



周围的人都比你厉害，你才会慢慢变强

博客园 首页 新随笔 联系 订阅 管理

随笔 - 495 文章 - 0 评论 - 29

公告

昵称： 山上有风景
园龄： 1年11个月
粉丝： 170
关注： 19
+加关注

< 2019年12月 >

日	一	二	三	四	五	六
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	1	2	3	4
5	6	7	8	9	10	11

搜索

找找看

常用链接

我的随笔
我的评论
我的参与
最新评论
我的标签

我的标签

STL(18)
SDN(15)
ThinkPHP3.2(1)

OpenCV---人脸检测

目录

- 一：相关依赖文件下载
- 二：实现步骤（图片检测）
 - （一）读取图片
 - （二）灰度转换
 - （三）获取人脸识别训练数据
 - （四）探测人脸，获取相关数据
 - （五）根据相关数据在原图像上画出人脸位置
 - （六）全部代码
- 三：实现视频检测人脸

步骤相同，我们只需要将视频中每一帧图像进行处理，调用上面的图像人脸检测即可

- 相关知识补充：
- （一）CascadeClassifier级联分类器
 - 基本原理
 - detectMultiScale方法

一：相关依赖文件下载

<https://github.com/opencv/opencv>

0

0

关注 | 顶部 | 评论

积分与排名

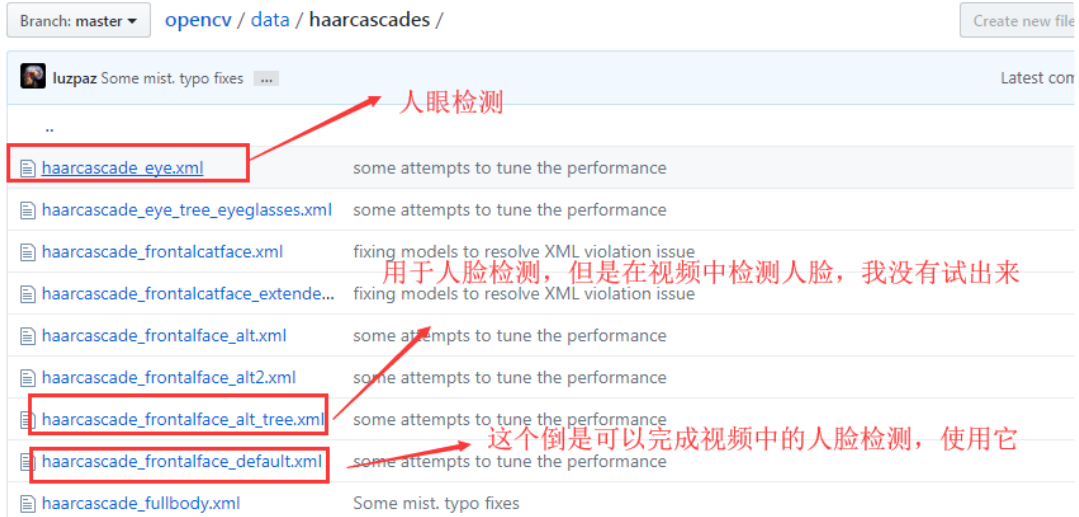
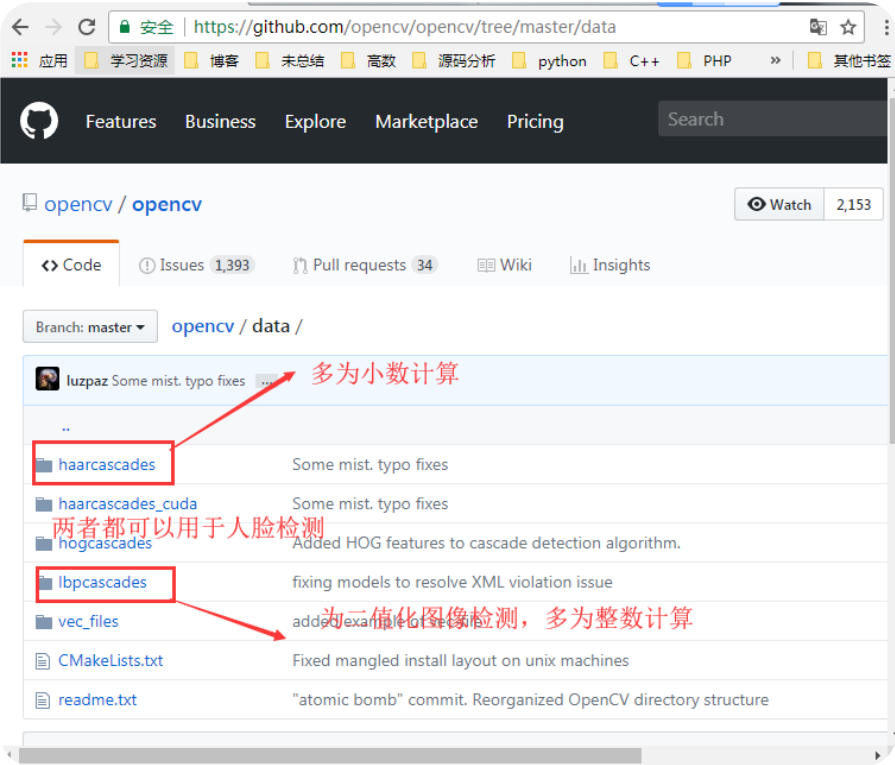
积分 - 211835
排名 - 1998

随笔分类

- C/C++(74)
- Html(2)
- Java(33)
- Javascript(19)
- OpenCV(29)
- PHP(2)
- Python(155)
- STL泛型编程(18)
- 单片机笔记（复习用）(3)
- 计算机网络(32)
- 其他知识(14)
- 设计模式(27)
- 数据结构(57)
- 数据库(8)
- 算法习题(43)
- 算法训练营
- 随笔所想(4)
- 图形界面编程
- 正则表达式(2)
- 转载推文(4)

随笔档案

- 2019年11月(5)
- 2019年10月(32)
- 2019年9月(21)
- 2019年7月(10)
- 2019年5月(8)
- 2019年4月(25)
- 2019年3月(8)
- 2019年2月(1)
- 2019年1月(12)
- 2018年12月(19)



二：实现步骤（图片检测）

（一）读取图片

```
image= cv.imread("./d.png") #读取图片
```

（二）灰度转换

```
gray = cv.cvtColor(image,cv.COLOR_BGR2GRAY) #在灰度图中检测人脸
```

（三）获取人脸识别训练数据

0 0

关注 | 顶部 | 评论

2018年9月(5)
2018年8月(95)
2018年7月(78)
2018年6月(26)
2018年5月(17)
2018年4月(22)
2018年3月(111)

最新评论

1. Re:python---
websocket的使用
网上的方法都不行,
换成gbk会报如下的错:
IndexError: index out
of range

--缘分天空0320

2. Re:python---
websocket的使用
@ 缘分天空0320太久没
用, 忘了。应该是字符编
码问题吧。这类问题网上
应该可以很容易找到方法
解决。你看看Python的
默认编码和代码是不是一
致。一般就是gbk和utf8
之间出错...

--山上有风景

3. Re:python---
websocket的使用
您好, 用了您的代码, 报
如下错误, 麻烦问下如何
解决呢? Traceback
(most recent call
last): File
"server3.py", line
101, in <module...

--缘分天空0320

4. Re:数据结构 (六) 查
找---线性索引查找
请问最后倒序排序的那个
代码怎么实现的? 可以发
一下吗?

```
face_detector =  
cv.CascadeClassifier("./haarcascade_frontalface_alt_tree.xml") #级联检测  
器获取文件
```

这个xml文件, 就是opencv在GitHub上共享出来的具有普适的训练好的数据。我们可以直接的拿来使用

(四) 探测人脸, 获取相关数据

```
faces = face_detector.detectMultiScale(gray, 1.1, 2)
```

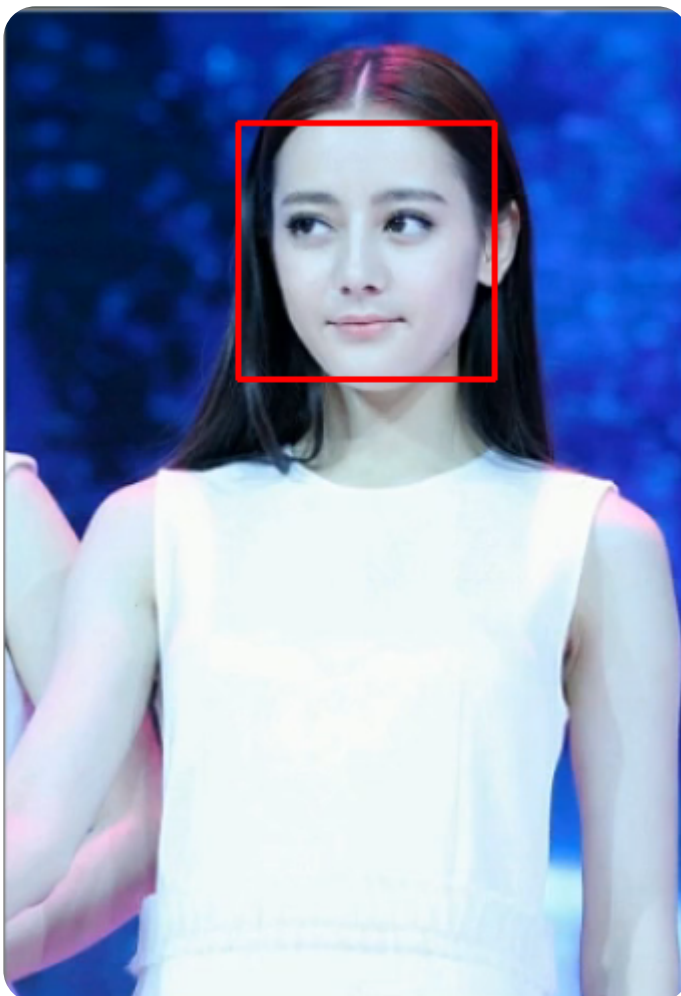
#第一个参数是灰度图像

#第二个参数是尺度变换, 就是向上或者向下每次是原来的多少倍, 这里是1.02倍

#第三个参数是人脸检测次数, 设置越高, 误检率越低, 但是对于迷糊图片, 我们设置越高, 越不易检测出来, 要适当降低

(五) 根据相关数据在原图像上画出人脸位置

```
for x,y,w,h in faces:  
    cv.rectangle(image, (x,y), (x+w,y+h), (0,0,255), 2)  
cv.imshow("face_detection", image)
```



0

0

关注 | 顶部 | 评论

--Viki-

5. Re:SDN实验---Ryu
的源码分析
@ 山上有风景谢谢! ...
--iRoy_33

阅读排行榜

- 1. python---websocket的使用(17253)
- 2. OpenCV---图像二值化(12513)
- 3. OpenCV---模板匹配matchTemplate(11534)
- 4. OpenCV---直线检测(8208)
- 5. python---基础知识回顾 (九) 图形用户界面----WxPython(7986)

评论排行榜

- 1. python---基础知识回顾 (九) 图形用户界面----Tkinter(4)
- 2. python---websocket的使用(3)
- 3. SDN实验---Ryu的源码分析(3)
- 4. 数据结构 (三) 串---KMP模式匹配算法之获取next数组(2)
- 5. 数据结构 (四) 树---树的存储结构(1)

推荐排行榜

- 1. 数据结构 (七) 排序---堆排序(11)
- 2. python---aiohttp的使用(6)
- 3. python---websocket的使用(4)

(六) 全部代码



```
import cv2 as cv
import numpy as np

def face_detect_demo(image):
    gray = cv.cvtColor(image,cv.COLOR_BGR2GRAY) #在灰度图像基础上实现的
    face_detector =
cv.CascadeClassifier("./haarcascade_frontalface_alt_tree.xml") #级联检测
器获取文件
    faces = face_detector.detectMultiScale(gray,1.1,2) #在多个尺度空间上
进行人脸检测
    #第一个参数是灰度图像
    #第二个参数是尺度变换，就是向上或者向下每次是原来的多少倍，这里是1.02倍
    #第三个参数是人脸检测次数，设置越高，误检率越低，但是对于迷糊图片，我们设置越高，
越不易检测出来，要试单降低
    for x,y,w,h in faces:
        cv.rectangle(image, (x,y), (x+w,y+h), (0,0,255),2)
    cv.imshow("face_detection",image)

src = cv.imread("./d.png") #读取图片
cv.namedWindow("input image",cv.WINDOW_AUTOSIZE) #创建GUI窗口,形式为自适应
cv.imshow("input image",src) #通过名字将图像和窗口联系

face_detect_demo(src)

cv.waitKey(0) #等待用户操作，里面等待参数是毫秒，我们填写0，代表是永远，等待用户
操作
cv.destroyAllWindows() #销毁所有窗口
```



三：实现视频检测人脸

步骤相同，我们只需要将视频中每一帧图像进行处理，调用上面的图像人脸检测即可



```
import cv2 as cv
import numpy as np
```

0

0

关注 | 顶部 | 评论

4. python---基础知识回顾 (九) 图形用户界面---WxPython(3)

5. Python图像处理库PIL中图像格式转换(3)

```
def face_detect_demo(image):
    gray = cv.cvtColor(image, cv.COLOR_BGR2GRAY) #在灰度图像基础上实现的
    face_detector =
cv.CascadeClassifier("./haarcascade_frontalface_default.xml") #级联检测器
获取文件

    faces = face_detector.detectMultiScale(gray, 1.1, 2) #在多个尺度空间上
进行人脸检测

    #第一个参数是灰度图像
    #第二个参数是尺度变换, 就是向上或者向下每次是原来的多少倍, 这里是1.02倍
    #第三个参数是人脸检测次数, 设置越高, 误检率越低, 但是对于迷糊图片, 我们设置越高,
越不易检测出来, 要试单降低

    for x, y, w, h in faces:
        cv.rectangle(image, (x, y), (x+w, y+h), (0, 0, 255), 2)
    cv.imshow("face_detection", image)

def video_face_detect():
    capture = cv.VideoCapture(0)
    while True:
        ret, frame = capture.read() #frame是每一帧图像, ret是返回值, 为0是表示
图像读取完毕

        frame = cv.flip(frame, 1)
        if ret == False:
            break

        face_detect_demo(frame)
        c = cv.waitKey(10)
        if c == 27:
            break

video_face_detect()

cv.waitKey(0) #等待用户操作, 里面等待参数是毫秒, 我们填写0, 代表是永远, 等待用户
操作
cv.destroyAllWindows() #销毁所有窗口
```



相关知识补充:

(一) CascadeClassifier级联分类器

是OpenCV中做人脸检测的时候的一个级联分类器
 有两种选择: 一是使用老版本的CvHaarClassifierCascade类。
 老版本的分类器只支持类Haar特征, 而新版本的分类器既可以使

0

0

[关注](#) | [顶部](#) | [评论](#)

基本原理

xml中存放的是训练后的特征池，特征size大小根据训练时的参数而定，检测的时候可以简单理解为就是将每个固定size特征（检测窗口）与输入图像的同样大小区域比较，如果匹配那么就记录这个矩形区域的位置，然后滑动窗口，检测图像的另一个区域，重复操作。

由于输入的图像中特征大小不定，比如在输入图像中眼睛是50x50的区域，而训练时的是25x25，那么只有当输入图像缩小到一半的时候，才能匹配上，所以这里还有一个逐步缩小图像，也就是制作图像金字塔的流程。

```
cascPath = "./haarcascade_frontalface_default.xml"
cv2.CascadeClassifier(cascPath)    #构造方法加载入xml特征文件
```

detectMultiScale方法

```
faces = faceCascade.detectMultiScale(gray, 1.1, 2, 30)
```

```
def detectMultiScale(self, image, scaleFactor=None, minNeighbors=None,
flags=None, minSize=None, maxSize=None): # real signature unknown;
restored from __doc__
```



#第一个参数image：是灰度图像

#第二个参数scaleFactor：是尺度变换，就是向上或者向下每次是原来的多少倍，这里是1.02倍

#第三个参数minNeighbors：是人脸检测次数，设置越高，误检率越低，但是对于迷糊图片，我们设置越高，越不易检测出来，要适当降低

#第四个参数flags：

CASCADE_DO_CANNY_PRUNING=1，利用canny边缘检测来排除一些边缘很少或者很多的图像区域

CASCADE_SCALE_IMAGE=2，正常比例检测 CASCADE_FIND_BIGGEST_OBJECT=4，只检测最大的物体

CASCADE_DO_ROUGH_SEARCH=8 初略的检测

#最后两个参数用来限制得到的目标区域的范围



注意：

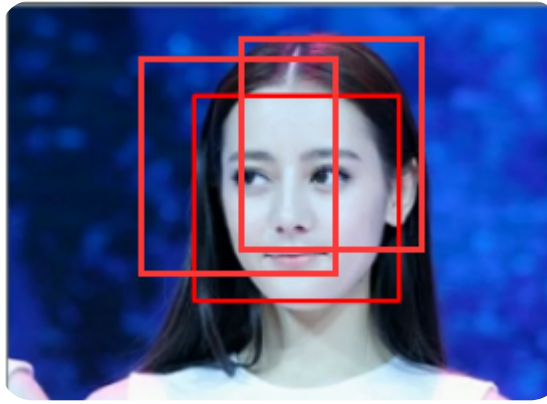
flags对于新的分类器没有用

设置minNeighbors：是在人脸附近进行指定次数的检测，获取最准确的范围，设置越高，误检率越低，但是对于迷糊图片，我们设置越高，越不易检测出来，要适当降低

0

0

关注 | 顶部 | 评论



作者：山上有风景

欢迎任何形式的转载，但请务必注明出处。

限于本人水平，如果文章和代码有表述不当之处，还请不吝赐教。

分类： OpenCV

好文要顶

关注我

收藏该文



山上有风景

关注 - 19

粉丝 - 170

+加关注

« 上一篇： OpenCV---分水岭算法

» 下一篇： OpenCV---数字验证码识别

posted @ 2018-07-10 08:29 山上有风景 阅读(1095) 评论(0) 编辑 收藏

[刷新评论](#) [刷新页面](#) [返回顶部](#)

注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)， [访问](#) 网站首页。

Copyright © 2019 山上有风景

Powered by .NET Core 3.1.0 on Linux

0

0

[关注](#) | [顶部](#) | [评论](#)