Software Engineering (UCS503) Project Report (BE, 3rd Year, November 2017)

Attendance Using Facial Recognition

Submitted by
Abhishek Handa (101503006)
Abhishek Sharma (101503008)
Akshay Sharma (101683035)

Group: COE-1

Supervisor
Dr. Sanmeet Bhatia
Assistant Professor
CSED
Thapar University, Patiala

COMPUTER SCIENCE ENGINEERING DEPARTMENT
THAPAR UNIVERSITY, PATIALA-147004, PUNJAB
INDIA
November, 2017

TABLE OF CONTENTS

Chapter No.	Topic	Page No.
1.	Introduction	3
1.1	Purpose of this Document	3
1.2	Scope of the Development Project	3
1.3	Definitions, abbreviations and acronyms	4
1.4	References	5
1.5	Overview	5
2.	Overall Description	6
2.1	Product Perspective	7
2.2	Product functions	8
2.3	User Characteristics	8
2.4	General Constraints, Assumptions and Dependencies	9
2.5	Apportioning of the requirements	9
3.	Specific Requirements	10
3.1	External Interface Requirements	10
3.2	Detailed Description of Functional Requirements	11
3.2.1	Functional Requirements for Student Welcome Screen	11
3.2.2	Functional Requirements for Staff Welcome Screen	12
3.2.3	Functional Requirements for Student cum Staff Welcome Screen	12
3.3	Performance requirements	12
3.4	Diagrams and Screenshots	13
3.5	Quality attributes	25
3.6	Other requirements	25
4.	Change History	25
5.	Document Approvers	25

1. Introduction

1.1 Purpose of this Document

The purpose of this SRS document is to provide a detailed overview of our software product, its parameters and goals. This document describes the project's target audience and its user interface, hardware and software requirements. **It defines how our client, team and audience see the product and its functionality.**

1.2 Scope of the Development Project

The goal is to design software for attendance system using Facial Recognition. First, look at a picture and find all the faces in it. Second, focus on each face and be able to understand that even if a face is turned in a weird direction or in bad lighting, it is still the same person. Third, be able to pick out unique features of the face that you can use to tell it apart from other people— like how big the eyes are, how long the face is, etc. Finally, compare the unique features of that face to all the people you already know to determine the person's name. The scope of this system is not just limited to the TU campus only as the same mechanism can be reused in other campuses as well. This system can also be implemented in the industrial sector as well.

The software must be able to perform the following operations:

- Login: The user login to the system by entering valid username & password. If username and password is incorrect, then user will not get access to the system.
- Capture image/video: The user uses the camera to capture the images of the suspected person so as to detect it for comparison with the database.
- Sketch matching: A technique in which we are using the templates of images to make sketches and retrieve it from the database if available.
- Interact: Detection and recognition of object which is placed in front of camera is done. All image processing algorithms are applied to identify the image.

Perform operation: If image is already present in system, then information related to object is retrieved through database and it will be displayed and message will be send. Otherwise image have to be added into the system.

- Add images: If the image is new to the system then it must have to be added in a system. Next time when we placed that image in provided space then system will identify the image.
- Detect and recognize: Another technique which we are using for facial recognition is to recognize the faces in the images and display the result.

1.3 Definitions

Definitions

Table 1 gives explanation of the most commonly used terms in this SRS document.

Table 1: Definitions for most commonly used terms

S.No.	Term	Definition
1	Face Localization	Seeks to determine the position of a single face within an image; the detection problem is simplified since the input image contains only one face.
2	Facial feature detection	Seeks to detect the presence and location of features, such as the mouth, nose, eyes, lips, ears, etc.; the detection problem is simplified since the input image contains only one face.
3	Facial expression recognition	Identifies the emotional states of humans, e.g. happy, sad, anger.
4	Face tracking	Face tracking methods estimates the location and possibly the orientation of a face in an image on a continuous basis within real time.

1.4 References

[1] Kaumalee Bogahawatte, Shalinda Adikari faculty of information technology, university of Moratuwa, Katubedda, Sri Lanka ©2013 IEEE.

[2] B. Szanto, P. Pozsegovics, Z. Vamossy, Sz. Sergyan Obuda Univesity of software technology, Budapest, Hungary © 2011 IEEE.

[3] http://www.slideshare.net/kylinfish/video-face-recognition-pattern-recognition-final-report-43037436?qid=f33ae900-4589-4877- a507-ab6973b24828&v=default&b=&from_search=16

[4] A. Jorgensen. AdaBoost and Histograms for Fast Face Detection, 2006.

[5] B.K.L. Erik Hjelmas, "Face Detection: A Survey," Computer Vision and Image Understanding, vol. 3, no. 3, pp. 236-274, Sept. 2001.

[6] D. van Wyk .http://www.cs.uwc.ac.za/index.php/Honours-2006/Desmond-VanWyk.html[online], November 2006

[7] J. Meynet. Fast Face Detection Using AdaBoost, July 2003.

[8] R. Lienhart. and J. Maydt., An extended set of Haar-like features for rapid object detection. In: IEEE ICIP 2002, Vol.1, pp 900-903.

[9] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In Proc. IEEE Conference on Computer Vision and Pattern Recognition, pages 511–518, Dec 2001.

[10] P. Viola and M. Jones. Robust real-time object detection. IEEE ICCV Workshop Statistical and Computational Theories of Vision, July 2001.

[11] Y. Freund and R.E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In Proceedings of the Second Europen Conference on Computational Learning Theory, pages 23–37. Springer-Verlag, 1995.

1.5 Overview

Determine whether or not there are any faces in the camera output and, if present, return the face locations of the images. The biggest face detected, being the user closest to the camera are scaled to a recognizable scale. This detection window is then passed to the face recognition system for recognition.

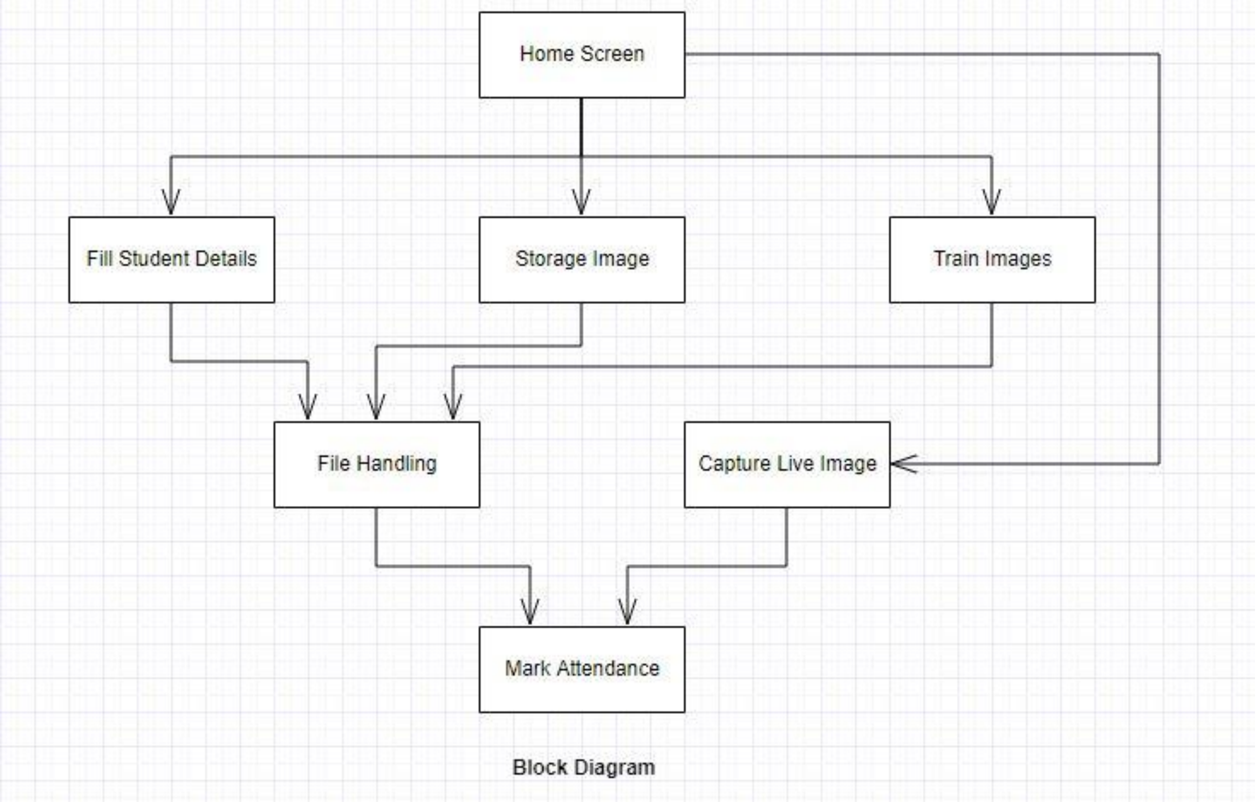
The first and foremost expectation for a face recognition system is that it must have a high degree of accuracy when recognizing people. The next highest expectation from a system is that people should be indicated who they are when the system recognized them. It must also be possible to add and remove people from the system that should be recognized.

The system is to accurately identify and locate human faces under the following conditions and circumstances:

1. upright, frontal face
2. minor variations in lighting conditions
3. and minor variations in facial expression.
4. minor variation in illumination
5. big enough scale in order to perform face recognition.

2. Overall Description

2.1 Product Perspective



Block Diagram For The Entire System

1. **User Image panel**
   The User Image panel displays the user image of the user currently selected in the User list panel.

2. **User List panel**
   The User List Panel lists the authorized users the Access Control system can recognize.

3. **Search field**
   The search field can be used to search for a user in the face recognition system. This feature can be very useful when the face recognition system have hundreds of users. The search function searches the substring of the search text in the username and lists all the matched usernames in alphabetical order, displaying the first matched users' image in the user image panel.

4. **Camera output**
   The camera output displays the camera's output. Used for capturing images when adding users to the face recognition system, and for monitor current user login activity. Add user button. The add user button adds an authorized user to the Access Control system. On execution of the

5. **Add User button**
   The Add User dialog is displayed.

6. **Remove user button**
   The remove button is only enabled when there are users in the system. The remove button removes a user from the Access Control system. On execution of the Remove User button the Confirm User Delete dialog is displayed.

7. **Recognition output**
   The Access Control system displays the output of the recognized user below camera output. When a user is not recognized by the Access Control system, the system displays Who are you, otherwise the system displays the user ID of the recognized user and the rate of the face recognition.

8. **Log panel**
   When a user is detected by the face detection system and recognized by the face recognition system, the Access Control system logs the user's information to the log panel. The log panel logs the user and the time at which the user logged in, the log file gets saves with the current date every time the user terminates the system.

9. **Acceptance threshold**
   The face recognition system has a threshold at which users should be recognized. With 0 being the lowest threshold value and with 1 being the highest threshold value. With 0 being the most lenient and 1 being the strictest at which the recognition operates. The Acceptance threshold spinner is used to adjust the acceptance threshold value used with which the ANNs output is compared.

2.2 Product Functions

The product should be able to perform the following operations:

1. It must be able to capture an image
2. It must be able to identify and display the face in the captured image.
3. It must be able to mark attendance on the Excel Sheet.
4. It must be able to discard the captured image if image not found in the database.
5. It must be able to update student details.
6. It must be able to give privileges to teacher and admin.
7. The attendance should be marked on the basis of current date
8. It must be able to calculate total attendance of the student

2.3 User Characteristics

The goal is to design software for attendance system using Facial Recognition for different users. These user types are listed below as follows:

1. Student
2. Staff
3. Dean
4. Admin

As one can see from the list, each user will have different educational background and expertise level in using the system. Our goal is to develop software that should be easy to use for all types of users. Thus, while designing the software one can assume that each user type has the following characteristics:

- The user is computer-literate and has little or no difficulty in accessing the database.
- In order to use the web camera, it is not required that a user be aware of the internal working of a web cam but he/she is expected to know what happens when the face is detected.

2.4 General Constraints, Assumptions and Dependencies

The following list presents the constraints, assumptions, dependencies or guidelines that are imposed upon implementation Attendance using facial Recognition including Access Control and Attendance Monitoring software:

1. There are memory requirements for database management
2. The product must have a user-friendly interface that is simple enough for all types of users to understand.
3. Response time for loading the software and for processing a transaction depends upon the quality of pictures taken by the webcam.
4. A general knowledge of basic computer skills is required.
5. The central database server and backup database servers should be updated regularly by the respective instructor.
6. The replication of data from central to the backup server has to be Asynchronous as Asynchronous solutions also provide a greater amount of protection by extending the distance between the primary and secondary locations of the data.

2.5 Apportioning of requirements

The Attendance using face recognition system is to be implemented in the following two phases:

i. **Pilot Phase:** Here the Attendance using face recognition system is implemented for few students in the Pilot Phase. Initially we will be providing access privileges for three types of users: student, staff and admin as they will be ones most involved in this phase.

ii. **Institute wide deployment:** Following the successful completion of the pilot phase, we plan to deploy the same across all the groups of Thapar University where attendance will be automated using face recognition system. We will prepare a database for all the students where we will keep at least 10 photos of each student in the database. We are only going to use basic image transformations like rotation and scale that preserve parallel lines (called affine transformations)

Here the same functionalities will be implemented in each phase, the only difference will be the number of transactions being carried out and the scale of implementation.

3. Specific Requirements

3.1 External Interface Requirements

The following list presents the external interface requirements:

1. The product requires good graphics usage.
2. The product does not require usage of sound or animation. The hardware and operating system requires a screen resolution not more than 320 x 240 pixels.
3. Sound is not an essential feature but it can be considered for future variants of the system where in the user will be greeted when the attendance is marked and alerted when not marked.

3.2 Detailed Description of Functional Requirements

Table 3 shows a template that We'll be using to describe functional requirements for four types of users: student, staff, dean and admin as one can easily deduce the functional requirements for other user types with this template.

Table 3: Template for describing functional requirements

Purpose	A description of the functional requirements and its reasons
Inputs	What are the inputs; in what form will they arrive; from what sources can the inputs come; what are the legal domains of each input.
Processing	Describes the outcome rather than the implementation; includes any validity checks on the data, exact timing of operation (if needed), how to handle unexpected or abnormal situations
Outputs	The form, shape, destination and volume of output; output timing; range of parameters in the output; unit of measure of the output; process by which output is stored or destroyed; process for handling error message produced

.

3.2.1 Functional Requirements for Student Welcome Screen

Table 4 gives the functional requirements for Student Welcome Screen.

Table 4: Functional Requirements for Student Welcome Screen

Purpose	This screen thus provides information specific to each student upon the successful identification of the ID no. and the access code with the values stored in the central database server.
Inputs	A student must have to face the camera in frontal upright position so that attendance can be correctly recorded.
Processing	If facial detection is successful, then attendance is marked on the excel sheet corresponding to the student.
Outputs	On successful detection of face a confirmation message will be played.

.

3.2.2 Functional Requirements for Staff Welcome Screen

Table 5 gives the functional requirements for Staff Welcome Screen.

Table 5: Functional Requirements for Staff Welcome Screen

Purpose	The screen provides access to the database.
Inputs	The admin provides unique login id and password to each teacher for login into the system.
Processing	The admin will be able to access all the records of the students.
Outputs	The changes occur in staff's member database will reflect in admins database whenever a staff member updates their database.

3.2.3 Functional Requirements for Admin Welcome Screen

Table 6 gives the functional requirements for Admin Welcome Screen.

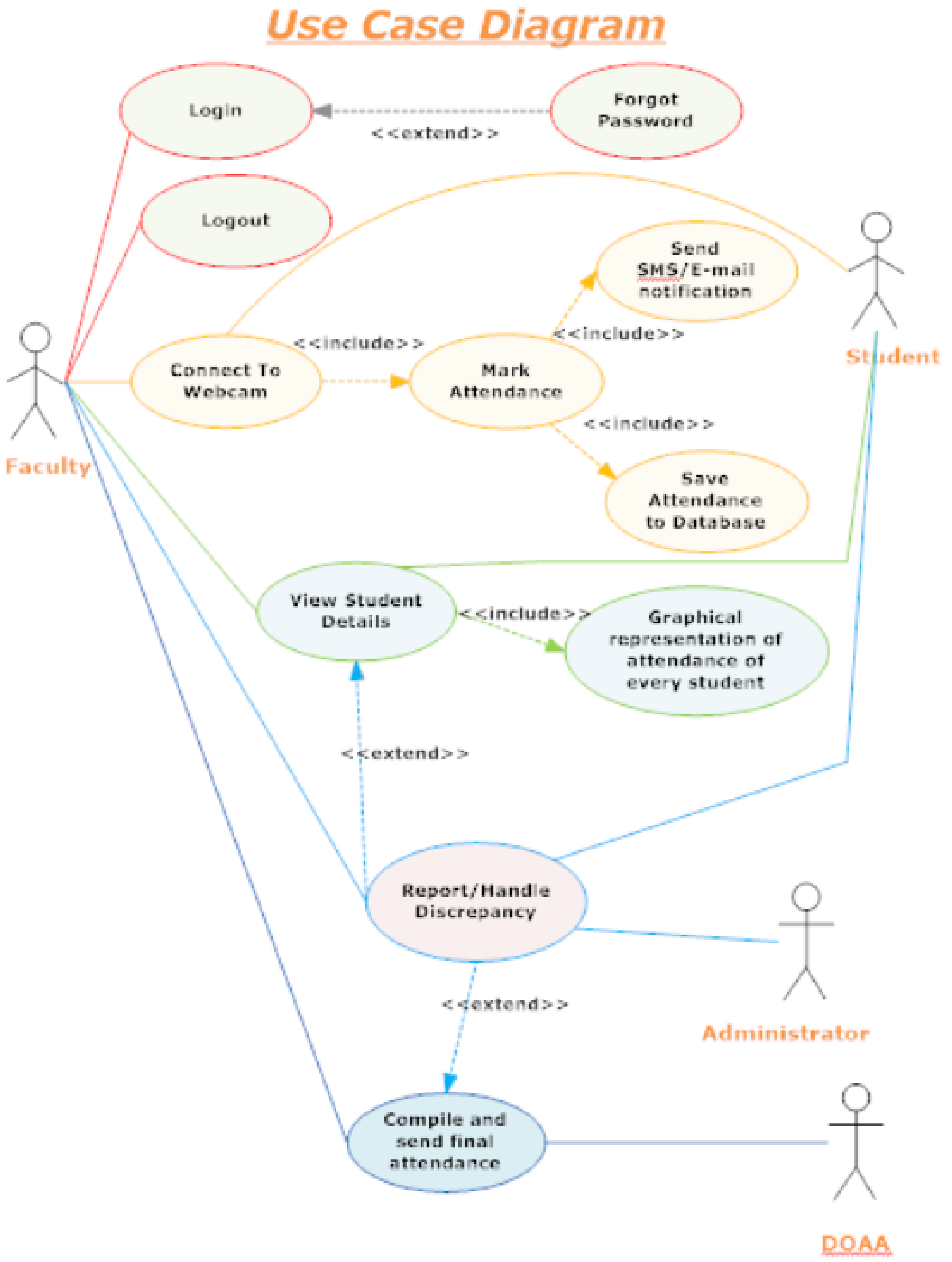
Purpose	The screen provides information specific to each staff member.
Inputs	A Teacher needs to login with their unique id and password for launching the Attendance using face recognition system software.
Processing	Teachers login credentials will be checked with the database values. If verified then system software starts working.
Outputs	Camera start working and now students can mark attendance.

3.3 Performance Requirements

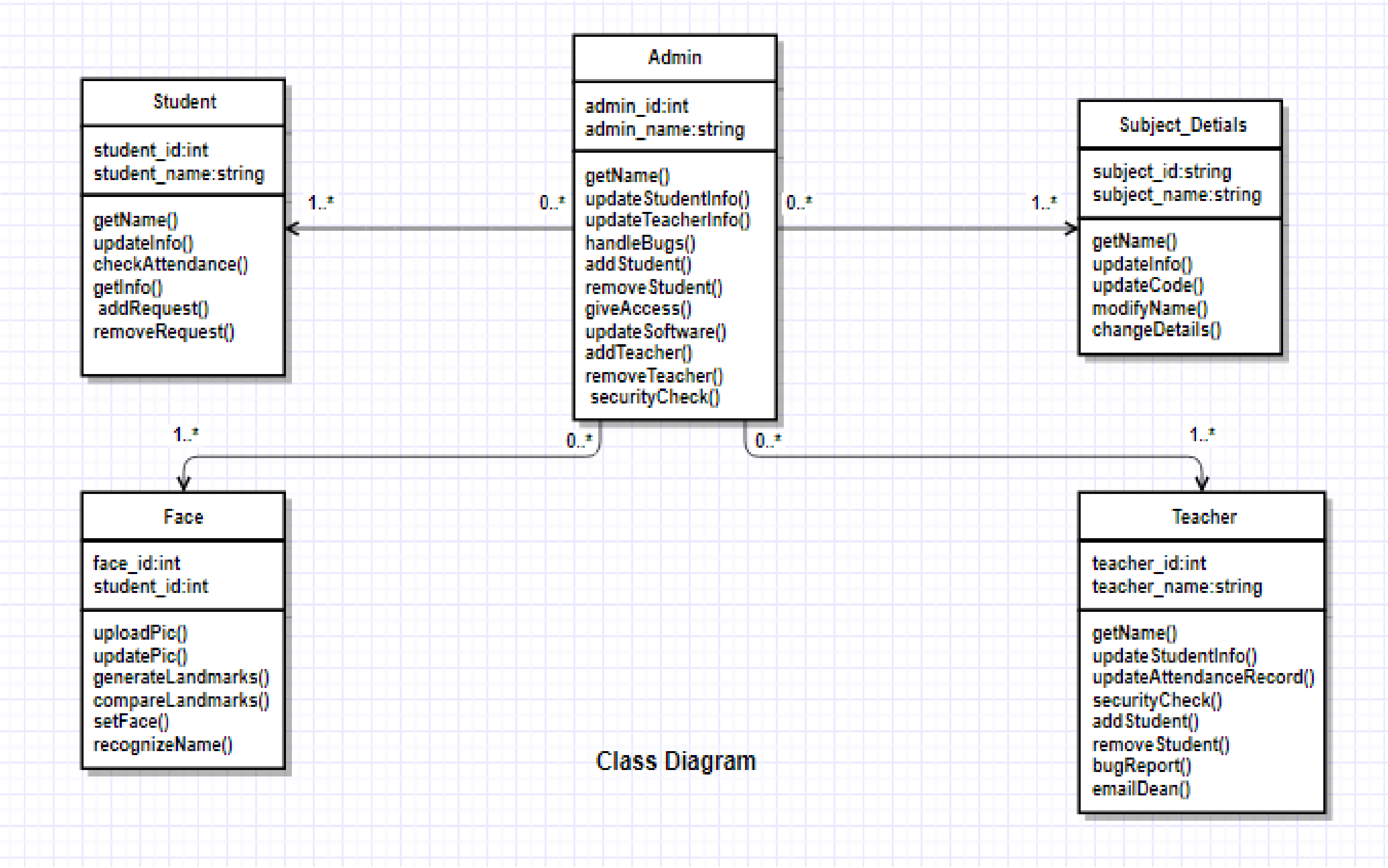
The software is designed for the attendance using facial recognition and can run from a standalone desktop PC. The software will support multiple user (staff members) access only if there are multiple terminals. Database will be handled by the Software. For normal conditions, 95% of the image processing should be processed in less than 5 seconds.

3.4 Diagrams

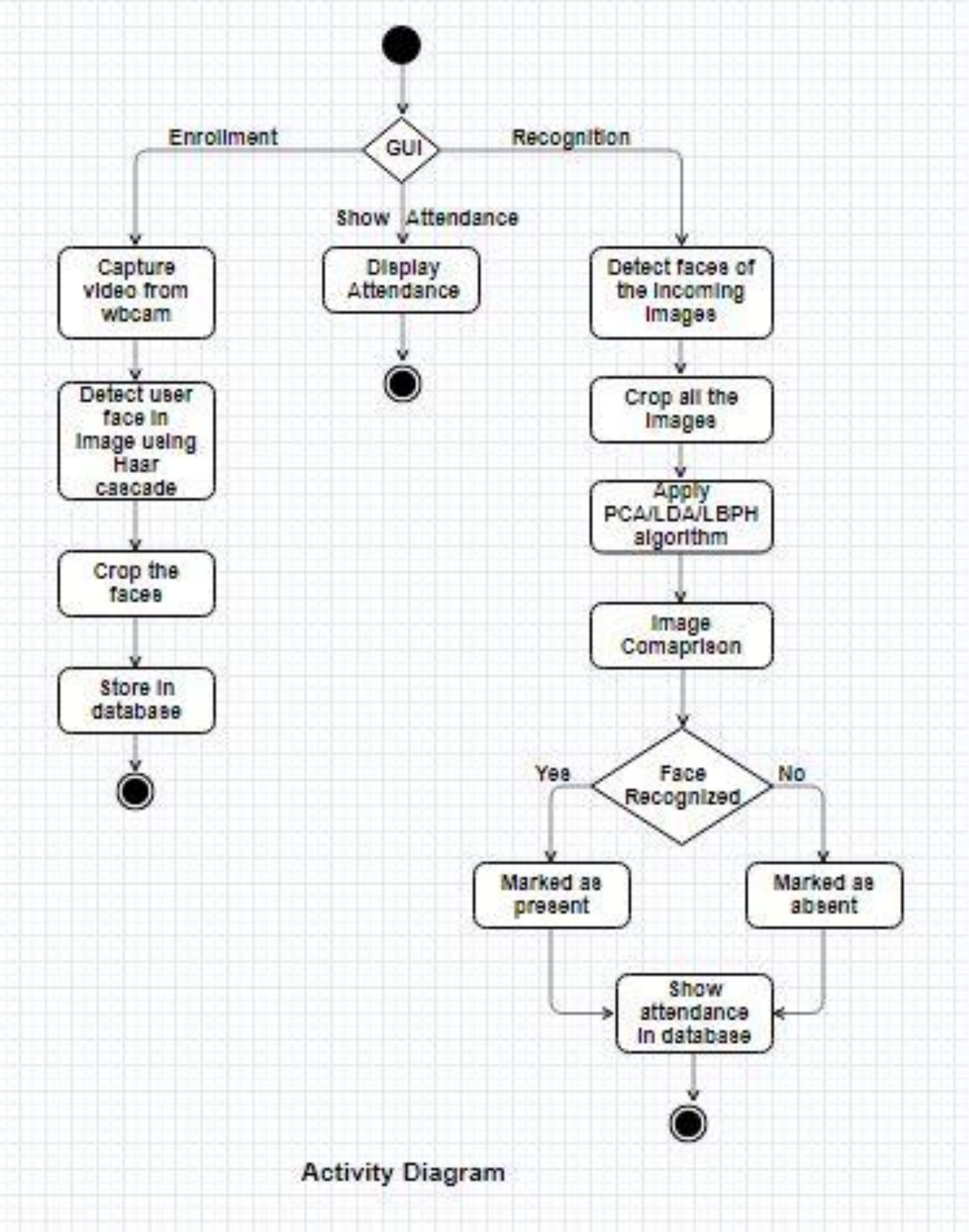
The figure shows the Use Case Diagram for the entire system



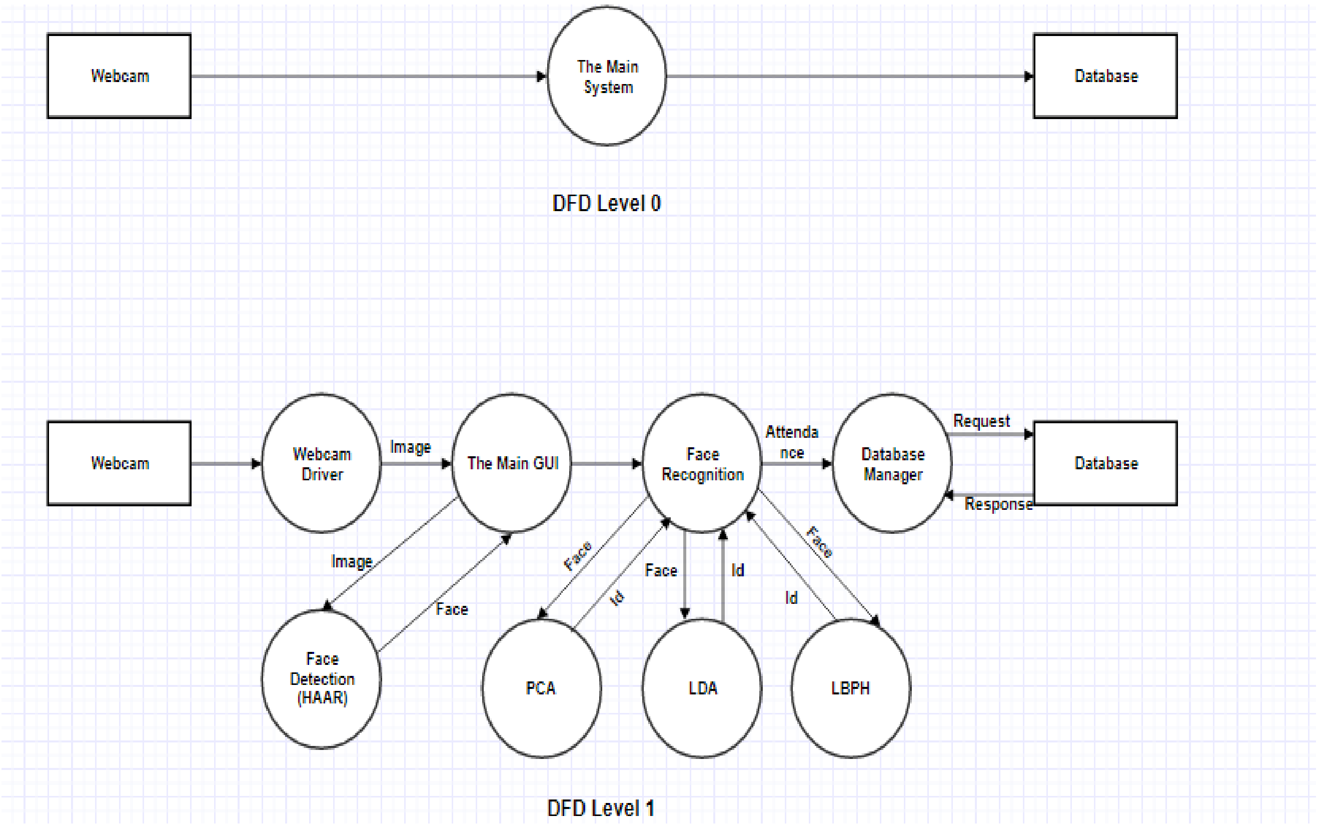
Class Diagram



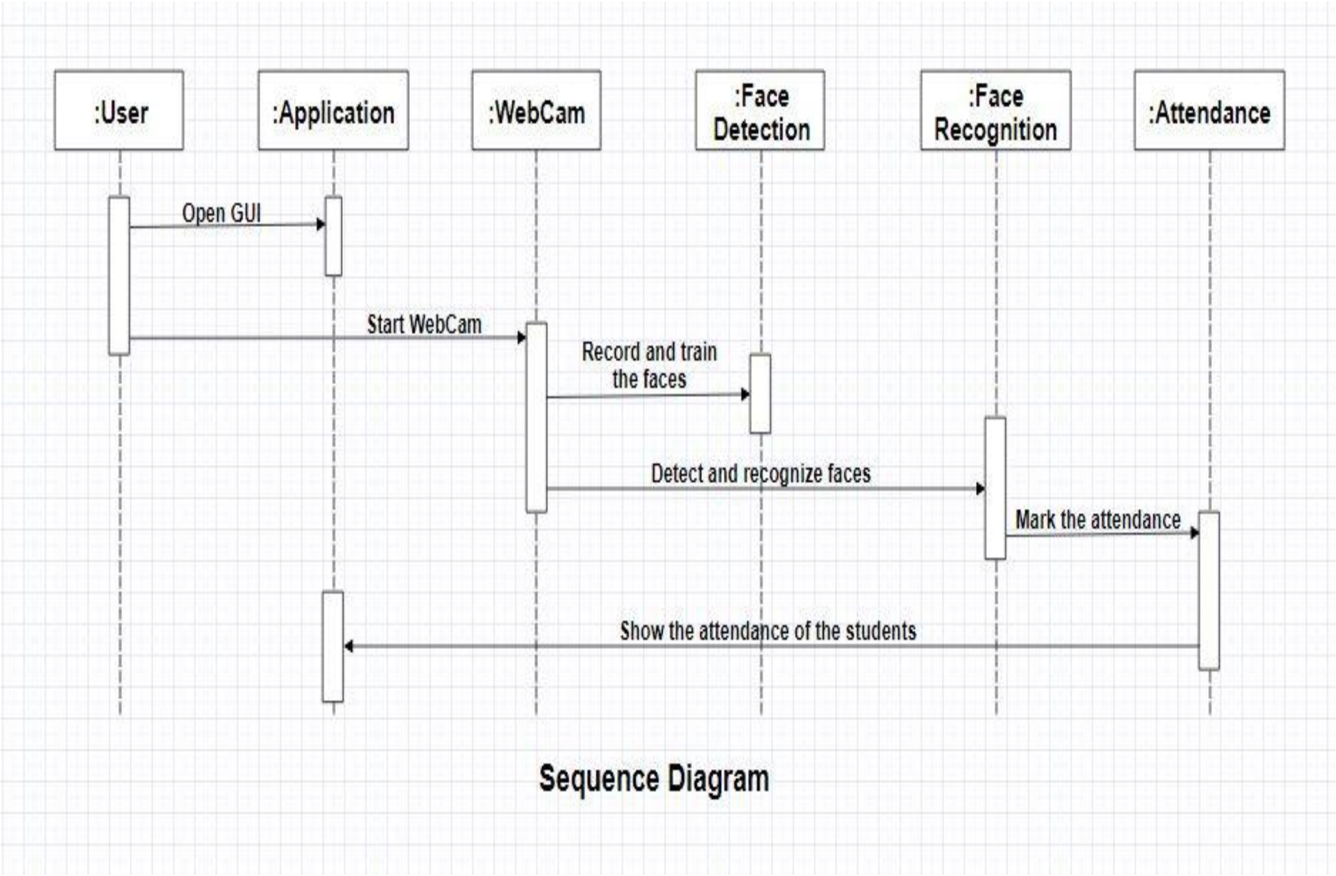
Activity Diagram



Data Flow Diagram



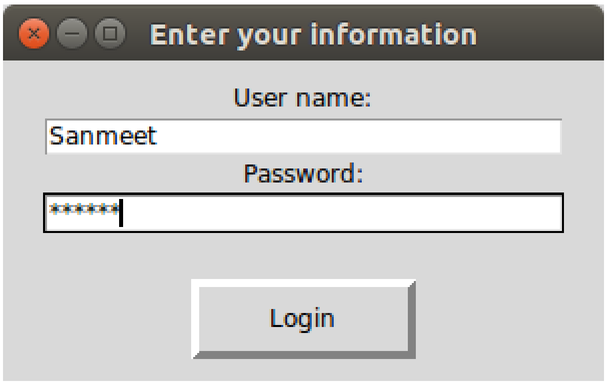
Sequence Diagram



Screenshots

Graphical User Interface

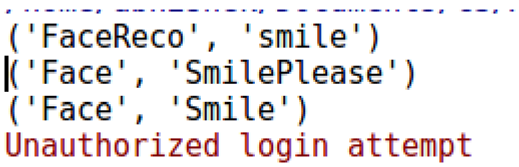
Screen 1



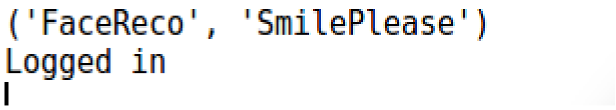
User name: Sanmeet

Password: ******

Results

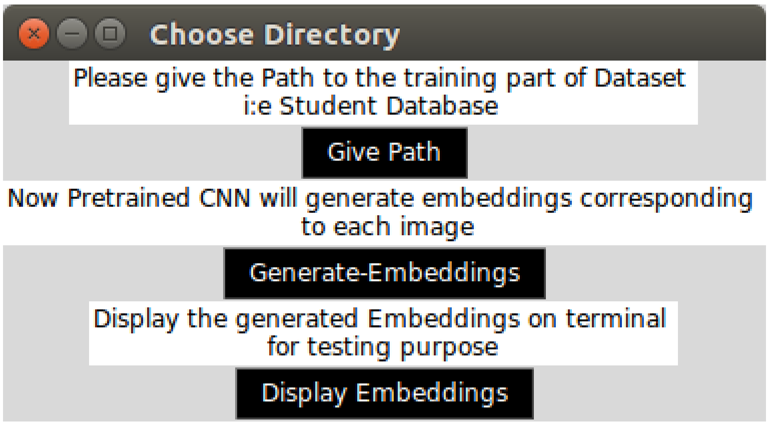


```
('FaceReco', 'smile')
('Face', 'SmilePlease')
('Face', 'Smile')
Unauthorized login attempt
```



```
('FaceReco', 'SmilePlease')
Logged in
```

Screen 2

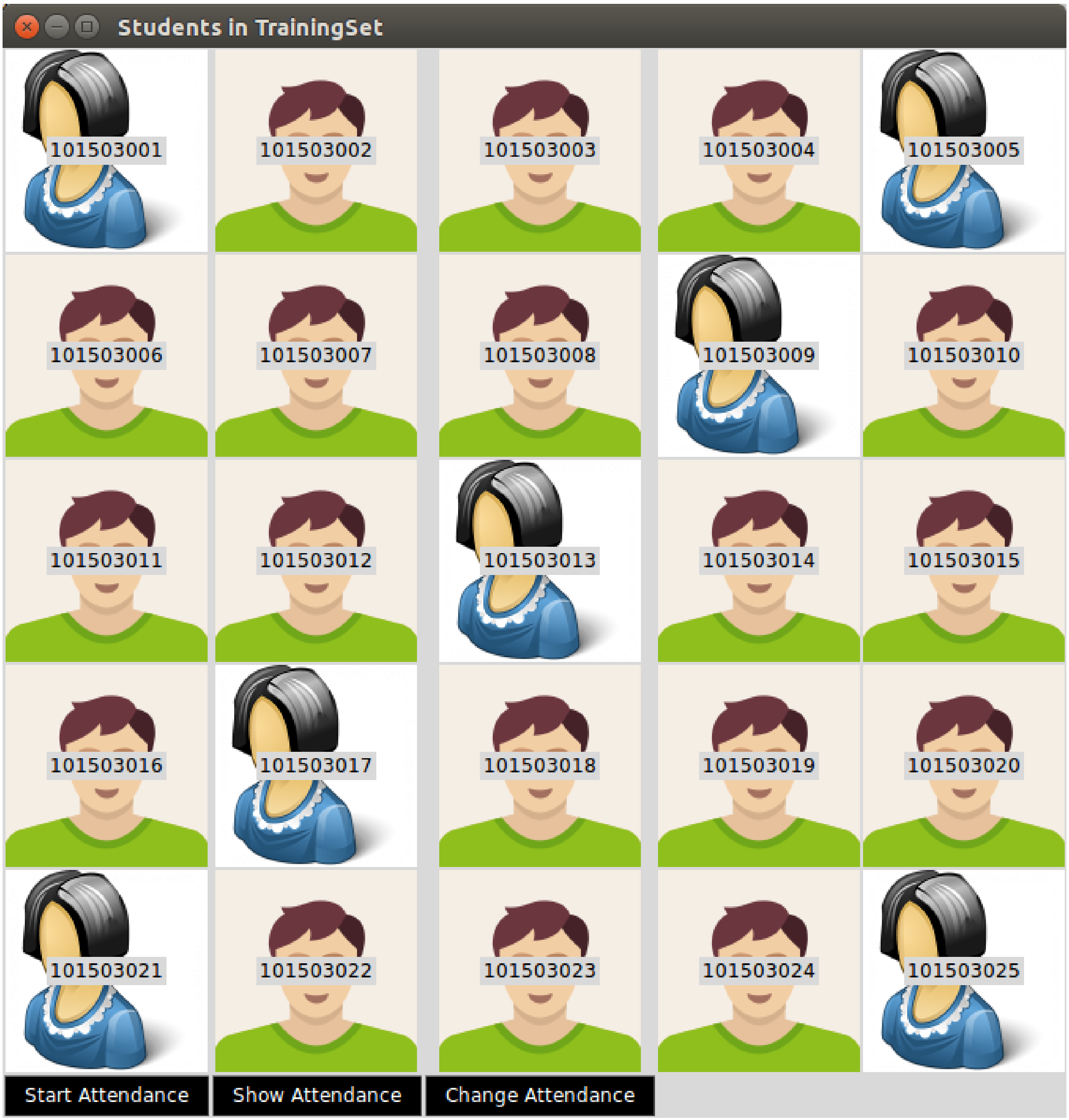


Please give the Path to the training part of Dataset i:e Student Database

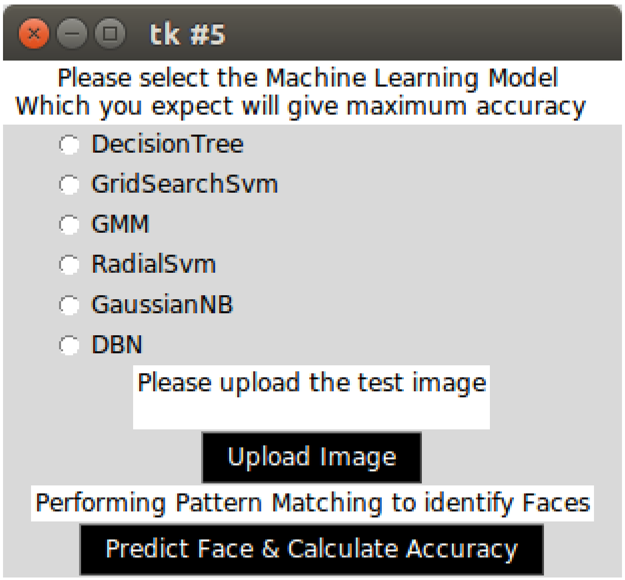
Now Pretrained CNN will generate embeddings corresponding to each image

Display the generated Embeddings on terminal for testing purpose

Screen 3


Students in TrainingSet
101503001
101503002
101503003
101503004
101503005
101503006
101503007
101503008
101503009
101503010
101503011
101503012
101503013
101503014
101503015
101503016
101503017
101503018
101503019
101503020
101503021
101503022
101503023
101503024
101503025
Start Attendance
Show Attendance
Change Attendance

Screen 4



Please select the Machine Learning Model
Which you expect will give maximum accuracy

- ○ DecisionTree
- ○ GridSearchSvm
- ○ GMM
- ○ RadialSvm
- ○ GaussianNB
- ○ DBN

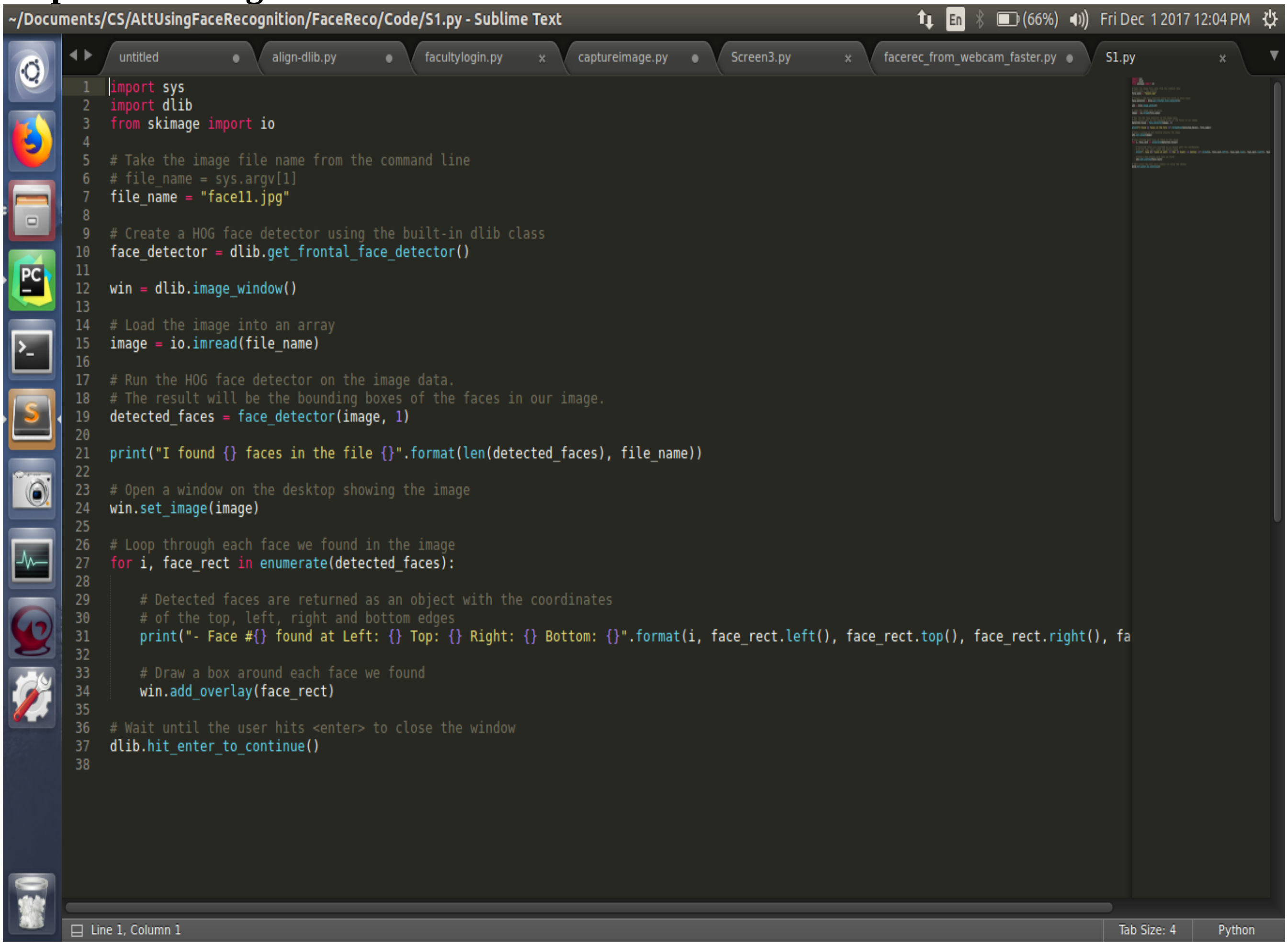
Please upload the test image

Performing Pattern Matching to identify Faces

Code Snippets

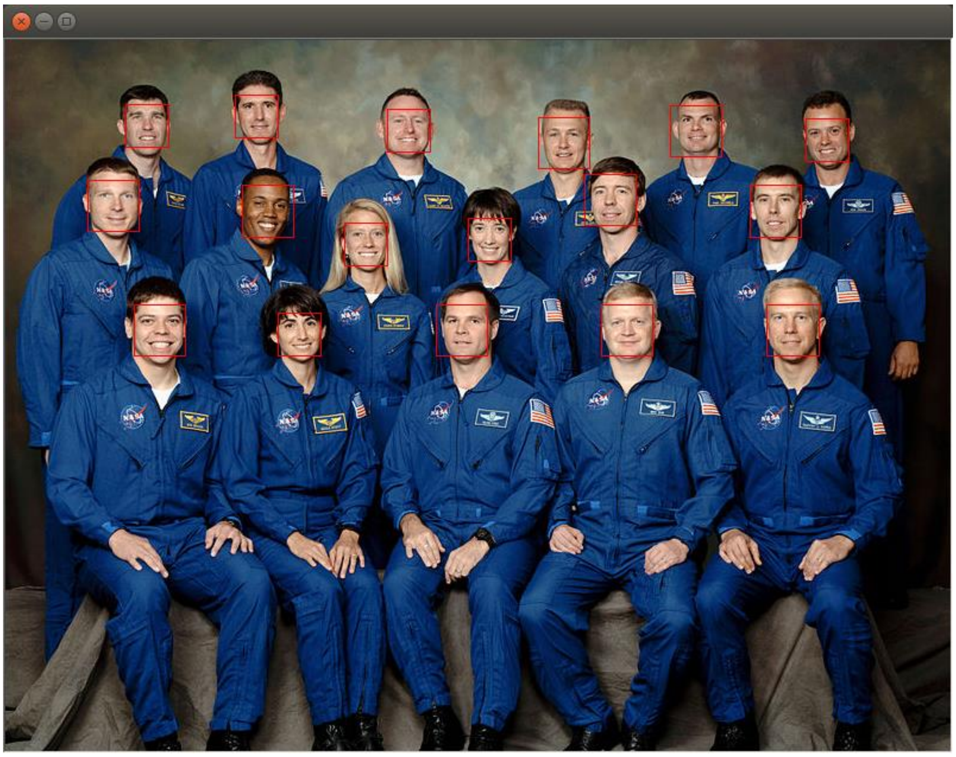
Test Image

Step 1: Detecting Faces



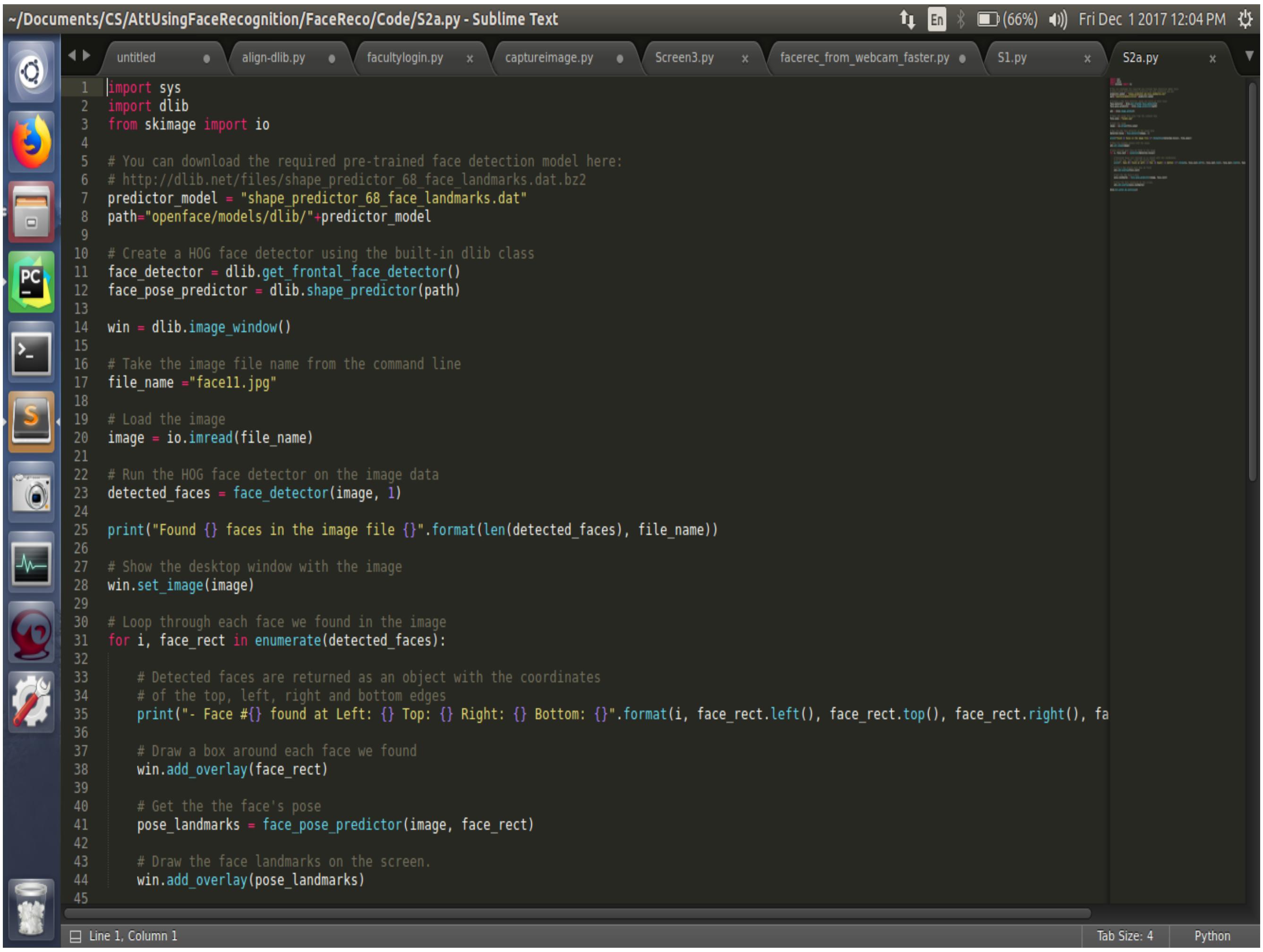
```python
import sys
import dlib
from skimage import io

# Take the image file name from the command line
# file_name = sys.argv[1]
file_name = "face11.jpg"

# Create a HOG face detector using the built-in dlib class
face_detector = dlib.get_frontal_face_detector()

win = dlib.image_window()

# Load the image into an array
image = io.imread(file_name)

# Run the HOG face detector on the image data.
# The result will be the bounding boxes of the faces in our image.
detected_faces = face_detector(image, 1)

print("I found {} faces in the file {}".format(len(detected_faces), file_name))

# Open a window on the desktop showing the image
win.set_image(image)

# Loop through each face we found in the image
for i, face_rect in enumerate(detected_faces):

    # Detected faces are returned as an object with the coordinates
    # of the top, left, right and bottom edges
    print("- Face #{} found at Left: {} Top: {} Right: {} Bottom: {}".format(i, face_rect.left(), face_rect.top(), face_rect.right(), fa

    # Draw a box around each face we found
    win.add_overlay(face_rect)

# Wait until the user hits <enter> to close the window
dlib.hit_enter_to_continue()
```

Result of above Code:



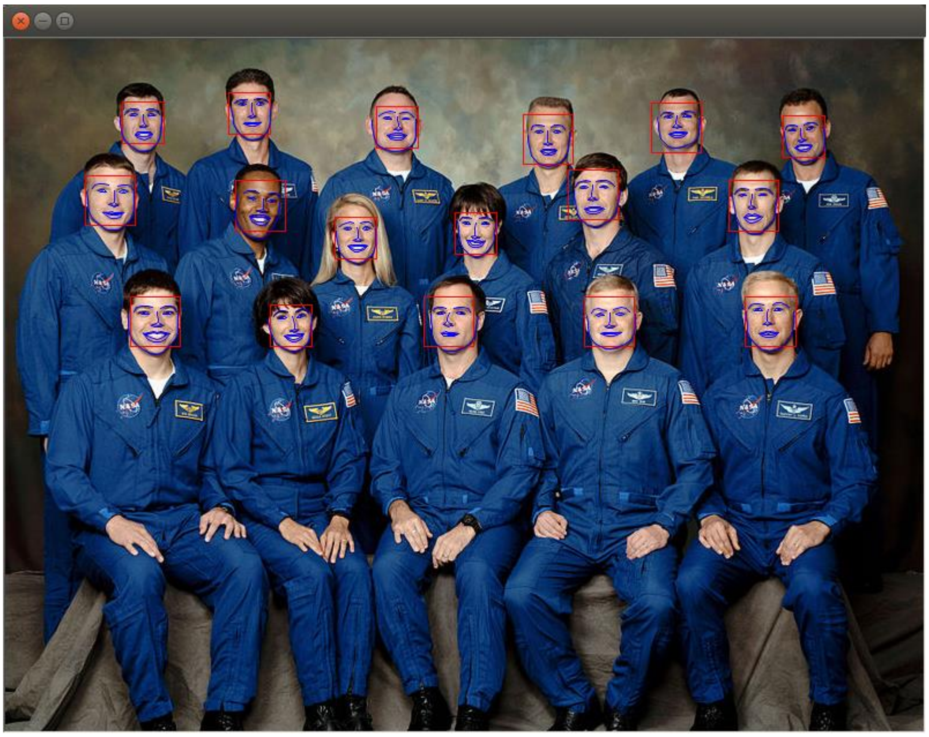
Step 2: Posing and Projecting Faces

a:

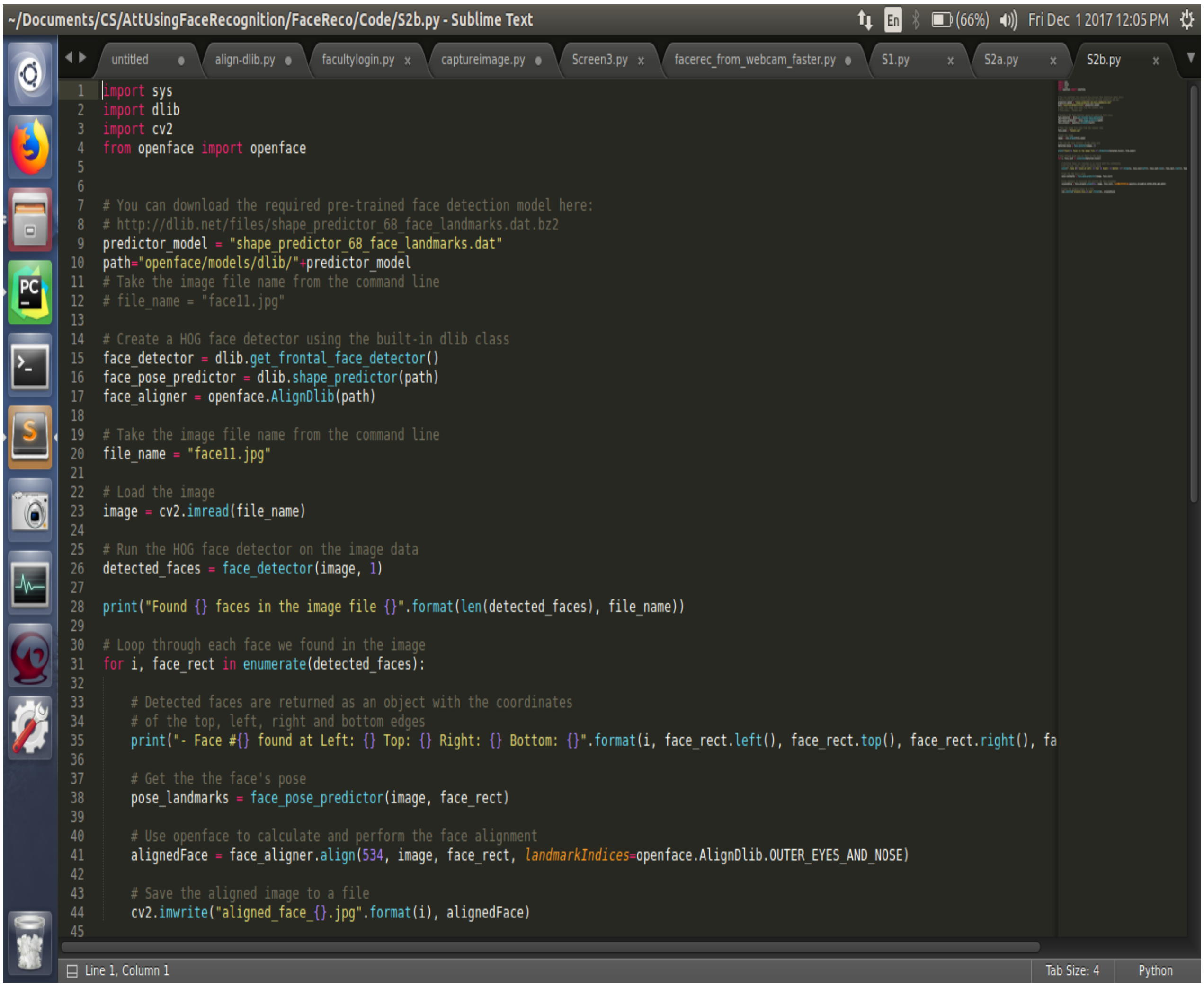


```python
import sys
import dlib
from skimage import io

# You can download the required pre-trained face detection model here:
# http://dlib.net/files/shape_predictor_68_face_landmarks.dat.bz2
predictor_model = "shape_predictor_68_face_landmarks.dat"
path="openface/models/dlib/"+predictor_model

# Create a HOG face detector using the built-in dlib class
face_detector = dlib.get_frontal_face_detector()
face_pose_predictor = dlib.shape_predictor(path)

win = dlib.image_window()

# Take the image file name from the command line
file_name ="face11.jpg"

# Load the image
image = io.imread(file_name)

# Run the HOG face detector on the image data
detected_faces = face_detector(image, 1)

print("Found {} faces in the image file {}".format(len(detected_faces), file_name))

# Show the desktop window with the image
win.set_image(image)

# Loop through each face we found in the image
for i, face_rect in enumerate(detected_faces):

    # Detected faces are returned as an object with the coordinates
    # of the top, left, right and bottom edges
    print("- Face #{} found at Left: {} Top: {} Right: {} Bottom: {}".format(i, face_rect.left(), face_rect.top(), face_rect.right(), fa

    # Draw a box around each face we found
    win.add_overlay(face_rect)

    # Get the the face's pose
    pose_landmarks = face_pose_predictor(image, face_rect)

    # Draw the face landmarks on the screen.
    win.add_overlay(pose_landmarks)

```

Result:

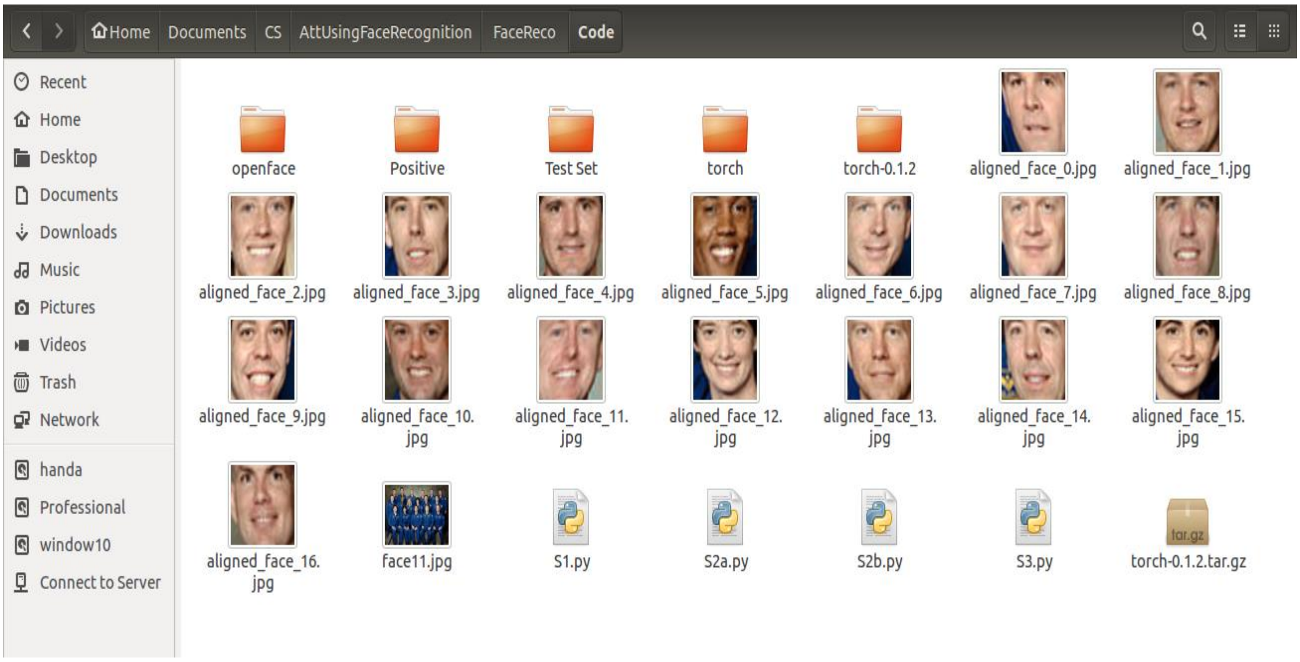


b:

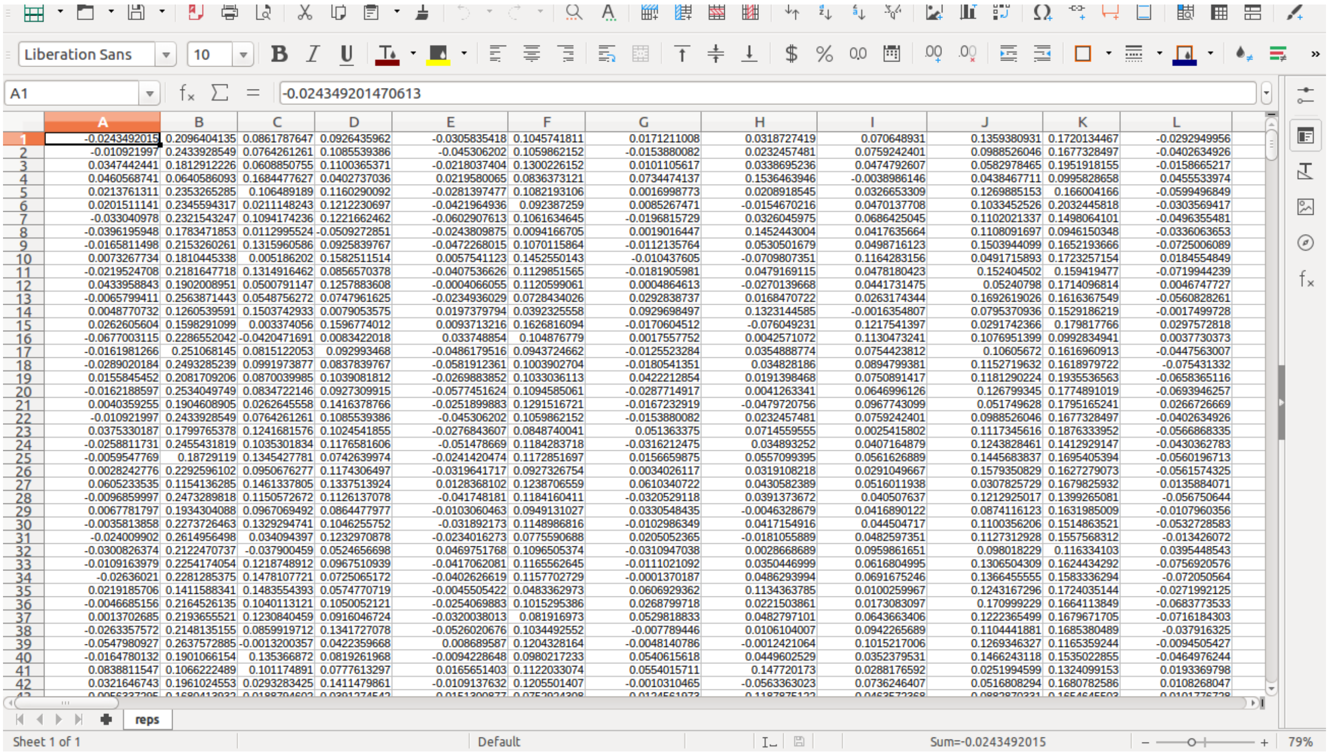


```python
import sys
import dlib
import cv2
from openface import openface


# You can download the required pre-trained face detection model here:
# http://dlib.net/files/shape_predictor_68_face_landmarks.dat.bz2
predictor_model = "shape_predictor_68_face_landmarks.dat"
path="openface/models/dlib/"+predictor_model
# Take the image file name from the command line
# file_name = "face11.jpg"

# Create a HOG face detector using the built-in dlib class
face_detector = dlib.get_frontal_face_detector()
face_pose_predictor = dlib.shape_predictor(path)
face_aligner = openface.AlignDlib(path)

# Take the image file name from the command line
file_name = "face11.jpg"

# Load the image
image = cv2.imread(file_name)

# Run the HOG face detector on the image data
detected_faces = face_detector(image, 1)

print("Found {} faces in the image file {}".format(len(detected_faces), file_name))

# Loop through each face we found in the image
for i, face_rect in enumerate(detected_faces):

    # Detected faces are returned as an object with the coordinates
    # of the top, left, right and bottom edges
    print("- Face #{} found at Left: {} Top: {} Right: {} Bottom: {}".format(i, face_rect.left(), face_rect.top(), face_rect.right(), fa

    # Get the the face's pose
    pose_landmarks = face_pose_predictor(image, face_rect)

    # Use openface to calculate and perform the face alignment
    alignedFace = face_aligner.align(534, image, face_rect, landmarkIndices=openface.AlignDlib.OUTER_EYES_AND_NOSE)

    # Save the aligned image to a file
    cv2.imwrite("aligned_face_{}.jpg".format(i), alignedFace)

```

Result: All the images are now splitted into separate files

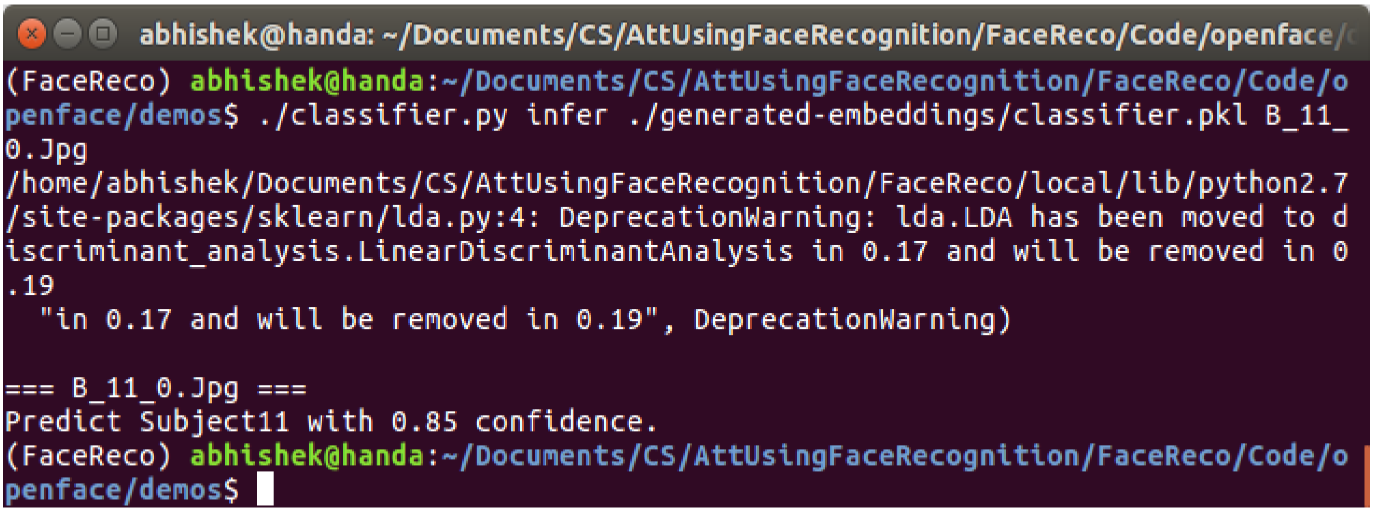


Step 3: Encoding Faces: 128 measurements generated corresponding to each face in the dataset



	A	B	C	D	E	F	G	H	I	J	K	L
1	-0.0243492015	0.2096404135	0.0861787647	0.0926435962	-0.0305835418	0.1045741811	0.0171211008	0.0318727419	0.070648931	0.1359380931	0.1720134467	-0.0292949956
2	-0.010921997	0.2433928549	0.0764261261	0.1085539386	-0.045306202	0.1059862152	-0.0153880082	0.0232457481	0.0759242401	0.0988526046	0.1677328497	-0.0402634926
3	0.0347442441	0.1812912226	0.0608850755	0.1100365371	-0.0218037404	0.1300226152	0.0101105617	0.0338695236	0.0474792607	0.0582978465	0.1951918155	-0.0158665217
4	0.0460568741	0.0640586093	0.1684477627	0.0402737036	0.0219580065	0.0836373121	0.0734474137	0.1536463946	-0.0038986146	0.0438467711	0.0995828658	0.0455533974
5	0.0213761311	0.2353265285	0.106489189	0.1160290092	-0.0281397477	0.1082193106	0.0016998773	0.0208918545	0.0326653309	0.1269885153	0.166004166	-0.0599496849
6	0.0201511141	0.2345594317	0.0211148243	0.1212230697	-0.0421964936	0.092387259	0.0085267471	-0.0154670216	0.0470137708	0.1033452526	0.2032445818	-0.0303569417
7	-0.033040978	0.2321543247	0.1094174236	0.1221662462	-0.0602907613	0.1061634645	-0.0196815729	0.0326045975	0.0686425045	0.1102021337	0.1498064101	-0.0496355481
8	-0.0396195948	0.1783471853	0.0112995524	-0.0509272851	-0.0243809875	0.0094166705	0.0019016447	0.1452443004	0.0417635664	0.1108091697	0.0946150348	-0.0336063653
9	-0.0165811498	0.2153260261	0.1315960586	0.0925839767	-0.0472268015	0.1070115864	-0.0112135764	0.0530501679	0.0498716123	0.1503944099	0.1652193666	-0.0725006089
10	0.0073267734	0.1810445338	0.005186202	0.1582511514	0.0057541123	0.1452550143	-0.010437605	-0.0709807351	0.1164283156	0.0491715893	0.1723257154	0.0184554849
11	-0.0219524708	0.2181647718	0.1314916462	0.0856570378	-0.0407536626	0.1129851565	-0.0181905981	0.0479169115	0.0478180423	0.152404502	0.159419477	-0.0719944239
12	0.0433958843	0.1902008951	0.0500791147	0.1257883608	-0.0004066055	0.1120599061	0.0004864613	-0.0270139668	0.0441731475	0.05240798	0.1714096814	0.0046747727
13	-0.0065799411	0.2563871443	0.0548756272	0.0747961625	-0.0234936029	0.0728434026	0.0292838737	0.0168470722	0.0263174344	0.1692619026	0.1616367549	-0.0560828261
14	0.0048770732	0.1260539591	0.1503742933	0.0079053575	0.0197379794	0.0392325558	0.0929698497	0.1323144585	-0.0016354807	0.0795370936	0.1529186219	-0.0017499728
15	0.0262605604	0.1598291099	0.003374056	0.1596774012	0.0093713216	0.1626816094	-0.0170604512	-0.076049231	0.1217541397	0.0291742366	0.179817766	0.0297572818
16	-0.0677003115	0.2286552042	-0.0420471691	0.0083422018	0.033748854	0.104876779	0.0017557752	0.0042571072	0.1130473241	0.1076951399	0.0992834941	0.0037730373
17	-0.0161981266	0.251068145	0.0815122053	0.092993468	-0.0486179516	0.0943724662	-0.0125523284	0.0354888774	0.0754423812	0.10605672	0.1616960913	-0.0447563007
18	-0.0289020184	0.2493285239	0.0991973877	0.0837839767	-0.0581912361	0.1003902704	-0.0180541351	0.034828186	0.0894799381	0.1152719632	0.1618979722	-0.075431332
19	0.0155845452	0.2081709206	0.0870039985	0.1039081812	-0.0269883852	0.1033036113	0.0422212854	0.0191398468	0.0750891417	0.1181290224	0.1935536563	-0.0658365116
20	-0.0162188597	0.2534049749	0.0834722146	0.0927309915	-0.0577451624	0.1094585061	-0.0287714917	0.0041263341	0.0646996126	0.126799345	0.1774891019	-0.0693946257
21	0.0040359255	0.1904608905	0.0262645558	0.1416378766	-0.0251899883	0.1291516721	-0.0167232919	-0.0479720756	0.0967743099	0.051749628	0.1795165241	0.0266726669
22	-0.010921997	0.2433928549	0.0764261261	0.1085539386	-0.045306202	0.1059862152	-0.0153880082	0.0232457481	0.0759242401	0.0988526046	0.1677328497	-0.0402634926
23	0.0375330187	0.1799765378	0.1241681576	0.1024541855	-0.0276843607	0.0848740041	0.051363375	0.0714559555	0.0025415802	0.1117345616	0.1876333952	-0.0566868335
24	-0.0258811731	0.2455431819	0.1035301834	0.1176581606	-0.051478669	0.1184283718	-0.0316212475	0.034893252	0.0407164879	0.1243828461	0.1412929147	-0.0430362783
25	-0.0059547769	0.18729119	0.1345427781	0.0742639974	-0.0241420474	0.1172851697	0.0156659875	0.0557099395	0.0561626889	0.1445683837	0.1695405394	-0.0560196713
26	0.0028242776	0.2292596102	0.0950676277	0.1174306497	-0.0319641717	0.0927326754	0.0034026117	0.0319108218	0.0291049667	0.1579350829	0.1627279073	-0.0561574325
27	0.0605233535	0.1154136285	0.1461337805	0.1337513924	0.0128368102	0.1238706559	0.0610340722	0.0430582389	0.0516011938	0.0307825729	0.1679825932	0.0135884071
28	-0.0096859997	0.2473289818	0.1150572672	0.1126137078	-0.041748181	0.1184160411	-0.0320529118	0.0391373672	0.040507637	0.1212925017	0.1399265081	-0.056750644
29	0.0067781797	0.1934304088	0.0967069492	0.0864477977	-0.0103060463	0.0949131027	0.0330548435	-0.0046328679	0.0416890122	0.0874116123	0.1631985009	-0.0107960356
30	-0.0035813858	0.2273726463	0.1329294741	0.1046255752	-0.031892173	0.1148986816	-0.0102986349	0.0417154916	0.044504717	0.1100356206	0.1514863521	-0.0532728583
31	-0.024009902	0.2614956498	0.034094397	0.1232970878	-0.0234016273	0.0775590688	0.0205052365	-0.0181055889	0.0482597351	0.1127312928	0.1557568312	-0.013426072
32	-0.0300826374	0.2122470737	-0.037900459	0.0524656698	0.0469751768	0.1096505374	-0.0310947038	0.0028668689	0.0959861651	0.098018229	0.116334103	0.0395448543
33	-0.0109163979	0.2254174054	0.1218748912	0.0967510939	-0.0417062081	0.1165562645	-0.0111021092	0.0350446999	0.0616804995	0.1306504309	0.1624434292	-0.0756920576
34	-0.02636021	0.2281285375	0.1478107721	0.0725065172	-0.0402626619	0.1157702729	-0.0001370187	0.0486293994	0.0691675246	0.1366455555	0.1583336294	-0.072050564
35	0.0219185706	0.1411588341	0.1483554393	0.0574770719	-0.0045505422	0.0483362973	0.0606929362	0.1134363785	0.0100259967	0.1243167296	0.1724035144	-0.0271992125
36	-0.0046685156	0.2164526135	0.1040113121	0.1050052121	-0.0254069883	0.1015295386	0.0268799718	0.0221503861	0.0173083097	0.170999229	0.1664113849	-0.0683773533
37	0.0013702685	0.2193655521	0.1230840459	0.0916046724	-0.0320038013	0.081916973	0.0529818833	0.0482797101	0.0643663406	0.1222365499	0.1679671705	-0.0716184303
38	-0.0263357572	0.2148135155	0.0859919712	0.1341727078	-0.0526020676	0.1034492552	-0.007789446	0.0106104007	0.0942265689	0.1104441881	0.1685380489	-0.037916325
39	-0.0547980927	0.2637572885	-0.0013200357	0.0422359668	0.008689587	0.1204328164	-0.0048140786	-0.0012421064	0.1015217006	0.1269346327	0.1165359244	-0.0094505427
40	-0.0164780132	0.1901066154	0.135366872	0.0819261968	-0.0094228648	0.0980217233	0.0540615618	0.0449602529	0.0352379531	0.1466243118	0.1535022855	-0.0464976244
41	0.0838811547	0.1066222489	0.101174891	0.0777613297	0.0165651403	0.1122033074	0.0554015711	0.147720173	0.0288176592	0.0251994599	0.1324099153	0.0193369798
42	0.0321646743	0.1961024553	0.0293283425	0.1411479861	-0.0109137632	0.1205501407	-0.0010310465	-0.0563363023	0.0736246407	0.0516808294	0.1680782586	0.0108268047

Step 4: Finding and recognizing Name using encoding



```
abhishek@handa: ~/Documents/CS/AttUsingFaceRecognition/FaceReco/Code/openface/
(FaceReco) abhishek@handa:~/Documents/CS/AttUsingFaceRecognition/FaceReco/Code/o
penface/demos$ ./classifier.py infer ./generated-embeddings/classifier.pkl B_11_
0.Jpg
/home/abhishek/Documents/CS/AttUsingFaceRecognition/FaceReco/local/lib/python2.7
/site-packages/sklearn/lda.py:4: DeprecationWarning: lda.LDA has been moved to d
iscriminant_analysis.LinearDiscriminantAnalysis in 0.17 and will be removed in 0
.19
  "in 0.17 and will be removed in 0.19", DeprecationWarning)

=== B_11_0.Jpg ===
Predict Subject11 with 0.85 confidence.
(FaceReco) abhishek@handa:~/Documents/CS/AttUsingFaceRecognition/FaceReco/Code/o
penface/demos$ 
```

Subject11 with 85% accuracy is predicted by our Model

3.5 Quality Attribute

The software is targeted towards a wide variety of users such as Student, staff, dean, etc. The product must load quickly and work well on a variety of terminals. It must also tolerate wide variety of input possibilities from a user, such as incorrect responses or unforeseen keystrokes.

3.6 Other Requirements

None at this time

4. Change History

271017	Version 1.0 – Initial Release
241117	Version 1.1 – Second Release

5. Document Approvers

SRS for Attendance using face recognition system is approved by:

(Dr. Sanmeet Bhatia)

(Assistant Professor)

(24/11/2017)