**Semantic Segmentation**

**점수 및 순위**

최종 리더보드 : 5위 - mIoU 0.6798

**환경**

학습과 추론을 위한 파이프라인은 파이썬 py 파일로 구현했고, 일회성 실험이나 데이터를 전처리하는 등의 일회성 실험은 노트북 파일로 구현했습니다.

P Stage 2에서 huggingface의 Trainer를 사용해보고 감명 받아서 이번에 semantic segmentation 학습 시 비슷하게 Trainer 클래스를 만들어서 사용했으며, Trainer 클래스 내에서도 epoch 함수, step 함수 등 세분화해서 각 단계별로 분리하여 사용하도록 구현했습니다. 이렇게 클래스와 클래스 내에서 각 기능별로 분리하여 구현하고 나니 학습시 특정 단계에서 코드 수정을 해야할 때 편리 해서 굉장히 만족스러웠던 시도였다고 생각합니다. 다른건 제쳐두더라도 hugginface의 Trainer처럼 비슷하게나마 구현해봤다는 경험으로 이번 컴피티션에서 좋은 수확을 걷지 않았나 생각합니다.

두번째로는 기존에는 매번 실험시 세팅 값을 하나하나 바꿀 때 해당 부분의 코드 라인으로 가서 하나씩 수정해야 했었는데, 이렇게 실험할 경우 기존과 비교해서 어떤 값이 변경되었는지 한눈에 확인하기 힘들고 바꿀때마다 해당 라인으로 가서 수정해야하기 때문에 매우 비효율적이었습니다. 그래서 이런 하이퍼 파라미터나 변경할 수 있는 값들은 따로 json 파일로 분리하여 해당 부분을 정의해놓고 학습 시 해당 파일을 읽어와서 사용하는 형태로 구현했습니다. 이렇게 사용했을 때 장점은 먼저 파일 하나로 실험시 세팅 값들을 한눈에 볼 수 있다는 장점과 해당 실험 당시에 어떤 세팅 값을 사용했는지 확인하기 위해 별다른 작업 없이 해당 json 파일만 따로 복사하면 된다는 장점이 있습니다.

개인적으로 위의 두 가지가 이번 semantic segmentation competition을 진행하면서 얻었던 가장 큰 수확(?)이라고 생각합니다. segmentation에 관한 이론이나 점수 향상 방법 등 많은 수확이 있었지만, 이런 경험만을 통해 얻을 수 있었던 발견이 가장 크지 않았을까 합니다.

추가적으로 실험 관리와 실험 결과 시각화 등을 위한 tool로는 neptune ai라는 라이브러리를 사용해서 관리했고, 기존에 제공된 model, loss function, optimizer 외에 따로 정의해서 사용하거나 설치해서 사용한 것들은 build.py와 builder 폴더를 사용하여 build를 import 해서 사용하는 방식으로 구현했습니다.

**회고**

일단 가장 아쉬운 점은 semantic segmentation 정리를 조금 빨리 해둘 걸이란 생각이 들었습니다... object detection 끝나고 같이 하려니까 기억도 잘 안나고 해서 아쉽습니다.

P Stage 3부터는 팀 단위로 컴피티션이 진행되서 팀 단위로 피어세션이 이뤄지고 정보도 같이 공유한다는 점이 좋았습니다. 개인적으로 팀원 분들도 굉장히 좋아서 재밌게 진행했던 것 같습니다. 특히 혼자서 진행하다보면 시간 상의 문제로 실험하지 못하는 것들이 많았는데 팀 단위로 분배해서 실험도 하고 각자 아이디어를 나눈다는 점에서 팀 단위 컴피티션의 장점이 가장 부각되는 것 같습니다.

컴피티션을 진행하면서 조금 아쉬웠던 점은 아무래도 시간 분배가 아니었을까 합니다. 2주 조금 안되는 시간 동안 적절하게 계획을 짜고 시간 분배를 잘 했어야 하는데 너무 모델을 선택하는 부분에 있어서 시간을 많이 써서 마지막에 점수를 끌어올리기 위한 기법을 적용하는데 상대적으로 시간이 부족했던 것이 아쉬웠습니다.

또 object detection 때도 마찬가지이지만 새로운 모델이나 새로운 기법을 적용해볼 때, 제대로 이해하고 쓰는게 아니라 먼저 써보고 성능 향상이 있을 때 뒤늦게 찾아본 점(?)을 고쳐야할 점이라 생각했습니다. 어떻게 보면 2주라는 시간이 짧다보니 제대로 이해하고 적용하는게 시간상 여건이 안될 때가 많지만 부스트캠프에서 자신의 스킬을 끌어올리기 위해 참여한다는 관점에서 점수에 너무 몰두하는 것보다 하나를 하더라도 확실히 하고 가는게 어땠을까 하는 생각이 들었습니다. 컴피티션을 하면서 어떤 모델이 semantic segmentation에서 성능이 좋고 어떤 기법을 사용하는게 성능 올리는데 좋다 이런 것들은 알면서도 그거 어떤 방식으로 성능을 향상시키는지에 대한 이해는 부족했던 것이 아쉬웠습니다.

이 부분에 대해서는 object detection 때 멘토님도 이야기하셨던 점이서 가장 피드백되야할 점이라고 생각됩니다.

그래도 부스트캠프에서 활동이나 이전에 했던 활동들이 대부분 개인으로 진행했던 것들이어서 팀 단위 협업 경험이 부족하단 생각을 가지고 있었는데 이번 기회에 협업이 어떤 것인지 조금은 알 수 있던 좋은 경험이라고 생각합니다.

노션으로 매 피어세션 때 나눴던 이야기를 정리하고, 각자 아이디어를 적고 구글 시트에 각자의 실험 결과를 적고 하는 등의 활동이 도움이 되었다고 생각했습니다. 제대로 된 협업 경험은 아닐 수 있지만 좋은 경험이었다고 생각했습니다!

**Object Detection**

**점수 및 순위**

최종 리더보드 : 9위 - mAP 50 0.4689

**환경**

object detection 때에는 기본 제공된 baseline 노트북 파일이 사용하기 편리하게 되어 있어서 따로 py 파일로 구성하지 않고 필요한 부분만 맞춰서 수정하는 방식으로 사용하였습니다.

mmDetection 깃헙 레포지토리나 공식 문서를 참고하면서 wandb 연동이나 best mAP 50 모델을 저장하는 기능을 config 파일 등에 추가해서 사용하면서, UniverseNet이나 Swin 같이 mmDetection으로 구현이 되었지만 기본 mmDetection에는 포함되어 있지 않는 모델들은 따로 파일을 받아 기존 mmDetection에 추가하거나 맞게 수정하는 등의 과정을 거쳐 기본 베이스라인 코드를 최대한 유지하면서 사용했습니다.

semantic segmentation 때와 비슷하게 피어세션에서 이야기한 내용은 간략하게 노션에 정리했고, 각자 맡은 역할이나 실험 등은 노션에 표로 정리하거나 실험 결과 등은 엑셀 시트에 기록하여 각자 어떤 실험을 했거나 결과가 어떤지 한눈에 보기 편하게 하고자 했습니다.

저 같은 경우는 mmDetection 사용이 처음이여서 mmDetection에 익숙해지는데 많은 시간을 썼습니다. 그러면서 얻은 유익한 정보나 사용하면 좋을 것 같은 기능들(config 파일 값을 한번에 보거나 best 모델만 저장하기)은 노션에 정리해서 올렸습니다.

**시도**

저희 팀은 semantic segmentation 때 시간 분배에 실패하여 마지막에 조금 아쉬운 부분이 있었는데, 이를 교훈삼아 이번에는 처음부터 계획을 잘 짜서 진행하고자 했습니다. 먼저 첫주는 상현님들 제외하고 다들 mmDetection이 처음이여서 각자 익숙해지면서 사용해보고 싶은 모델이나 실험을 하면서 결과를 공유하는 시간을 가졌습니다. 첫주 주말까지 각자 mmDetection에 익숙해지고, 앙상블에 사용할 모델을 하나씩 맡아 남은 마감 기간 동안 모델의 성능을 최대한 끌어올려 앙상블하는 방향으로 계획을 짰습니다.

저 같은 경우에는 다양한 모델을 사용해보고 싶어서 papers with code나 mmDetection 공식 문서에 benchmark 기록을 참고하면서 coco dataset의 object detection에서 좋은 성능을 보이는 모델을 가지고 테스트하는 방향으로 첫 주 동안 실험을 진행했습니다. 사실 이 때도 semantic segmentation 때와 똑같이 모델의 논문은 커녕 다른 모델과 어떤 점이 다른지도 모른채 그냥 적용해보는 식으로 실험을 진행했는데 멘토님과 시간을 갖고나서 swin 논문을 읽어보면서 리더보드에 목 맬 것이 아니라 성장하는게 중요하구나라는 생각이 들었습니다.

이렇게 첫 주 동안 각자 어느정도 실험을 끝내고 실험 결과와 느낀 점 등을 공유하면서 점수를 올리기 위해 어떤 방향으로 어떤 실험을 하면 좋을지 고민했습니다. 두번째 주는 각자 맡은 모델을 가지고 성능을 끌어올린 뒤에 마지막에 앙상블하는 식으로 진행되었는데, 저는 먼저 단일 모델로 가장 높은 성능을 보여준 UniverseNet으로 다양한 이미지 스케일을 적용해보면서, TTA와 InstaBoost 등을 더해가면서 성능 변화를 보면서 가장 높은 성능을 보이는 실험 환경에 마지막으로 모든 데이터(학습 + 검증)를 가지고 학습한 모델을 앙상블하는 모델로 사용했습니다.

**회고**

가장 먼저 드는 생각은... Public 리더보드 2위에서 최종 리더보드 9위로 떨어진 점 입니다... 컴피티션이 마감되고 그날 23시 조금 넘어서 최종 리더보드가 공개되었는데 팀원들 모두 확인하고 우울해져서 다들 망연자실 했었습니다... 다음날 피어세션 때 왜 이런 결과가 나왔을까 이야기를 나눴는데 아무래도 각자 모델 성능 끌어올린 뒤에 weighted boxes fusion을 사용해서 앙상블 했는데 이 때 제출한 모델의 public 리더보드 순위를 기준으로 가중치를 줘서 public 리더보드에 overfitting 된게 아닐까 하는 의견이 있었습니다. 가중치를 안줬을 때도 나쁜 점수가 아니였는데 2위에 욕심을 내려다가 최종 리더보드에서 추락하게 된 점이 가장 아쉬웠습니다.

그래도 2주 동안 나름대로 계획을 잡고 그 계획대로 실행했다는 점에서 팀 단위 컴피티션에서 좋은 경험을 했다고 생각합니다. 특히 어떤 벽에 막혔을 때 각자 나름대로 어떻게 해결할지 생각해보면서 고민하고 의견을 나누는 점에서 저는 팀 단위 컴피티션에서 이런 장점이 정말 좋다고 느꼈습니다. P Stage 4 때는 이것보다 더 성장한 팀이 될 수 있도록 고민하고 노력해보겠습니다~

이번 Object Detection 때는 조금 색다른 경험도 했는데, 이번 컴피티션 때 서버의 용량이 40기가 중 여유 공간이 20기가 밖에 되지 않아서 매번 실험할 때마다 결과를 확인하고 모델 파일을 지워줘야 하는 불편함이 있었는데, mmDetection에 익숙해지는 과정 중에 열심히 찾아보면서 best mAP 50 모델만 저장하는 방법을 찾게 되어서 토론 게시판에 글을 작성해서 공유 했었는데 많은 캠퍼분들에게 하트를 받아서 토론왕 발표를 하게되는 경험을 하게 되었습니다. 사실 이런 발표 경험도 적고 많은 사람들 앞에서 발표한 경험이 적어서 처음에 정말 걱정도 되고 떨렸는데, 전 날부터 혼자 리허설하면서 녹화한 것도 돌려보면서 최대한 열심히 좋은 모습 보이려고 했는데 당일날 그래도 만족스런 발표였다고 느껴서 좋았습니다. 또 팀원분들이 열심히 호응도 해주셔서 긴장도 많이 풀리고 정말 감사한 마음 뿐입니다. 사실 이런 발표 기회도 앞으로도 많이 있을텐데 마음 속으로는 이런 연습도 해야한다고 느끼면서도 이런 기회가 있을 때마다 무시하고 지나쳤는데 이렇게 또 한번 경험을 해보면서 한 발자국 성장한 기분이 들어서 뿌듯했습니다.

사실 P Stage 때 프로젝트를 기대하고 왔었는데 컴피티션 방식으로 진행한다고 해서 실망 많이 했었는데 나름대로 컴피티션 장점도 있고 그 장점을 잘 살려서 좋은 경험을 하는 것 같아서 내심 만족스런 부분이 많았습니다. 부스트캠프에 취업을 목적으로 참여하고 있기에 이런 컴피티션을 진행하면서도 개인적으로 취업에 도움이 될만한 것들을 스스로 잘 캐치해서 기록해야겠다는 생각도 들었습니다. 그래서 이번 P Stage 4 때는 의견이 맞는 팀원분들과 함께 OCR 컴피티션을 진행하면서 데모 페이지 같은 프로젝트도 병행해보려고 계획하고 있습니다.

이번 4주 동안 정말 열심히 하면서 높은 등수를 기록해보려고 잠도 줄여가면서 재밌게 했었는데 결과는 조금 아쉽지만 ㅎㅎ 좋은 팀원분들과 정말 좋은 경험을 해서 후회 없는 컴피티션이였습니다. 뭔가 이것저것 해본 건 많았다고 느꼈는데 막상 적으려고 보니 적은 것 같았는데 이번 P Stage 4 때에는 몰아서 적지 말고 틈틈히 블로그 같은 곳에 기록해야겠습니다. 남은 P Stage 4도 정말 재밌게 좋은 경험 했으면 좋겠습니다!