Homework 3
CSE 402: Biometrics and Pattern Recognition
Instructor: Dr. Arun Ross
Due Date: November 21, 2022 (11:00pm)
Total Points: 60

Note:

. You are permitted to discuss the following questions with others in the class.
. However, you must write up your own answers to these questions. Any indication to the contrary will be considered an act of academic dishonesty.
. A neatly typed report with detailed answers is expected. The report must be uploaded in D2L in PDF format.
. All outputs, such as graphs and images, must be included in the report.
. Any code developed as part of the assignment must be (a) included as an appendix in the report, as well as (b) archived in a single zip file and uploaded in D2L.
. Include a bibliography at the end of the report indicating the resources that you used (e.g., URL, scientific articles, books, etc.) to complete this homework.
. Please submit the report (PDF) and the code (Zip file) as two separate files in D2L.

1. [15 points] Consider a set of 1000 2-dimensional points here. Use Matlab (or any other software) to perform the following tasks:

(a) Compute and report the mean vector of these points.

Ans)

```
Mean Vector = [19.973930000000006, 19.935589999999998]
```

(b) Compute and report the covariance matrix of these points.

Ans)

```
Covariance matrix =  [[ 9.38948766  5.11930953]
 [ 5.11930953 10.37436205]]
```

Covariance matrix using PCA setup:

```
Covariance matrix_pca =  [[ 9389.4876551  5119.3095313]
 [ 5119.3095313 10374.3620519]]
```

(c) Compute and report the eigen-vectors and eigen-values of the covariance matrix. There should be two eigen-vectors and two eigen-values.

Ans)

```
Eigen Values = [ 4.73898557 15.02486414]
```

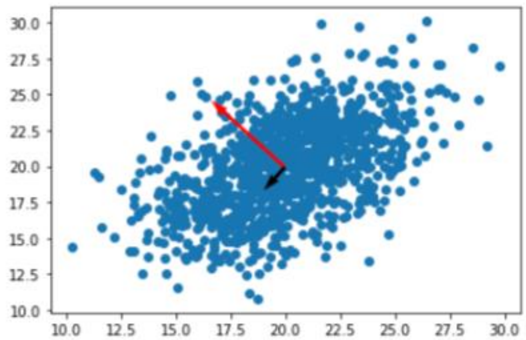
```
Eigen Vectors = [[-0.74018584 -0.67240235]
 [ 0.67240235 -0.74018584]]
```

eigen-vectors and eigen-values using PCA setup:

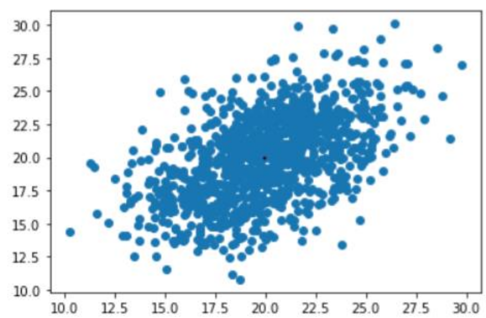
```
Eigen Values_pca = [ 4738.98557021 15024.86413679]

Eigen Vectors_pca = [[-0.74018584 -0.67240235]
 [ 0.67240235 -0.74018584]]
```

(d) Plot the eigen-vectors (along with the 1000 points) in a 2D graph and display the graph. Note that the eigen-vectors will have to originate from the mean of the points. The length of each vector should be in proportion to its corresponding eigen-value.

Ans)



Graph in PCA setup:



2. [15 points] One of the model-based face recognition methods described in the literature is Elastic Bunch Graph Matching (EBGM). The EBGM technique is discussed in pages 122 - 124 of the textbook. More details about this algorithm can also be found here. In the context of the EBGM algorithm, answer the following questions:

(a) What are fiducial points?

Ans) Fiducial points are also known as landmark points these points denote the corner of eyes, tip of the nose, corners of the mouth, homogeneous regions of the face and the chin.

- The use of fiducial points enables plotting of partial graph even if the face is tilted or occluded.

(b) What are Gabor Jets?

Ans) Gabor coefficients or jets characterize the local texture information around the fiducial or landmark point, the edge connecting any two nodes of the graph is labelled based on the distance between the corresponding fiducial points.

- The Gabor coefficient at a location in the image can be obtained by convolving the image with a complex 2D Gabor filter centered at that location.

- You can change Gabor jet or a set of coefficients by changing the orientation and frequency of the Gabor filters.

(c) What is a Face Bunch Graph (FBG)?

Ans) Face Bunch Graph (FBG) can be constructed in two stages from a training set of face images with a specific pose:

1. First stage is to manually mark the fiducial or landmark points and define the geometric structure of the image graph for one or few initial images.

2. In the second stage Face Bunch Graph is obtained from the individual image graphs by combining a representative set of individual graphs in a stack like structure.

(d) How are two face images compared using the EBGM algorithm?

Ans) Following steps show how two face images are compared using EBGM algorithm:

- Firstly, image graphs are computed for both the images (the graph corresponding to the gallery image is also sometimes refered to as model graph).
- The similarity of these image graphs is computed as the average similarity between the jets at the corresponding fiducial points.
- Since we know the fiducial points and their correspondence the two graphs can be matched successfully even with some missing nodes.

(e) What do you think are some of the limitations of the EBGM method for face recognition?

Ans) Some of the limitations of the EBGM method for face recognition are:

- It can only be applied to objects that have similar structure like frontal images that share common set of fiducial points.
- The graphs must be dynamic regarding both shape and attribute characteristics for arbitrary objects in the absence of fiducial points.

3. [10 points] Describe the three different levels at which facial characteristics can be organized from a biometrics perspective. Explain the types of features in each level and the role of these features in face recognition.

Ans) The three different levels are as follows:

1. Level 1 details consist of gross facial characteristics that are easily observable. Examples include the general geometry of the face and global skin colour. Such features can be used to quickly discriminate between (a) a short round face and an elongated thin face; (b) faces exhibiting predominantly male and female characteristics or (c) faces from different races. These features can be extracted even from low resolution face images.
2. Level 2 details consist of localized face information such as the structure of the face components (e.g., eyes), the relationship between facial components, and the precise shape of the face. These features are essential for accurate face recognition, and they require a higher resolution face image (30 to 75 IPD). The characteristics of local regions of the face can be represented using geometric or texture descriptors.
3. Level 3 details consist of unstructured, micro level features on the face, which includes scars, freckles, skin discoloration, and moles. One challenging face recognition problem where Level 3 details may be critical is the discrimination of identical twins.

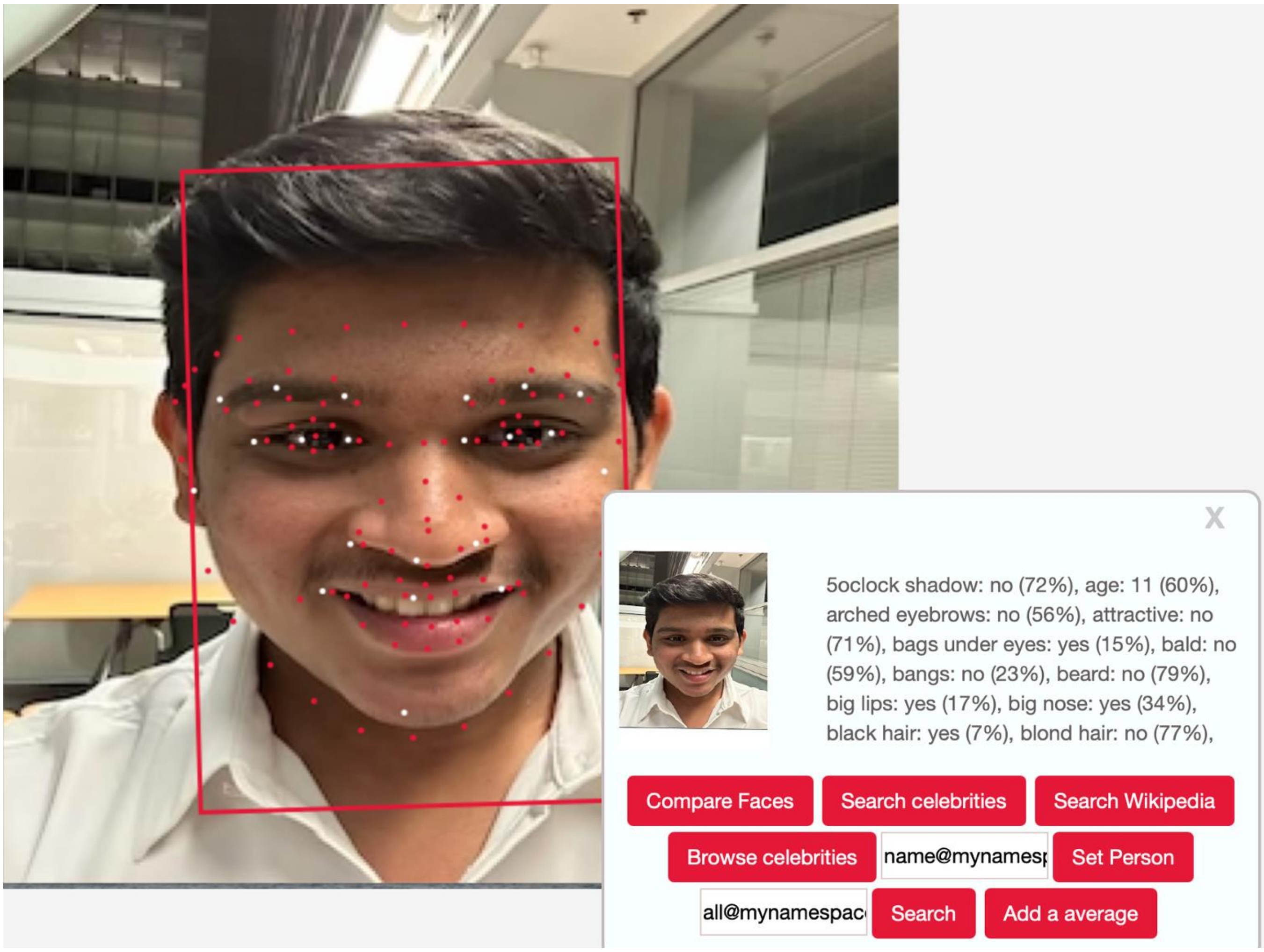
4. [10 points] There are a number of face matching and face analysis applications readily available. For example, Cloud Vision AI, Beta face, PimEyes, Luxand, BioID, Face2Gene, etc.

(a) Experiment with at least two of these applications, and include in your report the input(s) that you gave and the output(s) that was/were produced by each application.

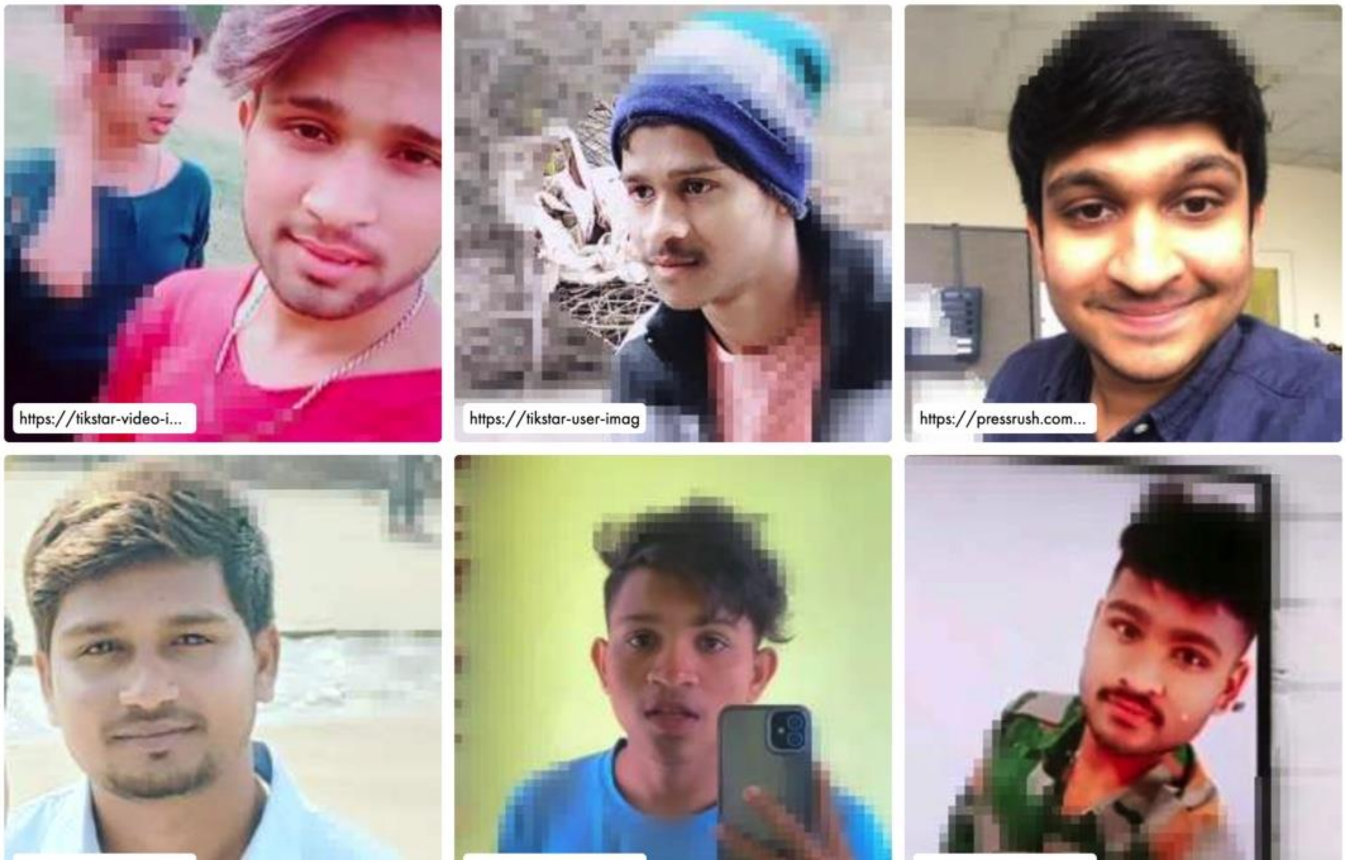
Ans) I experimented with the Beta face and PimEyes software and here are the results:

Input image:

Betaface:


X
5oclock shadow: no (72%), age: 11 (60%), arched eyebrows: no (56%), attractive: no (71%), bags under eyes: yes (15%), bald: no (59%), bangs: no (23%), beard: no (79%), big lips: yes (17%), big nose: yes (34%), black hair: yes (7%), blond hair: no (77%),
Compare Faces
Search celebrities
Search Wikipedia
Browse celebrities
name@mynamesp
Set Person
all@mynamespac
Search
Add a average

PimEyes:



(b) What are the pros and cons of deploying face recognition systems in public spaces? What are some of the ethical aspects of utilizing face recognition in general? Explain your answer in detail. (Note that face recognition is different from general face analytics).

Ans) Pros:

- It can be used to identify people in watchlist at airports and public places which can enhance the security.
- It can be used to identify people making nuisance in public spaces like stadiums, movie theatre, etc.
- It can be used to identify and find missing people using the facial data gathered.

Cons:

- Privacy of the individuals is compromised since they are almost constantly watched by the face recognition technology.
- Storage of the facial recognition data is one of the major concerns because if there is a leak then it would cause a serious security threat.
- False positive is a major concern. If the facial recognition system doesn't have 100% accuracy, then it might identify innocent people for any wrong doings done by others.

When it comes to ethical aspects of utilizing facial recognition the first thing is lack of transparency and consent of individuals, bias and accuracy concerns in the algorithm for facial recognition system.

5. [10 points] Using the face images that you collected in class, answer the following. You may collect additional selfies of your face, if needed.

(a) Show examples of face images exhibiting different types of intra-class variations. Explain the nature of the intra-class variations.

From the above pictures we can observer different types of intra-class variation like:

- Lighting like low lighting and high lighting
- Different facial expressions
- Variations in poses
- Age (which is not clearly evident but is one of the intra-class variation)

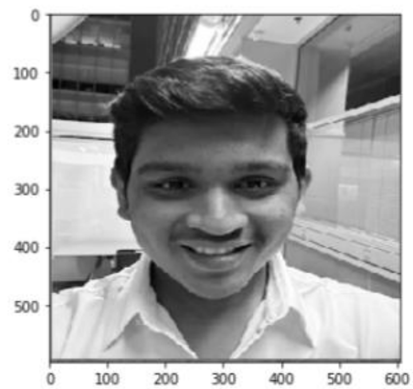
(b) Which portions of the face are more resilient to changes in expression and why?

Ans) The portions include your nose and ears these do not change because nose is a bone, and it doesn't move with the change in expression and when it comes to ears their movement is not totally connected to the movement of the facial expression hence they remain constant.

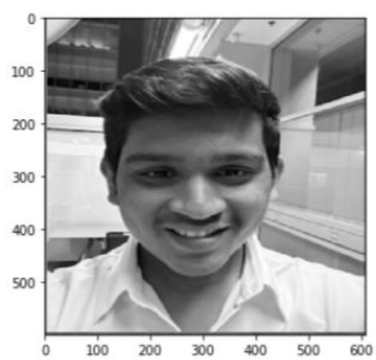
(c) Take one of the frontal face images, convert it to grayscale and convolve the grayscale image with average filters of the following dimensions: 3 x 3, 15 x 15, 45 x 45. Show the input image and the output images. Explain the differences in output pertaining to the 3 filters.

Ans) <u>Input image</u>:

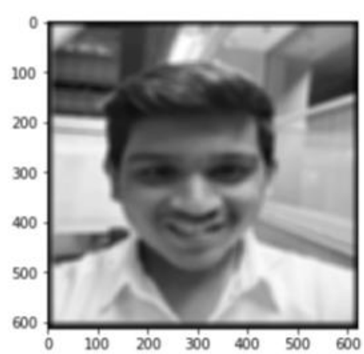
<u>Grayscale image</u>:



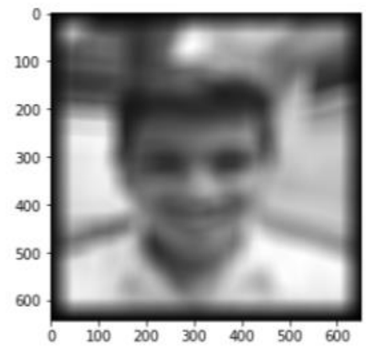
<u>Average filter (3 x 3)</u>:



<u>Average filter (15 x 15)</u>:



<u>Average filter (45 x 45)</u>:



The difference pertaining to the 3 filters is as dimensions increase the picture quality decreases (getting blurred images).

(d) From your perspective, what are some of the challenges in performing face recognition using selfies.

Ans) some of the challenges in performing face recognition using selfies:

- When people try to take selfies from different angles some part of the face might get crop out giving us an corrupted image which may result in false identification.

- The front cameras tend to have less megapixels which will give a poor-quality image when compared to normal images taken from back camera.

Appendix

```
import numpy as np
import math
from matplotlib import pyplot as plt
from PIL import Image
from skimage import color
from skimage import io
from scipy import signal
```

```
td_points = np.loadtxt("hw03_pca_data.txt")
td_points_matrix = np.matrix(td_points)
```

```
#1a

rows = 1000
i = 0

sum_values_left = 0
sum_values_right = 0
sum_val = 0
while i < rows:
    sum_values_left += td_points_matrix[i,0]
    sum_values_right += td_points_matrix[i,1]
    i+=1
mean_left = sum_values_left/rows
mean_right = sum_values_right/rows
mean_vector = [mean_left,mean_right]
print("Mean Vector =",mean_vector)
```

```
Mean Vector = [19.973930000000006, 19.935589999999998]
```

```
pca_data = []
td_points_new = []

for values in range(len(td_points)):
    pca_data = td_points[values] - mean_vector

    td_points_new.append(pca_data)

td_points_new_matrix = np.matrix(td_points_new)

td_points_new_matrix
```

```
matrix([[ 0.03607,  0.63441],
        [-3.63393,  4.93441],
        [ 3.70607,  0.70441],
        ...,
        [-3.55393, -2.97559],
        [ 1.41607,  1.61441],
        [-3.12393,  0.72441]])
```

```python
#1b
covariance_matrix_pca = (td_points_new_matrix.T) * (td_points_new_matrix)
print("Covariance matrix_pca = ", covariance_matrix_pca)
```

```
Covariance matrix_pca =  [[ 9389.4876551   5119.3095313]
 [ 5119.3095313 10374.3620519]]
```

```python
#1c
values,vectors = np.linalg.eig(covariance_matrix)
print("Eigen Values =",values)
```

```
Eigen Values = [ 4.73898557 15.02486414]
```

```python
#1c
print("Eigen Vectors =",vectors)
```

```
Eigen Vectors = [[-0.74018584 -0.67240235]
 [ 0.67240235 -0.74018584]]
```

```python
#1c
eig_values,eig_vectors = np.linalg.eig(covariance_matrix_pca)
array_eig_values = np.array(eig_values)
print("Eigen Values_pca =",array_eig_values)
```

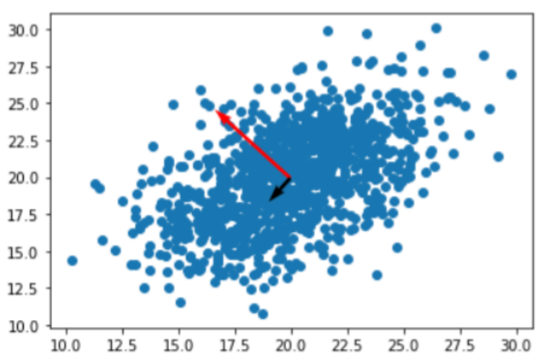
```
Eigen Values_pca = [ 4738.98557021 15024.86413679]
```

```python
#1c
array_eig_vec = np.array(eig_vectors)
print("Eigen Vectors_pca =",array_eig_vec)
```

```
Eigen Vectors_pca = [[-0.74018584 -0.67240235]
 [ 0.67240235 -0.74018584]]
```

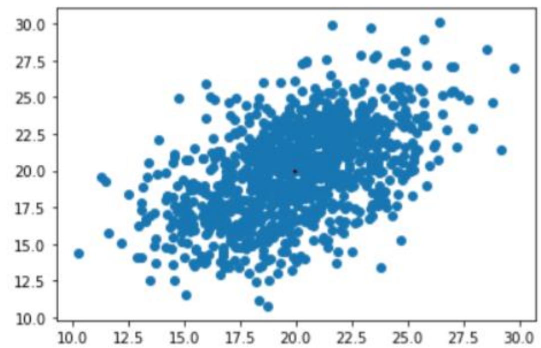
```python
#1d
plt.scatter(td_points[:,0],td_points[:,1])
plt.quiver(*mean_vector, *vectors[:,0],color = 'r',scale = values[0])
plt.quiver(*mean_vector, *vectors[:,1],scale = values[1])
plt.show
```

```
<function matplotlib.pyplot.show(*args, **kw)>
```



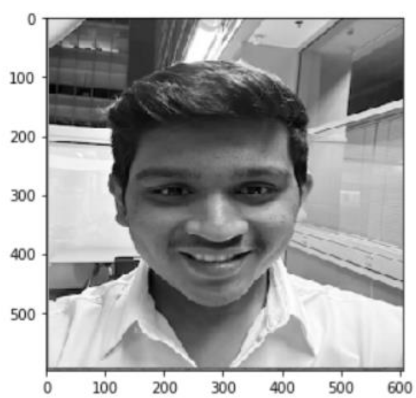
```python
#1d
plt.scatter(td_points[:,0],td_points[:,1])
plt.quiver(*mean_vector, *array_eig_vec[:,0],color = 'r',scale =array_eig_values[0]
plt.quiver(*mean_vector, *array_eig_vec[:,1],scale = array_eig_values[1])
plt.show
```

<function matplotlib.pyplot.show(*args, **kw)>



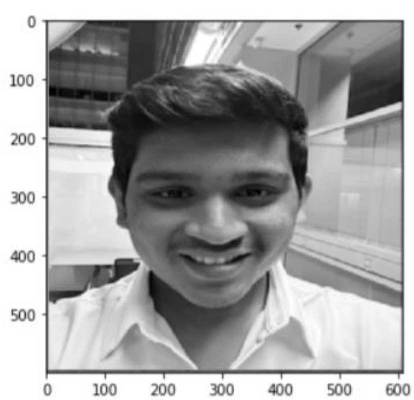

```python
#5c
img_gray = color.rgb2gray(io.imread('Frontal_image.png'))
io.imshow(img_gray)
```

<matplotlib.image.AxesImage at 0x7fdcf95db190>



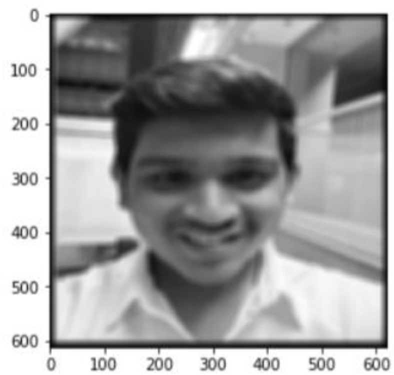

```python
convolve_3 = signal.convolve(img_gray,np.ones((3,3))/3**2)
io.imshow(convolve_3,cmap ='gray')
```

<matplotlib.image.AxesImage at 0x7fdcf9c81b90>



```
convolve_15 = signal.convolve(img_gray,np.ones((15,15))/15**2)
plt.imshow(convolve_15, cmap = 'gray')
```

<matplotlib.image.AxesImage at 0x7fdcfaaff290>




```
convolve_45 = signal.convolve(img_gray,np.ones((45,45))/45**2)
plt.imshow(convolve_45, cmap = 'gray')
```

<matplotlib.image.AxesImage at 0x7fdcf9e87f50>