Assignment 3

Daksh Gupta

A20351161

Report

Problem Statement:

Design, implement and evaluate a face recognition system

Gallery and Probe contents:

The gallery contains 115 subjects with multiple images per subject.

The probe contains 88 subjects with only 1 image per subject.

The gallery and probes have 58 subjects in common. The images have been converted to JPEG and a list of images is present in ‘gallery’ and ‘probe’ folders respectively.

Theory:

To build a face recognition system, the first step is to detect the face in an input image. After detecting the face, the face image needs to be enhanced to make the unique features of the face more visible and enhanced. Histogram equalization and gamma correction are pretty good techniques for this enhancement.

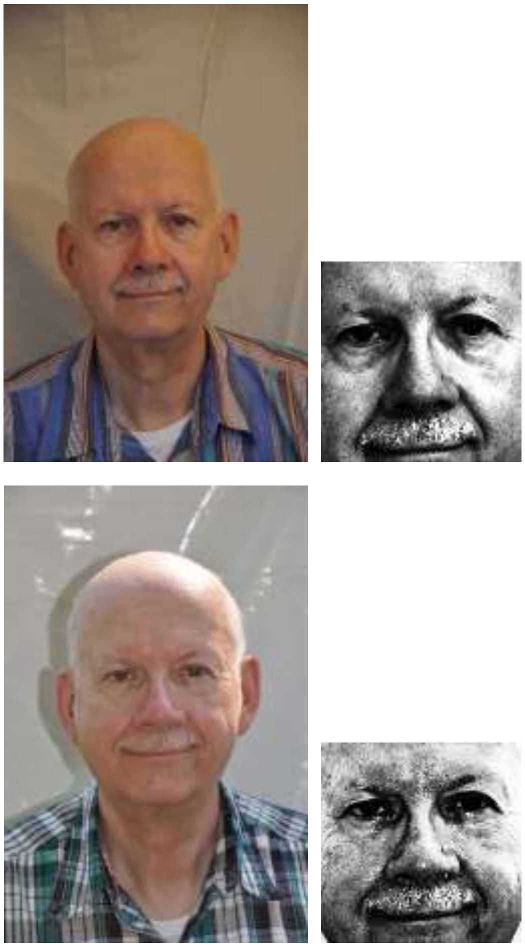
After clear face images are available, feature detection using SIFT or SURF can be used to get good quality features from the face images. These features can act as the biometric template for that face image.

The scores retrieved after comparison of these features is like a distance score, i.e., the lower the score, the better match it is. These scores can be used to evaluate the recognition system.

Implementation Details:

I used OpenCV 3.1 for this assignment. I did not use OpenCV’s predefined face recognizers, instead I manually created one using SURF features.

For getting the face images, I used the CascadeClassifier using haar cascade for the frontal face. For each raw image in the gallery and probe, I saved the face image in the same directory. The conversion results are as follows:



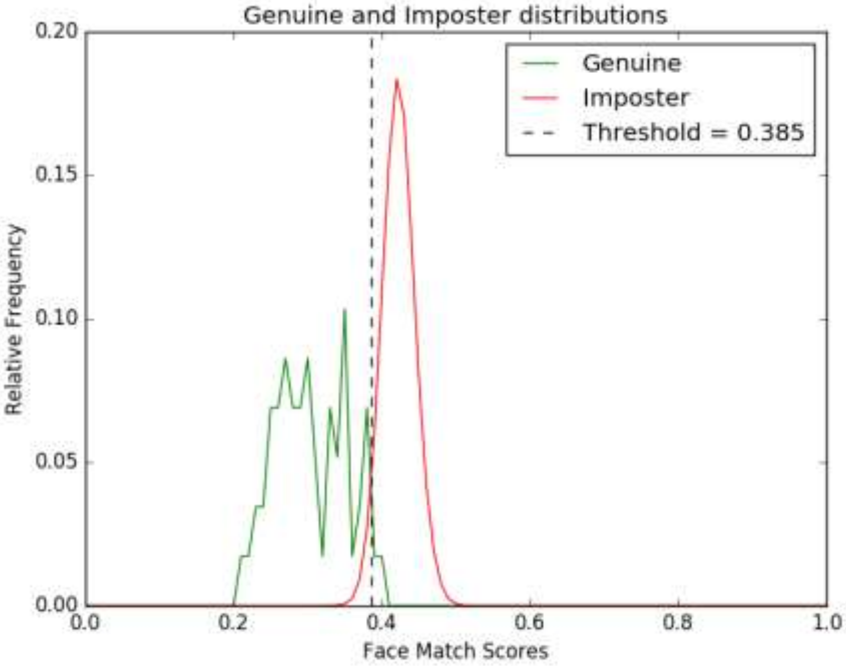
The raw image is processed for face detection and once the face is cropped from the image, gamma correction of 0.3 is applied to generate the visible results. This processing reduces the illumination differences for various conditions to a minimum, and helps the SURF feature detector to get good features. I have used the SURF features with a Hessian Threshold of 500.

For feature comparison I used the Brute Force Matcher of OpenCV. To get better results, when two image features are compared, I selected the best 500 feature matches and selected the mean distance of all of these 500 matches. This gave me match scores between 0 and 1, where a good match is closer to 0.

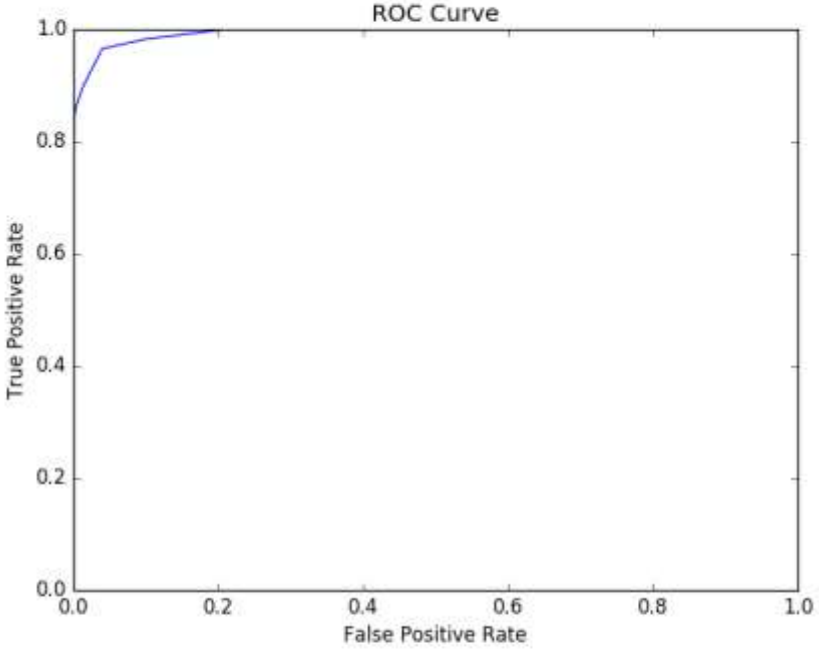
The comparison score matrix for all images in the gallery and all images in the probe is stored on the local disk with the name ‘dists.mat’ and the label matches with the name ‘matches.mat’. The rows of these matrices represent the number of images in the gallery and the columns represent the number of images in the probe. These files can be used to generate the performance evaluation curves.

Results:

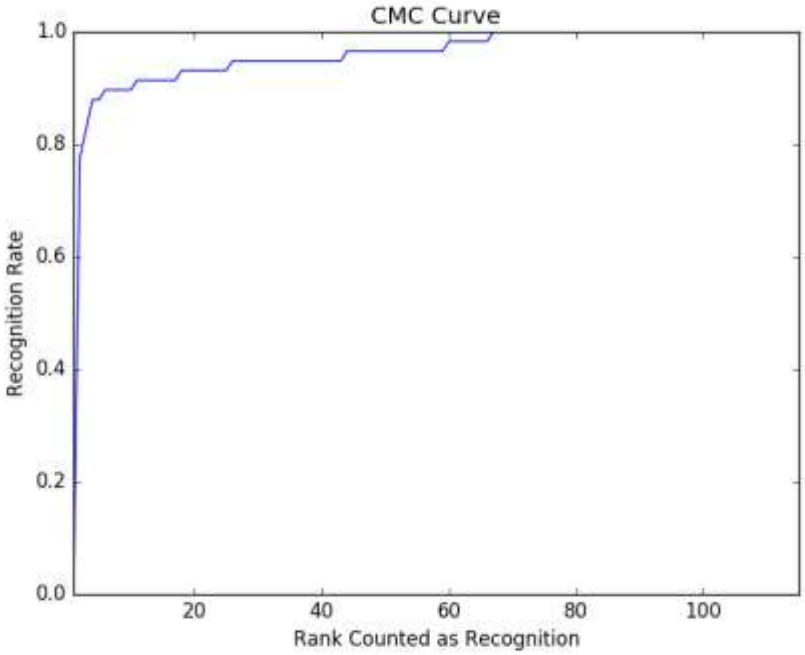
The created system had a pretty good performance. The plots are below:



The genuine and imposter distributions have a very less overlap between them. A threshold of 0.3849 was found to be ideal for most general purpose applications. This can be adjusted as per the security vs. ease tradeoff.



The ROC curve is very close to the ideal curve, suggesting that the performance of this system is very good.



Here is the CMC curve. The ranks range from 1-115 since that is the number of subjects enrolled in the gallery. The recognition rate is over 77% at rank 2. And reaches a 100% at the rank of 67. This performance was suggested by the previous curves as well since there are some False Negatives and False Positives.