

TensorFlow-FlexUNet-Image-Segmentation-BreastDM-DCE-MRI (2025/09/05)

Toshiyuki A. Arai: Software Laboratory @antillia.com

This is the first experiment of Image Segmentation for BreastDM-DCE(Dynamic Contrast-Enhanced)-MRI (Benign and Malignant) based on our TensorFlow FlexUNet (TensorFlow Flexible UNet Image Segmentation Model for Multiclass) and a 512x512 pixels [BreastDM-ImageMask-Dataset.zip](#) with colorized masks (benign:green, malignant:red), which was derived by us from the following dataset on the google drive [BreaDM.zip](#) specified in a repository [Breast-cancer-dataset](#)

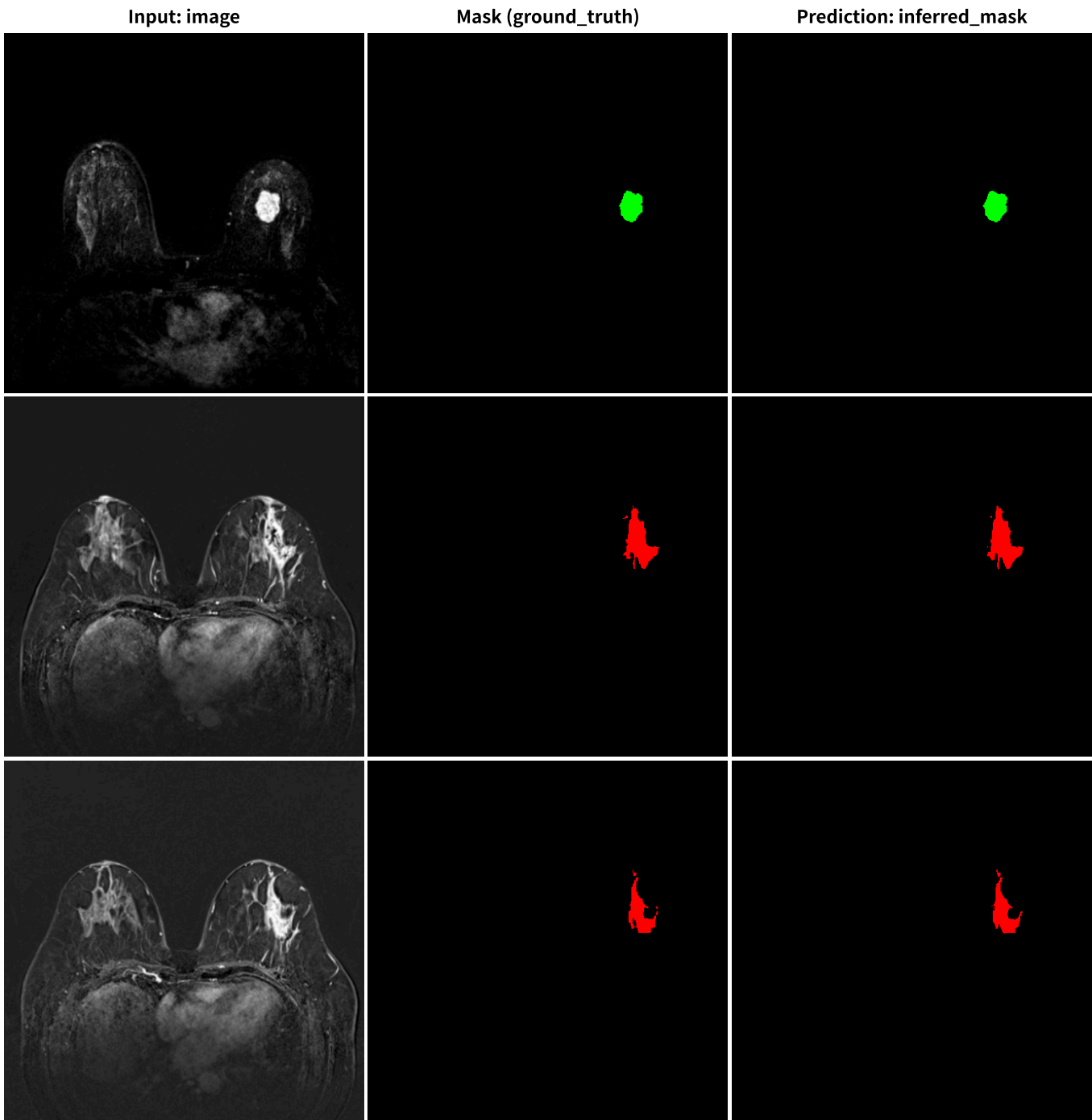
On BreaDM dataset, please refer to [BreastDM: A DCE-MRI dataset for breast tumor image segmentation and classification](#) , and [Breast-cancer-dataset](#)

Please see also our experiment for a singleclass segmentation model [Tensorflow-Image-Segmentation-Malignant-BreastDM](#)

Acutual Image Segmentation for 512x512 BreastDM images

As shown below, the inferred masks predicted by our segmentation model trained on the PNG dataset appear similar to the ground truth masks.

rgb_map = (benign:green, malignant:red)



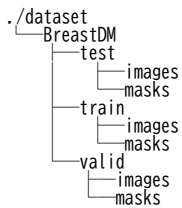
1. Dataset Citation

The dataset used here has been obtained from the google drive [BreaDM.zip](#) specified in [Breast-cancer-dataset](#)

On BreastDM dataset, please refer to [BreastDM: A DCE-MRI dataset for breast tumor image segmentation and classification](#)
 Xiaoming Zhao, Yuehui Liao, Jiahao Xie, Xiaxia He, Shiqing Zhang, Guoyu Wang
 , Jiangxiong Fang , Hongsheng Lu, Jun Yu
<https://doi.org/10.1016/j.compbimed.2023.107255>

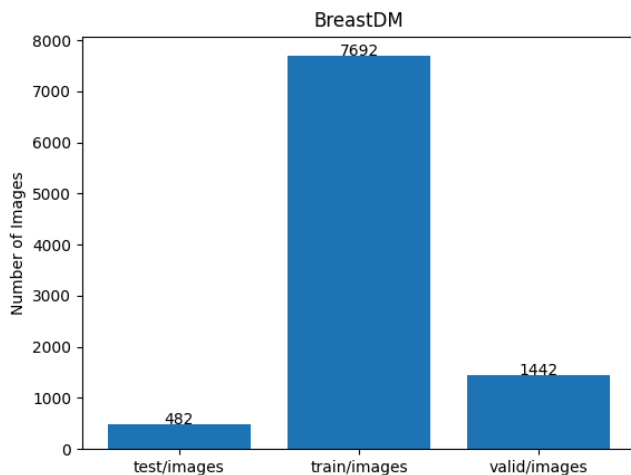
2 BreastDM ImageMask Dataset

If you would like to train this BreastDM Segmentation model by yourself, please download the dataset from the google drive [BreastDM-ImageMask-Dataset.zip](#).
 , expand the downloaded ImageMaskDataset and put it under **./dataset** folder to be



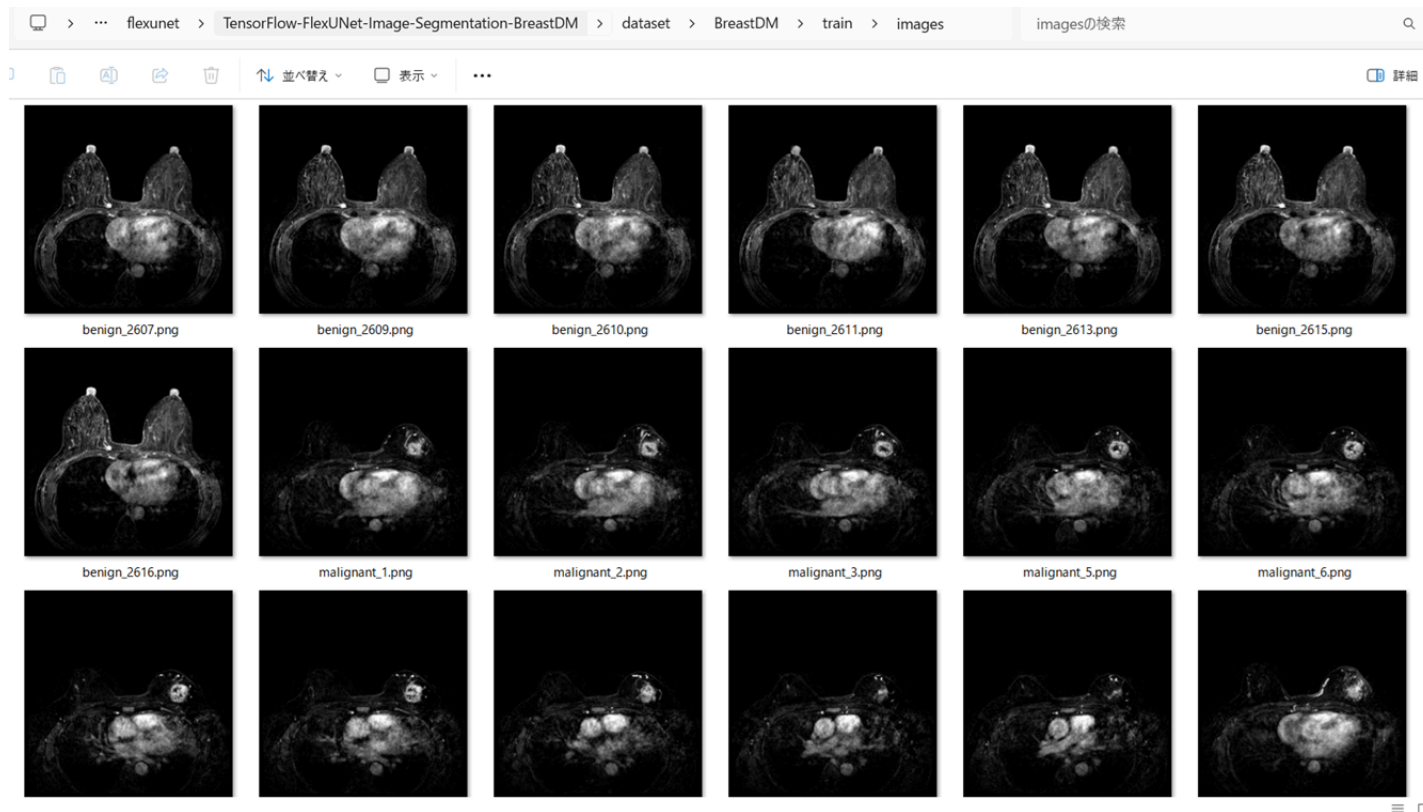
On the derivation of this dataset, please refer to [TrainMalignantImageMaskDatasetGenerator.py](#) in our repository [Tensorflow-Image-Segmentation-Malignant-BreastDM](#)

BreastDM Statistics

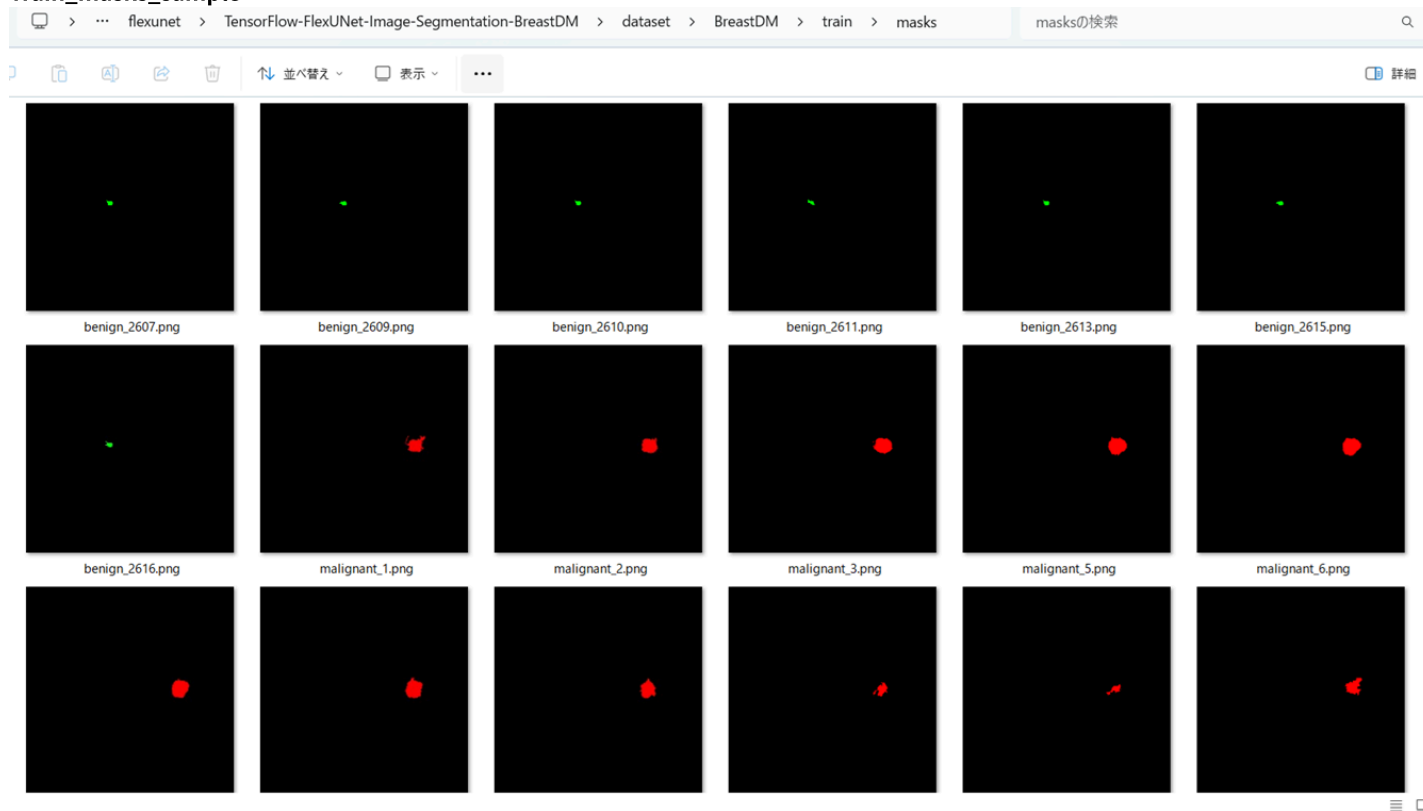


As shown above, the number of images of train and valid datasets is large enough to use for a training set of our segmentation model.

Train_images_sample



Train_masks_sample



3 Train TensorFlowFlexUNet Model

We trained BreastDM TensorFlowFlexUNet Model by using the following [train_eval_infer.config](#) file.
Please move to `./projects/TensorFlowFlexUNet/BreastDM` and run the following bat file.

>1. train.bat

, which simply runs the following command.

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

Model parameters

Defined a small **base_filters = 16** and large **base_kernels = (7,7)** for the first Conv Layer of Encoder Block of [TensorFlowFlexUNet.py](#) and a large num_layers (including a bridge between Encoder and Decoder Blocks).

```
[model]
;You may specify your own UNet class derived from our TensorFlowFlexModel
model          = "TensorFlowFlexUNet"
generator      = False
image_width    = 512
image_height   = 512
image_channels  = 3
num_classes    = 3

base_filters   = 16
base_kernels   = (7, 7)
num_layers     = 8
dropout_rate   = 0.05
dilation       = (1, 1)
```

Learning rate

Defined a very small learning rate.

```
[model]
learning_rate = 0.00007
```

Loss and metrics functions

Specified "categorical_crossentropy" and "[dice_coef_multiclass](#)".

```
[model]
loss          = "categorical_crossentropy"
metrics       = ["dice_coef_multiclass"]
```

Dataset class

Specified [ImageCategorizedMaskDataset](#) class.

```
[dataset]
class_name    = "ImageCategorizedMaskDataset"
```

Learning rate reducer callback

Enabled learning_rate_reducer callback, and a small reducer_patience.

```
[train]
learning_rate_reducer = True
reducer_factor        = 0.5
reducer_patience     = 4
```

Early stopping callback

Enabled early stopping callback with patience parameter.

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

RGB Color map

rgb color map dict for BreastDM 1+2 classes.

```
[mask]
mask_file_format = ".png"

; RGB colors    benign:green, malignant:red
rgb_map = {(0, 0, 0):0, (0, 255, 0):1, (255, 0, 0):2, }
```

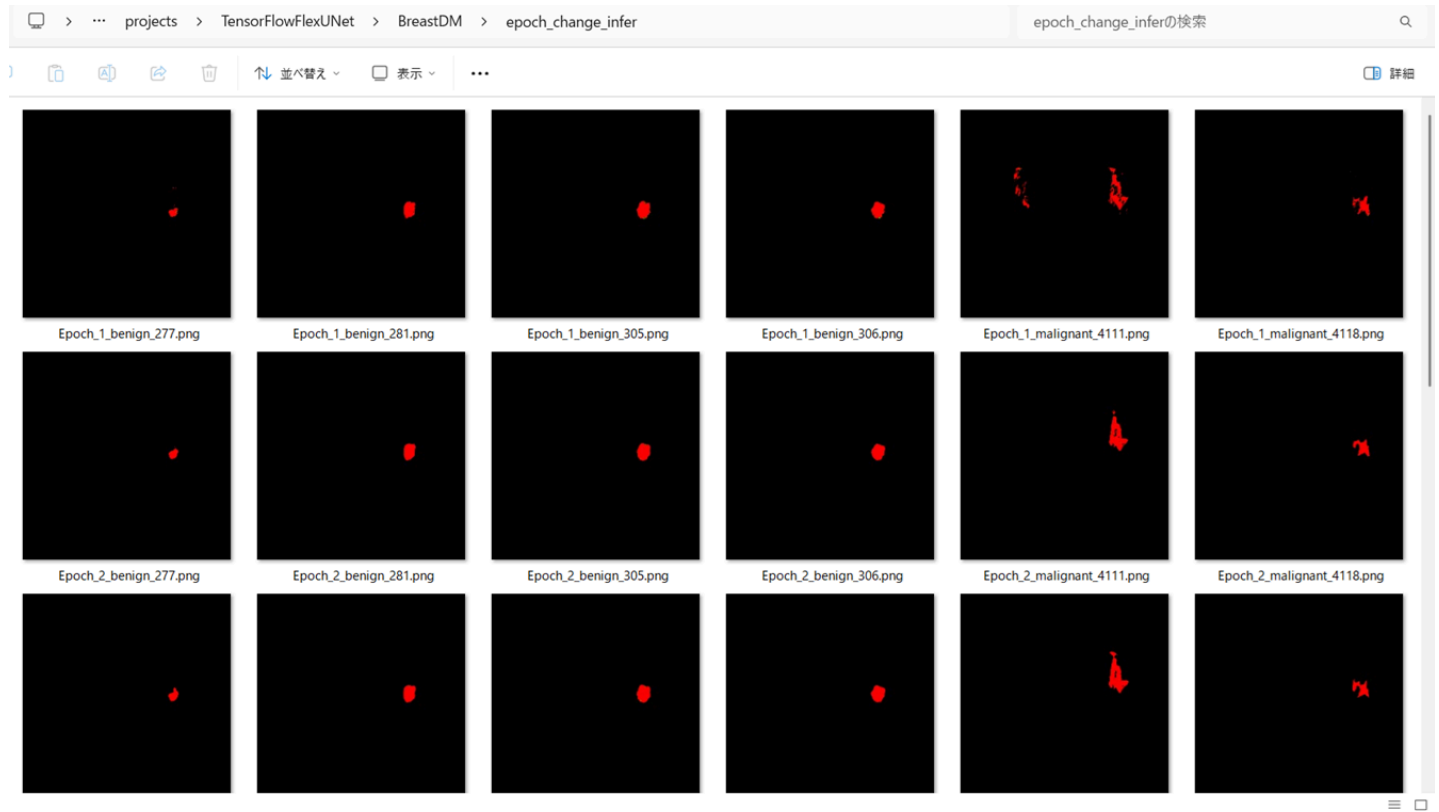
Epoch change inference callback

Enabled [epoch_change_infer callback \(EpochChangeInferencer.py\)](#).

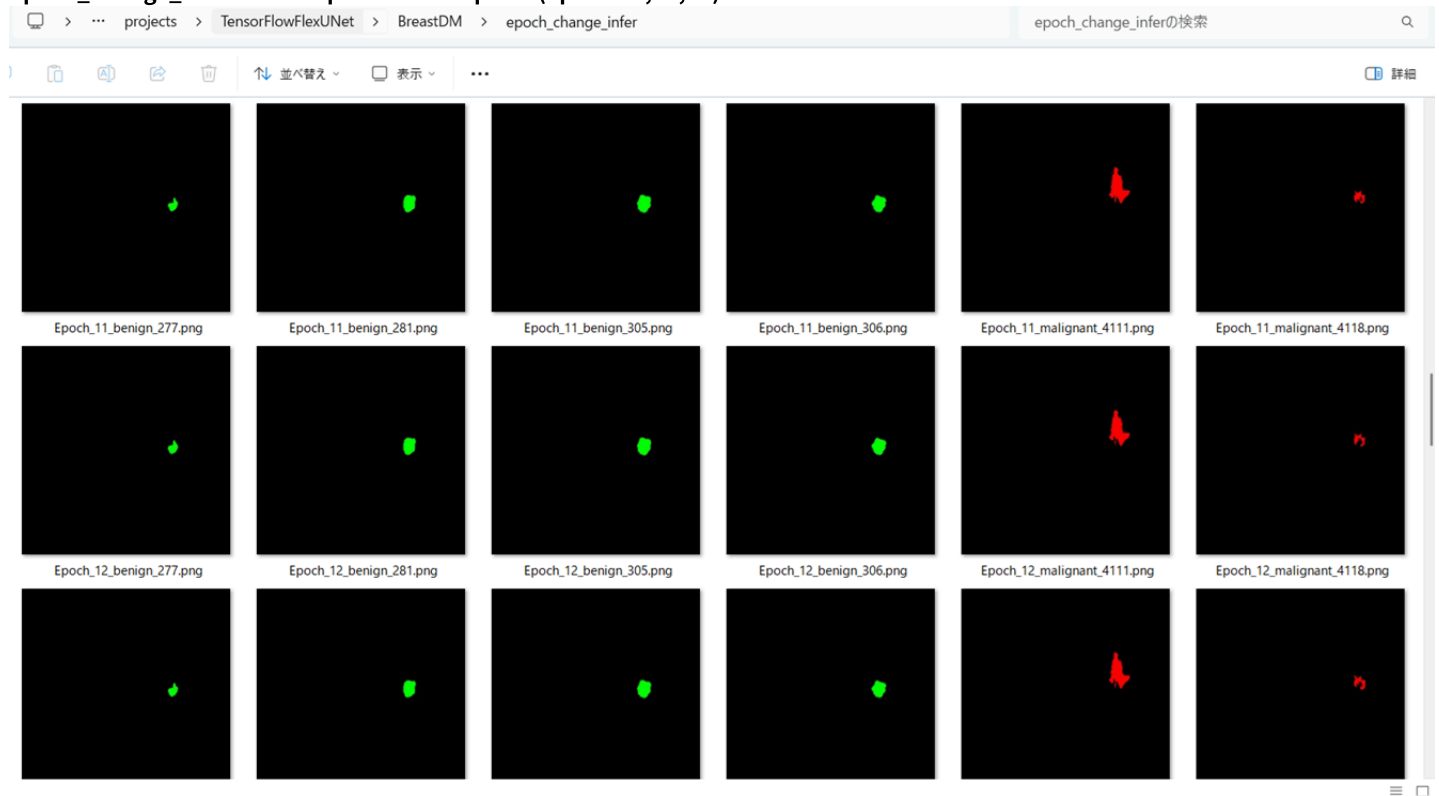
```
[train]
epoch_change_infer      = True
epoch_change_infer_dir  = "./epoch_change_infer"
num_infer_images        = 6
```

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.

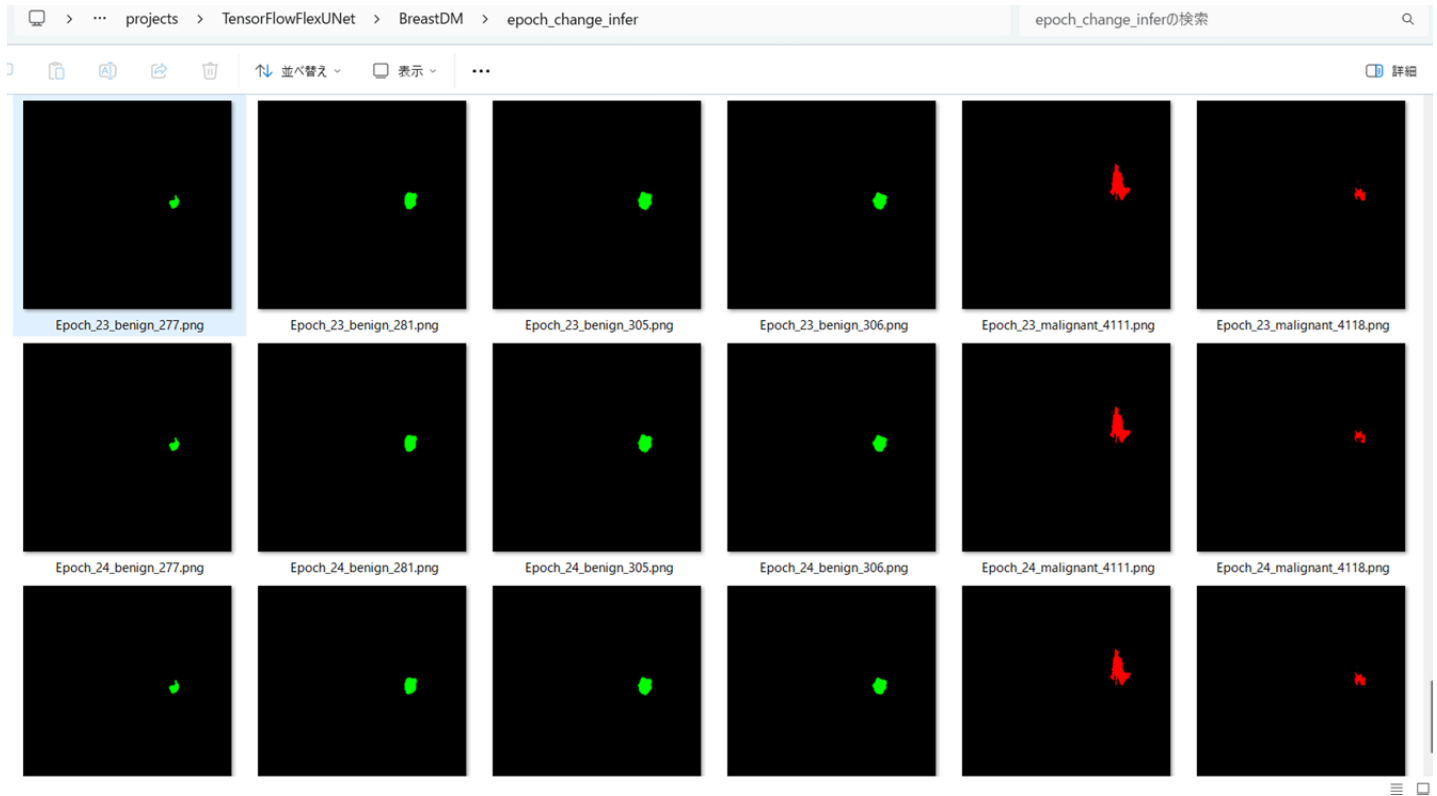
Epoch_change_inference output at starting (epoch 1,2,3)



Epoch_change_infer output at midpoint (epoch 11,12,13)



Epoch_change_infer output at ending (epoch 23,24,25)



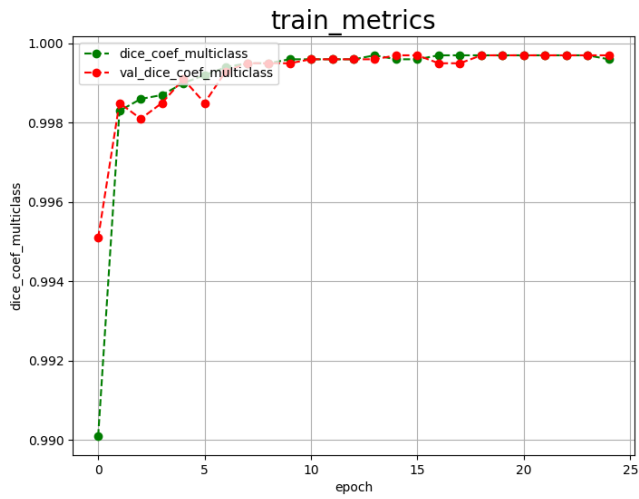
In this experiment, the training process was terminated at epoch 25.

```

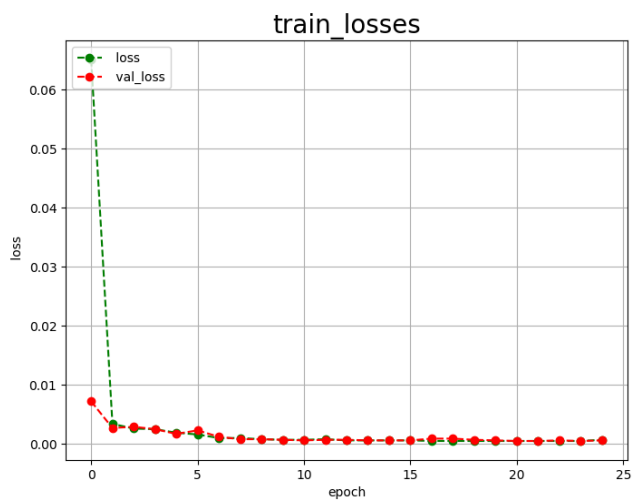
14: val_loss did not improve from 0.00064
692 [=====] - ETA: 0s - loss: 5.8444e-04 - dice_coef_multiclass: 0.9997 - val_loss: 6.4649e-04 - val_dice_coef_multiclass: 0.9996 - lr: 7.0000e-05
15/100
692 [=====] - ETA: 0s - loss: 5.9816e-04 - dice_coef_multiclass: 0.9996
15: val_loss improved from 0.00064 to 0.00062, saving model to ./models$best_model.h5
692 [=====] - ETA: 0s - loss: 5.9816e-04 - dice_coef_multiclass: 0.9996 - val_loss: 6.1548e-04 - val_dice_coef_multiclass: 0.9997 - lr: 7.0000e-05
16/100
692 [=====] - ETA: 0s - loss: 6.2839e-04 - dice_coef_multiclass: 0.9996
16: val_loss did not improve from 0.00062
692 [=====] - ETA: 0s - loss: 6.2839e-04 - dice_coef_multiclass: 0.9996 - val_loss: 6.2903e-04 - val_dice_coef_multiclass: 0.9997 - lr: 7.0000e-05
17/100
692 [=====] - ETA: 0s - loss: 5.3004e-04 - dice_coef_multiclass: 0.9997
17: val_loss did not improve from 0.00062
692 [=====] - ETA: 0s - loss: 5.3004e-04 - dice_coef_multiclass: 0.9997 - val_loss: 9.0030e-04 - val_dice_coef_multiclass: 0.9995 - lr: 7.0000e-05
18/100
692 [=====] - ETA: 0s - loss: 5.4712e-04 - dice_coef_multiclass: 0.9997
18: val_loss did not improve from 0.00062
692 [=====] - ETA: 0s - loss: 5.4712e-04 - dice_coef_multiclass: 0.9997 - val_loss: 8.5674e-04 - val_dice_coef_multiclass: 0.9995 - lr: 7.0000e-05
19/100
692 [=====] - ETA: 0s - loss: 5.3981e-04 - dice_coef_multiclass: 0.9997
19: val_loss did not improve from 0.00062
692 [=====] - ETA: 0s - loss: 5.3981e-04 - dice_coef_multiclass: 0.9997 - val_loss: 6.5400e-04 - val_dice_coef_multiclass: 0.9997 - lr: 7.0000e-05
20/100
692 [=====] - ETA: 0s - loss: 5.0670e-04 - dice_coef_multiclass: 0.9997
20: val_loss improved from 0.00062 to 0.00057, saving model to ./models$best_model.h5
692 [=====] - ETA: 0s - loss: 5.0670e-04 - dice_coef_multiclass: 0.9997 - val_loss: 5.6869e-04 - val_dice_coef_multiclass: 0.9997 - lr: 7.0000e-05
21/100
692 [=====] - ETA: 0s - loss: 5.2018e-04 - dice_coef_multiclass: 0.9997
21: val_loss improved from 0.00057 to 0.00052, saving model to ./models$best_model.h5
692 [=====] - ETA: 0s - loss: 5.2018e-04 - dice_coef_multiclass: 0.9997 - val_loss: 5.1701e-04 - val_dice_coef_multiclass: 0.9997 - lr: 7.0000e-05
22/100
692 [=====] - ETA: 0s - loss: 4.8905e-04 - dice_coef_multiclass: 0.9997
22: val_loss did not improve from 0.00052
692 [=====] - ETA: 0s - loss: 4.8905e-04 - dice_coef_multiclass: 0.9997 - val_loss: 5.2636e-04 - val_dice_coef_multiclass: 0.9997 - lr: 7.0000e-05
23/100
692 [=====] - ETA: 0s - loss: 4.8972e-04 - dice_coef_multiclass: 0.9997
23: val_loss did not improve from 0.00052
692 [=====] - ETA: 0s - loss: 4.8972e-04 - dice_coef_multiclass: 0.9997 - val_loss: 6.1951e-04 - val_dice_coef_multiclass: 0.9997 - lr: 7.0000e-05
24/100
692 [=====] - ETA: 0s - loss: 4.8809e-04 - dice_coef_multiclass: 0.9997
24: val_loss improved from 0.00052 to 0.00051, saving model to ./models$best_model.h5
692 [=====] - ETA: 0s - loss: 4.8809e-04 - dice_coef_multiclass: 0.9997 - val_loss: 5.1079e-04 - val_dice_coef_multiclass: 0.9997 - lr: 7.0000e-05
25/100
692 [=====] - ETA: 0s - loss: 7.2864e-04 - dice_coef_multiclass: 0.9996
25: val_loss did not improve from 0.00051
692 [=====] - ETA: 0s - loss: 7.2864e-04 - dice_coef_multiclass: 0.9996 - val_loss: 5.5468e-04 - val_dice_coef_multiclass: 0.9997 - lr: 7.0000e-05
26/100

```

[train_metrics.csv](#)



[train_losses.csv](#)



4 Evaluation

Please move to `./projects/TensorFlowFlexUNet/BreastDM` folder,
and run the following bat file to evaluate TensorFlowFlexUNet model for BreastDM.

`./2.evaluate.bat`

This bat file simply runs the following command.

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

```
-- dilation (1, 1)
-- kernel_size (6, 5)
-- dilation (1, 1)
-- kernel_size (3, 3)
-- dilation (1, 1)
-- kernel_size (3, 3)
-- dilation (1, 1)
-- kernel_size (3, 3)
-- dilation (1, 1)
-- kernel_size (3, 3)
-- dilation (1, 1)
-- kernel_size (3, 3)
-- dilation (1, 1)
-- kernel_size (3, 3)
-- dilation (1, 1)
-- kernel_size (3, 3)
+++ kernel_size (3, 3)
+++ dilation (1, 1)
+++ kernel_size (3, 3)
+++ dilation (1, 1)
+++ kernel_size (3, 3)
+++ dilation (1, 1)
+++ kernel_size (3, 3)
+++ dilation (1, 1)
+++ kernel_size (3, 3)
+++ dilation (1, 1)
+++ kernel_size (5, 5)
+++ dilation (1, 1)
+++ kernel_size (7, 7)
+++ dilation (1, 1)
=== WARNING: Not found [model] final_activation, return default value sigmoid
=== final_activation softmax
clipvalue 0.5
-- Optimizer Adam learning_rate 7e-05 clipvalue 0.5
== Loaded a weight file ./models/best_model.h5
E:\tensorflow\tensorflow\dataset\BreastDM/test/images/ masks_dir ../../dataset/BreastDM/test/masks/
-- num_image_files: 482 num_mask_files:482
-- num_classes 3 image_data_type <class 'numpy.uint8'>
-- num_images 482 512 512
-- num_classes 3 mask_data_type <class 'numpy.int8'>
=== WARNING: Not found [eval] batch_size, return default value 4
E:\tensorflow\tensorflow\dataset\BreastDM/train/images/ masks_dir ../../dataset/BreastDM/train/masks/
rFlow 2.0, as updates are applied automatically.
updates = self.state_updates
Test loss :0.0004
Test accuracy:0.9997
=== Saved ./evaluation.csv
```

The loss (categorical_crossentropy) to this BreastDM/test was very low and dice_coef_multiclass very high as shown below.

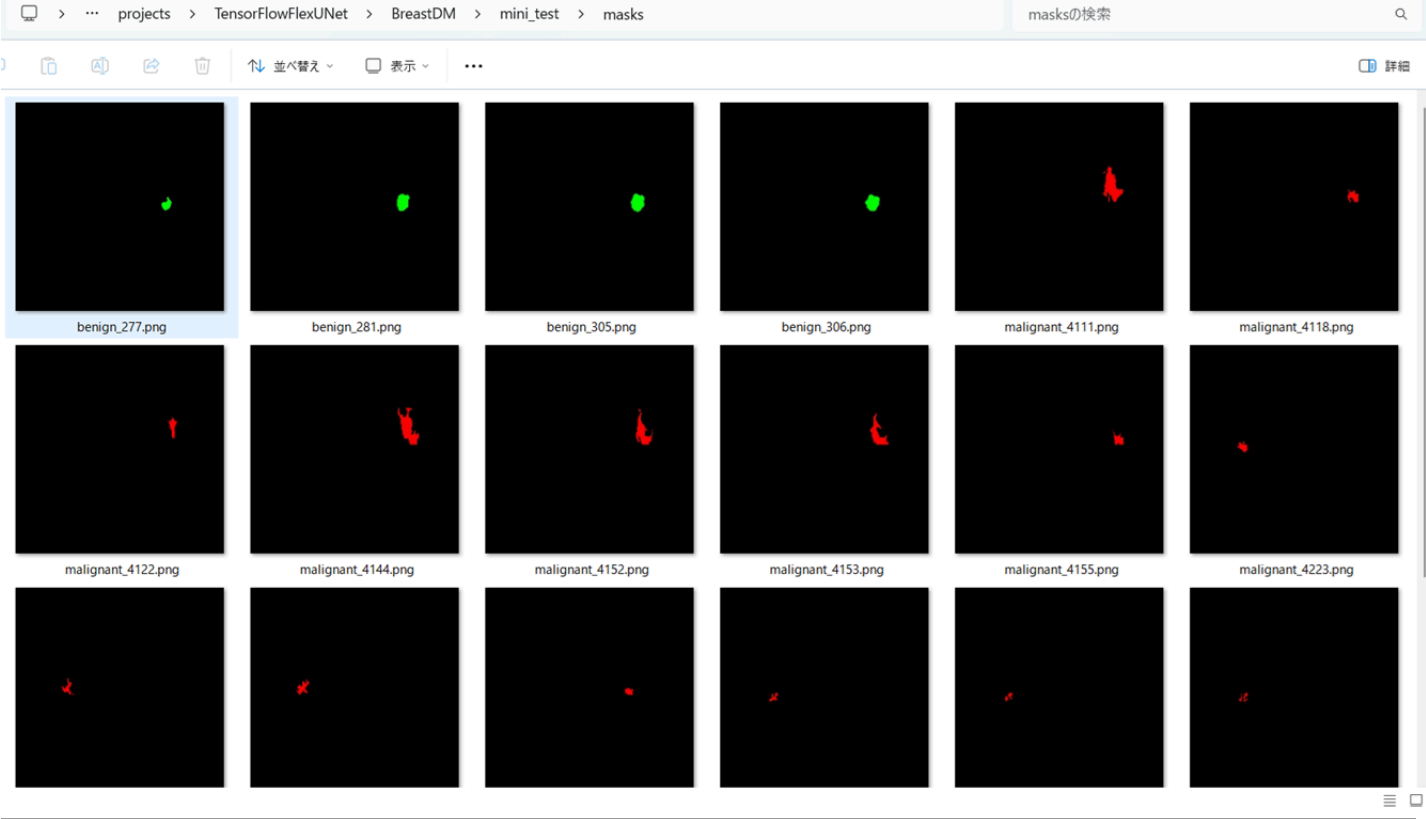
5 Inference

,and run the following bat file to infer segmentation regions for images by the Trained-TensorFlowFlexUNet model for BreastDM.

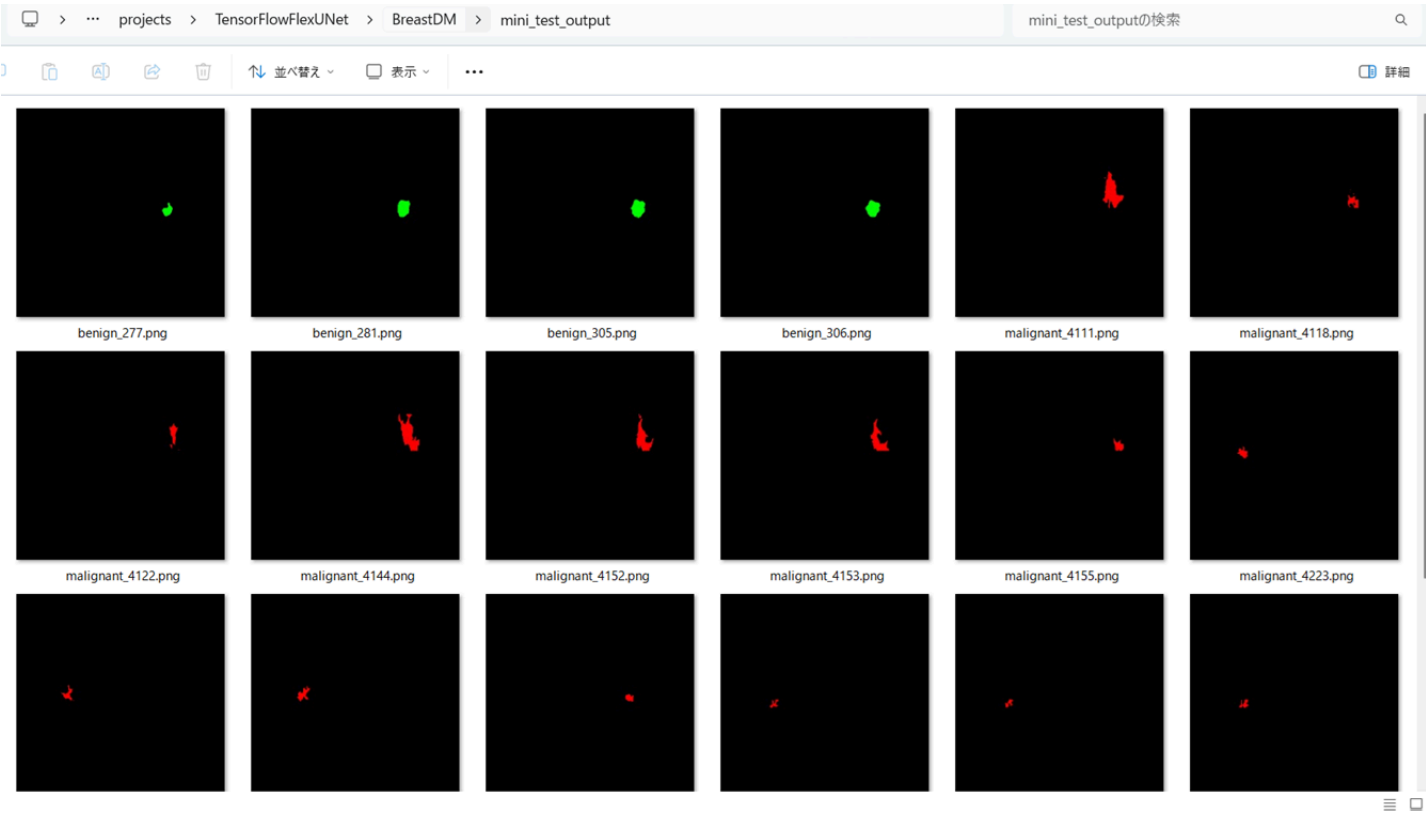
This simply runs the following command.

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


mini_test_mask(ground_truth)



Inferred test masks



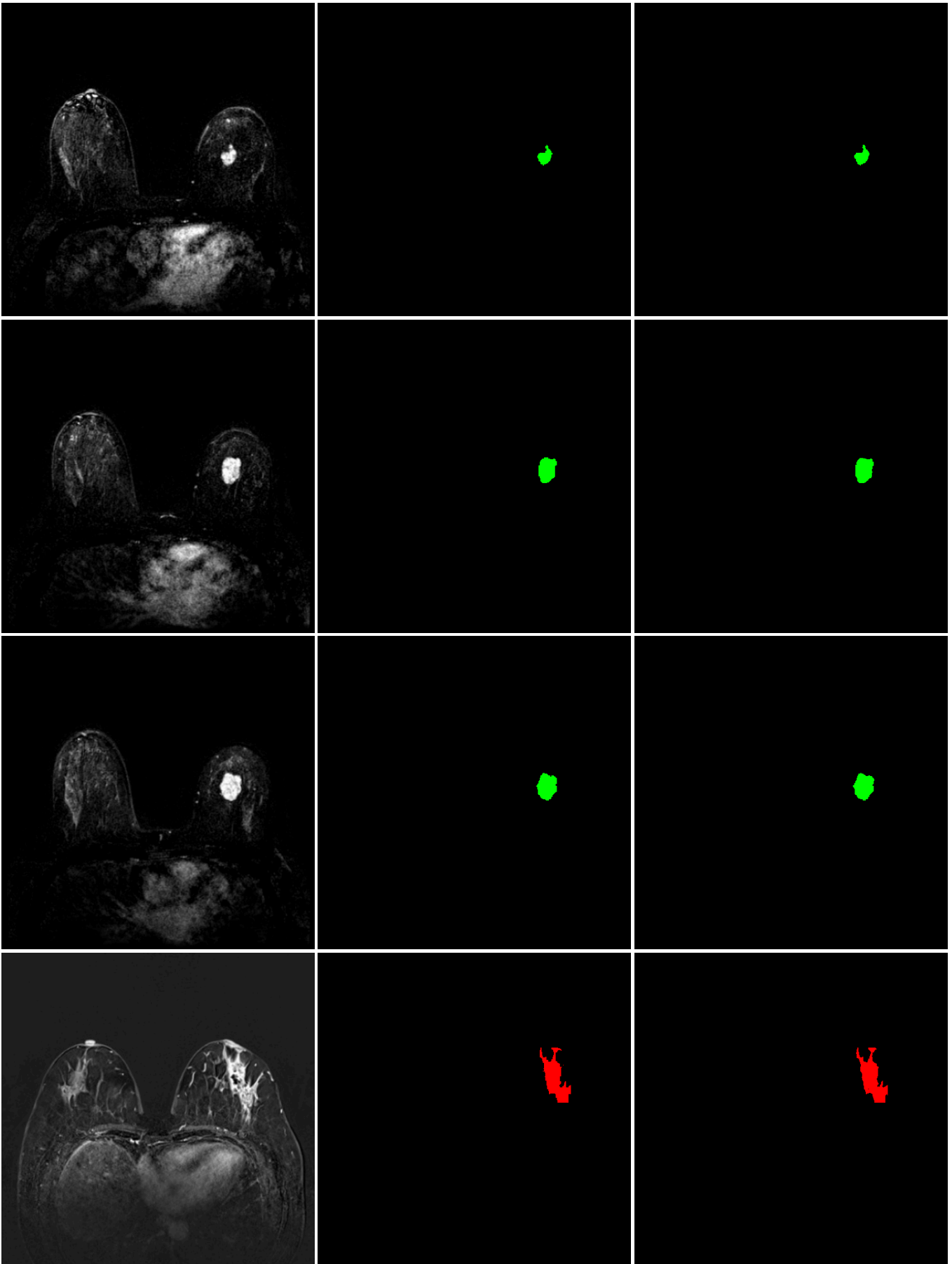
Enlarged images and masks of 512x512 pixels

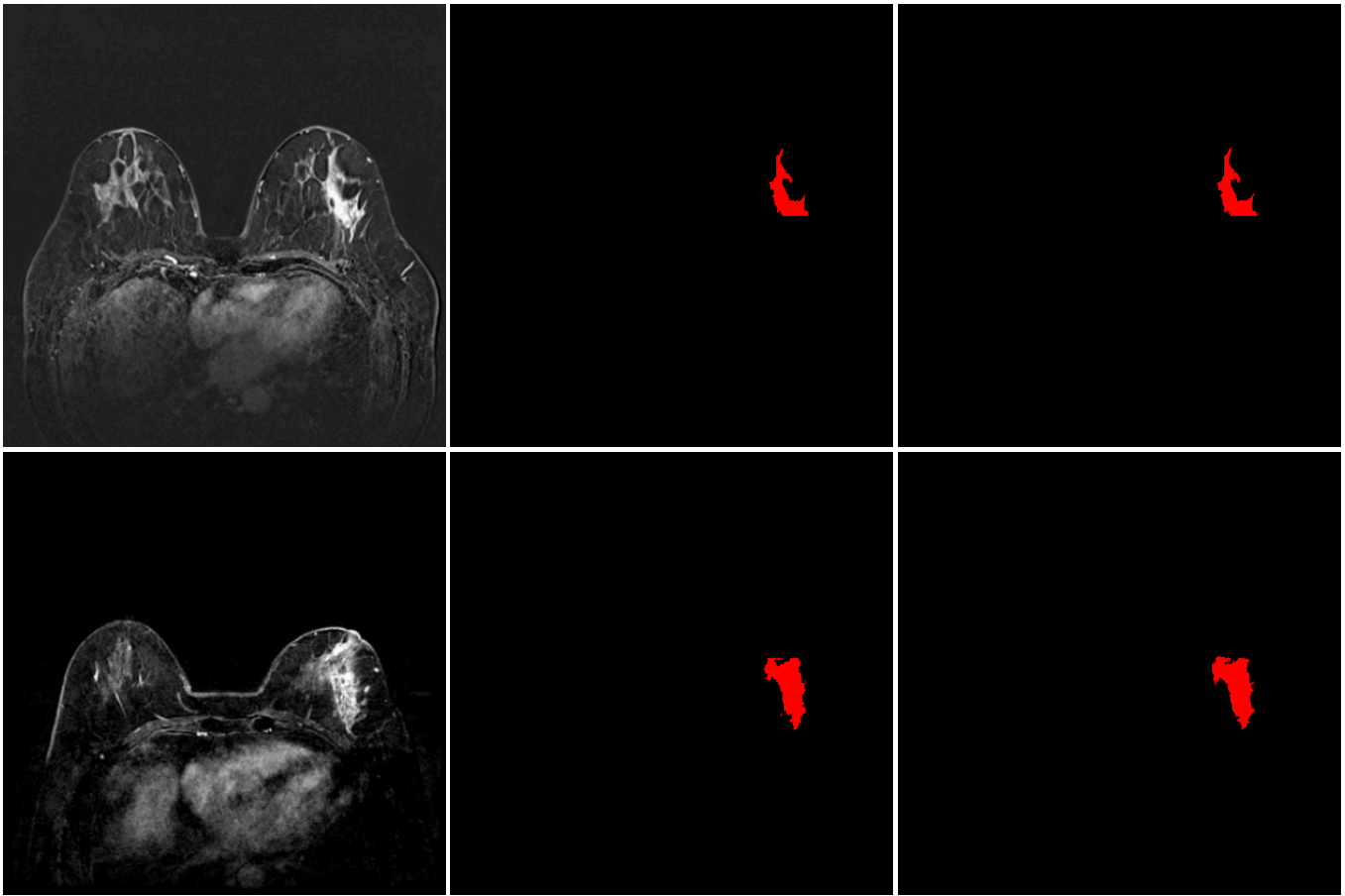
rgb_map = (benign:green, malignant:red)

Image

Mask (ground_truth)

Inferred-mask





References

1. BreastDM: A DCE-MRI dataset for breast tumor image segmentation and classification

Xiaoming Zhao, Yuehui Liao, Jiahao Xie, Xiaxia He, Shiqing Zhang, Guoyu Wang
, Jiangxiong Fang , Hongsheng Lu, Jun Yu

<https://doi.org/10.1016/j.compbiomed.2023.107255>

<https://www.sciencedirect.com/science/article/abs/pii/S0010482523007205>

2. Tensorflow-Image-Segmentation-Malignant-BreastDM

Toshiyuki Arai @antillia.com

<https://github.com/sarah-antillia/Tensorflow-Image-Segmentation-Malignant-BreastDM>