OpenCV Python

Linux Installation Manual

Version 1.5

e-con Systems

09/14/2023

e-con Systems

Think Camera. Think e-con.

Disclaimer

e-con Systems reserves the right to edit/modify this document without any prior intimation of whatsoever.

© Copyright e-con Systems. 2020. All rights reserved.

Contents

Introduction to OpenCV

Open Source Computer Vision Library (OpenCV) is an open source computer vision and machine learning software library. OpenCV libraries are used to communicate with Cameras. APIs introduced in the OpenCV can be supported with all e-Con systems cameras.

This document helps to install the OpenCV in Linux and build a sample code to access the camera with OpenCV.

Prerequisites

The prerequisites are as follows:

- Download OpenCV from here (https://github.com/opencv/opencv)
- Or download the official OpenCV release using wget :

```
$ cd ~
$ wget -O opencv.zip \
https://github.com/opencv/opencv/archive/4.6.0.zip
```

- Download OpenCV contrib from here (https://github.com/opencv/opencv_contrib)
- Or download the official OpenCV contrib release using wget :

```
$ wget -O opencv_contrib.zip \
https://github.com/opencv/opencv_contrib/archive/4.6.0.zip
```

- Unzip the archieves

```
$ unzip opencv.zip
$ unzip opencv_contrib.zip
```

- GCC 4.4.
- CMake 2.8.7 or higher.
- GTK+2.x or higher, including headers (libgtk2.0-dev).
- Python 3.6.9
- Library packages: libjpeg-dev, libpng-dev, libtiff-dev, libjasper-dev, libavcodec-dev, libavformat-dev, libswscale-dev, libv4l-dev, libxvidcore-dev, libx264-dev, libgtk-3-dev, libatlas-base-dev, gfortran.

Description

The following steps have been tested on Ubuntu 20.04 and with Python 3.8.10. OpenCV must work on any other relatively modern version of Linux OS.

© Copyright e-con Systems. 2020. All rights reserved.

Building OpenCV

The OpenCV is a sample command line application used to demonstrate some of the features of the e-con Cameras with OpenCV APIs. The steps to build OpenCV are as follows:

Step 1. Installing Dependencies
Step 2. Configuring OpenCV

Step 1 - Installing Dependencies

The below table lists the dependencies to be installed for using OpenCV.

Table 1: Installing Dependencies

Dependencies	Commands
Some general development libraries	`$ sudo apt-get install build-essential make cmake cmake-qt-gui g++ unzip pkg-config`
Video4Linux Camera development libraries	`$ sudo apt-get install libv4l-dev`
OpenGL development libraries for creating graphical windows	`$ sudo apt-get install libglew-dev`
GTK development libraries for creating graphical windows	`$ sudo apt-get install libgtk-3-dev`
Udev development libraries for accessing device information	`$ sudo apt-get install libudev-dev`
Libav video input or output development libraries	`$ sudo apt-get install libavformat-dev libavutil-dev libswscale-dev libavcodec-dev libavcodec-ffmpeg-extra56 libavformat-ffmpeg56 libavutil-ffmpeg54 libswscale-ffmpeg3 libdc1394-* libjpeg-dev libpng-dev libtiff-dev libjasper-dev libxvidcore-dev libx264-dev libopencv-dev`
Eigen3 math development libraries	`$ sudo apt-get install libeigen3-dev`
Python 3 headers and libraries	`$ sudo apt-get install python3.6-dev`
Pyudev library	`$ sudo python3 -m pip install pyudev`
Numpy library	`$sudo apt-get install python3-numpy`

In Ubuntu , the dependency libraries varies.

Based on the Ubuntu versions, the development libraries will differ, for example in ubuntu 18.04 libavcodec version will be 57. Thus, install dependencies which supports the ubuntu versions.

For 20.04 OS, extra dependencies are needed. They include: pkg-config, gfortran, openexr, libatlas-base-dev, libtbb2, libopenexr-dev, libgstreamer-plugins-base1.0-dev, libgstreamer1.0-dev

For 18.04 Os: libavcodec-extra57, pkg-config, libavformat57, libavutil55

Step 2 - Configuring OpenCV

The steps to configure OpenCV are as follows:

1. Replace the **videoio** folder inside the downloaded /opencv-4.6.0/modules directory with the folder provided by e-con Systems in this link (https:/github.com/econsystems/opencv/tree/master/Source)
2. To build and install OpenCV, go to the downloaded opencv-4.6.0 directory.

```
$ mkdir release
$ cd release
$ cmake -D CMAKE_BUILD_TYPE=RELEASE \
    -D CMAKE_INSTALL_PREFIX=/usr/local \
    -D BUILD_opencv_python3=ON \
    -D BUILD_opencv_world=ON \
    -D WITH_CUDA=OFF \
    -D INSTALL_PYTHON_EXAMPLES=ON \
    -D OPENCV_EXTRA_MODULES_PATH=~/<opencv_contrib-
4.6.0/modules \
    -D OPENCV_ENABLE_NONFREE=ON \
    -D OPENCV_GENERATE_PKGCONFIG=ON \
    -D BUILD_EXAMPLES=ON \
    -D WITH_GPHOTO2=OFF \
    -D BUILD_TESTS=OFF \
    -D WITH_VTK=OFF \
    -D WITH_GSTREAMER=OFF \
    -D BUILD_PERF_TESTS=OFF ..
```

3. To build and install the OpenCV libraries in the location **/usr/local/lib/,** you must run the following commands.

```
    $ sudo make -j4 && sudo make install
```

© Copyright e-con Systems. 2020. All rights reserved.

Building Sample Code

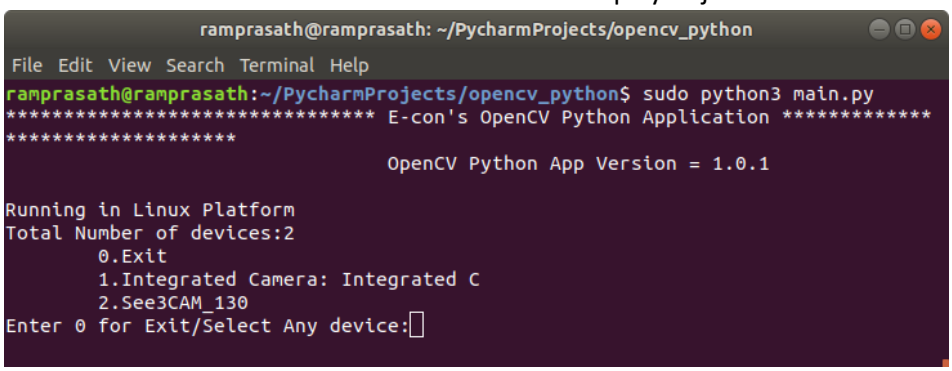
This section describes about how to build the sample code.

The steps to run sample application are as follows:

1. Download OpenCV Python sample command line application from here (https://github.com/econsystems/opencv/tree/master/Source).
4. To run the sample code, run the following command.

```
$ sudo python3 main.py
```

5. Number of devices connected to the PC will be displayed just as below shown.



Using Prebuilt binary

This section describes about how to build the sample code.

The steps to run sample application are as follows:

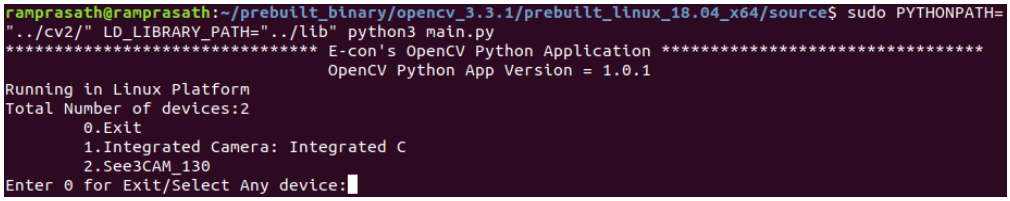
1. Download OpenCV Python Application package from here (https://github.com/econsystems/opencv/tree/master/Source).
2. Go to the source folder.
6. To run the sample code, run the following command.

```
   $ sudo PYTHONPATH="@path/cv2/"
LD_LIBRARY_PATH="@path/lib" python3 main.py
```

*eg:setting cv2 path /Binary/<Opencv_4.6.0>/Linux/<18.04_x64 or 20.04_x64>/cv2/

*eg:setting lib path Binary/<Opencv_4.6.0>/Linux/<18.04_x64 or 20.04_x64>/lib

7. Number of devices connected to the PC will be displayed just as below shown.



© Copyright e-con Systems. 2020. All rights reserved.

Troubleshooting

In this section, you can view the list of commonly occurring issues and their troubleshooting steps.

Camera Device connected, but the application does not displayed.

It seems like the camera is not properly detected by Linux, so check with the path **cd /sys/class/video4linux/** to detect the device.

CMake Error: The source directory does not appear to contain CMakeLists.txt.

1. Run cmake-gui from the terminal.
2. Provide the source and release folder path of the OpenCV project.
3. Configure and generate the solution.

Compiling command line application, libudev.so error adding symbols, DSO missing from command line.

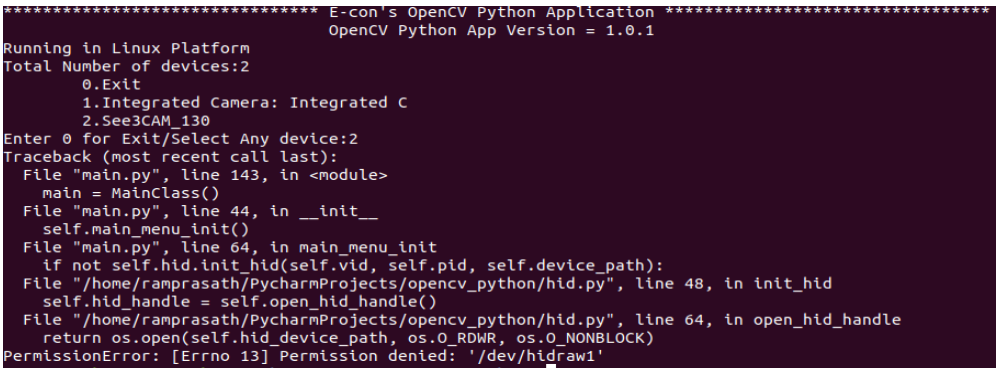
Include the ludev dependency in the compilation command.

Eg: CMAKE_CXX_FLAGS=”-ludev”

Camera displays in the command line but selecting the index does not end up with the preview.

Make sure the camera connected to the port starts with video0 **/sys/class/video4linux**.

PermissionError: [Errno 13] Permission denied: ‘/dev/hidraw’.



Provide **sudo** permission to the command.

```
sudo python3 main.py
```

In Ubuntu , the dependency libraries varies.

Based on the Ubuntu versions, the development libraries will differ, for example in ubuntu 16.04, libavcodec version will be 56, but in ubuntu 18.04 libavcodec version will be 57. Thus, install dependencies which supports the ubuntu versions.

For 20.04 OS, extra dependencies are needed. They include:

pkg-config, gfortran, openexr, libatlas-base-dev, libtbb2, libopenexr-dev, libgstreamer-plugins-base1.0-dev, libgstreamer1.0-dev

For 18.04 Os: `libavcodec-extra57 pkg-config libavformat57 libavutil55`

For 16.4Os: libavcodec-ffmpeg-extra56 libavformat-ffmpeg56 libavutil-ffmpeg54 libswscale-ffmpeg3

Error loading libopencv_world.so

```
sudo LD LIBRARY PATH=<path to the libopencv world.so>
python3 main.py
```

Instead, if you wish to add the path to libopencv_world.so, permanently to the library path. Go to /etc/ld.so.conf.d/. Create a config file named opencv.conf, mention the path inside the config file and give the command:

```
$ sudo ldconfig
```

ModuleNotFoundError: No module named 'cv2'

```
ramprasath@ramprasath:~/PycharmProjects/opencv_python$ python main.py
Traceback (most recent call last):
  File "main.py", line 4, in <module>
    import display
  File "/home/ramprasath/PycharmProjects/opencv_python/display.py", line 3, in <module>
    import cv2
ModuleNotFoundError: No module named 'cv2'
ramprasath@ramprasath:~/PycharmProjects/opencv_python$
```

Step 1: Be sure to check the site-packages (and even dist-packages) directory for the system install of Python located in /usr/local/lib/python3.6/site-packages/. Make sure (1) there is a cv2.so file in the site-packages directory and (2) its properly sym-linked to a valid file.

Step 2: If the cv2.so file is not present, check in your build/lib directory of your OpenCV build. There should be a cv2.so file there (if both cmake and make executed without error). If the cv2.so file is present, manually copy it into both the system site-packages (or dist-packages) directory.

AttributeError: 'cv2.VideoCapture' object has no attribute 'getDevices'

© Copyright e-con Systems. 2020. All rights reserved.

```
ramprasath@ramprasath:~/PycharmProjects/opencv_python$ sudo python main.py
****************************** E-con's OpenCV Python Application ********************************
                               OpenCV Python App Version = 1.0.1
Running in Linux Platform
Traceback (most recent call last):
  File "main.py", line 143, in <module>
    main = MainClass()
  File "main.py", line 44, in __init__
    self.main_menu_init()
  File "main.py", line 57, in main_menu_init
    device_info = Device.list_devices(self.cap)
  File "/home/ramprasath/PycharmProjects/opencv_python/device.py", line 22, in list_devices
    ret, device_count = cap.getDevices()
AttributeError: 'cv2.VideoCapture' object has no attribute 'getDevices'
ramprasath@ramprasath:~/PycharmProjects/opencv_python$
```

This error may come if you installed opencv previously without the e-consystems videoio module and the current build and install is not successful. Check in your build/lib directory of your OpenCV build. There should be a cv2.so file there (if both cmake and make executed without error). If the cv2.so file is present, manually copy it into both the system site-packages (or dist-packages) directory.

Then run the following command.

```
   $ sudo PYTHONPATH="@build/lib/python3"
LD_LIBRARY_PATH="@build/lib" python3 main.py
```

Undefined reference to main:

replace videoio module freshly ,delete opencv folders present in

/usr/local/include/

Rebuild opencv again.If problem still exist uncheck OpenCL,gstreamer and opencv test flags

If cv2 module was not builded:

1.Run cmake-gui in terminal

2.Provide the source and release folder path of the OpenCV project.

3.Search for python

4.Edit variabels as shown below

BUILD_opencv_python3	☑
BUILD_opencv_python_bindings_generator	☑
INSTALL_PYTHON_EXAMPLES	☑
OPENCV_FORCE_PYTHON_LIBS	☐
PYTHON2_EXECUTABLE	/usr/bin/python2.7
PYTHON2_INCLUDE_DIR	
PYTHON2_INCLUDE_DIR2	
PYTHON2_LIBRARY	
PYTHON2_LIBRARY_DEBUG	
PYTHON2_NUMPY_INCLUDE_DIRS	
PYTHON2_PACKAGES_PATH	lib/python2.7/dist-packages
PYTHON3_EXECUTABLE	/usr/bin/python3
PYTHON3_INCLUDE_DIR	/usr/include/python3.6m
PYTHON3_INCLUDE_DIR2	
PYTHON3_LIBRARY	/usr/lib/x86_64-linux-gnu/libpython3.6m.so
PYTHON3_LIBRARY_DEBUG	
PYTHON3_NUMPY_INCLUDE_DIRS	/home/vishnumurali/.local/lib/python3.6/site-packages/numpy/core/include
PYTHON3_PACKAGES_PATH	lib/python3.6/dist-packages

Support

Contact Us

If you need any support on OpenCV sample application, please contact us using the Live Chat option available on our website - https://www.e-consystems.com/

Creating a Ticket

If you need to create a ticket for any type of issue, please visit the ticketing page on our website - https://www.e-consystems.com/create-ticket.asp

RMA

To know about our Return Material Authorization (RMA) policy, please visit the RMA Policy page on our website - https://www.e-consystems.com/RMA-Policy.asp

General Product Warranty Terms

To know about our General Product Warranty Terms, please visit the General Warranty Terms page on our website - https://www.e-consystems.com/warranty.asp

© Copyright e-con Systems. 2020. All rights reserved.

Revision History

Rev	Date	Description	Author
1.0	19-May-2020	Initial Draft	Ramson Jehu K
1.1	29-July-2020	Added changes	M Vishnu Murali
1.2	15-August-2020	Added changes	Murali Mohan M
1.3	08-Sep-2020	Added few points in troubleshooting	M Vishnu Murali
1.4	29-July-2022	Updated document for 4.5.5 version	Nivedha Janarthanan
1.5	14-September-2023	Updated document for 4.6.0 version	Sushanth S