

CV-14 SCV

Image Classification

대회 리뷰

목차



" 5명의 S급이 모인 CV팀 "

팀원 : 김승기 김준태 이태경 전형우 정호찬

1. 프로젝트 개요

2. 역할 및 목표

3. 협업

4. 프로젝트 내용

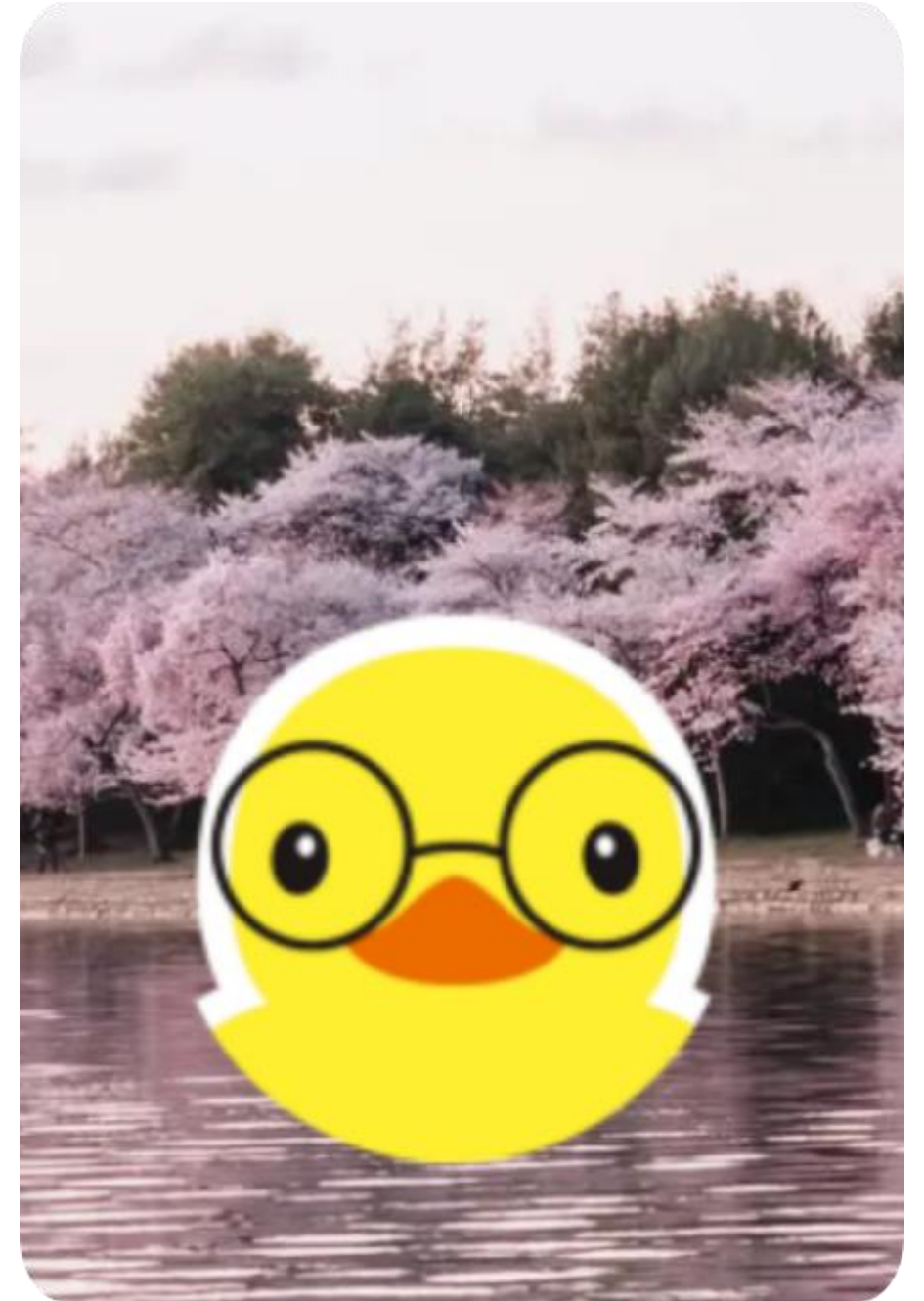
5. 회고 및 개선점

- EDA
- 모델링
- 데이터 전처리
- Augmentation
- Ensemble
- 결과

1. 프로젝트 개요

사람의 이미지만으로 **마스크 착용여부**, **나이**, **성별**을
구별하는 모델을 만드는 프로젝트

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



ex) 예시 데이터

1. 프로젝트 개요

Labeling

Aa Class 1	Mask	Gender	Age
0	Wear	Male	< 30
1	Wear	Male	>= 30 and < 60
2	Wear	Male	>= 60
3	Wear	Female	< 30
4	Wear	Female	>= 30 and < 60
5	Wear	Female	>= 60
6	Incorrect	Male	< 30
7	Incorrect	Male	>= 30 and < 60
8	Incorrect	Male	>= 60
9	Incorrect	Female	< 30
10	Incorrect	Female	>= 30 and < 60
11	Incorrect	Female	>= 60
12	Not Wear	Male	< 30
13	Not Wear	Male	>= 30 and < 60
14	Not Wear	Male	>= 60
15	Not Wear	Female	< 30
16	Not Wear	Female	>= 30 and < 60
17	Not Wear	Female	>= 60

Single Label

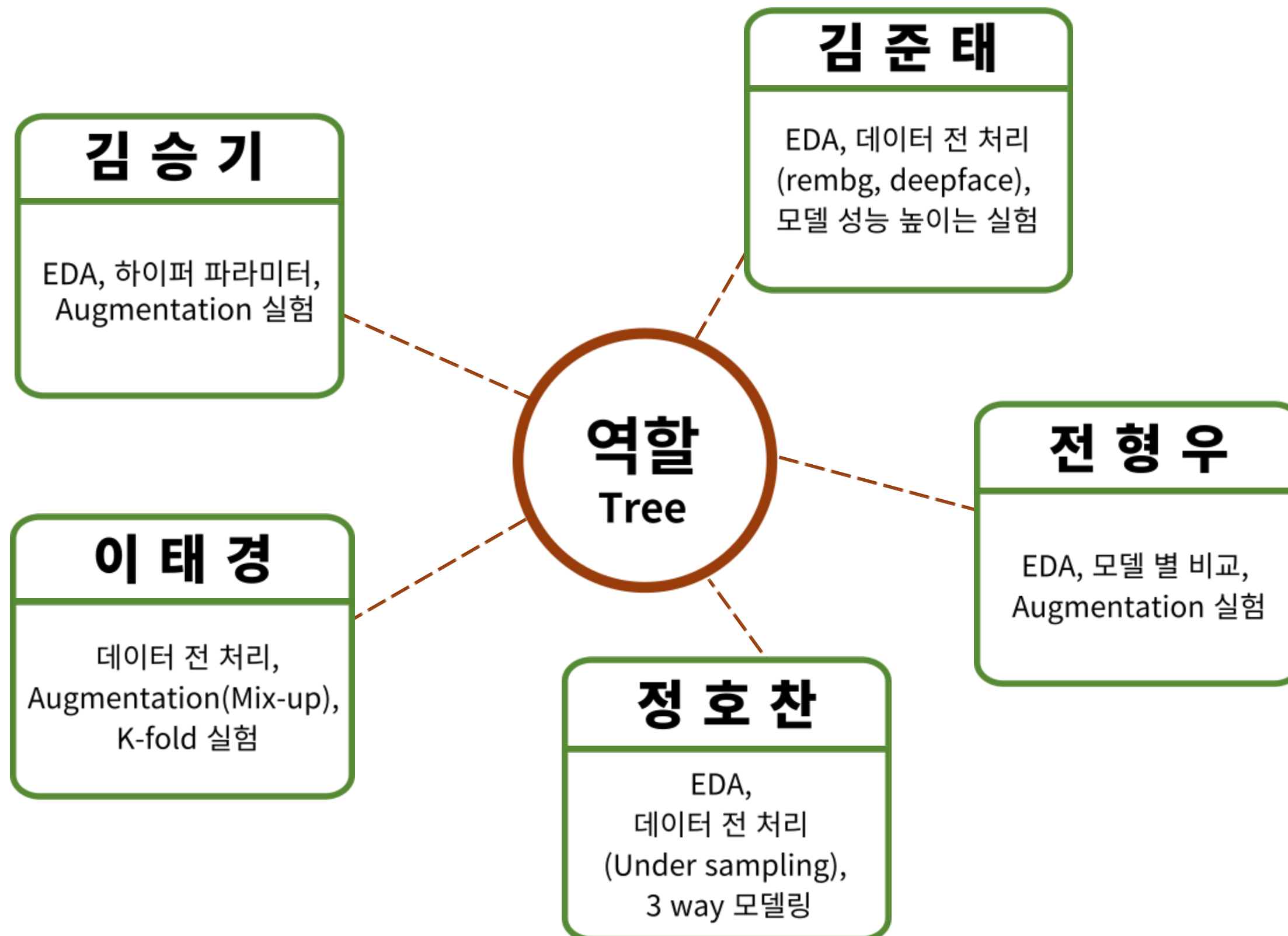


Aa Label	Age	Mask	Gender
0	Age < 30	Wear	Male
1	30 <= Age < 60	Incorrect	Female
2	60 <= Age	Not wear	

Multi Label

두가지 Label을 모두 활용해서 모델을 구성해보자

2. 역할 Tree



3. 팀 목표

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

4. 협업

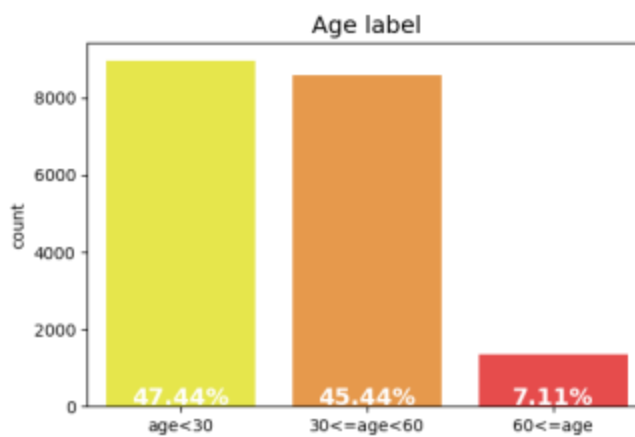
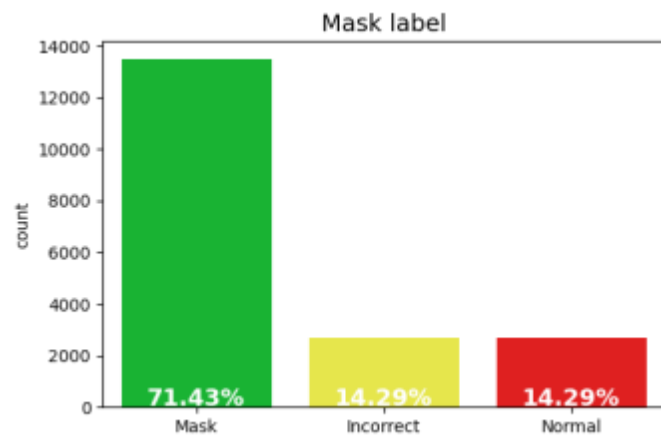
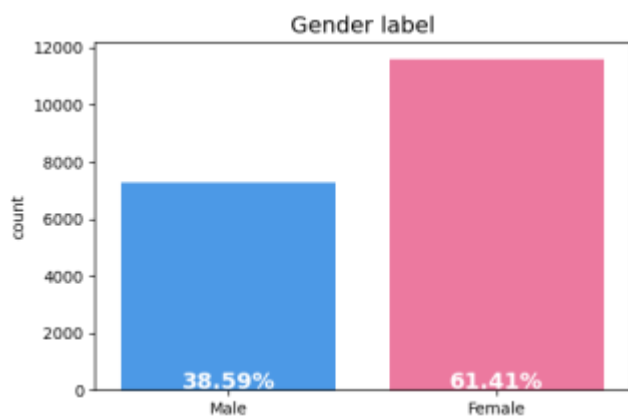
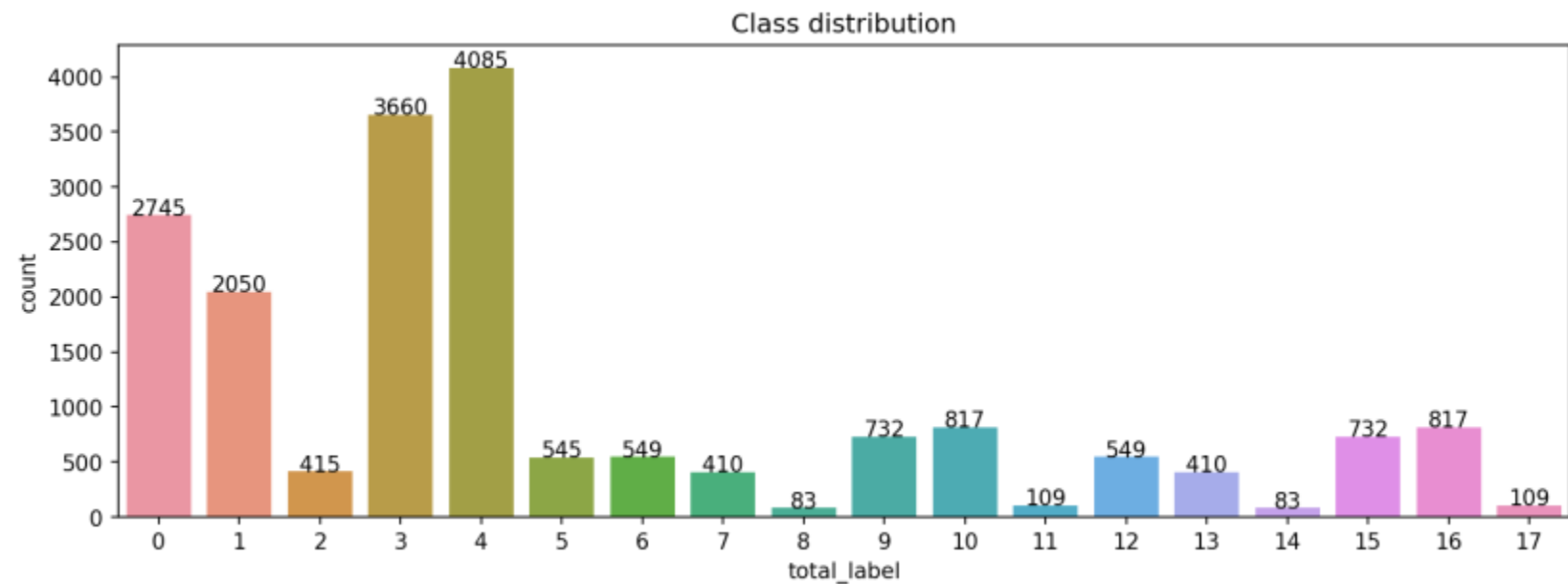
- **Notion**과 **Wandb**를 통한 프로젝트 진행 과정 및 실험 결과를 공유
- **Git**과 **Git hub**를 활용하여서 개인이 작성한 코드를 공유하고 서로 리뷰
- **Zoom**을 통해서 실시간으로 소통하면서 협업진행
- 이후 가장 좋은 모델의 코드를 선택해서 **모듈화** 및 **코드리뷰**



5. 프로젝트 내용

자세한 프로젝트 내용 링크 Link ☒

5. 프로젝트 - EDA

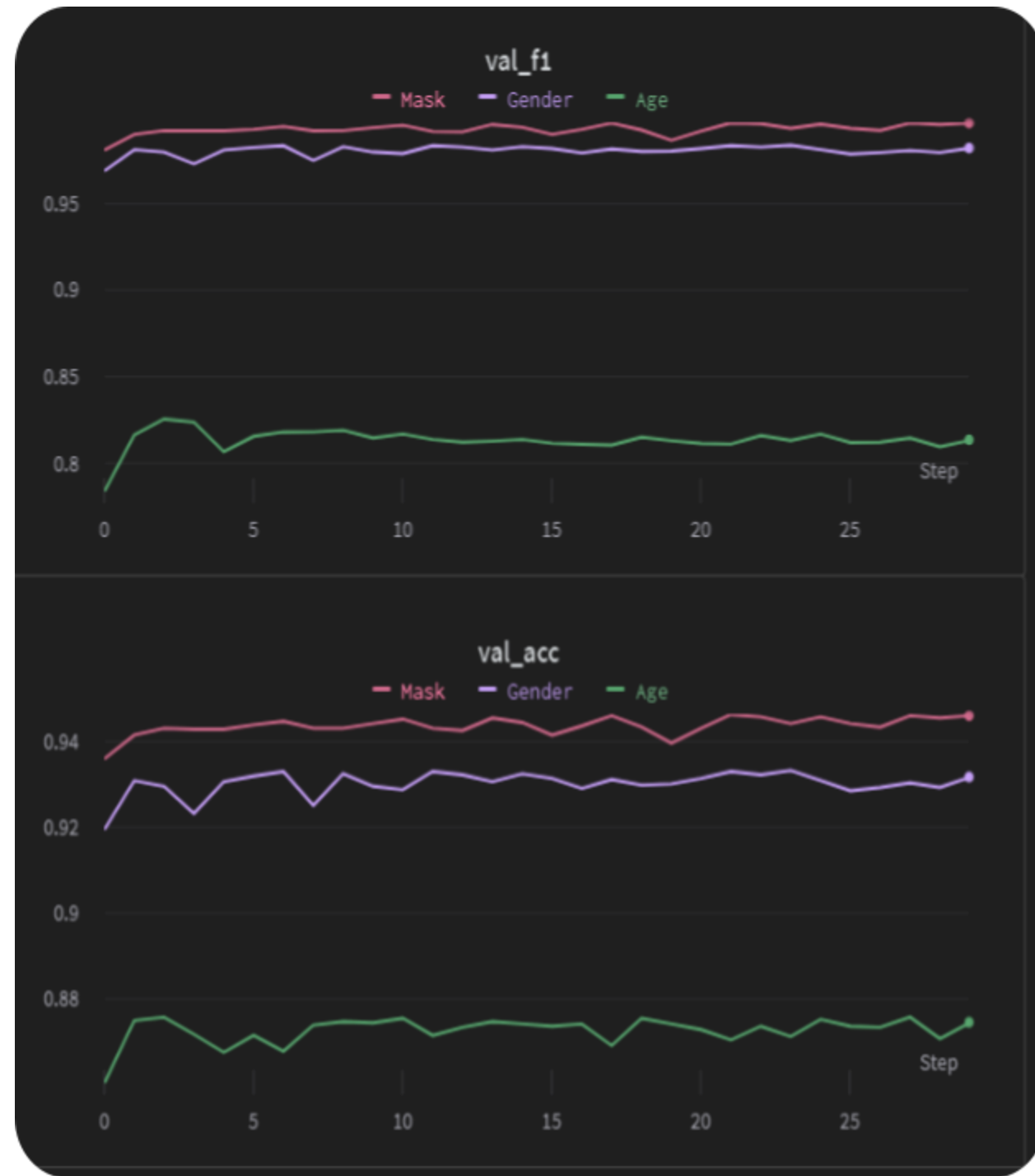


1. 모든 Label이 Imbalance



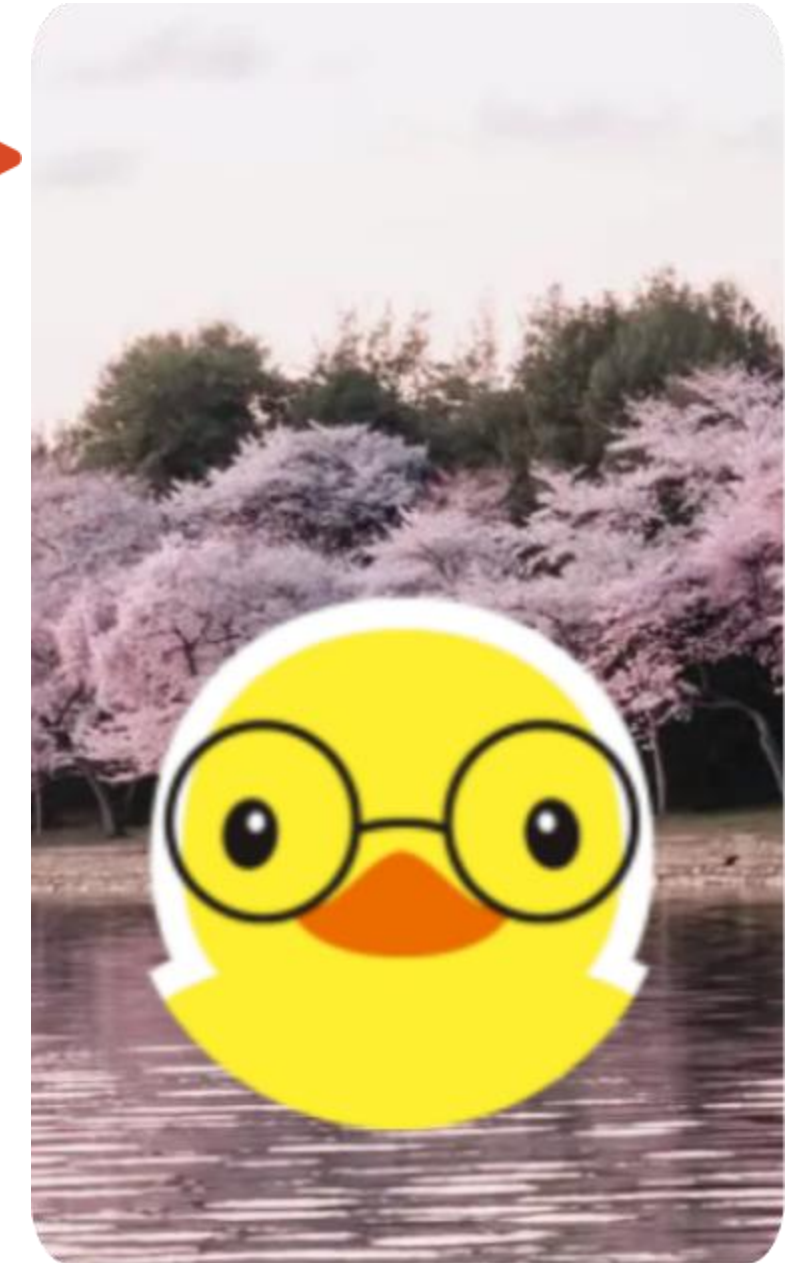
- (1) 이를 해소할 전처리 & 증강 필요
- (2) Train / Val Set 균형있게 구성 필요

5. 프로젝트 - EDA



3. 이미지의 뒷배경, 인물 중앙에 위치

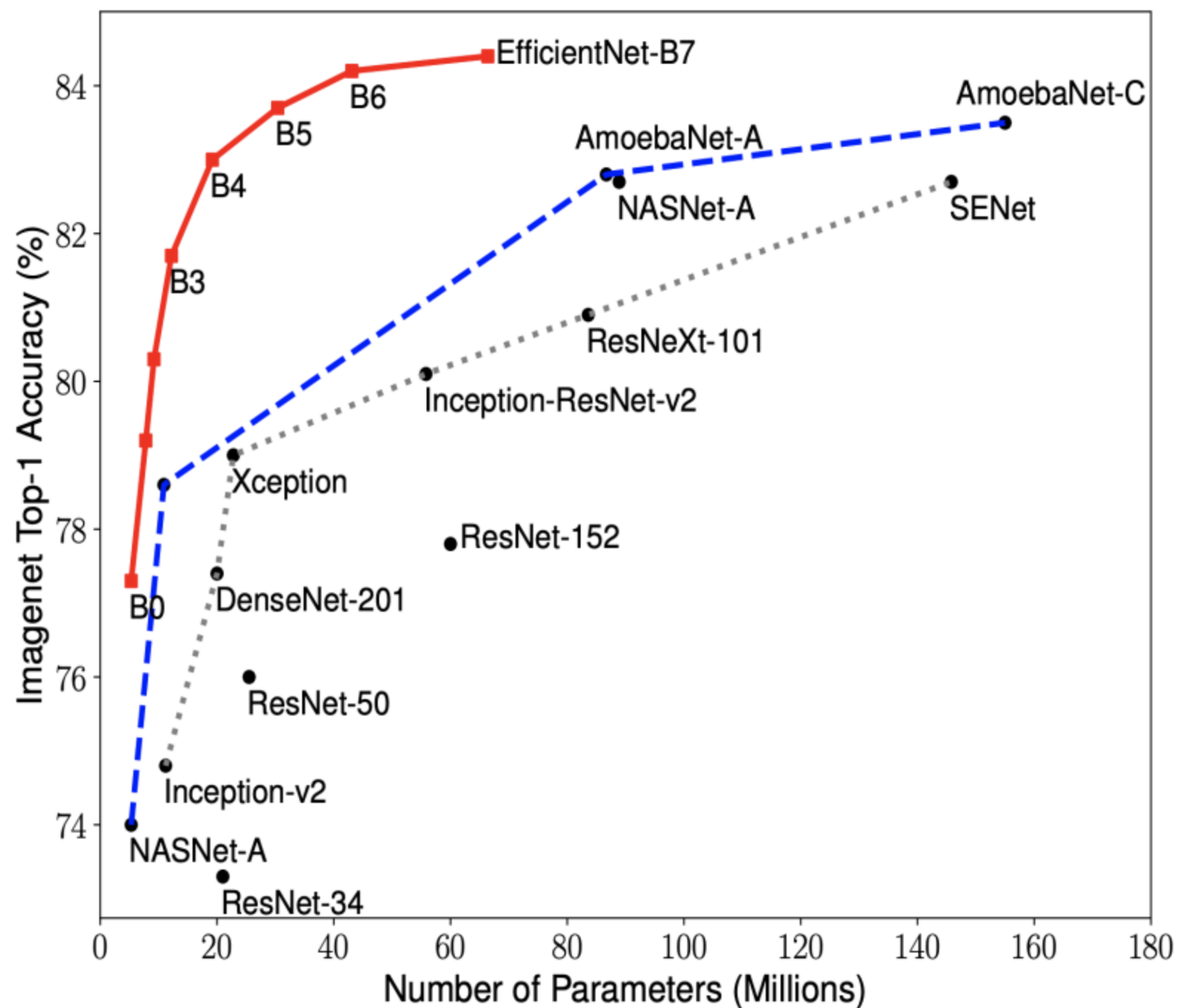
(1) Background는 noise가 될 수 있다



2. 각 Label 예측하는 Single Model

(1) Mask, Gender에 비해서 Age를 못 맞춤
(2) 해결해야할 문제는 'Age'!!

5. 프로젝트 - Modeling



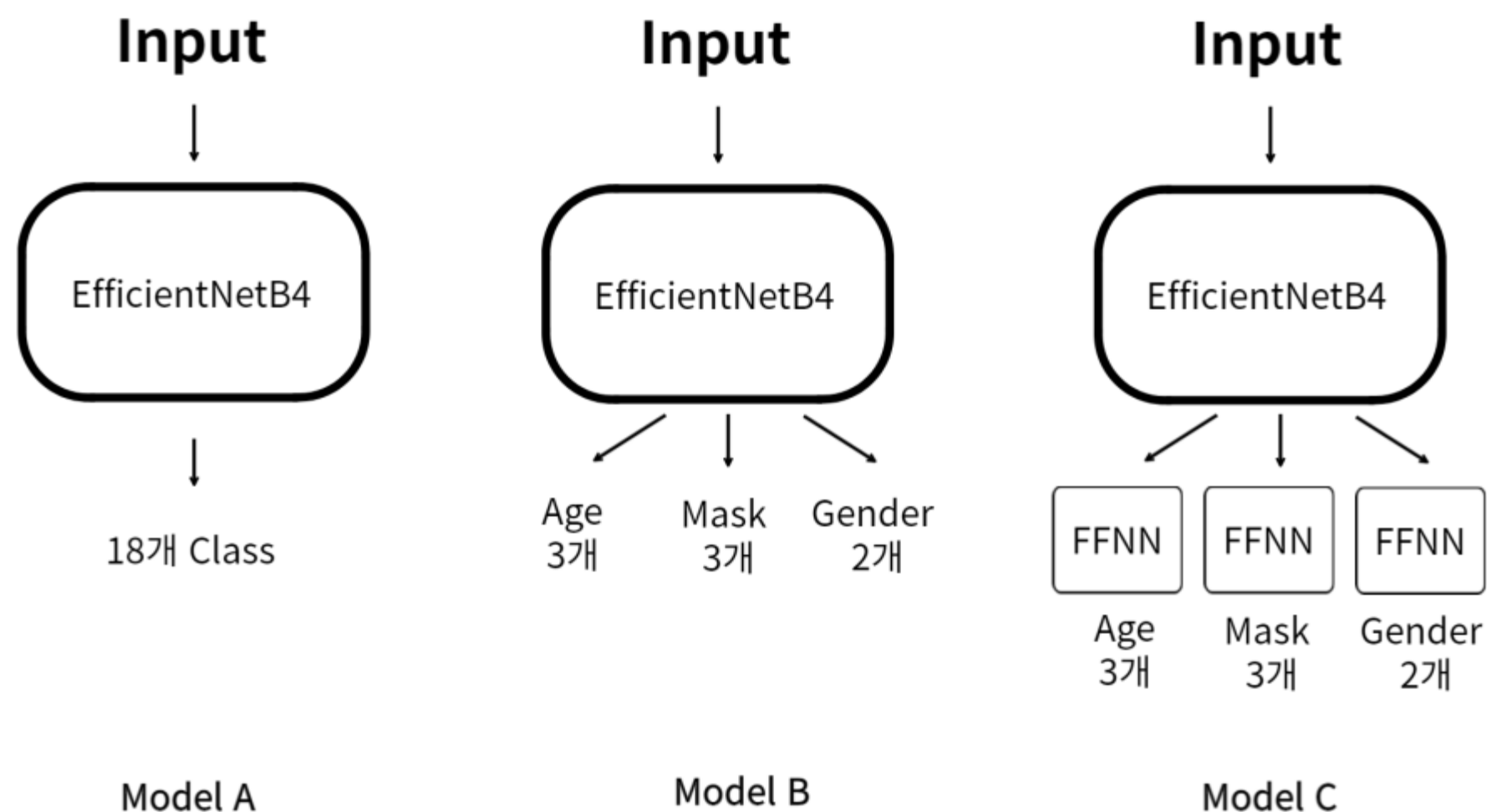
Model : EfficientnetB4 선정

선택이유 :

- Transformer 계열 경우 많은 데이터 필요,
Pretrained 데이터와 우리 데이터 연관성 떨어짐
- 따라서 적은 데이터에도 괜찮은 성능인 CNN 계열 선택
- Densenet, Resnet, Efficientnet를 비교하여, 준수
한 성능을 보인 Efficientnet을 선정 - [Notion 참조](#)
- 왼쪽 그래프에서 확인, B4까지 성능 차이 존재, B4 이
후 부터 큰차이 x, 따라서 효율성을 위해 B4 선택

5. 프로젝트 - Modeling

Modeling은 3가지 방식으로 진행



- **Model A** : One Way Model - Single Label 예측

- **Model B** : Three Way Model - shallow Branch

- 각 Branch를 통해 Multi Label 예측

- **Model C** : Three Way Model - deep Branch

- 깊은 Branch 사용 - 조금더 각 Label에 집중해보자!!

- **Multi Branch를 활용한 이유 :**

1) 각 Label이 서로 관계가 없음에도 영향을 줄 수 있다고 생각

- ex) Mask와 Age는 관계가 없지 않을까?

2) 각 Loss를 따로 계산하기 때문에 상대적으로 못맞추는 Loss를 더 크게 반영할 수 있지 않을까?

5. 프로젝트 - 데이터 전처리

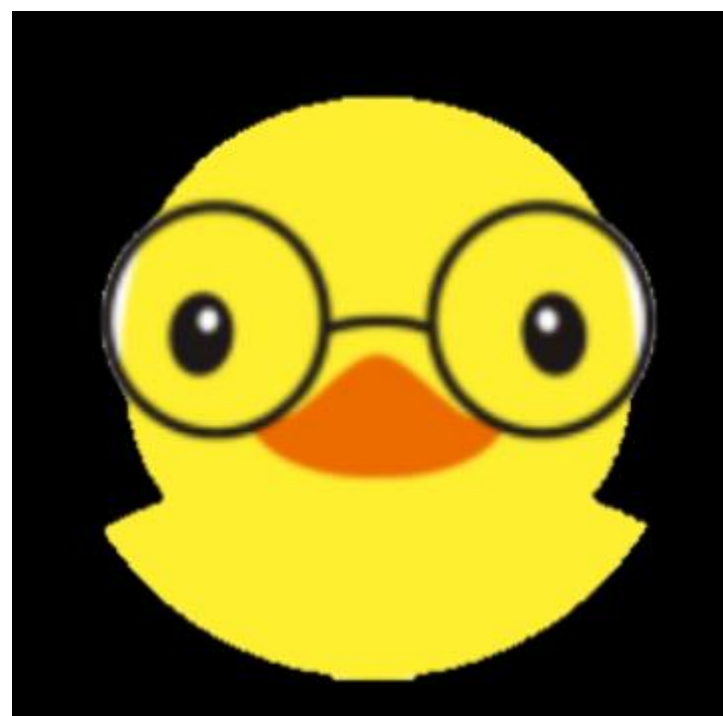
1. rembg & deepface



원본



rembg - 배경 제거



deepface -
얼굴 탐지하여 확대

1. Rembg 라이브러리를
활용해서 배경을 제거

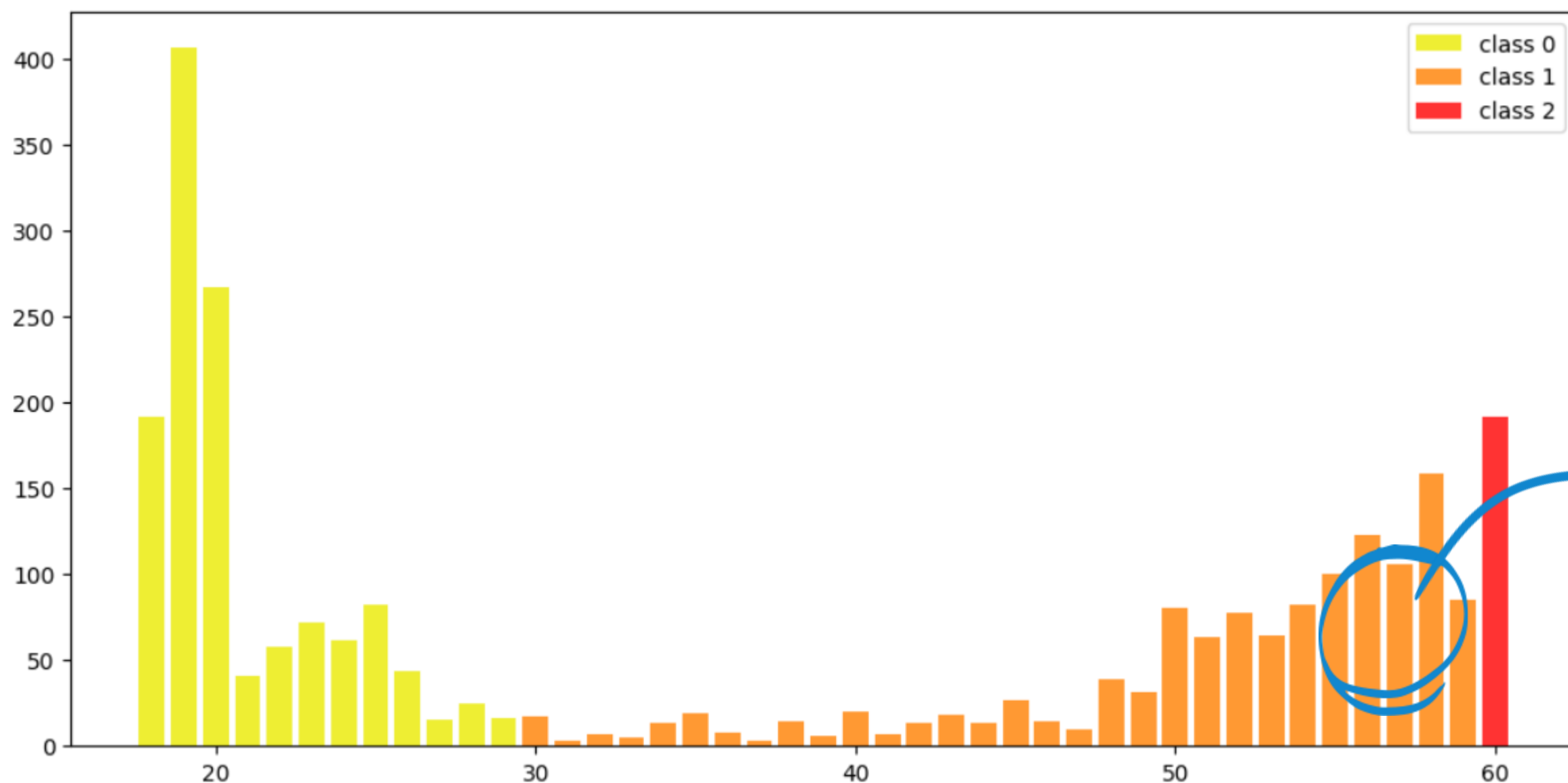
2. DeepFace를 활용해서
얼굴탐지 및 크롭



Noise를 제거하는 것이 목적

5. 프로젝트 - 데이터 전처리

2. 50대 후반(57,58,59) 데이터 remove



50대 후반 데이터 제거 이유 :

1. 60대 경계에 위치한 값들이 60대 범주보다 분포가 높기 때문에 분류를 방해한다
2. 경계값을 60대로 바꾸는 것은 오히려 잘못 Labeling된 데이터를 늘린다고 판단

5. 프로젝트 - 데이터 전처리

3. Train / Validation Set 구성

Baseline Code

- (1) 'Random split' : 모든 데이터를 랜덤적으로 Split - **train/val에 동일한 정보 존재 문제**
- (2) 'SplitByProfile' : 사람 기준으로 데이터를 Split - 위 문제 해결, **But 데이터가 불균형하게 Split**



ClassbalanceSplitByProfile

: 사람을 기준으로 하면서 class 별 stratify를 유지하여 8 : 2로 Split

```
train ratio [(0, 2195), (1, 1640), (2, 330)]  
val ratio [(0, 550), (1, 410), (2, 85), (3,
```

5. 프로젝트 - Augmentation

1. Mixup



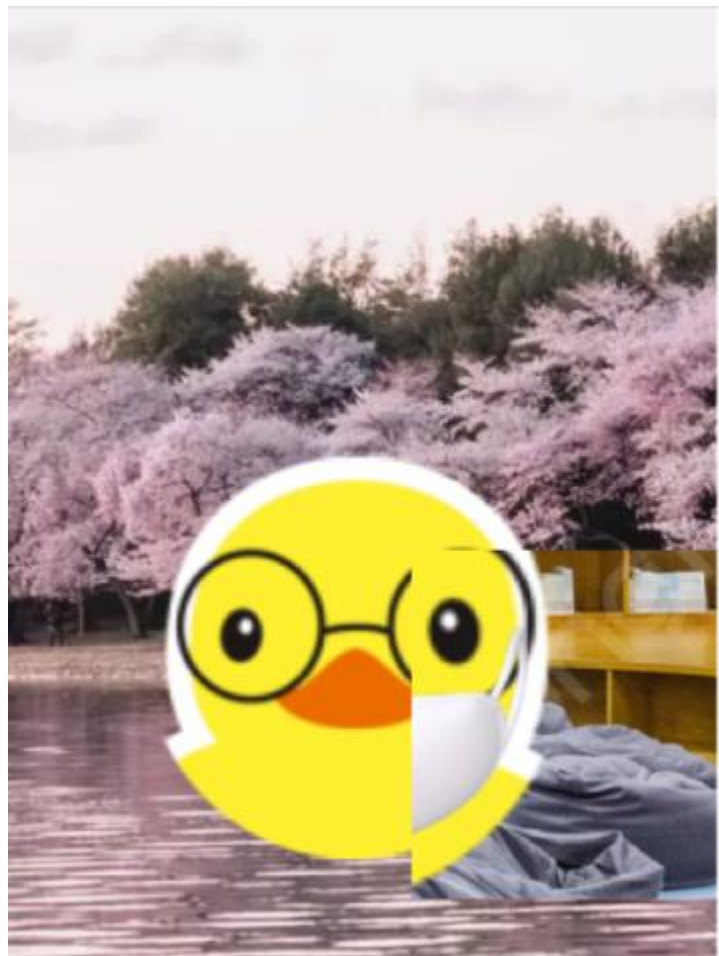
Mixup

데이터 불균형을 어느정도 해소하기 위해서
60대 데이터만 Mixup하여
추가 데이터 확보

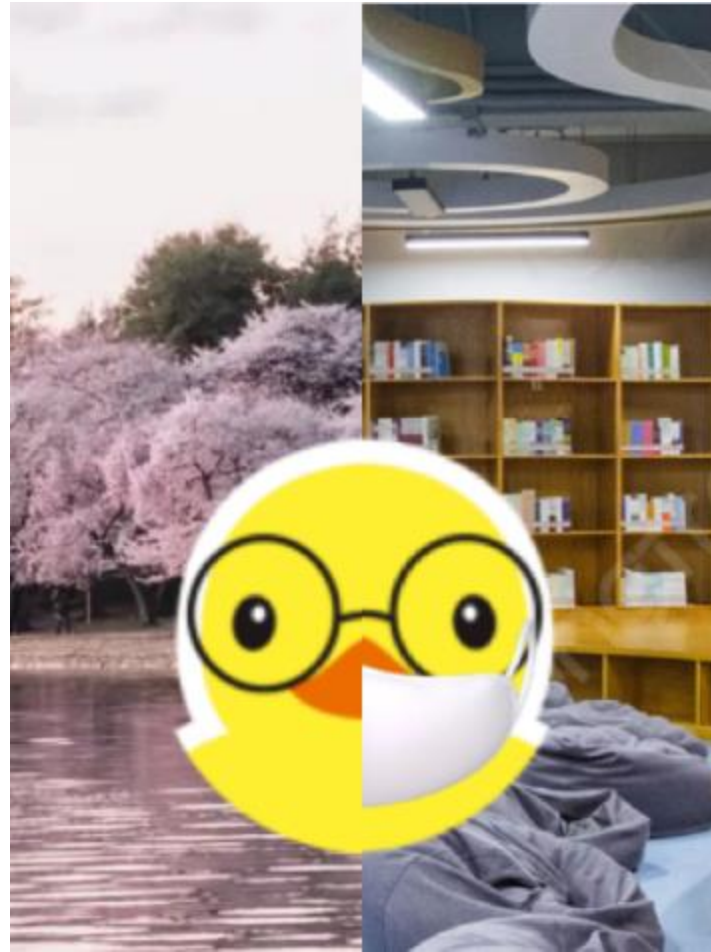
하지만 다른 Augmentation과 활용하면
과적합 되는 경우가 발생하여 Mixup 사용시
다른 Augmentation 사용 x

5. 프로젝트 - Augmentation

2. Cutmix



Basic Cutmix



Custom Cutmix

모델의 과적합을 줄이기 위해서
Cutmix 사용

그러나 Random 적인 Cutmix는 오히려
성능을 하락시킴 (**Basic Cutmix**)
= Label 정보를 손실



따라서 Label 정보를 유지하면서 5:5 비율로
Cutmix 실행 (**Custom Cutmix**)

5. 프로젝트 - Ensemble

2가지 방법으로 접근

1. Leader Board 기준으로 상위 3개의 모델을 Hard Voting

- Leader Board에 과적합 될 수 있다고 판단 (Ensemble 1)

2. Validation 기준으로 Robust한 분포를 맞추는 3개의 모델 Hard Voting

- 조금더 Robust한 모델을 구성할 수 있지 않을까?(Ensemble 2)



최종 모델로 앙상블한 2개 모델 제출

5. 프로젝트 - 결과

Hyper Parameter는 고정 (lr, Optimizer, Scheduler, Loss)

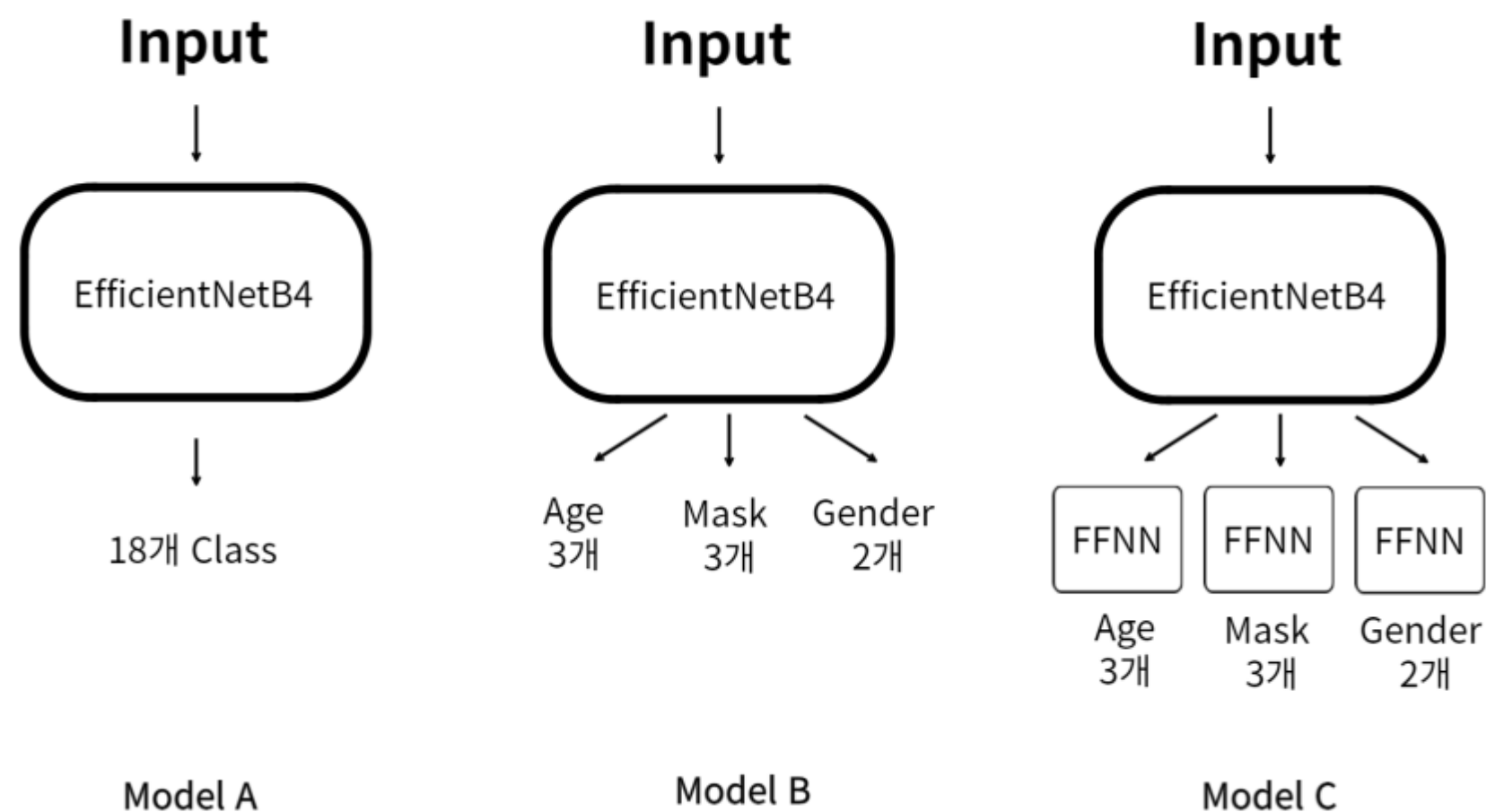
- LR : 0.0001
- Optimizer : Adam
- Scheduler : ReduceLROnPlateau
- Loss : Focal Loss

경험적 실험을 통해서 성능이 비슷했기 때문에
다른 요소를 비교하기 위해서 고정하여 모델링 진행

- 실험 내용은 Notion 참고

5. 프로젝트 - 결과

1. Modeling



(val 기준)

Method	Accuracy	F1 Score
Model A	0.8929	0.8953
Model B	0.8942	0.8842
Model C	0.9058	0.9254

(Model A와 B는 성능 차이가 크지 않음
Model C의 경우 Val 기준으로 가장 좋음)

그러나 Test 기준으로는 가장 낮음 - 과적합의 문제 발생

(validation Score 기준)

Method	Accuracy	F1 Score
Model A + C	0.8854	0.8884

Model A의 Feature Map을 가져와 Freeze하고
Model C의 Branch만 학습 -> 가장 좋은 Test 기록

5. 프로젝트 - 결과

2. 데이터 전처리

1. rembg & deepface

결과:

Method	Accuracy	F1 Score
CenterCrop + Resize	0.8704	0.8644
rembg	0.867	0.8628
rembg + Deepface	0.8948	0.918

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

2. 50대 후반(57,58,59) 데이터 remove

결과 : (validation Score 기준)

Method	Accuracy	F1 Score
Basic Data set	0.8794	0.796
57~59 Remove	0.862	0.8457

➡F1 Score 가 Validation, Test 에서 모두 상승

5. 프로젝트 - 결과

3. Augmentation

1. Mixup

결과: (validation Score 기준)

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

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

2. Cutmix

결과 : (validation Score 기준)

Method	Accuracy	F1 Score
Basic Augmentation	0.8704	0.8644
Basic Cutmix	0.8906	0.8999
Custom Cutmix	0.8929	0.8953

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

5. 프로젝트 - 결과

4. Ensemble

Ensemble 1

선택한 모델

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

Ensemble 2

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	0.8261	0.7868


Ensemble 2 가 1 보다 높은 Test Score 를 보임

조금 더 Robust 한 모델이 만들어지지 않았나 추론

5. 프로젝트 - 결과

5. 최종결과

Public Score

			<u>F1</u>	<u>ACC</u>
1 (-)	CV_14조		0.7868	82.6190

Private Score

			<u>F1</u>	<u>ACC</u>
1 (-)	CV_14조		0.7722	81.4762

Public 보다 Private에서 F1, Acc가 하락했지만
Public Score, Private Score 모두 1위 차지

6. 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
- Git과 같은 협업툴을 적극적으로 활용하지 못해서 아쉬웠다
- 실험 공유에 Convention을 정하지 못해서 아쉬웠다
- Python Code 버전 관리를 하지 못해서 동일한 조건을 맞추기 힘들었고 새로 구현한 기능의 호환성을 맞추기 힘들었다

6. 회고 및 개선방안

개선 방안

- Validation Set을 구성할 때 Robust한 분포의 Set을 구성하지 못해서 모델에 대한 객관적인 평가를 할 수 없었던 것 같다
- On-line으로 데이터를 mix-up 하는 것을 시도해 보지 못했다
- K-fold의 가설을 경험적으로 검증해 보지 못했다