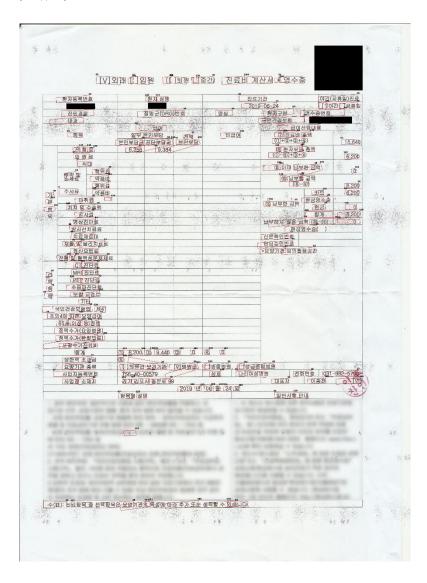
OCR Wrap-up Report

CV-9조 SCV 박상언 송지민 오왕택 이동호 이주헌 지현동

1. 프로젝트 개요

1-1. 프로젝트 주제

우리는 진료 확인서, 진단서와 같은 의학 데이터에서 글자가 존재하는 영역을 추출하는 프로젝트를 진행하였다. OCR 과정 중에서 글자의 내용을 확인하기 전에 영역을 잘 구하는 것이 주 목표였다.



2. 프로젝트 팀 구성 및 역할

이름	역할		
박상언	라벨링 가이드 작성, 라벨링 툴 실험(CVAT), 라벨링,		
	외부 데이터 가중치를 사용한 미세 조정, 깃헙 템플릿 작성		
송지민	라벨링 가이드 작성, 라벨링 툴 실험(Supervisely), 라벨링, 증강 기법 실험		
	(noise, CLAHE), 깃헙 템플릿 작성		
오왕택	라벨링 가이드 작성, 라벨링 툴 실험(Supervisely), 라벨링 검수,		
	데이터 시각화, 베이스라인 코드 실험, 데이터 증강 기법 실험(Curving), 깃		
	헙 템플릿 작성, UFO to COCO format 코드 작성		
이동호	라벨링 가이드 작성, 라벨링 툴 실험(LabelMe), 라벨링, EDA, 정규화 코드,		
이동호	라벨링 가이드 작성, 라벨링 툴 실험(LabelMe), 라벨링, EDA, 정규화 코드, 데이터 증강 기법 실험(Distortion, Gaussian noise, CLAHE), 외부 데이터 수		
이동호			
이동호	데이터 증강 기법 실험(Distortion, Gaussian noise, CLAHE) , 외부 데이터 수		
이동호	데이터 증강 기법 실험(Distortion, Gaussian noise, CLAHE), 외부 데이터 수 집, 외부 데이터 사전 학습, 깃헙 템플릿 작성, Aihub format to UFO format,		
	데이터 증강 기법 실험(Distortion, Gaussian noise, CLAHE), 외부 데이터 수 집, 외부 데이터 사전 학습, 깃헙 템플릿 작성, Aihub format to UFO format, COCO format to UFO format		
	데이터 증강 기법 실험(Distortion, Gaussian noise, CLAHE), 외부 데이터 수 집, 외부 데이터 사전 학습, 깃헙 템플릿 작성, Aihub format to UFO format, COCO format to UFO format 라벨링 가이드 작성, 라벨링 툴 실험(CVAT), 라벨링, 데이터 증강 기법 실험		

3. 프로젝트 수행 절차 및 방법

3-1. Time-line

_ 기간	수행한 일
1월 22일 ~ 1월 23일	강의 수강, 제공 데이터 및 코드 확인
1월 24일 ~ 1월 26일	Baseline code 작성, Git branch 생성, EDA, 라벨링 가이드 정의,
	라벨링 및 검수, 정답 형식 변환 코드 작성, 외부 데이터 수집
1월 29일 ~ 2월 1일	정규화 수행, 다양한 증강 기법 수행, 모형 학습

3-2. 프로젝트 수행 방법

- 1. EDA
- 2. Baseline code, 사용할 라벨링 툴 선택, 라벨링 가이드 정의, 라벨링 및 검수, 정답 형식 변환 코드(COCO to UFO, UFO to COCO)
- 3. 다양한 증강 기법 실험

4. 프로젝트 수행 결과

4-1. 데이터 살펴보기 및 라벨링 가이드

데이터의 글자는 크기, 간격 등이 모두 상이하였다. 그래서 데이터를 살펴보면서 라벨링 가이드를 같이 정의하였다. 어떻게 라벨링을 다시 해야 모형이 글자를 잘 검출할 수 있을지를 고민하였다. 그리고 우리는 아래와 같이 가이드 라인을 정하였다.

- 1. 최대한 글자의 밀도가 높도록 라벨링
- 2. 세로 글자는 높이를 기준으로 라벨링
- 3. 기본적으로 문장의 띄어쓰기 규칙으로 라벨링(단, 특수문자는 제외)
- 4. 특수문자는 폭 기준으로 결정
- 5. 두 줄은 붙어 있더라도 따로 라벨링
- 6. 애매한 부분들을 따로 정리 후 공유
- 7. 도장은 하나로 라벨링
- 8. 겹치는 글자는 각도 차이가 존재하는 경우 따로 잡기

라벨링 툴은 supervisely를 사용하였다. 라벨링을 위해 UFO 형식의 정답 데이터를 COCO 형식으로 바꾸어서 데이터를 살펴보면 위의 규칙에 맞게 라벨링을 수정하였다. 그리고 한 사람이 검수를 하여 이상한 부분들에 대하여 재라벨링을 진행하였다. 라벨링이 끝난 후 COCO 형식을 다시 UFO 형식으로 바꾸어서 학습을 진행하였다. 하지만, 라벨링을 우리가 다시 한 데이터셋에 대해서 기존의 학습을 했던 것보다 성능이 낮게 나와서 처음 주어진 데이터셋으로 실험을 진행하였다.

4-2. 증강 기법 실험

데이터를 살펴보았을 때, 종이의 색 또는 배경의 색감이 사진마다 다름을 확인할 수 있었다. 그래서 해당 부분을 최소화하는 것이 성능 향상에 도움이 될 것이라고 생각을 하였다. 그래서 EDA를 통해 학습 데이터의 통계값들을 확인하고, 정규화를 수행하였다. 정규화만을 수행하였을 때 성능 향상이 매우 높게 일어났다.

그리고 글자와 배경을 잘 구분할 수 있도록 대비를 크게하여 실험을 진행하였다. 하지만 이는 예상과는 다르게 성능 향상을 보여주지는 않았다. 이외에 가우시안 노이즈를 추가하였을 때, 성능 향상이 있는 것을 확인하였다.

기법	Recall	Precision	F1-score
기본	0.7411	0.7654	0.7530
정규화	0.9276	0.9033	0.9153
정규화 + 대비 조절(CLAHE)	0.8848	0.9168	0.9005
정규화 + Blur	0.7858	0.8071	0.7963

정규화 +	0.0222	0.0112	0.9217
Gaussian noise	0.9322	0.9113	0.9217

4-3. 외부 데이터 활용

Al hub에서 공공 데이터 외부 데이터를 추가로 300 장 사용하였다. 학습 방식은 크게 두 가지로 진행하였다. 처음에는 외부 데이터 300 장을 학습하고 우리 데이터로 미세 조정하는 방법과 두 번째는 같이 한 번에 학습하는 방식이었다. 다만, 두 가지 방식 모두 성능 향상을 보여주지 않았다.

5. 자체 평가 의견

5-1. 잘했던 점

- 라벨링 가이드 라인을 정함으로써 라벨링 하는 방식을 익혔다
- 데이터를 살펴보면서 어떠한 기법이 성능 향상을 보일지 고민할 수 있었다.

5-2. 아쉬웠던 점

- Distortion, Curving 등 기하학적 변환이 OCR 데이터 셋에서 어려워 수행하지 못한 점이 아쉽다

5-3. 프로젝트를 통해 배운 점

- 라벨링을 어떻게 하느냐에 따라서 성능 차이가 크게 난다는 것을 알 수 있었다.
- 무조건 많은 데이터보다는 양질의 데이터가 필요하다는 것을 알 수 있었다.
- 다양한 라벨링 툴을 공부할 수 있었다.