



글자 검출 프로젝트 Wrap up Report

1. Introduction

1-1. Problem Statement

스마트폰으로 카드를 결제하거나, 카메라로 카드를 인식할 경우 자동으로 카드 번호가 입력되는 경우를 흔히 볼 수 있다. 이는 OCR (Optimal Character Recognition) 기술이 적용된 사례로서, 이미지 속에 있는 문자를 컴퓨터가 인식할 수 있게 한다. 본 대회는 OCR 모델인 EAST의 성능을 오로지 데이터 센트릭(data centric) 관점에서 개선한다.

Data Centric vs. Model Centric

전자는 오로지 데이터 조작을 통해 모델의 성능을 개선하고 후자는 모델의 구조를 조작하여 모델의 성능을 개선한다. 현업에서는 모델을 수정하는데 드는 비용이 크기 때문에 전자를 택해야 하는 경우가 많다.

1-2. Competition Specifics

- Model : **EAST**(An Efficient and Accurate Scene Text Detector)

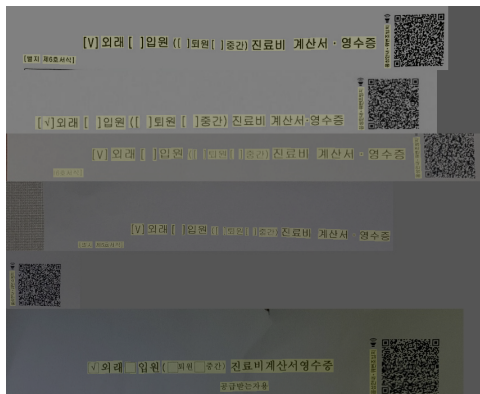
EAST 모델은 효과적인 텍스트 검출과 문자 인식을 위해 설계된 딥러닝 모델로, 속도가 빠르며 이미지에서 방향과 상관없이 인식할 수 있다.

- 제약사항: 모델과 관련된 `model.py`, `loss.py`, `east_dataset.py`, `detect.py` 코드 수정 불가
- Dataset : UFO(Upstage Format OCR) format annotation, 영수증 이미지 301장
- Evaluation metric : f1 score (DetEval)
- Experiment environment : GPU V100 server

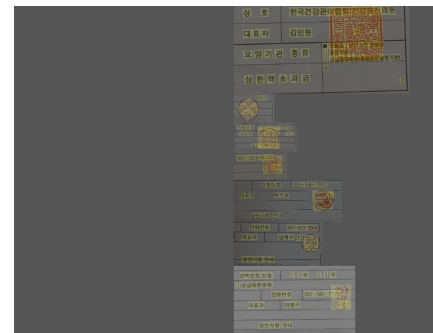
2. Experiment

2-1. Data

Dataset	Descriptions
train	기본 제공 train 데이터 100장
annotation	annotation 실습으로 만든 데이터 201장
cleaned	기본 제공 train 데이터와 annotation 실습으로 만든 데이터를 정제하여 합친 데이터 257장
public_ocr	AI Hub 사이트에서 제공하는 공공기관 문서 이미지 3900장
weak_data	모델의 취약 부분을 보강하기 위해 기본 제공 데이터와 정제된 annotation 실습 데이터 중 도장, 제목 부분을 crop하여 만든 8장의 데이터



weak_data 1



weak_data 2

annotation 데이터에는 일관되지 않은 라벨이 많아 정제하여 사용하였다. 모델 예측을 분석한 결과 도장과 글자가 겹친 경우와 우측 상단의 세로로 쓰여진 부분을 잘 예측하지 못하여 해당 부분을 보강하기 위한 데이터(weak_data)를 만들어 사용하였다.

2-2. Learning rate scheduler

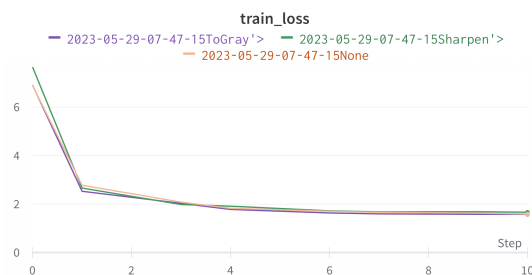
- MultiStepLR

초반 일정 에폭 동안 learning rate $1e-3$ 을 유지하다가 최대 에폭의 $1/2$ 에서 learning rate $1e-4$ 로 감소하는 방식이다. 초반부터 꾸준히 감소하는 LambdaLR은 train_loss가 일정 수준 이상으로 감소하지 않고 수렴했던 것으로 보아 높은 learning rate를 유지하는 구간이 중요했던 것을 추측할 수 있다.

2-3. Augmentations

ToGray와 Sharpen이 아무런 augmentation을 적용하지 않은 None과 비교해서 상대적으로 좋거나 비슷한 성능을 보였다.

8 에폭 기준으로 이런 성능을 보이는 것으로 보아 추가 학습을 오래 시켰다면 성능 향상에 도움이 되었을 것이다.



2-4. 학습 방법

총 4단계로 학습을 나눠서 진행

• 1차 학습

- 공공기관 문서 데이터(`public_ocr`)로 6 epochs 동안 pretrain 진행
- $lr = 1e-3$ 으로 변동 없이 진행

• 2차 학습

- 정제된 기존 데이터(`cleaned`)로 1차 fine tuning을 50 epochs 동안 진행
- $lr = 1e-3$ 으로 변동 없이 진행

• 3차 학습

- 기존 데이터로만 학습한 모델은 이미지 위쪽과 도장과 겹친 글자를 충분히 학습 못함
→ 그 부분들만 콜라주한 데이터(`weak_data`)로 18 epochs 동안 학습 진행
- lr 은 $1e-3$ 으로 진행 후, 15 epoch에서 $1e-4$ 로 진행
- full batch training을 방지하기 위해 batch size는 4로 조정

• 4차 학습

- 최종 tuning을 위해 clean data로 5 epochs 동안 학습 진행.
- $lr = 1e-3$, `batch_size = 8`

4. Conclusion

	Rank	F1 Score	Precision	Recall
Public Scores	15	0.9689	0.9658	0.9720
Private Scores	13	0.9709	0.9648	0.9770

Public 15등, f1 score 0.9689 → Private 13등, f1 score 0.9709

Private에서 점수가 오른 것을 보아 Public에 bias되지 않고 학습이 general하게 잘 된 것 같다.

5. 자체 평가 의견

잘했던 점

- 외부데이터 활용
- 데이터 분석 및 전략 수립

시도 했으나 잘 되지 않았던 것들

- 앙상블

아쉬웠던 점들

- data augmentation: validation에서 ToGray와 Sharpen이 좋은 성능을 보였는데, 시간 문제로 인해 최종 모델에 활용하지 못한 점이 아쉽다.
- 체력관리

프로젝트를 통해 배운점 또는 시사점

- random seed의 중요성