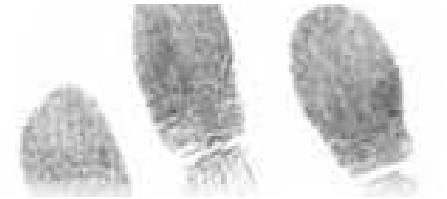




Face Detection

University at Buffalo CSE666 Lecture Slides

Resources:



Applications

- Biometrics
- Emotional Expression
- Person tracking
 - Surveillance
 - Scene analysis
 - Military applications
- General object location and recognition

Still photo vs. Video, still camera vs. Video, moving camera

most difficult

↑
easiest

?



Skin Color Approaches

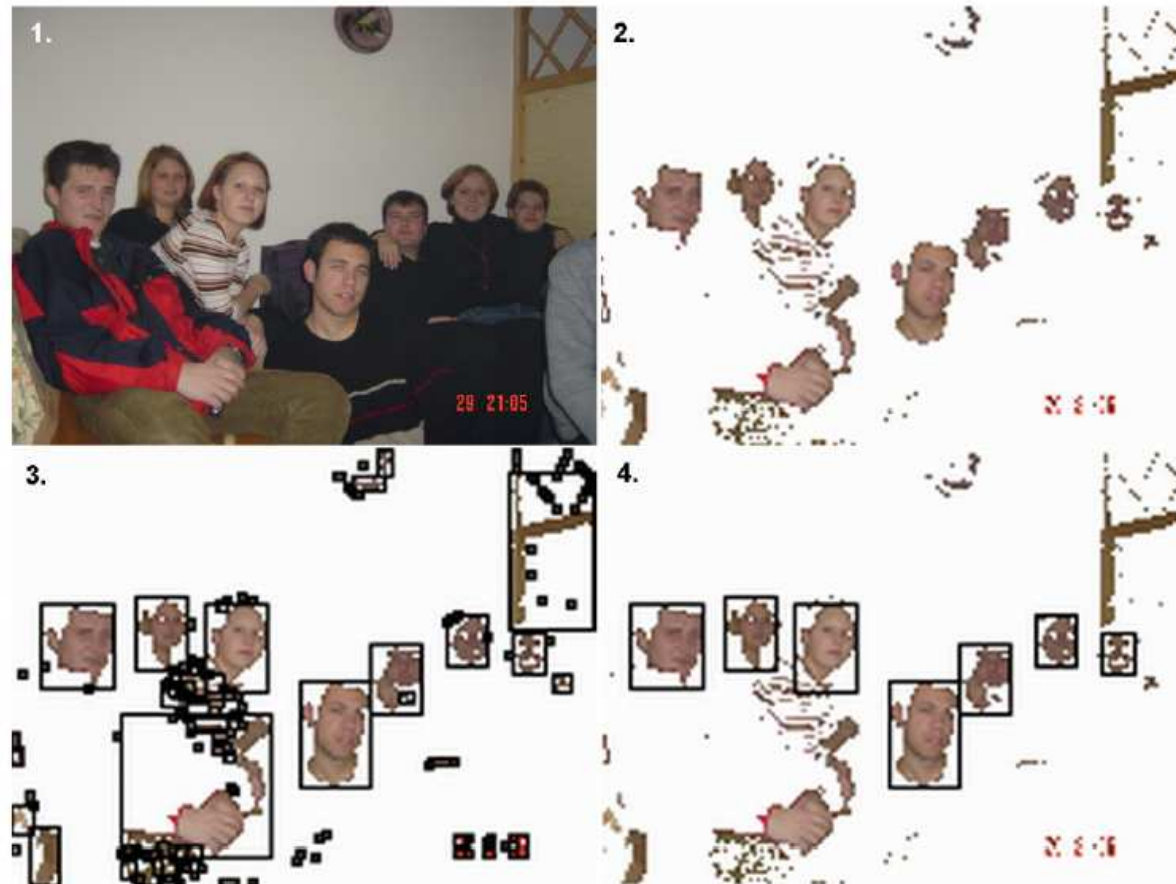


Fig. 2. Stages in the process of finding faces in an image in the installation "15 Seconds of Fame": 1) reduce the original image resolution of 2048×1536 pixels to 160×120 pixels, 2) eliminate all pixels that cannot represent a face, 3) segment all the regions containing face-like pixels using region growing, 4) eliminate regions, which cannot represent a face using heuristic rules.



Classification Approaches

- Select a region from image
- Extract some features
- Classify feature vector – face or no face

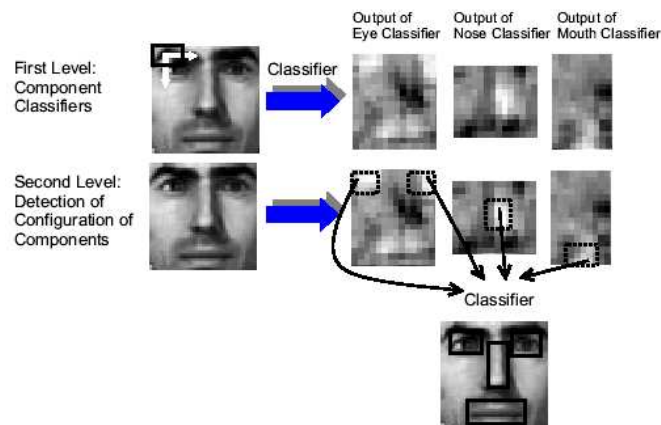


Figure 3. System overview of the component-based classifier using four components. On the first level, windows of the size of the components (solid lined boxes) are shifted over the face image and classified by the component classifiers. On the second level, the maximum outputs of the component classifiers within predefined search regions (dotted lined boxes) and the positions of the components are fed into the geometrical configuration classifier.

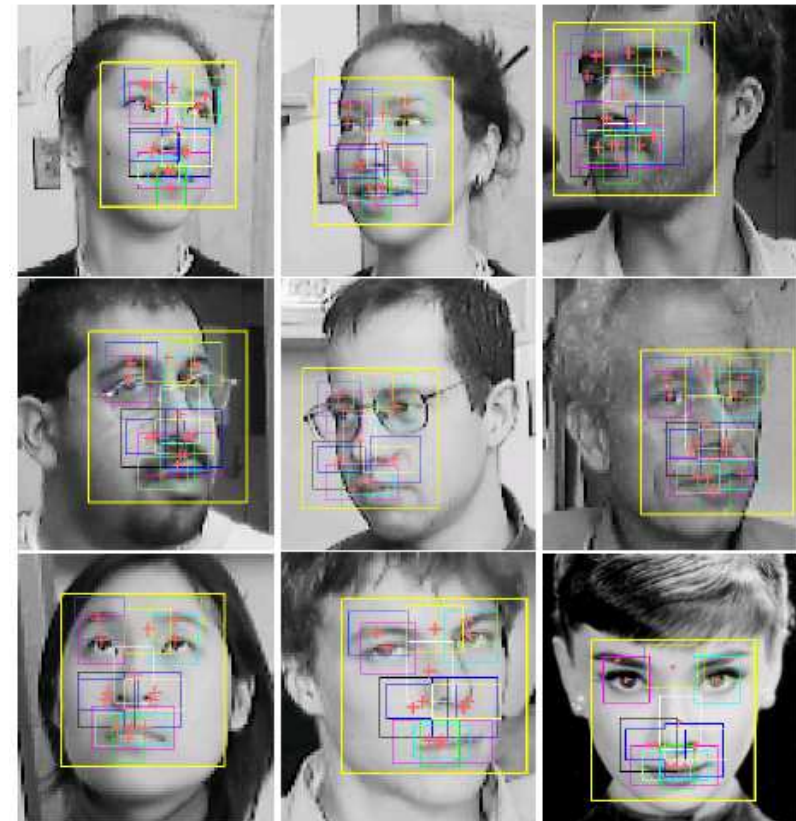
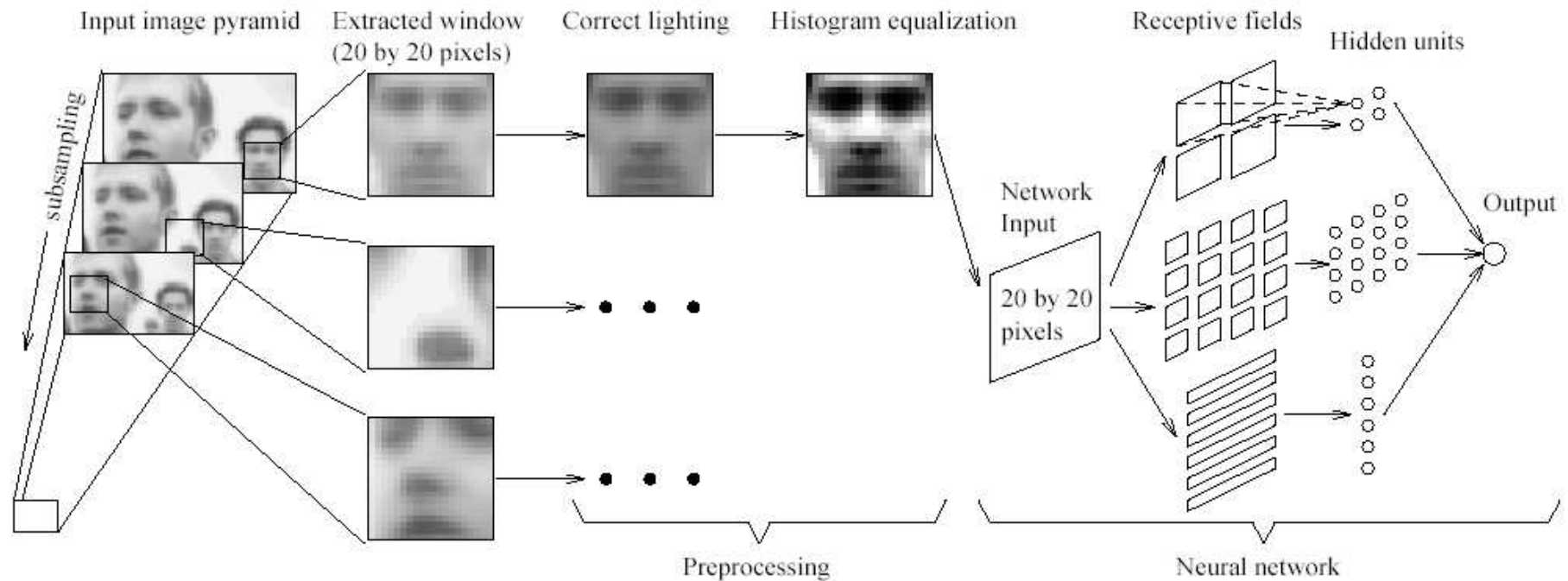


Figure 8. Faces detected by the 14 component system.



Neural Network





AdaBoost

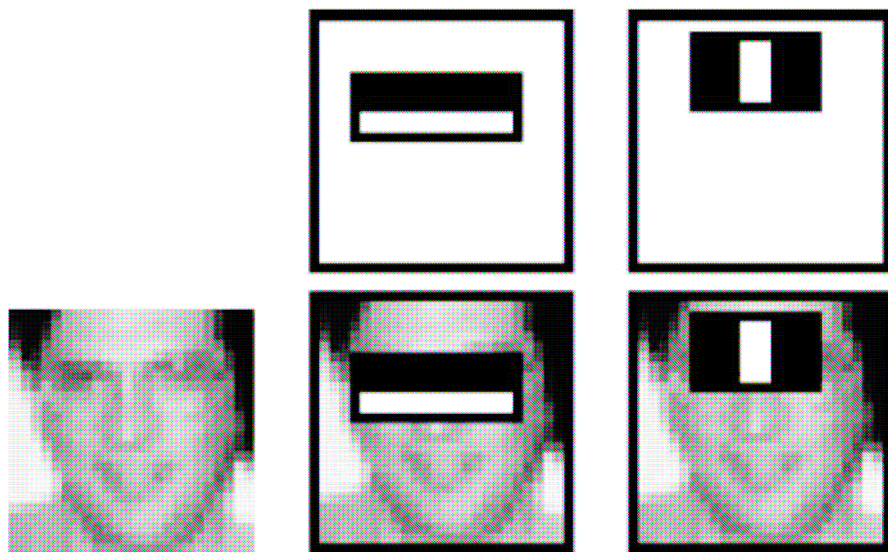
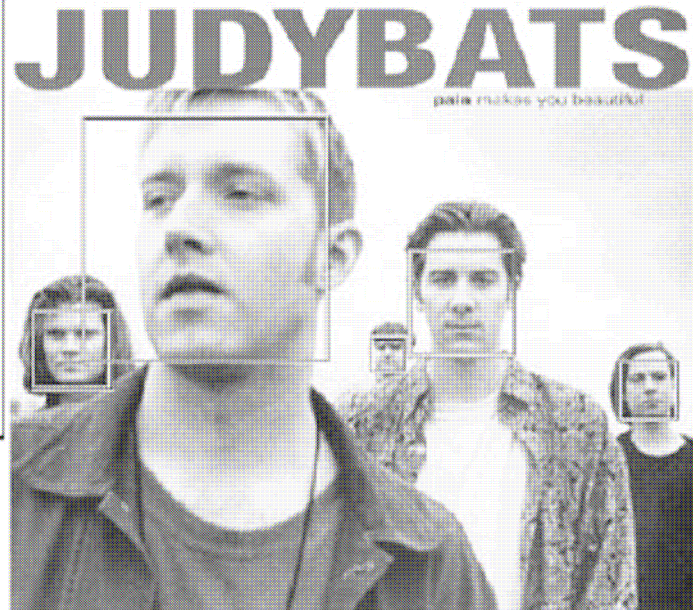


Figure 5. The first and second features selected by AdaBoost. The two features are shown in the top row and then overlayed on a typical training face in the bottom row. The first feature measures the difference in intensity between the region of the eyes and a region across the upper cheeks. The feature capitalizes on the observation that the eye region is often darker than the cheeks. The second feature compares the intensities in the eye regions to the intensity across the bridge of the nose.

- Iteratively chooses which ‘weak classifiers’ (features) are best for currently incorrectly classified images.
- Found best feature is added to final classifier in weighted sum.
- Additional cascading structure is used in paper



P. Viola, M. Jones “Robust Real-Time Face Detection”, 2004



Principal Component Analysis (PCA)

Previous discussion:

PCA as a feature extraction method

\mathbf{a}_i - eigenvectors of matrix $\mathbf{R}_x = E[\mathbf{x}\mathbf{x}^T]$

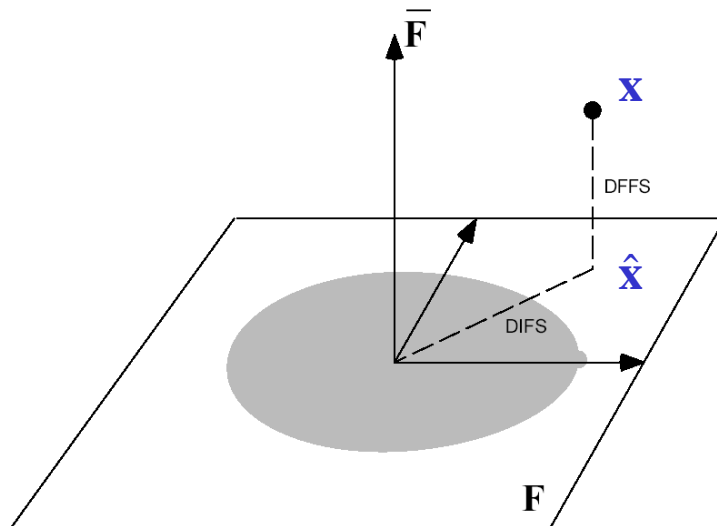
$\mathbf{A} = (\mathbf{a}_i)$ - Karhunen-Loeve transform

Features	Representation
$\mathbf{y} = \mathbf{A}\mathbf{x}$ - projection of original vector \mathbf{x} to KL basis	$\mathbf{x} = \sum_{i=1}^n y_i \mathbf{a}_i$ - representation of original vector \mathbf{x} in KL basis vectors
$\mathbf{y}_m = \mathbf{A}_m \mathbf{x}$ - m first PCA coefficients – ‘best’ features	$\hat{\mathbf{x}} = \sum_{i=1}^m y_i \mathbf{a}_i$ - the projection on subspace spanned by eigenvectors with m largest eigenvalues

$$E[(\mathbf{x} - \hat{\mathbf{x}})^2] = \sum_{i=m+1}^n \lambda_i \quad \text{- sum of the smallest } n - m \text{ eigenvalues}$$



PCA as appearance model



$\mathbf{y}_m = \mathbf{A}_m \mathbf{x}$ - m first PCA coefficients –

‘best’ features - should provide a good approximation to all the samples (for which PCA was trained)

$$\hat{\mathbf{x}} = \mathbf{A}_m^T \mathbf{y}_m$$

• Distance from feature space (DFFS) should be small:

$$DFFS = | \mathbf{x} - \hat{\mathbf{x}} |$$

- PCA provides a model on typical samples of the class
- Can use it to separate class from non-class samples



PCA for face detection

- Collect a set of face images for training
- Train PCA
- During face detection, for each query region, if projection of query region onto PCA feature space is close to original region:

$$| \mathbf{x} - \hat{\mathbf{x}} | < \theta$$

then accept region as a face region



But:

- many training images are required
- high complexity of PCA training



Active Shape Model (ASM)

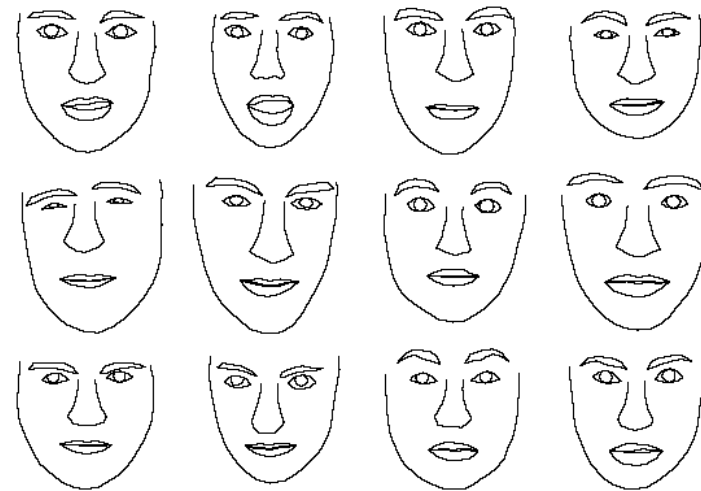
Idea:

use PCA not on the pixels of face image, but only on selected model points

Training

- 133 landmark points in each image represent the face shape
- Landmark points correspond to edges – points of large intensity change
- PCA training is performed using vectors with all 133 landmark point coordinates

$$v = (x_1, y_1, \dots, x_{133}, y_{133})$$





Active Shape Model (ASM)

Fitting ASM:

- Begin with best guess placement of shape on the query image
- Iteratively update shape so that
 1. PCA model holds $|\mathbf{v} - \hat{\mathbf{v}}| < \theta$
 2. The local profile at landmark points coincides with profiles of training images

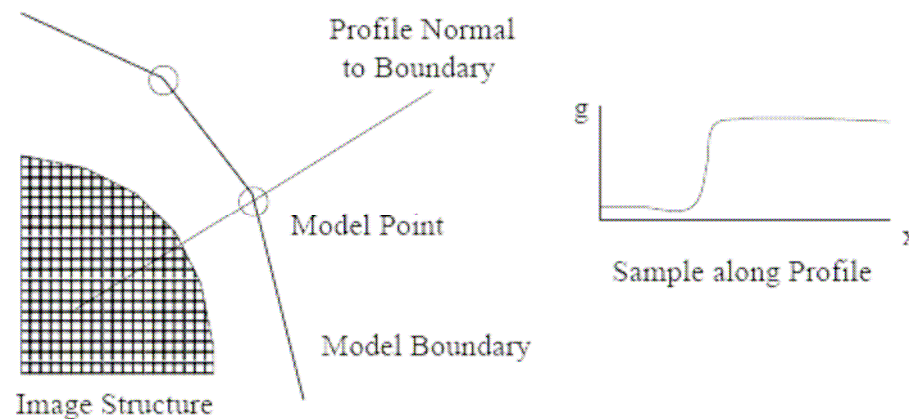


Figure 7.1: At each model point sample along a profile normal to the boundary



Active Shape Model (ASM)



Figure 7.5: Search using Active Shape Model of a face

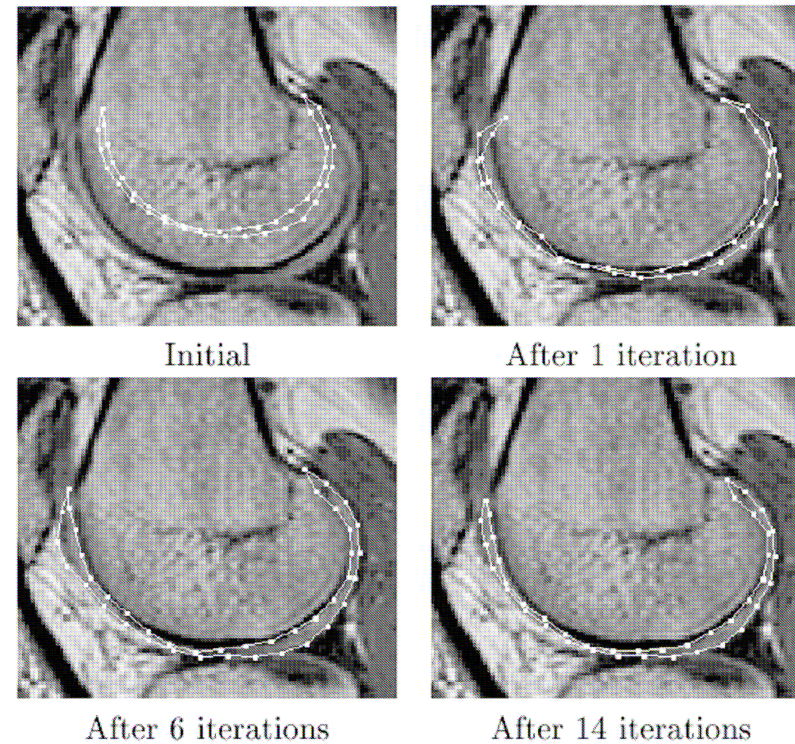


Figure 7.7: Search using ASM of cartilage on an MR image of the knee



Active Shape Model (ASM)



Initial



After 2 iterations



After 20 Iterations

Figure 7.6: Search using Active Shape Model of a face, given a poor starting point. The ASM is a local method, and may fail to locate an acceptable result if initialised too far from the target



Active Appearance Model (AAM)

- A generalization and improvement on ASM
- In ASM the correspondence between landmark points and local texture of the image (say, edge property) was calculated heuristically
- AAM: model the texture of the image as well as shape of landmark points by PCA

v -shape parameters

g - texture parameter

Train PCA on concatenated vectors **c = (v, g)**

-still have one PCA

- Model fitting error

$$| \mathbf{c} - \hat{\mathbf{c}} | < \theta$$

- Search for a good fit:

It is possible to train a search algorithm on difference images $I - \text{Image}(\hat{\mathbf{c}})$ using training data and perturbing it

⇒ Very fast searching algorithm



Active Appearance Model (AAM)



Figure 12: Texture deformation using the three largest principal modes (row-wise, top-down).



Active Appearance Model (AAM)



Figure 15: Combined deformation using the three largest principal modes (row-wise, top-down).



Active Appearance Model (AAM)

- We used implementation by M. Stegmann
- Works well for tracking



Figure 2: Example annotation of a face using 58 landmarks.