





```
1 activate py35
```

安装tensorflow(这是CPU版，GPU版去看上面那个教程)

```
1 conda install tensorflow
```

# OpenCV3

在这里原文建议用的是OpenCV3.0，因为

OpenCV 3.1 might crash on OSX after a while, so that's why I had to switch to version 3.0. See [open issue](#) and [solution](#) here.

在OSX上会有bug，但是我们这用的是windows。。win居然还有这种日子

用conda安装opencv，打开Anaconda Prompt:

```
1 conda install --channel https://conda.anaconda.org/menpo opencv3
```

conda自动匹配的是opencv3.1

直接安装

如果下载速度感人，在Anaconda Prompt里把下载源设置为国内地址

```
1 conda config --add channels https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/free/
2 conda config --set show_channel_urls yes
```

如果下载速度还是感人，去Unofficial Windows Binaries for Python Extension Packages (<http://www.lfd.uci.edu/~gohlke/pythonlibs/>)下载对应的opencv版本whl文件。在这个网站上对应python3.5d的只有OpenCV3.3，经过亲测，对于这个例子来说没啥影响。

下载 opencv\_python-3.3.0-cp35-cp35m-win\_amd64.whl 完毕后，用Anaconda Prompt进入到whl文件所在的目录：

```
1 pip install opencv_python-3.3.0-cp35-cp35m-win_amd64.whl
```

## matplotlib

```
1 conda install matplotlib
```

## PIL

python3.5已经没有以前常用的PIL，转用pillow代替

```
1 conda install pillow
```



## ■ 在线课程

[illegible]

# Object detector 程序测试

下载github上的源码 ([https://github.com/datitran/object\\_detector\\_app](https://github.com/datitran/object_detector_app))  
用Anaconda Prompt进入到源码所在目录，运行 object\_detection\_app.py

```
1 python object_detection_app.py
```

正常来说就会启动识别程序了。程序会打开一个窗口，摄像头的图像会实时显示，相关目标会用框框出来。

这个原文给的代码demo识别类别有限，到现在为止我只测出来几种不多的类别。

## 设置网络摄像头

这里用**智能手机**自带的摄像头来充当网络摄像头（貌似发现了家中淘汰的智能手机发光发热的机会）

## 下载网络摄像头APP

我用的是我的华为m2, 在官方的应用市场里找到了最上方的那款“IP摄像头”



打开之后的界面如上方右图所示，有很多摄像头参数可以自己调节。调完参数后菜单拉倒最下方，点“开启服务器” 网络摄像头就能开始正常工作了

362

图像处理复习（三）：特征点的搜索匹配和筛选 ([http://blog.csdn.net/github\\_38705794/article/details/77434753](http://blog.csdn.net/github_38705794/article/details/77434753))

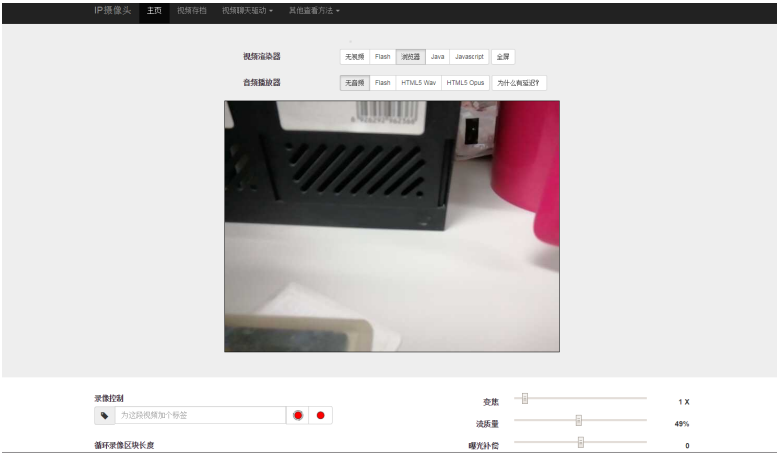
322

服务器，网络摄像头地址为192.168.1.100:8080

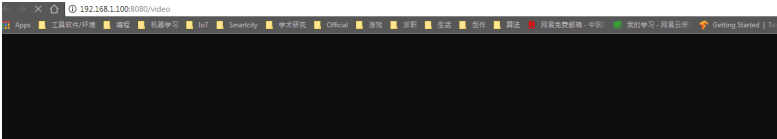
这个网络摄像头是针对**同一局域网**内的设备的，所以进行接下来的实验的电脑必须和网络摄像头，也就是智能手机处于同一局域网内（所以如果用智能手机当网络摄像头，就只能连无线网络了）

### 获取网络摄像头地址

点击“开启服务器”之后，在电脑的网页浏览器内输入摄像头app给你的局域网ip和端口（默认是8080），就可以进入网络摄像头的接待界面

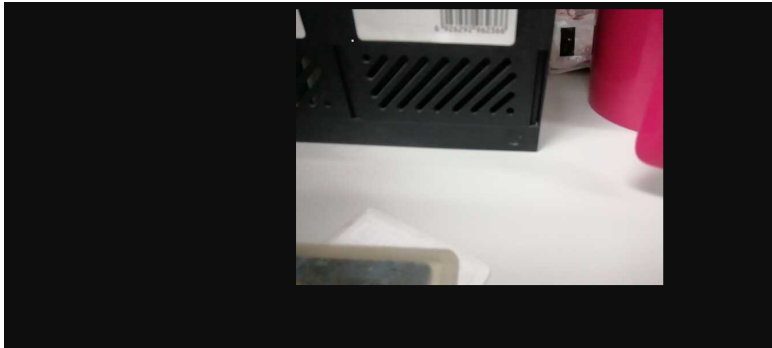


但是我们在用python程序调用摄像头时是不需要这个交互界面的，我们需要的是纯粹的视频：右键点击视频区域-在新选项卡中打开，最后获得了网络摄像头的地址 <http://192.168.1.100:8080/video>（我的地址，不同的环境可能有所不同）与图像





0



至此网络摄像头就配置完成了，这个图像可以被python程序直接调用

## python 调用网络摄像头

利用opencv可以直接调用并输出摄像头获得的图像：

```
1  # -*- coding: utf-8 -*-
2  import cv2
3  cv2.namedWindow("webcam test")
4  cam_url='http://192.168.1.100:8080/video'
5  vc=cv2.VideoCapture(cam_url)
6  if vc.isOpened():
7      rval, frame = vc.read()
8  else:
9      rval = False
10
11  while rval:
12      # frame=cv2.resize(frame,(100,100)) 调节输出图像的大小
13      cv2.imshow("webcam test", frame)
14      rval, frame = vc.read()
15      key = cv2.waitKey(20)
16      if key == 27: # exit on ESC
17          break
```



0



如果一切顺利，程序运行后系统会自动开启一个名叫“webcam test”的显示窗口，显示的内容正是你智能手机摄像头拍摄到的画面

到这里已经实现了“监控”的功能了，接下来要做的就是把监控和物体探测结合起来

## 监控 + 物体探测

这就需要改一下先前那个Github项目中的代码了，只需要改 object\_detection\_app.py 中的一些关于opencv语句即可

**object\_detection\_app.py**

```
1  ... ..
2  if __name__ == '__main__':
3      ... ..
4      input_q = Queue(maxsize=args.queue_size)
5      output_q = Queue(maxsize=args.queue_size)
6      pool = Pool(args.num_workers, worker, (input_q, output_q))
7
8      # webcam
9      cam_url='http://192.168.1.100:8080/video'
10     video_capture=cv2.VideoCapture(cam_url)
11     #video_capture = WebcamVideoStream(src=args.video_source,
12     #                                width=args.width,
13     #                                height=args.height).start()
14     fps = FPS().start()
15     t_start = time.time()
16     out = None
17     while True: # fps_numFrames < 120
18         if time.time()-t_start > 2: # 因为识别速度和视频帧率相差过大，为了使输出图像与摄像头输入保持同步，所以每两秒输
19             t_start = time.time()
20             _frame = video_capture.read()
21             frame=cv2.resize(frame,(int(args.width),int(args.height)))
22             input_q.put(frame)
23
24             t = time.time()
25             out = output_q.get()
26             cv2.imshow('Video', out)
27             fps.update()
28
29             print('[INFO] elapsed time: {:.2f}'.format(time.time() - t))
30     else:
```



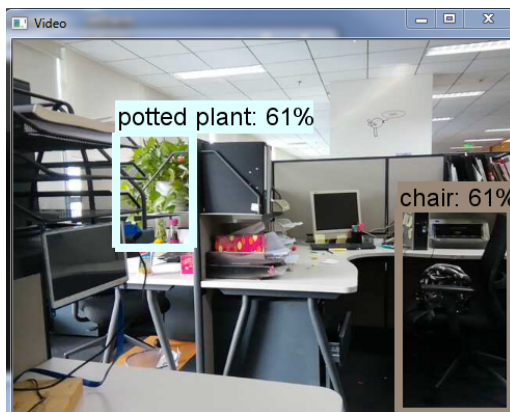
0

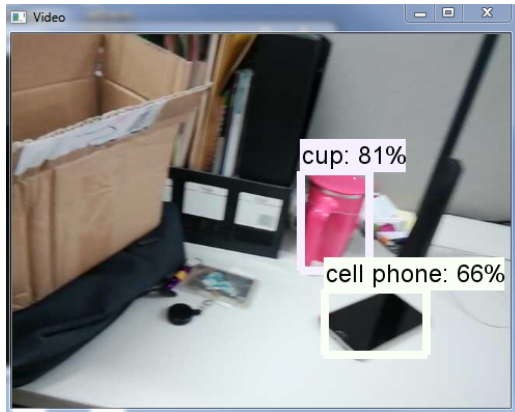


```
31     _frame = video_capture.read()
32     if out is not None:
33         cv2.imshow("Video", out)
34     else:
35         cv2.imshow("Video", frame)
36
37     if cv2.waitKey(1) & 0xFF == ord('q'):
38         break
39
40
41 ... ..
```

运行方法没变

**效果**





完工

可以带着小手机到处逛了

## 参考资料

关于anaconda3下安装opencv3 (<http://blog.csdn.net/nucyubg/article/details/52015426>)

opencv调用网络摄像头 (<http://robokite.blogspot.fr/2013/04/opencv-python-ip-webcam.html>)

OpenCV获取IP摄像头视频 (<http://www.cnblogs.com/taceywong/p/4858973.html>)

Building a Real-Time Object Recognition App with Tensorflow and OpenCV (<https://medium.com/towards-data-science/building-a-real-time-object-recognition-app-with-tensorflow-and-opencv-b7a2b4ebdc32>)

版权声明：本文为博主原创文章，未经博主允许不得转载。



发表你的评论

([http://my.csdn.net/weixin\\_35068028](http://my.csdn.net/weixin_35068028))

## 相关文章推荐

**论文学习笔记：Convolutional Neural Pyramid for Image Processing** (<http://blog.csdn.net...>)

论文学习笔记：Convolutional Neural Pyramid for Image Processing原文链接：Convolutional Neural Pyramid for Image...



github\_38705794 ([http://blog.csdn.net/github\\_38705794](http://blog.csdn.net/github_38705794))

2017年09月14日 02:08

111





win10+anaconda3+tensorflow-gpu一些报错的解决方法 (http://blog.csdn.net/github\_38705794)

其实这就是个踩坑记录作为一名苦逼研究生苦苦守在windows系统上，只因为当初买电脑的时候搞了一台无法装ubuntu的acer。之前用过一段时间的TensorFlow(https://www.te...

github\_38705794 (http://blog.csdn.net/github\_38705794) 2017年08月06日 13:38 2023



月薪40k的前端程序员都避开了哪些坑？

程序员薪水有高低，同样工作5年的程序员，有的人月薪30K、50K，有的人可能只有5K、8K。是什么因素导致了这种差异？

(http://www.baidu.com/cb.php?c=lgF\_pyfqHmknj0dP1f0lZ0qnfK9ujYzP1nYPH0k0Aw-5Hc3rHnYnHb0TAq15HfLPWRznjb0T1d9nH9-PjbsrHF-m17-rHRk0AwY5HDdnHc3nWRknjT0lgF\_5y9YIZ0lQzquZR8mLPbUB48ugfElAqspynElvNBnHqdlAdxTvqdThP-5yF\_UvTkn0KzujYk0AFV5H00TZcqn0KdpyfqHRLPjnvnfKEpyfqHc4rj6kP0KWpyfqP1cvrHnz0AqLUWYs0ZK45HcsP6KWThnqPHDvnH6)

OpenCV【6】---使用均值漂移（Mean Shift）算法实时跟踪视频流中的物体 (http://blog.csdn.net/FreeApe)

1 均值漂移算法简介 均值漂移算法是一种基于密度梯度上升的非参数方法，通过迭代运算找到目标位置，实现目标跟踪。所谓跟踪，就是通过已知的图像帧中的目标位置找到目标在下一帧中的位置。均值漂移算法显著的优...

FreeApe (http://blog.csdn.net/FreeApe) 2016年01月08日 20:08 2962

OpenCV中文网站例程——读视频文件和运动物体检测 (http://blog.csdn.net/c20081052/article/details/1390890)

来自于仕琪的讲稿《使用OpenCV进行图像处理》中的例程 程序如下： #include #include #include #include int main( int arg...

c20081052 (http://blog.csdn.net/c20081052) 2014年04月01日 20:10 890



VLC for Android 基于 Opencv 对 RTSP视频 实时人脸检测 (http://download.csdn.net/download/c20081052/9522292)

(http://download.csdn.net/download/c20081052/9522292) 2016年09月07日 14:59 22.9MB 下载



人人都能看懂的 AI 入门课

本课程将讲述人工智能的现状、应用场景和入门方法，并通过运用 TensorFlow，使得受众能清晰了解人工智能的运作方式。

(http://www.baidu.com/cb.php?c=lgF\_pyfqHmknjfrjc0lZ0qnfK9ujYzP1f4Pjn10Aw-5Hc4nj6vPjm0TAq15Hf4rjn1n1b0T1dWnyRsn1fYnWl-PyD3rj7-0AwY5HDdnHc3nWRknjT0lgF\_5y9YIZ0lQzqMpgwBUvqoQhP8QvIGIAPCmgfEmvq\_lyd8Q1R4uWl-n16kPWKWrrHnvnHRvnnNBuYD4PHqdlAdxTvqdThP-5HDknWFWmhhEusKzujYk0AFV5H00TZcqn0KdpyfqHRLPjnvnfKEpyfqHnsnj0YnsKWpyfqP1cvrHnz0AqLUWYs0ZK45HcsP6KWThnqn1b3nWf)




基于opencv的实时视频边缘检测处理 (http://download.csdn.net/download/c20081052/9522292)

(http://download.csdn.net/download/c20081052/9522292) 2015年07月24日 10:18 631B 下载

基于win10，GPU的Tensorflow Object Detection API部署及USB摄像头目标检测 (http://blog.csdn.net/github\_38705794/article/details/77776522)




TensorFlow安装 下载 TensorFlow models 配置依赖库 程序配置和运行1.TensorFlow安装安装教程在CSDN上有很多文章，但最好依据官方教程，因为TensorFlow...

 asukasmlriver (<http://blog.csdn.net/asukasmlriver>) 2017年12月02日 15:46 085


VLC for Android 基于 Opencv 对 RTSP视频 实时人脸检测 ([http://blog.csdn.net/chang\\_113...](http://blog.csdn.net/chang_113...))

最近项目上需要在Android客户端 通过获取 RTSP 的视频进行实时人脸检测，要做就是以下几点： 1、通过VLC 获取 获取 RTSP 2、对VLC中播放的视频进行实时截屏并保存在SD卡中 ...

 chang\_1134 ([http://blog.csdn.net/chang\\_1134](http://blog.csdn.net/chang_1134)) 2016年09月07日 15:02 2362


opencv-视频处理-实时前景检测-阈值法 (<http://blog.csdn.net/wxcdzhangping/article/details...>)

阈值法：对每一帧进行阈值处理。比较适合做晚上的情况，车灯的检测等，运动物体的亮度明显大于周围环境的情况。...

 wxcdzhangping (<http://blog.csdn.net/wxcdzhangping>) 2016年07月19日 08:59 8813


程 | 深度学习 + OpenCV，Python 实现实时视频目标检测 机器之心 09-21 (<http://blog.csdn.n...>)

教程 | 深度学习 + OpenCV，Python 实现实时视频目标检测 机器之心 09-21 选自 PyimageSearch 机器之心编译 参与：路雪、李泽南 使用 ...

 weixin\_40355324 ([http://blog.csdn.net/weixin\\_40355324](http://blog.csdn.net/weixin_40355324)) 2017年12月04日 21:29 9


opencv-视频处理-实时的前景检测-Vibe算法 (<http://blog.csdn.net/wxcdzhangping/article/de...>)

vibe算法是一种像素级的前景检测算法，实时性高，内存占有率低，前景检测准确率高。但是会出现“鬼影”。《ViBe: a powerful random technique to estimate t...

 wxcdzhangping (<http://blog.csdn.net/wxcdzhangping>) 2016年07月08日 17:34 8981


opencv-视频处理-实时的前景检测-Vibe算法 ([http://blog.csdn.net/Real\\_Myth/article/details/...](http://blog.csdn.net/Real_Myth/article/details/...))

vibe算法是一种像素级的前景检测算法，实时性高，内存占有率低，前景检测准确率高。但是会出现“鬼影”，当然基于对鬼影的处理，也会有相应的对vibe算法的改进。把下面三篇文章看明白，基本就会掌握vi...

 Real\_Myth ([http://blog.csdn.net/Real\\_Myth](http://blog.csdn.net/Real_Myth)) 2017年02月13日 11:29 898

opencv-视频处理-实时前景检测--三帧差法 (<http://blog.csdn.net/wxcdzhangping/article/deta...>)

假设下面的是视频流中的按时间先后顺序的任意三帧图片依次定义它们的灰度图为： $I_1, I_2, I_3$ ，代表在任意一点代表定义前两张的灰度的差为：定义后两张的灰度差为：...

 wxcdzhangping (<http://blog.csdn.net/wxcdzhangping>) 2016年07月07日 09:46 6052

Dell-1080Ti 深度学习Win 10 + Ubuntu 16.04.02 双系统安装+anaconda2+tensorflow+openc...

电脑基本参数：Dell 台式机，CPU i7，GPU 1080Ti，系统win10，SSD 512G HDD：2T 目的：双系统，Win10 安装在固态硬盘；Ubuntu安装在机械硬盘；并在Ubu...



tulip561 (<http://blog.csdn.net/tulip561>) 2017年06月29日 00:56 430

## 网络摄像机IPCamera RTSP直播播放网络/权限/音视频数据/花屏问题检测与分析助手EasyRTS...

前言最近在项目中遇到一个奇怪的问题，同样的SDK调用，访问海康摄像机的RTSP流，发保活OPTIONS命令保活，一个正常，而另一个一发就会被IPC断开，先看现场截图：图1：发OPTIONS，摄像机立...

xiejiaashu (<http://blog.csdn.net/xiejiaashu>) 2017年03月11日 00:26 2749

## 6用OpenCV编写的一个用来检测人脸的程序，可输入图片或者视频文件，也...

<http://download.csdn.net/detail/mikedadong/77776522> 2014年03月13日 00:40 342KB 下载

## win10+VS2013+OPENCV如何配置于仕琪人脸检测算法 ([http://blog.csdn.net/qz\\_23947237/a...](http://blog.csdn.net/qz_23947237/article/details/77776522))

win10+VS2013+OPENCV如何配置于仕琪人脸检测算法 从opencv中文论坛见识了于老师的人脸检测算法，感觉很厉害，导师让我们自己在电脑上跑一下人家的程序。于是在github上下载了解...

qq\_23947237 ([http://blog.csdn.net/qz\\_23947237](http://blog.csdn.net/qz_23947237)) 2016年07月19日 23:04 2154

## win10的visual studio 2015 安装opencv3.1，并且编写demo人脸检测 ([http://blog.csdn.net/...](http://blog.csdn.net/mikedadong/article/details/77776522))

Face Detection Using OpenCVWhat is OpenCVOpenCV (Open Source Computer Vision) is released under a BS...

mikedadong (<http://blog.csdn.net/mikedadong>) 2016年11月27日 16:19 1080

## OpenCV3.2+Qt5.8.0+Win10开发视频监控系统----（1）Qt之设置应用程序图标 ([http://blog.cs...](http://blog.csdn.net/qz_38880380/article/details/77776522))

OpenCV3.2+Qt5.8.0+Win10开发视频监控系统----（1）Qt之设置应用程序图标 设置 首先，创建一个图标格式的位图文件，其中...

qq\_38880380 ([http://blog.csdn.net/qz\\_38880380](http://blog.csdn.net/qz_38880380)) 2017年12月04日 16:03 211

## python+openCV利用cascade分类器训练实现实时视频车辆检测与车牌检测（一） ([http://blo...](http://blog.csdn.net/cyy930917/article/details/77776522))

研究生期间要做一个用行车记录仪抓拍车辆违章的项目，车辆检测算法是其中的一个组成部分，我选择的实现方式是用opencv来实现，废话不多说，先从动机开始说。 1.为什么处理视频选择的是检测而...

cyy930917 (<http://blog.csdn.net/cyy930917>) 2016年12月16日 14:36 4566

0

