st125333_Assignment1_DLCV

August 19, 2025

[1]:
```
!python --version
```

```
Python 3.12.4
```

[2]:
```
import sys
print(sys.executable)
```

```
C:\Users\satna\miniconda3\envs\myConda\python.exe
```

[3]:
```
import torch
print(torch.version.cuda)
print(torch.cuda.is_available())
```

```
12.6
False
```

[4]:
```
!nvidia-smi
```

```
'nvidia-smi' is not recognized as an internal or external command,
operable program or batch file.
```

[7]:
```
from ultralytics import YOLO
```

[8]:
```
image_files = [f"{i}.jpg" for i in range(1, 6)]
```

[9]:
```
modelv8n = YOLO("yolov8n.pt")
results = modelv8n(image_files)
```

```
0: 640x640 1 car, 1 airplane, 1 umbrella, 2 kites, 1 cup, 165.1ms
1: 640x640 1 car, 1 cow, 1 umbrella, 2 kites, 1 cup, 1 chair, 1 potted plant, 1
vase, 165.1ms
2: 640x640 3 persons, 1 tv, 1 cell phone, 165.1ms
3: 640x640 4 cups, 1 bowl, 1 banana, 1 clock, 165.1ms
4: 640x640 1 bowl, 1 sandwich, 3 oranges, 1 carrot, 1 dining table, 165.1ms
Speed: 6.4ms preprocess, 165.1ms inference, 5.1ms postprocess per image at shape
(1, 3, 640, 640)
```

[10]:
```
for idx, result in enumerate(results):
    boxes = result.boxes  # Boxes object for bounding box outputs
```

```
    masks = result.masks  # Masks object for segmentation masks outputs
    keypoints = result.keypoints  # Keypoints object for pose outputs
    probs = result.probs  # Probs object for classification outputs
    obb = result.obb  # Oriented boxes object for OBB outputs
    result.show()  # display to screen
    result.save(filename=f"outputs/result_v8n_{idx}.jpg")  # save to disk
```

[11]:
```
modelv12n = YOLO("yolo12n.pt")
results = modelv12n(image_files)
```

```
0: 640x640 1 car, 1 kite, 1 cup, 1 apple, 185.7ms
1: 640x640 1 car, 2 horses, 1 umbrella, 1 cup, 185.7ms
2: 640x640 2 persons, 1 tv, 1 cell phone, 185.7ms
3: 640x640 2 cups, 3 bowls, 1 dining table, 185.7ms
4: 640x640 7 oranges, 2 dining tables, 185.7ms
Speed: 5.7ms preprocess, 185.7ms inference, 1.9ms postprocess per image at shape
(1, 3, 640, 640)
```

[12]:
```
for idx, result in enumerate(results):
    boxes = result.boxes  # Boxes object for bounding box outputs
    masks = result.masks  # Masks object for segmentation masks outputs
    keypoints = result.keypoints  # Keypoints object for pose outputs
    probs = result.probs  # Probs object for classification outputs
    obb = result.obb  # Oriented boxes object for OBB outputs
    result.show()  # display to screen
    result.save(filename=f"outputs/result_v12n_{idx}.jpg")  # save to disk
```

[ ]:
```
```