CV-08 랩업 리포트

-Level-2 다국어 영수증 OCR 대회 리뷰-
CV-08조 팀 SEE-Beyond-Accuracy

1. Project outline

카메라로 영수증을 인식할 경우 자동으로 영수증 내용이 입력되는 어플리케이션이 있습니다. 이처럼 OCR (Optical 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.

OCR은 글자 검출 (Text detection), 글자 인식 (Text recognition), 정렬기 (Serializer) 등의 모듈로 이루어져 있습니다.

다국어 (중국어, 일본어, 태국어, 베트남어)로 작성된 영수증 이미지에 대한 OCR task를 수행합니다.

이미지에서 어떤 위치에 글자가 있는지를 예측하는 모델을 제작하고 학습 데이터 추정을 통한 Data-Centric 다국어 영수증 속 글자 검출을 진행합니다.

평가지표는 DetEval을 사용합니다. DetEval은 이미지 레벨에서 정답 박스가 여러개 존재하고, 예측한 박스가 여러개가 있을 경우, 박스끼리의 다중 매칭을 허용하여 점수를 주는 평가 방법 입니다.

프로젝트 전체 기간 (2주) : 10월 30일 (월) 10:00 ~ 11월 7일 (목) 19:00

2. Dataset

2.1 Dataset 구조

그림 1. dataset 구조도

Total (Train / Test) : 520 (400 / 120)

언어(4개) : chinese, japanese, thai, vietnamese

그림 2. UFO 파일 예시

word : 단어 단위의 텍스트 정보가 저장되는 곳
img_w : 이미지 너비
img_h : 이미지 높이
annotation_log : 주석 작업의 기록
license_tag : 데이터의 라이선스 정보
points : 텍스트의 좌표 값

2.2 베이스라인 및 사용 모델

EAST (An Efficient and Accurate Scene Text Detector; Zhou et al., 2017)

2.3 코드 수정제한

- model.py
- loss.py
- east_dataset.py
- detect.py

2.4 Annotation Format

- UFO

3. Team Members and Roles

Name	Role
임용섭	가이드 라인 작성, Hyper Parameter Tuning, Labeling
박재우	EDA, 데이터 전처리, Labeling
이상진	EDA, Backend, Labeling, 추가 데이터 탐색 및 학습
정지훈	EDA, 데이터 전처리, Labeling
천유동	데이터 전처리, 데이터 증강, 구분선 학습, 구분선 EDA
유희석	가이드라인 작성, 데이터 전처리, Labeling, Image Scale

4. EDA

✅ Annotation Error

A. 구분선

영수증의 바깥 영역에까지 라벨링(labeling)이 되어있는 경우가 존재, 배경을 포착하여 모델이 잘못 학습할거라 예상

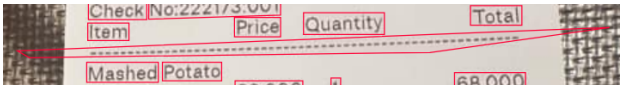

그림 3. 구분선을 잘못 annotation한 이미지

B. 글자 영역

글자 영역 BBox가 정확하지 않게 주석되어 있어, 모델이 올바른 텍스트 위치를 학습하지 못할거리 예상

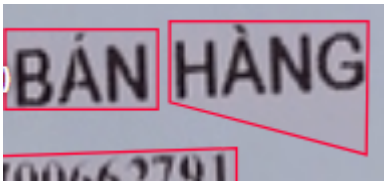

그림 4. 글자 영역 Annotation을 잘못한 이미지

C-1. 일관성 **(**시작 지점**)**

모든 BBox가 글자 좌 상단에서 시작해 시계 방향으로 주석되지 않아, 일관성을 해친다.

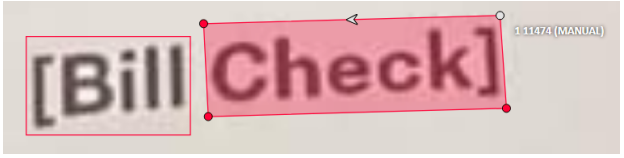

그림 5. 우 상단에서 반시계 방향으로 시작한 주석

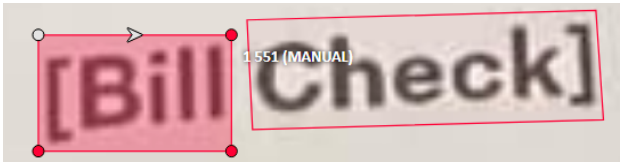

그림 6. 좌 상단에서 시계 방향으로 시작한 주석

C-2. 일관성 **(**박스 분리**)**

박스를 분리하는 기준이 다르며, 일관성을 해친다.

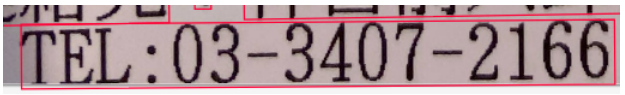

그림 7. 하나의 박스로 주석

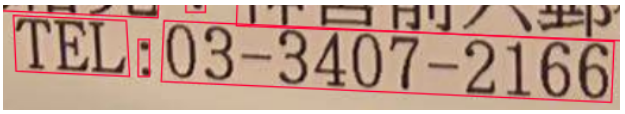

그림 8. 여러개의 박스로 주석

D. 박스 겹침

글자 영역이 휘어져 있어 하나의 박스로 주석처리 하려니 다른 BBox가 침범되는 문제점이 발생

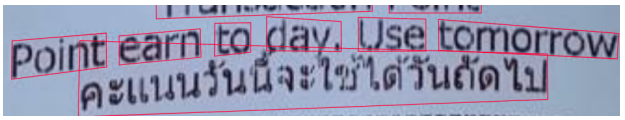

그림 9. BBox가 겹치는 이미지

E. 로고

로고와 글자를 같이 주석처리하여, 모델이 로고를 텍스트로 잘못 인식할 수 있다 예상

그림 10. 로고와 글자를 하나의 박스로 주석

🏃‍♂️**Approach :**

잘못된 주석을 Re-labeling 작업을 수행하거나, 배경 영역에 잘못된 주석을 삭제하여 데이터셋 정제

✅구분선

원본데이터 총 bbox개수 **31162**개

구분선 bbox가 **1343**개

약 **4.3%**정도 비율의 bbox가 구분선

🏃‍♂️**Approach :**
구분선을 따로 학습하거나 아예 빼놓고 진행

✅구분선 **Width & Height**

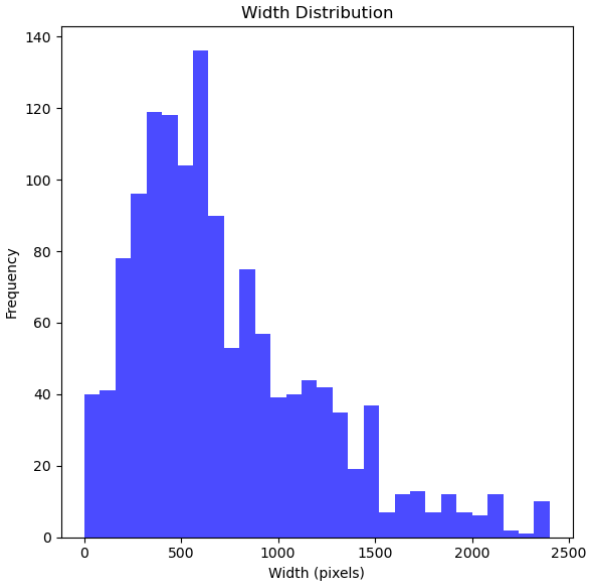

그림 11. 구분선 Width

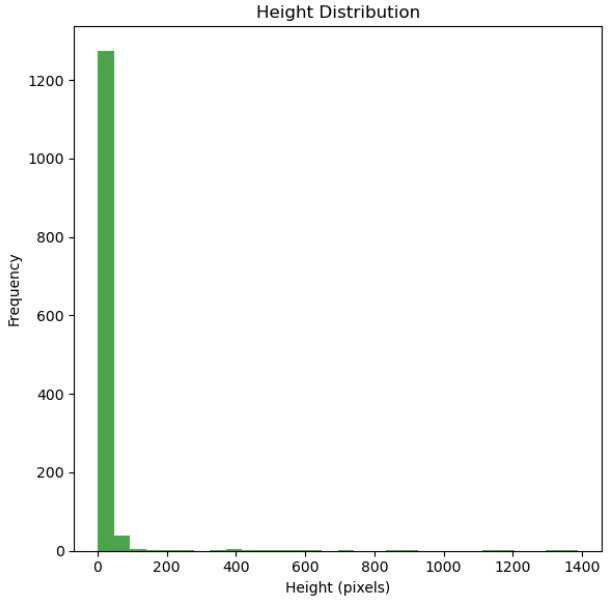

그림 12. 구분선 Height

5. Annotation Guidelines

✅ Guideline v1

1. 박스 포인트는 좌측 상단을 시작점으로 시계 방향으로 찍는다.
2. 점의 개수는 4개로 제한한다.
3. 글자 크기를 기준으로 1/2보다 큰 공백이 있으면 박스를 나눈다.
4. 곡선의 형태로 구성된 문자 및 구분선은 박스를 n개로 나눠 라벨링
5. 특수문자 또한 박스에 포함한다.

그림13. Guideline v1 annotation

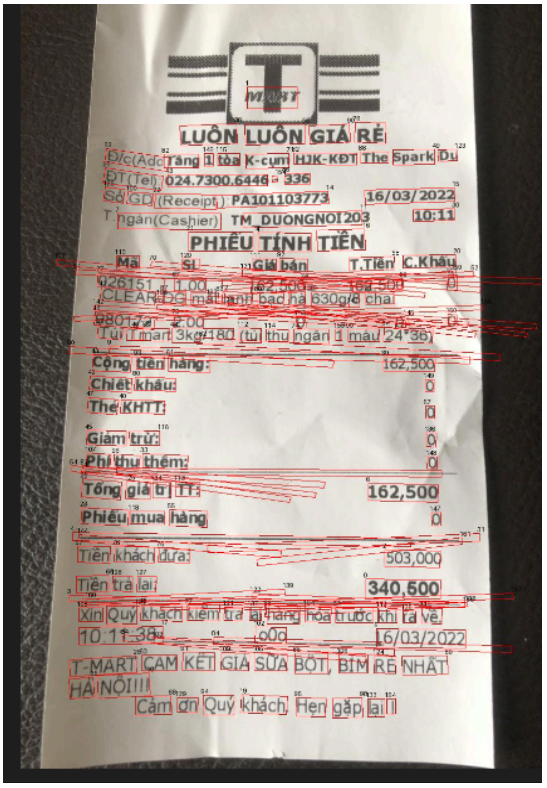

그림 14. Guideline v1 inference result

✅ Guideline v2

Ver1 결과 분석을 통해, 구분선을 여러 개의

작은 바운딩 박스로 인식하여 다양한 각도의 바운딩 박스가 과도하게 생성되고 있음을 확인. 이로 인해 전체적인 성능 저하가 발생한다고 판단하여, 구분선을 모두 제거하는 가이드라인을 수립

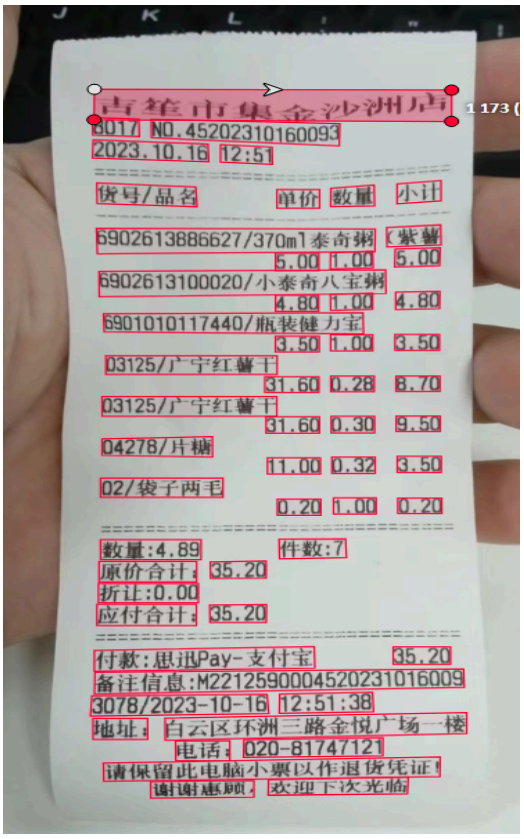

그림 15. Guideline v2 annotation

그림 16. Guideline v2 inference result

✅ Guideline v3

이미지 내 텍스트는 대부분 프린팅된 글자이므로, 손글씨와 특수문자가 모델에 혼란을 줄 수 있다는 가설을 세워 손글씨와 특수문자를 모두 제거하는 가이드라인을 수립

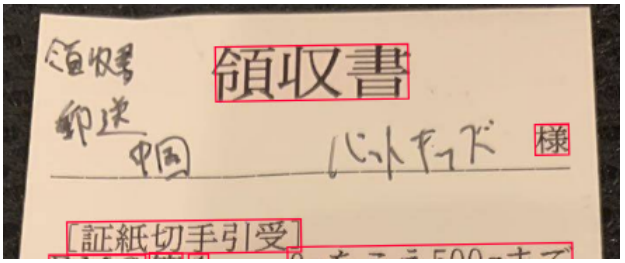

그림 17. 손글씨가 포함되어있는 이미지

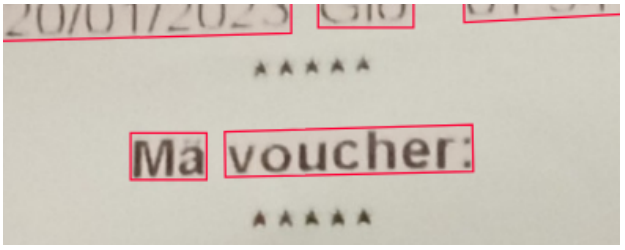

그림 18. 특수문자가 포함되어있는 이미지

✅ Guideline v4

휘어진 글자에 대한 바운딩 박스가 크기와 회전으로 인해 모델에 혼란을 줄 수 있다는 가설을 바탕으로, 휘어진 텍스트를 나누지 않고 여백이 생기더라도 하나의 직사각형 박스로 라벨링하도록 수정

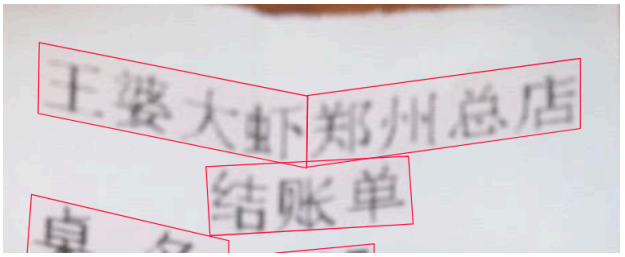

그림 19. 휘어진 글자 박스를 나누어 라벨링

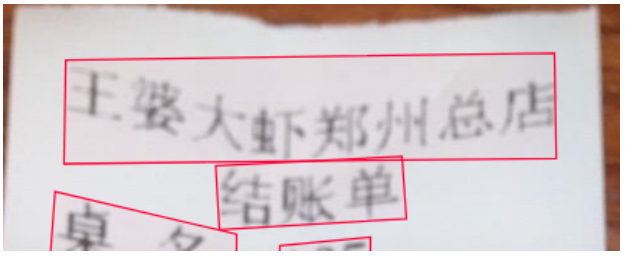

그림 20. 휘어진 글자 하나의 직사각형 박스로 처리

✅ Guideline Result

Method	Precision	Recall	F1

Original data	0.6590	0.8645	0.7479
Guideline v1	0.6833	0.8081	0.7405
Guideline v2	**0.9267**	**0.8699**	**0.8974**
Guideline v3	0.9191	0.8458	0.8809
Guideline v4	0.9146	0.8505	0.8812

표 1. 가이드라인 버전 별 성능지표.

Guideline v2에서 Precision과 F1-score가 가장 높아 성능이 크게 향상되었으며, 특히 구분선 제거와 특수문자 라벨링이 모델 성능 개선에 효과적이었습니다.

6. Data Preprocessing

✅ **Image sampling method**

학습 파이프라인에는 이미지를 고정된 크기로 resize 하는 과정이 있는데, 그 과정에서 소실되는 정보로 인해 성능 저하가 발생할 것이라고 생각했다.

resize시 sampling method를 달리해서 실험을 설계했다. baseline에서 사용된 Bilinear 기법은 인접한 픽셀 4개를 이용해 선형적으로 예측하는 방식이다. 빠른 연산속도를 가지고 있지만, 그 만큼 정확한 예측을 하지 못한다는 특징이 있다.

Bicubic 기법은 인접한 16개의 픽셀을 이용해 거리에 따른 가중 곱 연산을 이용해 예측하는 방식으로 bilinear보다 더 높은 성능을 보이는 것으로 알려져 있다.

Lanczos 기법은 bicubic과 동일하게 인접한 16개의 픽셀을 이용해 가중 곱 연산을 한다. 하지만 sin 함수를 이용해 더 높은 성능을 보이는 것으로 알려져 있다.

Method	Recall	Precision	f1 score
bilinear (baseline)	0.9023	0.8473	0.8740
bicubic	0.9104	0.8540	0.8813
lanczos	**0.9184**	**0.8627**	**0.8897**

표 2. 보간법 별 성능지표. Lanczos기법을 사용했을 때 가장 높은 성능을 기록했다.

7. Augmentation

다양한 상황에서 검출 성능을 유지하기 위한 증강기법들을 조합해서 학습 진행하였다. Base로 resize (bilinear), adjust height, rotate (10 degrees), crop, normalize을 포함시켜서 진행하였다.

Method	F1 score
Base + colorjitter	**0.8974**
Base + colorjitter + perspective	0.7698
Base + CLAHE	0.8240
Base + colorjitter + blur + shadow + GN	0.8606

Gaussian Noise(GN)

표 3. 증강 별 성능지표, Base에 Colorjitter를 사용할 때 제일 좋은 성능을 보였다.

Base 증강에 Colorjitter 증강이 가장 유효했다. Colrjitter의 경우 색상, 대비, 채도, 밝기를 무작위로 조절하는 증강으로 이러한 다양한 증강을 통해 CLAHE, blur, shadow, gaussian noise와 같이 특정 특징을 증강하는 기법보다 일반화가 더 잘 진행되었다고 판단하였다.

8. Additional Dataset

총 **5**개의 추가 데이터셋을 수집하고, Easy OCR을 이용해 Pseudo 레이블링을 진행하고 학습에 활용

1. Kaggle 베트남어 영수증
2. Roboflow 태국어 영수증
3. Clova synthdog 중국어 합성 데이터
4. Clova synthdog 일본어 합성 데이터
5. Clova CORD ver2 영어 영수증

Method	F1 score
Ver 2 + All	0.8536
Ver 2 + Kaggle + Roboflow + CORD	0.8693
Ver 2 + Kaggle + CORD	0.8715
Ver 2 + CORD	0.8818
Ver 2 Only	**0.8974**

표 4. 추가 데이터셋을 학습한 결과

Ver 2 데이터셋에 추가 데이터셋을 사용한 결과, **Ver 2 Only**가 가장 높은 성능을 보였다. 합성 데이터는 기존 영수증과 다른 특징이 있어, 제거했을 때 약간의 성능 향상이 있었지만, 여전히 **Ver 2 Only**보다 낮았다.

9. Ensemble

✅ **Test Score**와 가이드 라인의 다양성을 기준 **Ensemble**

Precision, Recall 점수와 모델의 다양성을 기준으로 상위 모델들을 선택하여 앙상블을 구성하였으며, 다양한 특징을 가진 모델들의 결합을 통해 성능을 극대화하고, 일반화 성능을 향상시키는 것을 목표로 하였다.

지금까지 학습한 모델들 중 상위의 점수를 기록한 모델들을 IoU점수 기준으로 hard voting 하는 방식으로 여러 가지 섞어보면서 앙상블 진행

Method	F1 Score
5 model IoU 0.3 vote 1	0.8655
5 model IoU 0.5 vote 2	0.9106
5 model IoU 0.7 vote 2	0.9052
11 model IoU 0.45 vote 6	0.9129
11 model IoU 0.45 vote 7	0.9073
17 model IoU 0.45 vote 6	**0.9200**
17 model IoU 0.45 vote 7	0.9180
17 model IoU 0.50 vote 7	0.9181
17 model IoU 0.50 vote 8	0.9152
17 model IoU 0.55 vote 6	**0.9196**
17 model IoU 0.55 vote 7	0.9163
17 model IoU 0.60 vote 8	0.9079

5 models : Shadow, Blur, Gaussian Aug + Lanczos_best + input resize 1600 + v2 + v2_5dataset

17 models : 5 models + ver1 + ver2 remove lines+ ver3_last + ver3_best + ver4 + CLAHE + Lanczos_latest+ input resize(1600,2048,3200) + bicubic, noBlur

10. Final Score

✅ 최종결과

- Public Score

- **4** / 24
- F1 Score : **0.9200**

- Private Score

- **5** / 24
- F1 Score : **0.9073**

Public Score보다 Private Score에서 하락하였다.

최종순위 ***5***위 기록

11. 협업 및 개발 환경

A. 협업 방법

i. Google Spreadsheet을 통한 프로젝트 진행 과정 및 실험 결과를 공유

ii. Git과 Github를 활용하여서 개인이 작성한 코드를 공유하고 서로 리뷰

iii. Zoom을 활용하여 실시간으로 소통하면서 협업진행

iv. Slack, 카카오톡, 구글 시트를 통해서 서버 사용현황 실시간 공유

B. 개발 환경 및 **Tool**

i. 개발언어 : Python

ii. 개발환경 : V100 32GB GPU

iii. Frame Work : Pytorch

iv. 협업툴 : MLFlow, Slack, Notion, Git, 카카오톡, Google Sheet

12. 팀 회고 및 개선 방안

✅ 잘했던 점

- 팀원들끼리 자유로운 아이디어의 공유가 좋았다
- 대회에 익숙하지 않은 팀원들도 잘 적응할 수 있었다
- 가설을 세우고 이를 검증하면서 모델링을 진행한 것이 좋았다
- 회의를 통해서 역할을 분담하고 체계적으로 진행한 것이 좋았다
- 학습 현황 알람, 스프레드 시트 연동 등의 기능 구현을 통해 실험을 편하게 할 수 있었다.

✅ 아쉬웠던 점

- 개인마다 진행 속도가 차이가 나서 아쉬웠다
- Git과 같은 협업툴을 적극적으로 활용하지 못해서 아쉬웠다