

Image classification Wrap-up Report

CV-7조 SCV

박상언 송지민 오왕택 이동호 지현동

1. 프로젝트 개요

1-1. 프로젝트 주제

우리는 사람 사진을 바탕으로 해당 사람이 마스크를 제대로 착용했는지, 성별 그리고 나이대를 분류하는 작업을 하였다. 마스크 라벨의 경우에는 착용하지 않는 Normal, 착용은 했지만 제대로 착용하지 않은 Incorrect 그리고 정상적으로 마스크를 착용한 Mask 라벨로 이루어져 있다. 성별은 남과 여로 나뉘어져 있고, 나이대는 30대 미만은 young, 30대 이상 그리고 60대 미만은 middle, 60대 이상은 old로 라벨링 되어 있다. 이번 프로젝트는 작게 보게 되면 총 3 가지 작업에 대해서 분류를 진행하면 되고, 최종 라벨은 각 3개의 분류 라벨의 조합으로 총 18개의 라벨을 분류하는 문제로 볼 수 있다.

Normal



Mask



Incorrect



2. 프로젝트 팀 구성 및 역할

이름	역할
박상언	CutMix 실험, 모형 실험, Git branch 관리
지현동	마스크, 성별, 나이 따로 실험 후 못 맞추는 문제 판별, 모형 실험, Mixup 실험
오왕택	Baseline code 작성, YOLO v8 Ultralytics로 사진 탐지, 모형 실험, Mixup 실험
이동호	GradCAM, 나이 회귀 모형 실험, 사전 학습된 U-Net으로 사람 분할, Wrap-up report 작성, 모형 실험
송지민	Data augmentation 기법 정리, Mixup 실험,

3. 프로젝트 수행 절차 및 방법

3-1. Time-line

기간	수행한 일
12월 11일 ~ 12월 12일	강의 수강, 제공 데이터 및 코드 확인
12월 13일 ~ 12월 14일	Baseline code 작성, Git branch 생성
12월 15일 ~ 12월 17일	데이터 살펴보기(EDA), 잘 맞추지 못하는 부분 확인, 나이 회귀 모형, 여러 가지 증강 기법 실험
12월 18일 ~ 12월 19일	분할 및 탐지 기법 적용, 여러 모형 실험
12월 20일 ~ 12월 21일	CutMix, Mixup 적용 및 여러 가지 손실 함수 적용, 모형 실험

3-2. 프로젝트 수행 방법

1. 데이터를 직접 하나씩 살펴보기
2. EDA 수행
3. Baseline code 및 logging tool 선택
4. Multi class vs Multi task 선택
5. 여러 증강 기법 적용
6. 탐지 및 분할 기법 적용
7. 모형 선택 및 손실 함수 등 하이퍼파라미터 조정

4. 프로젝트 수행 결과

4-1. 데이터 살펴보기

팀원들과 데이터를 나누어서 직접 하나씩 살펴보기로 하였다. 그리고 잘못된 정보가 있는지를 확인하였다.

우선 데이터의 정답 정보가 틀린 것을 확인하였다. 외형상 남자임에도 불구하고 여자의 성별을 지닌 데이터와 그 반대의 데이터들도 존재하였다. 해당 부분에 대해서는 라벨링을 수정하지 않고, 일종의 노이즈로 생각을 하여 그대로 학습을 진행하였다. 이와 같은 노이즈를 같이 학습할 경우, 모형의 일반화 성능이 올라갈 것으로 기대를 하였다.



두 번째는 나이의 old라는 라벨의 기준이 모호하다는 것이다. 데이터를 살펴보게 되면, 58세와 60세 인물의 외형 차이가 크게 나지 않는다. 그래서 해당 부분에 대해서도 고민을 하게 되었다.

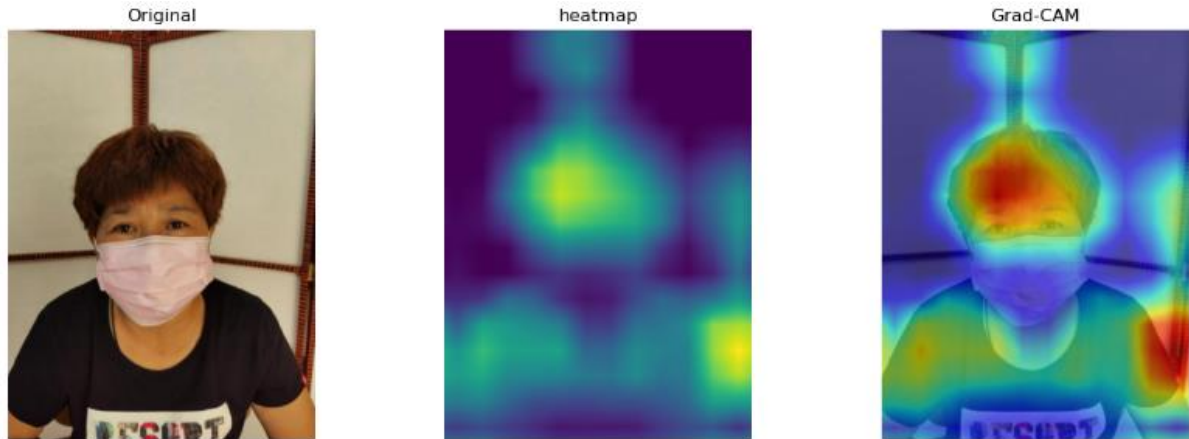
세 번째는 인물의 나이를 정확히 판별하기 위해서는 해상도를 유지해야 한다고 생각을 하였다. 그리고 증강 기법을 사용하게 되면 선명도 또는 주름 등을 부각시킬 수 있는 기법을 적용해야 된다고 생각을 하였다.

4-2. EDA

데이터를 살펴보게 되면, 전체적으로 라벨의 불균형이 심하다는 것을 알 수 있었다. 이에 따라 우리는 3 개의 작업을 나누어서 수행을하기로 하였다. 우리는 ResNet50으로 마스크 판별, 성별 판별 그리고 나이 판별을 수행하였다. 해당 모형은 마스크와 성별에 대한 부분에 대해서는 좋은 F1-score를 보여주었다. 하지만 나이에 대한 예측에서는 낮은 성능을 보였다. 그래서 나이를 가지고 ResNet50으로 회귀식을 추정하여 60세를 대략 몇 세로 예측하는지를 실험하였다. 실험 결과, 60세를 대략 57세에서 58세로 예측을 하

였다. 그래서 우리는 old라는 라벨의 기준을 58세로 변경하고 추가 분석을 진행하였다. 다만, 해당 실험은 60세를 두었을 때의 모형 성능이 더 높게 나타나 최종적으로는 원래 라벨을 기준으로 모형을 실험하였다.

GradCAM을 통해서 나이를 예측할 때, 어느 부분을 중점적으로 보는지를 확인하였다.



일반적으로는 마스크 위 부분 또는 의상에 대해서 모형이 집중되는 것을 알 수 있었다. 하지만 몇몇 사진에서는 배경에도 집중되는 부분이 있어 해당 부분은 추후에 분할 및 탐지 기법을 통해서 해결하기로 하였다.

4-3. Multi class vs Multi task

실험 결과, 작업을 나누는 것보다 총 18 개의 라벨을 예측하는 작업으로 진행한 것이 성능이 더 좋아서 전자의 방식을 선택하였다.

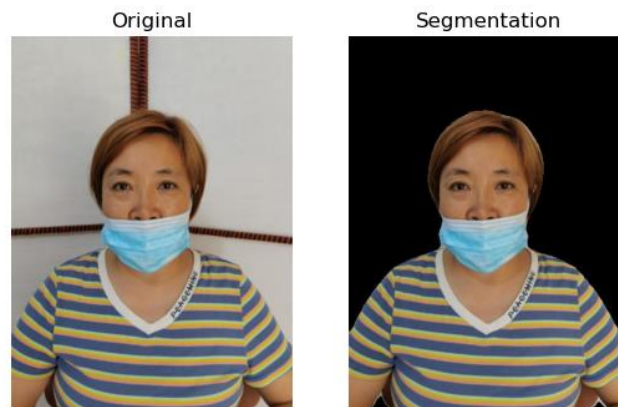
4-4. 여러 증강 기법 적용

Albumentations에서 제공하는 증강 기법 모두를 우리 데이터에 시각화를 진행하였다. 그리고 마스크와 성별은 잘 맞추는 반면에 나이를 잘 판별하지 못하기 때문에 어떠한 기법이 나이를 판별하는데 도움이 될 지를 생각하였다. Color jittering, Gaussian Noise, Gray Scale, Flip, Clahe 등을 적용하여 적용하기 전과 성능 실험을 해보았다. 우리는 전체적인 색감 조정이 나이 판별에 주요할 것이라고 기대를 하였지만, 실제 성능 개선에는 큰 영향이 없었다.

우리는 추가적인 증강 기법으로 CutMix와 Mixup을 적용해 보았다. GradCAM에서 사람의 눈 부분에 활성화가 잘 된다는 점을 생각하여 해당 증강 기법의 부분을 눈 부분의 좌표를 지정하여서 수행을 하였다. 이와 같은 증강 기법은 실제로 성능 향상을 이끌어 주었다. 하지만, 좌표를 지정하는 경우보다 임의로 영역을 정하여서 Mixup을 수행하는 것이 최고의 성능을 보여 해당 방법을 선택하였다.

4-5. 탐지 및 분할 기법 적용

GradCAM에서 확인한 바로는 모형이 인물에 집중을 하지 못하고 배경 부분에 집중을 하는 경우가 있었다. 이러한 부분을 바로 잡기 위해서 배경을 제거하는 작업을 수행하였다. 총 세 가지 방법을 시도해보았다. 첫 번째는 YOLO v8의 Ultralytics를 사용하여서 인물의 경계 상자만을 사용하는 방법, 두 번째는 사전 학습된 U-Net을 사용하여서 인물을 분할하여 사용한 방법, 마지막으로 두 개를 동시에 사용한 방법이다. 최종적으로는 탐지 모형만을 사용한 경우 모형의 성능이 제일 좋았다. 그리고 두 번째로 좋은 것은 분할 기법만 적용할 때였다.



4-6. 모형 및 성능 평가

모형은 ConvNext와 EfficientNet b6를 사용하였다. 기본적으로 해당 작업에서는 트랜스포머 기반 모형보다 합성곱 신경망 모형이 귀납적 편향이 더 크기 때문에 좋은 성능을 낼 것으로 기대를 하였다. 성능은 아래와 같다.

기법 및 모형	F1-score	정확도(%)
분할 + CutMix + EfficientNet b6	0.7453	79.6349
탐지 + Mixup + ConvNext (Private 4등)	0.7502	79.6508

5. 자체 평가 의견

5-1. 잘했던 점

- 프로젝트를 처음 진행하는 사람들도 하나의 작업을 처음부터 끝까지 수행을 할 수 있도록 하였다.
- 데이터를 직접 눈으로 파악하고, 기계의 입장에서 해당 작업을 어떻게 해야 잘 맞출 수 있을지를 고민하였다.

- Git, wandb 등 다양한 협업 툴을 사용하였다.

5-2. 아쉬웠던 점

- 처음에 시간 분배를 잘하지 못하여 모형 실험 부분을 많이 못하였다.
- Git에 보다 다양한 기능들(issue, merge 등)을 수행해보지 못하였다.
- 다양한 하이퍼파라미터 조정 실험 등을 못하였다.

5-3. 프로젝트를 통해 배운 점

- 여러 모형을 돌려가면서 실험을 하는 것보다 전처리 단계부터 하나씩 논리적인 근거를 찾아서 실험을 해보는 것의 중요함을 느꼈다.
- 데이터를 직접 보고 작업을 수행하는 것과 그냥 수행하는 것의 큰 차이를 느낄 수 있었다.

개인 회고

박상언

나는 내 학습목표를 달성하기 위해 무엇을 어떻게 했는가?

데이터 불균형을 해결하기 위해 data argument를 사용하고 실험을 통해 더 효율적인 argument 기법과 모델을 찾으려고 하였다.

Resnet과 gradCam을 통해 확인한 결과 마스크의 착용 여부와 성별은 잘 맞추었으나 나이를 구분하는데 미흡함이 있었고 나이의 경우 눈 주변과 옷 등을 중점으로 판별한다는 것을 알았다.

이런 결과를 바탕으로 해당 부분에 Cutmix를 실험하여 더 높은 정확도가 나오는지 확인하였다. 논문의 코드를 팀의 베이스라인 코드에 맞게 수정하여 적용하고 실험을 진행하였으며 cutmix가 어떤 성능을 보이는지 확인하였다

나는 어떤 방식으로 모델을 개선했는가?

GradCam의 결과를 바탕으로 눈 주위에 박스를 생성하여 Cutmix를 진행하려 하였다. 주어진 데이터셋 사진들은 인물이 중심에 있고 대부분의 인물이 중심점을 기준으로 상단에 눈이 있었기 때문에 이미지를 상하로 나누어 합성하였다. 성능이 예상보다 낮게 나오는 것을 확인하여 바운딩 박스를 무작위로 변경한 후 다시 실험을 진행하였고 더 나은 성능을 확인할 수 있었다.

내가 한 행동의 결과로 어떤 지점을 달성하고, 어떠한 깨달음을 얻었는가?

gradcam을 통해 확인한 부분을 바꾸는 것보다 무작위로 박스를 생성하여 바꾸는 것이 더 큰 효과를 발휘하였다. 이에 더해 이미지의 일부를 바꾸는 cutmix보다 이미지 전체를 합성하는 mixup이 더 높은 성능을 내는 것을 확인할 수 있었다. 이를 통해 모델의 학습 중요도와 성능을 올리는 방법을 확인할 수 있었다.

전과 비교해서, 내가 새롭게 시도한 변화는 무엇이고, 어떤 효과가 있었는가?

기존에는 코드를 내가 읽기 편하게 작성하였지만 이번에는 베이스라인 코드에 맞춰 작성하였다. 그러나 주석과 구조면에서 한계가 있었고 이를 팀원이 수정해주었다. 수정된 코드를 읽으면서 주석과 구조, 타입 등 타인과 코드를 공유할 때 주의해야 할 것에 관한 많은 지식을 얻을 수 있었다. 또한 코드를 공유하고 marge하는 과정에서 발생한 문제를 해결하면서 github에 대한 이해를 조금 더 높일 수 있었다.

마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?

Cutmix의 바운딩박스를 사람 얼굴에 맞게 잘라서 실험하지 못하였다. yolo등의 객체 탐지 알고리즘을 추가하여 구현하는 방법이 있었으나 시간내에 시도하지 못하였다. 또한 모델을

선택하고 튜닝하는 과정에서 부족함을 많이 느꼈다. 사용할 기법을 찾는 과정에서도 많은 도움을 받았으며 베이스가 부족하다는 것을 체감하였다. Github의 사용에서도 부족함을 많이 느꼈는데 main branch에 적극적으로 marge하지 못하고 내가 작업하는 branch에서 대부분의 작업을 수행하였다. 프로젝트에 미칠 영향 때문에 적극적으로 main branch를 활용하지 못한 것이 아쉽다. git관리를 Gui에 의존하여 수행한 점도 아쉬웠다. 명령어를 더 적극적으로 사용해야 하지만 편의성과 가시성 때문에 gui에 과의존하는 경향이 있었던 것 같다.

한계/교훈을 바탕으로 다음 프로젝트에서 스스로 새롭게 시도해볼 것은 무엇일까?

이번 프로젝트에서 부족함을 느꼈던 기초지식 부분에 대한 공부를 늘리고 논문 코드와 베이스 라인 코드를 더 많이 확인하면서 코드 구조와 작성에 관한 숙련도를 올려 나갈 것이다. github 또한 gui에 의존하지 않고 명령어만으로 대부분의 과정을 처리할 수 있게 더 많이 연습해야 한다.

송지민

처음 경험하는 AI 프로젝트로서, Classification Task의 전체적인 진행 과정을 이해하고, 기술적인 스킬 향상을 목표로 했습니다. 프로젝트를 진행함으로써 EDA, 데이터 전처리, 모델 선택, 훈련 및 평가의 전반적인 흐름을 배우고 특히, 데이터의 특징에 따라 모델이 어떻게 예측을 수행하는지에 대한 개념을 알기 위해 노력했습니다.

먼저, 제공된 Baseline code를 통해 전체 프로젝트 구조를 이해하고 필요한 도구와 라이브러리를 익혔습니다. 다양한 모델, 손실 함수, 옵티마이저를 클래스로 정의하고, 변경해면서 실험했습니다.

Data Augmentation에서 Albumentation 모듈의 픽셀 수준 변환 함수들을 모아 대략 40가지 기법을 한눈에 볼 수 있는 코드를 작성하고, 이를 바탕으로 사용할만한 함수를 찾았습니다. 특히, 마스크 착용 여부나 성별같은 경우는 잘 맞추지만 나이를 잘 맞추지 못해서 선명도 또는 주름 등을 부각시킬 수 있는 기법을 우선으로 생각했습니다. 그래서 찾은 CLAHE, Equalize, Emboss 등의 기법을 테스트 했습니다.

GradCAM을 봤을 때 모델이 사람만을 봐야하는데 배경도 같이 보는 경우가 있어서 배경을 제거하고 분류했을 때 결과가 궁금했습니다. U-Net segmentation을 이용해 Train과 Test 데이터셋의 배경을 검게 마스킹 처리하고, 그 이미지를 EfficientNet으로 학습한 후 결과를 봤을 때 의미있는 정확도 향상의 결과를 얻었습니다.

여러 Augmentation기법들을 적용하면서 반복적으로 테스트 했을 때 항상 성능 향상에 기여하지 않는다고 생각했습니다. 그래서 확률적으로 이미지에 변화를 주는 대신 항상 모든 이미지에 변화를 주어 선명도 또는 주름 등을 부각하여 테스트 했습니다. 이 경우에 아무런 효과를 주지 않았을 때보다 성능이 크게 좋아지지 않고 낮아지는 경우도 있는 등 편차가 커서 실패로 이어졌습니다. 이를 통해 더 깊은 모델에 대한 이해가 필요함을 느꼈고, 간결한 모델을 선택했으면 어땠을까

하는 아쉬움이 남았습니다.

Github를 통한 팀 협업을 사용하면서 시행착오를 겪어볼 수 있는 기회였는데 적극적으로 사용해 보지 못한 것이 아쉽습니다. 이번의 경험을 바탕으로 자연스럽게 사용할 수 있도록 공부해서 다음 프로젝트 때 제대로 활용하고 싶습니다.

오왕택

나는 내 학습목표 달성을 위해 무엇을 어떻게 했는가?

- 우리 팀의 학습 목표는 무턱대고 사용하기 전에 무엇인지 이해하고 사용하며 대회를 경험해보지 않았던 사람이 많았으므로 1등을 목표로 하기보다, 많은 지식들을 얻고 공유하고자 참여하였습니다.
- 대회가 시작하기 전, image classification 모델들을 세미나를 통해 각자 공부해서 발표해 보는 시간을 가졌습니다.
- 데이터 불균형 문제를 EDA와 데이터를 직접 뜯어보며 확인했고, 이 문제를 해결하는데 많은 실험 가설을 세웠습니다.
- 기존의 Baseline을 참고하지 않고 최대한 새롭고 실험하기 편한 Baseline 코드를 짜는데 집중했습니다.
- 팀원들이 편하게 사용할 수 있도록 Wandb 팀 space를 만들었고 config 파일을 만들어 사용했습니다.
- 실험 목표 중에 저는 Mixup과 YOLOv8을 활용해 데이터를 전처리하는데 주도적으로 실험하였습니다.

나는 어떤 방식으로 모델을 개선했는가?

- 데이터의 사람들은 모두 정형화 되어있어 mixup을 사용해서 사진 안에 사람들끼리 겹치게 한다면, 일반화나 성능 부분에서 좋을까 라는 가정을 가지고 코드를 만들어서 실험하였습니다.
- YOLOv8 탐지모형을 사용하여 사람을 추론한 사진들을 전처리로 활용하여 실험에 Resize하여 사용했습니다.
- YOLOv8로 탐지한 후, U-Net으로 분할까지 하여 이미지에 최대한 사람에 특성만 들어가게 만들어 데이터 전처리에 활용했습니다.

내가 한 행동의 결과로 어떤 지점을 달성하고, 어떤 깨달음을 얻었는가?

- Baseline 코드를 만드는데 오래 걸려서 팀원들과 코드를 늦게 돌릴 수 있어 아쉬웠습니다.
- Mixup은 어떤 모델에 넣든 간에, 일반화나 성능에 도움이 된다는 것을 알게 되었습니다. 이는 이미지가 정형화 되어있어서 Cutmix보다 높은 성능을 보인다는 것을 알 수 있었습니다.

- U-Net 을 활용한 분할보다는 YOLOv8 을 사용하여 탐지한 것이 더 성능이 좋았습니다. 이는 U-Net 이 이미지 배경을 단순히 모두 검정색으로 만들어서 그런지 정확한 이유는 모르겠지만 탐지가 더 성능이 좋았던 것을 알 수 있었습니다.
- YOLOv8 로 탐지한 후 U-Net 으로 분할한 이미지를 활용한 것은 오히려 성능에 악영향을 끼쳤습니다. 실험 가정 상, 가장 좋을 것으로 예상했지만, 결과가 좋지 않아 아쉬웠습니다.

마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?

- 대회 후반부로 갈수록 점수에 쫓겨 설정했던 목표보다는 단순 실험 반복을 실행한 것 같습니다.
- Github 사용에서의 부족함을 많이 느꼈고 Merge 를 적극적으로 하지 못하고 Branch 도 하나의 Branch 에서 진행한 것도 아쉬웠습니다.
- 하이퍼파라미터 튜닝도 많이 안하고 쓸 수 있는 기능을 많이 못써 아쉬웠습니다. 시간이 남았다면 할 수 있었던 실험들도 많았을 것 같습니다.

한계/교훈을 바탕으로 다음 프로젝트에서 시도해볼 것은 무엇인가?

- Git-Flow를 활용하고 Git convention을 정하고 잘 짜여진 체계에서 프로젝트를 진행하고 싶습니다.
- Git issue 같은 것들을 활용하고 Wandb Sweep 같은 기능을 잘 활용하고 싶습니다.
- Baseline 코드를 빠르게 작성하는 방법을 배우고 이를 실험용으로 빠르게 만드는 것을 공부할 것입니다.

이동호

1. 잘한 점

이번 프로젝트 시간도 긴 편은 아니었지만, 석사 시절에도 프로젝트의 기간이 긴 편이 아니었다. 석사 과정에서는 학습보다는 성과 또는 성능이 더 중요한 문제였다. 그래서 기존의 모형을 이해하지 못하고, 해당 작업이 어떠한 작업인지를 완벽하게 이해를 하지 못한 채 진행을 하였다. 그래서 단순히 성능이 좋다는 모형을 바꿔서 실험해보는 것에 그쳤다. 하지만 이번 프로젝트에서는 멘토님과 팀원들의 도움으로 급하지 않고 천천히 데이터를 살펴보는 연습부터 해보았고, 그 결과 단순히 모형을 바뀌는 것이 아닌 작업에 대한 명확한 이해를 할 수 있었던 것 같다. 예를 들어, 데이터를 살펴보면서 모형이 어느 부분에 집중을 하고 있는지, 또는 나이에 대해서는 왜 잘 맞추지 못하는지를 고민해보면서 딥러닝 모델링에 있어서 하나의 논리적인 사고를 기를 수 있었다.

그리고 그 동안은 협업 툴을 사용하기보다는 혼자서 고민을 하고 이를 해결하는 과정이 많았던 것에 비해 git, wandb 같은 협업 툴도 사용하면서 팀 차원에서의 프로젝트를 하는 방법에 대해서 익힐 수 있었다. 아직까지는 해당 협업 툴에 서툴르지만 앞으로의 추가적인 프로젝트를 통해서 이를 보완해나갈 수 있을 것이라고 기대가 된다.

그동안 배운 지식들을 통해 팀원들에게 해야할 일을 정해줄 수 있어서 좋았던 것 같다. 그리고 팀원들 역시 해당 부분에 대해 잘 따라주어서 이번 프로젝트가 성공적으로 마무리가 되었던 것

같다.

2. 아쉬운 점 및 개선 방안

초반에 시간 분배를 잘 하지 못한 것이 아쉬웠다. 우리 팀의 목표는 성능에 구애받지 않고, 모든 인원들이 다들 얻어가는 것이 있기를 바랬다. 다만, 해당 과정에서 계획에 대한 실수가 있었고 팀적인 차원에서 프로젝트를 뒤늦게 시작하게 됐다. 우리 팀은 프로젝트 경험이 있는 사람도 있었지만, 그렇지 않은 사람이 더 많았다. 그래서 나는 그들에게 스스로 한 번 베이스라인 코드를 작성하여서 모형을 구성해볼 기회를 주었지만, 이는 생각보다 많은 시간이 소요되었다. 그래서 이는 추후에 모형 및 손실 함수 부분에서의 실험 시간이 부족해지는 결과를 초래한 것 같다. 이 부분에 있어서 팀적으로 이야기를 해서 베이스 라인 코드 하나로 처음부터 쪽 진행을 했었어야 했다. 또한, 여기서 나 역시 팀원이 작성해준 베이스라인 코드를 사용하기보다도 시간 상의 문제로 내가 작성한 코드에서 따로 작업을 한 경우가 많았어서 이는 팀 프로젝트 부분에서 매우 큰 아쉬움으로 남는다.

다음 프로젝트부터는 이번 프로젝트보다 어려운 내용들이 많은 것으로 알고 있다. 미리 베이스라인 코드를 작성하고, 해당 개념들에 대해 팀원들과 공부를 하여서 프로젝트가 시작을 할 경우, 바로 다양한 실험들을 할 수 있는 환경을 만들 수 있도록 노력할 것이다.

지현동

- 나는 내 학습목표 달성을 위해 무엇을 어떻게 했는가?

데이터 불균형 해소를 위한 여러가지 argumentation 기법들과 여러 손실 함수와 최적화 기법들 그리고 모델을 직접 구현하거나 코드를 뜯어보면서 공부하고 공유하는 것을 팀 목표로 삼고 대회에 임했습니다.

저는 대회에서 제공해 준 argparse 을 사용한 베이스 코드와 팀 내 yaml 파일을 활용한 베이스 코드를 한 줄 한 줄 뜯어보면서 다양한 기법을 직접 구현해보는 것을 개인 목표로 삼았고, 공동의 목표를 위해 mixup을 적용한 코드를 분석하고 시각화해서 정리하여 팀에 공유하였습니다. 또 대회 submission 횟수 제한의 불편함을 개선하고자, 팀 내 최고 성적의 submission 과 비교해서 점수를 예상해볼 수 있는 계산기를 만들었습니다. 해당 계산기는 +-0.01 의 오차를 가졌고, 팀에 공유하여 실험이 원활히 진행 수 있도록 도왔습니다.

•나는 어떤 방식으로 모델을 개선했는가?

팀원들과 EDA 와 데이터를 직접 뜯어보면서 데이터의 불균형을 확인한 뒤에, 저는 크게 세 가지를 시도해봤습니다. 첫 번째로 stratified k-fold 을 시도했고, 두 번째로 mixup 적용하면서 약간의 성능 개선을 확인했습니다. 세 번째로 여러 손실 함수를 실험했고, 클래스 불균형을

해결하기 위해 등장한 오분류되기 쉬운 케이스에 대해 더 큰 가중치를 주는 focal loss 와 label smoothing 을 사용해서 성능을 개선했습니다.

●내가 한 행동의 결과로 어떤 지점을 달성하고, 어떤 깨달음을 얻었는가?

Stratified k-fold 는 학습에 오래 걸렸고, 괄목할 만한 성능 개선이 이루어지지 않았기 때문에 이후로는 사용하지 않았고, 데이터 불균형을 해결하기 위해 처음 시도하는 기법으로, stratified k-fold 를 선택하는 것은 효율적이지 못하다는 것을 깨달았습니다.

mixup 은 적용 후 확연한 성능 개선을 확인할 수 있었는데, 코드를 한 줄 한 줄 뜯어보면서 배치 사이즈의 tensor 가 어떻게 섞이는 지 시각화해서 확인할 수 있었고, label smoothing 과 비슷한 효과임을 깨달을 수 있었습니다.

●전과 비교하여 새롭게 시도한 변화는 무엇이고, 어떤 효과가 있었는가?

이전까지 torch vision 을 사용하다가, Timm 라이브러리를 처음 사용해봤는데, 사용이 편해서 시간을 아낄 수 있었고, 또한 모델이 다양하게 많아서 여러가지 모델을 실험해볼 수 있었다. 또한 tmux tool 을 배워서 활용했고, 실험 환경을 개선할 수 있었습니다.

●마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?

성능 향상을 위해 mobile vit 와 EfficientNet 을 활용했는데, 모델에 대한 이해도가 낮아서, 이론적인 접근이 아닌 마구잡이식으로 훈련시켰던 것 같아서 아쉽다. 또한 성능향상과 코드 분석에 많은 시간을 할애하면서, git 을 많이 활용해보지 못한 것이 아쉽습니다.

●한계/교훈을 바탕으로 다음 프로젝트에서 시도해볼 것은 무엇인가?

모델에 대한 베이스를 충분히 공부하고, 공동의 목표에 맞게 개인 목표를 조금 더 체계적으로 세워서 시간관리를 할 것 입니다. 또한 Git issue와 rebase도 시도해볼 것 입니다.