

# 마스크 착용 상태 분류

CV\_15조 열정열정열정!

## 프로젝트 개요

COVID-19의 확산으로 우리나라를 포함한 전 세계 사람들은 경제적, 생산적 활동에 많은 제약을 가지게 되었습니다. COVID-19 확산을 방지하기 위해서는 모든 사람이 올바르게 마스크를 착용하여 전파 경로를 원천 차단해야 합니다. 하지만, 이를 확인하기 위해서는 많은 인적자원이 소모되므로, 올바른 마스크 착용 여부를 자동으로 검출해주는 시스템이 필요합니다.

저희 팀은 프로젝트를 진행하기에 앞서 파이썬 버전, jupyter 노트북 등 개발 환경을 동일하게 맞추고 의견 공유를 위해 카카오톡, slack, 줌을 협업 tool로 사용하여 진행하였습니다.

또한, 저희는 이번 대회 목적 '지금까지 배운 내용들을 최대한 적용해 보는 것'이라고 생각했습니다. 그래서 각자의 역할을 정해 프로젝트를 진행하는 형식이 아니라 각자 써보고 싶은 augmentation, model을 적용해보고 '어떤 것을 해보니 좋았더라', '어떤 걸 해보니 좋지 않았다' 등의 의견 공유를 하는 형식으로 진행했습니다.

## 프로젝트 팀 구성 및 역할

팀명 : 열정열정열정! (CV\_15조)

팀원 : 민선아 : EfficientNet 모델 설계 및 multi-task learning을 통한 일반화 성능 개선

김승규 : augmentation을 통한 데이터 불균형 개선 및 실험

남권표 : AlexNet, resnet 모델 설계 및 실험

임동우 : regnet 계열 모델 설계 및 실험

유승우 : oversampling을 통한 데이터 불균형 개선 및 실험

## 프로젝트 수행 절차 및 방법

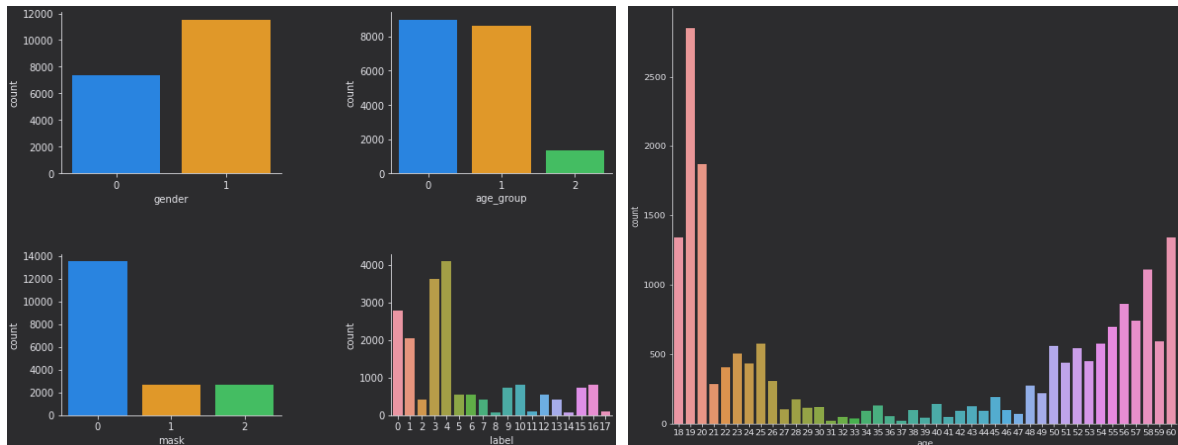
- 1) 이미지와 각 변수들에 대한 EDA
- 2) 데이터 불균형 문제를 해결하기 위한 augmentation, oversampling
- 3) 베이스라인 코드를 참고하여 다양한 pretrained 모델 Fine-tuning
- 4) kfold cross validation을 활용한 모델 검증
- 5) multi-task learning 및 ensemble을 통한 모델 일반화 성능 개선

## 프로젝트 수행 결과

### 1) EDA



- 전체 사람 수 : 4,500
- 한 사람 당 사진의 개수 : 7 (마스크 착용 5장, 이상하게 착용 1장, 미착용 1장)
- 이미지 크기 : (384, 512)
- Train set : 2700명(사진 수는  $2700 * 7$  장)
- Test set : 1800명(사진 수는  $1800 * 7$  장)
- Class : 18개의 클래스
  - mask : [올바르게 착용], [잘못된 착용], [착용하지 않음]
  - age : [30살 미만], [30살 이상 60살 미만], [60살 이상]
  - gender : [남성], [여성]

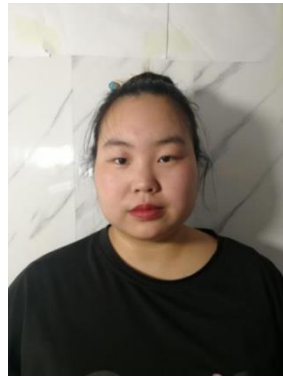


EDA 결과 중점적으로 봐야할 것은 나이와 마스크입니다. 나이의 그룹화 전에는 30,40대가 매우 적게 나타났지만, 그룹화 이후에는 2 범주인 60살이 매우 적게 나타난 것을 볼 수 있습니다. 또한, 마스크 착용 여부 분포에서 정상적으로 착용한 사람이 5명이기에 나이와 마스크 착용 여부에 따라 라벨 분포가 불균형을 띄는 것을 볼 수 있습니다.

따라서, 저희는 모델이 데이터가 부족한 60살을 예측하기 어려울 것이라 판단하여 59살 데이터들을 60살로 라벨링하여 분류를 진행하였습니다.



Female -> Male



Male -> Female



incorrect -> normal



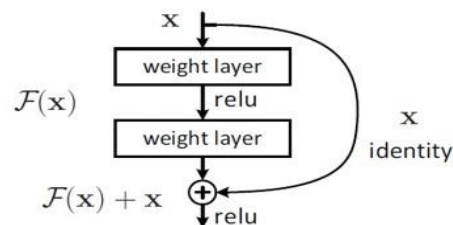
normal -> incorrect

또한, 저희는 이미지 전수조사를 실시하여 위의 사진과 같이 성별과 마스크 착용 여부에서 잘못된 라벨링을 발견하였으며, 총 16개의 레이블을 수정하였습니다. 성별의 경우 확실한 정답 레이블을 알 수 없으므로 저희가 판단하기에 확실히 잘못되었다고 생각되는 이미지들을 변경하였습니다.

## 2) 모델 개요

ResNet

- ResNet은 이전 층의 계산 결과를 다시 이용하는 residual learning을 사용한 방식으로 층이 깊어져도 발생하는 문제들을 어느 정도 예방할 수 있는 모델입니다.



## 3) 모델 선정 및 분석

### I. Architecture : resnet\_18

- a. Public score : 0.7456
- b. Train time Transformers
  - i. Resize(512,384)
  - ii. CenterCrop((320,256))
  - iii. ColorJitter(0.3,0.3,0.3,0.3)
  - iv. Normalize(on train validation dataset 18900장)
- c. Criterion : Cross Entropy Loss
- d. Optimizser : Adam
- e. Scheduler : StepLR(3epoch , 0.1)

- f. 검증 전략
  - i. 학습 데이터셋과 검증 데이터셋을 ID를 기준으로 나누어 데이터 누수현상을 방지하였고, 각 데이터셋의 label의 비율을 동일하게 설정하였습니다.
  - ii. 학습 과정에서 가장 높은 accuracy를 가진 모델을 저장하여 Test예측을 진행하였습니다.

## II. Architecture : Regnetx\_016 (timm)

- a. Public score : 0.7445 / Private score : 0.7270
- b. Train time augmentation (augmentations)
  - i. Resize(384, 384) > RandomResizedCrop(326, 326) > Resize(260, 260)
  - ii. OneOf(HorizontalFlip, VerticalFlip, RandomRotate90)
  - iii. OneOf(ShifScaleRotate, RandomBrightnessContrast, HueSaturationValue)
  - iv. Normalize (on train-valid dataset 18900장)
- c. Criterion : Focal loss
- d. Optimizer : AdamW
- e. 검증(Validation) 전략
  - i. 5-fold cross validation
  - ii. Valid dataset 1 (한 사람당 mask 1장 + norm 1장 + incorrect 1장 구성)  
Valid dataset 2 (한 사람당 mask 5장 + norm 1장 + incorrect 1장 구성)
  - iii. 학습 과정에서 macro-f1 score (on valid dataset 1) 가 가장 높았던 모델 저장
  - iv. 추가로 Accuracy, macro-f1 score (on valid dataset 2) 확인
- f. 앙상블 방법
  - i. 학습이 잘 되지 않았다고 판단했던 0,4 fold 제거하여 총 3개의 fold (1,2,3) 선택
  - ii. 여기에 Batch\_size만 64로 바꾼 후 1,3 fold 추가 선택
  - iii. 총 5개의 모델에 대해 Soft-Ensemble 적용
- g. 특이사항
  - i. Train Dataset 구성 시, 한 사람당 mask 1장 +norm 1장 + incorrect 1장 구성  
> Mask class imbalance 해소 목적
  - ii. 전체 Batch 중 50%에 Cutmix 적용하였으며, 마지막 2 epoch에 대해서는 적용 X  
> Label smoothing + Age class imbalance 해소 목적

## 4) 시연 결과

임동우_T3179	0.7445	79.0794	상세 보기	2022-03-03 17:46		
유승우_T3130	0.7456	80.1270	상세 보기	2022-03-03 17:38		

## 자체 평가 의견

저희 팀은 이번 대회에서 최대한 많은 교육 내용을 적용시키려고 노력했습니다. 학습한 내용을 바탕으로 다양한 실험을 통해 어떤 작업이 성능이 좋아지고 나빠지는지 확인하였으며, 이것들을 실습함으로써 이전에 배운 내용을 잘 흡수할 수 있었습니다.

저희는 data imbalance를 해결하기 위해 data augmentation, under & over sampling 등 다양한 실험을 진행하였으며, 조금 더 복잡한 모델을 사용하였습니다. 하지만, 이러한 작업들이 모델 성능 향상에 크게 효과가 없었고, 성능을 많이 올리지 못한 채 프로젝트를 마무리하게 되었습니다. 이를 통해 데이터 불균형이 학습에 큰 영향을 미친다는 것과 복잡한 모델이 항상 문제를 잘 해결하는 것은 아니라는 것을 알게 되었습니다.

이번 프로젝트는 분업이 아니라 각자 궁금했던 교육 내용들을 적용시켜 보면서 성능 개선 여부를 공유하는 방식으로 진행했습니다. 이 덕분에 각자 이전에 배운 내용을 잘 정리할 수 있었던 것 같습니다. 하지만, 팀원들과 분업해서 문제를 해결하려 하였다면 시간을 절약하여 더욱 성능을 높일 수 있었다고 생각되기에 많이 아쉽습니다. 또한, 저희는 주로 카카오톡, 줌으로 협업을 진행했는데, 이후 프로젝트에서는 깃허브나 노션과 같은 툴을 사용하여 코드 공유와 기간별 팀 활동을 자세히 기록하면 좋을 것 같습니다.

## 개인 회고

### ● 김승규\_T3037

○ 나는 내 학습목표를 달성하기 위해 무엇을 어떻게 했는가?

■ 우리 팀과 나의 학습목표는 무엇이었나? : 전체적인 프로세스, 데이터 전처리부터 모델링까지 해보는 경험을 하는 것을 목표로 했습니다. 따라서 사용할 모델이나, augmentation 방법 등 방향성에 대해서 의논을 하고 각자 코드를 작성해보는 방향으로 진행했습니다. 저는 그 중에서도 데이터를 다뤄보면서 augmentation에 집중해서 작업을 진행했습니다.

■ 개인 학습 측면 : 전체적인 진행을 해본 적이 없어서, 데이터 전처리부터 augmentation 하는 방법, 사소하게 풀더 이름을 수정하거나, 이미지를 열어보는 일도 익숙하지 않아 하나하나 검색해보고 정리해가면서 공부했습니다.

○ 나는 어떤 방식으로 모델을 개선했는가?

■ 사용한 지식과 기술 : Torchvision.transform에 있는 data augmentation 기능들 중 Randomhorizontalflip, Colorjitter, Centercrop, normalize, resize를 적절하게 사용해보려고 했습니다. 또한 모델은 resnet18을 기준으로 삼아 모델은 많이 건드리지 않고 다른 부분에서의 영향을 확인해보려고 했습니다. Augmentation은 Centercrop을 적절한 크기로 했을 때 효과가 좋았고, 사람 별로 데이터를 구분하는 것도 모델을 개선하는데 도움이 되었습니다. 모델을 직접 만드는 것도 좋지만 pretrained된 모델의 효과도 크다는 것을 느낄 수 있었습니다.

Model : resnet18 / Optimizer : AdamW / Loss : CrossEntropy func / Epoch : 10 / KFold

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

데이터가 어떤 편향이 있는지 등 데이터를 제대로 파악하고, 그에 맞는 전략을 세워야 효과가 좋다는 것을 느꼈습니다. 폴더를 열어 사진을 살펴보면서 마스크를 쓴 사진이 마스크만 다를 뿐 굉장히 유사하기 때문에 사람 별로 데이터를 나눠야 한다는 것을 생각해낼 수 있었기 때문입니다.

또한 팀원들의 이야기를 경청해서 좋은 아이디어가 있으면 적극적으로 코드에 반영해보려고 했습니다. 팀원들이 시도해보고 효과가 좋았을 때를 계속해서 공유해줘서 많은 도움이 되었습니다.

구체적으로는, 데이터의 편향을 처리할 방법을 찾아야 한다는 것, augmentation의 효과가 꼭 좋지는 않을 수 있다는 것, 너무 복잡한 모델은 간단한 task를 처리하기 어려울 수 있다는 것에 대해서 알 수 있었습니다.

○ 전과 비교해서, 내가 새롭게 시도한 변화는 무엇이고, 어떤 효과가 있었는가?

제 힘으로 전체적인 코드를 짜보려고 했고, 전체 연구과정에 대해서 파악하는 시간이 되었습니다.

○ 마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?

주먹구구식으로 코드를 바꾸면서 실험을 했던 것을 반성하게 되었습니다. 이전에 했던 실험 기록들이 날라가 버려서 기억에 의존해서 코드를 수정하고는 했던 점도 아쉬웠습니다. 베이스라인 코드를 보면서 깔끔하게 코드를 짜는 것에 많이 배울 수 있었습니다.

적절한 시간 배분을 하지 못했던 것, 팀원들끼리 일을 제대로 배분하지 못했던 것도 아쉬운 점이었습니다. 피어 세션 + 시간을 활용해서 방향성을 잡아보려고 했는데, 실험 특성 상 빠르게 되지 않았던 것도 있고, 리더보드에서 점수가 잘 오르지 않아 맞는 방향으로 하고 있는지 헷갈렸던 적이 많았습니다.

○ 한계/교훈을 바탕으로 다음 프로젝트에서 스스로 새롭게 시도해볼 것은 무엇일까?

다음 프로젝트에서는 베이스 라인과 같이 코드를 작성해보는 연습을 확실하게 하려고 합니다. 또한, pytorch lightning과 같은 도구를 사용해서 간단하게 코드를 줄여보고, tensorboard와 같은 기능을 활용해서 이전 실험과 직접적인 비교를 해보고, automl과 같은 도구도 사용해볼 생각입니다. 환경설정이 오래 걸릴 수 있으니 미리 캐글 대회로 테스트해보는 경험을 쌓으려고 합니다.

- 남권표\_T3072

제가 생각한 이번 대회 목표는 '지금까지 배운 내용을 최대한 활용해보자'였습니다. 그래서 eda, dataset, model들을 최대한 많이 다뤄봤습니다.

처음 eda를 수행할 때는 많이 어색했습니다. 그래서 게시판의 공유된 글들을 보고 감을 잡고 나름 대로의 방법으로 eda를 수행했습니다. 처음에는 간단히 그래프를 그려보는 것조차 '이렇게 하는게 맞나?' 망설여졌지만 요즘은 eda에 정답은 없다고 생각하게 되어 과감히 그래프를 그려보고 확인하는 편입니다.

eda를 수행하고 data의 분포를 확인해보니 60대, 마스크를 쓰지 않은 사람 등 부족한 데이터가 존재했고, 처음에는 단순히 augmentation을 통해서 부족한 데이터를 매꿔주자고 생각했습니다. 이때 모든 데이터에 augmentation을 적용하면 마스크를 쓴 사진과 같이 이미 많은 데이터 또한 늘어날 것이라고 생각해서 처음에는 데이터가 부족한 사진들에 대해서만 augmentation을 적용했습니다. 그 결과 augmentation을 사용하기 전보다 결과가 좋지 않았습니다. 단, 모든 데이터를 대상으로 적용해본 결과 오히려 결과가 좋았습니다.

augmentation을 수행해도 결과가 좋지 않았고 저는 이 문제가 데이터 편향에도 있지만 데이터가 부족해서 일어나는 현상이라고 생각했습니다. 그래서 validation set에 포함된 데이터도 학습에 사용하고 싶어 cross validation을 사용했습니다. 이때 early stopping도 동시에 적용시켜 각 fold의 모델을 저장해 단순 평균을 사용해 모델을 앙상블 해봤습니다. 그 결과 3% 정도 결과가 향상되었습니다.

이후 여러 augmentation을 적용해봤지만 좋은 결과는 보이지 못했습니다. 이 문제는 데이터 편향 때문에 발생하는 문제라 생각했고, undersampling 혹은 oversampling 중 하나를 사용해보려 했습니다. 하지만 undersampling을 사용하면 데이터가 너무 줄어들 것이라 판단하였고 oversampling을 해보려 했습니다. 이를 위해 부족한 데이터에 대해 augmentation을 수행한 사진들을 파일에 따로 저장하고 학습에 사용해봤습니다. 하지만 이 방법을 사용해도 성능이 올라가진 않았습니다.

이후 다른 방법을 시도해보다 대회가 끝났습니다.

이번 대회는 앙상블을 단순 평균 기법만 사용해본 것, 그리고 데이터를 더 많이 다뤄보지 못한 점이 아쉬웠던 것 같습니다. 처음에 cross-validation, early stopping 코드를 작성하는데 시간을 좀 뺐긴 것 같지만 두 개념은 확실히 알아간 것 같습니다. 다음 대회 때는 모델 관련 기법들 보다 데이터에 좀더 집중해 보고 싶습니다.

- 민선아\_T3078

## 우리 팀의 학습 목표

리더보드 성능 향상, 잘못된 labeling 탐색, 성능 향상을 위한 데이터 augmentation 방법, age 클래스의 불균형 문제 해소

## 목표 달성을 위해 한 일

### 공동 학습 측면 :

매일 아침마다 각자가 시도할 방식들을 정하고 그 방식을 시도한 결과를 피어 세션 때 공유하여 효과적이었던 방법들을 같이 적용해 봄

### 개인 학습 측면 :

- k-fold Cross validation 기법 학습 및 적용
- multi-task learning 방식 적용
- Imbalanced Sampling 적용
- Facenet\_pytorch의 MTCNN을 적용
- RandomGrayScale() augmentation 적용

## 모델 개선 방식

내가 사용한 기술 : Multi-task learning 방식 적용. Pretrained된 모델에 대한 classifier를 하나가 아닌 task별로 따로 두는 방법을 적용. Age, gender, mask 로 task를 나누고 각자 다른 loss를 사용해서 학습.

## 내가 한 행동으로 얻은 결과

F1 score 69.64 -> F1 score 71.74

Accuracy 77.58 -> Accuracy 77.66

## 마주한 한계, 아쉬웠던 점

- 기존의 pretrained된 모델만 바뀌가면서 학습하는 데에 시간을 많이 소비해서 새로운 방법론을 찾을 생각을 하지 못했던 점.
- 앙상블 방식을 적용해보지 못했던 것
- Facenet\_pytorch의 MTCNN의 오류를 해결하지 못해서 사용하지 못한 점

## 한계/교훈을 바탕으로 다음 프로젝트에서 스스로 새롭게 시도해볼 것

- kaggle이나 데이콘 같은 대회에서 비슷한 task를 해결한 상위 랭커들이 어떤 방식으로 풀어 나갔는지를 보면서 인사이트 얻기.
- 앙상블 방법론에 대해 공부하고 직접 적용해보기



- 유승우\_T3130

이번 프로젝트에서 제 학습 목표는 결과 제출까지 직접 코드를 구현해보는 것이었으며, 시간이 남으면 CUI 환경에서도 코딩을 해보는 것이었습니다. 막상 구현하려니 눈앞이 캄캄했지만, 팀원들의 도움과 앞서 배웠던 수업 내용과 과제를 참고하여 무사히 결과를 제출할 수 있었습니다.

저는 가장 먼저 EDA를 진행하여 이미지와 각 변수 데이터를 뜯어보았고, age와 mask 데이터가 불균형이라는 것을 알게 되었습니다. 하지만, 일단 Raw data 자체로 분류한 결과를 보고 싶었기에 우선 이 사실을 인지한 상태로 모델 설계를 시작하였습니다.

우선 가장 기초적인 CNN 모델인 AlexNet부터 vggnet, resnet, efficeintnet 순서로 분류함으로써 모델 별 정확도의 차이를 확인하고자 하였습니다. AlexNet으로 분류한 결과 역시 가장 단순한 모델이라서 그런지 정확도가 60% 정도로 높지는 않았기에, 이후 조금 더 성능이 좋다는 resnet-18과 efficientnet을 사용하였으며, 전이 학습을 통해 정확도를 높이려 하였습니다. 두 모델 모두 정확도는 높게 나타났지만, 과적합의 문제가 발생하였고, 불균형 때문인지 각 모델들이 age의 예측을 잘 못하는 것으로 나타났습니다.

이번 결과로 인해 확실히 raw data 그 자체로 분류하는 것은 성능을 높이기 어렵다고 판단하였습니다. 성능을 높이기 위해서는 불균형과 과적합 문제를 해결해야 한다 생각하였고 data augmentation 과 oversampling을 통해 문제를 해결해보려 하였습니다. 하지만, 오히려 성능이 떨어지는 효과가 발생했고, 이후 age 59살을 60살로 라벨링하여 resnet-18로 분류해보았는데, stratified KFold로 검증한 결과가 기존의 efficientnet보다 4%가량 높아졌기에 최종 모델로 제출하였습니다. 이를 통해, 무조건 모델이 좋다고 성능이 잘나오는 것이 아닌 모델에 데이터를 어떻게 다듬어서 넘겨줄지가 중요하다는 것을 깨달았습니다.

이전에는 과제를 중심으로 코드 이해와 빈칸을 채우면서 학습을 하였지만, 이번 프로젝트로 처음부터 끝까지 직접 코드를 구현함으로써, 어떤 개념을 어느 부분에 어떻게 사용해야 하는지 이해하였습니다. 하지만, 처음해본만큼 각 파트 별로 시간소비가 컸고, 그로 인해 hyper parameter tuning과 wandb 혹은 tensorboard와 같은 시각화 툴을 사용하지 못한 것이 아쉬웠습니다. 또한, 팀 별로 역할을 분담하여 작업하였다면 조금 더 다양한 실험을 해볼 수 있었고, 더욱 많은 배움을 얻을 수 있었을 것 같다는 아쉬움이 남습니다.

이번 프로젝트를 경험한 것으로 논리적으로 차근차근 생각하여 하나씩 해 나가야 한다는 것을 배웠고, 혼자 구현하는 것보다는 팀원과 협업하며 진행하면 더 많은 것을 진행하고 배울 수 있다는 것을 알게 되었습니다. 사실 첫 프로젝트라서 생각없이 우왕좌왕 진행했던 부분이 많았는데, 다음 프로젝트에서는 구체적으로 계획을 세워 조금 더 깔끔하게 프로젝트를 진행해보고 싶고, 주피터 랩을 벗어나 CUI환경에서 작업을 시도해보고 싶습니다.

- 임동우\_T3179

- 나는 내 학습 목표를 달성하기 위해 무엇을 어떻게 했는가?

- 처음에는 SOTA 모델을 비교하는 식으로 접근했지만, 유의미한 성과를 얻지는 못했습니다.
- 이때 18개의 class로 이루어진 label을 그대로 사용하는 것이 맞는지에 대한 고민을 했습니다. 그래서 Label을 mask, gender, age 3개의 task로 나눠보니 age estimation이 가장 어렵다는 것을 알 수 있었습니다.
- 또한, 데이터 수가 적고 불균형도 심했기 때문에, pretrained model의 일부 레이어를 freeze시키거나 batch를 구성하는 과정에서 class weight에 따라 sampling을 하기도 했지만, 성능 개선은 미미했습니다.

- 나는 어떤 방식으로 모델을 개선했는가?

- 데이터 전처리 과정에서 있었던 코드 실수를 고친 뒤로 점수가 오르기 시작했습니다.
- Mask class 불균형 해소를 위해 Mask 1장만 train dataset에 포함하도록 하였습니다. 그리고 Age class 불균형 해소를 위해 59살을 전부 60살로 Re-labeling 하였습니다.

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

- 결국 점수를 올리기 위해서는 데이터를 잘 파악하는 것이 중요하다는 깨달음을 얻을 수 있었습니다. 그리고 코드 상에 실수는 없는지 꼼꼼히 살펴봐야겠다는 교훈도 얻었습니다.

- 전과 비교해서 내가 새롭게 시도한 변화는 무엇이고, 어떤 효과가 있었는가?

- 전에는 label 자체를 바꾸기 보다는 최대한 모델이 불균형한 데이터를 잘 학습할 수 있게끔 모델링을 하곤 했었는데, 이번 대화를 해보면서 생각이 조금은 바뀌었습니다.

- 마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?

- Class 불균형은 어느정도 해결했다고 생각하지만, Age estimation은 어려웠던 것 같습니다. 그리고 valid score와 lb score 차이가 큰 것이 모델 선정을 어렵게 만들었습니다.
- 아쉬운 점은 여러가지 실험을 해보지 못한 것입니다. age를 3개로 구분하는 것이 아니라 young – not young, old – not old 로 학습을 시키고 이 결과를 soft-ensemble 하는 방식이나, mask 착용여부를 먼저 분류하고 이후에 gender/age를 예측하도록 하는 방식을 시도했으면 결과가 어땠는지 궁금합니다.

- 한계/교훈을 바탕으로 다음 프로젝트에서 스스로 새롭게 시도해볼 것은 무엇일까?

- Score에 심각한 차이가 있다면 이것을 먼저 해결해보려고 생각중입니다. 그리고 초기 단계에서 어떤 task로 해결하는 것이 옳을지 방향을 먼저 설정하는 것도 좋을 것 같습니다.