



# **Tutorial**

- □ 본 튜토리얼은 리눅스 환경에서 Docker를 사용한 방식과 소스코드를 사용 한 방식 두 가지로 설명되어 있습니다.
- Docker download : <a href="https://www.docker.com/">https://www.docker.com/</a>
  (Docker image는 Al-Hub에서 다운받을 수 있습니다.)
- Source code download : <a href="https://github.com/kotechnia/noise-reduction">https://github.com/kotechnia/noise-reduction</a>
- □ 시스템 사양은 다음과 같습니다.
- Ubuntu 20.04
- Python 3.8.10
- Torch 1.9.0+cu111
- CUDA 11.1
- cuDnn 8.2.0
- Docker 20.10.7

#### □ Docker

1. 다운받은 Docker image(noise-reduction.tar)를 load 합니다.

(Docker image는 AI-Hub 사이트에서 다운로드 가능합니다.)

\$ docker load -i noise-reduction

Loaded image: noise-reduction

2. Container를 실행합니다.

- \$ docker run -d -it --runtime=nvidia -v /data/:/workspace/share/data --name noise noise-reduction
- -v /data/:/workspace/share/data : 다운받은 "/data/"폴더를 docker container 안의 "/share/data" 폴더와 공유합니다. (학습에 필요한 data 는 AI-Hub 사이트에서 다운로드 가능합니다. )
- 3. 데이터 세트를 훈련(Train), 검증(Validation), 실험(Test) 세트로 8:1:1 분리합니다.
  - \$ docker exec -t noise bash -ic "python dataset\_maker.py --wav\_files\_dir=share/data --csv\_save\_path=share/dataset.csv"

[share/data] 폴더 안의 wav파일에 대하여 훈련(Train), 검증(validation), 시험 (test) 세트로 8:1:1 분류한 목록을 [share/dataset.csv]에 저장합니다. (wav 파일은 clean(sd)과 noisy(sn)두개 파일이 하나의 셋으로 이루어져 있어야 함.)

※ 생성된 share/dataset.csv파일은 다음과 같은 구조를 갖습니다.

sd_file_path	sn_file_path	train_val_test
share/data/sd_file1.wav	share/data/sn_file1.wav	TR
share/data/sd_file2.wav	share/data/sn_file2.wav	VA
share/data/sd_filen.wav	share/data/sn_filen.wav	TE

--wav\_files\_dir : 사용될 wav 파일 폴더 경로

--csv\_save\_path : csv 파일 경로

#### 4. AI 모델 훈련

\$ docker exec -t noise bash -ic "python train.py --model=models/tscn

- --csv file=share/dataset.csv --cme epochs=400
- --finetune\_epochs=40 --csr\_epochs=400 --batch\_size=8
- --multi\_gpu=True --preproc\_path=share/preproc"

[share/dataset.csv]의 목록으로 multi\_gpu로 사용하여 batch\_size를 8로 cme, finetune, csr 훈련을 각각 400, 40, 400회를 진행하며 진행단계에서 사용되는 전처리데이터를 [share/preproc]에 저장하고, 훈련이 끝나면 [models/tscn]에 모델을 저장합니다.

--model : 모델 저장경로

--csv file : 훈련에 사용될 데이터 세트 목록

--cme\_epochs : cme 훈련 횟수

--inetune epochs : finetune 훈련 횟수

--csr\_epochs : csr 훈련 횟수

--batch\_size : 훈련 batch 크기

--multi\_gpu : 다중 gpu 사용 여부 [ True, False ]

--preproc path : 전처리 저장경로

## 5. 노이즈 감소

단일 파일에 대해 노이즈 감소입니다.

\$ docker exec -t noise bash -ic "python denoise.py

- --model=models/tscn --noisy=sn1.wav --denoise=de1.wav
- --clean=sd1.wav"

[models/tscn]을 사용하여 [sn1.wav] 해당 음성을 노이즈 감소하여 [de1.wav]로 생성한 후 [sd1.wav]와 결과를 비교한다.

--model: AI모델 경로

--noisy : 노이즈 파일 경로

--denoise : 결과 파일 저장 경로

--clean(optional) : estoi 계산을 위한 깨끗한 음성 파일 경로

데이터 세트 목록 대해 노이즈 감소입니다.

\$ docker exec -t noise bash -ic "python denoise.py

--model=models/tscn --csv\_file=share/dataset.csv

--output\_dir=share/denoise"

[models/tscn] 모델을 사용하여 [share/dataset.csv]에서 시험(Test)으로 분류 된 목록으로 노이즈 감소를 하여 [share/denoise]에 저장한다.

--csv\_file : csv 파일 경로

--output\_dir : 결과 파일 저장 경로

- ☐ Source code
- 1. Source code를 다운로드 합니다.

github.com의 사이트에서 .zip 파일 또는 git을 이용하여 다운로드합니다.

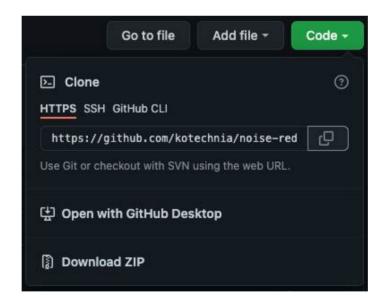
- GPU를 사용하는 시스템일 경우 CUDA와 cuDnn은 설치가 되어 있어야 합니다.
자세한 설치 방법은 <a href="https://docs.nvidia.com/cuda/cuda-installation-guide-linux/index.html">https://docs.nvidia.com/cuda/cuda-installation-guide-linux/index.html</a>
를 참조하세요.

qit을 이용한 다운로드

\$ git clone https://github.com/kotechnia/noise-reduction.git

## .zip 다운로드

https://github.com/kotechnia/noise-reduction.git 로 접속 후 "Download ZIP"을 클릭하여 다운로드 합니다.



다운받은 소스코드의 폴더 구조는 다음과 같습니다.

noise-reduction
— dataloader
— models
— tscn
— share (data share for docker dicrectory)
— data
— dataset\_nia2021\_noise\_reduction.csv
— tscn
— utils
— dataset\_maker.py
— denoise.py
— requirements.txt
— train.py

[dataloader]: 학습 데이터를 읽기 위한 모듈

[models] : 학습된 모델 저장 폴더

[share]: 데이터 공유 폴더(docker 사용시)

[dataset\_nia2021\_noise\_reduction.csv] : 미리 학습된 데이터에 대한 리스트

(Train, Validation, Test set 경로)

[tscn] : tscn 학습 모델 모듈

[utils] : 기타 유틸

## 2. 라이브러리 설치

사용된 라이브러리들의 버전입니다.

```
torch==1.9.0+cu111 -f https://download.pytorch.org/whl/torch_stable.html
tqdm==4.62.3
librosa==0.8.1
numpy==1.20.3
torchaudio==0.9.0
scipy==1.7.1
```

7

pystoi==0.3.3

matplotlib==3.4.3

pandas = = 1.3.3

openpyxl==3.0.9

다음의 명령어로 필요 라이브러리를 한번에 설치 가능 합니다.

([noise-reduction] 폴더의 requierment.txt 파일 참조)

\$ pip install -r requirement.txt

## 3. 데이터 세트 목록 생성

\$ python dataset\_maker.py --wav\_files\_dir=share/data
--csv\_save\_path=share/dataset.csv

## 4. AI 모델 훈련

- \$ python train.py --model=models/tscn --csv\_file=share/dataset.csv
- --cme epochs=400 —finetune epochs=40 --csr epochs=400
  - --batch\_size=8 --multi\_gpu=True --preproc\_path=share/preproc

## 5. 노이즈 감소

- \$ python denoise.py --model=models/tscn --noisy=share/sn1.wav
  - --denoise=share/de1.wav --clean=share/sd1.wav