**Biometrics and**

**Face Recognition**

**Introduction**

Over the last ten years or so, face recognition has become a popular area of research in computer vision and one of the most successful applications of image analysis and understanding. Because of the nature of the problem, not only computer science researchers are interested in it, but neuroscientists and psychologists also. It is the general opinion that advances in computer vision research will provide useful insights to neuroscientists and psychologists into how human brain works, and vice versa.

A general statement of the face recognition problem (in computer vision) can be formulated as follows: Given still or video images of a scene, identify or verify one or more persons in the scene using a stored database of faces [1].

There are many approaches used for the process of Face recognition. The general approach used for the process is Eigenfaces. Eigenfaces is the name given to a set of [eigenvectors](http://en.wikipedia.org/wiki/Eigenvector) when they are used in the [computer vision](http://en.wikipedia.org/wiki/Computer_vision) problem of human [face recognition](http://en.wikipedia.org/wiki/Facial_recognition_system). The approach of using eigenfaces for [recognition](http://en.wikipedia.org/wiki/Facial_recognition_system) was developed by Sirovich and Kirby (1987) and used by [Matthew Turk](http://en.wikipedia.org/w/index.php?title=Matthew_Turk&action=edit&redlink=1) and [Alex Pentland](http://en.wikipedia.org/wiki/Alex_Pentland) in face classification.

**Training**

Dataset used: <http://vision.ucsd.edu/~iskwak/ExtYaleDatabase/ExtYaleB.html>

The dataset used in this program consists of 10 persons with 15 images of each person stored in a folder named ‘sn’ (n=1 to 10).

Eg. Person s1 consists of:



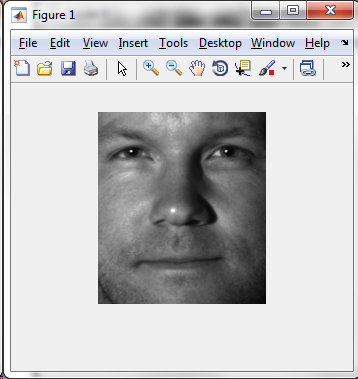
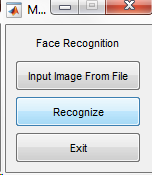
When the training is option is selected and the specific path is given, the program creates a file ‘training.mat’ and stores all the corresponding vectors of the eigenfaces in the file.

**Face Recognition**

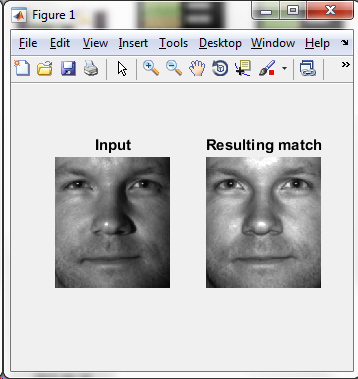
When an image is given as input to the program, the program returns the corresponding person’s image along with the name (in console).

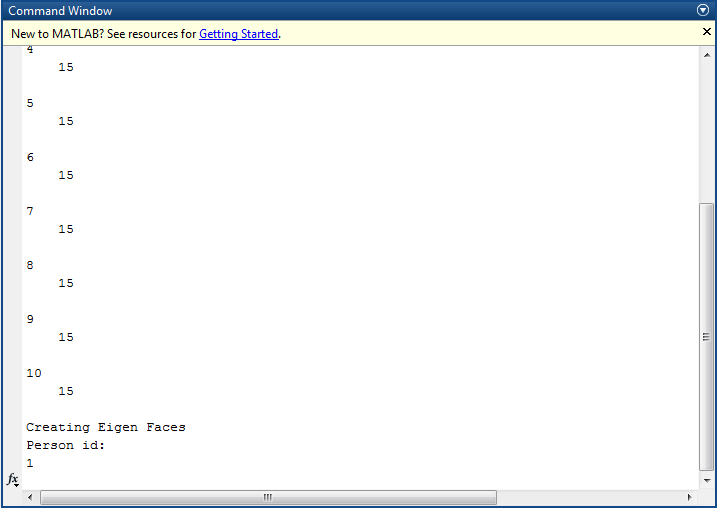
Eg.:

Input Image:

When the recognise button is clicked:



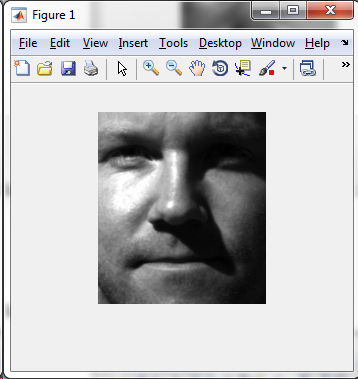


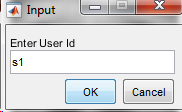
It’s a match..!!

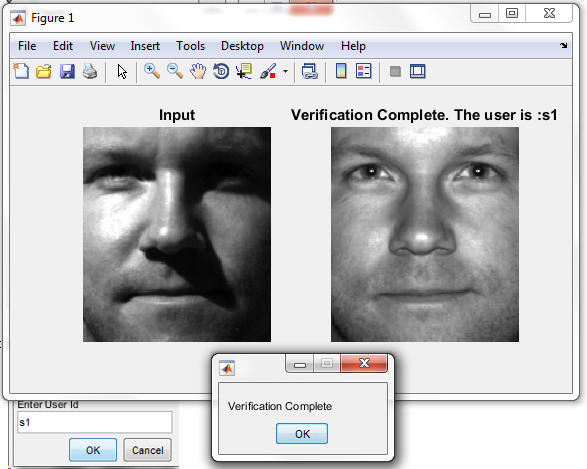
**Verification:**

In this process, the user is prompted to input an image along with the id (sn) of the person. If the image corresponds to the id, then it’s a match.

Eg.:

Input Image: 

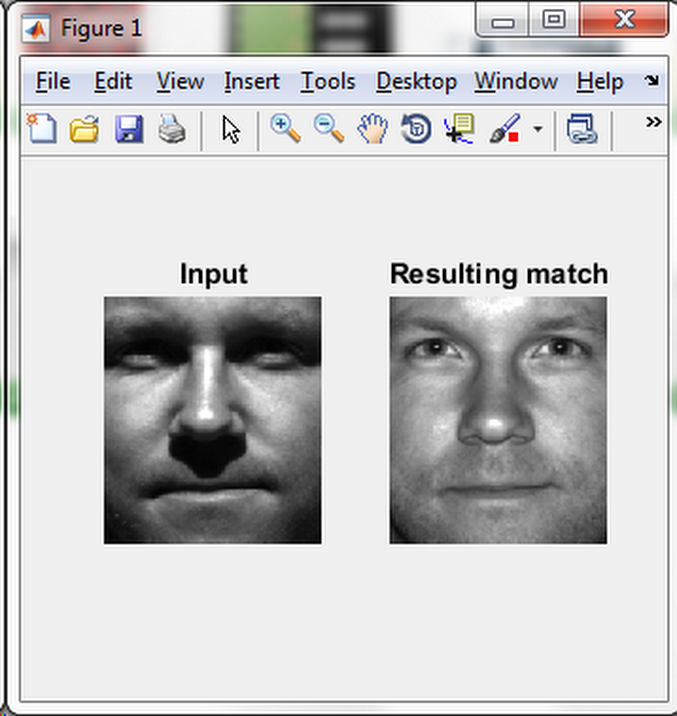
Input name: 

Result: 

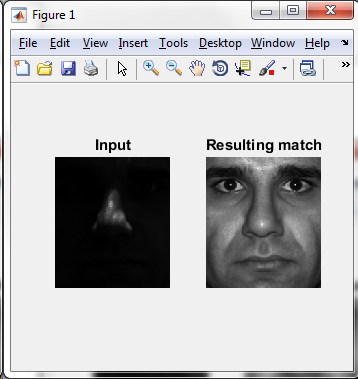
**Performance:**

The program was run 5 times and the result is as follows:

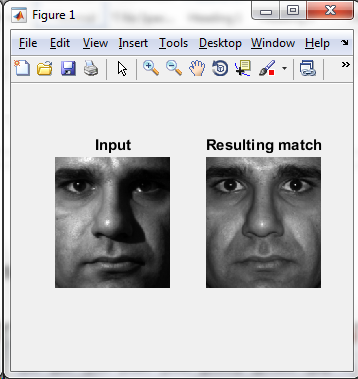
**Test case 1:**

 Input: s1; Result: s1. Match

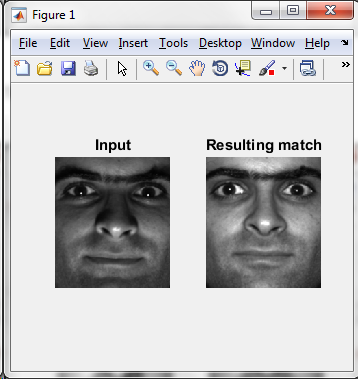
**Test case 2:**

 Input: s3; Result: s6; No Match

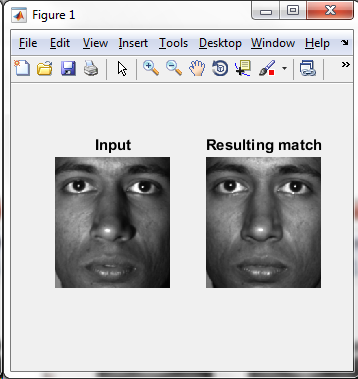
**Test case 3:**

 Input: s6; Result: s6; Match

**Test case 4:**

 Input: s9; Result: s9; Match

**Test case 5:**

 Input: s10; Result: s10; Match

The program was able to match the proper images 4 out of 5 times

**Additional Information**

**Hardware Specification**

Model : HP Pavilion dm4 Notebook PC

Processor : Intel(R) Core(TM) i5-2450M CPU @ 2.50GHz

Memory : 6.00 GB

System Type : 64-bit Operating System

Operating system : Windows 7 Home Premium

**Software Specification**

Software Used : MATLAB

Type : 64-bit (win64)

Build : R2015a (8.5.0.197613)

**References**

1. <http://www.face-rec.org/general-info/>
2. <http://vision.ucsd.edu/~iskwak/ExtYaleDatabase/ExtYaleB.html>
3. <http://www.mathworks.com/products/matlab/>