Face Recognition
CS-477 Computer Vision

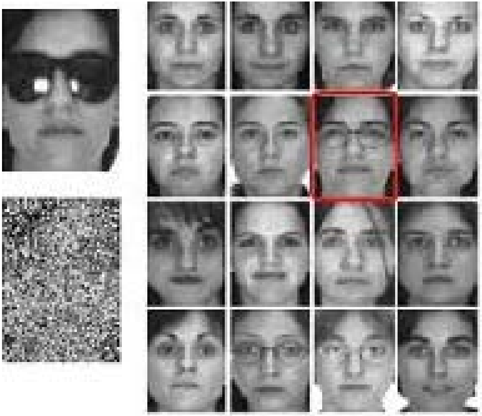
Dr. Mohsin Kamal
Associate Professor
dr.mohsinkamal@seecs.edu.pk

School of Electrical Engineering and Computer Science (SEECS)
National University of Sciences and Technology (NUST), Pakistan

1 Simple Approach

2 Face Recognition

3 Face recognition using Eigenfaces



1 Simple Approach

2 Face Recognition

3 Face recognition using Eigenfaces

How it can be done?

- Recognize faces (mug shots) using gray levels (appearance).
- Each image is mapped to a long vector of gray levels.
- Several views of each person are collected in the database during training.
- During recognition a vector corresponding to an unknown face is compared with all vectors in the database.
- The face from database, which is closest to the unknown face is declared as a recognized face.

Problems and solution

- Problems:
  - Dimensionality of each face vector will be very large (250,000 for a 512×512 image!)
  - Raw gray levels are sensitive to noise, and lighting conditions.
- Solution:
  - Reduce dimensionality of face space by finding principal components (eigen vectors) to span the face space
  - Only a few most significant eigen vectors can be used to represent a face, thus reducing the dimensionality

Eigen Vectors and Eigen Values

- An eigenvector corresponds to the real non zero eigenvalues which point in the direction stretched by the transformation whereas eigenvalue is considered as a factor by which it is stretched. In case, if the eigenvalue is negative, the direction of the transformation is negative.
- To find eigen values of a matrix A first find the roots of:

$$\det(A - \lambda I) = 0$$

- Then solve the following linear system for each eigen value to find corresponding eigen vector

$$(A - \lambda I)x = 0$$

Example

$$A = \begin{bmatrix} -1 & 2 & 0 \\ 0 & 3 & 4 \\ 0 & 0 & 7 \end{bmatrix}$$

Eigen Values: $\lambda_1 = 7, \lambda_2 = 3, \lambda_3 = -1$

Eigen vectors: $x_1 = \begin{bmatrix} 1 \\ 4 \\ 4 \end{bmatrix}, x_2 = \begin{bmatrix} 1 \\ 2 \\ 0 \end{bmatrix}, x_3 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$

1 Simple Approach

2 Face Recognition

3 Face recognition using Eigenfaces

- Collect all gray levels in a long vector u:
  $u = (I(1,1), \cdots, I(1,N), I(2,1), \cdots, I(2,N), \cdots, I(M,1), \cdots, I(M,N))^T$
- Collect n samples (views) of each of p persons in matrix A, which is of dimensions $MN \times pn$.
  $$A = \left[u_1^1, \cdots u_n^1, u_1^2, \cdots, u_n^2, \cdots u_1^p, \cdots, u_n^p\right]$$
- Form a correlation matrix L having dimensions $MN \times MN$ as
  $$L = AA^T$$
- Compute eigen vectors $(\phi_1, \phi_2, \phi_3, \phi_{n_1})$, of L, which form a bases for whole face space.

- Each face, u, can now be represented as a linear combination of eigen vectors

$$u = \sum_{i=1}^{n_1} a_i \phi_i$$

- Eigen vectors for a symmetric matrix are orthonormal:

$$\phi_i^T.\phi_j = \begin{cases} 1, & if \quad i = j, \\ 0, & if \quad i \neq j \end{cases} \tag{1}$$

$$u_x^T.\phi_i = \left(\sum_{i=1}^{n} a_i\phi_i\right)^T.\phi_i$$
$$u_x^T.\phi_i = \left(a_1\phi_1^T + a_2\phi_2^T + \cdots + a_i\phi_i^T + \cdots + a_n\phi_n^T\right).\phi_i$$
$$u_x^T.\phi_i = a_1\phi_1^T.\phi_i + a_2\phi_2^T.\phi_i + \cdots + a_i\phi_i^T.\phi_i + \cdots + a_n\phi_n^T.\phi_i$$

using the definition provided in equation 1, the above equation becomes

$$u_x^T.\phi_i = a_i \text{ or } a_i = u_x^T.\phi_i$$

L is a large matrix, computing eigen vectors of a large matrix is time consuming. Therefore, compute eigen vectors of a smaller matrix, C:

$$C = A^T A \tag{2}$$

Let α_i be eigen vectors of C, then $A\alpha_i$ are the eigen vectors of L:

$$C\alpha_i = \lambda_i \alpha_i$$

Using equation 2

$$A^T A\alpha_i = \lambda_i \alpha_i$$

Multiply A on both sides

$$AA^T(A\alpha_i) = \lambda_i(A\alpha_i)$$
$$L(A\alpha_i) = \lambda_i(A\alpha_i)$$

Training

- Create A matrix from training images
- Compute C matrix from A.
- Compute eigenvectors of C.
- Compute eigenvectors of L from eigenvectors of C.
- Select few most significant eigenvectors of L for face recognition.
- Compute coefficient vectors corresponding to each training image.
- For each person, coefficients will form a cluster, compute the mean of cluster.

Recognition

- Create a vector *u* for the image to be recognized.
- Compute coefficient vector for this *u*.
- Decide which person this image belongs to, based on the distance from the cluster mean for each person.

1 Simple Approach

2 Face Recognition

3 Face recognition using Eigenfaces

Step 1

Obtain face images such as $I_1, I_2, \cdots, I_m$

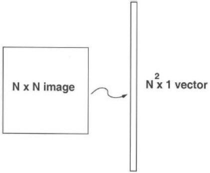
Step 2

- Represent each face image I_i in vector form Γ_i
- Suppose the training set consists of 16 images whose dimensions are 235×235 pixels.
- The resulting matrix will be 55225×16

$$I_i = \Gamma_i = [235 \times 235]$$

1	2	3
4	5	6
7	8	9

1	4	7	2	5	8	3	6	9



Step 3

Compute the average face vector Ψ

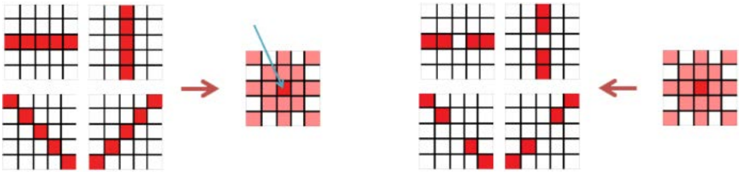
$$\Psi = \frac{1}{M}\sum_{i=1}^{M}\Gamma_i$$

$$\Psi = [55225 \times 1]$$

Step 4

Subtract the mean face, i.e.,

$$\phi_i = \Gamma_i - \Psi$$
$$\Psi_i = [55225 \times 1]$$



Step 5

Mean subtracted faces

$$A = [55225 \times 16]$$

Compute the covariance matrix:

$$C = AA^T = [55225 \times 16][16 \times 55225]$$
$$C = AA^T = [55225 \times 55225]$$

Matrix summed over all images

Step 6

Compute the eigen vector u_i of AA^T

$$AA^T u_i = \lambda_i u_i$$

$$Cu_i = \lambda_i u_i$$

$$u_i = [55225 \times 1]$$

The number of eigen vectors are 55225

Computationally not practical!

Step 6.1

Alternative solution:

$$C = A^T A = [16 \times 55225][16 \times 55225]$$
$$C = A^T A = [16 \times 16]$$

Compute the eigen vector v_i of $A^T A$

$$A^T A v_i = \lambda_i v_i$$
$$v_i = [16 \times 1]$$

The number of eigen vectors λ_i are 16

Matrix summed over all pixels

Step 6.2

How to get back the original eigen vectors:

$$Av_i = u_i$$

$$[55225 \times 16][16 \times 1] = [55225 \times 1]$$

Step 7

Calculate eigenfaces

Each eigen vector u_i is considered as eigen face which is calculated as:

$$Av_i = u_i$$

$$[55225 \times 16][16 \times 1] = [55225 \times 1]$$

All eigen faces are calculated as:

$$AV = u$$

$$[55225 \times 16][16 \times 16] = [55225 \times 16]$$

Step 8

Choose the most significant eigenfaces

1. To select those eigenvectors whose eigenvalues are above 1.
2. To choose all eigenvectors until the cumulative sum of the eigenvalues is around 95%

Suppose 6 eigen faces are selected based on the above criteria, so,

$$\theta = [u_1, u_2, u_3, u_5, u_6] = [55225 \times 6]$$

Step 9

Calculate weights

- We have to calculate the weights for each image in the training set
- The selected eigenfaces are used to calculate the weights as:

$$w = \theta^T \phi_i = [6 \times 55225][55225 \times 1]$$
$$w = [6 \times 1]$$

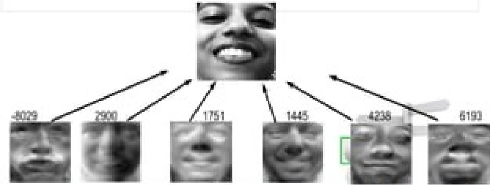
$[55225 \times 1]$ = face images

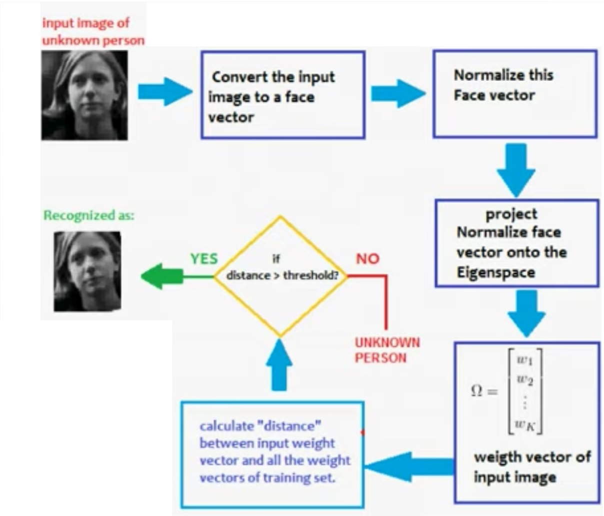
Step 10

- Represent the eigen faces in this space.
- Multiply the weight vector with the eigenfaces to reconstruct that image

$$\hat{\phi}_i = w^T\theta = [55225 \times 6][6 \times 1]$$
$$\hat{\phi}_i = [55225 \times 1]$$

- The calculated face is weighted sum of all eigen vectors




input image of unknown person
Convert the input image to a face vector
Normalize this Face vector
project Normalize face vector onto the Eigenspace
$\Omega = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_K \end{bmatrix}$
weigth vector of input image
calculate "distance" between input weight vector and all the weight vectors of training set.
if distance > threshold?
YES
NO
UNKNOWN PERSON
Recognized as:

- Given a test image Ψ
- Normalize it $\phi = \Gamma - \Psi$
- Project ϕ on the eigenface to calculate the weight vector.
$$w = \theta^T \phi_i = [6 \times 55225][55225 \times 1] = [6 \times 1]$$
- Compare test image's weight with all the training samples' weights.
- Pick the one with a minimum distance
- Reconstruct the image from the given weight
$$\hat{\phi}_i = w^T \theta = [55225 \times 6][6 \times 1]$$
$$\hat{\phi}_i = [55225 \times 1]$$

- Given a test image Ψ
- Normalize it $\phi = \Gamma - \Psi$
- Project ϕ on the eigenface to calculate the weight vector.
$$w = \theta^T \phi_i = [6 \times 55225][55225 \times 1] = [6 \times 1]$$
- Reconstruct the image from the given weight
$$\hat{\phi}_i = w^T \theta = [55225 \times 6][6 \times 1]$$
$$\hat{\phi}_i = [55225 \times 1]$$
- Compute the minimum distance between the reconstructed image and the real test image as:
$$e_d = ||\phi - \hat{\phi}_i||$$
- If
$$e_d < T_d \text{ then } \Gamma \text{ is face}$$