FFI Forsvarets forskningsinstitutt

Lecture 13.1
Face recognition

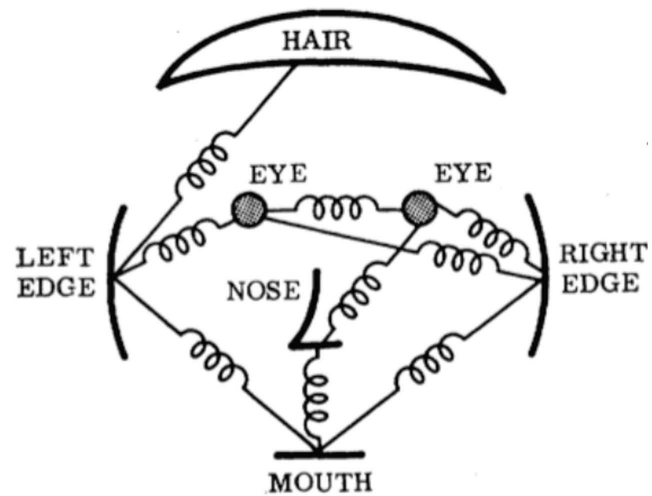
28.04.2015

Øivind Midtgaard
Principal Scientist

Some slides from L. Lazebnik, P. Viola and J. Hays

Face recognition

- **Detection** (*Given an image: where are the faces, if any?*)
  - Feature-based
    - Locate and verify arrangement of distinctive features (eyes, nose, mouth, etc)
  - Template-based
    - Match template model with image
  - Appearance-based
    - Compute basic features in sliding window and send to classifier



- **Recognition** (*Given a located face: which face is it?*):
  - Subspace processing
  - Active appearance models

Fischler, Elschlager. [Pictorial Structures](),1973

Face detection and recognition

Face recognition challenges

- Pose variations (relative camera and object)
- Illumination variations (directions, intensity, spectra)
- Occlusions (by other people or objects)
- Intra-class variations
  - Facial expressions
  - Glasses, beard, hair-style, etc

- Clutter and complex background

Consumer application: Apple iPhoto

http://www.apple.com/ilife/iphoto/

Consumer application: Apple iPhoto

- Things iPhoto thinks are faces

Face detection

- Faces are rare, typically 0-10 per image
- Sliding window detectors must still evaluate every location/scale combination in the image
- A megapixel image has in the order of 10^6 candidate face locations
- To avoid having one false positive per image on average, the false positive rate must be less than 10^{-6}
- For computational efficiency, the detector should spend as little time as possible on the non-face windows

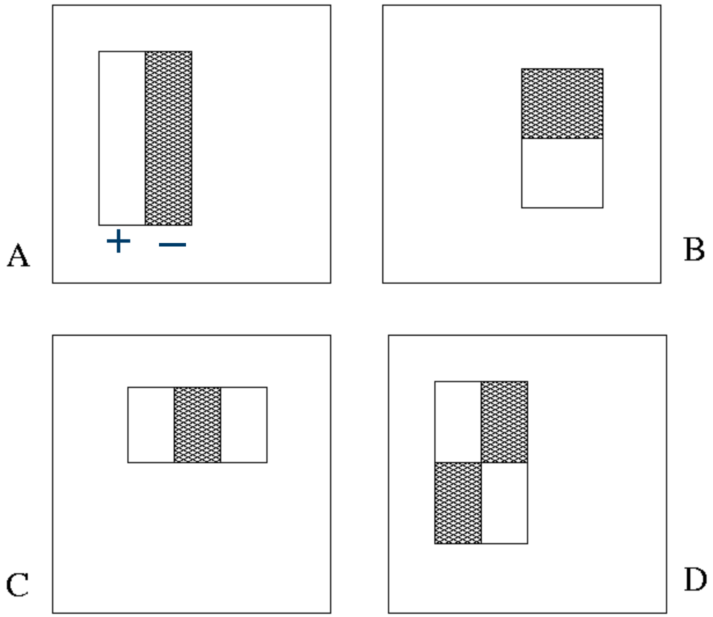
The Viola–Jones face detector

- Fast and reliable detection through the use of:
  - Integral image for fast feature computation
  - Boosting for feature selection and classifier training
  - Attentional cascade for fast rejection of most non-faces
- Slow training
- Lots of labeled training data required

P Viola, M Jones. Rapid Object Detection using a Boosted Cascade of Simple Features (2001).
P Viola, M Jones. *Robust real-time face detection.* IJCV 57(2), 2004.

Features

- Differences of pixel intensity sums
  - ‘Haar-like’ rectangular features capturing horizontal, vertical and diagonal image structures
  - Two, three and four adjacent rectangles of similar sizes
  - Value = Σ (intensity in white rect.) – Σ (intensity in grey rect.)
- Computed at various positions and scales within detection window



Integral image

- For an image, *i(x,y)*, the *integral image, ii(x,y),* contains the sum of the intensity values above and to the left of pixel (x,y), inclusive:

$$ii(x,y) = \sum_{x' \le x,\, y' \le y} i(x', y')$$

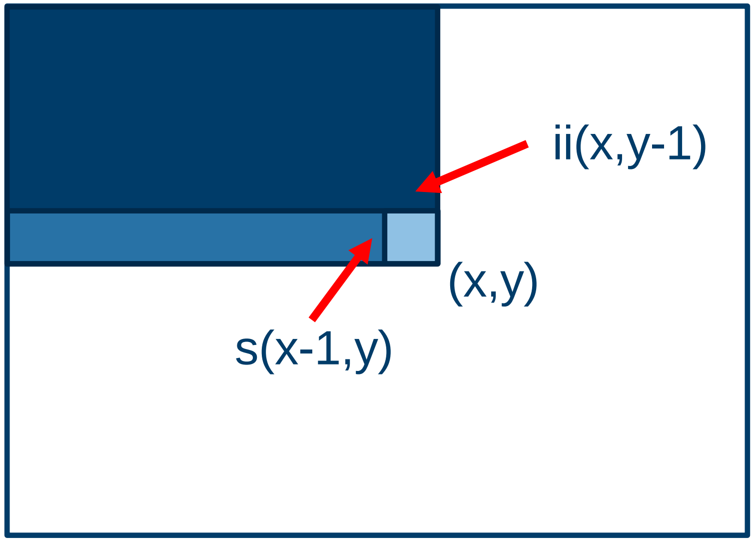
- Computed in single pass over the image *i(x,y)*:

  Cumulative row sum:

$$s(x,y) = s(x-1,y) + i(x,y)$$

  Integral image:

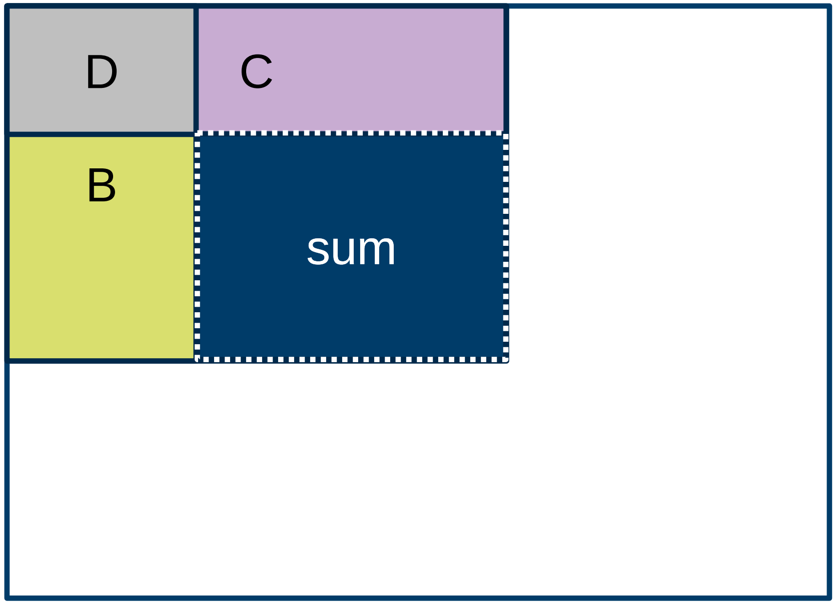
$$ii(x,y) = ii(x,y-1) + s(x,y)$$



Feature computation

- The intensity sum within an image rectangle:

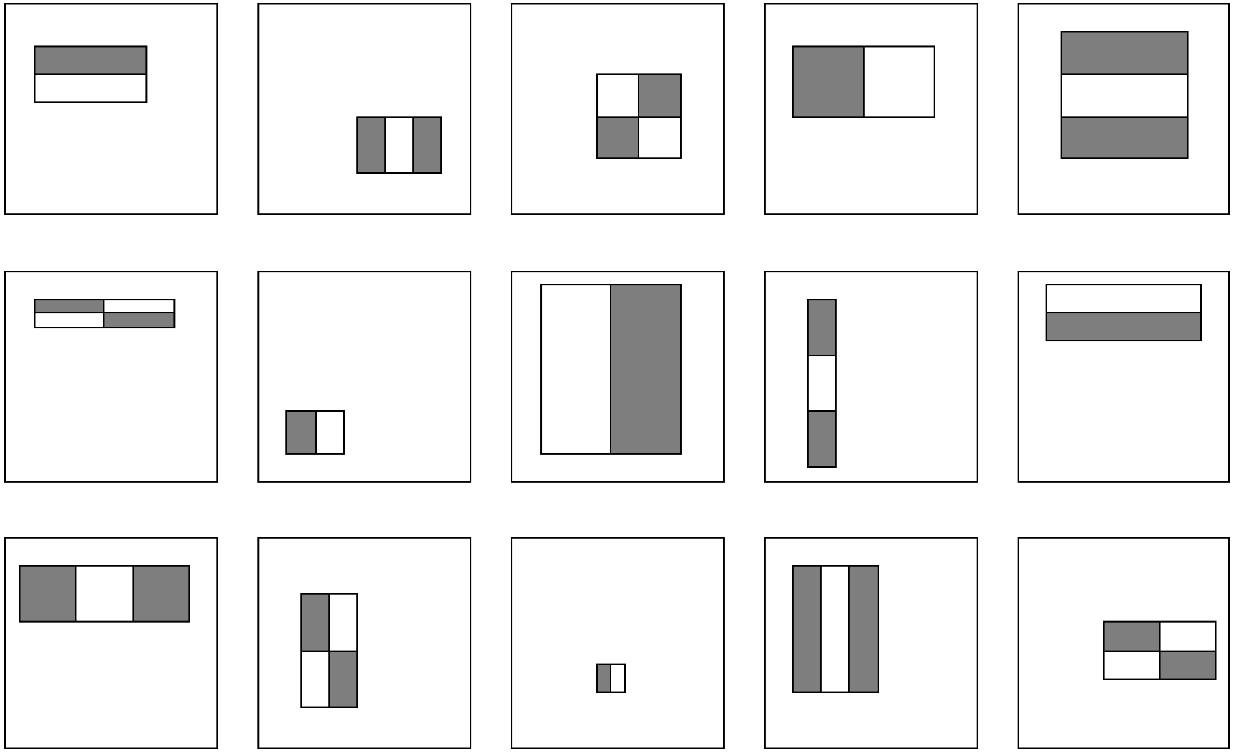
  sum = A – B – C + D
- Calculated from four integral image references regardless of rectangle size!
- Features consisting of 2, 3 and 4 rectangles are calculated from 6, 8 and 9 array references, respectively



- To detect objects at various scales, resize the features, not the image!

Feature selection

- For a 24x24 pixels detection window, the number of possible rectangle features is ~160,000!
- Need to select a tractable subset of discriminative features



Boosting

- Build a strong classifier by combining many “weak classifiers”, which need only be better than chance
- Initially, all training examples have equal weights
- In each boosting iteration:
  - Find the weak learner that achieves the lowest *weighted* training error
  - Raise the weights of training examples misclassified by current weak learner
- The final classifier is a weighted sum of the selected weak classifiers, with weights proportional to their achieved accuracy
- Exact formulas for re-weighting and combining weak learners depend on the particular boosting scheme (e.g., AdaBoost)

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

Boosting

- For weak learners based on single rectangle features, each classifier h_t is defined by: feature, parity (±1) and threshold

feature value

$$h_t(x) = \begin{cases} 1 & \text{if } p_t f_t(x) > p_t \theta_t \\ 0 & \text{otherwise} \end{cases}$$

sub-window parity threshold

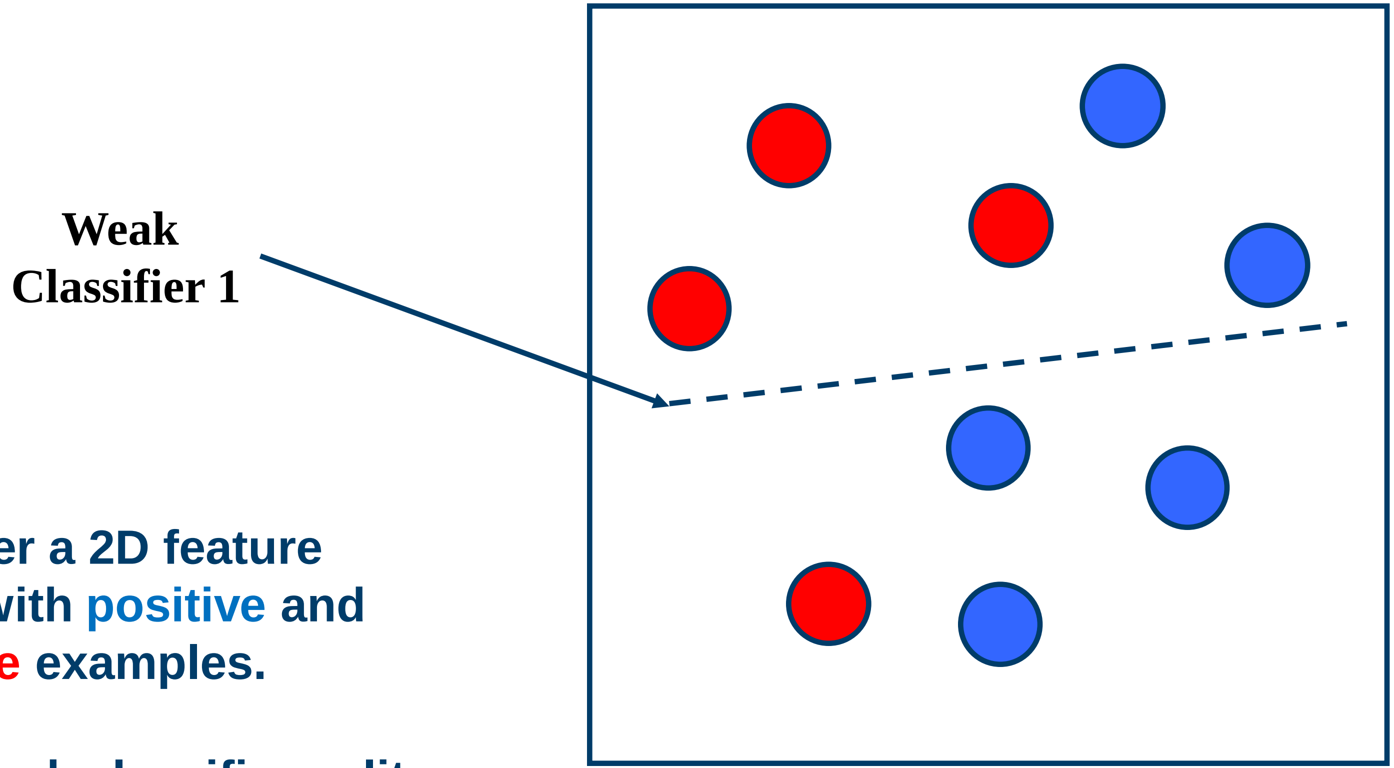
- Ensemble classifier is a weighted sum of weak classifiers:

num. iterations

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

learned weights

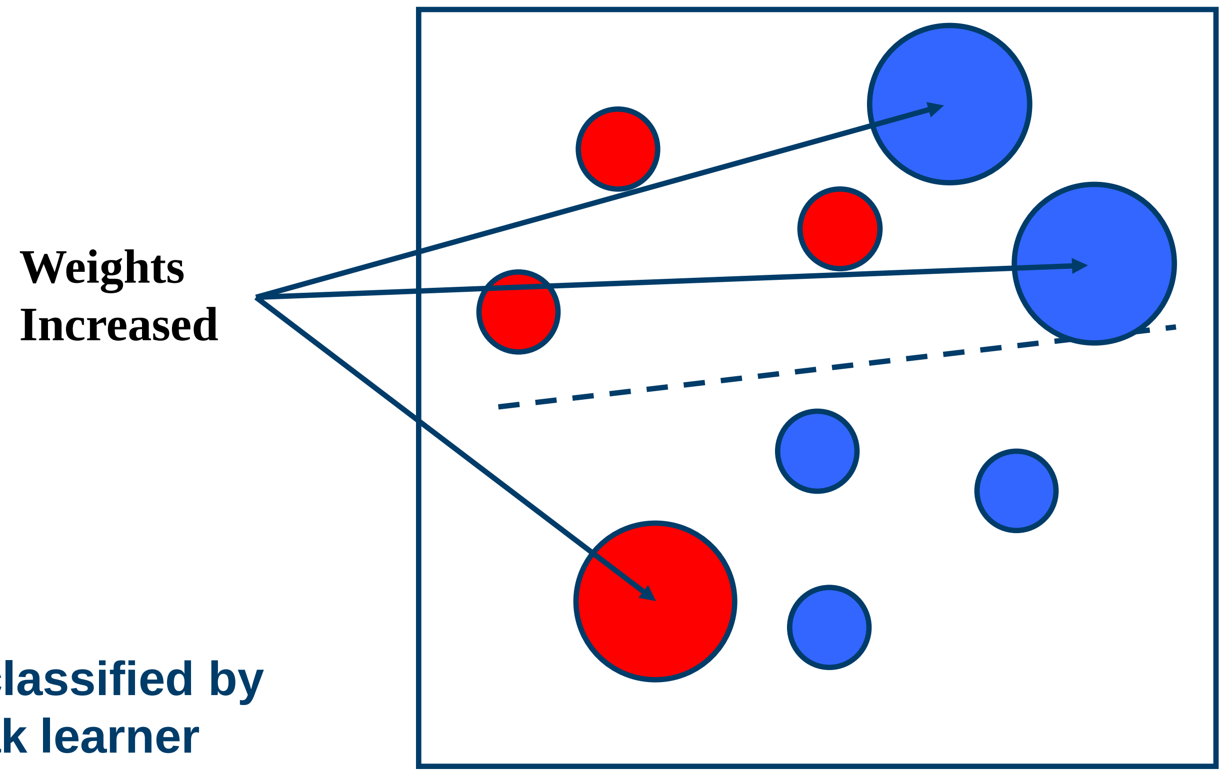
Boosting illustration



Consider a 2D feature space with positive and negative examples.

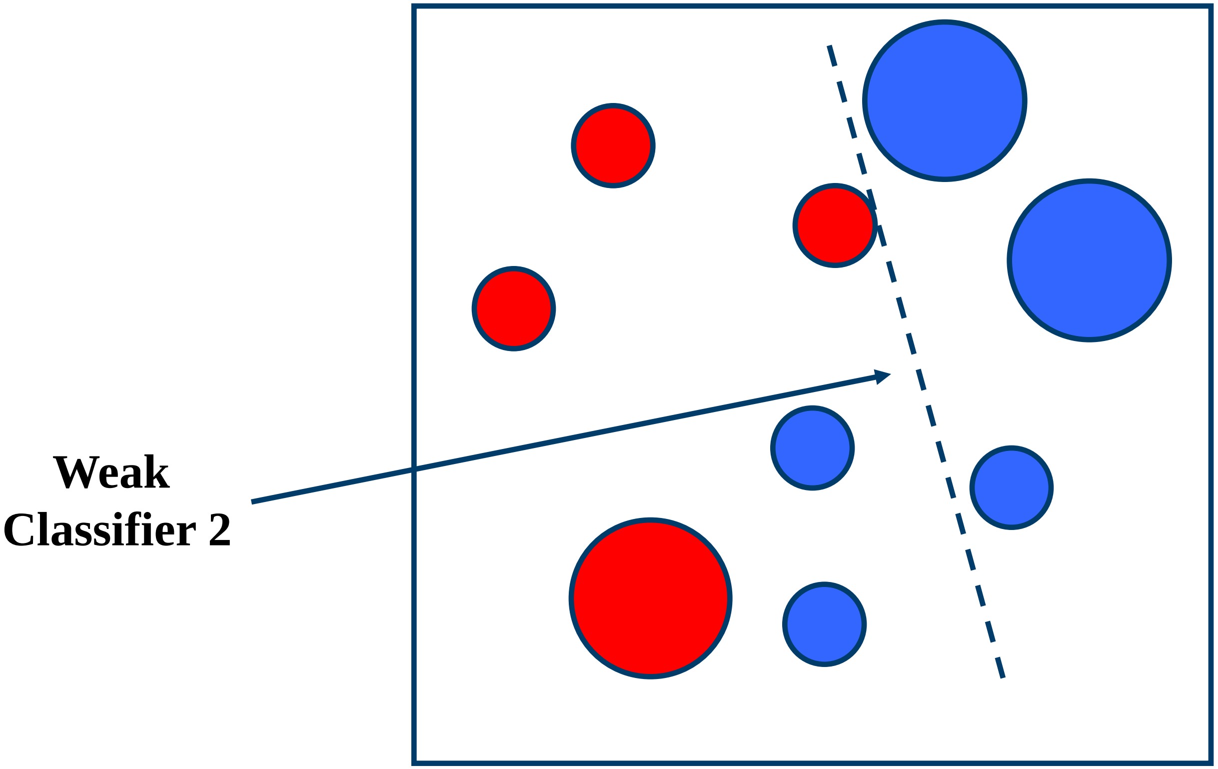
Each weak classifier splits the training examples with at least 50% accuracy.

Boosting illustration

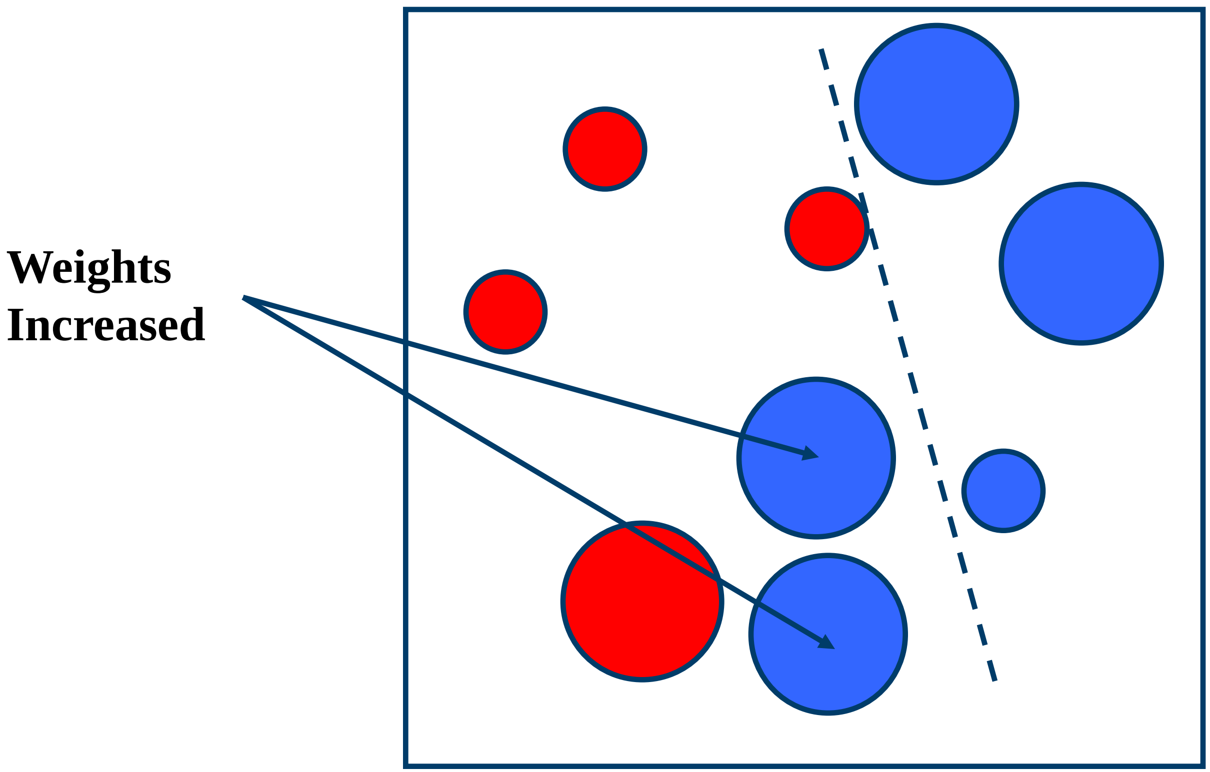


Examples misclassified by a previous weak learner are given more emphasis at future rounds.

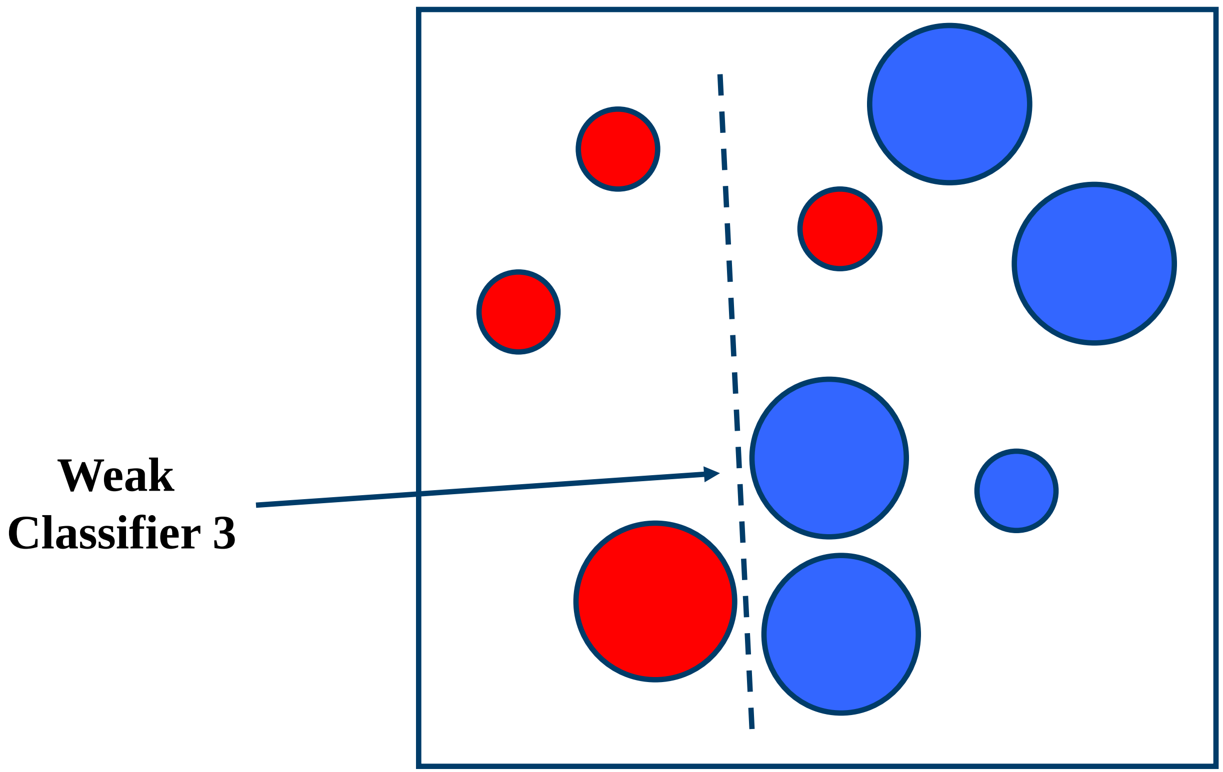
Boosting illustration



Boosting illustration

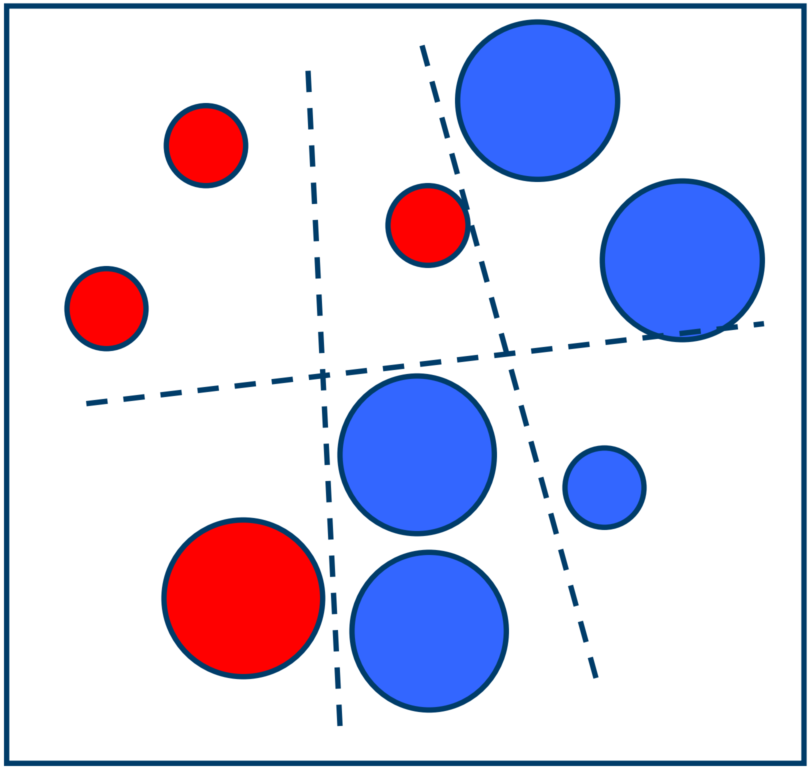


Boosting illustration



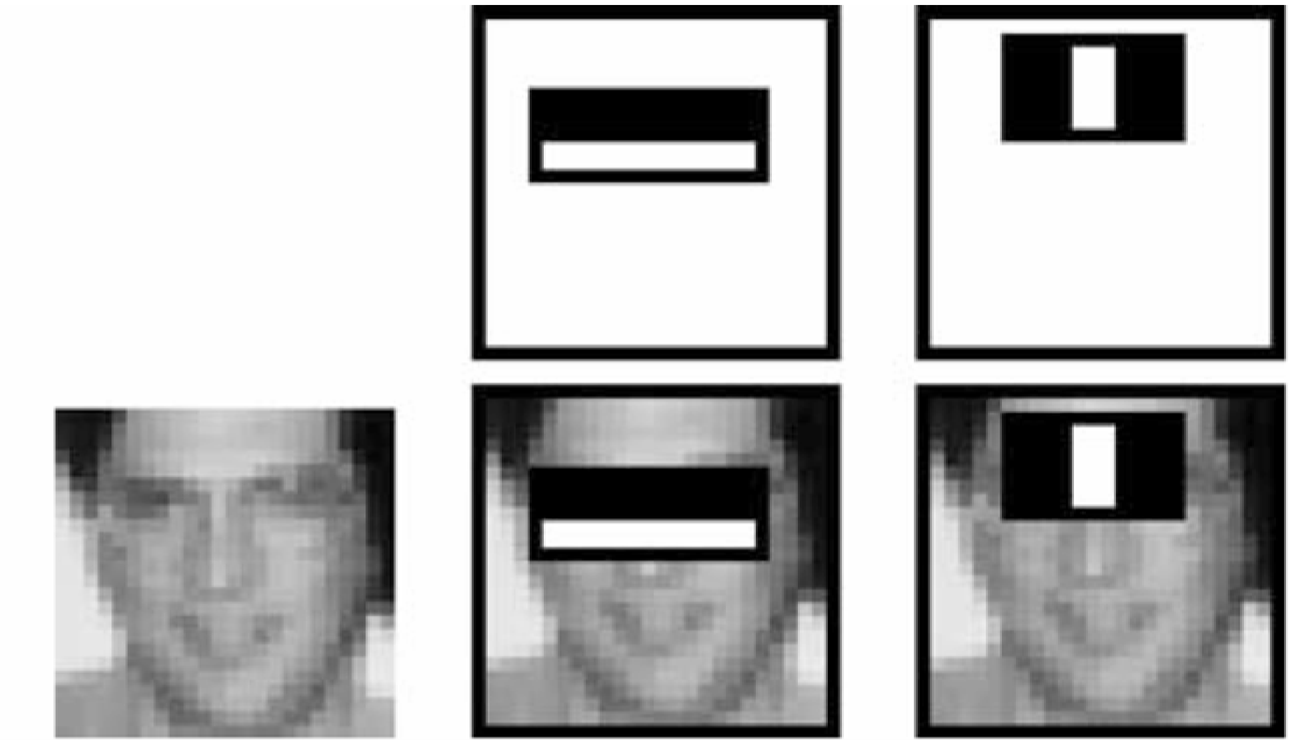
Boosting illustration

Final classifier is
a combination of weak
classifiers



Viola-Jones detector results

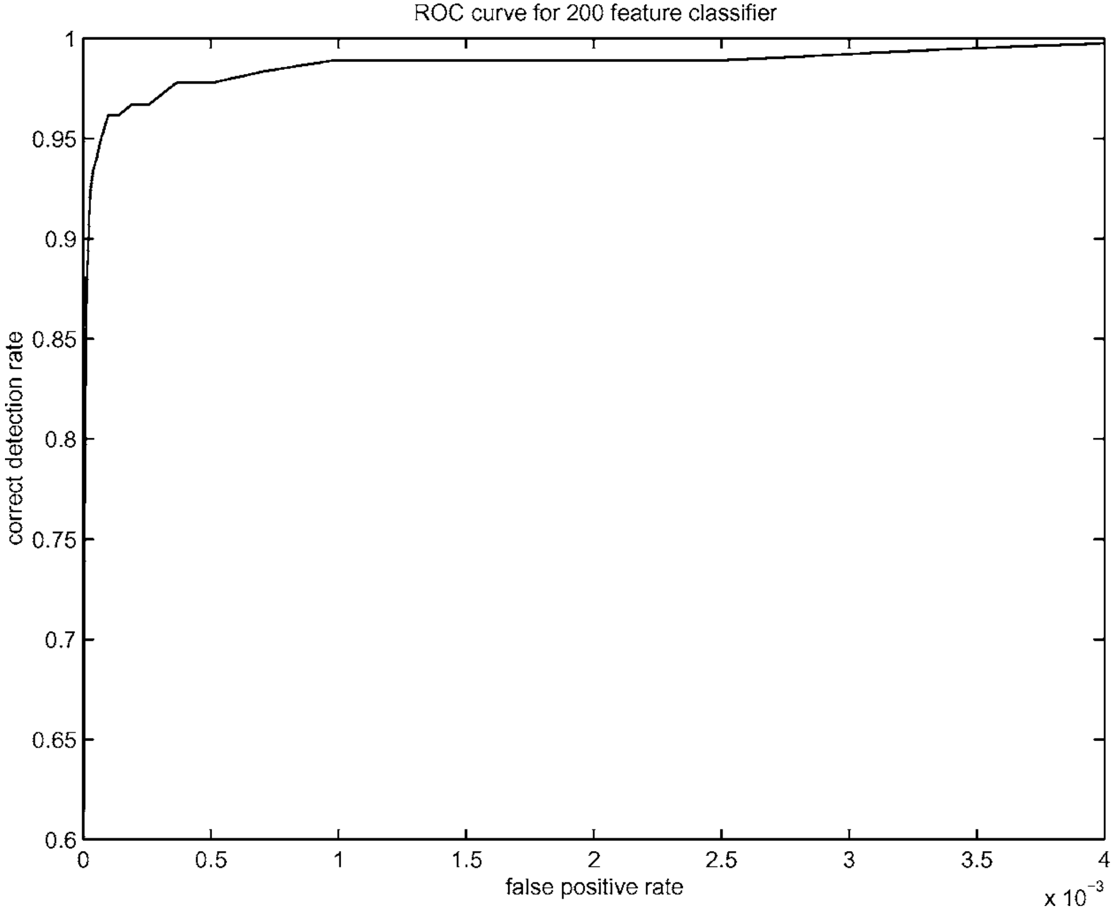
- First two features selected by boosting:



- This feature combination yielded 100% detection rate and 50% false positive rate

Boosting for face detection

- A 200-feature classifier yielded 95% detection rate and a false positive rate of 1 in 14084 (~$70 \cdot 10^{-6}$)
- Reduced false positive rate requires more features, but that increases processing time!

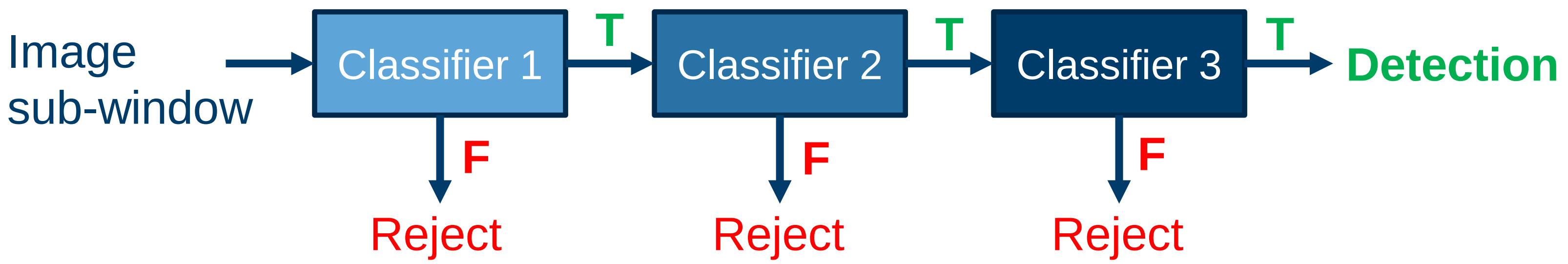


Receiver operating characteristic (ROC) curve

Attentional cascade

- The cascade starts with simple classifiers which reject many of the non-face sub-windows while detecting almost all faces
- Positive response from the first classifier triggers the evaluation of the second (more complex and accurate) classifier, and so on
- A negative outcome at any stage leads to the immediate rejection of the sub-window =>
  - True positive rates must be close to 1 for each stage
  - False positive rates can be moderate for each stage

    Example 10 stages cascade: $(0.99)^{10} \sim 0.9$, $(0.25)^{10} \sim 10^{-6}$



Training the cascade

- For each stage, specify:
  - Minimum acceptable target detection rate
  - Maximum acceptable false positive rate
- Add features to the current stage until its specified rates have been met
  - Need to lower classifier threshold to increase detection rate (as opposed to minimizing total classification error)
  - Test on a *validation set*
- Use false positives from current stage as the non-target training examples for the next stage
- If the overall false positive rate is too high, then add another stage

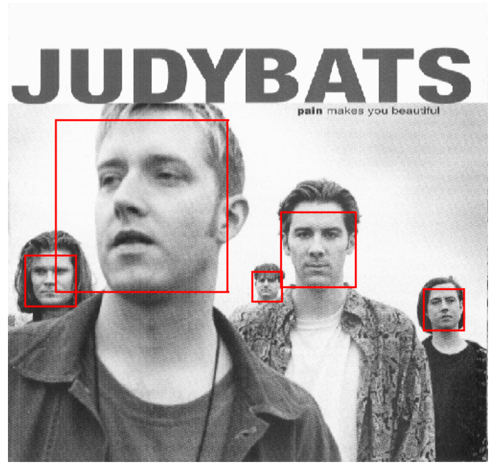
Training data

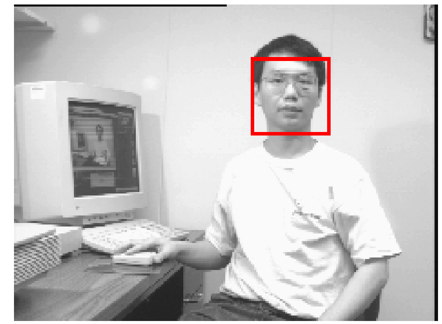
- 5000 faces
  - All frontal, rescaled to 24x24 pixels
  - 9500 non-face images
    - 350 million non-faces
  - Faces are normalized
    - Scale, translation
- Many variations
  - Across individuals
  - Illumination
  - Pose

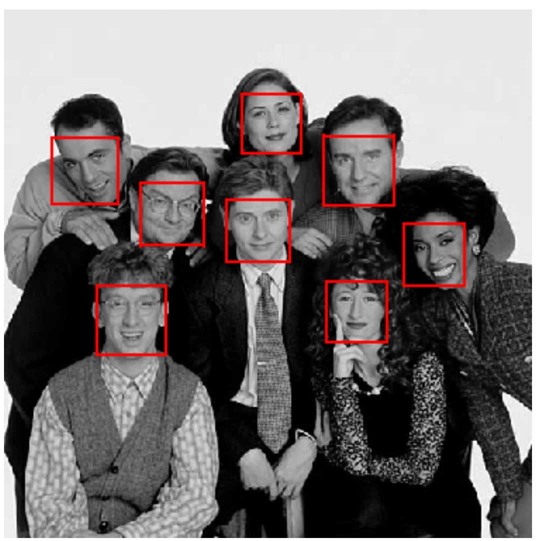
The implemented detector cascade

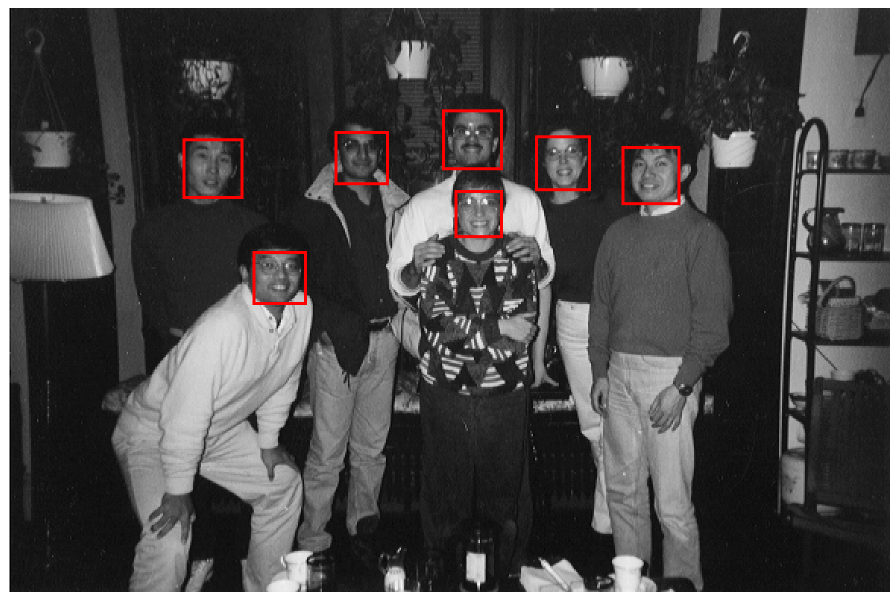
- 38 stages (classifiers)
  - 2, 10, 25, 25, 50, 50, 50, 75, 100,.., 200 features per stage
  - 6061 total used out of 160k candidate features
  - 8 features evaluated on average per sub-window
- Filter scanning
  - Scale detector rather than image (12 scales)
  - Scale steps = 1.25 (factor between two consecutive scales)
  - Translation 1*scale (# pixels between two consecutive windows)
- Non-max suppression: average coordinates of overlapping detections

Output of Face Detector on Test Images

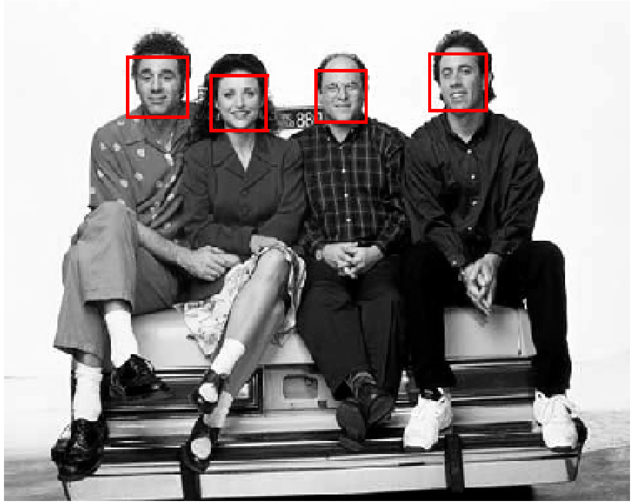


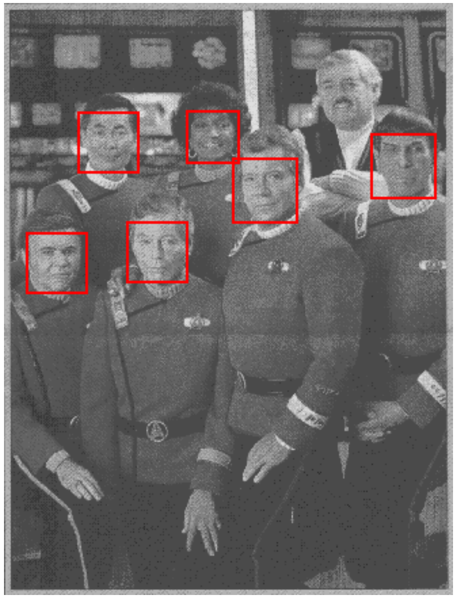


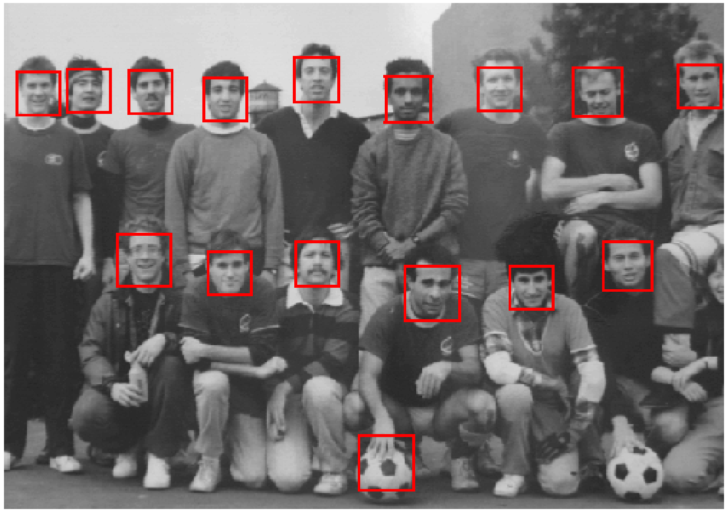




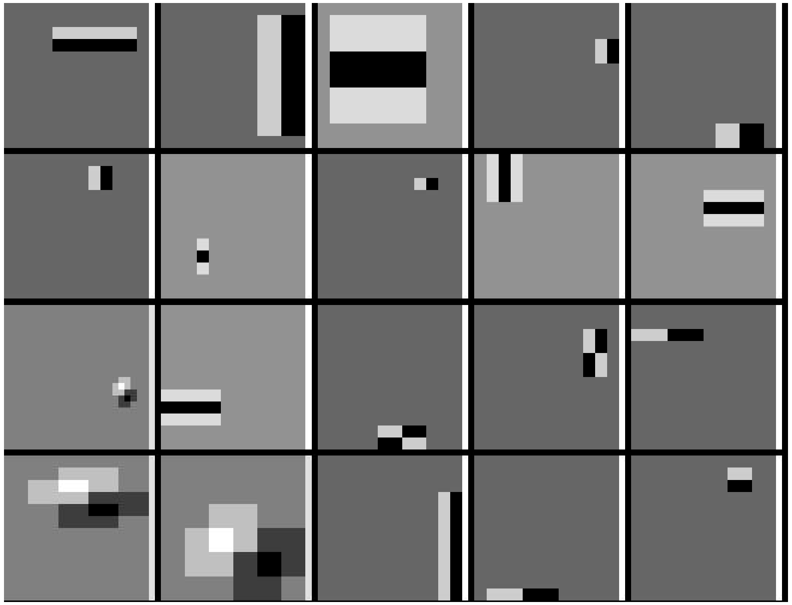
Output of Face Detector on Test Images





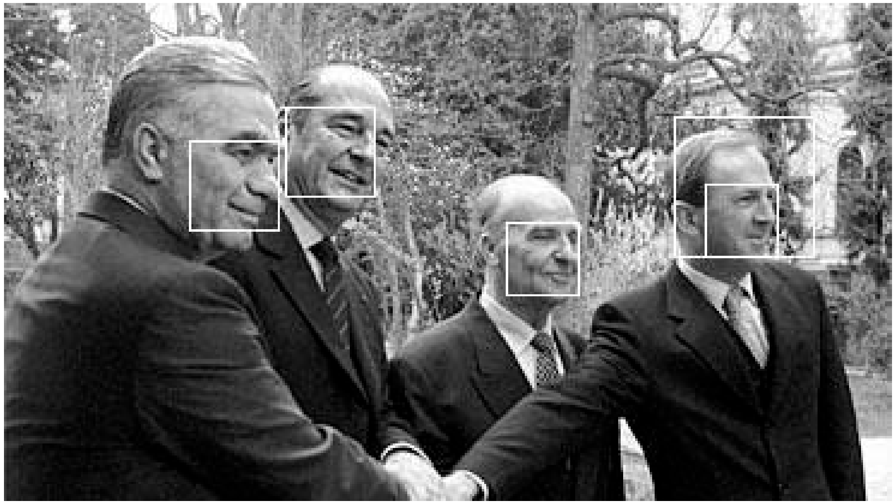


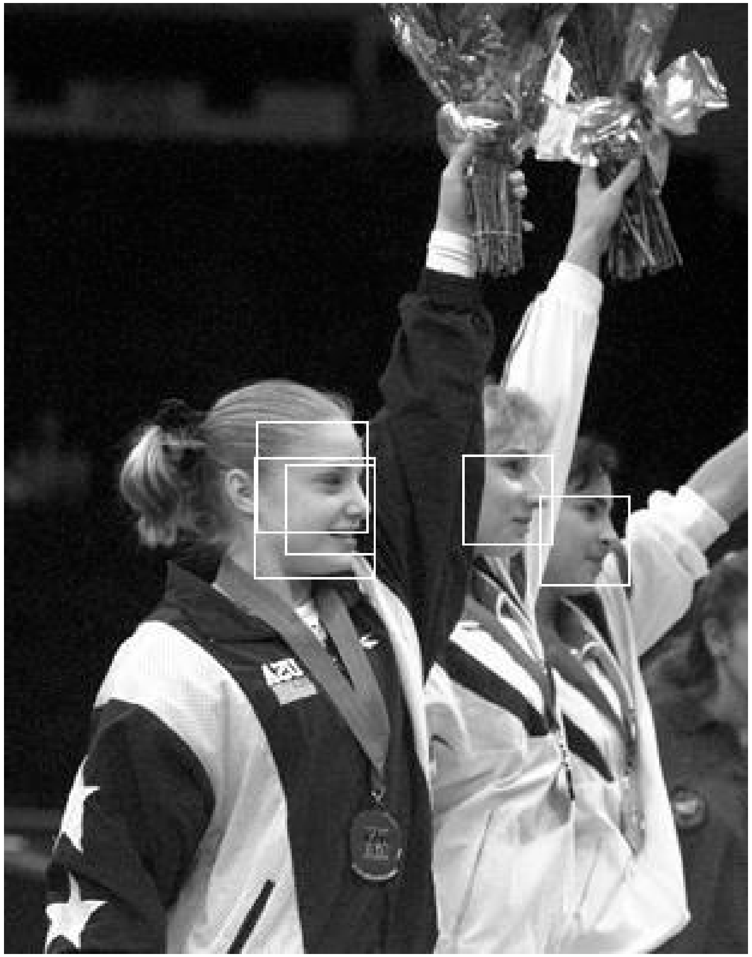
Profile detection



New features and training on profile examples required!

Profile detector results





Subspace processing

- Eigenpictures
- Eigenfaces
- Fisherfaces
- ..

Principal Component Analysis (PCA)

- Start with *N* training face images: $\left\{\underline{x}_j\right\}_{j=0}^{N-1}, \quad \underline{x}_j \in R^P$
- The mean image, *m*, and covariance matrix, *C*, of training set:

$$\underline{m} = \sum_{j=0}^{N} \underline{x}_j \qquad C = \frac{1}{N}\sum_{j=0}^{N-1}(\underline{x}_j - \underline{m})(\underline{x}_j - \underline{m})^T$$

- Eigenvalue decomposition gives the representation:

$$C = U\Lambda U^T = \sum_{i=0}^{N-1} \lambda_i \underline{u}_i \underline{u}_i^T$$

i'th eigenvalue

i'th eigenvector

- With the eigenvalues λ_i sorted in decreasing order, any image *x* can be approximated from its projection onto the *M*<*N* first eigenvectors:

$$\underline{x} \approx \underline{m} + \sum_{i=0}^{M-1} a_i \underline{u}_i \qquad a_i = (\underline{x} - \underline{m}) \cdot \underline{u}_i$$

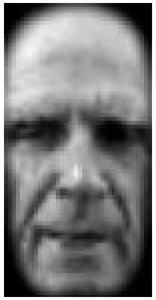
i'th principal component

Eigenfaces representation

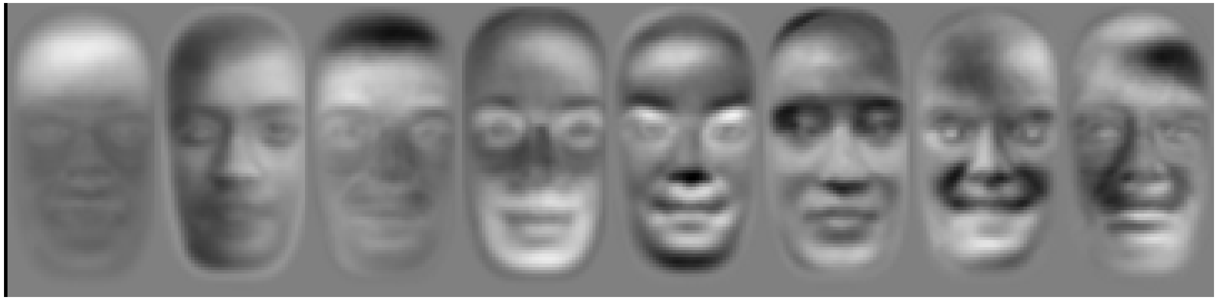
- The eigenfaces subset $u_1,..,u_M$ span the **Face Space**
- Any face image x can be represented in face space coordinates:

$$\underline{x} \approx \underline{m} + a_1\underline{u}_1 + a_2\underline{u}_2 + .. + a_M\underline{u}_M$$

Input image



First 8 eigenfaces



Reconstructed image

JPEG

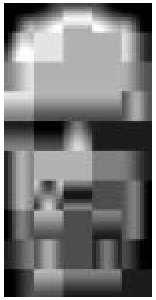


Image compressed with 8 eigenfaces (85 bytes) and with JPEG (530 bytes)
=> The eigenfaces constitute a compact basis for face image representation

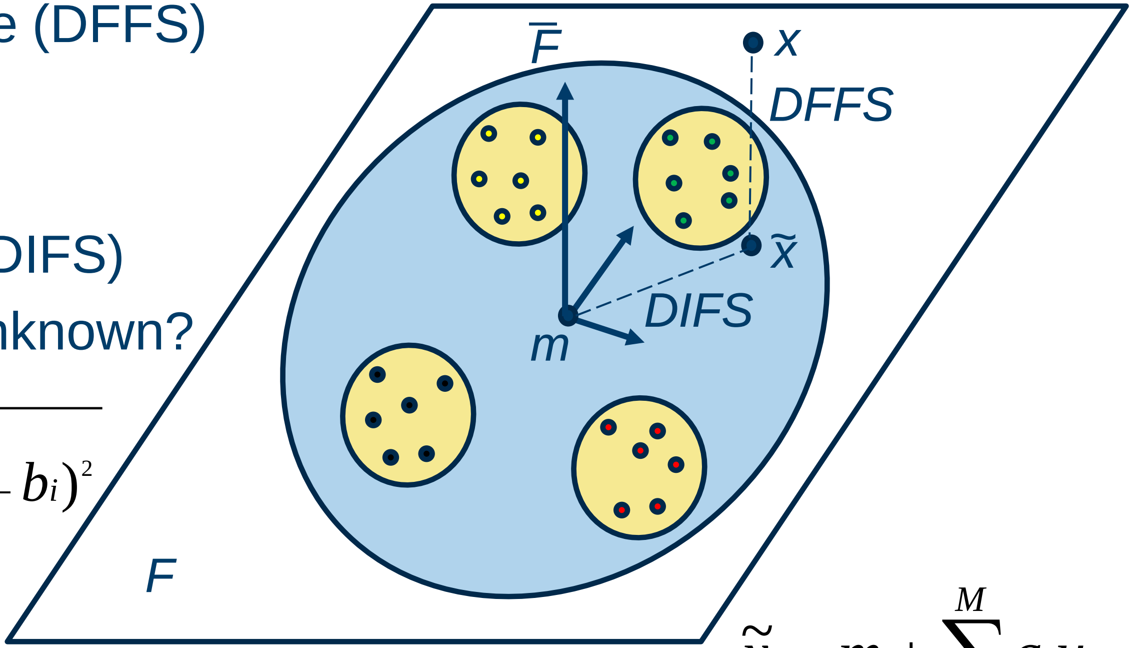
M Kirby, L Sirovich. Application of the karhunen-loeve procedure for the characterization of human faces PAMI 12(1), 1990.

M Turk, A Pentland. Eigenfaces for recognition. Journal of Cognitive Neuroscience 3(1) 1991.

Face space projection

- Face space: linear subspace spanned by first *M* eigenfaces
- Distance from face space (DFFS)
  - Is image *x* a face?
- Distance in face space (DIFS)
  - Which face is *x*, or unknown?

$$DIFS(\underline{x},\underline{y}) = \left\|\underline{\tilde{x}} - \underline{\tilde{y}}\right\| = \sqrt{\sum_{i=0}^{M-1}(a_i - b_i)^2}$$



- Mahalanobis distance:

$$DIFS'(\underline{x},\underline{y}) = \left\|\underline{\tilde{x}} - \underline{\tilde{y}}\right\|_{C^{-1}} = \sqrt{\sum_{i=0}^{M-1}(a_i - b_i)^2 / \lambda_i^2}$$

$$\underline{\tilde{x}} = \underline{m} + \sum_{i=0}^{M} a_i \underline{u}_i$$

$$a_i = (\underline{x} - \underline{m}) \cdot \underline{u}_i$$

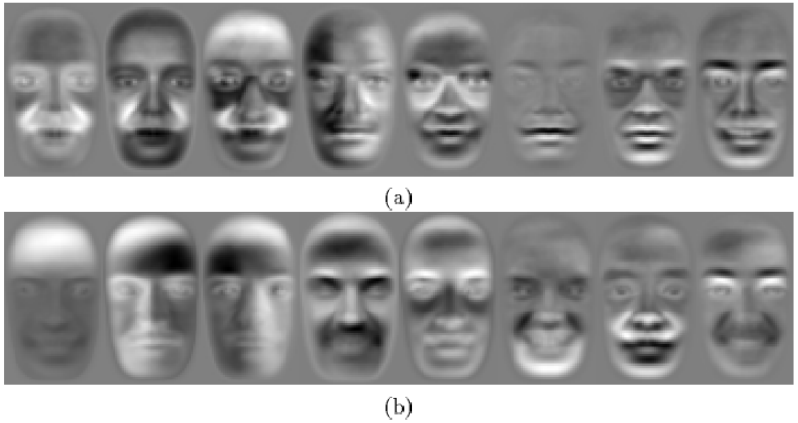
Eigenfaces limitations

- Images must be accurately scaled and aligned
- PCA identifies the directions of largest scatter in feature space, but fails to distinguish between **intra-class** and **extra-class** variability!
- Intra-class variations due to:
  - Illumination
  - Facial expressions
  - Pose
  - Glasses, hats, hair style,..

Images from Yale Face Database

Beyond single-set eigenfaces

- Fisher's linear discriminant (Fisherfaces)
  - Select directions that maximize the ratio of between-class and within-class variations
- Bayesian techniques
  - Intra-personal eigenfaces
  - Extra-personal eigenfaces
- View-based eigenspaces
  - Eigenspaces for various views
- Modular eigenspaces
  - Separate eigenspaces for eyes, nose, mouth, etc



P N Belhumeur, J P Hespanha, D L Kriegman. Eigenfaces vs Fisherfaces: Recognition using class specific linear projection. PAMI 19(7),1997.
B Moghaddam, A Pentland. Probabilistic visual learning for object representation. PAMI 1997.
B Moghaddam, T Jebara, A Pentland. Bayesian face recognition. Pattern Recognition, 2000.

Summary

- Face detection
  - Viola-Jones' sliding window detector
    - Rectangle features from integral image
    - Boosting for feature selection and training
    - Classifier cascade for fast processing
- Face recognition
  - Subspace processing
    - Eigenfaces
    - Fisherfaces
    - Bayesian methods
    - Modular / view-based eigenspaces

- Read also: Szeliski 14.1 and 14.2