Face Detection

Vikas.Chandrakant.Raykar
Prithu Sinha
University of Maryland,CollegePark

Goal

- To detect and localize human faces in any given grayscale/color image.
- Challenges: Invariant to
  - different illumination conditions
  - pose
  - camera orientation

Applications

- Face Detection is the first crucial step in face recognition, face tracking, pose estimation and expression recognition.
- Surveillance
- Video indexing/summarization, especially for new broadcasts and videos.

Important Survey papers

- Ming-Hsuan Yang, David J. Kriegman, Narendra Ahuja, "Detecting Faces in Images: A Survey" in IEEE Transactions on Pattern Analysis and Machine Intelligence (January 2002), Vol. 24, No. 1

- Henry A. Rowley, Shumeet Baluja, Takeo Kanade, "Neural Network-Based Face Detection" in IEEE Transactions on Pattern Analysis and Machine Intelligence (January 1998), Vol. 20, No. 1

Two Approaches:

- Feature based

  No training required
- Image based

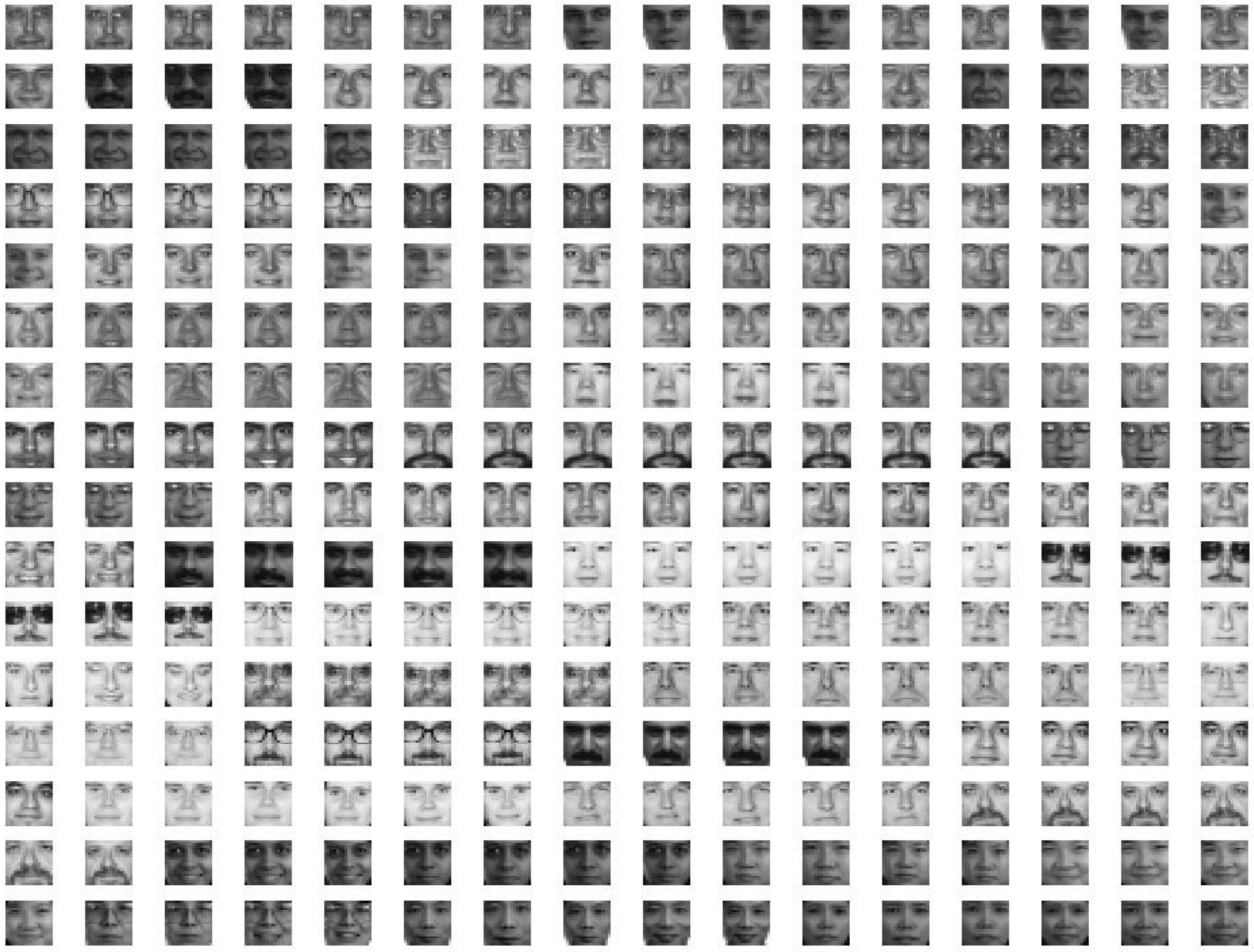
  Formulate as a two class problem face vs nonface.Difficulty in getting all nonfaces

Database

- CBCL Face Database #1
  (MIT Center For Biological and Computation Learning)
- Training set : 2,429 faces, 4,548 non-faces
- Test set : 472 faces, 23,573 non-faces
- 19X19 grayscale images as pgm files
- Test set both frontal as well as non frontal and rotated faces.

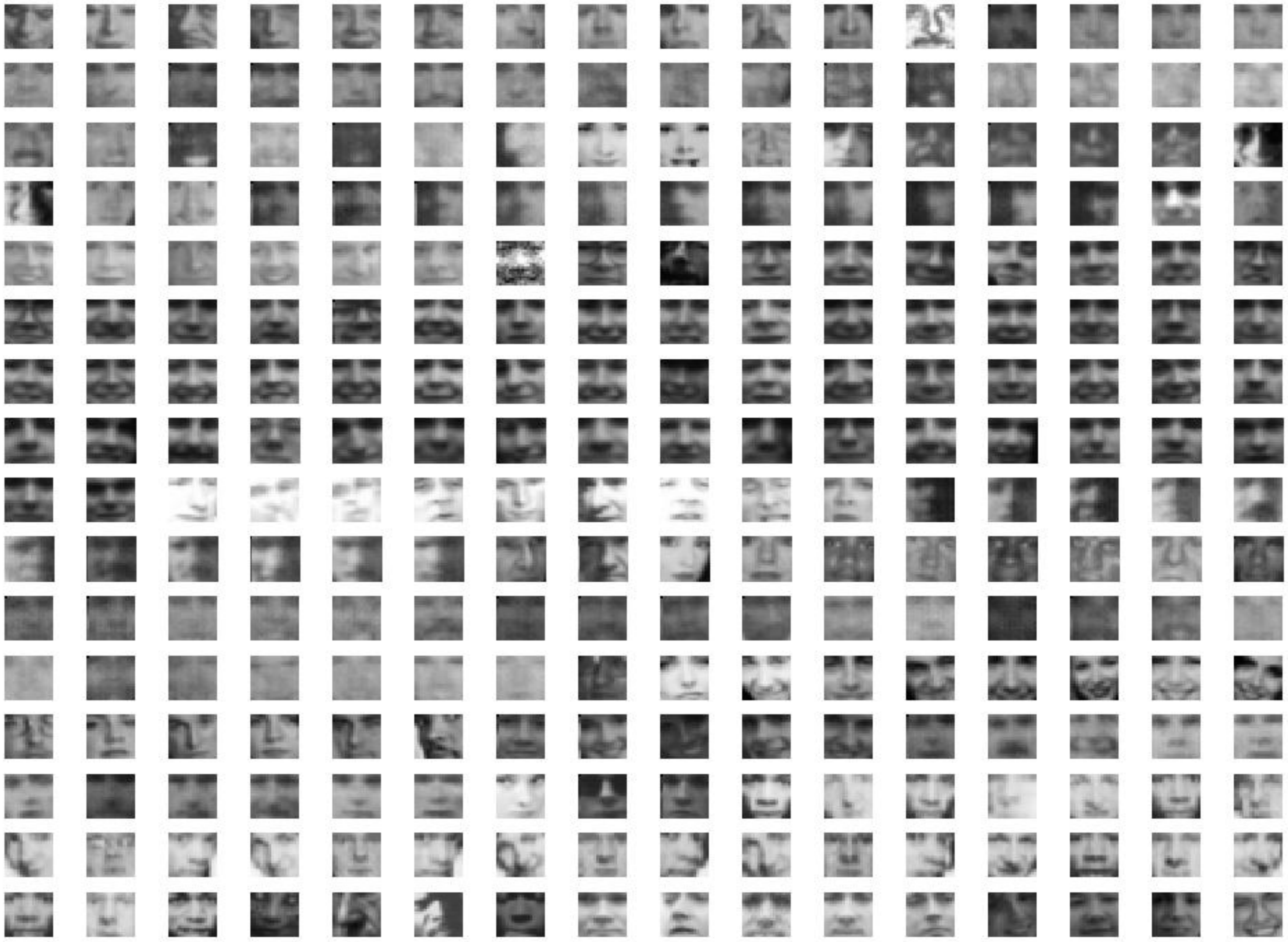
http://www.ai.mit.edu/projects/cbcl

Training face database

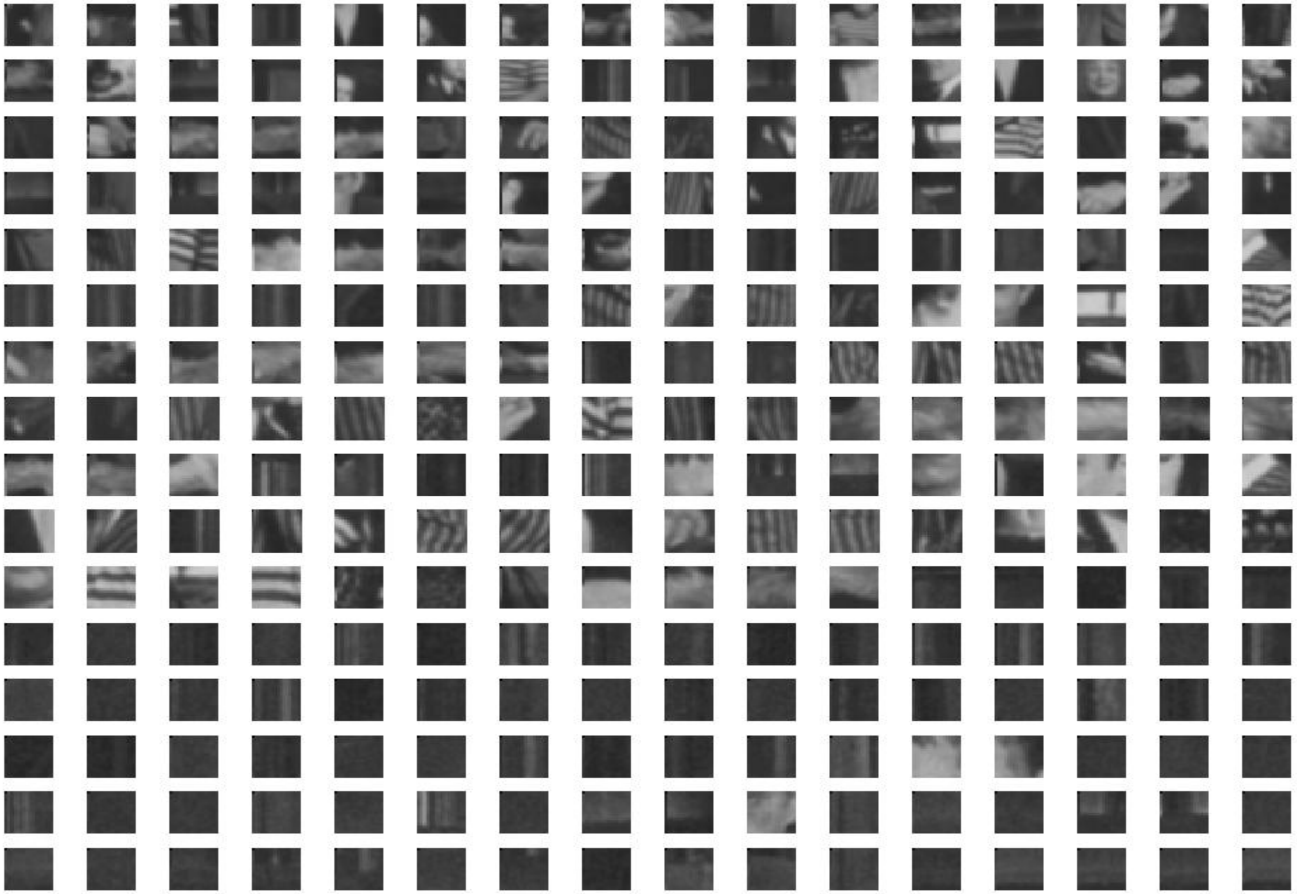


Training nonface database

Test Face database



Test nonface database



Evaluation criterion

- Pd Detection probability
- Pf False alarm (a nonface detected as a face)
- Pe Error probability
- $Pe=0.5*(Pf+(1-Pd)$
- Ideally require high Pd and low Pf
- Compromise between Pd and Pf

Our Approach

- Preprocessing
- Feature selection (entire image,PCA,KPCA)
- Training (NN)
- Classification (NN,KNN)
- Discriminant Analysis(LDA,KLDA,BDA,KBDA)
- Adaboost
- Color based approaches

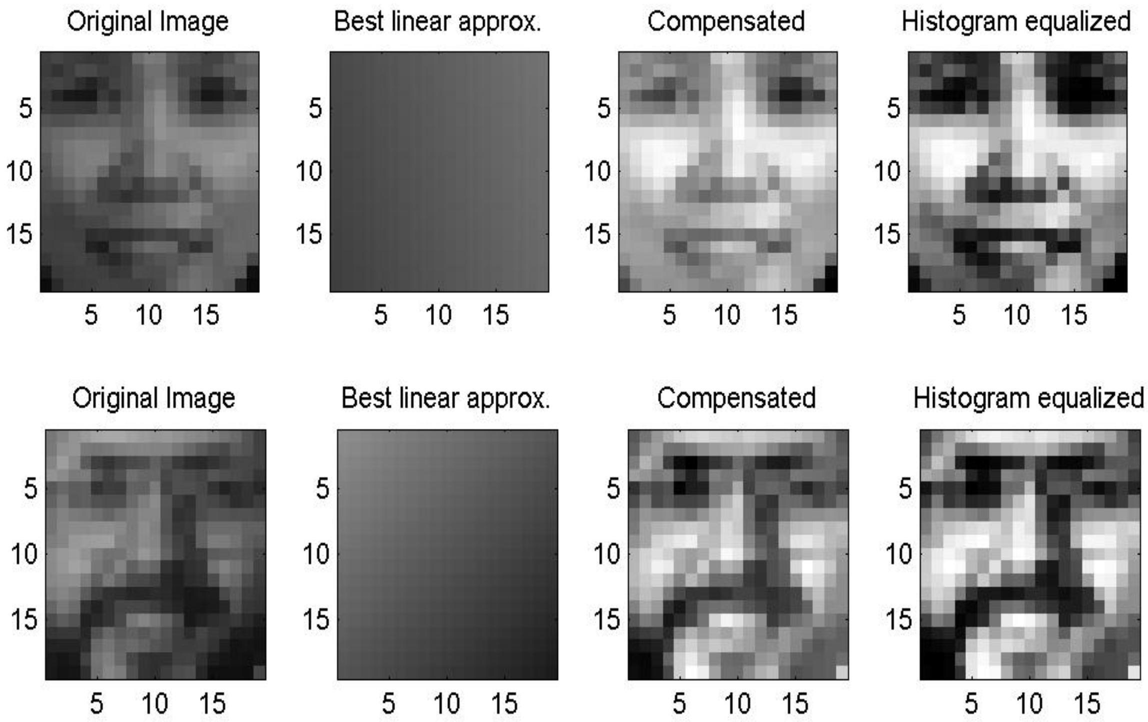
Preprocessing

- Database has already cropped images
- Lighting Compensation

  Subtract the best linear approximation

  of the image
- Histogram equalization to improve contrast

Example



Original Image
Best linear approx.
Compensated
Histogram equalized
5
10
15
5 10 15
Original Image
Best linear approx.
Compensated
Histogram equalized
5
10
15
5 10 15

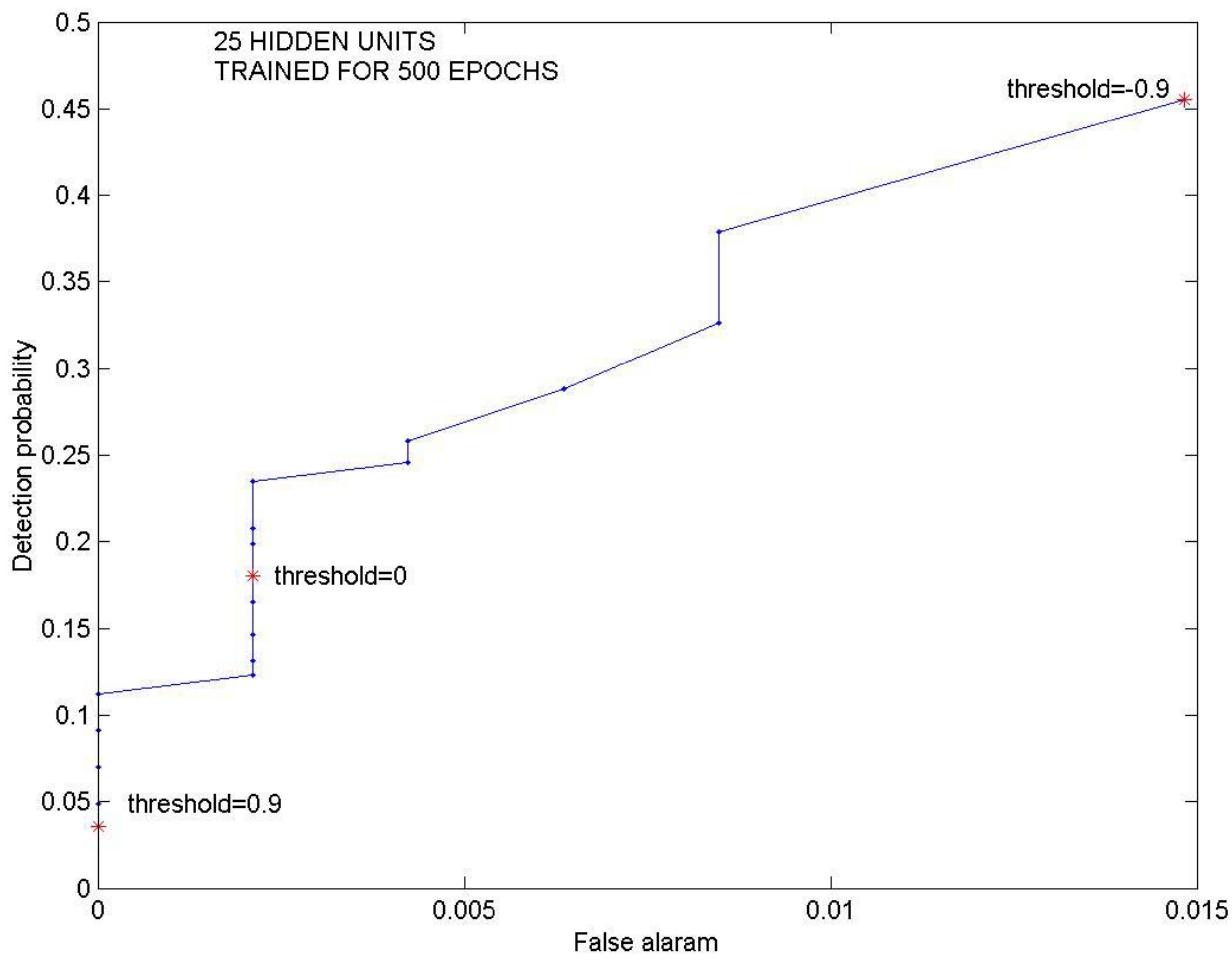
Neural Network

- Vectorize the image and use the 361 element vector as an input to the NN
- 3 layer network (h hidden units)
- One output unit(1 face/ -1 nonface)
- sigmoid activation function
- Training : Gradient descent with momentum

Results

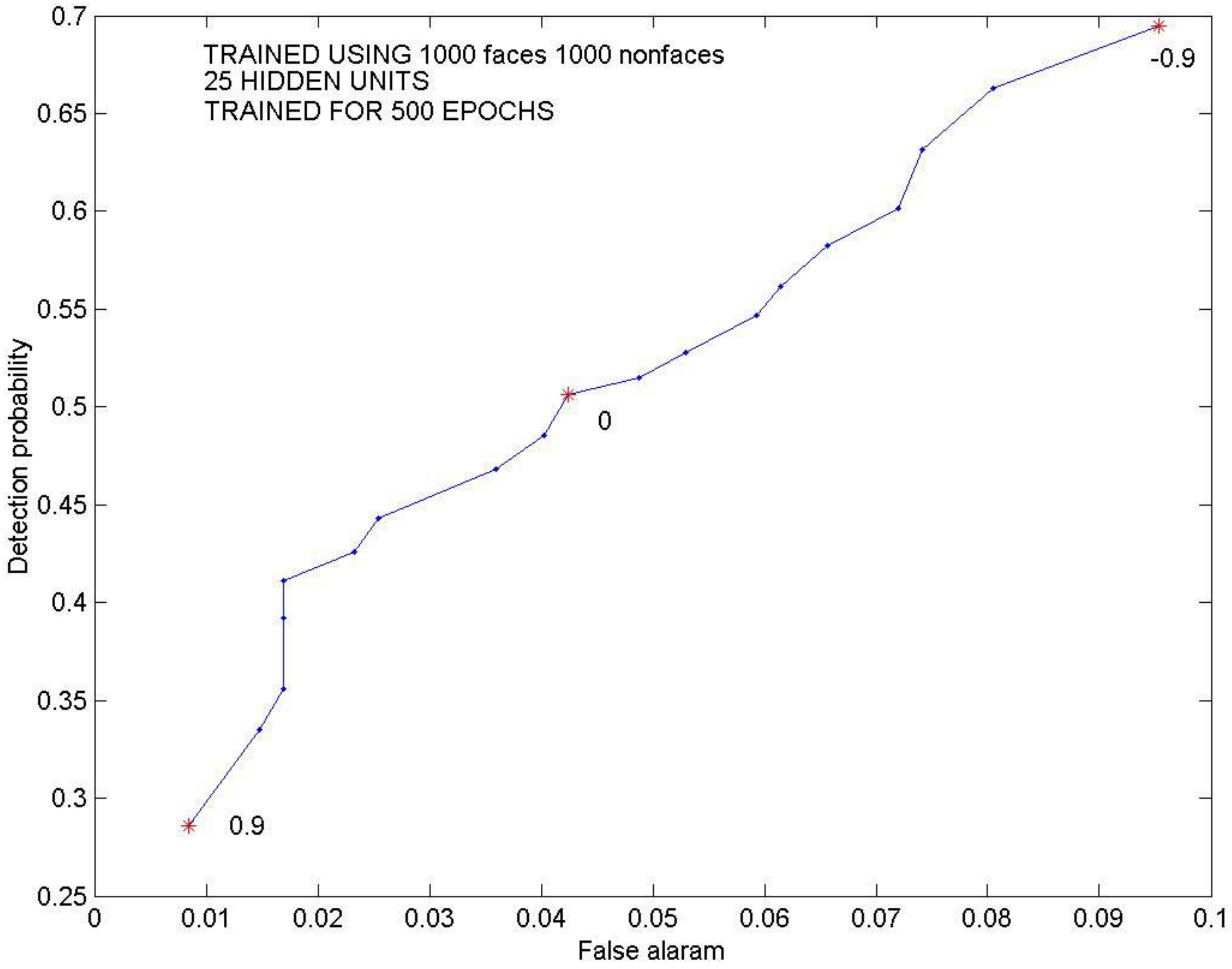
- Trained using 2,429 faces 4,548 non-faces
- Tested on 472 faces and 472 nonfaces
- H=25 hidden units
- 500 epochs(rate=0.1 momentum=0.8)

ROC-varying the threshold

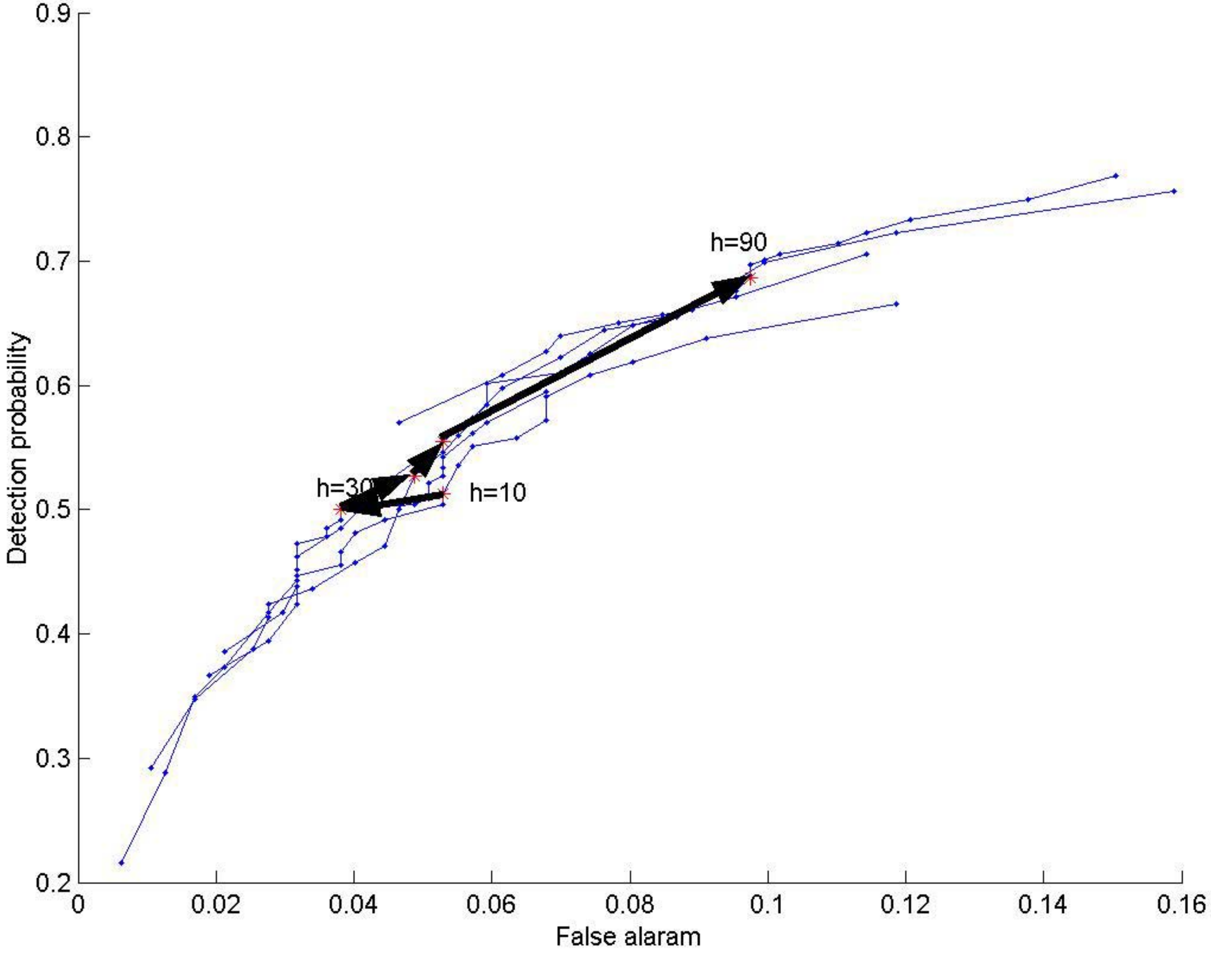


Comments

- The network is biased towards nonfaces since the number of nonfaces is more.
- We can get better detection probability is the number of faces is more than the number of nonfaces used in training. However the false alarm increases.
- We want high detection probability as we can reduce false alarms by certain heuristics.


TRAINED USING 1000 faces 1000 nonfaces
25 HIDDEN UNITS
TRAINED FOR 500 EPOCHS
-0.9
0
0.9
Detection probability
False alaram
0.7
0.65
0.6
0.55
0.5
0.45
0.4
0.35
0.3
0.25
0
0.01
0.02
0.03
0.04
0.05
0.06
0.07
0.08
0.09
0.1

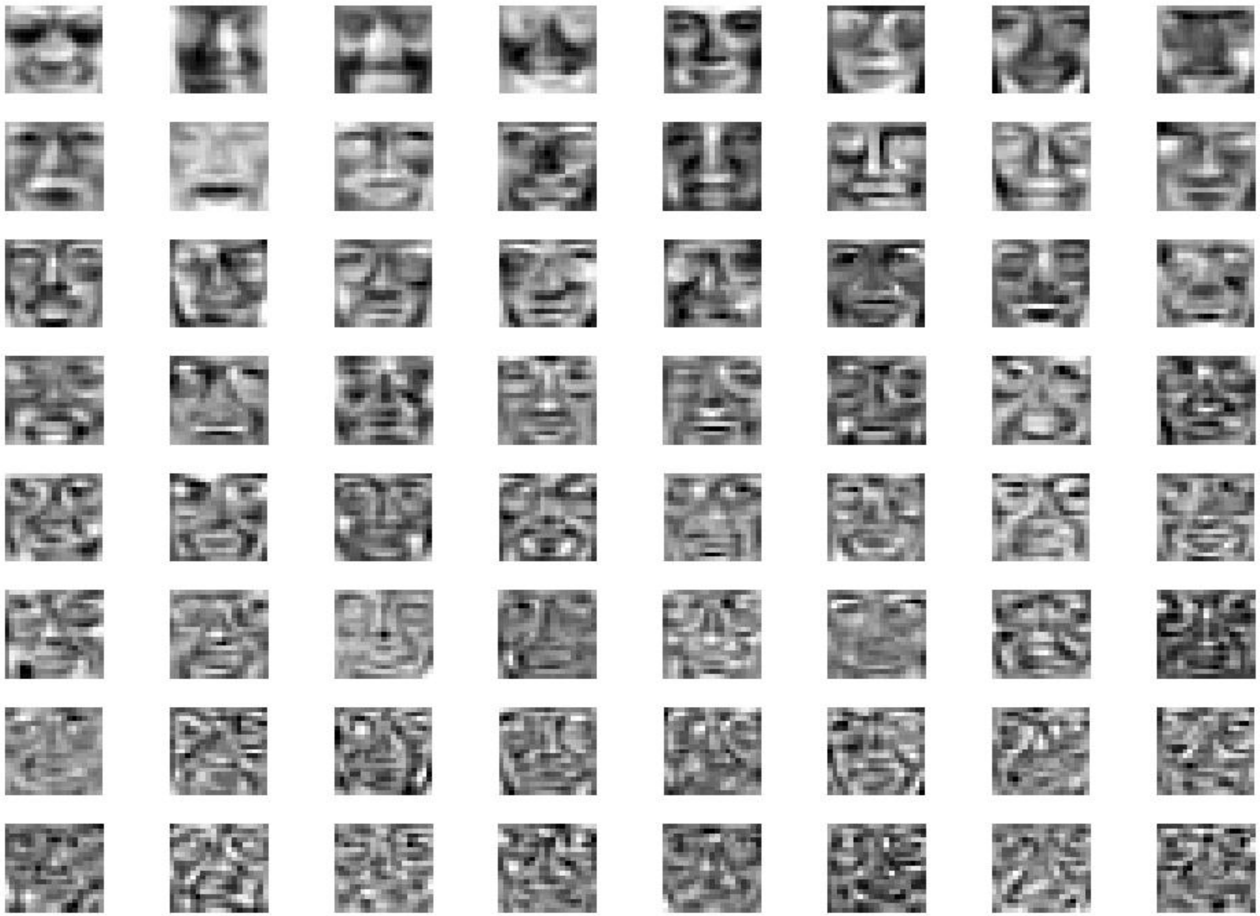
Effect of num of hidden layers



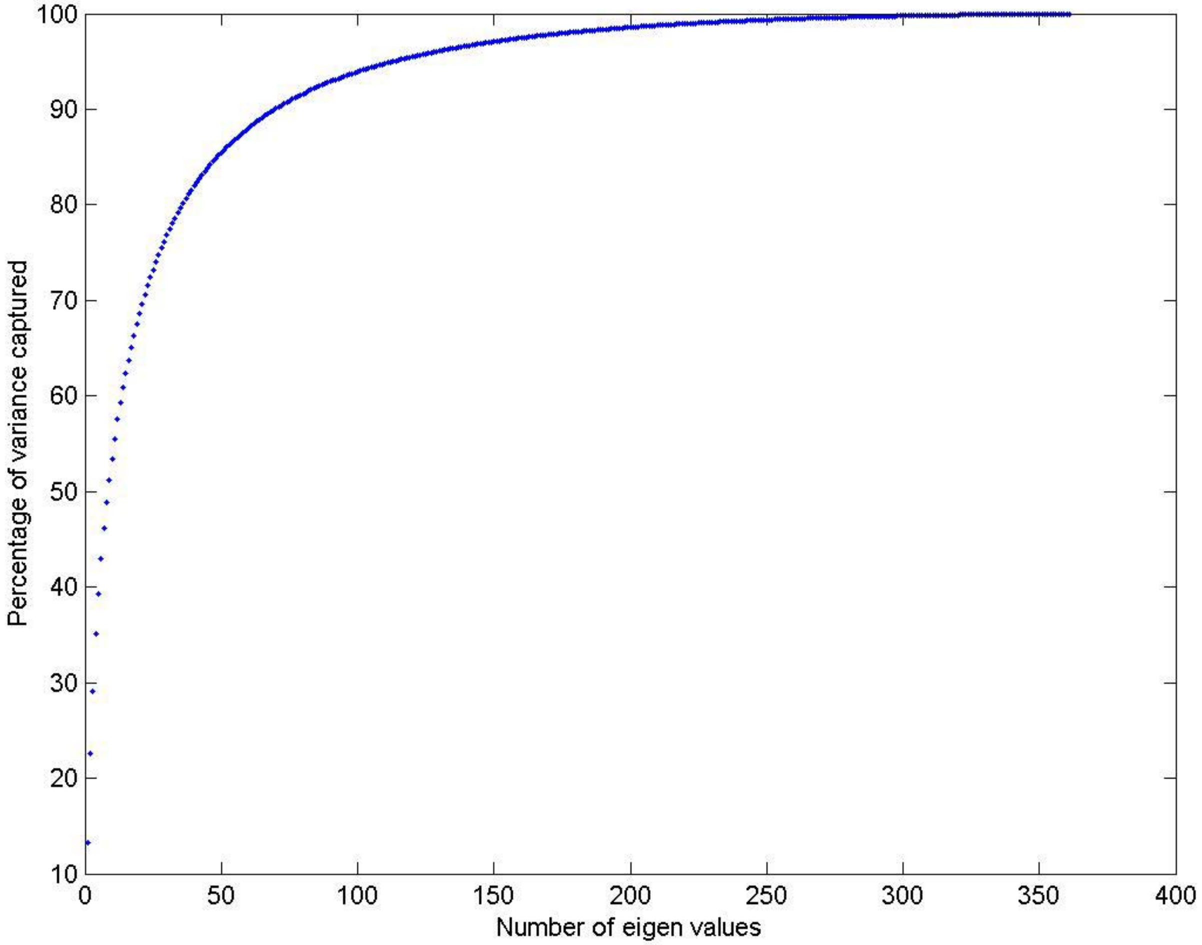
Principal Component Analysis (PCA)

- Subtract the mean
- Compute the scatter matrix
- Find the eigen values and their corresponding eigen vectors
- Select c largest to capture the desired variance
- Use the projections on the eigenvectors
- Did PCA only on the faces

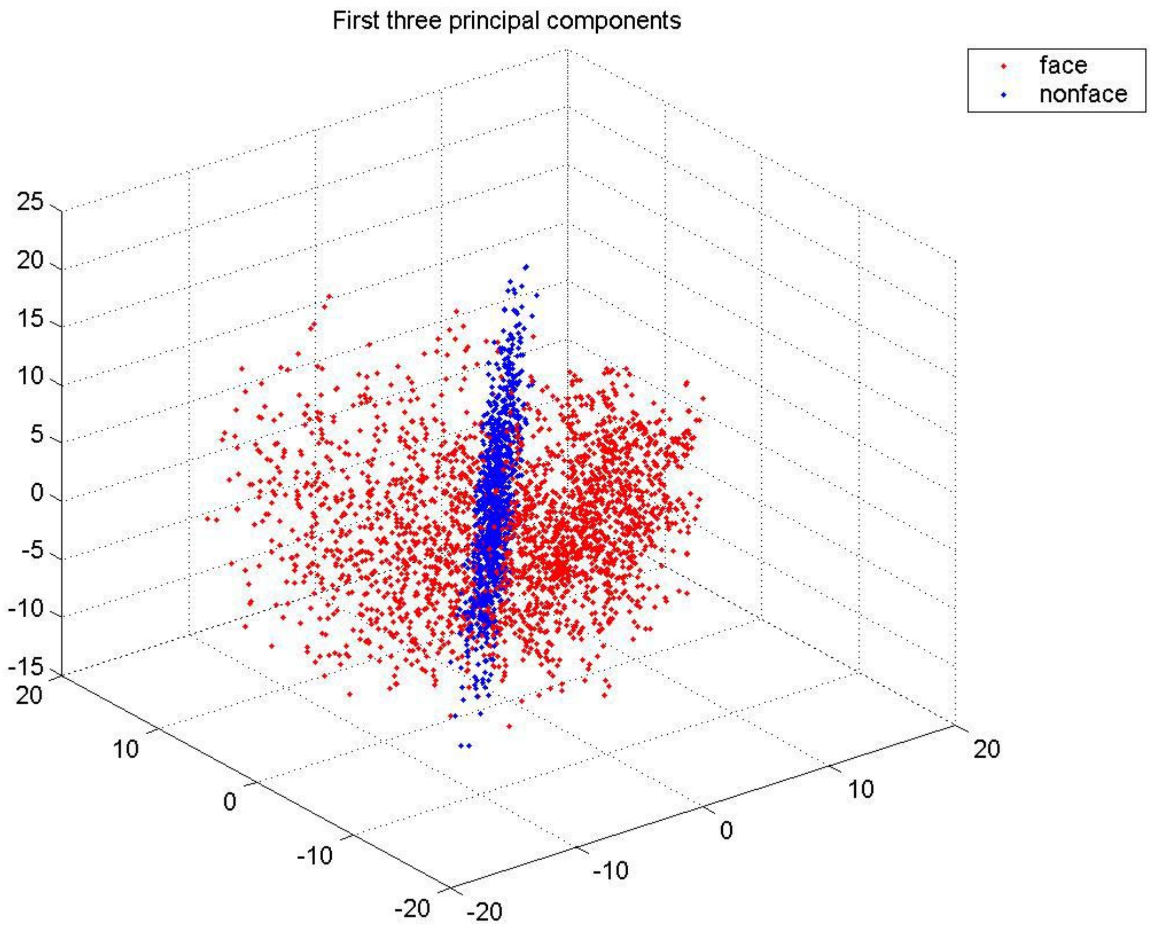
Eigen Faces



Percentage of variance captured



First three principal components



Classification based on PC’s

- K Nearest Neighbour(KNN)
  - 100 components to capture 95% of the variance
  - Classification based on one nearest neighbour
- Train a Neural network

Results

Threshold= 0 1000 faces 1000 nonfaces	Pd	Pf	Pe
PCA KNN	0.6624	0.0828	0.2102
PCA NN	0.6897	0.0938	0.2025

PCA KNN vs PCA NN

- PCA KNN Need to store all the training samples and compare with the test image. Can also use the mean face and nonface
- However PCA NN gives better results than PCA KNN

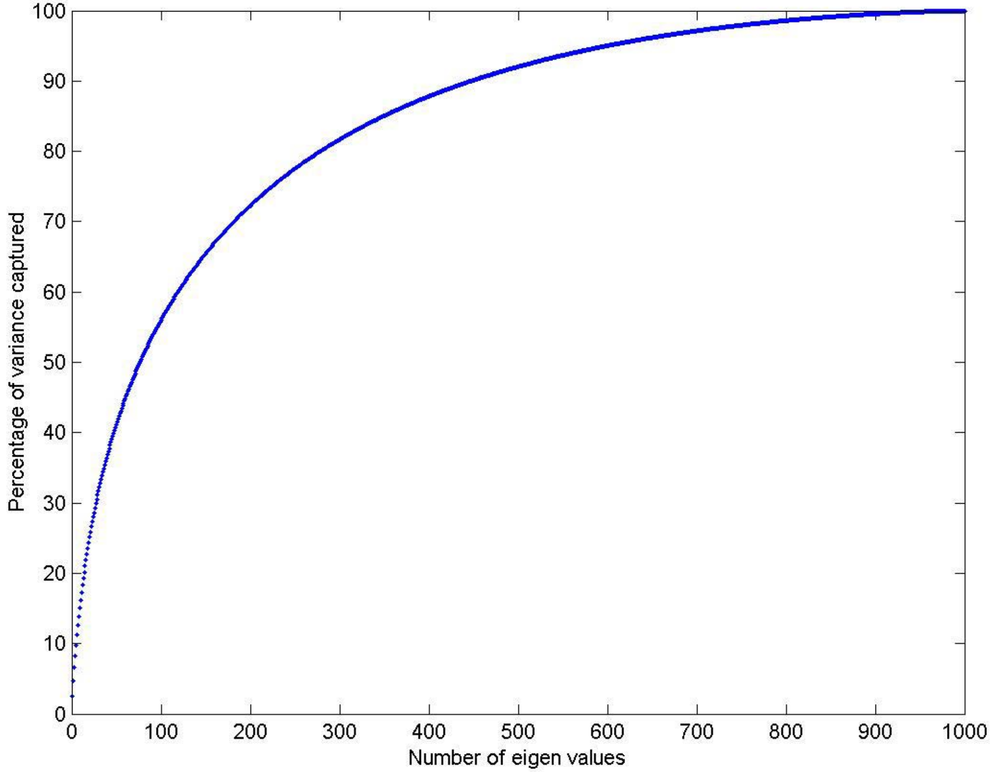
Kernel PCA

- PCA is linear
- Uses only second order statistics
- Can do PCA in feature space
- Express dot product in feature space in terms of kernel functions in the input space

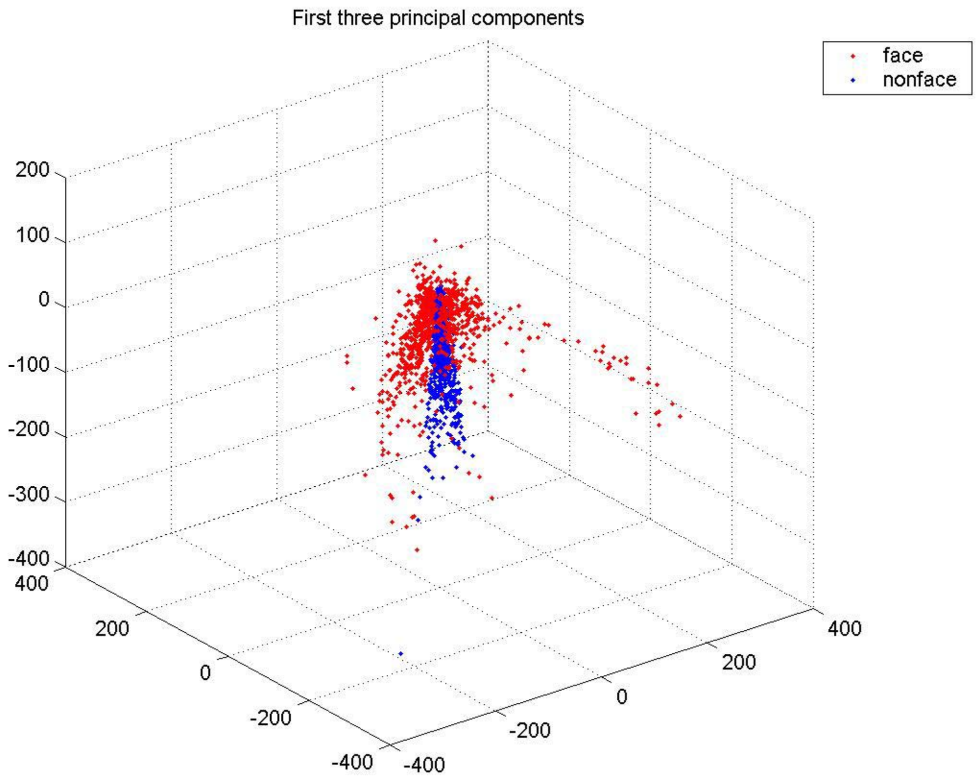
Kernels used

- Gaussian
- Polynomial

Percentage of Variance captured (poly 2 kernel)



First three KPC's



Results

	Pd	Pf	Pe
KPCA KNN			
KPCA NN			

PCA vs KPCA

	Pd	Pf	Pe
PCA-KNN	0.6624	0.0828	0.2102
PCA-NN	0.6897	0.0938	0.2025
KPCA-KNN			
KPCA-NN			

Comments

- KPCA gives lower performance than PCA(???)
- KPCA is computationally more intensive

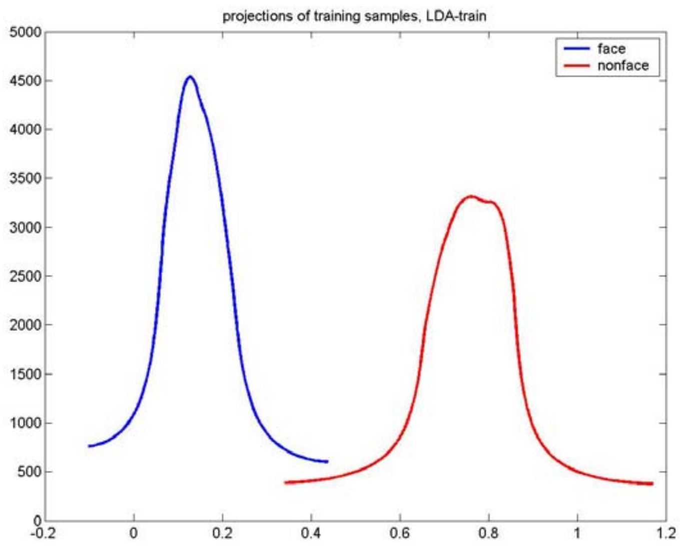
Linear Discriminant Analysis

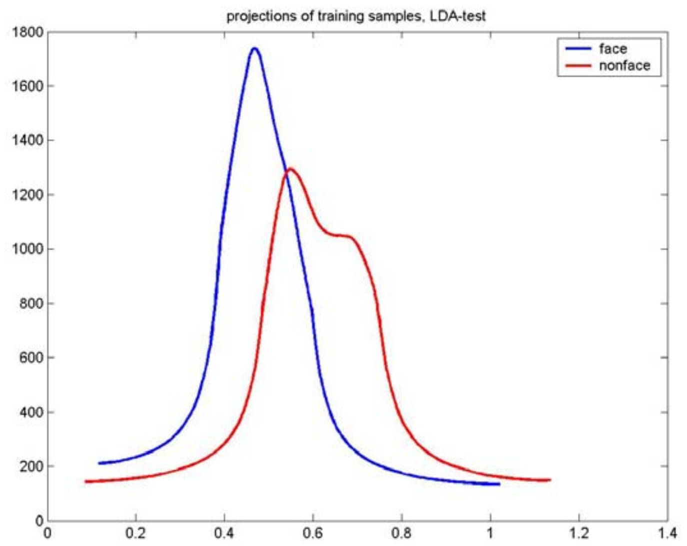
- PCA is unsupervised..so features found by PCA need not be discriminating among the classes
- LDA finds the direction which maximizes the distance between the projected means and minimizes the within class scatter

LDA

- Equations

Projections-LDA

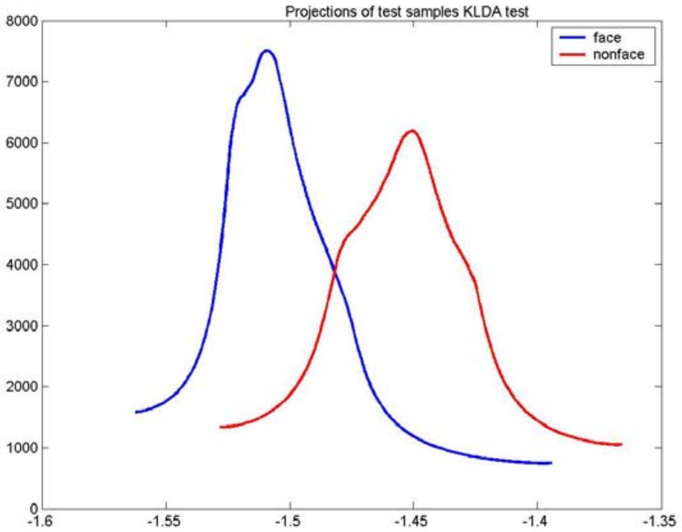


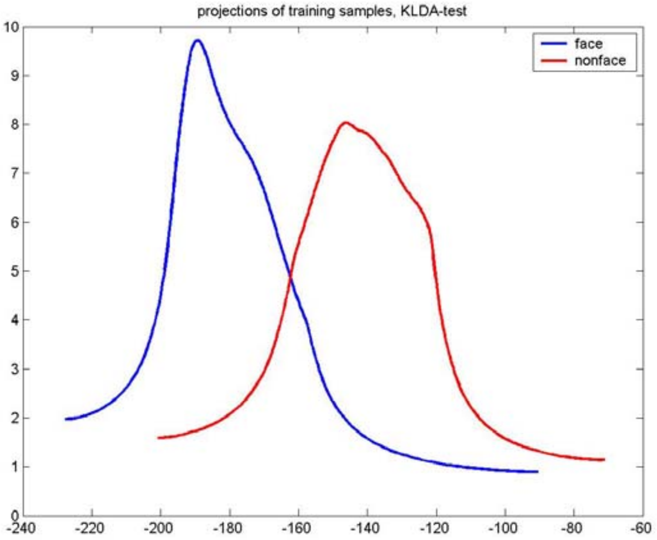


KLDA

- LDA in feature space
- Use kernels to compute the dot products

Projections(





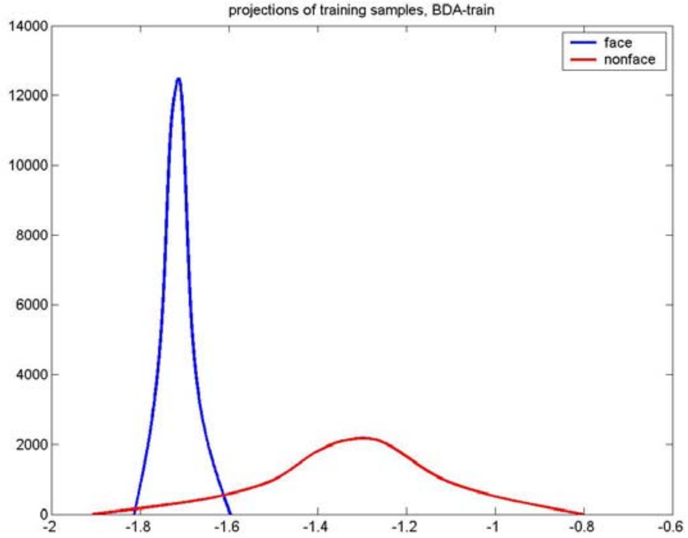
LDA vs KLDA

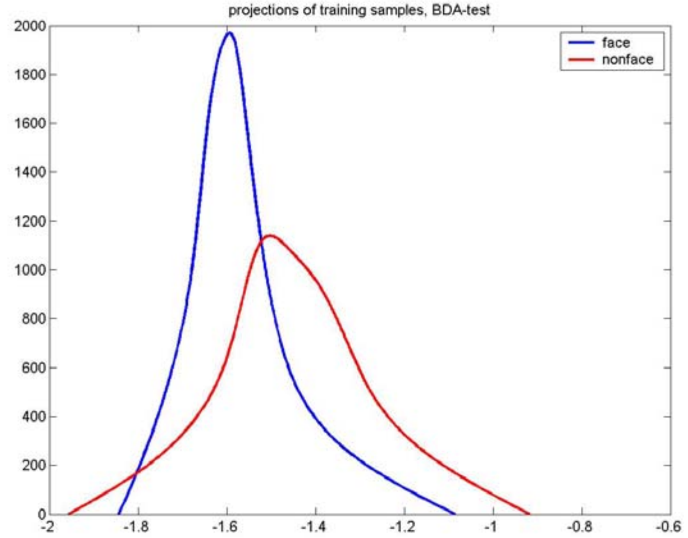
	Pd	Pf	Pe
LDA	0.3715	0.1274	0.3779
KLDA-GAUS			
KLDA-POLY			

Biased Discriminant Analysis BDA

- Push all the nonfaces as far away from the face
- Minimize the within class scatter for face only

Projections





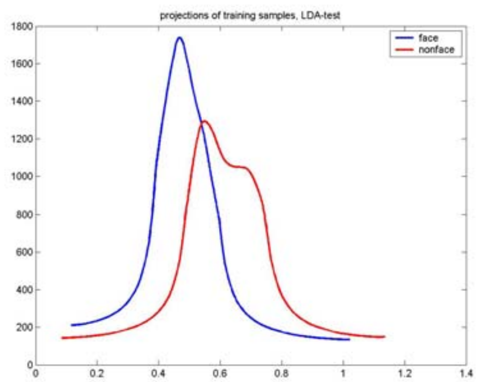
KBDA

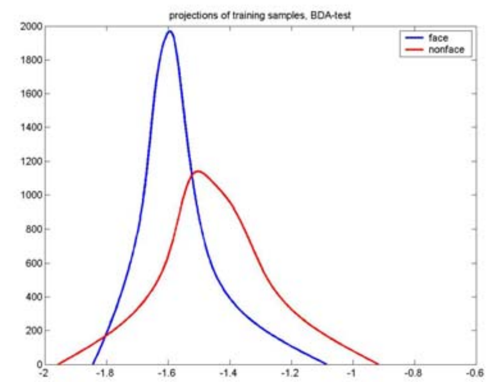
- BDA if feature space
- Used Gaussian and second degree polynomial kernels

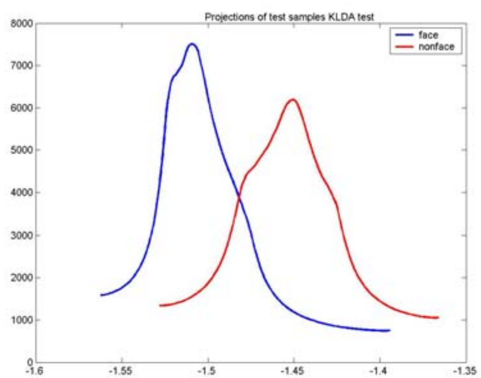
BDA/KBDA

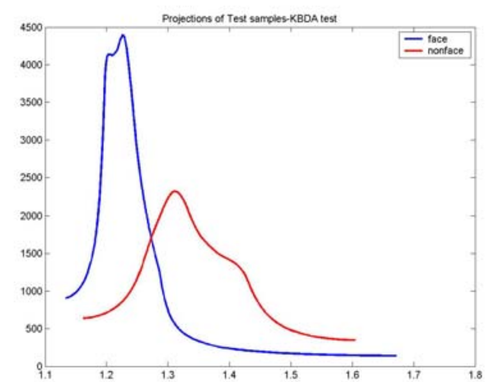
	Pf	Pd	Pe
BDA	0.6		
KBDA poly2			
KBDA gaus			

LDA/BDA/KLDA/KBDA





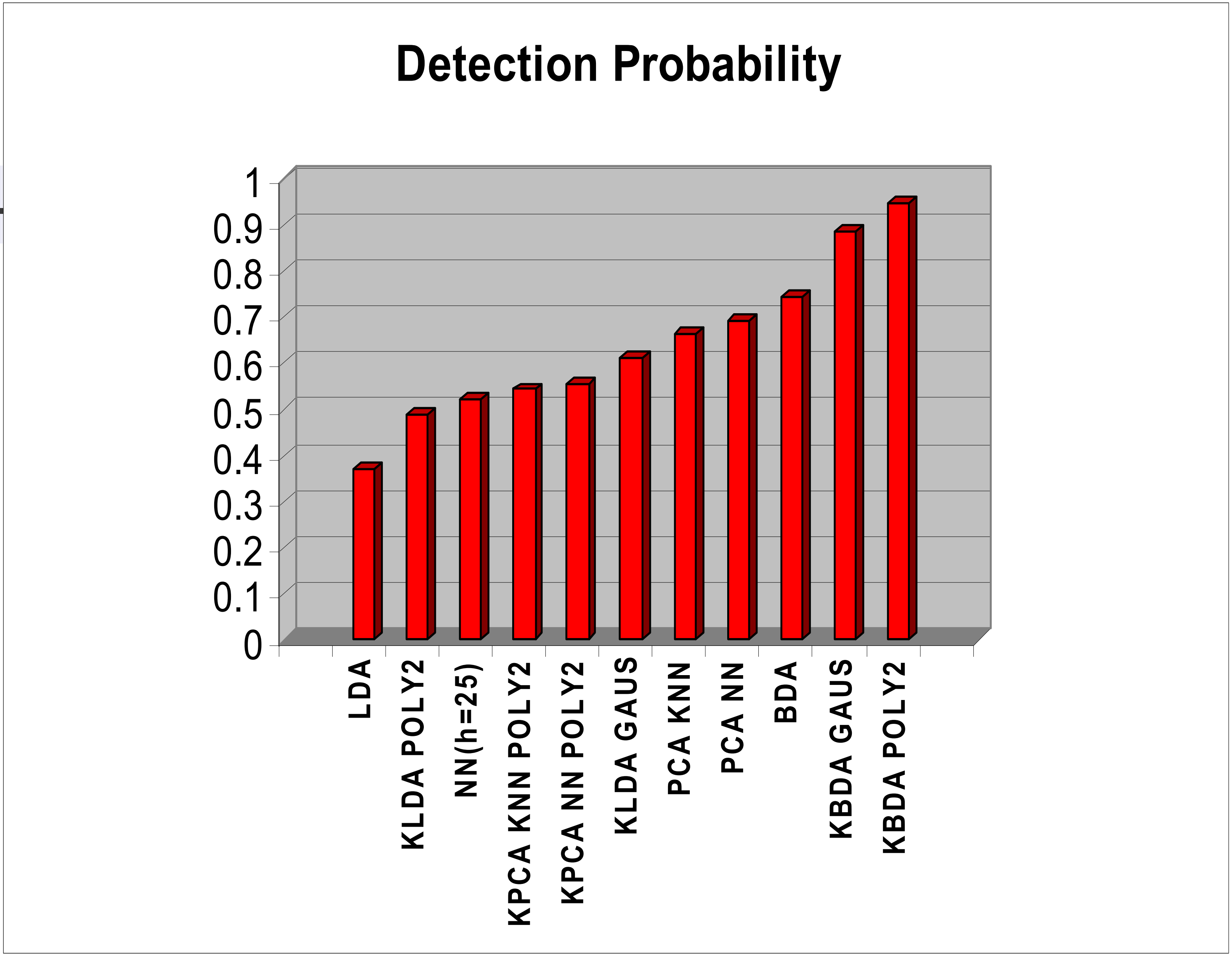




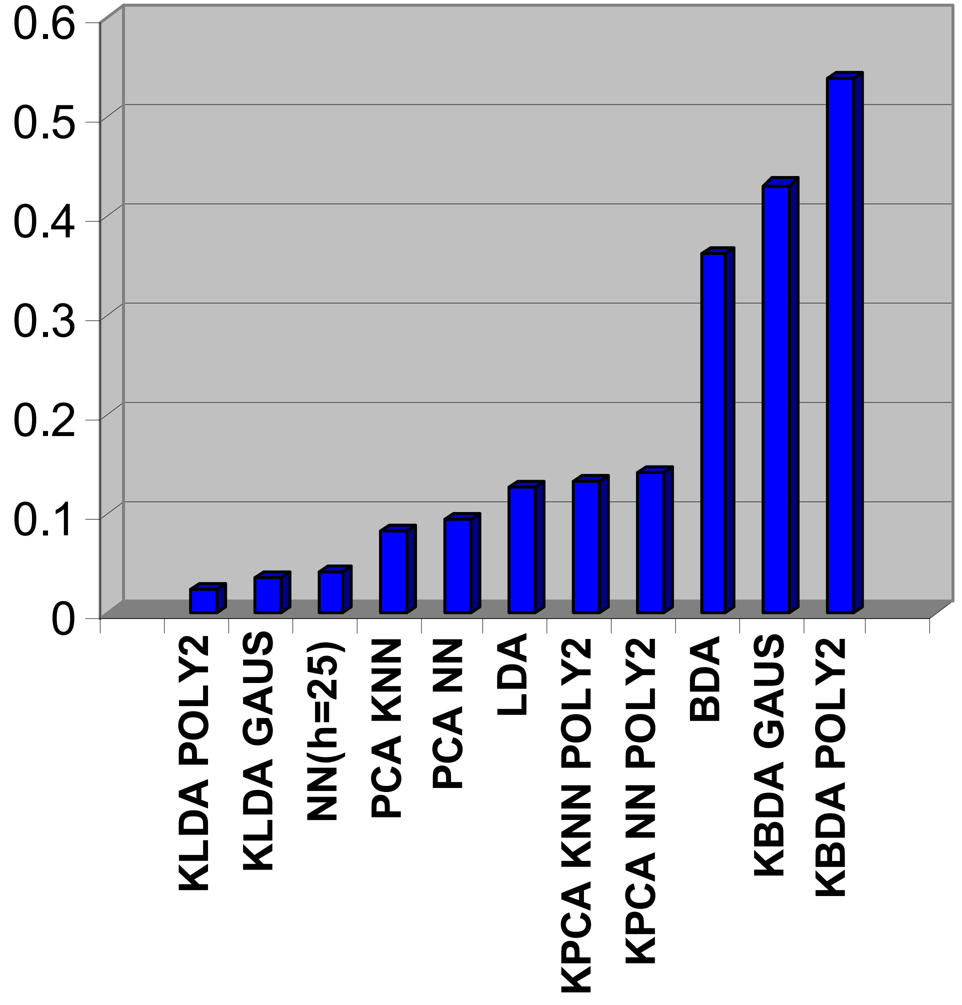
Adaboost

Comparison of all methods

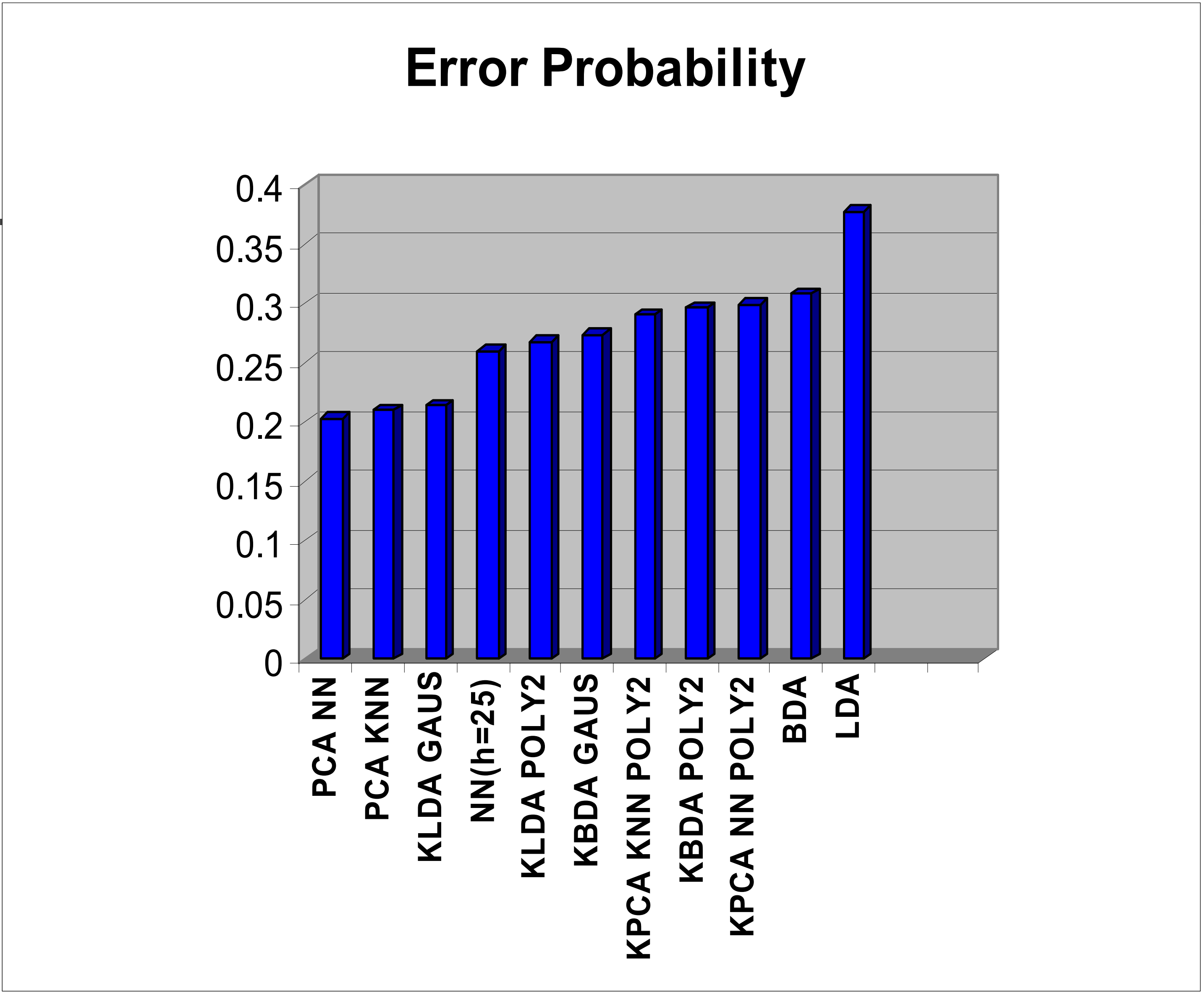
- Used 1000 faces and 1000 nonfaces for training
- 472 faces and 472 nonfaces for testing
- For NN based methods threshold was set to zero


Detection Probability
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
LDA
KLDA POLY2
NN(h=25)
KPCA KNN POLY2
KPCA NN POLY2
KLDA GAUS
PCA KNN
PCA NN
BDA
KBDA GAUS
KBDA POLY2

False Alarm



0.6
0.5
0.4
0.3
0.2
0.1
0
KLDA POLY2
KLDA GAUS
NN(h=25)
PCA KNN
PCA NN
LDA
KPCA KNN POLY2
KPCA NN POLY2
BDA
KBDA GAUS
KBDA POLY2



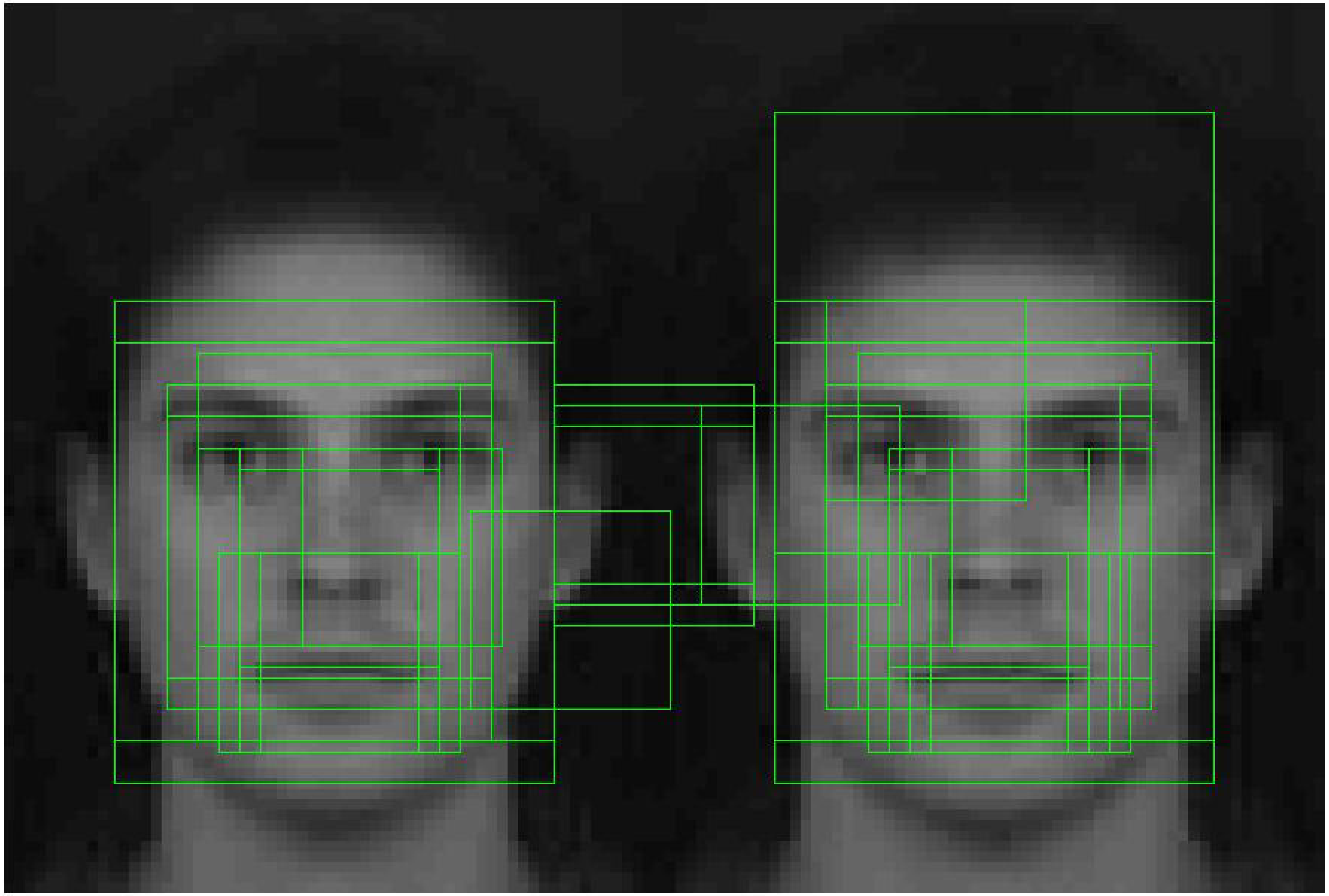
Error Probability
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
PCA NN
PCA KNN
KLDA GAUS
NN(h=25)
KLDA POLY2
KBDA GAUS
KPCA KNN POLY2
KBDA POLY2
KPCA NN POLY2
BDA
LDA

Conclusions

Detecting faces in the entire image

- A 19 x 19 window is slid over the entire image and the windowed image data is sent as a vector to the detector
- To detect faces of different sizes the scanning is repeated for successively smaller scales of the image by downsampling (typically by 1.2 to 1.4)

Output of the detector



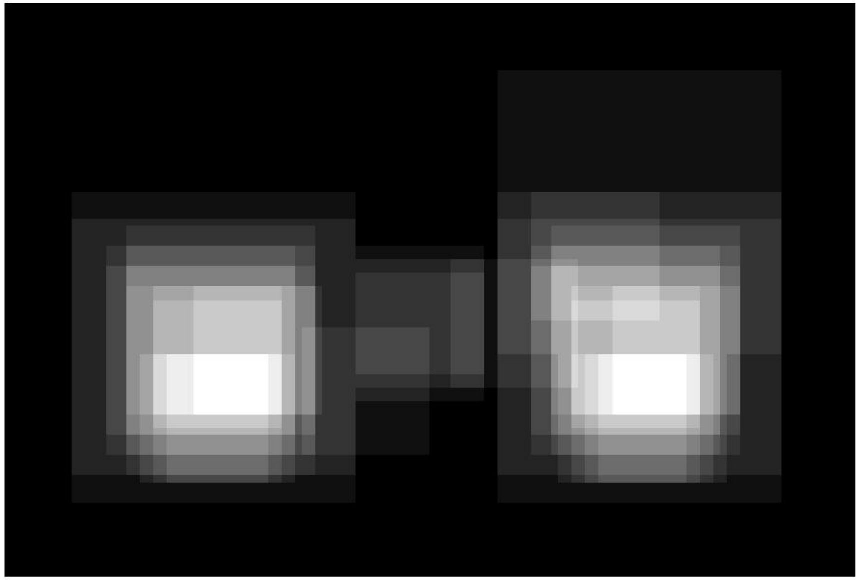
Reducing false alarm

- Multiple detections in areas of the image where there is a face, and false detections only appear in a single position.
- Can be used to significantly reduce false alarm
- Assuming faces do not overlap multiple detections in the image can be assumed to be a measure of high confidence that the detected area to be a face.

Consensus-voting scheme

- For each scale of the image, retain information about the number of times each pixel overlapped due to multiple detections.
- Process the image for all scales possible, we added all the vote matrices to give the final vote matrix
- Threshold the final vote matrix

After Consensus voting



After Thresholding

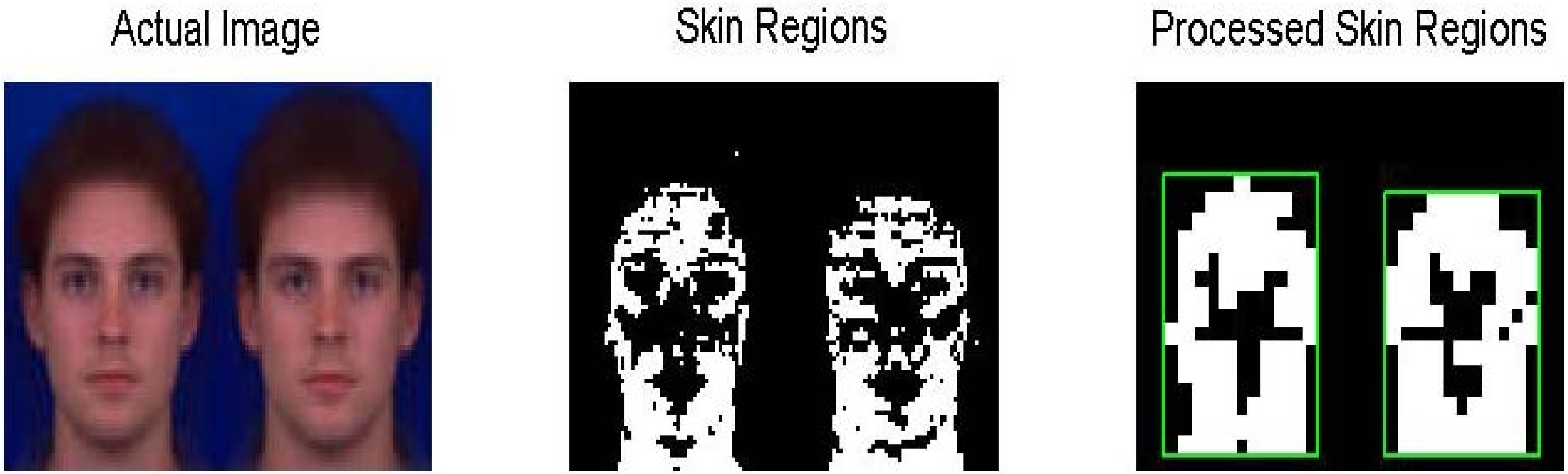
Find Connected components and their bounding box

Sample results

Why not use color info….?

- Detect skin regions.
- In an 8x8 block if the number of skin pixels is less than 32 eliminate the skin region.
- Find all the connected components and label them as face.

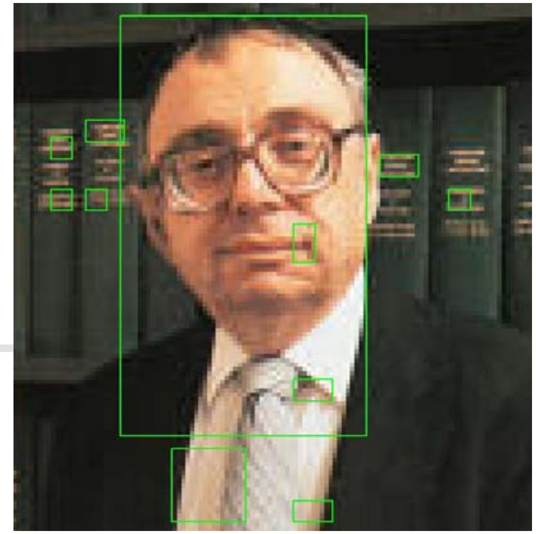
Color based face detection



How to detect skin regions?

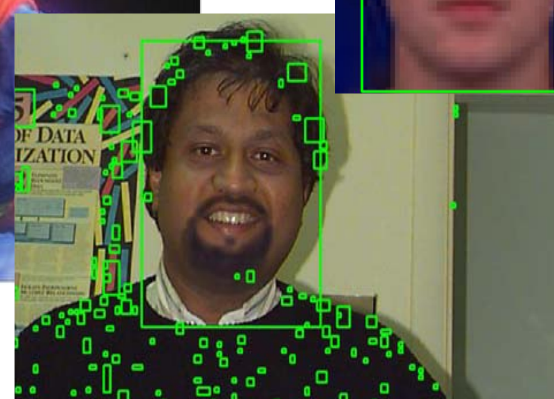
- RGB to Y Cb Cr
- Get the joint distribution of Cb and Cr for skin
- Under normal illuminations skin color occupy small regions of the color space
- $77<Cb<127$ $133<Cr<173$

Sample Results





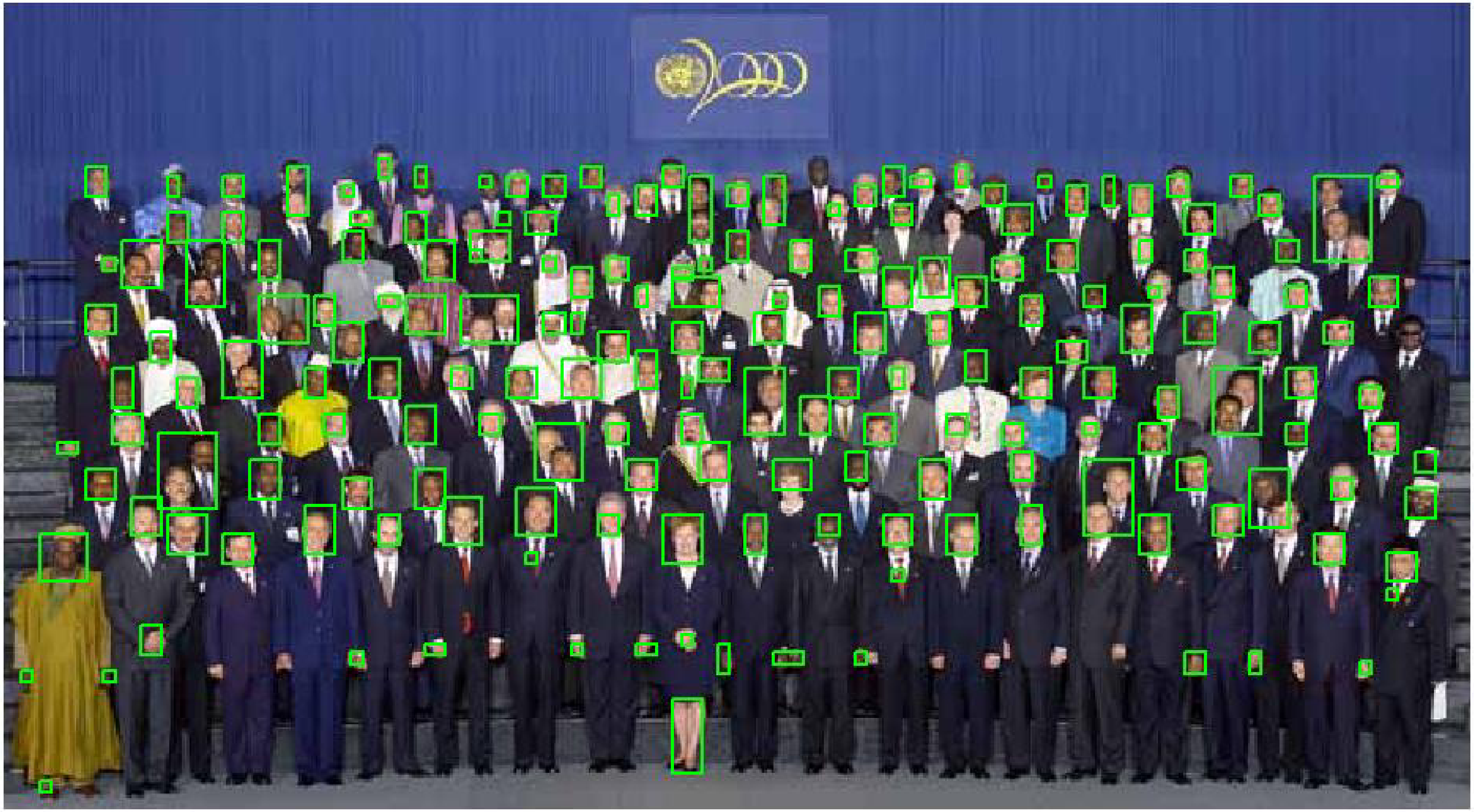





Eliminate false detections

- Use aspect ratio
- Search for face like features
- Use other detectors to validate face or not
- Use different color distribution models for different illumination conditions (indoor/outdoor)
- Ideal for real time applications where we have one camera and the color distribution model for that camera can be found

Sample result




THANK
YOU ALL