

南工骁鹰战队OpenCV编译安装指南

OpenCV的重要性不言而喻，是整个视觉程序的基础，本教程根据战队所需从零开始安装opencv和opencv_contrib。

安装编译环境 **Ubuntu 20.04** + **CMake 3.16.3** + **OpenCV 4.5.4**。

注意，请在**完全翻墙环境**(如战队内网环境)下使用本教程。

一、基础编译工具安装

```
sudo apt install cmake-qt-gui git vim build-essential pkg-config
```

二、OpenCV依赖安装

OpenCV重要依赖（必须安装）

```
sudo apt install python3-dev python3-numpy libtbb2 libtbb-dev libjpeg-dev  
sudo apt install libpng-dev libtiff-dev libdc1394-22-dev  
sudo apt install libeigen3-dev libgtk2.0-dev  
sudo apt install libavcodec-dev libavformat-dev libswscale-dev
```

OpenCV可选依赖（安装可以解决编译时的问题，优化性能等）

```
sudo apt install ccache  
sudo apt install libgstreamer1.0-dev libgstreamer-plugins-base1.0-dev  
sudo apt install libgtk-3-dev  
sudo apt install libavresample-dev  
sudo apt install libgphoto2-dev  
sudo apt install libopenblas-dev  
sudo apt install doxygen  
sudo apt install libhdf5-dev  
sudo apt install libgoogle-glog-dev  
sudo apt install libgflags-dev
```

三、OpenCV的下载

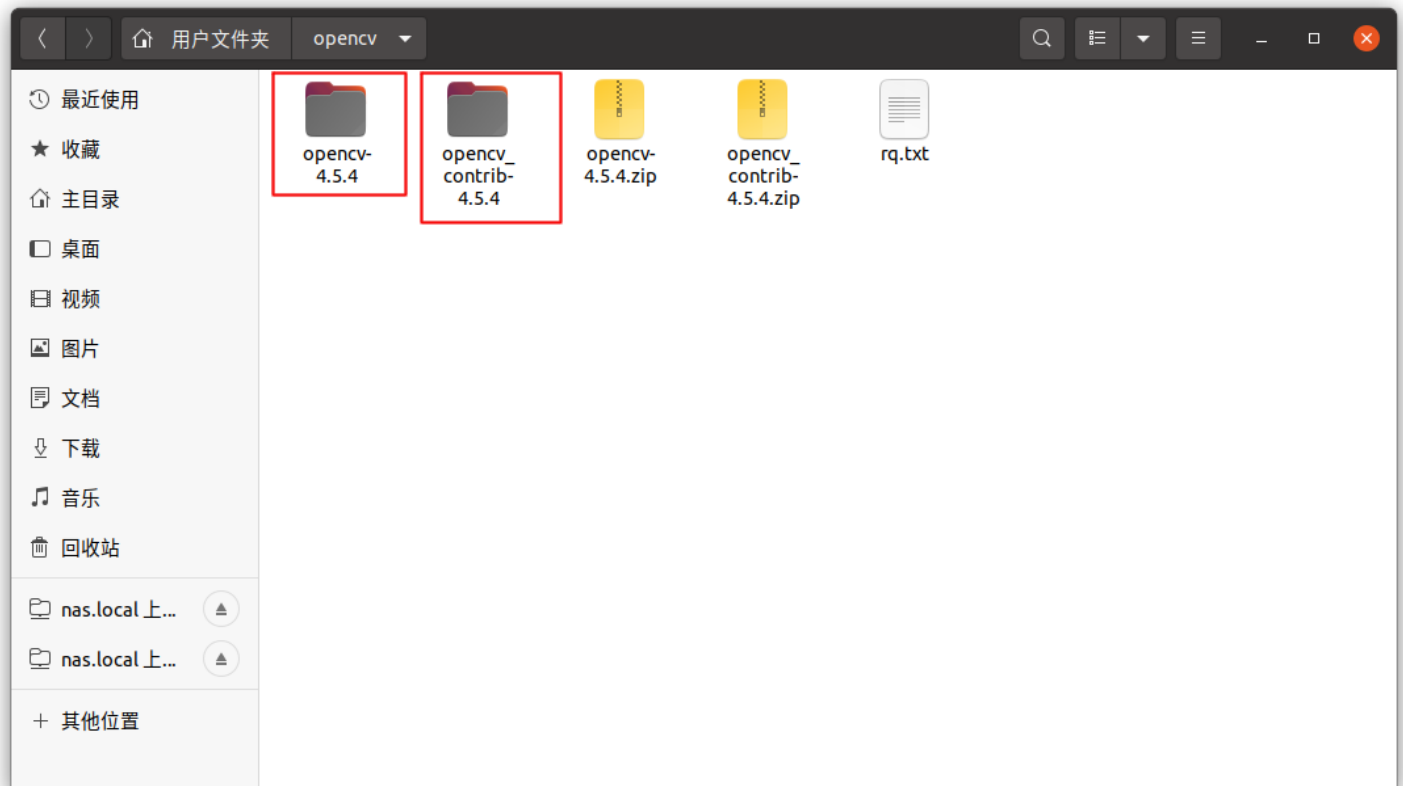
- 使用终端wget下载

在合适的目录下(如:~/opencv)打开终端，键入:

```
wget https://github.com/opencv/opencv/archive/refs/tags/4.5.4.zip -O opencv-4.5.4.zip
wget https://github.com/opencv/opencv_contrib/archive/refs/tags/4.5.4.zip -O
opencv_contrib-4.5.4.zip
```

- 在github网页上自行下载
略

下载完成后，在当前目录下解压，分别得到两个目录，文件结构如下图:



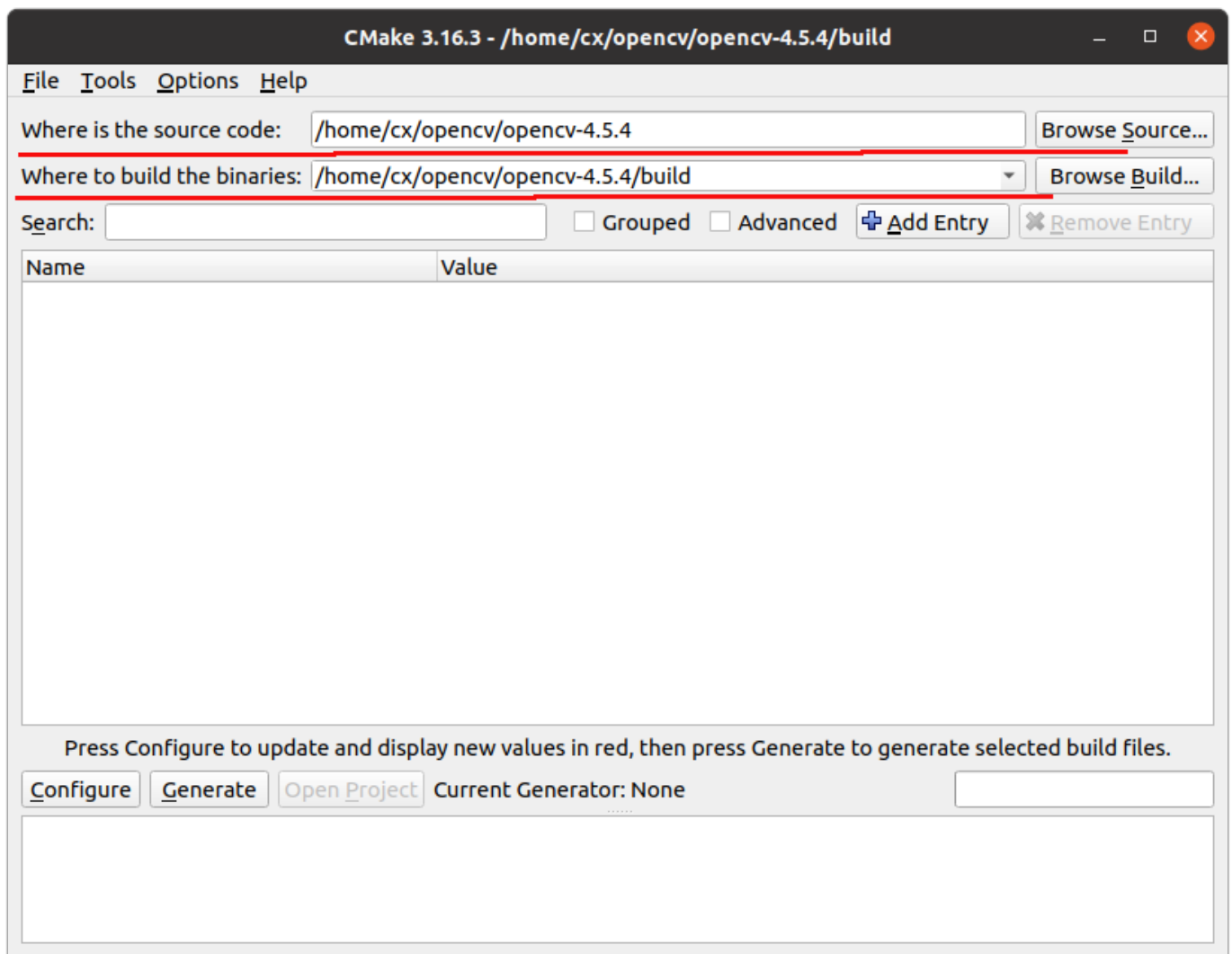
四、OpenCV的编译和安装

OpenCV编译选项基本配置和验证

- 使用终端在当前目录下打开，键入:

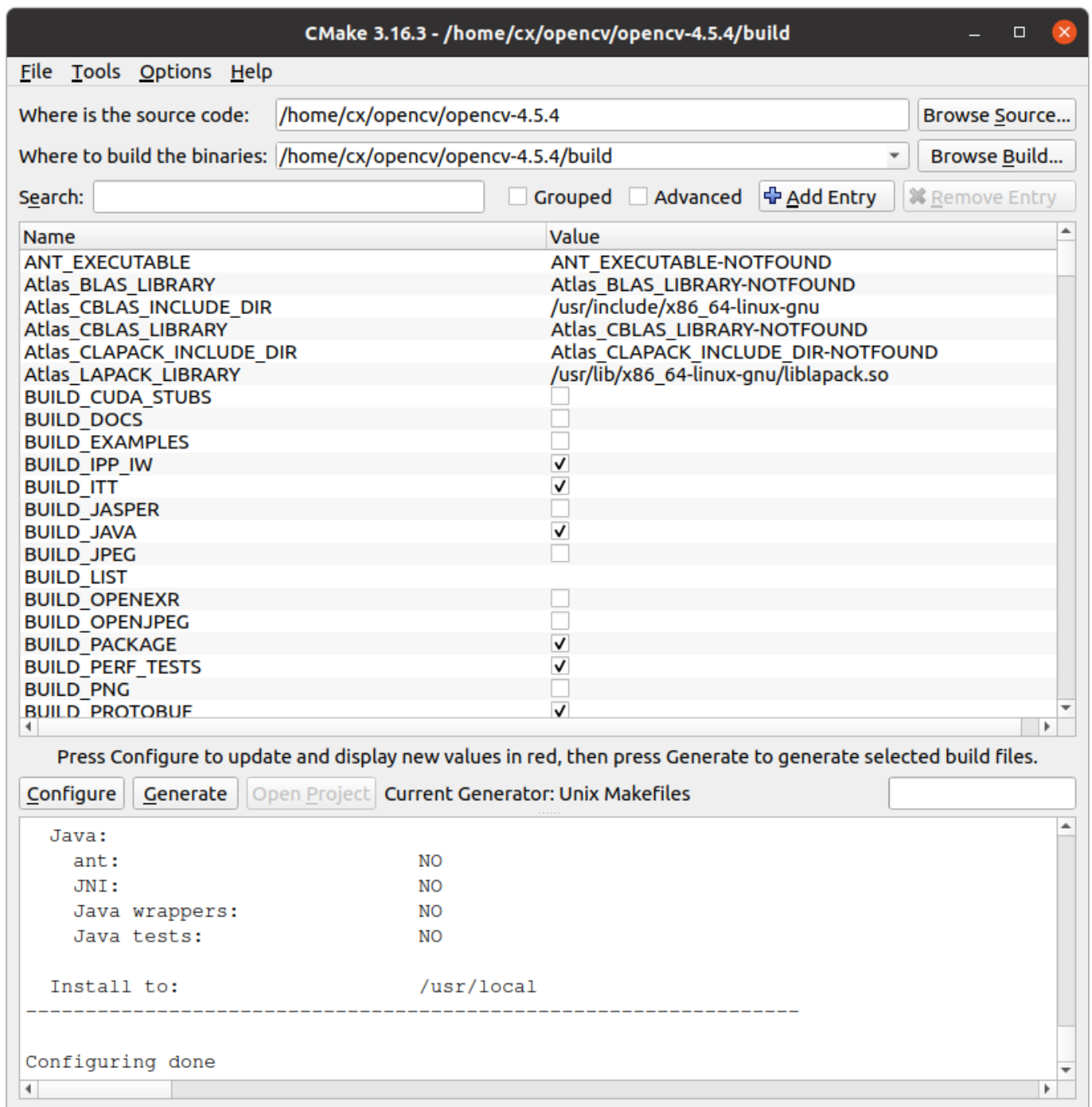
```
cmake-gui
```

- 出现如下画面，其中source code一栏填写opencv-4.5.4文件夹所在地址，而下一行在其路径后加/build就可以了。



点击configure，cmake会开始进行检查和下载必要的文件，在这个过程中可能出错，如果是在全局翻墙环境下，只需要重新清除dns缓存（百度搜索ubuntu清除dns缓存）后重新configure即可；如果是在非全局翻墙环境下，需要手动下载并更改cmake下载地址，在此不再赘述。

- configure完成后，如下图所示：



- 检查底部输出信息，如果之前的可选依赖没有安装的话，会有很多missing/not found的提示，如下图所示：

File Tools Options Help

Where is the source code: /home/cx/opencv/opencv-4.5.4

Browse Source...

Where to build the binaries: /home/cx/opencv/opencv-4.5.4/build

Browse Build...

Search:

☐ Grouped☐ Advanced

+ Add Entry

✖ Remove Entry

Name	Value
BUILD_IPP_IW	<input checked="" type="checkbox"/>
BUILD_ITT	<input checked="" type="checkbox"/>
BUILD_JASPER	<input type="checkbox"/>
BUILD_JAVA	<input checked="" type="checkbox"/>
BUILD_JPEG	<input type="checkbox"/>
BUILD_LIST	

Press Configure to update and display new values in red, then press Generate to generate selected build files.

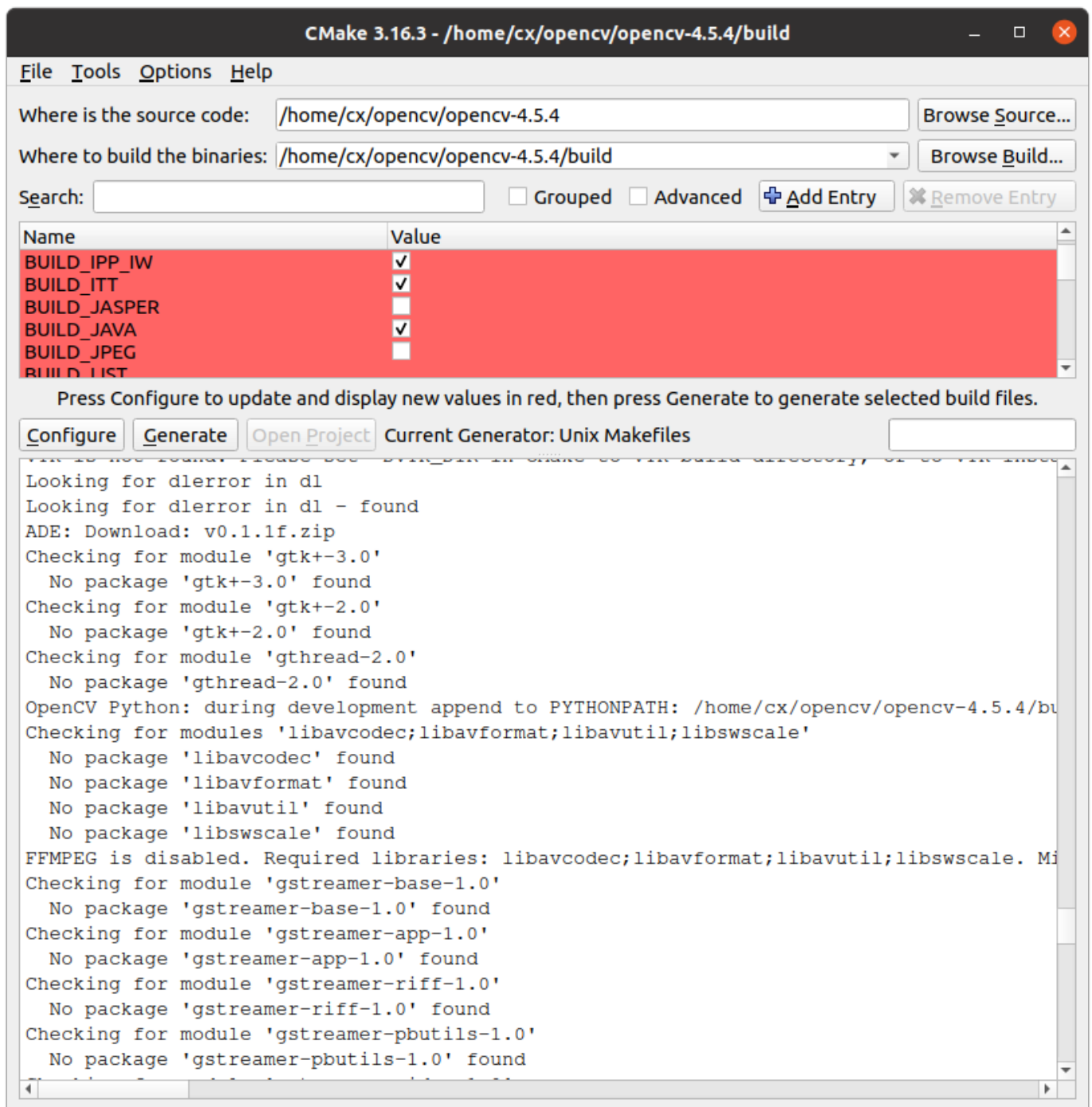
Configure

Generate

Open Project

Current Generator: Unix Makefiles

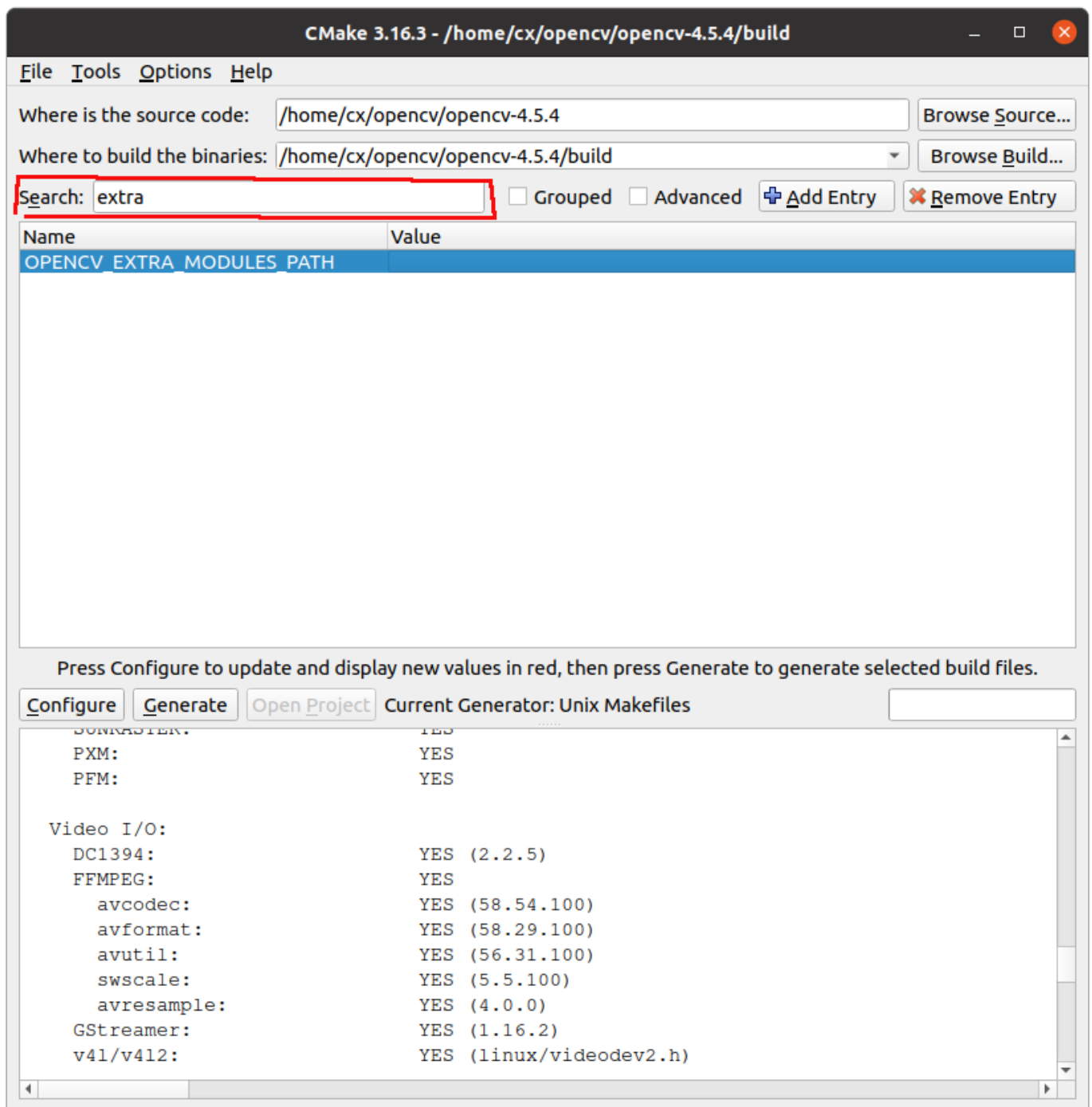
```
Detecting CXX compile features
Detecting CXX compile features - done
Check for working C compiler: /usr/bin/cc
Check for working C compiler: /usr/bin/cc -- works
Detecting C compiler ABI info
Detecting C compiler ABI info - done
Detecting C compile features
Detecting C compile features - done
Detected processor: x86_64
Could NOT find PythonInterp (missing: PYTHON_EXECUTABLE) (Required is at least version
Found PythonInterp: /usr/bin/python3 (found suitable version "3.8.10", minimum required
Found PythonLibs: /usr/lib/x86_64-linux-gnu/libpython3.8.so (found suitable exact versi
Looking for ccache - not found
Performing Test HAVE_CXX_FSIGNED_CHAR
Performing Test HAVE_CXX_FSIGNED_CHAR - Success
Performing Test HAVE_C_FSIGNED_CHAR
Performing Test HAVE_C_FSIGNED_CHAR - Success
Performing Test HAVE_CXX_W
Performing Test HAVE_CXX_W - Success
Performing Test HAVE_C_W
Performing Test HAVE_C_W - Success
Performing Test HAVE_CXX_WALL
Performing Test HAVE_CXX_WALL - Success
Performing Test HAVE_C_WALL
Performing Test HAVE_C_WALL - Success
```

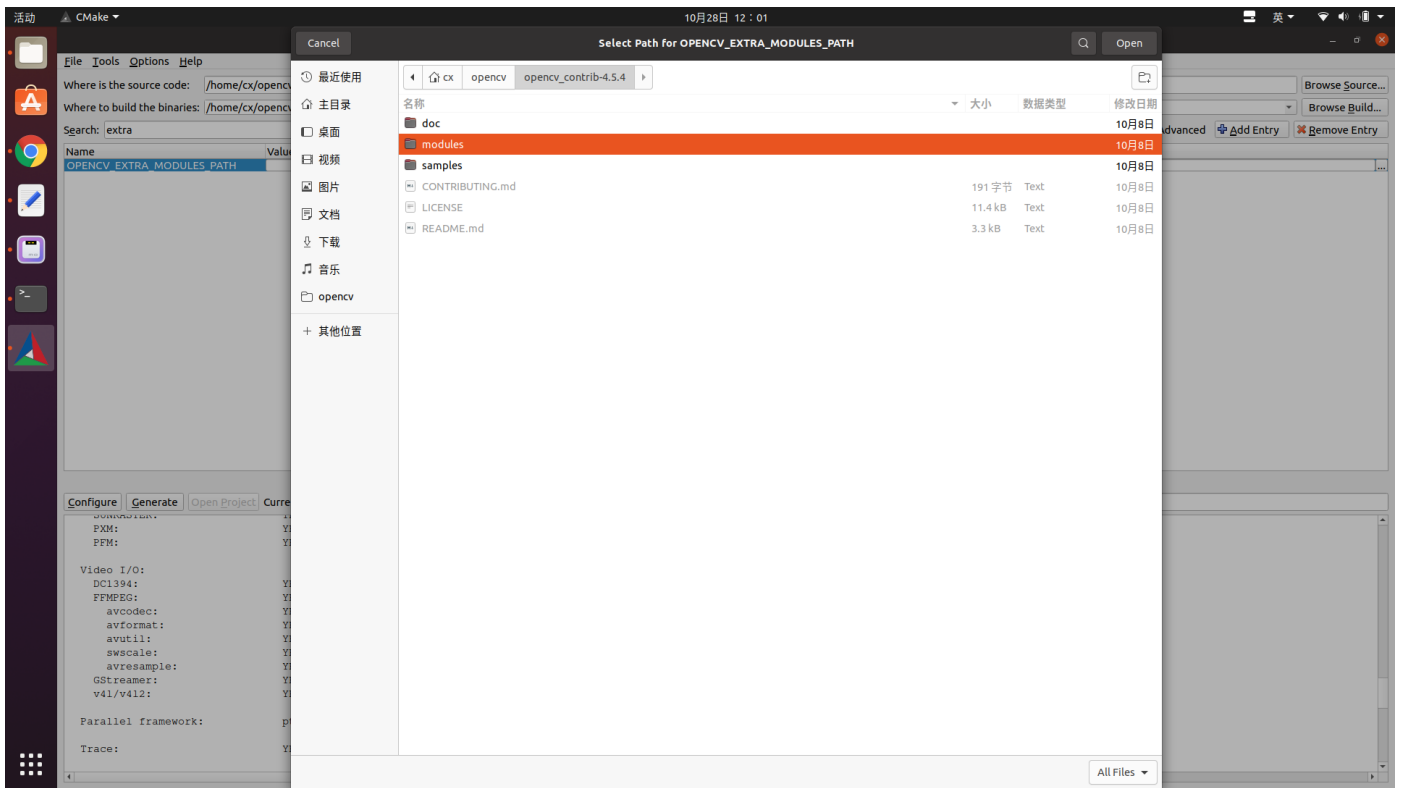


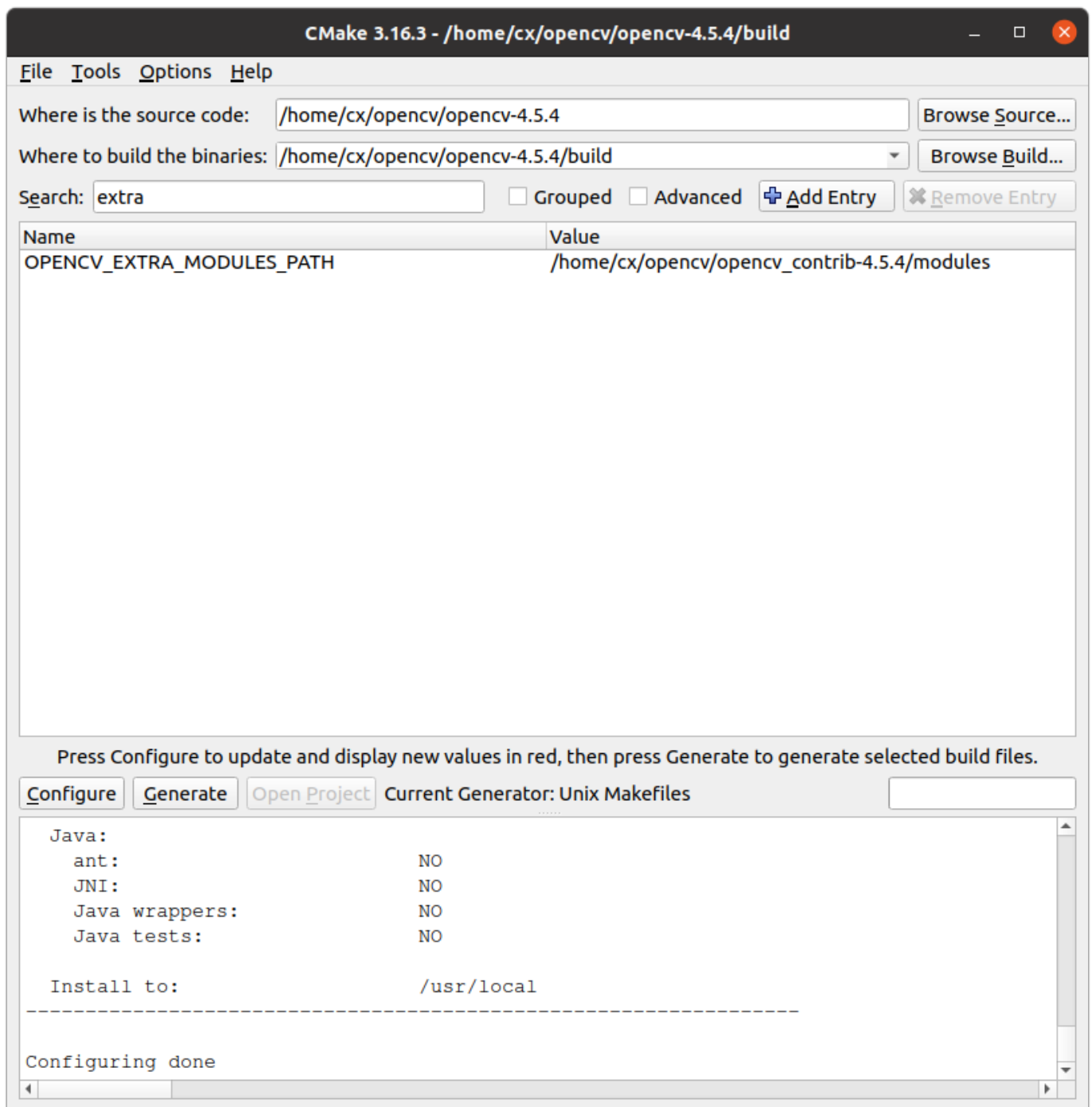
因此，建议安装所有的可选依赖。

OpenCV编译选项配置

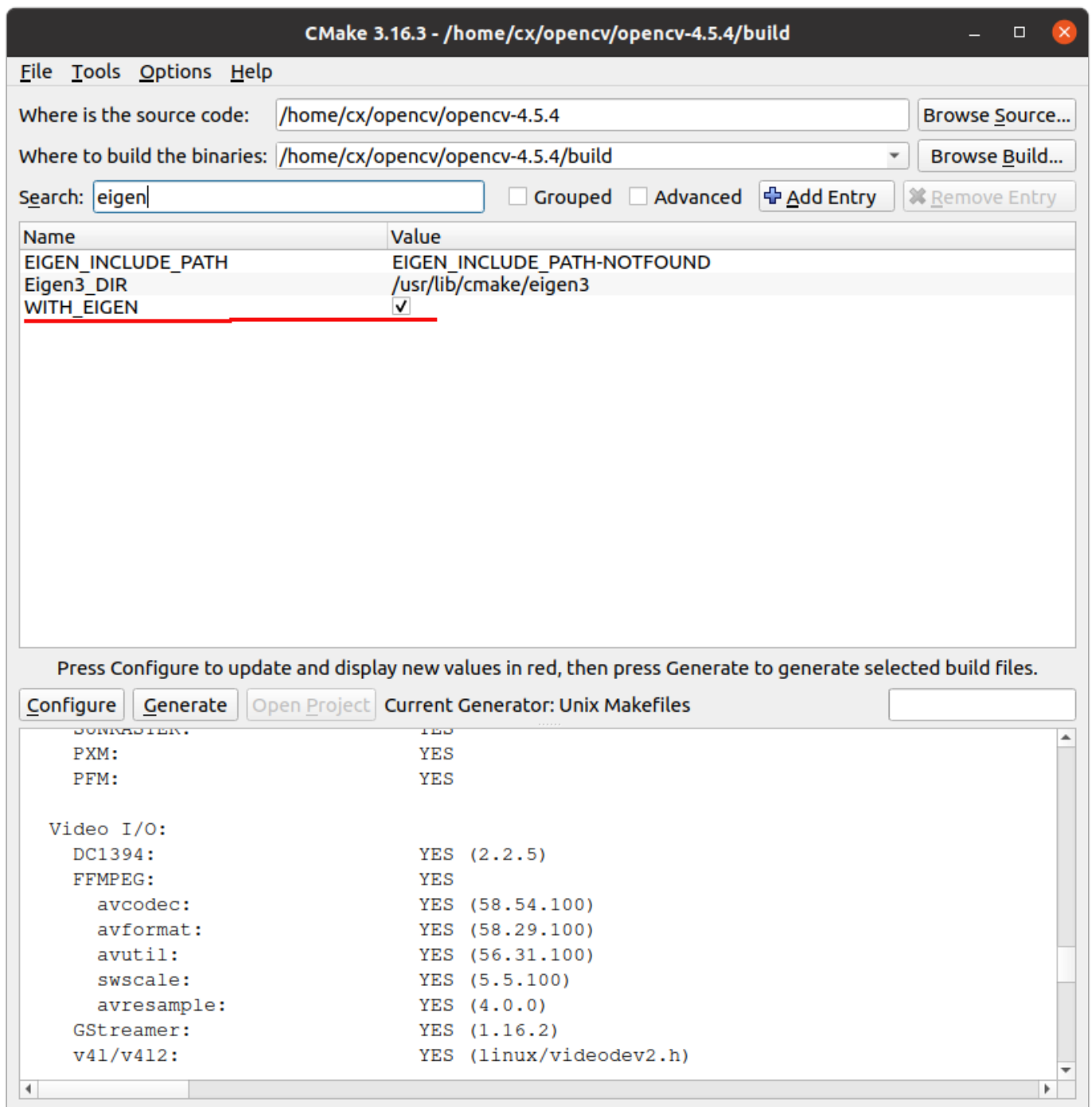
- 配置opencv_contrib，在search栏中搜索extra，将值更改为opencv_contrib-4.5.4文件夹中modules子文件夹的地址，如下图所示：



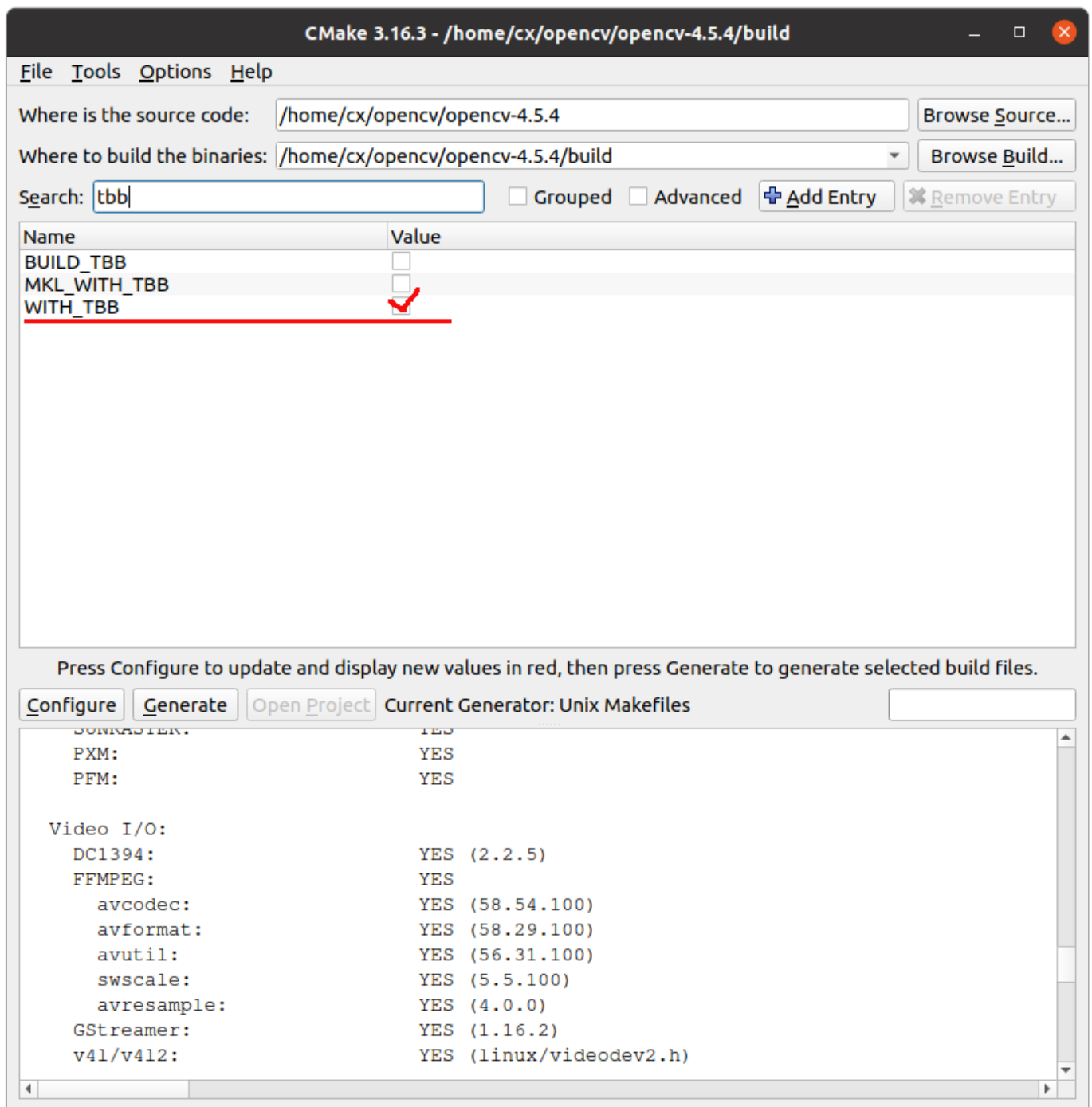




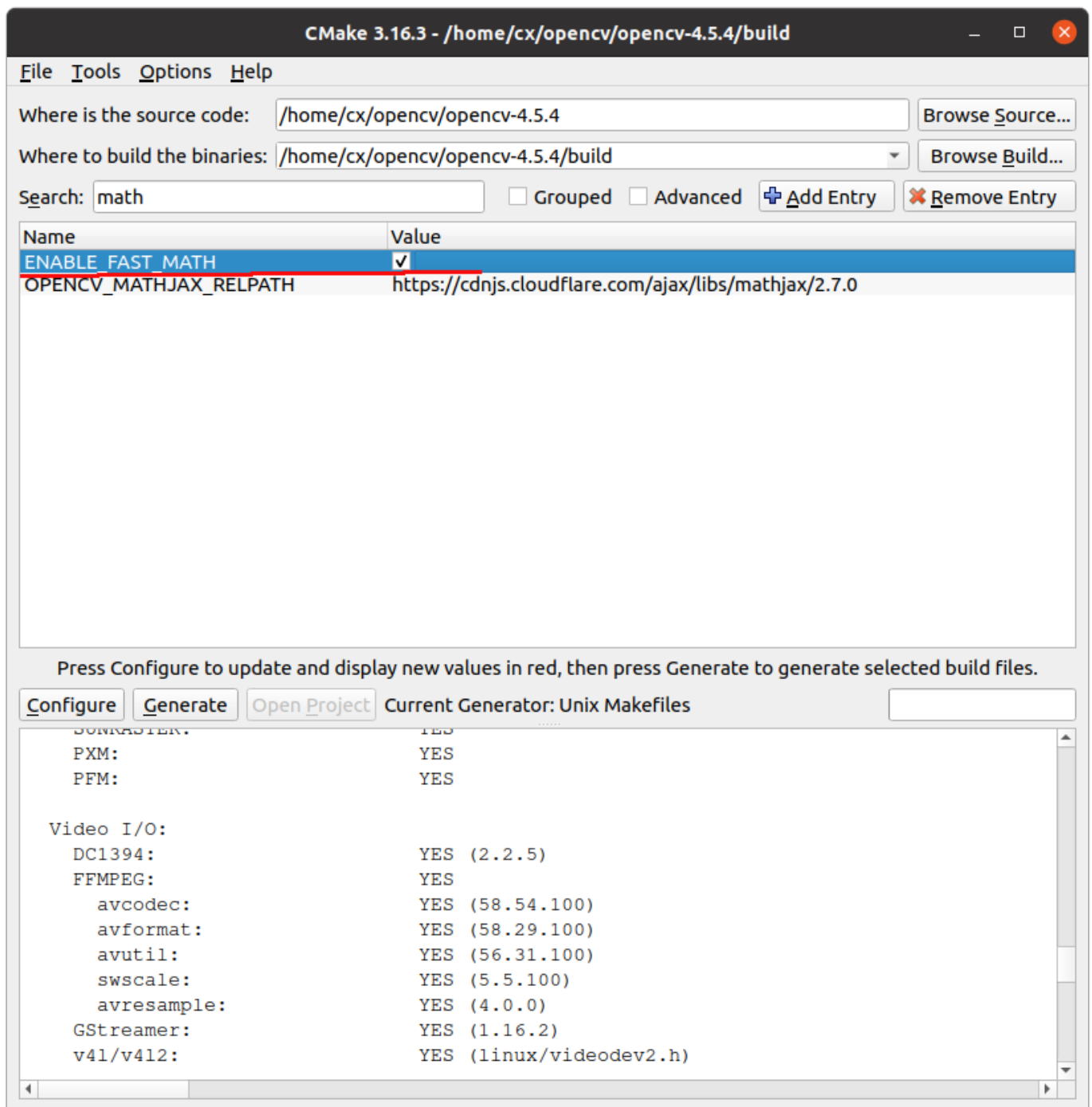
- 配置启用eigen3，在搜索框中搜索eigen，如下图所示勾选选项:



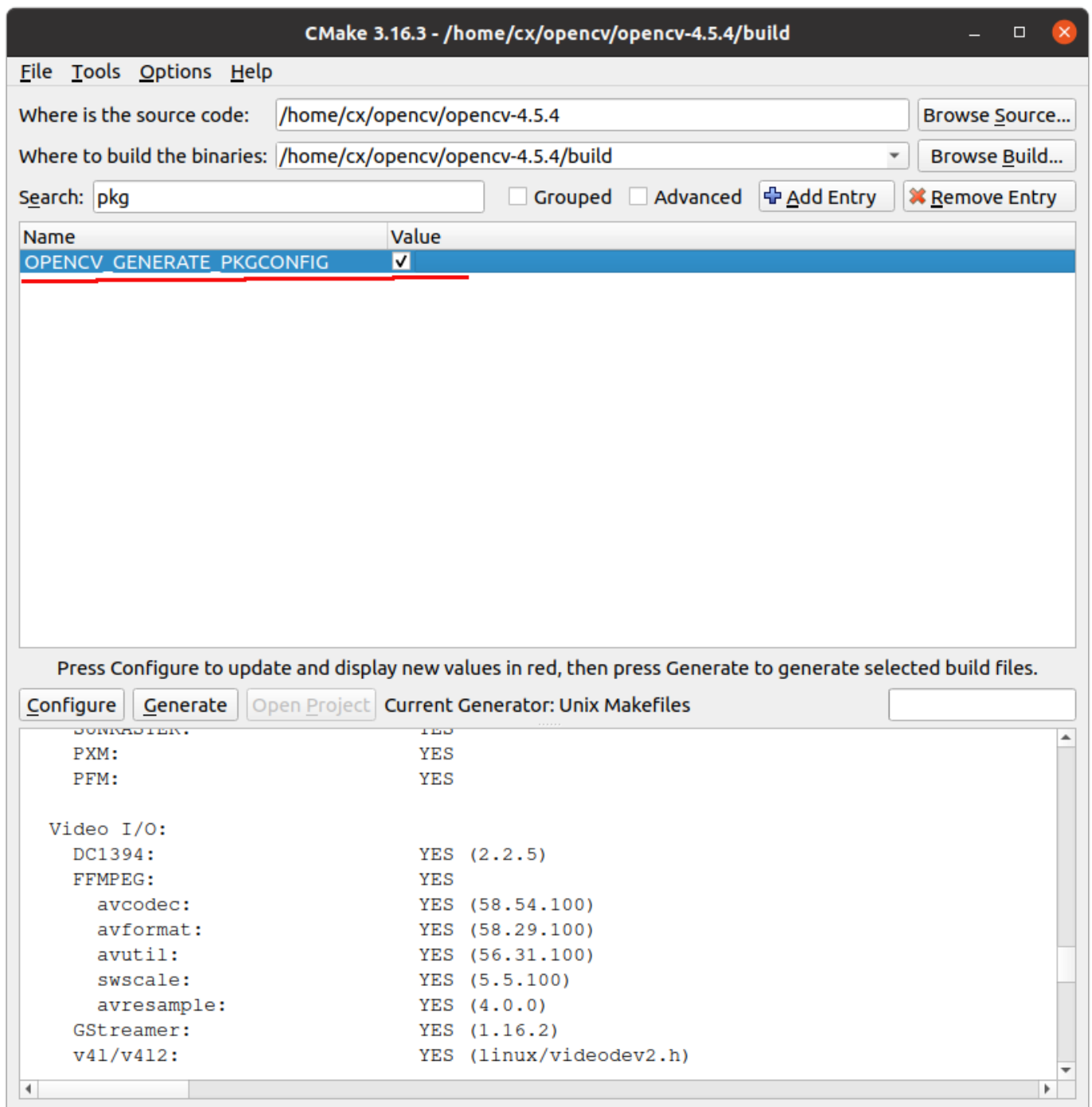
- 配置启用tbb，在搜索框中搜索tbb，如下图所示勾选选项:



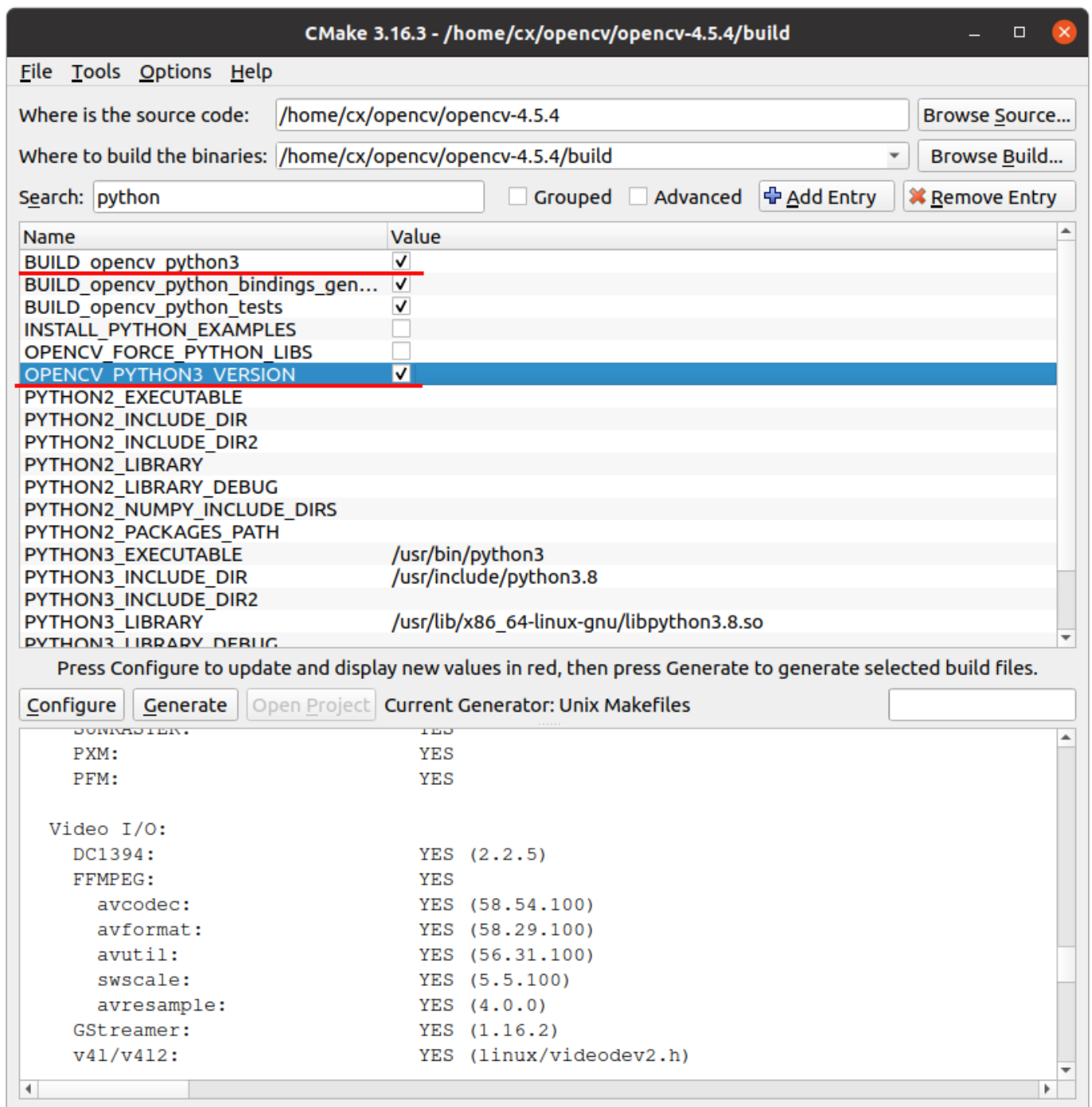
- 配置启用fast math，在搜索框中搜索math，如下图所示勾选选项:



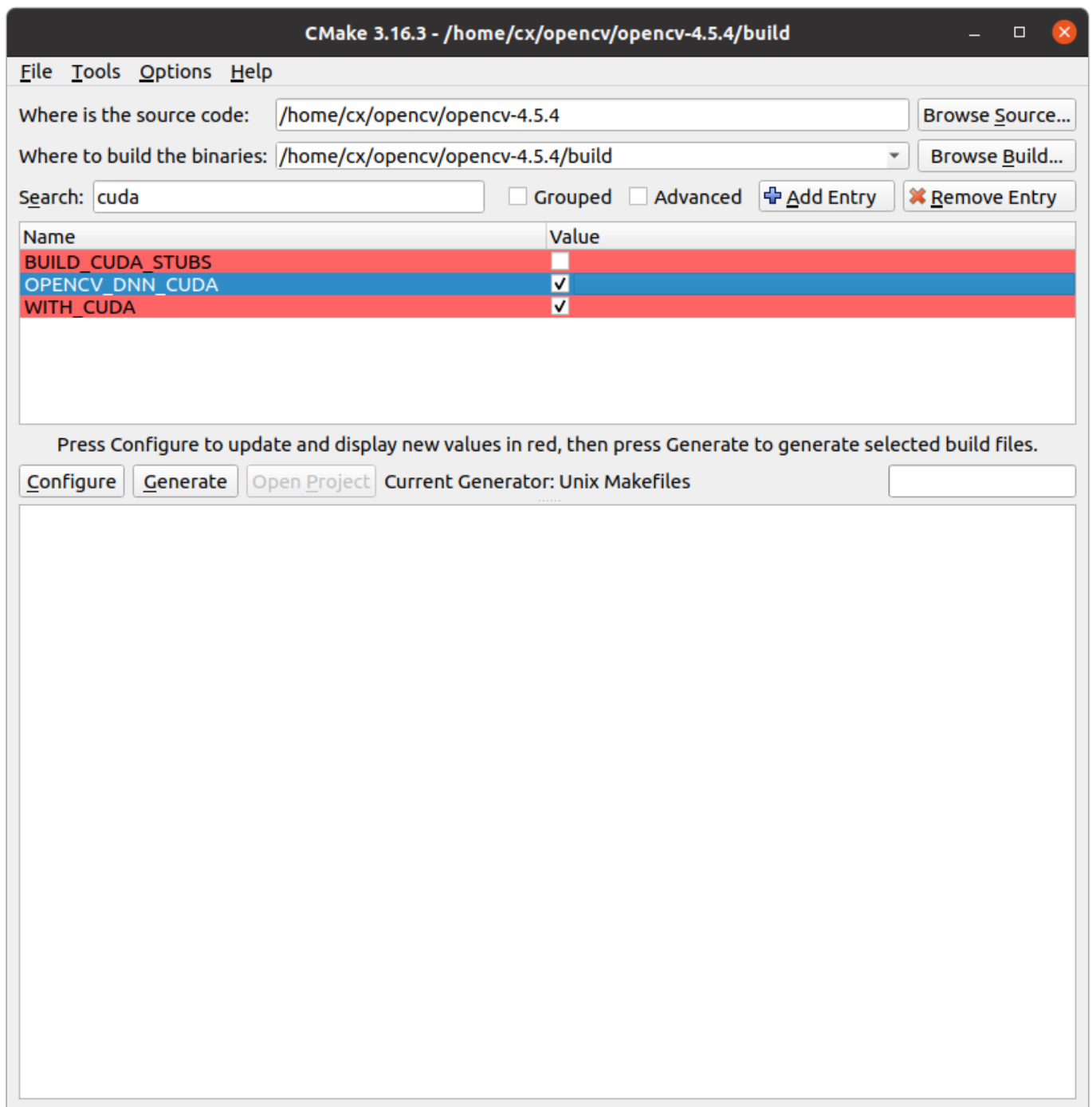
- 配置自动生成pkg-config文件，在搜索框中搜索pkg-config，如下图所示勾选选项:



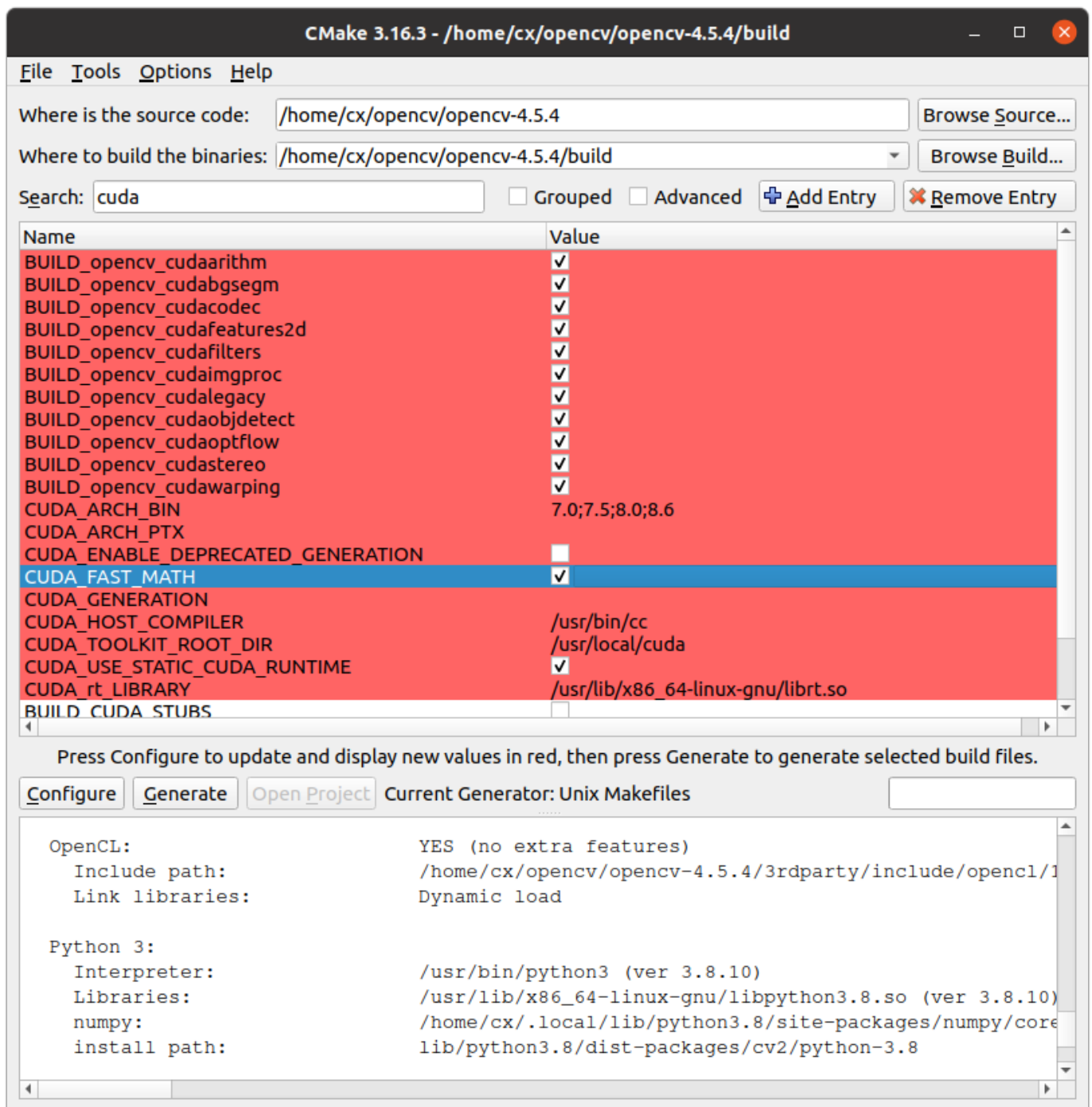
- （可选）配置生成Python模块，启用Python3的OpenCV支持，在搜索框中搜索python，如下图所示勾选选项：



- （可选）配置CUDA支持（N卡专用，并请提前安装好Cuda套件）
在搜索框中搜索cuda，如下图所示勾选选项：

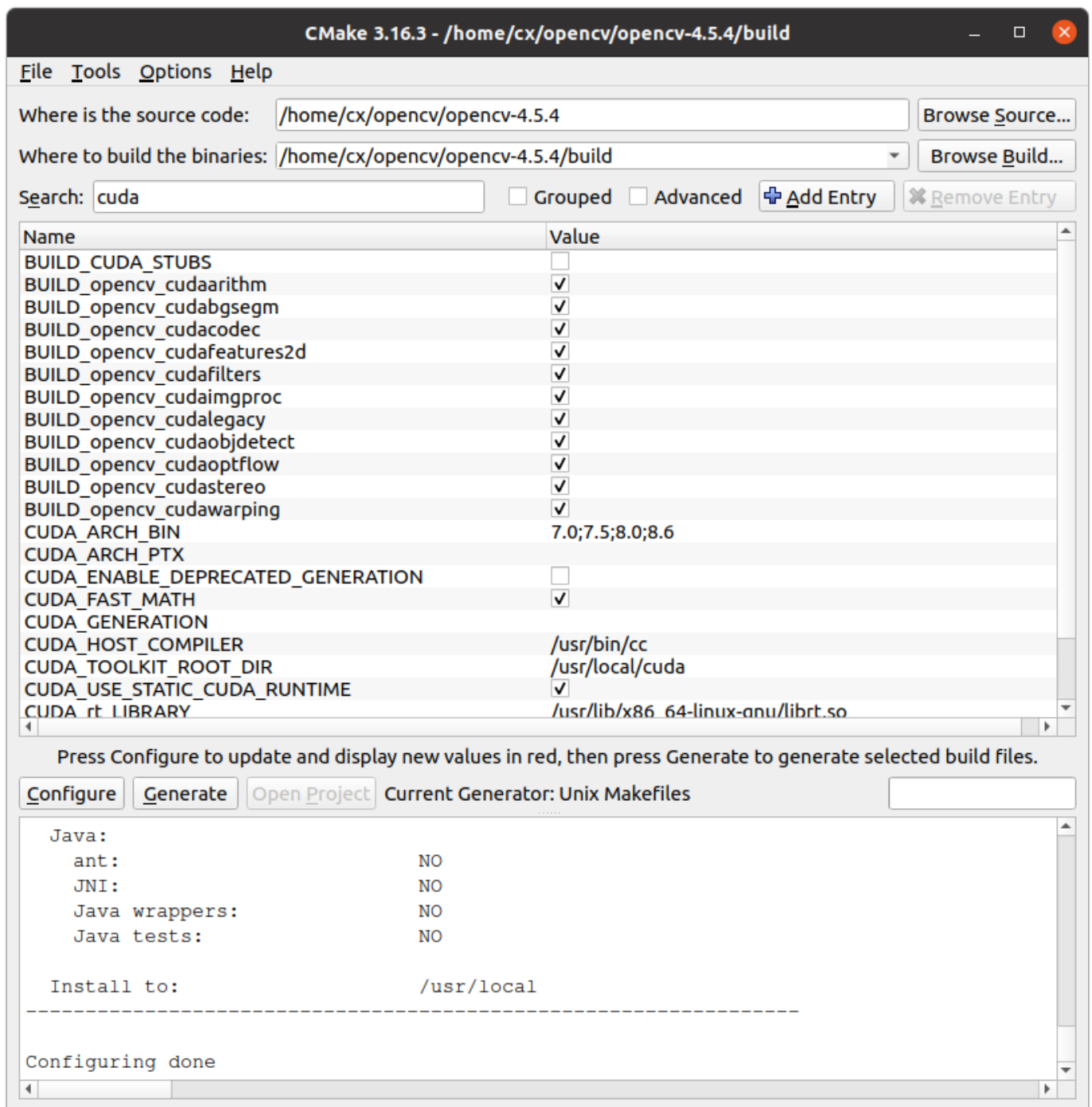


点击一次configure，此时有更多的选项出现，如下图所示勾选选项:

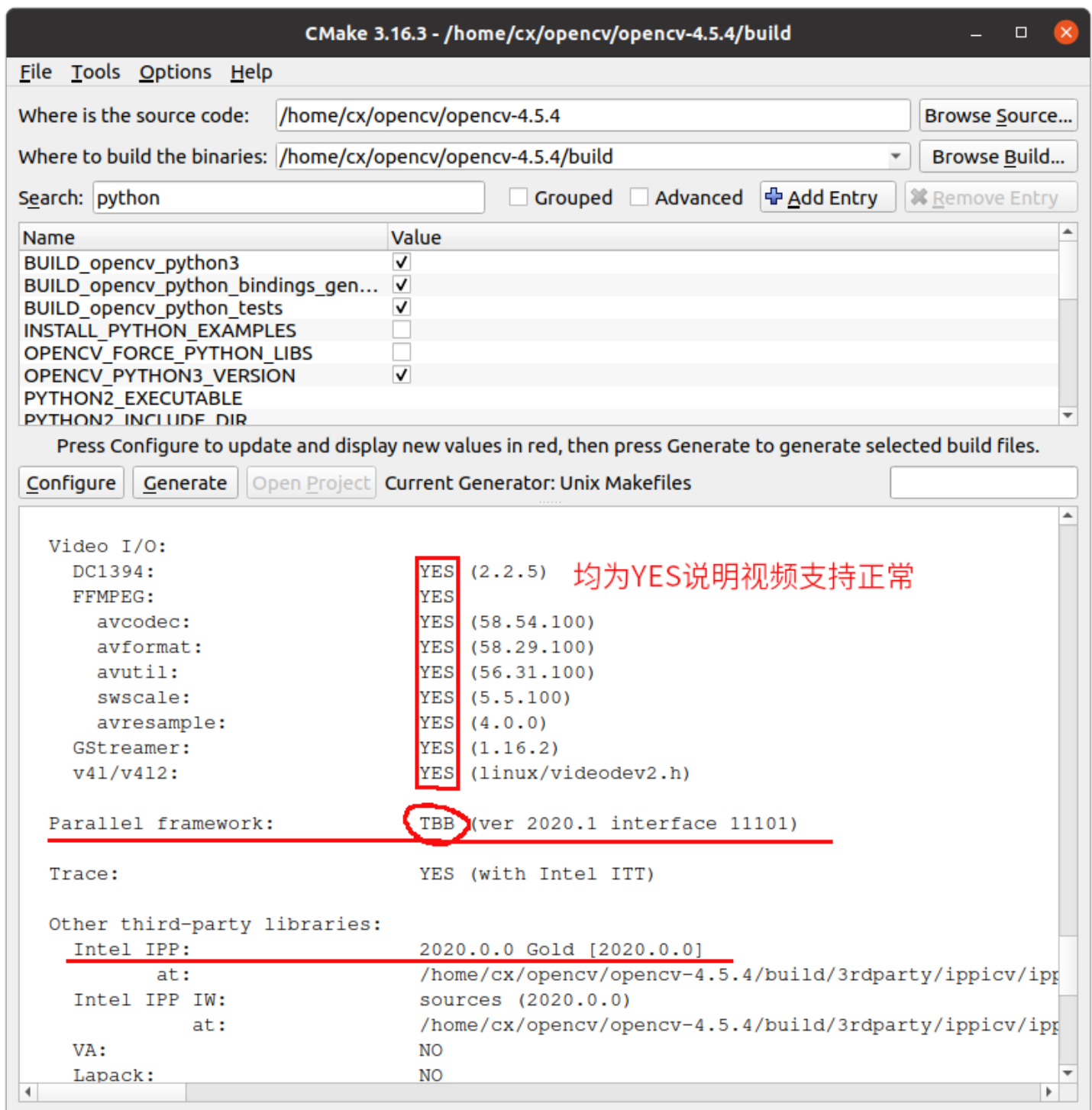


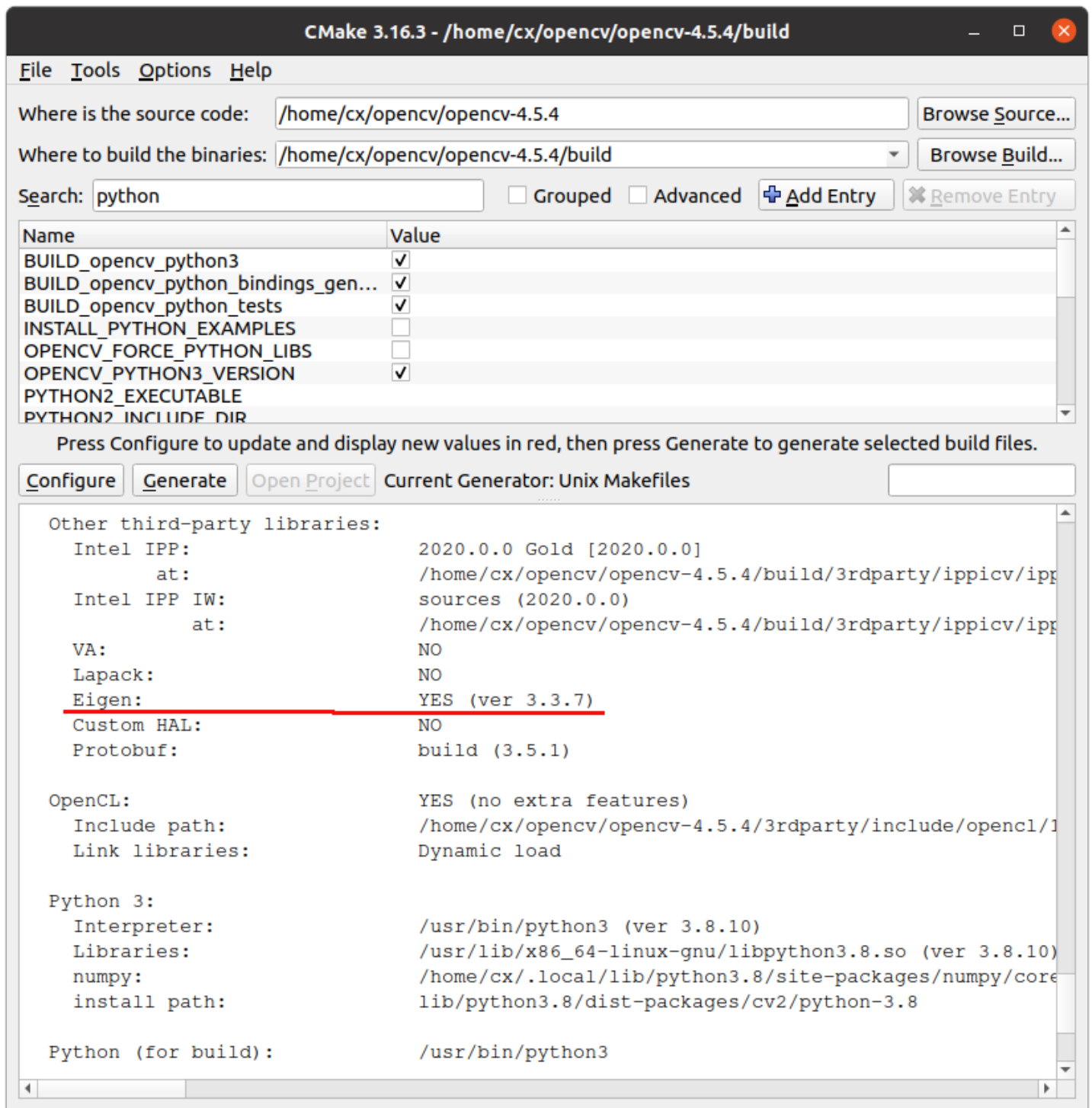
此时要注意CUDA_ARCH_BIN选项中会有很多小于7.0的数字，是为RTX系列之前的显卡准备的兼容选项，如果你是RTX20系列之后的显卡，可以像我这里这样把7.0之前的全部删掉，加快编译速度，关于CUDA_ARCH_BIN的更多内容请百度。

此时，再点一次configure，所有红色项都会变成白色，cuda配置完成。



在全部配置完成后，重复点击configure，直到没有红色项出现。
此时需要对输出信息进行检查，重点关注的几个部分如下：





检查完毕，确认无误后点击generate生成Makefile，完成配置。

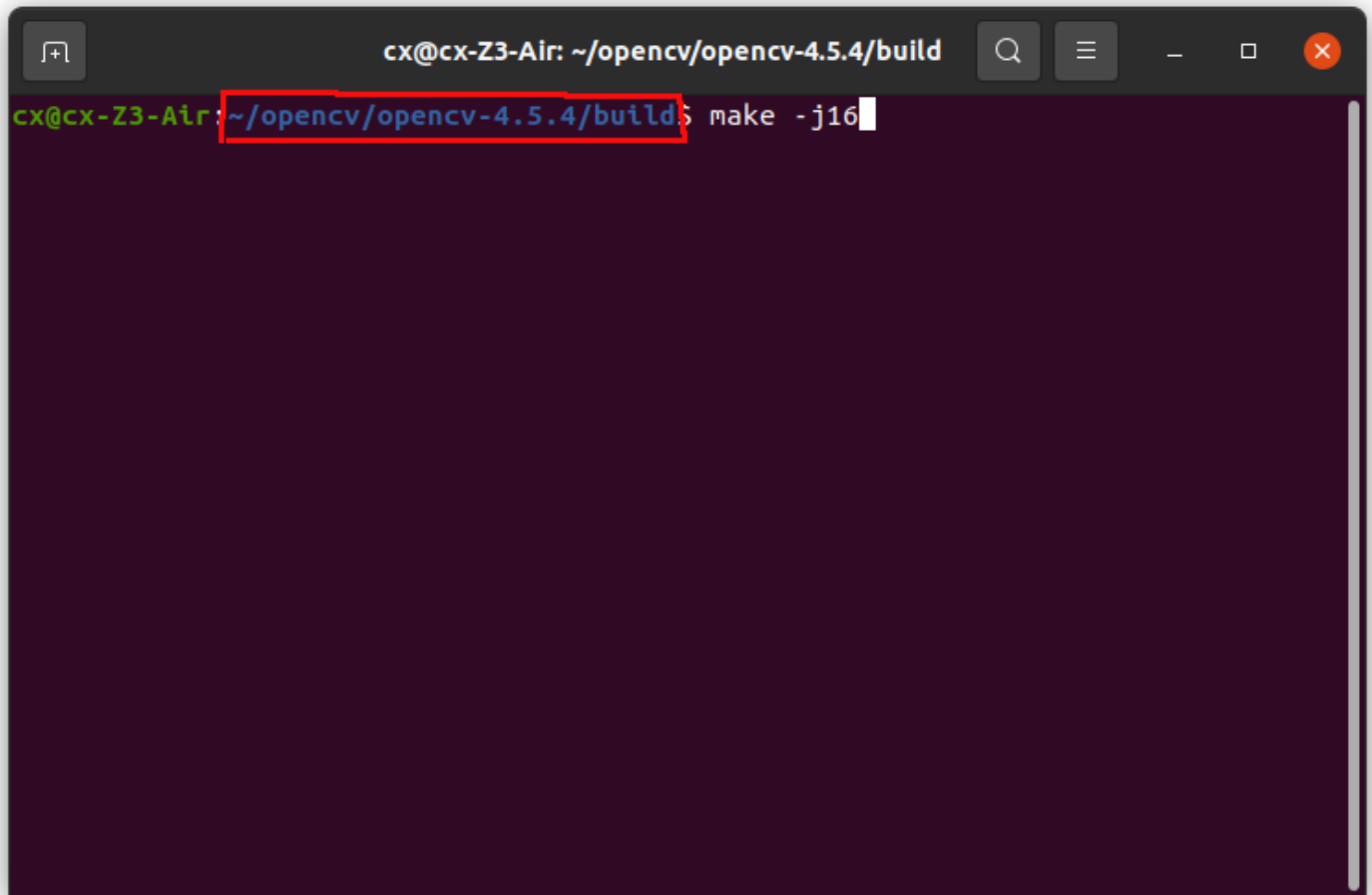
编译、安装

在opencv-4.5.4/build目录下打开终端，键入：

```
make -j16
```

其中，16代表make使用的线程数，建议与CPU线程数量相同，比如笔者的CPU是i7-11800H，8核16线程，故使用命令 `make -j16`。

如图所示:



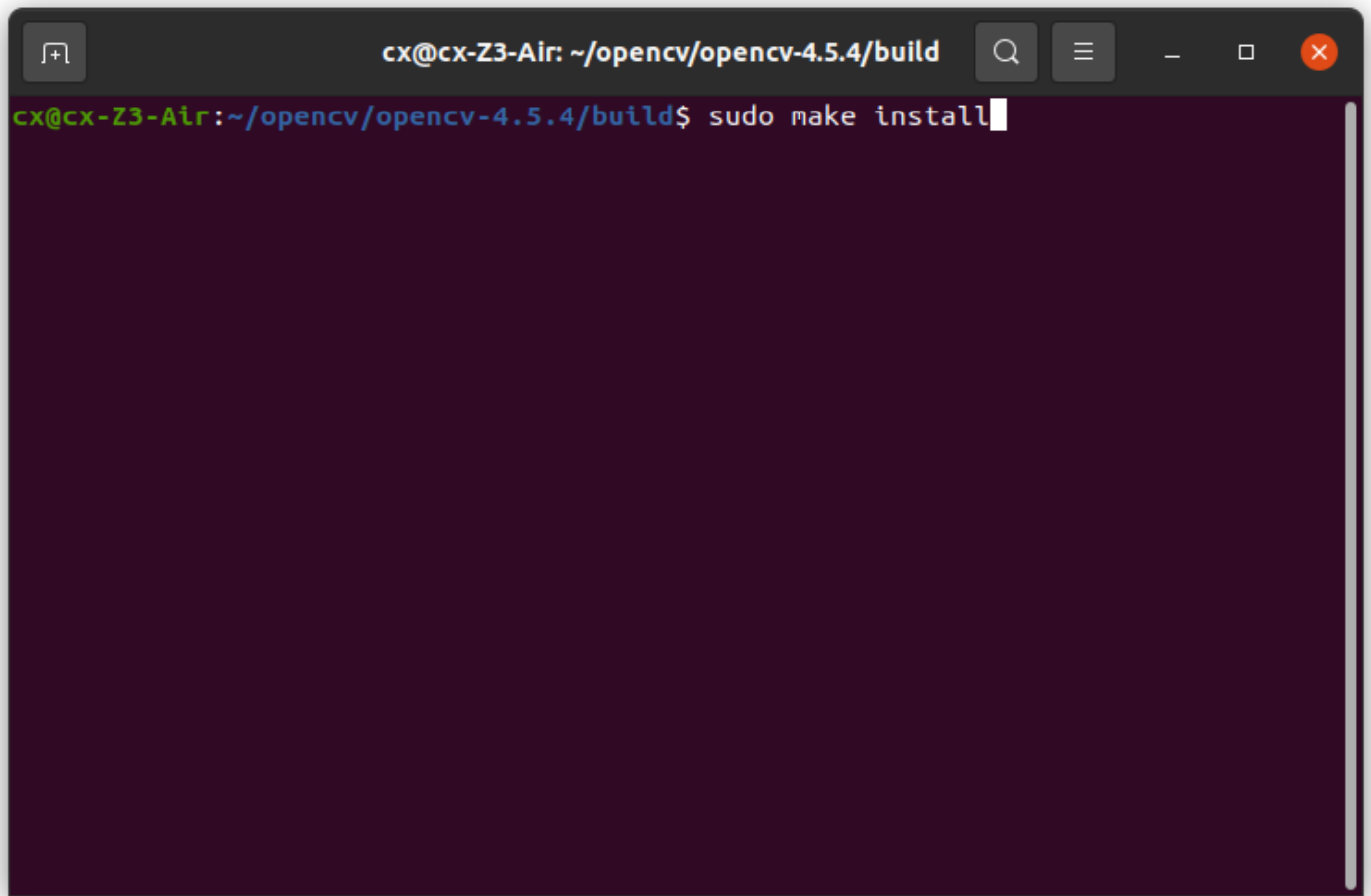
```
cx@cx-Z3-Air: ~/opencv/opencv-4.5.4/build
cx@cx-Z3-Air: ~/opencv/opencv-4.5.4/build$ make -j16
```

编译中:

```
cx@cx-Z3-Air: ~/opencv/opencv-4.5.4/build
dRgbaFile.cpp.o
[ 11%] Building C object 3rdparty/libwebp/CMakeFiles/libwebp.dir/src/enc/webp_enc.c.o
[ 11%] Building C object 3rdparty/libwebp/CMakeFiles/libwebp.dir/src/mux/anim_encode.c.o
[ 11%] Building CXX object 3rdparty/openexr/CMakeFiles/IlmImf.dir/IlmImf/ImfTimeCode.cpp.o
[ 11%] Building CXX object 3rdparty/openexr/CMakeFiles/IlmImf.dir/IlmImf/ImfTimeCodeAttribute.cpp.o
[ 11%] Building CXX object 3rdparty/openexr/CMakeFiles/IlmImf.dir/IlmImf/ImfVecAttribute.cpp.o
[ 11%] Building C object 3rdparty/libwebp/CMakeFiles/libwebp.dir/src/mux/muxedit.c.o
[ 11%] Building CXX object 3rdparty/openexr/CMakeFiles/IlmImf.dir/IlmImf/ImfVersion.cpp.o
[ 11%] Building CXX object modules/sfm/src/libmv/libmv/multiview/CMakeFiles/multiview.dir/projection.cc.o
[ 11%] Building CXX object 3rdparty/openexr/CMakeFiles/IlmImf.dir/IlmImf/ImfWav.cpp.o
[ 11%] Building C object 3rdparty/libwebp/CMakeFiles/libwebp.dir/src/mux/muxinternal.c.o
[ 11%] Building C object 3rdparty/libwebp/CMakeFiles/libwebp.dir/src/mux/muxread.c.o
```

等待编译完全完成后，在终端键入：

```
sudo make install
```

A terminal window with a dark background. The title bar at the top reads 'cx@cx-Z3-Air: ~/opencv/opencv-4.5.4/build'. The terminal shows the command 'cx@cx-Z3-Air:~/opencv/opencv-4.5.4/build\$ sudo make install' with a cursor at the end of the line. The window has standard Linux window controls (minimize, maximize, close) on the right side of the title bar.

```
cx@cx-Z3-Air: ~/opencv/opencv-4.5.4/build
cx@cx-Z3-Air:~/opencv/opencv-4.5.4/build$ sudo make install
```

至此，安装完成。

OpenCV完整性测试

在终端中键入：

```
sudo ldconfig
```

```
cx@cx-Z3-Air: ~/opencv/opencv-4.5.4/build
-- Installing: /usr/local/share/opencv4/haarcascades/haarcascade_upperbody.xml
-- Installing: /usr/local/share/opencv4/lbpcascades/lbpcascade_frontalcatface.xml
-- Installing: /usr/local/share/opencv4/lbpcascades/lbpcascade_frontalface.xml
-- Installing: /usr/local/share/opencv4/lbpcascades/lbpcascade_frontalface_improved.xml
-- Installing: /usr/local/share/opencv4/lbpcascades/lbpcascade_profileface.xml
-- Installing: /usr/local/share/opencv4/lbpcascades/lbpcascade_silverware.xml
-- Installing: /usr/local/bin/opencv_annotation
-- Set runtime path of "/usr/local/bin/opencv_annotation" to "/usr/local/lib:/usr/local/cuda/lib64"
-- Installing: /usr/local/bin/opencv_visualisation
-- Set runtime path of "/usr/local/bin/opencv_visualisation" to "/usr/local/lib:/usr/local/cuda/lib64"
-- Installing: /usr/local/bin/opencv_interactive-calibration
-- Set runtime path of "/usr/local/bin/opencv_interactive-calibration" to "/usr/local/lib:/usr/local/cuda/lib64"
-- Installing: /usr/local/bin/opencv_version
-- Set runtime path of "/usr/local/bin/opencv_version" to "/usr/local/lib:/usr/local/cuda/lib64"
-- Installing: /usr/local/bin/opencv_model_diagnostics
-- Set runtime path of "/usr/local/bin/opencv_model_diagnostics" to "/usr/local/lib:/usr/local/cuda/lib64"
cx@cx-Z3-Air:~/opencv/opencv-4.5.4/build$ sudo ldconfig
```

再键入:

```
pkg-config --libs opencv4
```

```
cx@cx-Z3-Air: ~/opencv/opencv-4.5.4/build
cx@cx-Z3-Air:~/opencv/opencv-4.5.4/build$ pkg-config --libs opencv4
-L/usr/local/lib -lopencv_gapi -lopencv_stitching -lopencv_alphamat -lopencv_aruco -lopencv_barcode -lopencv_bgsegm -lopencv_bioinspired -lopencv_ccalib -lopencv_cudabgsegm -lopencv_cudafeatures2d -lopencv_cudaobjdetect -lopencv_cudastereo -lopencv_dnn_objdetect -lopencv_dnn_superres -lopencv_dpm -lopencv_face -lopencv_freetype -lopencv_fuzzy -lopencv_hdf -lopencv_hfs -lopencv_img_hash -lopencv_intensity_transform -lopencv_line_descriptor -lopencv_mcc -lopencv_quality -lopencv_rapid -lopencv_reg -lopencv_rgbd -lopencv_saliency -lopencv_sfm -lopencv_stereo -lopencv_structured_light -lopencv_phase_unwrapping -lopencv_superres -lopencv_cudacodec -lopencv_surface_matching -lopencv_tracking -lopencv_highgui -lopencv_datasets -lopencv_text -lopencv_plot -lopencv_videostab -lopencv_cudaoptflow -lopencv_optflow -lopencv_cudalegacy -lopencv_videoio -lopencv_cudawarping -lopencv_wechat_qrcode -lopencv_xfeatures2d -lopencv_shape -lopencv_ml -lopencv_ximgproc -lopencv_video -lopencv_xobjdetect -lopencv_objdetect -lopencv_calib3d -lopencv_imgcodecs -lopencv_features2d -lopencv_dnn -lopencv_flann -lopencv_xphoto -lopencv_photo -lopencv_cudaimgproc -lopencv_cudafilters -lopencv_imgproc -lopencv_cudaarithm -lopencv_core -lopencv_cudev
cx@cx-Z3-Air:~/opencv/opencv-4.5.4/build$
```

若出现上图提示，则安装无误。

OpenCV实战测试

写Cmake程序测试，此处不再赘述。