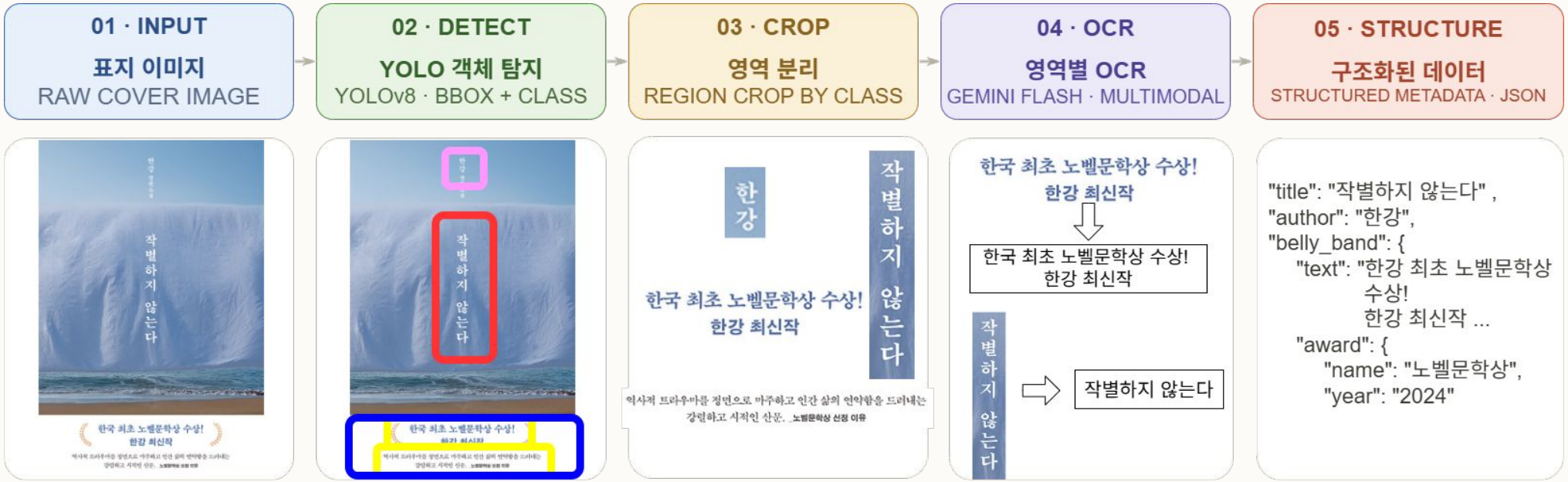
한국 소설 표지·띠지 텍스트 추출
- 객체탐지(YOLOv8)와 멀티모달 OCR(Gemini)의 활용



객체 탐지 OCR 어노테이션 Gemini 2.5 Flash

책 표지 이미지 내 띠지 영역을 탐지 - OCR로 띠지 텍스트 추출

표지와 띠지(belly band)란?

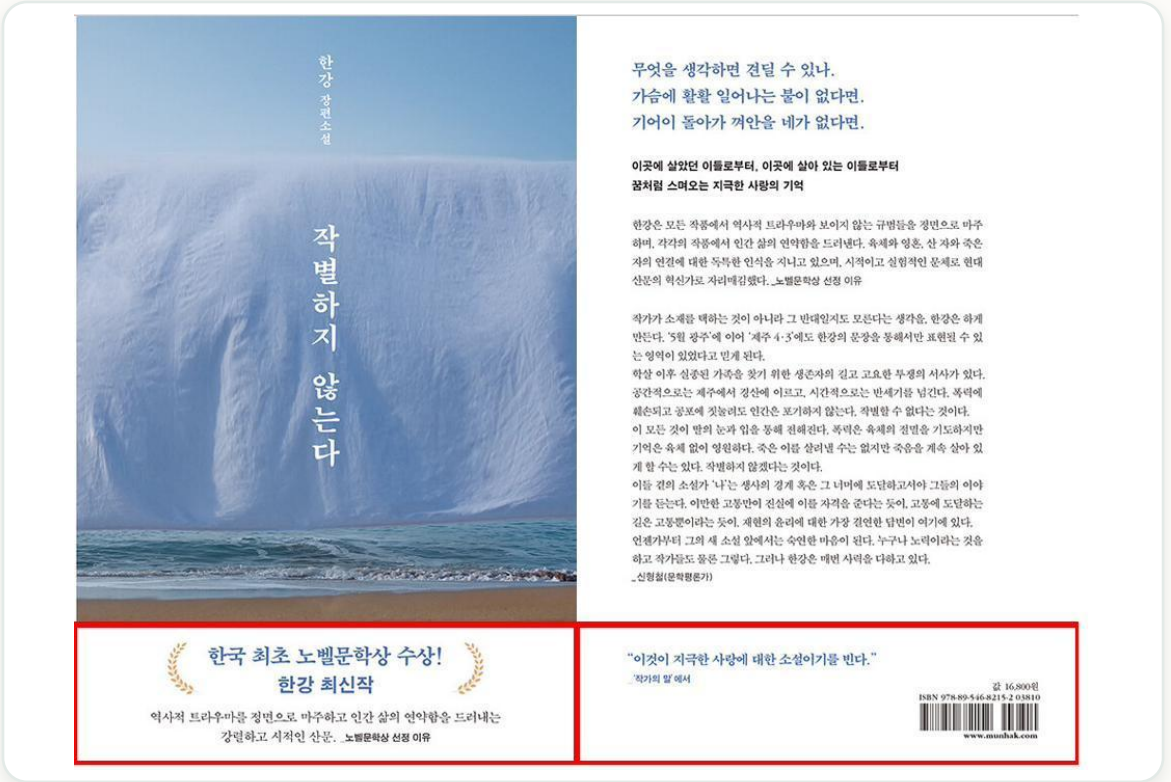
띠지는 책 표지를 가로로 감싸는 좁은 종이 띠다.

표지는 책의 내용을 한눈에 압축해 독자와 소통하는 첫 매체.
띠지는 그 위에 덧대어 책의 가치를 따로 강조하는 띠로,
일본어로는 오비(帯, obi)라 부름

한국·일본 출판에서 보편적으로 쓰이며, 본 표지와 달리
편집자가 자유롭게 교체할 수 있음

용어 · 띠지(obi)　용어 · belly band

핵심 정리
마케팅 카피, 수상 이력, 추천사, 판매 부수, 영상화 정보가 띠지에 압축적으로 표기됨



▲ 알라딘 상품 페이지 — 빨간 영역이 띠지에 해당한다

띠지에는 무엇이 담기는가

띠지는 편집자가 선별한 동적 메타데이터

- 마케팅 메시지 — 책의 핵심 가치를 한두 문장으로
- 수상 이력 — “노벨문학상 수상작가” 등
- 추천사 발췌 — 작가·평론가의 짧은 인용
- 판매 부수 — “100만 부 돌파”
- 미디어믹스 — 영화·드라마 원작 표기

인문학적 가치

어떤 추천인의 어떤 말이 표지화되는가는 모두 편집자의 의도적 선택. 추천인 네트워크·마케팅 텍스트 마이닝의 원천 데이터가 됨.

주의

띠지는 도서관·서점·소비자 단계에서 쉽게 폐기됨. 자동 수집·데이터화가 이뤄지지 않으면 영구히 사라질 수 있는 ‘휘발성 사료’. 그래서 디지털 인문학적 접근으로 보존 및 기록, 분석이 필요함.

영역 검출 우선(Detection-first) 파이프라인

먼저 영역을 나누고, 그 다음을 글자를 읽는다.

표지에 OCR을 바로 돌리면 본문과 띠지 글자가 섞이고, 그림을 글자로 오인식한다. 그래서 영역을 먼저 분리한다.

1 노이즈 제거
본문 외 시각 요소를 빼 인식 정확도를 높인다

2 구조 보존
추출 텍스트가 어떤 클래스인지 자동 표시된다

3 선택적 처리
필요한 영역만 OCR해 비용을 줄인다

▲ INPUT → DETECT → CROP → OCR → STRUCTURE

컴퓨터 비전의 두 가지 질문

위치를 찾는 일과 내용을 읽는 일은 서로 다른 기술이 맡는다.

①

어디에 텍스트가 있는가?

이미지 속 텍스트 영역의 위치를 찾는다

객체 탐지 (Object Detection)

②

무엇이라고 적혀 있는가?

찾아낸 영역의 문자를 읽어 낸다

광학 문자 인식 (OCR)

객체 탐지 모델 — YOLO

가볍지만 강력해, 고가 장비 없이 노트북에서도 돌아간다.

약 300만
파라미터 (nano 모델)

약 6 MB
모델 파일 크기

0.05초
CPU로 이미지 한 장 추론

7만 장
1시간에 분석 가능

객체 탐지는 이미지 속 객체를 '위치(bounding box)'와 '클래스(class)' 두 정보로 동시에 출력한다. 본 연구는 그중 가장 가벼운 YOLOv8 nano 모델을 핵심 모델로 채택했다.

설치 — 명령어 한 줄이면 끝난다

```
pip install ultralytics
```

핵심 정리

한 장의 이미지 안에 여러 객체가 있어도, 각각의 위치와 종류를 모두 한 번에 식별한다. 그래서 표지에서 제목·저자·띠지를 동시에 찾아낼 수 있다.

OCR은 어떻게 발전해 왔는가

세대가 거듭될수록 단순 인식을 넘어 시각적 문맥까지 이해한다.

세대	대표 기술	특징
1세대 · 전통 OCR	Tesseract, OCR4all	픽셀 기반 패턴 매칭, 인쇄 활자에 강하다
2세대 · 딥러닝 OCR	EasyOCR, PaddleOCR	CNN/Transformer 기반, 다양한 폰트·왜곡에 강하다
3세대 · 멀티모달 LLM	Gemini, GPT-4o, Claude	시각 문맥 이해·레이아웃 인식·자연어 후처리까지 통합

주의

띠지는 장식 폰트와 받침 처리가 까다롭다. 예비 실험에서 1·2세대는 오류가 잦았다(예: “채식주의자”→“재식주의자”, “노벨”→“노발”).

반면 3세대 멀티모달 LLM은 99% 이상의 정확도를 보였다.

어노테이션(annotation)이란 무엇인가

원시 데이터에 사람의 판단으로 의미 있는 라벨을 부여하는 작업이다.

컴퓨터 비전에서는 이미지 속 객체에 바운딩 박스를 그리고, 그 박스에 클래스 라벨을 붙이는 일을 말한다.

어노테이션은 지도학습(supervised learning)의 출발점이다. 모델은 라벨이 붙은 데이터를 보고 패턴을 익힌 뒤, 새 이미지에 그대로 적용한다.

핵심 정리

기계는 '띠지' 개념을 스스로 알지 못한다. 그래서 형식을 학습시켜야 한다. 어노테이션의 품질이 곧 모델 성능을 결정한다.

어노테이션의 확장

1,206장

사람이 직접 라벨링한 어노테이션

19,461장

모델이 자동으로 분류한 이미지

3,473권 (35.1%)

띠지를 보유한 것으로 확인된 책

2장

표지·띠지 어노테이션

Label Studio로 7개 클래스를 직접 라벨링하는 전 과정을 따라간다.

02

왜 Label Studio인가

여러 어노테이션 도구 중, 다음 다섯 가지 이유로 채택했다.

1. **오픈소스 · 무료**
   구독료나 사용량 제한이 없고, 코드가 공개돼 검증할 수 있다
2. **로컬 설치 가능**
   인터넷 없이 작동하고, 표지 이미지가 외부로 전송되지 않는다
3. **다양한 출력 포맷**
   YOLO·COCO·Pascal VOC로 내보내 학습에 바로 투입한다
4. **한국어 환경**
   라벨 이름을 한국어로 자유롭게 설정할 수 있다
5. **협업 지원**
   한 프로젝트에 여러 작업자를 두고 결과를 비교·검수한다

참고

기본 UI는 영어지만, 표시되는 라벨명을 한국어로 바꿀 수 있어 한국어 가이드라인을 세우기 좋다. 1,000장이 넘는 작업은 서버를 두고 여러 명이 분담하는 방식이 권장된다.

먼저 Python을 설치한다

프로그래밍 경험이 없어도 순서대로 따라 하면 된다.

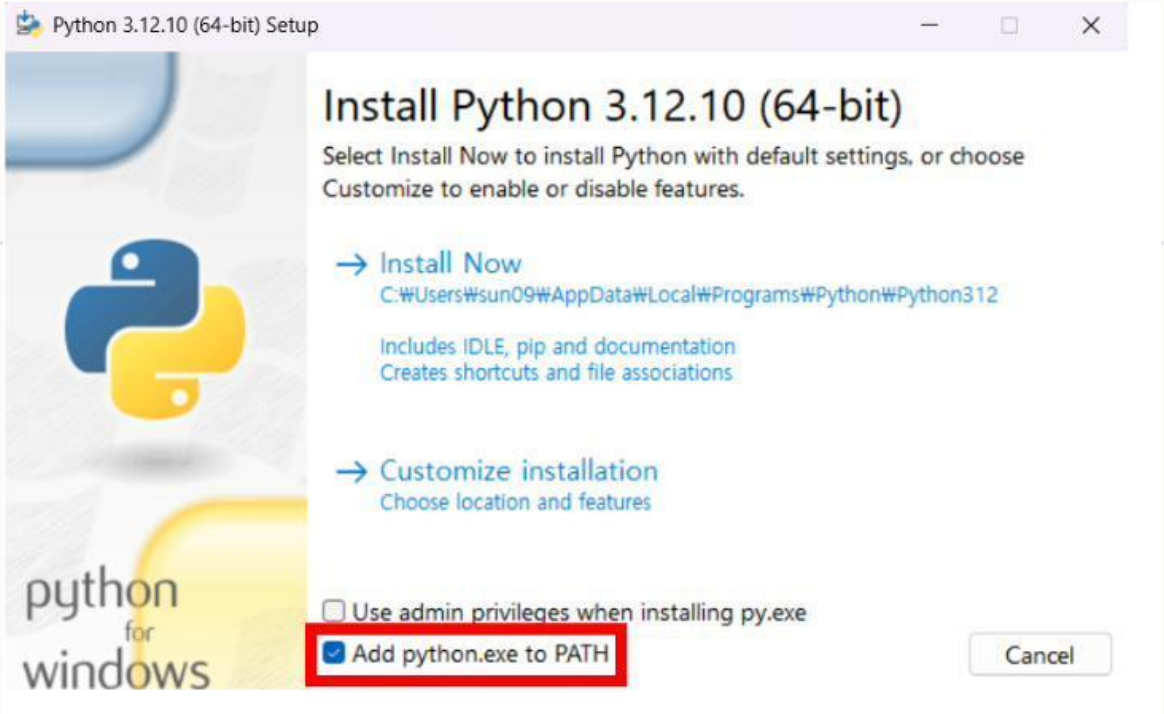
1 설치 파일 내려받기
3.12.10 버전의 exe 다운로드 주소로 직접 접속한다

2 설치 파일 실행
내려받은 exe를 두 번 클릭해 설치 창을 연다

3 ‘Add python.exe to PATH’ 체크
맨 아래 이 칸을 반드시 체크(✓)한다

4 ‘Install Now’ 클릭
‘Setup was successful’이 뜨면 설치 완료

5 설치 확인
cmd 창에 python --version 을 입력한다



▲ ‘Add python.exe to PATH’ 체크 화면

주의

python.org 첫 화면의 큰 노란 버튼은 누르지 않는다. 최신 버전이 깔리면 Label Studio가 작동하지 않으므로 **3.12.10**만 받는다.

먼저 Python을 설치한다

프로그래밍 경험이 없어도 순서대로 따라 하면 된다.

> **주의**
>
> python.org 첫 화면의 큰 노란 버튼은 누르지 않는다. 최신 버전이 깔리면 Label Studio가 작동하지 않으므로 **3.12.10**만 받는다.

명령어 한 줄로 Label Studio를 설치한다

‘pip’가 인터넷에서 필요한 프로그램을 자동으로 내려받아 설치한다.

1 명령 프롬프트 열기
검색창에 cmd 를 입력하고 ‘명령 프롬프트’를 연다

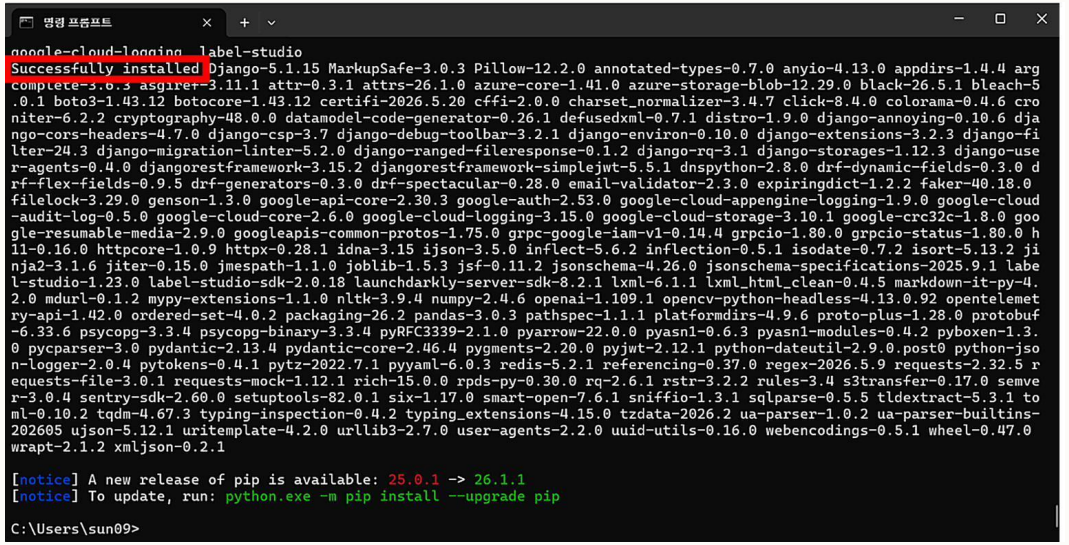
2 설치 명령어 입력
아래 명령어를 입력한다(설치까지 1~5분)

3 서버 실행
설치가 끝나면 서버를 실행한다

4 계정 생성
localhost:8080 화면에서 이메일로 회원가입한다

명령 프롬프트(cmd)

```
pip install label-studio

label-studio start
```



▲ ‘Successfully installed’ 출력 — 설치 완료

새 프로젝트를 만든다

‘Create Project’를 누르고 세 단계로 진행한다.

1 Project Name

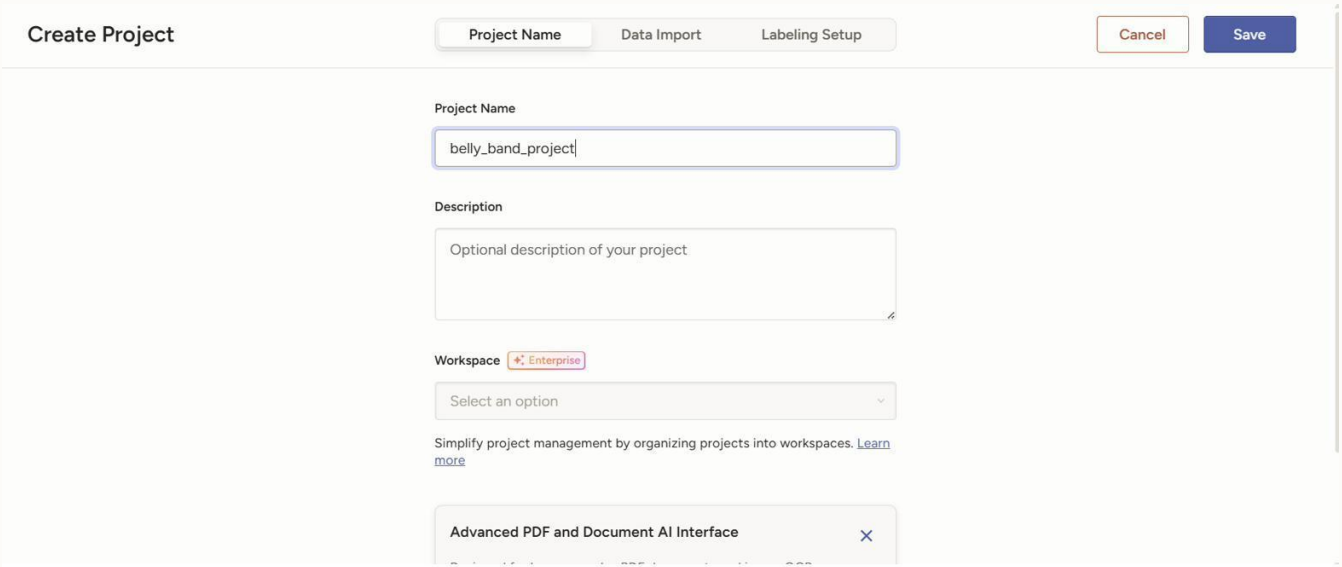
프로젝트 이름을 정한다 (예: belly_band_annotation)

2 Data Import

어노테이션할 이미지를 업로드한다

3 Labeling Setup

라벨의 종류와 클래스를 정의한다



▲ ① 이름 지정

▲ ② 데이터 업로드

▲ ③ 라벨링 설정

7개 클래스를 정의한다

Labeling Interface의 ‘Code’ 탭에 XML로 입력한다.

클래스	정의
belly_band	띠지 전체 영역
belly_band_text	띠지 안의 인쇄된 텍스트
title	책 제목
author	저자명
publisher	출판사 로고·표기
back_cover_text	뒷표지 추천사·줄거리·인용
barcode	ISBN 바코드

Labeling Interface · Code 탭

```
<View>
  <Image name="image" value="$image"/>
  <RectangleLabels name="label" toName="image">
    <Label value="title"  background="#FFA500"/>
    <Label value="author" background="#008000"/>
    <Label value="belly_band" background="#FF3333"/>
    <Label value="belly_band_text" background="#3333FF"/>
  </RectangleLabels>
</View>
```

참고
<Label> 한 줄이 곧 클래스 하나다. 이름과 색만 바꾸면 된다.

마우스로 바운딩 박스를 그린다

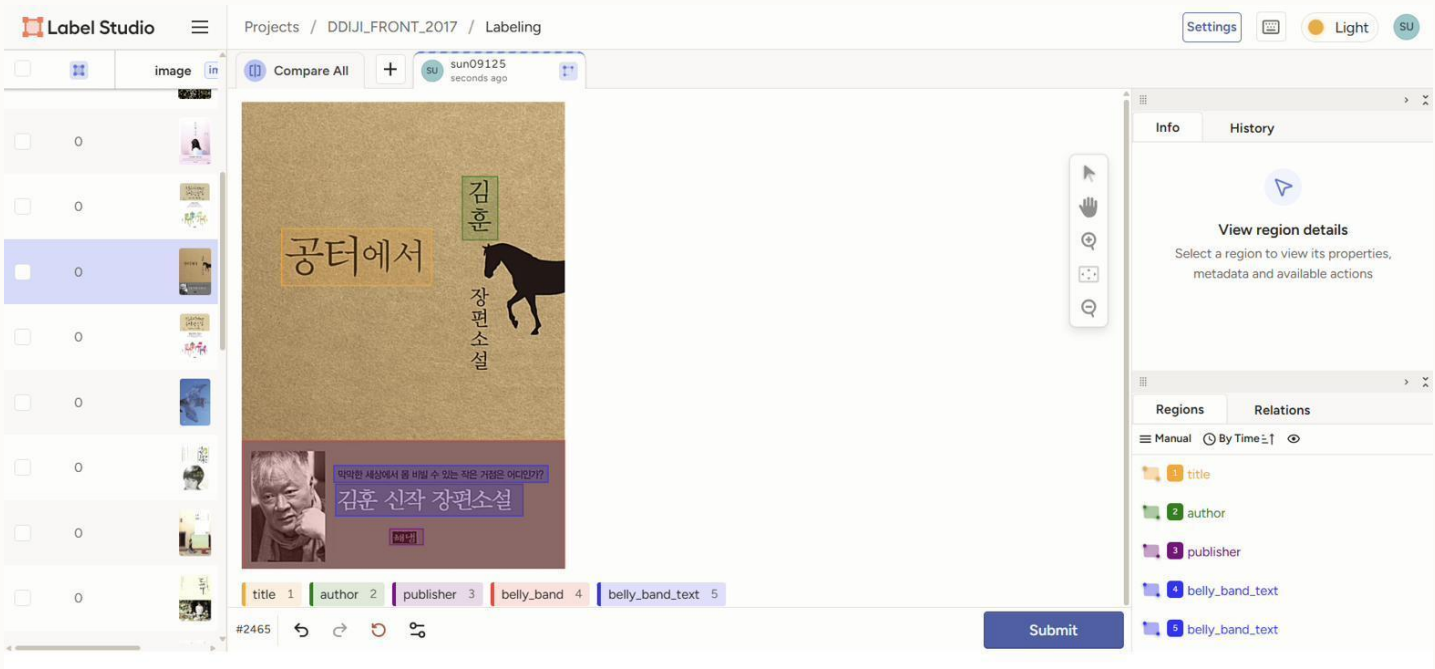
라벨을 먼저 고르고, 대각선으로 드래그해 사각형을 그린다.

1 클래스 선택
라벨 버튼을 누르거나 숫자 키(1~5)로 고른다

2 바운딩 박스 그리기
객체의 경계에 밀착시켜 드래그한다

3 박스 수정
모서리를 끌어 크기·위치를 조정한다

4 저장 · 다음
‘Submit’을 눌러 다음 이미지로 넘어간다



▲ 띠지가 있는 경우

작업 속도를 높이는 방법

단축키를 적극 쓰면 작업이 2~3배 빨라진다.

단축키	기능
숫자 키 1 ~ 9	해당 번호의 클래스 선택
Ctrl + Enter	저장 후 다음 이미지로 이동
Ctrl + Z	직전 작업 취소
Delete	선택한 박스 삭제
+ , −	이미지 확대 · 축소
Space + 드래그	화면 이동(패닝)

박스가 겹칠 때는 작은 박스를 먼저, 큰 박스를 나중에 그린다. 큰 박스를 먼저 그리면 그 위에 작은 박스를 그리기 어렵다.

같은 클래스라도 글자 크기·서체·내용이 뚜렷이 다르면 하나로 묶지 말고 각각 박스를 그린다. 모델이 경계를 더 분명하게 학습한다.

3장

객체 탐지 학습

어노테이션 데이터를 YOLOv8로 학습해 자동 검출 모델을 완성한다.

03

적은 데이터로 학습하는 비결 — 전이학습

백지가 아니라, 이미 사전학습된 모델에서 출발한다.

학습(training)은 '이미지와 정답 박스' 쌍을 반복해 보며 어떤 영역이 띠지·제목인지 스스로 찾아내는 과정이다. 그 결과는 하나의 가중치 파일(best.pt)로 저장된다.

본 연구는 가장 가벼운 nano 모델(yolov8n)을 쓰되, 대규모 데이터(COCO)로 사전학습된 모델에서 출발한다.

용어 · 전이학습(transfer learning)

용어 · 가중치 파일 best.pt

핵심 정리

사전학습 모델에서 출발하므로, 1,200여 장의 비교적 적은 어노테이션만으로도 7개 클래스를 추가로 가르쳐 실용적 성능에 도달한다.

COCO 사전학습 모델

대규모 데이터로 미리 학습된 출발점

+ 1,206장 어노테이션

띠지·제목 등 7개 클래스를 추가 학습

= 완성된 검출 모델

수만 장에 자동으로 영역을 검출한다

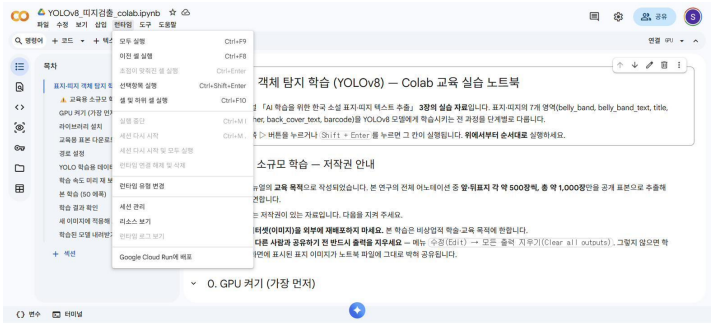
무료 GPU는 Google Colab에서 켠다

GPU가 있으면 학습이 CPU보다 수십 배 빠르다.

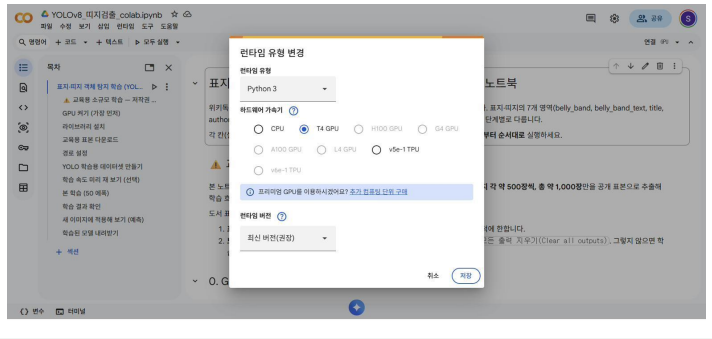
1 노트북 열기
제공된 .ipynb를 Colab에 업로드해 연다

2 런타임 유형 변경
‘런타임 → 런타임 유형 변경’ 메뉴로 들어간다

3 T4 GPU 선택
하드웨어 가속기를 ‘T4 GPU’로 바꾸고 저장한다



▲ ‘런타임’ 메뉴



▲ 하드웨어 가속기 → T4 GPU

주의
이 설정을 빠뜨리면 학습이 CPU로 진행되어 매우 느려진다. 반드시 확인한다.

학습 데이터의 형식과 분할

이미지와 라벨(.txt)을 짝지어 학습한다.

▲ YOLO 라벨(.txt) 형식

라벨 한 줄 예시

```
0  0.523  0.812  0.945  0.183
```

다섯 숫자는 클래스 번호, 박스 중심 x·y, 너비, 높이를 뜻한다.
좌표는 이미지 크기에 대한 0~1 상대 비율이다.

Train
840장
약 70% · 직접 보고 학습

Val
186장
약 15% · 과적합 점검

Test
180장
약 15% · 최종 평가

노트북을 위에서부터 차례로 실행한다

아홉 개의 셀을 다섯 단계로 묶어 순서대로 누른다(Shift+Enter).

1. **라이브러리 설치**
   pip install ultralytics 로 YOLOv8을 설치한다
2. **표본 다운로드 · 경로 설정**
   허깅페이스에서 표본을 자동으로 내려받는다
3. **YOLO 데이터셋 만들기**
   어노테이션을 변환하고 train·val·test로 나눈다
4. **본 학습 (50 에폭)**
   진행 상황이 에폭마다 표로 출력된다
5. **결과 확인 · 모델 내려받기**
   best.pt를 본인 컴퓨터로 다운로드한다

참고

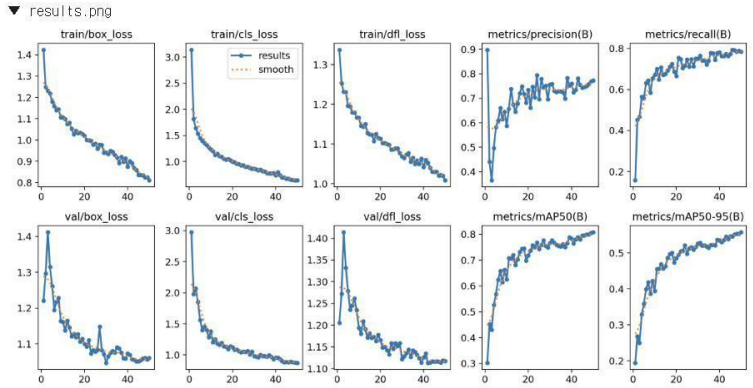
필요한 준비물은 '구글 계정' 하나뿐이다. 표본 데이터는 공개 저장소에서 자동으로 받아오므로 로그인·토큰이 필요 없다.

팁

'에폭(epoch)'은 학습 데이터 전체를 처음부터 끝까지 한 번 훑는 단위다. 50 에폭은 이 과정을 50번 반복한다는 뜻이다.

학습이 잘 됐는지 확인한다

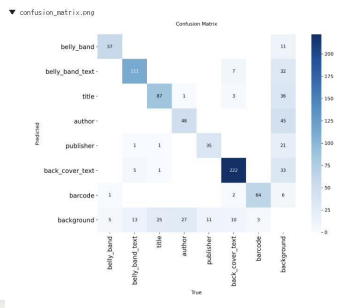
손실은 내려가고 mAP는 올라가다 완만해지면 정상이다.



▲ results.png — 손실·mAP 변화 그래프

```
Epoch   GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
46/50      2.3G      0.835     0.6735       1.03         12        640: 100% ──────────── 44/44 3.4it/s 12.9s
           Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% ──────────── 5/5 2.5it/s 2.0s
             all        144        740      0.746      0.794      0.795      0.545

Epoch   GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
47/50      2.3G     0.8338      0.653      1.021         12        640: 100% ──────────── 44/44 3.7it/s 11.9s
           Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% ──────────── 5/5 2.7it/s 1.9s
             all        144        740      0.749      0.793        0.8      0.551

Epoch   GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
48/50      2.3G     0.8238     0.6507      1.022         12        640: 100% ──────────── 44/44 3.5it/s 12.5s
           Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% ──────────── 5/5 3.3it/s 1.5s
             all        144        740      0.758      0.784      0.799      0.553

Epoch   GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
49/50      2.3G     0.8252     0.6434       1.02         12        640: 100% ──────────── 44/44 3.5it/s 12.7s
           Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% ──────────── 5/5 2.5it/s 2.0s
             all        144        740      0.769      0.789      0.805      0.554

Epoch   GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
50/50      2.3G     0.8102     0.6418       1.009         12        640: 100% ──────────── 44/44 3.7it/s 11.8s
           Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% ──────────── 5/5 1.8it/s 2.7s
             all        144        740      0.772      0.784      0.808      0.557
```

▲ 학습 로그 — 에폭마다 출력된다



▲ confusion_matrix.png

핵심 정리

실제 학습(전체 1,206장)에서 7개 클래스 평균 mAP@0.5 = 0.779를 얻었다. 핵심 클래스 belly_band는 정밀도 84% · 재현율 91%로 띠지 추출에 충분한 성능을 보였다.

4장

표지·띠지 텍스트 추출 (OCR)

04

검출된 띠지 영역에 OCR을 적용해 텍스트를 뽑고 구조화한다.

OCR을 두 단계로 나눈다

같은 모델(Gemini 2.5 Flash)을 쓰되, 입력 형식이 완전히 다르다.

Stage 1 이미지에서 원문 추출

검출된 띠지 박스를 잘라 Gemini에 보내 글자를 그대로 받는다.

이미지 → 평문 텍스트 (rawText)

OCR을 두 단계로 나눈다

같은 모델(Gemini 2.5 Flash)을 쓰되, 입력 형식이 완전히 다르다.

Stage 2 의미 블록으로 분해

평문을 다시 보내 추천사·수상·카피 등 11종으로 나눈다.

평문 → 구조화된 JSON

Gemini API 키를 발급받는다

Google AI Studio에서 무료로, 1분이면 된다.

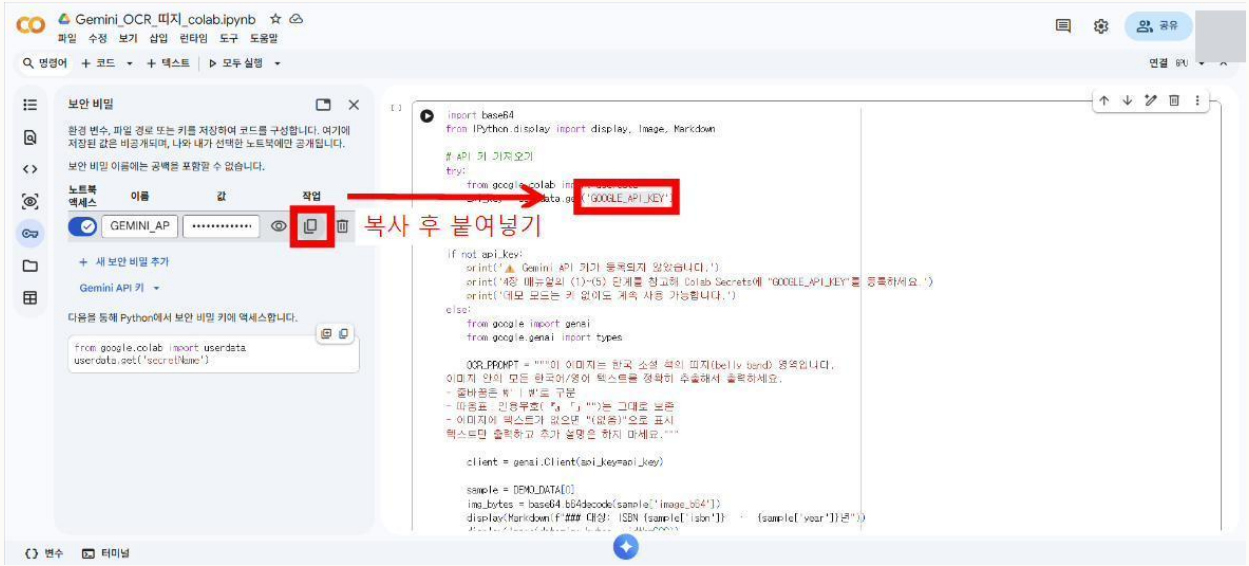
1 AI Studio 접속
aistudio.google.com 에 구글 계정으로 로그인한다

2 API 키 생성
‘Get API key’ → ‘키 만들기’ (AIza로 시작, 39자리)

3 키 복사
복사 아이콘을 눌러 안전한 곳에 따로 보관한다

4 Colab 보안 비밀에 등록
GOOGLE_API_KEY로 등록하고 ‘노트북 액세스’ 토글을 켠다

5 노트북에서 불러오기
userdata.get('GOOGLE_API_KEY') 로 가져온다



▲ Colab ‘보안 비밀’에 키 등록

주의

API 키는 절대 외부로 유출돼서는 안 되는 개인정보다. 다른 사람과 공유하거나 LLM에 입력해서도 안 된다.

짧고 명확한 OCR 프롬프트

모델에게 '무엇을, 어떻게' 출력할지만 알려 준다.

OCR_PROMPT

```
이 이미지는 한국 소설 책의 띠지(belly band)
영역입니다. 이미지 안의 모든 한국어/영어
텍스트를 정확히 추출해서 출력하세요.

- 줄바꿈은 ' | '로 구분
- 따옴표·인용부호(『』「」"")는 그대로 보존
- 텍스트가 없으면 "(없음)"으로 표시

텍스트만 출력하고 추가 설명은 하지 마세요.
```

다섯 가지 설계 결정

- **도메인 맥락 명시**
  — "띠지다"라고 알려 주면 정확도가 오른다
- **줄바꿈을 | 로 보존**
  — 시각적 줄 구분을 평문에 남긴다
- **인용부호 보존**
  — 『』「」가 추천사 식별의 단서다
- **'(없음)' 표기**
  — 빈 결과를 정형 문자열로 구분한다
- **설명 금지**
  — 군더더기를 막아 파싱을 단순화한다

텍스트를 11가지 의미 블록으로 나눈다

추천사·수상·언론 인용 등 유형으로 구조화한다.

블록 유형	의미
endorsement	외부 인물의 추천·평
pressCitation	언론·기관의 인용·평
mediaMention	매체 노출·언급
award	수상·후보
selection	매체·서점의 선정(수상 아님)
sales	판매 실적·해외 판권

블록 유형	의미
adaptation	영상화 언급
seriesBranding	시리즈 소속 표기
publisherCopy	출판사 광고 카피
authorNote	작가의 말
textExcerpt	본문 발췌
공통 필드	type · content · 발화자 등

참고

호출 시 response_mime_type='application/json'으로 출력을 강제하고, temperature=0.1로 매번 같은 결과가 나오게 한다.