**UFPR - Universidade Federal do Paraná**
*Bachelor of Computer Science - Topics in Computer Vision - CI-396*
Matheus Agio Nerone - GRR 20136694
matheus.agio@hotmail.com

# Problem 02: Iris Recognition

## How to execute

python <path to main.py>
If you are in the project root directory execute: python main.py

## Code Organization

The code is organized in the following way:

- Databases (folder): Place where the test database is located.
    - CASIA-Iris-Lamp-100 (folder): Test database with the right and left eye pictures of 15 people, each person has 10 pictures of each eye.
- iris_signaturizer.py: Implementation of the proposed method to create a signature of the iris, the code is modular so it can be used elsewhere, although I don't recommed, because the results are not good, as shown below.
- local_binary_patterns.py: Code responsible for calculating the local binarry patterns of an image.
- casia_iris_image_loader.py: code to load the Casia Iris Lamp database.
- main.py: Where the tests are done:
    1. Firstly the verification, the test begin calculating the hamming distances between every image, then the False Acceptance Rate and False Rejection Rate is calculated for a couple of thresholds, and finally the Equal Error Rate is calculated based on the FAR and FRR founds.
    2. Secondly the identification, it starts calculating the Linear Binary Pattern for all the images and then separate 10% of the images for testing and the 90% for training a SVM, then all the test images are predicted by the SVM and the accuracy is calculated, this process is done ten times and the average accuracy and the standard deviation are calculated.

**UFPR - Universidade Federal do Paraná**
*Bachelor of Computer Science - Topics in Computer Vision - CI-396*
Matheus Agio Nerone - GRR 20136694
matheus.agio@hotmail.com

# Results

The tests were run locally on my machine, because macalan was reporting the error on this issue, the tests were made using python 2.7.12

## Verification

The threshold is the maximum difference between the two signatures

| Threshold | False Rejection Rate(%) | False Acceptance Rate(%) |
|---|---|---|
| 0.05 | 97.57 | 0.17 |
| 0.10 | 88.14 | 2.40 |
| 0.15 | 79.78 | 5.64 |
| 0.20 | 59.47 | 19.78 |
| 0.25 | 37.59 | 43.62 |
| 0.30 | 27.98 | 56.53 |
| 0.35 | 13.17 | 78.07 |
| 0.40 | 8.87 | 85.74 |
| 0.45 | 3.40 | 95.09 |
| 0.50 | 0.89 | 98.77 |
| 0.55 | 0.49 | 99.44 |
| 0.60 | 0.17 | 99.89 |
| 0.65 | 0.07 | 99.96 |
| 0.70 | 0.01 | 99.99 |
| 0.75 | 0.01 | 100.00 |
| 0.80 | 0.00 | 100.00 |
| 0.85 | 0.00 | 100.00 |
| 0.90 | 0.00 | 100.00 |
| 0.95 | 0.00 | 100.00 |

The Equal Error Rate is near the threshold 0.25, with a False Rejection Rate of 37.60 and a False Acceptance Rate 43.63.

**UFPR - Universidade Federal do Paraná**
*Bachelor of Computer Science - Topics in Computer Vision - CI-396*
Matheus Agio Nerone - GRR 20136694
matheus.agio@hotmail.com

# Identification

Using the proposed method the results found were:

Accuracy: 89.7%
Standard Deviation: 0.048

# Image pre-processing

For every eye image the process shown below is executed:

1. A medianBlur filter is applied to the image to reduce the eyelashes and light noises on the image.
2. Then the image is thresholded so that only the pupil part is visible in black.
3. A Canny filter is applied to find the pupil borders.
4. The image with the pupil borders is returned.