KOOKMIN UNIVERSITY

BentoML

소프트웨어융합대학원
진혜진

실습(YOLO(PyTorch) 모델을 BentoML로 서빙)

- 가상환경(venv) 생성 및 활성화
  - python -m venv venv
  - .\venv\Scripts\activate

- 필수 패키지 설치 (CPU 기준)
  - pip install --upgrade pip
  - pip install bentoml ultralytics torch torchvision

C:₩Windows₩System32₩cmd.exe - python -m bentoml serve

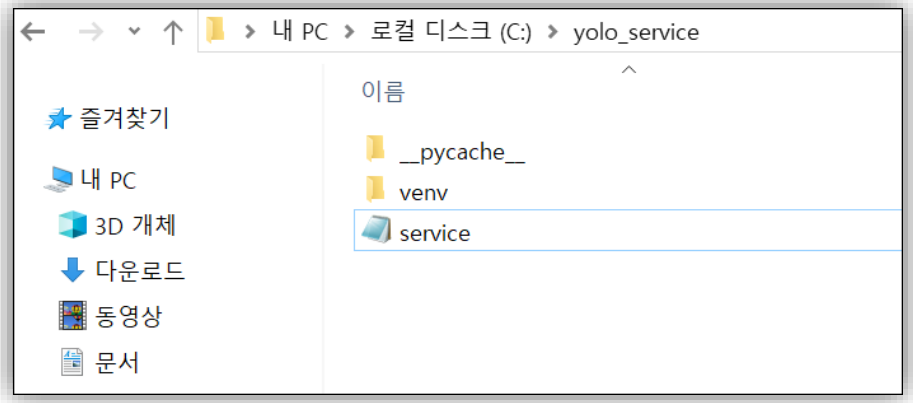
```
(venv) C:₩yolo_service>pip install bentoml ultralytics torch torchvision
Collecting bentoml
  Using cached bentoml-1.4.26-py3-none-any.whl.metadata (16 kB)
Collecting ultralytics
  Using cached ultralytics-8.3.217-py3-none-any.whl.metadata (37 kB)
Collecting torch
  Using cached torch-2.9.0-cp313-cp313-win_amd64.whl.metadata (30 kB)
Collecting torchvision
  Using cached torchvision-0.24.0-cp313-cp313-win_amd64.whl.metadata (5.9 kB)
Collecting a2wsgi>=1.10.7 (from bentoml)
  Using cached a2wsgi-1.10.10-py3-none-any.whl.metadata (4.0 kB)
Collecting aiohttp (from bentoml)
```

실습(YOLO(PyTorch) 모델을 BentoML로 서빙)

service.py

```python
# service.py
from __future__ import annotations
import bentoml

@bentoml.service
class Hello:
    @bentoml.api
    def greet(self, name: str = "Bento") -> str:
        return f" 풀스택서비스구축, {name}! 🎉"
```



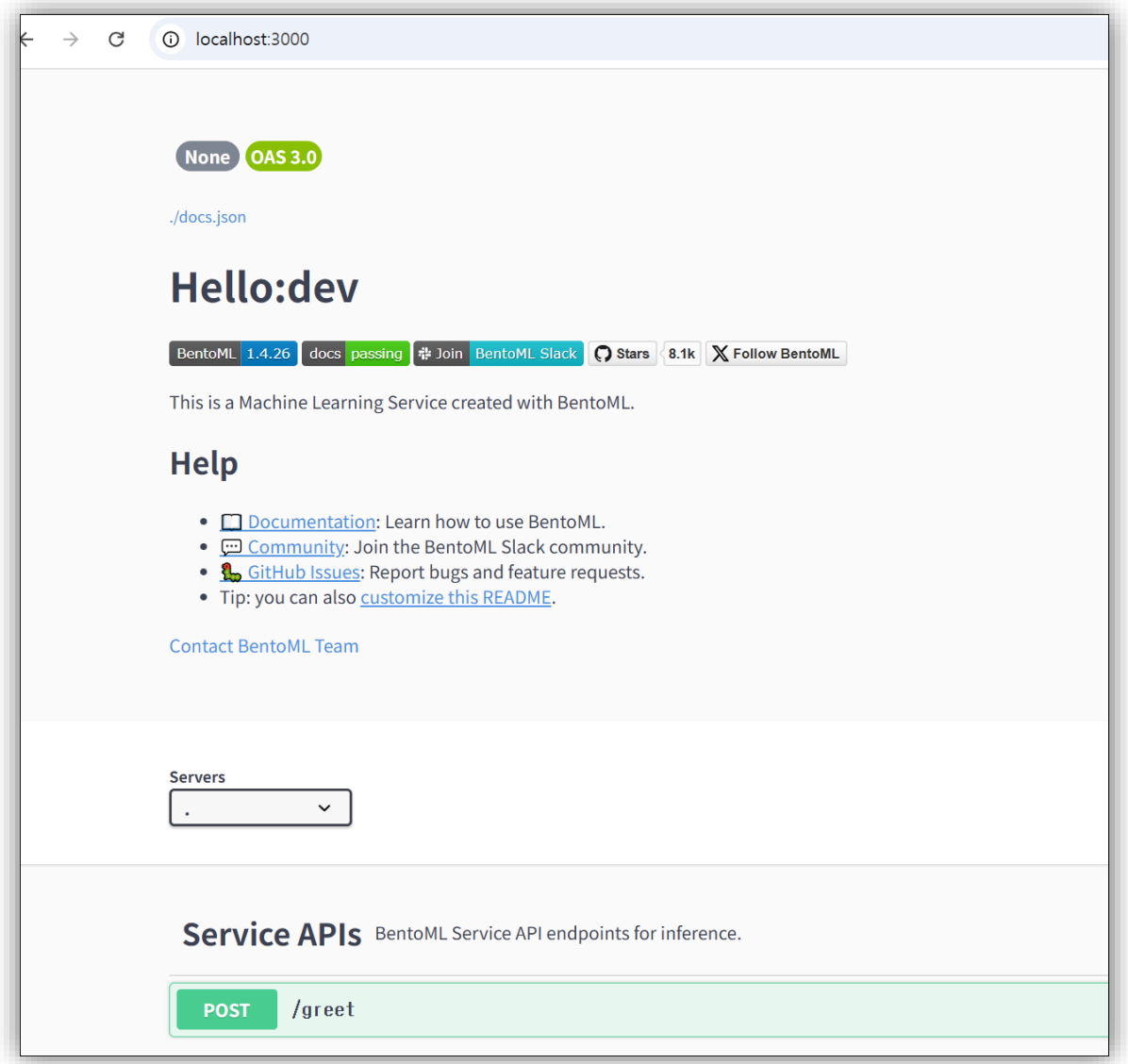
■ 서버 실행

- python -m bentoml serve

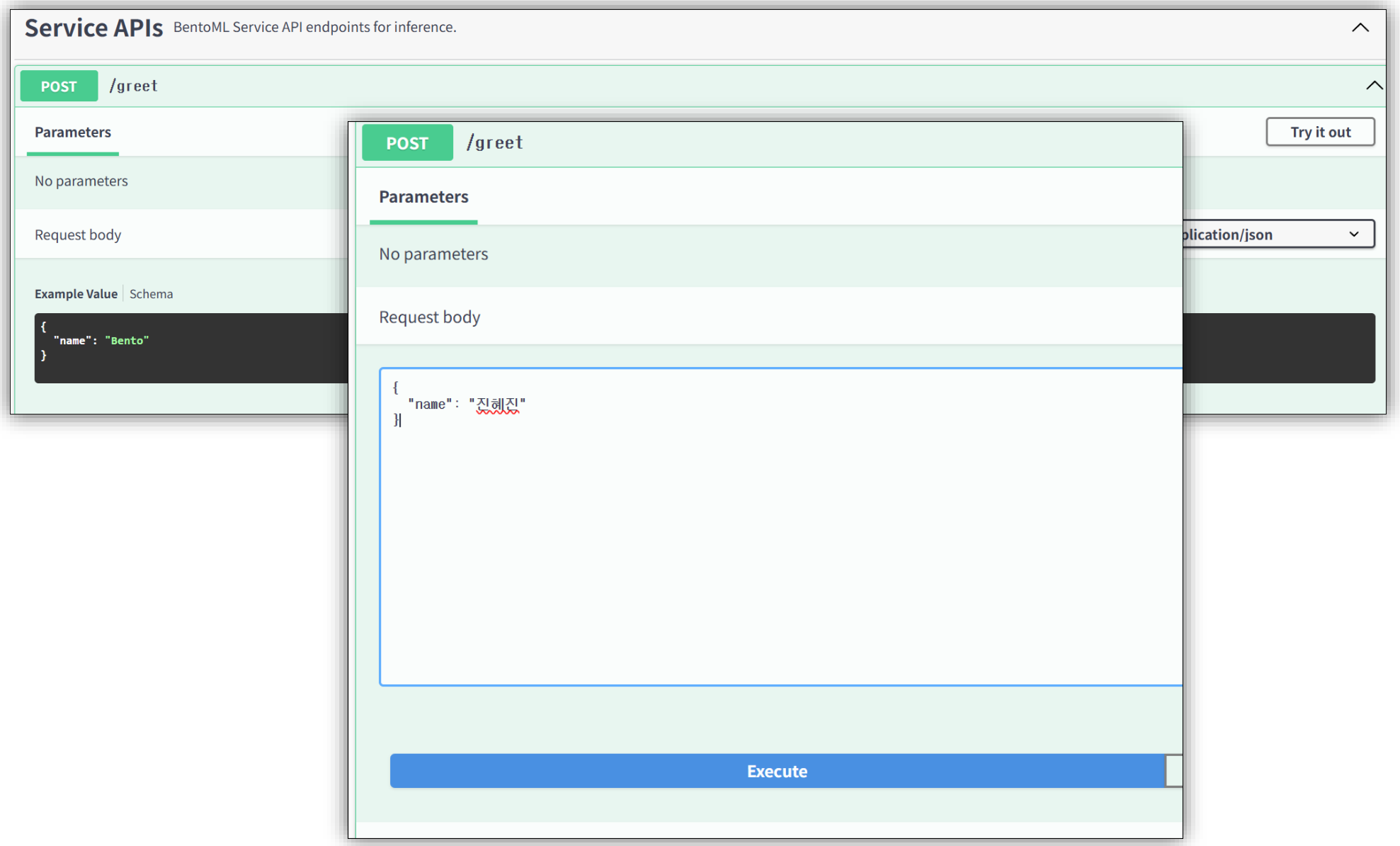
```
C:₩Windows₩System32₩cmd.exe - python -m bentoml serve
(venv) C:₩yolo_service>python -m bentoml serve
2025-10-18T10:43:10+0900 [INFO] [cli] Loading service from default location 'service.py'
2025-10-18T10:43:10+0900 [INFO] [cli] Loading service from default location 'service.py'
2025-10-18T10:43:11+0900 [INFO] [cli] Starting production HTTP BentoServer from "." listening on http:
//localhost:3000 (Press CTRL+C to quit)
2025-10-18T10:43:11+0900 [INFO] [:1] Loading service from default location 'service.py'
2025-10-18T10:43:12+0900 [INFO] [:1] Service Hello initialized
2025-10-18T10:43:27+0900 [INFO] [:1] 127.0.0.1:63598 (scheme=http,method=GET,path=/,type=,length=) (st
atus=200,type=text/html; charset=utf-8,length=2945) 136.015ms (trace=eb4f00a03920a19cfa93e14f4cac1d07,
span=54126877b3fc2320,sampled=0,service.name=Hello)
2025-10-18T10:43:27+0900 [INFO] [:1] 127.0.0.1:63598 (scheme=http,method=GET,path=/static_content/swag
ger-ui.css,type=,length=) (status=200,type=text/css; charset=utf-8,length=152059) 3.019ms (trace=4778d
9759004144c40b93bf84356563c,span=e822b85ffbc7591e,sampled=0,service.name=Hello)
```

실습(YOLO(PyTorch) 모델을 BentoML로 서빙)

- http://localhost:3000/



실습(YOLO(PyTorch) 모델을 BentoML로 서빙)



실습(YOLO(PyTorch) 모델을 BentoML로 서빙)

KOOKMIN UNIVERSITY

■ 응답 확인

파일 C:/Users/hjjin/Downloads/response_1760752672523.html

풀스택서비스구축, 진혜진! 🎉

Responses

Curl

```
curl -X 'POST' \
  'http://localhost:3000/greet' \
  -H 'accept: text/plain' \
  -H 'Content-Type: application/json' \
  -d '{
  "name": "진혜진"
}'
```

Request URL

```
http://localhost:3000/greet
```

Server response

Code	Details
200	Response body

```
풀스택서비스구축, 진혜진! 🎉
```

Download

실습(YOLO(PyTorch) 모델을 BentoML로 서빙)

- YOLO(Object Detection) 서비스 만들기
  - Ultralytics YOLOv5s 모델을 로드해, 이미지 업로드 → 바운딩박스 결과(JSON)

```
# service.py
from __future__ import annotations
from typing import List, Dict, Any
import io

import bentoml
from PIL import Image

@bentoml.service
class YOLOService:
    def __init__(self) -> None:

        from ultralytics import YOLO

        self.model = YOLO("yolov5s.pt")
```

실습(YOLO(PyTorch)


```python
    self.model = YOLO("yolov5s.pt")

@bentoml.api
def detect(self, image: Image.Image) -> List[Dict[str, Any]]:
    """
    입력: 이미지 파일 (multipart/form-data 'image')
    출력: [ {class_id, class_name, confidence, bbox_xyxy}, ... ]
    """
    results = self.model(image)
    r0 = results[0]
    boxes = r0.boxes
    names = r0.names

    output: List[Dict[str, Any]] = []
    for box in boxes:
        cls_id = int(box.cls[0])
        conf = float(box.conf[0])
        x1, y1, x2, y2 = [float(v) for v in box.xyxy[0].tolist()]
        output.append(
            {
                "class_id": cls_id,
                "class_name": names.get(cls_id, str(cls_id)),
                "confidence": round(conf, 4),
                "bbox_xyxy": [x1, y1, x2, y2],
            }
        )
    return output

@bentoml.api
def detect_image(self, image: Image.Image) -> Image.Image:

    results = self.model(image)
    r0 = results[0]

    plotted = r0.plot()

    img = Image.fromarray(plotted[:, :, ::-1])
    return img
```

실습(YOLO(PyTorch) 모델을 BentoML로 서빙)

■ 실행

- python service.py
- python -m bentoml serve

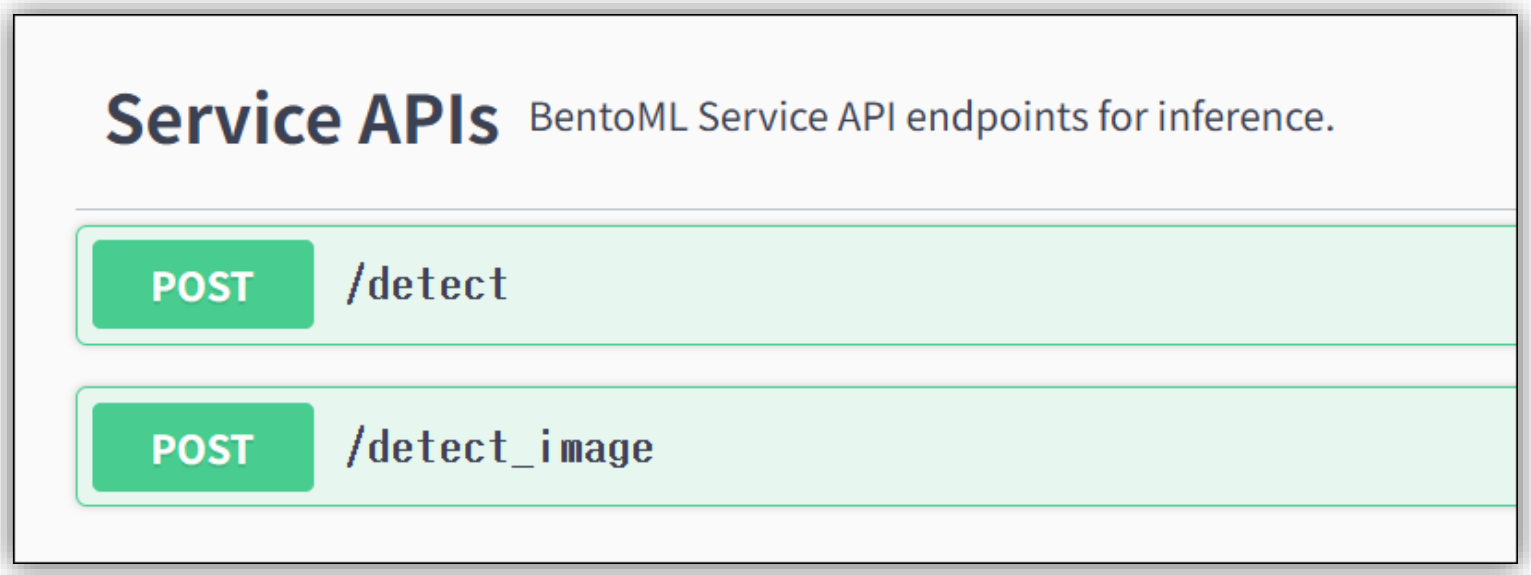
C:₩Windows₩System32₩cmd.exe - python -m bentoml serve

```
(venv) C:₩yolo_service>python service.py

(venv) C:₩yolo_service>python -m bentoml serve
2025-10-18T11:10:57+0900 [INFO] [cli] Loading service from default location 'service.py'
2025-10-18T11:10:57+0900 [INFO] [cli] Loading service from default location 'service.py'
2025-10-18T11:10:58+0900 [INFO] [cli] Starting production HTTP BentoServer from "." listening on htt
//localhost:3000 (Press CTRL+C to quit)
2025-10-18T11:10:58+0900 [INFO] [:1] Loading service from default location 'service.py'
PRO TIP  Replace 'model=yolov5s.pt' with new 'model=yolov5su.pt'.
YOLOv5 'u' models are trained with https://github.com/ultralytics/ultralytics and feature improved p
formance vs standard YOLOv5 models trained with https://github.com/ultralytics/yolov5.
```

실습(YOLO(PyTorch) 모델을 BentoML로 서빙)

- 테스트
  - http://localhost:3000/
  - Swagger UI



실습(YOLO(PyTorch) 모델을 BentoML로 서빙)

KOOKMIN UNIVERSITY

Code | Details

200

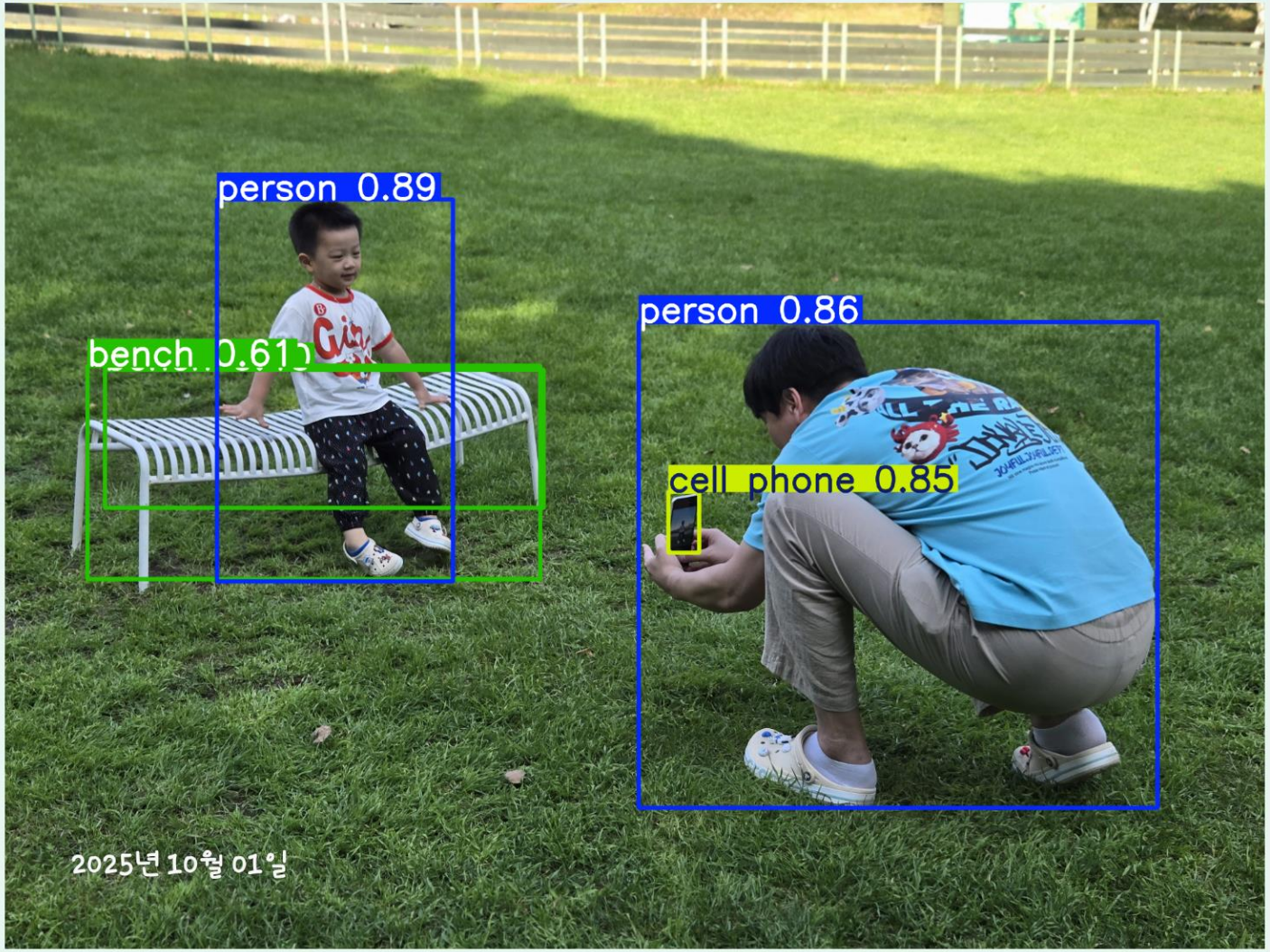
Response body


```
[
  {
    "class_id": 0,
    "class_name": "person",
    "confidence": 0.8864,
    "bbox_xyxy": [
      673.7805786132812,
      623.9683227539062,
      1422.970458984375,
      1834.655029296875
    ]
  },
  {
    "class_id": 0,
    "class_name": "person",
    "confidence": 0.8587,
    "bbox_xyxy": [
      2012.352783203125,
      1012.474853515625,
      3659.3056640625,
      2552.7939453125
    ]
  },
  {
    "class_id": 67,
    "class_name": "cell phone",
    "confidence": 0.852,
```


Code | Details

200

Response body



PyTorch 모델 저장/관리(BentoML Model Store)

- save_model.py

```python
# save_model.py
from ultralytics import YOLO
import bentoml


y = YOLO("yolov5s.pt")
torch_model = y.model


tag = bentoml.pytorch.save_model(
    name="yolov5s_ultra",
    model=torch_model,
    signatures={"__call__": {"batchable": True}}
)
print("saved:", tag)
```

실습(YOLO(PyTorch) 모델을 BentoML로 서빙)

- 실행
  - python save_model.py
- 로컬 Model Store 목록 확인
  - bentoml models list
  - bentoml models get yolov5s_ultra:latest

```
(venv) C:₩yolo_service>bentoml models list
Tag                                Module            Size
yolov5s_ultra:vstbvjvlzkvbepd4     bentoml.pytorch   35.25 MiB
iris_rf_model:kp5lt3fgi6ueypd4     bentoml.sklearn   182.36 KiB
iris_rf_model:hyxk6zvgio46wpd4     bentoml.sklearn   182.22 KiB
```

```
(venv) C:₩yolo_service>bentoml models get yolov5s_ultra:latest
name:
version:
module:
labels:
options:
  partial_kwargs:
metadata:
context:
  framework_name:
  framework_versions:
    torch:
  bentoml_version:
  python_version:
signatures:
  __call__:
    batchable:
    batch_dim:
```