Lab Assignment #3

st125457 - Ulugbek Shernazarov

Tasks:

1. Capture screenshots on the following:
- Python Version
- Activated Conda Environment
- Device Name using "wmic path win32_VideoController get name" command. (for Windows Users Only or Specify your CPU such as M1 or M2 if you use other than Windows)
- CUDA Availability Check using torch.cuda.is_available(). (Expect True for GPU Devices and False for CPU Devices)
- Git Installation Completion (Git Bash or Git Clone Command Output)

1. Collect 5 different images from the public source(internet). Each image should include at least 5 different objects (eg. car, dog, cat, motorbike, TV). Run your YOLOv8 Pretrained Model with two different sizes (n and x), get the predictions. Are all objects detected from your images? why?

Task 1

Env Configuration

```
!python --version

Python 3.9.19

!conda --version

conda 22.9.0

!conda env list

# conda environments:
#
base                     D:\DataScience\Anaconda3
dl4cv                 *  D:\DataScience\Anaconda3\envs\dl4cv
petdjango                D:\DataScience\Anaconda3\envs\petdjango
quantory                 D:\DataScience\Anaconda3\envs\quantory


!wmic path win32_VideoController get name

Name
```

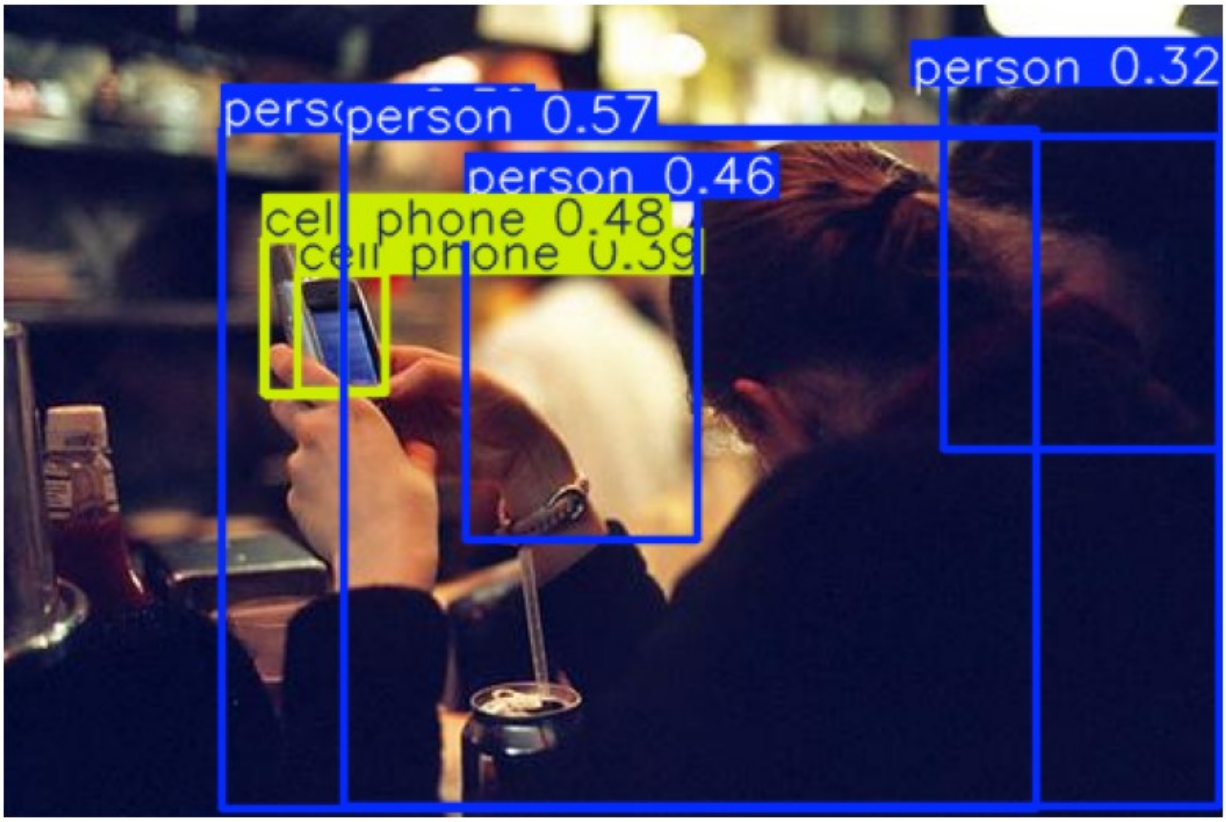
```
Intel(R) UHD Graphics 630

NVIDIA GeForce GTX 1650 with Max-Q Design



!git --version

git version 2.33.1.windows.1

import torch

torch.cuda.is_available()

True
```

Task2

Collect 5 different images from the public source(internet). Each image should include at least 5 different objects (eg. car, dog, cat, motorbike, TV). Run your YOLOv8 Pretrained Model with two different sizes (n and x), get the predictions. Are all objects detected from your images? why?

```
from ultralytics import YOLO
import cv2
import matplotlib.pyplot as plt
import numpy as np

import os

model1_path = os.path.join('models/yolov8n.pt')
model2_path = os.path.join('models/yolov8x.pt')

model1 = YOLO(model1_path)
model2 = YOLO(model2_path)

Downloading
https://github.com/ultralytics/assets/releases/download/v8.2.0/yolov8x
.pt to 'models\yolov8x.pt'...

100%|██████████| 131M/131M [00:04<00:00, 32.6MB/s]

img_list = os.listdir('dataset')

for img_name in img_list:
    img_path = os.path.join('dataset', img_name)

    results = model1(img_path, device='cpu')

    for result in results:
```

```
        rgb_image = result.plot()
        bgr_image = cv2.cvtColor(np.array(rgb_image),
cv2.COLOR_RGB2BGR)

        plt.figure(figsize=(10, 10))
        plt.imshow(bgr_image)
        plt.axis('off')
        plt.show()


image 1/1 d:\2021\AIT\Master's\DL4CV\homework3\dataset\1.jpg: 448x640
4 persons, 3 cell phones, 116.5ms
Speed: 5.0ms preprocess, 116.5ms inference, 3.0ms postprocess per
image at shape (1, 3, 448, 640)
```

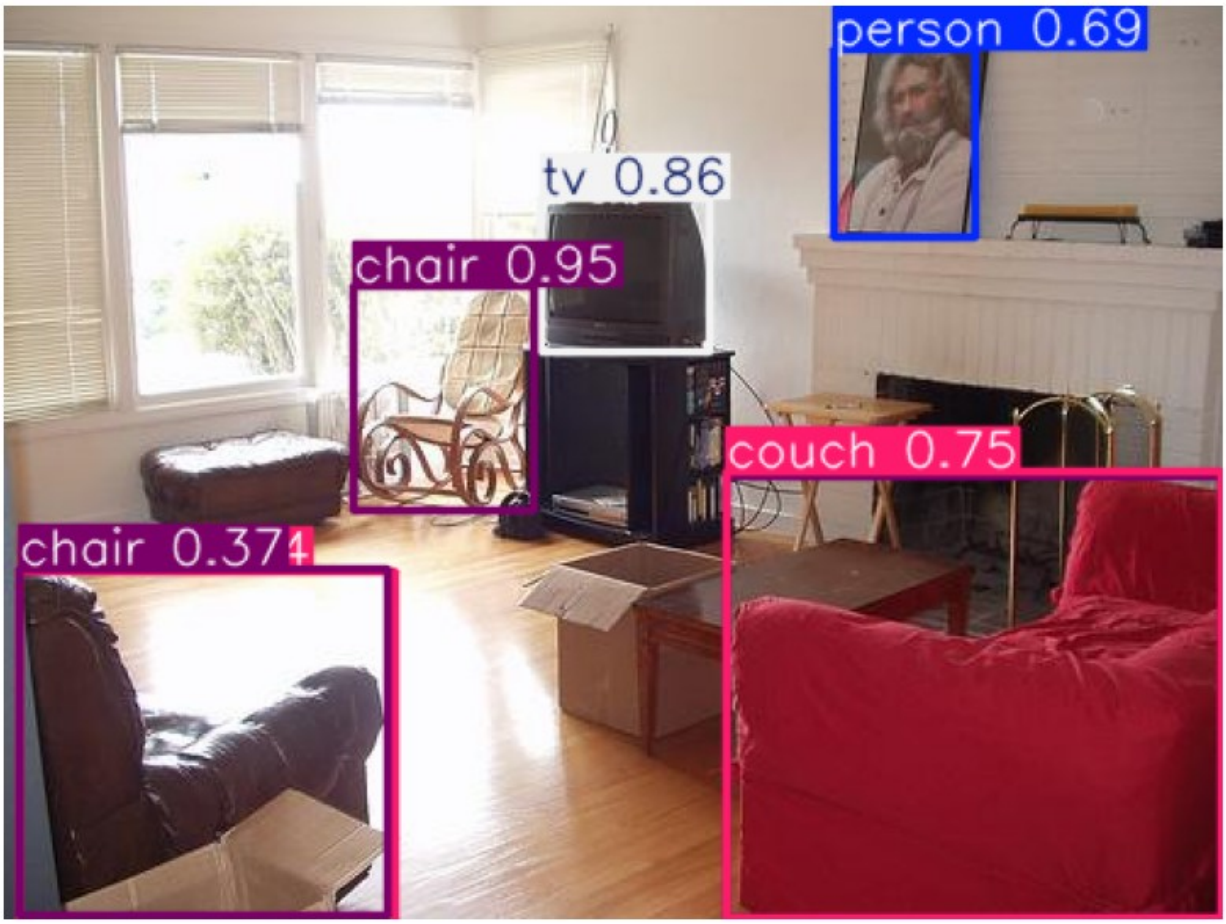



```
image 1/1 d:\2021\AIT\Master's\DL4CV\homework3\dataset\2.jpg: 640x480
5 persons, 1 handbag, 125.0ms
Speed: 3.0ms preprocess, 125.0ms inference, 3.0ms postprocess per
image at shape (1, 3, 640, 480)
```


```
image 1/1 d:\2021\AIT\Master's\DL4CV\homework3\dataset\3.jpg: 640x480
5 persons, 3 cars, 1 traffic light, 166.0ms
```

```
Speed: 3.0ms preprocess, 166.0ms inference, 5.0ms postprocess per
image at shape (1, 3, 640, 480)
```

```
image 1/1 d:\2021\AIT\Master's\DL4CV\homework3\dataset\4.jpg: 480x640
1 person, 3 chairs, 2 couchs, 1 tv, 147.1ms
Speed: 3.0ms preprocess, 147.1ms inference, 7.0ms postprocess per
image at shape (1, 3, 480, 640)
```

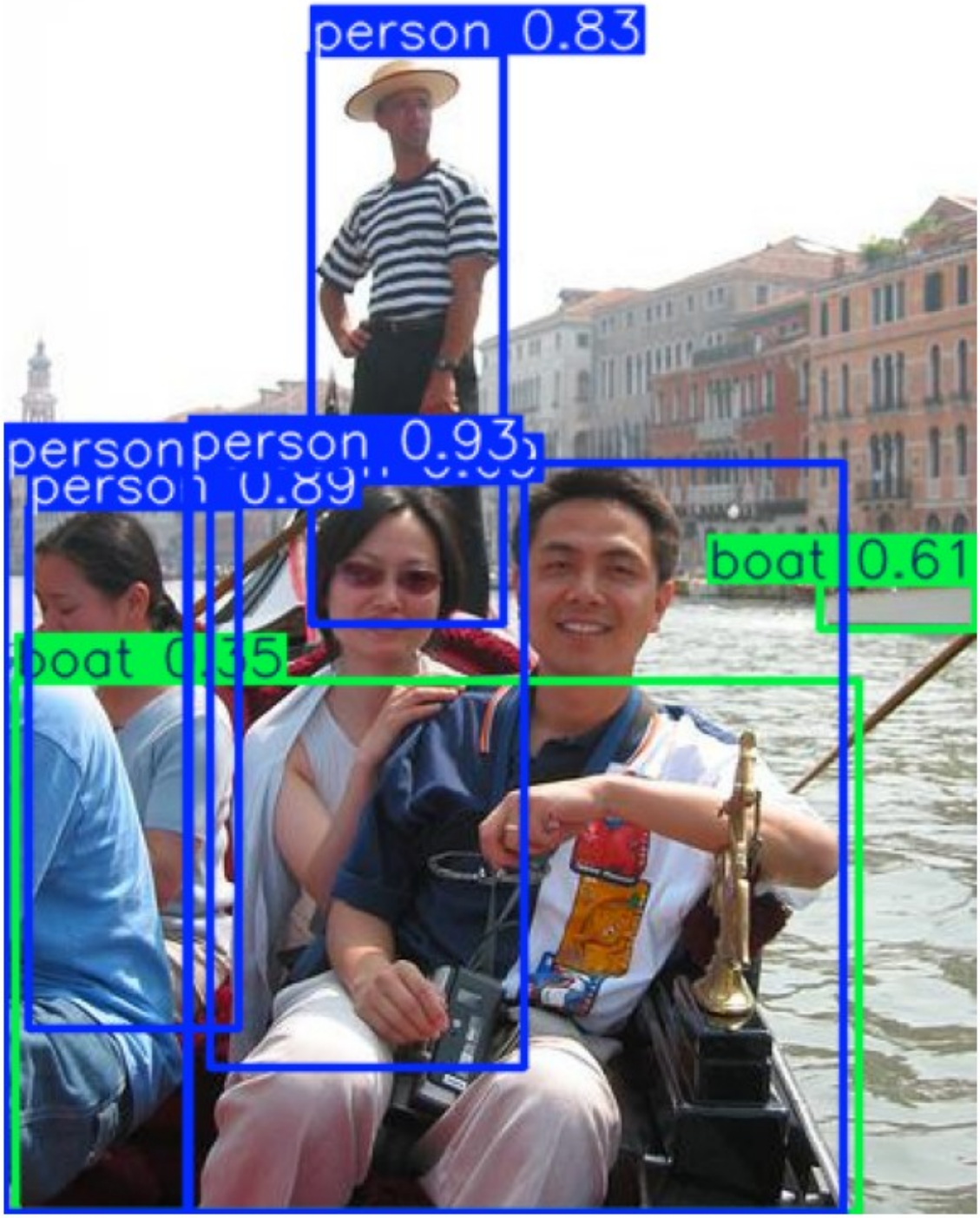



```
image 1/1 d:\2021\AIT\Master's\DL4CV\homework3\dataset\5.jpg: 416x640
6 persons, 3 traffic lights, 1 chair, 234.3ms
Speed: 3.0ms preprocess, 234.3ms inference, 2.0ms postprocess per
image at shape (1, 3, 416, 640)
```


```
img_list = os.listdir('dataset')

for img_name in img_list:
    img_path = os.path.join('dataset', img_name)

    results = model2(img_path, device='cpu')

    for result in results:
        rgb_image = result.plot()
        bgr_image = cv2.cvtColor(np.array(rgb_image),
cv2.COLOR_RGB2BGR)

        plt.figure(figsize=(10, 10))
        plt.imshow(bgr_image)
        plt.axis('off')
        plt.show()
```

```
image 1/1 d:\2021\AIT\Master's\DL4CV\homework3\dataset\1.jpg: 448x640
4 persons, 1 bottle, 2 cups, 2 cell phones, 1081.2ms
Speed: 3.0ms preprocess, 1081.2ms inference, 3.0ms postprocess per
image at shape (1, 3, 448, 640)
```

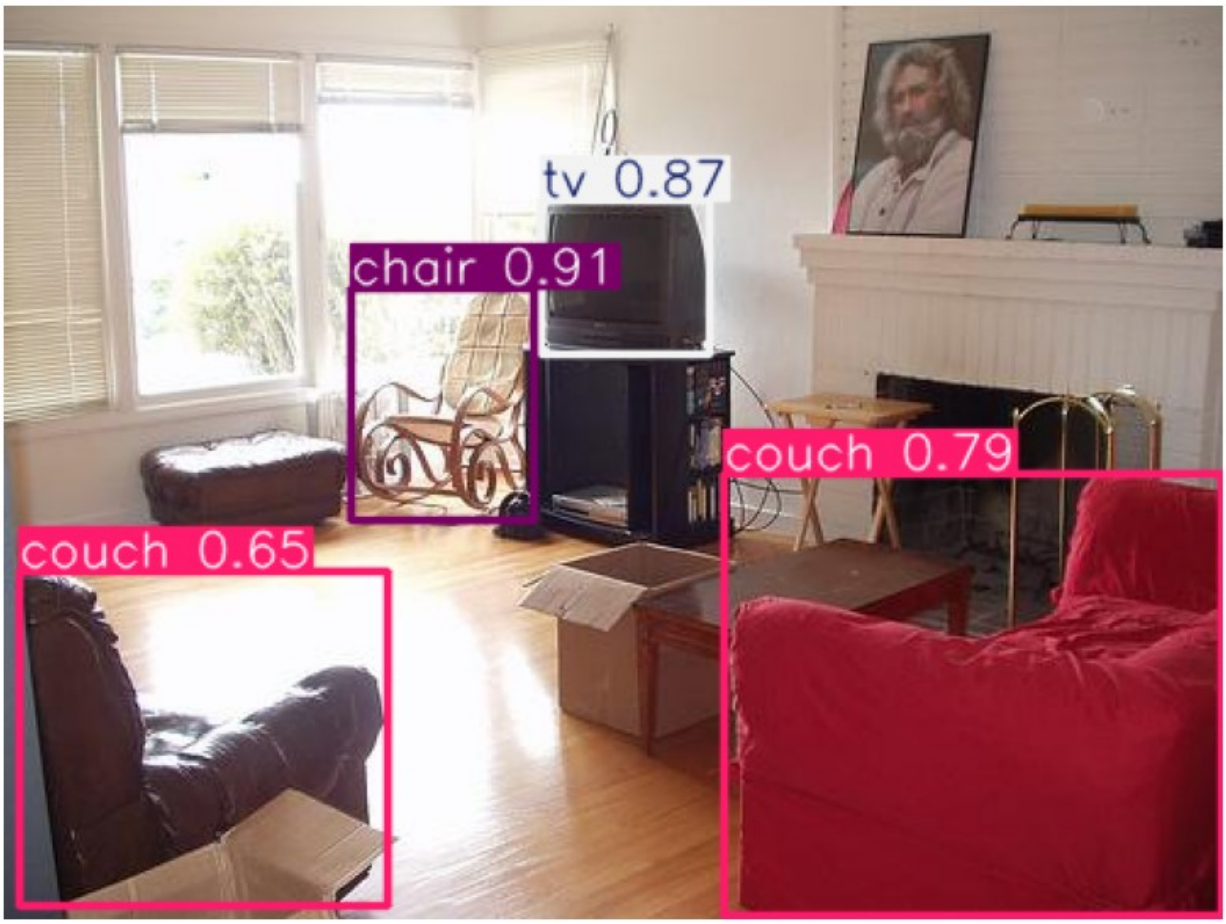

```
image 1/1 d:\2021\AIT\Master's\DL4CV\homework3\dataset\2.jpg: 640x480
5 persons, 2 boats, 1619.2ms
Speed: 6.0ms preprocess, 1619.2ms inference, 2.0ms postprocess per
image at shape (1, 3, 640, 480)
```



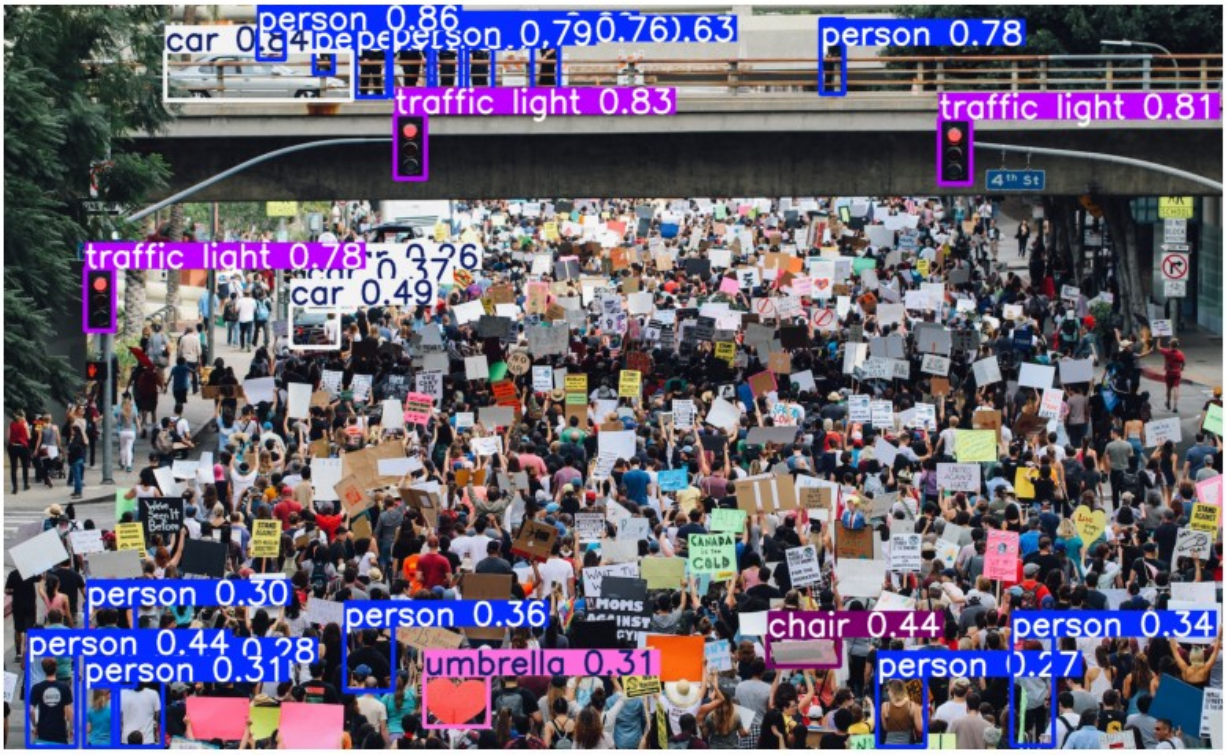

```
image 1/1 d:\2021\AIT\Master's\DL4CV\homework3\dataset\3.jpg: 640x480
6 persons, 3 cars, 1 bus, 1 traffic light, 1211.5ms
```

```
Speed: 3.0ms preprocess, 1211.5ms inference, 3.0ms postprocess per
image at shape (1, 3, 640, 480)
```

```
image 1/1 d:\2021\AIT\Master's\DL4CV\homework3\dataset\4.jpg: 480x640
1 chair, 2 couchs, 1 tv, 1415.4ms
Speed: 3.0ms preprocess, 1415.4ms inference, 3.0ms postprocess per
image at shape (1, 3, 480, 640)
```




```
image 1/1 d:\2021\AIT\Master's\DL4CV\homework3\dataset\5.jpg: 416x640
15 persons, 4 cars, 3 traffic lights, 1 umbrella, 1 chair, 1417.5ms
Speed: 6.9ms preprocess, 1417.5ms inference, 6.0ms postprocess per
image at shape (1, 3, 416, 640)
```



Conclusion

So, yolov8 extreme large model processes much more outputs, and it seems like it performs much better than nano model. The reason behind why it is happening so is the difference in the number of coefficients (the size of deel neural network, number of layers, weights, coefficients), it has more parameters, more anchor boxes. The nano model was developed to run inside the smart-cameras or iot devices with not so much processing power. The extreme large model requires much more calculations, FLOPs, and processing power.