

Data-Centric : 글자 검출 프로젝트

CV-07 Wrap-UP Report

Project Outline

- OCR (Optimal Character Recognition)은 이미지 속의 문자를 컴퓨터가 인식 할 수 있도록 하는 컴퓨터 비전 분야의 대표적인 기술로, 글자 검출(text detection), 글자 인식(text recognition), 정렬기(Serializer)등의 모듈로 구성됩니다.
- 본 프로젝트에서 다루는 것은 글자 검출 task 입니다. 즉, 진료비 영수증 이미지 파일로 구성된 데이터셋에 대하여 글자의 영역을 정확하게 탐지할 수 있는 모델을 구성하는 것을 목표로 합니다. 다만 Data Centric이라는 주제의 취지에 따라 베이스라인 코드에서 주어진 모델을 그대로 활용해야 한다는 제약이 있습니다.
- 이번 대회에서는 구성한 모델로부터 생성된 UFO 형식의 output.csv 파일을 제출하여 평가를 받게 됩니다. 해당 파일에는 글자 영역으로 감지된 부분인 bounding box의 좌표정보가 포함되어 있으며, DetEval 방식으로 평가가 이루어집니다.

Team Members

- 강동기 : json파일 작성, Data Labeling, EDA
- 김한규 : 베이스라인 모델 분석, Data Labeling
- 민하은 : Data Labeling, EDA, Dataset 비교실험 수행
- 심유승 : 가설설정 및 실험 설계, Dataset 제작, Data Labeling
- 안채연 : Dataset 비교실험 수행, Data Labeling, EDA
- 이하연 : Data Labeling, 서버 환경 설정, 학습모델 재학습 코드작성

Project Progression

1. 베이스라인 모델 분석

- 이번 대회에서 활용된 베이스라인 모델은 EAST 모델인데, 대회의 규정에 따라 우선 주어진 베이스라인 코드와 train dataset을 그대로 활용하여 모델을 학습시켰습니다.
- 학습된 모델로 test data를 inference하여 얻은 output은 다음의 성능을 보였습니다.

Table 1. Baseline Model performance

F1 Score	Recall	Precision
0.8815	0.8881	0.8751

- 다만 학습된 모델에서 얻은 output을 visualization하여 분석한 결과 다음 네 가지의 문제점을 확인할 수 있었습니다.
 - ① 약 16%의 이미지에서 얼룩과 같은 노이즈를 글씨로 잘못 인식
 - ② 약 33%의 이미지에서 상단 제목 부분을 글씨로 잘 인식하지 못함
 - ③ 약 20%의 이미지에서 QR코드의 일부를 글씨로 잘못 인식

④ 약 19%의 이미지에서 QR코드 옆의 세로방향 글씨를 잘 인식하지 못함

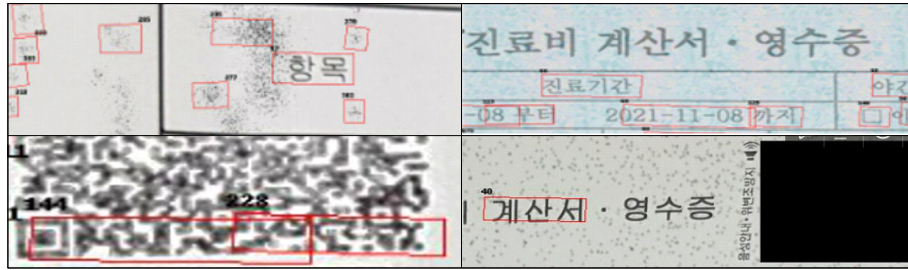


Fig 1. 베이스라인 모델 문제점 분석

2. 가설 설정

1) 가설 A - 문제점① 관련

- 얼룩과 같은 노이즈를 글씨로 인식하는 문제점과 관련하여, train dataset은 test dataset과 달리 이미지에 특별한 노이즈가 없었습니다.
- 따라서 모델이 노이즈와 관련된 학습을 제대로 하지 못하였다고 가설을 세웠습니다.

2) 가설 B - 문제점②,③,④ 관련

- 문서의 상단 제목, QR코드, QR코드 옆 세로방향의 글씨에서 발견되는 문제점과 관련하여, 진료비 영수증 데이터에서 대부분의 annotation은 표 안에 쓰여진 작은 글씨들에 대한 것이라는 점에 착안하였습니다.
- 즉, 문서의 상단 제목과 QR코드 및 QR코드 옆 세로방향 글씨는 문서에서 차지하는 비중이 매우 적고, annotation의 대부분을 차지하는 유형의 특징(표 안에 있다는 점, 글씨 크기)을 갖고 있지 않아 모델이 제대로 학습하지 못하였다고 가설을 세웠습니다.

3. Train Dataset 제작

1) Dataset A - 노이즈 추가

- Test dataset에서 글씨로 잘못 인식하는 노이즈와 최대한 유사한 노이즈를 가하기 위하여 그림판의 에어브러쉬를 활용해 원본 데이터셋 100장의 이미지 중 절반인 50장에 노이즈를 추가하여 데이터셋을 제작하였습니다.
- 따라서 제작된 dataset A는 노이즈가 추가된 이미지 50장과 노이즈가 없는 이미지 50장으로 총 100장으로 구성되어 있습니다.

2) Dataset B - 상단 제목 및 QR코드 부분 비중 늘리기

- 원본 train dataset 중에서 12장의 이미지를 선별한 뒤, 해당 이미지의 상단 제목부분과 QR코드 부분을 그림판을 활용해 복사하여 원본이미지 여러부분에 합성하였습니다.
- 생성된 이미지들에 대하여는 LabelMe 툴을 활용하여 labeling을 진행하였습니다.
- 이렇게 합성된 12장의 이미지들에 대하여 4가지 서로 다른 방법으로 augmentation하여 총 60장의 이미지를 생성하였습니다.
- Dataset B는 다음의 5가지 하위 dataset으로 구성되어 있습니다.
 - a. 60장의 이미지
 - b. 60장의 이미지 + dataset A 중 40장 (1~20, 51~70)
 - c. 60장의 이미지 + dataset A 중 60장 (1~30, 51~80)
 - d. 60장의 이미지 + dataset A 중 80장 (1~40, 51~90)
 - e. 60장의 이미지 + dataset A 전체

[illegible]







4. 실험 수행 및 결과

- 원본 dataset으로 학습시킨 모델에 대하여 dataset A를 활용해 seed 187, learning rate 1e-5, 40 epochs의 조건으로 학습을 시켜 모델을 구성하였습니다.
- Baseline model에서 recall 0.0316, precision 0.0511만큼 개선됨에 따라 F1 score 가 0.0464 개선된 0.9279의 결과를 얻을 수 있었습니다.

F1 Score	Recall	Precision
0.9279	0.9197	0.9362

- 실험 ①에서 학습된 모델에 대하여 dataset B의 5가지 dataset을 활용해 seed 187, learning rate 1e-5, 20 epochs의 조건으로 학습을 시켜 모델을 구성하였습니다.
- 실험 ①의 모델과 비교할 때 이미지 60장으로만 구성된 Dataset B(a)에서는 성능이 하락하였으나, 이를 제외한 모든 dataset에서 성능개선을 확인할 수 있었습니다.
- 특히 dataset B(d)에서 recall 0.0105, precision 0.01이 개선되어 F1 score가 0.0102 개선된 0.9381의 결과를 얻을 수 있었습니다.

Dataset	F1 Score	Recall	Precision
Dataset B(a)	0.9113	0.9003	0.9226
Dataset B(b)	0.9307	0.9211	0.9404
Dataset B(c)	0.9328	0.9242	0.9414
Dataset B(d)	0.9381	0.9302	0.9462
Dataset B(e)	0.9317	0.9255	0.9380

순위	팀 이름	팀 멤버	f1 ↕	recall ↕	precision ↕	제출 횟수	최종 제출
4 (2 ▲)	CV_07조	     	0.9508	0.9424	0.9593	32	5d

- 이번 프로젝트에서는 데이터셋을 제작해보고 이를 통해 모델의 성능을 개선키는 경험을 할 수 있었는데, AI 프로젝트에서 데이터가 차지하는 비중을 체감할 수 있었습니다.
- 최종 리더보드에서는 4위 score를 달성하였습니다.