

## Lesson 2 OpenCV Installation

The system used in this article is the official 7-10 buster. Python 3.7 is already installed in the system, so I won't mention how to install it here.

### 1. Modify the Raspberry Pi download source system

The default software download server of the Raspberry Pi is officially designated. If you encounter slow file downloads or errors, you can go to [\4. Extended Courses\3. Extended Courses-Raspberry Pi Motherboard Basic Courses\Lesson 5 Replace Software Download source method](#) for learning

### 2. Install numpy

#### Numpy overview:

Each image has a lot of pixels, which also leads to a lot of array processing in the program. Numpy is a Python extension library. Its processing efficiency for multi-dimensional arrays is much stronger than Python's own array structure, and it can improve the readability of our code.

Numpy is widely used in the field of machine learning due to its powerful multi-dimensional array and matrix computing capabilities.

#### Install Numpy:

Open the Raspberry Pi command line interface and enter the following command to install the Python scientific computing library numpy.

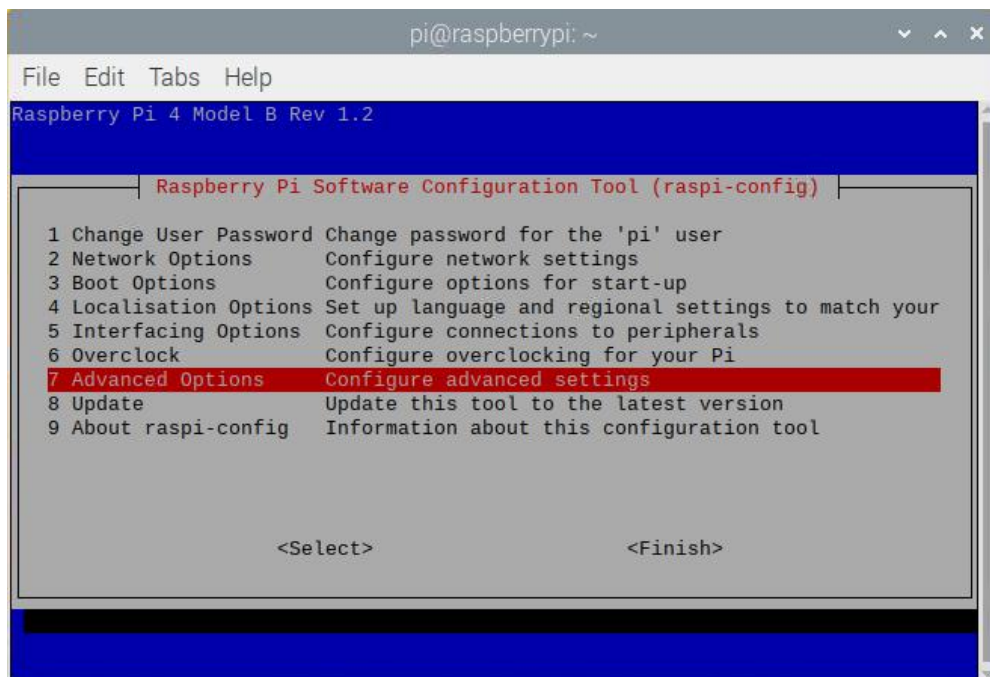
```
sudo pip3 install numpy
```

### 3. In the Raspberry Pi settings to expand the root directory to the entire SD card

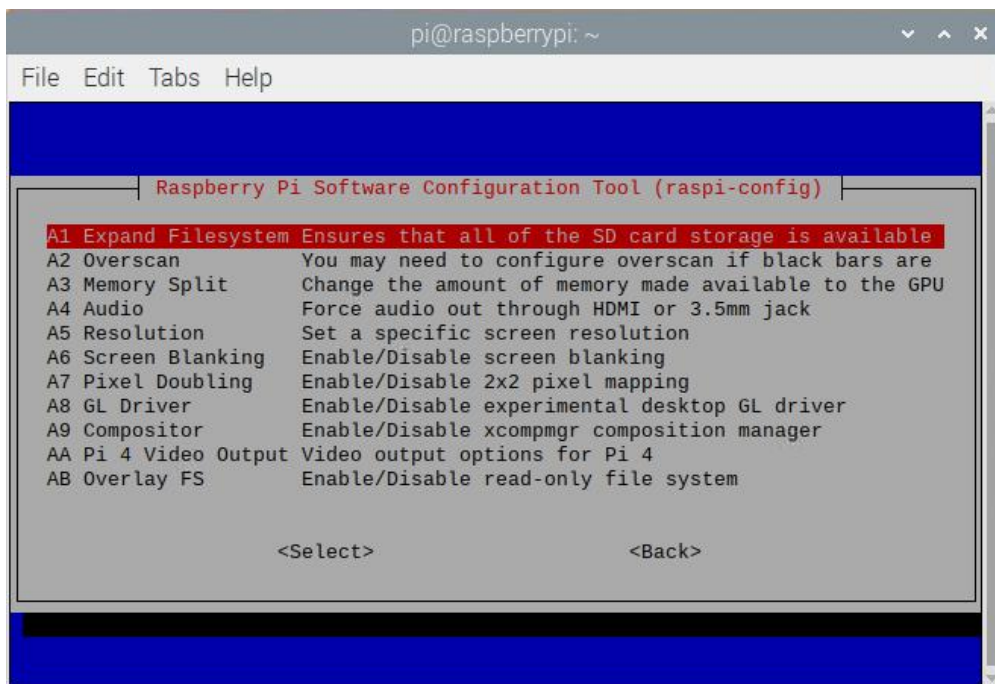
This time to build the OpenCV development environment, at least use a 16G TF card. It is necessary to make full use of the TF storage space, and the TF card space needs to be expanded to the whole TF card.

1) Enter the command line, the following Raspberry Pi configuration screen appears, select 7 Advanced Options:

```
sudo raspi-config
```



2) After pressing enter button, select A1 **Expand Filesystem**, press enter button to confirm the selection, the Raspberry Pi will automatically execute and restart.



#### 4. Install the libraries required by OpenCV

Install OpenCV dependencies to run the following eight commands in sequence. Note that four -dev packages must be installed in the penultimate command.

```
sudo apt-get install build-essential git cmake pkg-config -y
sudo apt-get install libjpeg8-dev -y
sudo apt-get install libtiff5-dev -y
sudo apt-get install libjasper-dev -y
sudo apt-get install libpng12-dev -y
sudo apt-get install libavcodec-dev libavformat-dev libswscale-dev libv4l-dev -y
sudo apt-get install libgtk2.0-dev -y
sudo apt-get install libatlas-base-dev gfortran -y
```

#### 5. Download opencv

Download the two compressed packages to Raspberry Pi directory the

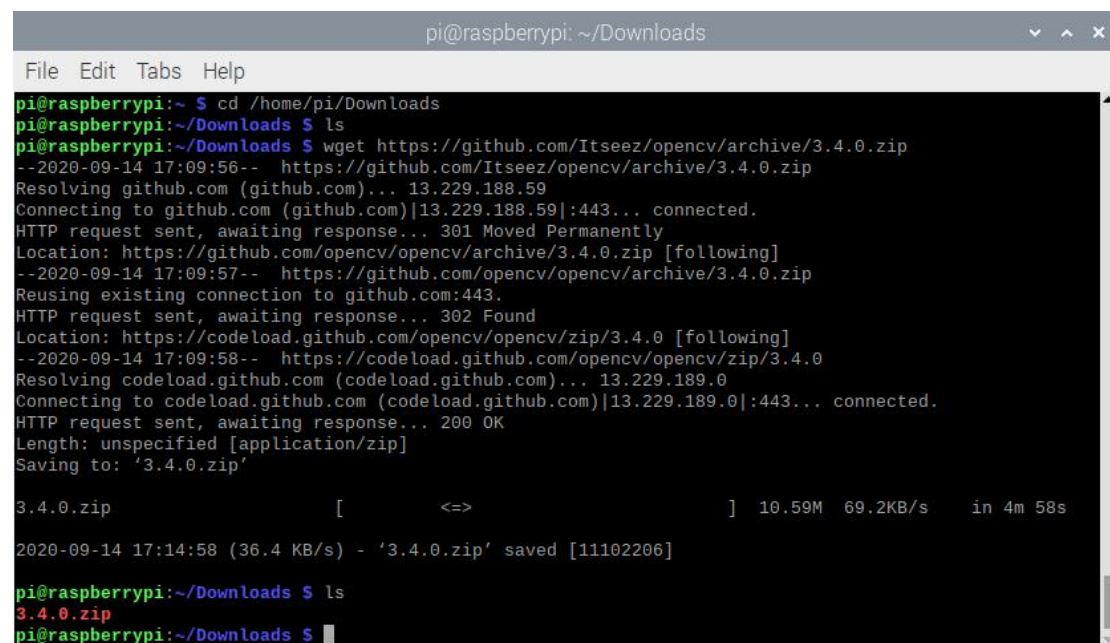
/home/pi/Downloads.

1) Enter the following command on the command line, and switch to the Downloads directory:

```
cd /home/pi/Downloads
```

2) Download the first installation package:

```
wget https://github.com/Itseez/opencv/archive/3.4.0.zip
```



```
pi@raspberrypi: ~/Downloads
File Edit Tabs Help
pi@raspberrypi:~ $ cd /home/pi/Downloads
pi@raspberrypi:~/Downloads $ ls
pi@raspberrypi:~/Downloads $ wget https://github.com/Itseez/opencv/archive/3.4.0.zip
--2020-09-14 17:09:56-- https://github.com/Itseez/opencv/archive/3.4.0.zip
Resolving github.com (github.com)... 13.229.188.59
Connecting to github.com (github.com)[13.229.188.59]:443... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: https://github.com/opencv/opencv/archive/3.4.0.zip [following]
--2020-09-14 17:09:57-- https://github.com/opencv/opencv/archive/3.4.0.zip
Reusing existing connection to github.com:443.
HTTP request sent, awaiting response... 302 Found
Location: https://codeload.github.com/opencv/opencv/zip/3.4.0 [following]
--2020-09-14 17:09:58-- https://codeload.github.com/opencv/opencv/zip/3.4.0
Resolving codeload.github.com (codeload.github.com)... 13.229.189.0
Connecting to codeload.github.com (codeload.github.com)[13.229.189.0]:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [application/zip]
Saving to: '3.4.0.zip'

3.4.0.zip           [          <=>          ] 10.59M  69.2KB/s   in 4m 58s

2020-09-14 17:14:58 (36.4 KB/s) - '3.4.0.zip' saved [11102206]

pi@raspberrypi:~/Downloads $ ls
3.4.0.zip
pi@raspberrypi:~/Downloads $
```

3) Download the second installation package:

```
wget https://github.com/Itseez/opencv_contrib/archive/3.4.0.zip
```

4) After downloading, rename the first compressed package to opencv-3.4.0.zip, and rename the second compressed package to opencv\_contrib-3.4.0.zip

```
sudo mv 3.4.0.zip opencv-3.4.0.zip
sudo mv 3.4.0.zip.1 opencv_contrib-3.4.0.zip
```

```
pi@raspberrypi: ~/Downloads
File Edit Tabs Help
pi@raspberrypi:~ $ cd Downloads/
pi@raspberrypi:~/Downloads $ ls
3.4.0.zip 3.4.0.zip.1
pi@raspberrypi:~/Downloads $ sudo mv 3.4.0.zip opencv-3.4.0.zip
pi@raspberrypi:~/Downloads $ ls
3.4.0.zip.1 opencv-3.4.0.zip
pi@raspberrypi:~/Downloads $ sudo mv 3.4.0.zip.1 opencv_contrib-3.4.0.zip
pi@raspberrypi:~/Downloads $ ls
opencv-3.4.0.zip opencv_contrib-3.4.0.zip
pi@raspberrypi:~/Downloads $
```

5) Unzip these two compressed packages

```
cd /home/pi/Downloads
unzip opencv-3.4.0.zip
unzip opencv_contrib-3.4.0.zip
```

## 6. Set compilation parameters

1) Enter the following command line to create a build folder to store compiled files

```
cd /home/pi/Downloads/opencv-3.4.0
mkdir build
cd build
```

2) Set the CMAKE parameters. Note that the following is a line of commands (including the last two points), and you need to wait patiently for about 15 minutes:

```
cmake -D CMAKE_BUILD_TYPE=RELEASE -D CMAKE_INSTALL_PREFIX=/usr/local -D
INSTALL_C_EXAMPLES=ON -D INSTALL_PYTHON_EXAMPLES=ON -D OPENCV_EXTRA_MODU
LES_PATH=/home/pi/Downloads/opencv_contrib-3.4.3/modules -D BUILD_EXAMPL
ES=ON -D WITH_LIBV4L=ON PYTHON3_EXECUTABLE=/usr/bin/python3.7 PYTHON_INC
LUDE_DIR=/usr/include/python3.7 PYTHON_LIBRARY=/usr/lib/arm-linux-gnueab
```

```
ihf/libpython3.7m.so PYTHON3_NUMPY_INCLUDE_DIRS=/usr/lib/python3/dist-packages/numpy/core/include ..
```

3) According to the figure below, judge whether you have successfully configured CMAKE

The left screenshot shows the CMake configuration output for a failed build. It lists various options and their values, but ends with an error message: "Configuring incomplete, errors occurred!". A red 'X' is drawn over the error message. The right screenshot shows the CMake configuration output for a successful build. It lists the same options and values, but ends with a message: "Build files have been written to: /home/pi/Downloads/opencv-3.4.0/build". A green checkmark is drawn over the success message.

4) If it fails, it may be because the path of opencv and python in the configuration is wrong, and you need to change the path appropriately according to your version. If successful, you can start the most important compilation.

## 7. Compile OpenCV

1) The last and most important step: compilation. To ensure that the Raspberry Pi has at least 5G of storage space, it is recommended to use the command line tool on the Raspberry Pi desktop to run this command instead of using a remote ssh connection. Because it takes too long to execute the command, if ssh is disconnected in the middle, it is impossible to know whether the installation has been completed.

```
cd /home/pi/Downloads/opencv-3.4.0/build
make
```

2) After two hours of compilation, it was 40% complete. Wait for five hours of compilation. Note that during this period, the Raspberry Pi must have sufficient



power supply and do not run other tasks to avoid reporting errors due to insufficient memory.

```

pi@raspberrypi: ~/Downloads/opencv-3.4.0/build
File Edit Tabs Help
[ 99%] Building CXX object samples/tapi/CMakeFiles/example_tapi_hog.dir/hog.cpp.o
[ 99%] Linking CXX executable ../../bin/tapi-example-hog
[ 99%] Built target example_tapi_hog
Scanning dependencies of target example_tapi_ufacetedetect
[ 99%] Building CXX object samples/tapi/CMakeFiles/example_tapi_ufacetedetect.dir/ufacetedetect.cpp.o
[ 99%] Linking CXX executable ../../bin/tapi-example-ufacetedetect
[ 99%] Built target example_tapi_ufacetedetect
Scanning dependencies of target example_tapi_bgfg_segm
[ 99%] Building CXX object samples/tapi/CMakeFiles/example_tapi_bgfg_segm.dir/bgfg_segm.cpp.o
[ 99%] Linking CXX executable ../../bin/tapi-example-bgfg_segm
[ 99%] Built target example_tapi_bgfg_segm
Scanning dependencies of target example_tapi_opencv_custom_kernel
[ 99%] Building CXX object samples/tapi/CMakeFiles/example_tapi_opencv_custom_kernel.dir/opencv_custom_kernel.cpp.o
[ 99%] Linking CXX executable ../../bin/tapi-example-opencv_custom_kernel
[ 99%] Built target example_tapi_opencv_custom_kernel
Scanning dependencies of target example_tapi_squares
[ 99%] Building CXX object samples/tapi/CMakeFiles/example_tapi_squares.dir/squares.cpp.o
[100%] Linking CXX executable ../../bin/tapi-example-squares
[100%] Built target example_tapi_squares
Scanning dependencies of target example_tapi_camshift
[100%] Building CXX object samples/tapi/CMakeFiles/example_tapi_camshift.dir/camshift.cpp.o
[100%] Linking CXX executable ../../bin/tapi-example-camshift
[100%] Built target example_tapi_camshift
Scanning dependencies of target example_tapi_tvll_optical_flow
[100%] Building CXX object samples/tapi/CMakeFiles/example_tapi_tvll_optical_flow.dir/tvll_optical_flow.cpp.o
[100%] Linking CXX executable ../../bin/tapi-example-tvll_optical_flow
[100%] Built target example_tapi_tvll_optical_flow
Scanning dependencies of target example_tapi_clahe
[100%] Building CXX object samples/tapi/CMakeFiles/example_tapi_clahe.dir/clahe.cpp.o
/home/pi/Downloads/opencv-3.4.0/samples/tapi/clahe.cpp: In function 'int main(int, char**)':
/home/pi/Downloads/opencv-3.4.0/samples/tapi/clahe.cpp:52:75: warning: cast between incompatible function types from 'void (*)
(int)' to 'cv::TrackbarCallback' {aka 'void (*)(int, void*)'} [-Wcast-function-type]
    createTrackbar("Tile Size", "CLAHE", &tilesize, 32, (TrackbarCallback)TSize_Callback);
                                                                    ~~~~~
/home/pi/Downloads/opencv-3.4.0/samples/tapi/clahe.cpp:53:77: warning: cast between incompatible function types from 'void (*)
(int)' to 'cv::TrackbarCallback' {aka 'void (*)(int, void*)'} [-Wcast-function-type]
    createTrackbar("Clip Limit", "CLAHE", &cliplimit, 20, (TrackbarCallback)Clip_Callback);
                                                                    ~~~~~
[100%] Linking CXX executable ../../bin/tapi-example-clahe
[100%] Built target example_tapi_clahe
pi@raspberrypi:~/Downloads/opencv-3.4.0/build $

```

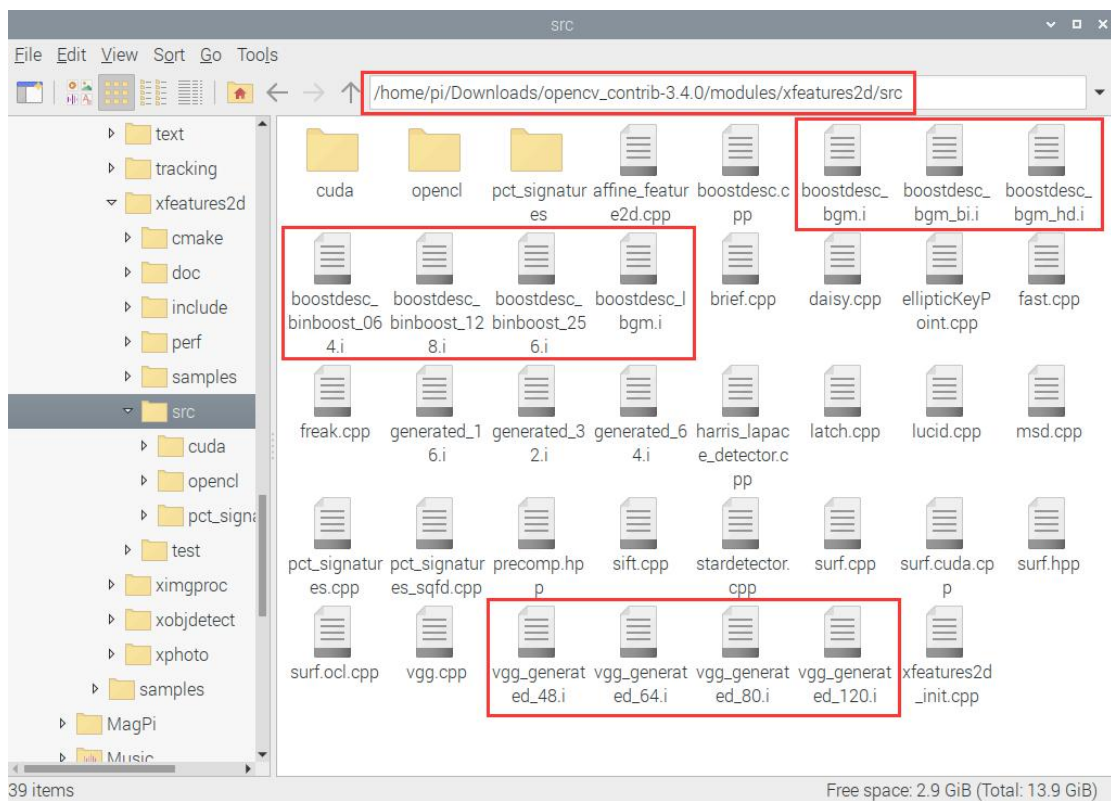
**Note:** If you meet an error like the one below:

```

[ 79%] Building CXX object modules/xfeatures2d/CMakeFiles/opencv_xfeatures2d.dir
/src/boostdesc.cpp.o
/home/pi/Downloads/opencv_contrib-3.4.0/modules/xfeatures2d/src/boostdesc.cpp:64
6:20: fatal error: boostdesc_bgm.i: No such file or directory
    #include "boostdesc_bgm.i"
           ~~~~~
compilation terminated.

```

We can unzip the [boostdesc\\_bgm.i....rar](#) compressed package in the directory where the current document is located, and copy all the files inside to the directory [opencv\\_contrib/modules/xfeatures2d/src/](#)



Recompile again.

If you encounter something like "fatal error: opencv2/xfeatures2d/cuda.hpp:

No such file or directory"

```
[ 96%] Building CXX object modules/stitching/CMakeFiles/opencv_stitching_pch_dep
help.dir/opencv_stitching_pch_dephelp.cxx.o
In file included from /home/pi/Downloads/opencv-3.4.0/modules/stitching/include/
opencv2/stitching.hpp:49,
      from /home/pi/Downloads/opencv-3.4.0/modules/stitching/src/prec
omp.hpp:59,
      from /home/pi/Downloads/opencv-3.4.0/build/modules/stitching/op
encv_stitching_pch_dephelp.cxx:1:
/home/pi/Downloads/opencv-3.4.0/modules/stitching/include/opencv2/stitching/deta
il/matchers.hpp:52:12: fatal error: opencv2/xfeatures2d/cuda.hpp: No such file o
r directory
# include "opencv2/xfeatures2d/cuda.hpp"
~~~~~
compilation terminated.
make[2]: *** [modules/stitching/CMakeFiles/opencv_stitching_pch_dephelp.dir/buil
d.make:63: modules/stitching/CMakeFiles/opencv_stitching_pch_dephelp.dir/opencv_
stitching_pch_dephelp.cxx.o] Error 1
make[1]: *** [CMakeFiles/Makefile2:21309: modules/stitching/CMakeFiles/opencv_st
itching_pch_dephelp.dir/all] Error 2
make: *** [Makefile:163: all] Error 2
```

The command to find files can be executed in the root directory:



```
find . -name "cuda.hpp"
```

It turns out that the absolute path of the cuda.hpp file is located at:

**/home/pi/Downloads/opencv\_contrib-3.4.0/modules/xfeatures2d/include/  
opencv2/xfeatures2d/cuda.hpp**

So, according to the prompt information, we can change

**/home/pi/Downloads/opencv-3.4.0/modules/stitching/include/opencv2/stitching/detail/matchers.hpp** in 52 line

```
#include "opencv2/xfeatures2d/cuda.hpp"
```

modified to the following absolute path form

```
#include "/home/pi/Downloads/opencv_contrib-3.4.0/modules/xfeatures2d/include/opencv2/xfeatures2d/cuda.hpp"
```

Recompile

If there is an error similar to the following

```
/home/pi/Downloads/opencv-3.4.0/build/modules/python_bindings_generator/pyopencv_generated_ns_reg.h:2779:30: warning: cast between incompatible function types from 'PyObject* (*)(PyObject*, PyObject*, PyObject*)' {aka 'object* (*)(object*, object*, object*)'} to 'PyCFunction' {aka 'object* (*)(object*, object*)'} [-Wcast-function-type]
{"inpaint", (PyCFunction)pyopencv_cv_xphoto_inpaint, METH_VARARGS | METH_KEYWORDS, "inpaint(src, mask, dst, algorithmType) -> None\n. @brief The function implements different single-image inpainting algorithms.\n. \n. See the original paper @cite He2012 for details.\n. \n. @param src source image, it could be of any type and any number of channels from 1 to 4. In case of\n. 3- and 4-channels images the function expect them in CIElab colorspace or similar one, where first\n. color component shows intensity, while second and third shows colors. Nonetheless you can try any\n. colorspaces.\n. @param mask mask (CV_8UC1), where non-zero pixels indicate valid image area, while zero pixels\n. indicate area to be inpainted\n. @param dst destination image\n. @param algorithmType see xphoto::InpaintTypes"},
~~~~~
make[2]: *** [modules/python3/CMakeFiles/opencv_python3.dir/build.make:63: modules/python3/CMakeFiles/opencv_python3.dir/_src/c2/cv2.cpp.o] Error 1
make[1]: *** [CMakeFiles/Makefile2:21633: modules/python3/CMakeFiles/opencv_python3.dir/all] Error 2
make: *** [Makefile:163: all] Error 2
```

You can add "PyString\_AsString(obj);" near line 885 in the **/home/pi/Downloads/opencv-3.4.0/modules/python/src2/cv2.cpp** file (as shown in the figure below) (char \*)"

```

878
879 template<>
880 bool pyopencv_to(PyObject* obj, String& value, const char* name)
881 {
882     (void)name;
883     if(!obj || obj == Py_None)
884         return true;
885     char* str = (char *)PyString_AsString(obj);
886     if(!str)
887         return false;
888     value = String(str);
889     return true;
890 }
891
892 template<>
893 bool pyopencv_to(PyObject* obj, Size& sz, const char* name)
894 {
895     (void)name;
896     if(!obj || obj == Py_None)
897         return true;
898     return PyArg_ParseTuple(obj, "ii", &sz.width, &sz.height) > 0;
899 }

```

After saving the modify, compile again.

1) After the make compile command is executed, execute the following installation command, it will take one minute to execute the command:

```
sudo make install
```

## 8. Test OpenCV on Python3

1) After installation, enter the following command in the command line:

```
python3
import cv2
cv2.__version__
```

2) If the result shown in the figure below appears, it means that the OpenCV installation under the Python3 environment is successful.

```
-- Installing: /usr/local/share/OpenCV/samples/python/floodfill.py
-- Installing: /usr/local/share/OpenCV/samples/python/gabor_threads.py
-- Installing: /usr/local/share/OpenCV/samples/python/gaussian_mix.py
-- Installing: /usr/local/share/OpenCV/samples/python/grabcut.py
-- Installing: /usr/local/share/OpenCV/samples/python/hist.py
-- Installing: /usr/local/share/OpenCV/samples/python/houghcircles.py
-- Installing: /usr/local/share/OpenCV/samples/python/houghlines.py
-- Installing: /usr/local/share/OpenCV/samples/python/inpaint.py
-- Installing: /usr/local/share/OpenCV/samples/python/kalman.py
-- Installing: /usr/local/share/OpenCV/samples/python/kmeans.py
-- Installing: /usr/local/share/OpenCV/samples/python/lappyr.py
-- Installing: /usr/local/share/OpenCV/samples/python/letter_recog.py
-- Installing: /usr/local/share/OpenCV/samples/python/lk_homography.py
-- Installing: /usr/local/share/OpenCV/samples/python/lk_track.py
-- Installing: /usr/local/share/OpenCV/samples/python/logpolar.py
-- Installing: /usr/local/share/OpenCV/samples/python/morphology.py
-- Installing: /usr/local/share/OpenCV/samples/python/mosse.py
-- Installing: /usr/local/share/OpenCV/samples/python/mouse_and_match.py
-- Installing: /usr/local/share/OpenCV/samples/python/mser.py
-- Installing: /usr/local/share/OpenCV/samples/python/opencv_version.py
-- Installing: /usr/local/share/OpenCV/samples/python/opt_flow.py
-- Installing: /usr/local/share/OpenCV/samples/python/peopledetect.py
-- Installing: /usr/local/share/OpenCV/samples/python/plane_ar.py
-- Installing: /usr/local/share/OpenCV/samples/python/plane_tracker.py
-- Installing: /usr/local/share/OpenCV/samples/python/squares.py
-- Installing: /usr/local/share/OpenCV/samples/python/stereo_match.py
-- Installing: /usr/local/share/OpenCV/samples/python/texture_flow.py
-- Installing: /usr/local/share/OpenCV/samples/python/tst_scene_render.py
-- Installing: /usr/local/share/OpenCV/samples/python/turing.py
-- Installing: /usr/local/share/OpenCV/samples/python/video.py
-- Installing: /usr/local/share/OpenCV/samples/python/video_threaded.py
-- Installing: /usr/local/share/OpenCV/samples/python/video_v4l2.py
-- Installing: /usr/local/share/OpenCV/samples/python/watershed.py
pi@raspberrypi:~/Downloads/opencv-3.4.0/build $ python3
Python 3.7.3 (default, Apr 3 2019, 05:39:12)
[GCC 8.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import cv2
>>> cv2.__version__
'3.4.0'
>>>
```