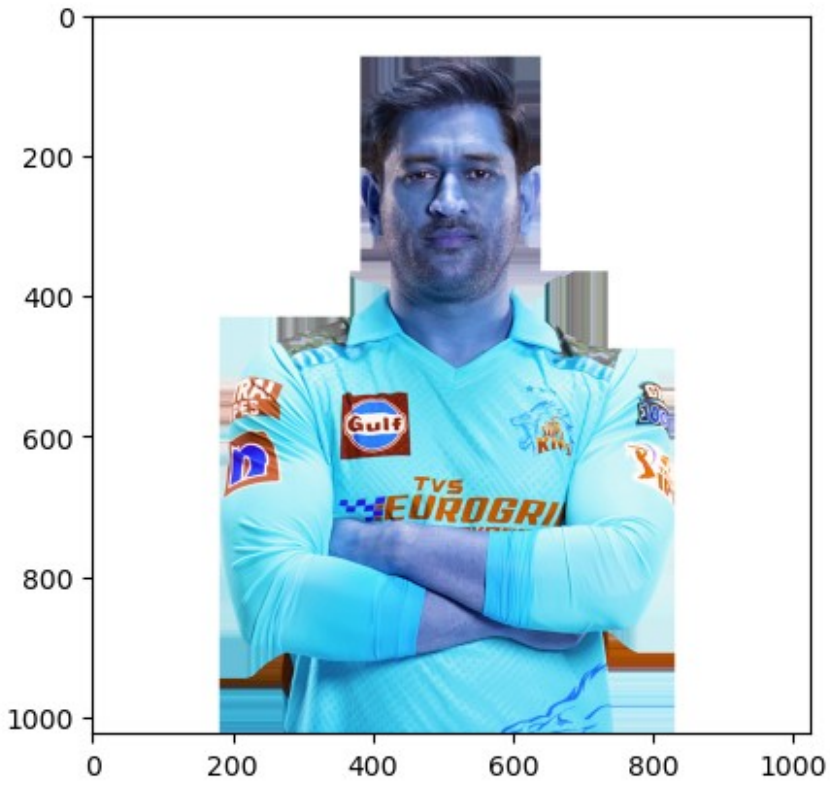
```
import numpy as np
import cv2
import matplotlib
from matplotlib import pyplot as plt
%matplotlib inline

img = cv2.imread('./test_images/msd2.jpg')
img.shape
```

```
(1024, 1024, 3)
```

```
plt.imshow(img)
```

```
<matplotlib.image.AxesImage at 0x2b49d50df10>
```




```
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
gray.shape
```

```
(1024, 1024)
```

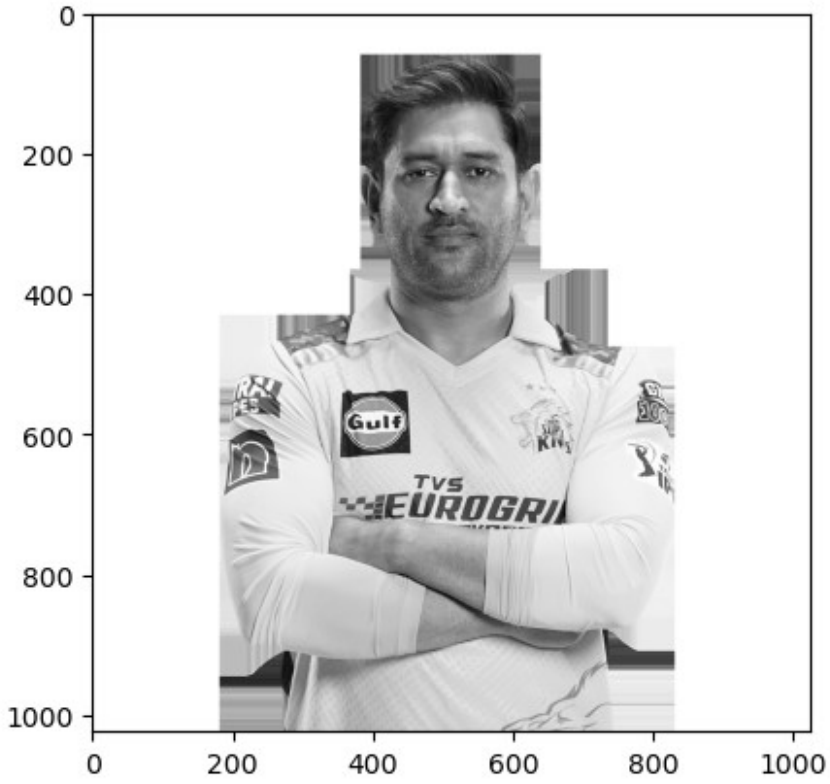
```
gray
```

```
array([[255, 255, 255, ..., 255, 255, 255],
       [255, 255, 255, ..., 255, 255, 255],
       [255, 255, 255, ..., 255, 255, 255],
       ...,
```

```
       [255, 255, 255, ..., 255, 255, 255],
       [255, 255, 255, ..., 255, 255, 255],
       [255, 255, 255, ..., 255, 255, 255]], dtype=uint8)
```

```
plt.imshow(gray, cmap='gray')
```

```
<matplotlib.image.AxesImage at 0x2b49d56de90>
```




```
face_cascade =
cv2.CascadeClassifier('./opencv/haarcascades/haarcascade_frontalface_d
efault.xml')
eye_cascade =
cv2.CascadeClassifier('./opencv/haarcascades/haarcascade_eye.xml')

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

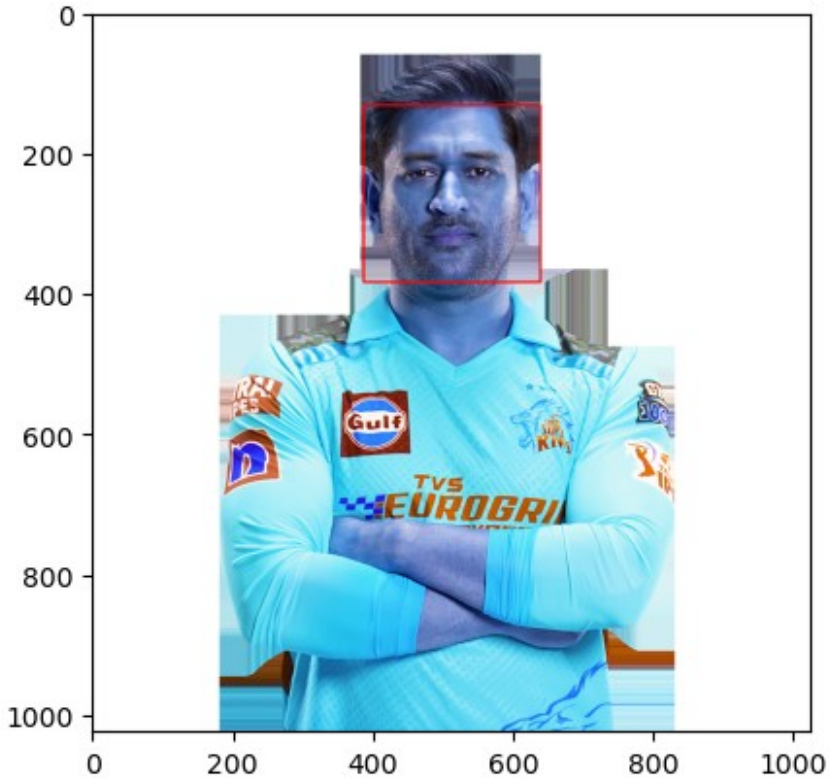
```
array([[387, 132, 252, 252]])
```

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

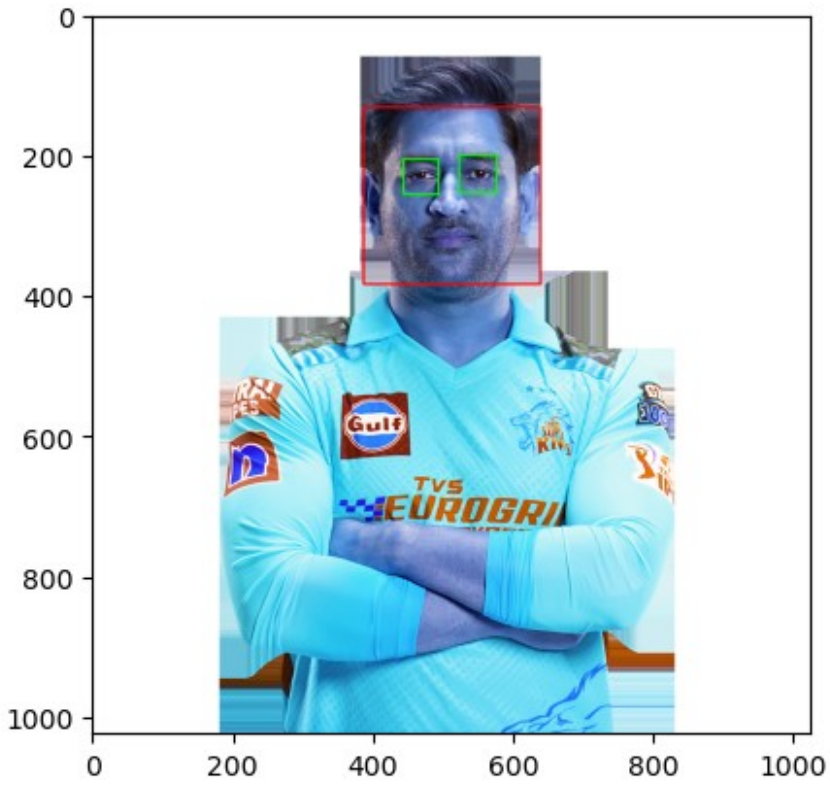
```
(387, 132, 252, 252)
```

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

```
<matplotlib.image.AxesImage at 0x2b49efdfb50>
```

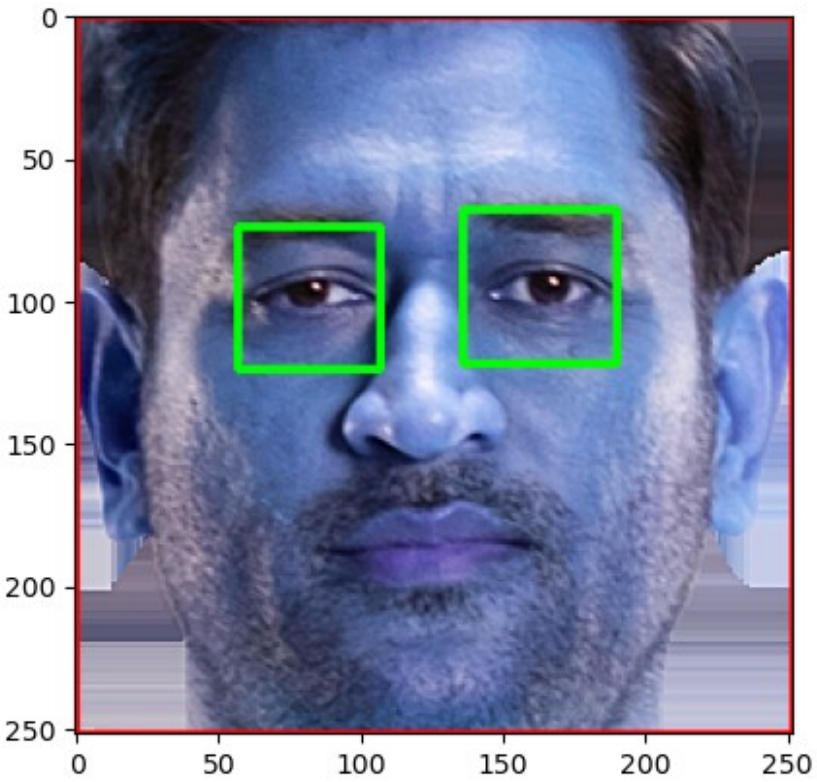



```
cv2.destroyAllWindows()
for (x,y,w,h) in faces:
    face_img = cv2.rectangle(img,(x,y),(x+w,y+h),(255,0,0),2)
    roi_gray = gray[y:y+h, x:x+w]
    roi_color = face_img[y:y+h, x:x+w]
    eyes = eye_cascade.detectMultiScale(roi_gray)
    for (ex,ey,ew,eh) in eyes:
        cv2.rectangle(roi_color,(ex,ey),(ex+ew,ey+eh),(0,255,0),2)


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



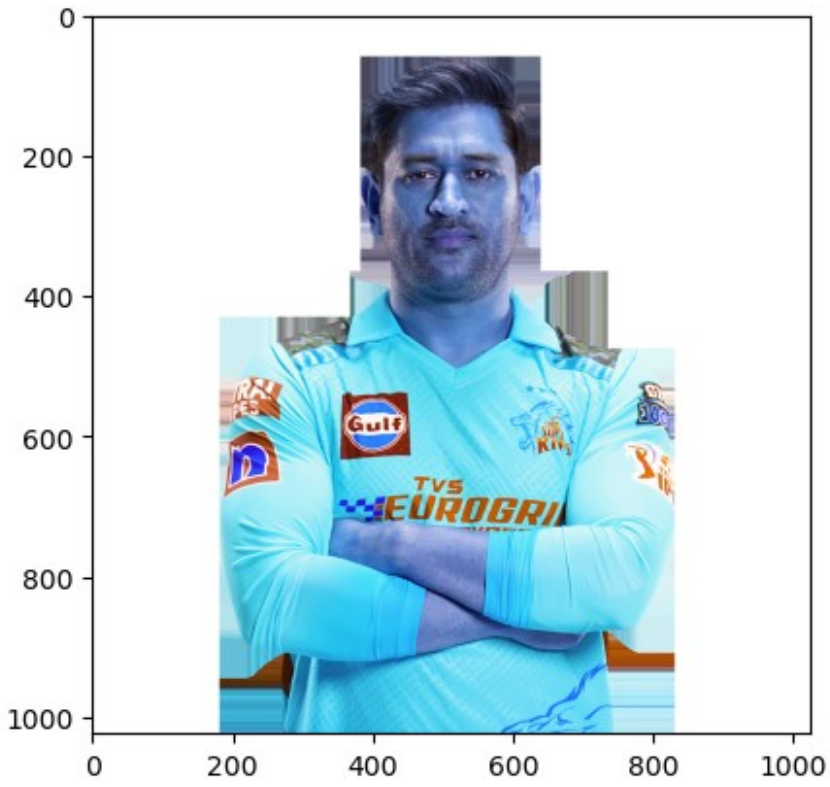

```
%matplotlib inline
plt.imshow(roi_color, cmap='gray')
```

```
<matplotlib.image.AxesImage at 0x2b48ee066d0>
```



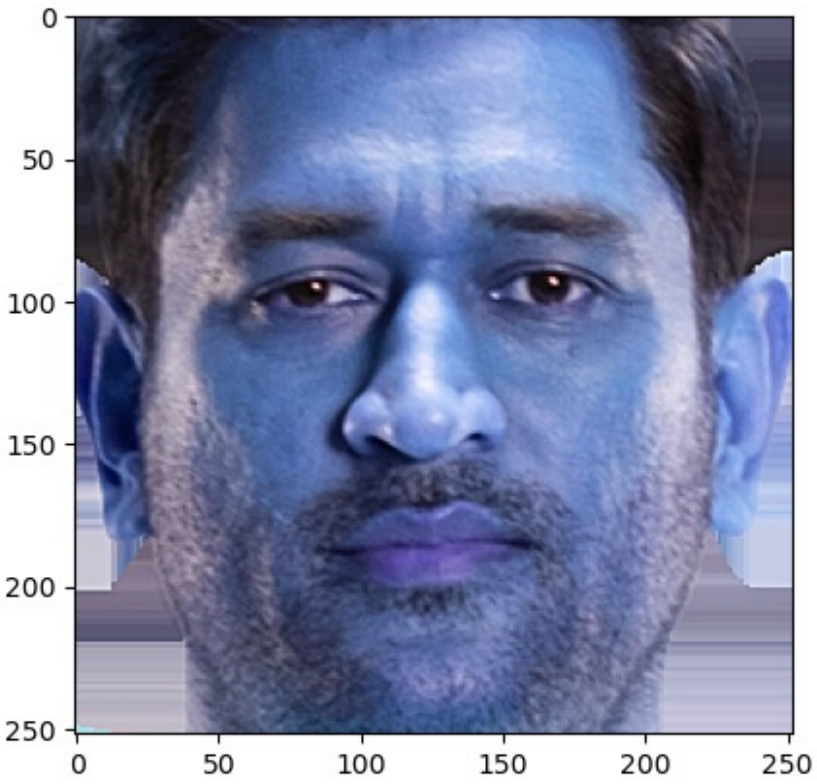

```
def get_cropped_image_if_2_eyes(image_path):
    img = cv2.imread(image_path)
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    faces = face_cascade.detectMultiScale(gray, 1.3, 5)
    for (x,y,w,h) in faces:
        roi_gray = gray[y:y+h, x:x+w]
        roi_color = img[y:y+h, x:x+w]
        eyes = eye_cascade.detectMultiScale(roi_gray)
        if len(eyes) >= 2:
            return roi_color

original_image = cv2.imread('./test_images/msd2.jpg')
plt.imshow(original_image)
```

```
<matplotlib.image.AxesImage at 0x2b49f03de90>
```



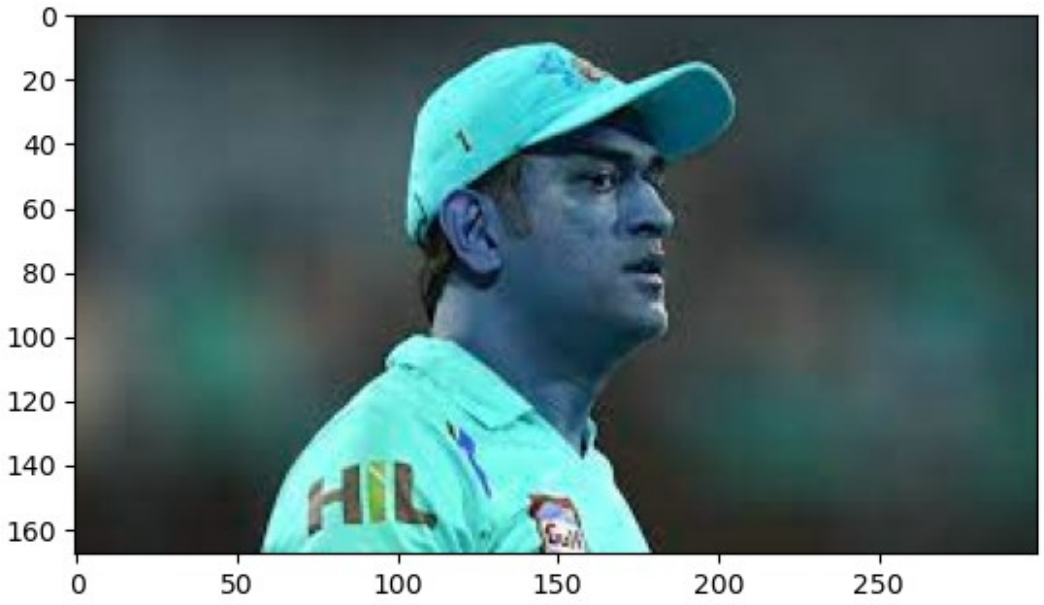

```
cropped_image = get_cropped_image_if_2_eyes('./test_images/msd2.jpg')
plt.imshow(cropped_image)
```

```
<matplotlib.image.AxesImage at 0x2b49f0a5f10>
```




```
org_image_obstructed = cv2.imread('./test_images/msd1.jpg')
plt.imshow(org_image_obstructed)
```

```
<matplotlib.image.AxesImage at 0x2b49b481a50>
```



```
cropped_image_no_2_eyes =
get_cropped_image_if_2_eyes('./test_images/msd1.jpg')
cropped_image_no_2_eyes

path_to_data = "./dataset/"
path_to_cr_data = "./dataset/cropped/"

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

img_dirs
```

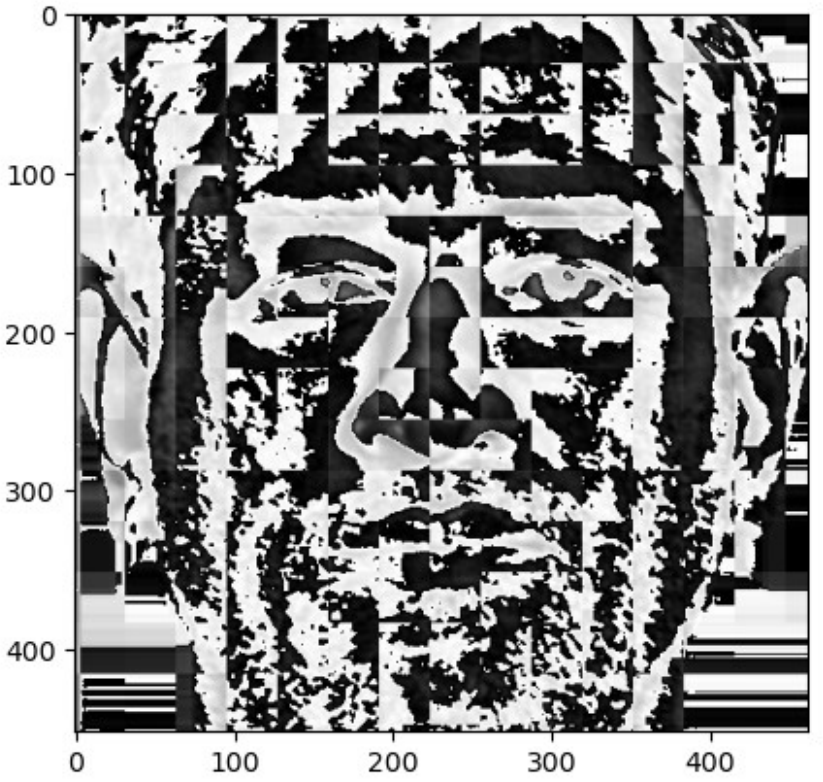
```
['./dataset/lionel_messi',
 './dataset/maria_sharapova',
 './dataset/ms_dhoni',
 './dataset/roger_federer',
 './dataset/serena_williams']
```

```
import shutil
if os.path.exists(path_to_cr_data):
     shutil.rmtree(path_to_cr_data)
os.mkdir(path_to_cr_data)

cropped_image_dirs = []
celebrity_file_names_dict = {}

for img_dir in img_dirs:
    count = 1
    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

lionel_messi
Generating cropped images in folder:  ./dataset/cropped/lionel_messi
maria_sharapova
Generating cropped images in folder:
./dataset/cropped/maria_sharapova
ms_dhoni
Generating cropped images in folder:  ./dataset/cropped/ms_dhoni
roger_federer
Generating cropped images in folder:  ./dataset/cropped/roger_federer
serena_williams
Generating cropped images in folder:
./dataset/cropped/serena_williams
```

```
import cv2
import matplotlib.pyplot as plt
import numpy as np
import pywt

def w2d(img, mode='haar', level=1):
    # Convert to grayscale if the image is not already in grayscale
    if len(img.shape) == 3:
        img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

    # Ensure that img is in the correct data type (float32)
    imArray = np.float32(img)
    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

# Load the image 'sharapova1.jpg'
cropped_img = cv2.imread('./test_images/msd5.jpg')

# Apply wavelet transformation and display the result
im_har = w2d(cropped_img, 'db1', 5)
```

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




```
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
```

```
{'lionel_messi': ['./dataset/cropped/lionel_messi\\lionel_messi1.png',
  './dataset/cropped/lionel_messi\\lionel_messi10.png',
  './dataset/cropped/lionel_messi\\lionel_messi11.png',
  './dataset/cropped/lionel_messi\\lionel_messi12.png',
  './dataset/cropped/lionel_messi\\lionel_messi13.png',
  './dataset/cropped/lionel_messi\\lionel_messi14.png',
  './dataset/cropped/lionel_messi\\lionel_messi15.png',
  './dataset/cropped/lionel_messi\\lionel_messi16.png',
  './dataset/cropped/lionel_messi\\lionel_messi17.png',
  './dataset/cropped/lionel_messi\\lionel_messi18.png',
  './dataset/cropped/lionel_messi\\lionel_messi19.png',
  './dataset/cropped/lionel_messi\\lionel_messi2.png',
```

```
  './dataset/cropped/lionel_messi\\lionel_messi20.png',
  './dataset/cropped/lionel_messi\\lionel_messi21.png',
  './dataset/cropped/lionel_messi\\lionel_messi22.png',
  './dataset/cropped/lionel_messi\\lionel_messi23.png',
  './dataset/cropped/lionel_messi\\lionel_messi24.png',
  './dataset/cropped/lionel_messi\\lionel_messi25.png',
  './dataset/cropped/lionel_messi\\lionel_messi26.png',
  './dataset/cropped/lionel_messi\\lionel_messi27.png',
  './dataset/cropped/lionel_messi\\lionel_messi28.png',
  './dataset/cropped/lionel_messi\\lionel_messi29.png',
  './dataset/cropped/lionel_messi\\lionel_messi3.png',
  './dataset/cropped/lionel_messi\\lionel_messi30.png',
  './dataset/cropped/lionel_messi\\lionel_messi31.png',
  './dataset/cropped/lionel_messi\\lionel_messi32.png',
  './dataset/cropped/lionel_messi\\lionel_messi33.png',
  './dataset/cropped/lionel_messi\\lionel_messi34.png',
  './dataset/cropped/lionel_messi\\lionel_messi35.png',
  './dataset/cropped/lionel_messi\\lionel_messi36.png',
  './dataset/cropped/lionel_messi\\lionel_messi37.png',
  './dataset/cropped/lionel_messi\\lionel_messi38.png',
  './dataset/cropped/lionel_messi\\lionel_messi39.png',
  './dataset/cropped/lionel_messi\\lionel_messi4.png',
  './dataset/cropped/lionel_messi\\lionel_messi40.png',
  './dataset/cropped/lionel_messi\\lionel_messi41.png',
  './dataset/cropped/lionel_messi\\lionel_messi42.png',
  './dataset/cropped/lionel_messi\\lionel_messi5.png',
  './dataset/cropped/lionel_messi\\lionel_messi6.png',
  './dataset/cropped/lionel_messi\\lionel_messi7.png',
  './dataset/cropped/lionel_messi\\lionel_messi8.png',
  './dataset/cropped/lionel_messi\\lionel_messi9.png'],
 'maria_sharapova': ['./dataset/cropped/maria_sharapova\\
maria_sharapova1.png',
  './dataset/cropped/maria_sharapova\\maria_sharapova10.png',
  './dataset/cropped/maria_sharapova\\maria_sharapova11.png',
  './dataset/cropped/maria_sharapova\\maria_sharapova12.png',
  './dataset/cropped/maria_sharapova\\maria_sharapova13.png',
  './dataset/cropped/maria_sharapova\\maria_sharapova14.png',
  './dataset/cropped/maria_sharapova\\maria_sharapova15.png',
  './dataset/cropped/maria_sharapova\\maria_sharapova16.png',
  './dataset/cropped/maria_sharapova\\maria_sharapova17.png',
  './dataset/cropped/maria_sharapova\\maria_sharapova18.png',
  './dataset/cropped/maria_sharapova\\maria_sharapova19.png',
  './dataset/cropped/maria_sharapova\\maria_sharapova2.png',
  './dataset/cropped/maria_sharapova\\maria_sharapova20.png',
  './dataset/cropped/maria_sharapova\\maria_sharapova21.png',
  './dataset/cropped/maria_sharapova\\maria_sharapova22.png',
  './dataset/cropped/maria_sharapova\\maria_sharapova23.png',
  './dataset/cropped/maria_sharapova\\maria_sharapova24.png',
  './dataset/cropped/maria_sharapova\\maria_sharapova25.png',
```

```
  './dataset/cropped/maria_sharapova\\maria_sharapova26.png',
  './dataset/cropped/maria_sharapova\\maria_sharapova27.png',
  './dataset/cropped/maria_sharapova\\maria_sharapova28.png',
  './dataset/cropped/maria_sharapova\\maria_sharapova29.png',
  './dataset/cropped/maria_sharapova\\maria_sharapova3.png',
  './dataset/cropped/maria_sharapova\\maria_sharapova30.png',
  './dataset/cropped/maria_sharapova\\maria_sharapova31.png',
  './dataset/cropped/maria_sharapova\\maria_sharapova32.png',
  './dataset/cropped/maria_sharapova\\maria_sharapova33.png',
  './dataset/cropped/maria_sharapova\\maria_sharapova34.png',
  './dataset/cropped/maria_sharapova\\maria_sharapova35.png',
  './dataset/cropped/maria_sharapova\\maria_sharapova36.png',
  './dataset/cropped/maria_sharapova\\maria_sharapova37.png',
  './dataset/cropped/maria_sharapova\\maria_sharapova38.png',
  './dataset/cropped/maria_sharapova\\maria_sharapova39.png',
  './dataset/cropped/maria_sharapova\\maria_sharapova4.png',
  './dataset/cropped/maria_sharapova\\maria_sharapova40.png',
  './dataset/cropped/maria_sharapova\\maria_sharapova41.png',
  './dataset/cropped/maria_sharapova\\maria_sharapova42.png',
  './dataset/cropped/maria_sharapova\\maria_sharapova43.png',
  './dataset/cropped/maria_sharapova\\maria_sharapova44.png',
  './dataset/cropped/maria_sharapova\\maria_sharapova45.png',
  './dataset/cropped/maria_sharapova\\maria_sharapova46.png',
  './dataset/cropped/maria_sharapova\\maria_sharapova47.png',
  './dataset/cropped/maria_sharapova\\maria_sharapova5.png',
  './dataset/cropped/maria_sharapova\\maria_sharapova6.png',
  './dataset/cropped/maria_sharapova\\maria_sharapova7.png',
  './dataset/cropped/maria_sharapova\\maria_sharapova8.png',
  './dataset/cropped/maria_sharapova\\maria_sharapova9.png'],
 'ms_dhoni': ['./dataset/cropped/ms_dhoni\\ms_dhoni1.png',
  './dataset/cropped/ms_dhoni\\ms_dhoni10.png',
  './dataset/cropped/ms_dhoni\\ms_dhoni11.png',
  './dataset/cropped/ms_dhoni\\ms_dhoni12.png',
  './dataset/cropped/ms_dhoni\\ms_dhoni13.png',
  './dataset/cropped/ms_dhoni\\ms_dhoni14.png',
  './dataset/cropped/ms_dhoni\\ms_dhoni15.png',
  './dataset/cropped/ms_dhoni\\ms_dhoni16.png',
  './dataset/cropped/ms_dhoni\\ms_dhoni17.png',
  './dataset/cropped/ms_dhoni\\ms_dhoni18.png',
  './dataset/cropped/ms_dhoni\\ms_dhoni19.png',
  './dataset/cropped/ms_dhoni\\ms_dhoni2.png',
  './dataset/cropped/ms_dhoni\\ms_dhoni20.png',
  './dataset/cropped/ms_dhoni\\ms_dhoni21.png',
  './dataset/cropped/ms_dhoni\\ms_dhoni22.png',
  './dataset/cropped/ms_dhoni\\ms_dhoni23.png',
  './dataset/cropped/ms_dhoni\\ms_dhoni24.png',
  './dataset/cropped/ms_dhoni\\ms_dhoni25.png',
  './dataset/cropped/ms_dhoni\\ms_dhoni26.png',
  './dataset/cropped/ms_dhoni\\ms_dhoni3.png',
```

```
  './dataset/cropped/ms_dhoni\\ms_dhoni4.png',
  './dataset/cropped/ms_dhoni\\ms_dhoni5.png',
  './dataset/cropped/ms_dhoni\\ms_dhoni6.png',
  './dataset/cropped/ms_dhoni\\ms_dhoni7.png',
  './dataset/cropped/ms_dhoni\\ms_dhoni8.png',
  './dataset/cropped/ms_dhoni\\ms_dhoni9.png'],
 'roger_federer': ['./dataset/cropped/roger_federer\\
roger_federer1.png',
  './dataset/cropped/roger_federer\\roger_federer10.png',
  './dataset/cropped/roger_federer\\roger_federer11.png',
  './dataset/cropped/roger_federer\\roger_federer12.png',
  './dataset/cropped/roger_federer\\roger_federer13.png',
  './dataset/cropped/roger_federer\\roger_federer14.png',
  './dataset/cropped/roger_federer\\roger_federer15.png',
  './dataset/cropped/roger_federer\\roger_federer16.png',
  './dataset/cropped/roger_federer\\roger_federer17.png',
  './dataset/cropped/roger_federer\\roger_federer18.png',
  './dataset/cropped/roger_federer\\roger_federer19.png',
  './dataset/cropped/roger_federer\\roger_federer2.png',
  './dataset/cropped/roger_federer\\roger_federer20.png',
  './dataset/cropped/roger_federer\\roger_federer21.png',
  './dataset/cropped/roger_federer\\roger_federer22.png',
  './dataset/cropped/roger_federer\\roger_federer23.png',
  './dataset/cropped/roger_federer\\roger_federer24.png',
  './dataset/cropped/roger_federer\\roger_federer25.png',
  './dataset/cropped/roger_federer\\roger_federer26.png',
  './dataset/cropped/roger_federer\\roger_federer27.png',
  './dataset/cropped/roger_federer\\roger_federer28.png',
  './dataset/cropped/roger_federer\\roger_federer29.png',
  './dataset/cropped/roger_federer\\roger_federer3.png',
  './dataset/cropped/roger_federer\\roger_federer30.png',
  './dataset/cropped/roger_federer\\roger_federer31.png',
  './dataset/cropped/roger_federer\\roger_federer4.png',
  './dataset/cropped/roger_federer\\roger_federer5.png',
  './dataset/cropped/roger_federer\\roger_federer6.png',
  './dataset/cropped/roger_federer\\roger_federer7.png',
  './dataset/cropped/roger_federer\\roger_federer8.png',
  './dataset/cropped/roger_federer\\roger_federer9.png'],
 'serena_williams': ['./dataset/cropped/serena_williams\\
serena_williams1.png',
  './dataset/cropped/serena_williams\\serena_williams10.png',
  './dataset/cropped/serena_williams\\serena_williams11.png',
  './dataset/cropped/serena_williams\\serena_williams12.png',
  './dataset/cropped/serena_williams\\serena_williams13.png',
  './dataset/cropped/serena_williams\\serena_williams14.png',
  './dataset/cropped/serena_williams\\serena_williams15.png',
  './dataset/cropped/serena_williams\\serena_williams16.png',
  './dataset/cropped/serena_williams\\serena_williams17.png',
  './dataset/cropped/serena_williams\\serena_williams18.png',
```

```
  './dataset/cropped/serena_williams\\serena_williams19.png',
  './dataset/cropped/serena_williams\\serena_williams2.png',
  './dataset/cropped/serena_williams\\serena_williams20.png',
  './dataset/cropped/serena_williams\\serena_williams21.png',
  './dataset/cropped/serena_williams\\serena_williams22.png',
  './dataset/cropped/serena_williams\\serena_williams23.png',
  './dataset/cropped/serena_williams\\serena_williams24.png',
  './dataset/cropped/serena_williams\\serena_williams25.png',
  './dataset/cropped/serena_williams\\serena_williams26.png',
  './dataset/cropped/serena_williams\\serena_williams27.png',
  './dataset/cropped/serena_williams\\serena_williams28.png',
  './dataset/cropped/serena_williams\\serena_williams29.png',
  './dataset/cropped/serena_williams\\serena_williams3.png',
  './dataset/cropped/serena_williams\\serena_williams30.png',
  './dataset/cropped/serena_williams\\serena_williams31.png',
  './dataset/cropped/serena_williams\\serena_williams32.png',
  './dataset/cropped/serena_williams\\serena_williams33.png',
  './dataset/cropped/serena_williams\\serena_williams34.png',
  './dataset/cropped/serena_williams\\serena_williams35.png',
  './dataset/cropped/serena_williams\\serena_williams36.png',
  './dataset/cropped/serena_williams\\serena_williams4.png',
  './dataset/cropped/serena_williams\\serena_williams5.png',
  './dataset/cropped/serena_williams\\serena_williams6.png',
  './dataset/cropped/serena_williams\\serena_williams7.png',
  './dataset/cropped/serena_williams\\serena_williams8.png',
  './dataset/cropped/serena_williams\\serena_williams9.png']}
```

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

```
{'lionel_messi': 0,
 'maria_sharapova': 1,
 'ms_dhoni': 2,
 'roger_federer': 3,
 'serena_williams': 4}
```

```
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_har.reshape(
```

```
32*32,1)))
        X.append(combined_img)
        y.append(class_dict[celebrity_name])
```

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

```
4096
```

```
X = np.array(X).reshape(len(X),4096).astype(float)
X.shape
```

```
(182, 4096)
```

```
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

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)
```

```
0.5217391304347826
```

```
print(classification_report(y_test, pipe.predict(X_test)))
```

```
              precision    recall  f1-score   support

           0       0.60      0.60      0.60        10
           1       0.67      0.77      0.71        13
           2       0.67      0.22      0.33         9
           3       0.29      0.40      0.33         5
           4       0.36      0.44      0.40         9

    accuracy                           0.52        46
   macro avg       0.52      0.49      0.48        46
weighted avg       0.55      0.52      0.51        46
```

```
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

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]
        }
    }
}

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
```

	model	best_score	best_params
0	svm	0.661640	{'svc__C': 1, 'svc__kernel': 'linear'}
1	random_forest	0.552116	{'randomforestclassifier__n_estimators': 10}
2	logistic_regression	0.691270	{'logisticregression__C': 1}

```
best_estimators
```

```
{'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=1, solver='liblinear'))])}
```

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

```
0.5652173913043478
```

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

```
0.5217391304347826
```

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

```
0.6739130434782609
```

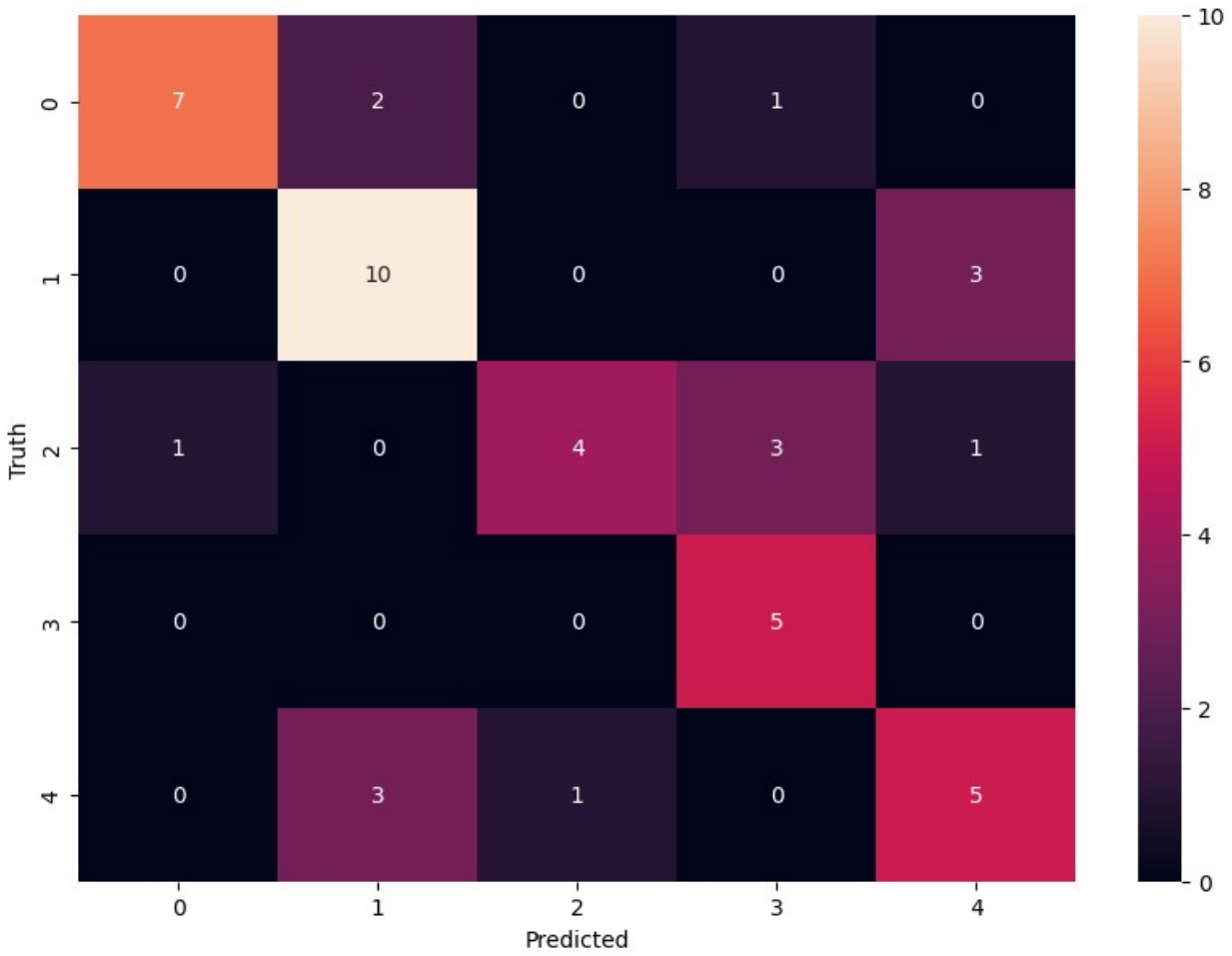
```
best_clf = best_estimators['logistic_regression']
```

```
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, best_clf.predict(X_test))
cm
```

```
array([[ 7,  2,  0,  1,  0],
       [ 0, 10,  0,  0,  3],
       [ 1,  0,  4,  3,  1],
       [ 0,  0,  0,  5,  0],
       [ 0,  3,  1,  0,  5]], dtype=int64)
```

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

```
Text(95.72222222222221, 0.5, 'Truth')
```




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

```
Defaulting to user installation because normal site-packages is not
writeable
Requirement already satisfied: joblib in c:\programdata\anaconda3\lib\
site-packages (1.2.0)
```

```
['saved_model.pkl']
```

```
import json
with open("class_dictionary.json","w") as f:
    f.write(json.dumps(class_dict))
```