

Build and test OpenCV 4 from Git on Ubuntu 18.04

This article reviews building a simple OpenCV 4 video playback application using the OpenCV Git repos on Ubuntu Linux 18.04.

updated: Nov 01, 2018

advertisement

This article reviews building a simple OpenCV 4 video playback application using the OpenCV Git repos on Ubuntu Linux 18.04. This may prove helpful to others since some tweaking was required even after following other resources. If you are having problems getting simple applications to work, maybe this article will help.

First *opencv*, *opencv_contrib*, and *opencv_extra* were all cloned at the same directory level using git:

```
$ mkdir /build/opencv
$ cd /build/opencv

$ git clone https://github.com/opencv/opencv.git
$ git clone https://github.com/opencv/opencv_contrib.git
$ git clone https://github.com/opencv/opencv_extra.git

$ ls
opencv  opencv_contrib  opencv_extra
```

Listed below are the packages we installed for OpenCV. Of course, other standard build related packages are required (e.g., build-essential, python, etc.). However, the tricky part seems to be installing the right video support libraries. And make sure to notice that the -dev packages are required; the standard (non dev) package will be installed by default.

```
$ sudo apt install cmake libtbb2 ffmpeg libgtk2.0-dev libavformat-dev libswscale-dev libtbb
```

Optionally, use dpkg to verify packages were properly installed. An example is shown below:

```
$ dpkg -l | grep ffmpeg
ii  chromium-codecs-ffmpeg-extra  70.0.3538.77-0ubuntu0.18.04.1
ii  ffmpeg                        7:3.4.4-0ubuntu0.18.04.1
```

Next we build the opencv library following the guidance of the O'Reilly [Learning OpenCV](#) book:

```
$ mkdir /build/opencv/opencv/release
$ cd /build/opencv/opencv/release

$ cmake -DCMAKE_BUILD_TYPE=RELEASE -DCMAKE_INSTALL_PREFIX=/usr/local -DOPENCV_EXTRA_MODULES_

$ make
$ sudo make install
```

Below is the configuration summary output from cmake. Note the availability of the ffmpeg video I/O support. Without this, calls to cap.open() produce the following error: **"Unable to stop the stream: Inappropriate ioctl for device"**

```
-- Lapack: NO
-- Eigen: NO
-- Custom HAL: NO
-- Protobuf: build (3.5.1)
--
-- OpenCL: YES (no extra features)
--   Include path: /build/opencv/opencv/3rdparty/include/opencl/1.2
--   Link libraries: Dynamic load
--
-- Python (for build): /usr/bin/python2.7
--
-- Java:
--   ant: NO
--   JNI: NO
--   Java wrappers: NO
--   Java tests: NO
```

```
--      java tests.      NO
--
--      Install to:      /usr/local
```

Make sure your libraries are being found by the dynamic linker. The output below assumes you installed your libraries to /usr/local/lib and that this directory is listed in one of the conf files (e.g., libc.conf) in /etc/ld.so.conf.d

```
$ sudo ldconfig

$ ldconfig -p | grep opencv
    libopencv_videostab.so.4.0 (libc6,x86-64) => /usr/local/lib/libopencv_videostab.so
    libopencv_videostab.so (libc6,x86-64) => /usr/local/lib/libopencv_videostab.so
    libopencv_videoio.so.4.0 (libc6,x86-64) => /usr/local/lib/libopencv_videoio.so.4.0
    libopencv_videoio.so (libc6,x86-64) => /usr/local/lib/libopencv_videoio.so
    libopencv_video.so.4.0 (libc6,x86-64) => /usr/local/lib/libopencv_video.so.4.0
    libopencv_video.so (libc6,x86-64) => /usr/local/lib/libopencv_video.so
```

Finally, we build a simple video playback application found on [Learning OpenCV](#) github page: [example_02-03.cpp](#).

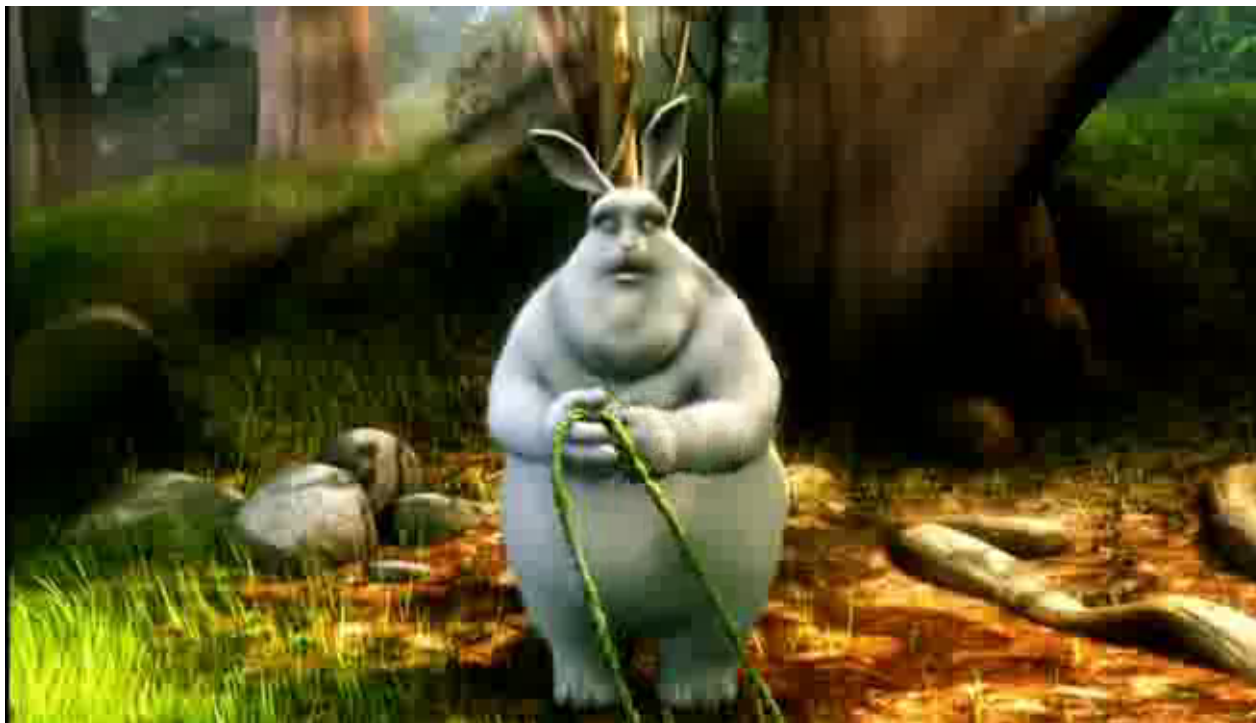
```
$ cd /build
$ git clone https://github.com/oreillymedia/Learning-OpenCV-3_examples.git
$ cd /build/Learning-OpenCV-3_examples
```

We test it with the big_buck_bunny.mpg found in opencv_extra/testdata/highgui/video/.

```
$ cp /build/opencv/opencv_extra/testdata/highgui/video/big_buck_bunny.mpg .
$ ls *.mpg
big_buck_bunny.mpg

$ gcc -g example_02-03.cpp -I/usr/local/include/opencv4 -lstdc++ -lopencv_imgcodecs -lopencv
-lopencv_core -lopencv_imgproc -lopencv_videoio -lopencv_video -lopencv_videostab \
-o example_02-03
```

```
$ ./example_02-03 big_buck_bunny.mpg
```



Screenshot from running example_02_03 with OpenCV

Didn't find an answer to your question? Post your issue below or in our [new FORUM](#), and we'll try our best to help you find a solution.

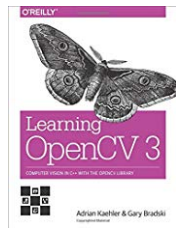
And please note that we update our site daily with new content related to our open source approach to network security and system design. If you would like to be notified about these changes, then please follow us on [Twitter](#) and join our [mailing list](#).

share



subscribe to mailing list:

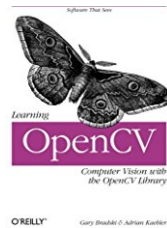
Shop Related Products



Learning OpenCV 3: Computer Vision in C++ with the OpenCV Library

\$56.37 ~~\$84.99~~

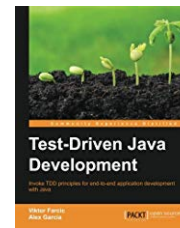
(15)



Learning OpenCV: Computer Vision with the OpenCV Library

\$47.49 ~~\$49.99~~

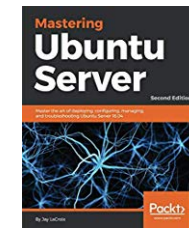
(51)



Test-Driven Java Development

\$49.99

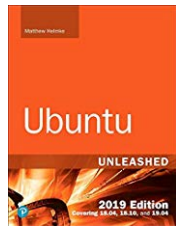
(10)



Mastering Ubuntu Server: Master the art of deploying, configuring, managing, an...

\$44.99

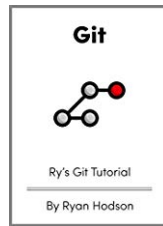
(5)



Ubuntu Unleashed 2019 Edition: Covering 18.04, 18.10, 19.04 (13th Edition)

\$36.96 ~~\$59.99~~

(4)



Ry's Git Tutorial

\$0.00

(208)



Mastering Ubuntu Server: Master the art of deploying, configuring, managing, an...

\$22.39

(5)



Professional Git

\$26.39

(10)

Ads by Amazon

Date: Sept. 12, 2017

Author: HenryC

Comment:

Absolutely fantastic explanation.. it was the missing information that I needed to enable compiling opencv with video support. Using code::blocks also. super thanks

reply

Date: Nov. 30, 2018

Author: Erdem Tuna

Comment:

Thank you a lot for the tutorial. This is the only tutorial that made the opencv workspace work for me. Maybe could you add some more details about the "cmake" building options?

reply

Add a new comment here or reply to one above:

name

email address

your email address will be kept private

comment

Sign in

authenticate with a 3rd party for enhanced features, such as image upload



notify me of a reply

Submit