[一、安装 2](#_Toc1159207857)

[1、安装CUDA(8.0) 2](#_Toc1973463227)

[（1）我这里选择runfile安装 3](#_Toc76094623)

[（2）选择”.rpm”安装 4](#_Toc698622210)

[（3）解决方法 4](#_Toc60869775)

[（4）添加环境变量 6](#_Toc426868576)

[（5）解决一直安装默认仓库CUDA的方法 6](#_Toc284472126)

[（6）验证CUDA是否安装成功 7](#_Toc1375766079)

[cd /usr/local/cuda-8.0/samples 7](#_Toc753149372)

[2、安装CUDNN 7](#_Toc2132973213)

[3、安装OPENCV 8](#_Toc1073287716)

[a）安装依赖 8](#_Toc1410000739)

[b）编译opencv 9](#_Toc1981949602)

[C）参考网址 11](#_Toc902064084)

[d）总结 11](#_Toc1595459048)

[二、使用Yolo 11](#_Toc123906991)

[1、安装Yolo 11](#_Toc1754526557)

[（1）下载源码 11](#_Toc1245136447)

[（2）下载权重文件 12](#_Toc1928055264)

[（3）测试 12](#_Toc1016984615)

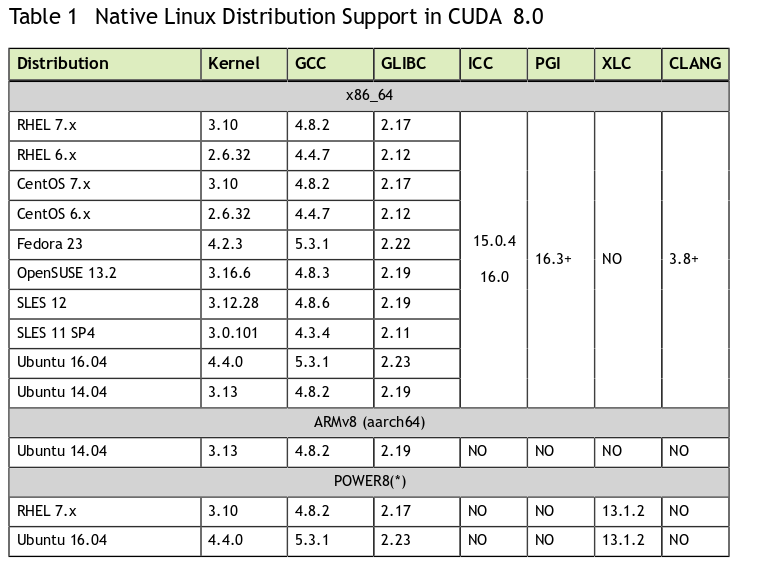
[三、杂记 13](#_Toc1101822329)

**Yolo有关**

# 一、安装

环境：centos7.3 + Quadro K620 + 32G内存

## 1、安装CUDA(8.0)



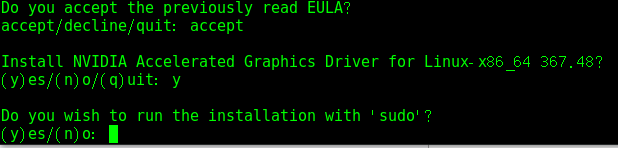
查看本机内核需要安装的一些依赖

下载CUDA网址：

<https://developer.nvidia.com/cuda-toolkit-archive>

参考安装网址：  
<https://www.jianshu.com/p/59f80d77d705>

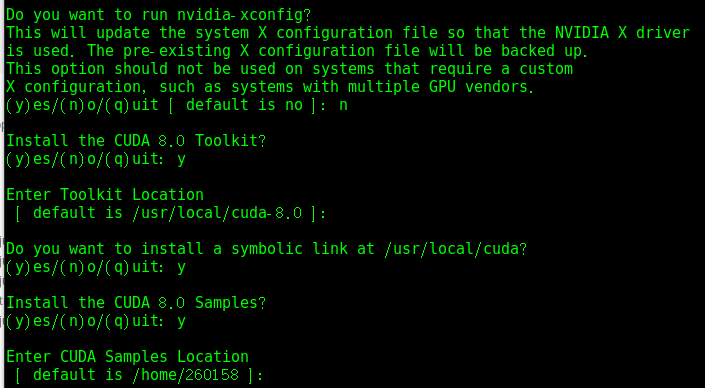
### （1）我这里选择runfile安装

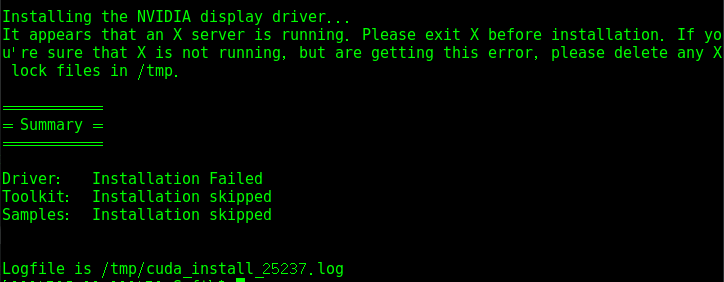


2017-12-30 09-48-47 的屏幕截图

2017-12-30 09-56-36 的屏幕截图

1. 接受协议？是
2. 同意安装驱动不？同意
3. 是否需要用sudo命令？用
4. 安装驱动需要用sudo命令？如果上一步选择是，则不会有这个问题。会提示你输密码。
5. 是否要安装OpenGL，一定不要安装。





按照以上的选择安装失败。

### （2）选择”.rpm”安装

这种方式也安装的不是我下载的。

解决以上的问题，删除NuxDextop仓库

vi /etc/yum.repos.d/nux-dextop.repo

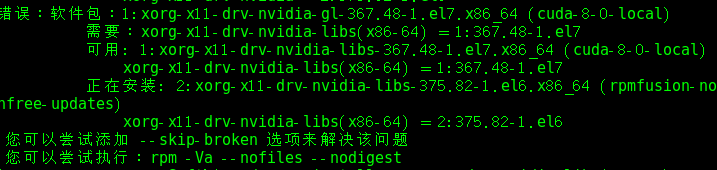
将文件中的enabled=1改成enabled=0，然后保存。

参考网址：

<http://blog.csdn.net/yaoqiwaimai/article/details/77163261>

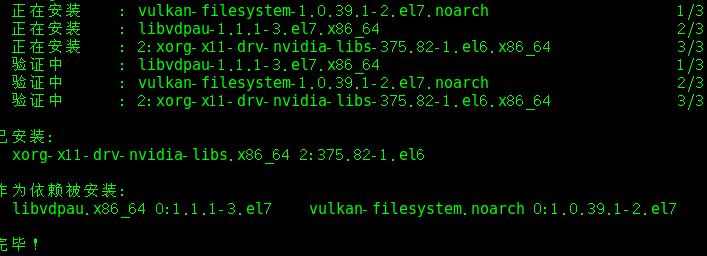
### （3）解决方法

会出现以下问题



解决办法1：

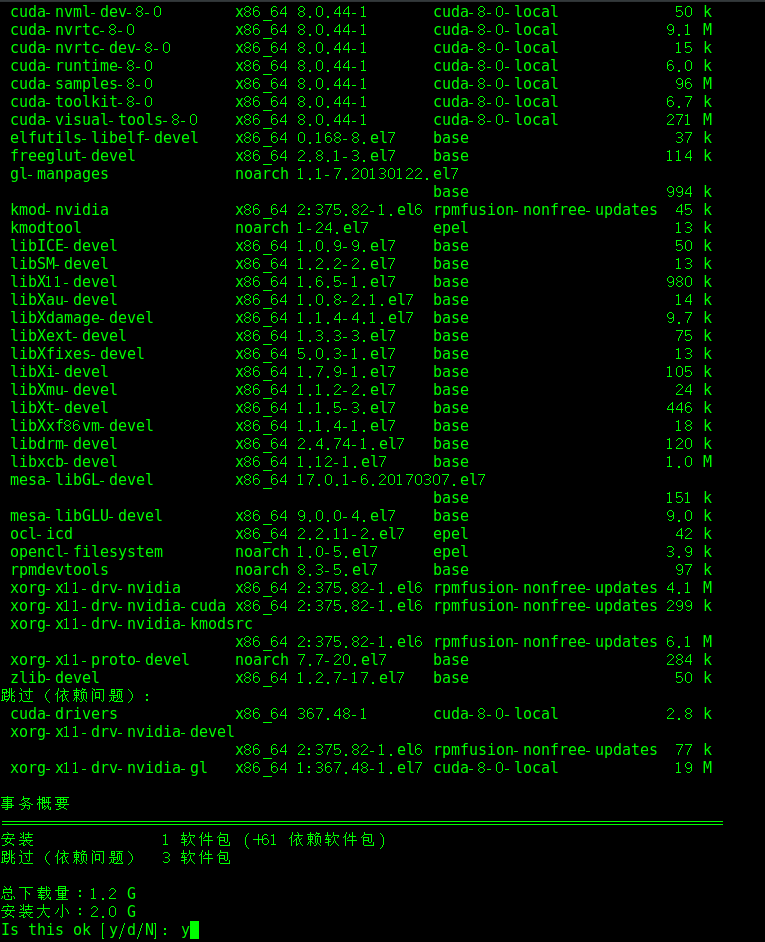
2017-12-30 16-59-20 的屏幕截图



事实证明如上办法不行。

解决办法2：

Sudo yum install --skip -broken

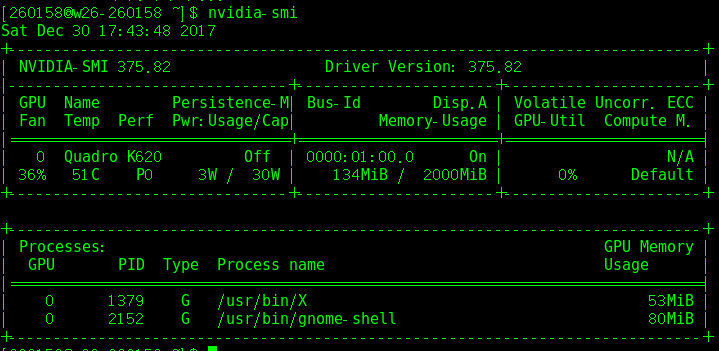


然后安装成功。

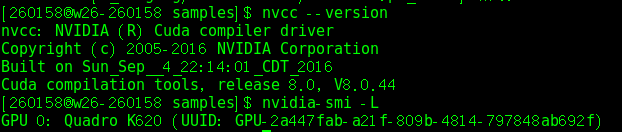
运行命令：nvidia-smi

结果：找不到命令

重启后，运行命令nvidia-smi



驱动加载成功。



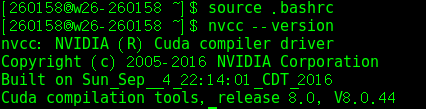
### （4）添加环境变量

添加环境变量：在～/ 目录下的.bashrc文件中添加

export CUDA\_HOME=/usr/local/cuda-8.0

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

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



### （5）解决一直安装默认仓库CUDA的方法

参考网址：

<http://blog.csdn.net/yaoqiwaimai/article/details/77163261>

Vi /etc/yum.repos.d/nux-dextop.repo

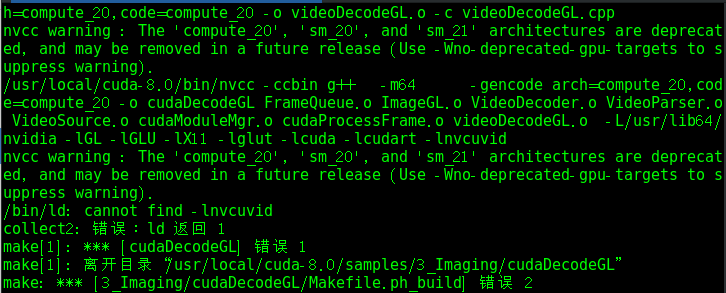
将文件中的enabled=1改成enabled=0，然后保存。

成功解决。

### （6）验证CUDA是否安装成功

## cd /usr/local/cuda-8.0/samples

Sudo make

出现以下的错误：  


目测是没有编译成功。待解决

测试cuda是否安装成功

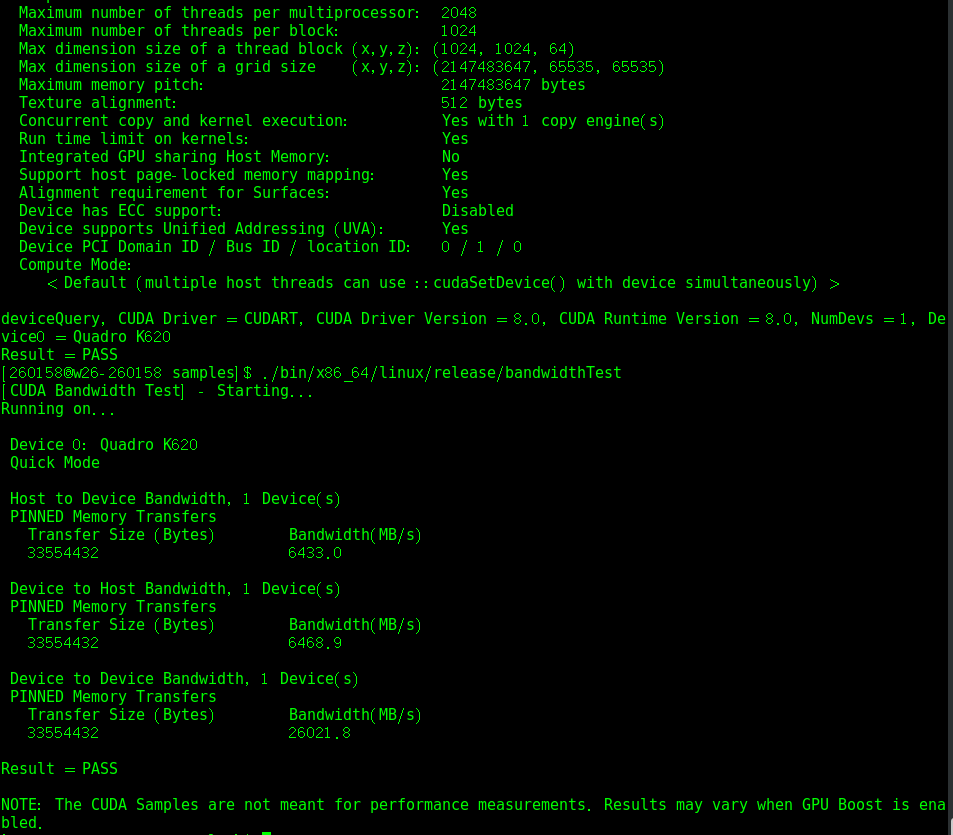
# cd /usr/local/cuda/samples/

# make

# ./bin/x86\_64/linux/release/deviceQuery

# ./bin/x86\_64/linux/release/bandwidthTest

显示 Result = PASS 则测试通过



虽然编译没有通过，但按照网上的资料，还是安装成功了。

## 2、安装CUDNN

下载网址：

<https://developer.nvidia.com/rdp/cudnn-download>

cuDNN是 CUDA的一个深度网络加速库，也是深度学习框架所需要的。安装cuDNN的过程实际上是下载cuDNN库并拷贝到CUDA的目录中。下载cuDNN库后，进入下载目录执行以下命令即完成安装。

tar xzvf cudnn-8.0-linux-x64-v7.tgz **#是7版本的**

sudo cp -P cuda/include/cudnn.h /usr/local/cuda/include

sudo cp -P cuda/lib64/lib\* /usr/local/cuda/lib64

sudo chmod a+r /usr/local/cuda/include/cudnn.h

sudo chmod a+r /usr/local/cuda/lib64/lib\*

因为没有软连接，所以下面的tensorflow报错。

设置软连接

cd /usr/local/cuda/lib64/

sudo rm -rf libcudnn.so libcudnn.so.7 #删除原有动态文件

sudo ln -s libcudnn.so.6.0.21 libcudnn.so.6 #生成软链接

sudo ln -s libcudnn.so.6 libcudnn.so #生成软链接

注意！！软链接中的数字7.0.5等参照文件夹lib64中原有的软链接名

sudo ldconfig

## 3、安装OPENCV

我安装的版本是2.4.10，安装教程去官网查找，需要安装依赖。

网址：<https://opencv.org/>

步骤

### a）安装依赖

--->  添加源

sudo yum -y install epel-release

--->gcc,g++等这些基本的工具,一般都有，没有的话再安装

sudo yum -y install gcc gcc-c++ cmake python-devel numpy

--->然后是opencv的依赖项

sudo yum -y install gtk2-devel gimp-devel gimp-devel-tools gimp-help-browser zlib-devel libtiff-devel libjpeg-devel libpng-devel libavc1394-devel libraw1394-devel libdc1394-devel jasper-devel jasper-utils swig python libtool nasm libv4l-devel gstreamer-plugins-base-devel

--->没有安装成功的包：

2017-12-30 18-56-51 的屏幕截图

Notes：安装的时候需要一大堆依赖（宁可信其有），且官网ubuntu的依赖和centos的依赖有所区别，要注意。

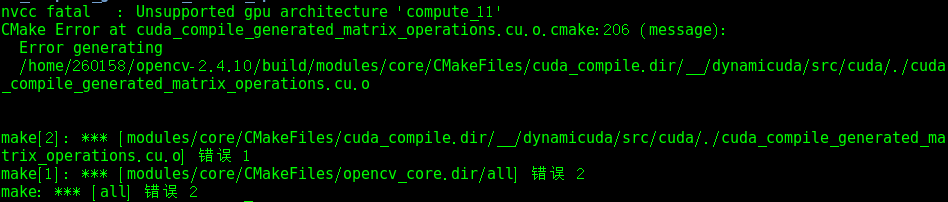
### b）编译opencv

我这里是已经安装成功CUDA后，开始编译OPENCV，所有中间会与没有装CUDA的有些出入。耗时也比较长。

按照官网步骤来

<https://docs.opencv.org/master/d7/d9f/tutorial_linux_install.html>

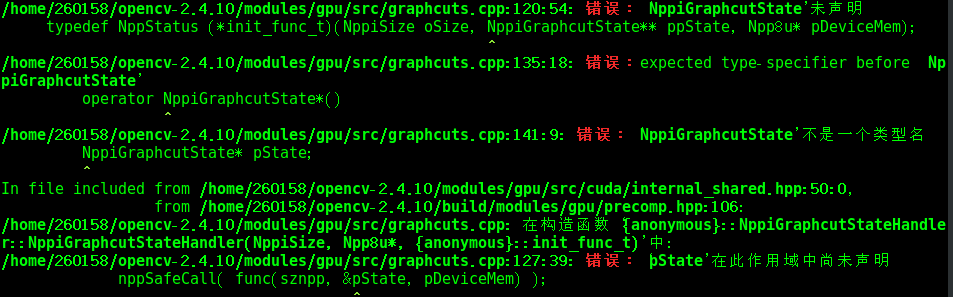
--->make错误1

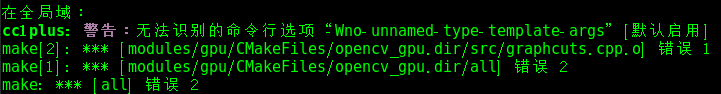


解决办法：

cmake -D CMAKE\_BUILD\_TYPE=RELEASE -D CMAKE\_INSTALL\_PREFIX=/usr/local -D CUDA\_GENERATION=Kepler ..

--->make错误2





解决办法：

cuda8.0较新，opencv-2.4.11较早，要编译通过需要修改源码：

修改/data/opencv-2.4.11/modules/gpu/src/graphcuts.cpp

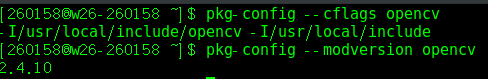
将 #if !defined (HAVE\_CUDA) || defined (CUDA\_DISABLER)

改为

#if !defined (HAVE\_CUDA) || defined (CUDA\_DISABLER) || (CUDART\_VERSION >= 8000)

重新编译即可。

安装成功



### C）参考网址

<http://blog.csdn.net/kakitgogogo/article/details/52490010>

<https://www.cnblogs.com/hxbbing/p/4894813.html>

### d）总结

注意: Anconda 自带是python-opencv，不属于opencv源码库， 加载其路径是不可以的，必须要编译opencv源码，生成库文 件和添加环境变量。参考以下网址：

<http://www.cnblogs.com/wyuzl/p/7889007.html>

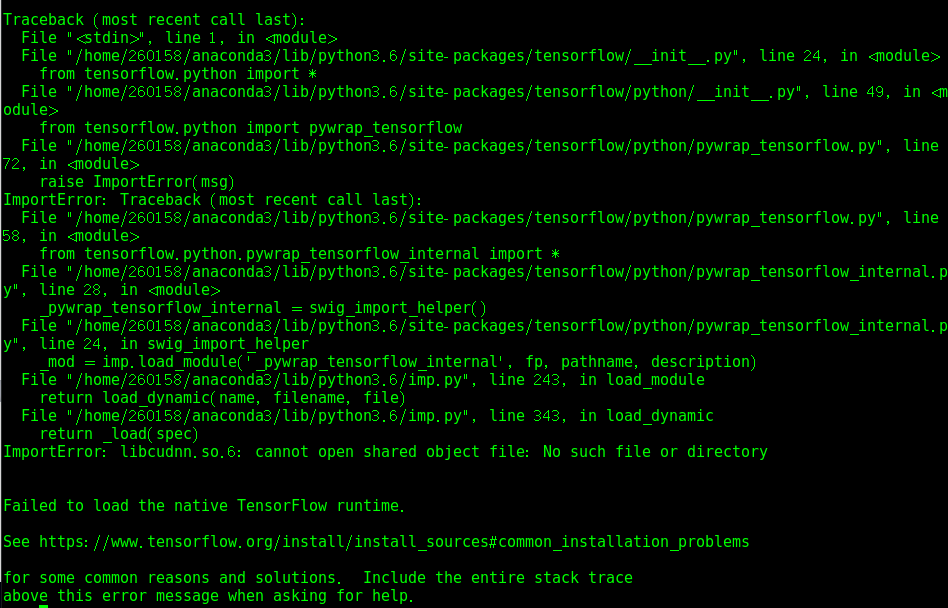
## 4、安装tensorflow-gpu

命令：pip install tensorflow

装的是最新版本的1.4.1

2018-01-02 13-51-02 的屏幕截图

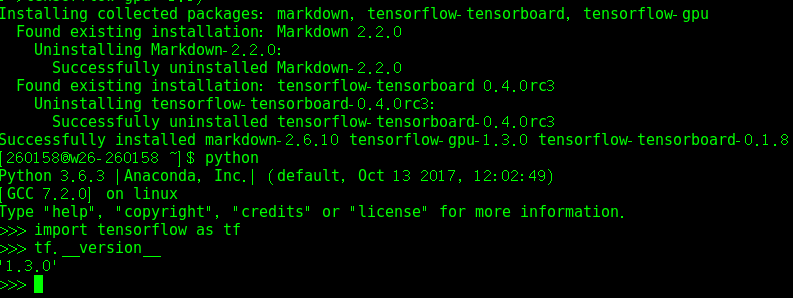
加载时出现错误



tensorflow使用pip在线安装是非常方便的，可最新版本偏偏不支持cuda8，仅支持cuda7.5.因此只能使用源码安装tensorflow.

截至2017.08.15 Tensorflow 尚不支持cuda9.0，所以建议用cuda 8.0 + cuDnn 6.0，这也是Tensorflow r1.3的缺省配置。

我现在改成tensorflow1.3，和上面对应的默认配置。

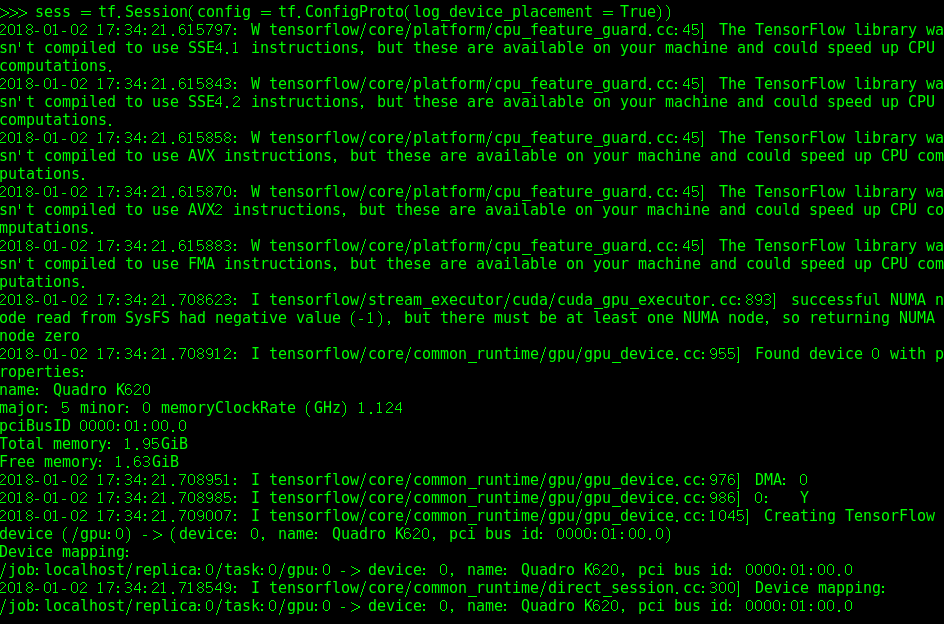


安装成功。

查看是否安装的是GPU版本

import tensorflow as tf

sess = tf.Session(config=tf.ConfigProto(log\_device\_placement=True))



安装成功。

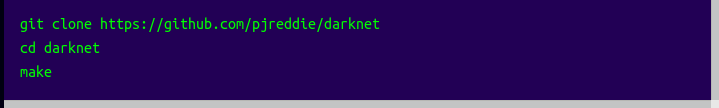
## 5、安装Caffe

# 二、使用Yolo

## 1、安装Yolo

官方网址：<https://pjreddie.com/darknet/yolo/>

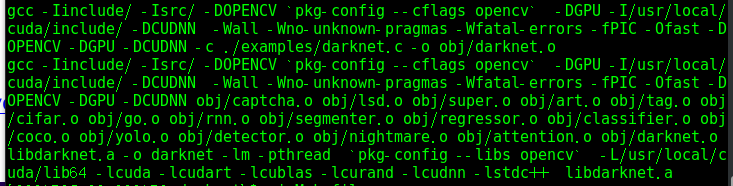
### （1）下载源码



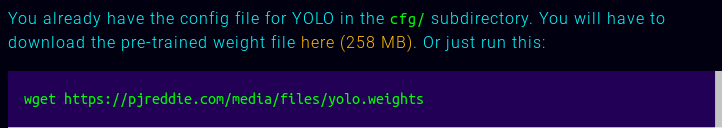
修改Makefile，根据自己的配置修改。将对应的0改为1。



然后进行编译,编译成功。

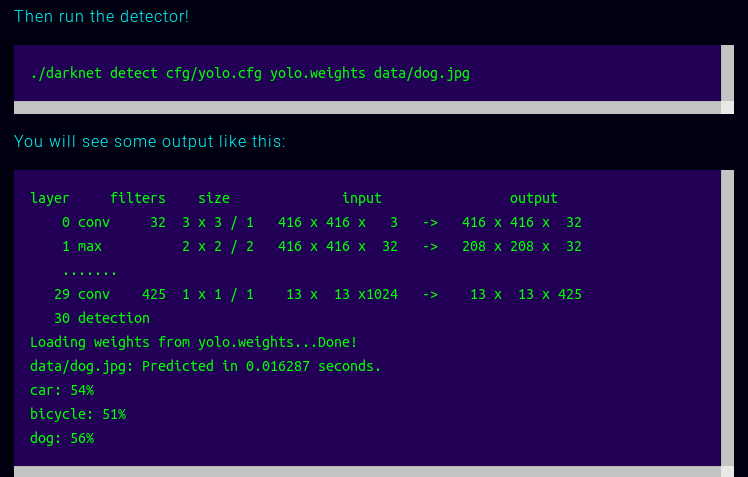


### （2）下载权重文件



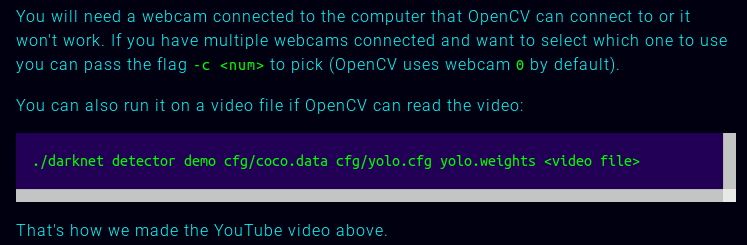
### （3）测试

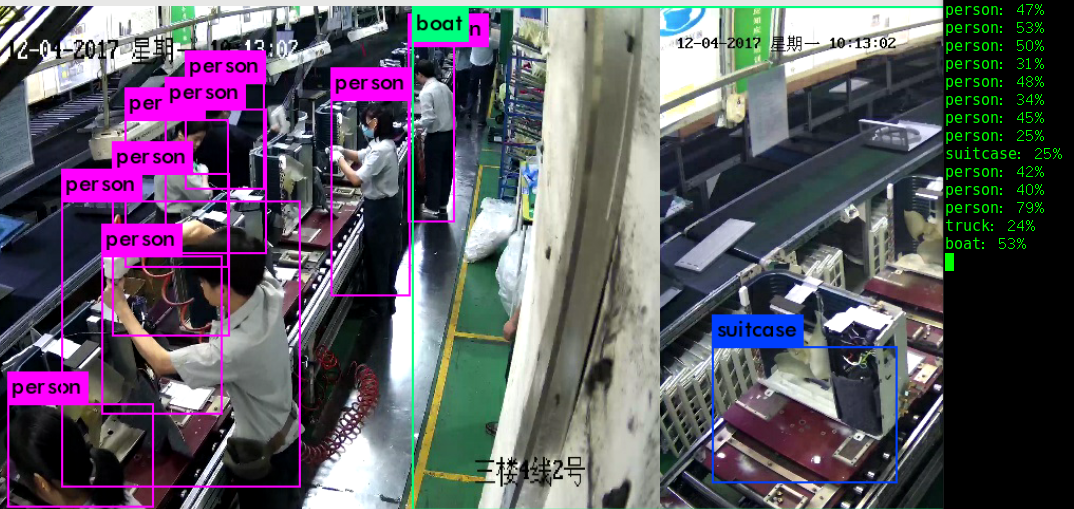
测试图片：



测试成功。

测试视频：





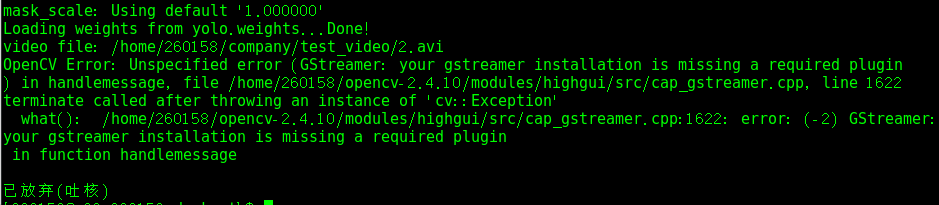
测试成功。

### （4）测试视频问题



后3个视频可以正常的读出来，但前3个视频是不可以的。可能是因为我后3个视频是自己用OPENCV生成的，而前3个是直接下载的。

错误如下：



在用opencv处理过后，就可以读取视频了。

## 2、训练自己的数据集

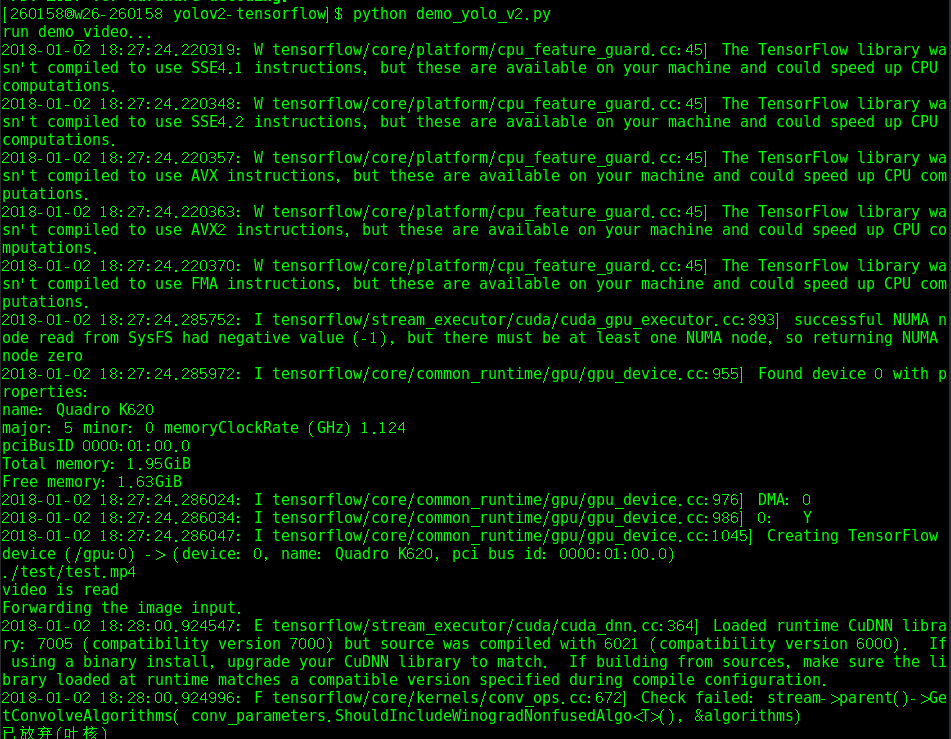
# 三、杂记

## 1、测试yolov2-tensorflow

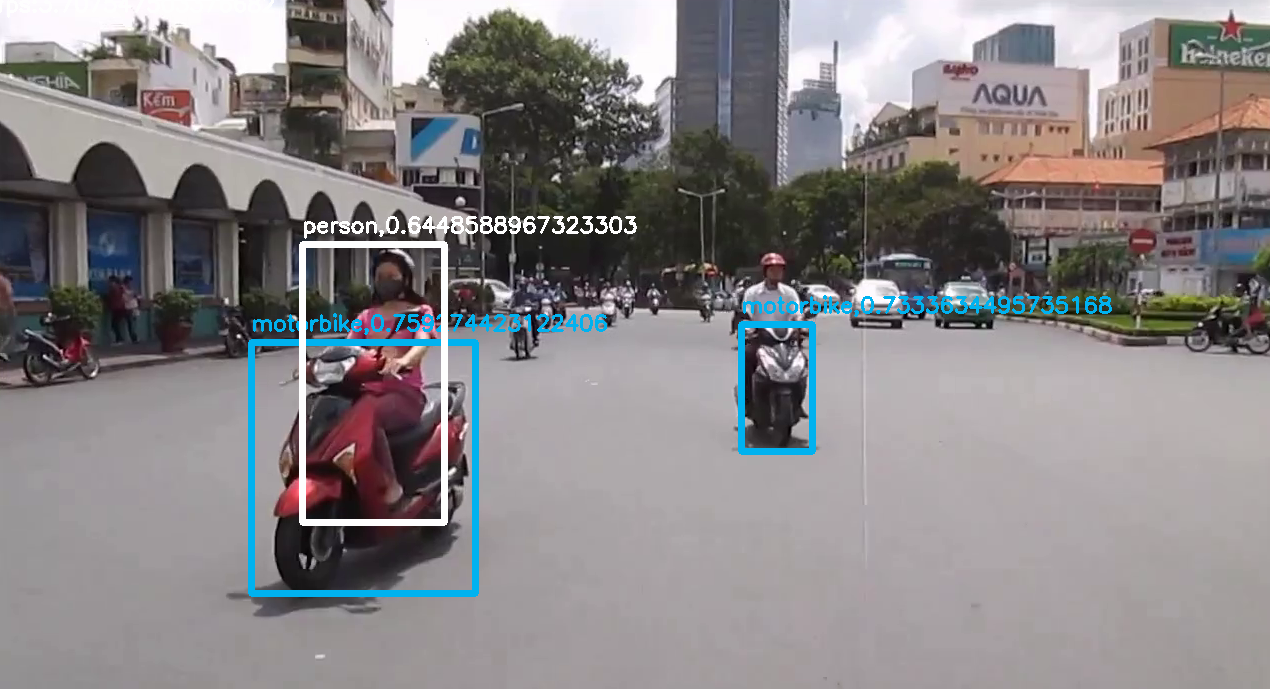
网址：<https://github.com/shishichang/yolov2-tensorflow>

出现错误：

大概意思就是源码是用CUDNN6.0.21版本编译的，不兼容，我的是v7版本的。



在改成v6后，测试自带视频，就成功了。如下：



## 2、QT配置

下载网址：

<https://www1.qt.io/download-open-source/?utm_referrer=https%3A%2F%2Fwww.qt.io%2Fdownload-qt-for-application-development#section-2>

用快捷方式打开的QT不能输入中文，

运行export QT\_IM\_MODULE=iBus

然后运行./qtcreator.sh，就可以输入中文了。

--->配置并运行OPENCV

在安装好Qt之后，只需要2步：

（1）新建项目

（2）修改pro文件

在pro文件里面，填写以下内容即可：

INCLUDEPATH += /usr/local/include \

/usr/local/include/opencv \

/usr/local/include/opencv2

LIBS += /usr/local/lib/libopencv\*