Name: S Sri Manish Goud
Roll Number: 20110174

PROBABILITY AND RANDOM PROCESSES ASSIGNMENT - 2

Assumptions: All the images are upright.

Working of the Algorithm used:

- Detection of faces: Firstly, I extracted the images that contained faces in them using the extracting_faces() function. Then make a separate folder containing all those detected images. Now, this folder is used to recognise the images.
- Average image: The faces are flattened into a 1D vector and the average is found

$$\text{Average image} = \text{mean\_image} = \sum_i flattened\ face_i / Total\ number\ of\ images$$

- Normalized image: The normalized faces for each face are calculated as below

$$\text{Normalized image} = \text{image} - \text{Average image}$$

- Covariance matrix: We have to find the covariance matrix of the matrix containing the images as columns

$$\text{matrix} = [\text{img1, img2, img3, ……..}]$$
$$\text{Covariance matrix} = \text{matrix} \cdot \text{matrix}^T$$
$$\text{The size of matrix is } N^2 \times M$$
$$M = \text{Total number of images}$$
$$\text{The size of covariance matrix is } N^2 \times N^2$$

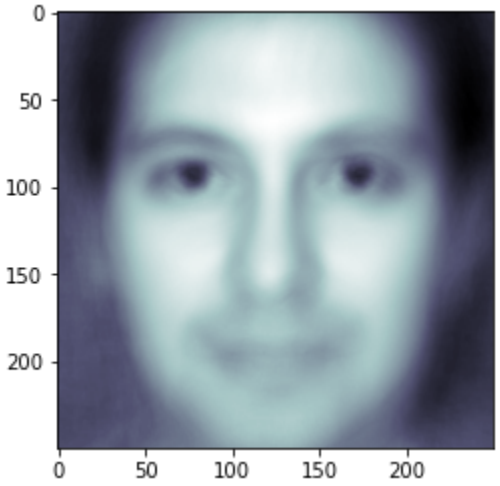
Which is very huge so we use dimensionality reduction and reduce it to MxM size.

$$\text{By making the Covariance matrix} = \text{matrix}^T \cdot \text{matrix}$$

- Eigenvectors from Covariance matrix: Since the reduced covariance matrix is of size MxM we get utmost M eigenvalues and hence M eigenvectors. But, since the value of M may be large we take a value K and take K largest eigenvectors of the matrix and these K eigenvectors are actually the K eigen faces. I took K as 36.
- Finding the weights: Weights of the image are calculated by projecting them onto the eigenfaces.
- Face recognition: We use Support Vector Machines(SVM) to recognize the faces using the weights. If the vector difference between the test image and train image is very less than the specified threshold set by SVM then the image is recognized and returned.

Results Obtained:

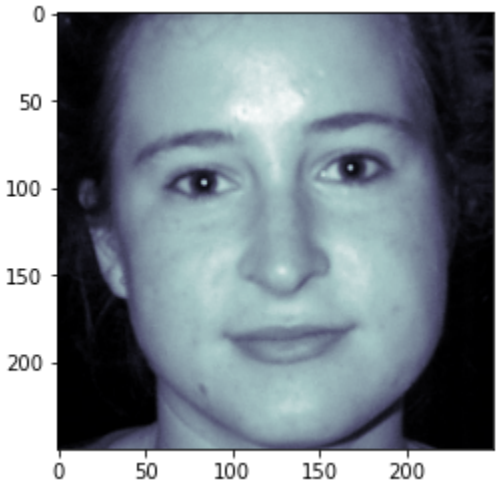
Mean_image(Average face):

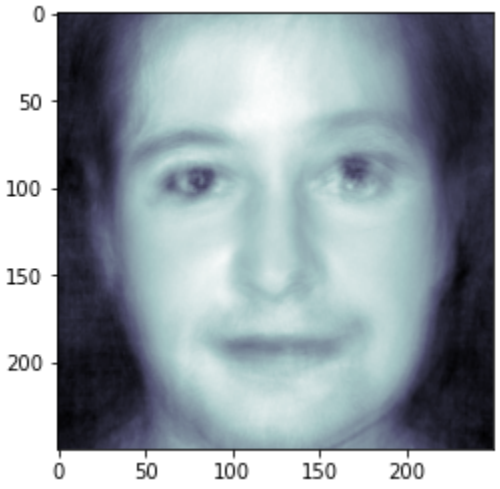


Eigenfaces obtained for K=36:


eigen faces
eigenface 1
eigenface 2
eigenface 3
eigenface 4
eigenface 5
eigenface 6
eigenface 7
eigenface 8
eigenface 9
eigenface 10
eigenface 11
eigenface 12
eigenface 13
eigenface 14
eigenface 15
eigenface 16
eigenface 17
eigenface 18
eigenface 19
eigenface 20
eigenface 21
eigenface 22
eigenface 23
eigenface 24
eigenface 25
eigenface 26
eigenface 27
eigenface 28
eigenface 29
eigenface 30
eigenface 31
eigenface 32
eigenface 33
eigenface 34
eigenface 35
0
50
100
150
200
0
100
200

Reproduced image from the eigen faces(Random):





Final image recognition predictions:

Accuracy:

A) When run on a training dataset input

	precision	recall	f1-score	support
1	1.00	1.00	1.00	16
2	1.00	1.00	1.00	14
3	1.00	1.00	1.00	4
4	1.00	1.00	1.00	18
5	1.00	1.00	1.00	12
6	1.00	1.00	1.00	19
7	1.00	1.00	1.00	16
8	1.00	1.00	1.00	4
9	1.00	1.00	1.00	15
10	1.00	1.00	1.00	5
11	1.00	1.00	1.00	4
12	1.00	1.00	1.00	5
13	1.00	1.00	1.00	12

14	1.00	1.00	1.00	15
15	1.00	1.00	1.00	23
16	1.00	1.00	1.00	21
17	1.00	1.00	1.00	4
18	1.00	1.00	1.00	14
19	1.00	1.00	1.00	16
20	1.00	1.00	1.00	17
21	1.00	1.00	1.00	18
22	1.00	1.00	1.00	18
23	1.00	1.00	1.00	16
25	1.00	1.00	1.00	1
26	1.00	1.00	1.00	5
27	1.00	1.00	1.00	15
28	1.00	1.00	1.00	15
accuracy			1.00	342
macro avg	1.00	1.00	1.00	342
weighted avg	1.00	1.00	1.00	342

Train Score Accuracy: 1.0

The accuracy is 1.0 as expected because we are feeding the training data set as input to the ML model inorder to make it learn as it learns this input also.

B) On a testing dataset input:

	precision	recall	f1-score	support
1	1.00	0.80	0.89	5
2	0.86	1.00	0.92	6
3	0.00	0.00	0.00	1
4	1.00	0.75	0.86	4
5	1.00	1.00	1.00	9
6	1.00	1.00	1.00	4
7	1.00	1.00	1.00	4
9	1.00	1.00	1.00	6
11	1.00	1.00	1.00	1
13	1.00	1.00	1.00	6
14	1.00	1.00	1.00	5
15	1.00	1.00	1.00	2
16	1.00	1.00	1.00	2
18	1.00	1.00	1.00	5
19	1.00	1.00	1.00	3
20	0.00	0.00	0.00	0
21	1.00	1.00	1.00	2
22	0.50	1.00	0.67	2

23	1.00	0.83	0.91	6
24	0.00	0.00	0.00	1
27	1.00	1.00	1.00	5
28	0.88	1.00	0.93	7
accuracy			0.94	86
macro avg	0.83	0.84	0.83	86
weighted avg	0.94	0.94	0.94	86

Test Score Accuracy: 0.9418604651162791

When run on a test dataset input the model produces an accuracy of around 94% which is good. Hence our model is able to recognise the images with good accuracy.

References:

https://www.geeksforgeeks.org/ml-face-recognition-using-eigenfaces-pca-algorithm/
https://ieeexplore.ieee.org/abstract/document/6793549
https://ieeexplore.ieee.org/document/139758