

덕성여자대학교 & 해성여자고등학교

2025학년도 꿈이룸 창의융합인재 두번째 시간

# 인공지능기술로 할 수 있는 일을 직접 실습해보기



하윤정 (hayunjong83@gmail.com)

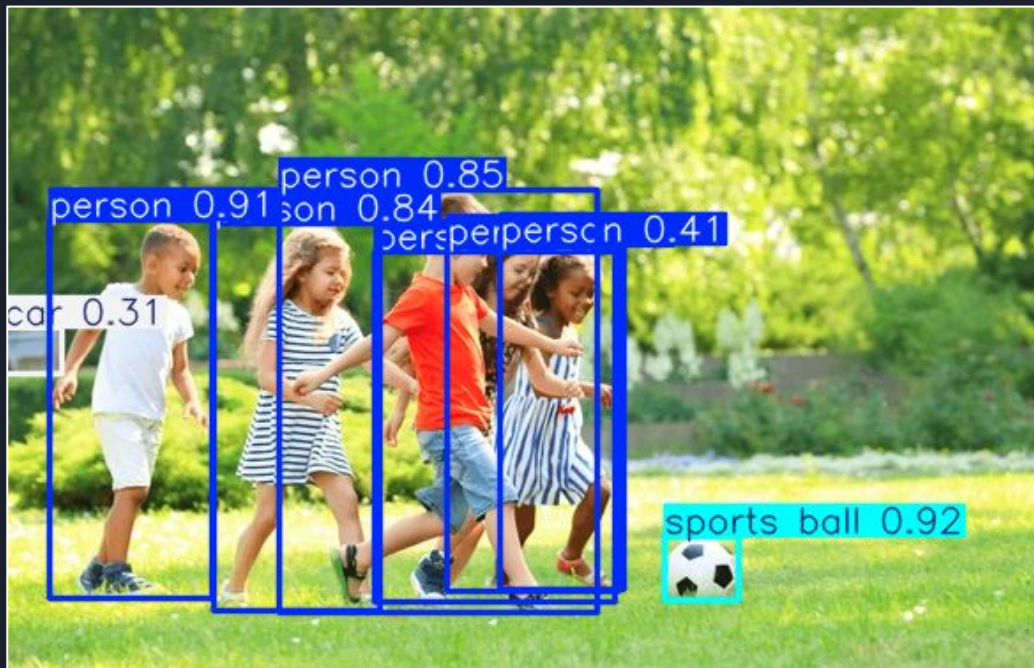


## 실습 1.

# Yolov11을 이용한 객체 탐지 실습

# 객체 탐지(Object Detection)

물체를 찾아내고 그 위치를 표현하는 기술

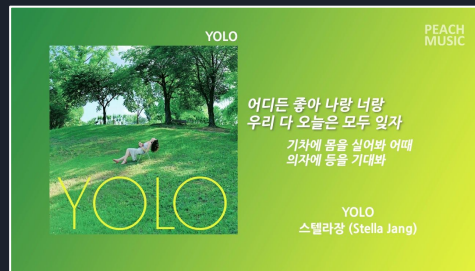


- 물체의 종류와 위치를 동시에 알아낸다.
- 바운딩박스(bounding box)와 클래스 레이블이 함께 출력된다.
- 실시간으로 입력되는 영상에도 객체 탐지 기술을 적용할 수 있다.

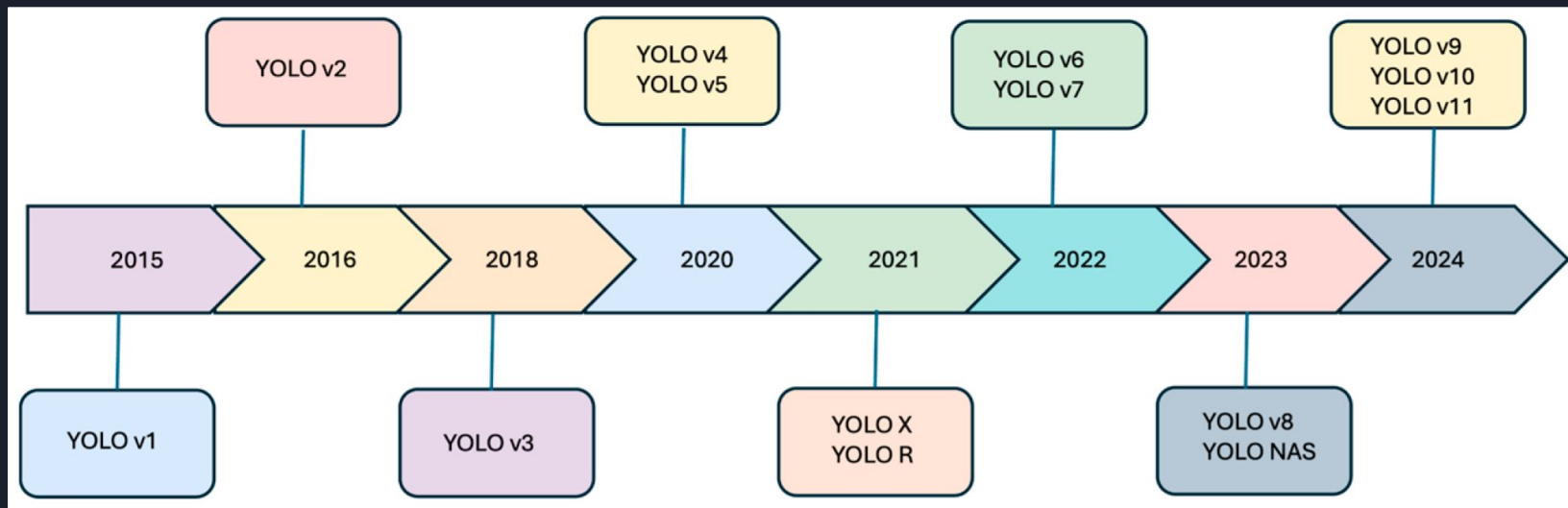
# YOLO(You Only Look Once)

## 실시간 객체 탐지 모델의 대명사

- 한 번에 전체 이미지를 분석하여 빠르게 탐지한다.
- 단순한 모델 구조(architecture)를 갖고, 매우 빠른 탐지 예측 속도
- CCTV와 같은 실시간 검출 시스템에 많이 사용된다.



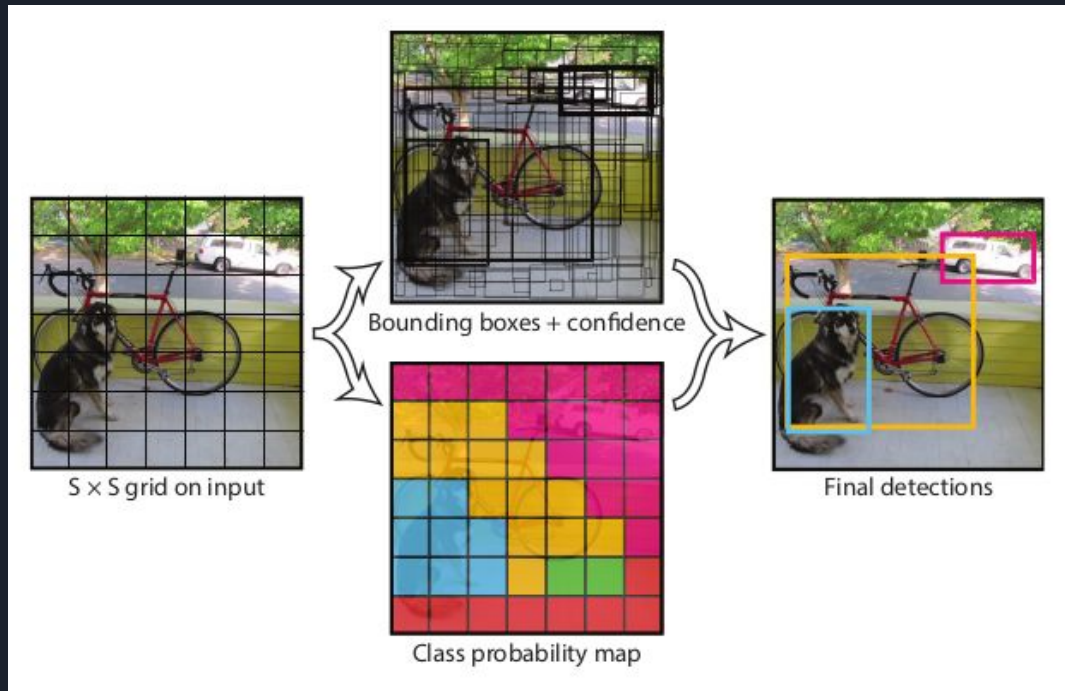
스텔라장(Stella Jang) - YOLO



2025년 현재 yolov12까지 공개되었다.

# YOLO의 작동 원리

한 번에 전체 이미지를 분석해서, 물체의 종류(class)와 위치(bounding box)를 동시에 예측

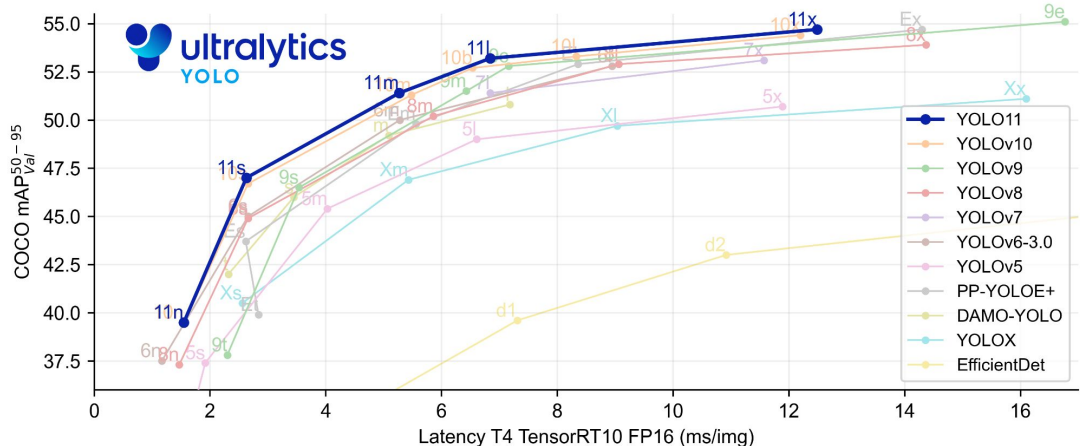


1. 입력 이미지를 분할  
(  $S \times S$  개의 그리드로 나눈다.)
2. 그리드의 각 셀마다 다음을 각각 예측한다.
  - B개의 바운딩 박스 좌표
  - 각 박스의 신뢰도
  - C개의 클래스 확률
3. 같은 물체를 중복탐지하지 않도록 후처리를 거쳐서 최종 결과 획득

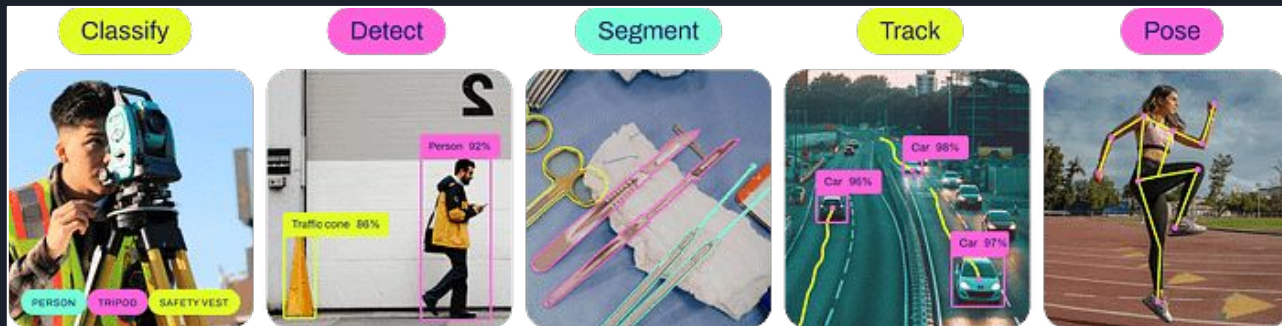
# YOLOv11

## 다양한 모델 크기

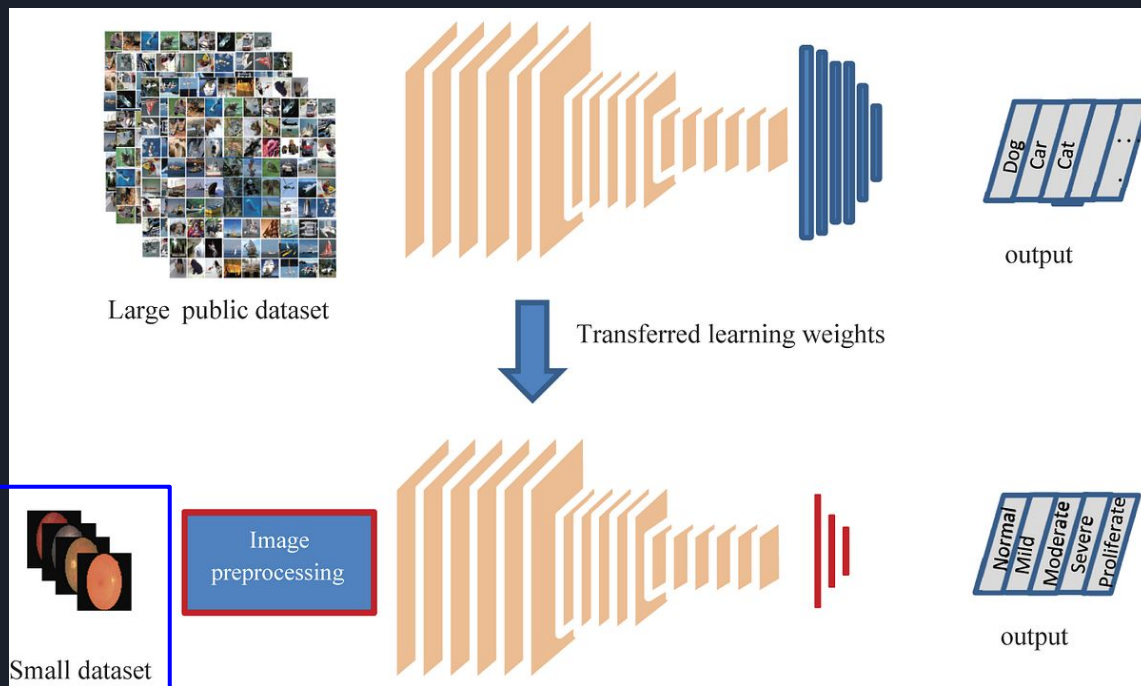
- n ( Nano )
- s ( Small )
- m ( Medium )
- l ( Large )
- x ( eXtra large )



- **더 작은 모델로도**  
(적은 수의 매개변수)  
**더 높은 정확도를 보인다.**
- **다양한 환경에서 실행 가능**  
(엣지 디바이스, 클라우드 ..)
- **다양한 종류의 작업 지원**  
(분류, 이미지분할,  
포즈 예측 ...)



# 우리가 원하는 객체를 탐지하고 싶을 때



사전 학습 모델  
(Pre-training model)

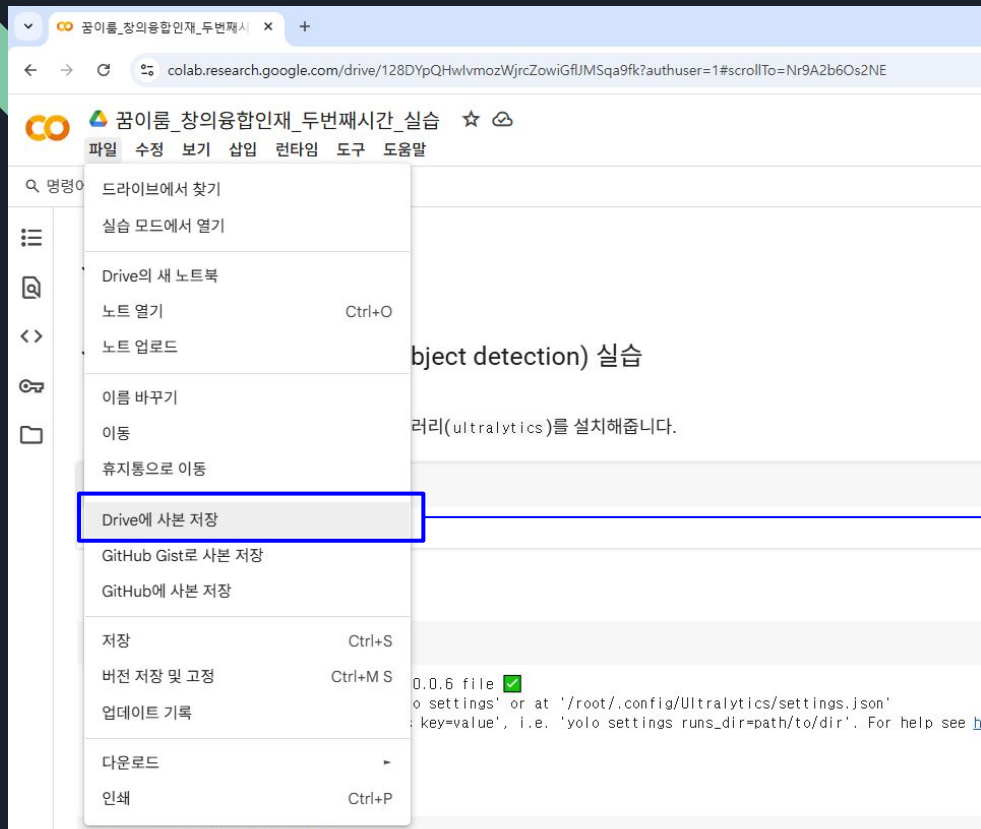
미세 조정(fine-tuning)을  
통한 전이 학습  
(transfer learning)

직접 수집하고  
라벨링한  
커스텀 데이터셋

Small dataset



## 실습 파일 링크 : 꿈이름 창의 융합 인재 두번째 실습



구글에서 제공하는 **코랩**

- 주피터 노트북 스타일의 실행 환경
- 설치 과정 없이 구글드라이브에서 무료로 사용할 수 있다.
- 인공지능을 위한 가속 장치인 GPU, TPU 등을 사용할 수 있다.

파일 > Drive에 사본 저장  
을 통해서 위의 실습 파일을  
자신의 구글 드라이브에 저장하여야  
이후에 자신의 실습 내용을  
다시 확인할 수 있다.



## ✓ YOLO를 사용한 객체 탐지(object detection) 실습

Yolo를 사용할 수 있게 해주는 **파이썬 라이브러리(ultralytics)**를 설치해줍니다.



```
!pip install ultralytics
```

숨겨진 출력 표시

파이썬에서는 누군가 미리 만들어 놓은  
라이브러리 또는 패키지를 설치하여  
손쉽게 원하는 작업을 할 수 있다.

코랩에서 하나의  
명령어 셀을 실행할 때  
그림의 실행버튼을  
누르거나  
shift + 엔터키를 눌러  
진행할 수 있다.

코랩(colab)에서 파이썬 패키지 관리자인 pip 를 사용할 때에는  
느낌표 (!)를 붙인 설치 명령어를 써야 한다.

ultralytics  
YOLO11  
Unleashing Next-Gen AI

Discover more

- Enhanced feature extraction architecture
- Faster processing with balanced accuracy
- Higher precision using 22% fewer parameters
- Easily deployable on edge, cloud, and GPU systems
- YOLO11 enterprise models providing enterprise-grade accuracy

Download the App

다양한 YOLO 모델을 제공하는  
ultralytics 라이브러리 ([링크](#))

Download on the App Store GET IT ON Google Play

설치한 라이브러리를 사용을 위해 불러온다.

설치한 라이브러리를 불러옵니다.

✓  
4초



```
1 import ultralytics
```

객체 탐지에 사용할 **인공지능 모델을 선언한다.**

설치한 라이브러리에서 객체 탐지를 위한 모델을 불러옵니다.

✓  
1초



```
1 from ultralytics import YOLO  
2 model = YOLO("yolo11n.pt")
```



Downloading <https://github.com/ultralytics/assets/releases>  
100% ██████████ 5.35M/5.35M [00:00<00:00, 175MB/s]

**모델의 정확도와 모델의 크기에 따른 속도를 고려하여**

**스스로 모델을 선택할 수 있다.**

yolo11n.pt, yolo11s.pt, yolo11m.pt, yolo11l.pt, yolo11x.pt

모델	파일 이름
YOLO11	yolo11n.pt yolo11s.pt yolo11m.pt yolo11l.pt yolo11x.pt
YOLO11-seg	yolo11n-seg.pt yolo11s-seg.pt yolo11m-seg.pt yolo11l-seg.pt yolo11x-seg.pt
YOLO11-pose	yolo11n-pose.pt yolo11s-pose.pt yolo11m-pose.pt yolo11l-pose.pt yolo11x-pose.pt
YOLO11-obb	yolo11n-obb.pt yolo11s-obb.pt yolo11m-obb.pt yolo11l-obb.pt yolo11x-obb.pt
YOLO11-cls	yolo11n-cls.pt yolo11s-cls.pt yolo11m-cls.pt yolo11l-cls.pt yolo11x-cls.pt

객체 탐지할 이미지를 인터넷에서 찾아옵니다.

**원에서 복사한 주소를 붙여 넣는다.**

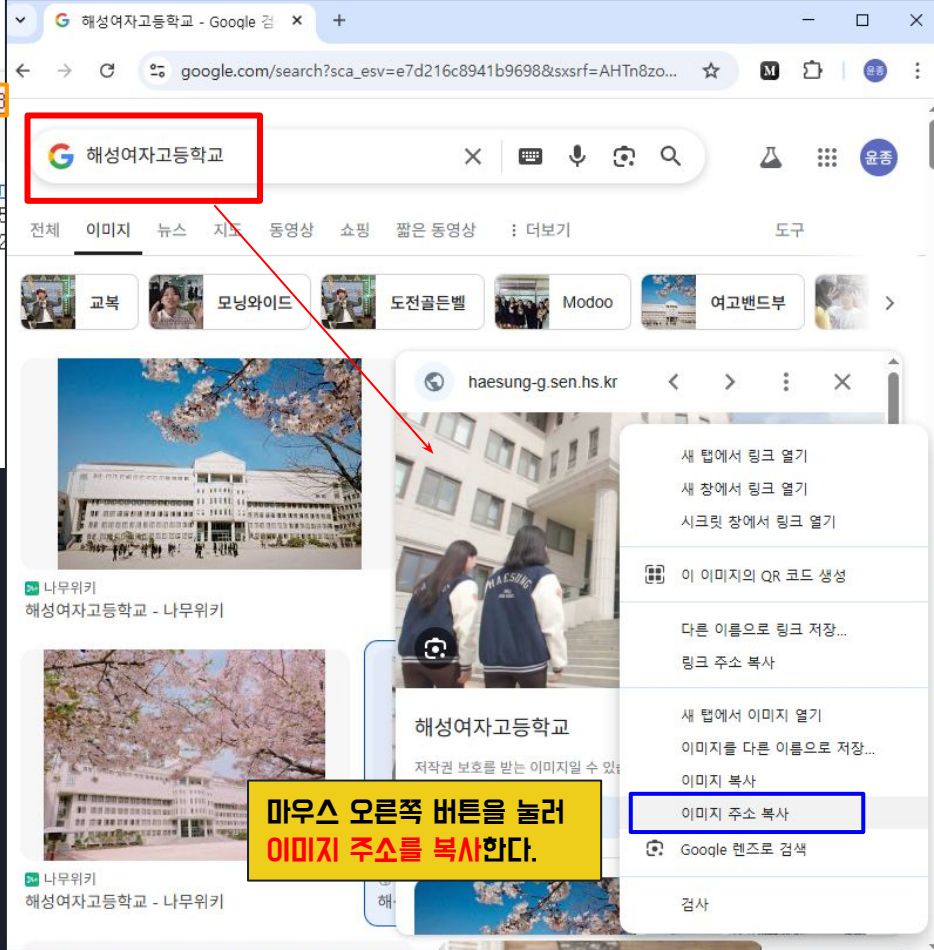
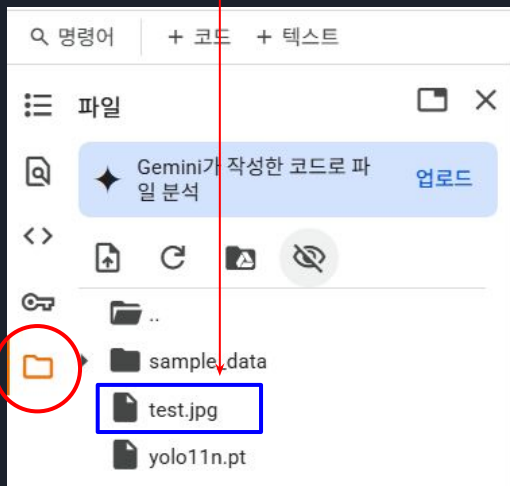
```
1 img_url = https://hostncd.ssem.or.kr/api/img.php?syncno=58253
2 !wget {img_url} -O test.jpg
```

**검색한 이미지를 test.jpg라는 이름으로 저장한다.**

```
--2025-05-15 09:48:37-- https://hostncd.ssem.or.kr/api/img.php?syncno=58253
Resolving hostncd.ssem.or.kr (hostncd.ssem.or.kr)... 103.157.158.25
Connecting to hostncd.ssem.or.kr (hostncd.ssem.or.kr)|103.157.158.25:443...
HTTP request sent, awaiting response... 200 OK
Length: 29010 (28K) [image/jpeg]
Saving to: 'test.jpg'

test.jpg      100%[=====>]  28.33K   116KB/s

2025-05-15 09:48:39 (116 KB/s) - 'test.jpg' saved [29010/29010]
```



단 한 줄의 명령어로 손쉽게 객체 탐지 작업을 수행할 수 있다.

객체 탐지 모델을 사용하여, 추론(inference)작업을 수행합니다.

```
1 results = model("test.jpg")
```

0초 7초

image 1/1 /content/test.jpg: 384x640 11 persons, 371.8ms  
Speed: 6.9ms preprocess, 371.8ms inference, 30.3ms postprocess per image at shape (1, 3, 384, 640)

소요시간

```
1 results[0].save("result.jpg")
```

0초

'result.jpg'

추론 결과를 저장한다.

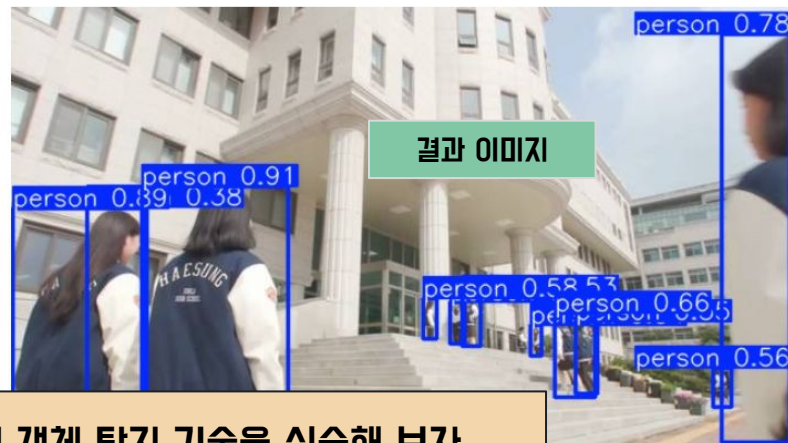
```
1 from IPython.display import Image  
2 Image("test.jpg")
```

0초



```
1 Image("result.jpg")
```

0초



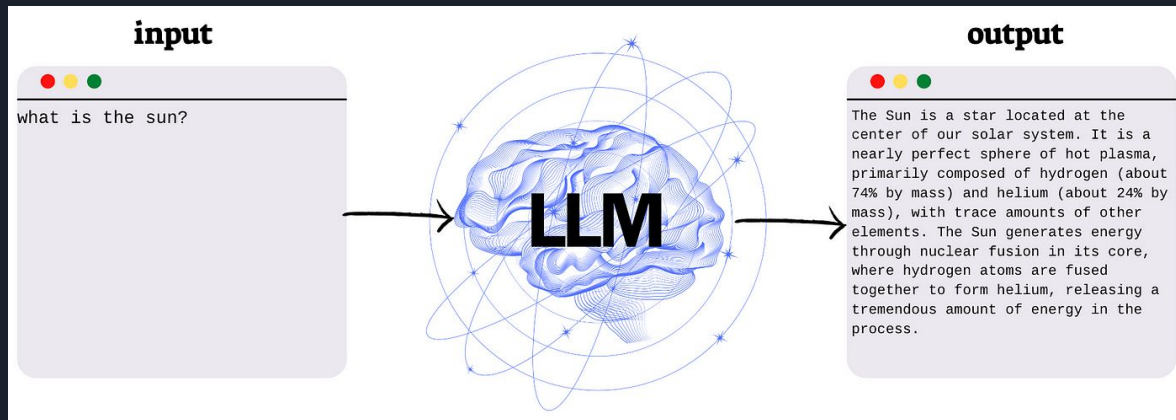
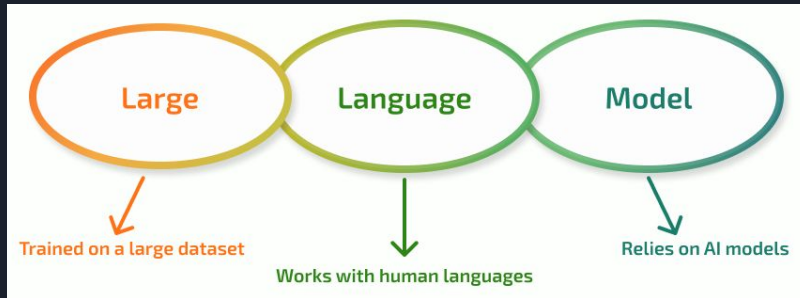
직접 다양한 사진, 이미지를 사용하면서 객체 탐지 기술을 실습해 보자.



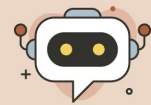
## 실습 2.

# Gemini api를 이용한 질문-응답 서비스

# 대규모 언어모델(LLM)



## 6 Large Language Model Use Cases



Chatbots



Text generation



Language translation



Sentiment analysis



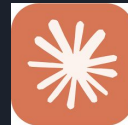
Question answering



Code generation

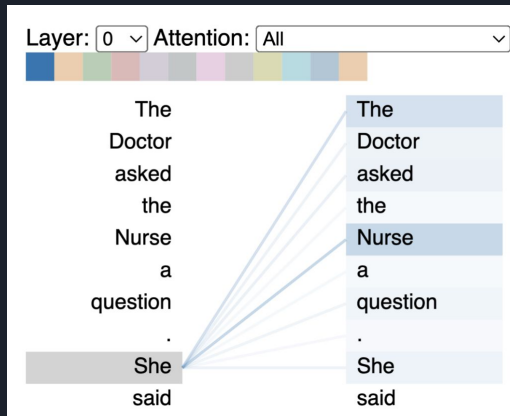
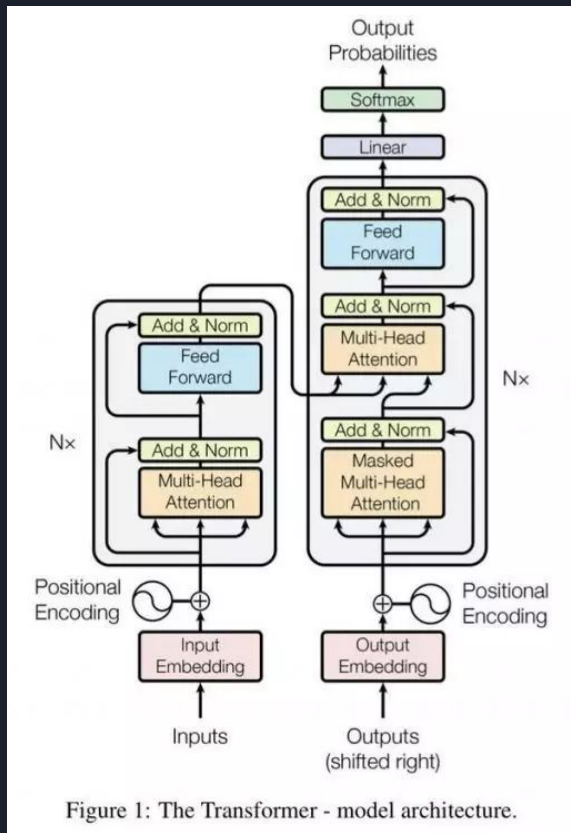
smith.ai

- 사람이 사용하는 자연어를 학습해서, 사람처럼 대화
- 처음에는 텍스트 생성을 목표로 하였지만, 번역, 요약, 코드 생성 등 다양한 작업을 할 수 있다.





# LLM의 동작 원리

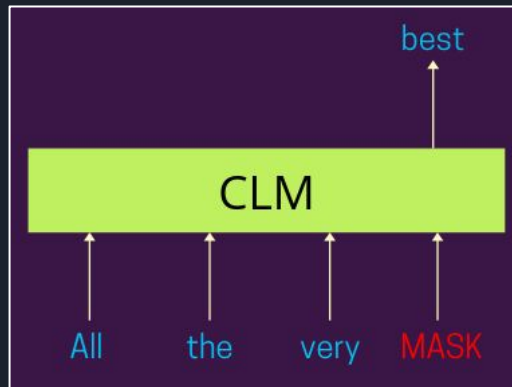


## 트랜스포머(Transformer) 모델

: 입력된 문장의 모든 단어를 동시에 처리하면서, 각 단어가 서로 가지는 관계를 참고하여 계산한다.

## 셀프 어텐션(self-attention)

: 단어 간의 중요도를 계산할 때, 하나의 단어가 (자신이 속한) 문장 안의 다른 단어들을 얼마나 참고하는 지 계산한다.

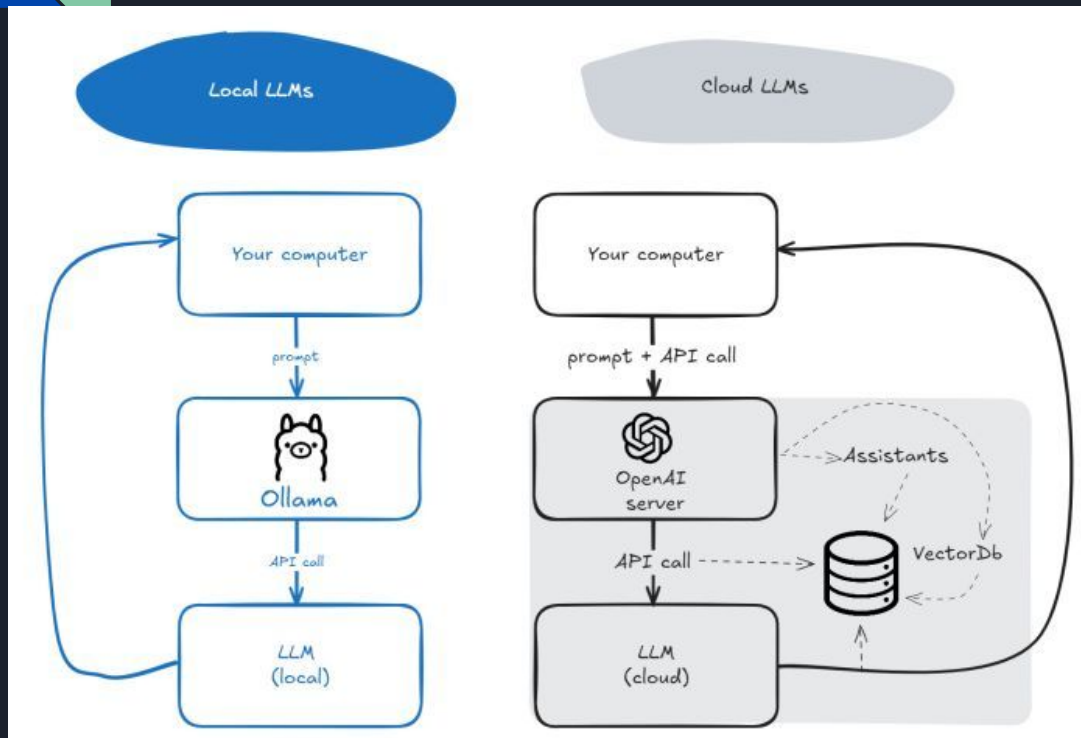


## Causal Language Model

: 이전에 나온 단어들을 기반으로 새로 나올 단어를 예측하는 언어모델



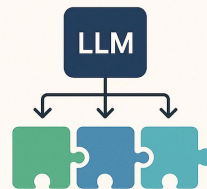
# LLM을 우리가 사용하는 방법



## OPEN-WEIGHT MODELS vs API-ONLY LLMs

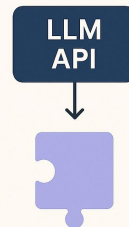
Ecosystem Fragmentation Explained

OPEN-WEIGHT  
MODELS



VS

API-ONLY LLMs



## Free Open Source LLMs



# LLM API를 활용한 질문-응답 챗봇

API 키만 발급받는다면, 어떠한 LLM api도 손쉽게 사용 가능



구글의 AI 언어모델 : Gemini

- 구글의 최신 LLM 기술로 만든 텍스트 생성 모델
- 질문-응답, 요약, 번역, 글쓰기 등 창의적이고 다양한 작업 처리 가능
- 웹 브라우저나 자신의 애플리케이션과 연동하여 쉽게 사용할 수 있다.



실습 파일 링크 : [꿈이를 창의 융합 인재 두번째 실습](#)

동일한 실습 파일의 두번째 부분을 사용합니다.

사용할 라이브러리를 설치한다.

구글의 LLM 모델을 사용할 수 있는 라이브러리를 설치해 줍니다.

✓  
2초



```
1 !pip install -q -U google-generativeai
```

설치한 라이브러리를 사용을 위해 불러온다.

설치한 라이브러리를 불러옵니다.

✓  
1초

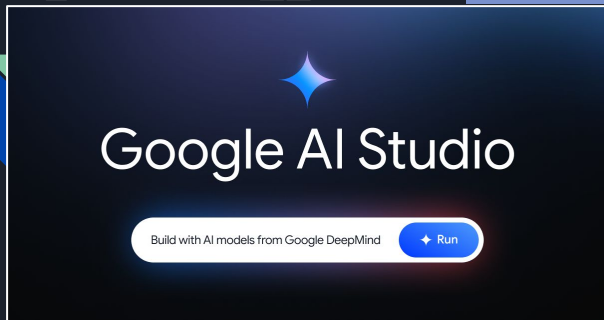


```
1 import google.generativeai as genai
```

이번에도 실습 1 과 동일한 방식으로  
구글 LLM API를 사용할 수 있는  
파이썬 라이브러리를 설치하고  
불러온다.

단, 두 가지 방법을 비교하기 위하여  
예전 라이브러리를 우선 사용해본다.  
뒤의 실습과 설명 이후에는  
최신 라이브러리를 활용하는 편이 좋다

구글 LLM API를 무료로 제공해주는 [구글 AI Studio](#)에 접속



API 키

+ API 키 만들기

Gemini API를 빠르게 테스트하세요

API 빠른 시작 가이드

```
curl "https://generativelanguage.googleapis.com/v1beta/models/gemini-2.0-flash:generateContent?"  
key=GEMINI_API_KEY" #  
-H 'Content-Type: application/json' #
```

API 키 생성됨

API 키는 안전하게 사용합니다. 다른 사람이 볼 수 있는 코드에 키를 공유하거나 삽입하지 마세요.

AlzaSyDBjbnlftKRFZ7uBc0Tm6m0Jrx8T

복사

생성된 API키를 눌러서 확인 후 복사해준다

프로젝트 번호

Google Cloud 프로젝트 이름

API 키

생성일

요금제

...1428	Gemini API	...gWP8	2025. 5. 13.	Free Set up Billing View usage data
---------	------------	---------	--------------	---

API 키는 **다른 사람이 모르게 보안 비밀로 저장**해야 한다.  
Gemini 라는 이름의 보안 비밀키로 저장해 코랩에서 사용한다.

Google AI Studio에서 발급받은 API키를 왼쪽메뉴 중 **보안**에 저장하고, 이를 불러옵니다.

```
1 from google.colab import userdata
2 GOOGLE_API_KEY=userdata.get('Gemini')
3
4 genai.configure(api_key=GOOGLE_API_KEY)
```

불러온 라이브러리에  
자신의 API 키를 입력하는 과정

## 보안 비밀

환경 변수, 파일 경로 또는 키를 저장하여 코드를 구성합니다.  
된 값은 비공개되며, 나와 내가 선택한 노트북에만 공개됩니다.

<> 보안 비밀 이름에는 공백을 포함할 수 없습니다.



노트북  
액세스

이름

값

작업



Gemini



+ 새 보안 비밀 추가

Gemini API 키 ▾

다음은 통해 Python에서 보안 비밀 키에 액세스합니다.

```
from google.colab import userdata
userdata.get('secretName')
```

참고) 아래는 구글에서 제공하여서 사용할 수 있는 인공지능 모델의 종류입니다.

```
2초 ▶ 1 for m in genai.list_models():
      2     if 'generateContent' in m.supported_generation_methods:
      3         print(m.name)
```

```
🔍 models/gemini-1.0-pro-vision-latest
models/gemini-pro-vision
models/gemini-1.5-pro-latest
models/gemini-1.5-pro-001
models/gemini-1.5-pro-002
models/gemini-1.5-pro
models/gemini-1.5-flash-latest
models/gemini-1.5-flash-001
models/gemini-1.5-flash-001-tuning
models/gemini-1.5-flash
models/gemini-1.5-flash-002
models/gemini-1.5-flash-8b
models/gemini-1.5-flash-8b-001
models/gemini-1.5-flash-8b-latest
models/gemini-1.5-flash-8b-exp-0827
models/gemini-1.5-flash-8b-exp-0924
```

**구글 AI Studio에서 제공하는 모든 모델을 무료로 사용할 수는 없지만  
최신의 고성능 모델들도 무료로 제공되고 있습니다.**

```
models/gemini-2.5-pro-exp-03-25
models/gemini-2.5-pro-preview-03-25
models/gemini-2.5-flash-preview-04-17
models/gemini-2.5-flash-preview-04-17-thinking
models/gemini-2.5-pro-preview-05-06
models/gemini-2.0-flash-exp
```

**models/gemini-2.0-flash 이번 실습에 사용할 모델**

```
models/gemini-2.0-flash-001
models/gemini-2.0-flash-exp-image-generation
models/gemini-2.0-flash-lite-001
models/gemini-2.0-flash-lite
models/gemini-2.0-flash-preview-image-generation
models/gemini-2.0-flash-lite-preview-02-05
models/gemini-2.0-flash-lite-preview
models/gemini-2.0-pro-exp
models/gemini-2.0-pro-exp-02-05
models/gemini-exp-1206
models/gemini-2.0-flash-thinking-exp-01-21
models/gemini-2.0-flash-thinking-exp
models/gemini-2.0-flash-thinking-exp-1219
models/learnlm-2.0-flash-experimental
models/gemma-3-1b-it
models/gemma-3-4b-it
models/gemma-3-12b-it
models/gemma-3-27b-it
```

✓ [15] 1 model = genai.GenerativeModel('gemini-2.0-flash')

원하는 질문을 입력하여 인공지능 모델의 대답을 얻어봅시다

✓ [16] 1 response = model.generate\_content("대한민국의 제 2의 도시를 알려줘")

✓ 0초 ▶ 1 print(response.text)

↻ 대한민국의 제2의 도시는 \*\*부산광역시\*\*입니다.

인공지능 모델에게  
다양한 질문을 던져보고  
응답의 정확성을  
평가해보자.

하지만 지금의 질문-응답은 구글 인공지능 모델이 답변하는 방식을 바꾸지는 못한다.  
우리는 필요에 따라서 인공지능 모델 LLM이 답변하는 방식이나 스타일을 바꿀 수 있다.



이제 특별한 응답 형식을 만들 수 있는 **시스템 프롬프트**를 사용하여, 동일한 질문을 해봅시다.

✓  
0초

```
[18] 1 system_prompt = "당신은 6살 아이입니다. 아이의 말투로 대답해주세요"  
2 question = "대한민국의 제 2의 도시를 알려줘."
```

**시스템 프롬프트를 활용하면  
LLM이 답변하는 방식을 바꿀 수 있다.**

✓  
1초

```
[19] 1 from google import genai  
2 from google.genai import types  
3  
4 client = genai.Client(api_key=G00GLE_API_KEY)  
5
```

**최신 라이브러리를  
불러와 사용한다.**

**시스템 프롬프트를 통해  
인공지능 모델에게 **특정 역할을 부여**하고  
**따라야 할 규칙**을 정해줄 수 있다.**

✓  
0초

```
[20] 1 response = client.models.generate_content(  
2     model="gemini-2.0-flash",  
3     config=types.GenerateContentConfig(  
4         system_instruction=system_prompt),  
5     contents=question  
6 )
```

**시스템 프롬프트가 입력되는 곳**  
**사용자 질문이 입력되는 곳**

✓  
0초

```
1 print(response.text)
```

음... 부산! 바다도 있고, 맛있는 것도 엄청 많아! 해운대도 있고, 신기한 아쿠아리움도 있어! 짱 재밌어! 😊

**LLM에게 다양한 역할과 규칙을 부여하고, 답변의 변화를 살펴보자.**



**Q & A**

**감사합니다**