1026 討論
>[time=Mon, Oct 26, 2020 3:35 PM]

關鍵字 ：
face re-enactment
face2face

* 如果要做換臉的話 GAN （生成對抗網路） 考慮中
參考資料
https://github.com/wywu/ReenactGAN
https://www.aili.com.tw/message2_detail/54.htm

https://github.com/datitran/face2face-demo
https://www.books.com.tw/products/0010868212?sloc=main

1208
:::warning
pictPath 路徑要跟著你的電腦去改
:::
```
# -*- coding: utf-8 -*-
"""
Created on Tue Dec  8 22:27:17 2020

@author: haha3206
"""


import cv2

def FaceRecognition(img):
    faces = face_cascade.detectMultiScale(img)
    cv2.rectangle(img, (img.shape[1]-120, img.shape[0]-20), (img.shape[1], 
img.shape[0]), (0,255,255),-1)
    cv2.putText(img,"Find "+str(len(faces))+" face",(img.shape[1]-110,img.shape[0]-
5),cv2.FONT_HERSHEY_COMPLEX,0.5,
                (255,0,0),1)
```

```
        print(faces)

        for x,y,w,h in faces:
            cv2.rectangle(img ,(x,y),(x+w,y+h),(255,0,0),2)
            cv2.imshow('fr',img)
            '''
        if faces!=():

cv2.rectangle(img ,(faces[0][0],faces[0][1]),(faces[0][0]+faces[0][2],faces[0][1]+faces[
0][3]),(255,0,0),2)
            cv2.imshow('fr',img)
            '''
pictPath = r'C:\Users\book2\anaconda3\Lib\site-
packages\cv2\data\haarcascade_frontalface_default.xml'
face_cascade = cv2.CascadeClassifier(pictPath)
cap = cv2.VideoCapture(0)
while cap.isOpened():
        ret,img = cap.read()

        if ret:
            #cv2.imshow('aa',img)
            FaceRecognition(img)
            if cv2.waitKey(1) & 0xFF == 32:
                break

        else:
            break
cap.release()
cv2.destroyAllWindows()
```
:::success
dlib 結果較準確
:::
```
import dlib
import cv2

#選擇第一隻攝影機
```

```
cap = cv2.VideoCapture( 0)
#調整預設影像大小，預設值很大，很吃效能
cap.set(cv2. CAP_PROP_FRAME_WIDTH, 650)
cap.set(cv2. CAP_PROP_FRAME_HEIGHT, 500)

#取得預設的臉部偵測器
detector = dlib.get_frontal_face_detector()
#根據 shape_predictor 方法載入 68 個特徵點模型，此方法為人臉表情識別的偵測器
predictor = dlib.shape_predictor( 'shape_predictor_68_face_landmarks.dat')
  #當攝影機打開時，對每個 frame 進行偵測
while(cap.isOpened()):
    #讀出 frame 資訊
    ret, frame = cap.read()

    #偵測人臉
    face_rects, scores, idx = detector.run(frame, 0)

    #取出偵測的結果
    for i, d in enumerate(face_rects):
      x1 = d.left()
      y1 = d.top()
      x2 = d.right()
      y2 = d.bottom()
      text = " %2.2f ( %d )" % (scores[i], idx[i])

      #繪製出偵測人臉的矩形範圍
      cv2.rectangle(frame, (x1, y1), (x2, y2), ( 0, 255, 0), 4, cv2. LINE_AA)


      #標上人臉偵測分數與人臉方向子偵測器編號
      cv2.putText(frame, text, (x1, y1), cv2. FONT_HERSHEY_DUPLEX,
      0.7, ( 255, 255, 255), 1, cv2. LINE_AA)

      #給 68 特徵點辨識取得一個轉換顏色的 frame
      landmarks_frame = cv2.cvtColor(frame, cv2. COLOR_BGR2RGB)

      #找出特徵點位置
```

```
            shape = predictor(landmarks_frame, d)

            #繪製 68 個特徵點
            for i in range( 68):
                cv2.circle(frame,(shape.part(i).x,shape.part(i).y), 3,( 0, 0, 255), 2)
                #cv2.putText(frame, str(i),(shape.part(i).x,shape.part(i).y),cv2.
FONT_HERSHEY_COMPLEX, 0.5,( 255, 0, 0), 1)
        #輸出到畫面
        cv2.imshow( "Face Detection", frame)

        #如果按下 ESC 键，就退出
        if cv2.waitKey( 10) == 32:
            break
#釋放記憶體
cap.release()
#關閉所有視窗
cv2.destroyAllWindows()
```

https://github.com/AliaksandrSiarohin/first-order-model