

# **Indian Institute of Technology, Guwahati**



## **EE 657 Assignment Report**

**Ankit Chahar -140102007  
Raunak Agrawal -140107005**

# Task 1

To run the code first run TrainCode.m file to train according to the given images and then run Code1.m to see the classified output.

The TrainCode.m is running in approximately 16 seconds.

The code1.m is running in approximately 8 seconds.

The total time taken by code to run is approximately 25 seconds.

Here we made a rudimentary pattern classifier using bayesian classifier concept. The training characters given to us were 'e', 'c', 'l'. The classifier was successfully made with an overall average accuracy of 92.55%.

## 1(a) - Accuracy

The average accuracies are :-

Case i) For the case where the covariance matrix is Identity matrix.

Accuracy = 90.667

Case ii) For the case where the covariance matrix is common diagonal covariance matrix.

Accuracy = 93.333

Case iii) For the case where the covariance matrix is modelled by separate covariance matrix.

Accuracy = 93.667

## **1(b) – Misclassification Examples**

### **Case (i)**

- a)The image TestCharacters/1/215.jpg is incorrectly classified to 2.  
The correct class is 1.
- b)The image TestCharacters/1/236.jpg is incorrectly classified to 3.  
The correct class is 1.
- c)The image TestCharacters/2/210.jpg is incorrectly classified to 3.  
The correct class is 2.
- d)The image TestCharacters/2/273.jpg is incorrectly classified to 1.  
The correct class is 2.

### **Case (ii)**

- a)The image TestCharacters/1/245.jpg is incorrectly classified to 2.  
The correct class is 1.
- b)The image TestCharacters/2/232.jpg is incorrectly classified to 3.  
The correct class is 2.
- c)The image TestCharacters/2/245.jpg is incorrectly classified to 1.  
The correct class is 2.
- d)The image TestCharacters/3/238.jpg is incorrectly classified to 2.  
The correct class is 3.

### **Case (iii)**

- a)The image TestCharacters/1/201.jpg is incorrectly classified to 2.  
The correct class is 1.
- b)The image TestCharacters/2/205.jpg is incorrectly classified to 1.  
The correct class is 2.
- c)The image TestCharacters/3/239.jpg is incorrectly classified to 2.  
The correct class is 3.
- d)The image TestCharacters/3/226.jpg is incorrectly classified to 1.  
The correct class is 3.

## Task 2

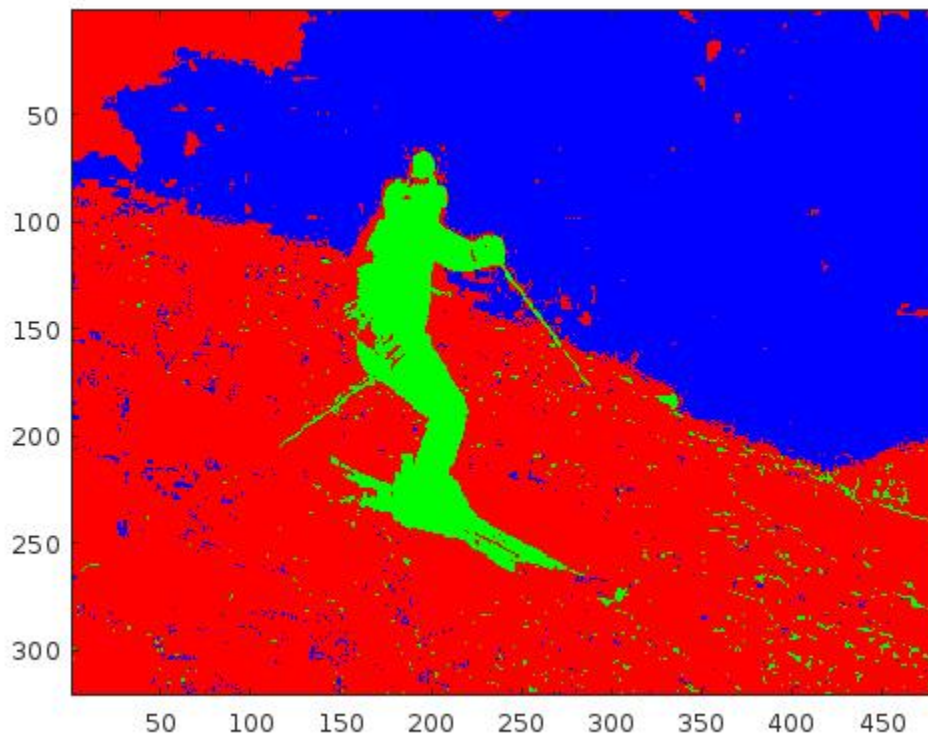
To run the code we need to run Code2.m.

The total time taken by code to run for 20 iterations is approximately 110 seconds.

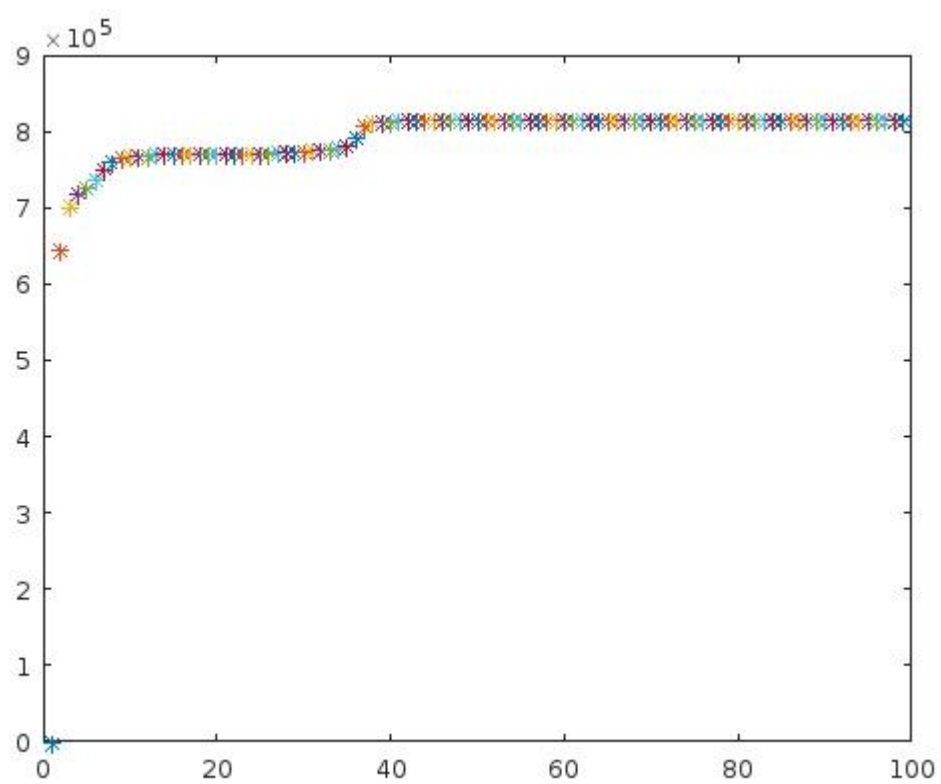
In this task we were given an image and we had to segment it into 3 coloured output image namely red,green,blue based on the initial values of means given to us. The concept of Gaussian Mixture Models (GMM) was used to cluster the points according to the updated means and covariances.

### Segmented Output

The output is with number of iterations being 100.



## Log Likelihood Convergence

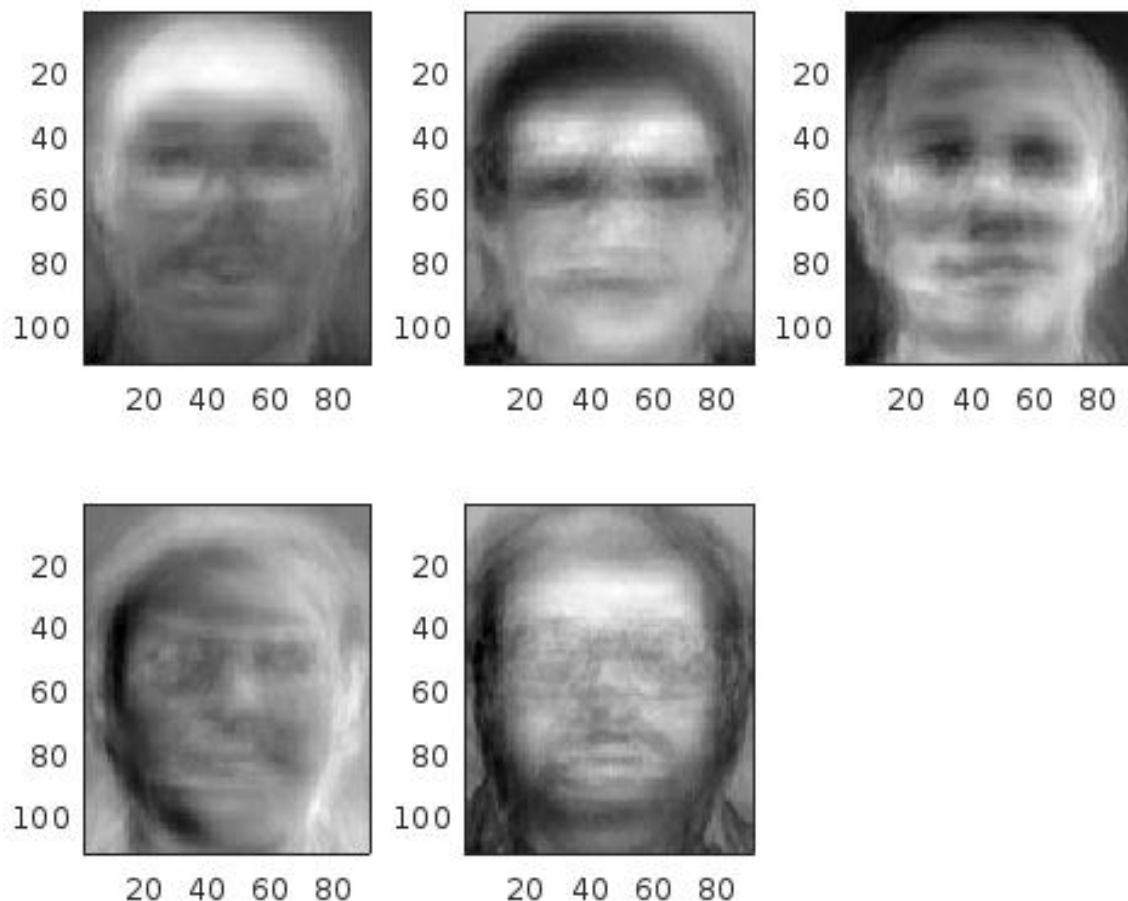


## Task 3

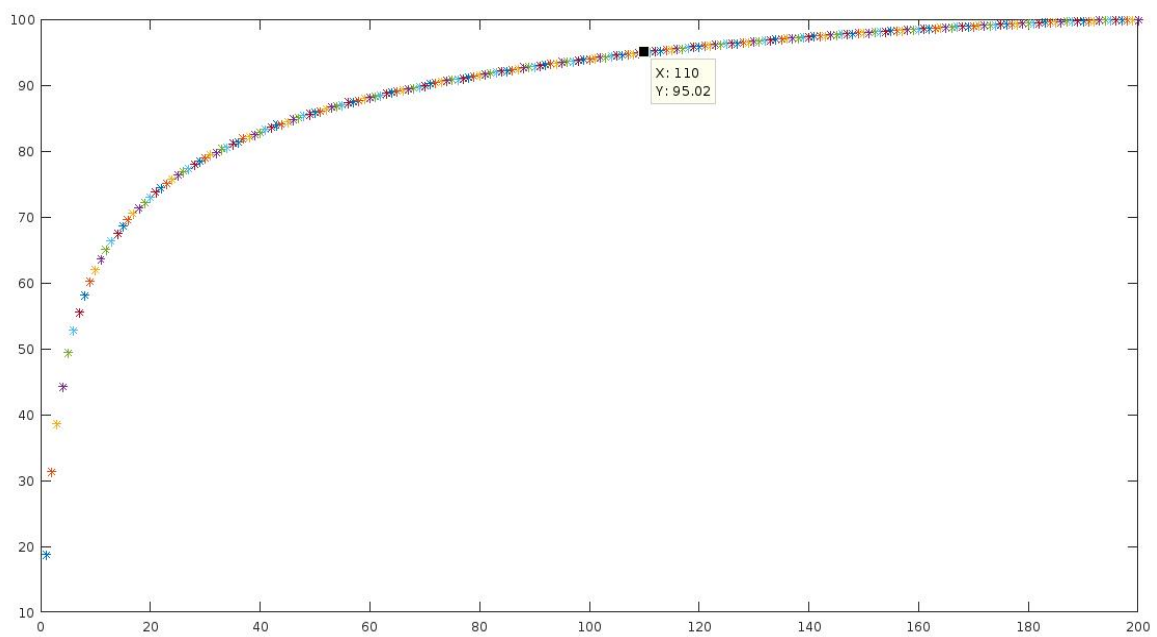
The total time taken by code to run is approximately 3 seconds.

In this task we used KL Transform to reduce the dimensions of the feature vectors using the eigen vectors of the covariance matrix. This was then used to reconstruct 2 input faces using the reduced dimension space. The images were successfully reconstructed with maximum clarity when all the eigen vectors were used. The reason being that 200 eigen vectors are the projection of the higher dimension into a lower dimension space. From the result we can see that image of face 2 is much clearer than image of face 1 as the projection of the second image had recovered more data in the lower dimension space than the image of face 1.

### The Eigenfaces of top 5 eigenvectors



**Graph depicting percentage of total variance retained in reduced space  
versus  
the number of dimensions**



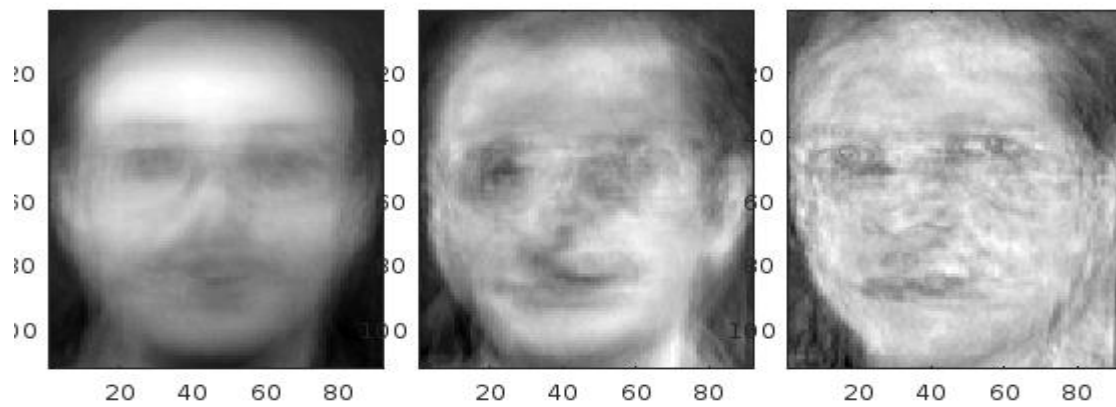
## Reconstructed Image of Face 1

The mean square in the following images are :-

i)  $1.7672e+03$

ii) 797.0230

iii) 333.6108





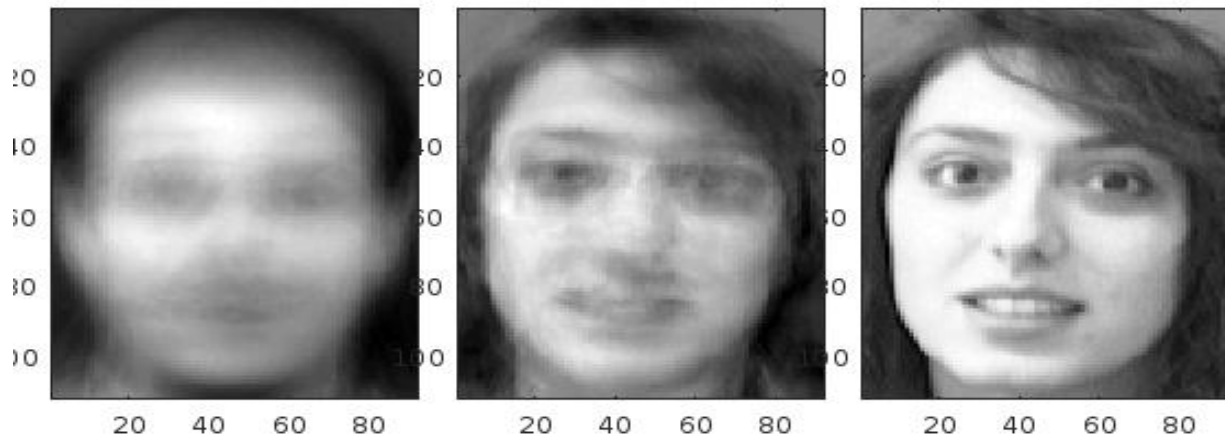
## Reconstructed Image of Face 2

The mean square in the following images are :-

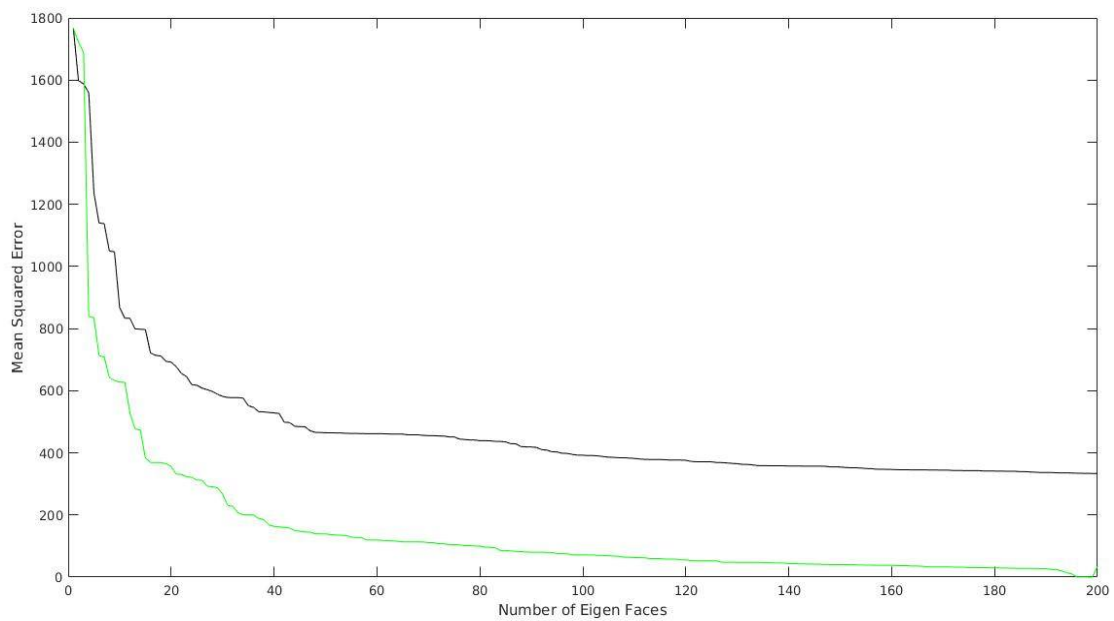
i)  $1.7666e+03$

ii) 384.1851

iii) 34.6363



## Mean Square Error for the Images

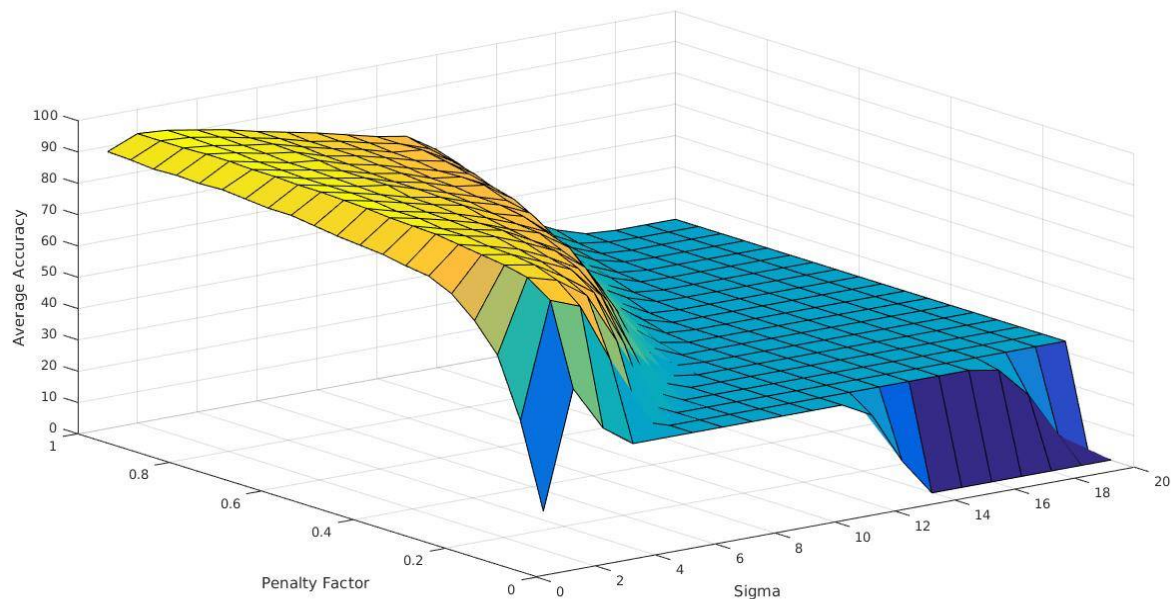


## Task 4

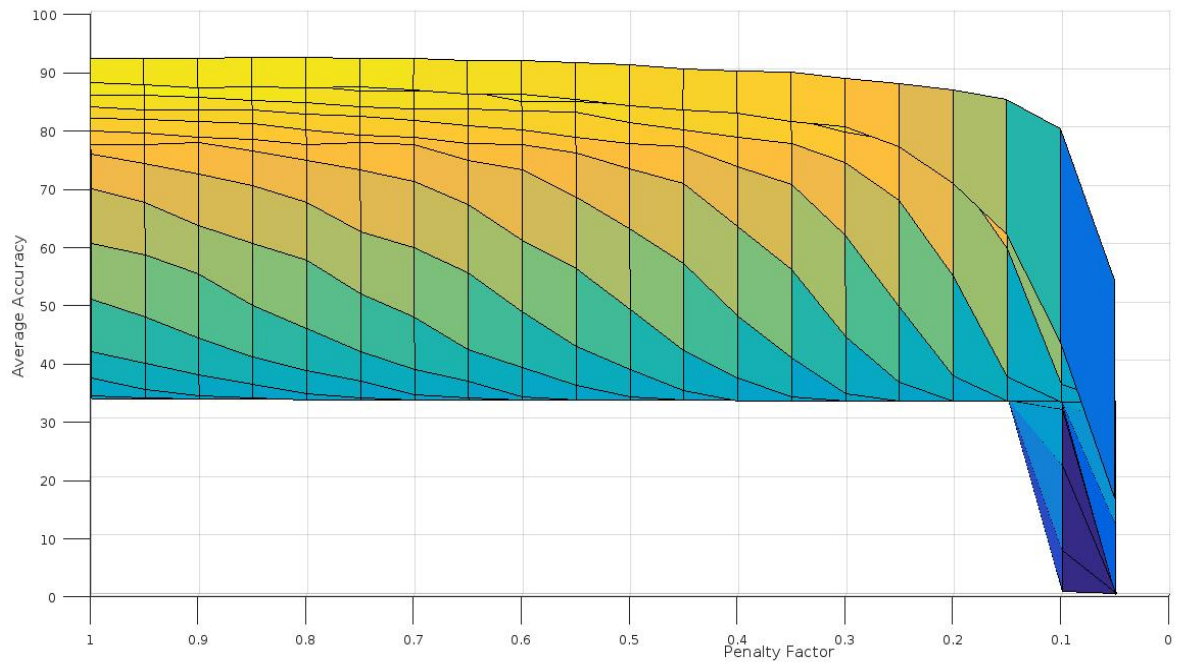
The total time taken by code to run is approximately 38 seconds.

In this task we had to build a SVM Classifier for 3 classes using Radial Basis Functions. This was done using the 'fitsvm' command in MATLAB. We did not use LibSVM as the accuracy of the results were unsatisfactory. We then changed the Penalty Factor and Precision values to see the changes in the Average Accuracy of the classifier. The results were plotted as shown below.

### The Average accuracy variation with Penalty Factor and Gamma



## Average Accuracy variation with Penalty Factor



## Average Accuracy variation with Precision

