Face recognition in python using Opencv 2.4.10

Requirements:

- Opencv 2.4.10
- Python 2.7.10
- Numpy 1.11.0

First thing is we need to make our own datasets, Here we're not going to use any predefined datasets. So to create a datasets we'll use our webcam to capture Image and to detect faces and save these face images to a folder. we'll use opencv haar feature-based cascade classifiers to detect faces. First we need to load the required XML classifiers. Then load our input image (or video) in grayscale mode. Now we find the faces in the image. If faces are found, it returns the positions of detected faces as Rect(x,y,w,h).

```
#creating database
import cv2, sys, numpy, os
haar_file = 'haarcascade_frontalface_default.xml'
datasets = 'datasets'  #All the faces data will be present this
folder
sub_data = 'aquib'     #These are sub data sets of folder, for my
                        faces I've used my name use always
                        different name for different people.

path = os.path.join(datasets, sub_data)
if not os.path.isdir(path):
    os.mkdir(path)
(width, height) = (130, 100) # defining the size of images


face_cascade = cv2.CascadeClassifier(haar_file)
webcam = cv2.VideoCapture(0) #'0' is use for my webcam, if you've
                              any other camera attached use '1'


# The program loops until it has 30 images of the face.
count = 1
while count < 31:
    (_, im) = webcam.read()
    gray = cv2.cvtColor(im, cv2.COLOR_BGR2GRAY)
    faces = face_cascade.detectMultiScale(gray, 1.3, 5)
    for (x,y,w,h) in faces:
        cv2.rectangle(im,(x,y),(x+w,y+h),(255,0,0),2)
        face = gray[y:y + h, x:x + w]
        face_resize = cv2.resize(face, (width, height))
        cv2.imwrite('%s/%s.png' % (path,count), face_resize)
    count += 1

    cv2.imshow('OpenCV', im)
    key = cv2.waitKey(10)
    if key == 27:
        break
```

Save this script as *data.py* and run through the terminal. If everything goes well, you'll see a folder 'aquib' in the database folder. Atleast make 10 to 12 different database.

Now the time is for training and testing run the code below

```
import cv2, sys, numpy, os
haar_file = 'haarcascade_frontalface_default.xml'
datasets = 'datasets'
# Part 1: Create fisherRecognizer
print('Training...')
# Create a list of images and a list of corresponding names
(images, labels, names, id) = ([], [], {}, 0)
for (subdirs, dirs, files) in os.walk(datasets):
    for subdir in dirs:
        names[id] = subdir
        subjectpath = os.path.join(datasets, subdir)
        for filename in os.listdir(subjectpath):
            path = subjectpath + '/' + filename
            label = id
            images.append(cv2.imread(path, 0))
            labels.append(int(label))
        id += 1
(width, height) = (130, 100)

# Create a Numpy array from the two lists above
(images, labels) = [numpy.array(lis) for lis in [images, labels]]

# OpenCV trains a model from the images
# NOTE FOR OpenCV2: remove '.face'
model = cv2.createFisherFaceRecognizer()
model.train(images, labels)

# Part 2: Use fisherRecognizer on camera stream
face_cascade = cv2.CascadeClassifier(haar_file)
webcam = cv2.VideoCapture(0)
while True:
    (_, im) = webcam.read()
    gray = cv2.cvtColor(im, cv2.COLOR_BGR2GRAY)
    faces = face_cascade.detectMultiScale(gray, 1.3, 5)
    for (x,y,w,h) in faces:
        cv2.rectangle(im,(x,y),(x+w,y+h),(255,0,0),2)
        face = gray[y:y + h, x:x + w]
        face_resize = cv2.resize(face, (width, height))
        #Try to recognize the face
        prediction = model.predict(face_resize)
        cv2.rectangle(im, (x, y), (x + w, y + h), (0, 255, 0), 3)

        if prediction[1]<500:

            cv2.putText(im,'%s - %.0f' %
(names[prediction[0]],prediction[1]),(x-10, y-10),
cv2.FONT_HERSHEY_PLAIN,1,(0, 255, 0))
     else:
       cv2.putText(im,'not recognized',(x-10, y-10),
cv2.FONT_HERSHEY_PLAIN,1,(0, 255, 0))

    cv2.imshow('OpenCV', im)
    key = cv2.waitKey(10)
    if key == 27:
        break
```