




마스크 착용 상태 분류

카메라로 촬영한 사람 얼굴 이미지의 마스크 착용 여부를 판단하는 Task

#부스트캠프5기 #컴퓨터비전

Day | 2023.04.12 ~ 2023.04.20 19:00

CV 5조 랩업리포트(Public 5위, Private 2위) 제출용

1	CV_14조		0.7722	81.4762	78	1d
2	CV_05조		0.7668	81.8254	77	20h
3	CV_09조		0.7613	81.3175	64	21h

1. 프로젝트 개요

1-1. 개요

Class Description:

마스크 착용여부, 성별, 나이를 기준으로 총 18개의 클래스가 있습니다.

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

- COVID-19의 전염력으로 인해 경제적, 생산적인 활동이 제약을 받고 있으며, 이를 방지하기 위해 모든 사람이 마스크 착용이 중요하다. 하지만 넓은 공공장소에서 모든 사람들의 마스크 착용 상태를 검사하기 위해서는 추가적인 인적자원이 필요하므로, 사람 얼굴 이미지로 자동으로 마스크 착용 상태를 가려내는 시스템이 필요하다. 이 시스템이 공공장소 입구에 갖춰져 있다면 적은 인적자원으로도 충분히 검사가 가능할 것이다.
- 이미지가 주어지면 마스크 착용 여부, 성별, 나이에 따라 총 18개의 클래스로 구분하게 된다.

1-2. 협업 환경

- 팀 구성 및 컴퓨팅 환경: 5인 1팀, 인당 V100 서버를 VSCode 와 SSH로 연결하여 사용
- 협업 환경: Notion, Slack, GitHub, Wandb, TensorBoard, Google Drive
- 의사 소통: 카카오톡, Zoom, Slack, Offline

1-3. 데이터셋 구성

```
data
├── eval
│   ├── images
│   └── info.csv
└── train
    ├── images
    │   ├── 000001_female_Asian_45
    │   │   ├── incorrect_mask.jpg
    │   │   ├── mask1.jpg
    │   │   ├── mask2.jpg
    │   │   ├── mask3.jpg
    │   │   ├── mask4.jpg
    │   │   ├── mask5.jpg
    │   │   └── normal.jpg
    │   └── ...
    └── train.csv
```

eval: test 데이터셋
-images: 12600개의 이미지 폴더
-info.csv: inference

train: train 데이터셋
-images: 18900개의 이미지 폴더
-train.csv: 2700명의 신상 정보

- 모든 데이터셋은 아시아인 남녀로 구성되어 있고 나이는 20대부터 70대까지 다양하게 분포되어 있음
- 전체 사람 명 수 : 4,500
- 한 사람당 사진의 개수: 7 [마스크 착용 5장, 이상하게 착용(코스프레, 텍스처) 1장, 미착용 1장]
- 이미지 크기: (384, 512)

1-4. 평가 방법

- Submission 파일에 대한 평가는 F1 Score를 통해 진행됨

2. 프로젝트 팀 구성 및 역할

공동 역할: EDA, Backbone 모델 리서치, 모델 성능 실험

이름	역할
오서영	EDA, resnext50 모델 설계 inference EDA, 최종 모델 앙상블 구성 등
이현구	오라벨링 1차 조사, seresnext, Data 특성 파악, augmentation 가설적립 및 실험
김동우	오라벨링 전수조사, resnext101, resnext50 성능 실험, Data-Augmentation Policy 수립 등

이름	역할
정현진	EDA, 다양한 모델 성능 실험, augmentation 실험, SK-Fold ensemble 실험
임재규	swin transformer, convnext 성능 실험, data augmentation 실험, dataset 라벨링 오류 조사

3. 프로젝트 수행 절차 및 방법

3-1. 팀 목표 설정

- 1주차 : 강의 듣기, BaseLine code 이해, 다양한 모델 리서치, 모델 논문 읽기
- 2주차 : Data augmentation, Ensemble 전략 수립

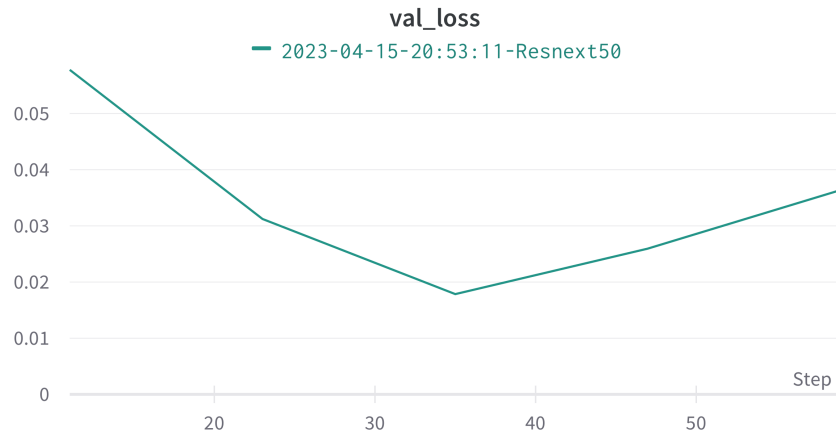
Aa 이름	📅 날짜	☰ 태그
<u>ensemble 및 최종 제출</u>	@2023년 4월 20일	
<u>data augmentation</u>	@2023년 4월 17일 → 2023년 4월 19일	
<u>모델링 및 개인 실험</u>	@2023년 4월 14일 → 2023년 4월 16일	
<u>강의 수강</u>	@2023년 4월 10일 → 2023년 4월 13일	
<u>EDA</u>	@2023년 4월 10일 → 2023년 4월 11일	

3-2. 프로젝트 수행 과정

3-2-1. validation 기준 설정

문제1. validation 지표의 신뢰성이 떨어짐

- 학습할 수록 validation score는 상승하지만 test score는 하락함



해결1. 자체 validation 기준(epochs=3, early stopping=1)

- 일반적으로 에폭 3회까지 val loss가 떨어지다가 4회에 크게 상승하는 패턴
- 그 이전에 학습이 종료되었을 때 좋은 test score
- 이후 학습하면 val loss는 하락하지만 test score는 떨어짐. → 과적합 발생

문제2. 기존 데이터셋을 이미지 별로 구분할 시 Train-Dataset과 Validation Dataset에 같은 사람이 포함됨.

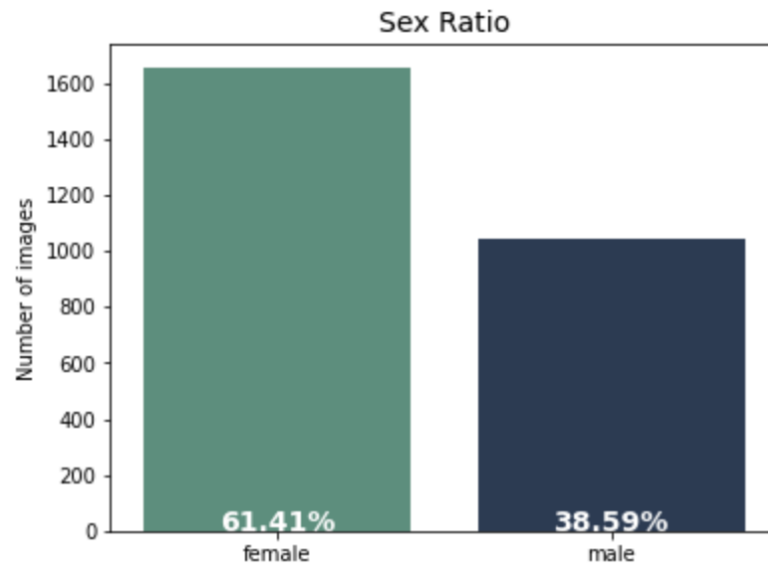
- 위 경우 학습 시 학습할 수록 Validation Score가 항상 높게 나오는 문제점 발생

해결2. 데이터셋을 이미지 별이 아닌 사람 별로 구분하는 Dataset을 구성

- Train-Dataset과 Validation-Dataset에 같은 사람이 포함되지 않게 만듦.
- 더 신뢰할 수 있는 validation score를 위해 데이터셋의 라벨 분포를 유지하며 Train set과 validation set으로 데이터 분할하였음.
- 상기 방법으로 만든 Validation-Dataset을 통해 모델의 Acc 및 F1 Score을 평가하였음.

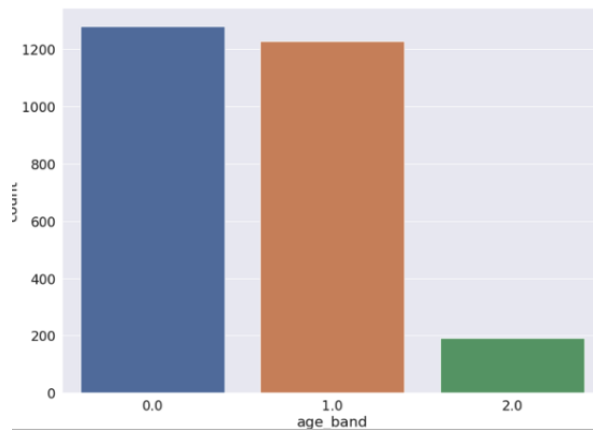
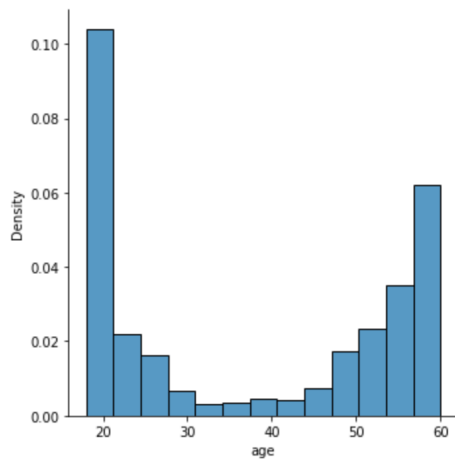
3-2-2. 라벨 불균형

문제1. 성별 불균형



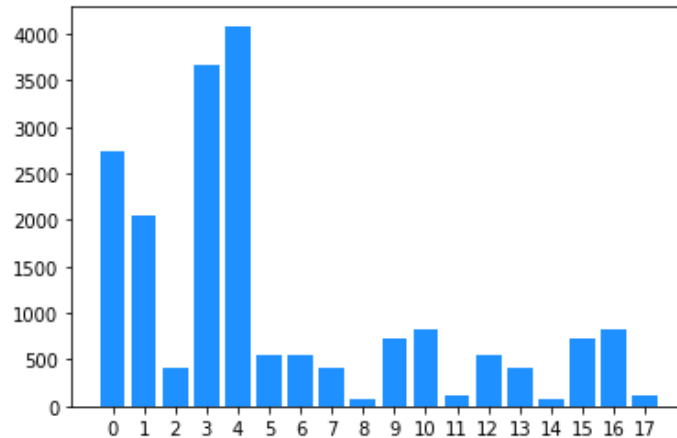
- gender: female(1658) > male(1042)

문제2. 연령 불균형



- age: 18 ~ 60의 범위. 60세 초과 데이터는 존재하지 않음.
- age_band가 3인 60세의 데이터는 전체 2700개 중 192개(약 0.1%)에 불과하여 굉장히 불균형 문제가 심각.

문제3. 라벨 불균형



- 가장 많은 4번 class와 가장 적은 14번 class 간 차이가 4002개
- 특히 age에 관련된 라벨 불균형이 심각함

해결3-1. cutmix를 통해 부족한 라벨의 데이터를 upsampling



```
5      408
2      351
17     108
11     100
14      78
8       65
dtype: int64
```

- 원래 cutmix는 dataset에서 불러온 이미지에 적용하는 방식. 따라서 원래 훈련 데이터의 분포를 따를 수 밖에 없다.
- 그렇다면 부족한 라벨에 대해서만 cutmix를 진행하여 데이터 증강의 효과를 볼 수 있을까? → 추가로 이미지를 생성하여 학습 데이터로 활용해보자!
- 2, 5, 8, 11, 14, 17 라벨(old age)에 대하여 cutmix 실행, 총 555개의 이미지를 생성하였음.
- augmentation은 upsampling보다 전체 훈련 데이터의 분포를 다양화하는 역할로 활용해야 함 → 기존의 cutmix 학습 방법이 나올 수 있음.

해결 3-2. 라벨 재정의

- old 이미지가 부족하고 59세의 경우 old의 느낌이 많아 59세까지를 old class로 분류함.

해결 3-3. Focal Loss

- Hard Case에 높은 loss를 부여하여 라벨 불균형 해결에 적합

문제4. 잘못된 라벨링

- 남자임에도 여자로 라벨링된 것과 같은 오라벨링 데이터가 상당수 존재함.
- 남녀 또는 연령대 별 혼란을 불러 일으키는 데이터 이상치들이 다수 존재함.

해결4. data cleansing

- 먼저 전체 이미지 18,900장에 대한 전수 조사를 통해 약 20~30개의 오라벨링 데이터 검출
→ 라벨 수정
- 성능에 영향을 미칠만한 outlier 50~60개 또한 검출(주로 남녀 및 연령대 혼란이 있는 데이터) → 논의를 통해 임의로 라벨 수정

3-2-3. Backbone 모델 실험

• 실험 통제 조건

```
- Optimizer : AdamW  
- Loss : Focal Loss  
- Learning Rate : 0.0001  
- Learning Rate Scheduler : StepLR (step_size = 3, gamma = 0.7)
```

• 실험 모델 List

ResNet50 , ResNet101 , ResNet152 , ResNext50 , ResNext101 , ResNest50 , Swin Transformer ,
ConvNext-B , DenseNet , Wide ResNet , ViT

• 실험 결과

- ResNest50 , ResNext50 가 가장 좋은 성능을 보임

3-2-4. Data-Augmentation

- 이번 대회에서 제공된 데이터셋은 클래스 불균형 문제가 있었기 때문에 과적합을 완화하기 위해 data augmentation 기법을 활용하여 다양한 학습 데이터를 생성함. 데이터의 종류와 특성에 맞춰 적절한 증강 방법을 선택하여 사용함.

- Data-Augmentation은 첫번째로 원본 Data에서 모델에게 **중요한 feature**를 극대화하고, 불필요한 정보를 제거하는 역할을 함.
- 두번째로 input데이터의 주요 정보가 손실되지 않는 한도 내에서 데이터를 다양하게 변형시켜 모델이 학습 시 데이터를 다양하게 바로볼 수 있도록 하여 모델의 성능을 높이는 역할을 함.

중요 feature 추출

- Grayscale(3)
- crop
- Normalize
- RandomAutocontrast
- RandomAdjustSharpen

다양성 부여

- RandomApply
- RandomHorizontalFlip
- RandomCrop
- RandomPerspective
- ColorJitter
- Gray-Scale 변환 후 얼굴 전체, 이목구비, 옷스타일 정보를 3채널에 저장하는 Transform

3-4. Ensemble 전략 수립

- 단일 모델을 사용할 경우 추론/예측에 편향이 발생할 수 있으므로, 앙상블 기법을 활용하여 더욱 정확한 예측 결과를 도출하였다. 이를 위해, 다양한 모델들을 조합하여 최적의 성능을 발휘하는 앙상블을 구성하였으며, 각 클래스 별로 모델들이 예측한 확률을 합산하여 가장 높은 클래스를 선택하는 soft voting 방법을 사용
- 데이터가 18,900개로 볼륨이 부족하고, 라벨 간 불균형이 심함. 따라서 데이터를 train과 validation set으로 나누기보다 전체 데이터를 기반으로 학습시키는 방법이 더 효과적일 것으로 판단함. 이에 전체 data를 기반으로 학습하면서 validation의 신뢰성을 확보하기 위해 Stratified-K-Fold로 다양한 데이터셋을 구성하여 이를 기반으로 학습시킨 모델 중 Best Top을 선정해서 Ensemble을 진행함.

첫 번째 실험

- ResNeXt50 단일 모델을 Stratified-K-Fold(K=10)로 학습시킨 모델 중 best 3개 모델을 앙상블

→ F1-Score: 0.7598 Acc: 81.3651 달성

오서영_T5125	0.7598	81.3651	상세 보기	2023-04-18 22:33
-----------	--------	---------	-----------------------	------------------

두 번째 실험

- ResNeSt50 단일 모델을 Stratified-K-Fold(K=5)로 학습시킨 모델을 사용
- 다양성을 위해 Age Dataset을 58세 이상, 58세 이상, 59세 이상을 Old로 구성한 모델을 앙상블 함.

→ F1-Score: 0.7619 Acc: 80.7778 달성

이현구_T5167	0.7619	80.7778	상세 보기	2023-04-18 23:42
-----------	--------	---------	-----------------------	------------------

세 번째 실험

- 상기 언급한 모델 6개를 합쳐서 soft-voting 방법으로 앙상블

→ F1-Score: 0.7625 Acc: 81.2063 달성

김동우_T5026	0.7625	81.2063	상세 보기	2023-04-20 17:40
-----------	--------	---------	-----------------------	------------------

최종 Ensemble

backbone	dataset	augmentation	K	option
Resnext50	data	CustomAugmentation	10	
Resnext50	data	CustomAugmentation	10	
Resnext50	data	CustomAugmentation	0(no validation)	
Resnest50	data	CustomAugmentation	5	
Resnest50	data	CustomAugmentation	5	
Resnest50	data	CustomAugmentation	5	

- 다양한 분포를 보인 모델을 앙상블로 구성해보며 최적의 결과를 내는 앙상블을 찾는 실험을 시도
- 많은 실패와 어려움이 있었지만, 서로의 분포를 잘 보완해줄 수 있는 모델을 선택하여 앙상블을 구성하니, 가장 좋은 결과를 낼 수 있었음.

→ **F1-Score: 0.7659 Acc: 81.556 달성**

오서영_T5125

0.7659

81.5556

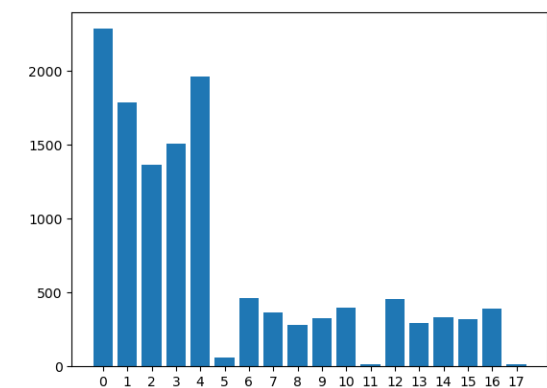
상세 보기

2023-04-20 18:42

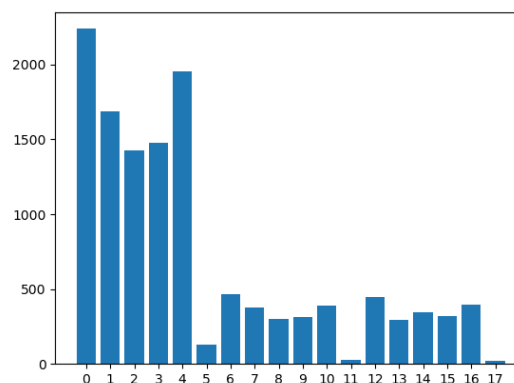
Stratified-K-Fold(K=10) Ensemble 실험

Stratified-K-Fold(K=10)으로 학습 시킨 ResNet 단일 모델을 다양하게 ensemble 해봄.

- 10개 모두 ensemble한 경우



- validation accuracy가 높은 모델 3개만 ensemble한 경우





10개의 모델을 모두 ensemble하는 것보다 3개의 모델만 ensemble하는 것이 class 5, 11, 17 같이 old클래스를 더 잘 분류한다. 하지만 성능이 좋았던 모델과 비교하였을 때 old 클래스를 잘 분류하지 못하는 것으로 보아 성능이 떨어질 것으로 예상된다.

4. 프로젝트 수행 결과

4-1. 결과

Public 5등, f1 score 0.7660 → Private 2등, f1 score 0.7668

5 (-)	CV_05조		0.7660	81.4286	77	1d
2 (3 ▲)	CV_05조		0.7668	81.8254	77	1d

4-2. 최종 제출

- 프로젝트 템플릿

```
code
├─ dataset.py
├─ train.py
├─ inference.py
├─ loss.py
└─ model.py
```

5. 자체 평가 의견

잘했던 점

- Data augmentation을 분류 과제에 아주 알맞게 선정하였음.
- 하나의 모델만을 학습시켜 사용하지 않고 여러 모델을 결합하는 방식으로 성능을 끌어올림
- 개개인의 연구 결과나 성과를 잘 공유하여 팀원들의 전반적인 insight가 상승하였음.
- Augmentation, model 탐색과 같은 효율적인 분업

시도 했으나 잘 되지 않았던 것들

- 하나의 모델에 classifier를 3개 (mask/age/gender) 사용한 결과 오히려 성능이 나빠졌다.
- 성별 구분과 연령대 구분 시 머리 스타일과 옷 스타일의 정보 또한 중요할 것 이라고 생각하여 원본 이미지를 Gray Scale로 변경하고 이목구비, 옷상태와 같이 다양한 형태를 3개의 채널에 담아 학습해보았지만 성능 향상과 크게 연결되지 않았음.

- Task별 3개의 모델을 3개씩 만들고 총 9개의 모델을 앙상블하여 1개의 모델을 만들었으나 성능이 단일 모델에 비해 많이 떨어졌음.
- Focal loss에 라벨 분표별로 weight 추가하여 학습하였으나 데이터 수가 많은 class의 recall이 아주 떨어졌음. weight 산정을 잘못하였다고 판단함.
- Stratified-K-Fold(K=10)으로 학습시킨 모델을 앙상블한 결과 큰 성능 향상을 볼 수 없었다.
- cutmix를 통해 부족한 라벨의 데이터만 증강해보려 했으나 적절한 에폭 수를 맞추기가 어려웠고 성능 향상을 보지 못했다.

아쉬웠던 점들

- 시간 한계 상 데이터 cleansing 작업을 더 많이 시도해보지 못한 점이 아쉽다.
- 모델의 성능 향상을 위해 CNN backbone과 transformer를 결합한 하이브리드 모델을 시도해 보았지만 CNN에서 추출한 작은 feature map는 적합하지 않았다.
- 다양한 종류의 loss함수를 실험해보지 않거나 병합하지 않았다는 점.
- Class balance Loss 적용 시 성능 향상이 있었지만 시간상 적용하지 못하였다.
- Hyper parameter tuning은 해보지 못하였다.
- 전체 데이터에 대해서 cutmix를 시도해보지 못한 점이 아쉽다.

프로젝트를 통해 배운점 또는 시사점

- 이론적으로 좋은 방법이 모든 task에서 좋은 것은 아닐 수도 있다.
- 모델이 이미지를 더 잘 인식하도록 validation와 test 데이터셋에 data augmentation이 적용될 수 있다는 점.
- 모델의 성능을 평가하는 지표가 되는 validation score를 신뢰성 있도록 하기 위해서는 validation set을 잘 구성하는 것이 중요하다.
- 과적합이 실제 데이터를 활용한 task에는 정말 중요한 문제라는 것을 깨달았다.

6. 개인 회고

오서영

학습 목표를 달성하기 위해 시도한 점

이번 프로젝트의 목표는 사용자의 얼굴을 촬영한 이미지에서 사용자의 마스크 착용 여부, 나이, 성별을 추론하는 것이었습니다. 이를 위해서 EDA, 이미지 전처리와 모델 학습 등을 진행하였습니다.

- EDA

본격적인 모델링에 들어가기 전, 데이터에 대한 이해를 위해 EDA를 진행하였습니다. feature 중 age가 old인 사람이 전체 데이터의 약 0.1% 수준이라는 점과 마스크를 착용한 사진이 착용하지 않았거나 잘못 착용한 사진보다 5배 많다는 점을 발견할 수 있었습니다.

- 신뢰할 수 있는 validation

학습 데이터에서 단순히 train과 validation을 나누었을 때, 모델이 validation에서는 accuracy와 f1 score 모두 1에 가까운 성능을 보였습니다. 그러나 test에서는 f1 score 0.65 정도의 현저히 떨어지는 성능을 보이는 것을 관찰할 수 있었습니다.

이로부터 추측할 수 있는 부분은 두 가지 입니다. 학습 데이터와 테스트 데이터의 분포가 다를 수 있다는 점, 그리고 라벨 불균형 문제가 심각하여 단순 split으로 나눈 훈련 데이터에서 특정 라벨에 대한 과적합이 발생할 수 있다는 점입니다. 즉 일반적인 split이 아닌 신뢰성 있는 validation 기준이 필요했습니다.

이를 해결하기 위해 Stratified K-Fold에서 K를 10으로 늘려 안정적인 학습을 진행하였고, 그중 accuracy가 가장 높은 3개의 모델을 앙상블하여 활용하였습니다. 또한 실험적으로 에폭 3회 내 외에서 validation loss가 크게 상승할 때 과적합이 발생하는 점을 발견할 수 있었습니다. 이 점에서 착안하여 에폭을 4회 이내로 줄이고 early stoping(patience=1)를 적용하여 자체적인 validation 기준을 만들었습니다.

- 과적합

모델을 많이 학습시킬 수록 validation loss가 줄어들길래 실제 성능이 향상되는 줄로 알았는데, 에폭 5회, 20회 모두 에폭 3회보다 성능이 하락했습니다. 에폭 20회의 추론 분포를 확인해보니 학습 데이터의 분포를 그대로 따라가는 반면, 에폭 3회는 학습 데이터에 적던 2번이나 5번 라벨에 대해서도 높은 답변율을 보였습니다.

이를 통해 학습 데이터와 테스트 데이터 간의 분포 차이가 있음을 확인하였습니다. 그래서 과적합 방지를 위해 에폭을 4회 이내로 줄이고 early stopping, dropout, weight decay 등의 방법론을 적용하였습니다.

구현 방식

- resnext50 backbone + fc layer(dropout, batchnormalization, relu) 18개 추론 모델

이미지 분류 문제에 최적화된 모델을 리서치 하던 중 resnext50을 발견하여 backbone으로 활용하였습니다. 처음에는 한 개의 출력층만 쌓아 2048개의 노드를 18개로 연결하였는데, 하나의 층을 더 쌓고 dropout과 relu를 적용하여 큰 성능 향상을 보았습니다. 이를 통해 과적합 방지가 이번 task의 핵심임을 다시금 확인하였습니다.

- resnext50 backbone + fc layer(dropout, batchnormalization, relu) mask, gender, age 모델 앙상블

단일모델로 18개의 라벨을 추론하는 것이 아니라 마스크, 성별, 나이라는 세 개의 과제로 나눠서 세 개의 모델을 학습시켰습니다. 단순히 더해서 라벨을 추론하는 방법과 선형 모델에 넣어 학습

시키는 방법 모두 성능 하락이 있어 이후로 활용하지는 않았습니다.

- AdamW

Adam은 빠른 속도로 많이 쓰이는 optimizer이지만, 일반화 성능이 떨어진다고 한다. 그래서 weight decay를 추가한 AdamW를 활용하여 과적합을 방지하고 일반화 성능을 높였다.

- cutmix

학습 데이터가 현저히 적기 때문에, 데이터 증강 기법으로 cutmix를 활용해보려 했습니다. 특히 라벨 불균형 문제를 해결하고자 부족한 old 라벨에 대해서만 데이터를 증강시켰습니다. 증강 데이터와 원본 데이터를 섞어서 함께 활용하고자 했으나 pytorch 구현의 문제로 우선 증강 데이터를 학습시킨 뒤 원본 데이터를 학습시키는 방향으로 진행했습니다.

결론적으로 증강 데이터에 대하여 모델이 biased되어 성능이 악화되는 결과를 낳았습니다. 이 점에서 데이터 증강 기법은 원본 데이터의 분포를 다양화하는 데 활용되어야 하는 것이지, 설계자의 의도대로 데이터의 분포를 변경하는 데 쓰려고 하면 오히려 불안정한 학습을 초래할 수 있다는 점을 배웠습니다.

- 앙상블

Stratified K-Fold에서 accuracy가 제일 높은 top 3 모델로 soft voting하였고 성능이 크게 향상하였습니다. 추가적으로 성능을 높일 길을 찾다가 앙상블은 다른 분포의 모델을 활용할 때 성능 발전이 크다는 내용을 보았습니다.

그래서 top3 중 분포가 비슷한 2개 중 1개를 빼고, 대신 validation 없이 에폭 2번으로 학습시킨 모델을 포함시켰습니다. 최종적으로 제가 설계한 모델 3개와 팀원이 설계한 모델 3개를 soft voting하였습니다. public 기준 f1 0.7659, private에서는 0.7668로 score 향상을 보이며 높은 일반화 성능을 보였습니다.

저희는 팀원끼리 학습환경을 통일하지 않았는데, 오히려 다양한 세팅으로 학습시키는 방법이 일반화에 도움이 되었다는 생각이 들었습니다.

마주한 한계/아쉬웠던 점

- cutmix

cutmix를 전체 데이터셋에 대해서 적용해보고 싶었는데, 촉박한 시간 문제로 진행해보지 못한 점이 아쉽습니다. 다음 프로젝트에서는 원래 방식대로의 cutmix를 구현해보고 성능 변화를 확인하고 싶습니다.

- 협업 환경

협업 환경을 통일하지 않은 상태로 각자의 코드 가지고 프로젝트를 진행했는데, 후반부에 코드를 교환할 일이 있을 때 제 코드와 호환이 되도록 수정해야 해서 불편했습니다. 다만 저희 팀의 시너지는 다양한 학습환경에서 왔다고 생각하여 전체 학습 세팅을 맞추는 필요는 없을 것 같습니다. 다음부터는 기본 베이스라인 코드를 통일하여 구현 방식을 맞춤으로써 코드를 복붙해도 이용하기 쉽도록 진행하고 싶습니다.

- config

발표를 들어보니 yaml config 파일을 통해 창을 닫아도 학습이 진행되도록 구현한 팀이 있었습니다. 저는 매번 코드를 실행하고 노트북을 실행하고 있어야 해서 불편한 점이 많았는데, 다음부터는 yaml 파일을 활용해 보아야겠습니다.

이현구

학습 목표를 위해 시도한 점

학습 목표

이번 task에서 필요한 이론을 적립하고 적립된 이론을 바탕으로 해당 업무를 위한 코드를 파이썬으로 구현하고 다양한 기법들을 실험하여 결과를 도출해 보는 것

- 평가 지표 재설정
 - 베이스 코드에 dataset은 train과 val에 같은 얼굴이 들어가므로 같은 얼굴이 중복되지 않도록 코드를 수정함.
- sk-fold 구현
 - 데이터 셋이 적은 문제를 해결하기 위해 평가 지표를 재설정 후 18개의 class가 모두 같은 분포로 들어가도록 sk-fold를 구현함.
- 모델 앙상블
 - 더 높은 성능을 얻기 위해 sk-fold로 3개의 데이터셋을 만들어 3개의 모델을 train하고 이 모델들을 앙상블하였음.
- Data augmentation
 - 모델에게 최적의 Data를 feeding하기 위해 normalize를 하지 않았으며, 흑백 변환, 주요 특징 구간 crop을 진행하였다.

한계 및 아쉬웠던 점

- 18개의 class를 3개의 task로 나누고 3개의 모델을 만들어서 학습 및 앙상블을 진행하였는데 결과가 좋지 않아서 아쉬웠다.
- git 등을 활용한 협업이 부족하였다.
- 아웃라이어 데이터를 학습에 활용하지 않는 기법 등의 아이디어가 부족하였다.
- validation 시 모델 지표를 확인 할 때 각각의 class에 대한 loss 및 recall을 확인하지 않아 특정 class에 대한 정확도 확인을 제대로 수행하지 못하였다.

앞으로 시도할 점

- 좀더 정확한 지표 설정
- git 활용 등 협업 툴 사용 및 협업
- 다양한 모델 탐색 및 모델 수정

김동우

1. 나는 내 학습목표를 달성하기 위해 무엇을 어떻게 했는가?

- 이번 Task는 이미지를 성별, 나이, 마스크 착용 여부에 따라 18개의 클래스로 분류하는 임무였다. 이를 완수하기 위해 먼저 Image-Classification에 적합한 모델와 훈련 방법에 대한 리서치를 진행하였다. 모델 리서치 및 학습 방법에 대한 지식을 쌓은 후 이를 실험해보기 위해 Dataset, train, inference 등등 훈련과 학습에 필요한 모든 코드들을 Pytorch template에 맞게 파이썬을 이용해서 직접 구현해보았다. 그 후 학습한 지식을 바탕으로 논리적인 검증을 바탕으로 한 가설을 세우고 이를 코드로 구현하며 실험을 통해 가설을 검증하는 방식으로 학습을 진행하였다. 특히 Data-Augmentation과 앙상블에서 많은 방법론들을 실험해보았는데 그 과정에서 많은 실패도 있었지만 나름 소기의 성과도 거둘 수 있었던 것 같다.

2. 마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?

- 2주라는 시간이 제한된 관계로 모델 개선을 위한 다양한 각도의 접근을 해보지 못 한 것이 아쉽다. 이번 Task에서는 성능 향상을 위해 더 좋은 모델을 찾고 성과가 잘 나온 여러 모델을 앙상블하는, 모델과 관련된 접근법을 많이 사용한 것 같다. 시간이 더 있었다면 불균형한 데이터를 어떻게 보완하고 Outlier를 어떻게 처리하는 것과 같은 데이터 관련 접근을 더 해보려고 했는데 시간 관계상 그렇게 하지 못한 것이 매우 아쉽다.

3. 한계/교훈을 바탕으로 다음 프로젝트에서 시도해보고 싶은 점은 무엇인가?

- 다음 프로젝트에서는 초반 데이터 처리 작업에 좀 더 집중해보고 싶다. 데이터에 대한 이해가 부족한 상황에서 모델에 바로 접근을 하다보니 뭔가 모델이 발전할수록 사상누각 같은 느낌이 들었던 것 같다. 다음에는 데이터에 대한 이해를 바탕으로 데이터 불균형, 이상치 등 다양한 이슈들을 먼저 탄탄하게 해결하고 모델 쪽으로 넘어가면 좋을 것 같다.

정현진

학습 목표

데이터 분석부터 학습, 추론까지 모든 과정을 경험해보고 이론으로 학습했던 기법들을 실제 적용해 보는 것이 이번 프로젝트의 개인적인 학습 목표였다. 최대한 다양한 기법을 사용하고 실제로 성능이 좋아지는지 확인해보고 싶었다.

모델 개선 시도

ResNet, MobileNet, Inception V3, DenseNet, ViT와 같이 다양한 모델을 사용해보았다. 모두 이미지 분류에서 좋은 성능을 보여준 모델인데 마스크 착용 상태 분류에서는 꽤 다른 성능을 보여줬다. 하지만 모두 이번 task에 가장 적합한 모델을 아니었다.

60세 이상 클래스를 잘 분류하지 못하는 것이 data imbalance 문제라고 생각하고 Focal loss에 Effective number를 weight로 사용하는 CB-Focal loss를 사용해 보았다. CB-Focal loss를 사용하면 확실히 60세 이상 클래스를 더 많이 분류하는 것을 확인하였다.

배운 점

이번 프로젝트에서 가장 중요했던 것은 validation score를 신뢰성 있게 만드는 것이었다. 모델의 성능을 비교해보기 위해서는 우선 validation set을 잘 구성하여 성능 비교의 지표를 만드는 것이 우선되어야 함을 배웠다.

Normalize나 모델 freeze, 다양한 augmentation과 같이 모델의 성능을 향상시킨다고 배웠던 기법들이 오히려 모델의 성능을 저하시켰다. 좋다고 알려진 방법이 모든 task에서 좋은 것이 아니며 현재 task를 잘 분석하여 그에 맞는 기법을 적용해야 좋은 결과를 얻을 수 있을 것 같다.

한계 및 아쉬운 점

실험을 설계함에 있어서 어떠한 가설을 세우고 이를 증명하는 방식이 아니라, 그저 다른 사람들이 좋다고 하는 방법을 적용해보는 방식으로 실험을 하였다. 과정을 되돌아보니 현재 task를 분석하고 문제를 해결하고자 하는 것보다는 점수를 올리기 위한 실험에 치중되어 있는 것 같아 아쉽다.

다음 프로젝트에서 시도할 것

- 실험을 설계할 때 문제가 무엇인지 파악하고, 가정을 세워 실험을 진행할 것이다.
- 실험 결과, 알게 된 점을 더 적극적으로 공유할 것이다.
- github, wandb 같은 도구를 사용하여 협업에 집중하고 싶다.

임재규

학습 목표를 달성하기 위해 시도한 점

저는 강의에서 배운 내용을 보완하고 더욱 구체적으로 이해하기 위해 인터넷이나 온라인 동영상 등 다양한 자료들을 찾아보고 있습니다. 또한, 이번 대회에서 사용한 다양한 기술과 모델들이 왜 우수한 결과를 보이는지에 대해서는 관련 논문들을 깊이 있게 읽어보고 있습니다. 이를 통해, 논

문에서 제시된 실험 결과와 분석을 바탕으로 이번 대회에서 어떤 모델과 기술을 적용하는 것이 가장 적합하며, 최고의 성능을 발휘할 수 있는지에 대해 생각해보고 있습니다.

마주한 한계와 아쉬웠던 점

일반적으로는 모델의 깊이가 깊을수록 더 좋은 성능을 나타낸다는 것이 널리 알려져 있지만, 이번 대회에서는 실험 결과가 예상과는 달리 나타났습니다. 너무 얇은 모델을 사용할 경우 성능이 좋지 않은 것은 사실이지만, 적절한 깊이를 사용하는 것이 중요하다는 것을 발견하게 되었습니다.

또한 아쉬웠던 점은 다양한 종류의 loss 함수를 실험해보지 않거나 병합하지 않았다는 점입니다. 팀원들과 합의하에 focal loss를 사용했지만, 추가적으로 나이를 예측하는 데 중요한 역할을 하는 loss에 가중치를 더해 더 나은 성능을 얻을 수 있을 것이라고 생각합니다.

한계 또는 교훈을 바탕으로 다음 프로젝트에서 시도해보고 싶은 점

앞으로의 프로젝트에서는 기존 모델들을 사용하는 것이 아닌 직접 모델을 제작하고자 합니다. 이렇게 함으로써 성능보다는 모델 제작 과정에서 직접적으로 경험을 쌓으며 모델에 어떠한 변화가 성능을 영향을 미치는지 더욱 구체적으로 이해할 수 있을 것이라고 생각합니다.

모델 개선 방식

먼저, 제가 다양한 모델들을 실험해 보았습니다. 그리고 그 중에서 가장 우수한 모델을 선택한 후, 자체적인 기술을 도입하여 성능을 높이는 방안을 선택하였습니다. 이번 대회에서 제공된 데이터셋은 클래스 불균형 문제가 있었기 때문에 과적합을 완화하기 위해 데이터 증강 기법을 활용하여 다양한 학습 데이터를 생성했습니다. 데이터의 종류와 특성에 맞춰 적절한 증강 방법을 선택하여 사용했습니다.

또한, 단일 모델을 사용할 경우 추론/예측의 편향이 발생할 수 있으므로, 앙상블 기법을 활용하여 더욱 정확한 예측 결과를 도출하였습니다. 이를 위해, 다양한 모델들을 조합하여 최적의 성능을 발휘하는 앙상블을 구성하였으며, 각 클래스 별로 모델들이 예측한 확률을 합산하여 가장 높은 클래스를 선택하는 soft-voting 방법을 사용하였습니다.

해본 시도 중 어떠한 실패 경험과 실패의 과정에서 얻은 교훈

저는 모델의 성능 향상을 위해 cnn backbone과 vision transformer를 결합한 하이브리드 모델을 시도해 보았습니다. 그러나 vision transformer는 정사각형 이미지를 입력으로 받아야 하기 때문에, cnn에서 추출한 작은 이미지는 적합하지 않은 문제가 있었습니다. 이에 따라, 최종적으로는 단일 vision transformer를 사용하게 되었습니다. 하이브리드 모델은 성능이 우수할 수 있지만, 단지 리더보드 성적 올리기만이 아닌 데이터셋과 서비스 사용 가능성을 고려하여 해당 모델을 적용하는 것이 적절한지 미리 판단하는 것이 중요하다는 결론을 도출했습니다.

한 행동의 결과로 얻은 깨달음

이전에는 validation과 test 데이터셋에 data augmentation을 적용하는 것은 지양해야 한다는 고정관념을 가지고 있었습니다. 하지만 최근에는 모델이 이미지를 더 잘 인식하도록 도와줄 수 있는 data augmentation이 적용될 수 있다는 것을 깨달았습니다. 그러나 이를 적용할 때, train 데이터셋에서 사용한 것과 동일하게 valid와 test에 적용해선 안된다는 것도 배웠습니다. 이유는

valid와 test 데이터셋에서는 random crop, random rotation 등의 기법이 실제 데이터와 다를 가능성이 있기 때문입니다. 따라서 이러한 기법들은 제거하여 학습을 진행했습니다.