```
# mount Google Drive
from google.colab import drive
drive.mount('/content/gdrive')
%cd gdrive/MyDrive
```

```
Mounted at /content/gdrive
/content/gdrive/MyDrive
```

```
#importing all the required libraries
import matplotlib.pyplot as plt
import numpy as np
import os
import random
import time
import torch
import cv2
```

```
#loading the images

list_img = os.listdir("yolov7/open_image")
```

```
len(list_img)
```

```
24
```

```
# Pip install method (recommended)
%pip install ultralytics
```

```
Looking in indexes: https://pypi.org/simple, https://us-
python.pkg.dev/colab-wheels/public/simple/
Collecting ultralytics
  Downloading ultralytics-8.0.18-py3-none-any.whl (259 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 259.3/259.3 KB 11.8 MB/s eta
0:00:00
ent already satisfied: requests>=2.23.0 in
/usr/local/lib/python3.8/dist-packages (from ultralytics) (2.25.1)
Requirement already satisfied: tqdm>=4.64.0 in
/usr/local/lib/python3.8/dist-packages (from ultralytics) (4.64.1)
Requirement already satisfied: opencv-python>=4.6.0 in
/usr/local/lib/python3.8/dist-packages (from ultralytics) (4.6.0.66)
Collecting thop>=0.1.1
  Downloading thop-0.1.1.post2209072238-py3-none-any.whl (15 kB)
Requirement already satisfied: Pillow>=7.1.2 in
/usr/local/lib/python3.8/dist-packages (from ultralytics) (7.1.2)
Requirement already satisfied: torch>=1.7.0 in
/usr/local/lib/python3.8/dist-packages (from ultralytics)
(1.13.1+cu116)
Requirement already satisfied: torchvision>=0.8.1 in
/usr/local/lib/python3.8/dist-packages (from ultralytics)
(0.14.1+cu116)
Requirement already satisfied: pandas>=1.1.4 in
```

```
/usr/local/lib/python3.8/dist-packages (from ultralytics) (1.3.5)
Collecting sentry-sdk
  Downloading sentry_sdk-1.14.0-py2.py3-none-any.whl (178 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 178.9/178.9 KB 19.0 MB/s eta
0:00:00
ent already satisfied: scipy>=1.4.1 in /usr/local/lib/python3.8/dist-
packages (from ultralytics) (1.7.3)
Requirement already satisfied: matplotlib>=3.2.2 in
/usr/local/lib/python3.8/dist-packages (from ultralytics) (3.2.2)
Requirement already satisfied: tensorboard>=2.4.1 in
/usr/local/lib/python3.8/dist-packages (from ultralytics) (2.9.1)
Requirement already satisfied: numpy>=1.18.5 in
/usr/local/lib/python3.8/dist-packages (from ultralytics) (1.21.6)
Requirement already satisfied: seaborn>=0.11.0 in
/usr/local/lib/python3.8/dist-packages (from ultralytics) (0.11.2)
Requirement already satisfied: ipython in
/usr/local/lib/python3.8/dist-packages (from ultralytics) (7.9.0)
Requirement already satisfied: psutil in
/usr/local/lib/python3.8/dist-packages (from ultralytics) (5.4.8)
Requirement already satisfied: PyYAML>=5.3.1 in
/usr/local/lib/python3.8/dist-packages (from ultralytics) (6.0)
Requirement already satisfied: cycler>=0.10 in
/usr/local/lib/python3.8/dist-packages (from matplotlib>=3.2.2-
>ultralytics) (0.11.0)
Requirement already satisfied: python-dateutil>=2.1 in
/usr/local/lib/python3.8/dist-packages (from matplotlib>=3.2.2-
>ultralytics) (2.8.2)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!
=2.1.6,>=2.0.1 in /usr/local/lib/python3.8/dist-packages (from
matplotlib>=3.2.2->ultralytics) (3.0.9)
Requirement already satisfied: kiwisolver>=1.0.1 in
/usr/local/lib/python3.8/dist-packages (from matplotlib>=3.2.2-
>ultralytics) (1.4.4)
Requirement already satisfied: pytz>=2017.3 in
/usr/local/lib/python3.8/dist-packages (from pandas>=1.1.4-
>ultralytics) (2022.7)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in
/usr/local/lib/python3.8/dist-packages (from requests>=2.23.0-
>ultralytics) (1.24.3)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.8/dist-packages (from requests>=2.23.0-
>ultralytics) (2022.12.7)
Requirement already satisfied: chardet<5,>=3.0.2 in
/usr/local/lib/python3.8/dist-packages (from requests>=2.23.0-
>ultralytics) (4.0.0)
Requirement already satisfied: idna<3,>=2.5 in
/usr/local/lib/python3.8/dist-packages (from requests>=2.23.0-
>ultralytics) (2.10)
Requirement already satisfied: setuptools>=41.0.0 in
/usr/local/lib/python3.8/dist-packages (from tensorboard>=2.4.1-
```

```
>ultralytics) (57.4.0)
Requirement already satisfied: grpcio>=1.24.3 in
/usr/local/lib/python3.8/dist-packages (from tensorboard>=2.4.1-
>ultralytics) (1.51.1)
Requirement already satisfied: tensorboard-plugin-wit>=1.6.0 in
/usr/local/lib/python3.8/dist-packages (from tensorboard>=2.4.1-
>ultralytics) (1.8.1)
Requirement already satisfied: protobuf<3.20,>=3.9.2 in
/usr/local/lib/python3.8/dist-packages (from tensorboard>=2.4.1-
>ultralytics) (3.19.6)
Requirement already satisfied: google-auth<3,>=1.6.3 in
/usr/local/lib/python3.8/dist-packages (from tensorboard>=2.4.1-
>ultralytics) (2.16.0)
Requirement already satisfied: werkzeug>=1.0.1 in
/usr/local/lib/python3.8/dist-packages (from tensorboard>=2.4.1-
>ultralytics) (1.0.1)
Requirement already satisfied: tensorboard-data-server<0.7.0,>=0.6.0
in /usr/local/lib/python3.8/dist-packages (from tensorboard>=2.4.1-
>ultralytics) (0.6.1)
Requirement already satisfied: wheel>=0.26 in
/usr/local/lib/python3.8/dist-packages (from tensorboard>=2.4.1-
>ultralytics) (0.38.4)
Requirement already satisfied: markdown>=2.6.8 in
/usr/local/lib/python3.8/dist-packages (from tensorboard>=2.4.1-
>ultralytics) (3.4.1)
Requirement already satisfied: google-auth-oauthlib<0.5,>=0.4.1 in
/usr/local/lib/python3.8/dist-packages (from tensorboard>=2.4.1-
>ultralytics) (0.4.6)
Requirement already satisfied: absl-py>=0.4 in
/usr/local/lib/python3.8/dist-packages (from tensorboard>=2.4.1-
>ultralytics) (1.3.0)
Requirement already satisfied: typing-extensions in
/usr/local/lib/python3.8/dist-packages (from torch>=1.7.0-
>ultralytics) (4.4.0)
Requirement already satisfied: pygments in
/usr/local/lib/python3.8/dist-packages (from ipython->ultralytics)
(2.6.1)
Collecting jedi>=0.10
  Downloading jedi-0.18.2-py2.py3-none-any.whl (1.6 MB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.6/1.6 MB 59.6 MB/s eta
0:00:00
ent already satisfied: pickleshare in /usr/local/lib/python3.8/dist-
packages (from ipython->ultralytics) (0.7.5)
Requirement already satisfied: pexpect in
/usr/local/lib/python3.8/dist-packages (from ipython->ultralytics)
(4.8.0)
Requirement already satisfied: traitlets>=4.2 in
/usr/local/lib/python3.8/dist-packages (from ipython->ultralytics)
(5.7.1)
Requirement already satisfied: decorator in
```

```
/usr/local/lib/python3.8/dist-packages (from ipython->ultralytics)
(4.4.2)
Requirement already satisfied: prompt-toolkit<2.1.0,>=2.0.0 in
/usr/local/lib/python3.8/dist-packages (from ipython->ultralytics)
(2.0.10)
Requirement already satisfied: backcall in
/usr/local/lib/python3.8/dist-packages (from ipython->ultralytics)
(0.2.0)
Collecting urllib3<1.27,>=1.21.1
  Downloading urllib3-1.26.14-py2.py3-none-any.whl (140 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 140.6/140.6 KB 16.6 MB/s eta
0:00:00
ent already satisfied: cachetools<6.0,>=2.0.0 in
/usr/local/lib/python3.8/dist-packages (from google-auth<3,>=1.6.3-
>tensorboard>=2.4.1->ultralytics) (5.2.1)
Requirement already satisfied: six>=1.9.0 in
/usr/local/lib/python3.8/dist-packages (from google-auth<3,>=1.6.3-
>tensorboard>=2.4.1->ultralytics) (1.15.0)
Requirement already satisfied: rsa<5,>=3.1.4 in
/usr/local/lib/python3.8/dist-packages (from google-auth<3,>=1.6.3-
>tensorboard>=2.4.1->ultralytics) (4.9)
Requirement already satisfied: pyasn1-modules>=0.2.1 in
/usr/local/lib/python3.8/dist-packages (from google-auth<3,>=1.6.3-
>tensorboard>=2.4.1->ultralytics) (0.2.8)
Requirement already satisfied: requests-oauthlib>=0.7.0 in
/usr/local/lib/python3.8/dist-packages (from google-auth-
oauthlib<0.5,>=0.4.1->tensorboard>=2.4.1->ultralytics) (1.3.1)
Requirement already satisfied: parso<0.9.0,>=0.8.0 in
/usr/local/lib/python3.8/dist-packages (from jedi>=0.10->ipython-
>ultralytics) (0.8.3)
Requirement already satisfied: importlib-metadata>=4.4 in
/usr/local/lib/python3.8/dist-packages (from markdown>=2.6.8-
>tensorboard>=2.4.1->ultralytics) (6.0.0)
Requirement already satisfied: wcwidth in
/usr/local/lib/python3.8/dist-packages (from prompt-
toolkit<2.1.0,>=2.0.0->ipython->ultralytics) (0.2.5)
Requirement already satisfied: ptyprocess>=0.5 in
/usr/local/lib/python3.8/dist-packages (from pexpect->ipython-
>ultralytics) (0.7.0)
Requirement already satisfied: zipp>=0.5 in
/usr/local/lib/python3.8/dist-packages (from importlib-metadata>=4.4-
>markdown>=2.6.8->tensorboard>=2.4.1->ultralytics) (3.11.0)
Requirement already satisfied: pyasn1<0.5.0,>=0.4.6 in
/usr/local/lib/python3.8/dist-packages (from pyasn1-modules>=0.2.1-
>google-auth<3,>=1.6.3->tensorboard>=2.4.1->ultralytics) (0.4.8)
Requirement already satisfied: oauthlib>=3.0.0 in
/usr/local/lib/python3.8/dist-packages (from requests-oauthlib>=0.7.0-
>google-auth-oauthlib<0.5,>=0.4.1->tensorboard>=2.4.1->ultralytics)
(3.2.2)
Installing collected packages: urllib3, jedi, thop, sentry-sdk,
```

```
ultralytics
  Attempting uninstall: urllib3
    Found existing installation: urllib3 1.24.3
    Uninstalling urllib3-1.24.3:
      Successfully uninstalled urllib3-1.24.3
Successfully installed jedi-0.18.2 sentry-sdk-1.14.0 thop-
0.1.1.post2209072238 ultralytics-8.0.18 urllib3-1.26.14
```

```
from ultralytics import YOLO

modelv8 = YOLO("yolov8n.pt")

for i in range(len(list_img)):
  list_img[i]='yolov7/open_image/'+ list_img[i]

np_img = []
for i in range(len(list_img)):
  np_img.append(cv2.imread(list_img[i]))

np_img_copy = np_img[:]

%%time
results = modelv8.predict(np_img)
```

```
Ultralytics YOLOv8.0.18 🚀 Python-3.8.10 torch-1.13.1+cu116 CPU
YOLOv8n summary (fused): 168 layers, 3151904 parameters, 0 gradients,
8.7 GFLOPs

CPU times: user 6.17 s, sys: 484 ms, total: 6.65 s
Wall time: 7.52 s
```

```
type(results)
```

```
list
```

```
results
```

```
[Ultralytics YOLO <class 'ultralytics.yolo.engine.results.Boxes'>
masks
 type: <class 'torch.Tensor'>
 shape: torch.Size([7, 6])
 dtype: torch.float32
  + tensor([[8.75000e+02, 2.04000e+02, 9.70000e+02, 5.42000e+02,
8.37272e-01, 0.00000e+00],
         [4.59000e+02, 2.68000e+02, 5.47000e+02, 5.87000e+02,
8.32995e-01, 0.00000e+00],
         [7.59000e+02, 2.52000e+02, 8.82000e+02, 6.07000e+02,
8.09783e-01, 0.00000e+00],
         [0.00000e+00, 8.30000e+01, 9.80000e+02, 5.99000e+02,
6.86175e-01, 7.00000e+00],
         [5.83000e+02, 2.96000e+02, 6.96000e+02, 4.08000e+02,
6.86045e-01, 0.00000e+00],
```

```
         [6.21000e+02, 4.49000e+02, 6.99000e+02, 5.54000e+02,
5.38134e-01, 2.40000e+01],
         [0.00000e+00, 8.30000e+01, 6.28000e+02, 5.95000e+02,
5.23223e-01, 7.00000e+00]]),
 Ultralytics YOLO <class 'ultralytics.yolo.engine.results.Boxes'>
masks
 type: <class 'torch.Tensor'>
 shape: torch.Size([25, 6])
 dtype: torch.float32
  + tensor([[5.70000e+02, 4.09000e+02, 8.30000e+02, 6.78000e+02,
9.01913e-01, 0.00000e+00],
         [4.90000e+01, 3.77000e+02, 3.30000e+02, 6.75000e+02,
7.45638e-01, 0.00000e+00],
         [4.00000e+00, 3.37000e+02, 1.80000e+02, 6.70000e+02,
7.36081e-01, 0.00000e+00],
         [4.18000e+02, 3.21000e+02, 5.10000e+02, 4.49000e+02,
6.84721e-01, 0.00000e+00],
         [4.81000e+02, 2.25000e+02, 5.25000e+02, 3.64000e+02,
6.76758e-01, 0.00000e+00],
         [2.26000e+02, 4.15000e+02, 5.33000e+02, 6.77000e+02,
6.47860e-01, 0.00000e+00],
         [1.50000e+02, 3.73000e+02, 3.35000e+02, 6.32000e+02,
5.93642e-01, 0.00000e+00],
         [4.65000e+02, 3.24000e+02, 6.21000e+02, 5.74000e+02,
5.92337e-01, 0.00000e+00],
         [3.41000e+02, 3.13000e+02, 4.38000e+02, 4.21000e+02,
5.47132e-01, 0.00000e+00],
         [6.46000e+02, 3.37000e+02, 7.76000e+02, 5.04000e+02,
5.13334e-01, 0.00000e+00],
         [1.05000e+02, 2.84000e+02, 1.52000e+02, 3.38000e+02,
5.10180e-01, 0.00000e+00],
         [2.25000e+02, 3.04000e+02, 3.14000e+02, 4.08000e+02,
4.31297e-01, 0.00000e+00],
         [1.64000e+02, 2.84000e+02, 2.13000e+02, 3.38000e+02,
3.53158e-01, 0.00000e+00],
         [0.00000e+00, 3.39000e+02, 7.30000e+01, 4.91000e+02,
3.35743e-01, 0.00000e+00],
         [6.28000e+02, 3.25000e+02, 7.49000e+02, 4.40000e+02,
3.28104e-01, 0.00000e+00],
         [6.92000e+02, 3.01000e+02, 7.32000e+02, 3.44000e+02,
3.27756e-01, 0.00000e+00],
         [5.42000e+02, 3.76000e+02, 5.96000e+02, 4.46000e+02,
3.18715e-01, 4.60000e+01],
         [2.22000e+02, 3.05000e+02, 3.15000e+02, 4.90000e+02,
3.13540e-01, 0.00000e+00],
         [0.00000e+00, 3.36000e+02, 7.30000e+01, 6.35000e+02,
3.08696e-01, 0.00000e+00],
         [8.07000e+02, 3.00000e+02, 8.76000e+02, 3.95000e+02,
2.92066e-01, 6.90000e+01],
         [3.04000e+02, 4.05000e+02, 3.90000e+02, 5.18000e+02,
```

```
2.84444e-01, 5.60000e+01],
         [2.10000e+01, 3.36000e+02, 1.81000e+02, 5.49000e+02,
2.84159e-01, 0.00000e+00],
         [5.07000e+02, 3.25000e+02, 6.19000e+02, 4.46000e+02,
2.81118e-01, 0.00000e+00],
         [6.28000e+02, 3.27000e+02, 7.70000e+02, 4.69000e+02,
2.73176e-01, 0.00000e+00],
         [2.07000e+02, 2.80000e+02, 2.53000e+02, 3.25000e+02,
2.56013e-01, 0.00000e+00]]),
 Ultralytics YOLO <class 'ultralytics.yolo.engine.results.Boxes'>
masks
 type: <class 'torch.Tensor'>
 shape: torch.Size([8, 6])
 dtype: torch.float32
  + tensor([[2.10000e+01, 1.45000e+02, 1.70000e+02, 5.39000e+02,
8.98457e-01, 0.00000e+00],
         [8.05000e+02, 4.71000e+02, 1.01500e+03, 5.69000e+02,
6.75249e-01, 6.30000e+01],
         [6.42000e+02, 4.57000e+02, 8.02000e+02, 5.48000e+02,
4.80973e-01, 6.30000e+01],
         [6.38000e+02, 3.64000e+02, 7.11000e+02, 4.49000e+02,
4.77708e-01, 0.00000e+00],
         [9.10000e+02, 3.11000e+02, 1.02300e+03, 4.59000e+02,
4.45932e-01, 0.00000e+00],
         [8.25000e+02, 3.24000e+02, 9.14000e+02, 4.45000e+02,
3.67154e-01, 0.00000e+00],
         [7.16000e+02, 3.47000e+02, 7.72000e+02, 4.49000e+02,
2.71804e-01, 0.00000e+00],
         [2.00000e+02, 4.15000e+02, 7.06000e+02, 5.53000e+02,
2.52012e-01, 0.00000e+00]]),
 Ultralytics YOLO <class 'ultralytics.yolo.engine.results.Boxes'>
masks
 type: <class 'torch.Tensor'>
 shape: torch.Size([3, 6])
 dtype: torch.float32
  + tensor([[2.15000e+02, 5.00000e+01, 7.91000e+02, 7.01000e+02,
8.56826e-01, 0.00000e+00],
         [2.13000e+02, 5.44000e+02, 4.97000e+02, 7.00000e+02,
7.88244e-01, 5.60000e+01],
         [5.18000e+02, 2.92000e+02, 5.92000e+02, 3.71000e+02,
2.70522e-01, 6.70000e+01]]),
 Ultralytics YOLO <class 'ultralytics.yolo.engine.results.Boxes'>
masks
 type: <class 'torch.Tensor'>
 shape: torch.Size([1, 6])
 dtype: torch.float32
  + tensor([[113.00000, 216.00000, 896.00000, 584.00000,   0.92341,
7.00000]]),
 Ultralytics YOLO <class 'ultralytics.yolo.engine.results.Boxes'>
masks
```

```
 type: <class 'torch.Tensor'>
 shape: torch.Size([0, 6])
 dtype: torch.float32
  + tensor([], size=(0, 6)),
 Ultralytics YOLO <class 'ultralytics.yolo.engine.results.Boxes'>
masks
 type: <class 'torch.Tensor'>
 shape: torch.Size([0, 6])
 dtype: torch.float32
  + tensor([], size=(0, 6)),
 Ultralytics YOLO <class 'ultralytics.yolo.engine.results.Boxes'>
masks
 type: <class 'torch.Tensor'>
 shape: torch.Size([0, 6])
 dtype: torch.float32
  + tensor([], size=(0, 6)),
 Ultralytics YOLO <class 'ultralytics.yolo.engine.results.Boxes'>
masks
 type: <class 'torch.Tensor'>
 shape: torch.Size([0, 6])
 dtype: torch.float32
  + tensor([], size=(0, 6)),
 Ultralytics YOLO <class 'ultralytics.yolo.engine.results.Boxes'>
masks
 type: <class 'torch.Tensor'>
 shape: torch.Size([3, 6])
 dtype: torch.float32
  + tensor([[3.40000e+01, 3.05000e+02, 5.95000e+02, 9.98000e+02,
7.39602e-01, 0.00000e+00],
         [2.47000e+02, 4.98000e+02, 2.96000e+02, 6.36000e+02,
7.34110e-01, 2.70000e+01],
         [1.33000e+02, 8.59000e+02, 3.83000e+02, 1.02400e+03,
4.06956e-01, 0.00000e+00]]),
 Ultralytics YOLO <class 'ultralytics.yolo.engine.results.Boxes'>
masks
 type: <class 'torch.Tensor'>
 shape: torch.Size([2, 6])
 dtype: torch.float32
  + tensor([[2.42000e+02, 3.43000e+02, 3.82000e+02, 4.97000e+02,
3.91732e-01, 6.20000e+01],
         [8.20000e+01, 1.97000e+02, 3.13000e+02, 6.70000e+02,
2.99582e-01, 0.00000e+00]]),
 Ultralytics YOLO <class 'ultralytics.yolo.engine.results.Boxes'>
masks
 type: <class 'torch.Tensor'>
 shape: torch.Size([1, 6])
 dtype: torch.float32
  + tensor([[234.00000,  29.00000, 833.00000, 761.00000,   0.85220,
0.00000]]),
 Ultralytics YOLO <class 'ultralytics.yolo.engine.results.Boxes'>
```

```
masks
 type: <class 'torch.Tensor'>
 shape: torch.Size([0, 6])
 dtype: torch.float32
  + tensor([], size=(0, 6)),
 Ultralytics YOLO <class 'ultralytics.yolo.engine.results.Boxes'>
masks
 type: <class 'torch.Tensor'>
 shape: torch.Size([15, 6])
 dtype: torch.float32
  + tensor([[5.56000e+02, 2.00000e+02, 7.77000e+02, 5.72000e+02,
9.02662e-01, 0.00000e+00],
         [0.00000e+00, 5.00000e+00, 1.34000e+02, 5.67000e+02,
8.70928e-01, 0.00000e+00],
         [7.70000e+01, 1.42000e+02, 2.75000e+02, 5.70000e+02,
8.47831e-01, 0.00000e+00],
         [4.17000e+02, 2.62000e+02, 4.88000e+02, 4.08000e+02,
7.45451e-01, 0.00000e+00],
         [4.80000e+02, 2.73000e+02, 5.58000e+02, 4.10000e+02,
7.07003e-01, 0.00000e+00],
         [3.68000e+02, 2.92000e+02, 4.27000e+02, 4.10000e+02,
4.91883e-01, 0.00000e+00],
         [7.76000e+02, 2.12000e+02, 9.43000e+02, 4.95000e+02,
4.35982e-01, 1.20000e+01],
         [1.64000e+02, 4.31000e+02, 4.47000e+02, 5.71000e+02,
4.10349e-01, 2.80000e+01],
         [2.99000e+02, 2.95000e+02, 3.57000e+02, 4.06000e+02,
4.04060e-01, 0.00000e+00],
         [1.98000e+02, 4.23000e+02, 4.12000e+02, 4.96000e+02,
3.62418e-01, 2.80000e+01],
         [5.52000e+02, 2.96000e+02, 5.98000e+02, 3.57000e+02,
3.53927e-01, 0.00000e+00],
         [4.44000e+02, 4.76000e+02, 5.94000e+02, 5.75000e+02,
3.53062e-01, 2.80000e+01],
         [7.03000e+02, 2.97000e+02, 7.39000e+02, 3.32000e+02,
2.92022e-01, 0.00000e+00],
         [5.77000e+02, 2.83000e+02, 6.40000e+02, 3.56000e+02,
2.87269e-01, 0.00000e+00],
         [5.49000e+02, 3.58000e+02, 5.78000e+02, 4.14000e+02,
2.74343e-01, 4.10000e+01]]),
 Ultralytics YOLO <class 'ultralytics.yolo.engine.results.Boxes'>
masks
 type: <class 'torch.Tensor'>
 shape: torch.Size([1, 6])
 dtype: torch.float32
  + tensor([[5.50000e+01, 6.73000e+02, 4.30000e+02, 1.02400e+03,
2.51997e-01, 5.30000e+01]]),
 Ultralytics YOLO <class 'ultralytics.yolo.engine.results.Boxes'>
masks
 type: <class 'torch.Tensor'>
```

```
 shape: torch.Size([1, 6])
 dtype: torch.float32
  + tensor([[3.82000e+02, 3.06000e+02, 1.02200e+03, 7.61000e+02,
5.25129e-01, 6.00000e+00]]),
 Ultralytics YOLO <class 'ultralytics.yolo.engine.results.Boxes'>
masks
 type: <class 'torch.Tensor'>
 shape: torch.Size([3, 6])
 dtype: torch.float32
  + tensor([[0.00000e+00, 1.07000e+02, 3.90000e+02, 1.02400e+03,
9.18839e-01, 0.00000e+00],
         [2.83000e+02, 1.86000e+02, 5.75000e+02, 1.02200e+03,
7.62341e-01, 0.00000e+00],
         [0.00000e+00, 3.50000e+02, 3.10000e+01, 4.11000e+02,
5.37041e-01, 0.00000e+00]]),
 Ultralytics YOLO <class 'ultralytics.yolo.engine.results.Boxes'>
masks
 type: <class 'torch.Tensor'>
 shape: torch.Size([19, 6])
 dtype: torch.float32
  + tensor([[3.53000e+02, 1.32000e+02, 6.20000e+02, 4.23000e+02,
8.38578e-01, 0.00000e+00],
         [4.97000e+02, 3.68000e+02, 5.34000e+02, 4.54000e+02,
8.29666e-01, 3.90000e+01],
         [5.05000e+02, 1.98000e+02, 1.02300e+03, 6.75000e+02,
8.12024e-01, 0.00000e+00],
         [0.00000e+00, 2.41000e+02, 4.10000e+02, 6.69000e+02,
7.85713e-01, 0.00000e+00],
         [0.00000e+00, 5.99000e+02, 3.41000e+02, 6.77000e+02,
6.58886e-01, 5.60000e+01],
         [2.08000e+02, 1.87000e+02, 3.84000e+02, 4.29000e+02,
6.00069e-01, 0.00000e+00],
         [4.49000e+02, 6.10000e+01, 5.11000e+02, 1.31000e+02,
5.57872e-01, 0.00000e+00],
         [8.99000e+02, 1.52000e+02, 1.02300e+03, 3.00000e+02,
5.33262e-01, 0.00000e+00],
         [2.98000e+02, 3.20000e+02, 3.44000e+02, 4.51000e+02,
5.27044e-01, 3.90000e+01],
         [4.10000e+02, 3.60000e+01, 4.54000e+02, 8.50000e+01,
4.90813e-01, 7.40000e+01],
         [2.74000e+02, 5.40000e+01, 3.63000e+02, 1.32000e+02,
4.60952e-01, 0.00000e+00],
         [3.04000e+02, 1.41000e+02, 4.43000e+02, 2.91000e+02,
4.21267e-01, 0.00000e+00],
         [5.38000e+02, 4.20000e+01, 5.98000e+02, 1.37000e+02,
4.19096e-01, 0.00000e+00],
         [5.37000e+02, 4.30000e+01, 6.33000e+02, 1.37000e+02,
3.63729e-01, 0.00000e+00],
         [2.07000e+02, 1.86000e+02, 4.45000e+02, 4.63000e+02,
3.49924e-01, 0.00000e+00],
```

```
         [9.71000e+02, 2.75000e+02, 9.96000e+02, 3.41000e+02,
3.32680e-01, 3.90000e+01],
         [3.37000e+02, 1.42000e+02, 4.44000e+02, 2.66000e+02,
3.28238e-01, 0.00000e+00],
         [8.10000e+02, 3.65000e+02, 9.49000e+02, 4.19000e+02,
2.98776e-01, 5.60000e+01],
         [3.46000e+02, 4.27000e+02, 6.16000e+02, 5.47000e+02,
2.72791e-01, 6.00000e+01]]),
 Ultralytics YOLO <class 'ultralytics.yolo.engine.results.Boxes'>
masks
 type: <class 'torch.Tensor'>
 shape: torch.Size([9, 6])
 dtype: torch.float32
  + tensor([[7.92000e+02, 3.40000e+02, 9.93000e+02, 7.16000e+02,
9.07119e-01, 0.00000e+00],
         [6.03000e+02, 3.43000e+02, 8.00000e+02, 7.16000e+02,
8.87946e-01, 0.00000e+00],
         [3.80000e+02, 3.65000e+02, 5.89000e+02, 7.16000e+02,
8.82033e-01, 0.00000e+00],
         [3.90000e+01, 3.22000e+02, 2.72000e+02, 7.15000e+02,
8.77361e-01, 0.00000e+00],
         [2.32000e+02, 3.87000e+02, 3.93000e+02, 7.16000e+02,
8.67137e-01, 0.00000e+00],
         [6.62000e+02, 4.51000e+02, 6.94000e+02, 5.74000e+02,
7.64628e-01, 2.70000e+01],
         [3.67000e+02, 3.59000e+02, 4.46000e+02, 5.07000e+02,
7.61771e-01, 0.00000e+00],
         [3.43000e+02, 4.91000e+02, 3.81000e+02, 5.79000e+02,
4.69785e-01, 2.70000e+01],
         [8.14000e+02, 4.48000e+02, 8.47000e+02, 6.92000e+02,
2.69333e-01, 2.70000e+01]]),
 Ultralytics YOLO <class 'ultralytics.yolo.engine.results.Boxes'>
masks
 type: <class 'torch.Tensor'>
 shape: torch.Size([0, 6])
 dtype: torch.float32
  + tensor([], size=(0, 6)),
 Ultralytics YOLO <class 'ultralytics.yolo.engine.results.Boxes'>
masks
 type: <class 'torch.Tensor'>
 shape: torch.Size([0, 6])
 dtype: torch.float32
  + tensor([], size=(0, 6)),
 Ultralytics YOLO <class 'ultralytics.yolo.engine.results.Boxes'>
masks
 type: <class 'torch.Tensor'>
 shape: torch.Size([20, 6])
 dtype: torch.float32
  + tensor([[6.78000e+02, 4.00000e+00, 1.02300e+03, 6.69000e+02,
9.09320e-01, 0.00000e+00],
```

```
         [3.79000e+02, 2.56000e+02, 5.92000e+02, 5.86000e+02,
8.01535e-01, 0.00000e+00],
         [4.73000e+02, 1.52000e+02, 5.27000e+02, 2.69000e+02,
7.55963e-01, 0.00000e+00],
         [6.31000e+02, 1.35000e+02, 6.99000e+02, 3.33000e+02,
7.55742e-01, 0.00000e+00],
         [1.54000e+02, 1.78000e+02, 1.90000e+02, 2.94000e+02,
6.56878e-01, 0.00000e+00],
         [1.90000e+01, 1.56000e+02, 7.40000e+01, 3.68000e+02,
6.33197e-01, 0.00000e+00],
         [3.84000e+02, 1.38000e+02, 4.46000e+02, 3.16000e+02,
6.28481e-01, 0.00000e+00],
         [2.21000e+02, 1.48000e+02, 2.70000e+02, 3.21000e+02,
6.00746e-01, 0.00000e+00],
         [2.68000e+02, 1.43000e+02, 3.26000e+02, 3.19000e+02,
5.99727e-01, 0.00000e+00],
         [5.60000e+02, 1.57000e+02, 6.23000e+02, 3.48000e+02,
5.69604e-01, 0.00000e+00],
         [1.11000e+02, 1.83000e+02, 1.48000e+02, 3.11000e+02,
5.08603e-01, 0.00000e+00],
         [1.85000e+02, 1.74000e+02, 2.22000e+02, 2.96000e+02,
4.83917e-01, 0.00000e+00],
         [6.76000e+02, 1.29000e+02, 8.12000e+02, 3.51000e+02,
4.49706e-01, 2.00000e+00],
         [5.19000e+02, 1.64000e+02, 5.54000e+02, 3.01000e+02,
4.43303e-01, 0.00000e+00],
         [5.60000e+01, 1.50000e+02, 1.24000e+02, 3.24000e+02,
4.10990e-01, 0.00000e+00],
         [0.00000e+00, 1.49000e+02, 3.30000e+01, 2.45000e+02,
3.99486e-01, 0.00000e+00],
         [5.43000e+02, 1.50000e+02, 5.87000e+02, 3.00000e+02,
3.86842e-01, 0.00000e+00],
         [5.77000e+02, 4.59000e+02, 7.29000e+02, 5.71000e+02,
3.11936e-01, 5.60000e+01],
         [3.14000e+02, 1.70000e+02, 3.53000e+02, 2.99000e+02,
2.83250e-01, 0.00000e+00],
         [2.54000e+02, 3.75000e+02, 9.38000e+02, 6.78000e+02,
2.74465e-01, 8.00000e+00]]),
 Ultralytics YOLO <class 'ultralytics.yolo.engine.results.Boxes'>
masks
 type: <class 'torch.Tensor'>
 shape: torch.Size([11, 6])
 dtype: torch.float32
  + tensor([[2.14000e+02, 1.17000e+02, 7.29000e+02, 6.13000e+02,
9.26095e-01, 0.00000e+00],
         [5.45000e+02, 1.34000e+02, 8.36000e+02, 5.54000e+02,
8.84632e-01, 0.00000e+00],
         [1.07000e+02, 1.81000e+02, 3.62000e+02, 5.72000e+02,
8.47854e-01, 0.00000e+00],
         [1.38000e+02, 3.92000e+02, 2.48000e+02, 5.79000e+02,
```

```
7.97918e-01, 5.60000e+01],
         [4.34000e+02, 4.69000e+02, 4.99000e+02, 6.16000e+02,
7.18869e-01, 4.00000e+01],
         [3.84000e+02, 4.71000e+02, 4.40000e+02, 6.22000e+02,
6.63171e-01, 3.90000e+01],
         [8.30000e+02, 3.44000e+02, 9.52000e+02, 5.50000e+02,
5.62907e-01, 5.60000e+01],
         [7.86000e+02, 4.23000e+02, 1.02300e+03, 6.74000e+02,
5.40777e-01, 5.80000e+01],
         [4.58000e+02, 3.52000e+02, 5.02000e+02, 4.81000e+02,
4.98751e-01, 2.70000e+01],
         [0.00000e+00, 2.15000e+02, 1.67000e+02, 5.59000e+02,
4.46715e-01, 5.60000e+01],
         [9.41000e+02, 2.55000e+02, 1.02400e+03, 4.63000e+02,
2.94481e-01, 0.00000e+00]]),
 Ultralytics YOLO <class 'ultralytics.yolo.engine.results.Boxes'>
masks
 type: <class 'torch.Tensor'>
 shape: torch.Size([8, 6])
 dtype: torch.float32
  + tensor([[5.03000e+02, 1.33000e+02, 7.95000e+02, 5.80000e+02,
7.58295e-01, 0.00000e+00],
         [0.00000e+00, 1.20000e+02, 1.19000e+02, 6.31000e+02,
7.45652e-01, 0.00000e+00],
         [9.13000e+02, 2.91000e+02, 1.02400e+03, 5.69000e+02,
6.80909e-01, 0.00000e+00],
         [7.51000e+02, 1.73000e+02, 9.14000e+02, 5.82000e+02,
6.51038e-01, 0.00000e+00],
         [9.20000e+02, 5.52000e+02, 1.02300e+03, 6.58000e+02,
6.27026e-01, 0.00000e+00],
         [4.98000e+02, 5.06000e+02, 7.44000e+02, 6.58000e+02,
5.55051e-01, 0.00000e+00],
         [3.89000e+02, 2.42000e+02, 5.32000e+02, 5.14000e+02,
4.75268e-01, 0.00000e+00],
         [5.87000e+02, 5.07000e+02, 7.45000e+02, 6.58000e+02,
2.84247e-01, 0.00000e+00]])]
```

```
len(results)
```

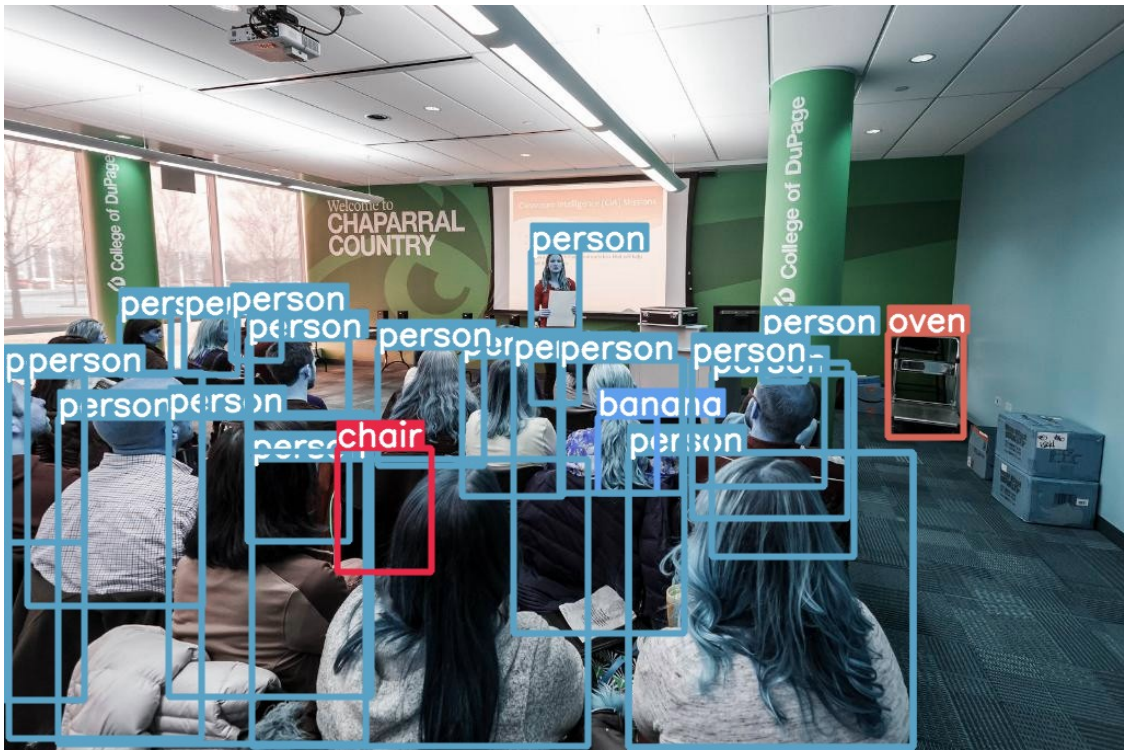
```
24
```

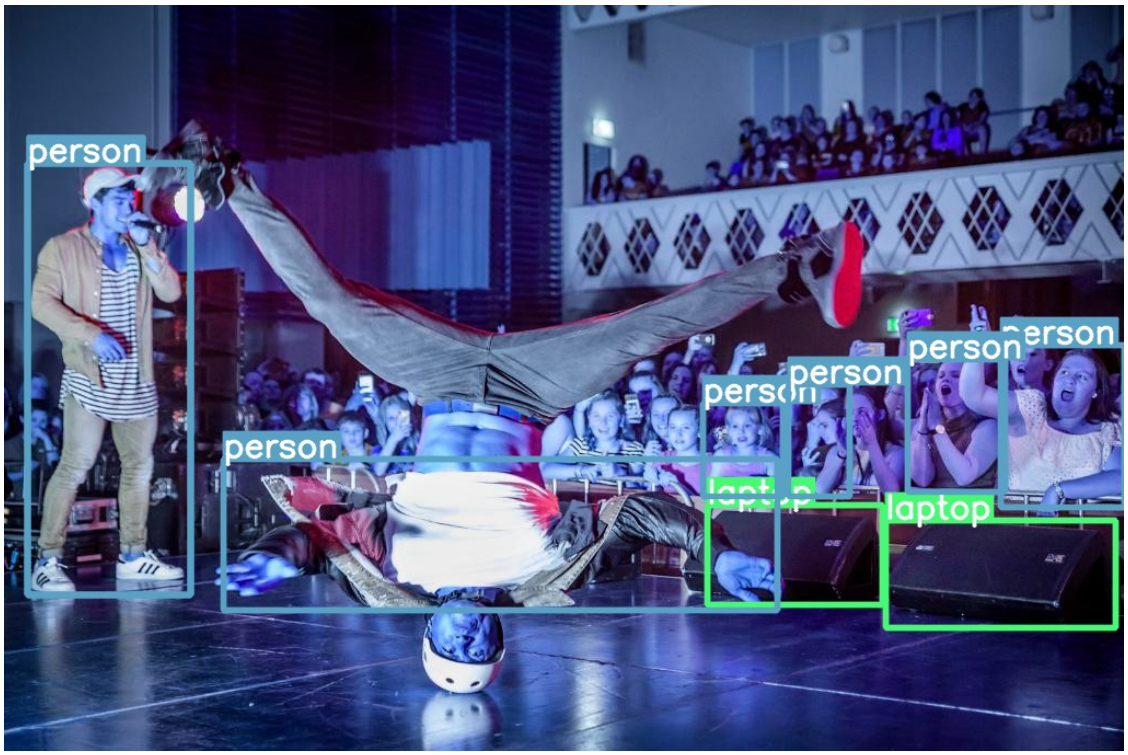
```
def box_label(image, box, label='', color=(128, 128, 128),
txt_color=(255, 255, 255)):
  lw = max(round(sum(image.shape) / 2 * 0.003), 2)
  p1, p2 = (int(box[0]), int(box[1])), (int(box[2]), int(box[3]))
  cv2.rectangle(image, p1, p2, color, thickness=lw,
lineType=cv2.LINE_AA)
  if label:
    tf = max(lw - 1, 1)  # font thickness
    w, h = cv2.getTextSize(label, 0, fontScale=lw / 3, thickness=tf)
[0]  # text width, height
```

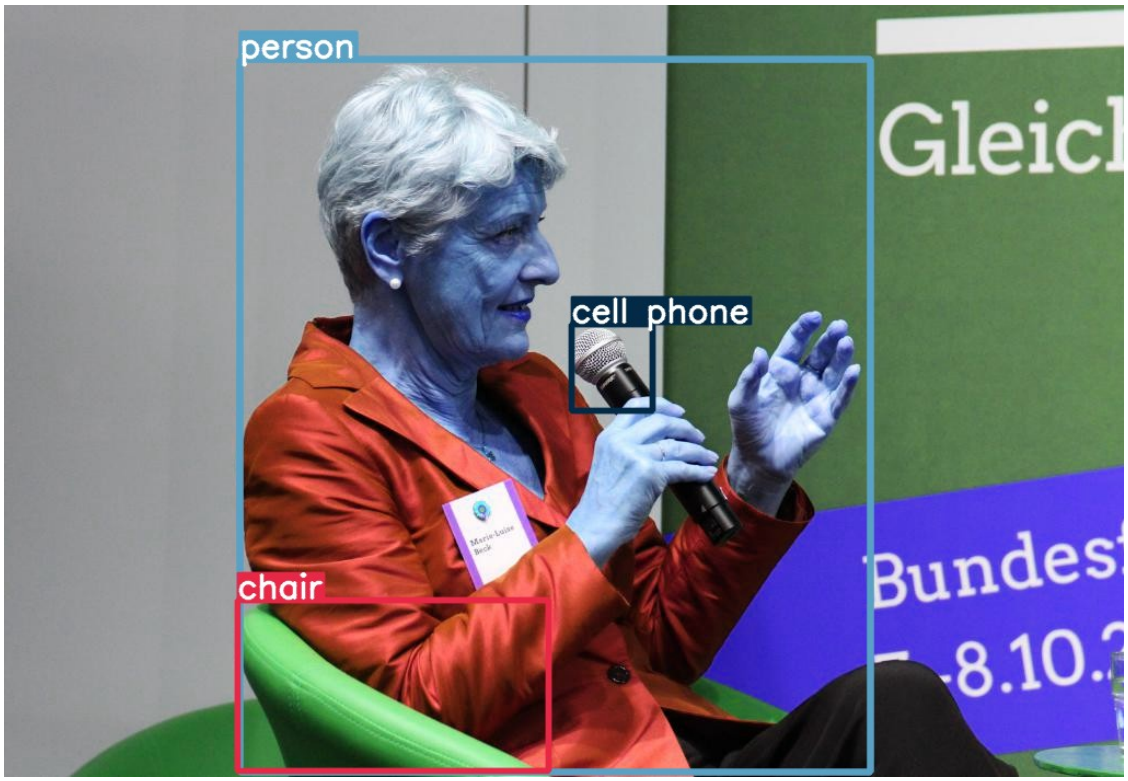
```
    outside = p1[1] - h >= 3
    p2 = p1[0] + w, p1[1] - h - 3 if outside else p1[1] + h + 3
    cv2.rectangle(image, p1, p2, color, -1, cv2.LINE_AA)  # filled
    cv2.putText(image,
                label, (p1[0], p1[1] - 2 if outside else p1[1] + h +
2),
                0,
                lw / 3,
                txt_color,
                thickness=tf,
                lineType=cv2.LINE_AA)

from google.colab.patches import cv2_imshow
def plot_bboxes(image, boxes, labels=[], colors=[], score=True,
conf=None):
  #Define COCO Labels
  if labels == []:
    labels = {0: u'__background__', 1: u'person', 2: u'bicycle',3:
u'car', 4: u'motorcycle', 5: u'airplane', 6: u'bus', 7: u'train', 8:
u'truck', 9: u'boat', 10: u'traffic light', 11: u'fire hydrant', 12:
u'stop sign', 13: u'parking meter', 14: u'bench', 15: u'bird', 16:
u'cat', 17: u'dog', 18: u'horse', 19: u'sheep', 20: u'cow', 21:
u'elephant', 22: u'bear', 23: u'zebra', 24: u'giraffe', 25:
u'backpack', 26: u'umbrella', 27: u'handbag', 28: u'tie', 29:
u'suitcase', 30: u'frisbee', 31: u'skis', 32: u'snowboard', 33:
u'sports ball', 34: u'kite', 35: u'baseball bat', 36: u'baseball
glove', 37: u'skateboard', 38: u'surfboard', 39: u'tennis racket', 40:
u'bottle', 41: u'wine glass', 42: u'cup', 43: u'fork', 44: u'knife',
45: u'spoon', 46: u'bowl', 47: u'banana', 48: u'apple', 49:
u'sandwich', 50: u'orange', 51: u'broccoli', 52: u'carrot', 53: u'hot
dog', 54: u'pizza', 55: u'donut', 56: u'cake', 57: u'chair', 58:
u'couch', 59: u'potted plant', 60: u'bed', 61: u'dining table', 62:
u'toilet', 63: u'tv', 64: u'laptop', 65: u'mouse', 66: u'remote', 67:
u'keyboard', 68: u'cell phone', 69: u'microwave', 70: u'oven', 71:
u'toaster', 72: u'sink', 73: u'refrigerator', 74: u'book', 75:
u'clock', 76: u'vase', 77: u'scissors', 78: u'teddy bear', 79: u'hair
drier', 80: u'toothbrush'}
  #Define colors
  if colors == []:
    #colors = [(6, 112, 83), (253, 246, 160), (40, 132, 70), (205, 97,
162), (149, 196, 30), (106, 19, 161), (127, 175, 225), (115, 133,
176), (83, 156, 8), (182, 29, 77), (180, 11, 251), (31, 12, 123), (23,
6, 115), (167, 34, 31), (176, 216, 69), (110, 229, 222), (72, 183,
159), (90, 168, 209), (195, 4, 209), (135, 236, 21), (62, 209, 199),
(87, 1, 70), (75, 40, 168), (121, 90, 126), (11, 86, 86), (40, 218,
53), (234, 76, 20), (129, 174, 192), (13, 18, 254), (45, 183, 149),
(77, 234, 120), (182, 83, 207), (172, 138, 252), (201, 7, 159), (147,
240, 17), (134, 19, 233), (202, 61, 206), (177, 253, 26), (10, 139,
17), (130, 148, 106), (174, 197, 128), (106, 59, 168), (124, 180, 83),
(78, 169, 4), (26, 79, 176), (185, 149, 150), (165, 253, 206), (220,
```

```
87, 0), (72, 22, 226), (64, 174, 4), (245, 131, 96), (35, 217, 142),
(89, 86, 32), (80, 56, 196), (222, 136, 159), (145, 6, 219), (143,
132, 162), (175, 97, 221), (72, 3, 79), (196, 184, 237), (18, 210,
116), (8, 185, 81), (99, 181, 254), (9, 127, 123), (140, 94, 215),
(39, 229, 121), (230, 51, 96), (84, 225, 33), (218, 202, 139), (129,
223, 182), (167, 46, 157), (15, 252, 5), (128, 103, 203), (197, 223,
199), (19, 238, 181), (64, 142, 167), (12, 203, 242), (69, 21, 41),
(177, 184, 2), (35, 97, 56), (241, 22, 161)]
    colors = [(89, 161, 197),(67, 161, 255),(19, 222, 24),(186, 55,
2),(167, 146, 11),(190, 76, 98),(130, 172, 179),(115, 209, 128),(204,
79, 135),(136, 126, 185),(209, 213, 45),(44, 52, 10),(101, 158, 121),
(179, 124, 12),(25, 33, 189),(45, 115, 11),(73, 197, 184),(62, 225,
221),(32, 46, 52),(20, 165, 16),(54, 15, 57),(12, 150, 9),(10, 46,
99),(94, 89, 46),(48, 37, 106),(42, 10, 96),(7, 164, 128),(98, 213,
120),(40, 5, 219),(54, 25, 150),(251, 74, 172),(0, 236, 196),(21, 104,
190),(226, 74, 232),(120, 67, 25),(191, 106, 197),(8, 15, 134),(21, 2,
1),(142, 63, 109),(133, 148, 146),(187, 77, 253),(155, 22, 122),(218,
130, 77),(164, 102, 79),(43, 152, 125),(185, 124, 151),(95, 159, 238),
(128, 89, 85),(228, 6, 60),(6, 41, 210),(11, 1, 133),(30, 96, 58),
(230, 136, 109),(126, 45, 174),(164, 63, 165),(32, 111, 29),(232, 40,
70),(55, 31, 198),(148, 211, 129),(10, 186, 211),(181, 201, 94),(55,
35, 92),(129, 140, 233),(70, 250, 116),(61, 209, 152),(216, 21, 138),
(100, 0, 176),(3, 42, 70),(151, 13, 44),(216, 102, 88),(125, 216, 93),
(171, 236, 47),(253, 127, 103),(205, 137, 244),(193, 137, 224),(36,
152, 214),(17, 50, 238),(154, 165, 67),(114, 129, 60),(119, 24, 48),
(73, 8, 110)]

  #plot each boxes
  for box in boxes:
    #add score in label if score=True
    if score :
      label = labels[int(box[-1])+1] + " " + str(round(100 *
float(box[-2]),1)) + "%"
    else :
      label = labels[int(box[-1])+1]
    #filter every box under conf threshold if conf threshold setted
    if conf :
      if box[-2] > conf:
        color = colors[int(box[-1])]
        box_label(image, box, label, color)
    else:
      color = colors[int(box[-1])]
      box_label(image, box, label, color)

  #show image
  #image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)

  try:
    import google.colab
    IN_COLAB = True
```

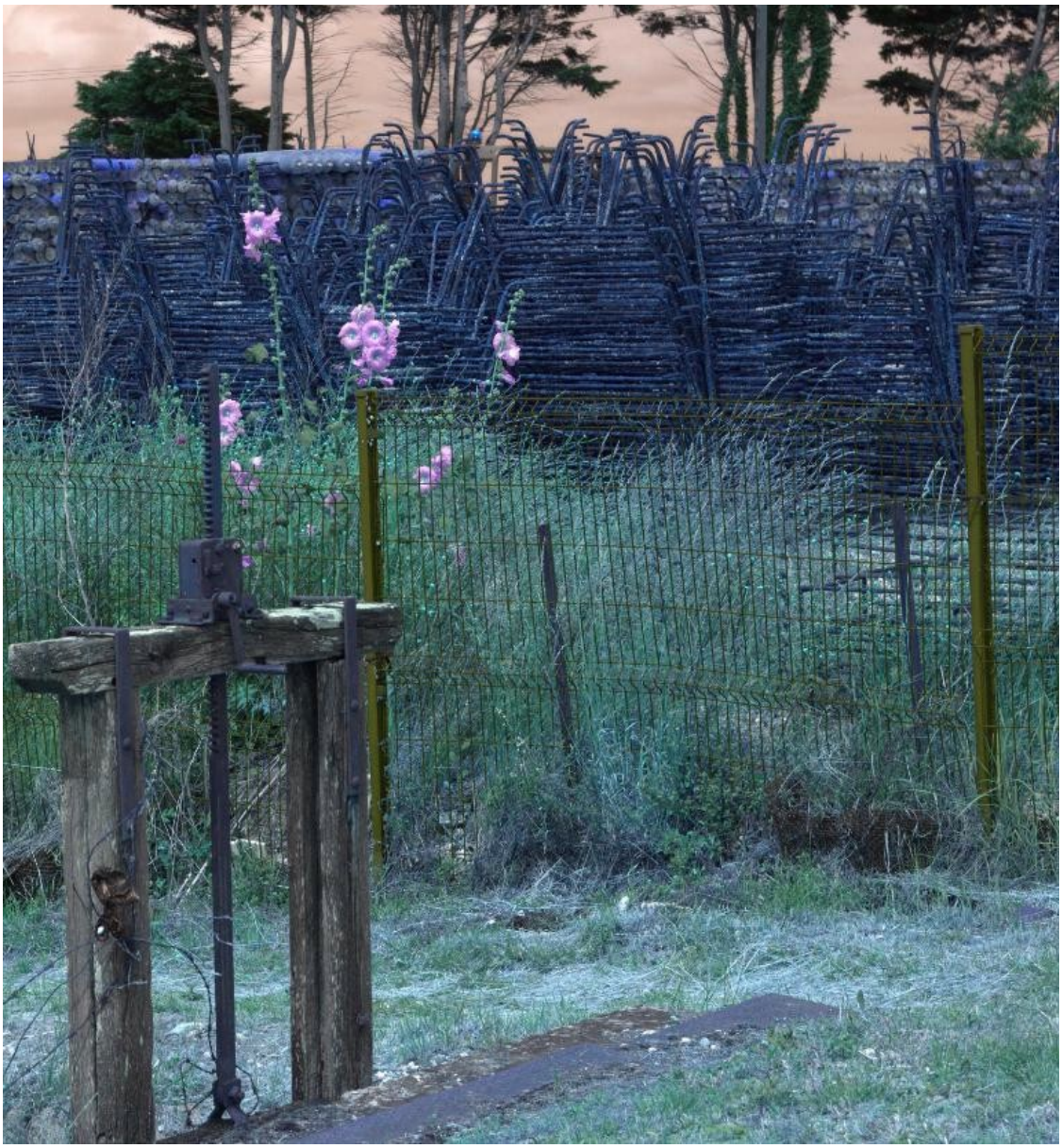
```
  except:
    IN_COLAB = False


  if IN_COLAB:
    cv2_imshow(image) #if used in Colab
  else :
    cv2.imshow(image) #if used in Python

#Plotting the output of YOLOv8
%matplotlib inline
plt.figure(figsize=(10,10), dpi=200)
for i in range(len(list_img)):
  plot_bboxes(np_img[i], results[i].boxes.boxes, score=False)
```


College of DuPage
Welcome to
CHAPARRAL
COUNTRY
College of DuPage
person
person
person
person
person
person
person
person
person
person
person
person
person
oven
chair
banana


person
person
person
person
person
person
laptop

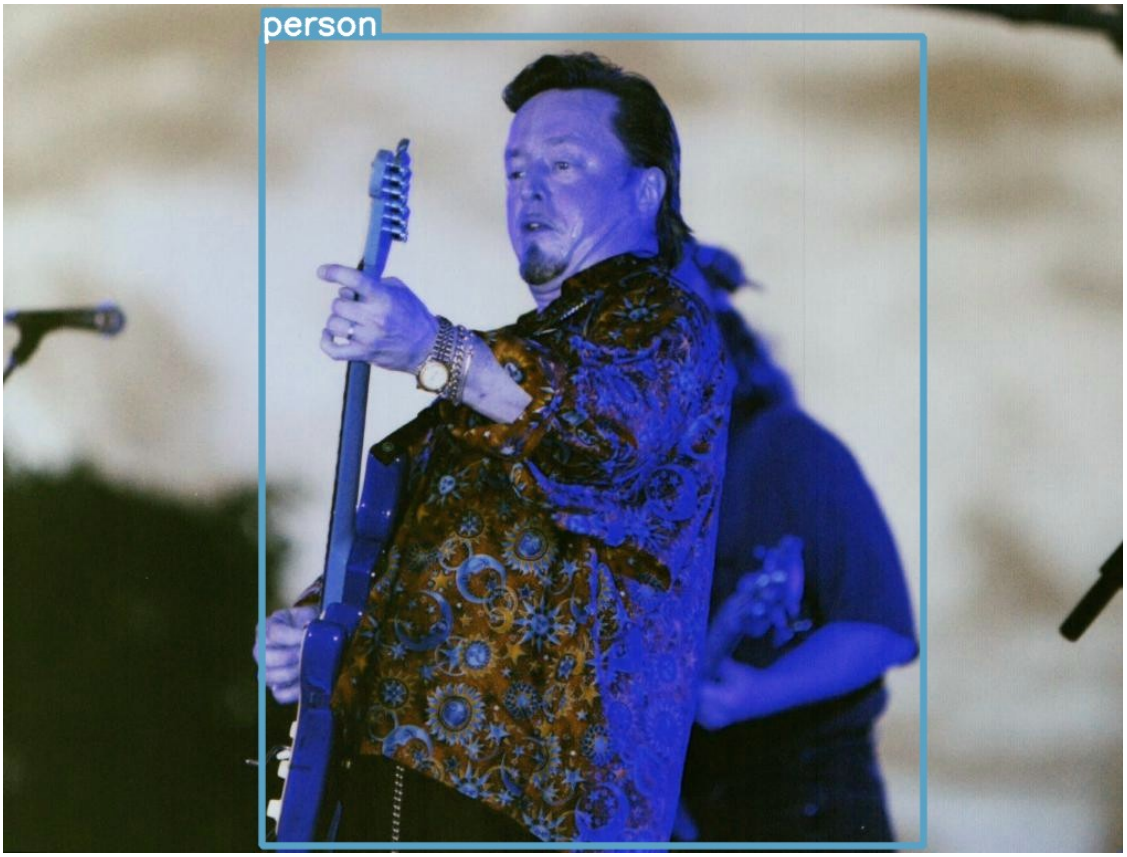

person
cell phone
chair


truck
OVERLAND PARK
OPFD
R42
Tyler Silvest - January 2017

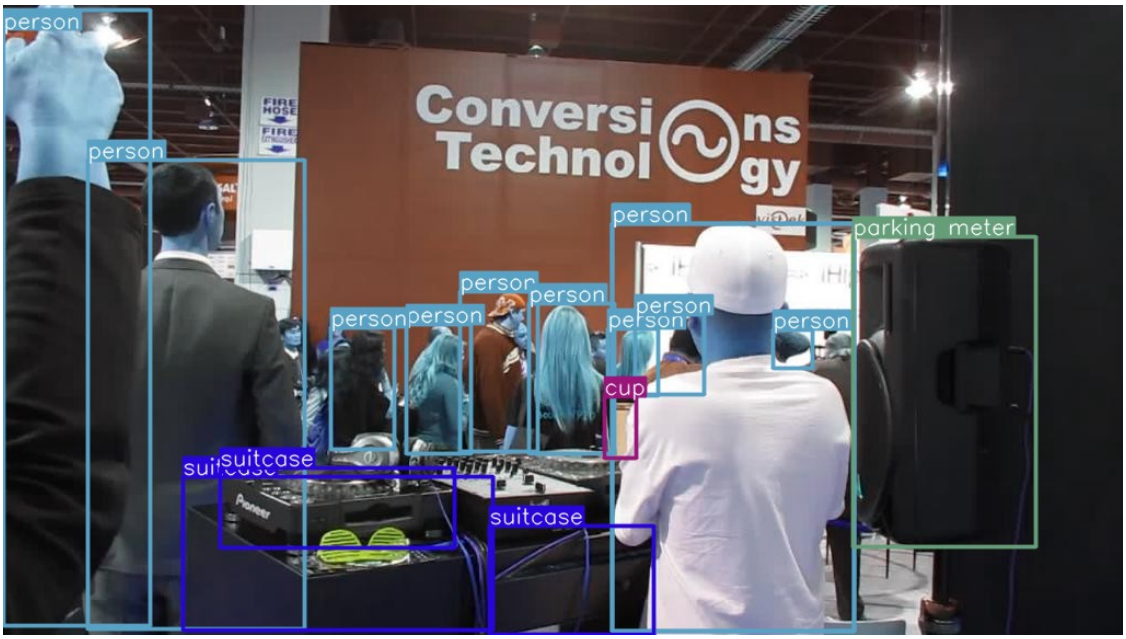


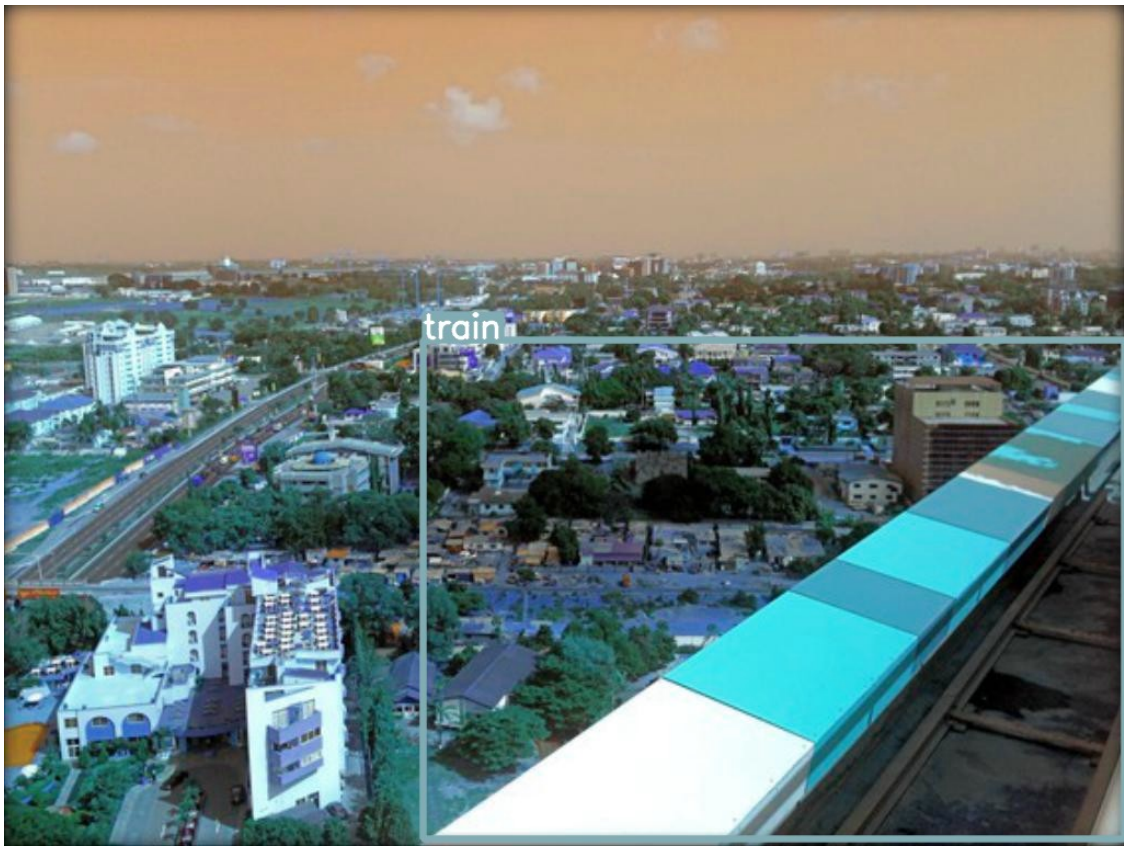

person
tie
person
VALOR
DEPARTMENT OF DEFENSE
UNITED STATES OF AMERICA


person
tv

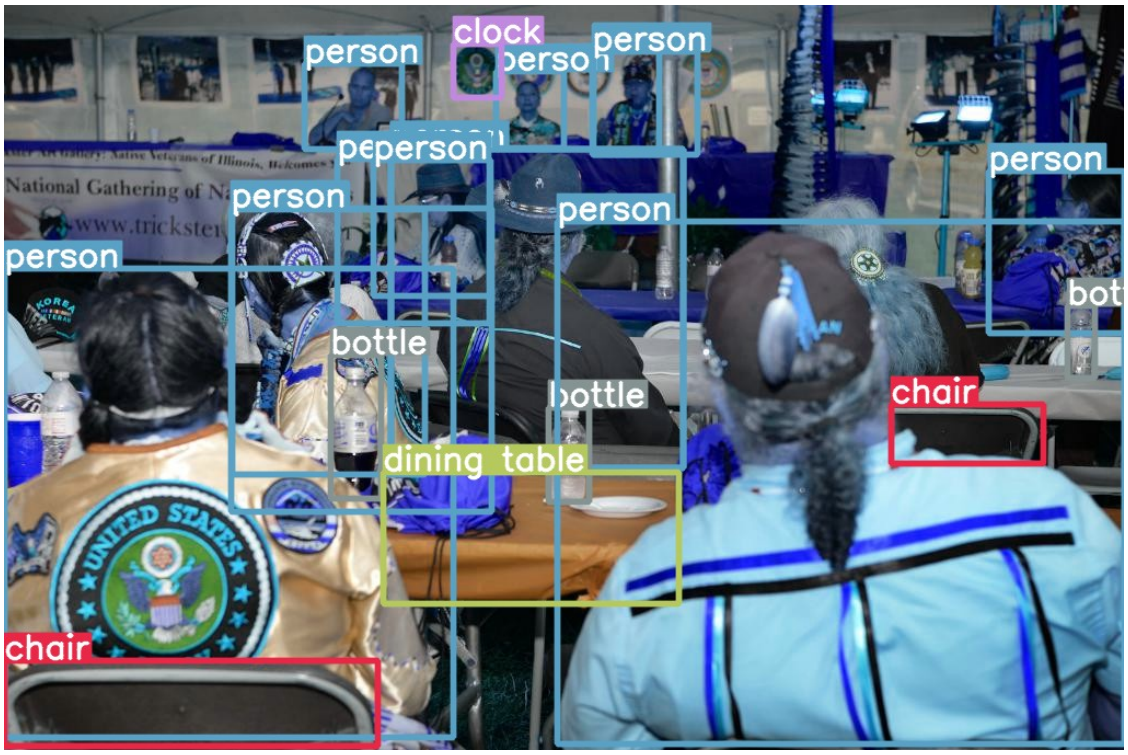

person

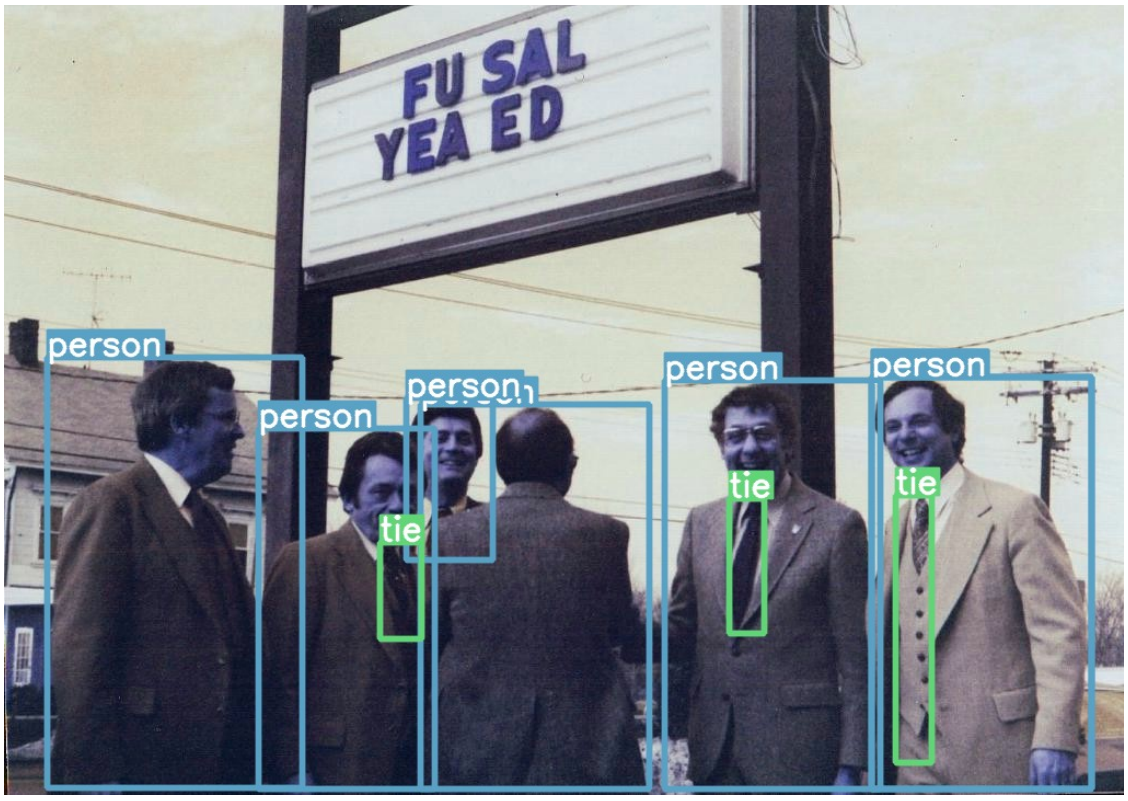

©Álbum 3488 mcr

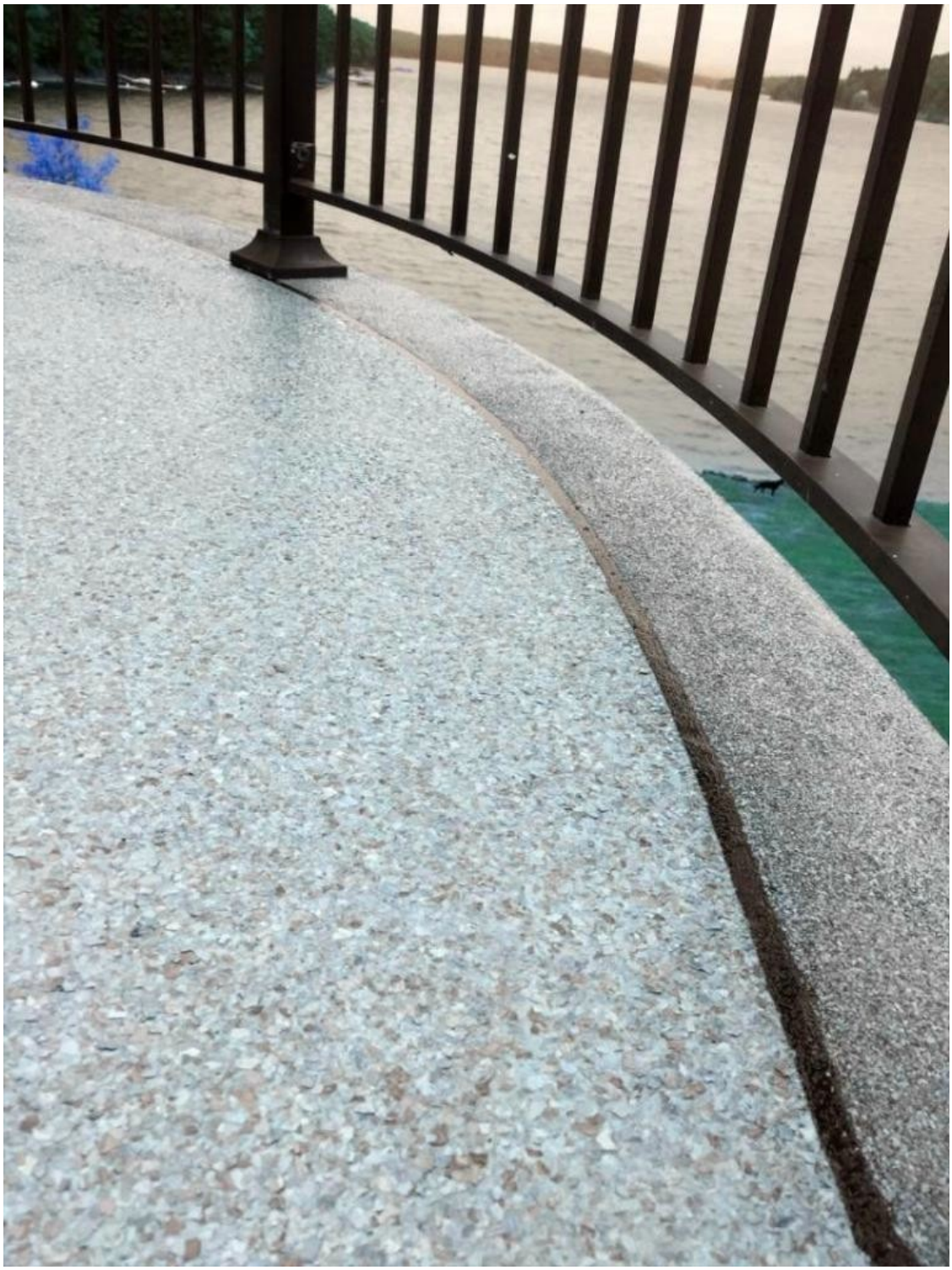

person
person
Conversi ns
Technol gy
person
parking meter
person
person
person
person
person
person
cup
suitcase
suitcase
Pioneer

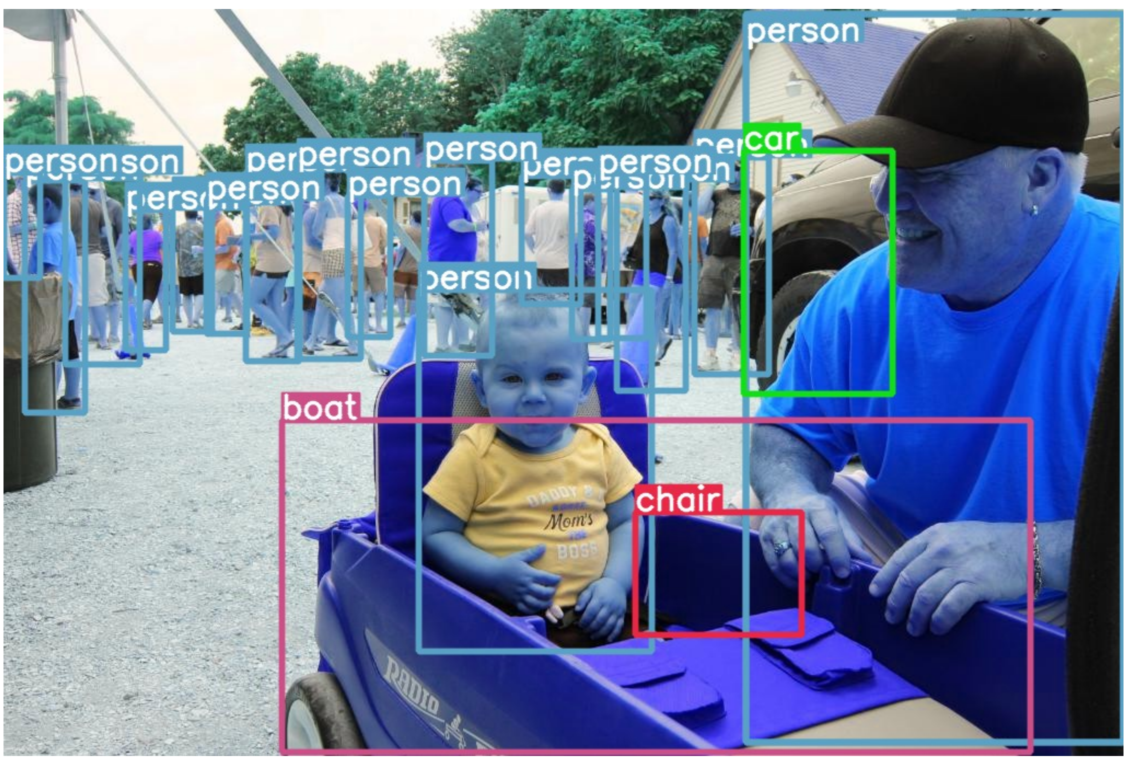

train

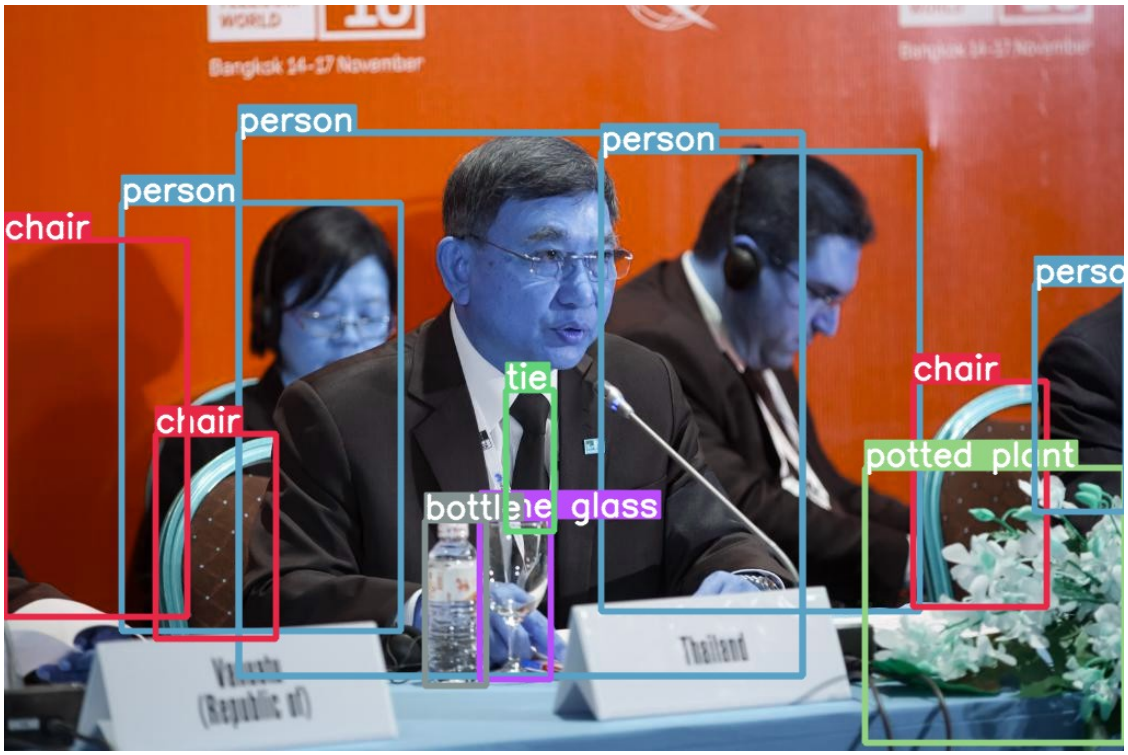

person
person
person

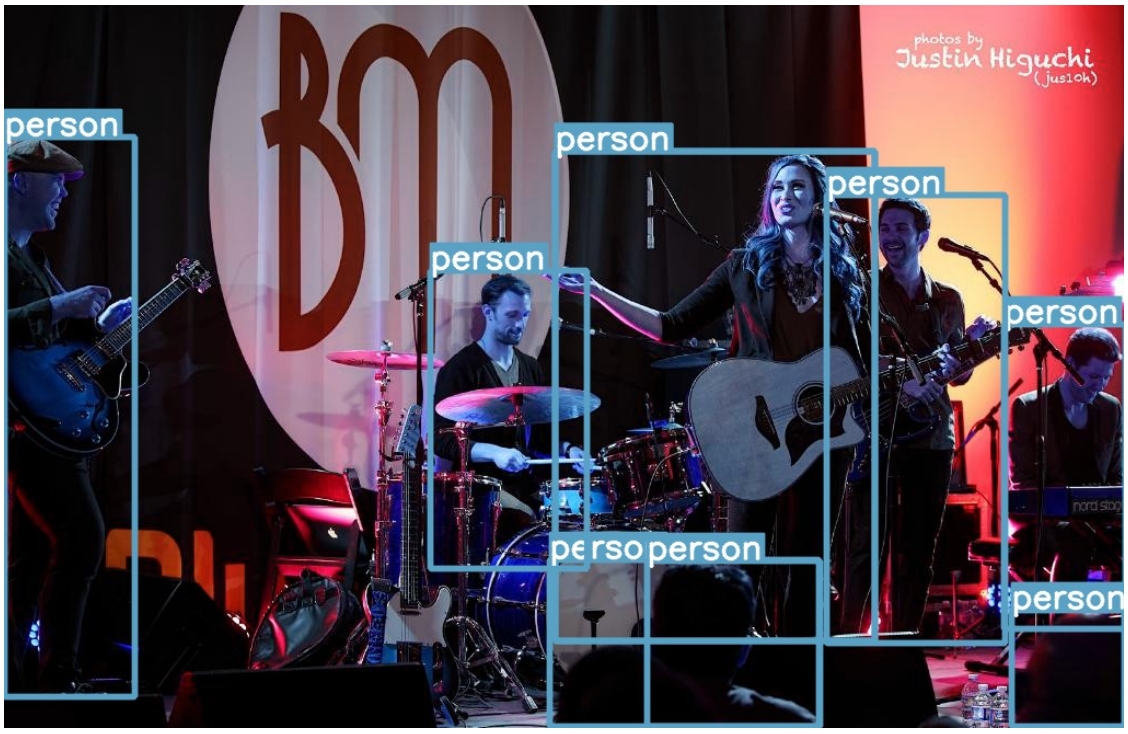

clock
person
person
peperson
person
person
person
person
bot
bottle
bottle
chair
dining table
chair


FU SAL
YEA ED
person
person
person
tie
person
tie
person
tie




person
car
person
boat
chair





<Figure size 2000x2000 with 0 Axes>

Dataset2 (50)

```
# mount Google Drive
from google.colab import drive
drive.mount('/content/gdrive')
%cd gdrive/MyDrive
```

```
Drive already mounted at /content/gdrive; to attempt to forcibly
remount, call drive.mount("/content/gdrive", force_remount=True).
[Errno 2] No such file or directory: 'gdrive/MyDrive'
/content/gdrive/MyDrive
```

```
#loading the images

list_img2 = os.listdir("yolov7/unsplash-images-collection")
list_img2 = list_img2[:50]
len(list_img2)
```

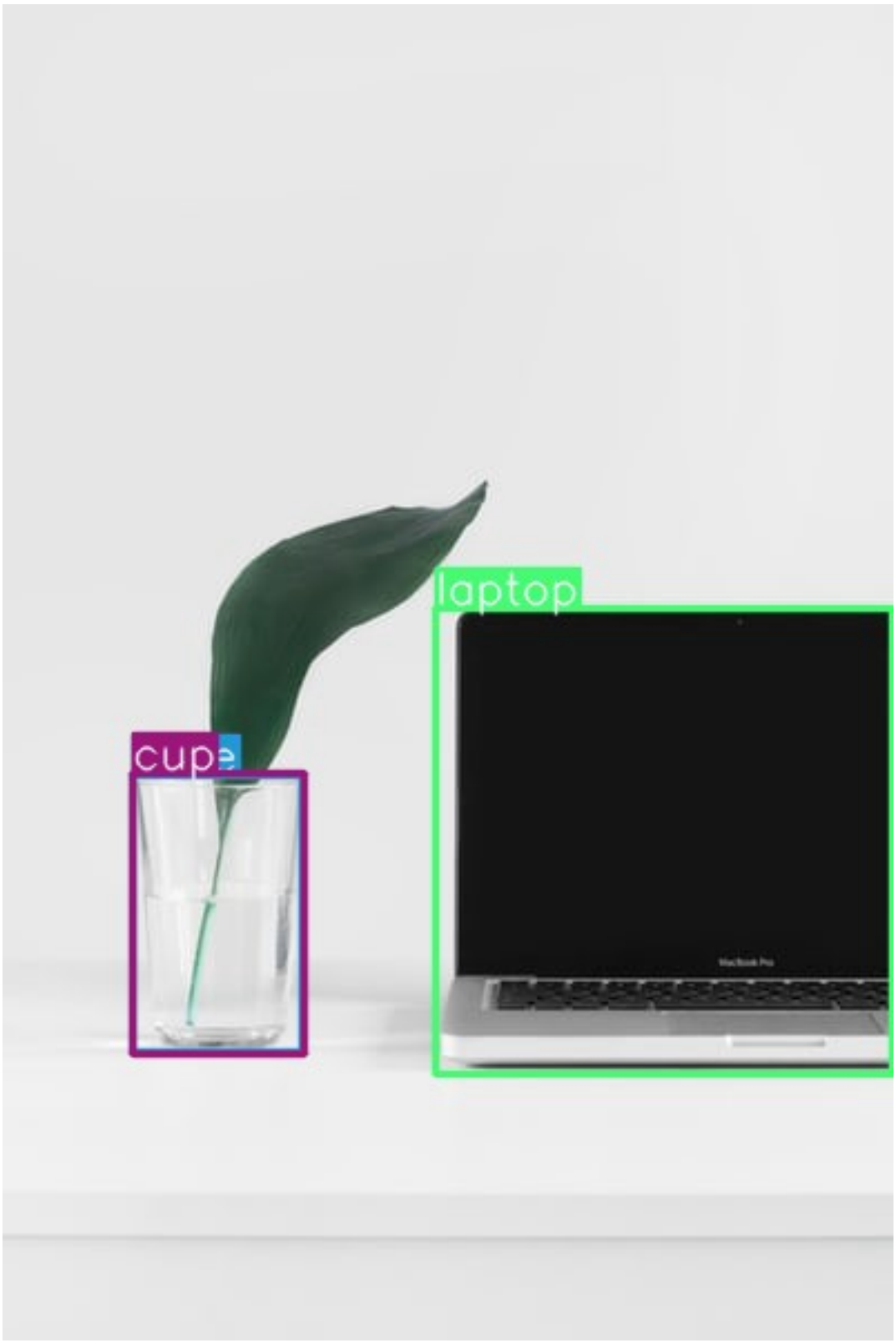
```
50
```

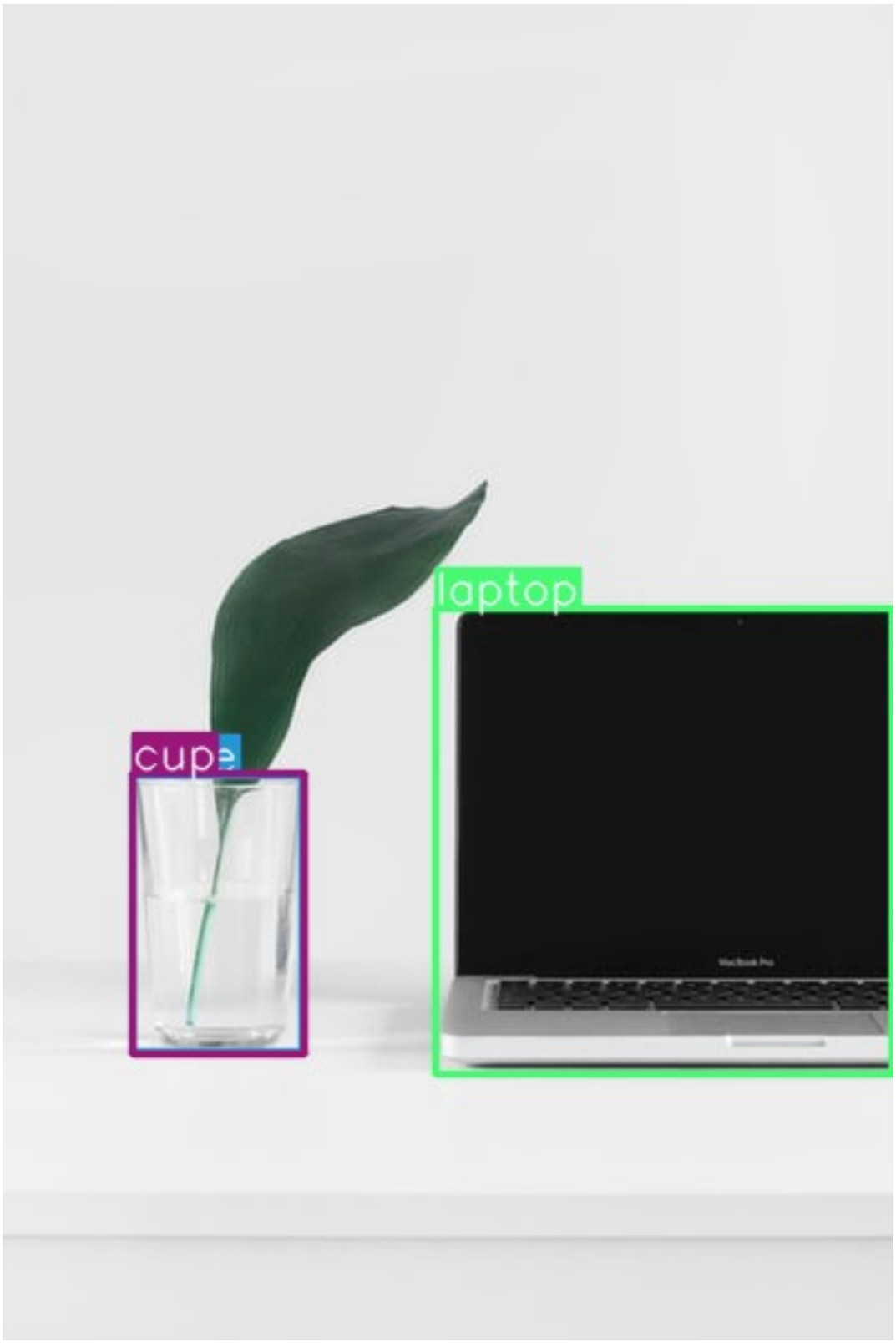
```
for i in range(len(list_img2)):
  list_img2[i]='yolov7/unsplash-images-collection/'+ list_img2[i]

np_img2 = []
for i in range(len(list_img2)):
  np_img2.append(cv2.imread(list_img2[i]))

%%time
results = modelv8.predict(np_img2)
```

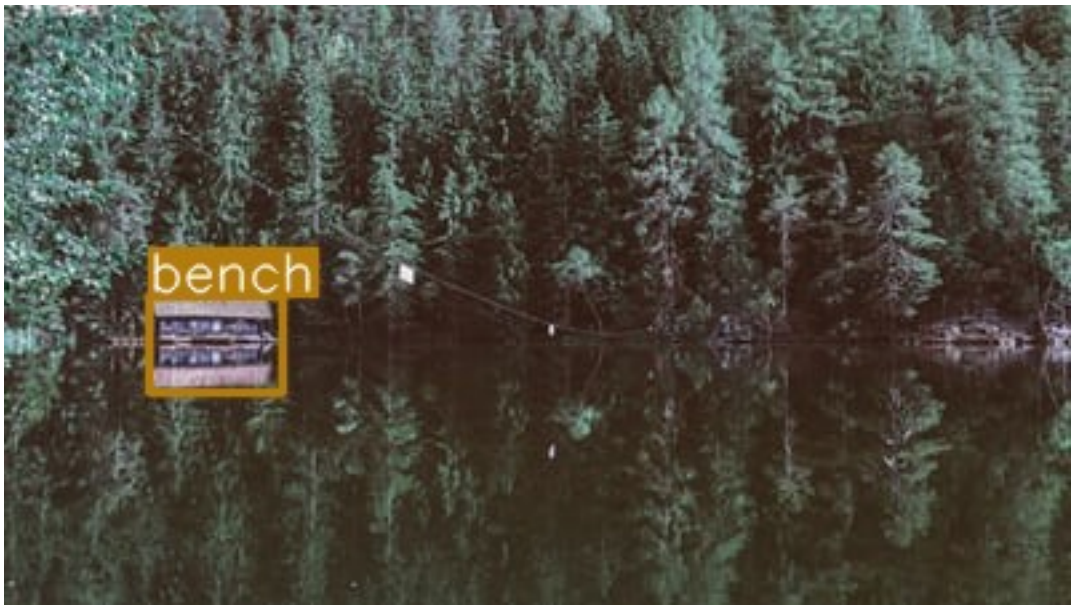
```
CPU times: user 10.5 s, sys: 1.37 s, total: 11.9 s
Wall time: 11.9 s
```

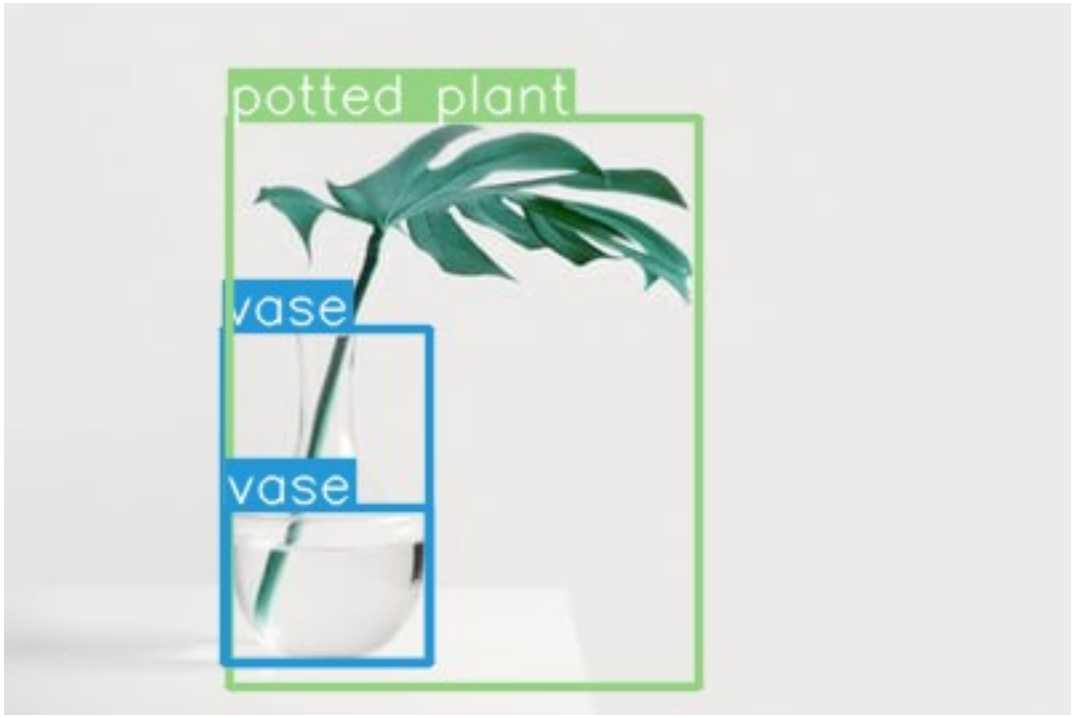
```
#Plotting the output of YOLOv8
%matplotlib inline
plt.figure(figsize=(10,10), dpi=200)
for i in range(len(list_img2)):
  plot_bboxes(np_img2[i], results[i].boxes.boxes, score=False)
```


vase


laptop
cup


laptop
cup


bench

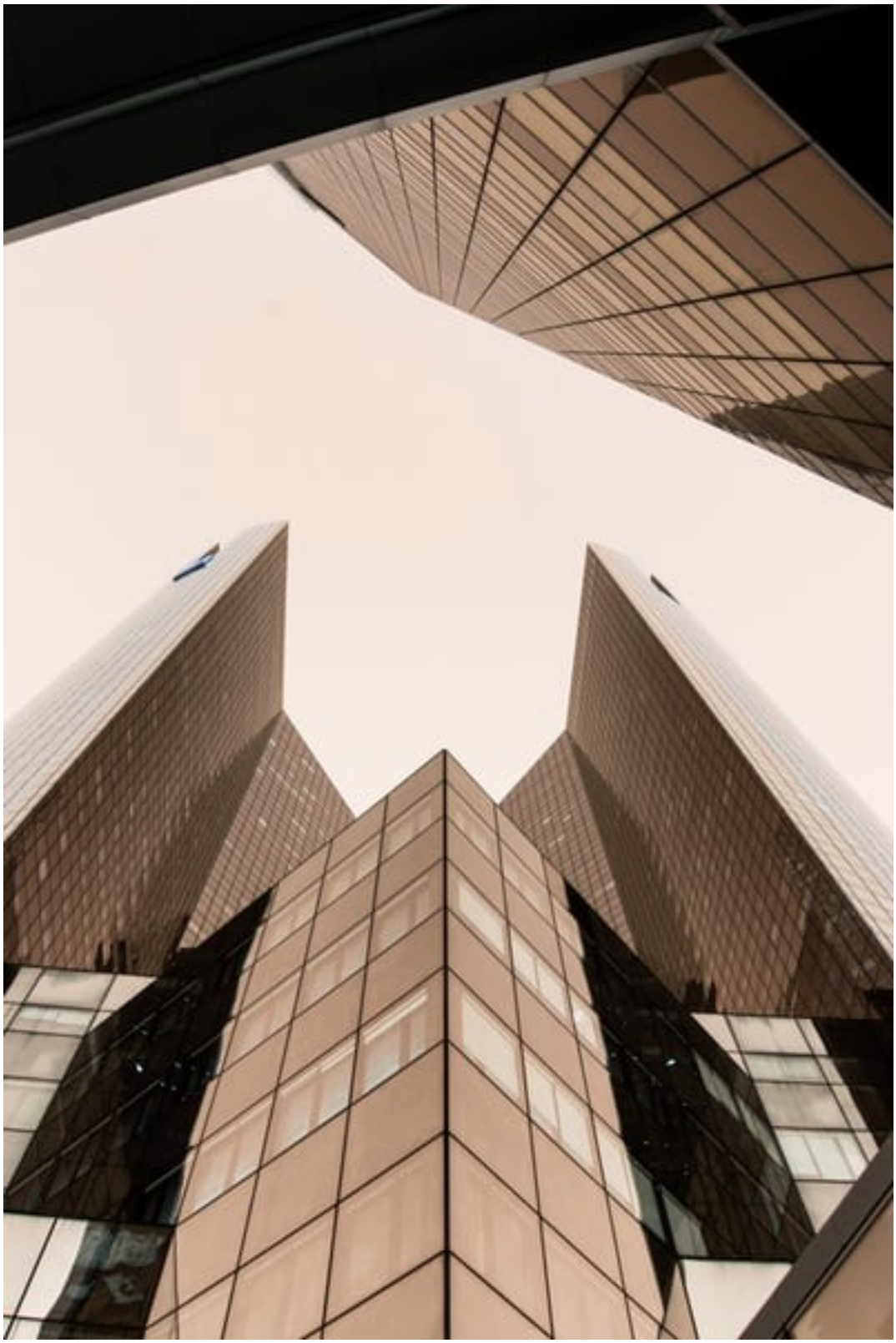

potted plant
vase
vase

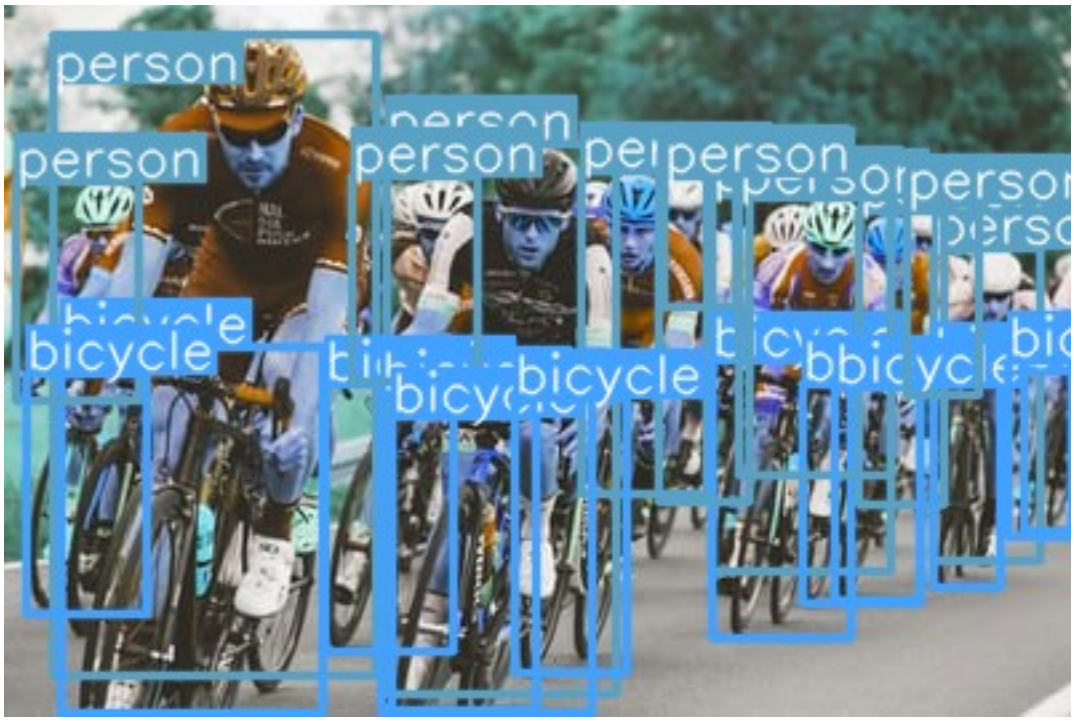

person


surfboard

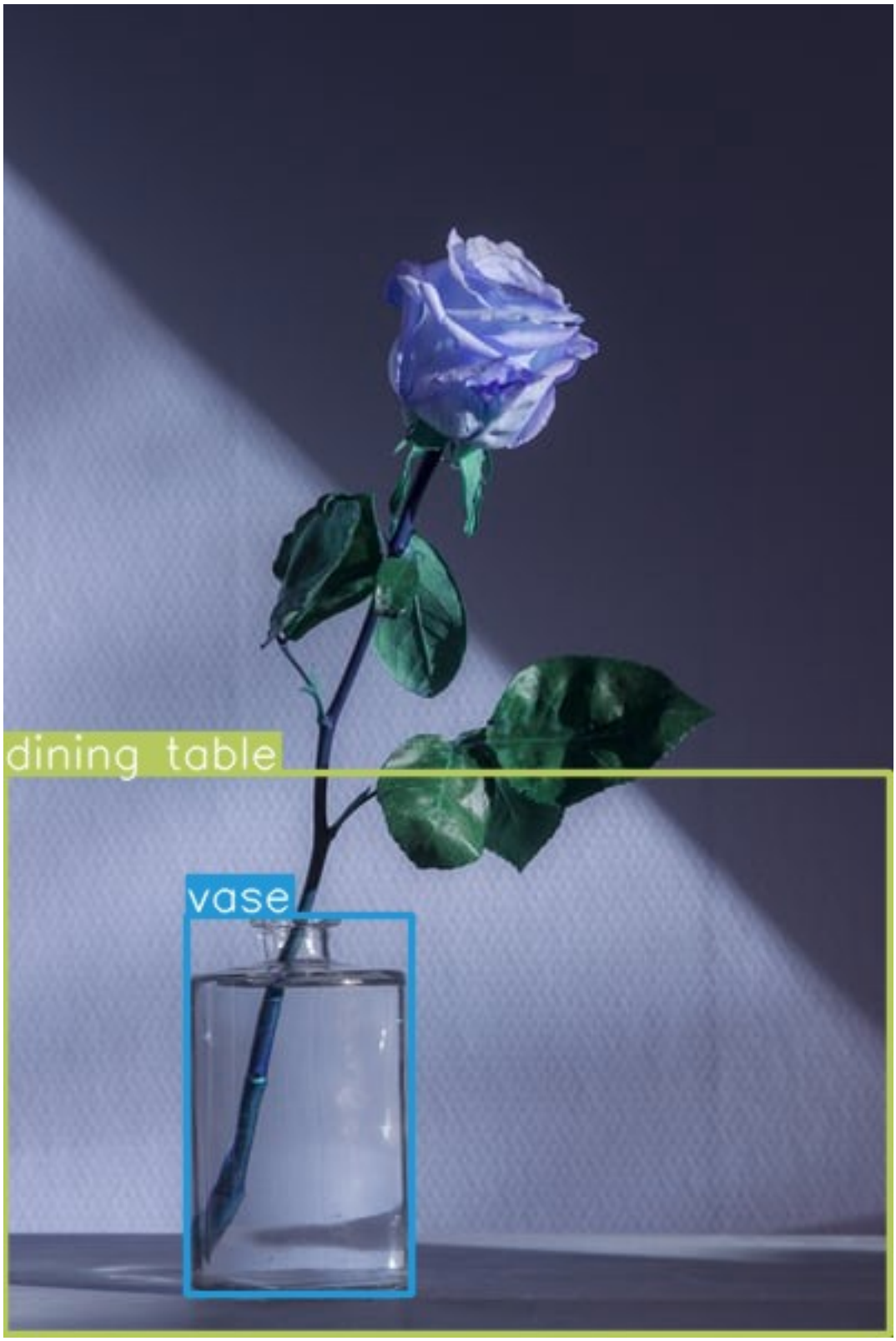

person


person

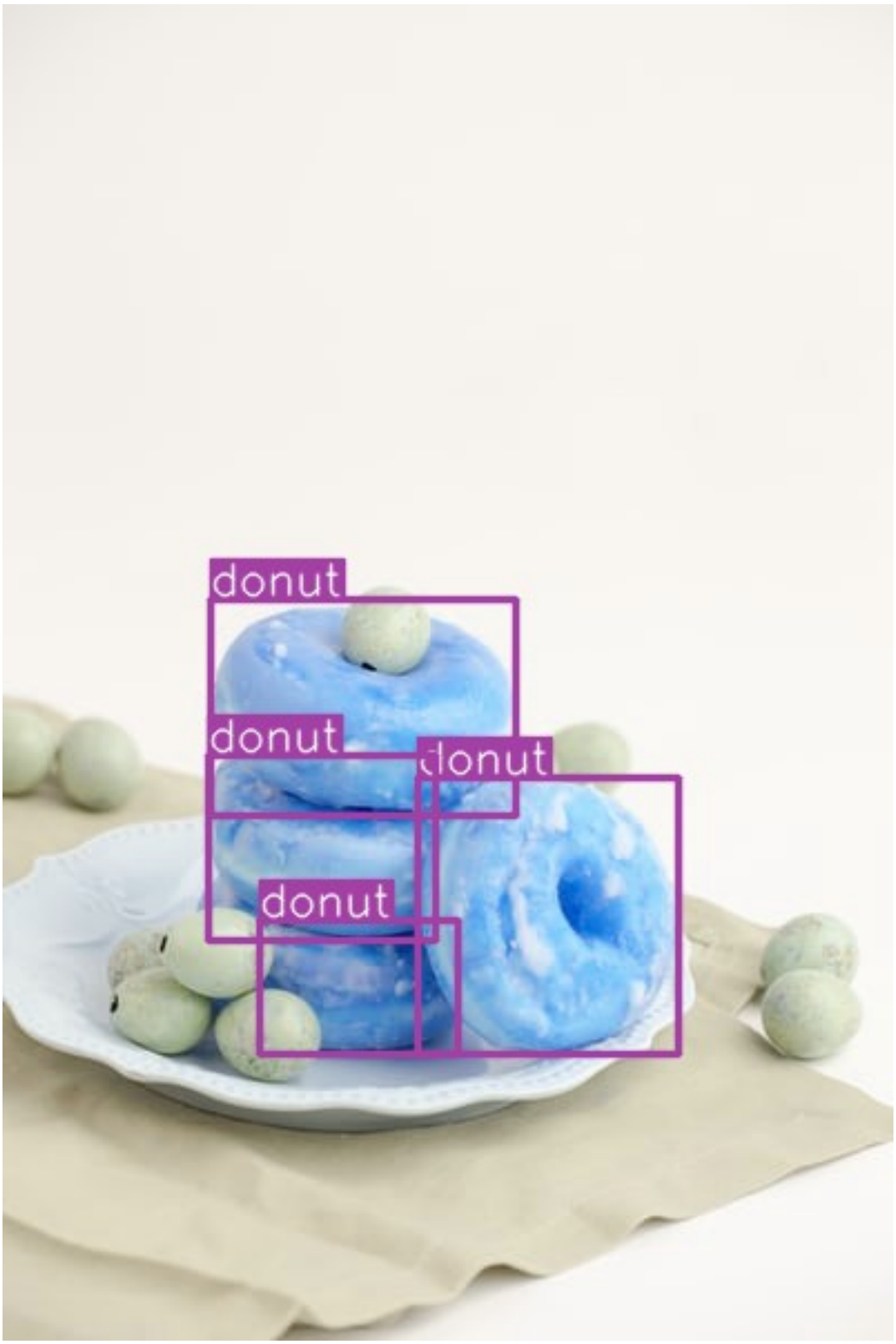



person
person
person
person
person
bicycle
bicycle
bicycle


PISO 1

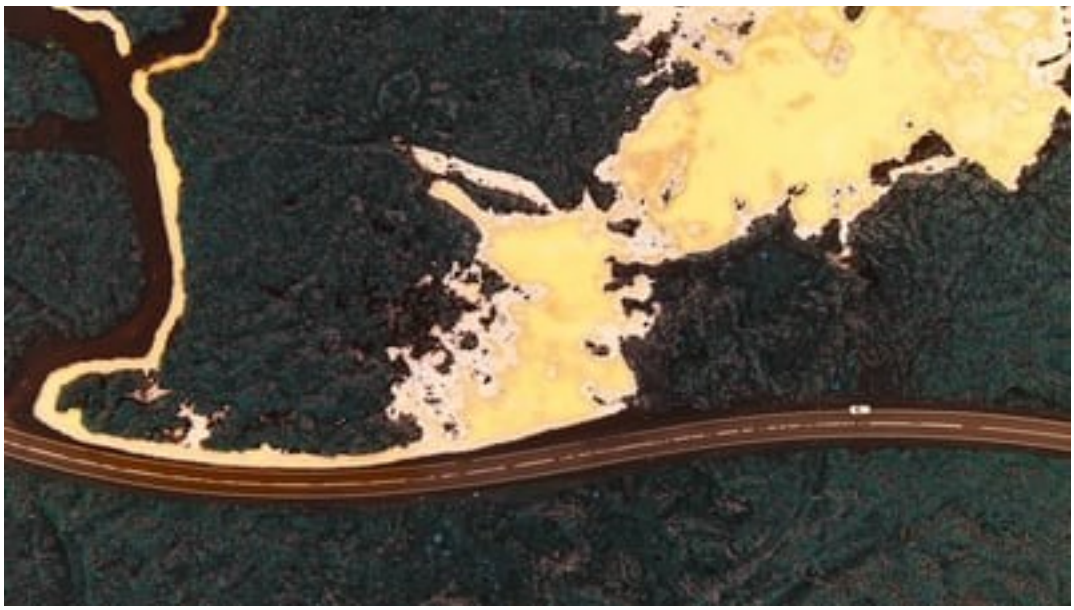

dining table
vase


person


donut
donut
donut
donut


person
person

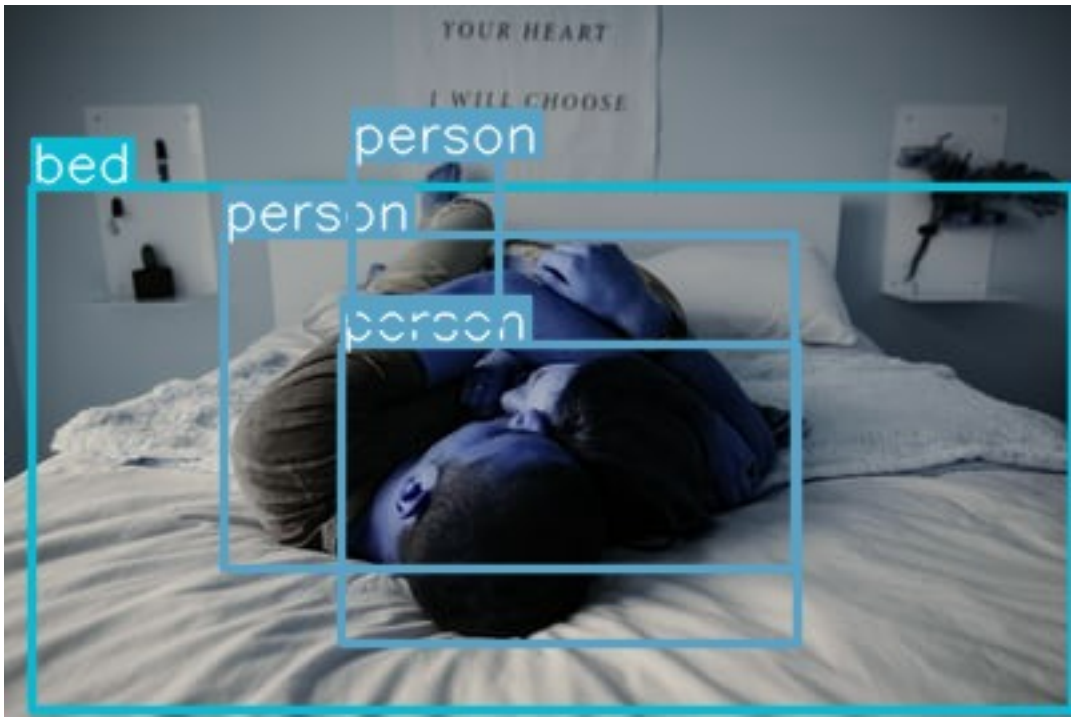

donut



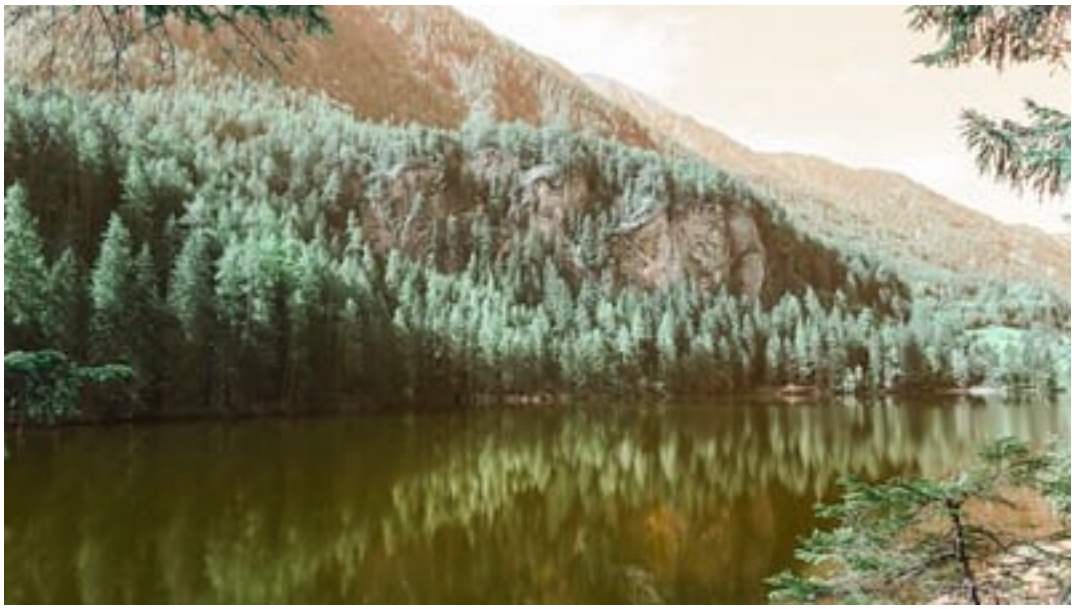

person
cake


bench

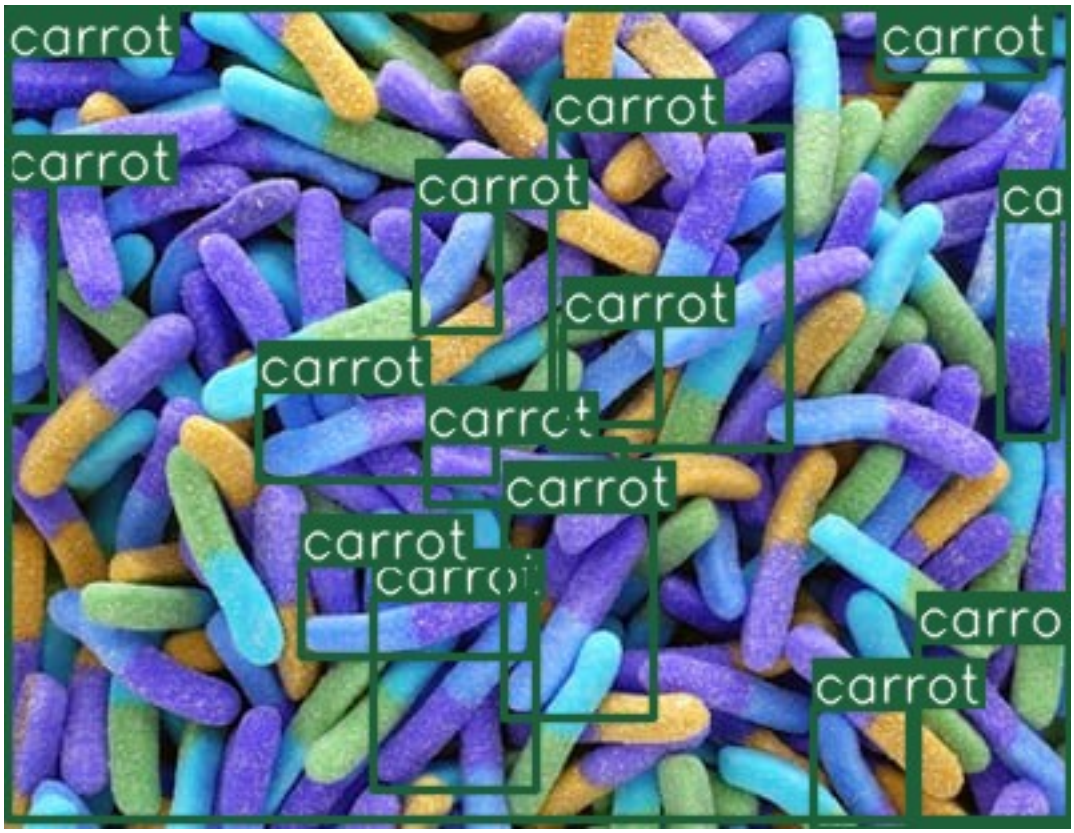

donut


YOUR HEART
I WILL CHOOSE
bed
person
person
person

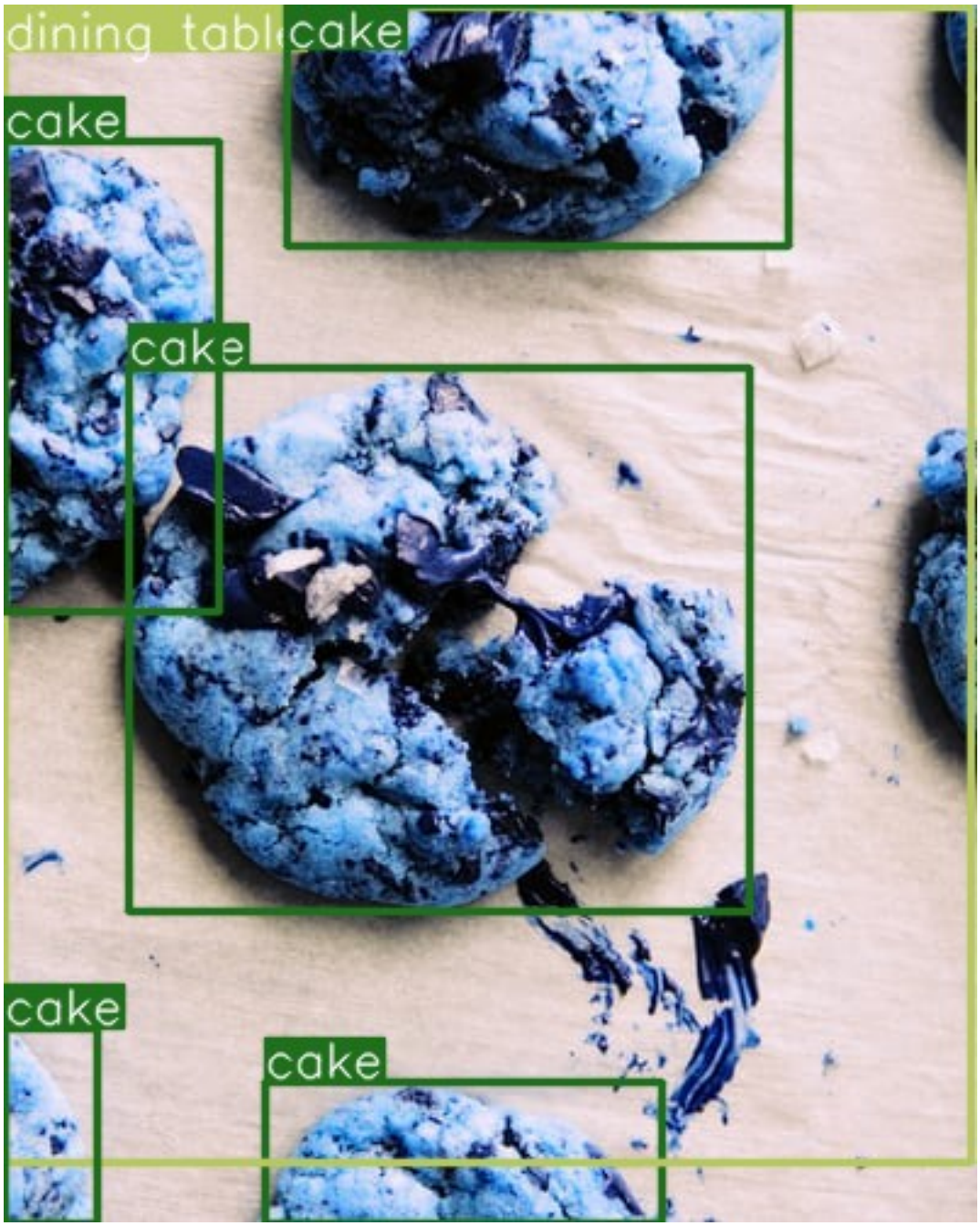

person

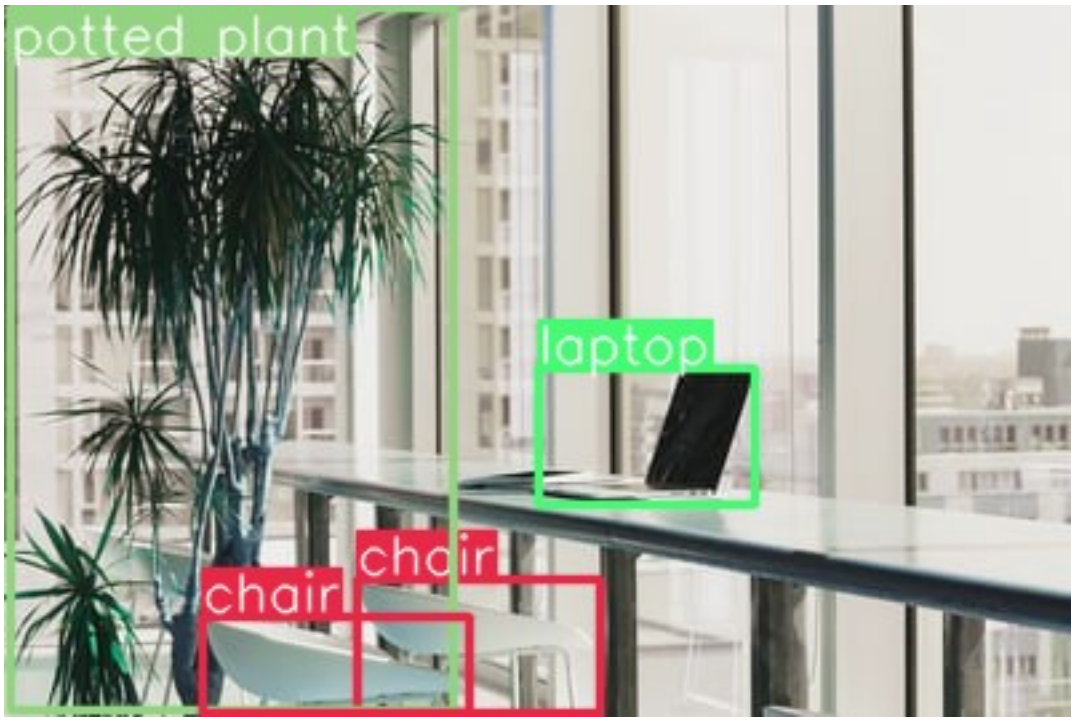



person
person


carrot
carrot
carrot
carrot
carrot
carrot
carrot
carrot
carrot
carrot
carrot
carrot


person
person
person
handbag


dining table
cake
cake
cake
cake
cake

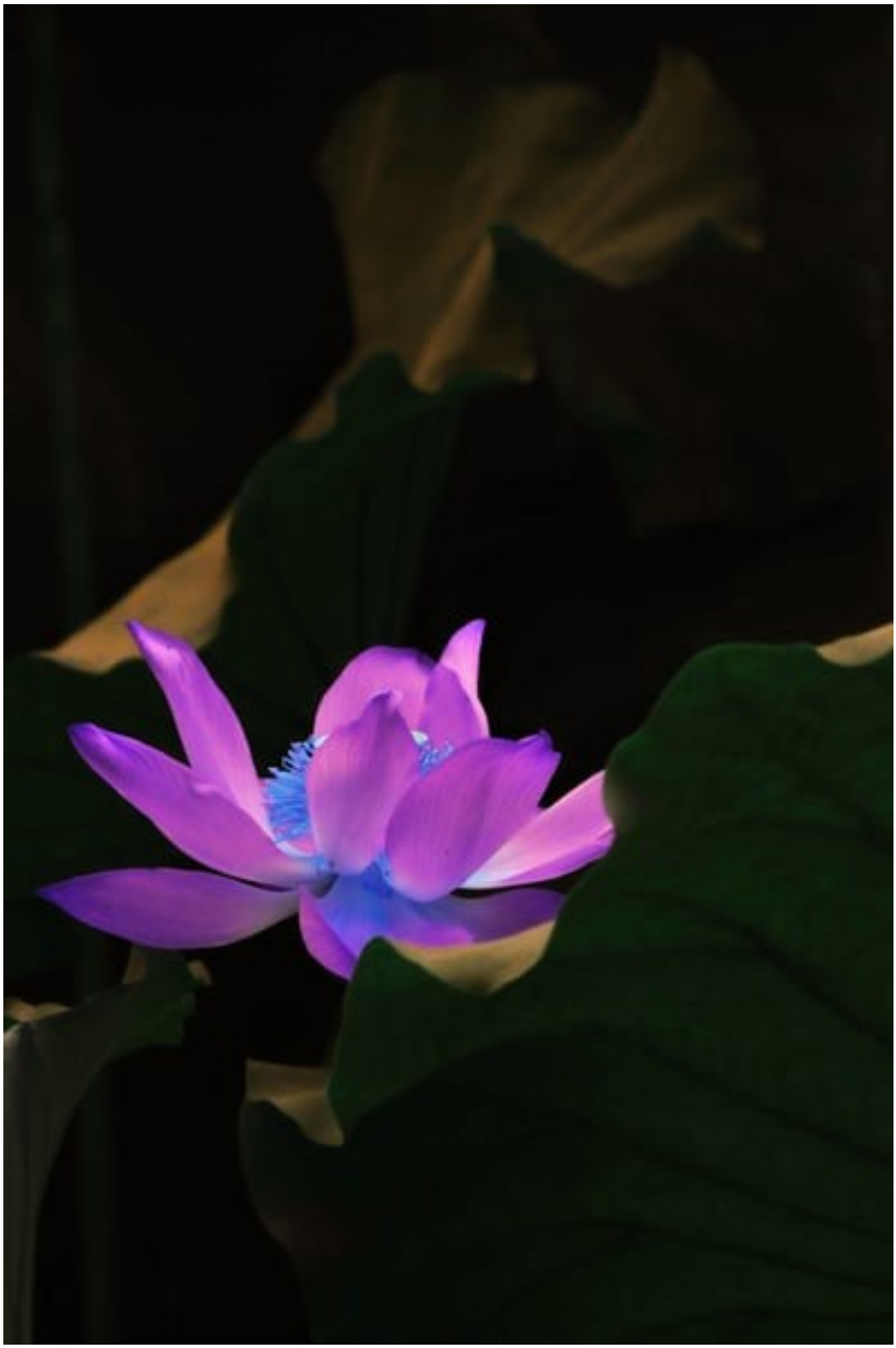

potted plant
laptop
chair
chair


THIS IS
THE SIGN
YOU'VE BEEN
LOOKING FOR


cake
cake


sports ball


giraffe



<Figure size 2000x2000 with 0 Axes>

Dataset3 (60)

```
# mount Google Drive
from google.colab import drive
drive.mount('/content/gdrive')
%cd gdrive/MyDrive
```

```
Drive already mounted at /content/gdrive; to attempt to forcibly
remount, call drive.mount("/content/gdrive", force_remount=True).
[Errno 2] No such file or directory: 'gdrive/MyDrive'
/content/gdrive/MyDrive
```

```
#loading the images

list_img3 = os.listdir("yolov7/images")
#list_img3 = list_img23[:50]
len(list_img3)
```

```
60
```

```
for i in range(len(list_img3)):
  list_img3[i]='yolov7/images/'+ list_img3[i]

np_img3 = []
for i in range(len(list_img3)):
  np_img3.append(cv2.imread(list_img3[i]))

%%time
results = modelv8.predict(np_img3)
```

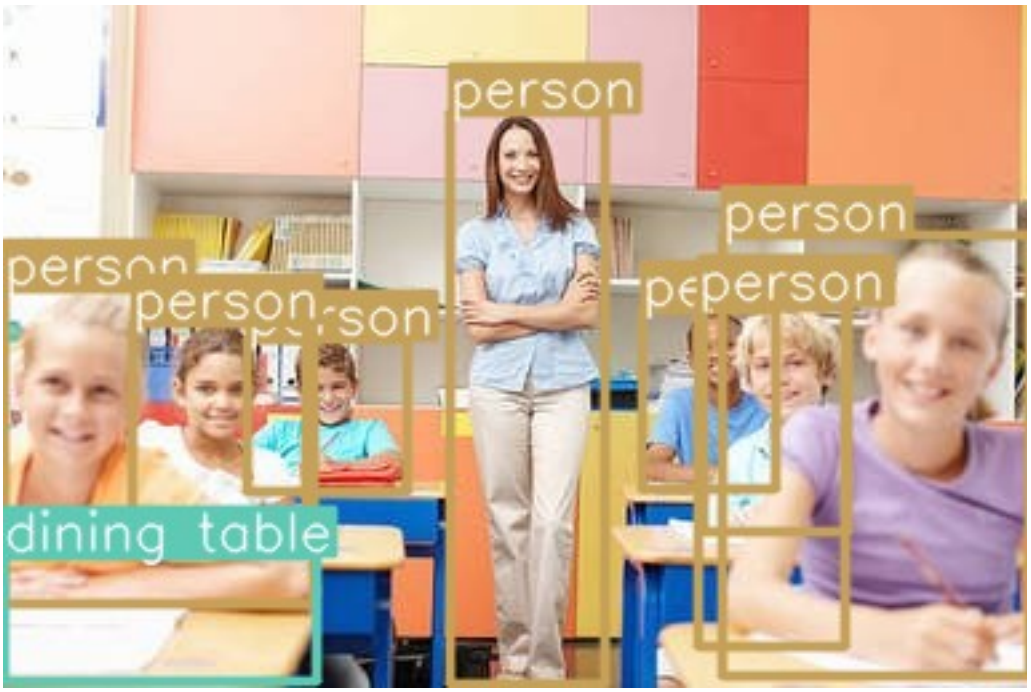
```
CPU times: user 13.3 s, sys: 298 ms, total: 13.6 s
Wall time: 13.6 s
```

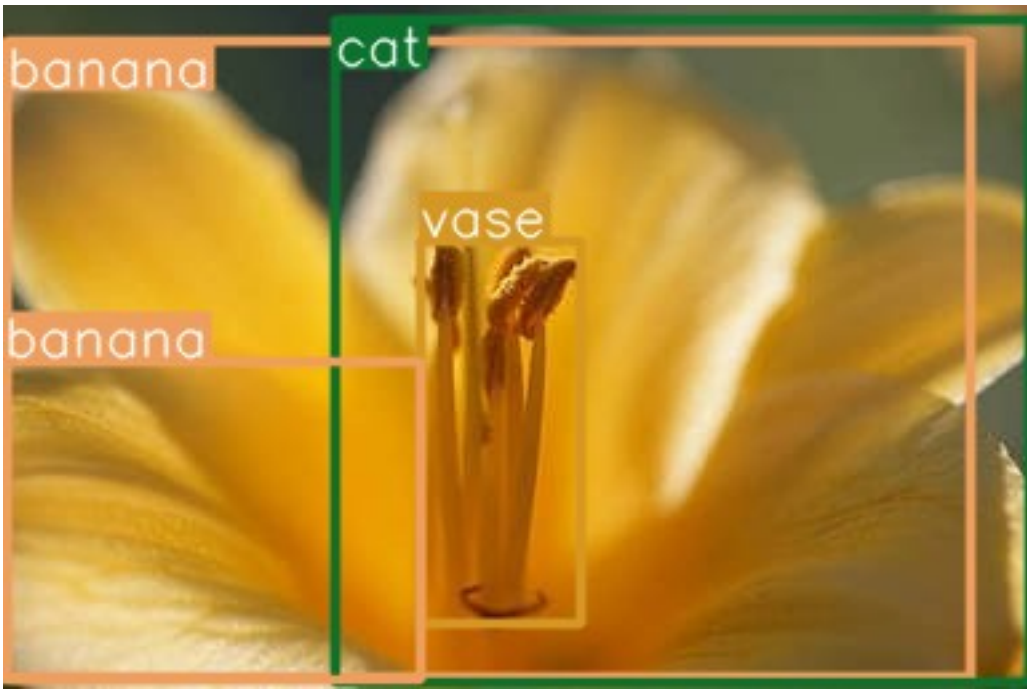
```
#Plotting the output of YOLOv8
%matplotlib inline
plt.figure(figsize=(10,10), dpi=200)
for i in range(len(list_img3)):
  plot_bboxes(np_img3[i], results[i].boxes.boxes, score=False)
```


dining table


wine glass


carrot


person
person
person
person
person
dining table

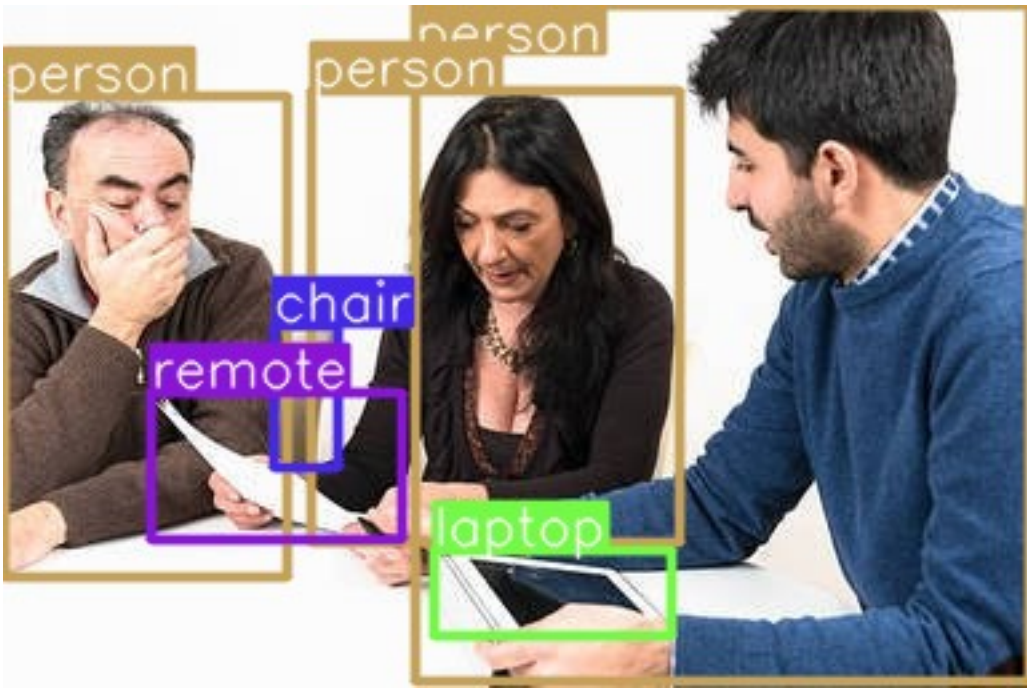

banana
cat
vase
banana


cakerke


bird
bird
bird
bird
bird
bird
bird
bird
bird
bird
bird


person
person

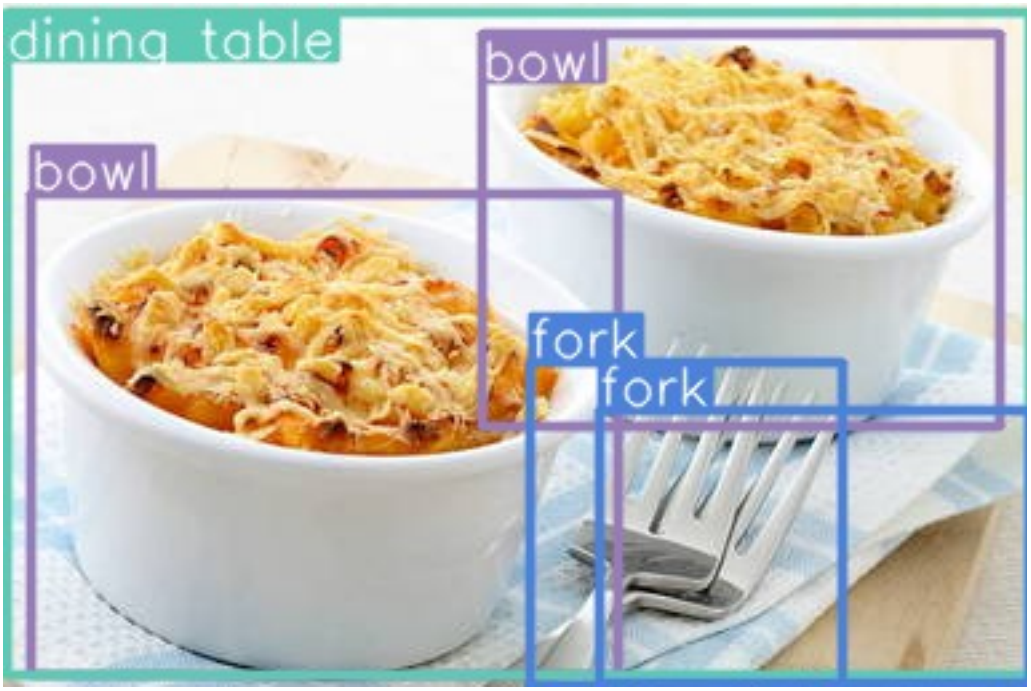

teddy bear


person
person
person
chair
remote
laptop


cat


dining table
orange

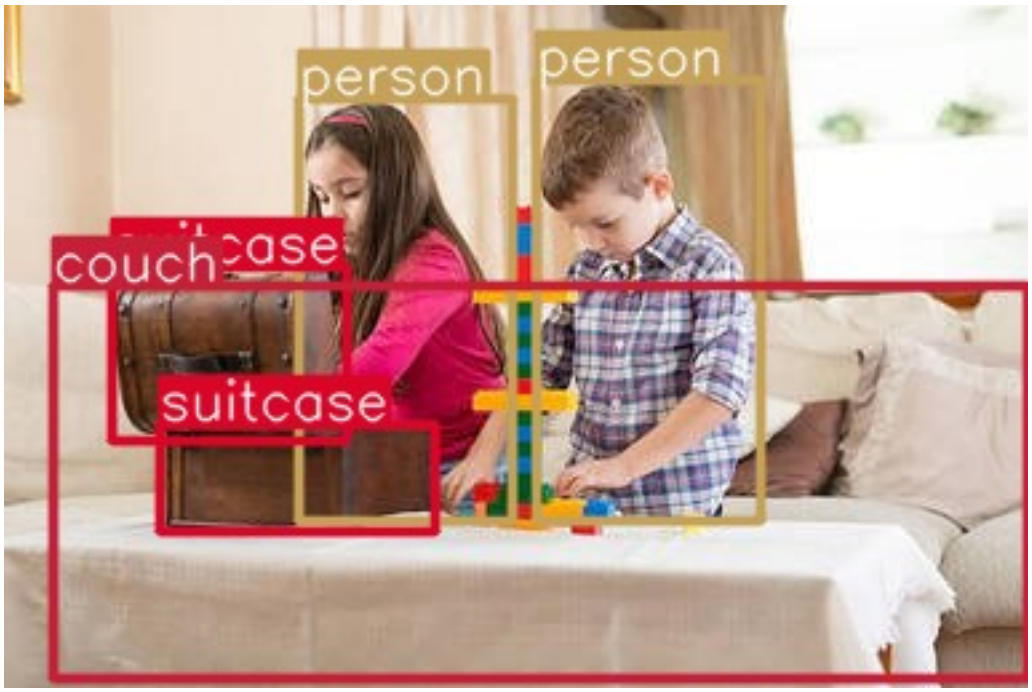

boat
boat


dining table
bowl
bowl
fork
fork


bowl
apple

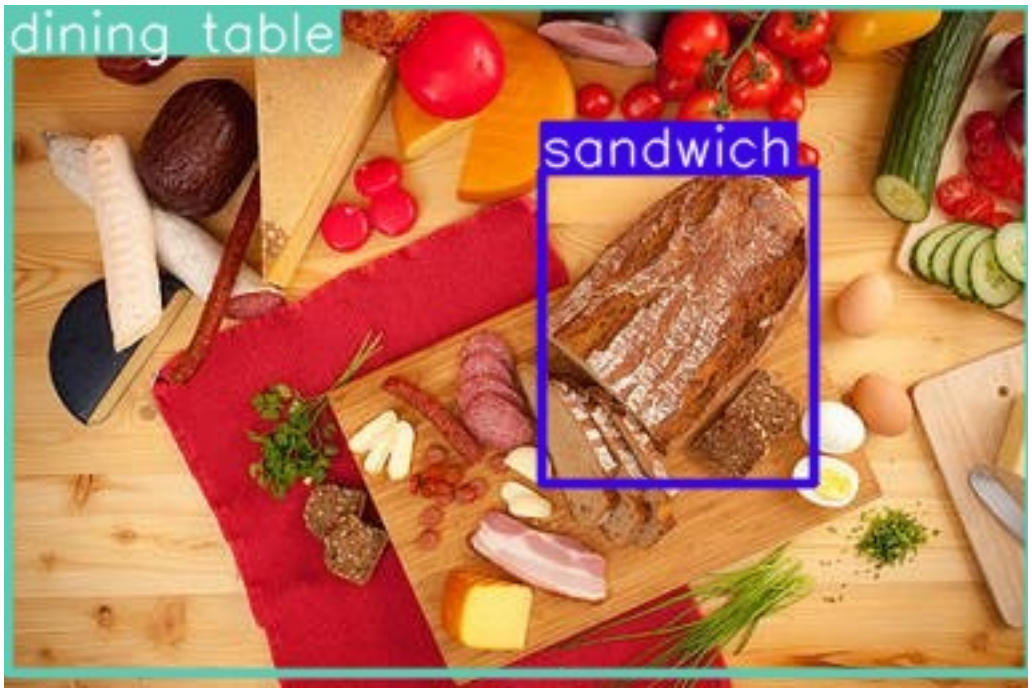

person
person
person

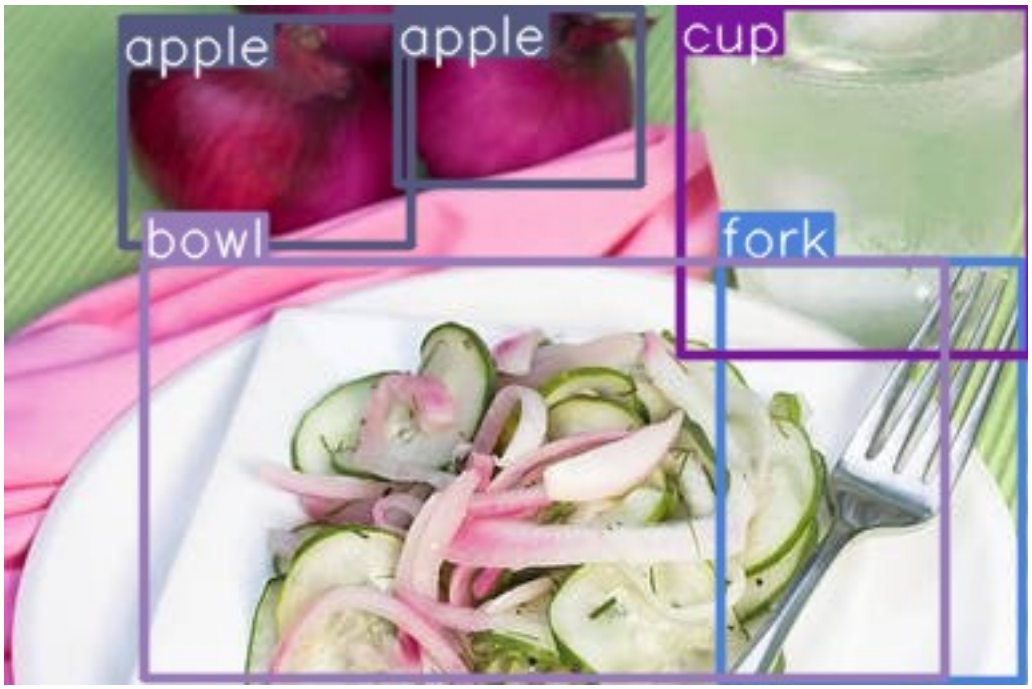

cup
bottle
bottle
cake
cake


person
person
couch
suitcase


person

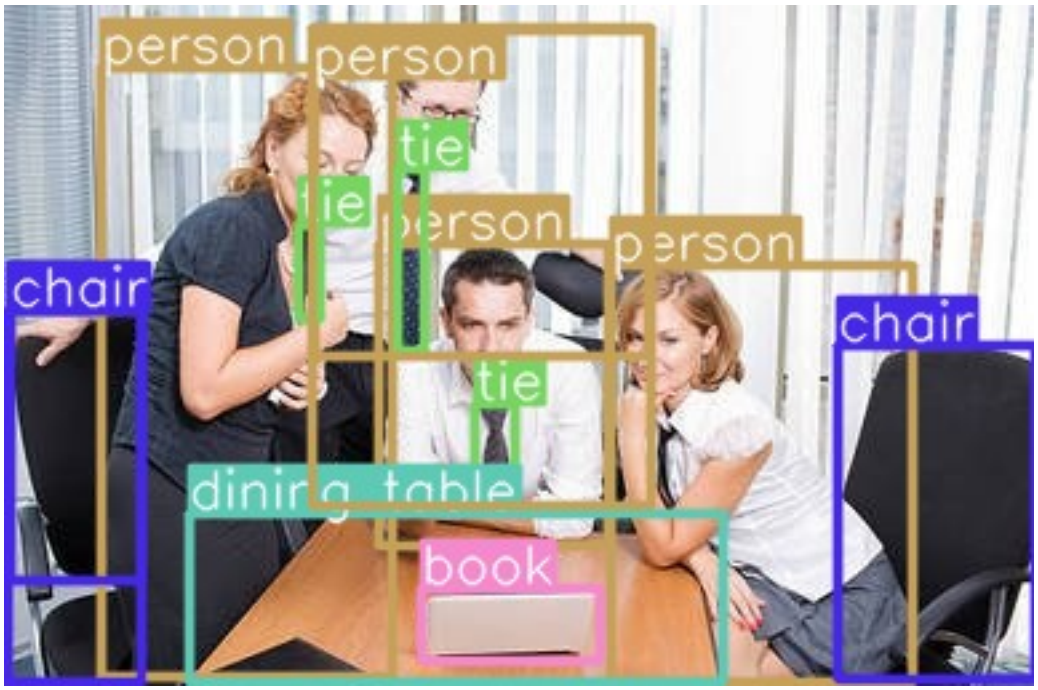

person
potted plant
potted plant

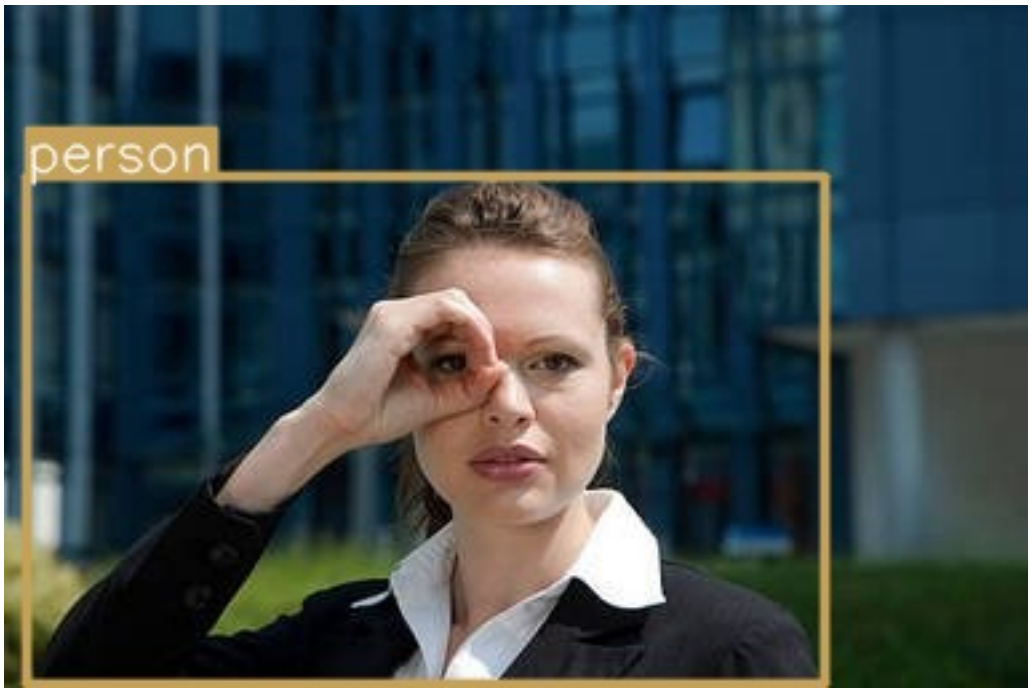

dining table
sandwich

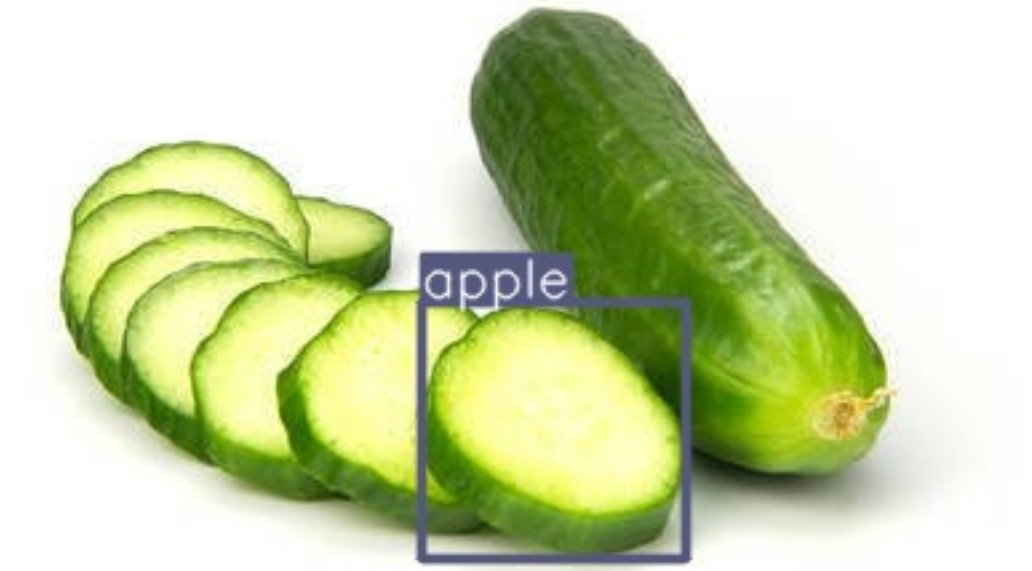

apple
apple
cup
bowl
fork


person


person


person
person
tie
tie
person
person
chair
chair
tie
book

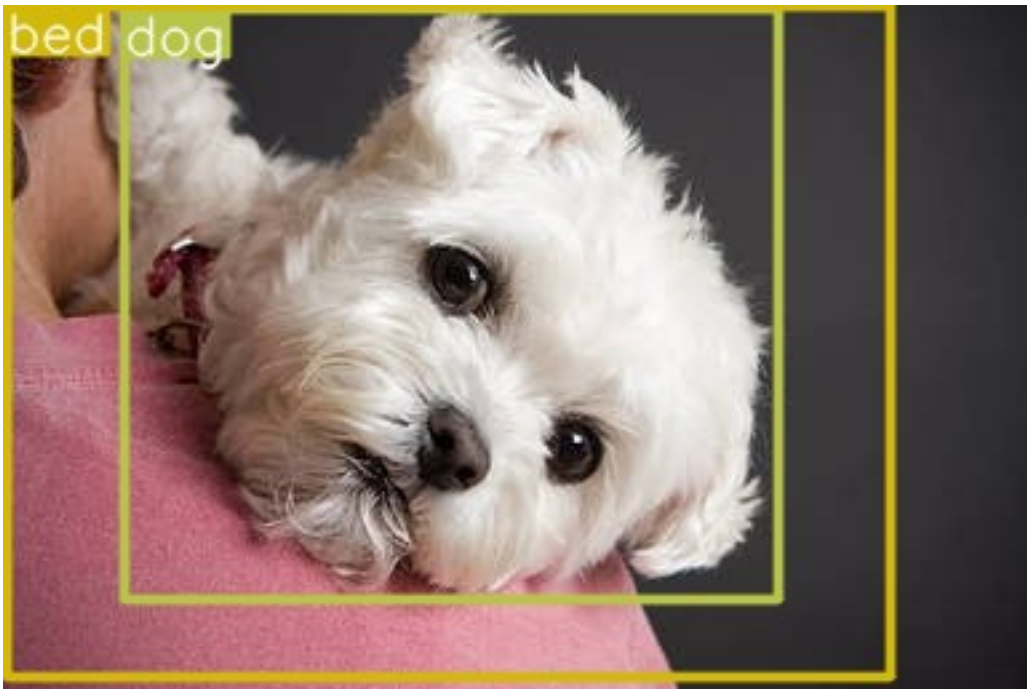

person


apple


dining table
cup


frisbee


person:on


bed
dog


orange


wine glass
dining table
dining table


person
person
person
person
person
person


car


car


car


car


person


clock
cake
cake
cake
cake
cake
cake
cake
cake
cake
cake


cup
carrot
carrot


person

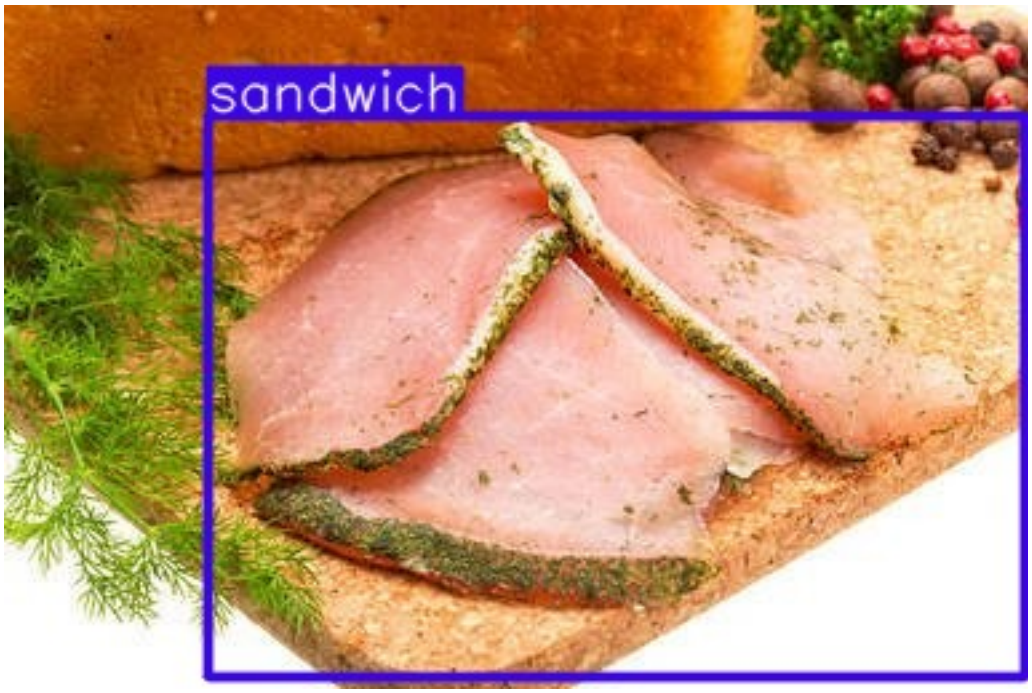

toothbrush


potted plant
banana


orarorange
eddy bear
apple


dining table
spo


cup
cup
cup
cup



<Figure size 2000x2000 with 0 Axes>