

Spine Labeling Project User Manual

2023-2 Creative Integrated Design 1 Project Team A

Created by:

2020-19291 전민주

2020-19851 오이석

1. Library 준비

해당 프로젝트를 실행하기 위해서는 우리가 개발한 파일들을 담은 Repo와 더불어, Python(≥ 3.9), PyTorch, GPU 사용환경(CUDA, CUDNN)이 구현되어야 한다. 그 위에 위의 링크에서 VerSe 데이터셋과 nnUNet을 설치하여야 한다.

Custom Files

다음 Repo에 접속해서 다음 파일들을 받으면 된다.

Spine Labeling Project

Dataset101: Processed & Trained Result of VerSe2020 Dataset. Trained CT image (**.nii.gz**) excluded due to large memory.

results: Images (sagittal plane, etc) of true/inferenced data.

README.md: This file.

compute_dice.py: Computing dice coefficient for resulting CT images.

data_utilities.py: Data utilities for taking planed images.

datagen_verse.py: Preprocessing of VerSe dataset for nnUNet use. Includes various preprocessing options.

example_image.py: Draw plane image from CT images.

example_image_centroid.py: Inference centroid of each spine, using mean.

example_image_demo.py: Centroid inference demo.

example_image_median.py: Inference centroid of each spine, using median, a robust statistic.

view_image_w_label.py: Simultaneously compute dice coefficient and draw plane images.

All codes except **data_utilities.py** were made by ourselves. **data_utilities.py** is based on [VerSe Repository](#)'s file, which is modified for our use.

Dataset

VerSe Dataset은 그 용량이 매우 크기에 다음 GitHub Repository에서 받을 수 있다.

VerSe: Large Scale Vertebrae Segmentation Challenge

Deep Learning Library (nnUNet)

학습의 기반이 된 딥러닝 라이브러리는 nnUNet으로, 다음 GitHub Repository에서 받을 수 있다.

nnUNet

마지막으로 `nibabel` 패키지가 필요하다. 다음과 같이 설치할 수 있다.

```
pip install nibabel
```

2. Preprocessing

VerSe 데이터 셋을 활용하기 위해서는 가장 먼저 이를 전처리해주어야 한다. 우리의 코드 `datagen_verse.py` 를 통해 원하는 옵션으로 데이터를 전처리해주면 된다. 옵션에 따라 projection, rescale, reorientation 등을 다르게 할 수 있으니, 이 부분에서 원하는 스타일대로 전처리 옵션을 주면 된다. 데이터 전처리 후의 디렉토리의 형태는 다음과 같다.

```
nnUNet_raw/
├── Dataset101_VerSe2020
│   ├── dataset.json # metadata of the dataset
│   ├── imagesTr # training set rawdata
│   │   ├── sub-verse004_0000.nii.gz # _0000: single input channel (greyscale)
│   │   ├── sub-verse007_0000.nii.gz
│   │   └── ...
│   ├── imagesTs # test set
│   │   ├── sub-verse005_0000.nii.gz
│   │   └── ...
│   └── labelsTr # training set label (heatmap)
│       ├── sub-verse004.nii.gz # no _0000 & same name as raw
│       ├── sub-verse007.nii.gz
│       └── ...
├── Dataset102_VerSe2020_2D
├── Dataset103_VerSe2019
└── Dataset104_VerSe2019_2D
```

3. UNet Preprocessing, Training, Estimating

nnUNet을 기반으로 전처리 시 사용한 projected image에 대한 학습을 시행한다. 이때, GPU의 활용을 최대화하기 위해서는 각 GPU 별로 fold를 지정하여 학습할 수 있다. 우선 nnUNet의 preprocess를 한다.

```
nnUNetv2_plan_and_preprocess -d DATASET_ID
```

그 이후 이미지가 project 되어있으므로 2D로 학습을 한다. 아래 예시는 GPU가 다섯개 있을 때의 5-fold CV 모델을 학습하는 방법이다. (106은 데이터셋 번호)

```
CUDA_VISIBLE_DEVICES=4 nnUNetv2_train 106 2d 0 --npz &
CUDA_VISIBLE_DEVICES=5 nnUNetv2_train 106 2d 1 --npz &
CUDA_VISIBLE_DEVICES=6 nnUNetv2_train 106 2d 2 --npz &
CUDA_VISIBLE_DEVICES=7 nnUNetv2_train 106 2d 3 --npz &
CUDA_VISIBLE_DEVICES=3 nnUNetv2_train 106 2d 4 --npz &
wait
```

그 후 가장 좋은 fold를 선택한다.

```
nnUNetv2_find_best_configuration -c '2d' -f 0 1 2 4 106
```

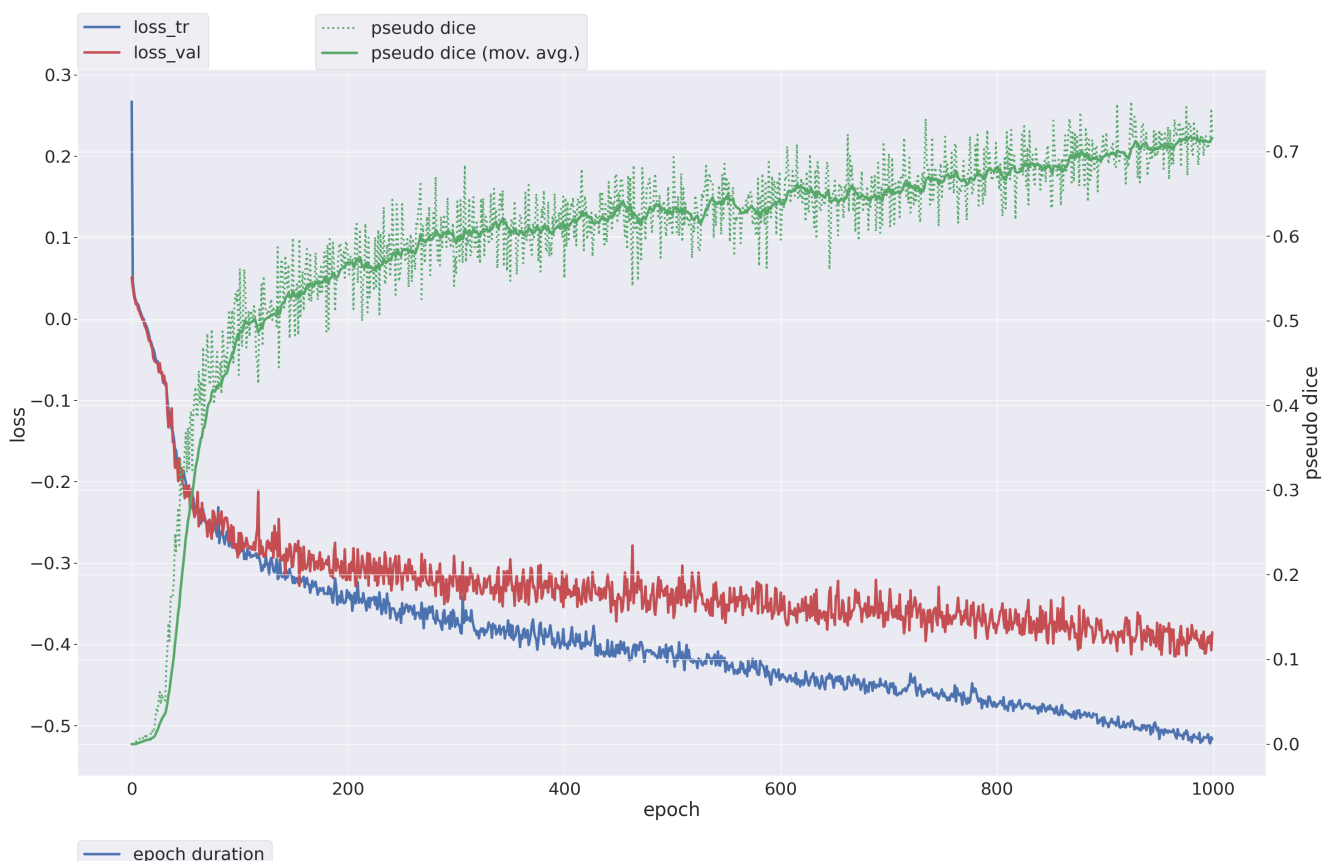
이 결과는 ensemble하여 주는데, sagittal plane과 coronal plane을 ensemble할 경우의 위와 같이 실행하면 된다. 하나의 모델만으로 하여 ensemble이 필요없는 경우 다음과 같이 실행한다.

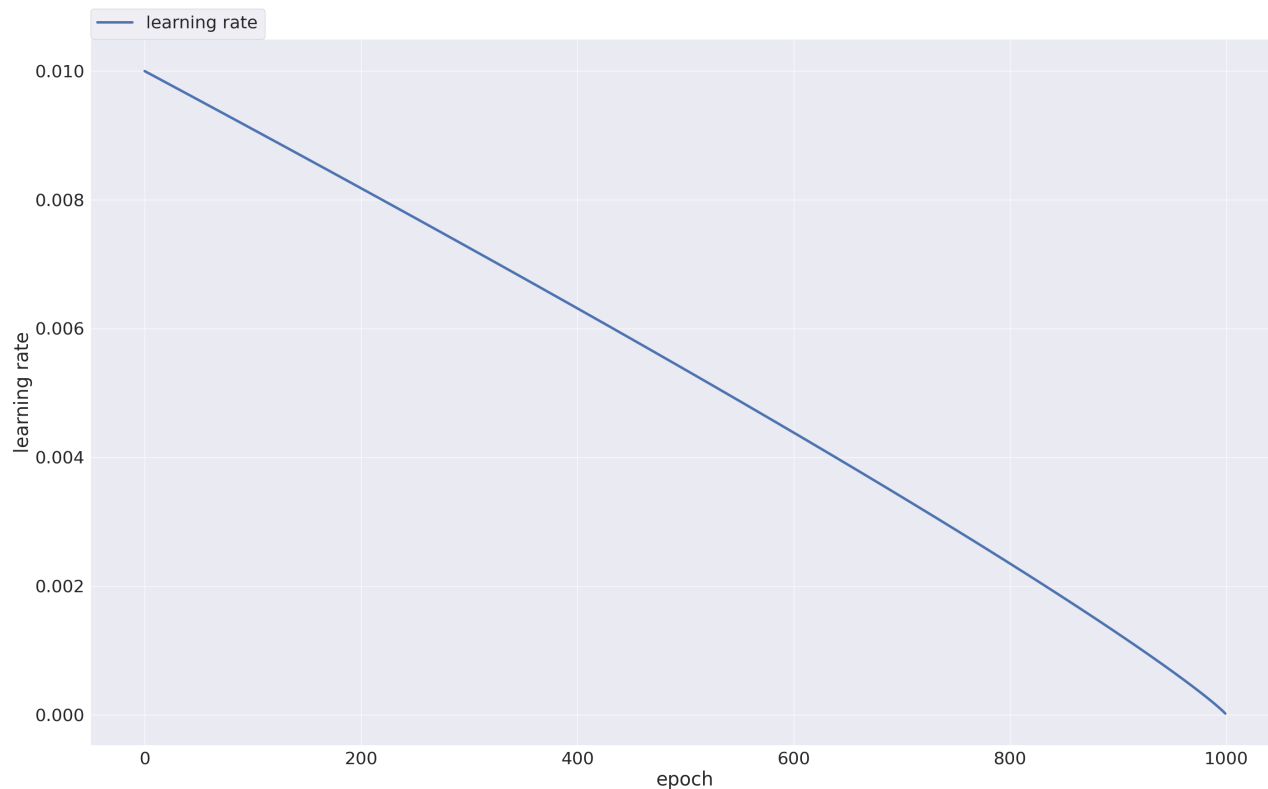
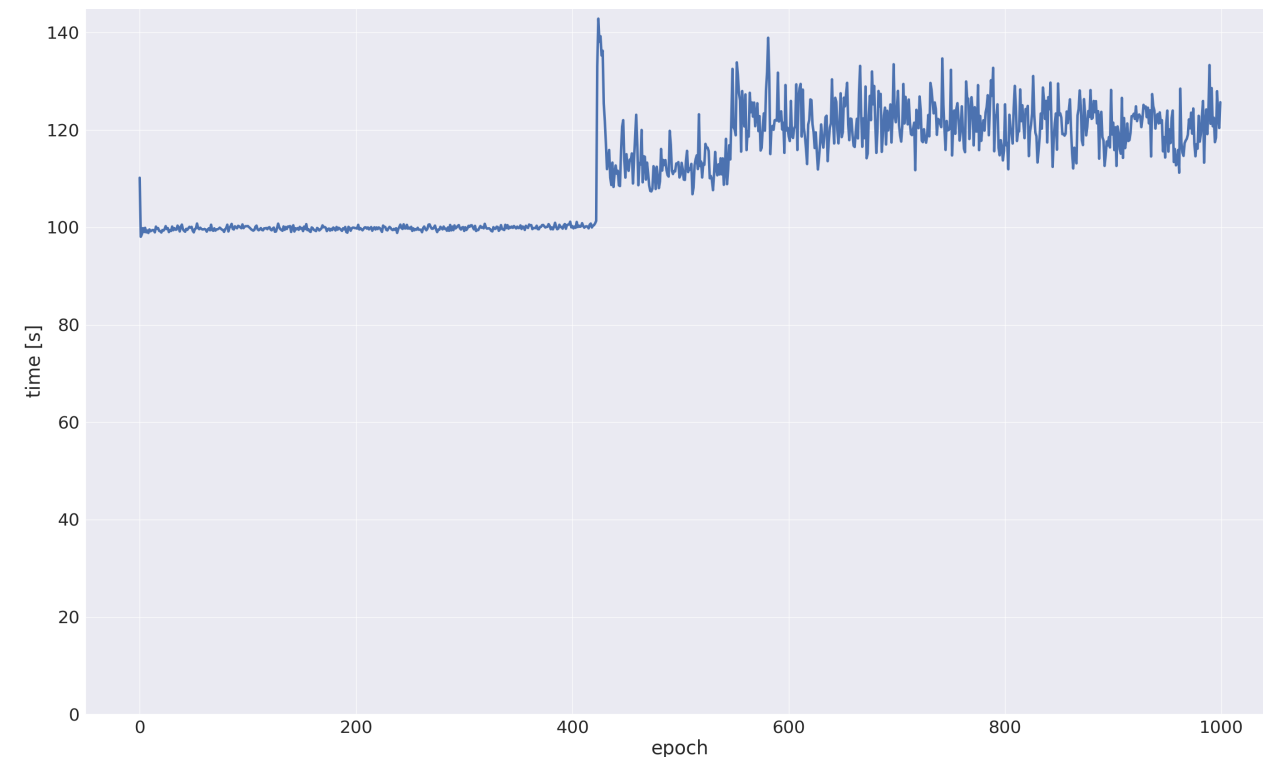
```
nnUNetv2_find_best_configuration -c '2d' -f 0 1 2 4 --disable_ensembling 106
```

그 후 predict한다.

```
nnUNetv2_predict -i INPUT_FOLDER -o OUTPUT_FOLDER -d DATASET_NAME_OR_ID -c
CONFIGURATION --save_probabilities
```

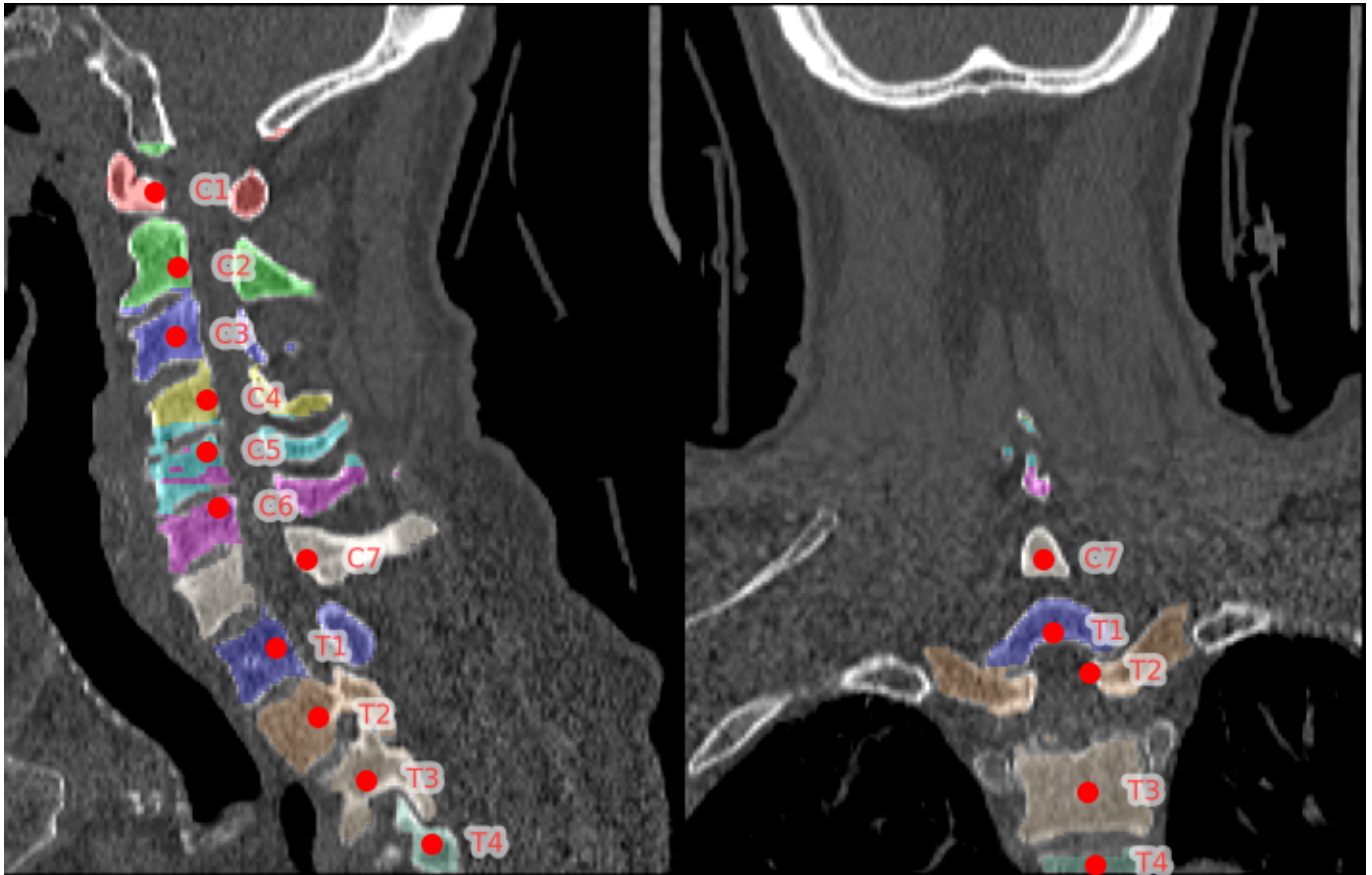
Train의 결과도 진행 시 다음과 같은 이미지로 확인할 수 있다.






4. Centroid Inference

추론한 이미지 파일의 경로를 지정하여 `example_image_median.py`를 실행한다. 실행 후에는 지정된 경로에 저장된 다음과 같은 사진을 확인할 수 있다. 이때, EDA를 통해 Threshold 값을 수동적으로 지정할 수도 있다. 우리의 기준은 50이다.



Centroid Inference 데모 실행 방법

모델을 학습할 시간이 없어 Centroid Inference만을 확인해보고 싶다면, 이  Github Repo를 clone한 후, repo 아래 다음 파일들을 넣어라.

Demo Files

그 후 `example_image_demo.py`를 실행하면 된다.