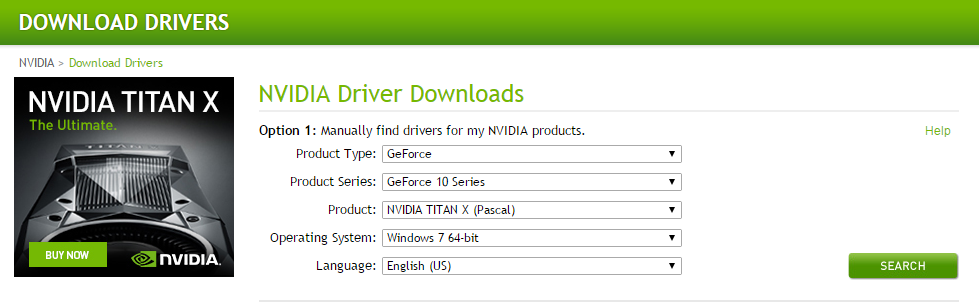
安装环境：

硬件：微星x99主板、i76800、GTX1080

系统：ubuntu16.04、python2.7

1. [安装nvidia驱动](http://blog.csdn.net/autocyz/article/details/52299889#t0)
2. [安装CUDA](http://blog.csdn.net/autocyz/article/details/52299889#t1)
3. [测试cuda的Samples](http://blog.csdn.net/autocyz/article/details/52299889#t2)
4. [使用cudnn](http://blog.csdn.net/autocyz/article/details/52299889#t3)
5. [matlab的安装与配置](http://blog.csdn.net/autocyz/article/details/52299889#t4)
6. [安装opencv310](http://blog.csdn.net/autocyz/article/details/52299889#t5)
7. [安装caffe](http://blog.csdn.net/autocyz/article/details/52299889#t6)

**1、安装nvidia驱动**

首先去官网上查看适合你GPU的驱动（<http://www.nvidia.com/Download/index.aspx?lang=en-us>）   
例如，本人的GPU适合的驱动如图：   


执行如下语句，安装

sudo add-apt-repository ppa:graphics-drivers/ppa

sudo apt-get update

sudo apt-get install nvidia-367

sudo apt-get install mesa-common-dev

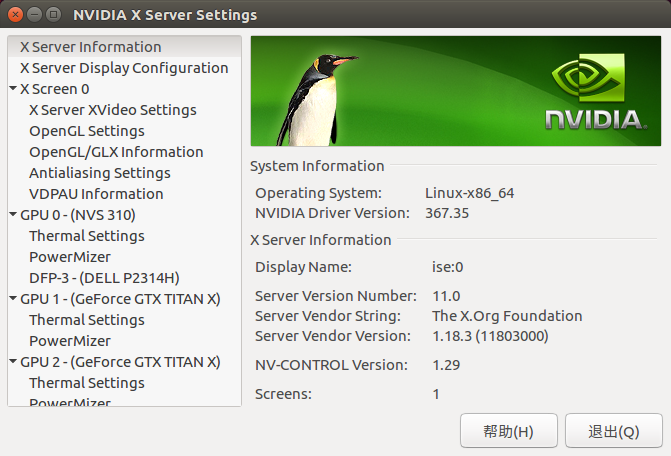
sudo apt-get install freeglut3-dev

执行完上述后，重启（reboot）。   
重启后输入：

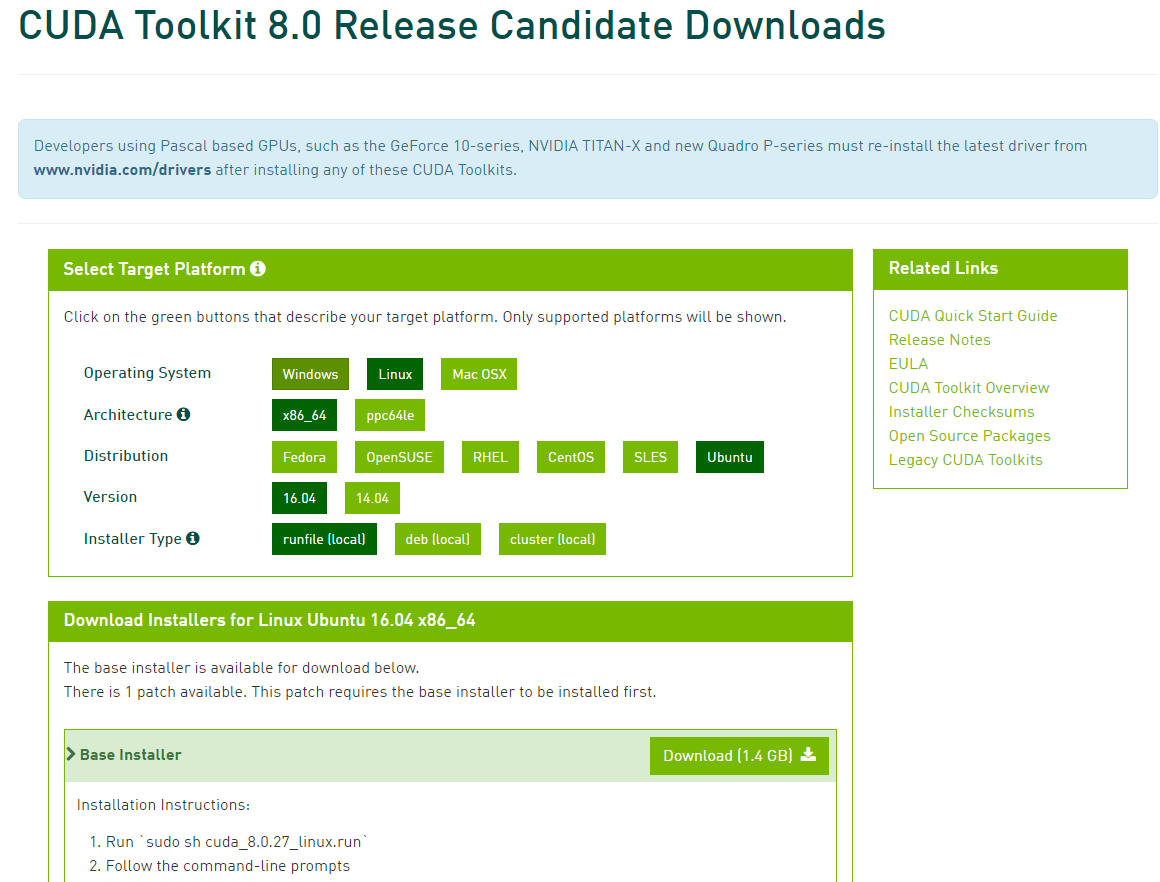
nvidia-smi

如果出现了你的GPU列表，则说明驱动安装成功了。另外也可以通过

nvidia-settings

查看自己机器上详细的GPU信息，本人机器的信息如下：   


**2、安装CUDA**

cuda是nvidia的编程语言平台，想使用GPU就必须要使用cuda。   
从这里下载cuda的安装文件   
<https://developer.nvidia.com/cuda-release-candidate-download>   
  
注意这里下载的是cuda8.0的runfile（local）文件。   
这里是nvidia给出的官方安装指南（遇到问题时可以查阅）：   
<http://docs.nvidia.com/cuda/cuda-installation-guide-linux/#axzz4HIBXnwyt>   
下载完cuda8.0后，执行如下语句，运行runfile文件：

sudo sh cuda\_8.0.27\_linux.run

执行后会有一系列提示让你确认，但是注意，有个让你选择是否安装nvidia361驱动时，一定要选择否，因为前面我们已经安装了更加新的nvidia367，所以这里不要选择安装。其余的都直接默认或者选择是即可。   
安装成功后会出现如下界面：

===========

= Summary =

===========

Driver: Not Selected

Toolkit: Installed in /usr/local/cuda-8.0

Samples: Installed in /home/textminer

Please make sure that

– PATH includes /usr/local/cuda-8.0/bin

– LD\_LIBRARY\_PATH includes /usr/local/cuda-8.0/lib64, or, add /usr/local/cuda-8.0/lib64 to /etc/ld.so.conf and run ldconfig as root

To uninstall the CUDA Toolkit, run the uninstall script in /usr/local/cuda-8.0/bin

Please see CUDA\_Installation\_Guide\_Linux.pdf in /usr/local/cuda-8.0/doc/pdf for detailed information on setting up CUDA.

\*\*\*WARNING: Incomplete installation! This installation did not install the CUDA Driver. A driver of version at least 361.00 is required for CUDA 8.0 functionality to work.

To install the driver using this installer, run the following command, replacing with the name of this run file:

sudo .run -silent -driver

Logfile is /opt/temp//cuda\_install\_6583.log

安装完毕后，再声明一下环境变量，并将其写入到 ~/.bashrc 的尾部:

export PATH=/usr/local/cuda-8.0/bin${PATH:+:${PATH}}

export LD\_LIBRARY\_PATH=/usr/local/cuda-8.0/lib64${LD\_LIBRARY\_PATH:+:${LD\_LIBRARY\_PATH}}

然后设置环境变量和动态链接库，在命令行输入：

$ sudo gedit /etc/profile

在打开的文件末尾加入：

export PATH = /usr/local/cuda/bin:$PATH

保存之后，创建链接文件：

sudo gedit /etc/ld.so.conf.d/cuda.conf

在打开的文件中添加如下语句：

/usr/local/cuda/lib64

然后执行

sudo ldconfig

使链接立即生效。

**3、测试cuda的Samples**

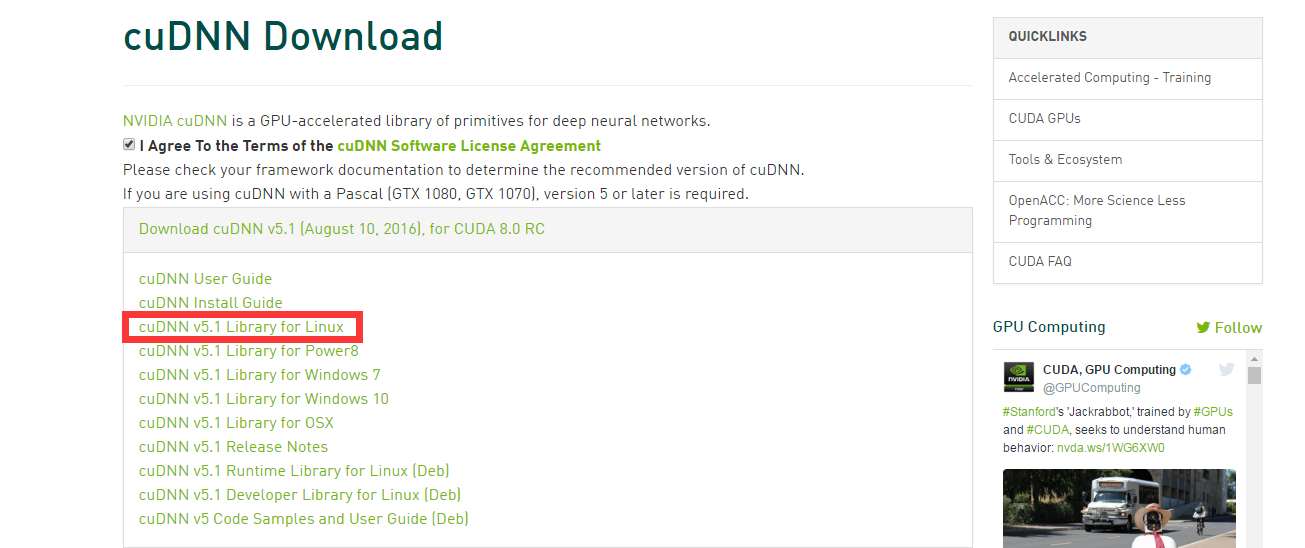
cd /usr/local/cuda-7.5/samples/1\_Utilities/deviceQuery

make

sudo ./deviceQuery

如果显示的是一些关于GPU的信息，则说明安装成功了。

**4、使用cudnn**

首先去官网下载你需要的cudnn，下载的时候需要注册账号。选择对应你cuda版本的cudnn下载。这里我下载的是cudnn5.1，是个压缩文件（.tgz）   
  
下载完cudnn5.0之后进行解压，cd进入cudnn5.1解压之后的include目录，在命令行进行如下操作：

sudo cp cudnn.h /usr/local/cuda/include/ #复制头文件

再将cd进入lib64目录下的动态文件进行复制和链接：

sudo cp lib\* /usr/local/cuda/lib64/ #复制动态链接库

cd /usr/local/cuda/lib64/

sudo rm -rf libcudnn.so libcudnn.so.5 #删除原有动态文件

sudo ln -s libcudnn.so.5.0.5 libcudnn.so.5 #生成软衔接

sudo ln -s libcudnn.so.5 libcudnn.so #生成软链接

**5、matlab的安装与配置**

在网盘上下载安装包<http://pan.baidu.com/s/1nuKJc9N>。里面有一个crack文件夹，用于破解；MATHWORKS\_R2014A.iso用于安装。

1、挂载iso（需新建matlab\_iso文件夹）：

sudo mount -o loop MATHWORKS\_R2014A.iso ～/matlab\_is

2、开始安装：

cd ~/matlab\_iso

sudo ./install

3、选择不联网安装

4、密钥随便输入，比如 12345-67890-12345-67890

5、激活：选择”license\_405329\_R2014a.lic”文件进行激活（在Crack文件夹下面）

6、将libmwservices.so复制到/usr/local/MATLAB/R2014a/bin/glnxa64中：

sudo cp libmwservices.so /usr/local/MATLAB/R2014a/bin/glnxa64/libmwservices.so

**6、安装opencv3.1.0**

从官网上下载opencv3.1.0   
<http://opencv.org/downloads.html>   
并将其解压到你要安装的位置，假设解压到了/home   
首先安装Ubuntu系统需要的依赖项，虽然我也不知道有些依赖项是干啥的，但是只管装就行，也不会占据很多空间的。

sudo apt-get install --assume-yes libopencv-dev build-essential cmake git libgtk2.0-dev pkg-config python-dev python-numpy libdc1394-22 libdc1394-22-dev libjpeg-dev libpng12-dev libtiff5-dev libjasper-dev libavcodec-dev libavformat-dev libswscale-dev libxine2-dev libgstreamer0.10-dev libgstreamer-plugins-base0.10-dev libv4l-dev libtbb-dev libqt4-dev libfaac-dev libmp3lame-dev libopencore-amrnb-dev libopencore-amrwb-dev libtheora-dev libvorbis-dev libxvidcore-dev x264 v4l-utils unzip

然后安装opencv需要的一些依赖项，一些文件编码解码之类的东东。

sudo apt-get install build-essential cmake git

sudo apt-get install ffmpeg libopencv-dev libgtk-3-dev python-numpy python3-numpy libdc1394-22 libdc1394-22-dev libjpeg-dev libpng12-dev libtiff5-dev libjasper-dev libavcodec-dev libavformat-dev libswscale-dev libxine2-dev libgstreamer1.0-dev libgstreamer-plugins-base1.0-dev libv4l-dev libtbb-dev qtbase5-dev libfaac-dev libmp3lame-dev libopencore-amrnb-dev libopencore-amrwb-dev libtheora-dev libvorbis-dev libxvidcore-dev x264 v4l-utils unzip

在终端中cd到opencv文件夹下，然后

mkdir build #新建一个build文件夹，编译的工程都在这个文件夹里

cd build/

cmake -D CMAKE\_BUILD\_TYPE=RELEASE -D CMAKE\_INSTALL\_PREFIX=/usr/local -D WITH\_TBB=ON -D WITH\_V4L=ON -D WITH\_QT=ON -D WITH\_OPENGL=ON -DCUDA\_NVCC\_FLAGS="-D\_FORCE\_INLINES" ..

cmake成功后，会出现如下结果，提示配置和生成成功：

-- Configuring done

-- Generating done

-- Build files have been written to: /home/ise/software/opencv-3.1.0/build

然后make编译就可以了

make -j8

上面是将opencv编译成功，但是并没有安装到我们的系统中，有很多的设置都没有写入到系统中，因此还要进行install。

sudo make install

sudo /bin/bash -c 'echo "/usr/local/lib" > /etc/ld.so.conf.d/opencv.conf'

sudo ldconfig

重启系统，重启系统后cd到build文件夹下：

sudo apt-get install checkinstall

sudo checkinstall

然后按照提示安装就可以了。   
使用checkinstall的目的是为了更好的管理我安装的opencv，因为opencv的安装很麻烦，卸载更麻烦，其安装的时候修改了一大堆的文件，当我想使用别的版本的opencv时，将当前版本的opencv卸载就是一件头疼的事情，因此需要使用checkinstall来管理我的安装。   
执行了checkinstall后，会在build文件下生成一个以backup开头的.tgz的备份文件和一个以build开头的.deb安装文件，当你想卸载当前的opencv时，直接执行dpkg -r build即可。

**7、安装caffe**

首先安装各种依赖包

sudo apt-get update

sudo apt-get install -y build-essential cmake git pkg-config

sudo apt-get install -y libprotobuf-dev libleveldb-dev libsnappy-dev libhdf5-serial-dev protobuf-compiler

sudo apt-get install -y libatlas-base-dev

sudo apt-get install -y--no-install-recommends libboost-all-dev

sudo apt-get install -y libgflags-dev libgoogle-glog-dev liblmdb-dev

sudo apt-get install -y python-pip

sudo apt-get install -y python-dev

sudo apt-get install -y python-numpy python-scipy

将终端cd到你要安装caffe的位置，执行如下指令，从github上clone caffe。

git clone https://github.com/BVLC/caffe.git //从github上git caffe

cd caffe //打开到刚刚git下来的caffe

sudo cp Makefile.config.example Makefile.config //将Makefile.config.example的内容复制到Makefile.config

//因为make指令只能make Makefile.config文件，而Makefile.config.example是caffe给出的makefile例子

sudo gedit Makefile.config //打开Makefile.config文件

打开之后修改如下内容：

//若使用cudnn，则将# USE\_CUDNN := 1 修改成： USE\_CUDNN := 1

//若使用的opencv版本是3的，则将# OPENCV\_VERSION := 3 修改为： OPENCV\_VERSION := 3

//若要使用python来编写layer，则需要将# WITH\_PYTHON\_LAYER := 1 修改为 WITH\_PYTHON\_LAYER := 1

//重要的一项 将# Whatever else you find you need goes here.下面的 INCLUDE\_DIRS := $(PYTHON\_INCLUDE) /usr/local/include LIBRARY\_DIRS := $(PYTHON\_LIB) /usr/local/lib /usr/lib

修改为： INCLUDE\_DIRS := $(PYTHON\_INCLUDE) /usr/local/include /usr/include/hdf5/serial

LIBRARY\_DIRS := $(PYTHON\_LIB) /usr/local/lib /usr/lib /usr/lib/x86\_64-linux-gnu /usr/lib/x86\_64-linux-gnu/hdf5/serial //这是因为ubuntu16.04的文件包含位置发生了变化，尤其是需要用到的hdf5的位置，所以需要更改这一路径

//若使用MATLAB接口的话，则要讲MATLAB\_DIR换成你自己的MATLAB安装路径

MATLAB\_DIR := /usr/local

MATLAB\_DIR := /usr/local/matlab2014a

打开makefile文件，

将

NVCCFLAGS +=-ccbin=$(CXX) -Xcompiler-fPIC $(COMMON\_FLAGS)

替换

NVCCFLAGS += -D\_FORCE\_INLINES -ccbin=$(CXX) -Xcompiler -fPIC $(COMMON\_FLAGS)

编辑/usr/local/cuda/include/host\_config.h，将其中的第115行注释掉：   
将

#error-- unsupported GNU version! gcc versions later than 4.9 are not supported!

改为

//#error-- unsupported GNU version! gcc versions later than 4.9 are not supported!

之后再

make all -j8

make runtest

make pycaffe

make matcaffe

注：

1、在安装opencv3.1时，可能会出现cudbu不匹配问题，这时候需要修改opencv中对cuda8.0限制文件，具体可以把错误google很容易得到解决，此处不具体列出

2、因为每个人的硬件软件各不相同，在安装时候，可能会碰到各种问题，这时候不要烦躁，一步一步看打印出来的日志，具体报的是什么错误，是缺少什么依赖库还是哪边编译不通过，一步一步解决

3、caffe依赖很多三方库这些库又可能依赖更上层的库，所以在安装时候可以分层安装，比如先安装gpu驱动，cuda算法库、python环境和依赖库，在安装matlab，然后在配置caffe编译变量，cmake、make

4、这是一个很精致的系统软件打包编译过程，大家可以按照这样思路去理解和编译它；问题会少很多

参考：

http://blog.csdn.net/lee\_j\_r/article/details/52693724