

Yahboom Jetson Nano B01 Sub - Ubuntu 20.04

Jetpack 4.6.5, jtop, TensorRT 8.2.6, VPI,
opencv 4.5, pytorch 1.13, torchvision 0.14.0, onnxruntime-gpu 설치 설명서

작성자: 정도윤(rabbitsun2@gmail.com)

작성일자: 2024-09-13(금)

목차

1. Ubuntu 20.04에서 GCC, G++ 9버전에서 8버전으로 하향하기(CUDA 10.2)	3
2. Ubuntu 20.04에서 NVIDIA jetson APT 주석 풀어주기	4
3. Ubuntu 20.04에서 NVIDIA Jetpack 4.6.5 설치하기(Ubuntu 20.04, Jetson Nano만 가능) ...	5
4. Ubuntu 20.04에서 .bashrc 설치하기	6
5. Ubuntu 20.04에서 TensorRT 설치하기	7
6. Ubuntu 20.04에서 NVIDIA VPI 설치하기	8
7. Ubuntu 20.04에서 OpenCV 4.5(opencv-python 4.5) 수동 설치	11
8. Ubuntu 20.04에서 Pytorch 1.13 설치하기(Q-engineeering)	19
9. Ubuntu 20.04에서 torchvision 설치하기(수동 컴파일)	21
10. Ubuntu 20.04에서 onnxruntime-gpu 설치하기	22

1. Ubuntu 20.04에서 GCC, G++ 9버전에서 8버전으로 하향하기(CUDA 10.2)

```
# install gcc and g++ version 8
$ sudo apt-get install gcc-8 g++-8
# setup the gcc selector
$ sudo update-alternatives --install /usr/bin/gcc gcc /usr/bin/gcc-9 9
$ sudo update-alternatives --install /usr/bin/gcc gcc /usr/bin/gcc-8 8
# setup the g++ selector
$ sudo update-alternatives --install /usr/bin/g++ g++ /usr/bin/g++-9 9
$ sudo update-alternatives --install /usr/bin/g++ g++ /usr/bin/g++-8 8
# if you want to make a selection use these commands
$ sudo update-alternatives --config gcc
$ sudo update-alternatives --config g++
```

gcc, g++을 8버전으로 바꿔야 CUDA 10.2에서 사용할 수 있음.

2. Ubuntu 20.04에서 NVIDIA Jetson APT 주석 풀어주기

jetpack 4.5를 비롯한 vpi를 설치하려면 NVIDIA Jetson common APT의 주석을 풀어주어야 한다.

```
cd /etc/apt/sources.list.d
```

```
vim nvidia-l4t-apt-source.list
```

```
libao.conf          vim
libaudit.conf       vtrgb
libblockdev         vulkan
libnl-3             wgetrc
libpaper.d          wildmidi
lightdm             wpa_supplicant
lintianrc           wpa_supplicant.conf
locale.alias        xattr.conf
locale.gen          xdg
localtime           xfce4
logcheck            xml
login.defs          xrdp
logrotate.conf      zsh_command_not_found

root@ubuntu:/etc# cd apt
root@ubuntu:/etc/apt# ls
apt.conf.d  preferences.d  sources.list.d  trusted.gpg.d
auth.conf.d  sources.list  sources.list.distUpgrade
root@ubuntu:/etc/apt# cd sources.list.d/
root@ubuntu:/etc/apt/sources.list.d# ls
nvidia-l4t-apt-source.list  nvidia-l4t-apt-source.list.distUpgrade
root@ubuntu:/etc/apt/sources.list.d# vim nvidia-l4t-apt-source.list
root@ubuntu:/etc/apt/sources.list.d# pwd
/etc/apt/sources.list.d
root@ubuntu:/etc/apt/sources.list.d# vim nvidia-l4t-apt-source.list
```

그림 1. nvidia-l4t-apt-source.list 파일 수정하기

```
# SPDX-FileCopyrightText: Copyright (c) 2019–2021 NVIDIA CORPORATION & AFFILIATES. All rights reserved.  
# SPDX-License-Identifier: LicenseRef-NvidiaProprietary  
#  
# NVIDIA CORPORATION, its affiliates and licensors retain all intellectual  
# property and proprietary rights in and to this material, related  
# documentation and any modifications thereto. Any use, reproduction,  
# disclosure or distribution of this material and related documentation  
# without an express license agreement from NVIDIA CORPORATION or  
# its affiliates is strictly prohibited.
```

```
deb https://repo.download.nvidia.com/jetson/common r32.7 main # disabled  
on upgrade to focal
```

```
deb https://repo.download.nvidia.com/jetson/t210 r32.7 main # disabled  
on upgrade to focal
```

```
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
  
"nvidia-l4t-apr-source.list" 12L, 740C          12.1      모두
```

그림 2. nvidia-l4t-apt-source.list 파일 수정하기

deb 주석을 두 개를 풀어준다.

3. Ubuntu 20.04에서 NVIDIA Jetpack 4.6.5 설치하기(Ubuntu 20.04, Jetson Nano만 가능)

```
apt search nvidia-jetpack
```

```
apt-get install nvidia-jetpack
```

이 명령어로 설치가 가능하다.

실험 결과로 Jetson Nano에서는 NVIDIA CUDA 사이트에 접속해서 별도로 CUDA 10.2를 설치하는 것이 아니다.

4. Ubuntu 20.04에서 .bashrc 설치하기

Jetpack 4.6.5를 설치하면 CUDA 10.2가 /usr/local/cuda-10.2에 설치된다.

경로를 지정해줘야 한다.

그리고 Python 3.8도 ARMv8 프로세서를 사용한다고 지정을 해줘야 한다.

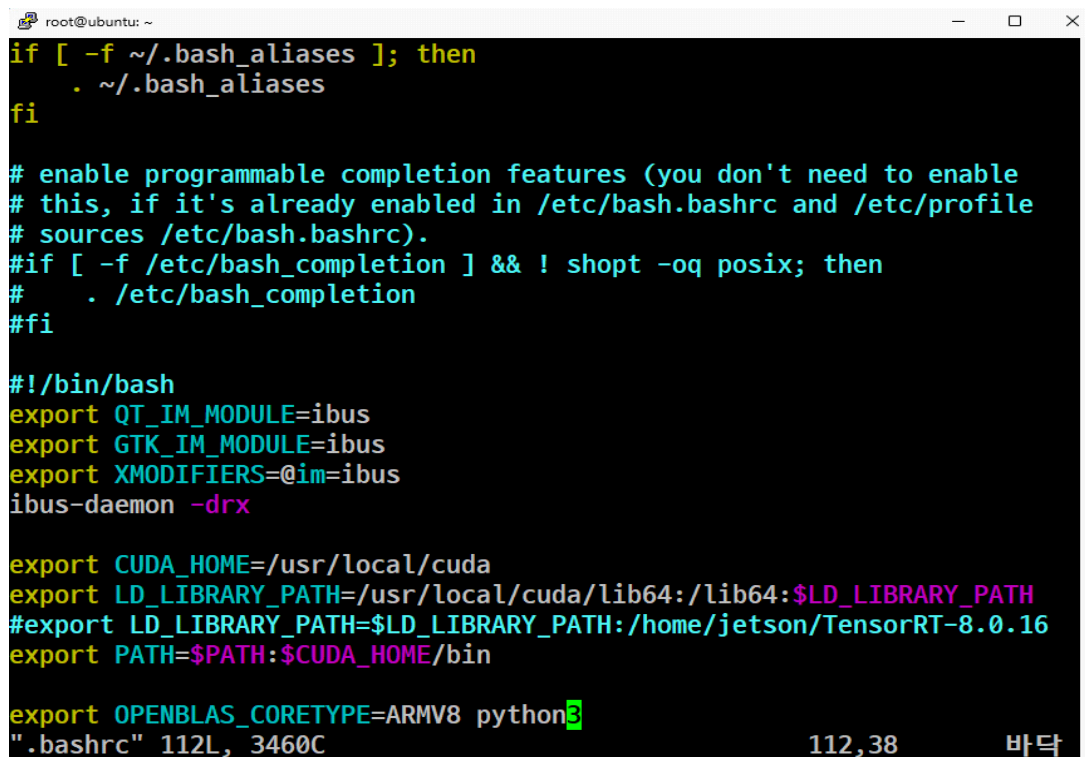
.bashrc 추가해야 할 내용

```
#!/bin/bash

export QT_IM_MODULE=ibus
export GTK_IM_MODULE=ibus
export XMODIFIERS=@im=ibus
ibus-daemon -drx

export CUDA_HOME=/usr/local/cuda
export LD_LIBRARY_PATH=/usr/local/cuda/lib64:/lib64:$LD_LIBRARY_PATH
#export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/home/jetson/TensorRT-8.0.16
export PATH=$PATH:$CUDA_HOME/bin

export OPENBLAS_CORETYPE=ARMV8 python3
```



```
root@ubuntu: ~
if [ -f ~/.bash_aliases ]; then
    . ~/.bash_aliases
fi

# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
#if [ -f /etc/bash_completion ] && ! shopt -oq posix; then
#    . /etc/bash_completion
#fi

#!/bin/bash
export QT_IM_MODULE=ibus
export GTK_IM_MODULE=ibus
export XMODIFIERS=@im=ibus
ibus-daemon -drx

export CUDA_HOME=/usr/local/cuda
export LD_LIBRARY_PATH=/usr/local/cuda/lib64:/lib64:$LD_LIBRARY_PATH
#export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/home/jetson/TensorRT-8.0.16
export PATH=$PATH:$CUDA_HOME/bin

export OPENBLAS_CORETYPE=ARMV8 python3
".bashrc" 112L, 3460C                               112,38      바닥
```

그림 3. .bashrc 파일 수정하기

5. Ubuntu 20.04에서 TensorRT 설치하기

수동 컴파일을 하지 말고, apt-get으로 제공되는 TensorRT로 설치하면 된다.

NVIDIA TensorRT 공식사이트에서 제공하는 tar 파일은 x86-64를 제공하므로 Jetson Nano에서 호환되지 않는다.

```
sudo apt-get install tensorrt
```

```
jetson_release -v
```

```
Vim: Finished.
```

```
jetson@ubuntu:~$ vim .bashrc
jetson@ubuntu:~$ ls
1_dev      Downloads  Public     examples.desktop
Desktop    Music      Templates  thinclient_drives
Documents  Pictures   Videos
jetson@ubuntu:~$ source .bashrc
jetson@ubuntu:~$ sudo -i
root@ubuntu:~# source .bashrc
root@ubuntu:~# sudo apt-get install tensorrt
패키지 목록을 읽는 중입니다... 완료
의존성 트리를 만드는 중입니다
상태 정보를 읽는 중입니다... 완료
다음 패키지가 자동으로 설치되었지만 더 이상 필요하지 않습니다:
  geoip-database libgeoip1
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  libnvinfer-bin libnvinfer-dev libnvinfer-doc libnvinfer-plugin-dev
  libnvinfer-plugin8 libnvinfer-samples libnvonnxparsers-dev
  libnvonnxparsers8 libnvparsers-dev libnvparsers8
다음 새 패키지를 설치할 것입니다:
  libnvinfer-bin libnvinfer-dev libnvinfer-doc libnvinfer-plugin-dev
  libnvinfer-plugin8 libnvinfer-samples libnvonnxparsers-dev
```

그림 4. TensorRT 설치하기

```
558 M바이트 아카이브를 받아야 합니다.
이 작업 후 988 M바이트의 디스크 공간을 더 사용하게 됩니다.
계속 하시겠습니까? [Y/n] Y
받기:1 https://repo.download.nvidia.com/jetson/common r32.7/main arm64
libnvinfer-plugin8 arm64 8.2.1-1+cuda10.2 [4,640 kB]
받기:2 https://repo.download.nvidia.com/jetson/common r32.7/main arm64
libnvparsers8 arm64 8.2.1-1+cuda10.2 [710 kB]
받기:3 https://repo.download.nvidia.com/jetson/common r32.7/main arm64
libnvonnxparsers8 arm64 8.2.1-1+cuda10.2 [620 kB]
받기:4 https://repo.download.nvidia.com/jetson/common r32.7/main arm64
libnvinfer-bin arm64 8.2.1-1+cuda10.2 [122 kB]
받기:5 https://repo.download.nvidia.com/jetson/common r32.7/main arm64
libnvinfer-dev arm64 8.2.1-1+cuda10.2 [81.3 MB]
받기:6 https://repo.download.nvidia.com/jetson/common r32.7/main arm64
libnvinfer-doc all 8.2.1-1+cuda10.2 [4,810 kB]
받기:7 https://repo.download.nvidia.com/jetson/common r32.7/main arm64
libnvinfer-plugin-dev arm64 8.2.1-1+cuda10.2 [4,704 kB]
받기:8 https://repo.download.nvidia.com/jetson/common r32.7/main arm64
libnvparsers-dev arm64 8.2.1-1+cuda10.2 [539 kB]
받기:9 https://repo.download.nvidia.com/jetson/common r32.7/main arm64
libnvonnxparsers-dev arm64 8.2.1-1+cuda10.2 [413 kB]
받기:10 https://repo.download.nvidia.com/jetson/common r32.7/main arm64
libnvinfer-samples all 8.2.1-1+cuda10.2 [460 MB]
77% [10 libnvinfer-samples 327 MB/460 MB 71%] 4,175 kB/s 31초
```

그림 5. TensorRT 설치하기

```

libnvinfer-plugin8 (8.2.1-1+cuda10.2) 설정하는 중입니다 ...
libnvinfer-plugin-dev (8.2.1-1+cuda10.2) 설정하는 중입니다 ...
libnvinfer-bin (8.2.1-1+cuda10.2) 설정하는 중입니다 ...
libnvinfer-samples (8.2.1-1+cuda10.2) 설정하는 중입니다 ...
tensorrt (8.2.1.9-1+cuda10.2) 설정하는 중입니다 ...
Processing triggers for libc-bin (2.31-0ubuntu9.16) ...
root@ubuntu:~# vim .bashrc
root@ubuntu:~# jetson_release -v
Software part of jetson-stats 4.2.10 - (c) 2024, Raffaello Bonghi
Model: NVIDIA Jetson Nano Developer Kit - Jetpack 4.6.5 [L4T 32.7.5]
NV Power Mode[0]: MAXN
Serial Number: [XXX Show with: jetson_release -s XXX]
Hardware:
- 699-level Part Number: 699-13448-0002-401 P.0
- P-Number: p3448-0002
- BoardIDs: p3448
- Module: NVIDIA Jetson Nano module (16Gb eMMC)
- SoC: tegra210
- CUDA Arch BIN: 5.3
- Codename: Porg
Platform:
- Machine: aarch64
- System: Linux
- Distribution: Ubuntu 20.04 focal

```

그림 6. TensorRT 설치하기

```

Hardware:
- 699-level Part Number: 699-13448-0002-401 P.0
- P-Number: p3448-0002
- BoardIDs: p3448
- Module: NVIDIA Jetson Nano module (16Gb eMMC)
- SoC: tegra210
- CUDA Arch BIN: 5.3
- Codename: Porg
Platform:
- Machine: aarch64
- System: Linux
- Distribution: Ubuntu 20.04 focal
- Release: 4.9.337-tegra
- Python: 3.8.10
jtop:
- Version: 4.2.10
- Service: Active
Libraries:
- CUDA: 10.2.300
- cuDNN: 8.2.1.32
- TensorRT: 8.2.1.9
- VPI: Not installed
- OpenCV: Not installed
root@ubuntu:~#

```

그림 7. TensorRT 설치하기

6. Ubuntu 20.04에서 NVIDIA VPI 설치하기

수동 컴파일을 하지 말고, apt-get으로 제공되는 libnvvpi1으로 설치하면 된다.

```
sudo apt install libnvvpi1
```

```
sudo jetson_release -v
```

```
root@ubuntu:~# sudo apt install libnvvpi1
패키지 목록을 읽는 중입니다... 완료
의존성 트리를 만드는 중입니다
상태 정보를 읽는 중입니다... 완료
다음 패키지가 자동으로 설치되었지만 더 이상 필요하지 않습니다:
  geoip-database libgeoip1
Use 'sudo apt autoremove' to remove them.
다음 새 패키지를 설치할 것입니다:
  libnvvpi1
0개 업그레이드, 1개 새로 설치, 0개 제거 및 5개 업그레이드 안 함.
17.2 M바이트 아카이브를 받아야 합니다.
이 작업 후 105 M바이트의 디스크 공간을 더 사용하게 됩니다.
받기:1 https://repo.download.nvidia.com/jetson/common r32.7/main arm64
libnvvpi1 arm64 1.2.3 [17.2 MB]
내려받기 17.2 M바이트, 소요시간 9초 (1,971 k바이트/초)
Selecting previously unselected package libnvvpi1.
(데이터베이스 읽는중 ...현재 200847개의 파일과 디렉터리가 설치되어 있습니다.)
Preparing to unpack .../libnvvpi1_1.2.3_arm64.deb ...
Unpacking libnvvpi1 (1.2.3) ...
libnvvpi1 (1.2.3) 설정하는 중입니다 ...
Processing triggers for libc-bin (2.31-0ubuntu9.16) ...
Progress: [ 80%] [#####.....]
```

그림 8. NVIDIA-VPI 설치하기

```
Selecting previously unselected package libnvvpi1.
(데이터베이스 읽는중 ...현재 200847개의 파일과 디렉터리가 설치되어 있습니다.)
Preparing to unpack .../libnvvpi1_1.2.3_arm64.deb ...
Unpacking libnvvpi1 (1.2.3) ...
libnvvpi1 (1.2.3) 설정하는 중입니다 ...
Processing triggers for libc-bin (2.31-0ubuntu9.16) ...
root@ubuntu:~# jetson_release -v
Software part of jetson-stats 4.2.10 - (c) 2024, Raffaello Bonghi
Model: NVIDIA Jetson Nano Developer Kit - Jetpack 4.6.5 [L4T 32.7.5]
NV Power Mode[0]: MAXN
Serial Number: [XXX Show with: jetson_release -s XXX]
Hardware:
- 699-level Part Number: 699-13448-0002-401 P.0
- P-Number: p3448-0002
- BoardIDs: p3448
- Module: NVIDIA Jetson Nano module (16Gb eMMC)
- SoC: tegra210
- CUDA Arch BIN: 5.3
- Codename: Porg
Platform:
- Machine: aarch64
- System: Linux
- Distribution: Ubuntu 20.04 focal
```

그림 9. NVIDIA-VPI 설치하기

```
Hardware:
- 699-level Part Number: 699-13448-0002-401 P.0
- P-Number: p3448-0002
- BoardIDs: p3448
- Module: NVIDIA Jetson Nano module (16Gb eMMC)
- SoC: tegra210
- CUDA Arch BIN: 5.3
- Codename: Porg
Platform:
- Machine: aarch64
- System: Linux
- Distribution: Ubuntu 20.04 focal
- Release: 4.9.337-tegra
- Python: 3.8.10
jtop:
- Version: 4.2.10
- Service: Active
Libraries:
- CUDA: 10.2.300
- cuDNN: 8.2.1.32
- TensorRT: 8.2.1.9
- VPI: 1.2.3
- OpenCV: Not installed
root@ubuntu:~#
```

그림 10. NVIDIA-VPI 설치하기

7. Ubuntu 20.04에서 OpenCV 4.5(opencv-python 4.5) 수동 설치

수동 컴파일로 OpenCV를 설치해야 aarch64에서 동작한다.

[1. 가상 메모리, 스왑 공간]

이미 했다면 안해도 됨

```
sudo apt-get update
```

```
sudo apt-get upgrade
```

```
sudo apt-get install nano
```

```
sudo apt-get install dphys-swapfile
```

/sbin/dphys-swapfile 파일 open

```
sudo nano /sbin/dphys-swapfile
```

Swap파일의 값이 다음과 같도록 값을 추가하거나, 파일 내 주석을 해제

```
# CONF_SWAPSIZE=4096
```

```
# CONF_SWAPFACTOR=2
```

```
# CONF_MAXSWAP=4096
```

값을 수정 후 [Ctrl] + [X], [y], [Enter]를 눌러 저장하고 닫으면 됨.

/etc/dphys-swapfile 파일 open

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

Swap파일의 값이 다음과 같도록 값을 추가하거나, 파일 내 주석을 해제

```
# CONF_SWAPSIZE=4096
```

```
# CONF_SWAPFACTOR=2
```

```
# CONF_MAXSWAP=4096
```

값을 수정 후 [Ctrl] + [X], [y], [Enter]를 눌러 저장하고 닫으면 됨.

재부팅

```
sudo reboot
```

[2. 기존에 설치 되어 있는 CUDA 버전 확인 후 삭제]

기존 cuda 버전 확인

```
pkg-config --modversion opencv
```

기존에 설치 된 cuda가 있다면 버전이 뜰 것.

설치 된 cuda가 없다면 'No package 'opencv' found' 출력 됨.

기존에 설치 된 cuda 버전 및 의존 패키지 삭제

```
sudo apt-get remove libopencv*
```

```
sudo apt-get autoremove
```

```
sudo find /usr/local -name "*opencv*" -exec rm {} \;
```

[3. 추가 패키지 설치]

```
sudo apt-get update
# sudo apt-get upgrade -> 이걸 자주 안하는 걸 권장한다 함..
sudo apt-get install -y build-essential cmake git unzip pkg-config
sudo apt-get install -y libjpeg-dev libpng-dev libtiff-dev
sudo apt-get install -y libavcodec-dev libavformat-dev libswscale-dev
sudo apt-get install -y libgtk2.0-dev libcanberra-gtk*
sudo apt-get install -y python3-dev python3-numpy python3-pip
sudo apt-get install -y python3-dev python3-numpy libtbb2 libtbb-dev
sudo apt-get install -y libxvidcore-dev libx264-dev libgtk-3-dev
sudo apt-get install -y libtbb2 libtbb-dev libdc1394-22-dev
sudo apt-get install -y libv4l-dev v4l-utils
sudo apt-get install -y libgstreamer1.0-dev libgstreamer-plugins-base1.0-dev
sudo apt-get install -y libavresample-dev libvorbis-dev libxine2-dev
sudo apt-get install -y libfaac-dev libmp3lame-dev libtheora-dev
sudo apt-get install -y libopencore-amrnb-dev libopencore-amrwb-dev
sudo apt-get install -y libopenblas-dev libatlas-base-dev libblas-dev
sudo apt-get install -y liblapack-dev libeigen3-dev gfortran
sudo apt-get install -y libhdf5-dev protobuf-compiler
sudo apt-get install -y libprotobuf-dev libgoogle-glog-dev libgflags-dev
sudo apt install ccache
```

[4. opencv zip file 다운 및 압축 해제]

```
wget -O opencv.zip https://github.com/opencv/opencv/archive/4.5.0.zip  
wget -O opencv_contrib.zip https://github.com/opencv/opencv_contrib/archive/4.5.0.zip
```

```
unzip opencv.zip  
unzip opencv_contrib.zip
```

[5. build 폴더 생성]

```
# 압축 해제 후 생성 된 opencv-4.5.0 폴더 안으로 이동  
cd opencv-4.5.0
```

```
# build 폴더 생성 후 이동  
mkdir build  
cd build
```

[6. 컴파일러 설치]

```
sudo apt-get install g++-9 (CUDA 10.2을 설치하면 9버전은 사용하면 안 된다.)
```

[7. cmake로 빌드하기] (1시간 30분 소요)

```
cmake -D CMAKE_BUILD_TYPE=RELEASE ₩
-D CMAKE_INSTALL_PREFIX=/usr/local ₩
-D OPENCV_EXTRA_MODULES_PATH=/root/opencv_contrib-4.5.0/modules ₩
-D EIGEN_INCLUDE_PATH=/usr/include/eigen3 ₩
-D WITH_OPENCLAMDBLAS=OFF ₩
-D WITH_GTK=OFF ₩
-D WITH_OPENCL=OFF ₩
-D WITH_CUDA=ON ₩
-D CUDA_FAST_MATH=ON ₩
-D OPENCV_DNN_CUDA=ON ₩
-D CUDA_ARCH_PTX="" ₩
-D WITH_CUDNN=ON ₩
-D WITH_CUBLAS=ON ₩
-D ENABLE_FAST_MATH=ON ₩
-D ENABLE_NEON=ON ₩
-D WITH_QT=OFF ₩
-D WITH_OPENMP=ON ₩
-D WITH_OPENGL=ON ₩
-D BUILD_JASPER=ON ₩
-D BUILD_TIFF=ON ₩
-D WITH_FFMPEG=ON ₩
-D WITH_GSTREAMER=ON ₩
-D WITH_TBB=ON ₩
-D BUILD_TBB=ON ₩
-D WITH_EIGEN=ON ₩
-D WITH_V4L=ON ₩
-D WITH_LIBV4L=ON ₩
-D WITH_VTK=ON ₩
-D BUILD_PYTHON=ON ₩
-D OPENCV_ENABLE_NONFREE=ON ₩
-D INSTALL_PYTHON_EXAMPLES=OFF ₩
-D BUILD_NEW_PYTHON_SUPPORT=ON ₩
-D OPENCV_GENERATE_PKGCONFIG=ON ₩
-D BUILD_OPENCV_WORLD=ON ₩
-D BUILD_PERF_TESTS=OFF ₩
-D BUILD_TESTS=OFF ₩
-D BUILD_OPENCV_PYTHON_TESTS=OFF ₩
-D BUILD_JAVA=OFF ₩
-D INSTALL_TESTS=OFF ₩
-D INSTALL_C_EXAMPLES=OFF ₩
-D BUILD_EXAMPLES=OFF ..
```

[8. 컴파일과 설치]

```
$ make -j4  
$ sudo rm -rf /usr/include/opencv4/opencv2  
$ sudo make install
```

[9. /usr/local/lib 경로가 출력되는지 확인해야 한다.]

```
$ cat /etc/ld.so.conf.d/*
```

만약 /usr/local/lib 출력되지 않으면 아래 명령어로 경로를 추가해 줍니다.

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

- 공유 라이브러리 정보를 갱신하여 경로 등이 적용되게 합니다.

```
$ sudo ldconfig
```

[10. jetson-release 확인(OpenCV 설치확인)]

```
jetson_release -v
```

[11. OpenCV-Python에서 버전 확인하기]

pip install로 opencv-python을 설치하면 안 된다.

```
python3  
import cv2  
cv2.__version__
```



```
root@ubuntu: ~/jetson/Downloads/opencv-4.5.0/build
root@ubuntu:~/jetson/Downloads/opencv-4.5.0/build# sudo ldconfig
root@ubuntu:~/jetson/Downloads/opencv-4.5.0/build# jetson_release -v
Software part of jetson-stats 4.2.10 - (c) 2024, Raffaello Bonghi
Model: NVIDIA Jetson Nano Developer Kit - Jetpack 4.6.5 [L4T 32.7.5]
NV Power Mode[0]: MAXN
Serial Number: [XXX Show with: jetson_release -s XXX]
Hardware:
- 699-level Part Number: 699-13448-0002-401 P.0
- P-Number: p3448-0002
- BoardIDs: p3448
- Module: NVIDIA Jetson Nano module (16Gb eMMC)
- SoC: tegra210
- CUDA Arch BIN: 5.3
- Codename: Porg
Platform:
- Machine: aarch64
- System: Linux
- Distribution: Ubuntu 20.04 focal
- Release: 4.9.337-tegra
- Python: 3.8.10
jtop:
- Version: 4.2.10
- Service: Active
Libraries:
```

그림 11. jetson_release로 OpenCV with CUDA 설치 확인

```
root@ubuntu: ~/jetson/Downloads/opencv-4.5.0/build
Hardware:
- 699-level Part Number: 699-13448-0002-401 P.0
- P-Number: p3448-0002
- BoardIDs: p3448
- Module: NVIDIA Jetson Nano module (16Gb eMMC)
- SoC: tegra210
- CUDA Arch BIN: 5.3
- Codename: Porg
Platform:
- Machine: aarch64
- System: Linux
- Distribution: Ubuntu 20.04 focal
- Release: 4.9.337-tegra
- Python: 3.8.10
jtop:
- Version: 4.2.10
- Service: Active
Libraries:
- CUDA: 10.2.300
- cuDNN: 8.2.1.32
- TensorRT: 8.2.1.9
- VPI: 1.2.3
- OpenCV: 4.5.0 - with CUDA: YES
root@ubuntu:~/jetson/Downloads/opencv-4.5.0/build#
```

그림 12. jetson_release로 OpenCV with CUDA 설치 확인

```

- Codename: Porg
Platform:
- Machine: aarch64
- System: Linux
- Distribution: Ubuntu 20.04 focal
- Release: 4.9.337-tegra
- Python: 3.8.10
jtop:
- Version: 4.2.10
- Service: Active
Libraries:
- CUDA: 10.2.300
- cuDNN: 8.2.1.32
- TensorRT: 8.2.1.9
- VPI: 1.2.3
- OpenCV: 4.5.0 - with CUDA: YES
root@ubuntu:~/jetson/Downloads/opencv-4.5.0/build# python3
Python 3.8.10 (default, Jul 29 2024, 17:02:10)
[GCC 9.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import cv2
>>> cv2.__version__
'4.5.0'
>>>

```

그림 13. jetson_release로 OpenCV with CUDA 설치 확인

8. Ubuntu 20.04에서 Pytorch 1.13 설치하기(Q-engineeering)

Yahboom Jetson Nano B01 Sub, NVIDIA Jetson Nano에서 우분투 20.04를 설치하면 해당 명령어는 정상적으로 동작한다.

```
# install the dependencies (if not already onboard)
$ sudo apt-get install python3-pip libjpeg-dev libopenblas-dev libopenmpi-dev libomp-dev
$ sudo -H pip3 install future
$ sudo pip3 install -U --user wheel mock pillow
$ sudo -H pip3 install testresources
# above 58.3.0 you get version issues
$ sudo -H pip3 install setuptools==58.3.0
$ sudo -H pip3 install Cython
# install gdown to download from Google drive
$ sudo -H pip3 install gdown
# download the wheel
$ gdown https://drive.google.com/uc?id=1e9FDGt2zGS5C5Pms7wzHYRb0HuupngK1
# install PyTorch 1.13.0
$ sudo -H pip3 install torch-1.13.0a0+git7c98e70-cp38-cp38-linux_aarch64.whl
# clean up
$ rm torch-1.13.0a0+git7c98e70-cp38-cp38-linux_aarch64.whl
```

python3

import torch

torch.cuda.is_available()

```
root@ubuntu:~/jetson/Downloads# ls -al
total 376260
drwxr-xr-x  4 jetson jetson    4096  9월 13 17:41 .
drwxr-xr-x 21 jetson jetson    4096  9월 13 17:17 ..
drwxr-xr-x 13 jetson jetson    4096  9월 13 14:31 opencv-4.5.0
-rw-r--r--  1 jetson jetson 94260098  9월 13 10:13 opencv.zip
drwxr-xr-x  6 jetson jetson    4096 10월 11 2020 opencv_contrib-4.5.0
-rw-r--r--  1 jetson jetson 62260147  9월 13 10:13 opencv_contrib.zip
-rw-r--r--  1 jetson jetson 228740406 12월 21 2022 torch-1.13.0a0+git7c98e70-cp
38-cp38-linux_aarch64.whl
root@ubuntu:~/jetson/Downloads# sudo -H pip3 install torch-1.13.0a0+git7c98e70-c
p38-cp38-linux_aarch64.whl
Processing ./torch-1.13.0a0+git7c98e70-cp38-cp38-linux_aarch64.whl
Collecting typing-extensions (from torch==1.13.0a0+git7c98e70)
  Downloading typing_extensions-4.12.2-py3-none-any.whl.metadata (3.0 kB)
Downloading typing_extensions-4.12.2-py3-none-any.whl (37 kB)
Installing collected packages: typing-extensions, torch
Successfully installed torch-1.13.0a0+git7c98e70 typing-extensions-4.12.2
WARNING: Running pip as the 'root' user can result in broken permissions and con
flicting behaviour with the system package manager, possibly rendering your syst
em unusable.It is recommended to use a virtual environment instead: https://pip.
pypa.io/warnings/venv. Use the --root-user-action option if you know what you ar
e doing and want to suppress this warning.
root@ubuntu:~/jetson/Downloads#
```

그림 14. pytorch 설치하기

```

-rw-r--r-- 1 jetson jetson 62260147 9월 13 10:13 opencv_contrib.zip
-rw-r--r-- 1 jetson jetson 228740406 12월 21 2022 torch-1.13.0a0+git7c98e70-cp38-cp38-linux_aarch64.whl
root@ubuntu:~/jetson/Downloads# sudo -H pip3 install torch-1.13.0a0+git7c98e70-cp38-cp38-linux_aarch64.whl
Processing ./torch-1.13.0a0+git7c98e70-cp38-cp38-linux_aarch64.whl
Collecting typing-extensions (from torch==1.13.0a0+git7c98e70)
  Downloading typing_extensions-4.12.2-py3-none-any.whl.metadata (3.0 kB)
Downloading typing_extensions-4.12.2-py3-none-any.whl (37 kB)
Installing collected packages: typing-extensions, torch
Successfully installed torch-1.13.0a0+git7c98e70 typing-extensions-4.12.2
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package manager, possibly rendering your system unusable. It is recommended to use a virtual environment instead: https://pip.pypa.io/warnings/venv. Use the --root-user-action option if you know what you are doing and want to suppress this warning.
root@ubuntu:~/jetson/Downloads# python3
Python 3.8.10 (default, Jul 29 2024, 17:02:10)
[GCC 9.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import torch
>>> torch.cuda.is_available()
True
>>>

```

그림 15. pytorch 설치 후 CUDA 확인하기

9. Ubuntu 20.04에서 torchvision 설치하기(수동 컴파일)

torchvision을 pip로 설치하면 aarch64(ARM64)에서 호환이 되지 않으므로 수동으로 컴파일로 설치해야 한다.

```
[torchvision]
sudo apt install -y libjpeg-dev zlib1g-dev
git clone https://github.com/pytorch/vision torchvision
cd torchvision
git checkout v0.16.2
python3 setup.py install --user
```

```
python3
import torchvision
print(torchvision.__version__)
```

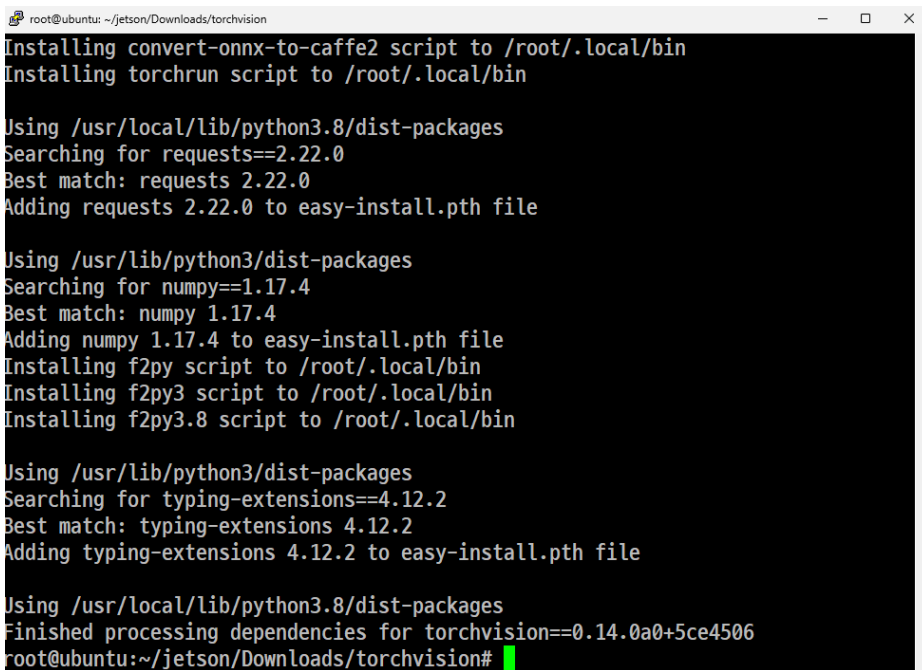


그림 16. torchvision 설치하기

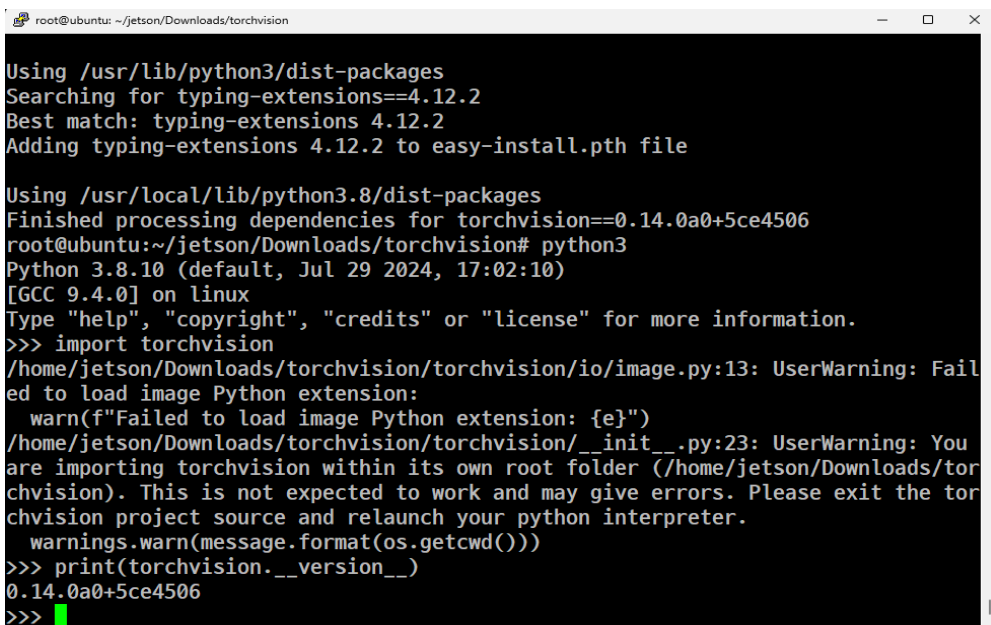


그림 17. torchvision 설치하기

10. Ubuntu 20.04에서 onnxruntime-gpu 설치하기

pip로 nvidia의 box에서 제공하는 onnxruntime으로 설치해야 한다.

```
[onnxruntime-gpu] (30분)
wget https://nvidia.box.com/shared/static/zostg6agm00fb6t5uisw51qi6kpcuwzd.whl -O
onnxruntime_gpu-1.17.0-cp38-cp38-linux_aarch64.whl
pip install onnxruntime_gpu-1.17.0-cp38-cp38-linux_aarch64.whl
```

```
Length: 52653729 (50M) [application/octet-stream]
Saving to: 'onnxruntime_gpu-1.17.0-cp38-cp38-linux_aarch64.whl'

onnxruntime_gpu-1.1 100%[=====>] 50.21M 16.1MB/s in 3.3s

2024-09-13 18:27:42 (15.3 MB/s) - 'onnxruntime_gpu-1.17.0-cp38-cp38-linux_aarch64.whl' saved [52653729/52653729]

root@ubuntu:~/jetson/Downloads# pip install onnxruntime_gpu-1.17.0-cp38-cp38-linux_aarch64.whl
Processing ./onnxruntime_gpu-1.17.0-cp38-cp38-linux_aarch64.whl
Collecting coloredlogs (from onnxruntime-gpu==1.17.0)
  Downloading coloredlogs-15.0.1-py2.py3-none-any.whl.metadata (12 kB)
Collecting flatbuffers (from onnxruntime-gpu==1.17.0)
  Downloading flatbuffers-24.3.25-py2.py3-none-any.whl.metadata (850 bytes)
Collecting numpy>=1.24.4 (from onnxruntime-gpu==1.17.0)
  Downloading numpy-1.24.4-cp38-cp38-manylinux_2_17_aarch64.manylinux2014_aarch64.whl.metadata (5.6 kB)
Requirement already satisfied: packaging in /usr/lib/python3/dist-packages (from onnxruntime-gpu==1.17.0) (20.3)
Requirement already satisfied: protobuf in /usr/lib/python3/dist-packages (from onnxruntime-gpu==1.17.0) (3.6.1)
Collecting sympy (from onnxruntime-gpu==1.17.0)
  Downloading sympy-1.13.2-py3-none-any.whl.metadata (12 kB)
```

그림 18. onnxruntime_gpu 설치하기

```
Downloading mpmath-1.3.0-py3-none-any.whl (536 kB)
536.2/536.2 kB 195.6 kB/s eta 0:00:00
Installing collected packages: mpmath, flatbuffers, sympy, numpy, humanfriendly, coloredlogs, onnxruntime-gpu
  Attempting uninstall: numpy
    Found existing installation: numpy 1.17.4
    Uninstalling numpy-1.17.4:
      Successfully uninstalled numpy-1.17.4
Successfully installed coloredlogs-15.0.1 flatbuffers-24.3.25 humanfriendly-10.0 mpmath-1.3.0 numpy-1.24.4 onnxruntime-gpu-1.17.0 sympy-1.13.2
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package manager, possibly rendering your system unusable.It is recommended to use a virtual environment instead: https://pip.pypa.io/warnings/venv. Use the --root-user-action option if you know what you are doing and want to suppress this warning.
root@ubuntu:~/jetson/Downloads# python3
Python 3.8.10 (default, Jul 29 2024, 17:02:10)
[GCC 9.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import torch
>>> torch.cuda.is_available()
True
>>> exit()
root@ubuntu:~/jetson/Downloads#
```

그림 19. onnxruntime_gpu 설치하기