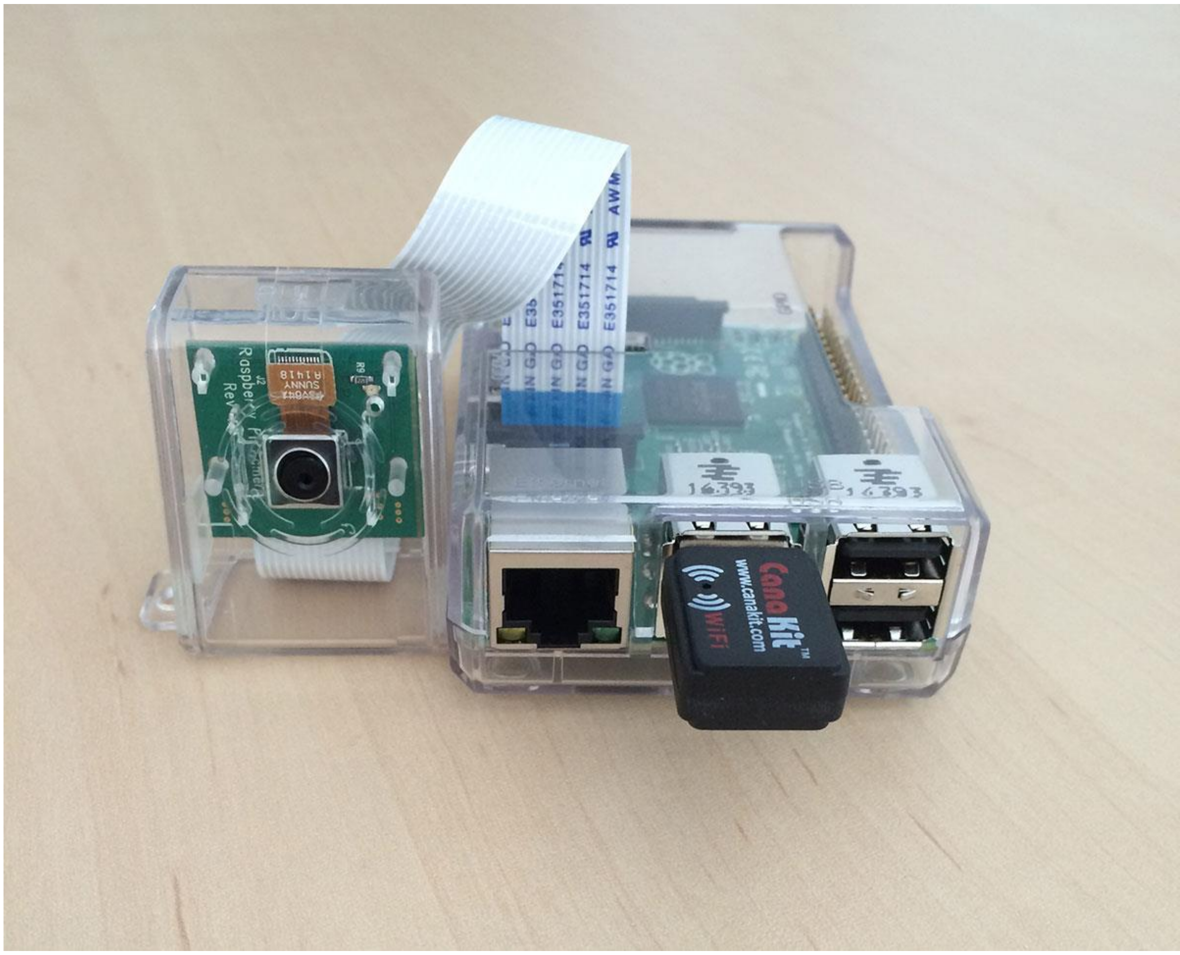

## Step 0:

Again, I’m going to assume that you have just unboxed your Raspberry Pi 2/B+. Open up a terminal and we’ll start by updating and upgrading installed packages, followed by updating the Raspberry Pi firmware:

Install OpenCV and Python your Raspberry Pi 2 and B+

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

## Step 1:

Install the required developer tools and packages:

Install OpenCV and Python your Raspberry Pi 2 and B+

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

Both build-essential and pkg-config are likely already installed, but just in case they are not, be sure to include them in your apt-get command.

**Timings:**

**Raspberry Pi B+:** < 2 minutes

**Raspberry Pi 2:** < 40 seconds

## Step 2:

Install the necessary image I/O packages. These packages allow you to load various image file formats such as JPEG, PNG, TIFF, etc.

Install OpenCV and Python your Raspberry Pi 2 and B+

```
$ sudo apt-get install libjpeg8-dev libtiff4-dev libjasper-dev libpng12-dev
```

**Timings:**

**Raspberry Pi B+:** < 5 minutes

**Raspberry Pi 2:** < 30 seconds

## Step 3:

Install the GTK development library. This library is used to build Graphical User Interfaces (GUIs) and is required for the highgui library of OpenCV which allows you to view images on your screen:

Install OpenCV and Python your Raspberry Pi 2 and B+

```
$ sudo apt-get install libgtk2.0-dev
```

**Timings:**

**Raspberry Pi B+:** < 10 minutes

**Raspberry Pi 2:** < 3 minutes

## Step 4:

Install the necessary video I/O packages. These packages are used to load video files using OpenCV:

Install OpenCV and Python your Raspberry Pi 2 and B+

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

**Timings:**

**Raspberry Pi B+:** < 5 minutes

**Raspberry Pi 2:** < 30 seconds

## Step 5:

Install libraries that are used to optimize various operations within OpenCV:

Install OpenCV and Python your Raspberry Pi 2 and B+

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

**Timings:**

**Raspberry Pi B+:** < 2 minutes

**Raspberry Pi 2:** < 30 seconds

## Step 6:

Install pip :
Install OpenCV and Python your Raspberry Pi 2 and B+

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

**Timings:**

**Raspberry Pi B+:** < 2 minutes

**Raspberry Pi 2:** < 30 seconds

## Step 7:

Install virtualenv and virtualenvwrapper :
Install OpenCV and Python your Raspberry Pi 2 and B+

```
$ sudo pip install virtualenv virtualenvwrapper
$ sudo rm -rf ~/.cache/pip
```

Then, update your ~/.profile file to include the following lines:
Install OpenCV and Python your Raspberry Pi 2 and B+

```
# virtualenv and virtualenvwrapper
export WORKON_HOME=$HOME/.virtualenvs
source /usr/local/bin/virtualenvwrapper.sh
```

Reload your .profile file:
Install OpenCV and Python your Raspberry Pi 2 and B+

```
$ source ~/.profile
```

Create your computer vision virtual environment:

```
$ mkvirtualenv cv
```

```
$ mkvirtualenv cv
```

**Timings:**

**Raspberry Pi B+:** < 2 minutes

**Raspberry Pi 2:** < 2 minutes

## Step 8:

Now we can install the Python 2.7 development tools:

Install OpenCV and Python your Raspberry Pi 2 and B+

```
$ sudo apt-get install python2.7-dev
```

***Note:*** *Yes, we are going to use Python 2.7. OpenCV 2.4.X does not yet support Python 3 and OpenCV 3.0 is still in beta. It's also unclear when the Python bindings for OpenCV 3.0 will be complete so I advise to stick with OpenCV 2.4.X for the time being.*
We also need to install NumPy since the OpenCV Python bindings represent images as multi-dimensional NumPy arrays:

Install OpenCV and Python your Raspberry Pi 2 and B+

```
$ pip install numpy
```

**Timings:**
**Raspberry Pi B+:** < 45 minutes
**Raspberry Pi 2:** < 15 minutes

## Step 9:

Download OpenCV and unpack it:

Install OpenCV and Python your Raspberry Pi 2 and B+

```
$ wget -O opencv-2.4.10.zip http://sourceforge.net/projects/opencvlibrary/files/opencv-unix/2.4.10/opencv-2.4.10.zip/download
$ unzip opencv-2.4.10.zip
$ cd opencv-2.4.10
```

Setup the build:

Install OpenCV and Python your Raspberry Pi 2 and B+

```
$ mkdir build
$ cd build
$ cmake -D CMAKE_BUILD_TYPE=RELEASE -D CMAKE_INSTALL_PREFIX=/usr/local -D BUILD_NEW_PYTHON_SUPPORT=ON -D INSTALL_C_EXAMPLES=ON -D INSTALL_PYTHON_EXAMPLES=ON  -D BUILD_EXAMPLES=ON ..
```

**Timings:**
**Raspberry Pi B+:** < 3 minutes
**Raspberry Pi 2:** < 1.5 minutes
Compile OpenCV:

Install OpenCV and Python your Raspberry Pi 2 and B+

```
$ make
```

***Important:*** *Make sure you’re in the* cv *virtual environment so OpenCV is compiled against the virtual environment Python and NumPy. Otherwise, OpenCV will be compiled against the system Python and NumPy which can lead to problems down the line.*

**Timings:**

**Raspberry Pi B+:** < 9.5 hours

**Raspberry Pi 2:** < 2.8 hours

Finally, we can install OpenCV:

Install OpenCV and Python your Raspberry Pi 2 and B+

```
$ sudo make install
$ sudo ldconfig
```

**Timings:**

**Raspberry Pi B+:** < 3 minutes

**Raspberry Pi 2:** < 1 minute

## Step 10:

If you’ve gotten this far in the guide, OpenCV should now be installed in /usr/local/lib/python2.7/site-packages

But in order to utilize OpenCV within our cv virtual environment, we first need to sym-link OpenCV into our site-packages directory:

Install OpenCV and Python your Raspberry Pi 2 and B+

```
$ cd ~/.virtualenvs/cv/lib/python2.7/site-packages/
$ ln -s /usr/local/lib/python2.7/site-packages/cv2.so cv2.so
$ ln -s /usr/local/lib/python2.7/site-packages/cv.py cv.py
```

## Step 11:

Finally, we can give our OpenCV and Python installation a test drive:

Install OpenCV and Python your Raspberry Pi 2 and B+

```
$ workon cv
$ python
>>> import cv2
>>> cv2.__version__
'2.4.10'
```

OpenCV and Python is now successfully installed on your Raspberry Pi!