

# Image Classification Wrap-up Report

CV 11 머신러닝

서지훈, 이준하, 이지유, 이채원, 최지욱

---

## 목 차

1. 프로젝트 개요 .....	1
1.1. 프로젝트 소개 .....	1
1.2. 데이터 개요 .....	1
1.3. 환경 .....	2
2. 프로젝트 팀 구성 및 역할 .....	2
3. 프로젝트 수행 절차 및 방법 .....	3
3.1. 주차별 팀 목표 .....	3
4. 프로젝트 수행 결과 .....	4
4.1. 데이터 EDA .....	4
4.2. 데이터 전처리 & 증강 .....	4
4.3. 베이스라인 선정 및 가설 수립 .....	4
4.4. 모델 개선 과정 .....	5
4.4.1. Multi-output & Age regression loss .....	5
4.4.2. Mislabel .....	6
4.4.3. Segmentation .....	6
4.4.4. Ensemble .....	6
4.5. 최종 결과 .....	7
5. 자체 평가 의견 .....	7
6. 개인 회고 .....	8
6.1. 서지훈 .....	8
6.2. 이준하 .....	9
6.3. 이지유 .....	10
6.4. 이채원 .....	11
6.5. 최지욱 .....	12

## 1. 프로젝트 개요

### 1.1. 프로젝트 소개

COVID-19의 치사율은 낮은 편에 속지만, 강력한 전염성으로 전 세계 사람들은 경제적, 생산적인 활동에 많은 제약을 가지게 되었다.

감염 확산 방지를 위해서는 모든 사람이 마스크로 코와 입을 가려서 감염자로부터의 전파 경로를 원천 차단해야 한다. 이를 위해 공공 장소에 있는 사람들은 반드시 마스크를 착용해야 할 필요가 있으며, 무엇보다도 코와 입을 완전히 가릴 수 있도록 올바르게 착용하는 것이 중요하다. 하지만 넓은 장소에서 모든 사람들의 올바른 마스크 착용 상태를 검사하기 위해서는 많은 인적자원이 필요할 것이다.

따라서, 우리는 사람의 얼굴 이미지만으로 마스크를 올바르게 착용한 것이 맞는지 가려낼 수 있는 시스템이 필요하다.

### 1.2. 데이터 개요

전체 데이터셋은 4,500명의 아시아인 남녀로 구성되어 있고 나이는 20대부터 70대까지 다양하게 분포한다. 사람 당 마스크 착용 5장, 이상하게 착용(코스크, 턱스크) 1장, 미착용 1장의 구성으로 (384, 512) 크기의 이미지 7장이 존재한다. 각 이미지는 하나의 클래스에 대응하며 클래스는 하단의 표와 같이 마스크 착용여부, 성별, 나이를 기준으로 총 18개가 존재한다. 모델은 입력값으로 이미지 한 장을 받아 결과값으로 해당 이미지에 대응하는 클래스를 숫자(0 ~ 17) 형태로 출력하여야 한다.

학습 데이터와 평가 데이터는 전체 데이터셋을 임의로 섞어서 6:4 비율로 분할하여 생성한다. public 테스트셋, private 테스트셋은 평가 데이터셋을 다시 1:1 비율로 임의 분할하여 생성한다. 대회 진행 중 리더보드 점수는 public 테스트셋으로 계산이 되고, 최종 순위는 private 테스트셋을 통해 산출한 점수로 확정된다.

Class	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

---

### 1.3. 환경

팀원은 5명으로, 인당 V100 서버 하나씩을 할당받았다.

협업 툴로는 Notion, Github, Wandb를 사용했고, Zoom, Slack을 사용하였다.

## 2. 프로젝트 팀 구성 및 역할

서지훈: 데이터 검토 및 잘못 라벨링된 데이터 수정

이준하: EDA 및 증강을 기반으로 한 base-line code 구현 및 SOTA model에 대한 논문 검토, face-toolbox-keras를 활용한 배경 제거, stratified kfold 및 Soft Voting을 구현하였고, 30여가지 조합에 대한 실험을 진행하였다.

이지유: 6가지 Model 클래스(ResNet 18, ResNet 34, EfficientNet B1, EfficientNet B2, ViT Tiny, ViT Small) 및 Model Wrapper 클래스, 모델 학습 과정, 모델 성능 지표 및 실험 정보 로깅 기능(wandb)을 구현하였다. 또한, Age Regression 및 Multi-Output 실험을 진행하고 Validation Data Leakage 이슈를 해결하였다.

이채원: grad-CAM으로 모델 visualization, DeeplabV3 segmentation 데이터를 생성하고, label smoothing 및 segmentation 실험을 진행하였다.

최지욱: Age Regression 및 Multi Output 관련 코드 구현, 모델 Metric 및 wandb log 코드 구현, Age Regression 및 Multi-Output 실험 진행, Validation Data Leakage 에러 분석 및 수정. Top-N 결과를 Hard Voting (ensemble)을 하였다.

### 3. 프로젝트 수행 절차 및 방법

#### 3.1. 주차별 팀 목표

Tasks		week 1								week 2							
		10	11	12	13	14	15	16		10	11	12	13	14	15	16	
preparation	Study																
	Make plan																
Environment	Collaborative Environments																
	Make python script																
Experiment	EDA																
	Augmentation																
	Hyper parameter tuning																
	Change loss, metrics																
	2-class gender classification model																
	Multi class classification																
	Change backbone																
	Add FC layer																
	Grad Cam																
	Segmentation (remove background)																
	Age regression loss																
	Remove miss label																
	Ensemble																
Summary results	Summit																
	Wrap up report																

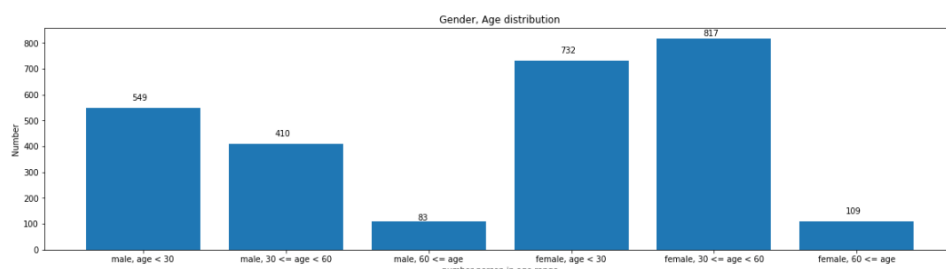
1주차는 강의 내용 및 baseline 코드를 이해하고, 성능 개선 방안을 생각하는 것을 목표로 하였으며, 해당 과정에서 주어진 데이터 분석 및 증강 적용, 선행 연구를 검토하였다. 2주차는 협업 환경을 구성 후, wandb를 활용한 실험 기록과 더불어 분류 체계, loss 및 backbone 모델 변경 등의 실험을 통해 성능 개선을 시도하였으며, Grad cam을 통해 모델의 representation을 분석하여, 추가 실험 방안을 모색하였고, 그 결과로 segmentation 기반의 배경 제거 및 miss label 제거 실험을 수행하였다. 최종적으로, 제출한 결과를 hard voting, stratified 5fold 및 soft voting 등을 구현하여, ensemble을 실험하였다.

## 4. 프로젝트 수행 결과

### 4.1. 데이터 EDA

훈련용 데이터 셋에는 2700명, 총 18900장의 사진이 존재한다.

각 이미지는 이산형 변수 Gender와 연속형 변수 Age를 가지며, 분포는 아래 그림과 같다.



또한, 마스크의 착용이나 성별에서 잘못 라벨링된 데이터가 존재하며, 이미지에서 사용자의 얼굴은 대부분 중앙 부근에 위치함을 확인하였다.

### 4.2. 데이터 전처리 & 증강

60대 이상의 데이터가 매우 적어 클래스 간 불균형이 크므로 데이터 증강을 위하여 albumentation 라이브러리의 RandomBrightnessContrast와 GaussNoise를 사용하였다.

또한, 사용자의 얼굴은 대부분 이미지의 중앙에 존재하므로 Centercrop을 사용하여 모델이 이미지의 필요한 부분인 얼굴에 집중하도록 하였다.

### 4.3. 베이스라인 선정 및 가설 수립

베이스라인은 우수한 성능으로 널리 알려진 모델인 EfficientNet B0로 설정하였다.

앞선 EDA의 결과로부터 데이터셋에 잘못 라벨링된 데이터가 존재한다는 것을 확인할 수 있었다. 따라서, 각 이미지 당 단일 18가지 클래스를 예측하는 모델보다 공통의 Backbone Network에 브랜치를 생성하여 Multi-Output(Mask, Gender, Age) 각각을 예측한 뒤 취합하는 모델이 더 좋은 성능을 보일 것으로 추측하였다.

또한, 데이터셋을 수작업으로 확인하는 과정에서 이미지 파일명에 나이 값이 클래스가 아닌, 연속형 변수로 기록되어 있는 것을 발견하였다. 이를 바탕으로, 실제 나이 값과 예측 나이 값 사이의 거리를 MSE Loss로서 손실 함수에 추가 반영하는 것이 더 좋은 성능으로 이어질 것으로 추측하였다.

## 4.4. 모델 개선 과정

### 4.4.1. Multi-output & Age regression loss

일반적으로 우수한 성능을 가지는 모델이 이 데이터셋에서도 우수한 성능을 보일 것이라고 확신할 수는 없으므로, 다양한 모델을 비교하여 이 데이터 셋에 적합한 모델을 탐색하였다.

모델 6개(ResNet18, ResNet34, EfficientNetB1, EfficientNetB2, ViTTiny\_Patch16\_384, ViTSmall\_Patch16\_384)를 돌려본 결과, EfficientNetB2가 가장 우수한 성능을 보였다.

Model	Validation F1 Score
EfficientNet B0 (baseline)	0.93093
ResNet 18	0.88177
ResNet 34	0.93165
EfficientNet B1	0.94594
<b>EfficientNet B2</b>	<b>0.95842</b>
ViT Tiny (Patch 16, 384)	0.76766
ViT Small (Patch 16, 384)	0.77515

하지만 Validation set으로 검증한 결과와 실제 제출한 테스트셋 간 성능 차이가 현격히 발생한다는 문제점이 있었다. 원인은 Python script 코드상의 오류로, 인스턴스 변수로 선언해야 할 부분을 클래스 변수로 선언하여 검증용 데이터셋이 학습과정에 사용되는 문제였다.

이 후 코드를 수정하고, 아래의 실험을 진행하였다. Age의 level을 분류하는 문제이지만, 실제 주어진 나이는 level이 아닌 실제 연속적인 나이이기 때문에 이를 최대한 분류 모델에 이용하고자 하였다. 3개의 level로 분류하는 경우 30세와 59세가 동일한 level이기 때문에 이에 대한 loss가 반영이 불가능하고, 20세와30세, 20세와 60세 간의 Loss를 동일하게 취급한다는 문제점이 있었다. 이러한 측면을 반영하여 모델에 연속형 변수 Age에 대한 예측을 하도록 구현하고 이에 대한 Loss를 반영하도록 하였다. 아래 실험에서는 Age에 대한 MSE loss에 대한 가중치를 [0, 0.05, 0.10]으로 실험하였다.

두 번째로는, 18개의 class로 분류하는 single-output 모델과 Mask, Gender, Age를 개별적으로 예측하도록 하는 multi-output 모델과 비교하고자 하였다.

EfficientNet B2		
Age Regression Weight	Single-output	Multi-output
0.0	0.7819	0.7940
0.05	0.7992	0.8044
0.10	0.8115	0.8185

여러 실험을 거친 결과, Age에 대한 MSE loss penalty 가중치는 0.1이 가장 높았고, Single-output 보다는 Multi-output이 근소하게 우수한 성능을 보였다.

#### 4.4.2. Mislabel

데이터 EDA에서 언급했던 것처럼, 잘못 라벨링된 데이터가 존재한다. 성별이 잘못 라벨링된 데이터는 9개, 마스크 라벨링이 잘못된 데이터는 2개가 존재했다. 이는 데이터의 오류이므로, 이를 수정하여 성능 개선을 시도하였다.

#### 4.4.3. Segmentation

- 필요성: Base model test 결과, age와 gender label에 대한 정확도가 mask label 보다 떨어지는 것을 raw 하게 확인하였다. 또한 validation accuracy와 test accuracy이 약 0.25 이상으로 차이가 심해 모델 학습에 이상이 있다고 판단하였다.
- Visualization with grad-CAM: 시각화 결과에서 age label이 'middle' 또는 'old'인 데이터에서 배경이나 옷 무늬의 feature에 모델이 집중하고 있으며, 모델이 얼굴이 아니라 배경 무늬와 같은 특정 요소를 이용하여 성별이나 나이를 예측하는 trivial solution을 찾은 것을 확인하였다. 이에 따라 배경 정보를 제거하면 모델의 성능이 개선되어 특정 성별이나 나이에 대한 편향성이 제거될 것으로 예측하고 segmentation을 진행했다.

- Segmentation dataset 생성

Deeplab v3을 통해 train set과 test set의 background를 제거한 새로운 dataset을 생성했다.

- Segmentation data를 이용해 학습한 모델의 성능 평가

동일한 test 입력에 대해 기존 모델, segmentation을 통해 배경을 제거한 이미지로 train한 모델의 마지막 layer를 visualization 하여 비교하였다. Segmentation trainset으로 train한 모델에서는 배경, 옷에서 규칙성을 찾는 기존 모델의 문제가 완화되었고, 얼굴 영역에 집중하고 있음을 확인할 수 있었다.

	Seg O	Seg X
Age 0.5	0.79153	0.8090
Age 1	0.7703	0.7829

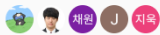

그러나 모델의 best validation f1 score를 비교했을 때 segmentation을 하기 전이 성능이 더 좋았다. 이는 가설과 달리, test set에서도 background가 train set과 어느 정도 유사성이 있기 때문인 것으로 추측하고 있다.

#### 4.4.4. Ensemble

성능이 높은 상위 5개 모델들에 대한 hard voting을 진행하였다. 이는 Majority voting으로 구현되며, 동일한 수가 voting된 경우 더 높은 성능을 갖는 모델의 결과를 인용하였다. 또한 Stratified 5fold를 적용하여 생성된 subset들을 학습한 모델들에 대한 soft voting을 시도하였다.



## 4.5. 최종 결과

순위	팀 이름	팀 멤버	f1 ↕	accuracy ↕	제출 횟수	최종 제출
14 (-)	CV_11조		0.7308	79.1270	56	1d
순위	팀 이름	팀 멤버	f1 ↕	accuracy ↕	제출 횟수	최종 제출
11 (3 ▲)	CV_11조		0.7288	78.3651	56	6d

위와 아래는 각각 public dataset, 전체 dataset을 기준으로 평가한 순위이다.

## 5. 자체 평가 의견

### 잘했던 점

1. Python script 기반의 코드를 작성하여 프로젝트를 진행하였다.
2. Multi output 문제로의 치환, 잘못된 라벨링 데이터 수정, Segmentation, 모델 변경 등 다양한 성능 개선 방법을 시도해보았다.

### 시도했으나 잘 되지 않았던 것들

1. Segmentation을 시도하여 모델 성능 개선을 시도했으나 실패하였다.

### 아쉬웠던 점

1. 처음부터 Python script를 이용하여 서로 간의 코드를 통일하여 진행했다라면 더 넉넉한 시간을 가지고 대회를 진행할 수 있었을 것 같다.
2. Python에 익숙하지 못해 학습 과정에서 발생한 실수로 학습 과정에서 많은 시간을 허비했다.
3. 실험 결과를 체계적으로 정리하지 않았다.
4. Train dataset을 잘못 구성하여 학습과정에서 Validation data의 leakage가 발생하였고 이에 따라 올바르게 모델을 평가하지 못하는 문제점이 생겼다. 결과적으로 해결하긴 하였으나, 대회 후반부에 해결이 되어 더 많은 실험을 진행하지 못했다.
5. Resize, optimizer, loss, learning rate scheduler 등을 다양하게 사용해보지 못했다.
6. TTA, Augmentation 등을 활용해보지 못했다.

### 프로젝트를 통해 배운 점 또는 시사

1. Python script 기반의 프로젝트 진행 방법을 배웠다.
2. WandB로 실험 결과 분석 및 협업하는 방법을 배웠다.

---

## 6. 개인 회고

### 6.1. 서지훈

저는 기존에 졸업과제를 인공지능 관련으로 하여 프로젝트를 진행해 본적이 있습니다. 하지만 그 프로젝트는 사실상 혼자 진행했고, 졸업과제 특성상 기간이 넉넉했기에 모델, loss 함수, optimizer, 전처리, 증강 등에서 매우 많은 변화를 시도하면서 성능 향상을 시도했습니다. 하지만 이번 프로젝트는 기간이 부족하여 그런 방법이 불가능했습니다. 때문에 일단 실행해보고 결과를 보자는 기존의 제 실험 방식은 불가능했고, 기존의 경험과는 전혀 다른 방식으로 프로젝트를 진행하게 되었습니다.

시간이 부족하기에 실험을 효율적으로 진행해야 했습니다. 때문에 성능을 높일 수 있을 만한 변경점을 찾아야 했고, 이를 위해 데이터 분석의 필요성을 크게 느꼈습니다. EDA에서 단순히 적절한 전처리와 증강 방법을 도출하는 것이 아닌, 배경이 많은 이미지의 특성을 바탕으로 필요한 사람에게 집중하기 위해 segmentation 기법을 사용하면 성능이 상승할 것이라는 가설과, 성능의 문제는 대부분 분류하기 어렵다고 판단되는 class인 age에서 발생할 것이므로 이를 해결하는 기법을 사용하면 성능이 상승할 것이라는 가설 등을 수립하는 과정을 보며 제가 기존에 얼마나 단순한 방식으로 성능 개선을 시도했는지 깨달았고, 많은 것을 느끼고 배웠습니다.

---

## 6.2. 이준하

본 대회는 사람의 얼굴 이미지가 주어지면, 해당 사람의 성별 및 연령대, 그리고 마스크를 썼는지 여부를 분류하는 문제였습니다. 이와 관련하여 1주차에는 관련 강의 내용 및 baseline code를 이해하고, 전처리, 증강, 파라미터 변경 등의 실험과 선행 연구를 찾아보는 등 성능 개선 방안을 수립하였으며, 2주차에는 협업 환경 구성 및 loss, backbone과 관련된 실험과, 결과 분석을 통한 추가 실험 방안 모색 및 배경 제거, 분류 요소 분리, ensemble 등 생각한 대부분의 방법을 실험을 통해 입증했습니다.

해당 과정들을 통해 크게 두가지에 대해 깊은 고찰을 할 수 있었습니다. 하나는 협업을 고려한 실험 구성에 대한 것이고, 다른 하나는 이미지 분류와 관련된 여러 요소에 대한 것입니다.

1주차에 각자 진행하던 실험을 2주차에 통합하는 과정에 많은 시간이 소요되었습니다. 조원들 모두 같은 강의를 들었음에도, 코드를 작성하는 방식 및 구조적인 부분이 서로 상이했고, 결국 이를 통합하는 과정에서 통합 자체에 많은 노력을 기울이다 보니, 문제의 본질에 소홀해지고, 당연히 검증해야 했던 여러 요소들을 놓쳤던 것 같습니다. 특히 서브셋 분리 과정에 오류가 없는지 각 서브셋의 길이를 측정하는 단순한 검증조차 실험이 시작된 지 한참 후에 시도하였다는 것이 아쉽습니다. 추후 진행되는 실험에서는 조원 간 실험 환경을 모두 동일하게 구성한 뒤 해당 환경 자체에 대한 검증 및 기록 체계를 갖춘 후, 개별 실험을 진행해야 할 것 같습니다.

또, 이미지라는 데이터의 특성에 대해 고민해 볼 수 있었고, 특히 이미지 분류를 위해 시도할 수 있는 실험 요소들에 대해 많은 고찰을 할 수 있었습니다. 기존 주어진 18개의 클래스 분류 문제를 성, 나이, 마스크 착용 여부의 멀티클래스 분류 문제로 치환하여 나이 부분의 예측력이 낮다는 것을 발견할 수 있었고, 나이를 회귀 문제로 접근할 수 있었습니다. 뿐만 아니라, gradcam으로 동일 배경의 여러 사진들에 대해 model의 representation을 확인하여, 배경이 갖는 의미를 파악할 수 있었고, 이에 모델이 배경을 통해 잘못 학습하고 있다고 생각하여 배경을 제거하는 등의 시도를 할 수 있었습니다.

상대적으로 간단해 보이는 분류 문제에서도 고민할 수 있는 요소들이 많다는 것을 느꼈고, 1주차와 2주차에 상반된 방식으로 실험을 진행하여, 협업의 가치를 느낄 수 있었으며, 게시판 및 마스터클래스를 통해 지식 공유의 중요성에 대해서도 다시금 확인할 수 있었습니다.

---

## 6.3. 이지유

### 학습 목표

본 챌린지의 개인 학습 목표는 인공지능 모델 개발 사이클을 완주하고 베이스라인 코드를 온전히 이해하는 것으로 설정하였습니다. 인공지능 모델 개발에 처음 도전하는 만큼 무리한 학습 일정을 계획하기보다 이후 챌린지에서 잘 활용할 수 있도록 기본기를 충실하게 쌓고자 하였습니다.

### 실험 과정 및 결과

우선, timm 라이브러리를 활용하여 6가지 Model 클래스(ResNet 18, ResNet 34, EfficientNet B1, EfficientNet B2, ViT Tiny, ViT Small) 및 Model Wrapper 클래스(Multi-Class Single-Output)와 모델 학습 과정을 파이썬 스크립트로 구현하였습니다. 또한, 모델 학습에 사용된 하이퍼파라미터와 모델 평가 결과를 기록하고자 wandb 라이브러리를 사용하여 실험 정보 로깅 기능을 구현하였습니다.

이후, 해당 스크립트를 바탕으로 모델 학습 및 평가를 진행하면서 모델의 예측 정확도를 개선하기 위하여 나이값에 대한 MSE Loss를 도입하고 모델의 구조를 Multi-Class Multi-Output으로 변경하였습니다. 또한, 검증 데이터셋과 테스트 데이터셋에 따른 모델 성능 차이 이슈의 원인을 파악하고자 스크립트 디버깅을 수행하였으며 그 결과 검증 데이터셋이 학습 과정에 사용되는 데이터 누출 현상을 발견하여 데이터셋 클래스 내 변수의 스코프(scope)를 수정함으로써 해결하였습니다.

### 느낀 점

우선, 모델 학습 과정에서 실험 관리 체계를 제대로 구축하지 않은 채 장시간 서로 다른 조건의 실험을 진행하였더니 실험 결과 분석 및 인사이트 도출 시 어려움을 겪었습니다. 따라서, 2차 챌린지에서는 실험을 직관적이고 체계적인 방식으로 관리할 수 있는 시스템을 도입하고자 합니다. 특히, 실험 결과를 자동으로 로깅해주는 툴은 많았지만 대부분 필요에 맞게 레이아웃 및 구성 요소를 추가/삭제/수정할 수 있는 자유도가 낮았고 정형화된 실험 로그 형식 또한 존재하지 않아서 해당 부분들을 보완할 수 있는 방안을 모색할 계획입니다.

다음으로, 인공지능 모델 개발 사이클을 완주함으로써 실험 결과로부터 모델에 대한 인사이트를 얻고 새로운 실험 방향을 제시하는 역량의 중요성을 새롭게 인지하게 되었습니다. 따라서, 2차 챌린지에서는 개발 역량보다 문제 해결 능력 배양에 집중하고자 합니다. 특히, Level 1에서 학습한 인공지능 이론을 주어진 문제 상황에 알맞게 적절히 적용하여 모델 개선 방안을 도출하는 연습을 꾸준히 실천할 계획입니다.

---

## 6.4. 이채원

### 학습 목표 달성을 위해 한 일

- 경연 대회가 처음이다 보니 강의를 우선적으로 듣고 공부한 뒤, 팀원들의 도움을 받아 프로젝트에 대한 감을 익혔습니다. 또한 리더보드 순위에만 집착하지 않고, 최대한 많은 시도를 하며 경험을 쌓는 데에 집중하였습니다.

- Week 01 :

팀에서 선택한 base Model을 기준으로, 다양한 Augmentation을 적용해 보는 간단한 실험을 진행했으며, 18개의 class를 갖는 큰 문제를 작은 3가지 문제로 치환해 multi class classification을 수행해 보았고, 라벨링이 잘못된 데이터로 인한 노이즈의 영향을 줄이기 위해 label smoothing loss를 사용해 보았습니다. 이는 실제 성능 향상으로 이어져 성취감을 느낄 수 있었습니다.

- Week 02 :

멘토링을 통해 단순한 parameter tuning 실험이 아니라, 성능이 나쁜 이유와 기존 모델의 문제를 파악한 뒤 논리적으로 가설을 세워서 실험해보는 방향으로 계획을 수정했습니다. 멘토링을 통해 경험에서 나오는 통찰력의 힘을 실감하게 되었습니다.

validation 성능이 높게 나오는 데에 비해 test 성능은 낮게 나온다는 문제점으로 부터 test set의 분포가 train set과 다를 것이라고 추측하였고, 모델이 잘못된 방법으로 학습하고 있을 수 있다는 가설을 세웠습니다. 이를 검증하기 위해 grad-CAM으로 visualization을 시도했습니다. 이 과정에서 모델의 구조와 파라미터, layer에 대해 깊게 공부했고, 실제로 모델이 사람의 얼굴보다는 배경과 옷 무늬를 보고 나이와 성별을 예측하는 방향으로 잘못 학습되었음을 눈으로 확인할 수 있었습니다.

Segmentation을 통해 배경 픽셀을 제거하기 위해 적합한 모델을 찾기 위해 노력했습니다. 모델을 비교해 보고 task에 가장 적절하다고 여겨지는 Deeplab v3을 이용해 배경 픽셀을 제거한 새로운 train set을 생성하였습니다. 이 과정에서 masked 이미지를 얻기 위해 직접 dataset 코드를 작성하면서, dataset의 구조에 대해 더 깊이 이해할 수 있었습니다.

### 한계점 및 느낀점, 깨달은 점

- 다른 조에서 선정한 segmentation 모델을 보면서 주어진 task에 가장 적합하고 빠른 모델을 찾는 일의 중요성을 느꼈습니다. 다음 프로젝트에서는 빠르게 실험을 진행하는 것도 중요하지만 우선적으로 기술 조사를 먼저 진행해야겠다고 생각했습니다.
- 깃허브를 통한 협업에 능숙하지 못하다는 것을 느꼈습니다. 다시 공부해서 다음 프로젝트에서는 어려움 없이 협업을 진행하고 싶습니다.
- 체계적인 공동 실험 환경의 구축의 중요성과, 실험 결과를 미리 미리 꼼꼼하게 기록하는 일의 필요성을 실감하게 되었습니다.

---

## 6.5. 최지욱

### 6.5.1 학습 목표

1주차에 competition에 필요한 강의들을 듣고, Jupyter notebook을 이용하여 실험적으로 모델을 만들어서, 괜찮은 성능을 보이는 결과를 만들었습니다. 하지만, jupyter notebook은 팀원들과 함께 협업이나 체계적인 실험을 하는데 단점들이 많았기 때문에 추후 재사용이 가능하고 option에 따라 실험 환경을 조정할 수 있는 python script를 이용한 학습에 초점을 맞추었습니다. 그리고 주어진 Data를 최대한 이해하려고 노력하였고 그에 맞게 모델과 Loss를 활용하는 것에 초점을 두었습니다.

### 6.5.2 시도한 점

주어진 데이터를 어떠한 구조의 모델로 구성해야 적합한지, 어떤 Loss를 이용하는 것이 올바른지를 중심으로 많은 고민을 해보았습니다. 이런 측면에서 팀에 2가지 아이디어를 제안하였고 실험을 진행하게 되었습니다.

1. Multi-output model: 기존 18개의 클래스로 분류하는 Single output model의 경우 Mask, Age, Gender 중 하나만 잘못 예측하더라도 모든 예측이 잘못된 예측으로 간주한다는 문제점이 있었습니다. 그리고 잘못 라벨링 된 데이터가 있는 경우(e.g., Gender) 올바르게 라벨링된 데이터(e.g., Age, Mask)도 잘못 라벨링된 것으로 간주되는 문제점이 있었습니다. 이런 문제점을 해결하고자 각각의 output에 대해 별개의 Cross entropy loss를 구하고 학습하도록 하는 Multi-task model 실험을 제안하였고, 구현하였습니다.

2. Age MSE Loss: 주어진 Baseline code에서는 Dataset에서 연속형 변수인 Age를 구간별로 3개의 level로 Class를 나누고 학습시에 Age 분류에 대한 Cross entropy loss를 구하도록 되어있었습니다. 하지만, 3개의 level로 분류하는 경우 30세와 59세가 동일한 level이기 때문에 이에 대한 loss가 반영이 불가능하고, 20세와 30세, 20세와 59세 간의 Loss를 동일하게 취급한다는 문제점이 있었습니다. 이런 문제를 해결하기 위해 모델에 연속형 변수 Age에 대한 예측을 하는 보조 task를 수행하는 Branch를 만들고 이에 대한 MSE Loss를 반영하도록 하였습니다.

### 6.5.3 추가적인 역할

팀원들이 모델을 여러 방향으로 이해하고 평가할 수 있도록, 각각의 Loss와 Metric(accuracy, macro-f1)을 구현하였고, 팀wandb project에 기록되도록 하였습니다. 실험 진행 간 Validation f1-score 지나치게 높은 것에 대해 dataset의 leakage를 지속적으로 의심하던 중 팀원과 함께 디버깅을 통해 dataset instance간 데이터를 공유하고 있던 문제점을 해결하였습니다.

### 6.5.4 한계 및 아쉬웠던 점

체계적인 실험을 위해 python script를 활용했지만, 통제해야 할 변수와 조작해야 할 변수를 설정하는 기본적인 실험 설계를 간과하여 정작 실험 결과를 이용하기 어려운 측면이 있었습니다. 추후 프로젝트에서는 알아보고자 하는 것의 목적에 맞게 실험 설계를 제대로 한 뒤 실험을 실시해야 할 필요성을 느꼈습니다.