Install OpenCV3 on Ubuntu

JUNE 6, 2017 BY VAIBHAW SINGH CHANDEL — 156 COMMENTS

In this post, we will provide step by step instructions for installing OpenCV 3 (C++ and Python) on Ubuntu.

> If you are still not able to install OpenCV on your system, but want to get started with it, we suggest using our docker images with pre-installed OpenCV, Dlib, miniconda and jupyter notebooks along with other dependencies as described in this blog.
> You can also use our Installation Script for OpenCV-3 and OpenCV-4 for Ubuntu 16.04 as described in this blog.

Step 1: Update packages

```
sudo apt-get update
sudo apt-get upgrade
```

Step 2: Install OS libraries

```
emove any previous installations of x264</h3>
udo apt-get remove x264 libx264-dev

e will Install dependencies now

udo apt-get install build-essential checkinstall cmake pkg
udo apt-get install git gfortran
udo apt-get install libjpeg8-dev libjasper-dev libpng12-de

 If you are using Ubuntu 14.04
udo apt-get install libtiff4-dev
 If you are using Ubuntu 16.04
udo apt-get install libtiff5-dev

udo apt-get install libavcodec-dev libavformat-dev libswsc
udo apt-get install libxine2-dev libv4l-dev
udo apt-get install libgstreamer0.10-dev libgstreamer-plug
udo apt-get install qt5-default libgtk2.0-dev libtbb-dev
udo apt-get install libatlas-base-dev
udo apt-get install libfaac-dev libmp3lame-dev libtheora-d
udo apt-get install libvorbis-dev libxvidcore-dev
udo apt-get install libopencore-amrnb-dev libopencore-amrw
udo apt-get install x264 v4l-utils

 Optional dependencies
udo apt-get install libprotobuf-dev protobuf-compiler
udo apt-get install libgoogle-glog-dev libgflags-dev
udo apt-get install libgphoto2-dev libeigen3-dev libhdf5-d
```

Step 3: Install Python libraries

```
sudo apt-get install python-dev python-pip python3-de
sudo -H pip2 install -U pip numpy
sudo -H pip3 install -U pip numpy
```

We will use Virtual Environment to install Python libraries. It is generally a good practice in order to separate your project environment and global environment.

```
# Install virtual environment
sudo pip2 install virtualenv virtualenvwrapper
sudo pip3 install virtualenv virtualenvwrapper
echo "# Virtual Environment Wrapper"  >> ~/.bashrc
echo "source /usr/local/bin/virtualenvwrapper.sh" >> ~
source ~/.bashrc

############# For Python 2 ############
# create virtual environment
mkvirtualenv facecourse-py2 -p python2
workon facecourse-py2

# now install python libraries within this virtual env
pip install numpy scipy matplotlib scikit-image scikit

# quit virtual environment
deactivate
##########################################

############ For Python 3 ############
# create virtual environment
mkvirtualenv facecourse-py3 -p python3
workon facecourse-py3

# now install python libraries within this virtual env
pip install numpy scipy matplotlib scikit-image scikit

# quit virtual environment
deactivate
##########################################
```

Step 4: Download OpenCV and OpenCV_contrib

We will download opencv and opencv_contrib packages from their GitHub repositories.

Step 4.1: Download opencv from Github

```
git clone https://github.com/opencv/opencv.git
cd opencv
git checkout 3.3.1
cd ..
```

Step 4.2: Download opencv_contrib from Github

```
git clone https://github.com/opencv/opencv_contrib.gi
cd opencv_contrib
git checkout 3.3.1
cd ..
```

Step 5: Compile and install OpenCV with contrib modules

Step 5.1: Create a build directory

```
cd opencv
mkdir build
cd build
```

Step 5.2: Run CMake

```
cmake -D CMAKE_BUILD_TYPE=RELEASE \
      -D CMAKE_INSTALL_PREFIX=/usr/local \
      -D INSTALL_C_EXAMPLES=ON \
      -D INSTALL_PYTHON_EXAMPLES=ON \
      -D WITH_TBB=ON \
      -D WITH_V4L=ON \
      -D WITH_QT=ON \
      -D WITH_OPENGL=ON \
      -D OPENCV_EXTRA_MODULES_PATH=../../opencv_cont
      -D BUILD_EXAMPLES=ON ..
```

Step 5.3: Compile and Install

```
# find out number of CPU cores in your machine
nproc
# substitute 4 by output of nproc
make -j4
sudo make install
sudo sh -c 'echo "/usr/local/lib" >> /etc/ld.so.conf.
sudo ldconfig
```

Step 5.4: Create symlink in virtual environment

Depending upon Python version you have, paths would be different. OpenCV's Python binary (cv2.so) can be installed either in directory site-packages or dist-packages. Use the following command to find out the correct location on your machine.

```
find /usr/local/lib/ -type f -name "cv2*.so"
```

It should output paths similar to one of these (or two in case OpenCV was compiled for both Python2 and Python3):

```
r Python 2 ############
lled in dist-packages
python2.6/dist-packages/cv2.so
python2.7/dist-packages/cv2.so
lled in site-packages
python2.6/site-packages/cv2.so
python2.7/site-packages/cv2.so

r Python 3 ############
lled in dist-packages
python3.5/dist-packages/cv2.cpython-35m-x86_64-linux-gnu.s
python3.6/dist-packages/cv2.cpython-36m-x86_64-linux-gnu.s
lled in site-packages
python3.5/site-packages/cv2.cpython-35m-x86_64-linux-gnu.s
python3.6/site-packages/cv2.cpython-36m-x86_64-linux-gnu.s
```

Double check the exact path on your machine before running the following commands

```
 Python 2 ############
s/facecourse-py2/lib/python2.7/site-packages
/lib/python2.7/dist-packages/cv2.so cv2.so

 Python 3 ############
s/facecourse-py3/lib/python3.6/site-packages
/lib/python3.6/dist-packages/cv2.cpython-36m-x86_64-linux-g
```

Step 6: Test OpenCV3

We will test a red eye remover application written in OpenCV to test our C++ and Python installations. Download RedEyeRemover.zip and extract it into a folder.

Step 6.1: Test C++ code

Move inside extracted folder, compile and run.

```
# compile
# There are backticks ( ` ) around pkg-config command
g++ -std=c++11 removeRedEyes.cpp `pkg-config --libs -
# run
./removeRedEyes
```

Step 6.2: Test Python code

Activate Python virtual environment

```
############ For Python 2 ############
workon facecourse-py2

############ For Python 3 ############
workon facecourse-py3
```

Quick Check

```
# open ipython (run this command on terminal)
ipython
# import cv2 and print version (run following command
import cv2
print cv2.__version__
# If OpenCV3 is installed correctly,
# above command should give output 3.3.1
# Press CTRL+D to exit ipython
```

Run RedEyeRemover demo

```
python removeRedEyes.py
```

Now you can exit from Python virtual environment.

```
deactivate
```

Whenever you are going to run Python scripts which use OpenCV you should activate the virtual environment we created, using workon command.

Subscribe & Download Code

If you liked this article and would like to download code (C++ and Python) and example images used in all posts in this blog, please subscribe to our newsletter. You will also receive a free Computer Vision Resource Guide. In our newsletter, we share OpenCV tutorials and examples written in C++/Python, and Computer Vision and Machine Learning algorithms and news.

Subscribe Now

FILED UNDER: COMPILE, INSTALL, OPENCV 3 TAGGED WITH: C++, LINUX, OPENCV 3, OPENCV 3.3, OPENCV 3.3.0, PYTHON, UBUNTU

LOAD COMMENTS

I am an entrepreneur with a love for Computer Vision and Machine Learning with a dozen years of experience (and a Ph.D.) in the field.

In 2007, right after finishing my Ph.D., I co-founded TAAZ Inc. with my advisor Dr. David Kriegman and Kevin Barnes. The scalability, and robustness of our computer vision and machine learning algorithms have been put to rigorous test by more than 100M users who have tried our products. Read More...

RECENT POSTS

Xeus-Cling: Run C++ code in Jupyter Notebook

Pose Detection comparison : wrnchAI vs OpenPose

Gender & Age Classification using OpenCV Deep Learning (C++/Python)

Invisibility Cloak using Color Detection and Segmentation with OpenCV

Fast Image Downloader for Open Images V4