Name: PRATHMESH R. KUMBHARE

CS6001 Assignment 3: Face vs. Non-face with Mixture of Gaussian

REPORT

A. OUTLINE FOR PROGRAM ALGORITHM:

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

1. Matlab 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 (for 3 classes as k=3) 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. This image matrix is passed through function named 'MOG_FCN' in which number of components of Gaussian (classes) and precision are defined. This function updates the parameter: lambda, mean & sigma and returns the final values. Another point to note that in the analysis, diagonal covariance matrix is used as there are some problems in the full covariance matrix computation (discussed in the last section) and it gives better results than spherical covariance matrix.
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 'MVGD1'. This function computes multivariate normal probability distribution function for the images using mean and covariance estimated from training images (for each class). Therefore, this data is required to pass through the function two times (for estimated face parameters & background parameters.) Then in the next part, the probability for each class is taken together. 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. One important point worth mentioning is that the values of probability are not normalized. So, these are not small numbers.
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

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

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. 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. We can see that the MATLAB in-built function 'mvnpdf' is less accurate than our made function 'mvgd1'. The computation for full covariance matrix is quite troublesome. MATLAB shows the determinant of that matrix as zero. I tried 2 methods to eliminate this issue. In first approach, I tried to replace all the values less than 0.1 to 0.1. In the second approach, I tried to use 'vpa' command from statistics toolbox to represent the values accurately up to 10 values after point. Both the approaches didn't yield any different results. Another possible reason for this could be rank deficient system. As the number of images from which we compute covariance matrix is comparably less than the dimension of covariance matrix, the rank for the matrix is not full. This results in the determinant value as zero.
2. Similarly, as the complexity of the model increases, the accuracy increases. But this also increases computation time. This can be adjusted by changing the values of k (number of components) and precision value (difference between successive iteration values) 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 varies in the efficiency. The main two reasons for this are covariance matrix computation & initialization.
2. It could be possible to increase the accuracy of the algorithm by considering full covariance matrix (by solving the 'positive definite matrix' issue) and replacing a better algorithm for initialization of the conditions. In our algorithm, we randomly initialize the parameters lambda, mean & sigma from the given training dataset. Therefore, it is largely possible that the program reaches a local solution. Only way to get better results is to re run the program multiple times. There are various other algorithms which initialize the parameters such that a global solution is achieved quickly.
3. Another approach would be to increase the variations in the position of faces of training images and number of images for the algorithm so that it can classify the challenging images better by reducing the rank deficient system problem discussed earlier.