

# CV-14 랩업 리포트

-Level-1 Image Classification 대회 리뷰-

CV-14 조 팀 SCV



## 1. Project outline

프로젝트의 큰 목적은 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 것입니다.

하지만 아래의 프로젝트 데이터 셋을 자세히 살펴보면 우리가 구별해야 하는 Label 의 경우 3 개의 하위 Class (나이, 성별, 마스크)로 구성되어 있으며, 아래의 EDA 를 통해서 확인할 수 있듯이 데이터 셋의 Label 과 Class 가 매우 Imbalance 한 것을 확인할 수 있습니다.

따라서 본 프로젝트의 최종 목적은 다음과 같습니다.

*“Imbalance 한 data set 을 활용하여 Multi-task label 을 구별하는 모델을 구성하는 것”*

## 2. Team Members and Roles

Name	Role
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

### ✓ Team Project Target

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

## 3. Project

### ✓ Detailed Project Contents :

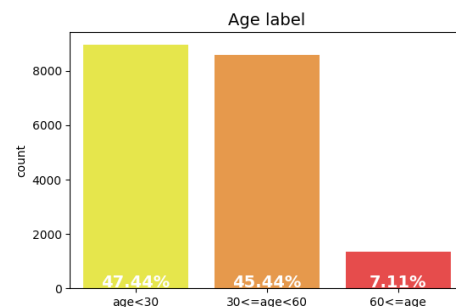
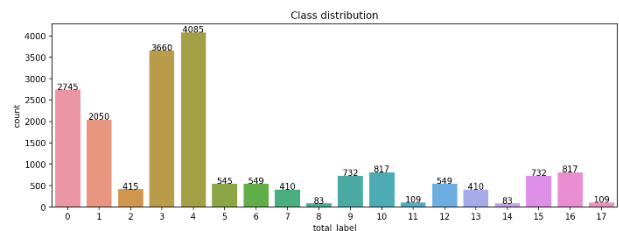
<https://bottlenose-oak-2e3.notion.site/wrap-up-report-51cfca5542d243eaf4ce1e4a62484ad>

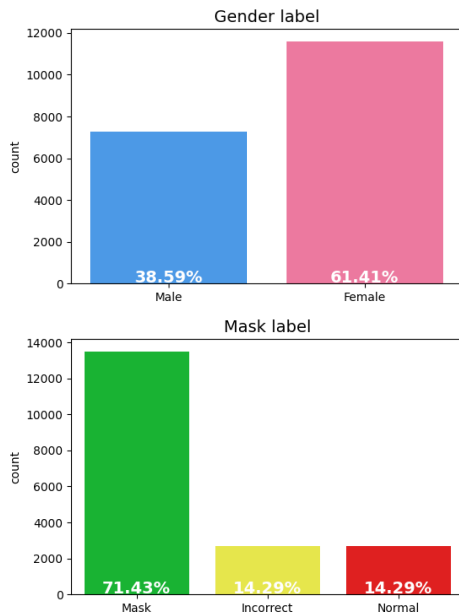
### 3.1 EDA

#### (1) EDA

#### ✓ Label Imbalance

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



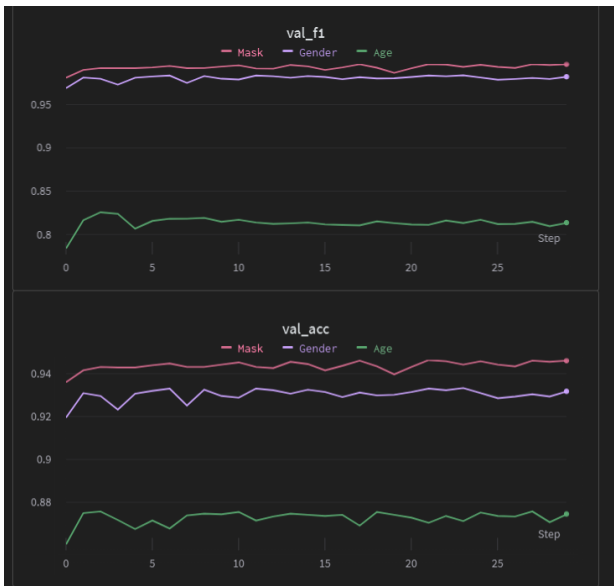


#### - 🏃 Approach :

- 1) 전처리 & 데이터 증강을 통한 해소해보자
- 2) Train/Validation set 을 균형 있게 구성해보자

#### ✅ 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



#### -- 🏃 Approach :

- 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

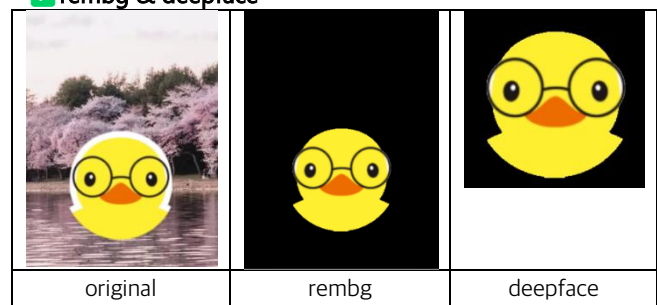
#### - 🏃 Approach :

- 1) 모델이 과소적합이 되는 경우 Background 를 지워보는 방법이나 CenterCrop 을 통해서 해결해볼 수 있을 것 같다

### 3.2 Data

#### (1) Data Preprocessing

##### ✅ rembg & deepface



#### - ★ Use Rembg and Remove Background

**장점** : Noise 가 제거되어 학습이 쉬워짐

**단점** : 학습 이외에 추가적인 시간 소요, 과적합증가

#### - ★ Use Deepface and FaceDetect & Crop

**장점** : Noise 가 제거됨

**단점** : 학습 이외에 추가적인 시간 소요

result: (validation Score)

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>

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

#### ✅ Remove late 50's

**Why Remove? :**

- 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

result: (validation Score)

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

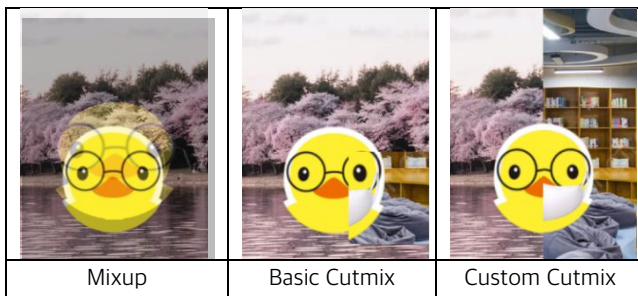
#### ✓ Train / Validation Set

- '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

## (2) Augmentation



#### ✓ Mix up

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

result: (validation Score)

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 사용해주지 않음

#### ✓ Cut mix

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

result: (validation Score)

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

## 3.3 Modeling

### (1) 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

Why EfficientnetB4 : 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 잡았고

CNN 계열의 DenseNet, Resnet, Efficientnet 을 비교한 결과 Efficientnet 이 꾸준히 좋은 성능을 보여서 Efficientnet 계열을 선택하였다

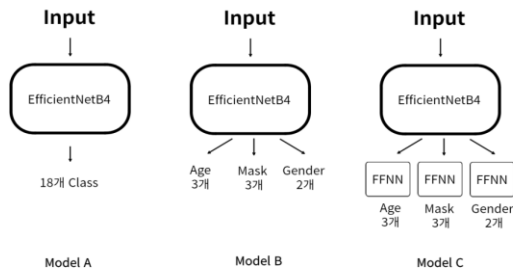
Model	Accuracy	F1 Score
<b>EfficientnetB4</b>	<b>0.8651</b>	0.7073
<b>Resnet101</b>	0.6857	<b>0.7526</b>
<b>Densenet161</b>	0.6931	0.7226

Efficientnet 논문에 나오는 그래프를 참고하여서 B4 이전까지는 성능 향상이 급격하지만 B4 이후에는 큰 성능 차이가 없다는 점을 고려하였고 Efficientnet 모델 별 실험을 통해서 B4 모델이 좋은 성능을 보여서 Base Model 로 채택하였다

Model	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

## ✅ 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

result: (validation Score)

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**

- **Plus Solution** : Model A 중 가장 높은 Test 점수를 기록한 Model 을 가져와서 Freeze 한 다음 Model C 의 Branch 만 학습

result: (validation Score)

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

## (2) 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 모델

을 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

## 4. Final Score

### ✅ 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 1위 기록”**

## 5. Cooperation

### 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

- ii.        개발환경 : V100 32GB GPU
- iii.        Frame Work : Pytorch
- iv.        협업툴 : Wandb, Slack, Notion, Git

## 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