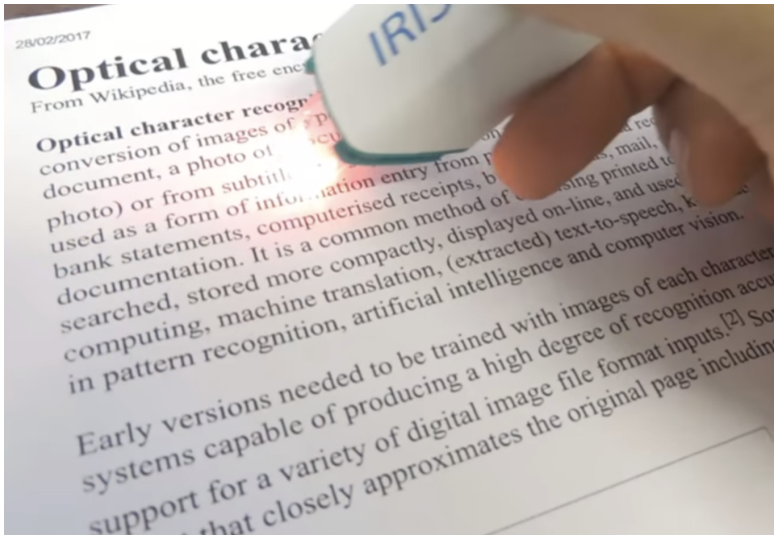
Data annotation

프로젝트 개요

스마트폰으로 카드를 결제하거나, 카메라로 카드를 인식할 경우 자동으로 카드 번호가 입력되는 경우가 있습니다. 또 주차장에 들어가면 차량 번호가 자동으로 인식되는 경우도 흔히 있습니다. 이처럼 OCR (Optimal Character Recognition) 기술은 사람이 직접 쓰거나 이미지 속에 있는 문자를 얻은 다음 이를 컴퓨터가 인식할 수 있도록 하는 기술로, 컴퓨터 비전 분야에서 현재 널리 쓰이는 대표적인 기술 중 하나입니다.



OCR task는 글자 검출 (text detection), 글자 인식 (text recognition), 정렬기 (Serializer) 등의 모듈로 이루어져 있습니다. 본 대회는 아래와 같은 특징이 있습니다.

- 본 대회에서는 '글자 검출' task 만을 해결
- 예측 csv 파일 제출 (Evaluation) 방식이 아닌 **model checkpoint 와 inference.py 를 제출하여 채점**하는 방식
- **Input** : 글자가 포함된 전체 이미지
- **Output** : bbox 좌표가 포함된 UFO Format

프로젝트 팀 구성 및 역할

- 남권표 : augmentation
- 장수호 : 학습 데이터 수집 및 가공
- 유승우 : 다양한 데이터 활용(ICDAR17, ICDAR19, AI-hub ocr 데이터)
- 조유진 : 다양한 데이터 활용(ICDAR17, ICDAR19)
- 김기훈 : 학습 데이터 조정
- 김승규 : 다양한 데이터 활용(ICDAR17, ICDAR19, AI-hub ocr 데이터)

프로젝트 수행 절차

1. upstage data 재가공
2. 역할 분담
3. 실험 진행 및 실험 결과 공유
4. 결과 종합

프로젝트 수행 결과

a. Upstage data 재가공
    a. Upstage data를 확인해본 결과 annotation이 부정확하거나 처리가 안된 이미지들 확인했다.
    b. Lableme를 통해 annotation을 재가공하여 학습에 사용했다.

b. 데이터 선정 및 실험 분석
    a. 선정
        a. upstage에서 제공한 데이터
        b. ICDAR17 MLT
        c. ICDAR19 MLT
    b. 실험 결과
        a. upstage에서 제공한 데이터를 학습시킨 결과
            - 기본 이미지(500장) + upstage에서 제공한 데이터(약 1280장) : `0.4517`
            - 다시 annotation을 적용하여 새롭게 학습한 결과 : `0.5078`
        b. ICDAR17 MLT를 사용하여 학습시킨 결과
            - 추출 언어를 en, ko이 포함되도록 설정 : 이미지 2000여장 추가하여(upstage 데이터까지 총 4000여장) 학습한 결과 `0.5620`로 증가
            - 언어 상관없이 모두 추출해 사용 : 총 8000여장(upstage 데이터 포함) 학습한 결과 `0.5816`으로 증가
            - ICDAR19 MLT 데이터까지 활용해 사용: 총 15000여장(언어 상관없이 추출, upstage 데이터 포함) 학습한 결과 `0.6443`으로 증가
        c. ICDAR19 MLT를 사용하여 학습시킨 결과
            - 기본 이미지 + 한글이 포함된 MLT2019 : 이미지 950여장 추가하여 학습한 결과 `0.5477`로 증가
            - 기본 이미지 + 한글이 포함된 MLT2019(제외 영역 포함) : 제외 영역 포함하여 학습한 결과 `0.5856`으로 증가
            - 불어를 제외한 모든 MLT2019 : 총 9000장의 이미지를 학습시켜 `0.6719`로 증가
        d. 제외영역을 학습에 포함시켰을 때와 아닐 때의 결과 비교
            a. ICDAR 데이터의 경우, annoation의 기준이 직접 annotation한 것들과 상당히 달랐다.
            b. 대회의 기준이 직접 annotation한 가이드와 같을 것이라 생각하여, ICDAR 데이터의 제외영역을 학습에 포함.
        e. Augmentation

augmentation가 적용된 이미지를 시각화해보면서 augmentation 기법을 선정했다.

a. 대회 기본 augmentation(flip, rotate, crop 등)

b. 이외 augmentation(RandomBrightnessContrast, GaussNoise, CLAHE 등)

위 augmentation을 사용하여 기본 500개의 이미지에 대한 f1 score가 `0.45`에서 `0.62`로 향상됐다.

c. 분석

a. 기본적으로 학습 데이터의 양이 많을수록 더 높은 점수를 얻었다.

b. augmentation을 활용했을때 500여장의 이미지만 사용했음에도 수천장을 학습시킨 결과와 비슷한 점수를 얻었다. augmentation을 적절히 활용하면 강력한 효과를 얻는 것 같다.

c. public test에서는 단일 종류 데이터가 더 점수를 얻었지만 private test에서는 점수가 더 낮아지는 경향이 있었다. 반면 여러 종류의 데이터를 섞어서 학습시켰을 때는 public test의 점수가 높지 않았지만 private test에서 점수가 유지되거나 향상되는 경향을 보였다. 단일 종류 데이터만 학습시키면 특정 형태에 과적합되어 성능의 불균형을 가져오기 때문인 것 같다.

c. 평가 및 개선

a. upstage에서 제공한 데이터를 EDA로 확인해본 결과 annotation이 덜 된 사진들이 있어서 추가적으로 labelme를 통해 annotation 작업을 진행시켰다.

b. 다양한 augmentation 기법을 사용하여 LB score를 비교하였다.

c. train과 validation을 분리시켜 validation precision, recall, hmean을 평가하였다.

d. 최종 제출

	Data	F1 score	recall	precision
Public Score	ICDAR17 MLT + upstage data	`0.6717`	`0.5772`	`0.8031`
	ICDAR19 MLT	`0.6719`	`0.5842`	`0.7906`
Private Score	ICDAR17 MLT + upstage data	`0.6596`	`0.5778`	`0.7685`
	ICDAR19 MLT	`0.6843`	`0.6039`	`0.7893`

제출하지 않았지만 Private 에서 제일 높았던 결과

	Data	F1 score	recall	precision
Public Score	ICDAR17 MLT + ICDAR19 MLT + upstage data	`0.6443`	`0.5549`	`0.7682`
Private Score	ICDAR17 MLT + ICDAR19 MLT + upstage data	`0.6882`	`0.5994`	`0.8079`

자체 평가 의견

a. 잘한 점들

a. 데이터를 확인하고, labelme를 사용하여 직접 annotation 수정

b. Aihub, ICDAR2019 등 다양한 데이터를 사용해본 것

c. 작은 박스들이 학습을 방해된다고 생각하여 학습 때 제외시켜본 것

d. train set, valid set으로 나누어서 과적합 여부를 파악해본 것

b. 시도 했으나 잘 되지 않았던 것들
    a. 처음 500개의 데이터를 기준으로 augmentation을 시도해봤는데 이후 다양한 데이터에 해당 기법을 적용했을 때 오류가 발생했다. 다음에는 다양한 데이터에 대한 에러 처리가 필요할 것 같다.
    b. 제외영역을 무시하지 않고 전부 학습시키는 쪽으로도 생각을 해 보았지만 생각보다 결과가 좋지 않았다.
    c. 다양한 optimizer를 적용시켜보려했지만 생각보다 학습이 느리고 잘 되지 않았다.
c. 아쉬웠던 점들
    a. 어떤 augmentation을 적용했을 때 효과적인지 알아냈지만, 적용하는데 시간이 걸려서 최종 제출을 못했다.
    b. ai hub 데이터를 활용해보려고 했지만, 너무 많은 데이터가 있어서 일부만 활용해야 했고, 실제 효과도 좋지 않았다.
    c. validation dataset에도 augmentation 기법을 적용하여 평가하다 보니 각 평가 지표가 낮게 나타나 제대로 비교하지 못했다.
    d. 학습 시 이미지가 돌아가는 경우가 있었는데, 오류를 빠르게 해결하지 못해서 성능 하락의 원인이 되었다.
d. 프로젝트를 통해 배운 점
    a. EDA를 통해 annotation이 잘 되었는지 확인하는 것이 성능 개선에서 가장 첫 번째로 생각해야 할 방법인 것을 알게 되었다.
    b. 학습시키는 데이터의 양이 많아질수록 성능 향상으로 이어진다는 것을 알게 되었다.
    c. 학습시키는 데이터가 다양했던 모델이 Public Data에 대해서 점수가 조금 낮았지만 Private Data에 대해서는 높은 성능을 보이는 것은 일반화가 더 잘 된 모델이기 때문이라고 생각된다.