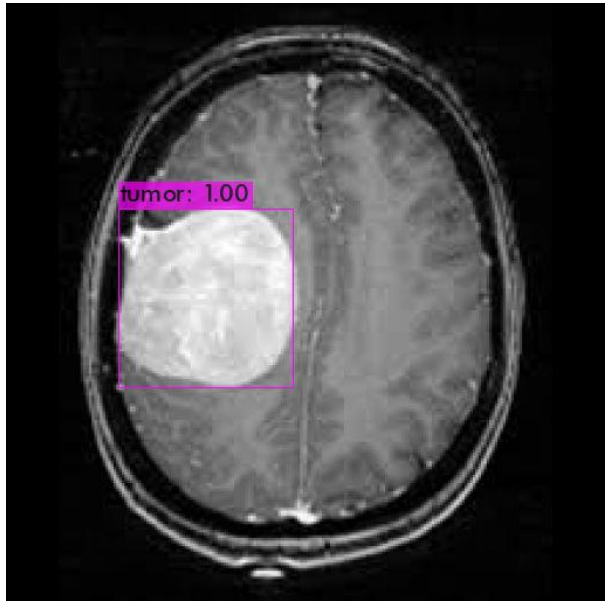
# Image-Object-Detection-YOLOv4-CPP-Jupyter

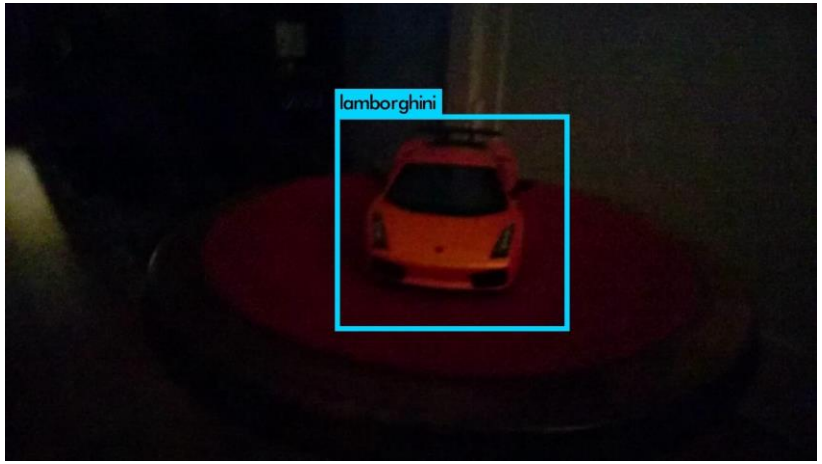
Russian Alexey Bochkovskiy, the maintainer of YOLO Darknet, found that the CSPNet detector developed by Wang Jianyao, the post-doc of the Chinese Academy of Sciences and director Liao Hongyuan, was fast and good, so he invited the Chinese Academy of Sciences to develop YOLOv4 for the backbone, and did 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.

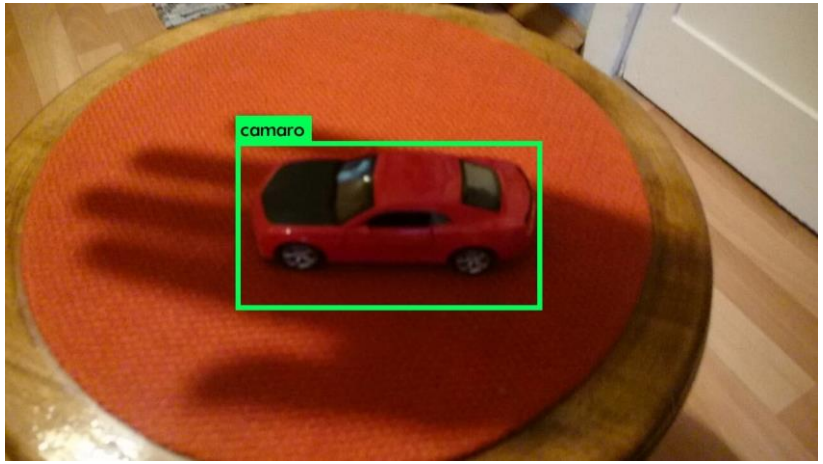
Version 20230223

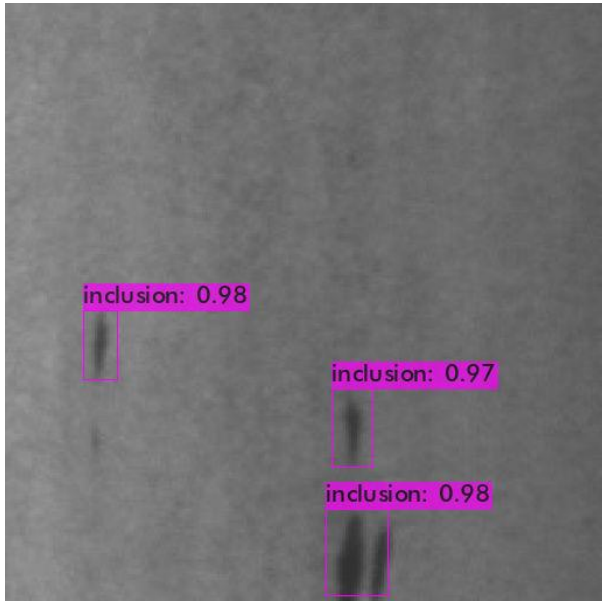
# Applications

- 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.



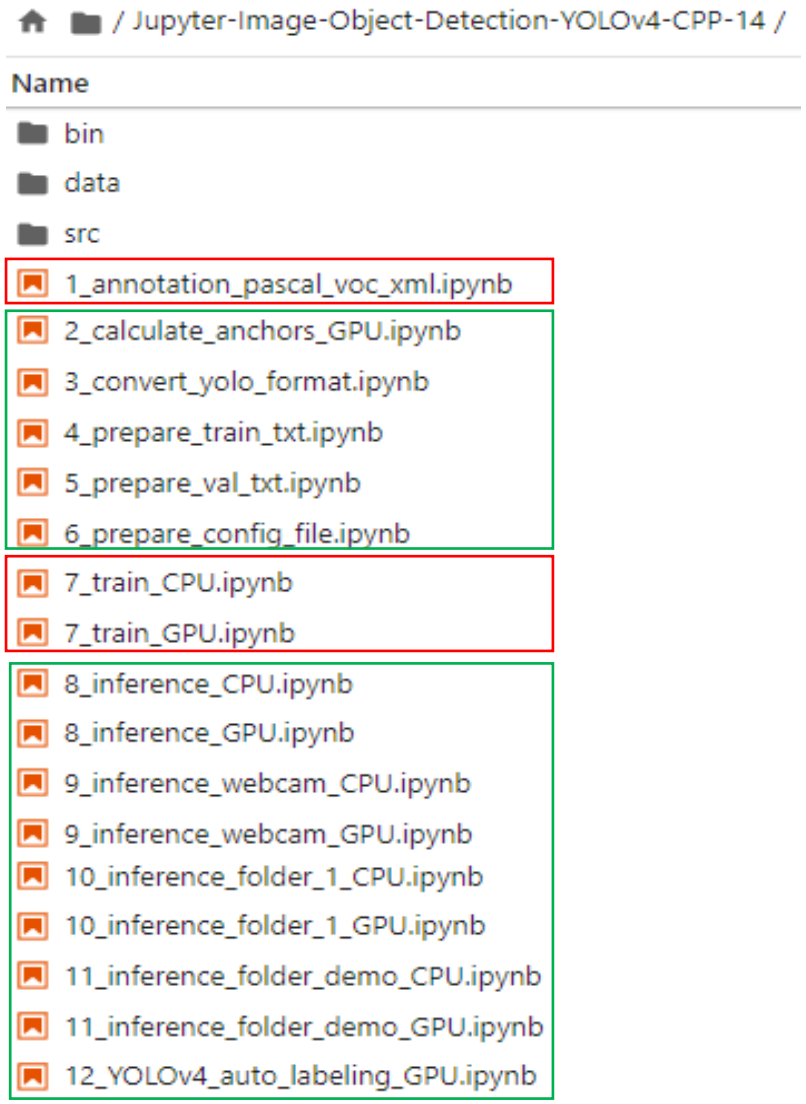



# How to use

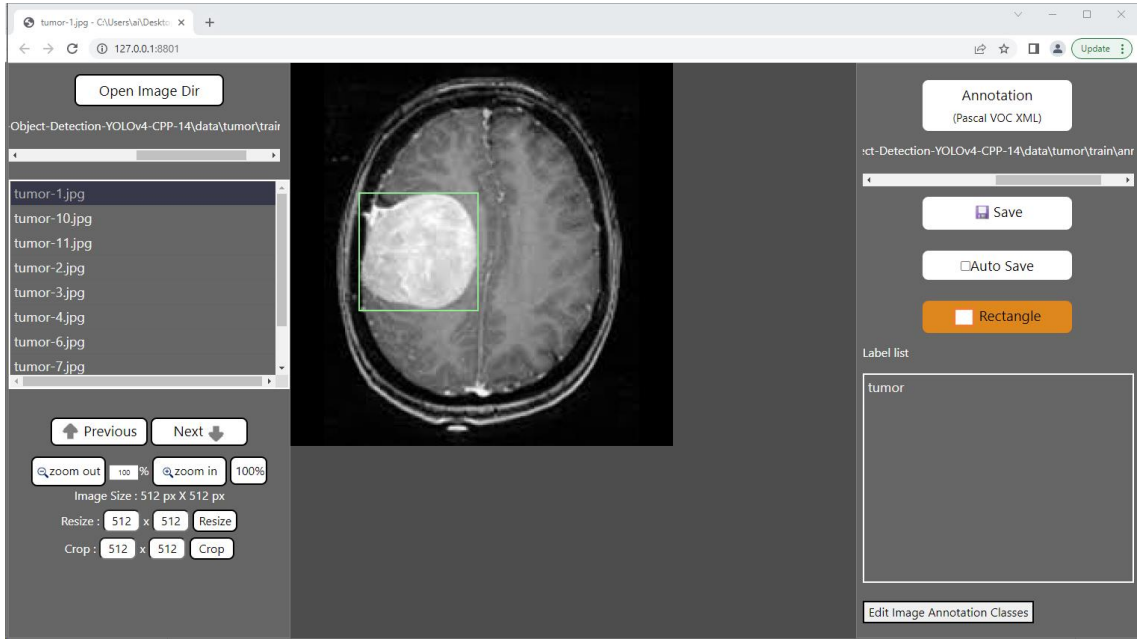
The main process is:

Annotate images -> Prepare files for training -> Training -> Inference



# 1_annotation_pascal_voc_xml.ipynb

Open the webpage for image annotation.




ipynb parameter:

- “port” is the port used by the webpage. If the port is occupied by the user, please change another port value by yourself.
- “image_folder” is the image path.
- “annotation_path” is the path to the annotation archive.

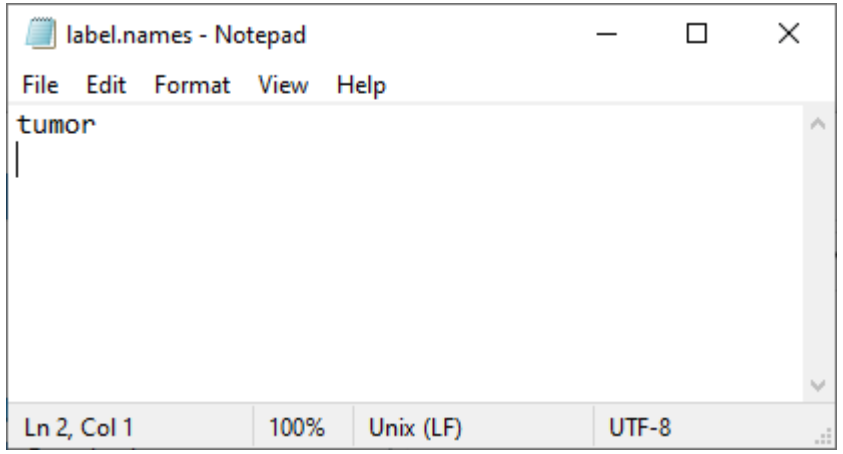
See Annotation.pdf for how to use annotation pages.

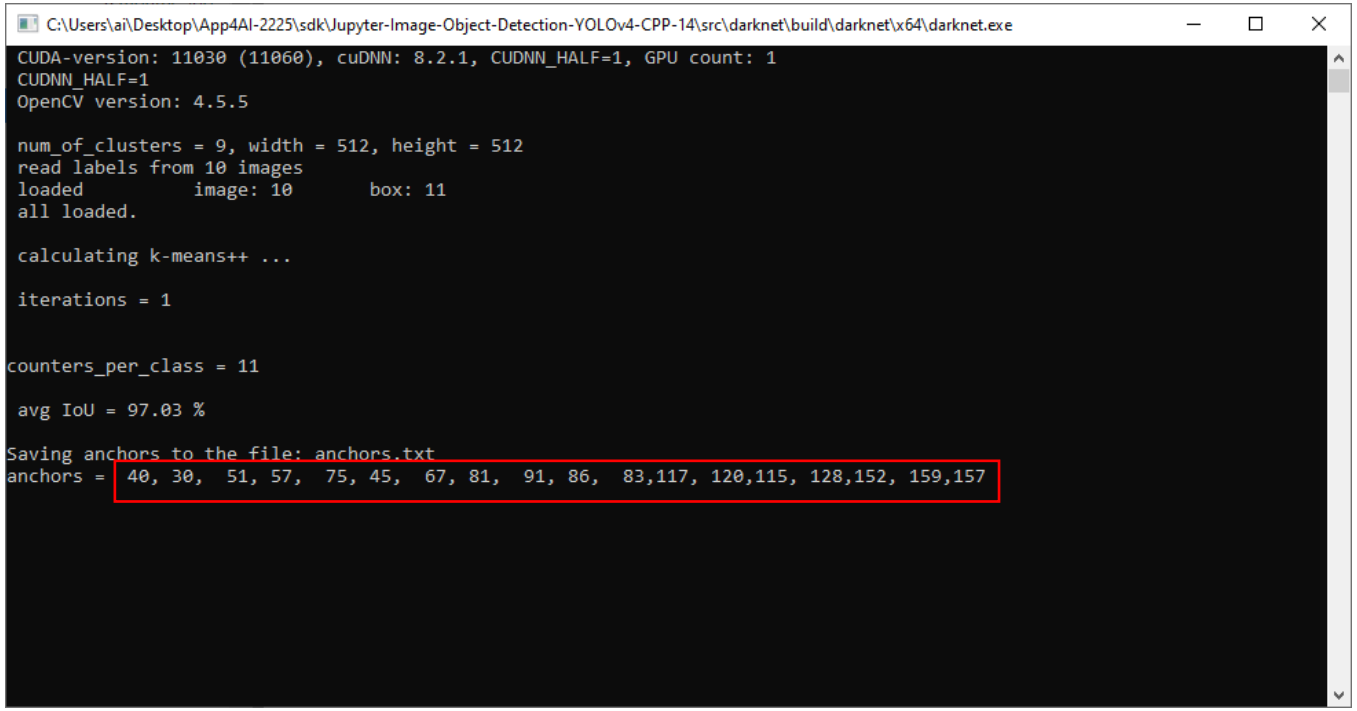
# 2_calculate_anchors_GPU.ipynb

Calculate the anchor value suitable for your dataset. Before running, please confirm the label.names in your dataset and whether the category filled in the content is correct.

supplement:

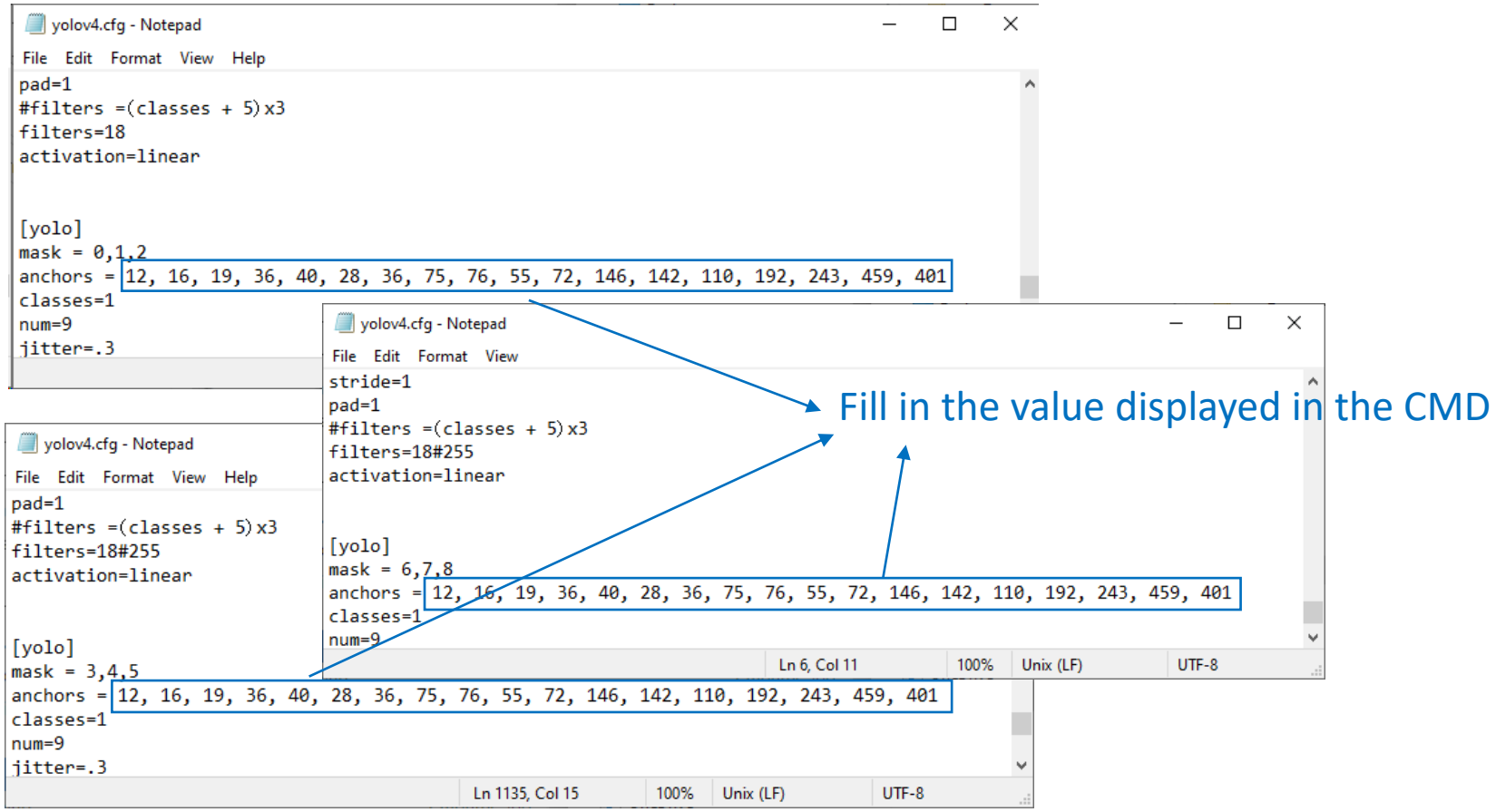
The content of label.names is the category name, excluding background, and the format distinguishes multiple categories by wrapping lines.



# Set yolov4.cfg anchor parameters

Fill in the value generated after running 2_calculate_anchors_GPU.ipynb into the yolov4.cfg anchor point parameter in the dataset dataset.

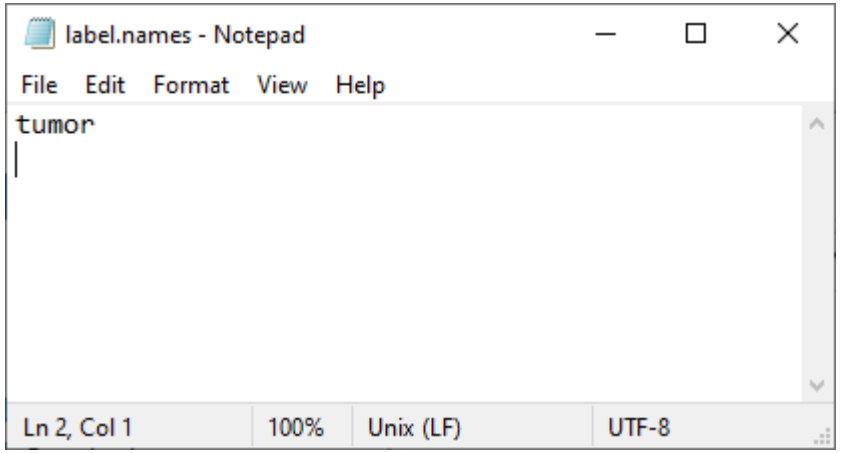


# 3_convert_yolo_format.ipynb

Convert the voc xml label file to the yolo format. Before running, please confirm label.names under the label_file path in #parameters and whether the content filled in the category is correct.

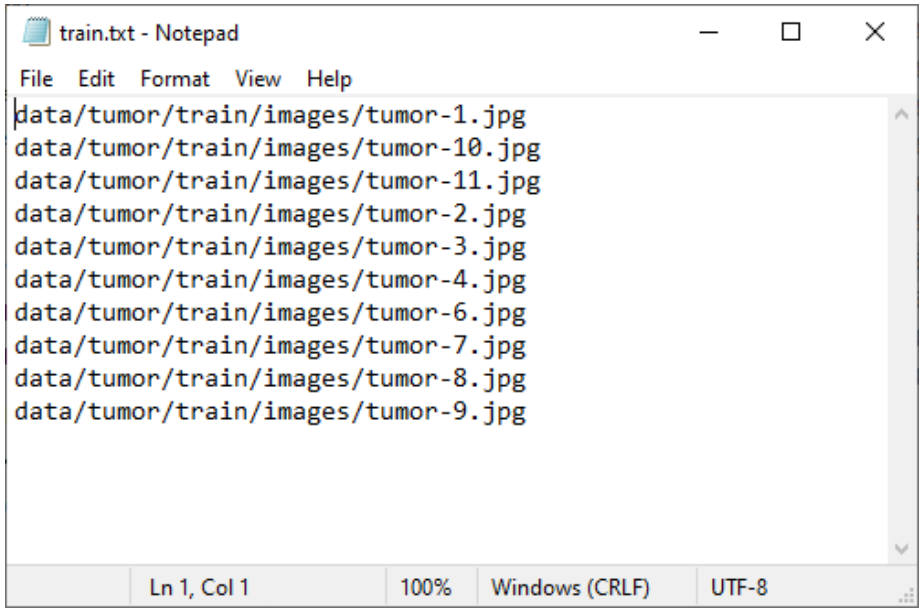
supplement:

The content of label.names is the category name without background.



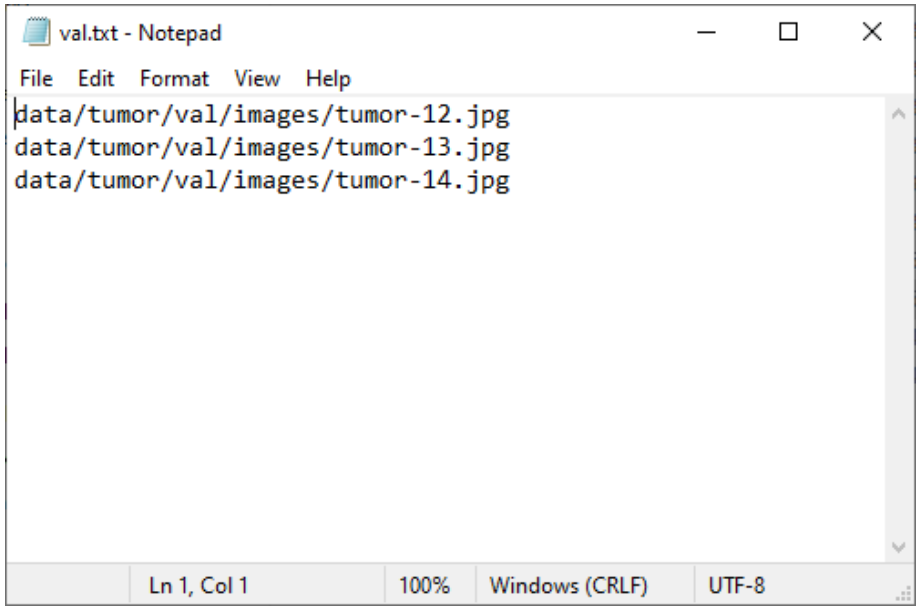
# 4_prepare_train_txt.ipynb

Generate training image path files “train.txt”.



# 5_prepare_val_txt.ipynb

Generate validation image path files “val.txt”.



# 6_prepare_config_file.ipynb

Set ipynb parameters, such as dataset name, number of categories.

```
import os
```

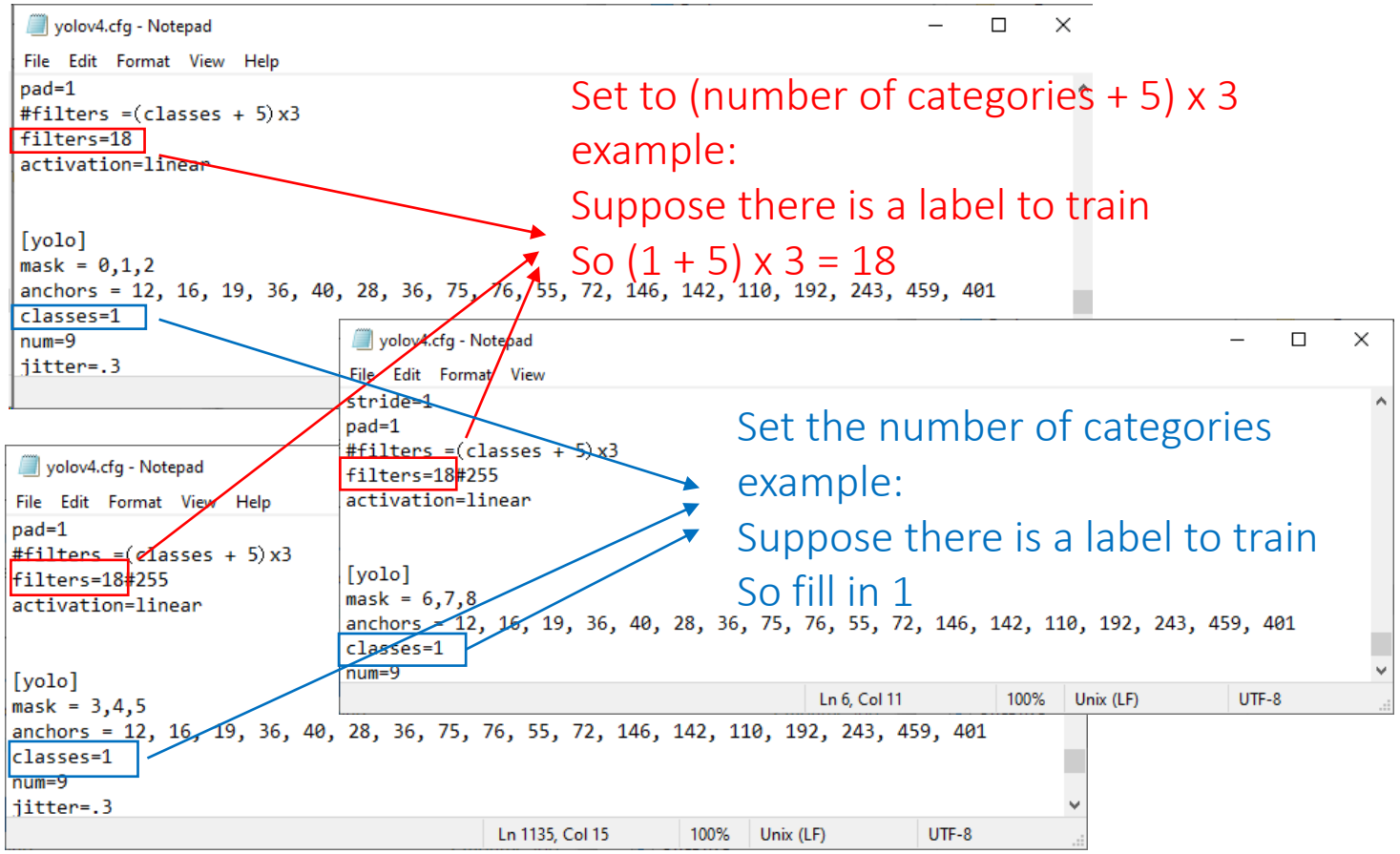
```
# parameters
classes= 1
train  = 'data/tumor/train.txt'
valid  = 'data/tumor/valid.txt'
names = 'data/tumor/label.names'
backup = 'data/tumor/model'
filename = 'data/tumor/voc.data'
```

Set the number of labels

set dataset name

# Set training yolov4.cfg related parameters

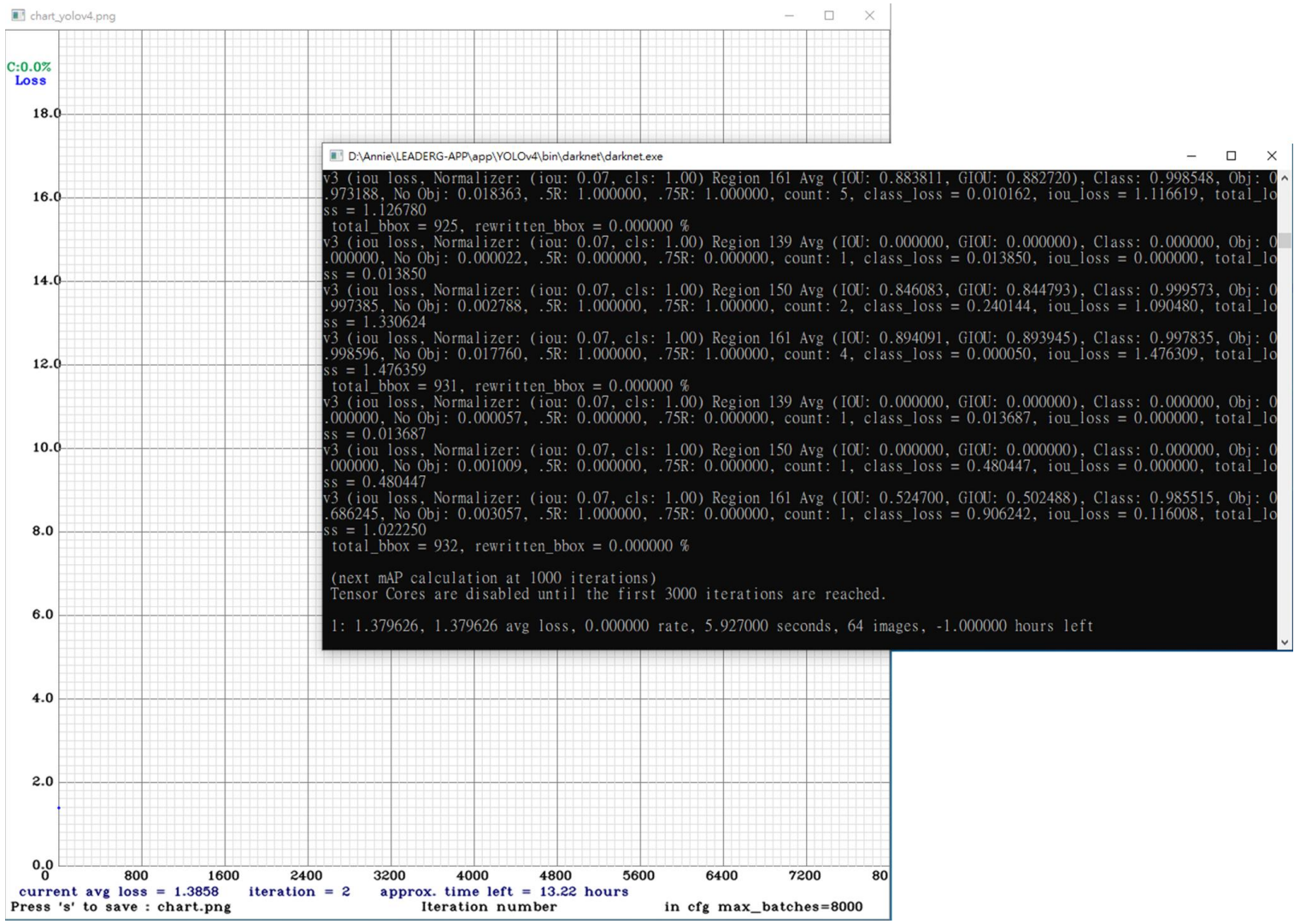
Set the parameters related to the number of file types in the yolov4.cfg file in the dataset.



# 7_train_GPU.ipynb

Start training.

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



# 8_inference_GPU.ipynb

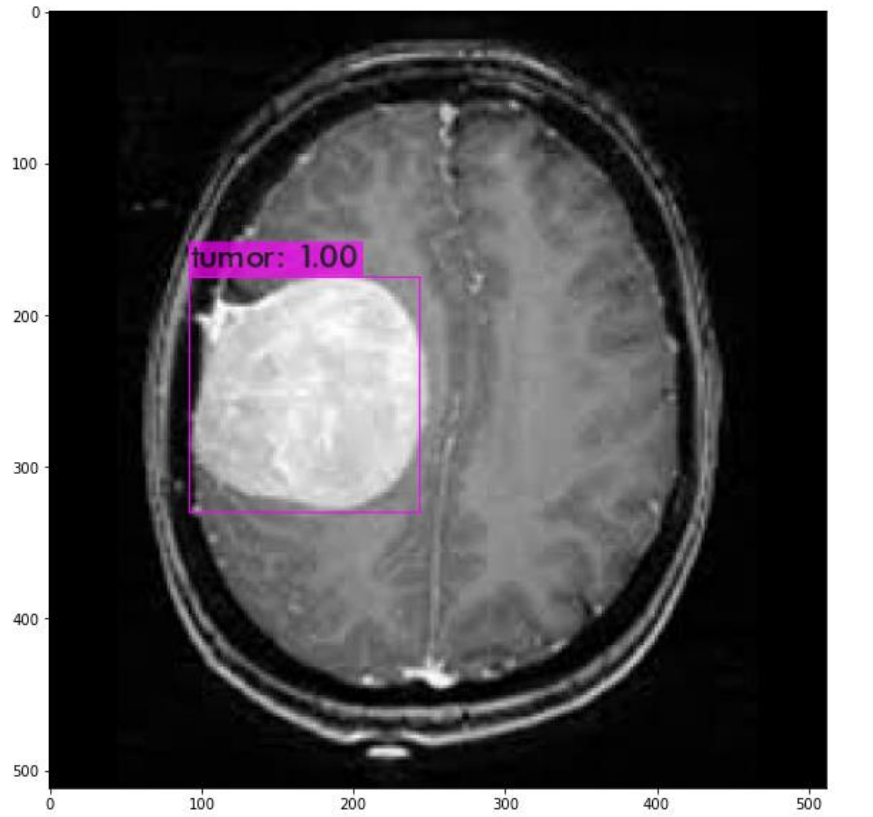
Infer a single image.

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

```
[6]: import matplotlib.pyplot as plt
     plt.rcParams["figure.figsize"] = (10, 10)

[7]: src = Image.open('predictions.jpg')
     plt.figure("src")
     plt.imshow(src)

[7]: <matplotlib.figure.AxesImage at 0x28bbfa33bb0>
```

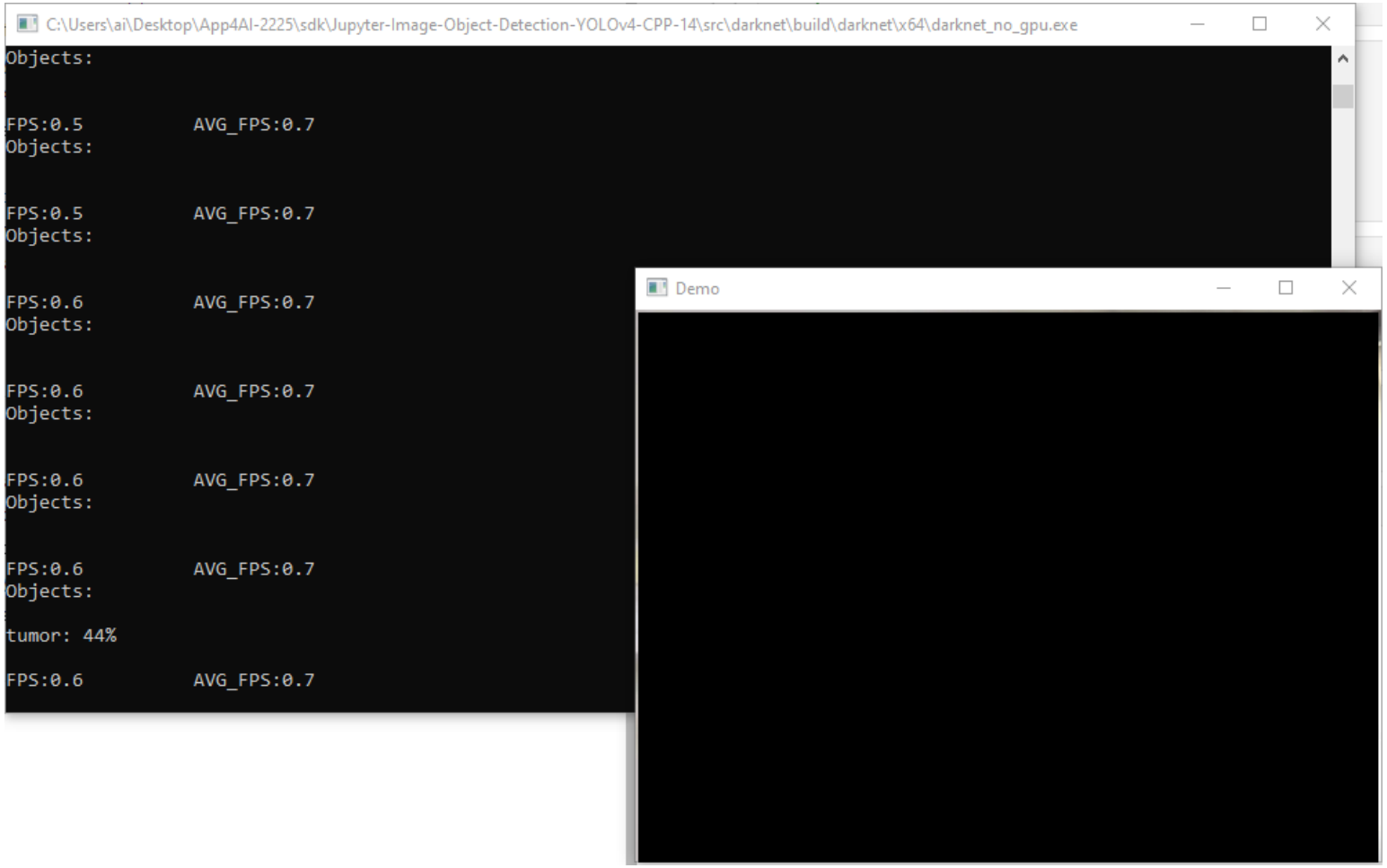


# 9_inference_webcam_GPU.ipynb

Infer the image of the webcam.

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

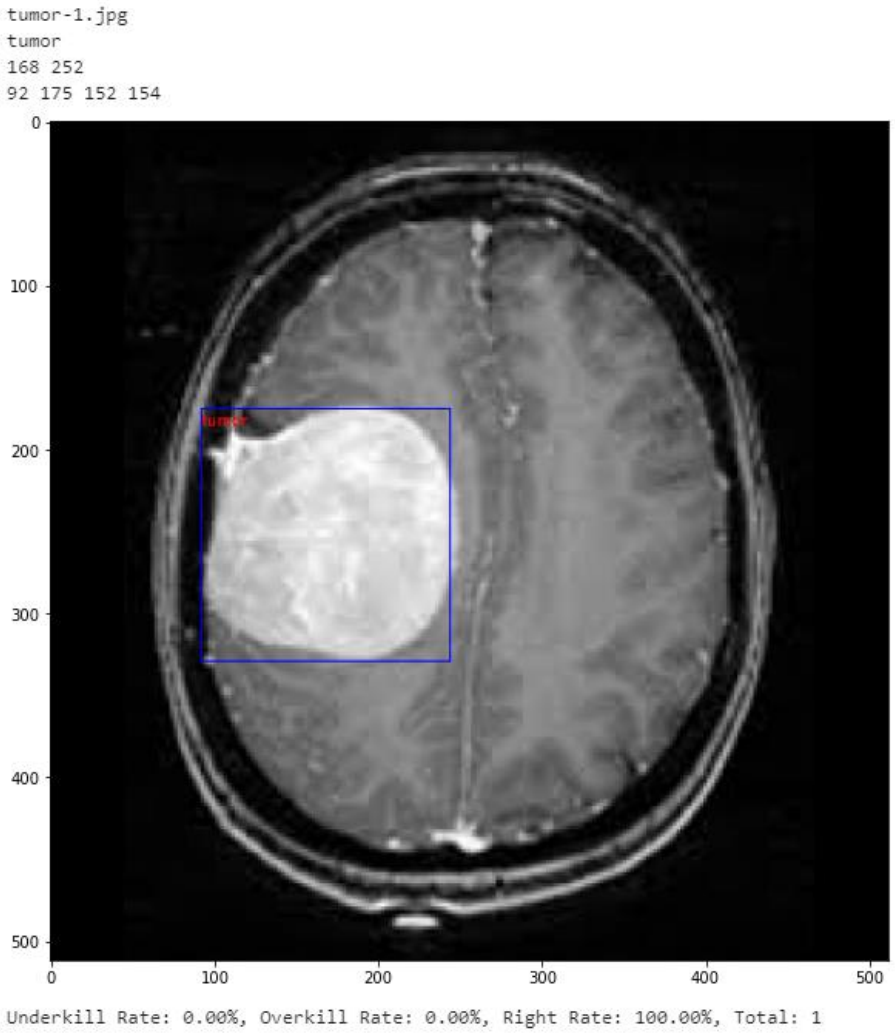
Press the “Esc” key on the display to turn the webcam off.



# 10_inference_folder_1_GPU.ipynb

Infer all images in the folder.

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

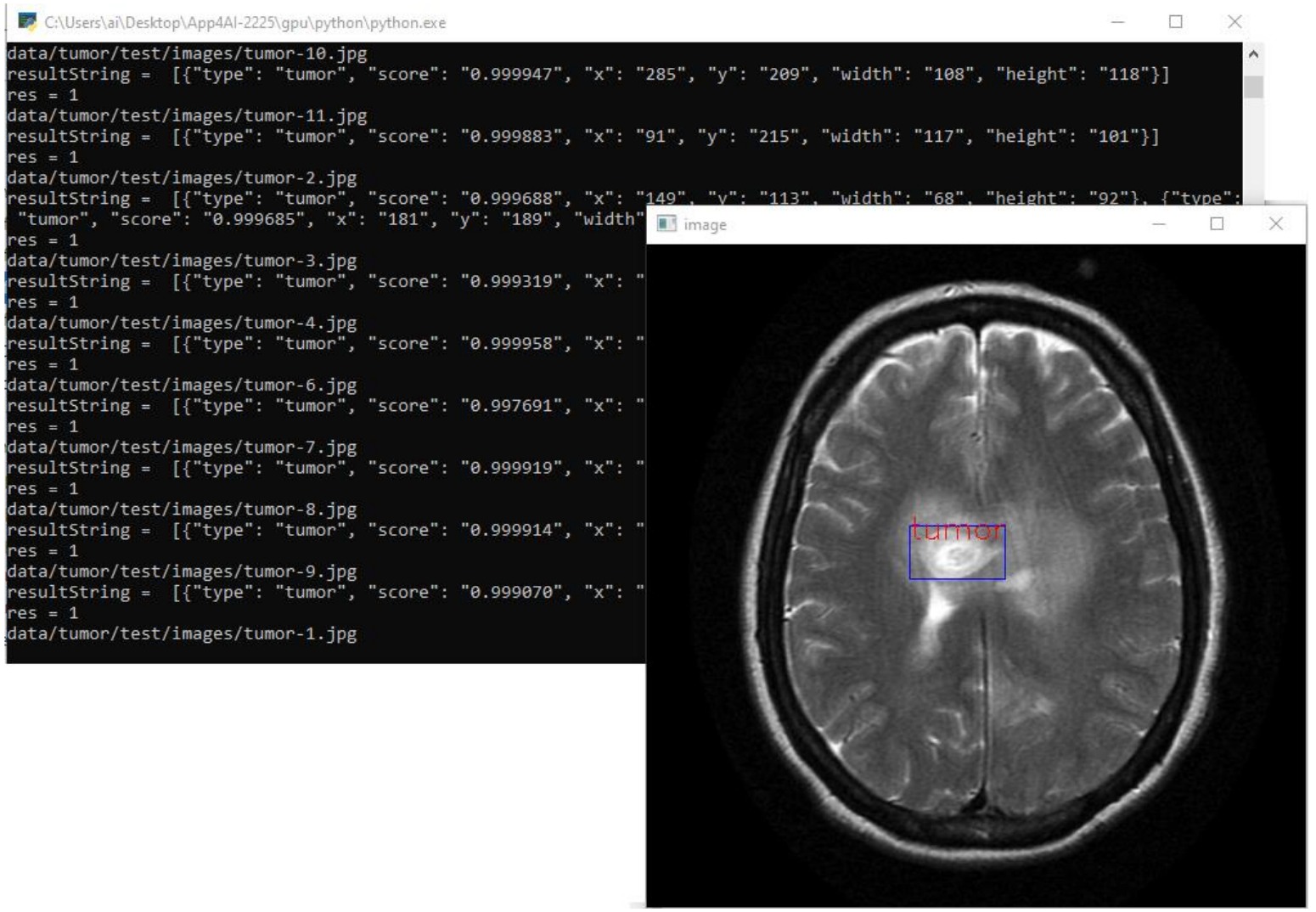


# 11_inference_folder_demo_GPU.ipynb

Continue to infer all images in the folder

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

Press the "Esc" key on the display to turn off continuous inference.



# 12_YOLOv4_auto_labeling_GPU.ipynb

According to the trained model, the images in the test folder in the dataset are automatically labeled, and the labeled images and labeled voc files are stored in the auto_labeling folder of the dataset.

# Reference

- Please refer to the readme.txt in the SDK folder.
- LEADERG AppForAI: https://www.leaderg.com/appforai-windows
- 