Iris recognition

Phase 1

Name	ID
Mahmoud Mohamed Saeed Ahmed	20180261
Mostafa Abdelsalam Abdelbaseer	20190534
Ali Mohamed Abdelsamea Ali	20190342
Doaa Osama Saad Ahmed	20190194
Abdelrahman Mohamed Abdelzaher	20190320

Abstract

Iris recognition is a form of biometric identification performed with computer vision. The iris is an internal organ whose texture is ran- dimly determined during embryonic gestation. It is at and its texture is relatively stable, as the iris is protected from external harm. All these features make it a very viable form of biometric recognition. We use an ellipse-fitting technique for iris segmentation, rectify the irises to rectangular coordinates, and feed the irises into a feature extractor. We use various features, such as HOG, DAISY, and a random forest classifier to obtain a 0% error on the MIU dataset.

Problem definition, objectives:

A biometric framework provides automated identification proof for an individual based on the individual's unique traits or features. As the demand for safe identification grows, and the human iris provides an excellent pattern for identification, the use of low-cost technology might enable iris recognition become a new standard in security. Iris recognition is widely regarded as the most accurate and trustworthy biometric identification system currently available. To assess the performance of iris recognition algorithms, picture quality, and acceptance rate, a test case based on open source code may be constructed .The picture quality of photographs as data in this project.

Iris recognition has been widely applied in various fields, e.g. door access-control systems, passport checking and banking services. An advantage of using biometric authentication is that it cannot be lost or forgotten, as the person has to be physically present during at the point of identification process.

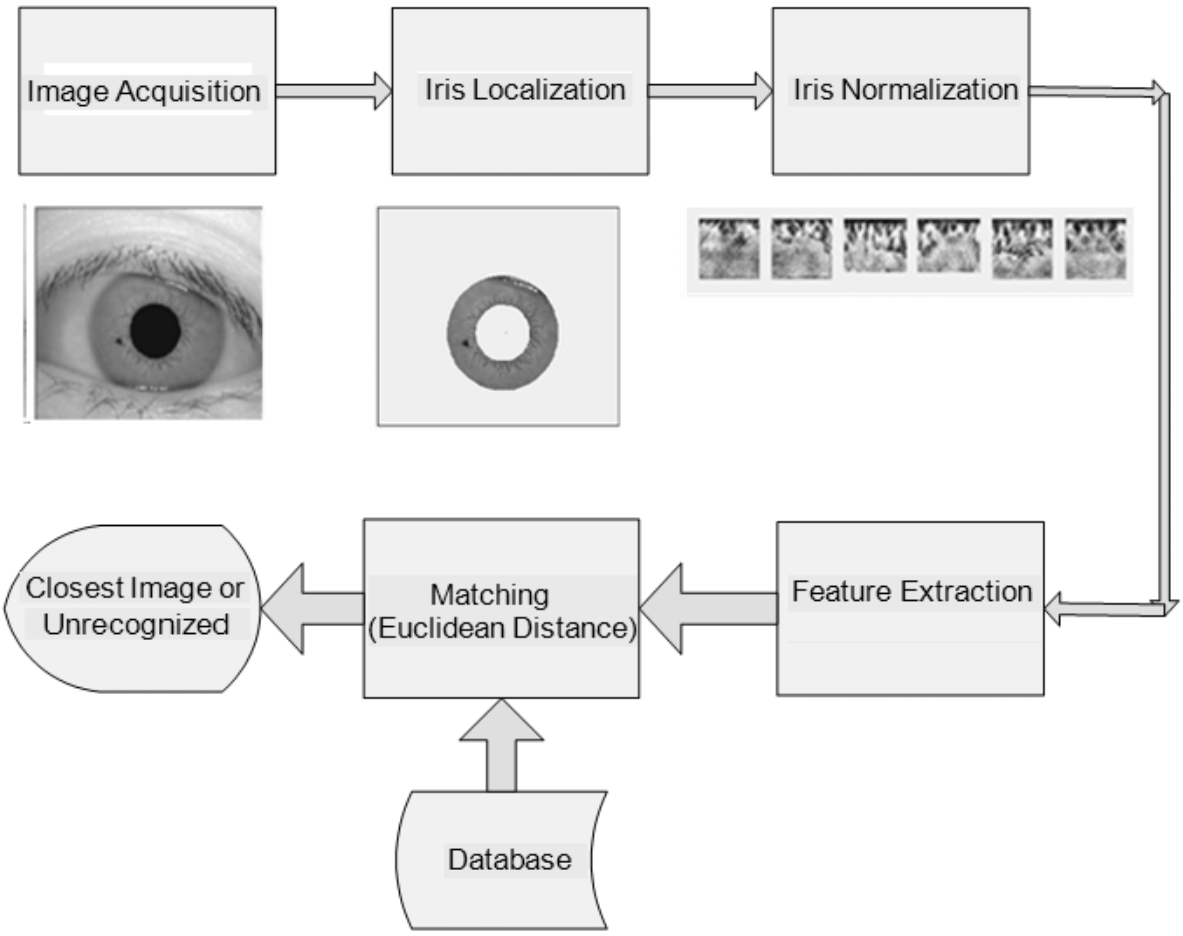
The purpose of this project will be to implement an iris recognition and identification system which can authenticate the claimed performance of the methodology.

Methodology:

The system is to be composed of a number of sub-systems, which correspond to each stage of iris recognition.

Some of these stages are:

1- Image acquisition: The iris image should be rich in iris texture as the feature extraction stage depends upon the image quality.
2- Segmentation: The first stage of iris recognition is to isolate the actual iris region in a digital eye image
3- Normalization: creating a dimensionally consistent representation of the iris region
(Canny Edging)



Enhancing techniques could be used

- Histogram enhancement, median filter, Sobel filter and averagefilter techniques:

```
import cv2
import numpy as np


from matplotlib import pyplot as plt # Histogram
# reads an input image

img = cv2.imread("luftwaffe.png", cv2.IMREAD_COLOR) # converted it to 2D Grey img
image = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

# plotting histogram for the original 2D Grey img cv2.imwrite("djfsd.png",image)


#plt.hist(image) #plt.title("Image Histogram") plt.xlabel("greyscale") plt.ylabel("freq")
# plt.figure() # plt.show()


# Histogram
# Histogram equalization

HisEqu = cv2.equalizeHist(image) cv2.imshow("Image Equalization", HisEqu) cv2.waitKey(1000) cv2.destroyAllWindows()


# writing img to file plt.hist(HisEqu)
plt.title("Equalized Image Histogram ") plt.xlabel("grayscale value") plt.ylabel("freq")
plt.figure() plt.show()


# Histogram equalization
# smoothing filters
# Average filter

blurred = cv2.blur(image, (10, 10)) cv2.imshow("Image average filter", blurred) cv2.waitKey(4000)
cv2.destroyAllWindows()

# Median filter

medBlur = cv2.medianBlur(image, 5) cv2.imshow('Image median filter', medBlur) cv2.waitKey(4000) cv2.destroyAllWindows()


# Sobel filter

sobelX = cv2.Sobel(image, cv2.CV_64F, 1, 0, ksize=3) print("Sobel X filter applied")
sobelY = cv2.Sobel(image, cv2.CV_64F, 0, 1, ksize=3) print("Sobel Y filter applied")
cv2.imshow('Image sobel X filter', sobelX) cv2.imshow('Image sobel y filter', sobelY) cv2.waitKey(4000) cv2.destroyAllWindows()

# Sobel filter
# smoothing filt
```