

# **Handout: Workshop on Python and OpenCV on Raspberry Pi**

## **Installing Python, Opencv and Anaconda in Ubuntu**

### **Python:**

1. Make sure we are in our home directory  
`cd ~`
2. Install Tools:  
`sudo apt-get install build-essential cmake git pkg-config`
3. Install File Format Libs  
`sudo apt-get install libjpeg8-dev libtiff5-dev libjasper-dev libpng12-dev`
4. Install Image -> GUI handler  
`sudo apt-get install libgtk2.0-dev`
5. Just in case we want to do any Video Processing:  
`sudo apt-get install libavcodec-dev libavformat-dev libswscale-dev libv4l-dev`
6. Install OpenCV Optimisation Libs  
`sudo apt-get install libatlas-base-dev gfortran`
7. Install Python 2.7  
`sudo -H apt-get install python2.7-dev`

### **Opencv**

8. `mkdir OpenCV-tmp`
9. `cd OpenCV-tmp`
10. `git clone https://github.com/Itseez/opencv.git`
11. `mv opencv opencv-3`
12. `mkdir build`
13. `cd build`
14. `cmake -D CMAKE_BUILD_TYPE=RELEASE -D CMAKE_INSTALL_PREFIX=/usr/local ..../opencv-3`
15. `make`
16. `sudo make install`

### **Anaconda**

17. `wget https://3230d63b5fc54e62148ec95ac804525aac4b6dba79b00b39d1d3.ssl.cf1.rackcdn.com/Anaconda-2.3.0-Linux-x86.sh`
18. `bash Anaconda-2.3.0-Linux-x86.sh`  
Note: Type yes when prompted every time.
19. `conda install opencv`

### **To check installation**

20. Python
21. `import cv2`

```
22. cv2.__version__  
23. import numpy  
24. import matplotlib
```

Note: There should not be any errors when import command is executed.

## Installing Python, Opencv and Anaconda in Windows

1. Below Python packages are to be downloaded and installed to their default locations.

1.1. Python-2.7.x. (<http://python.org/ftp/python/2.7.5/python-2.7.5.msi>)

1.2. Numpy. (<https://sourceforge.net/projects/numpy/files/NumPy/1.7.1/numpy-1.7.1-win32-superpack-python2.7.exe/download>)

1.3. Matplotlib ([https://sourceforge.net/projects/matplotlib/files/matplotlib/matplotlib-1.3.0/matplotlib-1.3.0.win32-py2.7.exe/download?use\\_mirror=excellmedia](https://sourceforge.net/projects/matplotlib/files/matplotlib/matplotlib-1.3.0/matplotlib-1.3.0.win32-py2.7.exe/download?use_mirror=excellmedia))

Install all packages into their default locations. Python will be installed to **C:/Python27/**.

2. After installation, open Python IDLE. Enter `import numpy` and make sure Numpy is working fine.

3. Download latest OpenCV release from <https://sourceforge.net/projects/opencvlibrary/files/> and double-click to extract it.

4. Goto **opencv/build/python/2.7** folder.

5. Copy **cv2.pyd** to **C:/Python27/lib/site-packages**.

6. Open Python IDLE and type following codes in Python terminal.

```
>>> import cv2  
>>> print cv2.__version__
```

If the results are printed out without any errors, congratulations !!! You have installed OpenCV-Python successfully.

### Anaconda for Windows (Optional)

7. Download Anaconda: <https://www.continuum.io/downloads>

8..Double click the Anaconda installer and follow the prompts to install to the default location.

NOTE: If you encounter any issues during installation, please temporarily disable your anti-virus software during install, then immediately re-enable it. If you have installed for all users, uninstall Anaconda and re-install it for your user only and try again.

Goto **opencv/build/python/2.7** folder. Copy **cv2.pyd** to **C:\Program Files\Anaconda2\Lib\site-packages**.

## Preparing your Raspberry Pi for computer vision

1. Connect your Pi to an Internet modem or router with an Ethernet cable.
2. Run the following command to restart the networking service:  
**sudo service networking restart**
3. Make sure that Raspberry Pi is connected to the Internet by typing in the following command:

**ping -c4 www.google.com**

4. Run the following commands in a sequence:

Advanced Package Tool (apt) is the utility that can be used to install and remove software in Debian and its variants. We need to use it to update the Pi software.

**sudo apt-get update**

This command synchronizes the package list from the source. Indexes of all the packages are refreshed. This command must be issued before we issue the upgrade command.

**sudo apt-get upgrade**

This will install the newest versions of the already installed software. Obsolete packages/utilities are not removed automatically. If the software is up to date, then it's left as it is.

**sudo rpi-update**

This command is used to upgrade the firmware. The kernel and firmware are installed as a Debian package, and hence, we will also get the updates. These packages are updated infrequently after extensive testing.

5. Now, we will need to install a few necessary packages and dependencies for OpenCV. Following is a list of packages we need to install. You just need to connect Pi to the Internet and type in `sudo apt-get install <package-name>`, where `<package-name>` is one of the following packages:

libopencv-dev	libpng3	libdc1394-22-dev
build-essential	libpnglite-dev	libdc1394-22
libavformat-dev	zlib1g-dbg	libdc1394-utils
x264	zlib1g	libv4l-0
v4l-utils	zlib1g-dev	libv4l-dev
Ffmpeg	pngtools	libpython2.6
libcv2.3	libtiff4-dev	python-dev
libcvaux2.3	libtiff4	python2.6-dev
libhighgui2.3	libtiffxx0c2	libgtk2.0-dev
python-opencv	libtiff-tools	libpngwriter0-dev
opencv-doc	libjpeg8	libpngwriter0c2

libcv-dev	libjpeg8-dev	libswscale-dev
libcvaux-dev	libjpeg8-dbg	libjpeg-dev
libhighgui-dev	libavcodec-dev	libwebp-dev
python-numpy	libavcodec53	libpng-dev
python-scipy	libavformat53	libtiff5-dev
python-matplotlib	libgstreamer0.10-0-dbg	libjasper-dev
python-pandas	libgstreamer0.10-0	libopenexr-dev
python-nose	libgstreamer0.10-dev	libgdal-dev
v4l-utils	libxine1-ffmpeg	python-tk
libgtkglext1-dev	libxine-dev	python3-dev
libpng12-0	libxine1-bin	python3-tk
libpng12-dev	libunicap2	python3-numpy
libpng++-dev	libunicap2-dev	libeigen3-dev

For example, if you want to install x264, you have to type `sudo apt-get install x264`. This will install the necessary package. Similarly, you can install all of the aforementioned packages in like manner. If a package is already installed on Pi, it will show the following message:

```
pi@pi02 ~ $ sudo apt-get install x264
Reading package lists... Done
Building dependency tree
Reading state information... Done
x264 is already the newest version.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
```

In such cases, don't worry. The package you wanted to install has already been installed, and it is up to date. Just proceed with the installation of all the other packages in the list one-by-one.

6. Finally, install OpenCV for Python by using the following command:

**`sudo apt-get install python-opencv`**

This is the easiest way to install OpenCV for Python. However, there is a problem with this. Raspbian repositories may not always contain the latest version of OpenCV.

## OpenCV basic Command Description

### 1. cv2.imread

Use the function `cv2.imread()` to read an image. The image should be in the working directory or a full path of image should be given. Second argument is a flag which specifies the way image should be read.

- `cv2.IMREAD_COLOR`: Loads a color image. Any transparency of image will be neglected. It is the default flag.
- `cv2.IMREAD_GRAYSCALE`: Loads image in grayscale mode
- `cv2.IMREAD_UNCHANGED`: Loads image as such including alpha channel

**Note: Instead of these three flags, you can simply pass integers 1, 0 or -1 respectively**

Example: `img = cv2.imread('messi5.jpg',0)`

### 2. cv2.imshow

Use the function `cv2.imshow()` to display an image in a window. The window automatically fits to the image size. First argument is a window name which is a string. second argument is our image. You can create as many windows as you wish, but with different window names.

Example: `cv2.imshow('image',img)`

### 3. cv2.waitKey()

`cv2.waitKey()` is a keyboard binding function. Its argument is the time in milliseconds. The function waits for specified milliseconds for any keyboard event. If you press any key in that time, the program continues. If 0 is passed, it waits indefinitely for a key stroke. It can also be set to detect specific key strokes like.

### 4. cv2.destroyAllWindows()

`cv2.destroyAllWindows()` simply destroys all the windows we created. If you want to destroy any specific window, use the function `cv2.destroyWindow()` where you pass the exact window name as the argument.

### 5. Cv2.imwrite

Use the function `cv2.imwrite()` to save an image. First argument is the file name, second argument is the image you want to save.

Example: `cv2.imwrite('messigray.png',img)`

### 6. Cv2.VideoCapture

To capture a video, you need to create a `VideoCapture` object. Its argument can be either the device index or the name of a video file. Device index is just the number to specify which camera. Normally one camera will be connected (as in my case). So I simply pass 0 (or -1). You can select the second camera by passing 1 and so on.

Example: `cap = cv2.VideoCapture(0)`

### 7. Cap.read()

`cap.read()` returns a bool (True/False). If frame is read correctly, it will be True. So you can check end of the video by checking this return value. Sometimes, cap may not have initialized the capture.

In that case, this code shows error. You can check whether it is initialized or not by the method `cap.isOpened()`. If it is True, OK. Otherwise open it using `cap.open()`.

## 8. Numpy – ndarray attributes

### **ndarray.ndim**

Outputs the number of axes (dimensions) of the array i.e. the rank.

### **ndarray.shape**

Outputs the dimensions of the array. This is a tuple of integers indicating the size of the array in each dimension. For a matrix with n rows and m columns, shape will be (n,m). The length of the shape tuple is therefore the rank, or number of dimensions, ndim.

### **ndarray.size**

Outputs the total number of elements of the array, equal to the product of the elements of shape.

### **ndarray.dtype**

Outputs an object describing the type of the elements in the array. One can create or specify dtype's using standard Python types. NumPy provides many, for example `bool_`, `character`, `int_`, `int8`, `int16`, `int32`, `int64`, `float_`, `float8`, `float16`, `float32`, `float64`, `complex_`, `complex64`, `object_`.