

HW2

Members of group: A20377366 Ziyu Liu
A20383068 Silin Huang
A20298648 Shuxian Shen

This iris biometrics system is designed based on LG2200 2008 data as gallery, LG4000 and LG2200 2010 data as probes, using Matlab as the working environment.

Firstly, we use the function **createiristemplate.m** in Libor Masek's open source iris matching codes to create templates and masks for all the images of the gallery and probes. Then we use the function **gethammingdistance.m** to compute the match score of an enrolled template and a template for recognition.

The matlab script **GID_LG2200** and **GID_LG4000** is implemented to traverse all the images in gallery and probes to compute the match scores. Then the matlab script **plot** is used to plot the genuine and impostor distribution plots and the matlab script **plotROC** and **plotCMC** are used to plot the ROC curves, and CMC curves.

p1.mat is all the workspace data after we finished the recognition using LG2200. **p2.mat** is all the workspace data after we finished the recognition using LG4000.

Because the images in gallery and probes is too much to traverse, if we traverse all the images, the running time is too big. we just choose 4 images of the left eye and right eye for each subject in gallery and probe.

Here is the pseudo-code to get the genuine and impostor distribution.

```
for each image in gallery:
    [template1, mask1] = createiristemplate(image_gallery)
    for each image in probe:
        [template2, mask2] = createiristemplate(image_gallery)
        match_score = gethammingdistance(template1, mask1, template2, mask2, filter)
        if two images from the same eyes of the same subject:
            genuine.add(match_score)
        else
            imposter.add(match_score)
        end
    end
end
```

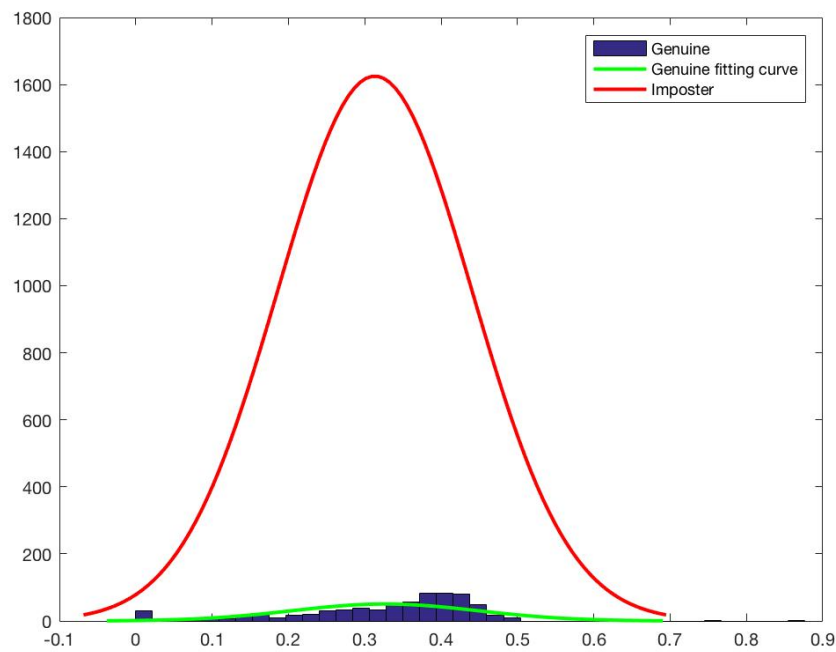
The scale of the data is too huge. In order to compute all the images, we think we need a super computer or using the Mapreduce.

Although we used the containers.Map to store all the templates and masks to reduce the running time and we just choose 4 eyes for each subject, for the LG2200, it takes about 4 hours to get the genuine and imposter and for the LG4000, it takes about 8 hours to get the genuine and imposter. All the images which are normalized are stored in GalleryTemplates and LG2200_probeTemplates and LG4000_probeTemplates.

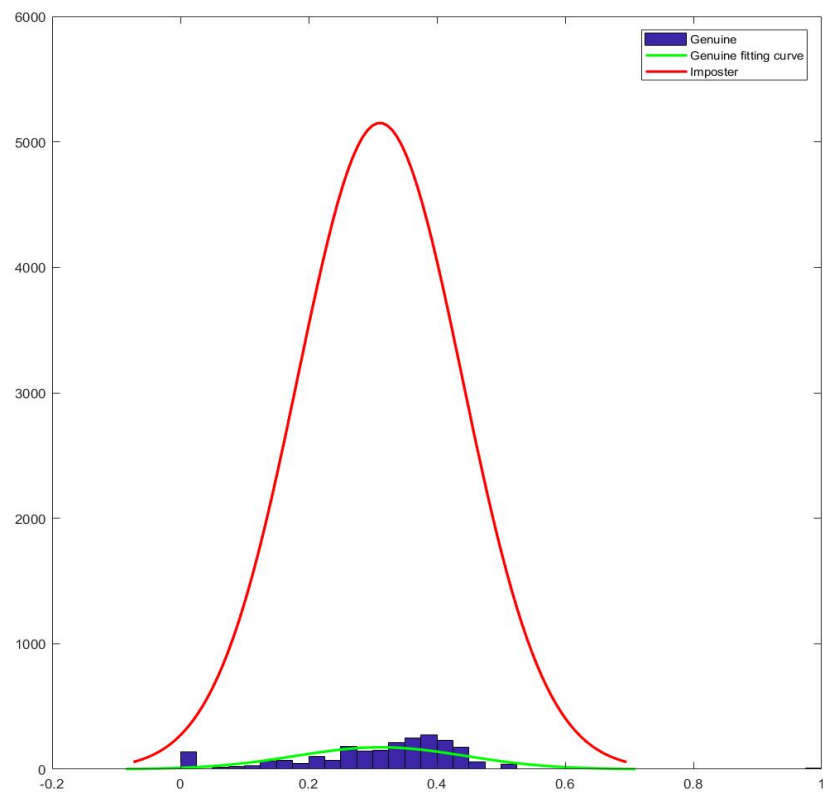
Because the number of images in the gallery and probe is too huge. It will take too much time to compute all the match score, we choose 2 images of the left eye and right eye for each subject in gallery and probe. As a result, the ROC and the genuine and impostor distribution are not like what are discussed in the class because the genuine is much fewer than the imposter. the genuine and impostor distribution plots are like below.

genuine and impostor distribution plots

for LG2200



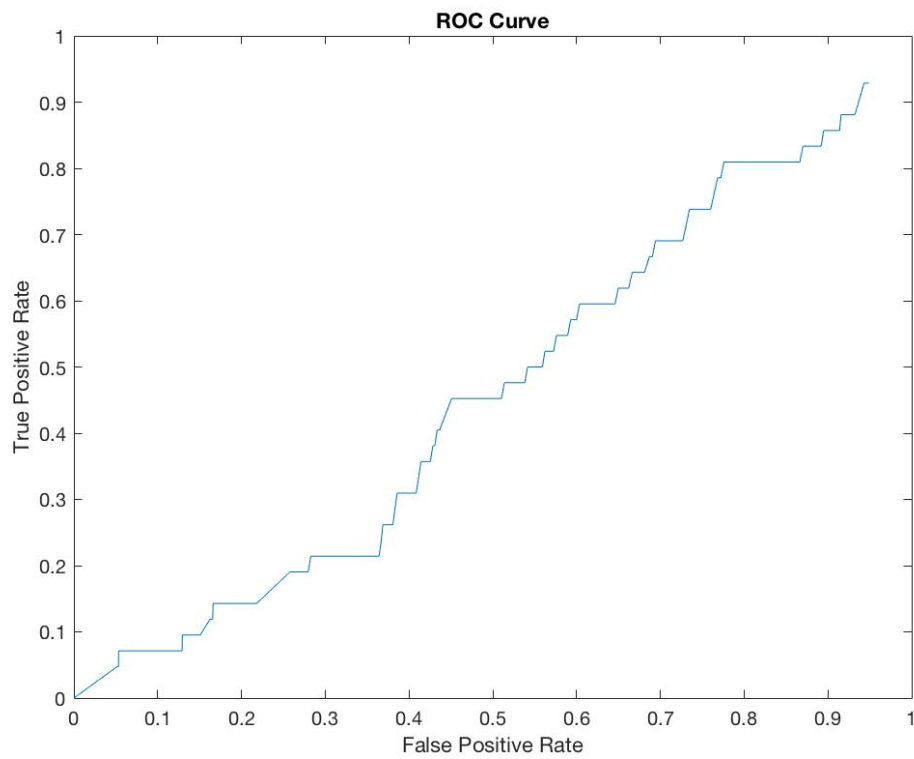
for LG4000



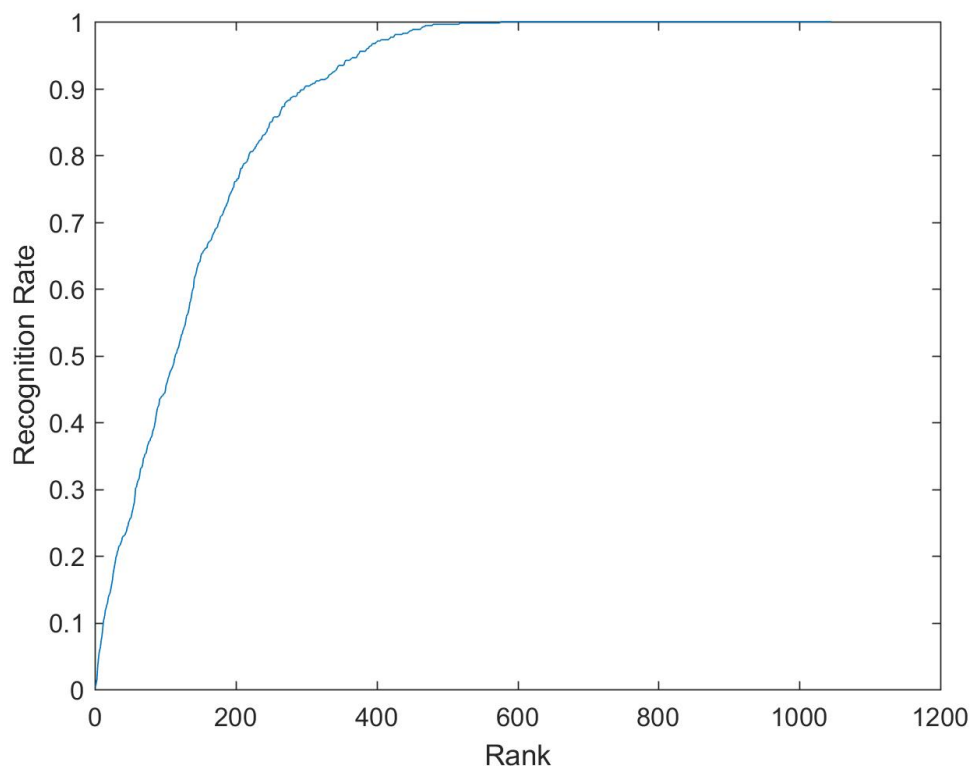
ROC curves, CMC curves

for LG2200:

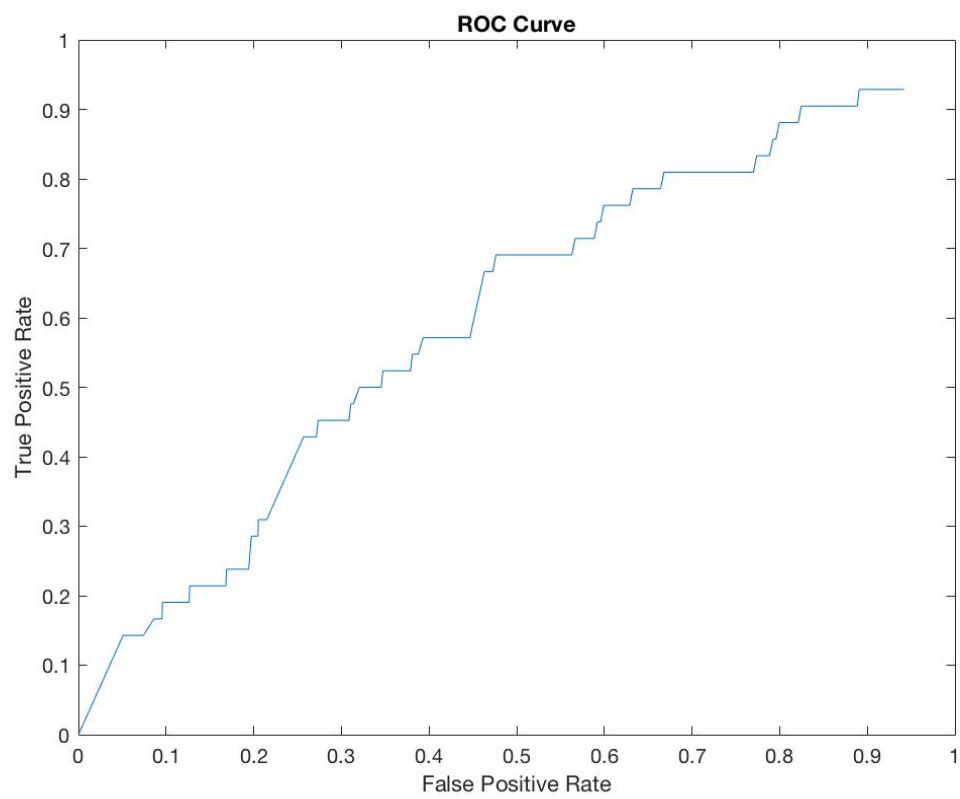
ROC is



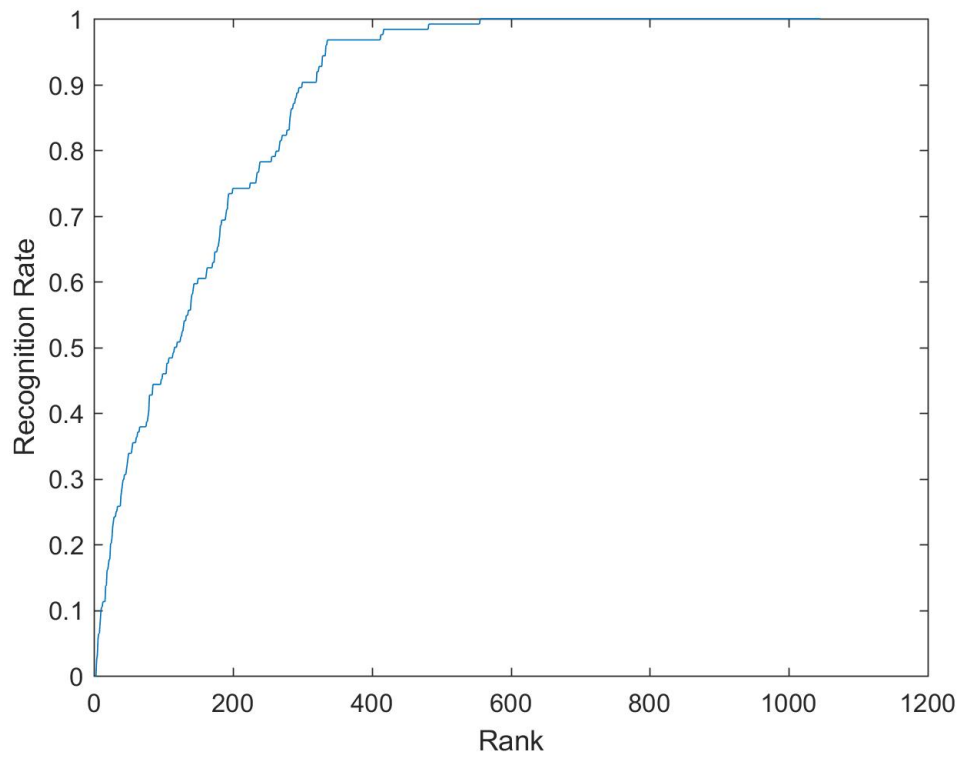
CMC is



for LG4000:
ROC is



CMC is



Our finds

1. There are some images which **createiristemplate.m** cannot use because the images is dim or the eye is not open completely. Such as the 05290d152.tiff, 06260d32.tiff, 06267d57.tiff.
2. The number of imposter is much more than the genuine.
3. Because the all genuine scores and imposter scores are distributed in similar interval. The ROC is like a 45 degree line.
4. If we want to deal with all the images of all the images, we must use mapreduce.

Reference: [https://www.researchgate.net/post/](https://www.researchgate.net/post/How_is_CMC_produced_recognition_rate_vs_Rank_for_unknown_faces)

How_is_CMC_produced_recognition_rate_vs_Rank_for_unknown_faces