

Hand Bone Segmentation

Wrap-up Report

Hyeonwoo Jung¹ Jaehoon Choi¹ Jiwan Park¹

Chanhyeok Leem¹ Danyou Lee¹ Changgi Min¹

1. 프로젝트 개요

1.1. 프로젝트 주제

뼈는 신체의 구조와 기능을 유지하는 데 중요한 역할을 하며, 뼈의 정확한 분할은 의료 진단과 치료 계획을 수립하는 데 필수적이다. 특히, 손뼈는 손등뼈의 겹치는 구조로 돼 있어 분할이 더욱 까다롭다.

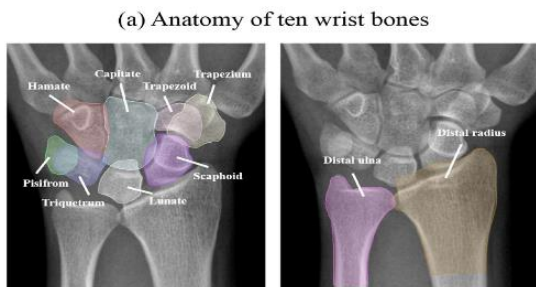


Fig 1. Hand Bone Segmentation 예시 [1]

본 프로젝트에서는 Hand bone segmentation 모델을 개발하여 의료 영상에서 손뼈를 정확히 분할함으로써 의료 진단, 수술 계획 수립, 의료 장비 제작, 의료 교육 등 다양한 의료 응용 분야에 기여하고자 한다. 딥러닝 기반의 세그멘테이션 기법은 복잡한 뼈 구조를 자동으로 분석하고 분할할 수 있어, 기존의 수작업 방식 대비 효율성과 정확도를 크게 향상할 수 있다. 이를 통해 의료 전문가들이 더 신속하고 정확한 결정을 내릴 수 있도록 돕고, 궁극적으로 환자의 치료 결과를 개선하는 데 기여할 것으로 기대된다.

1.2. 데이터 및 평가 방법

이미지 크기는 모두 2048 x 2048 로 학습 이미지는 800 장, 테스트 이미지는 288 장으로 이루어져 있다.

클래스는 손가락, 손등, 팔목 뼈로 구분된다. 손가락 관련 클래스는 f1 부터 f19 까지 총 19 개가 있으며, 손등 관련 클래스는 Trapezium, Trapezoid, Capitate, Hamate, Scaphoid, Lunate, Triquetrum, Pisiform 으로 이루어진다. 팔목 뼈는 Radius 와 Ulna 로 나뉘어, 총 29 개의 클래스로 구성된다.



Fig 2. Hand Bone Class 이미지

평가는 Dice Coefficient 를 사용했다. Dice Coefficient 는 두 집합 간의 유사성을 측정하는 지표로, 주로 분할된 이미지의 정확도를 평가할 때 사용된다. 이 지표의 값은 0 과 1 사이를 가지며 1 은 두 집합이 완벽히 일치함을, 0 은 전혀 일치하지

않음을 나타낸다.

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

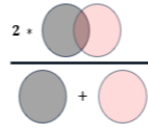


Fig 3. Dice Metric

1.3. 학습 환경 구성

GPU: Tesla V100-SXM2-32GB

CPU: Intel(R) Xeon(R) Gold 5120 CPU@ 2.20GHz

운영체제: Ubuntu 20.04.6 LTS (Focal Fossa)

2. 프로젝트 팀 구성 및 역할

정현우	<ul style="list-style-type: none"> - 전체 실험 설계 - 이상 데이터 클리닝 - Loss 식 구현 및 개선 - 최종 파이프라인 설계
최재훈	<ul style="list-style-type: none"> - Code Refactoring - Validation Strategy - Model Experiment - 최종 파이프라인 설계
박지완	<ul style="list-style-type: none"> - Yolo 초기 인사이트 제공 - nnUNet 구현 - Focal Loss 구현
임찬혁	<ul style="list-style-type: none"> - UNet++ ablation study - validation inference 결과 EDA - 후처리 기법 시도 - gamma interpolation 증강 실험
이단유	<ul style="list-style-type: none"> - 논문 사전 조사 - 모델 및 인코더 실험 - 후처리 기법 시도 - 데이터 증강 기법 실험
민창기	<ul style="list-style-type: none"> - UNet3+ 모델 구현 - 모델 예측 Crop 을 통한 훈련과 추론 파이프라인 구현 - 데이터 증강 기법 실험

3. 프로젝트 수행 절차 및 방법

내용	11월															
	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
EDA																
Data Preprocessing																
Baseline Refactoring																
Modeling																
Data Augmentation																
Loss Test																
Final merge, Ensemble																

Fig 4. 프로젝트 수행 절차 간트 차트

4. 프로젝트 수행 결과

4.2. 데이터 분석

4.2.1. 통계 분석

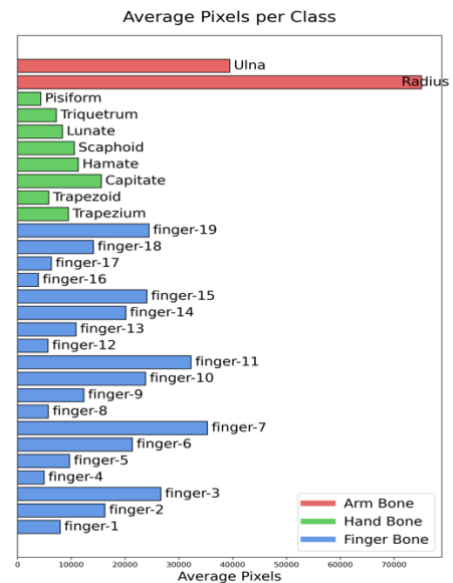


Fig 5. 클래스별 픽셀 수 평균 분포

클래스별 픽셀 수 분포를 분석하여, 각 클래스의 크기를 확인했다. 이미지 내에서 팔뼈는 큰 비중을 차지하는 반면, 손등뼈와 손가락은 상대적으로 작은 비중을 보여 작은 객체로 분류했다. 작은 객체의 예측 성능 저하 문제를 해결하기 위해 다양한 해상도로 이미지를 입력하여 작은 객체를 효과적으로 구분할 방법을 고려했다.

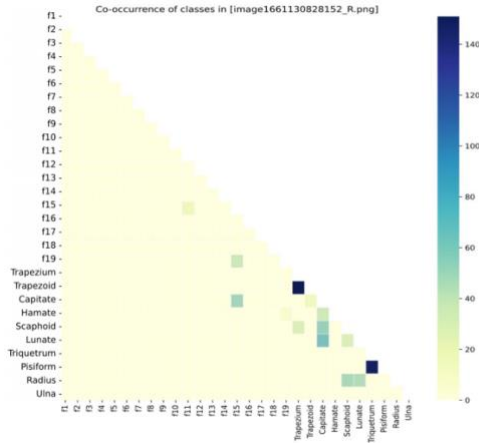


Fig 6. 각 클래스별 겹치는 영역 분포

Fig 6 을 보면 손등뼈 영역에서 객체 간 겹침이 많이 발생하는 것을 알 수 있다. 이러한 겹침 영역은 예측하기 어려운 부분으로 작용하며, multi-label segmentation task 에서 성능 저하의 원인이 된다. 이를 해결하기 위해 다양한 증강 기법을 적용하여 성능 개선을 도모했다.

4.2.2 Data Cleaning

이미지 데이터를 시각화한 결과, 몇 가지 문제점을 발견하고 이를 해결했다. 먼저, Fig 7 에서와 같이 X-ray 이미지에 반지가 포함된 사례가 관찰되었으며, 이를 노이즈로 간주하고 해당 이미지를 제거했다. 또한, 데이터의 일관성을 유지하기 위해 동일한 쌍을 이루는 이미지를 함께 삭제하였다.



Fig 7. 반지가 존재하는 (외부) 이미지 예시 [2]

훈련 데이터의 일부에서 영역 마스킹은 적절히 수행되었으나, 잘못된 라벨이 포함된 이미지를 발견했다. 이러한 오류를 올바르게 수정했고, 이를 통해 데이터의 품질을 개선하고 클렌징하는 과정을 마무리했다.

4.2.3. Stratified K Fold with MetaData

주어진 데이터와 함께 제공된 메타 데이터 (성별, 키, 몸무게, 나이)를 활용하여 EDA 를 진행한 결과, 키와 성별이 이미지 내에서 뼈가 차지하는 픽셀 수와 가장 큰 상관관계를 가지는 것으로 나타났다. 이를 바탕으로 validation set 을 구성할 때, 키와 성별에 따라 다양한 손뼈 크기를 포함하도록 조정하여 데이터 분포의 다양성을 확보했다.

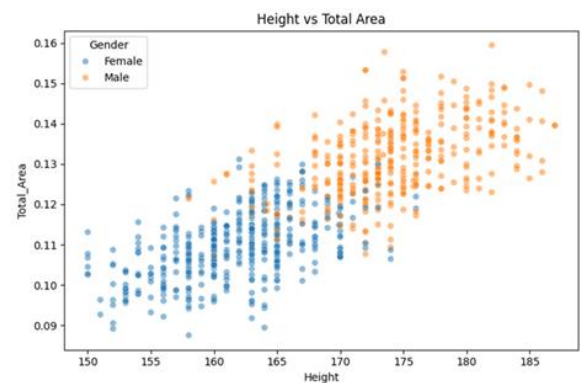


Fig 8. 키, 성별과 픽셀 분포와의 관계 그래프

Fig 8 의 EDA 결과를 반영하여 기존 group K-Fold 보다 개선된 validation 데이터 세트를 생성하기 위해 stratified group K-Fold [3] 방식을 도입하였다. 성별은 두 그룹으로, 키는 4 분위 수로 나누어 총 8 개의 그룹을 생성한 뒤, 각 그룹의 비율이 train set 과 validation set 에서 최대한 유사하게 유지되도록 데이터를 분할했다.

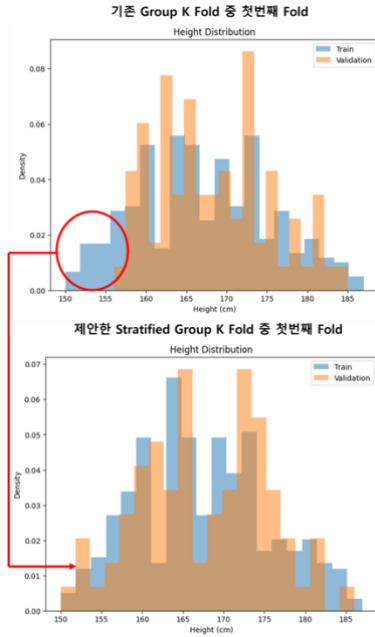


Fig 9. Validation set 분할 방식 간 키 분포 비교

Fig 9 를 보면 키 분포가 stratified group K-Fold 에서 상당히 개선된 것을 알 수 있다. 특히, 8 그룹 비율의 제공 오차 합은 기존 group K-Fold 에서 0.015 였으나, stratified group K-Fold 에서는 0.005 로 감소하며 데이터 분포 차이가 완화되었음을 확인했다.

4.3. Baseline 모델 선정 및 분석

의료 분야 segmentation 작업에서 강력한 성능을 보인다고 알려진 모델들에 대해서 실험을 진행했다. Segmentation Models PyTorch (SMP) [4] 라이브러리에서 지원하는 다양한 architecture 를 사용했으며, UNet3+ [5]와 nnUNet [6] 모델은 공식 github 를 참고하여 직접 구현하였다.

초기 실험에선 2048 x 2048 의 이미지를 image size 512 x 512 로 resize 하여 각 모델들의 test dice 점수를 평가했다. 그리고 초기 실험 성능이 좋았던 UNet++ 모델에 대해서는 추가로 encoder 와 image size 를 바꿔가며 실험을 진행했다. Image size 는 GPU 가 메모리가 허락하는 1536 x 1536 까지

resize 하여 실험했으며, encoder 는 SMP 와 Timm 라이브러리에서 제공하는 모델을 사용하였다.

Model	Encoder	Size	Dice
DeepLabV3+	Xception71	512	0.9478
	EfficientNet-B4	512	0.9443
YOLO11x		512	0.7754
nnUNet		512	0.9076
UNet	EfficientNet-B0	512	0.8808
UNet3+	HRNet-W64	512	0.9492
	HRNet-W64	1536	0.9615
UNet++	EfficientNet-B4	512	0.9443
	ResNet101	512	0.9483
		512	0.9533
	HRNet-W64	<u>1024</u>	<u>0.9692</u>
		1536	0.9708

Table 1. Baseline 모델 성능비교

Table 1 을 보면 UNet++ [7]과 HRNet-W64 [8] encoder 를 사용한 모델이 가장 높은 성능을 보이고, image size 를 키울수록 성능이 좋아짐을 알 수 있다.

최종적으로 UNet++ 모델을 baseline 모델로 선정하였으며, HRNet-W64 encoder 를 선택하였다. 빠르게 성능 관찰하기 위해 Image size 512 x 512 에 대해서 실험을 진행하고, 최종적으로 1536 x 1536 에 대해서 성능을 평가하도록 계획을 수립했다.

4.4. Data Augmentation

4.4.1. 기본 증강

의료 데이터는 데이터셋이 제한적인 경우가 많아, 데이터 증강을 통해 데이터 양을 확장하고 모델 성능을 향상시키는 접근법이 여러 연구에서 효과적으로 활용되고 있다 [9]. 본 프로젝트에서도 이러한 방법론을 적용하여 모델의 일반화 성능을

개선하고자 하였다.

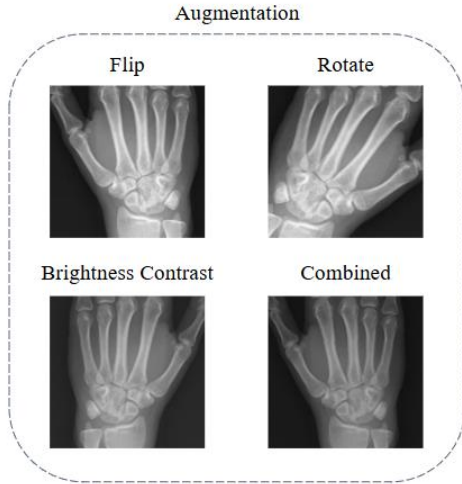


Fig 10. 기본 증강 기법 적용 예시

Train set 에서는 원본 데이터에 데이터 증강을 통해 생성된 이미지를 더하여 데이터의 양을 두 배로 확장하였다. 증강 기법은 test set 의 데이터 분포를 반영하도록 설계되었으며, 증강 데이터에는 좌우 반전을 위한 Horizontal Flip(적용 확률: 100%), Rotate(적용 확률: 80%, 적용 범위: $-15^{\circ} \sim 15^{\circ}$), Random Brightness and Contrast(적용 확률: 80%, 적용 범위: 밝기, 대비 각각 $-25^{\circ} \sim 25^{\circ}$)를 적용하였다. 이를 통해 데이터의 다양성을 확보하여 모델이 다양한 조명, 시점, 및 영상 품질에서도 강건하게 작동할 수 있도록 학습을 지원했다.

Model	Encoder	Input data	Dice
UNet++	HRNet-W64	Raw	0.9533
		Raw + Aug*	0.9550

Table 2. Input data 에 따른 성능 비교

Table 2 를 보면 train set 에 원본 이미지만을 사용한 경우보다 증강 이미지를 추가하여 데이터셋을 확장했을 때, 모델 성능이 향상되는 것을 알 수 있다. 이는 증강된 데이터가 학습 과정에서 더 다양한 특성을 반영하도록 돕고, 모델의 일반화 성능을 강화하는 데 기여했음을

보여준다.

4.4.2. Gamma Correction

기본 증강을 통해 모델의 성능을 개선한 이후, 겹치는 손목뼈 영역에서 성능을 더욱 향상시키기 위해 GC(gamma correction) 증강 기법을 적용했다. [10]

감마 보정은 의료 영상 분할 분야에서 효과적인 데이터 증강 기법으로 널리 사용되고 있으며 [11] 본 프로젝트에서도 이를 활용하여 dice 점수 향상을 도모하였다. 특히, 성능이 상대적으로 낮았던 손목뼈 영역의 검출 정확도를 높이기 위해 추가적으로 CLAHE 증강 기법[12]을 적용했다.



Fig 11. GC 및 CLAHE 증강 적용 예시

초기 실험에서는 별도의 증강 없이 ResNet101 을 encode 로 사용하는 UNet++ 모델을 학습시켰으며, 이때의 Dice 점수는 0.9483 이었다. 이는 데이터 증강 없이도 모델이 일정 수준의 학습 성능을 발휘할 수 있음을 보여주었다.

Model	Encoder	Augmentation	Dice
UNet++	ResNet101	x	0.9483
		GC + CLAHE	0.9513
		GC	0.9530

GC : Random gamma correction

Table 3. 증강 기법에 따른 성능 비교

Table 3 을 보면 gamma correction 만을 단독으로 적용한 실험에서는 $\text{gamma_limit}=(80, 200)$ 과 $p=0.3$ 의 확률로 명암을 조정한 결과 Dice 점수가 0.9530 으로 증가한 것을 알 수 있다. 하지만

CLAHE와 gamma correction을 함께 적용하여 명암 대비를 조정한 결과, dice 점수가 0.9513으로 gamma Correction만을 단독으로 적용했을 때보다 소폭 하락하였다.

이는 CLAHE 적용 과정에서 이미지 윤곽선이 과도하게 강조되며 발생한 노이즈 증가로 인해 모델 학습이 방해받았을 가능성이 있음을 시사한다.

Gamma correction 단독 사용은 데이터의 명암 분포를 효과적으로 조정하여 모델 성능을 개선하는데 기여하였다. 이후 실험에서 감마 보정의 범위를 넓게 설정하고 적용 확률을 낮추는 방식으로 추가 조정을 진행한 결과, 가장 안정적이고 효과적인 성능 향상을 확인할 수 있었다.

4.5. Loss

Hand Bone Segmentation에서 성능을 최적화하기 위해 다양한 손실 함수에 대해 실험했다.

본 실험에서는 UNet++ 모델을 사용하였으며, 입력 이미지 크기는 512 x 512로 고정하였다. 각 손실 함수가 모델 성능에 미치는 영향을 비교하고, 최적의 손실 함수를 선정하기 위해 다른 조건을 고정하고 분석을 진행했다.

Model	size	Loss	Dice
UNet++	512	BCE	0.9533
		Focal	0.9539
		BCE + IOU	0.9544
		Hybrid*	0.9548
	1536	BCE	0.9713
		BCE + IOU	0.9733

Hybrid* : Focal + MS-SSIM + IOU

Table 4. 손실함수에 따른 성능 비교

먼저, 기본적인 이진 교차 엔트로피(BCE) 손실

함수를 사용하여 실험을 진행하였다. BCE는 예측 확률과 실제 라벨 간의 픽셀별 차이를 측정하는 방식으로, 이진 분류 작업에 적합한 특성을 가지고 있다. 이 손실 함수를 사용한 결과, dice score는 0.9533으로 나타나 양호한 성능을 보였다.

다음으로, Focal Loss를 적용하였다. Focal Loss는 어려운 예측에 더 높은 가중치를 부여하는 방식으로, 주로 불균형 데이터셋에서 효과적이다. 그러나, 이 손실 함수를 적용한 결과 dice score는 0.9539로 BCE보다 미세하게 높은 수준에 그쳤다. focal loss가 어려운 예측에 집중할 수 있도록 도와주었지만, 전체적인 성능 향상에는 큰 차이를 보이지 않았다.

이후, BCE Loss와 IOU Loss를 결합한 손실 함수를 적용하였다. IOU Loss는 겹치는 구조를 다루는 데 효과적이며, 특히 손등뼈와 같은 겹치는 부분을 분할하는 데 유리할 것으로 예상되었다. 실험 결과, dice score는 0.9544로 BCE와 Focal Loss보다 높은 성능을 기록하였다. 이는 IOU Loss가 겹치는 영역을 더 잘 처리하여 성능을 향상시켰음을 보여준다.

마지막으로, Focal Loss, IOU Loss, MS-SSIM을 결합한 하이브리드 손실 함수(Hybrid Loss)를 실험에 적용하였다. MS-SSIM은 이미지의 구조적 유사도를 측정하는 손실 함수로, 시각적으로 더 자연스러운 결과를 도출할 수 있도록 돕는다[5]. 이 하이브리드 손실 함수를 적용한 결과, dice score는 0.9548로 가장 높은 성능을 기록하였다.

그러나 이 방법은 메모리 부족 문제와 학습 시간 증가를 초래하였다. 특히, 이미지 크기를 1536 x 1536으로 확대했을 때 메모리 부족 현상이

발생하였으며, MS-SSIM 이 많은 메모리와 연산 자원을 소모하는 것으로 분석되었다.

결국, 해상도를 유지하는 것이 중요한 상황에서 **BCE + IOU 조합**이 가장 안정적이고 효율적인 성능을 발휘하였다. BCE 는 초기 학습에서 수렴 속도가 빠르며, IOU 는 손등 뼈와 같은 겹치는 구조를 처리하는 데 유리한 특성을 보인다. 이러한 이유로, BCE + IOU 손실 함수 조합이 최종적으로 가장 적합한 손실 함수로 선택되었다.

4.7. 최종 모델 선정 및 분석

본 프로젝트에서는 손뼈 세그멘테이션을 위한 최적의 딥러닝 모델을 개발하고, 이를 고해상도 의료 영상에 적용하여 실용적인 결과를 도출하였다. 손뼈 구조는 복잡하고 세밀한 특징을 가지며, 이를 효과적으로 분할하기 위해 여러 모델과 학습 전략을 비교한 결과, U-Net++ 모델이 가장 우수한 성능을 보였다. 고해상도 이미지를 처리하는 과정에서 원본 해상도인 2048×2048 이미지를 1536×1536 으로 리사이즈하여 처리함으로써, 세부 정보를 충분히 유지하면서도 학습 속도와 메모리 사용량 간의 균형을 잘 맞출 수 있었다.

데이터 증강 기법으로는 원본 데이터셋에 감마 보정을 적용하여 조명 조건의 변화에 강건한 모델을 학습했다. 원본 데이터셋을 복사한 후, 감마 보정뿐만 아니라 다양한 데이터 증강 기법을 추가로 적용하여 두 개의 데이터셋을 결합하였다. 이를 통해 한 에포크에 사용되는 데이터를 두 배로 확장할 수 있었으며, 다양한 조명 조건과 영상 품질에서 모델이 강건하게 동작할 수 있도록 학습시킬 수 있었다.

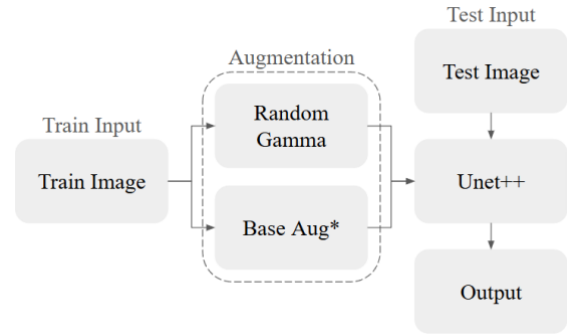


Fig 12. 최종 파이프라인

손실 함수로는 Binary Cross Entropy(BCE)와 Log IoU 를 결합하여 사용하였다. BCE 는 분할된 결과와 실제 레이블 간의 오차를 측정하는 데 유용하며, IoU 는 세그멘테이션 마스크와 실제 레이블 간의 겹치는 부분에 대한 정확도를 높이는 데 효과적이었다. 이 손실 함수 조합은 뼈 구조와 같은 복잡한 패턴을 잘 학습하는 데 적합한 성능을 보여주었다.

모델 평가 결과, 제안된 방법은 기존의 기본 U-Net 모델에 비해 고해상도 이미지를 처리할 때 더 정밀한 세그멘테이션 성능을 보였으며, 손 뼈의 주요 구조를 효과적으로 분할할 수 있음을 확인하였다.

Model	Size	Aug	Loss	Dice
UNet++ (baseline)	512	raw	BCE	0.9533
UNet++ (ours)	1536	GC + Base Aug	BCE + IOU	0.9733

Table 5. 최종 성능표

위 테이블을 보면 최종적으로 baseline 에서

우리의 증강 기법과 새로운 loss 등을 적용하였을 때, dice score 0.9533 에서 0.9733 으로 **2.10% 성능 향상**을 이뤄냈음을 알 수 있다.

5. Further study

5.2.1. Crop

메모리 제한으로 인해 원본 해상도인 2048×2048 크기의 이미지를 그대로 사용하여 학습을 진행하는 것은 불가능하였다. 이미지 해상도를 축소하거나 크기를 조정할 경우 원본 이미지의 정보가 손실되는 문제가 발생한다. 이 문제를 해결하기 위해 (a)~(c)의 방법을 활용하였다.

(a) Object Detection Train with YOLO

손 이미지를 부위별로 crop 하여 해상도 조정 없이 학습 및 추론을 수행하는 방식을 제안하였다. 손 이미지를 손가락, 손등, 팔뚝으로 구분하고, 각 부위별 X, Y 좌표의 최솟값과 최댓값을 활용하여 bounding box 라벨을 생성했다. 이를 YOLOv11x 모델에 학습시켰다.

(b) Crop Segmentation Train with UNet3+

기존 UNet++ 모델에선 전체 부위 중 손등뼈 부위가 가장 낮은 성능을 보였다. 손등뼈 부분에 대해 해당 부위만을 crop 하여 UNet3+ 모델을 훈련하여 손등뼈에 대한 segmentation 성능을 올리도록 설계했다. crop 된 이미지를 얻기 위해 라벨 데이터에서 손등뼈의 X, Y 좌표의 최솟값과 최댓값을 기반으로 중심점을 계산하였다. Inference 시 YOLO 모델의 불완전성을 고려하여 중심점에 랜덤값을 더하여 이를 보정했다. 보정된 값을 기준으로 이미지를 480×480 으로 crop 하여 손등 뼈에 대한 segmentation 학습을 진행하였다.

(c) Object Detection + Segmentation

Inference (YOLO + UNet3+)

추론 단계에서는 YOLOv11x 모델을 통해 손등뼈의 bounding box 를 탐지하고, bounding box 의 중심점을 기준으로 이미지를 480×480 로 crop 하여 UNet3+ 모델에서 추론을 수행하였다. 이후, crop 한 결과를 원본 이미지의 좌표로 변환하여 최종 결과를 생성하였다.

마지막으로, 원본 해상도를 축소하여 학습한 모델의 예측 결과 중 손등뼈 클래스의 값을 crop 기반 추론 결과로 대체하였다. (Cropped*)

Model	Encoder	Train data	Dice	
			Val (손등)	Test
UNet++	HRNet	전체	0.9497	0.9708
UNet3+	ResNet152	손등	0.9538	
Cropped*				0.9675

Cropped* : UNet++ 의 손등 클래스를 UNet3+ 모델로 대체

Table 6. Crop method 성능 비교표

위 테이블을 보면 validation dataset 기준으로 손등뼈 클래스 대한 성능 향상이 관찰되었으나, test 데이터셋 기준으로는 성능이 소폭 하락한 것을 알 수 있다.

(d) 문제 분석 및 개선사항

이미지를 Crop 하여 훈련 및 추론을 진행할 때, Crop 된 이미지에 손등뼈를 제외한 다른 부위의 뼈들도 일부 있었으나 손등뼈의 라벨정보만으로 훈련을 진행하였다.

모델에게 손등 뼈를 제외한 다른 뼈들과 배경을 구분할 정보를 제공하지 않아 모델에게 혼동을 주었다. 또한, 다른 뼈들의 정보가 없으니 그 뼈들의 변화에 모델이 적응하지 못하여 과적합 되어 validation 에서는 더 높은 점수가 나왔으나 test 에서 전체 이미지와 합산했을 때 낮은 점수를

기록한 것 같다. 따라서 손등뼈를 제외한 다른 부분의 label 을 배경으로 할당하여 학습할 시 더 나은 성능을 보일 것으로 추측한다.

5.2.2. UNet3+

SMP 에는 UNet3+가 존재하지 않아서 UNet3+의 구조를 직접 구현하였다. 또한, 해당 논문에 기술된 Hybrid Loss, DeepSupervision, CGM(classification-guided module) 등을 구현했다. Backbone 으로는 ResNet101, ResNet152, HRNet-W64 로 실험을 진행했다

Model	Encoder	Size	Dice
UNet++ (SMP)	ResNet101	512	0.9483
		512	0.9541
	HRNet-W64	<u>1024</u>	<u>0.9692</u>
		1536	0.9708
UNet3+ (구현)	ResNet101	512	0.9534
	HRNet-W64	1536	0.9615

Table 7. UNet++와 UNet3+ 성능비교표

UNet3+의 모델 구조를 다양하게 변경하며 실험을 해보았지만, UNet++의 성능이 더 좋아 UNet++을 baseline model 로 사용하게 되었다.

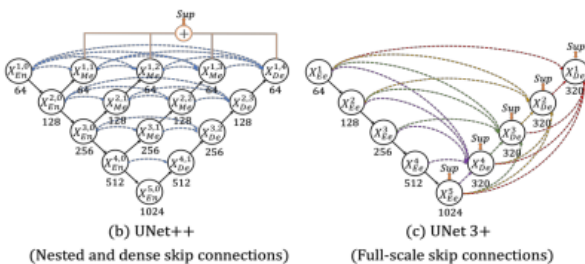


Fig 13. UNet++과 UNet3+ 모델 구조 비교 [5]

하지만, UNet3+ 모델이 UNet++ 모델의 비해 메모리적으로 경량화 되어있어, UNet3+을 활용하여 성능 향상을 도모했어도 좋았을 것 같다.

5.2.3 nnUNet

nnU-Net 는 Papers with Code 의 'Medical Segmentation Decathlon' 리더보드에서 dice 점수를 기준으로 3 위를 차지한 모델이다. 또한 2024 년 4 월까지 업데이트가 이루어진 최신 모델이다.

nnUNet 은 SMP 라이브러리에서 직접적으로 지원되지 않기 때문에 nnUNet 라이브러리를 설치하여 구현했다. 초기 구성에서는 Dice Loss 와 Cross-Entropy Loss 의 조합과 함께 2D U-Net 을 백본으로 사용했다. dice score 는 0.9072 로 UNet++보다 낮았기 때문에 baseline 모델로 활용되기엔 부족했다.

성능이 낮았던 주된 이유는 softmax 를 활성화 함수로 사용하는 multi-class segmentation 방식을 사용했기 때문이다. 이 방식은 각 픽셀이 단일 클래스에만 속할 수 있어, 각 픽셀이 여러 클래스를 가질 수 있는 multi-label segmentation task 인 이번 프로젝트에선 dice score 가 낮게 나온것으로 판단된다.

이러한 제한 사항을 해결하기 위해 활성화 함수가 softmax 에서 sigmoid 로 수정하여 multi-label segmentation 이 가능하도록 개선했다. 또한 Cross-EntropyLoss 대신 BCE Loss 를 활용 하여 각 클래스에 대한 독립적인 손실 계산이 가능해졌다.

최종적으로 성능이 개선 되었지만, UNet 을 backbone 으로 사용하는 한계점으로 UNet++보다 낮은 성능을 기록해 baseline 모델로 선정되진 못했다. 시간이 있었다면, backbone 을 최신 모델로 변경하고, 모델 구조를 개선하여 baseline 모델로 사용해도 좋았을 것 같다.

6. 자체 평가 의견

6.1. 잘했던 점들

- 데이터 클렌징
- Loss 개선
- 데이터 분석을 통해 train / valid split
- 데이터 셋에 적합한 증강 적용

6.2. 아쉬웠던 점들

- Crop 이 기대한 만큼 안 나왔다.
- 모델 개선 시도했지만 잘 안 됐다.
- 후처리 방식의 적용이 잘 안 됐다.

7. Reference

- [1] Bo-Kyeong Kang, et al., "Automatic Segmentation for Favourable Delineation of Ten Wrist Bones on Wrist Radiographs Using Convolutional Neural Network", Journal of Personalized Medicine, MDPI, 2022.
- [2] Keisuke Izumi, et al., "Ensemble detection of hand joint ankylosis and subluxation in radiographic images using deep neural networks", scientific reports, 2024.
- [3] StratifiedGroupKFold
https://scikit-learn.org/dev/modules/generated/sklearn.model_selection.StratifiedGroupKFold.html
- [4] Segmentation- Models-PyTorch(SMP)
<https://smp.readthedocs.io/en/latest/>
- [5] Huimin Huang, et al., "UNet 3+: A Full-Scale Connected UNet for Medical Image Segmentation", Computer Vision and Pattern Recognition, 2020
- [6] Fabian Isensee, et al., "nnU-Net Revisited: A Call for Rigorous Validation in 3D Medical Image Segmentation", Lecture Notes in Computer Science, 2024
- [7] Zongwei Zhou, et al., "UNet++: A Nested U-Net Architecture for Medical Image Segmentation", Computer Vision and Pattern Recognition, 2018
- [8] Jingdong Wang, et al., "Deep High-Resolution Representation Learning for Visual Recognition Publisher: IEEE Cite This PDF", IEEE, 2021
- [9] Mingyu Kim, et al., "Data Augmentation Techniques for Deep Learning-Based Medical Image Analyses", Journal of the Korean Society of Radiology, 2020
- [10] Shanto Rahman, et al., "An adaptive gamma correction for image enhancement", EURASIP Journal on Image and Video Processing, 2016
- [11] Ili Ayuni Mohd Ikhsan, et al., "An analysis of x-ray image enhancement methods for vertebral bone segmentation", IEEE, 2014
- [12] Khan, Sajid Ali, et al., "Contrast Enhancement of Low-Contrast Medical Images Using Modified Contrast Limited Adaptive Histogram Equalization", American Scientific Publishers,

1. 학습 목표 달성을 위한 노력

1.1. 학습 데이터 개선

- ✓ **문제 발견:** 학습 모니터링 과정에서 Validation 데이터셋의 값이 유독 높은 구간이 발견됨. 이를 이상하다고 판단하여 시각화한 결과, 일부 라벨링이 잘못된 데이터가 존재함을 확인함.
- ✓ **해결 방안:** 잘못 라벨링된 데이터를 수정하거나 제거하여 데이터셋의 품질을 개선함.
- ✓ **결과:** 올바른 Validation 데이터로 인해 보다 정확한 성능 평가가 가능해졌으며, 안정적인 학습을 이끌어낼 수 있었음.

1.2. Loss 식 개선

- ✓ **문제 발견:** 이미지 크기를 1536 으로 확장했을 때, **Out of Memory(OOM)** 문제가 발생하고 학습이 수렴하지 않는 현상이 관찰됨.
- ✓ **해결 방안:** 메모리 부족 문제를 해결하기 위해 BCE + IoU + MS-SSIM 에서 MS-SSIM 을 제거하고, 대신 메모리 효율성이 높은 **BCE + IoU Loss** 조합을 채택함.
- ✓ **결과:** 이미지 크기를 1536 으로 유지하면서도 안정적인 학습이 가능해졌고, Dice Score 0.9733 으로 가장 높은 성능을 달성함.

1.3. 최종 파이프라인 설계

- ✓ **문제 발견:** 최종적으로 각 실험 결과를 종합해 하나의 최적 모델을 만들어야 하는 상황이 발생함.
- ✓ **해결 방안:** 팀원들이 수행한 실험과 결과를 체계적으로 통합하여 데이터 전처리, 모델 구조, 손실 함수 등을 포함한 하나의 일관된 파이프라인을 설계하고, 다양한 조합을 비교·분석할 수 있도록 실험 환경을 표준화함.
- ✓ **결과:** 통합된 파이프라인을 통해 가장 높은 성능을 보이는 모델과 손실 함수 조합(BCE + IoU Loss)을 선정할 수 있었고, 효율적이고 안정적인 학습 과정을 확보함. 최종적으로 최고 성능의 Segmentation 모델을 완성함.

2. 마주한 한계와 아쉬웠던 점

- A. **새로운 모델 개발:** 새로운 모델을 직접 PyTorch 로 개발하려는 시도가 있었지만, 성능이 기존 모델에 비해 뒤처져 최종적으로 채택되지 못함. 이는 제한된 시간 내에 실험 및 최적화를 충분히 수행하지 못한 결과로, 모델 설계와 튜닝의 중요성을 다시 한번 느끼는 계기가 됨.
- B. **중앙 Crop 실험:** 모델이 성능이 저조했던 특정 영역을 개선하기 위해 중앙 크롭 데이터를 활용하여 집중적으로 학습시키는 방법을 시도했으나, 효과적인 성능 개선으로 이어지지 못 함.

3. 한계/교훈을 바탕으로 다음 프로젝트에서 시도해보고 싶은 점

- A. **새로운 모델 설계:** 새로운 모델 구조를 설계할 때, 최신 연구 동향을 분석하여 실험적인 아키텍처를 더 체계적으로 검증할 계획임. 이를 위해 논문 검색과 하이퍼파라미터 튜닝 도구를 적극적으로 활용해 효율적인 탐색을 시도하고자 함.
- B. **동적 Crop/중강 기법:** 특정 영역에 집중하는 데이터 기반 학습을 개선하기 위해 중앙 크롭과 같은 정적인 방법 대신, 모델이 예측에 실패하는 부분을 동적으로 식별하고 이를 학습에 반영할 수 있는 기법을 도입할 예정임.

1. 학습 목표 달성을 위한 노력

1.1 Code Refactoring

- **이유:** Segmentation-Models-PyTorch 라이브러리 기반의 베이스라인 코드를 팀원들이 더 간편하게 활용하고, 향후 유지보수를 용이하게 하기 위해 코드 리팩토링을 진행하였습니다.
- **내용:** 제공된 베이스라인 Jupyter Notebook 파일을 Python 스크립트로 변환하고, 모듈화 및 Config 파일을 도입하여 Model, Encoder, Loss, Augmentation 등 실험 요소를 손쉽게 추가 및 수정할 수 있도록 구조화하였습니다. 이를 통해 팀원 간의 협업을 효율화하고 실험의 반복성을 강화하였습니다.

1.2 Meta Data를 활용한 Stratified Group K-Fold 적용

- **가설:** Meta Data를 활용하여 Validation Dataset을 나누면 검증 성능이 개선될 것이라 가정하였습니다.
- **결과:** Meta Data에서 성별, 나이, 키, 몸무게와 각 이미지의 mask 픽셀 수 간의 상관관계를 EDA로 분석한 결과, 성별과 키가 가장 큰 상관관계를 갖는 것으로 확인되었습니다. 이를 바탕으로 성별은 두 그룹으로, 키는 4분위수로 나누어 총 8개의 그룹을 생성하였으며, 각 그룹의 비율이 Train과 Validation Set에서 최대한 유사하게 유지되도록 Stratified Group K-Fold를 적용하였습니다.

실험 결과, 기존 Group K-Fold의 8그룹 비율 제공 오차 합은 0.015였으나, 제안한 Stratified Group K-Fold에서는 **0.005**로 감소하며 데이터 분포 차이가 완화되었습니다. 이를 통해 Validation Set 분포가 Test Set과 더 유사해졌음을 확인할 수 있었습니다.

또한, 동일한 베이스라인 모델을 활용한 실험에서 제안한 Stratified Group K-Fold는 기존 Group K-Fold 대비 Val Dice와 Test Dice의 차이가 더 작아, Validation Dataset이 Test Dataset의 분포를 더 잘 반영하며 검증 성능이 개선되었음을 입증했습니다.

1.3 이미지 크기 증가를 통한 모델 성능 향상

- **가설:** Input image size를 증가시키면 작은 크기의 객체나 겹쳐있는 클래스를 더 잘 예측할 수 있을 것이라 가정하였습니다.
- **결과:** UNet++ 베이스라인 모델을 사용하여 512 크기의 이미지를 Validation Dataset에서 추론한 결과, 작은 크기의 뼈 객체들의 평균 Dice 점수가 전체 평균 Dice 점수인 0.9549보다 낮게 나타났습니다. 이에 따라 이미지 크기를 1024로 늘렸을 때, 평균 Dice 점수는 0.9706으로 **1.65%** 향상되었습니다.

세부적으로 살펴보면, 상대적으로 크거나 덜 겹치는 뼈들의 향상률은 1.51%였던 반면, 손가락 끝 뼈들의 평균 점수 향상률은 **2.59%**, 손목뼈 중 가장 겹쳐있는 뼈들의 평균 점수 향상률은 **1.88%**로 나타났습니다. 이를 통해 이미지 업스케일링이 작은 객체 및 겹쳐 있는 클래스를 보다 효과적으로 예측하는 데 기여한다는 결론을 도출하였습니다.

2. 마주한 한계와 아쉬웠던 점

- **손등뼈에 대한 추가적인 성능 개선 실패:** 이미지 업스케일링을 통해 전반적인 성능 향상을 이루었지만, 여전히 손등뼈와 같이 많이 겹쳐있는 구조에 대한 성능 개선에는 한계가 있었습니다. 이를 해결하기 위해 적합한 손실 함수 변경이나 후처리 방식을 고안하려 했으나, 자원의 제약으로 충분한 실험을 진행하지 못해 추가적인 성능 개선에는 실패했습니다.
- **라이브러리 내에서 제한된 모델 사용:** Segmentation Task에서 높은 성능을 보인 모델들 중 라이브러리에 구현되어 있지 않은 모델들이 많았으며, 이를 직접 구현하여 비교 실험을 진행하고자 했으나, 시간적 제약으로 인해 이를 수행하지 못한 점이 아쉬웠습니다.

3. 한계/교훈을 바탕으로 다음 프로젝트에서 시도해 보고 싶은 점

- **체계적인 실험 관리를 통한 자원 활용 최적화:** 위에서 언급한 시간과 자원의 부족은 체계적인 실험 관리의 미흡에서 비롯된 것으로 판단됩니다. 다음 프로젝트에서는 팀원들과의 논의를 더욱 강화하고, 협업 도구를 적극적으로 활용하여 자원의 낭비를 최소화하는 방안을 모색할 계획입니다. 이를 통해 보다 효율적이고 체계적인 실험 환경을 구축하고자 합니다.
- **오픈 소스를 활용한 모델 실험:** 오픈 소스 모델을 활용하고 커스터마이징하는 경험이 부족하다는 점을 인지하고, 다음 프로젝트에서는 문제에 적합한 모델을 직접 적용하고 수정하는 역량을 강화할 계획입니다. 이를 통해 주어진 문제에 최적화된 솔루션을 설계할 수 있는 능력을 기르고자 합니다.

1. 학습 목표 달성을 위한 노력

1.1. Yolo 구현

- ✓ **문제 발견:** yolo model 은 경량화 및 성능이 최적화 되어있고, 구현에 익숙해 초기 인사이트를 얻기 위해서 사용
- ✓ **해결 방안:** yolo 공식 사이트 성능 테이블에서 성능이 가장 좋은 YOLO11x-seg 구현
- ✓ **결과:** dice score 은 0.7754 로 smp 에서 지원하는 모델이 더 효과적이라고 판단 및 의미있는 증강기법, 부족한 부분에 대한 인사이트 제공

1.2. nnUNet 구현

- ✓ **문제 발견:** Papers with Code 의 'Medical Segmentation Decathlon' 리더보드에서 Dice 점수를 기준으로 3 위를 차지한 모델, 2024 년 4 월까지 업데이트가 이루어 비교적 최신 모델로 판단됨
- ✓ **해결 방안:** nnUNet baseline 구현
- ✓ **결과:** dice score 은 0.90 으로 좋은 성능을 보이고, 개선 가능성이 보임

1.2.1. nnUNet activation function, loss 개선

- ✓ **문제 발견:** multi-class segmentation 로 각 픽셀이 단일 클래스에만 속할 수 있어 Dice score 가 낮은 것으로 판단
- ✓ **해결 방안:** activation function 를 Softmax 에서 Sigmoid 로 변경하여 각 픽셀이 동시에 여러 클래스에 속할 수 있는 Multi-Label Segmentation 구현, CrossEntropyLoss 대신 BCEloss 가 사용되어 각 클래스에 대한 독립적인 손실 계산 구현
- ✓ **결과:** 성능은 약 4~5% 개선되었지만, smp unet++의 성능이 더 우수함

1.3. Loss 식 개선

- ✓ **문제 발견:** 기존의 BCE Loss 는 기본적인 이진 분류 손실 함수로 계산이 단순해서 Focal Loss 를 활용하고자 함
- ✓ **해결 방안:** $(1-y^{\wedge})y$ 식을 이용하여 쉬운 예측에 대한 손실 기여를 줄임
- ✓ **결과:** BCE loss 에 비해 성능은 약 1% 좋았지만, 눈에 띄게 유의미한 결과는 아니었음

2. 마주한 한계와 아쉬웠던 점

- ✓ **Backbone 수정:** Backbone 을 U-Net 에서 U-Net++로 변경하려고 할 때, 새로운 Backbone 을 라이브러리에 통합하는 과정에서 수많은 오류가 발생했다. 시간 제약으로 인해 SMP 라이브러리의 U-Net++를 사용하기로 결정했다. U-Net++을 Backbone 으로 사용했다면 더 나은 결과를 얻을 수 있었을 것으로 예상되어 시간 관리의 중요성을 느낌

3. 한계/교훈을 바탕으로 다음 프로젝트에서 시도해보고 싶은 점

- ✓ **새로운 모델 설계:** 새로운 모델 구조를 설계할 때, 최신 연구 동향을 분석하여 실험적인 아키텍처를 더 체계적으로 검증할 계획임. 이를 위해 논문 검색과 하이퍼파라미터 튜닝 도구를 적극적으로 활용해 효율적인 탐색을 시도하고자 함.

1. 학습 목표 달성을 위한 노력

이번 프로젝트의 목표는 손뼈 X-ray 사진을 가지고 뼈의 영역을 분할하는 모델의 성능을 올리는 것이었다. 프로젝트 초기 baseline 성능이 0.9703으로 매우 높았기 때문에 추가적인 성능 개선이 어려웠다. 특히 이번 segmentation task는 multi-label task로 한 픽셀에 여러 개의 클래스가 할당될 수 있었고, 이 부분이 성능 저하의 가장 큰 원인이었다.

이를 해결하기 위해 validation set에 대한 inference 결과를 시각화하고 분석했다.

지표	Ground Truth 상태	모델 예측 상태	의미
TP (초록색)	클래스 있음	같은 클래스로 예측	모델이 뼈의 크기와 종류를 정확하게 예측
FN (보라색)	클래스 있음	배경	모델이 뼈의 크기와 잘못 예측 (작게 예측)
FP (빨간색)	배경	클래스 있음	모델이 뼈의 크기를 잘못 예측 (크게 예측)
FP + FN (노란색)	클래스 있음	다른 클래스로 예측	모델이 뼈의 종류를 잘못 예측

FP는 뼈를 너무 크게 예측한 것, FN은 뼈를 너무 작게 예측한 것으로 판단하고 후처리 기법 등으로 이를 개선하고자 노력했다. 특히, 겹치는 영역에 대한 성능을 개선하기 위해 좋은 성능을 보였던 gamma correction 증강을 디테일 하게 적용해보고자 서로 다른 두 gamma correction 결과를 합한 이미지를 데이터 증강에 포함시켰지만 단일 gamma correction 과 큰 차이의 성능은 보이지 못했다.

baseline 모델이 UNet++로 확정된 이후로는 다른 조건을 고정하고 원하는 조건만을 바꾸어 성능을 평가하는 ablation study를 진행했다. UNet++에 UNet3+에서 성능이 좋다고 알려진 Hybrid Loss를 적용하여 가장 좋은 성능을 낼 수 있었지만, 최종적으로 사용한 BCE + IOU Loss와 큰 성능 차이가 나지 않는 반면에 메모리를 너무 차지하고 resize image size를 키우는데 한계가 발생해 최종적으로 적용하진 않았다.

2. 마주한 한계와 아쉬웠던 점

모델적으로 변화를 주고 싶어 여러 시도를 해봤지만 시간이 부족해 많은 시도를 못해본 것이 아쉽다. 특히 겹치는 부분의 영역을 개선하고자 다양한 후처리 기법을 적용해보았지만 기존보다 유의미한 결과를 얻지 못했다.

결국 후처리가 아닌 데이터 증강 측면에서 겹치는 부분에 대한 FP와 FN을 개선하고자 팀원이 적용했던 gamma correction 의 더 많은 경우의 gamma 값을 활용할 수 있도록 서로 다른 gamma correction 이미지를 합하여 gamma 값을 더욱더 미세하게 조정하여 성능을 올리려고 하였지만, 많은 gamma 값에 대해 실험을 해보지 못한 것이 아쉽다.

3. 한계/교훈을 바탕으로 다음 프로젝트에서 시도해보고 싶은 점

Hybrid Loss는 차이가 작지만, 그래도 가장 큰 성능을 냈던 Loss였다. 하지만 메모리 문제로 최종적으로 활용되지 못했다. 모델 구조를 개선하여 성능이 저하되지 않는 선에서 모델을 경량화하고, 제한된 메모리에 대해서 최고 성능을 낼 수 있도록 시도해보고 싶다. 또한 실험을 처음부터 설계하여 ablation study를 일찍 수행함으로써 보다 효율적으로 모델을 평가하고 개선해보고 싶다.

1. 학습 목표 달성을 위한 노력

1.1. 논문 조사

- 최적의 전처리 및 모델 설계를 위해 X-ray 영상 분할과 관련된 논문을 조사함.
- 특히, 우리 데이터셋의 특성(밝기 차이, 작은 손뼈 구조 등)에 적합한 데이터 증강 기법들을 조사하고 이를 시도함.
- 조사 결과를 바탕으로 학습 전략을 개선하고, 실험 설계에 적용하여 성능 향상에 기여하려 노력함.

1.2. 코드 리팩토링

- 코드의 가독성과 유지보수성을 높이기 위해 전체적으로 모듈화를 진행함.
- 실험 과정 관리 및 시각화를 위해 wandb를 연동하여 실험 기록을 자동화함.
- 상위 성능을 가진 3개의 체크포인트를 저장하도록 수정하여, 최적의 모델 선택 가능성을 높임.
- 학습 안정화를 위해 Early Stopping 기능을 도입하여 불필요한 자원 낭비를 줄임.

1.3. 모델 실험

- SMP 라이브러리에서 제공하는 다양한 모델(U-Net, DeepLabV3 등)과 인코더(ResNet, EfficientNet 등)를 사용하여 성능을 비교 실험함.
- 실험 결과, Baseline 성능이 이미 높게 설정되어 있어 모델 구조보다는 데이터의 전처리 및 후처리 과정이 더 큰 영향을 미친다는 것을 깨달음.
- 이를 바탕으로 모델보다는 데이터 중심적인 접근 방식으로 방향성을 조정함.

1.4. 증강 기법 실험

- 밝기와 대조를 조절하기 위해 주파수 도메인 필터링, 히스토그램 균등화, 감마 보정, CLAHE 등의 다양한 증강 기법을 시각화하고 효과를 분석함.
- 각 증강 기법의 파라미터 값을 조정하여 가장 적합한 설정을 찾고 모델 학습에 적용함.
- 데이터 증강이 모델의 성능에 미치는 영향을 체계적으로 평가하며 실험을 진행함.

1.5. 후처리

- 예측 확률 맵에서 최적의 Threshold 값을 탐색하여 Validation Set에서의 Dice Score를 기준으로 최적화를 시도함.
- Threshold 조정 외에도 추가적인 후처리 기법(예: CRF, Small Object Removal 등)을 시도하지 못한 점을 한계로 느끼며, 다음 프로젝트에서 적극적으로 탐구해볼 계획임.

2. 마주한 한계와 아쉬웠던 점

- 증강 실험:** 조사했던 다양한 증강 기법 중 일부만 실험하였으며, 모델이 데이터의 다른 측면(예: 구조적 변화, 노이즈 등)을 어떻게 받아들이는지 충분히 분석하지 못한 점이 아쉬웠음.
- 후처리:** Threshold Tuning만 진행하였으며, 복잡한 후처리 기법(예: Conditional Random Fields, Region-based Refinement)을 시도하지 못함.

3. 한계/교훈을 바탕으로 다음 프로젝트에서 시도해보고 싶은 점

- 모델 구현 강화:** 논문 조사에서 얻은 최신 아키텍처와 기법을 직접 구현하고 실험할 수 있도록 작업 계획을 세분화할 예정임.
- 더 창의적인 아이디어 발굴:** 기존 프로젝트 관련 연구 외에도 다른 연구나 일반적인 컴퓨터 비전 연구에서 사용되는 기법들을 폭넓게 조사하고, 이를 접목할 기회를 만들어 보고 싶음.

1. 학습 목표 달성을 위한 노력

이번 프로젝트는 이전에 학습했던 컴퓨터 비전(CV) 모델 파이프라인 전반을 경험하고자 설계하였음.

1.1. 모델 구현: UNet3+

- ✓ UNet3+ 모델을 구현하며 다양한 인코더를 사용할 수 있도록 레이어를 추가하고 디코더를 조정했음.
- ✓ 이를 통해 인코더-디코더 기반 모델의 구조를 깊이 이해하고 학습했음.

1.2. 모델 기반 Crop Train 및 Inference 파이프라인 구축

- ✓ 성능이 낮은 영역을 Object Detection하여 Crop한 이후 Segmentation하는 파이프라인을 설계했음.
- ✓ 두 모델(예: YOLO와 UNet3+)을 연결하는 과정에서 기존 Baseline 결과를 분석하고 데이터 기반의 가설을 세우는 능력을 키웠음.

1.3. 데이터 증강

- ✓ Test 데이터의 분포를 EDA하며 이에 맞는 데이터 증강 기법을 고민했음.
- ✓ 데이터를 증강하는 과정에서 다양한 가설을 세우고 검증하며 데이터의 특성을 더 깊이 이해했음.

1.4. 이진화를 통한 전처리 실험

- ✓ Multi-Level Otsu를 중심으로 이진화 및 다양한 전통적 CV 기법을 실험했음.
- ✓ 손뼈 이미지를 전처리하기 위해 Contour를 활용한 노이즈 제거, CLAHE 적용, Multi-Otsu Thresholding 등을 사용했음.
- ✓ 그러나 손뼈의 복잡한 구조, 이미지 조명의 다양성, 부위별 골밀도 차이로 인한 밝기 차이 등으로 인해 일관성 있는 결과를 얻지 못해 프로젝트 과정에서는 적용하지 않았음.
- ✓ 이를 통해 전처리 기법에 대한 이해와 데이터를 다각적으로 분석하는 능력을 키웠음.

1.5. 협업 도구 활용

- ✓ GitHub: 코드 공유 및 협업을 진행했음.
- ✓ Notion: 계획 수립 및 실험 결과를 공유했음.
- ✓ Discord API: 서버 사용 계획을 자동으로 공유했음.
- ✓ Wandb: 실험 기록 및 공유를 수행했음.

2. 마주한 한계와 아쉬웠던 점

- 문제 해결 과정에서의 딜레마:** 성능이 나오지 않는 상황에서 문제 해결에만 매몰되어 빠르게 전환하거나 다른 접근법을 시도하지 못했음 결과적으로 더 효율적인 방식을 모색하는 데 있어 시간이 지체되는 한계를 경험했음.
- 데이터 증강 실험 부족:** 시간 부족으로 인해 다양한 증강 기법을 충분히 실험해보지 못했음. 증강이 모델 성능에 미치는 영향을 체계적으로 분석하고 최적화하지 못한 점이 아쉬웠음.
- 전처리 기법의 일관성 부족:** 이진화 및 전통적 이미지 처리 기법에서 일관성 있는 결과를 얻지 못해, 프로젝트에서 활용하지 못한 점이 한계로 다가왔음. 전처리 과정이 오히려 모델 학습을 방해할 수 있다는 판단 아래 과감히 제외

했지만 개인적인 판단이었고, 구체적인 실험을 진행해보지 못한 점이 아쉬움.

3. 한계/교훈을 바탕으로 다음 프로젝트에서 시도해보고 싶은 점

- A. **문제 해결의 유연성 강화:** 문제점 분석과 동시에 빠르게 전환할 수 있는 실험 플랜을 미리 준비하고 실행하는 방법론을 적용하고 싶음. 실패 가능성을 염두에 둔 실험 플랜과 효율적인 자원 배분을 통해 문제 상황에 대처하고 싶음.
- B. **데이터 증강의 체계화:** 증강 기법을 사전에 카테고리화하고, 기법별로 실험을 체계적으로 설계하여 데이터 분포와 모델 성능 간의 연관성을 정량적으로 분석하고 싶음. 시간 부족 문제를 해결하기 위해 자동화된 증강 파이프라인 (AutoAugment 등)을 적극적으로 활용할 계획임.
- C. **전처리 및 분석 다양화:** 기존 전통적 CV 기법 외에 학습 기반의 전처리 기법(Segmentation, AutoEncoder 등)을 활용하여 데이터를 더 잘 이해하고 전처리 효율성을 높이하고자 함. 이진화를 포함한 전처리 기법이 학습에 미치는 영향을 체계적으로 검증할 계획임.
- D. **다중 모델 파이프라인의 고도화:** 이전 프로젝트에서 경험한 YOLO와 UNet3+ 병합 파이프라인을 발전시켜, 더 복잡한 멀티 모델 파이프라인을 설계하고 성능 분석을 심화하고 싶음. 다양한 Task 간의 연결성을 고려한 파이프라인을 통해 모델의 유연성과 확장성을 높이하고자 함.