Life2Coding

Solutions For Lifetime

HOW TO INSTALL OPENCV 3.4.0 WITH PYTHON 3 ON RASPBERRY PI 3

Introduction

OpenCV (Open Source Computer Vision) is a library of programming functions mainly aimed at real-time computer vision. It is released under a BSD license and hence it's free for both academic and commercial use. It has C++, Python and Java interfaces and supports Windows, Linux, Mac OS, iOS and Android. OpenCV was designed for computational efficiency and with a strong focus on real-time applications.

In this post, I will show you how to install OpenCV 3.4.0 with Python 3.5 on Raspberry Pi 3. First I will show you how to install OpenCV3 without any virtual environment and how you can also use a virtual environment to work with that build file as well.

Objectives:

Instruction to install OpenCV on Raspberry Pi.

- OpenCV version: 3.4.0
- Target platform: Raspberry Pi 3 B

Search Here

Search

Recent Posts


Split Video Channels into RGB components using OpenCV in Python

6 February, 2018


Convert Image Frames to Video File using OpenCV in Python

6 February, 2018


Extract Each Frame from a Video File using OpenCV in Python

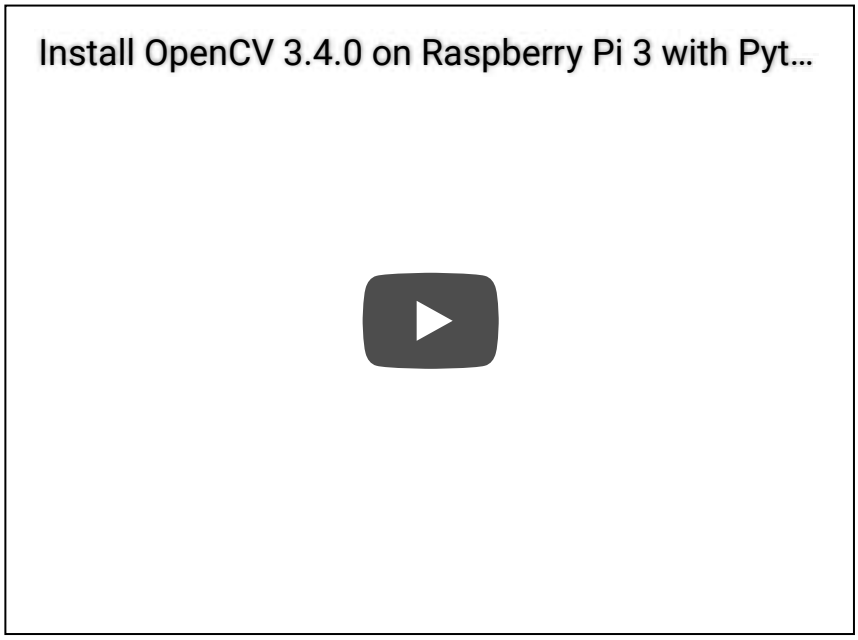
6 February, 2018


Play Video Files using OpenCV in Python

6 February, 2018

- OS: Raspbian Stretch
- Language: Python 3

Video Youtube:



Steps:

There are some steps to install OpenCV properly on Raspberry Pi 3 with Python 3. I will show all the steps to get it working properly.

Step 1: Expand filesystem

Type the following command to expand the Raspberry Pi3 file system

```
sudo raspi-config
```

Then select the following

- Advanced Options > A1 Expand filesystem > Press “Enter”

It will show a message “The root partition has been resized”.

Then you need to reboot your pi using the following command.

```
sudo shutdown -r now
```

Step 2: Free Up Some Space

The default OS will take around 15% if you are using 32GB memory card. But if you are using a 8GB memory card it might take 50% of all your space. So, it is better to remove some unused

Home Learning Arena Our Products Contact Us About Us

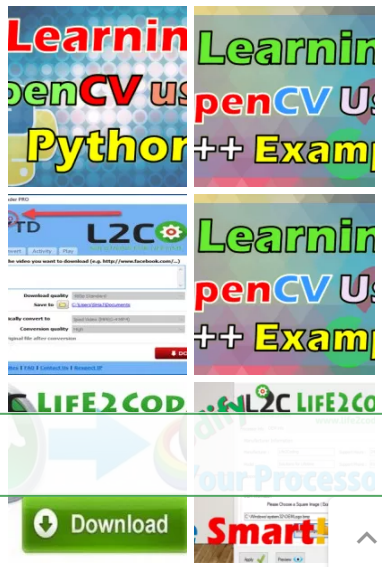

Save Webcam Video Feed to a File using OpenCV in Python

6 February, 2018

Categories

Learning Arena (71)
Computer Tricks (15)
Embedded Technology (21)
Arduino (19)
Microchip
Microcontrollers (1)
Raspberry Pi (1)
Image Processing (31)
OpenCV C++ (17)
OpenCV Python (14)
Web Development (4)
HTML+CSS (4)
Products (2)
URI Contest Programming (16)

Top Posts & Pages



packages like LibreOffice and Wolfram engine to free up some space on your pi.

You can do it simply typing the following command on the terminal window.

```
sudo apt-get purge wolfram-engine
sudo apt-get purge libreoffice*
sudo apt-get clean
sudo apt-get autoremove
```

Step 3: Install Dependencies

- The first step is to update and upgrade any existing packages:

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

- If you have been shown any error to fix you can type the following

```
sudo apt-get upgrade --fix-missing
```

- Then reboot your pi.

```
sudo shutdown -r now
```

After your pi boots up start the Terminal again. Do the following.

- Install CMAKE developer packages

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

- Install Image I/O packages

```
sudo apt-get install libjpeg-dev libtiff5-dev libjaspe
```

- Install Video I/O packages

```
sudo apt-get install libavcodec-dev libavformat-dev li
sudo apt-get install libxvidcore-dev libx264-dev -y
```

- Install the GTK development library for basic GUI windows

```
sudo apt-get install libgtk2.0-dev libgtk-3-dev -y
```

- Install optimization packages (improved matrix operations for OpenCV)

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

Step 4: Install Python 3, setuptools, dev and Numpy

- Install Python 3 and numpy

```
sudo apt-get install python3 python3-setuptools python
wget https://bootstrap.pypa.io/get-pip.py
sudo python3 get-pip.py
sudo pip3 install numpy
```

Step 5: Download the OpenCV 3.4 and contrib extra modules

```
cd ~
wget -O opencv.zip https://github.com/Itseez/opencv/ar
unzip opencv.zip
wget -O opencv_contrib.zip https://github.com/Itseez/o
unzip opencv_contrib.zip
```

Step 6: Compile and Install OpenCV 3.4.0 for Python 3

```
cd opencv-3.4.0
mkdir build
cd build
cmake -D CMAKE_BUILD_TYPE=RELEASE \
-D CMAKE_INSTALL_PREFIX=/usr/local \
-D BUILD_opencv_java=OFF \
-D BUILD_opencv_python2=OFF \
-D BUILD_opencv_python3=ON \
-D PYTHON_DEFAULT_EXECUTABLE=$(which python3) \
-D INSTALL_C_EXAMPLES=OFF \
-D INSTALL_PYTHON_EXAMPLES=ON \
-D BUILD_EXAMPLES=ON\
-D OPENCV_EXTRA_MODULES_PATH=~/opencv_contrib-3.4.0/m
-D WITH_CUDA=OFF \
-D BUILD_TESTS=OFF \
-D BUILD_PERF_TESTS= OFF ..
```

Step 7: Swap Space size before compiling to add more virtual memory

It will enable OpenCV to **compile with all four cores** of the Raspberry PI without any memory issues.

Open your **/etc/dphys-swapfile** and then edit the **CONF_SWAPSIZE** variable

```
sudo nano /etc/dphys-swapfile
```

It will open the nano editor for editing the CONF_SWAPSIZE. Change it like below:

```
# set size to absolute value, leaving empty (default)
# you most likely don't want this, unless you have an
# CONF_SWAPSIZE=100
CONF_SWAPSIZE=1024
```

Then save the changes you've made, press **Ctrl + O**. To exit nano, type **Ctrl + X**. If you ask nano to exit from a modified file, it will ask you if you want to save it. Just press **N** in case you don't, or **Y** in

case you do. It will then ask you for a filename. Just type it in and press **Enter.**

Then type the following lines to take it into effect

```
sudo /etc/init.d/dphys-swapfile stop
sudo /etc/init.d/dphys-swapfile start
```

Step 7: Finally Ready to be Compile

Type the following command to compile it using 4 cores of pi

```
make -j4
```

Step Optional: Compile with a single core of Pi

If you face any error while compiling due to memory issue you can start the compilation again with only one core using the following command

```
make clean
make
```

Step 8: Install the build on raspberry pi

After the successful build install the build using the following command

```
sudo make install
sudo ldconfig
```

Step 9: Verify the OpenCV build

After running make install, OpenCV + Python bindings should be installed **in usr/local/lib/python3.5/dist-packages** or **usr/local/lib/python3.5/site-packages**.

You need to use the **site-packages** or **dist-packages**. Look where it has been created and use that **site-packages** or **dist-packages**. In my case it is in **dist-packages**.

Again, you can verify this with the **ls** command:

```
ls -l /usr/local/lib/python3.5/dist-packages/
```

Look for a name like **cv2.so** and if it is not there then look for a name like **cv2.cpython-35m-arm-linux-gnueabihf.so** (name starting with cv2. and ending with .so). It might happen due to some bugs in Python binding library for Python 3.

We need to rename **cv2.cpython-35m-arm-linux-gnueabihf.so** to **cv2.so** using the following command:

```
cd /usr/local/lib/python3.5/dist-packages/
sudo mv /usr/local/lib/python3.5/dist-packages/cv2.cpy
```

Step 10: Testing OpenCV 3.4.0 install

```
pi@raspberrypi:~ $ python3
Python 3.5.3 (default, Jan 19 2017, 14:11:04)
[GCC 6.3.0 20170124] on linux
Type "help", "copyright", "credits" or "license" for m
>>> import cv2
>>> cv2.__version__
'3.4.0'
```

Step Optional: Remove the zip files to free up some space:

```
cd ~
rm opencv.zip opencv_contrib.zip
```

Step 11: Don't forget to change your swap size back!

Open your **/etc/dphys-swapfile** and then edit the **CONF_SWAPSIZE** variable

```
sudo nano /etc/dphys-swapfile
```

It will open the nano editor for editing the **CONF_SWAPSIZE**. Change it like below:

```
# set size to absolute value, leaving empty (default)
# you most likely don't want this, unless you have an
CONF_SWAPSIZE=100
# CONF_SWAPSIZE=1024
```

Then save the changes you've made, press **Ctrl + O**. To exit nano, type **Ctrl + X**. If you ask nano to exit from a modified file, it will ask you if you want to save it. Just press **N** in case you don't, or **Y** in case you do. It will then ask you for a filename. Just type it in and press **Enter.**

Then type the following lines to take it into effect

```
sudo /etc/init.d/dphys-swapfile stop
sudo /etc/init.d/dphys-swapfile start
```

Step 12 (Additional): Setting OpenCV for a virtual environment (Python 3)

Next section is for adding access from a virtual environment.

- Make sure that you have installed venv for Python 3.

```
sudo apt-get install python3-venv -y
```

- Make a virtual environment for OpenCV3 with Python3

```
python3 -m venv ~/cvpi
```

- To activate the venv you made, execute

```
source ~/cvpi/bin/activate
```

Let's make a symbolic link inside of your venv package folder.

```
ln -s /usr/local/lib/python3.5/dist-packages/cv2.so ~/
```

Here, **'~/cvpi'** is the virtual environment directory

The result will look like this.

```
(cvpi) pi@raspberrypi:~ $ cd ~/cvpi/lib/python3.5/sit
(cvpi) pi@raspberrypi:~/cvpi/lib/python3.5/site-packa
total 48
drwxr-xr-x 11 pi pi 4096 Mar 3 18:32 .
drwxr-xr-x 3 pi pi 4096 Mar 3 18:27 ..
lrwxrwxrwx 1 pi pi 45 Mar 3 18:32 cv2.so -> /usr/loca
-rw-r--r-- 1 pi pi 126 Mar 3 18:27 easy_install.py
drwxr-xr-x 17 pi pi 4096 Mar 3 18:31 numpy
drwxr-xr-x 2 pi pi 4096 Mar 3 18:31 numpy-1.14.1.dist
drwxr-xr-x 11 pi pi 4096 Mar 3 18:27 pip
drwxr-xr-x 2 pi pi 4096 Mar 3 18:27 pip-9.0.1.dist-in
drwxr-xr-x 5 pi pi 4096 Mar 3 18:30 pkg_resources
drwxr-xr-x 2 pi pi 4096 Mar 3 18:30 pkg_resources-0.0
drwxr-xr-x 2 pi pi 4096 Mar 3 18:27 __pycache__
drwxr-xr-x 5 pi pi 4096 Mar 3 18:27 setuptools
drwxr-xr-x 2 pi pi 4096 Mar 3 18:27 setuptools-33.1.1
```

Don't forget to install numpy for a new venv

```
pip3 install numpy
```

Step 13: Testing OpenCV 3.4.0 in the virtual environment

Now, check that you can use cv2 INSIDE of the virtual environment.

```
(cvpi) pi@raspberrypi:~ $ python3
Python 3.5.3 (default, Jan 19 2017, 14:11:04)
[GCC 6.3.0 20170124] on linux
Type "help", "copyright", "credits" or "license" for m
>>> import cv2
>>> cv2.__version__
'3.4.0'
```

Activate and Deactivate your Virtual Environment

Activate

```
source ~/cvpi/bin/active
```

Deactivate

```
deactivate
```

About | Latest Posts

Life2Coding
Technology Related Blog at Life2Coding

Feel free to contact us for your any kind of technical problems. We are here to help you.

Share this:

Related

Extract Each Frame from a Video File using OpenCV in Python

6 February, 2018
In "OpenCV Python"

How to Save OpenCV Image to a File in Python

4 February, 2018
In "OpenCV Python"

Convert Image Frames to Video File using OpenCV in Python

6 February, 2018
In "OpenCV Python"

Life2Coding 3 March, 2018 Embedded Technology, Raspberry Pi 3 Comments

← Split Video Channels into RGB components using OpenCV in Python

3 thoughts on “How to Install OpenCV 3.4.0 with Python 3 on Raspberry Pi 3”

1. omar
22 March, 2018 at 8:09 pm

hello, having a problem about Step 7: Finally Ready to be Compile just output
“make: *** No target specified and no makefile found. stop”

How can i solve this problem

↳ Reply

A.  Life2Coding Post author
13 April, 2018 at 6:52 am
Check the video tutorial (Link Above attached). I think your directory is different. Try to go to the exact directory.

↳ Reply

2.  Rodo
11 April, 2018 at 2:02 pm
Thank you very much, bro! Your post is informative and clear for newbie like me ^^.

↳ Reply

Leave a Reply

Your email address will not be published. Required fields are marked *

Comment

Name *

Email *

Website

post comment

Notify me of follow-up comments by email.

Notify me of new posts by email.