Tensorflow-Image-Segmentation-BUS-UC-Malignant (2025/05/14)

Sarah T. Arai
Software Laboratory antillia.com

This is the first experiment of Image Segmentation for BUS-UC-Malignant based on the latest Tensorflow-Image-Segmentation-API, and a pre-augmented BUS-UC-Malignant-ImageMask-Dataset.zip, which was derived by us from Mendeley Data: BUS_UC

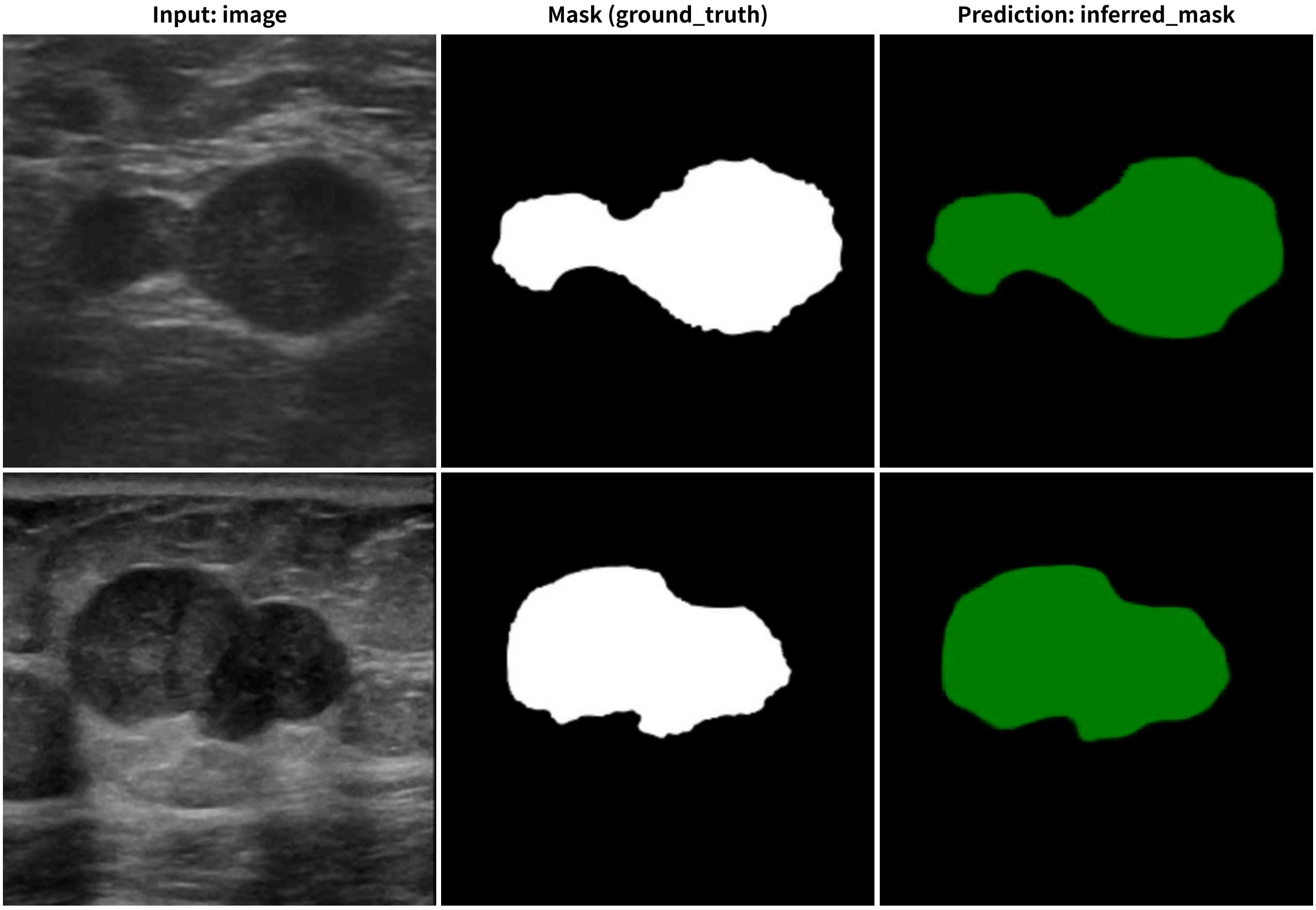
Data Augmentation Strategy:
To address the limited size of the BUS_UC,which contains 453 images and their corresponding masks in Malignant dataset, we employed an offline augmentation tool to generate a 512x512 pixels pre-augmented dataset, which supports the following augmentation methods.

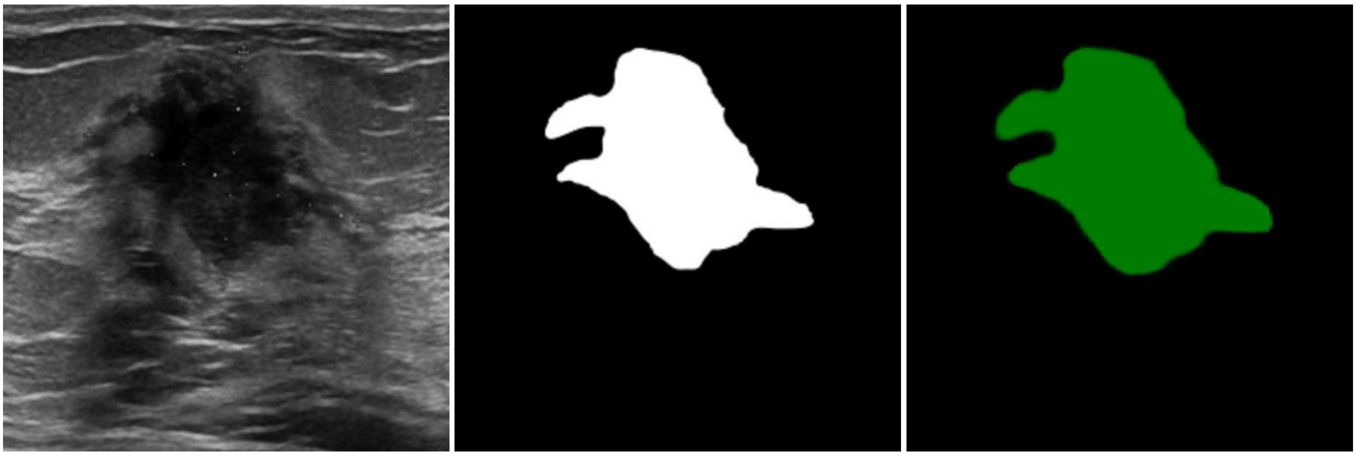
- Vertical flip
- Horizontal flip
- Rotation
- Shrinks
- Shears
- Deformation
- Distortion
- Barrel distortion
- Pincushion distortion

Please see also the following tools

- Image-Deformation-Tool
- Image-Distortion-Tool
- Barrel-Image-Distortion-Tool

Actual Image Segmentation for Images of 512x512 pixels
As shown below, the inferred masks look similar to the ground truth masks.





In this experiment, we used the simple UNet Model TensorflowSlightlyFlexibleUNet for this BUS-UC-Malignant Segmentation Model.
As shown in Tensorflow-Image-Segmentation-API. you may try other Tensorflow UNet Models:

- TensorflowSwinUNet.py
- TensorflowMultiResUNet.py
- TensorflowAttentionUNet.py
- TensorflowEfficientUNet.py
- TensorflowUNet3Plus.py
- TensorflowDeepLabV3Plus.py

1. Dataset Citation

We used the following dataset in Mendeley web site
Mendeley Data: BUS_UC

Description
The BUS_UC dataset includes 358 benign tumor images and 453 malignant tumor images. The resolution of Ultrasound images is 256 × 256 pixels. All these images were obtained from the website Ultrasound Cases (ultrasoundcases.info), which does not provide ground truth images. Therefore, with the help of an experienced radiologist, benign and malignant tumor images are annotated for segmentation and classification task.

Citation
If you use this dataset, please cite :
Ahmed Iqbal, Muhammad Sharif,
"Memory-efficient transformer network with feature fusion for breast tumor segmentation and classification task ",
Engineering Applications of Artificial Intelligence, 2023.

Institutions
COMSATS Institute of Information Technology - Wah Campus

Categories
Breast Cancer, Image Segmentation, Ultrasound, Image Classification

License
CC BY 4.0

2 BUS-UC-Malignant ImageMask Dataset

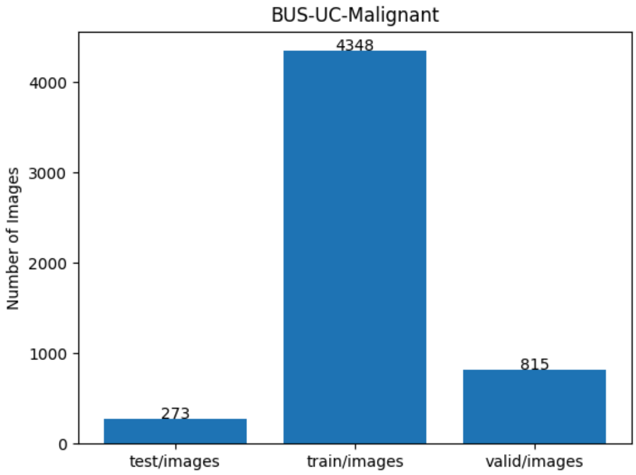
If you would like to train this BUS-UC-Malignant Segmentation model by yourself, please download the dataset from the google drive BUS-UC-Malignant-ImageMask-Dataset.zip , expand the downloaded and put it under **./dataset** folder as shown below.

```
./dataset
└─BUS-UC-Malignant
    ├─test
    │  ├─images
    │  └─masks
    ├─train
    │  ├─images
    │  └─masks
    └─valid
       ├─images
       └─masks
```

On the derivation of this datata, please refer to the following Python scripts:

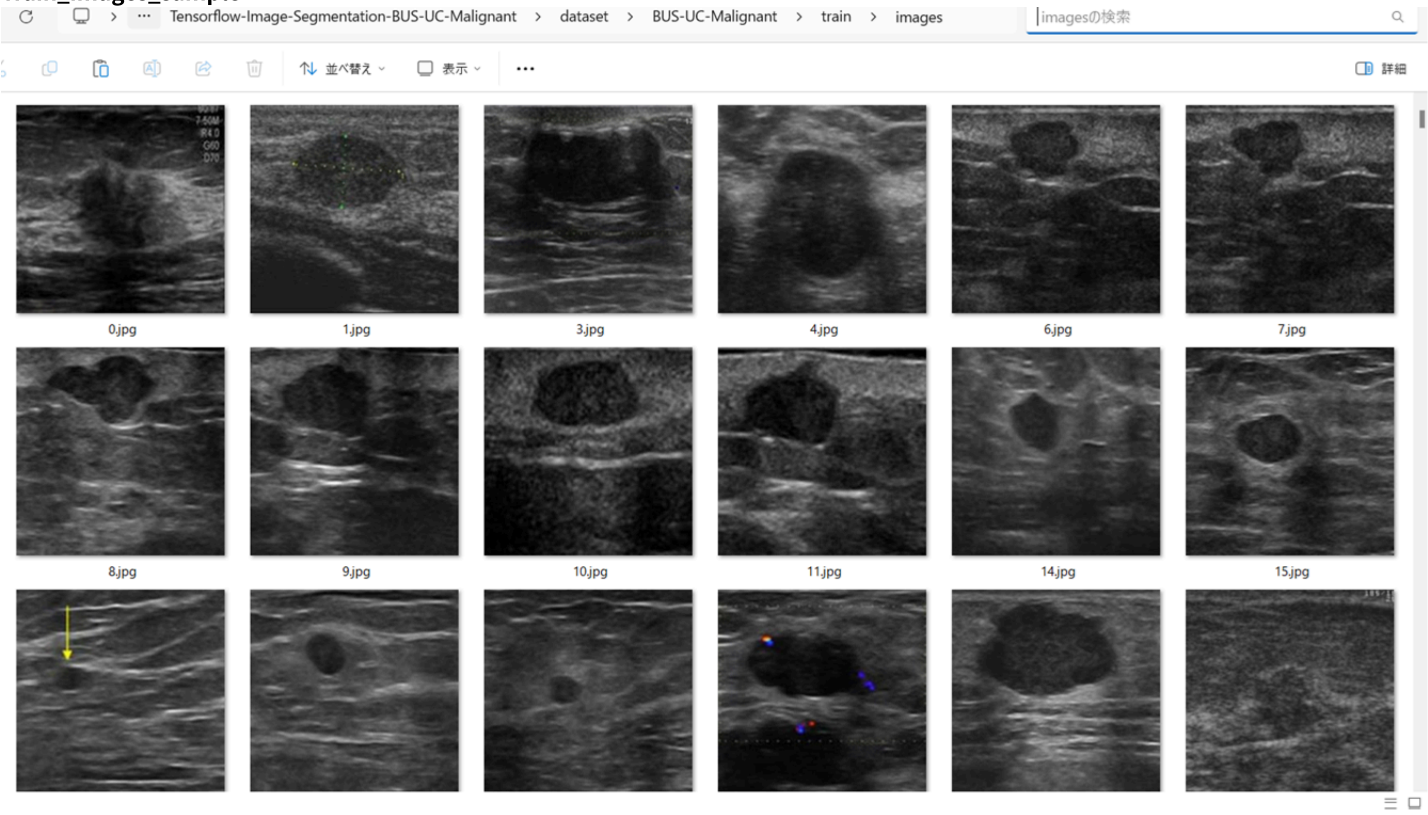
- ImageMaskDatasetGenerator.py
- split_master

BUS-UC-Malignant Dataset Statistics

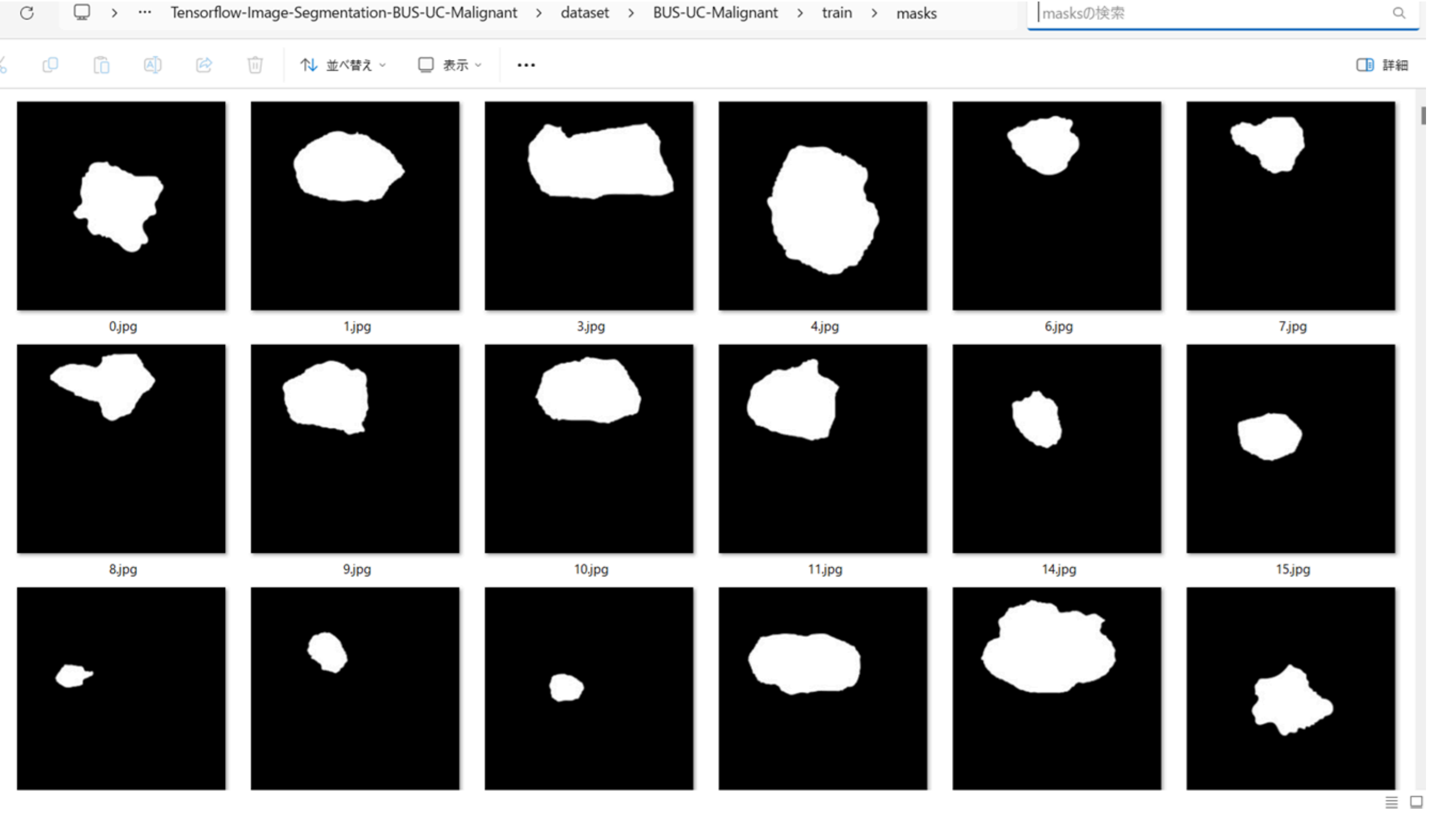


As shown above, the number of images of train and valid datasets is enough to use for a training set of our segmentation model.

Train_images_sample


Tensorflow-Image-Segmentation-BUS-UC-Malignant › dataset › BUS-UC-Malignant › train › images
imagesの検索
並べ替え
表示
詳細
0.jpg
1.jpg
3.jpg
4.jpg
6.jpg
7.jpg
8.jpg
9.jpg
10.jpg
11.jpg
14.jpg
15.jpg

Train_masks_sample


Tensorflow-Image-Segmentation-BUS-UC-Malignant > dataset > BUS-UC-Malignant > train > masks
masksの検索
並べ替え
表示
詳細
0.jpg
1.jpg
3.jpg
4.jpg
6.jpg
7.jpg
8.jpg
9.jpg
10.jpg
11.jpg
14.jpg
15.jpg

3 Train TensorflowUNet Model

We have trained BUS-UC-MalignantTensorflowUNet Model by using the following **train_eval_infer.config** file.
Please move to ./projects/TensorflowSlightlyFlexibleUNet/BUS-UC-Malignant and run the following bat file.

```
>1.train.bat
```

, which simply runs the following command.

```
>python ../../../src/TensorflowUNetTrainer.py ./train_eval_infer.config
```

Model parameters
Defined a small **base_filters** and large **base_kernels** for the first Conv Layer of Encoder Block of TensorflowUNet.py and a large num_layers (including a bridge between Encoder and Decoder Blocks).

```
[model]
base_filters   = 16
base_kernels   = (9,9)
num_layers     = 8
dilation       = (3,3)
```

Learning rate
Defined a small learning rate.

```
[model]
learning_rate  = 0.0001
```

Online augmentation
Disabled our online augmentation.

```
[model]
model         = "TensorflowUNet"
generator     = False
```

Loss and metrics functions
Specified "bce_dice_loss" and "dice_coef".

```
[model]
loss           = "bce_dice_loss"
metrics        = ["dice_coef"]
```

Learning rate reducer callback
Enabled learing_rate_reducer callback, and a small reducer_patience.

```
[train]
learning_rate_reducer = True
reducer_factor     = 0.4
reducer_patience   = 4
```

Early stopping callback

Enabled early stopping callback with patience parameter.

```
[train]
patience      = 10
```

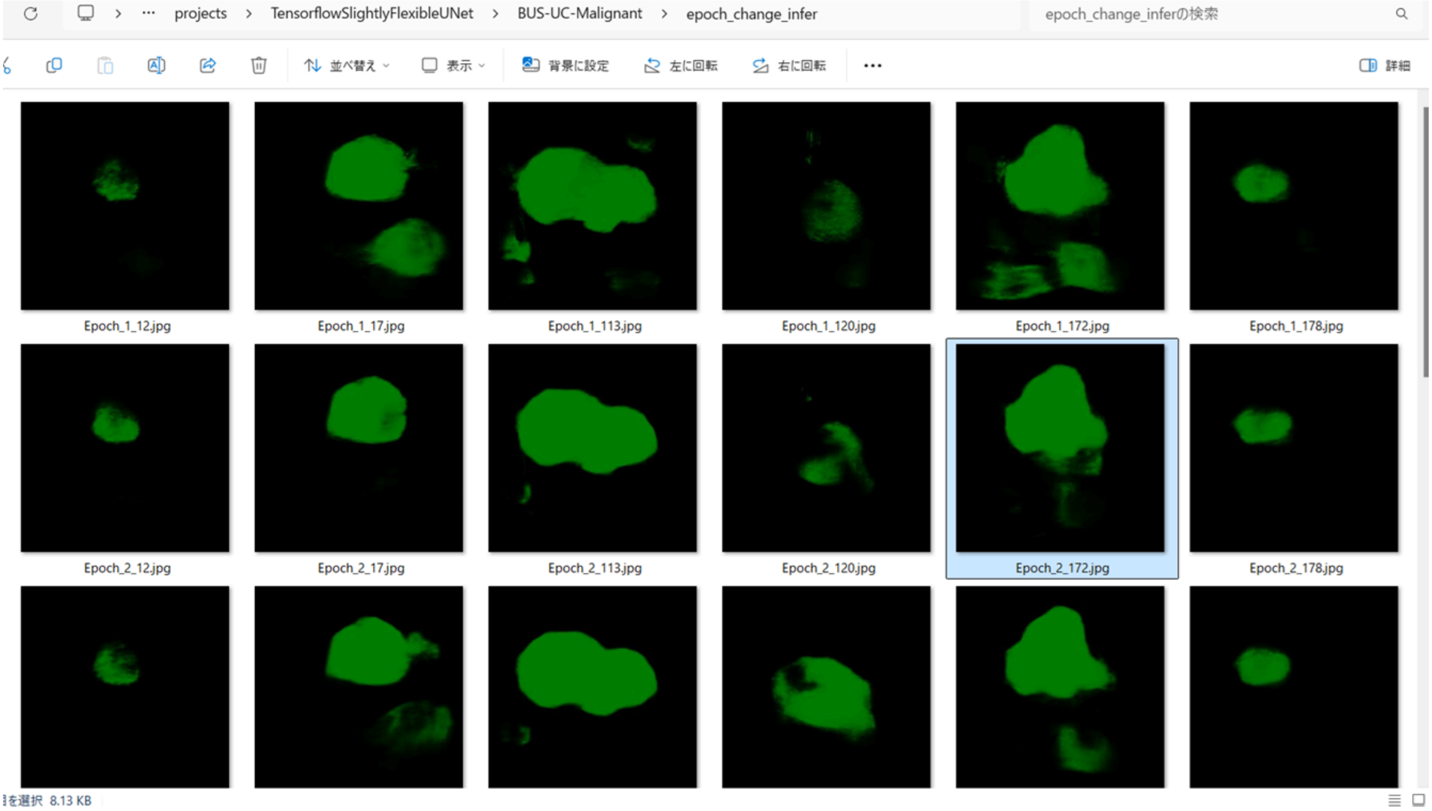
Epoch change inference callbacks

Enabled epoch_change_infer callback.

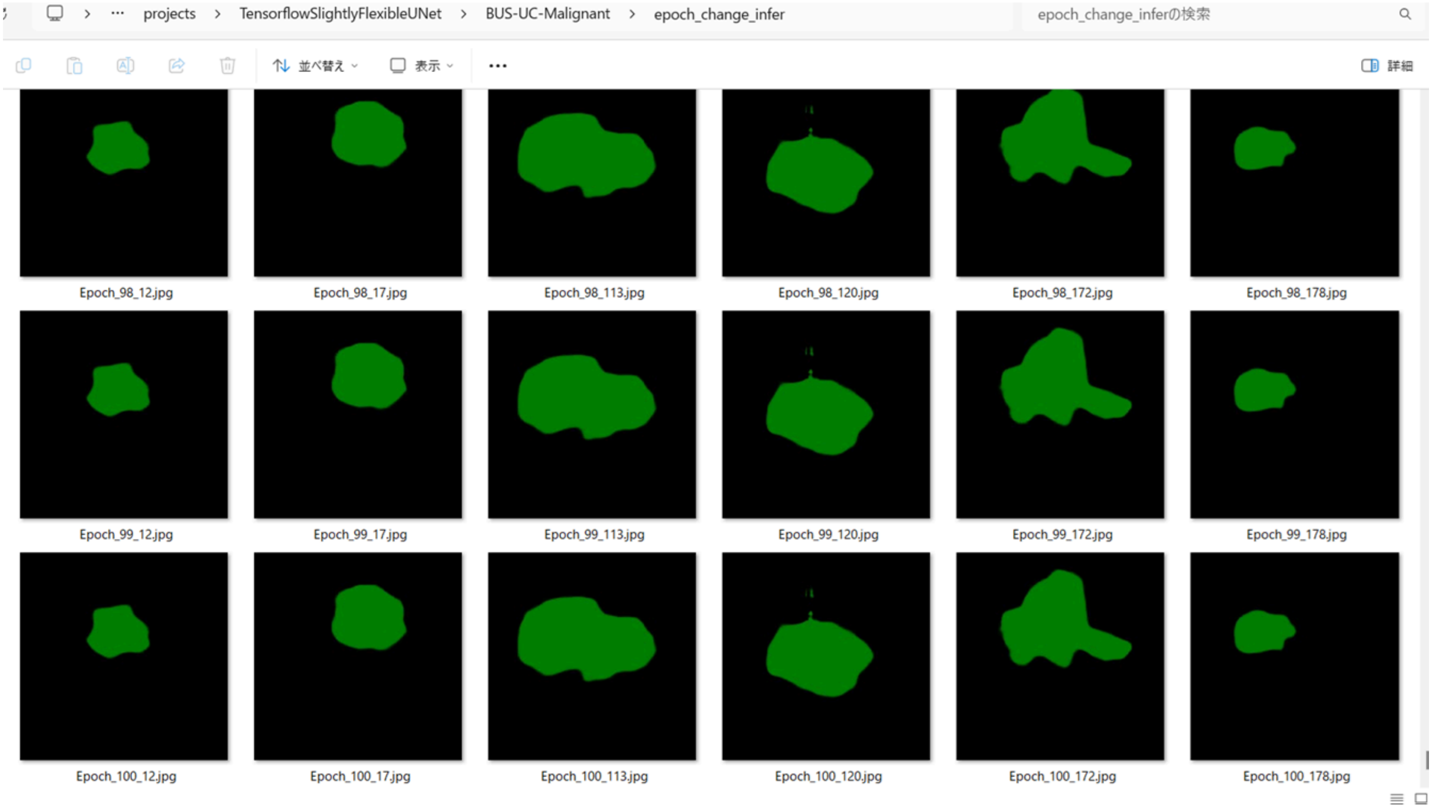
```
[train]
epoch_change_infer       = True
epoch_change_infer_dir   =  ”./epoch_change_infer”
epoch_changeinfer        = False
epoch_changeinfer_dir    = ”./epoch_changeinfer”
num_infer_images         = 6
```

By using this callback, on every epoch_change, the inference procedure can be called for 6 images in **mini_test** folder. This will help you confirm how the predicted mask changes at each epoch during your training process.

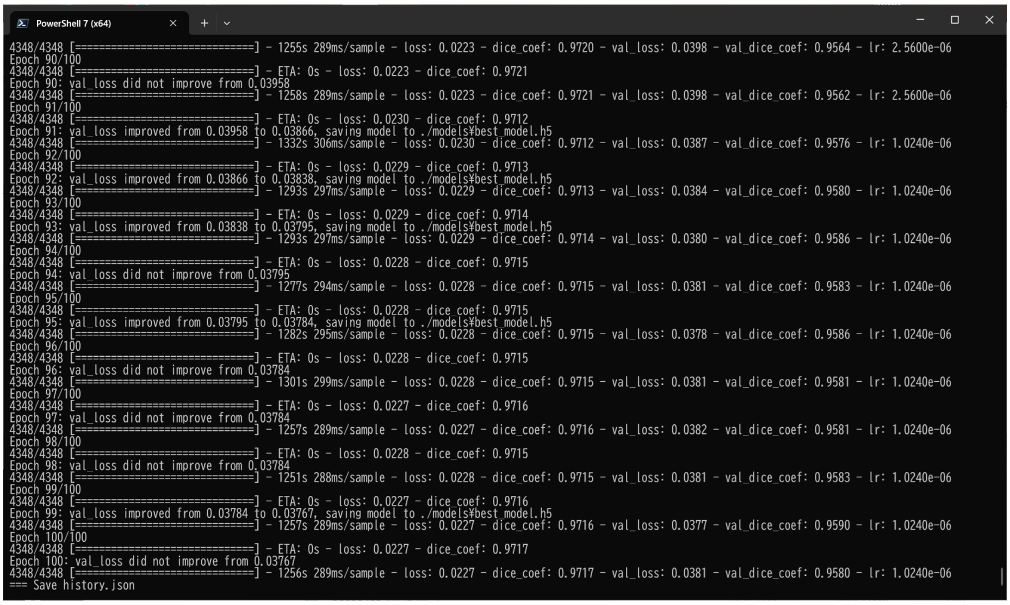
Epoch_change_inference output at start (epoch 1,2,3)



Epoch_change_inference output at end (epoch 98,99,100)

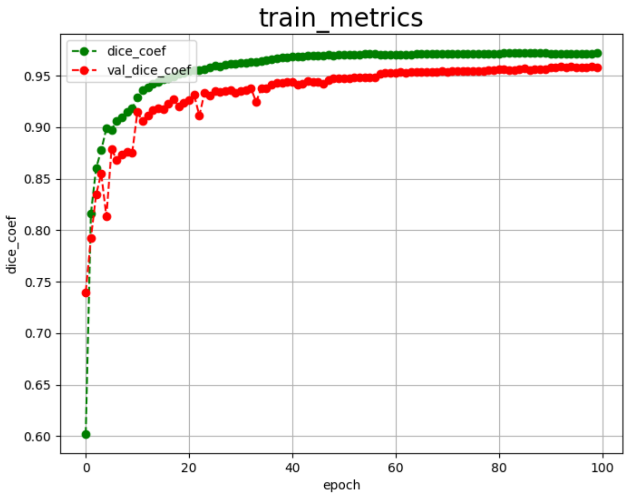


In this experiment, the training process was terminated at epoch 100.

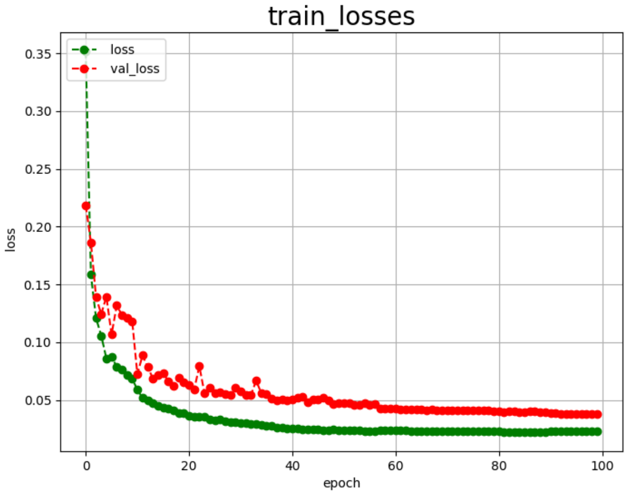


```
4348/4348 [==============================] - 1255s 289ms/sample - loss: 0.0223 - dice_coef: 0.9720 - val_loss: 0.0398 - val_dice_coef: 0.9564 - lr: 2.5600e-06
Epoch 90/100
4348/4348 [==============================] - ETA: 0s - loss: 0.0223 - dice_coef: 0.9721
Epoch 90: val_loss did not improve from 0.03958
4348/4348 [==============================] - 1258s 289ms/sample - loss: 0.0223 - dice_coef: 0.9721 - val_loss: 0.0398 - val_dice_coef: 0.9562 - lr: 2.5600e-06
Epoch 91/100
4348/4348 [==============================] - ETA: 0s - loss: 0.0230 - dice_coef: 0.9712
Epoch 91: val_loss improved from 0.03958 to 0.03866, saving model to ./models¥best_model.h5
4348/4348 [==============================] - 1332s 306ms/sample - loss: 0.0230 - dice_coef: 0.9712 - val_loss: 0.0387 - val_dice_coef: 0.9576 - lr: 1.0240e-06
Epoch 92/100
4348/4348 [==============================] - ETA: 0s - loss: 0.0229 - dice_coef: 0.9713
Epoch 92: val_loss improved from 0.03866 to 0.03838, saving model to ./models¥best_model.h5
4348/4348 [==============================] - 1293s 297ms/sample - loss: 0.0229 - dice_coef: 0.9713 - val_loss: 0.0384 - val_dice_coef: 0.9580 - lr: 1.0240e-06
Epoch 93/100
4348/4348 [==============================] - ETA: 0s - loss: 0.0229 - dice_coef: 0.9714
Epoch 93: val_loss improved from 0.03838 to 0.03795, saving model to ./models¥best_model.h5
4348/4348 [==============================] - 1293s 297ms/sample - loss: 0.0229 - dice_coef: 0.9714 - val_loss: 0.0380 - val_dice_coef: 0.9586 - lr: 1.0240e-06
Epoch 94/100
4348/4348 [==============================] - ETA: 0s - loss: 0.0228 - dice_coef: 0.9715
Epoch 94: val_loss did not improve from 0.03795
4348/4348 [==============================] - 1277s 294ms/sample - loss: 0.0228 - dice_coef: 0.9715 - val_loss: 0.0381 - val_dice_coef: 0.9583 - lr: 1.0240e-06
Epoch 95/100
4348/4348 [==============================] - ETA: 0s - loss: 0.0228 - dice_coef: 0.9715
Epoch 95: val_loss improved from 0.03795 to 0.03784, saving model to ./models¥best_model.h5
4348/4348 [==============================] - 1282s 295ms/sample - loss: 0.0228 - dice_coef: 0.9715 - val_loss: 0.0378 - val_dice_coef: 0.9586 - lr: 1.0240e-06
Epoch 96/100
4348/4348 [==============================] - ETA: 0s - loss: 0.0228 - dice_coef: 0.9715
Epoch 96: val_loss did not improve from 0.03784
4348/4348 [==============================] - 1301s 299ms/sample - loss: 0.0228 - dice_coef: 0.9715 - val_loss: 0.0381 - val_dice_coef: 0.9581 - lr: 1.0240e-06
Epoch 97/100
4348/4348 [==============================] - ETA: 0s - loss: 0.0227 - dice_coef: 0.9716
Epoch 97: val_loss did not improve from 0.03784
4348/4348 [==============================] - 1257s 289ms/sample - loss: 0.0227 - dice_coef: 0.9716 - val_loss: 0.0382 - val_dice_coef: 0.9581 - lr: 1.0240e-06
Epoch 98/100
4348/4348 [==============================] - ETA: 0s - loss: 0.0228 - dice_coef: 0.9715
Epoch 98: val_loss did not improve from 0.03784
4348/4348 [==============================] - 1251s 288ms/sample - loss: 0.0228 - dice_coef: 0.9715 - val_loss: 0.0381 - val_dice_coef: 0.9583 - lr: 1.0240e-06
Epoch 99/100
4348/4348 [==============================] - ETA: 0s - loss: 0.0227 - dice_coef: 0.9716
Epoch 99: val_loss improved from 0.03784 to 0.03767, saving model to ./models¥best_model.h5
4348/4348 [==============================] - 1257s 289ms/sample - loss: 0.0227 - dice_coef: 0.9716 - val_loss: 0.0377 - val_dice_coef: 0.9590 - lr: 1.0240e-06
Epoch 100/100
4348/4348 [==============================] - ETA: 0s - loss: 0.0227 - dice_coef: 0.9717
Epoch 100: val_loss did not improve from 0.03767
4348/4348 [==============================] - 1256s 289ms/sample - loss: 0.0227 - dice_coef: 0.9717 - val_loss: 0.0381 - val_dice_coef: 0.9580 - lr: 1.0240e-06
=== Save history.json
```

train_metrics.csv



train_losses.csv



4 Evaluation

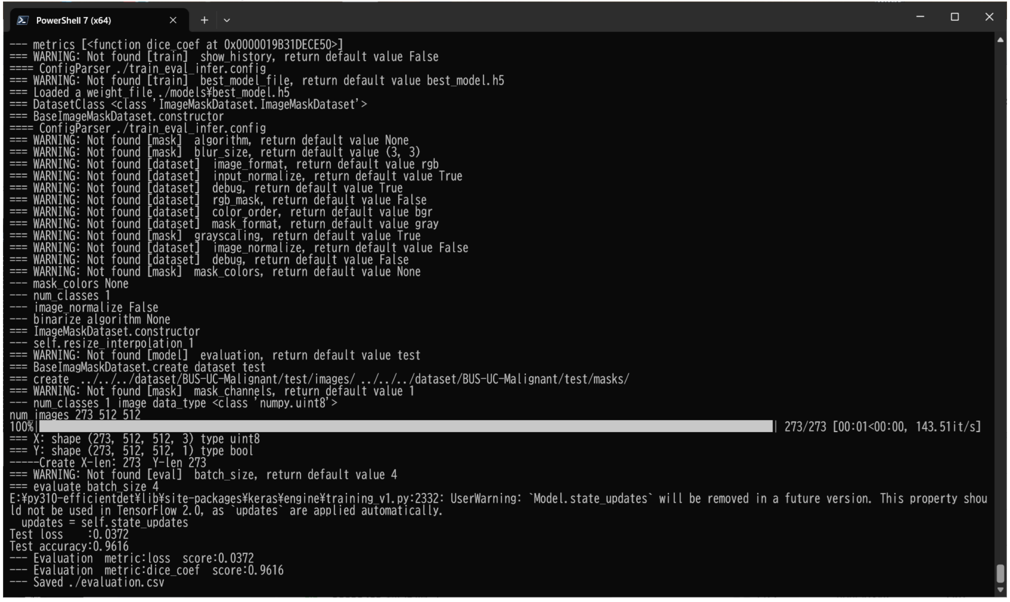
Please move to a **./projects/TensorflowSlightlyFlexibleUNet/BUS-UC-Malignant** folder, and run the following bat file to evaluate TensorflowUNet model for BUS-UC-Malignant.

```
./2.evaluate.bat
```

This bat file simply runs the following command.

```
python ../../../src/TensorflowUNetEvaluator.py ./train_eval_infer_aug.config
```

Evaluation console output:



```
--- metrics [<function dice_coef at 0x0000019B31DECE50>]
=== WARNING: Not found [train]  show_history, return default value False
==== ConfigParser ./train_eval_infer.config
=== WARNING: Not found [train]  best_model_file, return default value best_model.h5
=== Loaded a weight_file ./models¥best_model.h5
=== DatasetClass <class 'ImageMaskDataset.ImageMaskDataset'>
=== BaseImageMaskDataset.constructor
==== ConfigParser ./train_eval_infer.config
=== WARNING: Not found [mask]  algorithm, return default value None
=== WARNING: Not found [mask]  blur_size, return default value (3, 3)
=== WARNING: Not found [dataset]  image_format, return default value rgb
=== WARNING: Not found [dataset]  input_normalize, return default value True
=== WARNING: Not found [dataset]  debug, return default value True
=== WARNING: Not found [dataset]  rgb_mask, return default value False
=== WARNING: Not found [dataset]  color_order, return default value bgr
=== WARNING: Not found [dataset]  mask_format, return default value gray
=== WARNING: Not found [mask]  grayscaling, return default value True
=== WARNING: Not found [dataset]  image_normalize, return default value False
=== WARNING: Not found [dataset]  debug, return default value False
=== WARNING: Not found [mask]  mask_colors, return default value None
--- mask_colors None
--- num_classes 1
--- image_normalize False
--- binarize algorithm None
=== ImageMaskDataset.constructor
--- self.resize_interpolation 1
=== WARNING: Not found [model]  evaluation, return default value test
=== BaseImagMaskDataset.create dataset test
=== create  ../../../dataset/BUS-UC-Malignant/test/images/ ../../../dataset/BUS-UC-Malignant/test/masks/
=== WARNING: Not found [mask]  mask_channels, return default value 1
--- num_classes 1 image data_type <class 'numpy.uint8'>
num_images 273 512 512
100%|██████████████████████████████████████████████████| 273/273 [00:01<00:00, 143.51it/s]
=== X: shape (273, 512, 512, 3) type uint8
=== Y: shape (273, 512, 512, 1) type bool
-----Create X-len: 273  Y-len 273
=== WARNING: Not found [eval]  batch_size, return default value 4
=== evaluate batch_size 4
E:¥py310-efficientdet¥lib¥site-packages¥keras¥engine¥training_v1.py:2332: UserWarning: `Model.state_updates` will be removed in a future version. This property shou
ld not be used in TensorFlow 2.0, as `updates` are applied automatically.
  updates = self.state_updates
Test loss     :0.0372
Test accuracy:0.9616
--- Evaluation  metric:loss  score:0.0372
--- Evaluation  metric:dice_coef  score:0.9616
--- Saved ./evaluation.csv
```

Image-Segmentation-BUS-UC-Malignant evaluation.csv

The loss (bce_dice_loss) to this BUS-UC-Malignant/test was low, and dice_coef very high as shown below.

```
loss,0.0372
dice_coef,0.9616
```

5 Inference

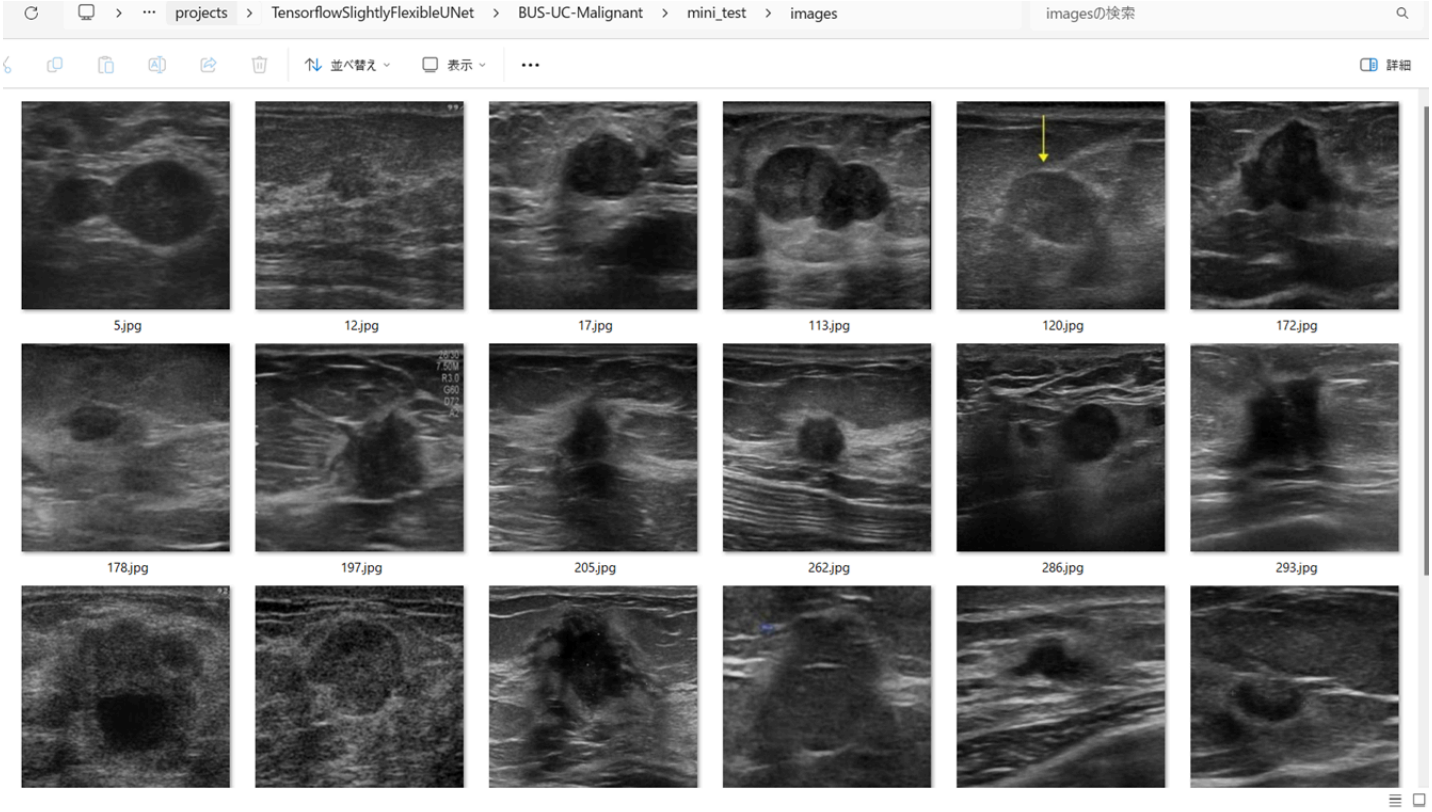
Please move to a **./projects/TensorflowSlightlyFlexibleUNet/BUS-UC-Malignant** folder
,and run the following bat file to infer segmentation regions for images by the Trained-TensorflowUNet model for BUS-UC-Malignant.

```
./3.infer.bat
```

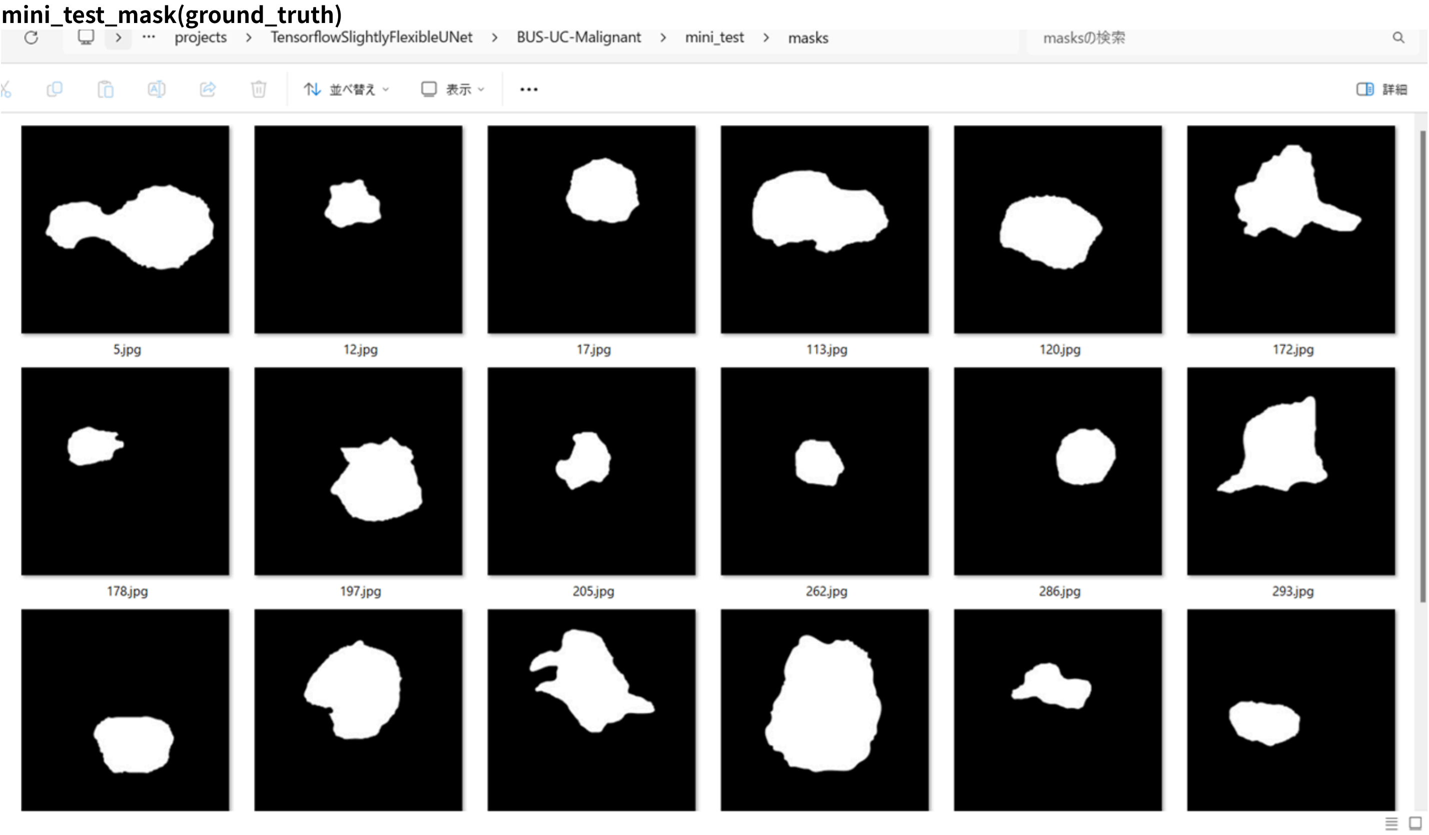
This simply runs the following command.

```
python ../../../src/TensorflowUNetInferencer.py ./train_eval_infer_aug.config
```

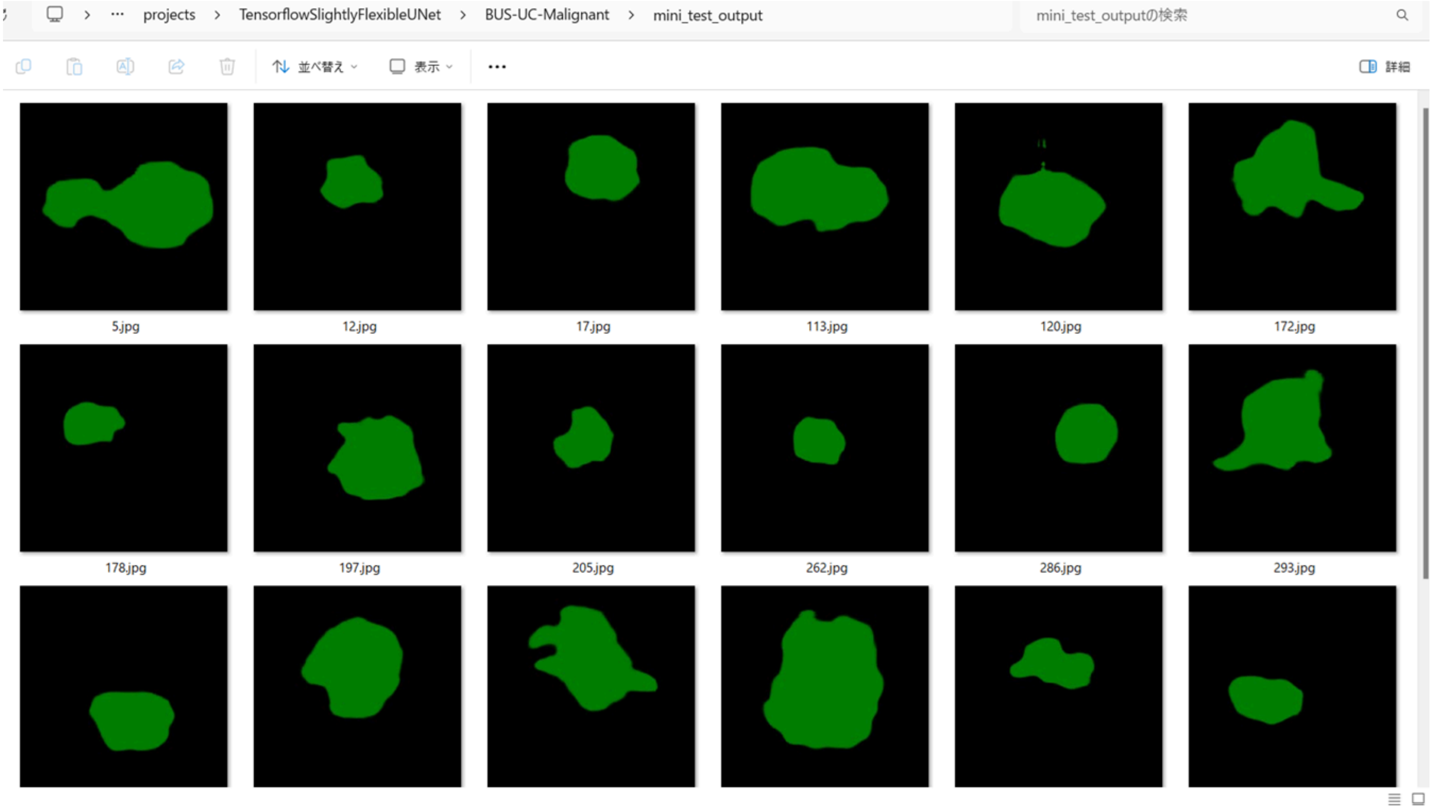
mini_test_images



mini_test_mask(ground_truth)

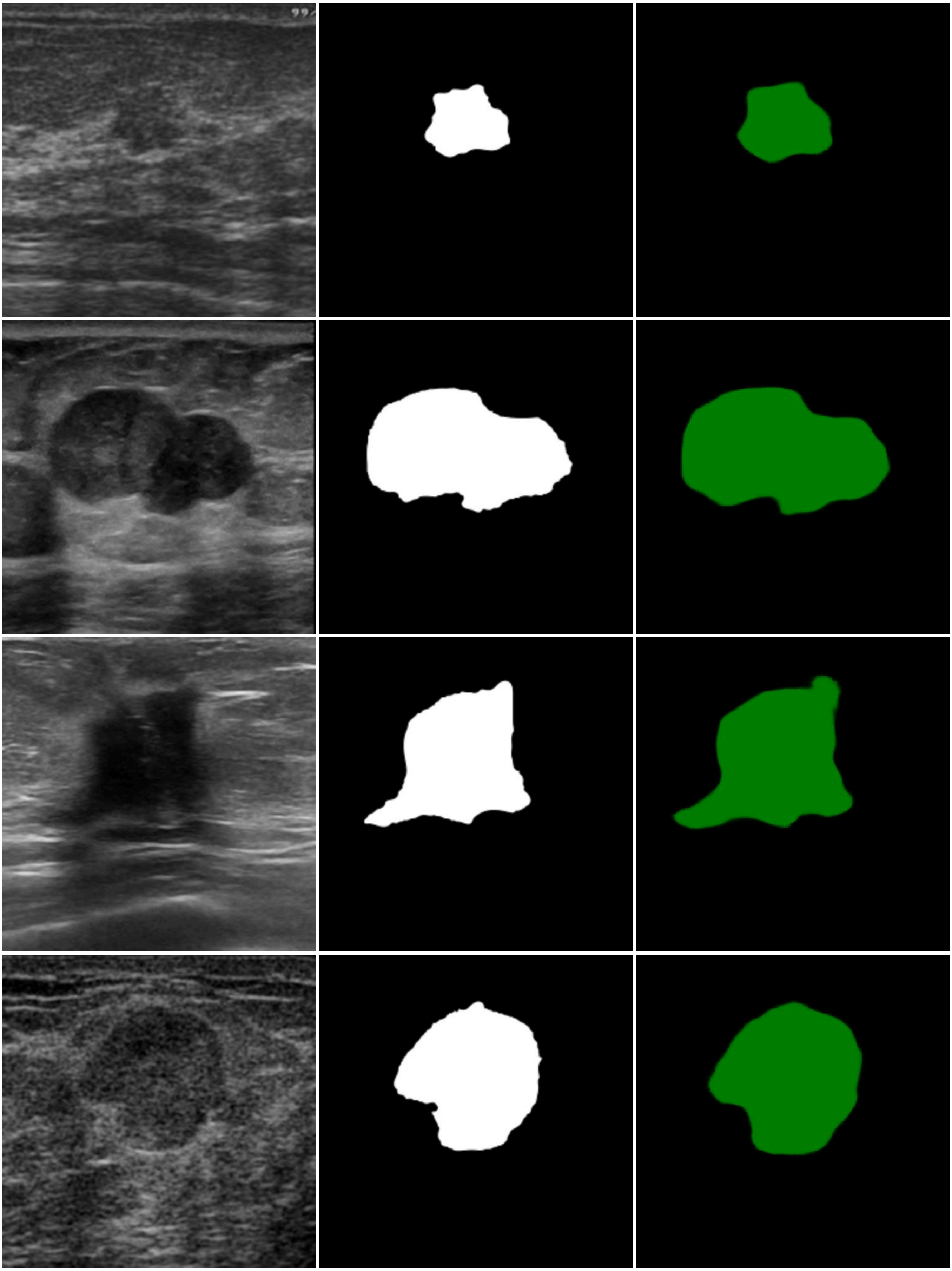


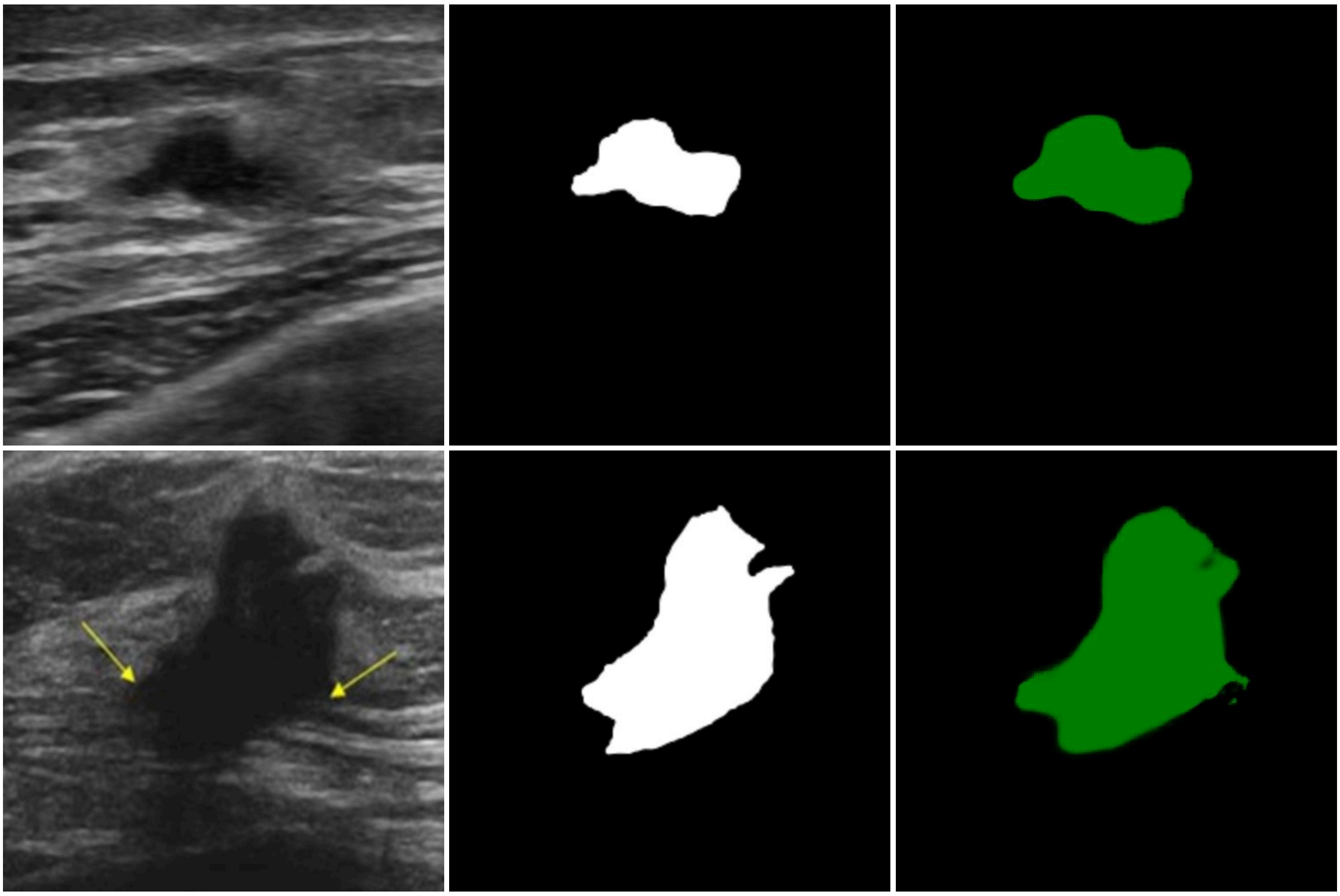
Inferred test masks



Enlarged images and masks

Image **Mask (ground_truth)** **Inferred-mask**





References

1. BUS_UC - Breast Ultrasound Dataset

https://www.kaggle.com/datasets/orvile/bus-uc-breast-ultrasound/code