Trainer model

Importing libraries:

In [8]:

```
import cv2
import os
import numpy as np
import pickle
```

Finding path of images:

In [9]:

```
BASE_DIR = os.path.dirname(os.path.abspath('__file__'))
image_dir = os.path.join(BASE_DIR,'images')
```

Face recognizer:

In [10]:

```
face_cascade = cv2.CascadeClassifier('cascades/data/haarcascade_frontalface_default.xml')
recognizer = cv2.face.LBPHFaceRecognizer_create()
```

Storing ID and face recognized images:

In [11]:

```
current_id = 0
label_ids = {}
x_train = []
y_labels = []
```

Recognizing faces:

In [12]:

```
for root,dirs,files in os.walk(image_dir):
    for file in files:
        if file.endswith('png') or file.endswith('jpg'):
            path = os.path.join(root,file)
            label = os.path.basename(root).replace(' ','-').lower()
            if not label in label_ids:
                label_ids[label] = current_id
                current_id += 1
            id_ = label_ids[label]

            img = cv2.imread(path)
            imgGray = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
            size = (500,500)
            final_image = cv2.resize(imgGray,size)

            faces = face_cascade.detectMultiScale(final_image,
                                                  scaleFactor = 1.5,
                                                  minNeighbors=5)
            for (x,y,w,h) in faces:
                dimensions = final_image[y:y+h,x:x+w]
                x_train.append(dimensions)
                y_labels.append(id_)
```

Saving ID as pickle file:

In [13]:

```
with open('labels.pickle','wb') as f:
    pickle.dump(label_ids,f)
```

Training the model and save it as .yml:

In [14]:

```
recognizer.train(x_train, np.array(y_labels))
recognizer.save('trained_faces.yml')
```

Facial Recognition:

Importing libraries:

In [1]:

```
import cv2
import pickle
```

Importing the trained data:

In [2]:

```
face_cascade = cv2.CascadeClassifier('cascades/data/haarcascade_frontalface_default.xml')
recognizer = cv2.face.LBPHFaceRecognizer_create()
recognizer.read('trained_faces.yml')
```

Reading the pickle file for ID:

In [4]:

```
labels = {}
with open('labels.pickle','rb') as f:
    lables1 = pickle.load(f)
    for v,k in lables1.items():
        labels[k] = v
```

Live Video Output:

In [5]:

```
cap = cv2.VideoCapture(0)
while True:
    success,frame = cap.read()
    gray = cv2.cvtColor(frame,cv2.COLOR_BGR2GRAY)
    faces = face_cascade.detectMultiScale(gray, scaleFactor=1.5,minNeighbors=5)
    for x,y,w,h in faces:
        dim_gray = gray[y:y+h,x:x+h]
        dim_color = frame[y:y+h,x:x+h]

        id_,confidence = recognizer.predict(dim_gray)
        if confidence>85:
            font = cv2.FONT_HERSHEY_SIMPLEX
            name = labels[id_]
            color = (255,255,255)
            stroke = 2
            cv2.putText(frame,name,(x,y),font,1,color,stroke)

        color = (0,255,0)
        stroke = 2
        cv2.rectangle(frame,(x,y),(x+w,y+h),color,stroke)

    cv2.imshow('Output',frame)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

cap.release()
cv2.destroyAllWindows()
```