**【安装教程】Ubuntu16.04中用CMake-gui安装OpenCV3.2.0和OpenCV\_contrib-3.2.0（图文）**

https://blog.csdn.net/jindunwan7388/article/details/80397700

初次尝试在Ubuntu上面安装OpenCV，遇到了各种奇奇怪怪的问题，也查了好多博客和相关说明，弄了好久好久终于，完成了。我本着方便和帮助更多跟我一样刚入门OpenCV的小伙伴的初衷，写下这篇文章，希望能帮助大家，让大家少走弯路。

为了更好地展示其中的每个细节，我特意重新安装了一遍，把自己的步骤自己再走一遍保证其正确性，顺便截图给大家看。当然在写的过程中，也参考了一些博客和网址，尽可能把问题讲清楚讲明白。相关参考见文章最后。

好了话不多说，下面进入正题！

**一、简要说明**

**1.为什么要安装 OpenCV\_contrib**

 OpenCV3.0以上的版本，把一些不稳定的函数放到了第三方库OpenCV\_contrib中（比如sift,surf等），为了应用这些功能，我们需要通过CMake把OpenCV\_contrib中的功能重新加入到OpenCV3.2.0中。

关于OpenCV\_contrib，可以参见官方说明：https://github.com/opencv/opencv\_contrib/blob/master/README.md

**2. 为什么要用CMake-gui（注意，这不是必须的）**

这里我主要使用CMake-gui这个图形化界面进行安装（因为自己对ubuntu的命令不太熟悉…总是出错但又不知道怎么解决掉）。当然，也可以用终端直接输入命令操作，可以参考其他博客，这里就不再讨论。

**二、下载和添加依赖包**

1、首先更新 apt-get，在安装前最好先更新一下系统，不然有可能会安装失败。在终端输入：

1. sudo apt-get update
2. sudo apt-get upgrade

2、接着安装官方给的opencv依赖包，在终端输入：

1. sudo apt-get install build-essential
2. sudo apt-get install cmake git libgtk2.0-dev pkg-config libavcodec-dev libavformat-dev libswscale-dev
3. sudo apt-get install python-dev python-numpy libtbb2 libtbb-dev libjpeg-dev libpng-dev libtiff-dev libjasper-dev libdc1394-22-dev
4. \然后下载OpenCV3.2.0和OpenCV\_contrib-3.2.0，下载需要等待一段时间

https://github.com/Itseez/opencv/archive/3.2.0.zip

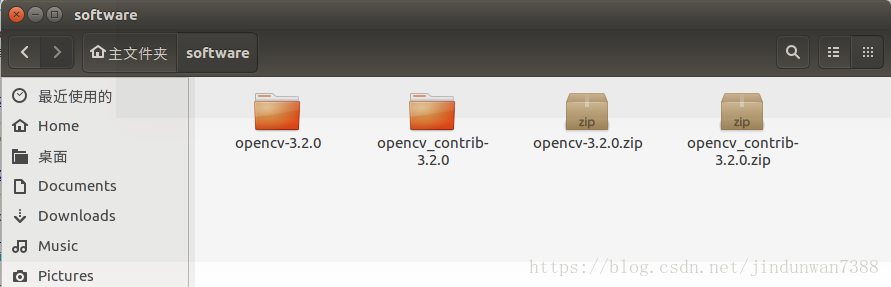
https://github.com/opencv/opencv\_contrib/archive/3.2.0.zip

3、接下来安装CMake-gui，在终端输入

sudo apt-get install cmake-qt-gui

**三、配置OpenCV**

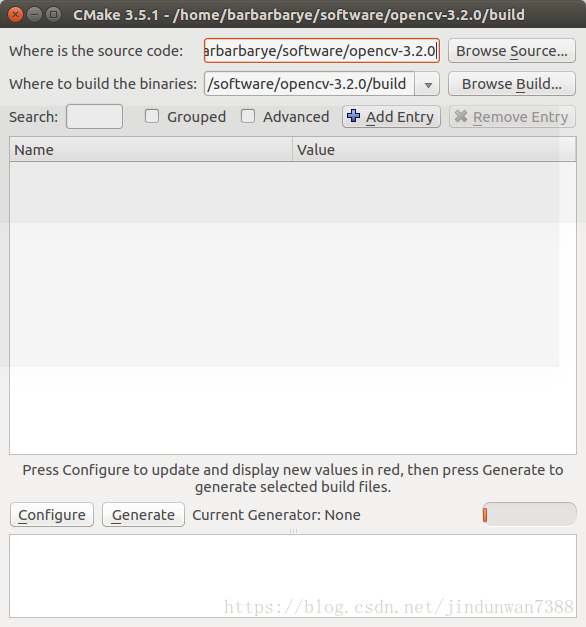
1、将OpenCV3.2.0和OpenCV\_contrib-3.2.0解压（提取），如图所示



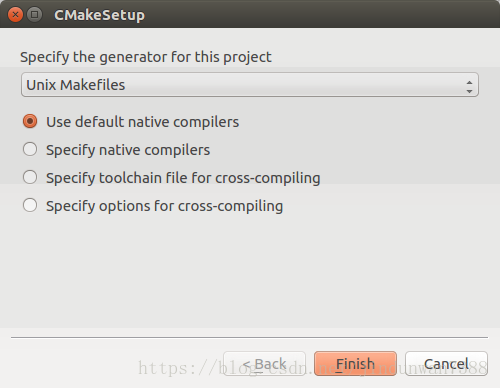
2、双击进入解压出来的OpenCV3.2.0文件夹，右键打开终端（或者在别处打开终端，通过输入cd opencv3.2.0进入当前目录下），然后依次输入（不要忘了第三行的最后的空格和两个点）：

1. mkdir build
2. cd build
3. cmake-gui ..

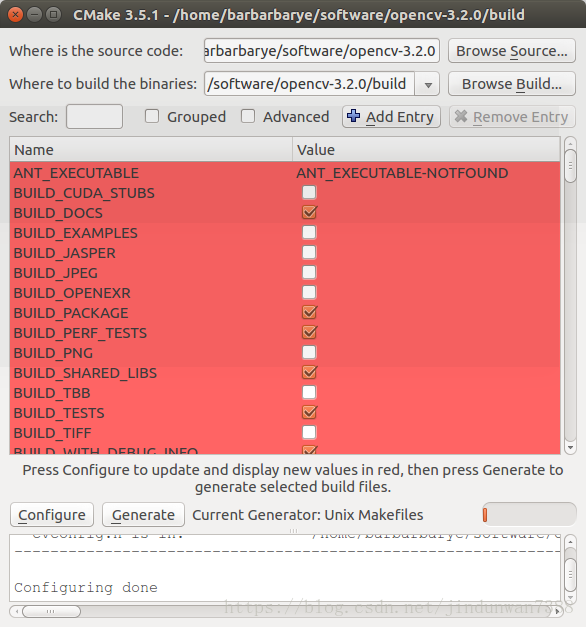
3、然后会弹出CMake的图形化界面，在上方的两个路径里面，选择好代码所在文件夹的路径和要安装的路径



4、点击左下方的Configure按钮，选择Unix Makefiles，选择Use default native compilers（默认），然后点击Finish



5、需要下载一些文件，要等待一段时间。然后CMake即载入默认配置，如下图所示：

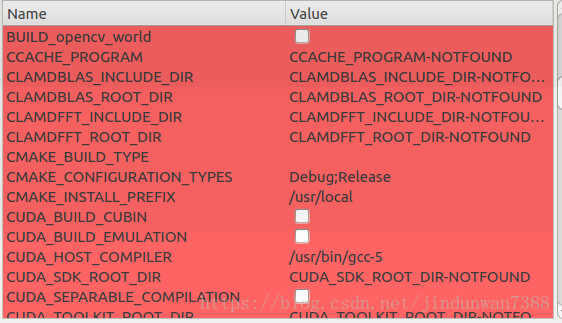


窗口的中间部分即配置列表，这里和使用cmake命令直接生成makefile文件一致的。正如上文所述，这里通过图形界面的方式来进行配置，更加直观方便。

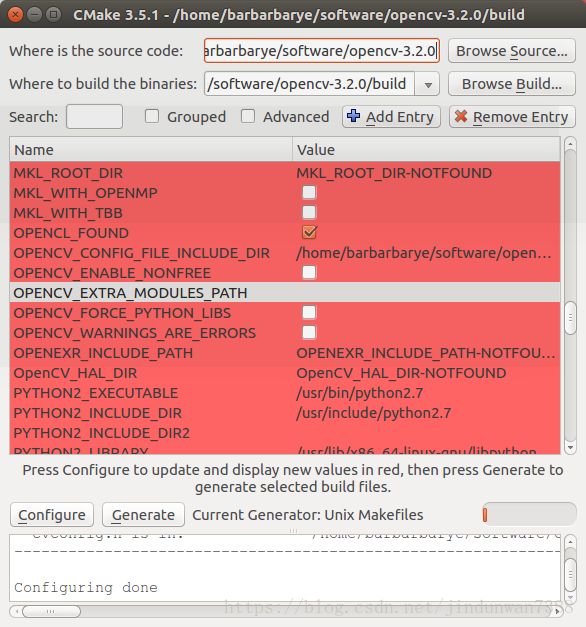
6、这里需要对两个地方进行修改：

（1）在CMAKE\_BUILD\_TYPE 值处输入RELEASE，其他保持不变（如果已经存在就不必修改）。

下方的CMAKE\_INSTALL\_PREFIX显示了默认的安装目录，生成makefile文件最后执行 make install时就会安装到这个目录，这里可以根据个人需求更改。我这里就不做修改了。

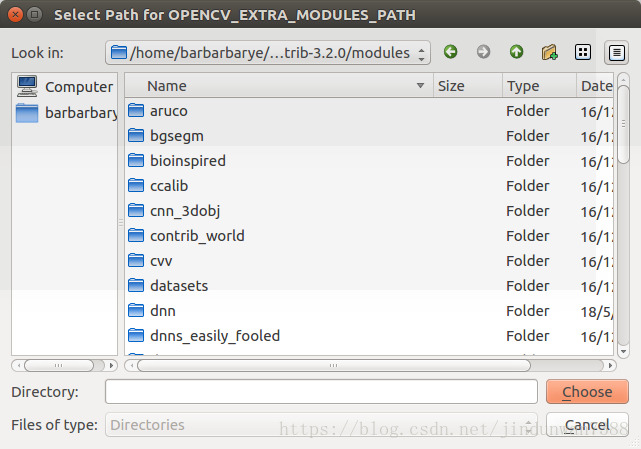


（2）在OPENCV\_EXTRA\_MODULES\_PATH处，选择输入目录（单击这一行后方空白处即可选中）



然后选择opencv\_contrib-3.2.0文件夹中的modules文件夹，注意，不是只选中opencv\_contrib-3.2.0文件夹就好了，需要选中里面的modules文件夹！    如下图所示，然后点Choose。

注：OPENCV\_EXTRA\_MODULES\_PATH 是用来指定要编译的扩展模块，其中包括OpenCV\_contrib模块。

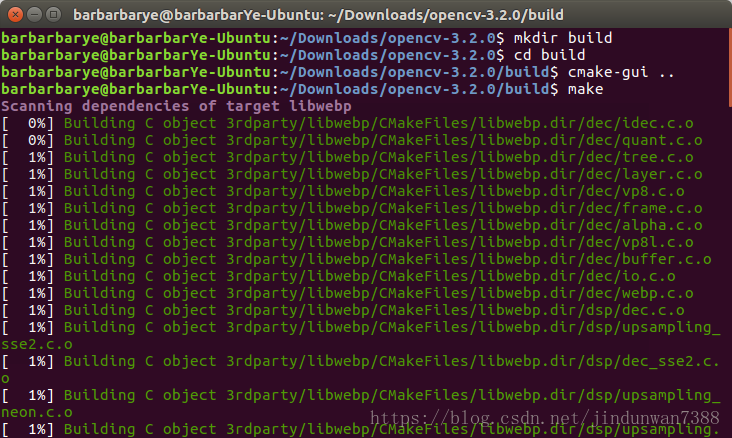


7、点击Generate生成配置文件，这一步应该比较快就完成了。

8、接着，在build目录下打开终端，输入

1. make
2. sudo make install

9、然后就开始安装了，这一步又需要等比较长的时间了……不要急，进行到这一步说明很快就能安装成功了～



当你执行完上面两行命令的时候，并看到100%的时候，恭喜你！安装成功！

**四、配置环境变量**

1、安装成功后还需要设置opencv的环境变量。打开文件：

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

2、将以下内容添加到最后：

/usr/local/lib

3、接下来配置库：

sudo ldconfig

4、更改环境变量：

sudo gedit /etc/bash.bashrc

5、在文件后添加：

1. PKG\_CONFIG\_PATH=$PKG\_CONFIG\_PATH:/usr/local/lib/pkgconfig
2. export PKG\_CONFIG\_PATH

6、保存退出，在运行下面的例程之前，需要重新开启终端来使配置生效。

到此，安装和配置的整个过程都完成了！

**五、测试**

1、在终端运行命令：pkg-config --cflags --libs opencv 出现下面信息：

1. licaibiao@ubuntu:~/test/OpenCV$ pkg-config --cflags --libs opencv
2. -I/usr/local/include/opencv -I/usr/local/include -L/usr/local/lib -lopencv\_shape -lopencv\_stitching -lopencv\_objdetect -lopencv\_superres -lopencv\_videostab -lopencv\_calib3d -lopencv\_features2d -lopencv\_highgui -lopencv\_videoio -lopencv\_imgcodecs -lopencv\_video -lopencv\_photo -lopencv\_ml -lopencv\_imgproc -lopencv\_flann -lopencv\_core
3. licaibiao@ubuntu:~/test/OpenCV$

2、运行一个打开图片的程序。

（1）在任意位置，新建一个文件夹test。

（2）创建DisplayImage.cpp文件，并输入以下代码，保存。

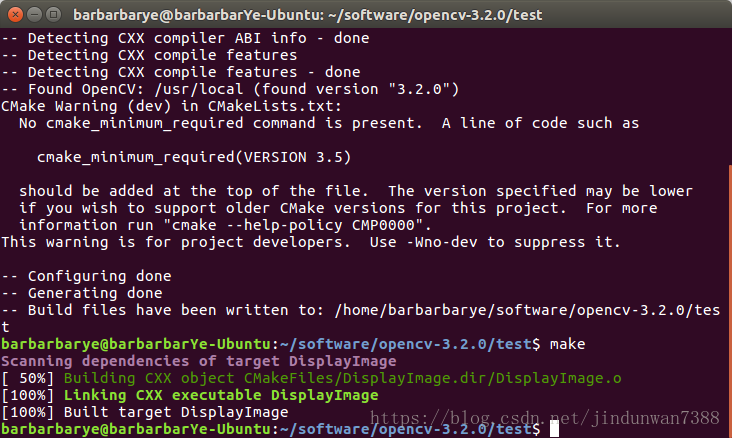
1. #include <cv.h>
2. #include <highgui.h>
3. #include <opencv2/core/core.hpp>
4. #include <opencv2/highgui/highgui.hpp>
6. using namespace cv;
7. int main( int argc, char\*\* argv )
8. {
9. Mat image;
10. image = imread( argv[1], 1 );
12. if( argc != 2 || !image.data )
13. {
14. printf( "No image data \n" );
15. return -1;
16. }
18. namedWindow( "Display Image", CV\_WINDOW\_AUTOSIZE );
19. imshow( "Display Image", image );
20. waitKey(0);
22. return 0;
23. }

（3）用CMake编译：添加一个CMakeLists.txt 文件，内容如下，保存

1. project( DisplayImage )
2. find\_package( OpenCV REQUIRED )
3. add\_executable( DisplayImage DisplayImage )
4. target\_link\_libraries( DisplayImage ${OpenCV\_LIBS} )

（4）在test文件夹中右键，打开终端，执行以下命令，生成执行文件 DisplayImage。（这里cmake后面一个点就够了，表示在当前目录下执行cmake操作）

1. cmake .
2. make

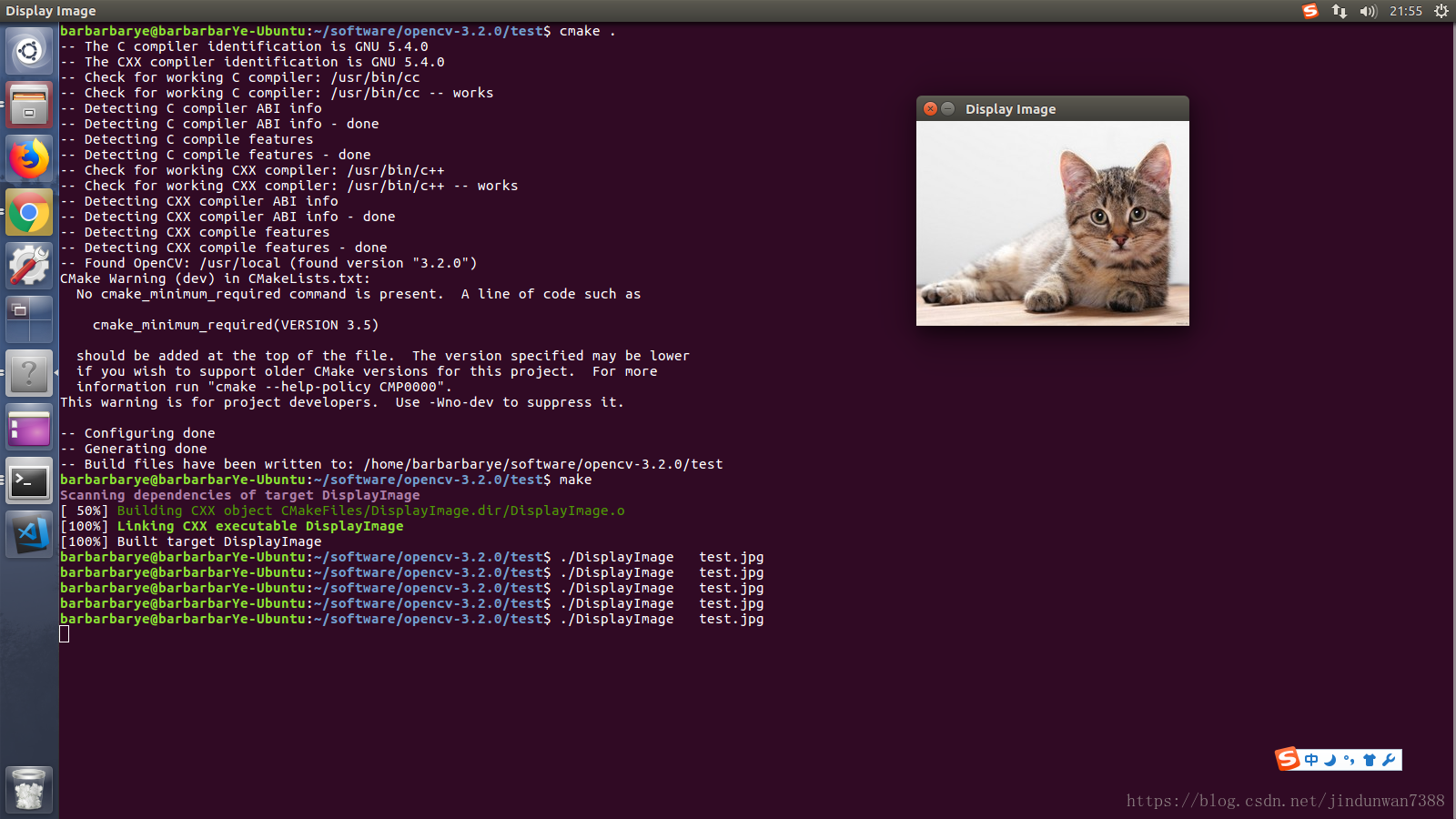


（5）选择一张图片，复制到test文件夹中，并且重命名为test.jpg

（6）在终端运行以下代码

./DisplayImage test.jpg

如果以上步骤都安装和配置成功，可以显示出图片：



大功告成！

希望大家都所有收获，有所成就！

参考其他相关博客：

https://blog.csdn.net/honyniu/article/details/46390097

https://www.cnblogs.com/asmer-stone/p/5089764.html

https://blog.csdn.net/u014696921/article/details/62884625

**个人心得**

最近刚刚接触人脸识别的具体技术，看到网上这么多这么多文章介绍人脸识别具体操作和前景，大受鼓舞，感觉自己前景一片光明啊～感觉一下子就能做成这么前沿的技术～仿佛一小子就能达到小目标走上人生巅峰～哈哈哈哈哈～满心欢喜地准备大干一场。结果，安装OpenCV的过程就给了我下马威……安装了好半天，才在Win10上面弄好OpenCV2.4，中间遇到了各种奇葩的错误，不过总算解决掉了。后来为了后续更好地操作，在ubuntu上也想装上一个，以为之前有经验了对吧，应该不久就能装好，结果，花了我更长的时间去弄安装，哇真的欲哭无泪，就算我真的是个ubuntu的菜鸟辣鸡，接触ubuntu不多，但也不要这样欺负我啊……心态真的要崩。最后终于弄成功了，也算是给自己的一点安慰，抚平心中的愤懑和暴躁，聊以自慰。

以上就是在ubuntu16.04中用CMake-gui安装OpenCV3.2.0和OpenCV\_contrib-3.2.0的所有内容，断断续续花了两天时间整合完成。看到自己终于写好了第一篇技术性博客，还是非常欣慰滴～

文章中难免有些疏漏和错误，还恳请各位读者指出～当然也欢迎大家与我交流讨论。