ECE 763 Project 1 Report

Jie Hu

02/18/2022

Introduction

In this lab report, I build up four models to solve the image classification problem. I choose the FDDB dataset, which includes face annotations with ellipse to point out the position of face in each image. In this lab report, I use the RGB image. To crop the face out of the image, I use bounding box of the ellipse and rescale the patch to 20x20 with cv2 package. For non-face image, I randomly crop the patch with size 20x20 that is out of the range of the position of the bounding box. The train dataset includes 1000 images with face and 1000 images without face. The test dataset contains 100 images with face and 100 images without face.

For the codes, *image_patch_process.ipynb* includes the data process to crop and resize images. *mode.ipynb* includes all the codes and training processes of four models used in this project. *dataset* folder contains the original images and trimmed face/no-face images and I packed them as a whole dataset (see .h5 files). *model* folder includes the results of each model.

Model 1: Single Gaussian Model

I visualized results for trained model as follows:

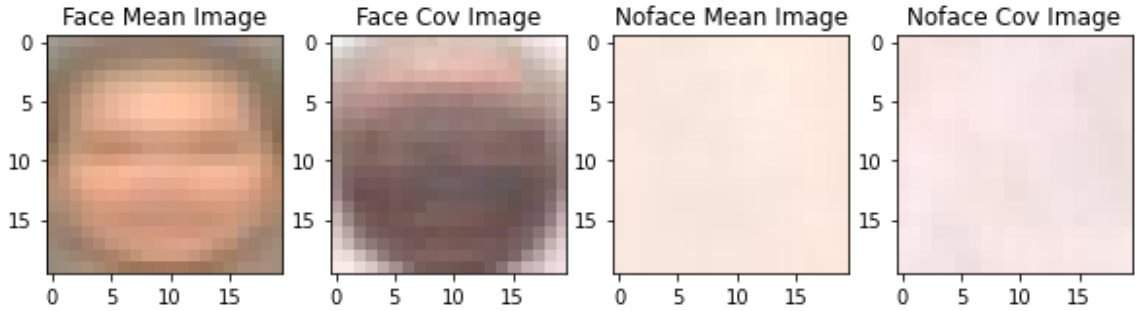

Figure 1. Visualization of Single Gaussian Model. The first figure is the mean for face image and the second figure is the covariance matrix for face image. The third figure and the fourth figure are for non-face image, respectively.

Note that to visualize covariance matrix, I assume each pixel are independent to each other and only pick the main diagonal of the covariance matrix to simplify the process. Table I gives the performance of Single Gaussian Model.

False Positive Rate	0.3
False Negative Rate	0.14
Misclassification Rate	0.22

Table I: Performance of Single Gaussian Model.

The ROC curve is attached in Figure 2.

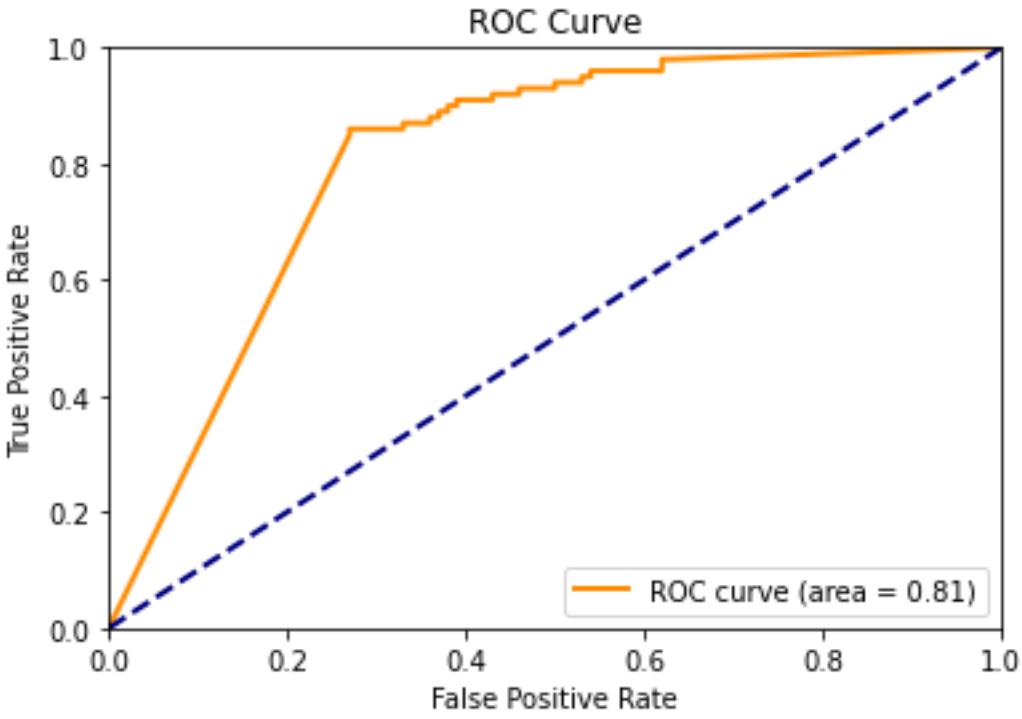

Figure 2. ROC curve of Single Gaussian Model

Model 2: Mixture of Gaussian Model

Starting from this model (model 2, model 3 and model 4), I use PCA from the scikit package to reduce the dimension of the data point from 1200 (because of the image size 20x20x3) to 100. Note that I do PCA on face images and no-face images separately. After training the model, I reconstruct the image from the vector with size 100x1 back to 1200x1 and resize it to 20x20x3. By doing so, I can use cv2 package to output the reconstructed images.

I considered 5 components in the Mixture of Gaussian Model and visualized results for trained model as follows:

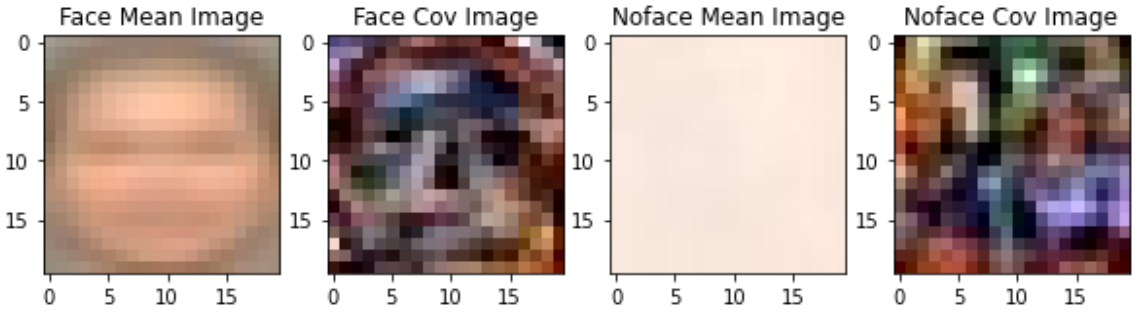

Figure 3. Visualization of Mixture of Gaussian Model, considering 5 components.

The performance of Mixture of Gaussian Model and ROC curve are given below.

False Positive Rate	0.44
False Negative Rate	0.1

Misclassification Rate	0.27

Table II. Performance of Mixture of Gaussian Model.

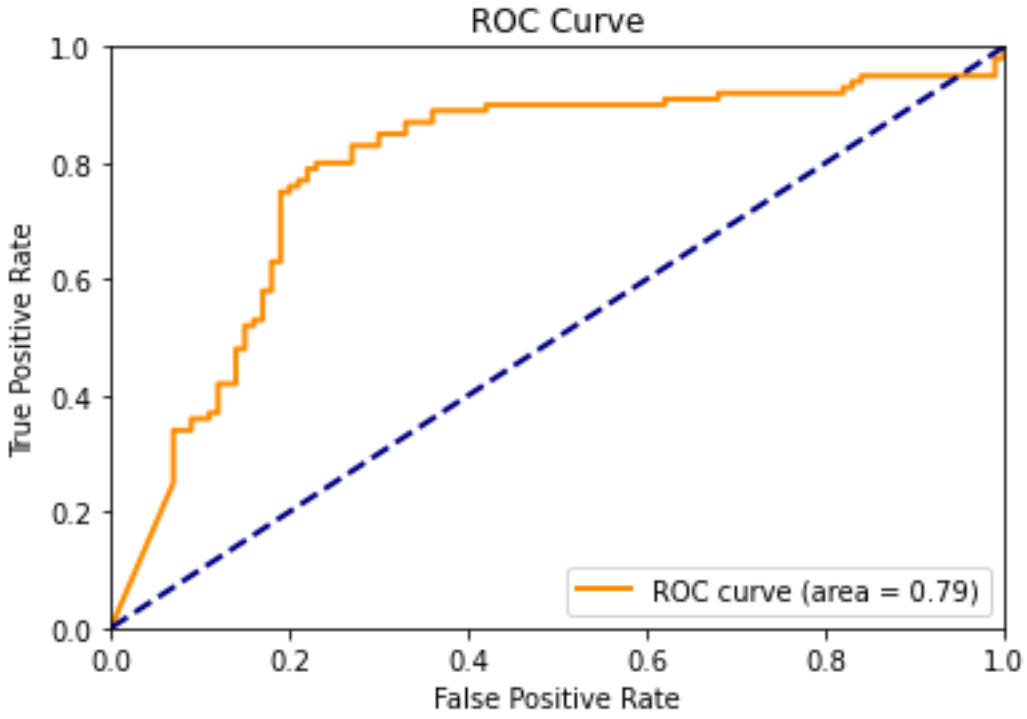

Figure 4. ROC curve of Mixture of Gaussian Model.

Model 3: T-distribution Model

I visualized the images for trained T-distribution Model as follows:

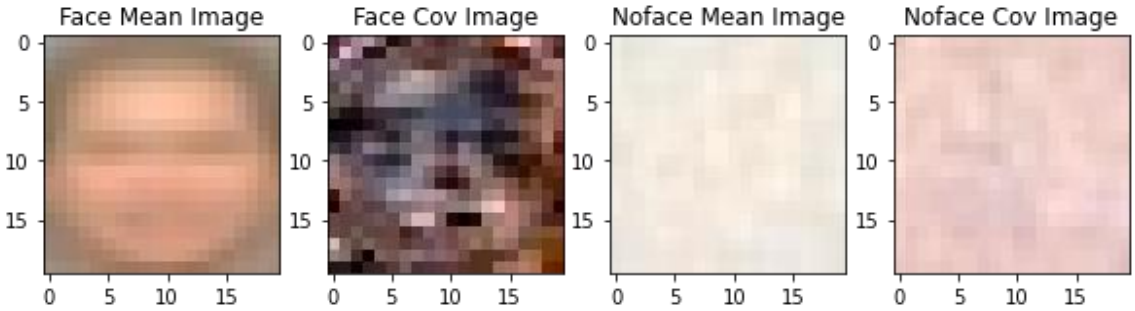

Figure 5. Visualization of T-distribution Model.

The performance of T-distribution Model and ROC curve are given below.

False Positive Rate	0.0
False Negative Rate	0.1
Misclassification Rate	0.5

Table III. Performance of T-distribution Model.

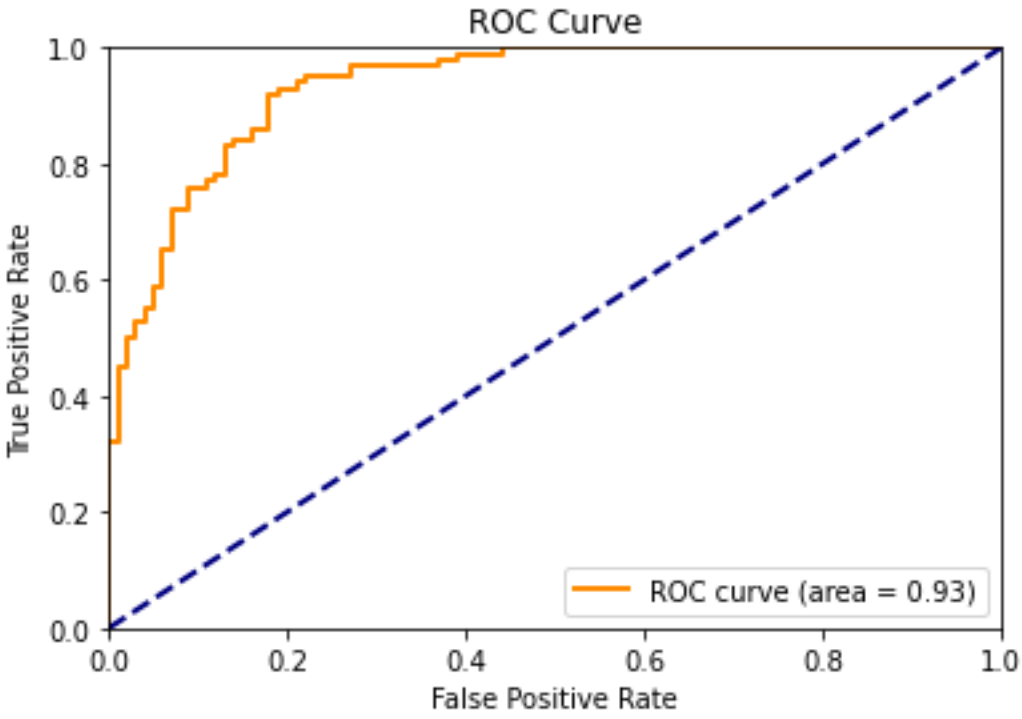

Figure 6. ROC curve of T-distribution Model.

Model 4: FactorAnalyzer Model

I set the number of factors $k = 50$ for FactorAnalyzer Model. The visualized mean and covariance matrix images for trained FactorAnalyzer Model are shown in Figure 7.

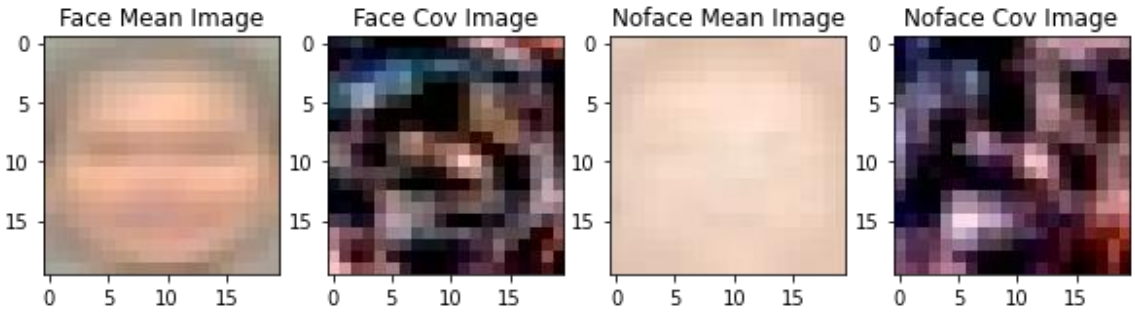

Figure 7. Visualization of FactorAnalyzer Model.

The performance of FactorAnalyzer Model and ROC curve are given below.

False Positive Rate	0.36
False Negative Rate	0.04
Misclassification Rate	0.2

Table IV. Performance of FactorAnalyzer Model.

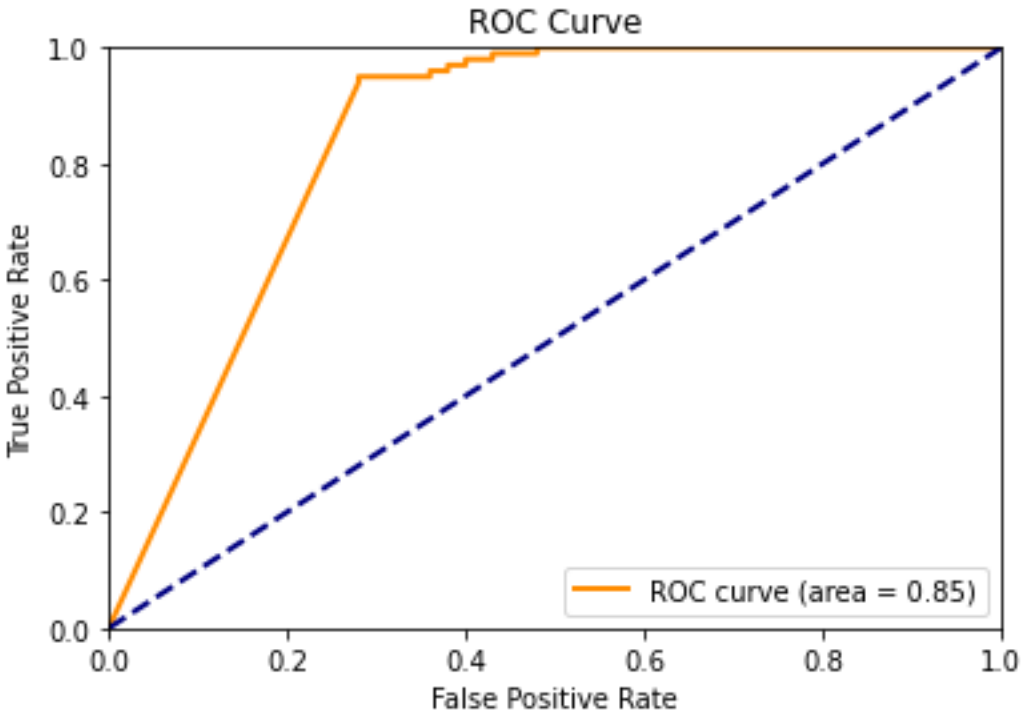

Figure 8. ROC curve of FactorAnalyzer Model.