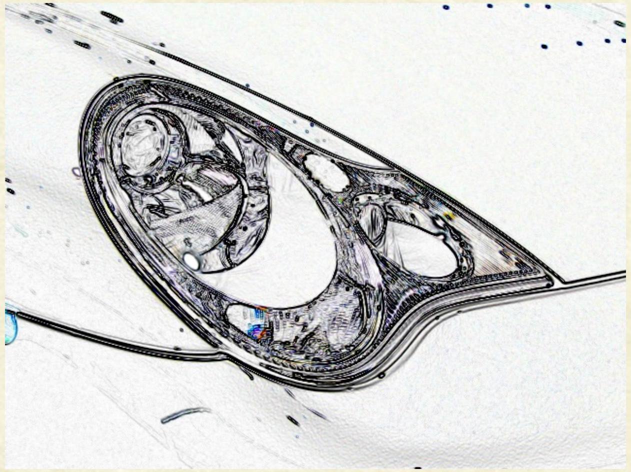
CS7.505: Computer Vision

Spring 2022: Face Detection and Recognition

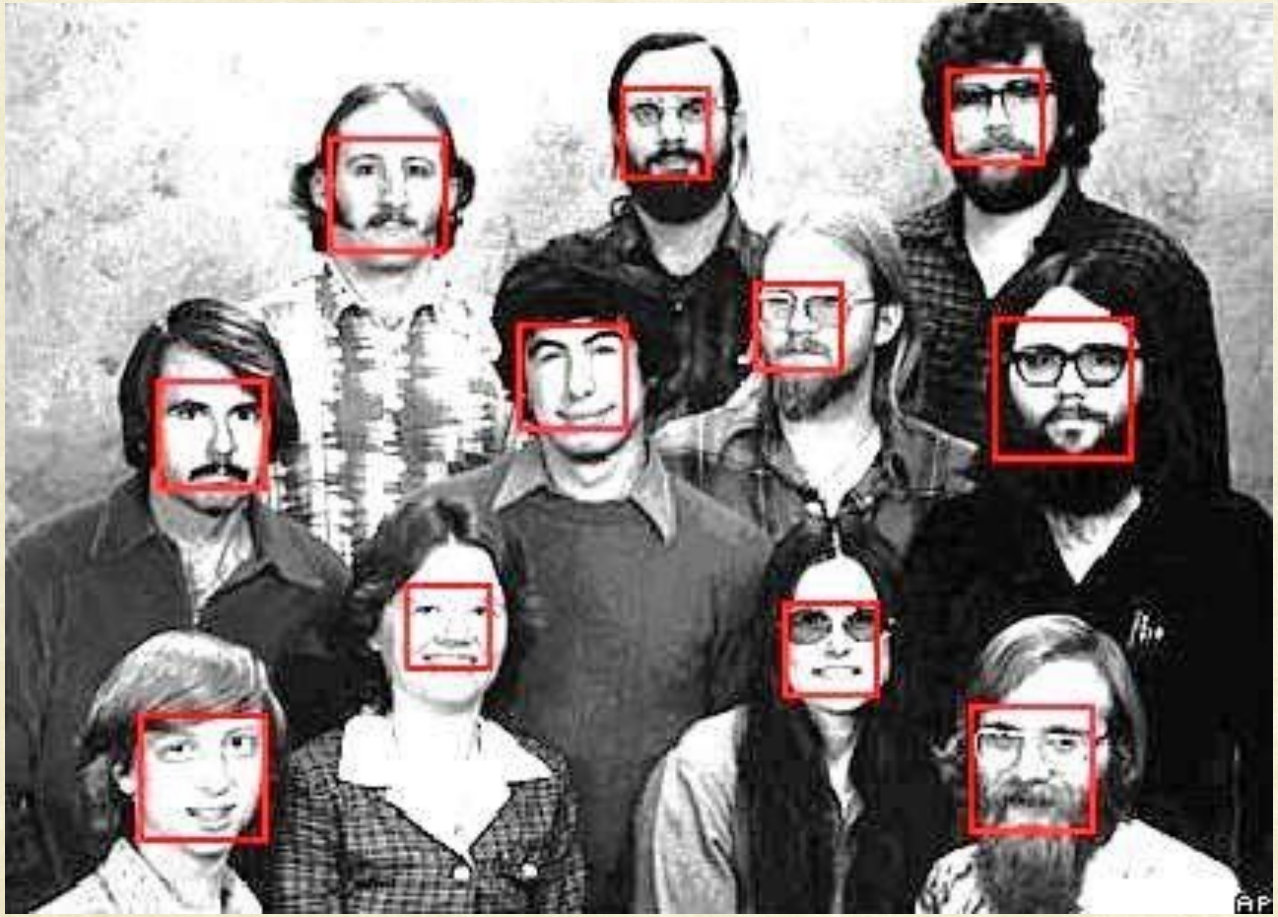


Anoop M. Namboodiri

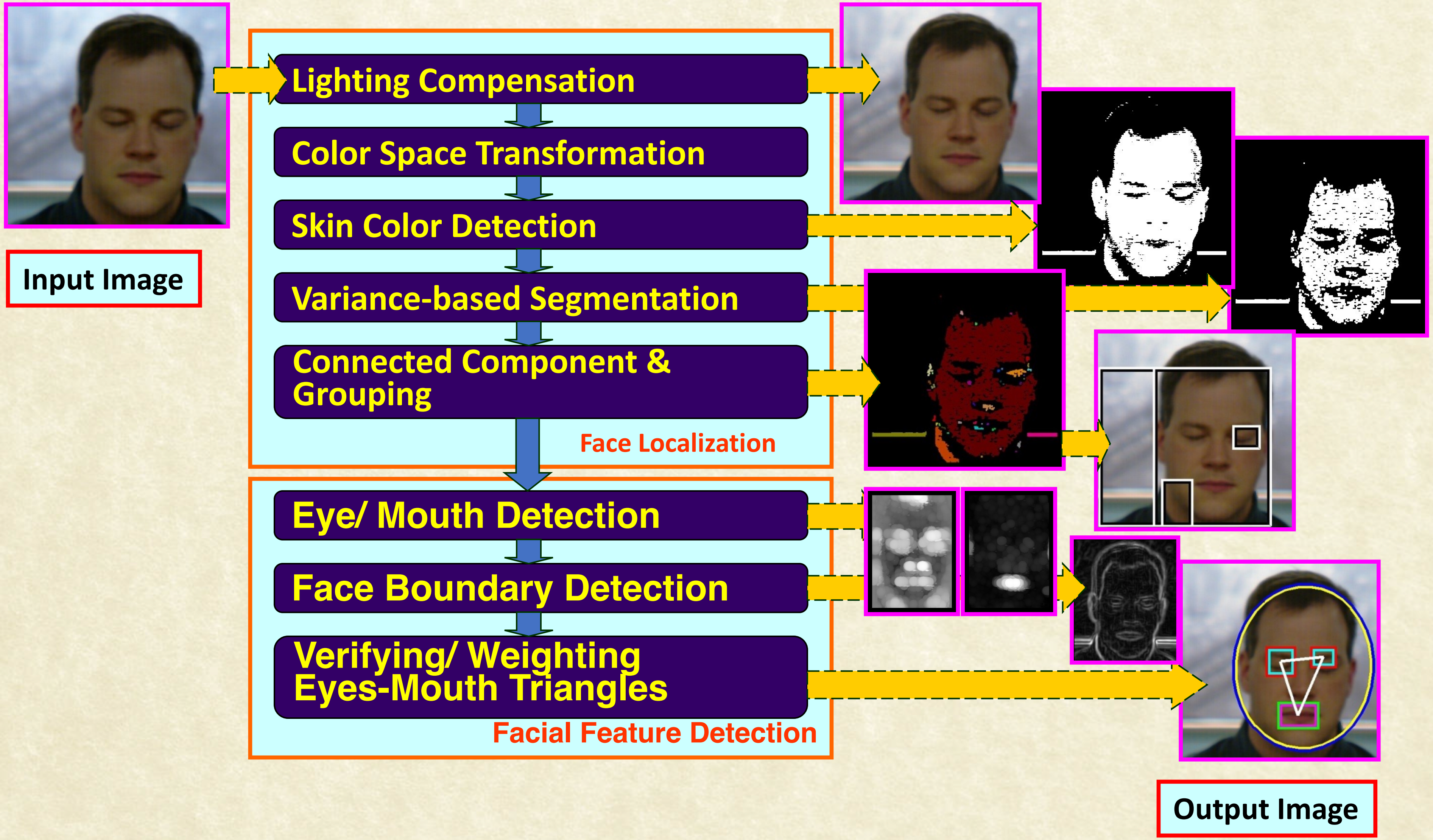
Biometrics and Secure ID Lab, CVIT,

IIIT Hyderabad

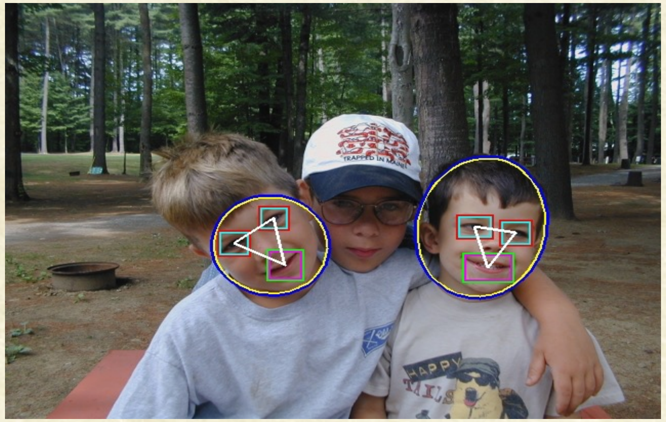
Face Detection



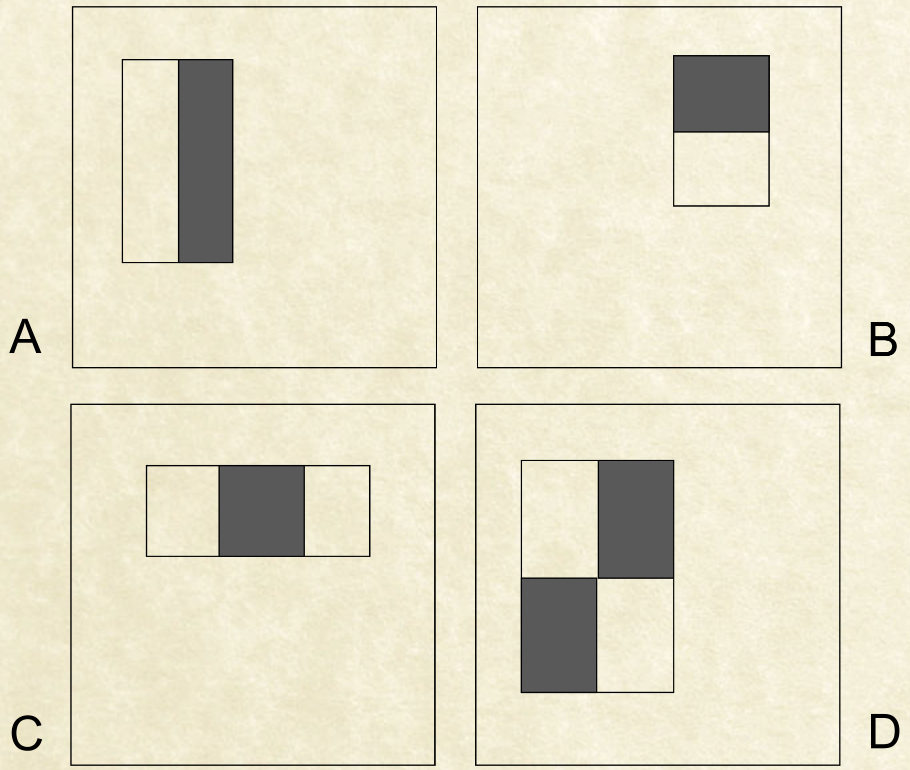
Face Detection Using Skin Color



Face Detection Using Skin Color

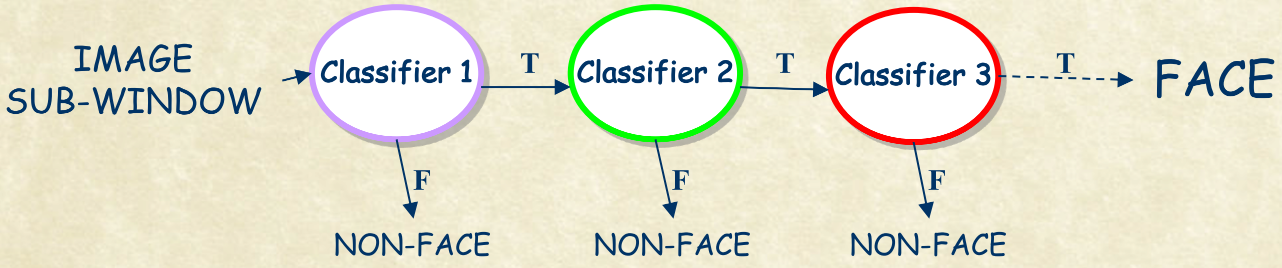


Viola and Jones Detector



Rectangular filters

$$f(x,y) = \sum_i p_b(i) - \sum_i p_w(i)$$



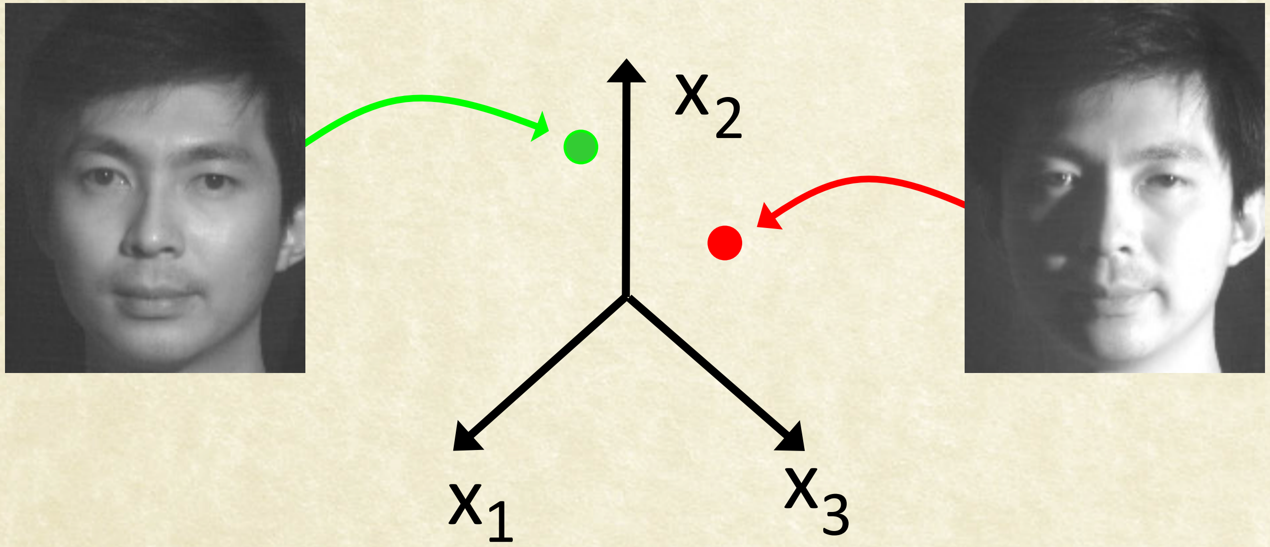
Face Recognition vs. Verification

- Face Authentication/Verification (1:1 matching)

- Face Identification/Recognition (1:N matching)

Image as a Feature Vector

- Consider an n-pixel image to be a point in an n-dimensional space, $\mathbf{x} \in \boldsymbol{R}^n$. Each pixel value is a coordinate of $\mathbf{x}$.
- Use Nearest Neighbor Classifier
- What is the problem?



$\{ R_j \}$ are set of training images.

$$ID = \arg\min_j dist(R_j, I)$$

Intra-Class Variability



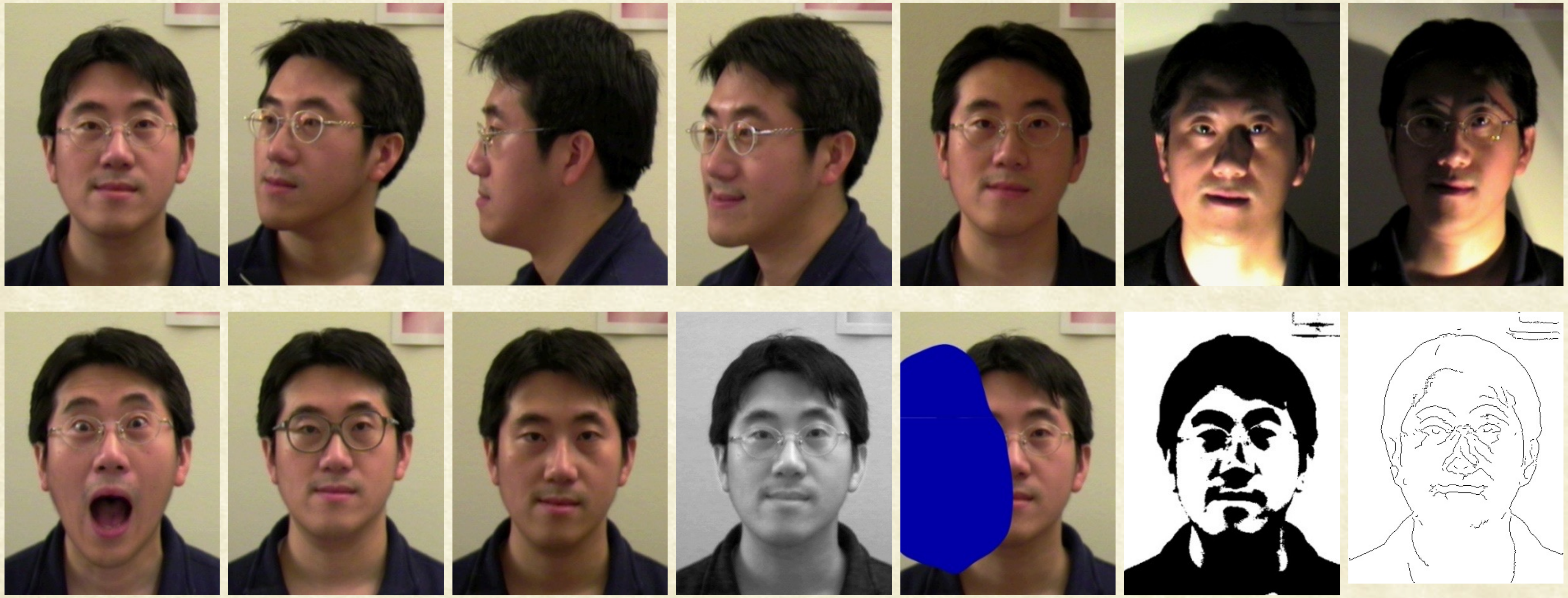







Intra-Class Variability

- Faces with intra-subject variations in pose, illumination, expression, accessories, color, occlusions, and brightness



Inter-Class Similarity

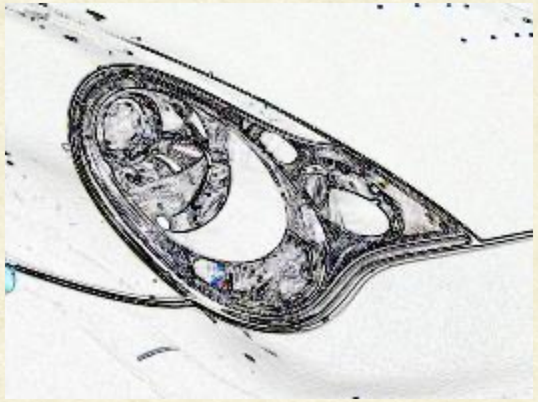
www.marykateandashley.com

news.bbc.co.uk/hi/english/in_depth/americas/2000/us_elections

Twins

Father and son

Eigen Faces for Recognition



Covariance : m samples, n features

Vectors 1 and 3 Cell (3, 1) or (1, 3)

$$A = \begin{bmatrix} 1 & 3 & 5 \\ 5 & 4 & 1 \\ 3 & 8 & 6 \end{bmatrix} \quad \xrightarrow{\text{Covariance}} \quad \begin{bmatrix} 2.67 & 0.67 & -2.67 \\ 0.67 & 4.67 & 2.33 \\ -2.67 & 2.33 & 4.67 \end{bmatrix}$$

Covariance Matrix

Variance:

$$s^2 = \frac{\sum (\overline{X} - X_i)^2}{N}$$

Covariance:

$$cov(\mathbf{X}, \mathbf{Y}) = \frac{1}{n} \sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})$$

$$Cov(M_a, M_b) = \frac{1}{m} \sum_{i=1}^{m} (q_{i,a} - \overline{q_a})(q_{i,b} - \overline{q_b})$$

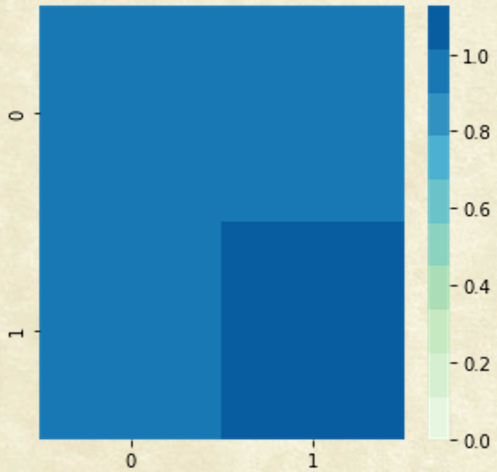
$$\begin{bmatrix} 0.39701 & 0.51117 \\ 0.55582 & 0.93003 \\ 0.59403 & 0.96645 \\ 0.51544 & 0.29759 \\ 0.85313 & 0.18118 \\ 0.88564 & 0.69114 \end{bmatrix}$$

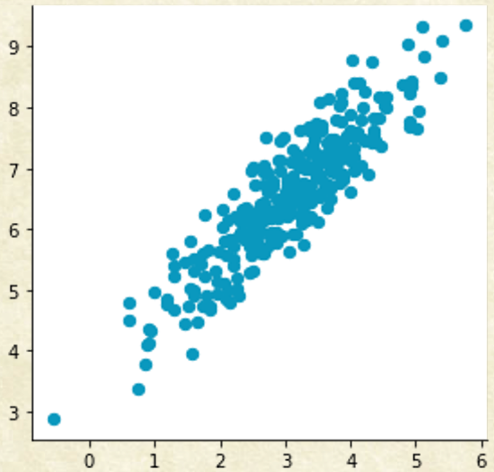
$$\begin{pmatrix} & M1 & M2 & M3 & \dots & Mn \\ S1 & q_{1,1} & q_{1,2} & q_{1,3} & \dots & q_{1,n} \\ S2 & q_{2,1} & q_{2,2} & q_{2,3} & \dots & q_{2,n} \\ S3 & q_{3,1} & q_{3,2} & q_{3,3} & \dots & q_{3,n} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ Sm & q_{m,1} & q_{m,2} & q_{m,3} & & q_{m,n} \end{pmatrix} \Rightarrow C = \begin{pmatrix} \mathrm{cov}(M_1,M_1) & \mathrm{cov}(M_1,M_2) & \cdots & \mathrm{cov}(M_1,M_n) \\ \mathrm{cov}(M_2,M_1) & \mathrm{cov}(M_2,M_2) & \cdots & \mathrm{cov}(M_2,M_n) \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{cov}(M_n,M_1) & \mathrm{cov}(M_n,M_2) & \cdots & \mathrm{cov}(M_n,M_n) \end{pmatrix}_{n \times n}$$

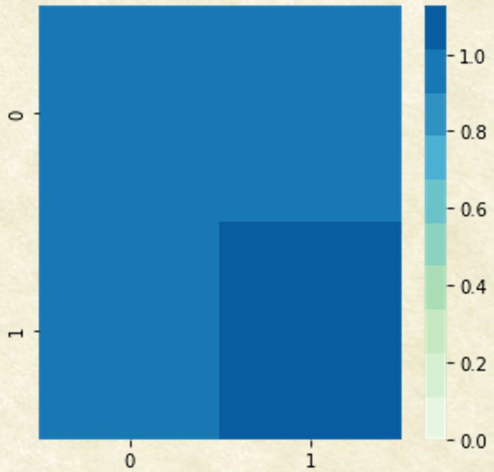
N-dimensional Covariance Matrix

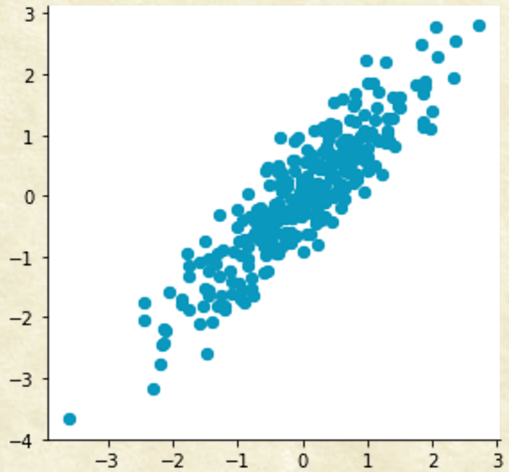
Mean Normalization

$$\boldsymbol{X}' = \boldsymbol{X} - \bar{x}$$







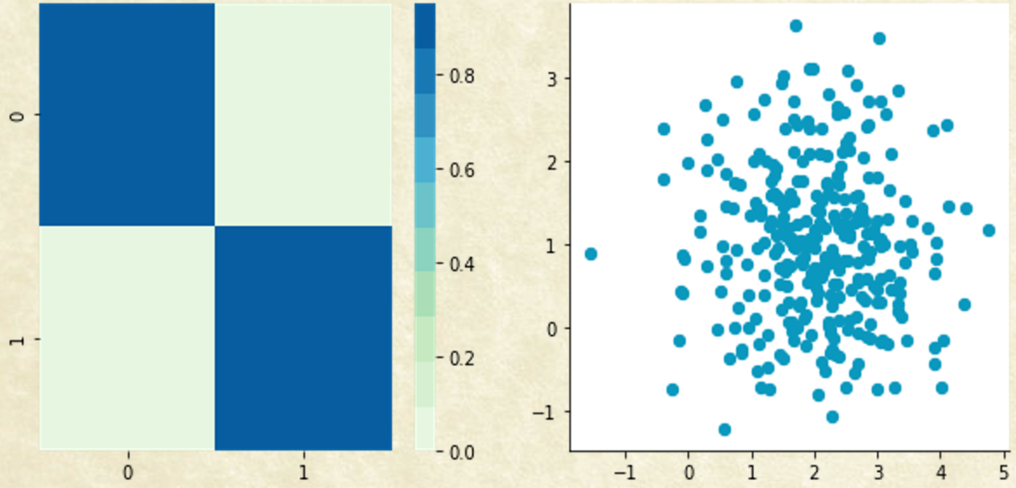


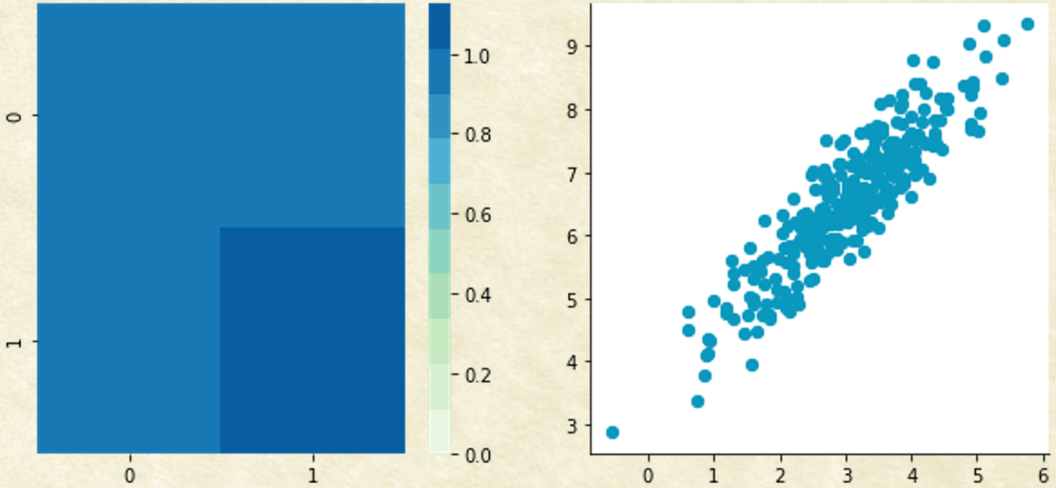
$$C = \begin{bmatrix} +0.95 & +0.92 \\ +0.92 & +1.12 \end{bmatrix}$$

$$C = \begin{bmatrix} +0.95 & +0.92 \\ +0.92 & +1.12 \end{bmatrix}$$

Covariance Matrix

- The off-diagonal elements give insights into the spread of data points in the feature space

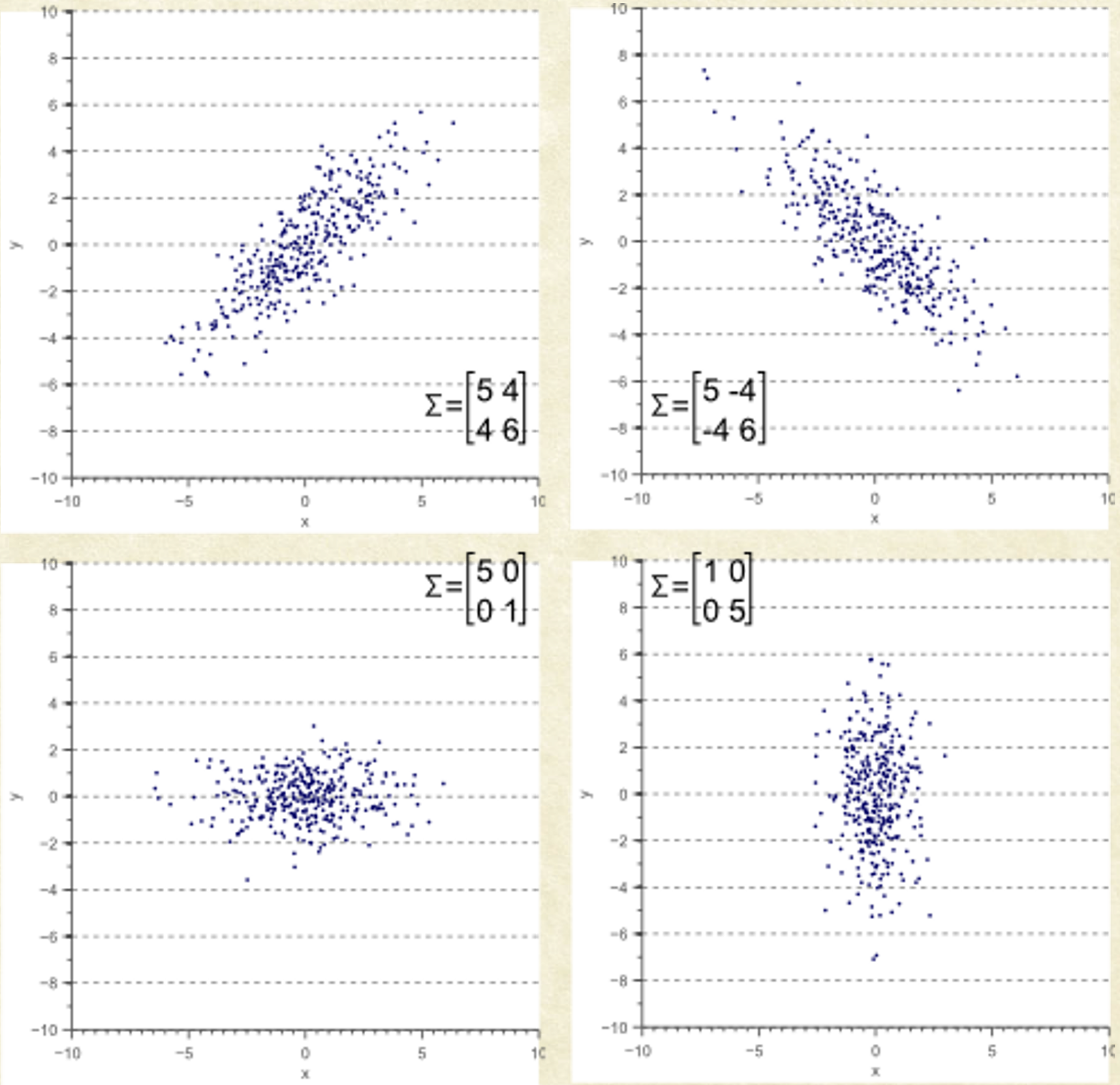




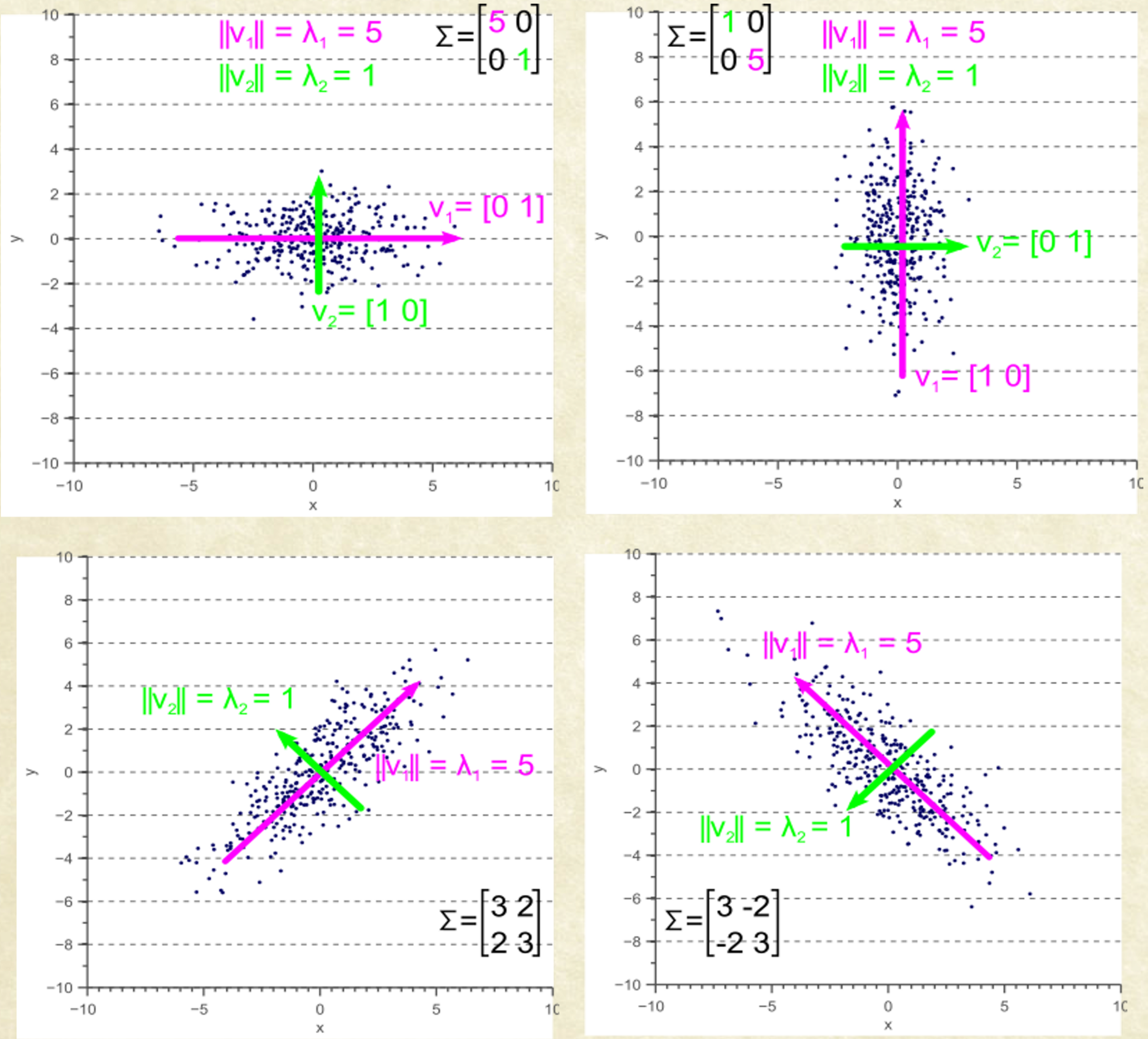
$$C = \begin{bmatrix} +0.95 & -0.04 \\ -0.04 & +0.87 \end{bmatrix}$$

$$C = \begin{bmatrix} +0.95 & +0.92 \\ +0.92 & +1.12 \end{bmatrix}$$

Covariance Matrix and Data Distribution



Eigen Analysis of Covariance Matrix

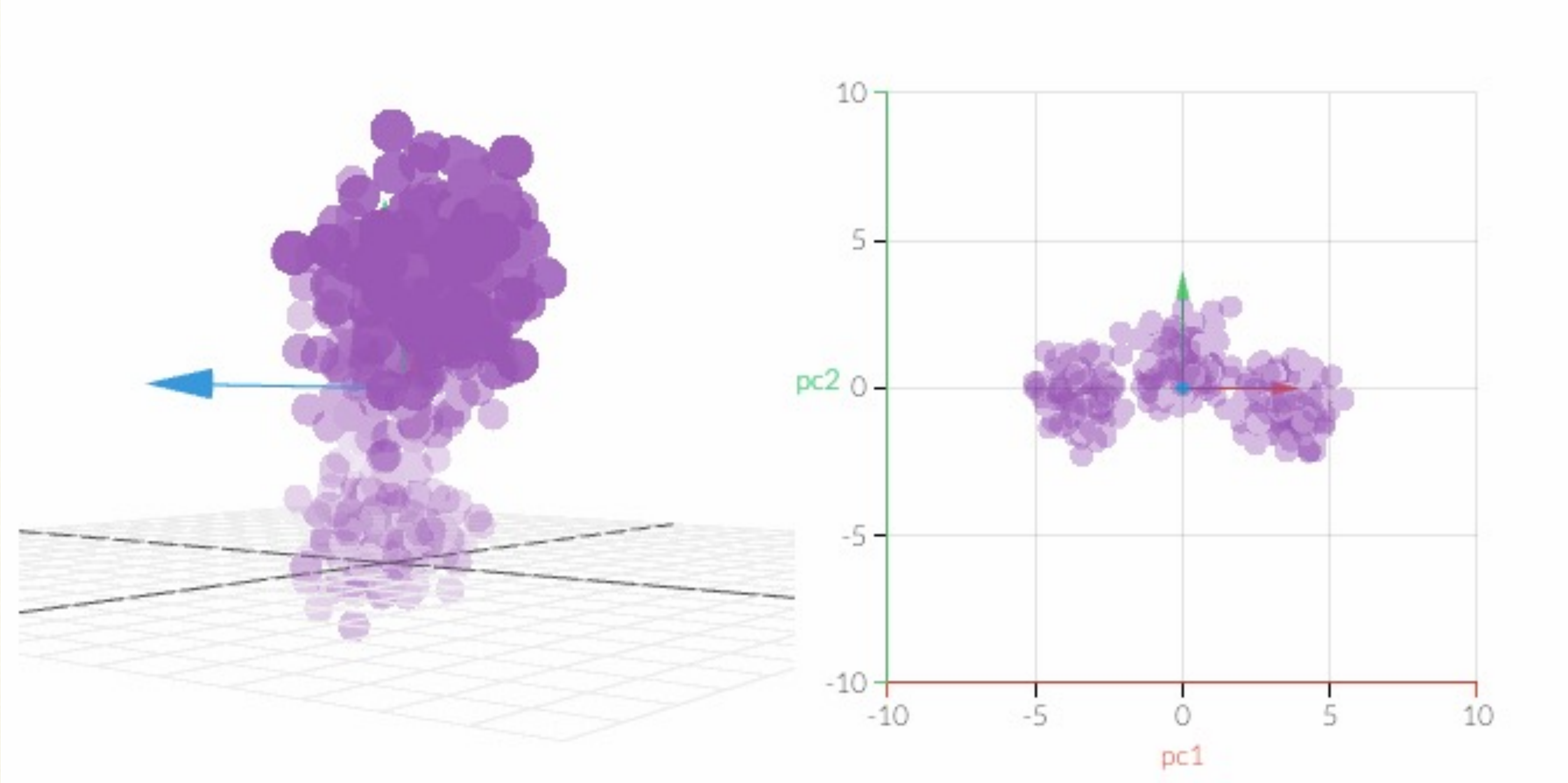


$$\Sigma\vec{v} = \lambda\vec{v}$$

Value of λ indicates `variance' (spread) in direction of eigenvector v associated with λ

v_1, v_2: *Principal Components*

3D to 2D



X1, X2, X3

Z1, Z2

PCA based Feature Extraction

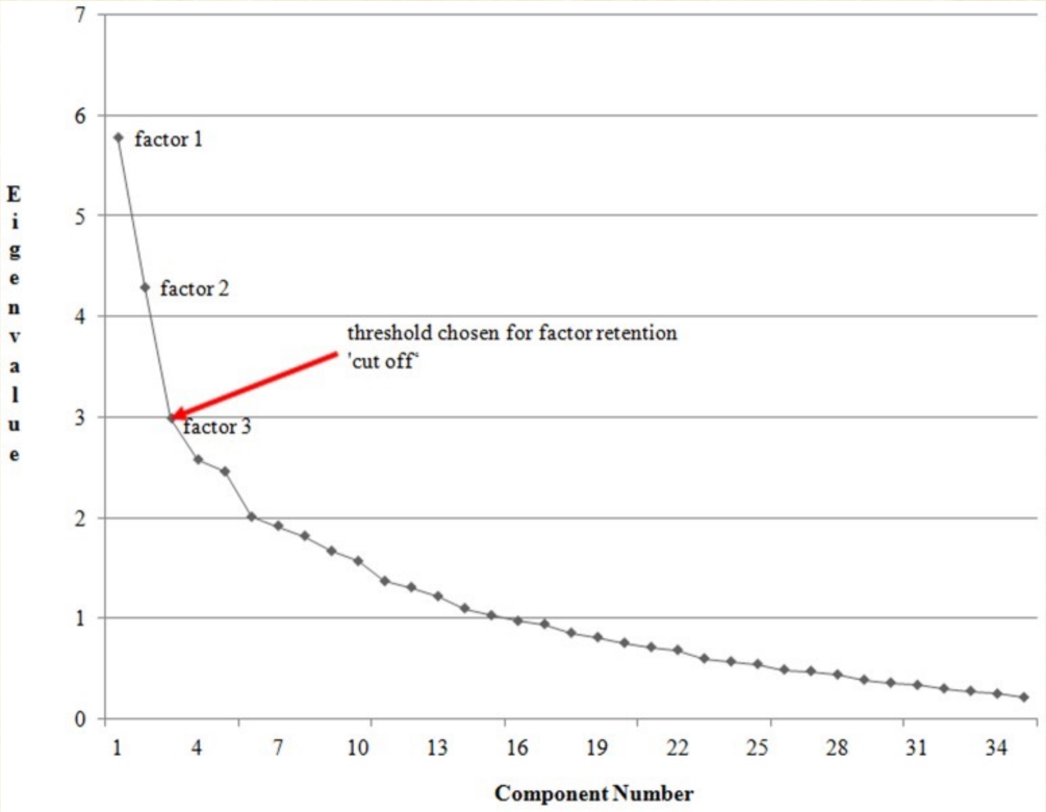
$$\overset{r \times 1}{\underset{Z}{\begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ . \\ . \\ z_r \end{bmatrix}}} = \overset{r \times d}{\underset{U}{\begin{bmatrix} u_{11} & u_{12} & u_{13} & . & . & . & . & . & u_{1d} \\ u_{21} & u_{22} & u_{23} & . & . & . & . & . & u_{2d} \\ u_{31} & u_{32} & u_{33} & . & . & . & . & . & u_{3d} \\ . & & & & & & & & \\ u_{r1} & u_{r2} & u_{r3} & . & . & . & . & . & u_{rd} \end{bmatrix}}} \overset{d \times 1}{\underset{X}{\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ . \\ . \\ . \\ . \\ . \\ . \\ . \\ x_d \end{bmatrix}}}$$

Each row in U is an eigen vector of co-variance Matrix

Appreciating PCA: Two Questions

$$\text{Eg.} \frac{\sum_{i=1}^{r} \lambda_i}{\sum_{i=1}^{d} \lambda_i} > 0.90$$

- How many Eigen vectors to select?
  - Ans: Eigen Vectors corresponding to the r largest Eigen values
- How much information is lost? Can we recover the old data/ information from the Low-D point?



$$\mathbf{x} = z_1\mathbf{u_1} + z_2\mathbf{u_2} + z_3\mathbf{u_3} + z_4\mathbf{u_4}$$

$$\mathbf{x} = z_1\mathbf{u_1} + z_2\mathbf{u_2} + z_3\mathbf{u_3} + z_4\mathbf{u_4}$$

$$\mathbf{x}' = z_1\mathbf{u_1} + z_2\mathbf{u_2}$$

$$\text{Loss in Information} = ||\mathbf{x} - \mathbf{x}'||$$

Note: z_3 and z_4 are small and also λ_3 and λ_4 are small

https://www.researchgate.net/profile/Collins_Ouma/publication/263548435/figure/fig1/AS:202705835433990@1425340245309/A-Scree-plot-of-eigenvalues-of-the-unrotated-factors-displaying-an-elbow-of-the-plot.png

Application to Faces

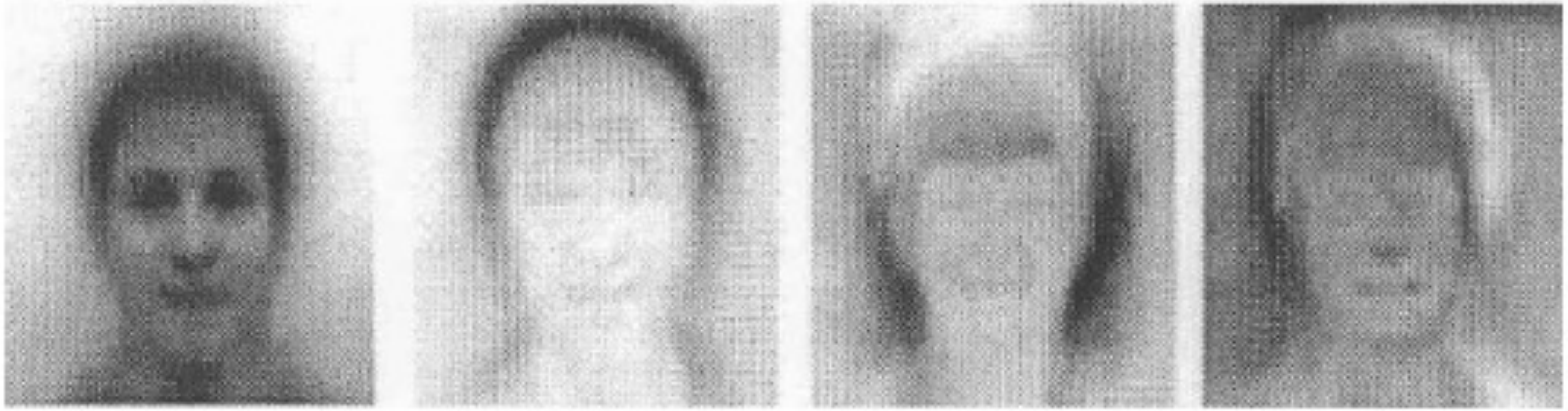
- Representing faces onto this basis

- Face reconstruction:

- Each face (minus the mean) Φ_i in the training set can be represented as a linear combination of the best K eigenvectors:

$$\hat{\Phi}_i - mean = \sum_{j=1}^{K} w_j u_j, \quad (w_j = u_j^T \Phi_i)$$

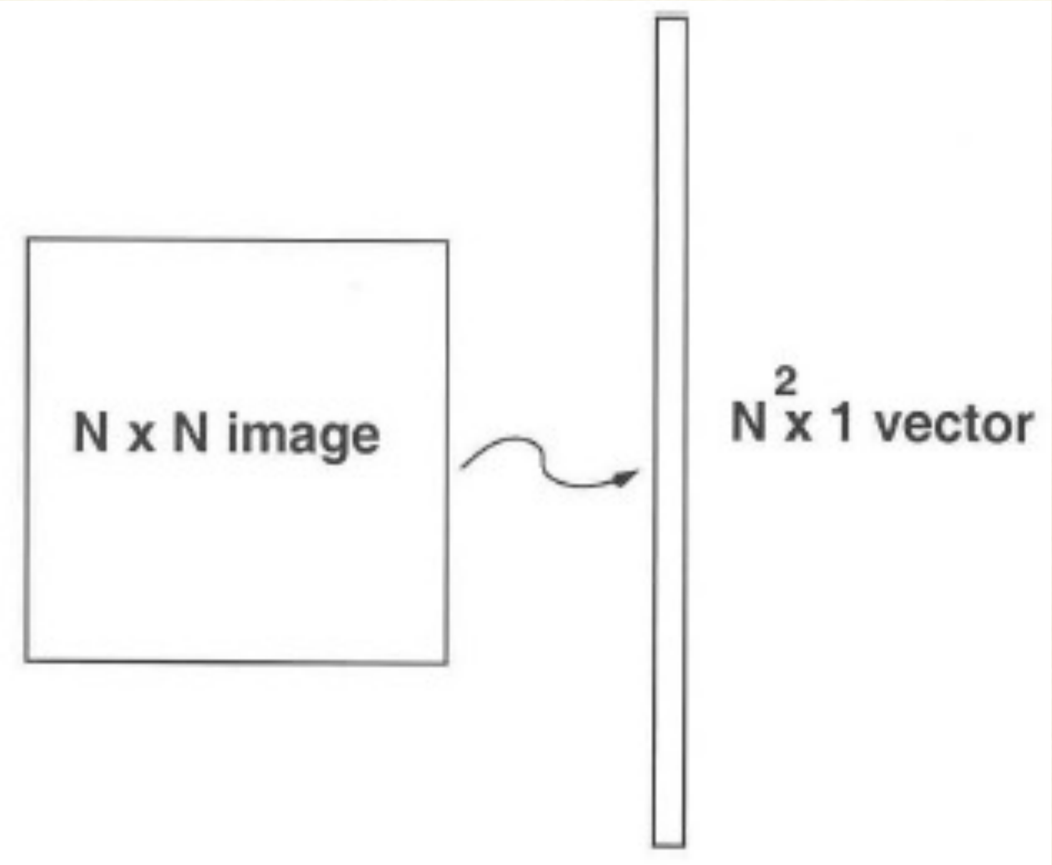
(we call the u_j's *eigenfaces*)



Eigenfaces

– M. Turk, A. Pentland, "Eigenfaces for Recognition", *Journal of Cognitive Neuroscience*, vol. 3, no. 1, pp. 71-86, 1991.

- Face Recognition
  - The simplest approach is to think of it as a template matching problem
  - Problems arise when performing recognition in a high-dimensional space.
  - Significant improvements can be achieved by first mapping the data into a *lower dimensionality* space.



Face Recognition using Eigenfaces

- Given an unknown face image Γ (centered and of the same size like the training faces) follow these steps:

Step 1: normalize Γ: $\Phi = \Gamma - \Psi$

Step 2: project on the eigenspace

$$\hat{\Phi} = \sum_{i=1}^{K} w_i u_i \quad (w_i = u_i^T \Phi) \qquad (where \| u_i \| = 1)$$

Step 3: represent Φ as: $\Omega = \begin{bmatrix} w_1 \\ w_2 \\ \dots \\ w_K \end{bmatrix}$

Step 4: find $e_r = \min_l \|\Omega - \Omega^l\|$ $\qquad \| \Omega - \Omega^l \| = \sum_{i=1}^{K} (w_i - w_i^l)^2$

Step 5: if $e_r < T_r$, then Γ is recognized as face l from the training set.

Challenge 2: Computing Eigen Vectors

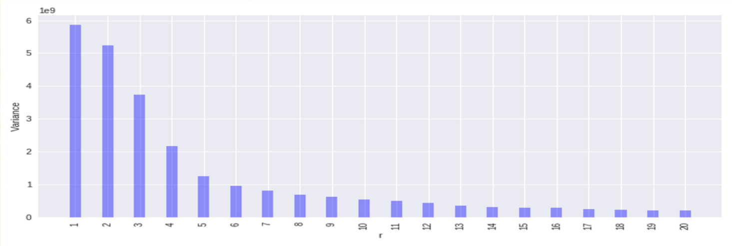
- For an NxN image (say 256 x 256), each feature vector is 65536 long
- Covariance matrix:
  - 65536 x 65536
- Training set: 16 classes, few hundred samples.

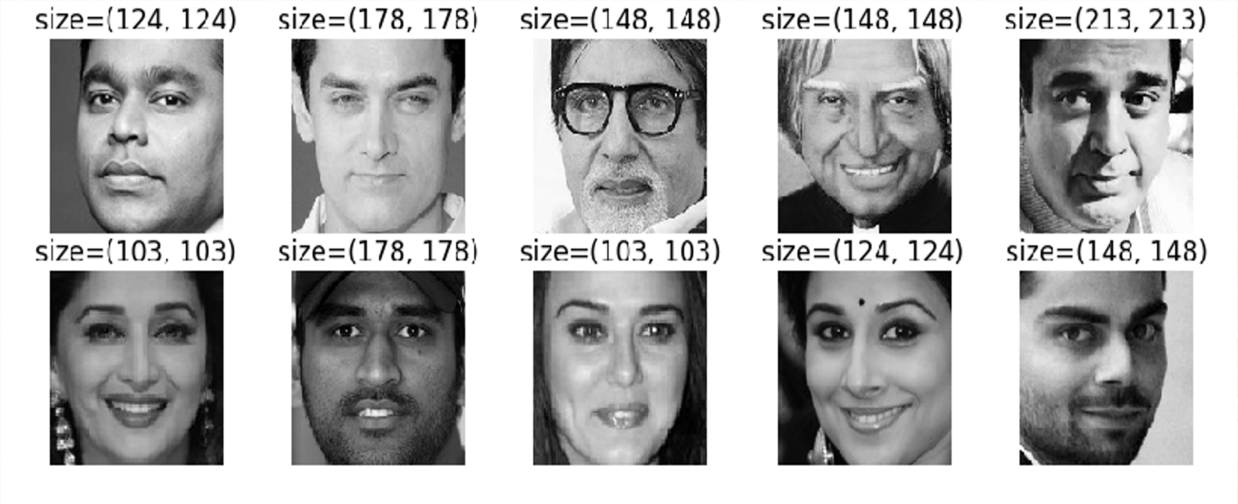
Eigenfaces

Mean Image

Basis Images

Recognize Indian Celebrities





- Rescale and Find mean of the face
- Find the Eigen Faces
- Find the weights to represent the face in Eigen space
- Reconstruct the image

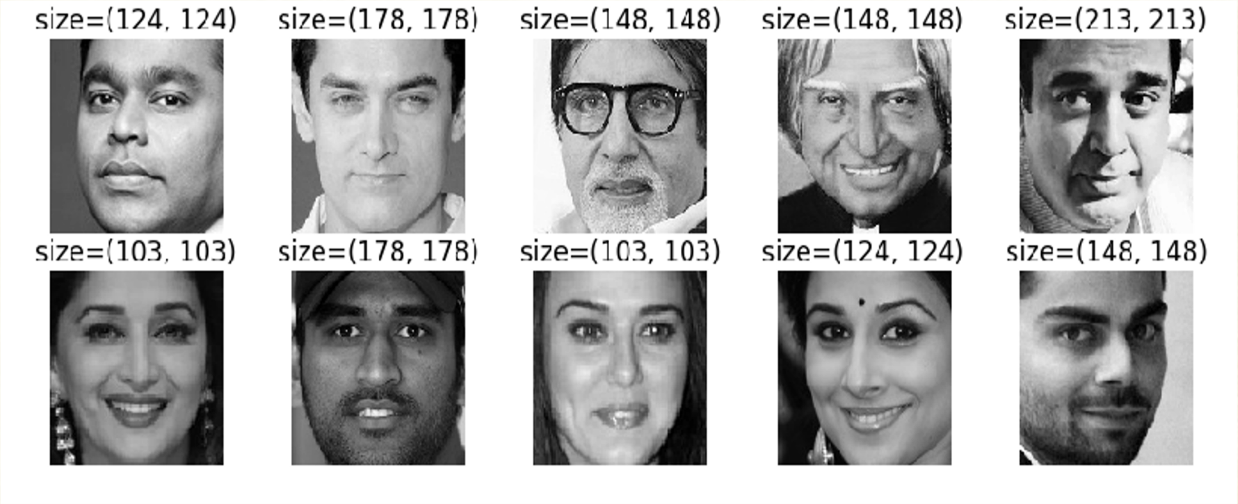
Eigen Vector 1

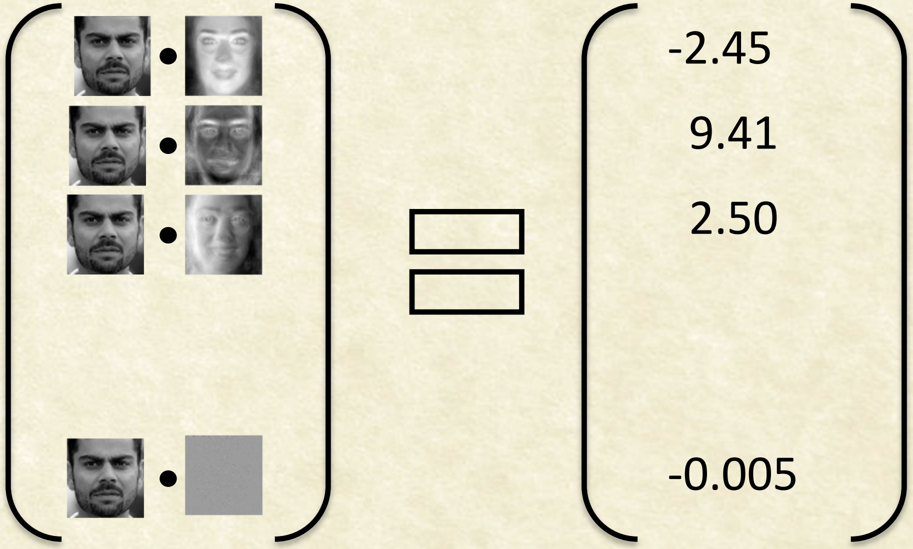
Eigen Vector 100

Eigen Vector 150

Eigen Vector 200

Recognize Indian Celebrities





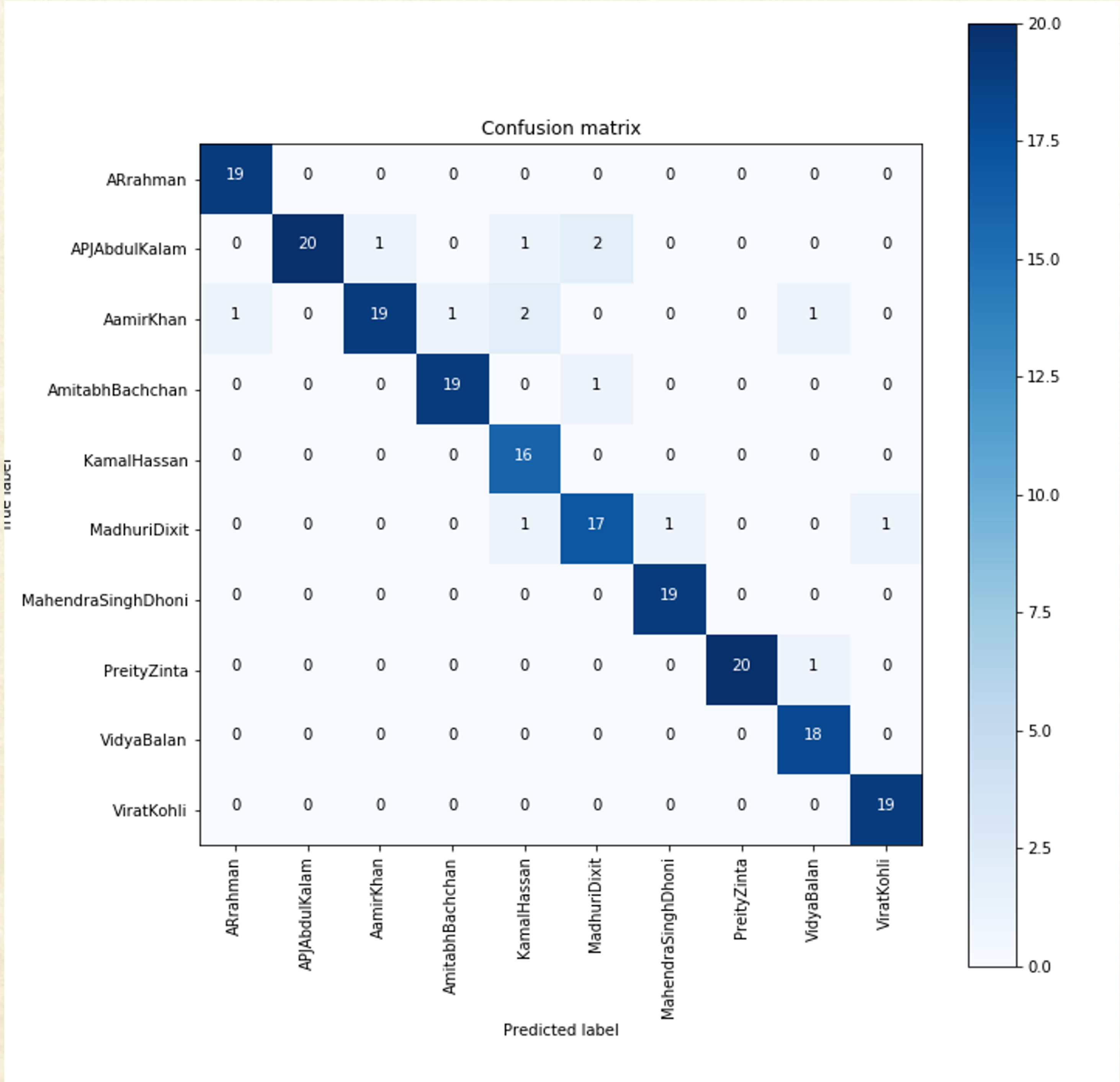
 =  + 

$$\hat{x} = \mu + w_1u_1+w_2u_2+w_3u_3+w_4u_4+ \ldots$$

Classification

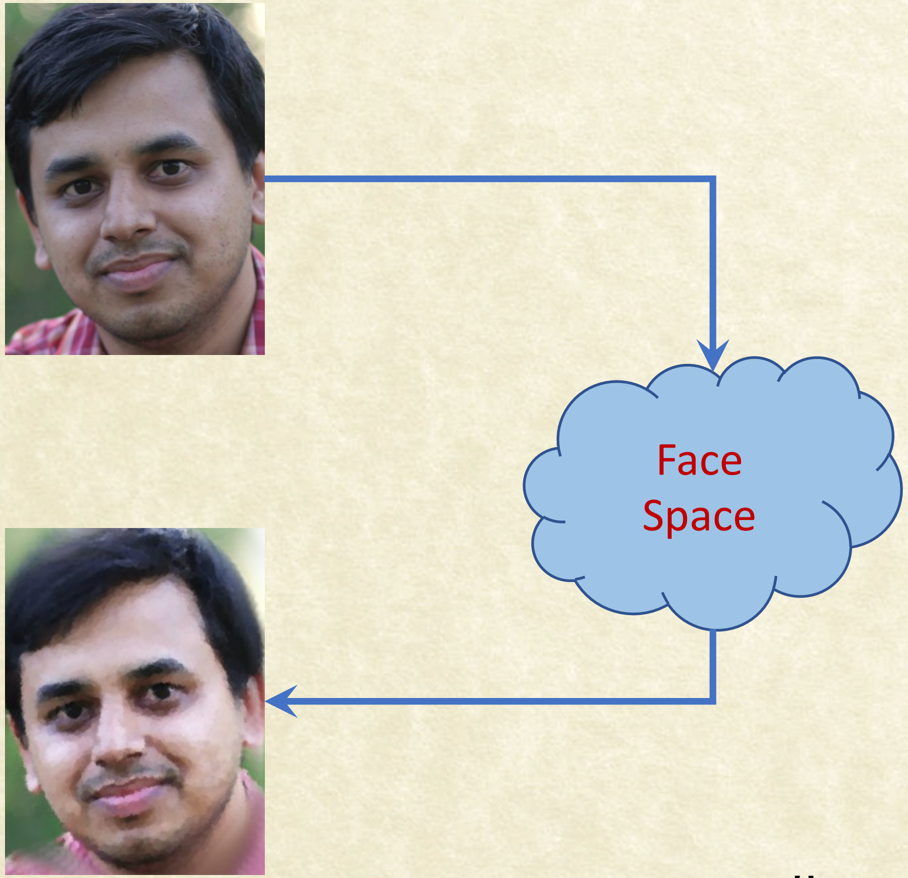
Training images: 400
Test images: 200

Accuracy: 96%



Face Detection using Eigenfaces

Φ



$\hat{\Phi}$

- Given an unknown image Γ

<u>Step 1:</u> compute $\Phi = \Gamma - \Psi$

<u>Step 2:</u> compute $\hat{\Phi} = \sum_{i=1}^{K} w_i u_i \quad (w_i = u_i^T \Phi)$

$(where \| u_i \| = 1)$

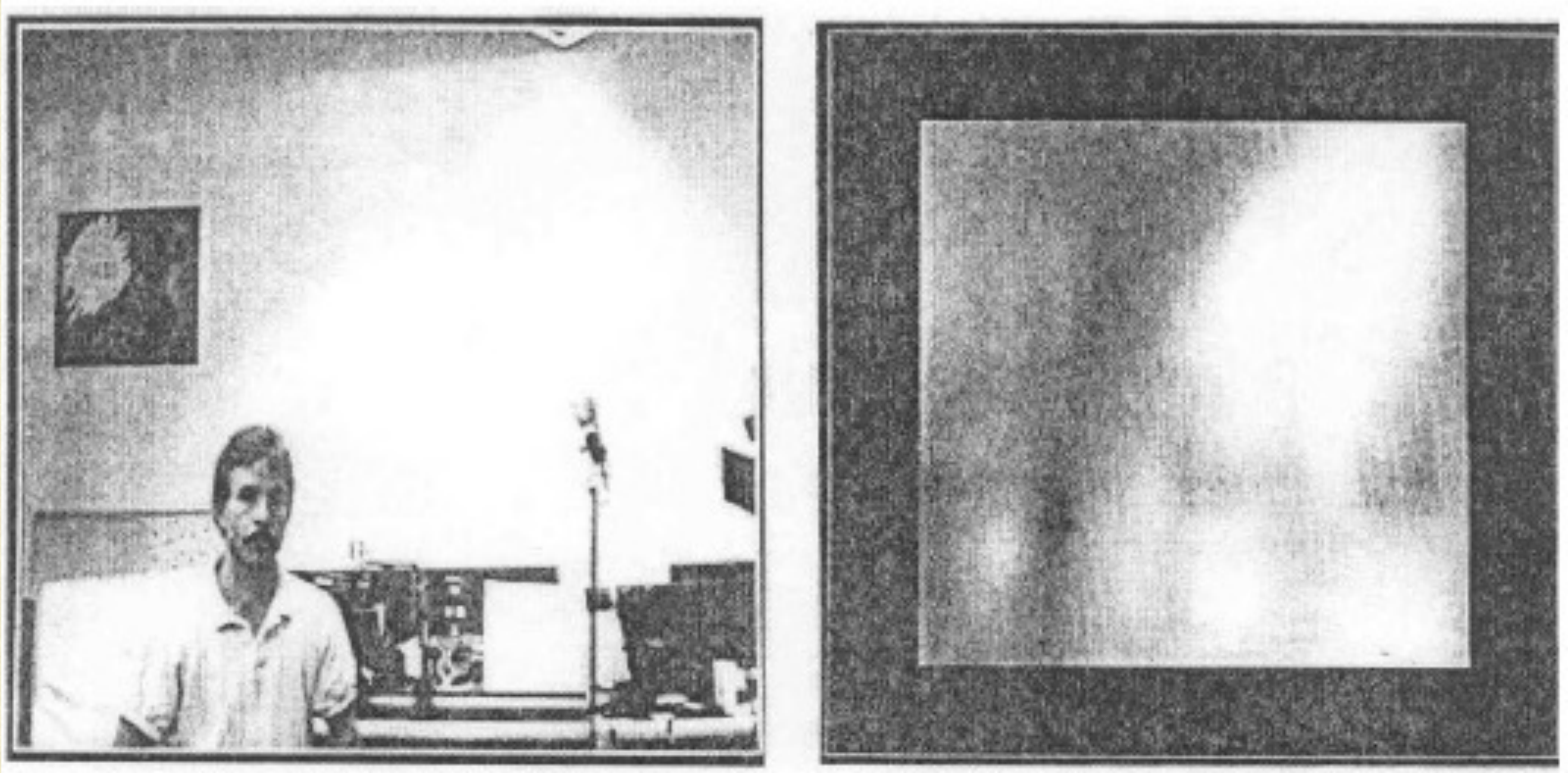
<u>Step 3:</u> compute $e_d = \|\Phi - \hat{\Phi}\|$

<u>Step 4:</u> if $e_d < T_d$, then Γ is a face.

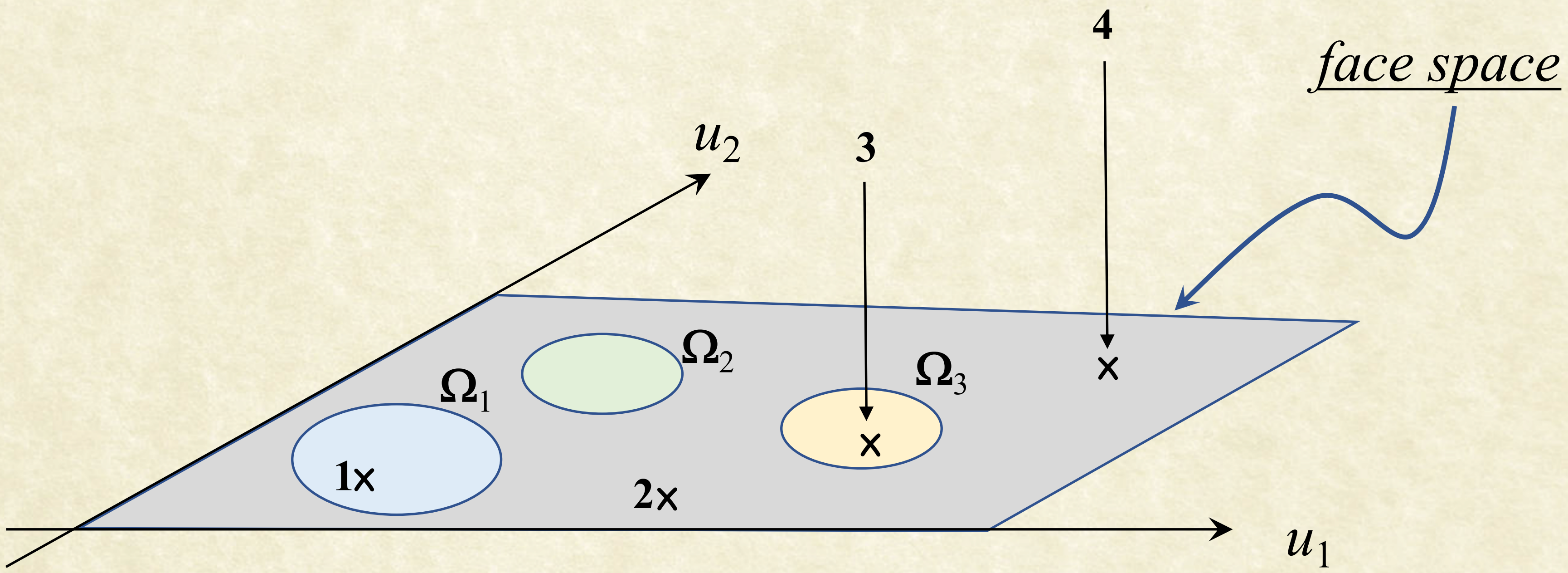
$e_d = \|\Phi - \hat{\Phi}\|$ is called distance from face space (dffs)

Distance from Face Space

Visualize dffs: $e_d = \|\Phi - \hat{\Phi}\|$



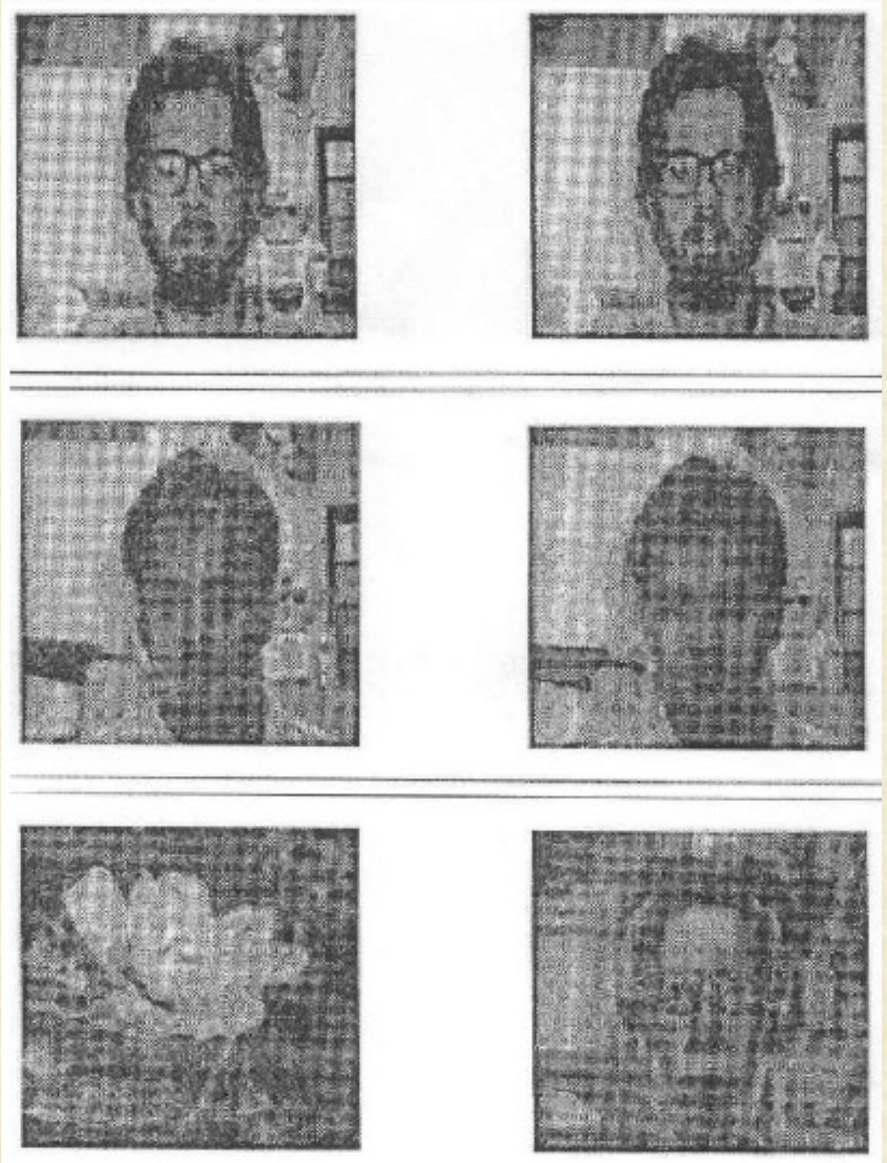
Face Detection using Eigenfaces



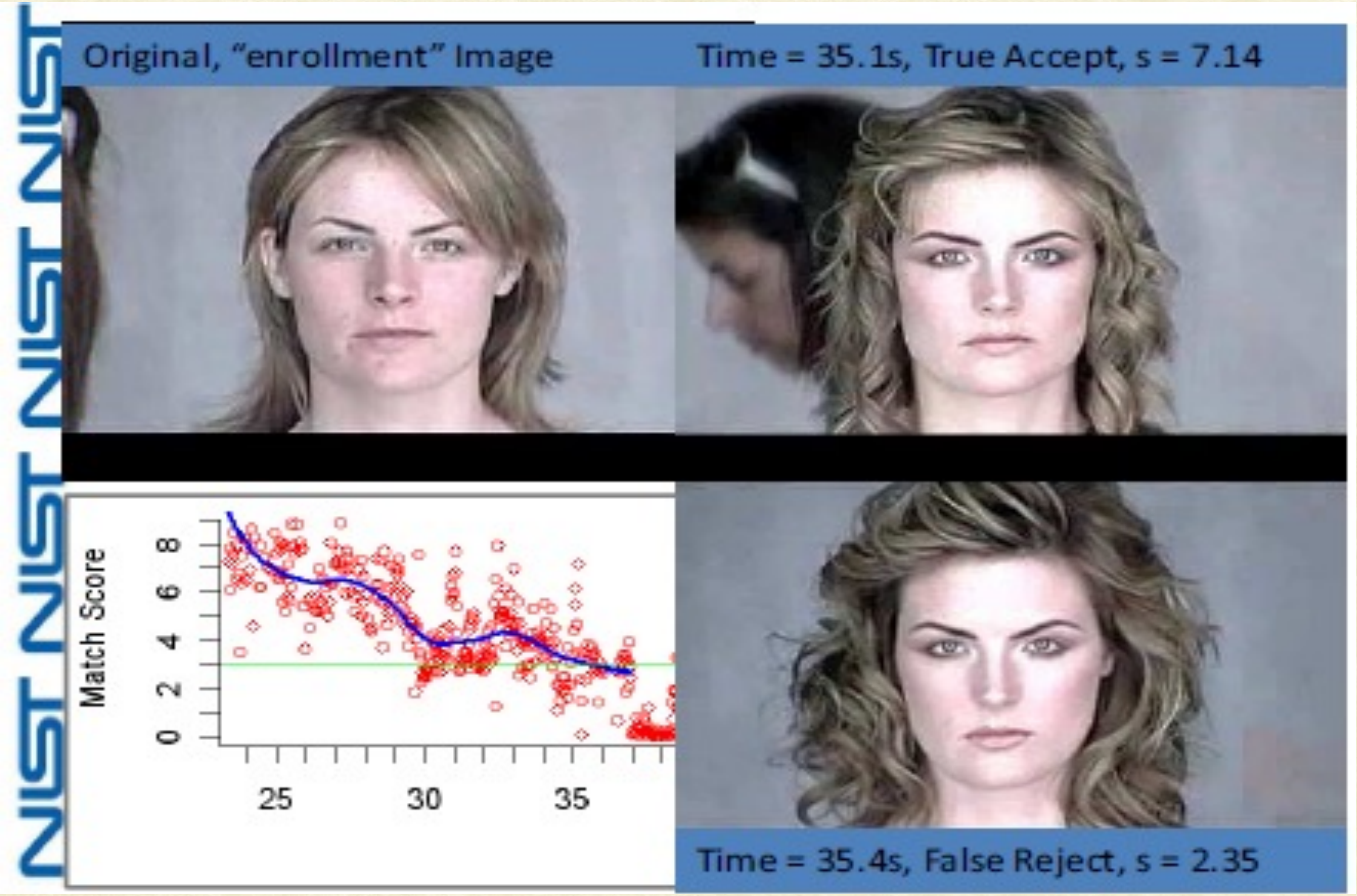
- Case 1: in face space AND close to a given face
- Case 2: in face space but NOT close to any given face
- Case 3: not in face space AND close to a given face
- Case 4: not in face space and NOT close to any given face

Reconstruction from Face Space

- Reconstructed face looks like a face.
- Reconstructed non-face also looks like a face !!
- Robust to partial face occlusion !!



Face Recognition Test: George Quinn (NIST)



Thank You