딥러닝을 활용한 물체 분류 및 인식

Deep Learning

- **딥러닝 (Deep Learning)**
  - ✓ 기계 학습 (Machine Learning)의 한 분야로, 인공 신경망 (Artificial Neural Network)을 사용해 데이터를 분석하고 예측하는 기술
  - ✓ **다층 신경망**(Deep Neural Network)을 통해 복잡한 데이터의 패턴을 학습
  - ✓ **특징 추출**: 딥러닝은 수동으로 특징을 추출하지 않고, 데이터로부터 스스로 특징을 학습하여 효율적으로 패턴을 인식

- **딥러닝의 주요 구성 요소**
  - ✓ **뉴런**: 입력을 받아 계산하여 출력을 생성하는 단위
  - ✓ **레이어**: 여러 개의 뉴런이 모여 구성된 구조 (입력층, 은닉층, 출력층)
  - ✓ **학습 과정**: 입력 데이터를 통해 가중치를 조정하면서 최적의 모델을 생성

- **딥러닝의 발전**
  - ✓ **대규모 데이터**와 **고성능 하드웨어**의 발전에 의해 최근 몇 년 간 급격히 발전
  - ✓ 이미지, 음성, 자연어 처리(NLP) 등 다양한 분야에서 활용

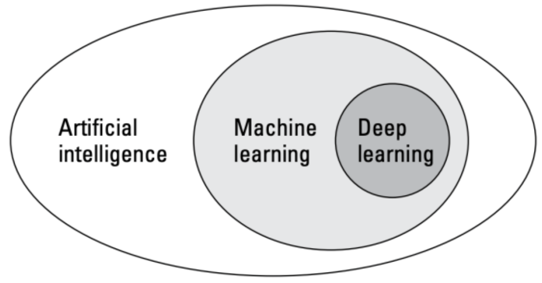
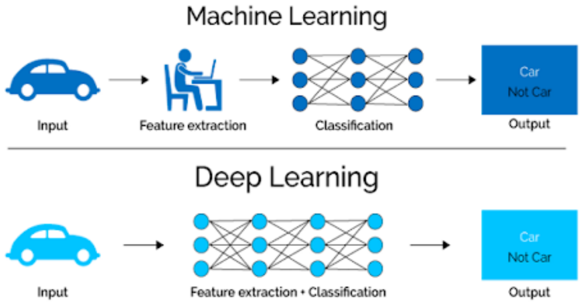
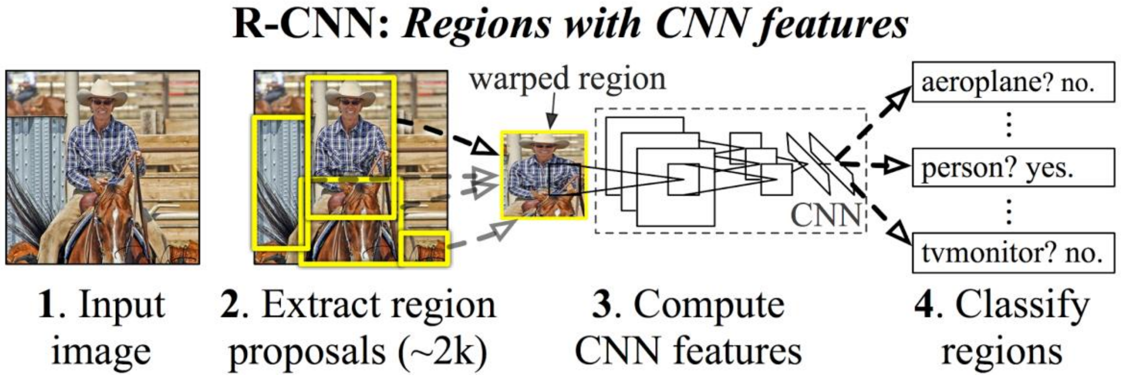


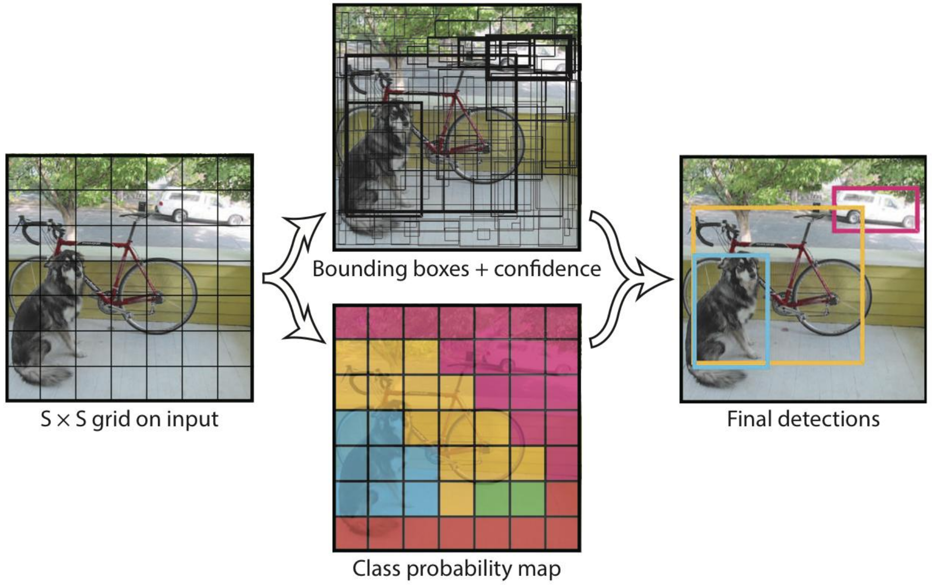
FIGURE 1-1: Deep learning, a subset of a subset of AI.



YOLO(You Only Look Once)

- 실시간 객체 인식 알고리즘
- 입력 이미지를 한 번만 보고 객체를 인식하는 특징
- 기존의 방법들(예: R-CNN)은 이미지 여러 부분을 여러 번 처리하는 방식에 비해 빠르고 효율적
- 주요 특징: 빠른 처리 속도, 높은 정확도, 실시간 처리 가능

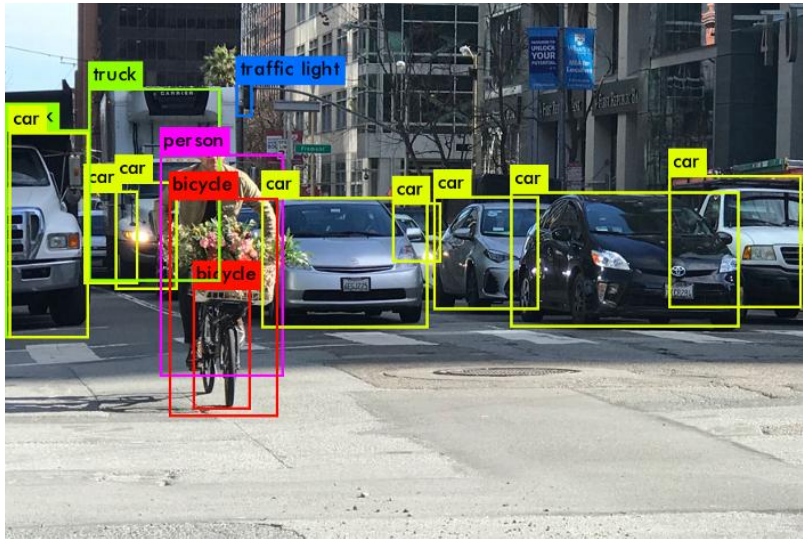


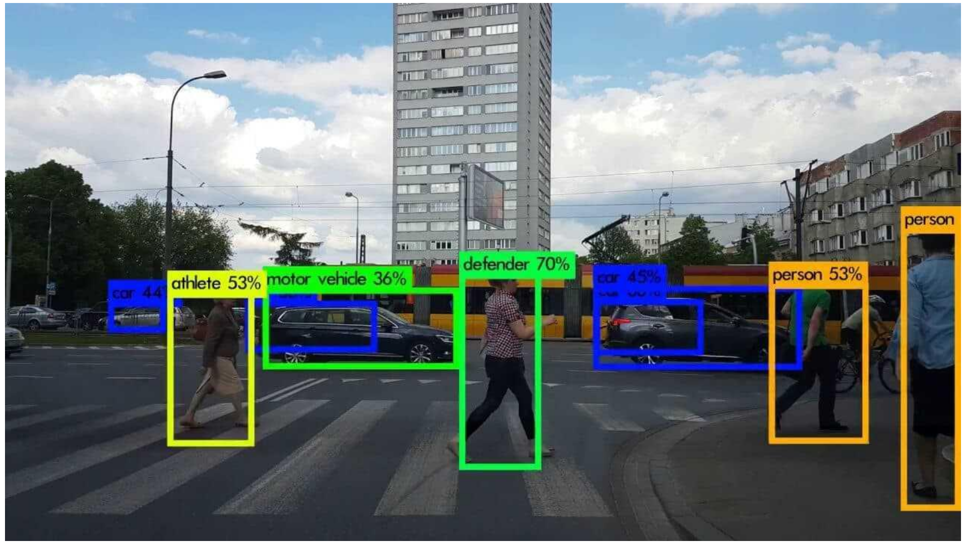


YOLO의 적용 사례 및 장점

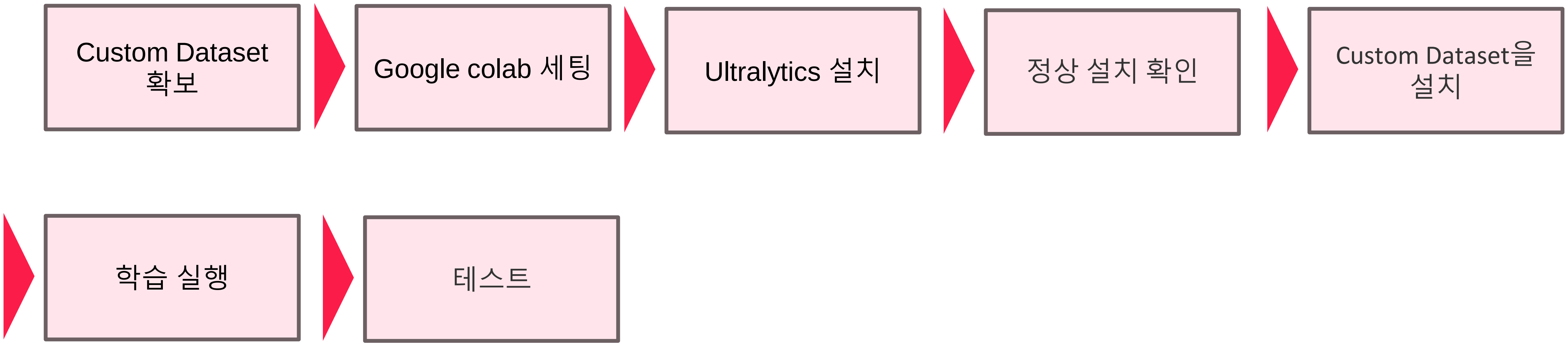
- **적용 사례**
  ✓자율주행차: 도로의 객체를 실시간으로 인식
  ✓보안 감시: CCTV 영상에서 사람, 차량, 기타 객체 식별
  ✓의료 영상 분석: CT, MRI 등에서 이상 객체 인식
  ✓로봇 비전: 로봇의 환경 인식 및 물체 인식

- **장점**
  ✓ 속도: 실시간 객체 인식이 가능
  ✓ 정확도: 높은 정확도를 유지하면서도 빠른 처리 속도 제공
  ✓ 단일 네트워크: 복잡한 네트워크 구조 없이 한 번의 예측으로 결과 도출





YOLO 사용해보기



Custom Dataset 확보

Google colab 세팅

Ultralytics 설치

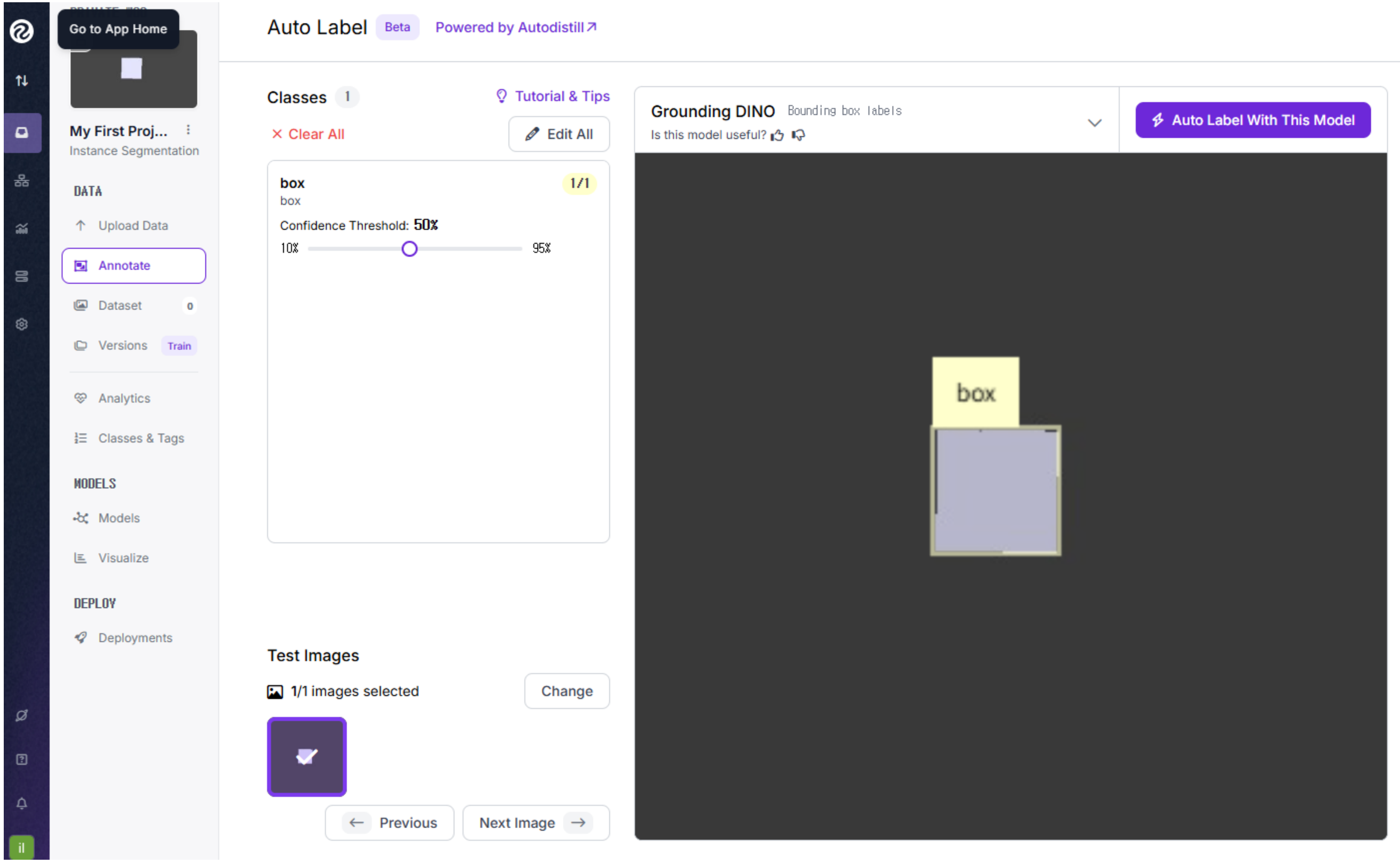
정상 설치 확인

Custom Dataset을 설치

학습 실행

테스트

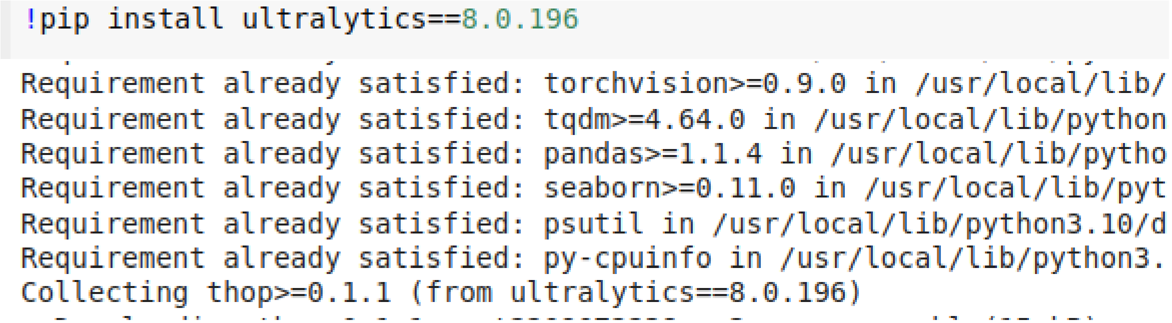
Custom Dataset 확보



Confidence Threshold: 50%

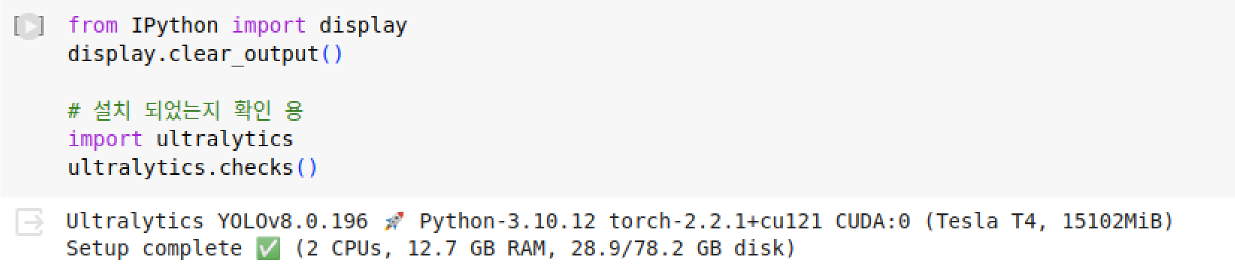
1/1 images selected

Google colab 세팅



```
!pip install ultralytics==8.0.196
```

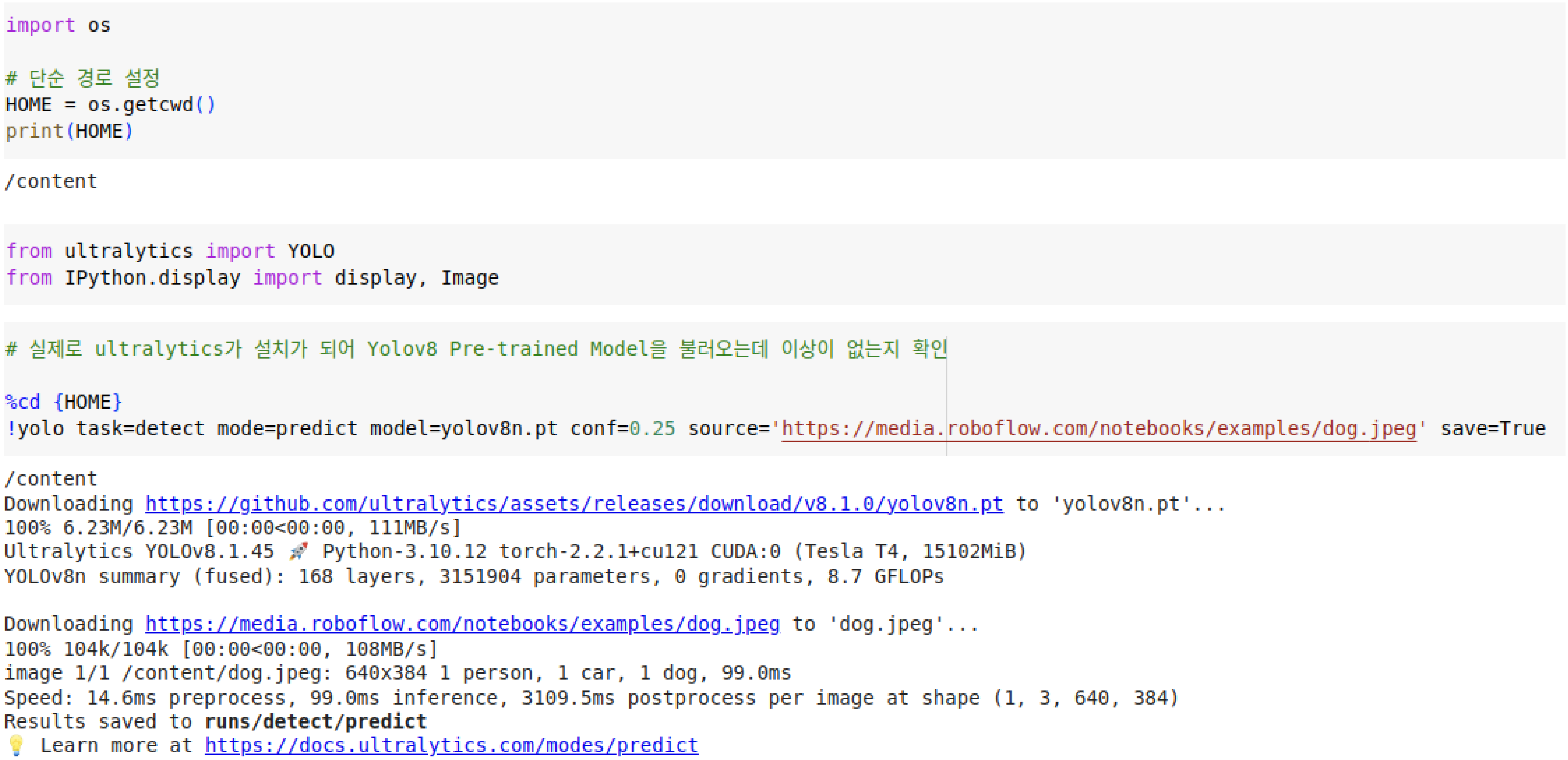
```
Requirement already satisfied: torchvision>=0.9.0 in /usr/local/lib/
Requirement already satisfied: tqdm>=4.64.0 in /usr/local/lib/python
Requirement already satisfied: pandas>=1.1.4 in /usr/local/lib/pytho
Requirement already satisfied: seaborn>=0.11.0 in /usr/local/lib/pyt
Requirement already satisfied: psutil in /usr/local/lib/python3.10/d
Requirement already satisfied: py-cpuinfo in /usr/local/lib/python3.
Collecting thop>=0.1.1 (from ultralytics==8.0.196)
```



```
from IPython import display
display.clear_output()

# 설치 되었는지 확인 용
import ultralytics
ultralytics.checks()
```

```
Ultralytics YOLOv8.0.196 🚀 Python-3.10.12 torch-2.2.1+cu121 CUDA:0 (Tesla T4, 15102MiB)
Setup complete ✅ (2 CPUs, 12.7 GB RAM, 28.9/78.2 GB disk)
```

정상 설치 확인



```
import os

# 단순 경로 설정
HOME = os.getcwd()
print(HOME)
```

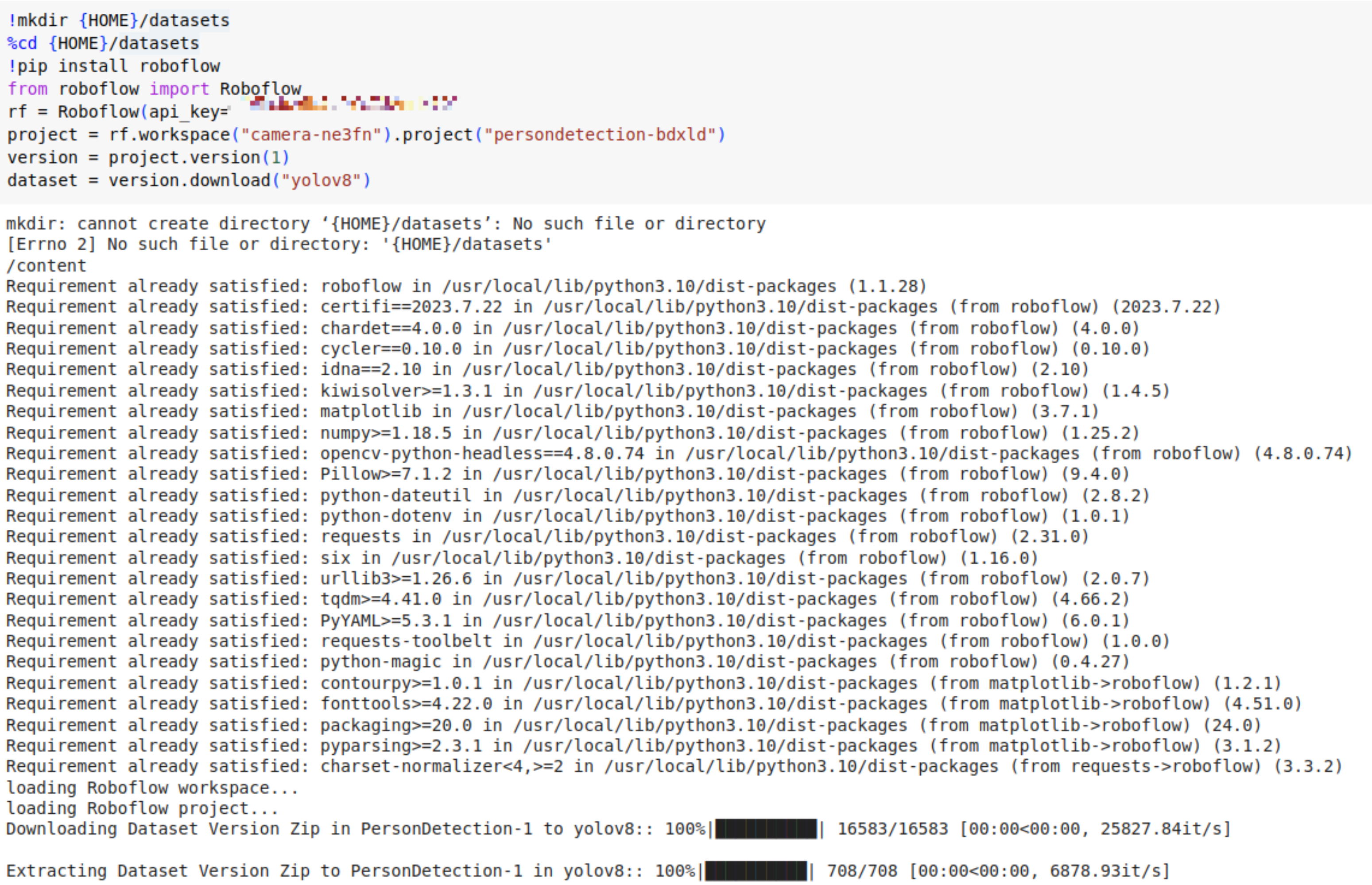
```
/content
```

```
from ultralytics import YOLO
from IPython.display import display, Image
```

```
# 실제로 ultralytics가 설치가 되어 Yolov8 Pre-trained Model을 불러오는데 이상이 없는지 확인

%cd {HOME}
!yolo task=detect mode=predict model=yolov8n.pt conf=0.25 source='https://media.roboflow.com/notebooks/examples/dog.jpeg' save=True
```

```
/content
Downloading https://github.com/ultralytics/assets/releases/download/v8.1.0/yolov8n.pt to 'yolov8n.pt'...
100% 6.23M/6.23M [00:00<00:00, 111MB/s]
Ultralytics YOLOv8.1.45 🚀 Python-3.10.12 torch-2.2.1+cu121 CUDA:0 (Tesla T4, 15102MiB)
YOLOv8n summary (fused): 168 layers, 3151904 parameters, 0 gradients, 8.7 GFLOPs

Downloading https://media.roboflow.com/notebooks/examples/dog.jpeg to 'dog.jpeg'...
100% 104k/104k [00:00<00:00, 108MB/s]
image 1/1 /content/dog.jpeg: 640x384 1 person, 1 car, 1 dog, 99.0ms
Speed: 14.6ms preprocess, 99.0ms inference, 3109.5ms postprocess per image at shape (1, 3, 640, 384)
Results saved to runs/detect/predict
💡 Learn more at https://docs.ultralytics.com/modes/predict
```

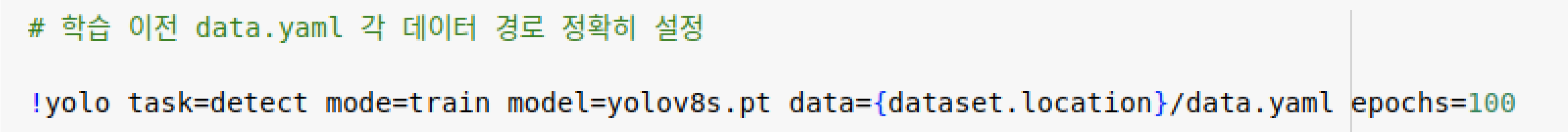
Custom Dataset을 설치



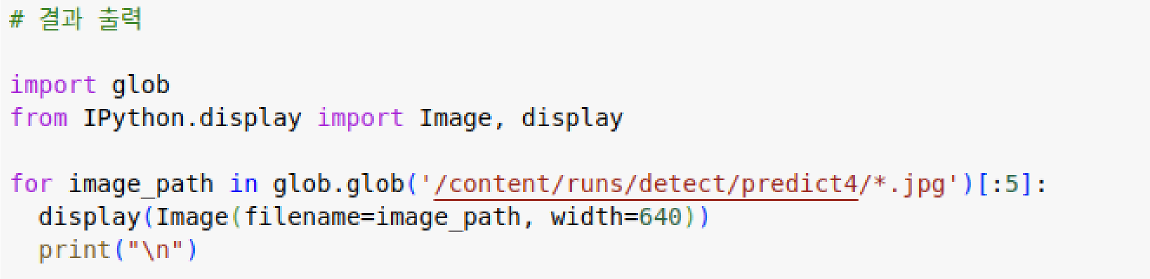
```
!mkdir {HOME}/datasets
%cd {HOME}/datasets
!pip install roboflow
from roboflow import Roboflow
rf = Roboflow(api_key=[redacted]
project = rf.workspace("camera-ne3fn").project("persondetection-bdxld")
version = project.version(1)
dataset = version.download("yolov8")
```

```
mkdir: cannot create directory '{HOME}/datasets': No such file or directory
[Errno 2] No such file or directory: '{HOME}/datasets'
/content
Requirement already satisfied: roboflow in /usr/local/lib/python3.10/dist-packages (1.1.28)
Requirement already satisfied: certifi==2023.7.22 in /usr/local/lib/python3.10/dist-packages (from roboflow) (2023.7.22)
Requirement already satisfied: chardet==4.0.0 in /usr/local/lib/python3.10/dist-packages (from roboflow) (4.0.0)
Requirement already satisfied: cycler==0.10.0 in /usr/local/lib/python3.10/dist-packages (from roboflow) (0.10.0)
Requirement already satisfied: idna==2.10 in /usr/local/lib/python3.10/dist-packages (from roboflow) (2.10)
Requirement already satisfied: kiwisolver>=1.3.1 in /usr/local/lib/python3.10/dist-packages (from roboflow) (1.4.5)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.10/dist-packages (from roboflow) (3.7.1)
Requirement already satisfied: numpy>=1.18.5 in /usr/local/lib/python3.10/dist-packages (from roboflow) (1.25.2)
Requirement already satisfied: opencv-python-headless==4.8.0.74 in /usr/local/lib/python3.10/dist-packages (from roboflow) (4.8.0.74)
Requirement already satisfied: Pillow>=7.1.2 in /usr/local/lib/python3.10/dist-packages (from roboflow) (9.4.0)
Requirement already satisfied: python-dateutil in /usr/local/lib/python3.10/dist-packages (from roboflow) (2.8.2)
Requirement already satisfied: python-dotenv in /usr/local/lib/python3.10/dist-packages (from roboflow) (1.0.1)
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages (from roboflow) (2.31.0)
Requirement already satisfied: six in /usr/local/lib/python3.10/dist-packages (from roboflow) (1.16.0)
Requirement already satisfied: urllib3>=1.26.6 in /usr/local/lib/python3.10/dist-packages (from roboflow) (2.0.7)
Requirement already satisfied: tqdm>=4.41.0 in /usr/local/lib/python3.10/dist-packages (from roboflow) (4.66.2)
Requirement already satisfied: PyYAML>=5.3.1 in /usr/local/lib/python3.10/dist-packages (from roboflow) (6.0.1)
Requirement already satisfied: requests-toolbelt in /usr/local/lib/python3.10/dist-packages (from roboflow) (1.0.0)
Requirement already satisfied: python-magic in /usr/local/lib/python3.10/dist-packages (from roboflow) (0.4.27)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib->roboflow) (1.2.1)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib->roboflow) (4.51.0)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib->roboflow) (24.0)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib->roboflow) (3.1.2)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests->roboflow) (3.3.2)
loading Roboflow workspace...
loading Roboflow project...
Downloading Dataset Version Zip in PersonDetection-1 to yolov8:: 100%|██████████| 16583/16583 [00:00<00:00, 25827.84it/s]

Extracting Dataset Version Zip to PersonDetection-1 in yolov8:: 100%|██████████| 708/708 [00:00<00:00, 6878.93it/s]
```

Custom Dataset을 설치



```
# 학습 이전 data.yaml 각 데이터 경로 정확히 설정

!yolo task=detect mode=train model=yolov8s.pt data={dataset.location}/data.yaml epochs=100
```



```
# 결과 출력

import glob
from IPython.display import Image, display

for image_path in glob.glob('/content/runs/detect/predict4/*.jpg')[:5]:
  display(Image(filename=image_path, width=640))
  print("\n")
```