

# Assignment 1 - Real Time Face Recognition System

- Pratik Puri Goswami

16110121

**Introduction** – Real time face recognition using dataset from AT&T database image training has been implemented in this assignment. Real time face recognition has several applications, some of them include use in phone cameras, in surveillance systems, for law enforcement purposes and many more. To implement real time face recognition, the algorithm based on the paper by Matthew A. Turk and Alex P. Pentland has been implemented in this assignment. The steps used in implementing the algorithm are as follows:

**1) Calculating Eigenfaces for Recognition** – To calculate the Eigenfaces, the following steps were used:

- a. The face vectors were stored in a matrix such that each column of the matrix represents a face vector. The dimensions of the matrix for 30 of the AT&T database images is 10304 x 300.

```
x=[]  
  
# Reading images in matrix x using opencv  
for i in range (1,31):  
    for j in range (1,10):  
        a = "att_faces/s" + str(i)+"/" + str(j) + ".pgm";  
        x.append(numpy.reshape(cv2.imread(a, 0), -1).astype(numpy.float64))  
  
# Arranging the matrix x such that each column of matrix x represents an im  
x = numpy.asarray(x)  
x = x.transpose()
```

- b. The mean of the face vectors was subtracted from individual face vectors.

Mean:

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

Mean subtracted from face:  $\Phi_i = \Gamma_i - \Psi$

- c. The covariance matrix is given by:

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

where  $A = [\Phi_1 \Phi_2 \dots \Phi_M]$  ( $N^2 \times M$  matrix)

Instead of computing  $AA^T$ ,  $A^T A$  was computed to reduce the processing time. Since,  $AA^T$  is a  $[N^2 \times N^2]$ , while  $A^T A$  is a  $[M \times M]$  matrix. Their eigen values remain same and their eigenvectors are related by  $u_i = A v_i$

Thus, the computation time for calculating the eigenvectors and eigenvalue was reduced.

- d. K eigenvectors corresponding to the K best eigenvalues were selected. K = M/2 eigenvectors were selected for this case corresponding to the K best eigenvalues, where M = total number of images used in training. The k eigenvector matrix was then normalized.

$$\|u_i\| = 1$$

- e. Once the eigenvectors were selected their weights were calculated by projecting the (Face vectors - mean) vectors onto the eigenvector plane.

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

Where w vector corresponds to the weight vector

**2) Face identification** – The following steps were used for face identification:

- a. Similar steps were followed for a new image. This new image is the image of a person whose faces have been used in the training of the images.
- b. The weight vector for a single new image was calculated and compared with the existing weights of the trained images. The weight vector of the person closest to the weight vector of the new image was declared as the identified person from the image.
- c. To find the closest weight vector, the weight vectors from pre-trained data and new image were subtracted and then normalized. The difference was then normalized. The person whose weight vector had the minimum value corresponding to the normalized output was declared as the person in the image.

$$e_r = \min_l \|\Omega - \Omega^l\|$$

- d. If  $e_r$  was found to be more than the threshold value, that means that the image given for identification does not belong to any of the images used for training.

**3) Face recognition** – The following steps were used for face recognition -

- a. Similar steps as above were followed and new weight vector was calculated for the unknown image.
- b. The weight vector was then multiplied with the reduced eigenvector of the unknown image and the new (Face vectors - mean) vector was found, which was then compared with the original (Face vectors - mean) vector without the data loss. (Lets say (Face vectors - mean) vector = phi)

$$\hat{\Phi} = \sum_{i=1}^K w_i u_i$$

- c. They were compared by finding the normalized output of the difference between the original and the new phi.

$$e_d = \|\Phi - \hat{\Phi}\|$$

- d. If  $e_d$  was found to be less than a threshold value, (4000 in this case), a face was present in the unknown image and face recognition was successful or else no face was present in the unknown image and face recognition was unsuccessful.

$$e_d < T_d,$$

## References:

- a) M. A. Turk and A. P. Pentland, "Face recognition using eigenfaces," *Proceedings. 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Maui, HI, USA, 1991, pp. 586-591.  
doi: 10.1109/CVPR.1991.139758  
keywords: {computerised pattern recognition;eigenvalues and eigenfunctions;unsupervised learning;eigenfaces;human faces;face recognition system;two-dimensional recognition;feature space;face space;face images;eigenvectors;Face recognition;Face detection;Humans;Character recognition;Computer vision;Head;Eyes;Nose;Computational modeling;Image recognition},  
URL: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=139758&isnumber=3774>
- b) Data download link: <https://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html>
- c) <https://www.learnopencv.com/eigenface-using-opencv-c-python/>
- d) [http://www.vision.jhu.edu/teaching/vision08/Handouts/case\\_study\\_pca1.pdf](http://www.vision.jhu.edu/teaching/vision08/Handouts/case_study_pca1.pdf)
- e) <http://www.cbsally.com/black-and-white-photography.html>