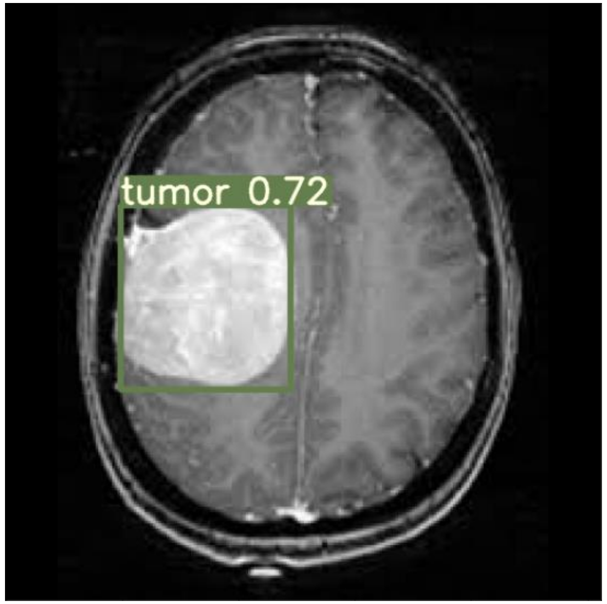
Image-Object-Detection-PyTorch-YOLOR-GPL-Jupyter

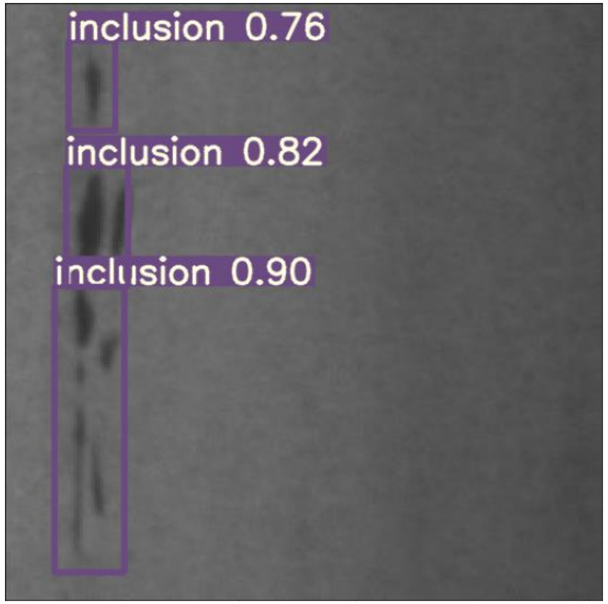
One of Taiwan's proud recent new works, YOLOR, the most powerful object detection algorithm last year, greatly reduces the amount of computation, and increases the speed without reducing the accuracy. We organized the code so that we can use JupyterLab to perform the training and inference steps in sequence, which is easier to use, and produced an instruction slideshow.

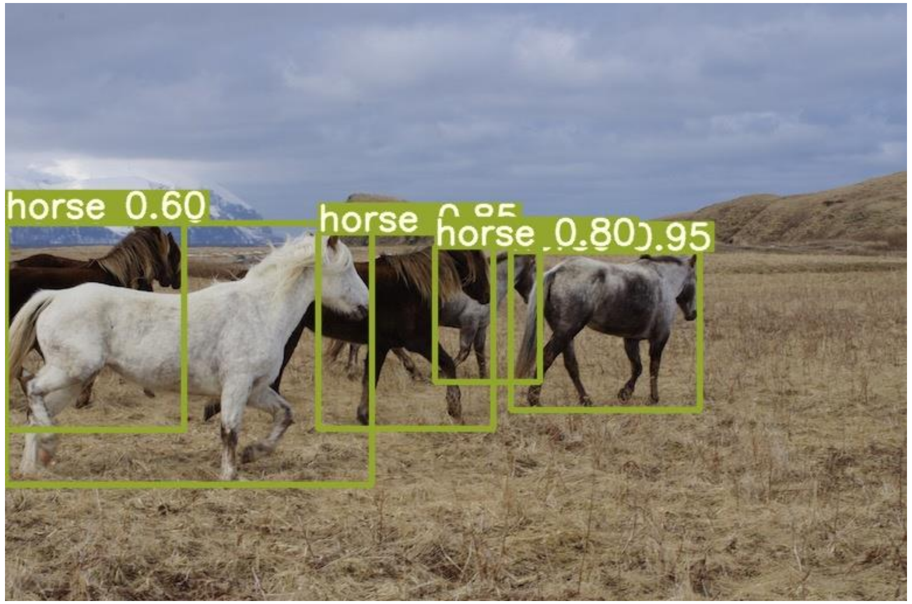
Version 20230223

Applications

- The YOLOR solution can be applied to factory defect detection, medical image analysis, biological image analysis, industrial safety image analysis, mask image analysis, etc.



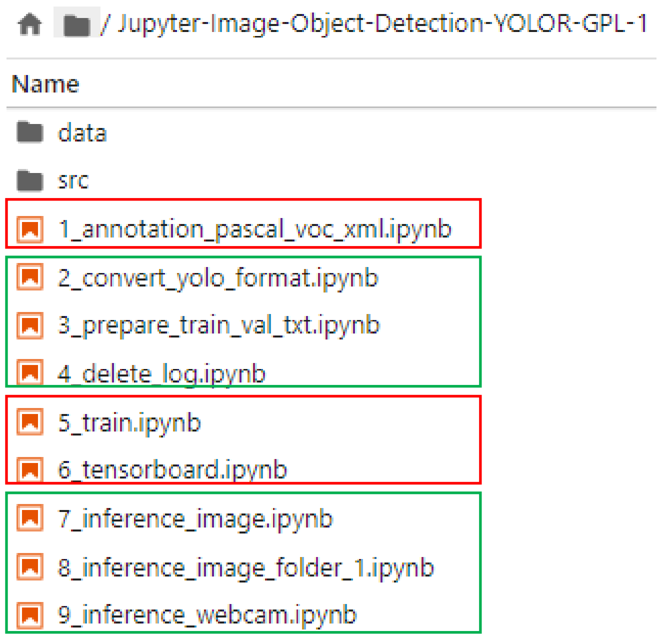




How to use

The main process is:

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



/ Jupyter-Image-Object-Detection-YOLOR-GPL-1

Name

data

src

1_annotation_pascal_voc_xml.ipynb

2_convert_yolo_format.ipynb

3_prepare_train_val_txt.ipynb

4_delete_log.ipynb

5_train.ipynb

6_tensorboard.ipynb

7_inference_image.ipynb

8_inference_image_folder_1.ipynb

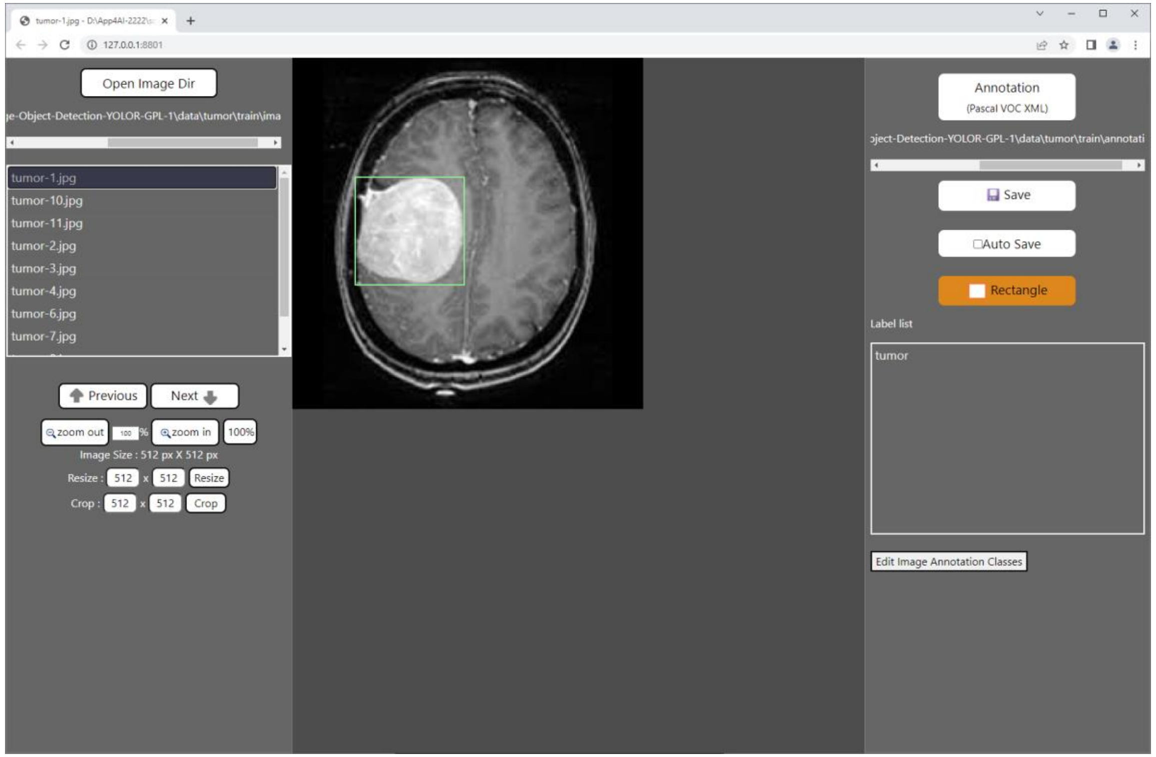
9_inference_webcam.ipynb

1_annotation_pascal_voc_xml.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.



tumor-1.jpg - D:\App4AI-2222\

127.0.0.1:8801

je-Object-Detection-YOLOR-GPL-1\data\tumor\train\ima

tumor-1.jpg
tumor-10.jpg
tumor-11.jpg
tumor-2.jpg
tumor-3.jpg
tumor-4.jpg
tumor-6.jpg
tumor-7.jpg

100 %

Image Size : 512 px X 512 px

Resize : 512 x 512

Crop : 512 x 512

ɔject-Detection-YOLOR-GPL-1\data\tumor\train\annotati

□Auto Save

Label list

tumor

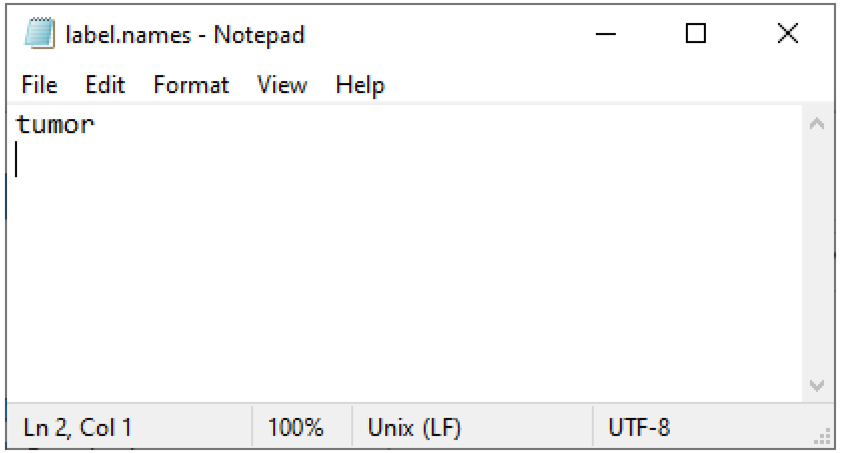
See Annotation.pdf for how to use annotation pages.

2_convert_yolo_format.ipynb

Convert the voc xml 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:

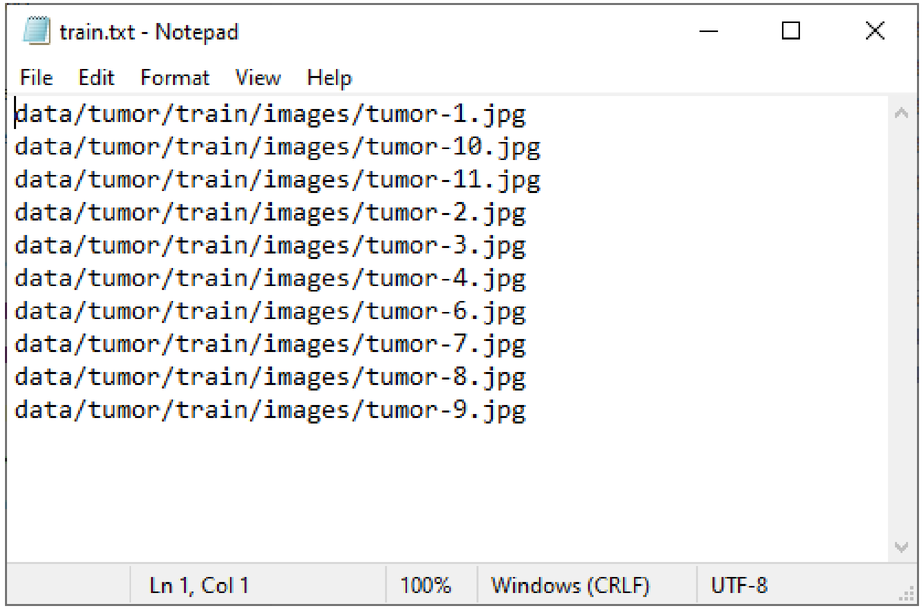
The content of label.names is the category name without background.



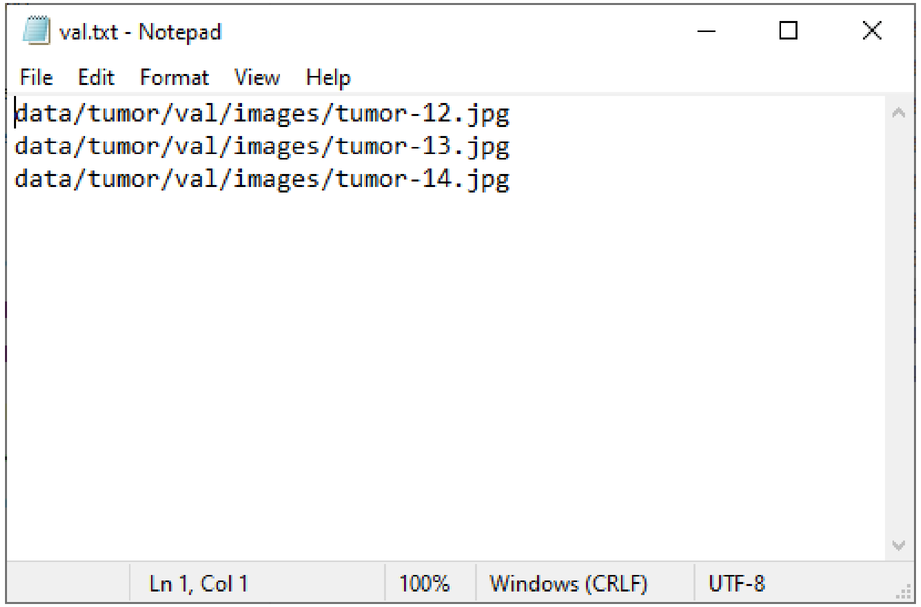
```
tumor
```

3_prepare_train_val_txt.ipynb

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



```
data/tumor/train/images/tumor-1.jpg
data/tumor/train/images/tumor-10.jpg
data/tumor/train/images/tumor-11.jpg
data/tumor/train/images/tumor-2.jpg
data/tumor/train/images/tumor-3.jpg
data/tumor/train/images/tumor-4.jpg
data/tumor/train/images/tumor-6.jpg
data/tumor/train/images/tumor-7.jpg
data/tumor/train/images/tumor-8.jpg
data/tumor/train/images/tumor-9.jpg
```



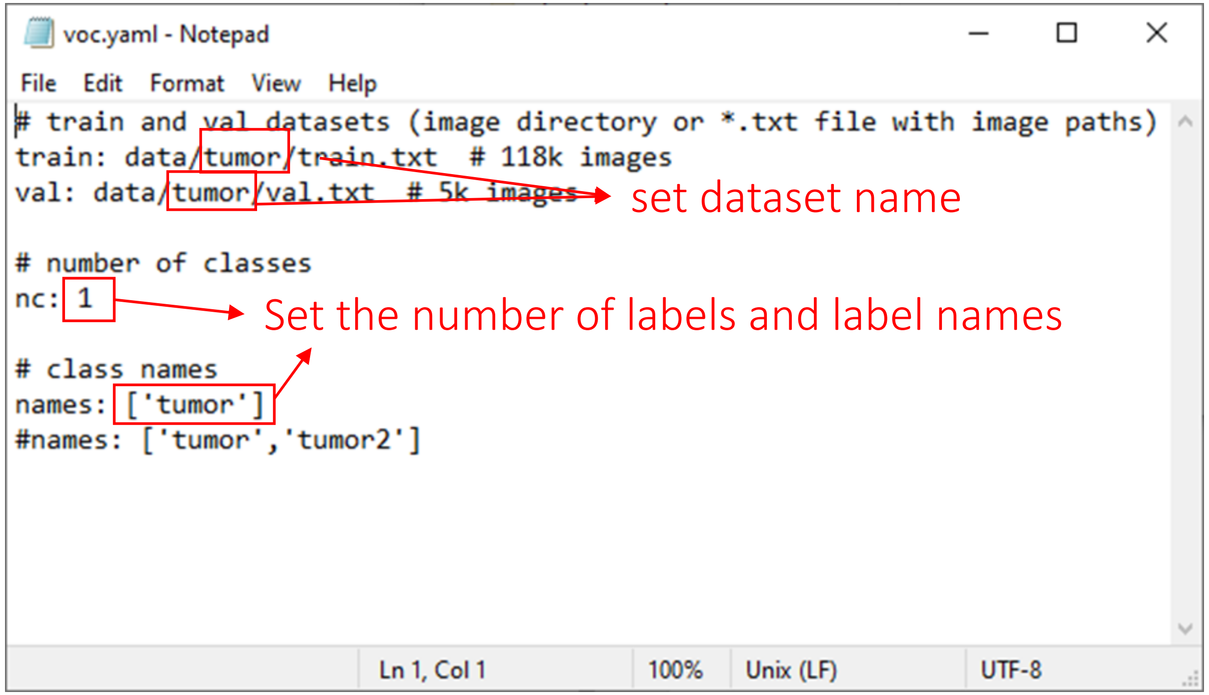
```
data/tumor/val/images/tumor-12.jpg
data/tumor/val/images/tumor-13.jpg
data/tumor/val/images/tumor-14.jpg
```

4_delete_log.ipynb

Delete the log files left over from previous training.

Set the training voc.yaml related parameters

Set the content of the voc.yaml file in the dataset, set the name of the data set, the number of categories and the name.

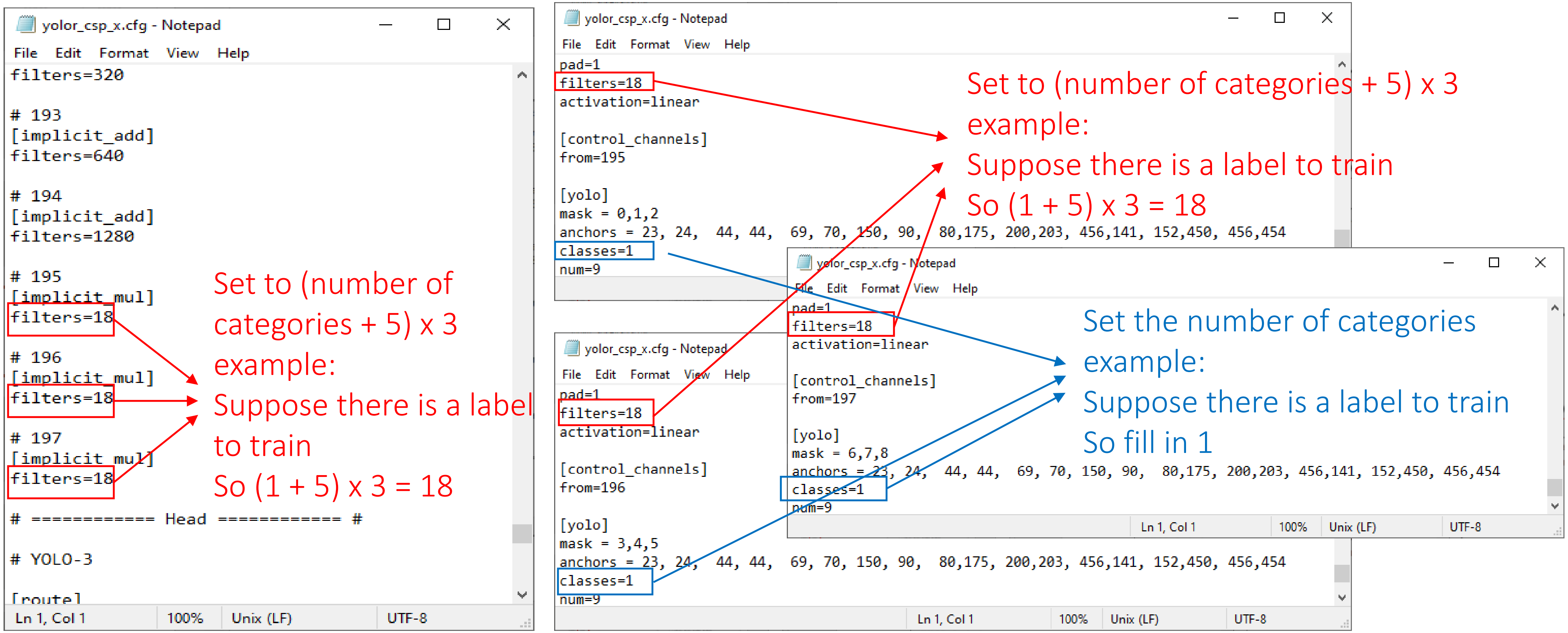

voc.yaml - Notepad
File Edit Format View Help
Ln 1, Col 1
100%
Unix (LF)
UTF-8

```
# train and val datasets (image directory or *.txt file with image paths)
train: data/tumor/train.txt  # 118k images
val: data/tumor/val.txt  # 5k images

# number of classes
nc: 1

# class names
names: ['tumor']
#names: ['tumor','tumor2']
```

set dataset name
Set the number of labels and label names

Set training yolor_csp_x.cfg related parameters

Set the content of the yolor_csp_x.cfg file in the dataset, set the number of categories.



```
filters=320

# 193
[implicit_add]
filters=640

# 194
[implicit_add]
filters=1280

# 195
[implicit_mul]
filters=18

# 196
[implicit_mul]
filters=18

# 197
[implicit_mul]
filters=18

# ============ Head ============ #

# YOLO-3

[route]
```

Set to (number of categories + 5) x 3
example:
Suppose there is a label to train
So (1 + 5) x 3 = 18

```
pad=1
filters=18
activation=linear

[control_channels]
from=195

[yolo]
mask = 0,1,2
anchors = 23, 24,  44, 44,  69, 70, 150, 90,  80,175, 200,203, 456,141, 152,450, 456,454
classes=1
num=9
```

```
pad=1
filters=18
activation=linear

[control_channels]
from=196

[yolo]
mask = 3,4,5
anchors = 23, 24,  44, 44,  69, 70, 150, 90,  80,175, 200,203, 456,141, 152,450, 456,454
classes=1
num=9
```

```
pad=1
filters=18
activation=linear

[control_channels]
from=197

[yolo]
mask = 6,7,8
anchors = 23, 24,  44, 44,  69, 70, 150, 90,  80,175, 200,203, 456,141, 152,450, 456,454
classes=1
num=9
```

Set to (number of categories + 5) x 3
example:
Suppose there is a label to train
So (1 + 5) x 3 = 18

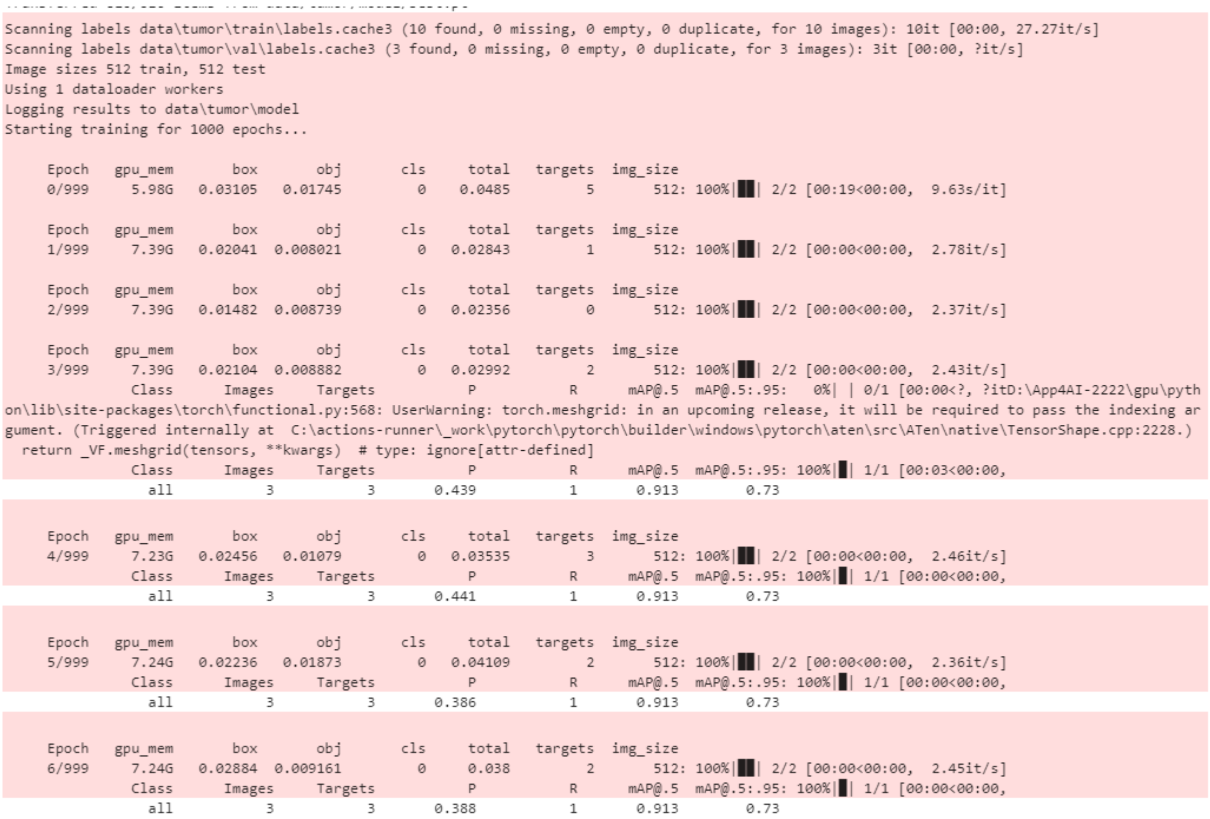
Set the number of categories
example:
Suppose there is a label to train
So fill in 1

5_train.ipynb

Start training.

ipynb parameter:

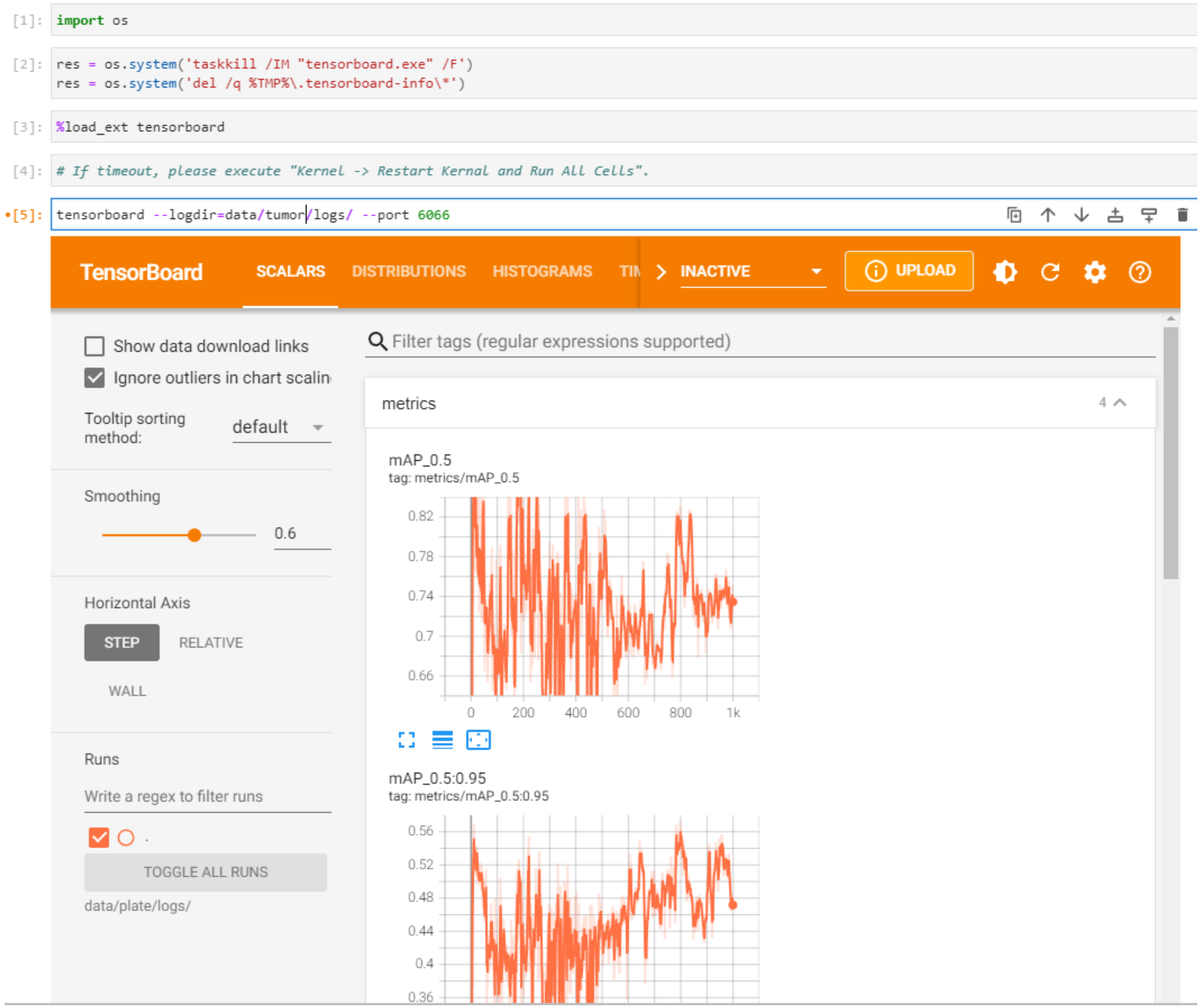
- dataset is the dataset name.
- weights_file is the pretrained model path used, None means not to use the pretrained model for training.
- devices is the GPU id used.
- epochs is the number of training epochs.



```
Scanning labels data\tumor\train\labels.cache3 (10 found, 0 missing, 0 empty, 0 duplicate, for 10 images): 10it [00:00, 27.27it/s]
Scanning labels data\tumor\val\labels.cache3 (3 found, 0 missing, 0 empty, 0 duplicate, for 3 images): 3it [00:00, ?it/s]
Image sizes 512 train, 512 test
Using 1 dataloader workers
Logging results to data\tumor\model
Starting training for 1000 epochs...

     Epoch   gpu_mem       box       obj       cls     total   targets  img_size
     0/999     5.98G   0.03105   0.01745         0    0.0485         5       512: 100%|██| 2/2 [00:19<00:00,  9.63s/it]

     Epoch   gpu_mem       box       obj       cls     total   targets  img_size
     1/999     7.39G   0.02041  0.008021         0   0.02843         1       512: 100%|██| 2/2 [00:00<00:00,  2.78it/s]

     Epoch   gpu_mem       box       obj       cls     total   targets  img_size
     2/999     7.39G   0.01482  0.008739         0   0.02356         0       512: 100%|██| 2/2 [00:00<00:00,  2.37it/s]

     Epoch   gpu_mem       box       obj       cls     total   targets  img_size
     3/999     7.39G   0.02104  0.008882         0   0.02992         2       512: 100%|██| 2/2 [00:00<00:00,  2.43it/s]
               Class    Images   Targets         P         R    mAP@.5  mAP@.5:.95:   0%|  | 0/1 [00:00<?, ?itD:\App4AI-2222\gpu\pyth
on\lib\site-packages\torch\functional.py:568: UserWarning: torch.meshgrid: in an upcoming release, it will be required to pass the indexing ar
gument. (Triggered internally at  C:\actions-runner\_work\pytorch\pytorch\builder\windows\pytorch\aten\src\ATen\native\TensorShape.cpp:2228.)
  return _VF.meshgrid(tensors, **kwargs)  # type: ignore[attr-defined]
               Class    Images   Targets         P         R    mAP@.5  mAP@.5:.95: 100%|█| 1/1 [00:03<00:00,
                 all         3         3     0.439         1     0.913      0.73

     Epoch   gpu_mem       box       obj       cls     total   targets  img_size
     4/999     7.23G   0.02456   0.01079         0   0.03535         3       512: 100%|██| 2/2 [00:00<00:00,  2.46it/s]
               Class    Images   Targets         P         R    mAP@.5  mAP@.5:.95: 100%|█| 1/1 [00:00<00:00,
                 all         3         3     0.441         1     0.913      0.73

     Epoch   gpu_mem       box       obj       cls     total   targets  img_size
     5/999     7.24G   0.02236   0.01873         0   0.04109         2       512: 100%|██| 2/2 [00:00<00:00,  2.36it/s]
               Class    Images   Targets         P         R    mAP@.5  mAP@.5:.95: 100%|█| 1/1 [00:00<00:00,
                 all         3         3     0.386         1     0.913      0.73

     Epoch   gpu_mem       box       obj       cls     total   targets  img_size
     6/999     7.24G   0.02884  0.009161         0     0.038         2       512: 100%|██| 2/2 [00:00<00:00,  2.45it/s]
               Class    Images   Targets         P         R    mAP@.5  mAP@.5:.95: 100%|█| 1/1 [00:00<00:00,
                 all         3         3     0.388         1     0.913      0.73
```

6_tensorboard.ipynb

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



```
[1]: import os
```

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

```
[3]: %load_ext tensorboard
```

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

```
•[5]: tensorboard --logdir=data/tumor/logs/ --port 6066
```

☐ Show data download links

☑ Ignore outliers in chart scalin

Tooltip sorting method: default

Smoothing 0.6

Horizontal Axis: STEP RELATIVE WALL

Runs

Write a regex to filter runs

☑ ○ .

TOGGLE ALL RUNS

data/plate/logs/

Filter tags (regular expressions supported)

metrics 4

mAP_0.5
tag: metrics/mAP_0.5

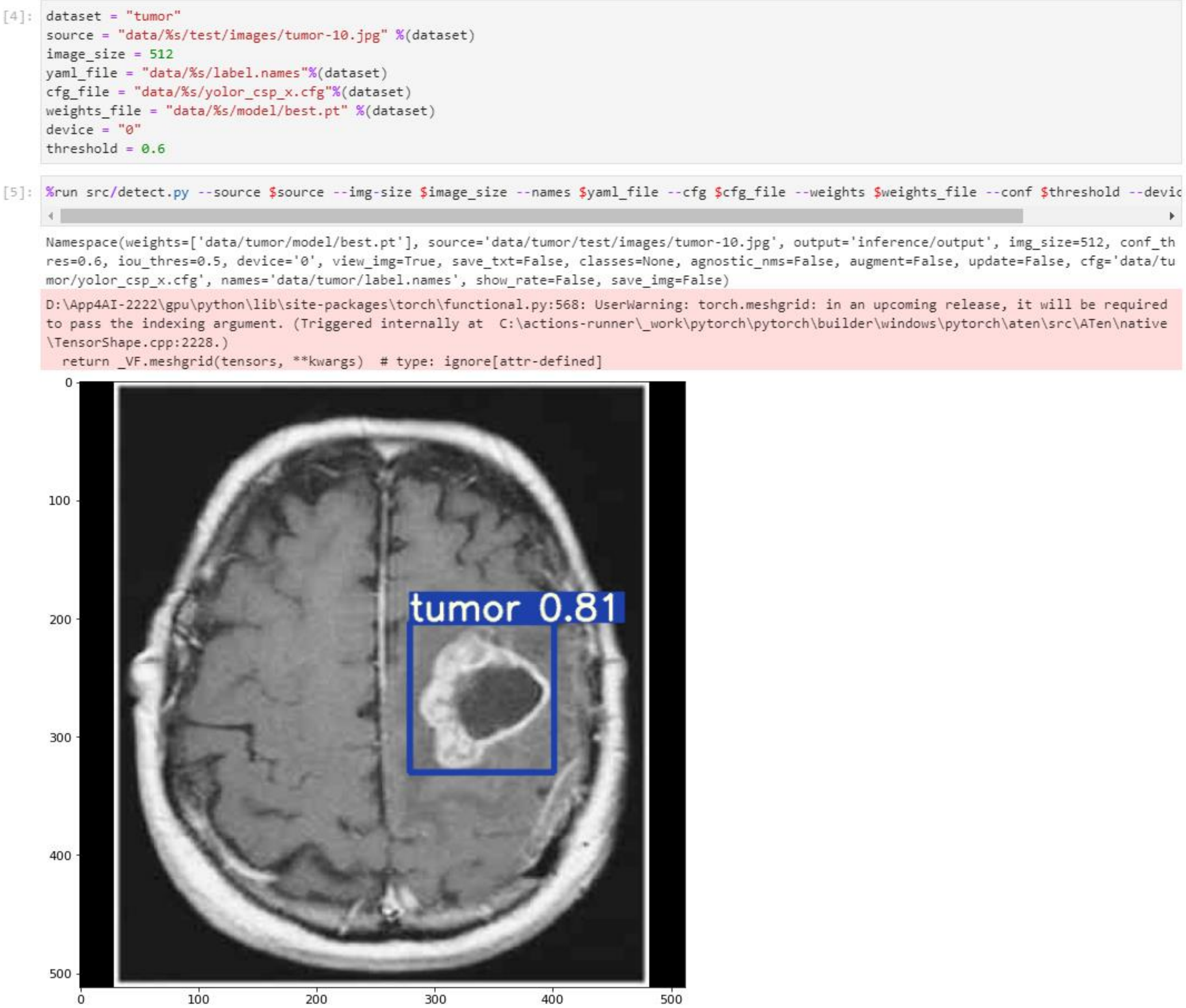
mAP_0.5:0.95
tag: metrics/mAP_0.5:0.95

7_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.



```
[4]: dataset = "tumor"
     source = "data/%s/test/images/tumor-10.jpg" %(dataset)
     image_size = 512
     yaml_file = "data/%s/label.names"%(dataset)
     cfg_file = "data/%s/yolor_csp_x.cfg"%(dataset)
     weights_file = "data/%s/model/best.pt" %(dataset)
     device = "0"
     threshold = 0.6
```

```
[5]: %run src/detect.py --source $source --img-size $image_size --names $yaml_file --cfg $cfg_file --weights $weights_file --conf $threshold --devic
```

```
Namespace(weights=['data/tumor/model/best.pt'], source='data/tumor/test/images/tumor-10.jpg', output='inference/output', img_size=512, conf_th
res=0.6, iou_thres=0.5, device='0', view_img=True, save_txt=False, classes=None, agnostic_nms=False, augment=False, update=False, cfg='data/tu
mor/yolor_csp_x.cfg', names='data/tumor/label.names', show_rate=False, save_img=False)
```

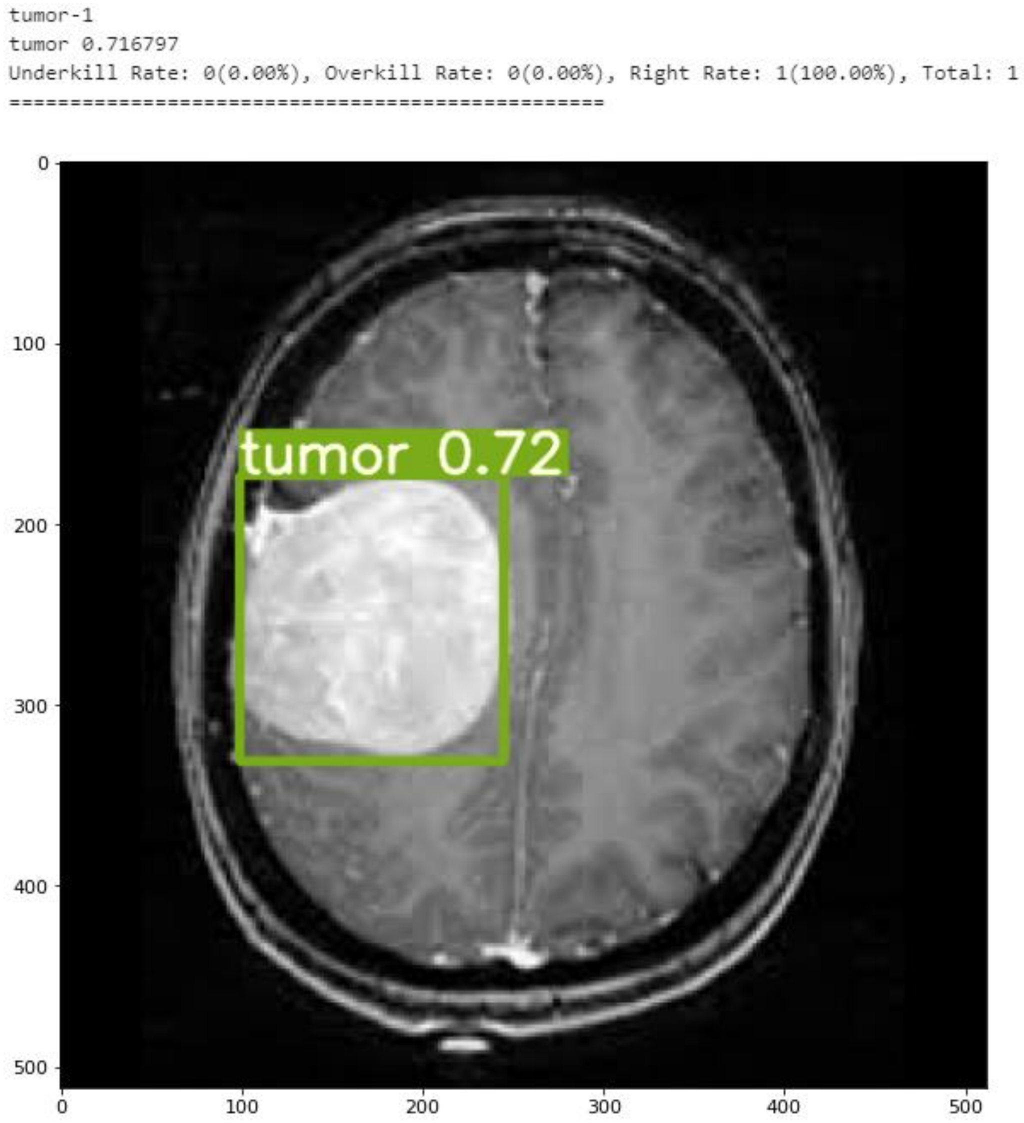
```
D:\App4AI-2222\gpu\python\lib\site-packages\torch\functional.py:568: UserWarning: torch.meshgrid: in an upcoming release, it will be required
to pass the indexing argument. (Triggered internally at  C:\actions-runner\_work\pytorch\pytorch\builder\windows\pytorch\aten\src\ATen\native
\TensorShape.cpp:2228.)
  return _VF.meshgrid(tensors, **kwargs)  # type: ignore[attr-defined]
```

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.



```
tumor-1
tumor 0.716797
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
-