ES 331 - Probability and Random Processes

Assignment 2 - Report

Siddhesh Kanawade, 20110199

TASK:

Design a real-time face recognition system using eigenfaces.

GOALS:

The primary goal of the task is to implement an algorithm for face recognition by using the method of Eigenvectors. The model(once trained on an authorized dataset with significant variation) would have the ability to learn new faces in an unsupervised manner.

DATASET:

Used dataset: AT&T "The Database of Faces" (formerly "The ORL Database of Faces")

Description: There were ten different images for 40 distinct subjects. According to the “face-rec.org” the images were taken at different times, varying the lighting, facial expressions and facial details.

Dimensions: height = 112 and width = 92

Image-format: **.pgm**

Image-format-used: **.png**

ALGORITHM:

Principle Component Analysis(PCA) is used to reduce the dimensionality of a dataset to lower dimension subspace by linear projection such that reconstruction error is minimized.

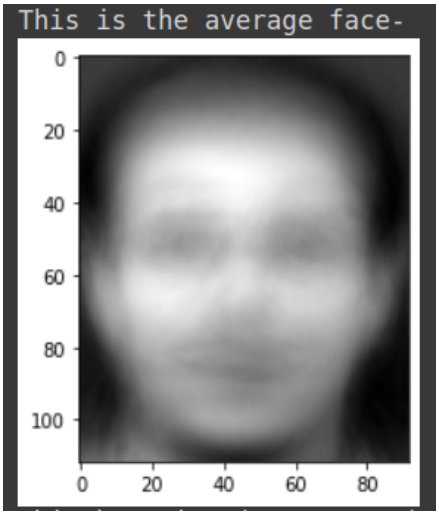
Find the vectors of the covariance matrix of the image that would best account for the distribution of the image in the entire space(eigenvectors with highest eigenvalues). Each image is represented as a linear combination of eigen faces.

Approach:

1. Process the dataset. Here, it was found that there was loss of data when we read .pgm file with popular python libraries like rasterio and PILLOW, hence drafted the code to convert .pgm files to .png files.

2. Store the image matrix into an array using the numpy library.

3. Split the image matrix into train and test matrix.
4. An average face is calculated. This is then subtracted from each of the vectors and the resulting difference is then stored in a matrix. Then the normalized faces are calculated and stored into the Phi matrix.



5. Find the Covariance matrix by taking the dot product of Phi.transpose() and Phi. We are taking the reverse product so that the dimensions is reduced from W*H to training size(SUB*train_per_sub).
6. Find the eigenvalues and eigenvectors from the covariance matrix. Define eigen pairs which have eigenvalues and corresponding eigenvectors which are then sorted in descending order of which top k pairs are chosen.

$$\mathbf{u}_l = \sum_{k=1}^{M} \mathbf{v}_{lk} \mathbf{\Phi}_k, \qquad l = 1, \ldots, M$$

7. Projecting the Normalized images onto the K vector Space.
8. To predict a test image, we convert the test image to the same size as the train images.
9. The test images are subtracted from the average face to get the normalized images.
10. Now, we project the test images on our eigenspace and calculate the distance of weights between our test images with each train image.
11. The minimum distance with the train image is the matched image with our test image. However, a threshold is required so that the minimum distance is below that threshold. Otherwise, it is an unknown face.

RESULTS:

Split: 7:1

Correct count : 114

Incorrect_count: 6

Accuracy of the model is **95.0 %**

The accuracy above is for the test data splitted from the original data. The accuracy will be lower for a new image.

REFERENCES:

1. P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001, Kauai, HI, USA, 2001, pp. I-I, doi: 10.1109/CVPR.2001.990517.
2. 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.
3. Pawangfg,“ML|Face Recognition Using Eigenfaces(Using PCA)”, https://www.geeksforgeeks.org/ml-face-recognition-using-eigenfaces-pca-algorithm/.