

OCR Project

# TrOCR 모델과 CRAFT 모델을 활용한 식품 영양성분표 텍스트 추출 시스템

문상흠 박창현 이동준  
위서현 조유경 한동우

# Contents 목차

## 01 주제 선정

가공식품의 식품 표시 인식 조사 결과  
주제 선정 이유

## 02 데이터 분석 방법 및 결과

분석방법 – NMF 모델링  
분석결과 – WordCloud

## 03 모델 선정

OCR소개  
Basic OCR  
TROCR

## 04 데이터 선정

AI허브 한국어 글자체 이미지

## 05 모델 평가 및 비교

Accuracy, WER, CER, Jamo

## 06 평가 결과

IoU 기반 Average CER  
PopEval Precision, Recall, F1-score  
워크플로우

# 01 주제 선정

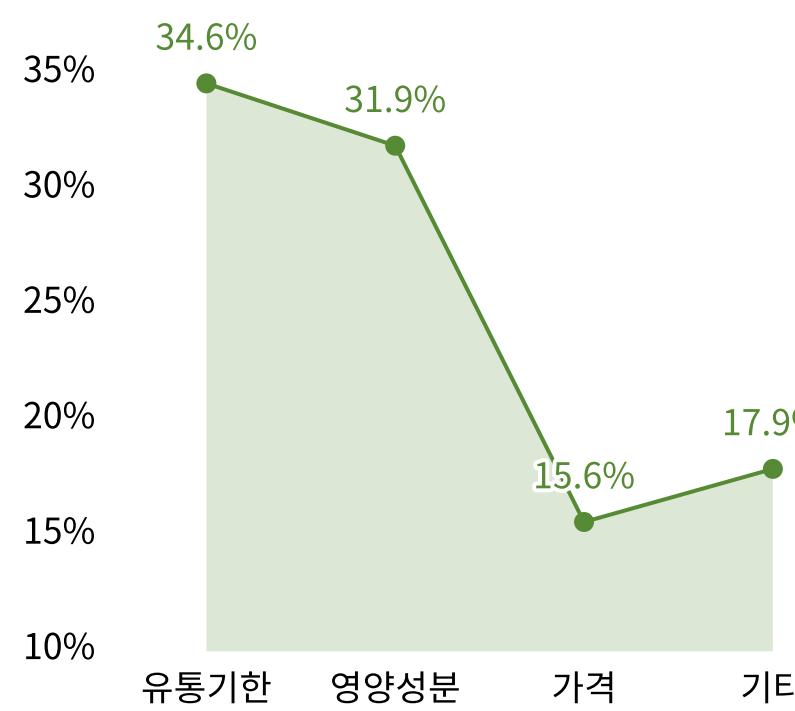
## 가공식품의 식품 표시 인식 조사 결과

- ◎ 소비자들이 식품 구매 시 중요하게 생각하는 요소들
- ◎ 1위 유통기한(34.6%), 2위 "영양성분(31.9%)", 3위 가격(15.6%)
- ◎ 식품 표시를 확인하는 이유
- ◎ 1위 건강과 안전을 위해(47.5%), 2위 "영양정보 확인을 위해(44.7%)"

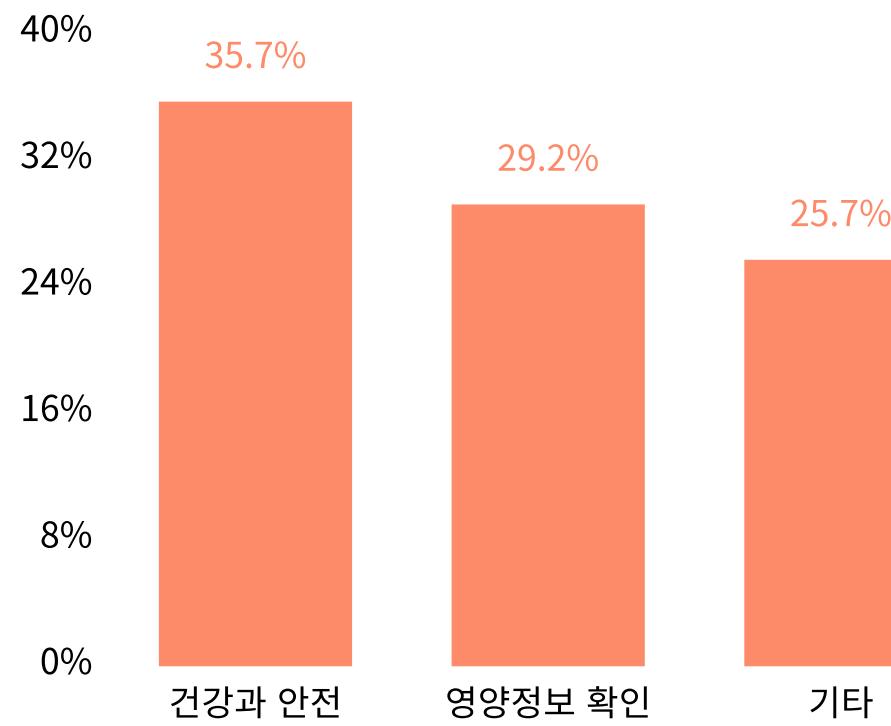
## 주제 선정 이유

- ◎ 문제 인식 : 식품영양성분표의 "**글씨가 작아 보기 어렵고(35.7%)**", 이해하기 어렵다는 이유(29.2%)로 확인이 어려운 현실
- ◎ 해결 방안 : OCR 기술을 통해 영양성분표를 디지털화하여 접근성을 높이고, 소비자가 식품 정보를 쉽게 이해할 수 있도록 돕기 위함

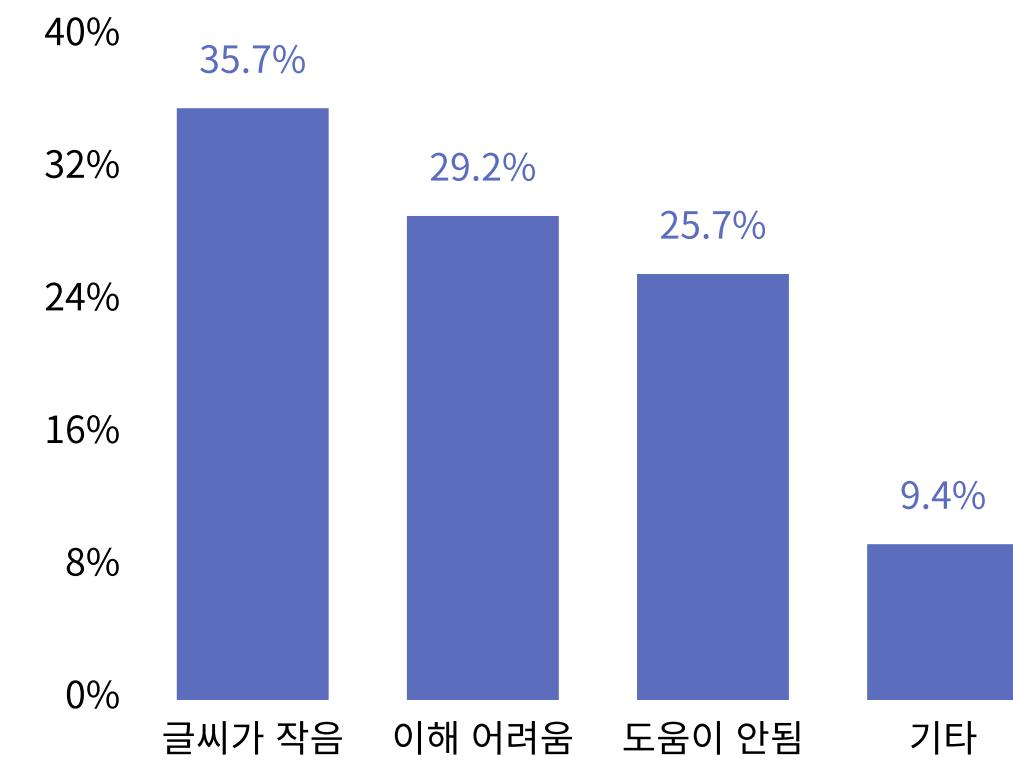
식품 구매 시 중요하게 생각하는 요소



식품 표시를 확인하는 이유



식품 표시를 확인하지 않는 이유



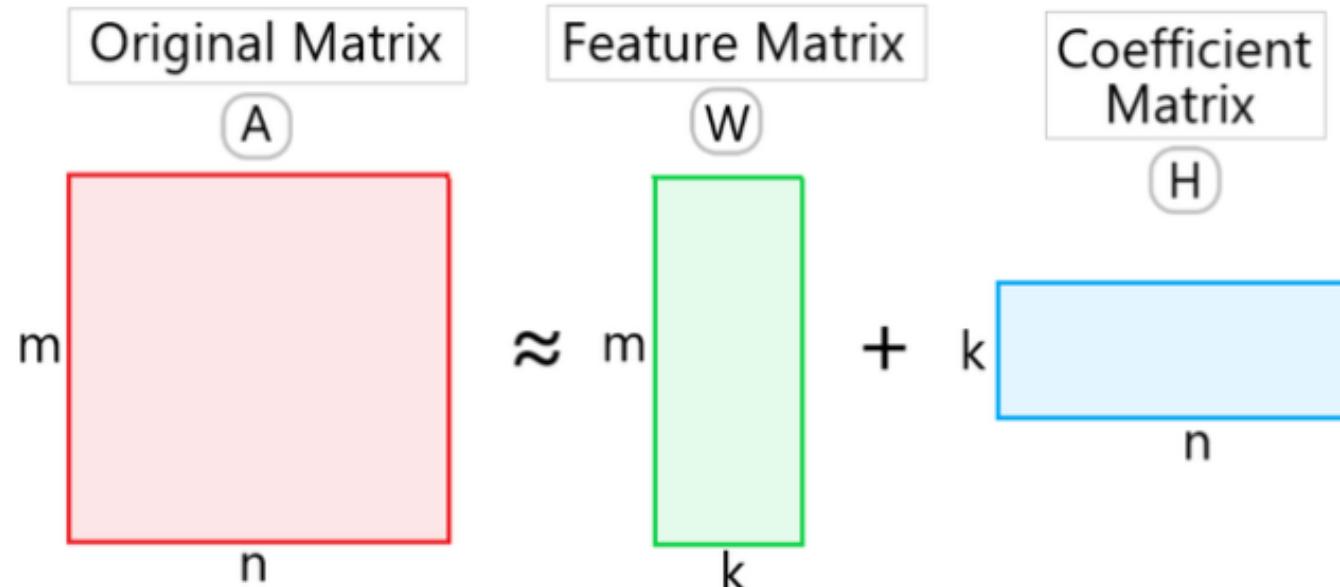
# 02 데이터 분석 방법 및 결과

## 분석 방법

- ◎ 데이터의 비음수 특성을 고려하여 NMF 모델링 적용
  - ◎ 자연스러운 주제 및 키워드 도출에 최적화된 방식 선택
  - ◎ 유저ID / 리뷰 내용 / 점수 / 좋아요 / 답변 내용 / 작성 시간 등 필요한 내용 수집

# 비음수 행렬분해

## NMF(Non-negative Matrix Factorization)



분석 결과

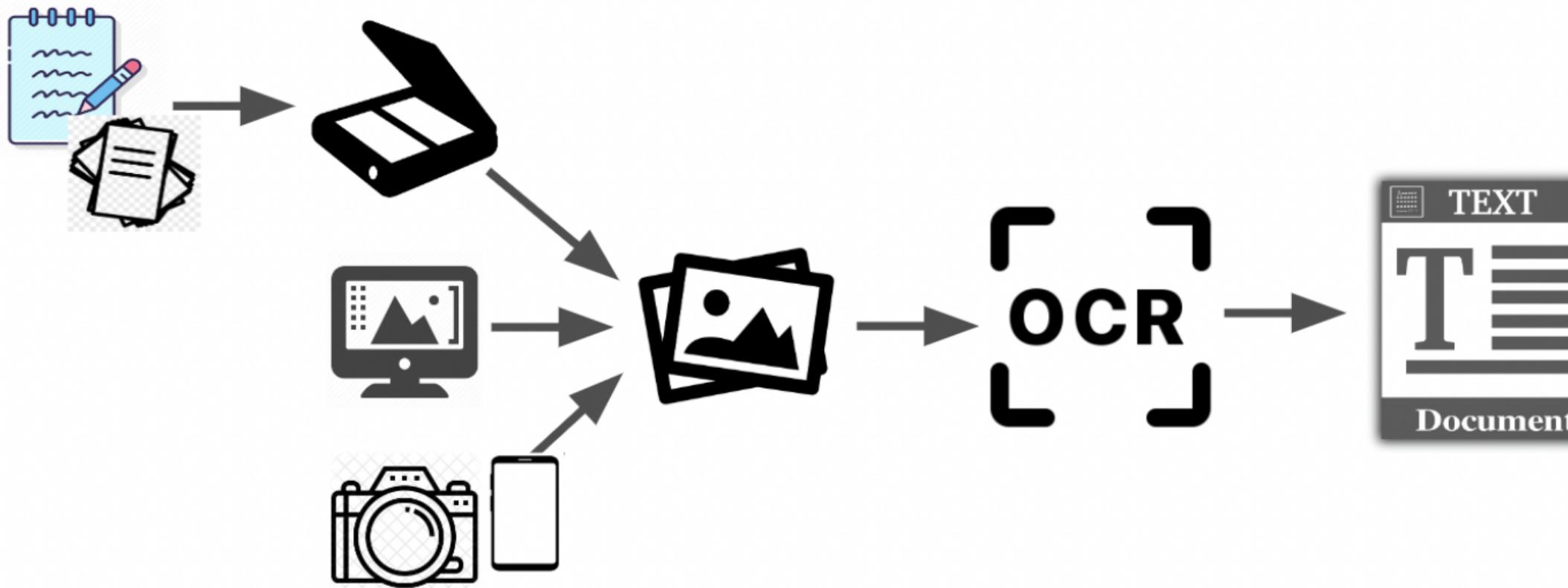
- ◎ 핵심 키워드 : **계산, 추가, 검색, 입력, 수정, 조절**
  - ◎ 사용자들의 주요 관심사 : **다이어트 목적, 전반적인 건강 개선**

FatSecret 어플 WordCloud



# 03 모델 선정

## OCR – 광학 문자 인식



- ◎ 이미지에 포함된 문자나 단어를 인식하고 이를 컴퓨터가 읽을 수 있는 디지털 텍스트로 변환하는 기술
- ◎ 주로 스캔된 문서나 사진에서 글자를 추출하여 편집 가능한 형태로 바꾸는 데 사용

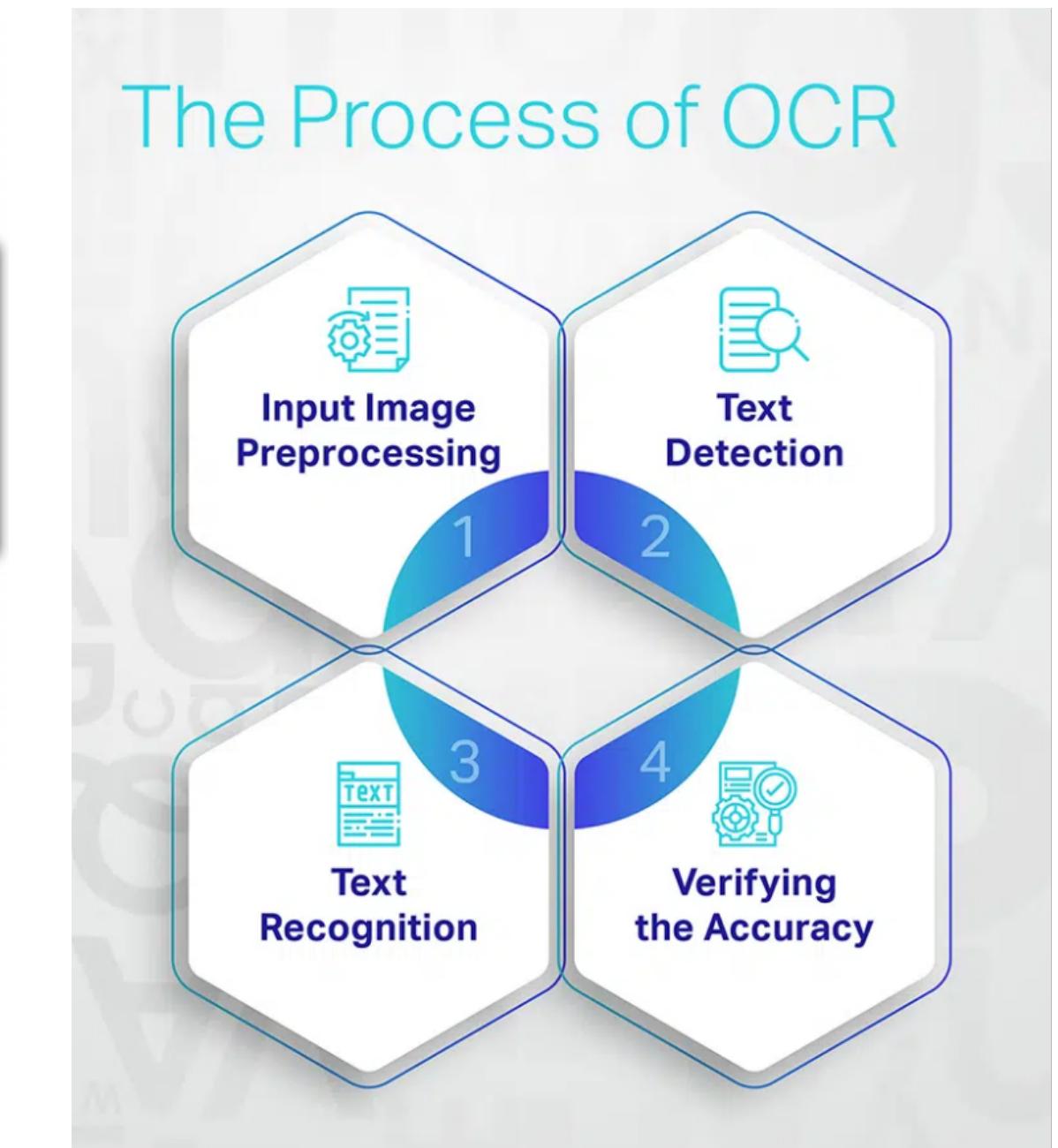
**1. Image Preprocessing** : 이미지의 노이즈 제거, 대비 조정, 이진화를 통해 이미지 품질 향상

**2. Text Detection** : 텍스트가 위치한 영역 식별, 바운딩 박스나 다른 형식으로 표시하여

후속 단계 처리를 도움

**3. Text Recognition** : 문자 패턴을 학습한 모델을 통해 문자를 정확하게 인식, 디지털 데이터로 변환

**4. Verifying the Accuracy** : 인식된 텍스트의 정확도 검증. 오인식된 문자나 문장에 맞지 않는 단어를 교정하고, 최종 텍스트를 보완하여 정확한 디지털 텍스트가 되도록 함



# 03 모델 선정

## <Basic OCR Model Architecture>

- ◎ 기존 OCR 모델은 Detection 단계에서 텍스트 영역을 식별하고, Recognition 단계에서 문자를 인식하여 디지털 텍스트로 변화하는 일련의 로직을 포함
- ◎ 전처리 과정을 통해 이미지의 노이즈를 제거하고 텍스트의 윤곽을 파악하여 인식률을 높임
- ◎ 적응형 임계값 처리, 문자 윤곽 추출, 단어 리스트 생성 등을 통해 텍스트의 연속성을 유지하며 인식
- ◎ 별도의 이미지 전처리 단계가 필수적으로 요구됨

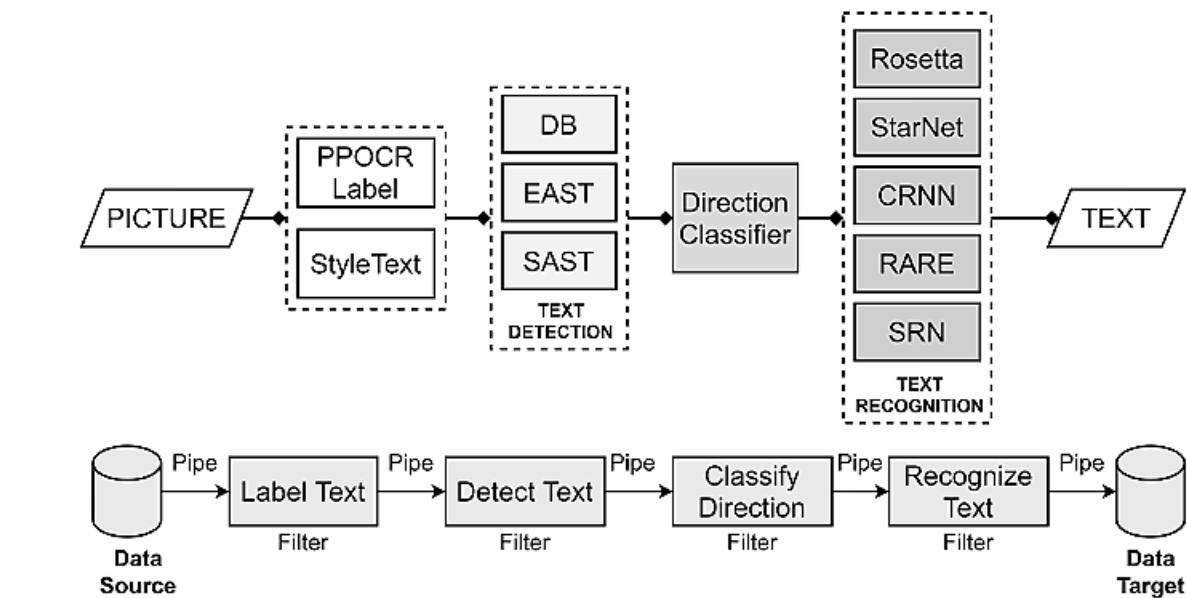
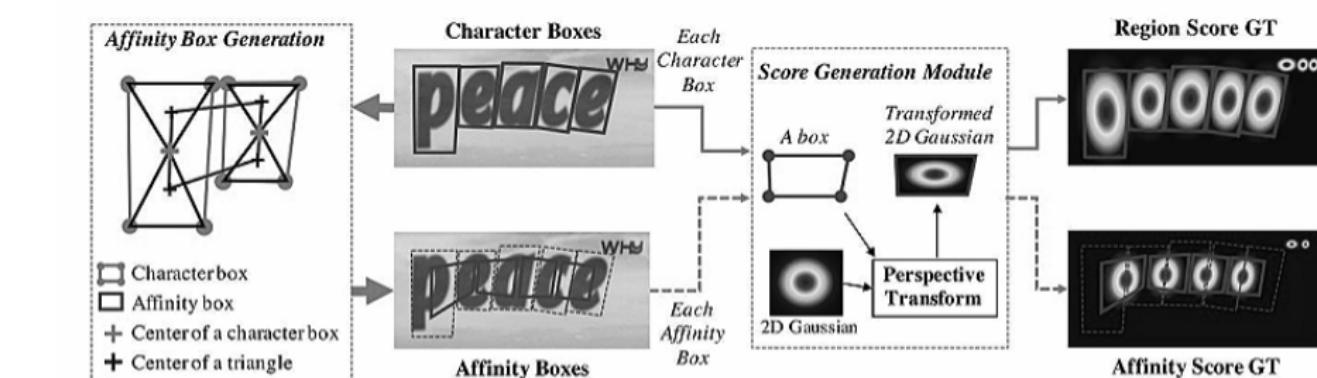
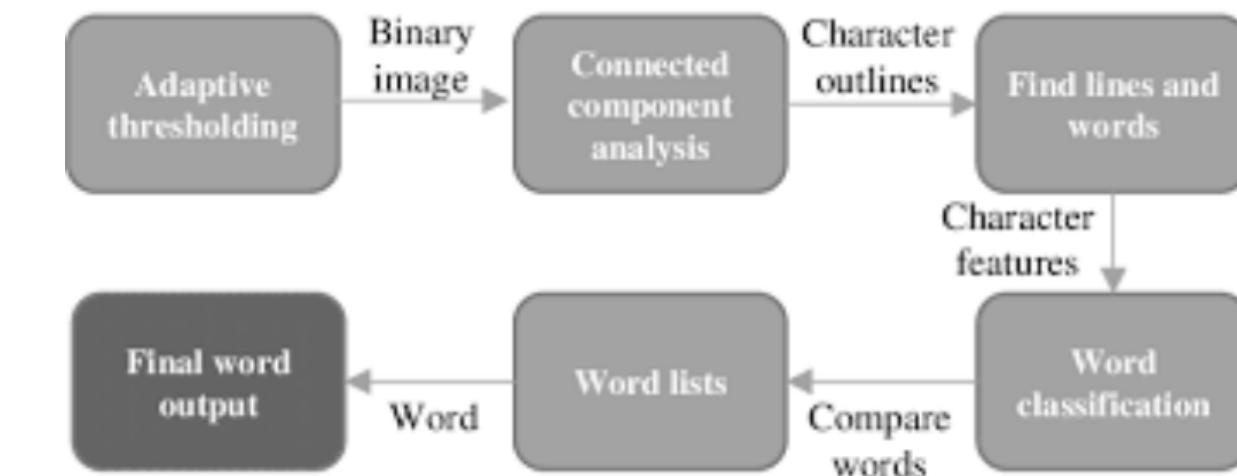
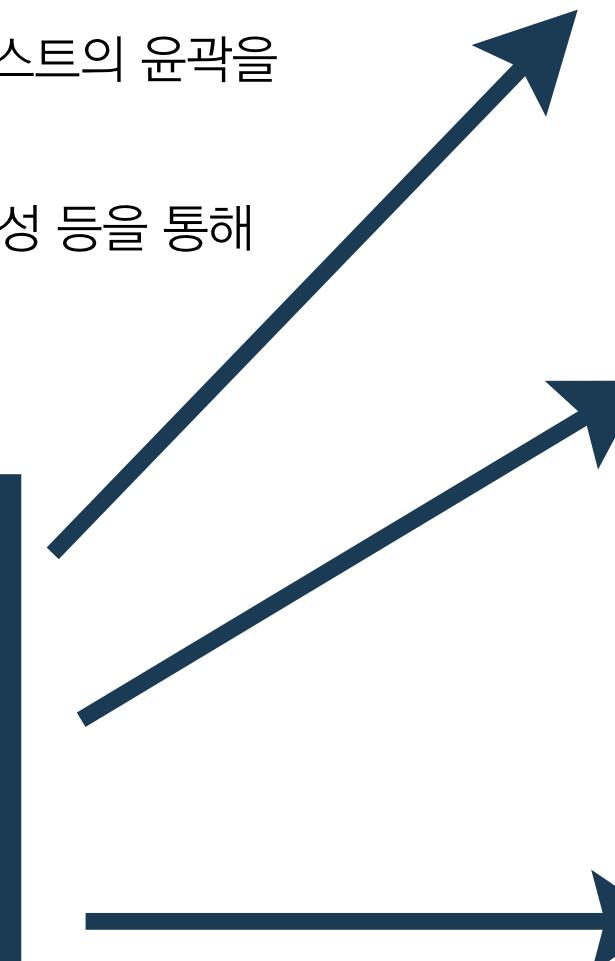
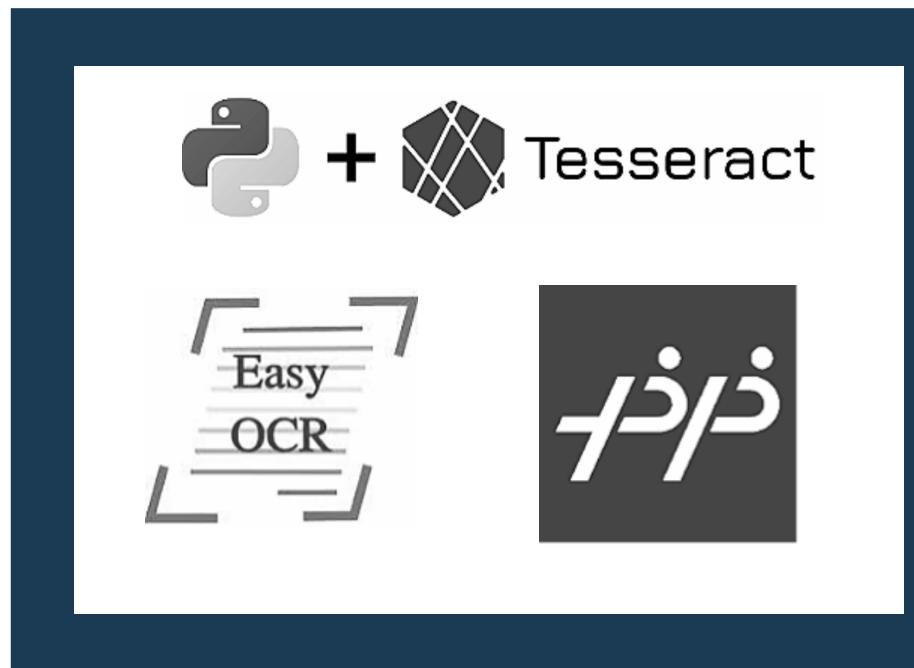


Figure: PaddleOCR Working Procedure

# 03 모델 선정

## 기존 OCR 모델의 한계

### 1. 복잡한 레이아웃에 대한 취약성

- 주로 CNN-RNN 구조를 사용하여 순차적으로 텍스트를 인식하므로, 복잡한 문서 레이아웃이나 다양한 텍스트 배치에 적응하기 어려움

### 2. 전처리에 의존하는 성능

- 이미지의 노이즈 제거, 대비 조정 등의 전처리 과정에 크게 의존하여, 전처리 없이 인식률이 크게 떨어짐

### 3. 다양한 글꼴 및 언어 지원의 한계

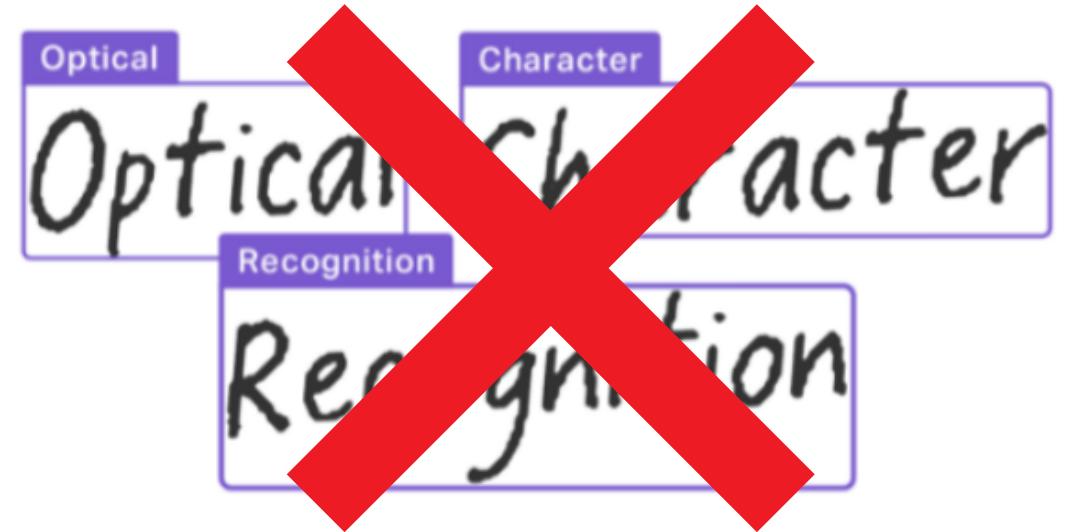
- 특정 글꼴과 언어에 최적화되어 있어 다국어 및 다양한 글꼴 인식에 한계가 있음

### 4. 문맥 이해의 부족으로 인한 인식 오류

- 개별 문자나 단어 단위로 인식하여 문장의 전체 문맥을 고려하기 어려워 문맥 오류 발생 가능

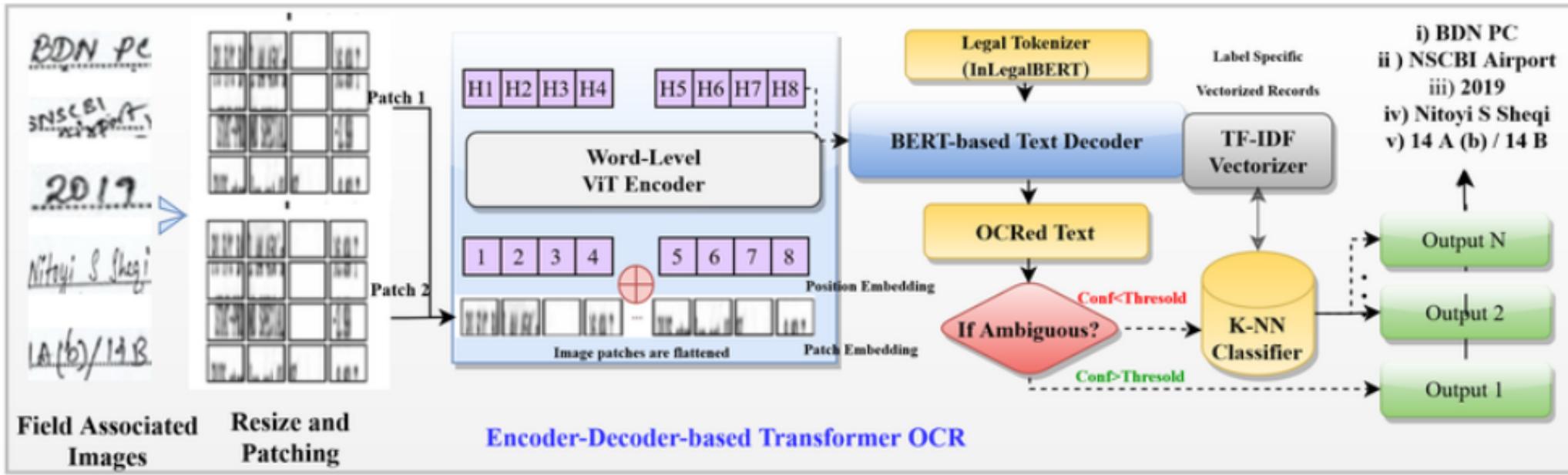
### 5. 순차 처리로 인한 속도 제한

- 기존의 RNN 기반 OCR은 순차적인 처리로 인해 인식 속도가 느린 경우가 많음



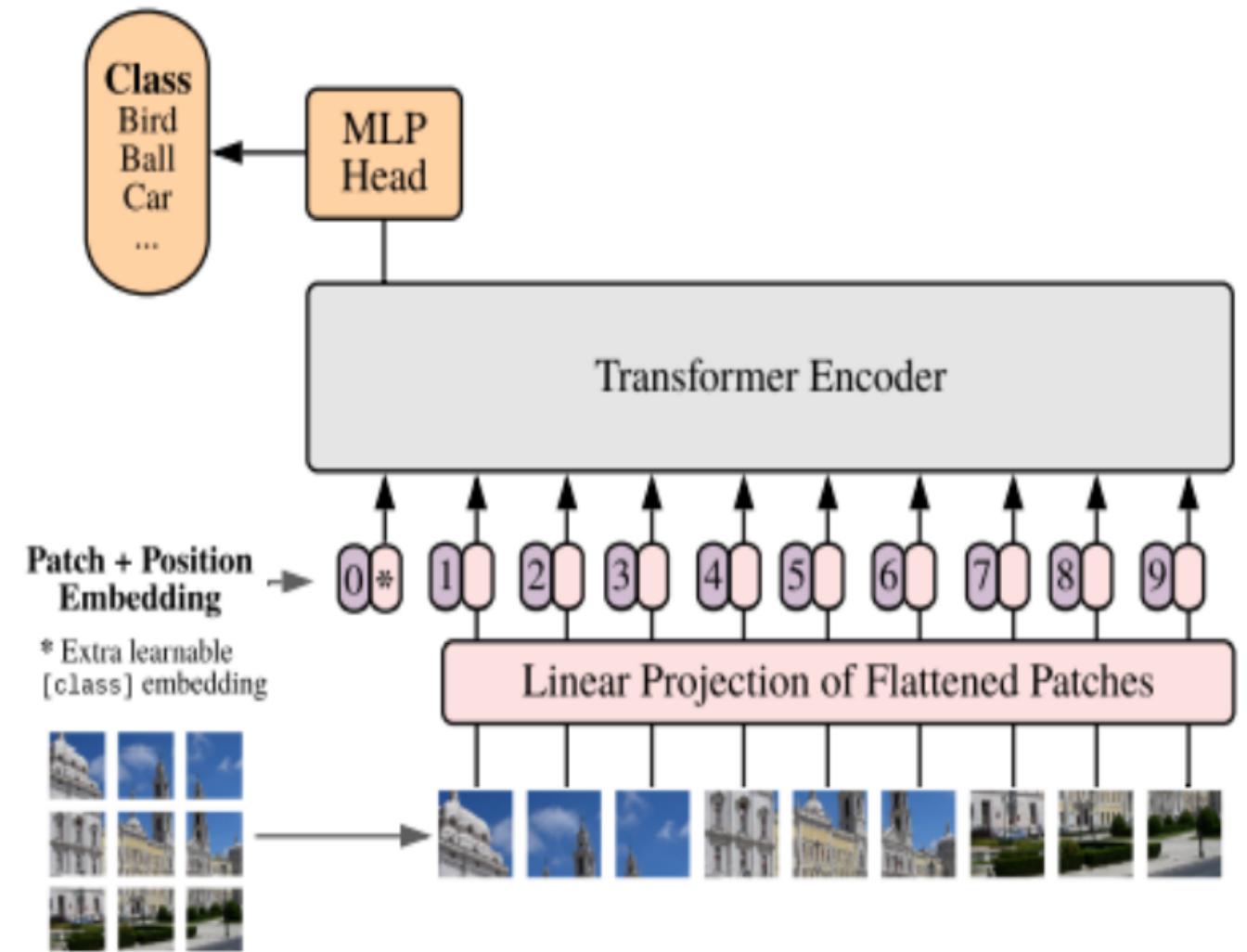
# 03 모델 선정

## TROCR(Transformer-based Optical Character Recognition)



- ◎ 간단한 텍스트 이미지 데이터에 대한 end-to-end 모델  
 ※ end-to-end model : 기존의 OCR 모델들이 수행하던 단계들을 수행하지 않고 처음부터 끝까지 하나의 모델이 전부 처리하는 방식의 모델
- ◎ CNN 기반의 로직을 통해 이미지에 대한 feature 학습
- ◎ 별도의 전처리가 필요하지 않고 원하는 이미지를 입력으로 넣어주면 text 추출하여 출력
- ◎ 기존의 OCR 모델들이 갖고 있던 detection 단계 수행하지 않음
- ◎ VIT(Vision Transformer)는 이미지를 분할해 패치 단위로 트랜스포머에 입력함으로써, 이미지 내 세부적인 텍스트 특징을 효과적으로 추출
- ◎ OCR에서 VIT는 이미지 분할 후 텍스트 영역을 인식하거나 추출하는 단계에서 높은 성능을 발휘해, 특히 복잡한 배경의 텍스트 인식에 효과적

## VIT(Vision Transformer)



# 03 모델 선정

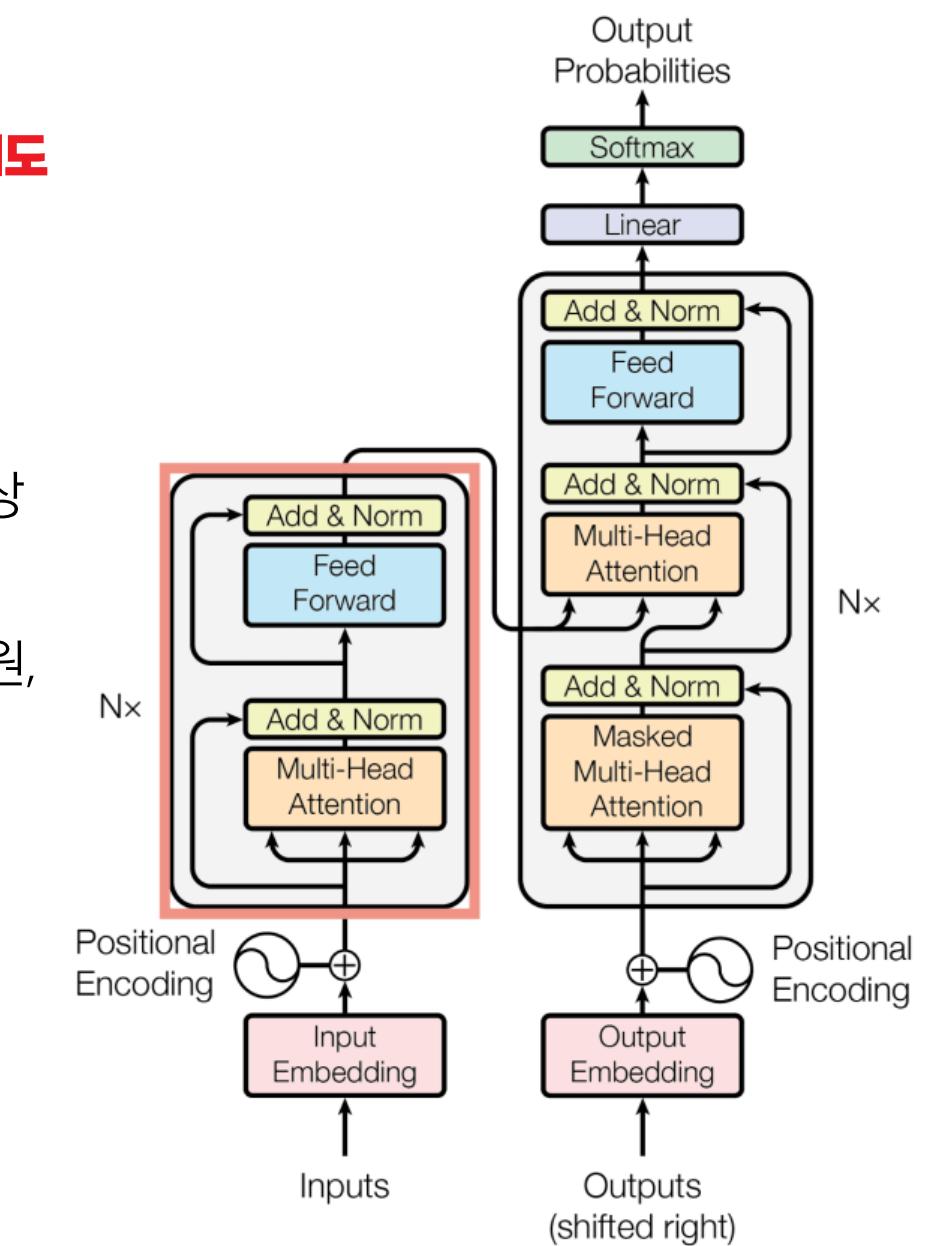
## 기존 OCR 모델의 한계

1. 복잡한 레이아웃에 대한 취약성
  - 주로 CNN-RNN 구조를 사용하여 순차적으로 텍스트를 인식하므로, 복잡한 문서 레이아웃이나 다양한 텍스트 배치에 적응하기 어려움
2. 전처리에 의존하는 성능
  - 이미지의 노이즈 제거, 대비 조정 등의 전처리 과정에 크게 의존하여, 전처리 없이 인식률이 크게 떨어짐
3. 다양한 글꼴 및 언어 지원의 한계
  - 특정 글꼴과 언어에 최적화되어 있어 다국어 및 다양한 글꼴 인식에 한계가 있음
4. 문맥 이해의 부족으로 인한 인식 오류
  - 개별 문자나 단어 단위로 인식하여 문장의 전체 문맥을 고려하기 어려워 문맥 오류 발생 가능
5. 순차 처리로 인한 속도 제한
  - 기존의 RNN 기반 OCR은 순차적인 처리로 인해 인식 속도가 느린 경우가 많음



## TROCR의 장점

1. 트랜스포머 아키텍처를 사용해 **병렬 처리가 가능**하며 이미지 내의 모든 텍스트 영역을 전역적인 문맥에서 이해하여 **복잡한 레이아웃에서도 높은 인식 정확도**를 제공
2. 트랜스포머 기반의 자체적인 문맥 학습으로 **이미지 전처리 없이도** 안정적인 인식 성능을 발휘, 전처리 단계를 줄여 데이터 처리 속도 향상
3. 사전 학습된 트랜스포머 모델을 활용하여 **다국어와 다양한 글꼴**을 지원, 여러 언어와 스타일에 적응할 수 있는 유연성을 제공
4. 트랜스포머의 **Self-Attention**을 통해 텍스트의 전체 문맥을 학습, 문장의 의미를 유지하며 높은 인식 정확도를 실현
5. 트랜스포머의 병렬 처리 구조를 활용해 속도와 효율성을 개선, **대량의 문제에서도 빠른 인식** 가능



# 04 데이터 선정

## AI Hub 한국어 글자체 이미지

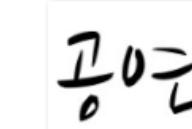
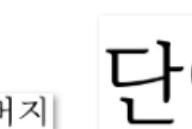
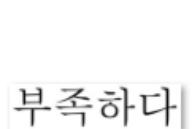
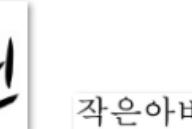
- ◎ Dataset 특징 : 복잡한 Text Image, '한국어' Text
  - ※ '한국어'의 경우 언어의 문법적, 어휘적 특성으로 인해 NLP에서도 어려운 분류에 속함
- ◎ 모델을 평가하여 선정 시 '한국어' 출력 정확도가 중요한 과제
- ◎ 기본 성능을 알아보기 위해 전처리가 필요하지 않은 심플한 Dataset 사용



## <Labeling Architecture>



## <Dataset Sample>

비행	대상자	배추김치	팔월	차라리	취직
02234400.png	02234402.png	02234404.png	02234406.png	02234408.png	02234410.png
적응하다	못되다	여기저기	계시다	기쁘다	신라
02234424.png	02234426.png	02234428.png	02234430.png	02234432.png	02234434.png
					
작은아버지	단어	부족하다	답	테니스	
02234448.png	02234450.png	02234452.png	02234454.png	02234456.png	02234458.png
					
친척	지리산	부서	순서	장미	긍정적
02234472.png	02234474.png	02234476.png	02234478.png	02234480.png	02234482.png

# 05 모델 평가 및 비교 - 평가지표



**Accuracy**

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$



**WER**  
(Word Error Rate)

$$\text{WER} = \frac{\text{S} + \text{D} + \text{I}}{\text{N}}$$

S : 대체된 단어 수  
D : 삭제된 단어 수  
I : 삽입된 단어 수  
N : 참조 텍스트의 총 단어 수



**CER**  
(Character Error Rate)

$$\text{CER} = \frac{\text{S} + \text{D} + \text{I}}{\text{N}}$$

S : 대체된 문자 수  
D : 삭제된 문자 수  
I : 삽입된 문자 수  
N : 참조 텍스트의 총 문자 수



**Jamo**  
(Python Library)

홍길동

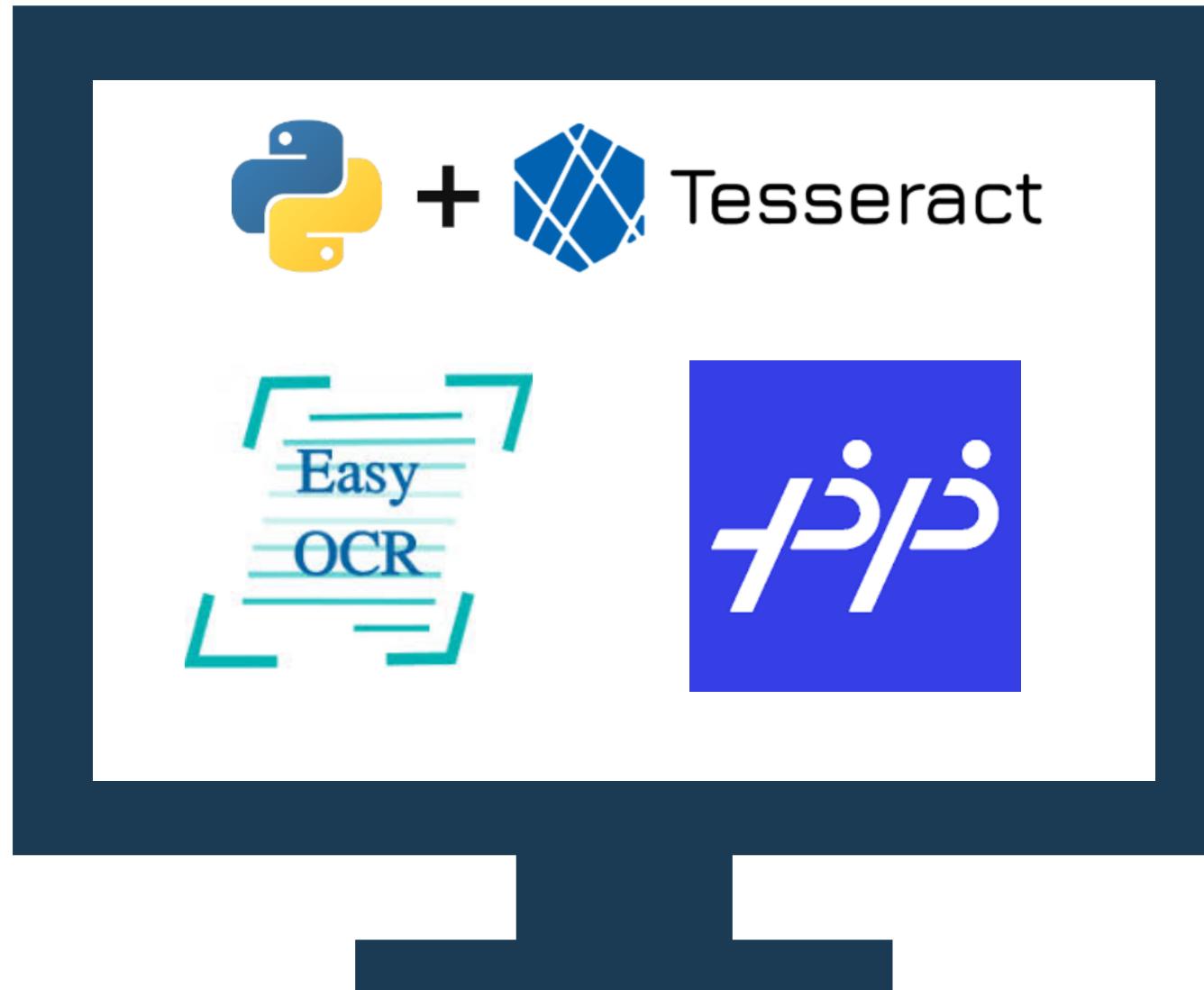
ㅎ,ㅏ,ㅗ,  
ㄱ,ㅣ,ㅓ,  
ㄷ,ㅏ,ㅗ

한글 자음, 모음 분리해주는  
Python Library Jamo

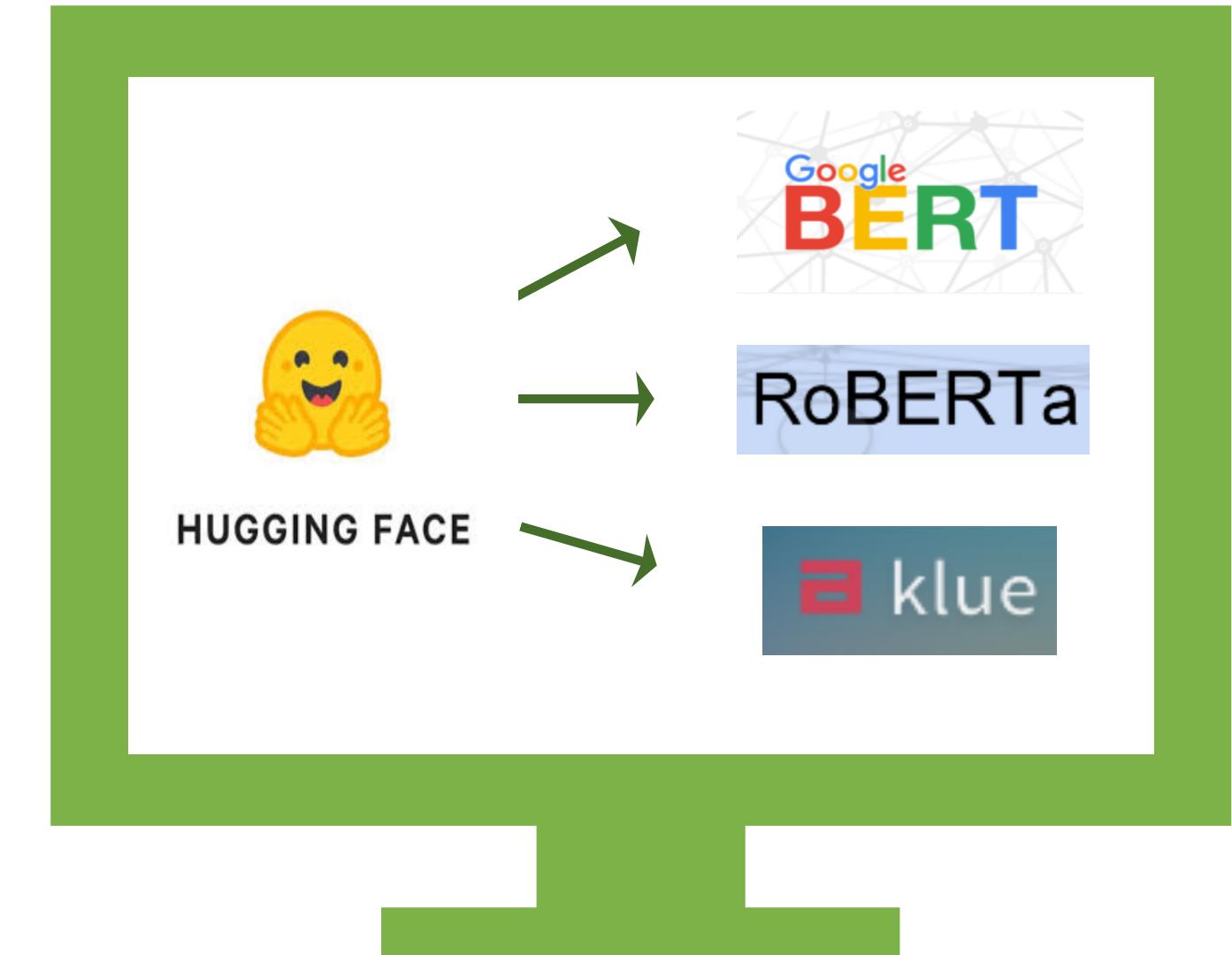
## 05 모델 평가 및 비교

# 모델 별 평가지표 적용

기존의 3개의 OCR모델  
(Pytesseract, EasyOCR, PaddleOCR)

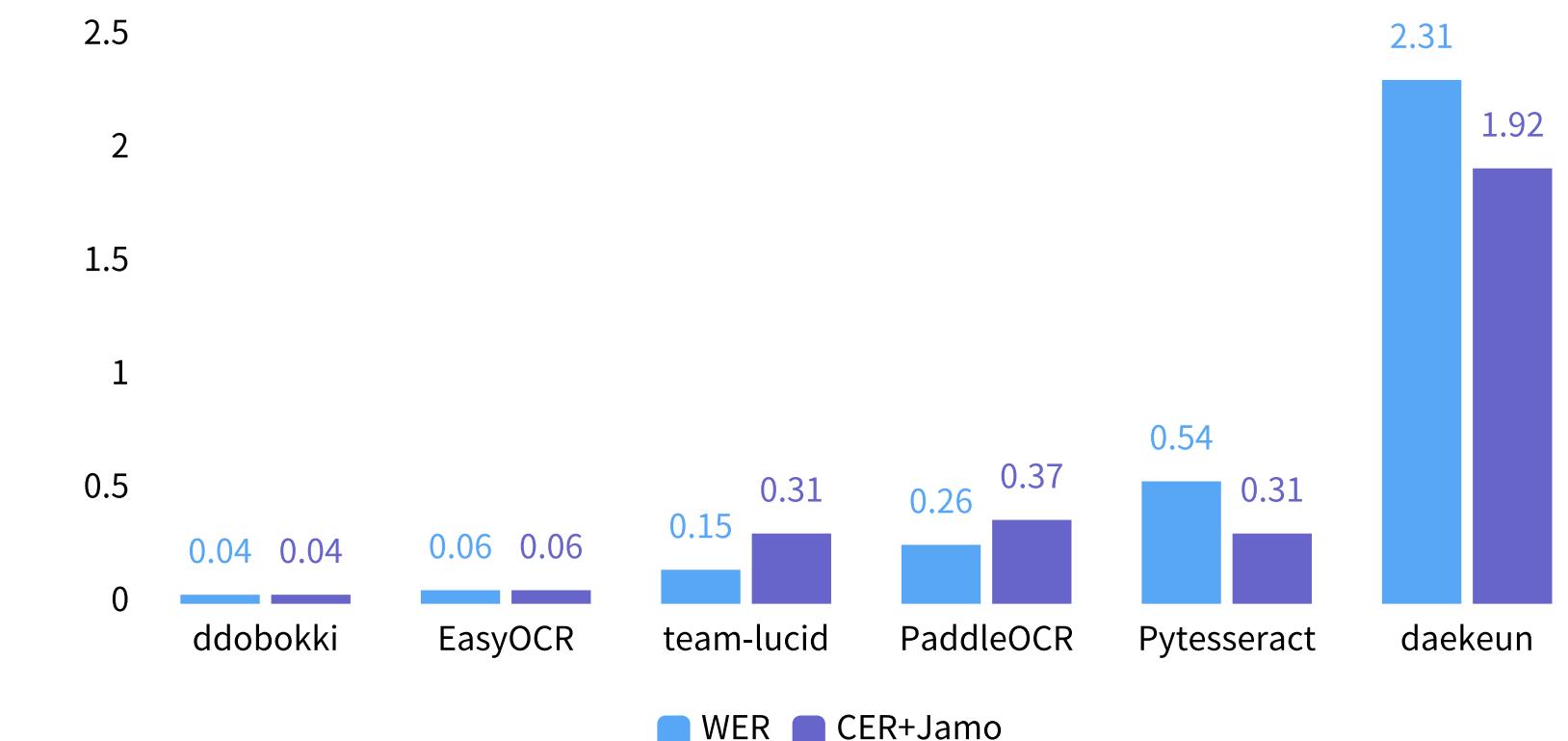
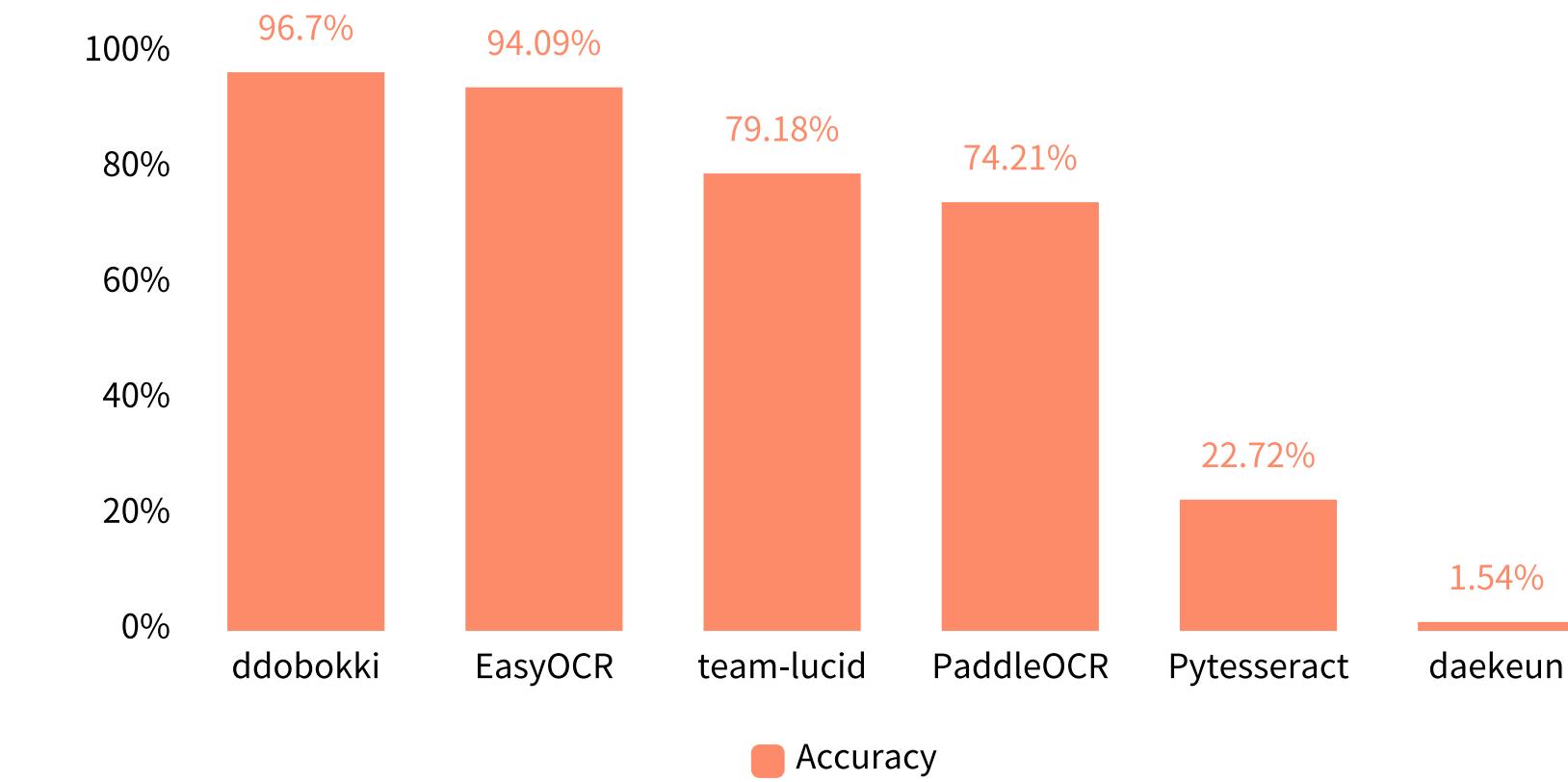


한국어 출력을 제공하는 3개의 TrOCR모델  
(ddobokii, team-lucid, daekeun)



# 05 모델 평가 및 비교(Results)

	Accuracy	WER	CER+Jamo
Pytesseract	22.72%	0.5385	0.3071
EasyOCR	94.09%	0.0576	0.0638
PaddleOCR	74.21%	0.2565	0.3742
ddobokki	<u>96.70%</u>	<u>0.0351</u>	<u>0.0444</u>
team-lucid	79.18%	0.1538	0.3149
daekeun	1.54%	2.3106	1.9283

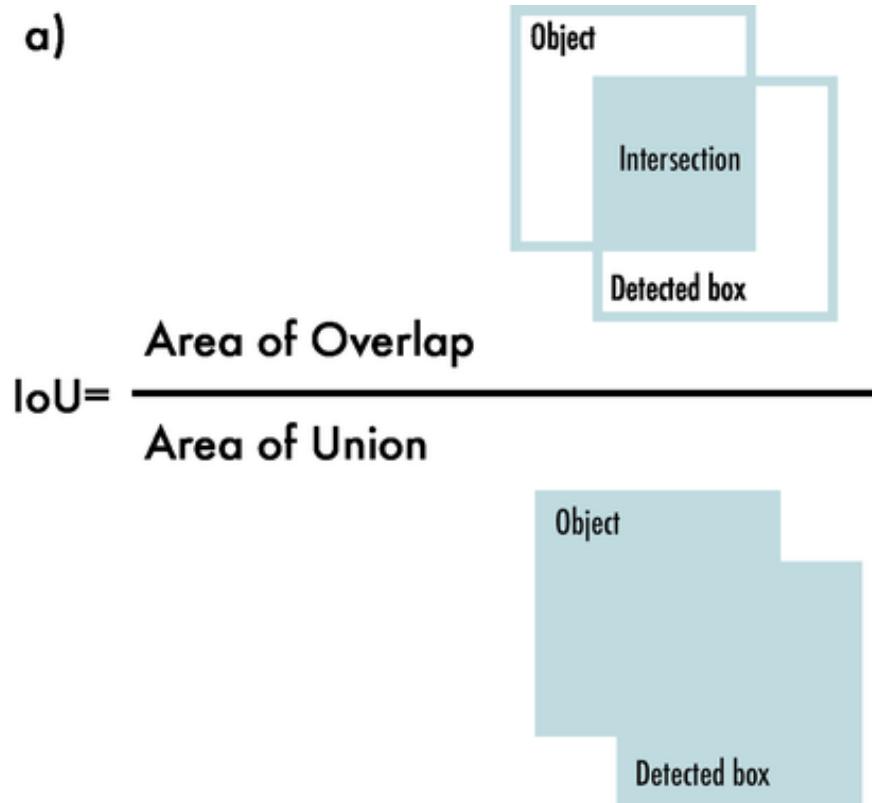


# 06 평가 결과 – IoU 기반 Average CER

## IoU(Intersection over Union)

- 정답(Ground Truth)과 예측(Prediction) 박스가 얼마나 겹치는지 확인 (50% 기준)
- IoU > 0.5 기준으로 Rough와 Tight 디텍션의 평균 CER을 산출하여 텍스트 영역 검출의 정확도를 평가

< IOU >



평가타입	Average CER(IoU > 0.5)
Rough 1	0.7813
Tight 1	0.3097
Rough 2	0.5633
Tight 2	0.3094
Rough 3	0.4944
Tight 3	0.2834
Rough 평균	0.613
Tight 평균	0.301

# 06 평가 결과 – PopEval Precision, Recall, F1-score

## PopEval

- PopEval은 Precision(정밀도), Recall(재현율), F1-score값을 중심으로 텍스트 박스 검출과 텍스트 인식을 종합 평가

## < PopEval >



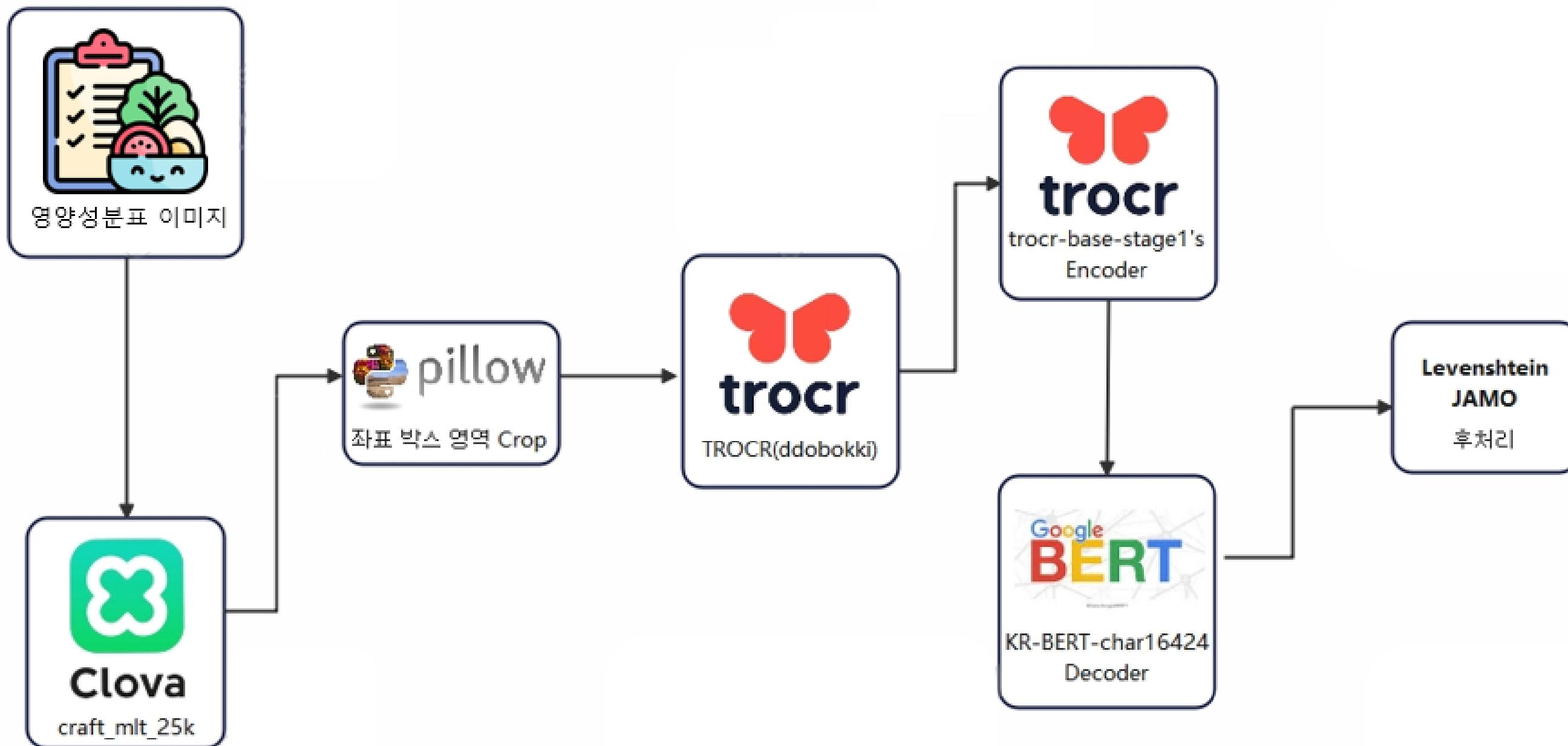
③ ① ②      ④ ⑤ ⑥  
 D E V I E W 2 0 1 9  
 X X E V V 2 0 1  
 ① ② ③      ④ ⑤ ⑥

$$\text{Recall} = \frac{\text{맞춘 글자 수}}{\text{정답 글자 수}} = \frac{6}{\text{len}("DEVIEW2019")} = \frac{6}{10} = 0.60$$

$$\text{Precision} = \frac{\text{맞춘 글자 수}}{\text{예측한 글자 수}} = \frac{6}{\text{len}("VIEW201") = 6 / 8 = 0.75}$$

평가타입	Precision	Recall	F1-score
Rough 1	92.5	74.9	82.8
Tight 1	89.4	84.3	86.8
Rough 2	91.3	75.5	82.7
Tight 2	87.6	83.4	85.4
Rough 3	93.5	76.3	84.0
Tight 3	89.3	83.9	86.5
Rough 평균	92.43	75.57	83.17
Tight 평균	88.77	83.87	86.23
전체 평균	90.6	79.72	84.7

## 06 워크플로우



OCR Project

감사합니다

문상흠 박창현 이동준  
위서현 조유경 한동우