Installing OpenCV on the Raspberry Pi

by Emmet Published Dec 19, 2019 Updated Feb 11, 2021

Beginner, Camera

Throughout this guide, we will walk you through the process of setting up OpenCV on your Raspberry Pi.

Successfully installing OpenCV to your Raspberry Pi requires a couple of different steps and a fair bit of patience.

For those who do not know what OpenCV is. It is a library of different programming functions that are aimed at dealing with real-time computer vision.

Using computer vision, you can interpret images and videos in real-time. Allowing you to perform tasks such as motion detection and facial recognition with relative ease.

The Raspberry Pi is an excellent platform for starting to learn OpenCV and also doubles as an affordable and small device.

Equipment List

Here is all the equipment that we recommend for this Raspberry Pi OpenCV tutorial.

Recommended

Raspberry Pi 1, 2, 3 or 4

Micro SD Card

Power Supply

Ethernet Cord or WiFi dongle (The Pi 3 and 4 has WiFi inbuilt)

Raspberry Pi camera or USB WebCam

Optional

Raspberry Pi Case

This tutorial was tested using the latest version of Raspbian Buster. If you are running an older version of Raspbian, you can use our guide to upgrade to Buster.

Installing Packages for OpenCV

In this section, we will be walking you through the process of installing all the packages you need to compile and run the OpenCV software.

As OpenCV requires so many packages on the Raspberry Pi, we will install these over a couple of steps.

1. Before proceeding, we should first update any preexisting packages.

You can update the currently installed packages by running the following two commands.

Terminal $

```
sudo apt update
sudo apt upgrade
```

2. Now we can start the process of installing all the packages we need for OpenCV to compile.

To start, run the command below. This command will install the packages that contain the tools needed to compile the OpenCV code.

Terminal $

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

3. Next, we are going to install the packages that will add support for different image and video formats to OpenCV.

Install these libraries to your Raspberry Pi with the following command.

Terminal $

```
sudo apt install libjpeg-dev libtiff-dev libjasper-dev libpng-dev libwebp-dev libopenexr-dev
sudo apt install libavcodec-dev libavformat-dev libswscale-dev libv4l-dev libxvidcore-dev libx264-dev libdc1394-22-dev libgstreamer-plugins-base1.0-dev libgstreamer1.0-dev
```

4. Our next step is to install all the packages needed for OpenCV’s interface by using the command below.

Terminal $

```
sudo apt install libgtk-3-dev libqtgui4 libqtwebkit4 libqt4-test python3-pyqt5
```

5. These next packages are crucial for OpenCV to run at a decent speed on the Raspberry Pi.

You can install these packages by running the following command.

Terminal $

```
sudo apt install libatlas-base-dev liblapacke-dev gfortran
```

6. The second last lot of packages thaat we need to install relate to the Hierarchical Data Format (HDF5) that OpenCV uses to manage data.

Install the HDF5 packages to your Pi by using the command below.

Terminal $

```
sudo apt install libhdf5-dev libhdf5-103
```

7. Finally, we can install the final few packages by using the command below.

These last few packages will allow us to compile OpenCV with support for Python on our Raspberry Pi.

Terminal $

```
sudo apt install python3-dev python3-pip python3-numpy
```

Before proceeding to the next section, make sure all the packages installed successfully.

Preparing your Raspberry Pi for Compiling OpenCV

1. With all the required packages to compile OpenCV on our Raspberry Pi now installed, we need to do some preparatory work before we can start the compilation process.

We will now need to temporarily increase the size of the swap space to help the process of compiling OpenCV on the Raspberry Pi.

The swap space is used by the operating system when the device has run out of physical RAM. While swap memory is a lot slower than RAM, it can still be helpful in certain situations.

Begin modifying the swap file configuration by running the following command.

Terminal $

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

2. While we are within this file, we need to find and replace the following line.

Find

Find >

```
CONF_SWAPSIZE=100
```

Replace With

Replace With >

```
CONF_SWAPSIZE=2048
```

Once changed, save the file by pressing `CTRL`+`X` followed by `Y` then `Enter`.

3. As we have made changes to the swapfile configuration, we need to restart its service by utilizing the command below.

Terminal $

```
sudo systemctl restart dphys-swapfile
```

By restarting the service, we are forcing it to recreate the swap file.

4. Next, let’s go ahead and clone the two OpenCV repositories we need to our Raspberry Pi.

Running these two commands will retrieve the latest available version of OpenCV from their git repository.

Terminal $

```
git clone https://github.com/opencv/opencv.git
git clone https://github.com/opencv/opencv_contrib.git
```

As these repositories are quite large, they may take some time to clone to your Raspberry Pi.

Compiling OpenCV on your Raspberry Pi

1. Let’s start by creating a directory called “**build**” within the cloned “**opencv**” folder and then changing the working directory to it.

Terminal $

```
mkdir ~/opencv/build
cd ~/opencv/build
```

In this folder, we will be compiling OpenCV on your Raspberry Pi.

2. Now that we are within our newly created build folder, we can now use `cmake` to prepare OpenCV for compilation on our Raspberry Pi.

Run the following command to generate the required makefile.

Terminal $

```
cmake -D CMAKE_BUILD_TYPE=RELEASE \
    -D CMAKE_INSTALL_PREFIX=/usr/local \
    -D OPENCV_EXTRA_MODULES_PATH=~/opencv_contrib/modules \
    -D ENABLE_NEON=ON \
    -D ENABLE_VFPV3=ON \
    -D BUILD_TESTS=OFF \
    -D INSTALL_PYTHON_EXAMPLES=OFF \
    -D OPENCV_ENABLE_NONFREE=ON \
    -D CMAKE_SHARED_LINKER_FLAGS=-latomic \
    -D BUILD_EXAMPLES=OFF ..
```

3. Once the make file has successfully finished generating, we can now finally move on to compiling OpenCV by running the command below.

We use the argument `-j$(nproc)` to tell the compiler to run a compiler for each of the available processors.

Doing this will significantly speed up the compilation process and allow each core on the Raspberry Pi to work on compiling OpenCV.

Terminal $

```
make -j$(nproc)
```

Please note that the compilation process can take considerable time. On our Raspberry Pi 4, this process took about 1 hour to complete.

4. When the compilation process finishes, we can then move on to installing OpenCV.

Luckily for us, this is a reasonably straightforward process and requires you to run the following command.

Terminal $

```
sudo make install
```

This command will copy all the required files into there needed locations automatically.

5. Now we also need to regenerate the operating systems library link cache.

The Raspberry Pi won’t be able to find our OpenCV installation if we don’t run the following command.

Terminal $

```
sudo ldconfig
```

♻ Cleaning up after Compilation

1. Now that we have finished compiling OpenCV, we no longer need to have such a large swap file.

Let’s again edit the swap file configuration by using the following command.

Terminal $

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

2. Within this file, you need to find and change the following line.

Find

Find >

```
CONF_SWAPSIZE=2048
```

Replace With

Replace With >

```
CONF_SWAPSIZE=100
```

When done, save the file by pressing `CTRL`+`X` followed by `Y` then `Enter`.

3. Now our final cleanup task requires us to restart the swap file service.

Restarting the service will downsize the file from 2GB to 100 MB.

Terminal $

```
sudo systemctl restart dphys-swapfile
```

Testing OpenCV on your Raspberry Pi

1. To test whether OpenCV is now installed to our Raspberry Pi, we will make use of our Python 3 installation.

Launch into the Python terminal by running the command below.

Terminal $

```
python3
```

2. While we are within Python, we can now import the OpenCV Python module using the command below.

By importing the module, we can first check to see if OpenCV will even load on our Pi.

Python >

```
import cv2
```

3. With the OpenCV module now imported, we should be able to retrieve its version.

To retrieve OpenCV's version, use the following command.

Python >

```
cv2.__version__
```

4. If everything is now working as intended and OpenCV has been successfully installed to your Raspberry Pi, you should see text like the following appear in the command line.

Python >

```
'4.1.2'
```

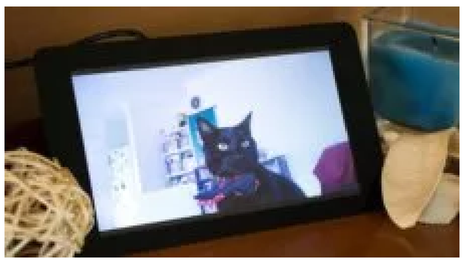
Hopefully, at this point you will now have OpenCV up and running.

If you have run into any issues or have any feedback on this Raspberry Pi OpenCV tutorial, then feel free to drop a comment below.

Easy Raspberry Pi WiFi Bridge

Raspberry Pi BOINC Tutorial

Installing XBian to your Raspberry Pi



A Simple Raspberry Pi Photo Frame

Installing Docker on the Raspberry Pi

How to Setup a Raspberry Pi AFP Server

14 Comments

Static_Shock on February 12, 2021 at 1:51 am

Took a very long time but it worked. Thanks

Reply

Bob on January 1, 2021 at 12:50 am

all went smooth, thanks for the clear instructions and explanations

Reply

ronen shemesh on December 30, 2020 at 10:32 pm

a very good tutorial, the best!!

Reply

Kim K on December 17, 2020 at 5:14 am

Thank you! Thank you! Thank you!! I've tried 5 different installation instructions over the last week. Yours worked!! Thank you so much!!

Reply

alem on October 17, 2020 at 6:22 am

Thank you so much. This tutorial was so helpful in the process of installing openCV on Raspberry PI 4

Reply

Aurimas on June 1, 2020 at 10:40 am

Thanks man! Works perfect.
It took about 3h to finish on RPi 3b+

Reply

Alexander Warren on April 22, 2020 at 11:46 am

Hi, in step 6 it says to run `sudo apt install libhdf5-dev libhdf5-103`, however when running that I get the response `E: Unable to locate package libhdf5-103`. After doing `apt search libhdf5-103`, it seems like that package doesn't exist? Do you have any ideas on how to fix this problem? Thanks.

Reply

Emmet on May 6, 2020 at 2:14 pm
🛡 Editor

Hi Alexander,

Make sure that you are running Raspbian Buster or later. I'm uncertain if those packages are available on older versions of Raspbian.

Cheers,
Emmet

Lionel Vigier on April 4, 2020 at 11:47 pm

Hi,
I have just installed cv2 using you tutorial. It works fine. Thank you very much
Regards

Reply

Stephane Laurin on March 30, 2020 at 1:26 pm

I get this erro message on the last commands. Everything else, I've followed. RPY-4 w/ 4GRAM… CV2 still not working!

```
pi@raspberrypi:~/opencv/build $ python3
Python 3.7.3 (default, Dec 20 2019, 18:57:59)
[GCC 8.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import cv2
Traceback (most recent call last):
File "", line 1, in
File "/home/pi/.local/lib/python3.7/site-packages/cv2/__init__.py", line 3, in
from .cv2 import *
ImportError: /home/pi/.local/lib/python3.7/site-packages/cv2/cv2.cpython-37m-arm-linux-gnueabihf.so: undefined symbol: __atomic_fetch_add_8
```

Reply

Emmet on April 1, 2020 at 2:27 pm
Editor

Hi Stephane,

I re-ran this tutorial last night on a Raspberry Pi running Raspbian Buster and everything compiled and installed correctly.

Make sure you do not have the pip version of OpenCV installed before running this tutorial.

Cheers,
Emmet

matoboor on March 17, 2020 at 11:34 pm

Thanks for nice tutorial. Worked like a charm.

Reply

Guilherme on December 27, 2019 at 9:22 am

Hello!
Very good tutorial. Just a small typo in the last portion of it:
sudo apt install pytohn3-dev python3-pip python3-numpy

the install says ” pytohn3-dev”, when it should actually be “python3-dev” .

Thanks for the tutorial!
Guilherme

Reply

Gus on December 27, 2019 at 2:00 pm
🛡 Editor

Hi Guilherme,

I have fixed the typo! Thanks for pointing that out.

Leave a Reply

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

Comment

Name *

Email *

☐ Notify me of follow-up comments by email.

Post Comment

Trending

[Build Your Own Raspberry Pi VPN Server](#) [How to setup a Raspberry Pi RFID RC522 Chip](#)

[Installing Deluge on the Raspberry Pi](#) [Running Zoom on the Raspberry Pi](#)

Back to Top ↑

Disclaimer & Privacy Policy | About us | Contact