Level-3 P-stage : Data Production

> Wrap-UP Report by Team AI-TEEN(CV18)

1. 프로젝트 개요

1-1. 프로젝트 주제

Optical Character Recognition(OCR)은 이미지 내의 문자를 컴퓨터가 인식할 수 있도록 하는 기술로, 글자 검출(text detection), 글자 인식(text recognition), 정렬기(serializer) 등의 세 모듈로 이루어져 있다. 이 프로젝트는 이 중 text detection 만이 구현된 스켈레톤 모델이 이미 배포된 (deployed) 상태라는 가정하에, 입력 데이터의 수집, 가공, 증강 등을 통해 모델 구조를 수정하지 않은 채로 성능을 향상하게 시키는 방법을 모색한다.

1-2. 활용 장비 및 협업 도구

- AI Stages 서버 내 V100, *Github*, *Notion*, WandB

1-3. 프로젝트 구조

- 학습 데이터 입력 및 출력
  - 입력 : ICDAR17-MLT, ICDAR19-MLT의 이미지, UFO Format annotation 정보
  - 출력 : 글자 영역 bbox
- 사용한 최종 모델 및 데이터셋
  - EAST(Backbone : VGG16), ICDAR2019-MLT 모든 언어의 글자가 있는 이미지

1-4. 기대 효과

우리 팀은 최종 프로젝트의 태스크를 `OCR`을 선정하였다. 이번 데이터 제작 CV 대회에서 좋은 성능을 보인 모델 파일을 최종 프로젝트의 기학습 가중치로 사용할 예정이다.

2. 프로젝트 팀 구성 및 역할

이름	역할
김서기	ICDAR 2017 데이터셋 적용
김승훈	ICDAR 2017, SynthText 데이터셋 적용
배민한	ICDAR 2017 데이터셋 적용
손지아	WandB 설정, ICDAR 2017, 2019 데이터셋 적용
이상은	ICDAR 2017 데이터셋 적용
조익수	ICDAR 2017 데이터셋 적용

3. 프로젝트 수행 절차 및 방법

3-1. 프로젝트 사전 기획 및 문제 정의

- 베이스라인 모델인 EAST 모델 정의된 파일 변경사항 없이 그대로 이용해야 한다.
  - `model.py`, `loss.py`, `east_dataset.py`, `detect.py` 파일은 변경해서는 안 된다.
  - 그 외 파일은 자유롭게 수정할 수 있다.
- cs 파일을 제출하던 기존 대회와는 다르게 서버를 제출하여 제출 서버에서 `inference.py`와 `model checkpoint`를기반으로 채점이 이루어진다.
- 평가 방법은 **DetEval** 방식으로 계산되어 진행된다.

3-2. 프로젝트 수행 절차 및 과정

- 1주차 : 데이터제작 CV 자료 수집 & 베이스라인 코드 실행
- 2주차 : 데이터 추가 & 베이스라인 코드 수정 & Wrap-up report 작성

4. 프로젝트 수행 결과

4-1 .데이터셋 추가 및 Input 크기 변화에 따른 성능 변화

Dataset	언어	수량	inputsize	LB score	비고
SynthText + ICDAR2017-MLT	KOR, ENG	850,000 + 536	512	0.372	**[1]**
ICDAR2017-MLT	KOR, ENG, ETC	536	512	0.527	**[1]**
ICDAR2017-MLT	KOR, ENG, ETC	1,063	512	0.563	**[2]**
ICDAR2017-MLT	KOR, ENG, ETC	1,063	1024	0.594	**[2]**
ICDAR2019-MLT	모든 언어	10,000	512	0.669	**[3]**
ICDAR2019-MLT	모든 언어	10,000	1024	0.640	**[3], [4]**

[1] Default (ICDAR2017-MLT 한글 데이터셋 200 epoch 학습)

- SynthText 의 경우, 시간 문제로 1 epoch 만 학습시키고 extractor 부분만 사용
- ICDAR2017-MLT 의 한글 데이터셋을 이용하여 전체 모델 학습 (extractor 부분도 재학습)

[2] ICDAR2017-MLT의 Valid set 추가, 200 epoch, batch 크기 다름

[3] ICDAR2017-MLT와 ICDAR2019-MLT 데이터셋 동일

[4] 85 epoch 학습

4-2. 최종 선정 모델

1) Models

Key	Value	비고
Text Detector	EAST *(fixed)*	
Backbone	VGG16 *(fixed)*	Pretrained Weight - ImageNet *(fixed)*
Optimizer	Adam	
Learning Rate	1e-4	
LR Scheduler	MultiStepLR	
Loss	EAST Loss *(fixed)*	
Batch Size	32	
Epochs	200	
Input size	512	

4-3. 대회 결과

	Public LB	Private LB
F_1 score	0.6690	0.6710 **(+0.002)**
Recall	0.5760	0.5830 **(+0.007)**
Precision	0.7790	0.7910 **(+0.012)**

- 최종 제출 모델은 Public LB에서 가장 성능이 높았던 모델로 제출하였다. 베이스라인 코드의 기본 설정에서 데이터셋만 추가해서 코드를 돌린 것이 가장 성능이 좋았다.
- 데이터셋을 추가한 것이 가장 큰 성능 향상을 보였고 그다음으로 Input size를 키웠을 때 성능 향상을 보였다. ICDAR2019 MLT의 모든 데이터를 사용하고 Input size를 1024로 수정한 후 200 epoch 정도 학습했으면 더 높은 점수을 받을 수 있었을 것 같은데 시간이 부족해서 아쉽다.
- 데이터를 추가한 만큼 학습 시간이 늘어나서 초매개변수를 변경해가면서 실험을 할 수 없었던 점 또한 아쉬운 점으로 남는다.

5. 자체 평가 의견

- 김서기 : 확실하게 데이터셋이 많을수록 점수가 올라가는 폭이 높았다. 그래서 이번 대회에서는 외부 데이터를 가져와서 사용하는 방법을 중점으로 프로젝트를 진행하였다.
- 김승훈 : SynthText 데이터셋을 늦게 사용하여 학습 시간이 너무 길어 데이터셋을 제대로 이용하지 못하여 아쉬웠다.
- 배민한 : 평가 metric에 대해 조금 더 알아보는 시간을 가졌으면 좋았을 것 같다. OCR 기술이 국내 산업에 다방면으로 사용되고 있는 것으로 보아 인식기뿐 아닌 검출기 부분에서도 추가적인 공부를 수행해야 할 것으로 판단된다.
- 손지아 : `convert_mlt.py` 로 ICDAR 데이터셋을 활용할 수 있었다. 학습 시간이 오래 걸리는데 대회 기간이 짧아서 다소 아쉽다. 대회 등수보다 학습하는 게 더 중요하다지만 등수도 계속해서 오르고 모델링 실력도 향상하는 것 같아서 뿌듯하다.
- 이상은 : dataset의 중요성을 새삼 다시 깨달을 수 있었고, OCR에 관한 추가적인 공부가 필요할 것 같다.
- 조익수 : 기간이 특히 짧아 형식이 다른 여러 데이터셋을 적용하는 것도 쉽지 않았다. 데이터의 양과 질 사이 밸런스를 실험하려면 체계적인 틀이 필요한 거 같다.