**Face Recognition and matching**

**Image and Vision Computing Project**

**CSE327**

**Submitted by:**

Ayush Raj 14BCE0105

Aviral Upadhyay 14BCE0250

**Submitted to:**

Prof. Srivani A

Fall 2016-2017

**Abstract**

Facial Recognition is an ever essential process in today’s world. It refers to the process of taking a face as an input, and displaying information mapped to that given face accordingly. It is has applications in multiple areas such as by police organisations to scan huge databases and accurately recognise criminals, by Facebook to recognise and tag people, by camera applications to highlight and improve the quality of face in pictures and so on. This project will take on the problem of taking input of faces and recognising them. The project will initially create a database of the histograms of all the pictures present in the database of the program in its first run. Next it will recognise input images.

**Introduction**

Facial Recognition is a software using which a computer detects a face and is capable of identifying or verifying that face. It has several applications from identification to security and essentially a 21st Century technique. However it was first pioneered in the mid-1960s. Bledsoe along with Helen Chan and Charles Bisson worked on using computers to recognise human faces. The success of the method could be measured by the number of records in the database. By 1997 many professional systems were developed and by 2007, Polar Rose technology could guess what a person would look like in 3 dimensions in 1.5 seconds. The US government sponsored several evaluation and challenge problems and they along with competitions such as Face Recognition Grand Challenge led to exponential improvement in face recognition technology.

There are essentially 4 techniques for facial recognition as of now. They are:

1. Traditional Technique: This methodology involves algorithms that identify facial features by extracting landmarks and features from an image of the subject’s face. It can use multiple images to have a better understanding of the variation of the subject’s relative position of jaw, nose, eyes, cheek etc.
2. 3 Dimensional Recognition: This technique uses sensors in 3 dimensions to capture information about shape of the surface. This information is then used to generate information on distinct features of surface of face such as shape of eye sockets, chin etc. It is a relatively new technique which is more accurate than the traditional method.
3. Skin Texture Analysis: This is an emerging trend which uses the visual details of the skin to generate unique patterns, lines and spots which can next be transformed to a mathematical form. This method is not very accurate in itself and it can be used as an add-on the previous two for improved performance.
4. Thermal Cameras: In this process we take the input data for facial recognition using thermal cameras. The sensor will detect only the shape of the head, ignoring any extra accessories. It is a new field researched into only in the 21st Century. The main issue with this is that the databases for facial recognition is limited. It uses low-sensitive, low resolution ferro-electric sensors that are capable of acquiring long wave thermal infrared.

Facial recognition is an essential and power tool no doubt. However it suffers from its own set of weakness and disadvantages when compared to other technologies. Technologies such as biometric and retina recognition offer higher degree of accuracy when compared to facial recognition. Still facial recognition is important as it doesn’t need cooperation from the test subjects. For example it can be installed in an airport to recognise a certain individual among the crowd. Facial recognition also struggles to perform under certain conditions. Viewing angle of the face, long hair, partially covered face often throw facial recognition software off track. Even poor lightning conditions or a smile can at times render face recognition systems ineffective.

We’re going to use the traditional method in MATLAB in this project. We’ll be utilising the inbuilt library of MATLAB to use algorithms such as Viola-Jones and Histogram of Gradient Objects. Section 2 of this project contains the Literature Review. It is followed by the methodology used in this project in section 3. Next we’re going to discuss the results, output and discussions in section 4. In section 5 we will conclude the paper.

**Literature Review**

Image recognition is an invaluable and omnipresent tool in today’s world. Be it in the sector of security of entertainment or of utility. It is everywhere. In this project we’ll be using methods of image processing to detect store and recognize faces. We’ll provide a program with several images of a person and using algorithms, teach them to recognize a specific face from the set of faces stored in the system’s memory.

The Viola-Jones object detection framework is the first object detection framework which can provide object detection rates in real time. It was initially proposed as a face detection framework by Viola-Jones which according to them was capable of processing images extremely rapidly while achieving high detection rates [ref.1]. In this project we’ll be using the Viola-Jones algorithm to simply detect a face in an image, draw a rectangular boundary across it and finally crop it out, discarding rest of the image. Computer Vision Toolbox from MATLAB has the implementation of Viola-Jones algorithm using vision.CascadeObjectDetector() and step() function that detects faces in an image. Although it can detect multiple faces, if there are multiple faces in the image, we’ll crop the top most left face from the image as the algorithm is good only for one face. Also there are several limitations to Viola Jones, one of them being only to detect the frontal view of the face and hence we have only implemented this program if the face has frontal view and not side way.

Next we will resize the image to a standard 112\*92 pixels and covert it to gray scale for ease of processing. Now we’ll use Histogram of Oriented Gradient (HOG). HOG is a way to differentiate between multiple images of multiple objects. According to [ref 3] it is a robust visual object recognition, which adopts linear SVM based human detection as a test case. (SVMs are support vector machines which are supervised learning models with associated learning algorithms). Here we’ll be using HOG to differentiate the face of one person from one another. We use it to create gradients on the given image of face. First, in order to compensate for errors in facial feature detection due to occlusions, pose and illumination changes, we will extract HOG descriptors from a regular grid. Second, fusion of HOG descriptors at different scales will allow us to capture important structure for face recognition. And lastly we’ll identify the necessity of performing dimensionality reduction to remove noise and make the classification process less prone to over-fitting [ref 4].

extractHOGFeature() is the function in MATLAB that take an input image as an argument and creates vector of the gradients which we can easily see when we plot the features. There are better methods available to differentiate between two images such as a fisher vector [ref 8], but it takes too much of computational time to be done on a simple laptop.

Error-Correcting Output Codes(ECOC) is an ensemble method designed for multi-class classification problem. In multi-class classification problem, the task is to decide one label from k > 2 possible choices. For example, in digit recognition task, we need to map each hand written digit to one of k = 10 classes. Some algorithms, such as decision tree, naive bayes and neural network, can handle multi-class problem directly. ECOC is a meta method which combines many binary classifiers in order to solve the multi-class problem.

Fitcecoc() is implementation of classification by ECOC already present in the machine learning toolbox, we use this toolbox to classify each face to a person.

After the image has been loaded and processed, we’ll store it’s gradients or features in a separate database.

The fitcecoc() is a classification function from MATLAB which classifies every image to a name and creates a database for it. We’ll be taking multiple input images of a person to teach the machine to recognize that specific person’s face. The fitcecoc() will create a class of all the faces that belongs to the same person. To match a test subject to each class we’ll be using predict function in the database created by fitcecoc().

In a facial recognition system, there are two steps, the first is training in which the machines learns which face belongs to which person (we can display other details about the detected image). The second step is matching each the test image to the database created using training and find out the person. The steps on how to perform and create a database, i.e. the basic idea of storing were taken from a Youtube video, ref 6. Just like a human being, the more the machine knows about the person the better it will be able to detect it from a test image, and hence either a better algorithm to extract the features or a bigger database will result in better output but still it can result in some errors and accuracy of 70% can be achieved easily, with a medium sized database(50 frontal images of a person)

**Methodology**

The project was completed using MATLAB.

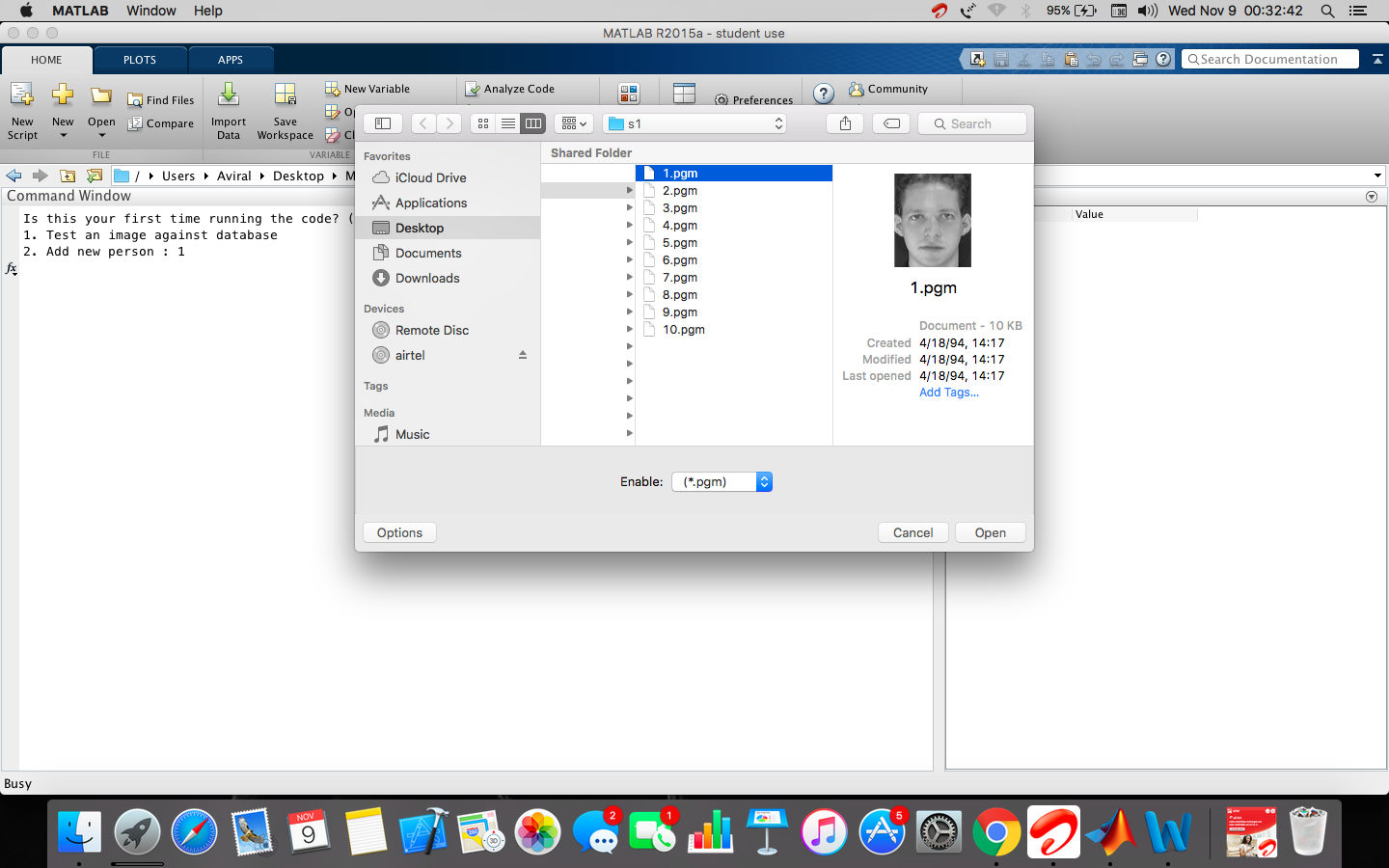
First the user chooses if it’s his first time running the program, if yes then the program creates .mat file as a database of image, which has all the feature of a particular person. Which can be seen in createDatabase.m, in here we use HOG features and multi class error correction code to classify eacjh person and differentiate all the person in the database folder and store it in a varible on the physical drive for quick processing in future.

During the consiqutive run, the user selects an image of a person, an format of image, the face of the person is extracted using Viola-Jones algorithm and then scale it down to 112\*92 grayscale image so as to generate 4192 HOG features as all the image in database are of this type. Then the MATLAB uses predict fucntion(machine learning toolbox) to predict the person’s name from the database. Then it displays 3 pictures of the same person.

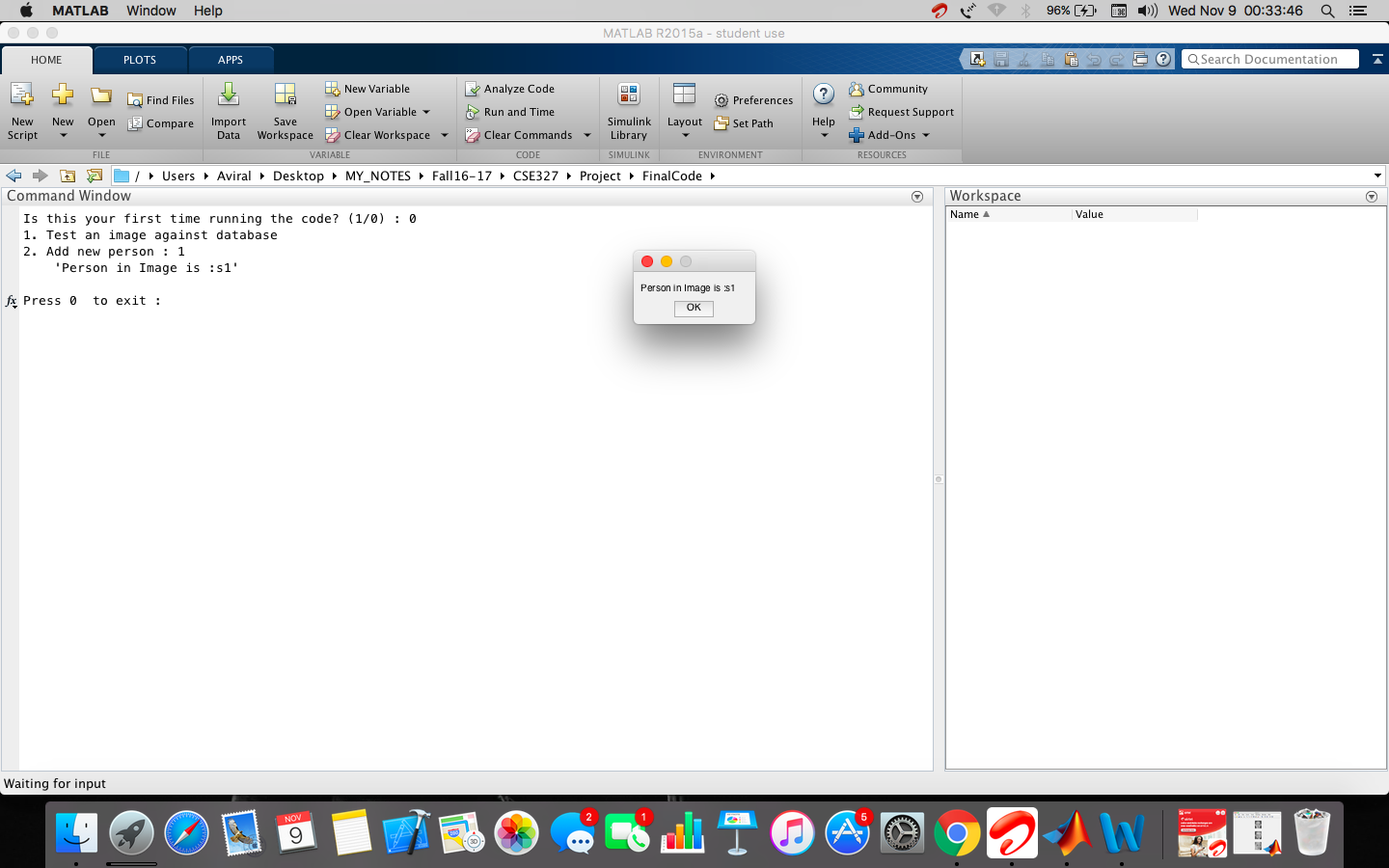
One can also add a new person to database, by simply selection the folder which has the person’s image, again all the images are converted to standard form and then stored in the database. createDatavase.m is used againhere to update database.mat.

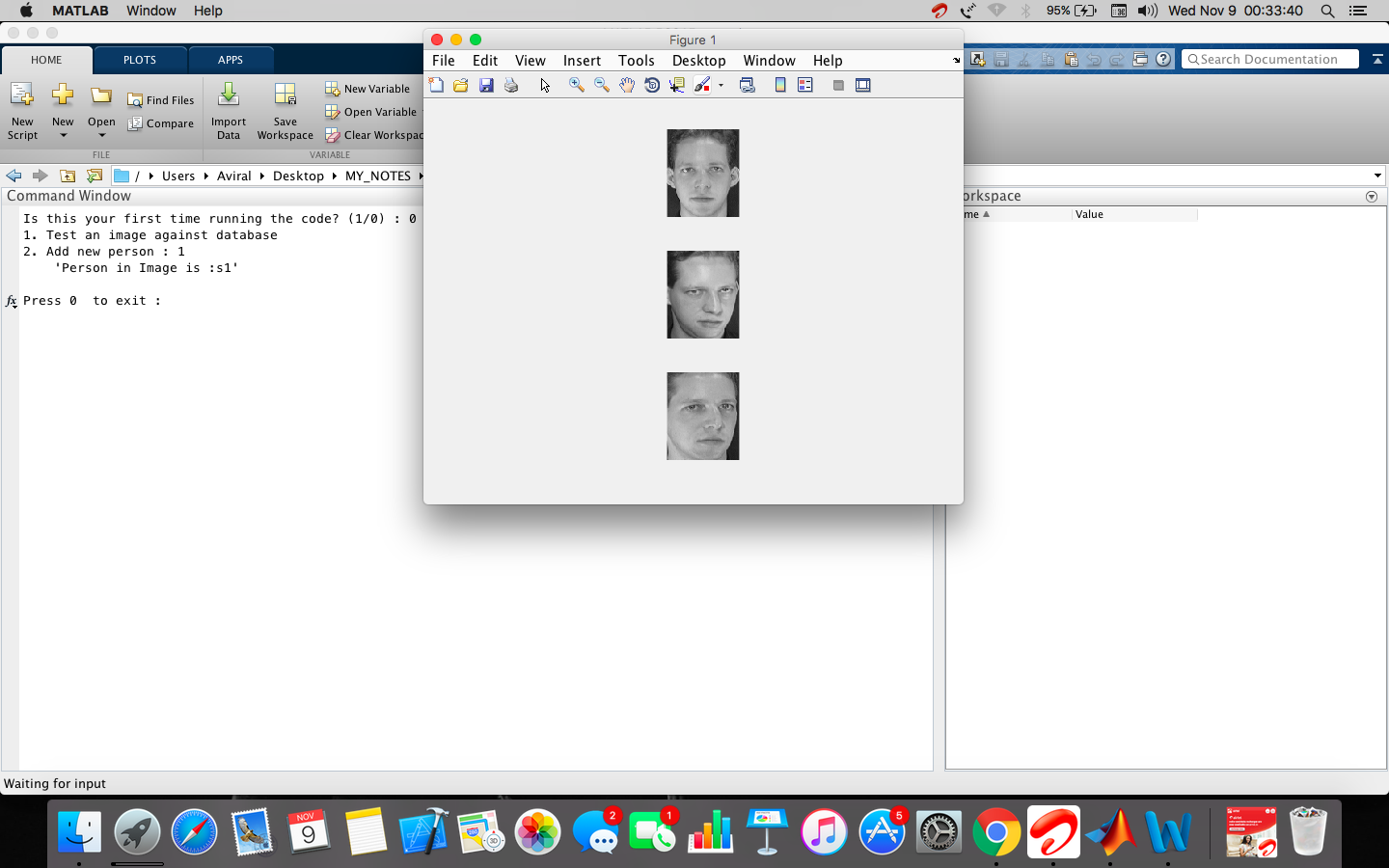
**Results and Output**

To run the code, just run “theCode.m” in matlab and follow on screen instructions.

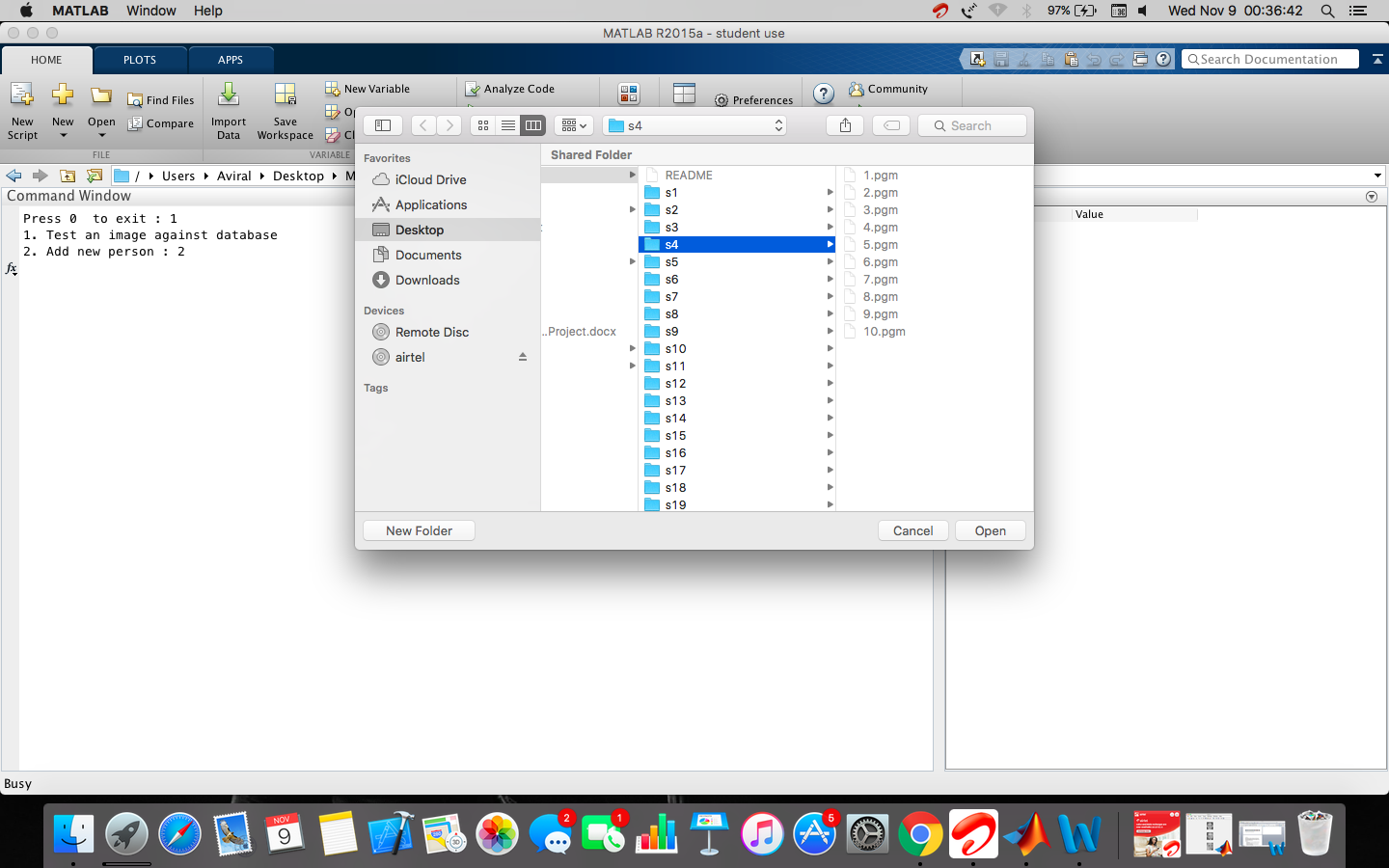
Selecting a test image to search person’s name.

We’ve already run the code for the first time. The database directory is selected by us. All the directories contained in it have their names as person’s name whose photograph is contained in the directory. Next we’re selecting a picture to be compared against all images in the database.

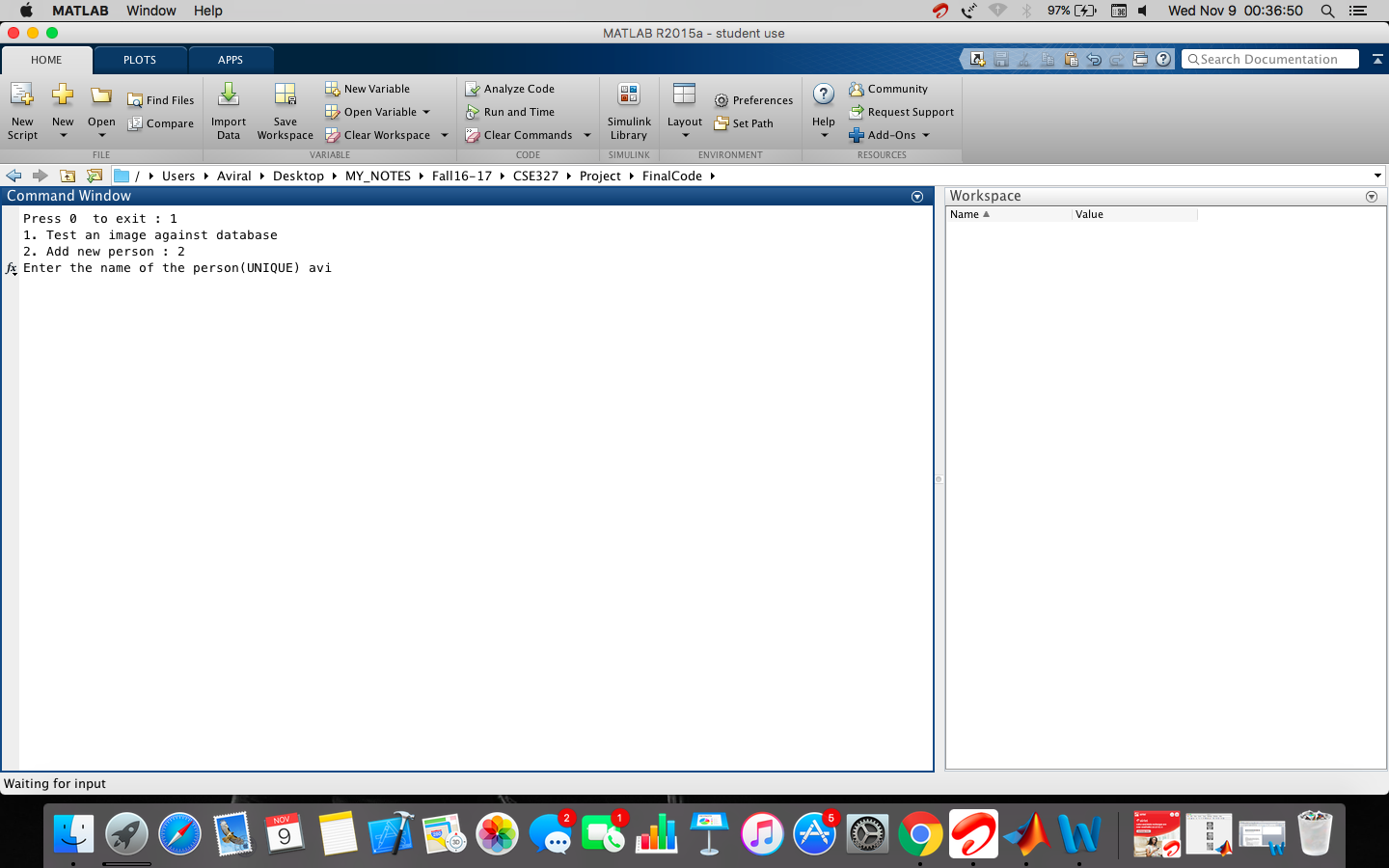
After the programs has detected that the given image corresponds to one of those in its database, it displays the directory name on the screen, aka the person’s name.



Here we’re showing a few more images of the person who was detected to be in the provided image.



Here we’re adding a new person to the database.

Selecting name of the new person

**Conclusion**

Facial recognition software’s prototype was developed using MATLAB which gives about 70% accuracy. Even though the accuracy does not measure upto the best facial recognition hybrid algorithms (which can offer accuracy of over 90%). It is a suitable illustration of how facial recognition technology works and what are the various steps pertaining to it. It can be used among various applications such as crime investigation, military or even for personal device security.

**References**

1. Robust Real-Time Face Detection (rev 2003) by Paul Viola and Micheal J Jones.
2. Rapid Object Detection using a Boosted Cascade of Simple Features by Paul Viola (2001)
3. Histograms of Oriented Gradients for Human Detection by Navneet Dalal, Bill Triggs (2010)
4. Face recognition using Histograms of Oriented Gradients by O. Déniz, G. Bueno, J. Salido, F. De la Torre (2011)
5. http://in.mathworks.com/help/stats/classificationecoc-class.html
6. <http://in.mathworks.com/help/stats/fitcecoc.html#bufm0tv>
7. <https://www.youtube.com/watch?v=syOgjbjSpGk>
8. http://www.vlfeat.org/api/fisher-fundamentals.html