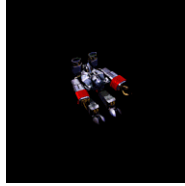


# CV-14 랩업 리포트

-Level-1 Image Classification 대회 리뷰-

CV-14 조 팀 SCV



## 1. 프로젝트 개요

COVID-19의 확산으로 우리나라는 물론 전 세계 사람들은 경제적, 생산적인 활동에 많은 제약을 가지게 되었습니다. 감염자의 입, 호흡기로부터 나오는 비말, 침 등으로 전파가 되기 때문에 마스크를 착용하는 건 COVID-19의 확산을 방지하는데 중요한 역할을 합니다. 마스크 착용 여부의 검사를 도와주기 위해 카메라로 비춰진 사람의 얼굴 이미지만으로 마스크를 쓰고 있는지, 쓰지 않았는지, 정확히 쓴 것이 맞는지 자동으로 가려낼 수 있는 모델을 제작하는 것이 프로젝트의 목적입니다

## 2. 팀 역할 및 프로젝트 팀 목표

이름	역할
김승기	EDA, Hyper Parameter, Augmentation
김준태	EDA, 데이터 전 처리, Modeling
이태경	데이터 전 처리, Augmentation, K-fold
전형우	EDA, Augmentation, Modeling
정호찬	EDA, 데이터 전 처리, Modeling

### 프로젝트 팀 목표

- Leader Board 성능 보다는 Robust 한 모델을 위한 설계를 해보자
- 우리가 직접 구현해보고 실험할 수 있는 것들에 시간을 투자하자

## 3. 협업

### A. 협업 방법

- Notion 과 Wandb 를 통한 프로젝트 진행 과정 및 실험 결과를 공유
- Git 과 Git hub 를 활용하여서 개인이 작성한 코드를 공유하고 서로 리뷰
- Zoom 을 활용하여 실시간으로 소통하면서 협업진행

### B. 개발 환경 및 Tool

- 개발언어 : Python
- 개발환경 : V100 32GB GPU
- Frame Work : Pytorch
- 협업툴 : Wandb, Slack, Notion, Git

## 4. 프로젝트 실험 내용

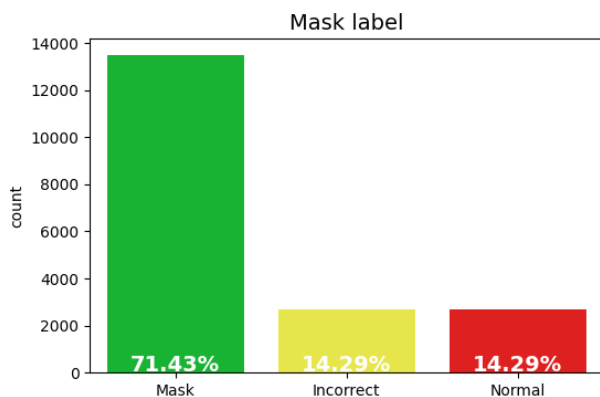
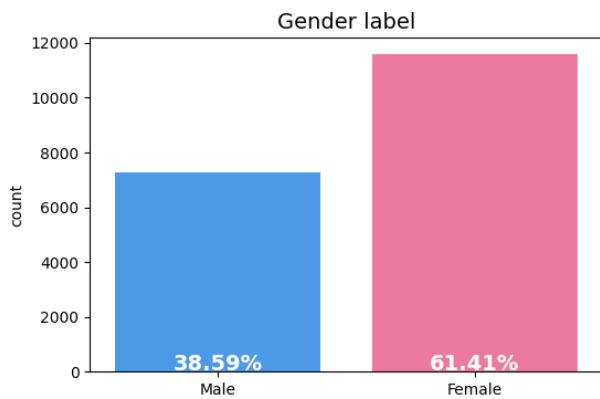
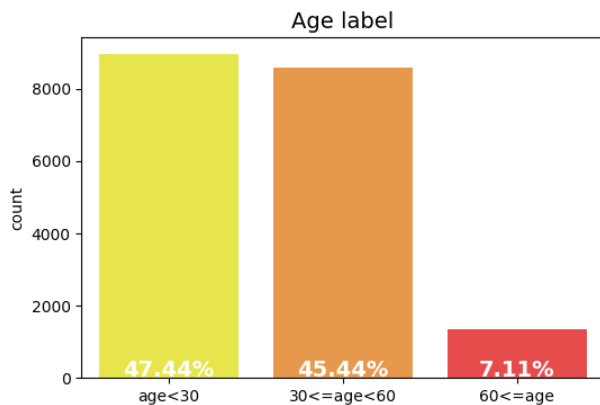
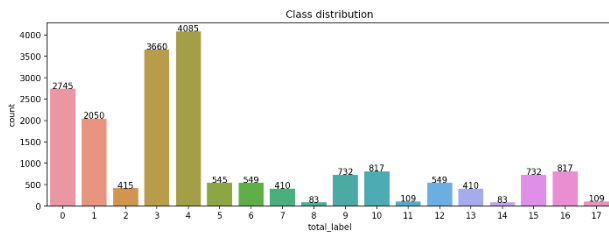
자세한 실험 내용 :

<https://bottlenose-oak-2e3.notion.site/wrap-up-report-51cfca5542d243eaaf4ce1e4a62484ad>

### (1) EDA

#### ✔ Label Imbalance

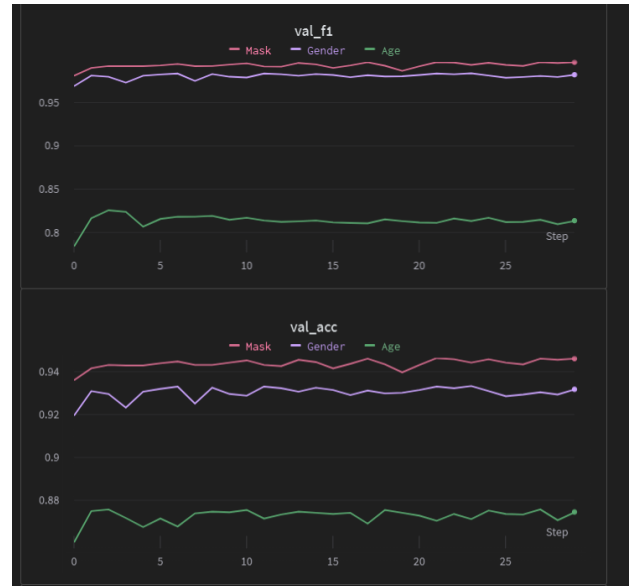
Age, Mask, Gender 의 모든 Label 이 불균형한 상태를 이루고 있었음



-접근방법 : 1) 전처리 & 데이터 증강을 통한 해소  
2) Train/Validation set 을 균형 있게 구성

### ✓ Single Model Test

Age, Mask, Gender 의 Label 를 각각 예측하는 Single Model 을 구성하여서 어떤 Label 을 예측하기 어려운지 난이도를 측정, 측정 결과 Age 가 다른 Mask, Gender 에 비해서 잘 맞추기 못하는 것으로 확인



-접근 방법 : 1) 해결해야 할 주요한 목표는 'Age' 예측

### ✓ Background

이미지 데이터에서 중요한 정보가 있는 인물의 위치는 대부분 정중앙에 위치하고 있음, 또한 인물 뒤에 배경이 데이터 마다 달랐음

-접근방법 : 1) Background 가 noise 가 될 수 있다

2) CenterCrop 과 같은 방식을 활용해볼 수 있다

## (2) Modeling

### ✓ Hyper Parameter

Loss, Optimizer, Lr 는 경험적 실험 결과 비슷한 성능을 보여서 아래와 같이 고정

- Loss : Focal Loss – 데이터 불균형을 해소하기 위해서
- Optimizer : Adam
- Lr : 0.0001

### ✓ Base Model : EfficientnetB4

**모델 선정이유** : Transformer 계열의 경우 많은 양의 데이터가 필요하다 하지만 우리가 갖고 있는 데이터의 수는 충분하지 않다. 이를 Pretrained 된 모델의 가중치를 가져와서 해결해 볼 수 있지만 우리가 갖고 있는 데이터와 Pretrained 에 활용되는 데이터의 유사도가 적기 때문에 Transformer 계열은 효과적이지 않을 수 있다. 또한 서비스를 제공하는 관점에서도 Transformer 계열의 경우 학습/추론에 많은 시간이 소요되기 때문에 효과적인 학습과 적은 데이터에서도 괜찮은 성능을 내기 위해서 CNN 계열의 모델로 방향을 잡았다

Method	Accuracy	F1 Score
<b>EfficientnetB4</b>	<b>0.8651</b>	0.7073
<b>Resnet101</b>	0.6857	<b>0.7526</b>
<b>Densenet161</b>	0.6931	0.7226

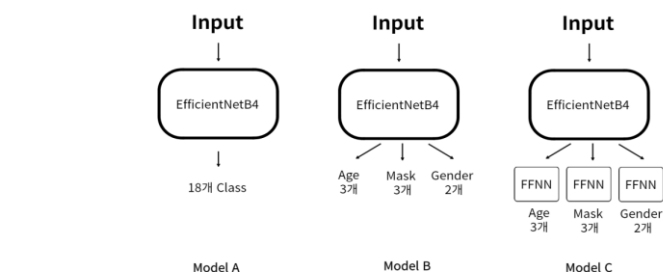
따라서 CNN 계열의 Densenet, Resnet, Efficientnet 을 비교한 결과 Efficientnet 이 꾸준히 좋은 성능을 보였고 이미지 데이터 원본 사이즈가 (384, 512)이기 때문에 원본 사이즈와 비슷한 (380, 380)을 input 으로 사용한 Efficientnet 을 선택하였다

Method	Accuracy	F1 Score
<b>EfficientnetB0</b>	0.6836	0.7265
<b>EfficientnetB2</b>	0.6894	<b>0.7531</b>
<b>EfficientnetB4</b>	<b>0.8651</b>	0.7073

또한 Efficientnet 의 크기별로 비교를 했을 때 F1 Score 는 큰 차이가 없었지만 Accuracy는 B4가 다른 모델에 비해서 압도적으로 높았기 때문에 B4 를 Base Model 로 채택하였다

### ✓ Modeling

Modeling 은 크게 3 가지 방식으로 나누어 진행하였다



-**Model A** : One Way Model 로 18 개의 Single Label 를 예측

-**Model B** : Three Way Shallow Model, 3 개의 Shallow 한 Branch 를 적용하여 Multi Label 를 예측

-**Model C** : Three Way Deep Model, 3 개의 Deep 한 Branch 를 적용하여 Multi Label 을 예측

결과 : (validation Score 기준)

Method	Accuracy	F1 Score
<b>Model A</b>	0.8929	0.8953
<b>Model B</b>	0.8942	0.8842
<b>Model C</b>	<b>0.9058</b>	<b>0.9254</b>

Model A 와 Model B 의 경우 성능차이를 보이지 않음

Model C 의 경우 Validation Score 가 가장 높았으나 Test Score 는 가장 낮았음 - **과적합 된 것 같음**

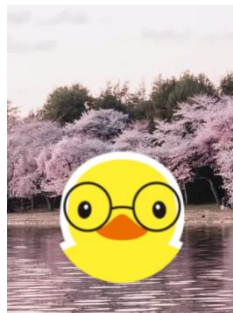
**해결방법** : Model A 중 가장 높은 Test 점수를 기록한 Model 을 가져와서 Freeze 한 다음 Model C 의 Branch 만 학습 (validation Score 기준)

Method	Accuracy	F1 Score
<b>Model A + C</b>	0.8854	0.8884

Model A 와 Model B 와 비교했을 때 Validation Score 는 차이를 보이지 않았지만 Test Score 는 높았음

### (3) Data Preprocessing

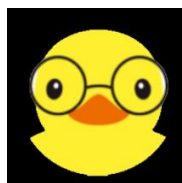
#### ✓ rembg & deepface



원본



rembg



DeepFace

-**Rembg** 라이브러리를 활용하여서 뒷 배경을 제거

장점 : Noise 가 제거되어 학습이 쉬워짐

단점 : 학습 이외에 추가적인 시간 소요 학습이 쉬워진 만큼 과적합 위험 증가

-**Deepface** 라이브러리를 활용하여서 얼굴을 탐지하고 Crop 함

장점 : Noise 가 제거됨

단점 : 학습 이외에 추가적인 시간 소요

결과:

Method	Accuracy	F1 Score
CenterCrop + Resize	0.8704	0.8644
rembg	0.867	0.8628
rembg + Deepface	<b>0.8948</b>	<b>0.918</b>

Rmbg와 Deepface 를 같이 사용했을 때 Validation, Test Score 가 모두 상승하였음

#### ✓ 50 대 후반 (57,58,59) 데이터 제거

제거이유 :

1) 60 대 경계에 위치한 값들이 60 대 범주보다 분포가 높기 때문에 분류를 방해한다고 파악

2) 50 대 후반의 데이터를 60 대 범주로 주는 것은 잘못 Labeling 된 데이터를 증가시키는 것으로 판단하였음, 따라서 이를 제거

결과 : (validation Score 기준)

Method	Accuracy	F1 Score
Basic Data set	0.8794	0.796
57~59 Remove	0.862	<b>0.8457</b>

➡F1 Score 가 Validation, Test 에서 모두 상승

#### ✓ Train / Validation Set 구성

- 'Random Split' : 모든 데이터를 랜덤적으로 Split - Train/Val

## 에 동일한 정보 존재 문제

- 'Split By Profile' : 사람 기준으로 데이터를 Split - **데이터가 불균형하게 Split 될 수 있음**

- 'Class balance Split By Profile' : 사람을 기준으로 class 별 stratify 를 유지하면서 Split - 위의 두 방법을 어느정도 해결

## (4) Augmentation

### ✓ Mixup



Mix up

60 대 데이터가 상대적으로 적은 불균형 상태를 해결하기 위해서 Off-line 으로 Mix up 을 진행하여서 60 대 데이터만 증폭 시켜 줌

결과: (validation Score 기준)

Method	Accuracy	F1 Score
Only Basic data set	0.862	0.8457
Use Mix-up data set	<b>0.8746</b>	<b>0.8792</b>

Validation, Test Score 가 상승하였음

하지만 추가적인 Augmentation 을 같이 활용하면 과적합 되는 경우도 있었다. 따라서 이를 옵션적으로 활용하여 Mix up Data 를 활용할 때는 다른 Augmentation 을 사용해주시 않음

### ✓ Cutmix



Basic Cutmix



Custom Cutmix

과적합을 피하기 위해서 Cut mix 를 추가적으로 활용

Validation Score 는 증가하였지만 Test Score 는 하락

-과적합이 되었음

따라서 Label 정보를 어느정도 보존해주면서 과적합을 방지할 수 있게 세로로 5:5 로 나누어 Custom Cut mix 진행

결과 : (validation Score 기준)

Method	Accuracy	F1 Score
Basic Augmentation	0.8704	0.8644
Basic Cutmix	0.8906	<b>0.8999</b>

Custom Cutmix	<b>0.8929</b>	0.8953
---------------	---------------	--------

Custom Cutmix 가 Basic Cutmix 와 사용하지 않았을 때 보다 Validation, Test Score 가 모두 상승

## (5) Ensemble

### ✓ Test Score 를 기준 Ensemble

Test Score 를 기준으로 3 개의 모델을 선택해서 Hard Voting 진행

### 선택한 모델

- Model A +57~59 Remove + Mix-up data
- **(Same Above)** + Split By Profile
- Model C + 17~20, 57~59 Remove

✓ Validation 예측 분포를 기준으로 Robust 예측을 하는 모델을 선택하여서 Hard Voting 진행

### 선택한 모델

- Model A +57~59 Remove + Mix-up data
- Model A +57~59 Remove + Custom Cutmix
- Model (A + C) + 57~59 Remove + rembg & deepface

결과 : (Test Score 기준)

Method	Accuracy	F1 Score
Ensemble 1	0.8126	0.7596
Ensemble 2	<b>0.8261</b>	<b>0.7868</b>

Ensemble 2 가 1 보다 높은 Test Score 를 보임

조금 더 Robust 한 모델이 만들어지지 않았나 추론

## 5. 결과

### ✓ 최종결과

#### - Public Score

순위	팀 이름	팀 멤버	f1 ↕	accuracy ↕
1 (-)	CV_14조		0.7868	82.6190

#### - Private Score

순위	팀 이름	팀 멤버	f1 ↕	accuracy ↕
1 (-)	CV_14조		0.7722	81.4762

Public Score 에 비해서 Private Score 가 소폭 감소하였지만

★ “최종순위 1 위 기록” ★

## 6. 팀 회고 및 개선방안

### ✅ 잘했던 점

- 팀원들끼리 자유로운 아이디어의 공유가 좋았다
- 대회에 익숙하지 않은 팀원들도 잘 적응할 수 있었다
- 성능에 집착하지 않았다
- 가설을 세우고 이를 검증하면서 모델링을 진행한 것이 좋았다
- 회의를 통해서 역할을 분담하고 체계적으로 진행한 것이 좋았다

### ✅ 아쉬웠던 점

- 개인마다 진행 속도가 차이가 나서 아쉬웠다
- Git 과 같은 협업툴을 적극적으로 활용하지 못해서 아쉬웠다
- 실험 공유에 Convention 을 정하지 못해서 아쉬웠다
- Python Code 버전 관리를 하지 못해서 동일한 조건을 맞추기 힘들었고 새로 구현한 기능의 호환성을 맞추기 힘들었다

## 7. 개인 회고

### ★ 김승기

#### - 내 개인적인 학습 목표

⚡ 저는 먼저 프로젝트 workflow 를 이해하고 협업 문화를 경험하는 것이 첫 목표였습니다. 첫 프로젝트인 만큼, 어떤 흐름으로 진행이 되고 팀원들과 정보를 공유하면서 커뮤니케이션을 경험해보려고 노력했습니다. 두 번째 목표는 지금까지 배운 기초 CV 지식들을 프로젝트를 통해 한번 적용해보면서 이론과 실전 사이의 차이를 느껴보려고 노력했습니다.

#### - 내 학습 목표를 달성하기 위해 한일

##### 1. baseline code 이해 및 customize

저는 주어진 baseline code 의 train.py 와 dataset.py 를 위주로 데이터 파이프라인을 이해 하려고 했습니다. 또한 실험의 편의성을 위해 CV 기초 프로젝트 강의를 참고해서 config parser 를 이용하는 방법을 적용해봤습니다. 마지막으로 실험의 자동화를 위해 셀 스크립트를 구현했습니다.

##### 2. EDA

데이터를 파악하기 위해 클래스의 분포를 확인해서 특정 클래스의 imbalance 를 파악했습니다. 시각화를 통해

파악한 데이터를 작업 육안으로 확인하면서 어떤 관계나 특징을 파악하려고 노력했습니다.

### 3. Hyperparameter 실험

Optimizer, scheduler, criterion 그리고 learning rate 에 대한 실험을 진행했습니다. 변인 통제를 한 실험을 통해 성능이 좋은 것을 골라서 모델에 적용했습니다.

### 4. Augmentation 실험

Augmentation 이 적용되지 않은 base augmentation 을 다양한 augmentation 기법들을 따로따로 그리고 묶어서 실험을 진행해서 성능을 비교했습니다. 특정 augmentation 을 적용할 경우 성능이 올라가기도 하지만, 대체로 여러가지 적용할 경우 학습 난이도가 올라가서 특정 validation accuracy 에 도달하기 위해서 더 많은 epoch 이 필요하거나 아예 도달하지 못 할 수도 있다는 것을 확인했습니다.

#### - 학습 목표를 달성하기 위한 과정 중 깨달은 점

프로젝트를 진행하면서 협업이 정말 중요하다는 것을 깨달았습니다. 혼자서 하는 일을 여러 명이 힘을 합치면 훨씬 일을 효율적으로 처리 할 수 있다는 것이 가슴에 와 닿았습니다. 지금 생각해보니 모델링, 데이터 처리 아니면 베이스라인 코드 셋업 같은 것은 어떻게든 혼자서 노력해서 공부할 수 있는 것이지만, 협업은 기회가 주어졌을 때 열심히 경험 해봐야겠다는 생각이 들었습니다.

#### - 마주한 한계

머신러닝 기법이나 통계적인 추론을 통해 문제 해결에 도움이 될만한 insight 를 얻을 수도 있을 것 같았지만 기본기가 부족하다 보니 시도 해볼 수 있는 방법론의 범위가 많이 좁아졌던 것 같습니다.

#### - 아쉬웠던 점

##### 1. 실험과 git 에 대한 convention 부족

기능 별로 브랜치 관리를 하고 commit 에 대한 convention 을 정하고 진행했으면 좋았을 것 같습니다. 실험의 경우에도 naming 이 통일 되지 않아서 파악하기가 어려웠습니다.

## 2. 시간 부족

시도해보고 싶은 것들은 많았지만 시간이 부족해서 다 해보지 못한 것이 아쉽습니다.

### 다음 프로젝트에서 해보고 싶은 것들

#### 1. git 협업

git-flow, trunk-based flow, github flow 등 특정 workflow 를 정해서 프로젝트를 진행해보고 싶습니다.

#### 2. mixed precision

Mixed precision 을 적용해서 실험 시간을 단축 할 것 입니다.

#### 3. 다양한 방법론 시도

공부하는 입장이기 때문에 결과에 좋지 않다는 판단이 들어도 다양한 방법론을 시도 해 볼 것입니다.

## ★ 김준태

### 프로젝트 학습 목표

- 인공지능 분야에서 처음 경험하는 협업 프로젝트였기 때문에 활발한 의사소통을 통해 팀원들의 의견을 조율하고 효율적인 협업을 통해 문제를 해결하며 프로젝트를 수행하면서 앞으로 있을 협업 활동에도 익숙해 질 수 있도록 하는 것이 목표였다.
- 주어진 문제에 대해 문제를 정의하고 배웠던 이론적인 지식을 기반으로 세운 가설을 세우고 실험을 통해 검증하는 경험을 하여 많은 인사이트를 얻는 것이 개인적인 학습 목표였다.

### 나는 내 학습 목표를 달성하기 위해 무엇을 어떻게 했는가?

- 팀원들의 EDA 결과를 종합하여 공유하고 팀원들의 의견을 수렴하여 어떠한 방향으로 팀이 나아갈 지 목표를 설정했다.
- 활발한 의사소통을 위해 팀 회의에 적극적으로 참여하였고 나의 실험 내용과 얻은 인사이트를 팀 노션을 통해 공유하였다.
- 학습한 모델 결과를 Confusion matrix 를 이용하여 나의 가설을 증명하기 위해 의도한 대로 학습이 되었는지 검증해보는 과정을 거쳤다. 이를 통해 test

score 없이도 실험 결과에 대한 인사이트를 얻을 수 있었다.

### 마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?

- 프로젝트 후반부에 Score 를 올리기자 집착한 것 같고, 부족한 시간을 활용하기 위해 체계적인 실험을 하지 못했다. 미리 시간을 효율적으로 활용할 수 있는 도구와 방법을 이용해 많은 실험을 해 놓았어야 했다고 생각했다.
- 나의 개인적인 학습 목표를 위한 실험 내용이 팀의 전체적인 프로젝트 진행 상황과 맞지 않았다. 팀원 간의 경험적인 차이가 있었기 때문에 프로젝트 진행 속도가 달랐고 이를 극복하기 위한 방법을 찾기가 어려웠다.
- 주어진 문제와 관련된 논문이나 프로젝트 내용을 찾아서 참고 했다면 우리의 실험적인 결과만으로 결정하기 어려운 내용을 의사 결정 시 많은 도움이 되었을 것 같다.

### 나는 어떤 방식으로 모델을 개선했는가?

- Data imbalance 문제를 해결하기 위해 60 대 나이 over sampling, 그 외 데이터 under sampling 하여 Data 의 balance 를 맞추려고 하였다. 결과적으로 60 대와 구별하기 힘들다고 생각한 50 대 후반의 데이터를 사용하지 않는 방법이 많은 성능 향상을 기록했다.
- 주어진 Data 를 봤을 때 Class 를 예측하기 위해선 인물의 얼굴 위치 정보가 중요하다고 생각했기 때문에 rembg 라이브러리를 활용하여 배경을 제거하였고 deepface 라이브러리를 활용하여 인물의 얼굴을 추출하는 전 처리 과정을 사용했다.
- Classification 문제에서 유용한 방법론인 Cutmix 방법 활용시 기존 방법처럼 Random 한 bbox 를 이용하여 mix 한다면 label 의 정보가 온전히 남아있기 힘들다고 판단하여 label 의 정보를 살릴 수 있도록 5:5 비율을 유지하며 세로방향으로 mix 하는 방법을 사용했다.
- mask, gender, age 에 대한 분류를 3 개의 branch 를 가진 모델을 활용하여 분류 성능을 높이도록 하였다. 하지만 branch 의 추가로 모델 파라미터 수가 증가 하여 overfitting 되는 경향이 보였고 이를 해결하기 위해 epoch 와 lr 을 줄였다. 그리고 branch 학습에



도움을 주기 위해 우리의 데이터로 학습한 좋은 성능을 내는 single model 의 가중치를 활용했다.

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

- Data imbalance 문제를 해결하기 위해 imbalanced sampler 를 train loader 에 활용해 봤지만 overfitting 되는 경향이 있었다.
- Data imbalance 문제를 해결하기 위해 60 대 데이터들을 서로 mix up 한 데이터를 학습에 활용해서 소폭 성능향상이 있었지만 overfitting 문제가 발생되기 쉬웠고 근본적인 문제 해결을 하기에는 어려웠다.
- overfitting 문제는 metric 과 loss 만으로 구별하기가 어려웠기 때문에 신뢰도 있는 validation set 을 구성하던가 혹은 따로 test set 을 만들어서 검증하는 방법을 활용해야겠다고 생각했다.
- 프로젝트 후반부에 부족한 시간에 score 를 올리고자 ablation study 방식으로 실험을 진행하지 못하였고 이로 인하여 실험 내용 정리가 어렵고 명확한 인사이트를 얻기가 굉장히 힘들어 졌다. 이전 실험 내용들을 미리 정리를 해 놓지 않은 상태에서 많은 실험이 쌓이면서 최종 모델을 선택하는 것이 어려웠다.

### 협업 과정에서 잘된 점/ 아쉬웠던 점은 어떤 점이 있는가?

#### 좋았던 점

- 오프라인으로 만나게 되어 팀원들끼리 많이 친해졌고 덕분에 의사소통도 더 활발히 진행될 수 있었던 것 같다.
- 가설이나 실험 결과에 대해 공유하고 토론해 보면서 다양한 관점에서의 문제를 해결하려는 생각들을 알게될 수 있어서 좋았다.
- 혼자 해결하기 어려웠던 코딩 이슈나 사용해보기 못했던 방법론에 대해 알려줘서 시간이 오래걸릴 수 있는 문제를 빠르게 해결할 수 있어서 좋았다.

#### 아쉬운 점

- 팀원 간 의사소통은 활발하게 되었지만 실험 내용, 코드를 정리하여 공유하는 것이 잘 되지 않았다.

따라서 프로젝트가 진행 될 수록 개인간의 격차가 커져서 프로젝트 진행 상황을 하나로 모으기 힘들었다.

### 한계/교훈을 바탕으로 다음 프로젝트에서 시도해보고 싶은 점은 무엇인가?

- 프로젝트 진행 전 충분한 회의와 의사 소통을 통해 팀의 목표를 설정할 것이다.
- 클린 코드를 만들고 협업 도구를 많이 활용하여 팀원들과 같이 프로젝트를 진행할 것이다.
- 실험 내용을 미리 정리하고 공유하여 다양한 관점을 통해 앞으로의 실험 내용을 정할 것이다.
- 리더보드 순위에 연연하지 않고 효율적인 프로젝트 진행을 통해 컨디션 관리를 할 것이다.

## ★ 이태경

### 개인적인 감상

- 너무 힘들었다.
- 팀원들에게 도움을 주지 못한게 많이 아쉽다.
- 아직 코딩 실력이 많이 부족하다는 걸 느꼈다
- 아는 모델이 많지 않아 모델 선정에 어려움을 느꼈으므로 모델에 대한 지식을 늘렸으면 좋겠다.
- 여러 공부를 많이 해봐야겠다는 생각을 했고 앞으로 많은 대회를 하게 될 것이기 때문에 멘탈 관리에 집중해야겠다는 생각을 했다.
- 체계적인 실험이 부족했다.
  - EDA 단계에서 어떤 실험들을 할지에 대해 알아봐야겠다

### 나는 내 학습 목표를 달성하기 위해 무엇을 어떻게 했는가

- 처음 경험해보는 대회이므로 프로젝트의 전체적인 흐름과 협업의 과정에 대해 배워야겠다고 생각했다.
- baseline code 에 대한 전반적인 이해를 확실하게 하기 위해 코드를 열심히 뜯어봤다. 이후 코드를 짜는 데에 도움을 많이 얻었다.

마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?

- 커뮤니케이션은 원만했으나 생각보다 협업이 원활하게 이루어지지 못했던 것 같다. 협업 툴도 많이 사용하지 않았다.
- 다른 사람들을 따라잡기 위해 노력하다보니 페이스가 많이 무너졌다.

한계/교훈을 바탕으로 다음 프로젝트에서 시도해보고 싶은 점은 무엇인가?

- 모델에 대해 많이 알지 못해 모델 선택에 어려움이 많았으므로 앞으로 꾸준히 논문을 읽으며 이에 대한 지식을 미리 알아놔야겠다.
- 다음 프로젝트때는 깃허브를 적극적으로 활용하며 여러 팀원들의 피드백을 들을 수 있는 기회가 있으면 좋겠다.
- 프로젝트 초기에 협업 툴의 규칙을 정확하게 만들어놓자
- 사람마다 속도가 다 다르므로 나의 페이스를 지키는 것을 목표로 하자.

나는 어떤 방식으로 모델을 개선했는가?

- 60 대의 데이터 분포가 많이 부족하여 해당 데이터들을 mixup 하여 데이터 수를 늘리는 작업을 진행했다.
- on line 상에서 데이터의 분포를 조정하기에는 여러가지 어려움이 있어 off line 으로 진행을 했는데 이 것이 오히려 과적합을 야기하는 경우도 발생했다.

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

- mixup 과정을 off-line 으로 구현을 하니 dataset 과 validation 에 데이터의 합친 결과와, 재료가 각각 존재할 수 있다는 것을 간과했다.
- 이를 고려하면, 현재 주어진 데이터들에 과적합될 수 있다는 생각을 하여 on-line 에서 구현하려고 노력했지만 끝내 해결하지 못했다.
- 이를 통해 코딩 실력이 아직 많이 부족하다는 점과, 미리 이러한 점에 대해 알 수 있었다면 시간 단축을 할 수 있었을 것 같다.
- 또한 K-fold 를 구현하기 위해 많은 시간을 투자했지만 data loader 부분이 복잡해서 실패했다.

- 복잡한 코드에 대한 명확한 이해 또는 baseline code 를 쉽게 쓸 수 있게 만들어보는 작업도 필요할 것 같다.

협업 과정에서 잘된 점/ 아쉬웠던 점은 어떤 점이 있는가?

- 아이디어를 제시하고 받아드리는 데에 거리낌이 없었다는 점이 팀원 모두가 협업에 적극적이었다.
- 하지만 github, wandb, notion 등 협업 툴을 통한 실험 과정과 실험 진행, 결과 등에 대해 공유가 많지 않아서 팀원들이 어떤 실험을 어느 정도까지 진행을 했는지, 앞으로 해야할 실험은 무엇인지가 체계적이지 않아 실시간 피드백이 어려워 팀으로써의 장점이 퇴색된 것 같다.

팀 내 자신의 역할, 프로젝트에서 사용한 자신의 기술 및 지식, 모델링 및 성능 개선 등 프로젝트 전 과정에서 자신이 기여한 내용과 개인의 구체적 성과를 정리

- 팀원들이 제시하는 아이디어들에 대해 검증과 예상되는 단점에 대해 토론
- 60 대의 데이터 분포가 적으므로 데이터 분포를 늘리기 위해 mixup 을 통해 60 대 데이터의 분포를 늘렸다.
- 기초적인 모델은 중복된 데이터들로 인해 과적합이 많이 이루어졌지만, Robust 한 모델에는 1~5 퍼센트 가량 성능 향상에 도움이 됐다.

프로젝트 뿐만 아니라 동료를 통해서 내가 어떤 것을 배웠고, 프로젝트 과정에서 내가 마주한 어떤 어려움을 어떻게 극복했는지 등 자신의 이야기가 온전히 담길 수 있도록 작성해 주세요

- 여러 아이디어를 공유하며 시야가 넓어지는 경험을 했다.
- 협업을 하기 위한 협업툴의 기반 다지기가 필요하다고 생각했다.
  - 처음에 규칙을 잘 다져놓으면 팔로워가 따라가기 쉬울 것 같다.
  - 또한 리더들도 이후 코드 수정 등에 불편함을 생각한다면 이와 같은 방식이 효율적이라고 생각이 된다.
- 프로젝트에 대한 경험을 쌓는다는 생각을 하면서 참여를 했지만 프로젝트를 진행하면서 팀원들에게 큰



도움을 주지 못한 것 같아 멘탈적으로 많이 흔들렸다.

- 이를 해결하기 위해 여러가지 자기 암시를 많이 했다.
- 그리고 나니 몸은 힘들어도 마음은 그렇게 힘들지 않게 됐다.

## ★ 전형우

### 개인적 감상

- 모든게 낯설었다
- 팀에게 도움이 되지 못하고 있는 것 같아 조금한 순간이 많았다.
- 다음에는 이것보다 무조건 잘해야지 하는 생각을 많이 했다.
- 주도적으로 내가 할 일을 찾아나가는 것이 중요하다는 생각을 했다.
- 체계적으로 내 실험에 대한 관리를 해야겠다고 느꼈다.

### 나는 내 학습 목표를 달성하기 위해 무엇을 어떻게 했는가?

- 낯선 환경에서 최대한 팀과 비슷한 환경을 맞추고, 베이스 라인 코드에 대한 이해를 기반으로 천천히 건드려보려고 최대한 노력했다.
- 내가 맡은 역할인 모델 별 비교를 하기 위해 다양한 모델에 대한 이론을 정리하고, 코드에 사용하는 모델과 loss 등에 대한 공부를 진행해 '알고 쓰자'의 마인드로 접근했다.
- 팀에게 도움이 되지 못해도 방해는 되지 말자는 생각으로 팀원의 작업을 이해하려고 노력했다.

### 마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?

- 내가 수행할 수 있는 코딩이 많지 않다는 것을 느껴서 베이스 라인 코드에 대한 많은 수정을 하지 못한 것이 아쉬웠다.
- 첫 협업인데 깃을 쓰는 것이 두려워 거의 쓰지 않았고 브랜치 관리 등 여러 말을 잘 못 알아들어 아쉬웠다.

### 한계/교훈을 바탕으로 다음 프로젝트에서 시도해보고 싶은 점은 무엇인가?

- 다음 프로젝트에서는 적극적으로 실패를 두려워하지 않고 코드를 변경해보아겠다.
- 협업에서의 깃을 쓰는 방법을 열심히 배우고 익숙해져야겠다.
- 팀에게 도움이 못 된다고 조금해하지 말고 천천히 내가 할 수 있는 것을 찾아 새로운 아이디어로 팀에게 도움이 되어야겠다.

### 나는 어떤 방식으로 모델을 개선했는가?

- 다양한 모델에 대해 버전마다 어떤 차이가 있는지 생각해내고 각 모델마다 학습을 진행해 성능 비교를 수행했다.
- Transformer 를 사용한 모델을 사용해보고 이와 기존 CNN 모델과의 성능을 비교했다.

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

- Inference.py 에서 무언가 잘못되어 내 output 이 아예 이상하게 나오는 현상을 결국 수정하지 못했다. 따라서 모델 학습에 대한 흥미가 많이 떨어졌다.
- 도저히 수정 불가능한 오류가 있을때 차라리 빠르게라도 초기화를 수행해서 다시 처음부터 빠르게 쫓아가자는 생각을 해야겠다.
- dataset, dataloader 의 코드를 잘 이해하지 못해 건드릴 엄두를 내지 못했다. os, listdir 등 path 와 관련된 여러 코드에 대한 공부가 부족했던 것 같다.
- 세팅 환경에 대한 이해가 오래 걸렸던 것 같다.

### 협업 과정에서 잘된 점/ 아쉬웠던 점은 어떤 점이 있는가?

- 오프라인 만남에서 팀원들과 가까워질 수 있었고, 이 때 회의를 통해 각자가 맡을 역할을 빠르게 나눠 진행이 빨랐다.
- 각자 맡은 위치에서의 역할을 너무 잘 해주었고, 이에 대한 공유와 병합도 빨랐다.
- 깃을 사용한 병합이 아닌 줌에서 코드를 공유하고, 노션에 파일을 올리는 등 깃을 통한 실시간 commit 이 이루어지지 않은 것은 아쉬웠다.

팀 내 자신의 역할, 프로젝트에서 사용한 자신의 기술 및 지식, 모델링 및 성능 개선 등 프로젝트 전 과정에서 자신이 기여한 내용과 개인의 구체적 성과를 정리

- 어떤 모델을 써야할지 판단하는 근거를 팀원들에게 제공했다.
- 현재의 CNN 계열 SOTA 모델인 EfficientNet 에 대해 공부할 수 있었고, 이 지식을 토대로 어떤 모델이 좋을 지 판단했다.

**프로젝트 뿐만 아니라 동료를 통해서 내가 어떤 것을 배웠고, 프로젝트 과정에서 내가 마주한 어떤 어려움을 어떻게 극복했는지 등 자신의 이야기가 온전히 담길 수 있도록 작성해 주세요**

- 가장 중요하다고도 볼 수 있는 모델을 정하기 위해 열심히 성능 비교를 했지만 결국 기존에 쓰던 EfficientB4 를 쓰게 되어 내가 한일이 없는 것 같아 아쉬웠다. 이런 경우 모델에 대한 튜닝을 수행해 더 성능이 좋은 모델을 만들어보려고 노력하는 것이 필요할 것 같다.
- 생전 처음 겪어보는 대회 환경에서 적응하기 위한 시간이 너무 오래 걸렸던 것 같다. 내가 뭘 해야할 지 몰라서 봉 뜨는 시간에 내가 뭘 할 수 있는지 찾아내고 그걸 일단 헤딩해보는 것이 내 성장에 더 도움이 될 것 같다.

## ★ 정호찬

### 개인적인 감상

- 다양한 방식으로 문제접근을 해보고 EDA 를 기반으로 가설 - 추론 - 검증의 단계로 프로젝트를 진행하여서 개인적으로 많은 교훈을 얻을 수 있었다
- 처음부터 Multi Label 문제로 편견을 갖고 접근했던 것이 조금 아쉽다 - 더 다양한 접근방식이 필요했던 것 같다
- 엄청나게 어려운 난이도의 문제는 아니었지만 그럼에도 불구하고 Data Imbalance 문제를 처음 접하다 보니까 어떤방식으로 해결해야할지 확신이 생기지 않았다
- 그래도 부스트캠프 과정에서 새롭게 배운 내용과 프로젝트를 진행하면서 새롭게 공부한 내용을 실전에 적용해볼 수 있어서 많은 발전을 할 수 있었다
- 또한 협업이라는 어려운 문제를 다루면서 나의 한계점도 느낄 수 있었고 이를 통해서 개선방향도 탐색할 수 있었다

### 나는 내 학습 목표를 달성하기 위해 무엇을 어떻게 했는가?

- 대회의 리더보드의 순위보다는 Robust 한 모델을 구성하는 능력을 달성 목표로 삼았다
- 또한 팀원들과 협업을 통해서 협업 능력을 향상시키는 것을 목표로 삼았다
- 이를 위해서 데이터적인 방식으로 모델링에 접근하였고 내가하고 있는 연구 내용과 의견을 시간이 있을 때 마다 서로 공유하면서 수정하고 검토하였다
- Loss, Optimizer, Lr 과 같이 직접적으로 구현하지 않는 하이퍼 파라미터 실험에 많은 시간을 소모하기보다는 직접 커스텀할 수 있는 Modeling, Augmentation, 전처리에 많은 집중을 하였다

### 마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?

- 인공지능에 대한 지식과 프로그래밍에 대한 지식이 개인마다 차이가 있었기 때문에 이를 하나의 흐름으로 통합하는 것이 힘들었다
- 결국 프로젝트 초반에는 어느 정도 비슷한 속도로 흘러갔지만 중반 이후부터는 격차가 벌어지는 것 같았다
- 막상 대회 후반부로 갈 수록 리더보드 점수에 쫓겨서 처음에 설정했던 목표를 벗어난 것 같다
- 서로 이해하고 있는 부분을 완벽하게 알 수가 없어서 통합 방법이 어려웠던 것 같다

### 나는 어떤 방식으로 모델을 개선했는가?

- Three Way 방식을 통해서 3 개의 Label 을 각각 구별하고 Loss 를 계산하는 방식으로 모델을 수정하였다
- 깊은 Layer 로 구성된 Three Way 방식의 경우 과적합의 문제가 있었기 때문에 기존에 Pretrained 된 모델을 불러와 Freeze 시키고 branch 만 학습시키는 방법으로 Modeling 을 시도하였다
- Cutmix 를 활용하여서 모델의 과적합을 해소하려고 하였다
- 50 대 후반의 경계값을 제거하고, 데이터에서 압도적으로 많은 분포를 보였던 18~20 대 데이터를 undersampling 하여서 data imbalance 문제를 해결하려고 하였다

- Age 의 범주를 변경하여서 data imbalance 문제를 해결하려고 하였다

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

- 생각보다 One Way 모델과 Three Way 모델의 성능차이가 크지 않았으며, 조금더 깊은 Layer 로 구성된 Three Way 의 경우 파라미터 수가 증가하였고 일부 데이터에 과적합 되는 결과를 보였다
- 가장 기본적인 Validation Set 구성에 신경을 많이 쓰지 못해서 객관적으로 모델링의 결과를 평가할 수단이 부족했던 것 같다
- Age 의 범주를 세분화하여서 예측 해야하는 Class 의 Imbalance 를 해결해보았지만 오히려 세분화로 인해서 예측해야 할 차원이 늘어나 모델 성능이 하락하였다
- 생각보다 Cutmix 가 오히려 성능이 하락하는 결과를 낳았다 - 하지만 팀원과 함께 랜덤적으로 Cutmix 된 데이터가 일부 Label 정보를 손실할 수 있다고 판단하여서 사진을 반으로 나눠 합쳐 Label 을 보존하는 방식으로 개선하였다
- 또한 이러한 실패를 기반으로 새로운 Solution 을 생각할 수 있었고 팀원들과 함께 교류를 하면서 내가 생각하지 못했던 점을 보완할 수 있었다

**협업 과정에서 잘된 점/ 아쉬웠던 점은 어떤 점이 있는가?**

**좋았던 점**

- 서로가 생각하고 있는 연구방향을 공유하면서 자유롭게 의견을 나누며 수정하면서 더 좋은 방향을 선정할 수 있었던 것 같다
- 서로 적극적으로 프로젝트에 참여하여서 문제를 해결하려는 의지가 좋았고 이를 통해서 프로젝트에서 좋은 성적을 낼 수 있었던 것 같다

**아쉬운 점**

- Git 과 같이 코드적인 부분의 협업툴을 적극적으로 활용하지 못했고 버전관리도 통합적으로 하지 못해서 동일한 조건에서 실험을 하지 못해 효율적인 비교가 부족했던 것 같다

- 코드를 모듈화 하지 못해서 새롭게 구현한 기능의 호환성 문제도 존재했던 것 같다

**한계/교훈을 바탕으로 다음 프로젝트에서 시도해보고 싶은 점은 무엇인가?**

- 코드를 모듈화 하여서 어느 환경에서도 동일한 조건의 실험을 진행하여 쉽게 비교할 수 있는 버전 관리를 시도해야겠다
- 팀원들이 진행하고 있는 내용을 실시간으로 파악하고 이를 통합적으로 관리하여서 모두가 함께 가는 방향으로 팀을 운영 해보아야겠다
- Git 과 같이 협업툴을 적극적으로 활용하여서 코드 버전을 잘 관리할 수 있는 협업 능력도 향상 시켜야겠다
- 체계적으로 역할을 배분하고 이를 통해서 결과를 내는 프로세스를 적용 해보아야겠다