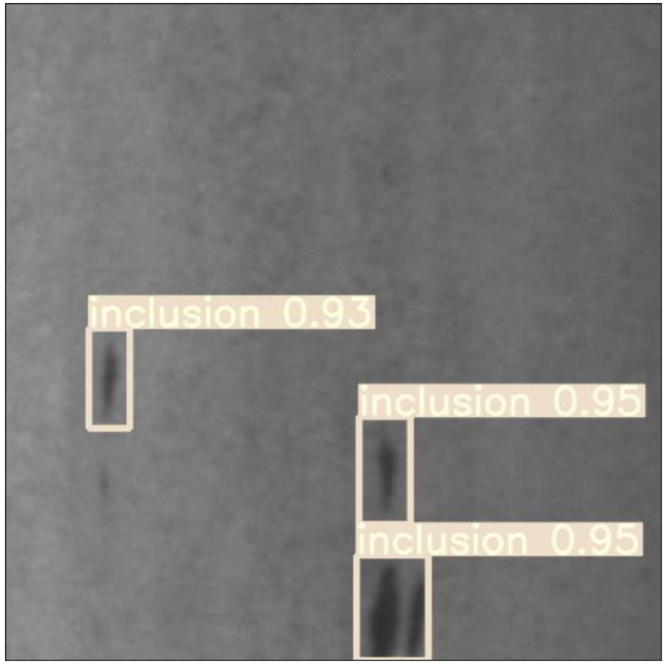
Image-Object-Detection-YOLOv7-PyTorch-GPL-Jupyter

The latest work of the pride of Taiwan, the most powerful object detection algorithm YOLOv7 at present, greatly reduces the amount of calculation, 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

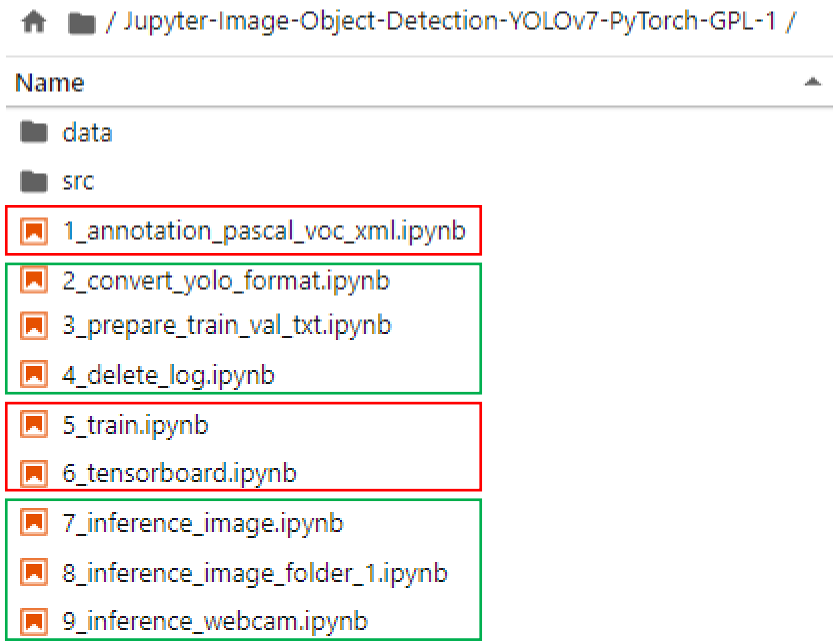
- The YOLOv7 solution can be applied to factory defect detection, medical image analysis, biological image analysis, industrial safety image analysis, mask image analysis, etc.



How to use

The main process is:

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



/ Jupyter-Image-Object-Detection-YOLOv7-PyTorch-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

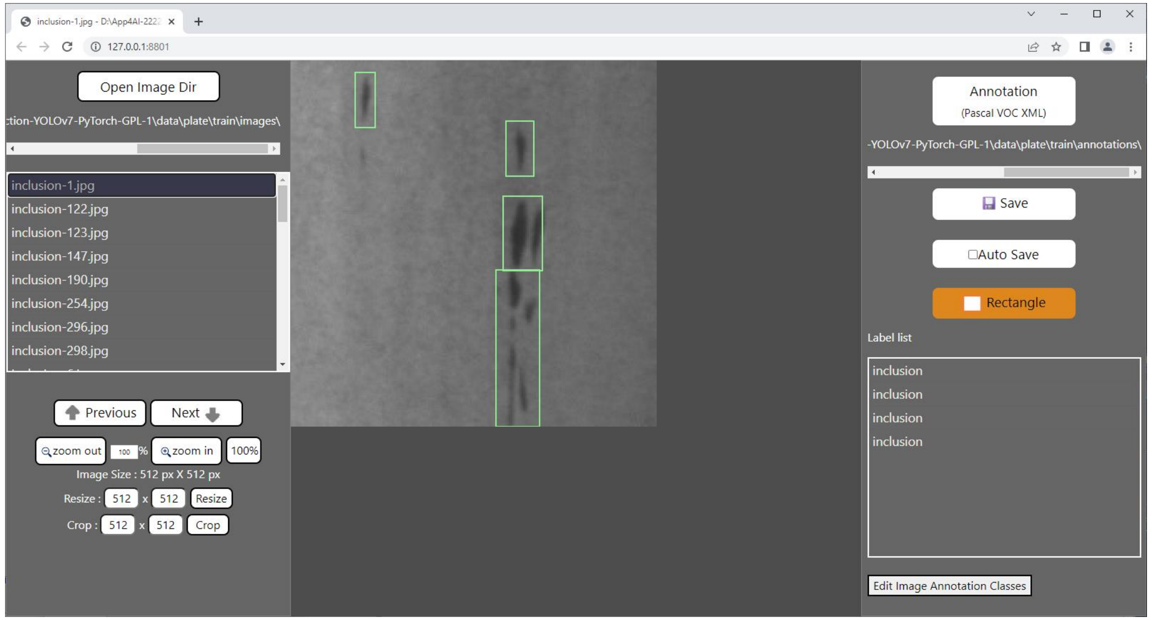
7_inference_image.ipynb

8_inference_image_folder_1.ipynb

9_inference_webcam.ipynb

1_annotation_pascal_voc_xml.ipynb

Open the webpage for image annotation.



:tion-YOLOv7-PyTorch-GPL-1\data\plate\train\images\

- inclusion-1.jpg
- inclusion-122.jpg
- inclusion-123.jpg
- inclusion-147.jpg
- inclusion-190.jpg
- inclusion-254.jpg
- inclusion-296.jpg
- inclusion-298.jpg

Image Size : 512 px X 512 px

-YOLOv7-PyTorch-GPL-1\data\plate\train\annotations\

- inclusion
- inclusion
- inclusion
- inclusion

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.

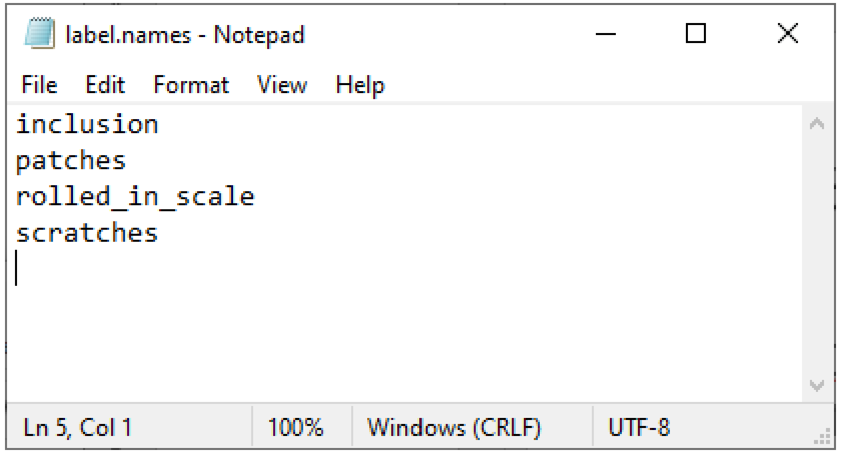
See Annotation.pdf for how to use annotation pages.

2_convert_yolo_format.ipynb

Convert the 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:

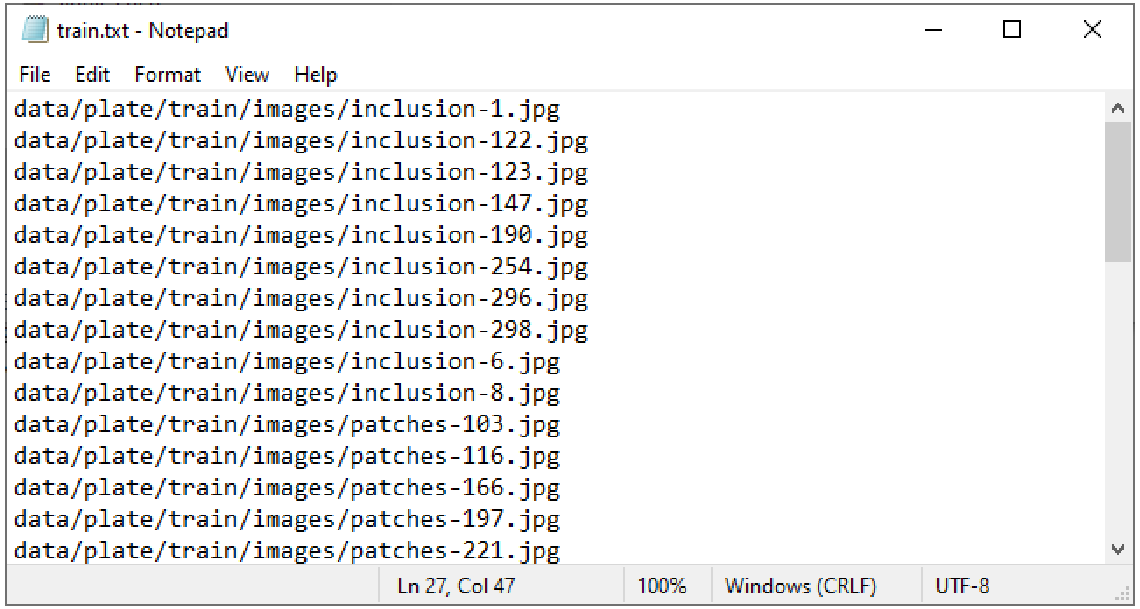
The content of label.names is the category name without background.



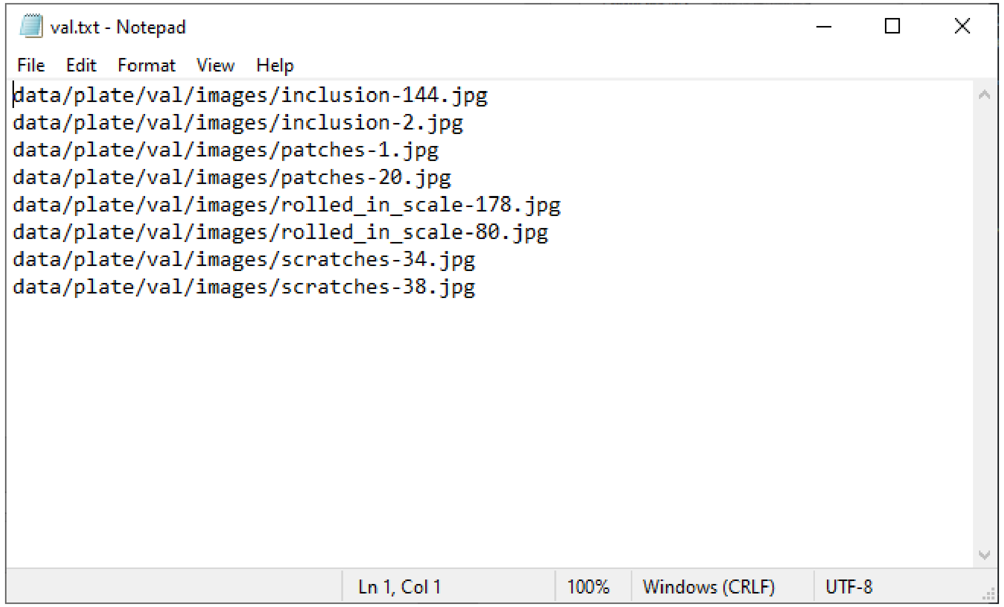
```
inclusion
patches
rolled_in_scale
scratches
```

3_prepare_train_val_txt.ipynb

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



```
data/plate/train/images/inclusion-1.jpg
data/plate/train/images/inclusion-122.jpg
data/plate/train/images/inclusion-123.jpg
data/plate/train/images/inclusion-147.jpg
data/plate/train/images/inclusion-190.jpg
data/plate/train/images/inclusion-254.jpg
data/plate/train/images/inclusion-296.jpg
data/plate/train/images/inclusion-298.jpg
data/plate/train/images/inclusion-6.jpg
data/plate/train/images/inclusion-8.jpg
data/plate/train/images/patches-103.jpg
data/plate/train/images/patches-116.jpg
data/plate/train/images/patches-166.jpg
data/plate/train/images/patches-197.jpg
data/plate/train/images/patches-221.jpg
```



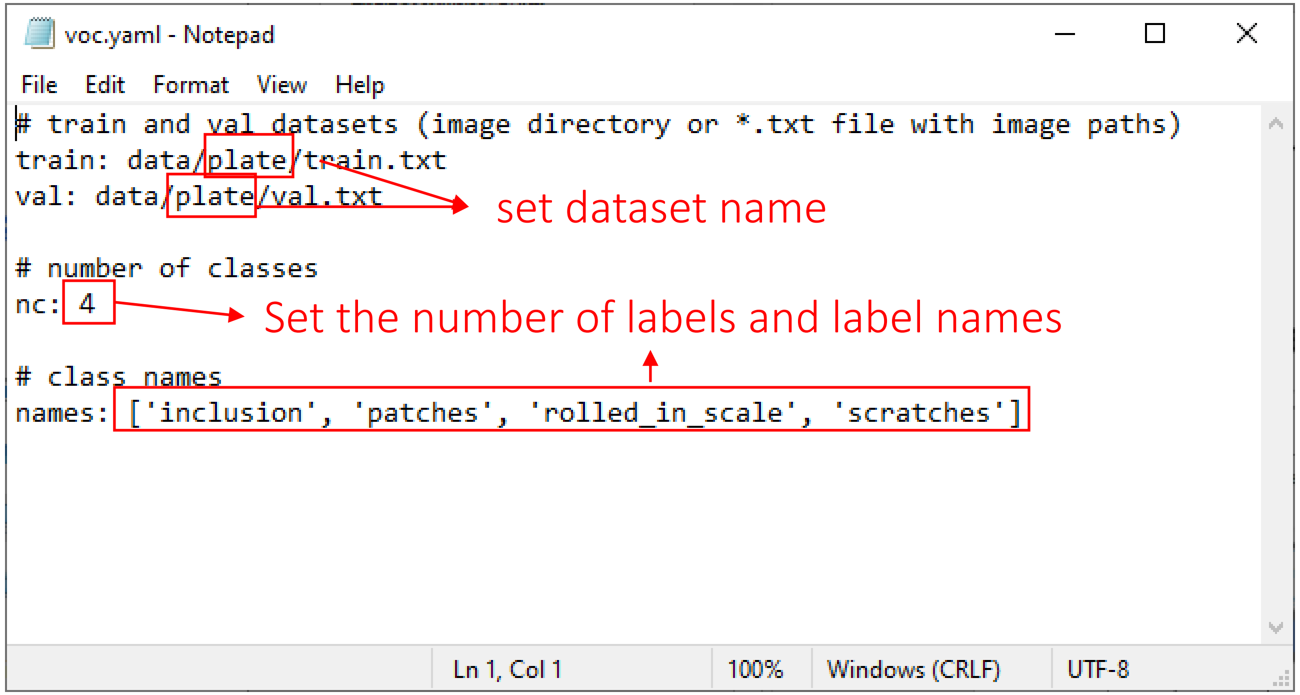
```
data/plate/val/images/inclusion-144.jpg
data/plate/val/images/inclusion-2.jpg
data/plate/val/images/patches-1.jpg
data/plate/val/images/patches-20.jpg
data/plate/val/images/rolled_in_scale-178.jpg
data/plate/val/images/rolled_in_scale-80.jpg
data/plate/val/images/scratches-34.jpg
data/plate/val/images/scratches-38.jpg
```

4_delete_log.ipynb

Delete the log files left over from previous training.

Set training related files

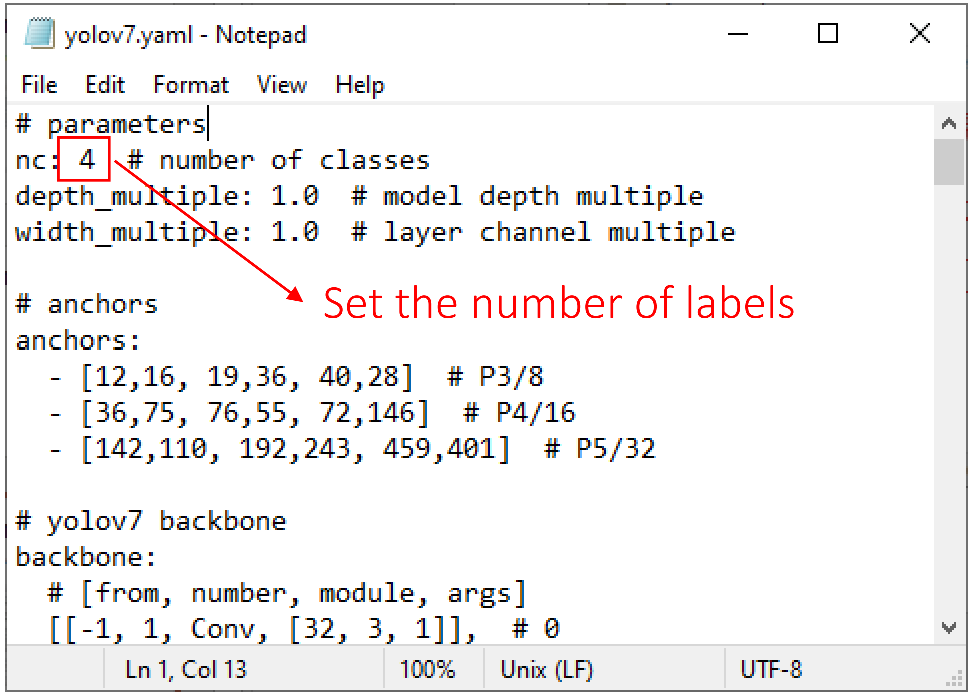
Set the content of the yolov7.yaml and voc.yaml files in the dataset, set the name of the data set, the number of categories and the name.



```yaml
# train and val datasets (image directory or *.txt file with image paths)
train: data/plate/train.txt
val: data/plate/val.txt

# number of classes
nc: 4

# class names
names: ['inclusion', 'patches', 'rolled_in_scale', 'scratches']
```

set dataset name

Set the number of labels and label names

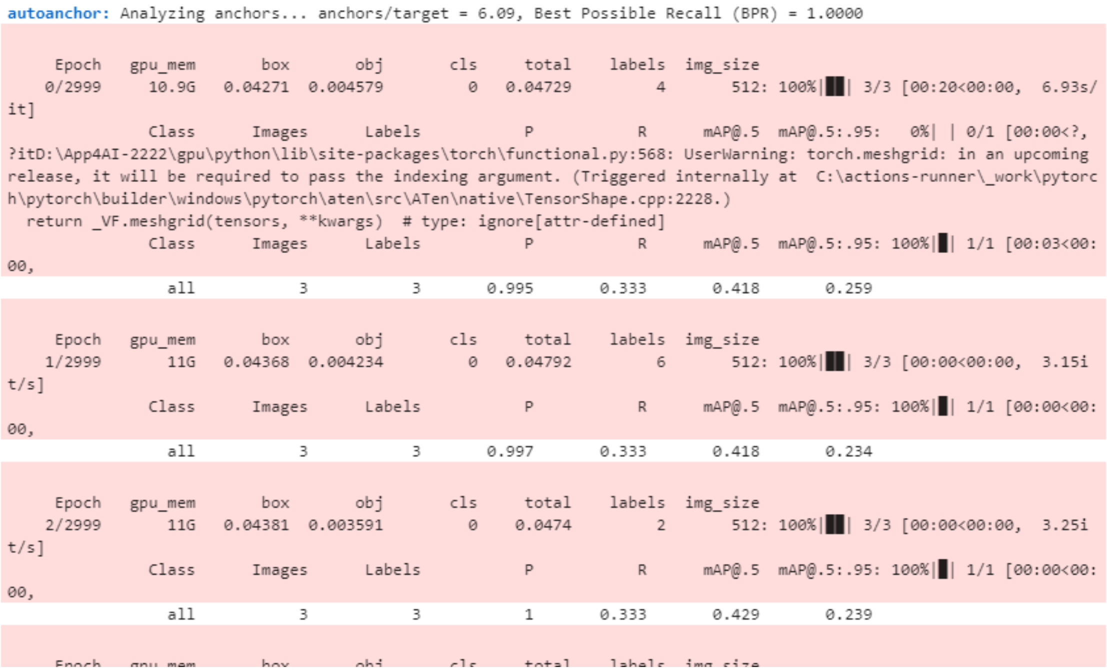


```yaml
# parameters
nc: 4  # number of classes
depth_multiple: 1.0  # model depth multiple
width_multiple: 1.0  # layer channel multiple

# anchors
anchors:
  - [12,16, 19,36, 40,28]  # P3/8
  - [36,75, 76,55, 72,146]  # P4/16
  - [142,110, 192,243, 459,401]  # P5/32

# yolov7 backbone
backbone:
  # [from, number, module, args]
  [[-1, 1, Conv, [32, 3, 1]],  # 0
```

Set the number of labels

5_train.ipynb

Start training.



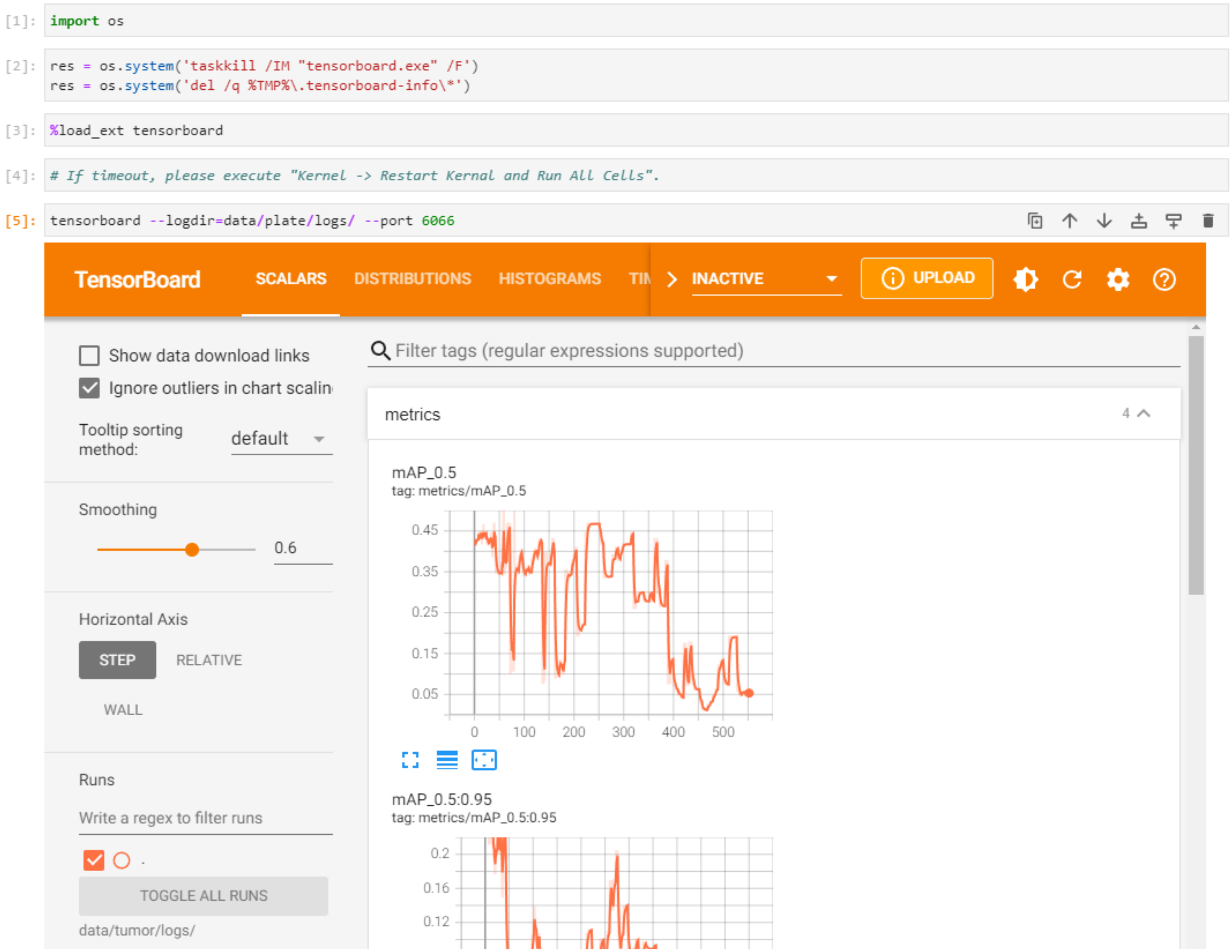
```
autoanchor: Analyzing anchors... anchors/target = 6.09, Best Possible Recall (BPR) = 1.0000

     Epoch   gpu_mem       box       obj       cls     total    labels  img_size
    0/2999     10.9G   0.04271  0.004579         0   0.04729         4       512: 100%|██| 3/3 [00:20<00:00,  6.93s/
it]
               Class     Images     Labels          P          R     mAP@.5  mAP@.5:.95:   0%|  | 0/1 [00:00<?,
?itD:\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\pytorc
h\pytorch\builder\windows\pytorch\aten\src\ATen\native\TensorShape.cpp:2228.)
  return _VF.meshgrid(tensors, **kwargs)  # type: ignore[attr-defined]
               Class     Images     Labels          P          R     mAP@.5  mAP@.5:.95: 100%|█| 1/1 [00:03<00:
00,
                 all          3          3      0.995      0.333      0.418      0.259

     Epoch   gpu_mem       box       obj       cls     total    labels  img_size
    1/2999       11G   0.04368  0.004234         0   0.04792         6       512: 100%|██| 3/3 [00:00<00:00,  3.15i
t/s]
               Class     Images     Labels          P          R     mAP@.5  mAP@.5:.95: 100%|█| 1/1 [00:00<00:
00,
                 all          3          3      0.997      0.333      0.418      0.234

     Epoch   gpu_mem       box       obj       cls     total    labels  img_size
    2/2999       11G   0.04381  0.003591         0    0.0474         2       512: 100%|██| 3/3 [00:00<00:00,  3.25i
t/s]
               Class     Images     Labels          P          R     mAP@.5  mAP@.5:.95: 100%|█| 1/1 [00:00<00:
00,
                 all          3          3          1      0.333      0.429      0.239

     Epoch   gpu_mem       box       obj       cls     total    labels  img_size
```

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.

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/plate/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/tumor/logs/

Filter tags (regular expressions supported)

metrics 4

mAP_0.5
tag: metrics/mAP_0.5

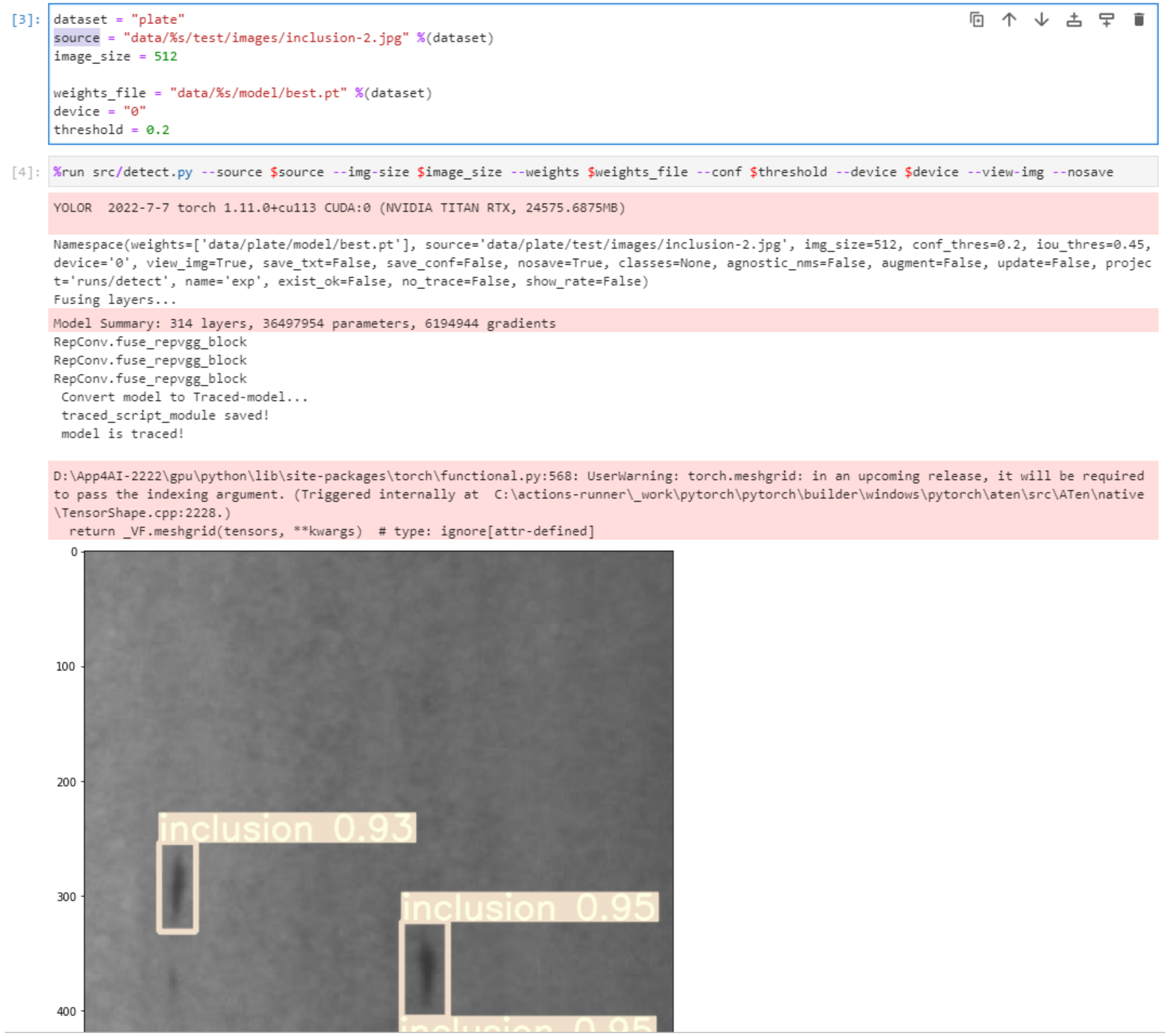
mAP_0.5:0.95
tag: metrics/mAP_0.5:0.95

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.



```
[3]: dataset = "plate"
     source = "data/%s/test/images/inclusion-2.jpg" %(dataset)
     image_size = 512

     weights_file = "data/%s/model/best.pt" %(dataset)
     device = "0"
     threshold = 0.2
```

```
[4]: %run src/detect.py --source $source --img-size $image_size --weights $weights_file --conf $threshold --device $device --view-img --nosave
```

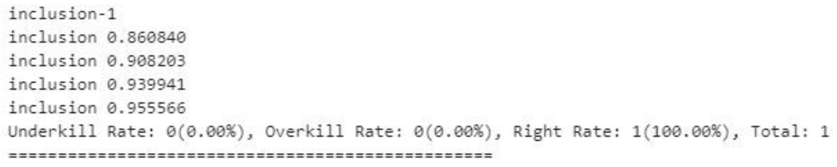
```
YOLOR  2022-7-7 torch 1.11.0+cu113 CUDA:0 (NVIDIA TITAN RTX, 24575.6875MB)

Namespace(weights=['data/plate/model/best.pt'], source='data/plate/test/images/inclusion-2.jpg', img_size=512, conf_thres=0.2, iou_thres=0.45,
device='0', view_img=True, save_txt=False, save_conf=False, nosave=True, classes=None, agnostic_nms=False, augment=False, update=False, projec
t='runs/detect', name='exp', exist_ok=False, no_trace=False, show_rate=False)
Fusing layers...
Model Summary: 314 layers, 36497954 parameters, 6194944 gradients
RepConv.fuse_repvgg_block
RepConv.fuse_repvgg_block
RepConv.fuse_repvgg_block
 Convert model to Traced-model...
 traced_script_module saved!
 model is traced!

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

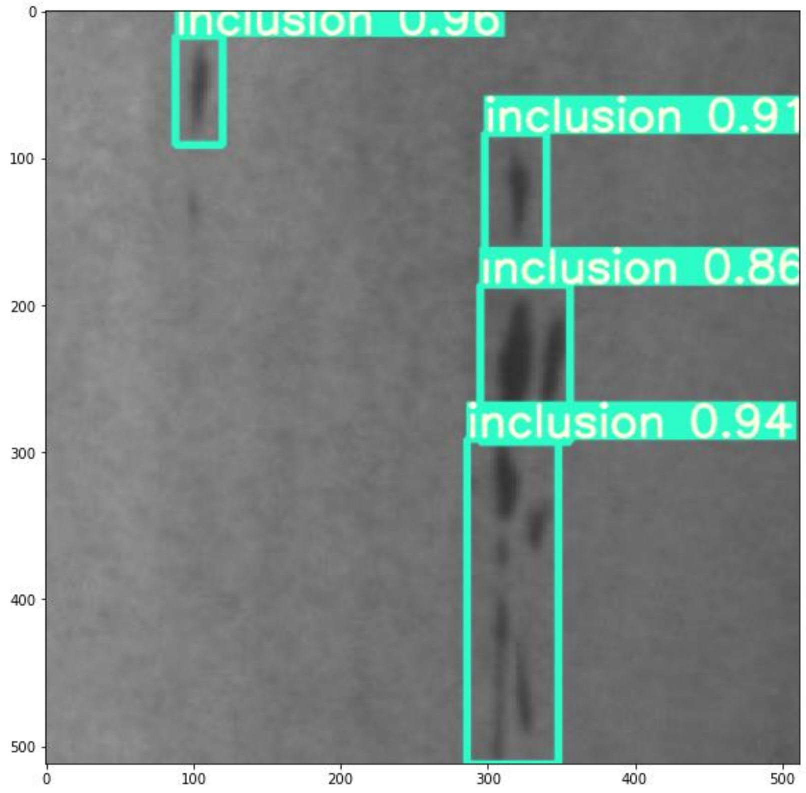
Infer all images in the folder.

ipynb parameter:

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



```
inclusion-1
inclusion 0.860840
inclusion 0.908203
inclusion 0.939941
inclusion 0.955566
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