

## 01 YOLO

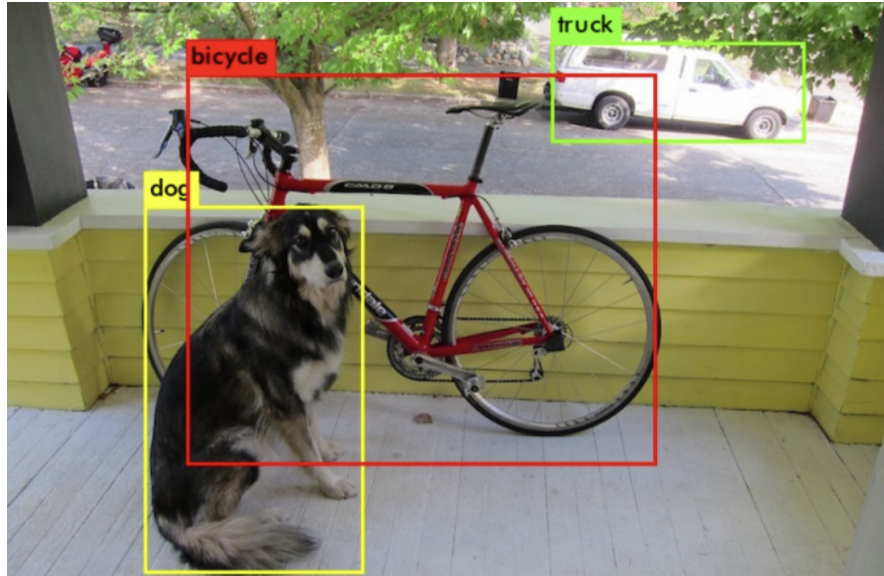
AI 에이전트 개발

# YOLO (You Only Look Once)

원티드랩

- [1. Object Detection: 개념과 원리](#)
- [2. YOLO](#)
- [3. 학습](#)
  - [1. 데이터 만들기](#)
  - [2. 모니터링 지표](#)
  - [4. 학습 결과 성능 지표](#)
- [4. 활용 예](#)

# 1. Object Detection: 개념과 원리



- 이미지를 입력받아 그 안에 있는 **사물(객체)**을 인식하고, 어디에 있는지 위치(**Box**)를 지정하는 기술
- 단순히 “고양이다”라고만 인식하는 건 **Image Classification**
- “고양이가 화면 오른쪽 아래 있어!”까지 알려주는 게 **Object Detection**

## 2. YOLO

**YOLO(You Only Look Once)** 는 이미지나 영상 속 객체(Object)를 빠르게 탐지해주는 인공지능 모델이다.

- YOLO는 이미지를 단 한 번(Only Once) 살펴보고, 그 안에 있는 사람, 자동차, 고양이, 우산 등 다양한 객체를 실시간으로 탐지할 수 있다.
- 정확도와 속도를 동시에 고려해 설계되어, CCTV, 드론 영상, 교통 감시, 셀프 계산대 등 현실적인 환경에서 매우 잘 작동한다.

### 1. Precision (정밀도)

- 정의: “예측한 것 중에 얼마나 맞았나?”
- 공식:

$$Precision = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}$$

### 2. Recall (재현율)

- 정의: “실제로 있는 것 중 얼마나 놓치지 않았나?”
- 공식:

$$Recall = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}}$$

### 3. Precision vs Recall

- Precision ↑ = 헛발질 줄이기 (있다고 한 건 확실히 있음)
- Recall ↑ = 놓치지 않기 (있는 건 다 잡음)
- 둘은 보통 **Trade-off**(서로 반비례) 관계라서 균형이 중요

#### 4. mAP (mean Average Precision)

- 정의: Precision과 Recall을 여러 IoU 기준에서 종합적으로 평가한 점수
- 설명:
  - 먼저, **Precision-Recall** 곡선을 그림곡선 아래 면적(AP, Average Precision)을 계산
  - 여러 클래스/IoU에 대해 평균 낸 게 **mAP**
- 표현 방식:
  - [mAP@0.5](#) : IoU ≥ 0.5 기준
  - [mAP@0.5:0.95](#) : IoU를 0.5~0.95까지 0.05 간격으로 측정 후 평균

#### 4. IoU (Intersection over Union)

- 정의: 예측한 박스와 정답 박스가 얼마나 겹치는지 측정
- 공식:

$$IoU = \frac{\text{겹치는 영역}}{\text{합집합 영역}}$$

### 3. 학습

#### 1. 데이터 만들기

<https://roboflow.com/>

Roboflow는 이미지 데이터셋을 손쉽게 만들고, 라벨링하고, 학습을 위한 형식으로 내보낼 수 있는 웹 기반 플랫폼이다.

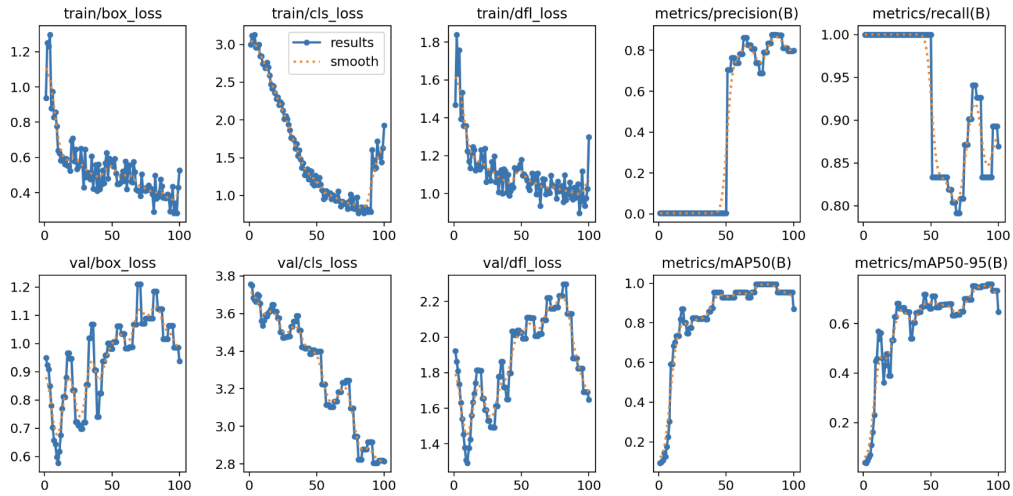
- 브라우저만 있으면 누구나 쉽게 이미지에 객체 라벨을 지정(=bounding box) 할 수 있다.
- YOLO에 맞는 형식으로 자동 변환 및 내보내기(export)가 가능하며, 학습 파이프라인 구축이 매우 간단해진다.

#### 2. 모니터링 지표

Epoch	GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size
1/100	0G	0.9372	2.995	1.468	37	640: 100%  1/1 [00:04<00:00, 4.22s/it]
	Class	Images	Instances	Box(P	R	mAP50 mAP50-95): 100%  1/1 [00:00<00:00,
	all	4	4	0.00336	1	0.0942 0.0395

항목	의미	예시 값	해석/비유
Epoch	전체 학습 반복 횟수 중 현재 단계	1/100	100번 중 1번째 학습 중
GPU_mem	GPU 메모리 사용량	0G	GPU 안 쓰고 CPU로 학습 중
box_loss	바운딩 박스 위치 예측 오차	0.9372	박스 위치를 얼마나 잘 맞췄는가 (낮을수록 좋음)
cls_loss	객체 클래스 분류 오차	2.995	무엇인지 맞췄는가 (낮을수록 좋음)
dfl_loss	박스 좌표 세부 분포 학습 오차	1.468	박스 테두리를 더 정밀하게 맞추는 값
Instances	현재 배치에서 학습된 객체 수	37	학습에 사용된 라벨 개수
Size	입력 이미지 크기 (학습용 변환 크기)	640	모든 이미지를 640×640으로 변환
Images	검증에 사용된 이미지 수	4	현재 validation 데이터 이미지 개수
Instances(val)	검증 데이터의 객체 수	4	validation에서 라벨링된 객체 개수
Box(P)	Precision (정밀도)	0.00336	"예측 중에 맞은 비율" (아직 낮음)
R	Recall (재현율)	1.0	"실제 객체를 모두 잡아냈는가" (100%)
mAP@50	IoU 0.5 기준 평균 정확도	0.0942	예측 박스와 정답 박스가 50% 이상만 겹치면 맞은 것으로 인정
mAP@50-95	IoU 0.5~0.95 종합 평균 정확도	0.0395	약 3.9%, 더 학습해야 개선됨

#### 4. 학습 결과 성능 지표



## 1) Loss 곡선

- **train/box\_loss, cls\_loss, df\_l\_loss**
  - 학습 데이터에 대한 손실값 (오차 정도)
  - 전체적으로 점점 감소 → 모델이 점점 더 정확하게 학습하고 있음
- **val/box\_loss, cls\_loss, df\_l\_loss**
  - 검증 데이터에 대한 손실값
  - 학습과 비슷하게 안정적으로 감소해야 "일반화"가 잘 된 것
  - 만약 학습은 줄고 검증은 증가한다면 **과적합** 신호

## 2) Precision & Recall

- **Precision**: 예측한 것 중 맞춘 비율
  - 일정 Epoch 이후 급격히 증가 → 모델이 "헛발질"을 줄임
- **Recall**: 실제 있는 것 중 맞춘 비율
  - 초반에는 1.0(모두 탐지) → 이후 점차 떨어지며 안정화
  - Precision과 Recall은 보통 **Trade-off** 관계 → 균형이 중요

## 3) mAP 지표 (오른쪽 아래 영역)

- **mAP@50**: IoU 50% 기준 → 상대적으로 관대한 기준
  - 학습이 진행되면서 0.9 이상으로 안정화 → 모델이 대부분의 객체를 잘 잡음
- **mAP@50-95**: IoU 50~95% 다양한 기준 평균 → 더 엄격한 평가
  - Epoch이 쌓일수록 점점 올라가며 안정화
  - 모델의 실질적 성능(정밀도 + 위치 정확도)를 보여줌

## 4. 활용 예

- 자율주행 자동차
  - 도로 위의 보행자, 신호등, 차량, 자전거 등을 실시간으로 탐지
  - 사고를 예방하고 차선 변경, 긴급 제동 같은 기능을 지원
- 보안/안전 관리
  - CCTV에서 침입자나 이상행동 탐지
  - 건설 현장에서 안전모, 안전조끼 착용 여부 자동 확인
  - 공장에서 위험 구역에 들어가는 사람이나 차량 자동 경고

- 소매점/리테일

- 계산대 없이 매장에서 물건을 집으면 자동으로 인식 (아마존 Go 같은 무인 편의점)
- 상품 진열 상태 모니터링, 품질 여부 확인

- 의료 영상 분석

- X-ray, CT, MRI 영상에서 병변(폐 결절, 종양 등)을 자동 탐지
- 의사의 진단 보조 시스템으로 활용

- 드론/로보틱스

- 드론이 하늘에서 교통 상황 모니터링, 화재 지역 탐지, 재난 구조 대상 탐색
- 로봇이 물체를 찾아 집는 작업(픽 앤 플레이스)에 활용

- 스마트 시티

- 교차로 교통량 분석, 불법 주정차 탐지
- 쓰레기 무단 투기 감지 및 도시 안전 관리