

1. 프로젝트 개요

A. 개요

마스크 착용여부를 판별하는 모델을 만드는 것이 이번 대회 목표입니다. 아시아인 남녀로 구성된 4,500 명의 사진으로 한 사람당 사진의 개수는 7 개로 (마스크 착용 5 장, 이상하게 착용 1 장, 미착용 1 장) 총 31,500 개의 데이터 셋이 제공됩니다. 이러한 데이터셋 이용하여 모델을 학습하고, 성별, 나이를 기준으로 총 18 개의 클래스를 분류하는 Task 를 수행합니다.

B. 환경

컴퓨팅 환경: 개인 V100 서버에서 jupyter 및 Terminal 사용

협업 환경: Notion, GitHub, Wandb

의사 소통: 카카오톡, Zoom, Slack

2. 프로젝트 팀 구성 및 역할

팀원	역할
전체	Wrap up report 작성, 실험 관리
박승희	팀내 의견 수집 및 의견 정리, 각 팀원 코드 모음 및 팀 실험 코드 관리, Age 모델 실험
김용우	code debugging, code implementation, outlier remove, weight normalization, Mask+Gender model experiment
이다현	모델 리서치, data augmentation, 모델 성능 실험, Age 모델 실험
이상민	EDA, 데이터 시각화, 모델 앙상블, 병렬모델 실험, upsampling 실험
이윤표	data augmentation, lr, scheduler, optimizer, loss, 모델 성능 실험, wandb 연동, Mask+gender 모델 실험

3. 프로젝트 수행 절차 및 방법

A. 팀 목표 설정

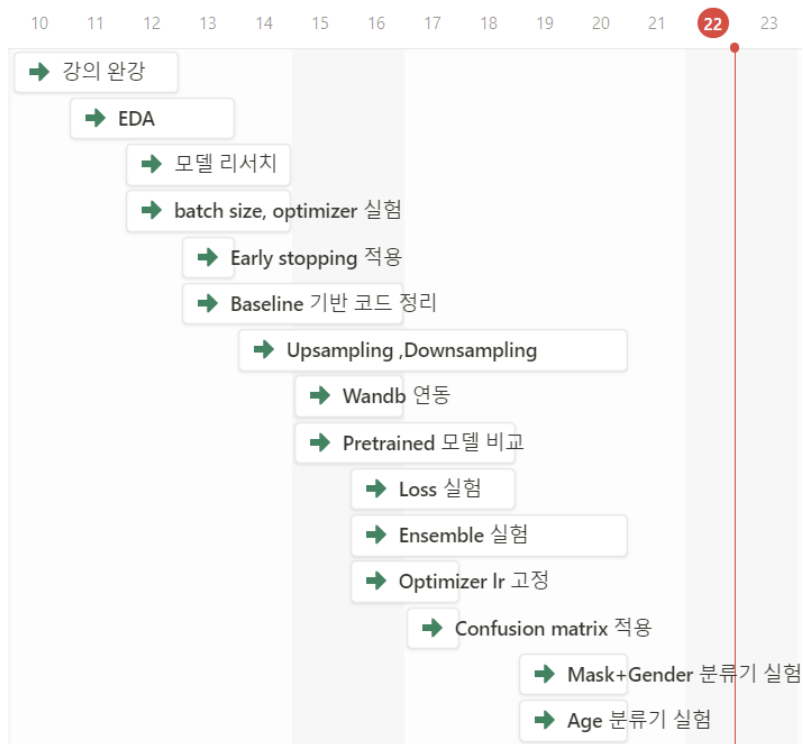
(4/10 ~ 4/12) 강의 전부 듣기, 데이터 분석

(4/13 ~ 4/17) 다양한 모델 성능 비교 및 baseline 코드 분석, 협업 툴 사용

(4/18 ~ 4/20) Ensemble 시도 및 모델 성능 평가 지표를 통한 분석

B. 프로젝트 진행 과정

(1) Timeline



(2) 협업 방식

- [1] 각 날짜마다 어떤 실험을 했고 어떤 결과가 있었는지 정리하여 Notion 페이지에 작성
- [2] 실험을 할 때 Wandb에 누가했는지 기록하도록 하여 서로의 실험 결과를 쉽게 비교
- [3] 각자 아침마다 자신이 진행한 실험과 결과에 대해서 이야기해보는 시간을 가짐
- [4] 공동 Github에 메인 branch 이외 팀원 별 하나의 branch를 생성하고 각자 실험해본 뒤, 팀원들과 상의 후 Merge
- [5] 소스코드와 일정을 관리하는 PM을 정하여 팀원들이 원활하게 소통할 수 있도록 도움

4. 프로젝트 수행 결과

4-1. 탐색적 분석 및 전처리

EDA(Exploratory Data Analysis)를 수행한 결과, 데이터의 클래스 간 분포가 불균형하다는 사실을 확인할 수 있었다. 특정 클래스에 대한 데이터가 과도하게 많거나 적은 경우가 존재했다. 성별과 마스크 착용 여부와 같은 클래스들은 불균형한 분포를 보였지만, 나이 클래스에서는 이러한 불균형이 더욱 심각한 것으로 나타난 것을 EDA를 통해 확인하였다.

1) 데이터 Augmentation

데이터 불균형은 모델의 학습과 일반화 능력에 부정적인 영향을 미칠 수 있으므로, 적절한 전처리 기법을 적용하여 불균형을 보완하는 것이 중요하다. 이를 위해 다양한 방법이 존재하는데, 예를 들어 적은 클래스에 대한 데이터를 추가 생성하는 Over-sampling 기법이 있으며, 많은 클래스에 대한 데이터를 제거하는 Under-sampling 기법도 적용할 수 있다.

데이터 불균형을 보완하는 작업은 모델의 성능을 높이고 Overfitting을 방지하며, 일반화 능력을 향상시키는 데 큰 도움이 될 것이라고 가정하고, Data Augmentation을 진행하였다.

대회 데이터셋에 이를 적용시켜 18개의 레이블 중 Over-sampling할 label을 선택하고, 다양한 data Augmentation 기법들을 사용하여 Over-sampling을 진행했다.

사용한 augmentation 기법

- RandomHorizontalFlip: 임의의 확률로 좌우반전
- ColorJitter: 밝기와 색도 같은 요소를 무작위로 조절
- RandomErasing: 임의의 확률로 data의 일부를 가림
- RandomGaussianBlur: 임의의 확률로 가우시안 필터를 사용해 blur기법을 적용

Under-sampling은 높은 비율을 차지하던 클래스의 데이터 셋을 줄이는 방법으로 데이터 불균형을 해소하지만, 학습에 사용되는 데이터 수가 줄어들기 때문에 학습의 성능에 영향이 갈 수 있다. 본 대회의 학습 데이터셋은 18900장으로 적은 수이기 때문에 이를 유의하여 적용해주었다.

다음과 같은 기법을 통해 특정 label에 데이터 Augmentation을 적용한 결과 미세한 성능 향상을 볼 수 있었다.

2) Transforms 기법

- CenterCrop(380,380): 사람의 얼굴을 집중적으로 학습하기 위해 이미지 중앙부분을 Crop
- ColorJitter: 색상과 밝기를 조절하여 마스크를 선명하게 하고, 주름을 선명하게 보이도록 처리

- RandomHorizontalFlip: 이미지 데이터의 가로 방향을 무작위로 뒤집음
- RandomAdjustSharpness: 선명도를 조절하여 주름이 더 잘 보일 수 있도록 조절

Sharpness을 제외했을 때 (bestAugmentation)가 추가했을 때(realAugmentation) 보다 f1 score가 0.001정도 더 증가함 18개의 classes에 적용해서 그런지 마스크와 성별에는 안 좋은 영향을 주었을 것으로 예상된다.

4-2. 모델 개요

18개의 클래스를 분류하는 하나의 Task에서 나이를 잘 분류하지 못하는 경향을 확인했다. 이를 해결하기 위해 18개의 클래스를 분류하는 하나의 Task에서, Age, Mask & Gender를 분류하는 두개의 Task로 나누어 Age model의 성능을 끌어올리기 위해 이러한 모델을 선정했다

(1): Age Model

<i>Model</i>	EfficientNet-B3
<i>Data preprocessing</i>	CenterCrop, ColorJitter, RandomHorizontalFlip, Normalize
<i>Input-size</i>	3X380X380
<i>Output-size</i>	3
<i>Optimizer</i>	Adam

(2): Mask & Gender Model

<i>Model</i>	EfficientNet-B3
<i>Data preprocessing</i>	CenterCrop, ColorJitter, RandomHorizontalFlip, Normalize
<i>Input-size</i>	3X380X380
<i>Output-size</i>	6
<i>Optimizer</i>	AdamW

4-3. 모델 선정 및 분석 & 평가 및 개선

초반에는 어떤 모델이 dataset에 적절할지 판단하기 위해 VGG19, ResNet34, DenseNet, ResNext 등의 모델을 train dataset으로 학습시키고 valid dataset으로 평가했다. 이 과정에서 pretrained된 모델이 그렇지 않은 모델에 비해 성능이 좋다는 것을 파악했다. 따라서 어떤 모델을 사용하더라도 pretrained된 모델을 사용하기로 했다.

팀원들이 기본적인 baseline에서 각각의 모델을 학습하여 나온 평가 결과는 다음과 같았다.

Resnet -> VGG19-> ResNext 순

김용우_T5044	0.6947	76.1746
이윤표_T5148	0.3298	39.6984
이다현_T5137	0.6471	73.3651

VGG19는 학습한 결과 과적합이 심했고, 파라미터가 많아 batch 사이즈를 키울 수 없어 후보에서 제외하기로 했다. ResNet34가 성능이 좋은 이유는 다른 network보다 적은 파라미터를 가지고 있기 때문에 과적합에 robust하기 때문이라고 판단했다. 이 과정에서 우리에게 주어진 task에서는 과적합을 방지하는 것이 중요하다고 생각했다.

과적합을 줄이는 것이 목적이라면 적은 parameter를 가진 모델을 사용하면 과적합을 방지할 수 있을 것이라고 판단하여 적은 parameter를 가져 과적합에 robust한 EfficientNet을 사용해보기로 했다. EfficientNet을 적용한 결과 ResNet34보다 valid dataset에서 f1-score가 더 높다는 것을 확인했다. 따라서 우리는 최종적으로 EfficientNet-b3를 backbone network로 선정하기로 했다.

이윤표_T5148	0.7289	77.0000
-----------	--------	---------

<efficientnetb3 train 결과>

주어진 dataset은 class 분포가 불균형 했기 때문에 loss를 선정할 때 class의 불균형을 해결할 수 있는 focal loss를 사용해보기로 했다. 또, 그 중에서도 기존 baseline에 있던 softmax로 구성된 focal loss가 아닌 cross-entropy기반의 focal loss를 사용하여 과적합 문제도 해결하고자 했다. 18개의 class를 한 번에 분류하는 모델에서는 focal loss를 사용하였고, 병렬 모델에서는 cross-entropy의 valid dataset과 test dataset에 대한 f1 score가 높았기 때문에 cross-entropy를 loss로 선정하였다.

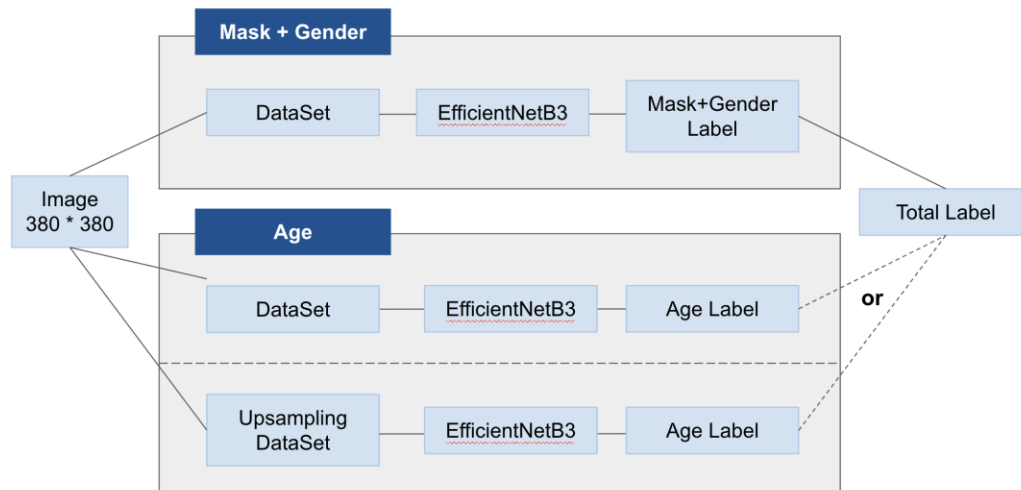
loss 비교(class 18)	f1 score
efficientNetB3 + cross entropy	0.7358
efficientNetB3 + focal loss (cross entropy)	0.7353
efficientNetB3 + focal loss (softmax)	0.7289
efficientNetB3 + label smooth	0.6603

데이터 불균형을 해소해주는 label smooth, focal loss를 사용하였으나 focal loss가 성능이 더 좋았으며 특히 cross entropy를 활용한 focal loss가 성능이 높았다. 결론적으로 cross entropy가 다른 loss에 비해 성능이 더 좋았다.

또한 노인에 가까운 중년과 중년에 가까운 노인의 차이가 불분명하여 노인과 중년의 decision boundary를 중년 쪽으로 이동시키면 중년과 노인의 경계에 있는 이미지들을 노인층으로 판단하여 모델의 성능이 향상될 수 있다고 생각하였다. 따라서 연령대를 나누는 기준을 60에서 59로 바

꾸는 작업도 추가되었다.

4-4. 최종 모델



병렬 모델	f1 score
마스크 분류기(tochvision-efficientNetB3, AdamW, focal loss(cross entropy), lr : 1e-5) 성별 분류기 (tochvision-efficientNetB3, AdamW, binary cross entropy, lr : 1e-5) 나이 분류기 (efficientnet_pytorch-efficientNetB3, Adam, cross entropy, lr : 1e-5)	0.7626
마스크&성별 분류기(efficientnet_pytorch-efficientNetB3(weight initialization), cross entropy, AdamW, lr : 1e-5) 나이 분류기 (efficientnet_pytorch-efficientNetB3,cross entropy, Adam, lr : 1e-5)	0.7698 (최종제출)
마스크&성별 분류기(efficientnet_pytorch-efficientNetB3(weight initialization), cross entropy, AdamW, lr : 1e-5) 나이 분류기 (efficientnet_pytorch-efficientNetB3,cross entropy, Adam, lr : 1e-5) [남자 노인 데이터셋 data upsampling(class 2->800, class 8->500, class 14->500)]	0.7609 (최종제출)

bestAugmentation을 적용하였으며, 각 분류기마다 같은 efficientNet을 사용해도 가중치 함수를 어떻게 가져오냐에 따라 성능 차이가 있어 분류기 별로 가장 성능이 좋은 efficientNet을 사용하였다. 또한 마스크와 성별을 구분해서 분류기를 생성하는 것과 마스크와 성별을 함께 구분하는 분류기의 성능 차이가 미비해 자원 리소스를 줄이기 위해 최종적으로 마스크와 성별을 합쳐 2개 분류기를 활용하였다.

Public 3등, mIoU 0.7698 => Private 4등 mIoU 0.7826

3 (-)	CV_17조		0.7698	80.6508	82	2d
4 (1 ▾)	CV_17조		0.7570	80.2063	82	2d

우리는 마스크 착용 여부/성별/연령 별 Data imbalance, 모델 선택, 과적합방지를 3가지 주된 key로 가정하여 크게 Data 측면과 모델 측면으로 접근하였고 wandb와 confusion matrix를 통해 타당성을 검증하였다.

1. 연령별 Data imbalance가 심각하였으며 특히 EDA를 통해 데이터 분포를 확인한 결과 남성 노인층의 data가 부족하여 upsampling 해주었을 때 실제로 성능향상을 확인할 수 있었다. Confusion matrix를 활용하여 다양한 upsampling을 적용하고 실험해 보았으나 유의미한 성능향상은 확인하지 못했고 심지어 성능이 감소하는 것을 확인할 수 있었다.
2. 병렬모델로 일반화 성능은 끌어올렸으나 최적의 일반화 모델은 찾지 못했다. train dataset를 학습했을 때보다 test dataset의 f1 score, accuracy가 약 20%씩 하락하는 것을 통해 trainset와 testset의 데이터 분포가 다르다는 것을 확인하였고, 일반화성능을 높이기 위해 과적합을 막는 방법론을 찾는 것이 중요하다고 판단하여 pretrained된 모델과 다양한 방법론을 적용하였다.

최종적으로 EfficientNetB3를 선정하고, Mask+Gender, Age를 분류하는 병렬모델을 통해 좋은 결과를 거둘 수 있었다.

5. 자체 평가 의견

잘한 점들

- ✓ 다같이 EDA를 진행하여 데이터 분석을 진행한 뒤 대회를 시작하였다
- ✓ baseline의 구조를 기초로 코드를 추가하여 5명의 팀원이 일관된 코드로 실험을 진행하였다.
- ✓ 짧은 시간이었으나 대회 기간 동안 최다 제출하여 실험에 대한 많은 insight를 얻었다
- ✓ 대회 기간동안 최고의 집중력과 열정을 가지고 팀원들과 협업하였다.
- ✓ 18개의 class를 분류하는 단일모델, (Mask, Gender, Age)를 분류하는 3개의 병렬 모델, Mask+Gender, Age를 분류하는 2개의 병렬 모델을 모든 팀원들이 효율적으로 나누어서 진행하였다

시도했으나 잘 되지 않았던 것들

- ✓ 과적합, data imbalance 문제 등을 해결하려고 많은 노력을 하였지만 모델의 일반화 성능을 최대치로 끌어올리지 못했다.
- ✓ Github 협업을 하기 위해 노력하였으나 한 branch 간 다수의 코드 push & pull, 여러 branch 간 merge가 원활하지 않아 아쉬웠다
- ✓ 초반에 각자의 환경이 달라서 실험 환경을 맞추느라 시간을 소비하였다. 초반부터 동일한 환경에서 실험했으면 더 좋았을 것 같다.

아쉬웠던 점들

- ✓ 프로젝트 기간이 촉박하여 다양한 아이디어 수행 및 철저한 변인 통제를 이루지 못했다.
- ✓ 앙상블을 적용하고 싶었으나 단일모델, 병렬모델등 앙상블에 적용할 다양한 모델의 결과가 부족하여 앙상블을 시도하지 못했다

프로젝트를 통해 배운 점 또는 시사점

- ✓ 객체지향적으로 작성된 baseline 기초를 기반으로 최종 코드를 완성함으로써 객체 지향 프로그래밍에 대한 이해도가 높아졌다
- ✓ 프로젝트를 수행하며 문제해결 능력과 협업 능력이 향상되었다.
- ✓ wandb를 사용하여 팀원들과 협업할 수 있었다.

6. 개인 회고

<이윤표>

학습목표를 달성하기 위해 한 점 / 모델 개선 방법

처음에는 주어진 데이터 분석을 통해 성별 데이터의 분포와 연령대에 따른 분포가 불균형 하다는 것을 알게 되었다. 본격적으로 프로젝트에 들어가기 전, 팀원들이 각자 다른 모델을 학습하며 성능을 비교해보았는데 나는 VGG19 pretrained 모델을 사용해보았다. Vgg19는 깊은 신경망으로 구성되어 있어 이미지 분류 작업에서 높은 정확도를 보일 것이라고 생각했기 때문이다. 하지만 학습결과 과적합이 매우 심했고, 약143M개의 파라미터를 가진 무거운 모델이어서 batch 사이즈를 키워서 학습하기가 힘들었다. 그래서 다음날, ResNet34로 모델을 바꾸어 보았다. ResNet34가 vgg보다 더 깊은 네트워크를 가지면서 더 적은 파라미터를 가지고 있기 때문에 학습 속도가 빠르고, 더 높은 일반화 능력을 가질 것이라고 생각했다. 실제로 팀원들끼리 모델을 비교하여 제출한 결과 ResNet34가 가장 높은 성능을 보이는 것을 확인했다. 따라서 모델을 resnet34로 고정시키고 augmentation 여부와 optimizer, batch size를 바꿔가며 다양한 실험을 진행하였다. 실험 결과 batch size: 128, optimizer: Adam, loss: cross entropy가 valid accuracy가 가장 높다는 결과를 얻을 수 있었다. 하지만 ResNet은 여전히 과적합이 심했고, 내가 아직 efficientnet을 시도해보지 않았다는

efficientnet_b3 기반 실험	f1 score
병렬 모델	
마스크, 성별 분류기 (cross entropy, AdamW) + 나이 분류기 (Adam, cross entropy)	0.7698
Transforms 비교 (AdamW 사용)	
efficientnet_b3 + cross entropy + transforms (CenterCrop, ColorJitter - 마스크와 주름 선명하게, RandomHorizontalFlip, Normalize)	0.7385
efficientnet_b3 + cross entropy + transforms (위의 transforms에서 RandomAdjustSharpness만 추가)	0.7377
efficientnet_b3 + cross_entropy + transforms (CenterCrop만 적용)	0.7358
Loss 비교 (Adam 사용)	
efficientnet_b3 + focal loss (cross entropy 기반)	0.7353
efficientnet_b3 + focal loss (softmax 기반)	0.7289
efficientnet_b3 + cross_entropy with labeling smooth loss	0.7178
efficientnet_b3 + labeling smooth	0.6603

것을 떠올렸다. Efficientnet이 imagenet 데이터셋을 사용하였기 때문에 다른 대표적인 딥러닝 모델들과 비교하여 top1~top5 정확도를 보여주고, 높은 이미지 해상도인 320x320을 사용하여 pretrain하였으며 내가 돌린 모델은 (380,380)으로 Centercrop 해주었기 때문에 가장 적합하다고 생각했다. Efficientnet으로 학습해본 결과 다른 모델들보다 높은 성능을 가진다는 것을 확인할 수 있었고, 이후 efficientnet_B3 기반의 다양한 실험을 진행하였다.

softmax 기반의 focal loss보다 cross entropy 기반의 loss가 결과가 조금 더 좋았다. 하지만 결론적으로는 cross entropy loss를 사용한 모델이 가장 성능이 좋아서 cross entropy를 사용하기로 했다. Cross entropy loss가 Focal loss보다 다중 클래스 분류에 조금 더 효과적으로 작동하기 때문에 이러한 결과가 나왔던 것 같다. Optimizer의 lr을 1e-3에서 1e-5로 바꾸어 준 점도 과적합에 가장 효과가 있었던 것 같다.

최종적으로는 Age의 불균형이 가장 심했고 18개의 classes에 대해서 한 번에 학습시키기보다는 마스크와 성별을 함께 학습시키고 Age를 따로 학습시키는 것이 더 효과적일 것이라고 생각하여 병렬 모델로 실험한 결과 성공적인 성과를 거둘 수 있었던 것 같다.

마주한 한계, 아쉬웠던 점

가장 아쉬웠던 점은 시간 부족과 다양한 모델 결과가 부족하여 앙상블 모델을 시도해보지 못했다는 점이다. 리더보드의 점수에 신경 쓰고 현재 가장 성능이 좋은 모델에서 큰 변화를 시도하기 두려워 새로운 시도를 하지 못해서 아쉽다. 두번째로는 k-fold 교차검증을 시도해보고 싶었으나 시도하지 못해서 내가 실제로 돌렸을 때의 accuracy와 리더보드의 accuracy 사이의 간극을 더욱 줄이지 못해 아쉽고 이외에도 sampler, 결과 분석, Mixed precision training 등 다양한 학습 기법을 사용해보지 못해서 아쉬웠던 것 같다.

다음 프로젝트에서 시도해보고 싶은 점

우선 팀원들 간의 협업 체계를 초반부터 정해 놓고 더 체계적으로 협업해보고 싶다. 또한, 오피스아워에서 알려주신 mixed precision training도 적용해보고 앙상블 모델도 시도해보고 싶다. 마지막으로 이번 프로젝트를 진행하면서 클린코드의 중요성을 깨달았기 때문에 다음 프로젝트에서는 남들이 이해하기 쉽도록 클린코드 작성을 위해 노력하고 싶다.

<박승희>

나는 내 학습목표를 달성하기 위해 무엇을 어떻게 했는가?

이번 팀프로젝트를 하기전에 데이콘 팀프로젝트를 한적이 있었는데 그때 주피터 노트북으로 모델 학습을 시키는 것이 아닌 파이썬 프로젝트 템플릿으로 작업하거나, 깃으로 코드 수정 등 현업의 방식과 유사하게 협업을 하고 싶었으나 팀원들 간의 기술 이해도 차이로 하지 못한 점이 아쉬웠고 그런 협업 방식을 주도하고 싶었습니다. 이번 프로젝트에서는 파이썬 프로젝트 템플릿과 깃허브, 완디비 등으로 협업을 이루기 위해 주도적으로 프로젝트 템플릿 형식으로 작성된 베이스라인 코드 위에 팀원들의 코드를 적용하고 깃을 통해 코드 수정 및 업데이트를 이루도록 유도하였습니다. 또한 팀원들도 객체 지향적으로 쓰여진 코드에 대한 이해와 깃에 대한 이해도가 높아 이러한 협업방식을 가능하게 해주었습니다.

추가적으로 짧은 시간내 최선의 성과를 내기 위해 팀워크를 위해 각자 무엇을 하고 있는지, 무엇을 할 건지 이야기 하는 시간을 주었고, 모든 팀원이 동등한 발언시간을 갖도록 유도해 한 사람의 치우쳐진 의견이 아닌 다양한 의견을 가질 수 있도록 노력하였습니다. 또한 팀원간 서로의 실험이 공유될 수 있도록 실험 공유 시간과 완디비 도입을 하는 등 노력하였으며 수행 전략 및 실험 방향성을 설정하는 역할을 하였습니다.

이러한 노력으로 끈끈한 팀워크와 적극적 협업을 이루어서 제가 달성하고 싶었던 목표를 달성해100% 만족스럽습니다.

마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?

다만 아쉬웠던 점은 개인적으로 협업을 위해 팀원들의 코드를 베이스라인코드에 적용, 팀 메인 브랜치 운영과 팀원들의 의견 수렴과 프로젝트 수행 전략 수립을 위해 커뮤니케이션에 많은 시간을 할애했기 때문에 적극적인 자료 수집 및 다양한 실험이 부족하다는 점이 아쉬웠으며, 스스로 팀을 이끌면서 다양한 실험을 수행하기엔 체력적으로, 능력적으로 아직 부족하다는 생각이 들었습니다. 또한 초반에 깃을 통한 팀의 실험 베이스 코드 공유 등 협업 프로세스가 구축되지 않아 팀원들이 초반에 했던 실험결과들을 객관적으로 비교하기가 어려워 팀원의 실험 아이디어들이 객관적으로 좋은 지 판단하는 게 어려웠습니다.

한계/교훈을 바탕으로 다음 프로젝트에서 시도해보고 싶은 점은 무엇인가?

다음 프로젝트에서도 높은 팀워크와, 높은 팀 수행력을 위해 협업프로세스 구축 및 적극적 커뮤니케이션을 할 것이며, 기존의 경험을 토대로 협력을 위한 시간을 단축시켜 자료 수집 및 실험 시간에 분배해 더 적극적인 실험 아이디어 제시와 실험 수행을 할 것입니다.

또한 팀원들간 실험의 객관적인 비교를 위해 시작부터 공통의 실험 환경을 갖춰 버려지는 시간, 실험 없이 주어진 시간, 서버 자원의 활용도를 최대한 높일 것입니다.

어떤 방식으로 모델을 개선했는가?

팀원들이 여러 아이디어와 실험을 보조하고 여러 실험으로 증명된 아이디어를 합쳐 더 좋은 모델이 나오도록 노력하였습니다.

단일 분류기 모델로 18개 label을 분류하는 실험에서는 기존 팀원이 실험에서 사용했던 모델을 가중치 초기화하고 실험하거나, loss함수, 옵티마이저를 변경하는 등 기존 팀원의 실험에서 더 좋은 결과를 위한 실험을 하였고, 문제를 병렬 모델로 나누어서 Mask, Gender, Age의 3개의 분류기를 가져가는 실험에서는 Age를 분류하는 모델을 맡아 그 전 실험에서 가장 좋았던 모델을 기초로 loss함수, 옵티마이저, Data Augmentation을 변경해 실험을 진행하였습니다.

직접 실험을 수행하는 것 이외로 각 팀원의 실험을 파악하고, 다른 팀원에게 공유함으로써 한 팀원이 실행 중인 실험에 대한 다양한 의견을 수집하고 중복되는 실험이 없도록 노력하였습니다.

내가 해본 시도 중 어떠한 실패를 경험했는가? 실패의 과정에서 어떠한 교훈을 얻었는가?

초반에 각 팀원들이 주피터 노트북으로 실행한 실험들을 검토하고 팀 공동의 학습 코드에서 동작될 수 있도록 재구현하는 작업을 많이 하였는데, 재현하는 시간을 너무 할애했 스스로 다양한 실험을 하는 시간이 줄어든 경험을 하였습니다.

다음 프로젝트에서는 팀원의 결과를 믿고, 재구현하는 시간을 줄여야 하며, 팀원 간 실험을 객관적으로 비교하기 위해, 실험 시작부터 통일된 모델 학습 코드, 팀 베이스 코드 구축을 실행을 해야함을 깨달았습니다.

또한 철저한 실험 변인 통제를 하려 노력하였으나 짧은 시간 내 높은 성능의 모델을 얻고 싶은 마음에 하나의 변인만 통제하는 것이 아닌 다수의 변인을 통제하게 되었고, 원하는 정도의 성능을 가진 모델을 얻었으나 어떤 변인이 얼마나 성능 향상에 기여했는지 설명하는 것이 힘들어졌습니다. 이 경험을 통해 성능 향상을 위한 실험도 중요하지만 어떤 변인이 어느 정도의 성능 향상을 기여하는지 확인하는 실험도 가져가야 타인에게 논리적인 실험 설명이 가능하다는 것을 깨달았습니다.

협업과정에서 잘된 점/아쉬웠던 점은 어떤 점이 있는가?

협업측면에서 잘된 점은 잡담을 통해 높은 친밀감을 가지고 팀원 간 동등한 의견 발언시간을 가져 높은 커뮤니케이션을 가능하게 했다는 점입니다.

반대로 아쉬웠던 점은 초반 팀 베이스 코드를 구축하지 못해 팀원의 실험을 객관적인 실험 조건에서 비교하지 못했고, 시간 부족으로 철저한 실험 변인 통제를 못했다는 점입니다.

<김용우>

나는 내 학습목표 달성을 위해 무엇을 어떻게 했는가?

나의 학습목표는 대회에서 제시한 task 를 정확하게 이해하고, 그 task 를 해결하기 위해 강의에서 배운 다양한 방법론을 적용하고, 그 과정에서 협업에 대한 이해도를 높이는 것이었다.

이러한 학습목표를 달성하기 위해 먼저 개인적으로 간단하게 모델을 만들어보며 task 에 대한 감을 익혔고, 이후에는 팀원들과 소통하며 업무를 분배하여 task 를 해결하기 위한 협업을 진행하였다. 성능 개선을 위해서 어떻게 할 것인지를 스스로 고민하고 팀원들이 성능 개선을 위해 제시한 방법들을 모두 최대한 이해하려고 노력했다. 또, 서툴더라도 git, wandb, notion 등의 협업 툴을 최대한 활용하고자 했다.

나는 어떤 방식으로 모델을 개선했는가?

강의에서 배웠던 방식을 다양하게 활용하여 모델을 개선하고 싶었다. 기본적인 backbone network 선정부터 outlier 를 제거하는 등의 심화적인 내용을 적용해보고, 성능에 좋은 영향이 있는 방법론들을 추려냈다. 내가 팀에서 맡은 task 가 리서치보다는 코드를 구현하는 데에 집중되어 있어 해보고 싶던 방법론들을 모두 적용하지는 못했지만, 팀원들이 제시한 방법론들을 직접 구현해보면서 해당 방법론이 왜 모델 성능 향상에 도움이 되는지, 어떤 방식으로 적용이 되는지, 실제로 모델 성능 향상에 도움이 되었는지를 생각해보았다.

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

대회 중 얻었던 가장 큰 지식은 객체 지향에 대한 이해와 협업에 대한 이해라고 할 수 있다. 객체 지향이 무엇인지 이해가 되지 않았었는데 대회를 진행하면서 객체 지향의 큰 틀이 이해가 되었다. 또, 대회가 협업을 어떻게 진행하면 좋을지, 협업 툴을 어떻게 활용해야 할지에 대해 생각하는 계기가 되었다. 아무래도 첫 대회가 때문에 협업을 완벽하게 할 순 없었다고 생각한다. 그래서 다음 대회를 위해 성공적인 협업은 무엇인가에 대해 고찰했다. 내가 생각하는 이상적인 대회에서의 협업을 간략하게 말하자면 다음과 같다.

먼저 task 에 대한 이해도를 높이기 위해 팀원 각자 task 를 간략하게 수행한다. 그 과정에서 얻은 인사이트를 바탕으로 git 등을 활용하여 객체 지향 방식으로 자신이 원하는 코드를 구현하여 팀원들과 코드를 합병한다. 그 다음 통제 변인과 조작 변인을 명확하게 설정하여 실험을 진행한다. 실험에서 얻은 인사이트를 통해 또 적용해보고 싶은 방법론은 팀원들과 상의하여 스스로 구현하여 코드에 추가하고, 팀원들에게 왜 그러한 방법론을 사용했는지 등을 설명한다. 이러한 방법을 사용하면 변인 통제가 명확하게 됨으로써 실험 효율이 높아질 것이고, 팀원들 모두 어떠한 방법론을 공부하는 데에 더 도움이 될 것이라고 기대된다. 이외에도 모델 성능 향상에 어떤 방법론이 좋은지에 대한 직관 등을 얻을 수 있었다.

전과 비교하여 새롭게 시도한 변화는 무엇이고, 어떤 효과가 있었는가?

전에는 모델을 만들 때 주피터 노트북을 활용했지만, 이번 대회에서는 python 파일을 활용했다. 이 방식을 통해 객체 지향에 대한 이해도를 높이고 통제 변인과 조작 변인 설정을 명확하게 할 수 있었고, 협업을 더욱 효율적으로 진행할 수 있었다.

마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?

대회를 진행하면서 한계를 마주한 적이 적지 않았다. 다음과 같이 정리할 수 있다.

1. 내가 적용한 방법론이 오히려 성능을 하락시켰다.
2. 객체 지향 방식에 대한 이해도가 낮아 코드를 객체 지향적으로 작성하려면 어떻게 해야 하는가에 대한 고민이 많았다. 이에 더해 클린 코드란 무엇인가에 대한 고민도 있었다.
3. 깃과 깃허브를 제대로 활용하지 못했다. 깃과 깃허브의 큰 장점 중 하나가 개인의 코드를 쉽게 공유하는 것이라고 생각하는데, 처음 제대로 사용하다 보니 각자 어떤 코드를 구현할지, 어떻게 main branch 에 merge 할지 합의가 안 된 상태로 깃허브를 관리해서 branch 가 과도하게 많아지고 그 과정에서 손실된 코드가 있었다.

한계/교훈을 바탕으로 다음 프로젝트에서 시도해볼 것은 무엇인가?

객체 지향적 코딩에 대한 이해도를 높여 다음 프로젝트에서는 객체 지향적이고 정돈된 코드를 만들고 싶다.

협업 툴 사용법을 더 공부하여 협업 툴을 완벽히 활용하고 싶다. 툴의 사용법도 중요하지만 툴을 사용하기 전에 어떤 방식으로 코드를 추가하고 디버깅할지 등의 사전 합의가 있어야 한다고 생각된다. 이 부분도 다음 프로젝트에서는 명확하게 정의하고자 한다.

또한 프로젝트 초기에 명확한 업무 분담을 해서 더욱 효율적으로 프로젝트가 진행될 수 있게 하고 싶다.

<이다현>

나는 내 학습목표 달성을 위해 무엇을 어떻게 했는가?

나의 학습목표는 대회의 데이터셋과 베이스라인 코드를 완벽하게 이해하고, 다양한 논리적인 방법론을 시도하여 성능을 높이는 것이었다. 이를 위해 EDA를 진행한 결과, 해당 데이터셋에서는 특정 라벨에 대한 데이터가 과도하게 몰려 있는 것을 확인할 수 있었다. 특히, age label에서 이러한 불균형이 더욱 심각한 것으로 나타났다. 이는 모델의 학습과 일반화 능력에 부정적인 영향을 미칠 수 있으므로, 적절한 전처리 기법을 적용하여 불균형을 보완하는 것이 중요하다고 생각하고, 데이터 불균형을 해소하겠다는 목표를 설정하였다.

나는 어떤 방식으로 모델을 개선했는가?

우선적으로 모델 선택을 위해 많은 pretrained model을 이용했다. 처음에는 레이어가 깊을수록 좋다고 생각하여, 깊은 레이어 모델부터 실험해봤다. 그러나, 예상과는 달리 모델이 더 복잡해질수록 성능이 저하되는 결과를 관측할 수 있었다. 이는 모델이 복잡할수록 불필요한 특징들까지 학습할 수 있는데, 이 부분에서 과적합(overfitting)이 발생했다고 판단하였다. 데이터셋이 적기 때문에, 모델의 깊이를 적절하게 선택하는 것이 중요하다는 것을 깨달았다. 실험을 통해 성능을 분석하면서, 성능을 올리기도 그 원인을 찾고자 노력하였다.

데이터 불균형 문제를 해결하기 위해 Focal Loss를 사용하였다. Focal Loss는 잘못 예측된 샘플의 가중치를 높이는 방식으로 학습을 진행한다. 이를 통해 모델이 더욱 강하게 어려운 샘플을 학습할 수 있게 되며, 데이터 불균형 문제를 해결할 수 있었고, early stopping을 도입하여 과적합을 방지하도록 설정하였다.

또한 데이터 불균형 문제를 해소하기 위해, 불균형한 클래스의 데이터를 삭제/증강 처리하여 데이터셋의 클래스 간 균형을 맞춰주었다. 하지만, 불균형을 완화시켜도 성능이 떨어지는 경우가 발생했다. 이는 age의 데이터 불균형을 해소하면서 다른 클래스(gender, mask)에 영향을 끼친 것이라고 판단하여 따로 학습하는 병렬 모델을 제시하였다.

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

병렬모델에 대해 조금 자세히 설명하자면, age에 대해서는 Over-sampling과 Under-sampling을 진행해주어 데이터 불균형을 해소하고, 각 클래스를 독립적으로 학습하면서 다른 클래스(gender, mask)는 전체 데이터셋

을 사용했다. 이를 통해, 불균형 문제를 해결할 수 있었다. 위 방법을 시도하면서, 정말 많은 인사이트를 얻어서 성장할 수 있던 계기가 됐다고 생각한다.

팀원들과 함께 대회를 진행하면서 협업의 중요성을 다시 한번 느끼게 되었다. 각자의 코드 방식과 문제 해결에 대한 시각이 다르기 때문에 이를 조율하는 과정이 어려웠다고 생각한다. 하지만, Wandb, Git와 Notion을 최대한 활용하여 각자의 구현 결과를 공유하고, 서로의 코드를 분석하면서 문제를 해결해 나갈 수 있었다. 이를 통해 서로의 아이디어를 공유하고 발전시키며, 함께 성장할 좋은 기회가 되었다고 생각한다.

마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?

구현한 Data Augmentation 이 성능 향상에 기여를 못했다는 한계에 마주해서 아쉬웠지만, 불균형을 해소하기 위해 다양한 방법을 시도하고 많은 인사이트를 얻을 수 있었다. 이번 프로젝트를 진행하면서 아쉬웠던 점은 더 많은 도전적인 시도를 해보지 못했다는 점이다. 사람을 조금 더 잘 인식하기 위해 배경을 제거하는 방법 혹은 사람 인식을 통해 사람을 먼저 인식하고 학습을 시키는 것처럼 많은 도전을 해보지 못해서 아쉬웠다. 다음 프로젝트에선 더욱 더 도전적인 시도를 하고, 실험을 통해 더 나은 성능을 얻을 것이다. 또한, 이번 프로젝트에서 얻은 인사이트와 경험을 살려서 다음 프로젝트에서 더욱 더 성장할 수 있도록 노력할 것이다.

한계/교훈을 바탕으로 다음 프로젝트에서 시도해볼 것은 무엇인가?

1. 조금 더 도전적인 방법론을 시도하기
2. 협업 툴에 대해 완벽히 이해하고 200% 활용하기. Trello 와 같은 관리 도구를 이용해 각 작업의 진행 상태를 관리하기
3. 코드의 가독성을 고려해 Clean Code 작성하기. 유지보수가 쉽도록, 코드 작성 요령을 팀원들과 미리 맞추고 코드 구현하기

<이상민>

1. 목표

프로젝트를 진행하기 전 아래와 같은 개인 목표를 설정했다.

초반 목표

- 리더보드 성능에 연연하지 말고 성장하는 것에 초점을 두자
- 다양한 가설을 세우고 이를 검증하자

후반 목표

- 리더보드 성능향상

프로젝트 초반에는 다양한 가설을 세우고 왜 그런지 학습하며 가설을 검증하는 방향으로 프로젝트를 진행했다. 그러나 시간이 지나고 마감일이 가까워질 수록 학습과 성장에 집중하기 보다는 리더보드 성적 향상에 중점을 두기 시작했다. 아마 생각보다 리더보드 성적이 높아 오히려 더욱 성적 향상에 집중하기 시작했던 것 같다.

처음 계획한 목표와는 점점 어긋나기 시작했지만 리더보드 성적을 향상시키기 위해 더 많이 공부하고 실험할 수 있어 결과적으로 리더보드 성적 향상과 개인의 성장이라는 두가지 목표를 달성할 수 있었다고 생각한다.

2. 프로젝트에서 마주한 한계점

데이터 업샘플링과 다양한 과적합 방지 기법(loss, regularization, optimizer , dropout, normalize)를 적용해 보았지만 아래 두가지를 해결할 수 없었다.

(1) Data imbalance

(2) over fitting