

# 환경 구축 정의서

## 초기환경 세팅

1. gpu driver 설치 (<https://www.nvidia.com/Download/index.aspx>)

위 링크를 통해 사용중인 gpu에 따라 드라이버를 설치합니다 (nvidia 그래픽 카드에 해당 예))

### NVIDIA Driver Downloads

Select from the dropdown list below to identify the appropriate driver for your NVIDIA product. [Help](#)

Product Type:

Product Series:

Product:

Operating System:

Download Type:  ?

Language:

SEARCH

2. CUDA Toolkit 설치 (<https://developer.nvidia.com/cuda-toolkit-archive>)

앞서 설치한 드라이버의 버전에 맞춰서 CUDA Toolkit을 설치합니다. 드라이버와 CUDA 버전은 아래 테이블을 참조해서 매칭합니다.

Table 3. CUDA Toolkit and Corresponding Driver Versions

CUDA Toolkit	Toolkit Driver Version	
	Linux x86_64 Driver Version	Windows x86_64 Driver Version
CUDA 11.5 GA	>=495.29.05	>=496.04
CUDA 11.4 Update 2	>=470.57.02	>=471.41
CUDA 11.4 Update 1	>=470.57.02	>=471.41
CUDA 11.4.0 GA	>=470.42.01	>=471.11
CUDA 11.3.1 Update 1	>=465.19.01	>=465.89
CUDA 11.3.0 GA	>=465.19.01	>=465.89
CUDA 11.2.2 Update 2	>=460.32.03	>=461.33
CUDA 11.2.1 Update 1	>=460.32.03	>=461.09
CUDA 11.2.0 GA	>=460.27.03	>=460.82
CUDA 11.1.1 Update 1	>=455.32	>=456.81
CUDA 11.1 GA	>=455.23	>=456.38
CUDA 11.0.3 Update 1	>= 450.51.06	>= 451.82
CUDA 11.0.2 GA	>= 450.51.05	>= 451.48
CUDA 11.0.1 RC	>= 450.36.06	>= 451.22
CUDA 10.2.89	>= 440.33	>= 441.22
CUDA 10.1 (10.1.105 general release, and updates)	>= 418.39	>= 418.96
CUDA 10.0.130	>= 410.48	>= 411.31
CUDA 9.2 (9.2.148 Update 1)	>= 396.37	>= 398.26
CUDA 9.2 (9.2.88)	>= 396.26	>= 397.44
CUDA 9.1 (9.1.85)	>= 390.46	>= 391.29

3. anaconda 설치 (<https://docs.conda.io/en/latest/miniconda.html>)

가상환경 생성 및 라이브러리 설치를 위해 anaconda를 설치합니다. 해당 프로젝트에서는 python3.9 버전의 miniconda3를 사용했습니다.

## Latest Miniconda Installer Links

Latest - Conda 4.10.3 Python 3.9.5 released July 21, 2021 [🔗](#)

Platform	Name	SHA256 hash
Windows	Miniconda3 Windows 64-bit	b33797064593ab2229a0135dc69001bea05cb56a20c2f243b1231213642e260a
	Miniconda3 Windows 32-bit	24f438e57ff2ef1ce1e93050d4e9d13f5050955f759f448d84a018d3cd12d6b
MacOSX	Miniconda3 MacOSX 64-bit bash	786de9721f43e2c7d2803144c635f5f6e4823483536dc141ccd82dbb927cd508
	Miniconda3 MacOSX 64-bit pkg	8fa371ae97218c3c005cd5f04b1f40156d1506a9bd1d5c078f89d563f4d16816
Linux	Miniconda3 Linux 64-bit	1ea2f885b4dbc3098662845560bc64271eb17085387a70c2ba3f29fff6f8d52f
	Miniconda3 Linux-aarch64 64-bit	4879820a10718743f945d88ef142c3a4b30dfc8e448d1ca08e019586374b773f
	Miniconda3 Linux-ppc64le 64-bit	fa92ee4773611f58ed9333f977d32bb64769292f605d518732183be1f3321fa
	Miniconda3 Linux-s390x 64-bit	1faed9abecf4a4dd4e0d8891fc2cdaa3394c51e877af14ad6b9d4aadba4e90d8

## 사용 툴 및 라이브러리 (requirements.txt)

```
_libgcc_mutex=0.1=conda_forge
_openmp_mutex=4.5=1_gnu
abs1-py=0.14.1=pyhd8ed1ab_0
aiohttp=3.7.0=py37h8f50634_0
alsa-lib=1.2.3=h516909a_0
async-timeout=3.0.1=py_1000
attrs=21.2.0=pyhd8ed1ab_0
blas=1.0=mkl
blinker=1.4=py_1
brotlipy=0.7.0=py37h5e8e339_1001
bzip2=1.0.8=h7f98852_4
c-ares=1.17.2=h7f98852_0
ca-certificates=2021.10.8=ha878542_0
cachetools=4.2.4=pyhd8ed1ab_0
cairo=1.16.0=h6cf1ce9_1008
certifi=2021.10.8=py37h89c1867_0
cffi=1.14.6=py37hc58025e_0
chardet=3.0.4=py37he5f6b98_1008
click=8.0.3=py37h89c1867_0
colorama=0.4.4=pyh9f0ad1d_0
cryptography=3.4.8=py37h5d9358c_0
cudatoolkit=11.0.221=h6bb024c_0
cyclers=0.10.0=py_2
dataclasses=0.8=pyhc8e2a94_3
dbus=1.13.6=h48d8840_2
expat=2.4.1=h9c3ff4c_0
fastapi==0.70.0
ffmpeg=4.4.0=hca11adc_0
fontconfig=2.13.1=hba837de_1005
freelut=3.2.1=h9c3ff4c_2
freetype=2.10.4=h5ab3b9f_0
fsspec=2021.10.0=pyhd8ed1ab_0
future=0.18.2=py37h89c1867_3
gettext=0.19.8.1=h0b5b191_1005
glib=2.68.4=h9c3ff4c_0
glib-tools=2.68.4=h9c3ff4c_0
gmp=6.2.1=h58526e2_0
gnutls=3.6.13=h85f3911_1
google-auth=1.35.0=pyh6c4a22f_0
google-auth-oauthlib=0.4.6=pyhd8ed1ab_0
graphite2=1.3.13=h58526e2_1001
grpcio=1.33.2=py37haffed2e_2
gst-plugins-base=1.18.5=hf529b03_0
```

gststreamer=1.18.5=h76c114f\_0  
harfbuzz=3.0.0=h83ec7ef\_1  
hdf5=1.12.1=nompi\_h2750804\_101  
icu=68.1=h58526e2\_0  
idna=2.10=pyh9f0ad1d\_0  
importlib-metadata=4.8.1=py37h89c1867\_0  
intel-openmp=2021.3.0=h06a4308\_3350  
jasper=2.0.14=ha77e612\_2  
jbig=2.1=h7f98852\_2003  
jpeg=9d=h36c2ea0\_0  
kiwisolver=1.3.2=py37h2527ec5\_0  
krb5=1.19.2=hcc1bbae\_2  
lame=3.100=h7f98852\_1001  
lcms2=2.12=h3be6417\_0  
ld\_impl\_linux-64=2.35.1=h7274673\_9  
lerc=3.0=h9c3ff4c\_0  
libblas=3.9.0=11\_linux64\_mkl  
libcbblas=3.9.0=11\_linux64\_mkl  
libclang=11.1.0=default\_ha53f305\_1  
libcurl=7.79.1=h2574ce0\_1  
libdeflate=1.8=h7f98852\_0  
libedit=3.1.20191231=he28a2e2\_2  
libev=4.33=h516909a\_1  
libevent=2.1.10=h9b69904\_4  
libffi=3.3=he6710b0\_2  
libgcc-ng=11.2.0=h1d223b6\_11  
libgfortran-ng=11.2.0=h69a702a\_11  
libgfortran5=11.2.0=h5c6108e\_11  
libglib=2.68.4=h3e27bee\_0  
libglu=9.0.0=he1b5a44\_1001  
libgomp=11.2.0=h1d223b6\_11  
libiconv=1.16=h516909a\_0  
liblapack=3.9.0=11\_linux64\_mkl  
liblapacke=3.9.0=11\_linux64\_mkl  
libllvm11=11.1.0=hf817b99\_2  
libnghttp2=1.43.0=h812cca2\_1  
libogg=1.3.4=h7f98852\_1  
libopencv=4.5.3=py37hbfc4018\_5  
libopus=1.3.1=h7f98852\_1  
libpng=1.6.37=hbc83047\_0  
libpq=13.3=hd57d9b9\_1  
libprotobuf=3.18.1=h780b84a\_0  
libssh2=1.10.0=ha56f1ee\_2  
libstdcxx-ng=11.2.0=he4da1e4\_11  
libtiff=4.3.0=h6f004c6\_2  
libuuid=2.32.1=h7f98852\_1000  
libuv=1.40.0=h7b6447c\_0  
libvorbis=1.3.7=h9c3ff4c\_0  
libwebp-base=1.2.1=h7f98852\_0  
libxcb=1.13=h7f98852\_1003  
libxkbcommon=1.0.3=he3ba5ed\_0  
libxml2=2.9.12=h72842e0\_0  
libzlib=1.2.11=h36c2ea0\_1013  
lz4-c=1.9.3=h295c915\_1  
markdown=3.3.4=pyhd8ed1ab\_0  
matplotlib=3.4.3=py37h89c1867\_1  
matplotlib-base=3.4.3=py37h1058ff1\_1  
mkl=2021.3.0=h06a4308\_520

mk1-service=2.4.0=py37h7f8727e\_0  
mk1\_fft=1.3.0=py37h42c9631\_2  
mk1\_random=1.2.2=py37h51133e4\_0  
multidict=5.2.0=py37h5e8e339\_0  
mysql-common=8.0.25=ha770c72\_3  
mysql-libs=8.0.25=hfa10184\_3  
ncurses=6.2=he6710b0\_1  
nettle=3.6=he412f7d\_0  
ninja=1.10.2=hff7bd54\_1  
nspr=4.30=h9c3ff4c\_0  
nss=3.69=hb5efdd6\_1  
numpy=1.21.2=py37h20f2e39\_0  
numpy-base=1.21.2=py37h79a1101\_0  
oauthlib=3.1.1=pyhd8ed1ab\_0  
olefile=0.46=pyhd3eb1b0\_0  
opencv=4.5.3=py37h89c1867\_5  
openh264=2.1.1=h780b84a\_0  
openjpeg=2.4.0=h3ad879b\_0  
openssl=1.1.1l=h7f98852\_0  
packaging=21.0=pyhd8ed1ab\_0  
pcre=8.45=h9c3ff4c\_0  
pillow=8.3.1=py37h2c7a002\_0  
pip=21.2.2=py37h06a4308\_0  
pixman=0.40.0=h36c2ea0\_0  
protobuf=3.18.1=py37hcd2ae1e\_0  
pthread-stubs=0.4=h36c2ea0\_1001  
py-opencv=4.5.3=py37h6531663\_5  
pyasn1=0.4.8=py\_0  
pyasn1-modules=0.2.7=py\_0  
pyparser=2.20=pyh9f0ad1d\_2  
pydeprecate=0.3.1=pyhd8ed1ab\_0  
pyjwt=2.2.0=pyhd8ed1ab\_0  
pyopenssl=21.0.0=pyhd8ed1ab\_0  
pyparsing=2.4.7=pyh9f0ad1d\_0  
pyqt=5.12.3=py37h89c1867\_7  
pyqt-impl=5.12.3=py37he336c9b\_7  
pyqt5-sip=4.19.18=py37hcd2ae1e\_7  
pyqtchart=5.12=py37he336c9b\_7  
pyqtwebengine=5.12.1=py37he336c9b\_7  
pysocks=1.7.1=py37h89c1867\_3  
python=3.7.11=h12debd9\_0  
python-dateutil=2.8.2=pyhd8ed1ab\_0  
python\_abi=3.7=2\_cp37m  
pytorch=1.7.1=py3.7\_cuda11.0.221\_cudnn8.0.5\_0  
pytorch-lightning=1.4.9=pyhd8ed1ab\_0  
pyu2f=0.1.5=pyhd8ed1ab\_0  
pyyaml=5.4.1=py37h5e8e339\_1  
qt=5.12.9=hda022c4\_4  
readline=8.1=h27cfd23\_0  
requests=2.25.1=pyhd3deb0d\_0  
requests-oauthlib=1.3.0=pyh9f0ad1d\_0  
rsa=4.7.2=pyh44b312d\_0  
setuptools=58.0.4=py37h06a4308\_0  
six=1.16.0=pyhd3eb1b0\_0  
sqlite=3.36.0=hc218d9a\_0  
tensorboard=2.6.0=pyhd8ed1ab\_1  
tensorboard-data-server=0.6.0=py37hf1a17b8\_0  
tensorboard-plugin-wit=1.8.0=pyh44b312d\_0

```
tk=8.6.11=h1ccaba5_0
torchaudio=0.7.2=py37
torchmetrics=0.5.1=pyhd8ed1ab_0
torchvision=0.2.1=py37_0
tornado=6.1=py37h5e8e339_1
tqdm=4.62.3=pyhd8ed1ab_0
typing_extensions=3.10.0.2=pyh06a4308_0
ujson=4.0.1=py37he6710b0_0
urllib3=1.26.7=pyhd8ed1ab_0
werkzeug=2.0.1=pyhd8ed1ab_0
wheel=0.37.0=pyhd3eb1b0_1
x264=1!161.3030=h7f98852_1
xorg-fixesproto=5.0=h7f98852_1002
xorg-inputproto=2.3.2=h7f98852_1002
xorg-kbproto=1.0.7=h7f98852_1002
xorg-libice=1.0.10=h7f98852_0
xorg-libsm=1.2.3=hd9c2040_1000
xorg-libx11=1.7.2=h7f98852_0
xorg-libxau=1.0.9=h7f98852_0
xorg-libxdmcp=1.1.3=h7f98852_0
xorg-libxext=1.3.4=h7f98852_1
xorg-libxfixed=5.0.3=h7f98852_1004
xorg-libxi=1.7.10=h7f98852_0
xorg-libxrender=0.9.10=h7f98852_1003
xorg-renderproto=0.11.1=h7f98852_1002
xorg-xextproto=7.3.0=h7f98852_1002
xorg-xproto=7.0.31=h7f98852_1007
xz=5.2.5=h7b6447c_0
yaml=0.2.5=h516909a_0
yarl=1.6.0=py37h8f50634_0
zipp=3.6.0=pyhd8ed1ab_0
zlib=1.2.11=h36c2ea0_1013
zstd=1.5.0=ha95c52a_0
```

## 가상환경 생성

### 1. 가상환경 만들기

```
$ conda create env python=3.7
```

### 2. 가상환경 활성화

```
$ conda activate env
```

### 3. requirements.txt를 사용하여 패키지 설치

```
$ pip install -r requirements.txt
```

※ 해당 프로젝트에서는 cuda11.0.22 를 사용했습니다. 만약 그래픽카드에 따라 다른 버전의 cuda 를 사용할 경우 아래와 같은 명령어를 사용하여 pytorch 라이브러리에 사용될 cuda 버전을 맞춰줍니다.

예) CUDA 10.2 버전 사용시

```
$ conda install pytorch torchvision torchaudio cudatoolkit=10.2 -c pytorch
```

# 디렉토리 구조

## LaneDetection

- └─ deployment [배포용 디렉토리]
- └─ pre\_productVersion [개발용 디렉토리]
- └─ common [model, src에 공통으로 쓰이는 모듈]
- └─ model [AI모델 디렉토리]
  - | └─ data [학습용 데이터]
    - | | └─ testset
    - | | | └─ image
      - | | | | └─ 0
      - | | | | └─ 1
      - | | | | └─ 2
      - | | | | └─ 3 ...
    - | | | └─ json
      - | | | | └─ 0
      - | | | | └─ 1
      - | | | | └─ 2
      - | | | | └─ 3 ...
- └─ saved\_model [학습된 모델 저장 디렉토리]
- └─ [preprocessing.py](#) [학습 데이터 생성 파일]
- └─ src [웹어플리케이션 구현 모듈 디렉토리]
  - └─ common [웹어플리케이션 구현 파일 디렉토리]
  - └─ main.py [웹어플리케이션 실행 모듈]
  - └─ inferenceModule.py [모델의 인퍼런스 모듈 정의]

## 데이터 셋 준비하는 방법

1. 다음과 같이 image와 json 폴더를 만듭니다.



2. Image와 json 각각에 동일한 이름으로 하위 폴더를 필요한 만큼 만듭니다.



0



1



2



4



5



6



8



9



10

3. 각 하위 폴더에는 다음과 같이 동일한 이름으로 image와 json을 하위 폴더에 넣습니다.



10\_12\_007953\_  
0224.json



10\_12\_007953\_  
0261.json



10\_12\_007954\_  
0203.json



10\_12\_007954\_  
0249.json



10\_12\_007954\_  
0255.json



10\_12\_007954\_  
0270.json



10\_12\_007953\_  
0224.jpg



10\_12\_007953\_  
0261.jpg



10\_12\_007954\_  
0203.jpg



10\_12\_007954\_  
0249.jpg



10\_12\_007954\_  
0255.jpg



10\_12\_007954\_  
0270.jpg

4. 준비된 이미지의 하위폴더 경로를 data 폴더에 있는 train.txt, test.txt, val.txt 각각에 용도에 맞게 입력해 줍니다. sample.txt를 보면 sample 폴더 아래에 image 폴더의 하위 폴더 경로가 입력됐음을 확인할 수 있습니다.

## train.py 실행 방법

1. 데이터가 준비되었으면, python train.py를 실행하여 학습을 진행할 수 있습니다.



#gpu가 1개인 경우

```
trainer = pl.Trainer(gpus=0, distributed_backend="ddp", callbacks=[EarlyStopping(monitor='val_loss', patience=10)])
```

#gpu가 4개인 경우

```
trainer = pl.Trainer(gpus=[0,1,2,3], distributed_backend="ddp", callbacks=[EarlyStopping(monitor='val_loss', patience=10)])
```

2. 학습이 진행되면 진행상태를 나타내는 프로그레스 바와 현재의 epoch값이 표시됩니다.

```
| Name | Type | Params
-----|-----|-----
0 | fcn | MobileNetV2 | 2.2 M
-----|-----|-----
2.2 M | Trainable params
0 | Non-trainable params
2.2 M | Total params
8.921 | Total estimated model params size (MB)
dataset size : 4
/home/sang/anaconda3/envs/3.7/lib/python3.7/site-packages/pytorch_lightning/utilities/distributed.py:68: UserWarning: The dataloader, val dataloader 0, does not have many workers which may be a bottleneck. Consider increasing the value of the 'num_workers' argument (try 16 which is the number of cpus on this machine) in the 'DataLoader' init to improve performance.
warnings.warn(*args, **kwargs)
Validation sanity check: 50% | 1/2 [00:00<00:00, 2.88it/s]
dataset size : 4
/home/sang/anaconda3/envs/3.7/lib/python3.7/site-packages/pytorch_lightning/utilities/distributed.py:68: UserWarning: The dataloader, train dataloader, does not have many workers which may be a bottleneck. Consider increasing the value of the 'num_workers' argument (try 16 which is the number of cpus on this machine) in the 'DataLoader' init to improve performance.
warnings.warn(*args, **kwargs)
Epoch 14: 100% | 8/8 [00:00<00:00, 31.97it/s, loss=1.32, v_num=8]
(3.7) sangsang-H470-HD3:~/NIA_AI_Data_Project/trafficlights |
```

## test.py 실행 방법

1. 학습 때와 마찬가지로 sample 폴더와 같은 폴더 구조로 test 폴더를 만들고 학습시에 저장되었던 가중치 파일의 경로를 python test.py 코드의 4번째 줄의 checkpoint\_path의 경로에 입력해준 후, 테스트를 진행하면 됩니다.

```
#예시
model =
NIA_SEGNet_module.load_from_checkpoint(checkpoint_path="lightning_logs/version_9/checkpoints/epoch=32-step=32.ckpt")
```

2. test.py를 실행하면 앞서 샘플 데이터를 테스트했을 때처럼 진행이 됩니다.

```
python test.py
```

## 버전관리

git을 통해 버전관리를 합니다.

<https://github.com/YOUR-USERNAME/YOUR-REPOSITORY>

1. 최초 다운로드 시 ex) 레포지토리 명 = dishes

github 원격 저장소로부터 다운로드 받습니다.

(\$ git clone <https://github.com/YOUR-USERNAME/YOUR-REPOSITORY>)

1. 클론하고자 하는 폴더를 컴퓨터내에 생성합니다.

2. gitbash를 실행하여 생성한 폴더의 경로로 이동합니다.

```
MINGW64:/d/youngjae/a.coding/f.git/dishes
Youngjae Choi@DESKTOP-9V2T54A MINGW64 /d/youngjae/a.coding/f.git/dishes (master)
$
```

```
MINGW64:/d/youngjae/a.coding/f.git/dishes-cloned
Youngjae Choi@DESKTOP-9V2T54A MINGW64 /d/youngjae/a.coding/f.git/dishes (master)
$ cd ..

Youngjae Choi@DESKTOP-9V2T54A MINGW64 /d/youngjae/a.coding/f.git
$ ls
dishes/  dishes-cloned/

Youngjae Choi@DESKTOP-9V2T54A MINGW64 /d/youngjae/a.coding/f.git
$ cd dishes-cloned/

Youngjae Choi@DESKTOP-9V2T54A MINGW64 /d/youngjae/a.coding/f.git/dishes-cloned
$
```

c. 다운받고자 하는 github 원격저장소의 주소를 입력해줍니다.

```
MINGW64:/d/youngjae/a.coding/f.git/dishes-cloned
Youngjae Choi@DESKTOP-9V2T54A MINGW64 /d/youngjae/a.coding/f.git/dishes (master)
$ cd ..

Youngjae Choi@DESKTOP-9V2T54A MINGW64 /d/youngjae/a.coding/f.git
$ ls
dishes/  dishes-cloned/

Youngjae Choi@DESKTOP-9V2T54A MINGW64 /d/youngjae/a.coding/f.git
$ cd dishes-cloned/

Youngjae Choi@DESKTOP-9V2T54A MINGW64 /d/youngjae/a.coding/f.git/dishes-cloned
$ git clone https://github.com/nyjchoi/dishes.git
```

**\$ git clone <https://github.com/nyjchoi/dishes.git> .**

뒤에 한칸 띄우고 .을 찍어줍니다. 이렇게 하면 자동으로 폴더를 만들지 않고 바로 클론 하게됩니다.

2. 변경사항 다운로드 시 ex) 레포지토리 명 = dishes

1. gitbash의 작업폴더를 변경합니다.

```
Youngjae Choi@DESKTOP-9V2T54A MINGW64 /d/youngjae/a.coding/f.git/dishes-cloned (
master)
$ cd ../dishes

Youngjae Choi@DESKTOP-9V2T54A MINGW64 /d/youngjae/a.coding/f.git/dishes (master)
$ |
```

b. git pull origin master를 입력합니다.

```
Youngjae Choi@DESKTOP-9V2T54A MINGW64 /d/youngjae/a.coding/f.git/dishes (master)
$ git pull origin master
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 3 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 472 bytes | 0 bytes/s, done.
From https://github.com/nyjchoi/dishes
* branch      master      -> FETCH_HEAD
   b83dcb6..e185e3f master  -> origin/master
Updating b83dcb6..e185e3f
Fast-forward
 README.md | 5 ++++-
 1 file changed, 4 insertions(+), 1 deletion(-)
Youngjae Choi@DESKTOP-9V2T54A MINGW64 /d/youngjae/a.coding/f.git/dishes (master)
$
```