Wrap-up Report

0. 프로젝트 개요

다국어 영수증 OCR

학습 데이터 추가와 수정을 통한 Data-Centric 다국어 영수증 속 글자 검출

1. 외부 데이터셋 추가

default dataset의 데이터 개수는 각 나라별 언어 당 100개, 총 400개로 학습하기에 충분하지 않은 데이터 양이라고 판단했다. 그래서 외부 데이터셋을 추가하여 데이터의 개수 자체를 늘리고, 추가할 외부 데이터셋의 퀄리티에도 신경썼다. 논의 결과, 추가한 데이터셋은 ICDAR과 CORD 데이터셋이다.

1.1. ICDAR 데이터셋

ICDAR 소개

	ICDAR 2015	ICDAR 2017	ICDAR 2019
Language	Latin scripts only	Multilingual (9가지 언어, 6가지 문자)	Multilingual (Latin scripts, 한자)
Size	1,500	18,000	10,166
Focus	Incidental Scene Text	Focused (Intentional) Scene Text	Focused (Intentional) Scene Text
Shape	Horizontal		

ICDAR은 **Incidental Scene Text Dataset** 중 하나로, 주변 환경 속에서 글자를 인식하는 대회이다. ICDAR 데이터셋을 통해 pre-training을 거친다면, 더 강건한 모델을 만들 수 있을 것이라고 가정하여 실험을 진행하였다. ICDAR 데이터셋 중에서는 2015 버전을 사용하였다.

ICDAR Dataset pre-training

- 이미지에 해당하는 Ground Truth가 text 형태로 되어 있어, 해당 부분을 하나의 json 파일로 합치는 과정을 진행했다. 이때, 대회 데이터셋 json 형태에 맞는 UFO 형식으로 annotation을 작업하였다.
- Don't care 부분인 '###'은 학습에서 제거했다.

ICDAR Results

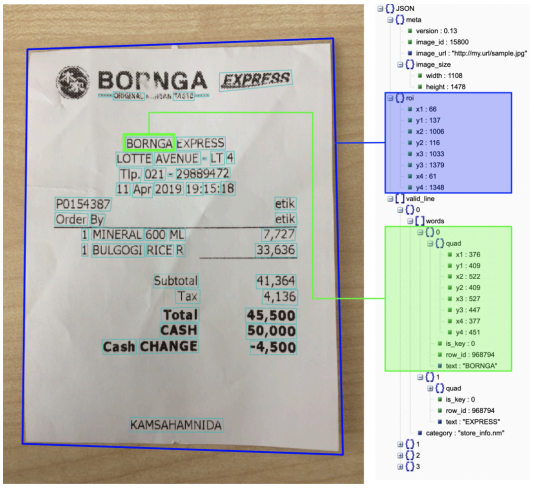
	NoChange_50epoch		0.4421 0.4021	0.4780 0.4542	0.4593 0.4265

	Pre_icdar_50epoch		0.3700 0.3813	0.3376 0.3480	0.3530 0.3639

- 비록 50epoch이지만, baseline에 비교하였을 때도 더 떨어진 성능을 보였다.
- 영수증 내 정보를 파악해야 하므로, 대회 목적에 맞지 않은 데이터셋을 활용하여 성능이 더 떨어진 것으로 파악되었다.

1.2 CORD 데이터셋

CORD 소개



CORD 데이터셋은 CLOVA AI에서 제작한 **영수증 OCR 데이터셋**이다. train 800, validation 100, test 100개로 총 1000개의 영수증 이미지를 보유하고 있다. 인도네시아에서의 영수증 이미지와 그에 해당하는 json 파일로 구성되어 있다.

CORD Dataset pre-training

- .parquet 확장자로부터 이미지와 그에 해당하는 json을 불러왔다.
- 이후, UFO 형식에 맞게 json annotation을 수정하였다.

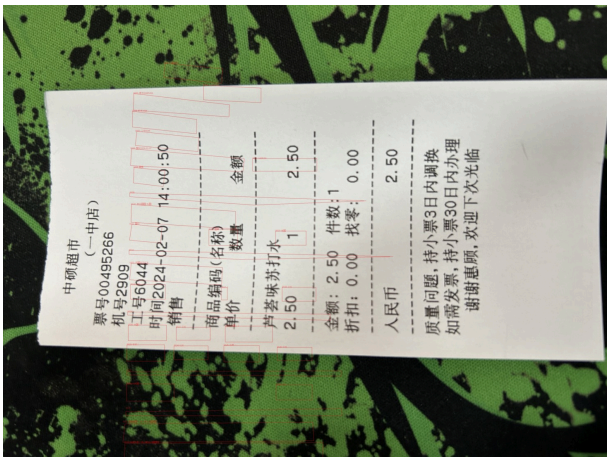
CORD Results

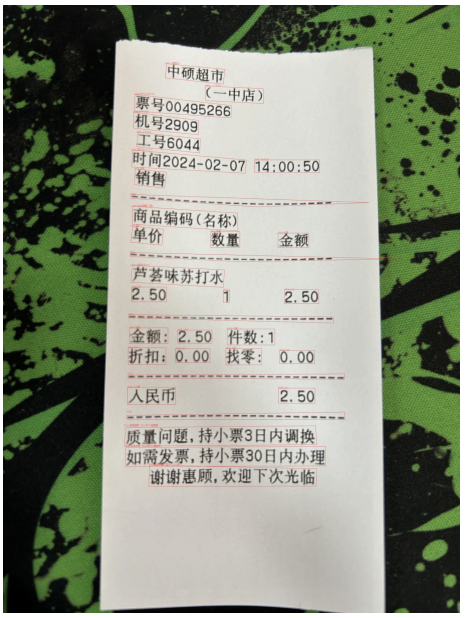
	NoChange_50epoch		0.4421 0.4021	0.4780 0.4542	0.4593 0.4265
	clova_test		0.5930 0.5699	0.5734 0.5415	0.5830 0.5553

- baseline에 비해, 더 향상된 성능을 보였다.
- 영수증 데이터셋이라는 유사점으로 인하여, 성능이 향상된 것으로 파악하였다.

2. 기존 데이터셋 이미지 분석

2.1 이미지 orientation 수정





문제 파악

위 이미지와 같이 데이터셋 파일내에 metadata 의 orientation 때문에 자동으로 이미지가 눕는 현상 발생했다.

해결 방안

EXIF 데이터를 확인해 이미지의 회전 정보를 읽고, 잘못된 방향으로 눕는 문제를 자동으로 수정했다.

2.2 라벨노이즈 제거

문제 파악

`Streamlit`을 이용한 이미지 분석 과정에서 `transcription` 필드에 Null 또는 빈 문자열 값이 포함된 데이터가 발견. 이러한 값들은 의미 있는 정보가 없어 모델 학습 시 불필요한 잡음을 유발할 가능성있음.

해결 방안

`train.json`에서 `transcription["word"]` 필드가 Null이거나 빈 문자열인 항목을 모두 제거하였습니다. 그 결과 모델의 성능이 향상되었습니다.

라벨노이즈 제거 결과

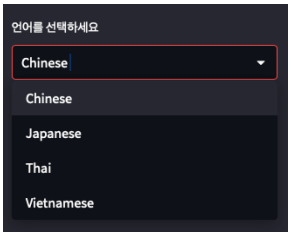
☐	NoChange_50epoch		0.4421 0.4021	0.4780 0.4542	0.4593 0.4265
☐	ReLabelling 50epoch		0.5503 0.5239	0.5088 0.4879	0.5288 0.5053

3. Streamlit을 이용한 시각화 구현

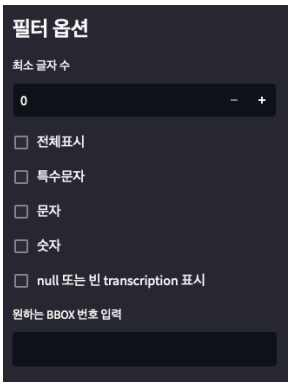
Train image BBOX와 Test image 결과를 확인하며 전략을 구상하기 원활하게 여러 기능을 갖춘 streamlit library를 이용하여 시각화 코드를 구현하였다.

전체화면

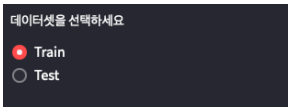
언어 선택 기능



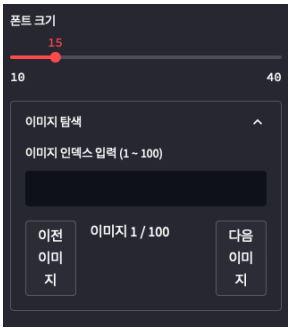
필터 구현



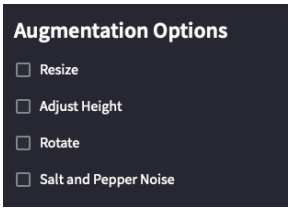
데이터셋 선택 기능



폰트 크기 및 이미지 검색



augmentation 시각화



- 언어 선택 기능
  - Chinese / Japanese / Thai / Vietnamese 총 4가지 언어에 대한 영수증 dataset으로 언어별 특성과 bbox 분포를 파악하기 위해 선택한 언어에 대한 이미지를 시각화하도록 구현하였다.
- 데이터셋 선택 기능
  - Train / Test image를 각각 시각화하여 annotation을 확인할 수 있게 하였다.
    - Train에서는 annotation 관련 여러 필터를 구현하여 특수한 bbox를 확인할 수 있게 하였고, augmentation iamge를 시각화하여 원하는 augmentation이 잘 되는지를 확인할 수 있게 하였다.
    - Test에서는 학습된 csv 파일에 대한 bbox의 분포를 알고 학습이 원하는 방향으로 잘 이루어졌는지 확인할 수 있게 하였다.
- 필터 옵션 선택 기능
  - 최소 글자 수 필터
    - 글자수가 너무 많은 annotation은 bbox 형태에 대한 ratio가 큰 값을 가질 수 있으니, 많은 글자 수를 가진 annotation들을 파악하고 수정하기 위해 글자 수 필터를 구현하였다.
  - 문자 필터
    - 전체표시 / 특수문자 / 문자 / 숫자 / null 또는 빈 transcription 선택지를 주어 선택한 항목이 하나라도 들어간 annotation들을 시각화 할 수 있게하여 특수문자나 null 또는 '' 등의 annotation들을 파악하고 수정할 수 있게 구현하였다.
  - bbox 검색 기능
    - 특정 bbox 확인을 용이하게하기 위해 해당 image에 대한 원하는 bbox 번호를 입력하면 해당 bbox만을 출력하게 하였다.
- 폰트 크기 조절 기능

    - 한 image에 많은 annotation이 있어 시각화시 bbox number나 transcription이 잘 안보이는 상황이 잦았다. 따라서 폰트 크기를 조절하여 시각화를 용이하게 하였다.
- **이미지 탐색 기능**
    - 이미지에 인덱스를 부여하여 인덱스로 검색이 가능하게 하였고, '이전 이미지', '다음 이미지' 버튼을 구현하여 순차적으로 이미지를 넘기며 확인할 수 있게 하였다.
- **augmentation 이미지 확인**
    - dataset.py에 augmentation code를 이용하여 augmentation이 이미지에 원하는 의도대로 적용이 되었는지 시각화하여 확인할 수 있게 하였다.

4. 새로운 Augmentation 적용

4.1 OCR 데이터에 맞는 방식 적용

- EDA를 기반으로 추가할 Augmentation을 선정하였다.
- 선 인식을 잘하지 못하거나 경계선을 선으로 인식하는 등의 문제를 해결하기 위해 인공적으로 텍스트를 추가하거나 선을 추가하는 방식을 사용하였다.
- 영수증 이미지들의 노이즈나 화질에 대한 문제를 해결하기 위해 노이즈를 추가하는 방식을 사용하였다.
- 영수증 이미지들이 다양한 조명 조건과 구부러짐을 가짐을 확인하여 다양한 각도와 조명 조건을 주는 방식을 추가하였다.

4.2 적용한 Augmentations

- **translate_image** : 텍스트와 이미지를 다른 위치로 이동시킨다.
- **perspective_transform** : 기울어진 텍스트와 다양한 각도의 텍스트 배치를 학습한다.
- **adjust_brightness_contrast_saturation** : 다양한 조명 조건을 적용한다.
- **add_gaussian_noise** : 모델이 노이즈가 있는 이미지에 대해 더 잘 작동하도록 노이즈를 추가한다.
- **add_salt_and_pepper_noise** : 이미지에 흰색, 검정색 점을 추가한다.
- **overlay_text** : 인공적으로 텍스트를 이미지에 추가하여 다양한 텍스트 패턴을 생성한다.
- **add_random_lines** : 인공적으로 선을 이미지에 추가한다.

4.3 파라미터 조정 후 실험 결과물

augmentation	epoch	**precision**	**recall**	**f1**	적용 파라미터
baseline	50	0.4421	0.4780	0.4593	Rotation Parameter: 90 (영구변경)
translate_image	50	0.4450	0.5144	0.4772	-10, 10픽셀 사이의 고정된 이동 비율
translate_image	50	0.3403	0.3747	0.3567	Ratio 5%로 실험
translate	150	0.6649	0.6934	0.6788	-10, 10픽셀 사이의 고정된 이동 비율 모두 적용
perspective_transform	50	0.2950	0.2983	0.2966	shift=5%
adjust_brightness_contrast_saturation	50	0.2194	0.3029	0.2545	brightness=(0.8, 1.2), contrast=(0.8, 1.5), saturation=(0.8, 1.5)
Gausian Noise	50	0.4159	0.4628	0.4381	std_range : (0.05, 0.15)

Gausian Noise 2		0.4707	0.4577	0.4641	mean : 0 ⇒ 이 설정은 보통 고정
Gausian Noise 3	50	0.4037	0.4351	0.4188	std_range : (0.01, 0.05) mean : 0 ⇒ 이 설정은 보통 고정
add_salt_and_pepper_noise	50 70	0.4970 0.5621	0.4940 0.6426	0.4955 0.5997	amount=0.02
overlay_text overlay_modify	50	0.3595 0.3513	0.3996 0.4961	0.3785 0.4114	text : “Sample Text” position : None ⇒ random font_size : 15 color : (0, 0, 0) ⇒ black text : “Sample Text” position : None font_size : 15 color : 이미지 평균에 따른 자동 변화
add_random_lines	50	0.4484	0.4524	0.4504	num_lines=5, thickness=2, max_attempts=10
translate + gausian	100	0.7086	0.7028	0.7057	-10, 10픽셀 사이의 고정된 이동 비율 std_range : (0.05, 0.15) 0.5 확률 적용
translate + gausian	150	0.6998	0.7315	0.7153	

4.4 최종 테스트 버전 기준

- 예측 결과 이미지가 좋은 성능을 보이는 Augmentation을 섞었다.
- 적절한 random 확률을 주고 다르게 적용했다.
- 고정된 값들은 다음과 같다.
  - Clova pretrained pths
  - epoch = 150
  - baseline Augmentation 적용

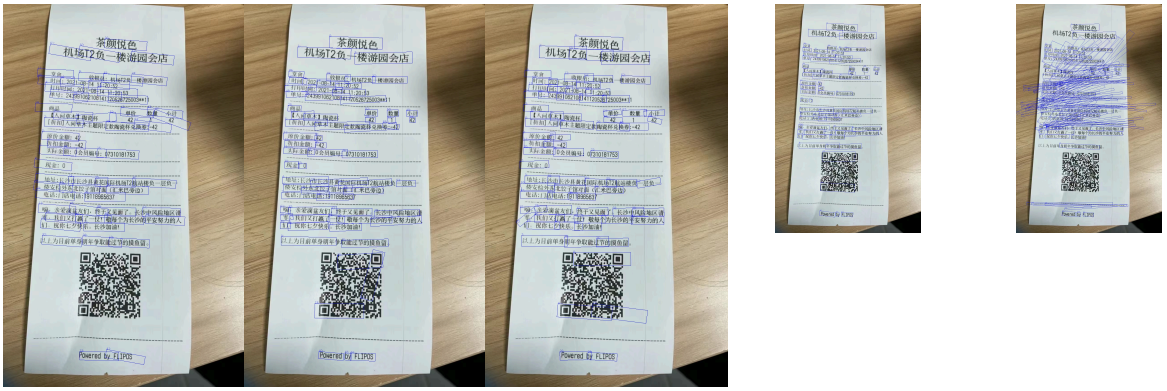
4.5 결과 분석

- baseline코드에 비하면 훨씬 깔끔하게 인식한다는걸 알 수 있다.
- baseline 코드보다 bbox가 기울어진 모습을 보였다.
  - 한 글자 인식할 때 주로 발생한다.
- Salt And Pepper는 bbox를 좀 크게 인식하는 경향과 스캔한 이미지에서 약세를 보였다.
- Gaussian은 bbox를 크게 인식하거나하는게 없다.
- Gasussian만 적용한 ver의 성능이 가장 좋았다.
- bbox는 많지만 글자에 가장 fit하게 인식하는건 baseline코드였다.

version	epoch	**precision**	**recall**	**f1**	적용 파라미터

version 1	150	3700	3376	3530	translate(50%), salt and pepper(50%), add line(50%)
version 2	150	0.7734	0.7032	0.7366	translate(50%), gaussian(50%), add line(50%)
version 3	150	0.5503	0.5088	0.5288	translate(50%), salt and pepper(35%), gaussian(35%), add line(50%)
version 4	150	0.6855	0.6681	0.6767	rotate 90

ver1. ver2. ver3. ver4. baseline



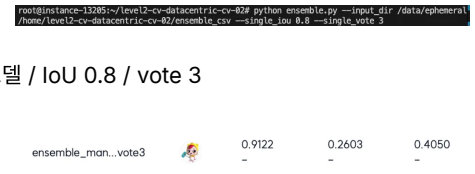
5. Ensemble 구현

hard voting 방식으로 학습한 모델들을 ensemble하여 성능을 끌어올리기 위해 ensemble 전략 설정 후 ensemble을 적용하였다.

Ensemble 전략

- bbox가 많이 나오는 model들을 모아 Ensemble
  - 모델을 담당하는 인원들의 'bbox가 많아야 성능이 좋게 나온다'는 의견을 반영하였다.
- 리더보드 score가 높은 model들을 모아 Ensemble
- bbox가 많이 나오는 모델 + socre가 높은 모델 전부 Ensemble

BBOX가 많은 models ensemble



6개 모델 / IoU 0.8 / vote 3

Precision은 0.9122가 나오지만 나머지 두 score가 현저히 낮은 모습을 보인다.

성능 좋은 models ensemble

7개 모델 / IoU 0.8 / vote 3

마찬가지로 precision은 0.892이 나오지만 나머지 두 score가 0.3381 / 0.4905 값을 가진다.

전부 다 앙상블

- 앙상블 코드 실행 시 결과 확인까지 2시간 이상의 시간이 걸릴 것으로 예상되었다. 리더보드 마감이 얼마 남지 않아 전부 다 앙상블하는 전략은 포기하였다.

최종 앙상블

- 3가지 score를 비교하였을 때 성능이 좋은 model들만을 모아 ensemble 했을 때 3가지 점수가 고르게 높게 나올 가능성이 높다고 판단하였다. 따라서 성능 좋은 모델들을 모아 파라미터를 수정해가며 앙상블을 진행하였다.
- 또한, 위에서 IoU를 0.8로 설정하였는데, 너무 높은 IoU 값으로 설정하여 bbox가 제대로 검출이 안된 모습을 보인듯하여 IoU 값을 낮춰 실험 진행했다.

IoU0.5 / vote 4

IoU가 낮아야 recall과 f1-scores가 높게 나오는 것을 확인하였다.

vote 개수에 영향을 파악하기 위해 다음으로는 vote 개수를 조절하였다.

IoU 0.5 / vote 3

vote를 4 → 3 으로 줄였을 때 Precision은 0.03 가량 감소하였고, recall은 0.04 f1은 0.01 가량 증가하는 모습을 보였다.

3개 score 전반의 성능을 올려야하니 vote는 3으로 고정하고 나머지 실험을 진행하였다.

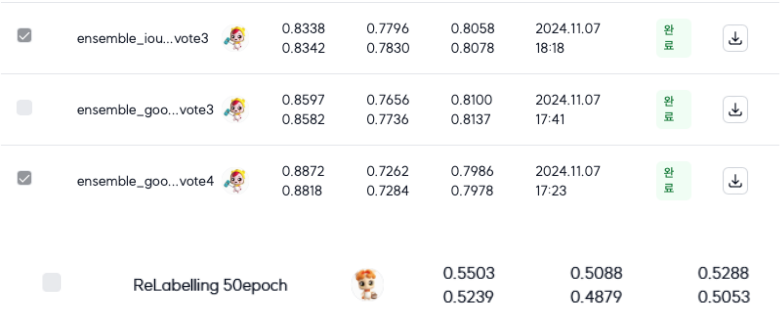
IoU 0.4 / vote 3

IoU를 감소시켰을 때 성능이 올랐으니 0.1 가량을 더 감소시켜보았다.

마찬가지로 precision은 소량 감소, recall과 f1은 소량 증가한 모습을 보였고, 현재 model들로는 해당 값 이상의 큰 변화는 없을 것으로 판단하였다.

최종 ensemble 모델

3개의 점수가 비교적 고르게 좋은 3개 모델 중 2개의 **모델이 선정되었고, public socre와 private score는 다음과 같다.**



☑	ensemble_iou...vote3	0.8338 0.8342	0.7796 0.7830	0.8058 0.8078	2024.11.07 18:18	완료	⬇
☐	ensemble_goo...vote3	0.8597 0.8582	0.7656 0.7736	0.8100 0.8137	2024.11.07 17:41	완료	⬇
☑	ensemble_goo...vote4	0.8872 0.8818	0.7262 0.7284	0.7986 0.7978	2024.11.07 17:23	완료	⬇
☐	ReLabelling 50epoch	0.5503 0.5239	0.5088 0.4879	0.5288 0.5053			