

# Semantic Segmentation Wrap-up Report

CV-03

김승기 김준영 신우진 전형우 천지은

## 1. 프로젝트 개요

뼈는 우리 몸의 구조와 기능에 중요한 영향을 미치기 때문에, 정확한 뼈 분할은 의료 진단 및 치료 계획을 개발하는 데 필수적입니다.

Bone segmentation은 인공지능 분야에서 중요한 응용 분야 중 하나로, 특히, 딥러닝 기술을 이용한 뼈 segmentation은 많은 연구가 이루어지고 있으며, 다양한 목적으로 도움을 줄 수 있습니다.

- 1. 질병 진단의 목적으로 뼈의 형태나 위치가 변형되거나 부러지거나 골절 등이 있을 경우, 그 부위에서 발생하는 문제를 정확하게 파악하여 적절한 치료를 시행할 수 있습니다.
- 2. 수술 계획을 세우는데 도움이 됩니다. 의사들은 뼈 구조를 분석하여 어떤 종류의 수술이 필요한지, 어떤 종류의 재료가 사용될 수 있는지 등을 결정할 수 있습니다.
- 3. 의료장비 제작에 필요한 정보를 제공합니다. 예를 들어, 인공 관절이나 치아 임플란트를 제작할 때 뼈 구조를 분석하여 적절한 크기와 모양을 결정할 수 있습니다.
- 4. 의료 교육에서도 활용될 수 있습니다. 의사들은 병태 및 부상에 대한 이해를 높이고 수술 계획을 개발하는 데 필요한 기술을 연습할 수 있습니다.

## 2. 프로젝트 팀 구성 및 역할

팀원	역할
전체	EDA, Online augmentation, Experiment
김승기	Offline augmentation, TTA
김준영	Accelerator, Baseline code
신우진	Edge detection metric
전형우	학습 속도 개선
천지은	Offline augmentation, Ensemble

### 3. 프로젝트 수행 절차



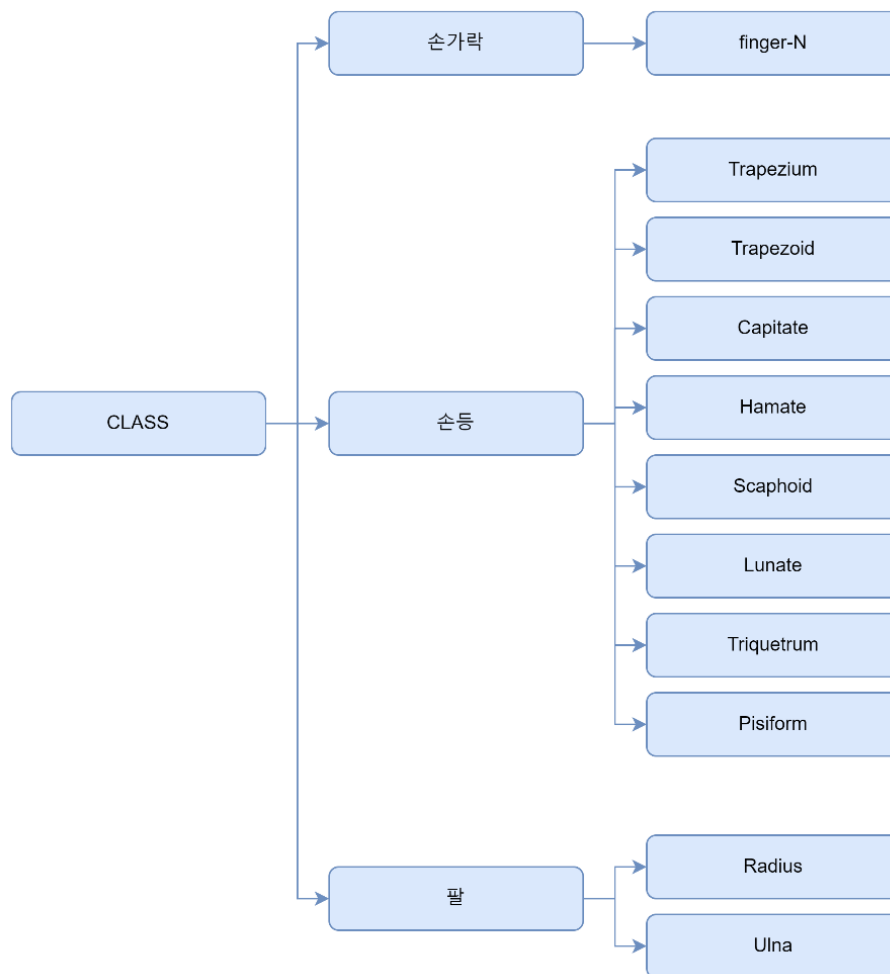
1. Seed 실험
  - 지난 OCR 대회에서 seed 에 따른 모델의 성능 차이가 심했기 때문에 이번 대회에서도 seed 의 중요성이 있는지 파악하기 위해 각자 다른 seed 를 설정해 베이스라인 코드를 실행하고 성능을 평가했습니다.
2. 모델 실험
  - 기본 PyTorch 모델에서 제공하는 FCN, Deeplab 의 각각 다른 backbone 을 통해 실험을 진행해 모델에 따른 성능의 차이가 있는 지 확인하고, SMP 라이브러리의 모델 역시 사용해 성능을 비교했습니다.
3. Augmentation 실험
  - 다양한 augmentation 을 적용한 결과를 시각화 해 어떤 augmentation 이 좋은 성능을 보일 지 사진을 보며 판단하고, offline 과 online augmentation 역시 수행하며 각 augmentation 에 따른 성능을 비교했습니다.
4. Loss 실험
  - 기존에 주어진 BCEWithLogitLoss 외에도 focal loss, dice loss, IOU loss 등과 각 loss 를 결합한 combined loss 등에 대해 성능 비교 실험을 진행했습니다.
5. Optimzer 실험
  - 기존의 Adam 외에도 AdamW, AdamP 등의 optimizer 과 lr scheduling 에 따른 성능 비교를 진행했습니다.

## 4. 프로젝트 수행 결과

### 4-1. EDA

#### 4-1-1. Dataset 구성

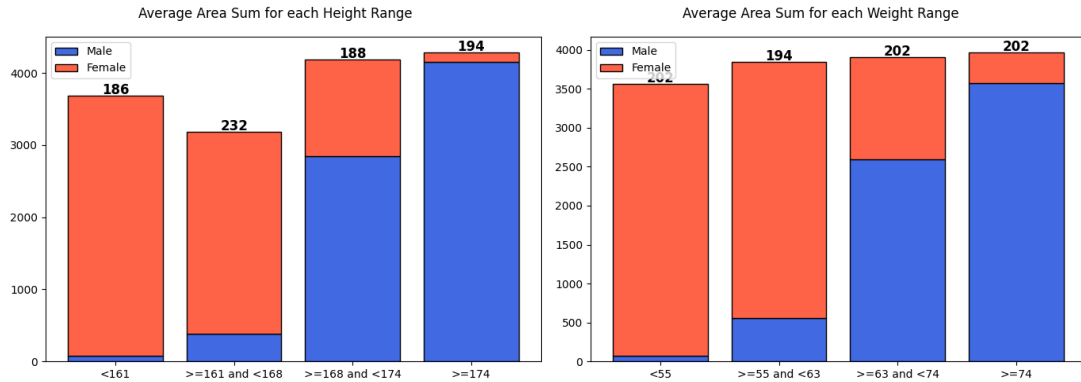
이미지의 크기는 2048 x 2048 x 3 으로 구성되어 있으며 양 손을 촬영했기 때문에 사람 별로 두 장의 이미지가 존재하며, 이는 폴더로 구분되어 있습니다. 그리고 train 800, test 300 장의 총 1100 장의 이미지가 있으며 각 이미지 내의 클래스는 다음과 같이 구성되어 있습니다.



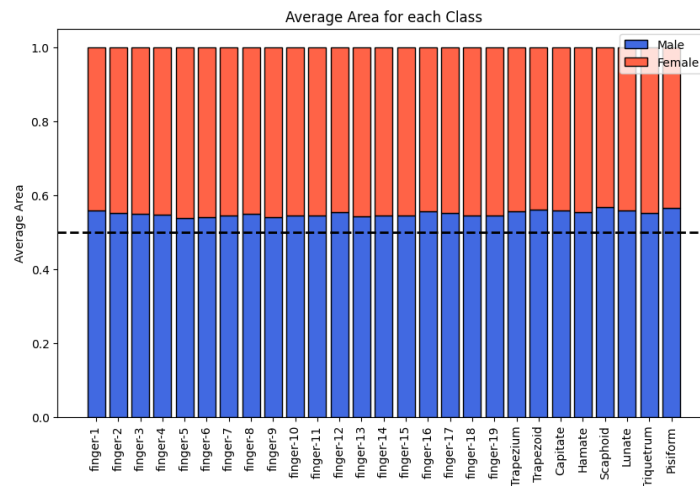
예측해야 하는 클래스는 'finger-1', 'finger-2', 'finger-3', 'finger-4', 'finger-5', 'finger-6', 'finger-7', 'finger-8', 'finger-9', 'finger-10', 'finger-11', 'finger-12', 'finger-13', 'finger-14', 'finger-15', 'finger-16', 'finger-17', 'finger-18', 'finger-19', 'Trapezium', 'Trapezoid', 'Capitate', 'Hamate', 'Scaphoid', 'Lunate', 'Triquetrum', 'Pisiform', 'Radius', 'Ulna' 의 29 개 클래스입니다.

## 4-1-2. Meta Information

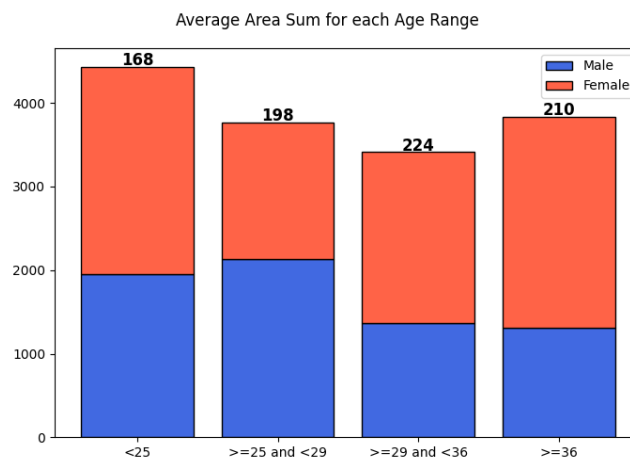
주어진 데이터에는 이미지 dataset 외에 사람의 meta 정보도 같이 주어져 있습니다. 이 정보는 나이 / 성별 / 체중(몸무게) / 키(신장)로 이루어져 있으며 성별에 따른 체중/키 분포는 다음과 같으며 당연히도 남자의 키와 체중이 여자에 비해 높은 경향을 보였습니다.



각 클래스에 따른 성별의 분포는 모든 클래스에서 거의 비슷하지만, 남자가 조금 많은 경향을 보였습니다.



각 나이에서의 성별 비율 역시도 비슷한 경향을 보였으며, <25와 ≥36에서는 여자의 비율이 조금 더 많은 모습을 보였습니다



### 4-1-3. Dataset 분석

1. 같은 사람 내에서도 팔 길이의 차이 때문에 양 손의 객체 영역이 다른 경우가 있었습니다.
2. 각 클래스의 면적에서는 finger-16, trapezoid, pisiform 이 평균적으로 가장 낮은 면적을 가져 모델이 해당 부분을 잡아내는 것에 어려움을 겪을 것으로 예상되었습니다
3. Ground Truth에서 같은 좌표 내에도 두 가지 이상의 class가 overlap된 경우가 있었습니다.
4. 대부분 이미지와 다르게 반지, 네일아트, 손목 임플란트, 흰 손가락 등 적은 양의 이상치들이 발견되었습니다.

### 4-1-4. Inference 분석

Validation set에 대해서 예측을 잘 하지 못하는 케이스에 대해 파악한 결과입니다.

finger-1	: 0.9371
finger-2	: 0.9604
finger-3	: 0.9666
finger-4	: 0.9312
finger-5	: 0.9500
finger-6	: 0.9685
finger-7	: 0.9683
finger-8	: 0.9380
finger-9	: 0.9575
finger-10	: 0.9698
finger-11	: 0.9641
finger-12	: 0.9326
finger-13	: 0.9534
finger-14	: 0.9671
finger-15	: 0.9618
finger-16	: 0.9079
finger-17	: 0.9333
finger-18	: 0.9604
finger-19	: 0.9653
Trapezium	: 0.9264
Trapezoid	: 0.8923
Capitate	: 0.9487
Hamate	: 0.9300
Scaphoid	: 0.9434
Lunate	: 0.9351
Triquetrum	: 0.9159
Pisiform	: 0.8687
Radius	: 0.9837
Ulna	: 0.9747

역시나 작은 객체인 finger-16, Trapezoid, Pisiform 에 대한 dice 스코어가 낮은 것을 볼 수 있었습니다.

반지 등 이상치 데이터에서 가장 큰 이상치인 반지를 착용한 이미지에 대해 예측하고 있는 결과를 확인한 결과 다음과 같이 Ground Truth 는 반지까지 손가락의 영역으로 잡고 있는 것에 비해 예측 결과는 반지는 물론 그 주변의 손가락도 잘 판별하지 못하는 것으로 보였습니다.

Finger bone 의 경계에 대한 예측을 확인한 결과 파란색이 정답이며 다른 색들은 각각 RED(클래스 라벨 오류) : 21%, YELLOW(과소예측) : 45%, GREEN(과대예측) : 33%의 비율로 오류가 발생하고 있었습니다. 이를 통해 경계 주변에 대한 미세한 예측을 잘 하지 못하는 것을 확인했습니다.

## 4-2. 문제점과 가설 설정

### 4-2-1. 문제점

1. 면적이 작은 class 에 대해 예측을 잘 하지 못합니다.
2. Finger bone 의 경계에 대한 예측을 제대로 하지 못합니다.
3. 반지와 같은 outlier 가 존재합니다.

### 4-2-2. 가설 설정

1. 면적이 작은 class 에 대해 예측을 잘 하지 못합니다.
  - 이미지를 512, 1024 로 줄이는 과정에서 생기는 문제라고 생각해, 면적이 작은 것과 큰 것들만 뽑아서 fine-tuning 을 진행하는 것으로 해결할 수 있다고 생각했습니다.
2. Finger bone 의 경계에 대한 예측을 제대로 하지 못합니다.
  - Augmentation
    - 경계가 뚜렷해지도록 하는 augmentation (Emboss, Sharpen 등)을 실행해서 경계를 더 잘 잡을 수 있게 하려고 했습니다.
  - 이미지 사이즈
    - 학습 이미지와 mask 를 줄이면서 생기는 정보 손실 때문에 발생하거나 inference 과정에서 bilinear interpolation 을 진행하며 생기는 오차로 예상되어 이를 해결하려고 했습니다.
3. 반지와 같은 outlier 가 존재합니다.
  - 일반적으로 반지와 같은 것들은 X-ray 를 찍을 때 착용하지 않은 상태에서 촬영하게 되는데, 해당 이미지들은 완벽한 outlier 로 판단되어 이런 데이터는 삭제를 하고 학습과 검증을 해서 영향을 파악하려고 했습니다.

## 4-3. Augmentation

### 4-3-1. Online Augmentation

- Gray-scale 의 이미지기 때문에 밝기 관련 augmentation 이 잘 동작할 것으로 예상되어, RandomBrightnessContrast 를 사용했습니다.
- 경계를 잘 잡는 것이 중요하다고 생각되어 경계의 강조가 효과적일 것으로 예상되는 Emboss, Sharpen, CLAHE 등의 augmentation 을 사용했습니다.

- 왼손 / 오른손에 대한 쌍으로 이미지가 존재하기에 HorizontalFlip 의 성능이 좋을 것으로 예상되어 사용했습니다.
- 실제로 Emboss, HorizontalFlip 등의 성능이 좋게 나왔습니다.

#### 4-3-2. Offline Augmentation

위 그림과 같이 기존 이미지 4 장을 붙여 새로운 하나의 이미지로 사용해 기존 train data 에 추가했습니다.

Online 에서 사용하던 HorizontalFlip 을 Offline 으로 자체 적용시켜 학습 자체에 사용할 데이터를 추가하기 위한 Offline HorizontalFlip 을 진행했습니다.

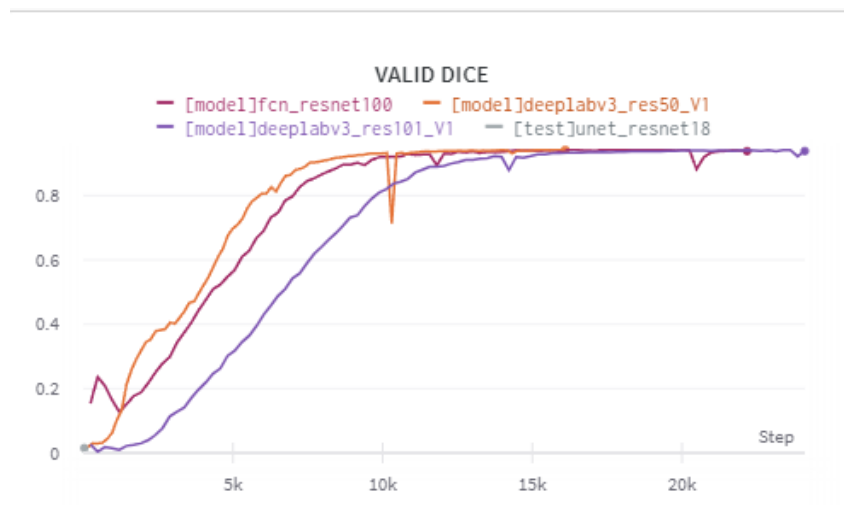
Rotate 역시 Offline 으로 사용해 더욱 학습 데이터의 양을 늘려서 학습에 사용했습니다.

데이터를 증진해서 학습하는 효과와 transfer learning 을 기대했고, 데이터의 양이 많아진 결과 성능의 향상을 보였습니다.

### 5. Model

#### 5-1. torchvision.models

FCN, Deeplabv3, Unet등의 각 backbone을 ResNet18, ResNet50, ResNet101 등으로 바꾸며 다양한 모델을 사용한 결과를 비교했습니다.



학습 초반에는 차이가 있었지만 후반으로 가며 valid dice가 수렴한 이후에는 큰 차이가 없었으며, 안정적으로 학습이 되어 수렴하는 모습을 보였습니다.

#### 5-2. MMSegmentation

MMSegmentation 모델을 사용하며 이미지 사이즈를 늘리고, Torchvision에 구현되어 있지 않은 SegFormer 모델을 사용하는 방향으로 라이브러리를 사용했습니다.

multi-scale feature를 뽑으며 빠른 실험을 진행했지만, 0.97의 스코어를 넘기는 것은 힘들었습니다.

### 5-3. SMP (Segmentation models with pretrained backbones)

다른 라이브러리인 SMP를 사용해 encoder 부분에 다양한 모델로 pre-trained가 되어 있는 모델을 사용하고자 했고 medical segmentation 분야에서 좋은 성능을 보이는 Unet과 Unet++가 구현되어 있어 해당 모델들을 EfficientNet-b5를 통해 encoder부분이 pre-trained 되어 있는 버전으로 사용했습니다.

Offline augmentation을 적용한 데이터를 추가로 사용하고, AMP 및 collate\_fn을 변경한 코드 등으로 학습 속도와 배치 사이즈, num\_worker 문제를 개선하고 실험을 진행한 결과 EfficientNet-b5를 pre-trained로 사용한 Unet++ 모델의 실험에서 단일 모델 최초로 0.97의 점수를 넘겼습니다.

## 6. Ensemble

### 6-1. Voting Ensemble






이미지의 각 픽셀별로 과반수 이상 true 라고 예측한 값으로 앙상블하는 방법입니다. Test dataset 에 대한 dice 값이 높은 모델 3 개로 앙상블 했을 때, Public dice 는 0.9709, Private 은 0.9711 으로 단일 모델의 성능보다 높게 나왔습니다.

### 6-2. Probability Mean Ensemble





모델이 예측한 확률을 0.3 초과인 값만 따로 저장해두고, 이를 통해 이미지의 각 픽셀별 평균이 0.5 초과인 값으로 앙상블하는 방법입니다. Test dataset 에 대한 Dice 값이 높은 모델 4 개로 앙상블 했을 때, Public dice 는 0.9706, Private 은 0.9710 으로 단일 모델의 성능보다 높게 나왔습니다.

## 7. 결과

Public

순위	팀 이름	팀 멤버	dice ↕
16 (-)	CV_03조	    	0.9709

Private

순위	팀 이름	팀 멤버	dice ↕
16 (-)	CV_03조	    	0.9711



## 팀 회고

---

### 잘했던 점

- 꾸준한 커뮤니케이션과 협업을 통해서 프로젝트를 진행했습니다.
- Git flow를 이용한 branch 관리, PR을 통한 코드 리뷰, commit convention을 잘 지켰습니다.
- 전부 적극적으로 커뮤니케이션에 참여했습니다.
- 실험 결과 분석이 잘되었습니다.

### 시도했으나 잘 되지 않았던 점

- 모델 탐색이 제대로 진행되지 않은 점이 있습니다.
- Github를 잘 활용하지 못했습니다.

### 아쉬웠던 점

- 이전 경험을 토대로 생각해봤을 때 transformer 계열의 모델들이 CNN모델보다 더 좋은 성능을 보였습니다. 그래서 Unet의 backbone을 변경하며 실험을 진행하는 것보다 먼저 MMSegmentation을 사용해 SegFormer를 실험했습니다. 그러나 예상과 달리 transformer보다 Unet에서 성능이 더 잘 나왔습니다. 하지만 시간 부족으로 인해 Unet의 backbone을 변경해가며 실험을 진행하지 못한 아쉬움이 남는 것 같습니다.
- 처음 실험하기 전에 모델에 대해 전체적인 성능을 파악하는 실험을 진행했었는데 이때 Unet을 실험하지 못한 부분이 아쉬움으로 남습니다.
- Model의 특징을 공부하고 해당 프로젝트에 필요한 model들을 해당 프로젝트에 적용하려고 했으나 다음과 같은 문제들을 만나면서 계획적으로 되지 않았던 점이 아쉬움이 남습니다. 또한 마지막 주차에 모델들을 돌아가며 실험을 진행하면서 생기는 시간 부족을 겪었습니다. 그리고 모델의 특징을 가설을 세웠던 것들과 매칭을 시키지 못한 점이 아쉽습니다.
- 할당 받은 메모리가 해당 프로젝트를 진행하기에 충분하지 않아 많은 실험을 하지 못한 점이 아쉬운 것 같습니다.
- 여러 library를 돌아다니다 보니 실험이 중복된 점이 아쉽습니다.

### **프로젝트를 통해 배운 점 또는 시사점**

- MMSegmentation 사용법을 익힐 수 있었습니다.
- 의료분야는 도메인 지식이 필수적으로 필요한 것 같습니다.

### **다음 프로젝트에서 시도할 것**

- 가설 검증을 진행하는데 있어서 기존 연구, 강의 내용을 제대로 이용하지 못한 것 같습니다.
- 하루에 할 수 있는 실험, 가설 설정, 가설 검증을 늘리고 싶습니다.
- 실험 자동화를 제대로 하면 좋을 것 같다고 생각을 했습니다.

## 개인 회고 : 김승기

---

### 1. 나의 학습 목표

이번 대회 제 학습 목표는 충분한 EDA를 통해서 가설을 설정하고 검증하는 방식으로 실험을 진행하고, 이 과정에서 대회에서 성능을 끌어올릴 수 있는 여러가지 방법론을 적용해보는 것이었습니다.

### 2. 나는 내 학습 목표 달성을 위해 무엇을 어떻게 했는가?

EDA를 통해서 현재 데이터셋의 특징을 파악하고, 현재의 task와 연관 지어서 가설을 설정했습니다.

대회의 평가 방법인 mean dice coefficient를 높이기 위해서는 작고 겹치는 클래스들을 잘 예측해야 한다고 판단하고, 그것에 대비한 augmentation을 구현하거나 실험했습니다.

### 3. 나는 어떤 방식으로 모델을 개선했는가?

RICAP augmentation 기법을 변형해서, 이미지 4장을 mosaic과 같은 형태로 이어 붙인 이미지로 만드는 offline augmentation을 구현했습니다.

회전한 손 이미지들을 다시 원상태로 돌리는 회전을 적용하고, 회전하지 않은 손 이미지들은 회전 시키는 방향으로 offline augmentation을 적용했습니다.

### 4. 전과 비교하여 새롭게 시도한 변화는 무엇이고, 어떤 효과가 있었는가?

실험 후 다시 validation 결과에 대한 분석을 반복하는 형태로 실험을 진행했었고, 실험의 방향성을 중간에 조정하는 것에 많은 도움이 되었습니다.

### 5. 마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?

- Dataloader에 대한 이해가 아직 많이 부족하다는 것을 느꼈습니다.
- Baseline code를 최적화 시키지 못했습니다.
- 아직 segmentation 모델들에 대해 깊이 이해하지 않고 있다는 것을 느꼈습니다.

### 6. 한계/교훈을 바탕으로 다음 프로젝트에서 시도해볼 것은 무엇인가?

조금 더 이론을 복습하고 들고 관련 논문들을 더 깊이 파고 들어서 진행을 하려고 노력할 것입니다.

- SOTA 모델이나 연구를 진행하는 프로젝트에 맞춰서 적용해보고 싶습니다.
- Dataloader 관련한 내용을 더 공부해보고, 코드에 적용해보고 싶습니다.

### 1. 나는 내 학습목표 달성을 위해 무엇을 어떻게 했는가?

EDA를 통해 문제점을 발견하고자 했고 해당 문제점들을 하나씩 해결해 가고자 했습니다. 또한 하나의 문제를 해결하고 또 다른 문제가 발견되었을 때 해당 문제를 또 가지고 있는 데이터가 없는지 EDA를 반복해서 수행했습니다.

Semantic segmentation에서 사용되는 모델, loss들의 특징들을 해당 프로젝트에 적용하고자 모델, loss들의 특징들을 정리했습니다.

### 2. 나는 어떤 방식으로 모델을 개선했는가?

Semantic segmentation에 어떤 모델들이 있고 모델들이 서로 어떤 문제를 해결하기 위해 만들어졌는지 공부했습니다. FCN부터 시작해서 deeplab, U-Net, SegFormer 모델에 대해 단점과 장점 등을 알 수 있었습니다.

Semantic segmentation을 수행하면 겪을 수 있는 문제인 메모리 부족 문제를 어떻게 해결할 수 있을지 고민할 수 있었던 시간이었습니다. 이미지의 개수가 적은 것도 문제였지만 이미지 크기를 일정 크기 이상으로 키울 때 batch size가 작은 것이 문제임을 발견하고 AMP와 gradient accumulation을 적용했습니다.

이미지의 밝기 정보가 서로 다른 것을 확인하고 밝기와 관련된 augmentation을 적용해서 모델의 성능을 조금 올릴 수 있었습니다.

### 3. 내가 한 행동의 결과로 어떤 지점을 달성하고, 어떤 깨달음을 얻었는가?

이번 프로젝트를 진행하면서 가설 설정과 가설 검증이 중요한 것을 많이 깨달았던 것 같습니다. 지금까지 3개의 프로젝트들을 진행했지만 가설 설정과 가설 검증이 제대로 이루어진 프로젝트는 처음인 것 같습니다. 물론 가설 설정과 가설 검증이 잘못 되었을 수도 있지만 알고 있는 지식을 가지고 가설을 설정하고 가설을 검증하면서 실험이 제대로 이뤄지고 있는지 또는 문제점이 더 발견되지는 않았는지 확실하게 파악할 수 있었습니다. 이렇게 진행을 하다 보니 이전 프로젝트들보다 더 뿌듯함이 느껴지는 프로젝트인 것 같습니다.

### 4. 전과 비교하여 새롭게 시도한 변화는 무엇이고, 어떤 효과가 있었는가?

EDA를 통해 가설을 설정하고 가설을 검증하는 것을 제대로 해 본 첫 프로젝트인 것 같습니다. 이전에도 가설 설정과 가설 검증을 하려고 했었지만 어딘가 모르게 부족한 부분이 많았던 것 같습니다. 그러나 이번 프로젝트에서는 팀에서 설정한 평가지표를 통해 문제를 분석하고 문제를 어떻게 해결해 갈지 같이 의논하고 해결했던 것들이 눈에 띄게 있었던 것 같습니다.

## 5. 마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?

Object detection 프로젝트를 수행할 때 모델을 먼저 탐색해보지 않은 것이 굉장한 아쉬움으로 남았었습니다. 그래서 이번 프로젝트에서는 모델을 먼저 탐색해 봤는데 성능 차이가 많이 나지 않아 가장 가벼운 모델로 프로젝트를 진행했었습니다. 그러나 데이터의 수를 늘리고 loss를 바꾸고 augmentation을 적용하고 나니 모델들의 성능 차이가 이전보다 늘어난 것을 뒤늦게 알 수 있었습니다. 모델의 한계가 정해져 있다는 것을 알 수 있었던 것 같습니다.

또한 AMP와 gradient accumulation의 특징들만 알고 어떻게 작동하는지 공부를 하지 않고 일단 사용했던 것이 가설을 설정하고 가설을 검증하는데 있어서 부족함이 드러났던 것 같습니다.

## 6. 한계/교훈을 바탕으로 다음 프로젝트에서 시도해볼 것은 무엇인가?

지금까지 프로젝트들은 정해진 주제, 정해진 평가지표가 있는 대회와 특징들을 가지고 있습니다. 그러나 이후 진행할 최종 프로젝트는 주제도 팀에서 정하고 평가지표도 팀에서 정한 것을 사용하게 될 것입니다. 그렇게 되면 가설 설정, 가설 검증, 모델 선택과 같이 이전에 잘해왔던 것들 앞으로 고쳐나가야 할 것들이 잘 지켜지지 않을 수도 있다는 생각을 합니다. 그래서 최종 프로젝트를 시작할 때 조금 더 구체적으로 계획을 세우고 어떻게 하면 소통을 더 많이 할 수 있을지 팀원과 이야기 해보고자 합니다.

최종 프로젝트에서는 지금까지 했던 것들을 최대한 쏟아부어서 적어도 이전에 했던 실수들을 하지 않도록 할 것입니다.

AMP와 gradient accumulation에 대해 공부를 해보고자 합니다.

이번 프로젝트에서 모델 경량화도 정말 중요한 것을 느낀 것 같습니다. 아무리 AMP, gradient accumulation을 적용해도 모델의 사이즈가 너무 크면 AMP, gradient accumulation을 적용하는 이유가 사라지는 것 같습니다. 그래서 이후 경량화에 대해서도 공부를 해볼까 합니다.

### 1. 나는 내 학습목표 달성을 위해 무엇을 어떻게 했는가?

생소한 segmentation 분야에서 프로젝트를 잘 진행하기 위해 강의를 들으며 segmentation 모델에 대해 공부했고, 다양한 라이브러리와 구현되어 있는 모델에 대해 코드를 이해하며 프로젝트를 진행했습니다.

### 2. 나는 어떤 방식으로 모델을 개선했는가?

이미지의 크기에 따른 성능 차이가 있는 것을 깨달았고, 이미지 사이즈에 따른 실험을 진행했습니다.

강의를 다시 들으며 Segmentation 모델들의 차이에 대해 공부를 하고, 그러던 중 Unet 계열의 모델이 medical segmentation의 적은 데이터에 대해 특화된 모델임을 알고 Unet과 이를 개선한 모델들에 대해 실험을 진행했고 단일 모델에서 가장 좋은 성능을 얻을 수 있었습니다.

### 3. 내가 한 행동의 결과로 어떤 지점을 달성하고, 어떤 깨달음을 얻었는가?

Unet 계열의 모델에 다양한 기법을 진행해 단일 모델에서 가장 좋은 DICE를 얻을 수 있었습니다. 이를 통해 무작정 실험하기보다는 task에 대한 이해를 기반으로 어떤 모델을 쓰면 좋을 지 미리 고민을 하고 선정 후 진행하는 것이 효율적인 것을 깨달았습니다.

학습 속도에서 batch size를 더 많이 넣기 위해 메모리 이슈를 해결하기 위한 방법을 사용했고, 이후 num\_worker와 batch size를 늘리는 것에 성공해 학습 속도가 크게 개선되었습니다. 이를 통해 학습 환경에 대한 개선을 빠르게 시도하면 이후 실험에 유리한 것을 깨달았습니다.

### 4. 전과 비교하여 새롭게 시도한 변화는 무엇이고, 어떤 효과가 있었는가?

다양한 라이브러리와 torchvision과 mmlibrary 외에도 다른 모델을 가지고 있는 외부 깃허브에서 라이브러리를 끌어오려고 시도했고, 해당 라이브러리에 구현되어 있는 Unet의 모델이 좋은 성능을 보였습니다.

Shared memory가 Linux 환경 자체에서 제한을 두고 있어 이를 개선하기 위한 방법을 찾았고, 다른 캠퍼분의 팁 덕분에 이를 개선할 수 있었고 학습 속도를 크게 늘릴 수 있었습니다.

### 5. 마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?

초반에 성능을 올리기 위해 모델별 실험을 진행하려고 했으나 torchvisions.model에서 간단한 비교만 한 후 넘어가 Unet 계열의 모델을 사용할 생각을 하지 못했고, mmseg에 대한 이해가 부

족해 초반에 시간을 조금 허비한 것 같아 아쉬웠습니다.

일정 이상 성능이 올라가지 않는 한계를 모델로 개선할 생각보다 다양한 기법과 하이퍼 파라미터 튜닝 등으로 개선해보려고 했던 점이 아쉬웠습니다.

## **6. 한계/교훈을 바탕으로 다음 프로젝트에서 시도해볼 것은 무엇인가?**

마주한 task에 대한 이해를 먼저 선행하고, 어떤 모델이 가장 좋은 효율을 보일 지 파악하고 이를 바탕으로 논리적으로 접근을 먼저 해야할 것 같습니다.

개인적인 실험을 보다 체계화해서 혼자 진행중인 실험이라도 다양한 변수들을 통제하며 공유하기 좋은 환경을 만들어봐야겠습니다.



## 개인 회고 : 천지은

---

### 1. 나는 내 학습목표 달성을 위해 무엇을 어떻게 했는가?

저의 학습목표는 팀원들과 지속적인 소통과 계획적인 실험을 통해 협업하는 것이 주된 목표였고, 실험은 EDA를 통해 먼저 데이터셋을 파악하고, 가설을 세워 논리적인 실험을 진행하는 것이었습니다. 이를 위해 실험 내용 및 결과에 대한 저의 의견을 적극적으로 공유했고, 실험 결과를 분석하는데도 시간과 노력을 들이도록 하였습니다.

### 2. 모델 개선을 위해 어떤 방식을 시도해봤는가?

주어진 데이터셋이 한 사람의 양손을 각각 X-Ray로 촬영한 것으로 구성되어 있고, 왼손과 오른손이 정확히 대칭적이지 않다는 것을 파악했습니다. 따라서 Offline Augmentation을 활용하여 Horizontal Flip을 통해 Upsampling을 진행해 성능을 개선시켰습니다.

Voting 방식 앙상블과 확률평균값 방식 앙상블을 구현하여 기존 단일 모델보다 앙상블을 통해 성능을 개선시켰습니다.

### 3. 마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?

Semantic Segmentation Task를 주어진 환경에서 수행하는데 큰 모델과 대용량 데이터를 처리하는데 한계를 많이 느꼈습니다. 원본 이미지의 크기가 2048x2048x3으로 크고, 한 이미지에 존재하는 GT Annotation의 개수도 많아 Offline Augmentation을 수행하는 시간이 오래 걸렸습니다.

앙상블을 진행할 때, 각 픽셀별 확률을 모두 비교하고 싶었지만 메모리 한계와 시간적 제약으로 인해 어려움이 있었습니다. 확률값을 함께 저장하면 파일 용량이 증가하고 이에 따른 로딩 시간도 증가하는 문제가 있었습니다. 앙상블 시간을 줄이기 위해 방법을 탐색하였고, 대용량 데이터 처리를 위해 CSV 형식 대신 Parquet 형식을 사용하여 파일 용량과 로딩 시간을 약 5배로 감소시켰습니다.

의료 도메인 지식이 부족하여 Inference 결과를 시각화하여 확인해봐도 제대로 예측을 한 것이 맞는지 모호했고, GT Annotation도 일부 각기 다르게 라벨링된 부분이 있어 라벨 클렌징을 하려 해도 무엇이 옳은 정답인지 알 수 없던 것이 한계로 크게 다가왔었습니다.

중반부터 동일한 모델로 일부만 변경해가며 실험을 진행했었는데, 더 이상 큰 성능 향상을 보이지 않을 때 모델의 한계를 느꼈었지만 시간 내에 더 나은 모델을 탐색하거나 최적의 모델을 찾지 못한 것이 아쉽습니다.

### 4. 한계/교훈을 바탕으로 다음 프로젝트에서 시도해볼 것은 무엇인가?

다음 프로젝트에서는 다양한 모델 아키텍처를 탐색하고 실험하기 위해 관련 논문과 자료를 조사하겠으며, 실험 결과를 철저히 분석하여 모델의 성능과 한계를 명확히 파악하고 인사이트를 도출하도록 노력할 것입니다. 또한, 메모리 한계에 부딪힐 경우를 대비하여 처리 속도를 향상시킬 수 있는 방법도 고려하도록 하겠습니다.

## 개인 회고 : 신우진

---

### 1. 나는 내 학습목표 달성을 위해 무엇을 어떻게 했는가?

이번 대회에의 학습 목표는 Inference를 자세하게 분석하여 문제의 원인을 분석하고 이를 기반으로 체계적인 실험을 이어나가는 것이었습니다. 학습 목표를 달성하기 위해 우선 첫번째로 Inference를 자세하게 분석할 수 있는 코드를 작성했습니다. 모델이 과소 예측하는지, 과대 예측을 하는지, 라벨을 잘 못 표기하는지, Interpolation 전과 후 오류가 어떻게 다른지 등등 여러 방면으로 분석하였고 찾은 문제를 정의하고 해결하려고 노력 했습니다.

### 2. 나는 어떤 방식으로 모델을 개선했는가?

Inference 분석 결과, 모델이 고해상도로 이미지를 제대로 복구하지 못하는 문제를 발견했습니다. 그래서 모델의 이미지 복구를 최소화 해주기 위해 512 \* 512 size 학습에서 1024 \* 1024 size로 변경하였습니다. 이로써 모델이 경계에서 만드는 오류를 줄일 수 있었습니다.

### 3. 내가 한 행동의 결과로 어떤 지점을 달성하고, 어떤 깨달음을 얻었는가?

체계적인 실험을 이어가기 위해서는 Inference 분석을 자세하게 하는 것이 매우 중요하다는 것을 알았습니다.

### 4. 전과 비교하여 새롭게 시도한 변화는 무엇이고, 어떤 효과가 있었는가?

이전 대회에서는 문제를 제대로 정의하지 않고 실험을 진행했었는데 이번 대회에서는 풀고자 하는 문제를 정확히 정의하는 것에 시간을 많이 사용했습니다. 이로써 조금 더 방향성 있는 실험이 가능해졌고 대회를 통해 얻는 경험과 지식도 풍부해 진 것 같습니다.

### 5. 마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?

Inference 분석을 위한 코드를 짤 때 다음번에 분석할 수 도 있는 부분들을 염두하면서 코드를 작성했으면 좋았을 텐데 그러지 못해 비슷한 코드를 여러번 작성했습니다. 코드의 재활용성을 염두하며 작성하지 못한 점 때문에 많은 시간을 낭비한 것이 첫번째로 아쉬운 점입니다.

Inference에서 내가 발견한 문제를 기존 연구에서 더 면밀히 찾으려 하지 않은 점이 아쉬운 점입니다. 문제 해결을 저의 직관에 의존하고 좀 더 체계적인 기존 연구를 참고하여 파악하지 않은 것이 아쉽습니다.

### 6. 한계/교훈을 바탕으로 다음 프로젝트에서 시도해볼 것은 무엇인가?

코드의 범용성을 생각하고 어떤 목적으로 코드를 작성하는지 우선 스스로 notion에 정리한 다

음에 code를 작성하려고 합니다.

대회와 관련된 기존 연구를 먼저 면밀히 살펴보는 연습을 해야 할 것 같습니다. 기존 연구에서 같은 문제를 어떻게 풀었는지 더 면밀하게 살펴보는 과정이 있어야 문제를 더 깊이 풀어볼 수 있을 것 같습니다.