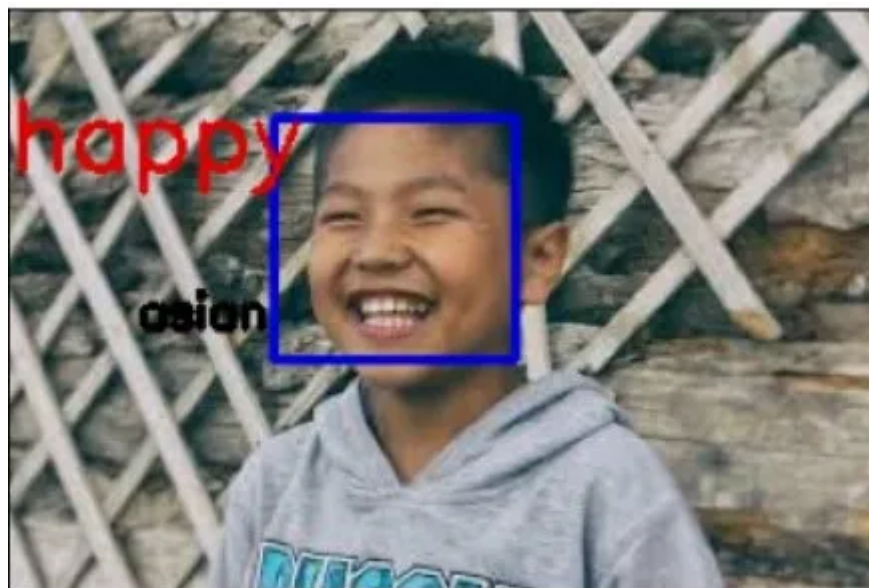


使用Python+OpenCV+Deepface實現人臉、精神和神經檢測

原創 磐慰慰 深度學習與計算機視覺 昨天

我們使用 OpenCV、deepface 和 haarcascade_frontal_default.xml 文件來檢測圖像中的人臉、面部表情和人種。



來自<https://unsplash.com/photos/vtwjyEelY08>下載此圖片

在我們開始之前，你已經知道了python基礎並且已經安裝了python

需要的其他安裝包是：

Opencv — 使用安裝它

deepface — 使用安裝它

matplotlib——使用安裝它

從 https://raw.githubusercontent.com/opencv/opencv/master/data/haarcascades/haarcascade_frontalface_default.xml 下載
haarcascade_frontalface_default.xml 將其保存在與你的項目相同的文件夾中。

什麼是 haarcascade_frontalface_default.xml 文件？

它是一個經過預習以拍攝並提供一些用於拍攝場景、個人或任何形狀的點的文件。

現在讓我們進入編碼部分（在 jupyter notebook 中損壞）。

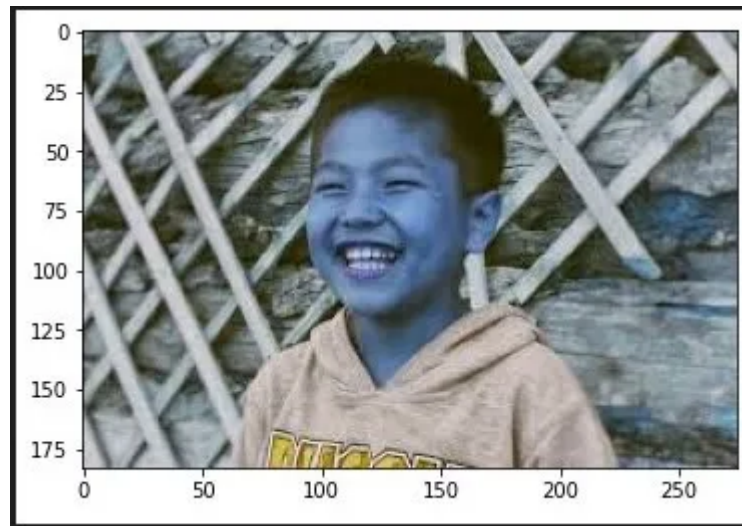
首先，讓我們需要導入的庫並加載happy_boy.jpg（我們之前下載的圖片），並應該圖片。

```
#importing cv2 and matplotlib
import cv2
import matplotlib.pyplot as plt
```

```
#loading image
img = cv2.imread("happy_boy.jpg") #loading image

#showing image using plt
plt.imshow(img)
```

输出：



每次我们显示图像时，它都会在 BGR 中，在显示之前我们需要将其转换为彩色照片。为此，我们使用 cv2 中的 cvtColor() 函数。

```
color_img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
plt.imshow(color_img)
```

输出：





使用 Deepface 预测图片中人/孩子的情绪和种族

导入 DeepFace 并分析我们的图片。

它将分析图像并将检测到的情绪和种族保存到“prediction”变量中。

```
#importing deepface library and DeepFace
from deepface import DeepFace

#this analyses the given image and gives values
#when we use this for 1st time, it may give many errors and some google drive links to download some '.h5' and zip fi
prediction = DeepFace.analyze(color_img)
```

让我们看看它预测了什么

```
#lets check what all the things DeepFace.analyze() function has analyzed
```

```
prediction
```

输出:

```
{'region': {'x': 94, 'y': 34, 'w': 63, 'h': 83},  
'emotion': {'angry': 9.303427472850956e-09,  
'disgust': 2.193804491366913e-12,  
'fear': 5.261768451503279e-09,  
'happy': 99.99998807907104,  
'sad': 5.065415464855505e-06,  
'surprise': 7.973710125334932e-11,  
'neutral': 1.1359878016037328e-05},  
'dominant_emotion': 'happy',  
'age': 27,  
'gender': 'Man',  
'race': {'asian': 73.75681243877484,  
'indian': 5.643263795570362,  
'black': 1.5398613163746189,  
'white': 2.0414629305281,  
'middle eastern': 0.3165866669840614,  
'latino hispanic': 16.702010733009026},  
'dominant_race': 'asian'}
```

在输出中，预测值是dominant_emotion 和dominant_race。剩下的是所有用于预测情绪和种族的点。

我们可以像访问任何其他字典数据类型一样访问dominant_emotion 和dominant_race 的键值。

```
▶ ▶≡ M4  
  
#lets see what emotion does the model predicted  
  
prediction['dominant_emotion']  
  
'happy'
```

所以 DeepFace.analyze() 预测了那个孩子的情绪——快乐和种族——亚洲人。

检测人脸

使用 OpenCV 检测人脸的代码非常短。

我们只需要haarcascade_frontalface_default.xml文件和 OpenCV 库来检测图像中的人脸。这是代码。

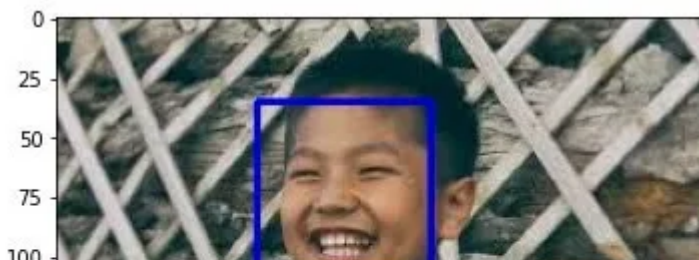
```
#loading our xml file into faceCascade using cv2.CascadeClassifier
faceCascade = cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade_frontalface_default.xml')

#detecting face in color_image and getting 4 points(x,y,u,v) around face from the image, and assigning those values to
faces = faceCascade.detectMultiScale(color_img, 1.1, 4)

#using that 4 points to draw a rectangle around face in the image
for (x, y, u, v) in faces:
    cv2.rectangle(color_img, (x,y), (x+u, y+v), (0, 0, 225), 2)

plt.imshow(color_img)
```

输出：





汇总

如你所见，我们检测到图像中孩子的情绪、种族和面孔，现在是将所有内容整合在一起的最后一部分。这意味着现在我们要在图像上展示情感和种族。

```
#choose font for text
font = cv2.FONT_HERSHEY_SIMPLEX

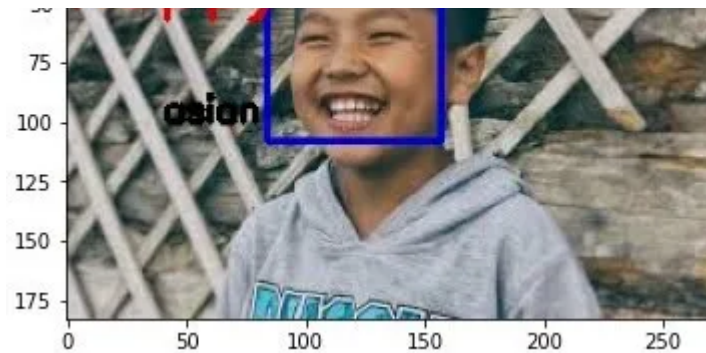
#for showing emotion on image
cv2.putText(color_img, prediction['dominant_emotion'], (0, 50), font, 1,
(225,0,0), 2, cv2.LINE_4)

#for showing race on image
cv2.putText(color_img, prediction['dominant_race'], (40, 100), font, 0.5,
(0,0,0), 2, cv2.LINE_4)

#finally displaying image
plt.imshow(color_img)
```

输出：





如果你不知道/理解我們在 `cv.putText()` 函數中寫的所有東西，這裡的語法如下

```
cv.putText(img, '要放置文字的*', '*要放置文字的位置', 字體, 文字大小, 顏色, 文字粗細, 文字文字)
```

這就是我們檢測人臉、人類精神和精神的方式。

現在去嘗試其他圖像。

我們也可以將它與網絡攝像頭或後置攝像頭一起使用，但速度非常慢。

☆ 結束 ☆

如果看到這裡，說明你喜歡這篇文章，請評論、點贊。微信搜索「uncle_pn」，歡迎添加小編微信「mthler」，每日朋友圈更新一篇文章博文。

↓掃描二維碼添加小編↓



贝加尔

阿尔巴尼亚



扫一扫上面的二维码图案，加我微信

喜歡這個內容的人還喜歡

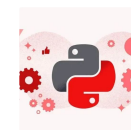
Python匯總篇，200+Python標準庫介紹（超全）

簡說Python



Python優化提速的8個小技巧

簡說Python



用Python執行SQL、Excel常見任務？10個方法全搞定！

簡說Python



