# Windows中Anaconda(Python3.6)配置OpenCV3.3+深度神经网络实现

OpenCV3.3的发布带来的深度学习dnn模块的支持，模块支持

* Caffe
* Tensorflow
* Torch

等多个深度学习框架，并且有多个网络API接口

* GoogleLeNet
* AlexNet
* SqueezeNet
* VGGNet
* ResNet

等等。贡献者Rynikov Alexander对项目的详细介绍可以点击：[这里](https://habrahabr.ru/company/intel/blog/333612/" \t "_blank)查看。但是是用俄语写的，可以谷歌翻译来看。以下是不同网络的性能对比（翻译的不太好），另有[官方链接](http://www.pyimagesearch.com/2017/08/21/deep-learning-with-opencv/" \t "_blank),有很详细的功能说明和代码解释。



图1.性能测试

废话不多说，以下是我Anaconda(Python3.6)中配置OpenCV3.3的过程，以及深度神经网络示例程序的实现

### ****一、Anaconda安装****

windows中安装十分简单，[下载](https://www.continuum.io/downloads" \l "windows" \t "_blank)对应版本，根据官网指南安装即可，不再赘述。

### ****二、opencv3.3安装****

要在Anaconda中安装opencv3.3需要先下载安装包，然后启动Anaconda虚拟环境，利用pip命令即可安装。

* opencv3.3下载地址：[http://www.lfd.uci.edu/~gohlke/pythonlibs/#opencv](http://www.lfd.uci.edu/~gohlke/pythonlibs/" \l "opencv" \t "_blank)
* 选择相应版本，我根据系统选择64位python3.6的 如图2所示。

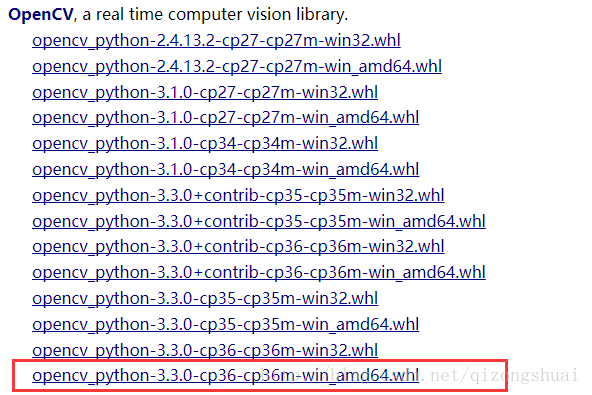
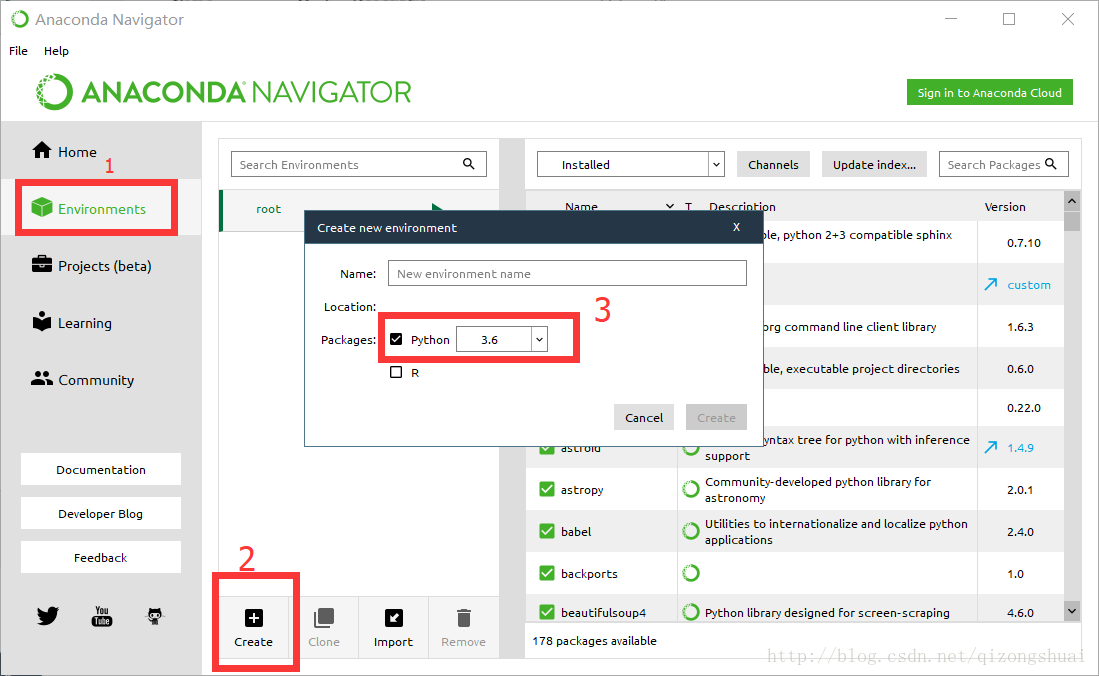


图2 OpenCV3.3 python安装版本下载

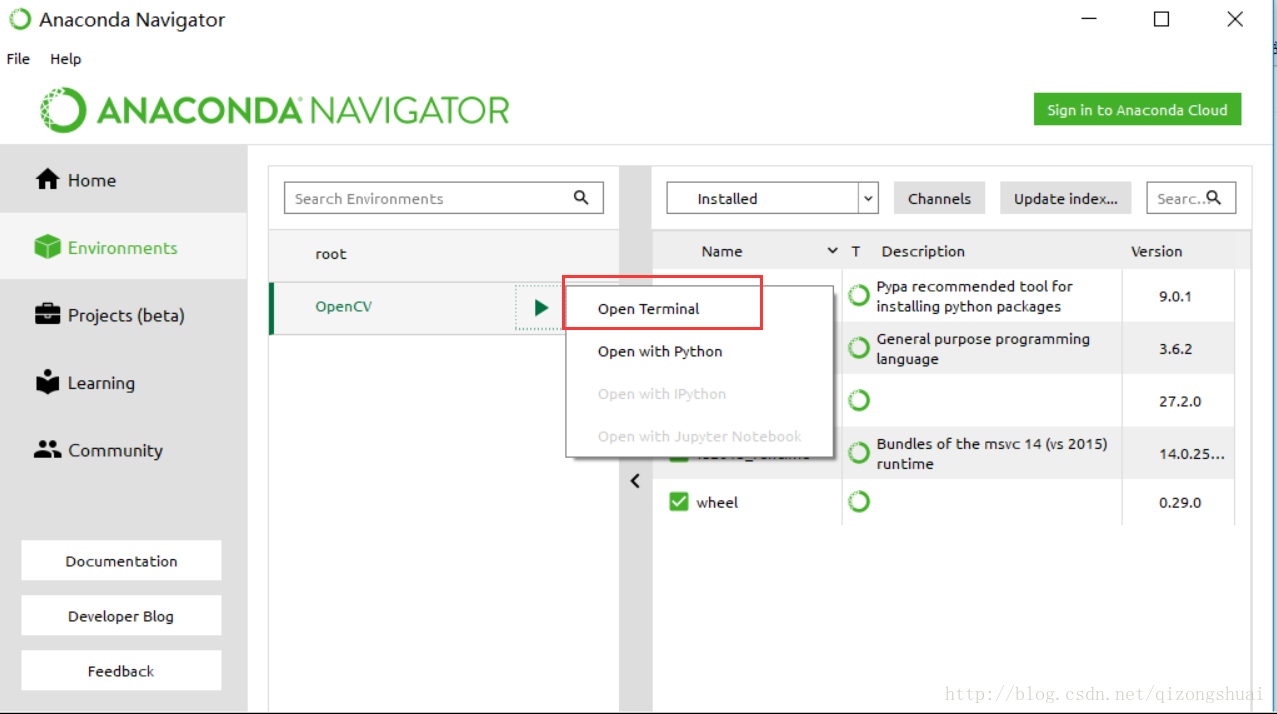
刚才已经安装好了Anaconda，现在在win程序启动处打开Anaconda Navigatro



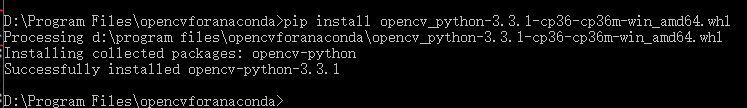
创建一个运行环境，选择python对应版本，起名opencv，我这里选择python3.6



在新建的opencv环境中打开一个命令终端



**在新打开的终端中，进入下载的opencv安装文件，执行pip安装，安装成功，没有什么问题**

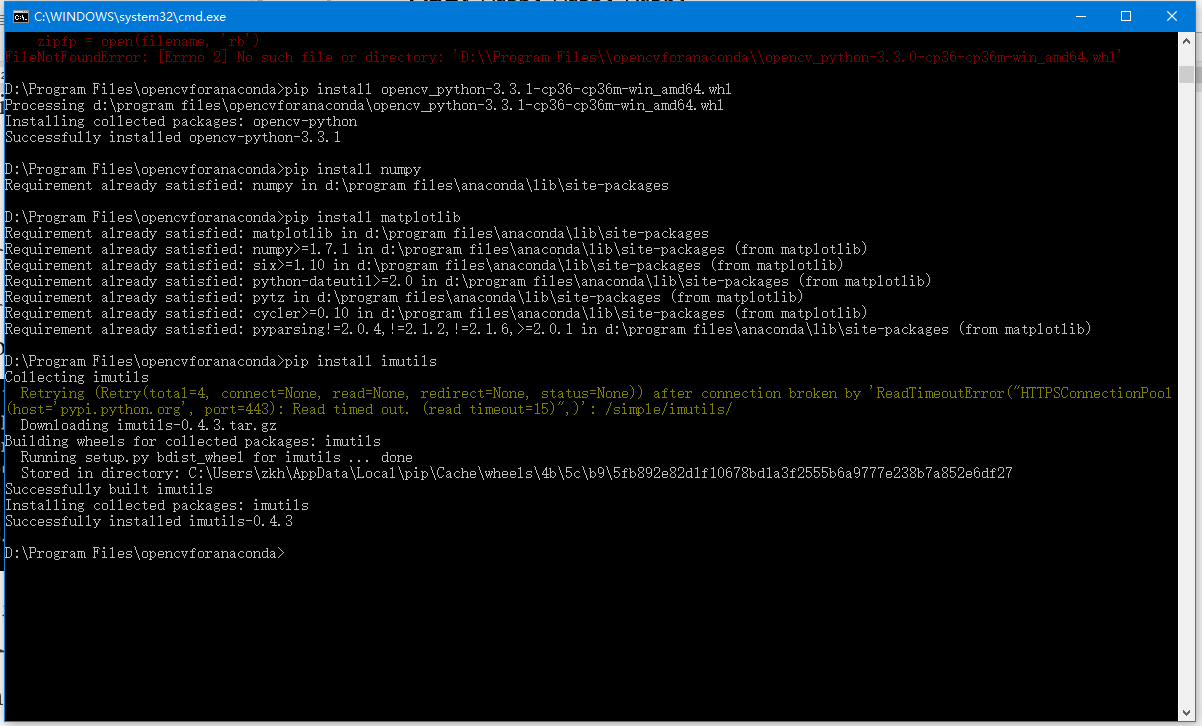


接下来要安装numpy和matplotlib，分别执行下列代码进行安装

pip install numpy

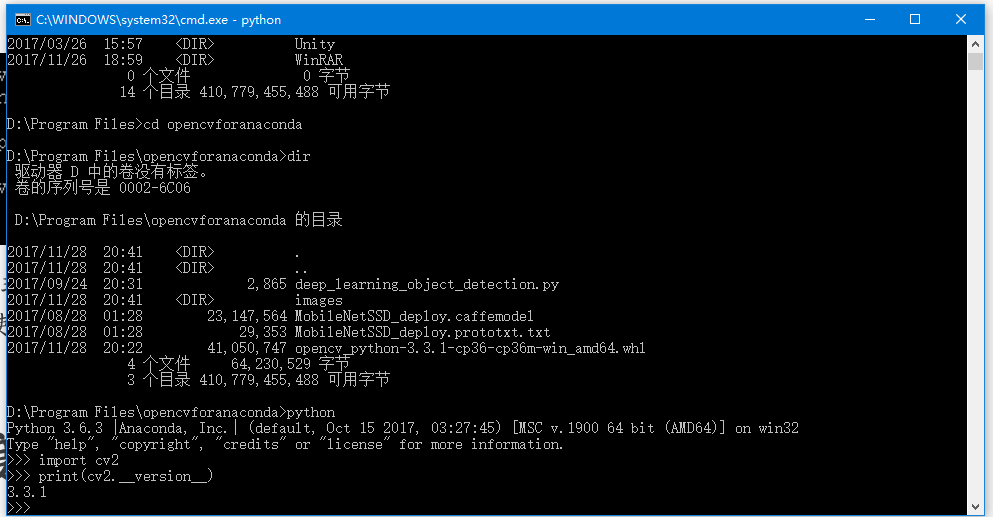
pip install matplotlib

pip install imutils(私人开发的opencv辅助图像处理工具)



目前位置，已经完成了Anaconda的opencv3.3的安装，在终端中运行测试，可以看出，链接库成功，打印出了3.3版本

import cv2



注：这种安装方式，只能在刚才创建的名为opencv的Anaconda环境中能使用，在普通cmd命令窗口不能使用opencv，原因是要另外配置环境变量，因为我的电脑里已经有一个opencv2.4.13了，为了避免两者混淆，我只在这个环境里使用。

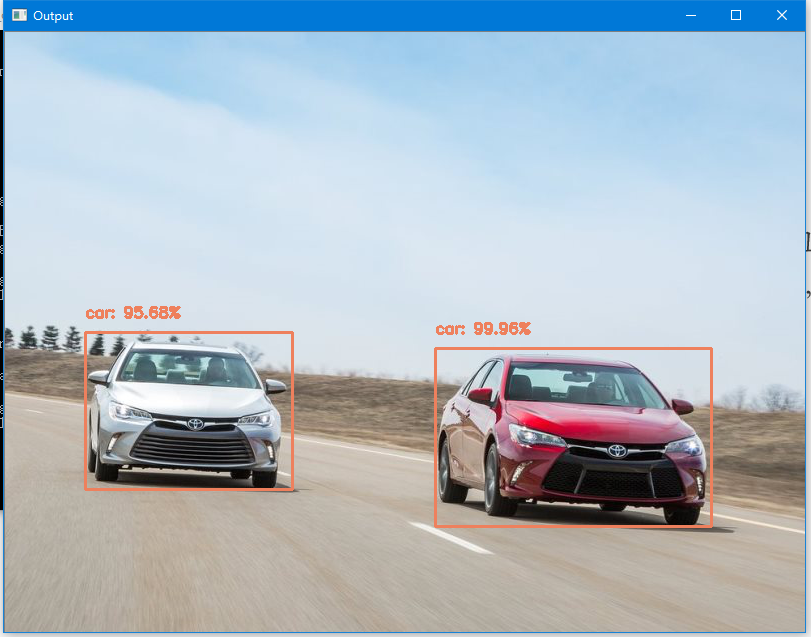
### ****三、深度神经网络的实现****

官网给出了示例代码，下载下来，跑一跑。

* 官网地址[http://www.pyimagesearch.com/2017/08/21/deep-learning-with-opencv/](http://www.pyimagesearch.com/2017/08/21/deep-learning-with-opencv/" \t "_blank)
* 在页面最下面填写邮箱地址，根据提示步骤可以直接获取示例程序源代码
* 获取源代码以后，解压，可以看到里面有写好的示例程序和测试图片
* 打开新建的名为opencv的环境里的终端，进入刚才下载的文件夹内，执行以下命令：

python deep\_learning\_object\_detection.py --image images/example\_01.jpg --prototxt MobileNetSSD\_deploy.prototxt.txt --model MobileNetSSD\_deploy.caffemodel

* 可以看到，终端上输出了对物体识别的得分情况，并且在图片上打上了识别效果



可见，有些识别的效果有点不忍直视，但是有些结果还是很不错了。而且速度也很快，这次识别所用的是googlelenet的caffe模型，也只能用跑好的网络来forward，想用opencv来训练估计还是不太行。总之，opencv提供的这个接口算是给全民ai做了不小贡献呢！

### ****四、函数框架简介****

* 在dnn中从磁盘加载图片：   
  cv2.dnn.blobFromImage   
  cv2.dnn.blobFromImages
* 用“create”方法直接从各种框架中导出模型：   
  cv2.dnn.createCaffeImporter   
  cv2.dnn.createTensorFlowImporter   
  cv2.dnn.createTorchImporter
* 使用“读取”方法从磁盘直接加载序列化模型：   
  cv2.dnn.readNetFromCaffe   
  cv2.dnn.readNetFromTensorFlow   
  cv2.dnn.readNetFromTorch   
  cv2.dnn.readhTorchBlob

从磁盘加载完模型之后，可以用.forward方法来向前传播我们的图像，获取分类结果

### 参考：

* <http://qiita.com/FukuharaYohei/items/5d49938ffd33d198f0c0>（需要google翻译）
* <http://qingmang.me/articles/-8015587844313365323/>量子位期刊
* <https://stackoverflow.com/questions/23119413/how-to-install-python-opencv-through-conda>
* <http://www.pyimagesearch.com/2017/08/21/deep-learning-with-opencv/>