

Image-Object-Detection-YOLOv8- PyTorch-GPL-Jupyter

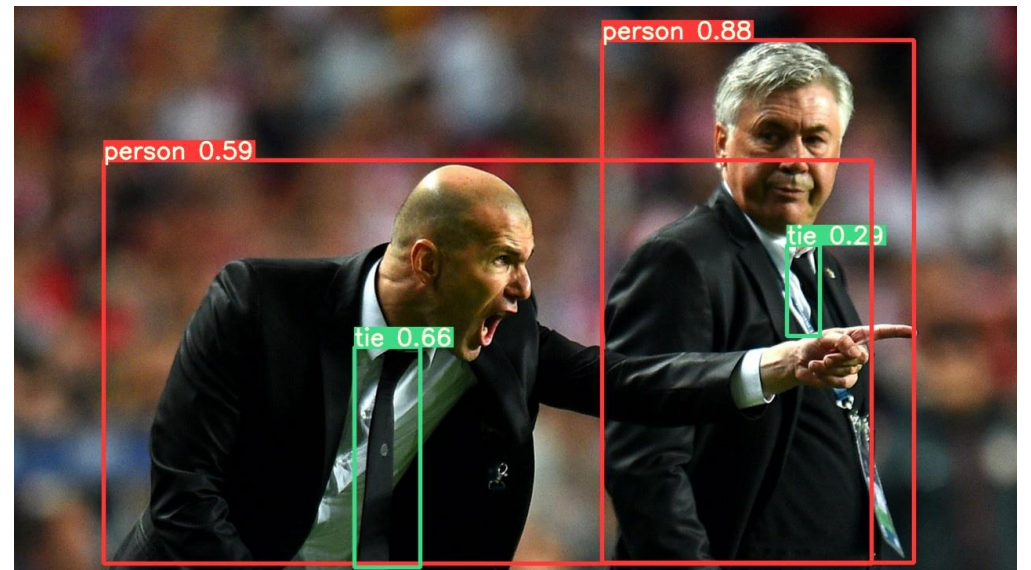
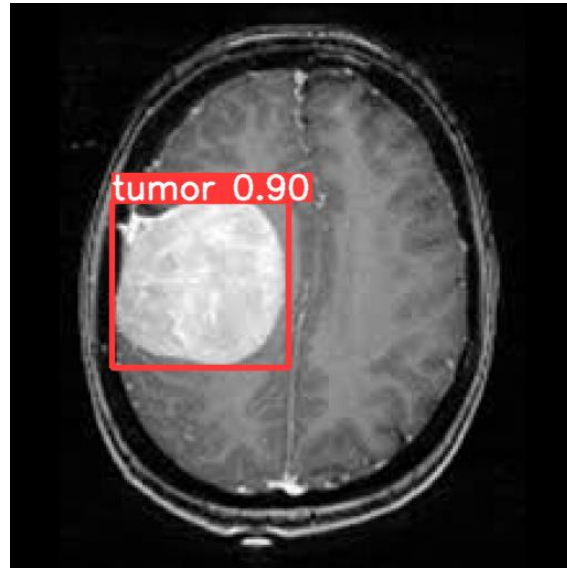
Ultralytics YOLOv8 is a cutting-edge, state-of-the-art (SOTA) model developed by Ultralytics.

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

Version 20230223

Applications

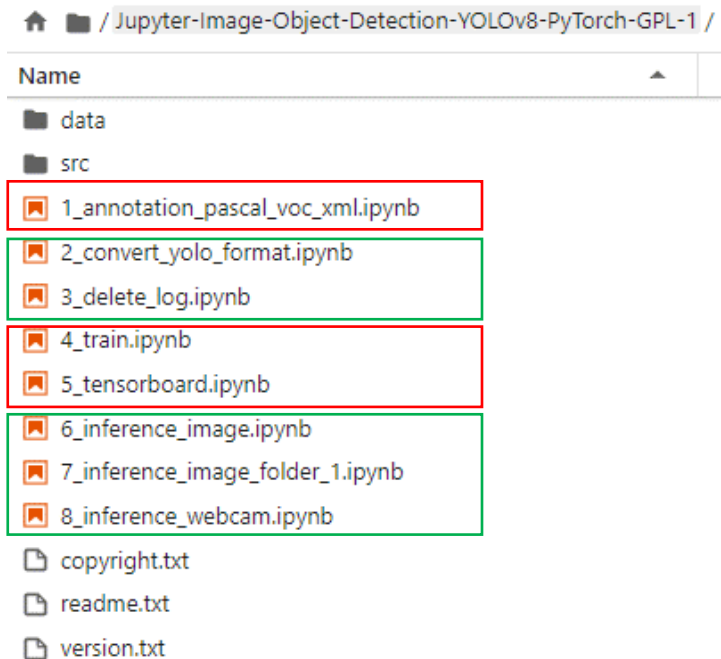
- The YOLOv8 solution can be applied to factory defect detection, medical image analysis, biological image analysis, industrial safety image analysis, mask image analysis, etc.



How to use

The main process is:

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

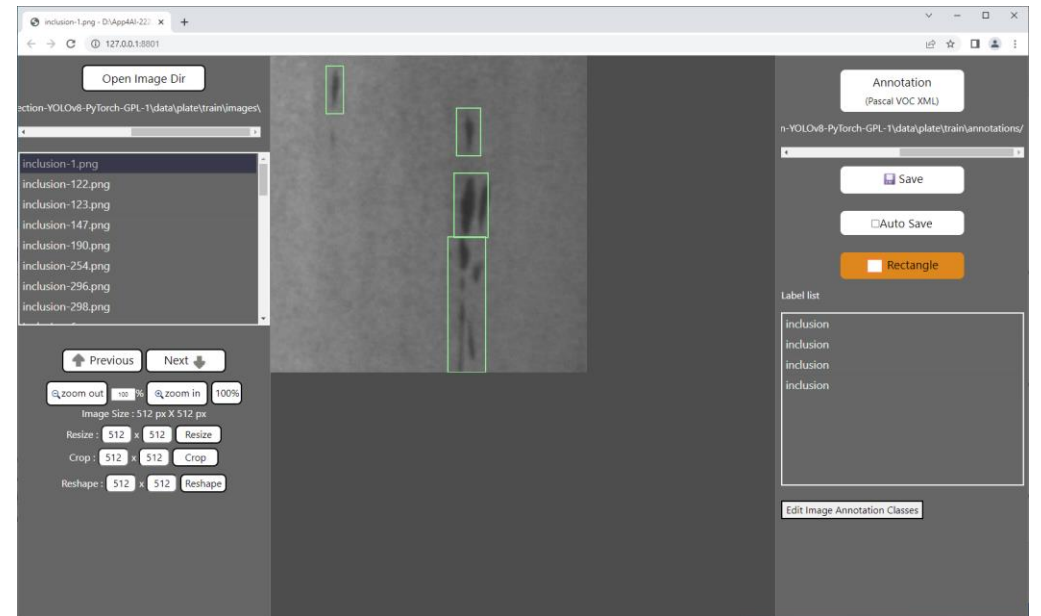


1_annotation_pascal_voc_xml.ipynb

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.

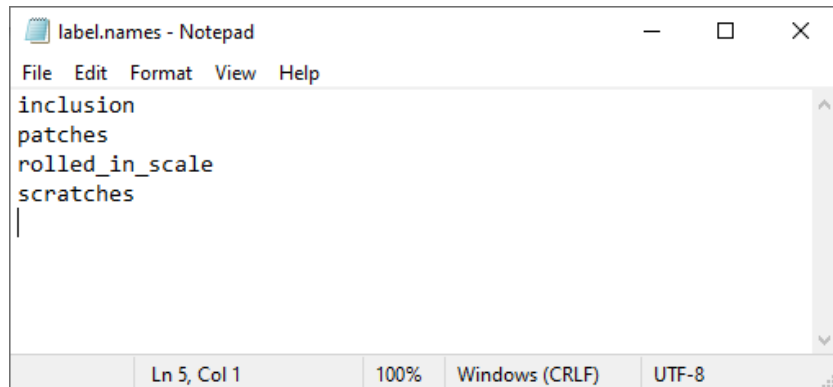
2_convert_yolo_format.ipynb

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.

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



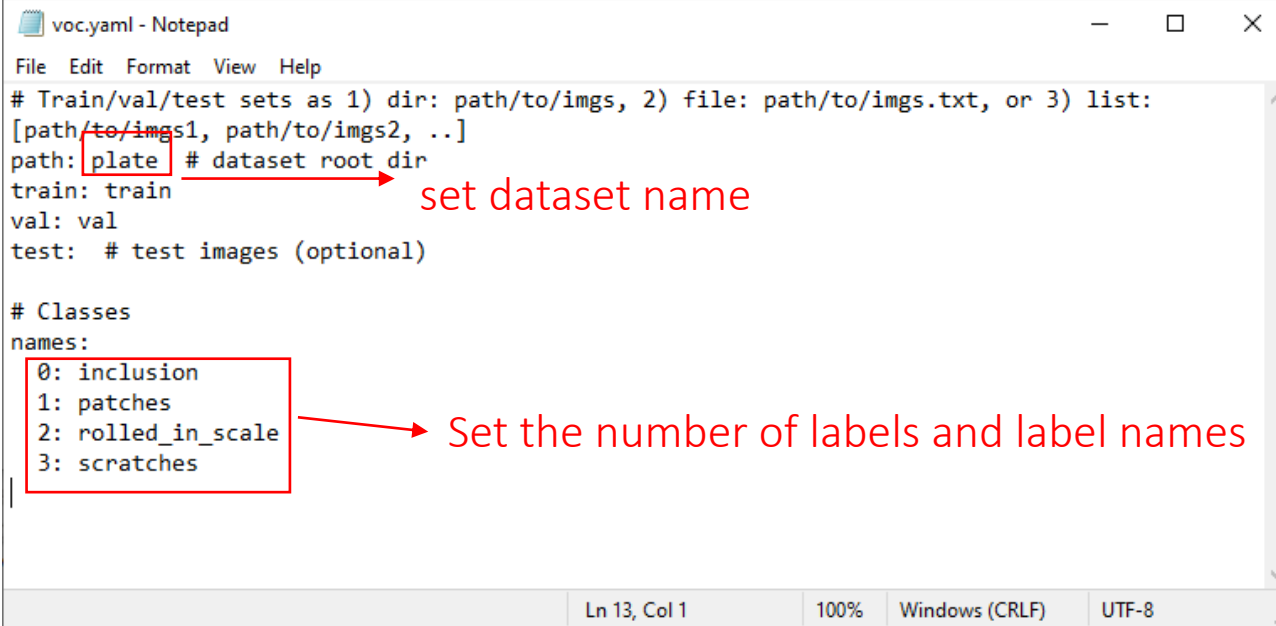
```
label.names - Notepad
File Edit Format View Help
inclusion
patches
rolled_in_scale
scratches
|
Ln 5, Col 1 100% Windows (CRLF) UTF-8
```

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: plate # dataset root dir
train: train
val: val
test: # test images (optional)

# Classes
names:
0: inclusion
1: patches
2: rolled_in_scale
3: scratches
```

The screenshot shows a Notepad window titled "voc.yaml - Notepad". The text content is as follows:

```
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: plate # dataset root dir
train: train
val: val
test: # test images (optional)

# Classes
names:
0: inclusion
1: patches
2: rolled_in_scale
3: scratches
```

Two red annotations are present:

- An arrow points from the text "set dataset name" to the word "plate" in the "path:" line.
- An arrow points from the text "Set the number of labels and label names" to the list of labels under the "names:" section.

The status bar at the bottom indicates "Ln 13, Col 1", "100%", "Windows (CRLF)", and "UTF-8".

4_train.ipynb

Start training.

ipynb parameter:

- dataset is the dataset name.
- weights_file is the pretrained model path used.
- devices is the GPU id used.
- epochs is the number of training epochs.

```
run_command()

torchvision.io.image.py:13: UserWarning: Failed to load image Python extension:
torch\_jit\_internal.py:751: UserWarning: Unable to retrieve source for @torch.jit\_overload function: <function\_DenseLayer.forward
at 0x000002B334082790>.
warnings.warn(f"Unable to retrieve source for @torch.jit\_overload function: {func}.")
torch\_jit\_internal.py:751: UserWarning: Unable to retrieve source for @torch.jit\_overload function: <function\_DenseLayer.forward
at 0x000002B334096880>.
warnings.warn(f"Unable to retrieve source for @torch.jit\_overload function: {func}.")
ultralytics YOLOv8.0.6 Python-3.9.12 torch-1.12.0+cu113 CUDA:0 (NVIDIA TITAN RTX, 24576MiB)
yoloEngineTrainer: task=detect, mode=train, model=data/tumor/model/yolov8x.pt, data=data/tumor/voc.yaml, epochs=1000, patience=5
0, batch=16, imgsz=512, save=True, cache=False, device=0, workers=4, project=data/tumor, name=model, exist_ok=True, pretrained=True,
optimizer=SGD, verbose=False, seed=0, deterministic=True, single_cls=False, image_weights=False, rect=False, cos_lr=False, close
_mosaic=10, resume=False, overlap_mask=True, mask_ratio=4, dropout=0.0, val=True, save_json=False, save_hybrid=False, conf=None, io
u=0.7, max_det=300, half=False, dnn=False, plots=True, source=None, show=False, save_txt=False, save_conf=False, save_crop=False, h
ide_labels=False, hide_conf=False, vid_stride=1, line_thickness=3, visualize=False, augment=False, agnostic_nms=False, retina_masks
=False, format=torchscript, keras=False, optimize=False, int8=False, dynamic=False, simplify=False, opset=17, workspace=4, nms=Fast
e, lr=0.01, lrf=0.01, momentum=0.937, weight_decay=0.0005, warmup_epochs=3.0, warmup_momentum=0.8, warmup_bias_lr=0.1, box=7.5, cl
s=0.5, dfl=1.5, fl_gamma=0.0, label_smoothing=0.0, nbs=64, hsv_h=0.015, hsv_s=0.7, hsv_v=0.4, degrees=0.0, translate=0.1, scale=0.
5, shear=0.0, perspective=0.0, flipud=0.0, flipud=0.0, flipud=0.0, flipud=0.0, mixup=0.0, copy_paste=0.0, cfg=None, hydra={'output_subdir': Non
e, 'run': {'dir': '.'}}, v5loader=False, show_rate=False, save_dir=data/tumor/model
Overriding model.yaml nc=80 with nc=1

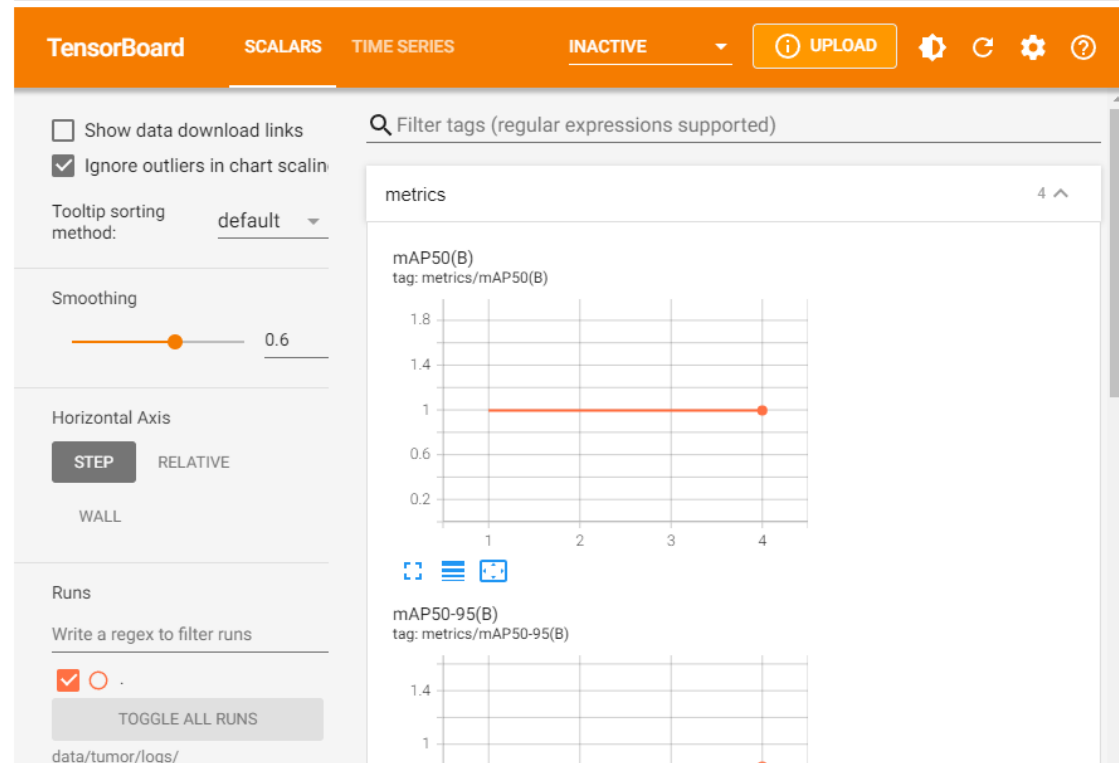
from n  params module                                arguments
0      -1  1      2320 ultralytics.nn.modules.Conv                [3, 80, 3, 2]
1      -1  1     115520 ultralytics.nn.modules.Conv                [80, 160, 3, 2]
2      -1  3     436800 ultralytics.nn.modules.C2f                [160, 160, 3, True]
3      -1  1     461440 ultralytics.nn.modules.Conv                [160, 320, 3, 2]
4      -1  6     3281920 ultralytics.nn.modules.C2f                [320, 320, 6, True]
5      -1  1     1844480 ultralytics.nn.modules.Conv                [320, 640, 3, 2]
6      -1  6     13117440 ultralytics.nn.modules.C2f                [640, 640, 6, True]
7      -1  1     3687680 ultralytics.nn.modules.Conv                [640, 640, 3, 2]
8      -1  3     6969600 ultralytics.nn.modules.C2f                [640, 640, 3, True]
9      -1  1     1025920 ultralytics.nn.modules.SPPF                [640, 640, 5]
10     -1  1          0 torch.nn.modules.upsampling.Upsample            [None, 2, 'nearest']
11     [-1, 6] 1          0 ultralytics.nn.modules.Concat                [1]
12     -1  3     7379200 ultralytics.nn.modules.C2f                [1280, 640, 3]
13     -1  1          0 torch.nn.modules.upsampling.Upsample            [None, 2, 'nearest']
14     [-1, 4] 1          0 ultralytics.nn.modules.Concat                [1]
15     -1  3     1948800 ultralytics.nn.modules.C2f                [960, 320, 3]
16     -1  1     922240 ultralytics.nn.modules.Conv                [320, 320, 3, 2]
17     [-1, 12] 1          0 ultralytics.nn.modules.Concat                [1]
18     -1  3     7174400 ultralytics.nn.modules.C2f                [960, 640, 3]
19     -1  1     3687680 ultralytics.nn.modules.Conv                [640, 640, 3, 2]
20     [-1, 9] 1          0 ultralytics.nn.modules.Concat                [1]
21     -1  3     7379200 ultralytics.nn.modules.C2f                [1280, 640, 3]
22     [15, 18, 21] 1     8718931 ultralytics.nn.modules.Detect                [1, [320, 640, 640]]

Model summary: 365 layers, 68153571 parameters, 68153555 gradients, 258.1 GFLOPs
```


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/tumor/logs/ --port 6066
```



6_inference_start.ipynb

Start the inference server.

ipynb parameter:

- dataset is the dataset name.
- 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]: import subprocess

[3]: dataset = "tumor"

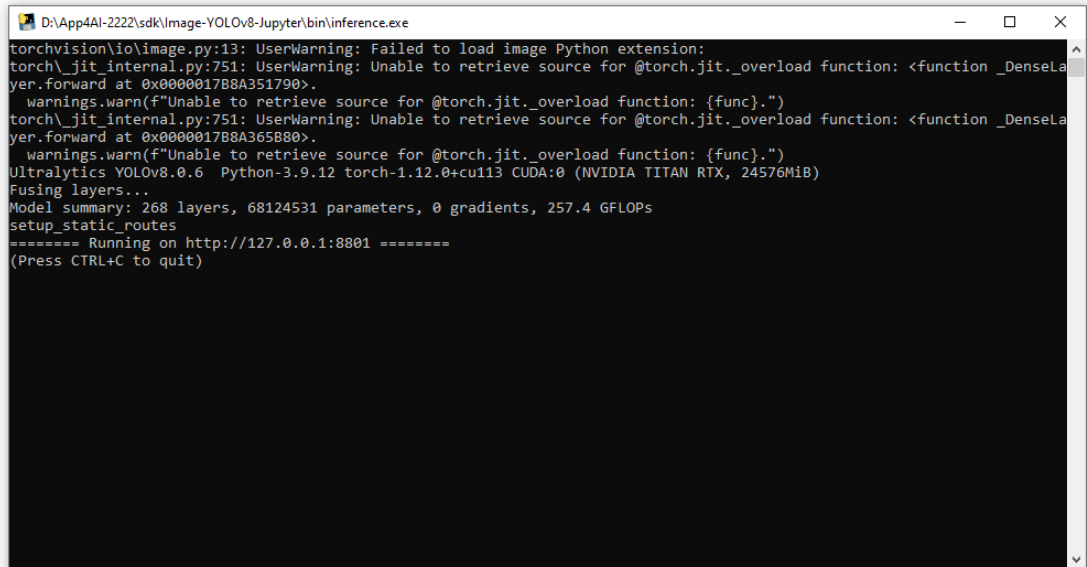
source = "data/%s/test/images/tumor-1.png" %(dataset)
image_size = 512

weights_file = "data/%s/model/best.pt" %(dataset)
device = "0" # 0, 1, 2, ... for Nvidia GPU or cpu for CPU
threshold = '0.5'

[4]: cmd = "bin/inference.exe --port 8801 --model_file " + weights_file + " --threshold " + threshold + " --img_size " + str(image_size)

[*]: subprocess.run(cmd, creationflags = subprocess.CREATE_NEW_CONSOLE)

[ ]:
```



```
D:\App4AI-2222\sdk\Image-YOLOv8-Jupyter\bin\inference.exe
torchvision\io\image.py:13: UserWarning: Failed to load image Python extension:
torch\_jit\_internal.py:751: UserWarning: Unable to retrieve source for @torch.jit._overload function: <function _DenseLayer.forward at 0x0000017B8A351790>.
  warnings.warn(f"Unable to retrieve source for @torch.jit._overload function: {func}.")
torch\_jit\_internal.py:751: UserWarning: Unable to retrieve source for @torch.jit._overload function: <function _DenseLayer.forward at 0x0000017B8A365B80>.
  warnings.warn(f"Unable to retrieve source for @torch.jit._overload function: {func}.")
Ultralytics YOLOv8.0.6 Python-3.9.12 torch-1.12.0+cu113 CUDA:0 (NVIDIA TITAN RTX, 24576MiB)
Fusing layers...
Model summary: 268 layers, 68124531 parameters, 0 gradients, 257.4 GFLOPs
setup_static_routes
===== Running on http://127.0.0.1:8801 =====
(Press CTRL+C to quit)
```

7_inference.ipynb

Send the image to the server for inference through curl and draw the result image after receiving the returned inference result.

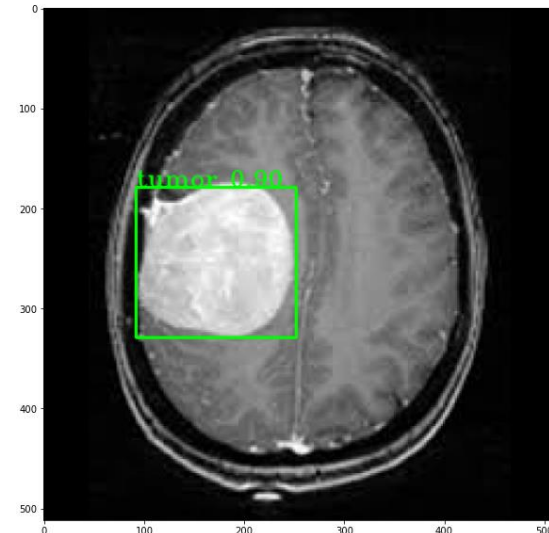
ipynb parameter:

- source is the inferred image path.
- port is the server port.

```
if os.path.exists(source):
    img = cv2.imread(source)

    for res in result_json:
        if 'object' in res:
            for obj in res['object']:
                cv2.rectangle(img, (int(obj['x']), int(obj['y']), int(obj['width']), int(obj['height'])), (0, 255, 0), 2)
                text = obj['type'] + " %.2f" % (float(obj['score']))
                cv2.putText(img, text, (int(obj['x']), int(obj['y']) - 1), cv2.FONT_HERSHEY_TRIPLEX, 0.75, (0, 255, 0), 1, cv2.LINE_AA)

    image = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
    plt.imshow(image)
    plt.show()
```



8_inference_stop.ipynb

Shut down the inference server.

```
[1]: # Copyright © 2023 LEADERG Inc. All rights reserved. Please keep it private. Publish to internet is not allowed.
```

```
[2]: import pycurl
```

```
[3]: url = "http://127.0.0.1:8801/api/stop"
try:
    timeout = 30

    curl = pycurl.Curl()
    curl.setopt(pycurl.HTTPHEADER, ['Expect:', 'Keep-Alive: 300', 'Connection: Keep-Alive'])
    curl.setopt(pycurl.VERBOSE, 1)
    curl.setopt(pycurl.URL, url)
    curl.setopt(pycurl.TIMEOUT, timeout)
    curl.setopt(pycurl.CONNECTTIMEOUT, timeout)

    curl.setopt(pycurl.USERAGENT, "Mozilla/5.0")
    curl.perform()
    curl.close()
except Exception as e:
    print(e)
```

```
(52, 'Empty reply from server')
```

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.