• Face is the most common biometric used by humans
• Applications range from static, mug-shot verification to a dynamic, uncontrolled face identification in a cluttered background
• Challenges:
  • automatically locate the face
  • recognize the face from a general view point under different illumination conditions, facial expressions, and aging effects

• Face Authentication/Verification (1:1 matching)

• Face Identification/Recognition (1:N matching)
● Access Control

![Access Control Image]

www.visage.com

● Video Surveillance (On-line or off-line)

![Video Surveillance Image]

Face Scan at Airports

www.facesnap.de
Why is Face Recognition Hard?

Many faces of Madonna

- Identify similar faces *(inter-class similarity)*
- Accommodate *intra-class variability* due to:
  - head pose
  - illumination conditions
  - expressions
  - facial accessories
  - aging effects
  - Cartoon faces
• Different persons may have very similar appearance

Twins

Father and son

• Faces with intra-subject variations in pose, illumination, expression, accessories, color, occlusions, and brightness
Sketch of a Pattern Recognition Architecture

Example: Face Detection

- Scan window over image
- Classify window as either:
  - Face
  - Non-face
### Detection Test Sets

<table>
<thead>
<tr>
<th>Data Set</th>
<th>Location</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMU Profile Face Test Set [13]</td>
<td>[ftp://eyes.ics.cs.cmu.edu/acs/20/ftp/beating_face_images.tar.gz]</td>
<td>208 gray scale images with faces in profile views.</td>
</tr>
<tr>
<td>Kodak Data Set [94]</td>
<td>Eastman Kodak Corporation</td>
<td>Faces of multiple sizes, pose and under varying illumination in color images. Designed for face detection and recognition.</td>
</tr>
</tbody>
</table>

### Profile views

**Schneiderman’s Test set**
Face Detection: Experimental Results

Test sets: two CMU benchmark data sets
Test set 1: 125 images with 483 faces
Test set 2: 20 images with 136 faces

<table>
<thead>
<tr>
<th>Method</th>
<th>Test Set 1</th>
<th>Test Set 2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Detection Rate</td>
<td>False Detections</td>
</tr>
<tr>
<td>Distribution based [154]</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Neural network [128]</td>
<td>92.5%</td>
<td>862</td>
</tr>
<tr>
<td>Naive Bayes classifier [100]</td>
<td>93.9%</td>
<td>88</td>
</tr>
<tr>
<td>Kullback relative information [24]</td>
<td>96.0%</td>
<td>12758</td>
</tr>
<tr>
<td>Support vector machine [107]</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Mixture of factor analyzers [173]</td>
<td>92.3%</td>
<td>82</td>
</tr>
<tr>
<td>Fisher linear discriminant [175]</td>
<td>93.8%</td>
<td>74</td>
</tr>
<tr>
<td>SNoW with primitive features [176]</td>
<td>94.2%</td>
<td>84</td>
</tr>
<tr>
<td>SNoW with multi-scale features [177]</td>
<td>94.5%</td>
<td>76</td>
</tr>
<tr>
<td>Inductive learning [38]</td>
<td>90%</td>
<td>N/A</td>
</tr>
</tbody>
</table>

[See also work by Viola & Jones, Rehg, more recent by Schneiderman]

Example: Finding skin
Non-parametric Representation of CCD

- Skin has a very small range of (intensity independent) colors, and little texture
  - Compute an intensity-independent color measure, check if color is in this range, check if there is little texture (median filter)
  - See this as a classifier - we can set up the tests by hand, or learn them.
  - get class conditional densities (histograms), priors from data (counting)

- Classifier is
  - if $p(\text{skin}|x) > \theta$, classify as skin
  - if $p(\text{skin}|x) < \theta$, classify as not skin
  - if $p(\text{skin}|x) = \theta$, choose classes uniformly and at random
Figure from “Statistical color models with application to skin detection,” M.J. Jones and J. Rehg, Proc. Computer Vision and Pattern Recognition, 1999 copyright 1999, IEEE.

Face Detection
Face Detection Algorithm

- Lighting Compensation
- Color Space Transformation
- Skin Color Detection
- Variance-based Segmentation
- Connected Component & Grouping
- Eye/ Mouth Detection
- Face Boundary Detection
- Verifying/ Weighting
- Eyes-Mouth Triangles
- Facial Feature Detection

Input Image

Output Image

Canon Powershot
Face Recognition: 2-D and 3-D

Time (video)

2-D Recognition Comparison

3-D

Face Database

Prior knowledge of face class

Recognition Data

Algorithms

Pose-dependent

Pose-invariant

Viewer-centered images

Object-centered Models

Appearance-based (Holistic)

Feature-based (Analytic)

Hybrid

PCA, LDA

LFA

EGBM

Gordon et al., 1995

Lengagne et al., 1996

Atick et al., 1996

Yan et al., 1996

Zhao et al., 2000

Zhang et al., 2000
**Image as a Feature Vector**

- Consider an n-pixel image to be a point in an n-dimensional space, $\mathbf{x} \in \mathbb{R}^n$.
- Each pixel value is a coordinate of $\mathbf{x}$.

**Nearest Neighbor Classifier**

\[ \{ R_j \} \] are set of training images.

\[ ID = \arg \min_j \text{dist}(R_j, I) \]
Comments

- Sometimes called “Template Matching”
- Variations on distance function (e.g. $L_1$, robust distances)
- Multiple templates per class - perhaps many training images per class.
- Expensive to compute k distances, especially when each image is big ($N$ dimensional).
- May not generalize well to unseen examples of class.
- Some solutions:
  - Bayesian classification
  - Dimensionality reduction

Eigenfaces (Turk, Pentland, 91) -1

- Use Principle Component Analysis (PCA) to reduce the dimensionality
How do you construct Eigenspace?

Construct data matrix by stacking vectorized images and then apply Singular Value Decomposition (SVD)

Eigenfaces

- **Modeling**
  1. Given a collection of n labeled training images,
  2. Compute mean image and covariance matrix.
  3. Compute k Eigenvectors (note that these are images) of covariance matrix corresponding to k largest Eigenvalues.
  4. Project the training images to the k-dimensional Eigenspace.

- **Recognition**
  1. Given a test image, project to Eigenspace.
  2. Perform classification to the projected training images.
Eigenfaces: Training Images

[ Turk, Pentland 01]

Eigenfaces

Mean Image  Basis Images
Difficulties with PCA

• Projection may suppress important detail
  – smallest variance directions may not be unimportant

• Method does not take discriminative task into account
  – typically, we wish to compute features that allow good discrimination
  – not the same as largest variance
Fisherfaces: Class specific linear projection

• An n-pixel image $\mathbf{x} \in \mathbb{R}^n$ can be projected to a low-dimensional feature space $\mathbf{y} \in \mathbb{R}^m$ by

$$ \mathbf{y} = \mathbf{Wx} $$

where $\mathbf{W}$ is an $n \times m$ matrix.

• Recognition is performed using nearest neighbor in $\mathbb{R}^m$.

• How do we choose a good $\mathbf{W}$?

PCA & Fisher’s Linear Discriminant

• Between-class scatter

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

• Within-class scatter

$$ S_W = \sum_{i=1}^{c} \sum_{\chi \in \chi_i} (\mathbf{x} - \mu_i)(\mu_i - \mu)^T $$

• Total scatter

$$ S_T = \sum_{i=1}^{c} \sum_{\chi \in \chi_i} (\mathbf{x} - \mu_i)(\mu_i - \mu)^T = S_B + S_W $$

• Where
  - $c$ is the number of classes
  - $\mu_i$ is the mean of class $\chi_i$
  - $|\chi_i|$ is number of samples of $\chi_i$. 

\[ \begin{align*}
\chi_1 & \quad \chi_2 \\
\mu_1 & \quad \mu_2 \\
\mu & \quad \mu \\
\chi_1 & \quad \chi_2
\end{align*} \]
PCA & Fisher’s Linear Discriminant

- **PCA (Eigenfaces)**
  \[ W_{PCA} = \arg \max_w \|W^T S_I W \| \]
  Maximizes projected total scatter

- **Fisher’s Linear Discriminant**
  \[ W_{\text{FLD}} = \arg \max_w \frac{W^T S_b W}{W^T S_w W} \]
  Maximizes ratio of projected between-class to projected within-class scatter

Four Fisherfaces From ORL Database
Eigenfaces and Fisherfaces