

# Installing OpenCV for python 3.X on Ubuntu

In this document, we are going through the instructions for installing OpenCV 3 on Ubuntu. First, check if you have python 3.X installed using one of these commands on the terminal:

```
$ python --version
$ python -V
```

If you have both Python 2.X and 3.X installed you might have to use this command:

```
$ python3 --version
```

By default, Ubuntu 14.04 and 16.04 ship in with Python 2.7 and Python 3.5 and Ubuntu 18.04 has Python 2.7 and Python 3.6 preinstalled. But for any reason, if you don't have it, here is what you have to do to install python 3.6:

## Install Python 3.6 in Ubuntu 14.04 and 16.04

```
$ sudo add-apt-repository ppa:deadsnakes/ppa
$ sudo apt update
$ sudo apt install python3.6
```

## Install Python3 in Ubuntu 18.04 (also works on 16.04)

```
$ sudo apt-get update
$ sudo apt-get install python3
```

We are going to install our desired packages using PIP so now it is time to check if you have PIP installed on your device. To do so use this command:

```
$ pip --version
```

NOTE: If you have both Python 2.X and 3.X installed you may see the pip version for python 2.X so use this command in such situation:

```
$ pip3 --version
```

If you don't have pip installed, use this command to install it:

```
$ sudo apt-get install python3-pip
```

Now we have the first elements to start installing OpenCV, but first, you might want to check if you have correctly installed python 3.X. To do so write this command on terminal:

```
$ python
```

In case of having both versions of python use this command:

```
$ python
```

And you must see something like this which is called python shell:

```
Python 3.5.2 (default, Nov 12 2018, 13:43:14)
[GCC 5.4.0 20160609] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> █
```

Exit the shell area using this script:

```
>>> exit()
```

Now it is time to install OpenCV, so follow the steps below:

## Step 1: Update packages

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

## Step 2: Install OS libraries

```
# Remove any previous installations of x264
$ sudo apt-get remove x264 libx264-dev

# We will Install dependencies now
$ sudo apt-get install build-essential checkinstall cmake pkg-config yasm
$ sudo apt-get install git gfortran
$ sudo apt-get install libjpeg8-dev libjasper-dev libpng12-dev

# If you are using Ubuntu 14.04
```

```

$ sudo apt-get install libtiff4-dev
# If you are using Ubuntu 16.04
$ sudo apt-get install libtiff5-dev

$ sudo apt-get install libavcodec-dev libavformat-dev libswscale-dev libdc1394-22-
dev
$ sudo apt-get install libxine2-dev libv4l-dev
$ sudo apt-get install libgstreamer0.10-dev libgstreamer-plugins-base0.10-dev
$ sudo apt-get install qt5-default libgtk2.0-dev libtbb-dev
$ sudo apt-get install libatlas-base-dev
$ sudo apt-get install libfaac-dev libmp3lame-dev libtheora-dev
$ sudo apt-get install libvorbis-dev libxvidcore-dev
$ sudo apt-get install libopencore-amrnb-dev libopencore-amrwb-dev
$ sudo apt-get install x264 v4l-utils

# Optional dependencies
$ sudo apt-get install libprotobuf-dev protobuf-compiler
$ sudo apt-get install libgoogle-glog-dev libgflags-dev
$ sudo apt-get install libgphoto2-dev libeigen3-dev libhdf5-dev doxygen

```

### Step 3: Install Python libraries

```

$ sudo apt-get install python3-dev python3-pip
$ sudo -H pip3 install -U pip numpy

```

And here are some other libraries use in this class so it's better to have them installed:

```

$ sudo pip3 install -U matplotlib scipy pillow scikit-learn

```

We will install OpenCV in a virtual environment. Using a virtual environment is a good way to separate your project environment from the 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" >> ~/.bashrc
$ source ~/.bashrc
# create virtual environment
$ mkvirtualenv testOne-py3 -p python3
$ workon testOne-py3

# now install python libraries within this virtual environment
$ pip install numpy scipy matplotlib scikit-image scikit-learn ipython

# 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.git
$ 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_contrib/modules \
-D BUILD_EXAMPLES=ON ..
```

### Step 5.3: Compile and Install

```
# find out the 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.d/opencv.conf'
$ sudo ldconfig
```

## Step 5.4: Create symlink in virtual environment

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 something like this(based on your python version):

```
## binary installed in dist-packages
$ /usr/local/lib/python3.5/dist-packages/cv2.cpython-35m-x86_64-linux-gnu.so
/usr/local/lib/python3.6/dist-packages/cv2.cpython-36m-x86_64-linux-gnu.so
## binary installed in site-packages
$ /usr/local/lib/python3.5/site-packages/cv2.cpython-35m-x86_64-linux-gnu.so
$ /usr/local/lib/python3.6/site-packages/cv2.cpython-36m-x86_64-linux-gnu.so
```

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

```
$ cd ~/.virtualenvs/testOne-py3/lib/python3.6/site-packages
$ ln -s /usr/local/lib/python3.6/dist-packages/cv2.cpython-36m-x86_64-linux-gnu.so
cv2.so
```

## Step 6: Test OpenCV3

Now we test if we have done things correct. Activate Python virtual environment:

```
$ workon testOne-py3
$ ipython
```

Write this script and after running it, you must get some output like this:

```
In [1] import cv2
In [2] print(cv2.__version__)
3.3.1
```

Now we are done with the installation and you can exit from python virtual environment

```
$ deactivate
```

NOTE: If you faced any problems in the middle of the process, it is recommended to start from the first step.

Enjoy working with OpenCV!

**Mohammad Akbarizadeh**

References:

- [1] <https://www.learnopencv.com/install-opencv3-on-ubuntu/>
- [2] <https://www.pyimagesearch.com/2016/10/24/ubuntu-16-04-how-to-install-opencv/>
- [3] [https://github.com/opencv/opencv\\_contrib](https://github.com/opencv/opencv_contrib)
- [4] <https://github.com/opencv/opencv>