

Image-Segmentation-YOLOv8- Pytorch-GPL-Jupyter

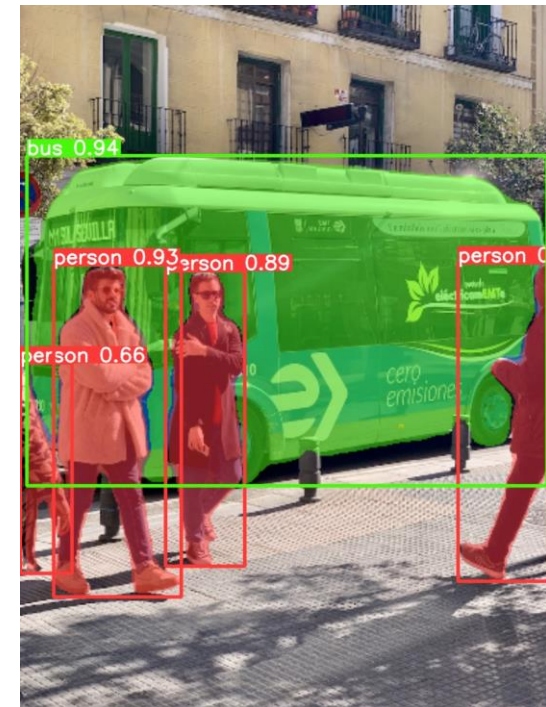
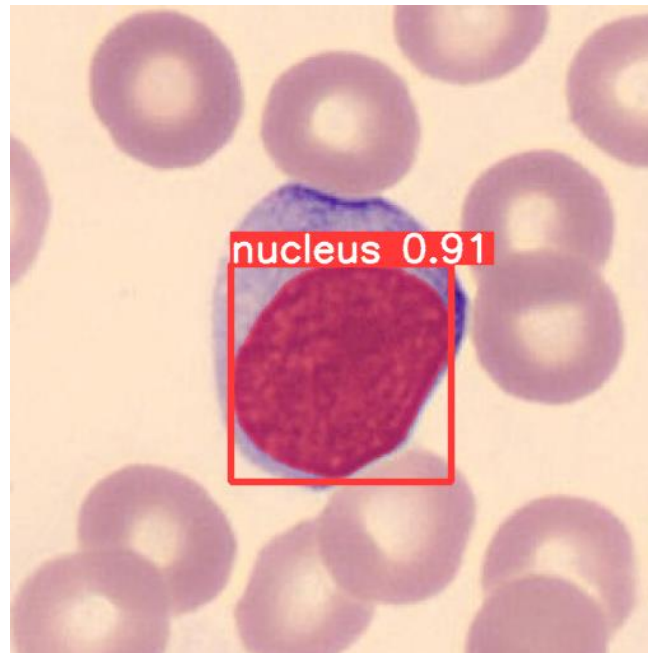
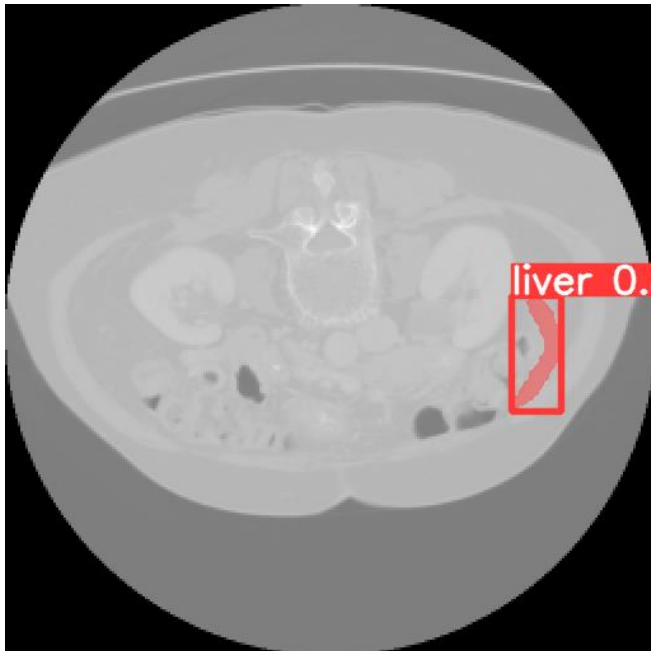
Ultralytics YOLOv8 is a cutting-edge, state-of-the-art (SOTA) model developed by Ultralytics.

It builds on the previous successful version of YOLO, introducing new features and improvements that further enhance its performance and flexibility.

Version 20230223

Applications

- The YOLOv8 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

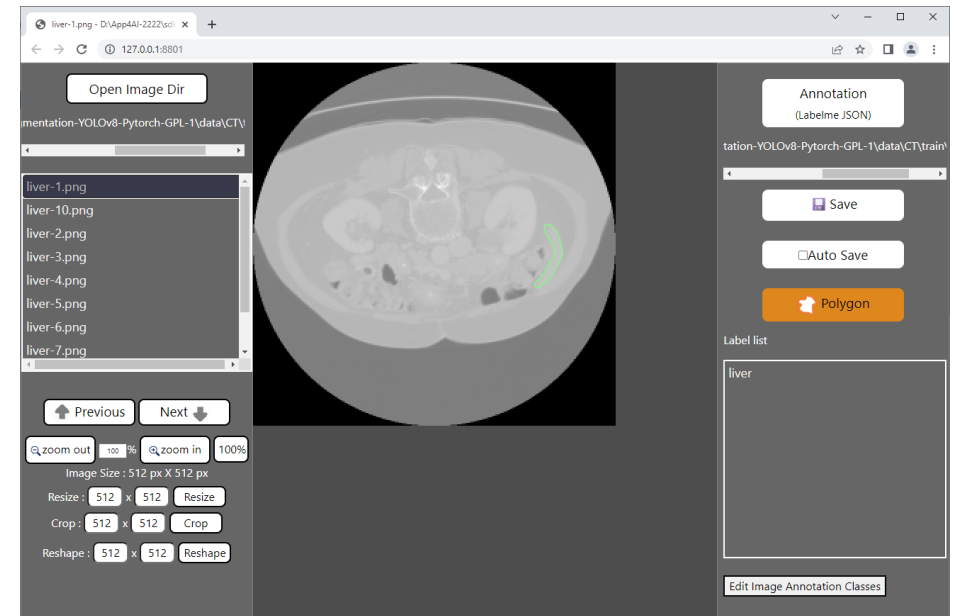
Name
data
src
1_annotation_labelme_json.ipynb
2_convert_yolo_format.ipynb
3_delete_log.ipynb
4_train.ipynb
5_tensorboard.ipynb
6_inference_image.ipynb
7_inference_image_folder_1.ipynb
8_inference_webcam.ipynb
copyright.txt
readme.txt
version.txt

1_annotation_labelme_json.ipynb

Open the webpage for image annotation.

ipynb parameter:

- “port” is the port used by the webpage. If the port is occupied by the user, please change another port value by yourself.
- “dataset” is the dataset name
- “label_folder” is the image of the train folder, it can also be changed to “val” to label the image of the val folder.



See Annotation.pdf for how to use annotation pages.

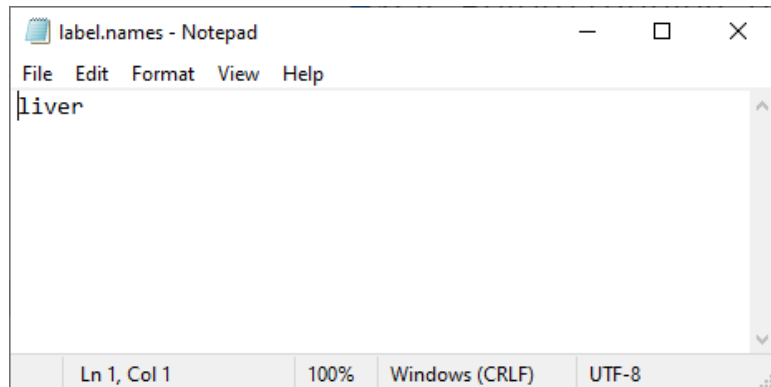
2_convert_yolo_format.ipynb

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.

supplement:

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

If there are more than two category names, represent each category name with a line break.

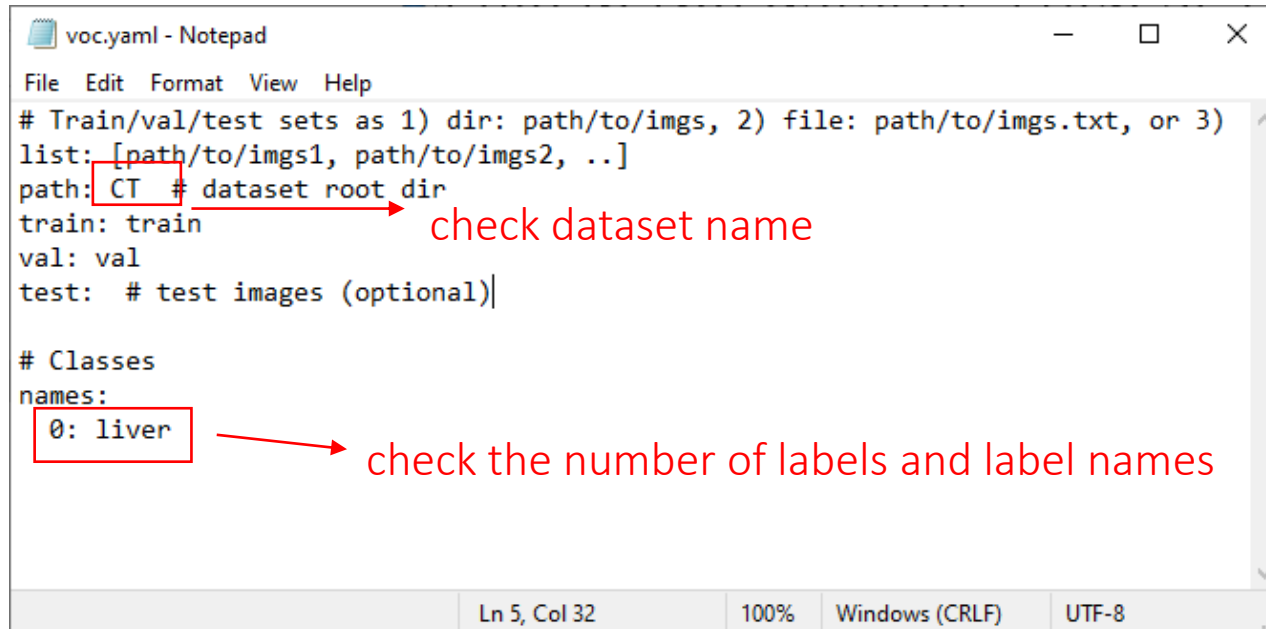


3_delete_log.ipynb

Delete the log files left over from previous training.

Set training related files

Confirm the content of the voc.yaml file in the dataset, such as the name of the dataset, the number of categories, and the name.



```
File Edit Format View Help
# 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: CT # dataset root dir
train: train
val: val
test: # test images (optional)

# Classes
names:
  0: liver
```

check dataset name

check the number of labels and label names

Ln 5, Col 32 100% Windows (CRLF) UTF-8

4_train.ipynb

Start training.

ipynb parameter:

- dataset is the dataset name.
- weights_file is the pretrained model path used,
None means not to use the pretrained model for training.
- devices is the GPU id used.
- epochs is the number of training epochs.

```
YOLOv8x-seg summary: 401 layers, 71751811 parameters, 71751795 gradients, 344.5 GFLOPs
Transferred 651/657 items from pretrained weights
optimizer: SGD(lr=0.01) with parameter groups 106 weight(decay=0.0), 117 weight(decay=0.0005), 116 bias
train: Scanning D:\App4AI-2222\sdk\Jupyter-Image-Segmentation-YOLOv8-Pytorch-GPL-1\data\CT\train\labels.cache... 10 ima
val: Scanning D:\App4AI-2222\sdk\Jupyter-Image-Segmentation-YOLOv8-Pytorch-GPL-1\data\CT\val\labels.cache... 10 images,
Image sizes 512 train, 512 val
Using 4 dataloader workers
Logging results to data\CT\model
Starting training for 1000 epochs...

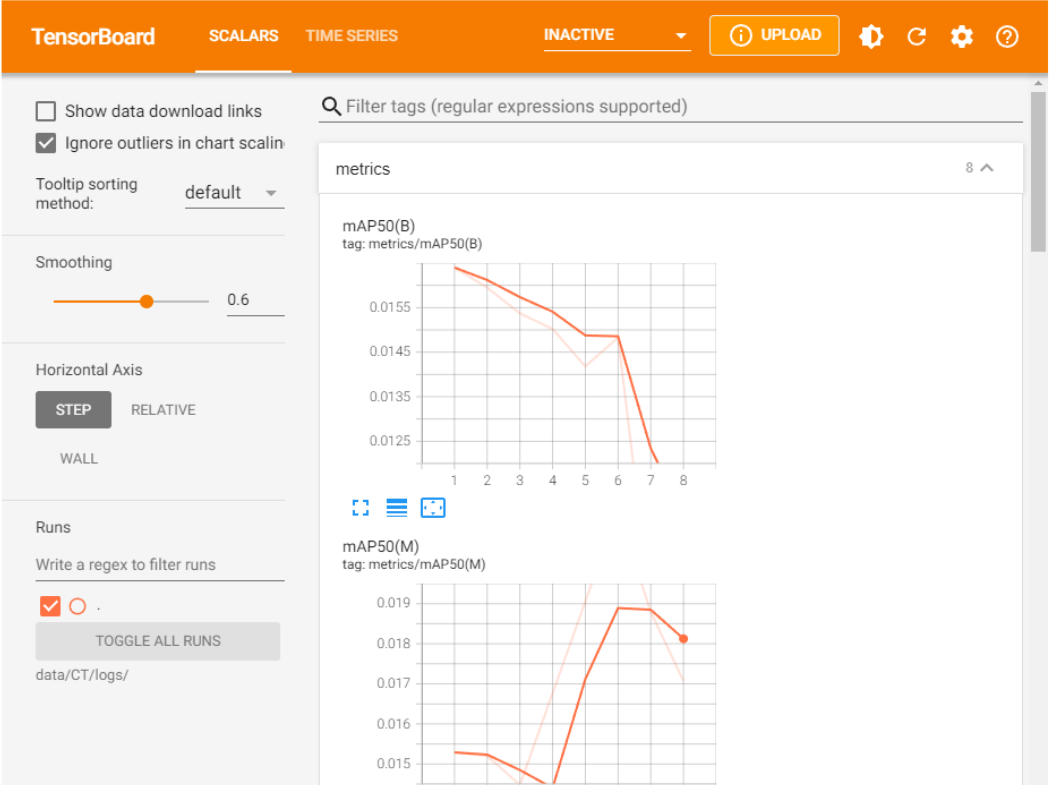
Epoch GPU_mem box_loss seg_loss cls_loss dfl_loss Instances Size
1/1000 6.43G 2.179 5.976 3.994 2.351 24 512: 100% ██████████ 1/1 [00:06<0
D:\App4AI-2222\gpu\python\lib\site-packages\torch\optim\lr_scheduler.py:131: UserWarning: Detected call of `lr_scheduler.step()` be
fore `optimizer.step()`. In PyTorch 1.1.0 and later, you should call them in the opposite order: `optimizer.step()` before `lr_sche
duler.step()`. Failure to do this will result in PyTorch skipping the first value of the learning rate schedule. See more details
at https://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 10 10 0.00233 0.7 0.0164 0.00489 0.002 0.6 0.0153 0.004
1

Epoch GPU_mem box_loss seg_loss cls_loss dfl_loss Instances Size
2/1000 6.7G 1.84 4.693 4.33 2.171 17 512: 100% ██████████ 1/1 [00:04<0
Class Images Instances Box(P R mAP50 mAP50-95 Mask(P R mAP
all 10 10 0.00233 0.7 0.016 0.00454 0.002 0.6 0.0152 0.0041
```


5_tensorboard.ipynb

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

```
[1]: # Copyright © 2023 LEADERG Inc. All rights reserved. Please keep it private. Publish to internet is not allowed.
[2]: import os
[3]: res = os.system('taskkill /IM "tensorboard.exe" /F')
    res = os.system('del /q %TMP%\tensorboard-info*')
[4]: %load_ext tensorboard
[5]: # If timeout, please execute "Kernel -> Restart Kernel and Run ALL Cells".
[6]: tensorboard --logdir=data/CT/logs/ --port 6066
```



The screenshot displays the TensorBoard web interface. The top navigation bar is orange and contains the text "TensorBoard" and "SCALARS TIME SERIES INACTIVE". Below the navigation bar, there are several settings and controls:

- Show data download links
- Ignore outliers in chart scaling
- Tooltip sorting method: default
- Smoothing: 0.6
- Horizontal Axis: STEP (selected), RELATIVE, WALL
- Runs: Write a regex to filter runs. A red checkmark and a red circle are visible. A "TOGGLE ALL RUNS" button is present.

The main content area shows two line charts:

- mAP50(B)**: tag: metrics/mAP50(B). The y-axis ranges from 0.0125 to 0.0155. The x-axis ranges from 1 to 8. The chart shows a red line that starts at approximately 0.0155 at x=1, decreases to about 0.0145 at x=5, and then drops sharply to about 0.0125 at x=7.
- mAP50(M)**: tag: metrics/mAP50(M). The y-axis ranges from 0.015 to 0.019. The x-axis ranges from 1 to 8. The chart shows a red line that starts at approximately 0.0155 at x=1, dips to about 0.0145 at x=5, then rises to about 0.0185 at x=6, and ends at about 0.018 at x=8.

6_inference_image.ipynb

Infer a single image.

ipynb parameter:

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

```
[1]: # Copyright © 2023 LEADERG Inc. All rights reserved. Please keep it private. Publish to internet is not allowed.
[2]: %matplotlib inline
[3]: import matplotlib.pyplot as plt
    plt.rcParams['figure.figsize'] = (10.0, 10.0)
[4]: dataset = "CT"
    dataset_path = "data/%s" %(dataset)
    source = "data/%s/test/images/liver-1.png" %(dataset)

    image_size = 512

    weights_file = "data/%s/model/best.pt" %(dataset)

    device = "0" # 0, 1, 2, ... for Nvidia GPU or cpu for CPU
    threshold = 0.5
[5]: %run src/predict.py model=$weights_file conf=$threshold source=$source imgsz=$image_size show=True device=$device project=$dataset_
```

Ultralytics YOLOv8.0.6 Python-3.9.12 torch-1.12.0+cu113 CUDA:0 (NVIDIA TITAN RTX, 24576MiB)
Fusing layers...
YOLOv8x-seg summary: 295 layers, 71721619 parameters, 0 gradients, 343.7 GFLOPs



liver, total count=1289
liver, count=1289

7_inference_image_folder_1.ipynb

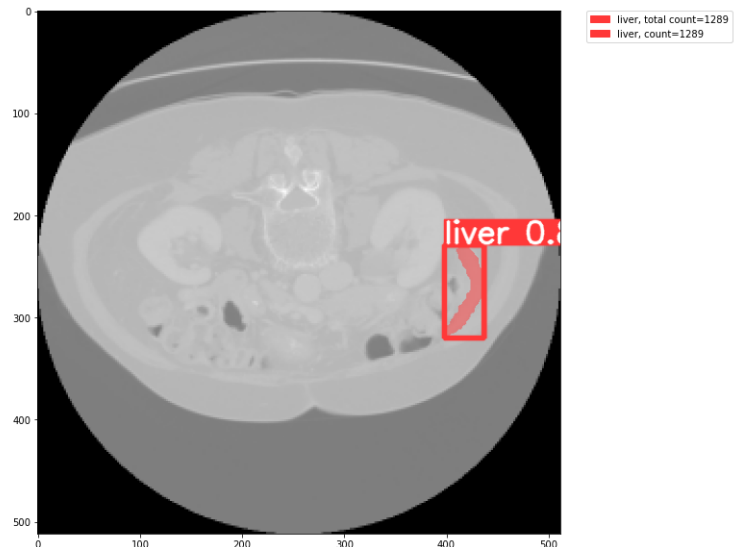
Infer all images in the folder.

ipynb parameter:

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

```
[5]: %run src/predict.py model=$weights_file conf=$threshold source=$source imgs=$image_size show=True device=$device project=$dataset_
```

Ultralytics YOLOv8.0.6 Python-3.9.12 torch-1.12.0+cu113 CUDA:0 (NVIDIA TITAN RTX, 24576MiB)
Fusing layers...
YOLOv8x-seg summary: 295 layers, 71721619 parameters, 0 gradients, 343.7 GFLOPs
Underkill Rate: 0(0.00%), Overkill Rate: 0(0.00%), Right Rate: 1(100.00%), Total: 1



liver, total count=1289
liver, count=1289

liver 0.

Image 1/10 D:\App4AI-2222\sdk\Jupyter-Image-Segmentation-YOLOv8-Pytorch-GPL-1\data\CT\test\images\liver-1.png: 512x512 1 liver, 42.0ms

8_inference_webcam.ipynb

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

It takes a while to display after running

Reference

- Please refer to the readme.txt in the SDK folder.
- LEADERG AppForAI: <https://www.leaderg.com/appforai-windows>
- Copyright © LEADERG INC. All rights reserved.