

# YOLOv8Classification

Ultralytics YOLOv8 is a cutting-edge state-of-the-art (SOTA) model developed by Ultralytics. Improve and optimize on the basis of the previous successful YOLO.

YOLOv8Classification uses the classification function in YOLOv8.

Version 20230223

# Applications

YOLOv8Classification can be applied to factory defect classification, medical image classification, biological image classification, mask image classification, etc.



# How to use

The main process is:

Select dataset -> preprocessing (prepare images, set training parameters) -> training  
-> inference images

LEADERG - YOLOv8Classification - Version 1

**Dataset**

select dataset

**Prepare**

**1. create ImageNet yaml**

**Train**

Batch Size  Image Size   
Workers  Epochs   
GPU ID  (0, 1, 2, ... for Nvidia GPU)   
Pretrained Model

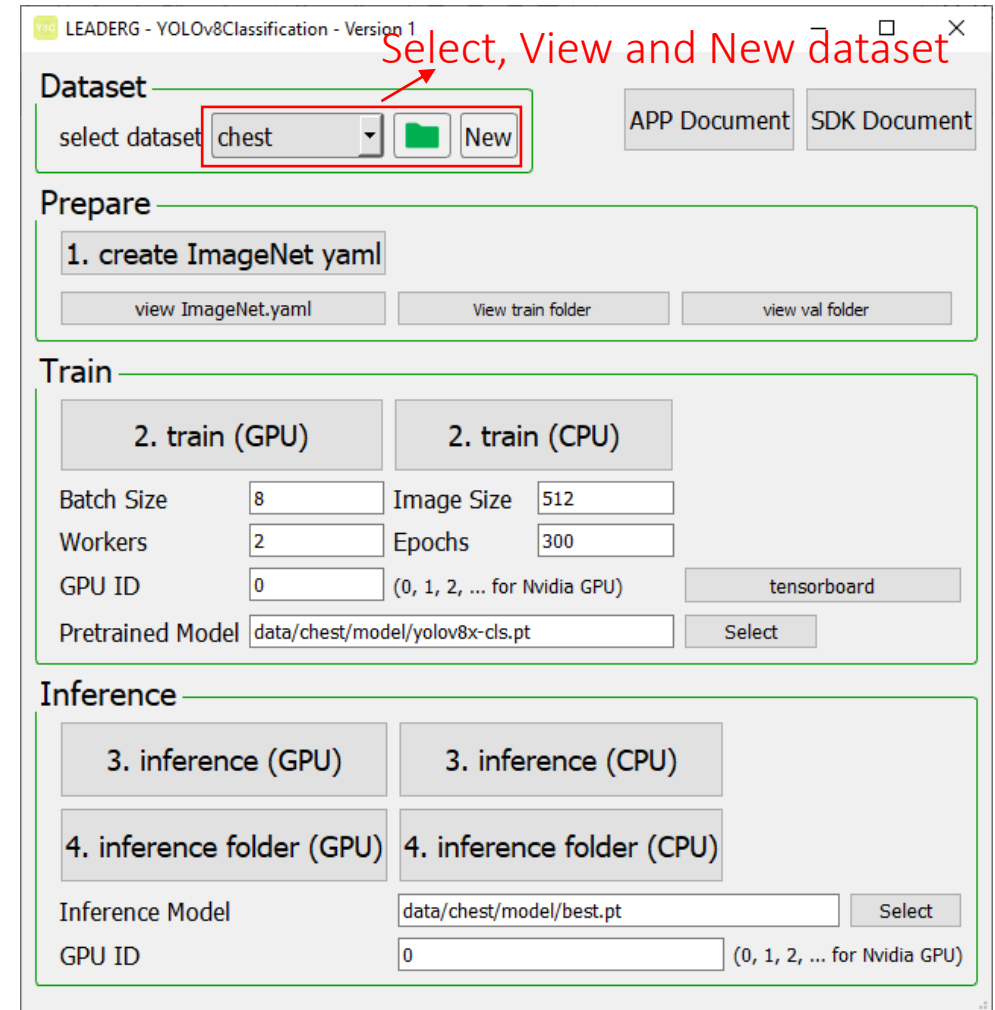
**Inference**

Inference Model    
GPU ID  (0, 1, 2, ... for Nvidia GPU)

# Select dataset

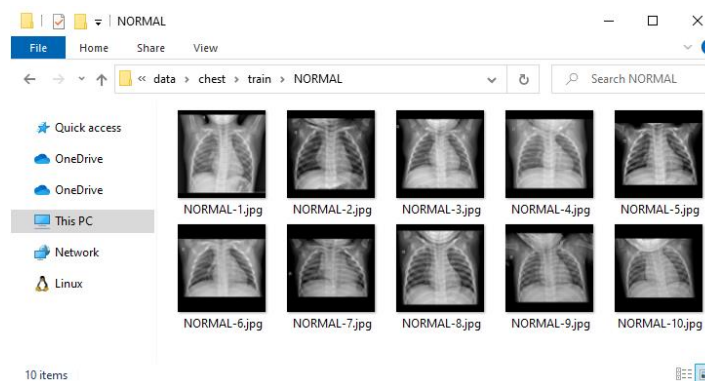
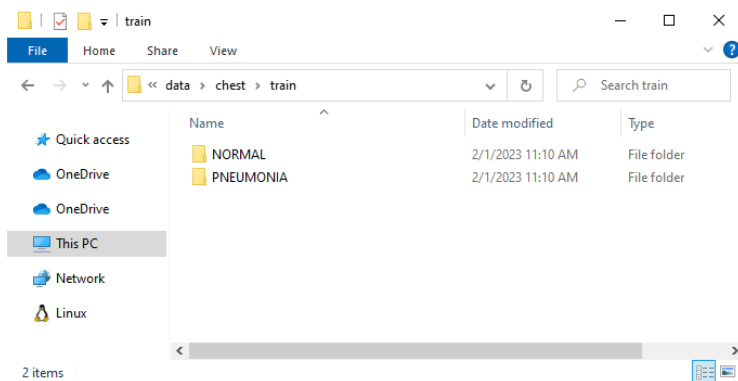
Select the dataset for training or inference.

- The “Folder” icon button next to the pull-down menu can open the data folder location, which is convenient for users to confirm and modify.
- If you want to create a new dataset by yourself, please press the "New" button, enter the dataset name in the pop-up window (only English and numbers can be used as the dataset name), and press "OK" to complete the creation, that is The name you just entered can be found in the pull-down menu.



# Prepare images

- After pressing the “folder” icon button, click the name of the dataset to be used, and put the images you want to train and infer in the images folder of the train, val, and test folders.
- For images placed in the train and val folders, please create a folder with the name according to the image category and then place the image in that category.
- Please use [label]-[sn].png file format for training, validation and test image. For example: "inclusion-1.png“, "inclusion-2.png".
- It is recommended to zoom or crop the train and val image files to a square image of approximately 512 x 512 size.



# 1.create ImageNet yaml

Once the image is ready, you can press a button to generate ImageNet.yaml.

You can press [view ImageNet.yaml] to confirm the content of the ImageNet.yaml file, 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
[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
```

# 2. train (GPU)

# 2. train (CPU)

Start training.

Note:

Pretrained Model is the pretrained model path used.

Epochs is the number of training epochs.

The screenshot shows the LEADERG - YOLOv8Classification - Version 1 application. The 'Train' section is active, with '2. train (GPU)' selected. A terminal window in the foreground displays training progress for epochs 2/300 through 7/300.

Epoch	GPU_mem	loss	Instances	Size	512: 100%	3/3	[00:00<00:00, 9.46it/s]
2/300	3.51G	0.07244	4	512: 100%	1/1	[00:00<00:00, 30.30it/s]	
3/300	3.51G	0.06676	4	512: 100%	1/1	[00:00<00:00, 24.39it/s]	
4/300	3.51G	0.06513	4	512: 100%	1/1	[00:00<00:00, 31.23it/s]	
5/300	3.51G	0.05084	4	512: 100%	1/1	[00:00<00:00, 8.98it/s]	
6/300	3.51G	0.04096	4	512: 100%	1/1	[00:00<00:00, 10.10it/s]	
7/300	3.51G	0.03563	4	512: 100%	1/1	[00:00<00:00, 32.26it/s]	

# 3. inference (GPU)

# 3. inference (CPU)

Infer a single image.

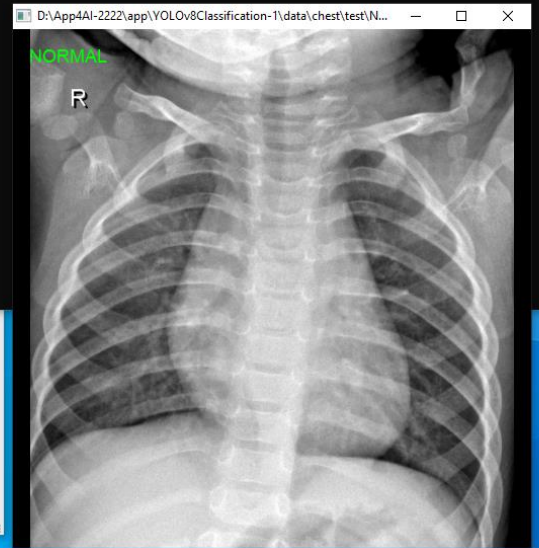
Choose the Inference Model yourself.

The screenshot displays a software interface for YOLOv8 classification. The interface is divided into several sections:

- Dataset:** A dropdown menu is set to "chest".
- Prepare:** A button labeled "1. create ImageNet yaml" is visible.
- Train:** A button labeled "2. train (GPU)" is visible. Below it, input fields for "Batch Size" (8), "Workers" (2), and "GPU ID" (0) are shown. A "Pretrained Model" field is set to "data/chest/model/best.pt".
- Inference:** Two buttons are visible: "3. inference (GPU)" (highlighted with a red box) and "3. inference (CPU)". Below them are buttons for "4. inference folder (GPU)" and "4. inference folder (CPU)". An "Inference Model" field is set to "data/chest/model/best.pt" and a "GPU ID" field is set to "0".

In the background, a terminal window shows the following output:

```
D:\App4AI-2222\app\YOLOv8Classification-1\bin\yolov8classification\predict.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\_DenseLa
yer.forward at 0x00000198B8C9DFE50>.
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\_DenseLa
yer.forward at 0x00000198B8CA00280>.
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...
YOLOv8x-cls summary: 133 layers, 56125762 parameters, 0 gradients, 153.8 GFLOPs
```





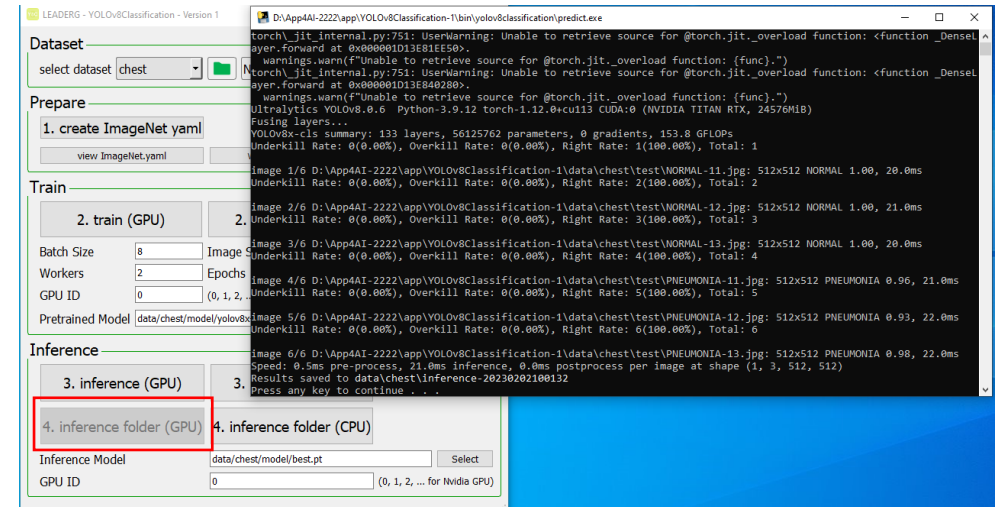
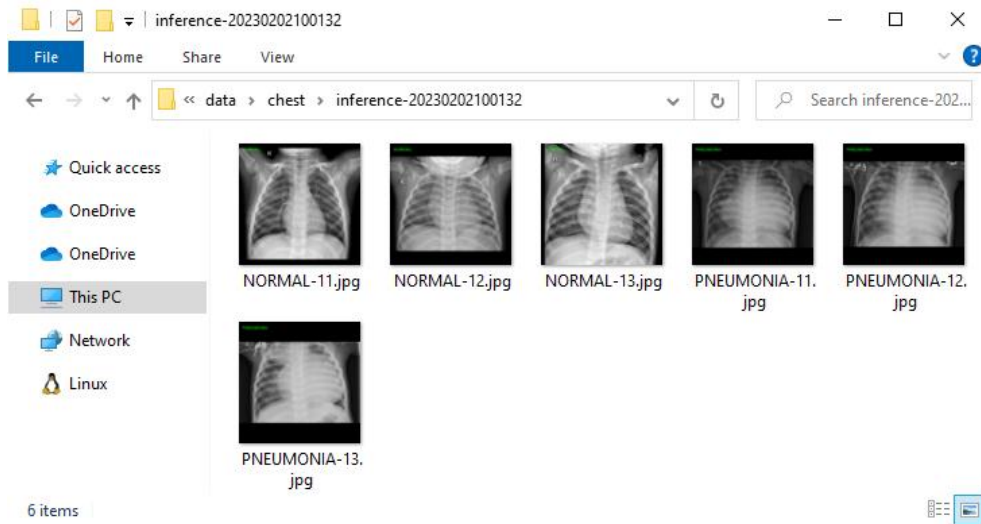
# 4. inference folder (GPU)

# 4. inference folder (CPU)

Infer all images in the folder.

Choose the Inference Model yourself.

Image results are stored in the "inference-XXX" folder.



# Reference

- Please refer to the readme.txt in the APP folder.
- LEADERG AppForAI: <https://www.leaderg.com/appforai-windows>
- Copyright © LEADERG INC. All rights reserved.