BoostCamp AI Tech

Project3: SemanticSeg

Wrap-up Report

CV-04

Member: 김세연, 김채리, 안지현, 김상유, 김태욱, 김윤서

목차

1. 프로젝트 개요

1.1 프로젝트 주제

Hand Bone Image Segmentation은 손 X-ray 이미지에서 29개의 뼈를 segmentation하는 프로젝트이다. 해당 대회에서는 정확한 bone segmentation을 통해 의료 진단 및 치료 계획 개발에 도움을 주는 딥러닝 모델 개발을 목표로 한다. 성능은 semantic segmentation task에서 대표적으로 사용되는 dice coefficient로 평가한다.

1.2 데이터 구성

- 입력으로 사용되는 X-ray 데이터는 train 이미지 800장, test 이미지 288장으로 이루어져 있으며, 양손을 촬영하여 사람 별로 두 장의 이미지가 존재함.

- 29개의 클래스는 크게 손가락, 손등, 팔로 구성되어 있으며, 손등의 Trapezium과 Trapezoid, Triquetrum과 Pisiform은 겹쳐져 있어 예측이 어렵다는 특징이 있음.

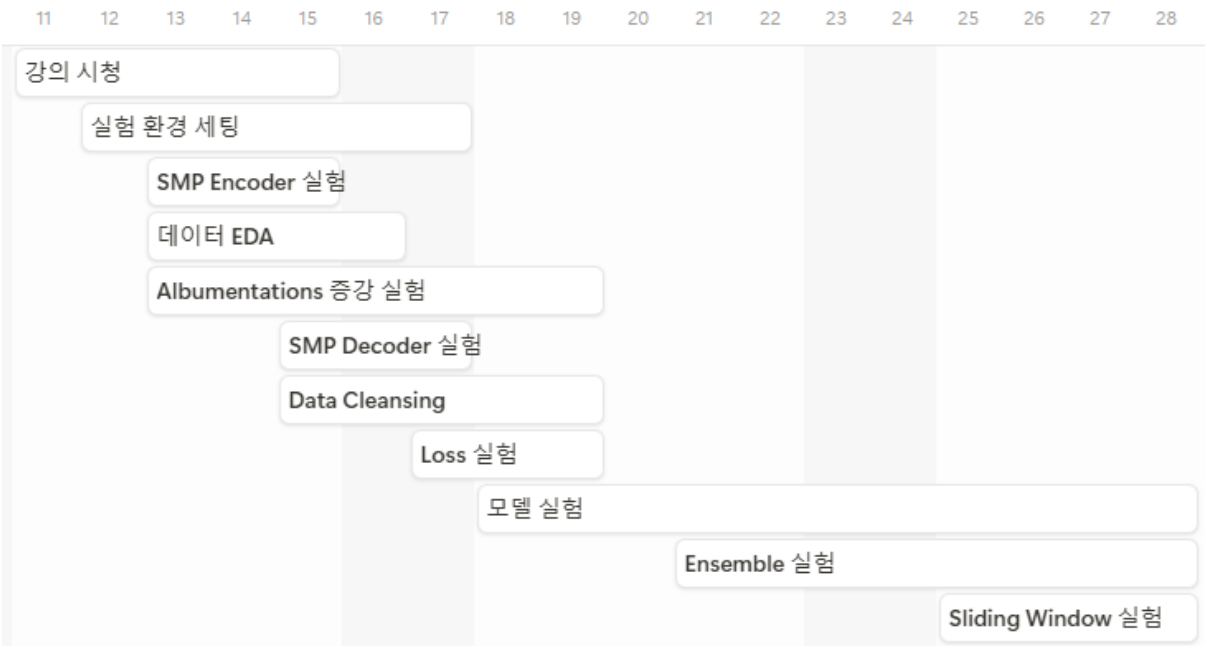
- 모든 이미지는 2048x2048 크기를 가짐

2. 프로젝트 팀 구성 및 역할

이름	역할
김세연	실험 파이프라인 구조화 및 설계, 모델 실험
김채리	실험 결과 시각화, Augmentation, Loss, 모델 실험
안지현	Wandb 실험 관리, Confusion Matrix 등의 에러 분석 시각화, 모델 실험
김상유	실험 속도 개선, 데이터 전처리, 증강, Loss, 모델 실험
김태욱	SMP Encoder 모델 실험, Ensemble
김윤서	데이터 EDA, Data Cleansing, 모델 실험

3. 프로젝트 수행절차 및 방법

3.1 프로젝트 일정



3.2 실험 관리

실험 구현 및 공유, 그리고 재현성 등 실험 관리의 효율을 높이기 위해 YAML 파일을 통해 실험 파라미터 및 구조를 변경할 수 있게 하였다. DATA, TRAIN, MODEL, INFERENCE, WANDB로 나뉘어져 있으며, 밑에 후술할 항목은 핵심적인 항목 위주로 설명하였다.

LOSS, OPTIMIZER, SCHEDULER

LOSS의 경우, torch.nn 라이브러리의 loss와 의료 분야에서 많이 쓰이는 monai 라이브러리의 loss를 사용할 수 있도록 구성하였으며, LOSS, OPTIMIZER, SCHEDULER 모두, 해당하는 파라미터의 이름을 인자로 하여 PARAM 항목 밑에 구성하여 유동적으로 사용할 수 있게 구현하였다.

TRANSFORMS

데이터 증강의 경우, albumentations 라이브러리를 활용했으며, 각 증강 기법은 '-'를 통해 리스트화하여 다중 증강을 가능하도록 구성하였다. 또한, 해당하는 증강 파라미터의 이름을 인자로 하여 PARAM 항목 밑에 구성하여 유동적으로 사용할 수 있게 구현하였다.

MODEL

```python
class BaseModel(nn.Module, ABC):
    def __init__(self):
        super().__init__()

    @abstractmethod
    def forward(self, x):
        pass

    @abstractmethod
    def get_model(self):
        pass
```

커스텀 모델 구현 시, 필수 메서드를 구현하지 않은 실수를 방지하고, 모델의 공통적인 구조 및 인터페이스를 가질 수 있도록 abstractmethod를 통해 강제화하였다. 하위 클래스 모델은 forward 메서드와 Finetuning에 활용될 수 있는 get_model 메서드를 구현하도록 설계되며, 이를 통해 새로운 모델 추가 및 기존 모델 수정 등의 작업 시 기존의 Train 코드의 추가적인 작업 없이 통합적으로 동작하고, 호환될 수 있게 하였다.

WANDB

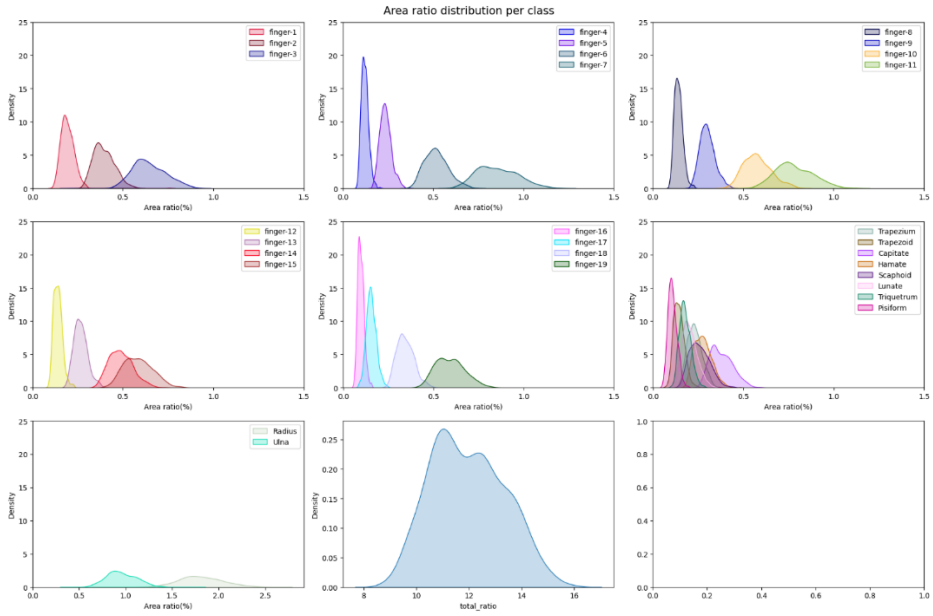
- Metric : Train Loss/Validation Loss 및 Validation 수행 시의 Average Dice 점수, 각 클래스별 Dice 점수 기록
- Config : 실험 시 사용한 Config 파일의 Train 세팅(LOSS, OPTIMIZER, SCHEDULER 등의 하이퍼 파리미터)
- Watch : 모델의 파라미터(가중치 및 편향), 그래이디언트를 히스토그램으로 기록
- Artifact : 학습 중 저장되는 모델 파일을 버전별로 기록, 모델 에러 분석 시 혼동행렬 및 경계 오류율과 클래스별 Dice 점수를 시각화하여 플롯 형태로 기록.

4. 프로젝트 수행 결과

4.1 데이터 분석

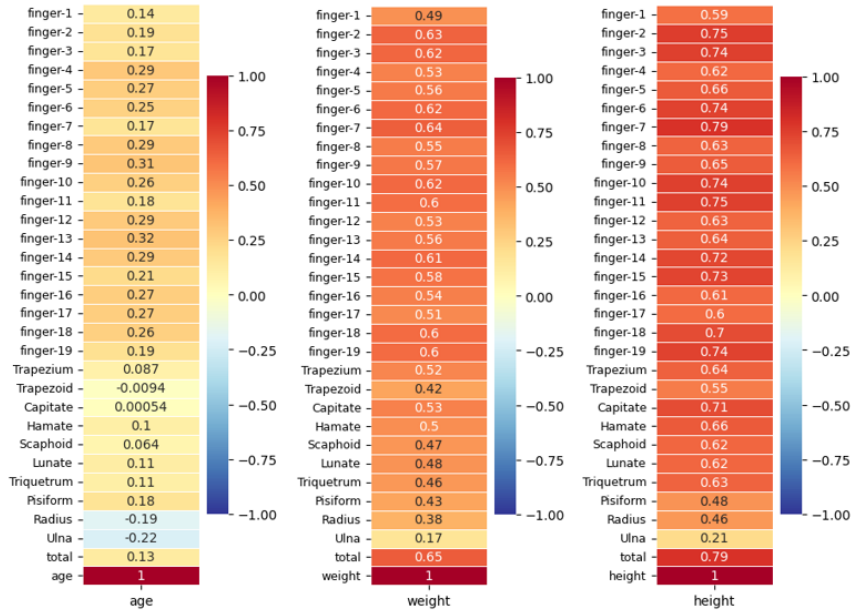
EDA는 크게 3가지 항목(클래스 별 면적 분포, 메타데이터와의 상관관계, 양손 간 유사도)을 분석하고 이 과정에서 발견한 이상치 데이터를 직접 확인하는 순서로 진행했다. 또한, 전체 분석은 이미지에서 각 클래스가 차지하는 면적을 기준으로 하였으며, 직관적인 이해를 위해 비율로 계산하여 사용하였다.

A. 클래스 별 면적 분포



팔 뼈에 해당하는 Radius와 Ulna의 크기 및 분산이 가장 큰 것을 관찰하였다. 이는 팔 뼈가 다른 뼈와 달리 일부분만 포함되기 때문에 엑스레이를 찍는 위치에 따라 변동성이 큰 것으로 유추할 수 있다.

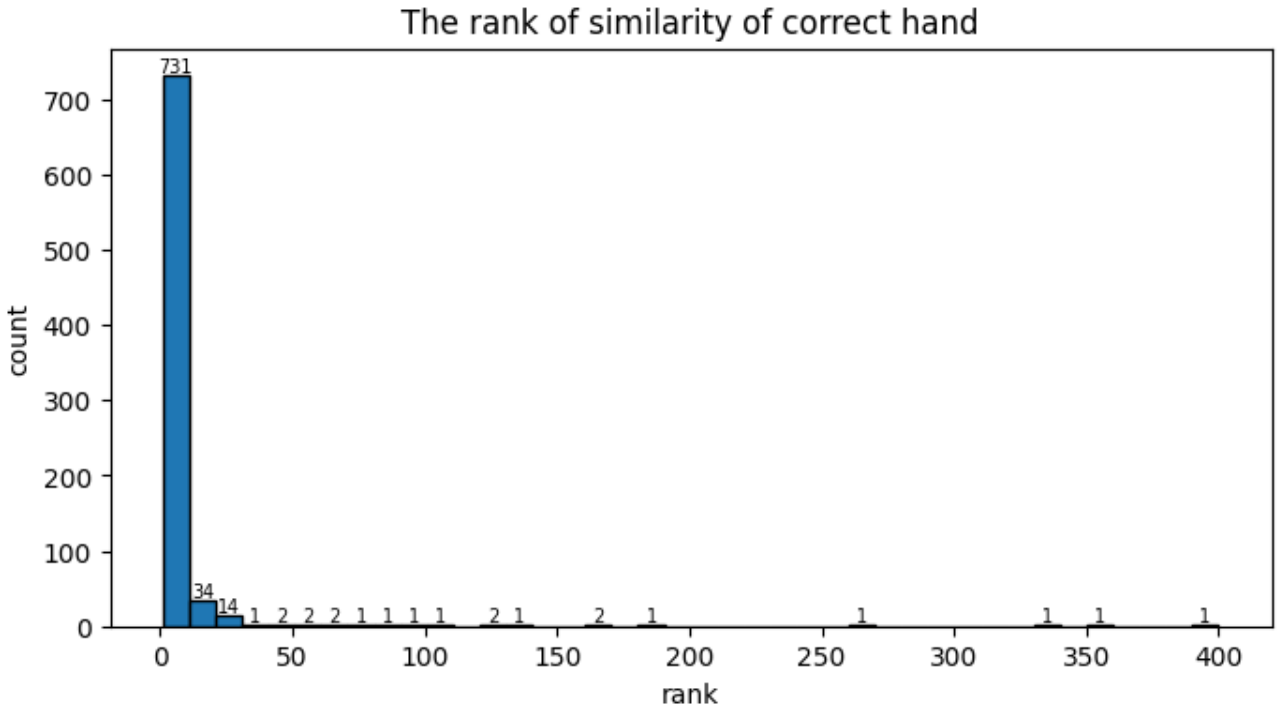
B. Meta data 와의 상관관계



나이에서는 유의미한 상관관계가 나타나지 않았지만, 몸무게와 신장은 대부분의 클래스에서 뼈의 면적과 양의 상관관계를 보였다. 특히 신장은 일부 클래스에서 0.7 이상의 강한 양의 상관관계를 보였는데, 이는 신장이 클수록 뼈의 면적이 넓어진다고 해석할 수 있다.

또한, 전체 뼈 면적이 남성은 평균 13.13%, 여성은 평균 10.97%로 성별에 따라서도 뼈의 면적에 차이가 존재함을 확인하였다.

C. 양손 간 유사도



그래프는 400개의 반대 손들과 cosine 유사도를 계산한 다음, 짝이 되는 손이 몇 번 째로 유사도가 높은지를 rank로 나타낸 그래프이다. A에서 관찰한 바와 같이 팔 뼈는 엑스레이를 찍는 위치에 영향을 많이 받기 때문에 팔 뼈는 제외하여 변수를 통제하였다. 800개의 손 중에 730개 이상이 10위 이내의 rank를 보였다는 점에서 양손 간 유사도가 높다고 판단하였다.

D. 데이터 확인

C에서 양손 간 유사도가 낮게 나온 case 11개를 시각화하여 분석해보았다. 여기서 클래스 라벨링이 잘못된 이미지, 반지를 낀 이미지, 손등 뼈가 팔 쪽에 라벨링된 이미지를 발견하였고, 한 손이 이러한 이상치를 가진 경우 유사도가 낮게 나타났음을 알 수 있었다.

Test 데이터에도 특이 케이스가 존재하는지 확인하는 과정에서 손목이 꺾인 이미지가 train에는 약 9.4%밖에 없는 데 비해 Test 데이터에도 특이 케이스가 존재하는지 확인하는 과정에서 손목이 꺾인 이미지가 train에는 약 9.4%밖에 없는 데에 비해, test에는 약 57.6%가 존재하는 것을 발견하였다. 회전되어 있는 뼈도 잘 예측할 수 있도록 이후 데이터 증강에 rotation을 추가하는 실험을 진행하였다.

4.2 Data Augmentation

A. Albumentation

가설 설정

본 데이터셋은 X-ray 데이터셋으로, 적절한 증강 기법을 적용함으로써 train, test dataset 간의 분포 차이를 줄이고, 보다 노이즈에 robust한 모델을 구축하고자 한다. 구체적으로 다음과 같은 증강 기법을 적용하고자 한다.

1) **CLAHE**: CLAHE는 히스토그램 평활화를 기반으로 이미지의 대비감을 높여 더욱 선명하게 만들어주는 기법이다. 이는 의료 영상 처리에 자주 사용되는 기법이다.
2) **Random Brightness Contrast**: 이미지의 밝기와 대비를 랜덤하게 조정함으로써 조명 변화가 있는 데이터에 robust한 모델 학습을 보장한다. 이는 밝기 차이로 이미지를 보여주는 x-ray 데이터에 적합한 augmentation으로 분석된다.
3) **Elastic Transform**: 이미지에 탄성변형을 적용하여 약간의 왜곡을 유발한다. 이를 통해 굴곡 있는 손 뼈 데이터에 적합한 증강을 부여함으로써 보다 다양한 데이터 학습이 가능하도록 한다.
4) **Blur**: 선명한 학습 데이터보다 흐리게 처리한 이미지로 학습함으로써 디테일이 감소된 데이터에도 정확한 segmentation을 수행하는 모델을 기대한다.
5) **Horizontal Flip**: 손 뼈 데이터의 특성상 좌우 반전된 형태를 지니고 있으므로 이를 적용한다.
6) **Random Rotation**: 학습데이터와는 달리 테스트 데이터에는 손이 꺾인 형태의 데이터가 50% 이상 발견되었다. 이에 테스트 데이터와 유사한 학습데이터로 모델 학습을 진행하기 위해 학습데이터를 랜덤하게 90도 회전하는 증강기법을 사용한다.

실험 결과

	Baseline	Test 1	Test 2	Test 3	Test 4	Test 5
증강 기법	X	CLAHE	rotate	Elastic transform + brightness	Elastic transform + brightness + CLAHE	Elastic transform + brightness + blur + horizontal flip
정확도	0.9650	0.9650	0.9627	0.9607	0.9604	0.9544

실험 결과, 유의미한 증강 기법은 확인하지 못하였다. 이는 기존 학습 데이터와 테스트 데이터 간의 차이가 거의 없으며, 기존 데이터만으로도 충분한 학습이 이루어지고 있음을 확인할 수 있다.

B. Sliding Window

가설 설정

기존 접근 방식에서는 이미지 해상도를 낮춘 뒤 모델 학습 및 예측을 수행하고, 결과를 다시 원본 해상도로 복원하는 과정에서 Bilinear Interpolation 기법이 사용된다. 이러한 과정이 모델의 예측 정확도를 저하시킬 가능성이 있다고 판단했다.

해결 방안

1. 2048x2048 해상도 유지

   - UNet (EfficientNet-b0) 모델 기준 **0.9642 -> 0.9499**로 성능이 떨어졌다.

   - UNet++ (EfficientNet-b8), DuckNet의 경우 메모리 문제로 학습이 불가능하다.

2. 슬라이딩 윈도우

- 2048x2048 이미지를 **1024x1024 윈도우 크기**로 나누고, **512 stride**를 적용하여 9개의 패치로 분할한 뒤 학습을 진행하였다.

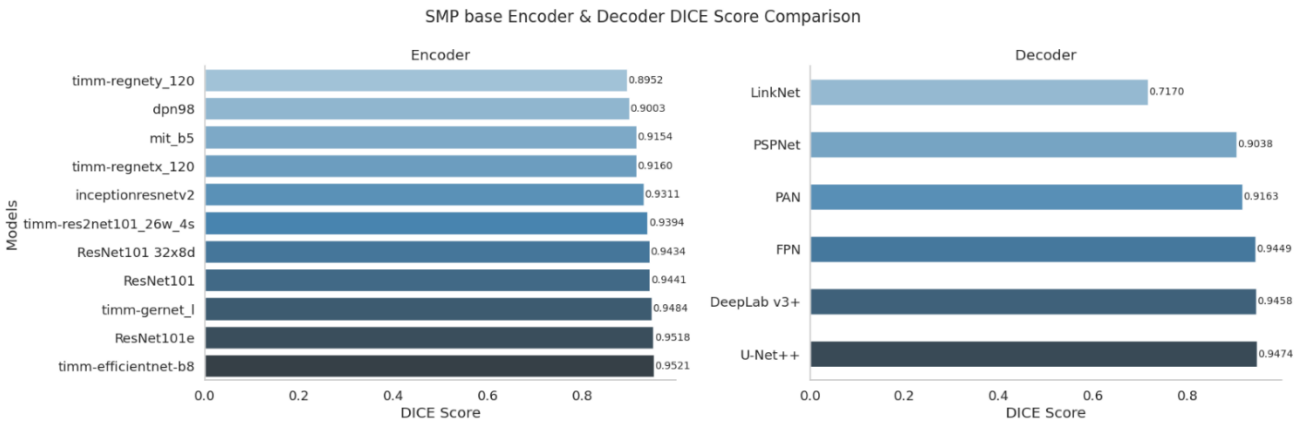
- 예측 시 겹치는 영역의 값은 평균으로 결과를 생성하였다.

- UNet++ (EfficientNet-b8) 모델 기준 **0.9718 -> 0.9730**으로 성능이 개선되었다.

결론

슬라이딩 윈도우 기법으로 고해상도 이미지 처리에서 발생하는 메모리 문제를 해결할 수 있으며 Bilinear Interpolation 과정에서 발생하는 예측 정확도 저하를 방지할 수 있다.

4.3 SMP Encoder, Decoder 성능 비교



SMP에 기본으로 내장되어 있는 Encoder와 Decoder의 성능을 비교하였다.

Encoder의 경우, resnet101, res2net101, regnetx, regnety, efficientnet-b8, Xception71 외 6개의 모델을 비교하였다. 그 중 efficientnet-b8 모델이 0.9520 수치의 높은 성능 결과를 가져왔으며, 해당 모델을 메인 실험 Encoder로 채택하였다.

Decoder의 경우, DeepLab v3+, FPN, LinkNet, PSPNet, U-Net++, PAN, MANet을 비교하였으며, MANet의 경우, 다른 모델과 공통된 하이퍼파라미터로 성능이 나오지 않아 제외하였다. Decoder 중 U-Net++가 0.9474로 가장 좋은 성능을 보였으며, 해당 모델을 메인 실험 Decoder로 채택하였다.

4.4 Extra Models

A. Duck-Net

가설 설정

DUCK-Net은 medical image segmentation 수행을 목적으로 만들어진 모델로 polyp segmentation에서 SOTA를 달성했다. 구조는 U-Net의 Encoder-Decoder 구조와 유사하나 크게 두 가지 차이점을 가지고 있다. 먼저, 6개의 변형된 convolutional block을 병렬적으로 연결한 custom block을 도입하였으며, Encoder의 각 resolution level에서 downsampling한 이미지를 더해주는 residual mechanism을 추가했다. DUCK-Net의 이러한 특성이 polyp의 위치를 찾고 경계를 디테일하게 예측하는 데에 기여한 것처럼 hand bone segmentation에서도 성능 개선에 효과적일 것이라 생각했다.

실험 결과

Metric	DUCKNet	Finetuning1	Finetuning2
Dice(public)	0.9436	0.97	0.9714

DUCKNet을 epoch30으로 학습시켰을 때 public 0.9436으로 U-Net++ 0.9438과 큰 차이를 보이지 않았다. 그러나 validation에서 평균 dice 점수가 지속적으로 상승하는 것을 보아 모델이 아직 충분히 수렴하지 않았다고 판단하여 finetuning을 추가로 진행하였다. 첫 번째 finetuning에서는 ExponentialLR scheduler를 사용하여 0.97로 상승하였으며, 두 번째 finetuning에서는 DiceFocal loss를 사용하여 0.9714까지 성능을 향상시켰다.

DUCKNet Backbone 연결

기존 SMP 백본을 사용할 수 없었던 DUCKNet에 백본을 연결하여 성능을 높이고자 하였다. 이를 위해 SMP로 Decoder를 구성하고, 이를 기존 DUCKNet Decoder의 Skip connection에 연결하기 위해 Conv2d를 통해 채널을 통일시켜 주었다. Encoder로 efficientnet-b0을 사용한 DUCKNet의 결과는 다

음과 같다.

Metric	DUCKNet	Finetuning1
Dice(public)	0.9485	0.9706

epoch30으로 학습시켰을 때 public 0.9485으로 0.05 정도의 성능 향상을 보였다. ExponentialLR scheduler을 통한 첫 번째 Finetuning은 0.9706으로 Vanilla-DUCKNet보다 약간의 성능 향상이 있었지만, 큰 차이는 보이지 않았다.

B. NNUnet

가설 설정

NNUNet은 medical image segmentation task의 성능을 극대화하기 위해 데이터셋의 특징을 자동으로 분석하여 최적화된 파이프라인을 구성하는 자동화 프레임워크이다. 이 프레임워크는 이미지 크기, voxel 크기, 클래스 비율, 대상 구조물의 특성, 이미지 강도 등의 데이터셋 특성을 분석하여 모델 아키텍처 설계, 이미지 전처리, 하이퍼파라미터 설정, 데이터 증강 기법, 후처리 방법, 그리고 학습 과정 등을 데이터에 맞게 자동으로 조정한다. 이러한 NNUNet을 통해 전문적인 의학 지식 없이 의료 데이터의 복잡한 특성을 반영하고, 학습 파이프라인을 최적화함으로써 더 높은 segmentation 성능을 얻을 수 있을 것이라고 생각하였다.

실험 결과

Metric	Fold 1	Fold 2	Fold 3
Dice(public)	0.9629	0.9628	0.9636

각 Fold 별로 1000 epoch을 실험하였으며, 기존의 SMP Base Encoder 및 Decoder, vanilla-DUCKNet과 비교했을 때, 약 0.1 ~ 0.2 정도의 성능 향상이 있었다. 하지만, 이는 패치 기반 학습과 같은 다양한 방법론, 더 많은 epoch 등이 적용된 결과로 생긴 성능이라고 판단되며, 실제로 타 모델에 Sliding window, Finetuning을 추가했을 때보다는 낮은 성능을 보였다. 이는 단순히 U-Net 모델 계열 모델의 한계 및 제공된 데이터의 단순함 (예를 들면 2D 데이터, 데이터 간 노이즈가 적음) 때문이라고 판단된다.

C. SwinUNETR

가설 설정

이전 실험인 EffiSegNet의 모델 구조는 SegNet Decoder에 Efficientnet을 Encoder로 사용한다.

- 이로 인해 이미지의 특징 추출이 효율적이며, 대체적으로 객체별 특징을 잘 포착함 또한 Decoder의 구조가 간단하기에 빠른 학습으로 다양한 실험이 가능했음.

- SegNet Decoder의 구조는 nearest neighbor 방식으로 모든 feature map을 한 번에 원본 크기로 upsampling하며 단순 덧셈으로 특징들을 결합함

- 계산 시 효율적이지만 세부적인 정보 손실로 인해 다양한 객체 크기를 가진 현재 task에는 다소 한계가 있음

이에 의료 딥러닝 라이브러리인 monai의 SwinUNETR 모델을 채택하여 객체의 세부적인 특징을 더 잘 포착하고자 하였다.

- Encoder는 Swin을 사용하여 self-attention을 활용해 전역적인 관계를 파악가능
- Decoder에서는 점진적으로 2배씩 해상도를 증가시키며 각 단계마다 skip connection으로 Encoder 특징들을 결합함

실험 결과

SwinUNETR은 768 사이즈의 이미지를 통해 기학습된 모델을 다시 1024 사이즈의 이미지, 그리고 2048 사이즈의 이미지를 사용하여 파인튜닝하였다.

Metric	Finetuning(1024)	Finetuning(2048), Sliding window
Dice	0.9694	0.9708

이전 실험인 EffiSegNet(0.9691)과 비교했을 때, 1024 사이즈 이미지로 학습한 모델은 제출 점수 기준으로 소폭(0.0003) 증가하였으며, 2048 사이즈의 이미지를 슬라이딩 윈도우 방식으로 학습한 모델의 경우는 1024 사이즈의 이미지를 사용한 경우보다도 좋아진 성능을 확인할 수 있었다.

D. SegFormer

가설 설정

SegFormer와 같은 Transformer 기반 모델은 전역적인 문맥 정보와 지역적인 특징을 효과적으로 포착할 수 있어 의료 영상 세분화에 적합하다. 특히 SegFormer는 다양한 스케일의 특징을 처리할 수 있어 손등 뼈와 같이 작은 스케일의 데이터 segmentation에도 robust한 성능을 보인다.

실험 결과

Metric	Fold 1
Dice	0.9714

실험 결과 0.9714로, finetuning전에도 다른 모델 평균보다 높은 성능을 확보하였다.

E. Upernet + SwinL

가설 설정

Swin Transformer 기반의 UperNet은 다중 스케일 정보를 처리하는 능력을 지니고 있어 복잡한 손등 뼈 구조를 지닌 본 데이터셋에 적합할 것으로 판단된다. 구체적으로, Swin transformer는 윈도우 기반의 모델로 고해상도 x-ray 이미지에서 중요 특징을 효과적으로 처리할 수 있을 것이다. 또한 UperNet의 다중 스케일 특징 통합 능력이 작은 세밀한 정보와 문맥적 정보를 통합적으로 처리하는 데에 효과적일 것이다.

실험 결과

Metric	Fold 1
Dice	0.9703

실험 결과 0.9703으로, Finetuning 이전에도 높은 성능을 보이는 것을 확인하였다.

4.5 Loss

가설 설정

모델 실험 결과, 특정 클래스를 잘 분류하지 못하는 것을 확인하였다. 또한 baseline code는 Binary classification loss를 적용하였으나, 본 대회의 평가 지표는 dice loss이다. 따라서 잘 못 맞추는 손등 뼈에 대해 더 높은 정확도를 확보하고, 평가 metric에 맞는 성능 향상 도모하고자 focal loss, dice loss 등을 추가로 실험하였다.

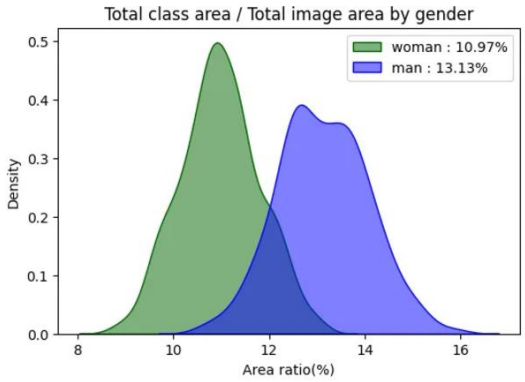
실험 결과

	실험1	**실험2**	**실험3**	**실험4**
Loss	BCE	Dice	DiceCE	DiceFocal
Dice Score	0.9630	0.9618	0.9634	**0.9650**

실험 결과, DiceFocal Loss가 가장 높은 dice score를 보이는 것을 확인하였다. 이는 잘 못 맞추는 손등 뼈에 대해 focal loss가 가중치를 부여하여 더 높은 성능을 확보하고, dice loss를 활용하여 본 대회의 평가 지표에 부합하는 학습이 이루어졌기 때문으로 분석된다.

4.6 적용한 방법론

A. Meta Data



성별에 따른 데이터 분포의 차이를 근거로, 남성과 여성 별 예측 모델을 별도로 생성하면 성능 향상이 있을 것으로 생각했다. **290개의 남성 이미지**와 **350개의 여성 이미지**를 사용하여 2개의 모델의 학습했다. 남성 학습 모델은 **0.9410**, 여성 학습 모델은 **0.9459**의 성능을 보였고, 각 성별에 맞는 이미지를 예측한 후 통합하여 제출한 결과 **0.9453**으로 각 모델이 학습한 성별을 특히 더 예측을 잘하는 경우는 나타나지 않았다.

B. Data Cleansing

K-fold를 이용해 5개의 모델로 각각 Validation Data를 예측한 후 AvgDice가 낮은 이미지를 노이즈로 간주하여 삭제하였다. **68장**의 데이터를 제거한 후 새롭게 학습한 결과 **0.9514 -> 0.9505**로 성능이 감소했다. K-fold 수행에 많은 시간이 소요된 데 비해 성능 개선 효과가 없어 해당 방법을 사용하지 않았다.

C. Pseudo Labeling

가장 성능이 좋은 모델을 활용해 Test Data 예측 결과를 Train Data로 활용했다. 리더보드 기준 **0.9685** 성능의 모델을 사용하여 **288장**의 Test Data를 학습데이터에 추가하여 UNet (Efficient b0) 모델을 새롭게 학습한 결과 **0.9607 -> 0.9583**로 성능이 감소했다. Test 이미지를 평가할 수단이 부족하여 모델의 잘못된 예측이 학습에 사용된 것이 성능 저하의 원인으로 분석되어, 이후 해당 방법은 사용하지 않았다.

D. Per-Class Thresholds

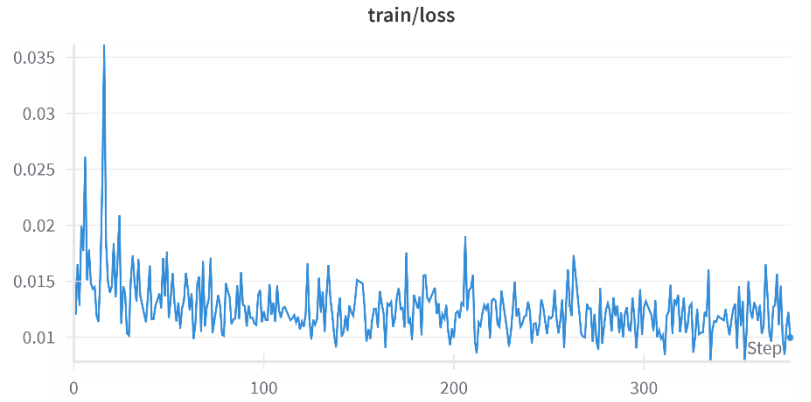
클래스별로 적절한 threshold가 존재할 것이라고 생각하여 Validation data를 기준으로 클래스별 threshold를 평가했다. 그 결과, finger 클래스는 높은 threshold에서, 손등 뼈 클래스는 낮은 threshold에서 성능이 좋은 것을 발견했다. 이후, Test data에서 **30epoch** 모델에 클래스별 threshold를 설정하여 예측 결과 **0.9612 -> 0.9361**로 성능이 향상됐다. 그러나 **80 epoch** 모델로 재실험한 결과

0.9723 -> 0.9724로 차이가 미미했다. 이는 적은 epoch 모델의 경우 학습이 충분하지 않아 예측 값이 중간 값에 머무르며 threshold 조정 효과가 컸던 반면 많은 epoch 모델의 경우 예측 값이 극단적으로 분포되어 효과가 크지 않았던 것으로 보인다.

E. CutMix / Mosaic

손뼈를 예측하는 과정에서 뼈의 위치보다 모양에 집중하여 segmentation이 가능한지 확인해고자 두 가지 증강 기법을 적용했다. **CutMix**는 두 이미지, **Mosaic**은 네 개의 이미지를 합쳐 사용하는 방식이다. 두 증강 기법으로 생성된 이미지는 경계가 부드럽게 이어지지 않고 손 뼈의 위치가 섞이게 된다. 학습 결과 **0.9602 -> 0.9395**로 성능이 크게 떨어졌다. 왼손과 오른손이 섞이는 등 다양한 변수가 학습을 방해한 것으로 보이며 뼈 모양만으로 segmentation을 수행하기에는 어려움이 있을 것으로 생각된다.

F. Stochastic Weight Averaging

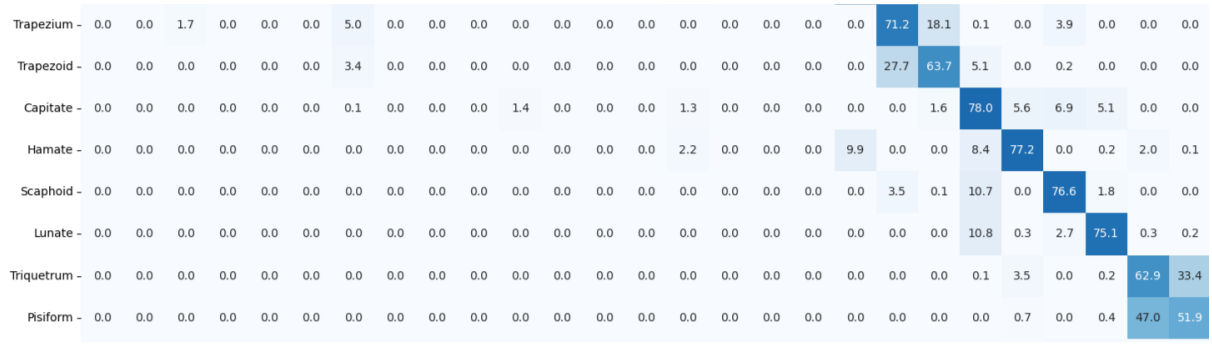


학습 과정에서 Loss가 일정하게 감소하지 않고 상승과 하락을 반복하며 다양한 Local Minima에 도달하는 경향이 관찰됐다. **Stochastic Weight Averaging(SWA)**을 이용하여 학습 과정에서 여러 시점의 가중치를 평균 내어 **Global optimal**에 도달하고자 했다. 실험결과 **0.9721 -> 0.9723**으로 성능이 소폭 향상됐다. SWA 기법은 기존 모델의 학습을 방해하지 않고 별도의 SWA 모델을 생성하기에 이후 모든 학습에서 SWA 기법을 사용하다.

G. Class Overlap preprocessing

가설 설정

주어진 대회 Class에서 Trapezium과 Trapezoid, Triquetrum과 Pisiform의 뼈가 겹쳐 모델의 전반적인 성능이 저하되었으며, NNUNet에서 Multi-Label을 지원하지 않아 겹치는 영역에 대한 전처리가 필요하였다. 이러한 겹치는 영역에 대해 별도의 클래스를 부여하여 학습하면 성능 저하를 막을 수 있다고 생각하였다.



NNUNet 자체 라이브러리에 이와 유사한 Region-based training을 지원하지만, 단일 클래스 분류에 추가적인 전처리가 필요한 점, 적용을 한다고 하더라도 결국 추론 시 단일 클래스 분류로 적용된다는 점이 있어 훈련 시는 클래스를 부여하고, 추론 시는 해당하는 클래스에 overlap 클래스를 넘겨주도록 직접 구현하였다. NNUNet으로 해당 방법을 비교하였으며, Validation 기준으로 겹치는 클래스에서 0.2 ~ 0.3 정도 증가했고, 특히, pisiform의 경우 0.82에서 0.885로 0.55 정도 증가함을 보였다. 전체 Validation DICE는 **0.93 -> 0.95**로 향상되었으며, 이후 진행한 모든 NNUNet 실험에서 해당 방법을 적용하였다.

H. Dense CRF

예측 결과를 시각화 했을 때 하나의 클래스가 다른 뼈의 일부까지 동시에 라벨링하는 경우가 관찰되었다. 픽셀 간 거리가 멀다면 같은 클래스에 속하지 않도록 Dense CRF로 후처리를 진행해보았다. 두 개의 모델에 적용하였을 때, public 점수가 한번은 0.0001 상승하고 한번은 0.0005 하락하였다. 성능 향상 효과가 확실하지 않고 inference 시간도 크게 늘어나 Dense CRF는 사용하지 않았다. 이미 예측의 정확도가 높을 때는 Dense CRF가 필요 이상으로 라벨을 축소시켜 성능이 하락하는 것으로 보인다.

4.7 최종 결과

위와 같은 실험을 진행한 결과물로 Hard Voting을 진행 후 최종적으로 0.9768 Dice Coefficient를 달성하였다.

개인 회고

김태욱

1. 나는 내 학습목표 달성을 위해 무엇을 어떻게 했는가?

이번 프로젝트는 Segmentation의 이해도를 높이는 것을 목표로 두고 진행하였습니다. 먼저, Segmentation 기법에 사용되는 Encoder를 탐색 및 실험을 진행하였고 진행한 모델의 구조가 프로젝트에 어떤 영향을 주는지 알아보며 이해도를 높였습니다.

2. 나는 어떤 방식으로 모델을 개선했는가?

SMP 라이브러리의 Encoder 모델 12개의 모델을 실험하며, 가장 높은 정확도를 가진 timm-efficientnet 모델을 메인 실험 모델로 선택하며 프로젝트 정확도를 개선하는데 도움이 되었다고 생각합니다.

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

Hard Voting Ensemble을 진행하여 여러 실험의 결과물을 이용해 각각 모델의 정확도와 비교하였을 때 높은 상승을 가져왔습니다. 이후 추가적인 Ensemble을 진행하며 Hard, Soft Voting의 이해도를 높일 수 있었습니다.

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

지난 프로젝트와 비교하였을 때 여러 가지 방법을 집중적으로 작업 및 실험을 하지 못한 것을 보완하여 하나의 실험이 종료될 때까지 한 실험주제 내에서 구체적으로 진행하였습니다. 해당 보완점으로 인하여 프로젝트에서 맡은 작업에 대해서 이전보다 이해도를 높일 수 있었습니다.

5. 마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?

마주한 한계로는 프로젝트를 처음 경험하면서 CV 도메인에 대한 지식이 부족하여 팀원들보다 많은 실험과 결과를 내지 못하여 아쉬웠습니다. 비록 프로젝트에 많은 기여를 주지 못하였지만 개인 학습이해 부분에서는 있을 정도로 크게 상승한 체감을 느꼈습니다.

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

이전까지 배웠던 프로젝트에서 얻게 된 경험과 지식을 이용하여 새로운 프로젝트를 진행하며 기여도를 높이려고 노력하고 있습니다. 이후 프로젝트도 마찬가지로 지금까지 경험했던 내용을 이용하여 결과물 퀄리티를 높일 수 있도록 노력하고 싶습니다.

김상유

1. 나는 내 학습목표 달성을 위해 무엇을 어떻게 했는가?

이번 프로젝트에서는 높은 순위를 달성하는 것을 목표로 삼았습니다. 성능을 높이기 위해 다른 Segmentation에서 주로 사용한 기법들을 찾아보고 강의에서 추천해 준 방법들을 모두 실험해 보는 것에 중점을 두었습니다. 빠르게 실험해 볼 수 있는 것을 먼저 해보고 시간이 오래 걸리는 것은 이후에 시도해 보았습니다.

2. 나는 어떤 방식으로 모델을 개선했는가?

팀원들의 실험 결과를 토대로 좋은 조합을 찾아서 최고 성능을 내고자 하였습니다. 일정 수준의 성능을 올린 후에는 슬라이딩 윈도우, SWA, Loss와 하이퍼파라미터 수정을 통해 단일 모델 성능을 최대한 높였습니다.

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

팀 내에게 가장 성능이 높은 단일 모델을 생성했고, 강의에서 언급한 내용 대부분을 실험해 보았습니다. 실험 과정에서 같은 내용의 실험이라도 환경에 따라서 성능 변화가 상이하다는 것을 깨달았고 단일 실험으로 결과를 평가하면 오류가 발생할 가능성이 높다는 것을 느꼈습니다.

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

이전 프로젝트에서는 단일 라이브러리를 이용한 실험이 많았습니다. 이번에는 SMP와 MMsegmentation외에도 커스텀 모델을 이용해 실험을 진행했습니다. 그 결과 모델의 다양성이 증가하여 앙상블에서 높은 성능 향상을 이뤄냈습니다.

5. 마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?

6명의 인원이 4개의 서버를 사용하는 과정에서 효율적인 서버 활용이 어려웠다고 생각합니다. 작동하지 않는 서버가 있는 경우가 많았고 이러한 점을 개선한다면 더 많은 실험이 가능했을 거라 생각됩니다.

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

다른 팀에서는 노션 API를 통해 서버 사용현황을 공유하여 서버 관리를 하였는데 이러한 방식을 시도해보고자 합니다. 또한 이번 프로젝트의 경우 짧은 기간에 많은 실험을 위해 각자의 코드가 조금씩 상이한 경우가 많았는데 최종 프로젝트에서는 팀원 모두가 동일한 코드를 공유하여 서로의 서버를 공유하기 편한 환경을 만들고 싶습니다.

김세연

1. 나는 내 학습목표 달성을 위해 무엇을 어떻게 했는가?

효율적인 실험 관리, 공유 및 구현을 위해 YAML 파일을 통한 실험 관리 코드를 작성했습니다. 특히, abstractmethod을 통해 모델 추가 및 수정 시 모델의 공통적인 구조을 가지게 하였으며, 이를 통해 Train 코드의 변경 없이 모델에 따라 훈련이 가능하도록 하였습니다. 또한, 기존의 basecode에 나아가 NNUNet v2라는 프레임워크를 시도하며, 다양한 파이프라인과 구조에 대한 이해를 쌓을 수 있었습니다.

2. 나는 어떤 방식으로 모델을 개선했는가?

DUCKNet의 성능 향상을 위해 모델 구조를 수정하여 SMP에서 제공하는 다양한 Backbone을 활용할 수 있도록 채널을 조정하였습니다. NNUNet의 경우, Multi-Label을 지원하지 않는다는 한계를 극복하고 중복되는 영역의 성능을 개선하기 위해 새로운 클래스로 해당 영역을 정의하여 학습 단계에서 명시적으로 처리하도록 하였으며, 추론 단계에서는 해당 겹치는 영역을 각 클래스에 적절히 분배할 수 있도록 구현하였습니다. 이를 통해 기존 NNUNet의 제약을 효과적으로 보완할 수 있었습니다.

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

성능 향상을 위해 데이터에 적합한 모델을 탐색하고 다양한 방법들을 적용하는 과정에서 추론 성능을 시각화하고 모델 간 성능을 비교 분석하며 모델 및 데이터 특성에 대한 깊은 이해를 쌓을 수 있었습니다. 단순히 모델의 결과를 확인하는 것에 그치지 않고, 왜 특정 모델이 혹은 특정 방법이 잘 작동하며 작동되지 않았는지에 대한 원인을 탐구할 수 있는 귀중한 경험이었습니다.

또한, NNUNet를 사용하며 단순히 프레임워크에서 주어진 기능을 그대로 활용함에서 나아가 상황에 맞는 방법을 모색하고 이를 직접 구현해가는 과정을 겪었습니다. 이 경험을 통해 문제 해결에 필요한 주도적인 사고의 중요성을 깨달을 수 있었습니다.

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

가장 큰 변화는 실험 관리 방식의 전환이었었습니다. 기존에 단순히 bash 스크립트를 통해 실험 환경과 세팅을 구성하는 방식으로 실험을 진행했으나, 이번 대회에서는 YAML 파일을 활용해 실험의 전반적인 청사진을 작성하고 관리할 수 있게 하였습니다. 이를 통해 실험 설정과 진행 사항을 더 체계적으로 기록하고, 실험 파라미터를 명확하게 정의함으로써 관리의 효율성을 크게 향상시켰습니다.

또한, 각 실험의 YAML 파일에 번호를 부여하고 이를 Notion과 Wandb와 동기화하여 팀원들이 실험의 세팅과 결과를 쉽게 파악할 수 있도록 구조화했습니다. 이로 인해 팀 내에서 실험이 어떻게 진행되었는지, 어떤 세팅에서 어떤 결과가 나왔는지를 직관적으로 확인할 수 있었고, 협업이 더욱 원활해질 수 있었다고 생각합니다.

나아가, 이전에는 실험 결과의 정리와 개선점 도출, 그리고 그것을 공유하는 데 충분한 시간을 할

애하지 못했으나, 이번에는 실험이 끝난 후 결과 분석과 개선점을 도출하고 이를 공유하는 데 더 많은 시간을 투자했습니다. 매일 데일리 스크럼 시간과 피어 세션에서 서로 발표하고 의견을 나누면서, 팀원들이 겪고 있는 문제점을 보다 명확하게 파악하고, 향후 실험에 반영할 수 있는 중요한 인사이트를 얻을 수 있었습니다.

5. 마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?

첫 번 째로는 시간적 한계가 있었습니다. 실험 결과 정리 및 인사이트 도출에 더 많은 시간을 할애하고자 했으나, 대회 마지막 주에는 실험의 수를 늘리는 데 집중할 수밖에 없었습니다. 이로 인해 실험을 더 세밀하게 분석하고 개선점을 도출하는 데 있어 다소 제한적인 시간을 갖게 되어 아쉬운 부분이 있었습니다.

두 번 째로는 서버의 한계였습니다. 기존 NNUNet에서는 U-Net 계열의 모델밖에 사용하지 못한다는 한계가 있어, U-Mamba 등 다른 모델의 사용을 시도했지만, CUDA 설치 문제로 서버에 이식을 하지 못하였습니다.

하지만, 이러한 한계 속에서도 다른 방법을 모색하여 해결 방법을 찾아내고, 주어진 시간 내에 서로의 업무를 공유함으로써 좋은 성적을 낼 수 있었다고 생각합니다.

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

마스터 클래스 시간에 다른 팀들의 발표를 들으면서 다양한 공유 방식이 존재한다는 점을 깨달았습니다. 특히, 노션 API를 활용하여 서버를 관리하고, 팀원들이 돌아가며 코드 리뷰를 하는 방식을 보았을 때, 이러한 접근 방법을 활용하면 바쁜 시간 속에서도 팀원의 업무 진행 상황을 보다 효율적으로 파악할 수 있을 것이라고 생각합니다. 다음 프로젝트에서는 이런 방식들을 적용하여 팀 간의 협업과 업무 공유를 더 원활하게 하고, 실험 관리와 코드 리뷰 과정을 더욱 체계적으로 개선해보고 싶습니다.

안지현

1. 나는 내 학습목표 달성을 위해 무엇을 어떻게 했는가?

이번 프로젝트에서는 모델링 및 모델 개선을 위한 실험에 익숙해지는 것이 목적이었고, 이를 위해 모델 구조를 팀의 공식 템플릿에 맞춰 구현하였다.

또한 실험 후 오류 분석을 위해 직접 멀티 레이블 세그멘테이션의 혼동행렬을 계산하는 코드와 추가로 경계 오류 및 클래스 별 Dice 점수 분포를 시각화하는 코드를 작성했다.

2. 나는 어떤 방식으로 모델을 개선했는가?

모델에서 풍부한 특징을 추출하기 위해 이미지 사이즈를 키워 파인튜닝하였고, 어려운 케이스의 데이터 학습을 완화하기 위해 DiceFocalLoss를 도입하였다. 디코더의 구조에 따라 학습 과정 중 업샘플링 시에 세그멘테이션 성능이 갈린다는 것을 가정하고 더욱 치밀한 구조의 디코더를 가진 모델을 채택하기도 하였으며, 트랜스포머 기반의 모델을 학습할 때는 특히 더욱 많은 특징 데이터를 모델에게 제공하기 위하여 원본과 같은 사이즈의 이미지로 학습하였다.

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

여러 모델을 사용하여 모델을 갈아끼우면서 성능을 증가한 것이 아닌 하나 혹은 두 개의 모델을 사용하여 성능을 증가시키기 위해 다양한 시도를 해봄으로써, 모델의 성능을 증가시키는 데에는 모델 구조만 있는 것이 아니고, 더욱 정밀한 특징 추출을 위한 데이터 전처리 및 세팅의 중요성을 알게 되었다.

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

실험 결과를 분석하기 위해 직접 오류 분석 코드를 작성했으며 이 과정에서 실제 모델의 평가지표인 Dice 점수를 계산할 때 어떤 흐름으로 계산이 되는지 알 수 있었다. 또한 실험 후 오류 분석을 통해 해당 Task에서 중요하게 생각할 지점이 어떤 것인지 스스로 가설을 세울 수 있게 된 것 같다.

5. 마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?

실험이 빠르게 진행되었고 또한 대회 기간이 길지 않았기에 체계적인 실험을 통해 정밀한 가설 검증을 해보지 못한 것이 아쉬웠다.

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

프로젝트가 항상 체계적인 방식으로 진행될 수는 없겠지만 실험 파이프라인부터 계획까지 세워서 허둥지둥하지 않고 해당 Task에서 중요한 게 무엇인지 생각해 볼 수 있는 시간을 확보하고 싶다. 또한 다음 실험 시에는 너무 많아 관리가 안되는 실험보다는 자료조사를 통해 선택과 집중을 해서 모델의 성능을 높이고 싶다.

김윤서

1. 나는 내 학습목표 달성을 위해 무엇을 어떻게 했는가?

이번 프로젝트에서는 좀 더 주도적으로 새로운 모델과 기법을 실험해보고자 하였다. 시도해보지 않은 모델을 직접 구현해보거나 후처리 방법을 적용해보며 실험 다양성을 높이고자 하였다.

2. 나는 어떤 방식으로 모델을 개선했는가?

DUCK-Net 학습 시 적절한 학습률을 찾기 위해 여러 번의 실험을 통해 조정하였고, 스케줄러와 loss를 바꿔 파인튜닝함으로써 최대한 성능을 높이고자 하였다. 또한, 결과 시각화에서 파악한 문제를 해결하기 위해 Dense CRF로 후처리를 시도하였고 이때 강도나 반복 횟수를 결정하기 위해 validation set으로 여러 번의 실험을 거쳐 파라미터를 결정하였다.

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

EDA와 데이터 확인 과정을 통해 여러 인사이트를 도출해냈다. 이전에 경험해봤던 EDA는 깊이가 얕아 데이터 구성을 파악하는 데에 그쳤는데, 이번 EDA에서는 시각화나 계산 방식을 바꿔가며 새로운 결론에 도달하고자 고민하였다. 여기서 발견한 인사이트를 실험의 근거로 활용하면서 데이터 분석의 방법과 중요성을 알 수 있었다.

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

이전까지는 라이브러리에서 제공하는 모델을 불러와서 사용하기만 했다면, 이번 프로젝트에서는 Vanilla DUCK-Net과 SMP의 pretrained encoder를 연결한 U-Net3+ 모델 코드를 직접 작성해보았다. 데이터의 해상도와 채널 수를 고려하여 레이어와 순전파 함수를 직접 정의해보면서 구조에 대한 이해도를 훨씬 높일 수 있었다.

5. 마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?

한정된 시간 안에 특정 방법이 효과가 있는지 판단해야 하기 때문에 하나의 실험으로 결론을 내린 경우가 여럿 있었다. 서로 다른 조합으로 실험해보는 등 최적의 조건을 위한 탐색이 좀 더 이뤄지지 못한 점이 아쉬웠다. 또한, 이전보다 기록은 체계화되었으나 실험 계획이나 결과 분석에 대한 대화가 많이 못 이뤄진 점이 아쉬웠다.

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

다음에는 실험의 횟수를 더 늘려 여러 실험 결과를 토대로 효과를 판단해보고자 한다. 이를 위해서는 실험에 대한 준비를 좀 더 미리 하여 빠르게 실험이 이뤄지도록 해야 할 것 같다. 또한, 실험 전후 팀원과 대화를 더 나누며 실험의 방향을 함께 논의해보면 좋을 것 같다.

김채리

1. 나는 내 학습목표 달성을 위해 무엇을 어떻게 했는가?

이번 프로젝트에서는 보다 다양한 실험을 코드를 직접 작성하며 수행하고자 하였다. 이에 csv를 추출하여 시각화를 하는 코드를 작성하였고, MMSegmentation을 본 대회에 맞춰 코드를 customize하여 실험하였다.

2. 나는 어떤 방식으로 모델을 개선했는가?

loss를 변경해보고, augmentation 기법을 적용해보았으며, sliding window, stochastic weight average를 MMSegmentation에서 적용해보았다.

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

Transformer 기반 모델을 학습시킴으로써 높은 성능의 모델을 도출하였다. 학습시간이 오래 걸리지만 transformer 모델의 성능이 매우 높다는 것을 느끼게 되었다.

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

보다 라이브러리를 적극적으로 customize해보았고, validation 결과를 csv파일로 추출하여 시각화하는 업무도 하였다. 또한 최종적으로 사용되지는 않았지만 바로 pdf로 시각화 결과를 추출하는 실험도 함으로써 보다 코드에 친숙해지는 기회가 되었다. 나아가 실험 결과를 최대한 잘 정리하여 팀원들과 공유하고자 하였고, 깃도 Issue, pull request 등 제대로 활용하는 시간이었다. 이에 협업 툴에 친숙해지는 효과를 얻었다.

5. 마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?

MMSegmentation customize하는 것이 어려웠다. SWA를 특정 epoch 이후에 적용하는 것으로 코드를 변경하고 싶었지만, 처음부터 적용되는 형태까지로만 적용해볼 수 있었다. 또한 대회에 집중하다보니 강의와 과제에 소홀했던 점이 아쉬움으로 남아있다.

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

이번 프로젝트를 통해 성장한만큼 다음 프로젝트에서는 더욱 적극적으로 코드를 customize하는 시도를 해야겠다.