

Image-Classification-YOLOv8- PyTorch-GPL-Jupyter

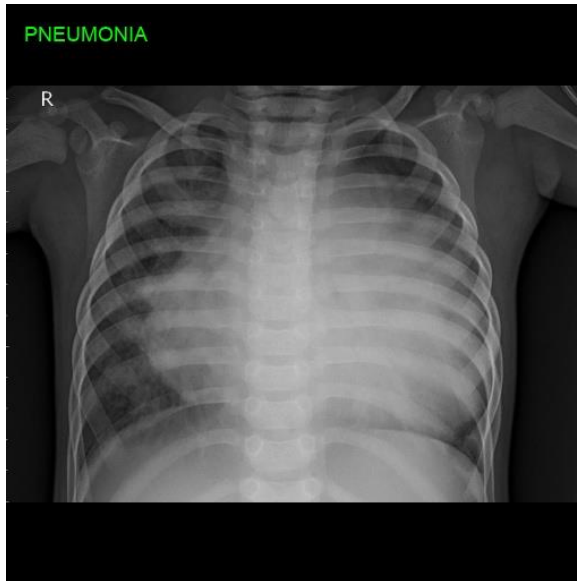
Ultralytics YOLOv8 is a cutting-edge, state-of-the-art (SOTA) model developed by Ultralytics.

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 classification, medical image classification, biological image classification, mask image classification, etc.



How to use

The main process is:

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

Name

data

src

1_create_ImageNet_yaml.ipynb

2_train.ipynb

3_tensorboard.ipynb

4_inference_image.ipynb

5_inference_image_folder_1.ipynb

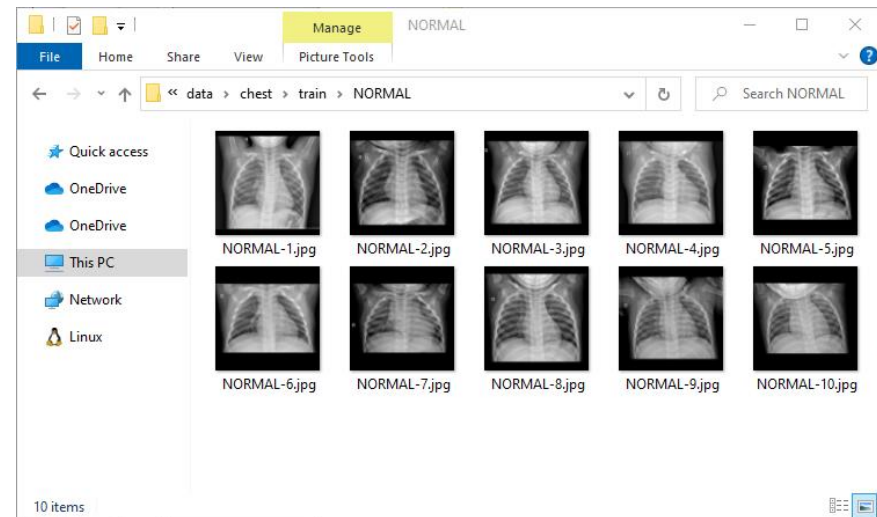
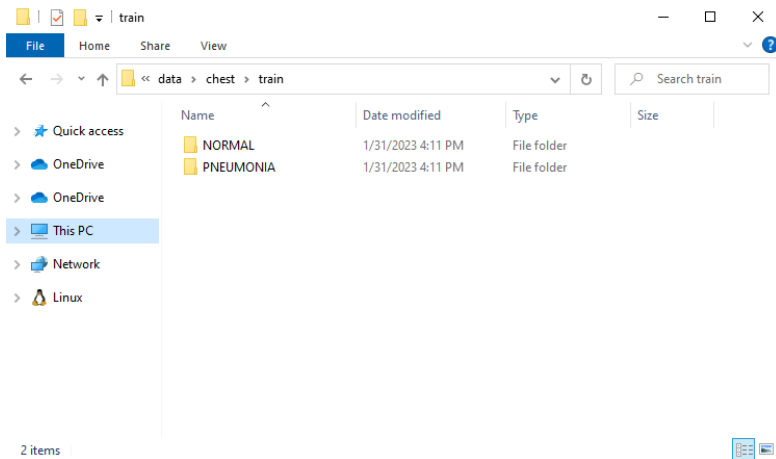
copyright.txt

readme.txt

version.txt

Prepare images

- Put the classified pictures into the train and val folders, create the category folder and put the images.
- Rename the image with the category name, like "image_category_name-XXX.png".

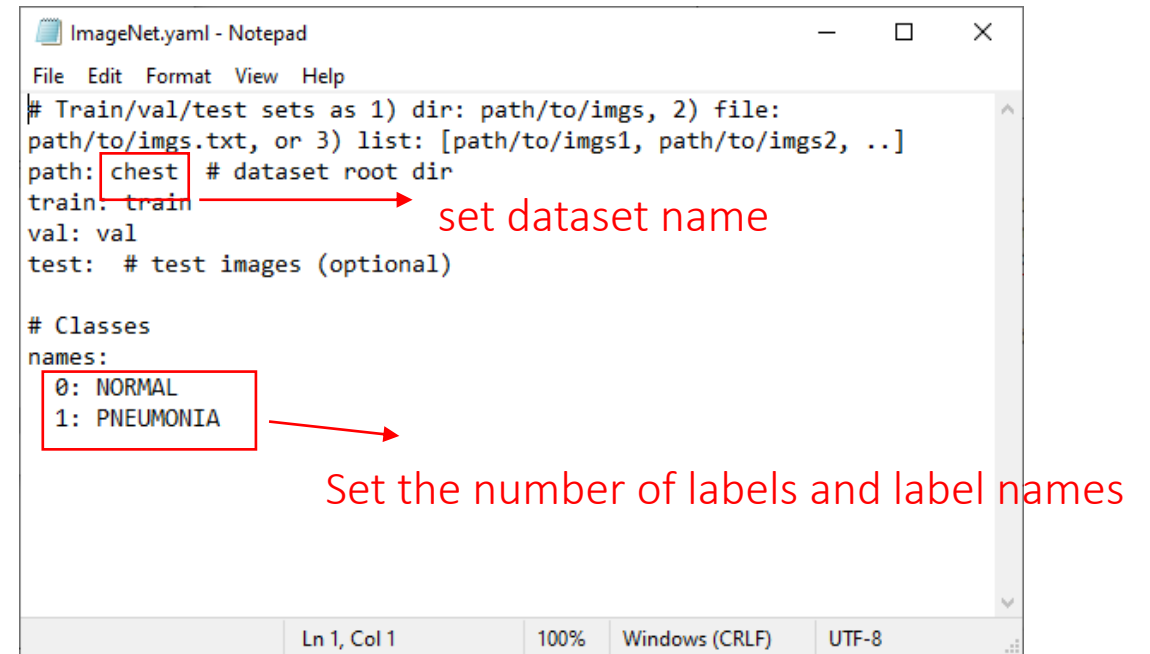


1_create_ImageNet_yaml

Change ImageNet.yaml file parameters after running.

supplement:

Confirm the content of the ImageNet.yaml file in the dataset, such as the name of the dataset, the number of categories, and the name.



```
ImageNet.yaml - Notepad
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: chest # dataset root dir
train: train
val: val
test: # test images (optional)

# Classes
names:
0: NORMAL
1: PNEUMONIA
```

set dataset name

Set the number of labels and label names

Ln 1, Col 1 100% Windows (CRLF) UTF-8

2_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 src/train.py model=$weights_file data=$yaml_file imgsz=$image_size device=$device epochs=$epochs workers=$workers batch=$batch_size
```

```
Ultraalytics YOLOv8.0.6 Python-3.9.12 torch-1.12.0+cu113 CUDA:0 (NVIDIA TITAN RTX, 24576MiB)
yolo\engine\trainer: task=classify, mode=train, model=data/chest/model/yolov8x-cls.pt, data=data/chest/ImageNet.yaml, epochs=300, patience=50, batch=16, imgsz=512, save=True, cache=False, device=0, workers=4, project=data/chest, 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, iou=0.7, max_det=300, half=False, dnn=False, plots=True, source=None, show=False, save_txt=False, save_conf=False, save_crop=False, hide_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=False, lr=0.01, lr_f=0.01, momentum=0.937, weight_decay=5e-05, warmup_epochs=0.0, warmup_momentum=0.8, warmup_bias_lr=0.1, box=7.5, cls=0.5, dfl=1.5, fl_gamma=0.0, label_smoothing=0.1, 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=0.5, mosaic=1.0, mixup=0.0, copy_paste=0.0, cfg=None, hydra={'output_subdir': None, 'run': {'dir': '.'}}, v5loader=False, show_rate=False, save_dir=data\chest\model
Overriding model.yaml nc=1000 with nc=2
```

	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	7375360	ultralytics.nn.modules.Conv	[640, 1280, 3, 2]
8	-1	3	27865600	ultralytics.nn.modules.C2f	[1280, 1280, 3, True]
9	-1	1	1643522	ultralytics.nn.modules.Classify	[1280, 2]

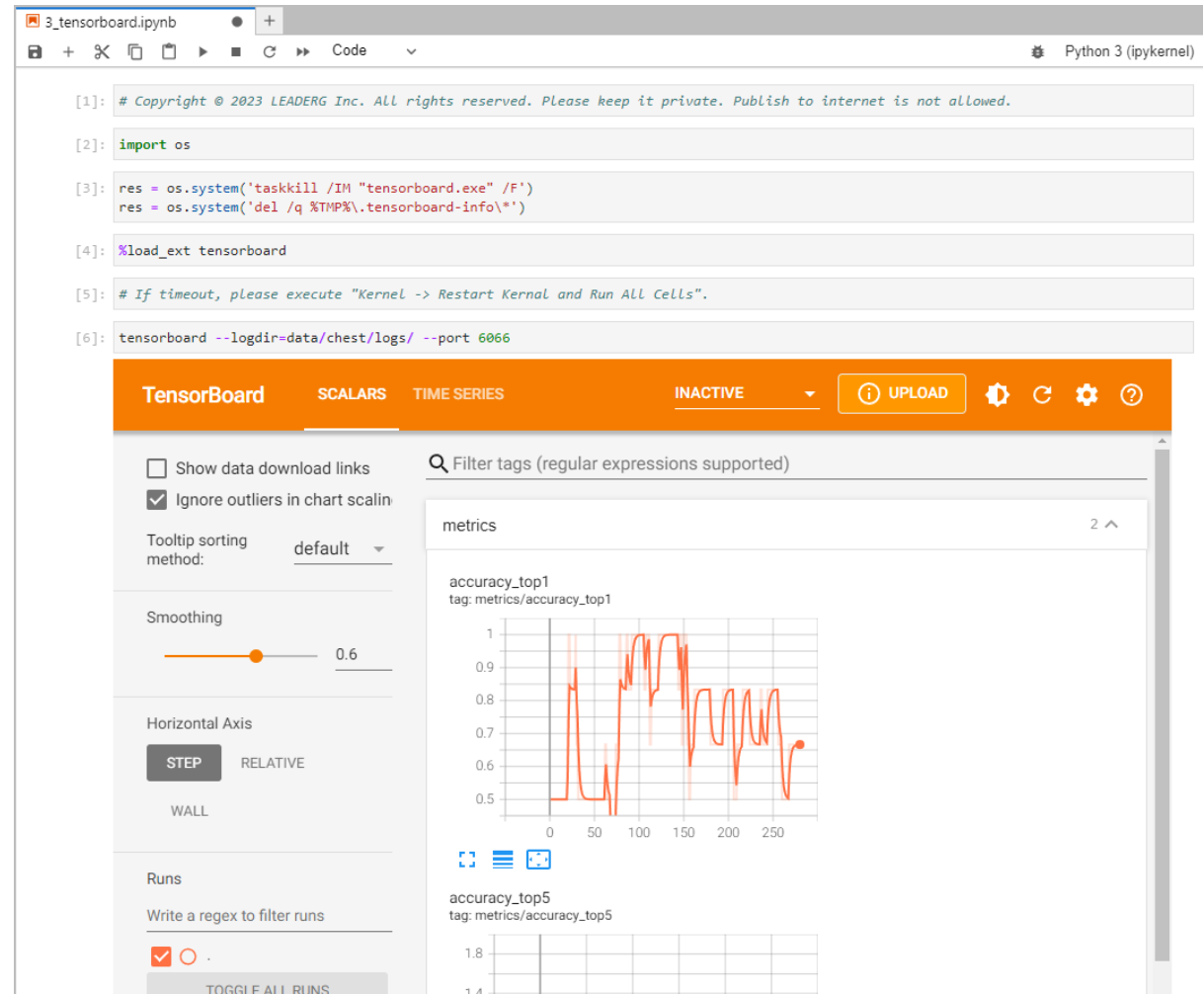
```
YOLOv8x-cls summary: 183 layers, 56144402 parameters, 56144402 gradients, 154.3 GFLOPs
optimizer: SGD(lr=0.1) with parameter groups 50 weight(decay=0.0), 51 weight(decay=5e-05), 51 bias
Image sizes 512 train, 512 val
Using 4 dataloader workers
Logging results to data\chest\model
Starting training for 300 epochs...
```

Epoch	GPU_mem	loss	Instances	Size	
1/300	6.09G	0.1105	4	512: 100%	██████████ 2/2 [00:03<00:00, 1.90s/it]
classes	top1_acc	top5_acc	top5_acc: 100%	██████████	1/1 [00:00<00:00, 30.30it/s]
all		0.5	1		

Epoch	GPU_mem	loss	Instances	Size	
2/300	6.1G	0.1098	4	512: 100%	██████████ 2/2 [00:00<00:00, 5.05it/s]
classes	top1_acc	top5_acc	top5_acc: 100%	██████████	1/1 [00:00<00:00, 27.03it/s]
all		0.5	1		

3_tensorboard.ipynb

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



The screenshot displays a Jupyter Notebook environment with the following code cells:

```
[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/chest/logs/ --port 6066
```

Below the code, the TensorBoard interface is shown. The top navigation bar includes 'TensorBoard', 'SCALARS', 'TIME SERIES', 'INACTIVE', and an 'UPLOAD' button. The left sidebar contains settings: 'Show data download links' (unchecked), 'Ignore outliers in chart scaling' (checked), 'Tooltip sorting method: default', 'Smoothing' slider at 0.6, 'Horizontal Axis' buttons for 'STEP', 'RELATIVE', and 'WALL', and a 'Runs' section with a filter input and a 'TOGGLE ALL RUNS' button.

The main content area displays a line chart for 'accuracy_top1' with tag 'metrics/accuracy_top1'. The y-axis ranges from 0.5 to 1.0, and the x-axis ranges from 0 to 250. The chart shows a fluctuating orange line that generally trends upwards from approximately 0.5 to 0.7. Below it, a table for 'accuracy_top5' with tag 'metrics/accuracy_top5' is visible, showing a y-axis from 1.4 to 1.8.

4_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.

```
dataset = "chest"
dataset_path = "data/%s" %(dataset)
source = "data/%s/test/NORMAL-11.jpg" %(dataset)
image_size = 512

weights_file = "data/%s/model/best.pt" %(dataset)
#weights_file = "yoloV8x-cls.pt"
device = "0" # 0, 1, 2, ... for Nvidia GPU or cpu for CPU

%run src/predict.py task=classify model=$weights_file source=$source imgs=$image_size show=True device=$device project=$dataset_path r
```

←

Ultralytics YOLOv8.0.6 Python-3.9.12 torch-1.12.0+cu113 CUDA:0 (NVIDIA TITAN RTX, 24576MiB)
Fusing layers...
YOLOv8x-cls summary: 133 layers, 56125762 parameters, 0 gradients, 153.8 GFLOPs

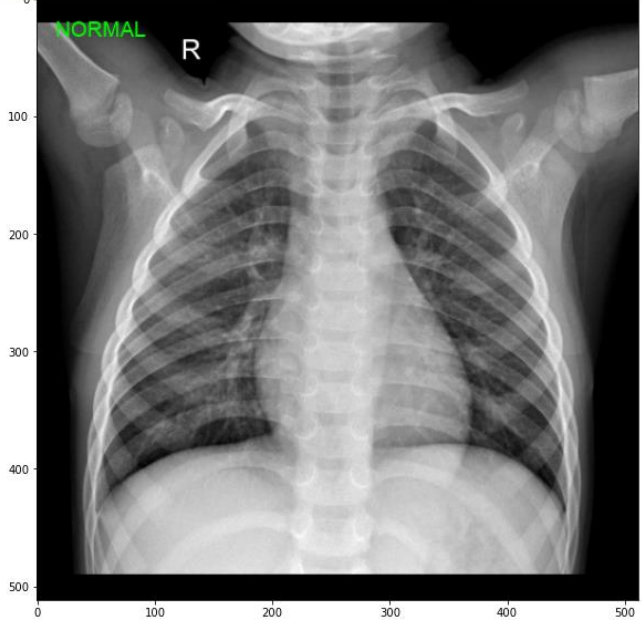


Image 1/1 D:\App4AI-2222\sdk\Jupyter-Image-Classification-YOLOv8-PyTorch-GPL-1\data\chest\test\NORMAL-11.jpg: 512x512 NORMAL 1.00, 22.0ms
Speed: 1.0ms pre-process, 22.0ms inference, 0.0ms postprocess per image at shape (1, 3, 512, 512)
Results saved to data\chest\inference-result

5_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.

```
dataset = "chest"
dataset_path = "data/%s" % (dataset)
source = "data/%s/test/" % (dataset)
image_size = 512

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

!run src/predict.py task=classify model=$weights_file source=$source imgs2=$image_size show=True device=$device project=$dataset_path r
```

Ultralytics YOLOv8.0.6 Python-3.9.12 torch-1.12.0+cu113 CUDA:0 (NVIDIA TITAN RTX, 24576MiB)
Fusing layers...
YOLOv8x-cls summary: 133 layers, 56125762 parameters, 0 gradients, 153.8 GFLOPs
Underkill Rate: 0(0.00%), Overkill Rate: 0(0.00%), Right Rate: 1(100.00%), Total: 1

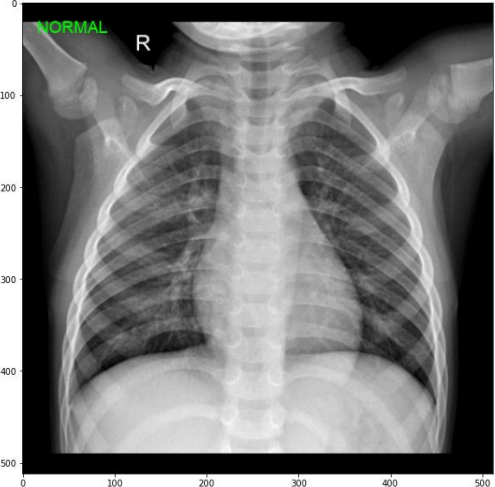


image 1/6 D:\App4AI-2222\sdk\Jupyter-Image-Classification-YOLOv8-PyTorch-GPL-1\data\chest\test\NORMAL-11.jpg: 512x512 NORMAL 1.00, 21.0ms

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.