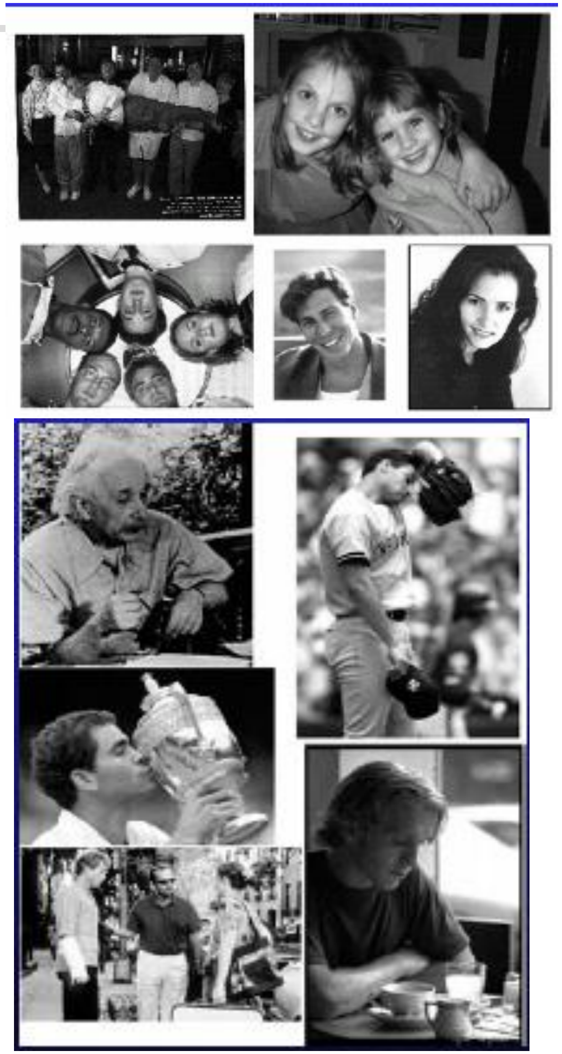
An Introduction to Face Detection and Recognition

Outline

- Face Detection
  - What is face detection?
  - Importance of face detection
  - Current state of research
  - Different approaches
    - One example
- Face Recognition
  - What is face recognition?
  - Its applications
  - Different approaches
    - One example
- A Video Demo

What is Face Detection?

- Given an image, tell whether there is any human face, if there is, where is it(or where they are).



Importance of Face Detection

- The first step for any automatic face recognition system system
- First step in many Human Computer Interaction systems
  - Expression Recognition
  - Cognitive State/Emotional State Recogntion
- First step in many surveillance systems
- Tracking: Face is a highly non rigid object
- A step towards Automatic Target Recognition(ATR) or generic object detection/recognition
- Video coding……

Face Detection: current state

- State-of-the-art:
  - Front-view face detection can be done at >15 frames per second on 320x240 black-and-white images on a 700MHz PC with ~95% accuracy.
  - Detection of faces is faster than detection of edges!
- Side view face detection remains to be difficult.

Face Detection: challenges

- Out-of-Plane Rotation: frontal, 45 degree, profile, upside down
- Presence of beard, mustache, glasses etc
- Facial Expressions
- Occlusions by long hair, hand
- In-Plane Rotation
- Image conditions:
  - Size
  - Lighting condition
  - Distortion
  - Noise
  - Compression

Different Approaches

- Knowledge-based methods:
  - Encode what constitutes a typical face, e.g., the relationship between facial features
- Feature invariant approaches:
  - Aim to find structure features of a face that exist even when pose, viewpoint or lighting conditions vary
- Template matching:
  - Several standard patterns stored to describe the face as a whole or the facial features separately
- Appearance-based methods:
  - The models are learned from a set of training images that capture the representative variability of faces.

Knowledge-Based Methods

- Top 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]

- 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 equalization followed by edge detection
- Level 3: search for eye and mouth features for validation

Knowledge-based Methods: Summary

- Pros:
  - Easy to come up with simple rules
  - 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

Feature-Based Methods

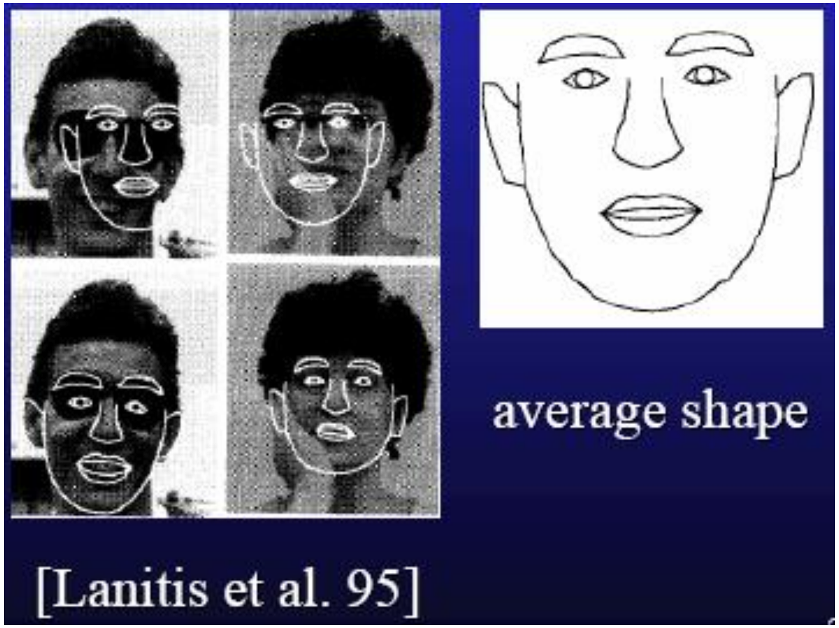
- Bottom-up approach: Detect facial features (eyes, nose, mouth, etc) first
- Facial features: edge, intensity, shape, texture, color, etc
- Aim to detect invariant features
- Group features into candidates and verify them

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

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



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)

Appearance-Based Methods: Classifiers

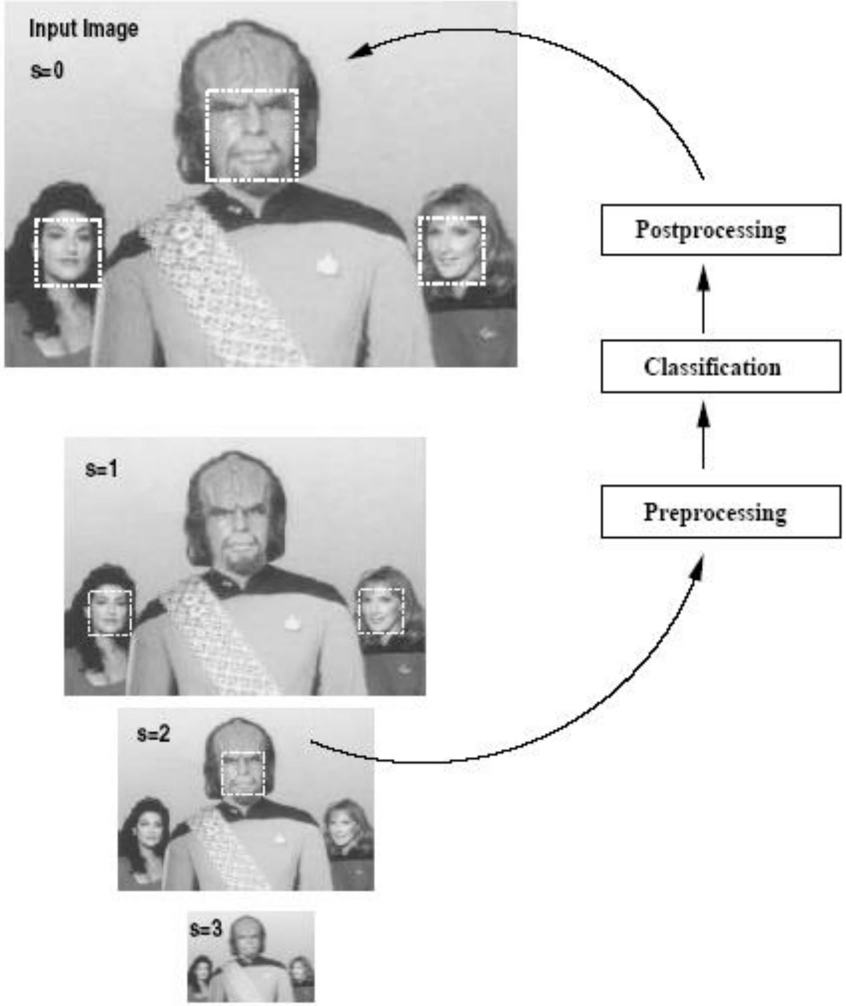
- Neural network
  - Multilayer Perceptrons
- Princiapl Component Analysis (PCA), Factor Analysis
- Support vector machine (SVM)
- Mixture of PCA, Mixture of factor analyzers
- Distribution Distribution-based method
- Naïve Bayes classifier
- Hidden Markov model
- Sparse network of winnows (SNoW)
- Kullback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

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

Exhaustive Search

- Across scales
- Across locations



Theory of Our Algorithm

- The maximum likelihood(ML) test: $L(O) = \frac{P_F(O)}{P_N(O)}$
- In estimating $P_F(O)$ or $P_N(O)$, our assumption is that the permuted version of $O = (o_1, \cdots, o_n)$, $O' = (o_{s_1}, \cdots, o_{s_n})$ comes from a k_{th} order Markov process $X' = (X'_1, \cdots, X'_n) = (X_{s_1}, \cdots, X_{s_n})$ where $s_1, \cdots, s_n$ is a permuted sequence from $1, \cdots, n$.
- The optimal permutation maximizes the Kullback divergence(also known as cross entropy) between $P_F(O')$ and $P_N(O')$, $H_{F||N}(X') = \sum_{O'} P_F(O') \log \frac{P_F(O')}{P_N(O')}$

Theory of Our Algorithm(2)

- k_{th} **order Markov process:** $P(X'_n = o'_n | X'_{n-1} = o'_{n-1}, \ldots, X'_1 = o'_1) = P(X'_n = o'_n | X'_{n-1} = o'_{n-1}, \ldots, X'_{n-k} = o'_{n-k})$
- **Chain rule of the Kullback divergence:** $H_{F||N}(Z_n, \ldots, Z_1) = \sum_{i=n}^{1} H_{F||N}(Z_i | Z_{i-1}, \ldots, Z_1)$
- $H_{F||N}(X') = \sum_{i=1}^{k} H_{F||N}(X_{s_i} | X_{s_{i-1}}, \ldots, X_{s_1}) + \sum_{i=k+1}^{n} H_{F||N}(X_{s_i} | X_{s_{i-1}}, \ldots, X_{s_{i-k}})$.
- **Optimal** $S^* = (s_1^*, \cdots, s_n^*)$**:** $H_{F||N}(X' = X(S^*)) \geq H_{F||N}(X(S)) \quad \forall\, S$

Theory of Our Algorithm(3)

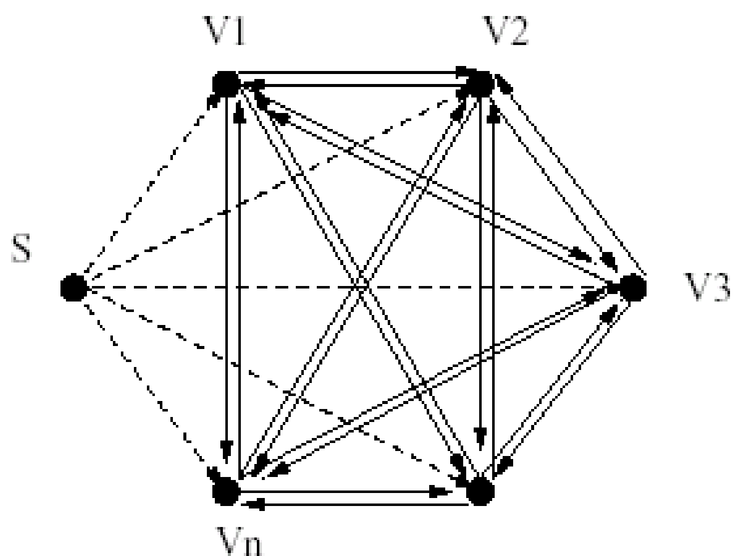
- An NP-Complete "travelling salesman problem"(see the next sub-section for more detail) searching for a Hamiltonian path.
- A first-order Markov process where the ML ratio test $L(O)$, or equivalently $L(O')$is simplified to the following form:

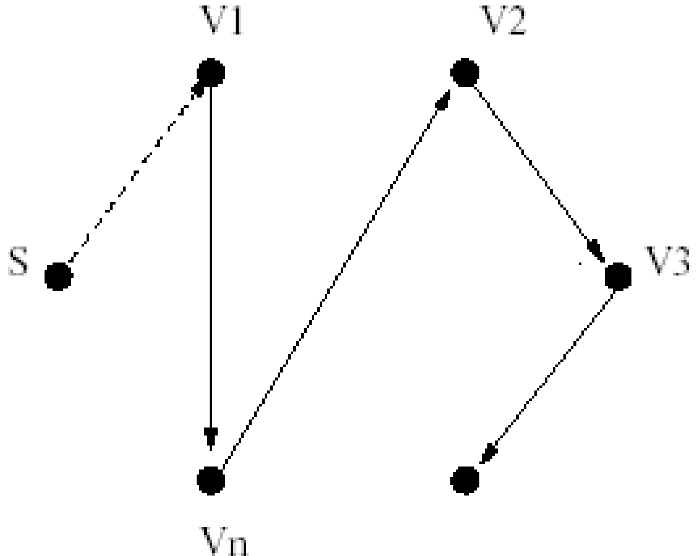
$$L(O') = \frac{P_F(O')}{P_N(O')} = \frac{(\prod_{i=n}^{2} P_F(o_{s_i^*} | o_{s_{i-1}^*}))P_F(o_{s_1^*})}{(\prod_{i=n}^{2} P_N(o_{s_i^*} | o_{s_{i-1}^*}))P_N(o_{s_1^*})}. \quad (1)$$

Instance of the "Travelling Salesman Problem"

- **When** $k = 1$,

$$H_{F||N}(X') = H_{F||N}(X_{s_1}) + \sum_{i=2}^{n} H_{F||N}(X_{s_i}|X_{s_{i-1}}). \qquad (2)$$



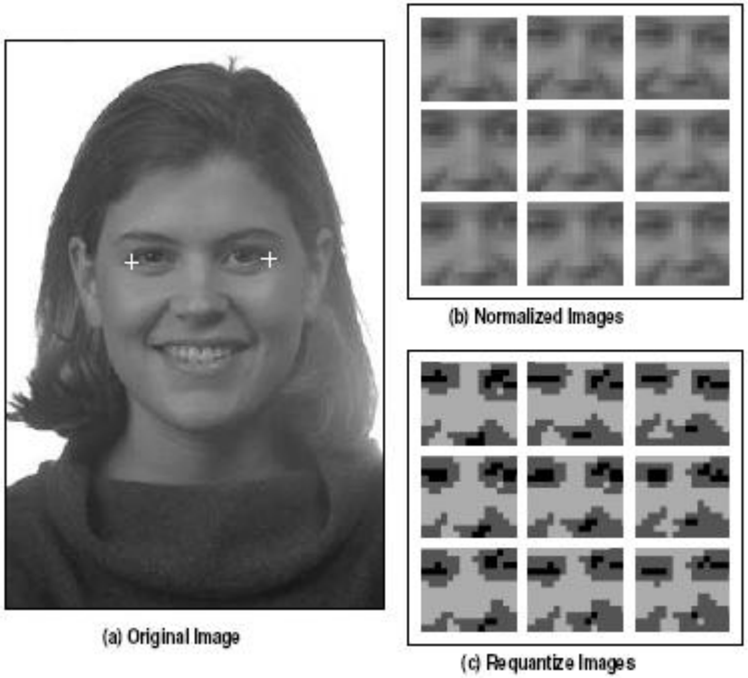


Intuition of Permutation

- When modelling face images as a k-th order Markov process, rows of the images are concatenated into long vectors. The pixels corresponding to the semantics(e.g, eyes, lips) will be scatted into different parts in the vectors. The Markovian property is not easy to be justified.
- If some permutation can be found to re-group those scattered pixels(i.e, to put all the pixels corresponding to eyes together, those for lips together), then the Markov assumption is more reasonable.

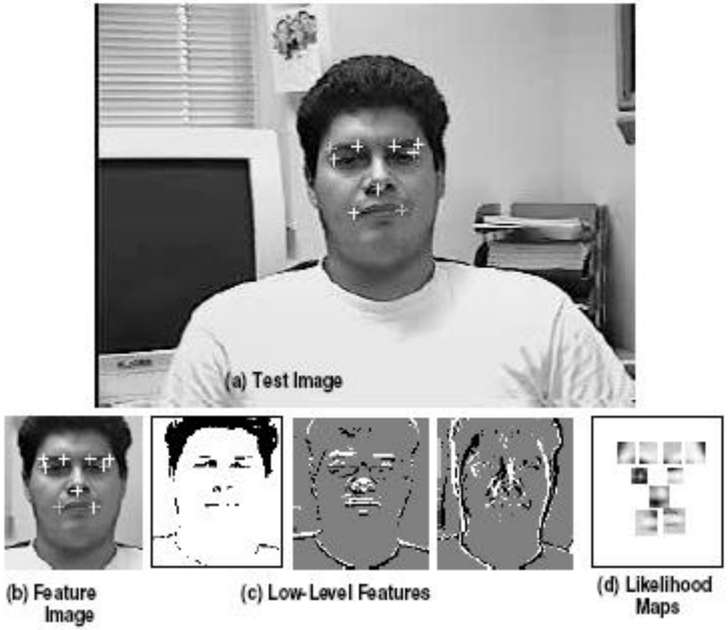
Preprocessing

- Rotation
- Scaling
- Quantizing



Facial Features Detection

- Region search



FERET Database

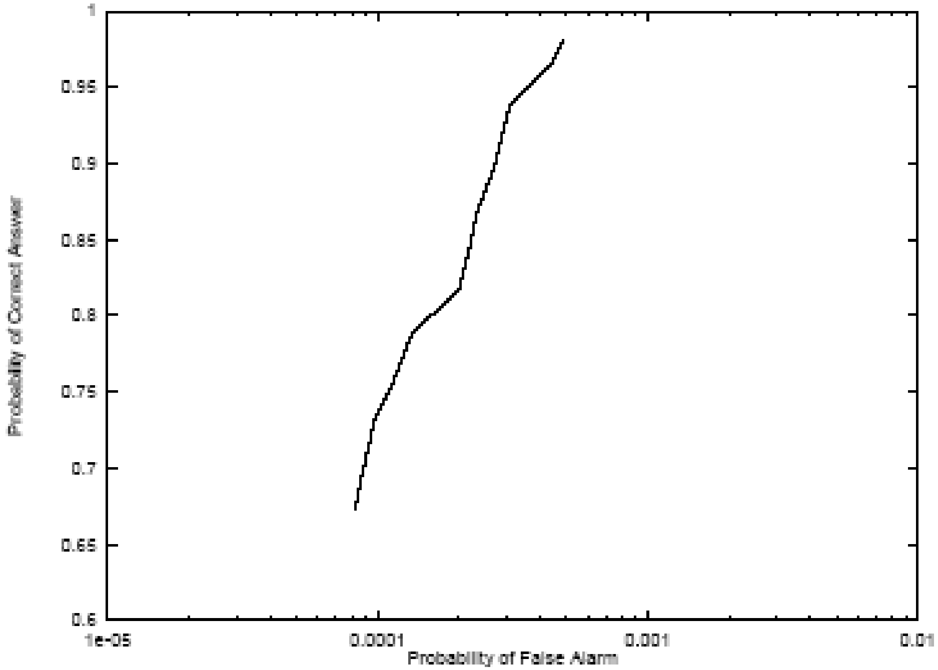
- Training data

Face and Facial Feature Detection

- The algorithm is also used to detect 9 facial features: 2 outer mouth corners, 2 outer eye corners, 2 outer eye-brow corners, 2 inner eye-brow corners and the center of the nostrils.

Evaluations

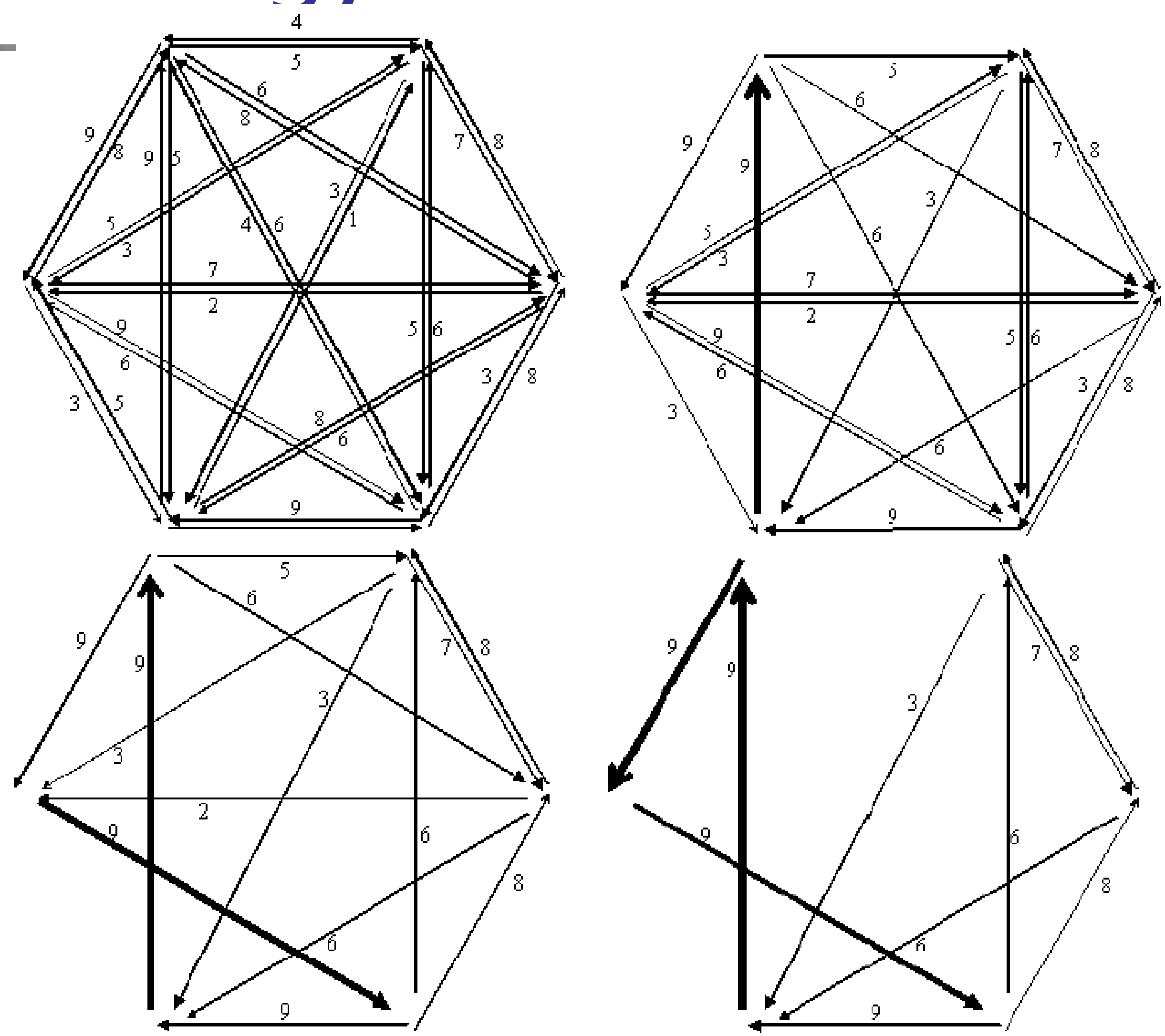
- ROC curve



Results

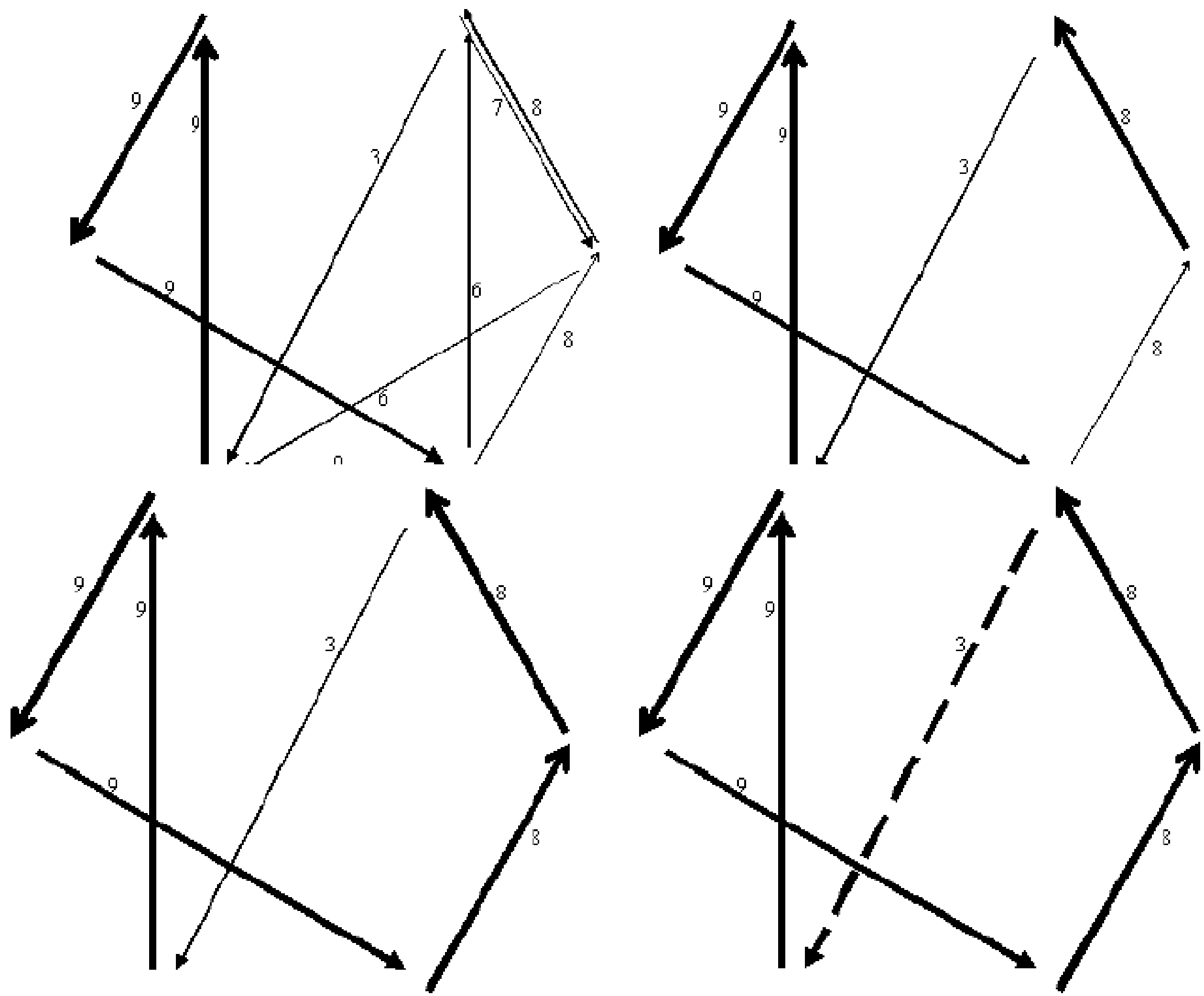
Search Strategy

- Kruskal

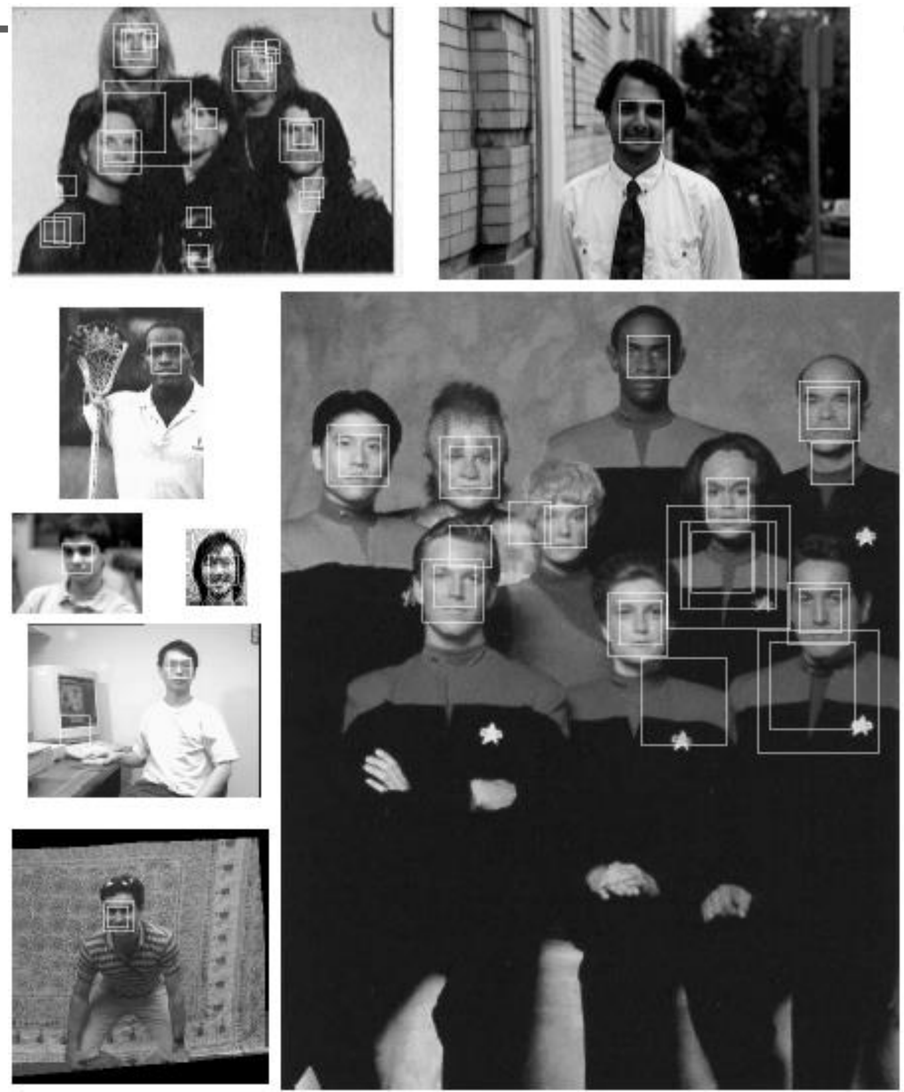


Search Strategy

- Kruskal



Detection Results



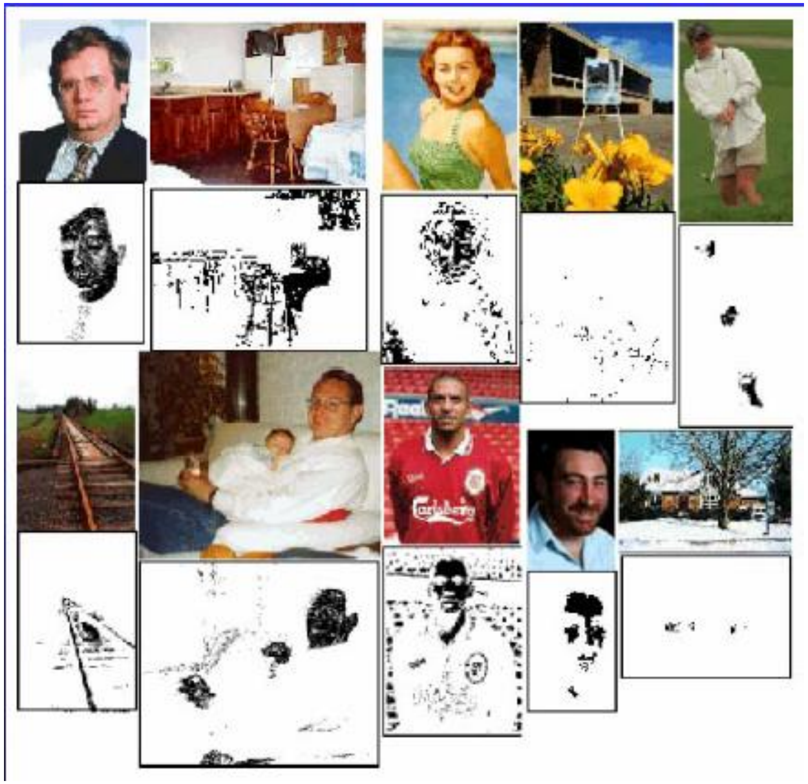
Side-View Face Detection

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

Color-Based Face Detector

- 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 tone line regions)



What is Face Recognition?

- A set of two task:
  - Face Identification: Given a face image that belongs to a person in a database, tell whose image it is.
  - Face Verification: Given a face image that might not belong to the database, verify whether it is from the person it is claimed to be in the database.

Difference between Face Detection and Recognition

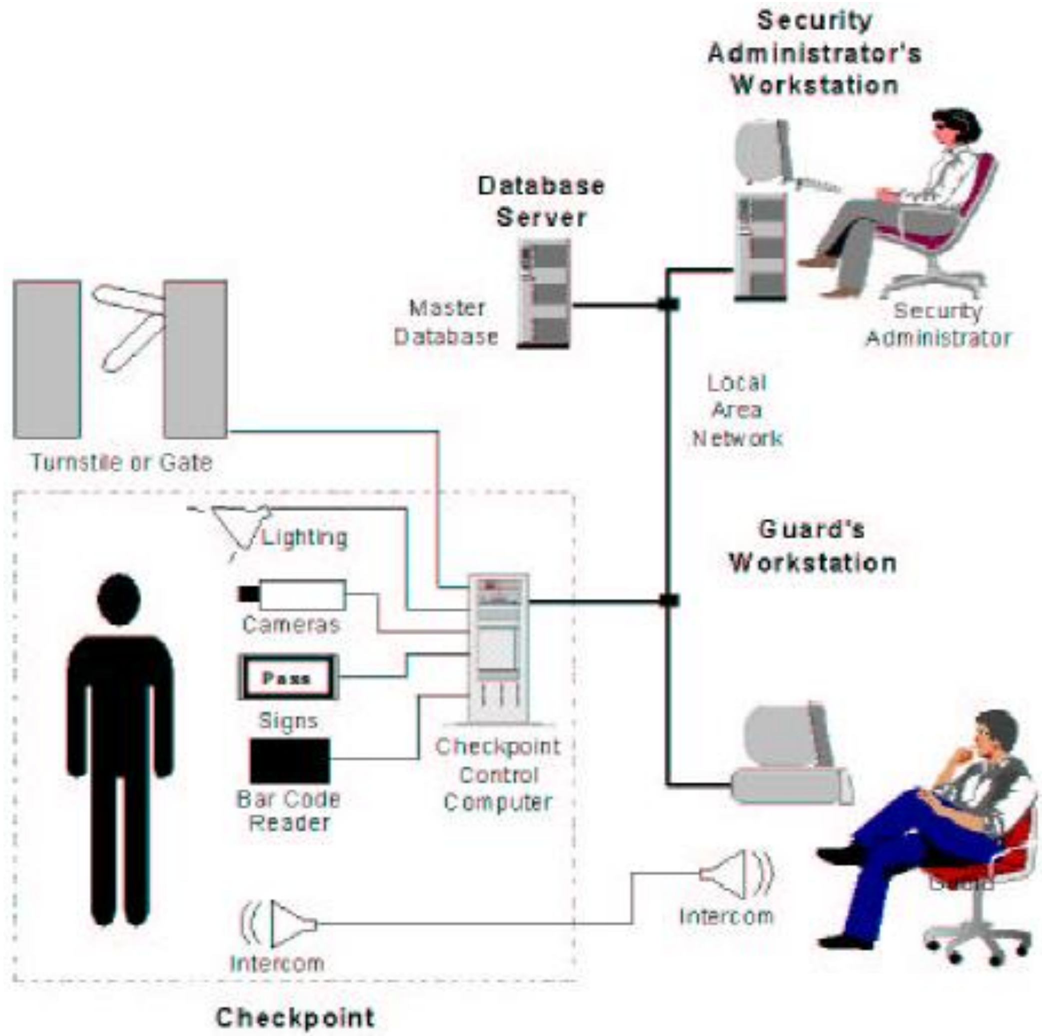
- Detection – two-class classification
  - Face vs. Non-face
- Recognition – multi-class classification
  - One person vs. all the others

Applications of Face Recognition

- Access Control
- Face Databases
- Face ID
- HCI - Human Computer Interaction
- Law Enforcement

Applications of Face Recognition

- Multimedia Management
- Security
- Smart Cards
- Surveillance
- Others



Different Approaches

- Features:
  - Features from global appearance
    - Principal Component Analysis(PCA)
    - Independent Component Analysis(ICA)
  - Features from local regions
    - Local Feature Analysis(LFA)
    - Gabor Wavelet
- Similarity Measure
  - Euclidian Distance
  - Neural Networks
  - Elastic Graph Matching
  - Template Matching
  - …

The PCA Approach - Eigenface

- The theory

$$\Psi = \frac{1}{M}\sum_{n-1}^{M}\Gamma_n$$

$$\Phi_i = \Gamma_i - \Psi$$

$$C = \frac{1}{M}\sum_{n-1}^{M}\Phi_n\Phi_n^T = AA^T$$

$$L = A^T A \quad L_{n,m} = \Phi_m^T\Phi_n$$

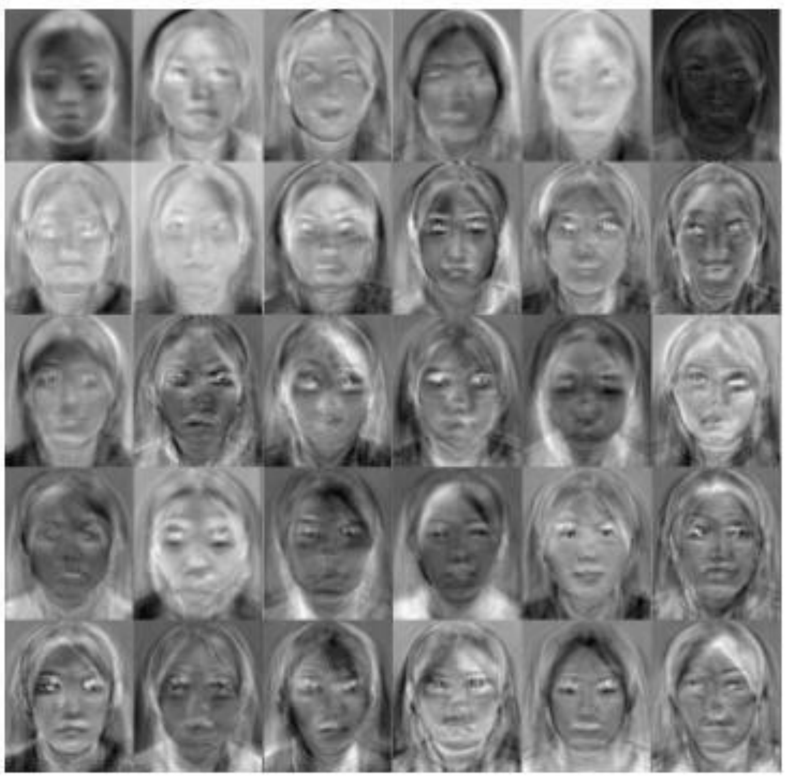
$$u_l = \sum_{k-1}^{M} v_{lk}\Phi_k \quad l = 1, \ldots, M$$

$$\omega_k = u_k^T(\Gamma_{\text{new}} - \Psi) \quad k = 1 \ldots M'$$

$$\Omega_{\text{new}}^T = \left[\begin{array}{cccc} \omega_1 & \omega_2 & \ldots & \omega_{M'} \end{array}\right]$$

The PCA Approach - Eigenface

- Eigenfaces – an example

Face Detection + Recognition

- Detection accuracy affects the recognition stage
- Key issues:
  - Correct location of key facial features(e.g. the eye corners)
  - False detection
  - Missed detection

A Demonstration

- https://www.youtube.com/watch?v=K4u4Dpl6NKk
- https://www.youtube.com/watch?v=PL3xJErjEgU