

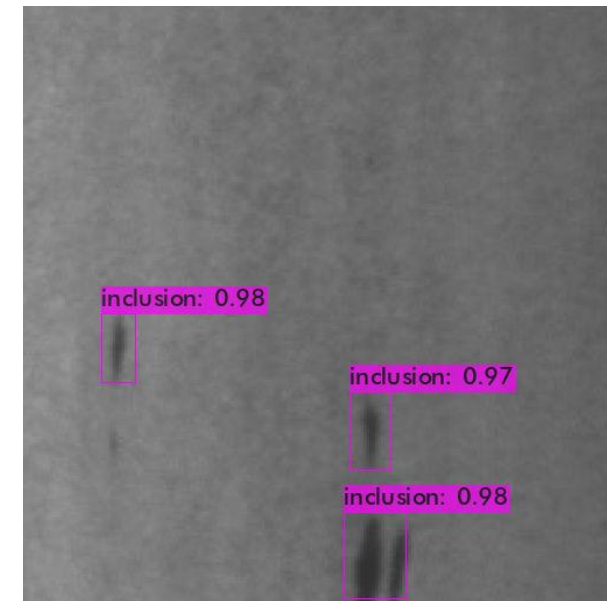
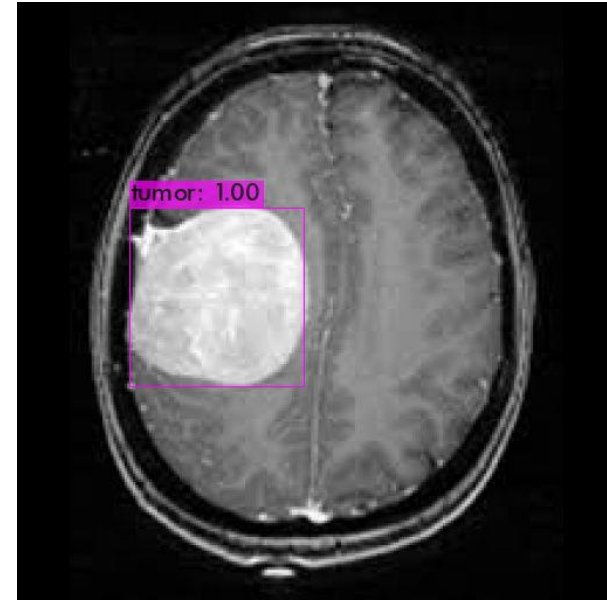
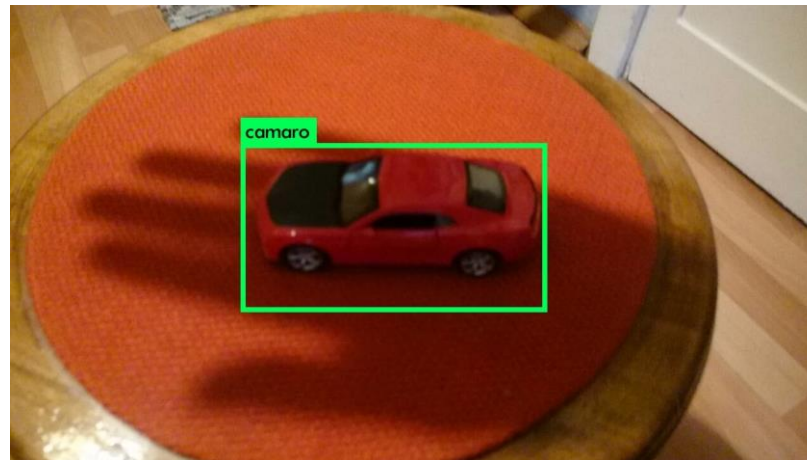
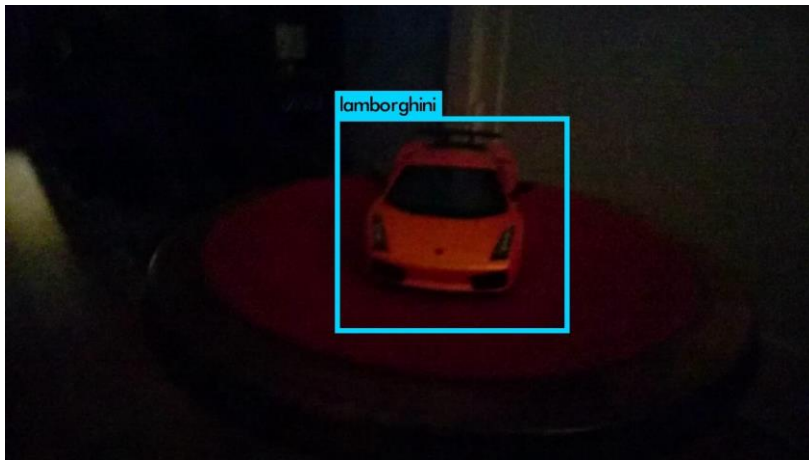
# YOLOv4

Russian Alexey Bochkovskiy, the maintainer of YOLO Darknet, found that the CSPNet detector developed by Wang Jianyao, the post-doc of the Chinese Academy of Sciences and director Liao Hongyuan, was fast and good, so he invited the Chinese Academy of Sciences to develop YOLOv4 for the backbone, and did various parts of the previous generation of YOLOv3. The improvement can not only maintain a certain detection speed, but also greatly improve the detection accuracy and reduce the usage of hardware. After integrating YOLOv4 into an APP, it is more convenient for everyone to use it quickly.

Version 20230223

# Applications

- YOLOv4 solutions can be applied to factory defect detection, medical image analysis, biological image analysis, industrial security image analysis, mask image analysis, luxury car brand 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 - YOLOv4 - Version 21

**Dataset**

Select Dataset: tumor View New

**Prepare**

1. annotation pascal voc xml 2. convert yolo format 3. prepare train txt 4. prepare val txt 5. prepare config file

**Parameters**

GPU ID: 0 Inference Threshold: 0.7

**Train**

6. train (GPU) 6. train (CPU)

**Inference**

7. inference (GPU) 8. inference webcam (GPU) 9. inference folder (GPU)  
7. inference (CPU) 8. inference webcam (CPU) 9. inference folder (CPU)

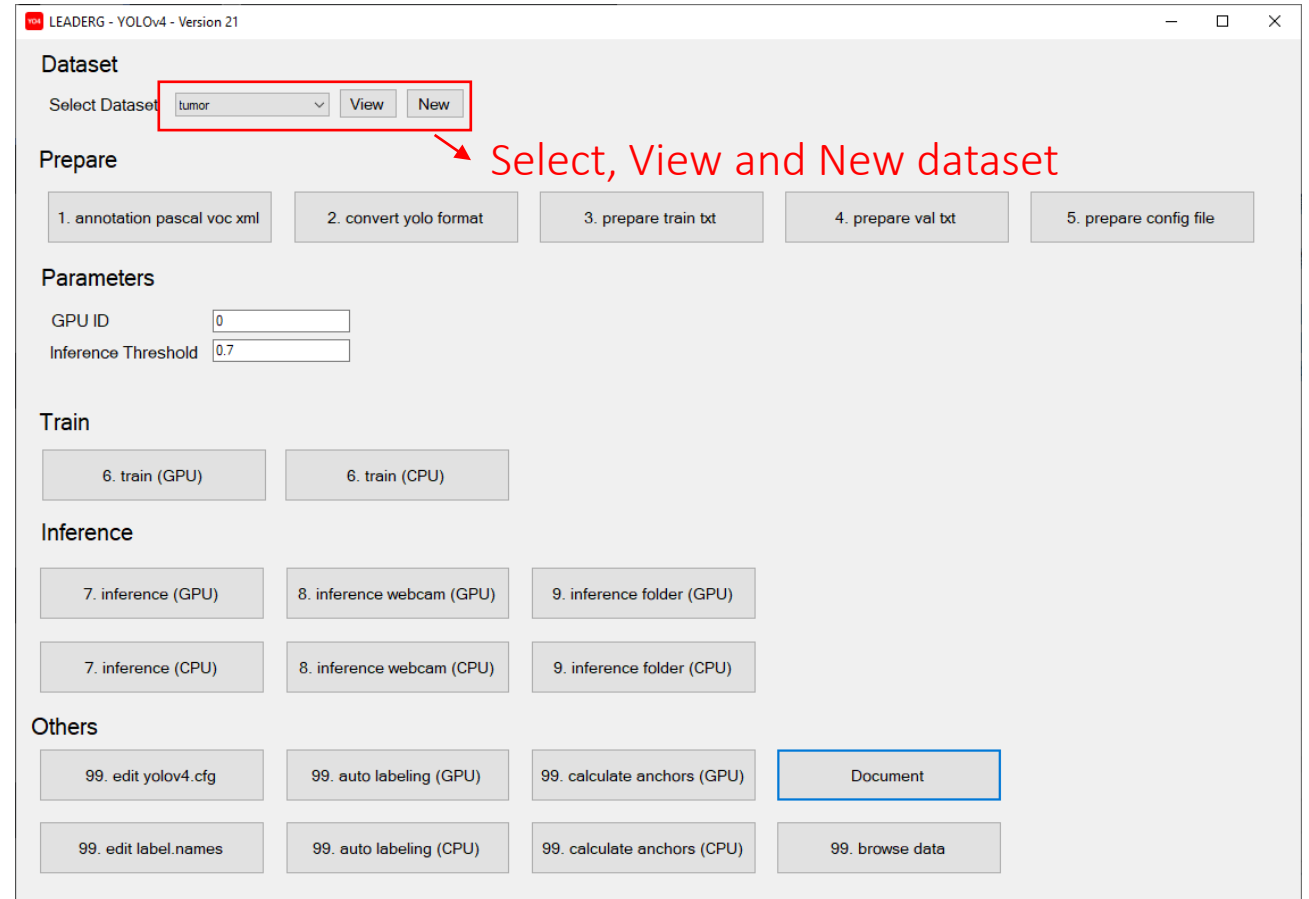
**Others**

99. edit yolov4.cfg 99. auto labeling (GPU) 99. calculate anchors (GPU) Document  
99. edit label.names 99. auto labeling (CPU) 99. calculate anchors (CPU) 99. browse data

# Select dataset

Select the dataset to train or infer.

- The "View" 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

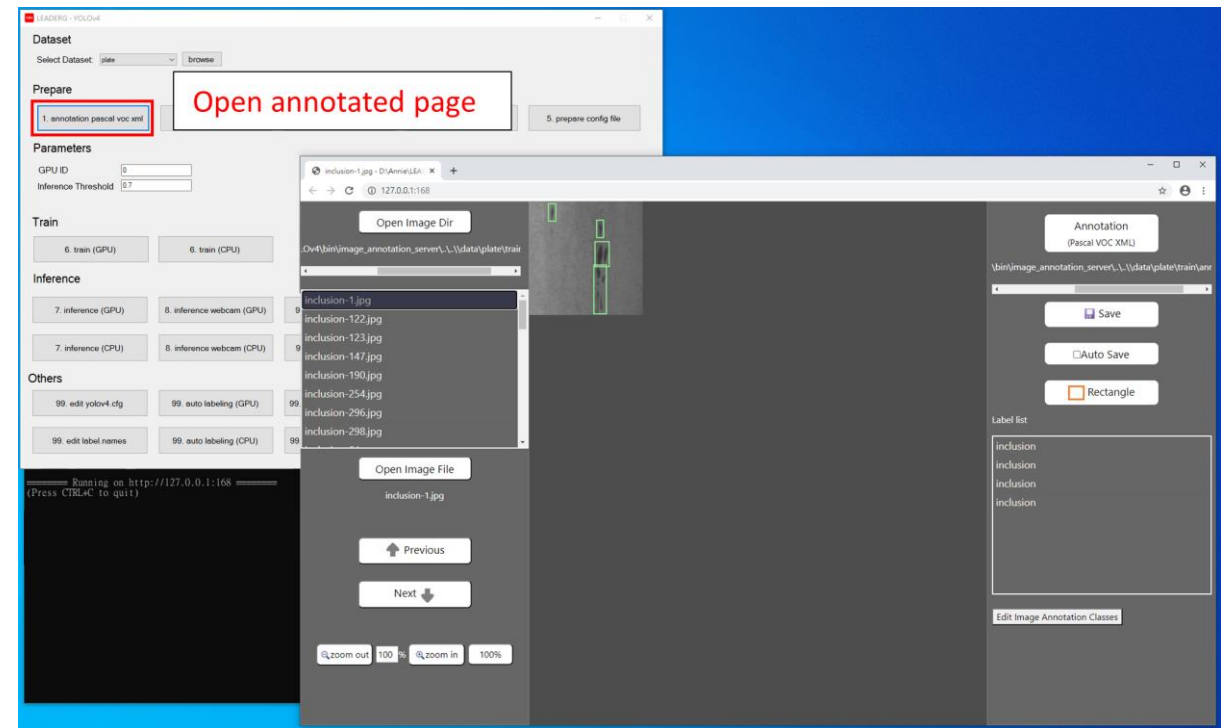
- After pressing the "View" button, click the name of the dataset to be used, and put the images you want to train and infer in the images folder of the train, val, and test folders. The images in the train folder must be at least 6.
- Please add "name to be marked-XXX file name" in front of the training, validation and test image files you put in, such as "tumor-1.png".
- It is recommended to zoom or crop the training and validation image files to a square image of approximately 512 x 512 size. The zoomed and cropped part of the image can be zoomed and cropped on the annotation page opened by "1.annotation pascal voc xml".

# 1. Annotation pascal voc xml

After running, it will open the annotation webpage for image annotation. Currently, only image formats with file extension .png is supported, and images can be cropped, resized, and labeled.

Note:

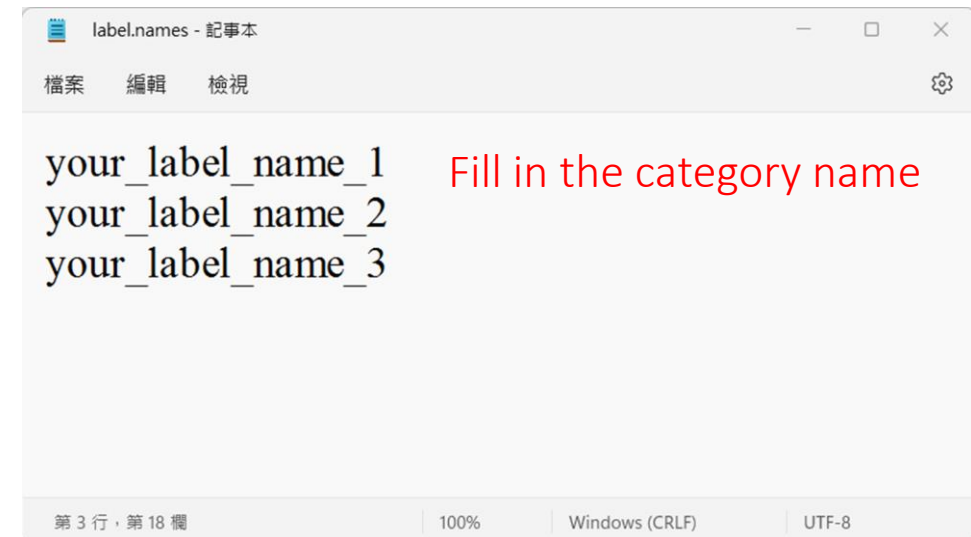
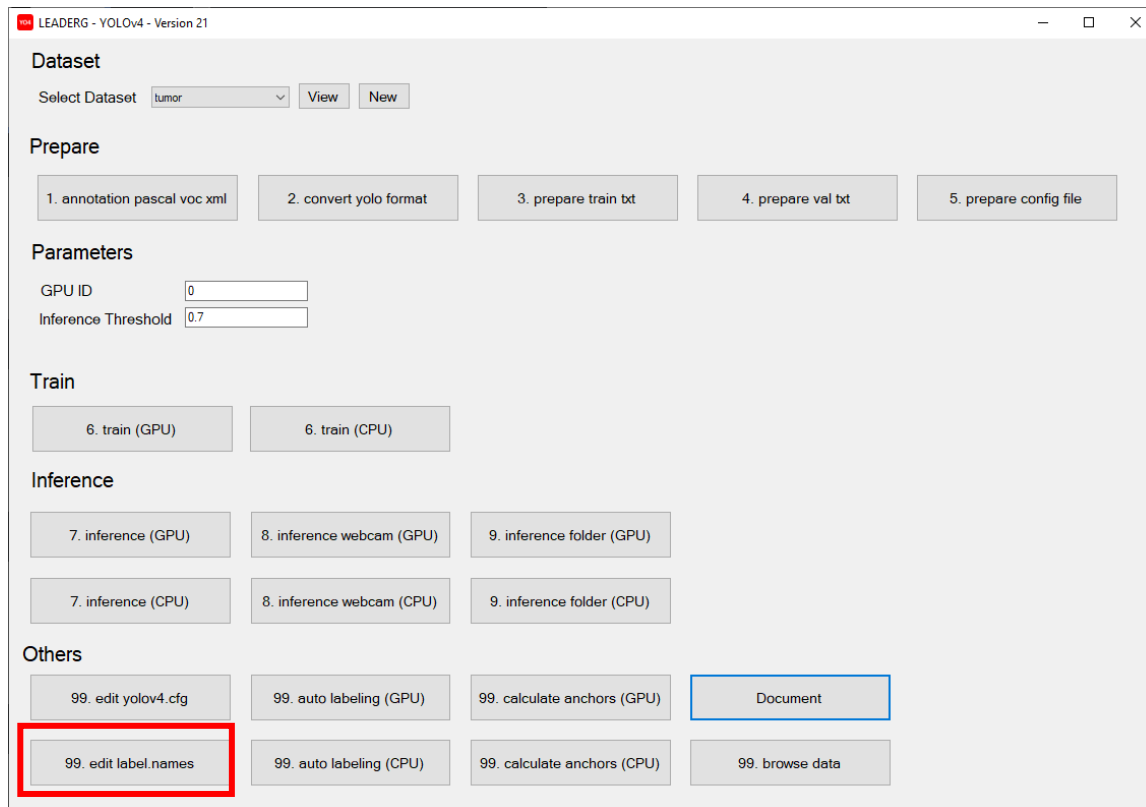
Both training and validation images need to be annotated. To switch to annotated validation images, please press "Open Image Dir" at the top left to select "your own dataset/val/images", then press "SELECT", then press Under "Annotation" in the upper right, select "Pascal VOC XML" and then select "own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



See Annotation.pdf for how to use annotation pages.

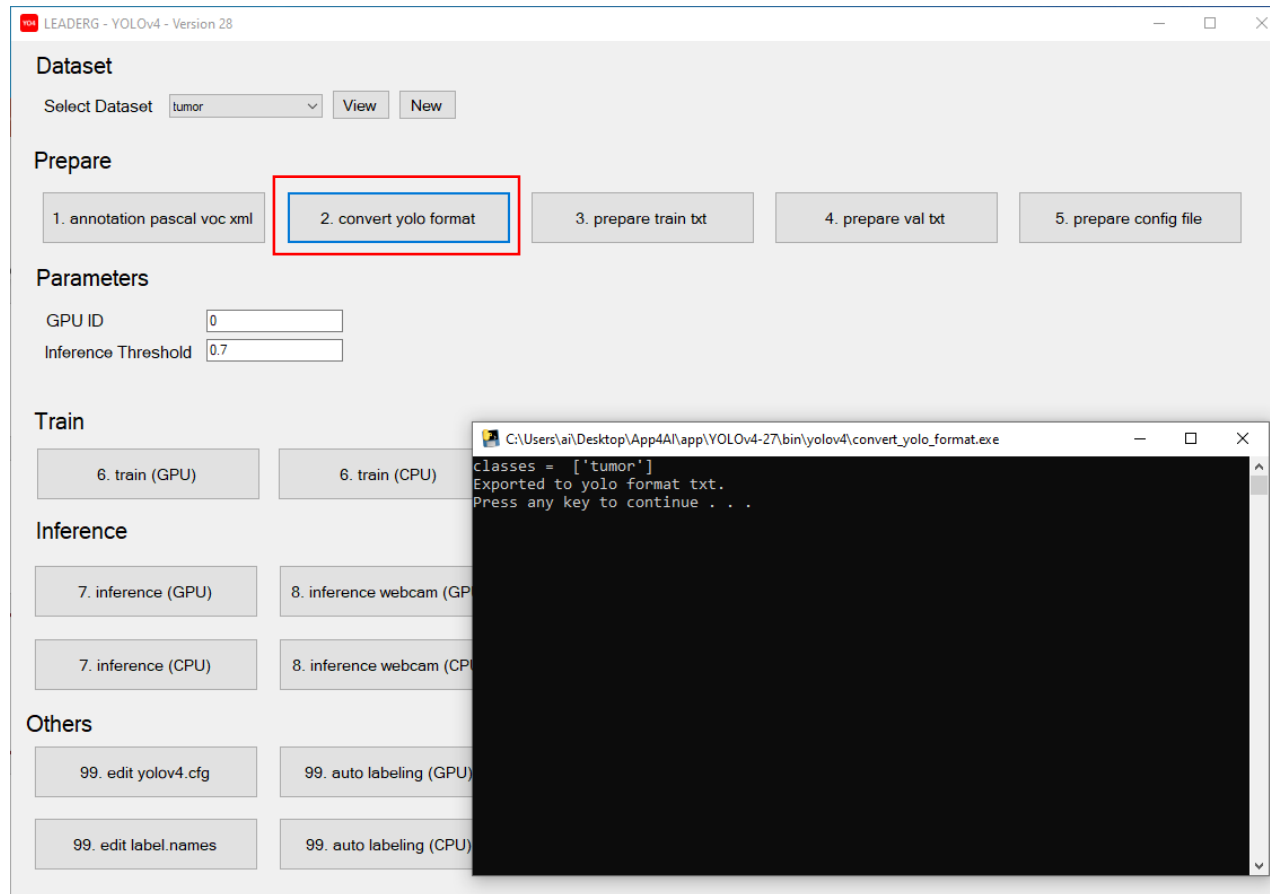
# 99.edit label.names

Press [99.edit label.names] to change to the same class names used for labeling. The format is to distinguish multiple categories by line breaks.



## 2.convert yolo format

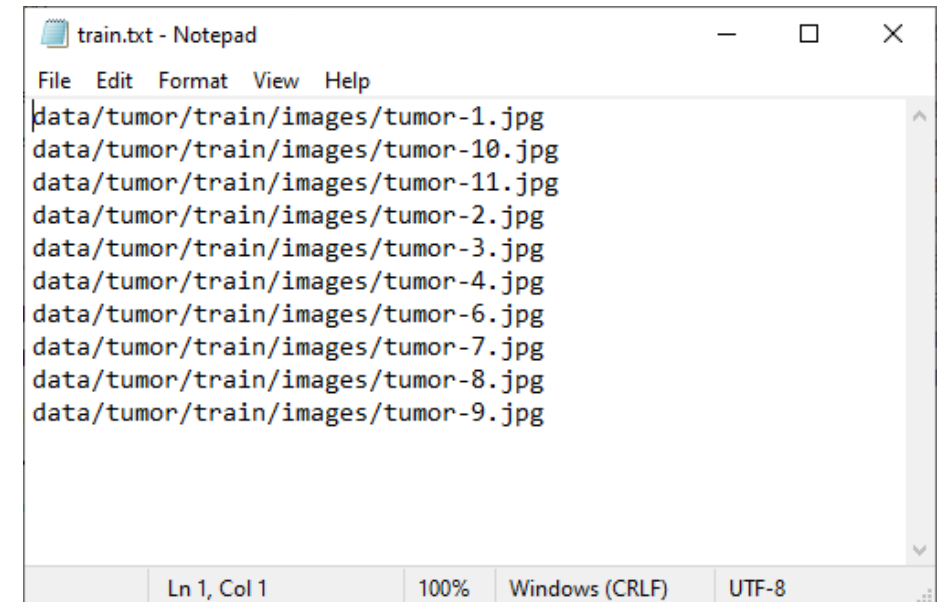
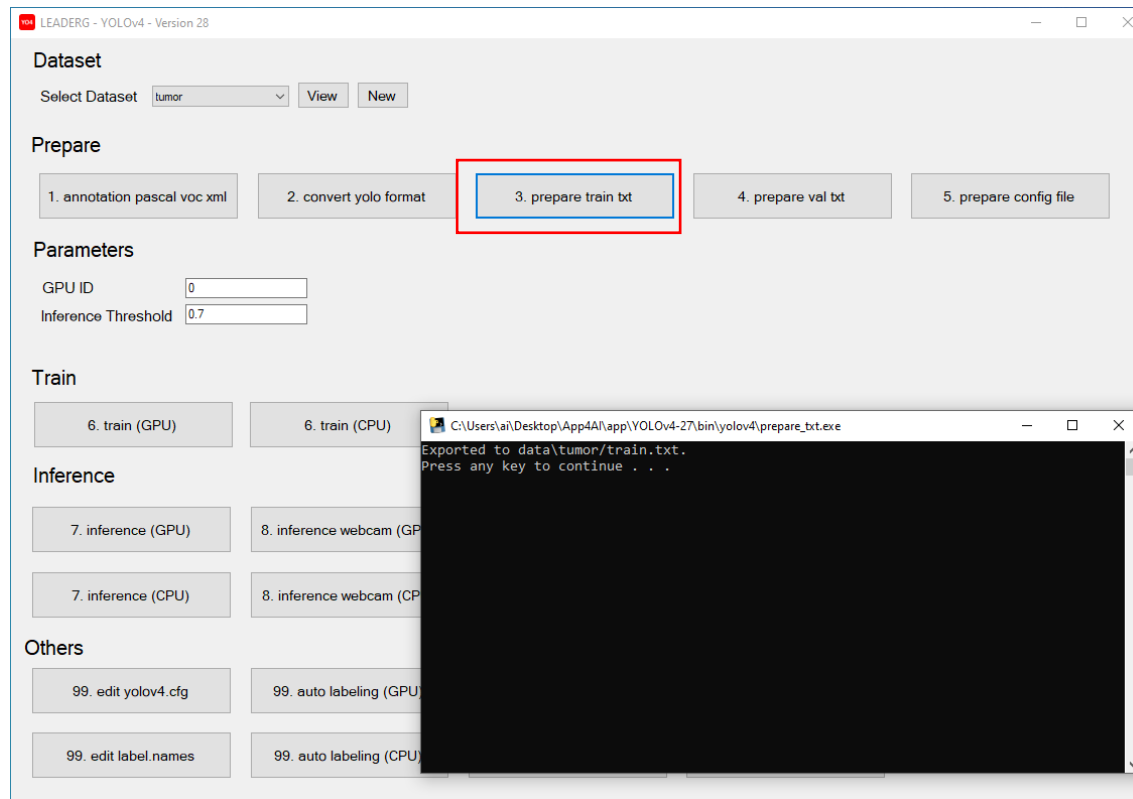
Convert the voc xml annotation file to yolo format.





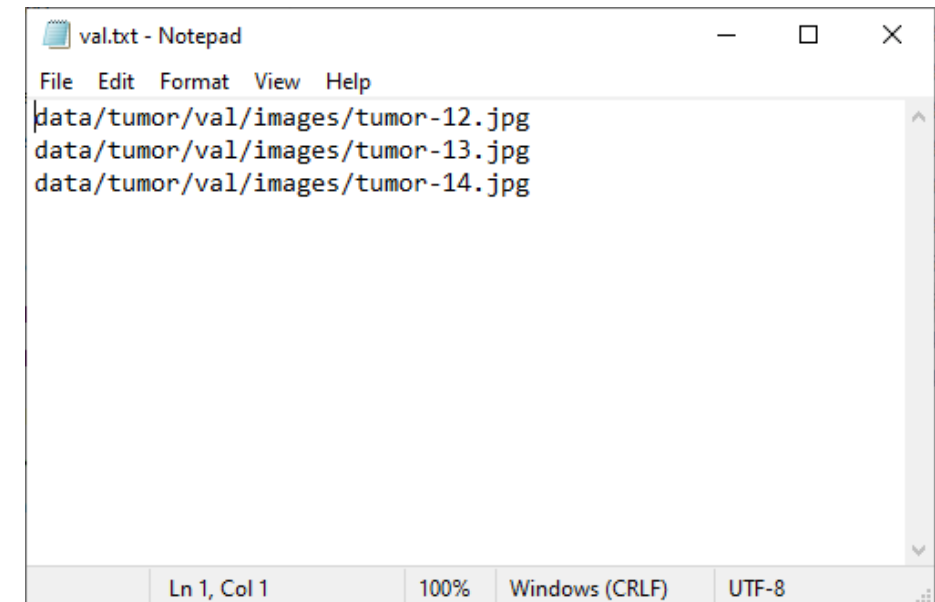
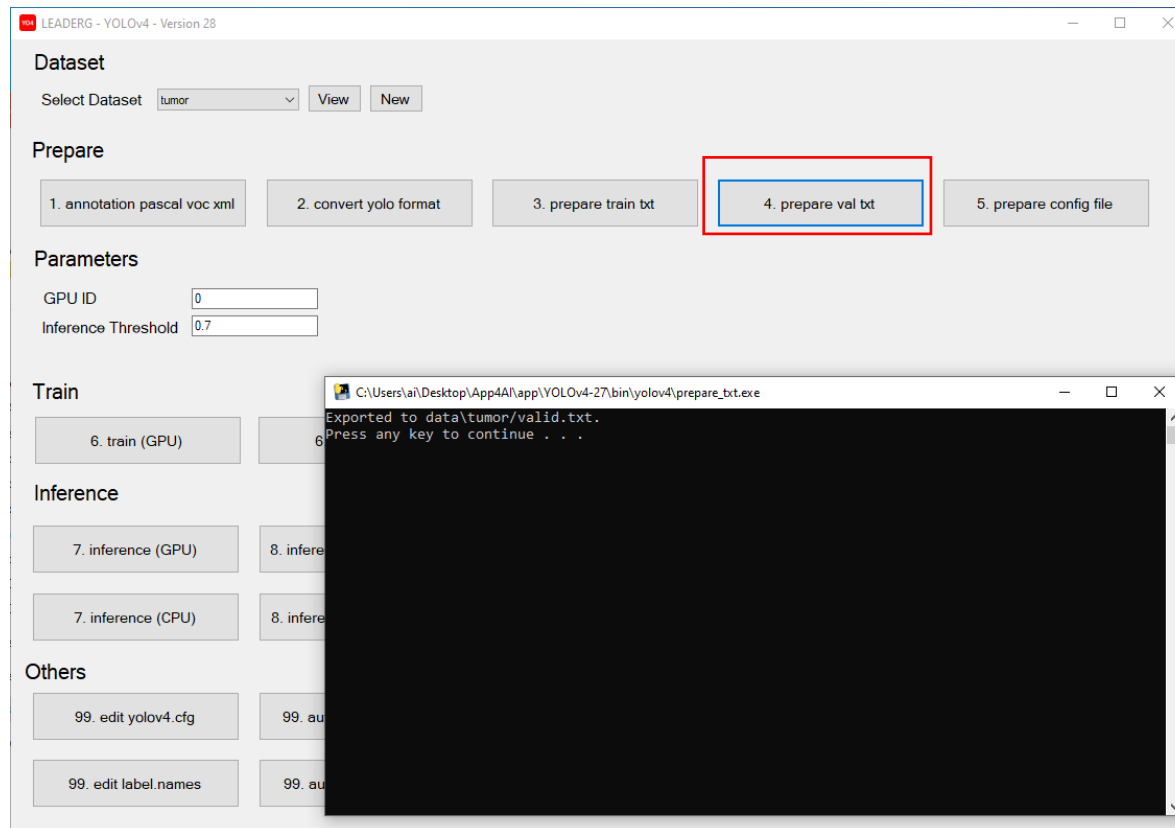
# 3.prepare train txt

Generate training image path files “train.txt”.



# 4.prepare val txt

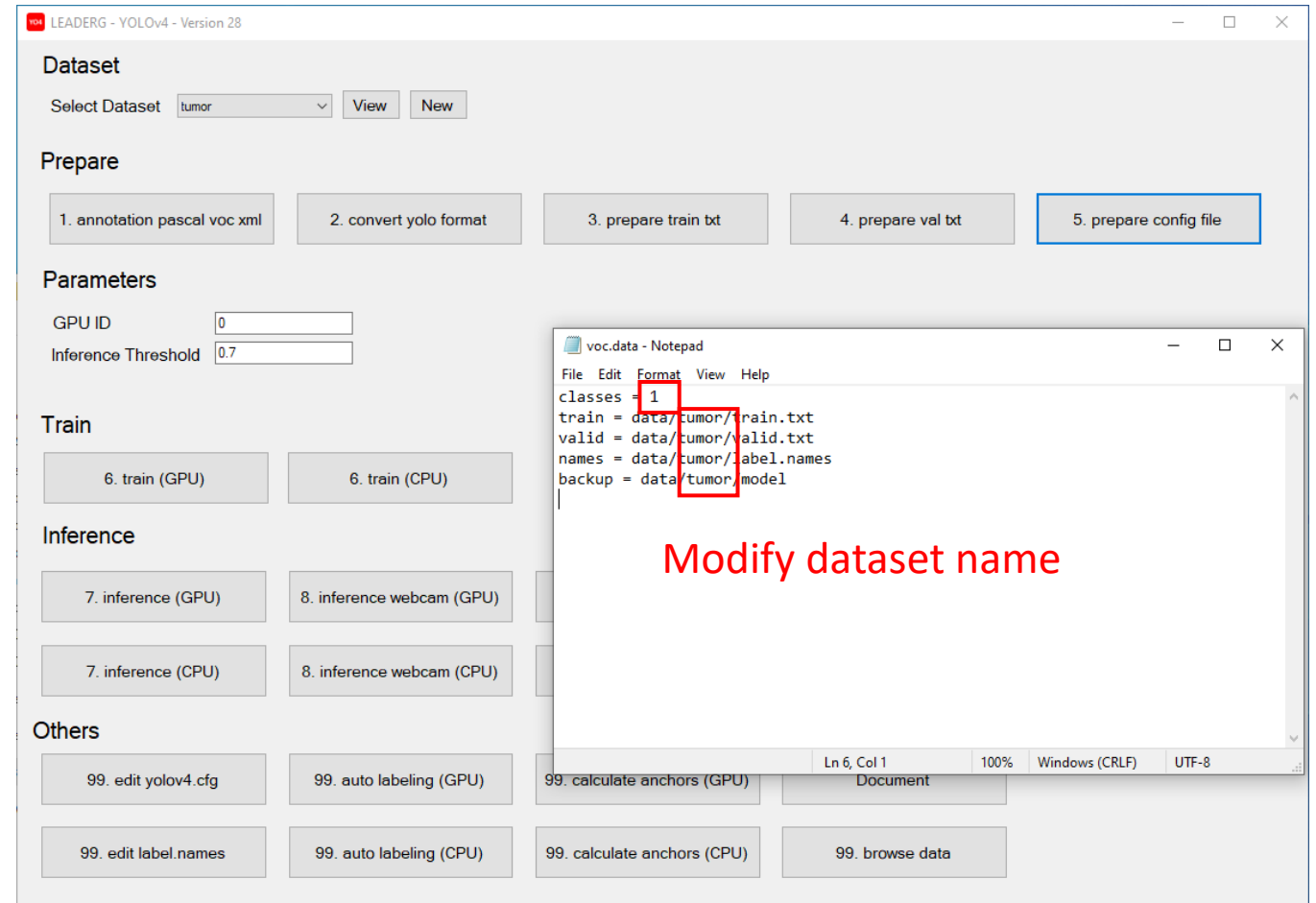
Generate validation image path files “val.txt”.



# 5.prepare config file

Open voc.data for editing. If there is an increase or decrease in categories , please be sure to modify the number of classes. If there are 5 categories, fill in 5 (background does not count), and so on.

In addition to modifying the category, if the dataset name of train, valid, names, backup is different from the dataset name displayed in "Select Dataset", you also need to modify the dataset name of voc.data.



# Set training yolov4.cfg related parameters

Press [99.edit yolov4.cfg] to set the parameters related to the number of categories in the dataset yolov4.cfg file.

The image shows three Notepad windows displaying the configuration file `yolov4.cfg`. The windows are arranged to show the process of editing parameters related to the number of categories in the dataset.

**Top Window (Initial State):**

```
File Edit Format View Help
pad=1
#filters=(classes+5)x3
filters=18
activation=linear

[yolo]
mask=0,1,2
anchors=12, 16, 19, 36, 40, 28, 36, 75, 76, 55, 72, 146, 142, 110, 192, 243, 459, 401
classes=1
num=9
jitter=.3
```

**Bottom-Left Window (Intermediate State):**

```
File Edit Format View Help
pad=1
#filters=(classes+5)x3
filters=18#255
activation=linear

[yolo]
mask=3,4,5
anchors=12, 16, 19, 36, 40, 28, 36, 75, 76, 55, 72, 146, 142, 110, 192, 243, 459, 401
classes=1
num=9
jitter=.3
```

**Bottom-Right Window (Final State):**

```
File Edit Format View Help
stride=1
pad=1
#filters=(classes+5)x3
filters=18#255
activation=linear

[yolo]
mask=6,7,8
anchors=12, 16, 19, 36, 40, 28, 36, 75, 76, 55, 72, 146, 142, 110, 192, 243, 459, 401
classes=1
num=9
```

**Annotations:**

- Red Text:** "Set to (number of categories + 5) x 3 example: Suppose there is a label to train So  $(1 + 5) \times 3 = 18$ ". Red arrows point from this text to the `filters=18` line in the top and bottom-left windows.
- Blue Text:** "Set the number of categories example: Suppose there is a label to train So fill in 1". Blue arrows point from this text to the `classes=1` line in the top, bottom-left, and bottom-right windows.

# Set yolov4.cfg anchor parameters

Press [99.calculate anchors] to calculate the anchor point value suitable for your dataset. Before running, please confirm [99.edit label.names] and whether the category filled in the content is correct.

Then press [99.edit yolov4.cfg] to set the yolov4.cfg file in the dataset, and fill in the value displayed in the red box into the yolov4.cfg anchor parameter.

The image shows a terminal window and two Notepad windows illustrating the process of setting anchor parameters for yolov4.cfg.

**Terminal Window:**

```
C:\Users\ai\Desktop\App4AI-2225\sdk\Jupyter-Image-Object-Detection-YOLOv4-CPP-14\src\darknet\build\darknet\x64\darknet.exe
CUDA-version: 11030 (11060), cuDNN: 8.2.1, CUDNN_HALF=1, GPU count: 1
CUDNN_HALF=1
OpenCV version: 4.5.5

num_of_clusters = 9, width = 512, height = 512
read labels from 10 images
loaded          image: 10      box: 11
all loaded.

calculating k-means++ ...

iterations = 1

counters_per_class = 11

avg IoU = 97.03 %

Saving anchors to the file: anchors.txt
anchors = 40, 30, 51, 57, 75, 45, 67, 81, 91, 86, 83,117, 120,115, 128,152, 159,157
```

**yolov4.cfg - Notepad (Top):**

```
File Edit Format View Help
pad=1
#filters =(classes + 5)x3
filters=18
activation=linear

[yolo]
mask = 0,1,2
anchors = 12, 16, 19, 36, 40, 28, 36, 75, 76, 55, 72, 146, 142, 110, 192, 243, 459, 401
classes=1
num=9
jitter=.3
```

**yolov4.cfg - Notepad (Bottom):**

```
File Edit Format View Help
stride=1
pad=1
#filters =(classes + 5)x3
filters=18#255
activation=linear

[yolo]
mask = 6,7,8
anchors = 12, 16, 19, 36, 40, 28, 36, 75, 76, 55, 72, 146, 142, 110, 192, 243, 459, 401
classes=1
num=9
jitter=.3
```

Arrows indicate the flow of data from the terminal to the configuration files. The text "Fill in the value displayed in the CMD" points to the anchor values in the bottom Notepad window.

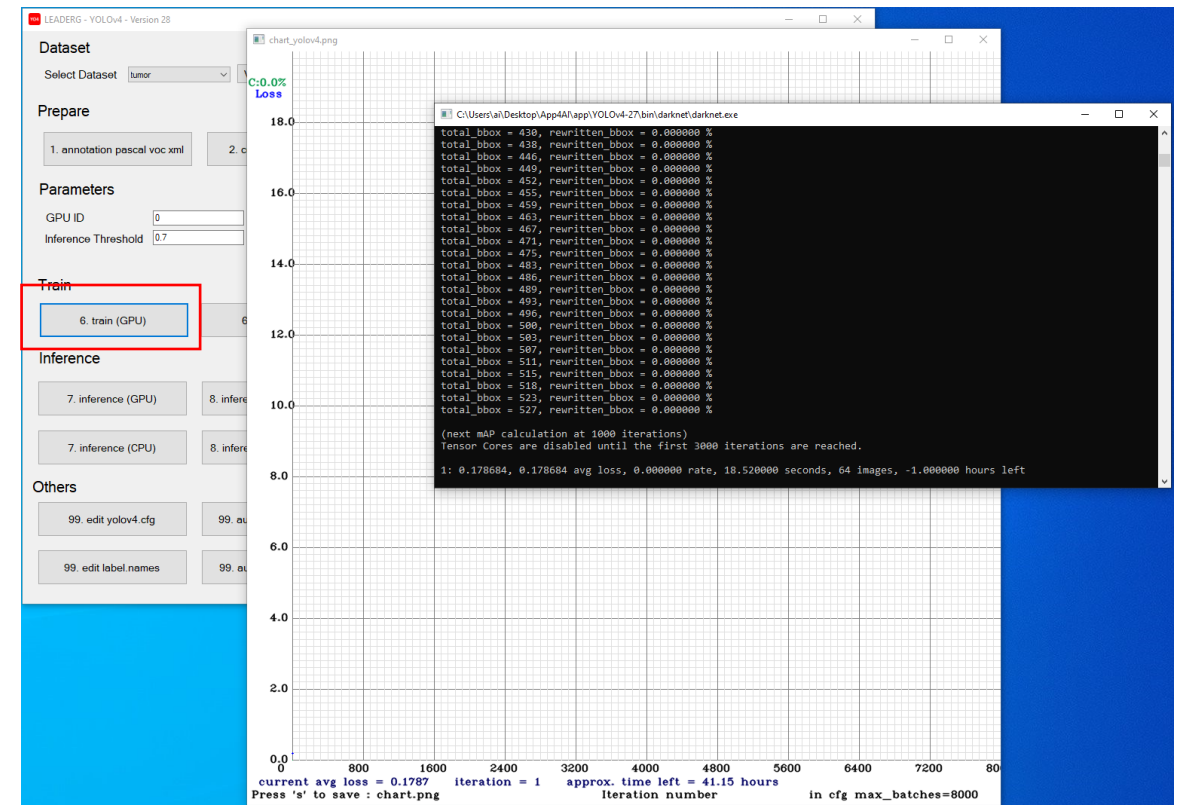
# 6. train (GPU)

Start training.

The training is divided into two types: GPU and CPU. You can choose GPU or CPU mode according to your own hardware.

Note:

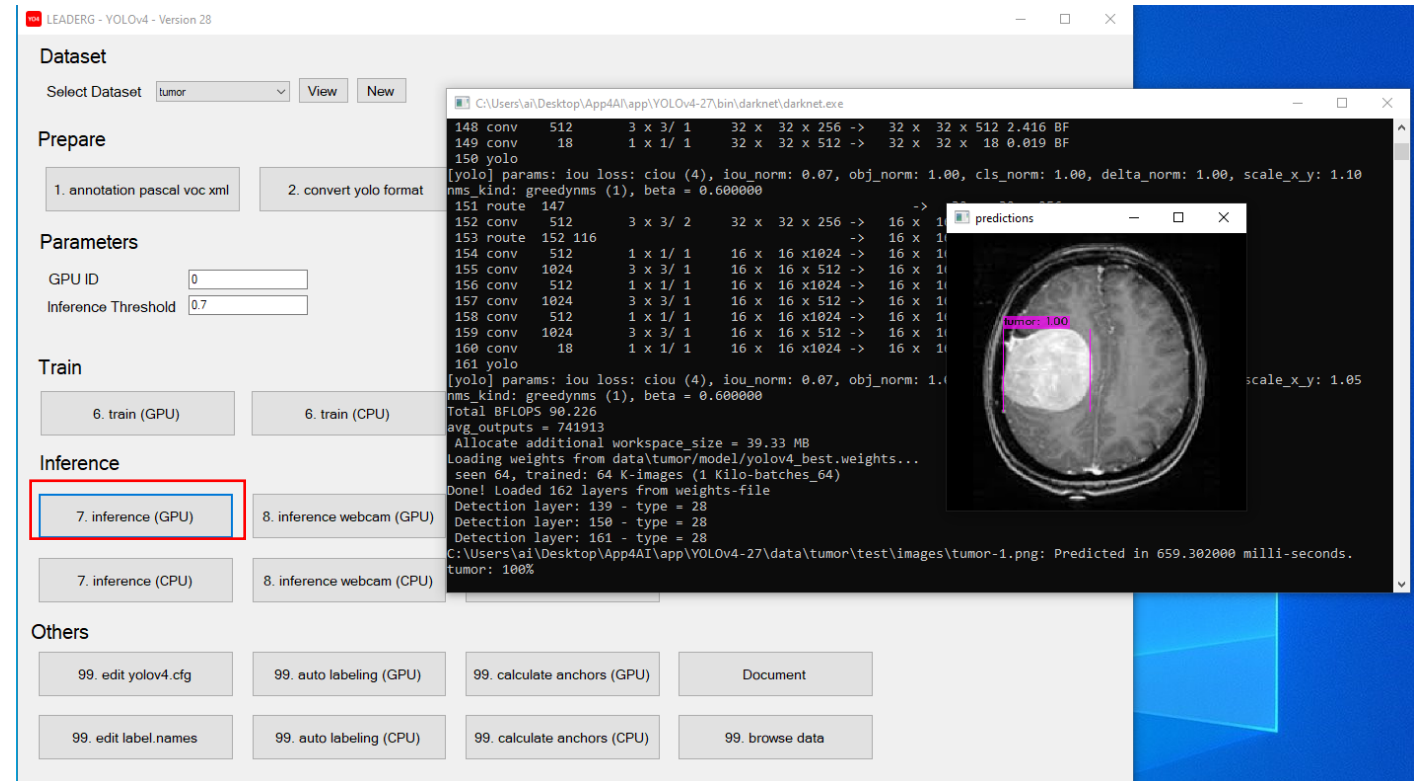
If the loss is nan when training again, it may be due to any problems or setting errors in the previous training, resulting in an abnormal pretrained model. Please go to data-template/model/yolov4\_best.weights and copy it to your own dataset/model folder, and then press [6. train (GPU)] or [6. train (CPU)] to retrain.



# 7. inference (GPU)

Infer a single image.

You can choose GPU or CPU mode according to your own hardware.

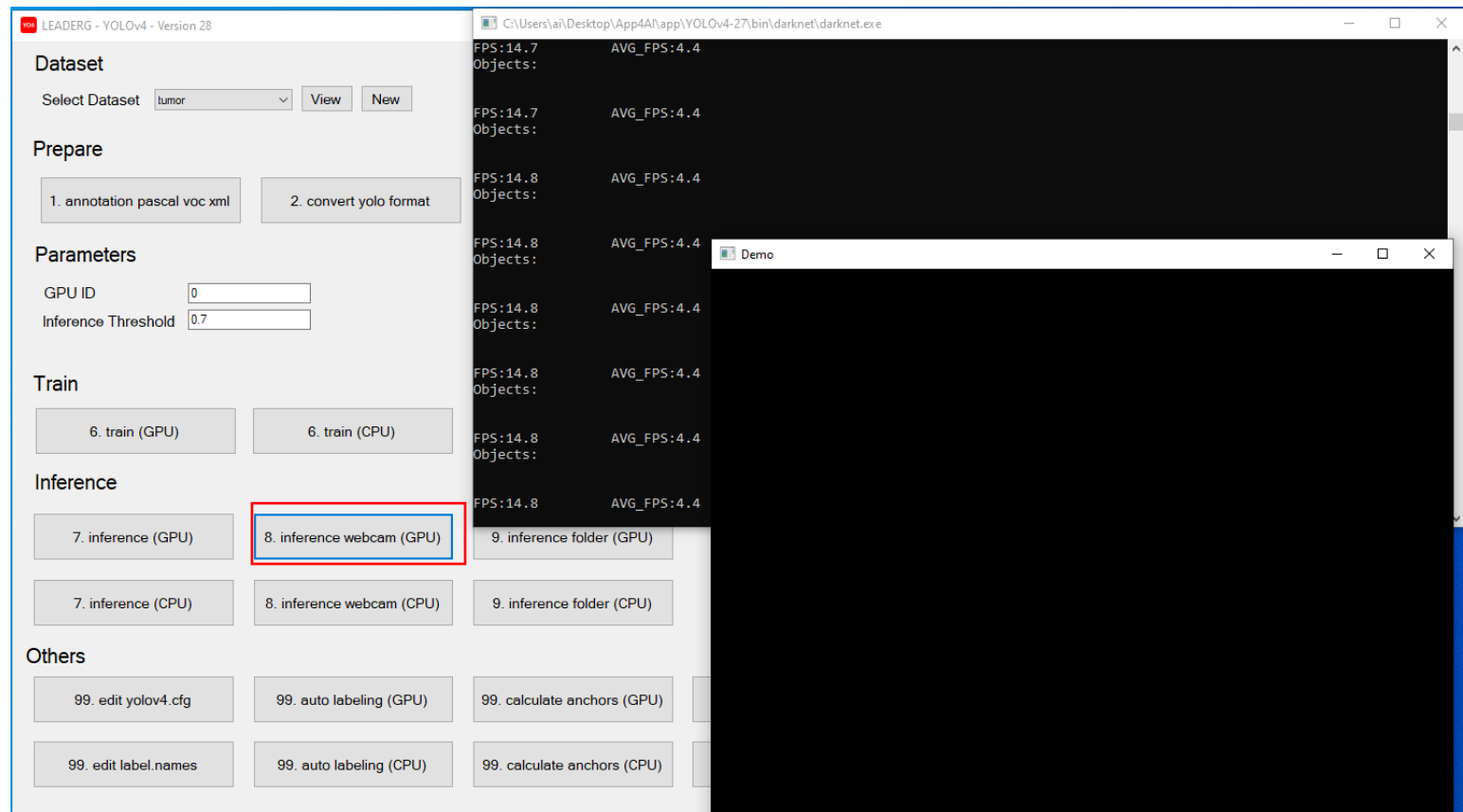


## 8. inference webcam (GPU)

Infer the image of the webcam.

You can choose GPU or CPU mode according to your own hardware.

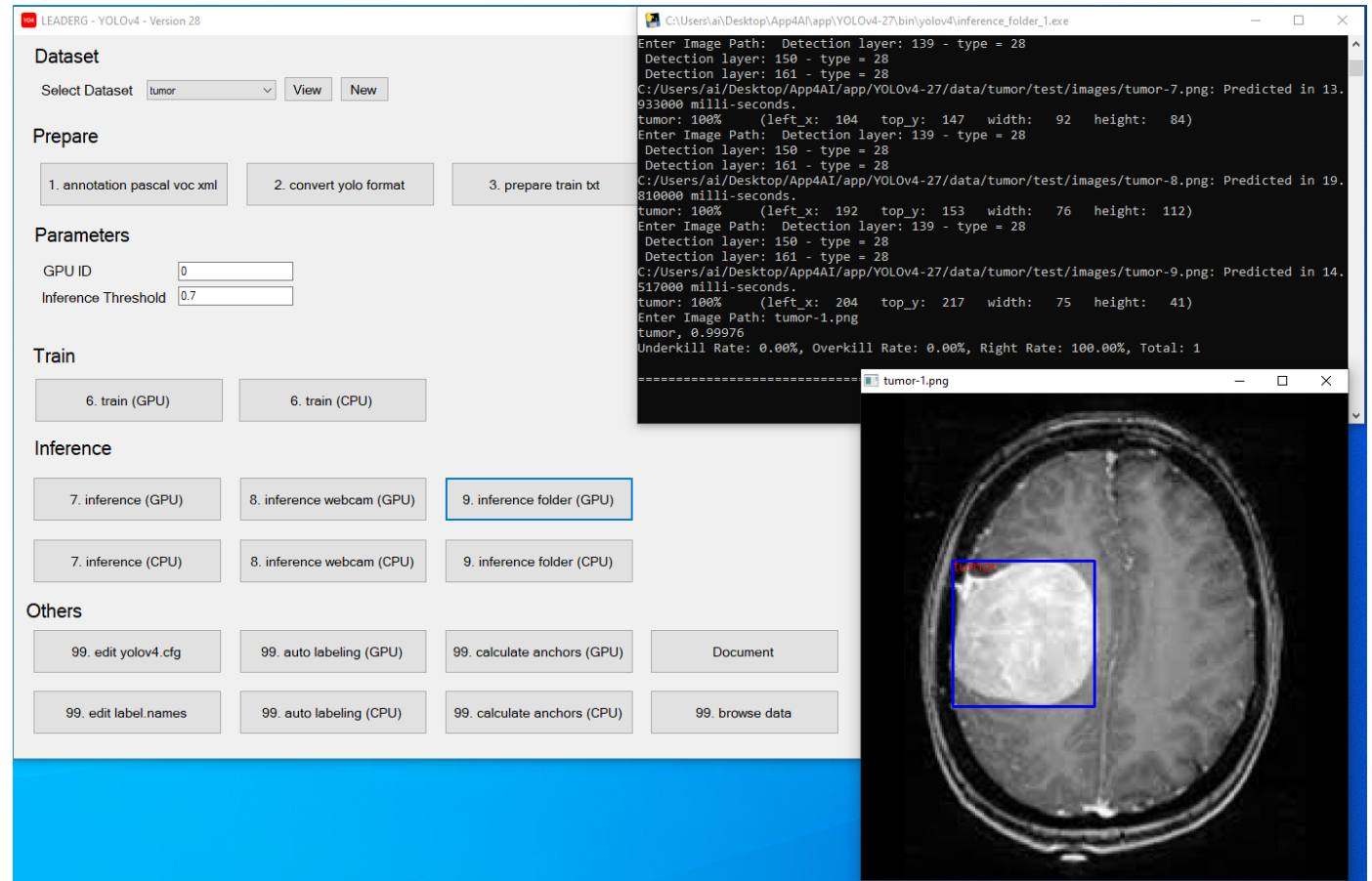
Press the “Esc” key on the display to turn the webcam off.





# 9. inference folder (GPU)

Infer all images in the folder.  
You can choose GPU or CPU mode according to your own hardware.



# Others Button

[99.edit yolov4.cfg] edit yolov4.cfg file.

[99.edit label.names] edit the name of the label category, which must be the same as the name and quantity used in the label page opened by [1. Annotation pascal voc xml].

[99.auto labeling] Automatically label the images in the auto\_labeling folder, and output the label file to the auto\_labeling folder.

[99.calculate anchors] use the training and validation datasets to calculate anchors suitable for your own dataset.

[Document] can open documentation or video webpage.

[99.browse data] can open the folder location.

# 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.