**EE 331 Probability and Random Processes**

**Assignment 1**

**Real Time Face Recognition**

**Name:** Girish Chandar G

**Roll No:** 16110057

**Introduction:**

The approach followed for the detection of human images treats face recognition as a 2-D recognition problem. Face images are projected onto a ‘face space’ whose basis vectors are the eigenvectors(called ‘eigenfaces’) of covariance matrix of image data matrix. The ‘face space’ best encodes the features of the face images. The database used is [ORL Face Database](http://www.cl.cam.ac.uk/Research/DTG/attarchive/pub/data/att_faces.zip).

**Eigenfaces:**

An important part of face recognition is to identify and extract features of face images. To extract the information from the face images we need to capture the variation in the face images, independent of any judgement of features and use it to encode and compare images.

The mathematical process followed it to calculate the eigenvectors of the covariance matrix of the set of images. These eigenvectors represent and capture set of features from the image. When displayed these eigenvectors show a blurry, ghostly image and are called ‘eigenfaces’.

Each face images can be perfectly represented as a linear combination of the eigenfaces. But the face images can also be approximated by linear combination of a set of ‘best’ eigenvectors. These eigenvectors can be calculated by PCA(Principal Component Analysis) by selecting the ‘best’ M eigenvectors corresponding to M largest eigenvalues. The face images are then projected onto this reduced face space and the weights are obtained which show how close the image is to the face space. The procedure followed in brief as follows are as follows:

* Acquire the training set of images and calculate the eigenfaces and also obtain the corresponding weights
* Input the test image and project it onto the face space and obtain the weights.
* Compare the difference of weights with the weights of face images already obtained. Apply a threshold to detect whether it is a face or not(Face Detection) and classify based on the minimum difference obtained(Face Recognition).

**Calculating Eigenfaces:**

Let face image *I* be a NxN matrix. Consider a N2x1 vector Γ corresponding to the face image *I*. We now calculate the mean face which is the sum of all Γ divided by total number of images. Let the total number of training images be M

The mean image obtained in the code is shown below along with some sample images from training dataset.

Sample Images from the training dataset



Mean Image

The mean face is then subtracted from each face image.

The covariance matrix is then calculated as follows:

where

But calculating the eigenvectors of *AAT* is very difficult and impractical as the size of the matrix is very large (in our program its 30912x30912). So we calculate the eigenvectors of *ATA* (at max in code 40x40) and then obtain the eigenvectors of *AAT* from them.

Let the eigenvectors of *ATA* be *v* and that of *AAT* be *u*. *u* and *v*  are related as follows.

The set of *u*’scorresponds to the best M eigenvectors of C. The number of eigenvectors can be reduced even more by K eigenvectors with corresponding K largest eigenvalues. Then the eigenvectors are normalized. These eigenvectors are called eigenfaces and they are the basis vectors for the face space. Some example eigenfaces before normalizing are shown below.

**Representing face images onto basis:**

Next we compute the weights by projecting the face images onto the face image

where is weight of ith images corresponding to

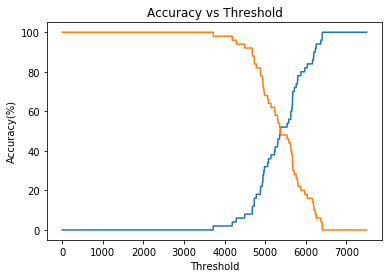
We can get the weight matrix *W* as

where the ith column contains the weights corresponding to the firstith face image.

**Face Detection and Recognition:**

An input test image is detected as a face or not based on how close it is to the face space.

* The test image is subtracted by the mean image and is projected on the face image and the weights are obtained corresponding to the eigenfaces *wt*.
* Next we compute , where is called the distance within face space.
* Next we select a threshold *T* is selected such that when then the test image is detected as a face. A plot of accuracy of detection and error of detection vs threshold is shown below. For a *T* of 6000 the accuracy comes out to be 82%.



* The closest image predicted for a given test image can be found by finding the training face image *i* for which for which is minimum. The prediction matrix is found in the program and is described in the READ\_ME(Face\_Recog). The accuracy of face recognition is 95%.

**References:**

* <https://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html>
* <http://didawiki.cli.di.unipi.it/lib/exe/fetch.php/mcl/1992_turk_eigenfaces_for_recognition.pdf>
* <http://www.vision.jhu.edu/teaching/vision08/Handouts/case_study_pca1.pdf>
* <https://ieeexplore.ieee.org/document/139758>
* <https://www.learnopencv.com/face-reconstruction-using-eigenfaces-cpp-python/>
* <https://www.learnopencv.com/eigenface-using-opencv-c-python/>