검출과 인식 모델을 하나로? : challenge 우승 OCR 서비스 모델 새 출시!

길태호, 서석민
NAVER CLOVA

NAVER DEVIEW 2023

CONTENTS

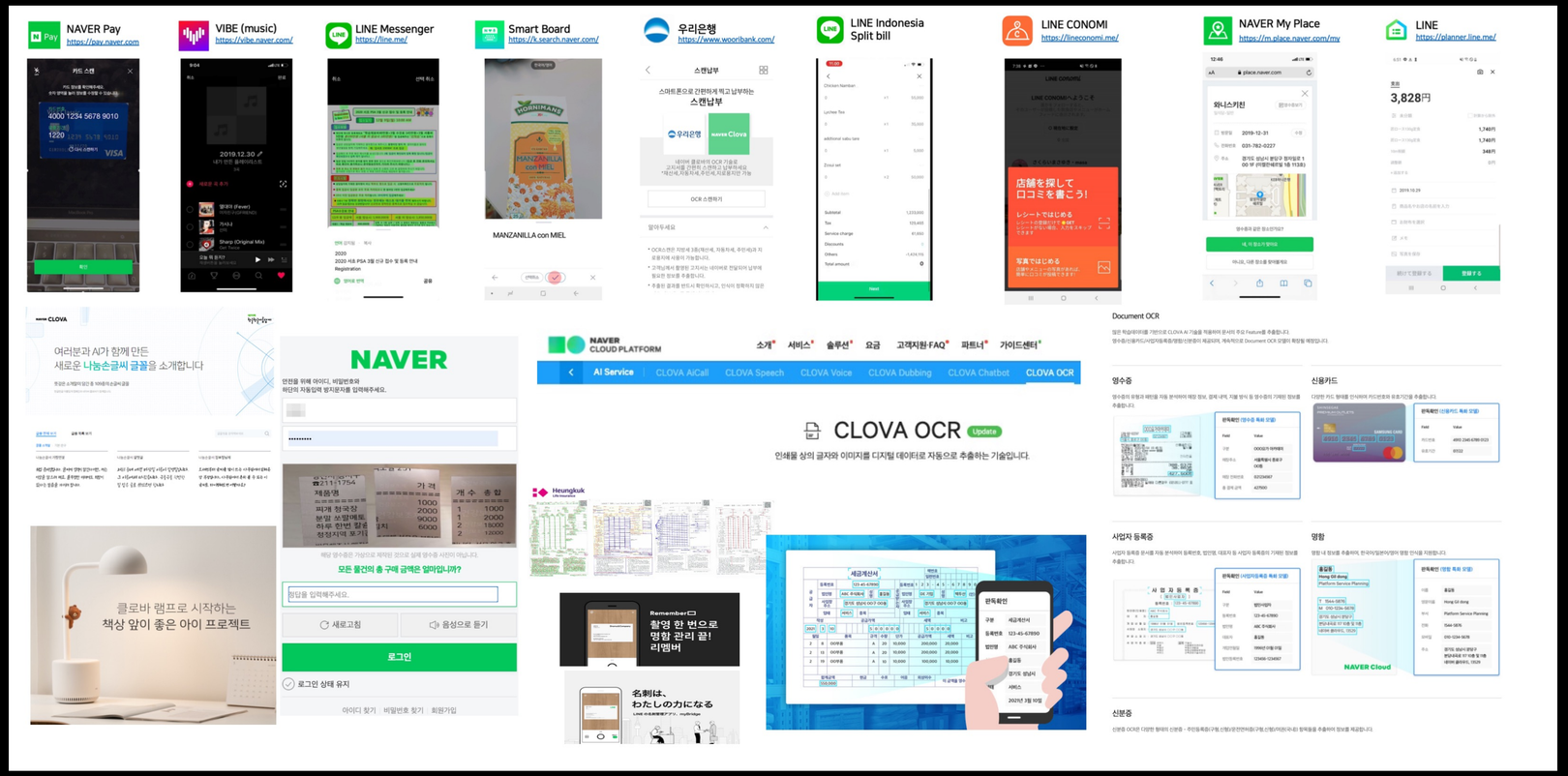
1. 기존 Clova OCR 서비스 모델

1.1 CLOVA OCR의 서비스 현황

NAVER
DEVIEW
2023

OCR이 적용된 서비스들

- Naver My Place의 영수증 인식, 신용카드 인식 등 많은 곳에서 OCR을 이용중



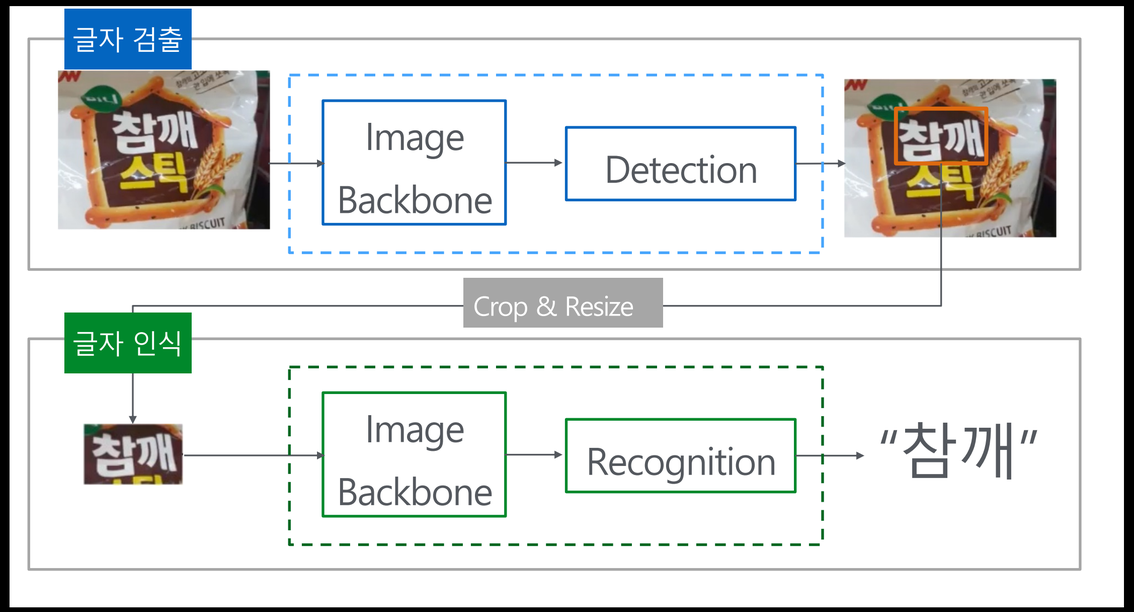
1.2 현재 OCR 모델 구성 (검출 + 인식)

NAVER DEVIEW 2023

OCR Service 모델 **2 Stage** 구조

- **글자 검출**: 이미지에서 글자에 해당하는 영역을 검출
- 검출된 영역을 Crop & Resize하여 Recognizer로 전달
- **글자 인식**: Recognizer에서 해당 글자를 인식

독립된 2개의 모델 (검출기 + 인식기)

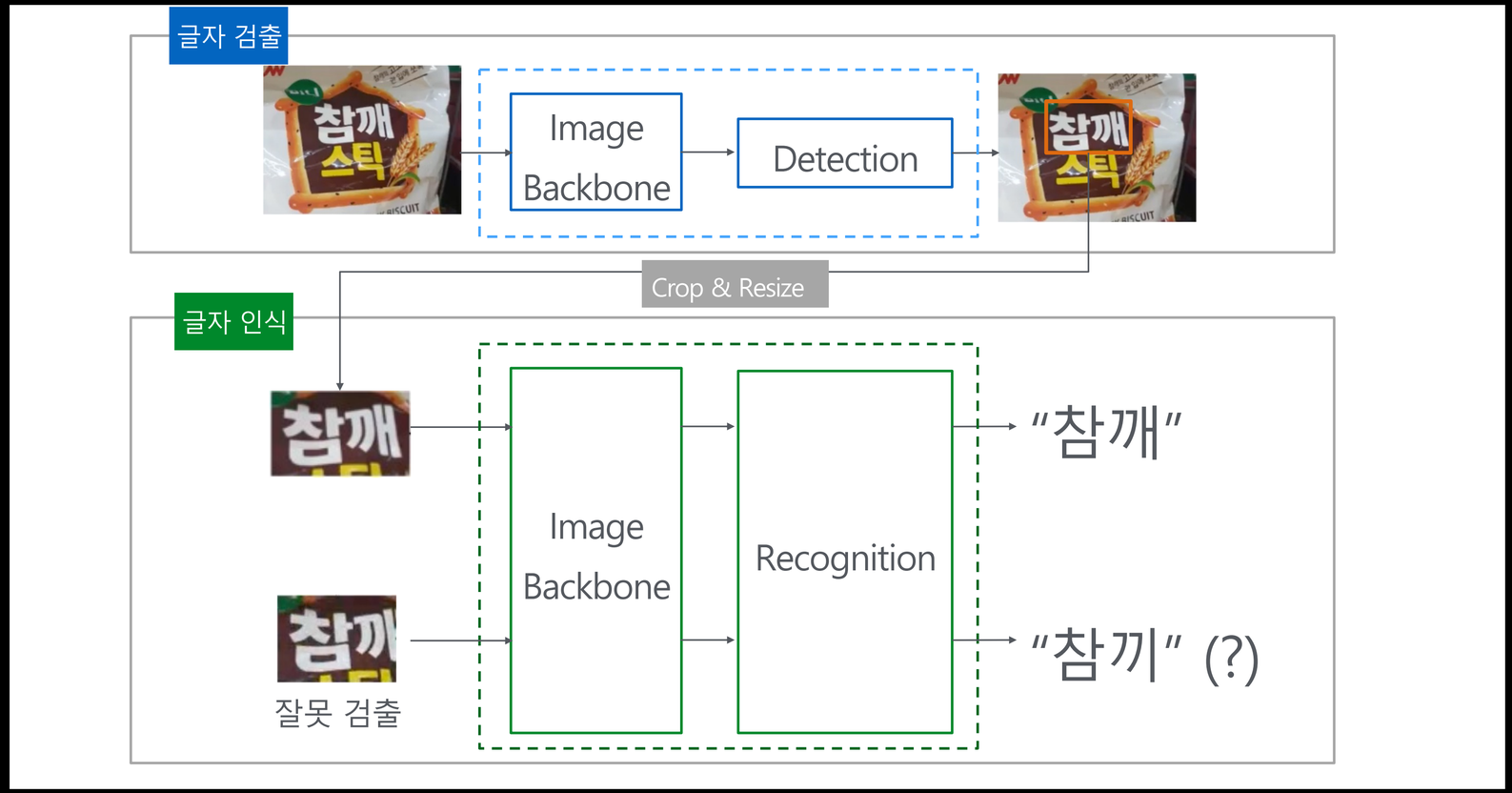


1.3 기존 모델에서 발견한 문제점

NAVER
DEVIEW
2023

2 Stage 모델의 문제점 1: 검출이 실수하면 인식 또한 실패

- Recognizer가 Crop 된 이미지가 아닌 **이미지 전체를 보면** 맞출 수 있지 않을까?



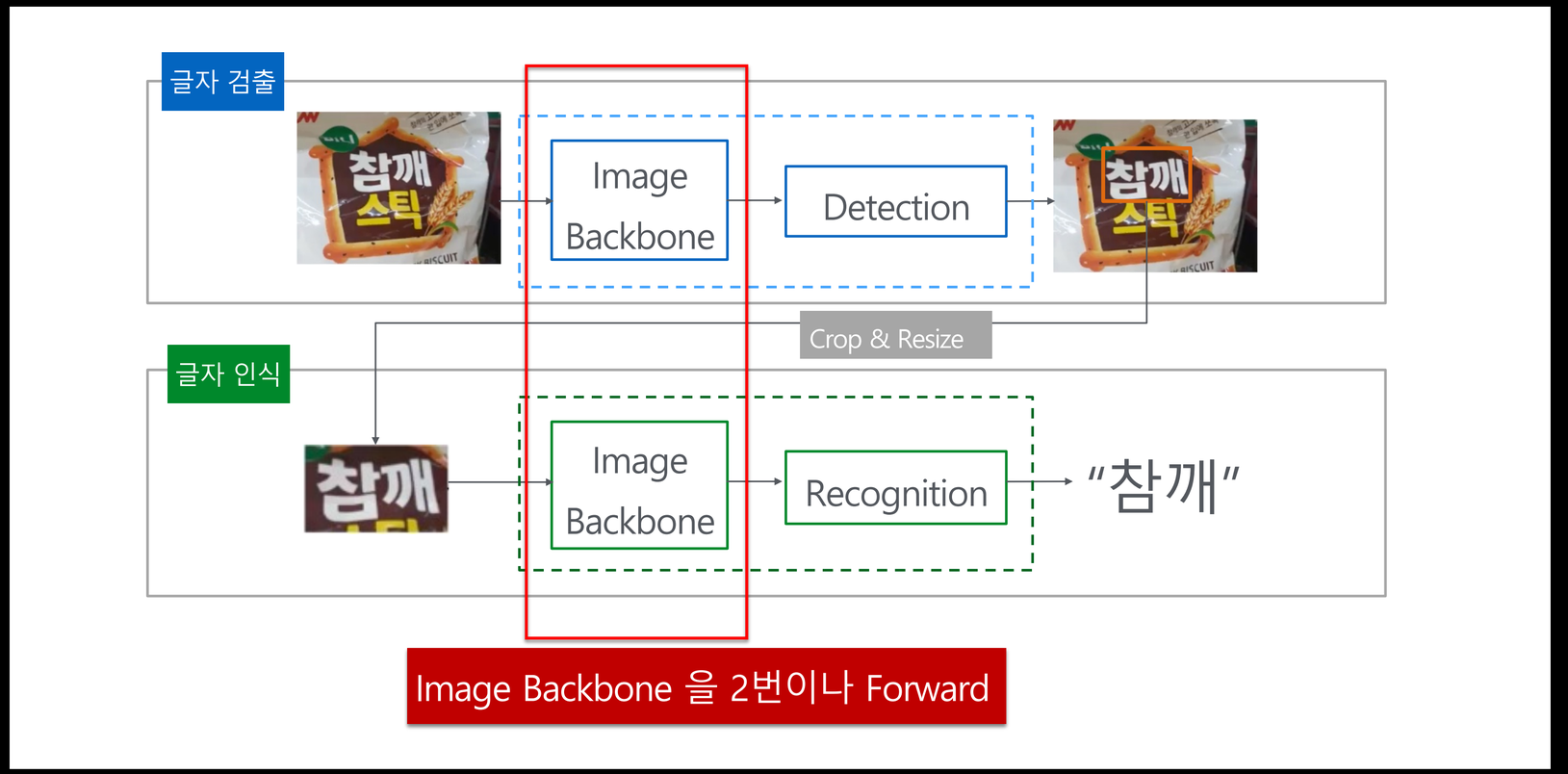
Crop & Resize 의 근본적인 문제

1.3 기존 모델에서 발견한 문제점

NAVER
DEVIEW
2023

2 Stage 모델의 문제점 2: 2개의 Image Backbone

- Image Backbone을 1개로 공유할 수 없을까?



1.3 기존 모델에서 발견한 문제점

NAVER
DEVIEW
2023

2 Stage 모델의 문제점 3: 검출기를 update하면 인식기도 새로 학습

- 서비스 유지 보수 측면에서 어려움
- 2개의 모델을 한번에 학습할 수 없을까?

1.3 기존 모델에서 발견한 문제점

NAVER
DEVIEW
2023

2 Stage 모델의 문제점

- 문제점 1: 검출이 실수에 강인하지 못함 (Crop & Resize)
- 문제점 2: 2개의 Image Backbone
- 문제점 3: 검출기 update 시 인식기도 맞추어 다시 학습해야 함

이러한 문제점들을 개선해보자 !

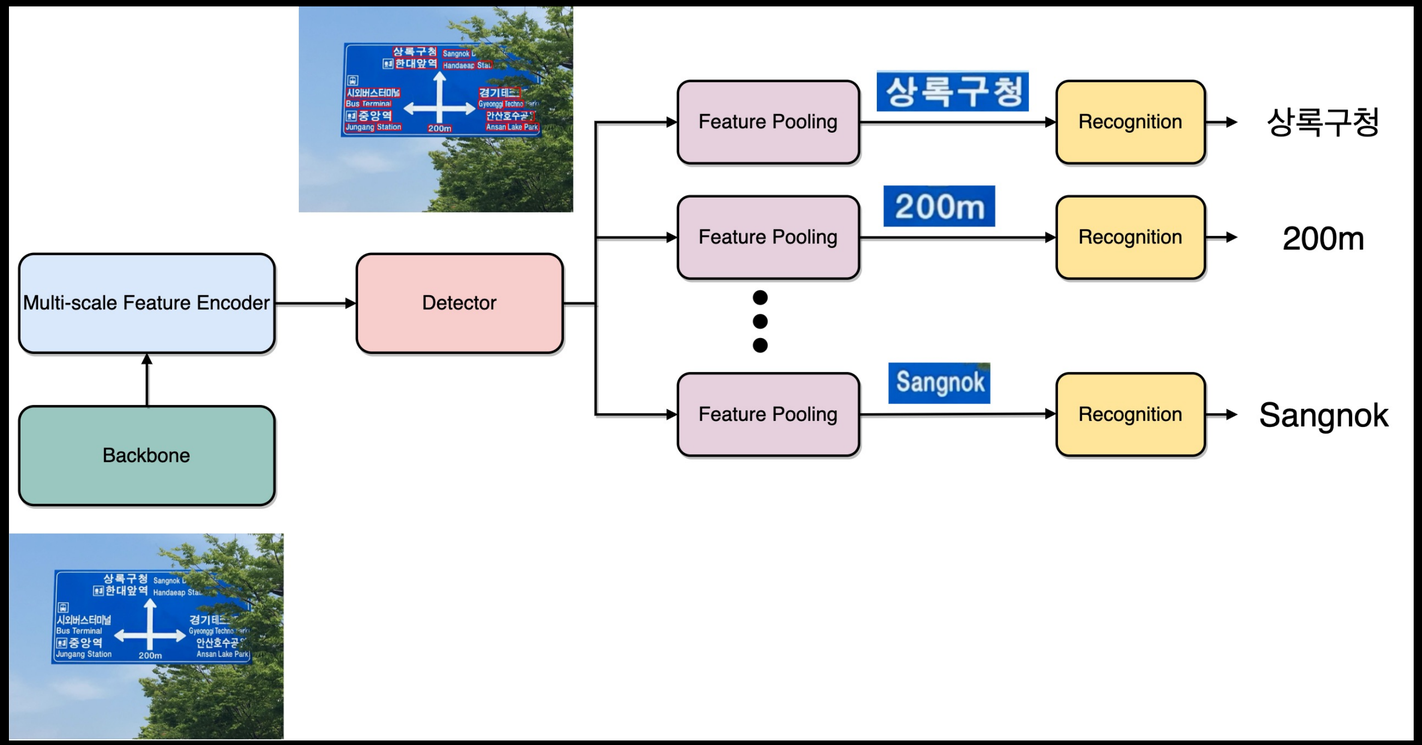
2. 글자 탐지와 인식이 한번에 된다고?

2.1 E2E OCR의 장점

NAVER
DEVIEW
2023

일반적인 End-to-End OCR 방법론

- 검출 + 각각의 Text에 대하여 Feature Pooling (예: RoI Pooling) + 인식

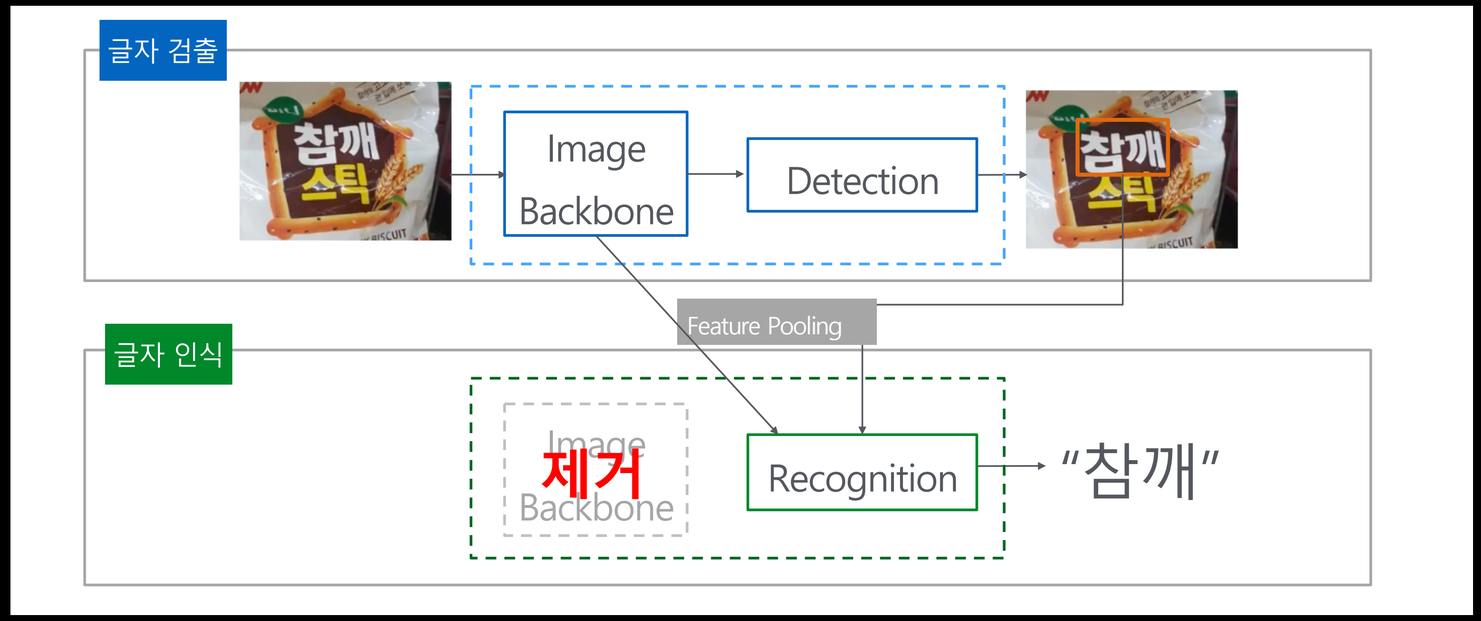


2.1 E2E OCR의 장점

NAVER
DEVIEW
2023

2 Stage 방법과 비교

- 검출과 인식을 한번에 학습 → 더 쉬운 유지 보수 (모델 버전 update 측면)
- **Image Backbone을 공유** → 2번의 Image Backbone Forward를 거칠필요가 없음



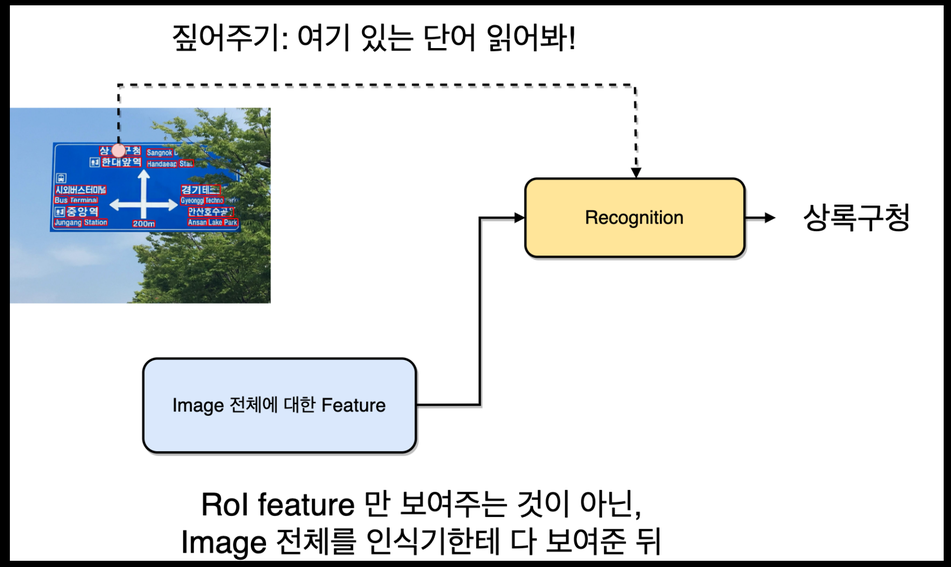
2.2 DEER의 장점

NAVER
DEVIEW
2023

DEER와 다른 E2E OCR 방법들 비교

- 기존: 어떤 단어를 인식? Feature Pooling 활용
  → 검출 실수가 나면 인식 Error
- DEER: 단순히 위치를 **짚어주기**
  → Recognizer가 RoI Feature가 아닌
  → Image 전체와 짚어준 점을 보게 됨
  → 검출 실수에 Robust

기존 E2E OCR 방법



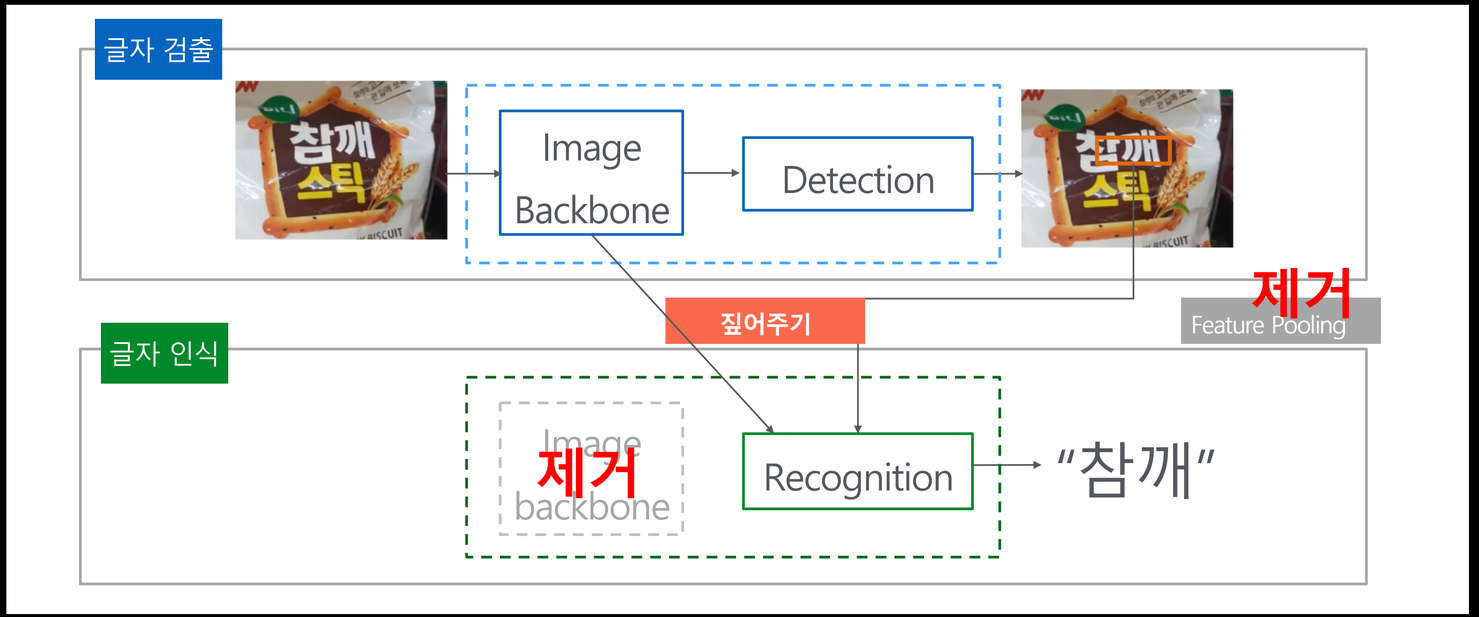
DEER

2.2 DEER의 장점

NAVER
DEVIEW
2023

DEER: 위치 짚어주기

- Recognizer의 입력: Image Feature 전체 + Text의 Location (짚어주기)
- Recognizer는 사소한 검출 실수에도 강인



2.3 DEER 뜯어보기

NAVER DEVIEW 2023

DEER: Detection-agnostic End-to-End Recognizer

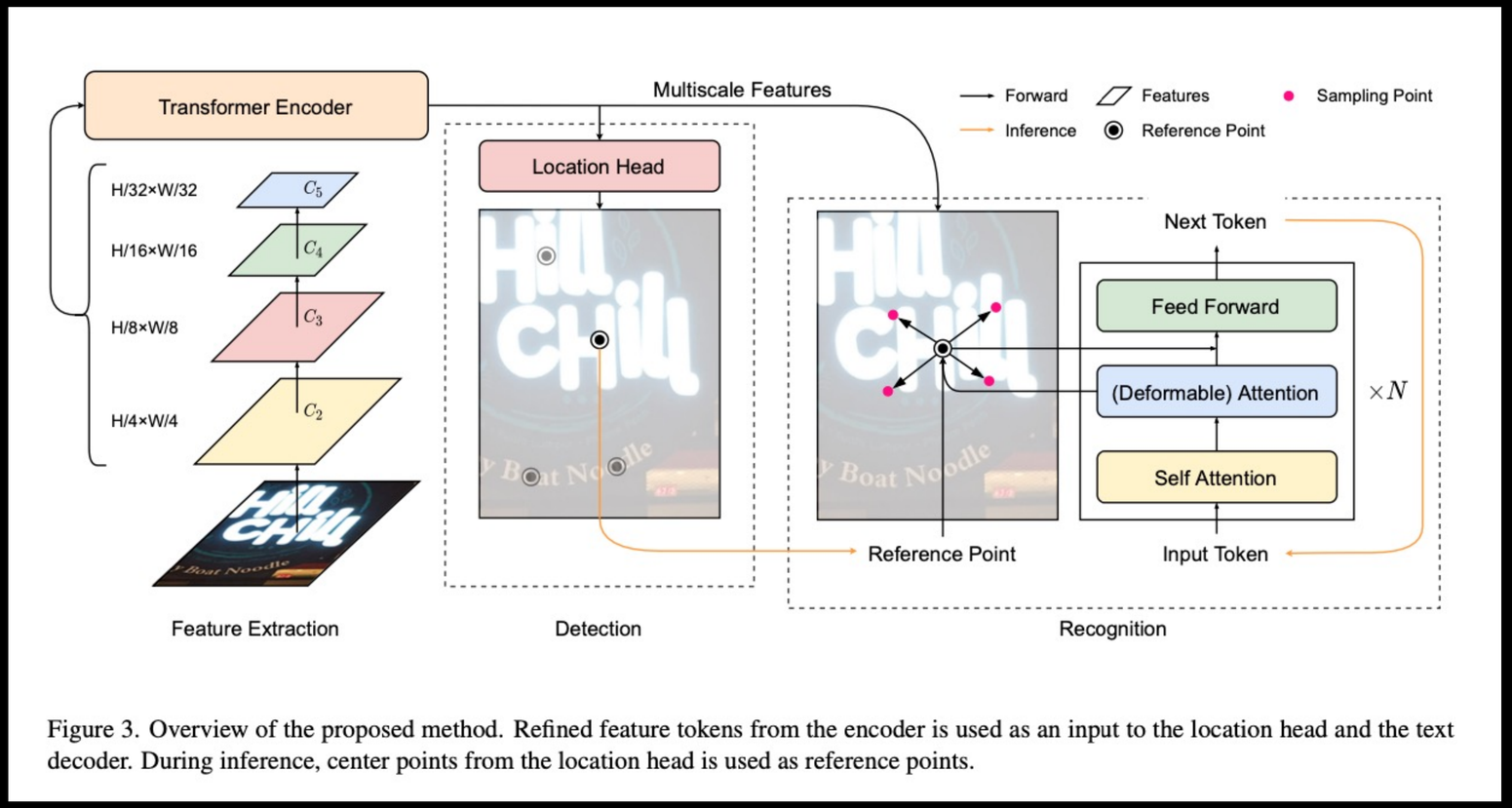


Figure 3. Overview of the proposed method. Refined feature tokens from the encoder is used as an input to the location head and the text decoder. During inference, center points from the location head is used as reference points.

- Backbone
- Transformer Encoder
- Location Head
- Text Decoder

** DEER : https://arxiv.org/abs/2203.05122*

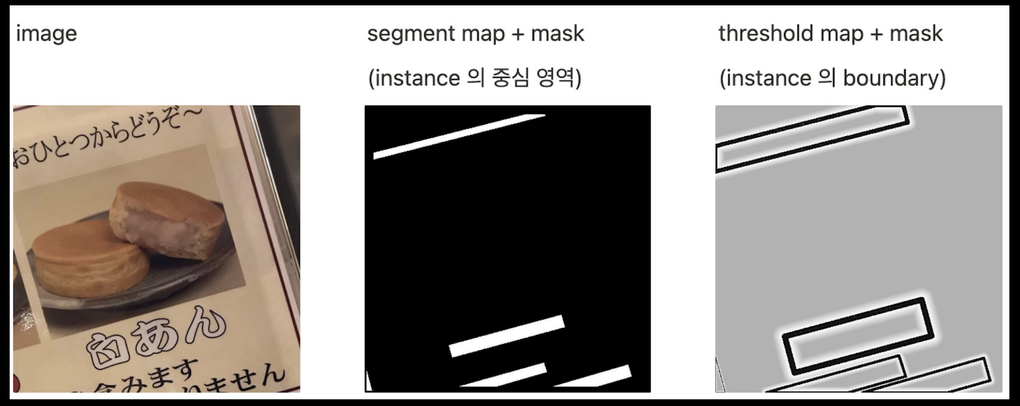
2.3 DEER 뜯어보기

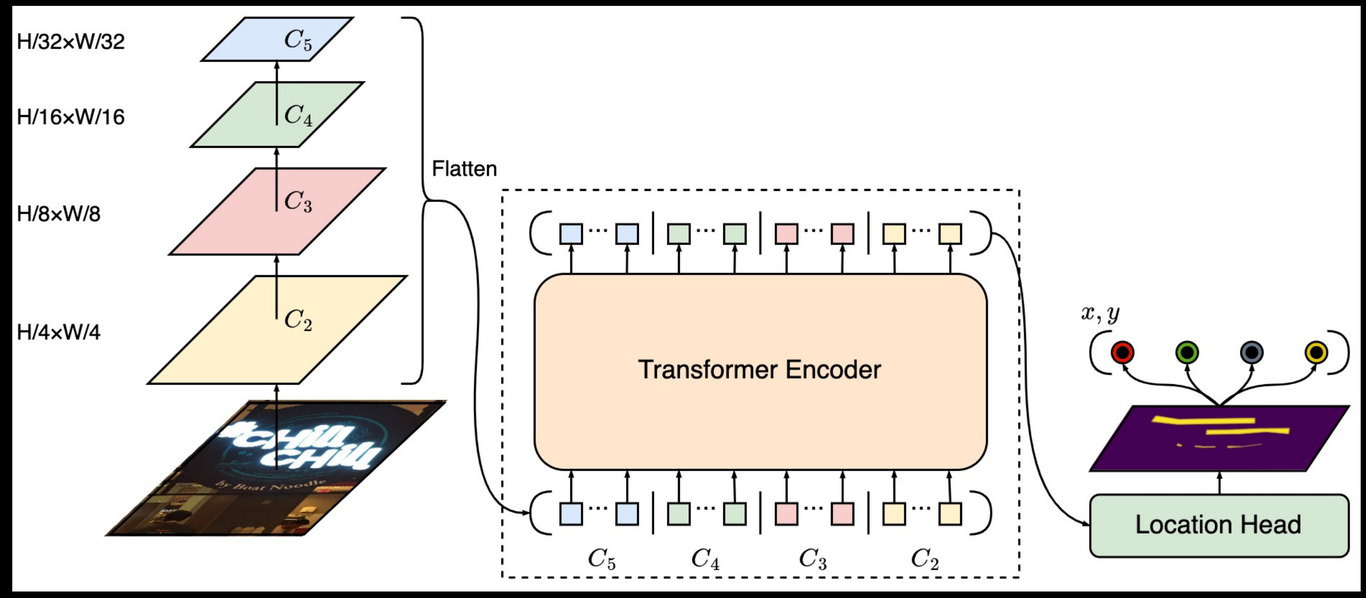
NAVER
DEVIEW
2023

Location Head

- Multi-scale Feature로부터 Text 검출
- DEER의 특성상 여러 검출기 모두 활용 가능

  → Differentiable Binarization 활용





** DB : https://arxiv.org/abs/1911.08947*

2.3 DEER 뜯어보기

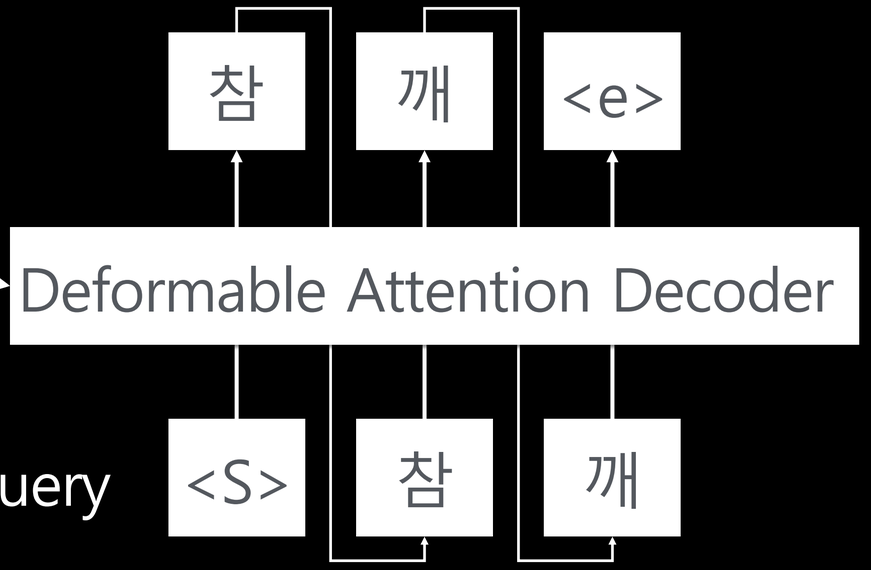
NAVER DEVIEW 2023

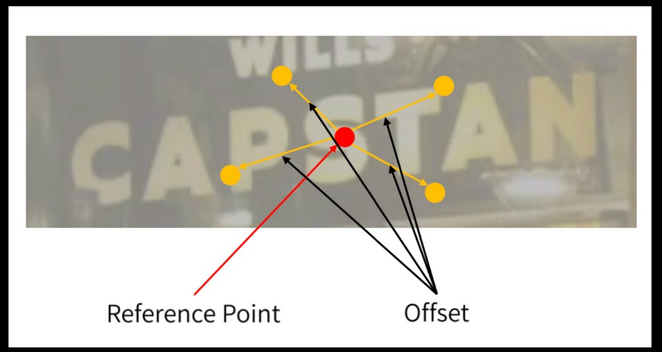
Text Decoder

- 각 Text를 Decoding (Auto-regressive 하게)
- 입력: DEER가 짚어준 위치 + Whole Image Feature (**not Pooling !**)
- Deformable Transformer Decoder 활용

Image 전체가 아닌 reference point 근처의 몇몇 point 만 활용하여 attention

Key, Value





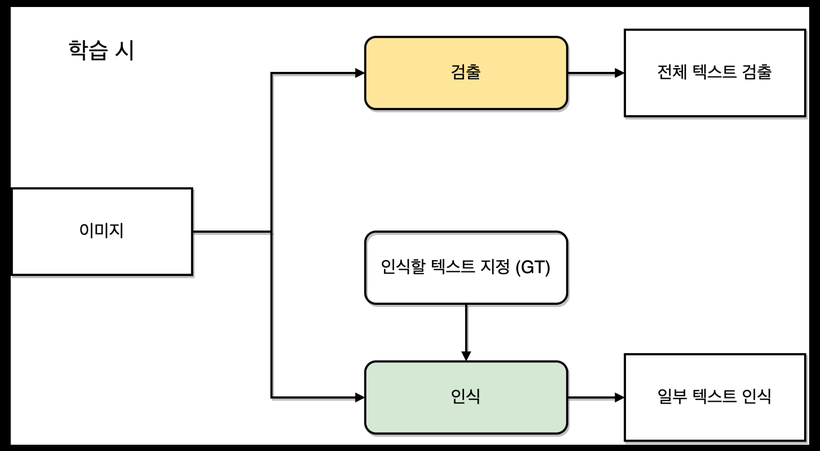
$$\text{DeformAttn}_h(A_{hqk}, p_{\text{ref}}, \Delta p_{hqk}) = W_h^o \left[\sum_{k=1}^{K} A_{hqk} \cdot W_h^k x(v, p_{\text{ref}} + \Delta p_{hqk}) \right],$$

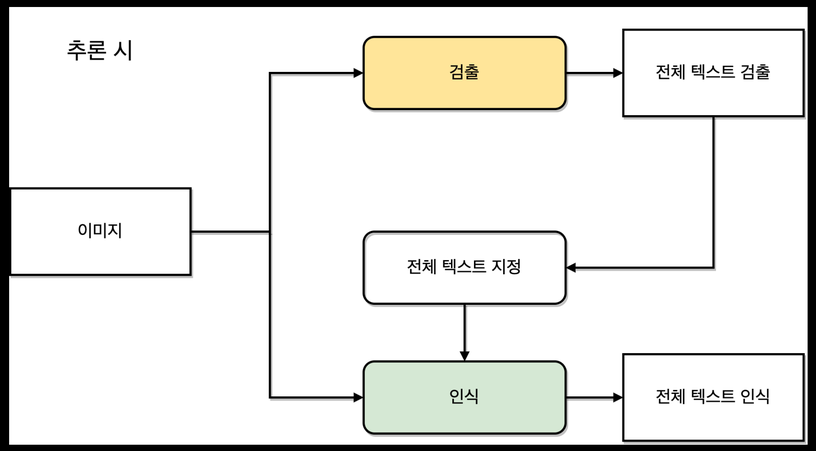
2.3 DEER 뜯어보기

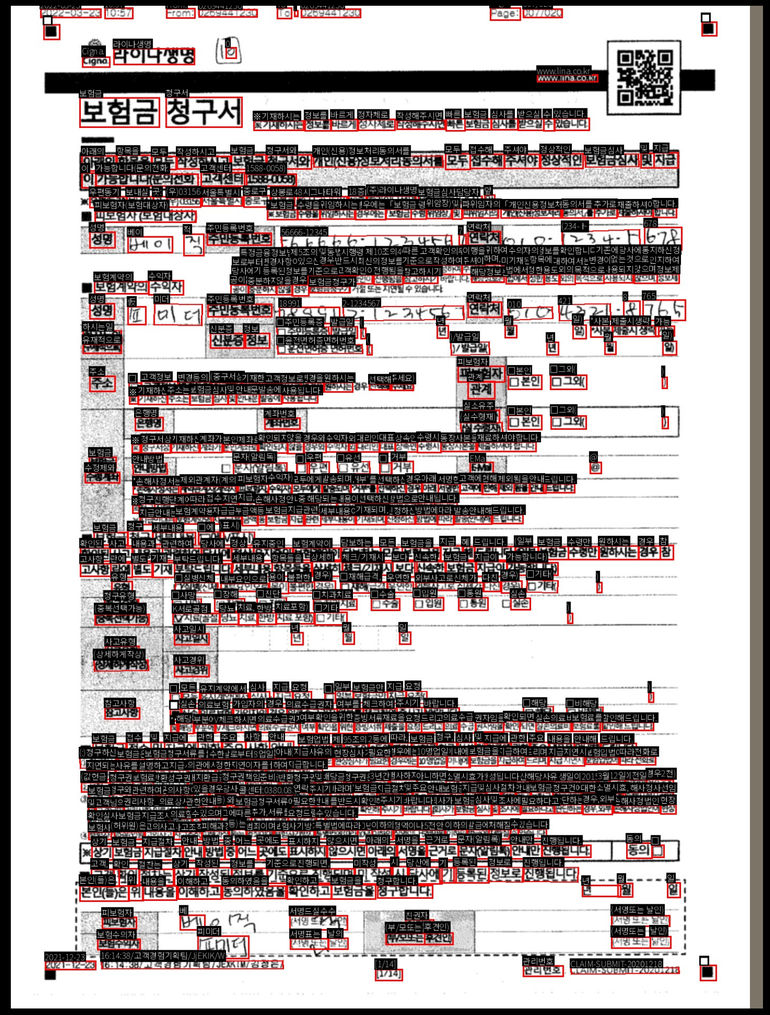
NAVER DEVIEW 2023

핵심 Idea

- OCR 분야에서 검출과 인식을 학습하기
  → 전체 Text 검출 학습: 충분히 가능
  → 전체 Text 인식 학습: 어려운 문제
- 해결책: 학습 시 샘플링한 대상만 인식 (짚어주기를 통해서)





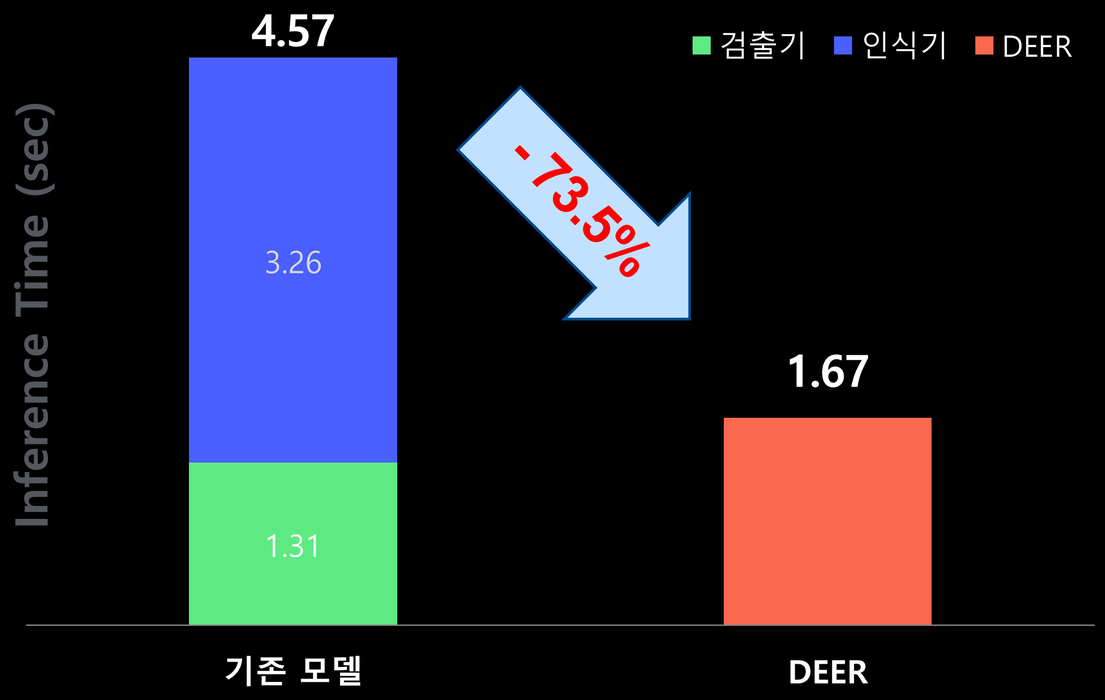


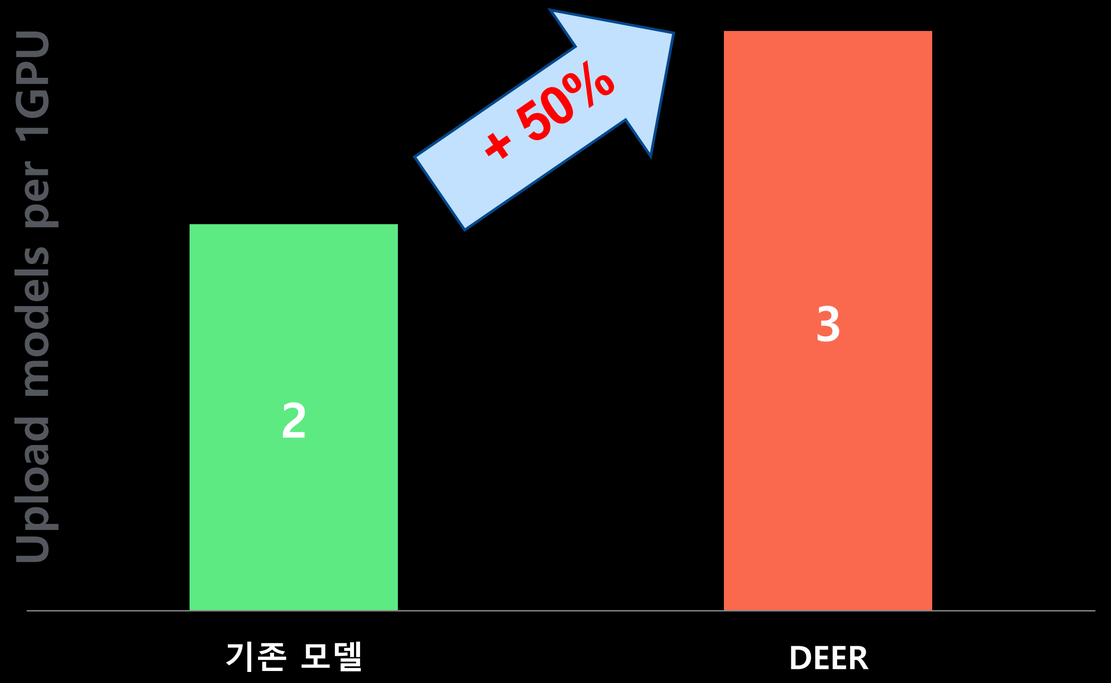
2.4 얼마나 빠르고 가볍다구요

NAVER
DEVIEW
2023

기존 모델 대비 Inference Time 및 Memory 비교

- 기존 대비 **Inference Time은 73% 줄었으며**, **GPU당 model은 50%** 더 들어갈 수 있음





2.5 척하면 척 잘 맞춰요

NAVER
DEVIEW
2023

검출 실수에 강인한 DEER

2.5 척하면 척 잘 맞춰요

NAVER
DEVIEW
2023

Rotation에 강인한 DEER

3. 새 모델은 이것도 할 수 있어요

3.1 Character Detection

NAVER
DEVIEW
2023

단어보다 자세하게: 문자 단위 검출

- 단어 단위 검출 뿐만 아니라 **문자 단위 검출** 기능도 추가
- 문제 해결
  - → **Model 관점**
  - → **Data 관점**

3.1 Character Detection

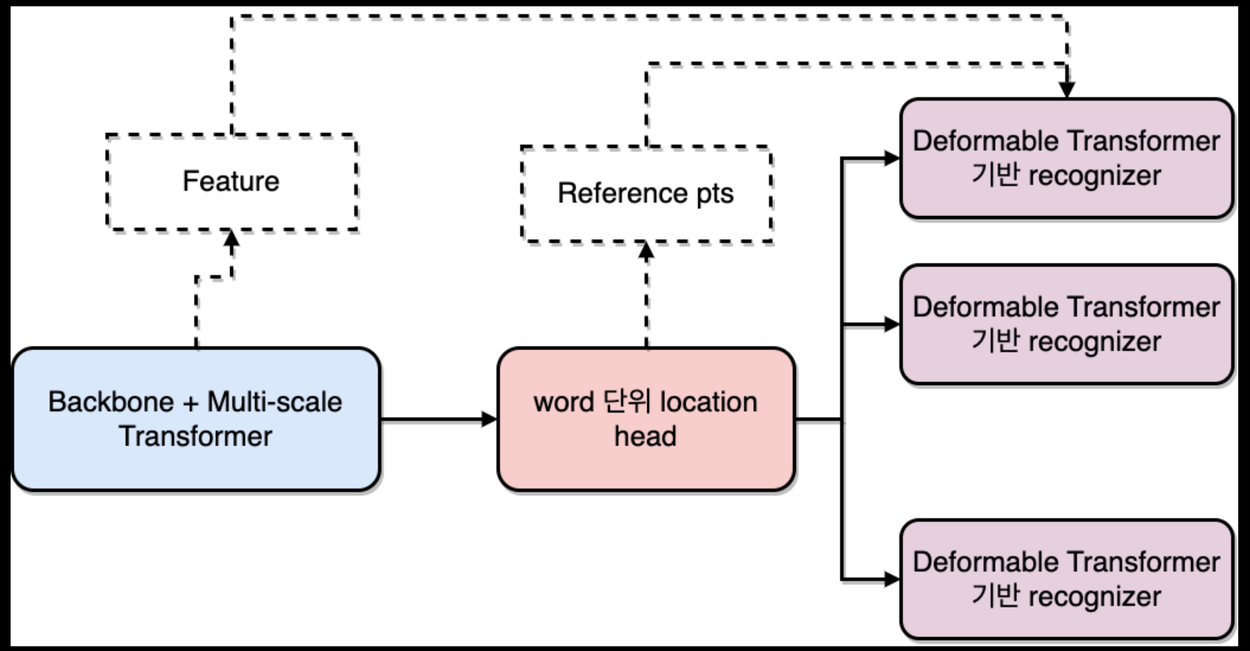
NAVER
DEVIEW
2023

Model 관점

- 기존 모델의 어느 곳에 character detector 를 추가하면 좋을까?
- 조건

  → 기존 ocr 모델 성능은 그대로면서

  → 아주 작은 연산량 추가만으로 해결

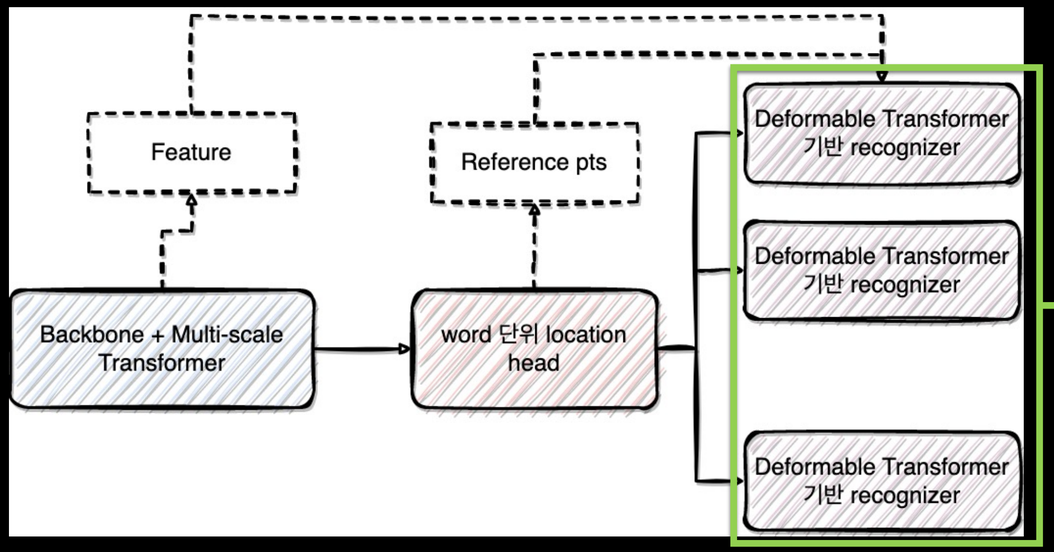


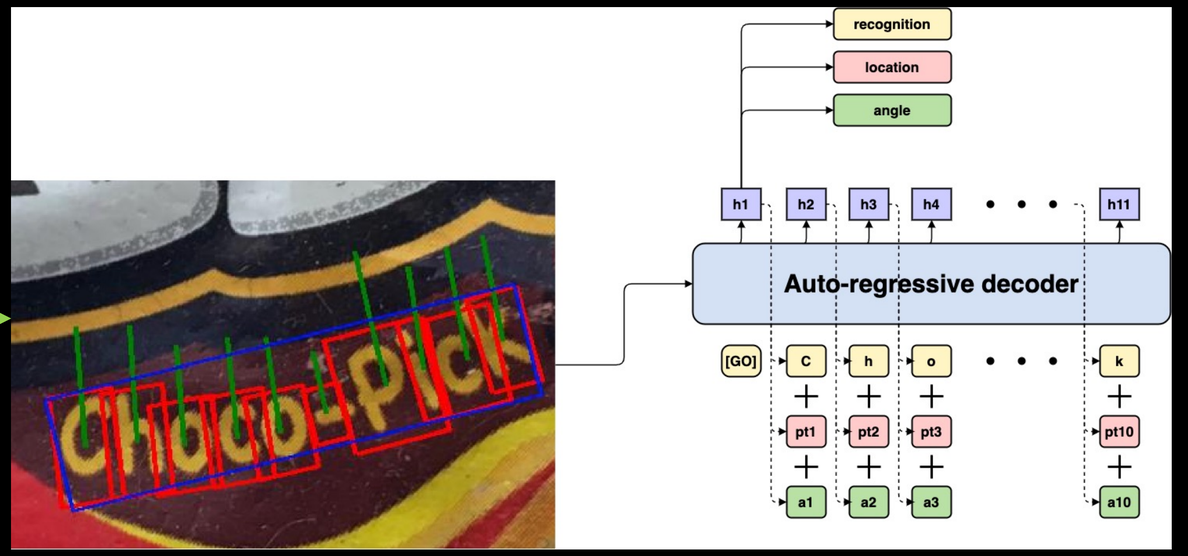
3.1 Character Detection

NAVER
DEVIEW
2023

기존 Recognizer에 기능 추가

- Recognizer가 인식 뿐만 아니라 char 위치까지 검출
- 장점: Recognizer 끝단에 Regressor 만 추가 → **가벼움**



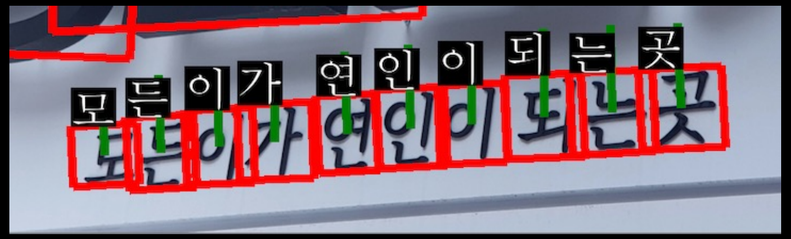


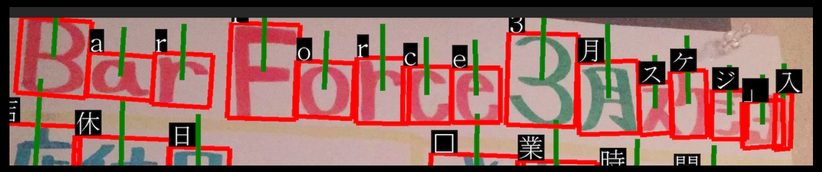
3.1 Character Detection

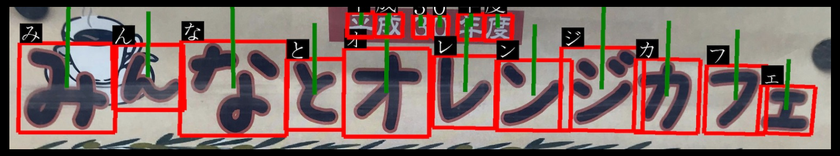
NAVER
DEVIEW
2023

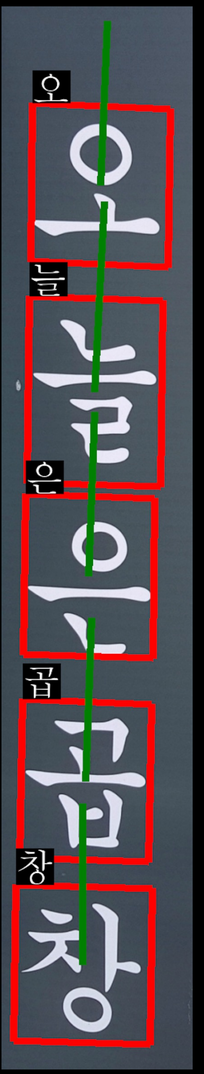
기존 Recognizer에 기능 추가

- Recognizer가 char 검출과 인식을 동시에 수행함 확인









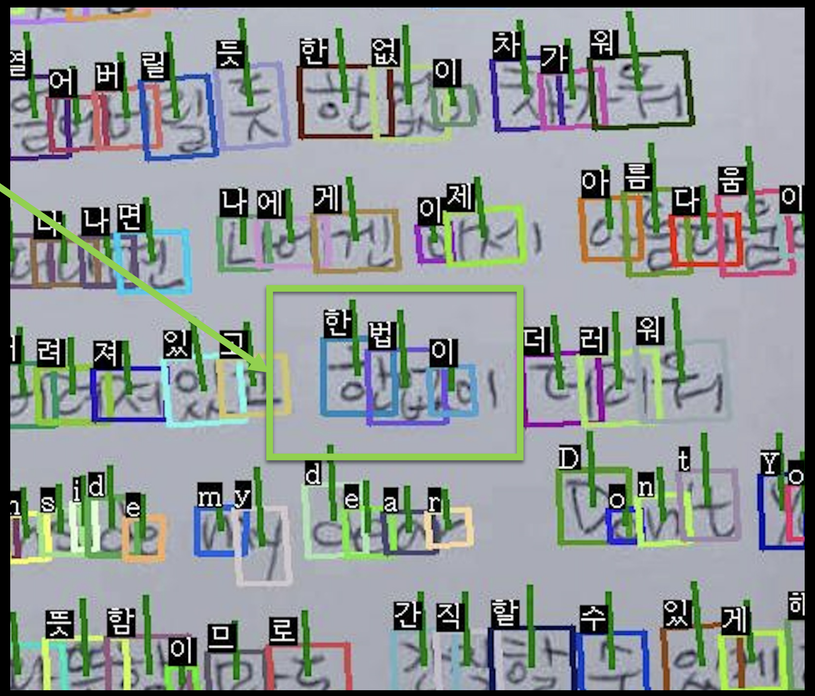
3.1 Character Detection

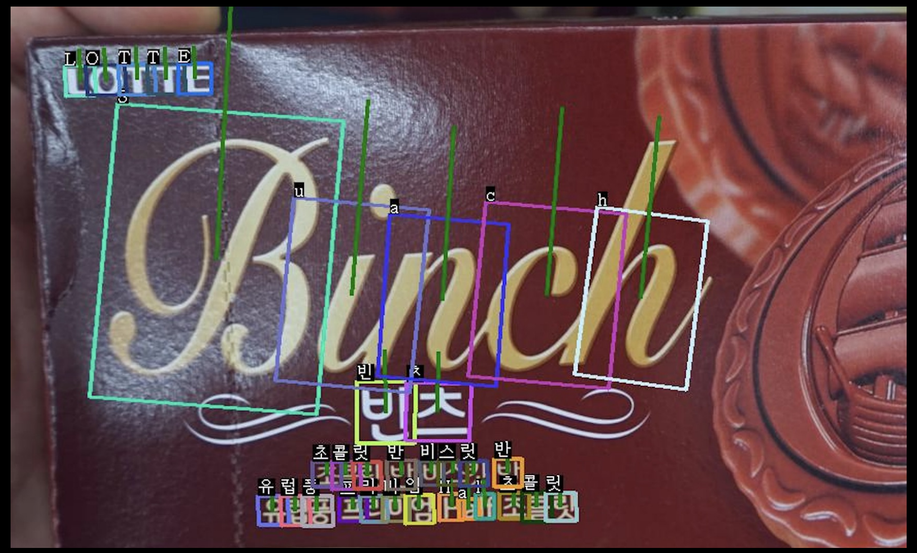
NAVER
DEVIEW
2023

기존 Recognizer에 기능 추가

- 인식 성능과 char 검출 성능은 밀접한 관계가 있음
- Recognition의 성능이 올라가면? → char 검출의 성능까지 따라 올라올 것임

한없이 → 한법이





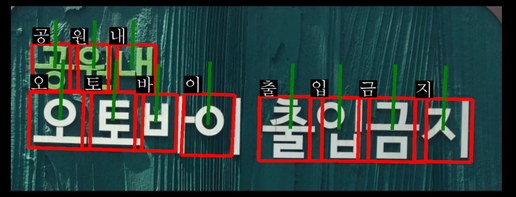
Binch → Buach

3.1 Character Detection

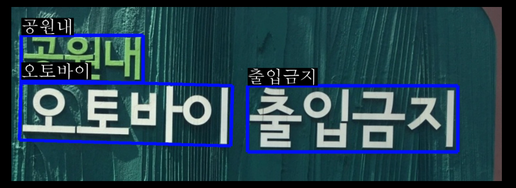
NAVER DEVIEW 2023

Data 문제

- Char 위치에 대한 annotation은 cost가 매우 비싸다
- 가지고 있는 모든 dataset에 대해서 char annotation을 갖고 있진 않음
- **5%** 의 word만 char annotation을 갖고 있음
- Semi-supervised Problem



Labeled data



Weakly-labeled data

(a) Rectangle (55s). (b) Quadrilateral (96s). (c) Character (581s). (d) Polygon (172s).

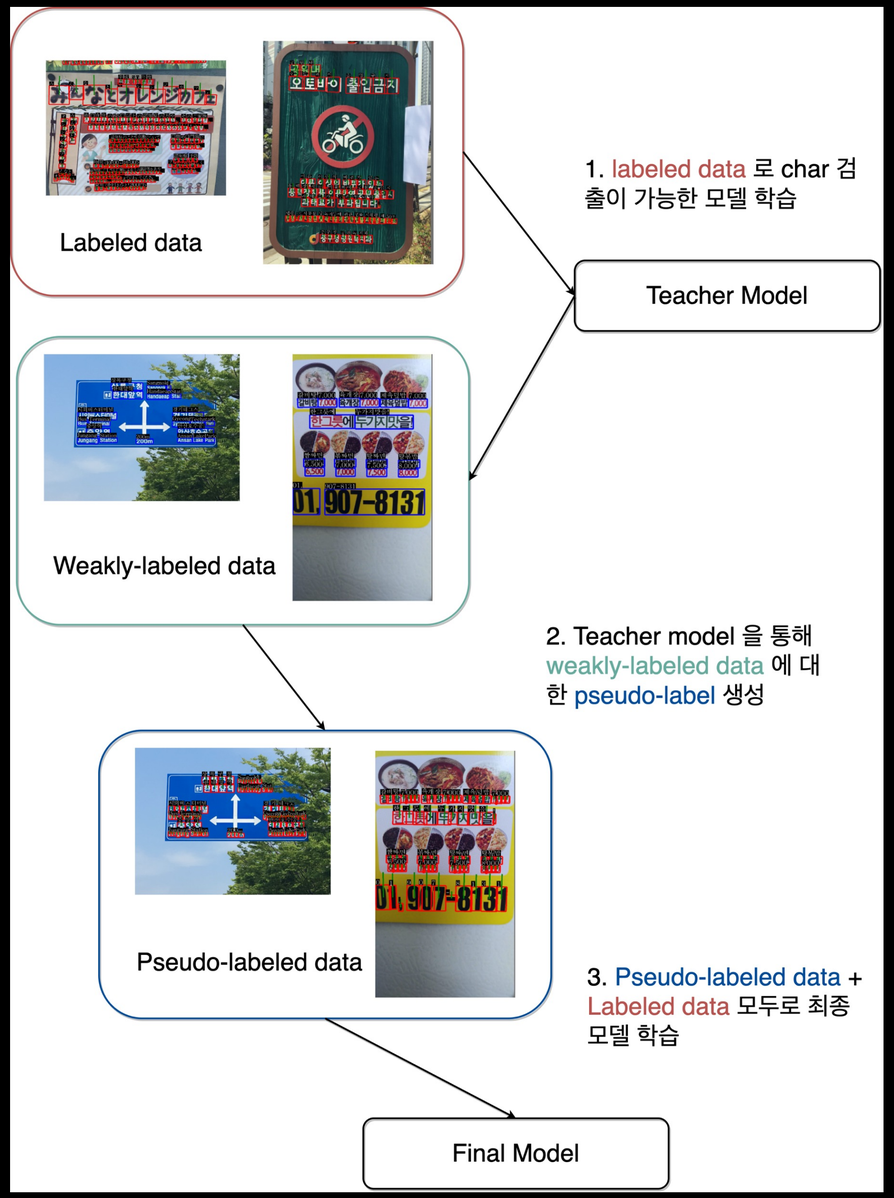
"SPTS: Single-Point Text Spotting", https://arxiv.org/abs/2112.07917

3.1 Character Detection

NAVER
DEVIEW
2023

Pseudolabeling 활용

- **Labeled data**로 teacher model 학습
- Teacher model은 **weakly-labeled data**에 대하여 **pseudo-label** 생성
- 최종 모델은 labeled data와 weakly-labeled data를 모두 활용하여 학습
- Teacher model로는 Differentiable Binarization 활용 → char 검출만이 목적



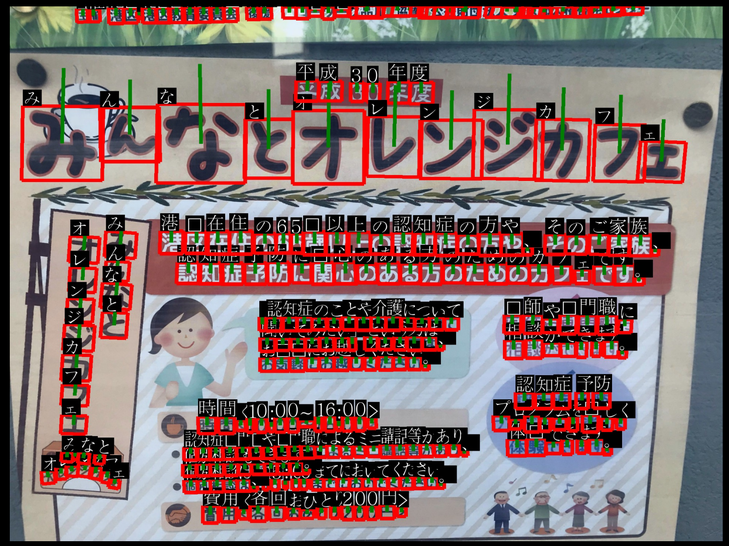
3.1 Character Detection

NAVER DEVIEW 2023

최종 결과

	AP@50
Teacher (DB)	83.4
Student (DEER)	**86.6**

- 최종 모델은 ocr과 char 검출 모두 수행가능
- Char 검출 목적의 teacher model보다도 char 검출 성능도 뛰어남

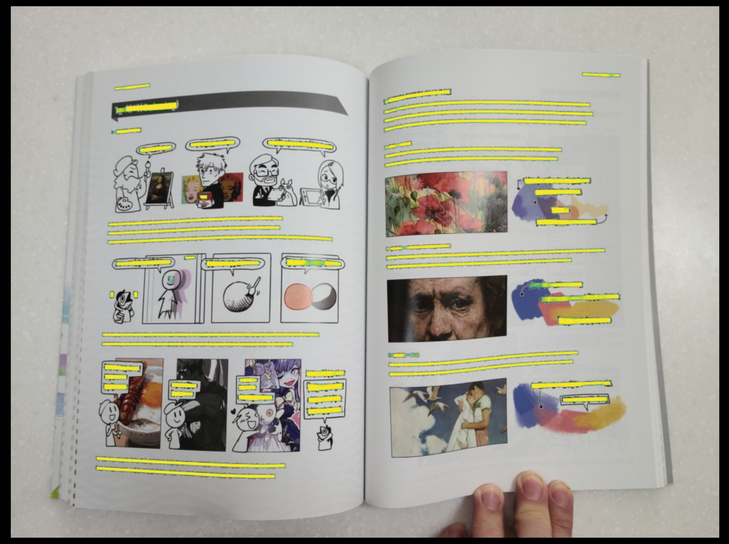


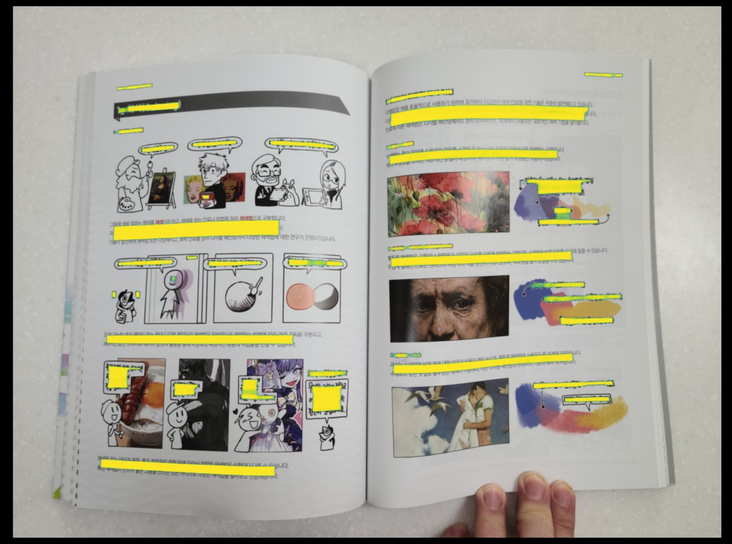
3.2 문서 구조 추출

NAVER
DEVIEW
2023

단어보다 큰 단위: **line 과 paragraph 검출**

- 문서의 구조를 이해하는 하나의 과정
- OCR이 문서에 대해 주로 사용되다보니깐
- 관련해서 고객들의 니즈 존재



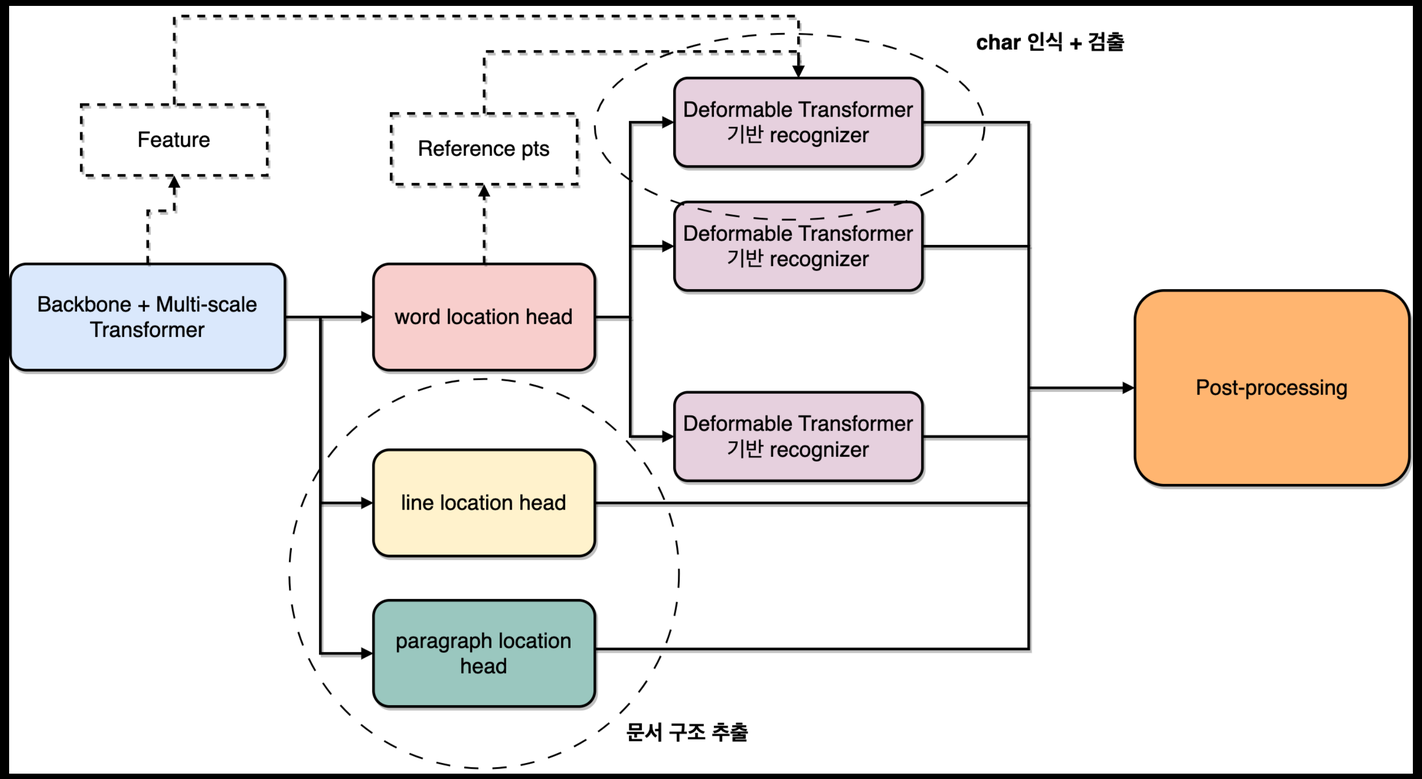


3.2 문서 구조 추출

NAVER
DEVIEW
2023

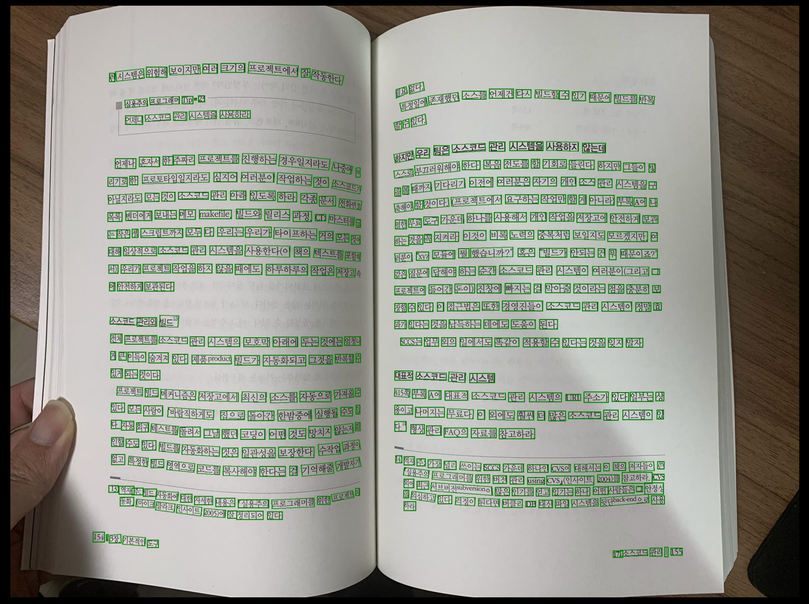
모델 구조

- Line/paragraph에 대한 Location Head 추가 (DB 기반)

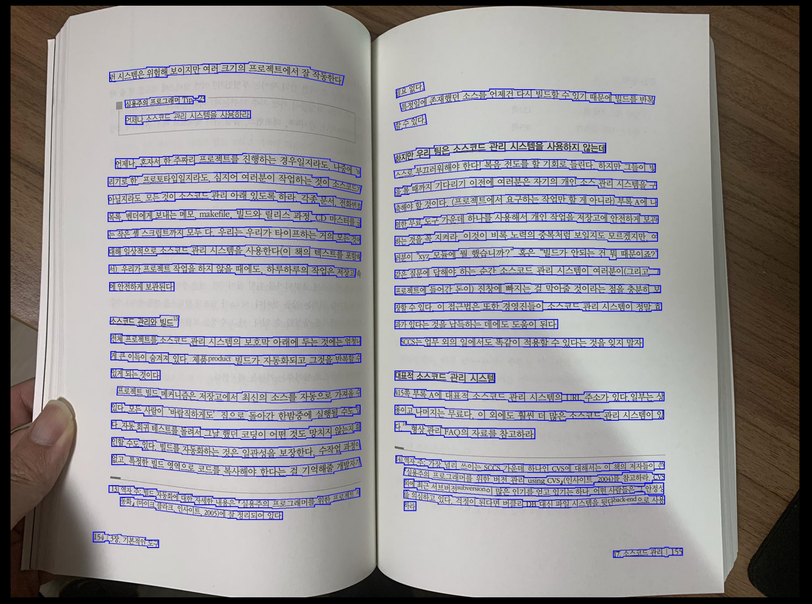


3.2 문서 구조 추출

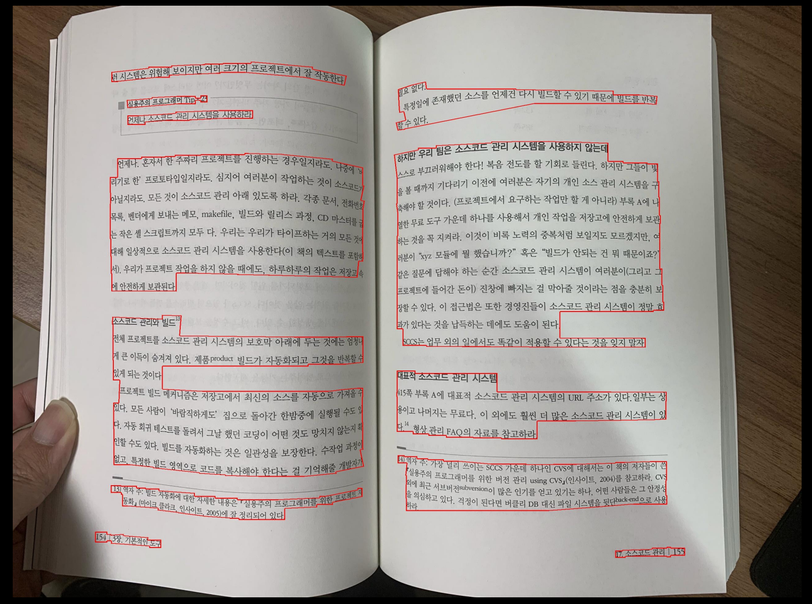
NAVER
DEVIEW
2023



단어 단위 검출



라인 단위 검출



문단 단위 검출

4. 국제 OCR challenge 1등

4.1 OOV Challenge

NAVER
DEVIEW
2023

DEER 모델이 세계적인 다른 모델과 비교해서 성능이 좋을까?

- 공식적인 대회에서 DEER 모델로 성능을 검증하자
- ECCV 2022 에서 주관한 Challenge 참여

  주제 : Out of Vocabulary Scene Text Understanding

4.1 OOV Challenge

NAVER
DEVIEW
2023

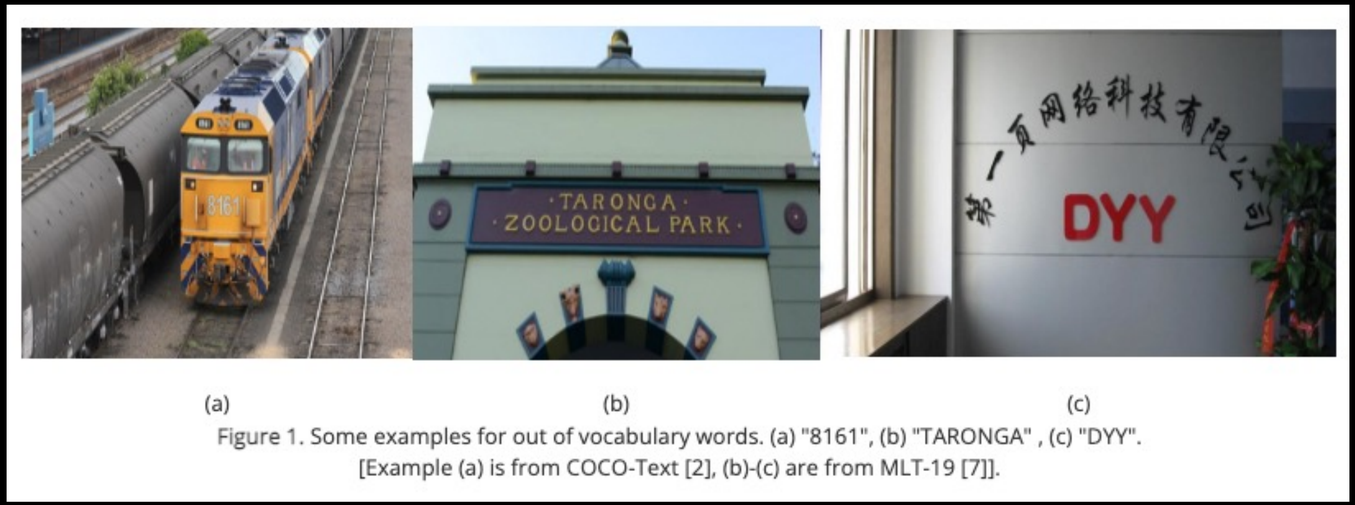
Out of Vocabulary Scene Text Understanding 란?

- Out of Vocab: 학습 때 전혀 보지 못했던 words

  → OOV 에 대한 평가 결과로 Challenge 등수를 매김

  → 언어 모델에 과도하게 의존하는 기존 benchmark 와는 차별

  → 실제 시나리오에 더 적합한 Task



(a) (b) (c)

Figure 1. Some examples for out of vocabulary words. (a) "8161", (b) "TARONGA" , (c) "DYY".
[Example (a) is from COCO-Text [2], (b)-(c) are from MLT-19 [7]].

4.2 Our Experience

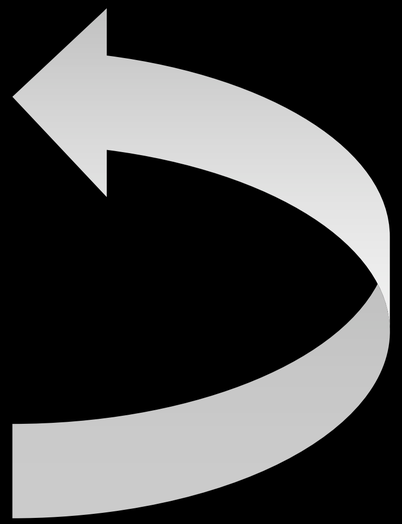
NAVER
DEVIEW
2023

Model Development

- 7개의 public data의 조합 → data 의 특성 분석
- Model 학습
- 다양한 Ablations & 정성 / 정량 평가

  → batch size, backbone, hyper-parameter

  , data 처리
- 모델 결과 분석 후 재학습

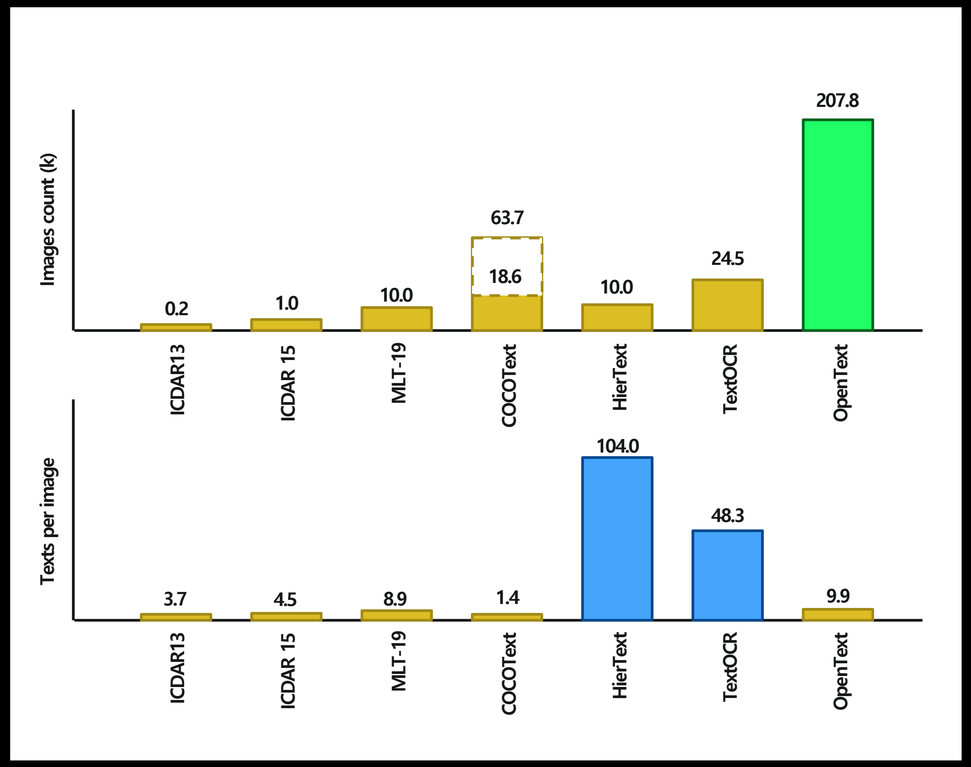


다음 과정 계속 반복

4.2 Our Experience

NAVER
DEVIEW
2023

Data Statics



- TextOCR 과 HierText 는 Image 당 텍스트의 개수가 많았음
  → **텍스트 개수에 상관없이 모델을 잘 학습해야 함**
  → **DEER 의 짚어주기와 고정수량 text 학습으로 안정적**

- Training Data Ratio
  → **Batch 구성시 Data Ratio**
  → **텍스트 개수에 비례하게 세팅**
  → **인식기가 모든 Text data를 잘 학습하도록 구성**

4.2 Our Experience

NAVER
DEVIEW
2023

Data Characteristics

MS COCO dataset 의 경우

- Box annotation
- Annotation box 가 tight 하게 설정되어 있지 않았음
- Sparse annotation
- 대소문자가 올바르게 annotation 되어 있지 않음

부정확한 검출 annotation → 검출 loss는 포함시키지 말자 = 인식 supervision 만 활용하자

Pseudo-labeling 으로 fix

DEER 모델의 특성

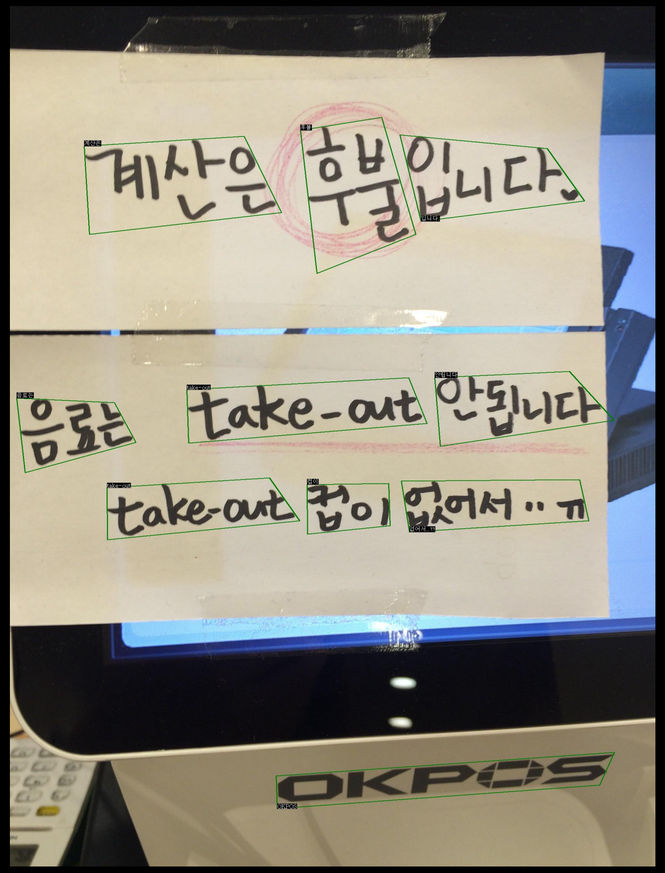
어떤 텍스트를 인식할지 모델은 Point 형태로 짚어주기 때문에 quad가 아닌 box 형태의 annotation이어도
인식 모델이 충분히 학습 가능

즉 **Noisy 한 Location Annotation 에도 인식기 학습 가능**

4.2 Our Experience

NAVER
DEVIEW
2023

다국어 관련



Option 1

- 검출기에서 다국어를 검출하지 않게 함 (오직 영어 검출기)
- 인식기는 영어만 인식 가능
  → 검출기가 외국어를 검출한 경우, 인식기는 학습시 외국어를 본적이 없기 때문에 생각치 못한 결과가 나올 수 있음

Option 2

- 검출기는 영어 + 다국어 모두 검출
- 인식기에서 영어 외 문자를 걸러낼 수 있는 능력을 학습 시킴 [unk]

Option 2 선택 !

4.2 Our Experience

NAVER
DEVIEW
2023

Ablation Results

	IV		OOV	
	init	**final**	**init**	**final**
ICDAR13	0.673	0.841	0.571	0.706
ICDAR15	0.648	0.723	0.333	0.377
MLT19	0.718	0.761	0.560	0.622
HierText	0.745	0.768	0.500	0.542
OpenImages	0.573	0.624	0.290	0.367
TextOCR	0.651	0.694	0.334	0.416
COCOText	0.233	0.404	0.131	0.259
Macro	0.668	**0.714**	0.389	**0.466**

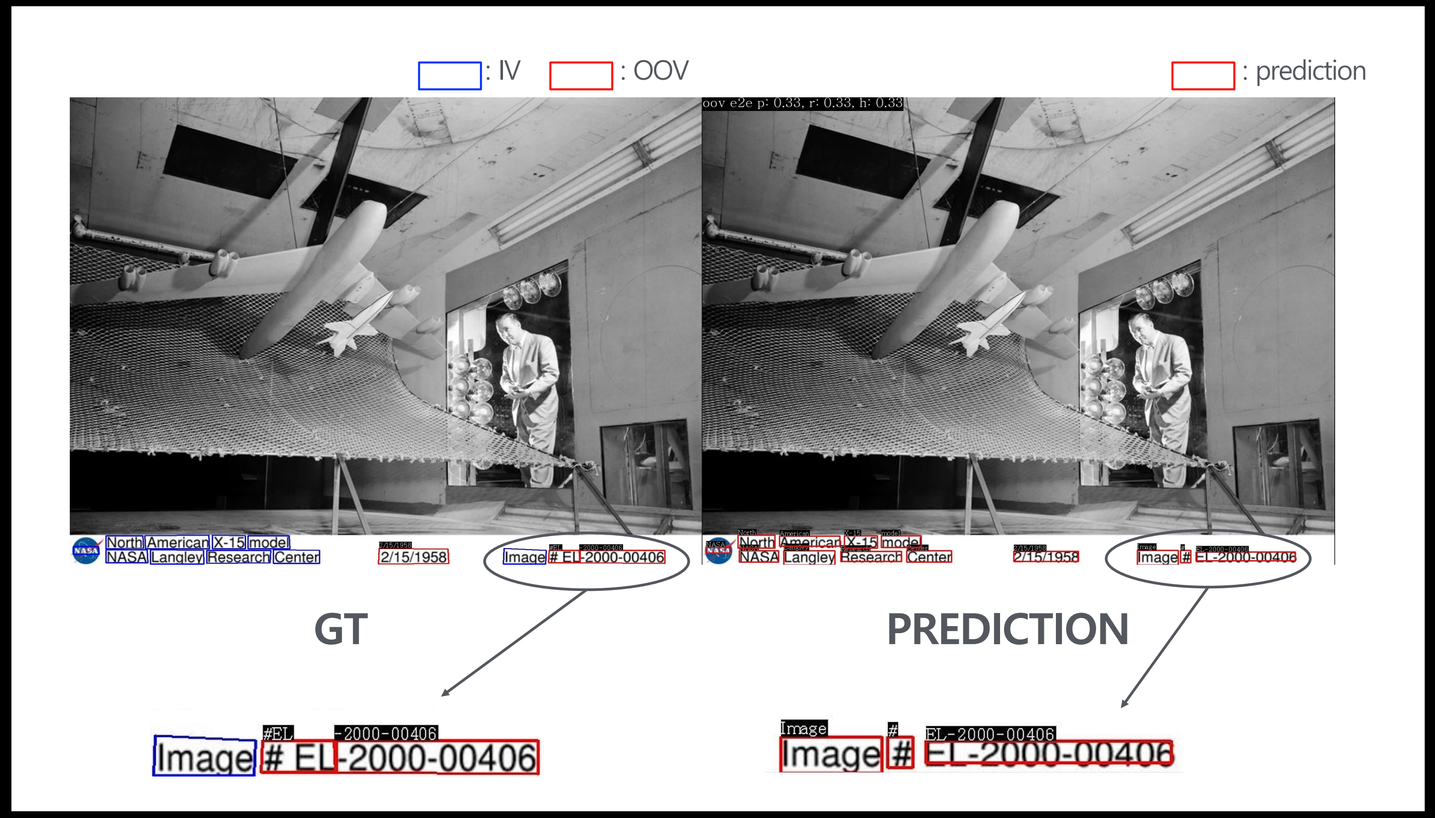
Higher Performance

- Large batch: 16 → 28
- Train resolution: 768 → 1024
- Backbone: ResNet50 → VoVNet39
- Add Synthetic Dataset
- COCO Text의 경우 Recognition Supervision만 활용하여 학습
- 영어외 문자는 인식기에서 거르게 함

4.2 Our Experience

NAVER
DEVIEW
2023

Analysis of OOV

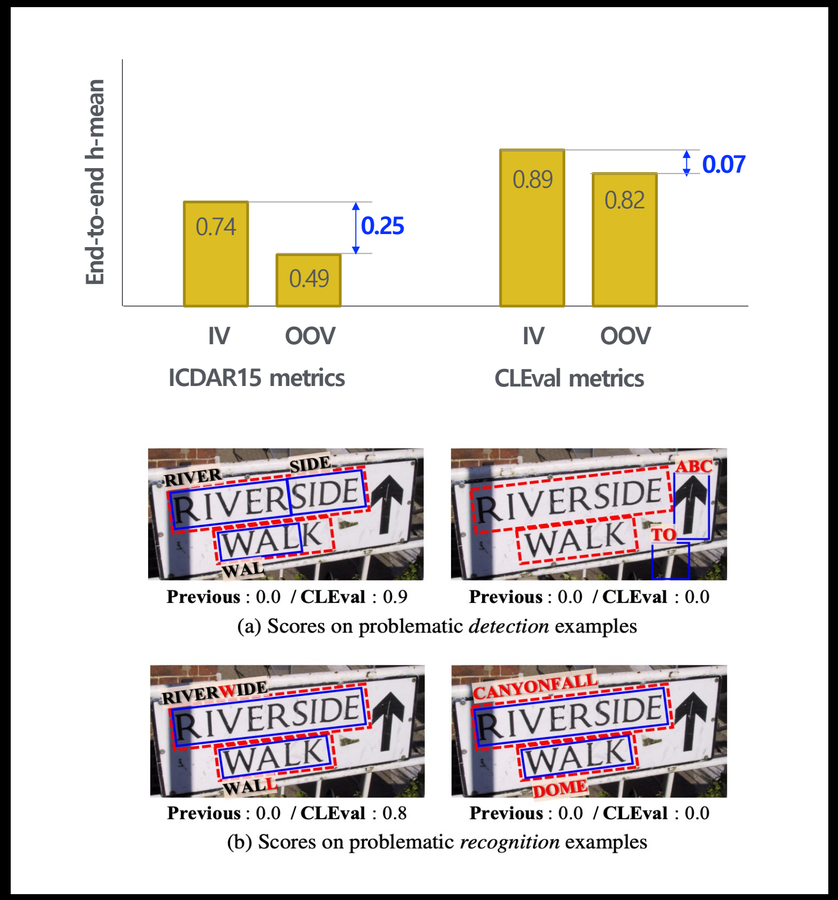


- Split/merge 에 대한 코너케이스
  → 이것들이 OOV가 되기도 함
- OOV에 대한 성능 평가를 위해서는
  Split/merge Case에 대한 Soft_Measure 필요

4.2 Our Experience

NAVER
DEVIEW
2023

Other metric: CLEval Metrics



- CLEval Metrics은 Character-Level Evaluation Metric으로 Split/Merge Cases에 대한 Penalty가 soft 하게 반영된 Metric
- CLEval metric에서, IV and OOV의 차이는 감소했음
- 제안한 모델이 IV에 비해 OOV를 어려워한다고 보기 힘듬

4.2 Our Experience

NAVER
DEVIEW
2023

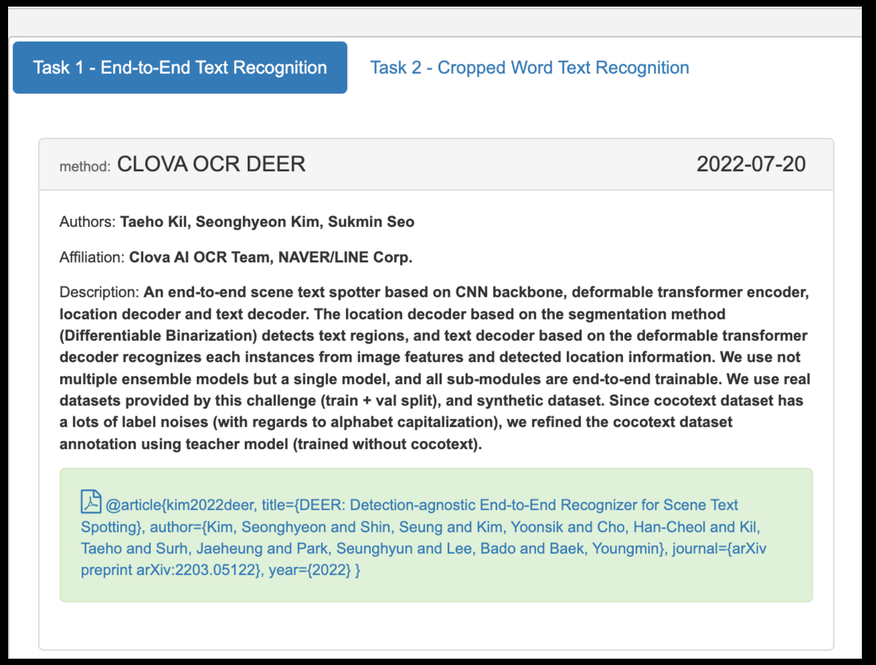
후기

- Clova E2E OCR 모델이 세계적으로 경쟁력있음 확인

  → 다른팀들은 2 Stage (검출 + 인식) 모델에 대한 Ensemble을 주로 활용

  → 우리는 Single Model에 대한 결과를 제출

- **Challenge 우승 !**



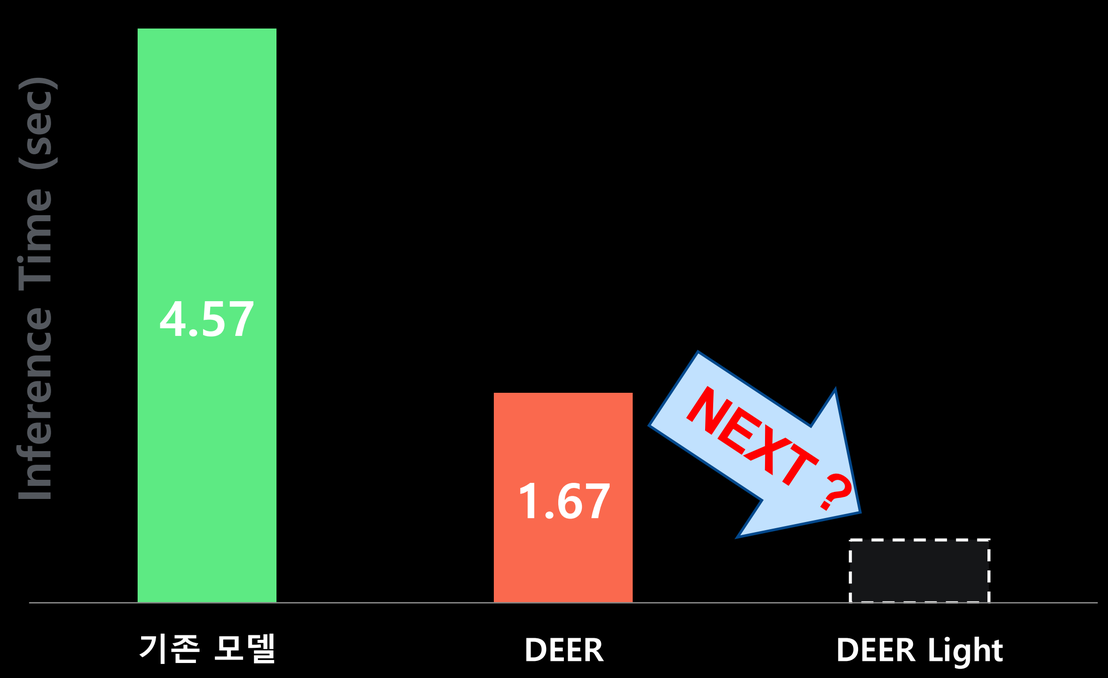
5. 서비스 배포를 위해

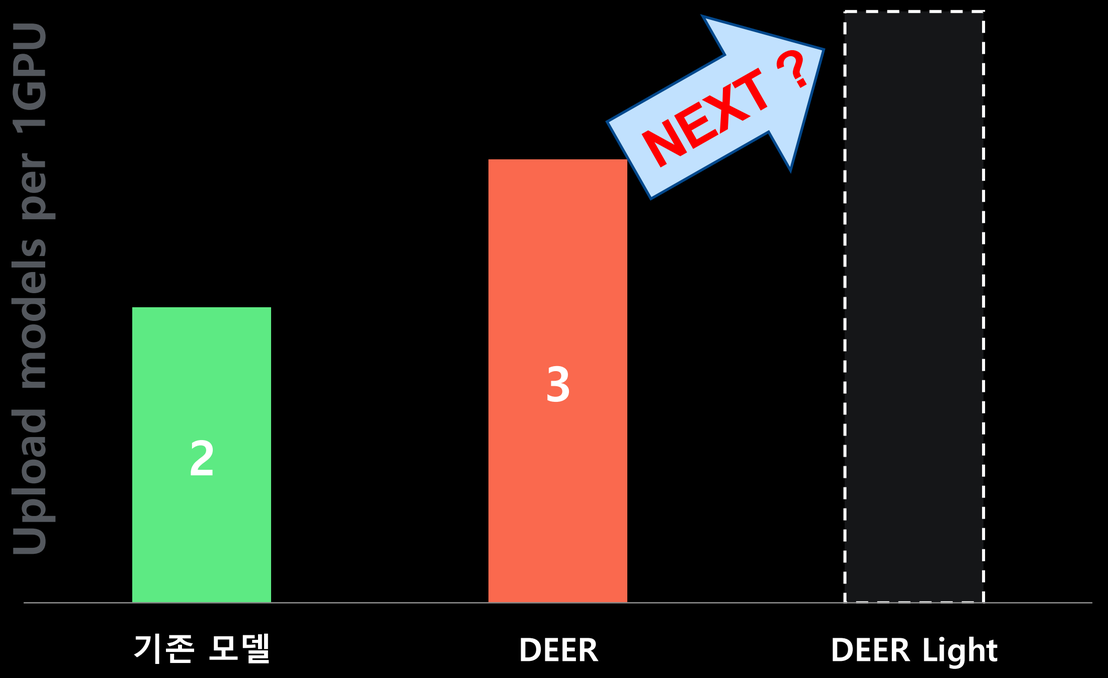
5.1 좀 더 가볍게~

NAVER
DEVIEW
2023

DEER Model 경량화

- 서비스 차원에서 가볍고 빠른 모델에 대한 Needs
- DEER는 기존 service model 대비 빠르고, 가볍지만 추가적으로 더 경량화 시도





5.1 좀 더 가볍게~

NAVER
DEVIEW
2023

Inference Time 분석

- Transformer Encoder에서 56%로 가장 많은 시간을 차지함
- Text Decoder도 22.1%로 두번째로 많은 시간을 차지함

  (단, Decoding Time은 텍스트 개수가 증가하면 같이 증가함)

	Inference time (ms)	비율
Backbone	0.165	16.1%
Prep_inps	0.054	5.3%
encoder	0.581	**56.6%**
decoder	0.227	22.1%
전체	1.027	

* Deer model, p40 gpu, 1 Image, 10 textbox

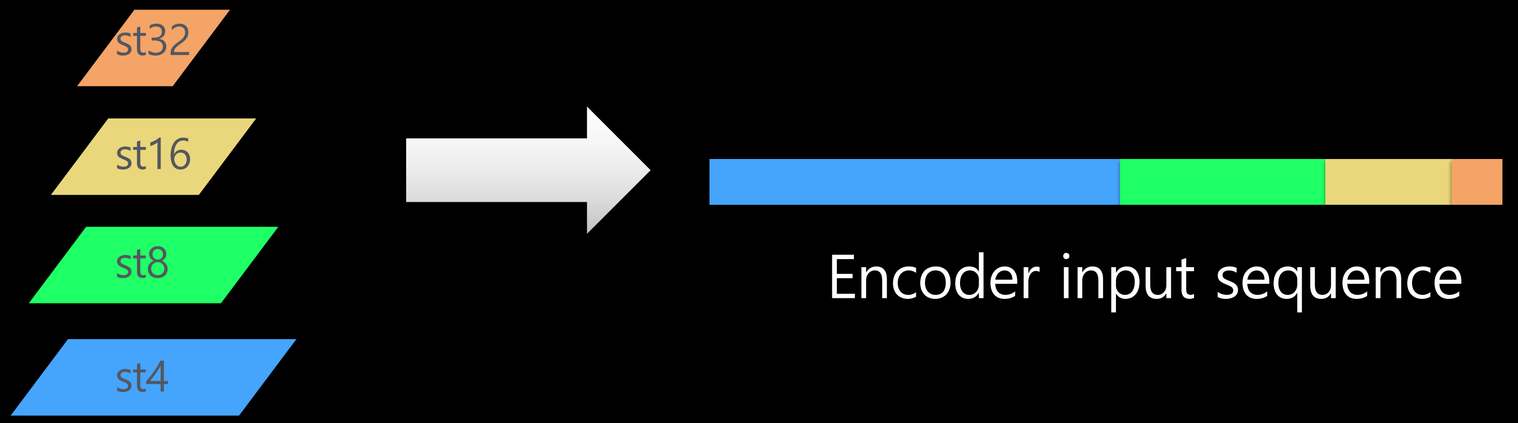
5.1 좀 더 가볍게~ (Encoder)

NAVER
DEVIEW
2023

왜 Encoder에서 가장 많은 시간이 필요할까?

- DEER는 Deformable Attention으로 key/value를 sparse하게 가져가서 시간복잡도 O(S)
- 그러나 Sequence 자체가 길기 때문에 여전히 많은 시간이 소요됨

  → OCR의 경우 작은 텍스트를 읽기 위해선 고화질 영상 입력이 필요

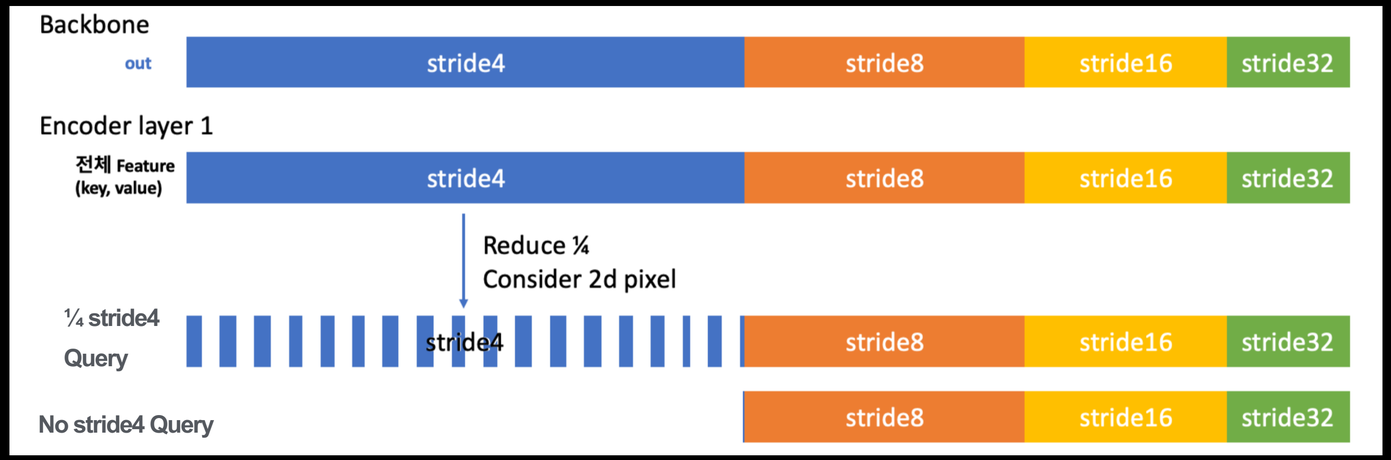


5.1 좀 더 가볍게~ (Encoder)

NAVER
DEVIEW
2023

Encoder에서 Inference Time을 줄일 방안

- Encoder에서 가장 긴 sequence인 stride4 feature를 query에서 줄이기
- 1안: stride4 feature를 query에서 완전 제거
- 2안: stride4 feature를 sparse하게 1/4만 query 제거

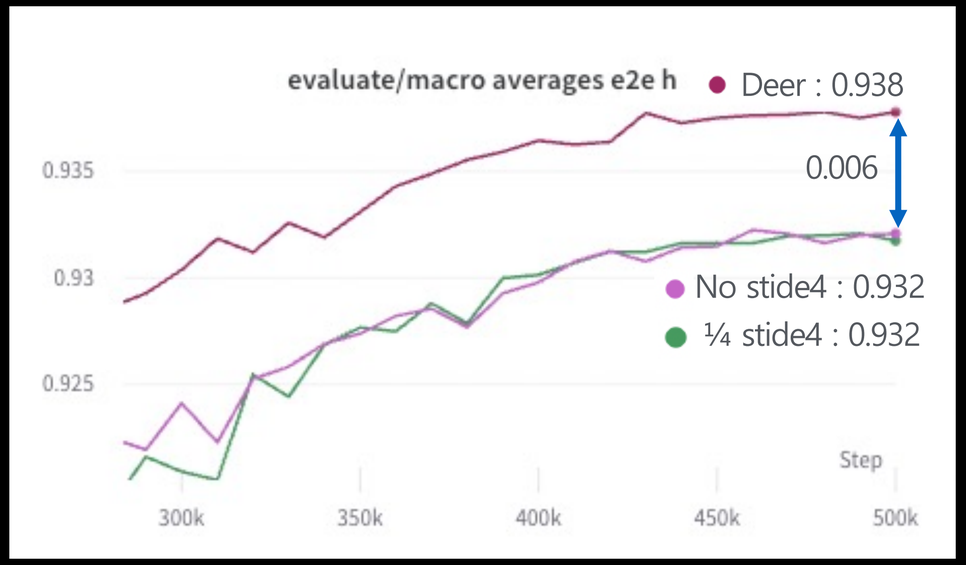


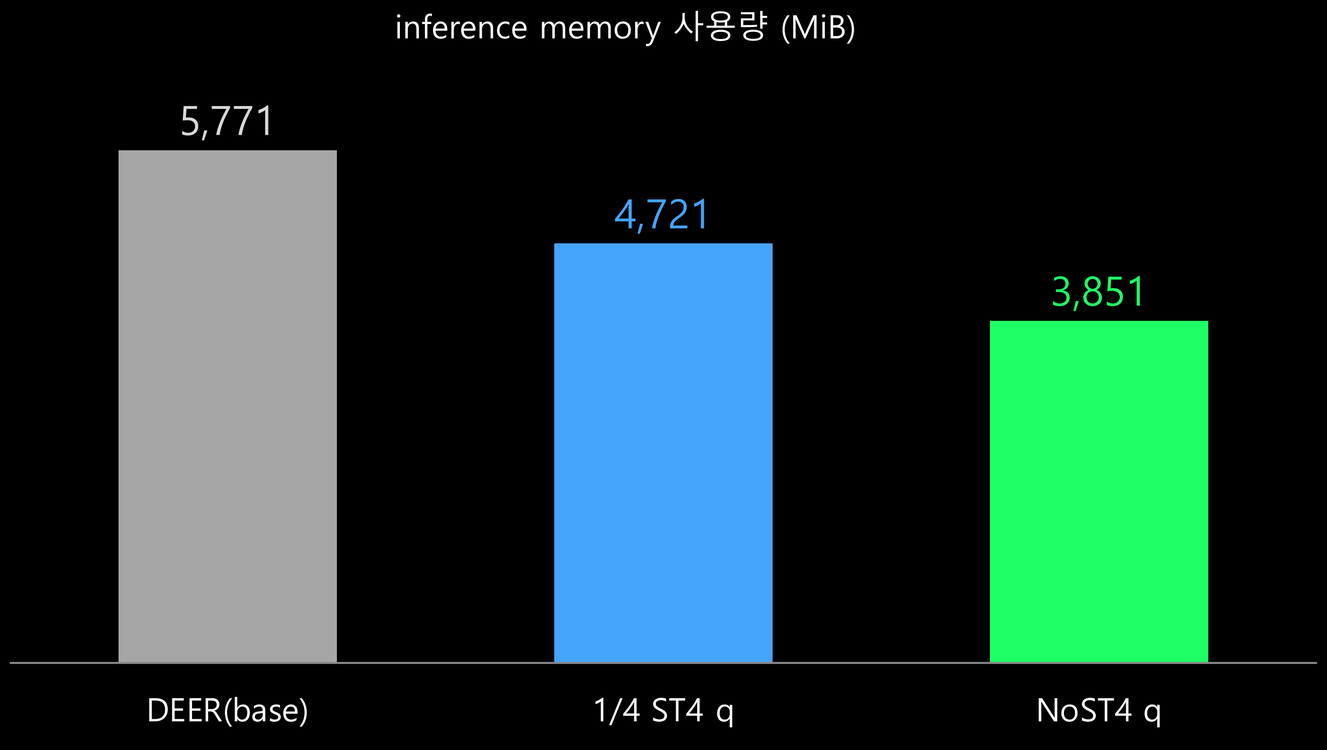
5.1 좀 더 가볍게~ (Encoder)

NAVER DEVIEW 2023

Encoder 경량화 모델과 기존 모델 비교

- 성능하락은 CLEval score 기준 0.006으로 크지 않음
- No Stride4 query 모델은 기존 모델 대비 **encoding time 70%** 감소 (전체 시간 30% 감소)
- **메모리 사용량 -33.3%**



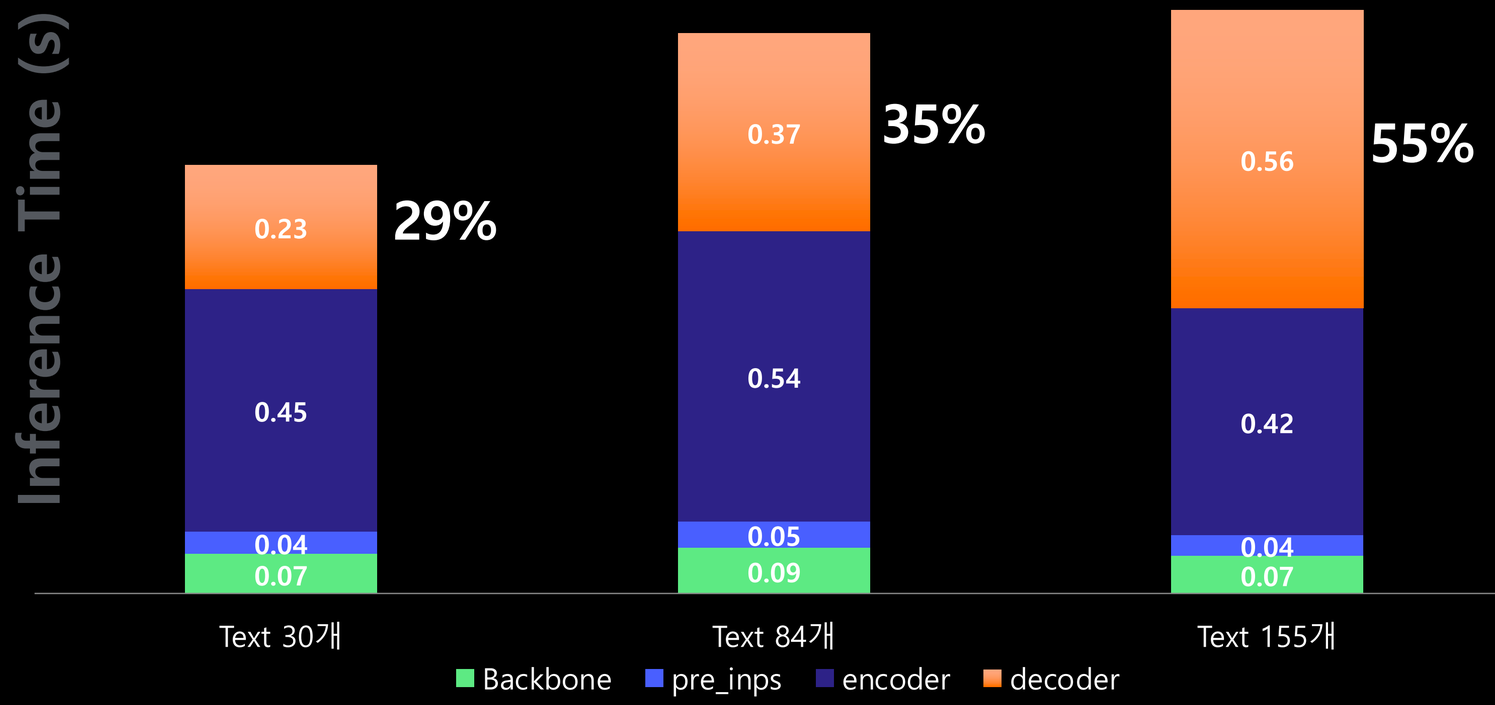


5.1 좀 더 가볍게~ (Decoder)

NAVER DEVIEW 2023

Text 개수별 Decoding 시간 비중은 계속 증가

- Text 155개 기준 decoding 시간은 55%이며 경량화 필요



5.1 좀 더 가볍게~ (Decoder)

NAVER DEVIEW 2023

Decoder 속도 증가 방법

- Decoding 시 배치 크기를 늘려, 한번에 많은 텍스트를 Inference
- 피크 메모리에는 변화가 없음
- 배치 크기를 1024까지 증가시켜 총 디코딩 시간을 73% 감소 시킴

배치 크기	총 시간 (693 박스)	박스 당 시간	속도 향상	피크 메모리	총 시간	속도 향상
64	0.512s	0.00074s	1.00x	4.27G	1.67s	0.0%
128	0.331s	0.00048s	1.55x	4.27G	1.48s	12.8%
256	0.212s	0.00031s	2.42x	4.27G	1.36s	22.8%
512	0.186s	0.00027s	2.75x	4.27G	1.33s	25.6%
1024	0.138s	0.00020s	3.72x	4.27G	1.28s	30.5%

5.2 Out-of-Domain Data 및 성능 검증

NAVER DEVIEW 2023

DEER 모델 OOD data에서 성능 평가

- In domain에서는 기존 모델 대비 좋은 성능이 검증완료
- 다양한 OOD에 대한 평가 필요

OK!

In Domain 데이터

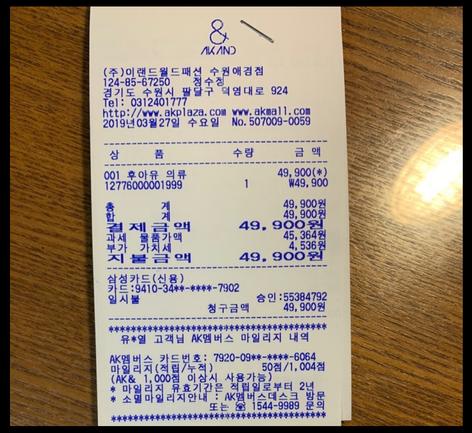
손글씨

의료비 영수증

은행

신용카드

명함



OK???

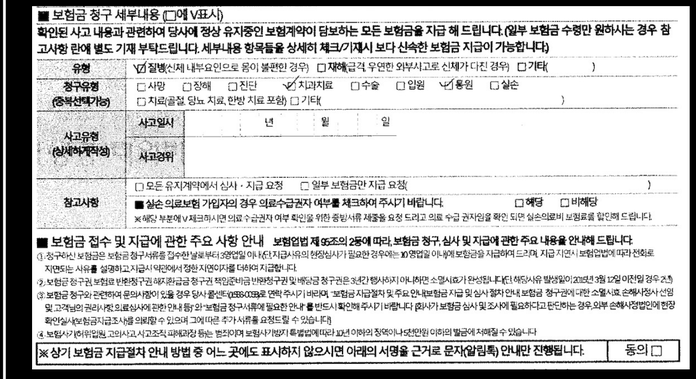
Out Of Domain 데이터

스캔 해서 사진으로 찍으면?

이상한 각도의 문서?

팩스로 받은 문서?

저화질 스캔 된 손글씨?



5.2 Out-of-Domain Data 및 성능 검증

NAVER DEVIEW 2023

OOD data 선정

- In domain data에 없거나 부족한 data List up
- 우선순위 지정
- Annotation Guide 수립

우선도	Aa 이름	언어	이미지 수집 특성	리소스
최우선	손글씨가 기입된 서식 문서	일본어	카메라	ja_document-handwritten_event
최우선	손글씨가 기입된 서식 문서	일본어	스캔	
최우선	손글씨가 기입된 서식 문서	일본어	팩스/저화질 스캔	ja_document-handwritten_event
최우선	손글씨가 기입된 서식 문서	한국어	카메라	팀 손글씨 수집
최우선	손글씨가 기입된 서식 문서	한국어	스캔	보험금 청구서
최우선	손글씨가 기입된 서식 문서	한국어	팩스/저화질 스캔	
필수	인보이스	일본어	카메라	ja_invoice_internal_prv
필수	인보이스	일본어	팩스/저화질 스캔	ja_invoice_internal_prv
필수	인보이스	영어	카메라	ja_invoice_internal_prv
필수	인보이스	영어	팩스/저화질 스캔	ja_invoice_internal_prv
필수	인보이스 외 테이블이 포함된 문서	일본어	카메라	
필수	인보이스 외 테이블이 포함된 문서	일본어	스캔	
필수	인보이스 외 테이블이 포함된 문서	일본어	팩스/저화질 스캔	
필수	인보이스 외 테이블이 포함된 문서	한국어	카메라	
필수	인보이스 외 테이블이 포함된 문서	한국어	스캔	재무팀 샘플
필수	인보이스 외 테이블이 포함된 문서	한국어	팩스/저화질 스캔	공공행정문서

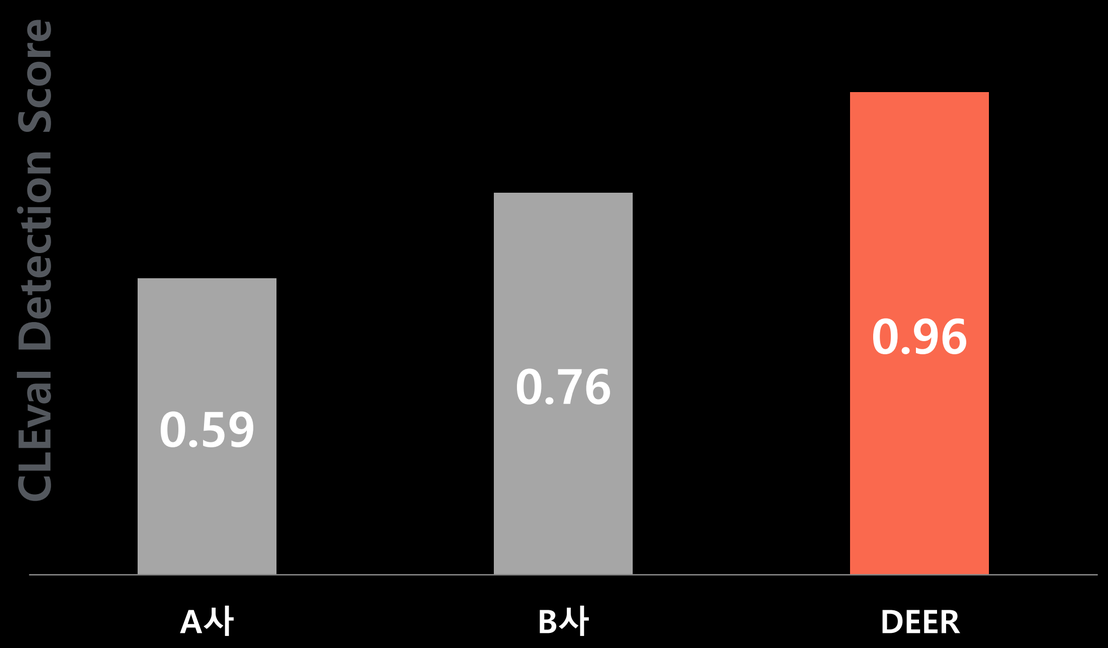
...

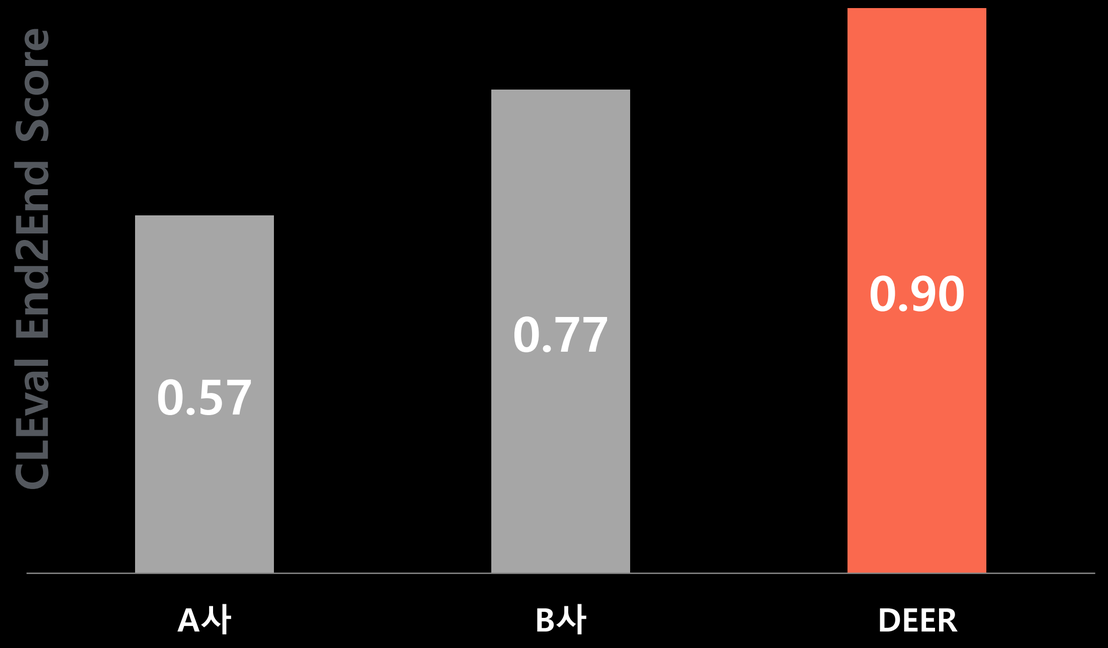
5.2 Out-of-Domain Data 및 성능 검증

NAVER
DEVIEW
2023

OOD data 성능비교 결과

- 학습 때 보지 못한 16개 도메인에서 각 100장의 이미지로 평가
- 글로벌 타 경쟁사 대비 높은 성능





Wrap up

NAVER
DEVIEW
2023

기존 2Stage service 모델의 문제점을 개선한 DEER 모델 개발

- 기존 대비 메모리도 적게 사용, 빠르고, 성능 강인한 모델을 개발
- 신규 기능 추가(Character detection, 문서 구조 추출)
- ECCV 2022 OOV Challenge에서 DEER model 1등 달성으로 성능 확인
- 서비스 배포를 위해 OOD dataset으로 타사 대비 성능 검증

Q & A

Thank You