

```
!pip install ultralytics
```

```
353.7/353.7 MB 7.2 MB/s eta 0:00:00
Downloading nvidia_cuda_cupti_cu12-12.4.127-py3-none-manylinux2014_x86_64.whl (13.8 MB)
13.8/13.8 MB 77.8 MB/s eta 0:00:00
Downloading nvidia_cuda_nvrtc_cu12-12.4.127-py3-none-manylinux2014_x86_64.whl (24.6 MB)
24.6/24.6 MB 26.8 MB/s eta 0:00:00
Downloading nvidia_cuda_runtime_cu12-12.4.127-py3-none-manylinux2014_x86_64.whl (883 kB)
883.7/883.7 kB 42.3 MB/s eta 0:00:00
Downloading nvidia_cudnn_cu12-9.1.0.70-py3-none-manylinux2014_x86_64.whl (664.8 MB)
664.8/664.8 MB 2.8 MB/s eta 0:00:00
Downloading nvidia_cufft_cu12-11.2.1.3-py3-none-manylinux2014_x86_64.whl (211.5 MB)
211.5/211.5 MB 6.8 MB/s eta 0:00:00
Downloading nvidia_curand_cu12-10.3.5.147-py3-none-manylinux2014_x86_64.whl (56.3 MB)
56.3/56.3 MB 10.4 MB/s eta 0:00:00
Downloading nvidia_cusolver_cu12-11.6.1.9-py3-none-manylinux2014_x86_64.whl (127.9 MB)
127.9/127.9 MB 9.5 MB/s eta 0:00:00
Downloading nvidia_cusparses_cu12-12.3.1.170-py3-none-manylinux2014_x86_64.whl (207.5 MB)
207.5/207.5 MB 5.2 MB/s eta 0:00:00
Downloading nvidia_nvjitlink_cu12-12.4.127-py3-none-manylinux2014_x86_64.whl (21.1 MB)
21.1/21.1 MB 48.3 MB/s eta 0:00:00
Downloading ultralytics_thop-2.0.14-py3-none-any.whl (26 kB)
Installing collected packages: nvidia-nvjitlink-cu12, nvidia-curand-cu12, nvidia-cufft-cu12, nvidia-cuda-runtime-cu12, nvidia-cud
Attempting uninstall: nvidia-nvjitlink-cu12
Found existing installation: nvidia-nvjitlink-cu12 12.5.82
Uninstalling nvidia-nvjitlink-cu12-12.5.82:
Successfully uninstalled nvidia-nvjitlink-cu12-12.5.82
Attempting uninstall: nvidia-curand-cu12
Found existing installation: nvidia-curand-cu12 10.3.6.82
Uninstalling nvidia-curand-cu12-10.3.6.82:
Successfully uninstalled nvidia-curand-cu12-10.3.6.82
Attempting uninstall: nvidia-cufft-cu12
Found existing installation: nvidia-cufft-cu12 11.2.3.61
Uninstalling nvidia-cufft-cu12-11.2.3.61:
Successfully uninstalled nvidia-cufft-cu12-11.2.3.61
Attempting uninstall: nvidia-cuda-runtime-cu12
Found existing installation: nvidia-cuda-runtime-cu12 12.5.82
Uninstalling nvidia-cuda-runtime-cu12-12.5.82:
Successfully uninstalled nvidia-cuda-runtime-cu12-12.5.82
Attempting uninstall: nvidia-cuda-nvrtc-cu12
Found existing installation: nvidia-cuda-nvrtc-cu12 12.5.82
Uninstalling nvidia-cuda-nvrtc-cu12-12.5.82:
Successfully uninstalled nvidia-cuda-nvrtc-cu12-12.5.82
Attempting uninstall: nvidia-cuda-cupti-cu12
Found existing installation: nvidia-cuda-cupti-cu12 12.5.82
Uninstalling nvidia-cuda-cupti-cu12-12.5.82:
Successfully uninstalled nvidia-cuda-cupti-cu12-12.5.82
Attempting uninstall: nvidia-cublas-cu12
Found existing installation: nvidia-cublas-cu12 12.5.3.2
Uninstalling nvidia-cublas-cu12-12.5.3.2:
Successfully uninstalled nvidia-cublas-cu12-12.5.3.2
Attempting uninstall: nvidia-cusparses-cu12
Found existing installation: nvidia-cusparses-cu12 12.5.1.3
Uninstalling nvidia-cusparses-cu12-12.5.1.3:
Successfully uninstalled nvidia-cusparses-cu12-12.5.1.3
Attempting uninstall: nvidia-cudnn-cu12
Found existing installation: nvidia-cudnn-cu12 9.3.0.75
Uninstalling nvidia-cudnn-cu12-9.3.0.75:
Successfully uninstalled nvidia-cudnn-cu12-9.3.0.75
Attempting uninstall: nvidia-cusolver-cu12
```

```
from ultralytics import YOLO
model =YOLO("yolov8m.pt")
model.names
```



```

44: 'spoon',
45: 'bowl',
46: 'banana',
47: 'apple',
48: 'sandwich',
49: 'orange',
50: 'broccoli',
51: 'carrot',
52: 'hot dog',
53: 'pizza',
54: 'donut',
55: 'cake',
56: 'chair',
57: 'couch',
58: 'potted plant',
59: 'bed',
60: 'dining table',
61: 'toilet',
62: 'tv',
63: 'laptop',
64: 'mouse',
65: 'remote',
66: 'keyboard',
67: 'cell phone',
68: 'microwave',
69: 'oven',
70: 'toaster',
71: 'sink',
72: 'refrigerator',
73: 'book',
74: 'clock',
75: 'vase',
76: 'scissors',
77: 'teddy bear',

```

```

from PIL import Image
import requests

```

```

image_paths = [
    'https://c4.wallpaperflare.com/wallpaper/144/375/674/nba-images-pictures-wallpaper-preview.jpg',
    #'https://images.stockcake.com/public/5/0/a/50a6e95f-83bb-41a2-b04e-8f6938a0dcb0_large/bustling-city-traffic-stockcake.jpg',
    'https://www.shutterstock.com/image-photo/collage-different-cute-pets-260nw-206750137.jpg',
    'https://thumbs.dreamstime.com/b/colorful-collage-flowers-birds-showcased-to-celebrate-international-day-biological-diversity-ai-ger
]

```

```

results_list = []

```

```

for image_path in image_paths:
    try:
        img = Image.open(requests.get(image_path, stream=True).raw)
        results = model.predict(img, save=True, conf=0.2, iou=0.5)
        results_list.append(results)
    except Exception as e:
        print(f"Error processing {image_path}: {e}")

```



```

0: 416x640 8 persons, 1 baseball glove, 1 skateboard, 920.6ms
Speed: 24.4ms preprocess, 920.6ms inference, 30.7ms postprocess per image at shape (1, 3, 416, 640)
Results saved to runs/detect/predict

0: 640x608 10 cats, 10 dogs, 1993.3ms
Speed: 3.2ms preprocess, 1993.3ms inference, 1.2ms postprocess per image at shape (1, 3, 640, 608)
Results saved to runs/detect/predict

0: 320x640 4 birds, 849.1ms
Speed: 6.8ms preprocess, 849.1ms inference, 1.1ms postprocess per image at shape (1, 3, 320, 640)
Results saved to runs/detect/predict

```

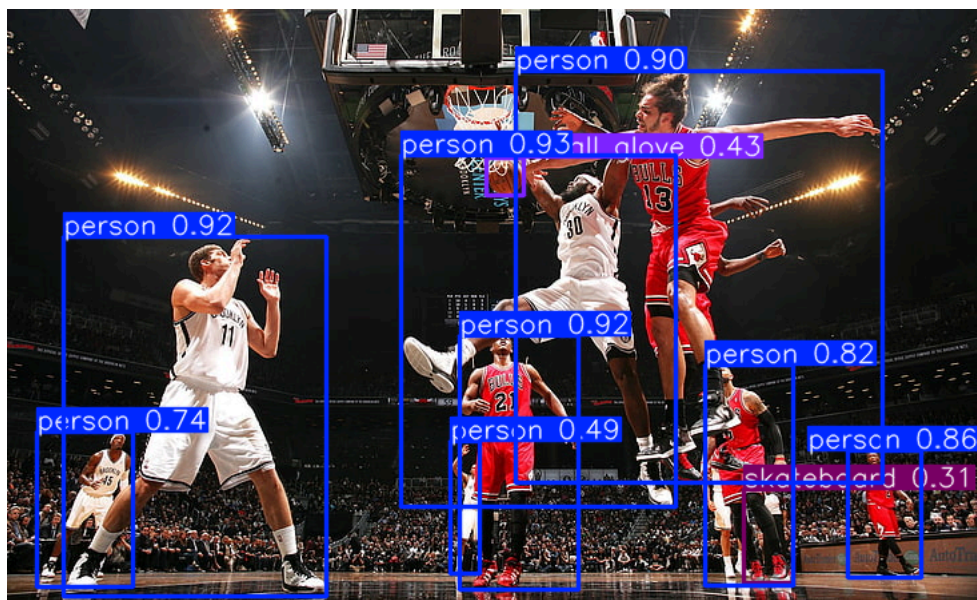
```

import cv2
for results in results_list:
    # Iterate through results for each image
    for result in results:
        # Iterate through detections within each image's results
        # Plot the detections for the current result (image)
        plot = result.plot()

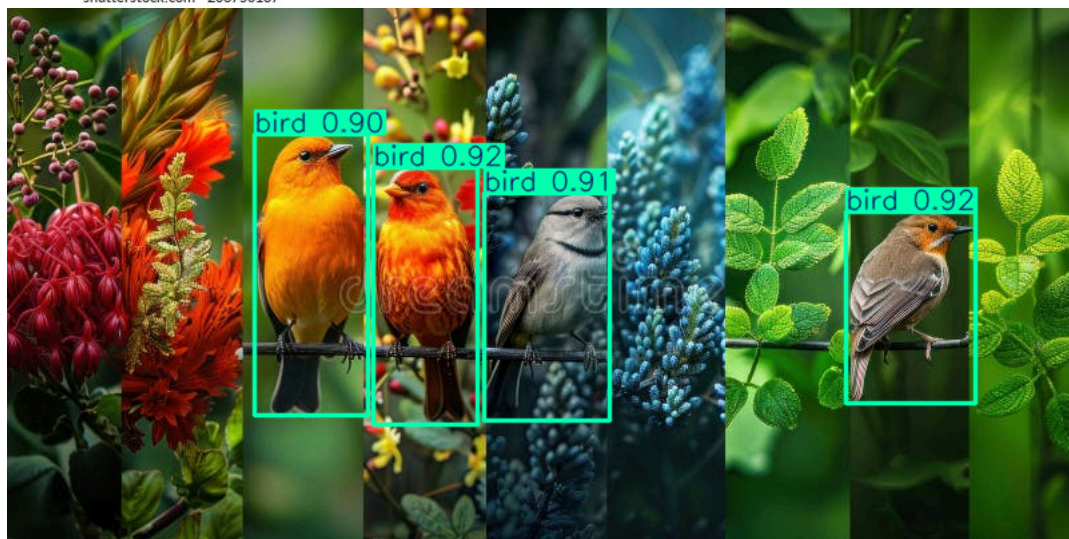
        # Convert the plot to RGB color space
        plot = cv2.cvtColor(plot, cv2.COLOR_BGR2RGB)

        # Display the image with detections
        display(Image.fromarray(plot))

```



shutterstock.com · 206750137



```
from ultralytics import YOLO, hub

# Load a pretrained model from the Ultralytics hub
segmentation_model = YOLO("yolov8m-seg")

# Now you can use the model
segmentation_model.names
```



```

35: 'baseball glove',
36: 'skateboard',
37: 'surfboard',
38: 'tennis racket',
39: 'bottle',
40: 'wine glass',
41: 'cup',
42: 'fork',
43: 'knife',
44: 'spoon',
45: 'bowl',
46: 'banana',
47: 'apple',
48: 'sandwich',
49: 'orange',
50: 'broccoli',
51: 'carrot',
52: 'hot dog',
53: 'pizza',
54: 'donut',
55: 'cake',
56: 'chair',
57: 'couch',
58: 'potted plant',
59: 'bed',
60: 'dining table',
61: 'toilet',
62: 'tv',
63: 'laptop',
64: 'mouse',
65: 'remote',
66: 'keyboard',
67: 'cell phone',
68: 'microwave',
69: 'oven',
70: 'toaster',
71: 'sink',
72: 'refrigerator',
73: 'book',
74: 'clock',
75: 'vase',
76: 'scissors',
77: 'teddy bear',
-- ..

```

```

# prompt: Perform instance segmentation on different images collected and display the results.
# Display and save the segmented image

```

```

from ultralytics import YOLO
from PIL import Image
import requests
import cv2
from IPython.display import display

```

```

# Assuming you have already installed ultralytics and other required packages

```

```

image_paths = [
    'https://c4.wallpaperflare.com/wallpaper/144/375/674/nba-images-pictures-wallpaper-preview.jpg',
    'https://images.stockcake.com/public/5/0/a/50a6e95f-83bb-41a2-b04e-8f6938a0dcb0_large/bustling-city-traffic-stockcake.jpg',
    'https://www.shutterstock.com/image-photo/collage-different-cute-pets-260nw-206750137.jpg',
    'https://thumbs.dreamstime.com/b/colorful-collage-flowers-birds-showcased-to-celebrate-international-day-biological-diversity-ai-ger
]
results_list = []

for image_path in image_paths:
    try:
        img = Image.open(requests.get(image_path, stream=True).raw)
        results = segmentation_model.predict(img, save=True, conf=0.2, iou=0.5, imgsz=640) # Added imgsz for better performance
        results_list.append(results)
    except Exception as e:
        print(f"Error processing {image_path}: {e}")

for results in results_list:
    for result in results:
        plotted_img = result.plot()
        plotted_img = cv2.cvtColor(plotted_img, cv2.COLOR_BGR2RGB)
        display(Image.fromarray(plotted_img))

        # Save the segmented image
        segmented_image_path = f"segmented_image_{image_paths.index(image_path)}.jpg" # Unique file name
        cv2.imwrite(segmented_image_path, cv2.cvtColor(plotted_img, cv2.COLOR_RGB2BGR)) # Convert back to BGR for saving
        print(f"Saved segmented image to {segmented_image_path}")

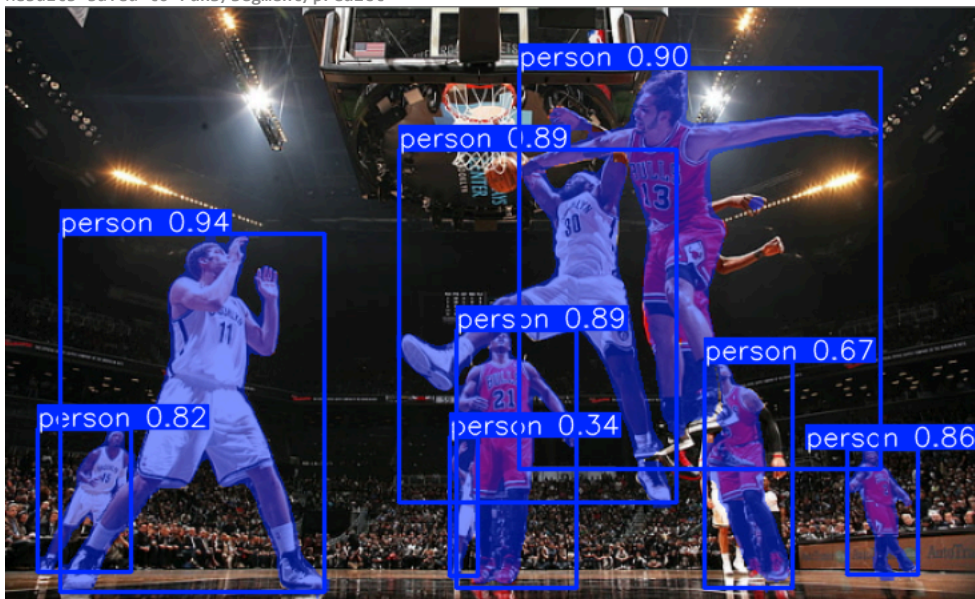
```



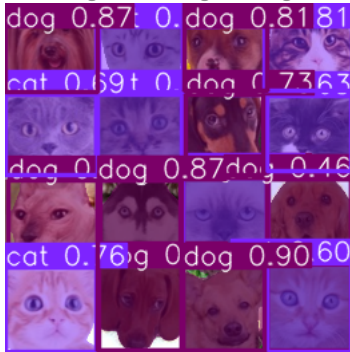

0: 416x640 8 persons, 976.1ms
Speed: 1.4ms preprocess, 976.1ms inference, 17.6ms postprocess per image at shape (1, 3, 416, 640)
Results saved to runs/segment/predict

0: 640x608 9 cats, 11 dogs, 1397.1ms
Speed: 1.8ms preprocess, 1397.1ms inference, 71.4ms postprocess per image at shape (1, 3, 640, 608)
Results saved to runs/segment/predict

0: 320x640 4 birds, 1 potted plant, 754.3ms
Speed: 1.3ms preprocess, 754.3ms inference, 6.7ms postprocess per image at shape (1, 3, 320, 640)
Results saved to runs/segment/predict



Saved segmented image to segmented_image_2.jpg



shutterstock.com • 206750137

Saved segmented image to segmented_image_2.jpg



Saved segmented image to segmented_image_2.jpg