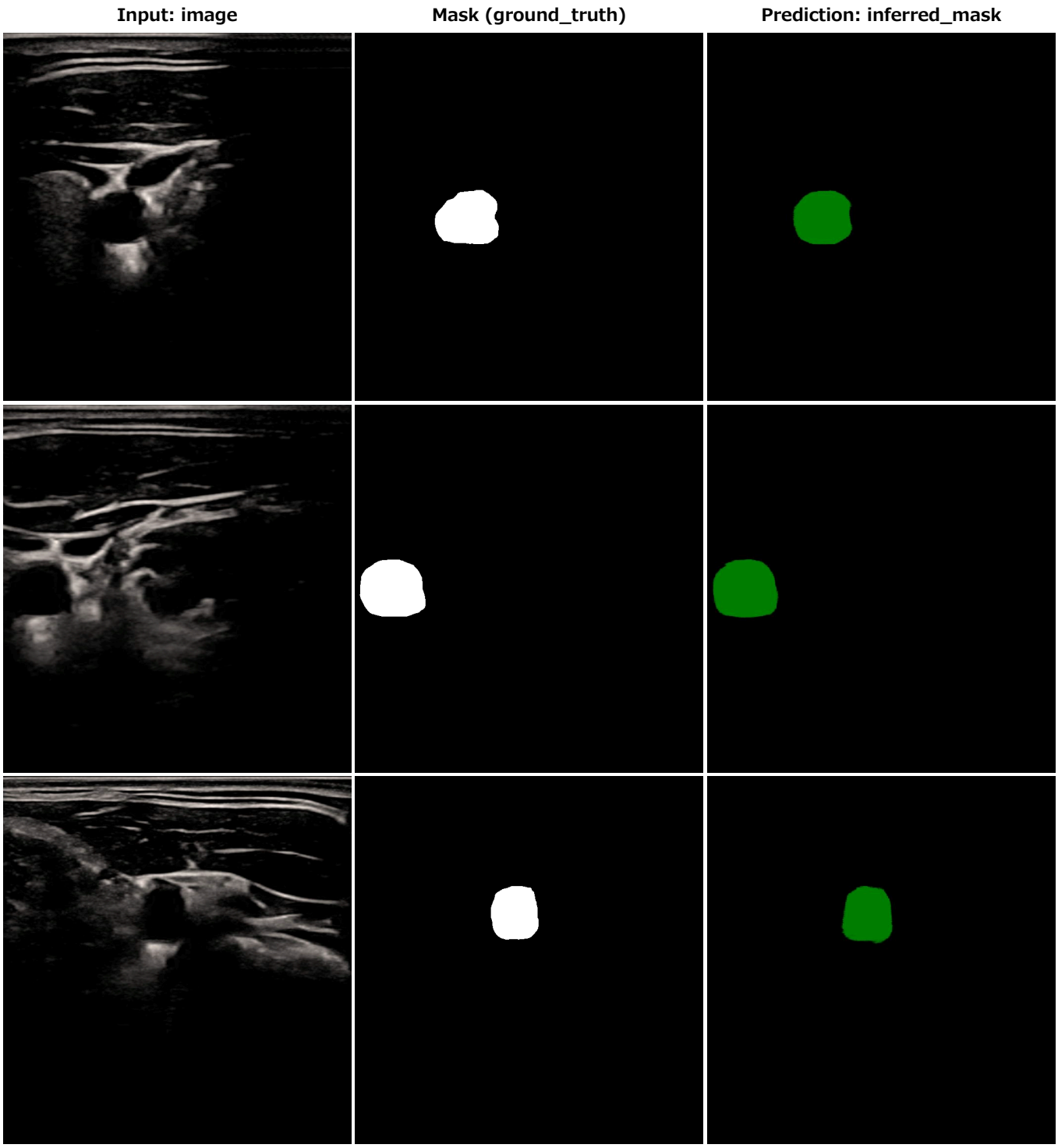
Tensorflow-Image-Segmentation-Common-Carotid-Artery-UltraSound-Images (2025/02/17)

Sarah T. Arai
Software Laboratory antillia.com

This is the first experiment of Image Segmentation for **CCAUS (Common Carotid Artery UltraSound) Images** based on the latest Tensorflow-Image-Segmentation-API, and Mendeley Common Carotid Artery Ultrasound Images

---

Actual Tiled Image Segmentation for Images of 709x749 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 CCAUSSegmentation 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

The dataset used here has been take from the following Mendeley website:
Common Carotid Artery Ultrasound Images

Momot, Agata (2022), “Common Carotid Artery Ultrasound Images”,
Mendeley Data, V1, doi: 10.17632/d4xt63mgjm.1

Description
The dataset consists of aquistified ultrasound images of the common carotid artery. The images were taken from a Mindary UMT-500Plus ultrasound machine with an L13-3s linear probe. The study group consisted of 11 subjects, with each person examined at least once on the left and right sides. 2 subjects were examined using the vascular modality and 8 using the carotid modality. Time series (DICOM) were converted to png files and cropped appropriately. Each person had 100 images, making a total of 1100 images. The dataset also includes corresponding expert masks (corresponding file name) made by a technician and verified by an expert. The collection can be used for carotid artery segmentation and geometry measurement and evaluation.
Image resolution: 709 x 749 x 3
Number of images: 2200
File format: PNG

License:
Creative Commons Attribution 4.0 International License.

2 CCAUS ImageMask Dataset

If you would like to train this CCAUS Segmentation model by yourself, please download the dataset **Common Carotid Artery Ultrasound Images**

The folder structure of the dataset is the following.

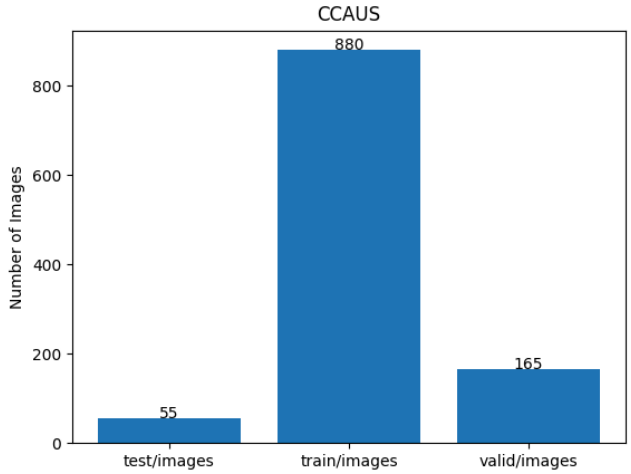
```
./Common Carotid Artery Ultrasound Images
    ├──Expert mask images
    │    ├──20220111217481000022VAS_slice_1069.png
    │    ├──20220111217481000022VAS_slice_1080.png
    │     ...
    │    └──20220207135920005 6VAS_slice_258.png
    └──US images
         ├──20220111217481000022VAS_slice_1069.png
         ├──20220111217481000022VAS_slice_1080.png
          ...
         └──20220207135920005 6VAS_slice_258.png
```

Please run the following Python script to split the original 709x749 pixels dataset into **test**, **train** and **valid** subsets.

- split_master.py

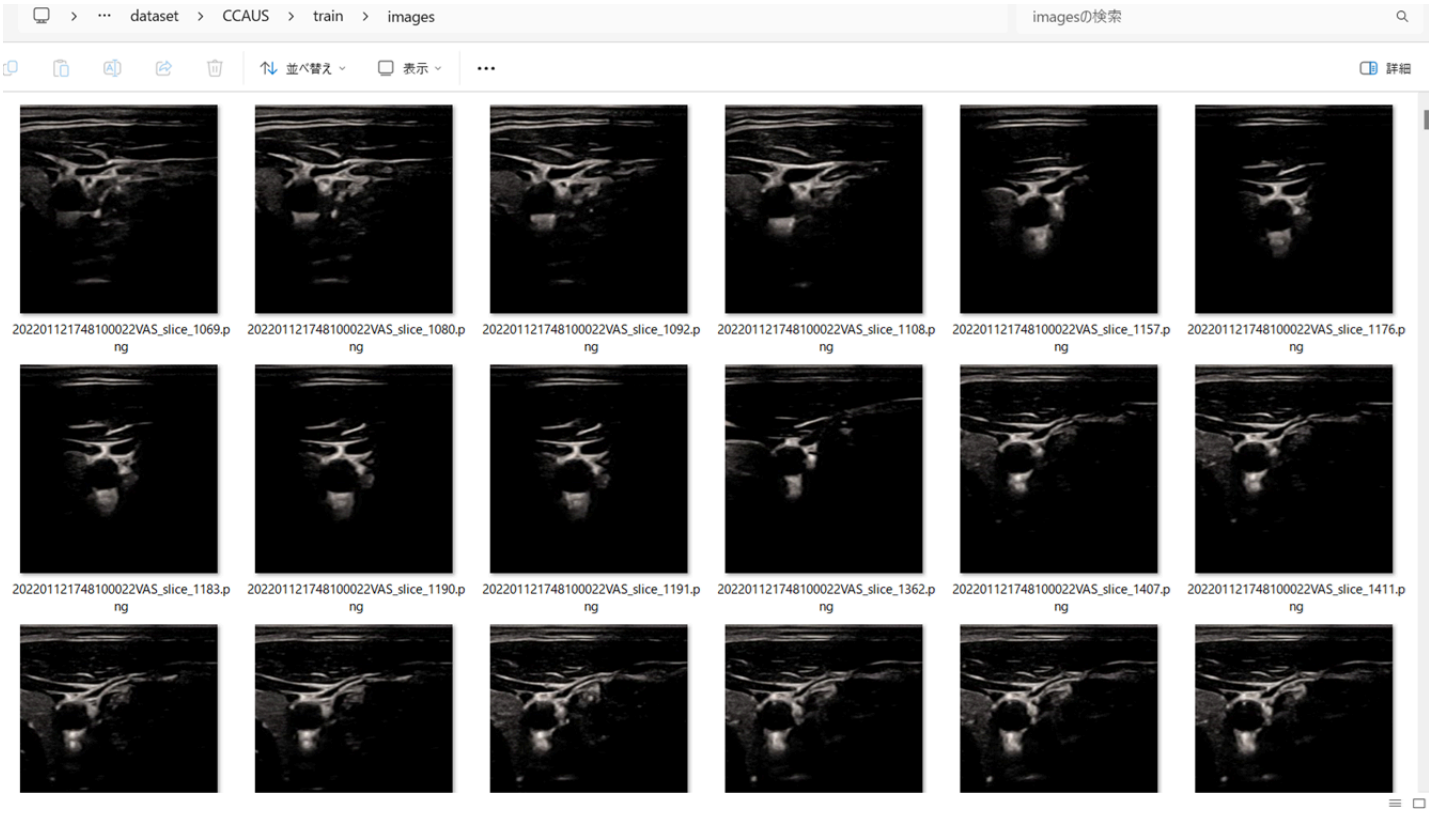
```
./dataset
└──CCAUS
    ├──test
    │    ├──images
    │    └──masks
    ├──train
    │    ├──images
    │    └──masks
    └──valid
         ├──images
         └──masks
```

CCAUS Statistics

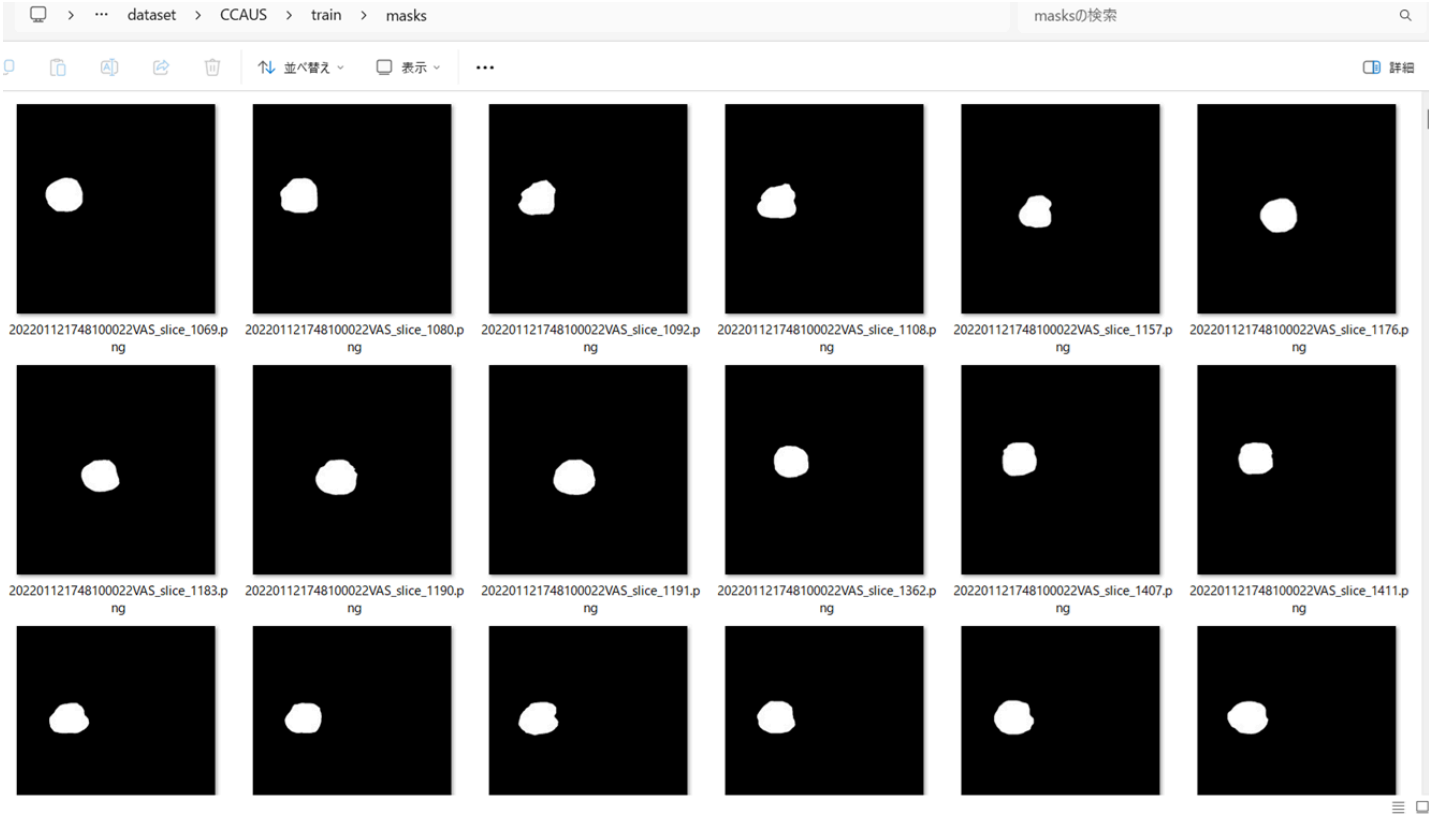


As shown above, the number of images of train and valid datasets is not so large to use for a training set of our segmentation model.

Train_images_sample



Train_masks_sample



3 Train TensorflowUNet Model

We have trained CCAUS TensorflowUNet Model by using the following **train_eval_infer.config** file.
Please move to ./projects/TensorflowSlightlyFlexibleUNet/CCAUSand run the following bat file.

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

, which simply runs the following command.

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

Model parameters
Enabled Batch Normalization.
Defined a small **base_filters=16** and large **base_kernels=(9,9)** 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 tool.

```
[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
```

Dataset class

Specified ImageMaskDataset class.

```
[dataset]
datasetclass  = "ImageMaskDataset"
resize_interpolation = "cv2.INTER_CUBIC"
```

Early stopping callback

Enabled early stopping callback with patience parameter.

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

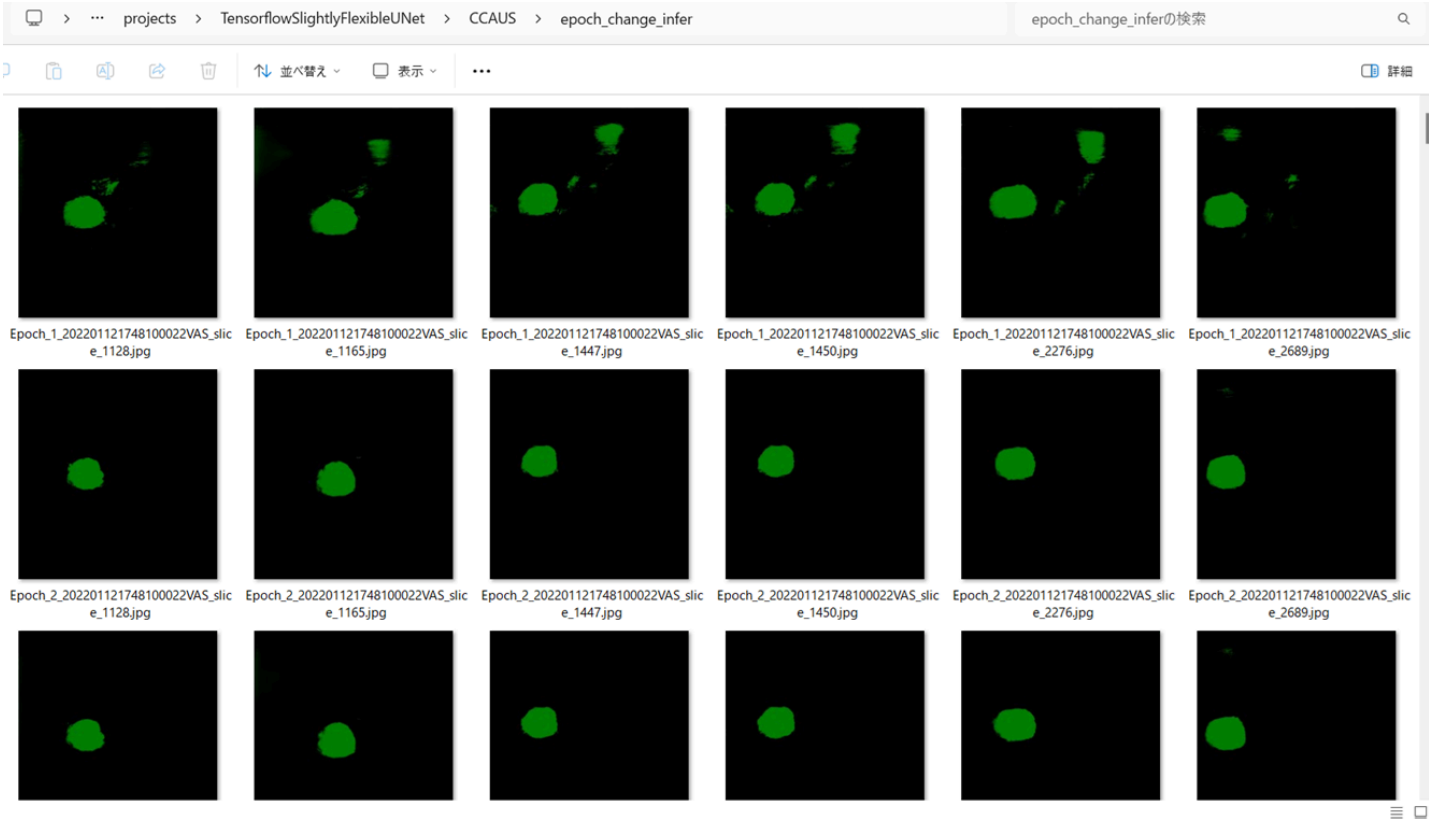
Epoch change inference callbacks

Enabled epoch_change_infer callback.

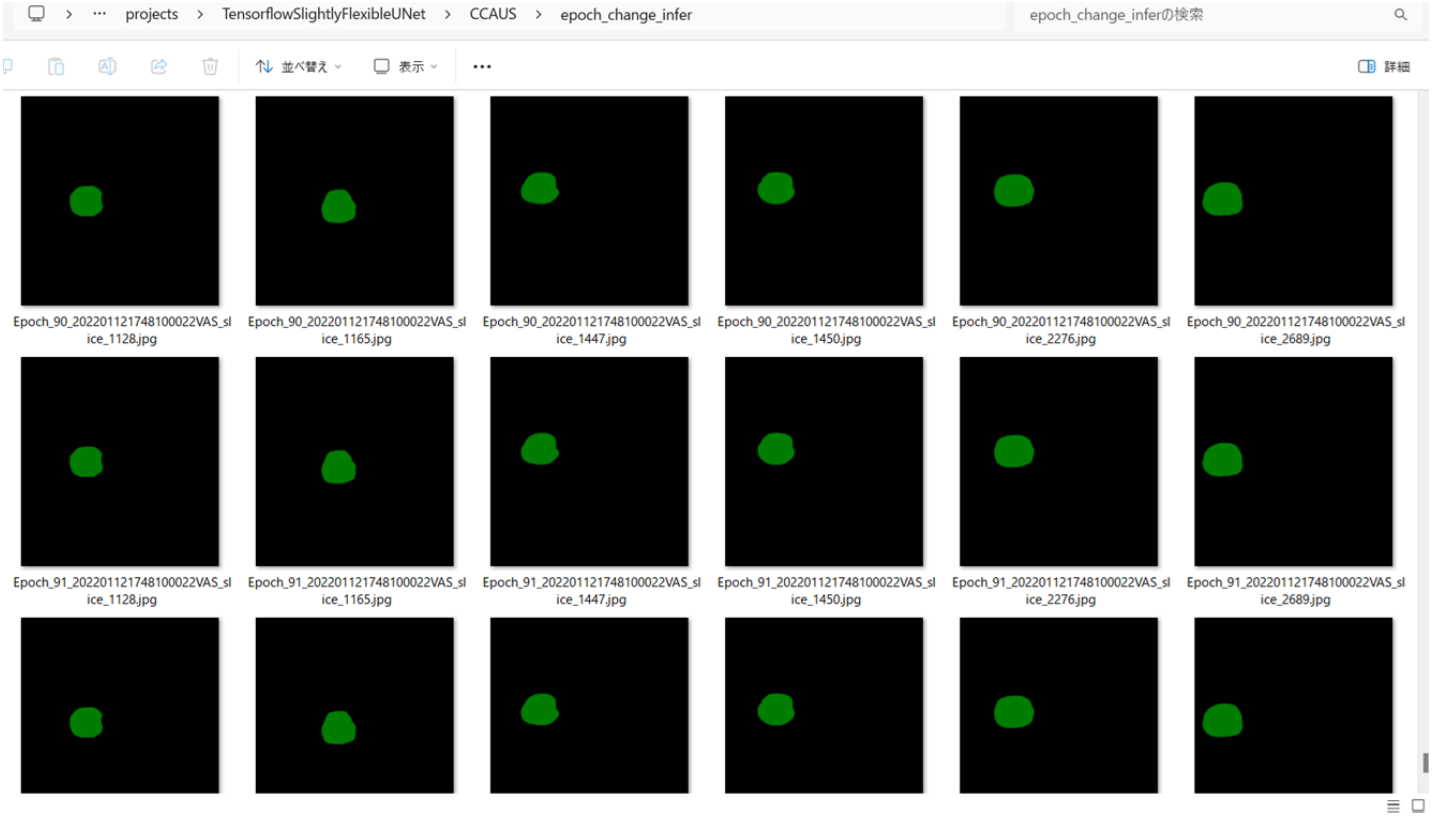
```
[train]

epoch_change_infer      = True
epoch_change_infer_dir  = "./epoch_change_infer"
epoch_change_tiledinfer = False
epoch_change_tiledinfer_dir = "./epoch_change_tiledinfer"
num_infer_images        = 6
```

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

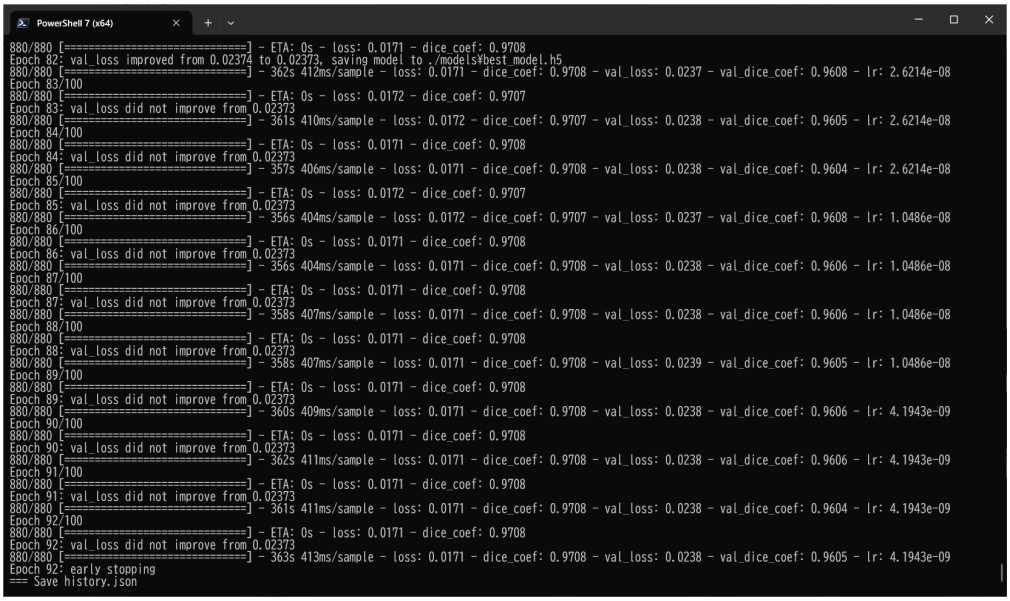
Epoch_change_inference output at starting (1,2,3)



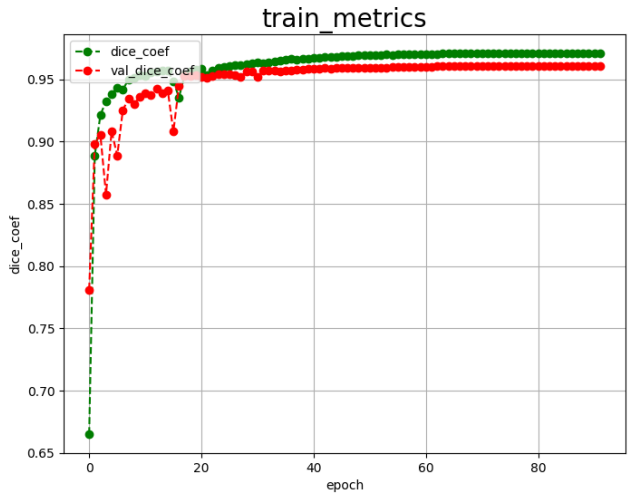
Epoch_change_inference output at ending (90,91,92)



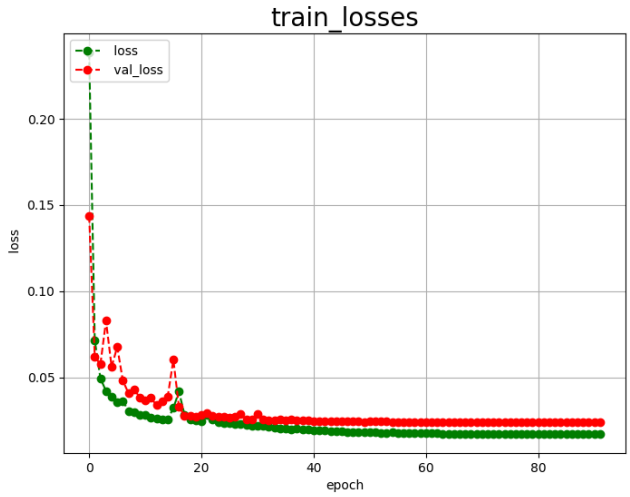
In this experiment, the training process was stopped at epoch 92 by EarlyStopping Callback.



train_metrics.csv



train_losses.csv



4 Evaluation

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

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

This bat file simply runs the following command.

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

Evaluation console output:

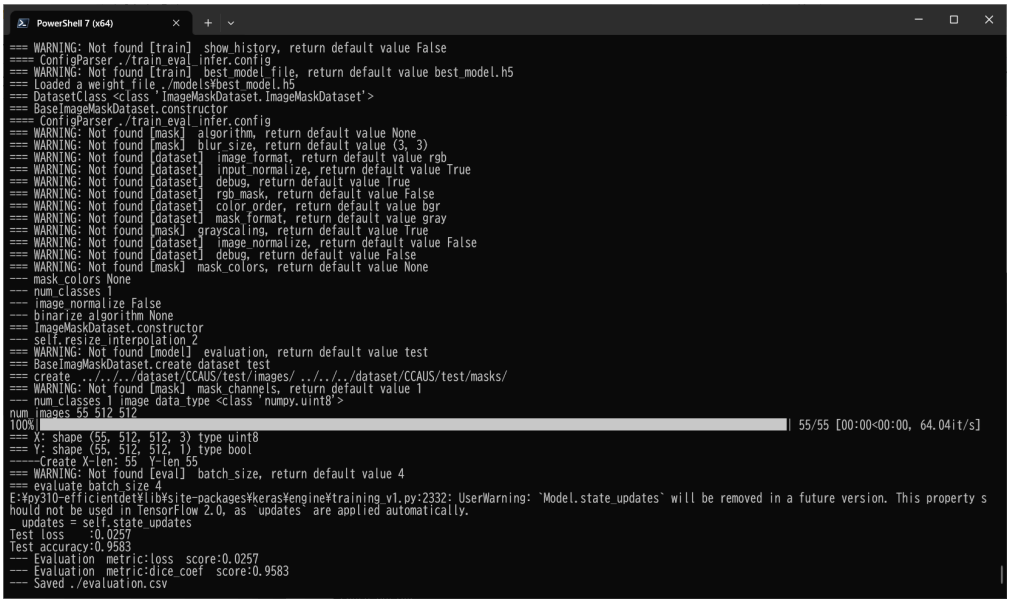


Image-Segmentation-CCAUS [evaluation.csv](./projects/TensorflowSlightlyFlexibleUNet/CCAUS/evaluation.csv)
The loss (bce_dice_loss) to this CCAUS/test was low, and dice_coef high as shown below.

```
loss,0.0257
dice_coef,0.9583
```

5 Inference

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

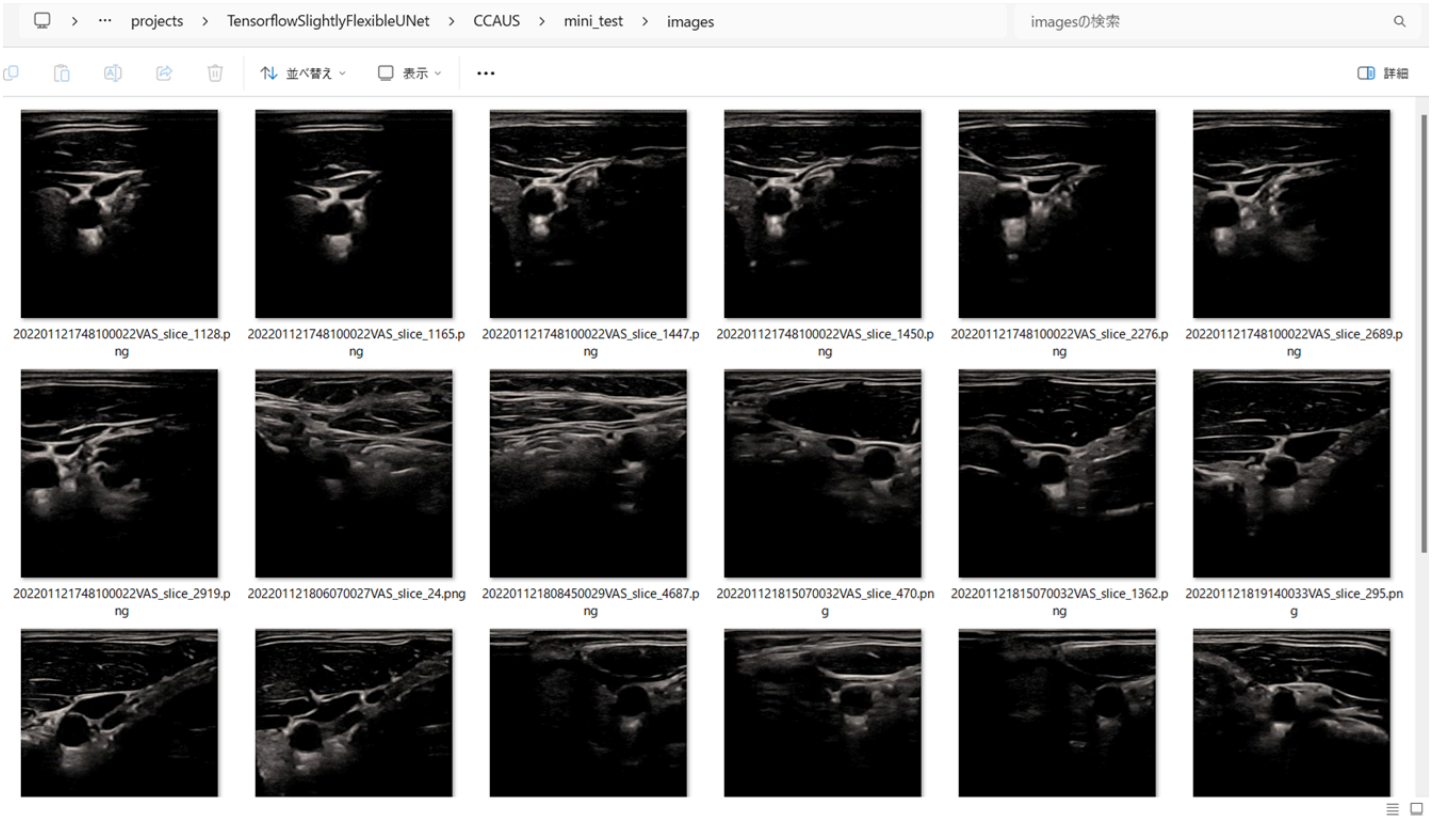
```
./3.infer.bat
```

This simply runs the following command.

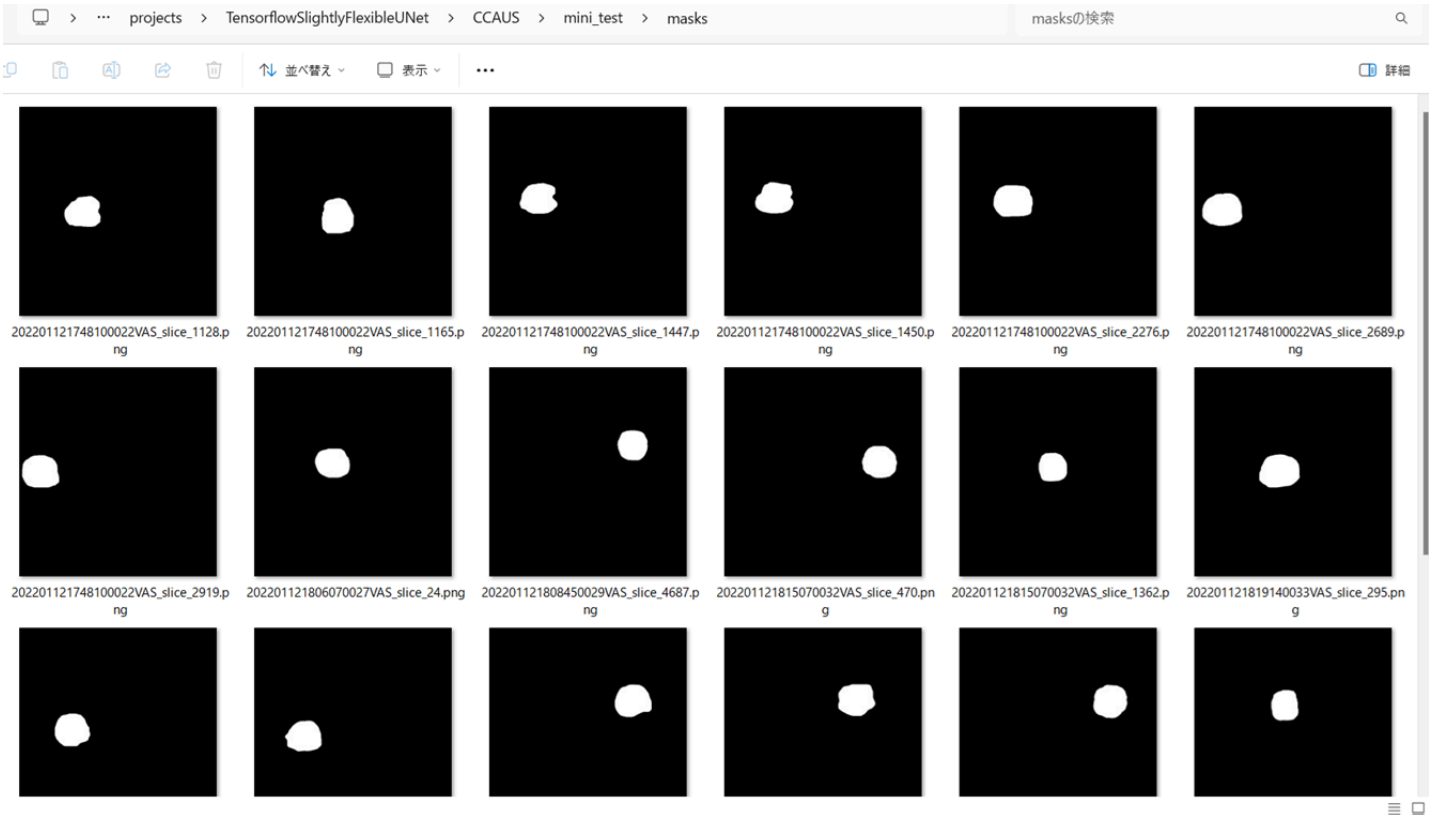
```
python ../../../src/TensorflowUNetInferencer.py ./train_eval_infer.config
```

---

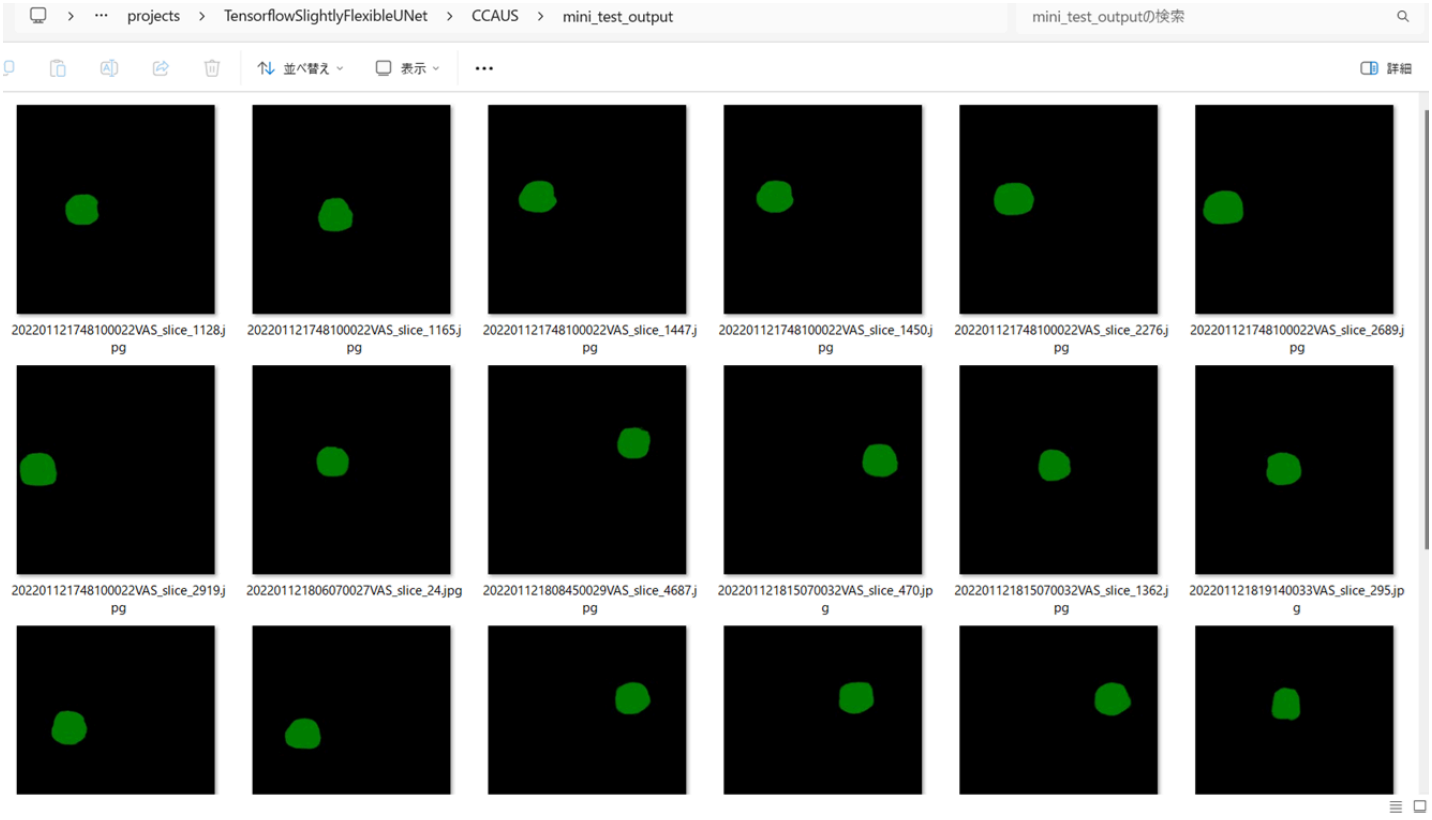
mini_test_images (709x749 pixels)



mini_test_mask(ground_truth)

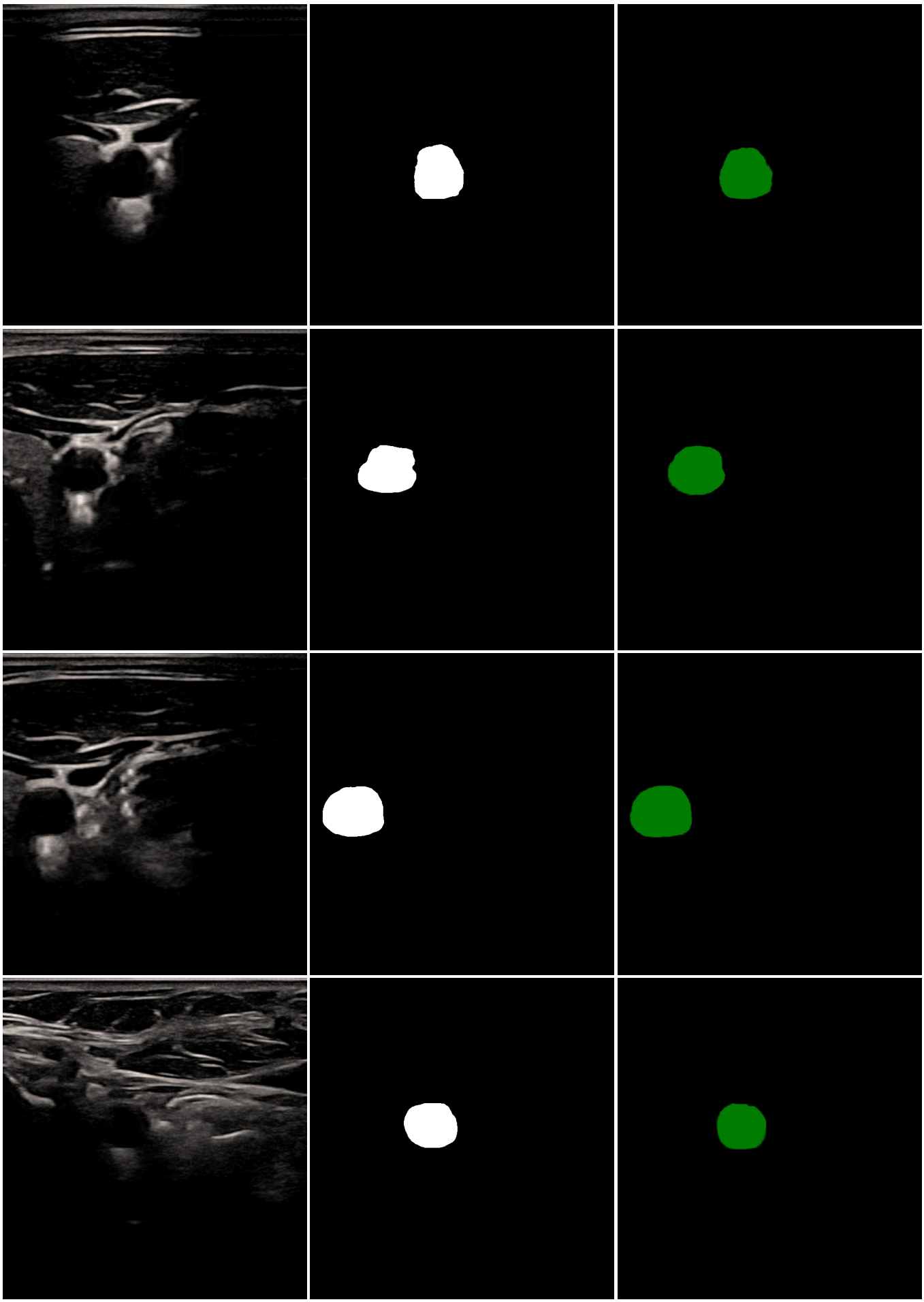


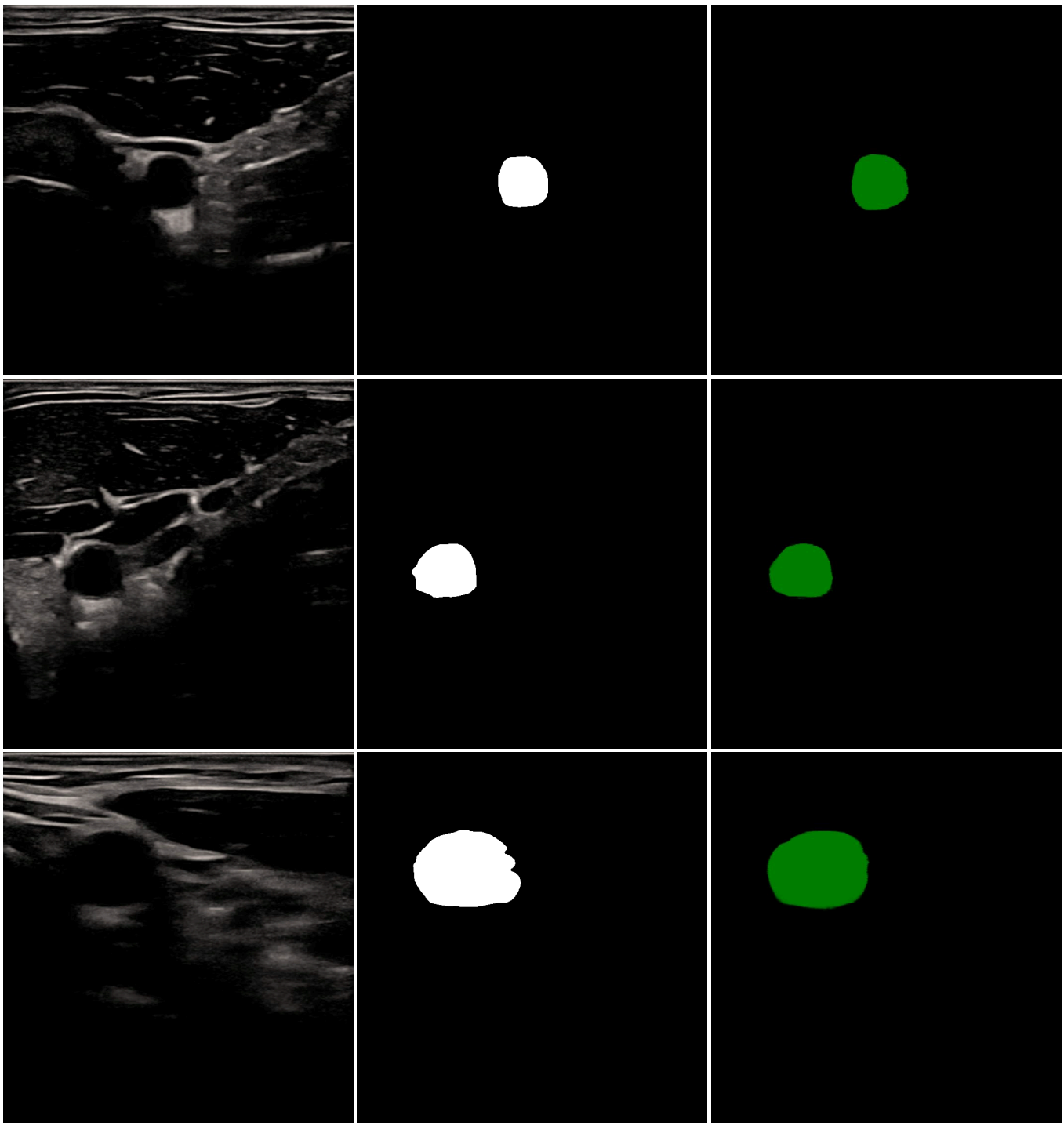
Inferred test masks (709x749 pixels)



Enlarged images and masks of 709x749 pixels

Image	Mask (ground_truth)	Inferred-mask





References

1. Ultrasound Common Carotid Artery Segmentation Based on Active Shape Model
Xin Yang, Jiaoying Jin, Mengling Xu, Huihui Wu, Wanji He, Ming Yuchi, Mingyue Ding
https://pmc.ncbi.nlm.nih.gov/articles/PMC3606761/

2. Automated Segmentation of Common Carotid Artery in Ultrasound Images
J. H. Gagan; Harshit S. Shirsat; Grissel P. Mathias; B. Vaibhav Mallya; Jasbon Andrade; K. V. Rajagopal
Published in: IEEE Access (Volume: 10)
https://ieeexplore.ieee.org/document/9785785

3. Method for Carotid Artery 3-D Ultrasound Image Segmentation Based on CSWin Transformer
Yanping Lin, Jianhua Huang, Wangjie Xu, Cancan Cui, Wenzhe Xu, Zhaojun Li
https://www.sciencedirect.com/science/article/abs/pii/S0301562922006342

4. Common Carotid Artery Ultrasound

https://www.cuh.nhs.uk/patient-information/ultrasound-scan-of-your-carotid-arteries/