我们先导入opencv库、操作系统接口模块、图像处理标准库和NumPy函数库

1. **import** cv2
2. **import** os
3. **from** PIL **import** Image
4. **import** numpy as np

接着我们定义一个getImageAndLabels函数来获取图像和姓名信息

1. **def** getImageAndLabels(path):

我们在函数最开始创建两个列表来分别储存人脸数据和姓名数据

1. facesSamples=[]
2. ids=[]

接着我们定义一个imagePaths变量来储存图片信息

1. imagePaths=[os.path.join(path,f)**for** f **in** os.listdir(path)]

然后加载我们的分类器（注意修改地址）

1. face\_detector = cv2.CascadeClassifier("C:/Users/33086/Downloads/opencv/sources/data/haarcascades\_cuda/haarcascade\_frontalface\_alt2.xml")

我们用for循环遍历列表中的图片

1. **for** imagePath **in** imagePaths:

以灰度图像方式打开

1. PIL\_img=Image.open(imagePath).convert('L')

将图像转化为数组

1. img\_numpy=np.array(PIL\_img,'uint8')

提取人脸特征

1. faces=face\_detector.detectMultiScale(img\_numpy)

提取文件名的第一部分赋值给id

1. id =int(os.path.split(imagePath)[1].split('.')[0])

我们将数据添加到两个列表中

1. **for** x,y,w,h **in** faces:
2. ids.append(id)
3. facesSamples.append(img\_numpy[y:y+h,x:x+w])

在函数的最后我们return这个函数

1. **return** facesSamples,ids

完整的函数为：

1. **def** getImageAndLabels(path):
2. facesSamples=[]#储存人脸数据（这是一个二维数组）
3. ids=[]#储存姓名数据
4. imagePaths=[os.path.join(path,f)**for** f **in** os.listdir(path)]  #储存图片信息
5. face\_detector = cv2.CascadeClassifier("C:/Users/33086/Downloads/opencv/sources/data/haarcascades\_cuda/haarcascade\_frontalface\_alt2.xml")
6. **for** imagePath **in** imagePaths:#遍历列表中的图片
7. PIL\_img=Image.open(imagePath).convert('L')#以灰度图像的方式打开
8. img\_numpy=np.array(PIL\_img,'uint8')#将图像转化为数组
9. faces=face\_detector.detectMultiScale(img\_numpy)#使用分类器获取人脸特征
10. id =int(os.path.split(imagePath)[1].split('.')[0])#提取图片.之前的东西作为id
11. **for** x,y,w,h **in** faces:
12. ids.append(id)#将id添加到ids这个列表内
13. facesSamples.append(img\_numpy[y:y+h,x:x+w])#把x,y,w,h添加到facesSamples这个列表
14. **return** facesSamples,ids  #返回

我们定义一个变量 path，指向一个包含人脸图像的文件夹（注意修改地址）

1. path="D:/positive"

然后获取图像数组和id标签数组

1. faces,ids=getImageAndLabels(path)#获取到了图像数组和id标签数组和姓名

接着我们加载识别器并训练

1. recognizer=cv2.face.LBPHFaceRecognizer\_create()
2. recognizer.train(faces,np.array(ids))

最后我们保存训练好的识别器到指定文件夹内

1. recognizer.write("D:/trainer/trainer.yml")

完整的代码如下：

1. import cv2
2. import os
3. from PIL import Image
4. import numpy as np
5. def getImageAndLabels(path):
6. facesSamples=[]#储存人脸数据（这是一个二维数组）
7. ids=[]#储存姓名数据
8. imagePaths=[os.path.join(path,f)**for** f in os.listdir(path)]  #储存图片信息
9. face\_detector = cv2.CascadeClassifier("C:/Users/33086/Downloads/opencv/sources/data/haarcascades\_cuda/haarcascade\_frontalface\_alt2.xml")
10. **for** imagePath in imagePaths:#遍历列表中的图片
11. PIL\_img=Image.open(imagePath).convert('L')#以灰度图像的方式打开
12. img\_numpy=np.array(PIL\_img,'uint8')#将图像转化为数组
13. faces=face\_detector.detectMultiScale(img\_numpy)#使用分类器获取人脸特征
14. id =**int**(os.path.split(imagePath)[1].split('.')[0])#提取图片.之前的东西作为id
15. **for** x,y,w,h in faces:
16. ids.append(id)#将id添加到ids这个列表内
17. facesSamples.append(img\_numpy[y:y+h,x:x+w])#把x,y,w,h添加到facesSamples这个列表
18. **return** facesSamples,ids  #返回
19. **if** \_\_name\_\_ == '\_\_main\_\_':
20. path="D:/positive"
21. faces,ids=getImageAndLabels(path)#获取图像数组和id标签数组和姓名
22. recognizer=cv2.face.LBPHFaceRecognizer\_create()#加载识别器
23. recognizer.train(faces,np.array(ids))#训练
24. recognizer.write("D:/trainer/trainer.yml")#保存文件