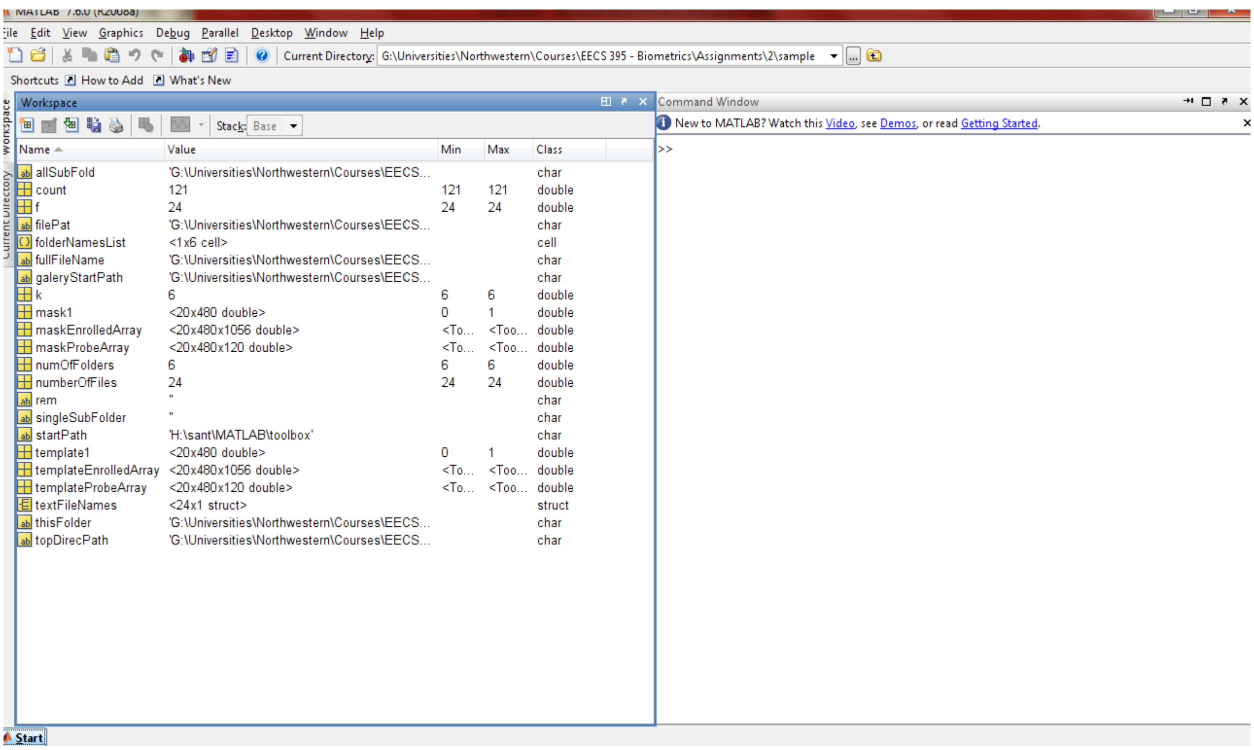
EECS 495 Biometrics
Assignment 2: Iris Recognition System
Karan Shah, Tina Shin, Santhosh Subramanian

## I. Design Analysis

- Enrolled data from the gallery (LG2200-2008-03-11_13)
- Probe 1 and Probe 2 are set of images of two different data sets (LG4000 and LG2200) taken from the year 2010-04-27_29
- Used matlab code to get template and mask values of enrolled images.
- Formed four 3D (m*n*p) arrays from the given data. Where m & n are number of rows and columns for template and mask of each image; p is total number of images in all folders. For Probe, only 10 folders were considered.
    - templateEnrolledArray (20*480*1056)
    - maskEnrolledArray (20 * 480 * 1056)
    - templateProbeArray (20 * 480 * 120)
    - maskProbeArray (20 * 480 * 120)




- We compared each image in the probe with all the images in the enrolled database. Then hamming distance for each image is calculated to get the smallest hamming distance value.
- The above step of generating hamming distance is repeated for the rest of the images in the probe.

- For this assignment, we considered 10 folders from the gallery, 10 folders from Probe1 (LG 4000) and 10 folders from probe 2 (LG2200). Below are the folders that we selected for getting hamming distance, distribution and roc curves.

  **Gallery:**
  - 02463
  - 04233
  - 04252
  - 04261
  - 04267
  - 04327
  - 04385
  - 04394
  - 04397
  - 04470

  **Probe1(LG4000):**
  - 02463
  - 04233
  - 04470
  - 0515
  - 05268
  - 05301
  - 05393
  - 05513
  - 05555
  - 05616

  **Probe 2(LG2200):**
  - 02463
  - 04233
  - 04327
  - 04385
  - 05669
  - 05733
  - 05752
  - 05766
  - 05774
  - 05805

- We used Libor Maesk's open source iris recognition software to segment the image and find hamming distance.

## II. Results

- Segmented image and its conversion to polar image:

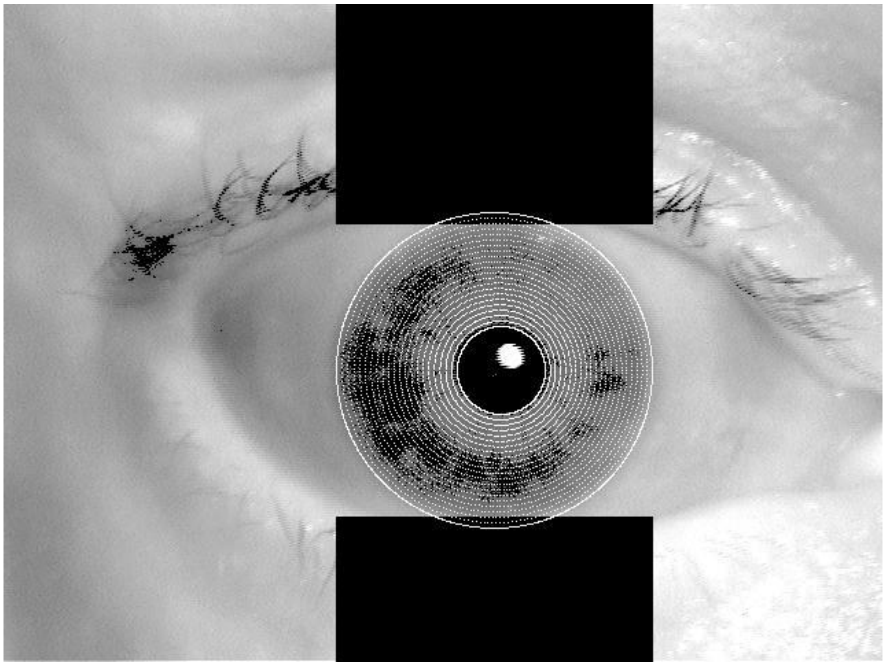


Fig 1: Finding the iris

Fig 2: Polar image

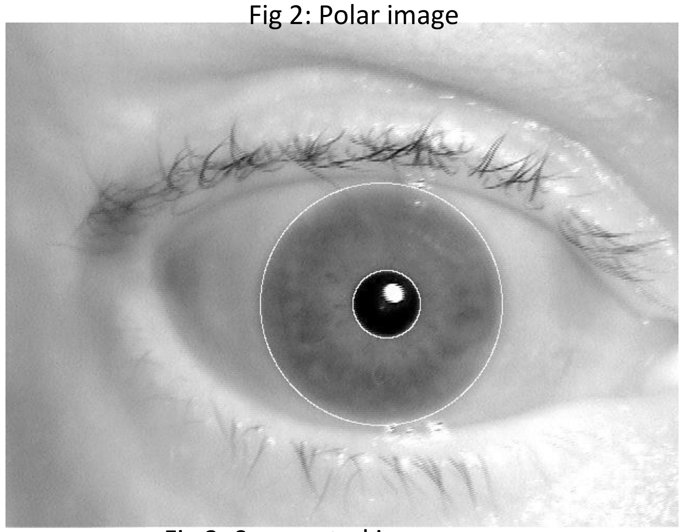


Fig 3: Segmented image

- Hamming distance for 10 different subjects with enrolled images in the database.

**LG4000(Probe1) with LG2200(Gallery)**

| Hamming Distance between gallery and probe 1 | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | S1 | S2 | S3 | S4 | S5 | S6 | S7 | S8 | S9 | S10 |
| P1 | 0.408903 | 0.438021 | 0.378378 | 0.25 | 0.451574 | 0.478587 | 0.436333 | 0.23 | 0.445055 | 0.411765 |
| P2 | 0.353801 | 0.359155 | 0.294118 | 0.277778 | 0.358333 | 0.425355 | 0.344828 | 0.323529 | 0 | 0.303571 |
| P3 | 0.373457 | 0.354545 | 0.321429 | 0.3 | 0.235294 | 0.433333 | 0.166667 | 0.324561 | 0.2 | 0.377778 |
| P4 | 0.309211 | 0.423077 | 0.38141 | 0.208333 | 0.25 | 0.407095 | 0.294118 | 0.14 | 0.328221 | 0.290323 |
| P5 | 0.334656 | 0.462709 | 0.419301 | 0.388889 | 0.340164 | 0.396164 | 0.413793 | 0.233333 | 0.451389 | 0.166667 |
| P6 | 0.384615 | 0.380658 | 0.25 | 0.25 | 0.373606 | 0.396552 | 0.258929 | 0.125 | 0 | 0.357143 |
| P7 | 0.434483 | 0.358711 | 0.35159 | 0.368421 | 0.375 | 0.415985 | 0.348673 | 0.397606 | 0.295455 | 0.327731 |
| P8 | 0.418667 | 0.166667 | 0.205882 | 0.166667 | 0.208333 | 0.388889 | 0.235294 | 0.285714 | 0.25 | 0.333333 |
| P9 | 0.340909 | 0.291209 | 0.375 | 0.083333 | 0.245098 | 0.285714 | 0.333333 | 0 | 0.238095 | 0.256098 |
| P10 | 0.384106 | 0.37037 | 0.351632 | 0.285714 | 0.370219 | 0.42492 | 0.389706 | 0.230769 | 0 | 0.413377 |

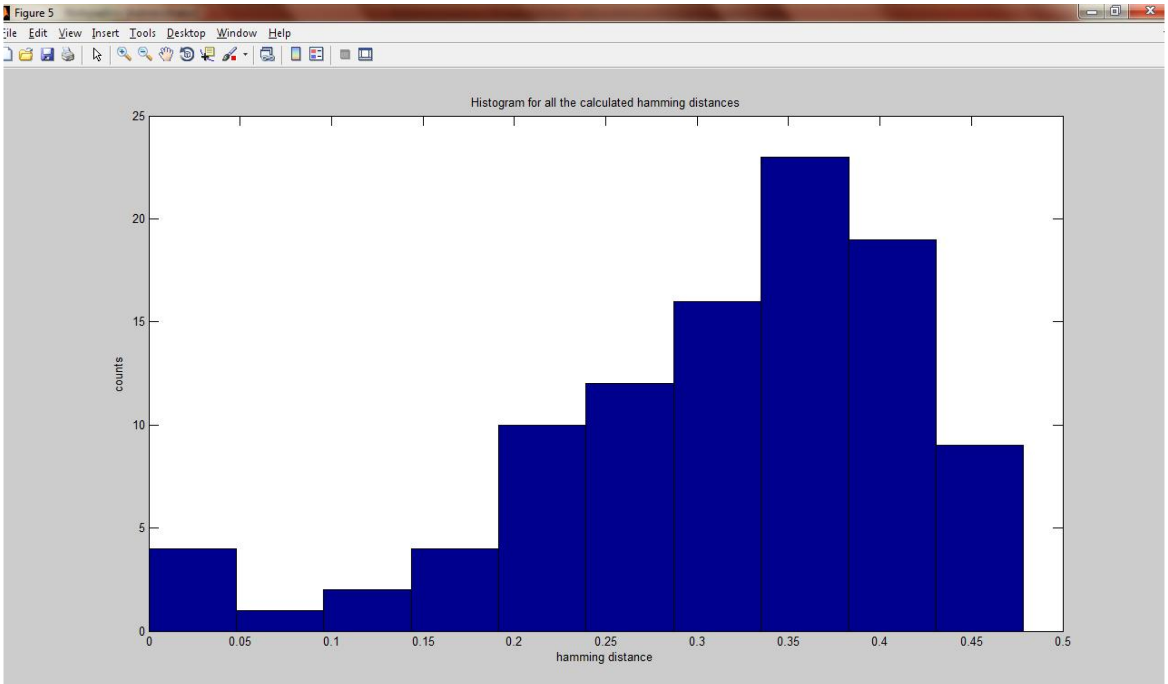
**LG2200(Probe2) with LG2200(Gallery)**

| Hamming Distance between gallery and probe 2 | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | S1 | S2 | S3 | S4 | S5 | S6 | S7 | S8 | S8 | S10 |
| P1 | 0.315728 | 0.404558 | 0.43241 | 0.34375 | 0.365385 | 0.414395 | 0.436333 | 0.346154 | 0.404891 | 0.37069 |
| P2 | 0.356643 | 0.344186 | 0.34106 | 0.205882 | 0.227273 | 0.371134 | 0.333333 | 0.160714 | 0.318182 | 0.356667 |
| P3 | 0.434532 | 0.440053 | 0.316667 | 0.329787 | 0.332335 | 0.396995 | 0.410465 | 0.295455 | 0.400262 | 0.385475 |
| P4 | 0.311111 | 0.400709 | 0.378906 | 0.2 | 0.3125 | 0.406893 | 0.355597 | 0.25 | 0.33871 | 0.242574 |
| P5 | 0.399554 | 0.407971 | 0.371429 | 0.3 | 0.415966 | 0.456117 | 0.410526 | 0.314815 | 0.444954 | 0.364286 |
| P6 | 0.301136 | 0.306122 | 0.25 | 0.185714 | 0.263158 | 0.19403 | 0.083333 | 0.066667 | 0.289256 | 0.227273 |
| P7 | 0.456705 | 0.464554 | 0.482036 | 0.462963 | 0.414474 | 0.473162 | 0.424658 | 0.410714 | 0.434055 | 0.433476 |
| P8 | 0.304878 | 0.359903 | 0.334158 | 0.40625 | 0.390909 | 0.353846 | 0.401639 | 0.25 | 0 | 0.355263 |
| P9 | 0.414802 | 0.382831 | 0.25 | 0.266667 | 0.443787 | 0.459101 | 0.425656 | 0.428571 | 0.462898 | 0.35034 |
| P10 | 0.408795 | 0.375758 | 0.297297 | 0.3 | 0.354331 | 0.406814 | 0.371795 | 0.368421 | 0.376923 | 0.375 |

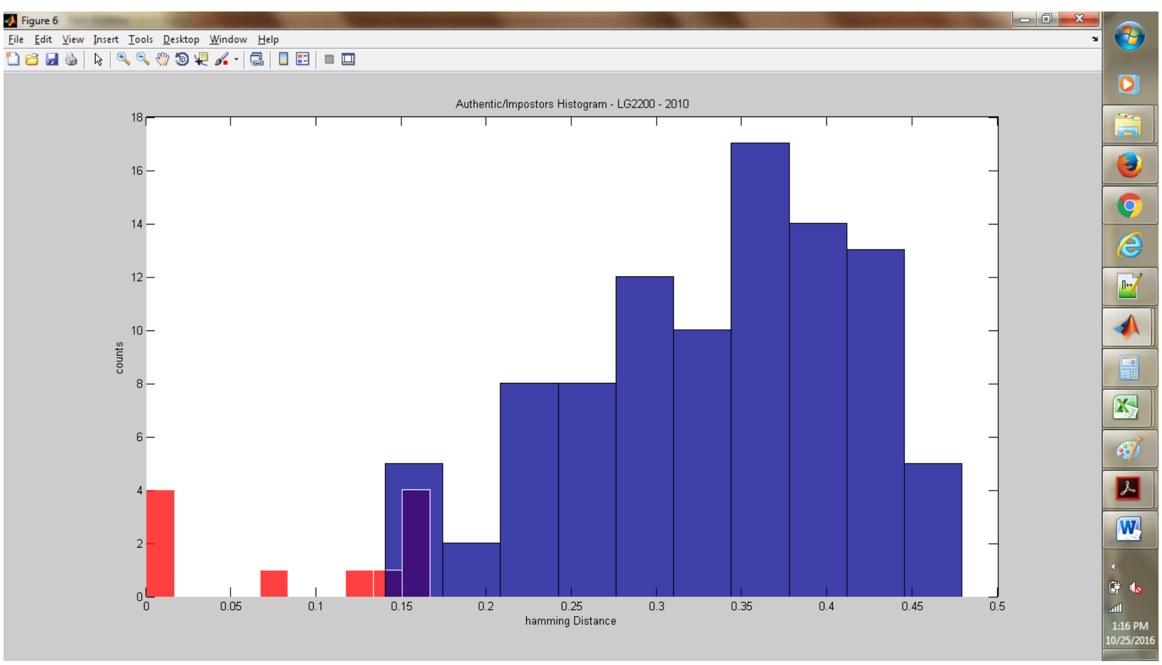
## III. Plots

## Histogram of Probe1 - LG 4000 - 2010

Histogram for 10 by 10 matrix calculated from 10 different subjects with 120 enrolled data.




Authentic/Impostor Histogram by fixing the threshold at 0.2



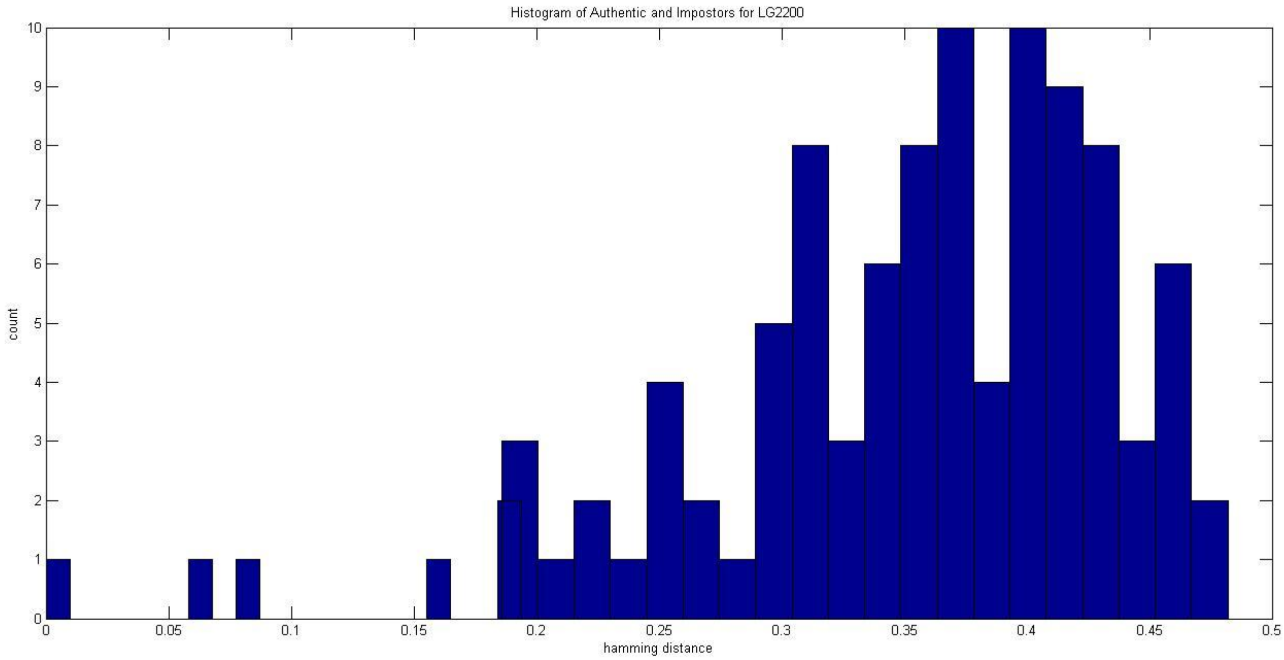
**MatlabCode for the Histogram for Probe1 and Probe2:**

```
probeout = a(:);
Figure(1)
histogram(authentic,20);
hold on
histogram(impostor,20)
ylabel('counts');
xlabel('hamming Distance');
title('Authentic/Impostors Histogram - LG2200 - 2010');
```

**Probe 2 – Authentic/Impostors Histogram of LG2200**

```
Figure(1)
hist(authentic,20);
hold on
hist(impostor,20)
ylabel('counts');
xlabel('hamming Distance');
title('Authentic/Impostors Histogram - LG2200 - 2010');
```

Authentic/Impostor Histogram by fixing the threshold at 0.2




- In probe 2 (LG2200), we can see the hamming distance of the images is coming as greater than 0.5 as we haven't selected any similar folders that are present in the gallery.

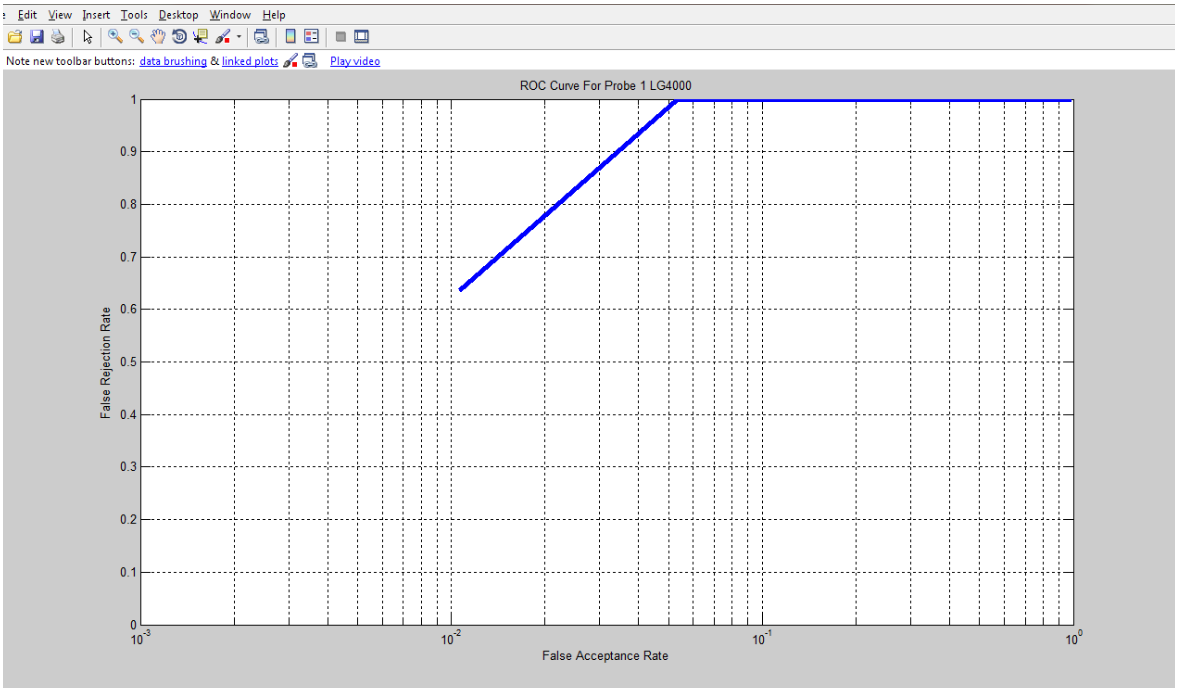
**ROC Curve** - Since we have the threshold at 0.2 and there are no overlapping between authentic and impostor distribution, we will get a perfect roc curve. For both the probes, we are generating a perfect system. We have used the following code for plotting ROC

[ver_rate, miss_rate, rates_and_threshs] = produce_ROC_PhD(auth, imp ,50);

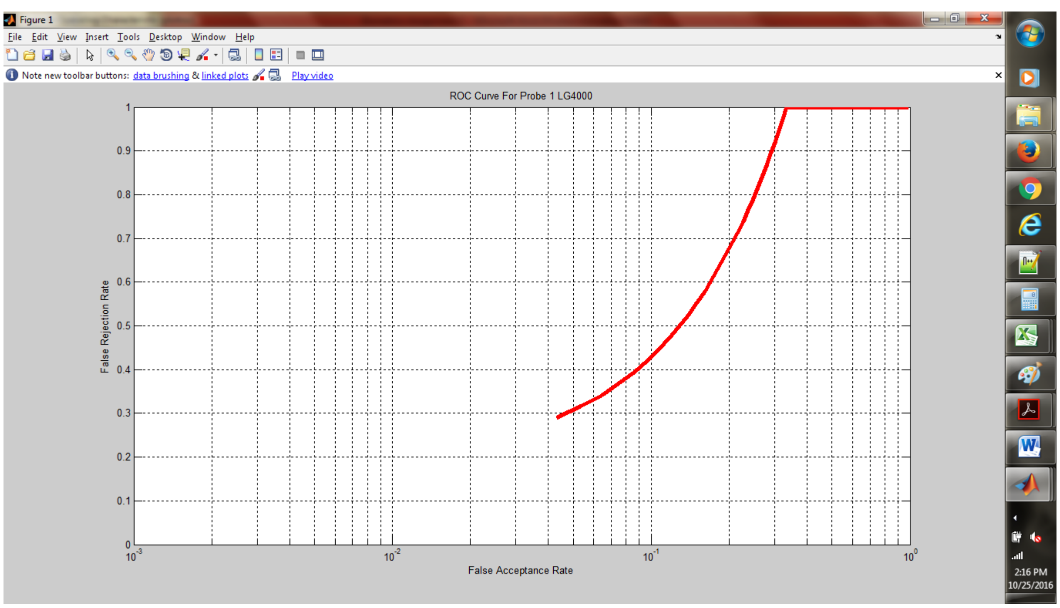
N=plot_ROC_PhD(ver_rate, miss_rate, 'b',4);

The plot of the ROC is shown below

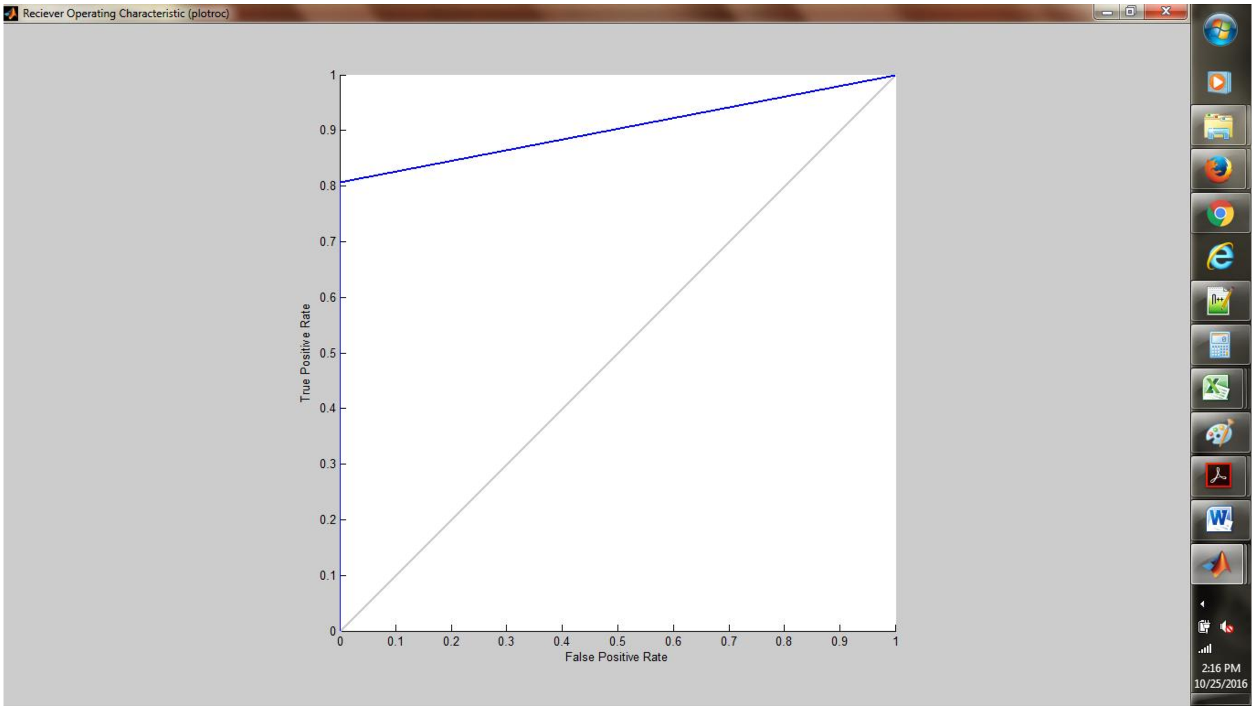
Threshold = 0.2




Threshold 0.3

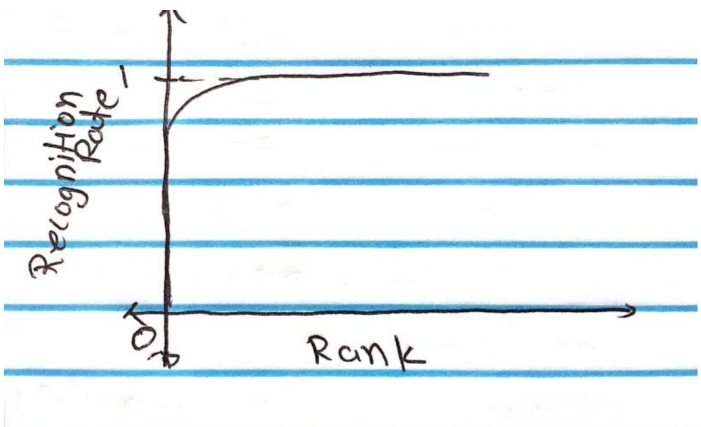


Plot for both thresholds




**CMC Curve** – We evaluate the performance of CMC curve using Rank and Recognition rate. If we find the hamming distance between the image and enrolled image of the same sample, we will get the correct score. The recognition rate is the ratio of number of correct images to the total number of images used.

Rank one recognition is the similarity of the samples that are close to one another in the same class.



## IV. Discussion

1) We could use clustering technique to find the authentic and impostor matrix
2) We could use more images that can be enrolled in the gallery and program the system to increase efficiency
3) Time complexity of the algorithm could be minimized by generating more database that are dynamic.

**References:**


1. https://www.mathworks.com/matlabcentral/fileexchange/35106-the-phd-face-recognition-toolbox/content/PhD_tool/plots/plot_ROC_PhD.m
2. https://www.mathworks.com/matlabcentral/fileexchange/35106-the-phd-face-recognition-toolbox/content/PhD_tool/plots/plot_ROC_PhD.m
3. https://www.mathworks.com/matlabcentral/fileexchange/35106-the-phd-face-recognition-toolbox/content/PhD_tool/eval/produce_CMC_PhD.m
4. Pattern Recognition by Vijay Kumar Mago & Nitin Bhatia
5. Libor Maesk's Iris Recognition System