Assignment 2

Daksh Gupta
A20351161

Report

Problem statement:

Design and asses an Iris Recognition system with LG2200_2008 dataset as the Gallery and LG2200_2010 and LG4000_2010 datasets as two different probes.

Gallery and Probe contents:

Gallery: Contains every eye pair from all of the 88 identities in LG2200_2008 dataset.

Probe: Contains the first eye pair from all of the 220 identities in LG2200_2010 dataset as one probe and from LG4000_2010 as another probe.

Comparison Theory:

I have designed my recognition system to consider pair of eyes rather one single eye to improve the accuracy.

My belief is that if left and right eye both have a good match, then the average hamming distance of the two should quite small. And if it's not a match then the average should be high.

John Daugman introduced a formula to compensate for the number of bits compared to re-normalize Fractional Hamming Distance:

$$HD_{norm} = 0.5 - (0.5 - HD_{raw}).\sqrt{\frac{n}{911}}$$

where, HD_{raw} is Fractional Hamming Distance
n is the Total number of bits compared

Implementation Details:

Overview:

- A list of valid pairs is sent fed to MATLAB script to segment iris from the eye images.
- Watchlist, probe templates and masks are created separately.
- Hamming distances between probe1 and watchlist, and probe2 and watchlist are calculated for left and right eye images.
- Left and right eye FHDs are separated and average is taken
- Genuine & Imposter distributions and ROC curves for the results are formed.

Segmentation:

To create a list of valid eye pairs, I used python to create text files, 'images.txt' and 'labels.txt', for each dataset.

I organized my folders as follows:

```
diagnostics/
LG2200_2008/
LG2200_2010/
LG4000_2010/
iriscode/
```

The dataset folders have a structure as follows:

```
../
02463/
04233/
04252/
.
.
images.txt
labels.txt
```

I did this for ease of access of the images.

Images.txt has the following content:

```
../LG2200_2008/02463/02463d1914.tiff
../LG2200_2008/02463/02463d1920.tiff
../LG2200_2008/02463/02463d1915.tiff
../LG2200_2008/02463/02463d1921.tiff
../LG2200_2008/02463/02463d1916.tiff
.
.
```

The odd rows of this file contain only paths to the right eye images and even rows contain paths to the left eye images. So line1 and line2 correspond to a pair. These pairs were found by analyzing the txt files in each folder of the dataset.

Labels.txt has the following contents:

```
02463
02463
02463
02463
02463
.
.
```

The folder ‘iriscode’ has all the MATLAB and python scripts.

These files are used to segment iris from the images and store the hough parameters on the local disk for every processed image using Libor Masek’s code in one go.

Changes in ‘segmentiris.m’:

- thresholding set to < 25 instead of < 100

Encoding:

The default implementation generates a template and mask of 20*480 bits for every iris image. The angular radius is set to 240. The width of the template is determined by 240*2*nscales. 'nscales' is set to 1 by default.

I have tweaked the original code of Libor Masek to work well for the provided datasets.

The tweaks are as follows for 'createiristemplate.m':

- nscales = 2
- minWaveLength = 12
- mult = 1
- sigmaOnF = .3

Comparison:

The function 'gethammingdistance' takes as arguments two templates and masks, and a scale parameter which was used to encode the iris templates. This code already performs rotation to find out the most accurate hamming distance between two templates.

The default is to find the best match between 8 circular shifts to the left and right. I tweaked this implantation to shift 16 shifts in each direction.

I introduced the re-normalization formula in this code to get better hamming distances.

```
hd1=0.5-(0.5-hd1)*realsqrt(totalbits/ceil((20*2*240*scales)/2.25));
```

Here the $\frac{n}{911}$ factor has been replaced to generate the correct results. So, in general, 2048 is the maximum number of common bits possible between two templates. The total number of bits is much larger in the current encoding and is dependent on 'nscales', so to find the correct denominator I used the above formula to give me a similar factor (since, 2048/911 $\approx$ 2.25 and 2048).

I formed a Matrix with rows having the size of total images in the gallery, i.e., 2*no. of pairs, and columns with total number of images in that probe. The contents of this matrix are the FHDs between the corresponding iris templates. I made sure that left and right eye images are never compared with each other. There are only left-left and right-right comparisons.

Forming the distributions and ROC curve:

One the comparison is complete left-left and right-right comparisons were separated from the comparison matrix.

And average is taken for these comparisons to get the pair comparison scores.

The gallery has multiple pairs for the same identity while the probes have only 1 pair per identity.

To find the data for the genuine distribution, only the best, or minimum, score from a single identity match was considered.

The imposter distribution consists of all other scores other than the genuine scores.

Results:

Tweaking the default encoding parameters generated some better results but the overall performance is not good. The accuracy of the systems is not what is expected out of an iris recognition system.

Out of all 88 subjects in the Gallery and 220 subjects in each of the probes, only 21 are the same, rest all are different people.

For LG2200_2010 dataset as probe:

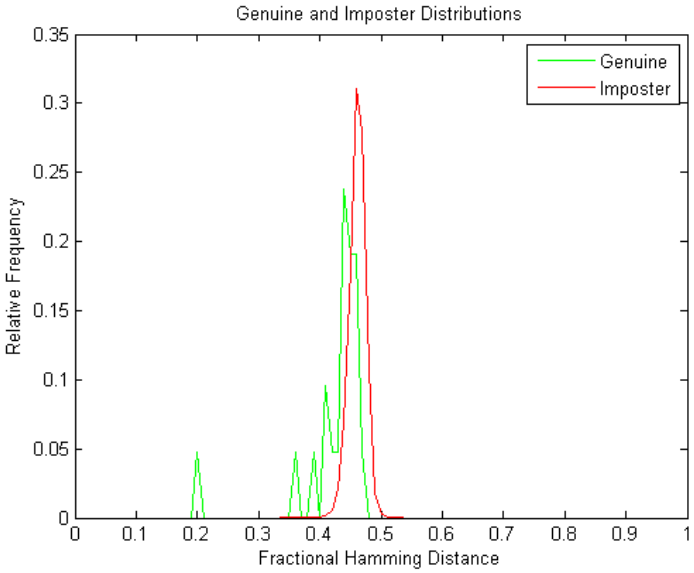

Figure 1 LG2200_2010 as probe

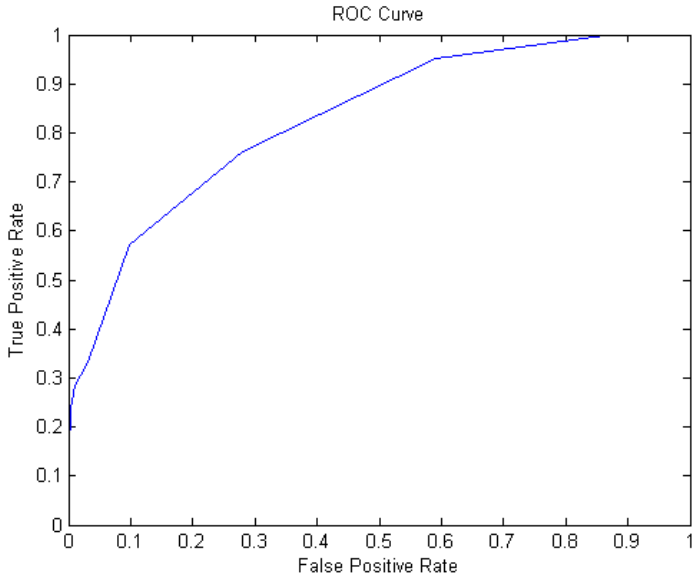

Figure 2 LG2200_2010 as probe

As a performance measuring metric, the system gave a TPR of about 55.1% at a FPR of 10% at a threshold of 0.4401

For LG4000_2010 dataset as probe:

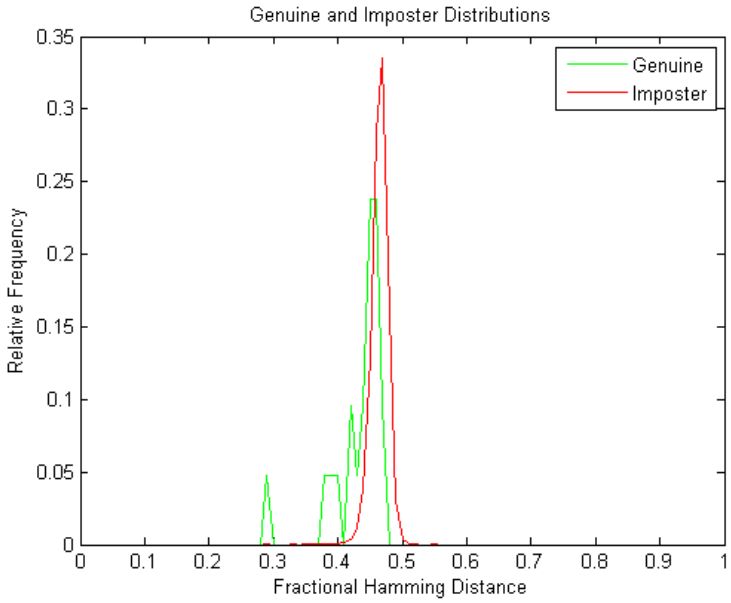

Figure 3 LG4000_2010 as probe

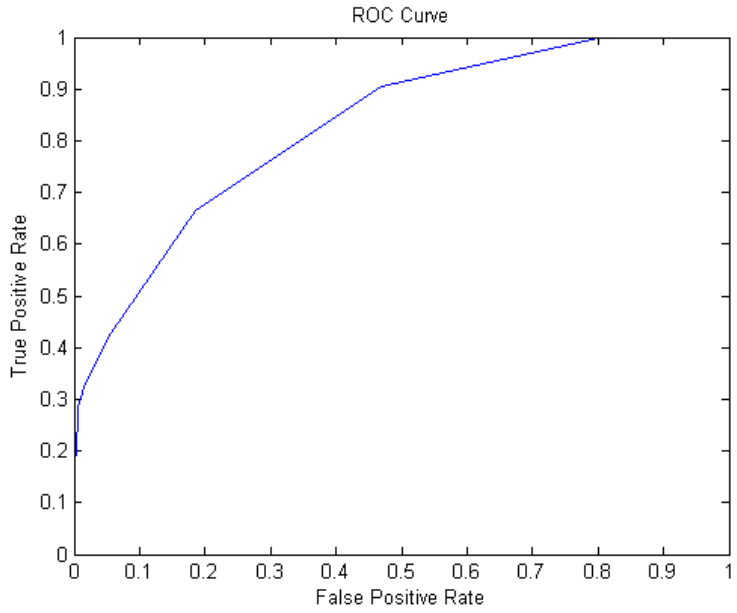

Figure 4 LG4000_2010 as probe

As a performance measuring metric, the system gave a TPR of about 55.31% at a FPR of 10% at a threshold of 0.4446

The performance metrics are not exact and were calculated after fitting a 4 degree polynomial to the plots.

The Genuine and Imposter distributions shown above are exact. Use of binomial distribution to approximate the distributions was avoided due to visual incorrectness of the plots, as shown below:

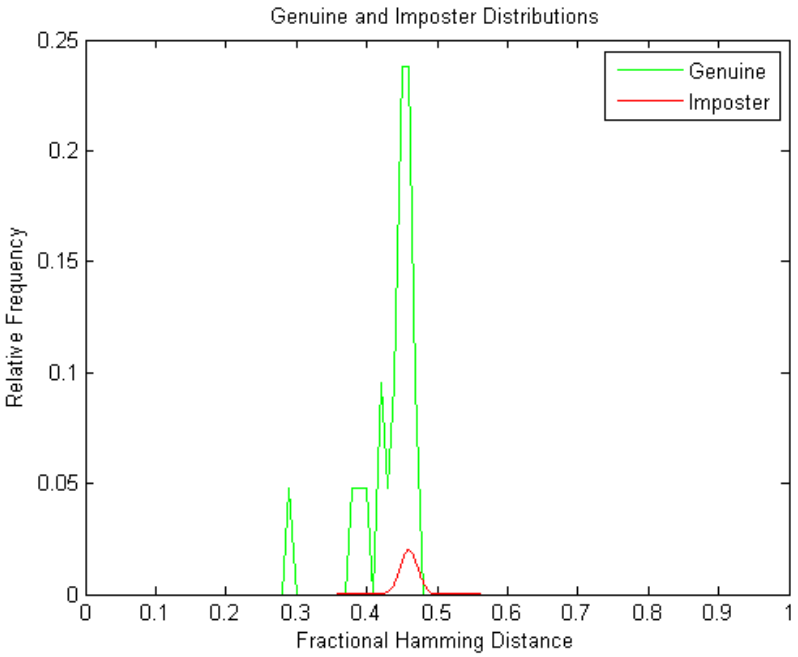

Figure 5 Only Imposter Distribution represented using Binomial PDF

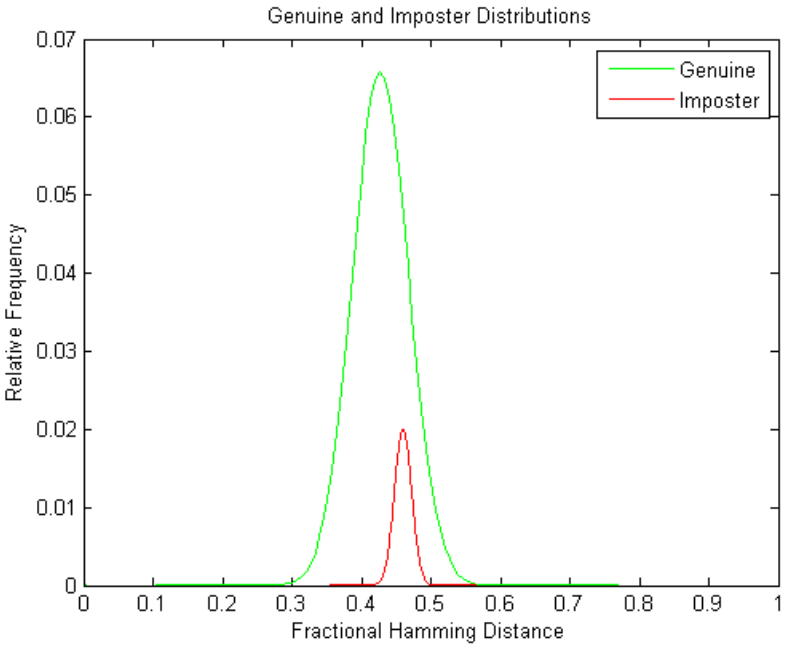

Figure 6 Genuine & Imposter Distributions represented using Binomial PDF