

복부 장기 CT 시각화 프로그램



정보컴퓨터공학부 201610323 이주승

정보컴퓨터공학부 201724564 전도현

정보컴퓨터공학부 201824435 김병관

지도교수 감 진 규

목 차

1. 서론.....	1
1.1. 연구 배경.....	1
1.2. 기존 문제점	1
1.3. 연구 목표.....	1
2. 연구 설계.....	2
2.1. 전체 파이프라인	2
2.2. 데이터 수집	2
2.3. nnUNet.....	3
2.4. 시각화 프로그램	4
3. 연구 내용.....	4
3.1. 기존 학습환경.....	4
3.2. 학습환경	5
3.3. 모델 학습.....	7
3.3.1. 모델 전처리	7
3.3.2. 모델 학습 및 예측	8
3.4. 평가 지표.....	9
3.4.1. Dice Similarity Coefficient (DSC)	9
3.4.2. Normalized Surface Distance (NSD)	9
3.5. 성능평가	10
3.6 시각화.....	14
4. 연구 결과 분석 및 평가.....	16
5. 결론 및 향후 연구 방향	18

6. 구성원별 역할 및 개발일정	20
7. 시연 계획서	21
8. 산학협력 활동보고서	22
9. 참고 문헌	22

1. 서론

1.1. 연구 배경

현대 의학에서 질병의 진단 및 치료를 위해 사용되는 영상으로 CT, MR, 초음파 영상 등이 주로 사용되고 있다. 컴퓨터단층촬영(CT), 자기공명촬영(MRI) 장치 등과 같은 의료기기를 통해 얻어진 인체내부의 단면 사진을 이용한 특정 장기의 분리, 종양 검출 및 3D 재구성 등과 같은 의료영상 처리기술은 치료나 수술을 위한 계획이나 방법 등을 결정하는데 매우 중요한 역할을 한다. 현재의 진단은 의료인들의 전문적인 지식과 경험에 의해 시각적으로 이루어지고 있지만, 컴퓨터를 이용하여 진단을 한다면 진단 시간을 단축하고 진단의 정확성 및 많은 양의 영상 데이터를 처리할 수 있다.

1.2. 기존 문제점

복부 CT를 이용한 장기 Segmentation과 관련된 논문은 흔히 찾아볼 수 있었다. 그러나 Segmentation방식이나 목표로 하는 질병에 집중하고 있고, 이를 직접적으로 시각화 하는 프로그램은 찾기 어려웠고, 기존의 시각화 프로그램인 3D Slicer의 경우 많은 기능을 가지고 있지만 프로그램을 이용하기에 무거운 측면이 있고, 별도의 전문적인 지식이 요구된다. 많은 Segmentation기법에 대한 연구가 활발히 진행되는 가운데 우리는 이러한 Segmentation된 이미지를 실제 환경에서 보다 사용하기 용이하도록 시각화 프로그램을 제작하고자 한다.

1.3. 연구 목표

해당 논문에서는 복부 CT를 통해 간, 신장, 비장, 췌장을 "AbdomenCT-1K: Is Abdominal Organ Segmentation A Solved Problem?"의 Fully Supervised Learning benchmark을 이용하여 Segmentation을 진행하고 이를 2D, 3D로 시각화 하는 프로그램을 만들 것이다. 학습 모델을 DSC, NSD를 통해 평가하여 원하는 수준의 학습이 진행되었는지 판단하고, 프로그램에서는 복부 CT와 학습을 통해 얻어낸 Label을 입력으로 하여 복부 단면을 보여준다. 이를 bounding box를 통해 특정 장기가 어떤 형태로 바뀌는지 쉽게 추적할 수 있도록 한다. 이러한 단면을 3D 렌더링을 통해 시각화하고, 확대/축소와 On/Off 기능을 통해 원하는 장기를 빠르게 볼 수 있도록 할 것이다.

2. 연구 설계

2.1. 전체 파이프라인

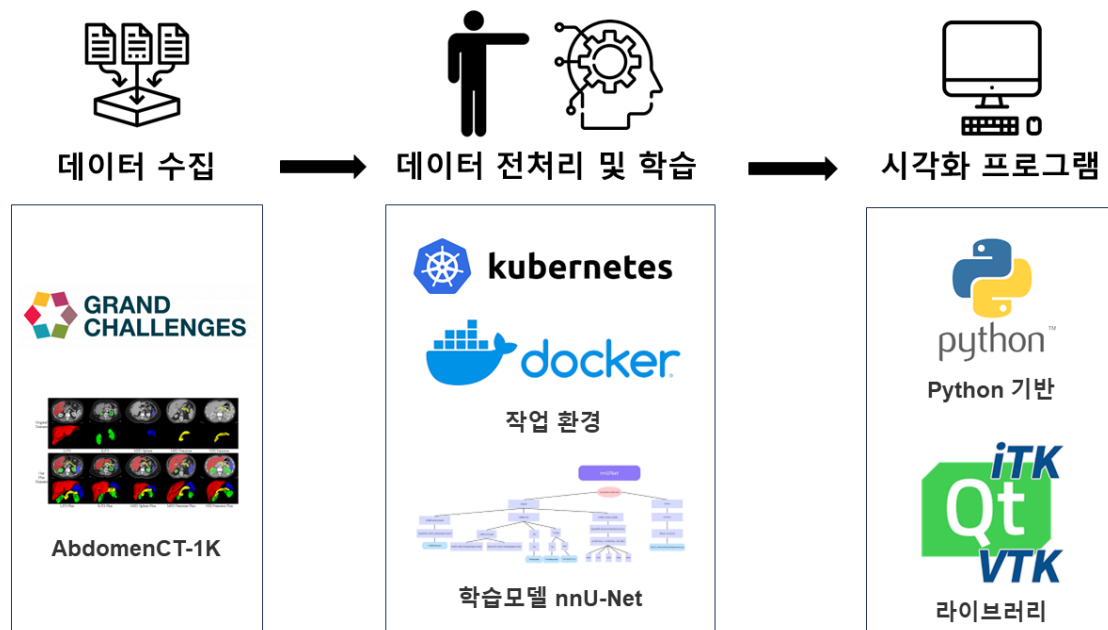


그림 1. 전체 파이프라인

2.2. 데이터 수집

이번 과제에서 다룬 데이터는 논문 "AbdomenCT-1K: Is Abdominal Organ Segmentation A Solved Problem?"의 공식 데이터셋을 사용하였다. 해당 논문에서는 기존에 있던 논문들의 단일 장기 데이터셋에서 간, 신장, 비장, 췌장의 복부CT 장기segmentation 데이터셋을 수집하였고 현재 활발히 연구되고 있는 주제인 fully supervised learning, semi-supervised learning, weakly supervised learning, continual learning에 대한 benchmark 1를 만들어 두었다. 우리는 fully supervised learning의 데이터셋과 성능 측정 지표를 사용하였다.

과제에서 사용할 데이터셋은 NIfTI(Neuroimaging Informatics Technology Initiative) 파일 형식을 가지고, 이는 의료 영상 데이터를 저장하는 데 사용되는 파일 형식이다. 이번 프로젝트에는 NIfTI 파일의 확장자인 .nii 파일을 입력 데이터로 사용할 것이다. Dicom과 다르게 NIfTI는 파일 하나에 3D 볼륨 데이터 정보를 전부가지고 있기 때문에 읽기 작업이 무겁다. NIfTI 파일은 크게header와body로 이루어져있다.

header의 경우 뇌 영상의 크기와 차원, 픽셀 값들의 데이터형식, 좌표계와 좌표축, 각 픽셀 크기 등과 같은 파일의 메타데이터를 저장한다. body의 경우 header에서 정의된 데이터 형

식에 따라 실제 픽셀 값을 저장한다. 보통 3D 또는 4D배열로 구성된다.

2.3. nnUNet

기존의 의료영상데이터를 다루는 모델에는 뇌, 간과 같이 서로 다른 장기간의 segmentation방식에 큰 차이가 있으며 표준화된 방식이 없었다. 이를 해결하고자 다양한 데이터에 대해 표준화된 파이프 라인을 찾을 수 있도록 nnUNet이 만들어지게 되었다. nnUNet의 hyper-parameter에는 Fixed configurations, Rule-Based configurations, Empirical configuration가 있다.

Fixed configuration는 learning rate, optimizer, loss function, epoch와 minibatch의 개수 등 고정되어 있는 parameter를 포함하고 있다.

Rule-Based configuration은 영상데이터의 종류에 따라 어떻게 normalization할 것인가, resize를 어떻게 할 것인가, segmentation을 하기 위해서는 patch size가 중요한데 해당 size를 몇으로 설정할 것인가, Network가 몇 번 pooling할 것인가 등을 포함하고 있다.

Empirical configuration은 학습시에는 2D, 3D, 3D cascade configuration을 모두 사용하고, 예측할 때는 2D, 3D, 3D cascade 중 하나 또는 세 경우를 ensemble하는 방식을 제공한다. 또한, cross-validation의 결과에 따라 네트워크를 어느 방향으로 변화시킬 것인지 결정한다.

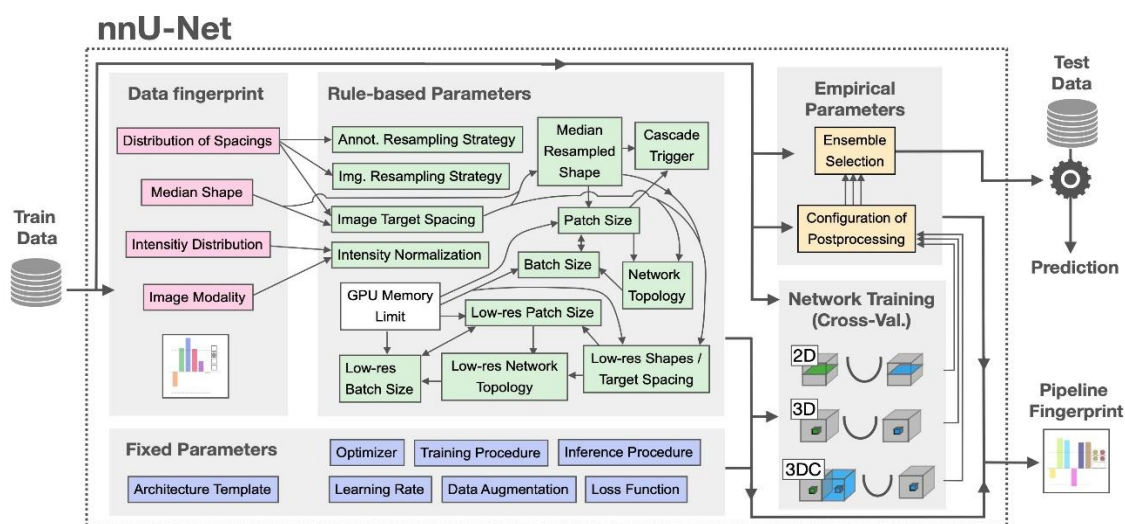


그림 2. nnUNet

2.4. 시각화 프로그램

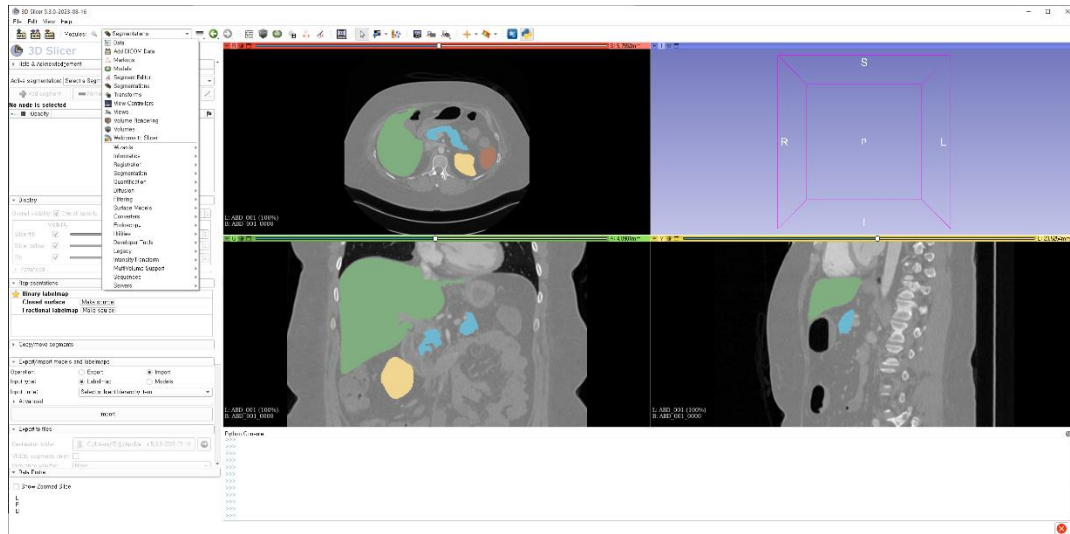


그림 3. 3D Slicer

기존의 시각화 프로그램인 3D Slicer의 경우 프로그램이 무겁고, 너무 기능이 많아 3D Slicer에 별도의 지식이 필요하다. 이번 과제에서 제작한 프로그램은 기존 프로그램보다 기능은 적지만, 직관적이고 가벼우며 필수적이라고 생각되는 기능을 탑재하고 있다. 아래는 본 과제에서 제작된 시각화 프로그램이다.

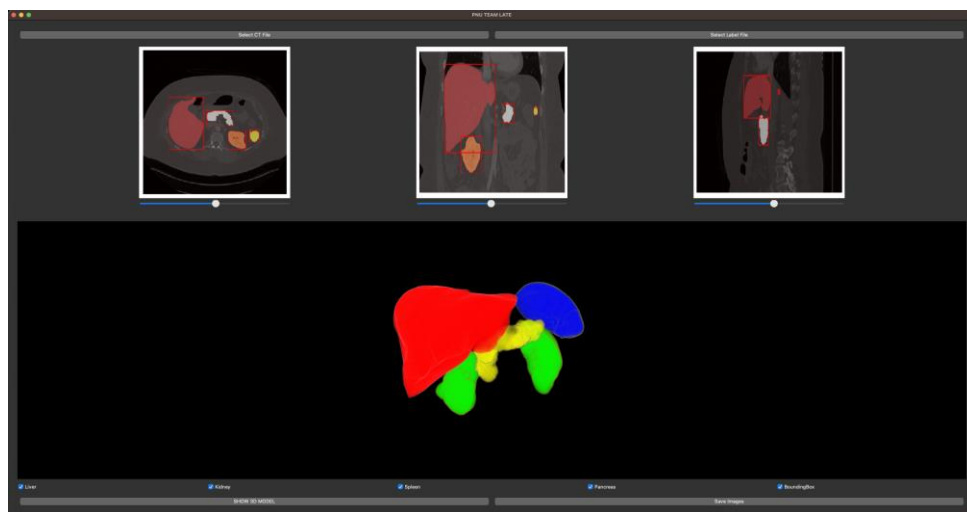


그림 4. PNU TEAM LATE 완성된 시각화 프로그램

3. 연구 내용

3.1. 기존 학습환경

기존에는 개발환경으로 학과서버를 사용하였다. 해당 서버는 개발 언어로 Python을 채택하

였고, CPU의 경우 Intel(R) Xeon(R) CPU E5-2620 v4 모델을 사용하였다. GPU의 경우 NVIDIA GeForce GTX Titan Xp D5X 12GB 모델 7개를 사용하였으며 해당 사양을 원활하게 사용하기 위해 CUDA를 사용하였다. 그러나 서버 문제로 인해 AI서버(GPU서버)를 사용하게 되었다.

3.2. 학습환경

기존에는 학과서버를 사용하였지만 서버문제로 인해 AI 서버(GPU 서버)를 사용하게 되었다. GPU서버란 인공지능 및 기계학습 작업을 수행하기 위해 고성능 그래픽 장치(GPU)를 장착한 서버를 말한다. Image Segmentation과 같이 대규모 데이터 학습에 적합하여 이전 보다 더 빨리 학습을 진행할 수 있게 되었다. AI 서버를 효과적으로 사용하기 위해 우리는 기존 Anaconda 환경이 아닌 쿠버네티스 플랫폼을 사용하였다.

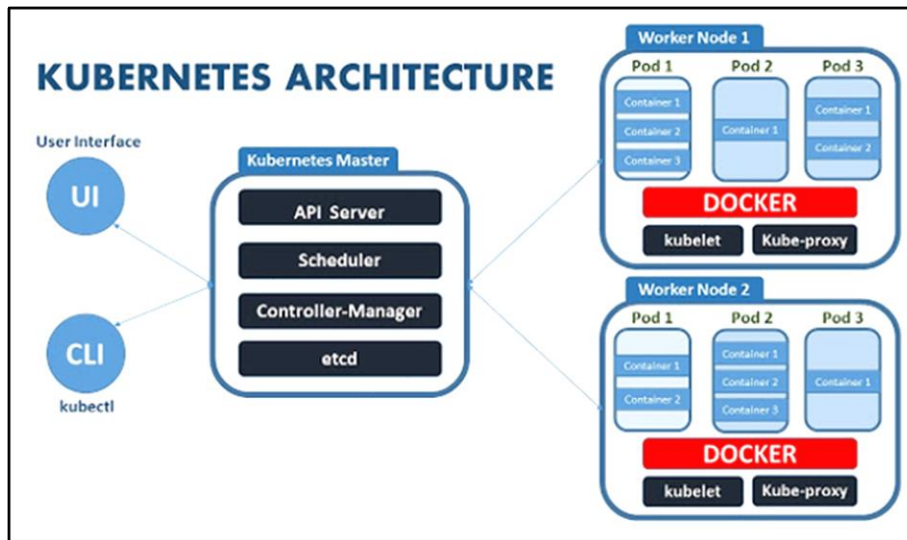


그림 5. Kubernetes Architecture

쿠버네티스는 Docker를 사용하는 컨테이너 기반 플랫폼이다. 다만 컨테이너를 하나만 띄워서 사용하는 도커와 달리 쿠버네티스는 하나이상의 컨테이너를 Pod로 묶어 컨테이너 간의 공유 리소스 및 호스트 환경을 쉽게 관리할 수 있었다. 또한 도커 기반이다 보니 Docker file을 수정하여 애플리케이션 환경을 쉽게 재현할 수 있었고 배포, 스케일링도 수월했다. 아키텍처는 위와 같다. 한 대의 마스터와 여러 대의 worker node로 구성되며 노드 안의 컴포넌트들의 생성을 위해 미리 정의된 yaml 파일이 존재해야 한다. 클라이언트는 오브젝트 생성, 변경, 삭제요구와 상태 조회를 하고 마스터는 k8s 클러스터, 파드 관리를 한다.

파드를 생성하는 방법으론 아래와 같은 명령어를 사용하였다.

```
python3 k8s_create_job.py -i kmskie96/nnunetv2:v9.05 -g 1 -c "cd /home/late && nvidia-smi"
```


3.3. 모델 학습

3.3.1. 모델 전처리

학습을 진행하기 위해서 아래와 같은 폴더 구조를 사용한다.

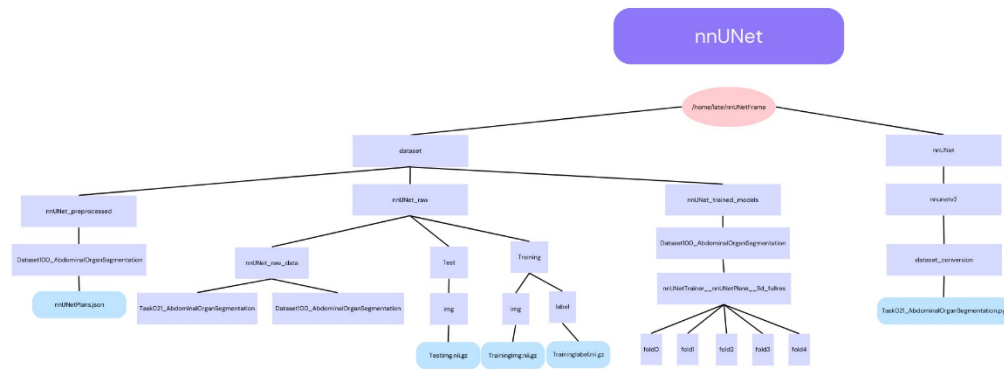


그림 9. nnUNet Architecture

우선 마운트 하게 될 /home/late에 nnUNetFrame 폴더를 만든 후 nnUNet 모델을 받아온다. 원본데이터와 마스크파일은 ~/Training 와 ~/Test 폴더안에 넣어둔다. 이후 명령어를 사용하여 아래와 같은 구조를 만든다.

```
python3 k8s_create_job.py -i kmskie96/nnunetv2:v9.05 -g 1 -c "cd /home/late &&
python3 Task021_BeyondCranialVaultAbdominalOrganSegmentation.py"
```

Task021_AbdominalOrganSegmentation/

├── dataset.json (dataset 의 metadata 를 포함한다)

├── imagesTr

├── imagesTs # optional

└── labelsTr

```
python3 k8s_create_job.py -i kmskie96/nnunetv2:v9.05 -g 1 -c "cd /home/late &&
nnUNetv2_convert_MSD_dataset -i Task021_AbdominalOrganSegmentation -
overwrite_id 100"
```

MSD dataset 을 nnUNet 형식으로 변환하여 기존에 있던 021 의 dataset 에 overwrite, Dataset100_AbdominalOrganSegmentation Directory 를 생성한다

```
python3 k8s_create_job.py -i kmskie96/nnunetv2:v9.05 -g 1 -c "cd /home/late &&
nnUNetv2_plan_and_preprocess -d 100 -c 3d_fullres -np 8
```

앞서 만든 지정된 데이터셋 100 을 3D network(3d_fullresolution)를 이용하여 전처리하면, nnUNet_preprocessed 폴더안에 Dataset100_AbdominalOrganSegmentation 폴더가 생기고 그 안에 nnUNetPlans.json 파일을 열어 원하는 batch size 로 수정할 수 있다. 그럼 학습하기위한 준비가 끝난다.

3.3.2. 모델 학습 및 예측

```
python3 k8s_create_job.py -i kmskie96/nnunetv2:v9.05 -g 1 -c "cd /home/late &&
nnUNetv2_train Dataset100_AbdominalOrganSegmentation 3d_fullres 0 1 2 3 4
```

모델을 학습시키기 위해 nnUNetv2 를 이용하여 998 개의 영상데이터를 학습시키는 명령어를 사용하였다. Dataset 인 Dataset018_AbdominalOrganSegmentation 을 학습시킬 것이고 학습에 사용할 network 를 3D data 에 대해 full-resolution 으로 처리하는 3d_fullres 로 지정한다. 더 정확한 학습을 위해 Fold 를 5 개로 나누는 5-fold cross validation 방식으로 진행하였다. 이후 아래 명령어로 imagesTs 에 저장된 이미지들을 학습시킨다.

```
python3 k8s_create_job.py -i kmskie96/nnunetv2:v9.05 -g 1 -c "cd
/home/late/nnUNetFrame/dataset/nnUNet_raw/nnUNet_raw_data/Dataset100_Abdomi
nalOrganSegmentation && nnUNetv2_predict -i imagesTs/ -o pred_nnUNetall/ -d
100 -c 3d_fullres -f 0 1 2 3 4
```

3.4. 평가 지표

3.4.1. Dice Similarity Coefficient (DSC)

DSC는 Segmentation결과의 정확성과 유사성을 측정하는 평가지표로 Ground Truth Mask¹와 Predicted Mask²를 3차원 NumPy 배열로 받아 각각의 교집합의 크기의 2배를 각 집합을 더한 값으로 나눈 비율로 계산된다. 수식으로는 $2 * |X \cap Y| / (|X| + |Y|)$ 로 표현할 수 있다. 이때 $|X|$ 는 Ground Truth Mask의 cardinality이고 $|Y|$ 는 Predicted Mask의 cardinality이다. DSC는 0에서 1의 사이 값을 가지며 1에 가까울수록 모델의 민감도와 정밀도가 높다.

3.4.2. Normalized Surface Distance (NSD)

NSD는 Ground Truth Mask와 Predicted Mask의 각 경계 간의 겹침을 측정하는 평가지표이다. 아래의 그림은 NSD를 시각적으로 설명하고 그에 따른 수식을 보여준다.

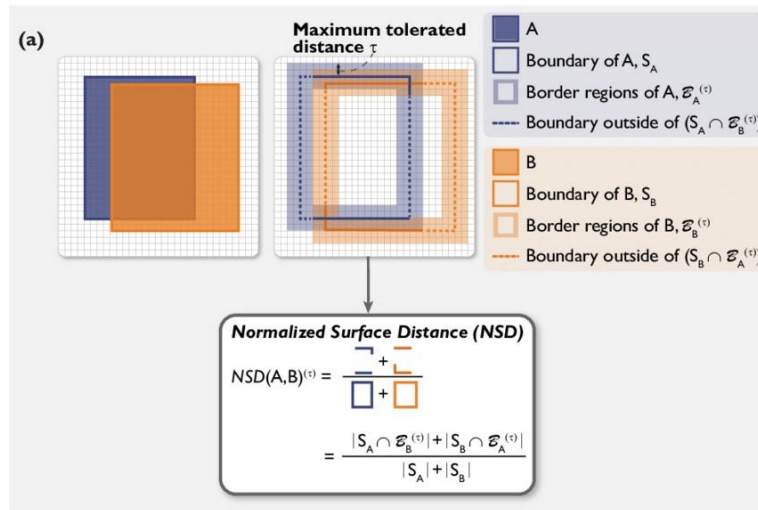


그림 10. NSD 수식

DSC만을 사용하지 않고 NSD 또한 사용하는 것은 DSC는 경계 오류를 잘 반영하지 못하기 때문이다. 그러나 NSD는 경계 오류에 민감하게 반응하며 이는 실제 임상 작업에 아주 중요하다. 또, NSD는 작은 경계 이탈은 무시한다는 점에 강점이 있다. 여러 관찰자 간의 작은 차이나 오류는 불가피하게 생길 수밖에 없으므로 이를 무시한다는 이점이 생기는 것이다. 아래의 그림 1은 NSD의 기능을 설명하기 위한 예시이다. 예시 그림의 score를 보면 알 수

¹ 데이터셋의 이미지나 영상에 대한 정확한 레이블 또는 참 값을 나타내는 이진 마스크

² 머신 러닝 모델 또는 컴퓨터 비전 알고리즘에 의해 예측된 마스크

있듯이 Segmentation 결과 경계 오류가 발생하였으나 DSC는 높은 score를 기록하고 있다. 그러나 NSD는 이를 포착하여 score가 DSC보다 낮게 나오는 것을 알 수 있다.

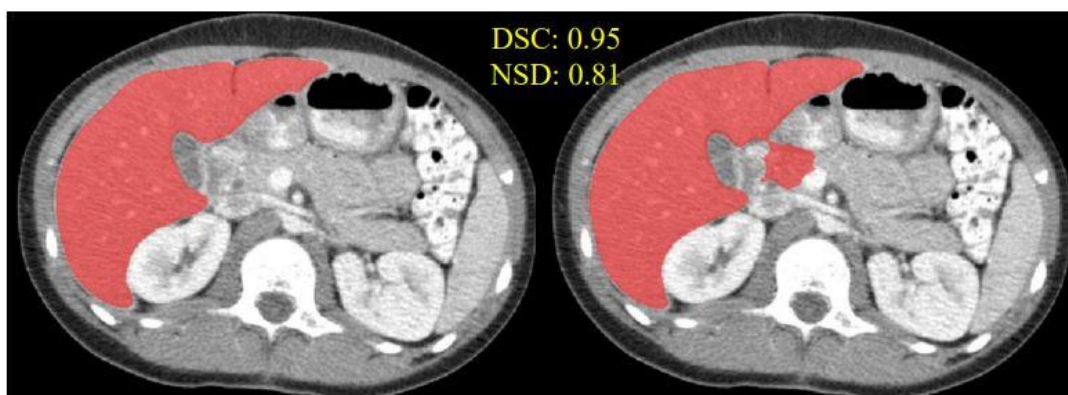


그림 11. (좌)Ground Truth, (우)Segmentation

3.5. 성능평가

총 1000개의 복부CT이미지와 Label파일 각각을 넣고 모델을 학습시켰다. 아래는 Batch size를 2와 5, 10으로 설정하여 epoch를 100까지 진행했을 때 Dice Score, Train Loss, Validation Loss, Learning rate를 측정한 것이다.

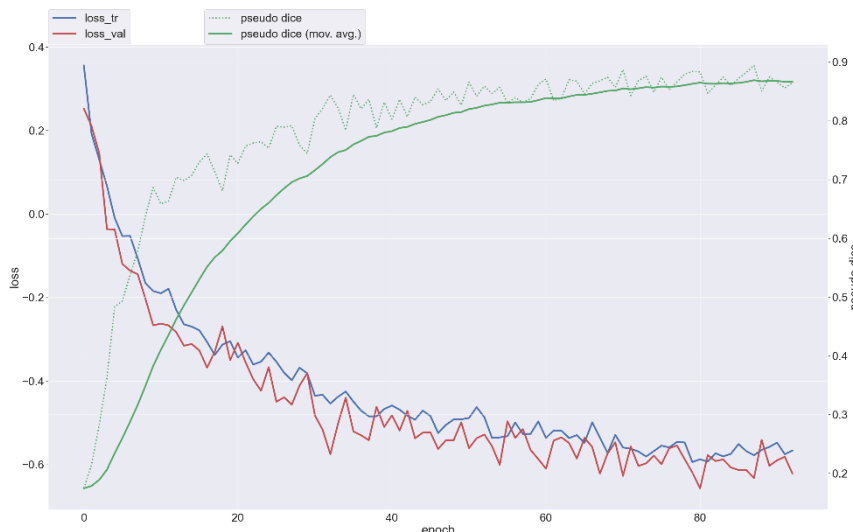


그림 12. batch size 2

Batch size가 2인 모델의 경우, Dice Score는 epoch가 진행됨에 따라 0.9에 수렴하는 결과를 보이고 있다. 1에 가까울수록 실제 이미지와 예측 결과가 유사하다는 것을 나타낸다. 모델이 훈련 데이터를 학습하면서 발생하는 오차나 손실을 나타내는 Train Loss의 경우에는 epoch가 진행됨에 따라 -0.6로 수렴하고 있다. 모델의 성능을 평가하기 위해 사용되며, 모델이 학습한 데이터가 아닌 검증 데이터에 대한 손실을 나타내는 Validation Loss 또한 epoch

가 진행됨에 따라 -0.6으로 수렴하고 있다.

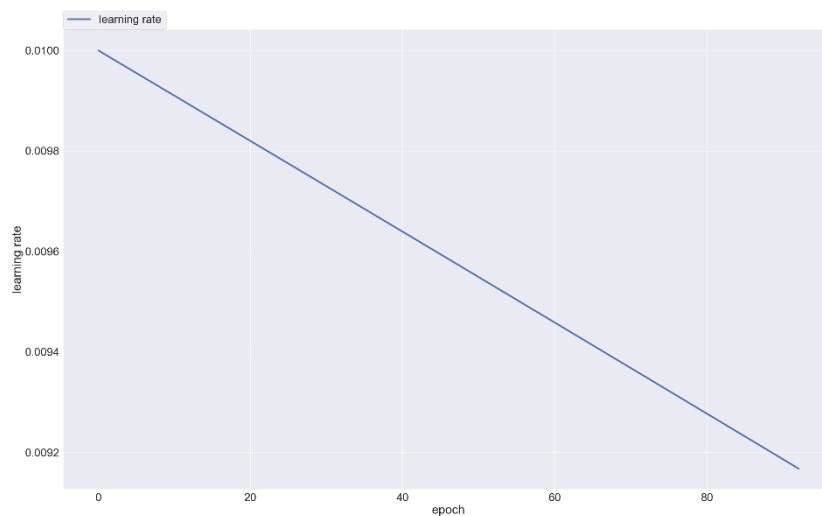


그림 13. batch size 2 Learning rate

nnUNet의 Learning rate는 0.01로 고정되어 있고, epoch가 진행됨에 따라 rate가 점차 감소하는 것을 볼 수 있다. Learning rate가 너무 작다면 수렴이 늦어지고 너무 크다면 loss값이 불안정해지는 경향이 있는데 그래프의 형태를 보아 Learning rate 적절하다고 추측할 수 있다.



그림 14. batch size 5

Batch size가 5인 모델의 경우, Dice Score는 epoch가 진행됨에 따라 0.9에 수렴하는 결과를 보이고 있다. Train Loss와 Validation Loss 또한 epoch가 진행됨에 따라 -0.8로 수렴하고 있다.

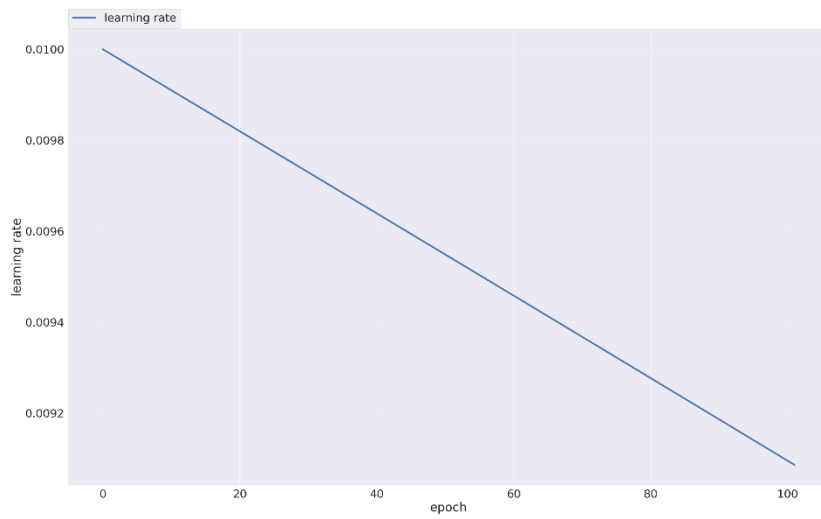


그림 15. Batch size 5 Learning rate

Batch size가 5인 모델의 경우에도 Learning rate는 감소하는 경향을 보인다.

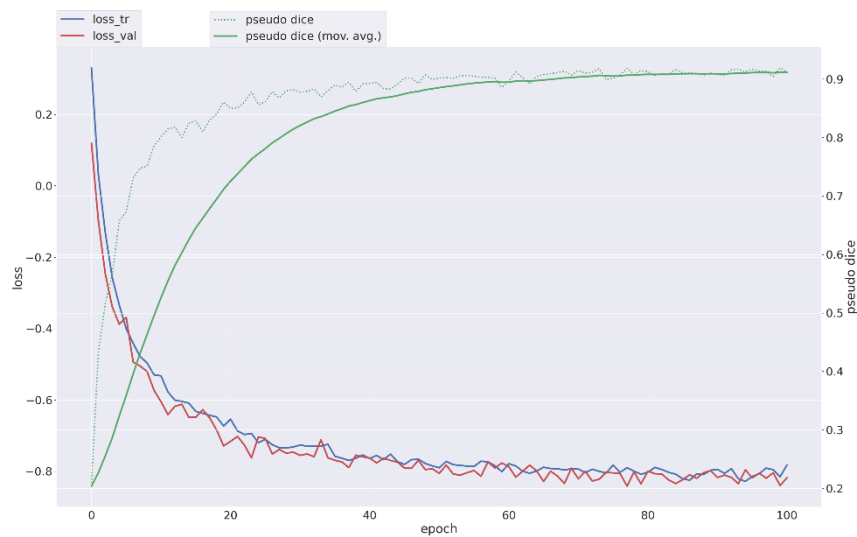


그림 16. batch size 10

Batch size가 10인 모델의 경우, Dice Score는 epoch가 진행됨에 따라 0.9에 수렴하는 결과를 보이고 있다. Train Loss와 Validation Loss 또한 epoch가 진행됨에 따라 -0.8로 수렴하고 있다.

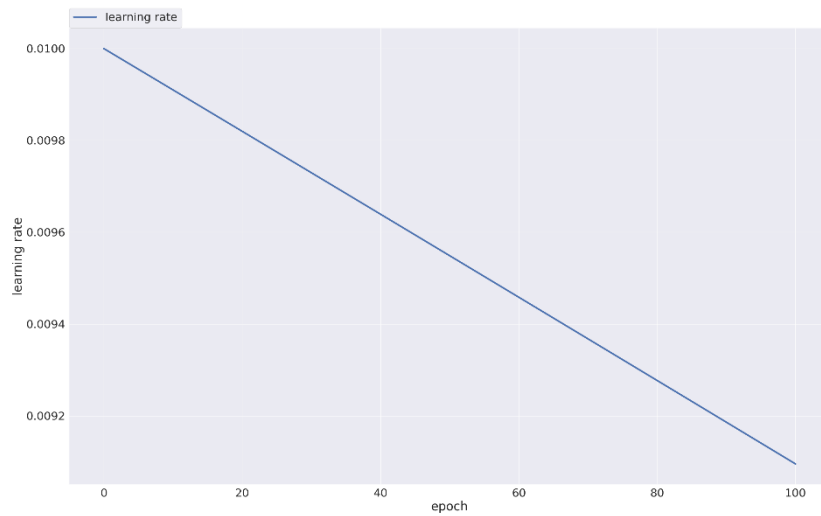


그림 17. Batch size 10 Learning rate

Batch size가 10인 모델의 경우에도 Learning rate는 감소하는 경향을 보인다.

의료데이터 특성상 학습한 모델을 test할 데이터셋을 다양하게 구하기 힘들어 하나의 데이터 셋으로 predict하였다. 그래서 평가지표로 사용한 score가 해당 data set에 의존하므로 학습 모델의 성능을 정확하게 표현한다고 생각하기 어렵다. 또 batch size를 2, 5, 10만을 사용하여 그 외의 값으로 batch size를 설정할 경우 loss와 dice score에 어떠한 영향을 끼치는지 알기 힘들다. 학습한 모델의 train loss와 validation loss는 각각 수치의 차이는 있어도 같은 경향성을 보이므로 overfitting이 발생하지 않은 것으로 보인다.

아래는 언급한 평가 지표인 DSC와 NSD를 통해 각 장기 별 NSD의 평균을 낸 결과이다. 모두 소수점 4번째 자리에서 반올림하였다.

Table 1. batch size 2 평가지표

Organ	Liver	Kidney	Spleen	Pancreas
DSC	0.9435	0.9284	0.9405	0.6713
NSD	0.7428	0.8947	0.9241	0.6582

Table 2. batch size 5 평가지표

Organ	Liver	Kidney	Spleen	Pancreas
DSC	0.9719	0.9255	0.9355	0.8473
NSD	0.8639	0.9062	0.9287	0.9197

Table 3. batch size 10 평가지표

Organ	Liver	Kidney	Spleen	Pancreas
DSC	0.9748	0.9498	0.9666	0.8476
NSD	0.8723	0.9367	0.9748	0.9204

Batch size를 2로 설정하였을 때와 5, 10으로 설정하였을 때 Score의 차이가 전반적으로 존재한다. 그러나 5로 설정하였을 때와 10으로 설정하였을 때의 차이는 비교적 크지 않았고, 10으로 설정하였을 때 epoch당 걸리는 시간이 5로 설정하였을 때보다 많이 길었기 때문에 batch size를 5로 설정한 모델을 가지고 시각화를 진행하였다.

3.6 시각화

과제에서 주로 Python 언어로 개발과 학습을 진행했기 때문에 GUI 시각화 작업도 Python 언어로 채택했다. 따라서 Python에서 GUI 애플리케이션을 만들기 위한 라이브러리인 PYQT5를 사용하였다. CT 이미지를 시각화하기 위해서는 별도의 라이브러리가 역시 필요한데, 이는 ITK를 활용하여 2D 이미지를 시각화 하였다. ITK는 "Insight Segmentation and Registration Toolkit"의 약자로, 영상 처리와 컴퓨터 비전 분야에서 널리 사용되는 오픈 소스 라이브러리이며, 특히 의료 영상 처리에 사용된다. ITK를 이용하여 3가지 CT 측면을 시각화할 수 있는 Axial View(상·하단 뷰), Sagittal View (측면 뷰), Coronal View(전면 뷰) 뷰어를 만들었다.

하지만 Pyqt5와 같이 병행해서 사용하는 데 있어서 이미지의 사이즈가 출력하고자 하는 크기가 아닌 정확한 이미지의 크기만 출력되었고, 2가지 CT 이미지를 레이어링 해야 하기 때문에 크기 설정에 있어서 민감할 수밖에 없다. 따라서 이를 해결하고자 ITK의 복잡성을 줄이고 간단한 인터페이스 기능을 중점으로 다룰 수 있는 SimpleITK 라이브러리를 활용하여, 원하는 크기의 이미지를 출력하였다.

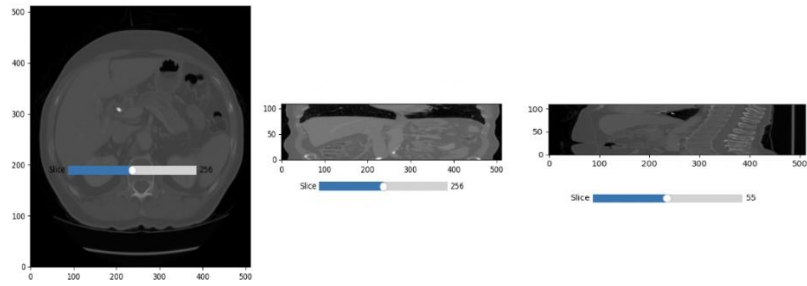


그림 18. PYQT5를 미사용 화면

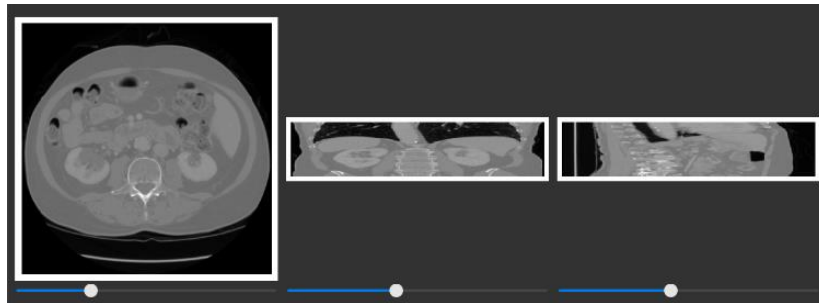


그림 19. PYQT5에 ITK를 적용한 크기가 한정된 화면

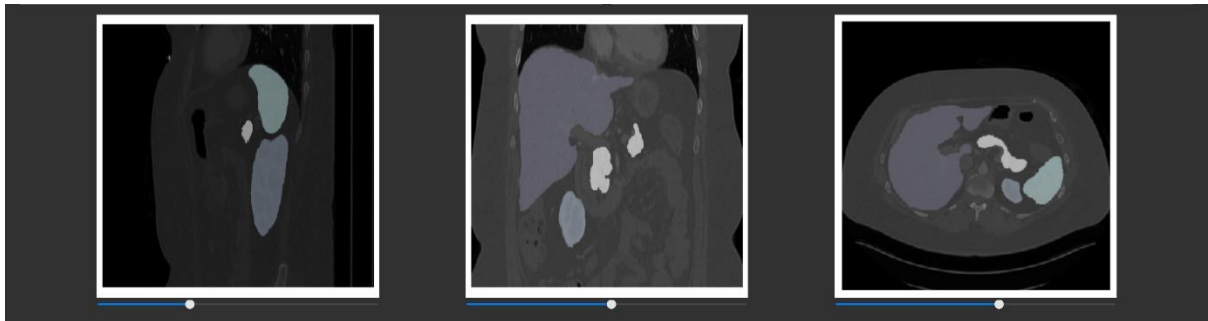


그림 20. PYQT5와 SimpleITK를 적용한 화면

ITK에서는 3D 모델링 하는 기능은 없기 때문에 별도의 기능을 따로 사용해야 한다. 따라서 이때 VTK 라이브러리를 활용하였다. VTK는 "Visualization Toolkit"의 약자로, 3차원 컴퓨터 그래픽스, 이미지 처리 및 시각화를 위한 오픈 소스 소프트웨어 시스템이다. VTK는 복잡한 3D 모델을 렌더링하고 상호 작용하는 기능을 제공하기 때문에, 해당 애플리케이션에 있어서 꼭 들어가야 하는 라이브러리다.

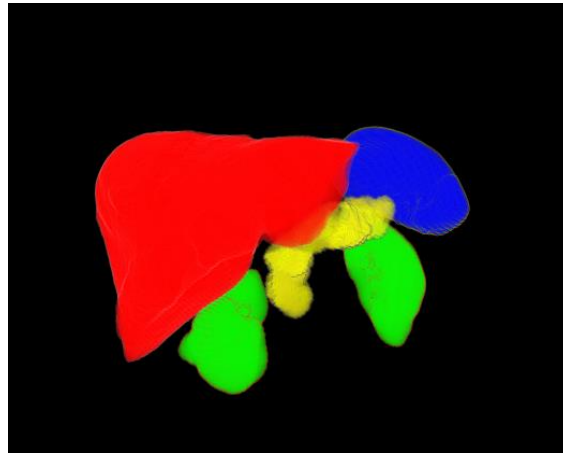


그림 21. VTK를 이용하여 만든 3D 모델

4. 연구 결과 분석 및 평가

제작한 GUI 애플리케이션은 다음과 같이 작동한다.

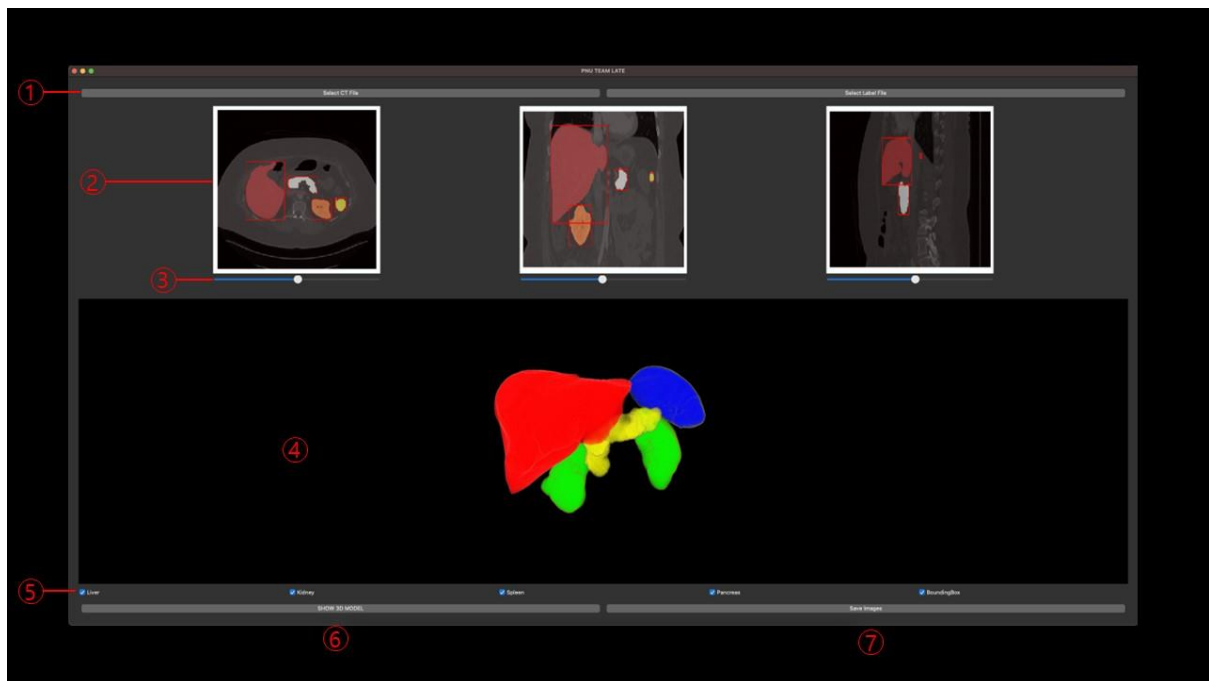


그림 22. 완성된 시각화 프로그램

① 애플리케이션 상단에 각 파일(Select CT File/Select LABEL File)을 선택하는 버튼이 있으며 파일을 선택하게 되면 버튼 아래 ②에 있는 3개의 CT 뷰어에 이미지를 출력한다.

② 뷰어 밑 ③ 슬라이더가 존재하다. 이를 조작하면, ② 즉 CT 뷰어에 출력되고 있는 이미지를 다음 CT 슬라이드로 넘어가게 해준다.

이때 ITK로 사용되는 CT 뷰어는 사용자가 지정한 슬라이더 바의 값을 CT 파일의 배열 값으로 지정하게 되며, 해당 배열에 있는 이미지를 PNG로 바꿔서 뷰어에 출력한다. 따라서 배열 값을 매번 수동으로 지정해주는 것은 비효율적이며 시각적으로도 안 좋기 때문에, 슬라이더 바를 사용해서 슬라이스(배열) 번호를 바꾸게 만들었다.

② ITK 뷰어에서는 이미지를 불러와도 ④VTK 뷰어에서는 바로 3D 모델링을 시각화 하지 않고 ⑥“Show 3D Model” 버튼을 눌러야 모델링을 실행하게 설정하였다. 그 이유는 3D 모델링은 환경에 따라 다르지만, 모델을 생성하고 뷰어에 시각화 하는 데 시간이 비교적 오래 걸리기 때문이다. 따라서 실수로 CT 파일을 선택하게 되거나 혹은 CT 파일을 변경했을 때 많은 리소스를 사용하지 않고, 시간을 줄일 수 있도록 별도의 생성 버튼을 만들었다.

④ VTK 뷰어에서 출력되는 3D 모델링 된 장기들은 뷰어 밑에 있는 ⑤체크 박스로 뷰어에 출력하고자 하는 장기를 볼 수 있다. 따라서 3D 모델링에선 Bounding Box 기능을 제외했다.

② ITK 뷰어에선 2가지 CT 이미지가 겹치기도 하며 정확한 범위를 쉽게 확인하기 위해서 ⑤ 체크 박스로 Bounding Box 기능을 구현하였지만, 정교한 박스를 만들기에는 어려움이 있어서 불완전한 상태이다. 이는 앞으로의 향후 연구 방향으로 목표를 잡을 수 있을 것이다.

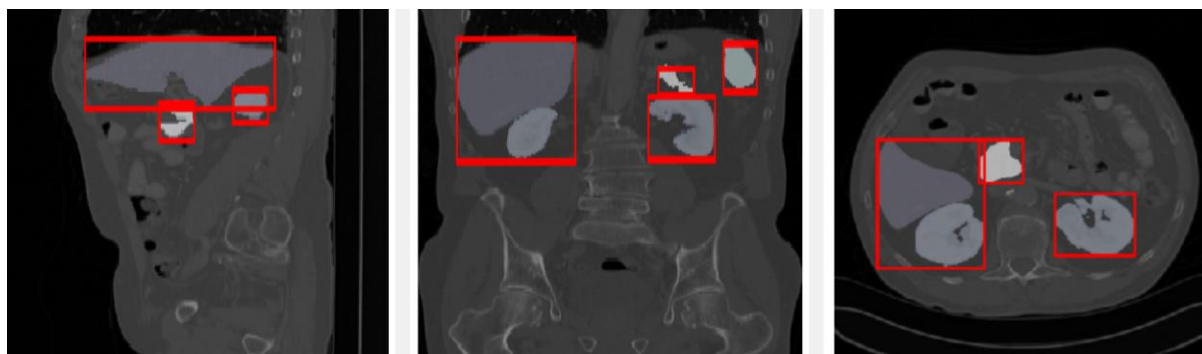


그림 23. Bounding Box의 불완전성

⑦에 Save Images 버튼을 누르면 TEAM_LATE.py 파일이 존재하는 경로 혹은 사용자가 지정한 경로에 하나의 이미지를 저장한다. 이미지는 현재 출력 되고 있는, Axial View(상·하단 뷰), Sagittal View (측면 뷰), Coronal View(전면 뷰)를 저장하고, VTK 뷰어에 띄워져 있는 3D MODEL을 저장한다.

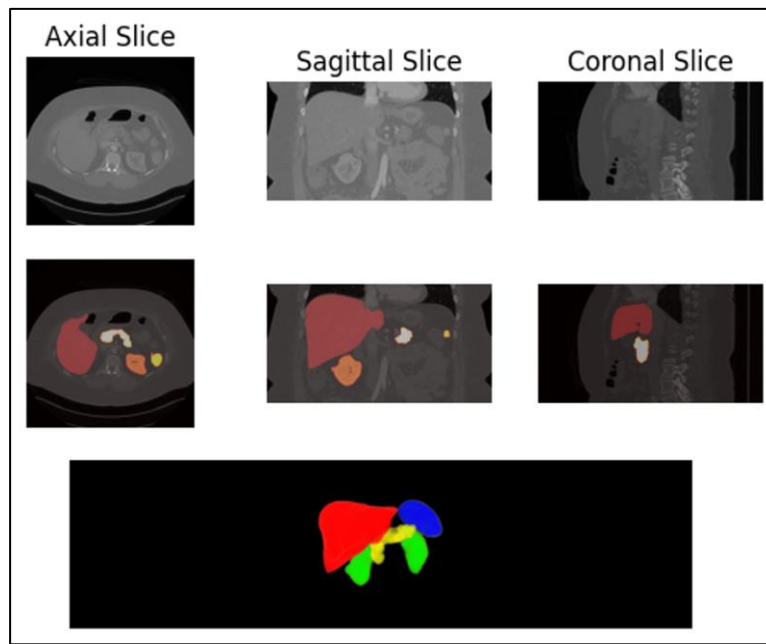


그림 24. Save Images 버튼을 누르면 생성되는 이미지 파일

5. 결론 및 향후 연구 방향

복부 내의 장기들에는 많은 질병들이 생겨 날 수 있다. 따라서 CT 검사를 통해 간암, 위암, 지방간, 췌장암, 급성 췌장염, 혈관 종 등 암 뿐만이 아니라 다양한 질병을 검사가 가능하다. 하지만 빠르고 확실하게 판별하기 어렵기 때문에 이를 돕고자 복부 CT의 장기들을 Segmentation을 진행하여 시각화 하는 프로그램을 개발하였다.

위 과정을 통해서 우선적으로 메디컬 데이터에 관한 학습을 통해서 CT 파일은 어떠한 형태로 이루어져 있고, 어떻게 다루는지 학습을 하였다.

nnU-Net을 통해서 메디컬 데이터를 다룰 수 있는 모델을 사용하여 Fully Supervised Learning(완전 지도학습)의 이해와 학습 후 결과를 도출하는 과정을 경험하며 머신 러닝에 대한 이해도를 높였다. 그리고 대표적으로 메디컬 데이터에 대한 시각화하고 다양한 기능을 제공하는 "3D slicer" 프로그램으로 3D 모델링에 대한 학습을 진행하고 어떻게 따로 Segmentation을 진행한 모델을 3D 모델링 하는지 학습을 하였다.

이후 "3D Slicer"는 팀이 목표로 하는 기능 이외에 너무 많은 기능을 소지하고 있기 때문에 프로그램이 다소 무겁다고 판단하여 별도의 프로그램을 만들기로 결정한 후 제작을 진행하였다. 프로그램을 별도로 만드는 과정에선 어떻게 효율적으로 GUI 인터페이스를 제작하는지 학습하며 PYQT5에 대한 이해도를 올렸다. 또한, 메디컬 데이터는 어떻게 시각화 가능한

지에 대한 학습을 통해 ITK와 VTK를 학습하고 뷰어를 제작하였다.

이를 통해 팀이 목표하고자 했던 장기 복부의 Segmentation과 시각화를 다루는 데 있어서 목표를 달성하였다. 하지만 이 과제에서는 할 수 있는 것이 더욱 있으며, 앞으로의 방향성도 다양하다. 우선 Fully Supervised Learning(완전 지도학습)이 아닌 Semi supervised Learning(반지도 학습)을 통해서 왜 Fully Supervised Learning이 더욱 좋은 지 혹은 Semi supervised Learning과 비교했을 때 어떠한 점수 차이가 나올 수 있는지에 대한 비교를 할 수 있는 방법이 있을 것이다.

Semi supervised Learning은 레이블이 있는 데이터와 레이블이 없는 데이터를 사용해서 학습을 진행하고 결과물을 도출하는 방법이다. Semi supervised Learning에 대해서 학습과 비교를 진행은 하였으나, Semi supervised Learning의 데이터가 Fully Supervised Learning에서 사용되는 데이터와 다르며, 학습을 진행하는 방식에 있어서 방법이 다르고 다소 어려웠기 때문에 해당 방법을 진행을 중단하였다.

또 하나의 차후 연구 방향성으로는 완성된 시각화 프로그램에 추가 기능을 더욱 추가하는 것이 될 수도 있다.

예를 들어 현재는 복부 장기들을 시각화하고 있지만, 차후 프로그램을 시작할 때 복부 장기 뿐만 아니라 다른 장기들을 나열하고 선택된 장기들을 시각화할 수 있는 방식으로 연구가 가능할 것이다. 또한 한 이미지의 현재 출력된 정보를 저장하는 것이 아닌, 모든 슬라이드를 한 번에 저장하고 GIF 파일로 계속해서 슬라이드를 연속적으로 보여주는 방법 또한 있을 것이다.

이처럼 방법은 다양하지만, 어떻게 무엇을 추가적으로 제작하는 방식에 따라서 많은 방향성이 제공될 수 있을 거 같다.

6. 구성원별 역할 및 개발일정

5월			6월			7월					8월				9월				10월		
22	29	5	12	19	26	3	10	17	24	31	7	14	21	28	4	11	18	25	1	7	15
관련 지식 습득																					
			학습 모델 개발																		
			테스트																		
								중간보고													
									데이터 <u>전처리</u>												
									모델 최적화, 시각화												
															프로그램 개발						
																		최종 테스트			
																			최종 보고서 / 발표 준비		

이름	역할
이주승	<ul style="list-style-type: none"> - 학습 모델 개발 및 최적화 - 학습 모델 테스트 - 데이터 전처리 및 시각화
전도현	<ul style="list-style-type: none"> - 학습 모델 개발 및 최적화 - 예측 결과 도출 및 시각화 - 데이터 전처리 모듈 수정
김병관	<ul style="list-style-type: none"> - 데이터 전처리 - 학습 모델 테스트 - 학습 모델 최적화 및 오류 수정
공통	<ul style="list-style-type: none"> - 필요한 지식 습득 - 보고서 작성 및 검토 - 발표 및 시연 준비

7. 시연 계획서

2023전기 졸업과제 시연계획서

번호	1	팀명	LATE	분과	A
과제명	복부 장기 CT 시각화 프로그램				
팀장	이주승				
발표자	이주승, 김병관				
소요시간	5분-10분				
사용 장비 및 네트워크 환경	개인 노트북 2대				
시연 방법	발표는 제출된 동영상으로 대체, 개인 노트북으로 과제 결과물 시연				
기타 주의사항	-				

8. 산학협력 활동보고서

(1) 성능 평가 멘토링 반영

Batch size를 5로 설정하여 Segmentation을 진행하였으나 비교군이 부족하다는 피드백을 통해 batch size를 2, 5, 10으로 설정하여 Segmentation을 진행하였다. Batch size 2와 5의 경우 장기 별 성능평가지표의 차이가 비교적 드러났지만, 5와 10의 경우 시간대비 성능이 크게 차이가 없었다. 다음에는 parameter의 비교 아니라 다른 모델을 사용하여 비교하는 방향도 생각해볼 수 있다.

(2) 연구 방향성 멘토링 반영

본 과제에서는 Fully Supervised Learning방식을 채택하였으나 추후에는 Semi Supervised Learning이나 Weakly Supervised Learning, Continual Learning방식으로 연구를 진행할 수 있을 것이다.

9. 참고 문헌

[1] AbdomenCT-1K: Is Abdominal Organ Segmentation A Solved Problem?

[2] nnU-Net: a self-configuring method for deep learning-based biomedical image segmentation

[3] <https://github.com/MIC-DKFZ/nnUNet>

[4] <https://itwiki.kr/w/%EC%BF%A0%EB%B2%84%EB%84%A4%ED%8B%B0%EC%8A%A4>