# Image-Segmentation-YOLOv7-Pytorch-GPL-Jupyter

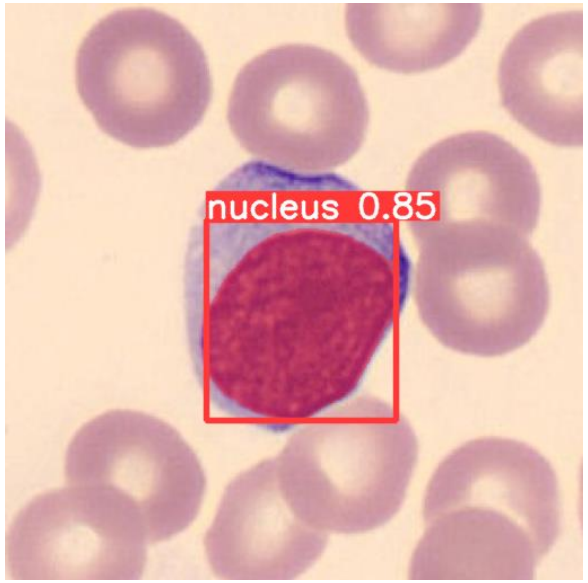
YOLOv7 is the most powerful object detection algorithm now. We also can use it to do instance segmentation instead of Mask R-CNN.

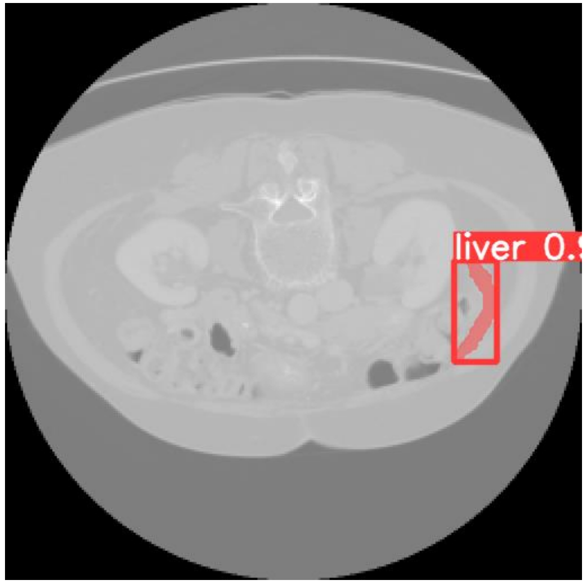
We organized the code so that we can use JupyterLab to perform the training and inference steps easily.

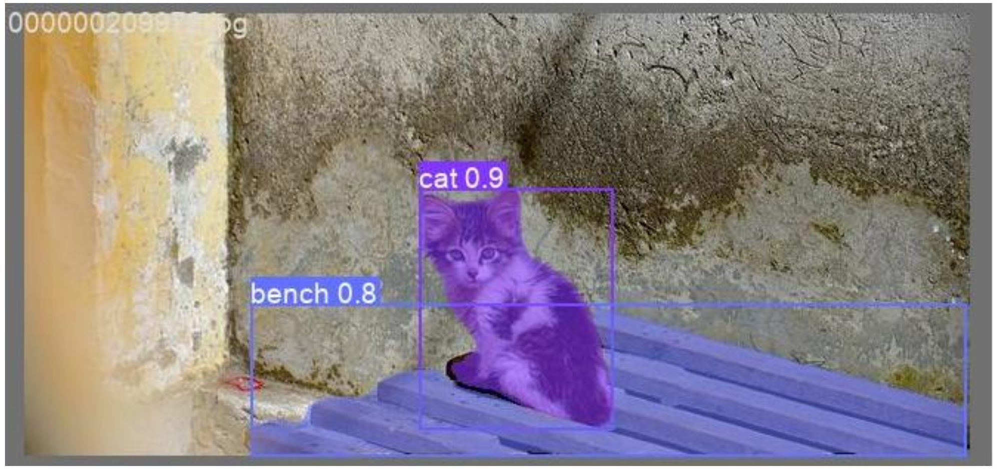
Version 20230223

# Applications

- The YOLOv7 segmentation can be applied to factory defect detection, medical image analysis, biological image analysis, industrial safety image analysis, etc.



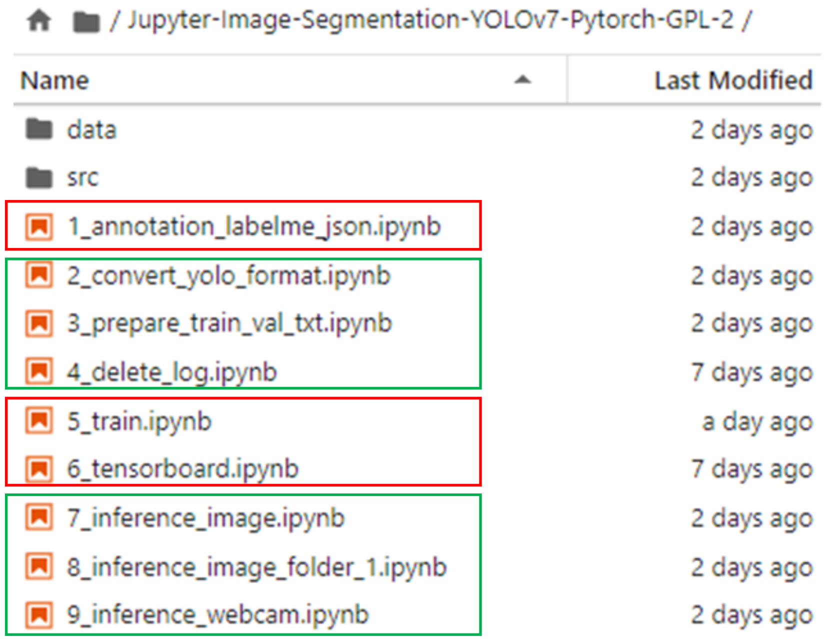




# How to use

The main process is:

Annotate images -> Prepare files for training -> Training -> Inference

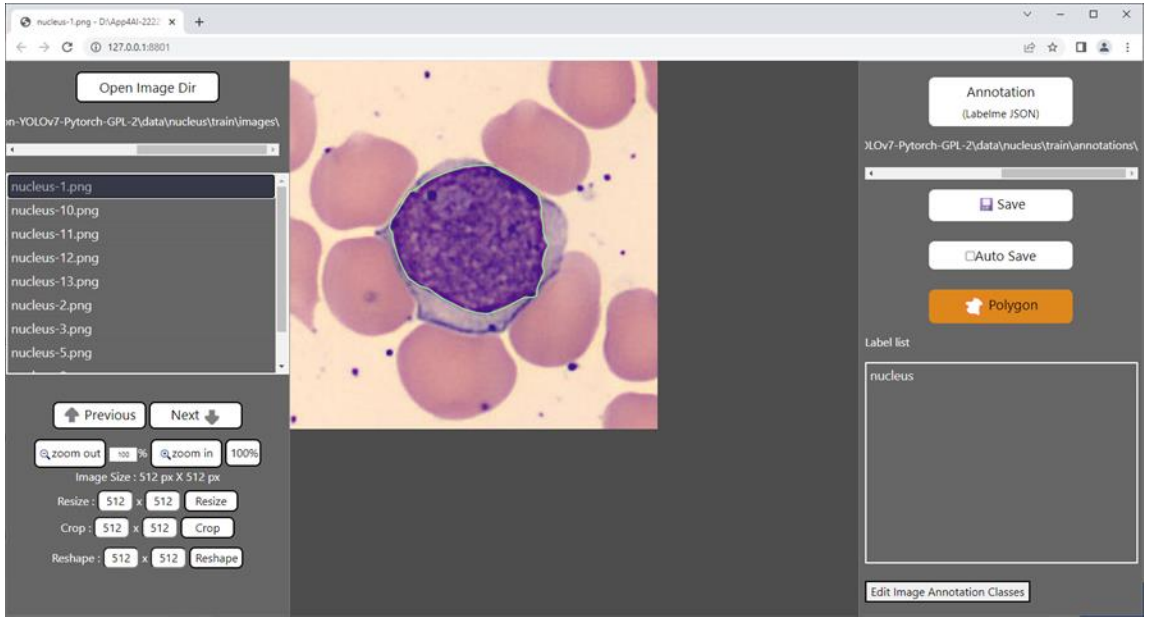


/ Jupyter-Image-Segmentation-YOLOv7-Pytorch-GPL-2 /

| Name | Last Modified |
|---|---|
| data | 2 days ago |
| src | 2 days ago |
| 1_annotation_labelme_json.ipynb | 2 days ago |
| 2_convert_yolo_format.ipynb | 2 days ago |
| 3_prepare_train_val_txt.ipynb | 2 days ago |
| 4_delete_log.ipynb | 7 days ago |
| 5_train.ipynb | a day ago |
| 6_tensorboard.ipynb | 7 days ago |
| 7_inference_image.ipynb | 2 days ago |
| 8_inference_image_folder_1.ipynb | 2 days ago |
| 9_inference_webcam.ipynb | 2 days ago |

# 1_annotation_labelme_json.ipynb

Open the web page for image annotation.



nucleus-1.png - D:\App4AI-222

127.0.0.1:8801

n-YOLOv7-Pytorch-GPL-2\data\nucleus\train\images\

nucleus-1.png
nucleus-10.png
nucleus-11.png
nucleus-12.png
nucleus-13.png
nucleus-2.png
nucleus-3.png
nucleus-5.png

100 %

Image Size : 512 px X 512 px

Resize : 512 x 512

Crop : 512 x 512

Reshape : 512 x 512

)LOv7-Pytorch-GPL-2\data\nucleus\train\annotations\

□Auto Save

Label list

nucleus

ipynb parameter:

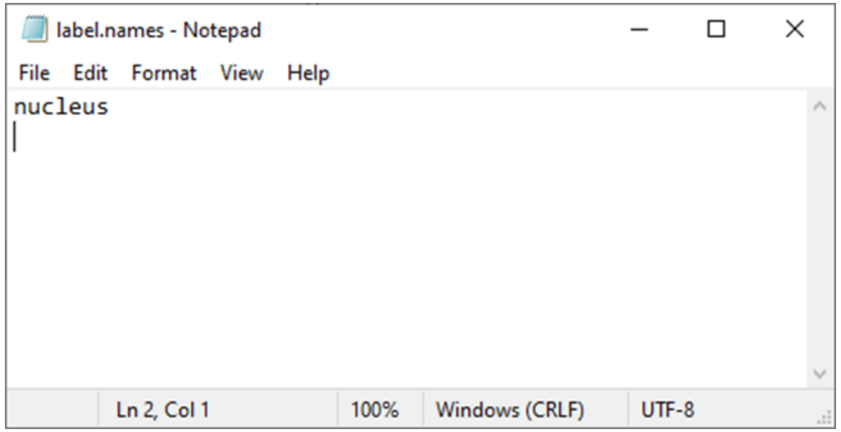
- “port” is the port used by the web page. If the port is occupied by other program, please change another port value by yourself.
- “dataset” is the dataset name

# 2_convert_yolo_format.ipynb

Convert the labelme json label file to the yolo format. Before running, please confirm label.names under the label_file path in #parameters and whether the content filled in the category is correct.

Remark:

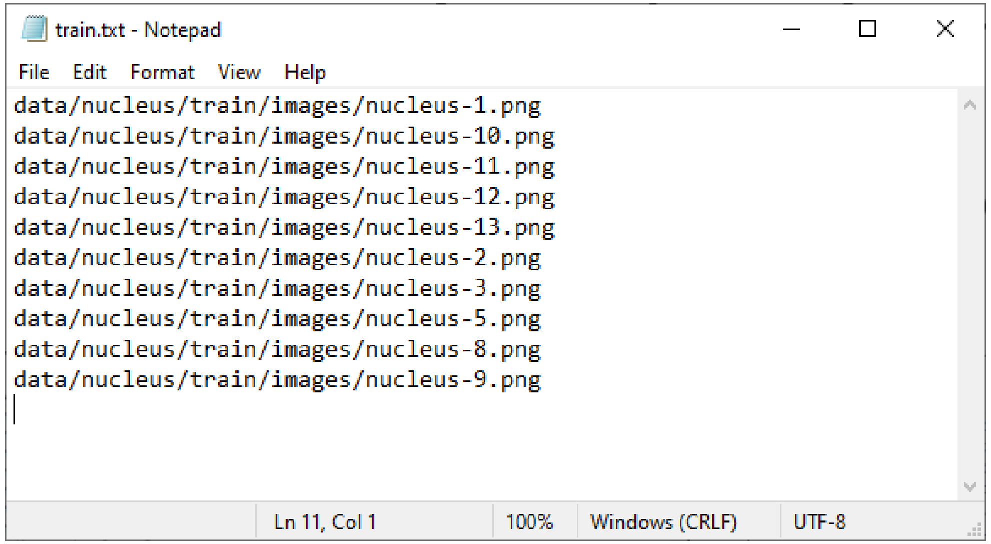
The content of label.names is the category name without background.



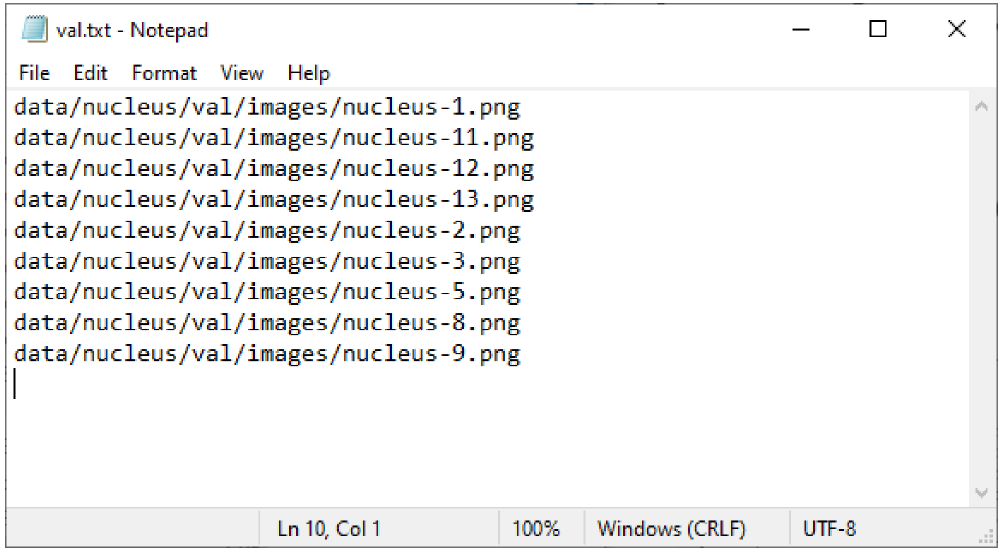
```
nucleus
```

# 3_prepare_train_val_txt.ipynb

Generate training and validation image path files train.txt and val.txt.



```
data/nucleus/train/images/nucleus-1.png
data/nucleus/train/images/nucleus-10.png
data/nucleus/train/images/nucleus-11.png
data/nucleus/train/images/nucleus-12.png
data/nucleus/train/images/nucleus-13.png
data/nucleus/train/images/nucleus-2.png
data/nucleus/train/images/nucleus-3.png
data/nucleus/train/images/nucleus-5.png
data/nucleus/train/images/nucleus-8.png
data/nucleus/train/images/nucleus-9.png
```



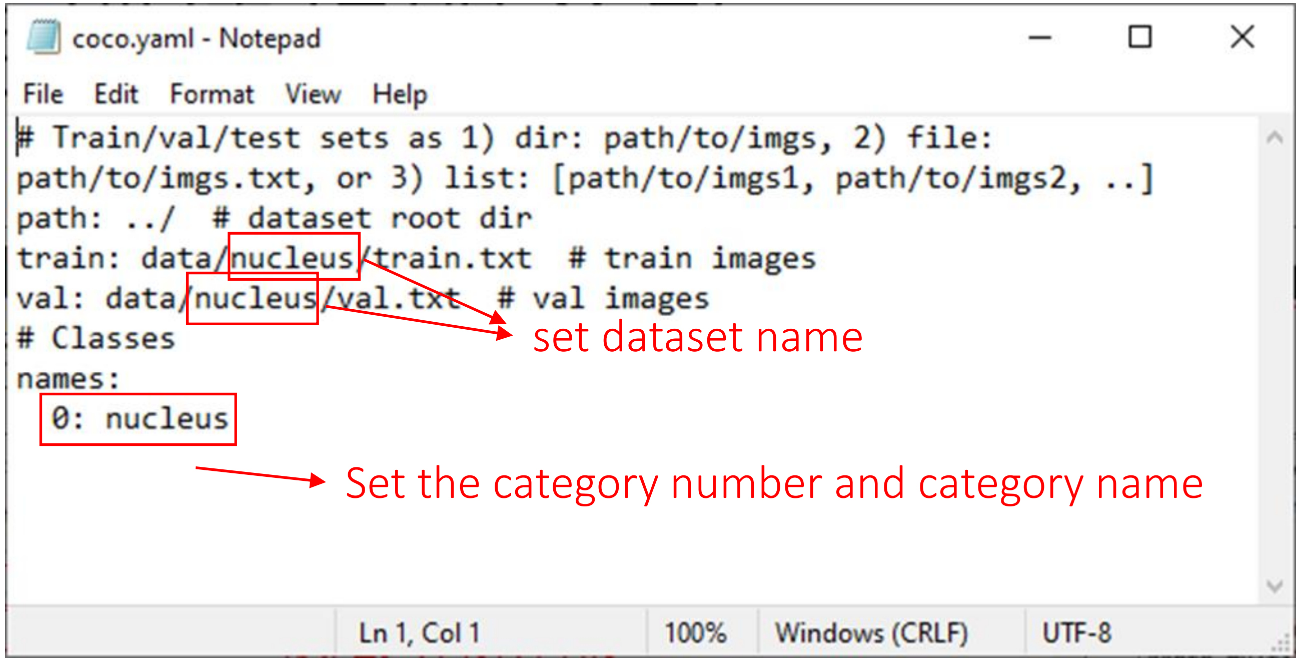
```
data/nucleus/val/images/nucleus-1.png
data/nucleus/val/images/nucleus-11.png
data/nucleus/val/images/nucleus-12.png
data/nucleus/val/images/nucleus-13.png
data/nucleus/val/images/nucleus-2.png
data/nucleus/val/images/nucleus-3.png
data/nucleus/val/images/nucleus-5.png
data/nucleus/val/images/nucleus-8.png
data/nucleus/val/images/nucleus-9.png
```

# 4_delete_log.ipynb

Delete the log files from previous training.

# Set training related files

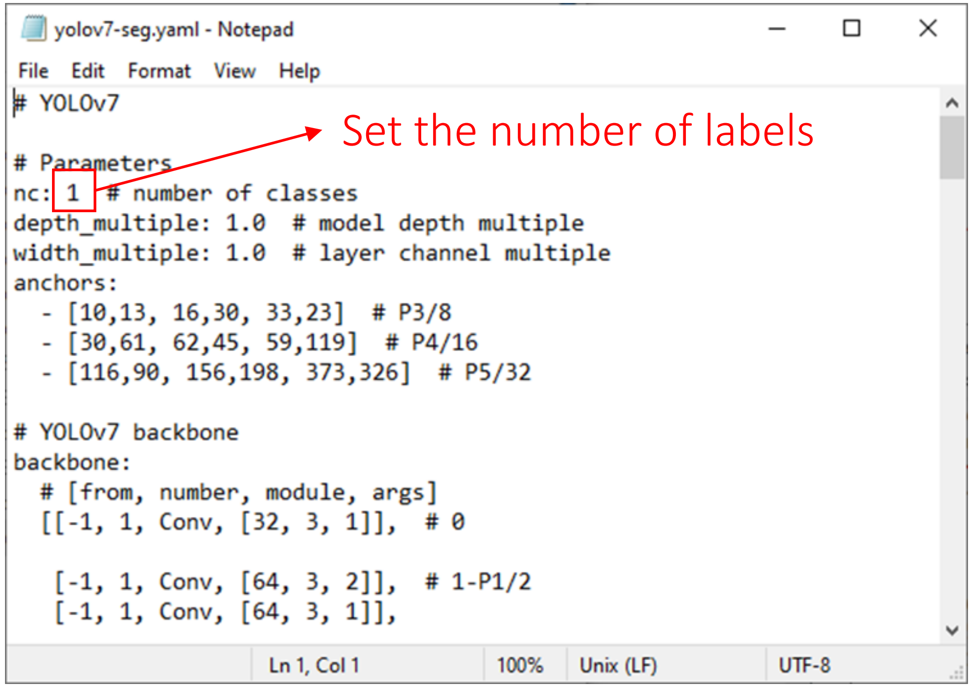
Set the content of the yolov7-seg.yaml and coco.yaml files in the dataset, set the name of the data set, the number of categories and the name.



```
# Train/val/test sets as 1) dir: path/to/imgs, 2) file:
path/to/imgs.txt, or 3) list: [path/to/imgs1, path/to/imgs2, ..]
path: ../  # dataset root dir
train: data/nucleus/train.txt  # train images
val: data/nucleus/val.txt  # val images
# Classes
names:
  0: nucleus
```

set dataset name

Set the category number and category name

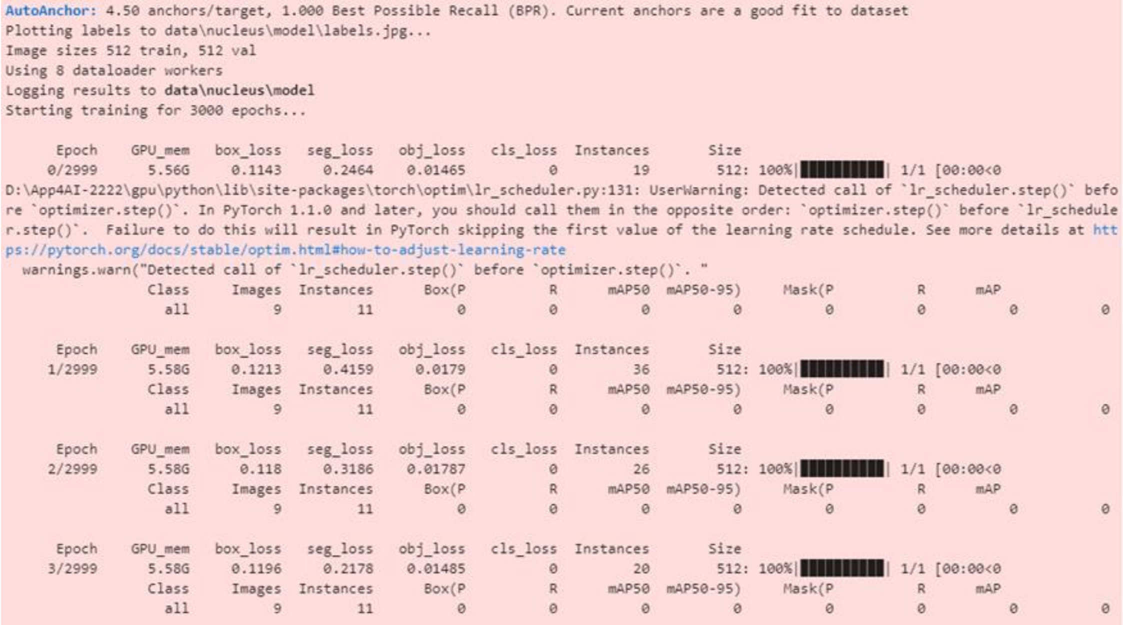


```
# YOLOv7

# Parameters
nc: 1  # number of classes
depth_multiple: 1.0  # model depth multiple
width_multiple: 1.0  # layer channel multiple
anchors:
  - [10,13, 16,30, 33,23]  # P3/8
  - [30,61, 62,45, 59,119]  # P4/16
  - [116,90, 156,198, 373,326]  # P5/32

# YOLOv7 backbone
backbone:
  # [from, number, module, args]
  [[-1, 1, Conv, [32, 3, 1]],  # 0

   [-1, 1, Conv, [64, 3, 2]],  # 1-P1/2
   [-1, 1, Conv, [64, 3, 1]],
```

Set the number of labels

# 5_train.ipynb

Start training.



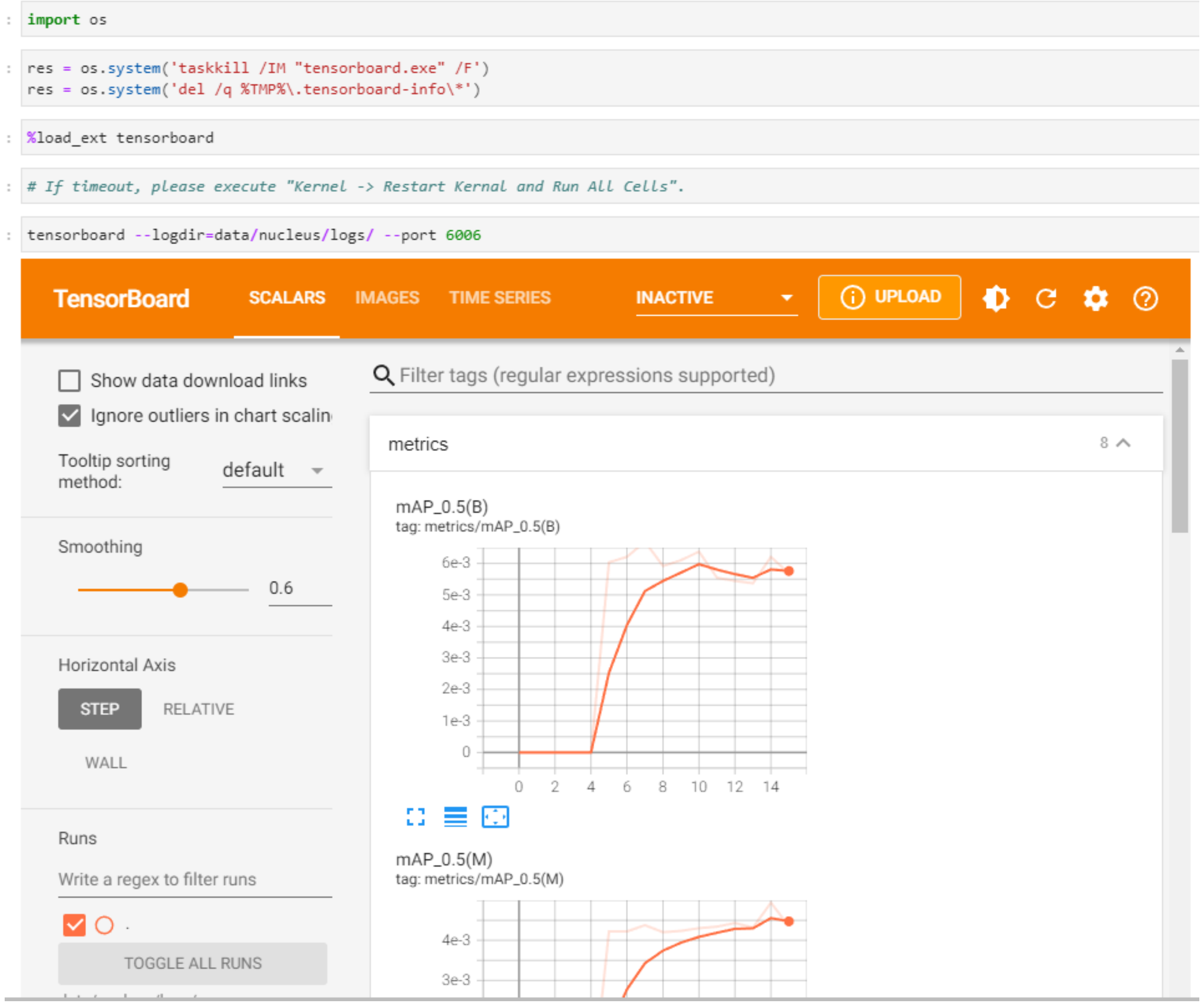
```
AutoAnchor: 4.50 anchors/target, 1.000 Best Possible Recall (BPR). Current anchors are a good fit to dataset
Plotting labels to data\nucleus\model\labels.jpg...
Image sizes 512 train, 512 val
Using 8 dataloader workers
Logging results to data\nucleus\model
Starting training for 3000 epochs...

     Epoch   GPU_mem  box_loss  seg_loss  obj_loss  cls_loss Instances      Size
    0/2999     5.56G    0.1143    0.2464   0.01465         0        19       512: 100%|██████████| 1/1 [00:00<0
D:\App4AI-2222\gpu\python\lib\site-packages\torch\optim\lr_scheduler.py:131: UserWarning: Detected call of `lr_scheduler.step()` befo
re `optimizer.step()`. In PyTorch 1.1.0 and later, you should call them in the opposite order: `optimizer.step()` before `lr_schedule
r.step()`.  Failure to do this will result in PyTorch skipping the first value of the learning rate schedule. See more details at htt
ps://pytorch.org/docs/stable/optim.html#how-to-adjust-learning-rate
  warnings.warn("Detected call of `lr_scheduler.step()` before `optimizer.step()`. "
               Class    Images Instances     Box(P         R     mAP50  mAP50-95)    Mask(P         R       mAP
                 all         9        11         0         0         0         0         0         0         0         0

     Epoch   GPU_mem  box_loss  seg_loss  obj_loss  cls_loss Instances      Size
    1/2999     5.58G    0.1213    0.4159    0.0179         0        36       512: 100%|██████████| 1/1 [00:00<0
               Class    Images Instances     Box(P         R     mAP50  mAP50-95)    Mask(P         R       mAP
                 all         9        11         0         0         0         0         0         0         0         0

     Epoch   GPU_mem  box_loss  seg_loss  obj_loss  cls_loss Instances      Size
    2/2999     5.58G     0.118    0.3186   0.01787         0        26       512: 100%|██████████| 1/1 [00:00<0
               Class    Images Instances     Box(P         R     mAP50  mAP50-95)    Mask(P         R       mAP
                 all         9        11         0         0         0         0         0         0         0         0

     Epoch   GPU_mem  box_loss  seg_loss  obj_loss  cls_loss Instances      Size
    3/2999     5.58G    0.1196    0.2178   0.01485         0        20       512: 100%|██████████| 1/1 [00:00<0
               Class    Images Instances     Box(P         R     mAP50  mAP50-95)    Mask(P         R       mAP
                 all         9        11         0         0         0         0         0         0         0         0
```

ipynb parameter:

- batch_size : batch size of training
- img_size: training image size
- dataset: dataset name to train
- yaml_file: the location of the coco.yaml file used for training
- cfg_file: yolov7-seg.yaml file location for training
- weights_file: the path of the pretrained model used, None means not to use the pretrained model for training
- device: GPU ID used for training
- hypFile: hyp.scratch-high.yaml file location for training
- save_model_path: save the location of the model file generated by training
- log_path: The location of the tensorbaord log file where the training is stored
- epochs: number of training epochs

# 6_tensorboard.ipynb

You can view the training loss curve and other related information by using TensorBoard.



```
import os
```

```
res = os.system('taskkill /IM "tensorboard.exe" /F')
res = os.system('del /q %TMP%\.tensorboard-info\*')
```

```
%load_ext tensorboard
```

```
# If timeout, please execute "Kernel -> Restart Kernal and Run All Cells".
```

```
tensorboard --logdir=data/nucleus/logs/ --port 6006
```

□ Show data download links

☑ Ignore outliers in chart scalin

Tooltip sorting method: default

Smoothing 0.6

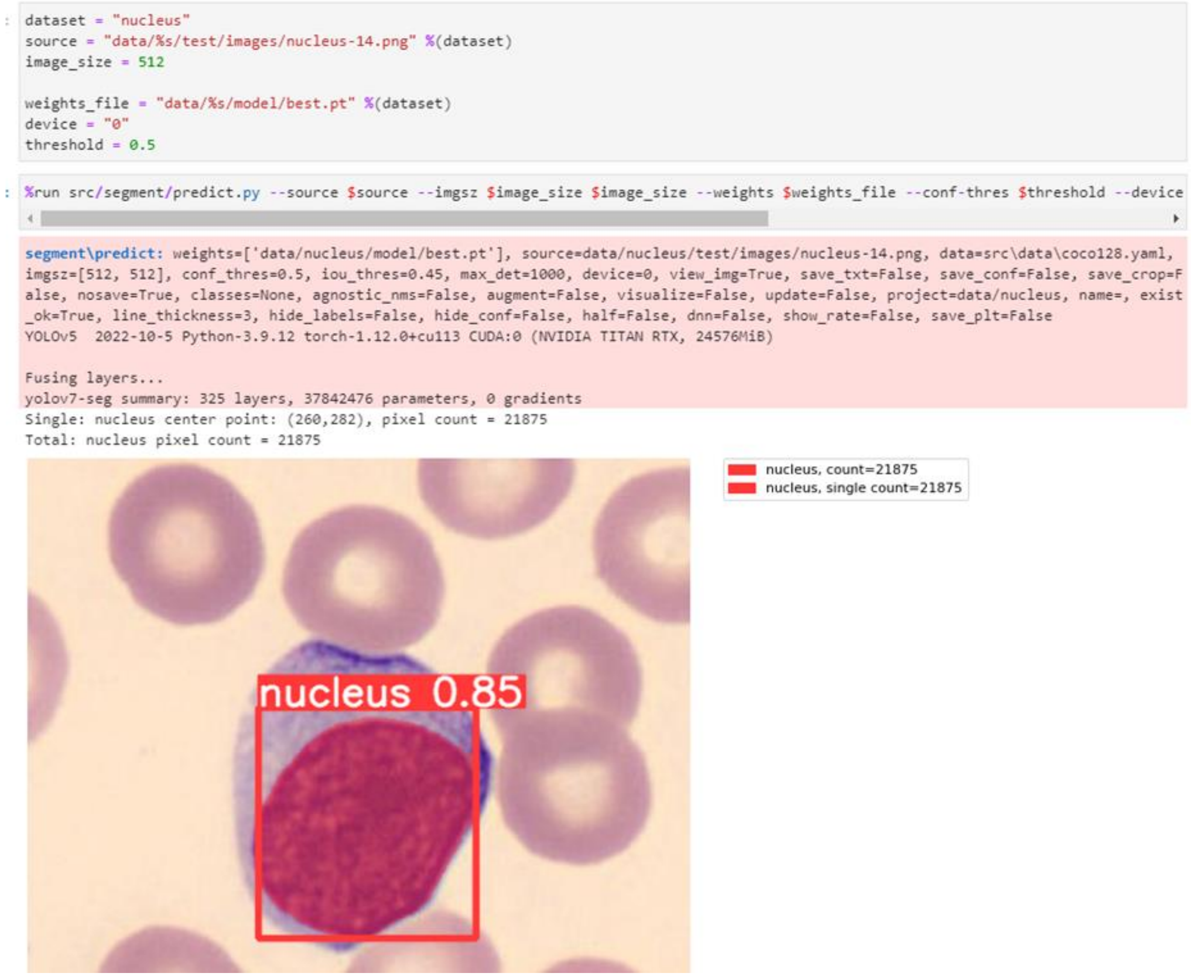
Horizontal Axis: STEP

# 7_inference_image.ipynb

Infer one single image.

ipynb parameter:

- dataset is the dataset name.
- source is the inferred image path.
- weights_file is the inference model path.



```
dataset = "nucleus"
source = "data/%s/test/images/nucleus-14.png" %(dataset)
image_size = 512

weights_file = "data/%s/model/best.pt" %(dataset)
device = "0"
threshold = 0.5
```

```
%run src/segment/predict.py --source $source --imgsz $image_size $image_size --weights $weights_file --conf-thres $threshold --device
```

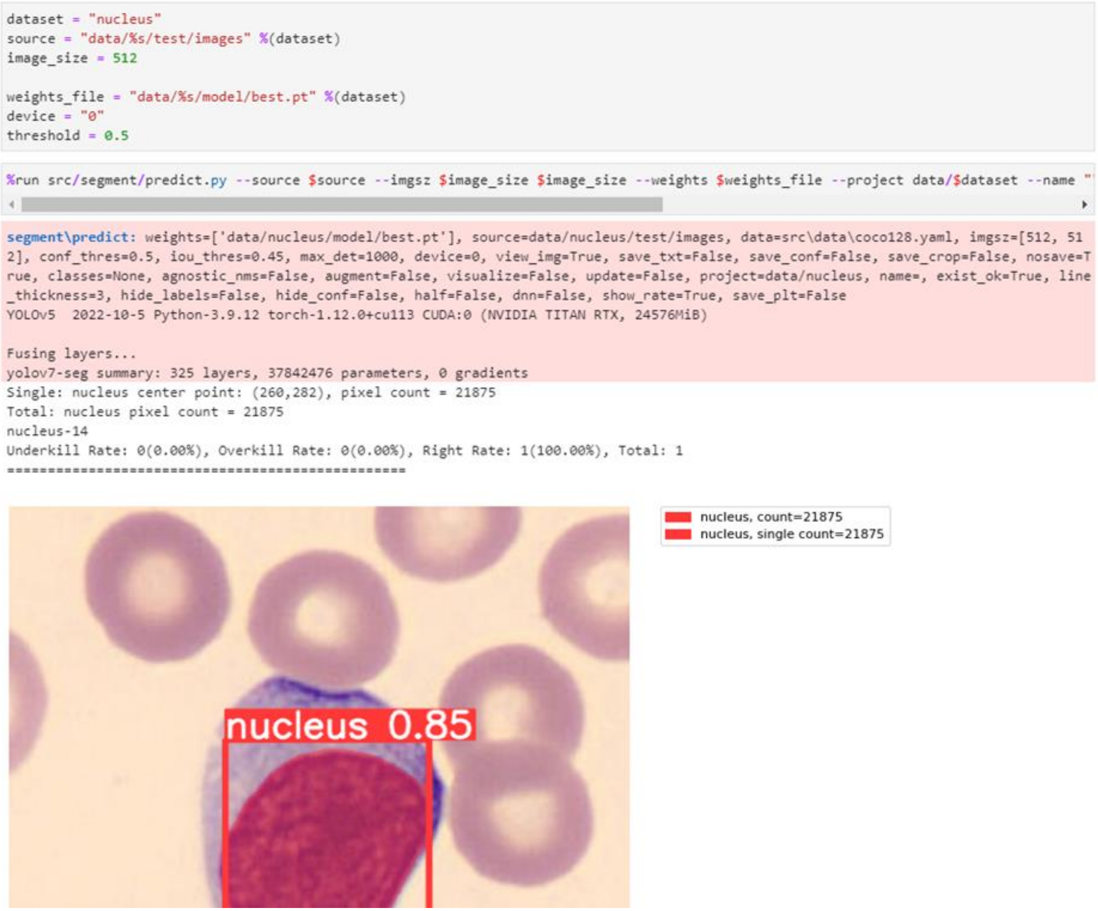
```
segment\predict: weights=['data/nucleus/model/best.pt'], source=data/nucleus/test/images/nucleus-14.png, data=src\data\coco128.yaml,
imgsz=[512, 512], conf_thres=0.5, iou_thres=0.45, max_det=1000, device=0, view_img=True, save_txt=False, save_conf=False, save_crop=F
alse, nosave=True, classes=None, agnostic_nms=False, augment=False, visualize=False, update=False, project=data/nucleus, name=, exist
_ok=True, line_thickness=3, hide_labels=False, hide_conf=False, half=False, dnn=False, show_rate=False, save_plt=False
YOLOv5  2022-10-5 Python-3.9.12 torch-1.12.0+cu113 CUDA:0 (NVIDIA TITAN RTX, 24576MiB)

Fusing layers...
yolov7-seg summary: 325 layers, 37842476 parameters, 0 gradients
Single: nucleus center point: (260,282), pixel count = 21875
Total: nucleus pixel count = 21875
```

# 8_inference_image_folder_1.ipynb

Infer all images in the folder.

ipynb parameter:

- dataset is the dataset name.
- source is the inferred image path.
- weights_file is the inference model path.



```
dataset = "nucleus"
source = "data/%s/test/images" %(dataset)
image_size = 512

weights_file = "data/%s/model/best.pt" %(dataset)
device = "0"
threshold = 0.5
```

```
%run src/segment/predict.py --source $source --imgsz $image_size $image_size --weights $weights_file --project data/$dataset --name "
```

```
segment\predict: weights=['data/nucleus/model/best.pt'], source=data/nucleus/test/images, data=src\data\coco128.yaml, imgsz=[512, 51
2], conf_thres=0.5, iou_thres=0.45, max_det=1000, device=0, view_img=True, save_txt=False, save_conf=False, save_crop=False, nosave=T
rue, classes=None, agnostic_nms=False, augment=False, visualize=False, update=False, project=data/nucleus, name=, exist_ok=True, line
_thickness=3, hide_labels=False, hide_conf=False, half=False, dnn=False, show_rate=True, save_plt=False
YOLOv5  2022-10-5 Python-3.9.12 torch-1.12.0+cu113 CUDA:0 (NVIDIA TITAN RTX, 24576MiB)

Fusing layers...
yolov7-seg summary: 325 layers, 37842476 parameters, 0 gradients
Single: nucleus center point: (260,282), pixel count = 21875
Total: nucleus pixel count = 21875
nucleus-14
Underkill Rate: 0(0.00%), Overkill Rate: 0(0.00%), Right Rate: 1(100.00%), Total: 1
=================================================
```

# 9_inference_webcam.ipynb

Infer the image of the webcam. Press “q” on the display to turn the webcam off.

# Reference

- Please refer to the readme.txt in the SDK folder.
- LEADERG AppForAI: https://www.leaderg.com/appforai-windows