OCR 개념 & 원리.

1page

OCR 이란? → 텍스트 인식.

이미지나 스캔된 문서에서 텍스트를 인식해 디지털 텍스트로 변환

VDU (visually-rich document understanding) 모델에서 OCR은 필수이다.
↳ 단순히 텍스트만 읽는 것이 아닌 문서의 시각요소(레이아웃, 구조, 텍스트, 이미지)까지 포함해서 문서의 의미와 의도를 파악함!

<OCR 단계>
이미지 전처리 (노이즈제거, 기울기, 대비조정) → 텍스트 영역감지 (bbox)
텍스트 추출 ←

OCR 특징
- 언어별 특화모델 필요
- 문서의 해상도, 품질이 성능에 영향을 준다.
- 이미지에 노이즈가 많거나 기울어져 있으면 정확도↓

NLP : 순수 텍스트 데이터 (txt, csv) 만 추출.
↕
VDU : 문서의 위치정보, 시각적 배치, 객체간 관계를 함께 이해. (시각적 문맥을 포함하는 AI)

VDU
- 문서 레이아웃
- OCR (텍스트 추출) ⇒ ~~Tesseract~~ / Paddle OCR ⇒ 딥러닝기반 높은 정확도, 한글 포함 다국어지원
  GPU 가속기 / 텍스트방향 / 수식 OCR / 테이블 OCR 감지
- 사람이 읽는 순서대로 문서요소 정렬 (Reading order prediction)
- 멀티모달 정보 통합 (시각정보 + OCR) → 문서의 의미와 논리적 구조까지 이해가능.

VDU의 핵심 기술 스택
- vision (모델) – YOLO, R-CNN — object detection.
- OCR : Paddle OCR / Tesseract — base라인코드에서는 Pytesseract로 텍스트 추출.
- 멀티모델 : Transformer → Donut / Layout LM / DOC Former
- 데이터셋 : DocVQA / FUNSD / PubLayNet / DocLayNet
- metrics
  - Layout detection : mAP
  - OCR : text accuracy
  - Reading order → 순서 예측 accuracy.

수식

Latex 추출방법 (일반 OCR로는 안됨) : Im2LaTeX 모델 (CNN + Attention 기반 딥러닝모델)
Paddle OCR + 수식전용모델 fine-tuning

OCR 성능개선
- 이미지 전처리
  이진화 (텍스트와 background 대비 극대화)
  Denoising (노이즈제거)
  Thresholding (임계값 적용)
  Deskewing (기울어진 문서를 회전보정)
- 모델 최적화
  Paddle OCR에서 rec_model은 언어별로 최적화된 모델로 변경 → 언어문제 해결.
  글자크기와 위치에 맞춘 det_model 선택.
  수식, 테이블 등 특수데이터에는 별도 특화모델 사용.
- 성능 튜닝 이미지 전처리
  이미지 해상도를 최소 300dpi로 보장.
  밝기, 대비 자동 보정 / 다국어 문서는 OCR 언어설정 (lang = "kor+eng")

base OCR모델

모델 성능 극대화 part.
OCR & 이미지 전처리

Multimodal : 서로 다른 Type의 데이터를 동시에 처리하는 AI.

↳ text / image / tabular / equation / speech (실제음성)

각각은 인코딩한다음, Transformer로 통합해서 복합적으로 처리한다.
(Attention 메커니즘)

<Transformer>

시작

- 이미지 → 작은 패치로 분할 → 임베딩 (벡터화)
- 텍스트 → 토큰화 → 임베딩
- 동일한 Transformer Encoder에 이미지 & 텍스트 벡터 입력
  이미지와 텍스트가 서로 attention 해서 의미를 찾음. (관계성)

Attention: 텍스트나 이미지 같이 서로 다른 Type의 데이터를 동일한 Transformer 임베딩 공간에 임베딩해서 서로의 의사관계에 집중할 수 있도록함. (self attention & multi attention)

문서 레이아웃 분석 (object detection)

문서(보고서, PDF, 논문)에서 제목, 본문, 표, 이미지, 수식, 캡션을 시각적 요소로 인지하고 구별함.

VDU에 있어 단순 OCR만 하게되면 문서의 구조를 읽기때문에 레이아웃 정의가 반드시 필요.

Bounding Box (bbox)
이미지에서 특정개체를 감싸는 사각형

Confidence Score
⇒ 예측 bbox가 실제 해당 클래스일 확률 (임계값 이상이면 해당개체로 판별)

MAP (object detection 평가지표)
MAP 값이 높을수록 precision 높음. (성능 good)

<YOLO 기반 문서분석> YOLOv11n - doclaynet (문서 전용 데이터셋으로 pretrained 된 모델)

YOLO : 단일 신경망으로 이미지 전체에서 객체를 한번에 탐지.
문서분석에서 텍스트, 표, 이미지, 수식등을 빠르게 탐지 가능. → 다양한 객체 동시탐색 가능.

YOLO 구조 : 이미지를 grid로 변환 / 각 셀이 객체 존재 여부 / bbox / 클래스 예측

<Transformer 기반 접근>

- Layout LM (Microsoft)
  OCR로 추출한 텍스트 토큰
  각 토큰의 BBox 좌표
  이미지 임베딩
  → Attention으로 문서복잡이해

- Donut
  네이버가 개발한 OCR-free 모델
  CNN + Transformer
  OCR단계 생략 → 속도 / 정확도 개선

img order prediction (텍스트의 논리순서)

정성 평가지표 35% 차지

가장 기본 방식 bbox 좌표기준 정렬 { 1차 정렬 : y좌표(세로) / 2차 정렬 : x좌표(가로) / 한계 : 왼쪽열 다 읽기 전에 오른쪽 열이 먼저 정렬됨. 테이블 내 순서 오류 } ⇒ 멀티컬럼 문서에 약함 / 간단하고 빠름.

- 그래프 기반 순서추론 (Text flow Graph) ⇒ 구조적, 논리순서에 강함. / 느리고 튜닝필요 (hyper parameter)

  Text block을 노드, 블록간 읽기관계를 엣지로 표현한 graph 구성

  위상정렬 (Topological Sort) 사용해 순서예측.

  - 각 텍스트 블록의 위치, 크기, 정렬 상태 비교
  - 인접블록 간 방향성 결정 (아래에 있는 블록 후순위)
  - 모든 노드를 그래프로 구성.
  - 위상정렬로 순서결정.

실전 챌린지 맞춤전략: Yolo로 BBox로 추출 (object detection) → PaddleOCR로 Text 추출 → Text flow graph order prediction. (텍스트 논리순서예측) ⊕ TTA (Test time Augmentation)

TTA: 추론할때 입력이미지를 여러방식으로 변형(증강)해서 여러결과를 추출 → 평균 or 앙상블하여 더 정확한 예측을 하는것.

TTA의 효과: 잘려있는 텍스트 박스 → 정렬에 유리 → TTA로 정확도개선 / OCR 에러로 Text 유실 → 여러 변형에서 텍스트 비교해 가장 신뢰도 높은 순서 측정.

↓

좌우회전 / flip / 해상도 변형 등 Augmentation → Voting / 평균 순서로 최종결정

Text flow graph ⊕ TTA 고도화 전략.

- 관계유형 → edge weight 줘서 soft graph 구성.
- 텍스트 정보 개선 보정.
- 임베딩 기반 의미 연결 : LayoutLM 처럼 텍스트 ⊕ BBox 임베딩 이웃블록이 다음블록과 논리적으로 이어지는가를 판단하는 classification 모델을 붙이는 것도 가능.
- 2단 레이아웃, 테이블이 섞인 문서 → 클러스터링 + 지역순서 정의.

Graph 기반 Top Down : 위치 좌표 위주 정렬 (bbox)

LayoutLM 활용 : 텍스트 의미 + 위치까지 고려한 정밀한 순서 판단 가능.

→ 서로 다른 기준으로 정렬 ⇒ Score Voting / Hybrid / 앙상블 구조로 결합

멀티모달 AI

- Vision + Language 모델: 이미지와 텍스트 각각 임베딩 → 하나의 임베딩 공간에서 사용 (이미지 & 텍스트)
- LayoutLM: 텍스트 토큰 + BBox (문서 구조 정보) + 이미지 특징을 하나의 Transformer로 처리.
- Donut: 이미지를 넣으면 바로 구조화된 텍스트를 디코딩 (OCR free 모델)
  - 메인 대신 보조라인 (엔드투엔드 추론 장점 & 실패케이스 백업용도)
- BLIP2 (이미지&텍스트 pretrained model): 캡셔닝/QA 등 범용 멀티모달

"Reading order prediction: Graph + LayoutLM"

KNN (후보제한)

Graph prior 후보엣지 같은 가벼운 규칙으로 그래프 제한

↓

LM Score

LayoutLM pair score (A,B) 쌍을 LayoutLM 입력으로 만들어 이어질 확률 추정 (hard / random negative로 학습)

↓

★Score fusion (late ensemble) ⇒ geometric ⊕ LM score

Topo sort 와 LayoutLM score ensemble

↓

A→B→C→A 구조 제거

w로 방향 가중치 만들고 사이클 제거 (가중치 낮은 edge drop)

위상정렬 & local tie breaker로 안정화

\# 학습·리소스 제약 → 해결 → LoRA로 타겟 수 / VRAM 사용 대폭 감소 (어댑터 가중치 저장)

↓

경량 fine-tuning

Score Fusion:

- Topo sort (기하학 score)
  - 기하학점수로 점수추출
  - KNN으로 후보축소 (리소스 절약)
  - 0°/90°/180°/270° TTA 적용
- LayoutLM (의미 score)
  - Pair score (LayoutLM 빌려기 점수)
  - LoRA (경량 fine-tuning)
  - hard/Random negative

→ 가중합 ensemble ⇒ 최종 그래프 edge 추출 ⇒ Topo sort로 최종 순서 결정

사이클제거: 가중치 큰 edge부터 넣되, 사이클 생기면 그 엣지는 패스

로컬 타이브레이커: indegree=0 후보가 많을때 자연스러운 읽기순서에 맞춰 우선순위를 정해서 블록결정

PPTX → 이미지 변환

YOLO / OCR / LayoutLM 모두 페이지 단위 이미지를 먹는다.

해상도 / 색공간 / 페이지 순서가 뒤틀리면 뒤에 전부 영향 ⇒ "이미지 전처리 중요!"

〈Albumentations 데이터 증강〉

- 레이아웃 보존
- YOLO BBox와 동일 변환을 적용해야 GT가 흔들리지 X
- OCR 학습에는 광학(Photometric) 변화가 특히 유효

YOLO 학습증강 (레이아웃 인식강화, 노이즈 전처리 강화)

⊕

OCR 크롭 증강 (노이즈가 껴있는 텍스트 bbox도 잘 인식하도록 광학위주 변형)

OCR 크롭 주의사항
- 심한 회전 / 퍼스펙티브 X
- train / validation split 전 증강함. => 검증/평가에는 미적용.

OCR 후 텍스트 정제
- 오자 정규화
- 화이트 스페이스/특수문 / 하이픈 처리 ⇒ 자연스러운 문장복원 (후처리)
- Latex / 수학 수식 정규화 (수식의 경우)

- BBox / order / text 라벨링관리 ⇒ 모델간 호환 위해 0-1000 정규화(정수) 권장
- Reading order V_1, V_2 모두 저장 ⇒ V_2 최종적용 (V_1은 디버깅용도)
- ID : doc_id + 페이지 index + block index ⇒ 재실행해도 동일 ID 할당

YOLO 문서탐지 (레이아웃 보정) → OCR 크롭 (텍스트 광학보정) → Paddle OCR (텍스트 추출) → 텍스트 후처리 (정제)

YOLO fine-tuning (멀티레이아웃탐지)

모델: yolov11n - doclaynet.pt

imgsz: 1024 / batch: 8~16

#클래스 불균형일경우 class weights 적용 / 과적합시 imgsz 960 / 896으로 낮춤.

#precision : 맞다고 예측한것중 실제로 맞은확률

$$\frac{TP}{TP+FP}$$

recall : 실제로 맞는것중 모델이 맞게 예측한 확률

$$\frac{TP}{TP+FN}$$

모델최적화

YOLO 조정: 80~120epoch, best AP 기준 선택
OCR 차이딩 retry threshold 튜닝
clean_text 룰 유효성 점검

LayoutLM pair classifier 학습비율
score function 가중치 하이퍼 파라미터 튜닝

rotated TTA (geo) 게이팅 임계값 조정
시간 예산 simulation → fail-safe 구조설계