Lab-03

Mrinal Bhan

211020428, DSAI

Tasks

Experiment No.01 - Write a program to perform Image Classification with localization, object detection, and image segmentation.

- Consider online images (one or more)with multiple objects.
- Apply for Image datasets.

Object Detection

In [2]:
```
from ultralytics import YOLO

# Load a model
model = YOLO('yolov8n.pt')  # load an official model

# Predict with the model
results = model('./test.png')  # predict on an image
```

```
image 1/1 c:\Users\Mrinal Bhan\College\Sem 5\CV\lab\Lab-3\test.png: 448x640 7 persons, 6 bicy
cles, 5 cars, 1 traffic light, 2 stop signs, 152.1ms
Speed: 3.0ms preprocess, 152.1ms inference, 4.0ms postprocess per image at shape (1, 3, 448,
640)
```

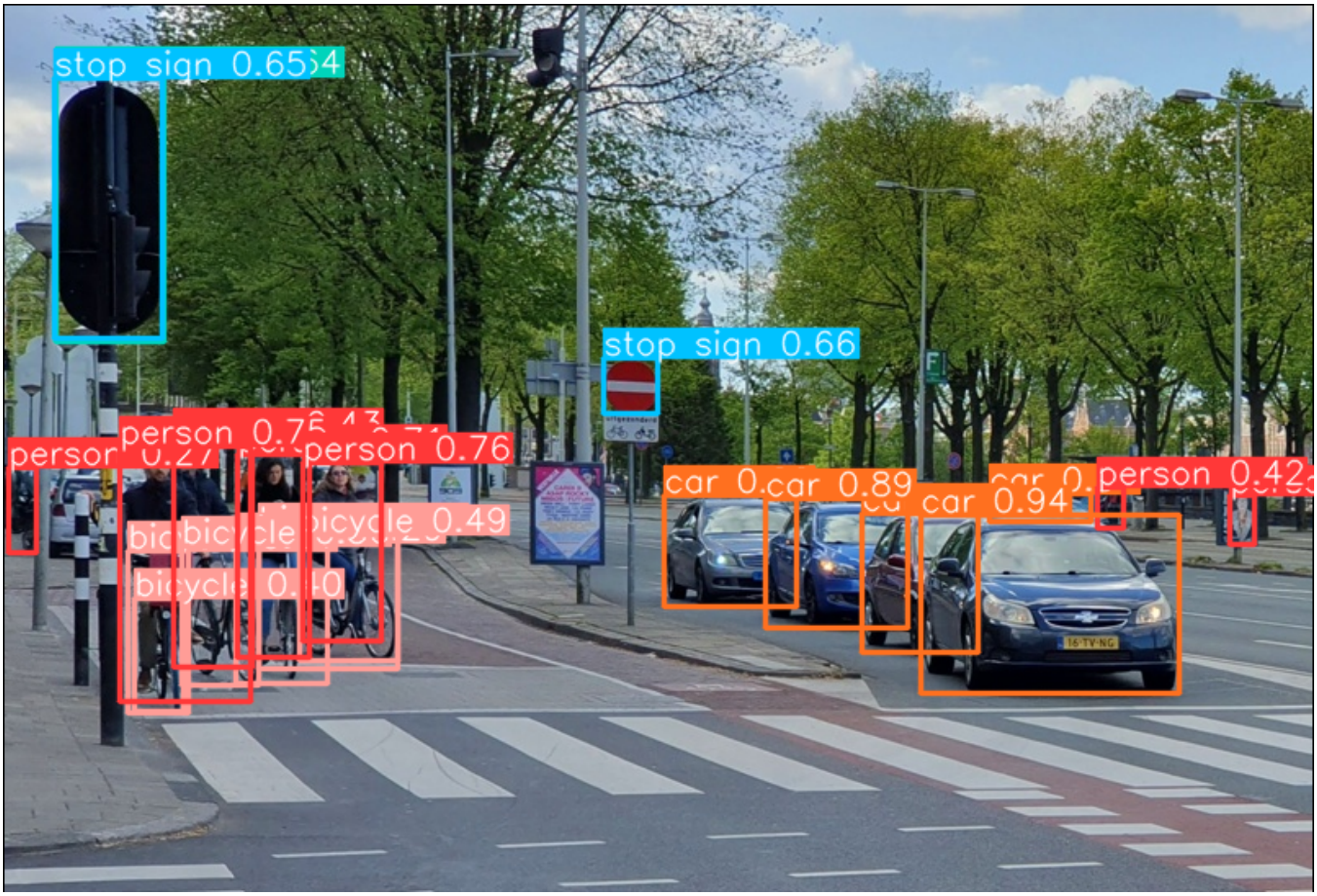
In [3]:
```
result = results[0]
print(len(result.boxes))

for box in result.boxes:
  class_id = result.names[box.cls[0].item()]
  cords = box.xyxy[0].tolist()
  cords = [round(x) for x in cords]
  conf = round(box.conf[0].item(), 2)
  print("Object type:", class_id)
  print("Coordinates:", cords)
  print("Probability:", conf)
  print("---")
```

```
21
Object type: car
Coordinates: [562, 313, 720, 422]
Probability: 0.94
---
Object type: car
Coordinates: [466, 303, 553, 381]
Probability: 0.89
---
Object type: car
Coordinates: [405, 300, 485, 369]
Probability: 0.82
---
Object type: car
Coordinates: [525, 311, 597, 397]
Probability: 0.79
---
Object type: person
Coordinates: [183, 281, 231, 390]
Probability: 0.76
---
Object type: person
Coordinates: [71, 270, 151, 426]
Probability: 0.75
---
Object type: person
Coordinates: [144, 276, 187, 400]
Probability: 0.71
---
Object type: stop sign
Coordinates: [367, 217, 401, 251]
Probability: 0.66
---
Object type: stop sign
Coordinates: [32, 45, 98, 205]
Probability: 0.65
---
Object type: traffic light
Coordinates: [33, 44, 98, 206]
Probability: 0.64
---
Object type: bicycle
Coordinates: [181, 324, 241, 402]
Probability: 0.49
---
Object type: person
Coordinates: [105, 266, 153, 406]
Probability: 0.43
---
Object type: person
Coordinates: [669, 295, 686, 321]
Probability: 0.42
---
Object type: bicycle
Coordinates: [80, 364, 111, 432]
Probability: 0.4
---
Object type: car
Coordinates: [603, 300, 666, 316]
Probability: 0.37
---
Object type: bicycle
Coordinates: [110, 331, 156, 417]
```

```
Probability: 0.36
---
Object type: person
Coordinates: [751, 301, 766, 331]
Probability: 0.32
---
Object type: bicycle
Coordinates: [77, 335, 113, 433]
Probability: 0.31
---
Object type: bicycle
Coordinates: [143, 330, 198, 415]
Probability: 0.29
---
Object type: bicycle
Coordinates: [157, 323, 241, 406]
Probability: 0.28
---
Object type: person
Coordinates: [4, 284, 21, 336]
Probability: 0.27
---
```

In [4]:
```
from PIL import Image
Image.fromarray(result.plot()[:,:,::-1])
```

Out[4]:



Object Segmentation

In [5]:
```
from ultralytics import YOLO

# Load a model
model = YOLO('yolov8n-seg.pt')  # load an official model
```

```
# Predict with the model
results = model('./test.png')  # predict on an image
```

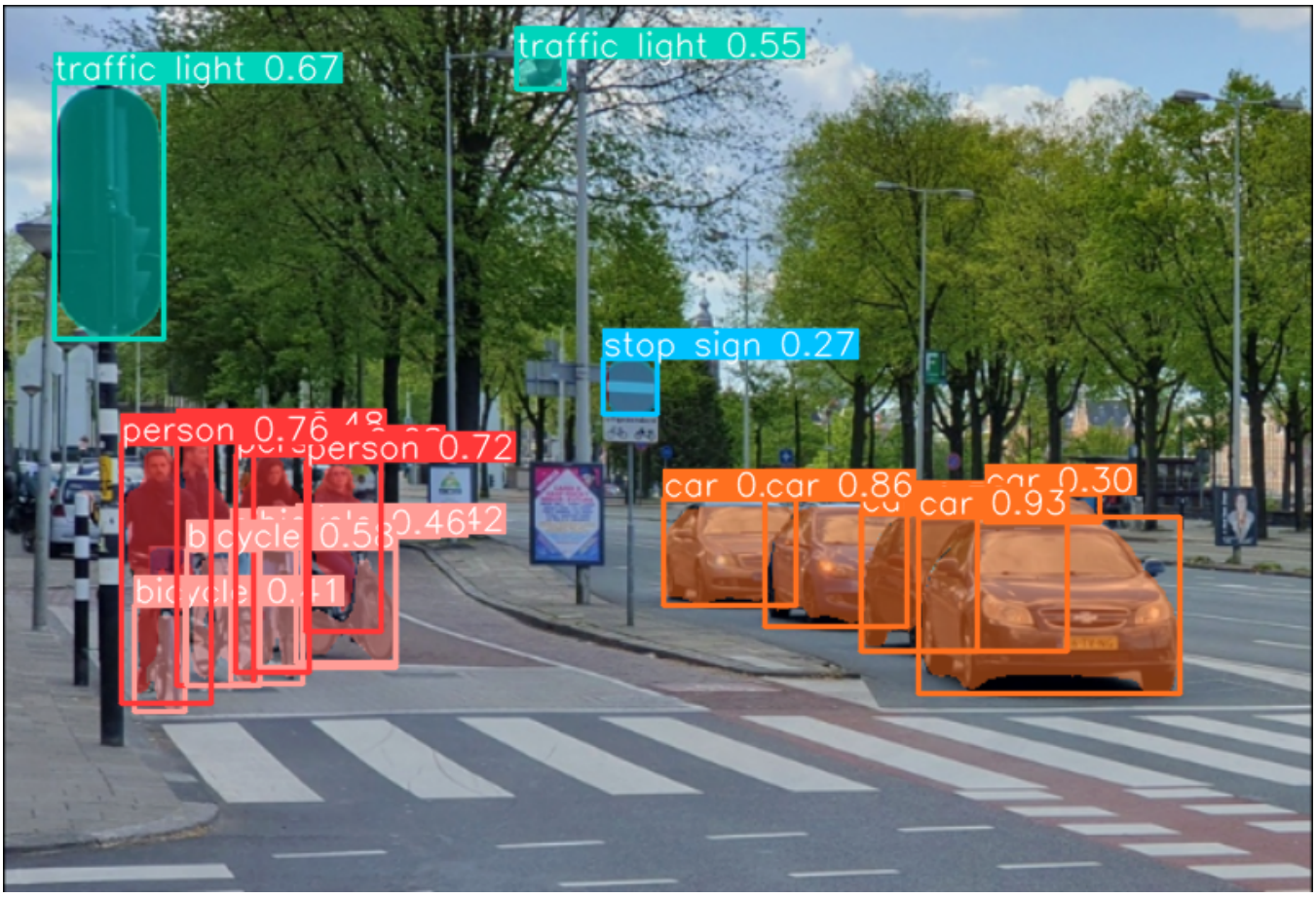
```
image 1/1 c:\Users\Mrinal Bhan\College\Sem 5\CV\lab\Lab-3\test.png: 448x640 4 persons, 5 bicycles, 6 cars, 2 traffic lights, 1 stop sign, 203.1ms
Speed: 4.0ms preprocess, 203.1ms inference, 29.1ms postprocess per image at shape (1, 3, 448, 640)
```

In [6]:
```
result = results[0]
print(len(result.boxes))
```

```
18
```

In [7]:
```
from PIL import Image
Image.fromarray(result.plot()[:,:,::-1])
```

Out[7]:



Object Classification

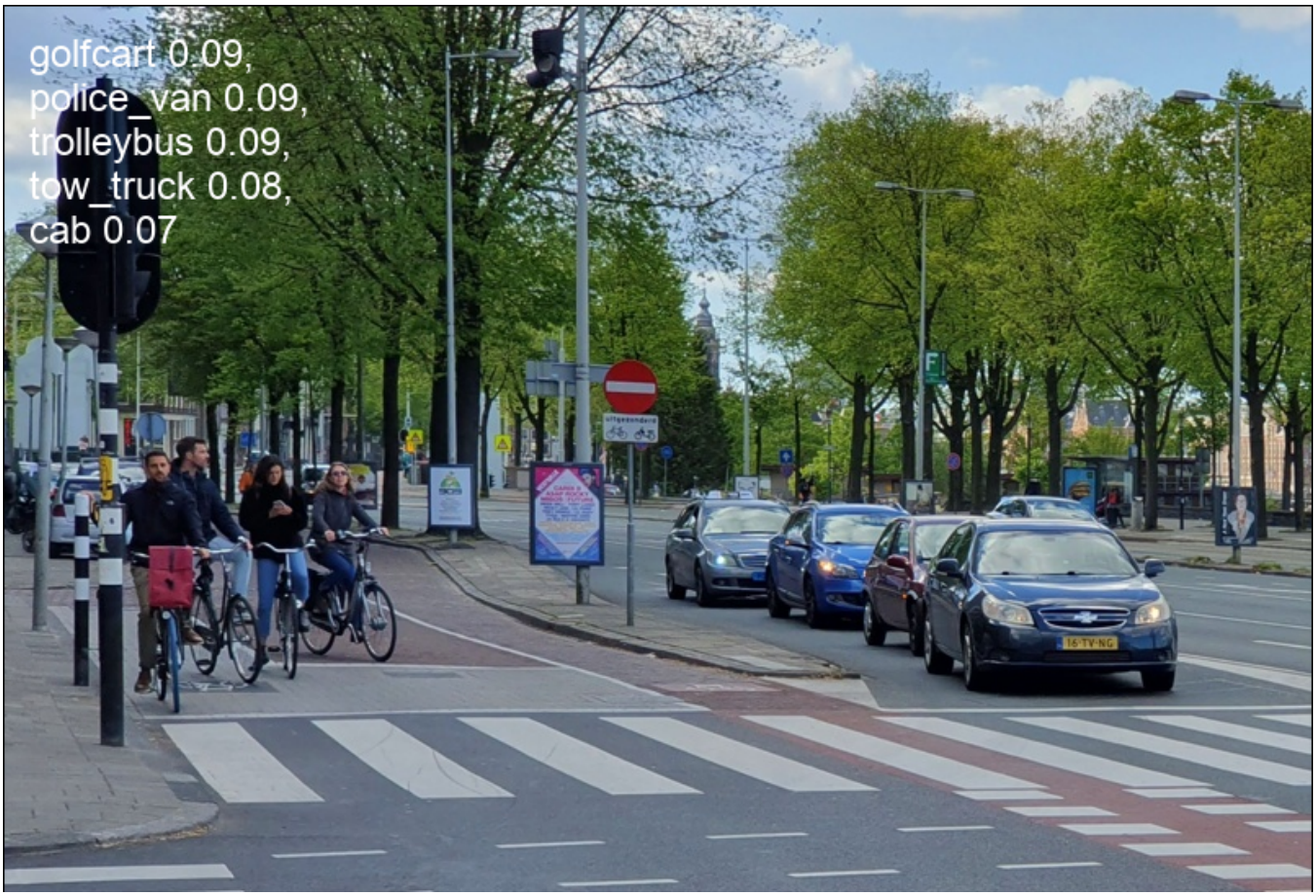
In [10]:
```
from ultralytics import YOLO

# Load a model
model = YOLO('yolov8n-cls.pt')  # load an official model

# Predict with the model
results = model('./test.png')  # predict on an image
```

```
image 1/1 c:\Users\Mrinal Bhan\College\Sem 5\CV\lab\Lab-3\test.png: 224x224 golfcart 0.09, police_van 0.09, trolleybus 0.09, tow_truck 0.08, cab 0.07, 34.0ms
Speed: 3.0ms preprocess, 34.0ms inference, 0.0ms postprocess per image at shape (1, 3, 224, 224)
```

In [11]:
```
result = results[0]
```

In [12]:
```
from PIL import Image
Image.fromarray(result.plot()[:,:,::-1])
```

Out[12]:


golfcart 0.09,
police_van 0.09,
trolleybus 0.09,
tow_truck 0.08,
cab 0.07

In [ ]: