

```
%matplotlib inline  
import matplotlib.pyplot as plt
```

```
!pip install kaggle  
!mkdir -p ~/.kaggle  
!mv kaggle.json ~/.kaggle/  
!chmod 600 ~/.kaggle/kaggle.json
```

```
Requirement already satisfied: kaggle in /usr/local/lib/python3.12/dist-packages (1.7.4.5)  
Requirement already satisfied: bleach in /usr/local/lib/python3.12/dist-packages (from kaggle) (6.3.0)  
Requirement already satisfied: certifi>=14.05.14 in /usr/local/lib/python3.12/dist-packages (from kaggle) (2025.10.5)  
Requirement already satisfied: charset-normalizer in /usr/local/lib/python3.12/dist-packages (from kaggle) (3.4.4)  
Requirement already satisfied: idna in /usr/local/lib/python3.12/dist-packages (from kaggle) (3.7)  
Requirement already satisfied: protobuf in /usr/local/lib/python3.12/dist-packages (from kaggle) (5.29.5)  
Requirement already satisfied: python-dateutil>=2.5.3 in /usr/local/lib/python3.12/dist-packages (from kaggle) (2.9.0.post0)  
Requirement already satisfied: python-slugify in /usr/local/lib/python3.12/dist-packages (from kaggle) (8.0.4)  
Requirement already satisfied: requests in /usr/local/lib/python3.12/dist-packages (from kaggle) (2.32.4)  
Requirement already satisfied: setuptools>=21.0.0 in /usr/local/lib/python3.12/dist-packages (from kaggle) (75.2.0)  
Requirement already satisfied: six>=1.10 in /usr/local/lib/python3.12/dist-packages (from kaggle) (1.17.0)  
Requirement already satisfied: text-unidecode in /usr/local/lib/python3.12/dist-packages (from kaggle) (1.3)  
Requirement already satisfied: tqdm in /usr/local/lib/python3.12/dist-packages (from kaggle) (4.67.1)  
Requirement already satisfied: urllib3>=1.15.1 in /usr/local/lib/python3.12/dist-packages (from kaggle) (2.5.0)  
Requirement already satisfied: webencodings in /usr/local/lib/python3.12/dist-packages (from kaggle) (0.5.1)  
mv: cannot stat 'kaggle.json': No such file or directory
```

```
!kaggle datasets download -d rumaisa90/suturing-videos  
!unzip suturing-videos.zip -d /content/suturing_dataset
```

```
Dataset URL: https://www.kaggle.com/datasets/rumaisa90/suturing-videos  
License(s): unknown  
Downloading suturing-videos.zip to /content/yolov5  
  0% 0.00/29.4M [00:00<?, ?B/s]  
100% 29.4M/29.4M [00:00<0:00, 1.70GB/s]  
Archive: suturing-videos.zip  
  inflating: /content/suturing_dataset/data/labels.txt  
  inflating: /content/suturing_dataset/data/videos/Suturing_B001.avi  
  inflating: /content/suturing_dataset/data/videos/Suturing_B002.avi  
  inflating: /content/suturing_dataset/data/videos/Suturing_B003.avi  
  inflating: /content/suturing_dataset/data/videos/Suturing_B004.avi  
  inflating: /content/suturing_dataset/data/videos/Suturing_B005.avi  
  inflating: /content/suturing_dataset/data/videos/Suturing_C001.avi  
  inflating: /content/suturing_dataset/data/videos/Suturing_C002.avi  
  inflating: /content/suturing_dataset/data/videos/Suturing_C003.avi  
  inflating: /content/suturing_dataset/data/videos/Suturing_C004.avi  
  inflating: /content/suturing_dataset/data/videos/Suturing_C005.avi  
  inflating: /content/suturing_dataset/data/videos/Suturing_D001.avi  
  inflating: /content/suturing_dataset/data/videos/Suturing_D002.avi  
  inflating: /content/suturing_dataset/data/videos/Suturing_D003.avi  
  inflating: /content/suturing_dataset/data/videos/Suturing_D004.avi  
  inflating: /content/suturing_dataset/data/videos/Suturing_D005.avi  
  inflating: /content/suturing_dataset/data/videos/Suturing_E001.avi  
  inflating: /content/suturing_dataset/data/videos/Suturing_E002.avi  
  inflating: /content/suturing_dataset/data/videos/Suturing_E003.avi  
  inflating: /content/suturing_dataset/data/videos/Suturing_E004.avi  
  inflating: /content/suturing_dataset/data/videos/Suturing_E005.avi
```

✓ getting the frames

```
!pip install roboflow  
  
from roboflow import Roboflow  
rf = Roboflow(api_key="bDXdHPqyqagn4ZLczX8n")  
project = rf.workspace("starters-3jatr").project("surgical_tool_suturing-jxmyh")  
version = project.version(2)  
dataset = version.download("yolov5")
```

```
Requirement already satisfied: roboflow in /usr/local/lib/python3.12/dist-packages (1.2.11)  
Requirement already satisfied: certifi in /usr/local/lib/python3.12/dist-packages (from roboflow) (2025.10.5)  
Requirement already satisfied: idna==3.7 in /usr/local/lib/python3.12/dist-packages (from roboflow) (3.7)  
Requirement already satisfied: cycler in /usr/local/lib/python3.12/dist-packages (from roboflow) (0.12.1)  
Requirement already satisfied: kiwisolver>=1.3.1 in /usr/local/lib/python3.12/dist-packages (from roboflow) (1.4.9)  
Requirement already satisfied: matplotlib in /usr/local/lib/python3.12/dist-packages (from roboflow) (3.10.0)
```

```
Requirement already satisfied: numpy>=1.18.5 in /usr/local/lib/python3.12/dist-packages (from roboflow) (2.0.2)
Requirement already satisfied: opencv-python-headless==4.10.0.84 in /usr/local/lib/python3.12/dist-packages (from roboflow) (4.1)
Requirement already satisfied: Pillow>7.1.2 in /usr/local/lib/python3.12/dist-packages (from roboflow) (11.3.0)
Requirement already satisfied: pi-heif<2 in /usr/local/lib/python3.12/dist-packages (from roboflow) (1.1.1)
Requirement already satisfied: pillow-avif-plugin<2 in /usr/local/lib/python3.12/dist-packages (from roboflow) (1.5.2)
Requirement already satisfied: python-dateutil in /usr/local/lib/python3.12/dist-packages (from roboflow) (2.9.0.post0)
Requirement already satisfied: python-dotenv in /usr/local/lib/python3.12/dist-packages (from roboflow) (1.2.1)
Requirement already satisfied: requests in /usr/local/lib/python3.12/dist-packages (from roboflow) (2.32.4)
Requirement already satisfied: six in /usr/local/lib/python3.12/dist-packages (from roboflow) (1.17.0)
Requirement already satisfied: urllib3>=1.26.6 in /usr/local/lib/python3.12/dist-packages (from roboflow) (2.5.0)
Requirement already satisfied: tqdm>=4.41.0 in /usr/local/lib/python3.12/dist-packages (from roboflow) (4.67.1)
Requirement already satisfied: PyYAML>=5.3.1 in /usr/local/lib/python3.12/dist-packages (from roboflow) (6.0.3)
Requirement already satisfied: requests-toolbelt in /usr/local/lib/python3.12/dist-packages (from roboflow) (1.0.0)
Requirement already satisfied: filetype in /usr/local/lib/python3.12/dist-packages (from roboflow) (1.2.0)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.12/dist-packages (from matplotlib->roboflow) (1.3.3)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.12/dist-packages (from matplotlib->roboflow) (4.60.1)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.12/dist-packages (from matplotlib->roboflow) (25.0)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.12/dist-packages (from matplotlib->roboflow) (3.2.5)
Requirement already satisfied: charset_normalizer<4,>=2 in /usr/local/lib/python3.12/dist-packages (from requests->roboflow) (3. loading Roboflow workspace...
loading Roboflow project...
```

```
# Install YOLOv8 and other dependencies
import matplotlib.pyplot as plt
```

```
import os
import cv2
import numpy as np
import torch
import pandas as pd
from tqdm import tqdm
import matplotlib.pyplot as plt
from scipy.stats import spearmanr
```

```
!git clone https://github.com/ultralytics/yolov5
%cd yolov5
%pip install -r requirements.txt
```

```
Requirement already satisfied: matplotlib>=3.3 in /usr/local/lib/python3.12/dist-packages (from -r requirements.txt (line 6))
Requirement already satisfied: numpy>=1.23.5 in /usr/local/lib/python3.12/dist-packages (from -r requirements.txt (line 7)) (2
Requirement already satisfied: opencv-python>=4.1.1 in /usr/local/lib/python3.12/dist-packages (from -r requirements.txt (line
Requirement already satisfied: pillow>10.3.0 in /usr/local/lib/python3.12/dist-packages (from -r requirements.txt (line 9)) (: Requirement already satisfied: psutil in /usr/local/lib/python3.12/dist-packages (from -r requirements.txt (line 10)) (5.9.5)
Requirement already satisfied: PyYAML>=5.3.1 in /usr/local/lib/python3.12/dist-packages (from -r requirements.txt (line 11)) (: Requirement already satisfied: requests>=2.32.2 in /usr/local/lib/python3.12/dist-packages (from -r requirements.txt (line 12)
Requirement already satisfied: scipy>=1.4.1 in /usr/local/lib/python3.12/dist-packages (from -r requirements.txt (line 13)) (1
Requirement already satisfied: thop>=0.1.1 in /usr/local/lib/python3.12/dist-packages (from -r requirements.txt (line 14)) (0.:
Requirement already satisfied: torch>=1.8.0 in /usr/local/lib/python3.12/dist-packages (from -r requirements.txt (line 15)) (2
Requirement already satisfied: torchvision>=0.9.0 in /usr/local/lib/python3.12/dist-packages (from -r requirements.txt (line 16
Requirement already satisfied: tqdm>=4.66.3 in /usr/local/lib/python3.12/dist-packages (from -r requirements.txt (line 17)) (4
Requirement already satisfied: ultralytics>=8.2.64 in /usr/local/lib/python3.12/dist-packages (from -r requirements.txt (line :
Requirement already satisfied: pandas>=1.1.4 in /usr/local/lib/python3.12/dist-packages (from -r requirements.txt (line 27)) (: Requirement already satisfied: seaborn>0.11.0 in /usr/local/lib/python3.12/dist-packages (from -r requirements.txt (line 28))
Requirement already satisfied: packaging in /usr/local/lib/python3.12/dist-packages (from -r requirements.txt (line 42)) (25.0
Requirement already satisfied: setuptools>=70.0.0 in /usr/local/lib/python3.12/dist-packages (from -r requirements.txt (line 4:
Requirement already satisfied: urllib3>=2.5.0 in /usr/local/lib/python3.12/dist-packages (from -r requirements.txt (line 51))
Requirement already satisfied: gitdb<5,>=4.0.1 in /usr/local/lib/python3.12/dist-packages (from gitpython>=3.1.30->-r requireme
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.12/dist-packages (from matplotlib>=3.3->-r requireme
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.12/dist-packages (from matplotlib>=3.3->-r requirements.i
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.12/dist-packages (from matplotlib>=3.3->-r requireme
Requirement already satisfied: kiwisolver>=1.3.1 in /usr/local/lib/python3.12/dist-packages (from matplotlib>=3.3->-r requireme
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.12/dist-packages (from matplotlib>=3.3->-r requireme
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.12/dist-packages (from matplotlib>=3.3->-r requi
Requirement already satisfied: charset_normalizer<4,>=2 in /usr/local/lib/python3.12/dist-packages (from requests>=2.32.2->-r Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.12/dist-packages (from requests>=2.32.2->-r requirements
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.12/dist-packages (from requests>=2.32.2->-r require
Requirement already satisfied: filelock in /usr/local/lib/python3.12/dist-packages (from torch>=1.8.0->-r requirements.txt (li
Requirement already satisfied: typing-extensions>=4.10.0 in /usr/local/lib/python3.12/dist-packages (from torch>=1.8.0->-r requi
Requirement already satisfied: sympy>=1.13.3 in /usr/local/lib/python3.12/dist-packages (from torch>=1.8.0->-r requirements.txt (li
Requirement already satisfied: networkx in /usr/local/lib/python3.12/dist-packages (from torch>=1.8.0->-r requirements.txt (li
Requirement already satisfied: jinja2 in /usr/local/lib/python3.12/dist-packages (from torch>=1.8.0->-r requirements.txt (line
Requirement already satisfied: fsspec in /usr/local/lib/python3.12/dist-packages (from torch>=1.8.0->-r requirements.txt (line
Requirement already satisfied: nvidia-cuda-nvrtc-cu12==12.6.77 in /usr/local/lib/python3.12/dist-packages (from torch>=1.8.0->
Requirement already satisfied: nvidia-cuda-runtime-cu12==12.6.77 in /usr/local/lib/python3.12/dist-packages (from torch>=1.8.0
Requirement already satisfied: nvidia-cuda-cupti-cu12==12.6.80 in /usr/local/lib/python3.12/dist-packages (from torch>=1.8.0->
Requirement already satisfied: nvidia-cudnn-cu12==9.10.2.21 in /usr/local/lib/python3.12/dist-packages (from torch>=1.8.0->-r Requirement already satisfied: nvidia-cublas-cu12==12.6.4.1 in /usr/local/lib/python3.12/dist-packages (from torch>=1.8.0->-r
```

```
Requirement already satisfied: nvidia-cusparse-cu12==12.5.4.2 in /usr/local/lib/python3.12/dist-packages (from torch>=1.8.0->-
Requirement already satisfied: nvidia-cusparselt-cu12==0.7.1 in /usr/local/lib/python3.12/dist-packages (from torch>=1.8.0->-
Requirement already satisfied: nvidia-nccl-cu12==2.27.3 in /usr/local/lib/python3.12/dist-packages (from torch>=1.8.0->-
Requirement already satisfied: nvidia-nvtx-cu12==12.6.77 in /usr/local/lib/python3.12/dist-packages (from torch>=1.8.0->-
Requirement already satisfied: nvidia-nvjitlink-cu12==12.6.85 in /usr/local/lib/python3.12/dist-packages (from torch>=1.8.0->-
Requirement already satisfied: nvidia-cufile-cu12==1.11.1.6 in /usr/local/lib/python3.12/dist-packages (from torch>=1.8.0->-
Requirement already satisfied: triton==3.4.0 in /usr/local/lib/python3.12/dist-packages (from torch>=1.8.0-> requirements.txt
Requirement already satisfied: polars in /usr/local/lib/python3.12/dist-packages (from ultralytics>=8.2.64->-r requirements.txt
Requirement already satisfied: ultralytics-thop>=2.0.18 in /usr/local/lib/python3.12/dist-packages (from ultralytics>=8.2.64->-
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.12/dist-packages (from pandas>=1.1.4->-r requirements.txt
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.12/dist-packages (from pandas>=1.1.4->-r requirements.txt
Requirement already satisfied: smmap<6,>=3.0.1 in /usr/local/lib/python3.12/dist-packages (from gitdb<5,>=4.0.1->gitpython>=3.-
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.12/dist-packages (from python-dateutil>=2.7->matplotlib>=3.3
Requirement already satisfied: mpmath<1.4,>=1.1.0 in /usr/local/lib/python3.12/dist-packages (from sympy>=1.13.3->torch>=1.8.0
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.12/dist-packages (from jinja2->torch>=1.8.0->-r requi
```

```
import ultralytics
from ultralytics import YOLO
```

```
device = "cuda" if torch.cuda.is_available() else "cpu"
```

```
!ls /content/surgical_tool_suturing-2
```

```
data.yaml README.dataset.txt README.roboflow.txt test train valid
```

```
!python train.py \
--weights /content/instrument_detector_model.pt \
--data /content/surgical_tool_suturing-2/data.yaml \
--epochs 50 \
--batch-size 16 \
--img 640 \
--project runs/train \
--name finetuned_instruments \
--exist-ok
```

```

best/precision 0.97653
    best/recall 0.98851
metrics/mAP_0.5 0.98487
metrics/mAP_0.5:0.95 0.60908
    metrics/precision 0.97653
    metrics/recall 0.98851
    train/box_loss 0.0254
        +8 ...

```

View run `finetuned_instruments` at: <https://wandb.ai/rumaisa1054-uet-taxila/YOLOv5/runs/twwdd9o4>
View project at: <https://wandb.ai/rumaisa1054-uet-taxila/YOLOv5>
synced 5 W&B file(s), 15 media file(s), 2 artifact file(s) and 0 other file(s)
bind logs at: ./wandb/run-20251106_090935-twwdd9o4/logs
WARNING ⚠ wandb is deprecated and will be removed in a future release. See supported integrations at <https://github.com/ultral>

```
!ls runs/train/*/weights/
```

```
best.pt last.pt
```

```
!python val.py \
--weights runs/train/finetuned_instruments/weights/best.pt \
--data /content/surgical_tool_suturing-2/data.yaml
```

```

val: data=/content/surgical_tool_suturing-2/data.yaml, weights=['runs/train/finetuned_instruments/weights/best.pt'], batch_size=YOLOv5 🎨 v7.0-446-ga53364d6 Python-3.12.12 torch-2.8.0+cu126 CUDA:0 (Tesla T4, 15095MiB)

Fusing layers...
Model summary: 267 layers, 46108278 parameters, 0 gradients, 107.6 GFLOPs
val: Scanning /content/surgical_tool_suturing-2/valid/labels.cache... 45 images, 0 backgrounds, 0 corrupt: 100% 45/45 [00:00<?, 1.33s/it]
      Class     Images   Instances       P       R     mAP50   mAP50-95: 100% 2/2 [00:02<00:00,  1.33s/it]
          all         45        87     0.977     0.989     0.985     0.609
Speed: 1.0ms pre-process, 29.0ms inference, 3.2ms NMS per image at shape (32, 3, 640, 640)
Results saved to runs/val/exp4

```

```
!zip -r /content/finetuned_instruments.zip /content/yolov5/runs/train/finetuned_instruments
```

```

adding: content/yolov5/runs/train/finetuned_instruments/ (stored 0%)
adding: content/yolov5/runs/train/finetuned_instruments/train_batch0.jpg (deflated 4%)
adding: content/yolov5/runs/train/finetuned_instruments/train_batch1.jpg (deflated 4%)
adding: content/yolov5/runs/train/finetuned_instruments/val_batch0_pred.jpg (deflated 11%)
adding: content/yolov5/runs/train/finetuned_instruments/train_batch2.jpg (deflated 3%)
adding: content/yolov5/runs/train/finetuned_instruments/confusion_matrix.png (deflated 40%)
adding: content/yolov5/runs/train/finetuned_instruments/events.out.tfevents.1762419782.5558386373f1.3023.0 (deflated 9%)
adding: content/yolov5/runs/train/finetuned_instruments/P_curve.png (deflated 23%)
adding: content/yolov5/runs/train/finetuned_instruments/events.out.tfevents.1762419994.5558386373f1.4472.0 (deflated 9%)
adding: content/yolov5/runs/train/finetuned_instruments/results.png (deflated 9%)
adding: content/yolov5/runs/train/finetuned_instruments/opt.yaml (deflated 50%)
adding: content/yolov5/runs/train/finetuned_instruments/labels_correlogram.jpg (deflated 46%)
adding: content/yolov5/runs/train/finetuned_instruments/hyp.yaml (deflated 45%)
adding: content/yolov5/runs/train/finetuned_instruments/val_batch1_pred.jpg (deflated 13%)
adding: content/yolov5/runs/train/finetuned_instruments/weights/ (stored 0%)
adding: content/yolov5/runs/train/finetuned_instruments/weights/last.pt (deflated 8%)
adding: content/yolov5/runs/train/finetuned_instruments/weights/best.pt (deflated 8%)
adding: content/yolov5/runs/train/finetuned_instruments/val_batch0_labels.jpg (deflated 12%)
adding: content/yolov5/runs/train/finetuned_instruments/labels.jpg (deflated 31%)
adding: content/yolov5/runs/train/finetuned_instruments/val_batch1_labels.jpg (deflated 13%)
adding: content/yolov5/runs/train/finetuned_instruments/F1_curve.png (deflated 17%)
adding: content/yolov5/runs/train/finetuned_instruments/results.csv (deflated 84%)
adding: content/yolov5/runs/train/finetuned_instruments/PR_curve.png (deflated 28%)
adding: content/yolov5/runs/train/finetuned_instruments/events.out.tfevents.1762419952.5558386373f1.4251.0 (deflated 9%)
adding: content/yolov5/runs/train/finetuned_instruments/events.out.tfevents.1762420175.5558386373f1.5286.0 (deflated 52%)
adding: content/yolov5/runs/train/finetuned_instruments/R_curve.png (deflated 18%)

```

```
%cd /content/yolov5
```

```
/content/yolov5
```

```

import torch
from yolov5.models.common import DetectMultiBackend
from yolov5.utils.torch_utils import select_device

device = select_device('0') # use GPU if available
model_path = "/content/yolov5/runs/train/finetuned_instruments/weights/best.pt"

model = DetectMultiBackend(model_path, device=device)

```

```

model.eval() # set to evaluation mode
names = model.names # class names
print(names)

YOLOv5 🚀 v7.0-446-ga53364d6 Python-3.12.12 torch-2.8.0+cu126 CUDA:0 (Tesla T4, 15095MiB)

Fusing layers...
Model summary: 267 layers, 46108278 parameters, 0 gradients, 107.6 GFLOPs
{0: 'tool'}

```

```

def read_video(video_path):
    cap = cv2.VideoCapture(video_path)
    if not cap.isOpened():
        raise ValueError(f"Cannot open video: {video_path}")
    fps = cap.get(cv2.CAP_PROP_FPS) or 30.0
    return cap, fps

```

```

def get_frame_rgb(cap):
    ret, frame = cap.read()
    if not ret:
        return None
    return cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)

```

```

%matplotlib inline
from yolov5.utils.general import non_max_suppression
def detect_objects(frame_rgb, model, conf_threshold=0.25):
    """Run inference and return Nx6 array [x1,y1,x2,y2,conf,cls]"""
    img = np.asarray(frame_rgb)
    img_tensor = torch.from_numpy(img).permute(2,0,1).float().unsqueeze(0) / 255.0
    img_tensor = img_tensor.to(model.device)
    if getattr(model, 'fp16', False):
        img_tensor = img_tensor.half()

    with torch.no_grad():
        pred = model(img_tensor)
        pred = non_max_suppression(pred, conf_threshold, 0.45)

    if len(pred) == 0 or pred[0] is None or pred[0].numel() == 0:
        return np.zeros((0,6))

    return pred[0].cpu().numpy()

def draw_boxes(frame_rgb, det):
    """Draw boxes on RGB frame using Matplotlib-friendly colors"""
    img = frame_rgb.copy()
    for row in det:
        x1, y1, x2, y2, conf, cls = row
        x1, y1, x2, y2 = map(int, [x1, y1, x2, y2])
        cv2.rectangle(img, (x1, y1), (x2, y2), (0,255,0), 2)
        cls_name = model.names[int(cls)] if int(cls) in model.names else str(int(cls))
        cv2.putText(img, f"{cls_name} {conf:.2f}", (x1, y1-6),
                   cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255,0,0), 1, cv2.LINE_AA)
    return img

# -----
# ③ Run on a single image
# -----
img_path = "/content/frame_0085.jpg"
frame = cv2.imread(img_path)
frame_rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)

det = detect_objects(frame_rgb, model, conf_threshold=0.1)
print("Detections:", det)

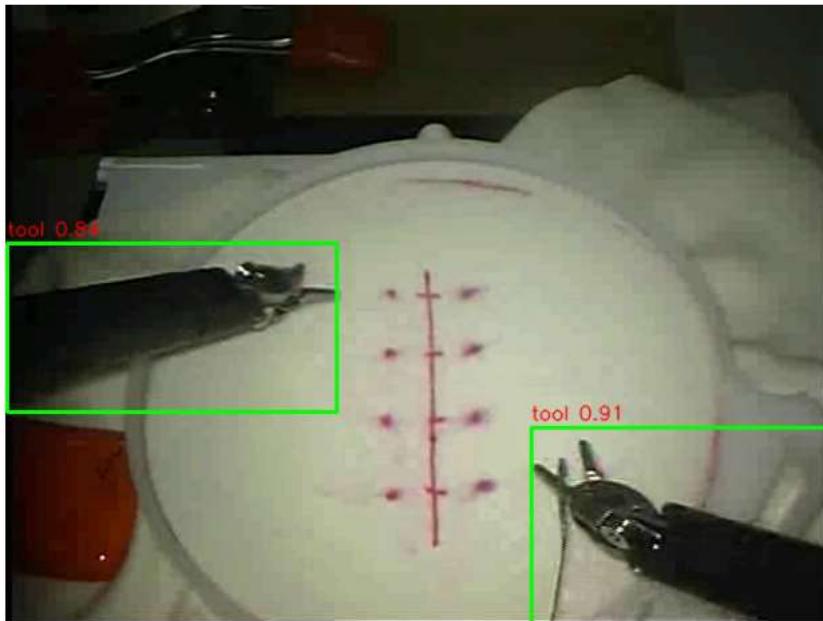
img_boxes = draw_boxes(frame_rgb, det)

# Display using matplotlib
plt.figure(figsize=(10,6))
plt.imshow(img_boxes) # RGB image
plt.axis('off')
plt.title("YOLOv5 Detections")
plt.show()

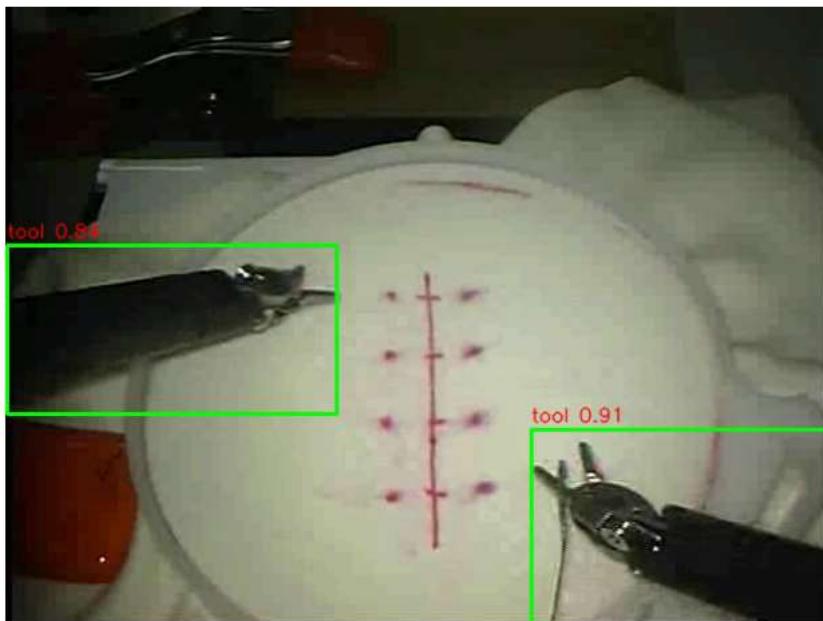
```

```
Detections: [[        408.6      328.72      641.14      482.01      0.9072      0]
 [       1.1067     185.28     257.48     316.56      0.8428      0]]
```

YOLOv5 Detections



YOLOv5 Detections



```
def compute_center(boxes):
    """
    boxes: Nx4 (x1,y1,x2,y2) or Nx6 (x1,y1,x2,y2,conf,cls)
    return mean center [cx, cy] or None if no boxes
    """
    if boxes is None or len(boxes) == 0:
        return None
    boxes_xy = boxes[:, :4]
    centers = np.stack([(b[0]+b[2])/2.0, (b[1]+b[3])/2.0) for b in boxes_xy ])
    return np.mean(centers, axis=0)
```

```
def compute_motion_metrics(motion_magnitude, frame_count, fps):
    if len(motion_magnitude) == 0:
        return 0.0, frame_count / (fps or 1.0), 0.0
    how_far = float(np.mean(motion_magnitude))
    how_long = float(frame_count / (fps or 1.0))
    how_steady = float(1.0 / (np.std(motion_magnitude) + 1e-8))
```

```

    return how_far, how_long, how_steady

sample_images = []
def process_video(video_path, model, conf_threshold=0.25, frame_skip=5, show_examples=3):
    cap, fps = read_video(video_path)
    prev_center = None
    motion_magnitude = []
    frame_count = 0
    sample_images = []
    shown = 0
    show_examples = 5

    while True:
        frame_rgb = get_frame_rgb(cap)
        if frame_rgb is None:
            break
        frame_count += 1
        if frame_count % frame_skip != 0:
            continue

        det = detect_objects(frame_rgb, model, conf_threshold=conf_threshold)
        # det: Nx6 numpy array [x1,y1,x2,y2,conf,cls]
        if det.shape[0] > 0:
            curr_center = compute_center(det)
        else:
            curr_center = None

        # store example render images for first few detections
        if shown < show_examples:
            disp = frame_rgb.copy()
            for row in det:
                x1, y1, x2, y2, conf, cls = row
                cv2.rectangle(disp, (int(x1), int(y1)), (int(x2), int(y2)), (0,255,0), 2)
                cls_name = model.names[int(cls)] if int(cls) in model.names else str(int(cls))
                cv2.putText(disp, f"{cls_name} {conf:.2f}", (int(x1), int(y1)-6),
                           cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255,0,0), 1, cv2.LINE_AA)

            sample_images.append(cv2.cvtColor(disp, cv2.COLOR_RGB2BGR)) # convert for display
            shown += 1

        if curr_center is not None and prev_center is not None:
            dist = float(np.linalg.norm(curr_center - prev_center))
            motion_magnitude.append(dist)

        if curr_center is not None:
            prev_center = curr_center
    plt.figure(figsize=(15, 10))
    for i, img in enumerate(sample_images):
        plt.subplot(1, len(sample_images), i + 1)
        plt.imshow(cv2.cvtColor(img, cv2.COLOR_BGR2RGB))
        plt.axis("off")
    plt.show()

    cap.release()
    how_far, how_long, how_steady = compute_motion_metrics(motion_magnitude, frame_count, fps)

    return how_far, how_long, how_steady, sample_images

import matplotlib.pyplot as plt
import cv2

# Call your function (it just returns frames, no plotting inside)
how_far, how_long, how_steady, sample_images = process_video(
    "/content/suturing_dataset/data/videos/Suturing_B001.avi", model
)

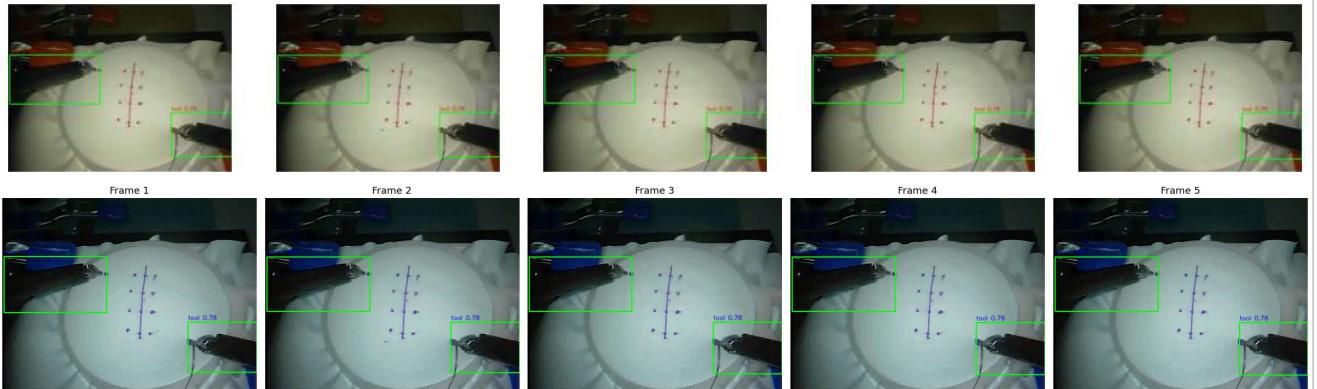
# Ensure we have at least one image
if len(sample_images) == 0:
    print("No sample images to display.")
else:
    n = len(sample_images)
    fig, axes = plt.subplots(1, n, figsize=(5*n, 5))
    if n == 1:
        axes = [axes] # Make iterable if only one image

```

```

for ax, img, i in zip(axes, sample_images, range(n)):
    # img should already be RGB
    ax.imshow(img)
    ax.set_title(f"Frame {i+1}")
    ax.axis("off")
plt.tight_layout()
plt.show()

```



```

def find_video_file(video_dir, base_name):
    for ext in [".mp4", ".avi", ".mov", ".MTS", ".mkv"]:
        candidate = os.path.join(video_dir, base_name + ext)
        if os.path.exists(candidate):
            return candidate
    # try full filename match
    candidate = os.path.join(video_dir, base_name)
    if os.path.exists(candidate):
        return candidate
    return None

```

```

def analyze_dataset(video_dir, count, label_file):
    # read label file: accept CSV or txt with two columns: filename,score
    if label_file.lower().endswith(".txt") or label_file.lower().endswith(".csv"):
        df_labels = pd.read_csv(label_file, sep=None, engine='python', header=None, names=["filename", "score"])
    else:
        df_labels = pd.read_csv(label_file)

    df_labels = df_labels.head(count)
    metrics = []
    print(f"\nProcessing {len(df_labels)} videos...\n")

    for _, row in tqdm(df_labels.iterrows(), total=len(df_labels)):
        base_name = str(row["filename"]).strip()
        vid_path = find_video_file(video_dir, base_name)
        if not vid_path:
            print(f"Missing video for: {base_name}")
            continue
        how_far, how_long, how_steady, samples = process_video(vid_path, model,
                                                               conf_threshold=0.95,
                                                               frame_skip=5,
                                                               show_examples=3)

        metrics.append({
            "filename": os.path.basename(vid_path),
            "how_far": how_far,
            "how_long": how_long,
            "how_steady": how_steady,
            "score": float(row["score"])
        })

    df_metrics = pd.DataFrame(metrics)
    df_metrics.to_csv("metrics_and_labels.csv", index=False)
    print("\nSaved metrics_and_labels.csv")
    return df_metrics

```

```

import warnings
warnings.filterwarnings("ignore") # Ignore all warnings

```

```
df_metrics = analyze_dataset("/content/suturing_dataset/data/videos", 10, "/content/suturing_dataset/data/labels.txt")

# Plotting and Spearman correlation (safe handling of constant columns)
if df_metrics.shape[0] > 0:
    print("\nSpearman correlations:")
    plt.figure(figsize=(12,4))
    for i, metric in enumerate(["how_far", "how_long", "how_steady"], 1):
        vals = df_metrics[metric].values
        scores = df_metrics["score"].values
        # handle constant arrays
        try:
            rho, pval = spearmanr(vals, scores)
        except Exception as e:
            print("Exception")
            rho, pval = np.nan, np.nan
        print(f"{metric}: ρ = {rho}, p = {pval}")
        plt.subplot(1,3,i)
        plt.scatter(vals, scores, alpha=0.6)
        plt.xlabel(metric); plt.ylabel("score")
        plt.title(f"{metric} vs score")
    plt.tight_layout()
    plt.show()
else:
    print("No metrics to plot (no videos processed).")
```


Processing 10 videos...

0% | 0/10 [00:00<?, ?it/s]



10% | 1/10 [00:20<03:03, 20.41s/it]



20% | 2/10 [00:40<02:41, 20.24s/it]



30% | 3/10 [00:59<02:17, 19.63s/it]



40% | 4/10 [01:15<01:50, 18.35s/it]



```
for i, img in enumerate(sample_images):
    plt.figure(figsize=(10, 6))
    plt.imshow(cv2.cvtColor(img, cv2.COLOR_BGR2RGB))
    plt.axis('off')
    plt.title(f"Detected Objects - Example {i+1}")
    plt.show()
```



60% | 6/10 [01:54<01:16, 19.06s/it]



70% | 7/10 [02:13<00:56, 18.92s/it]



80% | 8/10 [02:29<00:35, 17.91s/it]

