

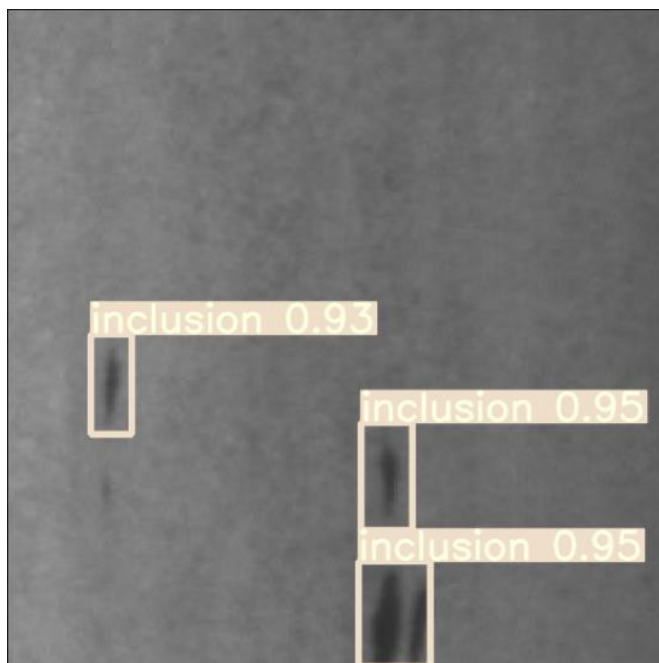
Jupyter-Image-Object-Detection- YOLOv7-PyTorch-GPL

LEADERG INC.

<https://www.leaderg.com>

簡介

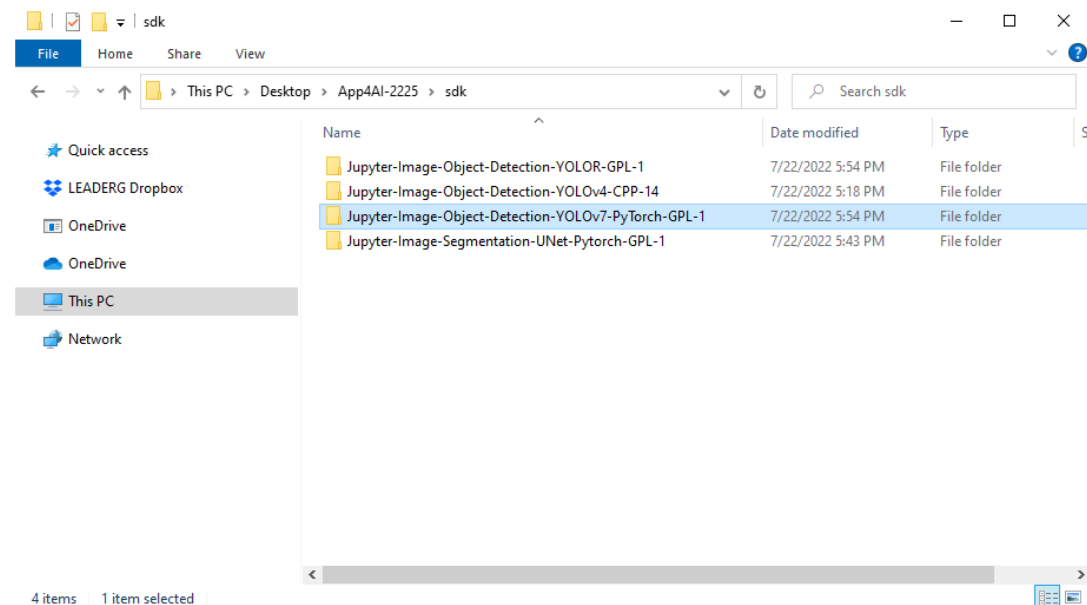
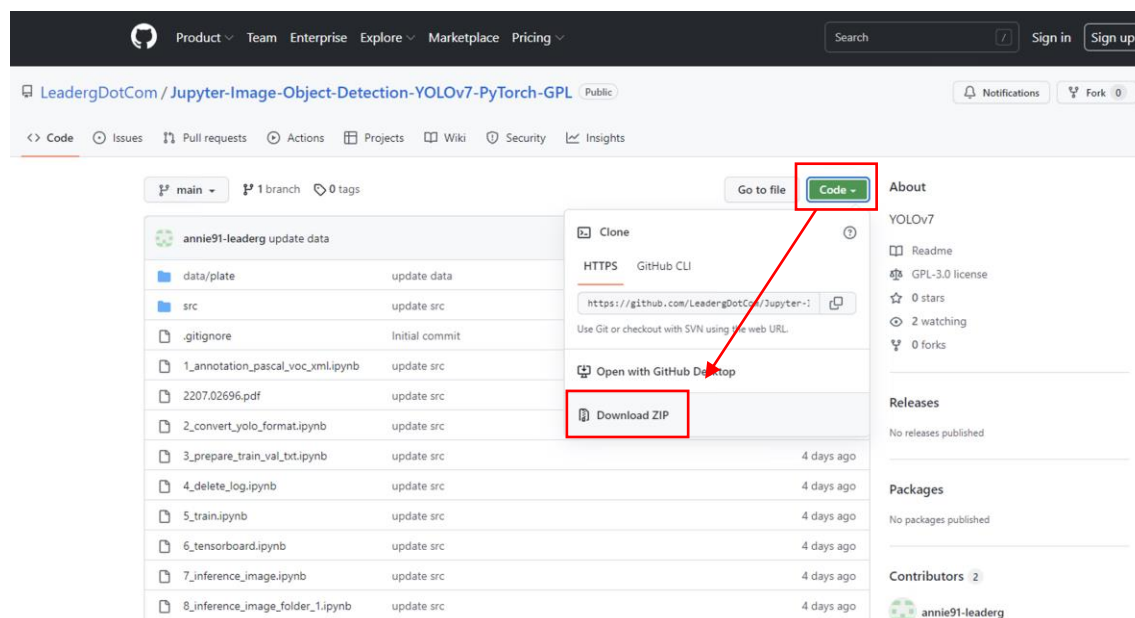
- YOLOv7 演算法可以應用在工廠瑕疵檢測、醫療影像分析、生物影像分析、工安影像分析、口罩影像分析等。



操作步驟及說明-下載 YOLOv7

從 GitHub 下載 zip 後，放至 App4AI/sdk 資料夾後，解壓縮 zip 檔。

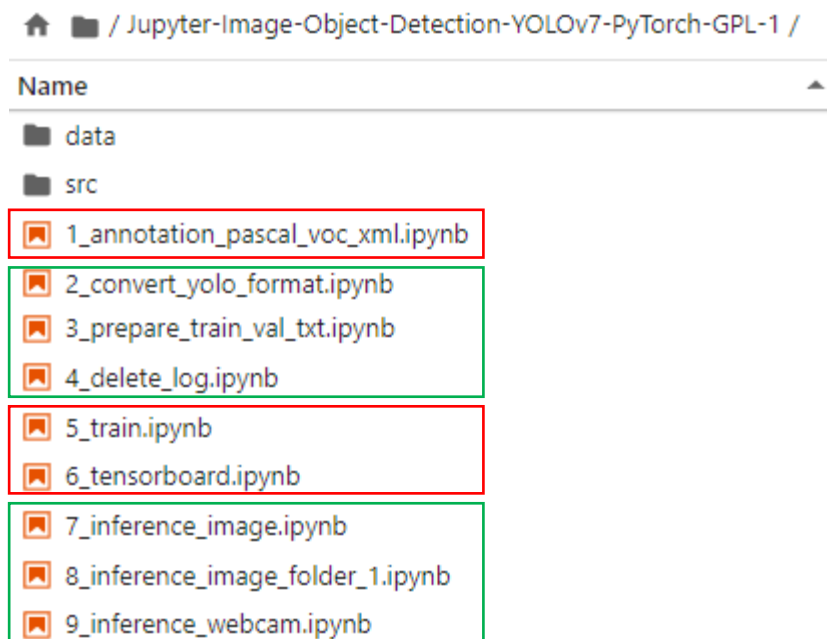
下載網址：<https://github.com/LeadergDotCom/Jupyter-Image-Object-Detection-YOLOv7-PyTorch-GPL>



操作步驟及說明

- 解決方案大致流程為：

標記影像 -> 準備訓練所需檔案 -> 訓練 -> 推論影像



操作步驟及說明-標記影像

運行 1_annotation_pascal_voc_xml.ipynb

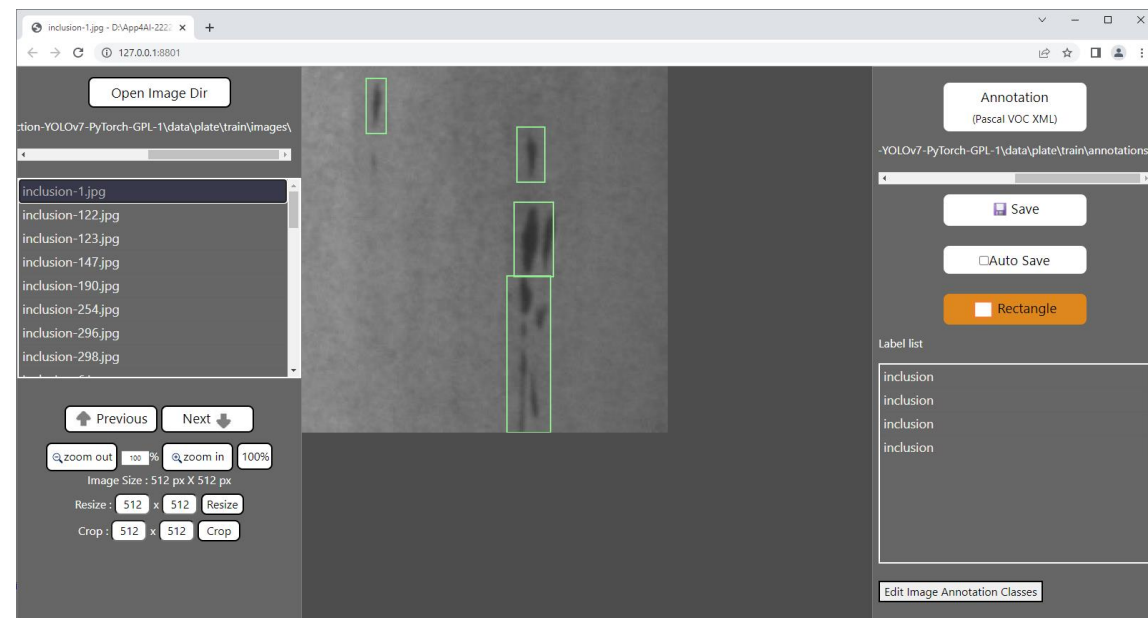
運行後會開啟標記軟體網頁，可進行影像標記。

ipynb 參數說明：

在 #parameter 中：

- port 是標記網頁的埠，預設 8801，若使用者有占用該 port，請自行更改 port 值。
- dataset 是資料集名稱。
- label_folder 是標記 train 資料夾的影像，也可以更改成 val，標記 val 資料夾的影像。

如何使用標註網頁，請參閱 Annotation.pdf。



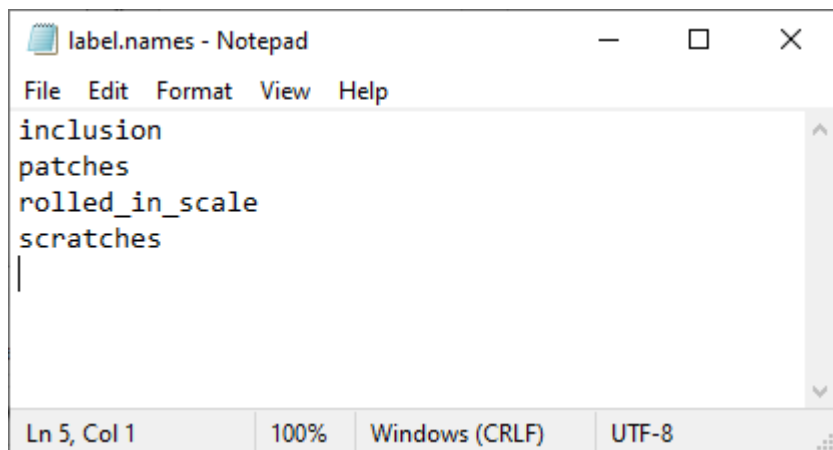
操作步驟及說明-準備訓練所需檔案

運行 2_convert_yolo_format.ipynb

將 voc xml 標記檔轉換成 yolo 的格式，運行前，請先確認 #parameters 內 label_file 路徑下的 label.names，內容所填寫的類別是否正確。

補充：

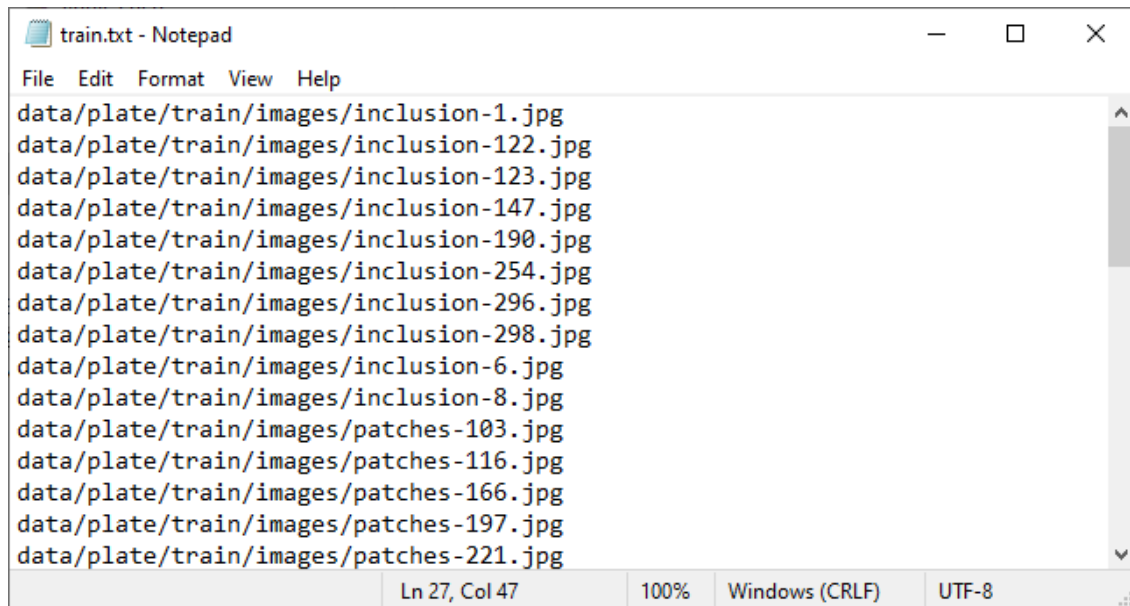
label.names 的內容為類別名稱，不含 background (背景)。



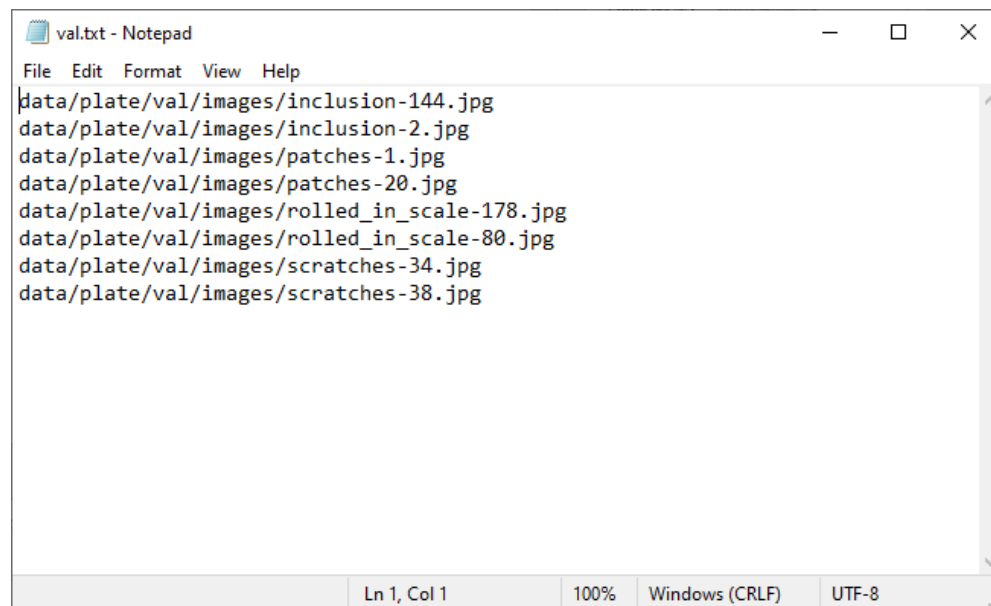
```
label.names - Notepad
File Edit Format View Help
inclusion
patches
rolled_in_scale
scratches
|
Ln 5, Col 1 100% Windows (CRLF) UTF-8
```

操作步驟及說明-準備訓練所需檔案

運行 3_prepare_train_val_txt.ipynb
產生訓練、驗證用的影像路徑檔。



```
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
```

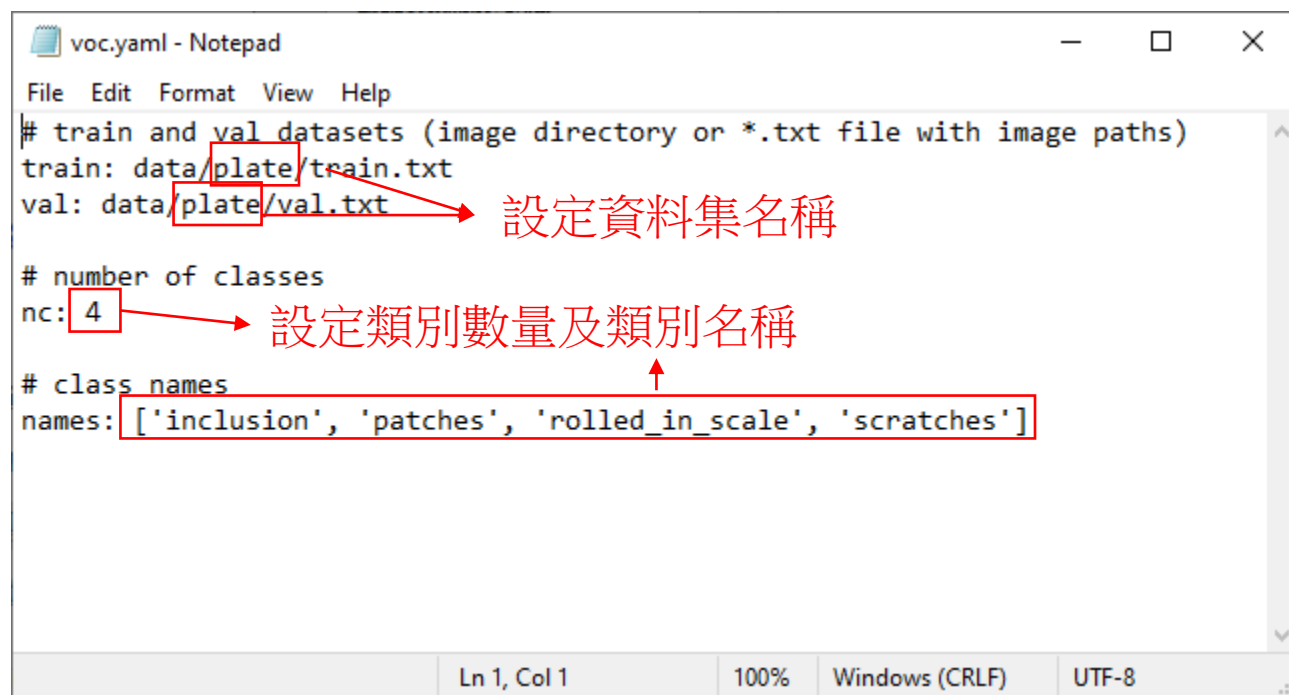
操作步驟及說明-準備訓練所需檔案

運行 4_delete_log.ipynb

刪除之前訓練殘留的 log 檔。

操作步驟及說明-準備訓練所需檔案

設定 dataset 資料集內 yolov7.yaml 與 voc.yaml 檔案內容，設定資料集名稱、類別數量及名稱。



The screenshot shows the 'voc.yaml' file in a Notepad window. The file content is as follows:

```
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

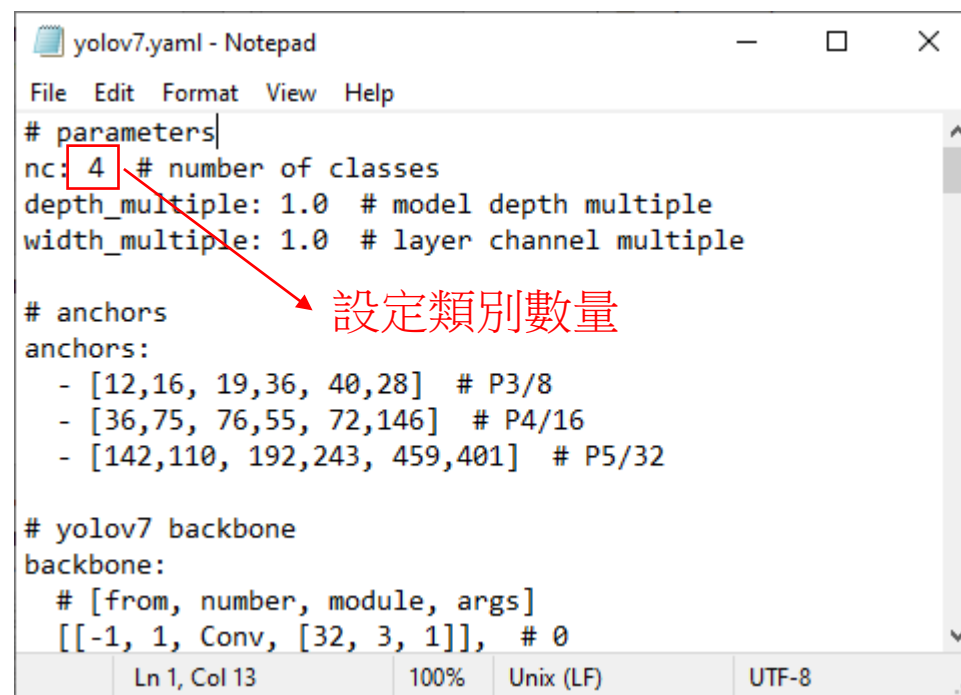
# number of classes
nc: 4

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

Annotations with red arrows and boxes:

- A red box around 'plate' in 'data/plate/train.txt' and 'data/plate/val.txt' has an arrow pointing to the text "設定資料集名稱".
- A red box around the number '4' in 'nc: 4' has an arrow pointing to the text "設定類別數量及類別名稱".
- A red box around the list of class names has an arrow pointing up to the text "設定類別數量及類別名稱".

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 content is as follows:

```
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 with red arrows and boxes:

- A red box around the number '4' in 'nc: 4' has an arrow pointing to the text "設定類別數量".

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

操作步驟及說明-訓練

運行 5_train.ipynb

開始訓練。

ipynb 參數說明：

在 #parameter 中：

- dataset 是資料集名稱。
- weights_file 是使用的 pretrained model 路徑，
None 代表不使用 pretrained model 進行訓練。
- devices 是使用的 GPU id。
- epochs 是訓練輪數。

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,	all	3	3	0.995	0.333	0.418	0.259

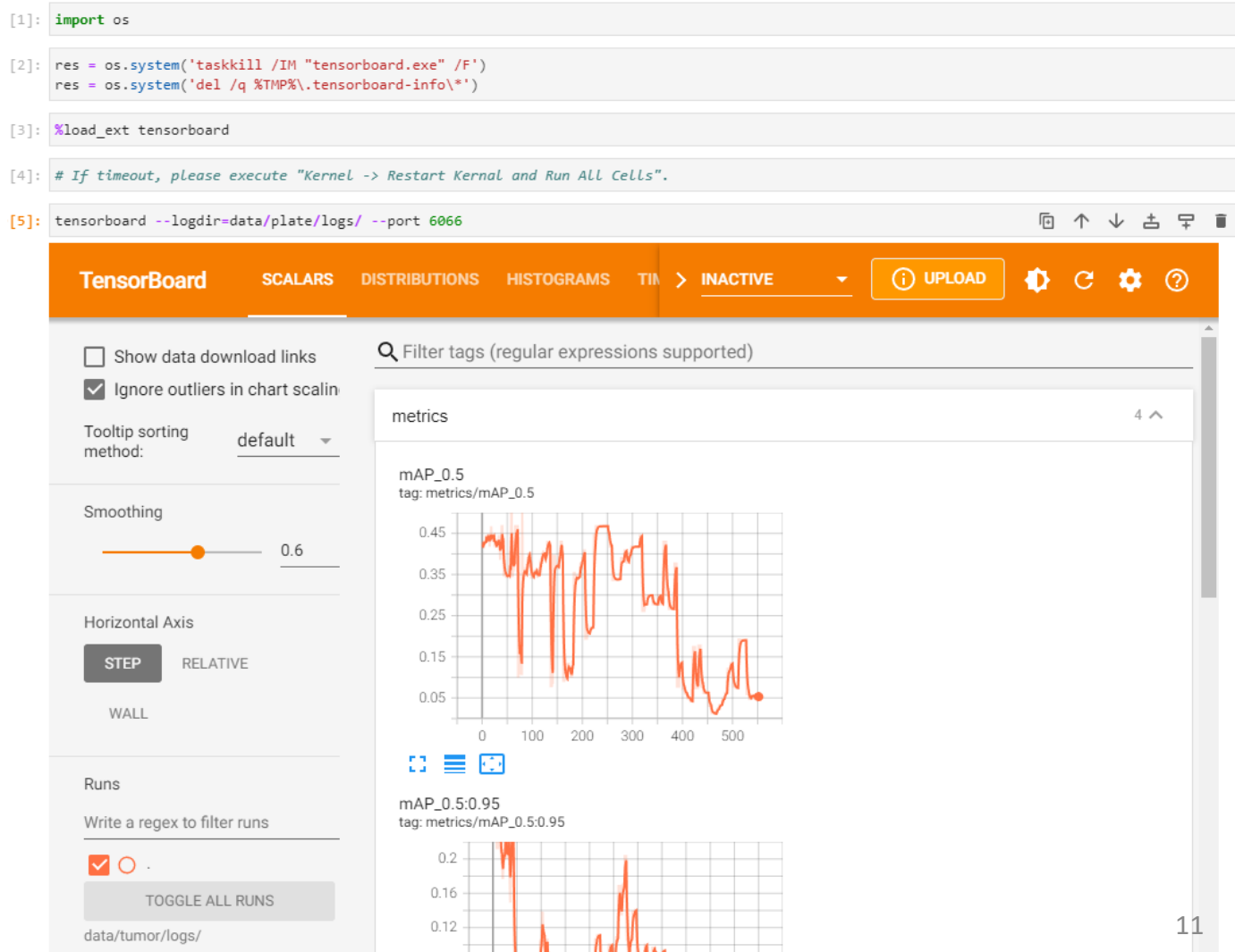
Epoch	gpu_mem	box	obj	cls	total	labels	img_size
1/2999	11G	0.04368	0.004234	0	0.04792	6	512: 100% █ 3/3 [00:00<00:00, 3.15it/s]
Class	Images	Labels	P	R	mAP@.5	mAP@.5:.95:	100% █ 1/1 [00:00<00:00,
00,	all	3	3	0.997	0.333	0.418	0.234

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% █ 1/1 [00:00<00:00,
00,	all	3	3	1	0.333	0.429	0.239

操作步驟及說明-訓練

運行 6_tensorboard.ipynb

可察看訓練的 loss 曲線
及其他相關資訊。



操作步驟及說明-推論影像

運行 7_inference_image.ipynb

推論單張影像。

ipynb 參數說明：

在 #parameter 中：

- dataset 是資料集名稱。
- source 是推論影像路徑。
- weights_file 是推論模型路徑。

```
[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
```



操作步驟及說明-推論影像

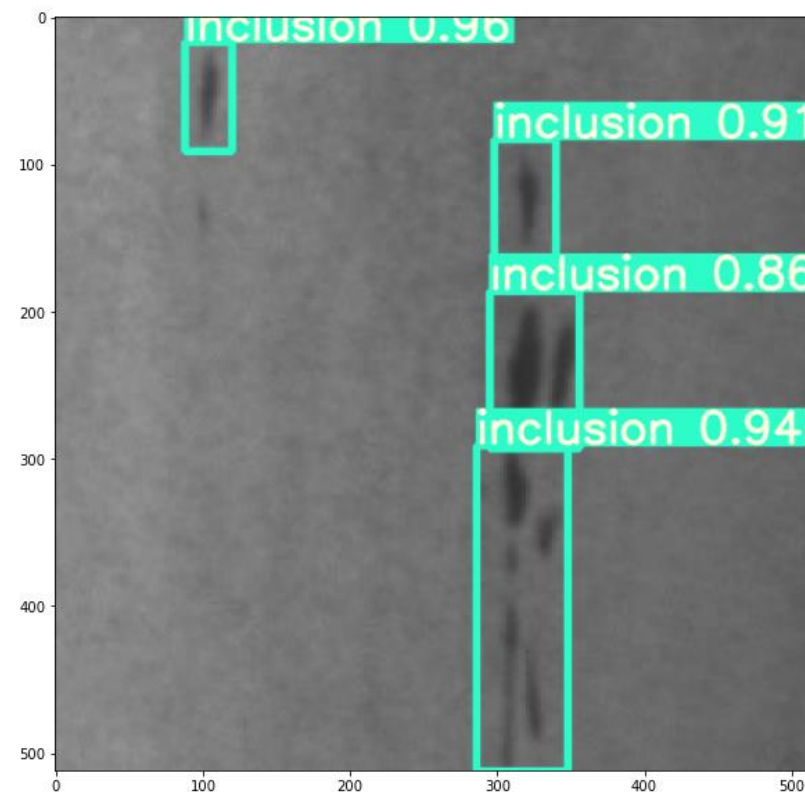
運行 8_inference_image_folder_1.ipynb
推論資料夾內所有影像。

ipynb 參數說明：

在 #parameter 中：

- dataset 是資料集名稱。
- source 是推論影像資料夾路徑。
- weights_file 是推論模型路徑。

```
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
=====
```



操作步驟及說明-推論影像

運行 9_inference_webcam.ipynb

推論 webcam 的影像，對顯示畫面按下“q”即可關閉相機。