

BoostCamp AI Tech

Project1 Wrap-Up Report

Sketch Image Classification

CV-23

김세연, 안지현, 김상유, 김태욱, 김윤서

목차

1. 프로젝트 개요

- 1.1 프로젝트 주제
- 1.2 프로젝트 데이터셋
- 1.3 프로젝트 코드 구조

2. 프로젝트 팀 구성 및 역할

3. 프로젝트 수행절차 및 방법

- 3.1 데이터 분석 및 증강
- 3.2 단일 모델 실험
- 3.3 모델 개선
- 3.4 모델 앙상블

4. 프로젝트 수행 결과

- 4.1 Data Augmentation
- 4.2 모델 실험
- 4.3 모델 평가
- 4.4 모델 앙상블
- 4.5 최종 모델 선정 및 분석

5. 자체 평가 의견

- 개인 회고

김세연

안지현

김상유

김태욱

김윤서

1. 프로젝트 개요

1.1 프로젝트 주제

Computer Vision에서는 다양한 형태의 이미지 데이터가 활용되고 있습니다. 그중 스케치는 인간의 상상력과 개념 이해를 반영하는 추상적이고 단순화된 형태의 이미지입니다.

이러한 스케치 데이터는 색상, 질감, 세부적인 형태가 비교적 결여되어 있으며, 대신 기본적인 형태와 구조에 초점을 맞춥니다.

이번 프로젝트는 이러한 스케치 데이터의 특성을 이해하고 스케치 이미지를 통해 모델이 객체의 기본적인 형태와 구조를 학습하고 인식하도록 함으로써, 일반적인 이미지 데이터와의 차이점을 이해하고 또 다른 관점에 모델 개발 역량을 높이는 데에 초점을 두었습니다.

1.2 프로젝트 데이터셋

데이터셋은 검수 및 정제된 ImageNet Sketch 데이터셋으로 이미지 수량이 많은 상위 500개의 객체로 이뤄져 있습니다.

이미지는 대부분 사람의 손으로 그려진 드로잉이나 스케치로 구성되어 있으며, 각 클래스는 객체, 동물, 인물, 풍경 등 다양한 카테고리를 가지고 있습니다

- 학습 이미지 15,021개
 - Class 개수: 500개
 - Class별 이미지 개수: 29~31개
 - 이미지 너비 및 높이: 대체로 400~800픽셀 사이에 분포
- 평가 데이터 10,014개

1.3 프로젝트 코드 구조

[Level1-Imageclassification-cv-23 GitHub](#)

2. 프로젝트 팀 구성 및 역할

팀원	역할
김세연	프로젝트 모듈화 및 모델 연구 UX, 모델, 앙상블 실험
안지현	Data Augmentation, Wandb 연동
김상유	Data preprocessing 및 모델, 앙상블 실험
김태욱	모델 학습 코드 개선, 모델, 앙상블 실험,
김윤서	Data Augmentation 및 실험 결과 시각화

3. 프로젝트 수행절차 및 방법

3.1 데이터 분석 및 증강

- 데이터 분석

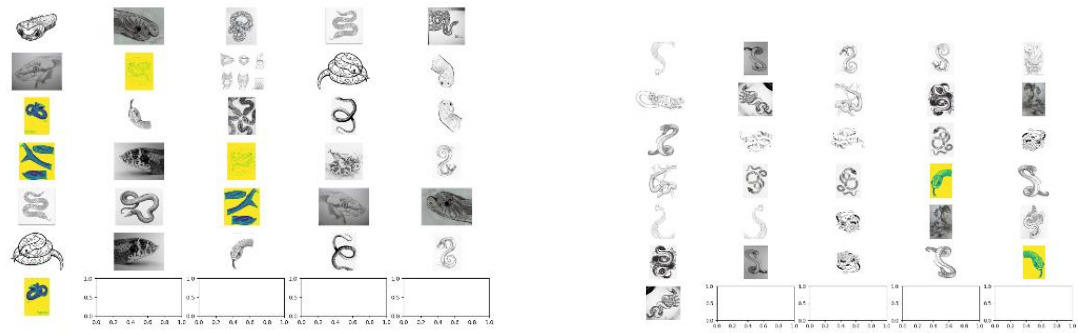
모든 이미지를 클래스별로 시각화하여 이미지의 특징을 파악해보았습니다.

- 유사한 클래스: 새, 뱀 등의 상위 카테고리로 묶을 수 있는 유사한 클래스가 다수 포함되어 있었습니다.
- 워터마크/텍스트: 워터마크나 텍스트가 객체와 겹쳐지는 경우가 있었습니다.
- 밝기: 밝기가 매우 높거나 전체적으로 어두워 경계가 잘 드러나지 않는 이미지가 있었습니다.
- 이미지의 다양성: 일부 클래스에서는 하나의 클래스에 같은 이미지가 flip된 버전으로 여러 장 포함되어 있어 이미지 분포가 다양하지 않았습니다.

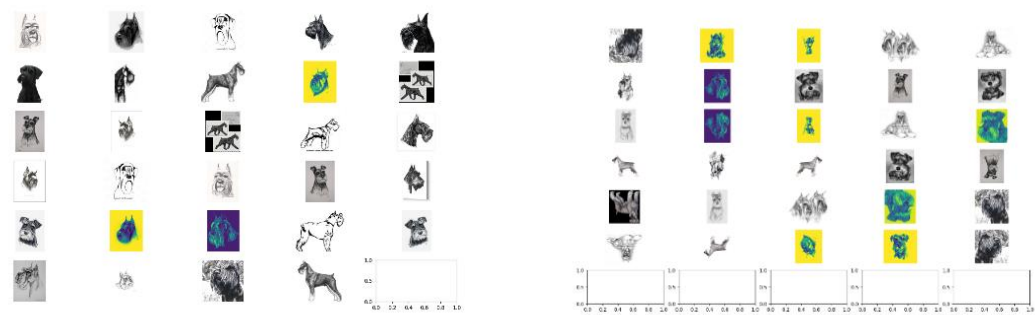
또한, 일부 이미지에는 여러 색이 포함되어 있었으나 대부분의 이미지가 흑백으로 이뤄져 있었습니다.

(아래 이미지에서 노란색 또는 보라색 배경으로 나타난 이미지들은 채널이 1개인 이미지가 기본 컬러맵 viridis를 사용한 결과입니다. 모두 흑백 이미지로 볼 수 있습니다.)

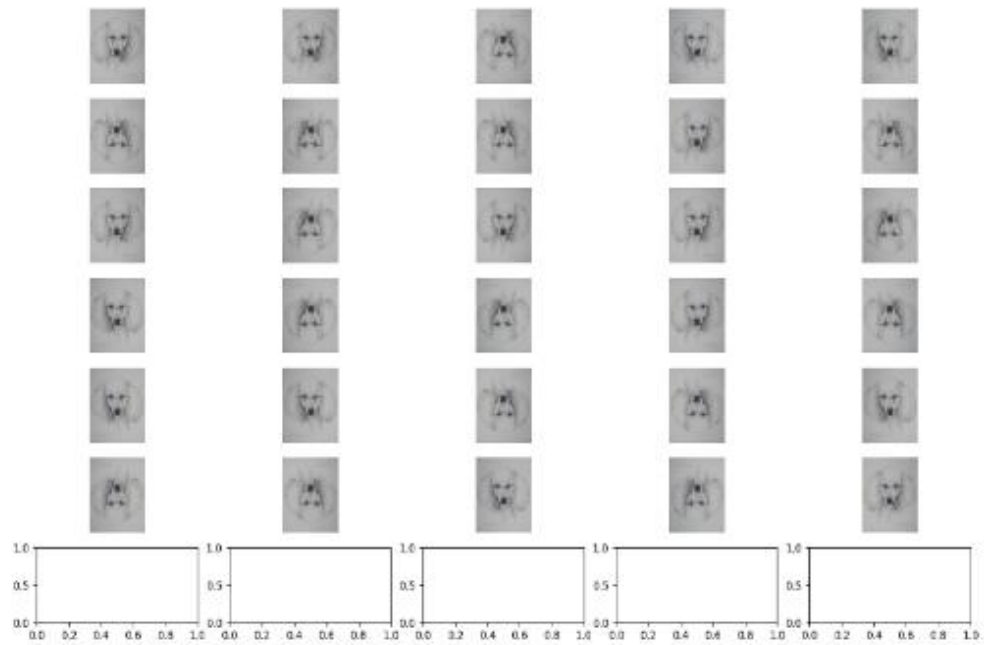
▣ 31, 32 클래스



▣ 114, 115 클래스



▣ 158 클래스



- 데이터 증강

데이터를 분석한 결과를 토대로 albumentations, torchvision 라이브러리를 이용해 다양한 데이터 증강 기법을 시도해보았습니다.

- RandAugment : 자동화된 증강 기법의 효과를 확인하고 비교해보고자 적용할 기법의 개수와 강도를 조절하며 randaugment 실험을 진행했습니다.
- AutoAugment : ImageNet 기반의 데이터이므로 ImageNet의 최적 증강 policy를 사용해 autoaugment 실험을 진행했습니다.
- Geometric transform : Horizontalflip, Rotation, Scaling, Shear과 같은 기하학적 변형을 적용해 이미지 다양성을 높였습니다.
- Morphological transform : 스케치 이미지 특성 상 선으로만 이루어진 이미지가 많고 선의 굵기가 이미지마다 다르므로 선의 굵기를 변형시킬 수 있는 erosion, dilation을 적용해보았습니다.
- Noise & Blur : 이미지에 노이즈나 손 떨림에 의한 블러가 존재할 수 있으므로 gaussian noise와 motion blur를 적용해보았습니다.
- CoarseDrop : 워터마크나 텍스트로 객체가 온전히 보이지 않는 경우를 고려하여 랜덤한 위치에 흰색 박스를 만드는 coarsedrop을 적용했습니다.
- Equalize : 이미지의 대비를 높여 경계를 선명하게 하고자 CLAHE와 같은 histogram equalize 기법을 적용해보았습니다.
- 추가적으로 데이터 전처리를 통해 성능을 실험해보았습니다.
- Resize : 가로 또는 세로로 긴 이미지가 다수 포함되어 있으므로 필요에 따라 padding을 사용해 resize 시 비율을 유지하도록 하였습니다.
- Normalize : 베이스라인에서는 ImageNet의 평균과 표준편차를 사용해 normalize 하였으나 스케치 데이터가 대체로 높은 밝기를 가지고 있음을 고려해 직접 데이터셋으로 계산한 평균과 표준편차를 사용했습니다.


3.2 단일 모델 실험

- 모델 선택 과정

Model Choose				
Model Name	Model Type	Train Loss	Validation Loss	
ResNetV2_50.a1h_in1k	timm	0.0375	0.801	0.801
ResNetV2_50	timm	0.0316	0.8267	0.8267
ResNet50	timm	0.0246	0.8343	0.8343
ResNet18 (Original model)	timm	0.0463	0.9409	0.9409
ResNet	timm	0.212	1.0431	1.0431
TResNet_M	timm	0.0716	1.0647	1.0647
TResNet_V2_L	timm	0.0311	1.0839	1.0839
DenseNet121	timm	0.0257	1.1589	1.1589
EfficientNet	timm	0.1019	1.1658	1.1658
InceptionResNetV2	timm	0.0438	1.4595	1.4595
ResNetV2_50x3_bit	timm	6.2171	6.2146	6.2146
RDNet_base	timm	100	100	100
ResNetV2_152	timm	100	100	100

- 초기 실험

초기 실험은 프로젝트 baseline code에 적용되어 있는 ResNet18을 기반으로 진행하였으며, 그 결과는 아래와 같았습니다.

resnet		0.7620 0.7670
--------	---	------------------

- 초기 실험 결과

ResNet18을 사용한 실험 결과, 훈련 데이터에 대한 Loss값은 하락하는 반면, 검증 데이터에 대해서는 값이 유지되거나 상승되는 과적합 현상이 나타났습니다. 이는 모델이 훈련 데이터에 적합하지 않다는 것을 나타내며, 추가적인 실험이 필요함을 시사합니다.



- 추가 실험

먼저, 기존 ResNet18 모델보다 높은 성능을 기대할 수 있는 다양한 모델들을 추가로 실험하였습니다. 구체적으로 ResNet50, ResNet101, 그리고 개선된 모델인 ResNetv2_50을 포함하여 실험을 진행했습니다.

추가로, ResNet 모델의 변형인 ResNeXt와 ConvNet의 재구성을 통한 ConvNext 모델과 Transformer의 인코딩 방식과 CNN의 Convolution Layer를 결합하여 성능을 극대화한 ViT, ViT-Hybrid 모델도 테스트하였습니다.

- 멀티모달 모델 및 다양한 아키텍처

특히 CLIP 모델과 유사한 방식인 EVA 모델과 다양한 네트워크 아키텍처를 쉽게 설계할 수 있는 장점을 가진 RegNetY 모델을 timm 라이브러리 내 ImageNet 데이터셋에서 사전훈련 되어 있는 모델로 실험한 결과, 아래와 같은 높은 정확도의 결과를 볼 수 있었습니다.

convnext_xxlarge		0.9080 0.9120	2024.09.24 01:51
vit_huge		0.9070 0.9020	2024.09.23 22:54

▣ 실험 결과

실험 후 [EVA02, RegNetY, ConvNext, ViT, Resnet101] 총 다섯 개의 모델을 최종적으로 선택하였습니다. 위 5개의 모델의 공통점으로는 timm 라이브러리 내에 사전학습 및 결과가 좋았던 모델 중 상위에 속하였으며, 실험 초기 과적합 문제를 어느정도 해결이 가능하다고 판단하였습니다.

3.3 모델 개선

▪ 모델 크기와 학습 소요 시간

○ 문제상황

- EVA와 같은 큰 규모의 모델을 학습시키기 위해서는 1epoch 당 최대 1시간이 넘는 시간이 소요되었습니다. 제한된 시간 안에 최대한 많은 실험을 위해서 모델의 학습 시간을 최소화할 필요가 있다고 느꼈습니다.

○ 해결방안

- Mixed Precision Training
 - 연산이 필요한 시점에서 FP16, 성능과 직접적인 연산의 FP32 사용
 - RegNetY와 EVA로 실험을 해본 결과 성능 차이가 없이 속도, 메모리 측면에서 효율적인 학습이 가능
- 빠른 학습 중단
 - 1~2 Epoch 기준으로 목표된 성능이 나오지 않을 경우 빠르게 중단

○ 기대효과

- 앙상블에 사용할 고성능 모델을 짧은 시간 안에 여러 개 학습

- **Loss Function**

- **문제상황**

- 모델 아키텍처를 제외하면 유사한 조건에서 학습하여 앙상블을 위한 학습 다양성 부족
 - 모든 모델이 특정 클래스에 대한 정확도가 낮아 앙상블의 효과 감소
 - Eva02의 경우 403 클래스를 모두 404로 예측하는 문제 발생
 - 특정 클래스에 대해 모델이 과도하게 확신할 가능성 존재
 - 이는 모델의 일반화 성능 저하 및 과적합으로 이어질 수 있음

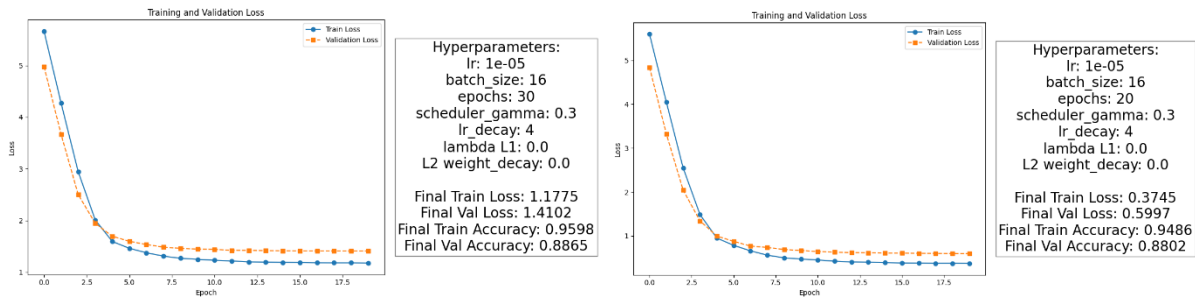
- **해결방안**

- Label Smoothing
 - Hard target을 soft target으로 바꾸어 모델의 확신 정도를 낮춤
 - 라벨 스무딩을 통해 모델이 클래스 간의 불균형 확신을 줄이고, 일반화 성능을 향상시킬 수 있다고 판단함.
 - Focal Loss
 - Easy case의 Loss를 줄이고 Hard case의 Loss의 비중을 높여 어려운 이미지에 대한 성능을 높임

- **기대효과**

- 모델의 다양성을 높여 앙상블 성능 향상
 - Hard Case에 대한 성능 향상
 - 모델의 오확신 방지
 - 과적합 방지 및 다양한 클래스에 대한 균형 잡힌 예측.

Eva 모델 기준으로 loss를 시각화했을 때,



➤ None Label Smoothing (왼쪽 이미지)

eva 모델 라벨스무딩을 하지 않았을 때, loss는 낮지만, 정확도는 낮음.

➤ Label Smoothing

eva 모델 라벨스무딩 0.1했을 때, Loss는 높지만, 정확도에서 조금의 향상을 보임.

3.4 모델 앙상블

▪ 모델 앙상블 조합

○ 문제 상황

- 비슷한 성능의 모델이 많아 앙상블 조합의 경우의 수가 너무 많음
- EVA 계열 모델은 특정 클래스에 대한 정확도가 매우 낮게 나타남
- EVA 계열의 평균 정확도는 매우 높은 편이다.

○ 해결방안

- EVA 모델이 어려워하는 클래스를 잘 예측하는 모델과 앙상블 시도
- 성능이 좋은 조합에서 모델을 하나씩 변경하며 시도

○ 기대효과

- 모델이 상호간 약점을 보완해 높은 성능 예상
- 모든 경우의 수를 다루지 않고 좋은 조합을 발견

4. 프로젝트 수행 결과

4.1 Data Augmentation

데이터 증강 효과를 확인하기 위해 다음과 같은 설정으로 고정하고 실험하였습니다.

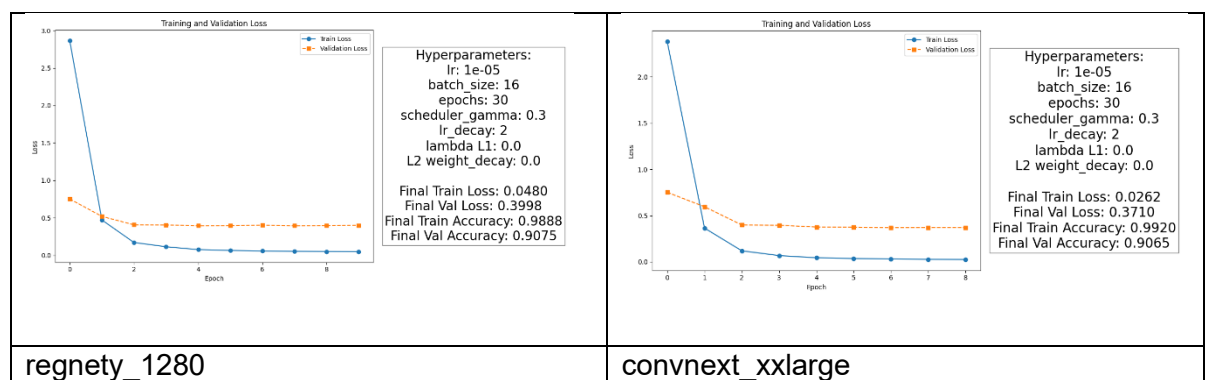
- model: resnet18 (timm)
- Epochs: 15 or 20
- Scheduler: StepLR(step_size = 8 epoch, gamma = 0.1)
- Optimizer = Adam(lr = 0.001)
- Loss = CrossEntropy
- 라이브러리: Albumentations, Torchvision

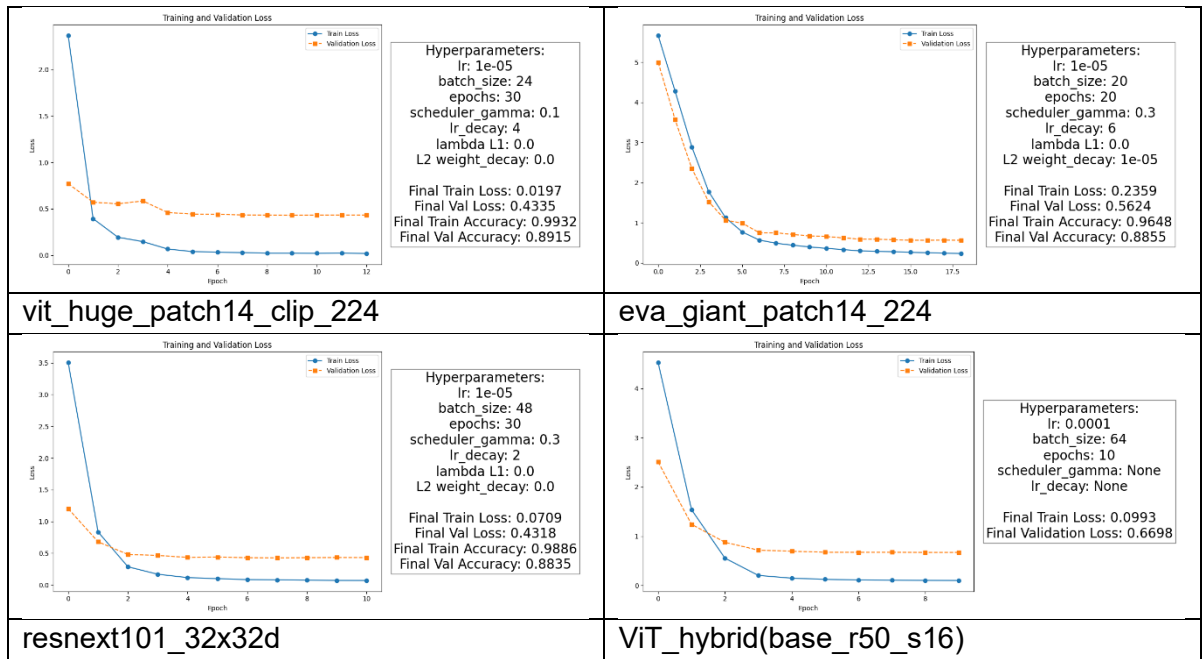
Setting	Val Loss	Val Accuracy	Time
None	1.2367	70.85%	30m
정규화 수치 변경	1.250	70.82%	30m
이미지 비율 유지(Resize)	1.2144	71.31%	32m
AutoAugment(Imagenet)	1.0408	74.31%	64m
Geo + Noise & Blur	1.017	75.81%	120m
RandAugment	1.0214	76.47%	74m
Geo + Morph + CDrop	0.948	77.34%	38m
HFlip + Rot + RBC	0.9432	77.90%	36m

모든 데이터 실험: [링크](#)

4.2 모델 실험

▪ Timm 모델





Label Smoothing + Focal Loss

모델	Val Loss	Val Acc	특징
RegNetY_1280	1.4075	89.62%	Label smoothing (0.1)
Convnext_xlarge	1.3209	90.58%	Label smoothing (0.1)
Eva_giant	0.7908	88.82%	Focal Loss, Label Smoothing (0.1)
Eva02_large	2.3547	91.05%	Focal Loss, Label Smoothing (0.3)
Eva02_large	2.3408	91.25%	Label Smoothing (0.3)

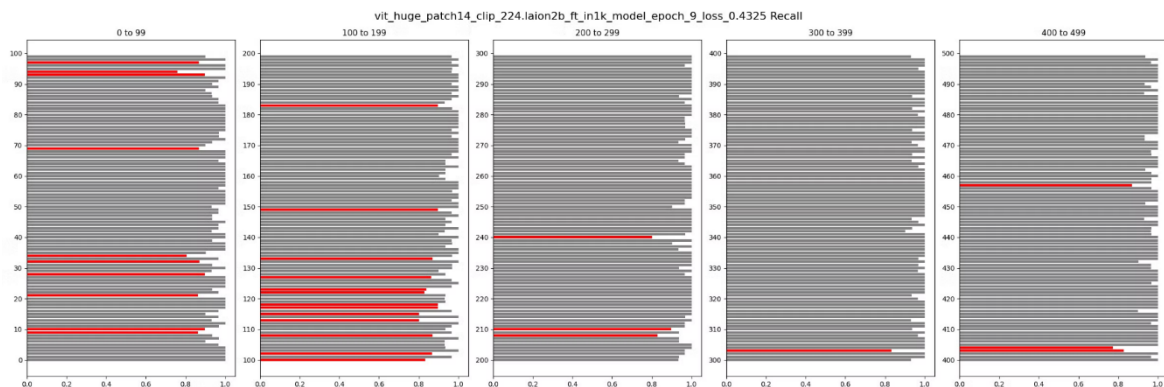
모든 모델 실험: [링크](#)

4.3 모델 평가

Recall per class

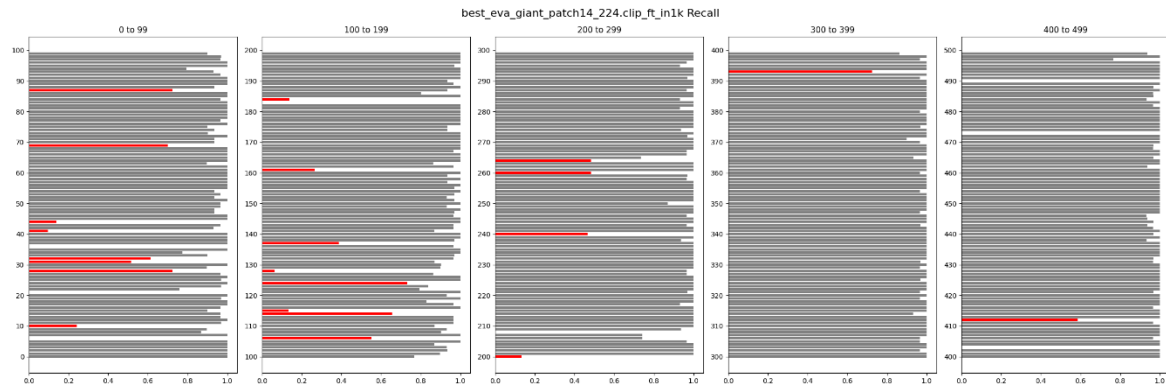
ViT huge

class	94	404	113	115	240
recall	75%	77%	80%	80%	80%



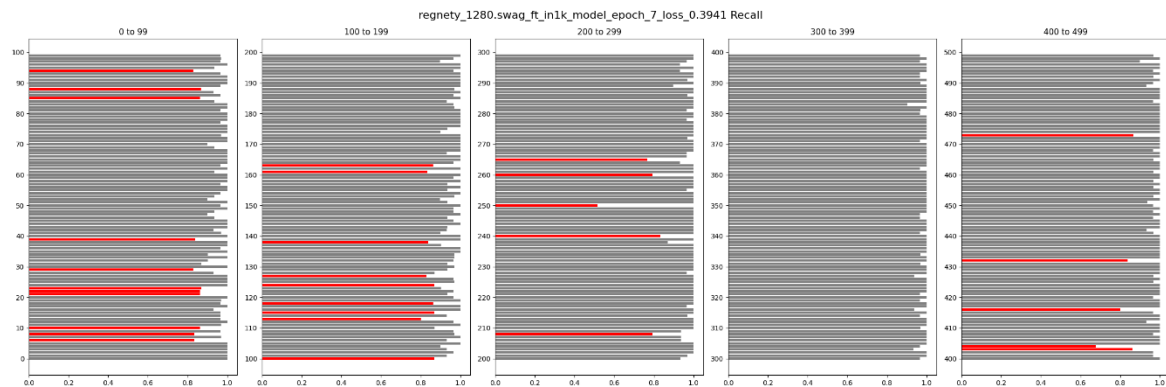
EVA_giant

class	6	21	36	183	208
recall	0%	0%	0%	0%	0%



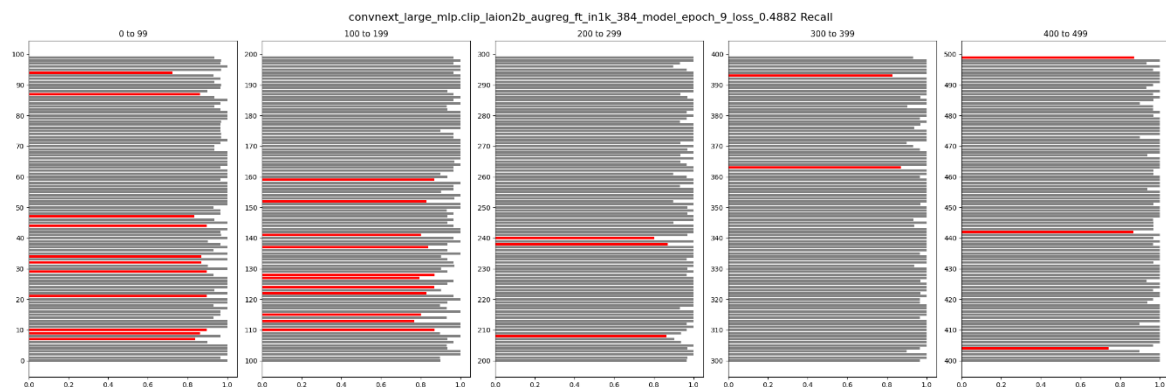
RegNetY_1280

class	250	404	265	208	260
recall	51%	67%	76%	79%	79%



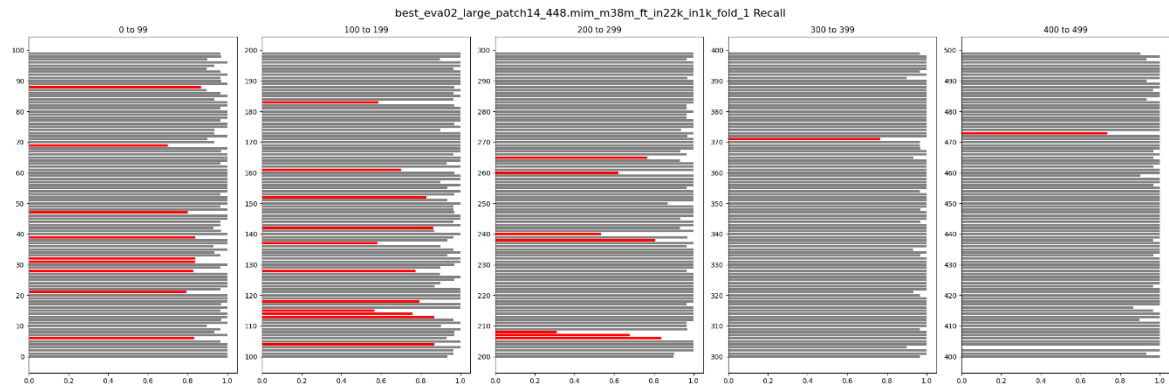
ConvNeXt_xxlarge

class	403	404	115	28	94
recall	72%	74%	76%	79%	79%



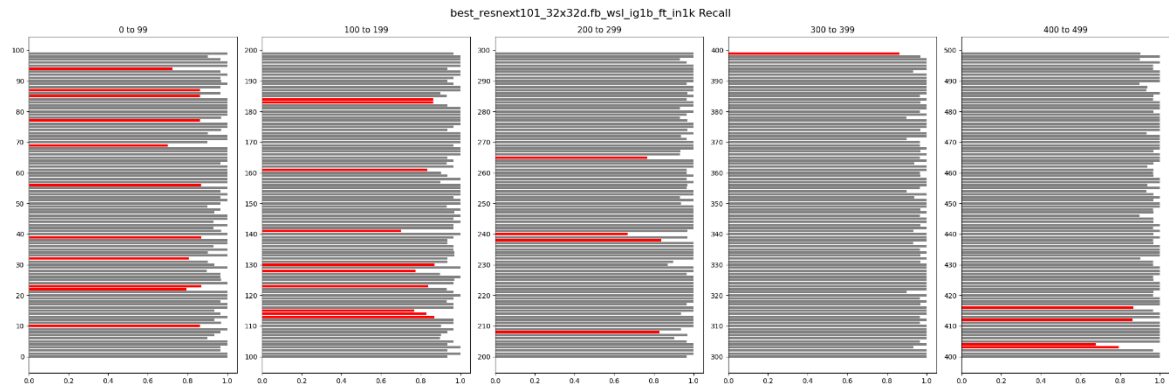
EVA02_large

class	403	208	240	115	137
recall	0%	31%	53%	56%	58%



ResNext101

class	240	404	69	141	94
recall	66%	67%	70%	70%	72%



모델 평가: [링크](#)

4.4 모델 앙상블

▪ Public Data Accuracy

앙상블 기법 - Soft Voting

0.9280 = [convnext_xxlarge, regnety_1280, resnext101, vit_huge]

+ resnet101 추가

0.9310 = [convnext_xxlarge, regnety_1280, resnext101, vit_huge, resnet101]

+ eva_giant 추가

0.9320 = [eva_giant, convnext_xxlarge, regnety_1280, resnext101, vit_huge, resnet101]

0.9340 = [eva02_large, convnext_xxlarge, regnety_1280, vit_huge]

+ eva_giant 추가

0.9370 = [eva_giant, eva02_large, convnext_xxlarge, regnety_1280, vit_huge]

+ regnety label smoothing 모델로 변경

0.9380 = [eva_giant, eva02_large, convnext_xxlarge, regnety_1280_label_smoothing, vit_huge]

+ eva02 label smoothing, focal loss 모델 추가

0.9400 = [eva_giant, eva02_large, eva02_large, convnext_xxlarge, regnety_1280, vit_huge]


+ convnext label smoothing 모델로 변경

0.9400 = [eva_giant, eva02_large, eva02_large, convnext_xxlarge, regnety_1280, vit_huge]

4.5 최종 모델 선정 및 분석

모델	Loss	Optim / Sched	Augmentation
Eva_giant		Adam / StepLR	Horizontalflip, Rotate, Randombrightnessc ontrast
Eva02_large	Focal Loss + Label smoothing(0.3)	Adam / StepLR	
Eva02_large	Label smoothing(0.3)	AdamW / StepLR	
Convnext_xxlarge	Label smoothing(0.1)	Adam / StepLR	
Regnety_1280	Label smoothing(0.1)	AdamW / CosineAnnealing	
Vit_huge	None	Adam / StepLR	

4.6 최종 리더보드

2	CV_23조		0.9370
---	--------	---	--------

5. 자체 평가 의견

1. 잘했던점

- ipynb를 py로 모듈화하고 앙상블, cross validation, loss 시각화 등의 기능을 추가하여 사용 편의성을 높인 것
- 리더보드에서 높은 성적으로 대회를 마무리한 것
- 효율적인 앙상블을 위해 클래스 별 정확도로 모델을 평가하고 분석한 것
- 실험 관리를 위해 Wandb sweep라는 새로운 플랫폼을 배운 것

2. 아쉬운 점

- 모델 실험 전 데이터 분석이 부족했던 것
- Grad-Cam이나 어텐션 매트릭스 시각화와 같은 모델 분석을 하지 않은 것
- Git에 대한 숙련도가 부족하여 코드 공유 이외의 목적으로 사용하지 못한 것

3. 개선방향

- 적극적인 Git 활용을 통한 협업 및 형상관리
- Wandb 등의 같은 딥러닝 도구를 활용한 효율적인 실험 수행
- Tmux, nohup 등의 도구로 원격 서버 활용하기

■ 개인 회고

- 나는 내 학습목표 달성을 위해 무엇을 어떻게 했는가?
- 나는 어떤 방식으로 모델을 개선했는가?
- 내가 한 행동의 결과로 어떤 지점을 달성하고, 어떤 깨달음을 얻었는가?
- 전과 비교하여 새롭게 시도한 변화는 무엇이고, 어떤 효과가 있었는가?
- 마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?
- 한계/교훈을 바탕으로 다음 프로젝트에서 시도해볼 것은 무엇인가?

김세연

1. 나는 내 학습목표 달성을 위해 무엇을 어떻게 했는가?

먼저 대회 전체 프로세스를 효율적으로 관리하고, 활용할 수 있도록 주어진 base code를 모듈화했습니다. 이 과정에서 util, model, train, inference, data 그리고 그 외 활용 source code인 src로 기능에 따라 별도의 함수와 모듈로 분리하고, argparse, shell scrip을 연결시켜 다양한 실험을 쉽게 수행할 수 있게 만들었습니다. 또한, early-stopping, loss visualization, cross validation, ensemble, L1, L2 규제 등의 기능을 구현했으며, 중복된 코드를 리팩토링하여 코드의 가독성과 효율성을 높였습니다.

2. 나는 어떤 방식으로 모델을 개선했는가?

Resnet 계열의 모델을 실험할 때, 각 클래스 별로 데이터가 30개 정도밖에 되지 않다는 것을 개선하기 위해 Stratified K-fold를 통해 성능을 개선했습니다. 이후, overfitting을 줄이기 위해 L1, L2를 구현 및 적용하였으며, eva2, convnext등 Large Model을 다룰 때, Label Smoothing 기법을 다양한 값으로 실험하였습니다. 특히, Eva Giant 모델에서는 Label Smoothing 값을 0.3으로 설정했을 때 성능이 저하되는 것을 확인하고, 최적의 값으로 0.1을 도출하여 성능을 최적화할 수 있었습니다.

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

최종 리더보드 기준 2위를 달성할 수 있었습니다. 이 과정에서 다양한 모델 실험과 하이퍼파라미터 튜닝이 성능 향상에 큰 역할을 했다는 것을 깨달았습니다. 또한, Large Model을 다룰 때, 작은 변화가 큰 영향을 미칠 수 있다는 점을 경험하며, 보다 세밀하고 근거에 기반한 튜닝의 필요성을 다시 한번 인식하게 되었습니다.

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

처음에는 CNN계열의 모델을 사용했으나 한계를 체감하고 이후에는 eva2, convnext,와 같은 Large Model을 시도했습니다, 특히, ViT 계열의 모델이 확실히 CNN 계열을 모델에 비해 saturation이 적게 일어난다는 것을 경험할 수 있었고, 정확도 0.2 이상의 더 나은 성능을 갖는 것을 확인할 수 있었습니다.

5. 마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?

우선, 명확한 목표 설정(가설 등)과 이를 뒷받침하는 근거에 기반한 실험이 부족했습니다. 특히 학습 이전의 EDA뿐만 아니라 confusion matrix 분석, 클래스별 정확도 평가 등을 통해 데이터에 대한 모델의 성능을 깊이 있게 이해하지 못한 점이 아쉬웠습니다. 또한, Git을 효과적으로 사용하기 위한 명확한 Convention을 미리 설정하지 못해 협업을 효율적으로 하지 못한 부분도 있었습니다. 마지막으로, WandB와 같은 MLOps 툴을 활용하지 않은 점도 개선이 필요한 부분으로 남았습니다.

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

데이터 분포, 모델 성능 분석 및 구조 파악 등을 통해 task에 맞는 가설을 세우고 이를 기반으로 실험을 설계해볼 것입니다. 또한, 팀 내의 Git Convention과 규칙을 사전에 마련하여 협업 효율성을 높이고 싶습니다.

안지현

1. 나는 내 학습목표를 달성하기 위해 무엇을 어떻게 했는가?

이론과 실습을 매치하는 것이 나의 이번 학습 목표였고, 먼저 eda와 baseline 코드를 돌려보면서 해당 프로젝트에 대한 파악을 우선 했으며, 학습목표를 이루기 위해 해당 코드 분석을 시작했다. 또한 내가 생각했던 방법론을 실제 코드로 구현하기 위해 학습 스케줄러, 데이터 증강 기법들을 사용할 수 있는 다양한 파이토치 클래스와 함수에 대해 공식 문서에서 찾아보고 각각 함수에 들어가는 arguments와 output값들의 형식에 대해 공부했다.

2. 나는 어떤 방식으로 모델을 개선했는가?

데이터 증강에 대해 신경을 썼고, AutoAugment 클래스를 사용했다. 또한 학습을 조금 더 효율적으로 하기 위해 WandB 를 프로젝트에 적용하여 하이퍼파라미터 튜닝을 자동화하기 위한 코드를 작성하였다.

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

코드 구현은 했지만 해당 코드가 잘 돌아가지는 않았고 효율적이지 않은 프로그램으로 오히려 학습이 느리게 진행되었다. 따라서 어떻게 해야 툴을 올바르게 사용할 수 있을지 많은 학습이 필요하다는 것을 느꼈다.

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

새로운 플랫폼(툴)의 사용을 직접 공식문서에서 배워 추가해보았다. 실제로 프로젝트에 도움이 되지는 않았지만 개인적으로는 많은 공부를 했었다.

5. 마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?

- git flow, jira등 다양한 협업 툴을 사용하지 못한 것.
- 가설 설정 등 실험 초기에 어떠한 세팅도 하지 않고 모델부터 돌려본 점
- Mlflow 등 효율적인 학습을 위한 실험 관리를 제대로 해보지 않은 점.

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

- 협업 관련 세팅을 먼저 하고 프로젝트를 진행할 것.
- 베이스라인과 eda를 통한 분석으로 가설 설정부터 할 것.
- 강의를 먼저 듣고 코드를 리뷰할 것.
- Mlflow를 사용하여 중앙 관리 식으로 실험관리를 해볼 것.

김상유

1. 나는 내 학습목표를 달성하기 위해 무엇을 어떻게 했는가?

데이터 분석 - 데이터 증강- 모델 학습- 모델 평가 - 모델 앙상블의 순서로 대회 진행에 필요한 파이프라인을 모두 경험해보고 리더보드에서 순위권 성적을 목표로 삼았습니다. 3주 간의 프로젝트 기간동안 Week1: 데이터 분석 + 증강, Week2: 모델 학습, Week3: 모델 평가 및 앙상블로 계획을 정해 모든 파이프라인을 경험해 보았으며 마지막 주차에 순위권에 진입하기 위해 수많은 조합의 앙상블을 시도했습니다.

2. 나는 어떤 방식으로 모델을 개선했는가?

단일 모델의 한계를 느끼고 앙상블에서 최고의 성능을 도출하기 위해 모델의 학습 다양성을 최대화하는 방식으로 모델을 개선했습니다. 데이터 전처리에서 의도적으로 노이즈를 추가하고 손실함수에 Label Smoothing, Focal Loss를 사용하여 모델의 다양성을 추가하였으며 Mixed Precision Training을 통해 학습 속도를 개선하여 많은 실험이 가능하게 하였습니다.

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

3주간의 노력의 결실로 최종 리더보드 2위를 달성하였습니다. 처음 대회를 시작할 당시 "순위권을 달성할 수 있을까?" 라는 의문점을 모두가 가지고 시작하였으나 포기하지 않고 마지막까지 최선을 다하면 불가능은 없다고 느꼈습니다. 이번 성과를 통해 앞으로 어떤 도전을 하더라도 겁먹지 않고 참여할 자신감을 얻었습니다.

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

이전 대회나 프로젝트에서 대부분 사전 학습된 모델이 아닌 직접 모델을 설계하고 데이터를 수집하여 학습시킨 경우가 많았습니다. 이번 대회에서는 스케치 이미지 분류라는 비교적 간단한 Task를 수행하면서 timm의 사전학습된 모델을 많이 사용해 보았던 점이 새롭게 시도한 변화입니다. 이미지넷 등의 대규모 데이터로 학습된 모델을 사용하자 직접 학습한 모델과 비교하여 매우 높은 성능을 보여줬고 다른 Task에서의 활용도 또한 높게 느껴집니다.

5. 마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?

- 부족한 학습 시간으로 더 많은 시도를 못해본 점
- Git을 적극적으로 활용하지 못한 점
- Tmux, nohup과 같은 세션 관리 도구를 사용하지 않은 점
- Wandb, MLFlow와 같은 딥러닝 도구를 활용하지 않은 점
- 실험 결과를 한 곳에 모으지 않고 분산시킨 점

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

이번에는 Git에 익숙하지 않아 효율적인 형상 관리에 어려움을 겪었지만 다음 번에는 Issue, branch를 적극 활용해보고자 합니다. 단순히 파이썬 코드만을 이용해 모듈화 시키는 것에 그치지 않고 Wandb와 같은 딥러닝 도구를 활용해서 효율적인 실험 및 기록을 수행하고자 합니다. 마지막으로 팀원의 코드나 결과를 함께 리뷰하며 서로 성장할 수 있는 피드백을 주고 받는 프로젝트로 만들어 나가고 싶습니다.

김태욱

1. 나는 내 학습목표 달성을 위해 무엇을 어떻게 했는가?

첫 프로젝트로서 아직 CV에 대한 지식이 충분하지 않아 다양한 모델을 학습시키면서 모델의 이해도를 쌓는 방향으로 목표를 두었습니다.

2. 나는 어떤 방식으로 모델을 개선했는가?

Baseline code를 살펴보고 모델 선택 과정에 따른 정확도 상승 영향이 높다는 것을 파악 후 ResNet부터 Eva 모델까지 여러 모델을 돌려보고 최적의 하이퍼파라미터 값을 찾는 작업을 주로 진행하였습니다. 이후 팀원의 조언으로 정규화 와 Label Smooting 등의 추가 실험을 통해 높은 정확도의 모델을 찾아 정확도 개선을 위해 노력했습니다.

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

프로젝트 중 많은 모델들을 실험하며 경험을 쌓고, 팀원들의 조언을 들으며 모델 구조에 대한 코드의 이해도가 많이 상승된 것을 체감했습니다. 이전의 경우는 강의를 듣더라도 이해가 되지 않았던 부분이 많았지만, 현재는 프로젝트 진행으로 얻은 지식을 활용하여 쉽게 이해가 가능한 만큼 큰 도움이 되었던 것 같습니다.

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

지난 과제나 개인 학습으로 모델을 사용했던 경우는 Original 모델 (학습되지 않은)을 사용하여, 이번 프로젝트에서도 실험 중 시도해보았지만 정확도가 낮고 해결할 방법을 찾지 못했습니다. 이후 멘토님의 조언을 통해 사전 학습된 timm 라이브러리의 대형 모델을 처음으로 사용해 보았고 위 정확도보다 높은 결과를 가져오게 되었습니다.

5. 마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?

아쉬웠던 점은 코드 내부의 손실함수, 정규화 등 세부 적인 코드 보완을 어렵다는 이유로 시도를 해보지 않았습니다. 이외 시각화 기법 등 여러 부분에서 팀내 작업 기여가 낮았던 것 같아 아쉬웠습니다.

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

이번 프로젝트를 계기로 배웠던 기법 및 구조를 다음 프로젝트에서 사용해보고 마주했던 한계를 해결하면서 동시에, 팀원에게 좀 더 도움이 될 방법과 편의성에 해당하는 코드 및 기능을 학습 및 사용 해 볼 생각입니다.

김윤서

1. 나는 내 학습목표 달성을 위해 무엇을 어떻게 했는가?

이번 팀 프로젝트에서는 데이터 분석과 시각화를 바탕으로 모델 성능을 개선하고 전체적인 프로세스를 이해하는 것을 목표로 하였습니다. 우선, 데이터 증강 파트를 맡아 '이미지 확인 - 데이터 증강 기법 시각화 - 가설에 따른 기법 선택 - 데이터 증강'의 순서로 여러 가지 조합을 실험해보았습니다. 또한, 이번 기회에 전체 딥러닝 프로세스를 이해하고자 베이스라인 코드를 꼼꼼히 살펴보고 모델 평가를 위한 코드를 작성하여 이해한 내용을 적용해보았습니다.

2. 나는 어떤 방식으로 모델을 개선했는가?

Torchvision과 Albumentations의 여러 데이터 증강 기법을 시도해보고 성능이 좋았던 조합을 단일 모델 실험 결과와 결합하여 성능을 개선하고자 하였습니다. 또한, 앙상블 시각 모델이 어떤 클래스를 잘 예측하는지 확인하고 서로 다른 클래스를 잘 맞추는 모델들을 결합할 수 있도록 클래스 별 재현율을 출력하고 시각화 하였습니다.

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

여러 가지 증강 기법들을 시도해보았지만, baseline의 조합은 넘어서지 못했습니다. 데이터 증강 시 더 면밀한 탐구와 구체적인 가설, 보다 다양한 접근법이 필요하다는 점을 느낄 수 있었습니다. 또한, 500개 클래스의 재현율을 시각화하는 과정에서 보기 쉬운 시각화에 대해 고민해보고 개선 방법에 대해 배울 수 있었습니다.

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

이전에는 코드를 이해하기보다는 무작정 사용만 하다 보니 딥러닝 프로세스에 대한 이해가 부족했는데, 이번에는 baseline 코드와 팀원이 모듈화한 코드에 대해 공부해보며 일련의 과정이 어떻게 이뤄지는지, 어떻게 모듈화를 하고 모듈을 활용할 수 있는지를 배울 수 있었습니다.

5. 마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?

아쉬웠던 점은 코드 내부의 손실함수, 정규화 등 세부 적인 코드 보완을 어렵다는 이유로 시도를 해보지 않았습니다. 이외 시각화 기법 등 여러 부분에서 팀내 작업 기여가 낮았던 것 같아 아쉬웠습니다.

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

이번 프로젝트를 계기로 배웠던 기법 및 구조를 다음 프로젝트에서 사용해보고 마주 했던 한계를 해결하면서 동시에, 팀원에게 좀 더 도움이 될 방법과 편의성에 해당하는 코드 및 기능을 학습 및 사용 해 볼 생각입니다.