

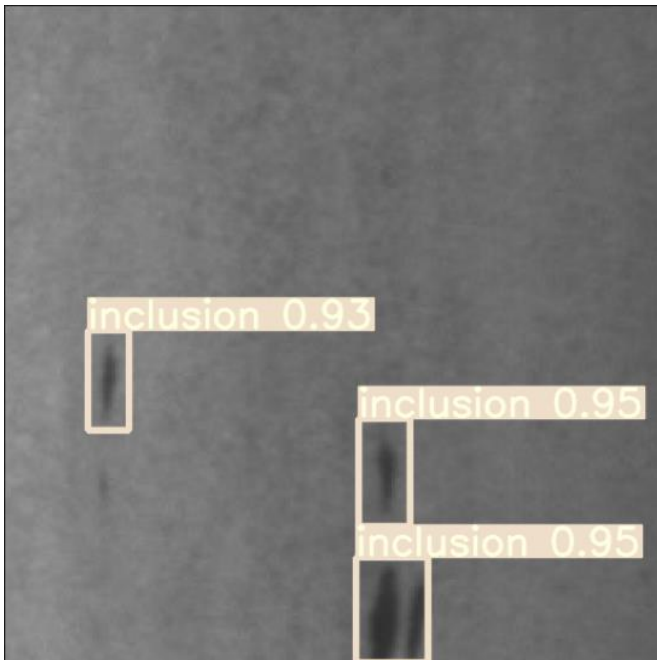
# YOLOR

The most powerful object detection algorithm in 2021.

Version 20230223

# Applications

YOLOR can be applied to factory defect detection, medical image analysis, biological image analysis, industrial security image analysis, mask image analysis, etc.



# How to Use

The main process is:

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

LEADERG - YOLOR - Version 4

**Dataset**

select dataset: tumor [Folder Icon] [Image Icon]

Document

**Prepare**

1. annotation pascal voc xml    2. convert yolo format    3. prepare txt

Annotation Port: 8080    View label.names  
View image annotation classes.txt

**Train**

4. train (GPU)    Batch Size: 16    Epochs: 3000    Workers: 8  
Image Size: 512    GPU Id: 0    (0, 1, 2, ... for Nvidia GPU)

4. train (CPU)    Pretrained Model: [Input Field]    Select

View voc.yaml    →    calculate anchors    →    Edit yolor\_csp\_x.cfg    tensorboard

**Inference**

Inference Model: data/tumor/model/best.pt    Select

Threshold: 0.5

GPU Id: 0    (0, 1, 2, ... for Nvidia GPU)

Webcam Id: 0

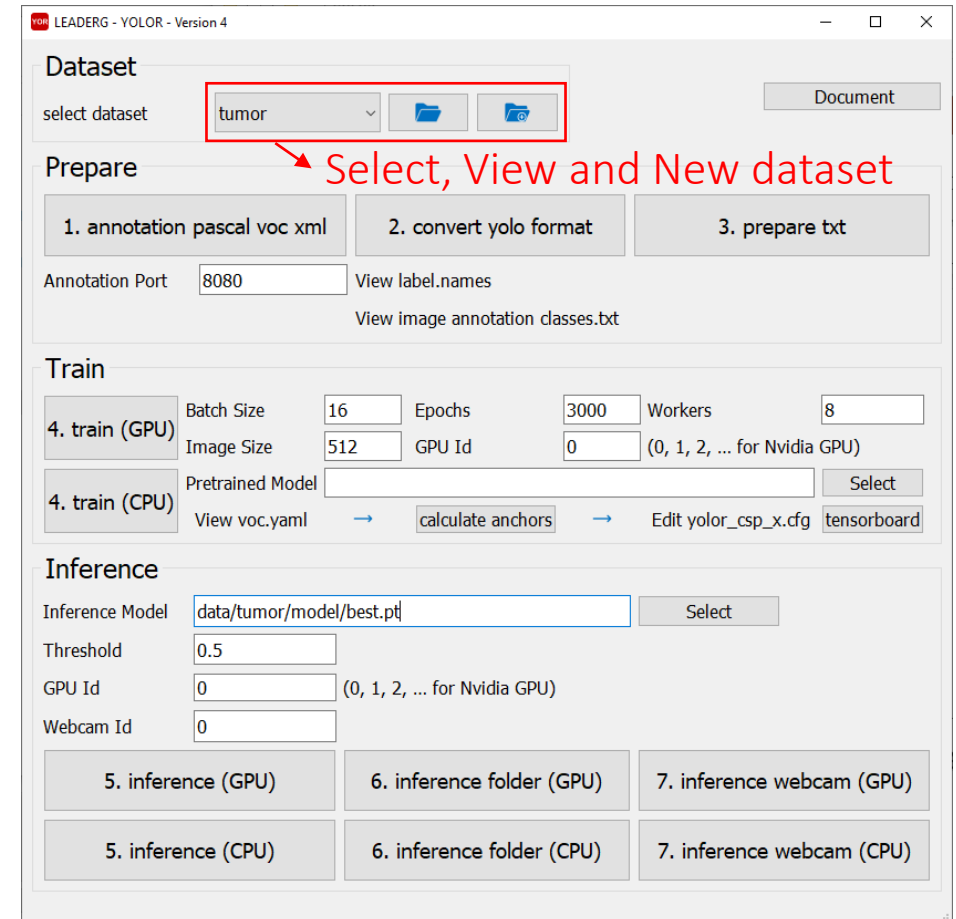
5. inference (GPU)    6. inference folder (GPU)    7. inference webcam (GPU)

5. inference (CPU)    6. inference folder (CPU)    7. inference webcam (CPU)

# Select Dataset

Select the dataset to train or infer.

- 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

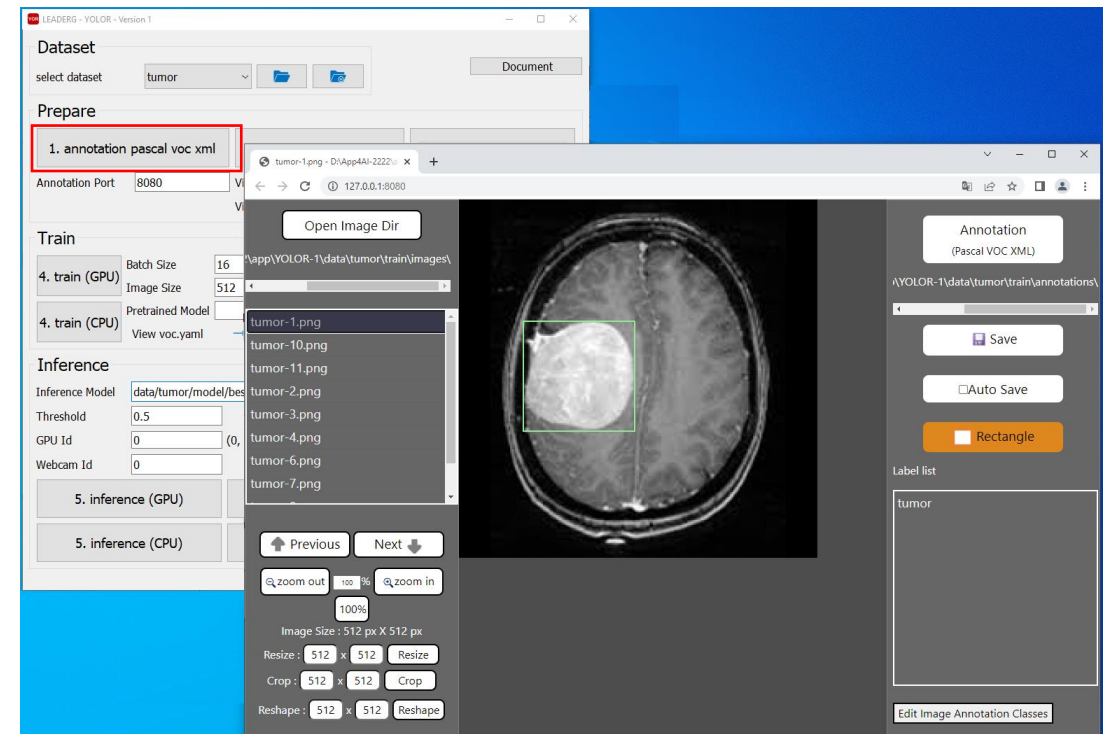
- After pressing the “folder” icon button, click the name of the dataset to be used, and put the images to the train, val, and test folders. The images in the train folder must be at least 9.
- Please use [label]-[sn].png file format for training, validation and test image. For example: "inclusion-1.png", "inclusion-2.png".
- It is recommended to crop the image files to a square image of 512 x 512 size by using “1.annotation pascal voc xml”.

# 1. annotation pascal voc xml

It opens the annotation webpage for image annotation. Only image with file extension .png is supported. And images can be cropped, resized, and labeled by using this tool..

Note:

Both training and validation images need to be annotated. To switch to annotated validation images, please press "Open Image Dir" at the top left to select "your own dataset/val/images", then press "SELECT", then press Under "Annotation" in the upper right, select "Pascal VOC XML" and then select "own dataset/val/annotations" and then press "SELECT".



Please refer to:

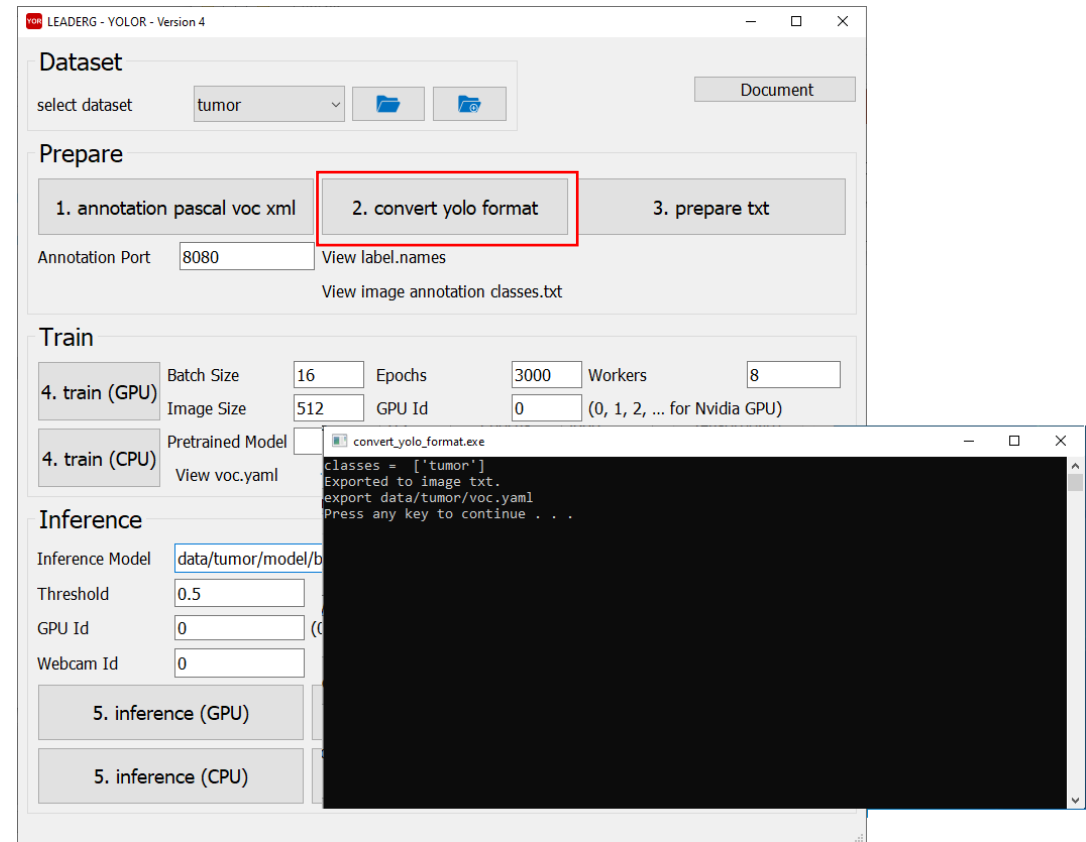
<https://www.leaderg.com/article/index?sn=11215>

## 2.convert yolo format

After confirming that the category names in the label.names file are correct, you can press the button to convert the voc xml annotation file to yolo format.

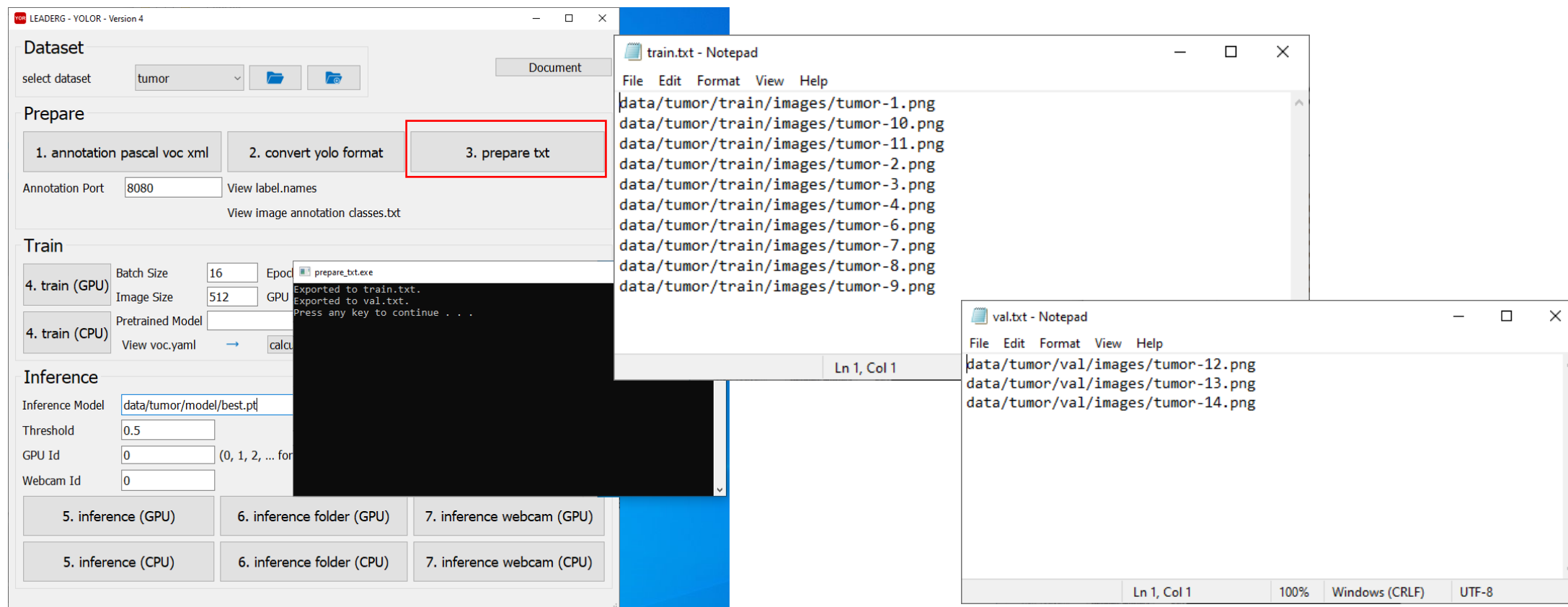
You can press [label.names] to edit the category names. The format uses a newline to distinguish multiple categories.

[image annotation classes.txt] is the category name used when the annotation webpage.



# 3.prepare txt

Generate image file list: “train.txt” and “val.txt”.





# Set yolor\_csp\_x.cfg related parameters

Set the content of the yolor\_csp\_x.cfg file in the dataset, set the number of categories.

The image displays three Notepad windows showing the configuration file `yolor_csp_x.cfg`. Red and blue annotations explain how to set the `filters` and `classes` parameters based on the number of categories.

**Red Annotations (Left and Top-Right Windows):**

- Left Window:** Shows the `filters` parameter being set to 18 in three different layers (lines 195, 196, and 197). A red arrow points from the text "Set to (number of categories + 5) x 3 example: Suppose there is a label to train So (1 + 5) x 3 = 18" to the `filters=18` values.
- Top-Right Window:** Shows the `filters` parameter being set to 18 in the `[yolo]` section (line 195). A red arrow points from the same text as above to the `filters=18` value.

**Blue Annotations (Bottom-Right Window):**

- Bottom-Right Window:** Shows the `classes` parameter being set to 1 in the `[yolo]` section (line 195). A blue arrow points from the text "Set the number of categories example: Suppose there is a label to train So fill in 1" to the `classes=1` value.

**Diagrammatic Connections:**

- Red arrows connect the red text explanation to the `filters=18` values in the left and top-right windows.
- Blue arrows connect the blue text explanation to the `classes=1` value in the bottom-right window.

# Set yolor\_csp\_x.cfg parameters

Press [Calculate Anchor] to calculate the appropriate anchor value for your dataset. Before running, please confirm whether the [label.names] and the category filled in the content are correct. Then press [edit yolov7.yaml] to set the anchor point. Fill in the numbers in the red box into the anchor points in parentheses.

The screenshot shows the LEADERG - YOLOR - Version 4 interface. The 'Dataset' section has 'tumor' selected. The 'Prepare' section has buttons for '1. annotation pascal voc xml', '2. convert yolo format', and '3. prepare txt'. The 'Train' section has buttons for '4. train (GPU)' and '4. train (CPU)'. The 'Inference' section has buttons for '5. inference (GPU)' and '5. inference (CPU)'. A terminal window shows the output of the 'gen\_anchors.exe' command, displaying the calculated anchor values. Two Notepad windows show the 'yolor\_csp\_x.cfg' file with the calculated anchor values filled in.

**Terminal Output:**

```
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 0x0000028783CDC280>.
warnings.warn(F"Unable to retrieve source for @torch.jit._overload function: {func}.")
Scanning labels data\tumor\train\labels.cache3 (10 found, 0 missing, 0 empty, 0 duplicate, for 10 images): 10it [00:00, 5003.34it/s]
Running kmeans for 9 anchors on 11 points...
thr=0.25: 1.0000 best possible recall, 8.55 anchors past thr
n=9, img_size=512, metric_all=0.571/0.978-mean/best, past_thr=0.590-mean: 40,30, 51,57, 75,45, 67,80, 91,86, 83,117, 120,114, 128,152, 159,157
Evolving anchors with Genetic Algorithm: 100% | 1000/1000 [00:00<00:00, 4366.85it/s]
thr=0.25: 1.0000 best possible recall, 8.55 anchors past thr
n=9, img_size=512, metric_all=0.571/0.978-mean/best, past_thr=0.590-mean: 40,30, 51,57, 75,45, 67,80, 91,86, 83,117, 120,114, 128,152, 159,157
anchors = 40,30, 51,57, 75,45, 67,80, 91,86, 83,117, 120,114, 128,152, 159,157
Press any key to continue...
```

**yolor\_csp\_x.cfg - Notepad (Top):**

```
pad=1
filters=18
activation=linear

[control_channels]
from=195

[yolo]
mask = 0,1,2
anchors = 23, 24, 44, 44, 69, 70, 150, 90, 80,175, 200,203, 456,141, 152,450, 456,454
classes=1
num=9
```

**yolor\_csp\_x.cfg - Notepad (Bottom):**

```
pad=1
filters=18
activation=linear

[control_channels]
from=197

[yolo]
mask = 6,7,8
anchors = 23, 24, 44, 44, 69, 70, 150, 90, 80,175, 200,203, 456,141, 152,450, 456,454
classes=1
num=9
```

Fill in the value displayed by CMD

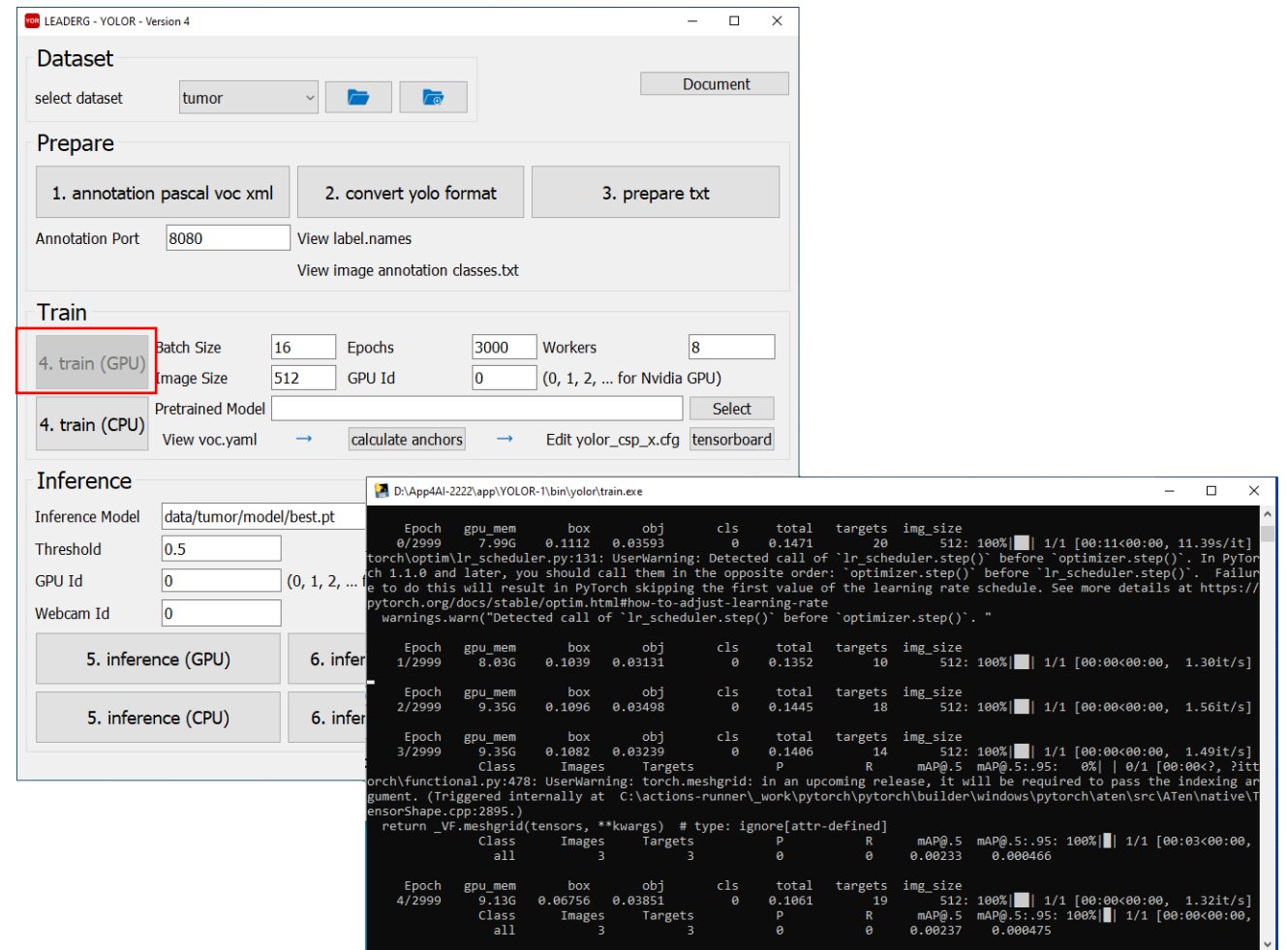
# 4. train

Start training.

Note:

Pretrained model file is the pretrained model path to be used. Empty means not to use the pretrained model for training.

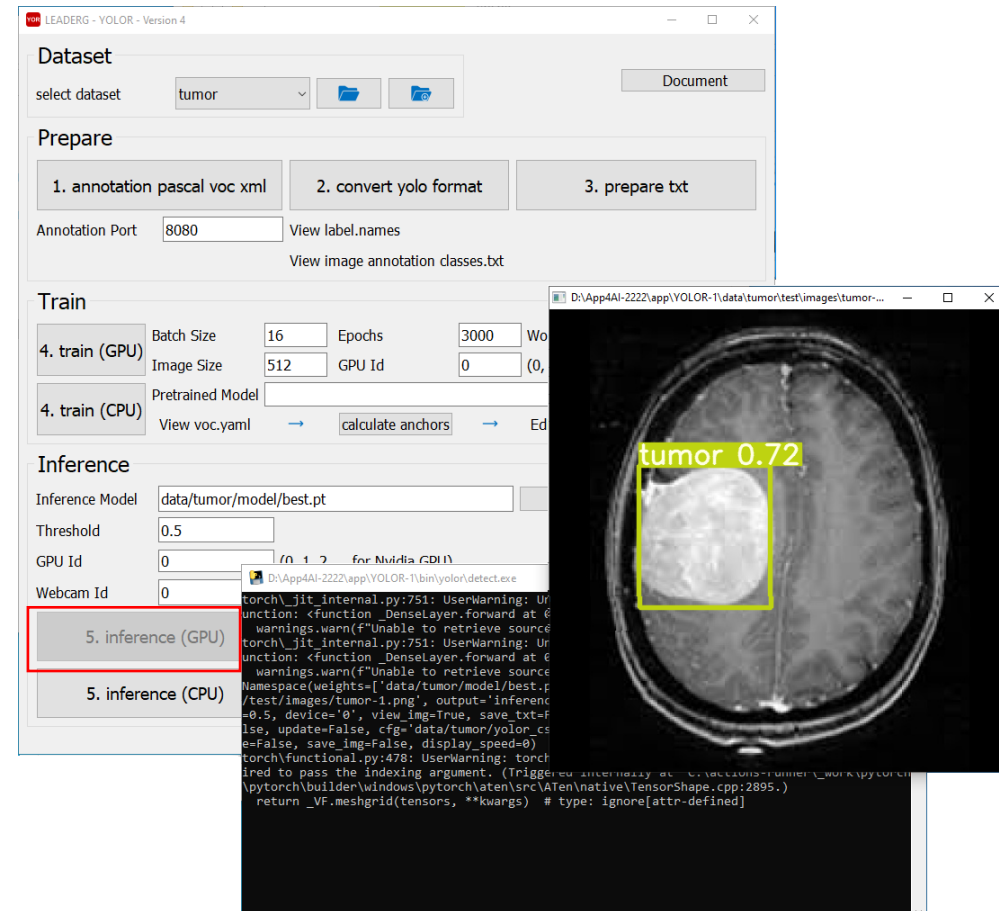
Epochs is the number of training epochs.



# 5. inference

Infer a single image.

Choose the Inference Model.

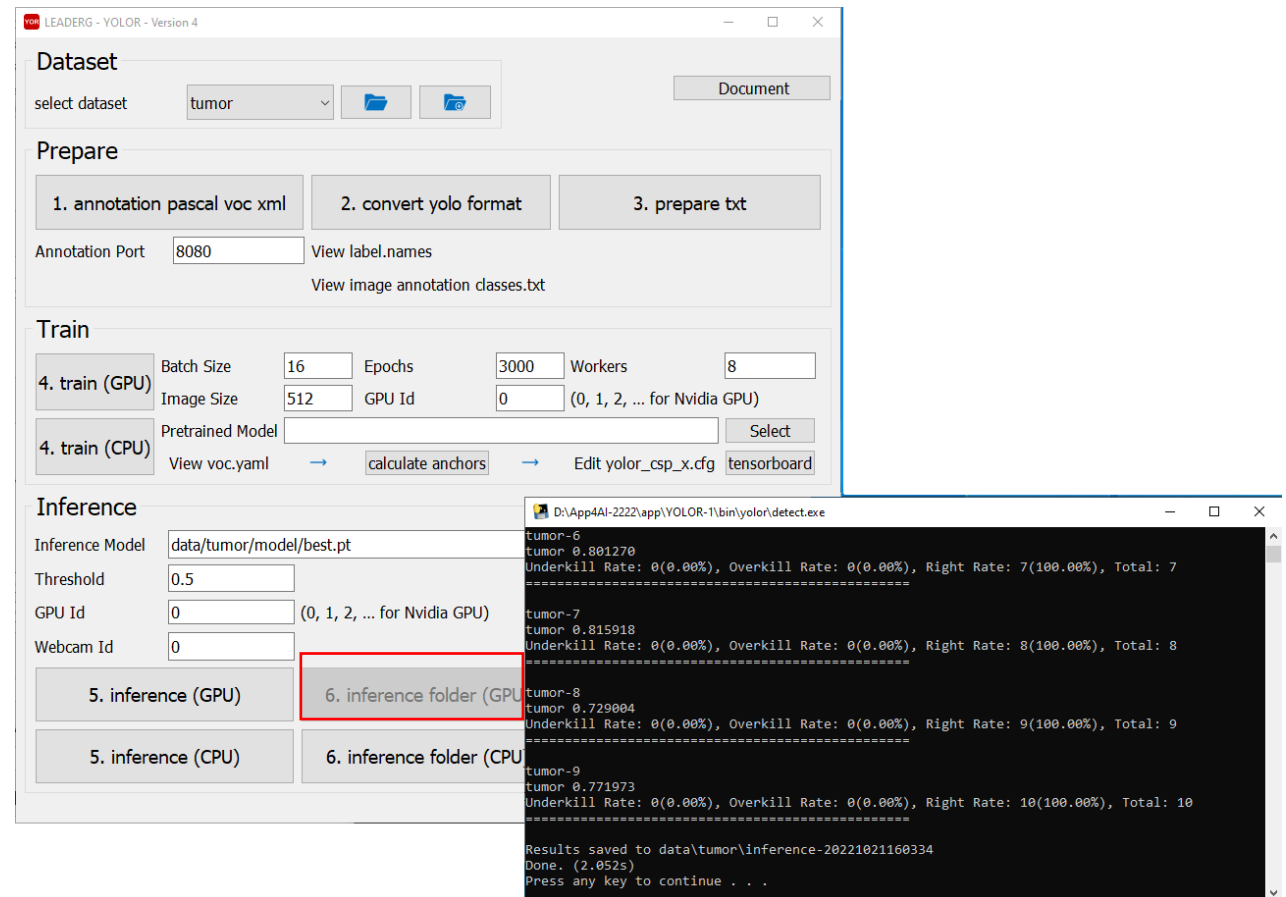
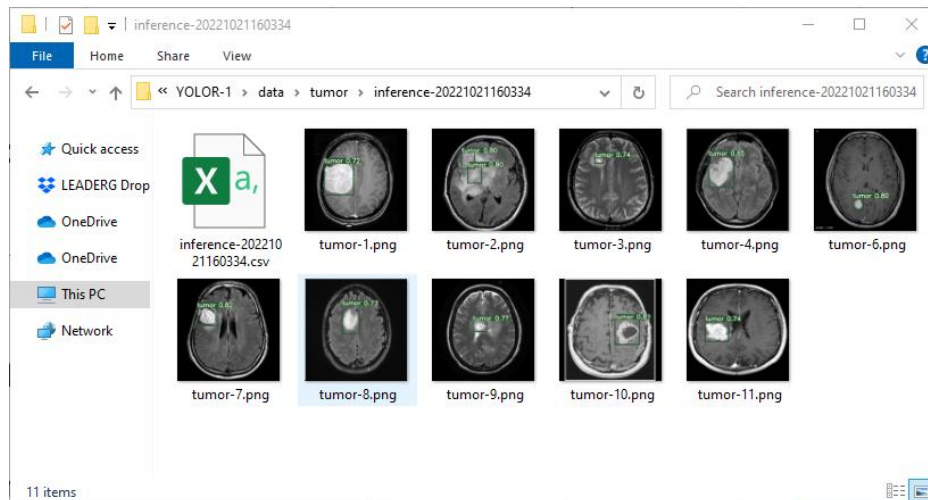


# 6. inference folder

Infer all images in the folder.

Choose the Inference Model.

Coordinate information and image results are stored in the "inference-XXX" folder.

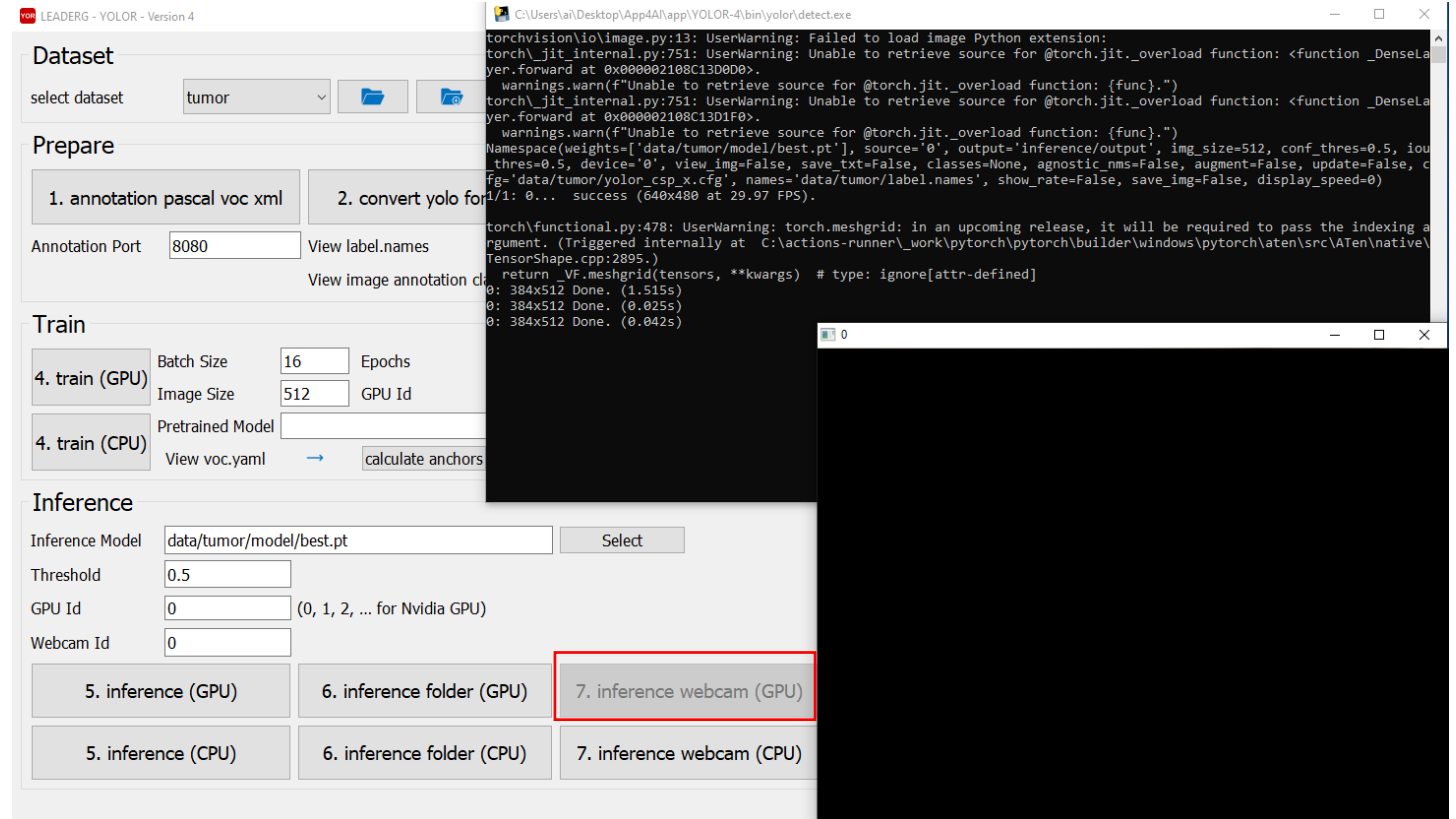


# 7. inference webcam

Infer the image of the webcam.

The Webcam Id parameter below can be set for the specified webcam.

Choose the Inference Model.



# 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.