**Reference:** <https://cv-tricks.com/installation/opencv-4-1-ubuntu18-04/>

If you want to do with virtual env

# **Compiling OpenCV with Gstreamer and contrib modules:**

## Installation:

# Installing gstreamer

> sudo apt-get update

> sudo apt-get install gstreamer1.0\*

> sudo apt install ubuntu-restricted-extras

> sudo apt install libgstreamer1.0-dev libgstreamer-plugins-base1.0-dev

> sudo apt-get install libgstreamer1.0-0 gstreamer1.0-plugins-base gstreamer1.0-plugins-good gstreamer1.0-plugins-bad gstreamer1.0-plugins-ugly gstreamer1.0-libav gstreamer1.0-doc gstreamer1.0-tools gstreamer1.0-x gstreamer1.0-alsa gstreamer1.0-gl gstreamer1.0-gtk3 gstreamer1.0-qt5 gstreamer1.0-pulseaudio

# Install required dependencies

> sudo apt-get update -y

> sudo apt-get remove -y x264 libx264-dev

> sudo apt-get install -y build-essential checkinstall cmake pkg-config yasm

> sudo apt-get install -y git gfortran

> sudo add-apt-repository -y "deb http://security.ubuntu.com/ubuntu xenial-security main"

> sudo apt-get install -y libjpeg8-dev libjasper-dev libpng12-dev

> sudo apt-get install -y libtiff5-dev

> sudo apt-get install -y libavcodec-dev libavformat-dev libswscale-dev libdc1394-22-dev

> sudo apt-get install -y libxine2-dev libv4l-dev

> sudo apt-get install -y libgstreamer1.0-dev libgstreamer-plugins-base1.0-dev

> sudo apt-get install -y qt5-default libgtk2.0-dev libtbb-dev

> sudo apt-get install -y libatlas-base-dev

> sudo apt-get install -y libfaac-dev libmp3lame-dev libtheora-dev

> sudo apt-get install -y libvorbis-dev libxvidcore-dev

> sudo apt-get install -y libopencore-amrnb-dev libopencore-amrwb-dev

> sudo apt-get install -y x264 v4l-utils

> sudo apt-get install -y libprotobuf-dev protobuf-compiler

> sudo apt-get install -y libgoogle-glog-dev libgflags-dev

> sudo apt-get install -y libgphoto2-dev libeigen3-dev libhdf5-dev doxygen

> sudo apt install python3-pip

> sudo -H pip3 install -U pip numpy

> sudo pip3 install numpy scipy matplotlib scikit-image scikit-learn ipython

# Download OpenCV from Github

> cd ~/Desktop/

> git clone https://github.com/opencv/opencv.git

> cd opencv

> git checkout 4.1.0

> cd ..

# Download OpenCV\_contrib from Github

> cd ~/Desktop/

> git clone https://github.com/opencv/opencv\_contrib.git

> cd opencv\_contrib

> git checkout 4.1.0

> cd ..

# Build

> cd ~/Desktop/opencv

> mkdir build

> cd build

# Cmake file generation

> cmake -D CMAKE\_BUILD\_TYPE=RELEASE \

-D CMAKE\_INSTALL\_PREFIX=/usr/local \

-D INSTALL\_C\_EXAMPLES=ON \

-D INSTALL\_PYTHON\_EXAMPLES=ON \

-D WITH\_TBB=ON \

-D WITH\_V4L=ON \

-D WITH\_QT=ON \

-D WITH\_OPENGL=ON \

-D OPENCV\_EXTRA\_MODULES\_PATH=/home/sarthak/Desktop/opencv\_contrib/modules \

-D BUILD\_EXAMPLES=ON \

-D OPENCV\_GENERATE\_PKGCONFIG=YES \

-D WITH\_GSTREAMER=ON ..

# To find the number of threads compatible in your machine run the following command.

# make -j8 will take time in compiling

> nproc

> make -j8

> sudo make install

> sudo sh -c 'echo "/usr/local/lib" >> /etc/ld.so.conf.d/opencv.conf'

> sudo ldconfig

# Modifying OpenCV4

> cd ~/Desktop/opencv/build/unix-install/

> sudo nano opencv4.pc

The file will look something like this:

prefix=/usr/local

exec\_prefix=${prefix}

libdir=${exec\_prefix}/lib

includedir\_old=${prefix}/include/opencv4/opencv2 <= You have add 2 at the end

includedir\_new=${prefix}/include/opencv4

Name: OpenCV

Description: Open Source Computer Vision Library

Version: 4.1.0

Libs: -L${exec\_prefix}/lib -lopencv\_gapi -lopencv\_stitching -lopencv\_aruco -lopencv\_bgsegm -lopencv\_bioinspired -lopencv\_ccalib -lopencv\_cvv -lopencv\_dnn\_objdetect -lopencv\_dpm -lopencv\_face -lopencv\_freetype -lopencv\_fuzzy -lopencv\_hdf -lopencv\_hfs -lopencv\_img\_hash -lopencv\_line\_descriptor -lopencv\_quality -lopencv\_reg -lopencv\_rgbd -lopencv\_saliency -lopencv\_sfm -lopencv\_stereo -lopencv\_structured\_light -lopencv\_phase\_unwrapping -lopencv\_superres -lopencv\_optflow -lopencv\_surface\_matching -lopencv\_tracking -lopencv\_datasets -lopencv\_text -lopencv\_dnn -lopencv\_plot -lopencv\_videostab -lopencv\_video -lopencv\_xfeatures2d -lopencv\_shape -lopencv\_ml -lopencv\_ximgproc -lopencv\_xobjdetect -lopencv\_objdetect -lopencv\_calib3d -lopencv\_features2d -lopencv\_highgui -lopencv\_videoio -lopencv\_imgcodecs -lopencv\_flann -lopencv\_xphoto -lopencv\_photo -lopencv\_imgproc -lopencv\_core

Libs.private: -ldl -lm -lpthread -lrt -L/usr/lib/x86\_64-linux-gnu -lGL -lGLU

Cflags: -I${includedir\_old} -I${includedir\_new}

> cd /usr/local/lib/

> mkdir pkgconfig

> sudo cp ~/Desktop/opencv/build/unix-install/opencv4.pc /usr/local/lib/pkgconfig/

# Edit bash rc

> sudo nano ~/.bashrc

# Add the following 2 lines at the end of the file i.e. copy the following lines at the end of .bashrc file.

PKG\_CONFIG\_PATH=$PKG\_CONFIG\_PATH:/usr/local/lib/pkgconfig

export PKG\_CONFIG\_PATH

> source ~/.bashrc

# You can verify that the path is added or not by executing the following command

> echo $PKG\_CONFIG\_PATH

## C++ Verification:

Please copy the following code into the newly created file and follow the instructions accordingly (test.cpp)

#include "opencv.hpp"

using namespace cv;

using namespace std;

int main( int argc, char\*\* argv )

{

cout << "OpenCV version : " << CV\_VERSION << endl;

cout << "Major version : " << CV\_MAJOR\_VERSION << endl;

cout << "Minor version : " << CV\_MINOR\_VERSION << endl;

cout << "Subminor version : " << CV\_SUBMINOR\_VERSION << endl;

}

# C++ Verification

> cd ~/Desktop/test\_code/

#Compile the program to generate the executable file

> g++ -std=c++11 test.cpp `pkg-config --libs --cflags opencv4` -o result

#Execute the program.

> ./result

## Python3 Verification:

Please copy the following code into the newly created file and follow the instructions accordingly test.py. Make it executable

import cv2

print("OpenCV version : {0}".format(cv2.\_\_version\_\_))

major\_ver, minor\_ver, subminor\_ver = (cv2.\_\_version\_\_).split('.')

print("Major version : {0}".format(major\_ver))

print("Minor version : {0}".format(minor\_ver))

print("Subminor version : {0}".format(subminor\_ver))

> cd ~/Desktop/test\_code/

> chmod +x test.py

> python3 test.py