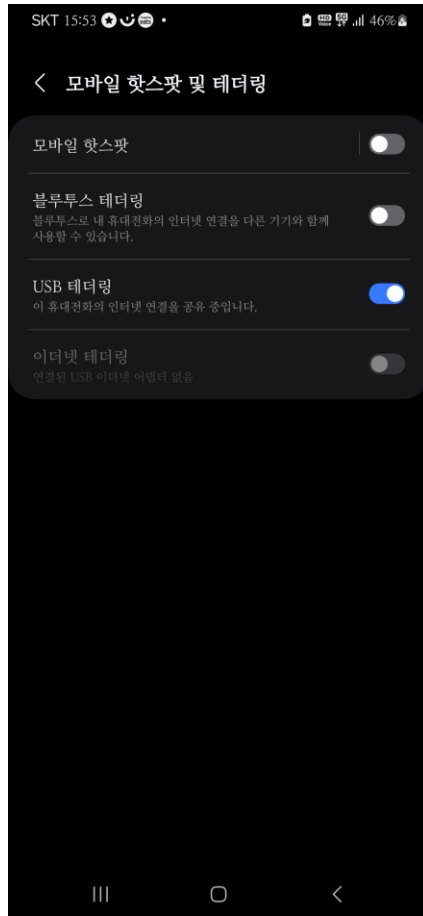


Tensorrt_demos 빌드 방 법 정리

저자 20201129 최준영

우선 AutoCar를 인터넷에 접속되도록 하는 것이 중요합니다.



USB 테더링을 통해 AutoCar의 USB 포트에 스마트폰의 usb 단자와 연결하여 테더링에 성공한 모습입니다.

ISP에 연결되었는지 여부를 확인

- File > New > Terminal 로 새 터미널 창의 띄웁니다.
- ping 8.8.8.8로 google dns의 공인 ip를 받아올 수 있는지 여부를 확인합니다.

빌드를 하기 위한 준비

- https://github.com/jkjung-avt/tensorrt_demos?tab=readme-ov-file#yolov4 에 접속하여 yolo v4의 빌드문서를 참조합니다.
- Which nvcc를 타이핑하여 cuda의 경로를 찾습니다.







```
> which nvcc
```

- /usr/local/cuda/bin/nvcc
- git clone https://github.com/jkjung-avt/tensorrt_demos.git
- cd \${HOME}/project/tensorrt_demos/yolo
- ./install_pycuda.sh(인터넷 접속 필요)
- \$ sudo pip3 install onnx==1.9.0(인터넷 접속 필요)

빌드를 해봅시다.

- `cd ${HOME}/project/tensorrt_demos/plugins`
- `make NVCC=/usr/local/cuda/bin/nvcc`
- 이때 NVCC는 make 파일 내부의 cuda의 경로를 지정하는 변수입니다. 해당 변수에 cuda의 경로를 명시하여 make에 인수로 넘겨줌으로써 cuda 경로를 잡지 못해 빌드가 실패하는 일을 막아줍니다.

/ tensorrt_demos-master / plugins /

Name	Last M
 gpu_cc.py	a j
 Makefile	9 d
 README.md	a j
 yolo_layer.cu	a j
 yolo_layer.h	a j
 yolo_layer.o	9 d

```
Makefile
1 CC=g++
2 LD=ld
3 CXXFLAGS=-Wall -std=c++11 -g -O
4
5 NVCC=/usr/local/cuda/bin/nvcc
6
7
```

모델을 최적화 합니다.

- `cd ${HOME}/project/tensorrt_demos/yolo`
- `python3 yolo_to_onnx.py -m yolov4-tiny-custom`
을 실행하여 모델을 onnx로 변환합니다.

`python3 onnx_to_tensorrt.py -m yolov4-tiny-custom`
을 실행하여 모델을 최적화 합니다.