Recent Advances in Face Detection

Ming-Hsuan Yang

myang@honda-ri.com

http://www.honda-ri.com http://vision.ai.uiuc.edu/mhyang

Honda Research Institute

Mountain View, California, USA

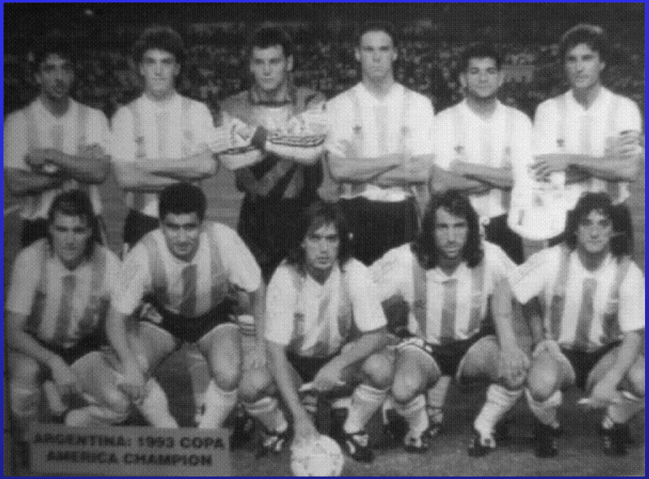
Outline

- Objective
  - Survey major face detection works
  - Address “how” and “why” questions
  - Pros and cons of detection methods
  - Future research directions
- Updated tutorial material

  http://vision.ai.uiuc.edu/mhyang/face-detection-survey.html

Face Detection

- Identify and locate human faces in an image regardless of their
  - position
  - scale
  - in-plane rotation
  - orientation
  - Pose (out-of-plane rotation)
  - and illumination



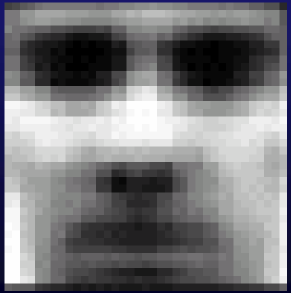
Where are the faces, if any?

Why Face Detection is Important?

- The first step for any automatic face recognition system
- First step in many surveillance systems
- Face is a highly non-rigid object
- A step towards Automatic Target Recognition (ATR) or generic object detection/recognition

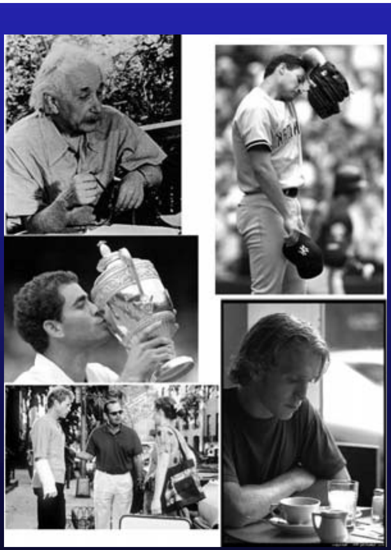
In One Thumbnail Face Image

- Consider a thumbnail 19 × 19 (361 pixels) face pattern
- 256^{361} possible combination of gray values
  - $256^{361} = 2^{8\times 361} = 2^{2888}$
- Total world population (as of 2000)
  - $6{,}000{,}000{,}000 \cong 2^{32}$
- About 90 times more than the world population



Why Face Detection Is Difficult?

- **Pose (Out-of-Plane Rotation)**: frontal, 45 degree, profile, upside down
- **Presence or absence of structural components**: beards, mustaches, and glasses
- **Facial expression**: face appearance is directly affected by a person's facial expression
- **Occlusion**: faces may be partially occluded by other objects
- **Orientation (In-Plane Rotation)**: face appearance directly vary for different rotations about the camera's optical axis
- **Imaging conditions**: lighting (spectra, source distribution and intensity) and camera characteristics (sensor response, gain control, lenses), resolution



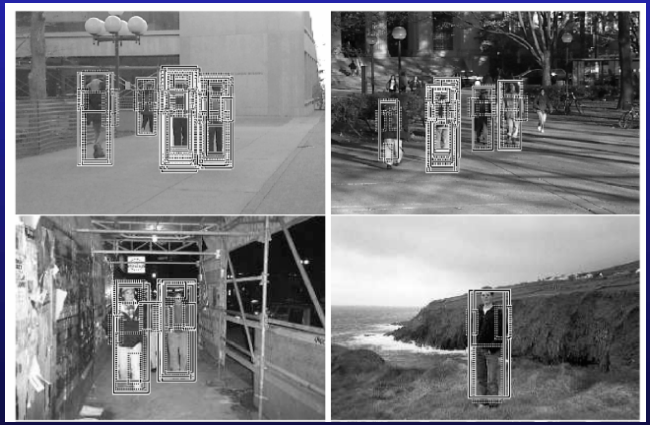
Related Problems

- Face localization:
  - Aim to determine the image position of a single face
  - A simplified detection problem with the assumption that an input image contains only one face
- Facial feature extraction:
  - To detect the presence and location of features such as eyes, nose, nostrils, eyebrow, mouth, lips, ears, etc
  - Usually assume that there is only one face in an image
- Face recognition (identification)
- Facial expression recognition
- Face tracking

Face Detection and Object Recognition

- Detection: concerns with a *category* of object
- Recognition: concerns with *individual* identity
- Face is a highly non-rigid object
- Many methods can be applied to other object detection/recognition

Car detection



Pedestrian detection

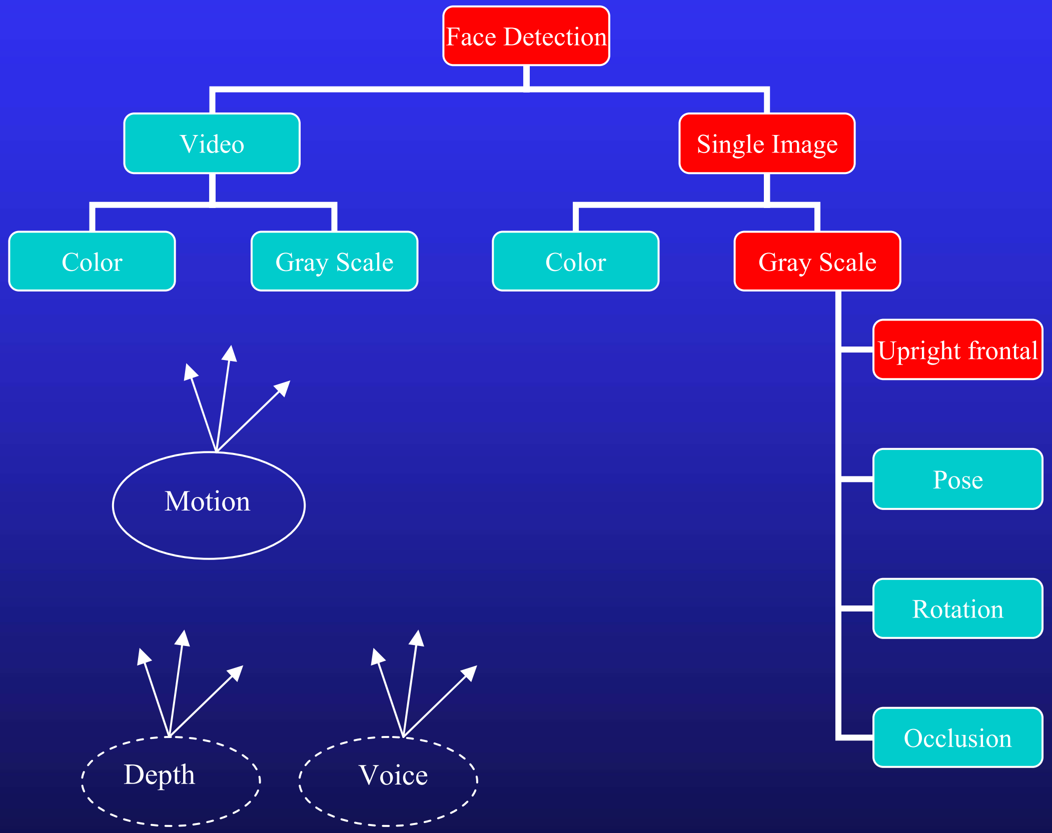
Research Issues

- Representation: How to describe a typical face?
- Scale: How to deal with face of different size?
- Search strategy: How to spot these faces?
- Speed: How to speed up the process?
- Precision: How to locate the faces precisely?
- Post processing: How to combine detection results?

Face Detector: Ingredients

- Target application domain: single image, video
- Representation: holistic, feature, holistic, etc
- Pre processing: histogram equalization, etc
- Cues: color, motion, depth, voice, etc
- Search strategy: exhaustive, greedy, focus of attention, etc
- Classifier design: ensemble, cascade
- Post processing: combing detection results

In This Tutorial



- Focus on detecting
  - upright, frontal faces
  - in a single gray-scale image
  - with decent resolution
  - under good lighting conditions
- See [Sinha 01] for detecting faces in low-resolution images

Methods to Detect/Locate Faces

- Knowledge-based methods:
  - Encode human knowledge of what constitutes a typical face (usually, the relationships between facial features)
- Feature invariant approaches:
  - Aim to find structural features of a face that exist even when the pose, viewpoint, or lighting conditions vary
- Template matching methods:
  - Several standard patterns stored to describe the face as a whole or the facial features separately
- Appearance-based methods:
  - The models (or templates) are learned from a set of training images which capture the representative variability of facial appearance

Many methods can be categorized in several ways

Agenda

- Detecting faces in gray scale images
  - Knowledge-based
  - Feature-based
  - Template-based
  - Appearance-based
- Detecting faces in color images
- Detecting faces in video
- Performance evaluation
- Research direction and concluding remarks

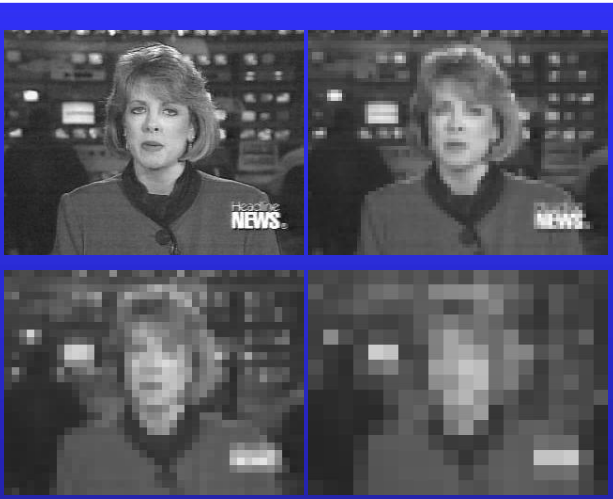
Knowledge-Based Methods

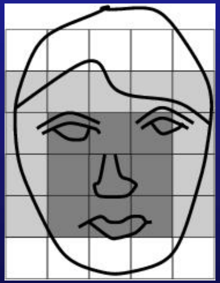
- Top-down approach: Represent a face using a set of human-coded rules
- Example:
  - The center part of face has uniform intensity values
  - The difference between the average intensity values of the center part and the upper part is significant
  - A face often appears with two eyes that are symmetric to each other, a nose and a mouth
- Use these rules to guide the search process

Knowledge-Based Method: [Yang and Huang 94]

- Multi-resolution focus-of-attention approach
- Level 1 (lowest resolution):

  apply the rule “the center part of the face has 4 cells with a basically uniform intensity” to search for candidates
- Level 2: local histogram equalization followed by edge detection
- Level 3: search for eye and mouth features for validation





Knowledge-Based Method: [Kotropoulos & Pitas 94]

- Horizontal/vertical projection to search for candidates

$$HI(x)=\sum_{y=1}^{n} I(x,y) \qquad VI(y)=\sum_{x=1}^{m} I(x,y)$$

- Search eyebrow/eyes, nostrils/nose for validation
- Difficult to detect multiple people or in complex background

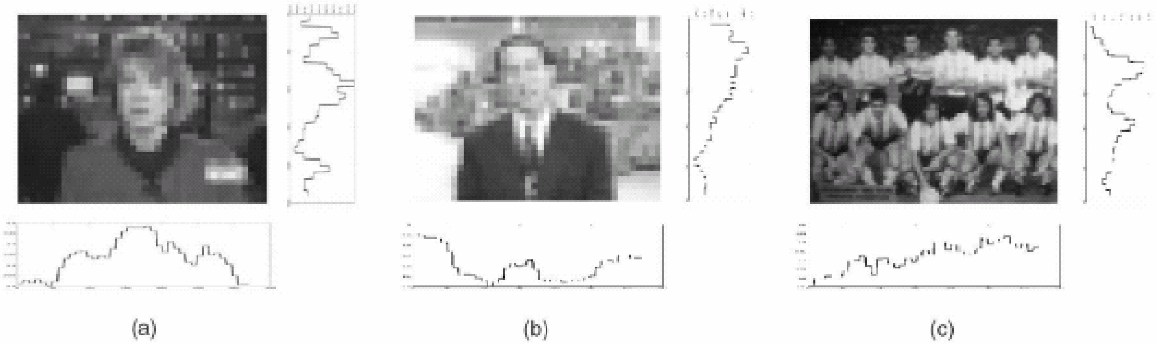


Fig. 3. (a) and (b) $n = 8$. (c) $n = 4$. Horizontal and vertical profiles. It is feasible to detect a single face by searching for the peaks in horizontal and vertical profiles. However, the same method has difficulty detecting faces in complex backgrounds or multiple faces as shown in (b) and (c).

[Kotropoulos & Pitas 94]

Knowledge-based Methods: Summary

- Pros:
  - Easy to come up with simple rules to describe the features of a face and their relationships
  - Based on the coded rules, facial features in an input image are extracted first, and face candidates are identified
  - Work well for face localization in uncluttered background
- Cons:
  - Difficult to translate human knowledge into rules precisely: detailed rules fail to detect faces and general rules may find many false positives
  - Difficult to extend this approach to detect faces in different poses: implausible to enumerate all the possible cases

Agenda

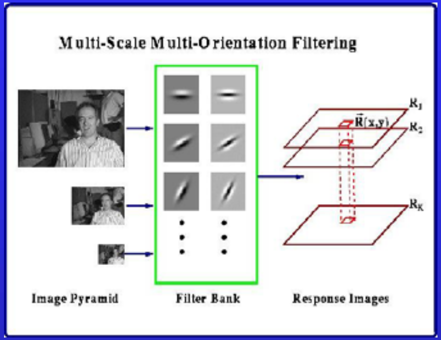
- Detecting faces in gray scale images
    - Knowledge-based
    - Feature-based
    - Template-based
    - Appearance-based
- Detecting faces in color images
- Detecting faces in video
- Performance evaluation
- Research direction and concluding remarks

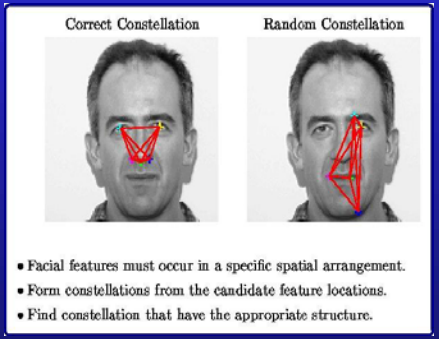
Feature-Based Methods

- Bottom-up approach: Detect facial features (eyes, nose, mouth, etc) first
- Facial features: edge, intensity, shape, texture, color, etc
- Aim to detect invariant features
- Group features into candidates and verify them

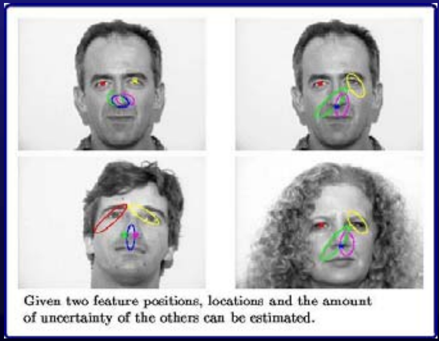
Random Graph Matching [Leung et al. 95]

- Formulate as a problem to find the correct geometric arrangement of facial features
- Facial features are defined by the average responses of multi-orientation, multi-scale Gaussian derivative filters
- Learn the configuration of features with Gaussian distribution of mutual distance between facial features
- Convolve an image with Gaussian filters to locate candidate features based on similarity
- Random graph matching among the candidates to locate faces





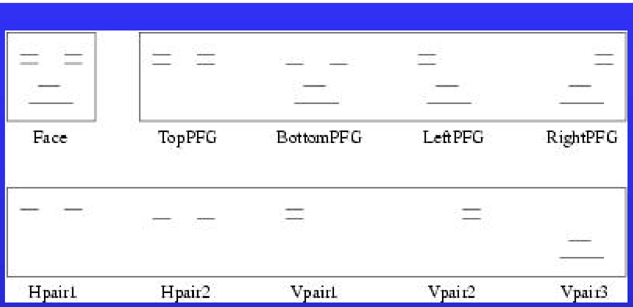
- Facial features must occur in a specific spatial arrangement.
- Form constellations from the candidate feature locations.
- Find constellation that have the appropriate structure.



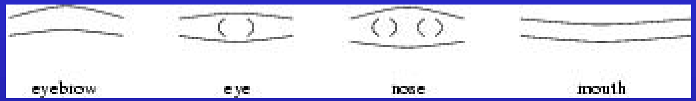
Given two feature positions, locations and the amount of uncertainty of the others can be estimated.

Feature Grouping [Yow and Cipolla 90]

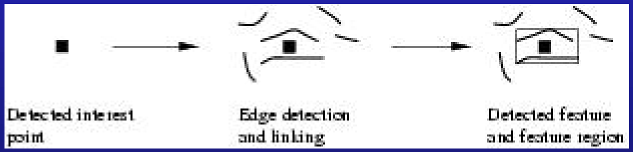
- Apply a 2nd derivative Gaussian filter to search for interest points
- Group the edges near interest points into regions
- Each feature and grouping is evaluated within a Bayesian network
- Handle a few poses
- See also [Amit et al. 97] for efficient hierarchical (focus of attention) feature-based method



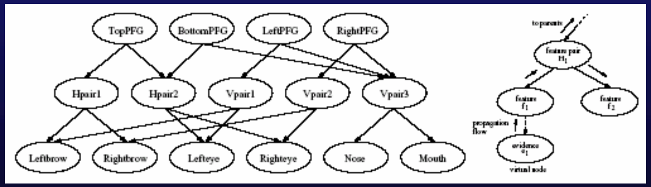
Face model and component



Model facial feature as pair of edges


Detected interest point
Edge detection and linking
Detected feature and feature region

Apply interest point operator and edge detector to search for features



Using Bayesian network to combine evidence

Feature-Based Methods: Summary

- Pros:
  - Features are invariant to pose and orientation change
- Cons:
  - Difficult to locate facial features due to several corruption (illumination, noise, occlusion)
  - Difficult to detect features in complex background

Agenda

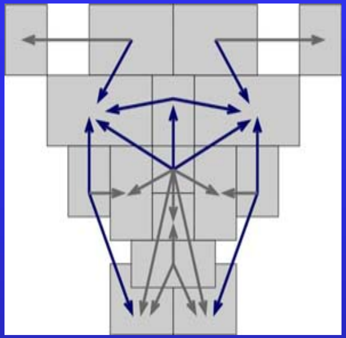
- Detecting faces in gray scale images
  - Knowledge-based
  - Feature-based
  - Template-based
  - Appearance-based
- Detecting faces in color images
- Detecting faces in video
- Performance evaluation
- Research direction and concluding remarks

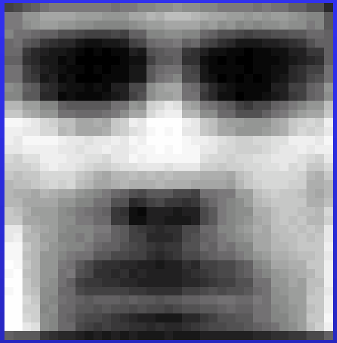
Template Matching Methods

- Store a template
  - Predefined: based on edges or regions
  - Deformable: based on facial contours (e.g., Snakes)
- Templates are hand-coded (not learned)
- Use correlation to locate faces

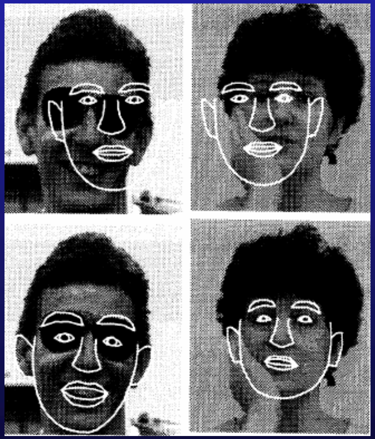
Face Template

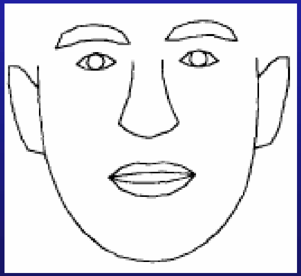
- Use relative pair-wise ratios of the brightness of facial regions (14 × 16 pixels): the eyes are usually darker than the surrounding face [Sinha 94]
- Use average area intensity values than absolute pixel values
- See also Point Distribution Model (PDM) [Lanitis et al. 95]





Ration Template [Sinha 94]





average shape

[Lanitis et al. 95]

Template-Based Methods: Summary

- Pros:
  - Simple
- Cons:
  - Templates needs to be initialized near the face images
  - Difficult to enumerate templates for different poses (similar to knowledge-based methods)

Agenda

- Detecting faces in gray scale images
  - Knowledge-based
  - Feature-based
  - Template-based
  - Appearance-based
- Detecting faces in color images
- Detecting faces in video
- Performance evaluation
- Research direction and concluding remarks

Appearance-Based Methods

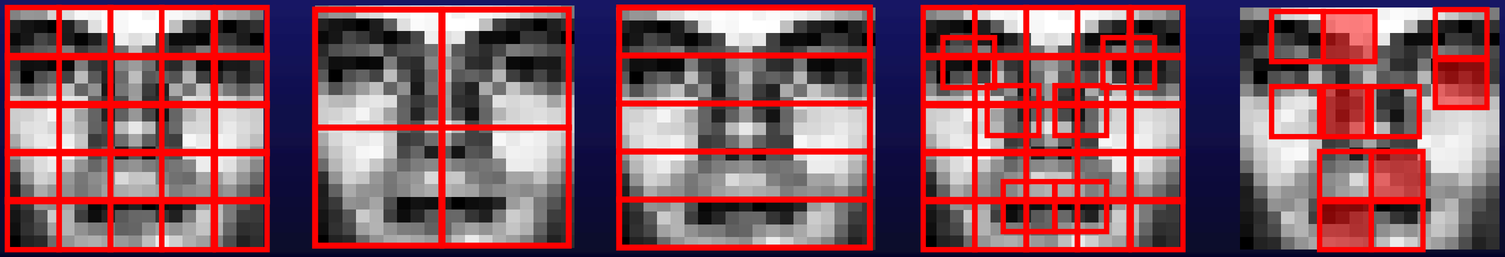
- Train a classifier using positive (and usually negative) examples of faces
- Representation
- Pre processing
- Train a classifier
- Search strategy
- Post processing
- View-based

Appearance-Based Methods: Classifiers

- Neural network: Multilayer Perceptrons
- Princiapl Component Analysis (PCA), Factor Analysis
- Support vector machine (SVM)
- Mixture of PCA, Mixture of factor analyzers
- Distribution-based method
- Naïve Bayes classifier
- Hidden Markov model
- Sparse network of winnows (SNoW)
- Kullback relative information
- Inductive learning: C4.5
- Adaboost
- …

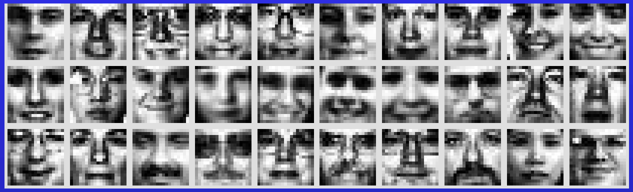
Representation

- Holistic: Each image is raster scanned and represented by a vector of intensity values
- Block-based: Decompose each face image into a set of overlapping or non-overlapping blocks
  - At multiple scale
  - Further processed with vector quantization, Principal Component Analysis, etc.

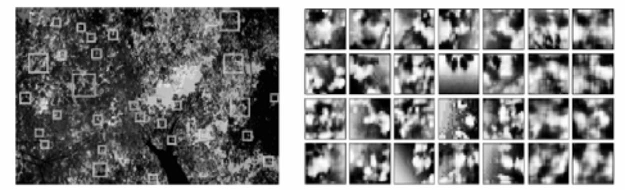


Face and Non-Face Exemplars

- Positive examples:
  - Get as much variation as possible
  - Manually crop and normalize each face image into a standard size (e.g., 19 × 19 pixels)
  - Creating virtual examples [Sung and Poggio 94]

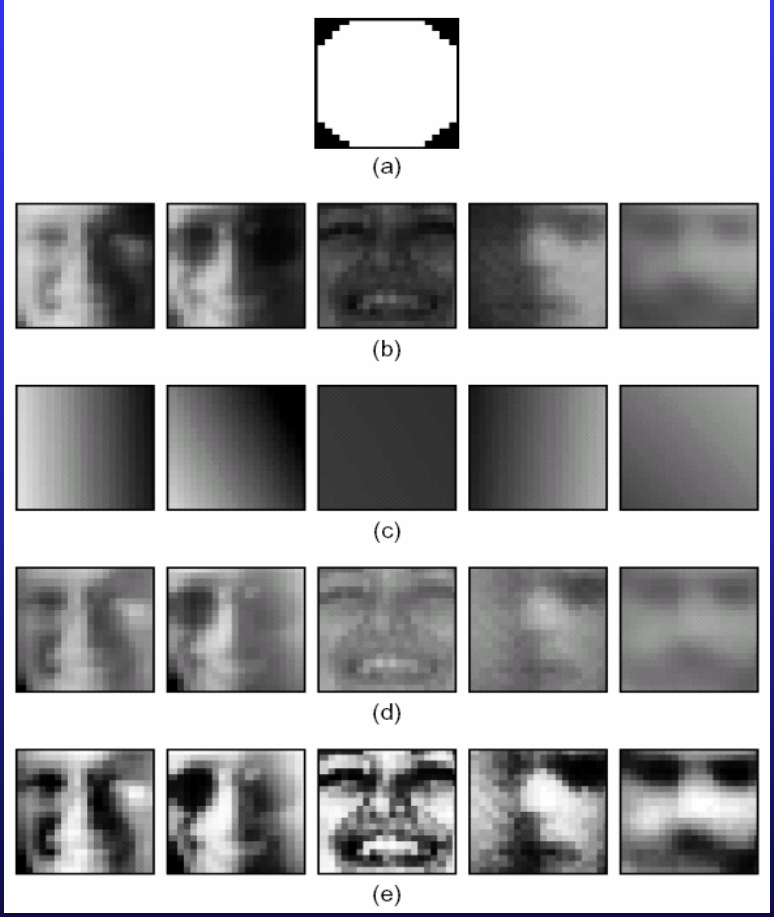


- Negative examples:
  - Fuzzy idea
  - Any images that do not contain faces
  - A large image subspace
  - Bootstraping [Sung and Poggio 94]



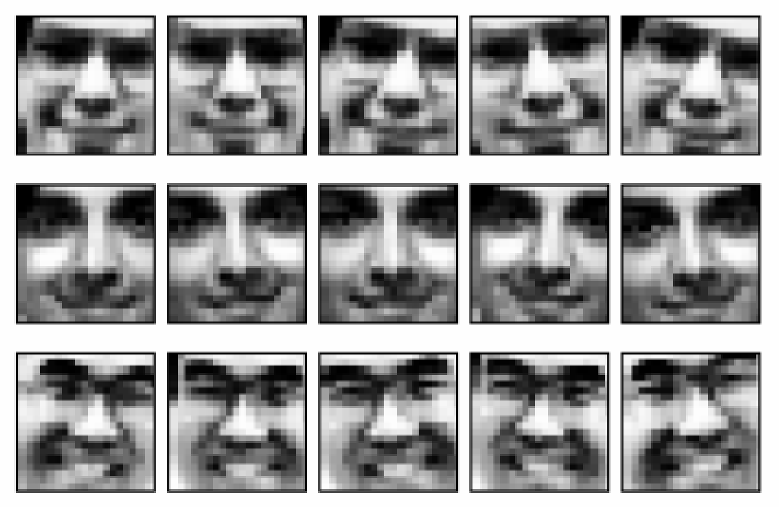
Distribution-Based Method [Sung & Poggio 94]

- Masking: reduce the unwanted background noise in a face pattern
- Illumination gradient correction: find the best fit brightness plane and then subtracted from it to reduce heavy shadows caused by extreme lighting angles
- Histogram equalization: compensates the imaging effects due to changes in illumination and different camera input gains



Creating Virtual Positive Examples

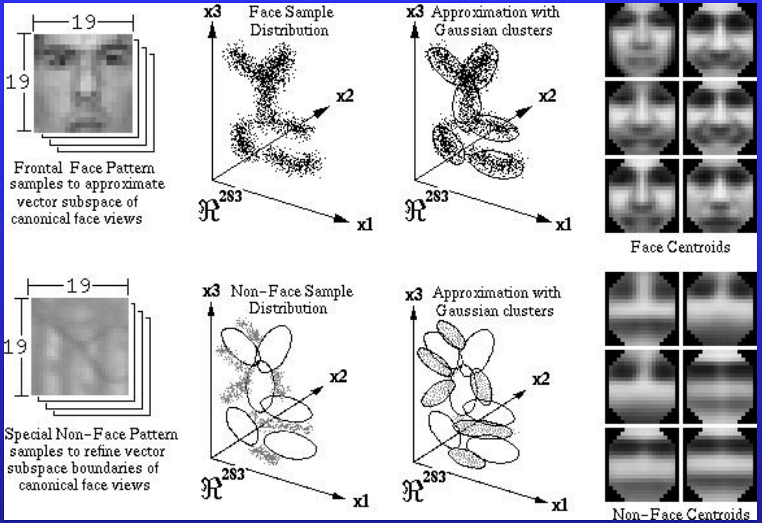
- Simple and very effective method
- Randomly mirror, rotate, translate and scale face samples by small amounts
- Increase number of training examples
- Less sensitive to alignment error



Randomly mirrored, rotated translated, and scaled faces [Sung & Poggio 94]

Distribution of Face/Non-face Pattern

- Cluster face and non-face samples into a few (i.e., 6) clusters using K-means algorithm
- Each cluster is modeled by a multi-dimensional Gaussian with a centroid and covariance matrix
- Approximate each Gaussian covariance with a subspace (i.e., using the largest eigenvectors)
- See [Moghaddam and Pentland 97] on distribution-based learning using Gaussian mixture model


19
19
Frontal Face Pattern samples to approximate vector subspace of canonical face views
x3
Face Sample Distribution
x2
x1
$\Re^{283}$
x3
Approximation with Gaussian clusters
x2
x1
$\Re^{283}$
Face Centroids
19
19
Special Non-Face Pattern samples to refine vector subspace boundaries of canonical face views
x3
Non-Face Sample Distribution
x2
x1
$\Re^{283}$
x3
Approximation with Gaussian clusters
x2
x1
$\Re^{283}$
Non-Face Centroids

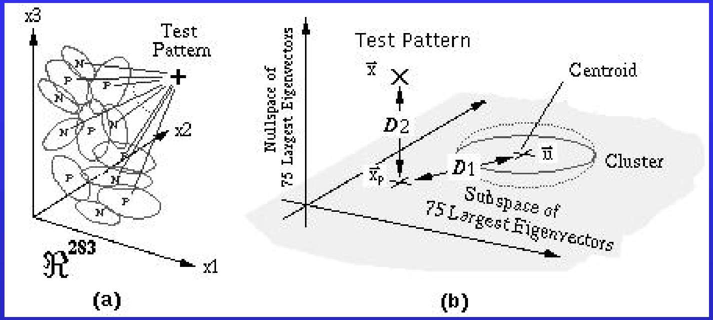
[Sung & Poggio 94]

$$p(\mathbf{x}) = \frac{1}{(2\pi)^{d/2}|\mathbf{\Sigma}|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{T}\mathbf{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right\}$$

x: face, non-face samples

Distance Metrics [Sung & Poggio 94]

- Compute distances of a sample to all the face and non-face clusters
- Each distance measure has two parts:
  - Within subspace distance (D_1): Mahalanobis distance of the projected sample to cluster center
  - Distance to the subspace (D_2): distance of the sample to the subspace
- Feature vector: Each face/non-face samples is represented by a vector of these distance measurements
- Train a multilayer perceptron using the feature vectors for face detection



[Sung and Poggio 94]

$$D_1 = \frac{1}{2}(d \ln 2\pi + \ln|\mathbf{\Sigma}| + (\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}} \mathbf{\Sigma}^{-1} (\mathbf{x}-\boldsymbol{\mu}))$$

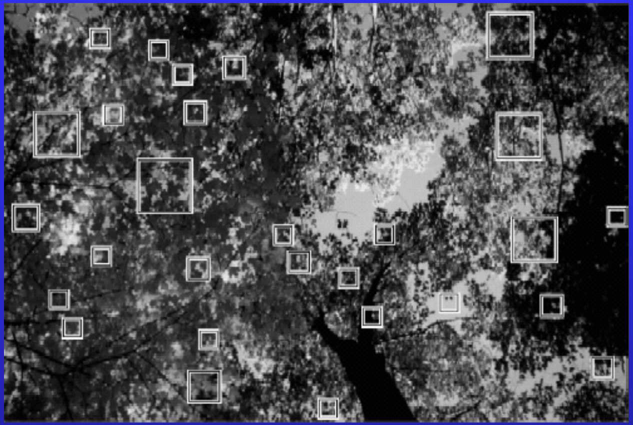
$$D_2 = \left|(\mathbf{x}-\mathbf{x}_p)\right|^2 = \left|(I - E_{75}E_{75}^{\mathrm{T}})(\mathbf{x}-\boldsymbol{\mu})\right|^2$$

- 6 face clusters
- 6 non-face clusters
- 2 distance values per cluster
- 24 measurements

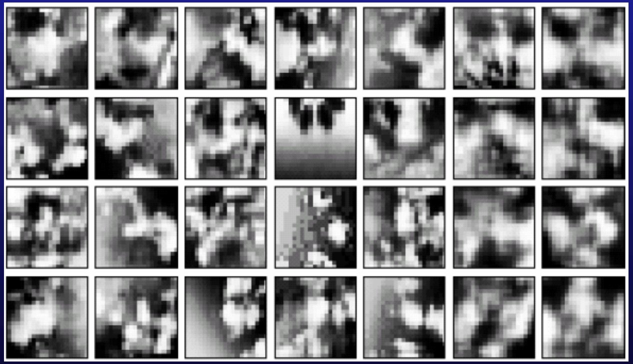
Bootstrapping [Sung and Poggio 94]

1. Start with a small set of non-face examples in the training set
2. Train a MLP classifier with the current training set
3. Run the learned face detector on a sequence of random images.
4. Collect all the non-face patterns that the current system wrongly classifies as faces (i.e., false positives)
5. Add these non-face patterns to the training set
6. Got to Step 2 or stop if satisfied

➔ Improve the system performance greatly

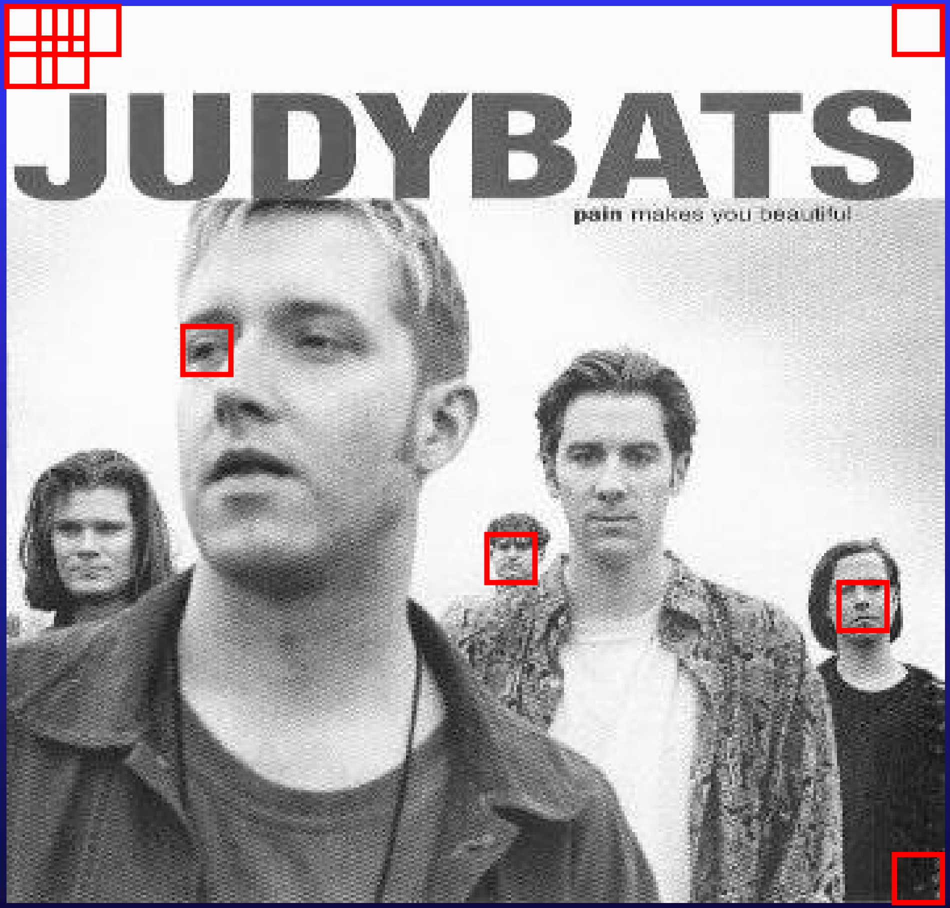


Test image

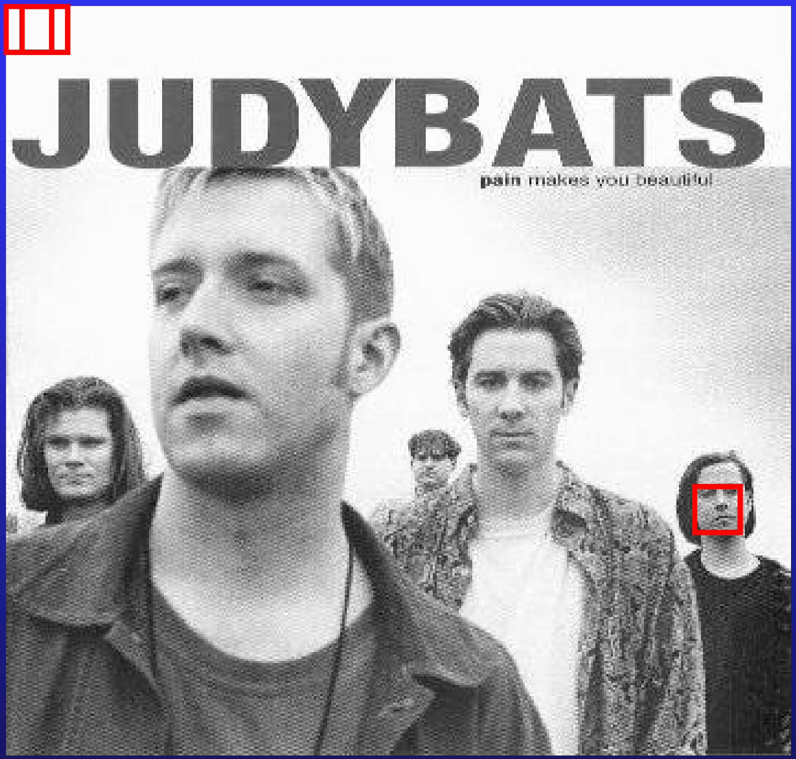


False positive detects

Search over Space and Scale

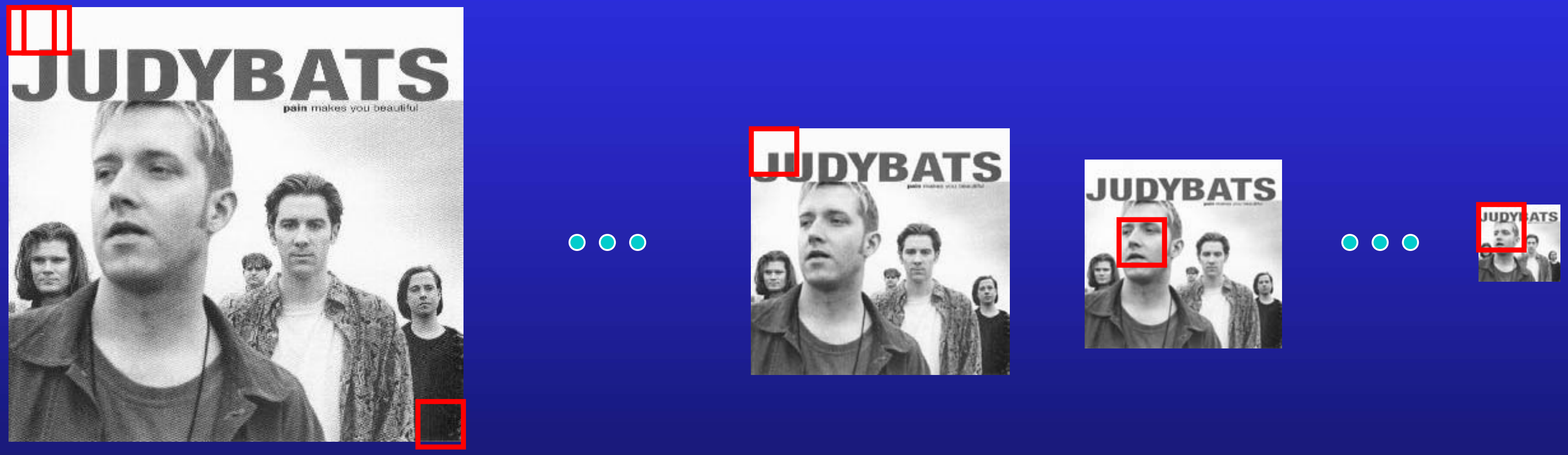


Scan an input image at one-pixel increments
horizontally and vertically



Downsample the input image by
a factor of 1.2 and continue to search

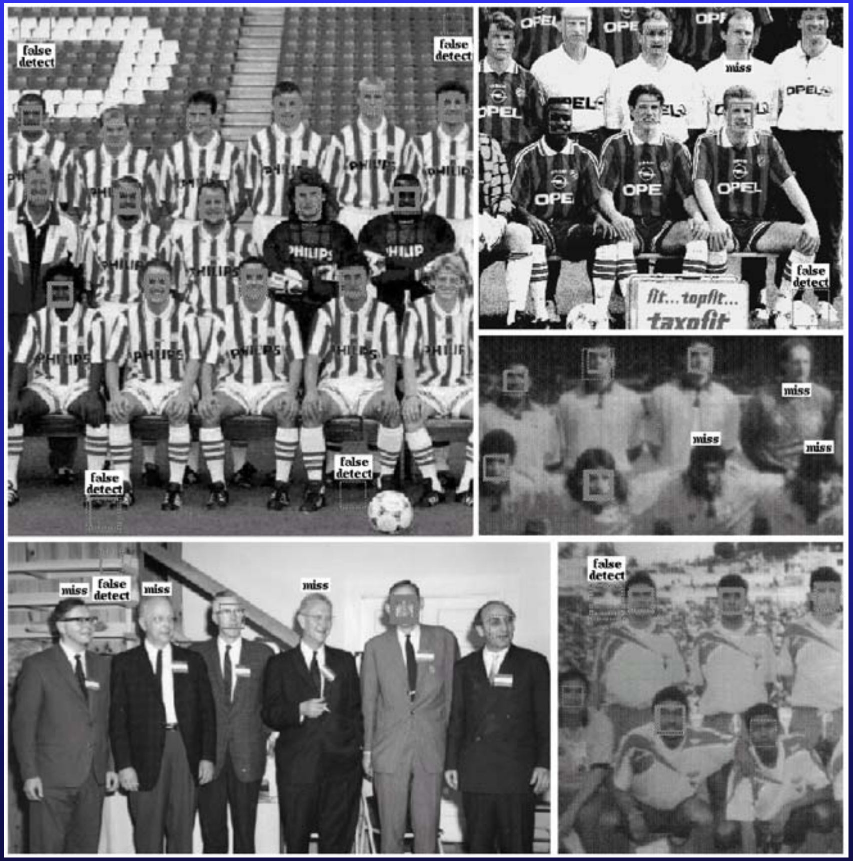
Continue to Search over Space and Scale



Continue to downsample the input image and search until the image size is too small

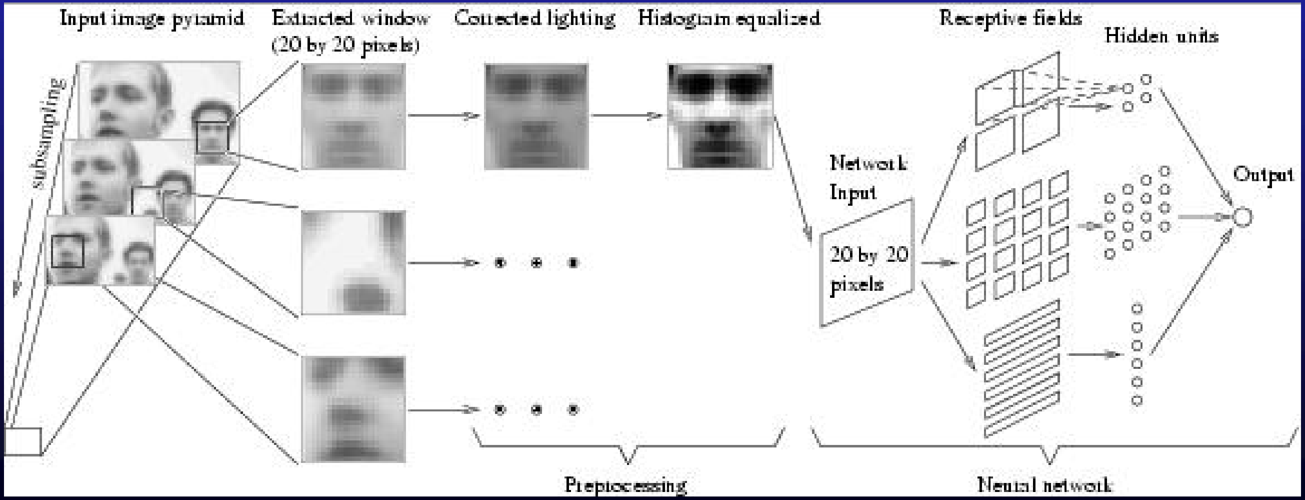
Experimental Results: [Sung and Poggio 94]

- Can be have multiple detects of a face since it may be detected
  - at different scale
  - at a slightly displaced window location
- Able to detect upright frontal faces



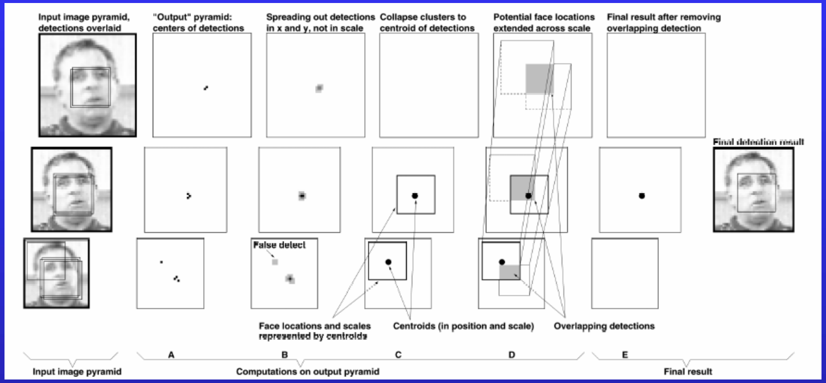
Neural Network-Based Detector

- Train multiple multilayer perceptrons with different receptive fields [Rowley and Kanade 96].
- Merging the overlapping detections within one network
- Train an arbitration network to combine the results from different networks
- Needs to find the right neural network architecture (number of layers, hidden units, etc.) and parameters (learning rate, etc.)

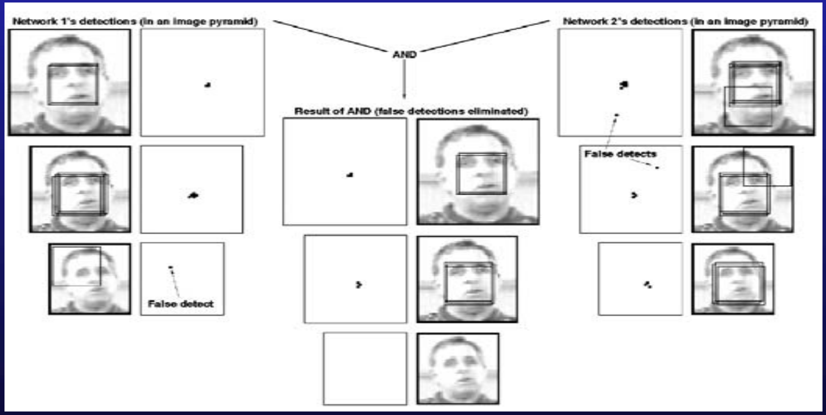


Dealing with Multiple Detects

- Merging overlapping detections within one network [Rowley and Kanade 96]
- Arbitration among multiple networks
  - AND operator
  - OR operator
  - Voting
  - Arbitration network

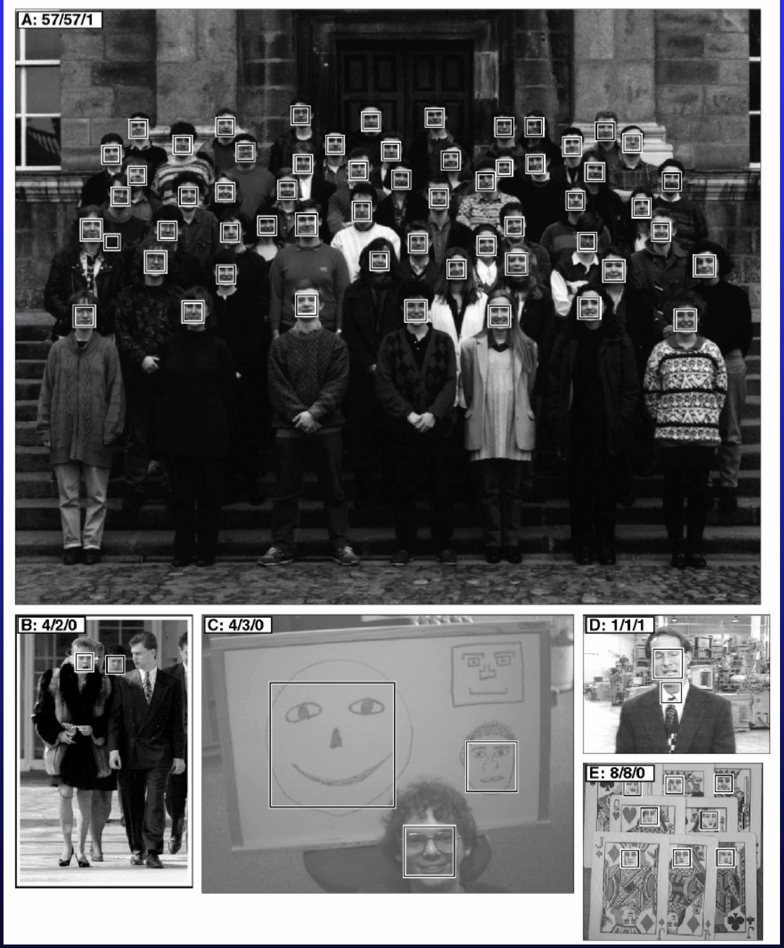

Input image pyramid, detections overlaid
"Output" pyramid: centers of detections
Spreading out detections in x and y, not in scale
Collapse clusters to centroid of detections
Potential face locations extended across scale
Final result after removing overlapping detection
Final detection result
False detect
Face locations and scales represented by centroids
Centroids (in position and scale)
Overlapping detections
A
B
C
D
E
Input image pyramid
Computations on output pyramid
Final result

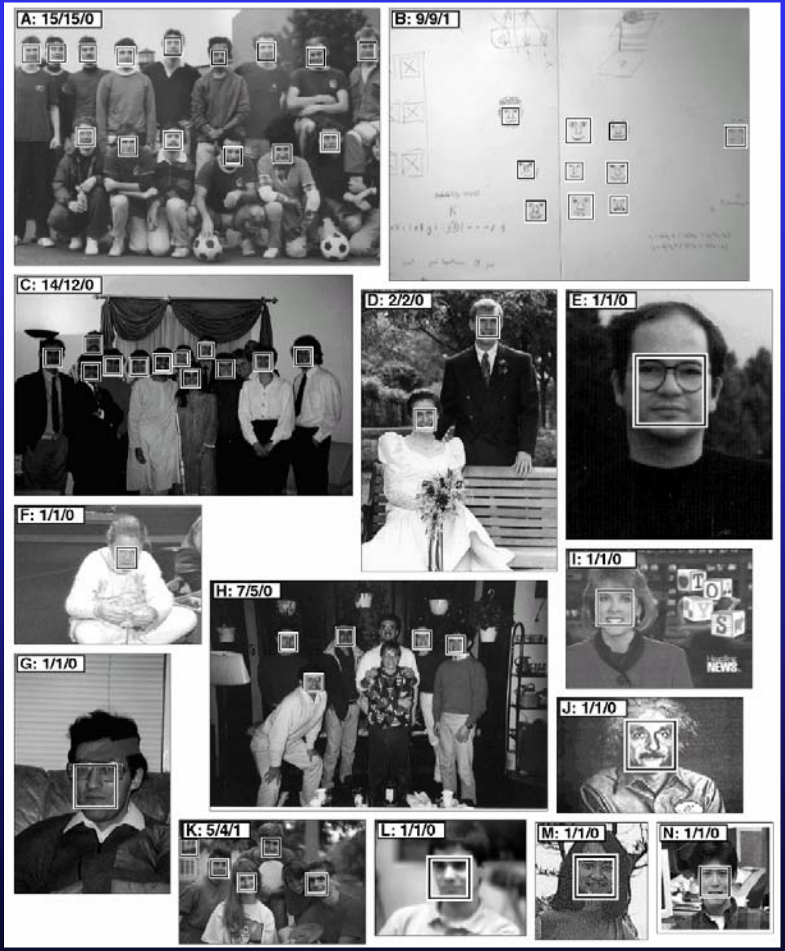
Merging overlapping results


Network 1's detections (in an image pyramid)
AND
Network 2's detections (in an image pyramid)
Result of AND (false detections eliminated)
False detects
False detect

ANDing results from two networks

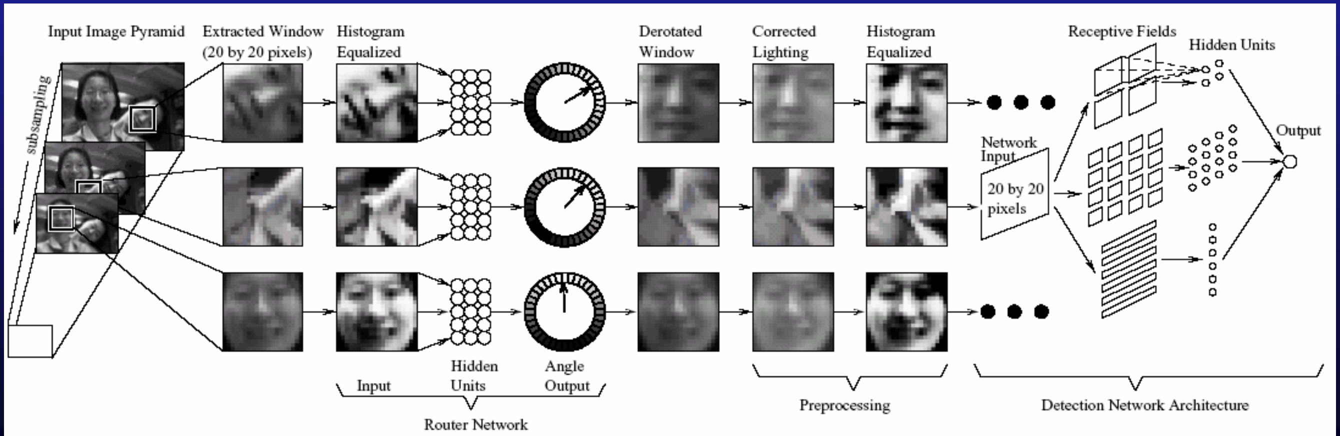
Experimental Results: [Rowley et al. 96]





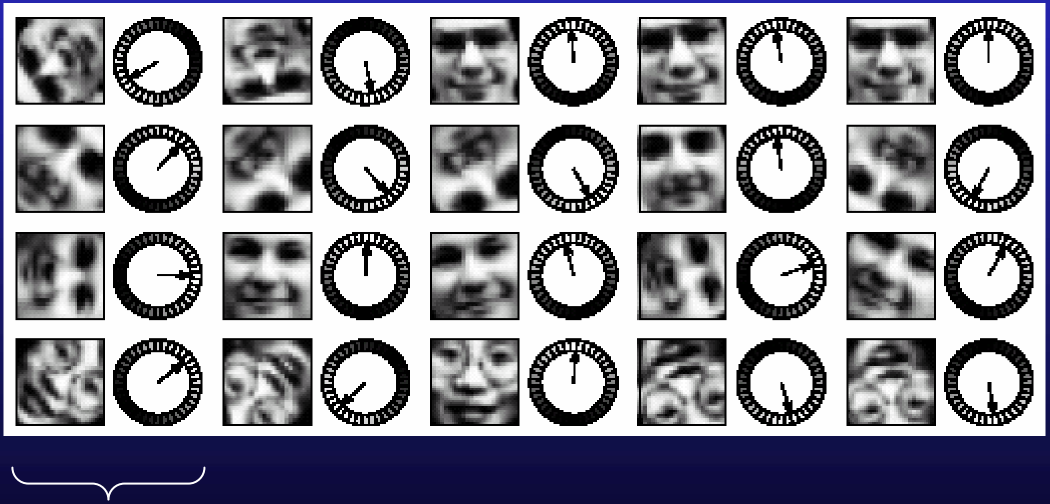
Detecting Rotated Faces [Rowley et al. 98]

- A router network is trained to estimate the angle of an input window
  - If it contain a face, the router returns the angle of the face and the face can be rotated back to upright frontal position.
  - Otherwise the router returns a meaningless angle
- The de-rotated window is then applied to a detector (previously trained for upright frontal faces)



Router Network [Rowley et al. 98]

- Rotate a face sample at 10 degree increment
- Create virtual examples (translation and scaling) from each sample
- Train a multilayer neural network with input-output pair



Input-output pair to train a router network

Experimental Results [Rowley et al. 98]

- Able to detect rotated faces with good results
- Performance degrades in detecting upright frontal faces due to the use of router network

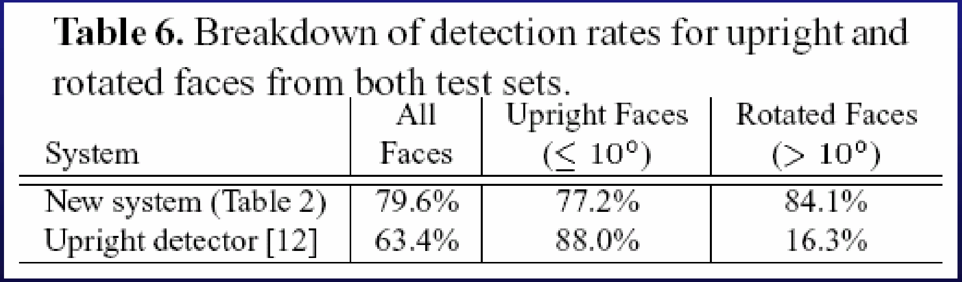
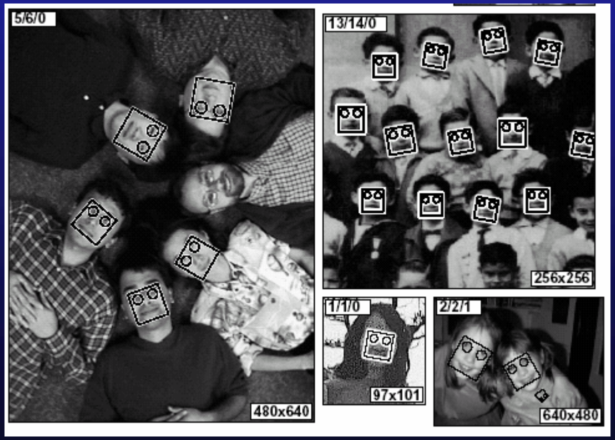


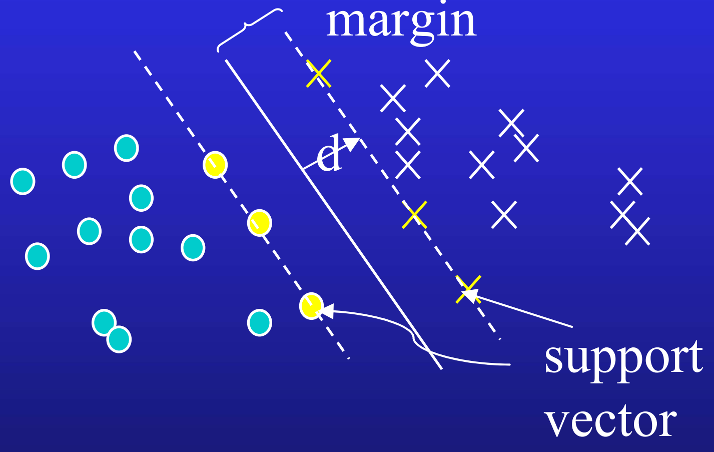
Table 6. Breakdown of detection rates for upright and rotated faces from both test sets.

System	All Faces	Upright Faces ($\leq 10^\circ$)	Rotated Faces ($> 10^\circ$)
New system (Table 2)	79.6%	77.2%	84.1%
Upright detector [12]	63.4%	88.0%	16.3%



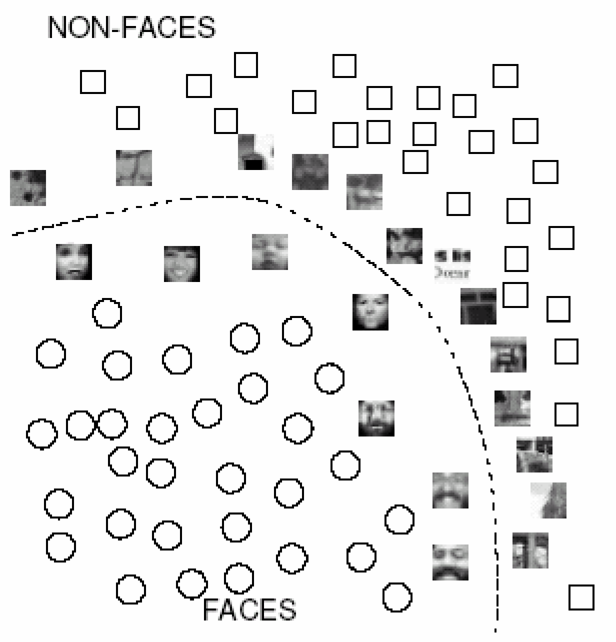
Support Vector Machine (SVM)

- Find the optimal separating hyperplane constructed by support vectors [Vapnik 95]
- Maximize distances between the data points closest to the separating hyperplane (large margin classifier)
- Formulated as a quadratic programming problem
- Kernel functions for nonlinear SVMs



SVM-Based Face Detector [Osuna et al. 97]

- Adopt similar architecture Similar to [Sung and Poggio 94] with the SVM classifier
- Pros: Good recognition rate with theoretical support
- Cons:
  - Time consuming in training and testing
  - Need to pick the right kernel



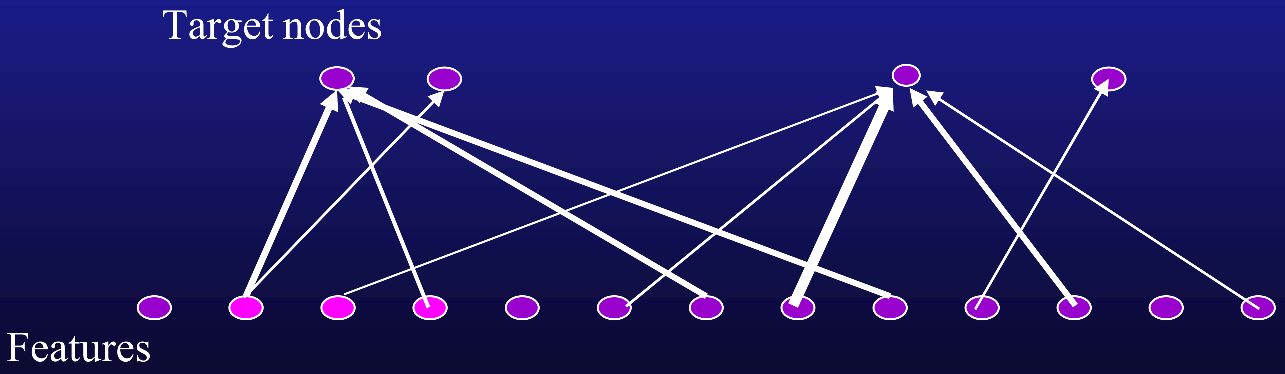
[Osuna et al. 97]

SVM-Based Face Detector: Issues

- Training: Solve a complex quadratic optimization problem
  - Speed-up: Sequential Minimal Optimization (SMO) [Platt 99]
- Testing: The number of support vectors may be large → lots of kernel computations
  - Speed-up: Reduced set of support vectors [Romdhani et al. 01]
- Variants:
  - Component-based SVM [Heisele et al. 01]:
    - Learn components and their geometric configuration
    - Less sensitive to pose variation

Sparse Network of Winnows [Roth 98]

- On line, mistake driven algorithm
- Attribute (feature) efficiency
- Allocations of nodes and links is data driven
  - complexity depends on number of active features
- Allows for combining task hierarchically
- Multiplicative learning rule



SNoW-Based Face Detector

- Multiplicative weight update algorithm:

Prediction is 1 iff $\mathbf{w} \bullet \mathbf{x} \geq \theta$

If Class $= 1$ but $\mathbf{w} \bullet \mathbf{x} \leq \theta$, $\mathbf{w}_i \leftarrow \alpha\, \mathbf{w}_i$ (if $\mathbf{x}_i = 1$) (promotion)

If Class $= 0$ but $\mathbf{w} \bullet \mathbf{x} \geq \theta$, $\mathbf{w}_i \leftarrow \beta\, \mathbf{w}_i$ (if $\mathbf{x}_i = 1$) (demotion)

Usually, $\alpha = 2$, $\beta = 0.5$

- Pros: On-line feature selection [Yang et al. 00]
- Cons: Need more powerful feature representation scheme
- Also been applied to object recognition [Yang et al. 02]

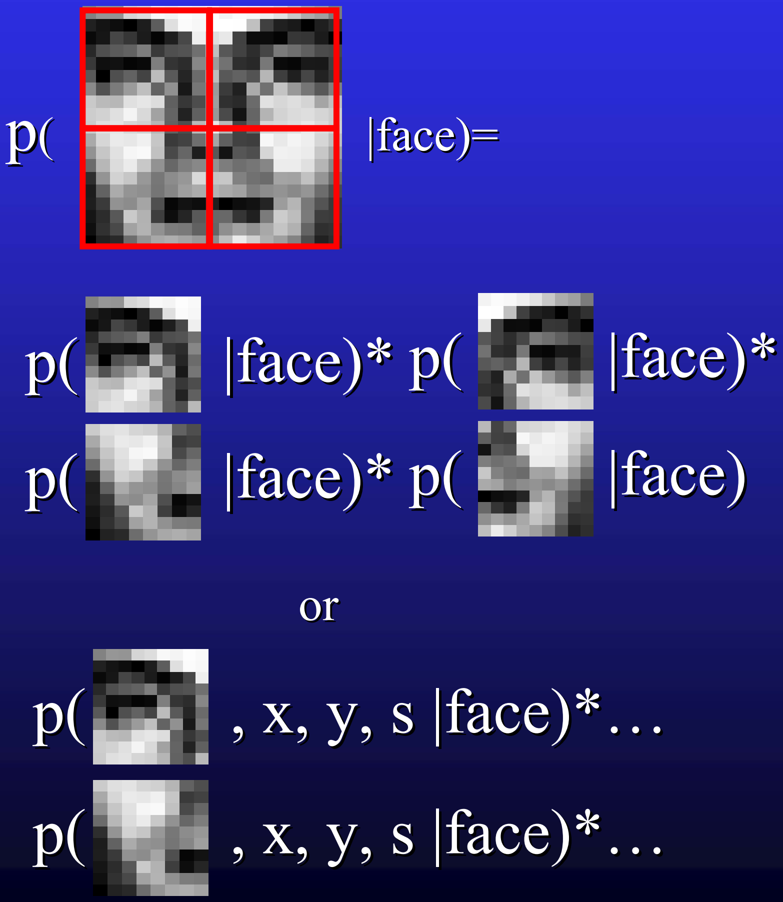
Probabilistic Modeling of Local Appearance [Schneiderman and Kanade 98]

$$p(region \mid object) = \prod_{k=1}^{n} p(subregion_k \mid object)$$

- Using local appearance
- Learn the distribution by parts using Naïve Bayes classifier
- Apply Bayesian decision rule

$$\frac{p(region \mid object)}{P(region \mid \overline{object})} > \lambda = \frac{p(\overline{ojbect})}{p(object)}$$

- Further decompose the appearance into space, frequency, and orientation
- Learn the joint distribution of object and position
- Also wavelet representation



p(|face)=

p(|face)* p(|face)*

p(|face)* p(|face)

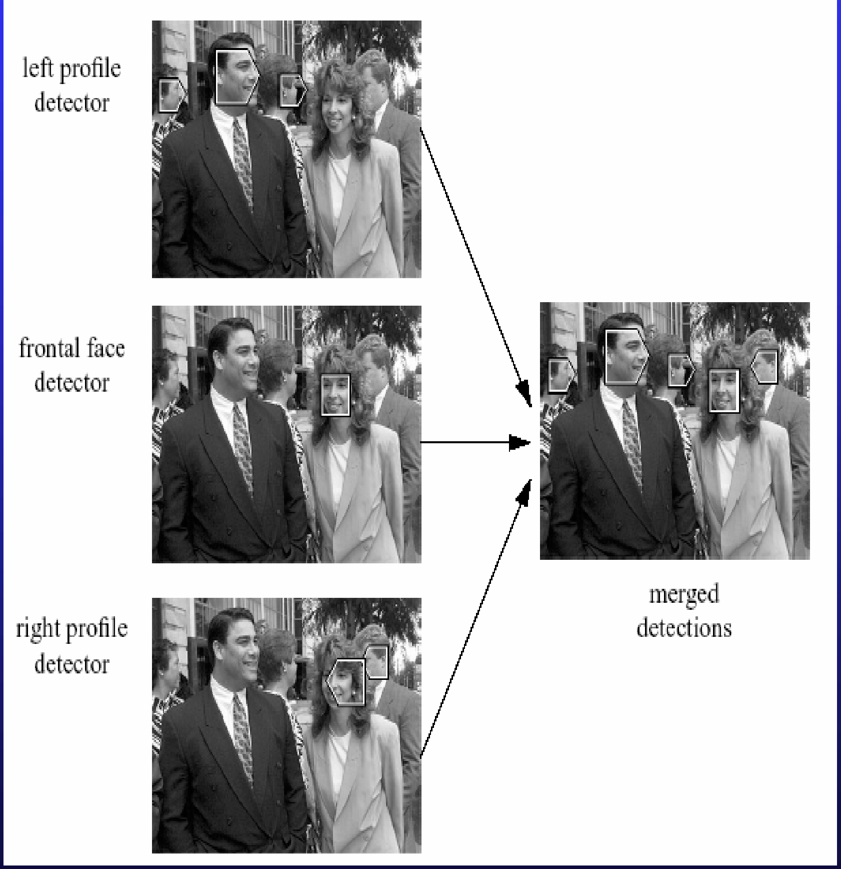
or

p(, x, y, s |face)*…

p(, x, y, s |face)*…

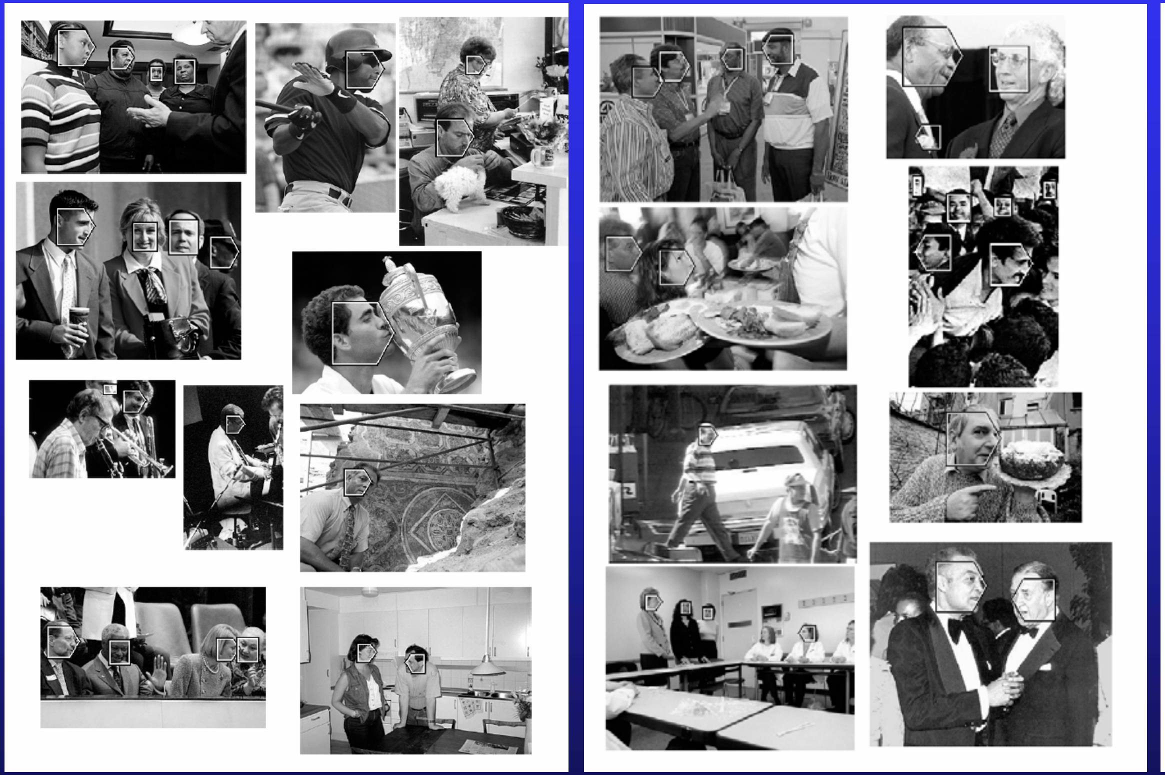
Detecting faces in Different Pose

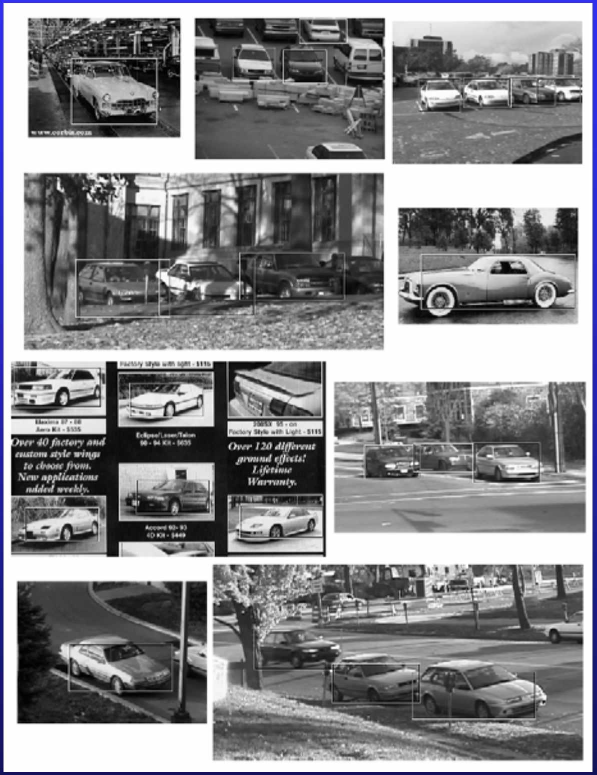
- Extend to detect faces in different pose with multiple detectors
- Each detector specializes to a view: frontal, left pose and right pose
- [Mikolajczyk et al. 01] extend to detect faces from side pose to frontal view



[Schneiderman and Kanade 98]

Experimental Results [Schneiderman and Kanade 98]



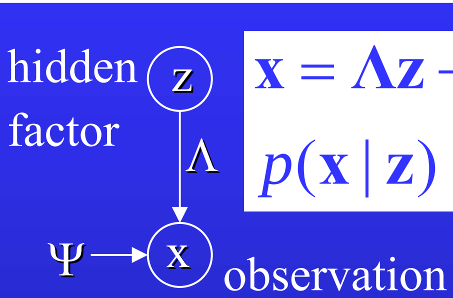

Over 40 factory and custom style wings to choose from. New applications added weekly.
Over 120 different ground effects! Lifetime Warranty.

Able to detect profile faces
[Schneiderman and Kanade 98]

Extended to detect cars
[Schneiderman and Kanade 00]

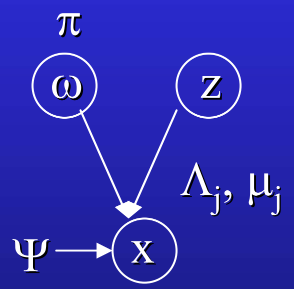
Mixture of Factor Analyzers [Yang et al. 00]

- Generative method that performs clustering and dimensionality reduction within each cluster
- Similar to probabilistic PCA but has more merits
  - proper density model
  - robust to noise
- Use mixture model to detect faces in different pose
- Using EM to estimate all the parameters in the mixture model
- See also [Moghaddam and Pentland 97] on using probabilistic Gaussian mixture for object localization



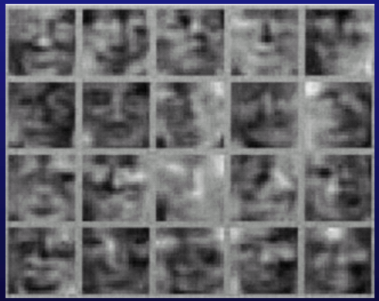
$$\mathbf{x} = \mathbf{\Lambda z} + \mathbf{u}$$

$$p(\mathbf{x} \mid \mathbf{z}) = N(\mathbf{\Lambda z}, \mathbf{\Psi})$$

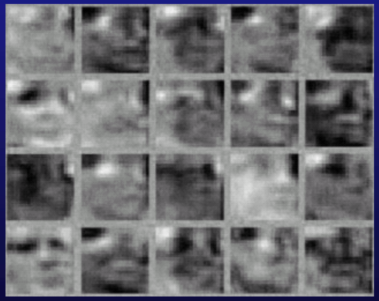


$$p(\mathbf{x} \mid \mathbf{z}, \omega_j) =$$
$$N(\mu_j + \mathbf{\Lambda}_j \mathbf{z}, \mathbf{\Psi})$$

mixture model



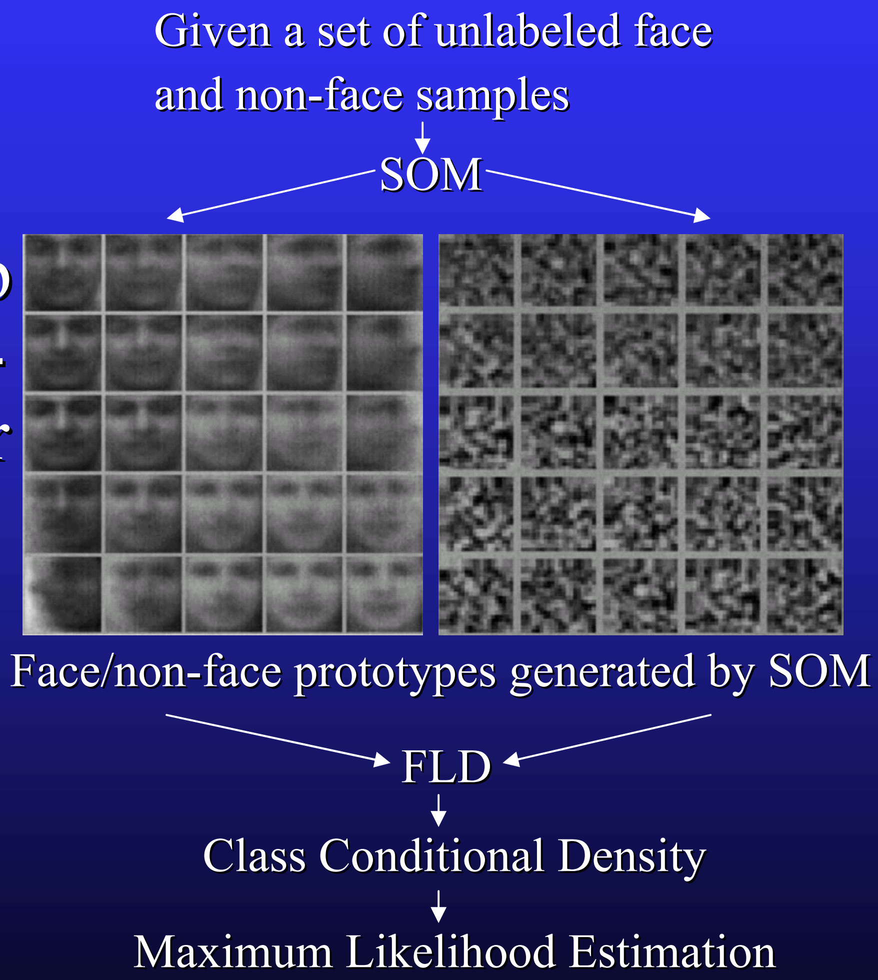
Factor faces for frontal view



Factor faces for 45° view

Fisher Linear Discriminant [Yang et al. 00]

- Fisherface (FLD) demonstrated good results in face recognition
- Apply Self-Organizing Map (SOM) to cluster faces/non-faces, and thereby labels for samples
- Apply FLD to find optimal projection matrix for maximal separation
- Estimate class-conditional density for detection



Given a set of unlabeled face and non-face samples

SOM

Face/non-face prototypes generated by SOM

FLD

Class Conditional Density

Maximum Likelihood Estimation

Adaboost [Freund and Schapire 95]

- Use a set of weak classifiers ($\varepsilon_t < 0.5$) and weighting on difficult examples for learning (sampling is based on the weights)
- Given: $(x_1, y_1), \ldots, (x_m, y_m)$ where $x_i \in X$, $y_i \in Y = \{-1,+1\}$

  Initialize $D_1(i)=1/m$

  For $t = 1, \ldots, T$:

  - Train a weak classifier using distribution D_t
    1. Get a weak hypothesis h_t: $X \rightarrow \{-1,+1\}$ with error $\varepsilon_t = \Pr_{i \sim Dt}[h_t(x_i) \neq y_i]$
    2. Importance of h_t:$\alpha_t = 1/2 \ln((1- \varepsilon_t)/ \varepsilon_t))$
    3. Update: $D_{t+1}(i)= D_t(i)/Z_t \times e^{-\alpha t}$ if $h_t\ (x)=y_i$ (correctly classified)

       $D_{t+1}(i)= D_t(i)/Z_t \times e^{\alpha t}$ if $h_t\ (x) \neq y_i$ (incorrectly classified)

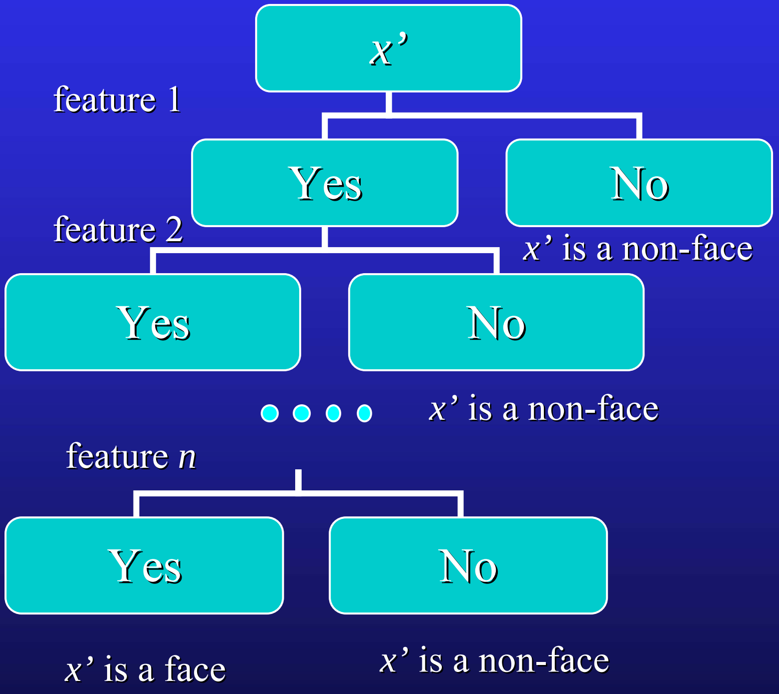
       where Z_t is a normalization factor
  - Aggregating the classifiers: $H(x)=\text{sign}(\Sigma_{t=1}\ \alpha_t\, h_t(x))$
- Perform well and does not overfit in empirical studies

Adaboost-Based Detector [Viola and Jones 00]

- Main idea:
  - Feature selection: select important features
  - Focus of attention: focus on potential regions
  - Use an integral graph for fast feature evaluation
- Use Adaboost to learn
  - A set of important features (feature selection)
    - sort them in the order of importance
    - each feature can be used as a simple (weak) classifier
  - A cascade of classifiers that
    - combine all the weak classifiers to do a difficult task
    - filter out the regions that most likely do not contain faces

Feature Selection [Viola and Jones 00]

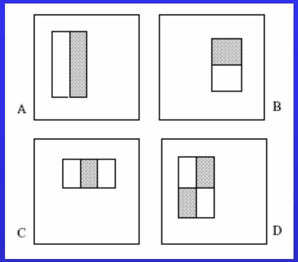
- Training: If *x* is a face, then *x*
  - most likely has feature 1 (easiest feature, and of greatest importance)
  - very likely to have feature 2 (easy feature)
  - …
  - likely to have feature n (more complex feature, and of less importance since it does not exist in all the faces in the training set)
- Testing: Given a test sub-image *x'*
  - if *x'* has feature 1:
    - Test whether *x'* has feature 2
      - Test whether *x'* has feature n
        - …
      - else …
    - else, it is not face
  - else, it is not a face
- Similar to decision tree



One simple implementation

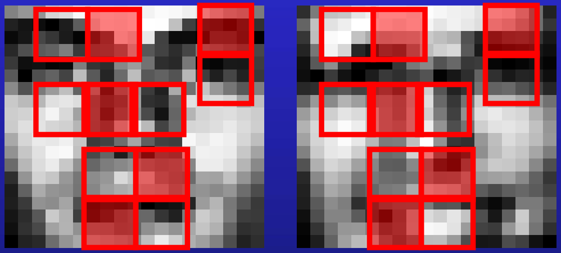
Boxlet As Weak Classifier [Viola & Jones 00]

- Boxlet: compute the difference between the sums of pixels within two rectangular regions
  - Compute boxlets all over a pattern
  - Harr-like wavelets
  - Over-complete representation: lots of boxlet features

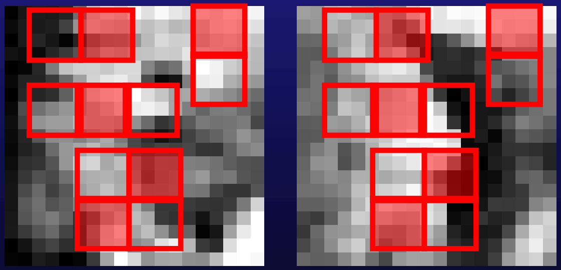


Boxlet:
2-rectangle,
3-rectangle,
4-rectangle

- For each boxlet *j*, compute $f_j(x)$ where x is a positive or negative example
- Each feature is used as a weak classifier
- Set threshold θ_j so that most samples are classified correctly: $h_j(x)=1$ if $f_j(x) < \theta_j$ (or $f_j(x) > \theta_j$)
- Sequentially select the boxlets that exist in most positive examples



face samples



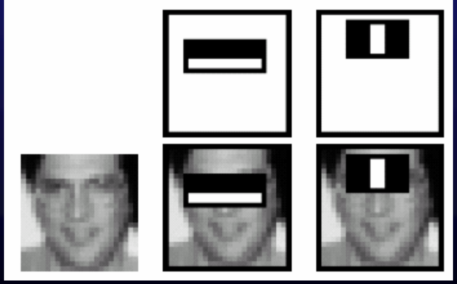
non-face samples

Selecting Features Using Adaboost

[Viola and Jones 00]

- For t=1, …, T
  - Construct a weak classifier using one single feature h_t $h_j(x)=\begin{cases}1 & \text{if } p_j f_j(x) < p_j\theta_j \\ 0 & \text{otherwise}\end{cases}$ where p_j is a parity bit and θ_j is a threshold
  - For each feature j, train a classifier h_j, the error is evaluated with respect to w_t, $\varepsilon_t=\Sigma_i w_i|h_j(x_i)-y_i|$
  - Choose the classifier h_t, with the loser error ε_t
  - Update the weights: $w_{t+1,i}=w_{t,i}\,\beta_t^{\,1-e_i}$, $e_i=0$ if x_i is correctly classified, and $e_i=1$ otherwise. $\beta_t = \varepsilon_t/(1-\varepsilon_t)$
- Final classifier:

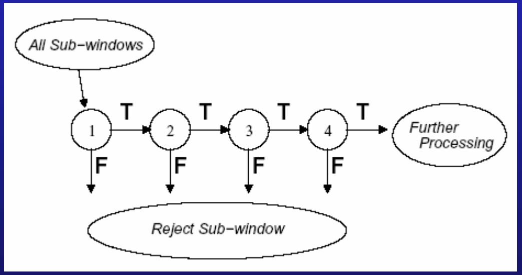
$$h_j(x)=\begin{cases}1 & \sum_{t=1}^{T}\alpha_t h_t(x) \ge \frac{1}{2}\sum_{t=1}^{T}\alpha_t \\ 0 & \text{otherwise}\end{cases}, \alpha_t = \log\frac{1}{\beta_t}$$



The top two boxlets selected by Adaboost

Attentional Cascade [Viola and Jones 00]

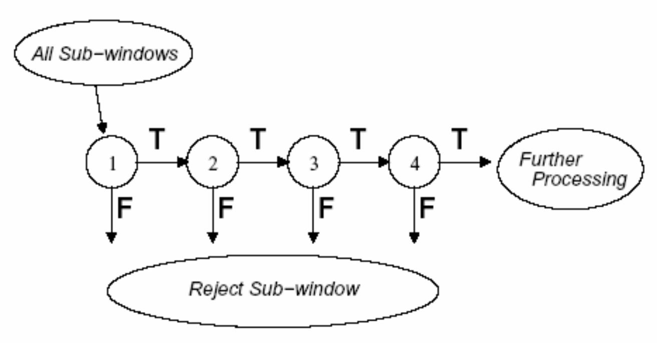
- Within an image, *most* sub-images are non-face instances
- Use smaller and efficient classifiers to reject *many* negative examples at early stage while detecting almost all the positive instances
- Simpler classifiers are used to reject the majority of sub-windows
- More complex classifiers are used at later stage to examine difficult cases
- Learn the cascade classifier using Adaboost, i.e., learn an ensemble of weak classifiers



Early stage classifier deals with easy instances while the deeper classifier faces more difficult cases.

Training Attentional Cascade

- Similar to decision tree
- Design parameters:
  - Number of cascade stages
  - Number of features of each stage
  - Threshold of each stage



[Viola and Jones 00]

- Example: 32 stage cascade classifier
  - 2-feature classifier in the firs stage → rejecting 60% non-faces while detecting 100% faces
  - 5-feature classifier in the second stage → rejecting 80% non-faces while detecting 100 % faces
  - 20-feature classifier in stages 3, 4, and 5
  - 50-feature classifier in stages 6 and 7
  - 100-feature classifier in stages 8 to 12
  - 200-feature classifier in stage 13 to 32

Adaboost-Based Detector: Summary

- Three main components [Viola and Jones 00] :
  - Integral graph: efficient convolution
  - Use Adaboost for feature selection
  - Use Adaboost to learn the cascade classifier
- Pros:
  - Fast and fairly robust; runs in real time.
- Cons:
  - Very time consuming in training stage (may take days in training)
  - Requires lots of engineering work
- Extended to multi-pose [Li et al. 02] [Viola and Jones 03]
- Kullback-Leibler Adaboost [Liu and Shum 03]
- Another greedy method: Anti-face [Keren 00]
- See also [Amit et al. 97] for efficient hierarchical (focus of attention) feature-based method

Appearance-Based Methods: Summary

- Pros:
  - Use powerful machine learning algorithms
  - Has demonstrated good empirical results
  - Fast and fairly robust
  - Extended to detect faces in different pose and orientation
- Cons
  - Usually needs to search over space and scale
  - Need lots of positive and negative examples
  - Limited view-based approach

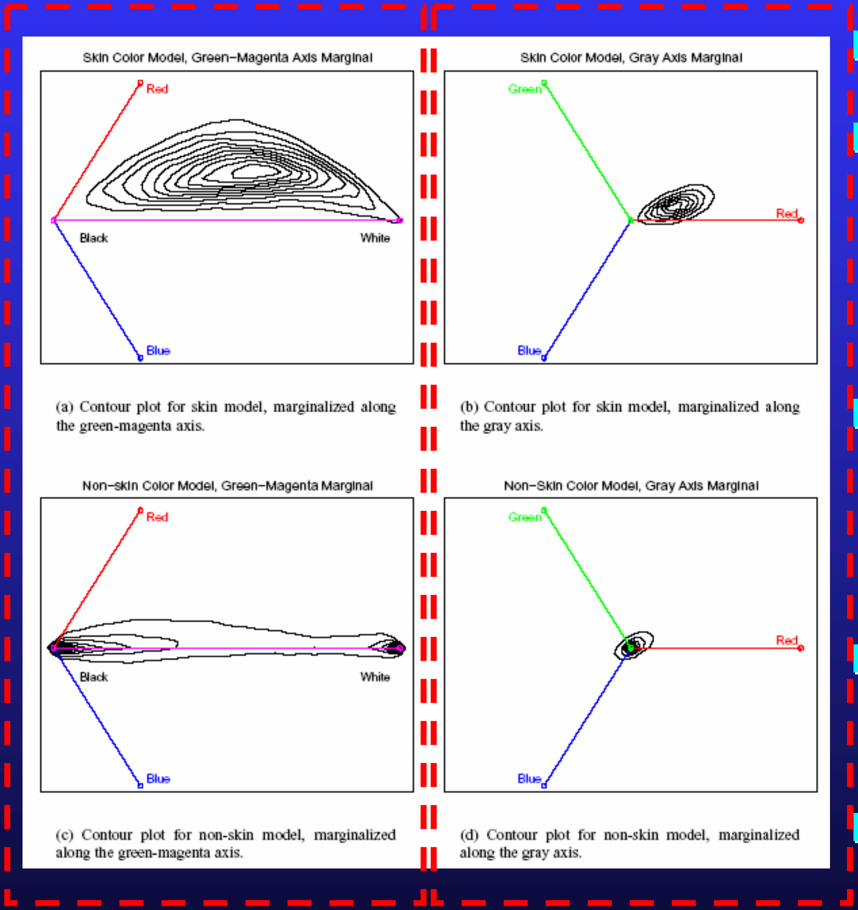
Agenda

- Detecting faces in gray scale images
  - Knowledge-based
  - Feature-based
  - Template-based
  - Appearance-based
- Detecting faces in color images
- Detecting faces in video
- Performance evaluation
- Research direction and concluding remarks

Color-Based Face Detector

- Distribution of skin color across different ethnic groups
  - Under controlled illumination conditions: compact
  - Arbitrary conditions: less compact
- Color space
  - RGB, normalized RGB, HSV, HIS, YCrCb, YIQ, UES, CIE XYZ, CIE LIV, …
- Statistical analysis
  - Histogram, look-up table, Gaussian model, mixture model, …

Skin and Non-Skin Color Model



Skin Color Model, Green–Magenta Axis Marginal

(a) Contour plot for skin model, marginalized along the green-magenta axis.

Skin Color Model, Gray Axis Marginal

(b) Contour plot for skin model, marginalized along the gray axis.

Non–skin Color Model, Green–Magenta Marginal

(c) Contour plot for non-skin model, marginalized along the green-magenta axis.

Non–Skin Color Model, Gray Axis Marginal

(d) Contour plot for non-skin model, marginalized along the gray axis.

[Jones and Rehg 02]

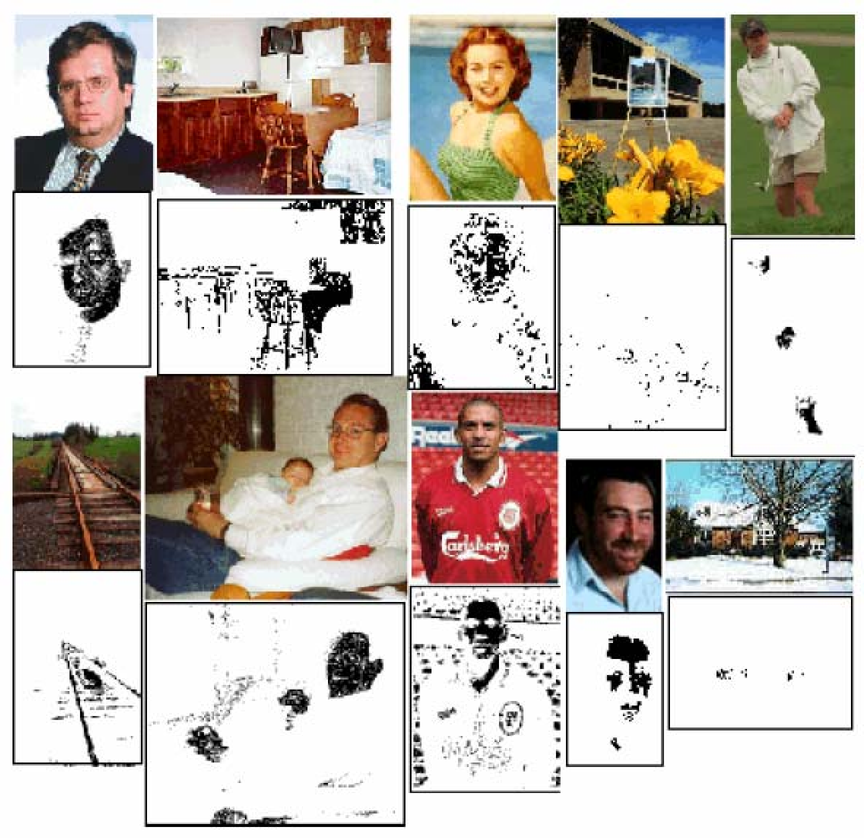
- Analyze 1 billion labeled pixels
- Skin and non-skin models:

$$P(rgb \mid skin) = \frac{s[rgb]}{Ts} \qquad P(rgb \mid \neg skin) = \frac{n[rgb]}{Tn}$$

$$\frac{P(rgb \mid skin)}{P(rgb \mid \neg skin)} \geq \theta$$

- A significant degree of separation between skin and non-skin model
- Achieves 80% detection rate with 8.5% false positives
- Histogram method outperforms Gaussian mixture model

Experimental Results [Jones and Rehg 02]



[Jones and Rehg 02]

- Does a decent job
- May have lots of false positives in the raw results
- Need further processing to eliminate false negatives and group color pixels for face detection

Color-Based Face Detector: Summary

- Pros:
  - Easy to implement
  - Effective and efficient in constrained environment
  - Insensitive to pose, expression, rotation variation
- Cons:
  - Sensitive to environment and lighting change
  - Noisy detection results (body parts, skin-tone line regions)

Agenda

- Detecting faces in gray scale images
  - Knowledge-based
  - Feature-based
  - Template-based
  - Appearance-based
- Detecting faces in color images
- Detecting faces in video
- Performance evaluation
- Research direction and concluding remarks

Video-Based Face Detector

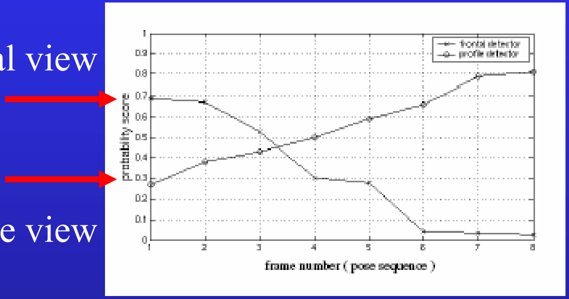
- Motion cues:
  - Frame differencing
  - Background modeling and subtraction
- Can also used depth cue (e.g., from stereo) when available
- Reduce the search space dramatically

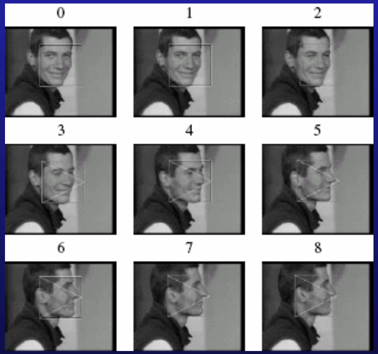
Face Detection in Video: [Mikolajczyk et al. 01]

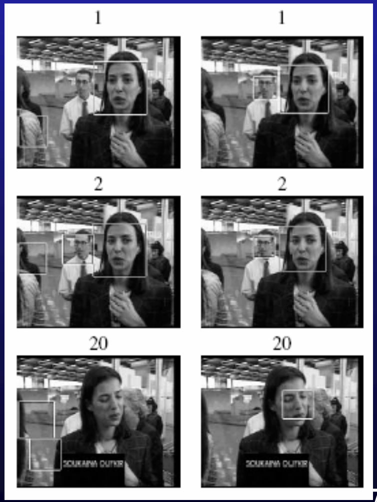
- Use two probabilistic detectors, one frontal $P_f(I,x,y,s)$ and one profile $P_p(I,x,y,s)$, based on [Schneiderman and Kanade 98].
- Predict state $s_t=(x_t,y_t,s_t,\theta_t)$ based on observation $z_t=(P_f, P_p)$ where x_t,y_t,s_t,θ_t are position, scale and pose angle.
- Using Condensation algorithm [Isard and Blake 96] to propagate the probability of detection and parameters over time.

P_f: frontal view

P_p: profile view

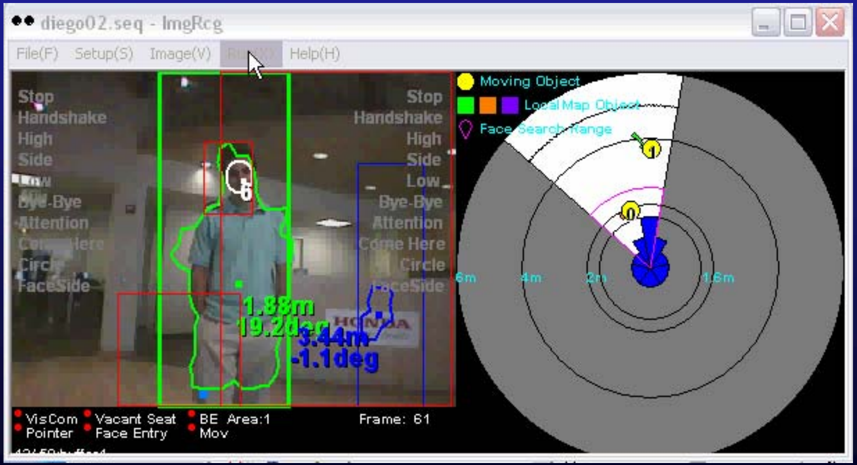






HONDA Humanoid Robot: ASIMO

- Using motion and depth cue
  - Motion cue: from a gradient-based tracker
  - Depth cue: from stereo camera
- Dramatically reduce search space
- Cascade face detector



Video-Based Detectors: Summary

- Pros:
  - An easier problem than detection in still images
  - Use all available cues: motion, depth, voice, etc. to reduce search space
- Cons
  - Need to efficient and effective methods to process the multimodal cues
  - Data fusion

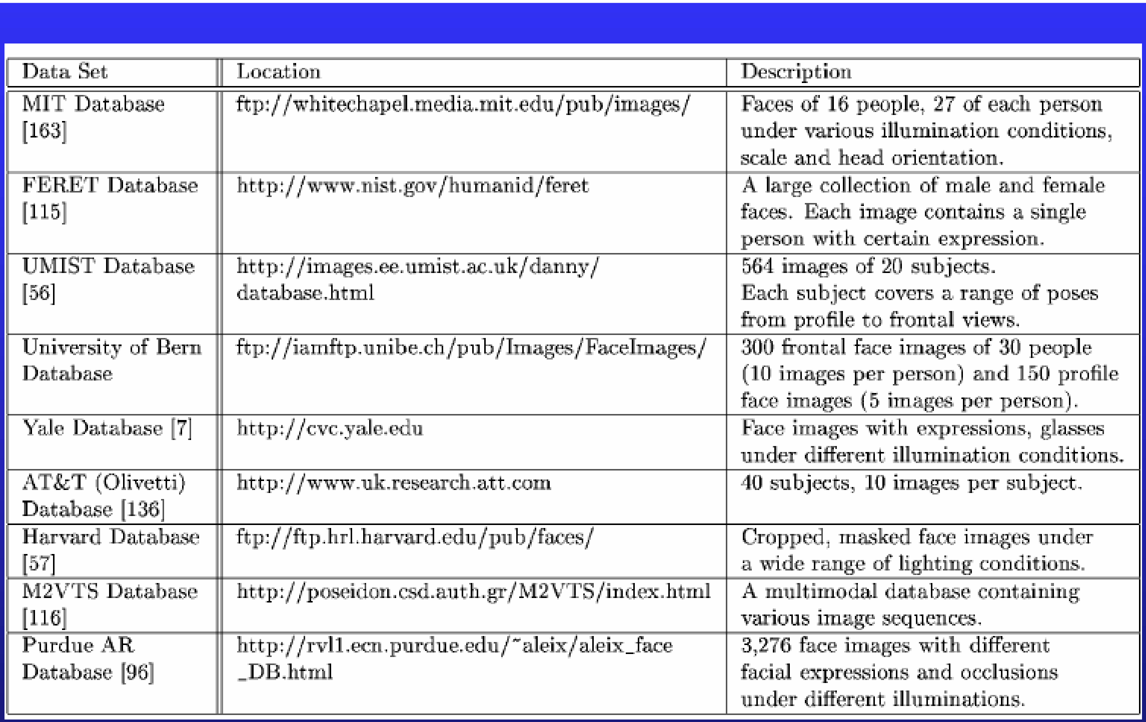
Agenda

- Detecting faces in gray scale images
  - Knowledge-based
  - Feature-based
  - Template-based
  - Appearance-based
- Detecting faces in color images
- Detecting faces in video
- Performance evaluation
- Research direction and concluding remarks

Performance Evaluation

- Tricky business
- Need to set the evaluation criteria/protocol
  - Training set
  - Test set
  - What is a correct detect?
  - Detection rate: false positive/negative
  - Precision of face location
  - Speed: training/test stage

Training Sets



Data Set	Location	Description
MIT Database [163]	ftp://whitechapel.media.mit.edu/pub/images/	Faces of 16 people, 27 of each person under various illumination conditions, scale and head orientation.
FERET Database [115]	http://www.nist.gov/humanid/feret	A large collection of male and female faces. Each image contains a single person with certain expression.
UMIST Database [56]	http://images.ee.umist.ac.uk/danny/database.html	564 images of 20 subjects. Each subject covers a range of poses from profile to frontal views.
University of Bern Database	ftp://iamftp.unibe.ch/pub/Images/FaceImages/	300 frontal face images of 30 people (10 images per person) and 150 profile face images (5 images per person).
Yale Database [7]	http://cvc.yale.edu	Face images with expressions, glasses under different illumination conditions.
AT&T (Olivetti) Database [136]	http://www.uk.research.att.com	40 subjects, 10 images per subject.
Harvard Database [57]	ftp://ftp.hrl.harvard.edu/pub/faces/	Cropped, masked face images under a wide range of lighting conditions.
M2VTS Database [116]	http://poseidon.csd.auth.gr/M2VTS/index.html	A multimodal database containing various image sequences.
Purdue AR Database [96]	http://rvl1.ecn.purdue.edu/~aleix/aleix_face_DB.html	3,276 face images with different facial expressions and occlusions under different illuminations.

← Mainly used for face recognition

← Need to crop and pre process the face images in the data set

- Cropped and pre processed data set with face and non-face images provided by MIT CBCL: http://www.ai.mit.edu/projects/cbcl/software-datasets/index.html

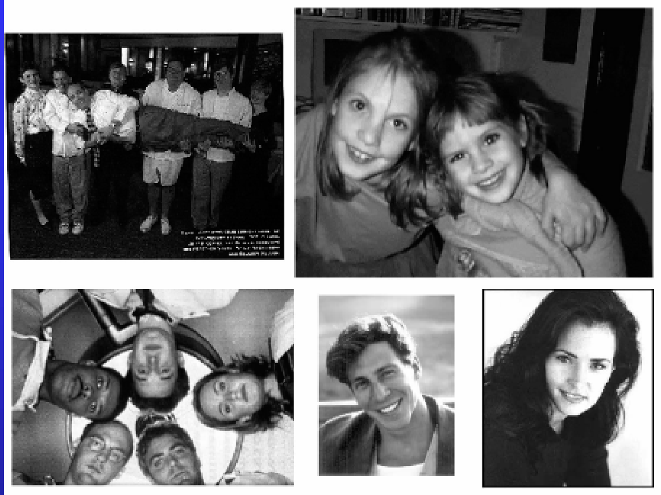
Standard Test Sets

- MIT test set (http://www.cs.cmu.edu/~har): subsumed by CMU test set
- CMU test set (http://www.cs.cmu.edu/~har) (de facto benchmark): 130 gray scale images with a total of 507 frontal faces
- CMU profile face test set (http://eyes.ius.cs.cmu.edu/usr20/ftp/testing_face_images.tar.gz) : 208 images with faces in profile views
- Kodak data set (Eastman Kodak Corp): faces of multiple size, pose and varying lighting conditions in color images

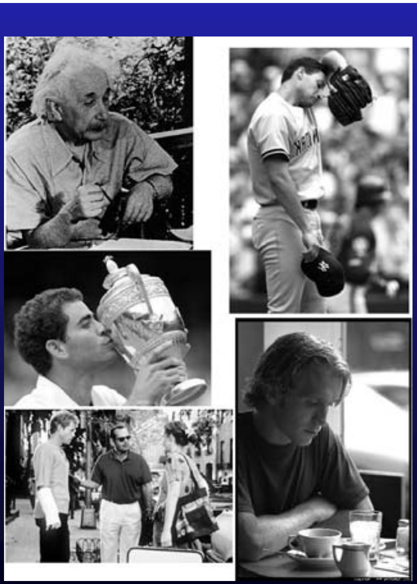
CMU Test Set I: Upright Frontal Faces

- 130 images with 507 frontal faces
- Collected by K.-K. Sung and H. Rowley
- Including 23 images used in [Sung and Poggio 94]
- Some images have low resolution
- De facto benchmark set

CMU Test Sets: Rotated and Profile Faces

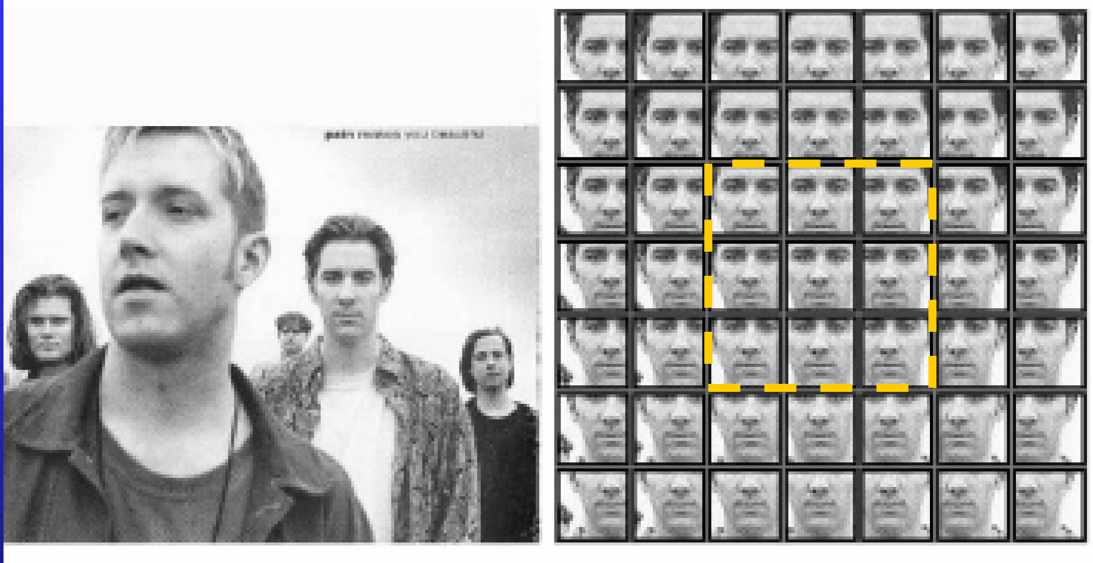


- 50 images with 223 faces in-plane orientation
- Collected by H. Rowley



- 208 images with 441 faces
- Collected by H. Schneiderman

What is a Correct Detect?

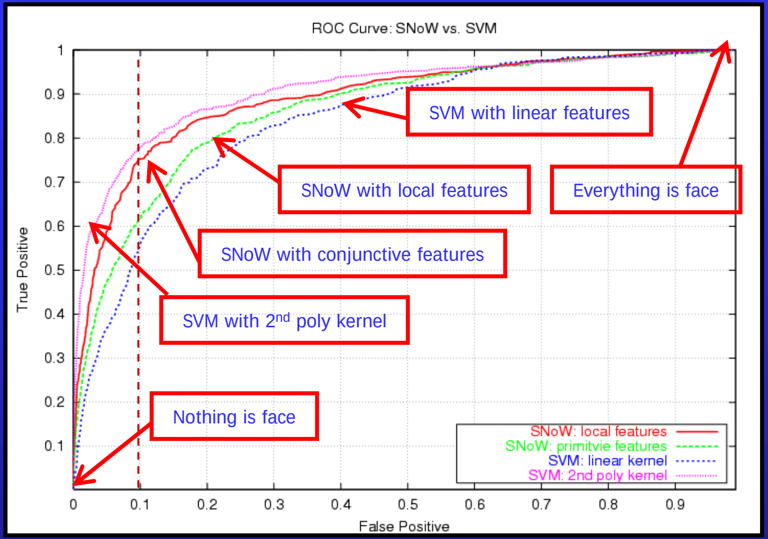


Which is a correct detect?

- Different interpretation of “correct detect”
- Precision of face location
- Affect the reporting results: detection, false positive, false negative rates

Receiver Operator Characteristic Curve

- Useful for detailed performance assessment
- Plot true positive (TP) proportion against the false positive (FP) proportion for various possible settings
- False positive: Predict a face when there is actually none
- False negative: Predict a non-face where there is actually one



SVM with linear features

SNoW with local features

Everything is face

SNoW with conjunctive features

SVM with 2nd poly kernel

Nothing is face

ROC Curve of a SVM-based detector (2nd order polynomial kernel): the detection rate is 78% with false positive rate of 10% (for a particular data set)

Agenda

- Detecting faces in gray scale images
  - Knowledge-based
  - Feature-based
  - Template-based
  - Appearance-based
- Detecting faces in color images
- Detecting faces in video
- Performance evaluation
- Research direction and concluding remarks

Research Issues

- Detect faces robustly under
  - varying pose: [Schneiderman and Kanade 00]
  - orientation: [Rowley and Kanade 98]
  - occlusion:
  - expression:
  - and varying lighting conditions
- Fast detector: [Viola and Jones 00]
- Precision
- Performance evaluation

Web Resources

- Face detection home page
  http://home.t-online.de/home/Robert.Frischholz/face.htm
- Henry Rowley's home page
  http://www-2.cs.cmu.edu/~har/faces.html
- Henry Schneiderman's home page
  http://www.ri.cmu.edu/projects/project_416.html
- MIT CBCL web page
  http://www.ai.mit.edu/projects/cbcl/software-datasets/index.html
- Face detection resources
  http://vision.ai.uiuc.edu/mhyang/face-detection-survey.html
- Google

References

- M.-H. Yang, D. J. Kriegman, and N. Ahuja, "Detecting Faces in Images: A Survey", *IEEE Transactions on Pattern Analysis and Machine Intelligence* (PAMI), vol. 24, no. 1, pp. 34-58, 2002.

- M.-H. Yang and N. Ahuja, *Face Detection and Hand Gesture Recognition for Human Computer Interaction*, Kluwer Academic Publishers, 2001.

- Web site:

  http://vision.ai.uiuc.edu/mhyang/face-detection-survey.html

Acknowledgements

Thanks for the help of the following people

- Narendra Ahuja
- Kevin Bowyer
- Jeffrey Ho
- Thomas Huang
- Michael Jones
- David Kriegman
- Thomas Leung
- Jongwoo Lim
- Krystian Mikolajczyk
- Baback Moghaddam
- Tomaso Poggio
- James Rehg
- Ryan Rifkin
- David Ross
- Dan Roth
- Henry Rowley
- Brian Scassellati
- Henry Schneiderman
- Paul Viola
- Kin Choong Yow
- Danny Yang
- …