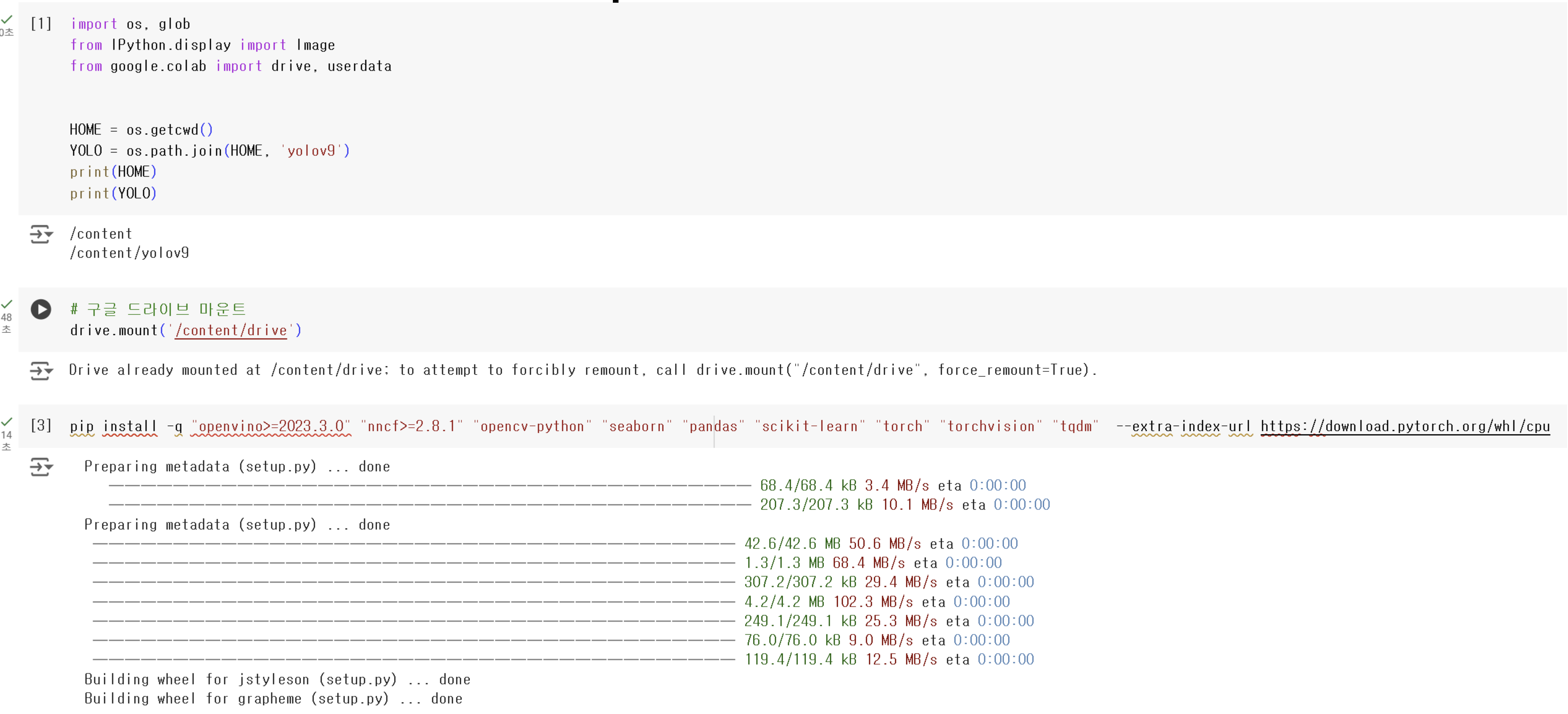
YOLOv9 Inferencing

하성욱 교수

YOLOv9모델 OpenVINO 모델로 변환



```
import os, glob
from IPython.display import Image
from google.colab import drive, userdata


HOME = os.getcwd()
YOLO = os.path.join(HOME, 'yolov9')
print(HOME)
print(YOLO)
```

```
/content
/content/yolov9
```

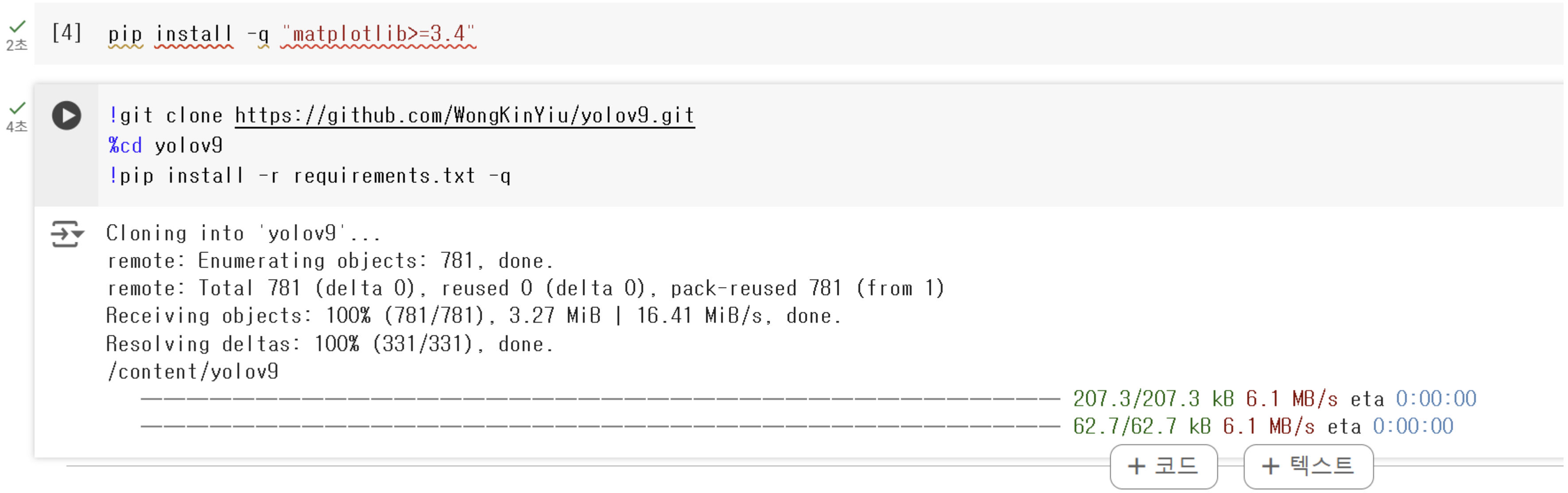
```
# 구글 드라이브 마운트
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
```

```
pip install -q "openvino>=2023.3.0" "nncf>=2.8.1" "opencv-python" "seaborn" "pandas" "scikit-learn" "torch" "torchvision" "tqdm"  --extra-index-url https://download.pytorch.org/whl/cpu
```

```
  Preparing metadata (setup.py) ... done
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 68.4/68.4 kB 3.4 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 207.3/207.3 kB 10.1 MB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 42.6/42.6 MB 50.6 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.3/1.3 MB 68.4 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 307.2/307.2 kB 29.4 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 4.2/4.2 MB 102.3 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 249.1/249.1 kB 25.3 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 76.0/76.0 kB 9.0 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 119.4/119.4 kB 12.5 MB/s eta 0:00:00
  Building wheel for jstyleson (setup.py) ... done
  Building wheel for grapheme (setup.py) ... done
```

YOLOv9모델 OpenVINO 모델로 변환

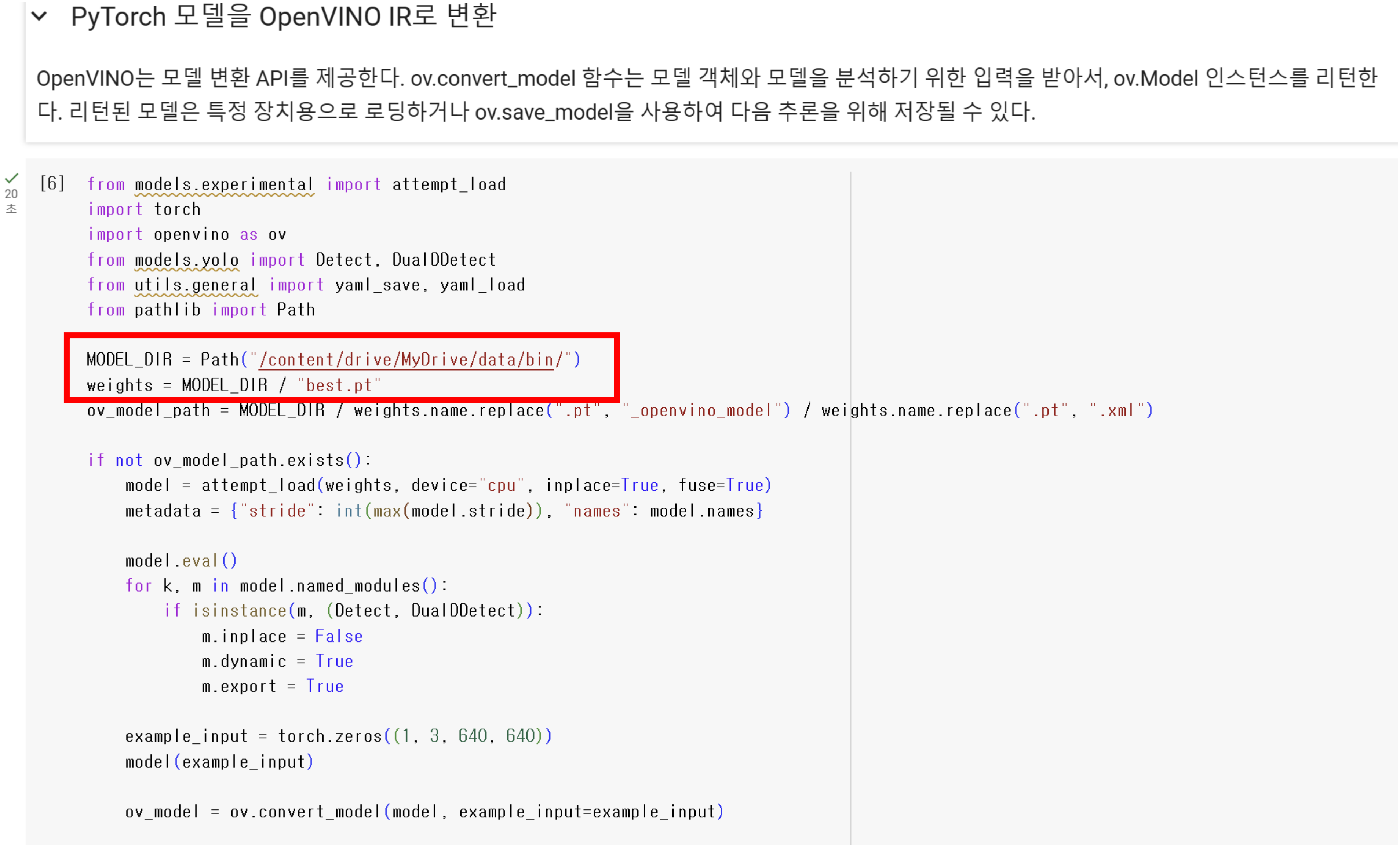


```
pip install -q "matplotlib>=3.4"
```

```
!git clone https://github.com/WongKinYiu/yolov9.git
%cd yolov9
!pip install -r requirements.txt -q
```

```
Cloning into 'yolov9'...
remote: Enumerating objects: 781, done.
remote: Total 781 (delta 0), reused 0 (delta 0), pack-reused 781 (from 1)
Receiving objects: 100% (781/781), 3.27 MiB | 16.41 MiB/s, done.
Resolving deltas: 100% (331/331), done.
/content/yolov9
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 207.3/207.3 kB 6.1 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 62.7/62.7 kB 6.1 MB/s eta 0:00:00
```

YOLOv9모델 OpenVINO 모델로 변환

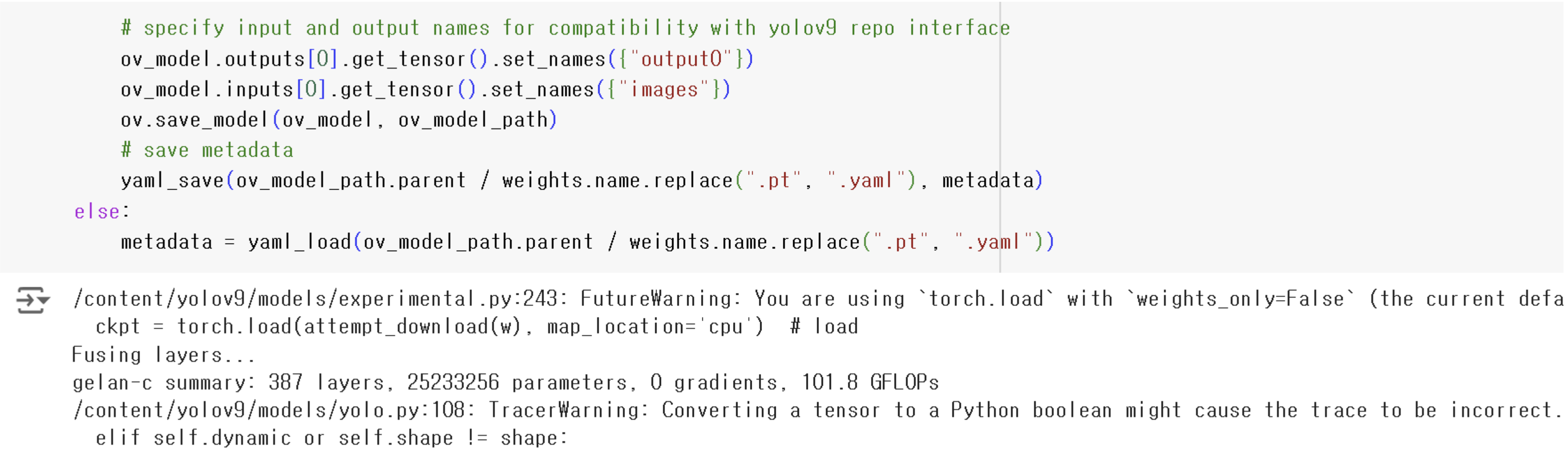


PyTorch 모델을 OpenVINO IR로 변환

OpenVINO는 모델 변환 API를 제공한다. ov.convert_model 함수는 모델 객체와 모델을 분석하기 위한 입력을 받아서, ov.Model 인스턴스를 리턴한다. 리턴된 모델은 특정 장치용으로 로딩하거나 ov.save_model을 사용하여 다음 추론을 위해 저장될 수 있다.

```python
from models.experimental import attempt_load
import torch
import openvino as ov
from models.yolo import Detect, DualDDetect
from utils.general import yaml_save, yaml_load
from pathlib import Path

MODEL_DIR = Path("/content/drive/MyDrive/data/bin/")
weights = MODEL_DIR / "best.pt"
ov_model_path = MODEL_DIR / weights.name.replace(".pt", "_openvino_model") / weights.name.replace(".pt", ".xml")

if not ov_model_path.exists():
    model = attempt_load(weights, device="cpu", inplace=True, fuse=True)
    metadata = {"stride": int(max(model.stride)), "names": model.names}

    model.eval()
    for k, m in model.named_modules():
        if isinstance(m, (Detect, DualDDetect)):
            m.inplace = False
            m.dynamic = True
            m.export = True

    example_input = torch.zeros((1, 3, 640, 640))
    model(example_input)

    ov_model = ov.convert_model(model, example_input=example_input)
```

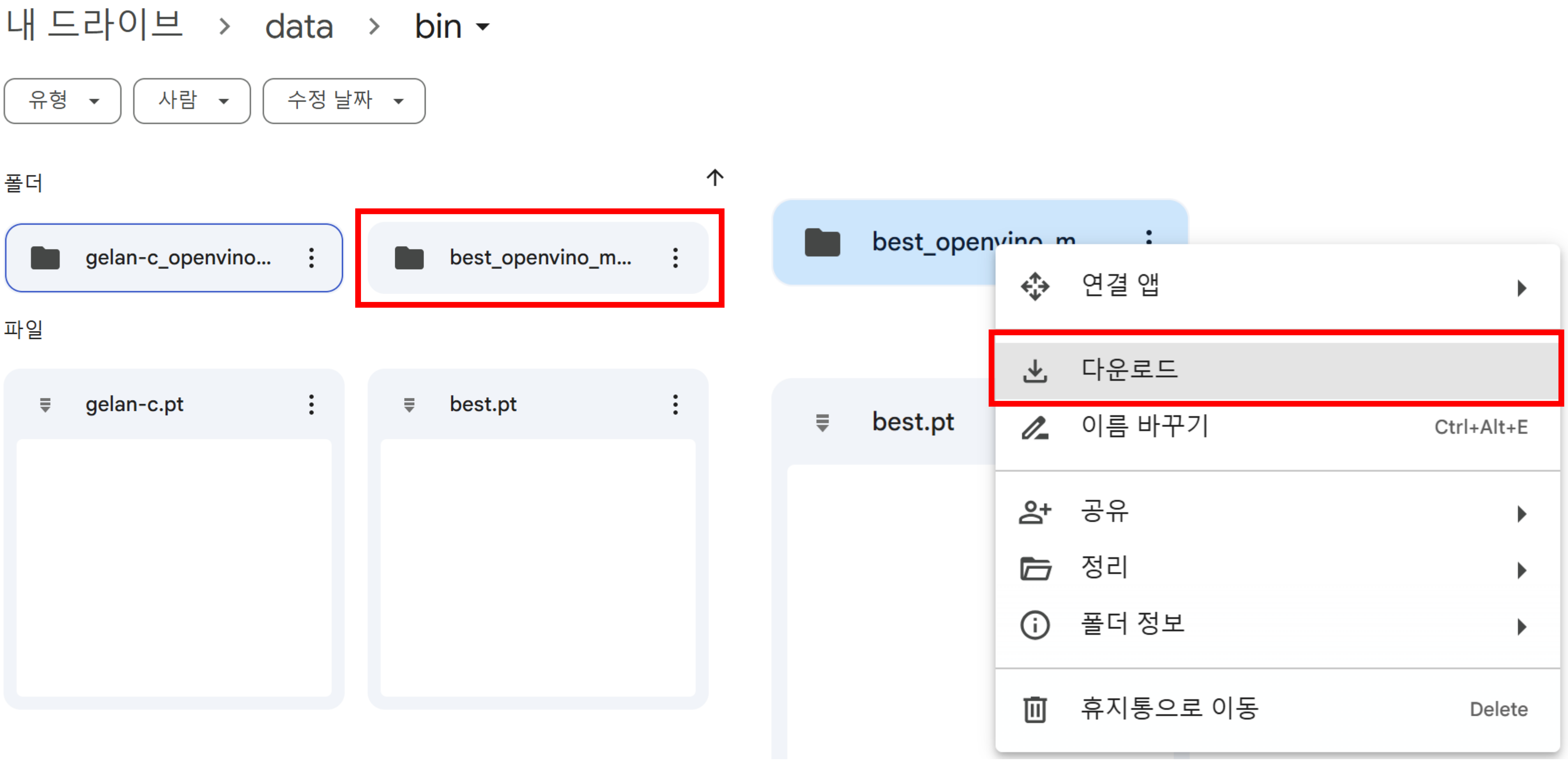
YOLOv9모델 OpenVINO 모델로 변환



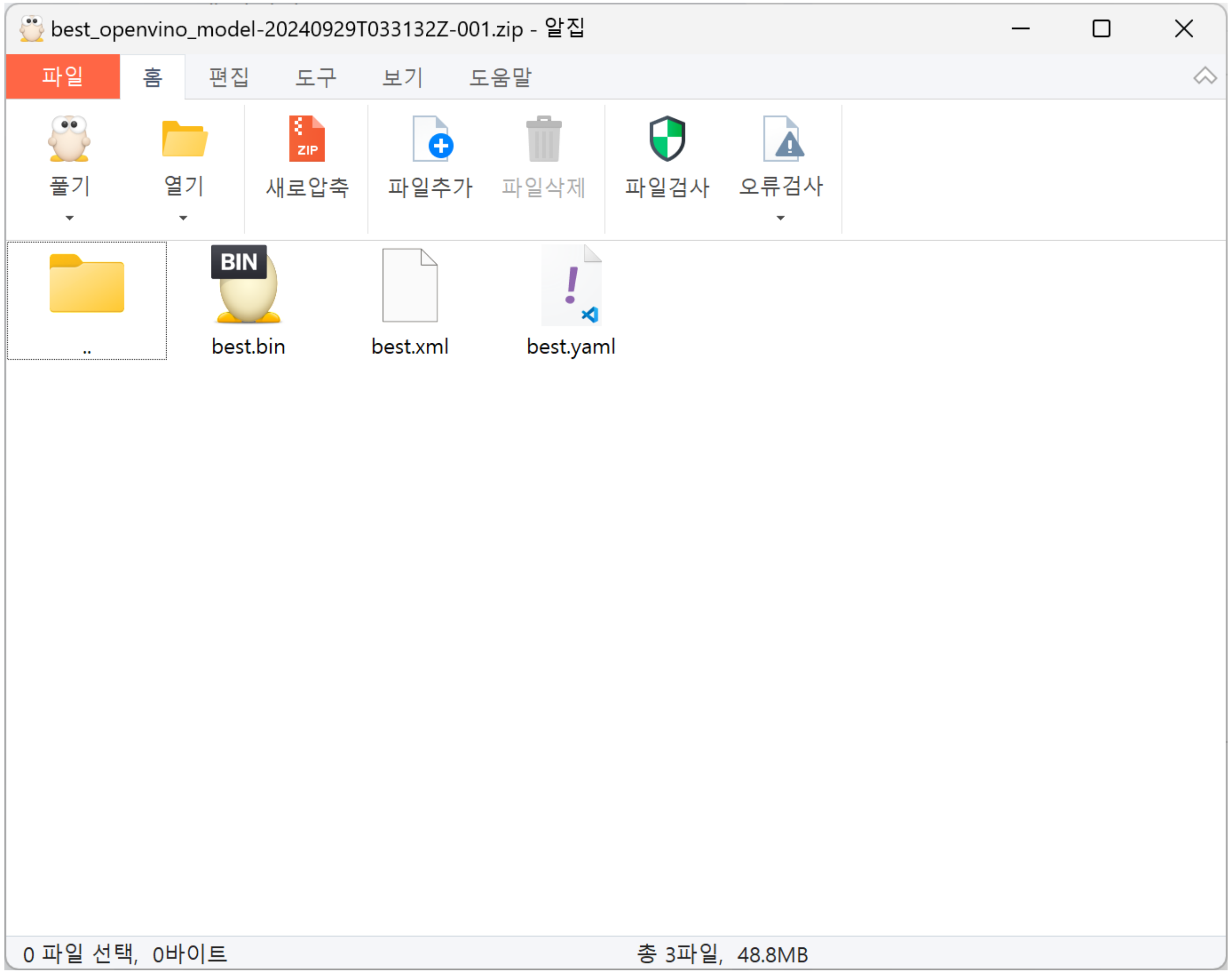
```
    # specify input and output names for compatibility with yolov9 repo interface
    ov_model.outputs[0].get_tensor().set_names({"output0"})
    ov_model.inputs[0].get_tensor().set_names({"images"})
    ov.save_model(ov_model, ov_model_path)
    # save metadata
    yaml_save(ov_model_path.parent / weights.name.replace(".pt", ".yaml"), metadata)
else:
    metadata = yaml_load(ov_model_path.parent / weights.name.replace(".pt", ".yaml"))
```

```
/content/yolov9/models/experimental.py:243: FutureWarning: You are using `torch.load` with `weights_only=False` (the current defa
  ckpt = torch.load(attempt_download(w), map_location='cpu')  # load
Fusing layers...
gelan-c summary: 387 layers, 25233256 parameters, 0 gradients, 101.8 GFLOPs
/content/yolov9/models/yolo.py:108: TracerWarning: Converting a tensor to a Python boolean might cause the trace to be incorrect.
  elif self.dynamic or self.shape != shape:
```

OpenVINO 모델 다운로드



OpenVINO 모델 다운로드



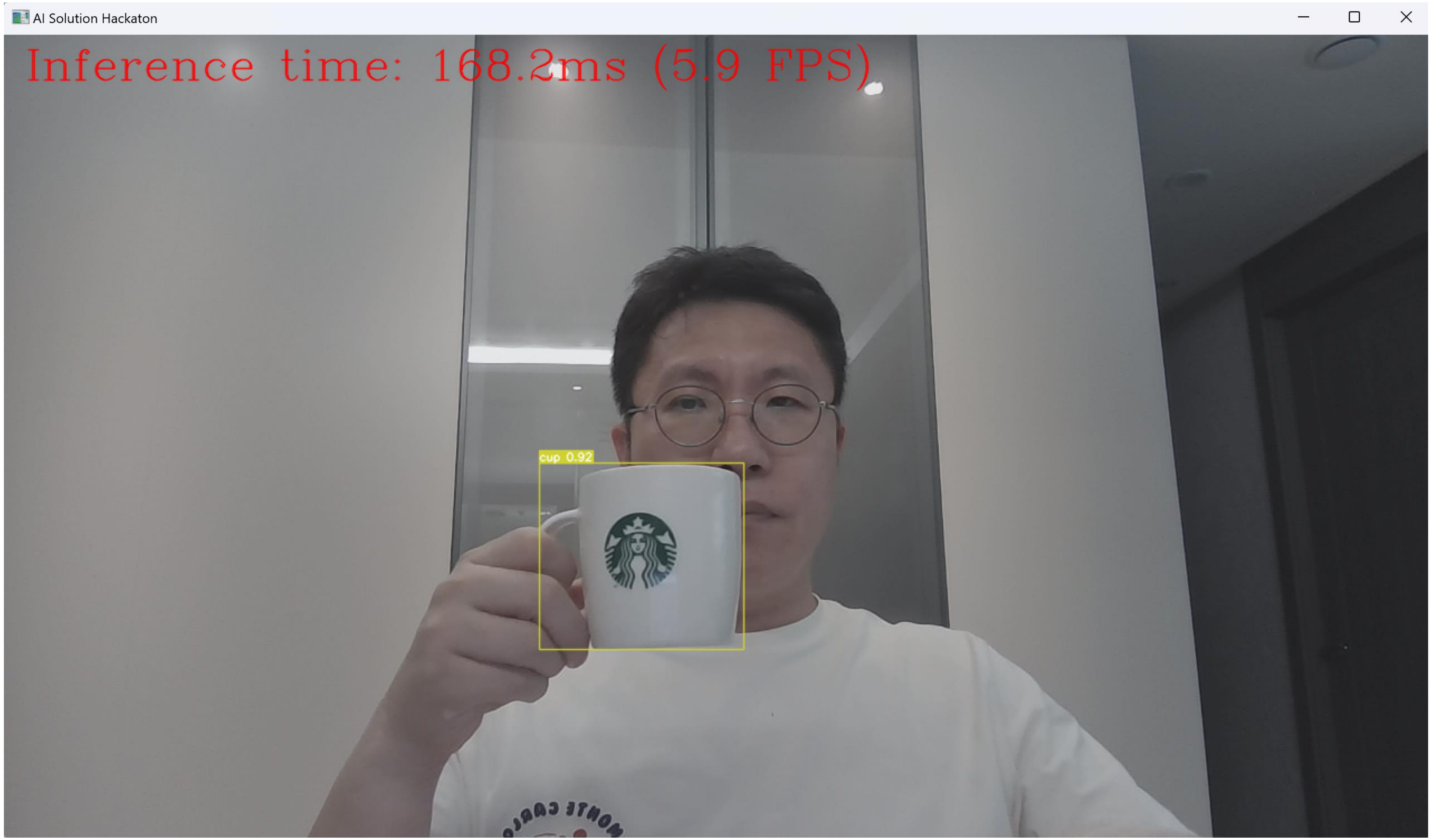
Camera 구동 프로그램

```
cd C:/camera

conda create –n camera_env python=3.11

conda activate camera_env

pip install -q "openvino>=2023.1.0”

pip install openvino-dev

pip install opencv-python

python solution.py
```

Camera 구동 프로그램

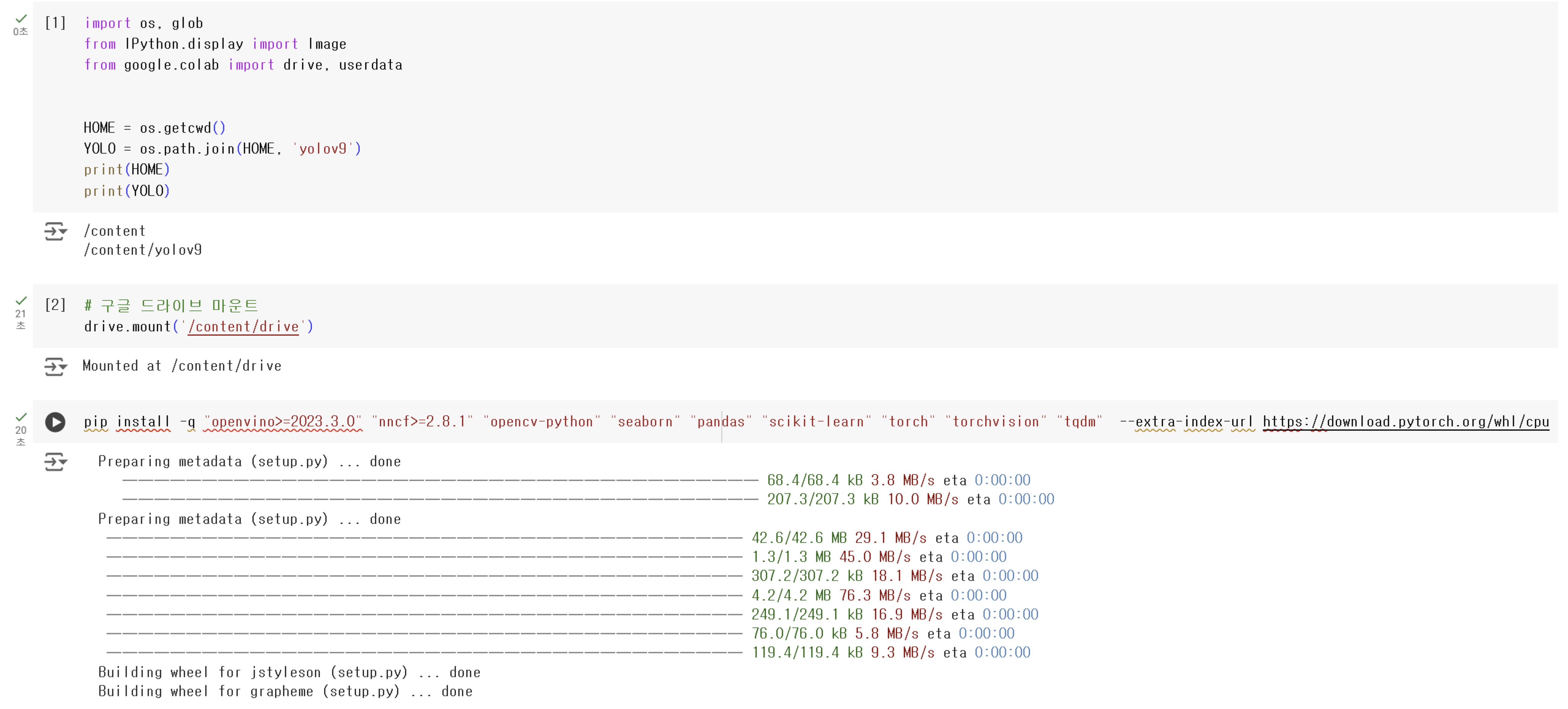
```
# Webcam
VIDEO_SOURCE = 0
#file
#VIDEO_SOURCE = 'test.mp4'
#VIDEO_SOURCE =
'https://storage.openvinotoolkit.org/repositories/openvino_notebooks/data/data/video/people
.mp4'

source=VIDEO_SOURCE
flip=True
use_popup=True
skip_first_frames=0
player = None

try:
    # Create a video player to play with target fps.
    player = VideoPlayer(source=source, flip=flip, fps=10,
skip_first_frames=skip_first_frames)
```

Camera 구동 프로그램



```
Inference time: 168.2ms (5.9 FPS)
```

양자화 기반 OpenVINO 모델 변환



```
import os, glob
from IPython.display import Image
from google.colab import drive, userdata


HOME = os.getcwd()
YOLO = os.path.join(HOME, 'yolov9')
print(HOME)
print(YOLO)
```

```
/content
/content/yolov9
```

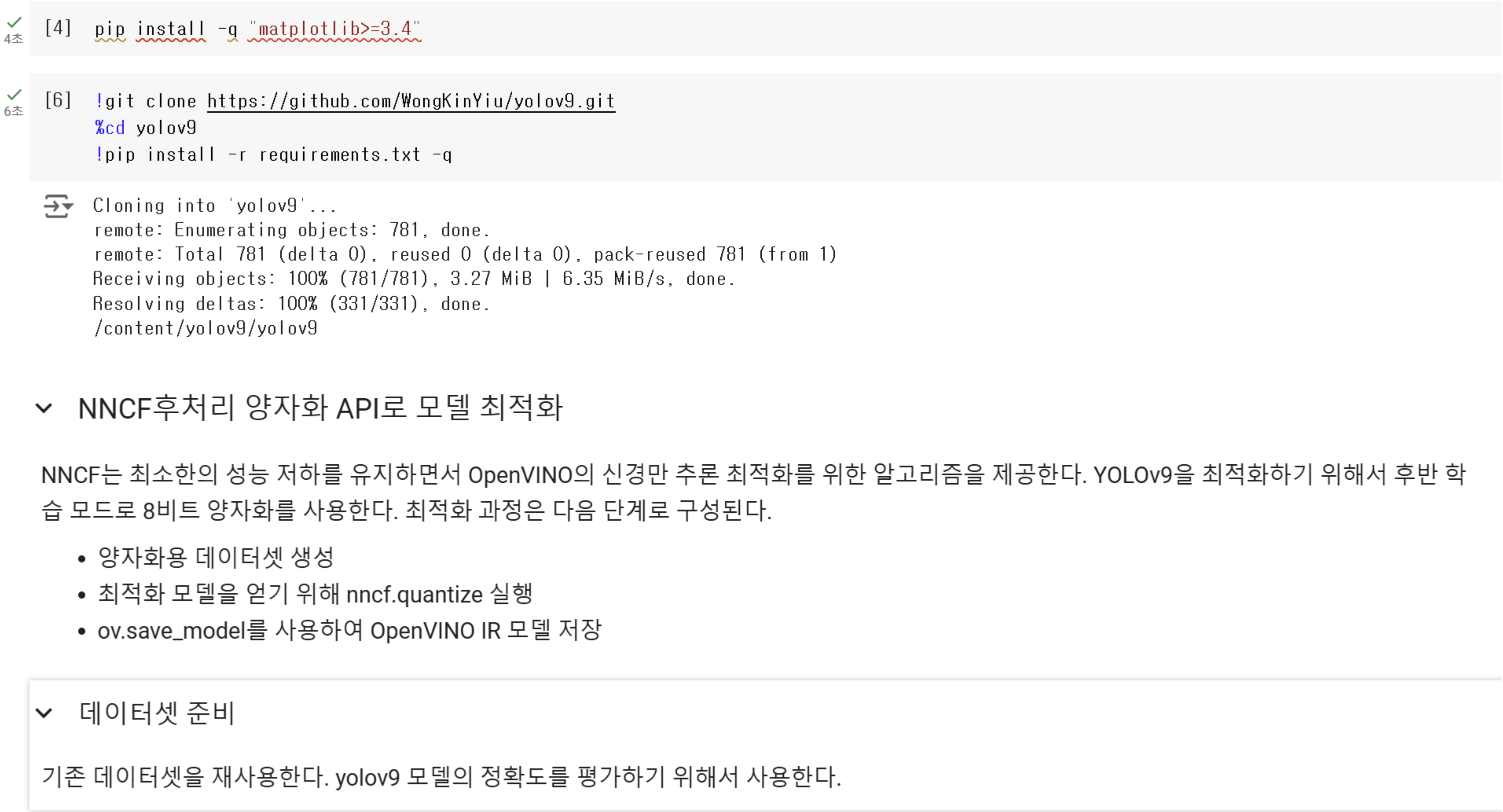
```
# 구글 드라이브 마운트
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
pip install -q "openvino>=2023.3.0" "nncf>=2.8.1" "opencv-python" "seaborn" "pandas" "scikit-learn" "torch" "torchvision" "tqdm"  --extra-index-url https://download.pytorch.org/whl/cpu
```

```
  Preparing metadata (setup.py) ... done
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 68.4/68.4 kB 3.8 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 207.3/207.3 kB 10.0 MB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 42.6/42.6 MB 29.1 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.3/1.3 MB 45.0 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 307.2/307.2 kB 18.1 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 4.2/4.2 MB 76.3 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 249.1/249.1 kB 16.9 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 76.0/76.0 kB 5.8 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 119.4/119.4 kB 9.3 MB/s eta 0:00:00
  Building wheel for jstyleson (setup.py) ... done
  Building wheel for grapheme (setup.py) ... done
```

양자화 기반 OpenVINO 모델 변환



```
[4]  pip install -q "matplotlib>=3.4"
```

```
[6]  !git clone https://github.com/WongKinYiu/yolov9.git
     %cd yolov9
     !pip install -r requirements.txt -q
```

```
Cloning into 'yolov9'...
remote: Enumerating objects: 781, done.
remote: Total 781 (delta 0), reused 0 (delta 0), pack-reused 781 (from 1)
Receiving objects: 100% (781/781), 3.27 MiB | 6.35 MiB/s, done.
Resolving deltas: 100% (331/331), done.
/content/yolov9/yolov9
```

NNCF후처리 양자화 API로 모델 최적화

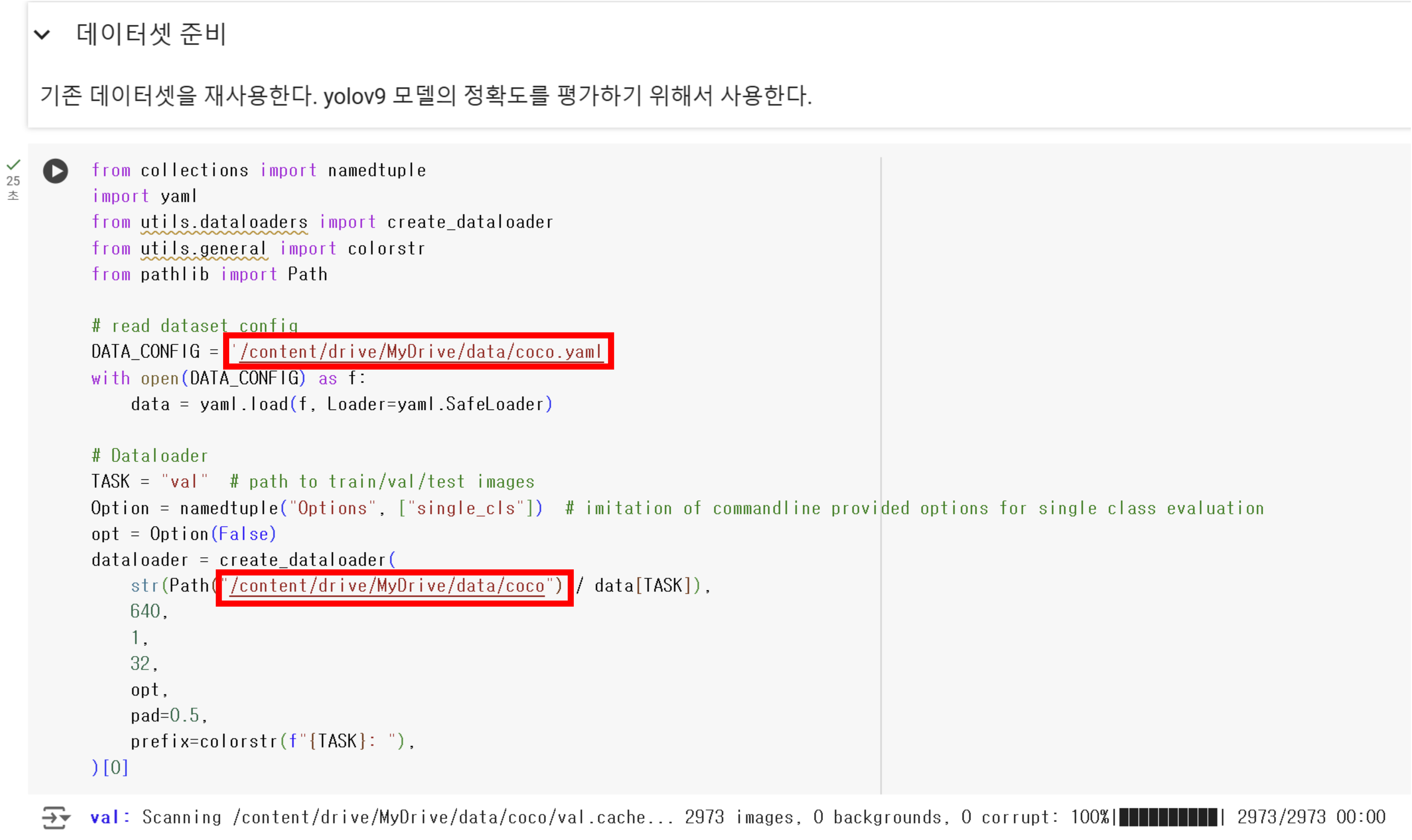
NNCF는 최소한의 성능 저하를 유지하면서 OpenVINO의 신경만 추론 최적화를 위한 알고리즘을 제공한다. YOLOv9을 최적화하기 위해서 후반 학습 모드로 8비트 양자화를 사용한다. 최적화 과정은 다음 단계로 구성된다.

- 양자화용 데이터셋 생성
- 최적화 모델을 얻기 위해 nncf.quantize 실행
- ov.save_model를 사용하여 OpenVINO IR 모델 저장

데이터셋 준비

기존 데이터셋을 재사용한다. yolov9 모델의 정확도를 평가하기 위해서 사용한다.

양자화 기반 OpenVINO 모델 변환



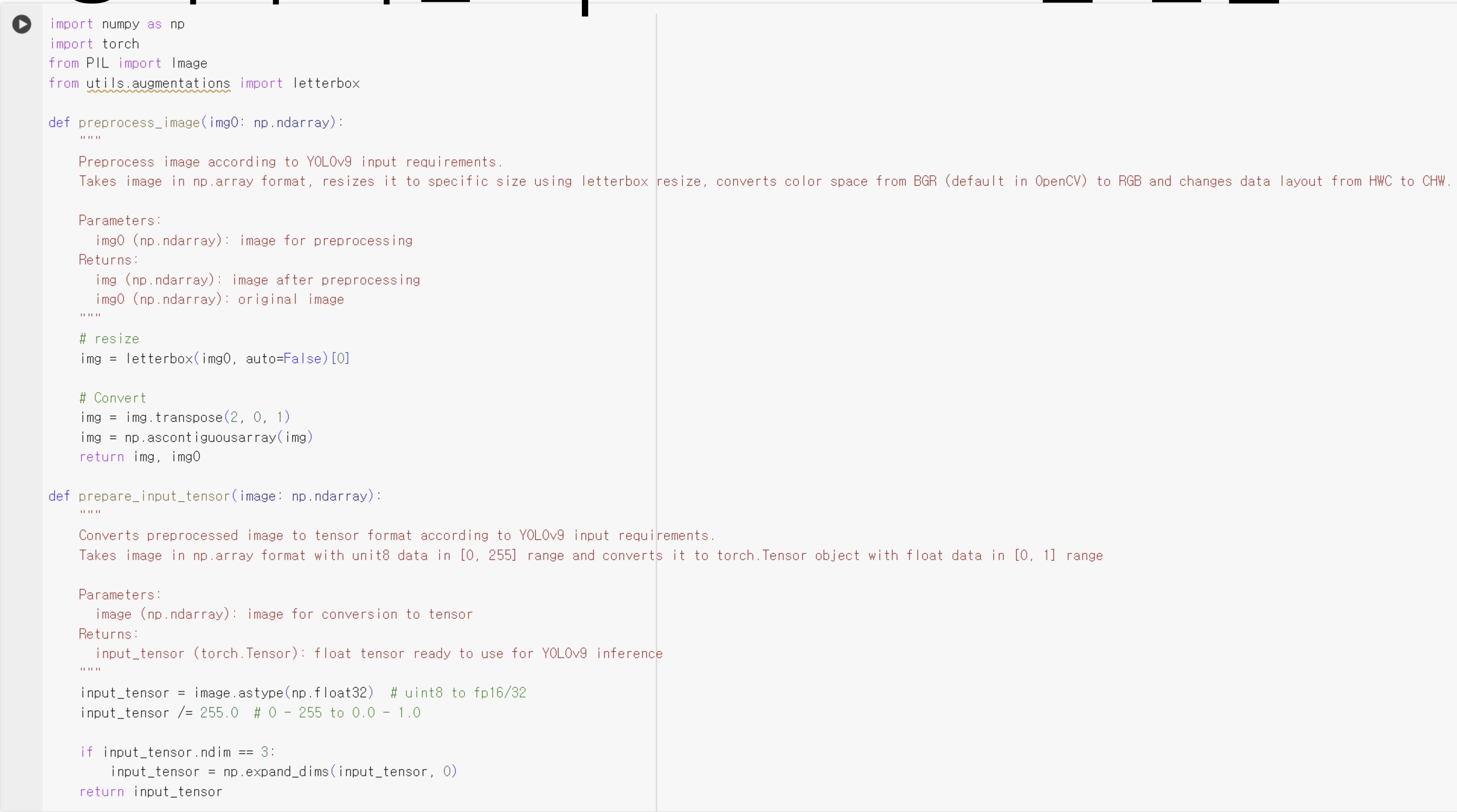
데이터셋 준비

기존 데이터셋을 재사용한다. yolov9 모델의 정확도를 평가하기 위해서 사용한다.

```python
from collections import namedtuple
import yaml
from utils.dataloaders import create_dataloader
from utils.general import colorstr
from pathlib import Path

# read dataset config
DATA_CONFIG = '/content/drive/MyDrive/data/coco.yaml'
with open(DATA_CONFIG) as f:
    data = yaml.load(f, Loader=yaml.SafeLoader)

# Dataloader
TASK = "val"  # path to train/val/test images
Option = namedtuple("Options", ["single_cls"])  # imitation of commandline provided options for single class evaluation
opt = Option(False)
dataloader = create_dataloader(
    str(Path("/content/drive/MyDrive/data/coco") / data[TASK]),
    640,
    1,
    32,
    opt,
    pad=0.5,
    prefix=colorstr(f"{TASK}: "),
)[0]
```

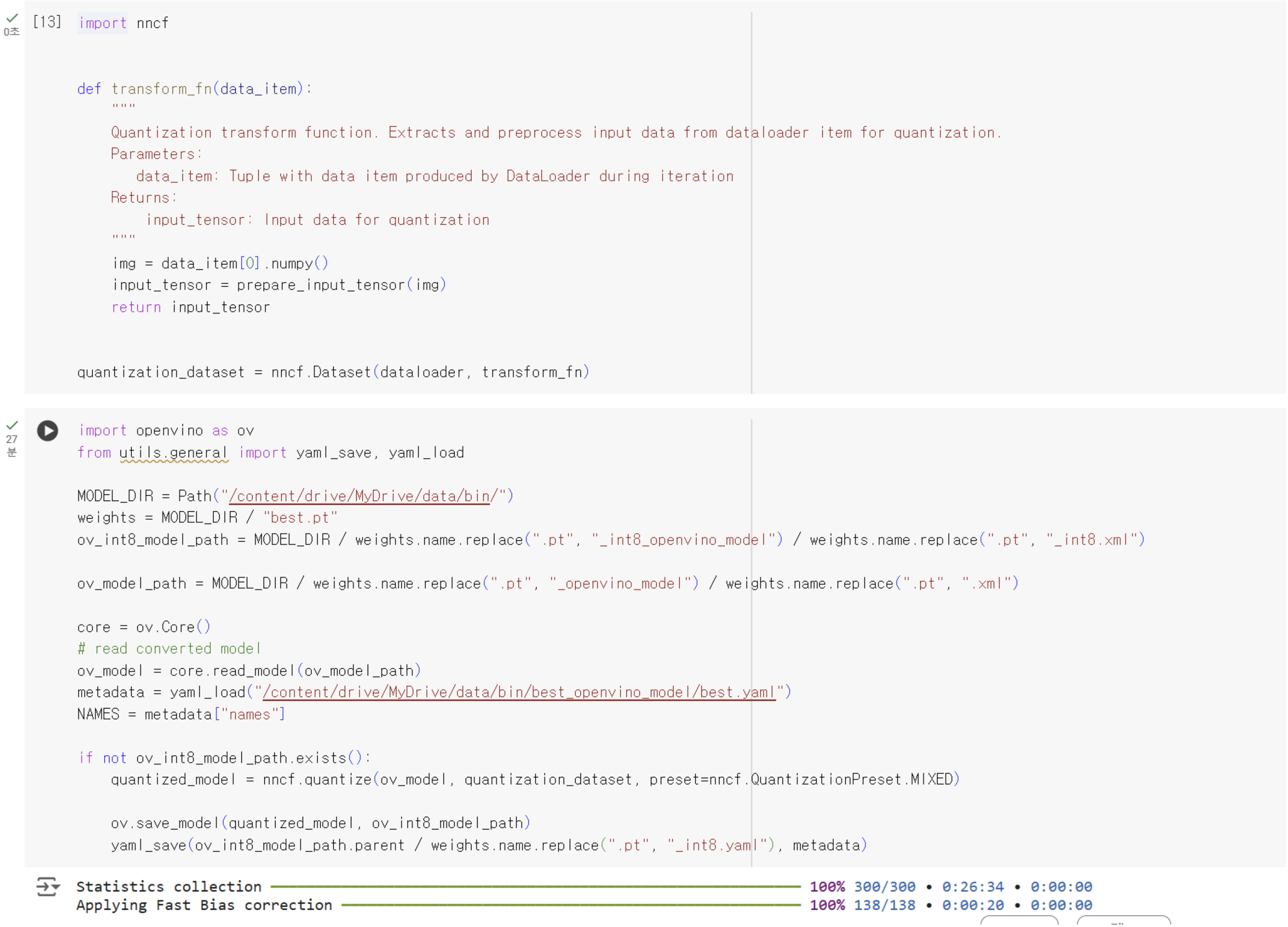
```
val: Scanning /content/drive/MyDrive/data/coco/val.cache... 2973 images, 0 backgrounds, 0 corrupt: 100%|██████████| 2973/2973 00:00
```

양자화 기반 OpenVINO 모델 변환



```python
import numpy as np
import torch
from PIL import Image
from utils.augmentations import letterbox

def preprocess_image(img0: np.ndarray):
    """
    Preprocess image according to YOLOv9 input requirements.
    Takes image in np.array format, resizes it to specific size using letterbox resize, converts color space from BGR (default in OpenCV) to RGB and changes data layout from HWC to CHW.

    Parameters:
      img0 (np.ndarray): image for preprocessing
    Returns:
      img (np.ndarray): image after preprocessing
      img0 (np.ndarray): original image
    """
    # resize
    img = letterbox(img0, auto=False)[0]

    # Convert
    img = img.transpose(2, 0, 1)
    img = np.ascontiguousarray(img)
    return img, img0

def prepare_input_tensor(image: np.ndarray):
    """
    Converts preprocessed image to tensor format according to YOLOv9 input requirements.
    Takes image in np.array format with unit8 data in [0, 255] range and converts it to torch.Tensor object with float data in [0, 1] range

    Parameters:
      image (np.ndarray): image for conversion to tensor
    Returns:
      input_tensor (torch.Tensor): float tensor ready to use for YOLOv9 inference
    """
    input_tensor = image.astype(np.float32)  # uint8 to fp16/32
    input_tensor /= 255.0  # 0 - 255 to 0.0 - 1.0

    if input_tensor.ndim == 3:
        input_tensor = np.expand_dims(input_tensor, 0)
    return input_tensor
```

양자화 기반 OpenVINO 모델 변환

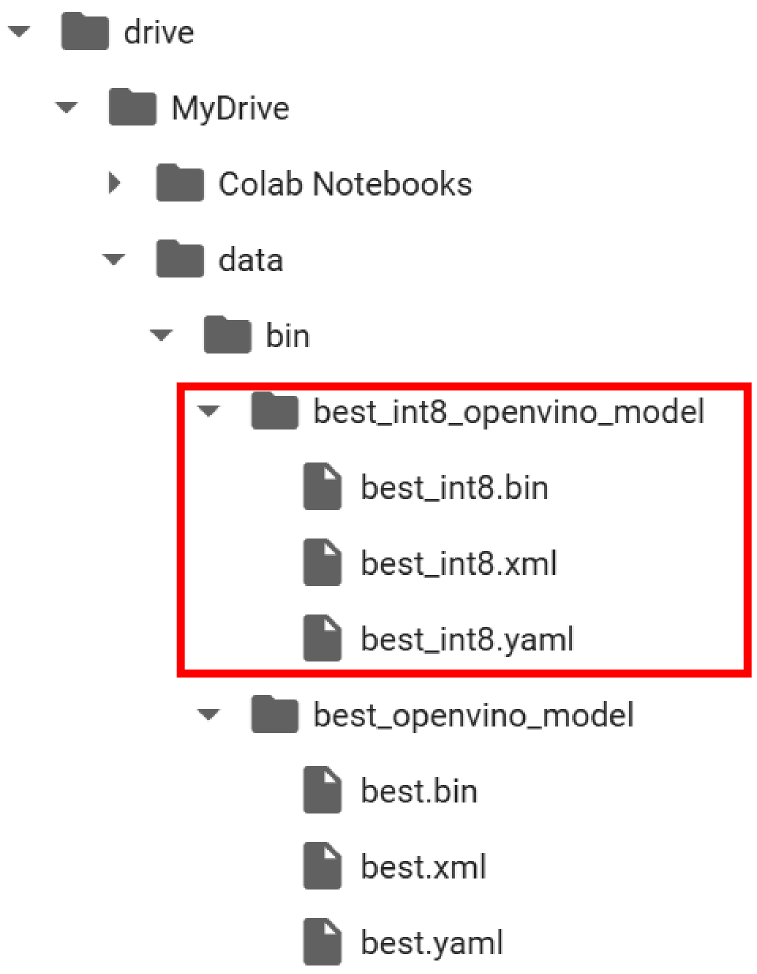


```python
import nncf


def transform_fn(data_item):
    """
    Quantization transform function. Extracts and preprocess input data from dataloader item for quantization.
    Parameters:
       data_item: Tuple with data item produced by DataLoader during iteration
    Returns:
        input_tensor: Input data for quantization
    """
    img = data_item[0].numpy()
    input_tensor = prepare_input_tensor(img)
    return input_tensor


quantization_dataset = nncf.Dataset(dataloader, transform_fn)
```

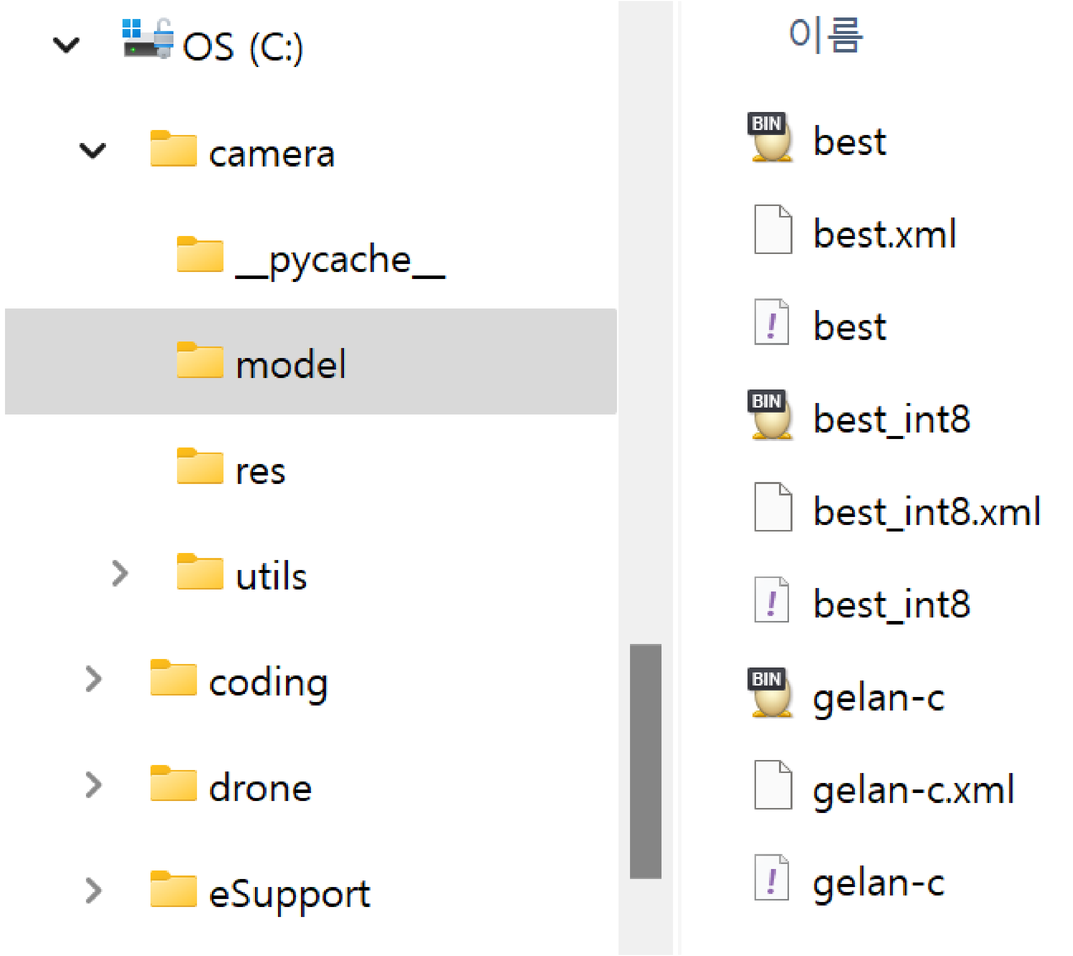
```python
import openvino as ov
from utils.general import yaml_save, yaml_load

MODEL_DIR = Path("/content/drive/MyDrive/data/bin/")
weights = MODEL_DIR / "best.pt"
ov_int8_model_path = MODEL_DIR / weights.name.replace(".pt", "_int8_openvino_model") / weights.name.replace(".pt", "_int8.xml")

ov_model_path = MODEL_DIR / weights.name.replace(".pt", "_openvino_model") / weights.name.replace(".pt", ".xml")

core = ov.Core()
# read converted model
ov_model = core.read_model(ov_model_path)
metadata = yaml_load("/content/drive/MyDrive/data/bin/best_openvino_model/best.yaml")
NAMES = metadata["names"]

if not ov_int8_model_path.exists():
    quantized_model = nncf.quantize(ov_model, quantization_dataset, preset=nncf.QuantizationPreset.MIXED)

    ov.save_model(quantized_model, ov_int8_model_path)
    yaml_save(ov_int8_model_path.parent / weights.name.replace(".pt", "_int8.yaml"), metadata)
```

```
Statistics collection ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 100% 300/300 • 0:26:34 • 0:00:00
Applying Fast Bias correction ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 100% 138/138 • 0:00:20 • 0:00:00
```

양자화 기반 OpenVINO 모델 변환



```
drive
  MyDrive
    Colab Notebooks
    data
      bin
        best_int8_openvino_model
          best_int8.bin
          best_int8.xml
          best_int8.yaml
        best_openvino_model
          best.bin
          best.xml
          best.yaml
```



```
OS (C:)
  camera
    __pycache__
    model
    res
    utils
  coding
  drone
  eSupport
```

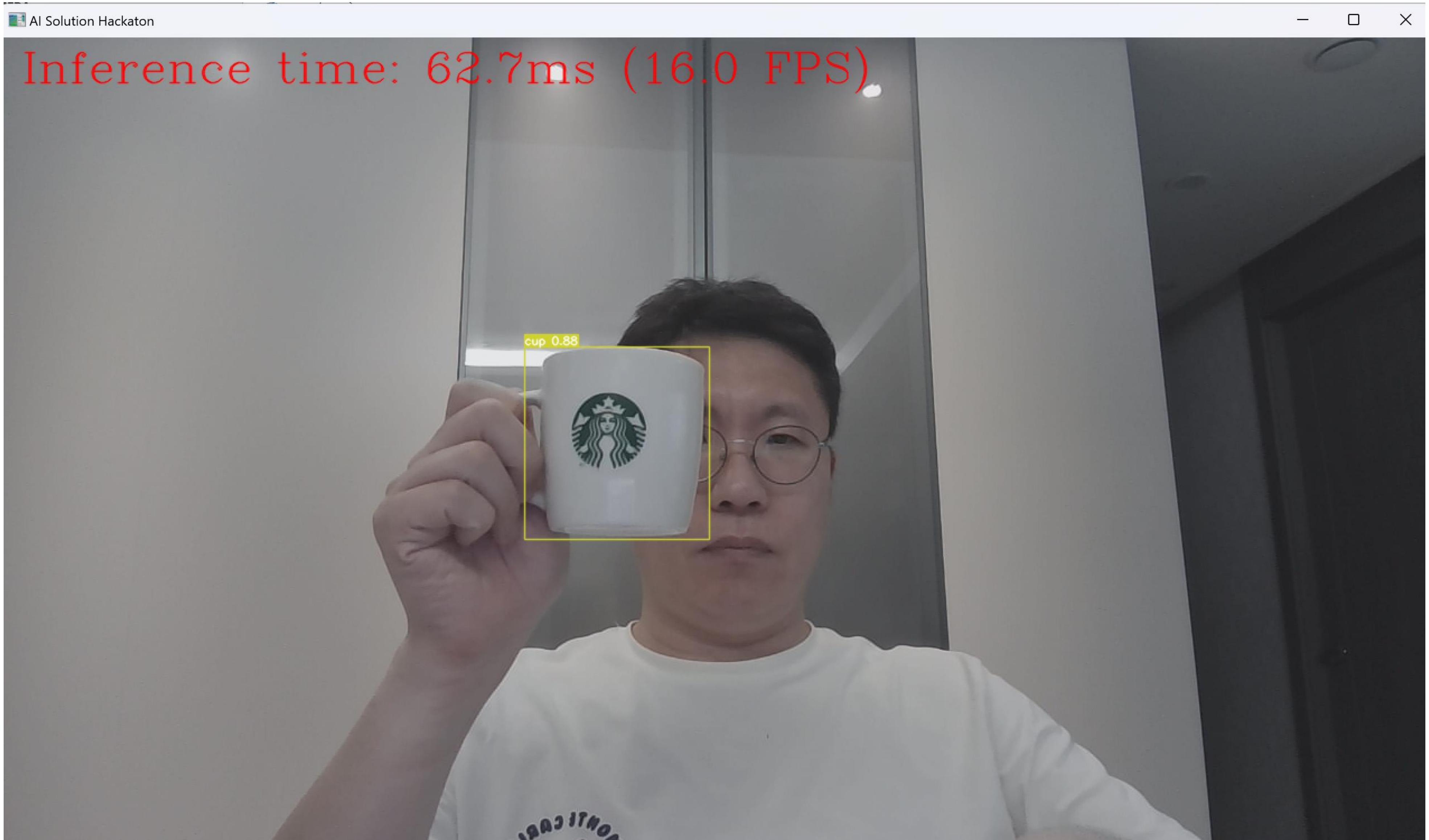
이름

```
best
best.xml
best
best_int8
best_int8.xml
best_int8
gelan-c
gelan-c.xml
gelan-c
```

Camera 구동 프로그램

```
cd C:/camera

conda create –n camera_env python=3.11

conda activate camera_env

pip install -q "openvino>=2023.1.0”

pip install openvino-dev

pip install opencv-python

python quant.py
```

양자화 기반 OpenVINO 모델 변환



Inference time: 62.7ms (16.0 FPS)

5.9FPS

↓

16.0FPS