

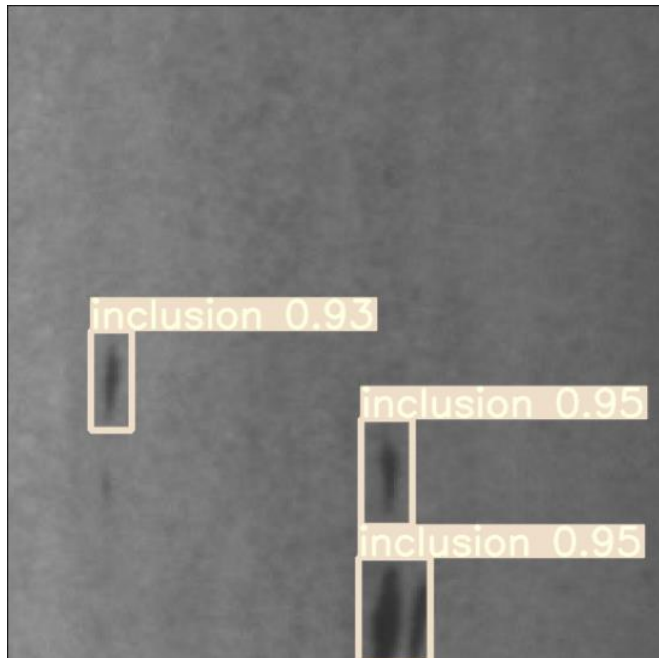
Jupyter-Image-Object-Detection- YOLOv7-PyTorch-GPL

LEADERG INC.

<https://www.leaderg.com>

Introduction

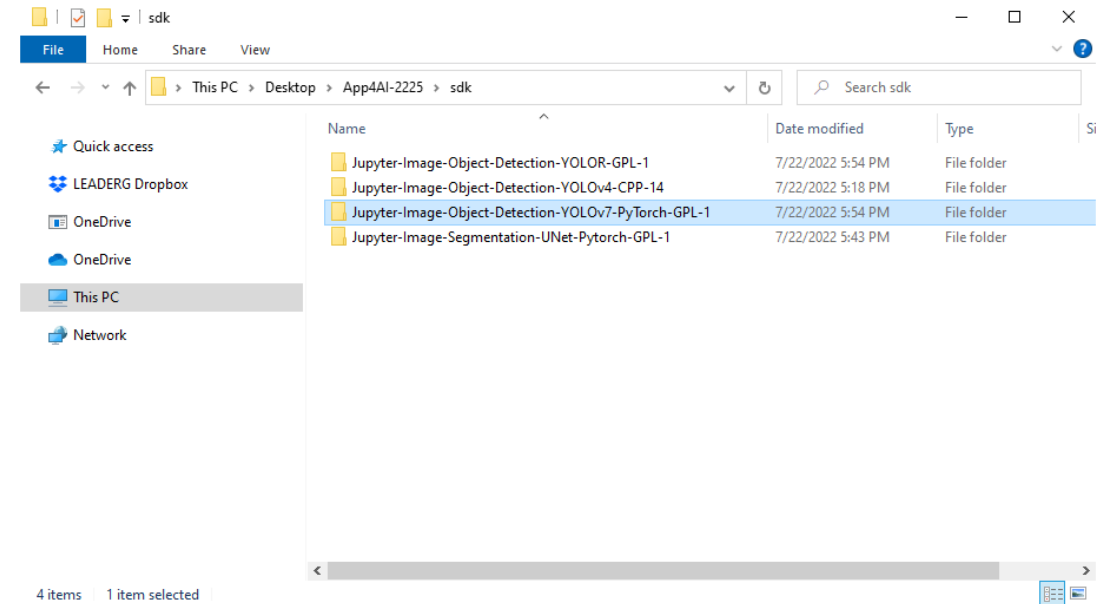
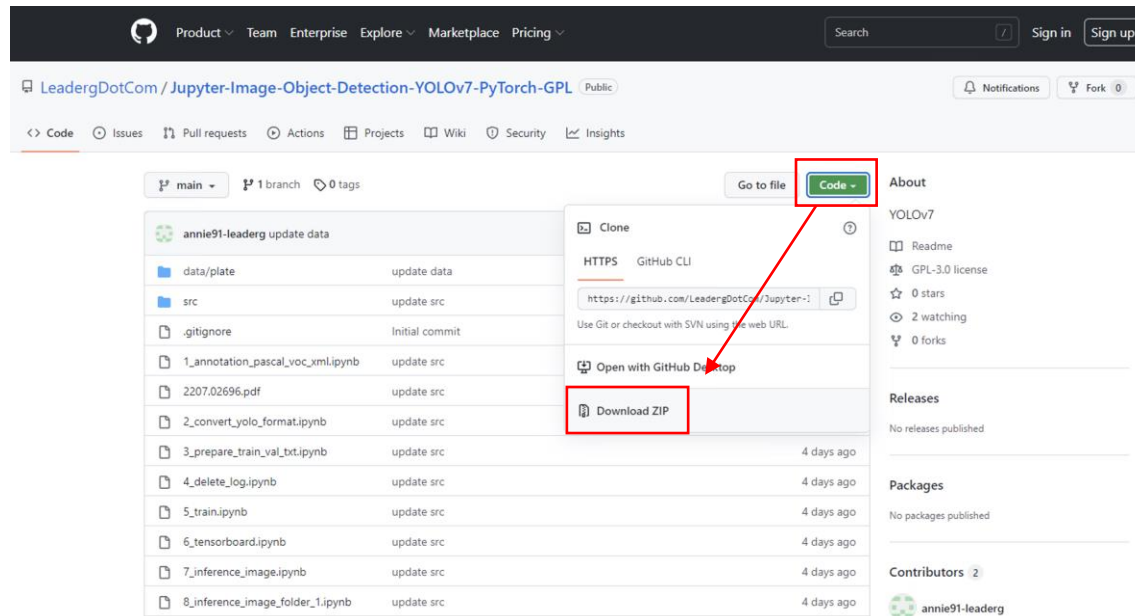
- The YOLOv7 solution can be applied to factory defect detection, medical image analysis, biological image analysis, industrial safety image analysis, mask image analysis, etc.



How to use – Download YOLOv7

Download zip file from GitHub. Put zip file to App4AI/sdk folder. Then unzip it.

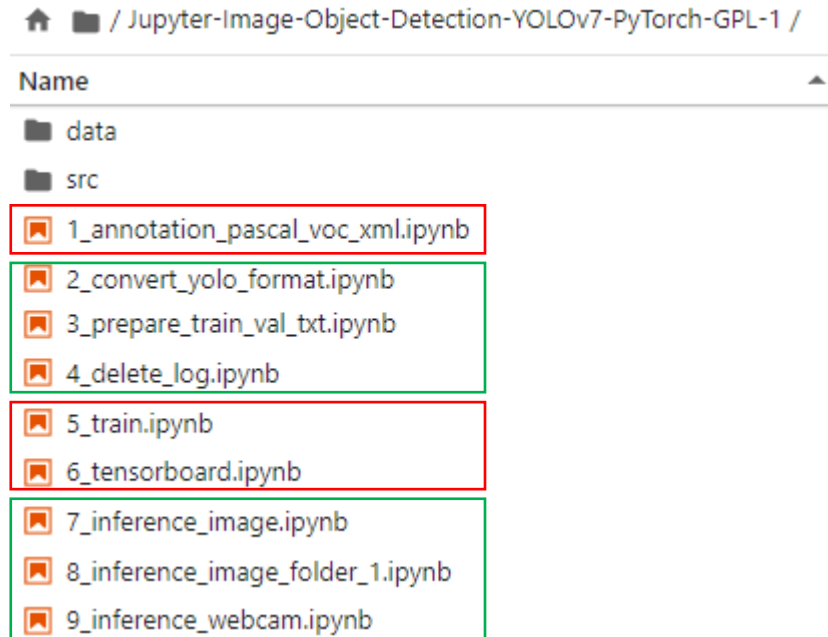
Download url: <https://github.com/LeadergDotCom/Jupyter-Image-Object-Detection-YOLOv7-PyTorch-GPL>



How to use

The main process is:

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



How to use - Annotate images

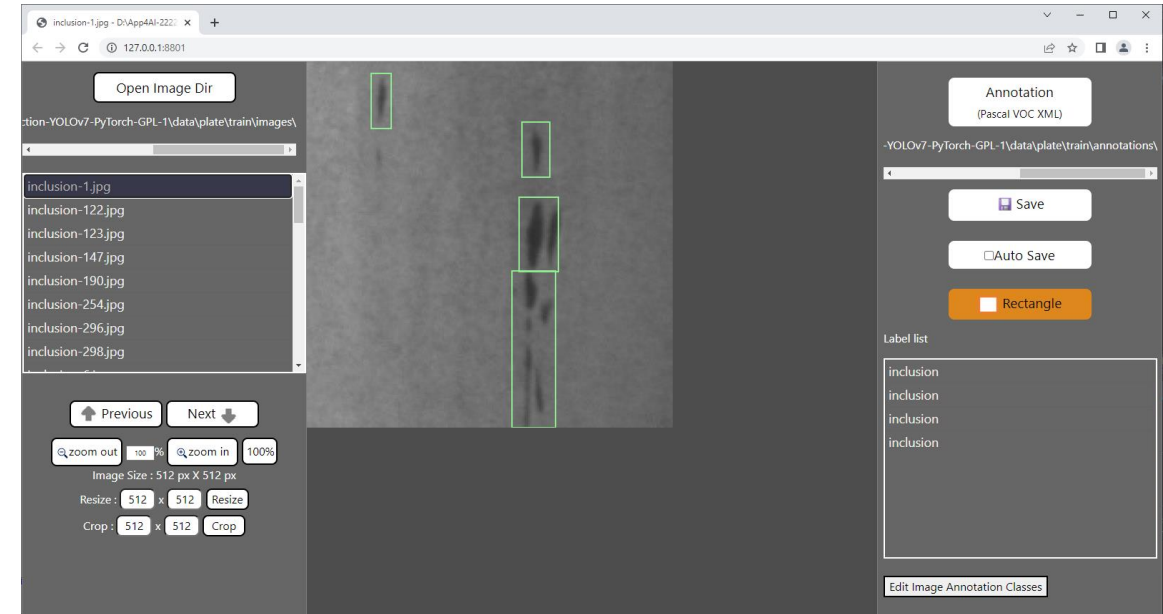
Run `1_annotation_pascal_voc_xml.ipynb`

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.

See Annotation.pdf for how to use annotation pages.



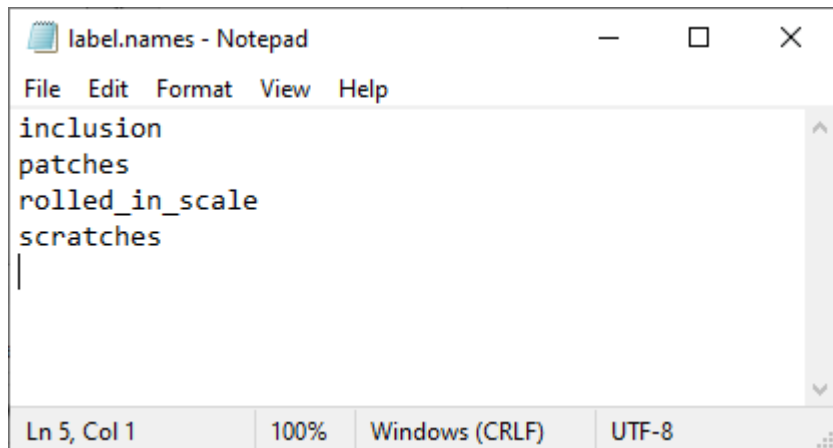
How to use - Prepare files for training

Run 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:

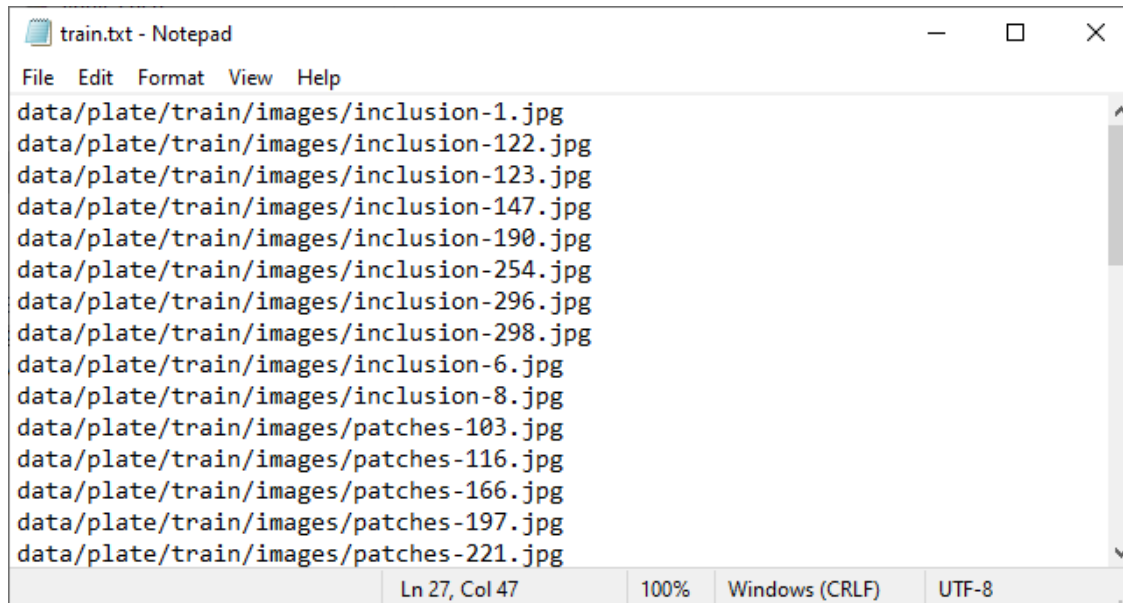
The content of label.names is the category name without background.



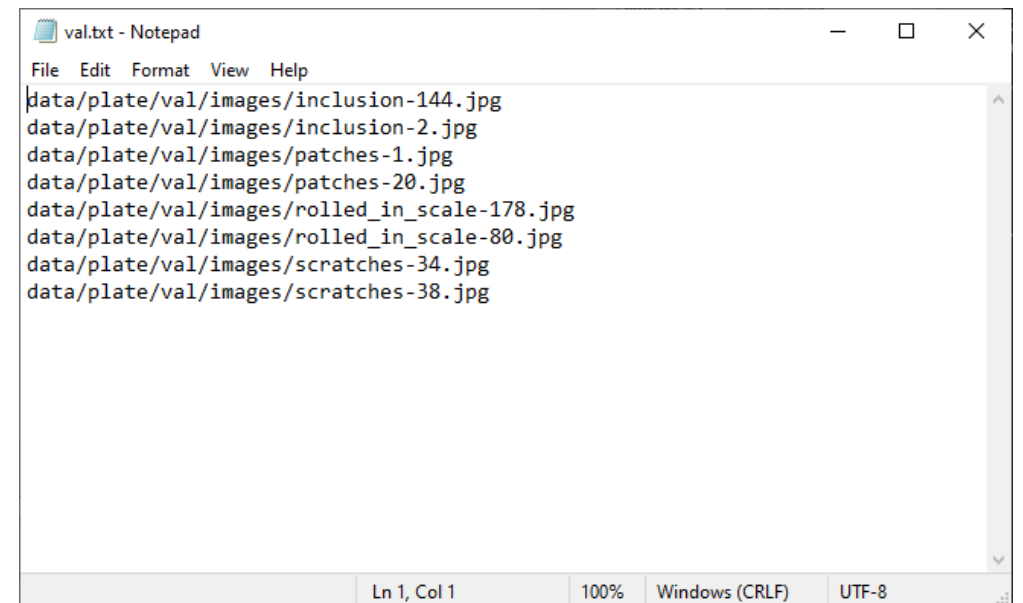
How to use - Prepare files for training

Run 3_prepare_train_val_txt.ipynb

Generate training image path and verification image path files for training.



```
train.txt - Notepad
File Edit Format View Help
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
Ln 27, Col 47 100% Windows (CRLF) UTF-8
```



```
val.txt - Notepad
File Edit Format View Help
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
Ln 1, Col 1 100% Windows (CRLF) UTF-8
```

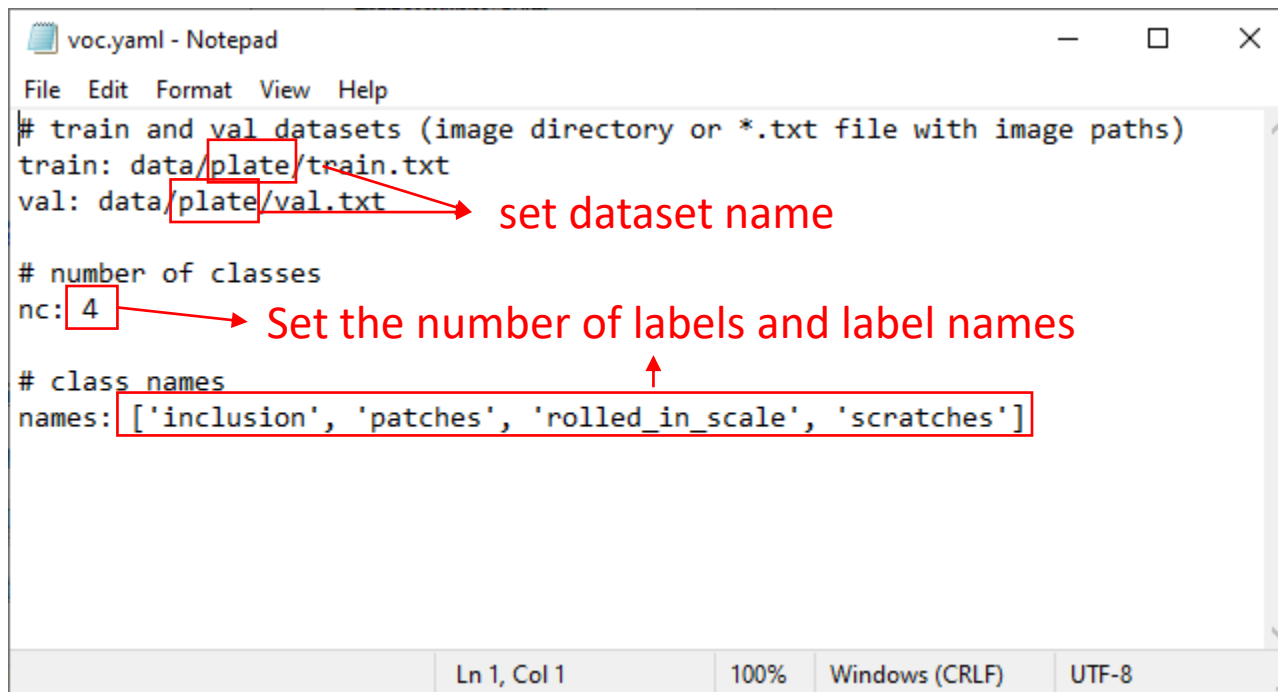
How to use - Prepare files for training

Run 4_delete_log.ipynb

Delete the log files left over from previous training.

How to use - Prepare files for training

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.



The screenshot shows the 'voc.yaml' file in a Notepad window. The file contains the following content:

```
File Edit Format View Help
# train and val datasets (image directory or *.txt file with image paths)
train: data/plate/train.txt
val: data/plate/val.txt

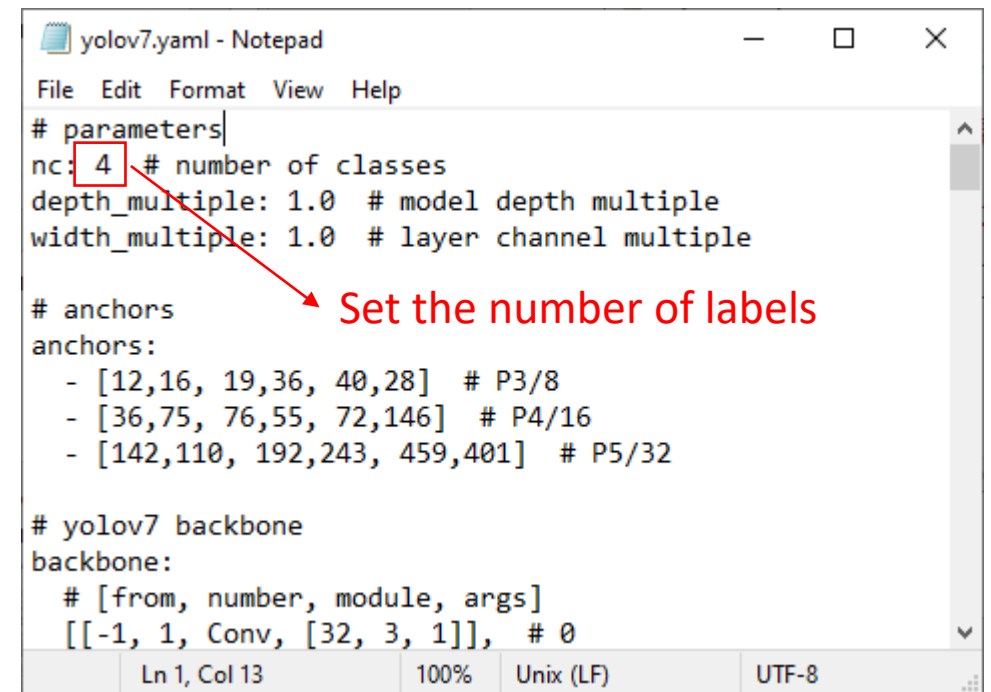
# number of classes
nc: 4

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

Annotations in red:

- A red box around 'plate' in the train and val paths, with an arrow pointing to the text "set dataset name".
- A red box around the number '4' in the 'nc' field, with an arrow pointing to the text "Set the number of labels and label names".
- A red box around the list of class names, with an arrow pointing up to the 'nc' field.

The status bar at the bottom indicates: Ln 1, Col 1 | 100% | Windows (CRLF) | UTF-8.



The screenshot shows the 'yolov7.yaml' file in a Notepad window. The file contains the following content:

```
File Edit Format View Help
# 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
```

Annotations in red:

- A red box around the number '4' in the 'nc' field, with an arrow pointing to the text "Set the number of labels".

The status bar at the bottom indicates: Ln 1, Col 13 | 100% | Unix (LF) | UTF-8.

How to use - Training

Run 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.

```
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\pytorch\pytorch\builder\windows\pytorch\aten\src\ATen\native\TensorShape.cpp:2228.)
00,	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		

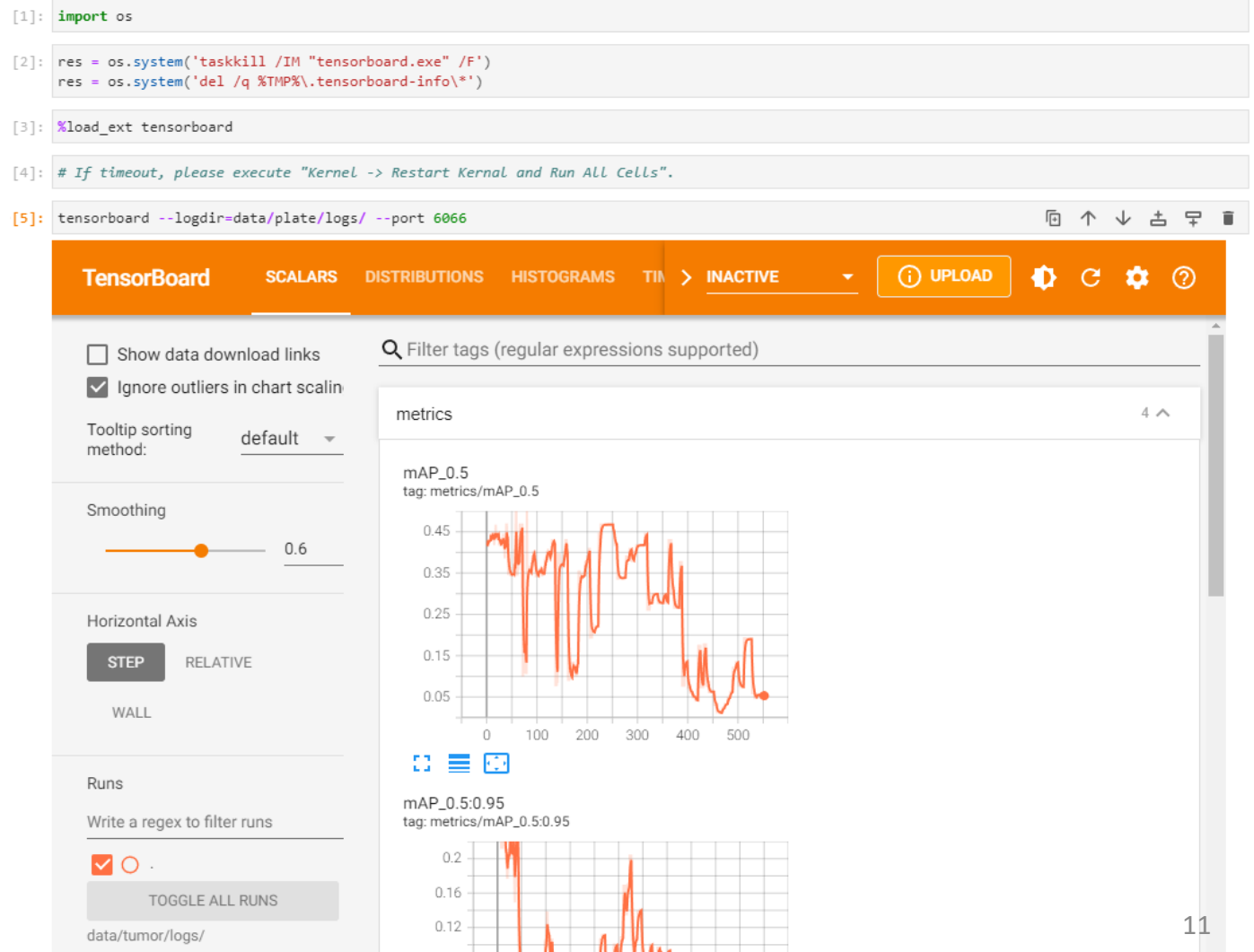
Epoch	gpu_mem	box	obj	cls	total	labels	img_size	
1/2999	11G	0.04368	0.004234	0	0.04792	6	512: 100%	3/3 [00:00<00:00, 3.15it/s]
Class	Images	Labels	P	R	mAP@.5	mAP@.5:.95	100% <th>1/1 [00:00<00:00,</th>	1/1 [00:00<00:00,
00,	Class	Images	Labels	P	R	mAP@.5	mAP@.5:.95: 100%	1/1 [00:00<00:00,
all	3	3	0.997	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, 3.25it/s]
Class	Images	Labels	P	R	mAP@.5	mAP@.5:.95	100% <th>1/1 [00:00<00:00,</th>	1/1 [00:00<00:00,
00,	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		

How to use - Training

Run 6_tensorboard.ipynb

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



How to use - Inference

Run 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

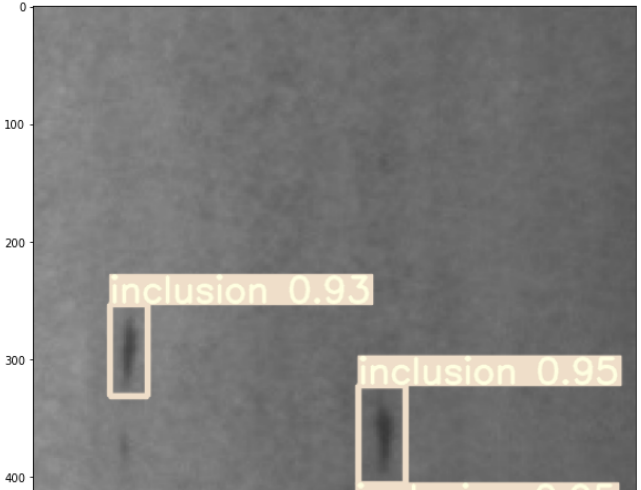
[4]: %run src/detect.py --source $source --img-size $image_size --weights $weights_file --conf $threshold --device $device --view-img --nosave

YOLOv4 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, project='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
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]

0
100
200
300
400
inclusion 0.93
inclusion 0.95
inclusion 0.95
```



How to use - Inference

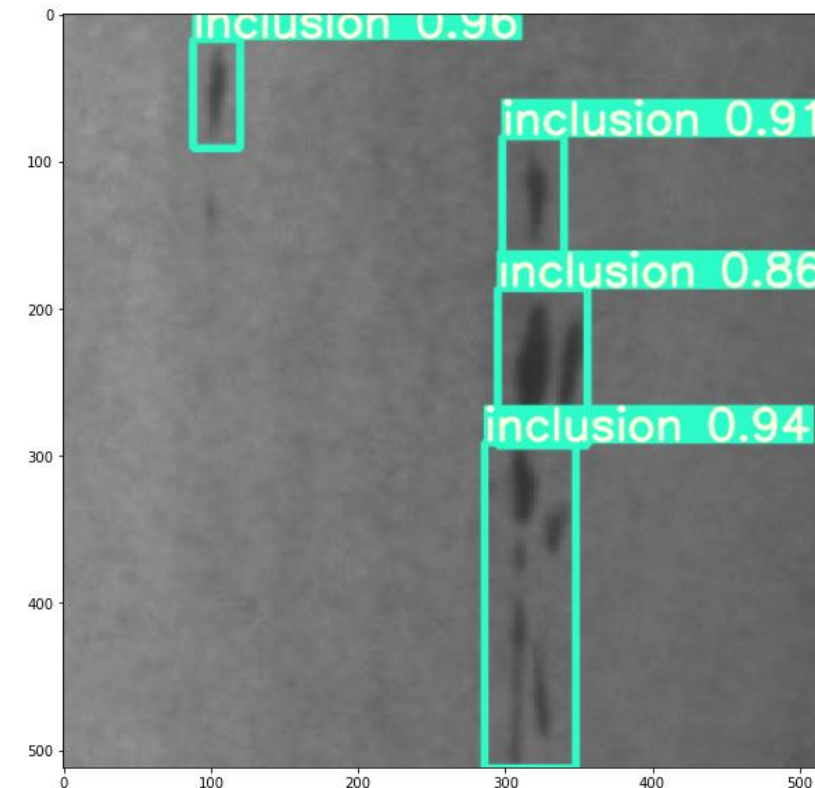
Run 8_inference_image_folder_1.ipynb

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
=====
```



How to use - Inference

Run 9_inference_webcam.ipynb

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