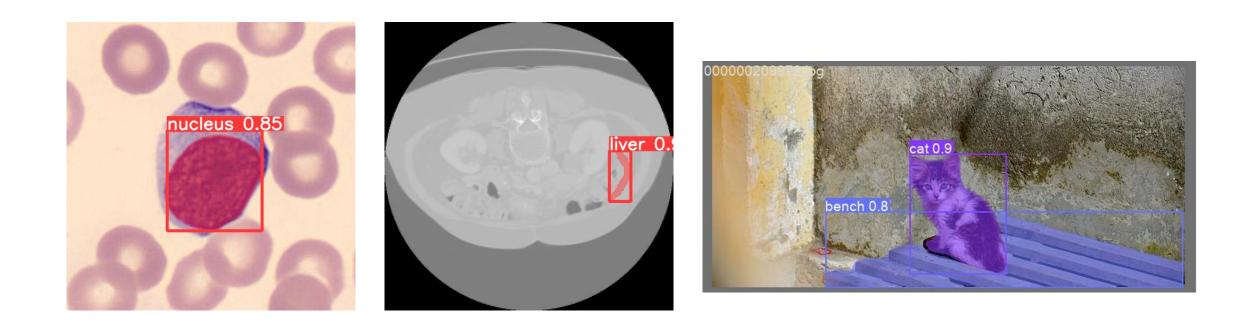
YOLOv7Seg

YOLOv7 is the most powerful object detection algorithm now. We also can use it to do instance segmentation instead of Mask R-CNN. After integrating YOLOv7Seg into an APP, it is more convenient for everyone to use it quickly.

Version 20230223

Applications

YOLOv7Seg 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

CT select dataset New Prepare 1. annotation labelme json 2. convert yolo format 3. prepare txt Annotation Port 8080 label.names image annotation classes.txt Train 6000 Workers 8 Batch Size 16 Epochs 4. train (GPU) 512 Image Size GPU Id 0 (0, 1, 2, ... for Nvidia GPU Select Pretrained Model 4. train (CPU) View coco.yaml calculate anchors 🔶 Edit yolov7-seg.yaml tensorboard Inference Inference Model data/CT/model/best.pt Select Threshold 0.7 0 GPU Id (0, 1, 2, ... for Nvidia GPU) 0 Webcam Id 5. inference (GPU) 6. inference folder (GPU) 7. inference webcam (GPU) 5. inference (CPU) 6. inference folder (CPU) 7. inference webcam (CPU)

LEADERG - YOLOv7Seg - Version 6

Dataset

- 🗆

Document

×

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.

EADERG - YOLOV79	Geg - Vers	ion 6							—		×
Dataset select dataset	СТ		,	New	,				Docum	ient	
Prepare			S	elect.	View	an	d N	ew dat	aset		
1. annotatio	on lab	elme jsor		· · · · · ·	ivert yolo f				prepare tx	t	
Annotation Port	808	0		label.names	image anr	notation	dasses.txt				
Train											
4. train (GF	PU)	Batch Size Image Siz		16 512		Epo GPU		6000 0	Workers (0, 1, 2, f	8 or Nvidia	GPU)
4. train (CF	4. train (CPU)		l Model o.yaml		Ilate anchors	•	Edit yo	Select blov7-seg.yam	l tenso	rboard	
Inference											
Inference Model	data/0	T/model/b	est.pt				Select	t			
Threshold	0.7										
GPU Id	0 (0, 1, 2, for Nvidia GPU)										
Webcam Id	0										
5. inference (GPU)		6. i	6. inference folder (GPU)			7. inference webcam (GPU)					
5. inference (CPU)			<mark>6.</mark> i	6. inference folder (CPU)			7. inference webcam (CPU)				

Prepare images

- After pressing the "folder" icon button, click the name of the dataset to be used, and put the PNG 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 "nucleus-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 labelme json".

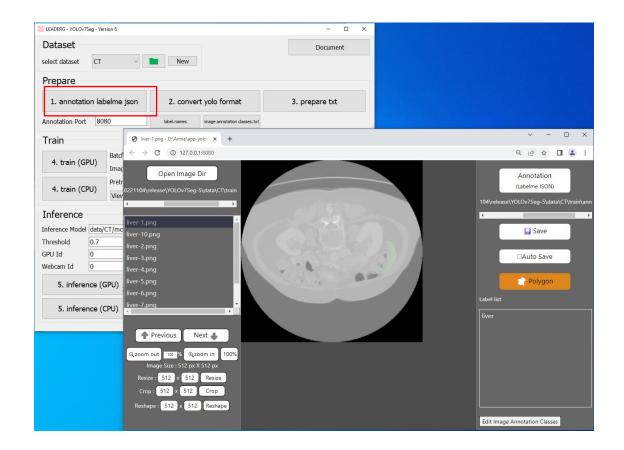
1. Annotation labelme json

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 dataset/val/images", then press "SELECT", then press Under "Annotation" in the upper right, select "Labelme JSON" and then select "your dataset/val/annotations" and then press "SELECT".

See Annotation.pdf for how to use annotation pages.

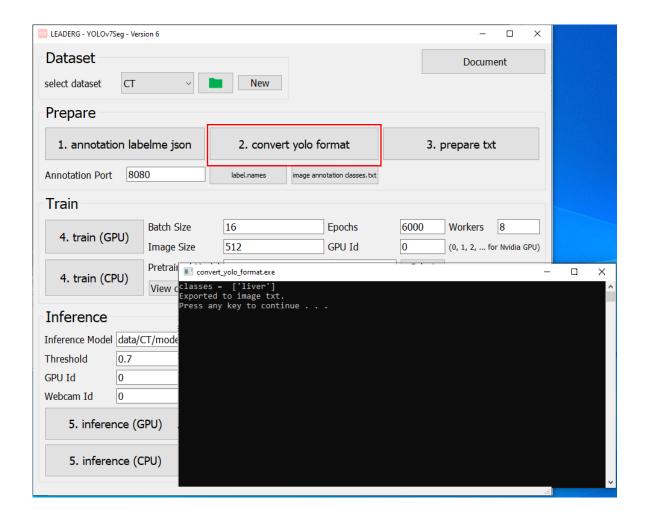


2.convert yolo format

After confirming that the category names in the label.names file are correct, you can press the button to convert the labelme json 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.



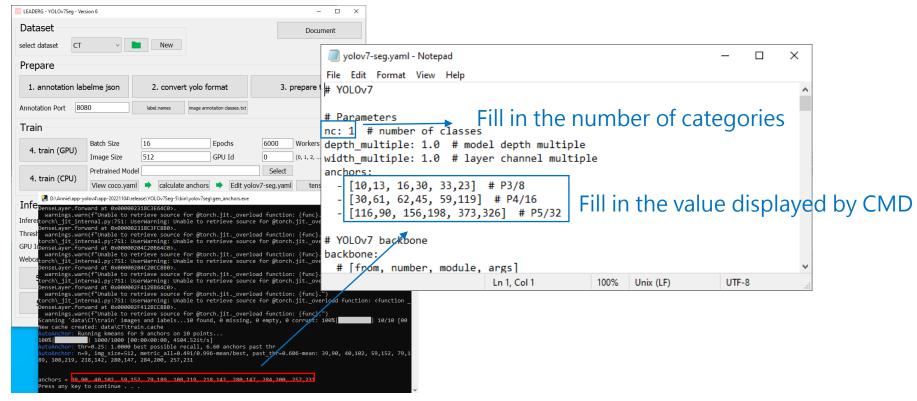
3.prepare txt

Generate training image path files "train.txt" and "val.txt".

ELADERG - VOLOv7Seg - Version 6	- 🗆 X	Train.txt - Notepad		×
Dataset	Document	File Edit Format View Help		
select dataset CT V New		data/CT/train/images/liver-1.png		^
Prepare		<pre>data/CT/train/images/liver-10.png data/CT/train/images/liver-2.png</pre>		
1. annotation labelme json 2. convert yolo format	3. prepare txt	<pre>data/CT/train/images/liver-3.png data/CT/train/images/liver-4.png data/CT/train/images/liver-5.png</pre>		
Annotation Port 8080 label.names image annotation classes.txt		data/CT/train/images/liver-6.png		
4. train (GPU) Batch Size 16 Epochs 60 Image Size 12 000000000000000000000000000000000000	00 Workers 8	<pre>data/CT/train/images/liver-7.png data/CT/train/images/liver-8.png data/CT/train/images/liver-9.png</pre>		
4. train (CPU) Pretrained Model Exported to train.txt. Exported to val.txt. View coco.yaml Press any key to continue		File Edit Format		×
Inference Inference Model data/CT/model/best.pt Threshold 0.7 GPU Id 0 (0, 1, 2, Webcam Id 5. inference (GPU) 6. ir 5. inference (CPU) 6. ir		data/CT/val/im data/CT/val/im data/CT/val/im data/CT/val/im data/CT/val/im data/CT/val/im data/CT/val/im data/CT/val/im	<pre>mages/liver-10.png mages/liver-2.png mages/liver-3.png mages/liver-4.png mages/liver-5.png mages/liver-6.png mages/liver-7.png mages/liver-8.png mages/liver-9.png</pre>	
		~	Ln 1, Col 1 100% Win	dows (CRLF) UTF-8

Set yolov7-seg.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-seg.yaml] to set the number of categories and anchor points. The number of categories must be the same as coco.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.



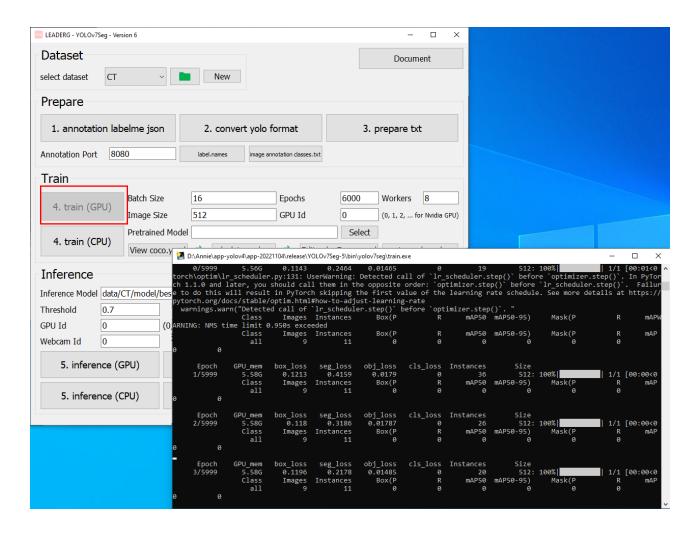
4. Train (GPU)4. Train (CPU)

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.



5. Inference (GPU)5. Inference (CPU)

Infer a single image.

Choose the Inference Model yourself.

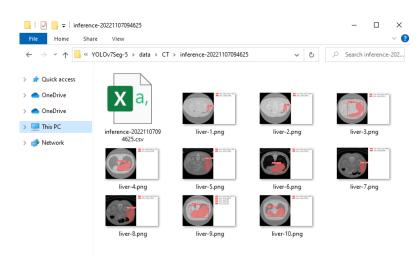
Dataset convision (io)(iage.pr):13: UserNaming: inable to retrieve source for @torch.jtr.overload function: <function: <="" <function:="" tu=""></function:>	🚾 LEADERG - YOLOv7Seg - Version 6	C:\Users\ai\Desktop\App4Al\app\YOLOv7Seg-6\bin\yolov7		- 🗆 X
1. annotation labelme json 2. convert yolo formalizer-alse, update-false, protect-date/CT, name=hone, exist_oksTrue, line_thickness=3, hTde_labels=False, htde_conf-Fal Annotation Port 8080 Woelnames mage smaller (Chick falled) Annotation Port 8080 Woelnames mage smaller (Chick falled) Train Fusing layers 900/000/7-863 Summary: 325 layers, 37842476 parameters, 0 gradients 4. train (GPU) Batch Size 16 E5 1. train (CPU) Pretrained Model Fusing layers 4. train (CPU) Pretrained Model Fusing layers Inference Inference Iver of calculate anchors Inference Model (data/CT/model/best.pt Select Threshold 0.7 (0, 1, 2, for Nvidia GPU) Webcam Id 0 (0, 1, 2, for Nvidia GPU) Webcam Id 0 7. inference webcam (GPU) 7. inference webcam (GPU)	select dataset CT V New	<pre>torch_jit_internal_pyi751: UserWarning: Un yer.forward at 0x000001DCB9A59670>. warnings.warn(f*Unable to retrieve source torch_jit_internal.pyi751: UserWarning: Un yer.forward at 0x000001DCB9A50880>. warnings.warn(f*Unable to retrieve source predict: weights=('data/CT/model/best.pt'), l.png, data=bin/data/ccoc128.yaml, imgsz=[5</pre>	<pre>able to retrieve source for @torch.jitoverload 4 e for @torch.jitoverload function: {func}.") able to retrieve source for @torch.jitoverload f e for @torch.jitoverload function: {func}.") source=C:/Users/ai/Desktop/App4AI/app/VOL0v7Sege(i12, 512), conf_thres=0.7, iou_thres=0.45, max_det</pre>	− Function: <function _densela<br="">5/data/CT/test/images/liver- 1000, device-0, view_img=Tr</function>
Train Full of a liver 1		<pre>alize=False, update=False, project=data/CT, se, half=False, dnn=False, show_rate=False, default message here: requirements: C:\User check failed.</pre>	name=None, exist_ok=True, line_thickness=3, hide_ save_plt=False, display_speed=0.0	_labels=False, hide_conf=Fal
Irain yolov7-seg summary: 325 layers, 37842476 parameters, 0 gradients 4. train (GPU) Batch Size 16 Image Size 512 G Pretrained Model Iver jixel count = 1262 Inference View coco.yaml (iver.1png) Inference Select Inference Model data/CT/model/best.pt Select Threshold 0.7 GPU Id 0 0 (0, 1, 2, for Nvidia GPU) Webcam Id 0 5. inference (GPU) 6. inference folder (GPU) 7. inference webcam (GPU) 7. inference webcam (GPU)	Annotation Port 8080 label.names image annot	^{at} YOLOv7 torch 1.12.0+cu113 CUDA:0 (NVIDIA T	TITAN RTX, 24576MiB)	
4. train (GPU) Batch Size Inage Size 512 Pretrained Model Inference Inference Model data/CT/model/best.pt Threshold 0.7 GPU Id 0 (0, 1, 2, for Nvidia GPU) Webcam Id 0 6. inference folder (GPU) 7. inference webcam (GPU)	Train	yolov7-seg summary: 325 layers, 37842476 pa	arameters, 0 gradients	
4. train (CPU) Inference Inference Inference Inference Model data/CT/model/best.pt Threshold 0.7 GPU Id 0 (0, 1, 2, for Nvidia GPU) Webcam Id 0 5. inference (GPU) 6. inference folder (GPU) 7. inference webcam (GPU)	4. train (GPU)	ESingle: liver center point: (420,270), pixe	el count = 1262	
Inference Model data/CT/model/best.pt Threshold 0.7 GPU Id 0 (0, 1, 2, for Nvidia GPU) Webcam Id 0 5. inference (GPU) 6. inference folder (GPU) 7. inference webcam (GPU)	4. train (CPU)		data\\CT\\None\\liver-1.png	liver, total count=1262
Threshold 0.7 GPU Id 0 (0, 1, 2, for Nvidia GPU) Webram Id 0 5. inference (GPU) 6. inference folder (GPU) 7. inference webcam (GPU)	Inference			
GPU Id 0 (0, 1, 2, for Nvidia GPU) Webram Id 0 5. inference (GPU) 6. inference folder (GPU)	Inference Model data/CT/model/best.pt	Select		
GPU Id 0 (0, 1, 2, for Nvidia GPU) Webcam Id 0 5. inference (GPU) 6. inference folder (GPU) 7. inference webcam (GPU)	Threshold 0.7		liver 0	
5. inference (GPU) 6. inference folder (GPU) 7. inference webcam (GPU)				
	Webcam Id 0			
5. inference (CPU) 6. inference folder (CPU) 7. inference webcam (CPU)	5. inference (GPU) 6. inference folder (GPU)	7. inference webcam (GPU)	and the second s	
	5. inference (CPU) 6. inference folder (CPU)	7. inference webcam (CPU)		

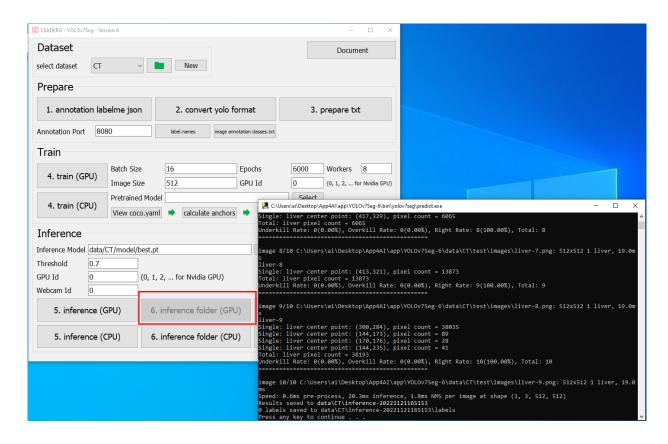
6. inference folder (GPU)6. inference folder (CPU)

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 can set the specified webcam.

Choose the Inference Model yourself.

Dataset 384x512 1 I/ver, 15.0ms select dataset CT Image Size 1 I/ver, 24.0ms Prepare 384x512 1 I/ver, 24.0ms 1. annotation labelme json 2. convert yolo formate: 384x512 1 I/ver, 24.0ms 384x512 1 I/ver, 24.0ms 384x512 1 I/ver, 24.0ms 384x512 1 I/ver, 24.0ms 1. annotation labelme json 2. convert yolo formate: 384x512 1 I/ver, 24.0ms 384x512 1 I/ver, 24.0ms 384x512 1 I/ver, 24.0ms 384x512 1 I/ver, 24.0ms 384x512 1 I/ver, 24.0ms 384x512 1 I/ver, 21.0ms 384x512 1 I/ver, 24.0ms 384x512 1 I/ver, 21.0ms 384x512 1 I/ver, 24.0ms 384x512 1 I/ver, 24.0ms 384x512 1 I/ver, 20.0ms 384x512 1 I/ver, 21.0ms 384x512 1 I/ver, 20.0ms 384x512 1 I/ver, 21.0ms 384x512 1 I/ver, 20.0ms 384x512 1 I/ver, 15.0ms 384x512 1 I/ver, 20.0ms 384x512 1 I/ver, 16.0ms 384x512 1 I/ver, 16.0ms 384x512 1 I/ver, 16.0ms 384	
Prepare 0: 343x512 1 1 iver, 15.0ms 0: 343x512 1 1 iver, 21.0ms 0: 343x512 1 1 iver, 24.0ms 0: 343x512 1 1 iver, 24.0ms 0: 343x512 1 1 iver, 24.0ms 0: 343x512 1 1 iver, 15.0ms 0: 343x512 1 1 iver, 16.0ms 0: 343x5	
1. annotation labelme json 2. convert yolo formate 0: 344x512 1 liver, 20.0ms Annotation Port 8080 label.names mage annotation 384x512 1 liver, 20.0ms 0: 384x512 1 liver, 20.0ms 0: 384x512 1 liver, 20.0ms 0: 384x512 1 liver, 20.0ms 0: 384x512 1 liver, 20.0ms 0: 384x512 1 liver, 20.0ms 0: 384x512 1 liver, 20.0ms 0: 384x512 1 liver, 20.0ms 0: 384x512 1 liver, 20.0ms 0: 384x512 1 liver, 20.0ms 0: 384x512 1 liver, 15.0ms 0: 384x512 1 liver, 30.0ms 0: 384x512 1 liver, 30.0ms 0: 384x512 1 liver, 30.0ms 0: 384x512 1 liver, 18.0ms 0: 384x512 1 liver, 18.0ms 0: 384x512 1 liver, 18.0ms 0: 384x512 1 liver, 18.0ms 0: 384x512 1 liver, 18.0ms 0: 384x512 1 liver, 18.0ms 0: 384x512 1 liver, 18.0ms 0: 384x512 1 liver, 18.0ms 0: 384x512 1 liver, 10.0ms 0: 384x512 1 liver, 10.0ms 0: 384x512 1 liver, 10.0ms 0: 384x512 1 liver, 10.0ms 0: 384x512 1 liver, 10.0ms 0: 384x512 1 liver, 10.0ms 0: 384x512 1 liver, 10.0ms 0: 384x512 1 liver, 10.0ms	
Image Size 16 Epcode: 384x512 1 liver, 15.0ms 4. train (GPU) Batch Size 16 Epcode: 384x512 1 liver, 28.0ms 9: 384x512 1 liver, 18.0ms 9: 384x512 1 liver, 38.0ms 9: 384x512 1 liver, 38.0ms 9: 384x512 1 liver, 18.0ms 9: 384x512 1 liver, 18.0ms 9: 384x512 1 liver, 18.0ms 9: 384x512 1 liver, 18.0ms 9: 384x512 1 liver, 18.0ms 9: 384x512 1 liver, 18.0ms 9: 384x512 1 liver, 18.0ms 9: 384x512 1 liver, 18.0ms 9: 384x512 1 liver, 18.0ms 9: 384x512 1 liver, 18.0ms 9: 384x512 1 liver, 18.0ms 9: 384x512 1 liver, 18.0ms 9: 384x512 1 liver, 18.0ms 9: 384x512 1 liver, 18.0ms 9: 384x512 1 liver, 18.0ms 9: 384x512 1 liver, 18.0ms 9: 384x512 1 liver, 18.0ms 9: 384x512 1 liver, 18.0ms 9: 384x512 1 liver, 18.0ms 9: 384x512 1 liver, 18.0ms 9: 384x512 1 liver, 18.0ms 9: 384x512 1 liver, 16.0ms 9: 384x512 1 liver, 16.0ms 9: 384x512 1 liver, 10.0ms 9: 384x512 1 liver, 17.0ms 9: 384x512 1 liver, 10.0ms 9: 384x512 1 liver, 21.0ms	
4. train (GPU) Batch Size 16 Epcode: 384x512 1 liver, 28.0ms Mage Size 512 GPU 9: 384x512 1 liver, 18.0ms 4. train (CPU) Pretrained Model 9: 384x512 1 liver, 14.0ms View coco.yaml Calculate anchors S84x512 1 liver, 16.0ms S84x512 1 liver, 16.0ms S84x512 1 liver, 16.0ms S84x512 1 liver, 17.0ms S84x512 1 liver, 21.0ms S84x512 1 liver	
4. train (CPU) View coco.yaml	X
Interence Model Judd/C1/Inodel/Desc.pt	
Threshold 0.7 GPU Id 0 (0, 1, 2, for Nvidia GPU) Webcam Id 0 (0, 1, 2, for Nvidia GPU)	
5. inference (GPU) 6. inference folder (GPU) 7. inference webcam (GPU)	
5. inference (CPU) 6. inference folder (CPU) 7. inference webcam (CPU)	

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.