

Face Recognition

Shrey Gupta

Software Used : MATLAB

Instructions on how to implement the code:

There are parts of the code which have been commented out . These are the part of the codes where an image is to be displayed. Since at one time, I was focusing on displaying only one element in the figure, I commented the other display functions. To display any specific image, remove the “%” sign before the line of code. The deleted “%” needs to be inserted at the part of the code where the any other active display function is present, so that only one figure is produced and it is easier to understand the output.

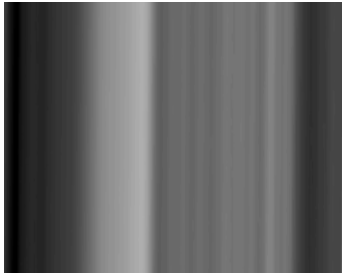
The lines of code where display lines are present are lines 46-48, 96-98, 101-103, 106-108, 111-113, 116-118, 121-123, 126-128, 131-133, 163-165, 172-174, 200.

The mean face image (display part is written at the lines 46-48).



The eigenfaces in U come out as follows:

Eigen1 =



Eigen2 =



Eigen3 =



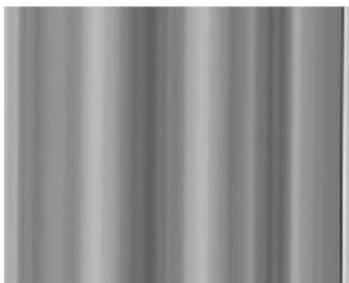
Eigen4 =



Eigen5 =



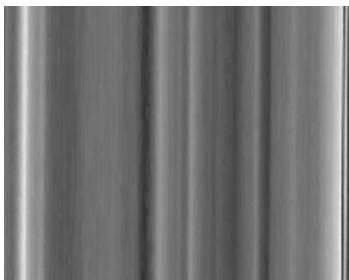
Eigen6 =



Eigen7 =



Eigen8 =



The value of Threshold1 is kept as $6e+07$ (which is equivalent to $6 * 10^7$)

The value of Threshold0 is difficult to decide from the given set of data. The value lies in between the distance_j value of the testfaces. As a result for one particular value of Threshold0,

most test images , would even fail to be recognized as an image. Hence, only Threshold1 is considered in this algorithm.

The distance₀ for each of the images, distance_j can be shown as follows.

Image	Distance0	DistanceJ
Subject01.centerlight	1.5137e+11	2.6060e+07
Subject01.happy	3.7022e+11	1.4011e+07
Subject01.normal	4.9231e+11	0
Subject02.normal	2.8724e+11	0
Subject03.normal	3.3578e+11	0
Subject07.centerlight	5.1149e+11	2.3159e+07
Subject07.happy	8.8064e+10	2.6006e+07
Subject07.normal	4.5452e+11	0
Subject10.normal	4.1726e+11	0
Subject11.centerlight	7.2378e+11	5.0995e+07
Subject11.happy	9.3122e+11	6.2184e+06
Subject11.normal	9.0761e+11	0
Subject12.normal	5.6101e+11	2.5597e+07
Subject14.happy	3.5652e+11	1.1498e+07
Subject14.normal	3.3623e+11	0
Subject14.sad	3.1547e+11	4.2058e+06
Subject15.normal	7.6179e+11	0

For the given apple image the distance0 is 3.6344e+11 and distancej is 2.3673e+07. It corresponds to the first training image.

The PCA coefficients can be given in the following table

PCA1	PCA2	PCA3	PCA4	PCA5	PCA6	PCA7	PCA8
1.3287e+07	4.0993e+06	2.7042e+07	3.1494e+07	-3.7208e+07	-8.0896e+07	-2.5925e+07	6.8107e+07
-5.3415e+07	3.1066e+07	-1.6124e+07	3.2324e+07	-7.9081e+05	1.5205e+07	-1.8758e+07	1.0493e+07
8.9937e+06	1.5647e+07	-8.3928e+06	-1.2146e+07	2.8655e+06	2.6895e+05	-1.1135e+07	3.8990e+06
-6.7272e+06	-3.1004e+06	5.0036e+06	-6.0512e+06	1.3198e+07	-3.9727e+06	-2.2832e+06	3.9331e+06
-2.7299e+06	3.0377e+06	7.2204e+06	-4.5670e+06	-4.7116e+06	1.9970e+06	2.6125e+05	-5.0778e+05
2.4375e+05	-3.9114e+06	-7.3537e+05	-1.1092e+06	-1.4492e+06	4.4566e+06	-2.1654e+06	4.6702e+06
-1.2519e+06	5.9115e+05	-2.0934e+06	-1.7820e+06	-6.9521e+05	-7.3317e+05	3.8759e+06	2.0886e+06
1.6125e-08	-6.3795e-09	-1.1936e-09	-1.2735e-08	-4.3318e-09	-1.5048e-09	1.1538e-08	-1.5184e-09

For each input test image, the corresponding output image from the training database is as follows

To check the final output , the algorithm points the index towards the image in the training image array. The recognized image is the one which corresponds to this training image. So I have provided the output image for the index corresponding to the training image array.

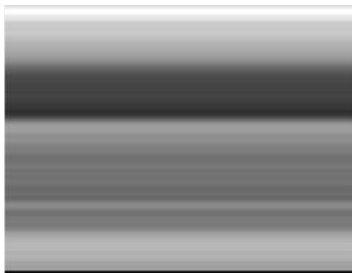


Input Image :

Normal Image:



Reconstructed Image:





Output Image:



Input Image :

Normal Image :



Reconstructed Image:



Output Image:



Input Image :

Normal Image:



Reconstructed Image:



Output Image:



Input Image :

Normal Image :



Reconstructed Image:

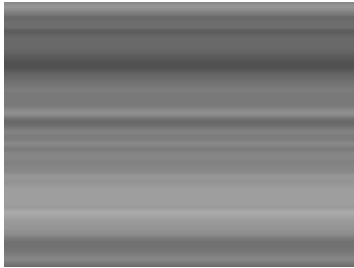


Output Image:



Input Image :

Normal Image:



Reconstructed Image:



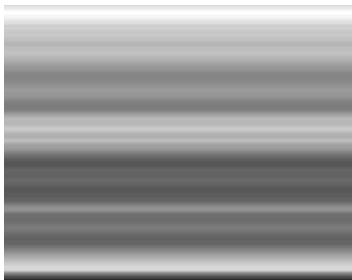


Output Image:

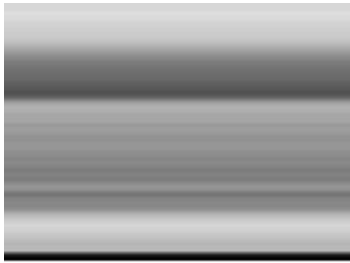


Input Image :

Normal Image::



Reconstructed Image:



Output Image:



Input Image :

Normal Image:



Reconstructed Image:



Output Image:



Input Image :

Normal Image:



Reconstructed Image



Output Image:

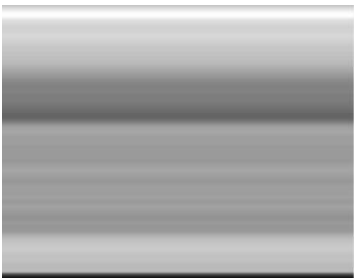


Input Image :

Normal Image:



Reconstructed Image:



Output Image:

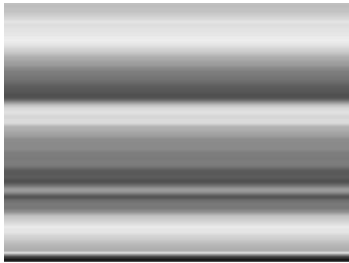


Input Image :

Normal Image:



Reconstructed Image:



Output Image:

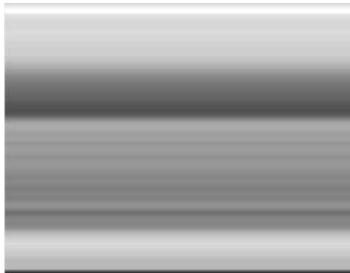


Input Image :

Normal Image:



Reconstructed image =



Output Image:

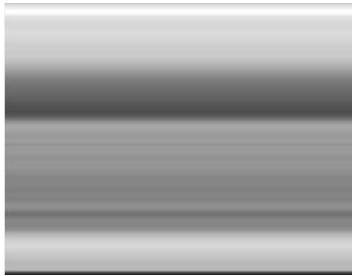


Input Image :

Normal Image:



Reconstructed image:



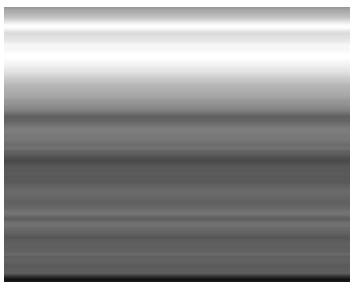


Output Image:

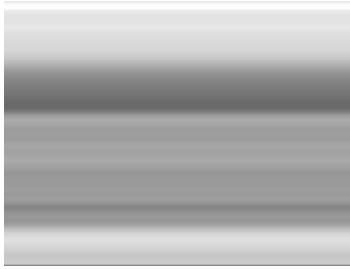


Input Image :

Normal Image:



Reconstructed Image:



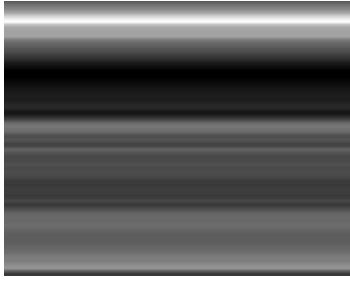
Output Image:



Input Image :

Normal Image:





Reconstructed Image:



Output Image:

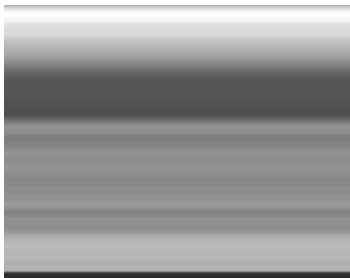


Input Image :

Normal Image:



Reconstructed Image:

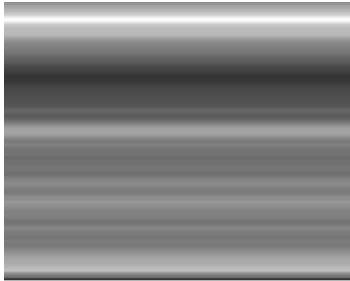


Output Image:

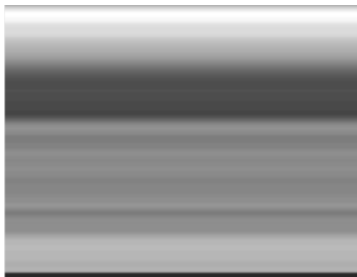


Input Image :

Normal Image:



Reconstructed Image:

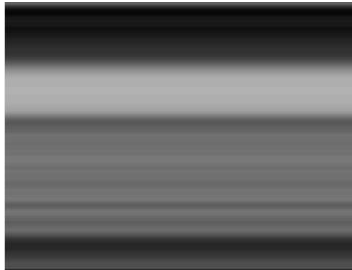


Output Image:



Input Image :

Normal Image:



Reconstructed Image:





Output Image:

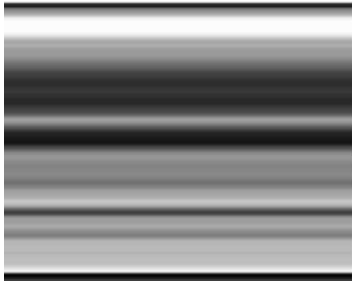


Input Image :

Normal Image:



Reconstructed Image:



Output Image:

It can be seen that out of the 17 test images, my algorithm returns 13 as the correct recognition, but that is not the case for 4 test images and the apple image. This is because Eigenface is a very naïve algorithm and will not give a 100% efficiency. The apple is detected as a face, as no Threshold0 is defined, because it is difficult to identify a threshold0 for the given imageset. It's value always lies somewhere in between which if chosen will result in recognizing face images as non-faces.

Even though, the eigen faces, normal and reconstructed faces were not coming out to be as faces, it is only because there is some issue in the display library of my MATLAB. The use of eigenface and the vectors however is correct in my algorithm as 13 input test image correctly indexed to the correct training image.