

# Mask Classification Wrap-up Report

CV-9조

구상모, 권규보, 배서현, 오한별, 이영진

## 1. 프로젝트 개요

### A. 개요

본 대회는 사람의 이미지와 메타데이터가 주어질 때, mask/gender/age의 label을 예측하는 multi-label classification 문제로 Mask는 미착용/오착용/정상착용, Gender는 남자/여자, Age는 30살 미만 / 30살 이상 ~ 60살 미만 / 60살 이상으로 구분된다. 최종적으로 분류해야 하는 데이터는 18개였다. ( mask 3개 \* gender 2개 \* age 3개)

### B. 환경

(팀 구성 및 컴퓨팅 환경) 5인 1팀, 인당 V100 서버를 VSCode와 SSH로 연결하여 사용하였다.

(협업 환경) Notion, GitHub

(의사 소통) 카카오톡, Zoom, Slack

## 2. 프로젝트 팀 구성 및 역할

전체	문제 정의, 계획 및 타임라인 수립, 모델 튜닝, 아이디어 제시
구상모	Face detection & Data imbalance 전략 수립 및 구현, Data Noise GUI 코드제작, three-way model 베이스라인 제작, Custom CutMix 구현, 시각화용 wandb 연동, 발표 자료 제작
권규보	train 전략 수립, validation data 학습에 이용, 비율이 랜덤한 CutMix 구현, f1 loss 도입, 앙상블 전략 수립 및 구현
배서현	EDA 정리 및 문서화, Data Augmentation 전략 수립
오한별	EDA(Data bias, label distribution), loss experiment, arcface loss experiment, 문서 정리
이영진	Face detection 전략 수립 및 구현, Data noise 수정 코드 제작, 앙상블 전략 수립 및 구현

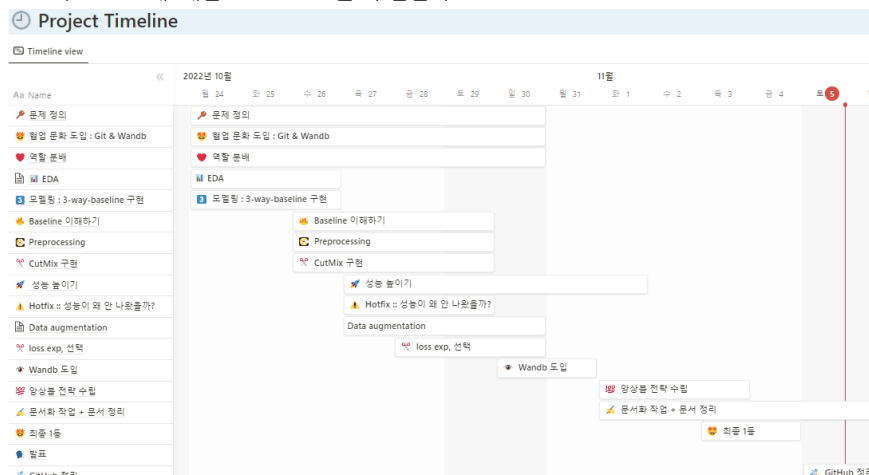
## 3. 프로젝트 수행 절차 및 방법

### A. 팀 목표 설정

- 경진대회의 모든 프로세스 한번씩 겪어보기
- 경진대회 F1 score 0.7 달성(1주차) → F1 score 0.8 달성(2주차)
- 팀원들 각자 구현해보고 싶은 것들 구현하고, 그에 대한 문제의식을 설명하기

### B. 프로젝트 사전기획

- (1) 프로젝트 Task에 대한 Time-line을 수립한다.



(2) 팀원 간의 마음가짐

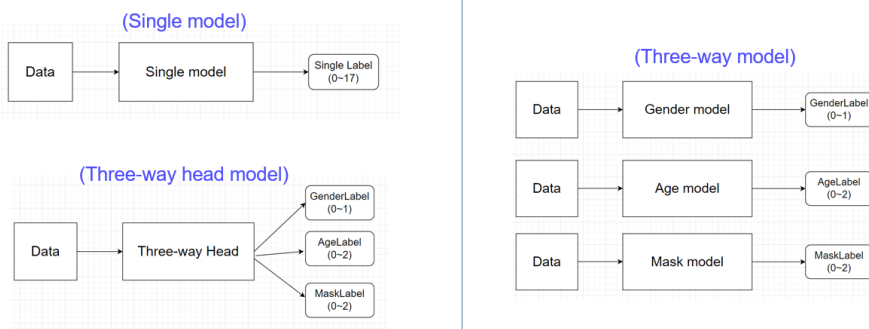
- [1] 서로 모르는 부분 있으면 계속 공유하기
- [2] 맡은 역할을 못해올 것 같으면, 어느 지점에서 막혔는지 이야기하고 팀원들은 독촉하지 말기
- [3] 어떤 선택을 할 때, 그에 대한 **의사결정의 근거를 기록**하고 확인하기
- [4] 칭찬은 아낌없이, 비판은 한 번 더 생각하고 말하기
- [5] 정해진 정답은 없으니, 마음껏 아이디어를 개발하기

(3) 팀원 간의 협업문화

- [1] AI Stages에 **제출 시 config.json을 모델 설명란에 기입**하기
- [2] 각자 브랜치를 개설하되, **대회가 끝난 이후 각자의 코드를 리팩토링하여 정리**하기로 하였다.

## C. 프로젝트 문제 정의 및 EDA

Label의 예측방법에 따라 크게 세 가지 접근 방식이 가능하다고 생각하였다.



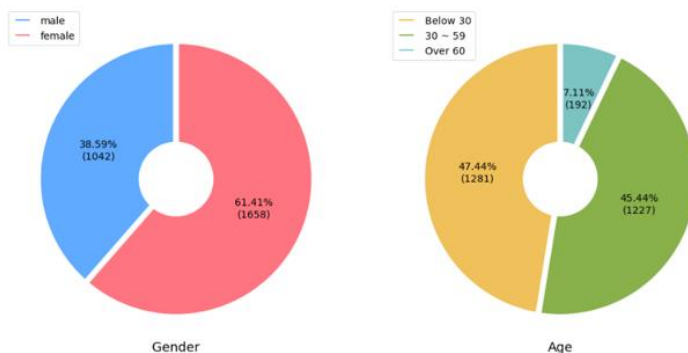
한 번에 label을 예측하는 single model, 한 모델에서 각각의 label을 예측하는 three-way head model, 병렬적인 세 모델이 각각을 예측하는 three-way (parallel) model이 그것이다. Three-way head model의 경우 실험과 샘플러 구현의 어려움으로 인해 기각하였고, Three-way (parallel) model의 경우 단일 age model을 개선할 방법을 찾지 못해, **최종적으로는 single model을 진행하였다!**

한편, 본 대회의 데이터셋을 EDA해본 결과 크게 3가지 문제점을 파악할 수 있었다.

[1] 이미지의 Noise : **사진마다 이미지에서 얼굴이 차지하는 비중이 달랐다.** 어떤 이미지는 얼굴이 사진의 대부분을 차지하는 반면, 어느 이미지는 얼굴 정보가 거의 없었다. 이는 즉, **Crop** 시 얼굴이 깔끔하게 나오지는 않았을 것을 의미한다.

[2] 레이블의 Noise : **잘못 표기된 레이블이나, 육안으로 보기도 헷갈리는 레이블이 있었다.** 네이버 API를 이용하여, 잘못 예측된 label을 1차적으로 CSV 형태로 반환받고, 이를 고치는 GUI 코드를 만들었다. 팀원들의 협력 하에 최종적으로 오분류된 mislabel.csv를 만들고, 이를 자동으로 고쳐주는 코드를 작성하였다.

[3] 데이터의 불균형 :



데이터 불균형이 심각하다. 60대 이상의 데이터가 약 7%로, 다른 age label의 20%도 안됨을 확인할 수 있다. 한편 본 데이터의 개수는 18,900장으로 충분히 많은 편은 아니기에 오버피팅이 쉽게 일어났다.

따라서 **오버피팅을 방지하면서도 데이터 불균형을 잘 헤쳐나갈 방법을 찾는 것**이 우리 팀의 핵심적인 과제였다.

## D. 데이터

### [1] DataSet & DataLoader

- (a) **Split by Profile** : 사람별로 구분해 train/val에 **같은 사람이 들어가는 것을 막아준다**. Train dataset의 정보가 validation dataset으로 흘러가는 것을 막아준다고 생각했다.
- (b) **WeightedRandomSampler** : 데이터가 불균형하여, label의 분포를 고려하여 **균등하게 샘플링해주는 샘플러를 구현하였다**.
- (c) **Age\_removal** : 모델이 보다 **확실한 feature를 학습하게끔**, 경계값인 20대 후반, 50대 후반 데이터를 제외하고 학습했다.
- (d) **Intra-class CutMix dataset** : **같은 label 간의 cutmix를 통해**, 전반적인 데이터의 오버샘플링을 꾀하였다. 같은 label 간의 CutMix가 **오버샘플링 기법인 SMOTE와 의미론적으로 유사하며**, data augmentation과 유사한 효과를 낸다고 생각하였다.

### [2] Data Augmentation

Data Augmentation은 모델의 오버피팅 가능성을 줄여야 하는 동시에, Test time 때의 데이터셋까지 고려하여 이루어져야한다고 생각했다. 본 대회 경우 Train/Test가 **동일한 환경에서 제작되었을 것이라 가정하여**, 일부 Soft Augmentation<sup>1</sup>만 사용했다.

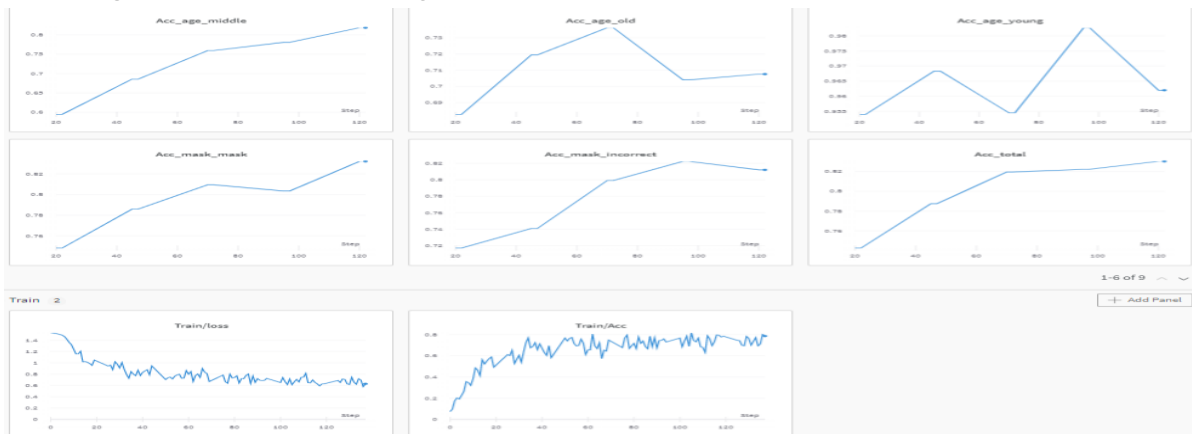
거의 모든 데이터가 정중앙에 정렬된 사람 얼굴이었고, 따라서 회전 + 상하반전 등의 변환은 일반화 가능성을 떨어뜨린다고 여겼다. 또한 **color 관련 변환을 이용하지 않았는데**, 우리는 우리의 모델이 **마스크와 사람의 얼굴의 색상 차이를 잡아내어 마스크를 분류한다고 여겼기 때문이다**. 따라서 ColorJitter 등의 변환은 마스크 탐지의 정확도를 떨어뜨린다고 간주했다.

되려, 모델이 **확실하게 얼굴의 표현만 학습할** 수 있게 전처리 단계에서 **Face detection 모델인 MTCNN**을 이용하였고 비슷한 얼굴들을 더욱 robust하게 구분할 수 있게끔, 이미지 간에 서로 정보를 섞어주는 cutmix를 적극적으로 이용하였다.







- (a) **Normalize** : ImageNet에서 사용되었던 RGB 픽셀 값의 평균과 표준편차 값을 이용하였다.
- (b) **MTCNN** : 얼굴과 그 주변부 정보를 남겨, 얼굴의 크기를 일관성있게 유지하고 배경의 노이즈를 제거하고자 했다.
- (c) **CenterCrop** : 만약 MTCNN이 얼굴을 탐지 못했다면, 이를 CenterCrop 시켰다.
- (d) **CutMix** : 같은 Label끼리(Oversampling 목적), 또는 다른 Label끼리(Robust training 목적) 이미지를 세로로 잘라, 2개의 이미지를 붙였다.
- (e) **RandErase** : 다 지우지 못한 배경의 noise를 제거하거나, 얼굴 정보를 일부 삭제해 보다 robust한 학습이 가능하게끔 했다.
- (f) **Resize** : EfficientB4의 기본 입력 사이즈인 (380, 380)에 맞게 사이즈를 조절했다.

## E. 시각화

1주차가 끝난 뒤, 성능을 더 올리기보단 지금 우리의 모델을 디버깅하고 Error case를 분석하는 것이 중요하겠다고 생각하여, wandb를 이용하여 모델을 추가적으로 디버깅하고자 하였다.



<sup>1</sup> Soft Augmentation : 이미지의 일부에만 noise를 주는 것을 의미한다. 회전, 뒤집기, 지우기 등의 일반적인 변환을 일컫는다.

image	label	pred	Age	Gender	Mask
	7	6	Age_GT:1, Age_pred:0	gender_GT:0, gender_pred:0	Mask_GT:1, Mask_pred:1
	7	8	Age_GT:1, Age_pred:2	gender_GT:0, gender_pred:0	Mask_GT:1, Mask_pred:1
	13	14	Age_GT:1, Age_pred:2	gender_GT:0, gender_pred:0	Mask_GT:2, Mask_pred:2
	16	15	Age_GT:1, Age_pred:0	gender_GT:1, gender_pred:1	Mask_GT:2, Mask_pred:2
	1	2	Age_GT:1, Age_pred:2	gender_GT:0, gender_pred:0	Mask_GT:0, Mask_pred:0
	7	8	Age_GT:1, Age_pred:2	gender_GT:0, gender_pred:0	Mask_GT:1, Mask_pred:1

← 17 - 22 of 504 → Export as CSV Columns... Reset Table

그래서, 학습이 끝난 후 각 3개의 label 별 분류 정확도, 그리고 validation step에서 모델이 잘못 분류한 것들을 시각화하고자 하였다. 이러한 시각화를 통해, 우리의 모델이 age prediction을 제일 어려워하며 MTCNN이 항상 얼굴을 잘 탐지하는 것은 아니라는 소중한 교훈을 얻을 수 있었다. 이로부터 2주차의 모델링 전략을 수립할 수 있었다.

## 4. 프로젝트 수행 결과

### A. 모델 평가 및 개선방향

초기 베이스라인은 ResNet34로 진행하였다. 이때 Train/valid score는 80% ~90%로 매우 높게 올라가는 데에 반해, 리더보드의 점수가 잘 나오지 않았다. Train/valid dataset이 Test dataset의 데이터 환경이 크게 다를 것이라 생각하진 않았기에, 우리는 모델의 오버피팅이 문제라고 생각하였다. 1만~2만여장의 작은 데이터를 모델이 모두 암기해버렸다고 가정하였다. 이 문제에 대한 접근법으로, 모델 갈래의 해결법과 데이터 갈래의 해결법으로 분류하여 모델을 개선하고자 하였다.

ResNet34로도 학습의 정확도가 이미 충분히 높게 나왔기에, 모델이 데이터의 특징을 학습하기에는 표현력이 이미 충분하다고 생각하였다. 하여, ResNet34보다 파라미터 수가 적으면서 이미지넷에서 성능이 더 좋았던 EfficientB4을 이후의 실험 모델로 택했다.

이후 데이터 불균형을 해결하기 위해 WeightedRandomSampler를 이용하였다. Accuracy 측면에서는 학습 시 크게 차이가 없었지만, label을 균등하게 배분해주기 때문에 F1 score가 어느정도 오를 것으로 생각했고, 실제로 오르는 것을 알 수 있었다. F1 score가 오르긴 했어도 여전히 모델이 오버피팅을 하고 있다 생각하여, 이때부터 Data augmentation 전략을 수립하였다.

팀원들끼리 EDA 결과를 공유하며, 우리의 데이터셋에서 image의 noise가 있고(얼굴 크기가 제각기 다르다는 것), label의 noise가 있음을 확인했다. 이러한 noise를 제거해주기 위해 MTCNN 모델을 이용하고, soft augmentation의 실험결과로 randerase를 적용해주었다. 그 결과 F1 score를 추가로 향상시킬 수 있었다.

이후 Cutmix 전략을 두 갈래로 고안하여, 동일한 클래스 간의 CutMix(오버샘플링 효과)와 다른 클래스 간의 CutMix(Robust learning 효과)를 구현하였다. 이러한 CutMix 방법론에 TTA까지 적용한 결과, F1 score를 획기적으로 끌어올려 1주차 LB 1위를 달성하였다.

그 후 2주차에는 wandb 시각화 툴을 본격적으로 연동하여 모델의 설명력을 높이고, ensemble 등의 기법을 고안하였다. Validation 데이터도 학습에 이용하고, CutMix를 반반씩 합치게 아니라 46~54% 로 다양한 비율로 합치는 방식으로 변형함으로써 성능을 한번 더 끌어올릴 수 있었다.


최종적으로 대회 종료날에는 LB 기준 1~3위 single 모델을 hard voting한 ensemble 모델과, single 1위 모델을 제출하였다. Private dataset에 이를 적용한 결과 F1 score와 Accuracy가 더 올라갔는데, 이는 우리의 모델이 class 간의 decision boundary를 robust하게 잘 학습한 덕분이라고 생각한다.

Metric learning을 통해 age를 더 잘 예측하고, 더욱 다양한 앙상블 기법을 응용하고자 했으나 시간제한으로 구현하지 못한 시도들도 있었다.

## B. 모델의 성능 결과

Model	Skills / Approach	F1 score	Accuracy
ResNet34	Only baseline	0.36	51%
EfficientB4	Split by profile + Sampler	0.56	65.4%
EfficientB4	(Same above) + MTCNN + RandErase	0.66	72%
EfficientB4	(Same above) + Intra-class cutmix	0.67	73%
EfficientB4	(Same above) + Intra-class cutmix + TTA	0.77	81%
EfficientB4	(Same above) + Inter-class cutmix(Training) + Validation data as train	0.76	82%
EfficientB4	Top-3 hard voting ensemble	0.8	83.1%

## C. 프로젝트 결과 및 사진

1	CV_9조		0.8007	82.8730	69	3d
---	-------	---	--------	---------	----	----

(Public) 1위 → (Private) 1위

## 5. 자체 평가 의견

### A. 잘했던 점

- [1] 서로 의견 제시를 할 때 항상 응원해주고 긍정적으로 생각하였다.
- [2] 한 가지 모델 구조만 시도하지 않고, 여러 구조에 대해 장단점을 생각한 뒤 선택했다.
- [3] 문제가 발생하면 함께 고민해주며 다같이 해결해보려 했다.
- [4] 프로젝트 중간중간마다, 어떤 것이 더 좋은 방법론일지 끊임없이 고민하고 그 필요성에 대해 소통했다.

### B. 아쉬웠던 점

- [1] 각자 하고 있는 것들을 완전하게 서로 공유하지는 못한 점이 아쉬워, 이에 대한 소통 능력을 발전시키고자 한다.
- [2] 각자의 문제를 종합하여 큰 관점에서 종합하기보단, 각자의 아이디어를 발전시켜 실험을 진행한 점이 아쉬웠다.
- [3] 초기 협업 문화를 도입하지 않아, 프로젝트 관리의 복잡도가 매우 커졌던 점이 아쉽다.
- [4] 코드 오류로 성능이 계속 안 나왔는데, 이 문제에 매몰되어 1주차의 실험 관리에 실패한 점이 아쉽다.

### C. 앞으로의 다짐과, 프로젝트를 통해 배운 점

- [1] 경청만이 좋은 소통방법은 아니다. 자신이 어떤 생각을 하고 있는지를 제대로 전달하고 다른 사람의 의견을 물어볼줄 알아야한다.
- [2] 협업의 필요성에 동의하고 모두가 그 문화에 익숙해야 프로젝트를 체계적으로 할 수 있다.
- [3] 프로젝트 관리를 체계적으로 할 다양한 도구들을 배우고 익숙해지자.
- [4] 앞으로는 더 세부적으로 명확히 정의된 역할분배를 하자.

## 6. 참고자료 및 문헌

- [1] Irwan Bello, William Fedus, Xianzhi Du, Ekin D. Cubuk, Aravind Srinivas, EfficientNet, Rethinking Model Scaling for Convolutional Neural Networks, 2019
- [2] Florian Schroff, Dmitry Kalenichenko, James Philbin, FaceNet: A Unified Embedding for Face Recognition and Clustering, 2015
- [3] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, Youngjoon Yoo, CutMix: Regularization Strategy to Train Strong Classifiers, 2019
- [4] Github, facenet-pytorch, <https://github.com/timesler/facenet-pytorch>
- [5] TowardDataScience, SMOTE

## 1. 팀과 나의 학습목표

처음으로 겪은 경진대회이자 협업 프로젝트였기에, 경진대회의 진행 흐름과 문화에 익숙해지는 것이 최우선 목표였다.  
수업에 나온 경진대회의 흐름에 따라, 각 단계에서 생길 수 있는 문제의식을 생각하고 그를 검증하는 것이 목표였다.  
또한 그간 배운 이론적 지식을 코드로서 직접 프로젝트에 녹여내고, 문제정의를 하며 ML 이론을 심도있게 고민하고 싶었다.

## 2. 내가 모델을 개선시킨 방법과 결과, 그리고 깨달음

(문제의식) – (방법론) – (의의) 순으로 기입하였습니다.

- A. [Data noise] **MTCNN**을 통한 data preprocessing : TTA에 적용하여 F1 score를 0.1 향상, **1주차 리더보드 1위 달성**
- B. [Data Imbalance] **WeightedRandomSampler** : 데이터를 고르게 배분하여 **새로운 2차 베이스라인 코드 제공**
- C. [Overfitting] **Custom CutMix** : cutmix 방법론을 고안하여 **1주차 1위 달성 + 최종 1위에 대한 빠른 방법론 제시**

처음에는 ResNet34를 Augmentation 없이 적용했다. Train 정확도가 매우 높게 나온 데 반해 Val 정확도와 리더보드 점수가 낮았다.  
이것이 **모델 오버피팅**이라고 생각하여, 우리가 해결해야 할 핵심과제를 **데이터와 모델 오버피팅**으로 규정했다.

### (A)Data noise와 MTCNN

우리의 사진 데이터에서는 얼굴이 차지하는 비중이 사진마다 다르다. 따라서 고정된 size의 Crop은 적합한 augmentation 기법이 아니라고 생각했다. 배경의 noise가 섞이거나, 얼굴의 일부만 잘려 나야 예측에 필요한 흰머리, 주름 등의 정보가 잘릴 수 있다고 생각하였다. 그래서 얼굴을 탐지하는 모델이 필요하겠다고 여겨, detect 시간이 빠르고 성능이 준수한 **MTCNN 라이브러리**를 도입하였다.

### (B) Data Imbalance와 WeightedRandomSampler

본 데이터에서는 60대 이상의 데이터가 별로 없어, train/valid에 label이 균등하게 들어가도록 하는 것이 중요했는데, 제공받은 베이스라인코드엔 이 기능이 없었다. 그래서 **WeightedRandomSampler**를 통해 각 label이 균등하게 batch에 들어가게 만들고, 이를 F1을 0.73은 보장하는 2차 베이스라인 코드로 지정했다. 덕분에 팀원들이 더 맘놓고 실험할 수 있어 뿌듯했다!

### (C) Model overfitting과 Custom cutMix( Intra-class Cutmix + Inter-class Cutmix)

모델의 표현력은 이미 충분하다 생각했기에, model의 robust한 학습을 위해 cutmix를 고안하였다. 기존의 cutmix는 두 이미지를 random하게 덧붙이지만, 우리의 데이터는 random하게 붙이면 label 정보가 사라진다고 생각해 세로 방향으로만 cutmix를 진행했다. 같은 label 간의 cutmix는 oversampling 효과가 있고, 다른 label 간의 cutmix는 robust learning의 효과가 있다고 생각했다. 이 아이디어를 심화시켜 최종 LB 1위 모델에 사용할 수 있었다.

### (D)방법론을 적용하면서 느낀 점들

프로젝트 초기에는 다양한 SOTA(최첨단) 방법론을 적재적소에 응용하는 것이 중요하다고 생각했지만, 대회를 진행할수록 현재 본인 프로젝트의 어떤 단계에 있는지를 진단하고, 지금의 문제의식을 올바르게 수립해 해결해나가는 것이 중요함을 알게 되었다.  
또한 그 해결에 대한 의사결정 기준은 결국 개념과 그에 대한 응용에서 출발함을 알 수 있었다. 대회를 진행하며 Sampler, Augmentation, Model의 표현력 등의 ML Fundamental을 더 고민해볼 수 있어 좋았다.

## 3. 마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?

1. 대회 1주차에 코드의 문제점을 발견해 성능이 계속 안 나왔는데, 이로 인해 1주차에 협업 문화를 올바르게 수립하지 못했다.
2. 원래는 나이/성별/마스크를 별도로 탐지하는 모델을 만들려고 했는데, age에 대한 파웨법을 찾지 못해 아쉬웠다.
3. Model soup, ArcFace 등 더 적용해보고 싶은 방법론이 많았는데 시간관계상 하지 못해 아쉬웠다.

## 4. 한계/교훈을 바탕으로, 다음 프로젝트에서 스스로 새롭게 시도해볼 것은 무엇인가?

1. 프로젝트 초기에 협업 도구와 관습을 정하여, 더욱 효과적으로 협업해보기
2. 실험 기록을 더욱 체계적으로 관리하고, 팀원들이 이해할 수 있도록 더 구조화된 코드를 제공하기

## 1. 우리 팀과 나의 학습목표

우리 팀은 level 2에서도 함께 하기로 했기 때문에 첫 프로젝트에서부터 앞으로의 협업에 필요한 도구들과 문화를 잘 정립하는 것이 목표였다. 개인적으로는 데이터를 분석하는 작업부터 모델을 학습시키고 앙상블까지 일련의 과정을 모두 경험해보는 것이 목표였고, 성능을 높이기 위해서 적용할 수 있는 방법론들에 대한 인사이트를 얻는 것이 두 번째 목표였다.

## 2. 나는 어떤 방식으로 모델을 개선했는가?

### A) Validation set을 학습에 이용

우리는 valid acc의 신빙성을 높이기 위해서 valid에서는 train에서 한 번도 보지 못 한 사람의 사진을 등장시키고 그걸로 검증해야 한다고 생각했기 때문에 MaskSplitByProfileDataset을 사용하였다. 하지만 위 데이터셋을 사용할 경우 우리가 사용할 수 있는 2700명의 데이터 중 540명의 데이터 (default val\_ratio가 0.2)는 학습에 이용되지도 못한 채 제출해야 하는 상황이 벌어졌고 이것이 너무 아깝다고 생각하였다. 그래서 train set으로 학습을 마치고도 내가 원하는 epoch만큼 valid set도 학습시키는 기능을 구현하였다. 어떤 조건으로 실험을 해보든 간에 valid set을 학습에 추가한다면 F1 score가 항상 증가하였고 2 epoch 기준 약 0.02 증가하였다. SOTA에도 이 방법을 적용하였다. 학습에 사용되는 데이터의 수를 늘리는 것이 성능 향상에 거의 항상 도움이 된다고 느꼈다.

### B) 비율이 랜덤한 CutMix 구현

기존 우리의 CutMix는 세로 방향으로 절반씩 자른 사진을 학습에 이용했다. 이에 항상 절반으로 자른 것보다 0.4, 0.6 / 0.55, 0.45처럼 CutMix 될 때마다 비율을 다르게 하면 더 좋은 효과가 나지 않을까라고 생각했다. 이에 내가 원하는 range를 입력해 놓으면 CutMix할 때마다 그 사이의 random 실수 lam을 뽑고 lam : 1-lam의 비율로 CutMix하는 기능을 구현하였다. 0.46~0.54의 range를 넣는 것이 가장 효과가 좋았으며(이것보다 더 큰 범위를 주게 되면 얼굴이 거의 다 손실된 채로 CutMix가 이루어진다.) 기존 CutMix보다 F1 score가 0.01~0.005 증가하였으며 SOTA에도 이 방법을 적용하였다.

### C) 앙상블

A와 B 방법에 F1 loss까지 적용했을 때 단일 모델로 가장 성능이 높게 나온 것이 F1 score 0.7875였다. 더 이상의 하이퍼 파라미터를 조절하는 실험으로는 이 스코어를 넘는 것이 불가능하다고 생각하여 hard voting을 진행하였다. 이때 비슷한 방법을 적용한 모델끼리 앙상블 시키는 것은 의미가 없다고 판단하여 (F1 loss, MTCNN), (CrossEntropy, top-2 Soft Voting Ensemble, MTCNN), (F1 loss, CentorCrop) 각각 팀 내 성능 1, 3, 5위를 기록하고 있지만 주요 특징은 다르다고 판단한 3가지 모델을 앙상블 시켰고 목표로 삼은 F1 score 0.8을 달성할 수 있었다. Hard voting 만으로도 유의미하게 성능을 상승시킬 수 있다는 것을 알 수 있었다.

## 3. 마주친 한계와 아쉬운 점, 다음 프로젝트에서 새롭게 시도할 것

모델이 age 판단을 잘하지 못하였는데, 이를 3-way로 age classifier의 성능을 끌어올리는 방향으로 해결하려 했으나 기존 것보다 좋은 성능을 내지 못했다. 또 아직 코드의 모듈화에 익숙하지 않아서 새로운 기능을 구현했을 때 그것이 팀원들의 코드에 잘 흡수되도록 짜는 것이 힘들었다. 다음 프로젝트에서는 더 구조화된 코드를 짜는 능력이 필요할 것 같다.

## [개인 회고] T4095 배서현

### 1. 학습 목표

- a. 말은 바 최선을 다하되 결과가 좋지 않더라도 실망하지 않기
- b. 말은 파트(EDA/Data Augmentation)에 적극적으로 참여하기

### 2. 프로젝트 과정

- a. 베이스코드에서 제공되었던 augmentation보다 더 나은 augmentation을 적용하기 위해 다양한 transform을 직접 적용하며 실험을 진행했다. Affine transformation이나 color transformation은 데이터셋의 특성에 적합하지 않다는 결론을 낼 수 있었다.
- b. 직접 구현한 Randaugment를 적용하고 싶었다. 기존 library에서 지원하는 Randaugment는 task 해결에 도움을 주지 않을 것 같아 새롭게 구현하고자 했다. 대회가 끝난 후 구현에 성공하여 모델에 적용해 본 결과 overfitting을 방지했으며 기존의 augmentation을 적용했을 때 보다 성능이 개선되었다.
- c. EDA를 취합한 후 문서화 시켰다.

### 3. 회고

- a. 모르는 부분에 대해 막연히 어렵다고 생각하며 회피했다. 따라 말았던 부분 외의 코드를 이해하지 못했고, 모르는 것이 생겼을 때에도 모두가 아는 것을 묻는 것 같아 망설였다. 협업에 임하는 자세가 잘못되어 있었고, 이를 반성한다.
- b. 대회가 끝난 후 Randaugment를 구현하는 것에 성공하고 모델에 적용한 일은 스스로도 칭찬할 일이라 생각한다. 그러나 다음 대회에서는 대회가 진행되는 시간 내에 적용할 수 있도록 목표를 구체적으로 설정해야겠다.
- c. 부스트캠프를 시작하고 처음 진행하는 프로젝트여서 그런지 협업 다운 협업을 하지 못했다. 이번의 경험을 발판삼아 다음 대회에서는 한층 더 체계적으로 협업을 진행하자는 이야기를 나누었다. 특히 GitHub를 기술적으로 이용하지 못했다는 아쉬움이 남는다.

### 4. 개선 방안

- a. 다음 프로젝트에서는 새로운 model을 적용해보거나 train 방법을 제시하는 등 이번 프로젝트에서 어렵게 느껴졌던 것들에 도전할 것이다. 그러기 위해 새로운 대회를 시작하기 전까지 기존의 베이스 코드와 팀 코드를 다시 살펴보기로 했다.
- b. 내가 맡은 일이 아니더라도, 어려워 보이더라도 모르는 것이 있다면 망설이지 않고 질문하며 명확히 이해할 것이다. 또한 원활한 협업을 위해 팀원들과 적극적으로 소통하는 자세가 필요한 것 같다.
- c. GitHub를 활용하는 방법을 배우고 익숙해지는 시간을 가질 것이다. 그와 함께 다음 프로젝트에 활용할 GitHub 템플릿도 찾아 볼 예정이다. 그리고 GitHub 뿐만 아니라 다른 협업 도구도 미리 사용해 볼 것이다.



**- 이번 프로젝트에서 나의 목표는 무엇이었는가?**

목표는 협업하는 방법을 배우는 것이었다. 서로 다른 목표들을 어떻게 취합해서 공동의 목표를 만들지, 아이디어가 생각났을 때 어떤 식으로 전달해야 할지 같은 방법들을 배우고 싶었다. 또한 딥러닝 프로젝트를 팀 프로젝트로 진행한 것은 처음이라 코드나 실험 결과를 어떻게 관리할 지에 대해서도 궁금했다.

**- 나는 내 학습목표를 달성하기 위해 무엇을 어떻게 했는가?**

피어세션 때 나름대로 의견 제시를 많이 하려고 했고, 실험한 결과를 Notion 에 모두 정리하였다. 코드도 최대한 다른 팀원들이 바뀐 부분이 어딘지 알아보기 쉽게 주석을 표시하거나 새로운 파일을 만들어 함수를 정의하려고 하였다.

**- 나는 어떤 방식으로 모델을 개선했는가?**

Sampler 코드를 추가하였다(성능 실험은 하지 못했다). 다양한 loss 함수에 따른 성능 변화를 테이블로 정리하여 loss 함수를 정하는 데에 기여하였다.

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

Data imbalance 의 문제 일부를 해결하고, loss 함수 선택에 도움을 주었다. 하지만 제대로 된 문서화와 체계적인 Github 관리를 하지 못하여 모든 실험을 정리하지 못했다는 사실을 끝내고 나서야 깨닫게 되었다. 다음 프로젝트에는 더 체계적으로 실험 내용을 정리하고, 폴더나 파일, Github 를 관리해야겠다는 생각이 들었다.

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

이전에 진행했던 프로젝트와는 다르게 팀 프로젝트였기 때문에 Github 을 처음 사용해봤다. 처음 사용해보는 것이었기 때문에 Github 협업 문화가 아직 잘 형성되지 않았었고, 내 이름으로 만든 branch 에 지금까지 코딩한 파일들을 업로드하는 것이 전부였다. Github 을 사용하는 데에도 나름의 규칙이 필요하고 체계가 있어야 한다는 점을 배웠다.

**- 마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?**

어떤 문제에 대해서 새로운 해결책을 제시하는 능력이 부족하다고 느꼈다. 새로운 해결책이라 함은 직접 코딩을 해야할 수도 있고, 처음보는 라이브러리를 사용해야할 수도 있는 경우를 의미하는데, 그런 점에서 무의식적으로 해보고 싶지 않다고 느꼈던 것 같다.

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

도전해보는 것이 다음 프로젝트의 목표이다. 직접 성능에 기여를 하든 안하든 시도해보고 거기서 또 다른 방향을 찾아 도전해볼 것이다.

**- 이번 프로젝트에서 나의 목표는 무엇이었는가?**

딥러닝을 이용한 프로젝트는 처음이라 전반적으로 코드가 어떻게 돌아가는지 알고 싶었고 end-to-end 로 코드를 분석해보고 싶었다. 아직 딥러닝에 대한 이해가 부족해서 최대한 내가 할 수 있는 선에서 성능 개선을 하려고 했다.

**- 나는 내 학습목표를 달성하기 위해 무엇을 어떻게 했는가?**

baseline 코드를 분석하여 전반적인 흐름에 대해 익혔고 내가 개선시킬 수 있는 부분에 대해 고민했다. 모델을 구성하는 것 보다 데이터를 다루는 부분이 익숙해서 외부 API 를 활용하여 데이터를 분석하고 잘못된 부분을 수정해 나갔다.

**- 나는 어떤 방식으로 모델을 개선했는가?**

inference 시에 결과로 나온 top 2 개의 label 을 가지고 ensemble 기법을 적용하여 성능을 향상시켰고 데이터 mislabeled 된 부분을 분석하여 수정하였다.

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

ensemble 기법으로는 약간의 성능 향상이 있었고, re-labelling 은 성능 향상을 보지 못했다. 후자의 경우 바뀐 데이터 개수가 많지 않아서 그런 것 같았고 데이터를 좀 더 정교하고 정확하게 정제할 수 있는 방법을 깊게 고민해봐야겠다.

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

처음 배운 딥러닝이고 처음 해본 딥러닝 프로젝트여서 비교 대상이 마땅치가 않아 성공적으로 끝마친것에 의미를 두고 싶다.

**- 마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?**

데이터를 가공하여 성능을 올리는 것이 생각보다 어렵다는 것을 느꼈고 모델을 새로 적용시켜보는 것을 해보지 못했다. 협업하는 과정에서도 각자 수행한 코드를 합치는 과정이 매끄럽지 못했고 따로 수행하는 느낌이 있었다.

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

다음 프로젝트때는 데이터를 직접 만드는 방법 등으로 더욱 다양하게 시도를 해보고 적절한 모델을 찾아 적용시켜보고 싶다. git을 좀 더 익혀서 협업을 통해 모델의 성능을 좀 더 올릴 수 있게 하고 싶다.