Image-Classification-YOLOv8-PyTorch-GPL-Jupyter

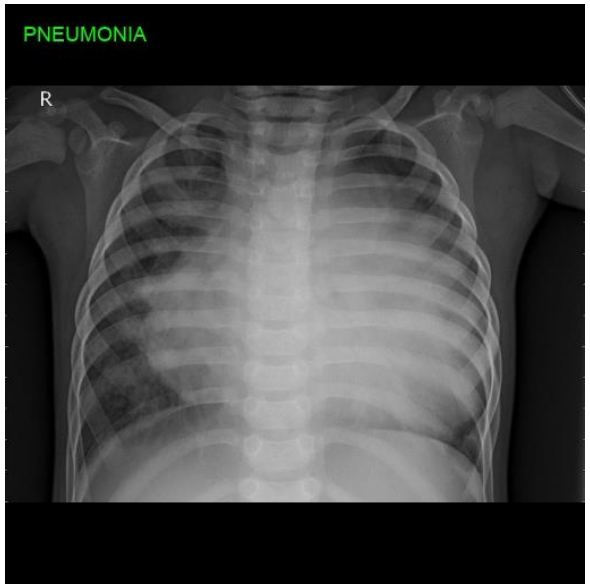
Ultralytics YOLOv8 is a cutting-edge, state-of-the-art (SOTA) model developed by Ultralytics.

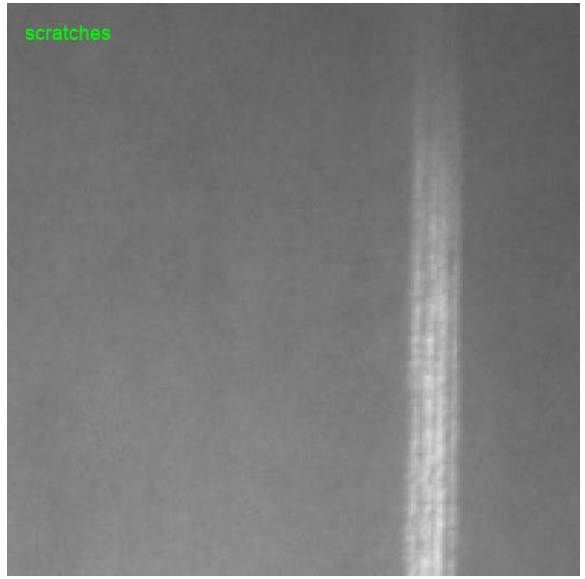
It builds on the previous successful version of YOLO, introducing new features and improvements that further enhance its performance and flexibility.

Version 20230223

Applications

- The YOLOv8 solution can be applied to factory defect classification, medical image classification, biological image classification, mask image classification, etc.

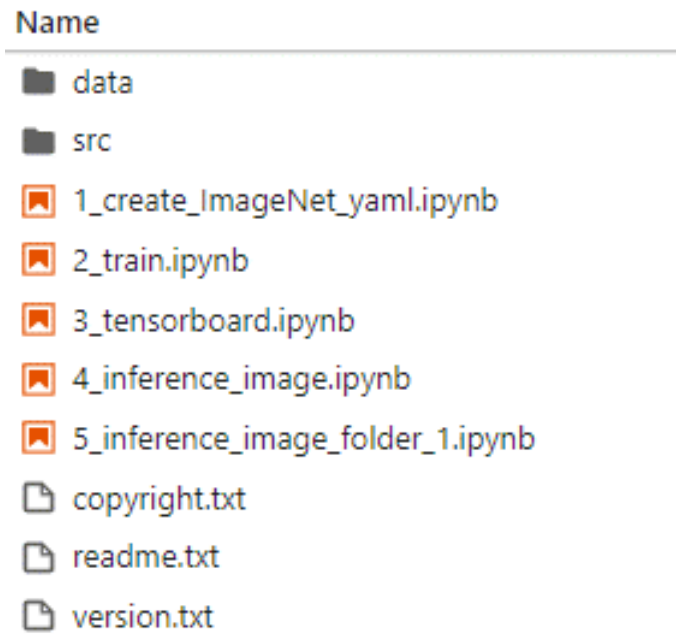




How to use

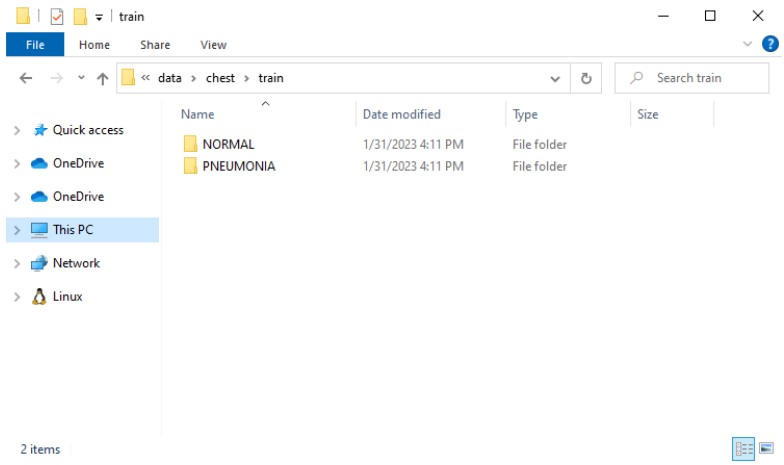
The main process is:

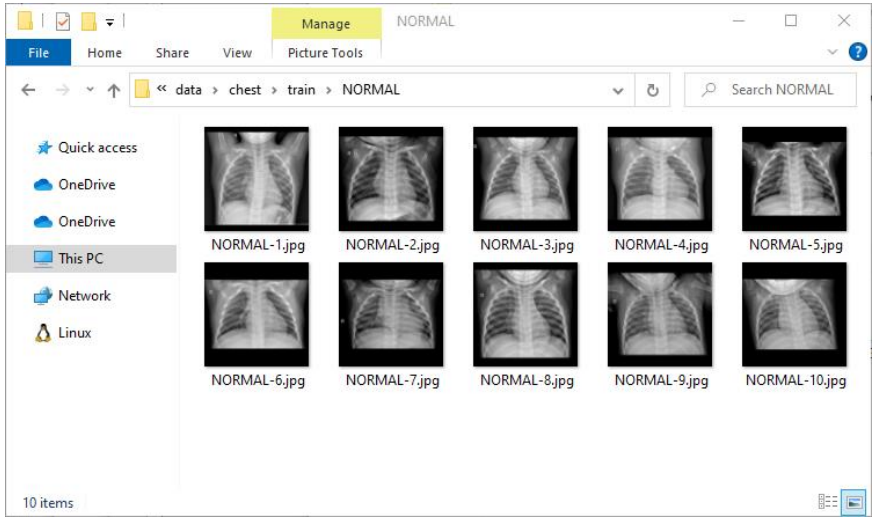
Prepare images -> Prepare files for training -> Training -> Inference



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".



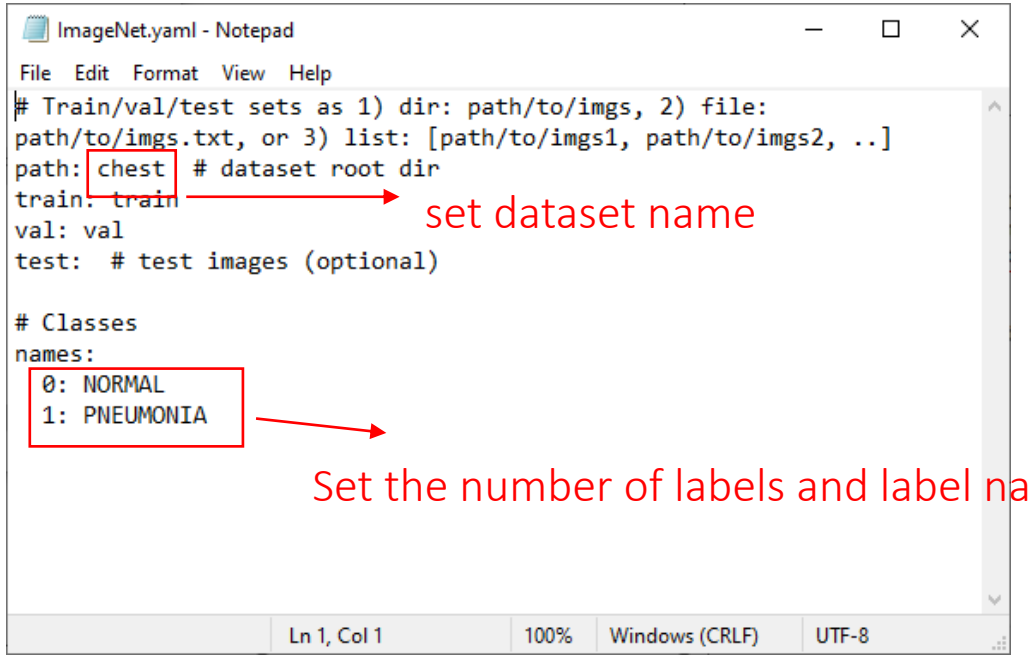


1_create_ImageNet_yaml

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.

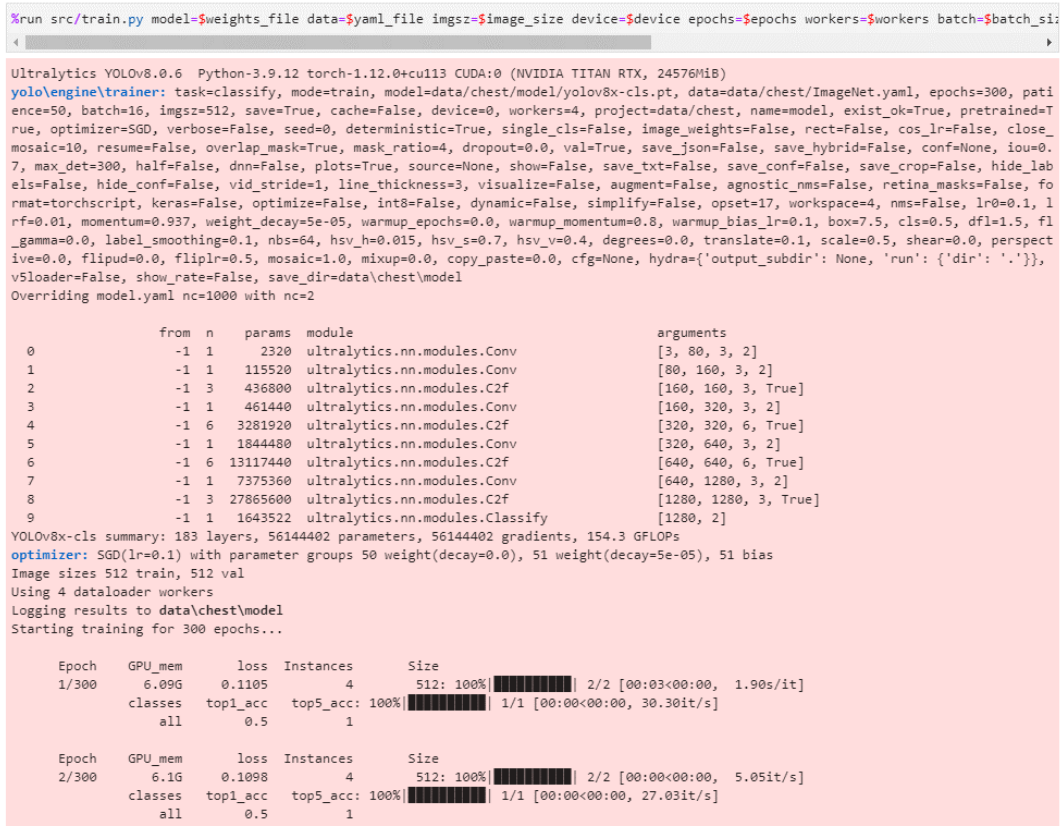


2_train.ipynb

Start training.

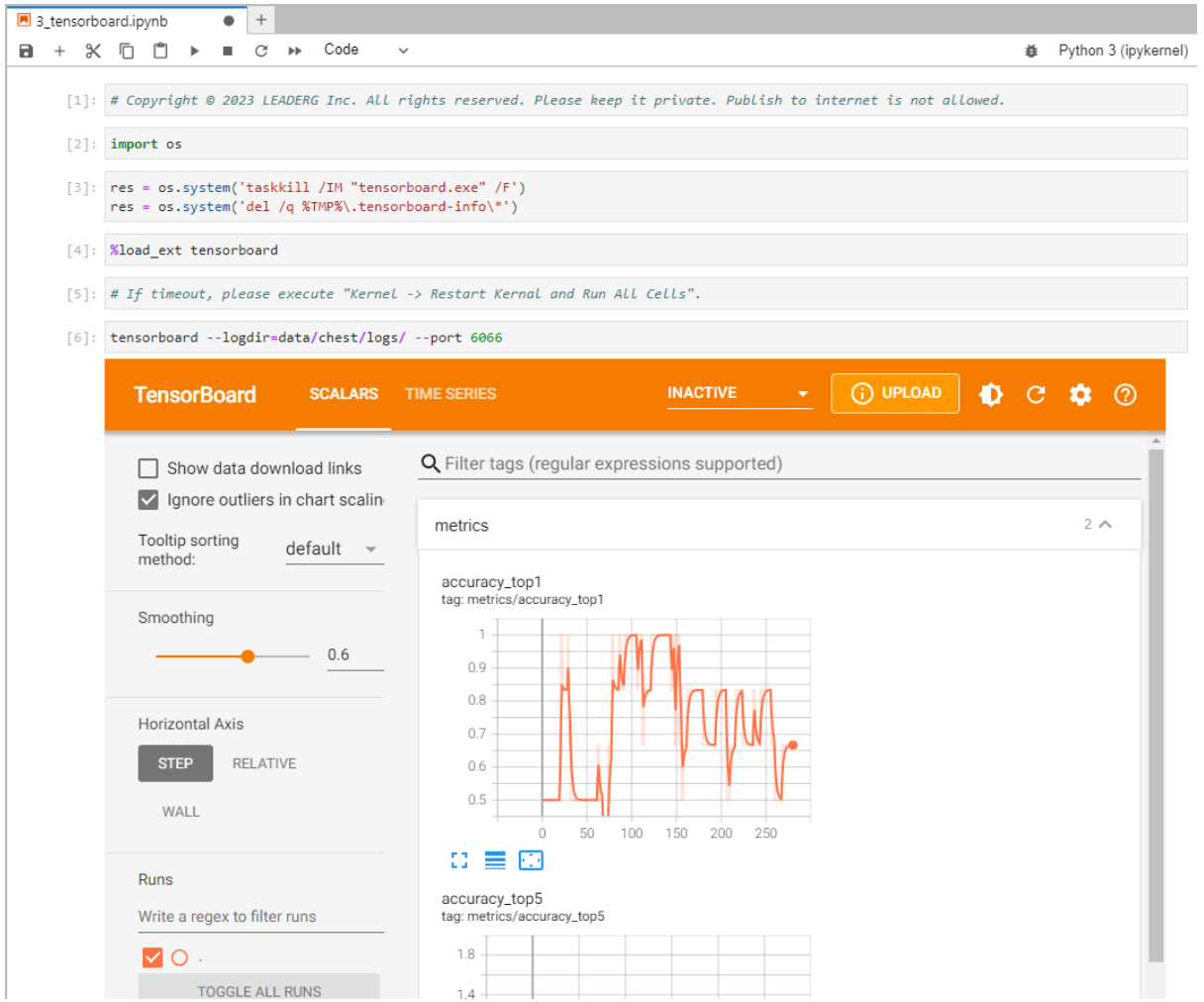
ipynb parameter:

- 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.



3_tensorboard.ipynb

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



4_inference_image.ipynb

Infer a single image.

ipynb parameter:

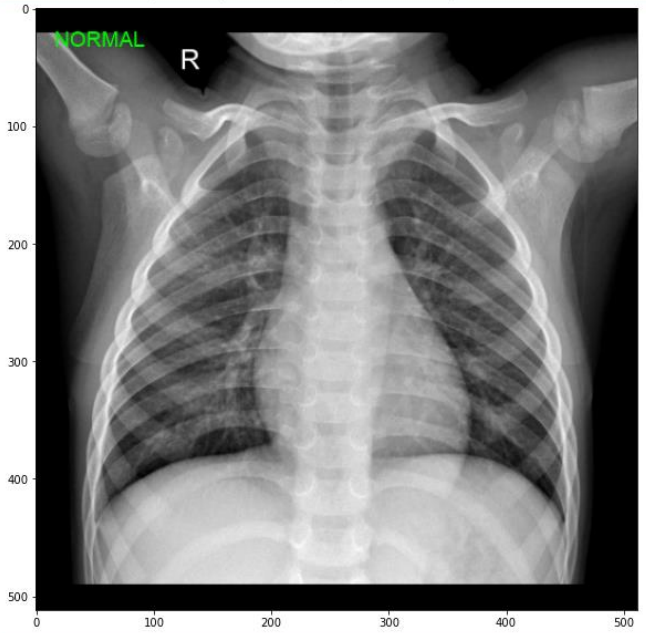
- dataset is the dataset name.
- source is the inferred image path.
- weights_file is the inference model path.

```
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 imgsz=$image_size show=True device=$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:

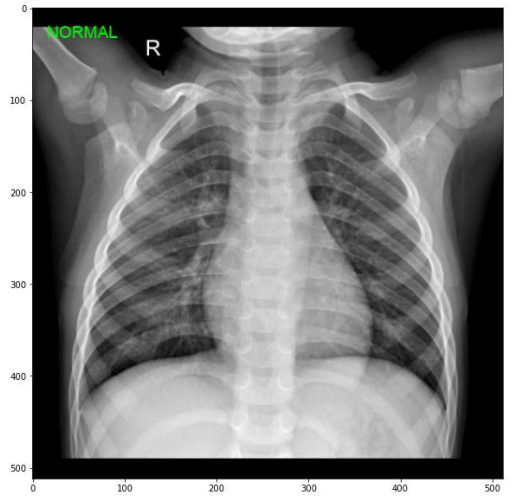
- dataset is the dataset name.
- source is the inferred image path.
- weights_file is the inference model path.

```
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 imgsz=$image_size show=True device=$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
-