```
!pip install -U -q ultralytics roboflow wandb opencv-python-headless matplotlib
```

```
import torch
print("Torch CUDA available:", torch.cuda.is_available(), "cuda:", torch.version.cuda)
```

```
Torch CUDA available: True cuda: 12.6
```

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/driv
```

```
!mkdir -p /content/drive/MyDrive/YOLOv8_surgical
```

```
!pip install roboflow --quiet

from roboflow import Roboflow
rf = Roboflow(api_key="PFN4A7FKw5wg0Uw62akT")
project = rf.workspace("pattern-recognition-and-computer-vision").project("surgical-tool-detection-yjf
version = project.version(5)
dataset_dir = version.download("yolov8")
print("Downloaded to:", dataset_dir)
```

```
loading Roboflow workspace...
loading Roboflow project...
Downloading Dataset Version Zip in Surgical-Tool-Detection--5 to yolov8:: 100%|██████████| 65382/65382

Extracting Dataset Version Zip to Surgical-Tool-Detection--5 in yolov8:: 100%|██████████| 4232/4232 [00
View Ultralytics Settings with 'yolo settings' or at '/root/.config/Ultralytics/settings.json'
Update Settings with 'yolo settings key=value', i.e. 'yolo settings runs_dir=path/to/dir'. For help see
Downloaded to: <roboflow.core.dataset.Dataset object at 0x7ea6bb6fbe30>
```

```
import os, random, matplotlib.pyplot as plt, cv2
```

```
base = dataset_dir.location
print("Base folder listing:", os.listdir("/content"))
print("Dataset folder listing:", os.listdir(base))
```

```
Base folder listing: ['.config', 'Surgical-Tool-Detection--5', 'drive', 'sample_data']
Dataset folder listing: ['README.roboflow.txt', 'test', 'README.dataset.txt', 'data.yaml', 'valid', 'tra
```

```
for split in ["train","valid","test"]:
    p = os.path.join(base, split, "images")
    if os.path.exists(p):
        print(split, "images:", len(os.listdir(p)))

img_path = random.choice([os.path.join(base, 'train', 'images', f) for f in os.listdir(os.path.join(base
img = cv2.cvtColor(cv2.imread(img_path), cv2.COLOR_BGR2RGB)
plt.figure(figsize=(8,6))
plt.imshow(img)
plt.axis('off')
```

```
train images: 1478
valid images: 421
test images: 211
(np.float64(-0.5), np.float64(639.5), np.float64(479.5), np.float64(-0.5))
```

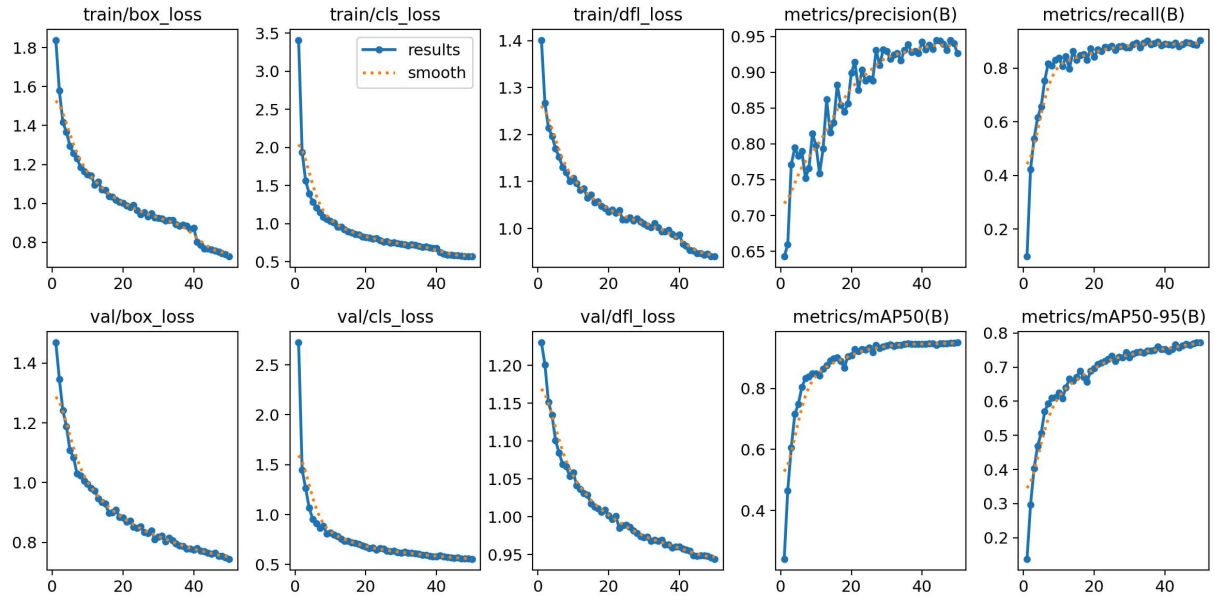
```
from ultralytics import YOLO

# Load model
model = YOLO("yolov8n.pt")

# Train
results = model.train(
    data=os.path.join(dataset_dir.location, "data.yaml"),
    epochs=50,
    imgsz=640,
    batch=16,
    project="/content/drive/MyDrive/YOLOv8_surgical",
    name="exp_yolov8n_50",
    save=True,
    device=0
)
```

```
Ultralytics 8.3.225 🚀 Python-3.12.12 torch-2.8.0+cu126 CUDA:0 (Tesla T4, 15095MiB)
Model summary (fused): 72 layers, 3,008,378 parameters, 0 gradients, 8.1 GFLOPs
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% ━━━━━━━━━━━━
                   all        421       4072      0.926      0.905       0.95      0.773
        artery_forceps        252        551      0.959      0.893       0.96      0.762
             aspirator          7          7      0.862          1      0.995      0.794
          bending_shear        188        220      0.925      0.718      0.891      0.694
         circular_spoon         50         50       0.98       0.86       0.93      0.785
            core_needle         17         17      0.838          1      0.971      0.915
            fine_needle         20         20          1      0.938       0.99      0.782
           iris_scissors         60         77      0.865      0.935      0.922      0.664
      operating_scissors         53         54      0.887      0.944      0.927      0.686
       rongeur_forceps_1        302        569      0.952      0.939      0.976      0.875
       rongeur_forceps_2         70         70      0.953      0.986      0.986      0.851
                 scalpel        315        570      0.918      0.826      0.912      0.711
              stripping         287        287      0.934      0.891      0.946      0.794
                tweezers        320        930      0.924      0.835      0.941      0.674
    wire_grabbing_pliers        302        650      0.964      0.904      0.958      0.837
Speed: 0.2ms preprocess, 2.0ms inference, 0.0ms loss, 3.9ms postprocess per image
Results saved to /content/drive/MyDrive/YOLOv8_surgical/exp_yolov8n_502
```

```
import glob, os
run_dir = sorted(glob.glob("/content/drive/MyDrive/YOLOv8_surgical/exp_yolov8n_50*"))[-1]
print("Run dir:", run_dir)
```

```
Run dir: /content/drive/MyDrive/YOLOv8_surgical/exp_yolov8n_502
```

```
from PIL import Image, ImageOps
img = Image.open(os.path.join(run_dir, "results.png"))
display(img)
```



Inference

```
from ultralytics import YOLO
model = YOLO(os.path.join(run_dir, "weights", "best.pt"))

val_img = dataset_dir.location + "/valid/images"
out_folder = "/content/predictions"
!mkdir -p {out_folder}
results = model.predict(source=val_img, save=True, save_dir=out_folder, conf=0.35, imgsz=640)
print("Saved predictions to", out_folder)
```

Show hidden output

```
!yolo export model={run_dir}/weights/best.pt format=onnx
```

```
Ultralytics 8.3.225 🚀 Python-3.12.12 torch-2.8.0+cu126 CPU (Intel Xeon CPU @ 2.00GHz)
💡 ProTip: Export to OpenVINO format for best performance on Intel hardware. Learn more at https://docs.
Model summary (fused): 72 layers, 3,008,378 parameters, 0 gradients, 8.1 GFLOPs

PyTorch: starting from '/content/drive/MyDrive/YOLOv8_surgical/exp_yolov8n_502/weights/best.pt' with inp
requirements: Ultralytics requirements ['onnx>=1.12.0', 'onnxslim>=0.1.71', 'onnxruntime'] not found, at

requirements: AutoUpdate success ✅ 3.0s
WARNING ⚠️ requirements: Restart runtime or rerun command for updates to take effect


ONNX: starting export with onnx 1.20.0rc1 opset 22...
WARNING ⚠️ ONNX: simplifier failure: module 'onnx.helper' has no attribute 'float32_to_bfloat16'
ONNX: export success ✅ 4.3s, saved as '/content/drive/MyDrive/YOLOv8_surgical/exp_yolov8n_502/weights/

Export complete (4.7s)
Results saved to /content/drive/MyDrive/YOLOv8_surgical/exp_yolov8n_502/weights
Predict:         yolo predict task=detect model=/content/drive/MyDrive/YOLOv8_surgical/exp_yolov8n_502/w
Validate:        yolo val task=detect model=/content/drive/MyDrive/YOLOv8_surgical/exp_yolov8n_502/weigh
Visualize:       https://netron.app
💡 Learn more at https://docs.ultralytics.com/modes/export
```

```
pred_img_path = random.choice([os.path.join('/content/runs/detect/predict/'+ img) for img in os.listdir(
pred_img = cv2.cvtColor(cv2.imread(pred_img_path), cv2.COLOR_BGR2RGB)
plt.figure(figsize=(8,6))
plt.imshow(pred_img)
plt.axis('off')
```

```
(np.float64(-0.5), np.float64(639.5), np.float64(479.5), np.float64(-0.5))
```