Jetson Nano Computer Vision Setup

Table of Contents

[I. OpenCV Installation & Set-up 2](#_heading=h.gjdgxs)

[II. Camera Module Set-up 6](#_heading=h.30j0zll)

[III. OpenCV & Camera Module Compatibility 7](#_heading=h.2et92p0)

## **OpenCV Installation & Set-up**

This section outlines the process to successfully install a compatible OpenCV version for the Jetson Nano.

A default version of OpenCV would have been pre-installed in the Jetson Nano, but the following steps are essential to be carried out to ensure that the camera module (in our case - IMX219-77 Camera) is fully functionable and compatible to be able to carry out computer vision purposes.

Note that the OpenCV version described and covered below is compatible with **Python 3.6.9** version, in which updated versions MIGHT NOT WORK. There are a few key useful commands as stated below:

| **Commands** | **Description** |
| --- | --- |
| python -c 'import cv2; print(cv2.\_\_version\_\_)' | Check and print out the OpenCV version.   * **Compatible version: 4.5.1** * Incompatible version: 4.1.1 * Incompatible version: 3.2.0 |
| free -h | Check the swap memory.   * Original Swap: 1.9 G * Updated Swap: 5.9 G |

#### Creating a virtual environment (Optional)

Virtual environments are better choice when you are running multiple different kind of applications on the similar device. However, considering if you are using the Jetson Nano for only 1 application then there is no need of virtual environment.

| **Commands** | **Description** |
| --- | --- |
| python3 -m virtualenv -p python3 env --system-site-packages | Create a separate virtual environment. |
| source env/bin/activate | Activate the environment. |

#### Creating Swap File

Swap file is created to tackle the low running memory issue during installation of applications. Follow the steps below to increase the memory space from 1.9 G 🡪 5.9 G. Note that these commands should be executed line-by-line.

| **Commands** | **Description** |
| --- | --- |
| sudo fallocate -l 4G /var/swapfile | Line 1 |
| sudo chmod 600 /var/swapfile | Line 2 |
| sudo mkswap /var/swapfile | Line 3 |
| sudo swapon /var/swapfile | Line 4 |
| sudo bash -c 'echo "/var/swapfile swap swap defaults 0 0" >> /etc/fstab’ | Line 5 |

Reboot the Jetson Nano and double check to ensure that the swap memory or space should be “Swap: 5.9 G”, using command free -h.

#### Installing dependencies for OpenCV

There are a few dependencies, also known as third-party packages / libraries which requires installing before downloading OpenCV. Note that there is a possibility that a handful of these commands have already been downloaded in the Jetson Nano, but it should still be executed on a line-by-line basis for precaution purposes, according to the list below.

| **Commands** |
| --- |
| sudo sh -c "echo '/usr/local/cuda/lib64' >> /etc/ld.so.conf.d/nvidia-tegra.conf“ |
| sudo ldconfig |
| sudo apt-get install build-essential cmake git unzip pkg-config |
| sudo apt-get install libjpeg-dev libpng-dev libtiff-dev |
| sudo apt-get install libavcodec-dev libavformat-dev libswscale-dev |
| sudo apt-get install libgtk2.0-dev libcanberra-gtk\* |
| sudo apt-get install python3-dev python3-numpy python3-pip |
| sudo apt-get install libxvidcore-dev libx264-dev libgtk-3-dev |
| sudo apt-get install libtbb2 libtbb-dev libdc1394-22-dev |
| sudo apt-get install libv4l-dev v4l-utils |
| sudo apt-get install libgstreamer1.0-dev libgstreamer-plugins-base1.0-dev |
| sudo apt-get install libavresample-dev libvorbis-dev libxine2-dev |
| sudo apt-get install libfaac-dev libmp3lame-dev libtheora-dev |
| sudo apt-get install libopencore-amrnb-dev libopencore-amrwb-dev |
| sudo apt-get install libopenblas-dev libatlas-base-dev libblas-dev |
| sudo apt-get install liblapack-dev libeigen3-dev gfortran |
| sudo apt-get install libhdf5-dev protobuf-compiler |
| sudo apt-get install libprotobuf-dev libgoogle-glog-dev libgflags-dev |

#### Installing OpenCV (Compatible version)

The final step would be to install the compatible version of OpenCV – 4.5.1 version. Steps can be categorized into the four sections below, each with their own respective commands.

* 1. **Downloading**

| **Commands** | **Description** |
| --- | --- |
| wget -O [opencv.zip](https://www.youtube.com/redirect?event=comments&redir_token=QUFFLUhqa2NRZWRYamNVVDRNOVFVeUZUNVJ6aHliQU10QXxBQ3Jtc0trbEtpRHh0T1NQamF2OE9ORFBwSy1RNDJUeHpZTlFOSmRlbVlFRF9XN0E5cmU0Mml1eDZUTElXVEF3cnJEWmwyUDF2a19laFVYMHc0Q1lnVDdaQ0F2eTJacWFXNTk4OVF4STBRS3dKUEpvRkVkT3FQQQ&q=http%3A%2F%2Fopencv.zip%2F&stzid=UgzHycXony_VUGL5Alx4AaABAg) [https://github.com/opencv/opencv/archive/4.5.1.zip](https://www.youtube.com/redirect?event=comments&redir_token=QUFFLUhqbGN6SXJRYXcwR1UtMm5fUkNXLW0tR3BJX3pGQXxBQ3Jtc0trN1N6VkhCZjMzSFM4VzZwWDZBbkROY3haMklldVRpZlFsQ1lOa2FtMWNSNi0tMGcyemtySjM2MnphemRmY1J3TXk2T0oxSmVtNWxnSC1QU1BGS0VhUUYyY1hlQzgtYzRRNklMdk10bnZ1WHRjNWRtVQ&q=https%3A%2F%2Fgithub.com%2Fopencv%2Fopencv%2Farchive%2F4.5.1.zip&stzid=UgzHycXony_VUGL5Alx4AaABAg) | Download the zip file for OpenCV 4.5.1 version. |
| wget -O opencv\_[contrib.zip](https://www.youtube.com/redirect?event=comments&redir_token=QUFFLUhqbWJUNUwzc2xST3ZOeEd5a1NWZzJjTUZQb2FvQXxBQ3Jtc0tud3Y1Q0k0S1FQMWEtZXBWRVFkSDVFZnRzNUxvd01hV3ZLOU5YSlVsNy1jdExCbVp3TU9BNEZnYVVQdUpHTzdRUnpNMWk4Y2FnQmRYTFZobzZ3M1c4TC1oamc1cTJ6bjJZMld2TndQT2FUb0VMUXF6aw&q=http%3A%2F%2Fcontrib.zip%2F&stzid=UgzHycXony_VUGL5Alx4AaABAg) [https://github.com/opencv/opencv\_contrib/archive/4.5.1.zip](https://www.youtube.com/redirect?event=comments&redir_token=QUFFLUhqa3VaVGRzbzlMNHV5YTdoMzI5UnIzVmF3OUVhQXxBQ3Jtc0tuaGVrRExSeEFiVENNSURSYmRyTHRyRFBHUXJxSWd3R3FZVlVEM0hFb213ZTlXS1h5VTc2b3pvMU1uakdTY1J1YmF4S3RreVpxLVpacXRQQmV0TURMVHN1SWQxMDhmVnkzcHJpeWhKa3hNR2RCal9sTQ&q=https%3A%2F%2Fgithub.com%2Fopencv%2Fopencv_contrib%2Farchive%2F4.5.1.zip&stzid=UgzHycXony_VUGL5Alx4AaABAg) | Download the zip file for OpenCV contrib 4.5.1 version. |
| unzip [opencv.zip](https://www.youtube.com/redirect?event=comments&redir_token=QUFFLUhqbUNPLXdsNWRUTlpZaDN0SjlvX2dfY2M4ODMwQXxBQ3Jtc0trOG9MM2FBNW9JbU1HNk53YldkYTV4OFBJeU4yTThteWplb0V2T2RkaHYtRnNRNEJLMXJ5ZzJJVmFTNnd5YTNqUE1ubFBjT2wwUmhWLXhLZ0VJcnAxQ1JESG90OHF2cHp2N3lJa1RWdDZZbi1WZUJPMA&q=http%3A%2F%2Fopencv.zip%2F&stzid=UgzHycXony_VUGL5Alx4AaABAg) | Unzip OpenCV |
| unzip opencv\_[contrib.zip](https://www.youtube.com/redirect?event=comments&redir_token=QUFFLUhqbWJ2eHZUQUttUW1nOEJvWFppRC1UcnZ5dWdpUXxBQ3Jtc0trUWUwMUIxd25XNHJ0M2xXVGM3eVJwNG12bllCaEVleWlKOExBRlZ6SUU1MGJ5dVR1NmdSVjJBVmdtckFVQ2I0eGZjUDRDRWFsZFRaaEdZY1p2U29LR0RFQ3dYVXdrV0RFeTBrYlZMUy1zLU9Gb2NnYw&q=http%3A%2F%2Fcontrib.zip%2F&stzid=UgzHycXony_VUGL5Alx4AaABAg) | Unzip OpenCV contrib |

* 1. **Renaming the directories**

| **Commands** | **Description** |
| --- | --- |
| mv opencv-4.5.1 opencv | Move OpenCV 4.5.1 version to the folder named opencv. |
| mv opencv\_contrib-4.5.1 opencv\_contrib | Move OpenCV contrib 4.5.1 version to the folder named opencv\_contrib. |
| rm [opencv.zip](https://www.youtube.com/redirect?event=comments&redir_token=QUFFLUhqa21VT0pPWFdiTU5LSWJvT2VKczJkZmJLQzlWZ3xBQ3Jtc0ttZGd6YXlvRUotd2hrUjdhejNUbkNPY0VhT3dPMnhpcGVfY0dVSG0ycXF1SkgyS1d6N0lPd0Q5N2c4RjhBRmV2LTNCakh5UkFaWkMyMWhjOXc1WFktclRiMkdRVE1qWDdnSGN3cWhkUllXVmlQSVJaNA&q=http%3A%2F%2Fopencv.zip%2F&stzid=UgzHycXony_VUGL5Alx4AaABAg) | Remove zip files to save space. |
| rm opencv\_[contrib.zip](https://www.youtube.com/redirect?event=comments&redir_token=QUFFLUhqa2RrbnZUTWU3RXA2ME1RLVZZUkxaSWYzUmlMUXxBQ3Jtc0trVDhhRHdMS2pyQV8tdVU1N3FRYy13LXZrYVZyYUFUdHM1blVBV3YyQS1obmRKWDJMYmNqS3VRVGpfdmJoNzg1aGFWcXl1YXZzTWVRZkpIMEZUdGgzeDdqNlM2dzJvVUo1clVWYV9zRVUwcllUVUNhcw&q=http%3A%2F%2Fcontrib.zip%2F&stzid=UgzHycXony_VUGL5Alx4AaABAg) | Remove zip files to save space. |

* 1. **Building**

| **Commands** | **Description** |
| --- | --- |
| cd ~/opencv | Command directory to the opencv folder. |
| mkdir build | Make a build directory. |
| cd build | Command directory to the build directory. |

Next up, copy and paste this entire block of commands below, and command “make -j4”. Note that the full building process will take approximately 2 hours to complete.

*cmake -D CMAKE\_BUILD\_TYPE=RELEASE -D CMAKE\_INSTALL\_PREFIX=/usr -D OPENCV\_EXTRA\_MODULES\_PATH=~/opencv\_contrib/modules -D EIGEN\_INCLUDE\_PATH=/usr/include/eigen3 -D WITH\_OPENCL=OFF -D WITH\_CUDA=ON -D CUDA\_ARCH\_BIN=5.3 -D CUDA\_ARCH\_PTX="" -D WITH\_CUDNN=ON -D WITH\_CUBLAS=ON -D ENABLE\_FAST\_MATH=ON -D CUDA\_FAST\_MATH=ON -D OPENCV\_DNN\_CUDA=ON -D ENABLE\_NEON=ON -D WITH\_QT=OFF -D WITH\_OPENMP=ON -D WITH\_OPENGL=ON -D BUILD\_TIFF=ON -D WITH\_FFMPEG=ON -D WITH\_GSTREAMER=ON -D WITH\_TBB=ON -D BUILD\_TBB=ON -D BUILD\_TESTS=OFF -D WITH\_EIGEN=ON -D WITH\_V4L=ON -D WITH\_LIBV4L=ON -D OPENCV\_ENABLE\_NONFREE=ON -D INSTALL\_C\_EXAMPLES=OFF -D INSTALL\_PYTHON\_EXAMPLES=OFF -D BUILD\_NEW\_PYTHON\_SUPPORT=ON -D BUILD\_opencv\_python3=TRUE -D OPENCV\_GENERATE\_PKGCONFIG=ON -D BUILD\_EXAMPLES=OFF ..*

Once completed, the next steps are carried out to finish the installation process and remove unwanted junk files to clear up the memory space.

| **Commands** | **Description** |
| --- | --- |
| cd ~ | Line 1 |
| sudo rm -r /usr/include/opencv4/opencv2 | Line 2 |
| cd ~/opencv/build | Line 3 |
| sudo make install | Line 4 |
| sudo ldconfig | Line 5 |
| make clean | Line 6 |
| sudo apt-get update | Line 7 |

* 1. **Verifying**

It is important to verify the installation of the compatible version before running any related computer vision code. To verify, run the following commands and check that the OpenCV version printed is 4.5.1:

| **Commands** | **Description** |
| --- | --- |
| python3 | Open up python3. |
| import cv2 | Import cv2 into the python environment |
| cv2.\_*version\_* | Print out the OpenCV version. |

Following the steps described above, we should now have the compatible version of OpenCV – 4.5.1 installed in our Jetson Nano.

## **Camera Module Set-up**

The camera module we will be using for our GDP would be the **IMX219-77 Camera**.

#### Connecting the Camera Module to Jetson Nano

Prior to connecting a camera module to the Jetson Nano, ensure that the power supply is not connected. Lift the plastic tabs of the CSI camera connector slot gently and place the ribbon cable fully into the slot. Once the ribbon cable is in place, gently press down the plastic tabs to lock them in place. Key to note that the blue marking should face towards you or outside of the board. For a more detailed step-by-step visual, refer to the link below:

* <https://www.youtube.com/watch?v=dHvb225Pw1s>

#### Testing the Camera Module

Once the camera module is fully connected, power up the Jetson Nano and open the terminal window and run the following commands. The testing process will require us to capture a photo via real-life video using the camera module.

| **Commands** | **Description** |
| --- | --- |
| ls /dev/video0 | Outputs the functioning CSI camera. |
| ls /dev/video\* | Outputs a list of functioning cameras. |
| nvgstcapture-1.0 --orientation=2 | Turns on the camera and captures real-life video. Note that the orientation can be easily altered based on your preference. |

The orientation of the camera module set-up is important, as it should be based on the position of the camera module being mounted relative to the drone.

Up till this current step, we have a functioning camera module powered by the Jetson Nano, along with a compatible version of OpenCV being installed in this device. However, further steps will be required in setting-up the code to link OpenCV to this specific camera, which will be covered in the next section below.

## OpenCV & Camera Module Compatibility