YOLOv8Segmentation

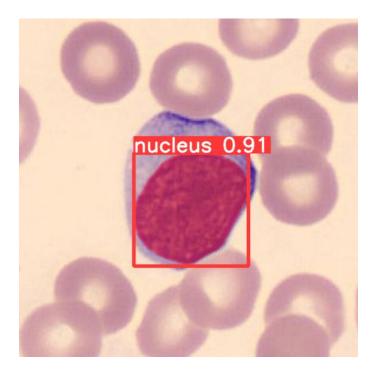
Ultralytics YOLOv8 is a cutting-edge state-of-the-art (SOTA) model developed by Ultralytics. Improve and optimize on the basis of the previous successful YOLO.

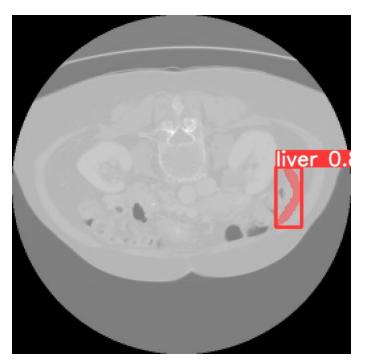
YOLOV8Segmentation uses the segmentation function in YOLOv8.

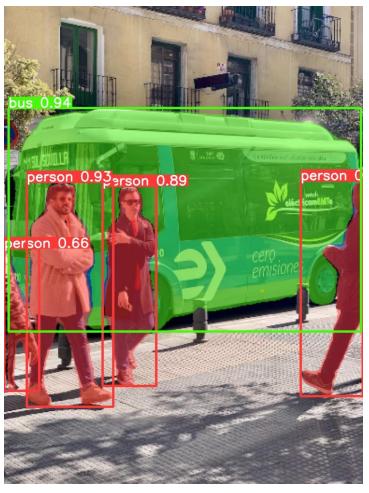
Version 20230223

Applications

YOLOv8Segmentation can be applied to factory defect segmentation, 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

🚥 LEADERG - YOLOv8Se	gmentation - Vers	ion 1		– 🗆 ×
Dataset select dataset	СТ	▼ ► New	APP Documer	SDK Document
Prepare				
1. annotation	labelme jso	n 2. convert yolo form	at	
Port	8080	view label.names	view image	_annotation.txt
Train				
3. train	(GPU)	3. train (CPU)		
Batch Size	8	Image Size 512		
Workers	2	Epochs 1000		
GPU ID	0	(0, 1, 2, for Nvidia GPU)	view voc.yaml	tensorboard
Pretrained Model	data/CT/mode	/yolov8x-seg.pt	Select	
Inference —				
4. inference	e (GPU)	5. inference folder (GPU	J) 6. inference v	webcam (GPU)
4. inference	e (CPU)	5. inference folder (CPL	J) 6. inference	webcam (CPU)
Inference Model		data/CT/model/best.pt		Select
Threshold		0.5		
GPU ID		0	(0, 1, 2, for Nvidia	a GPU)
Webcam ID		0		

Select dataset

Select the dataset for training or inference.

- The "Folder" icon button next to the pulldown 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.

		<u> Select, V</u>	<u>'lew a</u>	<u>na ivew</u>	dat	ase
LEADERG - YOLOv	Segmentation - Ve	rsion 1			-	
Jataset select dataset	СТ		New	APP Documen	t SDK [Docume
repare —						
1. annotatio	n labelme js	on 2. convert yol	o format			
Port	8080	view label.na	mes	view image_	annotation.1	txt
rain						
3. tra	in (GPU)	3. train (CPU)			
Batch Size	8	Image Size 51	2			
Workers	2	Epochs 10	00			
GPU ID	0	(0, 1, 2, for Nvidia 0	SPU)	view voc.yaml	tenso	orboard
Pretrained Mod	lel data/CT/mod	el/yolov8x-seg.pt		Select		
nference —						
4. inferer	ice (GPU)	5. inference folde	er (GPU)	6. inference w	vebcam	I (GPU
4. inferer	nce (CPU)	5. inference folde	er (CPU)	6. inference w	ebcam	n (CPU
Inference Mode	el	data/CT/model/best.pt			Se	lect
Threshold		0.5				
GPU ID		0	(0, 1, 2, for Nvidia	GPU)	
Webcam ID		0				

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 "liver-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".

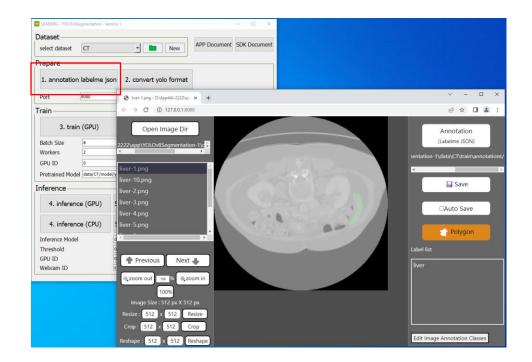
1. annotation labelme json

After running, it will open the annotation webpage for image annotation. Currently, only image formats with file extension .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 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.

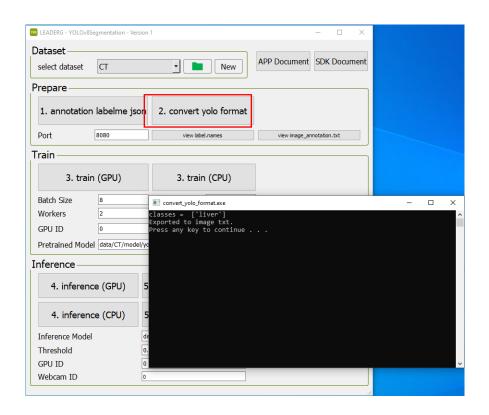


2.convert yolo format

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.



Check voc.yaml parameters

Confirm the content of the voc.yaml file in the dataset, such as the name of the dataset, the number of categories, and the name.

//////////////////////////////////////				_		\times
File Edit Format View Help						
<pre># Train/val/test sets as 1) list: [path/to/imgs1, path/to path: CT # dataset root dir train: train val: val test: # test images (option)</pre>	heck dataset r		le: path/to/img	s.txt	, or	3) ^
<pre># Classes names: 0: liver</pre>	k the number	of lal	pels and lab	el na	ame	:S
	Ln 5, Col 32	100%	Windows (CRLF)	UTF-8		

LEADERG - YOLOv8Seg	mentation - Vers	ion 1			_		×
Dataset	СТ	<u> </u>	New	APP Document	SDK I	Docum	nent
Prepare							_
1. annotation l	abelme jso	2. convert	yolo format				
Port	3080	view labe	el.names	view image_an	notation.	txt	
Train ———							
3. train	(GPU)	3. trair	n (CPU)				
Batch Size	8	Image Size	512				
Workers	2	Epochs	1000				
GPU ID	0	(0, 1, 2, for Nvic	lia GPU)	view voc.yaml	tens	orboard	I
Pretrained Model	data/CT/model	/yolov8x-seg.pt		Select			
Inference —							
4. inference (GPU) 5		5. inference folder (GPU)		6. inference we	ebcam	n (GP	U)
4. inference (CPU) 5		5. inference folder (CPU)		6. inference webcam (CPU)			J)
Inference Model		data/CT/model/best.p	data/CT/model/best.pt		Se	lect	
Threshold		0.5	.5				
GPU ID		0		(0, 1, 2, for Nvidia G	PU)		
Webcam ID		0					

3. train (GPU) 3. train (CPU)

Start training.

Note:

Pretrained Model is the pretrained model path used.

Epochs is the number of training epochs.

Select dataset CT	• New	APP Document SDK Document
Prepare		,
1. annotation labelme js	on 2. convert yolo format	
Port 8080	view label.names	<pre>torchvisionliolimage.pv13: UserNarning: Failed to load image Python extension: torch_jlt_internal.py:75: UserNarning: Unable to retrieve source for @torch.jlt_overload function: <function _dense<br="">ayer.forward at @x0000026581070430.</function></pre>
Train 3. train (GPU)	3. train (CPU)	<pre>warnings.warn("Unable to retrieve source for glorch.jit.overload function: (func).") torch.jit.thermal.py:Sit.UserNarning: Unable to retrieve source for glorch.jit.overload function: <function _dense<br="">ayer.forward at ex000002680100070, warnings.wan("Unable to retrieve source for glorch.jit.overload function: {func].") torchvisioniloiksage.py:13: UserNarning: Failed to load image Python extension: torch.jit.thermal.py:Sit.UserNarning: Failed to load image Python extension:</function></pre>
Batch Size 8 Workers 2	Image Size 512 Epochs 1000	ayer.forward at exceeded1985/07430. warnings.warn("bmable to retrieve source for @torch.jit.overload function: [func].") torch.jit.internal.py:751: UperMarning: Umable to retrieve source for @torch.jit.overload function: cfunction _Dense "warnings.warn("bmable to retrieve source for @torch.jit.overload function: (func].")
GPU ID 0 Pretrained Model data/CT/mod	(0, 1, 2, for Nvidia GPU) el/yolov8x-seg.pt	<pre>inings.wein() ondoire to recrete source to goot.jir.jeridou function. [func].] 1/b00 5.266 2.277 5.543 4.07 2.662 512: 100%] 2/2 [00:034 torchioptimilin_scheduler.py:131: Userwarning: Detected call of 'In_scheduler.step()' before 'Discheduler.step()'. Forgi rch 1.1.0 and later, you should call them in the opposite order: 'optimizer.step()' before 'In scheduler.step()'.</pre>
Inference		ure to do this will result in Pytorch skipping the first value of the learning rate schedule. See more details at http ://pytorch.org/docs/stable/optim.html#Mow-to-adjust-learning-rate warnings.warn("Detected call of "in scheduler.step()' before 'optimizer.step()". "
4. inference (GPU)	5. inference folder (GPU)	Class Images Instances Box(P R mAPS0-95) Mask(P R mA 6.orchvision\lo\image.py:13: UserWarning: Falled to load image Python extension: torch_ji_intermal.py:75: UserWarning: Unable to retrieve source for @torch.jit_overload function: <function_dense< td=""></function_dense<>
4. inference (CPU)	5. inference folder (CPU)	agen_formard at ex0000022588T04309. warnings warn(5"unable to retrieve source for @torch.jit, overload function: (func).") 6 torch.jit.internal.pp:751: UserNarning: Unable to retrieve source for @torch.jit_overload function: <function_dense agen_formard at e0000002258T06709.</function_dense
Inference Model	data/CT/model/best.pt	warnings.warn(f"Unable to retrieve source for @torch.jitoverload function: {func}.")
Threshold	0.5	
GPU ID Webcam ID	0	(0, 1, 2, for Nvidis GPU)

4. inference (GPU)4. inference (CPU)

Infer a single image.

Choose the Inference Model yourself.

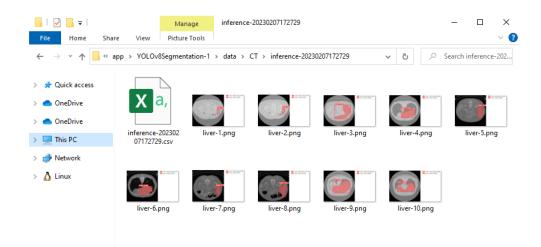
LEADERG - YOLO	v8Segmentation - Ver	on 1 🛛 🔁 D:\App4AI-2222\app\YOLOv8Segmentation-1\bin\yolov8seg\predict.exe	- 🗆 ×
Dataset	СТ	<pre>torchvision\io\image.py:13: UserWarning: Failed to load image torch_jit_internal.py:751: UserWarning: Unable to retrieve s <function_denselayer.forward 0x00000202334493a0="" at="">. warnings.warn(f"Unable to retrieve source for @torch.jit torch_jit_internal.py:751: UserWarning: Unable to retrieve s <function_denselayer.forward 0x0000020233495e0="" at="">.</function_denselayer.forward></function_denselayer.forward></pre>	<pre>source for @torch.jitoverload function: overload function: {func}.")</pre>
Prepare —		warnings.warn(f"Unable to retrieve source for @torch.jitc Ultralytics YOLOv8.0.6 Python-3.9.12 torch-1.12.0+cu113 CUDA	
1. annotatio	on labelme jso	Fusing layers	
Port	8080		
Train ——			
3. tra	ain (GPU)		
Batch Size	8	Image	
Workers	2	Epoch D:\App4AI-2222\app\YOLOv8Segmentatio	
GPU ID	0	(0, 1, 2	liver, total count=1289 liver, count=1289
Pretrained Mo	del data/CT/mode	/yolov8x-se	
Inference			
4. infere	nce (GPU)	5. inference folder (GPU) 6. infere	liver 0.
4. infere	nce (CPU)	5. inference folder (CPU) 6. infere	
Inference Mod	del	data/CT/model/best.pt	
Threshold		0.5	
GPU ID		0 (0, 1, 2, fo	
Webcam ID		0	

5. inference folder (GPU) 5. 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.



M LEADERG - YOLOv8Segmentation - Version	1	– 🗆 X	
Dataset			
select dataset CT	- Ne	V App4AI-2222\spp\YOLOv8Segmentation-1\bin\yolov8seg\predict.exe -	o x
Prepare		Underkill Rate: 0(0.00%), Overkill Rate: 0(0.00%), Right Rate: 3(100.00%), Total: 3	^
		<pre>image 3/10 D:\App4AI-2222\app\YOLOv8Segmentation-1\data\CT\test\images\liver-2.png: 512x512 1 liver, Underkill Rate: 0(0.00%), Overkill Rate: 0(0.00%), Right Rate: 4(100.00%), Total: 4</pre>	45.0ms
1. annotation labelme json	2. convert yolo fo	<pre>mat image 4/10 D:\App4AI-2222\app\YOLOv8Segmentation-1\data\CT\test\images\liver-3.png: 512x512 1 liver, Underkill Rate: 0(0.00%), Overkill Rate: 0(0.00%), Right Rate: 5(100.00%), Total: 5</pre>	39.0ms
Port 8080	view label.names	image 5/10 D:\App4AI-2222\app\YOLOv8Segmentation-1\data\CT\test\images\liver-4.png: 512x512 1 liver,	32.0ms
Train		Underkill Rate: 0(0.00%), Overkill Rate: 0(0.00%), Right Rate: 6(100.00%), Total: 6	
		<pre>image 6/10 D:\App4AI-2222\app\YOLOv8Segmentation-1\data\CT\test\images\liver-5.png: 512x512 1 liver, Underkill Rate: 0(0.00%), Overkill Rate: 0(0.00%), Right Rate: 7(100.00%), Total: 7</pre>	32.0ms
3. train (GPU)	3. train (CPU	<pre>image 7/10 D:\App4AI-2222\app\YOL0v8Segmentation-1\data\CT\test\images\liver-6.png: 512x512 1 liver, Underkill Rate: 0(0.00%), Overkill Rate: 0(0.00%), Right Rate: 8(100.00%), Total: 8</pre>	33.0ms
Batch Size 8	Image Size 512	<pre>image 8/10 D:\App4AI-2222\app\YOLOv8Segmentation-1\data\CT\test\images\liver-7.png: 512x512 1 liver,</pre>	30.0ms
Workers 2	Epochs 1000	Underkill Rate: 0(0.00%), Overkill Rate: 0(0.00%), Right Rate: 9(100.00%), Total: 9	
GPU ID	(0, 1, 2, for Nvidia GPU)	<pre>image 9/10 D:\App4AI-2222\app\YOLOv8Segmentation-1\data\CT\test\images\liver-8.png: 512x512 1 liver, Underkill Rate: 0(0.00%), Overkill Rate: 0(0.00%), Right Rate: 10(100.00%), Total: 10</pre>	30.0ms
Pretrained Model data/CT/model/yo	blov8x-seg.pt	image 10/10 D:\App4AI-2222\app\VOLOv8Segmentation-1\data\CT\test\images\liver-9.png: 512x512 1 liver,	20 Amc
Inference		Speed: 0.6ms pre-process, 35.3ms inference, 3.1ms postprocess per image at shape (1, 3, 512, 512) Results saved to data\Clinference-20230207170909	2.5.000
4. inference (GPU) 5	, inference folder (G	Press any key to continue	v
4. inference (CPU) 5	i. inference folder (C	2U) 6. inference webcam (CPU)	
Inference Model da	ata/CT/model/best.pt	Select	
Threshold 0.	5		
GPU ID 0		(0, 1, 2, for Ividia GPU)	
Webcam ID 0			

6. inference webcam (GPU)6. inference webcam (CPU)

Infer the image of the webcam.

 When using GPU mode, it may take some time to display the webcam image initially, please be patient.

The Webcam Id parameter below can set the specified webcam.

LEADERG - YOLOv8	Segmentation - Versio		4AI-2222\app\YOLOv8Segmentation-1\bin\yolo	v8seg\predict.exe	- 🗆 X	
Dataset ——			12 17.0ms 12 17.0ms		^ C	
select dataset	СТ		12 18.0ms 12 21.1ms			
Select dataset	СГ	0: 384x5	12 27.0ms			
repare —			L2 17.0ms L2 17.0ms			
•		0: 384×5	12 18.0ms			
1. annotation	n labelme jsor	n 2. convert yc <mark>0: 384x5</mark>	12 20.0ms 12 15.0ms			
		0: 384x5	12 18.0ms			
Port	8080	view label.no: 384x5				
			12 18.0ms 12 26.0ms			
rain		0: 384x5	12 33.9ms	I 0		- 🗆
7 huni			12 25.0ms 12 34.0ms			
S. trai	n (GPU)	0: 384X5	12 19.0ms			
Batch Size	8		12 18.0ms 12 20.0ms			
Workers	2	0: 384x5	12 22.0ms 12 20.0ms			
		0: 384x5	12 22.0ms			
GPU ID	0	(0, 1, 2, for Nvidia 0: 384x5 0: 384x5	12 36.0ms 12 27.0ms			
Pretrained Mod	el data/CT/model/					
nference —				3		
4. inferen	ce (GPU)	5. inference folder (GPU	6. inference webcam (GPU)			
4. inferen	ce (CPU)	5. inference folder (CPU)	6. inference webcam (CPU)	-		
Inference Mode	I [data/CT/model/best.pt	Select			
Threshold	[0.5				
GPU ID	[0	(0, 1, 2, for Nvidia GPU)			
Webcam ID		~				

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.