**Opencv安装**

本文档为<https://blog.csdn.net/cocoaqin/article/details/78163171>复制版，由于版主环境配置和流程与本项目完全相同，这里就不再重新编写一次了。

1.去官网下载opencv，在本教程中选用的时opencv3.4.1，其他版本的配置方法异曲同工。   
下载链接[http://opencv.org/releases.html](http://opencv.org/releases.html" \t "_blank)，选择sources版本

2.解压下载下来的zip包

unzip opencv-3.4.1.zip

3.进入到解压后的文件包中

4.安装依赖库和cmake ，如果提醒需要apt-get update，那就先sudo su进入root权限，再sudo apt-get update，然后在执行下面命令

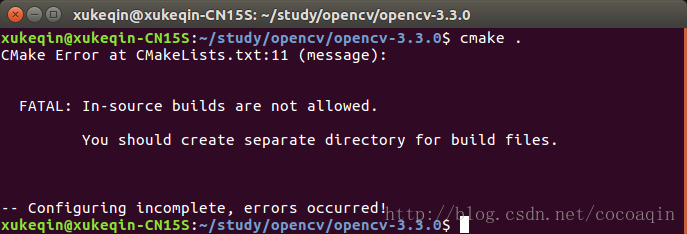
sudo apt-get install cmake

sudo apt-get install build-essential libgtk2.0-dev libavcodec-dev libavformat-dev libjpeg.dev libtiff4.dev libswscale-dev libjasper-dev

5.安装完cmake之后执行命令 ,创建编译文件夹，不创建的会提示（如下图）   
In-source builds are not allowed.

mkdir my\_build\_dir

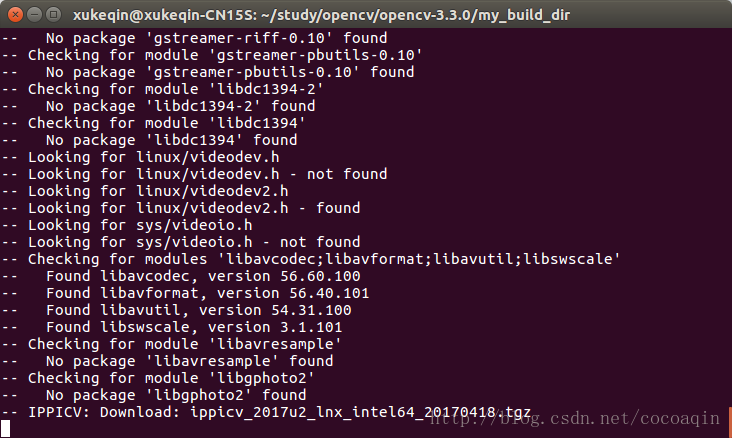
cd my\_build\_dir



6.cmake一下

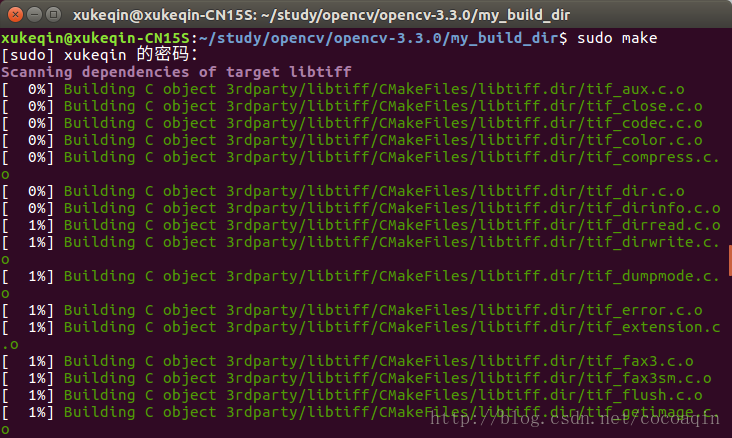
cmake -D CMAKE\_BUILD\_TYPE=Release -D CMAKE\_INSTALL\_PREFIX=/usr/local ..

注意：如果已经在新的文件夹中编译，但是还会出现之前的报错，把cmakecache.txt删了再编译就可

期间可能会下载一个东西，等待一会儿就OK   


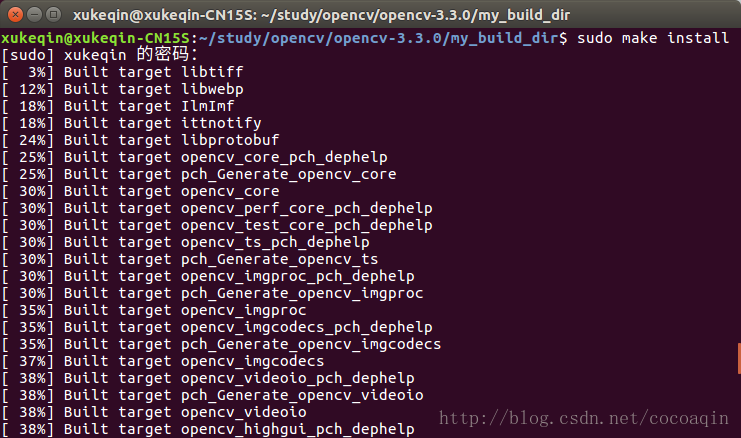
7.执行命令，漫长的编译过程

sudo make



8.执行命令

sudo make install



9.sudo make install 执行完毕后OpenCV编译过程就结束了，接下来就需要配置一些OpenCV的编译环境首先将OpenCV的库添加到路径，从而可以让系统找到

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

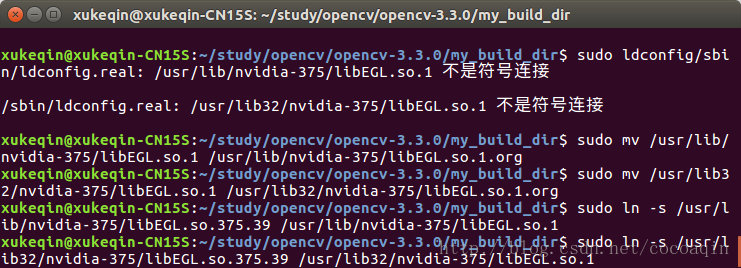
执行此命令后打开的可能是一个空白的文件，不用管，只需要在文件末尾添加

/usr/local/lib



10.执行如下命令使得刚才的配置路径生效

sudo ldconfig

这个时候可能会提示错误如下图所示（上部分）   
   
sbin/ldconfig.real: /usr/lib/nvidia-375/libEGL.so.1 不是符号连接   
/sbin/ldconfig.real: /usr/lib32/nvidia-375/libEGL.so.1 不是符号连接   
出现这个错误可能是由于系统BUG所致

只需要分别输入上面图下部分的四行命令就可解决   
sudo mv /usr/lib/nvidia-375/libEGL.so.1 /usr/lib/nvidia-375/libEGL.so.1.org   
sudo mv /usr/lib32/nvidia-375/libEGL.so.1 /usr/lib32/nvidia-375/libEGL.so.1.org   
sudo ln -s /usr/lib/nvidia-375/libEGL.so.375.39 /usr/lib/nvidia-375/libEGL.so.1   
sudo ln -s /usr/lib32/nvidia-375/libEGL.so.375.39 /usr/lib32/nvidia-375/libEGL.so.1

11.配置bash

sudo gedit /etc/bash.bashrc

在最末尾添加

PKG\_CONFIG\_PATH=$PKG\_CONFIG\_PATH:/usr/local/lib/pkgconfig

export PKG\_CONFIG\_PATH

保存，执行如下命令使得配置生效

source /etc/bash.bashrc

更新

sudo updatedb

12.至此所有的配置都已经完成   
下面用一个小程序测试一下

找到   
cd到opencv-3.4.1/samples/cpp/example\_cmake目录下   
我们可以看到这个目录里官方已经给出了一个cmake的example我们可以拿来测试下   
按顺序执行

cmake .

make

./opencv\_example

即可看到打开了摄像头，在左上角有一个hello opencv   
即表示配置成功