<https://github.com/shenzhao-kaleido/anno.git>

**Usage**

call trainProcess function in trainprocess.py as

trainProcess(riverDataset, "water\_v2", "river\_train.npy", [0, 1500], [1950, 2150], "model\_1")

trainProcess(riverDataset, "water\_v2", "river\_train.npy", [0, 1500], [1950, 2150], "scibble\_model")

"water\_v2" for folder that contains files, download from <https://www.kaggle.com/datasets/gvclsu/water-segmentation-dataset/download?datasetVersionNumber=4>

For other datasets, you can find related download sources from myDataset.py file

Liver: <https://www.kaggle.com/datasets/andrewmvd/liver-tumor-segmentation>

Brain: <https://www.kaggle.com/datasets/awsaf49/brats20-dataset-training-validation>

Person: https://www.kaggle.com/datasets/tapakah68/segmentation-full-body-tiktok-dancing-dataset

https://www.kaggle.com/datasets/tapakah68/segmentation-full-body-mads-dataset

https://www.kaggle.com/datasets/tapakah68/supervisely-filtered-segmentation-person-dataset

"river\_train.npy" is the name list for all images, you can find it in npyFile folder

\_nonzero indicates that it only contains images with targets (images without targets are removed)

[0, 1500] is the training range: use images from #0 to # 1500 in the name list as the train set

[1950, 2150] is the validating range: use images from #1950 to #2150 in the name list as the valid set

"model\_1" is the saved model name, saved as model\_1.pt

you could search "NEED CHANGE" to find places that need to change

although I include create\_scribble function in trainprocess.py, it does take a long time to create scribble,

you might want to generate scribble and save it someplace. you could find scribble\_gen to do so

model.py is the U-Net from github <https://github.com/jaxony/unet-pytorch/blob/master/model.py>

**files may be helpgful:**

1. You may find some functions useful in mayHelp folder
2. ace.py is what I used as contract enhancement method to MRI. You may want to normalize images in another way
3. You can find samples of created polygons and filtered masks in polysample.ipynb.
4. You can find many other different ways to generate scribbles in scribbles.ipynb.

Two main ways are tried: polygon and randon walk

1. Polygon (looks better)

Generate a polygon from mask

Decompose polygon into triangles

Connect these triangles (in many ways)

1. Random walk

Start at a random point

Move to a random direction until a known barrier in front (mask boundaries or previous path)

Change to another direction, repeat until nowhere to go

1. Xxxtest.ipynb are used to get some sample output from model
2. Maciej\_check is to compare our annotation to provided mask
3. To annotate by ourselves, Maciej wanted to do it remotely and with me. So you may need an online annotator.

I used from <https://github.com/jsbroks/coco-annotator>

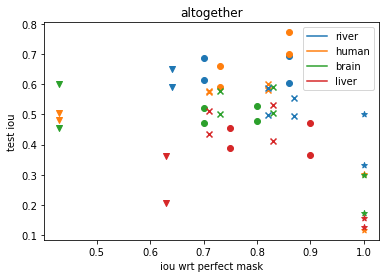
You need a Linux environment with docker to set the server up, and then forward your port to a public address

You can find my previous instruction for Maciej’s annotation in a word “annotation instruction”

1. In word price, you can find the prices of different annotation styles for several annotation service providers. The price can be regarded as a reference for annotation time.

**To do:**

1. Annotate and estimate annotation time of different methods
2. Find advanced methods for repeated training; I suggest CRF



Possible outcome