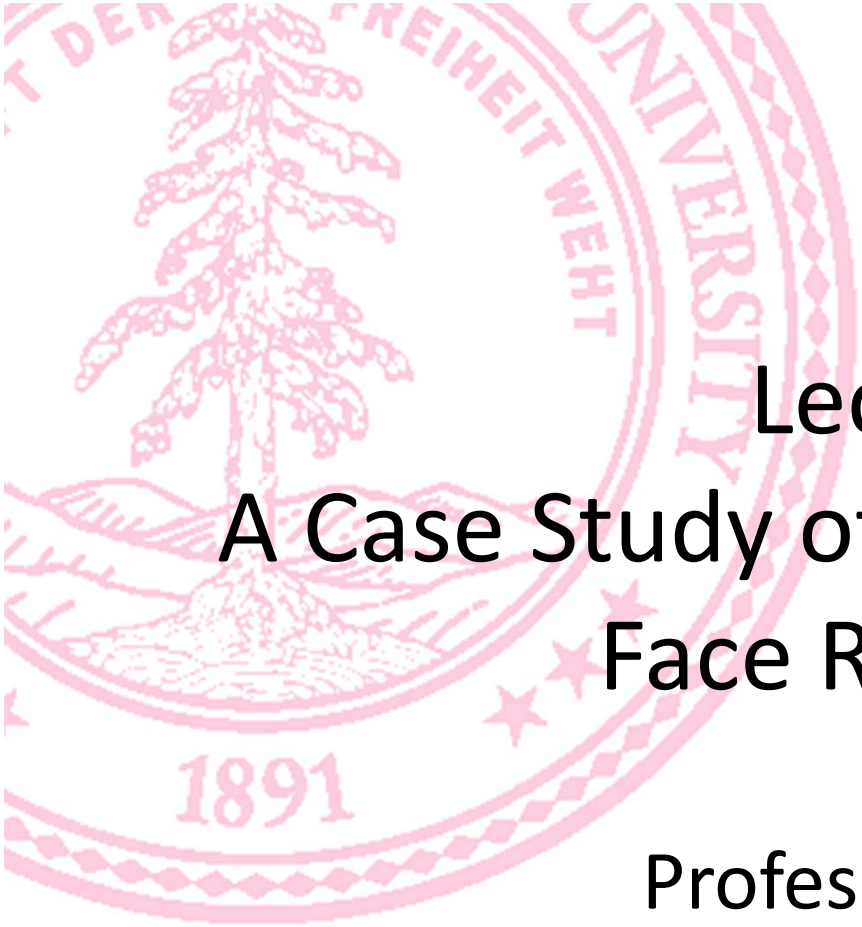


Lecture 2:
A Case Study of Computer Vision – Face Recognition

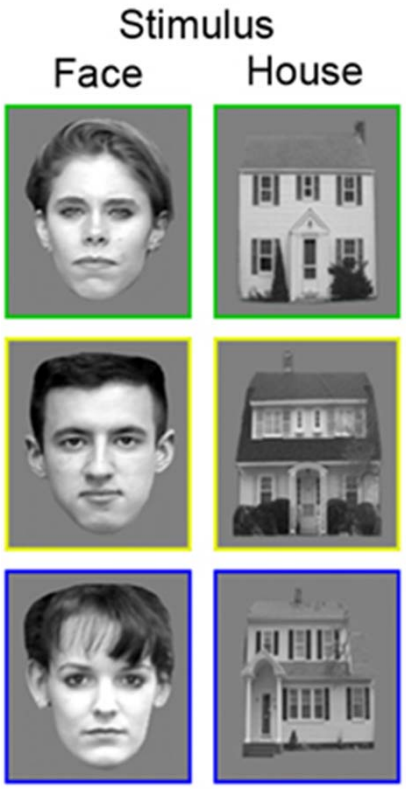
Professor Fei-Fei Li

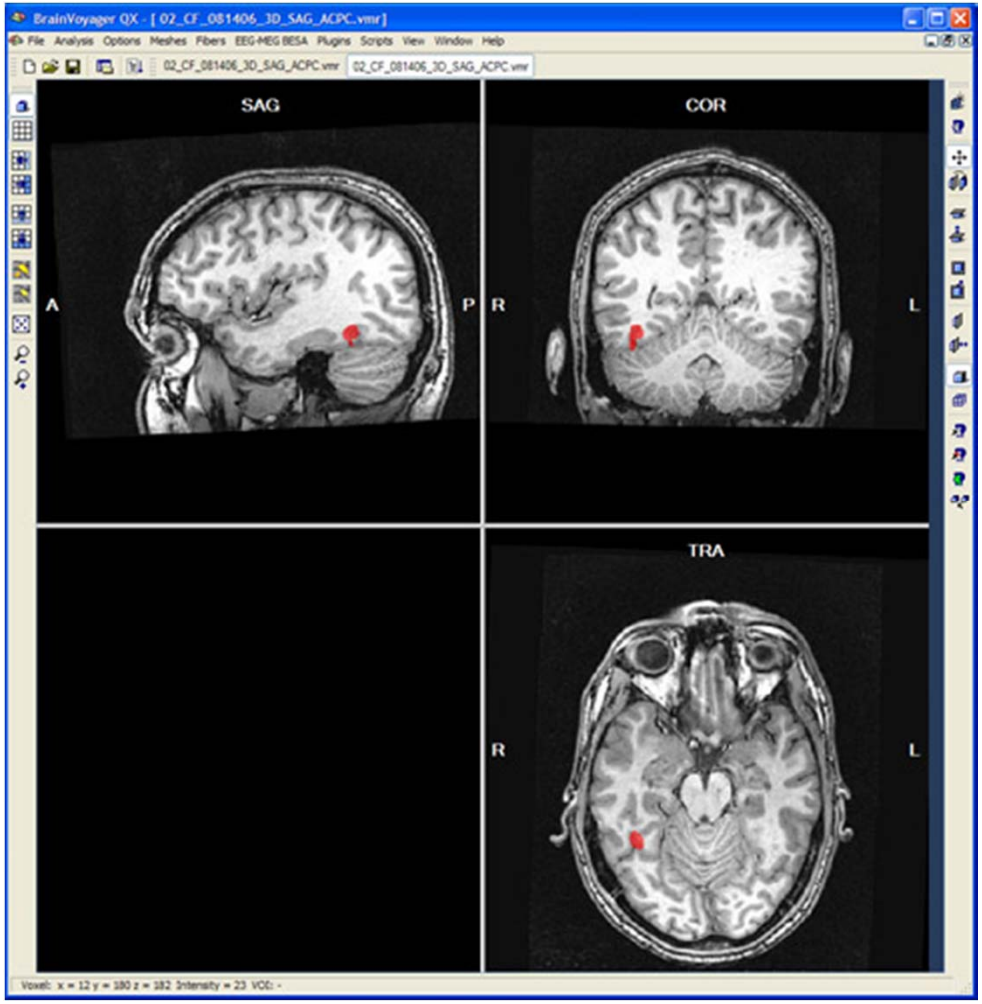
Stanford Vision Lab

What we will learn today

- Recognition problems
- Face discrimination
  - Principal Component Analysis (PCA) and Eigenfaces (Problem Set 1 (Q1))
  - Linear Discriminant Analysis (LDA) and Fisherfaces
- Face detection
  - SVM (Problem Set 0 (Q2))
  - Boosting

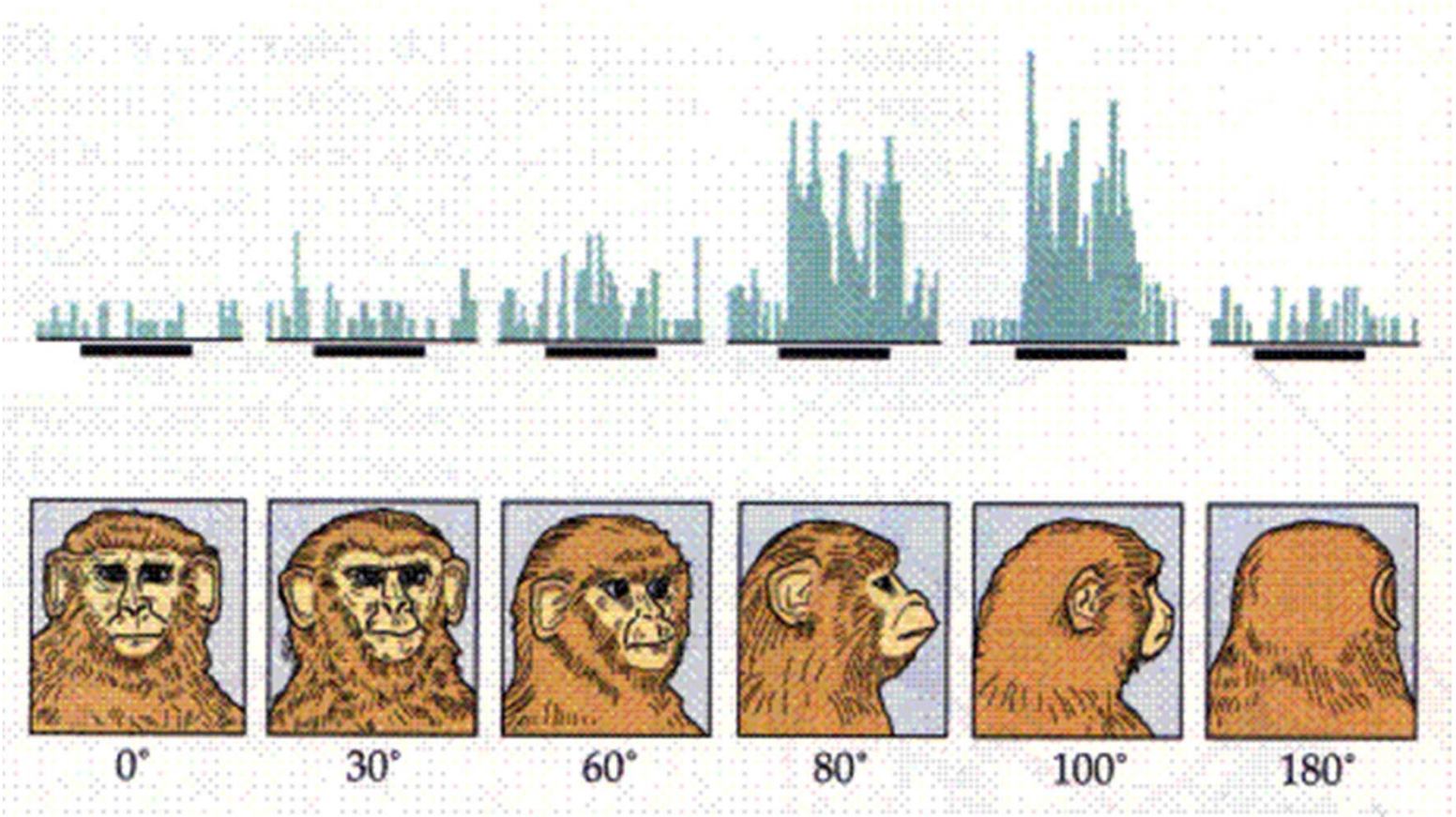
“Faces” in the brain





Kanwisher, et al. 1997

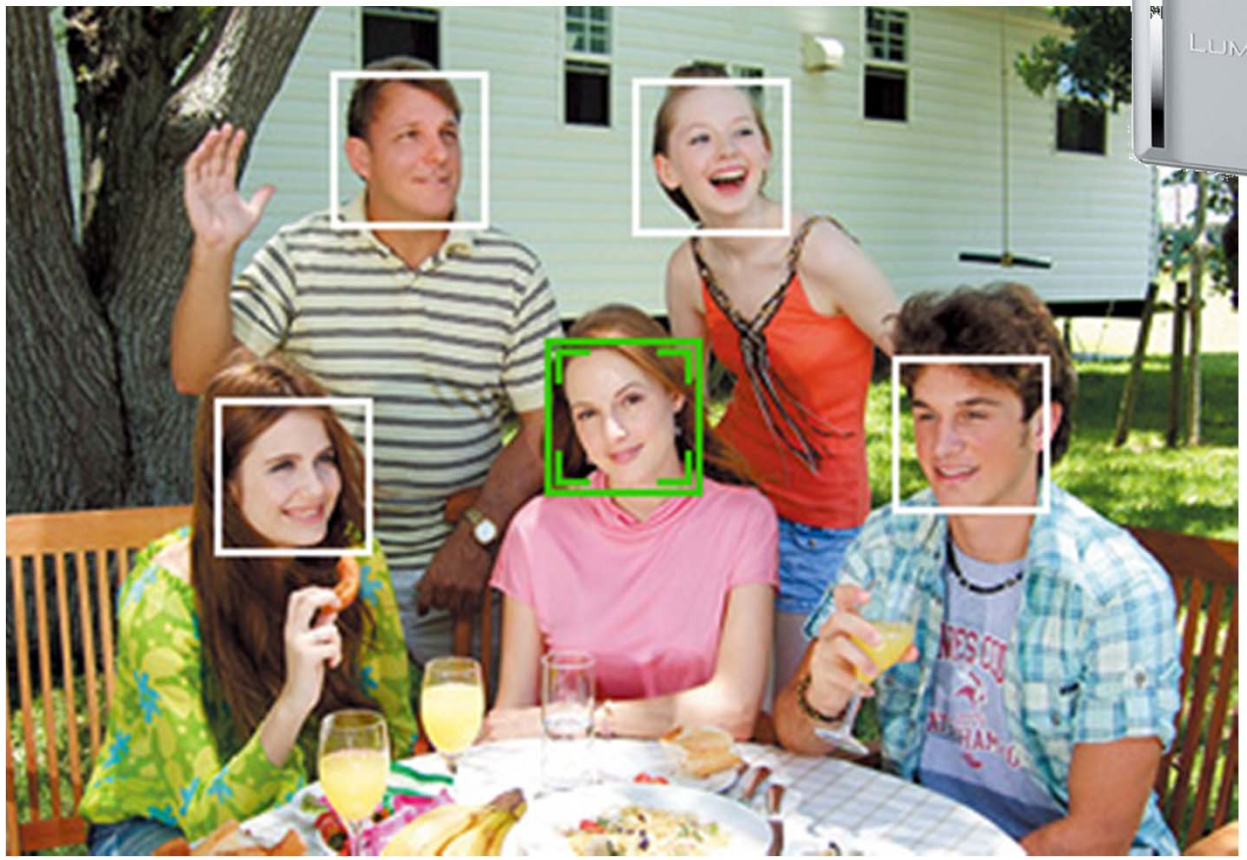
“Faces” in the brain



Courtesy of Johannes M. Zanker

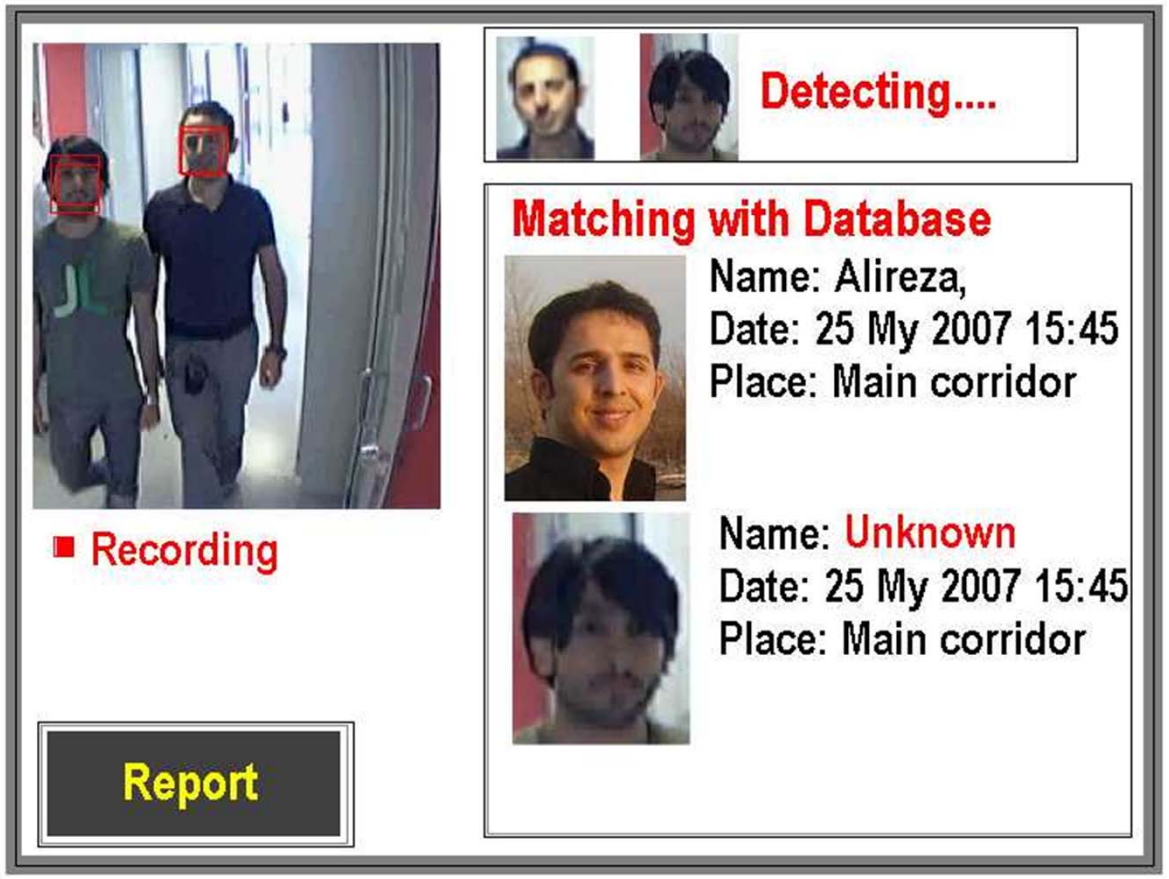
Face Recognition

- Digital photography



Face Recognition

- Digital photography
- Surveillance

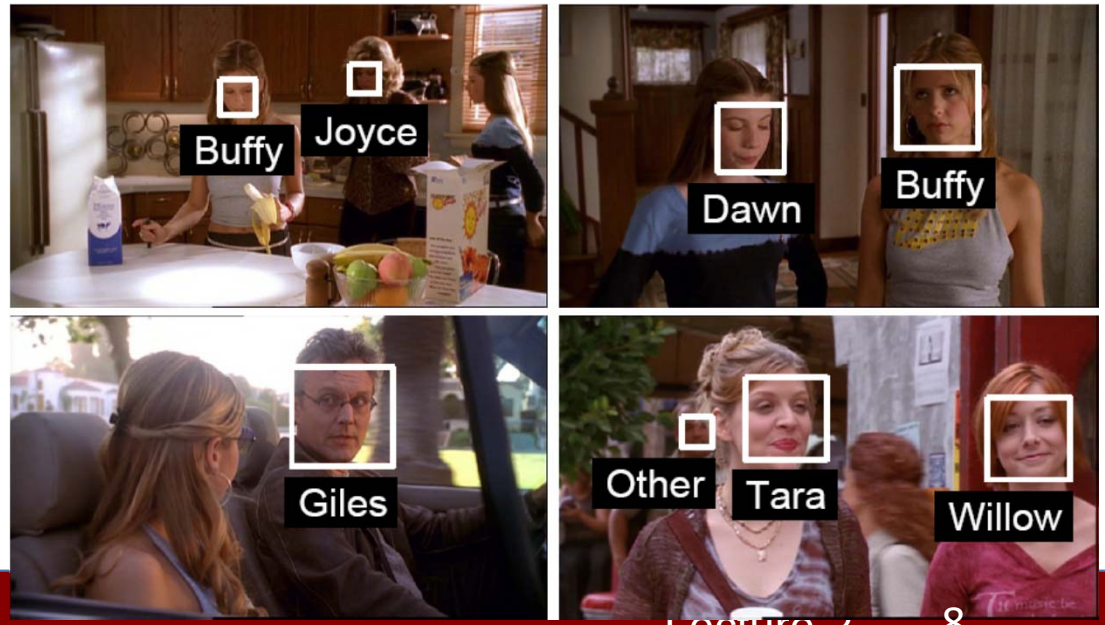


Face Recognition

- Digital photography
- Surveillance
- Album organization

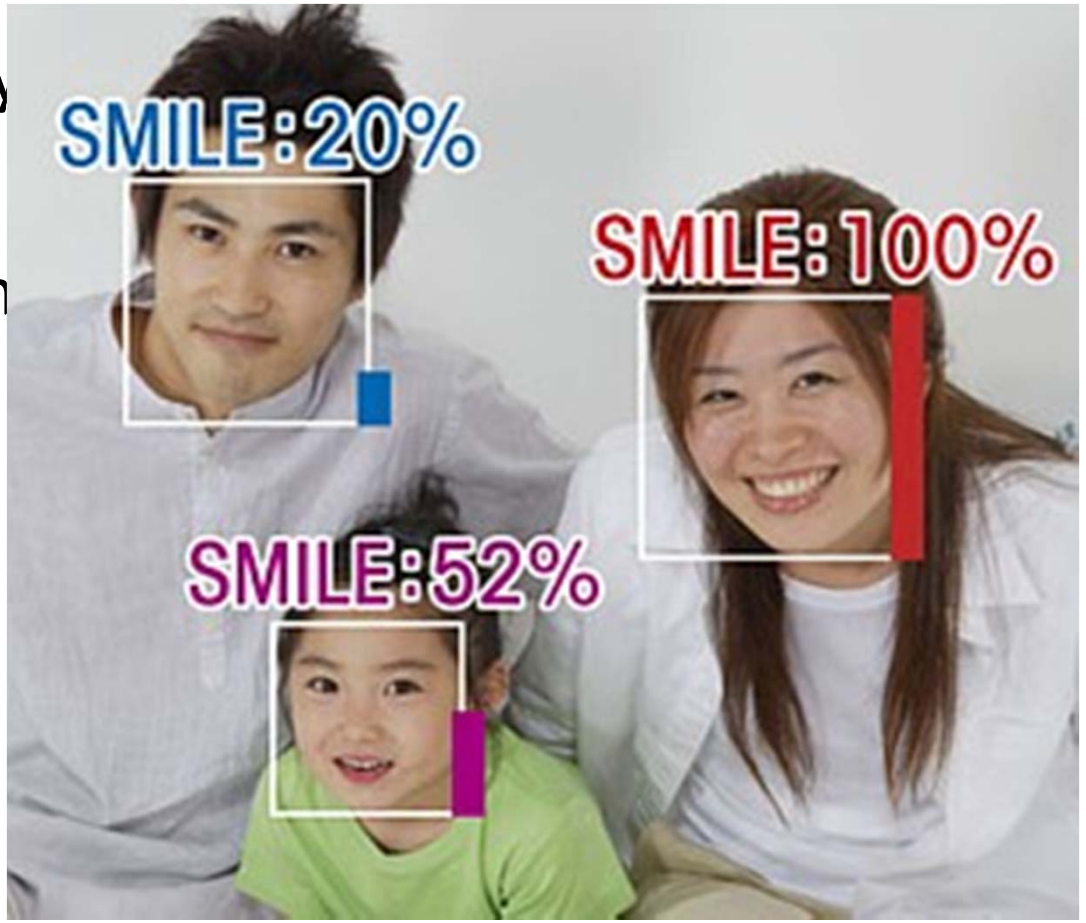
Face Recognition

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



Face Recognition

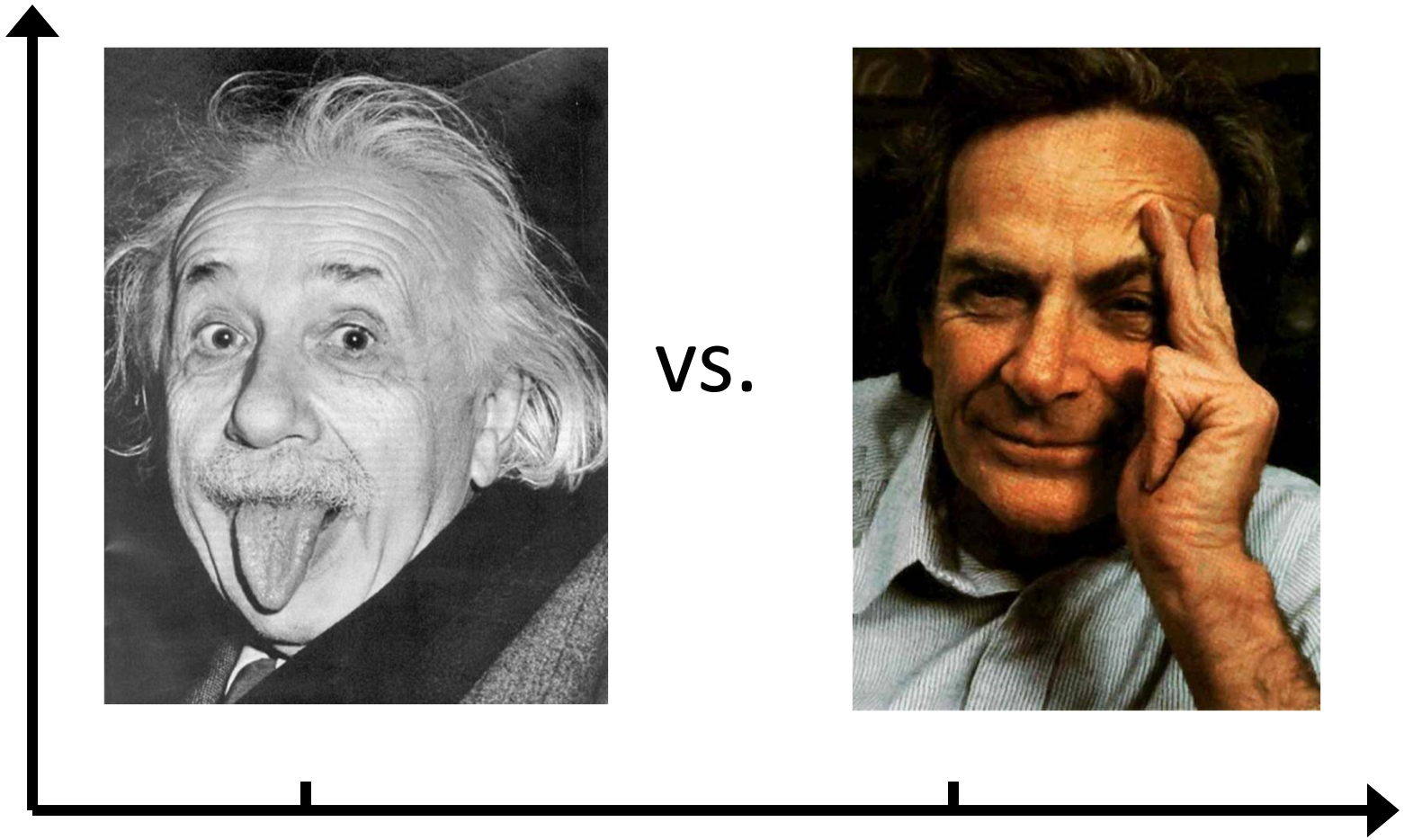
- Digital photography
- Surveillance
- Album organization
- Person tracking/id.
- Emotions and expressions



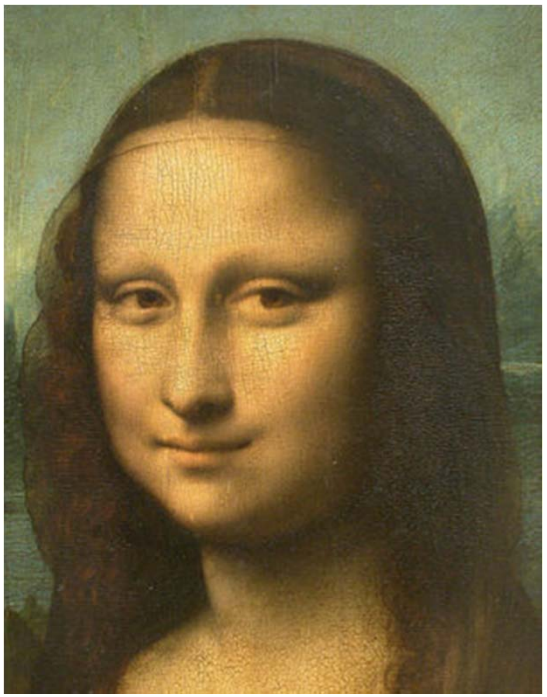
Face Recognition

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

What's 'recognition'?



What's 'recognition'?



vs.

Identification

Categorization

No localization

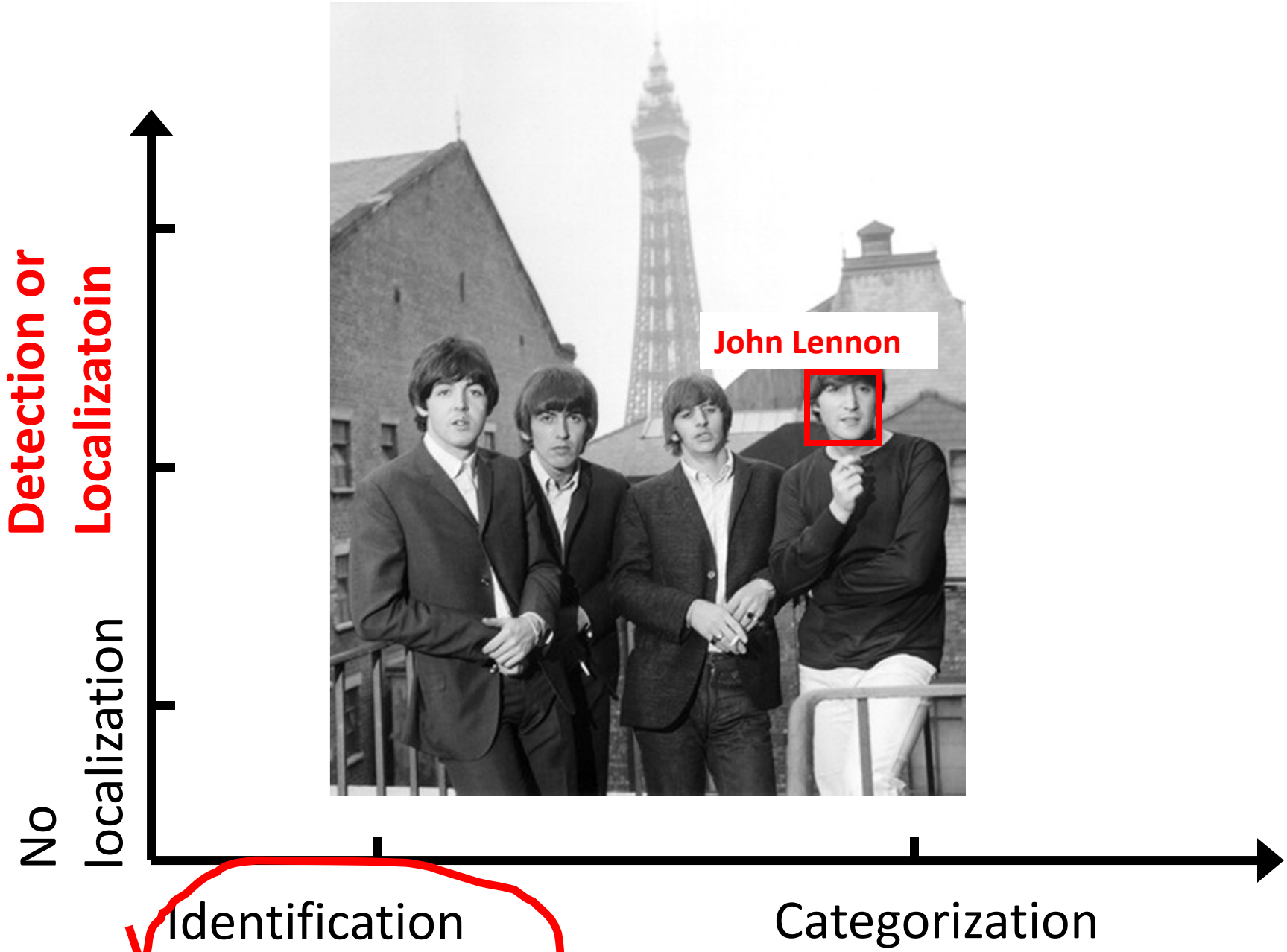
Identification

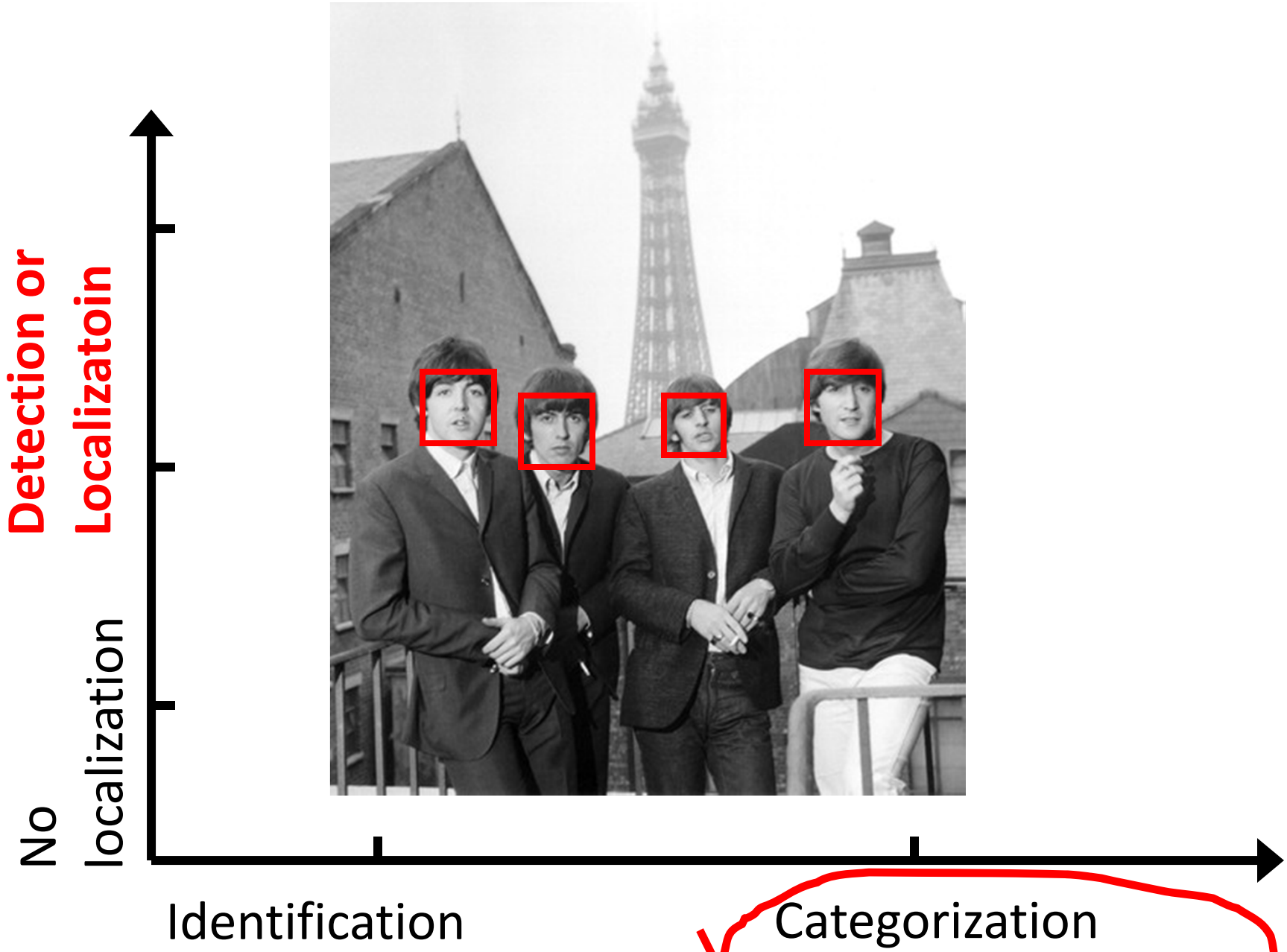
Categorization

No localization

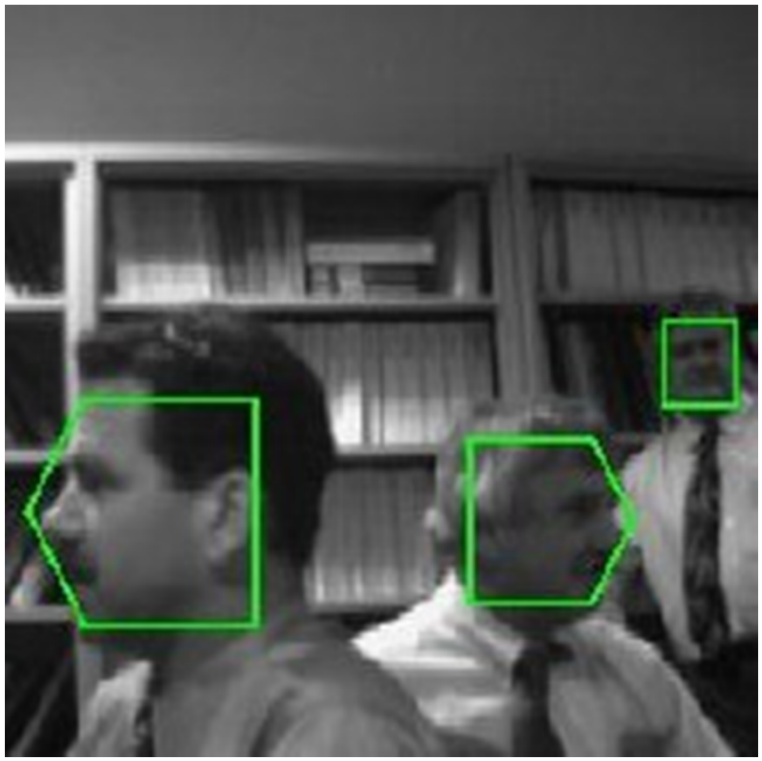
Identification

Categorization


Detection or
Localizatoin
No
localization
John Lennon
Identification
Categorization


Detection or
Localizatoin
No
localization
Identification
Categorization

Pose recognition



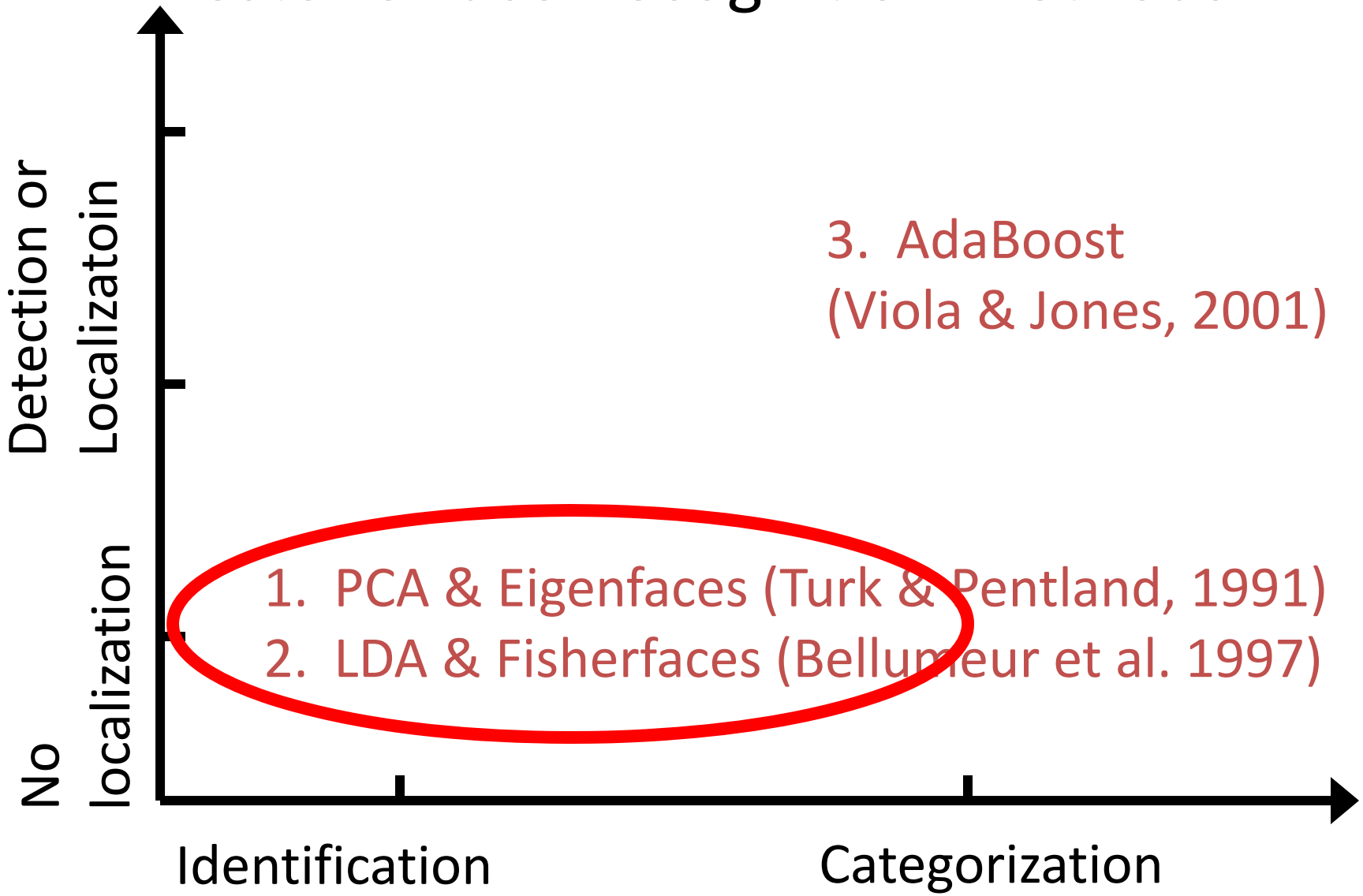
Detection or Localizatoin

No localization

Identification

Categorization

Milestone Face Recognition methods

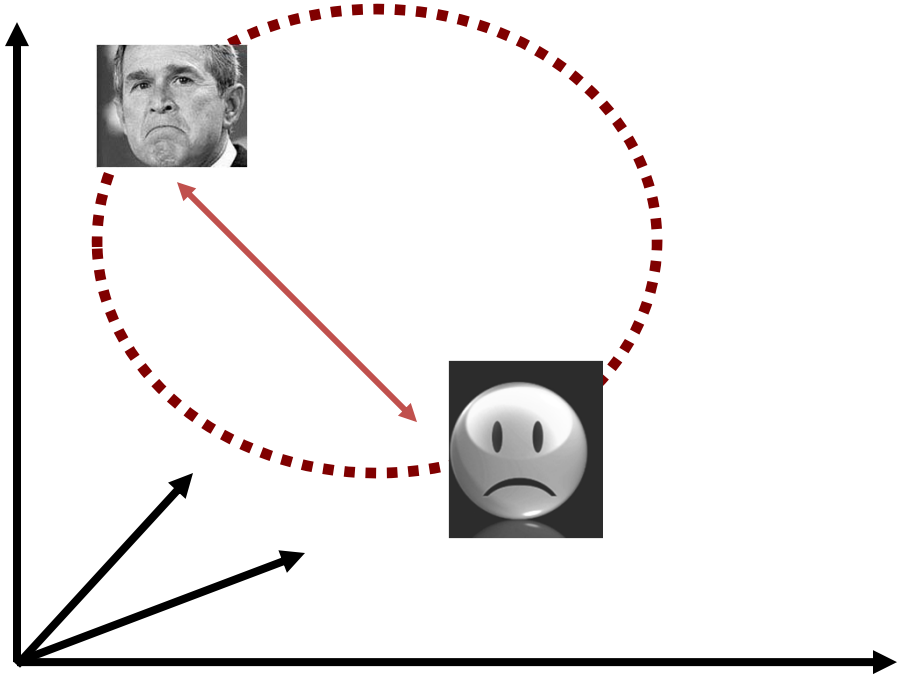


Eigenfaces and Fishfaces

- Principle Component Analysis (PCA)
- Linear Discriminant Analysis (LDA)

1. Turk and Pentland, Eigenfaces for Recognition, *Journal of Cognitive Neuroscience* **3** (1): 71–86.
2. 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.

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}

[Thanks to Chuck Dyer, Steve Seitz, Nishino]

Key Idea

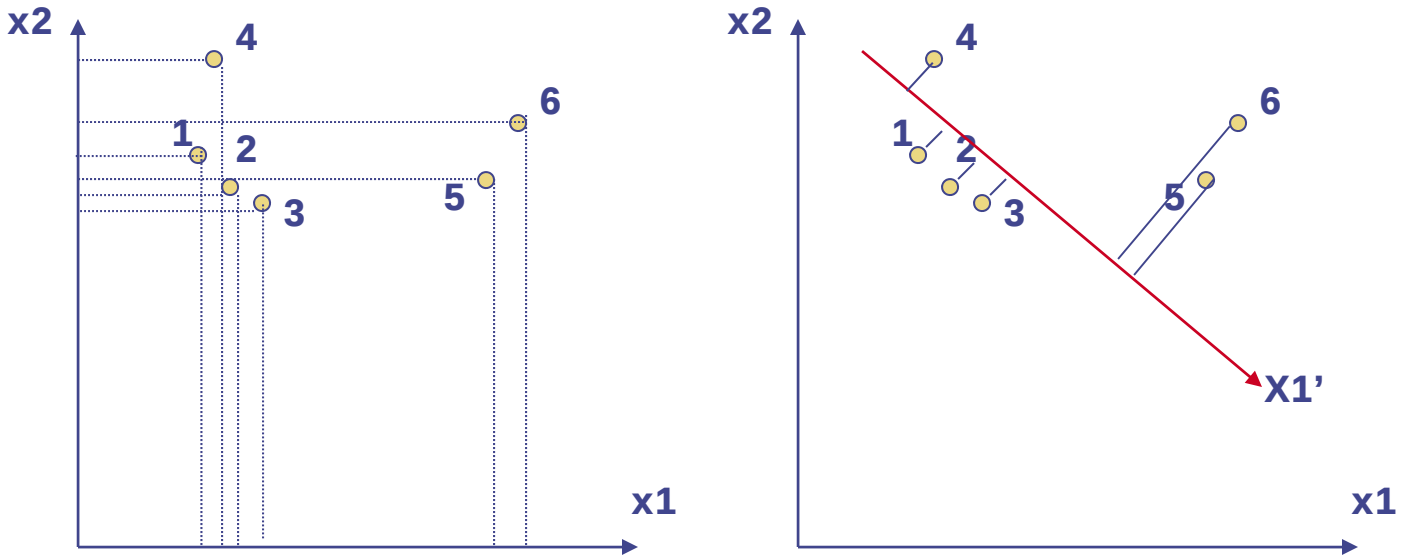
- Images in the possible set $\chi = \{\hat{x}\}$ are highly correlated.
- So, compress them to a low-dimensional subspace that captures key appearance characteristics of the visual DOFs.

- 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.

EIGENFACES: [Turk and Pentland 91]

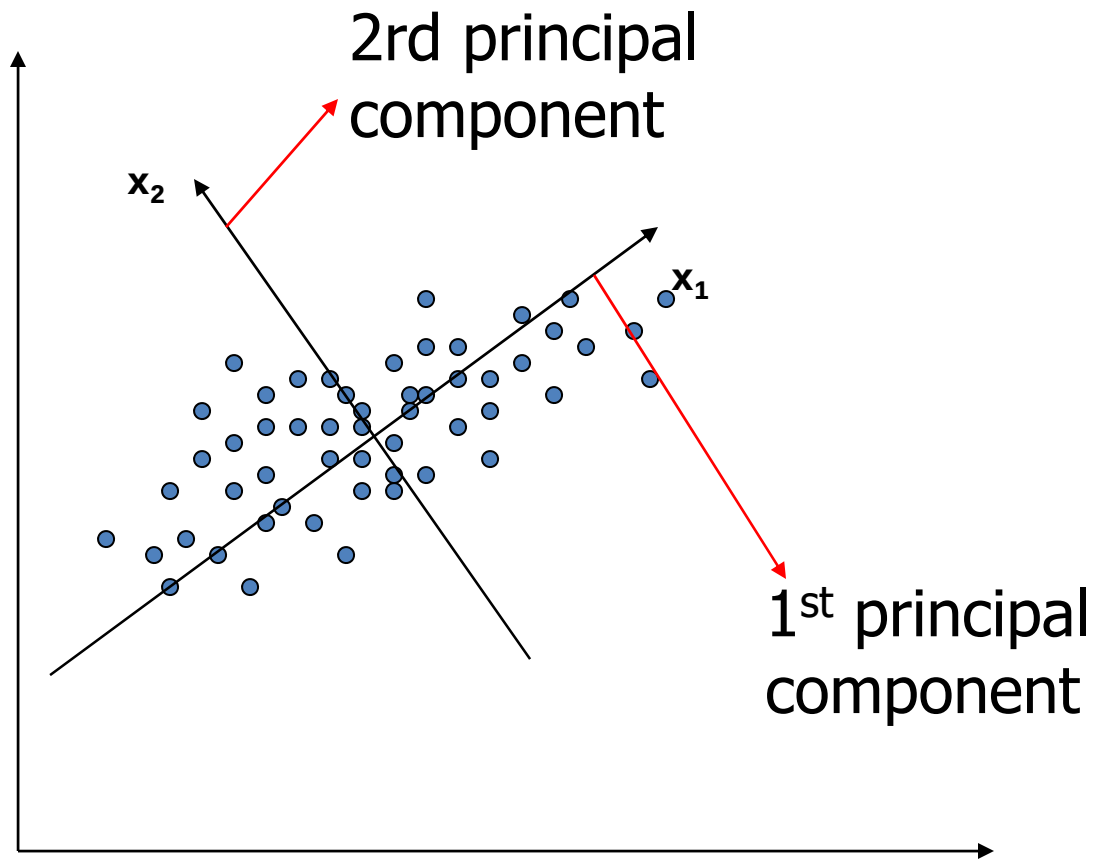
USE PCA for estimating the sub-space



PCA projection

• Computes n-dim subspace such that the projection of the data points onto the subspace has the largest variance among all n-dim subspaces.

USE PCA for estimating the sub-space



PCA Mathematical Formulation

PCA = eigenvalue decomposition of a data covariance matrix

Define a transformation, W,

$$y_j = W^T x_j \qquad j = 1, 2 \ldots N$$

m-dimensional | **Orthonormal** $W \in \mathbb{R}^{n \times m}$ | **n-dimensional**

$$S_T = \sum_{j=1}^{N} (x_j - \bar{x})(x_j - \bar{x})^T$$ = Data Scatter matrix

$$\tilde{S}_T = \sum_{j=1}^{N} (y_j - \bar{y})(y_j - \bar{y})^T = W^T S_T W$$ = Transf. data scatter matrix

$$W_{opt} = \arg\max_W \left| W^T S_T W \right| = [w_1 \; w_2 \; \ldots \; w_m]$$

Eigenvectors of S_T

Image space

Face space

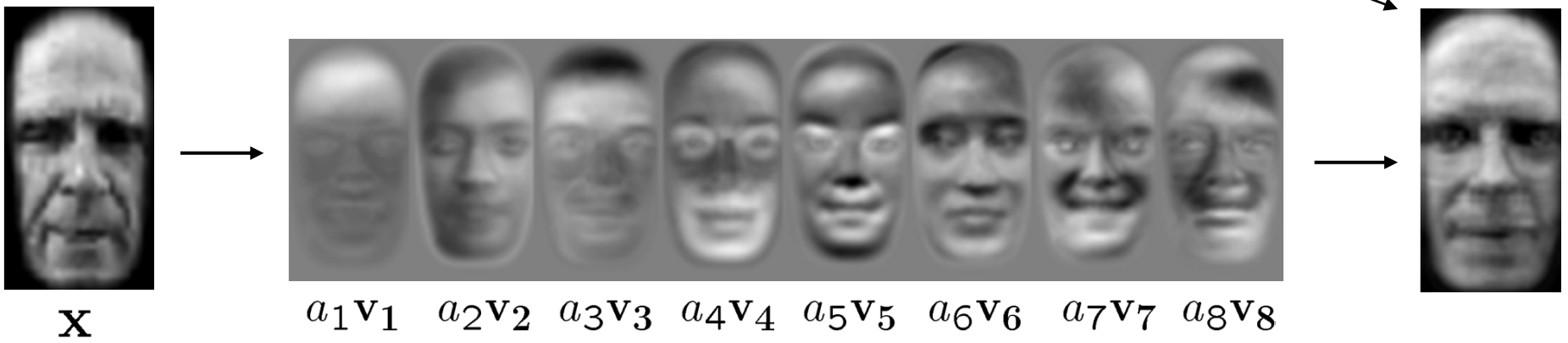
- Computes n-dim subspace such that the projection of the data points onto the subspace has the largest variance among all n-dim subspaces.
- Maximize the scatter of the training images in face space

Projecting onto the Eigenfaces

- The eigenfaces $\mathbf{v_1}$, ..., $\mathbf{v_K}$ span the space of faces
  - A face is converted to eigenface coordinates by

$$\mathbf{x} \rightarrow (\underbrace{(\mathbf{x}-\overline{\mathbf{x}})\cdot\mathbf{v_1}}_{a_1}, \underbrace{(\mathbf{x}-\overline{\mathbf{x}})\cdot\mathbf{v_2}}_{a_2}, \ldots, \underbrace{(\mathbf{x}-\overline{\mathbf{x}})\cdot\mathbf{v_K}}_{a_K})$$

$$\mathbf{x} \approx \overline{\mathbf{x}} + a_1\mathbf{v_1} + a_2\mathbf{v_2} + \ldots + a_K\mathbf{v_K}$$



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 $u = 1/N \Sigma x_i$

3. Compute the difference image $\varphi_i = x_i - u$

Algorithm

4. Compute the covariance matrix (total scatter matrix)

$$S_T = (1/N) \Sigma \phi_i \phi_i^T = BB^T, \quad B=[\phi_1, \phi_2 \dots \phi_N]$$

5. Compute the eigenvectors of the covariance matrix S_T
6. Compute training projections a_1, a_2... a_N

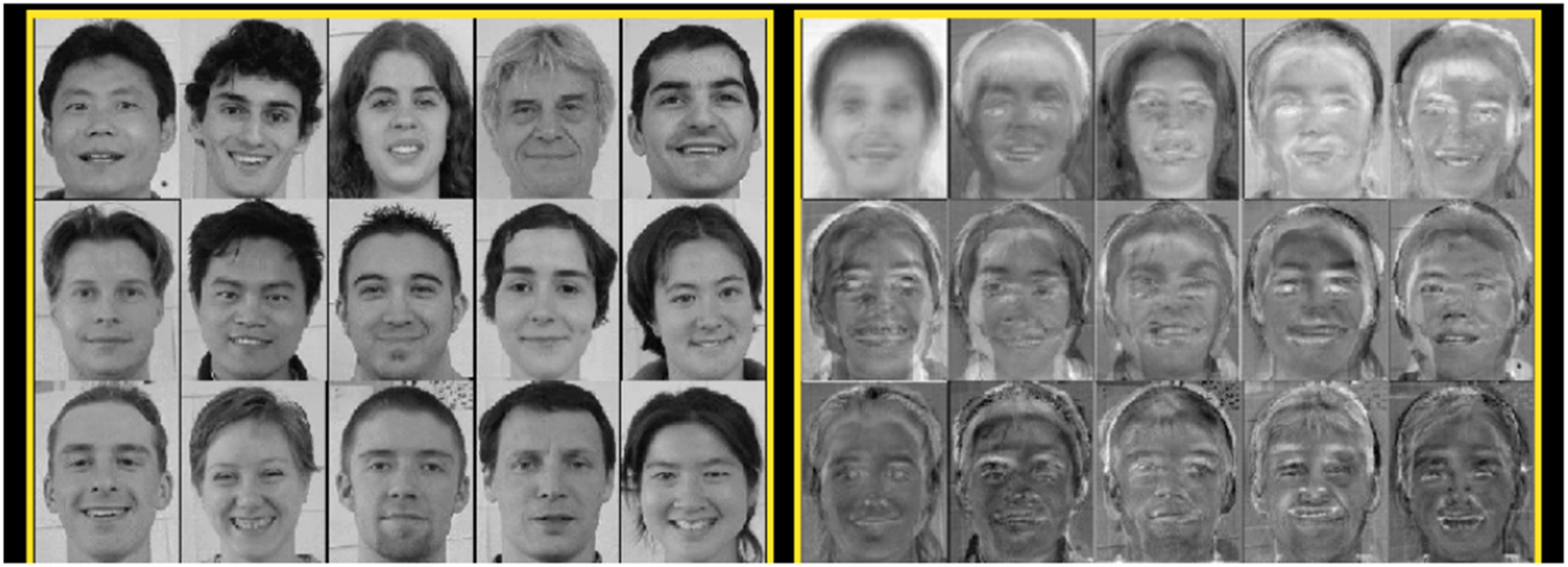
Testing

1. Take query image X
2. Project X into Eigenface space (W = {eigenfaces})
   and compute projection $\omega_i = W (X - u)$,
3. Compare projection ω_i with all training N projections a_i

Illustration of Eigenfaces

- The visualization of eigenvectors:

These are the first 4 eigenvectors from a training set of 400 images (ORL Face Database).



Eigenfaces look somewhat like generic faces.

Reconstruction and Errors

P = 4

P = 200

P = 400

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

Summary for Eigenface

Pros

- Non-iterative, globally optimal solution

Limitations

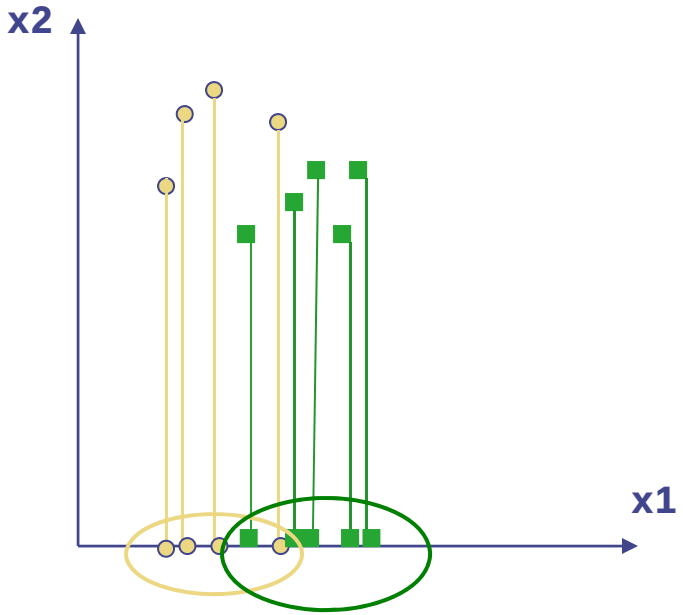
•PCA projection is **optimal for reconstruction** from a low dimensional basis, but **may NOT be optimal for discrimination…**

Linear Discriminant Analysis (LDA)
Fisher's Linear Discriminant (FLD)

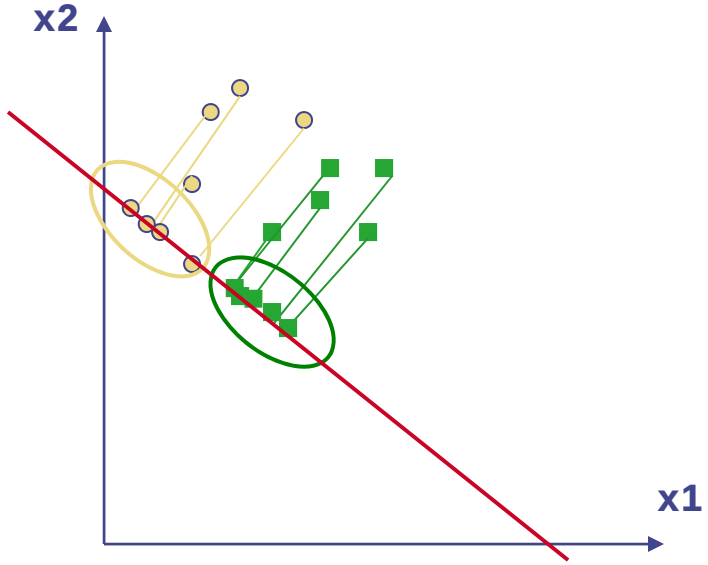
- Eigenfaces exploit the max scatter of the training images in face space
- Fisherfaces attempt to maximise the **between class scatter**, while minimising the **within class scatter**.

Illustration of the Projection

- Using two classes as example:

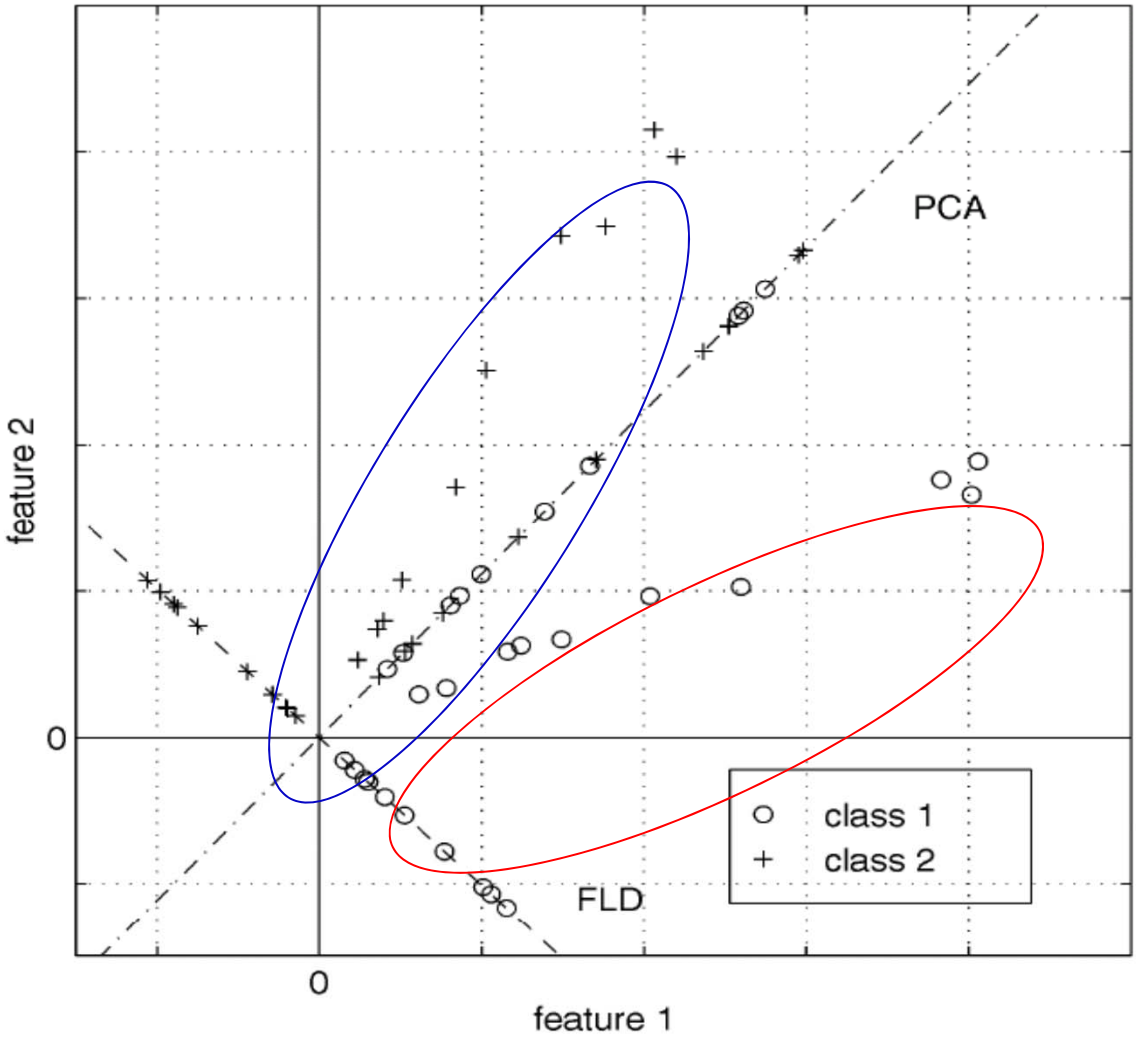


Poor Projection



Good

Comparing with PCA

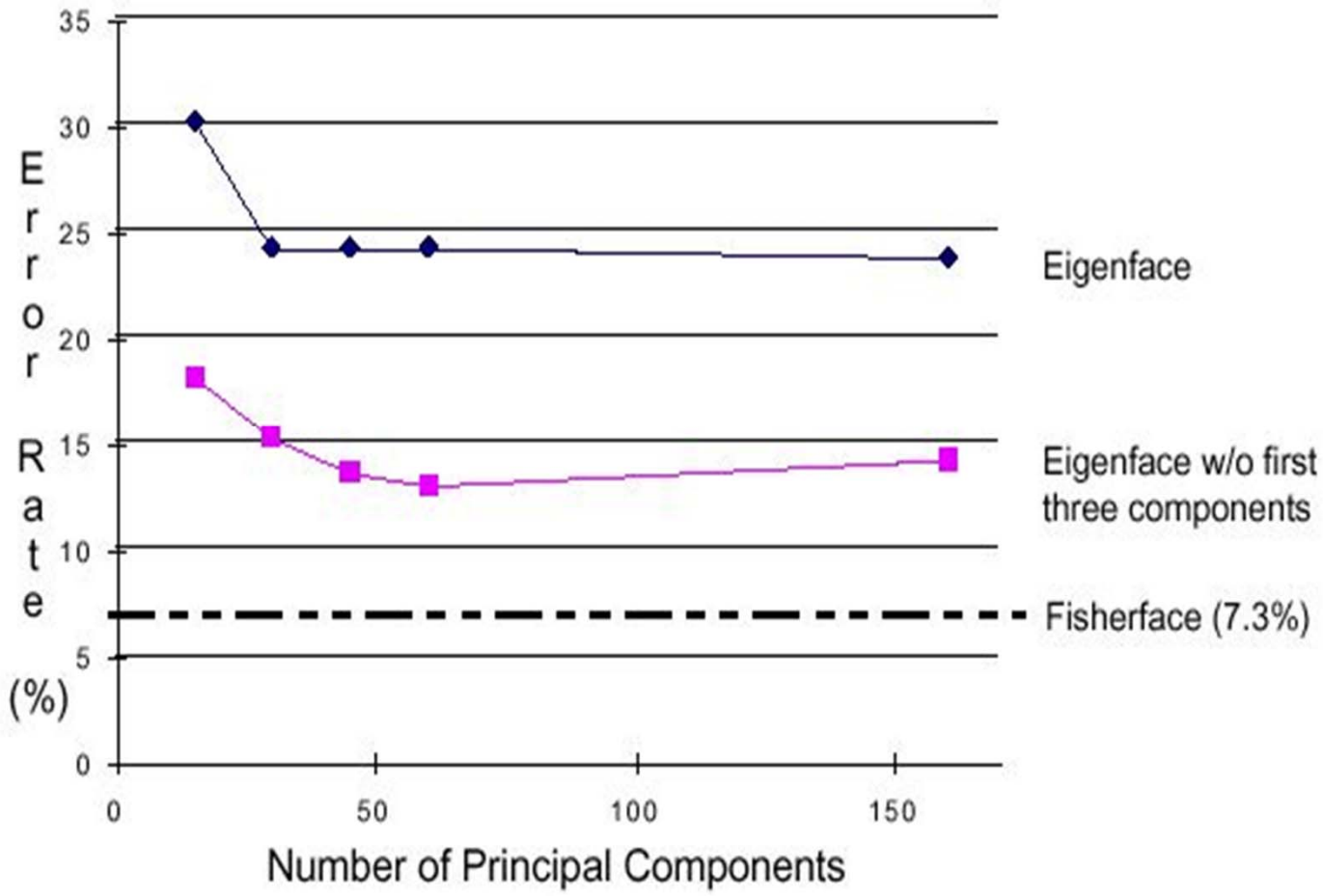


Results: Eigenface vs. Fisherface (1)

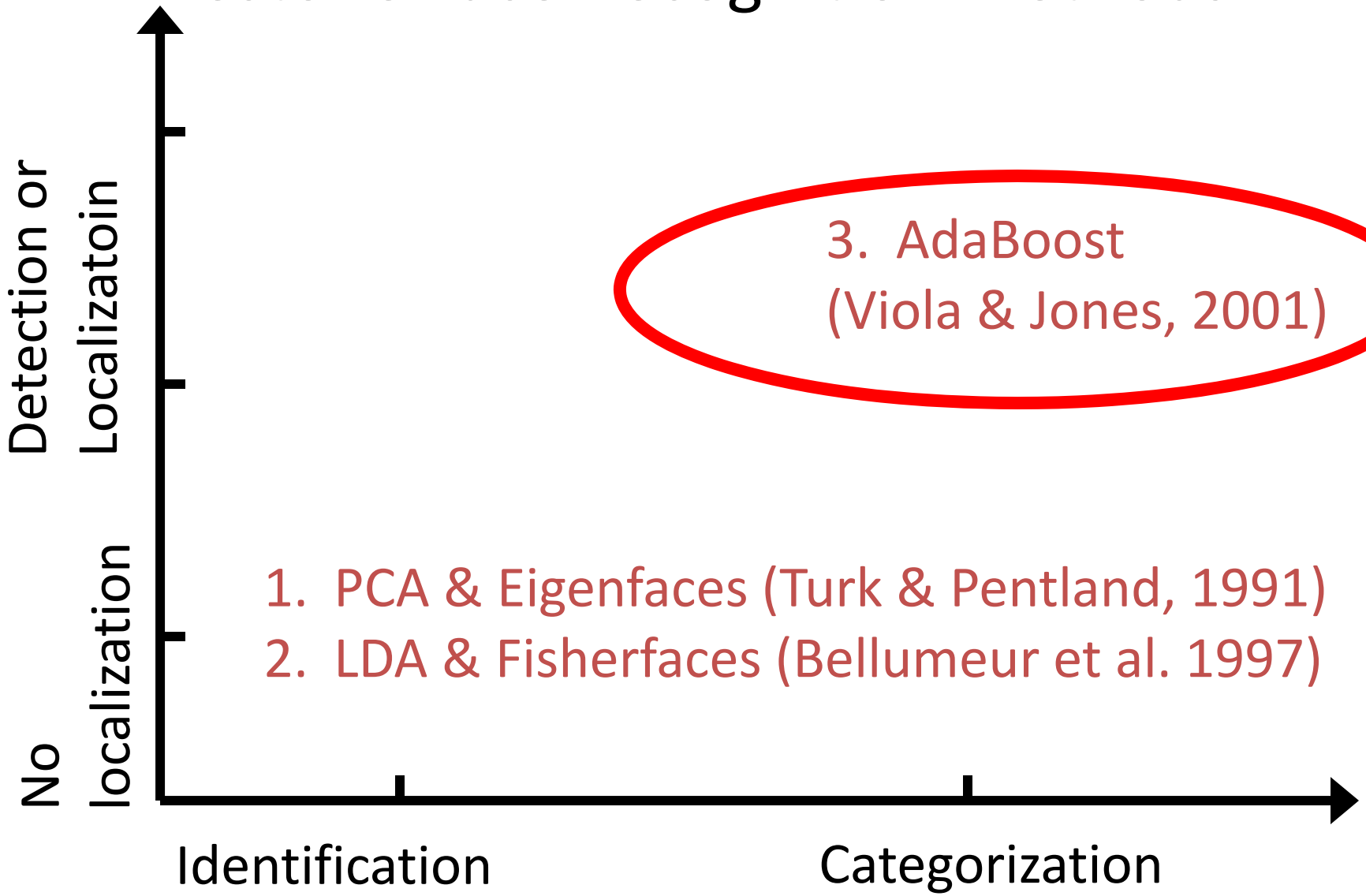
- Input: 160 images of 16 people
- Train: 159 images
- Test: 1 image

- Variation in Facial Expression, Eyewear, and Lighting

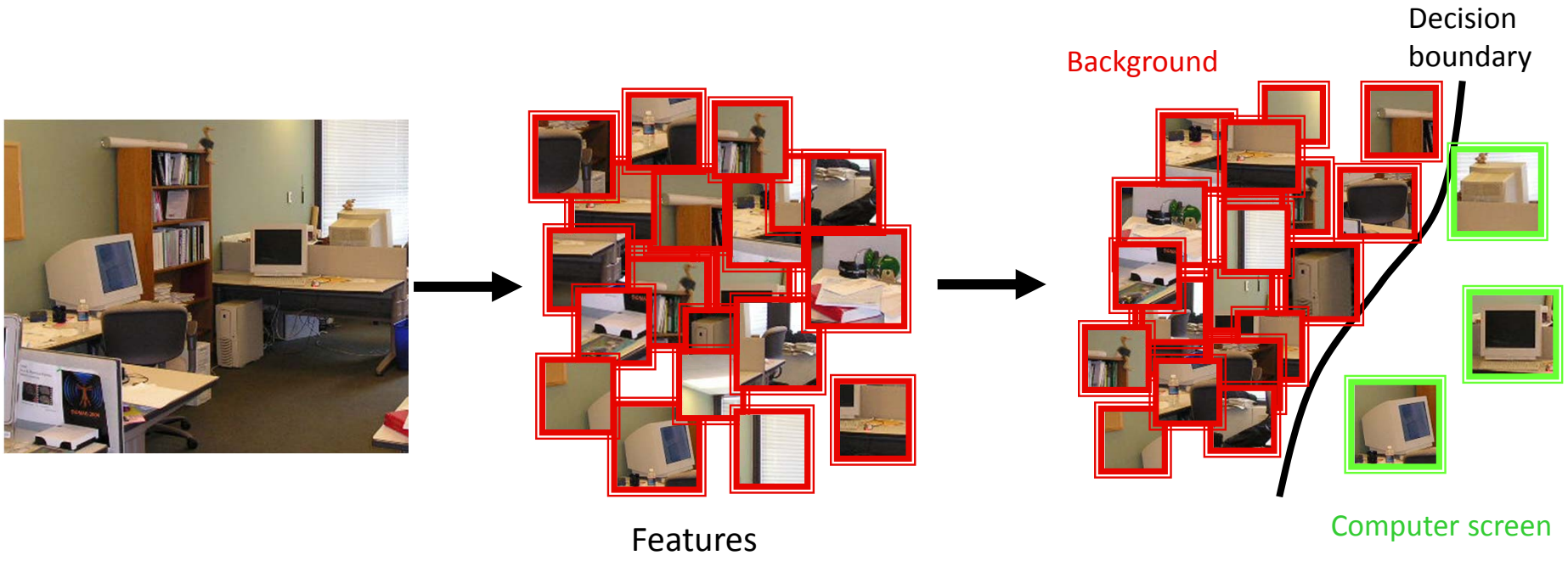
Eigenface vs. Fisherface (2)



Milestone Face Recognition methods

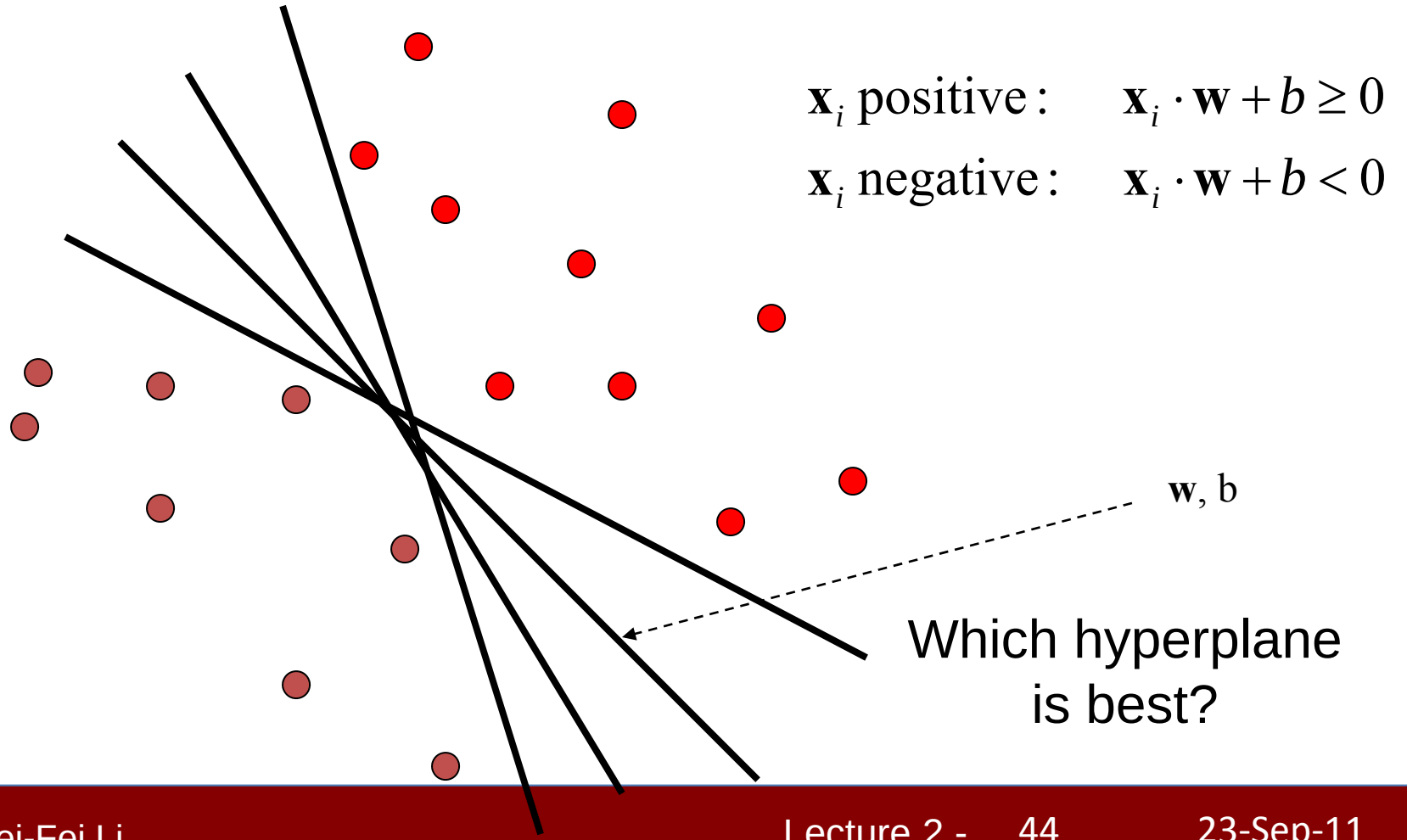


Detecting foreground objects: A binary classification formulation



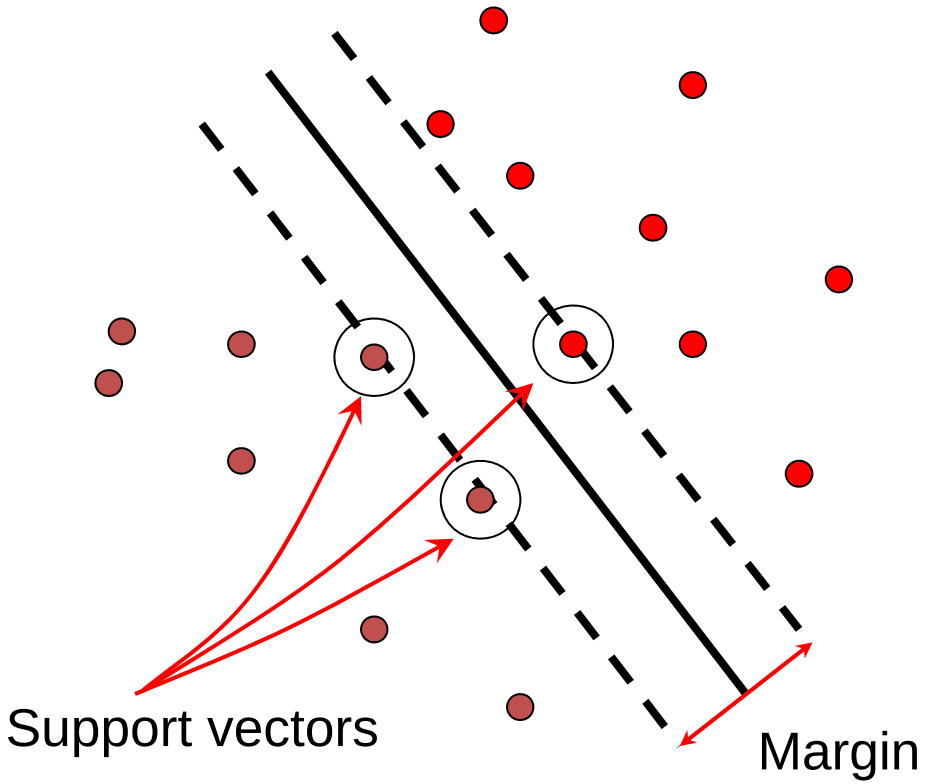
Linear classifiers

- Find linear function (*hyperplane*) to separate positive and negative examples



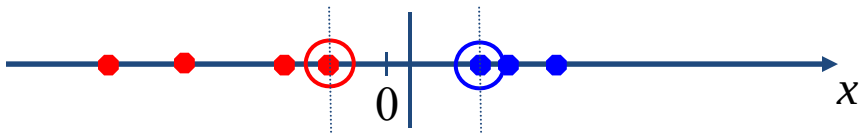
Support vector machines

- Find hyperplane that maximizes the *margin* between the positive and negative examples



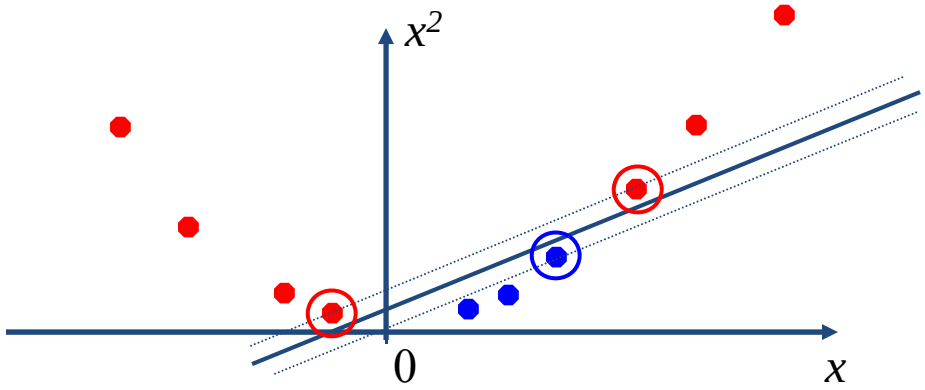
Nonlinear SVMs

- Datasets that are linearly separable work out great:



- But what if the dataset is just too hard?

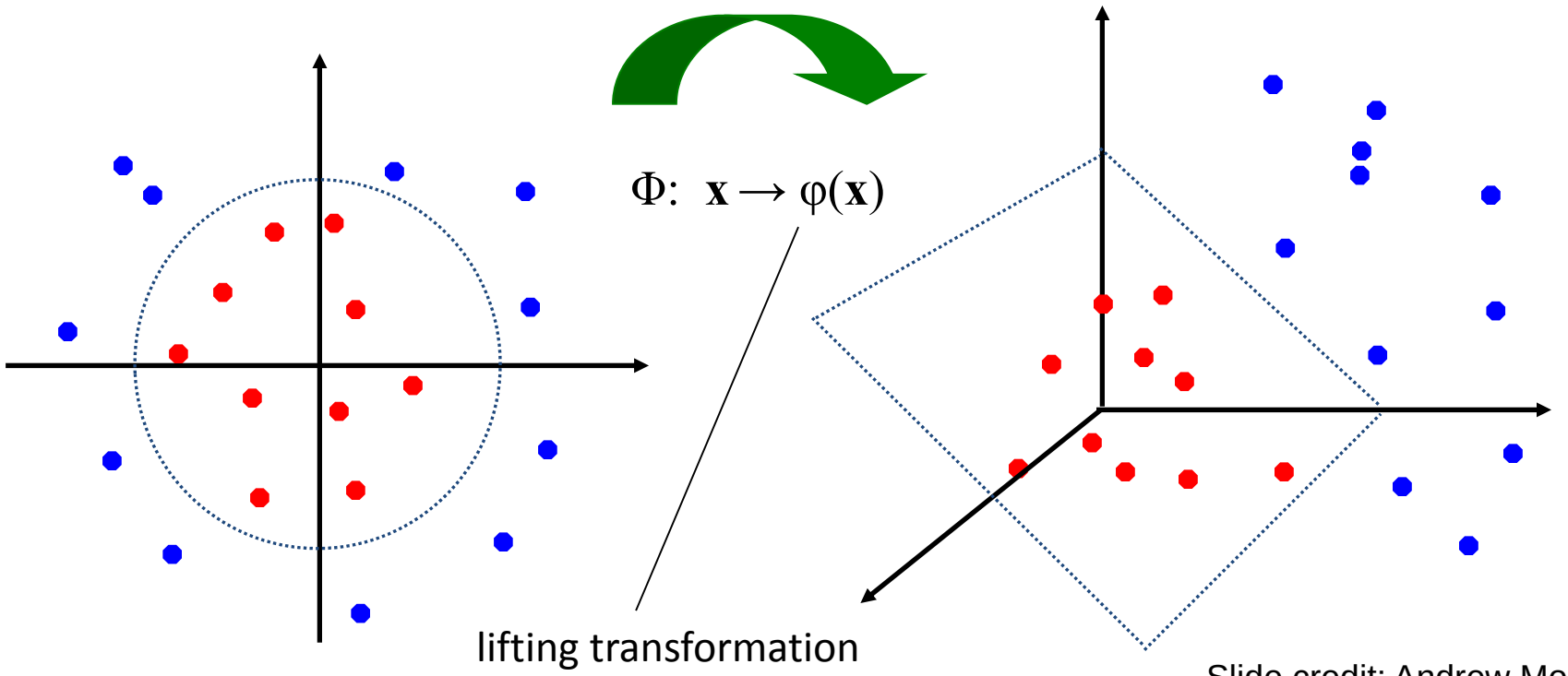
- We can map it to a higher-dimensional space:



Slide credit: Andrew Moore

Nonlinear SVMs

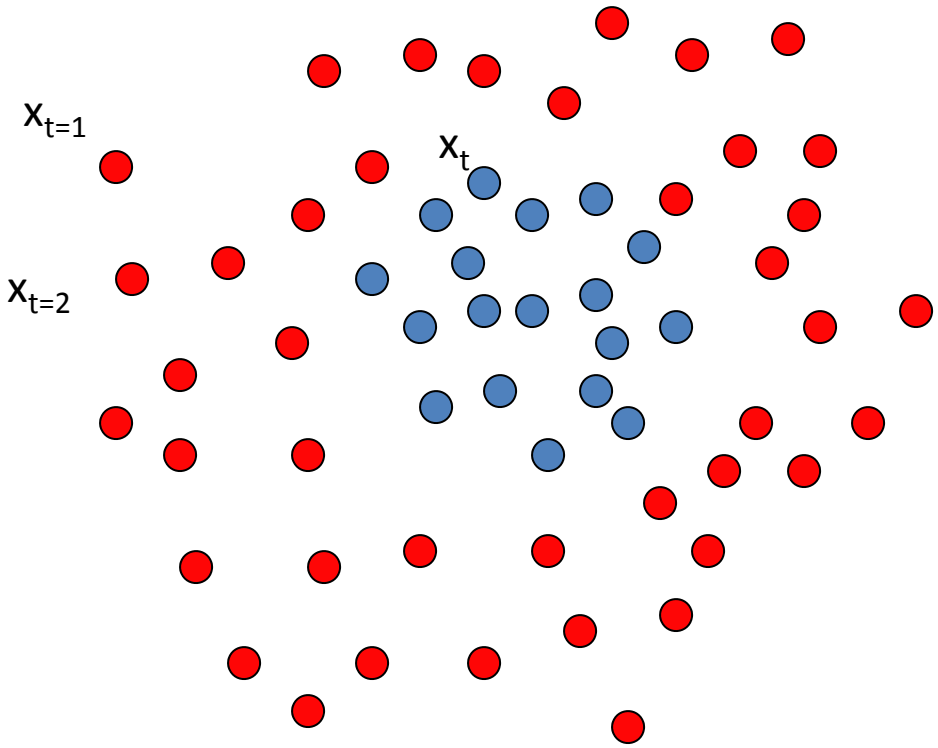
- General idea: the original input space can always be mapped to some higher-dimensional feature space where the training set is separable:



Slide credit: Andrew Moore

Boosting

Y. Freund and R. Schapire, A short introduction to boosting, *Journal of Japanese Society for Artificial Intelligence*, 14(5):771-780, September, 1999.



Each data point has

a class label:

$$y_t = \begin{cases} +1 & (\\ -1 & (\end{cases}$$

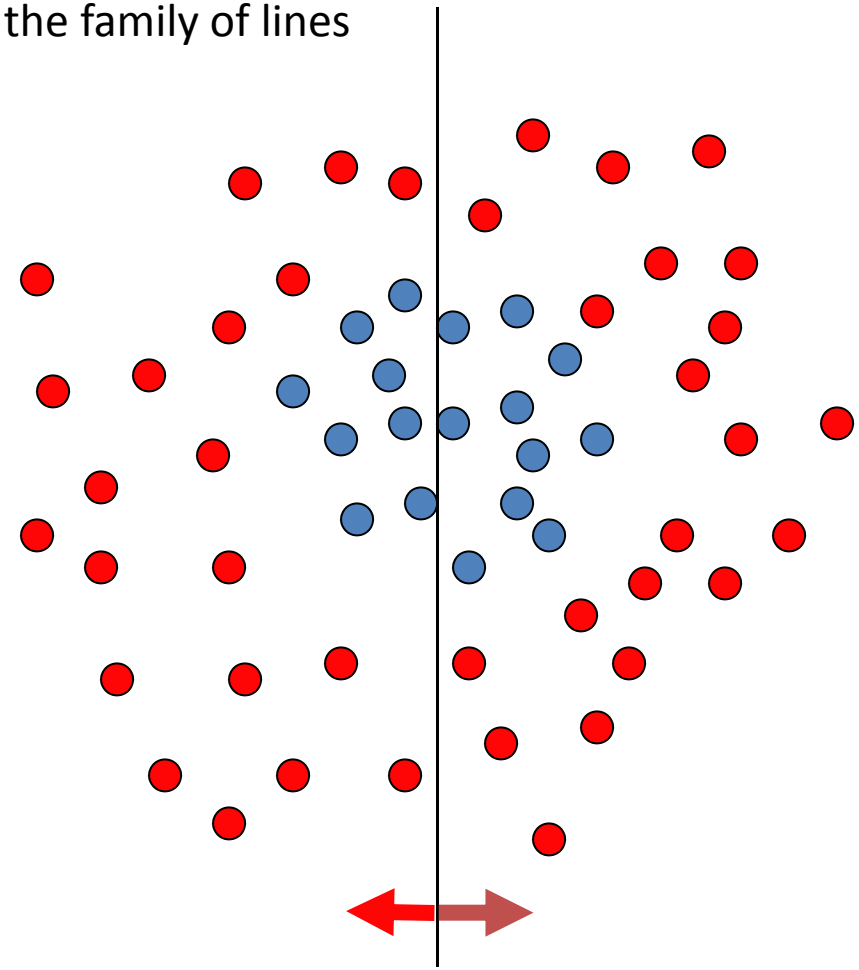
and a weight:

$w_t = 1$

- It is a sequential procedure:

Toy example

Weak learners from the family of lines



Each data point has

a class label:

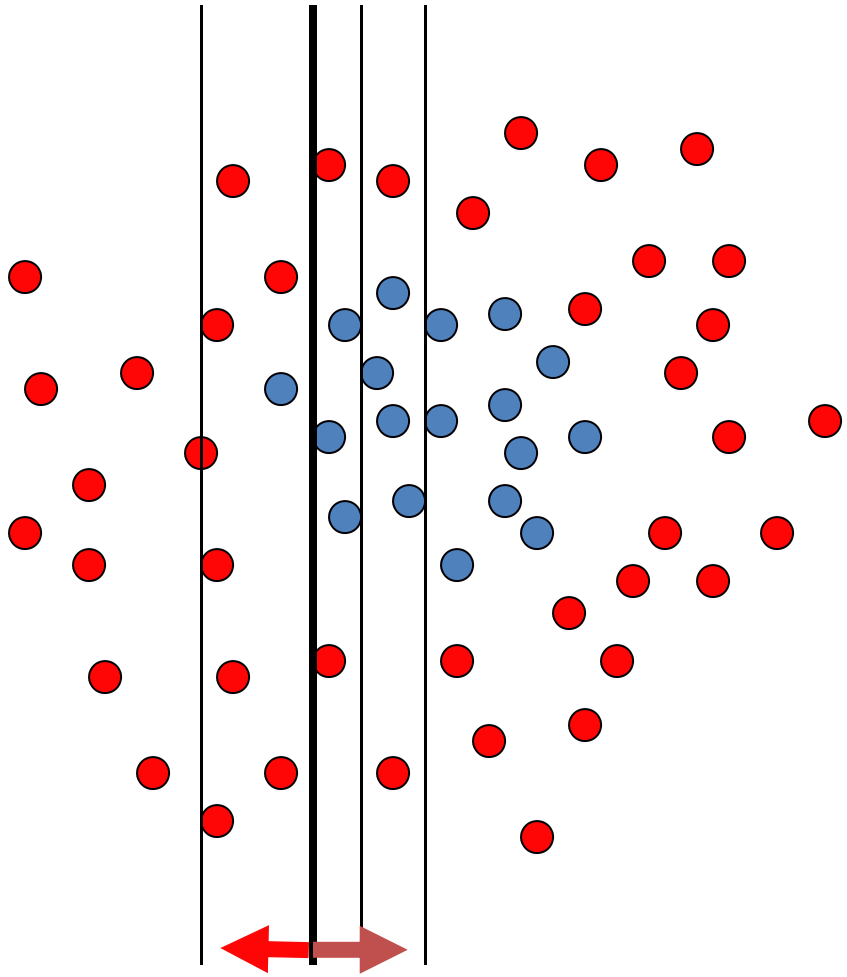
$$y_t = \begin{cases} +1 & (\\ -1 & (\end{cases}$$

and a weight:

$w_t = 1$

h => p(error) = 0.5 it is at chance

Toy example



This one seems to be the best

Each data point has

a class label:

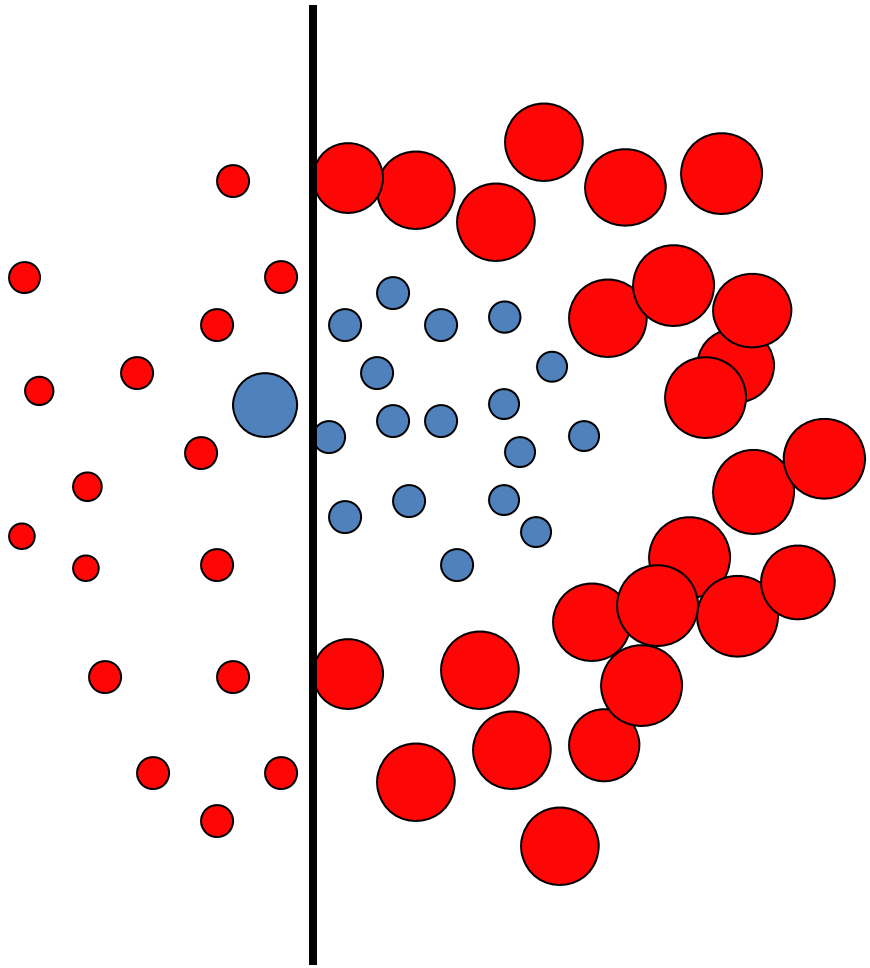
$$y_t = \begin{cases} +1 & (\\ -1 & (\end{cases}$$

and a weight:

$w_t = 1$

This is a '**weak classifier**': It performs slightly better than chance.

Toy example



Each data point has

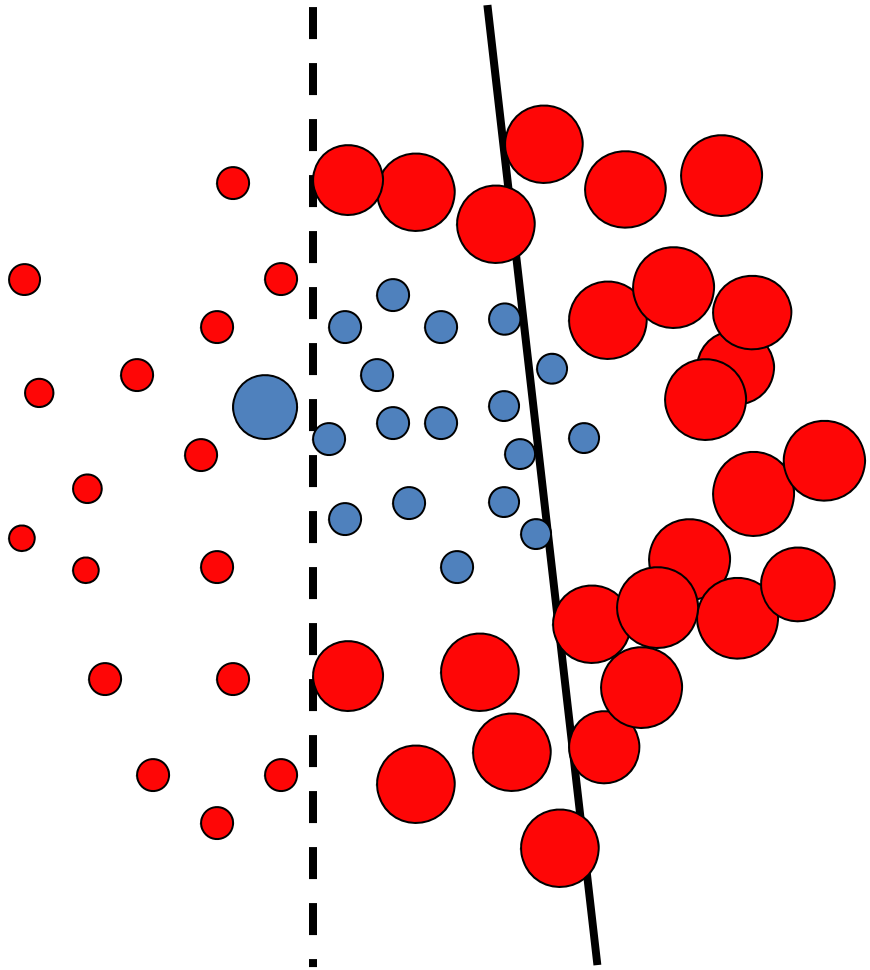
a class label:

$$y_t = \begin{cases} +1 & (\ ) \\ -1 & (\ ) \end{cases}$$

We update the weights:

$$w_t \leftarrow w_t \exp\{-y_t H_t\}$$

Toy example



Each data point has

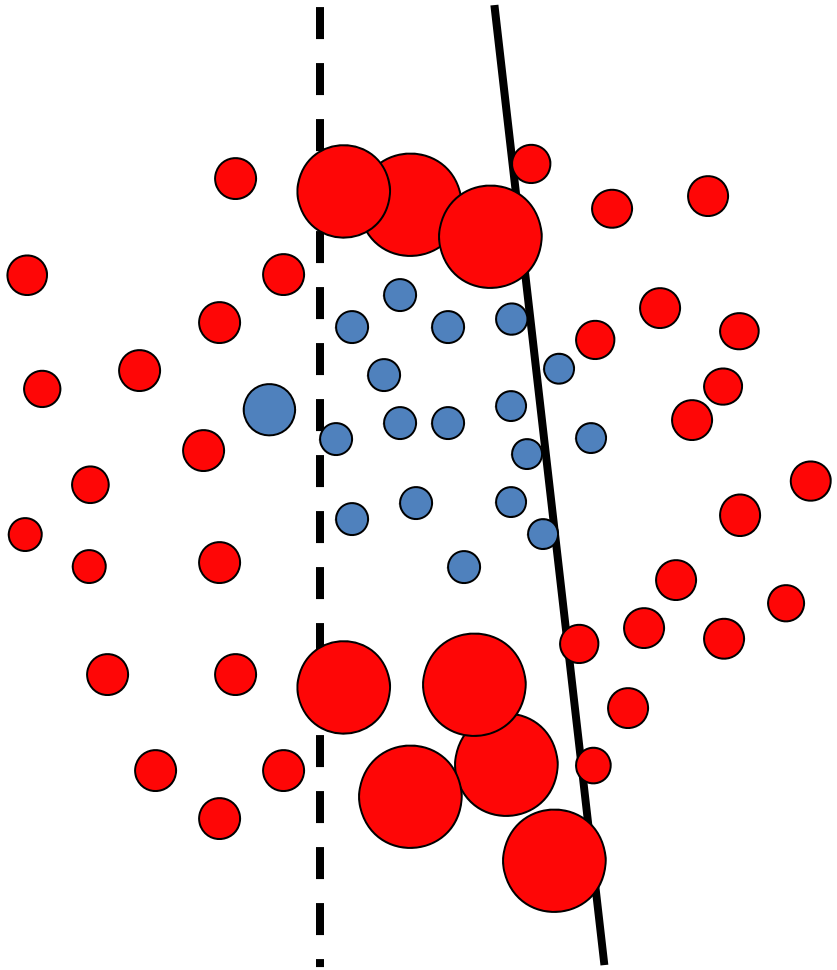
a class label:

$$y_t = \begin{cases} +1 & (\\ -1 & (\end{cases}$$

We update the weights:

$$w_t \leftarrow w_t \exp\{-y_t H_t\}$$

Toy example



Each data point has

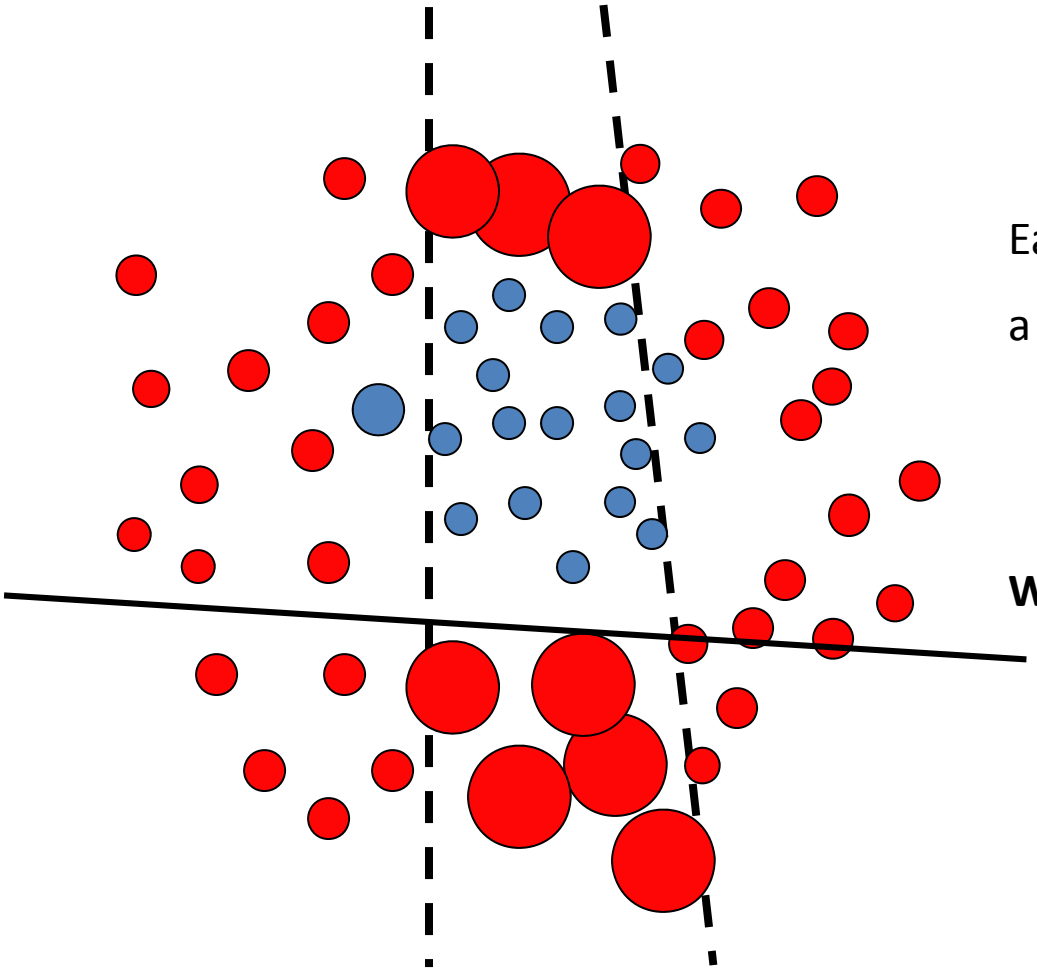
a class label:

$$y_t = \begin{cases} +1 & (\ ) \\ -1 & (\ ) \end{cases}$$

We update the weights:

$$w_t \leftarrow w_t \exp\{-y_t H_t\}$$

Toy example



Each data point has

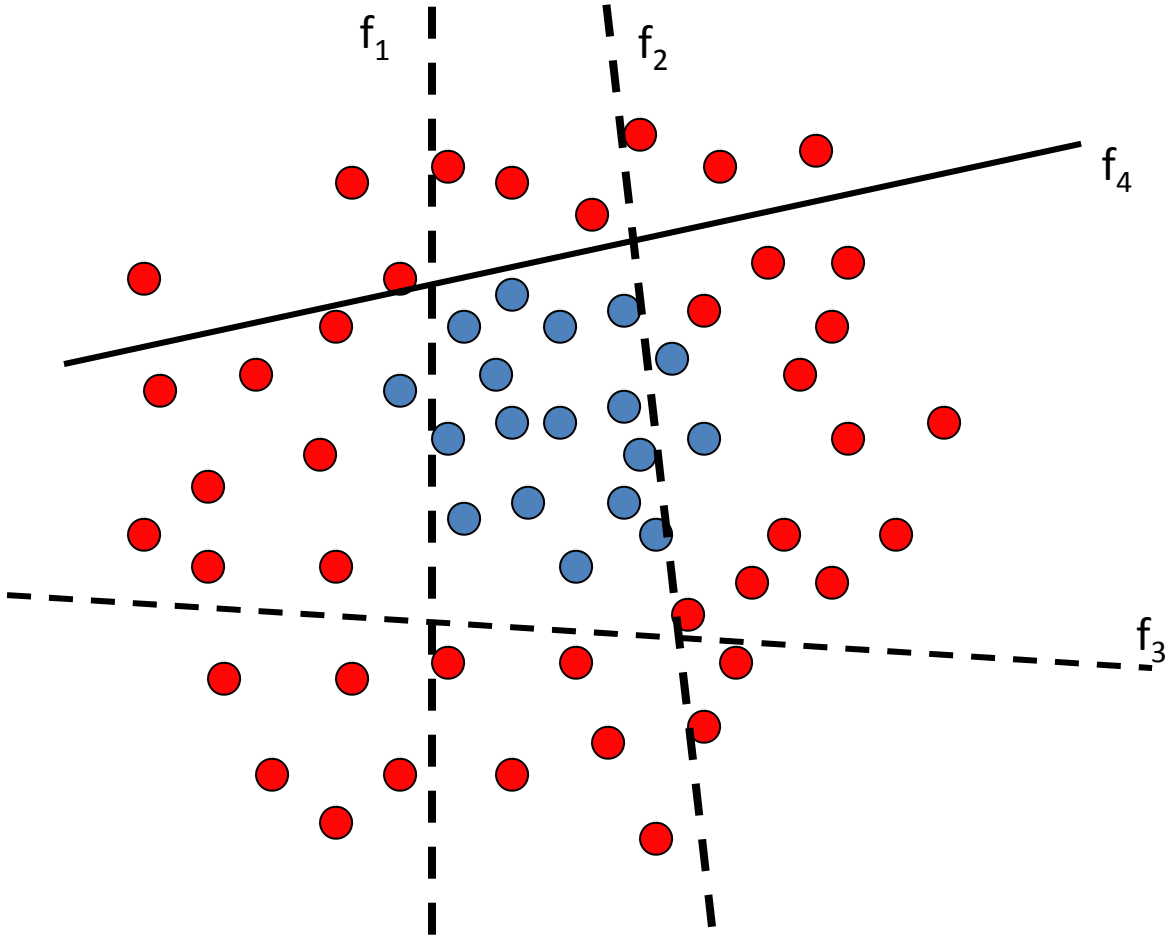
a class label:

$$y_t = \begin{cases} +1 & (\ ) \\ -1 & (\ ) \end{cases}$$

We update the weights:

$$w_t \leftarrow w_t \exp\{-y_t H_t\}$$

Toy example



The strong (non- linear) classifier is built as the combination of all the weak (linear) classifiers.

Boosting

- Defines a classifier using an additive model:

$$h(x) = \alpha_1 h_1(x) + \alpha_2 h_2(x) + \alpha_3 h_3(x) + \cdots$$

Strong classifier

Features vector

Weight

Weak classifier

Boosting

- Defines a classifier using an additive model:

$$h(x) = \alpha_1 h_1(x) + \alpha_2 h_2(x) + \alpha_3 h_3(x) + \cdots$$

Strong classifier

Features vector

Weight

Weak classifier

- We need to define a family of weak classifiers

$h_k(x)$ form a family of weak classifiers

Why boosting?

- A simple algorithm for learning robust classifiers
  - Freund & Shapire, 1995
  - Friedman, Hastie, Tibshhirani, 1998
- Provides efficient algorithm for sparse visual feature selection
  - *Tieu & Viola, 2000*
  - *Viola & Jones, 2003*
- Easy to implement, not requires external optimization tools.

Boosting - mathematics

- Weak learners

value of rectangle feature

$$h_j(x) = \begin{cases} 1 & \text{if } f_j(x) > \theta_j \\ 0 & \text{otherwise} \end{cases}$$

threshold

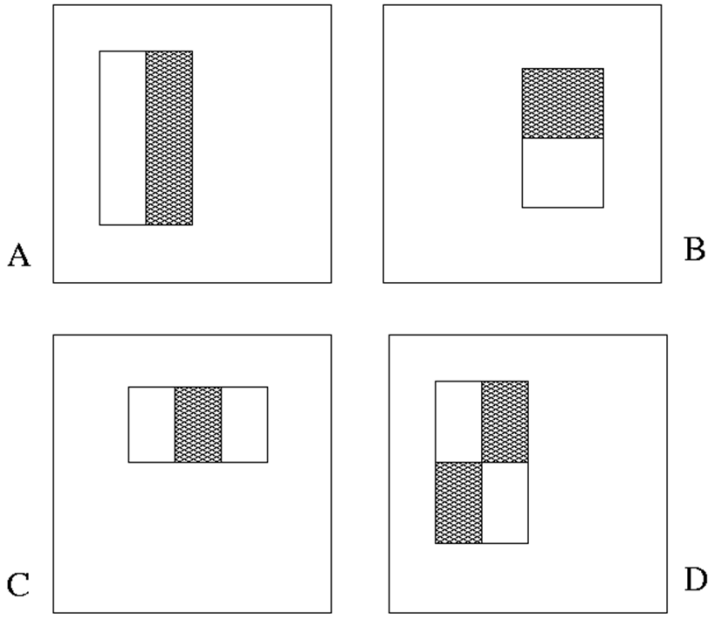
- Final strong classifier

$$h(x) = \begin{cases} 1 & \sum_{t=1}^{T} \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^{T} \alpha_t \\ 0 & \text{otherwise} \end{cases}$$

Weak classifier

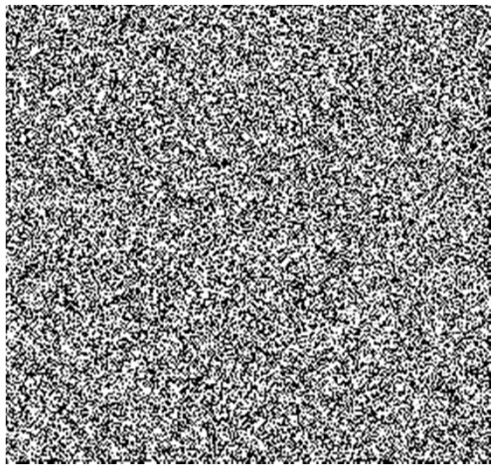
- 4 kind of Rectangle filters
- *Value =*

∑ (pixels in white area) –
∑ (pixels in black area)

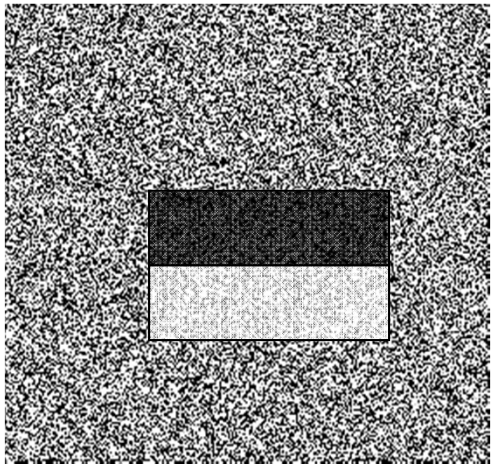


Credit slide: S. Lazebnik

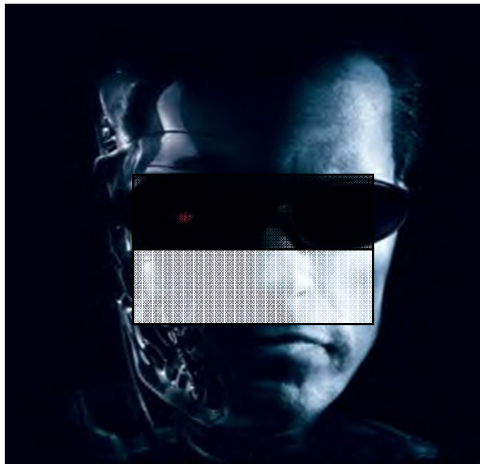
Weak classifier



Source



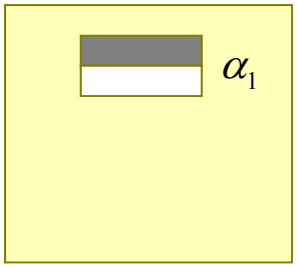
Result



Credit slide: S. Lazebnik

Viola & Jones algorithm

1. Evaluate each rectangle filter on each example



$(x_1, 1)$ $(x_2, 1)$ $(x_3, 0)$ $(x_4, 0)$ $(x_5, 0)$ $(x_6, 0)$

$\cdots\cdots$ (x_n, y_n)

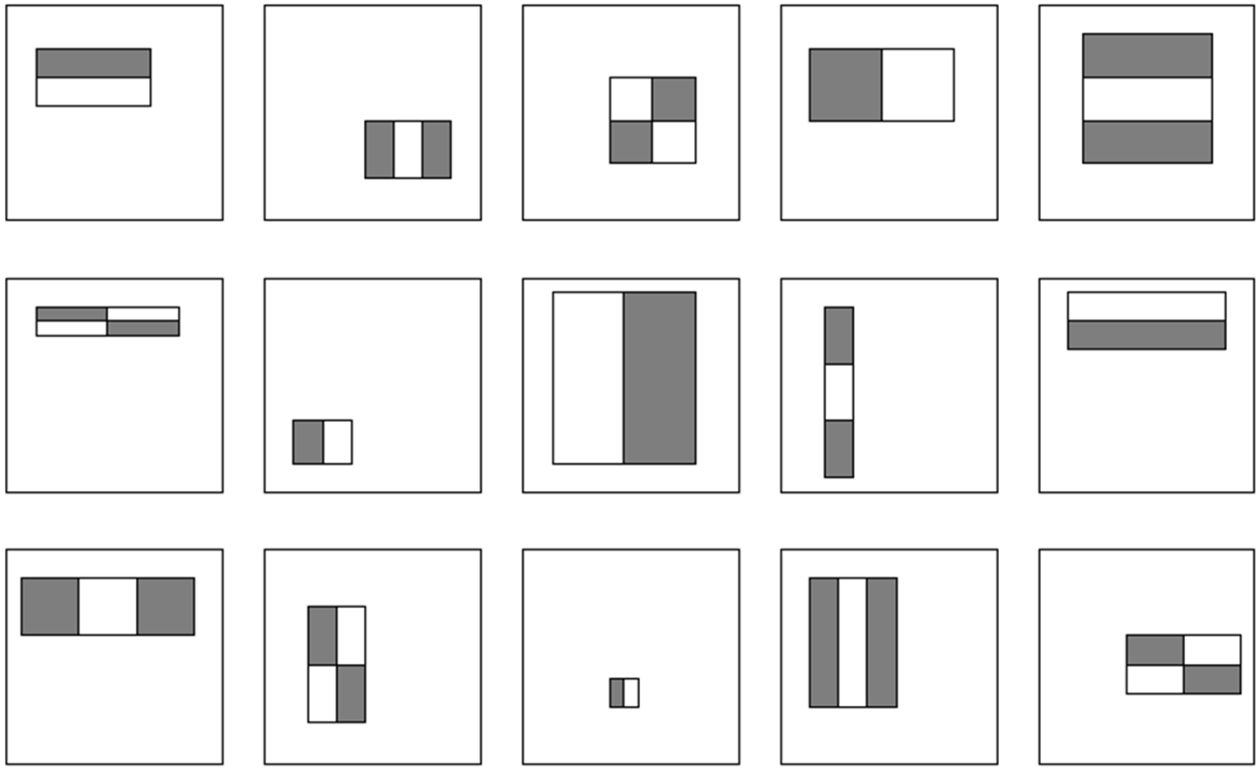
0.8 0.7 0.2 0.3 0.8 0.1

Weak classifier $h_j(x) = \begin{cases} 1 & \text{if } f_j(x) > \theta_j \\ 0 & \text{otherwise} \end{cases}$ ← threshold

P. Viola and M. Jones. *Rapid object detection using a boosted cascade of simple features*. CVPR 2001.

Viola & Jones algorithm

- For a 24x24 detection region,



P. Viola and M. Jones. *Rapid object detection using a boosted cascade of simple features*. CVPR 2001.

Viola & Jones algorithm

2. Select best filter/threshold combination

a. Normalize the weights $w_{t,i} \leftarrow \frac{w_{t,i}}{\sum_{j=1}^{n} w_{t,j}}$

$$h_j(x) = \begin{cases} 1 & \text{if } f_j(x) > \theta_j \\ 0 & \text{otherwise} \end{cases}$$

b. For each feature, j $\varepsilon_j = \sum_i w_i \left| h_j(x_i) - y_i \right|$

c. Choose the classifier, h_t with the lowest error ε_t

3. Reweight examples

$$w_{t+1,i} = w_{t,i} \beta_t^{1-|h_t(x_i)-y_i|} \qquad \beta_t = \frac{\varepsilon_t}{1-\varepsilon_t}$$

P. Viola and M. Jones. *Rapid object detection using a boosted cascade of simple features*. CVPR 2001.

Viola & Jones algorithm

4. The final strong classifier is

$$h(x) = \begin{cases} 1 & \sum_{t=1}^{T} \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^{T} \alpha_t \\ 0 & \text{otherwise} \end{cases}$$

$$\alpha_t = \log \frac{1}{\beta_t}$$

The final hypothesis is a weighted linear combination of the T hypotheses where the weights are inversely proportional to the training errors

P. Viola and M. Jones. *Rapid object detection using a boosted cascade of simple features*. CVPR 2001.

Viola & Jones algorithm

- A "paradigmatic" method for real-time object detection
- Training is slow, but detection is very fast
- Key ideas
  - *Integral images* for fast feature evaluation
  - *Boosting* for feature selection
  - *Attentional cascade* for fast rejection of non-face windows

P. Viola and M. Jones. *Rapid object detection using a boosted cascade of simple features*. CVPR 2001.

Credit slide: S. Lazebnik

The implemented system

- Training Data
  - 5000 faces
    - All frontal, rescaled to 24x24 pixels
  - 300 million non-faces
    - 9500 non-face images
  - Faces are normalized
    - Scale, translation
- Many variations
  - Across individuals
  - Illumination
  - Pose

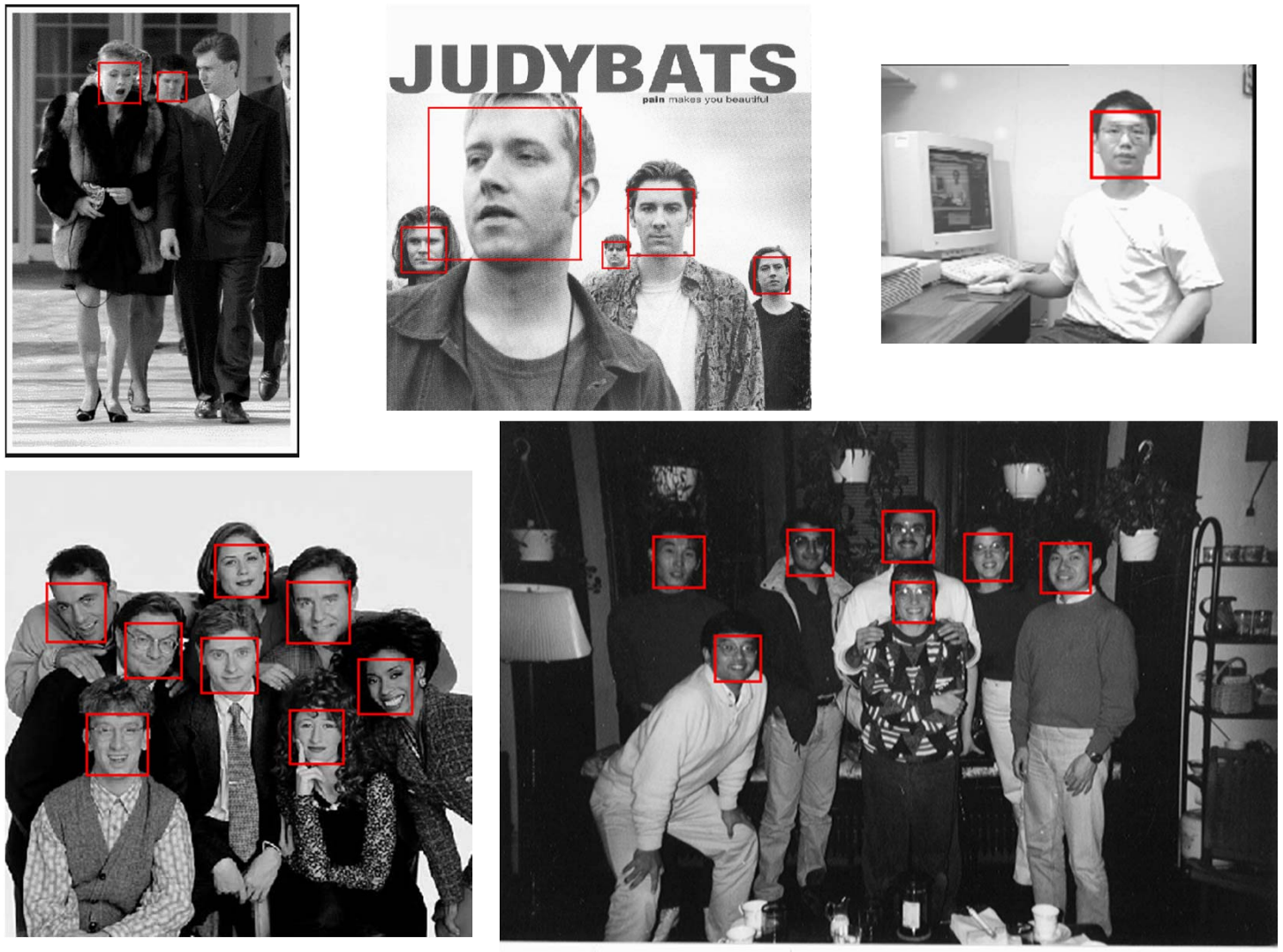
P. Viola and M. Jones. *Rapid object detection using a boosted cascade of simple features*. CVPR 2001.

System performance

- Training time: “weeks” on 466 MHz Sun workstation
- 38 layers, total of 6061 features
- Average of 10 features evaluated per window on test set
- “On a 700 Mhz Pentium III processor, the face detector can process a 384 by 288 pixel image in about .067 seconds”
  - 15 Hz
  - 15 times faster than previous detector of comparable accuracy (Rowley et al., 1998)

P. Viola and M. Jones. *Rapid object detection using a boosted cascade of simple features*. CVPR 2001.

Output of Face Detector on Test Images



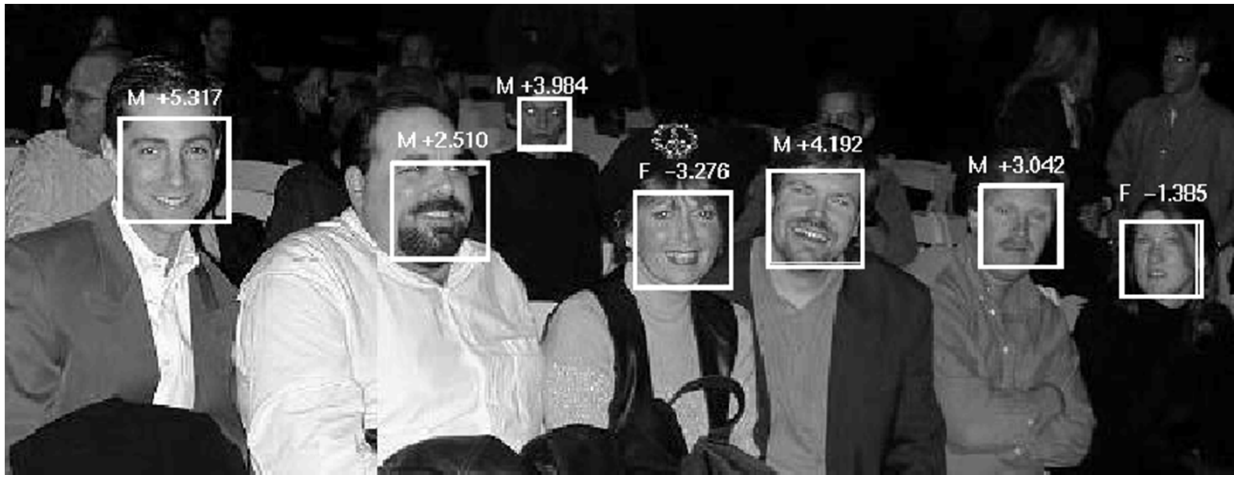
P. Viola and M. Jones. *Rapid object detection using a boosted cascade of simple features*. CVPR 2001.

Other detection tasks

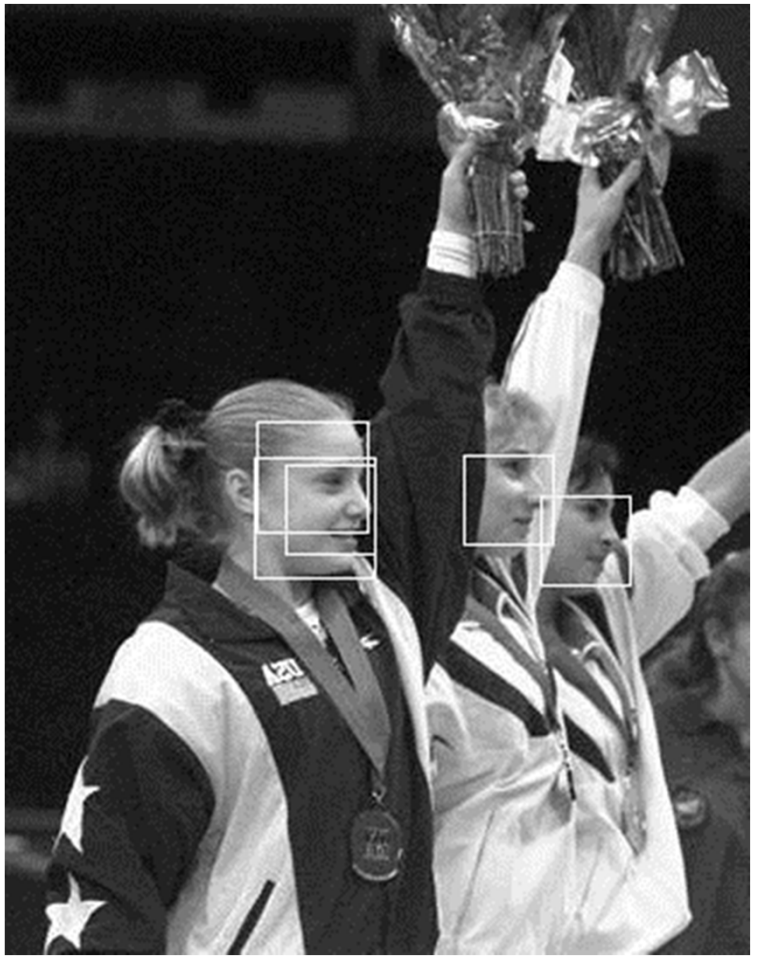
Facial Feature Localization

Profile Detection

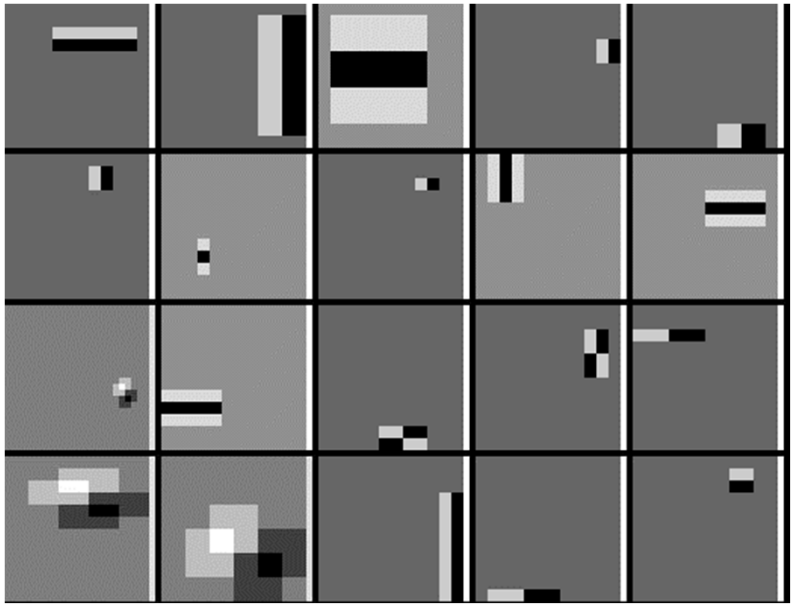
Male vs. female



Profile Detection



Profile Features



Face Image Databases

- Databases for face recognition can be best utilized as training sets
  - Each image consists of an individual on a uniform and uncluttered background
- Test Sets for face detection
  - MIT, CMU (frontal, profile), Kodak

Experimental Results

- Test dataset
  - MIT+CMU frontal face test set
  - 130 images with 507 labeled frontal faces

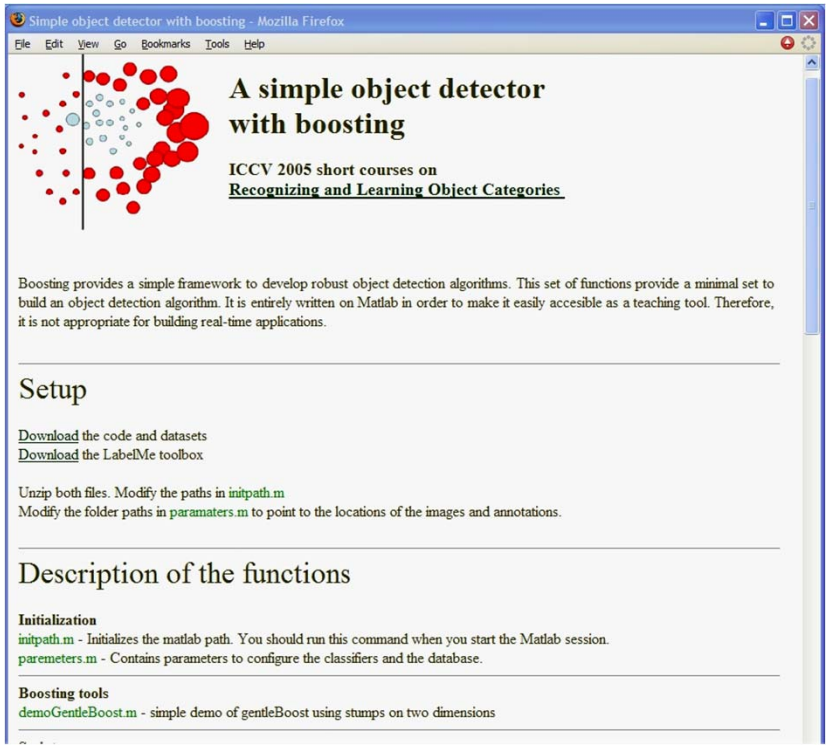
False detection	10	31	50	65	78	95	110	167	422
AdaBoost	78.3	85.2	88.8	89.8	90.1	90.8	91.1	91.8	93.7
Neural-net	83.2	86.0	-	-	-	89.2	-	90.1	89.9

MIT test set: 23 images with 149 faces
Sung & poggio: detection rate 79.9% with 5 false positive
AdaBoost: detection rate 77.8% with 5 false positives

Sharing features with Boosting

Sharing features: efficient boosting procedures for multiclass object detection

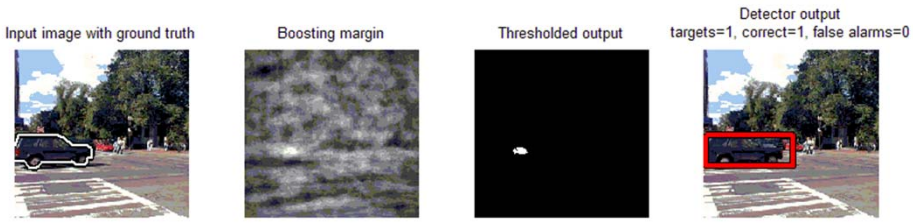
A. Torralba, K. P. Murphy and W. T. Freeman Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR). pp 762-769, 2004.



http://people.csail.mit.edu/torralba/iccv2005/

Matlab code

- Gentle boosting
- Object detector using a part based model



What we have learned today

- Recognition problems
- Face discrimination
  - Principal Component Analysis (PCA) and Eigenfaces (Problem Set 1 (Q1))
  - Linear Discriminant Analysis (LDA) and Fisherfaces
- Face detection
  - SVM (Problem Set 0 (Q2))
  - Boosting

Supplementary materials

Fisher Faces: Linear Discriminant Analysis

Variables

- N Sample images: $\{x_1, \cdots, x_N\}$
- c classes: $\{\chi_1, \cdots, \chi_c\}$
- Average of each class: $\mu_i = \frac{1}{N_i} \sum_{x_k \in \chi_i} x_k$
- Total average: $\mu = \frac{1}{N} \sum_{k=1}^{N} x_k$

Scatters

- Scatter of class i:

$$S_i = \sum_{x_k \in \chi_i} (x_k - \mu_i)(x_k - \mu_i)^T$$

- Within class scatter:

$$S_W = \sum_{i=1}^{c} S_i$$

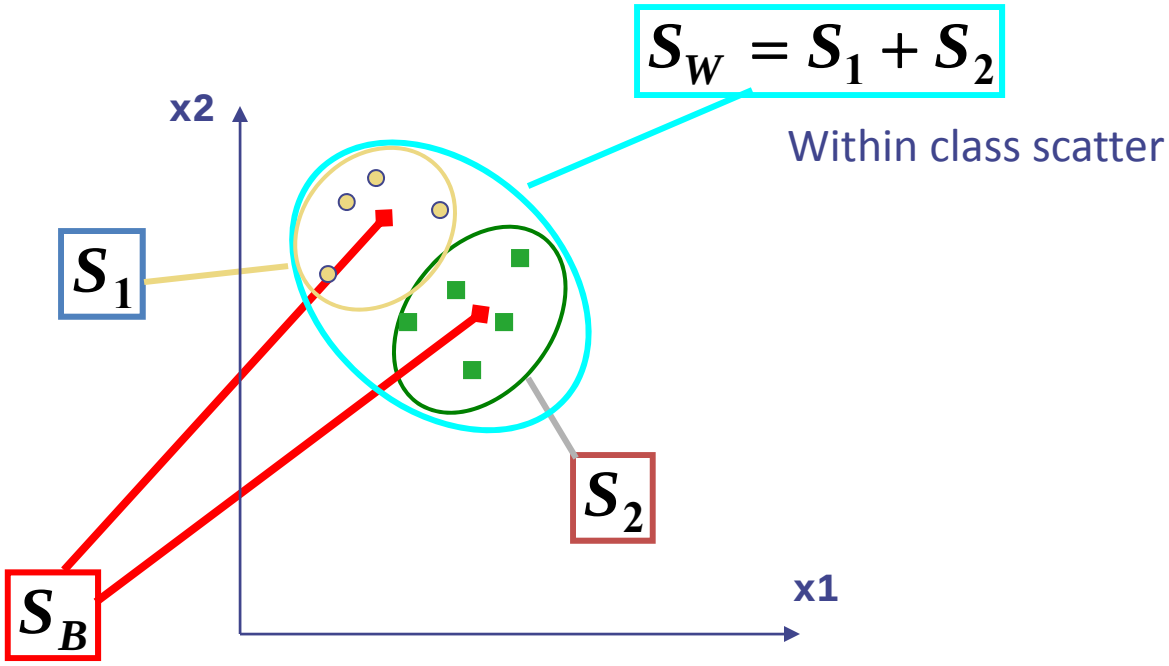
- Between class scatter:

$$S_B = \sum_{i=1}^{c} |\chi_i| (\mu_i - \mu)(\mu_i - \mu)^T$$

- Total scatter:

$$S_T = S_W + S_B$$

Illustration



$$S_i = \sum_{x_k \in \chi_i} (x_k - \mu_i)(x_k - \mu_i)^T$$

$$S_W = \sum_{i=1}^{c} S_i$$

$$S_B = \sum_{i=1}^{c} |\chi_i| (\mu_i - \mu)(\mu_i - \mu)^T$$

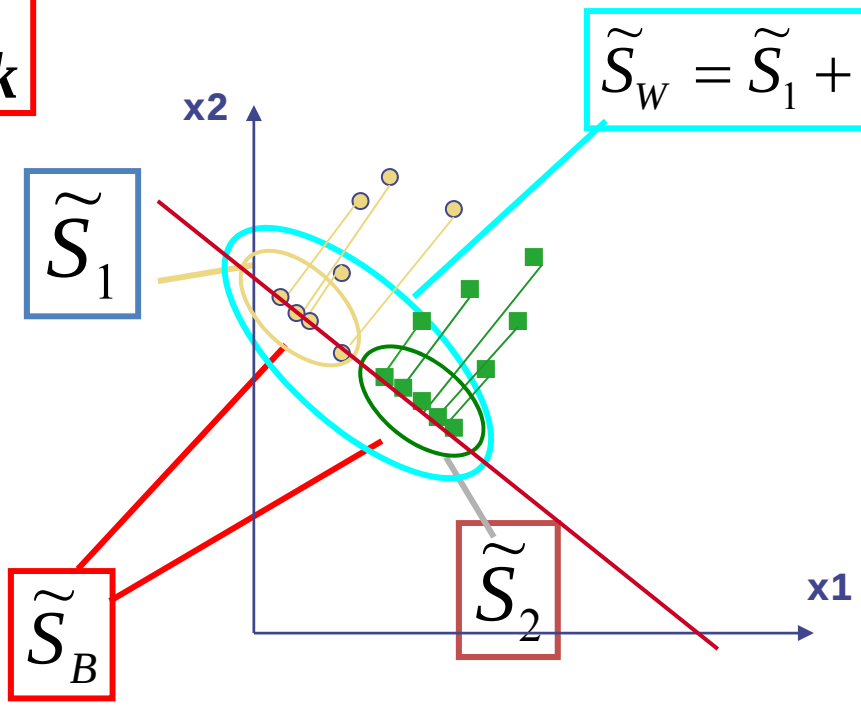
Mathematical Formulation (1)

- After projection: $y_k = W^T x_k$

- Between class scatter (of y's): $\tilde{S}_B = W^T S_B W$
- Within class scatter (of y's): $\tilde{S}_W = W^T S_W W$

Illustration

$$y_k = W^T x_k$$



$$S_W = \sum_{i=1}^{c} S_i$$

$$S_B = \sum_{i=1}^{c} |\chi_i| (\mu_i - \mu)(\mu_i - \mu)^T$$

$$\tilde{S}_W = W^T S_W W$$

$$\tilde{S}_B = W^T S_B W$$

Mathematical Formulation

- The desired projection:

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

- How is it found ? → Generalized Eigenvectors

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

- If S_w has full rank, the generalized eigenvectors are eigenvectors of $S_W^{-1} S_B$ with largest eigen-values

Training/ Testing

Projection in Eigenface

Projection $\omega_i = W_{opt} (X - u)$,

W_{opt} = {fisher-faces}