<기술적인 도전>

* 점수 및 순위

LB : 0.7585 52등

LB 최종 : 0.7491 44등

* 검증(Validation) 전략

모든 training set을 병합한 후, KFold를 사용하여 5 fold로 나누었습니다.

* 사용한 모델 아키텍처 및 하이퍼 파라미터
  1. 아키텍쳐 : ResNet50
  2. Augmentation : Centercrop(250), Resize((224, 224)), RandomHorizontalFlip(0.5), ToTensor, Normalize(mean=(0.5,), std=(0.5,))
  3. Image size : 224 x 224
  4. Optimizer : Adam
  5. Scheduler : stepLR을 사용했으나, 초반부터 낮은 lr을 사용했더니 lr이 변하진 않았습니다.
  6. Learning rate : 1e-5
  7. Weight decay : 0.01
  8. Batch size : 32
  9. Loss function : CrossEntropy
* 앙상블 방법

시도했던 대부분의 방법들의 submission.csv를 분석해본 결과, accuracy 및 f1 score에서 차이가 크지 않고, 실제 output의 결과들고 크게 다르지 않아, 앙상블을 하기 위해 필요한 cost보다 기대할 수 있는 퍼포먼스 향상이 너무 적을 것 같아 쓰지 않았습니다.

* 시도했으나 잘 되지 않은 것들

많은 사람들이 썼던 Efficientnet을 사용해 보았으나, 저에게 한해서는 성능이 오히려 하락해였습니다. 그래서 과감히 EfficientNet은 포기하고, Resnet을 계속 사용했습니다. 피어세션을 통해 일반적인 crossentropy가 아닌 다른 custom function을 사용하면 더 좋은 성능을 얻은 사람이 있다고 들어 focal loss 같은 다른 loss function을 사용해 보았으나, 역시 성능 향상을 얻지는 못했습니다. 데이터의 수가 많이 부족한 것 같아 다양한 augmentation 방법들을 찾아보고 적용해 보았지만, 성능이 오히려 떨어지는 경우가 많았습니다.

<학습 과정에서의 교훈>

이러한 Kaggle 형식의 경쟁은 처음 경험해보는 소중한 시간이였습니다. Pytorch를 사용해 많은 도움을 받기는 했지만, 처음부터 끝까지 프로젝트를 완성한 것도 처음이였습니다. 그 과정에서 수업만을 들을 때에는 알 수 없었던 여러가지 시행착오들을 겪었습니다. 이러한 문제들을 해결하기 위해 여러가지 시도를 해보았습니다. 수업 시간에 배운 것들도 몇 가지 시도해 봤는데 그런 것들 이 항상 통하지는 않는다는 것도 배웠습니다. 그래도 그 중에 하나는 먹힌다는 것 역시 여러 시행착오를 통해 배울 수 있었습니다.

<마주한 한계와 도전숙제>

제가 될 것이라고 생각했던 아이디어를 실제로 적용하면 안 되는 경우가 상당히 많았습니다. 실패를 받아들이고 다른 새로운 무언가를 다시 시도하는 과정이 힘들고 버거웠습니다. 이러한 저의 한계를 극복하고 끊임없이 새로운 아이디어를 생각하는 능력과 그 아이디어를 계속해서 실행하고 실험하는 끈기를 얻는 것이 다음 스테이지에서 저에게 필요한 도전과제라 생각합니다.