

Hand Bone Image Segmentation WrapUP Report

김보현, 김재환, 김진석, 박진영, 성기훈, 양호철¹

¹Naver Boostcamp AI Tech, CV 13조

1 프로젝트 개요

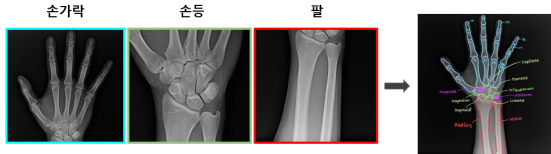


Figure 1: Overview

뼈는 우리 몸의 구조와 기능에 중요한 영향을 미치기 때문에, 정확한 뼈 분할은 의료 진단 및 치료 계획을 개발하는 데 필수적이다. Bone Segmentation은 인공지능 분야에서 중요한 응용 분야 중 하나로, 특히, 딥러닝 기술을 이용한 뼈 Segmentation은 많은 연구가 이루어지고 있으며, 다양한 목적으로 도움을 줄 수 있다.

Dataset은 train 800장, test 288장, 크기(2048, 2048, 3)로 주어졌다. Class는 29개의 뼈가 있다. Annotation: polygon point 형식으로 제공되었다. Framework는 PyTorch, smp, mmsegmentation, ultralytics를 사용했다.

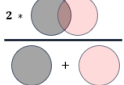
$$Dice = \frac{2 * |A \cap B|}{|A| + |B|}$$


Figure 2: Metric

Test set의 Dice coefficient로 평가한다. Semantic Segmentation에서 사용되는 대표적인 성능 측정 방법이다.

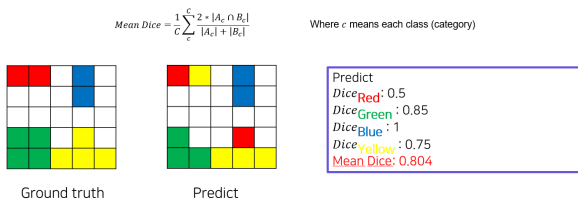


Figure 3: Example of Mean Dice

모델의 예측을 RLE (Run Length Encoding) 한 후에 제출했다.

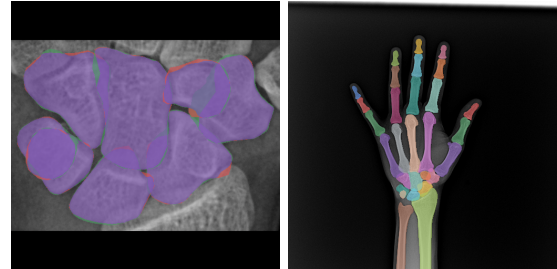


Figure 4: Weight & Bias

2 Tools

2.1 Wandb

기본으로 제공하는 데이터셋인 ImageMask를 사용할 경우 마스크의 시각화가 가능한 것으로 일견 보이는데, 문제는 주어진 29개의 클래스 마스크들은 다들 서로 겹치는 부분이 일부 존재하기에, 하나로 합치면 정보의 손실이 필연적으로 발생할 수밖에 없으며, 합치지 않고 올리면 많은 마스크 맵을 위한 UX가 구성되어 있지 않아 정상적으로 사용이 불가능하다. 따라서 이 마스크들을 적절히 겹쳐지지 않는 클래스들끼리 묶어서 그룹 마스크 맵을 만들어 업로드하면, 이 문제를 일부 보완하여 사용할 수 있는 형태로 업로드가 됩니다.

그러나, 기본적으로 제공되는 모델을 사용해도 모든 클래스가 Dice Metric 기준으로 모두 85점 이상 찍히기 때문에, 단순 라벨과 예측값을 시각화 해 업로드 하는 것으로는, 예측값의 마스크 특성을 자세히 살펴볼 수 없었다. GT와 Pred 마스크 간의 차이가 얼마나 존재하는지 알기 위해, 라벨에만 포함되는 영역을 1번, 예측값에만 포함되는 부분을 2번, 그리고 그 두 개가 겹치는 부분을 3번으로 클래스를 지정하여 Wandb 상에서 모델별로 예측값이 어떤 특성을 갖고 있는지 후에 기술할 Confusion Matrix와 함께 잘 살펴볼 수 있었다.

2.2 MMSegmentation

강의 시간에도 소개했던 대로, MMSegmentation은 Semantic Segmentation을 위한 라이브러리이기 때문에, 멀티 마스크에 대한 지원을 하지 않는다, 따라서 이를 해결하려면 모델 및 여러 모듈, 증강, 데이터셋 모두 수정을 거쳐야 했는데, 전반적인 부분은 오피스아워와 동일하게 구현하였고, 추론 부분은 조금 다르게 구현하였는데, 일반적인 MMEEngine의 흐름에 맞추어 Test를 진행하면 Evalutaor에 설정한 Metric이 결과값이 나오기 때문에, 라

이브러리 내에 이미 존재하는 cityscapemetric 내에서 csv파일을 생성하는 것에서 착안하여 RLEncoding이라는 metric를 새로 만들어 예측값 기반으로 마스크 맵을 만들고, RL 인코딩을 진행하여 저장까지 해당 metric 내에서 진행하도록 구현했고, 정상적으로 추론 값을 만들어 낼 수 있었다.

2.3 YOLO

YOLOv11x-seg YOLO 아키텍처를 기반으로 instance segmentation 기능을 강화한 모델로 객체들이 겹쳐 있는 경우에도 각각의 객체를 개별적으로 구분하기 위해 사용했다.

YOLOv8x-seg yolo11x-seg와 비교하기 위해 사용하였고 큰 성능차이는 없었다.

3 프로젝트 팀 구성 및 역할

이름	역할
김보현	YOLO / Recording
김재환	SAM2UNet / Ensemble
김진석	Augmentation / Refactoring
박진영	Loss function / Model Test
성기훈	Building Experiment Infrastructure
양호철	Data Preprocessing

4 프로젝트 수행 절차 및 방법

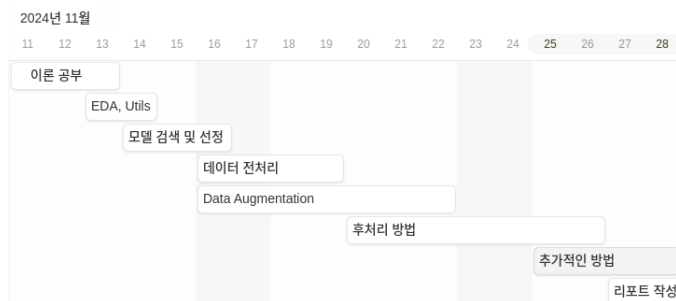


Figure 5: Project Schedule

4.1 시각화

matplotlib, wandb로 이미지 및 annotation을 시각화하여 EDA를 진행했다.

4.2 EDA 및 문제 제기

CVAT에 호환되는 annotation format인 Dataro format으로 변환하는 코드를 작성하여 수정했다.

train dataset장을 전수조사했다.

4.2.1 finger-3의 라벨 일관성 문제

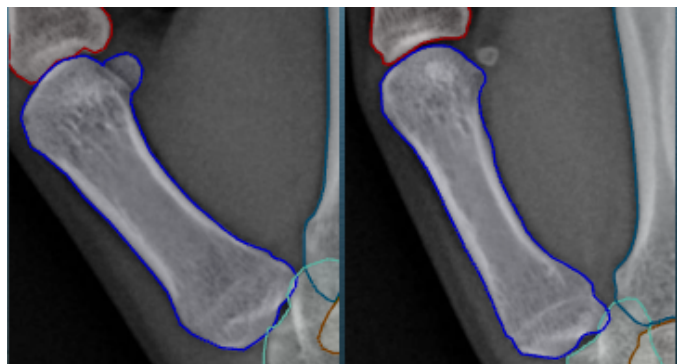


Figure 6: finger-3의 Sesamoid Bone 포함여부 비교

annotation 중 일부가 sesamoid bone을 finger-3에 포함되지 않은 점 발견했다.

sesamoid bone을 모두 제외한 것과 모두 포함한 것의 제출 성능 비교 결과 0.0008점 차이로 모두 포함하는 것으로 결정했다.

4.2.2 라벨링 오류 문제

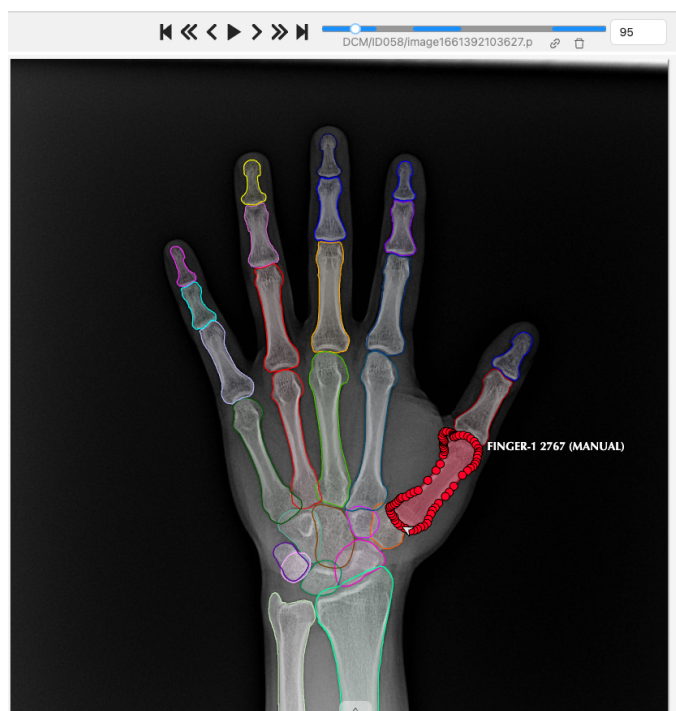


Figure 7: label이 바뀐 finger-1, finger-3

ID058의 왼손 사진에서 finger-1, finger-3의 label이 바뀐 것을 확인하고 annotation을 수정했다.

4.2.3 이상치 문제



Figure 8: 반지 착용

ID363의 오른손 사진에서 반지를 발견하고 양손 데이터를 모두 제거했다.

4.2.4 finger-15의 라벨 일관성 문제



Figure 9: finger-15의 라벨링 문제

뒤쪽 데이터에서 finger-15의 라벨링이 대다수의 다른 데이터와는 다르게 덜 되어있는 부분 발견했다.

4.2.5 면적 치우침 문제

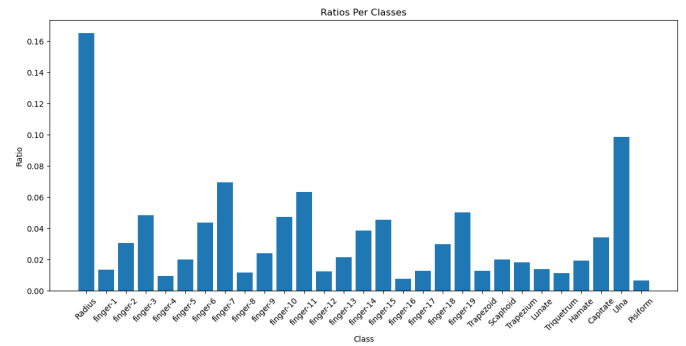


Figure 10: 상대적으로 높은 비율의 면적을 가지는 Radius, Ulna 특정 Class의 면적이 상대적으로 큰 비율을 갖는 것을 확인했다.

4.2.6 label overlap 문제

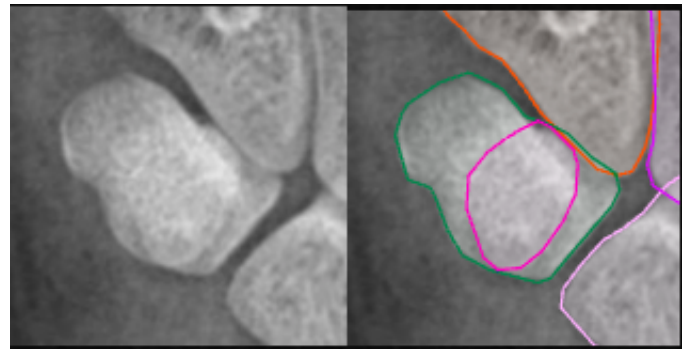


Figure 11: 손목 부근에서 겹치는 라벨

손목 부근에서 육안으로 확인하기 어렵고 라벨이 완전히 겹치는 annotation이 존재하는 것을 확인했다.

4.3 가설 설정

EDA 결과를 바탕으로 데이터에 대한 가설을 제시했다.

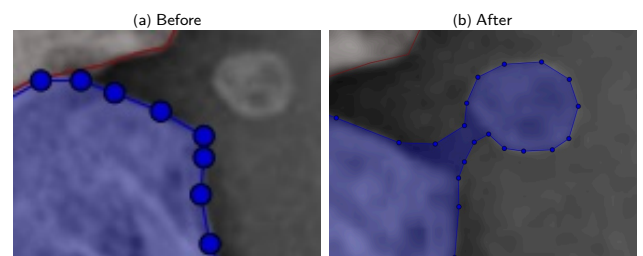


Figure 12: CVAT를 이용한 re-labeling

finger-3의 라벨 일관성 문제는 대다수의 annotation이 Sesamoid Bone을 포함하는 라벨이 많기 때문에 이 방법이

Ground Truth에 가까울 것으로 예상했다.

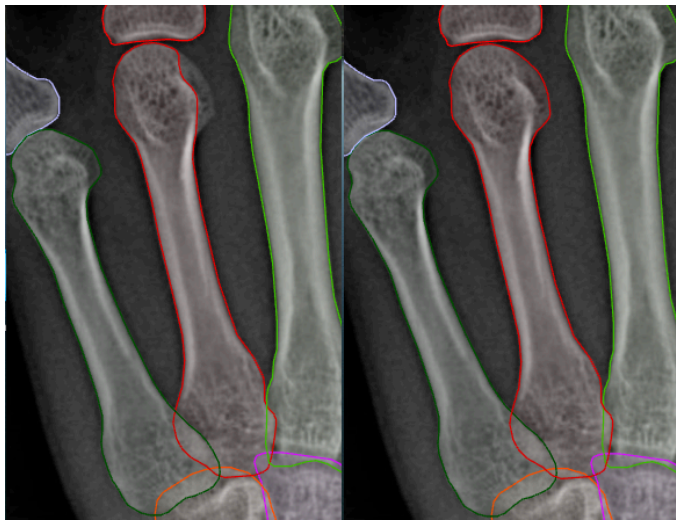


Figure 13: finger-15의 라벨링 수정

finger-15에서 포함되지 않은 부분을 포함하는 것이 Ground Truth에 가까울 것으로 예상했다.

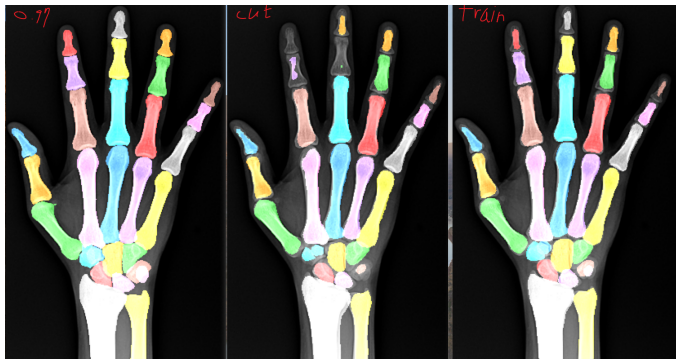


Figure 14: Label의 Min, Max 좌표를 기반으로 한 Crop 및 padding의 성능 확인

Area가 비교적 큰 Radius, Ulna의 경우, 해당 데이터가 전체 모델의 일반화 성능에 부정적인 영향을 미칠 가능성이 있다. 이를 해결하기 위해 해당 영역의 크기를 줄이는 방안을 고려해야 한다. 이 과정에서 일반화 성능이 떨어질 우려가 있으므로 이에 대한 보완책도 함께 검토해야 한다. 또한, 데이터셋에 완전 겹치는 뼈가 존재하는 상황에서는 이를 어떻게 처리할지 명확한 방법을 정의할 필요가 있다. 만약 CVAT와 같은 도구를 사용한다면, 해당 도구의 포맷과 호환되도록 데이터를 수정해야 할지도 검토가 필요하다.

추가로 나이, 성별, 체중, 키와 같은 메타데이터를 학습에 적용할 수 있는 방법을 모색하기로 했다. 추가적인 학습 정보를 모델에 제공하기 위함이다. 현재 모델의 정확도가 이미 높은 상태라면, 데이터 증강보다는 앙상블 기법이 더 효과적일 수 있을거라고 추측했다. 다양한 하이퍼파라미터를 고려하여

최적의 조합을 찾기 위해 wandb sweeps 가이드와 같은 도구를 활용해보기로 했다.

손목 부근의 겹치는 뼈들은 Augmentation, Super Resolution 등의 기법을 이용하면 모델의 예측 성능을 높일 수 있을 것으로 예상했다.

5 Super Resolution

추론 성능이 상대적으로 낮은 손바닥 뼈들(Trapezoid, Trapezium, Scaphoid, Capitate, Hamate, Triquetrum, Pisiform, Lunate)에 대한 추론성능을 높이기 위한 방법으로 Super Resolution을 활용해보기로 했다. 학습 및 추론 방법은 다음과 같다.

주어진 annotation을 활용하여 손바닥 부분을 crop한 뒤 Super Resolution을 활용하여 해상도를 개선시켜 저장한다. 이후 이 데이터셋을 활용하여 손바닥을 집중적으로 학습하는 모델을 만들었다. 전체 이미지로 학습한 모델과 손바닥에 대해서만 학습한 모델의 추론 결과를 앙상블하여 최종 결과를 만들었다.

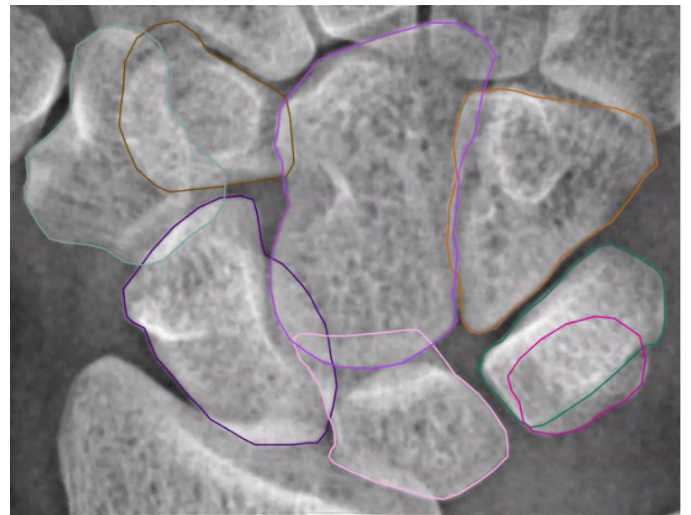


Figure 15: Super Resolution

6 Model

Table 1: Model

Framework	Size	Model	encoder	DICE
smp	1024 × 1024	U-Net++	effb3	0.968
MMSeg	1024 × 1024	UPerNet	HRNet	-
smp	1024 × 1024	UPerNet	HRNet	0.9705
MMSeg	512 × 512	Segformer	mixViT	0.962
Huggingface	512 × 512	Segformer	Hierarchical Transformer	0.919
smp	512 × 512	PSPNet	effb7	0.934
-	1024 × 1024	SAM2UNet	SAM2-hiera	0.9638
Ultralytics	2048 × 2048	yolov8		0.94
Ultralytics	2048 × 2048	yolov11		0.94

Instance segmentation 우리 데이터들 중 손등 뼈들이 많이 겹쳐있다. semantic segmentation은 이미지의 모든 픽셀에 대해 해당 클래스 레이블을 할당하지만, 동일한 클래스의 객체들이 겹쳐 있을 경우 이를 하나의 영역으로 인식하는 한계가 있다. 그렇다면 instance segmentation을 활용했을 때 손등 뼈들을 더 잘 인식할 것이라는 가설을 세우고, ultralytics의 yolov8-seg모델과 yolov11-seg모델을 통해 instance segmentation을 수행했다.

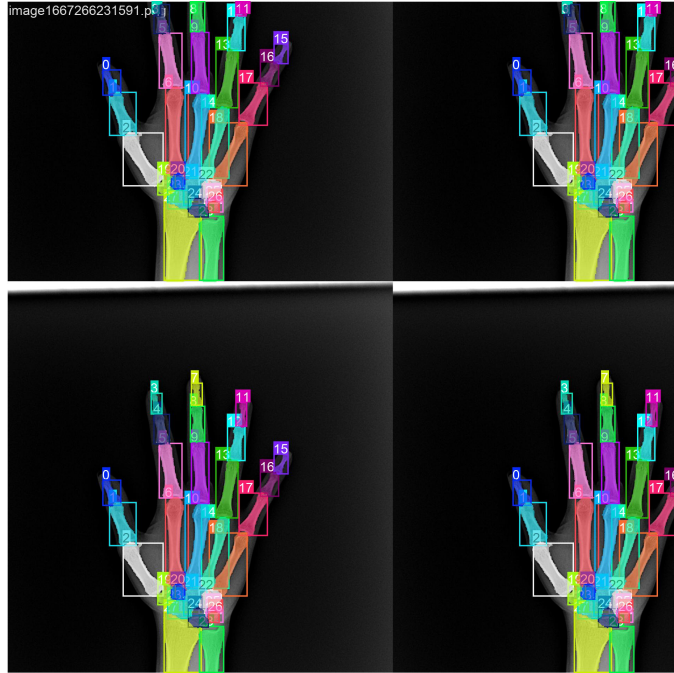


Figure 16: instance segmentation 시각화

7 Augmentation

Color Augmentation을 사용하면 손바닥에 겹쳐있어 구분하기 어려운 뼈들에 대한 성능을 높일 수 있을 것으로 예상했고, Geometric Augmentation을 사용하면 다양한 shape을 갖는 뼈들에 대해 더 정확하게 예측할 것으로 예상된다.

7.1 Color

실험해 본 Color Space Augmentation으로는 CLAHE, ColorJitter, RandomBrightnessContrast, RandomGamma, BilateralFilter 등이 있다.



Figure 17: Color Augmentations

7.1.1 CLAHE



Figure 18: CLAHE와 EqualizeHist 비교

CLAHE와 EqualizeHist 증강 결과를 시각화해 본 결과, CLAHE는 성능 향상이 기대되는 반면에 EqualizeHist는 증강 결과를 시각화해봤을 때 성능이 향상에 도움이 되지 않고 오히려 학습에 방해가 될 것으로 판단했다. 따라서 EqualizeHist 대신 CLAHE만 적용하기로 결정했다.

7.2 Noise & Distortion

육안으로 구분이 어려운 손목 부근의 뼈들을 구분하기 위해 Edge Augmentation, Sharpen, ElasticTransform, Blur, GridMask와 같은 Noise 및 Distortion 관련 Augmentation을 시도했으나, 과도한 증강은 오히려 이미지를 왜곡시켜 성능이 악화되기에 여러 증강을 동시에 적용할 때 최종적으로 이를 제외했다.

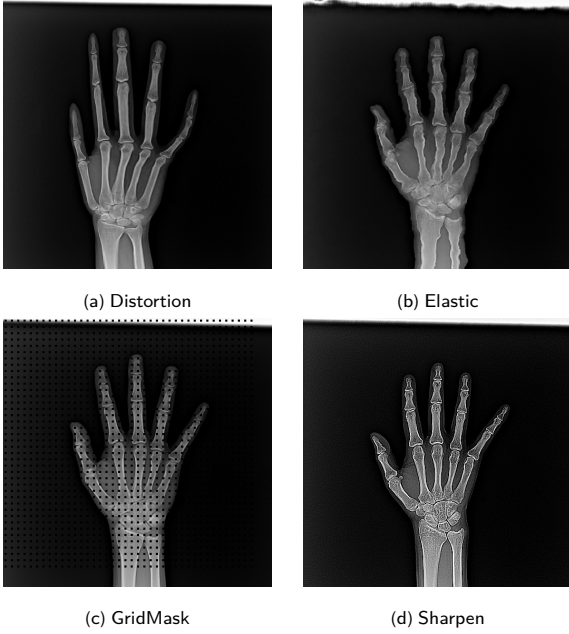


Figure 19: Noise & Distortion Augmentations

7.3 Geometry

대부분의 class가 일정한 shape를 가졌지만 손이 틀어져 있는 데이터와 같이 약간 다른 형태의 데이터도 존재했다. 따라서 class의 형태를 변경하지 않는 HorizontalFlip, Affine(Scale, Translate, Rotate, Shear) 등의 Spatial Augmentation을 선정했다.

7.3.1 Affine

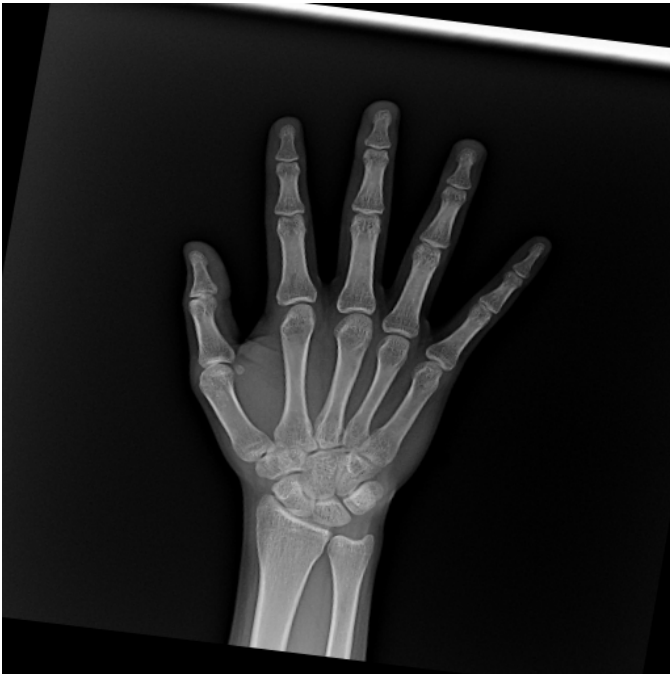


Figure 20: Affine

여러가지 Geometric Augmentation을 한 번에 적용시킬 수 있는 Affine을 사용했다. 약간의 scale 조정과 (-10, 10) 범위의 rotation을 사용했다.

7.3.2 Cypaste

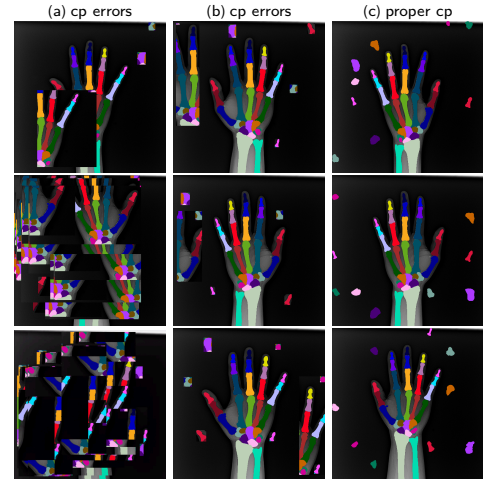


Figure 21: Our trial and error of cp

CopyPaste의 경우 라이브러리에 포함되지 않은 탓에 여러 시행착오 끝에 직접 구현하여 사용했다. 하지만 이를 사용하면 기대했던 바와는 달리 모델의 성능이 떨어졌는데, 그 이유로는 CopyPaste 증강은 다양한 위치에 이미지를 추가하여 원하는 클래스를 추가적으로 학습시켜 일반화 성능을 높이는 반면, 모델이 학습하는 각 클래스의 주변 정보, 패턴 등에 대한 반례를 만들어 클래스의 주변 정보가 고정된 데이터를 사용하는 대회 성능에는 부정적인 영향을 끼치는 것으로 파악되었다.

Augmentation 실험결과를 종합하여 Resize, CLAHE, Affine, HorizontalFlip, Normalize를 함께 적용하기로 결정했다. 그 결과 Confusion matrix에서 그림 27과 같은 변화를 확인할 수 있다.

8 Optimizer

Optimizer로는 Adam과, AdamW를 사용했다. 데이터 셋에 있는 대부분의 데이터의 클래스별 배치가 유사하기 때문에 학습이 진행됨에 따라 과적합될 가능성이 있다고 예상했다. 또한 일부 데이터는 손목이 꺾인 상태로 촬영된 데이터를 포함하고 있기 때문에 일반화 성능을 유지해야한다고 생각해 다양한 패턴에 대해 더 나은 성능을 보여주는 강력한 옵티마이저인 AdamW를 사용했다.

9 Scheduler



Figure 22: cosine annealing warm restarts의 custom scheduler

Adam은 자체적으로 adaptive learning rate를 제공하므로 scheduler 없이도 안정적인 학습이 가능할 것이라고 생각해 초기에는 scheduler를 사용하지 않은 Adam을 사용했다. 100에폭 학습 기준, 대부분의 모델이 60에폭 이전에 최고점에 가까운 val/dice를 도출하고 있었다. 이에 val/dice가 개선되지 않으면 lr을 낮추는 ReduceLROnPlateau 스케줄러를 사용해 학습을 진행하였으나, Adam만 활용해서 학습할 때에 비하여 성능 차이를 거의 보이지 않았다. 이후 cosine annealing을 사용해 local minima에 빠지지 않는 것에 집중하였고, 나아가 더욱 세밀한 학습률 조절을 위해 cosine annealing을 custom하여 cosine annealing warm restarts를 만들어 실험했다. 결론적으로, cosine annealing을 주로 사용했다.

10 Loss function

Loss Test	IoU	Dice	Focal	WBCE	IoU+BCE	Dice+BCE
Public Score	0.9496	-	-	0.9388	0.9485	0.9490

Table 2: Loss test

대회의 리더보드에서 사용하는 Dice 뿐만 아니라 여러가지 Loss를 적용해보고, 여러 Loss를 더해서 사용하는 Hybrid Loss도 실험해봤다. 그 결과, 단독으로 사용하는 것은 IoU Loss가, 여러 Loss를 더해서 사용하는 것은 BCE와 Dice를 사용하는 것이 가장 좋은 성능을 냈다. 다양한 모델과 상황에 대해 테스트하면 더 확실하게 Loss를 적용할 수 있겠지만 시간상 하나의 모델에 대해 테스트를 진행하였고 IOU 단독으로 loss function으로 사용했을 때 점수가 가장 높았지만 IoU만으로는 모든 모델에 대해 잘 적용이 될지 알 수 없었기에 안전하게 BCE와 결합한 Loss function 중 더 높은 점수를 기록한 Dice+BCE Loss를 사용하게 되었다.

11 Pseudo labeling

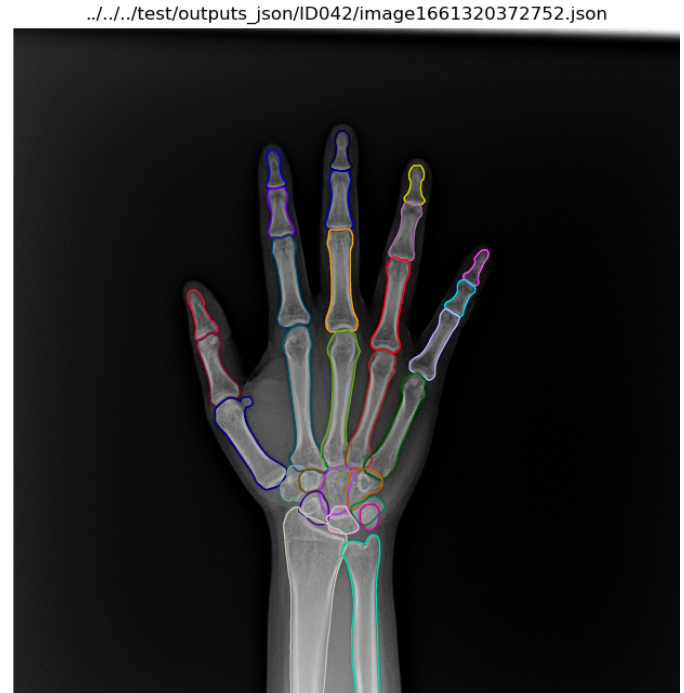


Figure 23: Test Dataset의 inference 결과를 label로 변환한 후 시각화

리더보드에서 높은 public score를 가지는 csv를 가지고 pseudo labeling을 시도해보기로 했지만, 시간이 부족한 관계로 실험을 해보진 못했다.

12 Ensemble

모델의 probability를 가지고 앙상블하는 soft voting과 결과 csv를 가지고 앙상블하는 hard voting을 시도했다. 또한 Super Resolution을 활용한 모델을 활용하기 위한 앙상블도 구현하여 사용했다.

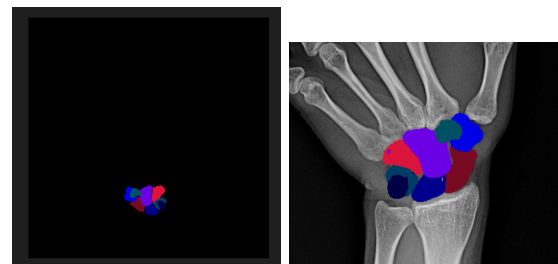


Figure 24: Super Resolution한 이미지로 추론 후 원래 해상도의 위치로 복원

12.1 Hard voting ensemble

모델의 학습 결과를 test하여 얻을 수 있는 csv 파일의 같은 라인에 있는 데이터들을 읽고 하드 보팅을 할 수 있는 방식으로 구현하였다. 손바닥만 크롭하여 학습해 클래스가 적은 csv들은 형식을 맞춰주는 코드를 작성하여 똑같은 형태로 하드 보팅할 수 있게 했다. 다양한 모델들의 csv를 가지고 있었기 때문에 어떤 모델의 csv를 포함할지, threshold를 어떻게 지정해줄지 여러 실험을 거쳤다.

Table 3: Hard Voting Ensemble Model

Framework	Model	Encoder	Data
smp	UNet++	HRNet	All
smp	UNet++	HRNet	Palm
smp	UPerNet	mit-b5	All
smp	UPerNet	EfficientNet-B7	All
smp	UPerNet	HRNet	All
smp	UPerNet	HRNet	Palm
MMSeg	UPerNet	HRNet	All
MMSeg	Segformer	mixViT	All
Ultralytics	yolov11	-	All
-	SAM2UNet	SAM2-hiera	All

12.2 Soft voting ensemble

여러 모델(.pt 확장자)의 픽셀별 추론 결과 logit인 sigmoid output(threshold 연산 이전)의 평균을 구한 뒤 클래스별 threshold를 지정하여 클래스를 결정하는 소프트보팅 방법을 사용했다. MM-Seg에서 불러온 체크포인트 파일을 불러오지 못해 SMP 기반의 모델들에 대해서만 앙상블을 하였다.

Table 4: Soft Voting Ensemble Model

Framework	Model	Encoder	Data
smp	UPerNet	EfficientNet-B7	All
smp	UNet++	HRNet	All
smp	UPerNet	HRNet	All
smp	DeepLabV3+	EfficientNet-B7	Palm
smp	UNet++	HRNet-w64	Palm
smp	UPerNet	HRNet-w64	Palm

13 프로젝트 결과

순위	팀이름	팀멤버	dice	제출횟수
내등수 14	CV_13조		0.9739	92

Figure 25: public score

dice dice (최종)	생성일시
0.9739	2024.11.28 18:57
0.9749	

Figure 26: score

13.1 자체 평가 의견

13.1.1 잘했던 점

새로운 팀원들과 처음으로 팀합을 맞추면서 다양한 가설을 제시하고 여러 아이디어들에 대한 토론을 하였고 실제로 구현하여 다양한 측면에서 테스트를 진행하였다. WandB, Google sheet 등을 활용하여 모델 테스트 현황을 보기 좋게 정리하였고 노션, 깃허브를 활용하여 프로젝트 전반에 관한 진행 상황을 기록하고 코드를 공유하여 팀적인 움직임을 더 수월하게 할 수 있었다. 또한 다양한 시각화를 하면서 EDA를 수행하고 validation 결과를 공유하고 분석하여 모델 향상을 위한 새로운 기법 등을 제시할 수 있었다.

13.1.2 시도 했으나 잘 되지 않았던 것들

프로젝트를 진행하면서 다양한 시도들을 했지만 시간 및 컴퓨터 자원의 부족으로 아이디어만 제시되고 실제로 실현하지 못한 경우도 많았다. 특히 모델마다 Augmentation 테스트를 하지 않고 한 모델에 모든 테스트를 진행하였기 때문에 각 모델에 따른 Augmentation의 유효성에 대한 테스트를 못한 점이 아쉬웠다. MMSegmentation 라이브러리를 이번 프로젝트에 맞게 구축하였지만 시간이 부족해 많은 모델에 대한 테스트를 하지 못했다. 또한 seed를 고정해도 실험 재현이 불가능했던 문제가 있었지만 이에 대한 해결을 하지 못하였고 Cosine annealing warm restarts 스케줄러에 대한 실험을 많이 하지 못해 아쉬웠다. 학습 데이터의 배경을 지우고 너무 큰 면적을 차지하는 Radius, Ulna의 면적을 줄여서 학습을 시켰지만 시각화 했을 때 오히려 segmentation을 잘 하지 못하는 모습을 보여서 사용되지 못했다. 앙상블에 대해서도 더 많은 테스트를 진행할 수 있었으면 좋았겠지만 각 클래스마다 예측과 ground truth 사이의 차이를 정량적으로 분석하지 못해 디테일한 테스트를 할 수 없었던 것이 아쉬웠다.

13.1.3 아쉬운 점

데이터셋과 함께 주어진 메타 데이터를 멀티모달 형식으로 활용해보자는 아이디어가 나왔었는데 후순위로 밀려 활용하지 못했다. 다양한 scale에서 학습 및 추론한 결과를 앙상블하는 방법과 sliding window 또한 비슷한 이유로 사용해보기 못했고, pseudo-labeling 같은 경우에는 구현에는 성공했으나 아이디어 자체가 늦게 논의되어 실험할 시간이 부족했다. CVAT 포맷 변환 코드 작성의 지연으로 다양한 전처리 실험을 해보지 못했다. 시간이 상대적으로 적게 걸리는 task를 빠르게 구현하지 못하여 지연된

점이 아쉽다.

13.2 프로젝트를 통해 배운점 또는 시사점

대회를 진행하면서 좋은 모델을 선정하는 것 뿐만 아니라 Dataset의 전처리를 통해서도 성능을 향상시킬 수 있다는 점, Albumentations 등의 library를 활용한 augmentation 말고도 postprocessing 등 다양한 이미지 처리 방법을 직접 구현해 학습에 활용할 수 있었고, 학습된 모델의 약점을 보완하기 위해 약점에 특화된 모델, 이를테면 손바닥 부분을 Crop한 뒤 Super Resolution하여 손바닥 부분에 특화된 모델을 만들어 추론결과를 합치는 방법을 생각해 내어 유효한 성능향상을 보인 점 또한 재밌었다.

14 개인 회고

14.1 김보현

14.1.1 학습 목표 달성을 위한 노력

강의에서 배운 점을 적용하고 더불어서 instance segmentation 모델을 가져와서 우리 task에 맞게 바꾸어 적용하려고 시도했다. DICE가 0.94대로 내가 세운 가설과 다르게 성능이 오르지 않았지만 앙상블에 넣었을 때 성능 향상을 시킬 수 있었다. 주말에 제출 횟수가 남으면 다양한 모델로 hard voting ensemble을 해보며 어떻게 앙상블해야 좋은지 파악했고 결과적으로 같은 모델끼리 앙상블 하고 그 결과들끼리 또 앙상블 했을 때 결과가 가장 좋았다.

14.1.2 마주한 한계와 아쉬웠던 점

detectron2를 써보고 싶었는데 이슈만 파고 시간 관계상 못해봐서 아쉽다. 그리고 greedy soup도 마지막 날에 시도했는데 시간이 부족해서 못해봐서 아쉽다. ultralytics 모델을 batch 1로 학습했는데 (out of memory) 2 이상으로 해봤으면 더 잘되지 않았을까 하는 아쉬움이 남는다.

14.1.3 한계/교훈을 바탕으로 다음 프로젝트에서 시도해보고 싶은점

체계적으로 데드라인을 정해놓고 미리미리 해놓으면 좋을 것 같다. 다양한 framework를 공부하고 적용해보고 싶다.

14.2 김재환

14.2.1 학습 목표 달성을 위한 노력

작성된 베이스라인의 라이트닝 모듈을 기반으로 SAM2UNet이라는 SOTA 모델을 Github에서 가져와 프로젝트에 사용할 수 있도록 하였다. 원활한 서버 관리를 위하여 서버의 GPU 사용 현황과 저장 공간 현황을 슬랙으로 알려주는 기능을 추가했고, 체크

포인트를 활용한 resume과 같은 편의 기능을 도입했다. 데이터 증강과 모델 테스트에 기여했고, 하드 보팅 앙상블을 시행했다.

14.2.2 마주한 한계와 아쉬웠던 점

코드를 이해하는 능력이 부족해서, 코드 구현까지 시간이 너무 오래 걸렸다. 팀원들의 도움으로 파이토치 라이트닝 모듈과 같은 다양한 코드를 접해볼 수 있고 큰 경험이 되었는데, 본인은 기능적으로 프로젝트에 크게 기여하지 못한 것 같아 아쉽다.

14.2.3 한계/교훈을 바탕으로 다음 프로젝트에서 시도해보고 싶은점

완벽한 학습이 선행된 이후에 코드 작성을 해야 할 것이고, 코드 읽는 능력을 많이 길러야 할 것 같다. 프로젝트 진행을 하기 앞서서 알고리즘 테스트가 됐든 복습이 됐든 코딩 경험을 좀 더 늘리고, 개발자로서 전문성을 가질 수 있게 노력할 것이다. 구현은 차치하고서라도, 이론적 역량을 키워 더 다양한 방법론을 제시할 수 있도록 하겠다.

14.3 김진석

14.3.1 학습 목표 달성을 위한 노력

제한된 서버와 시간을 효율적으로 활용하기 위해 대략적인 프로젝트 일정을 제안했다. 이후, 다양한 모델의 성능을 효율적으로 테스트하고 개선하기 위해 정기적인 회의를 진행했다. 이러한 회의에서는 테스트 결과를 공유하고, 모델의 단점을 분석하며, 개선 방향에 대해 논의했다. 모델 테스트 과정에서 필요한 기능인 Confusion Matrix를 새롭게 구현하거나, 팀원들과 협업하여 개선이 필요한 영역을 Refactoring하는 작업도 병행했다. 이를 통해 프로젝트의 전반적인 코드 품질과 성능을 향상시켰다. Augmentation 테스트를 주로 담당하였고, 단순히 Albumentation 라이브러리만 사용하는 것에 그치지 않고, Bilateral Filter나 Unsharp Mask와 같은 이미지 처리 기법을 직접 구현하여 사용하려는 시도도 해보았다. 이러한 접근 방식은 이후 다른 프로젝트를 진행하면서 더욱 다양한 Augmentation 방법을 시도할 수 있는 기반이 될 것이라 생각한다.

14.3.2 마주한 한계와 아쉬웠던 점

팀원들이 작성한 Lightning 기반의 베이스 코드를 바탕으로 다양한 방법을 시도해본 것은 좋은 경험이었다. 하지만 이번 프로젝트에서 아쉬웠던 점은 필요한 기능을 직접 구현하는 코드를 작성하는 데 많은 노력을 기울이지 못했다는 점이다.

14.3.3 한계/교훈을 바탕으로 다음 프로젝트에서 시도해보고 싶은점

라이브러리에 제공되지 않은 SOTA 모델을 Github에서 가져와 사용해 보는 작업을 시도해보고 싶으며 Augmentation도 라이브

러리 외에 다른 이미지 처리도 사용해보고 다른 실험이나 코드 작업에 시간 투자를 더 진행하고자 한다. 구현하는 능력은 개인적으로 중요한 부분이라고 생각하기 때문에, 앞으로는 이러한 영역에서도 더 적극적으로 참여하고 기여할 수 있도록 노력할 계획이다.

14.4 박진영

14.4.1 학습 목표 달성을 위한 노력

작성된 베이스라인 코드를 토대로 SMP 및 MMsegmentation 라이브러리를 활용해서 다양한 모델 테스트를 하였다. 성능 향상을 위해 Cosine Annealing 스케줄러와 LR 등 하이퍼파라미터 튜닝을 하였고 다양한 Loss function 테스트를 하였다. 또한 validation 을 통해 손가락이나 큰 뼈들에 대한 Segmentation은 잘 수행하지만 손바닥에 있는 겹쳐있고 작은 뼈들에 대한 Dice score가 낮은 것을 확인하고 이들에 대한 추가 학습을 위해 crop한 후 Super Resolution을 하여 새로운 training set를 생성하였다. 이 training set를 학습시키고 전체 이미지에 대해 학습한 모델과 앙상블하여 최종 output을 출력하였다.

14.4.2 마주한 한계와 아쉬웠던 점

SMP 및 MMsegmentation 라이브러리 외에도 깃허브나 hugging face에서 SOTA 모델을 가져와서 테스트를 해보려 했지만 실현 되진 못하였다. MMsegmentation에 있는 모델들을 적용하는데도 상당한 시행착오가 있었으며 아이디어만 제시하고 실현하지 못한 적이 많은 것이 아쉽다. 또한 시각화를 더 디테일하게 해서 Test 이미지에 대한 분석을 하지 못해 앙상블을 하는 과정에서 어떤 클래스에 가중치를 줘야하는 지에 대한 판단을 할 수 없었던 것이 아쉬웠다.

14.4.3 한계/교훈을 바탕으로 다음 프로젝트에서 시도해보고 싶은점

아이디어를 제시하는 것도 좋지만 이들 실행할 수 있는 실천력을 키우고 싶고 다음에는 코드에 대한 이해력을 높이는데 더 많은 시간을 투자하여 구현 능력을 키우고 싶다. 또한 본인들이 작업하고 있는 것에 대한 공유가 원활하지 못했던 것 같아 다음에는 더 체계적으로 프로젝트를 진행할 수 있도록 Git 관리나 Notion을 더욱 적극적으로 사용해 보려고 한다.

14.5 성기훈

14.5.1 학습 목표 달성을 위한 노력

팀원들이 원활하게 테스트를 할 수 있도록 라이트닝 모듈 기반 베이스라인을 빠르게 작성하고, 그 후에 MMsegmentation을 사용하기 위해 코드를 수정/추가하였으며, wandb 상에서 적절하게 마스크 맵을 시각화하기 위해 코드를 작성했으며, 손목뼈만 학

습한 모델의 아이디어를 제시하고 이를 추론할 수 있게 전처리 후 후처리하여 다른 모델과 앙상블 할 수 있는 기반 코드를 작성했습니다. 코드와 라이브러리를 전반적으로 살펴보며 생기는 아이디어를 적극적으로 공유했고, 여러 부분을 실제로 적용하여 구현했습니다.

14.5.2 마주한 한계와 아쉬웠던 점

MMsegmentation을 정상적으로 다 적용 후에 적용 방법을 안내 받아 그로 인해 낭비된 시간이 아쉬웠고, 점수를 올리기 위해서 가장 중요했던 부분을 시간이 모자라 충분히 투자하지 못한 점이 아쉬웠습니다. 그리고 1주, 2주차에 너무 집중을 많이 해서 3주차에 집중력이 풀린 탓에 어떤 방법이라도 더 시도해 볼 수 있었는데 하지 못했다는 아쉬움이 남습니다. 그리고 전반적인 코드 작성에만 시간을 투자하다보니 다양한 부분들도 아예 관여하지 못한 것도 같아 아쉬움이 남습니다.

14.5.3 한계/교훈을 바탕으로 다음 프로젝트에서 시도해보고 싶은점

다음에는 코드적인 부분에 투자하는 시간을 조금 덜어내고, 거시적인 관점으로 문제를 바라봐 넓은 시야로 다양한 해결책을 제시하고 시도해보고 싶은 마음입니다.

14.6 양호철

14.6.1 학습 목표 달성을 위한 노력

라벨을 직접 수정하는 것이 필요했기 때문에 CVAT 관련 변환코드를 작성했지만 모든 데이터셋을 활용하는 것이 불가능한 코드였습니다. 조금만 생각하면 모든 데이터셋을 주고 받을 수 있는 코드를 만들 수 있었는데, 깊게 생각해보지 못해 긴 시간 지체되었고, 한 번에 완전체로 마무리하지 못해 모델 테스트, 증강 테스트 도중에 리팩토링 하게되어 테스트에도 집중하지 못했습니다. 또한 테스트 결과나 학습 데이터를 시각화하기 위한 코드를 작성했으나 제 자신이 아니면 팀원들이 바로 이해하고 사용할 수 없었습니다.

14.6.2 마주한 한계와 아쉬웠던 점

프로젝트 시작 시점에는 데이터 전처리나 시각화 관련한 내용의 구현을 도맡아 팀원들이 신경쓰지 않아도 원하는 전처리 및 시각화를 수행할 수 있게하고 싶었는데 적절하게 구현하지 못했습니다. 라이브러리에서 모델을 가져와서 사용하는 코드를 기필코 작성해보고자 했지만 건드려보지 못했습니다.

14.6.3 한계/교훈을 바탕으로 다음 프로젝트에서 시도해보고 싶은점

matplotlib에 집착하지 않고 streamlit, wandb 등의 모듈을 사용한 시각화를 구현하여 EDA 전담까지 아니더라도 이에 관련해서는

주축이 되어 팀에 도움을 주고 싶습니다. 추가로 학습 코드에도
기여를 해보고 싶습니다.

