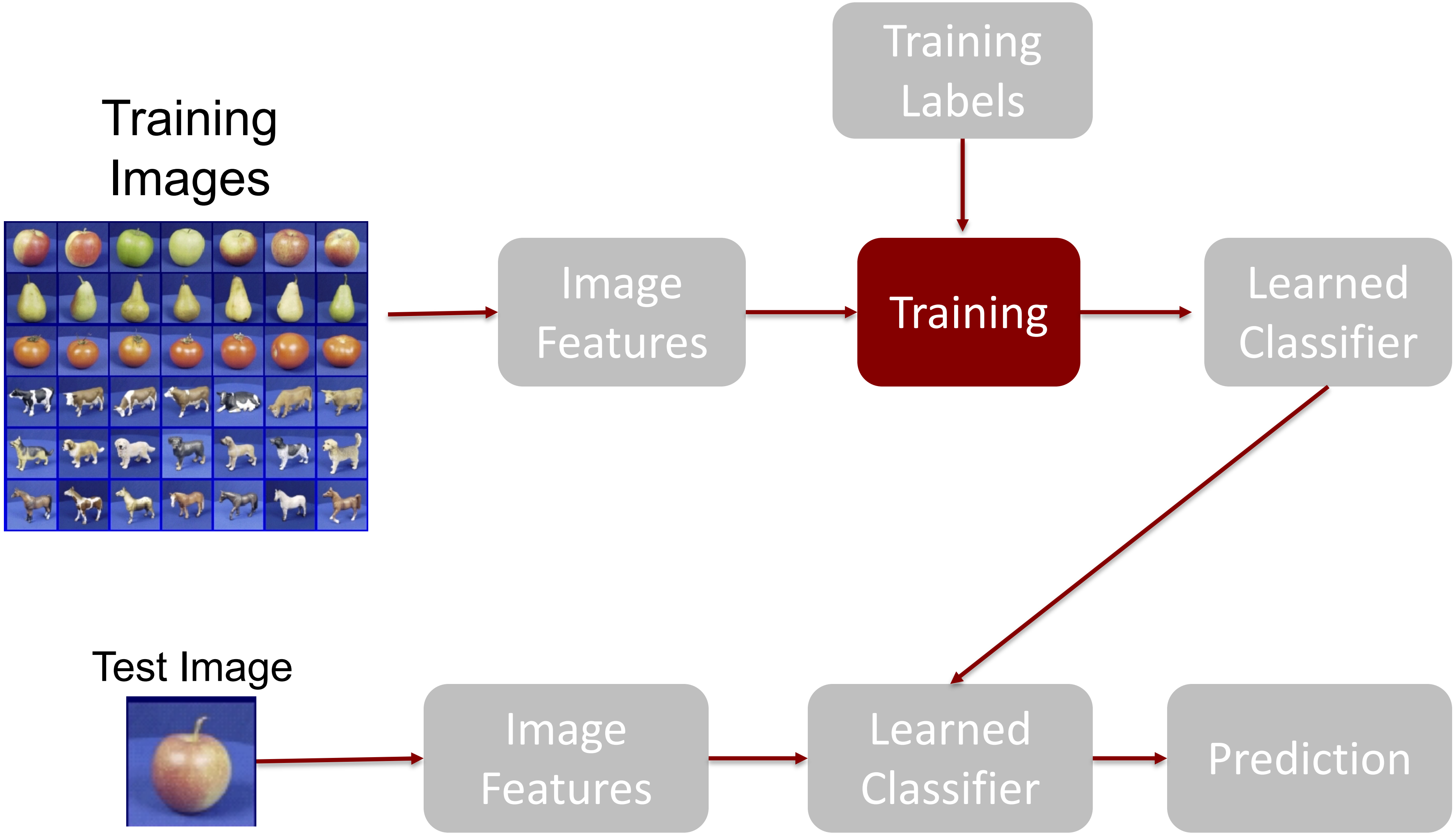
Invariance?
- 🙂 Translation
- 🙂 Scale
- 🙂 Rotation (in-planar)
- 🙁 Occlusion

Texture: Filter banks

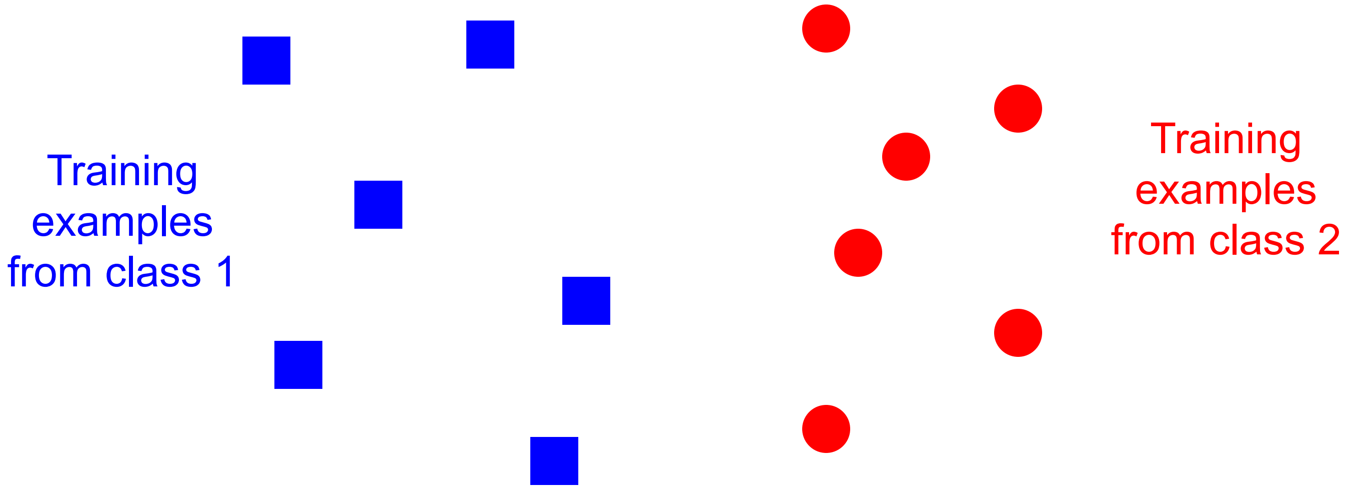


Invariance?
- 🙂 Translation
- ? Scale
- ? Rotation (in-planar)
- 🙁 Occlusion

A simple pipeline - Training

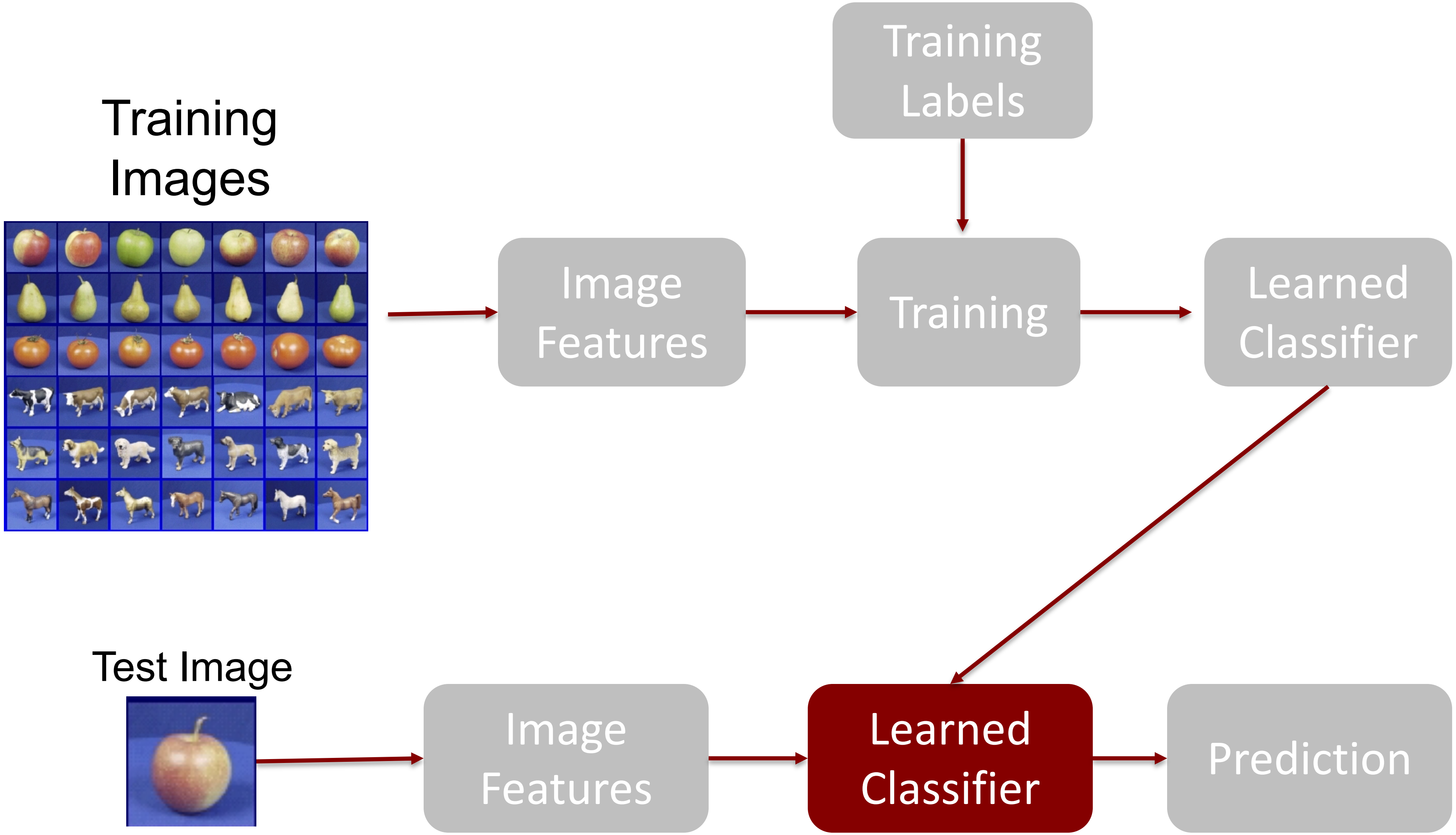


Classifiers: Nearest neighbor

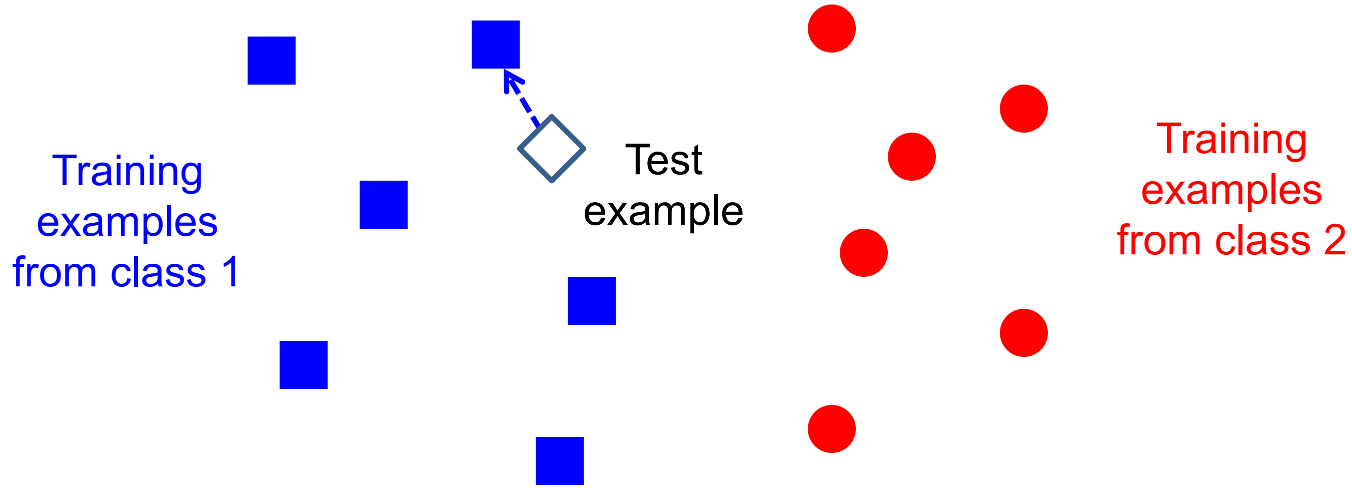


Slide credit: L. Lazebnik

A simple pipeline - Training



Classifiers: Nearest neighbor



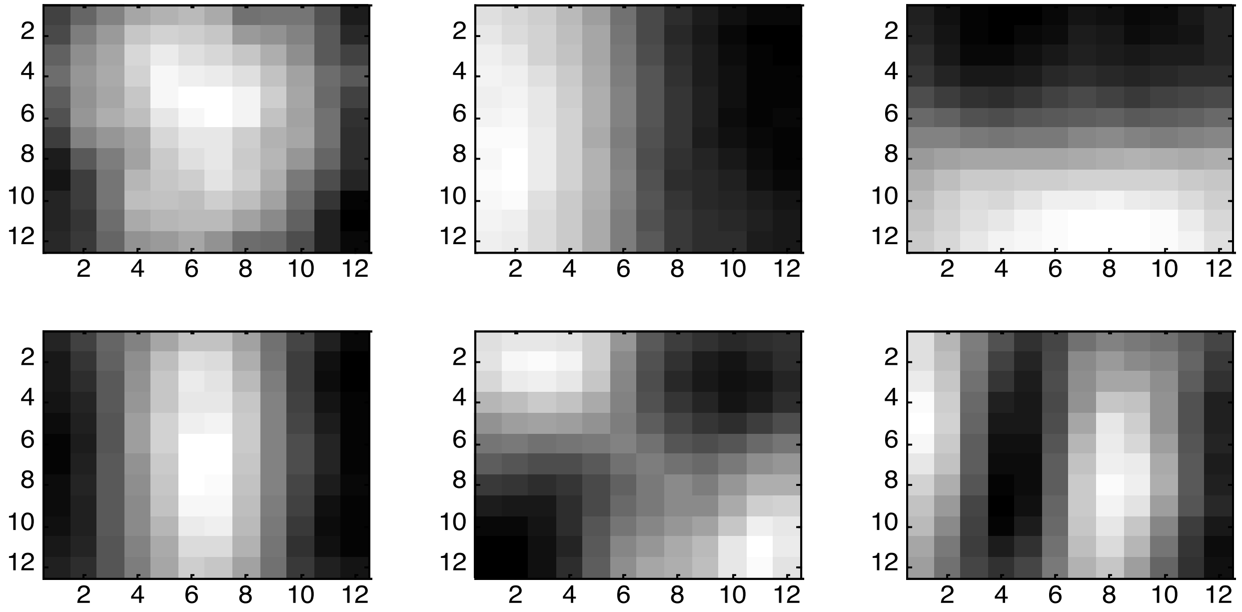
Slide credit: L. Lazebnik

Let’s recap

- A simple object recognition pipeline with kNN
- PCA

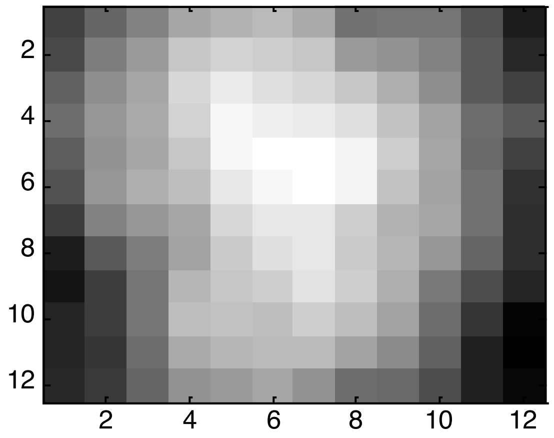
PCA compression: 144D -> 6D

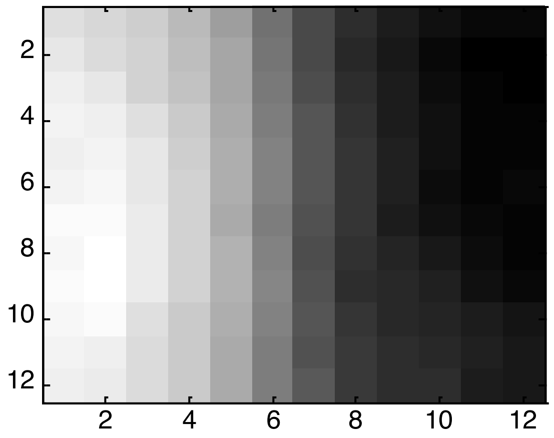
6 most important eigenvectors

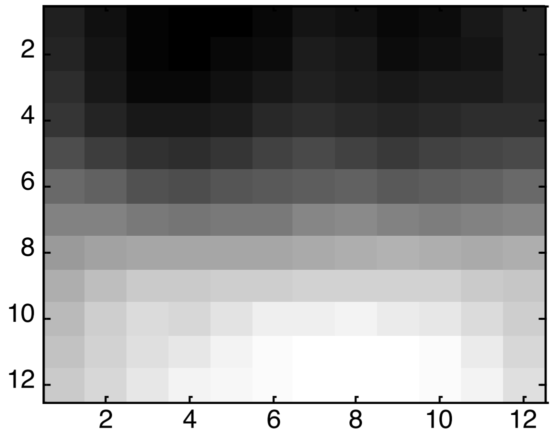


PCA compression: 144D) 3D

3 most important eigenvectors







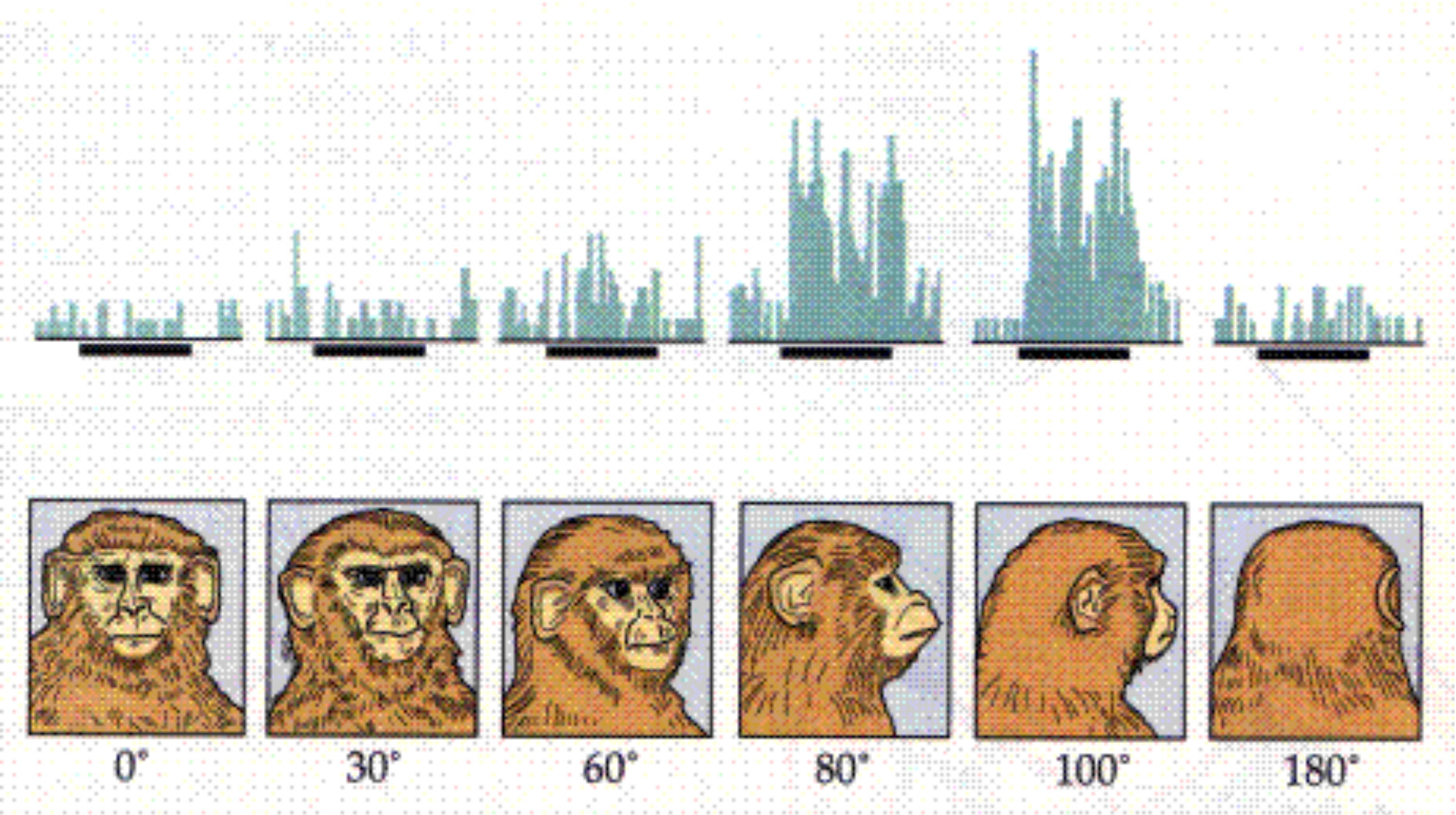
What we will learn today

- 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.

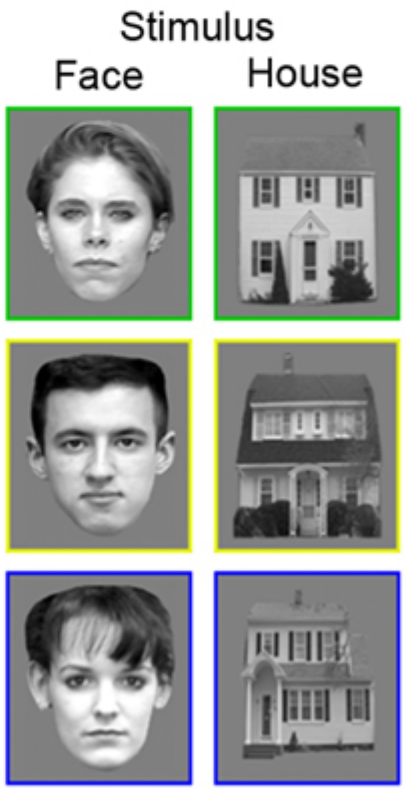
P. Belhumeur, J. Hespanha, and D. Kriegman. "Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection". *IEEE Transactions on pattern analysis and machine intelligence* **19** (7): 711. 1997.

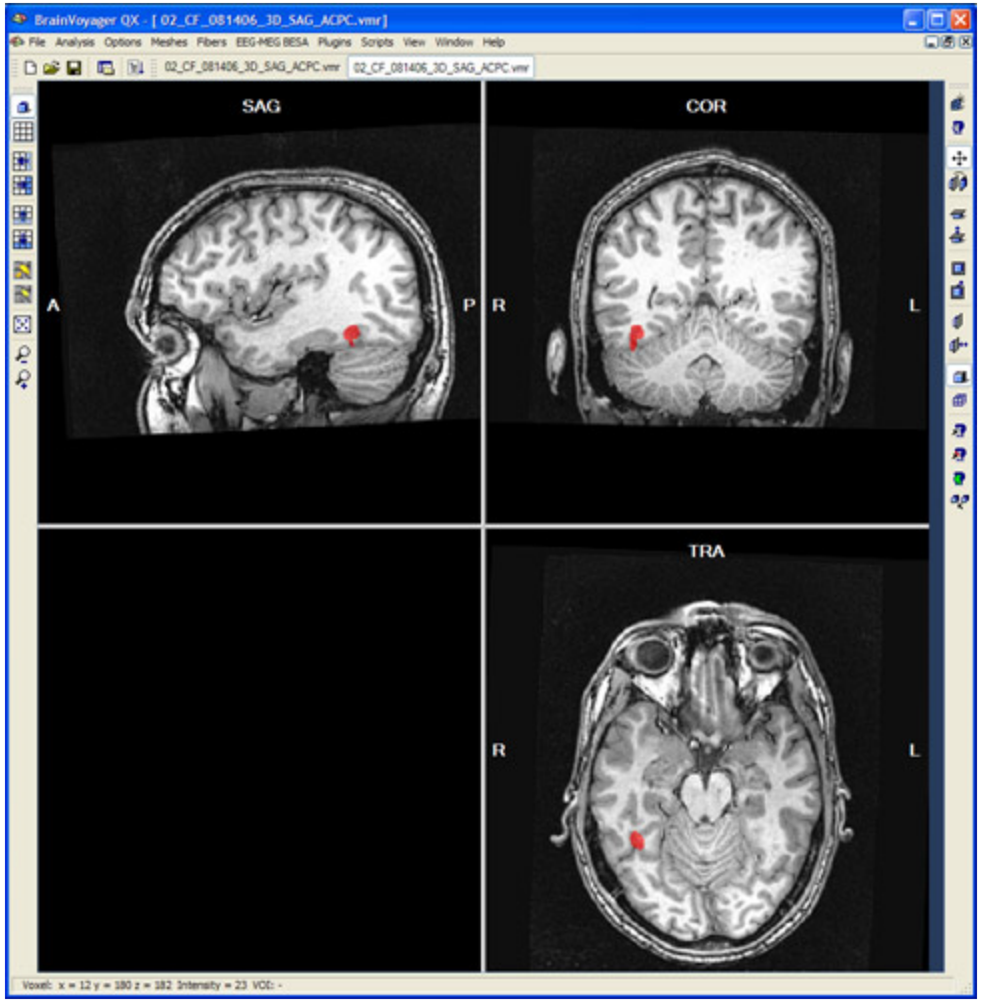
“Faces” in the brain



Courtesy of Johannes M. Zanker

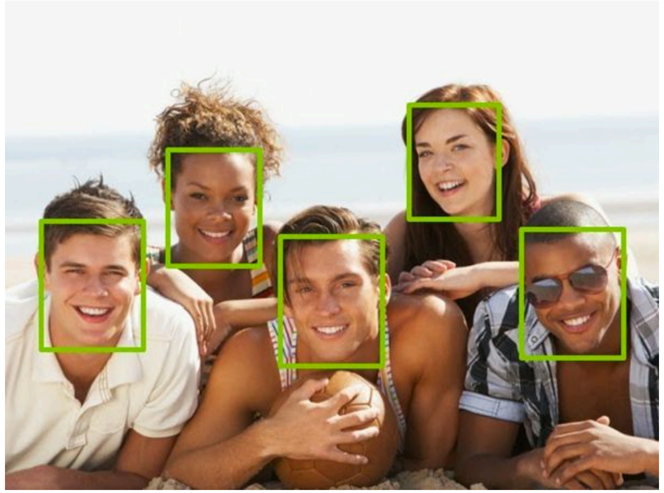
“Faces” in the brain fusiform **face area**



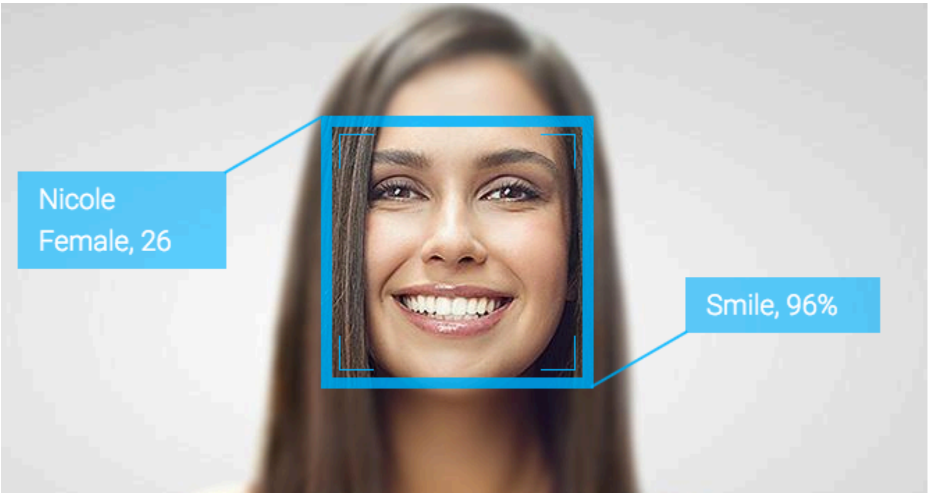


Kanwisher, et al. 1997

Detection versus Recognition



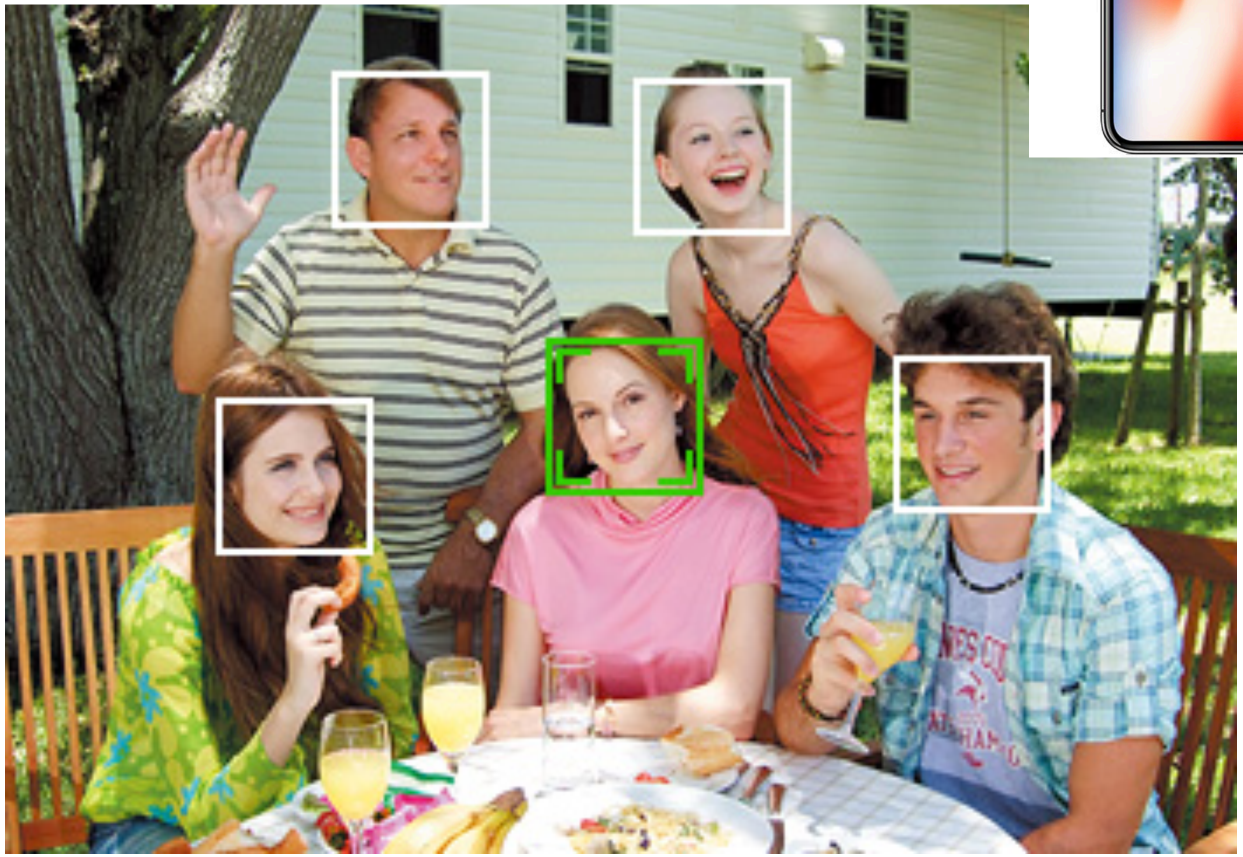
Detection finds the faces in images



Recognition recognizes WHO the person is

Face Recognition

- Digital photography



Face Recognition

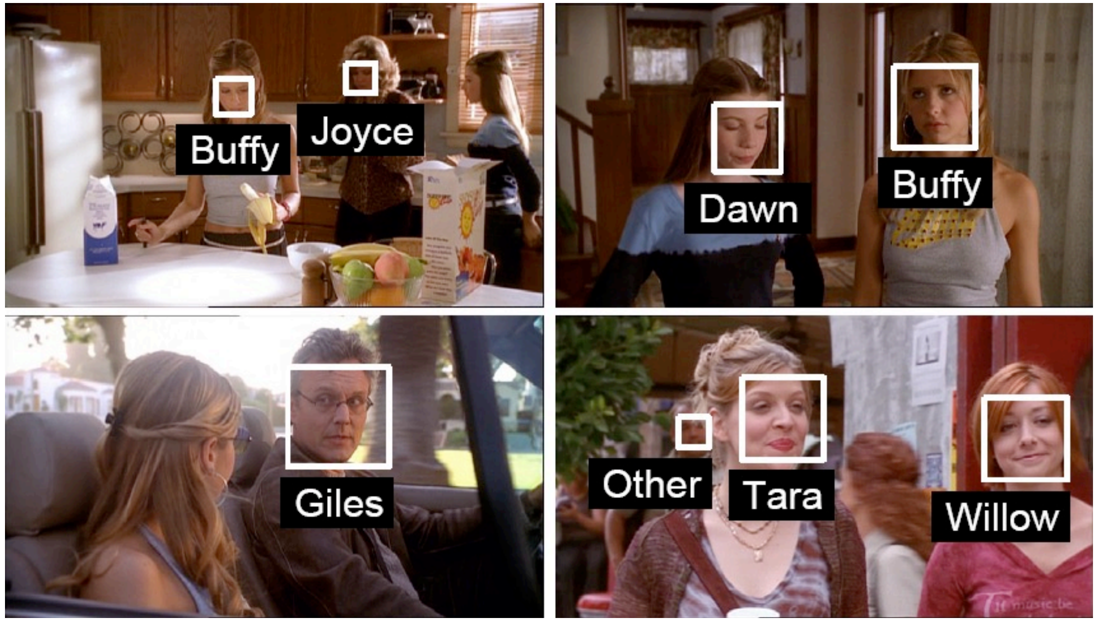
- Digital photography
- Surveillance

Face Recognition

- Digital photography
- Surveillance
- Album organization

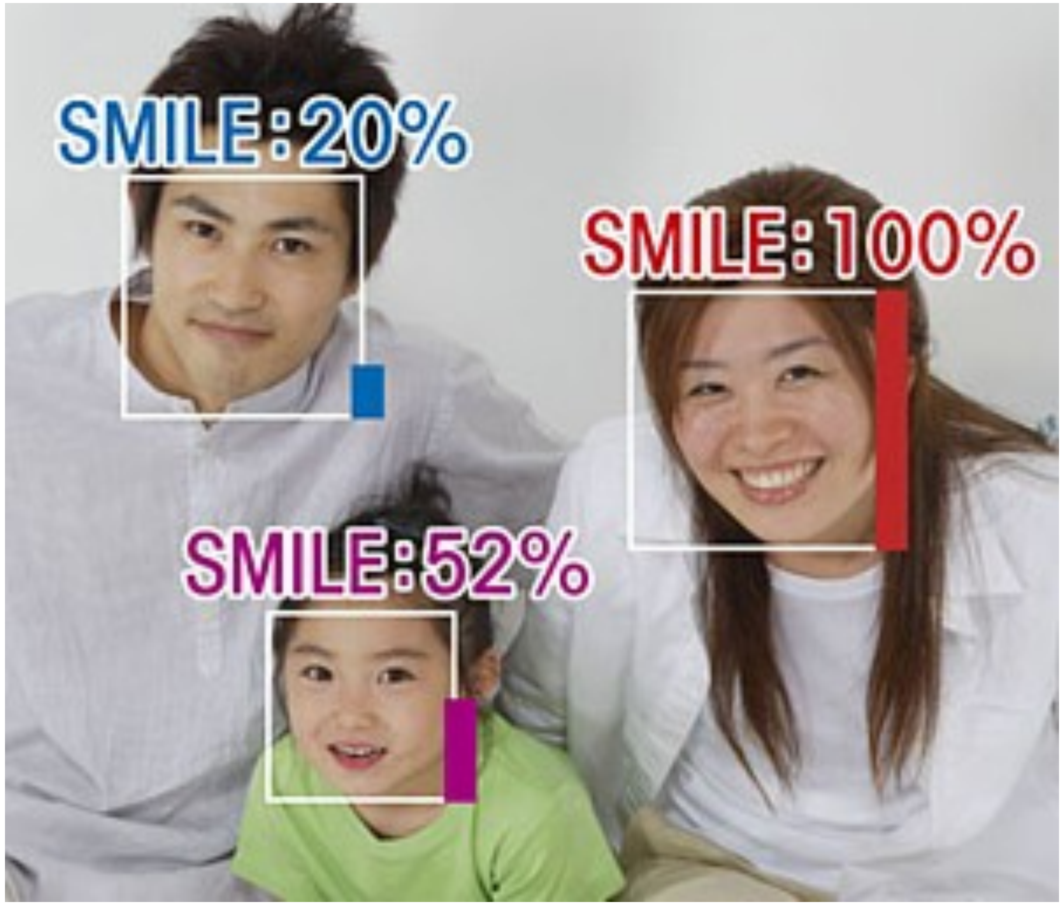
Face Recognition

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



Face Recognition

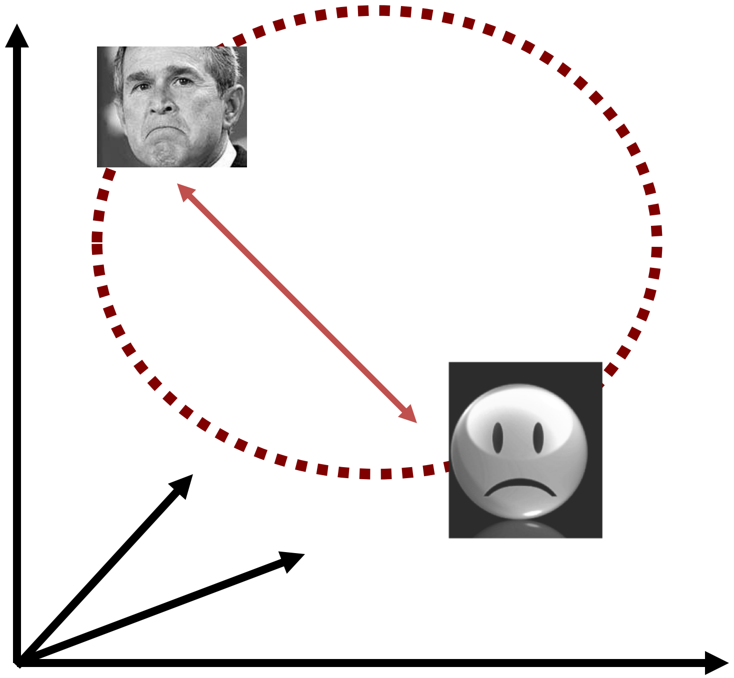
- Digital photography
- Surveillance
- Album organization
- Person tracking/id.
- Emotions and expressions



Face Recognition

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

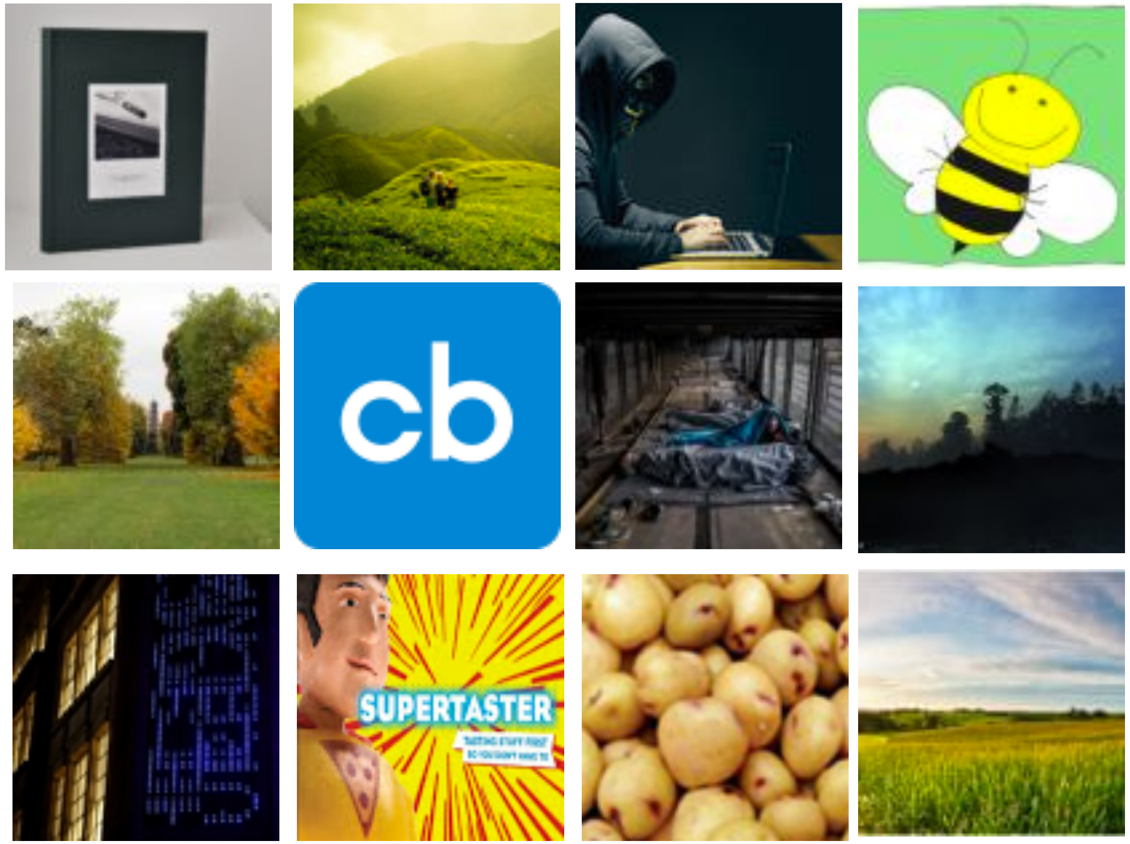
The Space of Faces



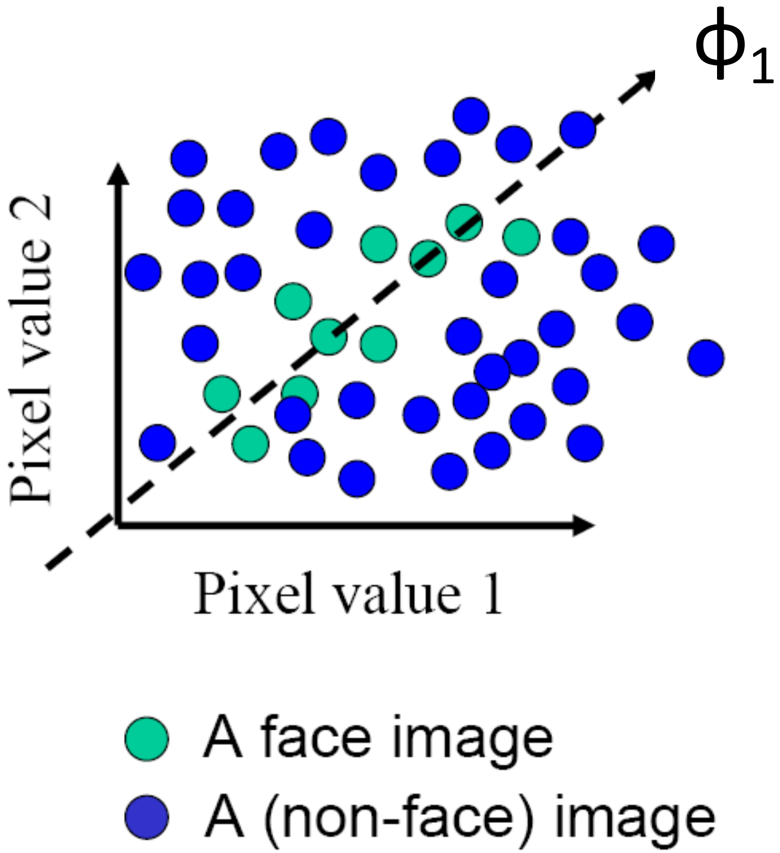
- An image is a point in a high dimensional space
  - If represented in grayscale intensity, an N x M image is a point in R^{NM}
  - E.g. 100x100 image = 10,000 dim

Slide credit: Chuck Dyer, Steve Seitz, Nishino

100x100 images can contain many things other than faces!



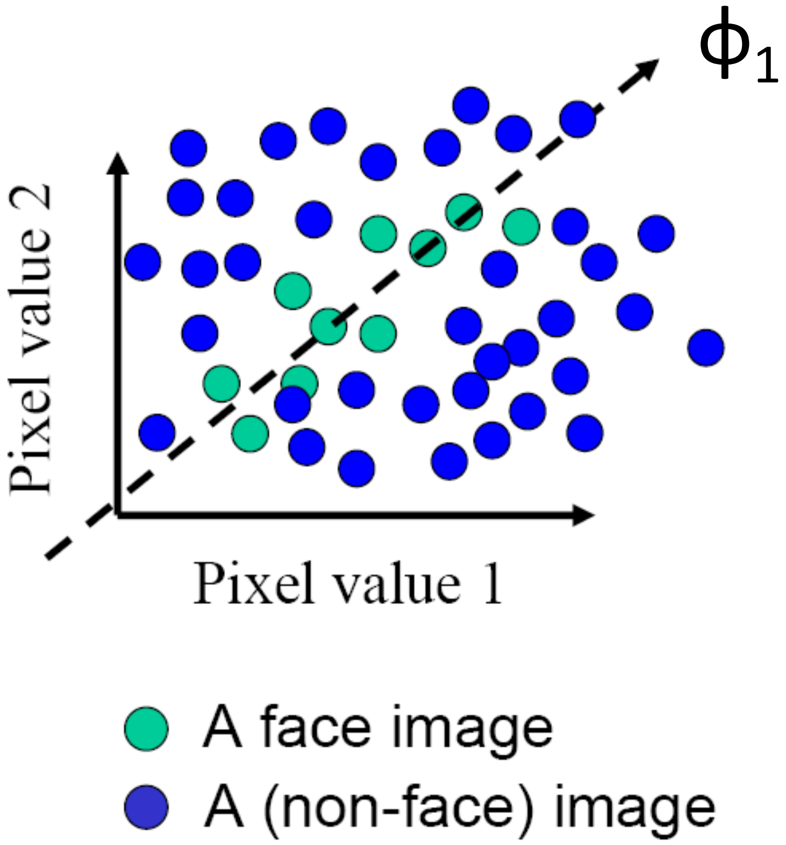
The Space of Faces

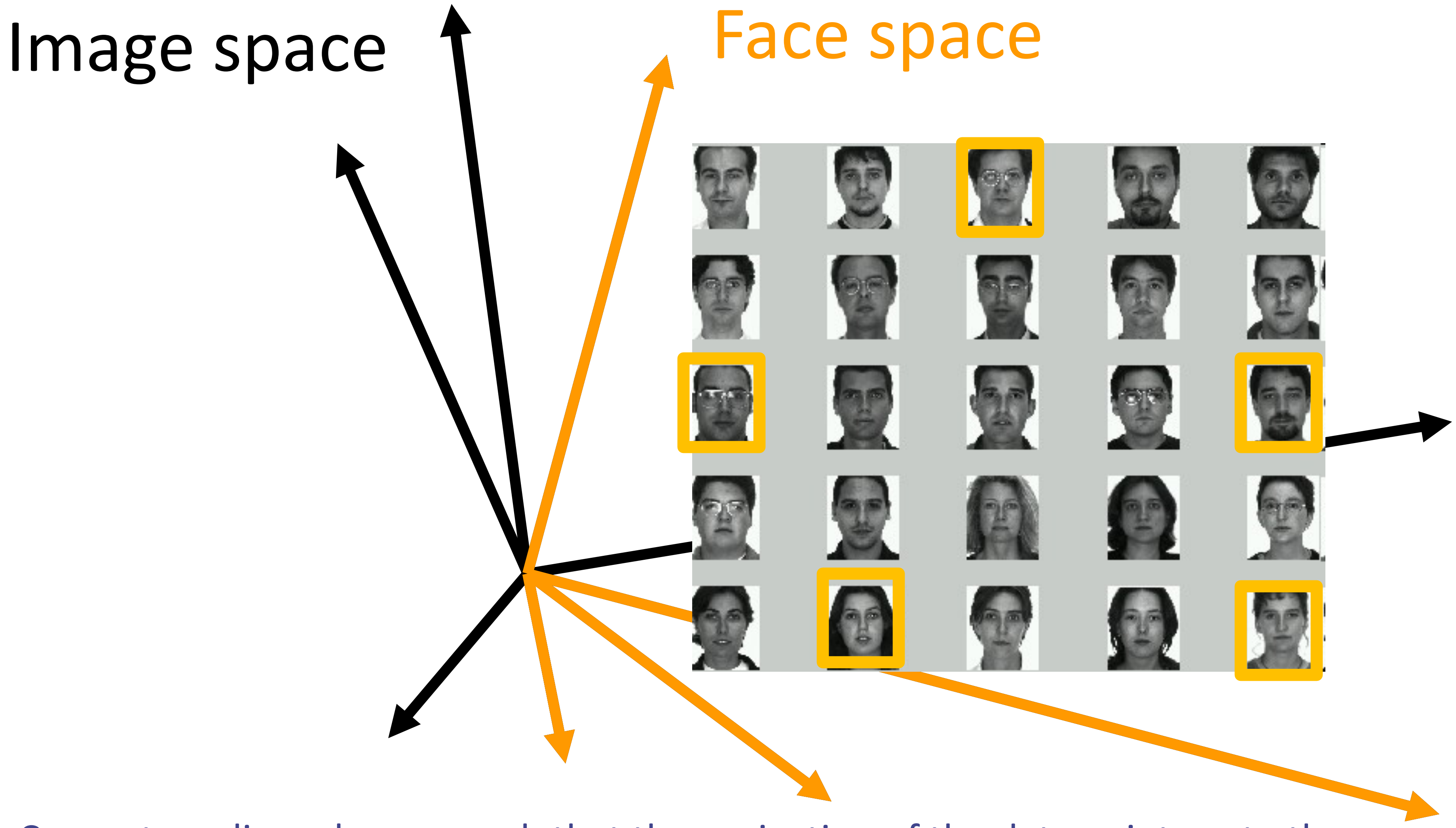


- An image is a point in a high dimensional space
  - If represented in grayscale intensity, an N x M image is a point in R^{NM}
  - E.g. 100x100 image = 10,000 dim
- However, relatively few high dimensional vectors correspond to valid face images
- We want to effectively model the subspace of face images

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

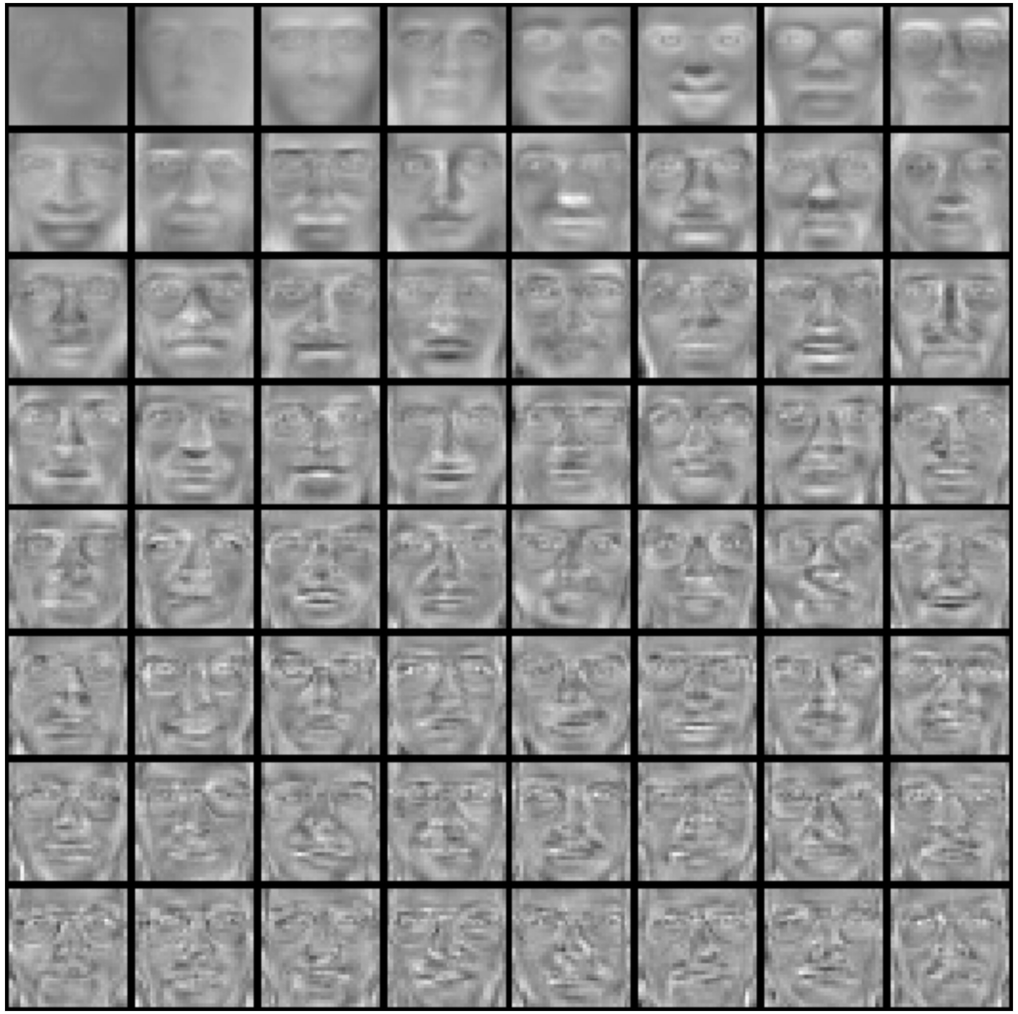
- Assume that most face images lie on a low-dimensional subspace determined by the first k (k<<d) directions of maximum variance
- Use PCA to determine the vectors or “eigenfaces” that span that subspace
- Represent all face images in the dataset as linear combinations of eigenfaces

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

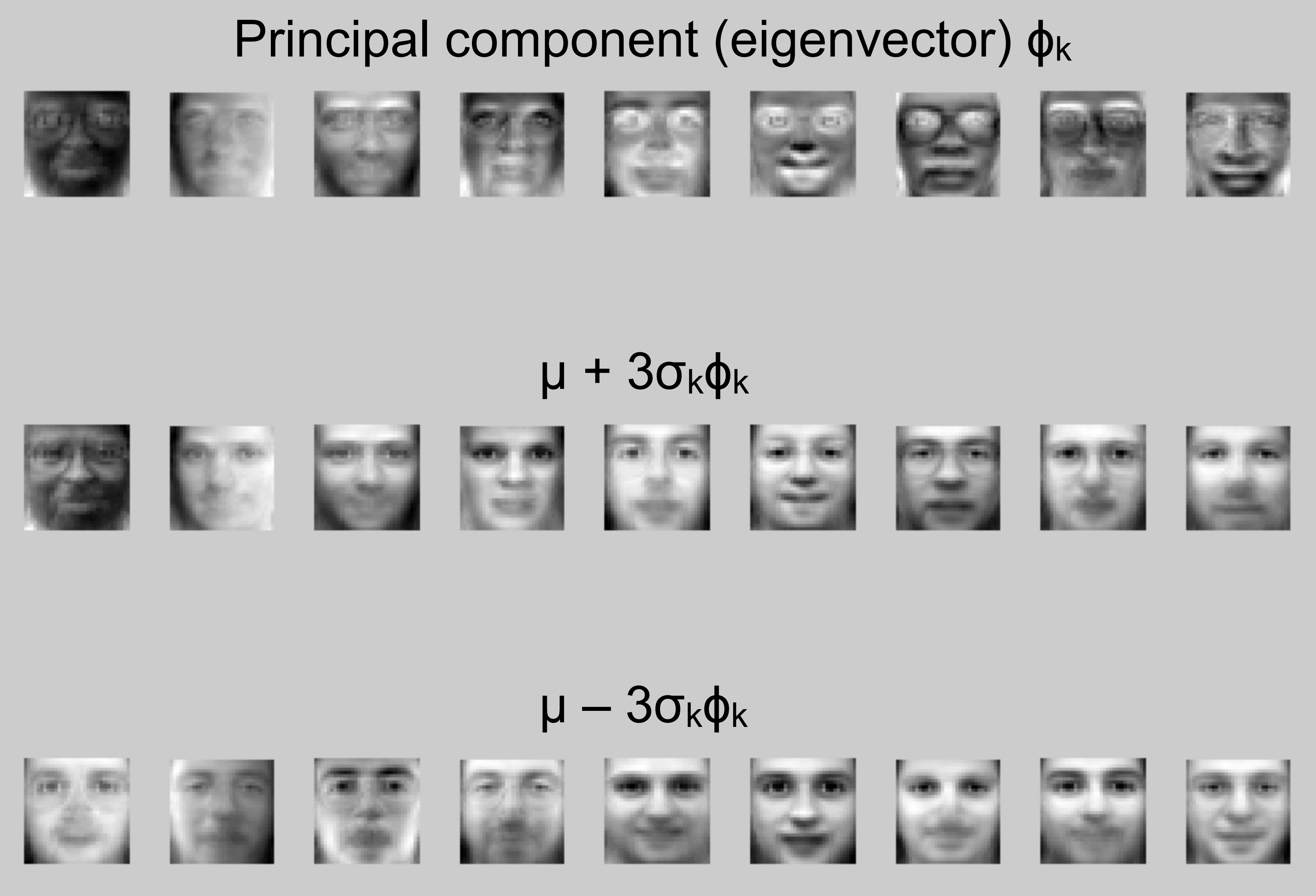
Training images: $\mathbf{x}_1,\ldots,\mathbf{x}_N$

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

Mean: μ



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

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

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

$$= X - \mu 1^T = X - \frac{1}{n} X 1 1^T = X\left(I - \frac{1}{n} 1 1^T \right)$$

Eigenface algorithm

4. Compute the covariance matrix

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

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

$$x_i \rightarrow \left(x_i^c \cdot \phi_1, x_i^c \cdot \phi_2, \dots, x_i^c \cdot \phi_K\right) \equiv \left(a_1, a_2, \dots, a_K\right)$$

7. Visualize the estimated training face x_i

$$x_i \approx \mu + a_1\phi_1 + a_2\phi_2 + \dots + a_K\phi_K$$

Eigenface algorithm

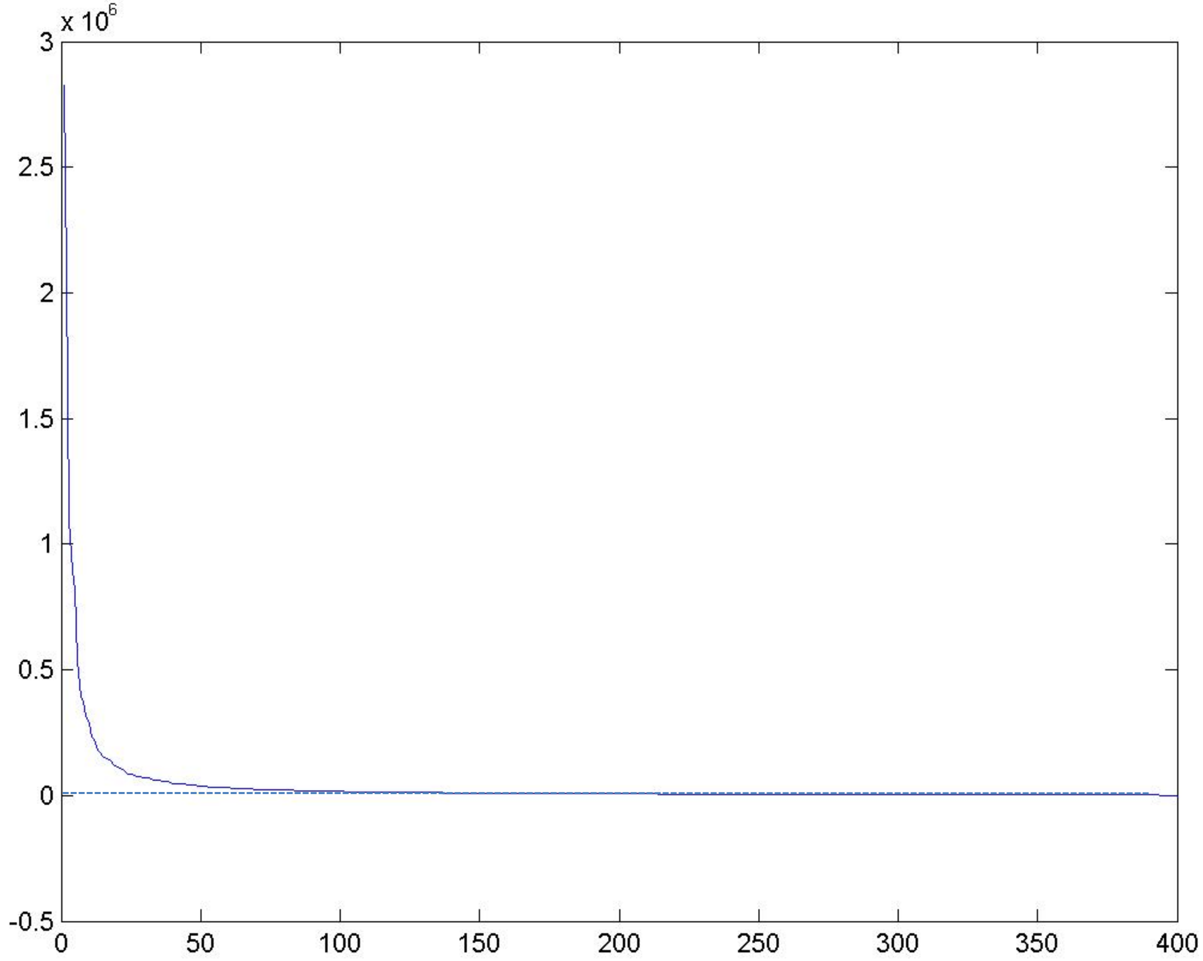
6. Compute each training image x_i 's projections as

$$x_i \rightarrow \left(x_i^c \cdot \phi_1, x_i^c \cdot \phi_2, \ldots, x_i^c \cdot \phi_K \right) \equiv \left(a_1, a_2, \ldots, a_K \right)$$

7. Visualize the reconstructed training face x_i

$$x_i \approx \mu + a_1\phi_1 + a_2\phi_2 + \ldots + a_K\phi_K$$

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

$$t \rightarrow \left((t-\mu)\cdot\phi_1, (t-\mu)\cdot\phi_2, \ldots, (t-\mu)\cdot\phi_K \right) \equiv \left(w_1, w_2, \ldots, w_K \right)$$

  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)
  - Makes no effort to preserve class distinctions

Summary for Eigenface

Pros

- Non-iterative, globally optimal solution

Limitations

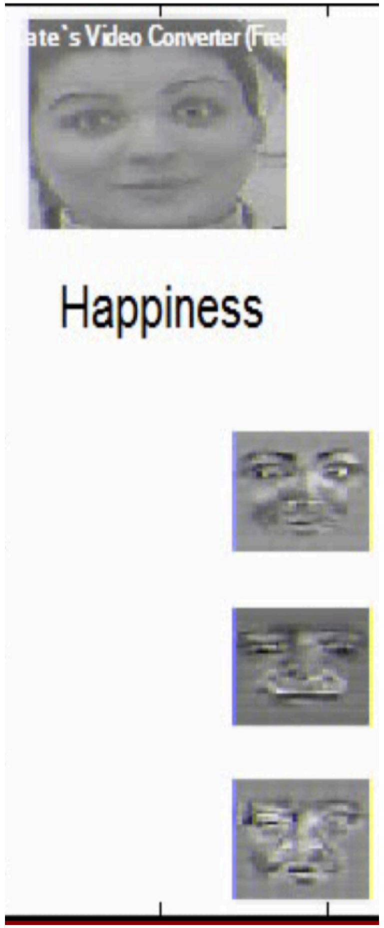
• PCA projection is **optimal for reconstruction** from a low dimensional basis, but **may NOT be optimal for discrimination…** Is there a better dimensionality reduction?

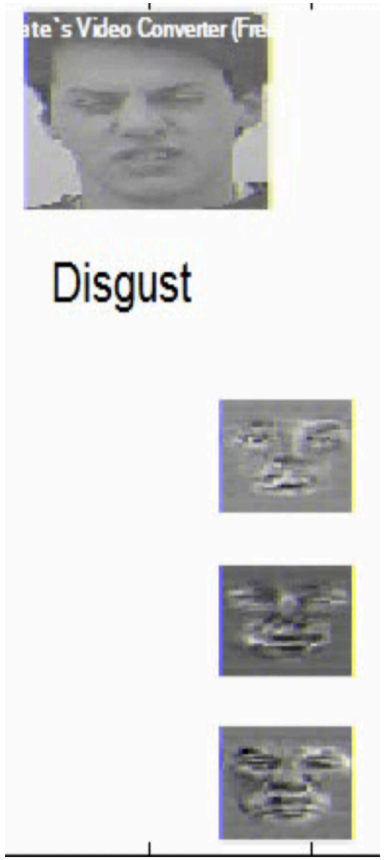
Besides face recognitions, we can also do Facial expression recognition

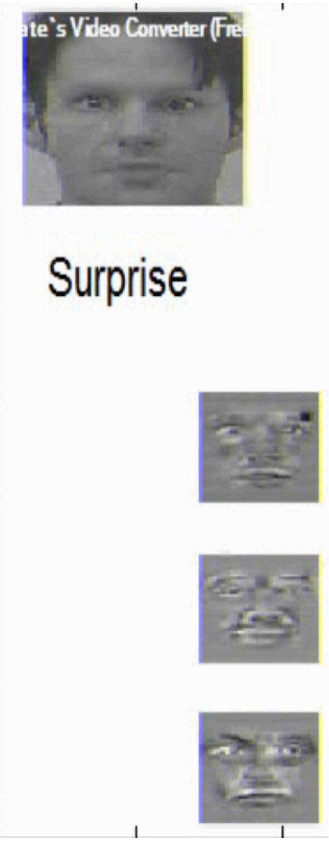
Happiness subspace (method A)

Disgust subspace (method A)

Facial Expression Recognition Movies (method A)







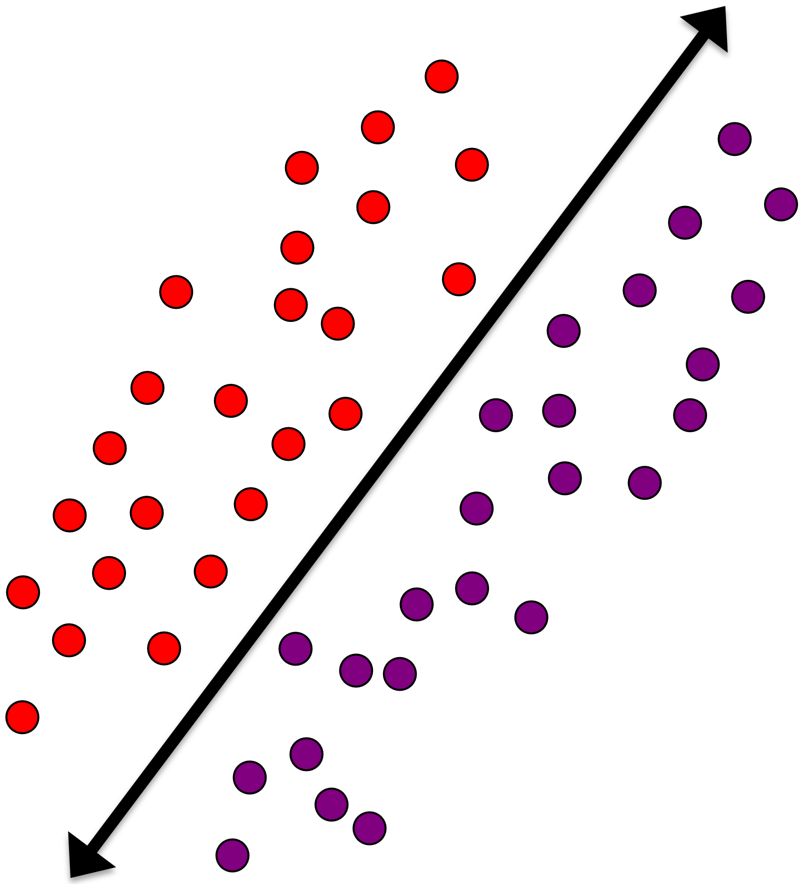
What we will learn today

- 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.

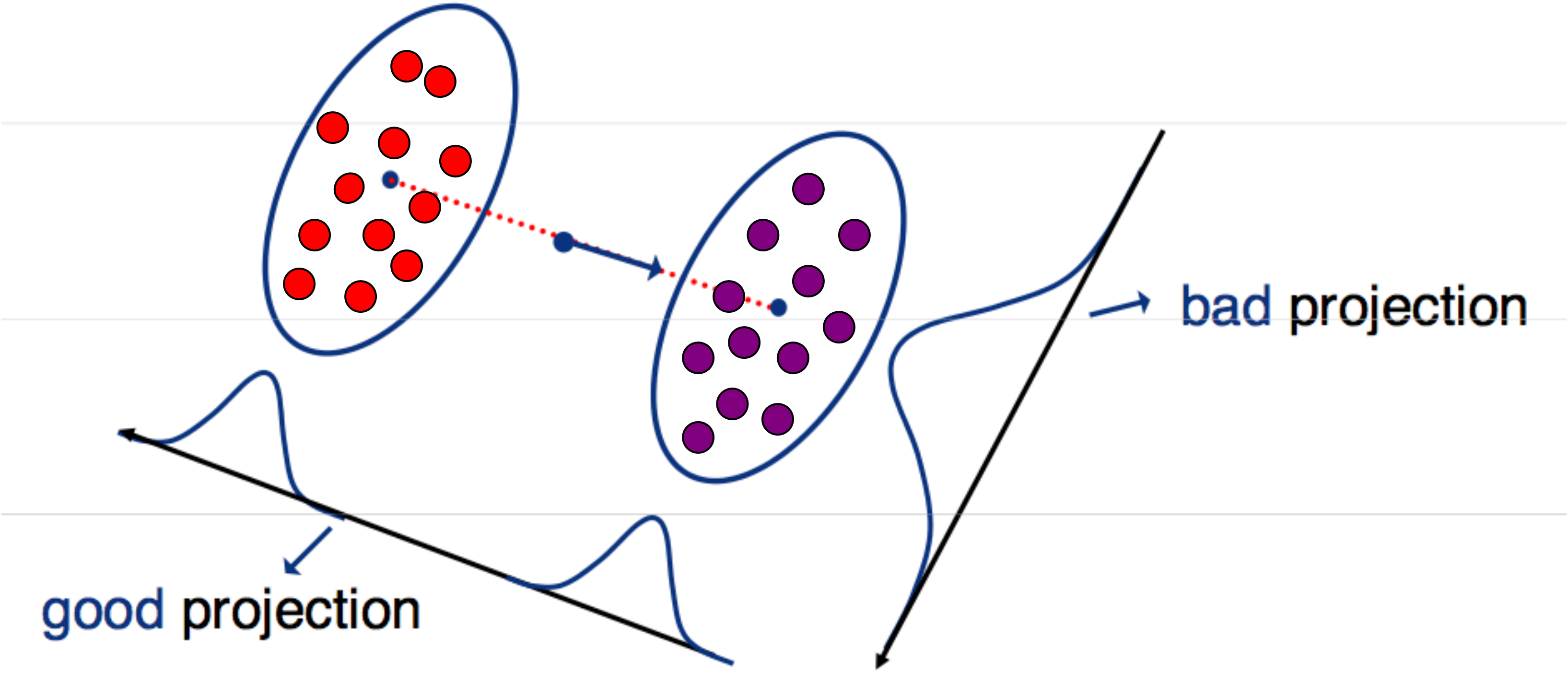
P. Belhumeur, J. Hespanha, and D. Kriegman. "Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection". *IEEE Transactions on pattern analysis and machine intelligence* **19** (7): 711. 1997.

Which direction will is the first principle component?



Fischer’s Linear Discriminant Analysis

- Goal: find the best separation between two classes



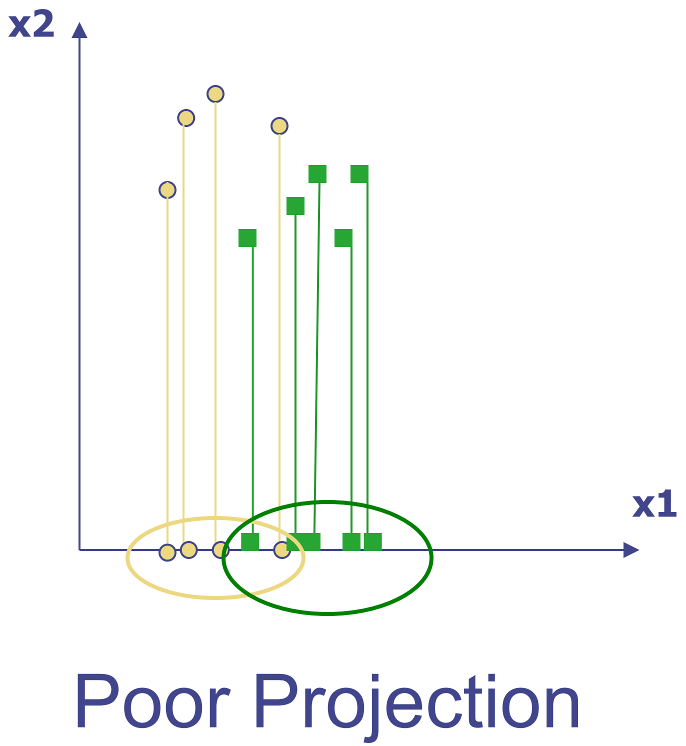
Slide inspired by N. Vasconcelos

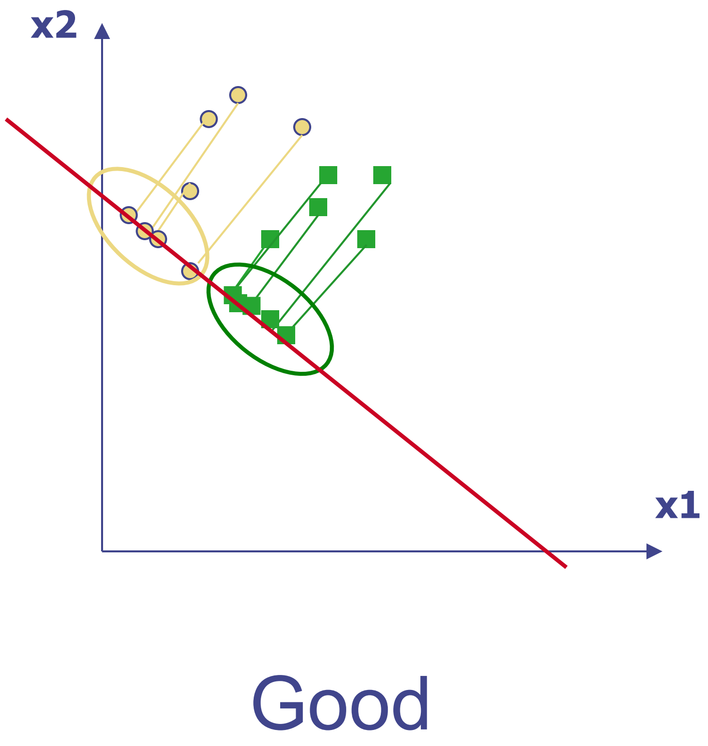
Difference between PCA and LDA

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

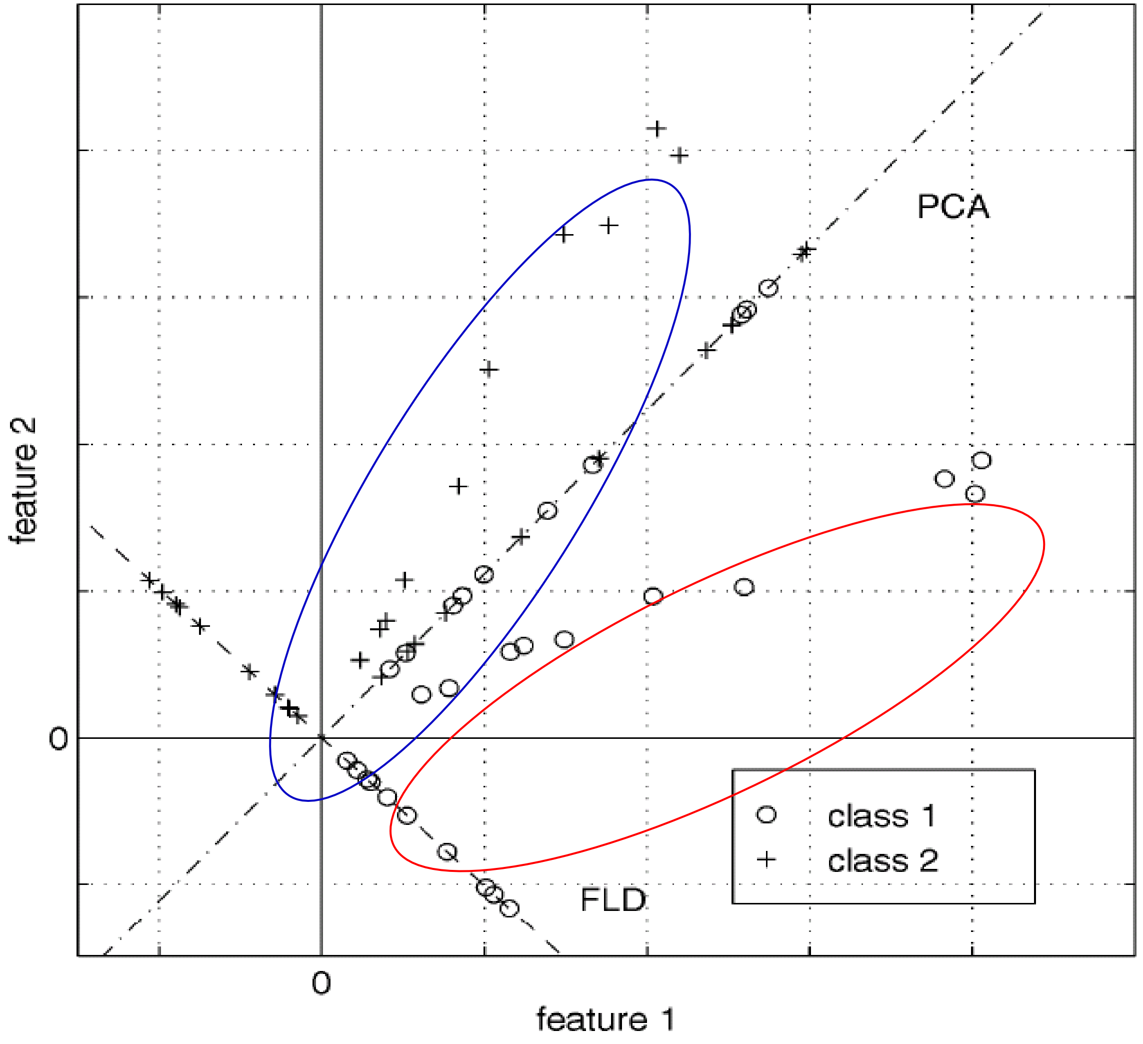
Illustration of the Projection

- Using two classes as example:





Basic intuition: PCA vs. LDA



LDA with 2 variables

- 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):

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

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

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

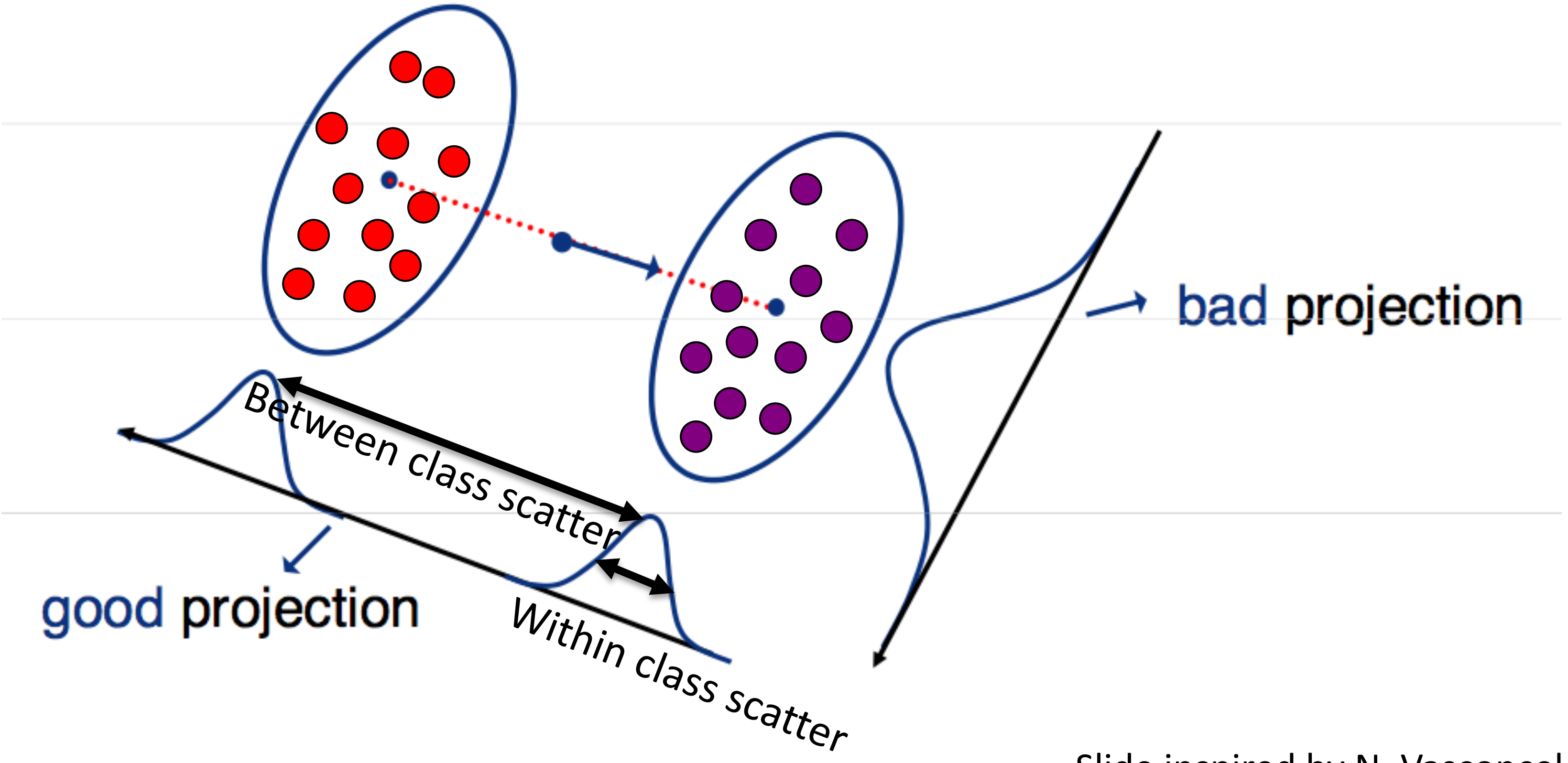
- And the **per class** covariance matrices be:

$$[(X - \mu_i)(X - \mu_i)^T \mid Y = i] = \Sigma_i$$

- 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

LDA with 2 variables

The following objective function:

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

Can be written as

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

LDA with 2 variables

- We can write the between class scatter as:

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

- Also, the within class scatter becomes:

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

Slide inspired by N. Vasconcelos

LDA with 2 variables

- 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:

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

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

Slide inspired by N. Vasconcelos

LDA with 2 variables

- The scatter variables

between class scatter

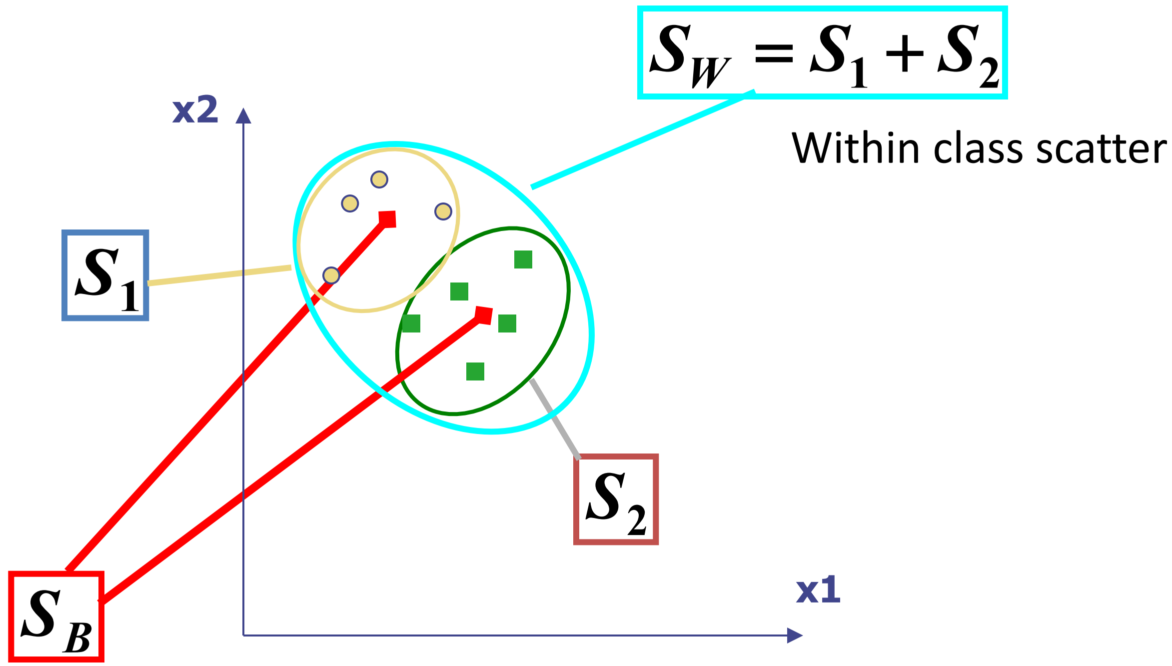
$$S_B = (\mu_1 - \mu_0)(\mu_1 - \mu_0)^T$$

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

within class scatter

Slide inspired by N. Vasconcelos

Visualization



Between class scatter

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.

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

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

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

- And maximize with respect to both w and λ

Slide inspired by N. Vasconcelos

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 $S_w^{-1} = (\Sigma_1 + \Sigma_0)^{-1}$

Slide inspired by N. Vasconcelos

Linear Discriminant Analysis (LDA)

- In this case

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

- And using the definition of S_B

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

- Assuming 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

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

Slide inspired by N. Vasconcelos

LDA with N variables and C classes

Variables

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

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: $$S_W = \sum_{i=1}^{c} S_i$$
- Between class scatter: $$S_B = \sum_{i=1}^{C} \sum_{j \neq i} (\mu_i - \mu_j)(\mu_i - \mu_j)^T$$

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:

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

- 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:

$$W_{opt} = \arg\max_{\mathbf{W}} \frac{\left|\tilde{S}_B\right|}{\left|\tilde{S}_W\right|} = \arg\max_{\mathbf{W}} \frac{\left|W^T S_B W\right|}{\left|W^T S_W W\right|}$$

Mathematical Formulation

$$W_{opt} = \arg\max_{W} \frac{\left|W^T S_B W\right|}{\left|W^T S_W W\right|}$$

- Solve generalized eigenvector problem:

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

Mathematical Formulation

- Solution: Generalized Eigenvectors

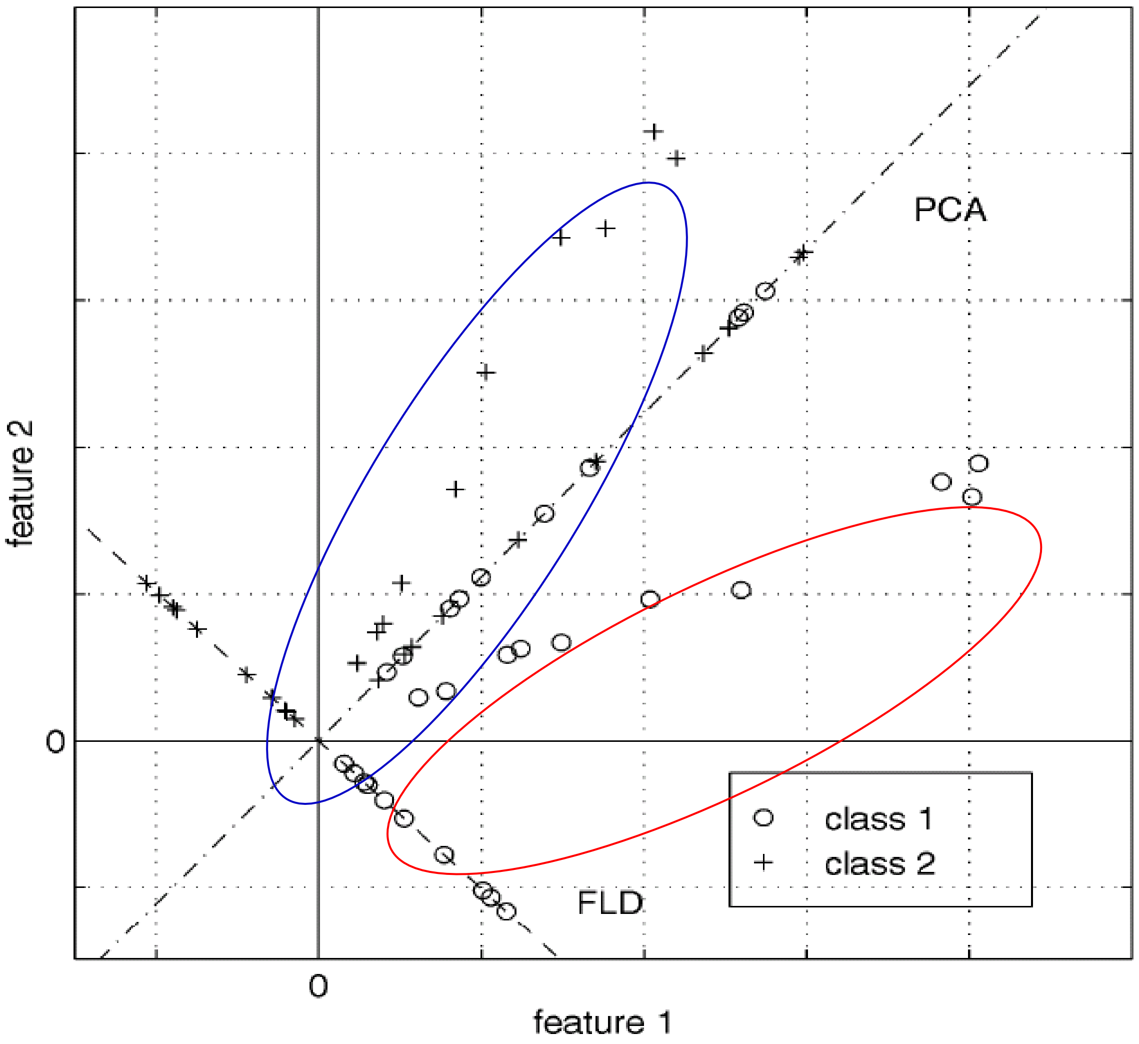
$$S_B w_i = \lambda_i S_W w_i \qquad i = 1, \ldots, m$$

- Rank of $\boldsymbol{W_{opt}}$ 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 maximise the **between class scatter**, while minimising the **within class scatter**.

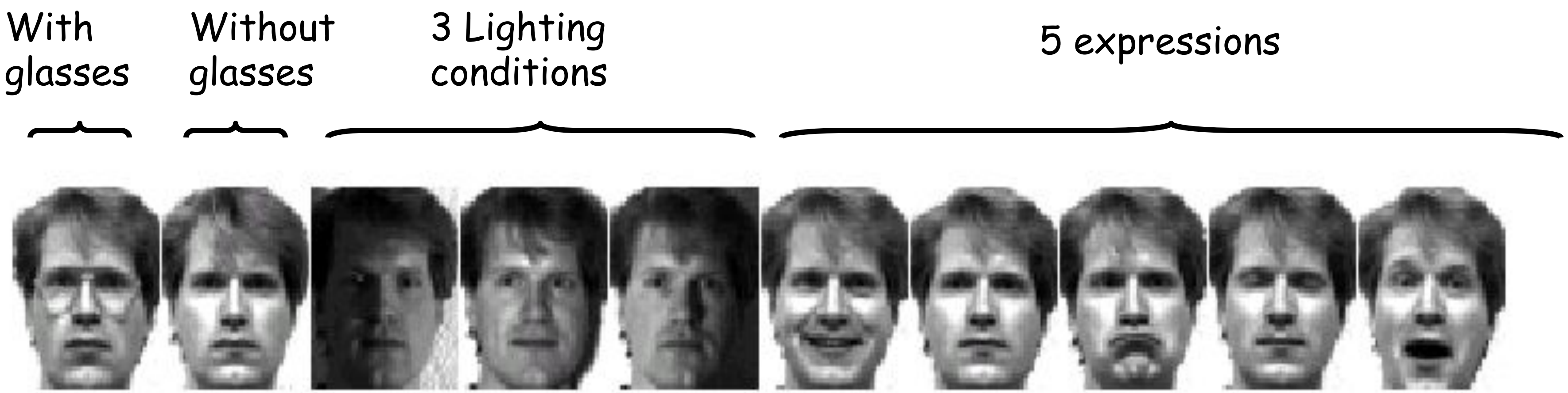
Basic intuition: PCA vs. LDA



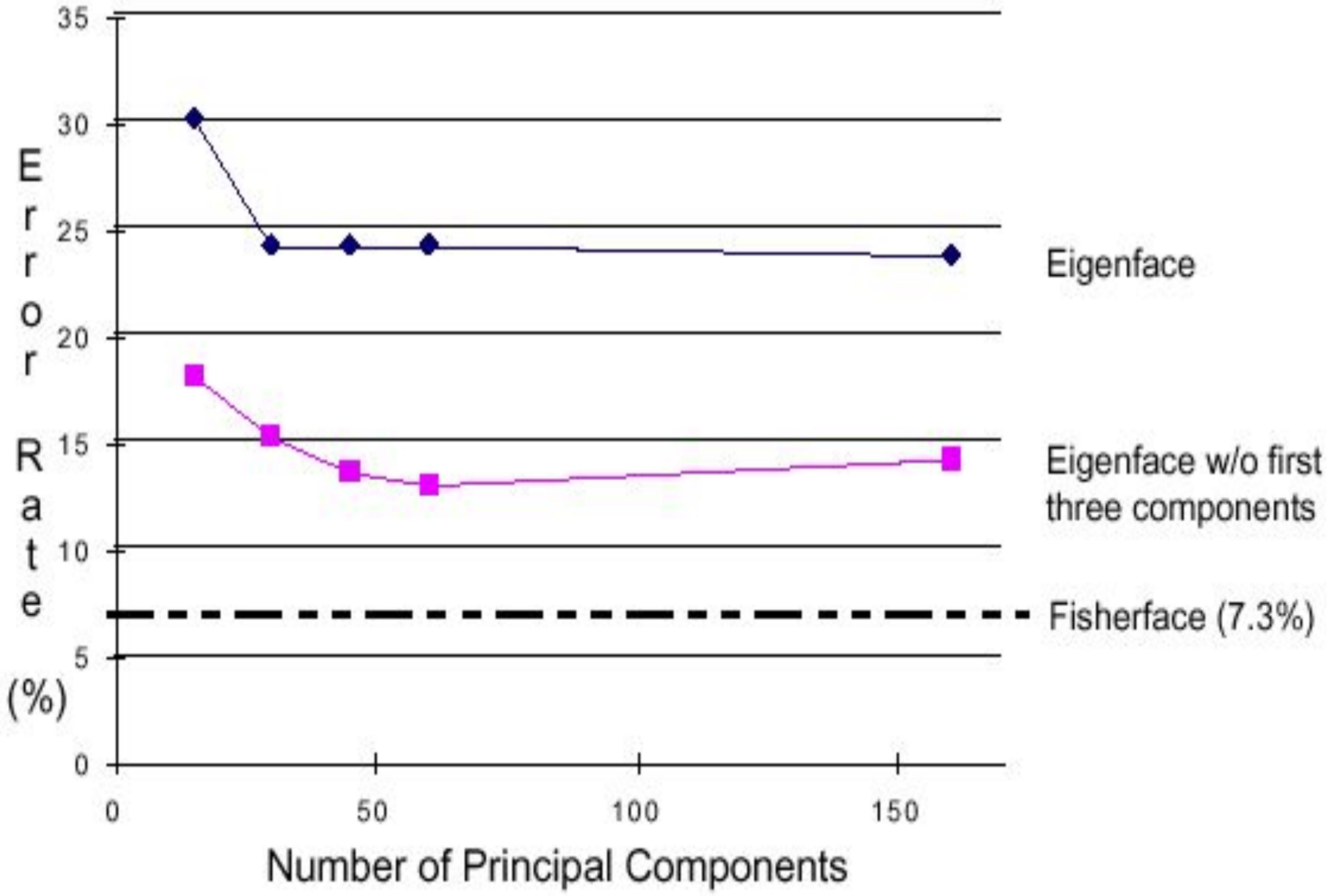
Results: Eigenface vs. Fisherface

- 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.