

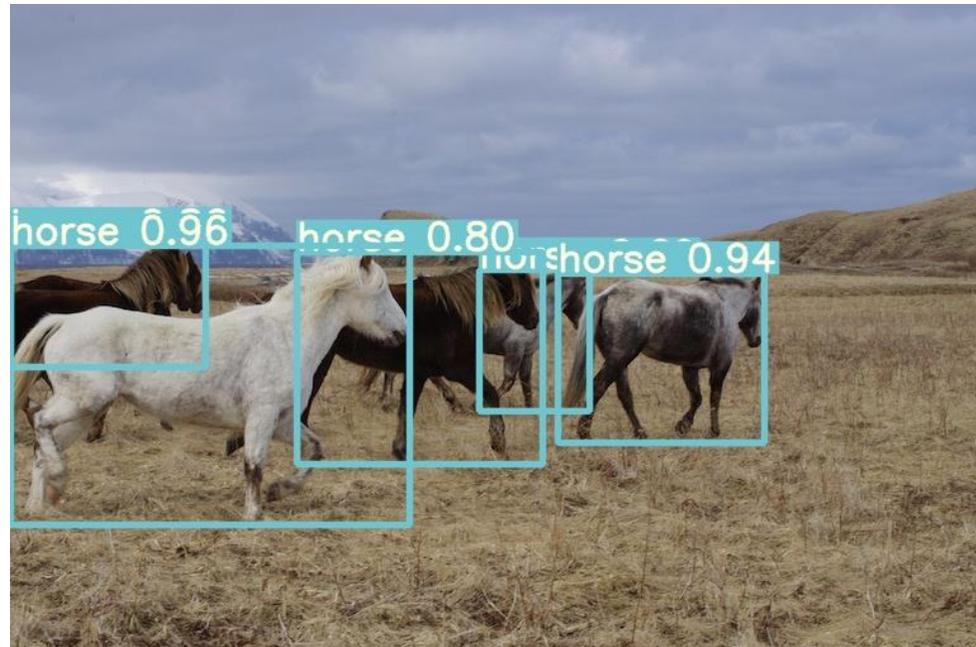
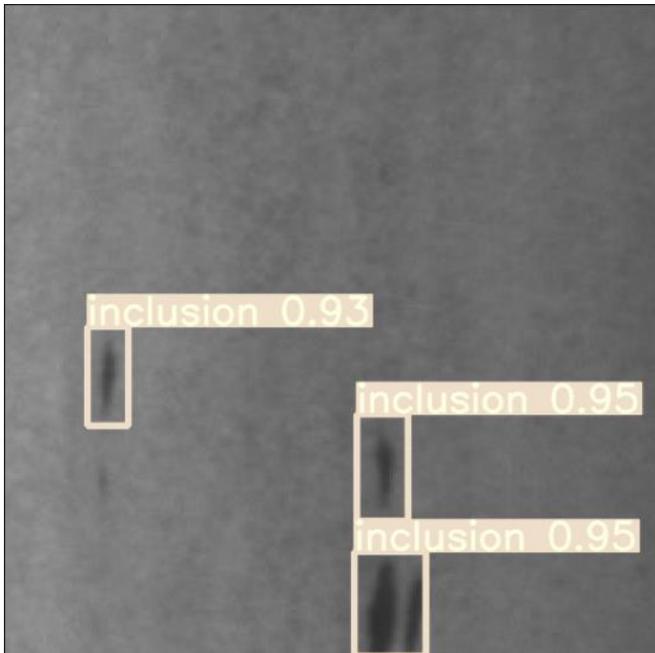
YOLOv7

The most powerful object detection algorithm in the year 2022.

Version 20230223

Applications

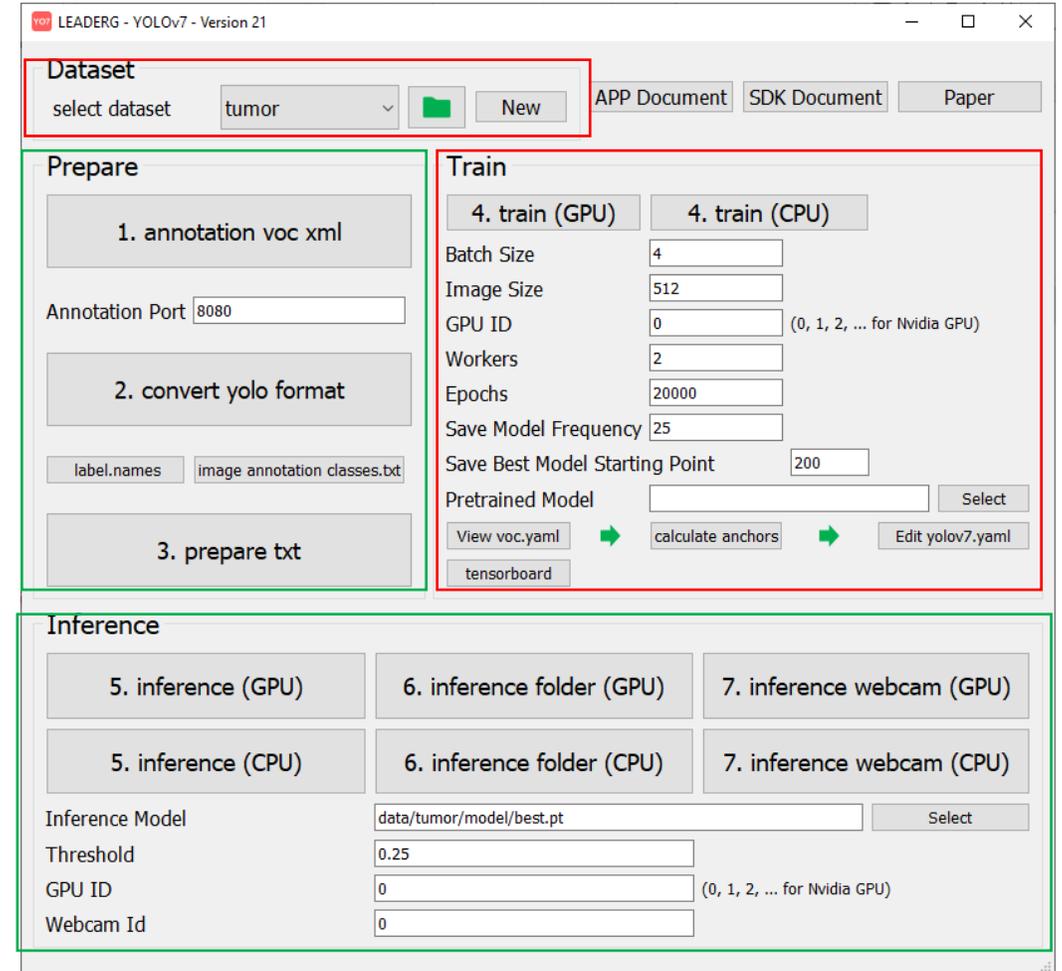
YOLOv7 can be applied to factory defect detection, medical image analysis, biological image analysis, industrial security image analysis, mask image analysis, etc.



How to use

The main process is:

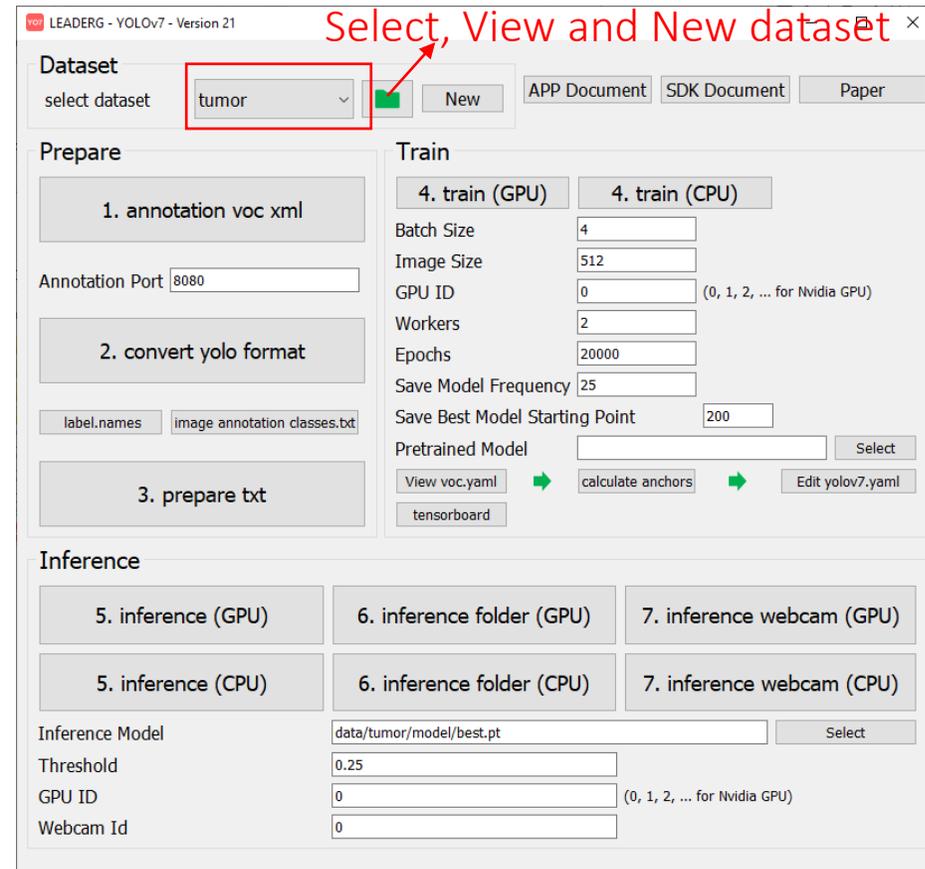
Select dataset -> preprocessing (prepare images, labels, set training parameters) -> training -> inference images



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. The images in the train folder must be at least 9.
- Please add "name to be marked-XXX file name" in front of the training, validation and test image files you put in, such as "inclusion-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 voc xml".

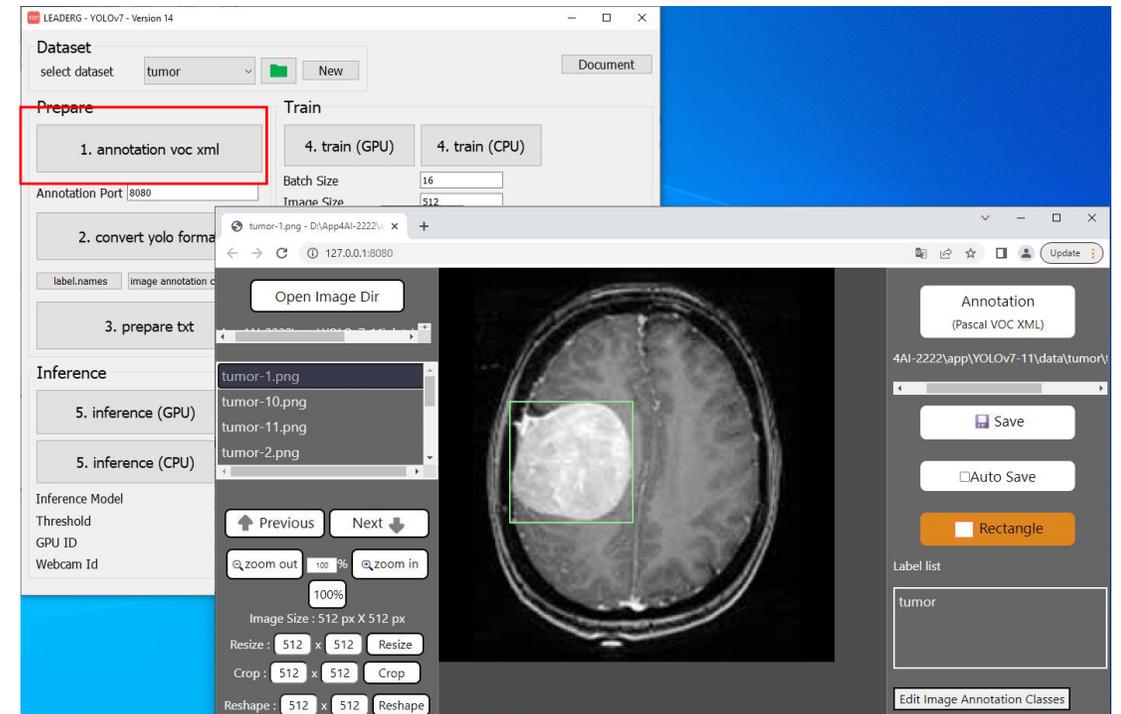
1. Annotation voc xml

After running, it will open the annotation webpage for image annotation. Currently, only image formats with file extension .jpeg or .jpg or .png are 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.

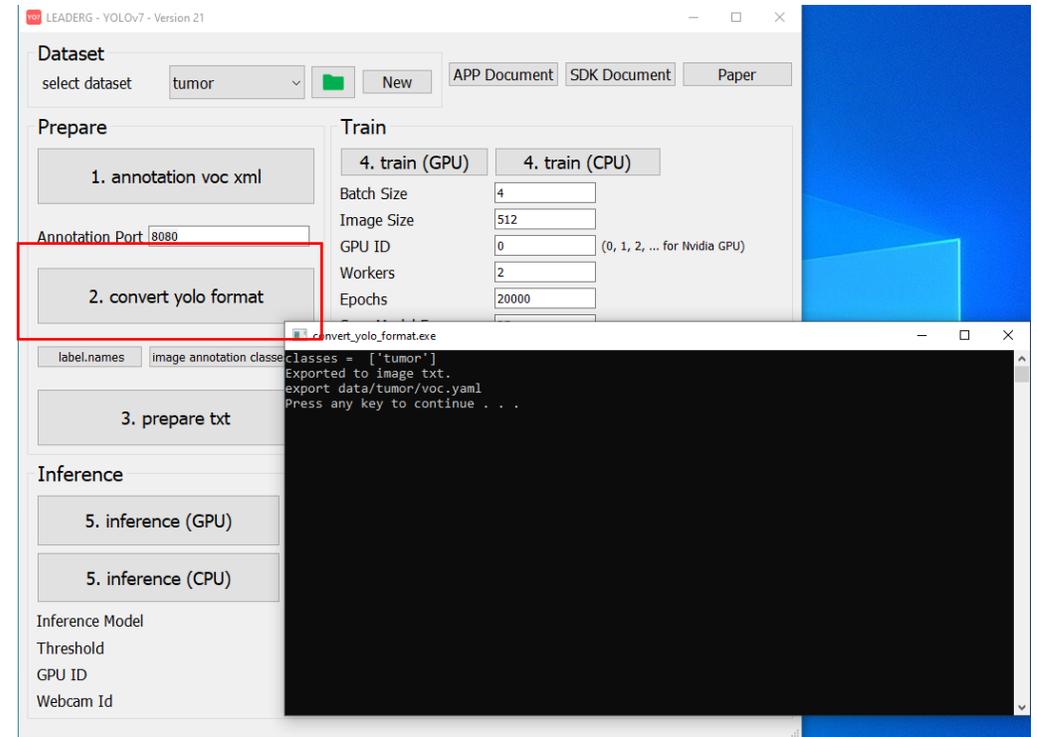


2.convert yolo format

After confirming that the category names in the label.names file are correct, you can press the button to convert the voc xml tag 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 training image path files “train.txt” and “val.txt”.

The screenshot displays the LEADERG - YOLOv7 - Version 17 interface. The 'Dataset' section is set to 'tumor'. The 'Prepare' section has '3. prepare txt' highlighted with a red box. The 'Train' section shows '4. train (GPU)' and '4. train (CPU)' buttons. A terminal window shows the command 'prepare_txt.exe' and its output: 'Exported to train.txt.', 'Exported to val.txt.', and 'Press any key to continue . . .'. Two Notepad windows show the contents of 'train.txt' and 'val.txt'.

```
train.txt - Notepad
File Edit Format View Help
data/plate/train/images/inclusion-1.jpg
data/plate/train/images/inclusion-122.jpg
data/plate/train/images/inclusion-123.jpg
data/plate/train/images/inclusion-147.jpg
data/plate/train/images/inclusion-190.jpg
data/plate/train/images/inclusion-254.jpg
data/plate/train/images/inclusion-296.jpg
data/plate/train/images/inclusion-298.jpg
data/plate/train/images/inclusion-6.jpg
data/plate/train/images/inclusion-8.jpg
data/plate/train/images/patches-103.jpg
data/plate/train/images/patches-116.jpg
data/plate/train/images/patches-166.jpg
data/plate/train/images/patches-197.jpg
data/plate/train/images/patches-221.jpg
data/plate/train/images/patches-237.jpg
data/plate/train/images/patches-287.jpg
data/plate/train/images/patches-43.jpg

val.txt - Notepad
File Edit Format View Help
data/plate/val/images/inclusion-144.jpg
data/plate/val/images/inclusion-2.jpg
data/plate/val/images/patches-1.jpg
data/plate/val/images/patches-20.jpg
data/plate/val/images/rolled_in_scale-178.jpg
data/plate/val/images/rolled_in_scale-80.jpg
data/plate/val/images/scratches-34.jpg
data/plate/val/images/scratches-38.jpg

Ln 1, Col 1 100% Windows (CRLF) UTF-8
```

Set yolov7.yaml parameters

Press [calculate anchors] to calculate the anchor point value suitable for your dataset. Before running, please confirm [label.names] and whether the category filled in the content is correct. Then press [edit yolov7.yaml] to set the number of categories and anchor points. The number of categories must be the same as voc.yaml. The value of the anchor point is a group of six numbers. Divide into three groups and fill in the 18 numbers in the red box into the anchors in parentheses.

The screenshot shows the YOLOv7 training interface with a Notepad window open to the `yolov7.yaml` file. The interface includes a 'Dataset' section with 'tumor' selected, a 'Prepare' section with steps 1-3, and a 'Train' section with '4. train (GPU)' selected. The 'Train' section has fields for Batch Size (4), Image Size (512), GPU ID (0), Workers (2), Epochs (20000), Save Model Frequency (25), and Save Best Model Starting Point (200). There are buttons for 'View voc.yaml', 'calculate anchors', and 'Edit yolov7.yaml'. The Notepad window shows the following configuration:

```
# parameters
nc: 4 # number of classes
depth_multiple: 1.0 # model depth multiple
width_multiple: 1.0 # layer channel multiple

# anchors
anchors:
  - [12,16, 19,36, 40,28] # P3/8
  - [36,75, 76,55, 72,146] # P4/16
  - [142,110, 192,243, 459,401] # P5/32

# yolov7 backbone
backbone:
  # [from, number, module, args]
  [[-1, 1, Conv, [32, 3, 1]], # 0

  [-1, 1, Conv, [64, 3, 2]], # 1-P1/2
  [-1, 1, Conv, [64, 3, 1]],
```

Annotations in the image include a blue arrow pointing to the `nc: 4` line with the text 'Fill in the number of categories', and another blue arrow pointing to the `anchors` list with the text 'Fill in the value displayed by CMD'. A red box highlights the list of anchor points in the terminal output: `anchors - 40,30, 51,57, 75,45, 67,80, 91,86, 83,117, 120,114, 128,152, 159,157`.

4. train (GPU)

4. train (CPU)

Start training.

Note:

Pretrained Model is the pretrained model path used, None means not to use the pretrained model for training.

Epochs is the number of training epochs.

Save Model Frequency is the model saves the model every few epochs.

Save Best Model Starting Point is the number of epochs, when the model is the current best model, save the current model.

LEADERG - YOLOv7 - Version 21

Dataset: select dataset: tumor [New] APP Document SDK Document Paper

Prepare: 1. annotation voc xml

Annotation Port: 8080

2. convert yolo format

label.names: image annotation classes.txt

3. prepare txt

Train: 4. train (GPU) 4. train (CPU)

Batch Size: 4

Image Size: 512

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

Workers: 2

Epochs: 20000

Save Model Frequency: 25

Save Best Model Starting Point: 200

Pretrained Model: [Select]

View voc.yaml calculate anchors Edit yolov7.yaml

tensorboard

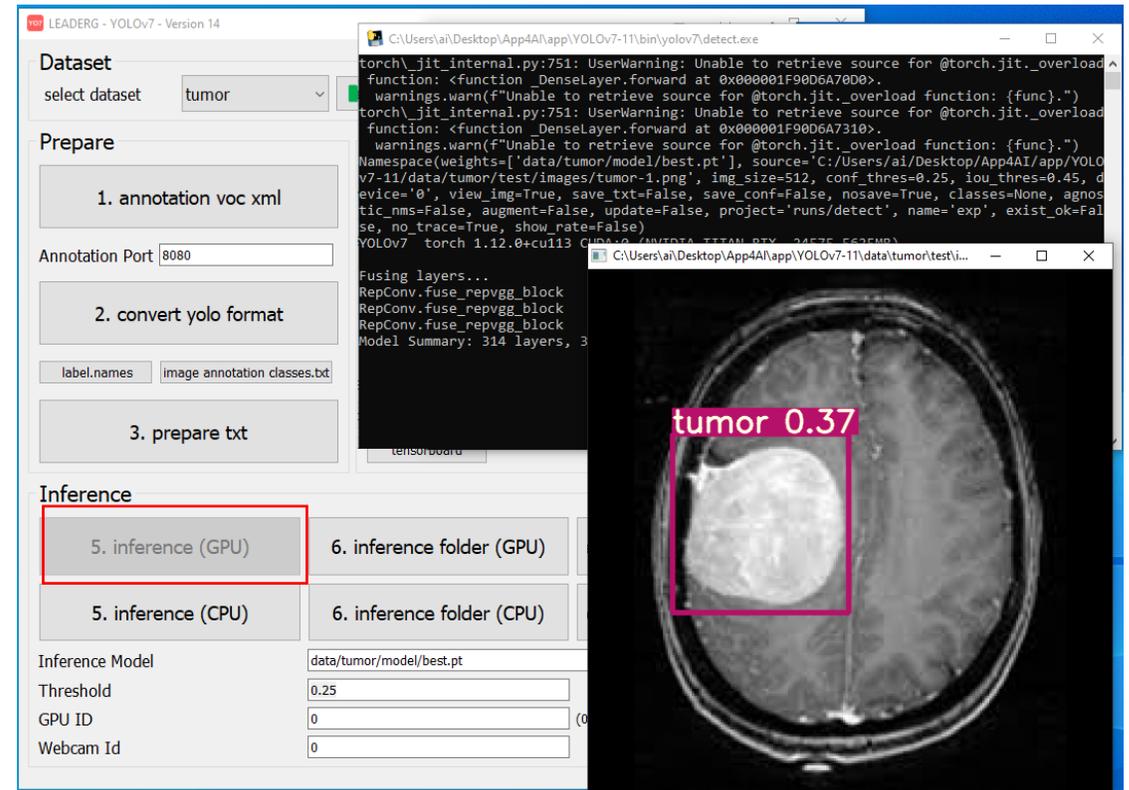
```
C:\Users\...\App44\app\YOLOv7-21\bin\yolov7\train.exe
orcvision\io\image.py:13: UserWarning: Failed to load image Python extension:
torchvision\io\image.py:13: UserWarning: Failed to load image Python extension:
torch\_jit\_internal.py:751: UserWarning: Unable to retrieve source for @torch.jit._overload function: <function _DenseLa
yer.forward at 0x00000153503E24C0>.
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 0x00000153503E2700>.
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 0x000002A30F5C24C0>.
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 0x000002A30F5C2700>.
warnings.warn(F"Unable to retrieve source for @torch.jit._overload function: {func}.")
0/19999 10.9G 0.05163 1.837 0 1.889 4 512: 100%| 3/3 [00:16<00:00, 5.58s/it]
Class all Images 3 Labels 0 P R mAP@.5 mAP@.5:.95: 100%| 1/1 [00:01<00:00,
Epoch gpu_mem box obj cls total labels img_size
1/19999 11G 0.05114 1.827 0 1.878 7 512: 100%| 3/3 [00:00<00:00, 4.48it/s]
Class all Images 3 Labels 0 P R mAP@.5 mAP@.5:.95: 100%| 1/1 [00:00<00:00,
Epoch gpu_mem box obj cls total labels img_size
2/19999 11G 0.0458 1.798 0 1.843 2 512: 100%| 3/3 [00:00<00:00, 5.29it/s]
Class all Images 3 Labels 0 P R mAP@.5 mAP@.5:.95: 100%| 1/1 [00:00<00:00,
Epoch gpu_mem box obj cls total labels img_size
3/19999 11G 0.0405 1.748 0 1.789 3 512: 67%| 2/3 [00:00<00:00, 5.88it/s]
```

5. inference (GPU)

5. inference (CPU)

Infer a single image.

Choose the Inference Model yourself.



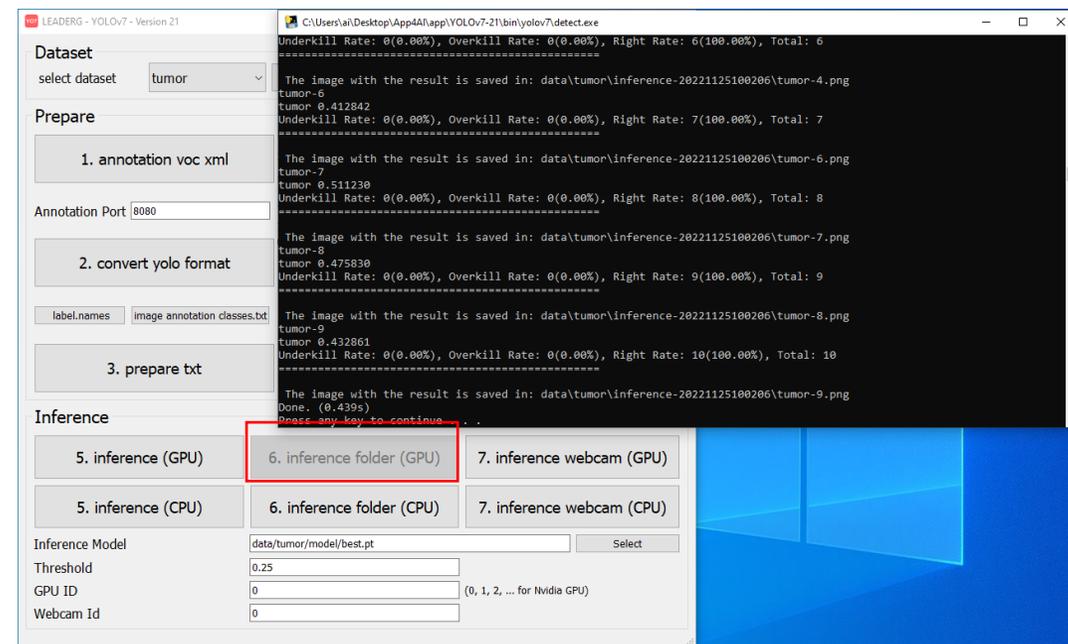
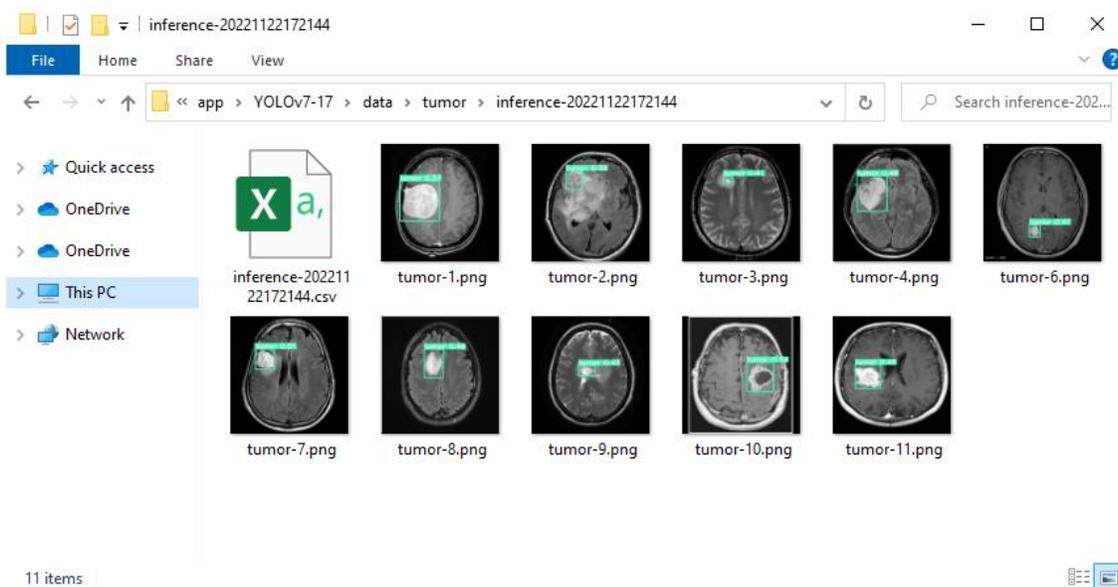
6. inference folder (GPU)

6. inference folder (CPU)

Infer all images in the folder.

Choose the Inference Model yourself.

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



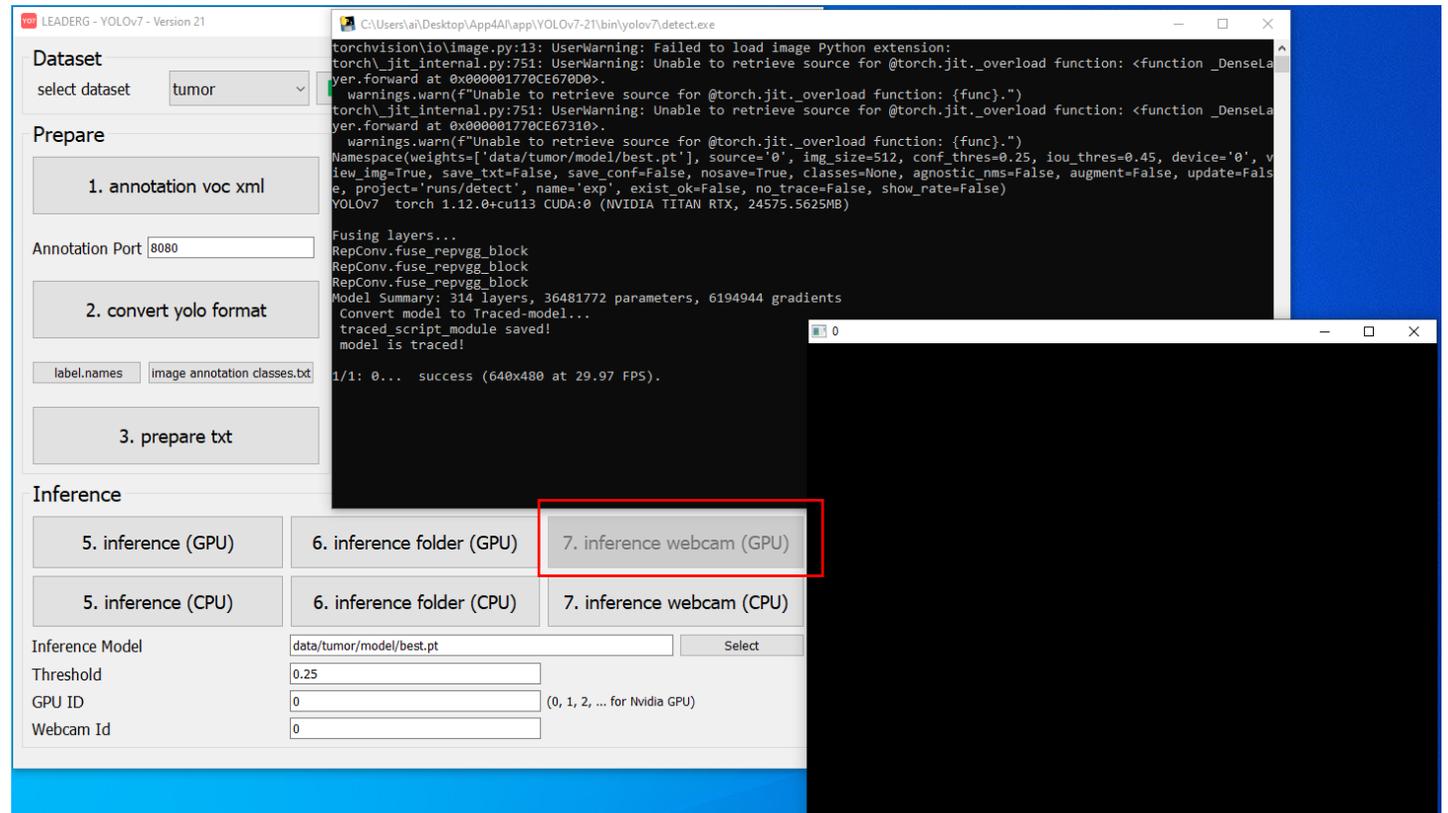
7. inference webcam (GPU)

7. inference webcam (CPU)

Infer the image of the webcam.

The Webcam Id parameter below can set the specified webcam.

Choose the Inference Model yourself.



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.