

# OpenCV 설치 및 사용방법 교안

최종 업데이트 - 2018. 10. 31

OpenCV 3.1.0 과 opencv\_contrib(extra modules)를 컴파일하여 Ubuntu 16.04 에 설치하는 방법을 설명합니다. 2018.10.31 현재 OpenCV 최신버전은 3.4.3 이지만, 본 교안에서는 걸음걸이 프로젝트에 사용되는 tf-pose-estimation 실행에 문제가 있는 관계로 3.1.0 버전을 사용한 점 참고하시기 바랍니다.

## - 목차 -

1. OpenCV 2.4 버전 제거
2. OpenCV 컴파일 전 필요한 패키지 설치
3. OpenCV 설정과 컴파일 및 설치
4. OpenCV 설치 결과 확인
  - 4.1. C/C++
  - 4.2. Python

## 1. OpenCV 2.4 버전 제거

OpenCV 2.4 대 버전이 설치되어 있으면 새로 설치하는 OpenCV 3.1.0 버전이 제대로 동작하지 않기 때문에 제거해 주어야 합니다.

아래처럼 보이면 OpenCV 2.4 대 버전이 설치가 안되어 있는 상태이므로 다음 단계로 넘어가도 됩니다.

```
pir1@pir1-Precision-Tower-5810:~$ pkg-config --modversion opencv
Package opencv was not found in the pkg-config search path.
Perhaps you should add the directory containing `opencv.pc'
to the PKG_CONFIG_PATH environment variable
No package 'opencv' found
```

다음 명령으로 OpenCV 라이브러리 설정 파일을 포함해서 기존에 설치된 OpenCV 패키지를 삭제하고 진행해야 합니다.

```
pir1@pir1-Precision-Tower-5810:~$ sudo apt-get purge libopencv* python-opencv
pir1@pir1-Precision-Tower-5810:~$ sudo apt-get autoremove
```

## 2. OpenCV 컴파일 전 필요한 패키지 설치

OpenCV 를 컴파일하는데 사용하는 것들이 포함된 패키지들을 설치합니다.

**build-essential** 패키지에는 C/C++ 컴파일러와 관련 라이브러리, make 같은 도구들이 포함되어 있습니다.

**cmake** 는 컴파일 옵션이나 빌드된 라이브러리에 포함시킬 OpenCV 모듈 설정등을 위해 필요합니다.

```
pir1@pir1-Precision-Tower-5810:~$ sudo apt-get install build-essential cmake
```

**pkg-config** 는 프로그램 컴파일 및 링크시 필요한 라이브러리에 대한 정보를 메타파일(확장자가 .pc 인 파일)로부터 가져오는데 사용됩니다.

터미널에서 특정 라이브러리를 사용한 소스코드를 컴파일시 필요한 컴파일러 및 링커 플래그를 추가하는데 도움이 됩니다.

```
pir1@pir1-Precision-Tower-5810:~$ sudo apt-get install pkg-config
```

특정 포맷의 이미지 파일을 불러오거나 기록하기 위해 필요한 패키지들입니다.

```
pir1@pir1-Precision-Tower-5810:~$ sudo apt-get install libjpeg-dev libtiff5-dev libjasper-dev libpng12-dev
```

특정 코덱의 비디오 파일을 읽어오거나 기록하기 위해 필요한 패키지들입니다.

```
pir1@pir1-Precision-Tower-5810:~$ sudo apt-get install libavcodec-dev libavformat-dev libswscale-dev libxvidcore-dev libx264-dev libxine2-dev
```

Video4Linux 패키지는 리눅스에서 실시간 비디오 캡처를 지원하기 위한 디바이스 드라이버와 API 를 포함하고 있습니다.

```
pir1@pir1-Precision-Tower-5810:~$ sudo apt-get install libv4l-dev v4l-utils
```

GStreamer 는 비디오 스트리밍을 위한 라이브러리입니다.

```
pir1@pir1-Precision-Tower-5810:~$ sudo apt-get install libgstreamer1.0-dev libgst-plugins-base1.0-dev
```

OpenCV 에서는 highgui 모듈을 사용하여 자체적으로 윈도우 생성하여 이미지나 비디오들을 보여줄 수 있습니다.

윈도우 생성 등의 GUI 를 위해 gtk 또는 qt 를 선택해서 사용가능합니다.

여기서는 qt4 를 지정해주었습니다. QImage 와 Mat 간의 변환에는 영향을 주지 않습니다.

```
pir1@pir1-Precision-Tower-5810:~$ sudo apt-get install libqt4-dev
```

그외 선택 가능한 패키지는 다음과 같습니다.

libgtk2.0-dev

libgtk-3-dev

libqt5-dev

OpenGL 지원하기 위해 필요한 라이브러리입니다.

```
pir1@pir1-Precision-Tower-5810:~$ sudo apt-get install mesa-utils libgl1-mesa-dr  
i libqt4-opengl-dev
```

OpenCV 최적화를 위해 사용되는 라이브러리들입니다.

```
pir1@pir1-Precision-Tower-5810:~$ sudo apt-get install libatlas-base-dev gfortra  
n libeigen3-dev
```

python2.7-dev 와 python3-dev 패키지는 OpenCV-Python 바인딩을 위해 필요한 패키지들입니다.

Numpy 는 매트릭스 연산등을 빠르게 처리할 수 있어서 OpenCV 에서 사용됩니다.

```
pir1@pir1-Precision-Tower-5810:~$ sudo apt-get install python2.7-dev python3-dev  
python-numpy python3-numpy
```

### 3. OpenCV 설정과 컴파일 및 설치

소스 코드를 저장할 임시 디렉토리를 생성하여 이동 후, OpenCV 3.1.0 소스코드를 다운로드 받아 압축을 풀어줍니다.

```
pir1@pir1-Precision-Tower-5810:~/opencv$ wget -O opencv.zip https://github.com/o  
pencv/opencv/archive/3.1.0.zip
```

```
pir1@pir1-Precision-Tower-5810:~/opencv$ unzip opencv.zip
```

opencv\_contrib(extra modules) 소스코드를 다운로드 받아 압축을 풀어줍니다.

SIFT, SURF 등을 사용하기 위해 필요합니다.

```
pir1@pir1-Precision-Tower-5810:~/opencv$ wget -O opencv_contrib.zip https://github.com/  
opencv/opencv_contrib/archive/3.1.0.zip
```

```
pir1@pir1-Precision-Tower-5810:~/opencv$ unzip opencv_contrib.zip
```

다음처럼 두 개의 디렉토리가 생성됩니다.

```
pir1@pir1-Precision-Tower-5810:~/opencv$ ls -d */
opencv-3.1.0/  opencv_contrib-3.1.0/
```

opencv-3.1.0 디렉토리로 이동하여 build 디렉토리를 생성하고 build 디렉토리로 이동합니다.

컴파일은 build 디렉토리에서 이루어집니다.

```
pir1@pir1-Precision-Tower-5810:~/opencv$ cd opencv-3.1.0/
pir1@pir1-Precision-Tower-5810:~/opencv/opencv-3.1.0$ mkdir build
pir1@pir1-Precision-Tower-5810:~/opencv/opencv-3.1.0$ cd build
pir1@pir1-Precision-Tower-5810:~/opencv/opencv-3.1.0/build$
```

cmake 를 사용하여 OpenCV 컴파일 설정을 해줍니다.

```
pir1@pir1-Precision-Tower-5810:~/opencv/opencv-3.1.0/build$ cmake -D CMAKE_BUILD_TYPE=RELEASE \
> -D CMAKE_INSTALL_PREFIX=/usr/local \
> -D WITH_TBB=OFF \
> -D WITH_IPP=OFF \
> -D WITH_1394=OFF \
> -D BUILD_WITH_DEBUG_INFO=OFF \
> -D BUILD_DOCS=OFF \
> -D INSTALL_C_EXAMPLES=ON \
> -D INSTALL_PYTHON_EXAMPLES=ON \
> -D BUILD_EXAMPLES=OFF \
> -D BUILD_TESTS=OFF \
> -D BUILD_PERF_TESTS=OFF \
> -D WITH_QT=ON \
> -D WITH_OPENGL=ON \
> -D OPENCV_EXTRA_MODULES_PATH=../../opencv_contrib-3.1.0/modules \
> -D WITH_V4L=ON \
> -D WITH_FFMPEG=ON \
> -D WITH_XINE=ON \
> -D BUILD_NEW_PYTHON_SUPPORT=ON \
> ../
```

다음처럼 Python 3 라이브러리 항목(Libraries, numpy, packages path 등)이 보이지 않는 경우에는

```
-- Python 2:
-- Interpreter:          /usr/bin/python2.7 (ver 2.7.12)
-- Libraries:           /usr/lib/x86_64-linux-gnu/libpython2.7.so (ver 2.7.12)
-- numpy:               /usr/lib/python2.7/dist-packages/numpy/core/include (ver 1.1
1.0)
-- packages path:       lib/python2.7/dist-packages
--
-- Python 3:
-- Interpreter:          /home/pirl/anaconda3/bin/python3 (ver 3.6.5)
--
-- Python (for build):   /usr/bin/python2.7
```

해당 경로들을 직접 적어줍니다.

```
-D PYTHON3_INCLUDE_DIR=/usr/include/python3.6.5m \
-D PYTHON3_NUMPY_INCLUDE_DIRS=/usr/lib/python3/dist-packages/numpy/core/include/ \
-D PYTHON3_PACKAGES_PATH=/usr/lib/python3.6.5/dist-packages \
-D PYTHON3_LIBRARY=/usr/lib/x86_64-linux-gnu/libpython3.6.5m.so \
```

컴파일을 시작하기 전에 사용 중인 컴퓨터의 CPU 코어수를 확인합니다.

```
pir1@pir1-Precision-Tower-5810:~/opencv/opencv-3.1.0/build$ cat /proc/cpuinfo | grep processor | wc -l
8
```

여기서 8은 물리적 코어수가 아니라 논리적 코어수입니다. 즉, 4코어 8스레드 CPU라는 뜻입니다.

make 명령을 사용하여 컴파일을 시작합니다.

-j 다음에 위에서 확인한 숫자를 붙여서 실행해줍니다.

```
pir1@pir1-Precision-Tower-5810:~/opencv/opencv-3.1.0/build$ make -j8
```

컴파일 결과물을 설치합니다.

```
pir1@pir1-Precision-Tower-5810:~/opencv/opencv-3.1.0/build$ sudo make install
```

다음과 같이 **/etc/ld.so.conf.d/** 디렉토리에 **/usr/local/lib**를 포함하는 설정파일이 있는지 확인합니다.

```
pir1@pir1-Precision-Tower-5810:~/opencv/opencv-3.1.0/build$ cat /etc/ld.so.conf.d/*
/usr/lib/x86_64-linux-gnu/libfakeroot
# Multiarch support
/lib/i386-linux-gnu
/usr/lib/i386-linux-gnu
/lib/i686-linux-gnu
/usr/lib/i686-linux-gnu
# libc default configuration
/usr/local/lib
# Multiarch support
/lib/x86_64-linux-gnu
/usr/lib/x86_64-linux-gnu
/usr/lib/nvidia-384
/usr/lib32/nvidia-384
/usr/lib/nvidia-384
/usr/lib32/nvidia-384
# Legacy biarch compatibility support
/lib32
/usr/lib32
# Legacy biarch compatibility support
/libx32
/usr/libx32
```

**/usr/local/lib**이 출력되지 않았다면 다음 명령을 추가로 실행해야 합니다.

```
pir1@pir1-Precision-Tower-5810:~/opencv/opencv-3.1.0/build$ sudo sh -c 'echo "/usr/local/lib" > /etc/ld.so.conf.d/opencv.conf'
```

**/usr/local/lib**을 찾은 경우나 못찾아서 추가한 작업을 한 경우 모두 컴파일시 opencv 라이브러리를 찾을 수 있도록 다음 명령을 실행합니다.

```
pir1@pir1-Precision-Tower-5810:~/opencv/opencv-3.1.0/build$ sudo ldconfig
```

## 4. OpenCV 설치 결과 확인

### 4.1. C/C++

C/C++를 위해 OpenCV 라이브러리 사용가능 여부를 확인합니다.

문제 없으면 설치된 OpenCV 라이브러리의 버전이 출력되는데, 아래처럼 **opencv** 를 찾을 수 없다고 나오면 추가 작업이 필요합니다.

```
pir1@pir1-Precision-Tower-5810:~/opencv/opencv-3.1.0/build$ pkg-config --modversion opencv
Package opencv was not found in the pkg-config search path.
Perhaps you should add the directory containing 'opencv.pc'
to the PKG_CONFIG_PATH environment variable
No package 'opencv' found
```

pkg-config 명령이 **/usr/local/lib/pkgconfig** 경로에 있는 opencv.pc 파일을 찾을 수 있도록 해줘야 합니다.

```
pir1@pir1-Precision-Tower-5810:~/opencv/opencv-3.1.0/build$ sudo sh -c 'echo PKG_CONFIG_PATH=$PKG_CONFIG_PATH:/usr/local/lib/pkgconfig >> /etc/bash.bashrc'
pir1@pir1-Precision-Tower-5810:~/opencv/opencv-3.1.0/build$ sudo sh -c 'echo export PKG_CONFIG_PATH >> /etc/bash.bashrc'
```

로그아웃했다가 로그인하면 적용이됩니다. 다시 확인했을 때, OpenCV 버전이 출력되어야 합니다.

```
pir1@pir1-Precision-Tower-5810:~/opencv/opencv-3.1.0/build$ pkg-config --modversion opencv
3.1.0
```

**pkg-config --libs --cflags opencv** 명령을 실행했을 때 다음처럼 나오면 문제없이 설치가 완료된 것입니다.

```
pir1@pir1-Precision-Tower-5810:~/opencv/opencv-3.1.0/build$ pkg-config --libs --cflags opencv
-I/usr/local/include/opencv -I/usr/local/include -L/usr/local/lib -lopencv_cudabgsegm -lopencv_cudaobjdetect -lopencv_cudastereo -lopencv_stitching -lopencv_cudafeatures2d -lopencv_superres -lopencv_cudacodec -lopencv_videostab -lopencv_cudaoptflow -lopencv_cudalegacy -lopencv_cudawarping -lopencv_aruco -lopencv_bgsegm -lopencv_bioinspired -lopencv_calib -lopencv_cvv -lopencv_dpm -lopencv_face -lopencv_photo -lopencv_cudaimgproc -lopencv_cudafilters -lopencv_freetype -lopencv_fuzzy -lopencv_hdf -lopencv_img_hash -lopencv_line_descriptor -lopencv_optflow -lopencv_reg -lopencv_rgbd -lopencv_saliency -lopencv_stereo -lopencv_structured_light -lopencv_phase_unwrapping -lopencv_surface_matching -lopencv_tracking -lopencv_datasets -lopencv_text -lopencv_dnn -lopencv_plot -lopencv_xfeatures2d -lopencv_shape -lopencv_video -lopencv_ml -lopencv_cudaarithm -lopencv_ximgproc -lopencv_calib3d -lopencv_features2d -lopencv_highgui -lopencv_videoio -lopencv_flann -lopencv_xobjdetect -lopencv_imgcodecs -lopencv_objdetect -lopencv_xphoto -lopencv_imgproc -lopencv_core -lopencv_cudev
```



/usr/local/share/OpenCV/samples/ 경로에 OpenCV 예제코드들이 설치됩니다.

```
pir1@pir1-Precision-Tower-5810:~/opencv/opencv-3.1.0/build$ ls /usr/local/share/OpenCV/samples
aruco          cvv            fuzzy          phase_unwrapping  saliency        text
bgsegm         datasets      gpu            plot               stereo          tracking
bioinspired    dnn           img_hash       python             structured_light xfeatures2d
ccalib         dpm           line_descriptor reg                surface_matching ximgproc
cpp            face          optflow        rgbd               tapi            xphoto
```

facedetect 예제 코드를 다음과 같이 컴파일 합니다.

```
$ g++ -o facedetect facedetect.cpp $(pkg-config --libs --cflags opencv)
$ ./facedetect --cascade="/usr/local/share/OpenCV/haarcascades/haarcascade_frontalface_alt.xml" --nested-cascade="/usr/local/share/OpenCV/haarcascades/haarcascade_eye_tree_eyeglasses.xml" --scale=1.3
```

## 4.2. Python

python 에서 opencv 라이브러리를 사용가능한지는 다음처럼 확인합니다.

```
pir1@pir1-Precision-Tower-5810:~/opencv$ python
Python 3.6.5 |Anaconda, Inc.] (default, Apr 29 2018, 16:14:56)
[GCC 7.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import cv2
>>> cv2.__version__
'3.1.0'
```

예제 코드를 실행하면 터미널에 표시되면서 카메라 영상에 얼굴이 검출된 결과를 얻을 수 있습니다.

```
$ python /usr/local/share/OpenCV/samples/python/facedetect.py --cascade "/usr/local/share/OpenCV/haarcascades/haarcascade_frontalface_alt.xml" --nested-cascade "/usr/local/share/OpenCV/haarcascades/haarcascade_eye_tree_eyeglasses.xml" /dev/video0
```

컴파일에 사용했던 opencv 소스코드 디렉토리는 **rm -rf opencv** 로 삭제합니다.



이번에는 Anaconda 로 OpenCV 를 설치하는 것을 소개합니다. 만약 Ubuntu 데스크탑에 Anaconda 가 설치되어 있다면 다음 명령어 한 줄을 통해 설치됩니다. - 끝 -

```
conda install -c menpo opencv=3.1.0
```