Image Classification

Wrap up Report

2021/08/23 ~ 2021/09/02

Round26 (26 조)

boostcamp

목차

I. 프로젝트 개요

1. 프로젝트 주제

COVID-19 의 확산으로 공공 장소에 있는 사람들은 반드시 마스크를 착용해야 할 필요가 있으며, 무엇 보다도 코와 입을 완전히 가릴 수 있도록 올바르게 착용하는 것이 중요하다. 따라서 이번 프로젝트 목표는 사진을 입력 받아 나이와 성별에 따른 마스크 착용 상태를 분류하는 것이다.

2. 프로젝트 개요

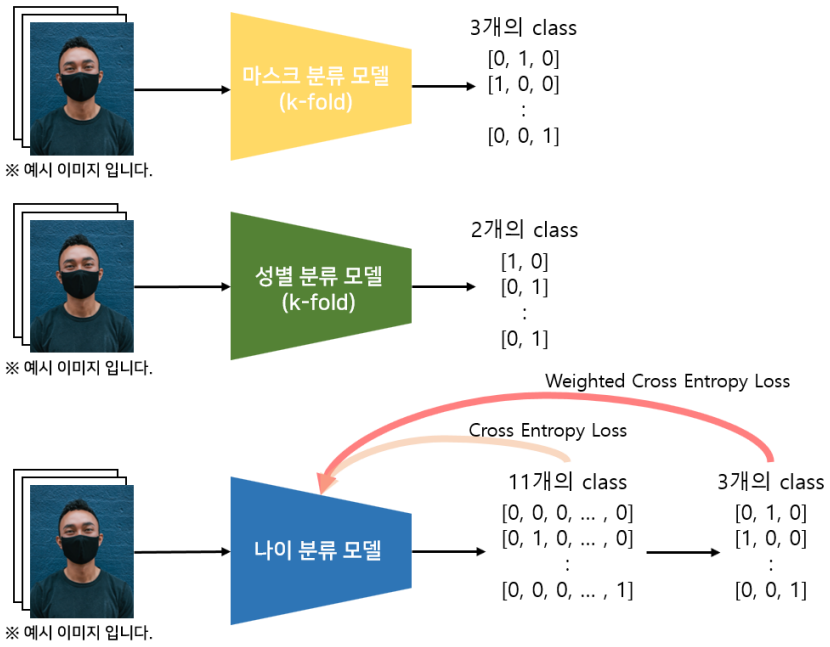
문제 정의 : 카메라로 촬영한 사람 얼굴 이미지의 마스크 착용 여부/나이/성별을 판단하는 작업
(18 개의 클래스를 나이(3 개), 성별(2 개), 마스크 착용 상태(3 개)로 세분화하여 분류)

입력값 : 2700 명의 마스크 정착용, 미착용 혹은 오착용한 사진(사이즈 : [512, 384, 3])
결과값 : 총 18 개의 클래스를 예측 (결과값으로 0~17 에 해당되는 숫자)

3. 활용 장비 및 재료 (개발 환경 등)

GPU : AI Stages - NVIDIA V100
Python IDE : Jupyter Notebook, VSCode / Visualization tool : Wandb

4. 프로젝트 구조



5. 기대 효과

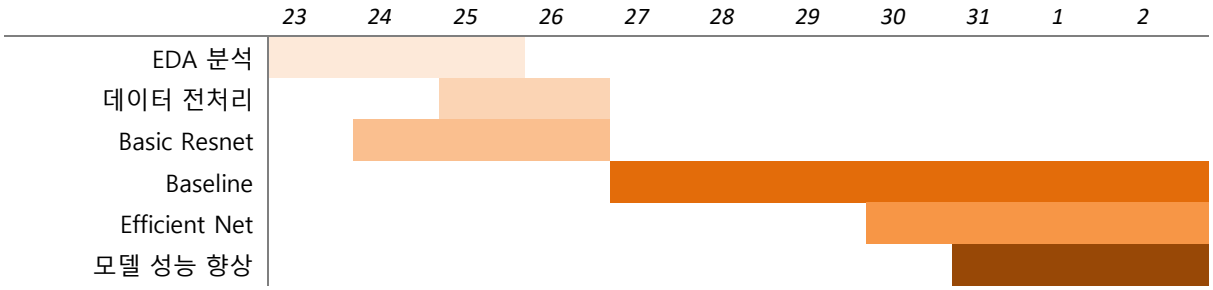
카메라로 비춰진 사람 얼굴 이미지만으로 마스크 정착용 여부를 자동으로 가려낼 수 있는 시스템이 공공장소 입구에 갖춰져 있다면 적은 인적자원으로도 충분히 검사가 가능할 것이다.

II. 프로젝트 팀 구성 및 역할

- **이기성**(팀장) : 시각화툴 도입에 기여, 솔선수범
- **고재욱**(팀원) : 데이터 EDA 를 분석하는데 집중
- **김다인**(팀원) : 모델 성능 향상 기여
- **김현욱**(팀원) : 외부 데이터 조사
- **문석암**(팀원) : 데이터 Augmentation 을 여러 방면으로 실험
- **이예빈**(팀원) : 커다란 프레임워크를 잡는 것과 모델 구현에 기여

III. 프로젝트 수행 절차 및 방법

- **Gantt chart**



IV. 프로젝트 수행 결과

1. 탐색적 분석 및 전처리

분석

성별 (남 1042 / 여 1658) -> 38.59% : 61.41%

한 사람 당 이미지 비율 : 마스크 착용 / 잘못 착용 / 미착용 (5:1:1)

나이 : 아래 표 참조

전처리

1. 데이터 셋 Face-Crop
2. 나이 클래스의 불균형을 완화시키기 위해 11 개의 클래스로 세분화하여 클래스 당 데이터의 개수를 균일하게 만들었다. (한 클래스당 데이터 약 200 개가 되도록)

나이(세)	18	19	20	21~24	25~29	30~48	49~52	53~55	56~57	58~59	60	총계
사람수	1281					1227					192	2700
사람수	192	407	267	233	182	256	252	246	229	224	192	2700
class	0	1	2	3	4	5	6	7	8	9	10	11
Age label	0					1					2	3

2. 모델 개요

제출 모델 기준 나이, 성별, 마스크 착용 상태를 분류하는 독립적인 모델을 구현하여 최종 클래스로 합산하였다. 특히 나이 분류 모델은 다른 모델과 달리 비교적 낮은 성능을 보여 다양한 시도를 통해 성능을 올리는데 많은 노력을 했다.

3. 모델 선정 및 분석

ResNet18, VGG16, Efficientnet-b4, MultiLabelCNN 등의 모델을 시험해 보았고, 최종적으로 validation f1 score 가 높은 Efficientnet-b4 를 사용하기로 했다.

모델의 구성 방식도 크게 2 가지가 있었는데, 하나의 모델로 18 개의 클래스를 분류하는 방법과 성별, 마스크, 나이를 각각 구별하는 모델 세 개를 만드는 경우였다. 그리고 이 때 나이 분류 모델의 경우 성별 예측 결과를 토대로 남자인 경우와 여자인 경우를 각각 예측할지, 혹은 성별과 나이를 독립적으로 판단하여 분류할지도 추가적으로 고민했다.

하나의 모델로 18 개의 클래스를 분류하는 경우, 성능이 나쁘지 않았지만 마스크 착용 여부와 성별, 나이 분류 문제는 독립적인 문제라고 생각하여 맞지 않다고 판단했다.

성별, 마스크, 나이를 각각 구별하는 모델 세 개를 만드는 경우, 각각의 모델에 맞게 개선하여 성능 향상을 기대하였으며 특히 성능 부분에서 가장 분류가 잘 되지 않는 나이 분류 모델에 성능을 향상시키기 위한 방안으로 제안 되었고 이를 통해 실제로 Validation 에서 나이 분류 모델의 성능이 비교적 낮은 것을 확인하였다. 따라서 나이 분류 모델의 성능만 향상시킨다면 전체적인 성능 향상을 기대해볼 수 있었다.

4. 모델 평가 및 개선

마스크, 성별 분류 모델

같은 사람의 사진 7 장은 모두 같은 validation set 에 들어가도록 stratified 5-fold set 을 구성해 모델을 평가할 수 있었다. 마스크/성별 분류 모델 모두 loss function 으로 Cross-Entropy 를, optimizer 를 Adam 으로 사용했을 때 적은 epoch 에서도 평균 0.97~0.99 정도의 충분히 좋은 validation f1 score 를 보여주었기에 나이 분류 모델에 남은 시간을 더 투자해보기로 판단하였다. 따라서 마스크/성별 분류 모델은 추가 실험을 거치지 않고 5-fold 학습 결과를 바탕으로 OOF 앙상블 방법을 적용한 추론 과정을 거쳐 최종 선택했다.

나이 분류 모델

나이를 3 개의 클래스로 분류한 모델의 경우에는 validation f1 score 가 80, 나이를 11 개의 클래스로 분류한 경우에는 3 개의 클래스에 대한 validation f1 score 가 85 가 나왔다.

나이를 11 개의 클래스로 세분화한 후 학습 시 같은 사람의 이미지가 모델 평가시 들어가지 않도록, 나이 클래스가 균등하도록 train 과 validation dataset 을 나누었다. 모델 아웃풋과 정답값 사이의 Cross-Entropy loss 뿐만 아니라 모델 아웃풋을 0 과 1 사이 값으로 정규화하여 더해 3 개의 클래스에 대한 확률값과 정답값 사이의 weighted Cross-Entropy loss 를 추가적으로 사용해 모델을 학습시켰다.

V. 모델을 개선하기 위해 했던 노력들

1. Data Augmentation

데이터셋의 특징 때문에 색조/채도, blur, 좌우반전 등의 Augmentation 기법들을 적용했으나 큰 효과를 보지 못 했다. 데이터의 train 과정에 배경이 영향을 주지 않을까 해서 배경을 제거하고 얼굴 인식을 사용해서 face crop 과정을 진행했다. 그리고 Cut-Mix 를 적용하여 이미지를 반으로 잘라 모델이 이미지 반만 보고도 예측 할 수 있도록 시도를 했다.

최종적으로 각각의 모델의 여러 Data Augmentation 을 적용했으나 성별, 마스크 분류 모델은 Base Augumentation 에서 성능이 가장 좋았으며, 나이 분류 모델은 Base Augmentation 과 Face crop 을 적용하였다.

2. Loss Function

기본적으로 Cross-Entropy loss 를 사용해서 모델의 f1 score 를 비교했다. 나이를 분류하는 모델에서 오차가 많이 발생해 이를 줄이기 위해 f1 loss, focal loss, label-smoothing loss 를 사용했지만 더 나은 효과를 보지는 못 했다.

3. Hyperparameter

성별, 마스크 분류 모델은 최종적으로 배치 사이즈 12, Learning rate 0.0001, Adam optimizer, StepLR scheduler 을 적용했으며, 나이 분류 모델은 배치 사이즈 32, learning rate 0.0001, Adam optimizer, StepLR scheduler 을 사용했다. 최종 제출본에서는 제출 횟수의 부족으로 사용하지 못했지만 타 모델에서는 Cosine annealing lr scheduler 를 활용하여 validation 에서 성능향상을 관찰하였다.

4. 시연 결과

- 모델 성능

F1 (Rank)	Accuracy		
0.729 → 0.712	77.603 → 76.825	0.715 → 0.712	77.064 → 76.825
0.723 → 0.705	79.079 → 78.302	0.712 → 0.702	75.032 → 74.032
0.718 → 0.716	78.810 → 78.492	0.712 → 0.723	75.651 → 75.667

VI. 자체 평가 의견

1. 잘한 점들

- K-fold validation 을 통해 학습한 모델의 성능을 자체 평가하고 OOF 앙상블을 통해 추론 과정을 거친 제출물들의 경우가 그렇지 않았던 제출물들 보다 private 에서 점수 변동이 훨씬 적음을 대회가 끝나고 확인할 수 있었다.
- 마지막에 성능이 좋은 코드를 공유하여 남은 시간 동안 나이 분류 모델에 집중하였다.
- 포기하지 않고 끝까지 다양한 시도를 하여 배운 것이 많았다.
- 좋은 팀원을 만나 협업을 배울 수 있었다.

2. 그 외 시도했으나 잘 되지 않았던 것들

- Vision Transformer 사용
- 전통적인 Machine learning (SVM 등) 실험
- Efficientnet-b4 와 Resnet18 을 앙상블 시도

3. 아쉬웠던 점들

- 시간이 더 있었다면 나이 분류 모델에도 K-fold 기법을 적용했을텐데 아쉽다.
- 다양한 시도를 했음에도 Resnet18 의 public 스코어를 Efficientnet-b4 가 따라잡지 못했다. 대회 기간 내내 Resnet18 의 기본 코드로 제출한 점수를 따라잡기 위해 노력을 많이 했다.
- 여러가지 모델 전처리 방법들을 적용해보았는데 대부분 오히려 점수가 조금씩 내려갔다. 어느 방법에서도 희망을 보지 못하니 점점 baseline 에 머무는 느낌이었다.
- Stratify 한 K-fold validation set 을 너무 늦게 구축한 것이 아쉽다.
- Pseudo labeling 을 미리 적용해보지 못한 것이 아쉽다. 마지막 1 번의 제출에서 pseudo labeling 을 적용한 코드를 내보았는데 성능이 나쁘지 않았다. Pseudo labeled image 는 validation 으로 쓰지않고 미리 구축해놓은 5-fold 데이터셋의 train 부분에만 넣어 모델을 학습시켰다면 좋은 성능을 기대해볼 수 있었을 것 같다.
- 전체적으로 코드가 완전히 통일되지 않는 느낌이 있었다. baseline 을 기반으로 두 번째 주 월요일부터 깃헙을 이용해 공동작업했으면 어땠을까 생각한다.
- 최종 제출본을 선택할 때 리더보드 스코어가 큰 모델에 현혹되지 않고 근거와 신뢰성이 뚜렷한 모델을 선택하는 것이 좋았다.