YOLOv8Classification

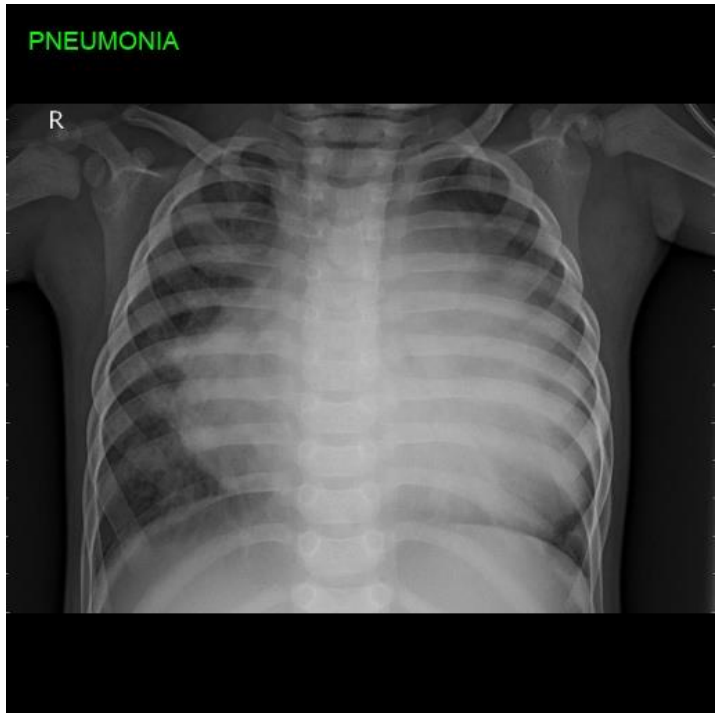
Ultralytics YOLOv8 is a cutting-edge state-of-the-art (SOTA) model developed by Ultralytics. Improve and optimize on the basis of the previous successful YOLO.

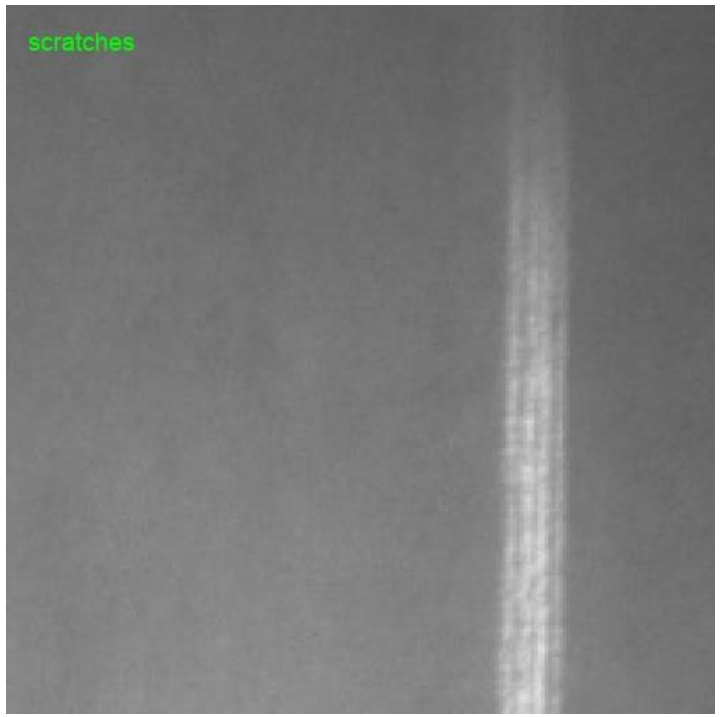
YOLOV8Classification uses the classification function in YOLOv8.

Version 20230223

Applications

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

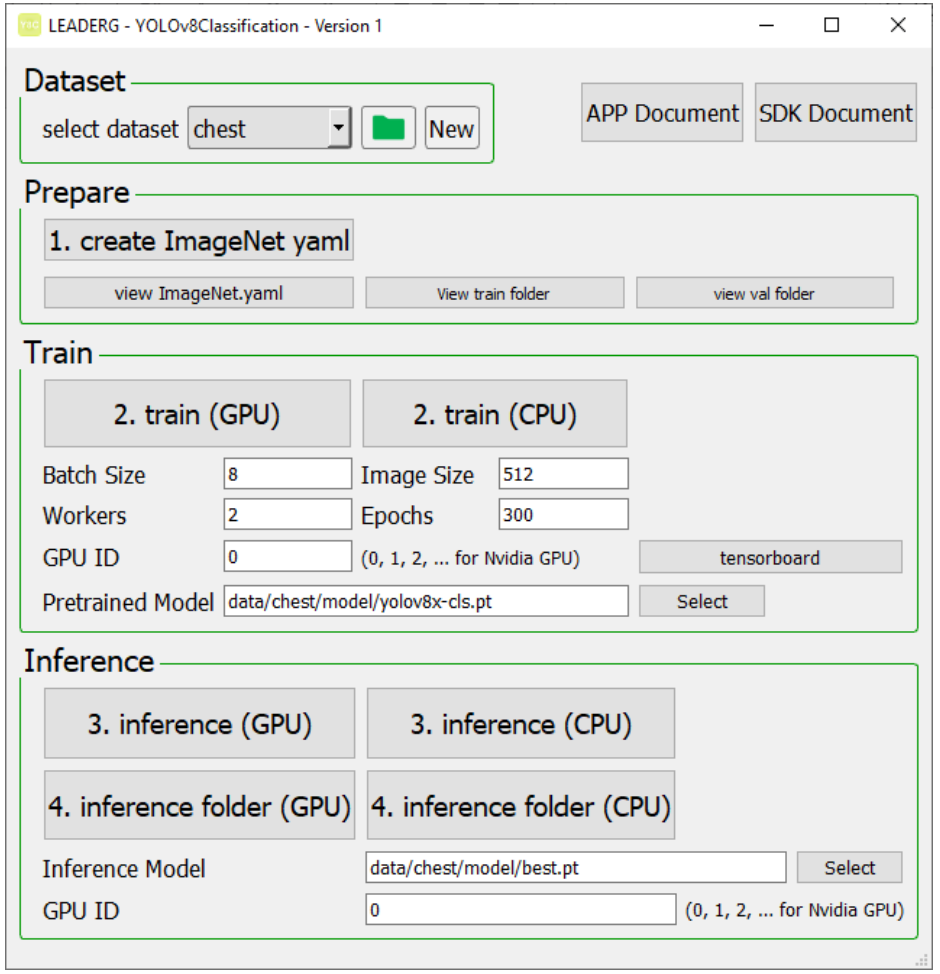




How to use

The main process is:

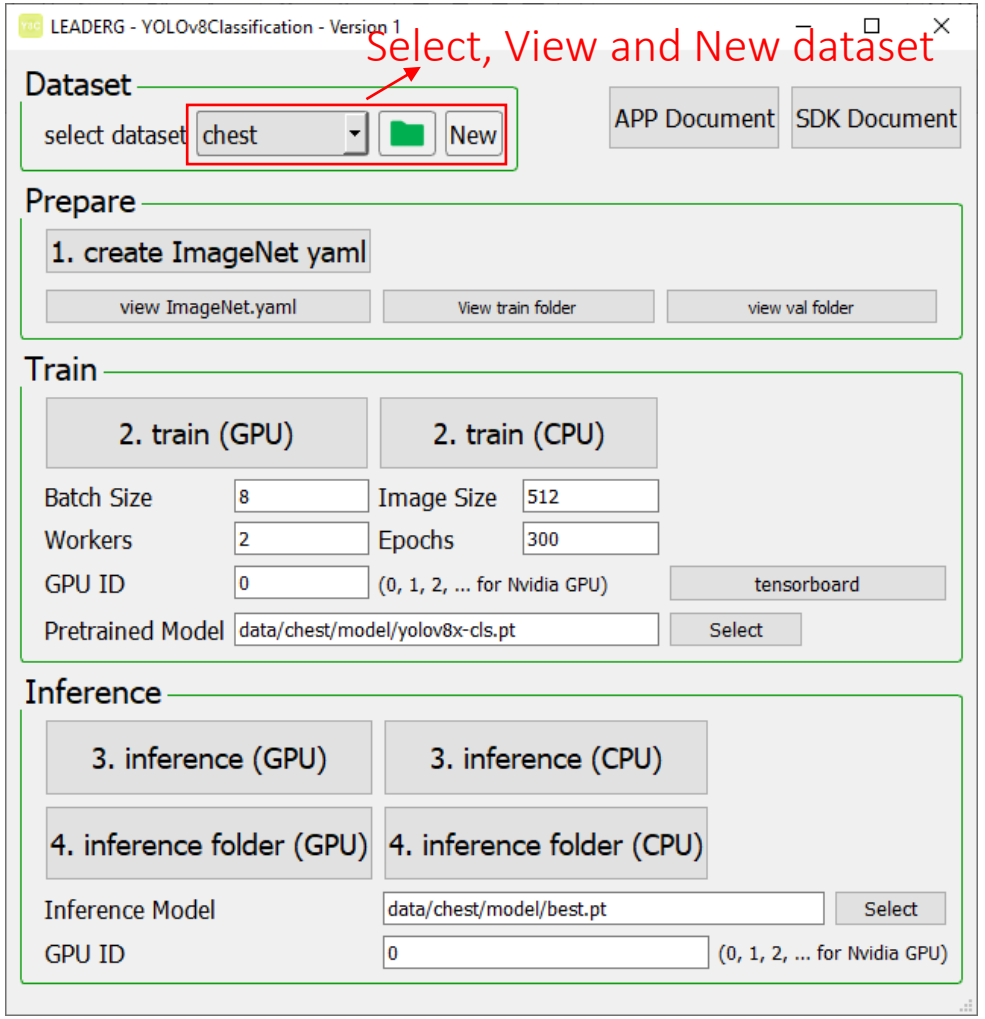
Select dataset -> preprocessing (prepare images, set training parameters) -> training -> inference images



Select dataset

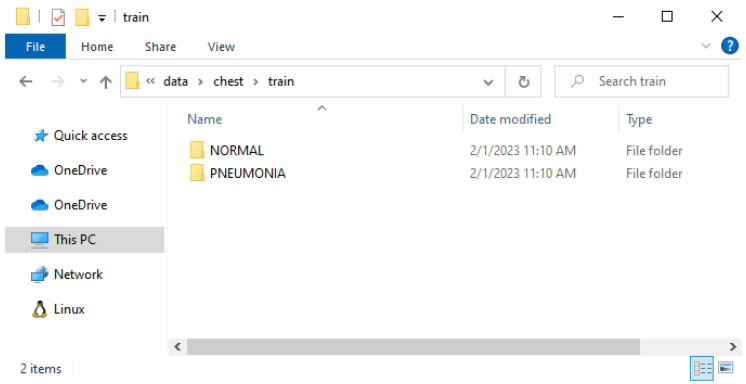
Select the dataset for training or inference.

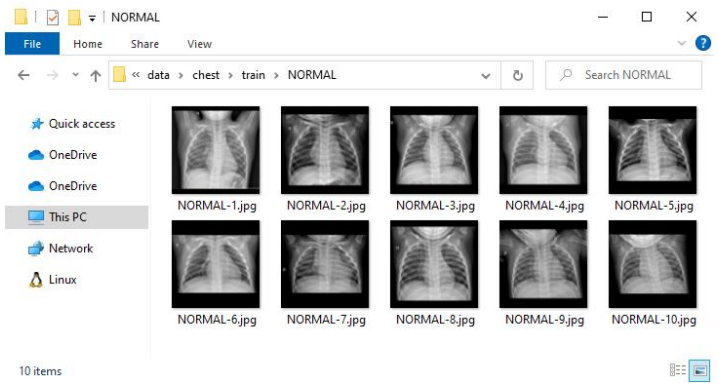
- The “Folder“ icon button next to the pull-down menu can open the data folder location, which is convenient for users to confirm and modify.
- If you want to create a new dataset by yourself, please press the "New" button, enter the dataset name in the pop-up window (only English and numbers can be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

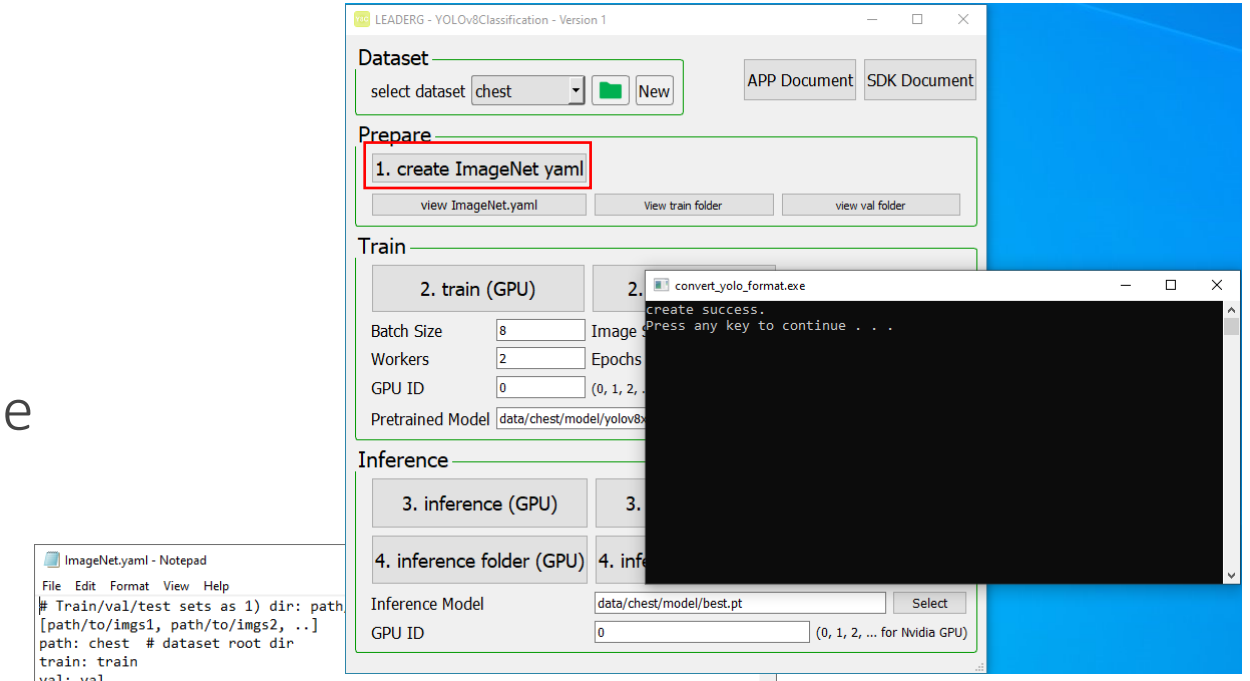


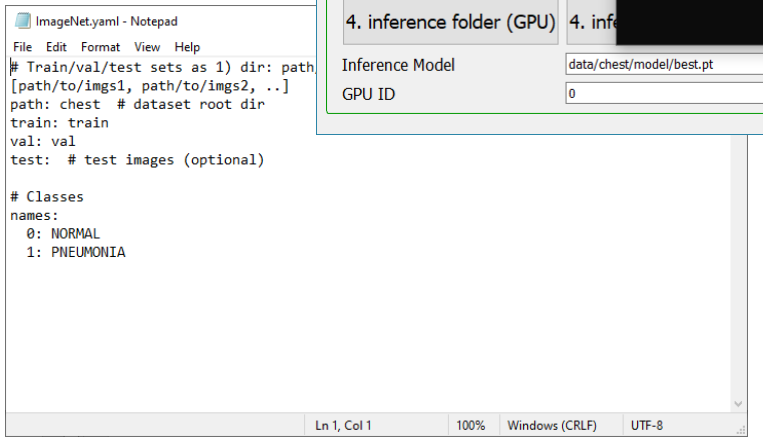


1.create ImageNet yaml

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.





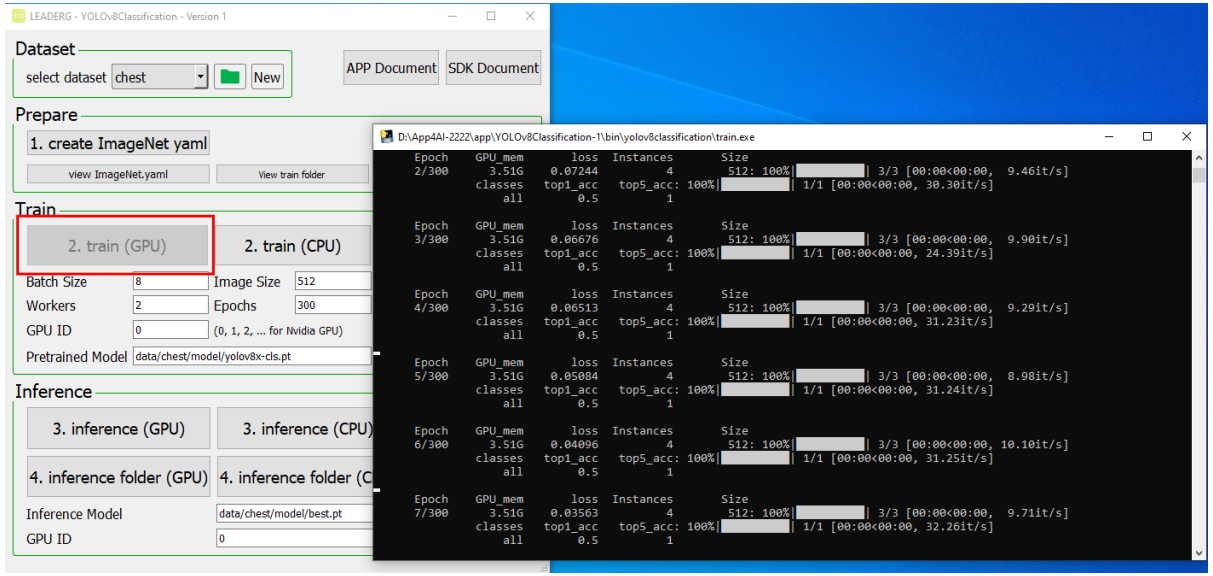
2. train (GPU)
2. train (CPU)

Start training.

Note:

Pretrained Model is the pretrained model path used.

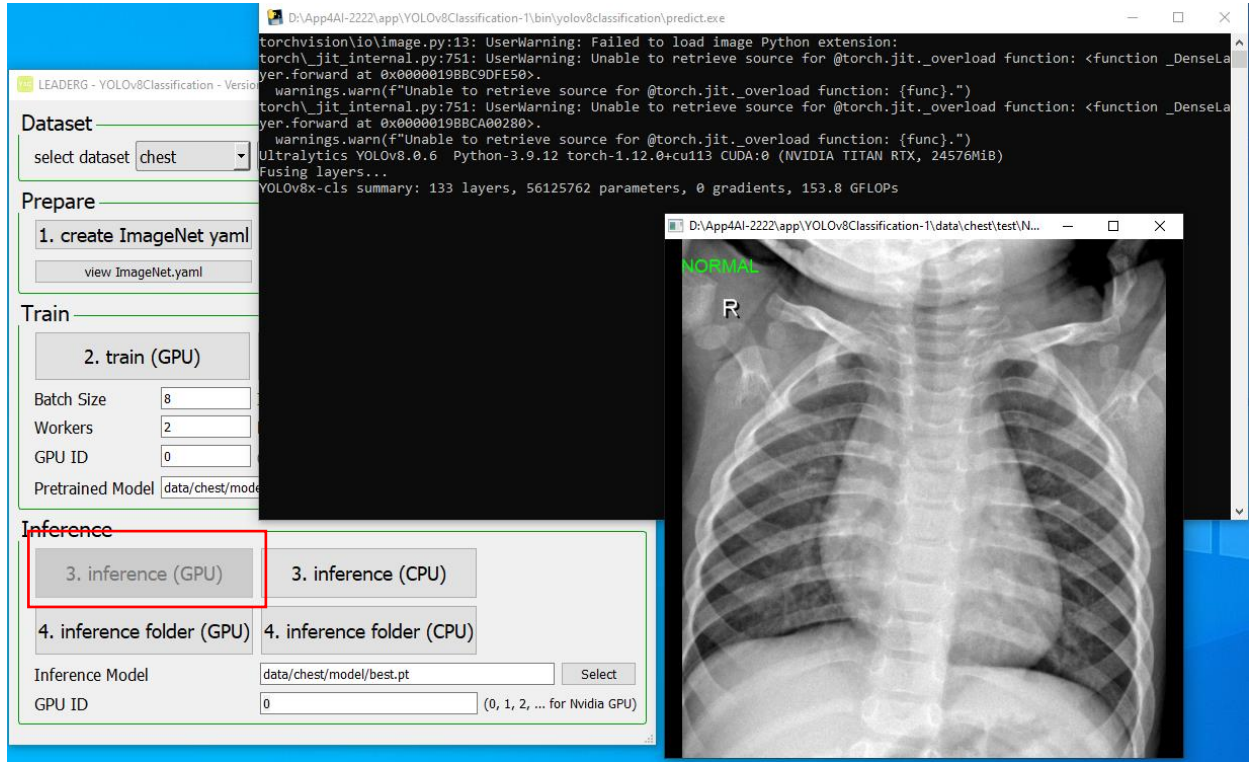
Epochs is the number of training epochs.



3. inference (GPU)
3. inference (CPU)

Infer a single image.

Choose the Inference Model yourself.

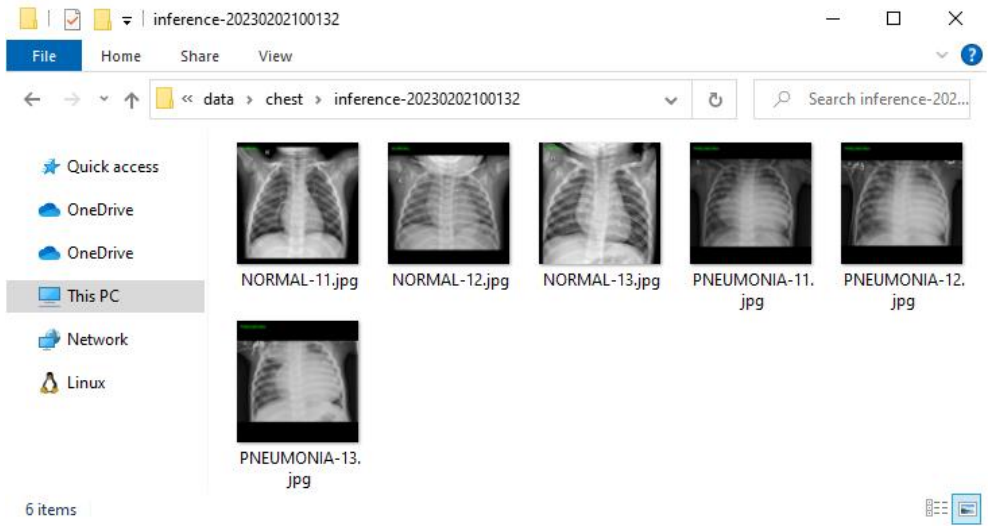


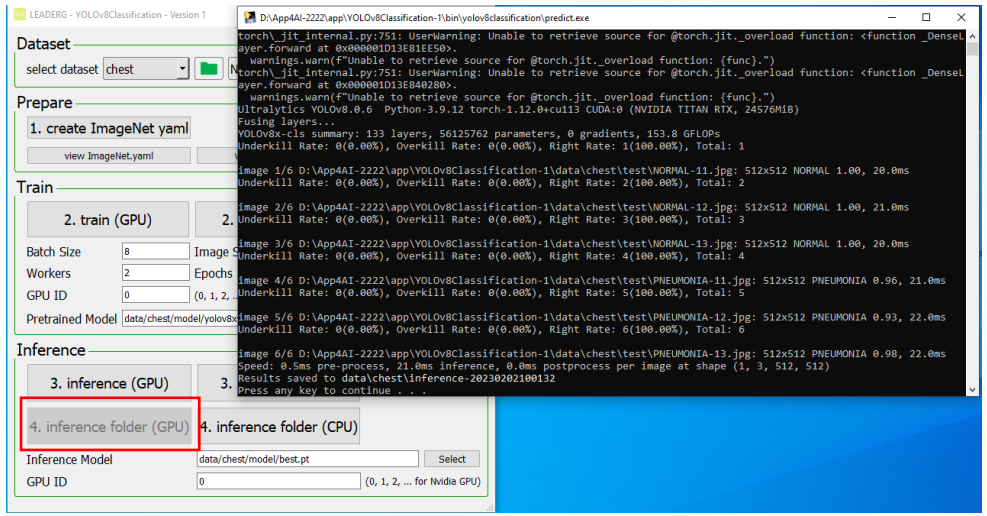
4. inference folder (GPU)
4. inference folder (CPU)

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
-