

Project 3 Wrap-up Report

level2 Data Centric
CV03 뻔뻔(Fun Fun)

1. 프로젝트 개요

1.1 프로젝트 주제

학습 데이터 추가 및 수정을 통한 이미지 속 글자 검출 성능 개선 대회

1.2 프로젝트 상세

- Dataset: UFO format, 진료비 영수증 200장 (train 100, test 100)
- Model: EAST
- Evaluation: DetEval
- Computing Environment: GPU V100
- Cooperation tool: Github, Notion, Slack, Wandb
- 베이스라인 코드 중 model.py, loss.py, east_dataset.py, detect.py는 변경사항 없이 그대로 사용해야 함

2. 프로젝트 팀 구성 및 역할

강현서	외부 데이터 수집 및 학습, Fine-Tuning, Optimizer 실험
김정택	CI(linter, formatter) 구현, Augmentation 실험
박진영	Metric 출력, Fine-Tuning
선경은	EDA, Augmentation 구현
이선우	외부 데이터 수집 및 학습, 모델 Fine-Tuning, Ensemble
최현우	Validation 구현, Wandb 연동, Augmentation 실험

3. 프로젝트 수행 절차

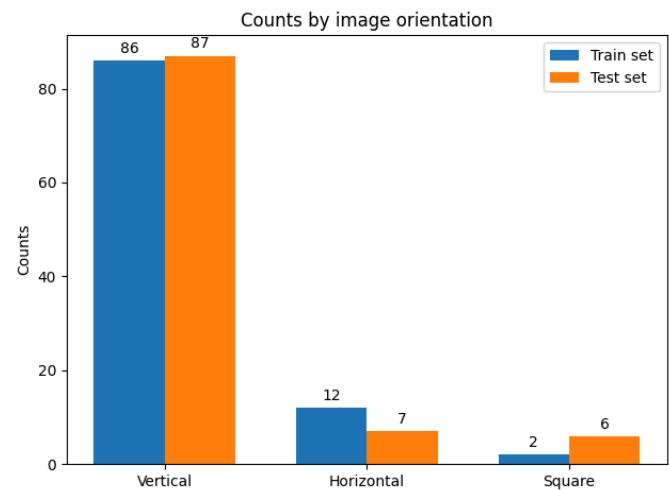
3.1 프로젝트 일정



4. 프로젝트 수행 방법

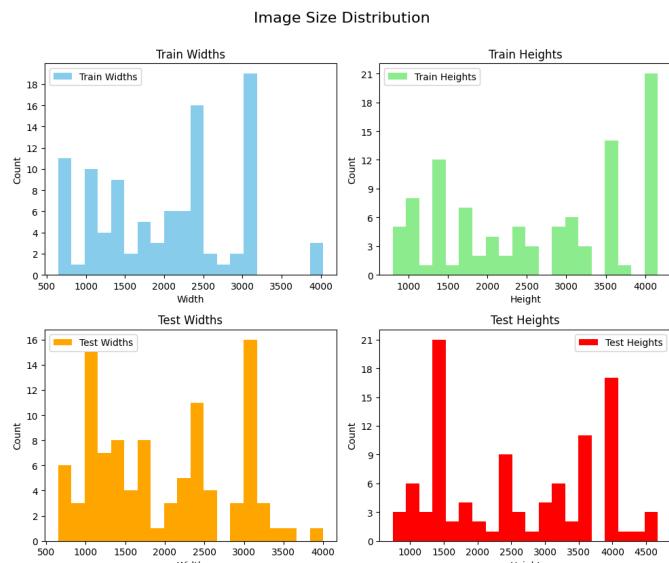
4.1 EDA

각 데이터셋의 이미지 방향 분포



- 세로 방향 이미지 86% 이상

각 데이터셋의 이미지 가로, 세로 사이즈 분포



Train set 이미지 열어보기

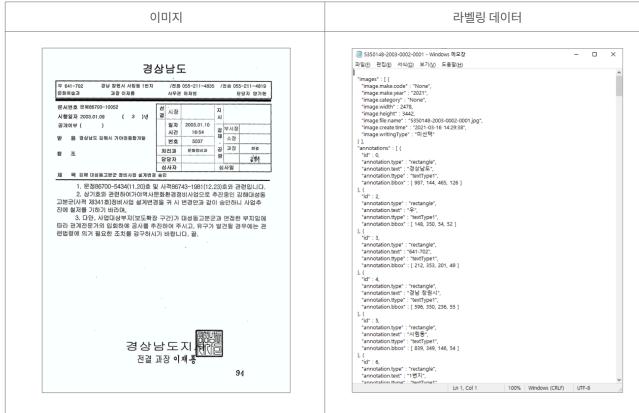
- 선명한 이미지 : 94장 | 접힌 자국 이미지 : 4장 | 흐릿한 이미지 : 2장

Test set 이미지 열어보기

- 선명한 이미지 : 48장 (학습데이터에 비해 어두움) | 노이즈 적음 : 18장 | 노이즈 많음 : 34장

4.2 Pretraining & Fine-tuning

약제비 영수증과 유사한 성격을 가진 OCR 데이터를 추가적으로 이용하고자 함
AI-Hub의 공공행정문서 OCR, OCR 데이터(공공) 데이터셋 총 두 개를 이용하여 EAST 모델을 학습시킨 후, 어느정도 성능이 나온 모델에 대해 Fine-Tuning을 진행함



공공행정문서 OCR 데이터 예시

Dataset	데이터 양	f1 score	recall	precision
공공행정문서 OCR	860장	0.6070	0.4843	0.8132
OCR 데이터(공공)	1000장	0.4253	0.3141	0.6581

(LB score, 150 epoch)

추가 데이터의 경우 데이터 종류보다 데이터 양이 많은 것이 더 좋은 성능을 낼 것이라고 생각했는데, 위 결과를 통해 데이터 종류가 더 중요함을 알게 되었다. 결과적으로 성능이 어느 정도 나온 공공행정문서 OCR 데이터를 이용했고, 이 모델로 Fine-Tuning을 진행함

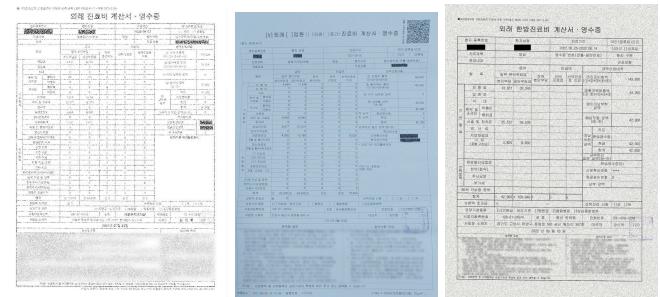
Method	f1 score	recall	precision
Pre-trained Model	0.6070	0.4843	0.8132
Base Model	0.8887	0.8848	0.8926
Fine-tuning Model	0.8968	0.8853	0.9086

(LB score)

4.3 Augmentation

4.3.1 Color Noise

Test 데이터셋과 유사하도록 white, blue, black noise를 추가 및 실험 진행함



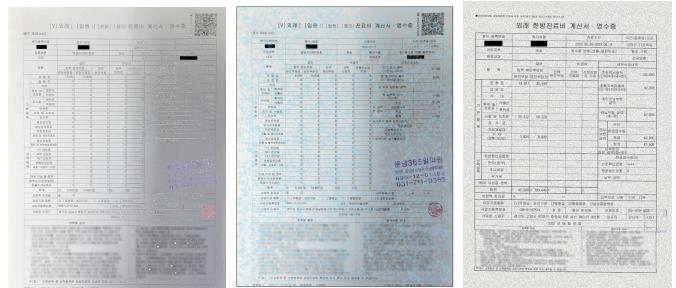
white pixel noise blue pixel noise black pixel noise

Method	f1 score	recall	precision
No Method	0.8887	0.8848	0.8926
Color noise	0.5979	0.4537	0.8765
No Method (Fine-tuning)	0.8968	0.8853	0.9086
Color noise + RandomBrightness (Fine-tuning)	0.9265	0.9299	0.9233

(LB score)

4.3.2 Customize

Test 데이터셋과 유사하도록 Black pixel noise, RandomSnow, Blue polygon noise를 추가 및 실험 진행



RandomSnow blue polygon noise black pixel noise

Method	f1 score	recall	precision
No Method (Fine-tuning)	0.8968	0.8853	0.9086
Customize (Fine-tuning)	0.9308	0.9117	0.9507

(LB score)

4.3.3 Crop Image

모든 모델이 공통적으로 QR Code 좌측에 세로로 적힌 글자를 못잡아내는 것을 확인. 이를 해결하기 위해 Crop Image 만을 가지고 Fine-tuning 진행



Crop Image

Method	f1 score	recall	precision
No Method (Fine-tuning)	0.9308	0.9117	0.9507
Crop Image (Fine-tuning)	0.9080	0.8871	0.9298

(LB score)

4.4 Optimizer

기본 Optimizer인 Adam의 경우 일반화 성능이 떨어지는 경우가 있다고 하여, 일반화에 비교적 더 Robust한 AdamW, SGD를 이용하여 비교함

Optimizer	f1 score	recall	precision
AdamW	0.7919	0.7873	0.7966
Adam	0.8887	0.8848	0.8926
SGD	0.4445	0.3875	0.5210

(LB score, 150 epoch)

가장 좋은 결과를 보인 Adam을 그대로 이용함

4.5 Ensemble

기본 양상을 그대로 사용하면 성능이 더 낮아짐
높은 precision을 보여준 모델들로만 양상을 실행한 결과, 성능 조금 향상
False positive의 가능성 있는 박스들을 지웠을 때 성능 향상

Method	f1 score	recall	precision
Base Ensemble	0.8953	0.9543	0.8432
w/ Higher precision	0.9103	0.9420	0.8807
Delete False Positive Boxes	0.9500	0.9472	0.9528

5. 프로젝트 결과

Dataset	f1 score	recall	precision
Public test set	0.9308	0.9117	0.9507
Private test set	0.9420	0.9170	0.9685