Name: PRATHMESH R. KUMBHARE

CS6001 Assignment 2: Face Classification

REPORT

A. OUTLINE FOR PROGRAM ALGORITHM:

The algorithm for implementing the program is given below in steps:

1. Matlab function files are created for each model separately. Different '.m' files are created for models by their names. The program then executes the corresponding model & saves all the workspace variables for corresponding model in the same folder. Images for Mean and co-variance of face as well as background images (training dataset) are plotted to provide some visualization aid.
2. Factors which can be varied to get different results for a model are: calculation of multivariate normal probability distribution function (mvnpdf) & covariance matrix.
3. The program firstly creates an 'IMAGE_MATRIX' for training face images by looping through each file in the directory of training images. Inside the loop, the image data is converted into column vector and a matrix consisting column vectors of all images is saved. The mean is calculated for this matrix and then it is averaged. Covariance matrix is computed from the earlier obtained value of mean. Since in this program we are only considering diagonal form of covariance matrix. This diagonal covariance matrix is saved for future calculations. Similarly, mean and diagonal covariance is calculated for training background images.
4. For the inference algorithm, firstly an 'IMAGE_MATRIX' is created for testing face images using process explained in the above point. This image data is passed through a function named 'MVNGD1'. This function computes multivariate normal probability distribution function for the images using mean and covariance estimated from training images. Therefore, this data is required to pass through the function two times (for estimated face parameters & background parameters.) If the probability for face parameter is higher than background, then it is classified as face image. Otherwise it is a background image, which is not true. So, it is counted as false face image. Similar process is carried out for testing background images to classify.
5. In the last part accuracy for face & background image detection is computed (In percentage.) All the workspace variables are saved in '.mat' file under the model name. Also, image number for falsely detected face and background images are saved for further analysis.

B. RESULTS:

This is the table for all the models implemented using the program for **diagonal covariance matrix**:

	MODEL	FACE ACCURACY (%)	BACKGROUND ACCURACY (%)	TOTAL ACCURACY (%)
1	RGB	90.94	63.29	71.35
2	HSV	65.94	49.11	54.02
3	YCbCr	79.78	11.2	59.79
5	R	49.56	80.85	71.73
6	G	66.37	73.93	71.73
7	B	74.56	74.46	74.49
8	GRAY	66.37	75	72.48
9	GRADIENT	96.55	77.83	83.29

Table for models using the program for **diagonal covariance matrix using MATLAB function mvnpdf**:

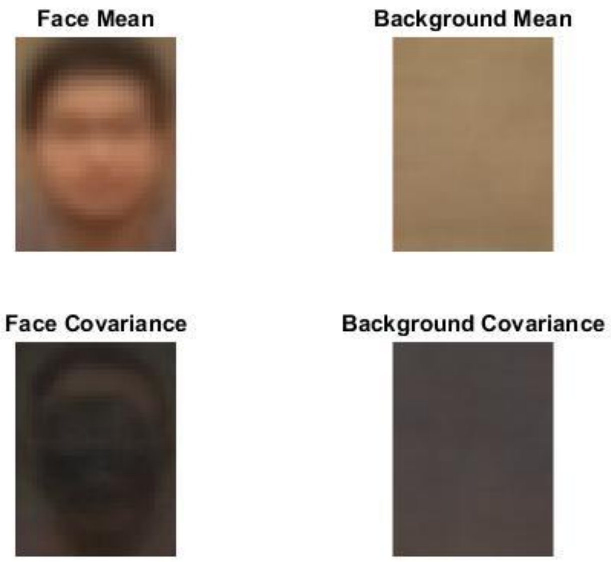
	MODEL	FACE ACCURACY (%)	BACKGROUND ACCURACY (%)	TOTAL ACCURACY (%)
1	RGB	93.1	0	27.13
2	HSV	98.7	0	28.76
3	YCbCr	100	0	29.14
4	R	100	17.73	41.7
5	G	100	24.46	46.48
6	B	100	37.23	55.52
7	GRAY	100	21.09	44.09
8	GRADIENT	0	0	0

Results for each model are given below separately.

1. RGB Model:

This model uses values of red, green and blue to describe the image.

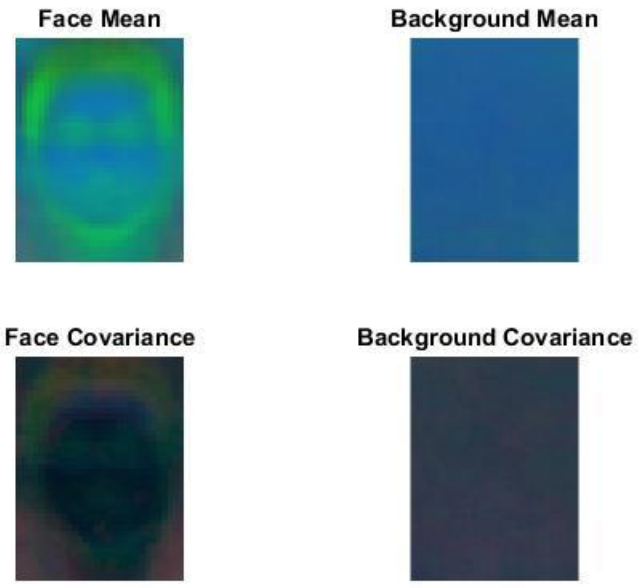
Mean & Covariance for face & background training images:



2. HSV model:

This model converts RGB values into Hue, Saturation & Value to describe image.

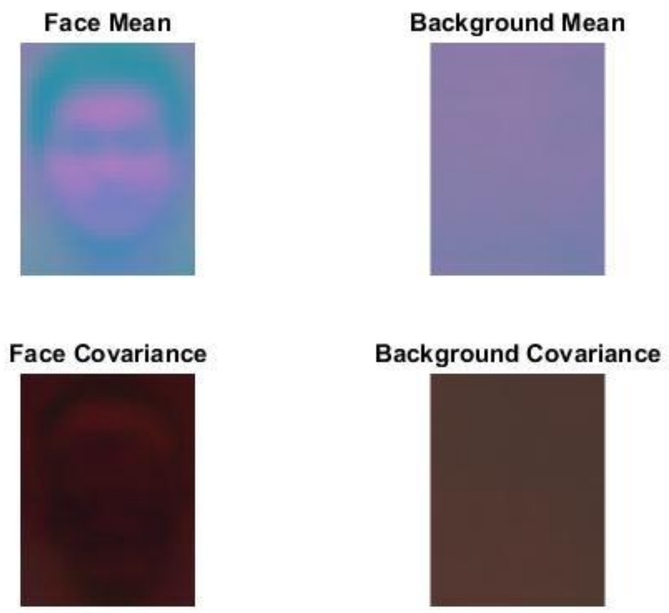
Mean & Covariance for face & background training images:



3. YCbCr model:

This model converts RGB values into Y, Cb and Cr values to describe the image.

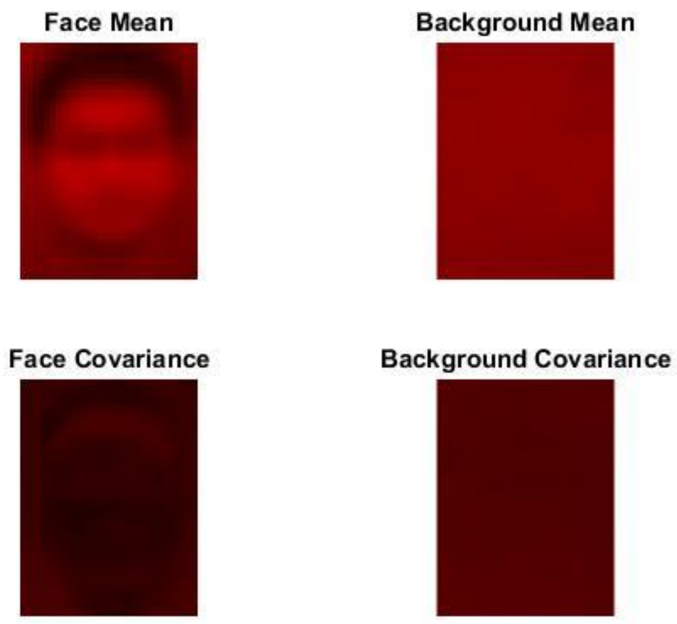
Mean & Covariance for face & background training images:



4. R Model:

In this model, values only from red channel of the image are taken to estimate mean and covariance.

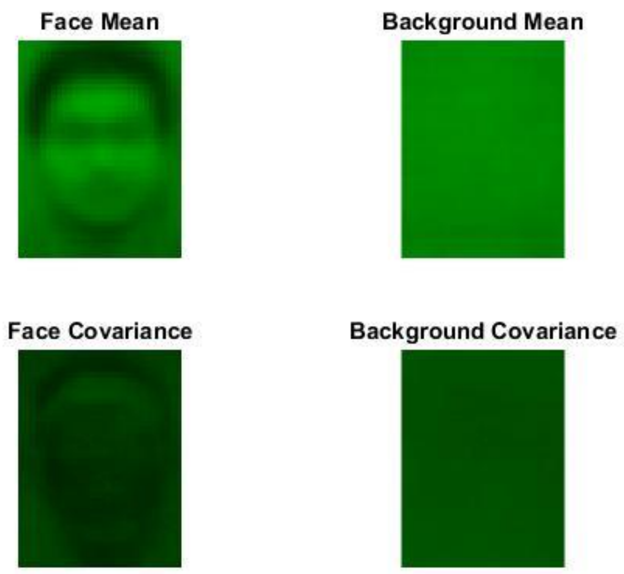
Mean & Covariance for face & background training images:



5. G Model:

In this model, values only from green channel of the image are taken to estimate mean and covariance.

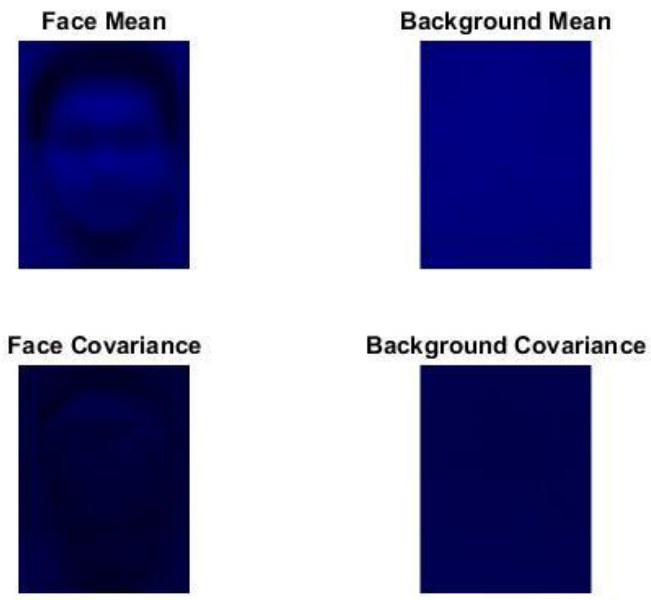
Mean & Covariance for face & background training images:



6. B Model:

In this model, values only from blue channel of the image are taken to estimate mean and covariance.

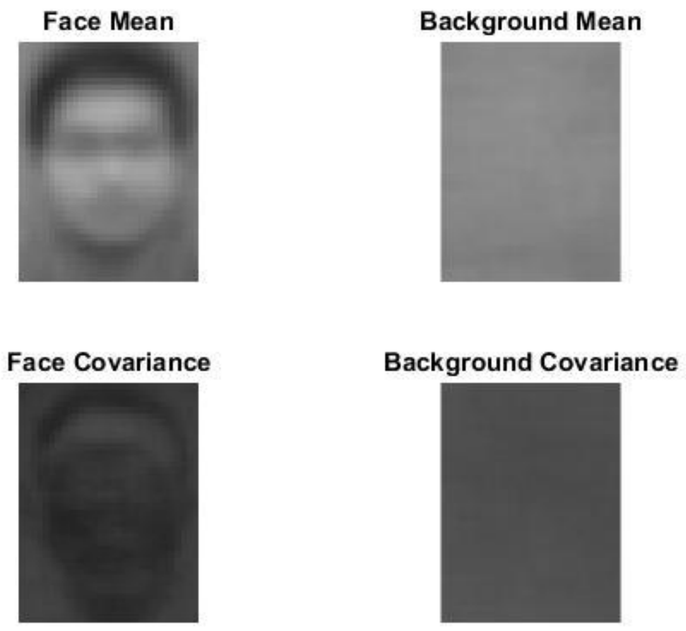
Mean & Covariance for face & background training images:



7. GRAY Model:

In this model, image is converted to grayscale. These values in single channel are taken to estimate mean and covariance.

Mean & Covariance for face & background training images:



8. GRADIENT Model:

In this model, the image matrix is used to compute gradient magnitude and gradient direction matrix using a standard algorithm called ‘sobel operator’. This matrix is used to estimate mean and covariance.

Mean & Covariance for face & background training images:

1. Visualization for this model is not included in this program. Although, you can create images by passing the mean and covariance matrix of RGB model through ‘im_grad’ function.

C. General Observation & Comments:

1. Selection of algorithm for computation of probability distribution is an important factor. MATLAB has ready to use function named ‘mvnpdf’ which computes probability distribution for given mean and sigma values. Similar calculation can be done by creating function separately. As you can see from the tables presented earlier, in built function is less accurate. But it is considerably faster in computation. The reason for accuracy difference is that the created function does computation in logarithm scale, which eliminates the ‘infinity’ problem encountered in the ‘mvnpdf’. Hence, a trade-off between accuracy & computation is there.
2. Similarly, as the complexity of the model increases, the accuracy increases. But this also increases computation time. This can be observed from table 1. The accuracy for RGB model is more than R, G, B, Gray models.

D. Summary and Concluding Comments:

1. The aim of this assignment was to classify images into face and background type. From the results, it is evident that the algorithm is moderately efficient in classification. The main two reasons for this are covariance matrix & probability density function.
2. It could be possible to increase the accuracy of the algorithm by considering full covariance matrix (given that we can solve the problem of ‘positive definite matrix’) and replacing the multivariate normal distribution curve. More complicated curves like Mixture of Gaussians can be implemented.
3. Another approach would be to increase the variations in the position of faces of training images for the algorithm so that it can classify the challenging images better.