**Ubuntu 16.04: How to install OpenCV**

*by*[**Adrian Rosebrock**](https://www.pyimagesearch.com/author/adrian/)*on October 24, 2016*



Over the past two years running the PyImageSearch blog, I’ve authored two tutorials detailing the required steps to install OpenCV (with Python bindings) on Ubuntu. You can find the two tutorials here:

* [**Install OpenCV 3.0 and Python 2.7+ on Ubuntu 14.04**](https://pyimagesearch.com/2015/06/22/install-opencv-3-0-and-python-2-7-on-ubuntu/)
* [**Install OpenCV 3.0 and Python 3.4+ on Ubuntu 14.04**](https://pyimagesearch.com/2015/07/20/install-opencv-3-0-and-python-3-4-on-ubuntu/)

However, with *support of Ubuntu 14.04 winding down* and ***Ubuntu 16.04 set as the next LTS*** (with support until April 2021), I thought it would be appropriate to create a ***new, updated Ubuntu + OpenCV install tutorial.***

Inside this tutorial, I will *document*, *demonstrate*, and *provide detailed steps* to install OpenCV 3 on Ubuntu 16.04 with *either* Python 2.7 or Python 3.5 bindings.

Furthermore, this document has been *fully updated* from my previous Ubuntu 14.04 tutorials to use the latest, updated packages from the

apt-get

  repository.

**To learn how to install OpenCV on your Ubuntu 16.04 system, keep reading.**

***Note:****Don’t care about Python bindings and simply want OpenCV installed on your system (likely for C++ coding)? No worries, this tutorial will still work for you. Follow along with the instructions and perform the steps — by the end of this article you’ll have OpenCV installed on your system. From there, just ignore the Python bindings and proceed as usual.*

**Ubuntu 16.04: How to install OpenCV**

Before we get into this tutorial, I want to mention that Ubuntu 16.04 actually ships out-of-the-box with *both* Python 2.7 and Python 3.5 installed. The actual versions (as of 24 October 2016) are:

* Python 2.7.12 (used by default when you type

python

  in your terminal).

* Python 3.5.2 (can be accessed via the

python3

  command).

Again, it’s worth repeating that Python 2.7 is *still* the default Python version used by Ubuntu. There are [**plans to migrate to Python 3**](https://wiki.ubuntu.com/Python) and use Python 3 by default; however, as far as I can tell, we are still a long way from that actually becoming a reality.

In either case, this tutorial will support ***both*** Python 2.7 and Python 3. I’ve highlighted the steps that require you to make a *decision* regarding which version of Python you would like to use. Make sure you are *consistent* with your decision, otherwise you will inevitably run into compile, linking, and import errors.

Regarding *which* Python version you should use…*I’m not getting into that argument*. I’ll simply say that you should use whichever version of Python you are comfortable with and use on a daily basis. Keep in mind that Python 3 is the future — but also keep in mind that porting Python 2.7 code to Python 3 isn’t terribly challenging either once you understand the differences between the Python versions. And as far as OpenCV goes, OpenCV 3 doesn’t care which version of Python you’re using: the bindings will work just the same.

All that said, let’s get started installing OpenCV with Python bindings on Ubuntu 16.04.

**Step #1: Install OpenCV dependencies on Ubuntu 16.04**

Most (in fact, *all*) steps in this tutorial will be accomplished by using your terminal. To start, open up your command line and update the

apt-get

  package manager to refresh and upgrade and pre-installed packages/libraries:

[**[](https://www.pyimagesearch.com/2016/10/24/ubuntu-16-04-how-to-install-opencv/#pyi-pyimagesearch-plus-optin-modal)→ Click here to download the code**](https://www.pyimagesearch.com/2016/10/24/ubuntu-16-04-how-to-install-opencv/#pyi-pyimagesearch-plus-optin-modal)

Ubuntu 16.04: How to install OpenCV

$ sudo apt-get update

$ sudo apt-get upgrade

Next, let’s install some developer tools:

[**[](https://www.pyimagesearch.com/2016/10/24/ubuntu-16-04-how-to-install-opencv/#pyi-pyimagesearch-plus-optin-modal)→ Click here to download the code**](https://www.pyimagesearch.com/2016/10/24/ubuntu-16-04-how-to-install-opencv/#pyi-pyimagesearch-plus-optin-modal)

Ubuntu 16.04: How to install OpenCV

$ sudo apt-get install build-essential cmake pkg-config

The

pkg-config

  package is (very likely) already installed on your system, but be sure to include it in the above

apt-get

  command just in case. The

cmake

  program is used to automatically configure our OpenCV build.

OpenCV is an image processing and computer vision library. Therefore, OpenCV needs to be able to load various image file formats from disk such as JPEG, PNG, TIFF, etc. In order to load these images from disk, OpenCV actually calls other image I/O libraries that actually facilitate the loading and decoding process. We install the necessary ones below:

[**[](https://www.pyimagesearch.com/2016/10/24/ubuntu-16-04-how-to-install-opencv/#pyi-pyimagesearch-plus-optin-modal)→ Click here to download the code**](https://www.pyimagesearch.com/2016/10/24/ubuntu-16-04-how-to-install-opencv/#pyi-pyimagesearch-plus-optin-modal)

Ubuntu 16.04: How to install OpenCV

$ sudo apt-get install libjpeg8-dev libtiff5-dev libjasper-dev libpng12-dev

Okay, so now we have libraries to load images from disk — *but what about video?* Use the following commands to install packages used to process video streams and access frames from cameras:

[**[](https://www.pyimagesearch.com/2016/10/24/ubuntu-16-04-how-to-install-opencv/#pyi-pyimagesearch-plus-optin-modal)→ Click here to download the code**](https://www.pyimagesearch.com/2016/10/24/ubuntu-16-04-how-to-install-opencv/#pyi-pyimagesearch-plus-optin-modal)

Ubuntu 16.04: How to install OpenCV

$ sudo apt-get install libavcodec-dev libavformat-dev libswscale-dev libv4l-dev

$ sudo apt-get install libxvidcore-dev libx264-dev

OpenCV ships out-of-the-box with a very limited set of GUI tools. These GUI tools allow us to display an image to our screen (

cv2.imshow

 ), wait for/record keypresses (

cv2.waitKey

 ), track mouse events (

cv2.setMouseCallback

 ), and create simple GUI elements such as sliders and trackbars. Again, you shouldn’t expect to be building full-fledged GUI applications with OpenCV — these are just simple tools that allow you to debug your code and build very simple applications.

Internally, the name of the module that handles OpenCV GUI operations is

highgui

 . The

highgui

  module relies on the GTK library, which you should install using the following command:

[**[](https://www.pyimagesearch.com/2016/10/24/ubuntu-16-04-how-to-install-opencv/#pyi-pyimagesearch-plus-optin-modal)→ Click here to download the code**](https://www.pyimagesearch.com/2016/10/24/ubuntu-16-04-how-to-install-opencv/#pyi-pyimagesearch-plus-optin-modal)

Ubuntu 16.04: How to install OpenCV

$ sudo apt-get install libgtk-3-dev

Next, we install libraries that are used to optimize various functionalities inside OpenCV, such as matrix operations:

[**[](https://www.pyimagesearch.com/2016/10/24/ubuntu-16-04-how-to-install-opencv/#pyi-pyimagesearch-plus-optin-modal)→ Click here to download the code**](https://www.pyimagesearch.com/2016/10/24/ubuntu-16-04-how-to-install-opencv/#pyi-pyimagesearch-plus-optin-modal)

Ubuntu 16.04: How to install OpenCV

$ sudo apt-get install libatlas-base-dev gfortran

We’ll wrap up Step #1 by installing the Python development headers and libraries for both Python 2.7 and Python 3.5 (that way you have both):

[**[](https://www.pyimagesearch.com/2016/10/24/ubuntu-16-04-how-to-install-opencv/#pyi-pyimagesearch-plus-optin-modal)→ Click here to download the code**](https://www.pyimagesearch.com/2016/10/24/ubuntu-16-04-how-to-install-opencv/#pyi-pyimagesearch-plus-optin-modal)

Ubuntu 16.04: How to install OpenCV

$ sudo apt-get install python2.7-dev python3.5-dev

***Note:****If you do not install the Python development headers and static library, you’ll run into issues during****Step #4****where we run*

*cmake*

*to configure our build. If these headers are not installed, then the*

*cmake*

*command will be unable to automatically determine the proper values of the Python interpreter and Python libraries. In short, the output of this section will look “empty” and you will not be able to build the Python bindings. When you get to****Step #4****, take the time to compare your output of the command to mine.*

**Step #2: Download the OpenCV source**

At the time of this article’s publication, the most recent version of OpenCV is

3.1.0

 , which we download a

.zip

  of and unarchive using the following commands:

[**[](https://www.pyimagesearch.com/2016/10/24/ubuntu-16-04-how-to-install-opencv/#pyi-pyimagesearch-plus-optin-modal)→ Click here to download the code**](https://www.pyimagesearch.com/2016/10/24/ubuntu-16-04-how-to-install-opencv/#pyi-pyimagesearch-plus-optin-modal)

Ubuntu 16.04: How to install OpenCV

$ cd ~

$ wget -O opencv.zip https://github.com/Itseez/opencv/archive/3.1.0.zip

$ unzip opencv.zip

When new versions of OpenCV are released you can check the official **[OpenCV GitHub](https://github.com/opencv/opencv" \t "_blank)** and downloaded the latest release by changing the version number of the

.zip

 .

However, we’re not done downloading source code yet — we also need the **[opencv\_contrib](https://github.com/opencv/opencv_contrib" \t "_blank)** repository as well:

[**[](https://www.pyimagesearch.com/2016/10/24/ubuntu-16-04-how-to-install-opencv/#pyi-pyimagesearch-plus-optin-modal)→ Click here to download the code**](https://www.pyimagesearch.com/2016/10/24/ubuntu-16-04-how-to-install-opencv/#pyi-pyimagesearch-plus-optin-modal)

Ubuntu 16.04: How to install OpenCV

$ wget -O opencv\_contrib.zip https://github.com/Itseez/opencv\_contrib/archive/3.1.0.zip

$ unzip opencv\_contrib.zip

Why are we bothering to download the contrib repo as well?

Well, we want the *full install* of OpenCV 3 to [**have access to features (no pun intended) such as SIFT and SURF**](https://pyimagesearch.com/2015/07/16/where-did-sift-and-surf-go-in-opencv-3/). In OpenCV 2.4, SIFT and SURF were included in the *default* installation of OpenCV. However, with the release of OpenCV 3+, these packages have been moved to contrib, which houses either (1) modules that are currently in development or (2) modules that are marked as “non-free” (i.e., patented). You can learn more about the reasoning behind the SIFT/SURF restructuring in [**this blog post**](https://pyimagesearch.com/2015/07/16/where-did-sift-and-surf-go-in-opencv-3/).

***Note:****You might need to expand the commands above using the “<=>” button during your copy and paste. The*

*.zip*

*in the*

*3.1.0.zip*

*may be cutoff in smaller browser windows. For convenience, I have included the full URL of both the*

*opencv*

*archive as well as the*

*opencv\_contrib*

*archive below:*

* [***https://github.com/Itseez/opencv/archive/3.1.0.zip***](https://github.com/Itseez/opencv/archive/3.1.0.zip)
* [***https://github.com/Itseez/opencv\_contrib/archive/3.1.0.zip***](https://github.com/Itseez/opencv_contrib/archive/3.1.0.zip)

*I also want to mention that****both****your*

*opencv*

*and*

*opencv\_contrib*

*versions should be the same (in this case,*

*3.1.0*

*). If the versions numbers do not matchup, you could very easily run into compile time errors (or worse, runtime errors that are near impossible to debug).*

**Step #3: Setup your Python environment — Python 2.7 or Python 3**

We are now ready to start configuring our Python development environment for the build. The first step is to install

pip

 , a Python package manager:

[**[](https://www.pyimagesearch.com/2016/10/24/ubuntu-16-04-how-to-install-opencv/#pyi-pyimagesearch-plus-optin-modal)→ Click here to download the code**](https://www.pyimagesearch.com/2016/10/24/ubuntu-16-04-how-to-install-opencv/#pyi-pyimagesearch-plus-optin-modal)

Ubuntu 16.04: How to install OpenCV

$ cd ~

$ wget https://bootstrap.pypa.io/get-pip.py

$ sudo python get-pip.py

I’ve mentioned this in *every single OpenCV + Python install tutorial I’ve ever done*, but I’ll say it again here today: I’m a *huge fan* of both **[virtualenv](https://virtualenv.pypa.io/en/latest/" \t "_blank)** and **[virtualenvwrapper](https://virtualenvwrapper.readthedocs.org/en/latest/" \t "_blank)**. **These Python packages allow you to create *separate, independent* Python environments for *each* project that you are working on.**

In short, using these packages allows you to solve the *“Project X depends on version 1.x, but Project Y needs 4.x* dilemma. A fantastic side effect of using Python virtual environments is that you can keep your system Python neat, tidy, and free from clutter.

While you can certainly install OpenCV with Python bindings without Python virtual environments, ***I highly recommend you use them*** as other PyImageSearch tutorials leverage Python virtual environments. I’ll also be assuming that you have both

virtualenv

  and

virtualenvwrapper

  installed throughout the remainder of this guide.

If you would like a full, detailed explanation on why Python virtual environments are a *best practice*, you should absolutely [**give this excellent blog post on RealPython a read**](https://realpython.com/blog/python/python-virtual-environments-a-primer/). I also provide some commentary on why I personally prefer Python virtual environments [**in the first half of this tutorial**](https://pyimagesearch.com/2016/05/02/accessing-rpi-gpio-and-gpio-zero-with-opencv-python/).

Again, let me reiterate that it’s ***standard practice*** in the Python community to be leveraging virtual environments of some sort, so I suggest you do the same:

[**[](https://www.pyimagesearch.com/2016/10/24/ubuntu-16-04-how-to-install-opencv/#pyi-pyimagesearch-plus-optin-modal)→ Click here to download the code**](https://www.pyimagesearch.com/2016/10/24/ubuntu-16-04-how-to-install-opencv/#pyi-pyimagesearch-plus-optin-modal)

Ubuntu 16.04: How to install OpenCV

$ sudo pip install virtualenv virtualenvwrapper

$ sudo rm -rf ~/get-pip.py ~/.cache/pip

Once we have

virtualenv

  and

virtualenvwrapper

  installed, we need to update our

~/.bashrc

  file to include the following lines at the *bottom* of the file:

[**[](https://www.pyimagesearch.com/2016/10/24/ubuntu-16-04-how-to-install-opencv/#pyi-pyimagesearch-plus-optin-modal)→ Click here to download the code**](https://www.pyimagesearch.com/2016/10/24/ubuntu-16-04-how-to-install-opencv/#pyi-pyimagesearch-plus-optin-modal)

Ubuntu 16.04: How to install OpenCV

# virtualenv and virtualenvwrapper

export WORKON\_HOME=$HOME/.virtualenvs

source /usr/local/bin/virtualenvwrapper.sh

The

~/.bashrc

  file is simply a shell script that Bash runs whenever you launch a new terminal. You normally use this file to set various configurations. In this case, we are setting an environment variable called

WORKON\_HOME

  to point to the directory where our Python virtual environments live. We then load any necessary configurations from

virtualenvwrapper

 .

To update your

~/.bashrc

  file simply use a standard text editor. I would recommend using

nano

 ,

vim

 , or

emacs

 . You can also use graphical editors as well, but if you’re just getting started,

nano

  is likely the easiest to operate.

A more simple solution is to use the

cat

  command and avoid editors entirely:

[**[](https://www.pyimagesearch.com/2016/10/24/ubuntu-16-04-how-to-install-opencv/#pyi-pyimagesearch-plus-optin-modal)→ Click here to download the code**](https://www.pyimagesearch.com/2016/10/24/ubuntu-16-04-how-to-install-opencv/#pyi-pyimagesearch-plus-optin-modal)

Ubuntu 16.04: How to install OpenCV

$ echo -e "\n# virtualenv and virtualenvwrapper" >> ~/.bashrc

$ echo "export WORKON\_HOME=$HOME/.virtualenvs" >> ~/.bashrc

$ echo "source /usr/local/bin/virtualenvwrapper.sh" >> ~/.bashrc

After editing our

~/.bashrc

  file, we need to reload the changes:

[**[](https://www.pyimagesearch.com/2016/10/24/ubuntu-16-04-how-to-install-opencv/#pyi-pyimagesearch-plus-optin-modal)→ Click here to download the code**](https://www.pyimagesearch.com/2016/10/24/ubuntu-16-04-how-to-install-opencv/#pyi-pyimagesearch-plus-optin-modal)

Ubuntu 16.04: How to install OpenCV

$ source ~/.bashrc

***Note:****Calling*

*source*

*on*

*.bashrc*

*only has to be done****once****for our current shell session. Anytime we open up a new terminal, the contents of*

*.bashrc*

*will be****automatically****executed (including our updates).*

Now that we have installed

virtualenv

  and

virtualenvwrapper

 , the next step is to actually *create* the Python virtual environment — we do this using the

mkvirtualenv

  command.

But before executing this command, you need to make a choice: Do you want to use Python 2.7 or Python 3?

The outcome of your choice will determine which command you run in the following section.

**Creating your Python virtual environment**

If you decide to use Python 2.7, use the following command to create a Python 2.7 virtual environment:

[**[](https://www.pyimagesearch.com/2016/10/24/ubuntu-16-04-how-to-install-opencv/#pyi-pyimagesearch-plus-optin-modal)→ Click here to download the code**](https://www.pyimagesearch.com/2016/10/24/ubuntu-16-04-how-to-install-opencv/#pyi-pyimagesearch-plus-optin-modal)

Ubuntu 16.04: How to install OpenCV

$ mkvirtualenv cv -p python2

Otherwise, use this command to create a Python 3 virtual environment:

[**[](https://www.pyimagesearch.com/2016/10/24/ubuntu-16-04-how-to-install-opencv/#pyi-pyimagesearch-plus-optin-modal)→ Click here to download the code**](https://www.pyimagesearch.com/2016/10/24/ubuntu-16-04-how-to-install-opencv/#pyi-pyimagesearch-plus-optin-modal)

Ubuntu 16.04: How to install OpenCV

$ mkvirtualenv cv -p python3

Regardless of which Python command you decide to use, the end result is that we have created a Python virtual environment named

cv

  (short for “computer vision”).

You can name this virtual environment whatever you like (and create as many Python virtual environments as you want), but for the time bing, I would suggest sticking with the

cv

  name as that is what I’ll be using throughout the rest of this tutorial.

**Verifying that you are in the “cv” virtual environment**

If you ever reboot your Ubuntu system; log out and log back in; or open up a new terminal, you’ll need to use the

workon

  command to re-access your

cv

  virtual environment. An example of the

workon

  command follows:

[**[](https://www.pyimagesearch.com/2016/10/24/ubuntu-16-04-how-to-install-opencv/#pyi-pyimagesearch-plus-optin-modal)→ Click here to download the code**](https://www.pyimagesearch.com/2016/10/24/ubuntu-16-04-how-to-install-opencv/#pyi-pyimagesearch-plus-optin-modal)

Ubuntu 16.04: How to install OpenCV

$ workon cv

To validate that you are in the

cv

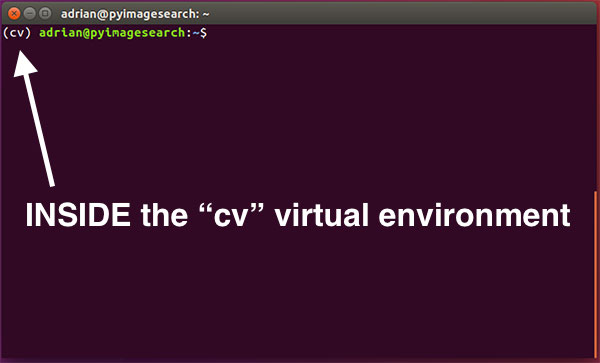
  virtual environment, simply examine your command line — *if you see the text*

*(cv)*

*preceding your prompt, then you****are****in the*

*cv*

*virtual environment:*

**Figure 1:** Make sure you see the *“(cv)”* text on your prompt, indicating that you ***are*** in the *cv* virtual environment.

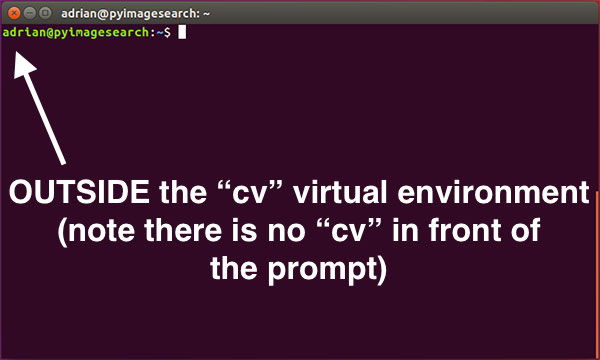
Otherwise, if you ***do not*** see the

cv

  text, then you ***are not*** in the

cv

  virtual environment:

**Figure 2:** If you do not see the *“(cv)”* text on your prompt, then you ***are not*** in the *cv* virtual environment and need to run the *“workon”* command to resolve this issue.

To access the

cv

  virtual environment simply use the

workon

  command mentioned above.

**Install NumPy into your Python virtual environment**

The final step before we compile OpenCV is to install **[NumPy](http://www.numpy.org/" \t "_blank)**, a Python package used for numerical processing. To install NumPy, ensure you are in the

cv

  virtual environment (otherwise NumPy will be installed into the *system*version of Python rather than the

cv

  environment). From there execute the following command:

[**[](https://www.pyimagesearch.com/2016/10/24/ubuntu-16-04-how-to-install-opencv/#pyi-pyimagesearch-plus-optin-modal)→ Click here to download the code**](https://www.pyimagesearch.com/2016/10/24/ubuntu-16-04-how-to-install-opencv/#pyi-pyimagesearch-plus-optin-modal)

Ubuntu 16.04: How to install OpenCV

$ pip install numpy

**Step #4: Configuring and compiling OpenCV on Ubuntu 16.04**

At this point, all of our necessary prerequisites have been installed — we are now ready to compile and OpenCV!

But before we do that, double-check that you are in the

cv

  virtual environment by examining your prompt (you should see the

(cv)

  text preceding it), and if not, use the

workon

  command:

[**[](https://www.pyimagesearch.com/2016/10/24/ubuntu-16-04-how-to-install-opencv/#pyi-pyimagesearch-plus-optin-modal)→ Click here to download the code**](https://www.pyimagesearch.com/2016/10/24/ubuntu-16-04-how-to-install-opencv/#pyi-pyimagesearch-plus-optin-modal)

Ubuntu 16.04: How to install OpenCV

$ workon cv

After ensuring you are in the

cv

  virtual environment, we can setup and configure our build using CMake:

[**[](https://www.pyimagesearch.com/2016/10/24/ubuntu-16-04-how-to-install-opencv/#pyi-pyimagesearch-plus-optin-modal)→ Click here to download the code**](https://www.pyimagesearch.com/2016/10/24/ubuntu-16-04-how-to-install-opencv/#pyi-pyimagesearch-plus-optin-modal)

Ubuntu 16.04: How to install OpenCV

$ cd ~/opencv-3.1.0/

$ mkdir build

$ cd build

$ cmake -D CMAKE\_BUILD\_TYPE=RELEASE \

-D CMAKE\_INSTALL\_PREFIX=/usr/local \

-D INSTALL\_PYTHON\_EXAMPLES=ON \

-D INSTALL\_C\_EXAMPLES=OFF \

-D OPENCV\_EXTRA\_MODULES\_PATH=~/opencv\_contrib-3.1.0/modules \

-D PYTHON\_EXECUTABLE=~/.virtualenvs/cv/bin/python \

-D BUILD\_EXAMPLES=ON ..

The above commands change directory to

~/opencv-3.1.0

 , which if you have been following this tutorial is where you downloaded and unarchived the

.zip

  files.

***Note:****If you are getting an error related to*

*stdlib.h: No such file or directory*

*during either the*

*cmake*

*or*

*make*

*phase of this tutorial you’ll also need to include the following option to CMake:*

*-D ENABLE\_PRECOMPILED\_HEADERS=OFF*

*. In this case I would suggest deleting your*

*build*

*directory, re-creating it, and then re-running CMake with the above option included. This will resolve the*

*stdlib.h*

*error. Thank you to Carter Cherry and Marcin for pointing this out in the comments section!*

Inside this directory we create a sub-directory named

build

  and change into it. The

build

  directory is where the actual compile is going to take place.

Finally, we execute

cmake

  to configure our build.

Before we move on to the actual compilation of OpenCV, ***make sure you examine the output of CMake!***

To do this, scroll down the section titled

Python 2

  and

Python 3

 .

***If you are compiling OpenCV on Ubuntu 16.04 with Python 2.7 support***, make sure the

Python 2

  section includes valid paths to the

Interpreter

 ,

Libraries

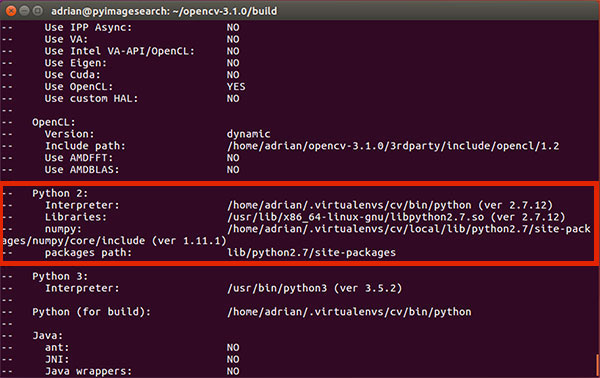
 ,

numpy

 , and

packages path

 . Your output should be similar to mine below:

**Figure 3:** Ensuring that Python 2.7 will be used when compiling OpenCV 3 for Ubuntu 16.04.

Examining this output, you can see that:

1. The

Interpreter

  points to the Python 2.7 binary in the

cv

  virtual environment.

1. Libraries

  points to the Python 2.7 library (which we installed during the final step of Step #1).

1. The

numpy

  value points to our NumPy installation in the

cv

  virtual environment.

1. And finally, the

packages path

  points to

lib/python2.7/site-packages

 . When combined with the

CMAKE\_INSTALL\_PREFIX

 , this means that after compiling OpenCV, we’ll find our

cv2.so

  bindings in

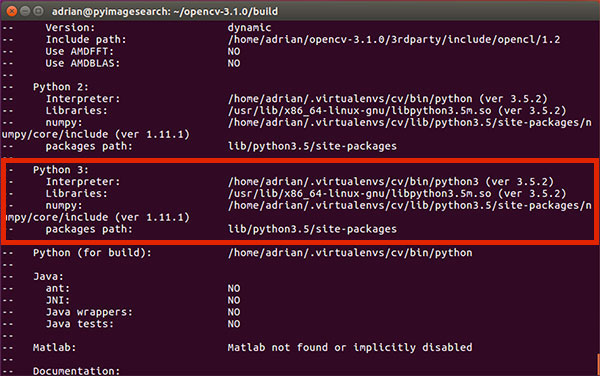
/usr/local/lib/python2.7/site-packages/

 .

Similarly, ***if you’re compiling OpenCV 16.04 with Python 3 support***, you’ll want to ensure your

Python 3

  section looks similar to mine below:

**Figure 4:** Validating that Python 3 will be used when compiling OpenCV 3 for Ubuntu 16.04.

Again, notice how my

Interpreter

 ,

Libraries

 ,

numpy

  and

packages path

  have all been correctly set.

If you ***do not*** see the

cv

  virtual environments in these variable paths, ***it’s almost certainly because you are NOT in the***

***cv***

***virtual environment prior to running CMake!***

If that is indeed the case, simply access the

cv

  virtual environment by calling

workon cv

  and re-run the CMake command mentioned above.

Assuming your CMake command exited without any errors, you can now compile OpenCV:

[**[](https://www.pyimagesearch.com/2016/10/24/ubuntu-16-04-how-to-install-opencv/#pyi-pyimagesearch-plus-optin-modal)→ Click here to download the code**](https://www.pyimagesearch.com/2016/10/24/ubuntu-16-04-how-to-install-opencv/#pyi-pyimagesearch-plus-optin-modal)

Ubuntu 16.04: How to install OpenCV

$ make -j4

The

-j

  switch controls the number of processes to be used when compiling OpenCV — you’ll want to set this value to the number of processors/cores on your machine. In my case, I have a quad-core processor, so I set

-j4

 .

Using multiple processes allows OpenCV to compile faster; however, there are times where race conditions are hit and the compile bombs out. While you can’t really tell if this is the case without *a lot* of previous experience compiling OpenCV, if you do run into an error, my first suggestion would be to run

make clean

  to flush the build, followed by compiling using only a single core:

[**[](https://www.pyimagesearch.com/2016/10/24/ubuntu-16-04-how-to-install-opencv/#pyi-pyimagesearch-plus-optin-modal)→ Click here to download the code**](https://www.pyimagesearch.com/2016/10/24/ubuntu-16-04-how-to-install-opencv/#pyi-pyimagesearch-plus-optin-modal)

Ubuntu 16.04: How to install OpenCV

$ make clean

$ make

Below you can find a screenshot of a successful OpenCV + Python compile on Ubuntu 16.04:

**Figure 5:** Successfully compiling OpenCV 3 for Ubuntu 16.04.

The last step is to actually *install* OpenCV 3 on Ubuntu 16.04:

[**[](https://www.pyimagesearch.com/2016/10/24/ubuntu-16-04-how-to-install-opencv/#pyi-pyimagesearch-plus-optin-modal)→ Click here to download the code**](https://www.pyimagesearch.com/2016/10/24/ubuntu-16-04-how-to-install-opencv/#pyi-pyimagesearch-plus-optin-modal)

Ubuntu 16.04: How to install OpenCV

$ sudo make install

$ sudo ldconfig

**Step #5: Finish your OpenCV install**

You’re coming down the home stretch, just a few more steps to go and your Ubuntu 16.04 system will be all setup with OpenCV 3.

**For Python 2.7:**

After running

sudo make install

 , your Python 2.7 bindings for OpenCV 3 should now be located in

/usr/local/lib/python-2.7/site-packages/

 . You can verify this using the

ls

  command:

[**[](https://www.pyimagesearch.com/2016/10/24/ubuntu-16-04-how-to-install-opencv/#pyi-pyimagesearch-plus-optin-modal)→ Click here to download the code**](https://www.pyimagesearch.com/2016/10/24/ubuntu-16-04-how-to-install-opencv/#pyi-pyimagesearch-plus-optin-modal)

Ubuntu 16.04: How to install OpenCV

$ ls -l /usr/local/lib/python2.7/site-packages/

total 1972

-rw-r--r-- 1 root staff 2016608 Sep 15 09:11 cv2.so

***Note:****In some cases, you may find that OpenCV was installed in*

*/usr/local/lib/python2.7/dist-packages*

*rather than*

*/usr/local/lib/python2.7/site-packages*

*(note*

*dist-packages*

*versus*

*site-packages*

*). If your*

*cv2.so*

*bindings are not in the*

*site-packages*

*directory, be sure to check*

*dist-pakages*

*.*

The final step is to **[sym-link](https://en.wikipedia.org/wiki/Symbolic_link" \t "_blank)** our OpenCV

cv2.so

  bindings into our

cv

  virtual environment for Python 2.7:

[**[](https://www.pyimagesearch.com/2016/10/24/ubuntu-16-04-how-to-install-opencv/#pyi-pyimagesearch-plus-optin-modal)→ Click here to download the code**](https://www.pyimagesearch.com/2016/10/24/ubuntu-16-04-how-to-install-opencv/#pyi-pyimagesearch-plus-optin-modal)

Ubuntu 16.04: How to install OpenCV

$ cd ~/.virtualenvs/cv/lib/python2.7/site-packages/

$ ln -s /usr/local/lib/python2.7/site-packages/cv2.so cv2.so

**For Python 3.5:**

After running

sudo make install

 , your OpenCV + Python 3 bindings should be located in

/usr/local/lib/python3.5/site-packages/

 . Again, you can verify this using the

ls

  command:

[**[](https://www.pyimagesearch.com/2016/10/24/ubuntu-16-04-how-to-install-opencv/#pyi-pyimagesearch-plus-optin-modal)→ Click here to download the code**](https://www.pyimagesearch.com/2016/10/24/ubuntu-16-04-how-to-install-opencv/#pyi-pyimagesearch-plus-optin-modal)

Ubuntu 16.04: How to install OpenCV

$ ls -l /usr/local/lib/python3.5/site-packages/

total 1972

-rw-r--r-- 1 root staff 2016816 Sep 13 17:24 cv2.cpython-35m-x86\_64-linux-gnu.so

I’ve been puzzled regarding this behavior ever since OpenCV 3 was released, but for some reason, when compiling OpenCV with Python 3 support, the output

cv2.so

  filename is different. The actual filename might vary for you, but it should look something similar to

cv2.cpython-35m-x86\_64-linux-gnu.so

 .

Again, I have no idea exactly *why* this happens, but it’s a very easy fix. All we need to do is rename the file:

[**[](https://www.pyimagesearch.com/2016/10/24/ubuntu-16-04-how-to-install-opencv/#pyi-pyimagesearch-plus-optin-modal)→ Click here to download the code**](https://www.pyimagesearch.com/2016/10/24/ubuntu-16-04-how-to-install-opencv/#pyi-pyimagesearch-plus-optin-modal)

Ubuntu 16.04: How to install OpenCV

$ cd /usr/local/lib/python3.5/site-packages/

$ sudo mv cv2.cpython-35m-x86\_64-linux-gnu.so cv2.so

After renaming

cv2.cpython-35m-x86\_64-linux-gnu.so

  to simply

cv2.so

 , we can sym-link our OpenCV bindings into the

cv

  virtual environment for Python 3.5:

[**[](https://www.pyimagesearch.com/2016/10/24/ubuntu-16-04-how-to-install-opencv/#pyi-pyimagesearch-plus-optin-modal)→ Click here to download the code**](https://www.pyimagesearch.com/2016/10/24/ubuntu-16-04-how-to-install-opencv/#pyi-pyimagesearch-plus-optin-modal)

Ubuntu 16.04: How to install OpenCV

$ cd ~/.virtualenvs/cv/lib/python3.5/site-packages/

$ ln -s /usr/local/lib/python3.5/site-packages/cv2.so cv2.so

**Step #6: Testing your OpenCV install**

**Congratulations, you now have OpenCV 3 installed on your Ubuntu 16.04 system!**

To verify that your installation is working:

1. Open up a new terminal.
2. Execute the

workon

  command to access the

cv

  Python virtual environment.

1. Attempt to import the Python + OpenCV bindings.

I have demonstrated how to perform these steps below:

[**[](https://www.pyimagesearch.com/2016/10/24/ubuntu-16-04-how-to-install-opencv/#pyi-pyimagesearch-plus-optin-modal)→ Click here to download the code**](https://www.pyimagesearch.com/2016/10/24/ubuntu-16-04-how-to-install-opencv/#pyi-pyimagesearch-plus-optin-modal)

Ubuntu 16.04: How to install OpenCV

$ cd ~

$ workon cv

$ python

Python 3.5.2 (default, Jul 5 2016, 12:43:10)

[GCC 5.4.0 20160609] on linux

Type "help", "copyright", "credits" or "license" for more information.

>>> import cv2

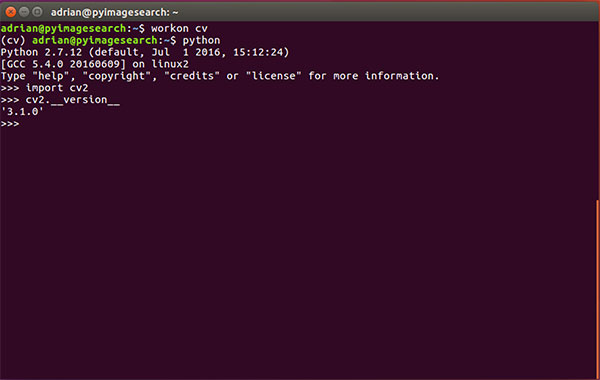
>>> cv2.\_\_version\_\_

'3.1.0'

>>>

As you can see, I can import my OpenCV bindings into my Python 3.5 shell.

Below follows a screenshot of me utilizing the same steps outlined in this tutorial and importing OpenCV bindings into a Python 2.7 shell:

**Figure 6:** Ensuring that I can successfully import my Python + OpenCV bindings on Ubuntu 16.04.

Thus, regardless of which Python version you decide to use, simply follow the steps detailed in this tutorial and you’ll be able to install OpenCV + Python on your Ubuntu 16.04 system.

Once OpenCV has been installed, you can delete both the

opencv-3.1.0

  and

opencv\_contrib-3.1.0

  directories (along with their associated

.zip

  files):

[**[](https://www.pyimagesearch.com/2016/10/24/ubuntu-16-04-how-to-install-opencv/#pyi-pyimagesearch-plus-optin-modal)→ Click here to download the code**](https://www.pyimagesearch.com/2016/10/24/ubuntu-16-04-how-to-install-opencv/#pyi-pyimagesearch-plus-optin-modal)

Ubuntu 16.04: How to install OpenCV

$ cd ~

$ rm -rf opencv-3.1.0 opencv\_contrib-3.1.0 opencv.zip opencv\_contrib.zip

But again, *be careful when running this command!* You’ll want to make sure you have properly installed OpenCV on your system prior to blowing along these directories. Otherwise, you’ll need to restart the entire compile process!

**Troubleshooting and FAQ**

In this section, I address some of the common questions, problems, and issues that arise when installing OpenCV on Ubuntu 16.04.

***Q.*** When I execute

mkvirtualenv

  or

workon

 , I get a “command not found error”.

***A.*** There are three primary reasons why you would be getting this error message, all of which are related to **Step #3:**

1. First, make sure you have installed

virtualenv

  and

virtualenvwrapper

  using the

pip

  package manager. You can verify this by running

pip freeze

 , examining the output, and ensuring that you see both

virtualenv

  and

virtualenvwrapper

  in the list of installed packages.

1. Your

~/.bashrc

  file may not be updated correctly. To diagnose this, use a text editor such as

nano

  and view the contents of your

~/.bashrc

  file. At the bottom of the file, you should see the proper

export

  and

source

  commands are present (again, check **Step #3** for the commands that should be appended to

~/.bashrc

 ).

1. After editing your

~/.bashrc

  file, you may have forgotten to

source

  it and reload its contents. Make sure you run source

~/.bashrc

  after editing it to ensure the contents are reloaded — this will give you access to the

mkvirtualenv

  and

workon

  commands.

***Q.*** Whenever I open a new terminal, logout, or reboot my Ubuntu system, I cannot execute the

mkvirtualenv

  or

workon

  commands.

***A.*** See **reason #2** from the previous question.

***Q.***When I (1) open up a Python shell that imports OpenCV or (2) execute a Python script that calls OpenCV, I get an error:

Import Error: No module named cv2

 .

***A.*** Unfortunately, the exact cause of this error message is extremely hard to diagnose as there are *multiple* reasons this could be happening. In general, I recommend the following suggestions to help diagnose and resolve the error:

1. Make sure you are in the

cv

  virtual environment by using the

workon cv

  command. If this command gives you an error, then see the first question in this FAQ.

1. If after you’ve ensured your

~/.bashrc

  file has been updated properly and

source

 ‘d, then try investigating the contents of the

site-packages

  directory in your

cv

  virtual environment. You can find the

site-packages

  directory in

~/.virtualenvs/cv/lib/python2.7/site-packages/

  or

~/.virtualenvs/cv/lib/python3.5/site-packages/

  depending on your Python version. Make sure that (1) there is a

cv2.so

  file in this

site-packages

  directory and (2) that it’s properly sym-linked to a valid, existing file.

1. Be sure to check the

site-packages

  (and even

dist-packages

 ) directory for the system install of Python located in

/usr/local/lib/python2.7/site-packages/

  and

/usr/local/lib/python3.5/site-packages/

 , respectively. Ideally, you should have a

cv2.so

  file there.

1. If all else fails, check in your

build/lib

  directory of your OpenCV build. There *should* be a

cv2.so

  file there (provided that both

cmake

  and

make

  executed without error). If the

cv2.so

  file is present, *manually copy it* into both the system

site-packages

  directory as well as the

site-packages

  directory for the

cv

  virtual environment.

**So, what’s next?**

Congrats! You now have a brand new, fresh install of OpenCV on your Ubuntu 16.04 system — and I’m sure you’re just itching to leverage your install to build some awesome computer vision apps…

…but I’m also willing to bet that *you’re just getting started learning computer vision and OpenCV*, and probably feeling a bit confused and overwhelmed on exactly where to start.

Personally, I’m a big fan of **learning by example**, so a good first step would be to have some fun and [**read this blog post on detecting cats in images/videos**](https://pyimagesearch.com/2016/06/20/detecting-cats-in-images-with-opencv/). This tutorial is meant to be *very hands-on* and demonstrate how you can (quickly) build a Python + OpenCV application to detect the presence of cats in images.

And if you’re *really interested* in leveling-up your computer vision skills, you should definitely check out my book, [***Practical Python and OpenCV + Case Studies***](https://pyimagesearch.com/practical-python-opencv/). My book not only *covers the basics of computer vision and image processing*, but also teaches you how to solve real-world computer vision problems including ***face detection in images and video streams***, ***object tracking in video***, and ***handwriting recognition***.

**[](https://pyimagesearch.com/practical-python-opencv/)**

So, let’s put that fresh install of OpenCV 3 on your Ubuntu 16.04 system to good use — [***just click here to learn more about the real-world projects you can solve using Practical Python and OpenCV.***](https://pyimagesearch.com/practical-python-opencv/)

**Summary**

In today’s blog post, I demonstrated how to install ***OpenCV 3*** with either Python 2.7 or Python 3 bindings on your ***Ubuntu 16.04 system***.

For more OpenCV install tutorials on other operating systems (such as OSX, Raspbian, etc.), [**please refer to this page**](https://pyimagesearch.com/opencv-tutorials-resources-guides/) where I provide additional links and resources.

***But before you go…***

If you’re interested in learning more about OpenCV, computer vision, and image processing, ***be sure to enter your email address in the form below to be notified when new blog posts are published!***



**Join the PyImageSearch Newsletter and Grab My FREE 17-page Resource Guide PDF**

Enter your email address below to **join the PyImageSearch Newsletter** and **download my FREE 17-page Resource Guide PDF** on Computer Vision, OpenCV, and Deep Learning.

Top of Form

DOWNLOAD THE PDF!

Bottom of Form



**About the Author**

Hi there, I’m Adrian Rosebrock, PhD. All too often I see developers, students, and researchers wasting their time, studying the wrong things, and generally struggling to get started with Computer Vision, Deep Learning, and OpenCV. I created this website to show you what I believe is the best possible way to get your start.