

# Install OpenCV-Python in Fedora

## Goals

In this tutorial

- We will learn to setup OpenCV-Python in your Fedora system. Below steps are tested for Fedora 18 (64-bit) and Fedora 19 (32-bit).

## Introduction

OpenCV-Python can be installed in Fedora in two ways, 1) Install from pre-built binaries available in fedora repositories, 2) Compile from the source. In this section, we will see both.

Another important thing is the additional libraries required. OpenCV-Python requires only **Numpy** (in addition to other dependencies, which we will see later). But in this tutorials, we also use **Matplotlib** for some easy and nice plotting purposes (which I feel much better compared to OpenCV). Matplotlib is optional, but highly recommended. Similarly we will also see **IPython**, an Interactive Python Terminal, which is also highly recommended.

## Installing OpenCV-Python from Pre-built Binaries

Install all packages with following command in terminal as root.

```
$ yum install numpy opencv*
```

Open Python IDLE (or IPython) and type following codes in Python terminal.

```
>>> import cv2 as cv
>>> print( cv.__version__ )
```

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

It is quite easy. But there is a problem with this. Yum repositories may not contain the latest version of OpenCV always. For example, at the time of writing this tutorial, yum repository contains 2.4.5 while latest OpenCV version is 2.4.6. With respect to Python API, latest version will always contain much better support. Also, there may be chance of problems with camera support, video playback etc depending upon the drivers, ffmpeg, gstreamer packages present etc.

So my personal preference is next method, i.e. compiling from source. Also at some point in time, if you want to contribute to OpenCV, you will need this.

## Installing OpenCV from source

Compiling from source may seem a little complicated at first, but once you succeeded in it, there is nothing complicated.

First we will install some dependencies. Some are compulsory, some are optional. Optional dependencies, you can leave if you don't want.

### Compulsory Dependencies

We need **CMake** to configure the installation, **GCC** for compilation, **Python-devel** and **Numpy** for creating Python extensions etc.

```
yum install cmake
yum install python-devel numpy
yum install gcc gcc-c++
```

Next we need **GTK** support for GUI features, Camera support (libdc1394, v4l), Media Support (ffmpeg, gstreamer) etc.

```
yum install gtk2-devel
yum install libdc1394-devel
yum install ffmpeg-devel
yum install gstreamer-plugins-base-devel
```

### Optional Dependencies

Above dependencies are sufficient to install OpenCV in your fedora machine. But depending upon your requirements, you may need some extra dependencies. A list of such optional dependencies are given below. You can either leave it or install it, your call :)

OpenCV comes with supporting files for image formats like PNG, JPEG, JPEG2000, TIFF, WebP etc. But it may be a little old. If you want to get latest libraries, you can install development files for these formats.

```
yum install libpng-devel
yum install libjpeg-turbo-devel
yum install jasper-devel
yum install openexr-devel
yum install libtiff-devel
yum install libwebp-devel
```

Several OpenCV functions are parallelized with **Intel's Threading Building Blocks** (TBB). But if you want to enable it, you need to install TBB first. ( Also while configuring installation with CMake, don't forget to pass -D WITH\_TBB=ON. More details below.)

```
yum install tbb-devel
```

OpenCV uses another library **Eigen** for optimized mathematical operations. So if you have Eigen installed in your system, you can exploit it. ( Also while configuring installation with CMake, don't forget to pass -D WITH\_EIGEN=ON. More details below.)

```
yum install eigen3-devel
```

If you want to build **documentation** ( *Yes, you can create offline version of OpenCV's complete official documentation in your system in HTML with full search facility so that you need not access internet always if any question, and it is quite FAST!!!* ), you need to install **Doxygen** (a documentation generation tool).

```
yum install doxygen
```

## Downloading OpenCV

Next we have to download OpenCV. You can download the latest release of OpenCV from [sourceforge site](#). Then extract the folder.

Or you can download latest source from OpenCV's github repo. (If you want to contribute to OpenCV, choose this. It always keeps your OpenCV up-to-date). For that, you need to install **Git** first.

```
yum install git
git clone https://github.com/opencv/opencv.git
```

It will create a folder OpenCV in home directory (or the directory you specify). The cloning may take some time depending upon your internet connection.

Now open a terminal window and navigate to the downloaded OpenCV folder. Create a new build folder and navigate to it.

```
mkdir build
cd build
```

## Configuring and Installing

Now we have installed all the required dependencies, let's install OpenCV. Installation has to be configured with CMake. It specifies which modules are to be installed, installation path, which additional libraries to be used, whether documentation and examples to be compiled etc. Below command is normally used for configuration (executed from build folder).

```
cmake -D CMAKE_BUILD_TYPE=RELEASE -D CMAKE_INSTALL_PREFIX=/usr/local ..
```

It specifies that build type is "Release Mode" and installation path is /usr/local. Observe the -D before each option and .. at the end. In short, this is the format:

```
cmake [-D <flag>] [-D <flag>] ..
```

You can specify as many flags you want, but each flag should be preceded by -D.

So in this tutorial, we are installing OpenCV with TBB and Eigen support. We also build the documentation, but we exclude Performance tests and building samples. We also disable GPU related modules (since we use OpenCV-Python, we don't need GPU related modules. It saves us some time).

\*(All the below commands can be done in a single cmake statement, but it is split here for better understanding.)\*

- Enable TBB and Eigen support:

```
cmake -D WITH_TBB=ON -D WITH_EIGEN=ON ..
```

- Enable documentation and disable tests and samples

```
cmake -D BUILD_DOCS=ON -D BUILD_TESTS=OFF -D BUILD_PERF_TESTS=OFF -D BUILD_EXAMPLES=OFF ..
```

- Disable all GPU related modules.

```
cmake -D WITH_OPENCV=OFF -D BUILD_opencv_gpu=OFF -D BUILD_opencv_gpuarithm=OFF -D BUILD_opencv_gpubgsegm=OFF -D
BUILD_opencv_gpucodec=OFF -D BUILD_opencv_gpufeatures2d=OFF -D BUILD_opencv_gpufilters=OFF -D
BUILD_opencv_gpuimgproc=OFF -D BUILD_opencv_gpulegacy=OFF -D BUILD_opencv_gpuoptflow=OFF -D
BUILD_opencv_gpustereo=OFF -D BUILD_opencv_gpuwarping=OFF ..
```

- Set installation path and build type

```
cmake -D CMAKE_BUILD_TYPE=RELEASE -D CMAKE_INSTALL_PREFIX=/usr/local ..
```

Each time you enter cmake statement, it prints out the resulting configuration setup. In the final setup you got, make sure that following fields are filled (below is the some important parts of configuration I got). These fields should be filled appropriately in your system also. Otherwise some problem has happened. So check if you have correctly performed above steps.

```
...
-- GUI:
--   GTK+ 2.x:                YES (ver 2.24.19)
--   GThread :                YES (ver 2.36.3)
--
-- Video I/O:
```

```
-- DC1394 2.x: YES (ver 2.2.0)
-- FFMPEG: YES
--   codec: YES (ver 54.92.100)
--   format: YES (ver 54.63.104)
--   util: YES (ver 52.18.100)
--   swscale: YES (ver 2.2.100)
--   gentoo-style: YES
-- GStreamer:
--   base: YES (ver 0.10.36)
--   video: YES (ver 0.10.36)
--   app: YES (ver 0.10.36)
--   riff: YES (ver 0.10.36)
--   pbutils: YES (ver 0.10.36)
--
-- V4L/V4L2: Using libv4l (ver 1.0.0)
--
-- Other third-party libraries:
--   Use Eigen: YES (ver 3.1.4)
--   Use TBB: YES (ver 4.0 interface 6004)
--
-- Python:
--   Interpreter: /usr/bin/python2 (ver 2.7.5)
--   Libraries: /lib/libpython2.7.so (ver 2.7.5)
--   numpy: /usr/lib/python2.7/site-packages/numpy/core/include (ver 1.7.1)
--   packages path: lib/python2.7/site-packages
...

```

Many other flags and settings are there. It is left for you for further exploration.

Now you build the files using make command and install it using make install command. make install should be executed as root.

```
make
su
make install

```

Installation is over. All files are installed in /usr/local/ folder. But to use it, your Python should be able to find OpenCV module. You have two options for that.

1. **Move the module to any folder in Python Path** : Python path can be found out by entering `import sys; print(sys.path)` in Python terminal. It will print out many locations. Move /usr/local/lib/python2.7/site-packages/cv2.so to any of this folder. For example,
 

```
su mv /usr/local/lib/python2.7/site-packages/cv2.so /usr/lib/python2.7/site-packages
```

But you will have to do this every time you install OpenCV.

2. **Add /usr/local/lib/python2.7/site-packages to the PYTHON\_PATH**: It is to be done only once. Just open/.bashrc and add following line to it, then log out and come back.

```
export PYTHONPATH=$PYTHONPATH:/usr/local/lib/python2.7/site-packages
```

Thus OpenCV installation is finished. Open a terminal and try 'import cv2 as cv'.

To build the documentation, just enter following commands:

```
make doxygen
```

Then open opencv/build/doc/doxygen/html/index.html and bookmark it in the browser.

## Additional Resources

## Exercises

1. Compile OpenCV from source in your Fedora machine.