# Jetson Nano Setup

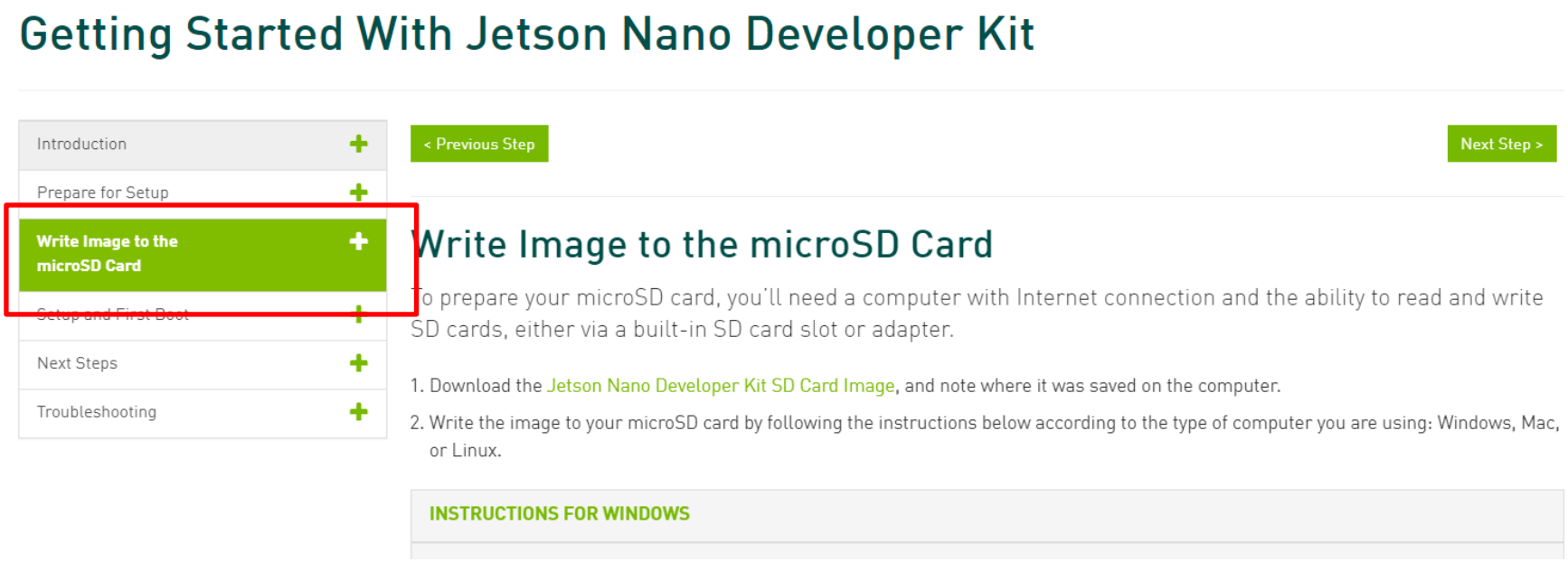
조 영 혁

노다시스템

# Jetson Nano Setup

## 1. Setting up Jetson with JetPack

https://developer.nvidia.com/embedded/learn/get-started-jetson-nano-devkit




Download : Jetpack 4.2.3

# Jetson Nano Setup

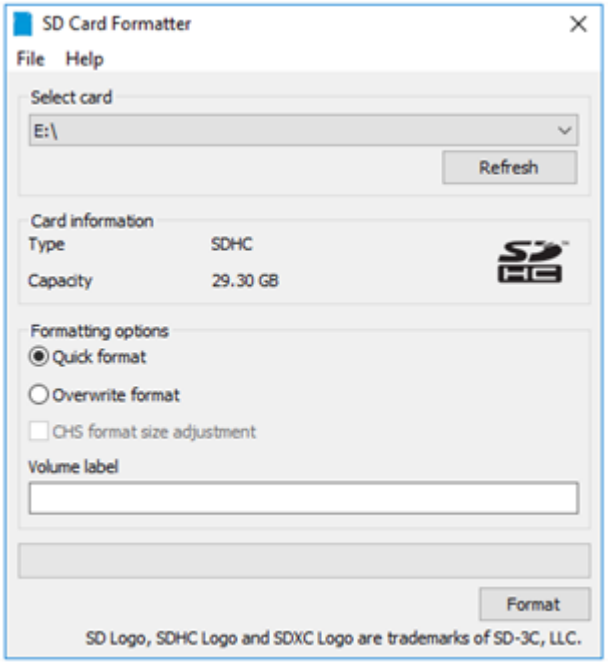
## Write Image to the microSD Card

To prepare your microSD card, you'll need a computer with Internet connection and the ability to read and write SD cards, either via a built-in SD card slot or adapter.

1. Download the Jetson Nano Developer Kit SD Card Image, and note where it was saved on the computer.
2. Write the image to your microSD card by following the instructions below according to the type of computer you are using: Windows, Mac, or Linux.

**INSTRUCTIONS FOR WINDOWS**

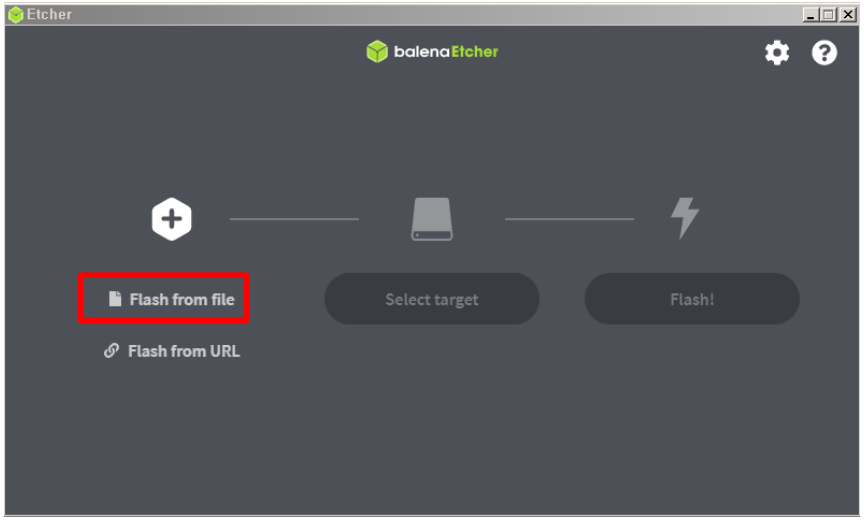
Format your microSD card using SD Memory Card Formatter from the SD Association.




1. Download, install, and launch SD Memory Card Formatter for Windows.
2. Select card drive
3. Select "Quick format"
4. Leave "Volume label" blank
5. Click "Format" to start formatting, and "Yes" on the warning dialog

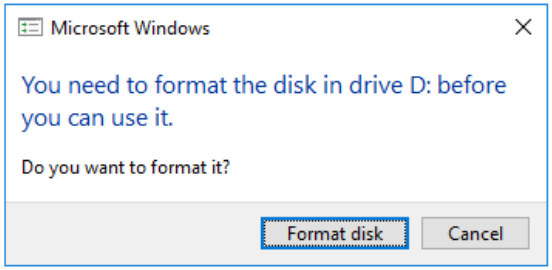
# Jetson Nano Setup

1. Download, install, and launch Etcher.



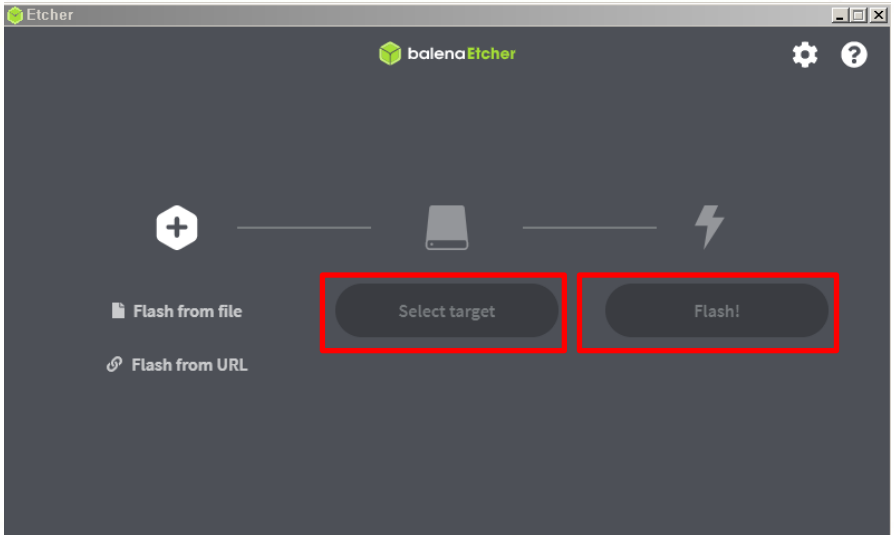

2. Click "**Flash from file**" and choose the zipped image file downloaded earlier.

3. Insert your microSD card if not already inserted.
   Click *Cancel* (per this explanation) if Windows prompts you with a dialog like this:



# Jetson Nano Setup

4. Click "Select target" and choose the correct device.

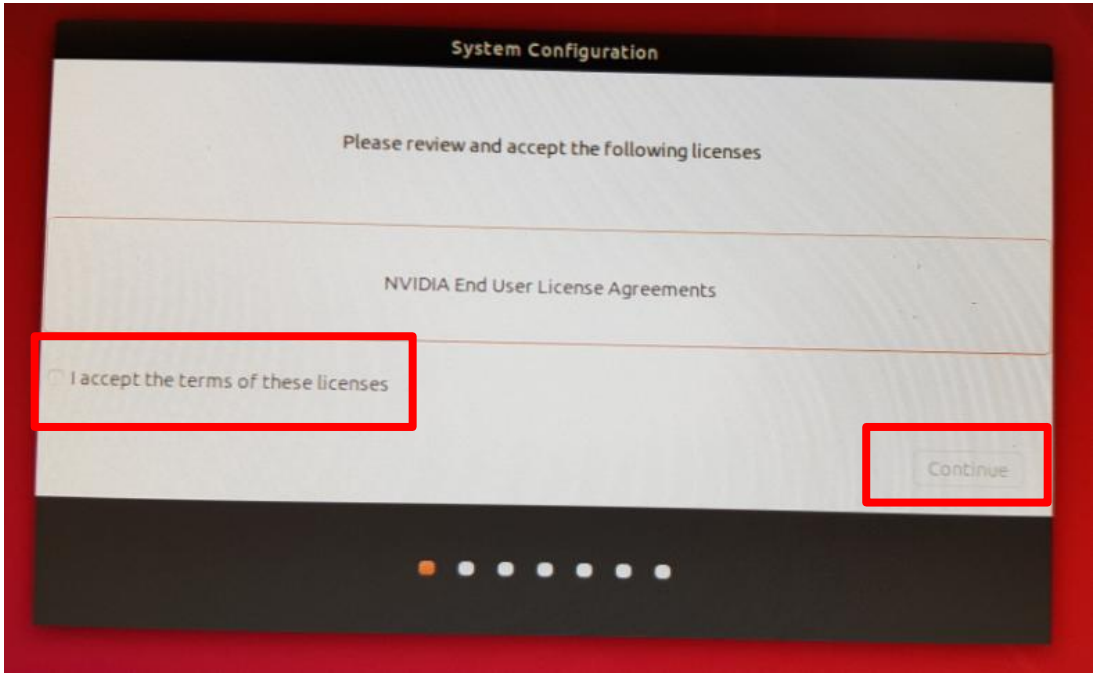



5. Click "Flash!" It will take Etcher about 10 minutes to write and validate the image if your microSD card is connected via USB3.

6. After Etcher finishes, Windows may let you know it doesn't know how to read the SD Card. Just click Cancel and remove the microSD card.

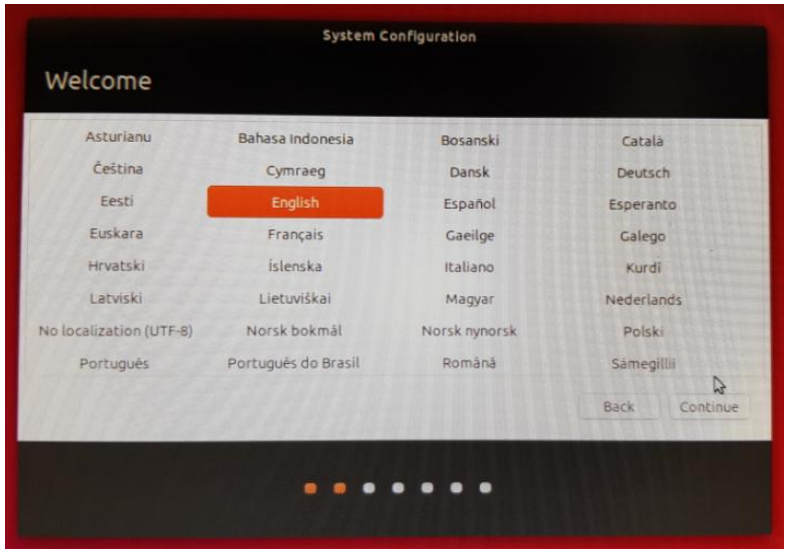
# Jetson Nano Setup

## 2. Building the Project from Source

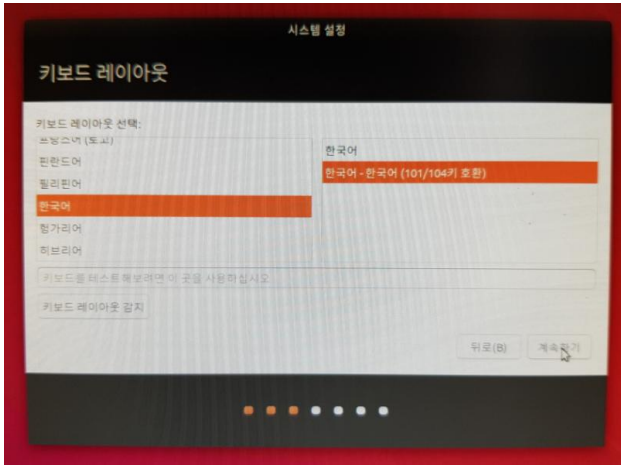


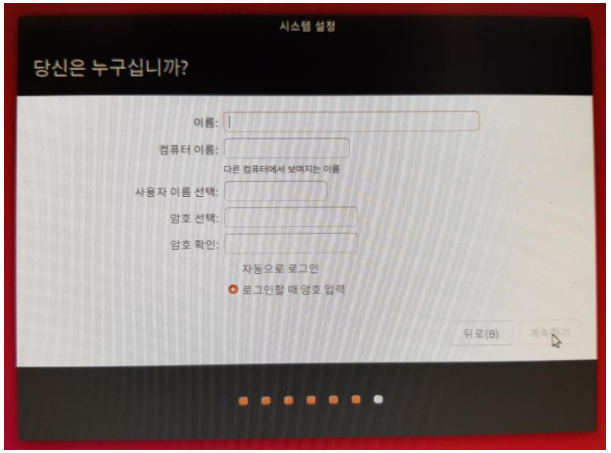
# Jetson Nano Setup

## 2. Building the Project from Source







# Jetson Nano Setup

## 2. Building the Project from Source

< Update >

To download the code

```
$ sudo apt-get update
$ sudo apt-get install git cmake
```

: /etc/atp/sources.list 패키지 정보에서 update
: git의 cmake 빌드 스크립트 install함

clone the jetson-inference project

```
$ git clone https://github.com/dusty-nv/jetson-inference
$ cd jetson-inference
$ git submodule update --init
```

: /etc/atp/sources.list 패키지 정보에서 update
: git의 cmake 빌드 스크립트 install함
: 하위모듈 update

Python Development Packages

```
$ sudo apt-get install libpython3-dev python3-numpy
```

- Numpy : numerical python의 약자
- Libpython3-dev : python dev 패키지 ( lib 컴파일을 위한 파일들 )

# Jetson Nano Setup

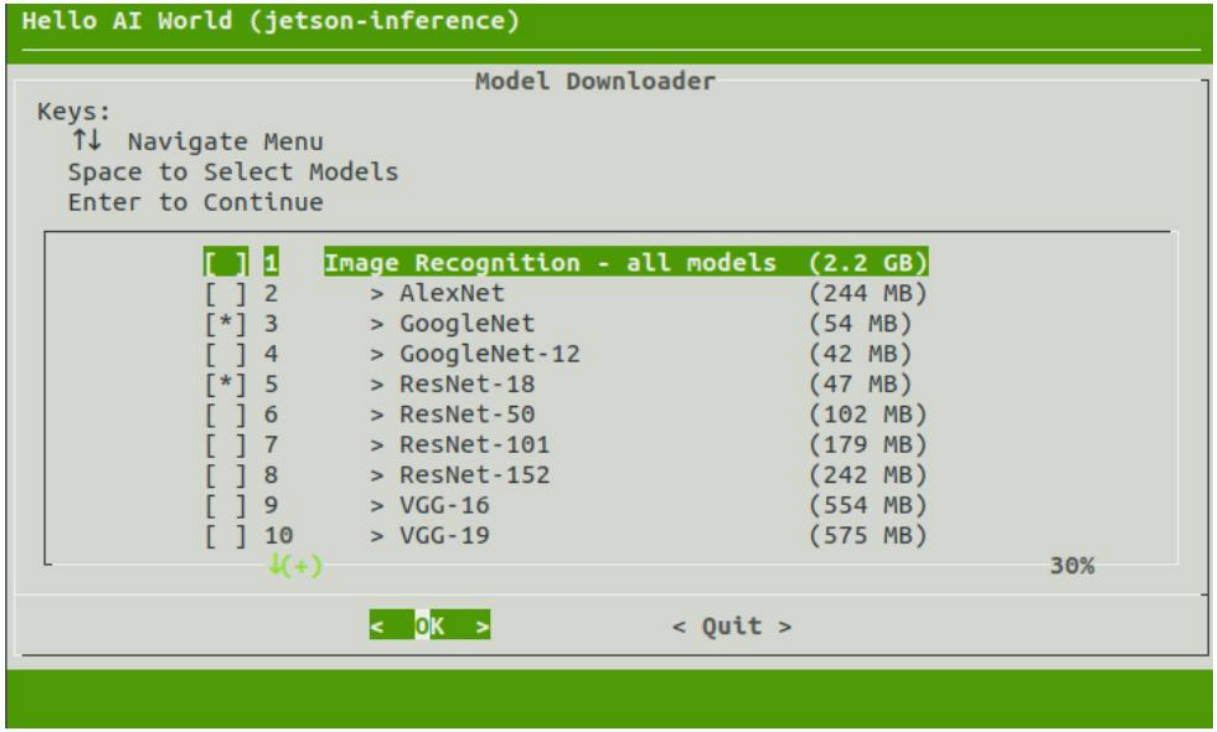
Configuring with CMake

```
$ cd jetson-inference
$ mkdir build
$ cd build
$ cmake ../
```

추가 다운로드

```
$ cd jetson-inference/tools
$ ./download-models.sh
```

: build script 실행 (cmakelists.txt)

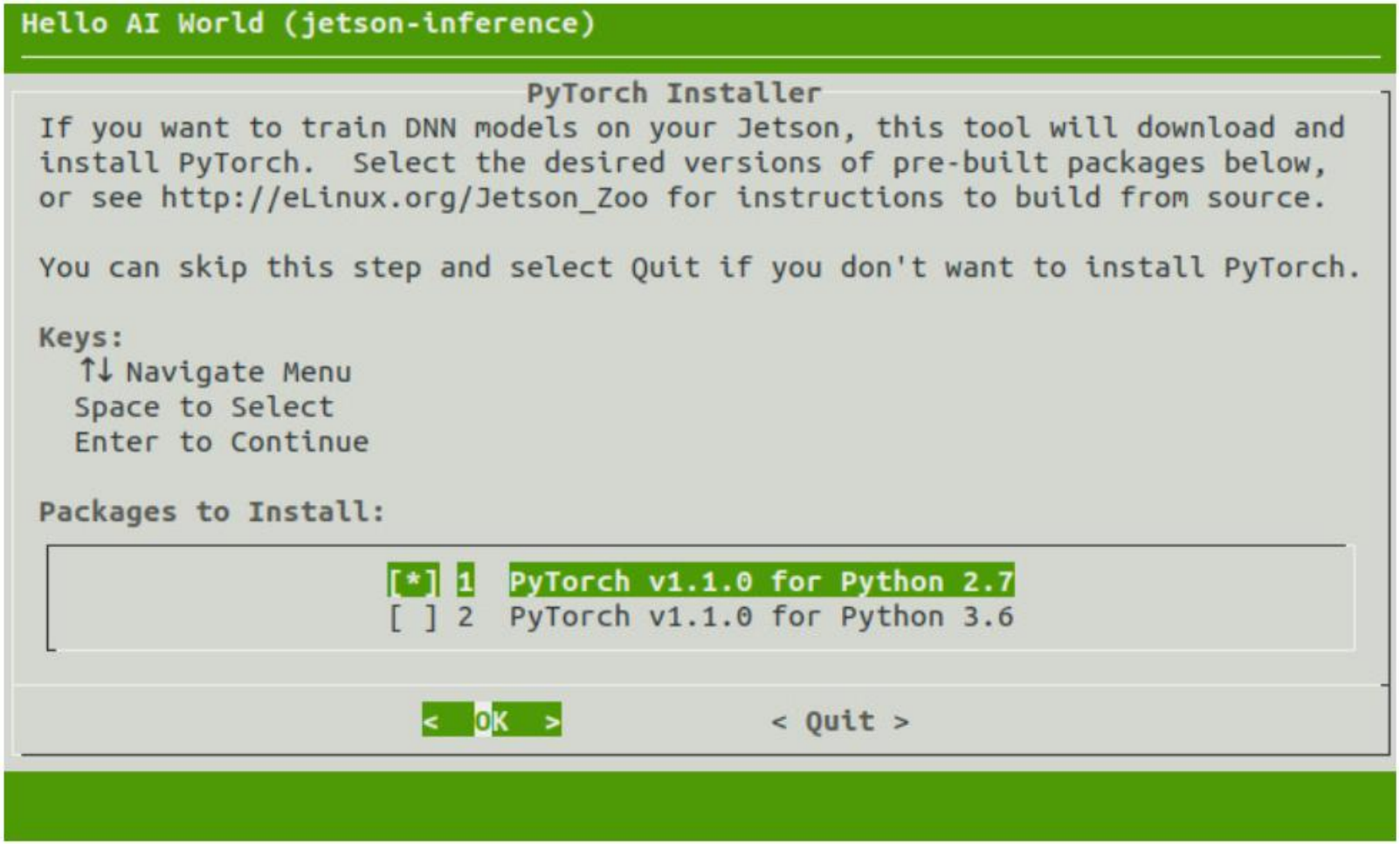


# Jetson Nano Setup

Installing PyTorch

추가 다운로드

```
$ cd jetson-inference/build
$ ./install-pytorch.sh
```



# Jetson Nano Setup

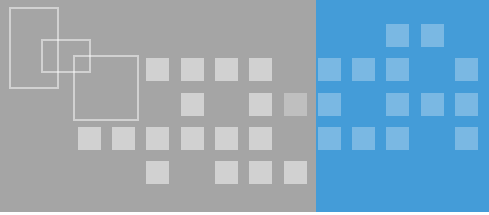
Compiling the Project

```
$ cd jetson-inference/build
$ make
$ sudo make install
$ sudo ldconfig
```

: 소스파일을 build

```
|-build
   \aarch64
      \bin              where the sample binaries are built to
         \networks      where the network models are stored
         \images        where the test images are stored
      \include          where the headers reside
      \lib              where the libraries are build to
```

# Jetson Nano Setup



Quick Reference

```
$ sudo apt-get update
$ sudo apt-get install git cmake libpython3-dev python3-numpy
$ git clone --recursive https://github.com/dusty-nv/jetson-inference
$ cd jetson-inference
$ mkdir build
$ cd build
$ cmake ../
$ make
$ sudo make install
$ sudo ldconfig
```

# Jetson Nano Setup

## 3. 카메라 테스트

/jetson-inference/build/aarch64/bin

aarch64 : 64bit arm architecture
aarch32 : 32bit arm architecture

- camera-capture
- camera-viewer

- camrea-viewer.py

- https://github.com/dusty-nv/jetson-utils

# 부록1 – Linux 명령어

| 명령어 | 내용 | 기타 |
|---|---|---|
| ls | 디렉터리 내용의 목록을 보여줌 | -l , -a |
| cp | 파일을 복사 한다. | |
| rm | 파일을 삭제한다 | |
| echo | 설정된 path를 보여준다. | $PATH |
| cd | 현재의 디렉터리를 이동한다. | |
| mkdir | 디렉터리 생성한다 | |
| rmdir | 디렉터리를 삭제한다 | |
| pwd | 현재 작업중인 디렉터리의 경로를 표시한다. | |
| cat | 파일 내용을 표시한다. | |
| sudo | 다른 사용자의 보안 권한으로 명령어를 수행한다. | |
| chmod | 파일의 접근 권한을 변경한다. | |
| tar | 여러 개의 파일을 하나의 파일로 묶는다.<br>-c (묶기) -x (풀기) -f(읽거나 기록파일정의) –v(묶거나풀공있는파일보기) –z(gzip으로 압축하여 백업) | |
| apt-get | Ubuntu를 포함한 데비안 계열에서 소프트웨어 패키지 설치 및 관리 | |

# 부록2 – ubunt

< apt-get 명령어 > : ubuntu 포함한 Debian 계열의 리눅스 패키지 관리 명령어
 - apt ( Advanced packaging tool kit )

sudo apt-get update : /etc/apt/sources.list 인덱스에서 업데이트를 함
sudo apt-get upgrade : 설치된 패키지를 모두 최신 패키지로 upgrade함
sudo apt-get install 패키지이름 : 패키지 설치
sudo apt-get remove 패키지이름 : 패키지 삭제
sudo apt-get source 패키지 이름

# 부록3 – git hub명령어

< git 명령어 > : 분산버전 관리 시스템

git clone https://xxxxx : libraray를 clone 함
git add 파일명 : 파일
git –version : git 버전 표시
git init : 현재 디렉터리에 git 저장소 생성
git submodule update : 하위모듈의 변경 사항을 적용함

cmake : build script ( CMakeLists.txt 에서 관리 )
make : 소스를 컴파일하여 실행 파일로 만들어 줌
make install : 앞에 것을 실행하여 설치를 함

# THANK YOU

*Suggestions Questions*