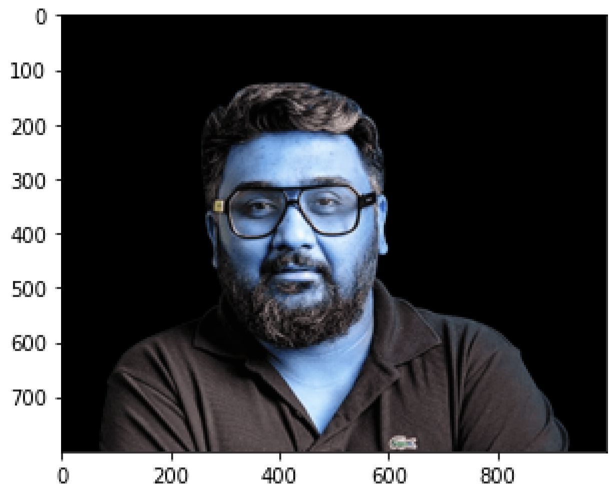
```
In [1]: import numpy as np
        import cv2
        import matplotlib
        from matplotlib import pyplot as plt
        %matplotlib inline
```

```
In [2]: img = cv2.imread('./Test_Images/Kunal Shah _ Founder & CEO_ CRED.png')
        img.shape

Out[2]: (800, 1000, 3)
```

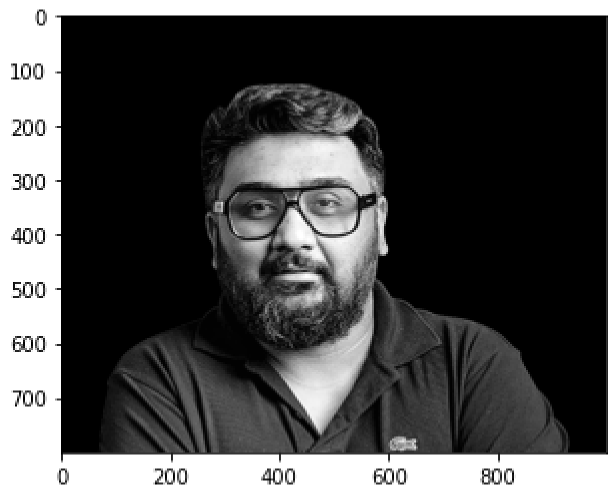
```
In [3]: plt.imshow(img)

Out[3]: <matplotlib.image.AxesImage at 0x261bcf92830>
```



```
In [4]: gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
        gray.shape

Out[4]: (800, 1000)
```

```
In [5]: gray

Out[5]: array([[0, 0, 0, ..., 0, 0, 0],
               [0, 0, 0, ..., 0, 0, 0],
               [0, 0, 0, ..., 0, 0, 0],
               ...,
               [0, 0, 0, ..., 0, 0, 0],
               [0, 0, 0, ..., 0, 0, 0],
               [0, 0, 0, ..., 0, 0, 0]], dtype=uint8)
```

```
In [6]: plt.imshow(gray, cmap='gray')

Out[6]: <matplotlib.image.AxesImage at 0x261bf574820>
```



In [7]:
```
face_cascade = cv2.CascadeClassifier('./opencv/haarcascades/haarcascade_frontalfac
eye_cascade = cv2.CascadeClassifier('./opencv/haarcascades/haarcascade_eye.xml')

faces = face_cascade.detectMultiScale(gray, 1.3, 5)
faces
```

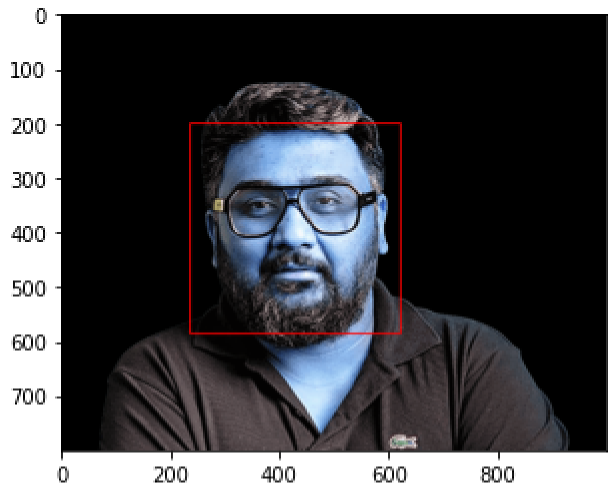
Out[7]:
```
array([[237, 200, 385, 385]])
```

In [8]:
```
(x,y,w,h) = faces[0]
x,y,w,h
```

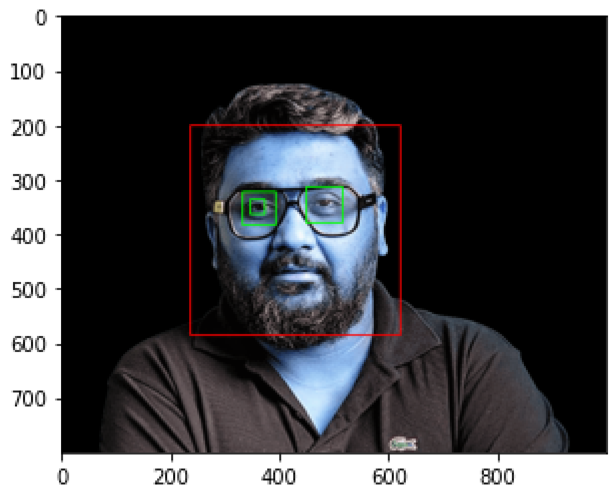
Out[8]:
```
(237, 200, 385, 385)
```

In [9]:
```
face_img = cv2.rectangle(img,(x,y),(x+w,y+h),(255,0,0),2)
plt.imshow(face_img)
```

Out[9]:
```
<matplotlib.image.AxesImage at 0x261bf6e4d00>
```

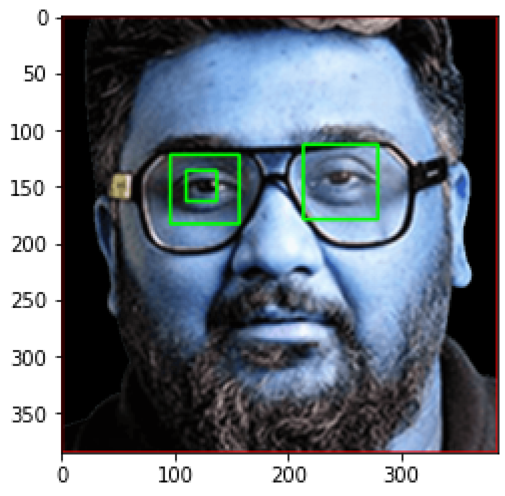


In [10]:
```
cv2.destroyAllWindows()
for (x,y,w,h) in faces:
    face_img = cv2.rectangle(img,(x,y),(x+w,y+h),(255,0,0),2)
    roi_gray = gray[y:y+h, x:x+w]
    roi_color = face_img[y:y+h, x:x+w]
    eyes = eye_cascade.detectMultiScale(roi_gray)
    for (ex,ey,ew,eh) in eyes:
        cv2.rectangle(roi_color,(ex,ey),(ex+ew,ey+eh),(0,255,0),2)


plt.figure()
plt.imshow(face_img, cmap='gray')
plt.show()
```



In [11]:
```
%matplotlib inline
plt.imshow(roi_color, cmap='gray')
```

Out[11]:
```
<matplotlib.image.AxesImage at 0x261d11bfb80>
```



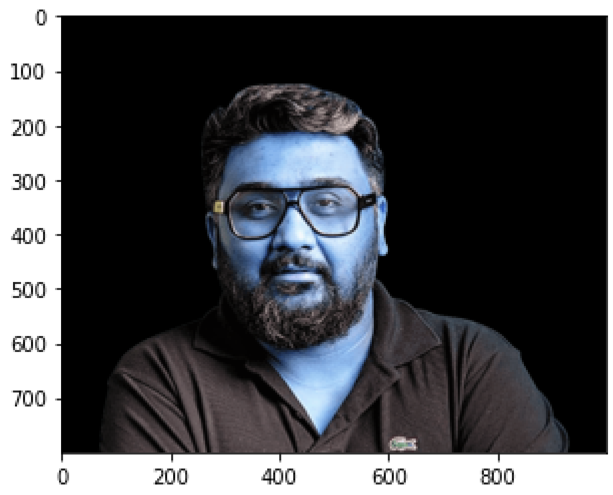
In [12]:
```
cropped_img = np.array(roi_color)
cropped_img.shape
```

Out[12]:
```
(385, 385, 3)
```

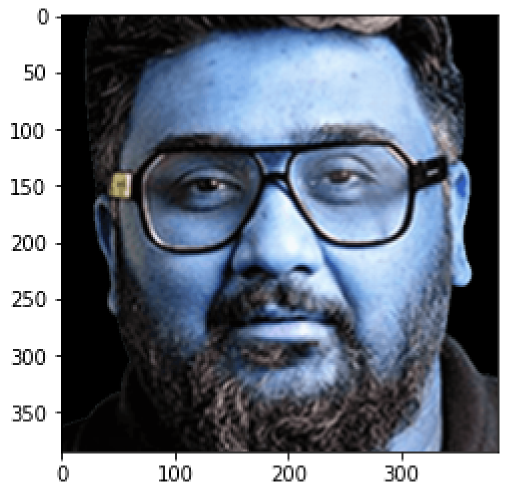
In [13]:
```
def get_cropped_image_if_2_eyes(image_path):
    img = cv2.imread(image_path)
    if img is not None:
        gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
        faces = face_cascade.detectMultiScale(gray, 1.3, 5)
        for (x,y,w,h) in faces:
            roi_gray = gray[y:y+h, x:x+w]
            roi_color = img[y:y+h, x:x+w]
            eyes = eye_cascade.detectMultiScale(roi_gray)
            if len(eyes) >= 2:
                            return roi_color
```

In [14]:
```
original_image = cv2.imread('./Test_Images/Kunal Shah _ Founder & CEO_ CRED.png')
plt.imshow(original_image)
```

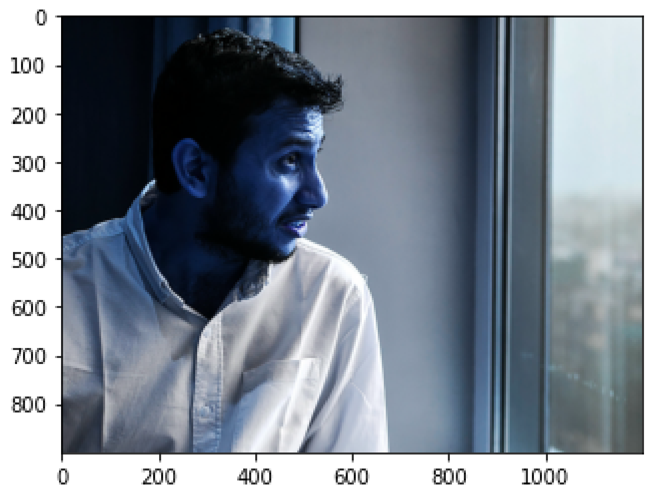
Out[14]:
```
<matplotlib.image.AxesImage at 0x261d122bb80>
```



```
In [15]: cropped_image = get_cropped_image_if_2_eyes('./Test_Images/Kunal Shah _ Founder & 
         plt.imshow(cropped_image)

Out[15]: <matplotlib.image.AxesImage at 0x261d1292260>
```



```
In [16]: org_image_obstructed = cv2.imread('./Test_Images/Ritesh Agarwal book of risk takin
         plt.imshow(org_image_obstructed)

Out[16]: <matplotlib.image.AxesImage at 0x261d17b33d0>
```



```
In [17]: cropped_image_no_2_eyes = get_cropped_image_if_2_eyes('./Test_Images/Ritesh Agarwa
         cropped_image_no_2_eyes

In [18]: path_to_data = "./dataset/"
```

```
path_to_cr_data = "./dataset/cropped/"
```

In [19]:
```
import os
img_dirs = []
for entry in os.scandir(path_to_data):
    if entry.is_dir():
        img_dirs.append(entry.path)
```

In [20]:
```
img_dirs
```

Out[20]:
```
['./dataset/Bhavish_Aggarwal',
 './dataset/cropped',
 './dataset/Kunal_Shah',
 './dataset/Ritesh_Agarwal',
 './dataset/Sachin_Bansal',
 './dataset/Shradha_Sharma',
 './dataset/Vijay_Sharma']
```

In [21]:
```
import shutil
if os.path.exists(path_to_cr_data):
     shutil.rmtree(path_to_cr_data)
os.mkdir(path_to_cr_data)
```

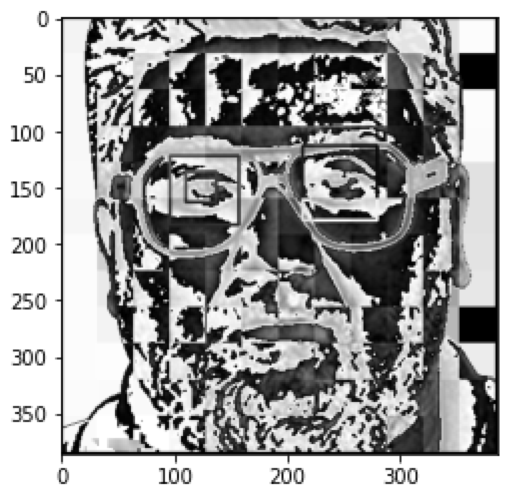
In [22]:
```
cropped_image_dirs = []
celebrity_file_names_dict = {}
for img_dir in img_dirs:
    count = 1
    celebrity_name = img_dir.split('/')[-1]
    celebrity_file_names_dict[celebrity_name] = []
    for entry in os.scandir(img_dir):
        roi_color = get_cropped_image_if_2_eyes(entry.path)
        if roi_color is not None:
            cropped_folder = path_to_cr_data + celebrity_name
            if not os.path.exists(cropped_folder):
                os.makedirs(cropped_folder)
                cropped_image_dirs.append(cropped_folder)
                print("Generating cropped images in folder: ",cropped_folder)
            cropped_file_name = celebrity_name + str(count) + ".png"
            cropped_file_path = cropped_folder + "/" + cropped_file_name
            cv2.imwrite(cropped_file_path, roi_color)
            celebrity_file_names_dict[celebrity_name].append(cropped_file_path)
            count += 1
```

```
Generating cropped images in folder:  ./dataset/cropped/Bhavish_Aggarwal
Generating cropped images in folder:  ./dataset/cropped/Kunal_Shah
Generating cropped images in folder:  ./dataset/cropped/Ritesh_Agarwal
Generating cropped images in folder:  ./dataset/cropped/Sachin_Bansal
Generating cropped images in folder:  ./dataset/cropped/Shradha_Sharma
Generating cropped images in folder:  ./dataset/cropped/Vijay_Sharma
```

In [23]:
```
import numpy as np
import pywt
import cv2

def w2d(img, mode='haar', level=1):
    imArray = img
    #Datatype conversions
    #convert to grayscale
    imArray = cv2.cvtColor( imArray,cv2.COLOR_RGB2GRAY )
    #convert to float
    imArray =  np.float32(imArray)
    imArray /= 255;
    # compute coefficients
    coeffs=pywt.wavedec2(imArray, mode, level=level)

    #Process Coefficients
    coeffs_H=list(coeffs)
    coeffs_H[0] *= 0;

    # reconstruction
    imArray_H=pywt.waverec2(coeffs_H, mode);
    imArray_H *= 255;
    imArray_H =  np.uint8(imArray_H)

    return imArray_H
```

In [24]:
```
im_har = w2d(cropped_img,'db1',5)
plt.imshow(im_har, cmap='gray')
```

Out[24]:
```
<matplotlib.image.AxesImage at 0x261cf4add80>
```



In [25]:
```
celebrity_file_names_dict = {}
for img_dir in cropped_image_dirs:
    celebrity_name = img_dir.split('/')[-1]
    file_list = []
    for entry in os.scandir(img_dir):
        file_list.append(entry.path)
    celebrity_file_names_dict[celebrity_name] = file_list
celebrity_file_names_dict
```

```
Out[25]: {'Bhavish_Aggarwal': ['./dataset/cropped/Bhavish_Aggarwal\\Bhavish_Aggarwal10.pn
         g',
           './dataset/cropped/Bhavish_Aggarwal\\Bhavish_Aggarwal13.png',
           './dataset/cropped/Bhavish_Aggarwal\\Bhavish_Aggarwal16.png',
           './dataset/cropped/Bhavish_Aggarwal\\Bhavish_Aggarwal17.png',
           './dataset/cropped/Bhavish_Aggarwal\\Bhavish_Aggarwal18.png',
           './dataset/cropped/Bhavish_Aggarwal\\Bhavish_Aggarwal2.png',
           './dataset/cropped/Bhavish_Aggarwal\\Bhavish_Aggarwal22.png',
           './dataset/cropped/Bhavish_Aggarwal\\Bhavish_Aggarwal23.png',
           './dataset/cropped/Bhavish_Aggarwal\\Bhavish_Aggarwal24.png',
           './dataset/cropped/Bhavish_Aggarwal\\Bhavish_Aggarwal25.png',
           './dataset/cropped/Bhavish_Aggarwal\\Bhavish_Aggarwal27.png',
           './dataset/cropped/Bhavish_Aggarwal\\Bhavish_Aggarwal28.png',
           './dataset/cropped/Bhavish_Aggarwal\\Bhavish_Aggarwal29.png',
           './dataset/cropped/Bhavish_Aggarwal\\Bhavish_Aggarwal30.png',
           './dataset/cropped/Bhavish_Aggarwal\\Bhavish_Aggarwal31.png',
           './dataset/cropped/Bhavish_Aggarwal\\Bhavish_Aggarwal32.png',
           './dataset/cropped/Bhavish_Aggarwal\\Bhavish_Aggarwal33.png',
           './dataset/cropped/Bhavish_Aggarwal\\Bhavish_Aggarwal34.png',
           './dataset/cropped/Bhavish_Aggarwal\\Bhavish_Aggarwal35.png',
           './dataset/cropped/Bhavish_Aggarwal\\Bhavish_Aggarwal37.png',
           './dataset/cropped/Bhavish_Aggarwal\\Bhavish_Aggarwal38.png',
           './dataset/cropped/Bhavish_Aggarwal\\Bhavish_Aggarwal39.png',
           './dataset/cropped/Bhavish_Aggarwal\\Bhavish_Aggarwal4.png',
           './dataset/cropped/Bhavish_Aggarwal\\Bhavish_Aggarwal40.png',
           './dataset/cropped/Bhavish_Aggarwal\\Bhavish_Aggarwal41.png',
           './dataset/cropped/Bhavish_Aggarwal\\Bhavish_Aggarwal42.png',
           './dataset/cropped/Bhavish_Aggarwal\\Bhavish_Aggarwal44.png',
           './dataset/cropped/Bhavish_Aggarwal\\Bhavish_Aggarwal45.png',
           './dataset/cropped/Bhavish_Aggarwal\\Bhavish_Aggarwal46.png',
           './dataset/cropped/Bhavish_Aggarwal\\Bhavish_Aggarwal47.png',
           './dataset/cropped/Bhavish_Aggarwal\\Bhavish_Aggarwal48.png',
           './dataset/cropped/Bhavish_Aggarwal\\Bhavish_Aggarwal49.png',
           './dataset/cropped/Bhavish_Aggarwal\\Bhavish_Aggarwal5.png',
           './dataset/cropped/Bhavish_Aggarwal\\Bhavish_Aggarwal50.png',
           './dataset/cropped/Bhavish_Aggarwal\\Bhavish_Aggarwal51.png',
           './dataset/cropped/Bhavish_Aggarwal\\Bhavish_Aggarwal52.png',
           './dataset/cropped/Bhavish_Aggarwal\\Bhavish_Aggarwal53.png',
           './dataset/cropped/Bhavish_Aggarwal\\Bhavish_Aggarwal54.png',
           './dataset/cropped/Bhavish_Aggarwal\\Bhavish_Aggarwal55.png',
           './dataset/cropped/Bhavish_Aggarwal\\Bhavish_Aggarwal56.png',
           './dataset/cropped/Bhavish_Aggarwal\\Bhavish_Aggarwal57.png',
           './dataset/cropped/Bhavish_Aggarwal\\Bhavish_Aggarwal58.png',
           './dataset/cropped/Bhavish_Aggarwal\\Bhavish_Aggarwal59.png',
           './dataset/cropped/Bhavish_Aggarwal\\Bhavish_Aggarwal6.png',
           './dataset/cropped/Bhavish_Aggarwal\\Bhavish_Aggarwal61.png',
           './dataset/cropped/Bhavish_Aggarwal\\Bhavish_Aggarwal62.png',
           './dataset/cropped/Bhavish_Aggarwal\\Bhavish_Aggarwal63.png',
           './dataset/cropped/Bhavish_Aggarwal\\Bhavish_Aggarwal64.png',
           './dataset/cropped/Bhavish_Aggarwal\\Bhavish_Aggarwal65.png',
           './dataset/cropped/Bhavish_Aggarwal\\Bhavish_Aggarwal66.png',
           './dataset/cropped/Bhavish_Aggarwal\\Bhavish_Aggarwal67.png',
           './dataset/cropped/Bhavish_Aggarwal\\Bhavish_Aggarwal68.png',
           './dataset/cropped/Bhavish_Aggarwal\\Bhavish_Aggarwal7.png',
           './dataset/cropped/Bhavish_Aggarwal\\Bhavish_Aggarwal70.png',
           './dataset/cropped/Bhavish_Aggarwal\\Bhavish_Aggarwal71.png',
           './dataset/cropped/Bhavish_Aggarwal\\Bhavish_Aggarwal72.png',
           './dataset/cropped/Bhavish_Aggarwal\\Bhavish_Aggarwal8.png',
           './dataset/cropped/Bhavish_Aggarwal\\Bhavish_Aggarwal9.png'],
          'Kunal_Shah': ['./dataset/cropped/Kunal_Shah\\Kunal_Shah1.png',
           './dataset/cropped/Kunal_Shah\\Kunal_Shah11.png',
           './dataset/cropped/Kunal_Shah\\Kunal_Shah15.png',
           './dataset/cropped/Kunal_Shah\\Kunal_Shah16.png',
           './dataset/cropped/Kunal_Shah\\Kunal_Shah17.png',
           './dataset/cropped/Kunal_Shah\\Kunal_Shah18.png',
           './dataset/cropped/Kunal_Shah\\Kunal_Shah19.png',
           './dataset/cropped/Kunal_Shah\\Kunal_Shah2.png',
           './dataset/cropped/Kunal_Shah\\Kunal_Shah23.png',
           './dataset/cropped/Kunal_Shah\\Kunal_Shah25.png',
           './dataset/cropped/Kunal_Shah\\Kunal_Shah27.png',
           './dataset/cropped/Kunal_Shah\\Kunal_Shah28.png',
           './dataset/cropped/Kunal_Shah\\Kunal_Shah29.png',
           './dataset/cropped/Kunal_Shah\\Kunal_Shah31.png',
           './dataset/cropped/Kunal_Shah\\Kunal_Shah32.png',
           './dataset/cropped/Kunal_Shah\\Kunal_Shah34.png',
           './dataset/cropped/Kunal_Shah\\Kunal_Shah35.png',
           './dataset/cropped/Kunal_Shah\\Kunal_Shah37.png',
           './dataset/cropped/Kunal_Shah\\Kunal_Shah38.png',
           './dataset/cropped/Kunal_Shah\\Kunal_Shah4.png',
           './dataset/cropped/Kunal_Shah\\Kunal_Shah42.png',
           './dataset/cropped/Kunal_Shah\\Kunal_Shah43.png',
           './dataset/cropped/Kunal_Shah\\Kunal_Shah44.png',
           './dataset/cropped/Kunal_Shah\\Kunal_Shah45.png',
           './dataset/cropped/Kunal_Shah\\Kunal_Shah46.png',
           './dataset/cropped/Kunal_Shah\\Kunal_Shah6.png',
           './dataset/cropped/Kunal_Shah\\Kunal_Shah7.png',
           './dataset/cropped/Kunal_Shah\\Kunal_Shah8.png',
           './dataset/cropped/Kunal_Shah\\Kunal_Shah9.png'],
          'Ritesh_Agarwal': ['./dataset/cropped/Ritesh_Agarwal\\Ritesh_Agarwal1.png',
           './dataset/cropped/Ritesh_Agarwal\\Ritesh_Agarwal10.png',
           './dataset/cropped/Ritesh_Agarwal\\Ritesh_Agarwal11.png',
           './dataset/cropped/Ritesh_Agarwal\\Ritesh_Agarwal12.png',
           './dataset/cropped/Ritesh_Agarwal\\Ritesh_Agarwal13.png',
           './dataset/cropped/Ritesh_Agarwal\\Ritesh_Agarwal14.png',
           './dataset/cropped/Ritesh_Agarwal\\Ritesh_Agarwal15.png',
           './dataset/cropped/Ritesh_Agarwal\\Ritesh_Agarwal16.png',
           './dataset/cropped/Ritesh_Agarwal\\Ritesh_Agarwal17.png',
           './dataset/cropped/Ritesh_Agarwal\\Ritesh_Agarwal18.png',
           './dataset/cropped/Ritesh_Agarwal\\Ritesh_Agarwal19.png',
           './dataset/cropped/Ritesh_Agarwal\\Ritesh_Agarwal2.png',
           './dataset/cropped/Ritesh_Agarwal\\Ritesh_Agarwal20.png',
           './dataset/cropped/Ritesh_Agarwal\\Ritesh_Agarwal21.png',
           './dataset/cropped/Ritesh_Agarwal\\Ritesh_Agarwal23.png',
           './dataset/cropped/Ritesh_Agarwal\\Ritesh_Agarwal24.png',
           './dataset/cropped/Ritesh_Agarwal\\Ritesh_Agarwal25.png',
           './dataset/cropped/Ritesh_Agarwal\\Ritesh_Agarwal26.png',
           './dataset/cropped/Ritesh_Agarwal\\Ritesh_Agarwal27.png',
           './dataset/cropped/Ritesh_Agarwal\\Ritesh_Agarwal28.png',
           './dataset/cropped/Ritesh_Agarwal\\Ritesh_Agarwal29.png',
           './dataset/cropped/Ritesh_Agarwal\\Ritesh_Agarwal3.png',
           './dataset/cropped/Ritesh_Agarwal\\Ritesh_Agarwal30.png',
           './dataset/cropped/Ritesh_Agarwal\\Ritesh_Agarwal31.png',
           './dataset/cropped/Ritesh_Agarwal\\Ritesh_Agarwal32.png',
           './dataset/cropped/Ritesh_Agarwal\\Ritesh_Agarwal33.png',
           './dataset/cropped/Ritesh_Agarwal\\Ritesh_Agarwal34.png',
           './dataset/cropped/Ritesh_Agarwal\\Ritesh_Agarwal35.png',
           './dataset/cropped/Ritesh_Agarwal\\Ritesh_Agarwal36.png',
           './dataset/cropped/Ritesh_Agarwal\\Ritesh_Agarwal37.png',
           './dataset/cropped/Ritesh_Agarwal\\Ritesh_Agarwal39.png',
           './dataset/cropped/Ritesh_Agarwal\\Ritesh_Agarwal4.png',
           './dataset/cropped/Ritesh_Agarwal\\Ritesh_Agarwal40.png',
           './dataset/cropped/Ritesh_Agarwal\\Ritesh_Agarwal41.png',
           './dataset/cropped/Ritesh_Agarwal\\Ritesh_Agarwal42.png',
           './dataset/cropped/Ritesh_Agarwal\\Ritesh_Agarwal43.png',
           './dataset/cropped/Ritesh_Agarwal\\Ritesh_Agarwal44.png',
           './dataset/cropped/Ritesh_Agarwal\\Ritesh_Agarwal45.png',
           './dataset/cropped/Ritesh_Agarwal\\Ritesh_Agarwal46.png',
           './dataset/cropped/Ritesh_Agarwal\\Ritesh_Agarwal47.png',
           './dataset/cropped/Ritesh_Agarwal\\Ritesh_Agarwal48.png',
           './dataset/cropped/Ritesh_Agarwal\\Ritesh_Agarwal49.png',
           './dataset/cropped/Ritesh_Agarwal\\Ritesh_Agarwal5.png',
           './dataset/cropped/Ritesh_Agarwal\\Ritesh_Agarwal50.png',
           './dataset/cropped/Ritesh_Agarwal\\Ritesh_Agarwal51.png',
           './dataset/cropped/Ritesh_Agarwal\\Ritesh_Agarwal52.png',
           './dataset/cropped/Ritesh_Agarwal\\Ritesh_Agarwal55.png',
           './dataset/cropped/Ritesh_Agarwal\\Ritesh_Agarwal57.png',
           './dataset/cropped/Ritesh_Agarwal\\Ritesh_Agarwal58.png',
           './dataset/cropped/Ritesh_Agarwal\\Ritesh_Agarwal59.png',
           './dataset/cropped/Ritesh_Agarwal\\Ritesh_Agarwal6.png',
           './dataset/cropped/Ritesh_Agarwal\\Ritesh_Agarwal60.png',
           './dataset/cropped/Ritesh_Agarwal\\Ritesh_Agarwal63.png',
           './dataset/cropped/Ritesh_Agarwal\\Ritesh_Agarwal64.png',
           './dataset/cropped/Ritesh_Agarwal\\Ritesh_Agarwal65.png',
           './dataset/cropped/Ritesh_Agarwal\\Ritesh_Agarwal66.png',
           './dataset/cropped/Ritesh_Agarwal\\Ritesh_Agarwal67.png',
           './dataset/cropped/Ritesh_Agarwal\\Ritesh_Agarwal68.png',
           './dataset/cropped/Ritesh_Agarwal\\Ritesh_Agarwal69.png',
           './dataset/cropped/Ritesh_Agarwal\\Ritesh_Agarwal7.png',
           './dataset/cropped/Ritesh_Agarwal\\Ritesh_Agarwal70.png',
           './dataset/cropped/Ritesh_Agarwal\\Ritesh_Agarwal71.png',
           './dataset/cropped/Ritesh_Agarwal\\Ritesh_Agarwal72.png',
           './dataset/cropped/Ritesh_Agarwal\\Ritesh_Agarwal73.png',
           './dataset/cropped/Ritesh_Agarwal\\Ritesh_Agarwal74.png',
           './dataset/cropped/Ritesh_Agarwal\\Ritesh_Agarwal75.png',
           './dataset/cropped/Ritesh_Agarwal\\Ritesh_Agarwal76.png',
           './dataset/cropped/Ritesh_Agarwal\\Ritesh_Agarwal77.png',
           './dataset/cropped/Ritesh_Agarwal\\Ritesh_Agarwal78.png',
           './dataset/cropped/Ritesh_Agarwal\\Ritesh_Agarwal79.png',
           './dataset/cropped/Ritesh_Agarwal\\Ritesh_Agarwal80.png',
           './dataset/cropped/Ritesh_Agarwal\\Ritesh_Agarwal81.png',
           './dataset/cropped/Ritesh_Agarwal\\Ritesh_Agarwal9.png'],
          'Sachin_Bansal': ['./dataset/cropped/Sachin_Bansal\\Sachin_Bansal1.png',
           './dataset/cropped/Sachin_Bansal\\Sachin_Bansal16.png',
           './dataset/cropped/Sachin_Bansal\\Sachin_Bansal17.png',
           './dataset/cropped/Sachin_Bansal\\Sachin_Bansal2.png',
           './dataset/cropped/Sachin_Bansal\\Sachin_Bansal21.png',
           './dataset/cropped/Sachin_Bansal\\Sachin_Bansal22.png',
           './dataset/cropped/Sachin_Bansal\\Sachin_Bansal23.png',
           './dataset/cropped/Sachin_Bansal\\Sachin_Bansal24.png',
           './dataset/cropped/Sachin_Bansal\\Sachin_Bansal25.png',
           './dataset/cropped/Sachin_Bansal\\Sachin_Bansal27.png',
           './dataset/cropped/Sachin_Bansal\\Sachin_Bansal28.png',
           './dataset/cropped/Sachin_Bansal\\Sachin_Bansal29.png',
           './dataset/cropped/Sachin_Bansal\\Sachin_Bansal3.png',
           './dataset/cropped/Sachin_Bansal\\Sachin_Bansal30.png',
           './dataset/cropped/Sachin_Bansal\\Sachin_Bansal32.png',
           './dataset/cropped/Sachin_Bansal\\Sachin_Bansal33.png',
           './dataset/cropped/Sachin_Bansal\\Sachin_Bansal36.png',
           './dataset/cropped/Sachin_Bansal\\Sachin_Bansal37.png',
           './dataset/cropped/Sachin_Bansal\\Sachin_Bansal38.png',
           './dataset/cropped/Sachin_Bansal\\Sachin_Bansal39.png',
           './dataset/cropped/Sachin_Bansal\\Sachin_Bansal41.png',
           './dataset/cropped/Sachin_Bansal\\Sachin_Bansal42.png',
           './dataset/cropped/Sachin_Bansal\\Sachin_Bansal46.png',
           './dataset/cropped/Sachin_Bansal\\Sachin_Bansal47.png',
           './dataset/cropped/Sachin_Bansal\\Sachin_Bansal48.png',
           './dataset/cropped/Sachin_Bansal\\Sachin_Bansal49.png',
           './dataset/cropped/Sachin_Bansal\\Sachin_Bansal5.png',
           './dataset/cropped/Sachin_Bansal\\Sachin_Bansal50.png',
           './dataset/cropped/Sachin_Bansal\\Sachin_Bansal52.png',
           './dataset/cropped/Sachin_Bansal\\Sachin_Bansal56.png',
           './dataset/cropped/Sachin_Bansal\\Sachin_Bansal57.png',
           './dataset/cropped/Sachin_Bansal\\Sachin_Bansal58.png',
           './dataset/cropped/Sachin_Bansal\\Sachin_Bansal59.png',
           './dataset/cropped/Sachin_Bansal\\Sachin_Bansal6.png',
           './dataset/cropped/Sachin_Bansal\\Sachin_Bansal60.png',
           './dataset/cropped/Sachin_Bansal\\Sachin_Bansal62.png',
           './dataset/cropped/Sachin_Bansal\\Sachin_Bansal63.png',
           './dataset/cropped/Sachin_Bansal\\Sachin_Bansal64.png',
           './dataset/cropped/Sachin_Bansal\\Sachin_Bansal66.png',
           './dataset/cropped/Sachin_Bansal\\Sachin_Bansal68.png',
           './dataset/cropped/Sachin_Bansal\\Sachin_Bansal72.png',
           './dataset/cropped/Sachin_Bansal\\Sachin_Bansal73.png',
           './dataset/cropped/Sachin_Bansal\\Sachin_Bansal75.png',
           './dataset/cropped/Sachin_Bansal\\Sachin_Bansal8.png',
           './dataset/cropped/Sachin_Bansal\\Sachin_Bansal9.png'],
          'Shradha_Sharma': ['./dataset/cropped/Shradha_Sharma\\Shradha_Sharma1.png',
           './dataset/cropped/Shradha_Sharma\\Shradha_Sharma10.png',
           './dataset/cropped/Shradha_Sharma\\Shradha_Sharma11.png',
           './dataset/cropped/Shradha_Sharma\\Shradha_Sharma12.png',
           './dataset/cropped/Shradha_Sharma\\Shradha_Sharma13.png',
           './dataset/cropped/Shradha_Sharma\\Shradha_Sharma14.png',
           './dataset/cropped/Shradha_Sharma\\Shradha_Sharma15.png',
           './dataset/cropped/Shradha_Sharma\\Shradha_Sharma16.png',
           './dataset/cropped/Shradha_Sharma\\Shradha_Sharma17.png',
           './dataset/cropped/Shradha_Sharma\\Shradha_Sharma18.png',
           './dataset/cropped/Shradha_Sharma\\Shradha_Sharma19.png',
           './dataset/cropped/Shradha_Sharma\\Shradha_Sharma2.png',
           './dataset/cropped/Shradha_Sharma\\Shradha_Sharma20.png',
           './dataset/cropped/Shradha_Sharma\\Shradha_Sharma24.png',
           './dataset/cropped/Shradha_Sharma\\Shradha_Sharma25.png',
           './dataset/cropped/Shradha_Sharma\\Shradha_Sharma27.png',
           './dataset/cropped/Shradha_Sharma\\Shradha_Sharma28.png',
           './dataset/cropped/Shradha_Sharma\\Shradha_Sharma3.png',
           './dataset/cropped/Shradha_Sharma\\Shradha_Sharma30.png',
           './dataset/cropped/Shradha_Sharma\\Shradha_Sharma31.png',
           './dataset/cropped/Shradha_Sharma\\Shradha_Sharma32.png',
           './dataset/cropped/Shradha_Sharma\\Shradha_Sharma34.png',
           './dataset/cropped/Shradha_Sharma\\Shradha_Sharma35.png',
           './dataset/cropped/Shradha_Sharma\\Shradha_Sharma37.png',
           './dataset/cropped/Shradha_Sharma\\Shradha_Sharma38.png',
           './dataset/cropped/Shradha_Sharma\\Shradha_Sharma39.png',
           './dataset/cropped/Shradha_Sharma\\Shradha_Sharma4.png',
           './dataset/cropped/Shradha_Sharma\\Shradha_Sharma41.png',
           './dataset/cropped/Shradha_Sharma\\Shradha_Sharma42.png',
           './dataset/cropped/Shradha_Sharma\\Shradha_Sharma45.png',
           './dataset/cropped/Shradha_Sharma\\Shradha_Sharma46.png',
           './dataset/cropped/Shradha_Sharma\\Shradha_Sharma5.png',
           './dataset/cropped/Shradha_Sharma\\Shradha_Sharma51.png',
           './dataset/cropped/Shradha_Sharma\\Shradha_Sharma52.png',
           './dataset/cropped/Shradha_Sharma\\Shradha_Sharma53.png',
           './dataset/cropped/Shradha_Sharma\\Shradha_Sharma54.png',
           './dataset/cropped/Shradha_Sharma\\Shradha_Sharma55.png',
           './dataset/cropped/Shradha_Sharma\\Shradha_Sharma57.png',
           './dataset/cropped/Shradha_Sharma\\Shradha_Sharma58.png',
           './dataset/cropped/Shradha_Sharma\\Shradha_Sharma6.png',
           './dataset/cropped/Shradha_Sharma\\Shradha_Sharma66.png',
           './dataset/cropped/Shradha_Sharma\\Shradha_Sharma67.png',
           './dataset/cropped/Shradha_Sharma\\Shradha_Sharma68.png',
           './dataset/cropped/Shradha_Sharma\\Shradha_Sharma69.png',
           './dataset/cropped/Shradha_Sharma\\Shradha_Sharma70.png',
           './dataset/cropped/Shradha_Sharma\\Shradha_Sharma71.png',
           './dataset/cropped/Shradha_Sharma\\Shradha_Sharma72.png',
           './dataset/cropped/Shradha_Sharma\\Shradha_Sharma73.png',
           './dataset/cropped/Shradha_Sharma\\Shradha_Sharma74.png',
           './dataset/cropped/Shradha_Sharma\\Shradha_Sharma75.png',
           './dataset/cropped/Shradha_Sharma\\Shradha_Sharma76.png',
           './dataset/cropped/Shradha_Sharma\\Shradha_Sharma8.png',
           './dataset/cropped/Shradha_Sharma\\Shradha_Sharma9.png'],
          'Vijay_Sharma': ['./dataset/cropped/Vijay_Sharma\\Vijay_Sharma1.png',
           './dataset/cropped/Vijay_Sharma\\Vijay_Sharma14.png',
           './dataset/cropped/Vijay_Sharma\\Vijay_Sharma16.png',
           './dataset/cropped/Vijay_Sharma\\Vijay_Sharma17.png',
           './dataset/cropped/Vijay_Sharma\\Vijay_Sharma19.png',
           './dataset/cropped/Vijay_Sharma\\Vijay_Sharma2.png',
           './dataset/cropped/Vijay_Sharma\\Vijay_Sharma20.png',
           './dataset/cropped/Vijay_Sharma\\Vijay_Sharma21.png',
           './dataset/cropped/Vijay_Sharma\\Vijay_Sharma22.png',
           './dataset/cropped/Vijay_Sharma\\Vijay_Sharma23.png',
           './dataset/cropped/Vijay_Sharma\\Vijay_Sharma24.png',
           './dataset/cropped/Vijay_Sharma\\Vijay_Sharma27.png',
           './dataset/cropped/Vijay_Sharma\\Vijay_Sharma3.png',
           './dataset/cropped/Vijay_Sharma\\Vijay_Sharma30.png',
           './dataset/cropped/Vijay_Sharma\\Vijay_Sharma31.png',
           './dataset/cropped/Vijay_Sharma\\Vijay_Sharma33.png',
           './dataset/cropped/Vijay_Sharma\\Vijay_Sharma34.png',
           './dataset/cropped/Vijay_Sharma\\Vijay_Sharma35.png',
           './dataset/cropped/Vijay_Sharma\\Vijay_Sharma38.png',
           './dataset/cropped/Vijay_Sharma\\Vijay_Sharma4.png',
           './dataset/cropped/Vijay_Sharma\\Vijay_Sharma41.png',
           './dataset/cropped/Vijay_Sharma\\Vijay_Sharma42.png',
           './dataset/cropped/Vijay_Sharma\\Vijay_Sharma43.png',
           './dataset/cropped/Vijay_Sharma\\Vijay_Sharma44.png',
           './dataset/cropped/Vijay_Sharma\\Vijay_Sharma46.png',
           './dataset/cropped/Vijay_Sharma\\Vijay_Sharma47.png',
           './dataset/cropped/Vijay_Sharma\\Vijay_Sharma48.png',
           './dataset/cropped/Vijay_Sharma\\Vijay_Sharma5.png']}
```

In [26]:

```
class_dict = {}
count = 0
for celebrity_name in celebrity_file_names_dict.keys():
    class_dict[celebrity_name] = count
    count = count + 1
class_dict
```

Out[26]:

```
{'Bhavish_Aggarwal': 0,
 'Kunal_Shah': 1,
 'Ritesh_Agarwal': 2,
 'Sachin_Bansal': 3,
 'Shradha_Sharma': 4,
 'Vijay_Sharma': 5}
```

In [27]:

```
X, y = [], []
for celebrity_name, training_files in celebrity_file_names_dict.items():
    for training_image in training_files:
        img = cv2.imread(training_image)
        scalled_raw_img = cv2.resize(img, (32, 32))
        img_har = w2d(img,'db1',5)
        scalled_img_har = cv2.resize(img_har, (32, 32))
        combined_img = np.vstack((scalled_raw_img.reshape(32*32*3,1),scalled_img_ha
        X.append(combined_img)
        y.append(class_dict[celebrity_name])
```

In [28]:

```
len(X[0])
```

Out[28]:

```
4096
```

In [29]:

```
X[0]
```

Out[29]:
```
array([[218],
       [227],
       [231],
       ...,
       [ 18],
       [ 46],
       [ 28]], dtype=uint8)
```

In [30]:
```
y[0]
```

Out[30]:
```
0
```

In [31]:
```
X = np.array(X).reshape(len(X),4096).astype(float)
X.shape
```

Out[31]:
```
(286, 4096)
```

In [33]:
```
from sklearn.svm import SVC
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.pipeline import Pipeline
from sklearn.metrics import classification_report
```

In [34]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=0)

pipe = Pipeline([('scaler', StandardScaler()), ('svc', SVC(kernel = 'rbf', C = 10)
pipe.fit(X_train, y_train)
pipe.score(X_test, y_test)
```

Out[34]:
```
0.8611111111111112
```

In [35]:
```
print(classification_report(y_test, pipe.predict(X_test)))
```

```
              precision    recall  f1-score   support

           0       0.85      0.92      0.88        12
           1       0.83      0.71      0.77         7
           2       0.79      0.83      0.81        18
           3       0.79      1.00      0.88        11
           4       1.00      0.94      0.97        17
           5       1.00      0.57      0.73         7

    accuracy                           0.86        72
   macro avg       0.88      0.83      0.84        72
weighted avg       0.87      0.86      0.86        72
```

In [36]:
```
from sklearn import svm
from sklearn.ensemble import RandomForestClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.pipeline import make_pipeline
from sklearn.model_selection import GridSearchCV
```

In [37]:
```
model_params = {
    'svm': {
        'model': svm.SVC(gamma='auto',probability=True),
        'params' : {
            'svc__C': [1,10,100,1000],
            'svc__kernel': ['rbf','linear']
        }
    },
    'random_forest': {
        'model': RandomForestClassifier(),
        'params' : {
            'randomforestclassifier__n_estimators': [1,5,10]
        }
    },
    'logistic_regression' : {
        'model': LogisticRegression(solver='liblinear',multi_class='auto'),
        'params': {
            'logisticregression__C': [1,5,10]
        }
    }
}
```

In [38]:
```
scores = []
best_estimators = {}
import pandas as pd
for algo, mp in model_params.items():
    pipe = make_pipeline(StandardScaler(), mp['model'])
    clf =  GridSearchCV(pipe, mp['params'], cv=5, return_train_score=False)
    clf.fit(X_train, y_train)
    scores.append({
        'model': algo,
        'best_score': clf.best_score_,
        'best_params': clf.best_params_
    })
    best_estimators[algo] = clf.best_estimator_

df = pd.DataFrame(scores,columns=['model','best_score','best_params'])
df
```

Out[38]:

	model	best_score	best_params
0	svm	0.813068	{'svc__C': 1, 'svc__kernel': 'linear'}
1	random_forest	0.537209	{'randomforestclassifier__n_estimators': 10}
2	logistic_regression	0.864673	{'logisticregression__C': 5}

In [39]:
```
best_estimators
```

Out[39]:
```
{'svm': Pipeline(steps=[('standardscaler', StandardScaler()),
                 ('svc',
                  SVC(C=1, gamma='auto', kernel='linear', probability=True))]),
 'random_forest': Pipeline(steps=[('standardscaler', StandardScaler()),
                 ('randomforestclassifier',
                  RandomForestClassifier(n_estimators=10))]),
 'logistic_regression': Pipeline(steps=[('standardscaler', StandardScaler()),
                 ('logisticregression',
                  LogisticRegression(C=5, solver='liblinear'))])}
```

In [40]:
```
best_estimators['svm'].score(X_test,y_test)
```

Out[40]:
```
0.9166666666666666
```

In [41]:
```
best_estimators['random_forest'].score(X_test,y_test)
```

Out[41]:
```
0.5694444444444444
```

In [42]:
```
best_estimators['logistic_regression'].score(X_test,y_test)
```

Out[42]:
```
0.8888888888888888
```

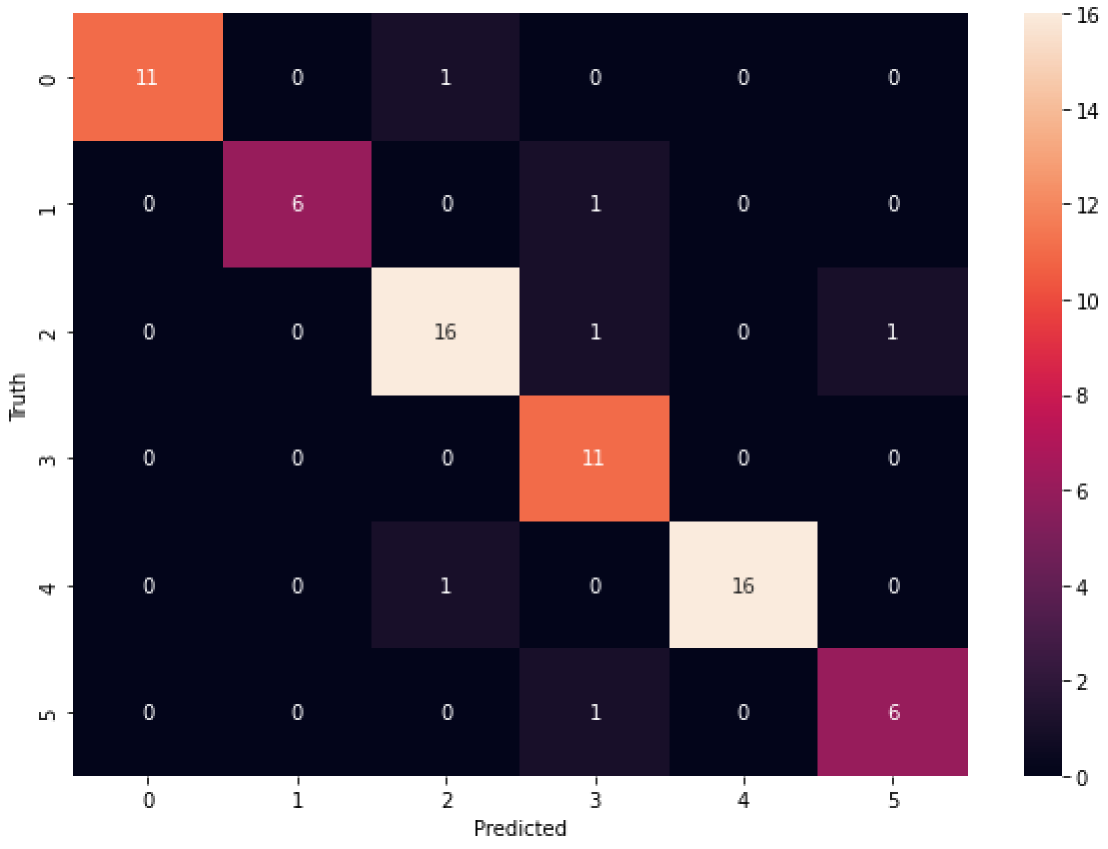
In [43]:
```
best_clf = best_estimators['svm']
```

In [44]:
```
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, best_clf.predict(X_test))
cm
```

Out[44]:
```
array([[11,  0,  1,  0,  0,  0],
       [ 0,  6,  0,  1,  0,  0],
       [ 0,  0, 16,  1,  0,  1],
       [ 0,  0,  0, 11,  0,  0],
       [ 0,  0,  1,  0, 16,  0],
       [ 0,  0,  0,  1,  0,  6]], dtype=int64)
```

In [45]:
```
import seaborn as sn
plt.figure(figsize = (10,7))
sn.heatmap(cm, annot=True)
plt.xlabel('Predicted')
plt.ylabel('Truth')
```

Out[45]:
```
Text(69.0, 0.5, 'Truth')
```



In [46]:
```
class_dict
```

Out[46]:
```
{'Bhavish_Aggarwal': 0,
 'Kunal_Shah': 1,
 'Ritesh_Agarwal': 2,
 'Sachin_Bansal': 3,
 'Shradha_Sharma': 4,
 'Vijay_Sharma': 5}
```

In [48]:
```
!pip install joblib
import joblib
# Save the model as a pickle in a file
joblib.dump(best_clf, 'saved_model.pkl')
```

```
Requirement already satisfied: joblib in c:\users\rhishikesh sonawane\appdata\loca
l\programs\python\python310\lib\site-packages (1.1.0)
```

```
WARNING: There was an error checking the latest version of pip.
```

Out[48]:
```
['saved_model.pkl']
```

In [49]:
```python
import json
with open("class_dictionary.json","w") as f:
    f.write(json.dumps(class_dict))
```

In [ ]: