

2023 전기 졸업과제 중간보고서

“복부 장기 segmentation 및 시각화”

(Abdominal OrganSegmentation & Visualization)

분과 : 인공지능(A)

팀명 : LATE

지도교수 : 감진규

정보컴퓨터공학부 201610323 이주승

정보컴퓨터공학부 201724564 전도현

정보컴퓨터공학부 201824435 김병관

부산대학교 정보컴퓨터공학부

\* \* \*

---

## 목 차

1. 요구조건 및 제약 사항 분석에 대한 수정사항.....	1
1.1. 요구조건 .....	1
1.2. 제약 사항 분석에 대한 수정사항.....	1
2. 설계 상세화 및 변경 내역.....	1
2.1. 데이터.....	1
2.2. 모델 설계 .....	2
2.3. 모델 구조 .....	3
3. 갱신된 과제 추진 계획.....	4
4. 구성원별 진척도 .....	4
5. 보고 시점까지의 과제 수행 내용 및 중간 결과.....	5
5.1. 모델 구현 .....	5
5.2. 모델 학습 및 예측.....	6
5.3. 모델 평가 지표.....	8
5.4. 실제 적용 .....	8
6. GUI 예상안.....	9
7. 참고 자료 .....	10

## 1. 요구조건 및 제약 사항 분석에 대한 수정사항

### 1.1. 요구조건

2016년 기준 우리나라 전체 암 발생 확률 9위인 담관암을 진단하려면 영상을 육안으로 변별해내야 한다. 의료진들의 담관암을 육안으로 찾는 수고를 덜 수 있도록 담관을 segmentation할 것이다. 그러나 의료데이터는 민감하므로 오픈소스로 제공되는 데이터셋을 사용할 예정이다. 하지만 오픈소스로 제공되는 담관에 관한 dataset은 공개되어 있지 않기 때문에, 담관의 주변 장기를 segmentation 하여 담관에 접근하는 방식을 채택하였다.

### 1.2. 제약 사항 분석에 대한 수정사항

담관에 대한 오픈된 dataset을 찾아볼 수 없어 주변 장기를 segmentation하는 것으로 수정하였다.

영상 데이터를 다루기 때문에 그 양이 방대하여 학습에 시간이 많이 걸릴 것으로 예측했었으나 학습 epoch를 낮추어 시간을 단축하였다.

영상 데이터에 쉽게 접근 및 조작하기 위해서, segmentation된 데이터를 GUI(graphical user interface)로 만들 것이다. 제작된 GUI 환경에서 누구나 쉽고 빠르게 장기들을 확인 수 있게, 복부 도면 위에 각 장기들의 Bounding Box의 형태로 만들어 접근할 것이다.

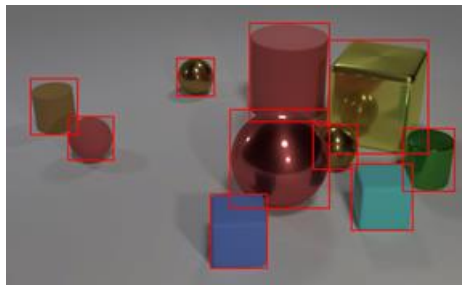


그림 1. Bounding box의 예시

## 2. 설계 상세화 및 변경 내역

### 2.1. 데이터

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

---

<sup>1</sup> 장기 segmentation 작업을 수행하기 위해 개발된 dataset과 그 작업에 대한 성능 측정 지표를 의미

과제에서 사용할 데이터셋은 NIfTI(Neuroimaging Informatics Technology Initiative) 파일형식을 가지고, 이는 의료 영상 데이터를 저장하는 데 사용되는 파일 형식이다. 이번 프로젝트에는 NIfTI 파일의 확장자인 .nii 파일을 입력데이터로 사용할것이다. Dicom과 다르게 NIfTI는 파일 하나에 3D 볼륨 데이터 정보를 전부가지고 있기 때문에 읽기 작업이 무겁다. NIfTI 파일은 크게 header와 body로 이루어져 있다. header의 경우 뇌 영상의 크기와 차원, 픽셀값들의 데이터형식, 좌표계와 좌표축, 각 픽셀 크기 등과 같은 파일의 메타데이터를 저장한다. body의 경우 header에서 정의된 데이터 형식에 따라 실제 픽셀 값을 저장한다. 보통 3D또는 4D배열로 구성된다.

## 2.2. 모델 설계

U-Net은 biomedical 분야에서 이미지분할(image segmentation)을 목적으로 만들어진 end-to-end 방식의 fully-convolutional network기반 학습 모델이다. nnUNet은 U-Net을 기반으로 확장시킨 3D버전으로 2D,3D 의료 영상을 segmentation하는데 사용하는 모델로 이번 과제에 사용하게 되었다.

영상 데이터 학습과정에서 nnUNet의 default epoch는 1000이었으나, 학습시간을 고려하여 5개의 fold로 나누어 각각의 fold를 100 epoch씩 학습하였다. 처음에 학습을 시작하였을 때는 batch size를 2로 설정하여 학습하였으나 속도와 정확도 향상을 위해 batch size를 5로 설정하여 학습하였다. batch size를 2로 설정하였을 때는 하나의 gpu를 사용하여도 무방하였으나 5로 설정하였더니 gpu를 최소 5개이상 사용해야한다는 사실을 확인 후 gpu를 분할하여 학습하였다.

학습시 nnUNet의 loss function은 Dice&cross-entropy를 사용하고, optimizer는 Adaptive Momentum Estimation<sup>2</sup>을 사용한다. learning rate는 0.01로 고정되어 있다. 이번 모델 학습에는 361개의 영상데이터를 통해 학습되었고, 테스트용 영상데이터가 따로 100개 존재한다.

## 2.3. 모델 구조

기존의 의료영상데이터를 다루는 모델에는 뇌, 간과 같이 서로 다른 modality간의 segmentation방식에 큰 차이가 있으며 표준화된 방식이 없었다. 이를 해결하고자 다양한 데이터에 대해 표준화된 파이프라인을 찾을 수 있도록 nnUNet이 만들어지게 되었다.

---

<sup>2</sup> 기울기의 과거 변화를 어느정도 유지함으로써 경로의 효율성을 주는 모멘텀의 값과 최신의 정보를 과거보다 크게 반영하는 RMSProp의 값을 모두 사용하는 방식.

nnUNet의 hyper-parameter에는 Fixed configurations, Rule-Based configurations, Empirical configuration가 있다.

(1) Fixed configuration : learning rate, optimizer, loss function, epoch와 minibatch의 개수 등 고정되어 있는 parameter를 포함하고 있다.

(2) Rule-Based configuration : 영상데이터의 종류에 따라 어떻게 normalization할 것인가, resize를 어떻게 할 것인가, segmentation을 하기 위해서는 patch size가 중요한데 이를 늘리고 Network가 몇번 pooling할 것인가 등을 포함하고 있다.

(3) Empirical configuration : nnUNet은 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.

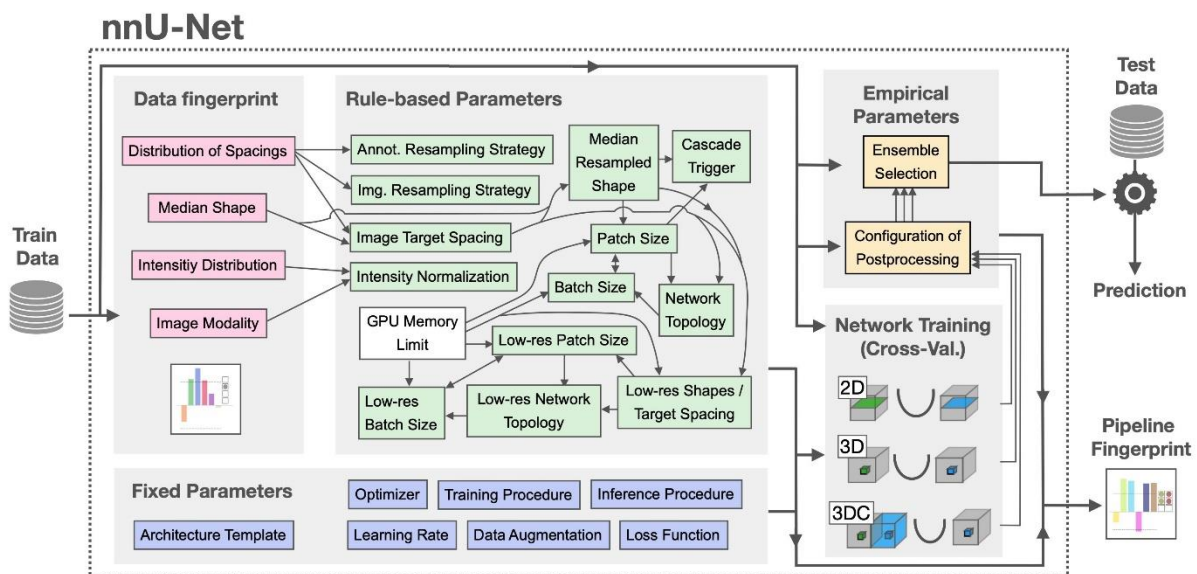


그림 2. nnUNet

### 3. 갱신된 과제 추진 계획

앞으로 해결해나가야하는 과제는 3D Slicer에 학습된 모델을 통해 얻어낸 mask파일과 segmentation하려는 영상데이터를 입력하여 bounding box를 이용하여 segmentation하는 것이다. 최종적으로 이를 프로그램으로 만들어 내는 것을 목표로 하고 있다.

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

#### 4. 구성원별 진척도

이름	진척도
이주승	학습 모델 개발 및 최적화 학습 모델 테스트 데이터 전처리 및 시각화
전도현	학습 모델 개발 및 최적화 예측 결과 도출 및 시각화 데이터 전처리 모듈 수정
김병관	데이터 전처리 학습 모델 테스트 학습 모델 최적화 및 오류 수정
공통	필요한 지식 습득 보고서 작성 및 검토 발표 및 시연 준비

## 5. 보고 시점까지의 과제 수행 내용 및 중간 결과

### 5.1. 모델 구현

먼저 가상환경 내에 nnUNet Directory를 생성하고 nnUNet을 설치한다. dataset\_conversion Directory 내에 Task018\_AbdominalOrganSegmentation.py 코드를 수정하여 dataset이 저장될 Directory의 이름, 위치, data들의 이름 등등을 설정한다. 이후 해당 파일을 실행하여 nnUNet\_raw\_data Directory내에 Task018\_AbdominalOrganSegmentation Directory 생성하고 아래와 같은 구조를 만든다.

```
Task018_AbdominalOrganSegmentation/  
├── dataset.json  
├── imagesTr  
├── imagesTs # optional  
└── labelsTr
```

**imagesTr**은 training image들을 포함하고 있고, nnUNet은 이 데이터를 사용하여 pipeline configuration, training with cross-validation를 수행한다.

**imagesTs**는 test image들을 포함하고 있고, nnUNet이 해당 데이터를 사용하지 않는다.

**labelsTr**은 training image들의 segmentation maps를 포함한다.

**dataset.json**은 dataset의 metadata를 포함한다.

```
nnUNetv2_convert_MSD_dataset -i Task018_AbdominalOrganSegmentation -
```

MSD<sup>3</sup> dataset을 nnUNet 형식으로 변환하여 기존에 있던 018의 dataset에 overwrite, Dataset018\_AbdominalOrganSegmentation Directory를 생성한다.

```
nnUNetv2_plan_and_preprocess -d 018 -c 3d_fullres -np 8
```

지정된 데이터셋 018을 3D network(3d\_fullres)를 이용하여 전처리하고, 3d\_fullres를 사용하기 위해 default인 8개의 process를 사용하였다.

nnUNet\_preprocessed Directory내의 Dataset018\_AbdominalOrganSegmentation Directory가 또 생성된다. 해당 Directory에 nnUNetPlans.json파일을 수정하여 원하는 batch size 설정한다.

---

<sup>3</sup> Medical Decathlon Challenge는 의료 영상 분석을 위해 사용되는 대규모 데이터셋의 집합이다. 이 데이터셋은 다양한 의료 영상 분할 문제를 다루기 위해 수집되었다.

## 5.2. 모델 학습 및 예측

모델 학습 및 예측에 앞서 개발 환경에 대해 설명하겠다. 개발 언어로는 Python을 채택하였고, 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.

```
nnUNetv2_train Dataset018_AbdominalOrganSegmentation 3d_fullres 0 1 2 3 4 -num_gpus 5
```

모델을 학습시키기 위해 nnUNetv2를 이용하여 361개의 영상데이터를 학습시키는 명령어를 사용하였다. dataset인 Dataset018\_AbdominalOrganSegmentation을 학습시킬 것이고 학습에 사용할 network를 3D data에 대해 full-resolution으로 처리하는 3d\_fullres로 지정한다. 더 정확한 학습을 위해 k-fold방식을 채택하기로 하여 총 5개의 fold로 학습을 진행하였다. - num\_gpus 5를 통해 5개의 GPU를 병렬로 활용하여 학습 속도를 높였다.

```
CUDA_VISIBLE_DEVICES=5 nnUNetv2_predict -i imagesTs/ -o pred_nnUNetall/ -d 018 -c 3d_fullres -f 0 1 2 3 4 -num_parts 2 -part_id 0 --continue_prediction
CUDA_VISIBLE_DEVICES=6 nnUNetv2_predict -i imagesTs/ -o pred_nnUNetall/ -d 018 -c 3d_fullres -f 0 1 2 3 4 -num_parts 2 -part_id 1 --continue_prediction
```

첫번째 명령어의 경우 5번 GPU를 사용하여 미리 학습된 모델로 imagesTs내의 test 데이터를 predict를 수행하고 predict된 결과는 pred\_nnUNetall Directory에 저장한다. 식별자가 018인 dataset을 5개의 fold로 나누어 학습된 모델을 통해 predict한다. -num\_part를 2로 설정하여 test 데이터를 두 개의 part로 나누어 0~100의 test 데이터 중 짝수번째 데이터에 대해 predict함을 의미한다.

두번째 명령어의 경우 6번 GPU를 사용하여 미리 학습된 모델로 imagesTs내의 test 데이터를 predict를 수행하고 predict된 결과는 pred\_nnUNetall Directory에 저장한다. 식별자가 018인 dataset을 5개의 fold로 나누어 학습된 모델을 통해 predict한다. -num\_part를 2로 설정하여 test 데이터를 두 개의 part로 나누어 0~100의 test 데이터 중 홀수번째 데이터에 대해 predict함을 의미한다.

위에서 언급하였듯이 nnUNet의 loss function은 Dice&cross-entropy이고 optimizer로는 Adaptive Momentum Estimation를 사용하였다. batch size를 2와 5로 설정하고 hold out방식을 사용하여 loss를 측정하였다. 아래의 그림3는 batch size를 2로 설정하였을때 epoch에 따른 loss, time, learning rate의 변화를 보여준다.



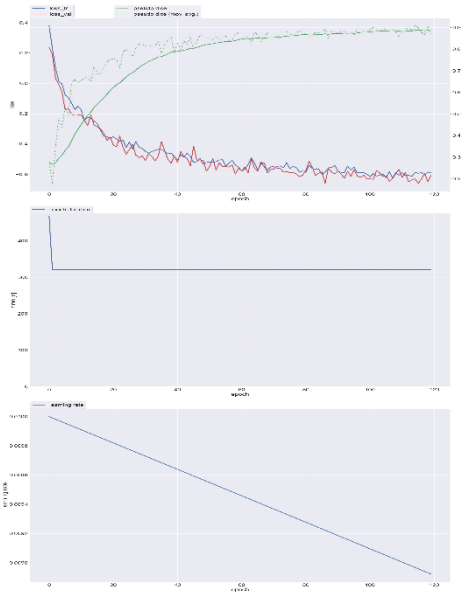


그림 3. batch size = 2 / loss변화량

아래의 그림 4은 batch size를 5로 설정하였을 때 epoch에 따른 loss, time, learing rate의 변화를 보여준다.

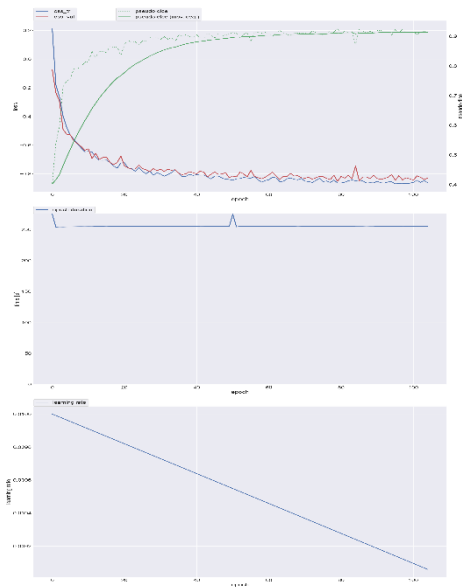


그림 4. batch size = 5 / loss변화량

batch size를 2로 설정했을때는 loss가 -0.6으로 수렴하는 경향을 보였으나, batch size를 5로 설정했을때는 loss가 더 낮아진 -0.8으로 수렴하였다. 또한, overfitting도 발생하지 않은 것을 확인하였다.

### 5.3. 모델 평가 지표

DSC(dice similarity coefficient)와 NSD(normalized Surface Dice)는 영상분할 평가를 위해 사용되는 평가 지표이다. DSC는 dice지수라고도 불리며, segmentation 모델의 예측결과와 실제 정답(ground truth) 사이의 유사성을 측정하는 지표이다. 주로 이진분할(binary segmentation)에서 사용된다. 0과 1 사이의 값을 가지며, 1에 가까울수록 모델의 성능이 높다.

NSD는 다중분할에서 사용되는 평가지표로, 다중 클래스의 DSC값을 평균화하여 얻는 값이다. Segmentation 평가지표로 일반적인 딥러닝 평가지표인 accuracy, recall, precision를 사용하지 않는 이유는 영상분할은 픽셀 단위로 이미지를 세분화하여 각 픽셀을 해당하는 클래스에 할당하는 작업이다. 클래스 불균형이란 특정 클래스의 픽셀수가 다른 클래스에 비해 월등히 많거나 적을 수 있다는 말이다. 예를들어 종양이나 병변과 같은 특정클래스의 영역은 적기 때문에 무시되는 문제가 발생할 수 있다. 이는 accuracy와 같은 평가지표를 사용하면 모델의 성능을 왜곡하게 된다. DSC, NSD는 클래스간 불균형을 고려하여 모델의 분할 성능을 정확하게 평가하는데 도움이 된다.

Test 이미지 파일 100개를 학습된 모델에 넣어 4개의 장기에 대한 DSC, NSD를 측정하였다.

Name	Liver_DSC	kidney_DSC	Spleen_DSC	Pancreas_DSC	Liver_NSD	kidney_NSD	Spleen_NSD	Pancreas_NSD
AVG	0.9719	0.9255	0.9355	0.8473	0.8639	0.9062	0.9287	0.9197

### 5.4. 실제 적용

학습된 모델을 통해 도출된 mask파일과 모델에 입력으로 들어간 영상데이터를 3D Slicer에 함께 입력으로 넣는다. 영상 데이터를 기반으로 예측된 label파일을 mask파일로 설정하면 각 축에 대한 단면이미지와 이를 합친 3D로 데이터를 시각화할 수 있다. 아래의 그림4가 실제로 3D Slicer를 통해 시각화 한 것이다.

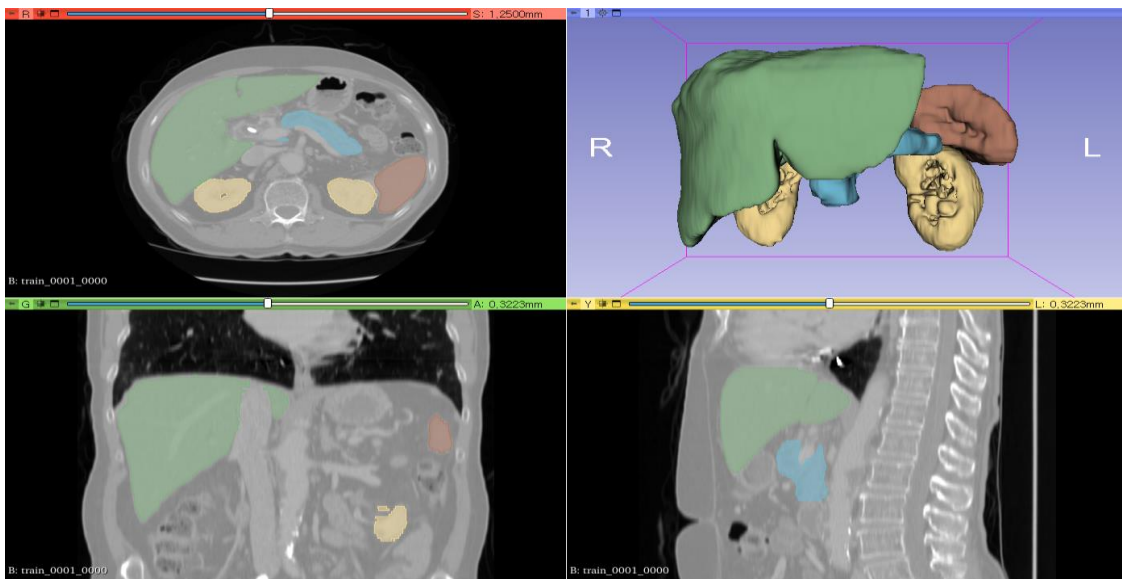


그림 5. 3D 시각화

## 6. GUI 예상안 및 시나리오

목표 : 사용자는 해당 프로그램으로 CT 파일을 넣어 4가지 장기를 따로 Segment 하여 원하는 장기를 3D도면으로 시각화 된 파일을 볼 수 있다.

시나리오

1. 사용자는 프로그램에 CT 폴더 경로를 설정한다.
2. 화면에는 기존의 3dSlicer처럼 4칸으로 구분되어 있다.
3. 프로그램 하단에 위치한 Segment 버튼을 누르면 해당 CT 파일의 4가지 장기를 Segment를 진행한다.
4. 상단 3개의 이미지는 CT파일의 단면을 서로 다른 축을 기준으로 시각화한다.
5. Segmentation된 3D 모델은 중단의 큰 화면에 출력된다. 각 장기는 Bounding Box를 통해 시각화된다.
6. 사용자가 보고 싶은 장기의 Check Box를 클릭하면, 해당 장기의 3D모델을 on/off 할 수 있다.

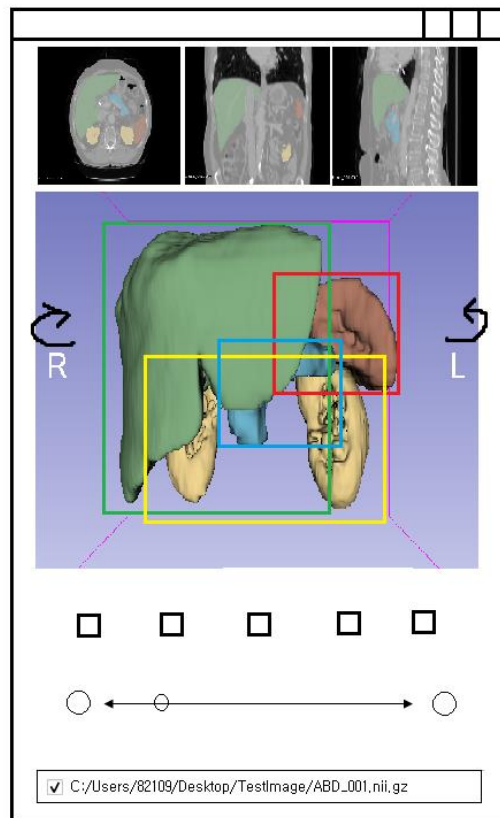


그림 6. GUI 예상안

## 7. 참고 자료

- [1] AbdomenCT-1K: Is Abdominal Organ Segmentation A Solved Problem?(<https://arxiv.org/abs/2010.14808>)
- [2] nnU-Net: Self-adapting Framework for U-Net-Based Medical Image Segmentation (<https://arxiv.org/abs/1809.10486>)
- [3] <https://github.com/MIC-DKFZ/nnUNet>