In [37]:
```
#I used a CodeBasics walkthrough video to complete this project. The foundation of
#The goal of this project was to gain more knowledge on what goes into completing a
#CodeBasics Link: https://youtube.com/playlist?list=PLeo1K3hjS3uvaRHZLl-jLovIjBP14Q
```

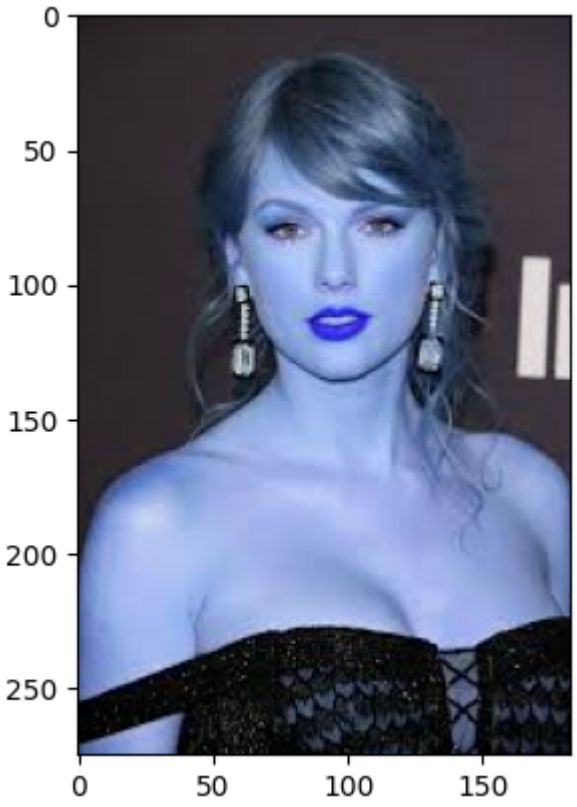
In [1]:
```
#import the important packages
import numpy as np
#cv2 will be the image reader
import cv2
import os
import shutil
#pywt will allow us to do feature extraction
import pywt
import matplotlib
from matplotlib import pyplot as plt
#the following command will allow matplotlib plots to be displayed inside the noteb
%matplotlib inline
```

In [2]:
```
#using a test image to get familiar with the cv2 package
img = cv2.imread(r"C:\Users\rodri\Documents\Image Classification Project\Dataset\Te
#The first number is the x coordinate, the second is the y coordinate, and the last
img.shape
```

Out[2]:
```
(275, 183, 3)
```

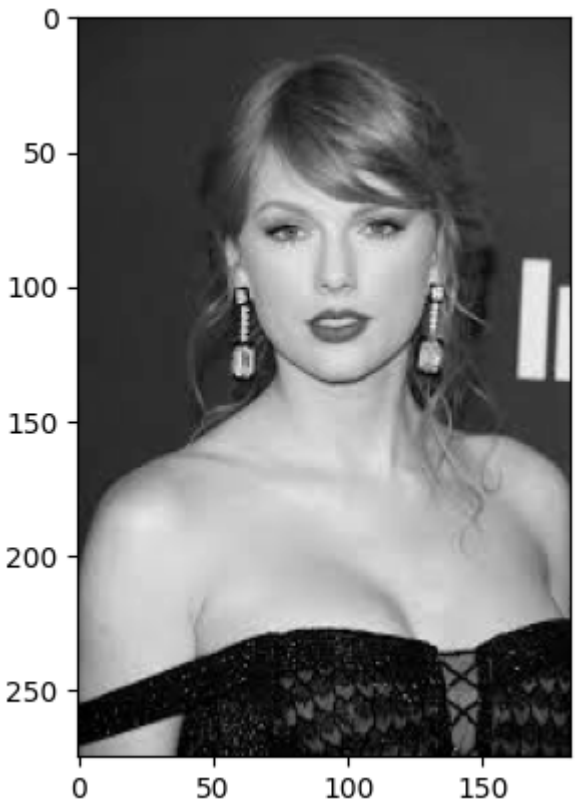
In [3]:
```
#To show the image
plt.imshow(img)
```

Out[3]:
```
<matplotlib.image.AxesImage at 0x2a155b0eb10>
```



In [4]:
```
#OpenCV works with grayscale images. To turn the blue picture gray run:
gray = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
#Then we display the new gray photo
plt.imshow(gray, cmap = 'gray')
```

Out[4]:
```
<matplotlib.image.AxesImage at 0x2a158d56f50>
```

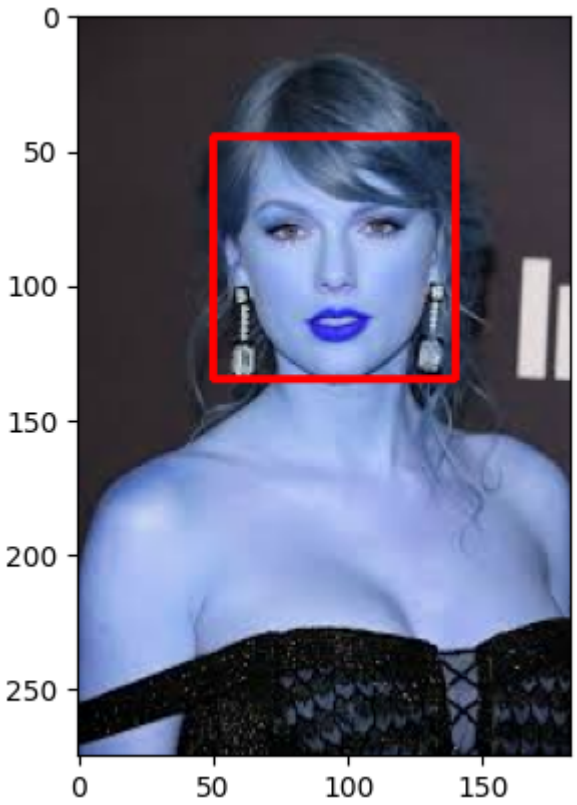


In [5]:
```
#Use the haar cascades from github to get the face recognition documents
face_cascade = cv2.CascadeClassifier(r"C:\Users\rodri\Documents\Image Classificatio
eye_cascade = cv2.CascadeClassifier(r"C:\Users\rodri\Documents\Image Classification
"""The gray referenced below refers to the gray image above, the second number dete
last number number refers to the minimum number of neighboring rectangles required
faces = face_cascade.detectMultiScale(gray,1.3,5)
"""After we run the faces command it will return something like: array([[352, 38, 2
The number of arrays depends on the number of faces. The example above only returne
faces
```

Out[5]:
```
array([[50, 45, 90, 90]])
```

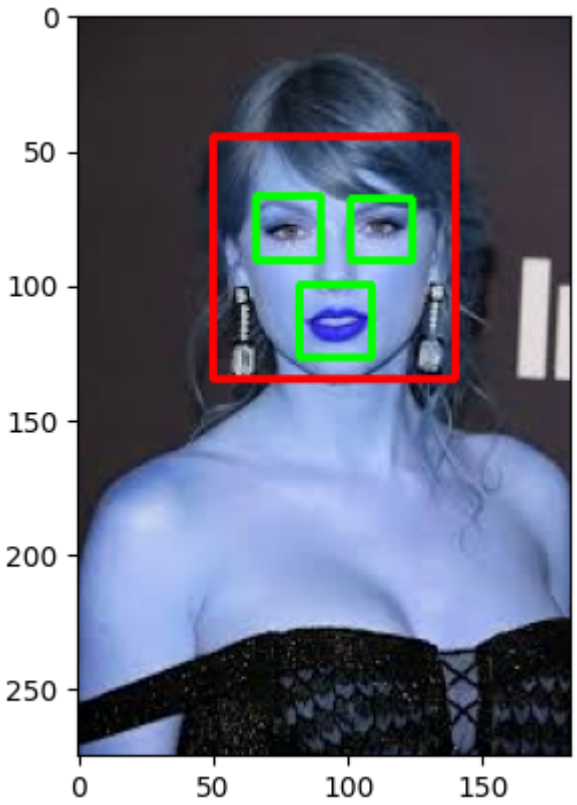
In [6]:
```
#Faces is a 2-d array. To assign labels to faces
(x,y,w,h) = faces[0]
#Drawing a rectangle around the face. We will be doing this to the orginal image th
#The (255,0,0) refers to the RGB colors. For this instance it is Red.
#The last number, 2 in this example, refers to the thickness of the rectangle borde
face_img = cv2.rectangle(img,(x,y),(x+w, y+h),(255,0,0),2)
#Print out the picture
plt.imshow(face_img)
```

Out[6]:
```
<matplotlib.image.AxesImage at 0x2a158d7e850>
```



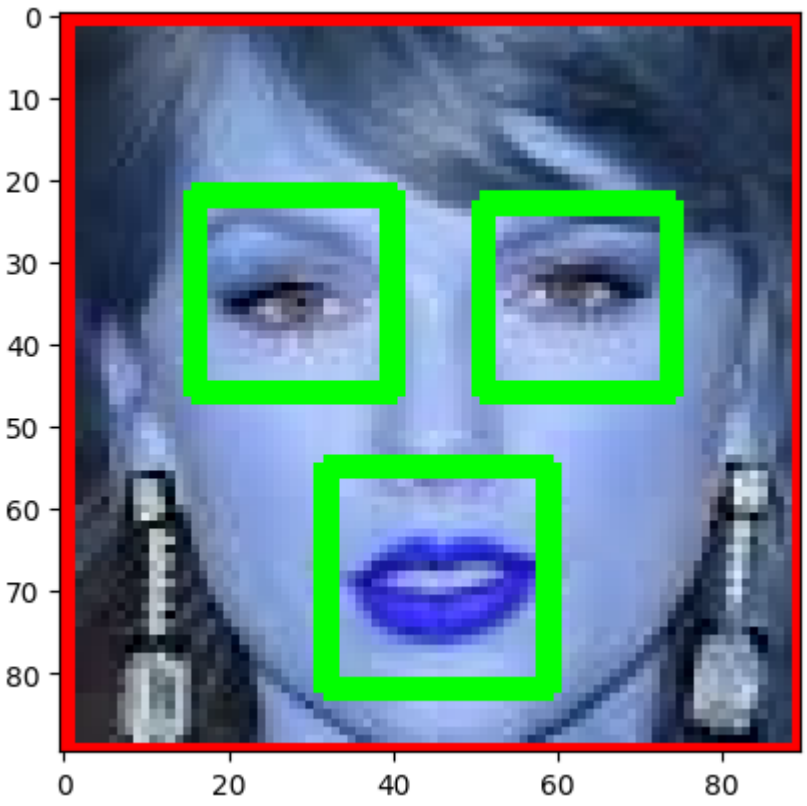
In [7]:

```
#To detect eyes and mouth using haar cascades
cv2.destroyAllWindows()
#(x,y,w,h) refers to the labels we set for 'faces'. 'Faces' gave us the x-coordinat
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)

#Prints out the picture with shapes
plt.figure()
plt.imshow(face_img, cmap='gray')
plt.show()
```



In [8]:
```
#To crop the photo to show the face only do
%matplotlib inline
plt.imshow(roi_color, cmap='gray')
```

Out[8]: <matplotlib.image.AxesImage at 0x2a177ab3fd0>



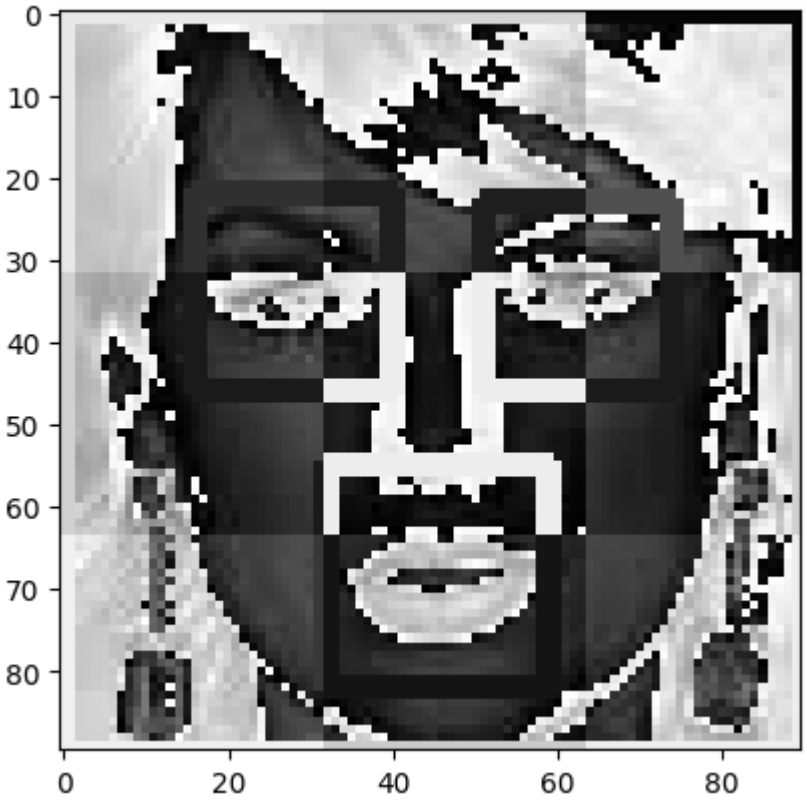
In [9]:
```
cropped_img = np.array(roi_color)
cropped_img.shape
```

Out[9]:
```
(90, 90, 3)
```

In [10]:
```
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 [11]:
```
#This will use
im_har = w2d(cropped_img,'db1',5)
plt.imshow(im_har, cmap='gray')
```

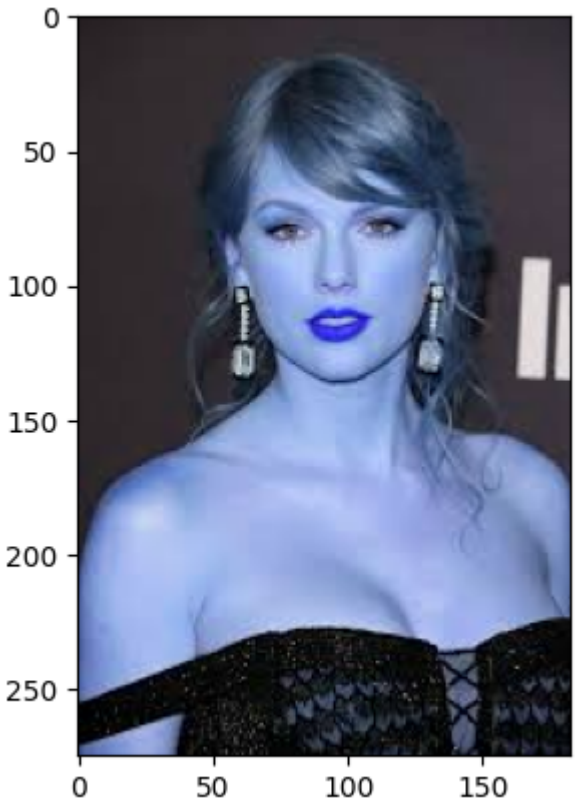
Out[11]:
```
<matplotlib.image.AxesImage at 0x2a177a76850>
```



In [12]:
```
"""If we want an effective way to do this process for multiple images, we would hav
create a function"""
#The following function condenses the process we did above
def get_cropped_image_if_2_eyes(image_path):
    img = cv2.imread(image_path)
    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
```

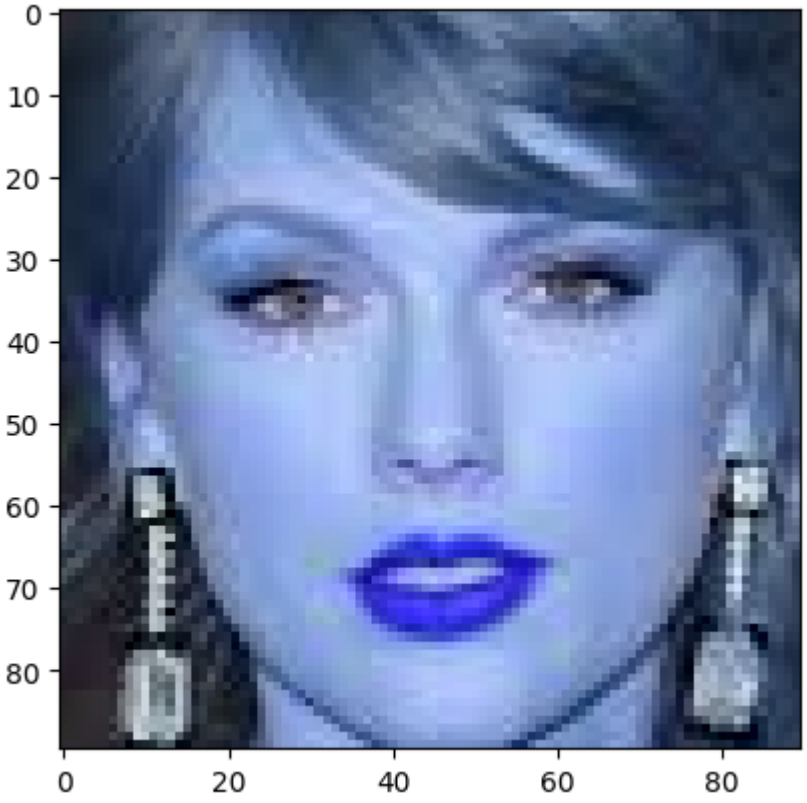
In [13]:
```
original_image = cv2.imread(r"C:\Users\rodri\Documents\Image Classification Project
plt.imshow(original_image)
```

Out[13]: <matplotlib.image.AxesImage at 0x2a177b3d950>



In [14]:
```
cropped_image = get_cropped_image_if_2_eyes(r"C:\Users\rodri\Documents\Image Classi
plt.imshow(cropped_image)
```

Out[14]:
```
<matplotlib.image.AxesImage at 0x2a177b66850>
```



In [15]:
```
#The ./ means the current directory that this notebook is in
path_to_data = r"C:\Users\rodri\Documents\Image Classification Project\Dataset"
path_to_cr_data = r"C:\Users\rodri\Documents\Image Classification Project\Dataset\C
```

In [16]:
```
#For my documents, I organized it so that every artist has their own folder.
#We will need to create a Python list with those folders using the OS module
img_dirs = []
for entry in os.scandir(path_to_data):
    if entry.is_dir():
        img_dirs.append(entry.path)
```

In [17]:
```
img_dirs
```

Out[17]:
```
['C:\\Users\\rodri\\Documents\\Image Classification Project\\Dataset\\Bruno Mars',
 'C:\\Users\\rodri\\Documents\\Image Classification Project\\Dataset\\Carrie Under
wood',
 'C:\\Users\\rodri\\Documents\\Image Classification Project\\Dataset\\Russ',
 'C:\\Users\\rodri\\Documents\\Image Classification Project\\Dataset\\Selena Quint
anilla',
 'C:\\Users\\rodri\\Documents\\Image Classification Project\\Dataset\\Test',
 'C:\\Users\\rodri\\Documents\\Image Classification Project\\Dataset\\Usher']
```

In [18]:
```
#We will be running this code multiple times so we will have to clean out the cropp
if os.path.exists(path_to_cr_data):
    shutil.rmtree(path_to_cr_data)
os.mkdir(path_to_cr_data)
```

In [19]:
```
#cropped_image_dirs will contain the cropped folder path for each of the artists
cropped_image_dirs = []
#celebrity_file_names_dict will contain the image paths
celebrity_file_names_dict = {}

#The following loop will get the name of each celebrity
for img_dir in img_dirs:
    count = 1
    #The delimiter is depending on the folder path that you used when assigning pat
    celebrity_name = img_dir.split('\\')[-1]
    #print(celebrity_name)
    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:  C:\Users\rodri\Documents\Image Classification Project\Dataset\Cropped_Bruno Mars
Generating cropped images in folder:  C:\Users\rodri\Documents\Image Classification Project\Dataset\Cropped_Carrie Underwood
Generating cropped images in folder:  C:\Users\rodri\Documents\Image Classification Project\Dataset\Cropped_Russ
Generating cropped images in folder:  C:\Users\rodri\Documents\Image Classification Project\Dataset\Cropped_Selena Quintanilla
Generating cropped images in folder:  C:\Users\rodri\Documents\Image Classification Project\Dataset\Cropped_Test
Generating cropped images in folder:  C:\Users\rodri\Documents\Image Classification Project\Dataset\Cropped_Usher
```

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

```
{'C:\\Users\\rodri\\Documents\\Image Classification Project\\Dataset\\Cropped_Bruno Mars': ['C:\\Users\\rodri\\Documents\\Image Classification Project\\Dataset\\Cropped_Bruno Mars\\Bruno Mars1.png',
  'C:\\Users\\rodri\\Documents\\Image Classification Project\\Dataset\\Cropped_Bruno Mars\\Bruno Mars10.png',
  'C:\\Users\\rodri\\Documents\\Image Classification Project\\Dataset\\Cropped_Bruno Mars\\Bruno Mars11.png',
  'C:\\Users\\rodri\\Documents\\Image Classification Project\\Dataset\\Cropped_Bruno Mars\\Bruno Mars12.png',
  'C:\\Users\\rodri\\Documents\\Image Classification Project\\Dataset\\Cropped_Bruno Mars\\Bruno Mars13.png',
  'C:\\Users\\rodri\\Documents\\Image Classification Project\\Dataset\\Cropped_Bruno Mars\\Bruno Mars2.png',
  'C:\\Users\\rodri\\Documents\\Image Classification Project\\Dataset\\Cropped_Bruno Mars\\Bruno Mars3.png',
  'C:\\Users\\rodri\\Documents\\Image Classification Project\\Dataset\\Cropped_Bruno Mars\\Bruno Mars4.png',
  'C:\\Users\\rodri\\Documents\\Image Classification Project\\Dataset\\Cropped_Bruno Mars\\Bruno Mars5.png',
  'C:\\Users\\rodri\\Documents\\Image Classification Project\\Dataset\\Cropped_Bruno Mars\\Bruno Mars6.png',
  'C:\\Users\\rodri\\Documents\\Image Classification Project\\Dataset\\Cropped_Bruno Mars\\Bruno Mars7.png',
  'C:\\Users\\rodri\\Documents\\Image Classification Project\\Dataset\\Cropped_Bruno Mars\\Bruno Mars8.png',
  'C:\\Users\\rodri\\Documents\\Image Classification Project\\Dataset\\Cropped_Bruno Mars\\Bruno Mars9.png'],
 'C:\\Users\\rodri\\Documents\\Image Classification Project\\Dataset\\Cropped_Carrie Underwood': ['C:\\Users\\rodri\\Documents\\Image Classification Project\\Dataset\\Cropped_Carrie Underwood\\Carrie Underwood1.png',
  'C:\\Users\\rodri\\Documents\\Image Classification Project\\Dataset\\Cropped_Carrie Underwood\\Carrie Underwood10.png',
  'C:\\Users\\rodri\\Documents\\Image Classification Project\\Dataset\\Cropped_Carrie Underwood\\Carrie Underwood11.png',
  'C:\\Users\\rodri\\Documents\\Image Classification Project\\Dataset\\Cropped_Carrie Underwood\\Carrie Underwood12.png',
  'C:\\Users\\rodri\\Documents\\Image Classification Project\\Dataset\\Cropped_Carrie Underwood\\Carrie Underwood13.png',
  'C:\\Users\\rodri\\Documents\\Image Classification Project\\Dataset\\Cropped_Carrie Underwood\\Carrie Underwood14.png',
  'C:\\Users\\rodri\\Documents\\Image Classification Project\\Dataset\\Cropped_Carrie Underwood\\Carrie Underwood15.png',
  'C:\\Users\\rodri\\Documents\\Image Classification Project\\Dataset\\Cropped_Carrie Underwood\\Carrie Underwood16.png',
  'C:\\Users\\rodri\\Documents\\Image Classification Project\\Dataset\\Cropped_Carrie Underwood\\Carrie Underwood17.png',
  'C:\\Users\\rodri\\Documents\\Image Classification Project\\Dataset\\Cropped_Carrie Underwood\\Carrie Underwood18.png',
  'C:\\Users\\rodri\\Documents\\Image Classification Project\\Dataset\\Cropped_Carrie Underwood\\Carrie Underwood19.png',
  'C:\\Users\\rodri\\Documents\\Image Classification Project\\Dataset\\Cropped_Carrie Underwood\\Carrie Underwood2.png',
  'C:\\Users\\rodri\\Documents\\Image Classification Project\\Dataset\\Cropped_Carrie Underwood\\Carrie Underwood20.png',
  'C:\\Users\\rodri\\Documents\\Image Classification Project\\Dataset\\Cropped_Carrie Underwood\\Carrie Underwood3.png',
```

```
  'C:\\Users\\rodri\\Documents\\Image Classification Project\\Dataset\\Cropped_Car
rie Underwood\\Carrie Underwood4.png',
  'C:\\Users\\rodri\\Documents\\Image Classification Project\\Dataset\\Cropped_Car
rie Underwood\\Carrie Underwood5.png',
  'C:\\Users\\rodri\\Documents\\Image Classification Project\\Dataset\\Cropped_Car
rie Underwood\\Carrie Underwood6.png',
  'C:\\Users\\rodri\\Documents\\Image Classification Project\\Dataset\\Cropped_Car
rie Underwood\\Carrie Underwood7.png',
  'C:\\Users\\rodri\\Documents\\Image Classification Project\\Dataset\\Cropped_Car
rie Underwood\\Carrie Underwood8.png',
  'C:\\Users\\rodri\\Documents\\Image Classification Project\\Dataset\\Cropped_Car
rie Underwood\\Carrie Underwood9.png'],
 'C:\\Users\\rodri\\Documents\\Image Classification Project\\Dataset\\Cropped_Rus
s': ['C:\\Users\\rodri\\Documents\\Image Classification Project\\Dataset\\Cropped_
Russ\\Russ1.png',
  'C:\\Users\\rodri\\Documents\\Image Classification Project\\Dataset\\Cropped_Rus
s\\Russ10.png',
  'C:\\Users\\rodri\\Documents\\Image Classification Project\\Dataset\\Cropped_Rus
s\\Russ11.png',
  'C:\\Users\\rodri\\Documents\\Image Classification Project\\Dataset\\Cropped_Rus
s\\Russ2.png',
  'C:\\Users\\rodri\\Documents\\Image Classification Project\\Dataset\\Cropped_Rus
s\\Russ3.png',
  'C:\\Users\\rodri\\Documents\\Image Classification Project\\Dataset\\Cropped_Rus
s\\Russ4.png',
  'C:\\Users\\rodri\\Documents\\Image Classification Project\\Dataset\\Cropped_Rus
s\\Russ5.png',
  'C:\\Users\\rodri\\Documents\\Image Classification Project\\Dataset\\Cropped_Rus
s\\Russ6.png',
  'C:\\Users\\rodri\\Documents\\Image Classification Project\\Dataset\\Cropped_Rus
s\\Russ7.png',
  'C:\\Users\\rodri\\Documents\\Image Classification Project\\Dataset\\Cropped_Rus
s\\Russ8.png',
  'C:\\Users\\rodri\\Documents\\Image Classification Project\\Dataset\\Cropped_Rus
s\\Russ9.png'],
 'C:\\Users\\rodri\\Documents\\Image Classification Project\\Dataset\\Cropped_Sele
na Quintanilla': ['C:\\Users\\rodri\\Documents\\Image Classification Project\\Data
set\\Cropped_Selena Quintanilla\\Selena Quintanilla1.png',
  'C:\\Users\\rodri\\Documents\\Image Classification Project\\Dataset\\Cropped_Sel
ena Quintanilla\\Selena Quintanilla2.png',
  'C:\\Users\\rodri\\Documents\\Image Classification Project\\Dataset\\Cropped_Sel
ena Quintanilla\\Selena Quintanilla3.png',
  'C:\\Users\\rodri\\Documents\\Image Classification Project\\Dataset\\Cropped_Sel
ena Quintanilla\\Selena Quintanilla4.png',
  'C:\\Users\\rodri\\Documents\\Image Classification Project\\Dataset\\Cropped_Sel
ena Quintanilla\\Selena Quintanilla5.png',
  'C:\\Users\\rodri\\Documents\\Image Classification Project\\Dataset\\Cropped_Sel
ena Quintanilla\\Selena Quintanilla6.png',
  'C:\\Users\\rodri\\Documents\\Image Classification Project\\Dataset\\Cropped_Sel
ena Quintanilla\\Selena Quintanilla7.png'],
 'C:\\Users\\rodri\\Documents\\Image Classification Project\\Dataset\\Cropped_Tes
t': ['C:\\Users\\rodri\\Documents\\Image Classification Project\\Dataset\\Cropped_
Test\\Test1.png'],
 'C:\\Users\\rodri\\Documents\\Image Classification Project\\Dataset\\Cropped_Ushe
r': ['C:\\Users\\rodri\\Documents\\Image Classification Project\\Dataset\\Cropped_
Usher\\Usher1.png',
```

```
  'C:\\Users\\rodri\\Documents\\Image Classification Project\\Dataset\\Cropped_Usher\\Usher10.png',
  'C:\\Users\\rodri\\Documents\\Image Classification Project\\Dataset\\Cropped_Usher\\Usher11.png',
  'C:\\Users\\rodri\\Documents\\Image Classification Project\\Dataset\\Cropped_Usher\\Usher12.png',
  'C:\\Users\\rodri\\Documents\\Image Classification Project\\Dataset\\Cropped_Usher\\Usher2.png',
  'C:\\Users\\rodri\\Documents\\Image Classification Project\\Dataset\\Cropped_Usher\\Usher3.png',
  'C:\\Users\\rodri\\Documents\\Image Classification Project\\Dataset\\Cropped_Usher\\Usher4.png',
  'C:\\Users\\rodri\\Documents\\Image Classification Project\\Dataset\\Cropped_Usher\\Usher5.png',
  'C:\\Users\\rodri\\Documents\\Image Classification Project\\Dataset\\Cropped_Usher\\Usher6.png',
  'C:\\Users\\rodri\\Documents\\Image Classification Project\\Dataset\\Cropped_Usher\\Usher7.png',
  'C:\\Users\\rodri\\Documents\\Image Classification Project\\Dataset\\Cropped_Usher\\Usher8.png',
  'C:\\Users\\rodri\\Documents\\Image Classification Project\\Dataset\\Cropped_Usher\\Usher9.png']}
```

In [21]:
```
#This for-loop will have the celebrity names be the key and assign a number as the
class_dict = {}
count = 0
for celebrity_name in celebrity_file_names_dict.keys():
    class_dict[celebrity_name] = count
    count = count + 1
class_dict
```

Out[21]:
```
{'C:\\Users\\rodri\\Documents\\Image Classification Project\\Dataset\\Cropped_Bruno Mars': 0,
 'C:\\Users\\rodri\\Documents\\Image Classification Project\\Dataset\\Cropped_Carrie Underwood': 1,
 'C:\\Users\\rodri\\Documents\\Image Classification Project\\Dataset\\Cropped_Russ': 2,
 'C:\\Users\\rodri\\Documents\\Image Classification Project\\Dataset\\Cropped_Selena Quintanilla': 3,
 'C:\\Users\\rodri\\Documents\\Image Classification Project\\Dataset\\Cropped_Test': 4,
 'C:\\Users\\rodri\\Documents\\Image Classification Project\\Dataset\\Cropped_Usher': 5}
```

In [26]:
```
X, y = [], []
for celebrity_name, training_files in celebrity_file_names_dict.items():
    for training_image in training_files:
        img = cv2.imread(training_image)
        #Scaling
        scalled_raw_img = cv2.resize(img, (32, 32))
        img_har = w2d(img,'db1',5)
        scalled_img_har = cv2.resize(img_har, (32, 32))
        #Vertically Stacking
        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 [27]:
```
#We got 4096 because that is the length of each image when you run len(X[0])
#The command X.shape should get you (162,4096)
X = np.array(X).reshape(len(X),4096).astype(float)
X.shape
```

Out[27]: (64, 4096)

In [28]:
```
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 [30]:
```
#X is the image and y is the name
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 will train the model on x_train and y_train
pipe.fit(X_train, y_train)
#pipe.score will check how good your model is. It will return a decimal like .33333
pipe.score(X_test, y_test)
```

Out[30]: 0.625

In [31]:
```
#Classification report
#It will predict y_test using X_test
print(classification_report(y_test, pipe.predict(X_test)))
```

```
              precision    recall  f1-score   support

           0       0.38      1.00      0.55         3
           1       0.83      1.00      0.91         5
           2       1.00      0.25      0.40         4
           3       0.00      0.00      0.00         1
           5       1.00      0.33      0.50         3

    accuracy                           0.62        16
   macro avg       0.64      0.52      0.47        16
weighted avg       0.77      0.62      0.58        16
```

```
C:\Users\rodri\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\met
rics\_classification.py:1509: UndefinedMetricWarning: Precision is ill-defined and b
eing set to 0.0 in labels with no predicted samples. Use `zero_division` parameter t
o control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
C:\Users\rodri\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\met
rics\_classification.py:1509: UndefinedMetricWarning: Precision is ill-defined and b
eing set to 0.0 in labels with no predicted samples. Use `zero_division` parameter t
o control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
C:\Users\rodri\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\met
rics\_classification.py:1509: UndefinedMetricWarning: Precision is ill-defined and b
eing set to 0.0 in labels with no predicted samples. Use `zero_division` parameter t
o control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
```

In [32]:

```
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 [33]:

```
#Trying out different models with different parameters
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 [34]:

```
scores = []
best_estimators = {}
import pandas as pd
for algo, mp in model_params.items():
    pipe = make_pipeline(StandardScaler(), mp['model'])
    #Cross validating
    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
```

```
C:\Users\rodri\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\model_selection\_split.py:737: UserWarning: The least populated class in y has only 1 members, which is less than n_splits=5.
  warnings.warn(
C:\Users\rodri\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\model_selection\_split.py:737: UserWarning: The least populated class in y has only 1 members, which is less than n_splits=5.
  warnings.warn(
C:\Users\rodri\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\model_selection\_split.py:737: UserWarning: The least populated class in y has only 1 members, which is less than n_splits=5.
  warnings.warn(
```

Out[34]:

	model	best_score	best_params
0	svm	0.642222	{'svc__C': 1, 'svc__kernel': 'linear'}
1	random_forest	0.557778	{'randomforestclassifier__n_estimators': 10}
2	logistic_regression	0.584444	{'logisticregression__C': 1}

In [38]:
```
best_estimators['svm'].score(X_test,y_test)
```

Out[38]:
```
0.75
```

In [39]:
```
best_estimators['random_forest'].score(X_test, y_test)
```

Out[39]:
```
0.6875
```

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

Out[40]:
```
0.5625
```

In [41]:
```
#Out of the above models, svm had the highest number
best_clf = best_estimators['svm']
```

In [42]:
```
#Now we will create a confusion_matrix. This shows how many times one person was ac
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, best_clf.predict(X_test))
cm
```

Out[42]:
```
array([[2, 0, 0, 1, 0],
       [0, 5, 0, 0, 0],
       [1, 0, 2, 0, 1],
       [0, 0, 1, 0, 0],
       [0, 0, 0, 0, 3]], dtype=int64)
```

In [43]:
```
#Saving the trained model
import joblib
joblib.dump(best_clf, 'saved_model.pkl')
```

Out[43]:
```
['saved_model.pkl']
```

In [45]:
```
#Save class dictionary useful for the Python Flask Server
import json
with open("class_dictionary.json","w") as f:
    f.write(json.dumps(class_dict))
```

In [ ]:
```

```