Tensorflow-Image-Segmentation-Augmented-ALCAPA (2025/02/15)

This is the first experiment of Image Segmentation for **ALCAPA (Anomalous Left Coronary Artery from Pulmonary Artery)** based on the latest Tensorflow-Image-Segmentation-API, and ALCAPA-ImageMask-Dataset.zip, which was derived by us from the dataset ImageALCAPA_BIBM_Publish.change2zip in **ImageALCAPA**

Data Augmentation Strategy:
To address the limited size of ImageALCAPA dataset, we employed an online augmentation tool to augment the dataset, which supports the following augmentation methods.

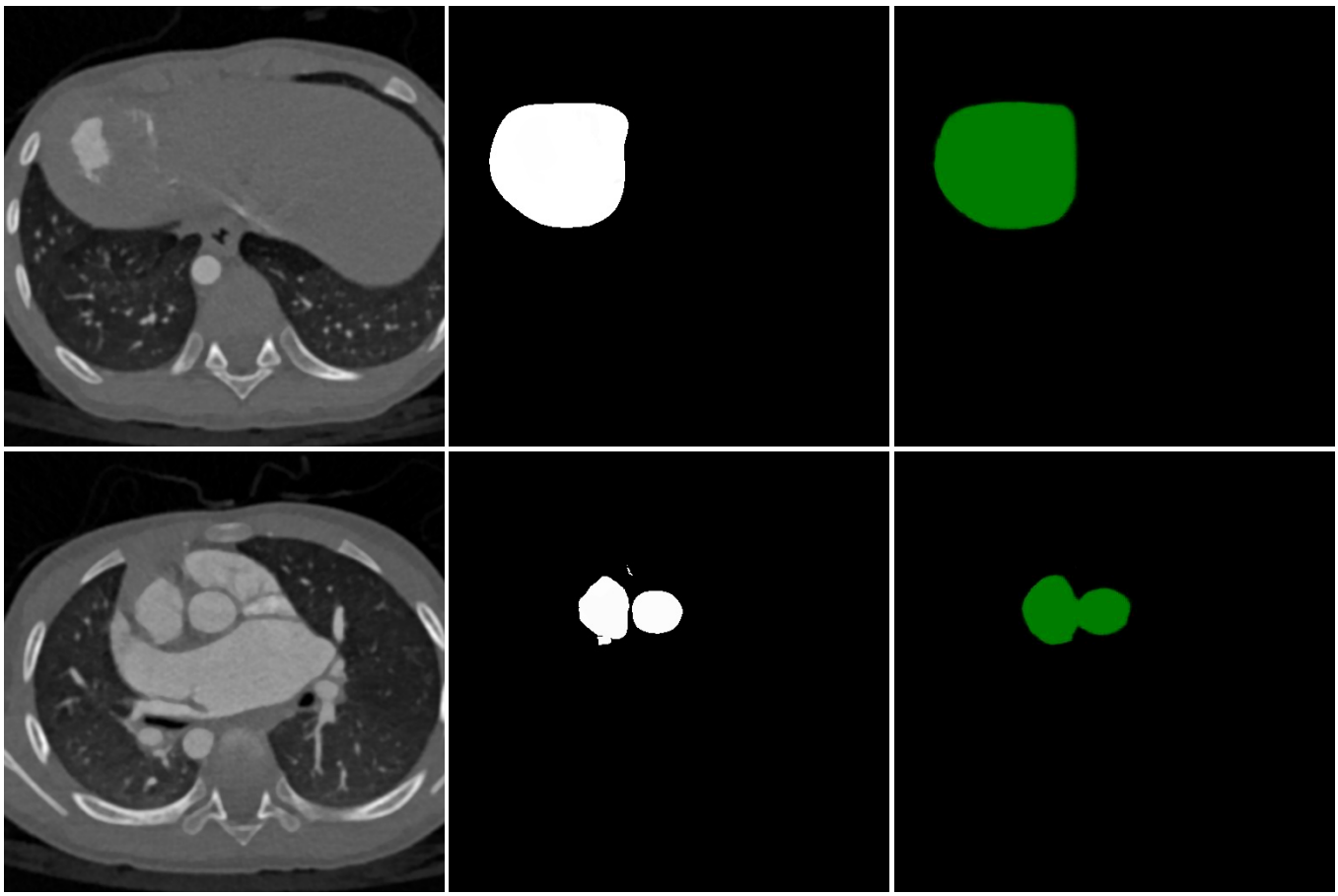
- Vertical flip
- Horizontal flip
- Rotation
- Shrinks
- Shears
- Deformation
- Distortion
- Barrel distortion
- Pincushion distortion

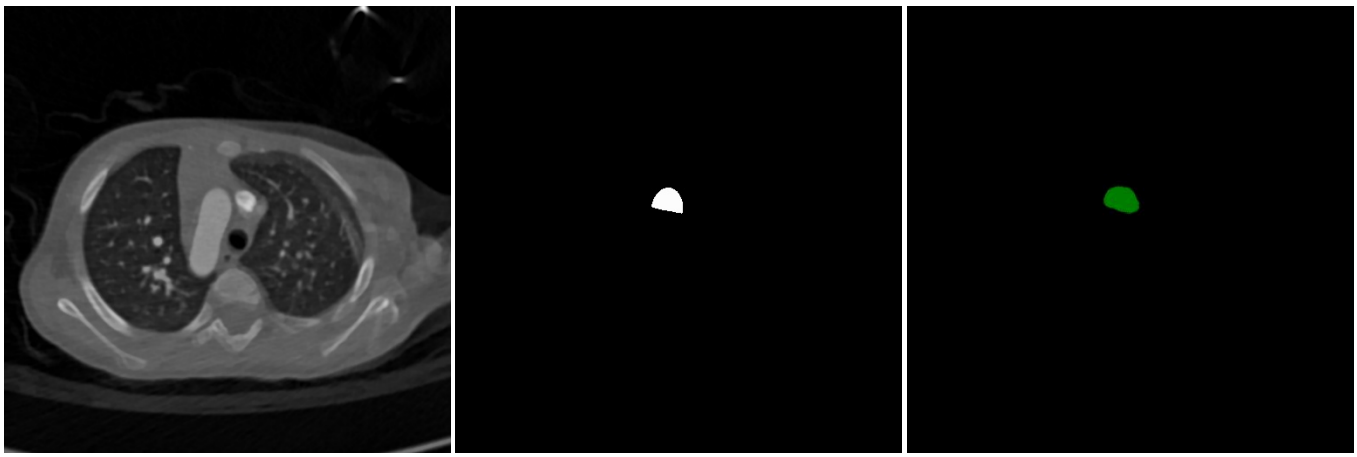
Please see also the following tools

- 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.

Input: image	Mask (ground_truth)	Prediction: inferred_mask





In this experiment, we used the simple UNet Model TensorflowSlightlyFlexibleUNet for this ALCAPA 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

The dataset used here has been taken from **ImageALCAPA_BIBM_Publish.change2zip** in the kaggle website **ImageALCAPA**

About Dataset
The ImageALCAPA dataset totally consists of 30 3D CTA images gathered from Guangdong Provincial Peoples' Hospital from June 17, 2016, to August 8, 2021. These images are acquired by a SOMATOM Definition Flash CT machine. All the images are pre-operative ALCAPA CTA images whose top and bottom are around the neck and the brachiocephalic vessels, respectively, in the axial view. The segmentation labeling is performed by a team of two cardiovascular radiologists who have extensive experience in ALCAPA. For each image, one radiologist fulfills the labelling while the other verifies it afterwards. The segmentation includes seven substructures: Myo, LV, RV, PA, Ao, LCA, and RCA, and the labelling of each image goes by 1-1.5 hours.

Please see also:
ImageALCAPA: A 3D Computed Tomography Image Dataset for Automatic Segmentation of Anomalous Left Coronary Artery from Pulmonary Artery

License
Apache 2.0

2 ALCAPA ImageMask Dataset

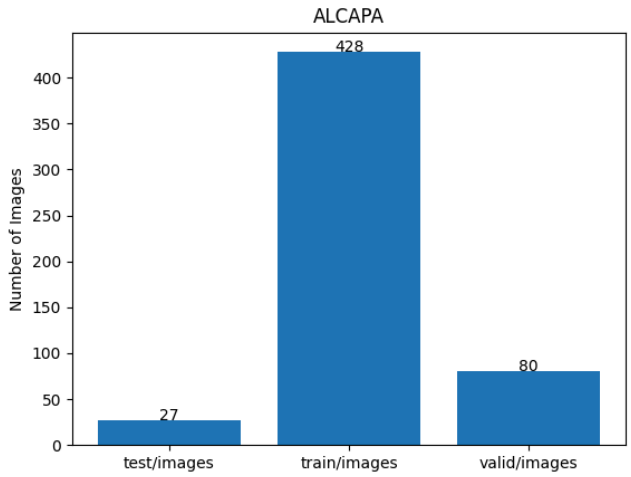
If you would like to train this ALCAPA Segmentation model by yourself, please download our 512x512 pixels JPG dataset from the google drive ALCAPA-ImageMask-Dataset.zip , expand the downloaded ImageMaskDataset and put it under **./dataset** folder to be

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

On the derivation of this ImageMask Dataset, please refer to the following Python scripts.

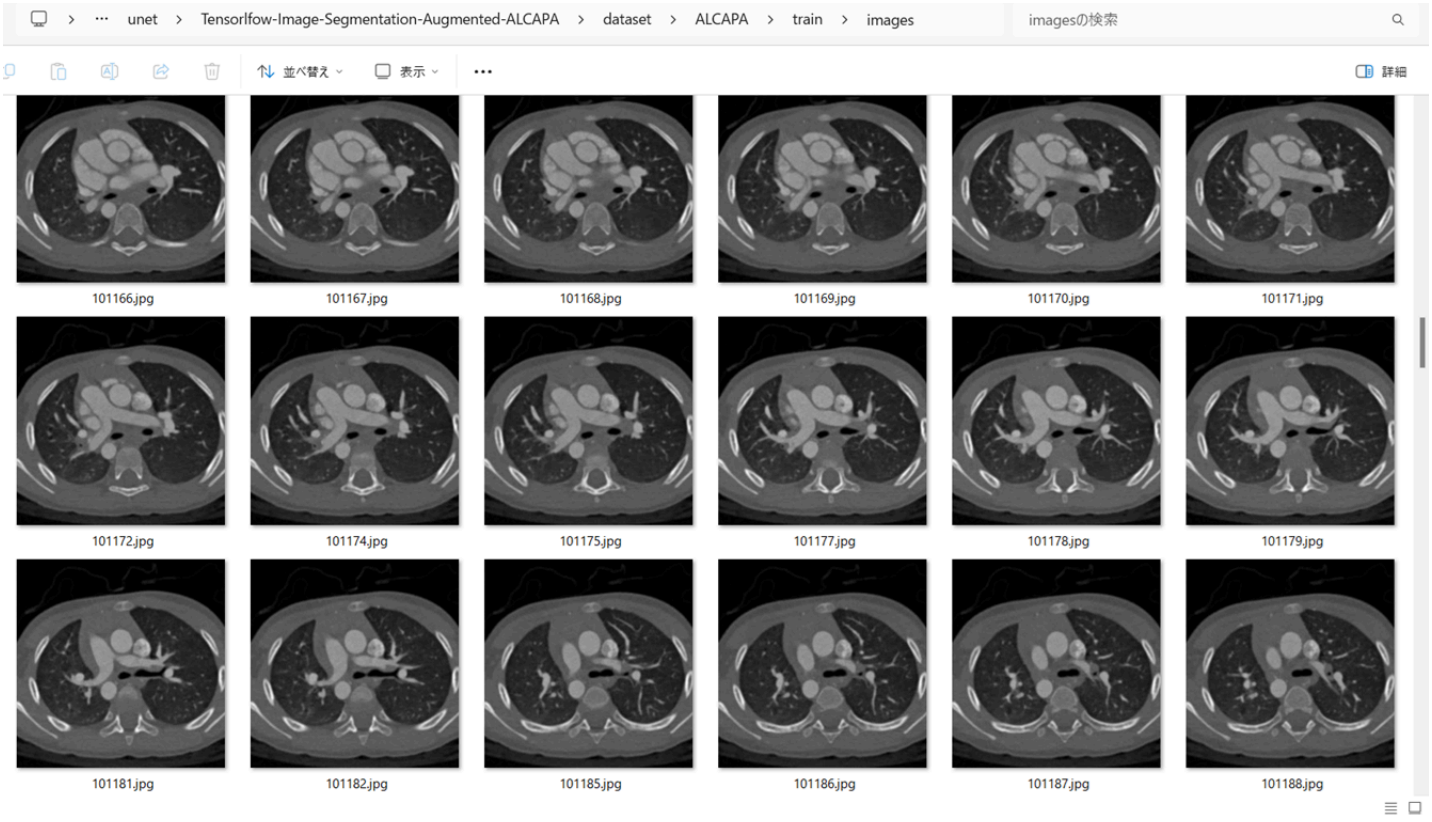
- ImageMaskDatasetGenerator.py
- split_master.py

ALCAPA Dataset Statistics

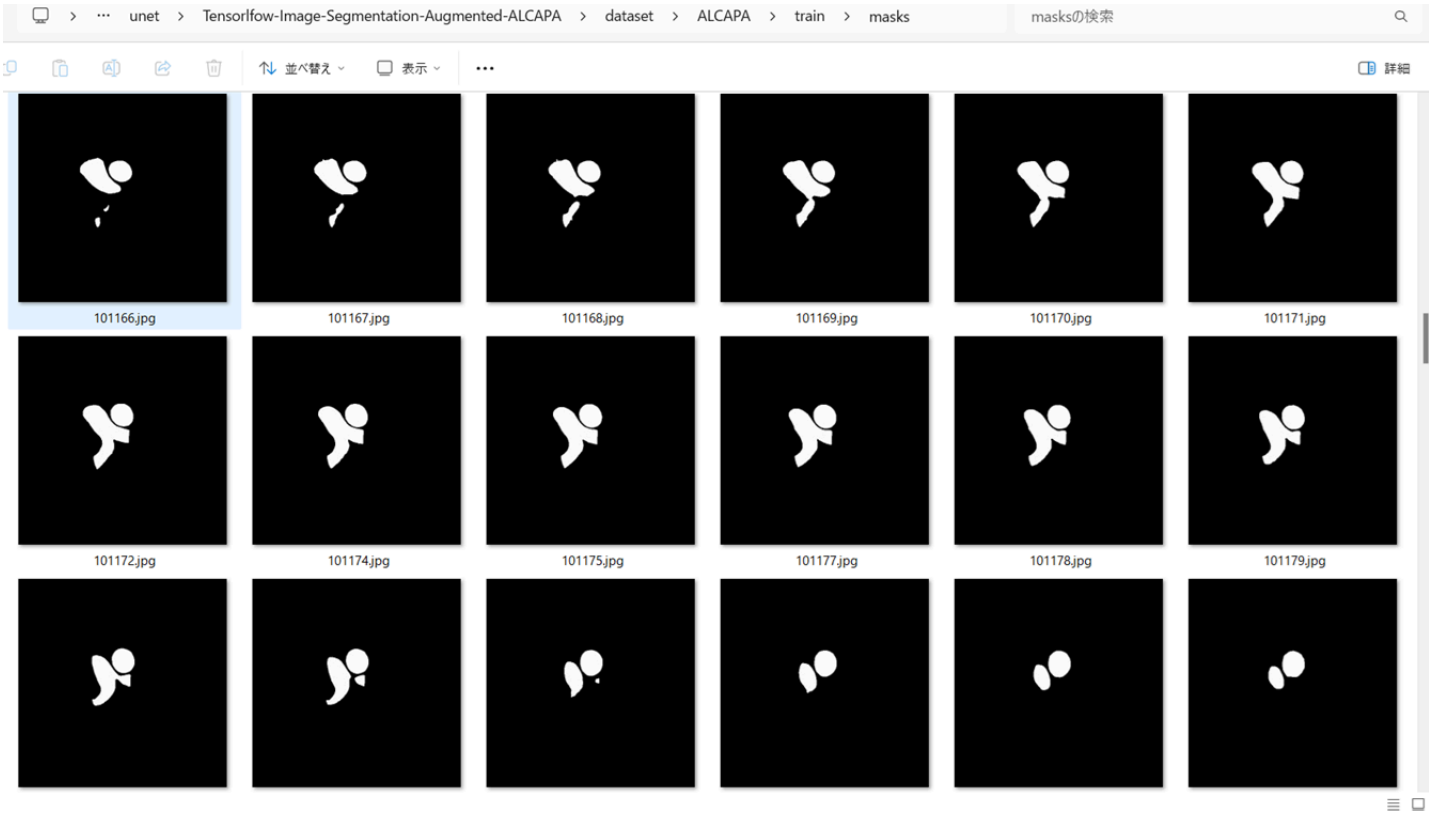


As shown above, the number of images of train and valid datasets is not large enough to use for a training set of our segmentation model. therefore we used an online augmentation tool ImageMaskAugmentor.py to improve generalization performance.

Train_images_sample



Train_masks_sample



3 Train TensorflowUNet Model

We have trained ALCAPATensorflowUNet Model by using the following **train_eval_infer.config** file.
Please move to ./projects/TensorflowSlightlyFlexibleUNet/ALCAPA 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
```

Learning rate
Defined a small learning rate.

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

Online augmentation
Enabled our online augmentation tool.

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

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     = 0.3
reducer_patience   = 4
```

Early stopping callback
Enabled early stopping callback with patience parameter.

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

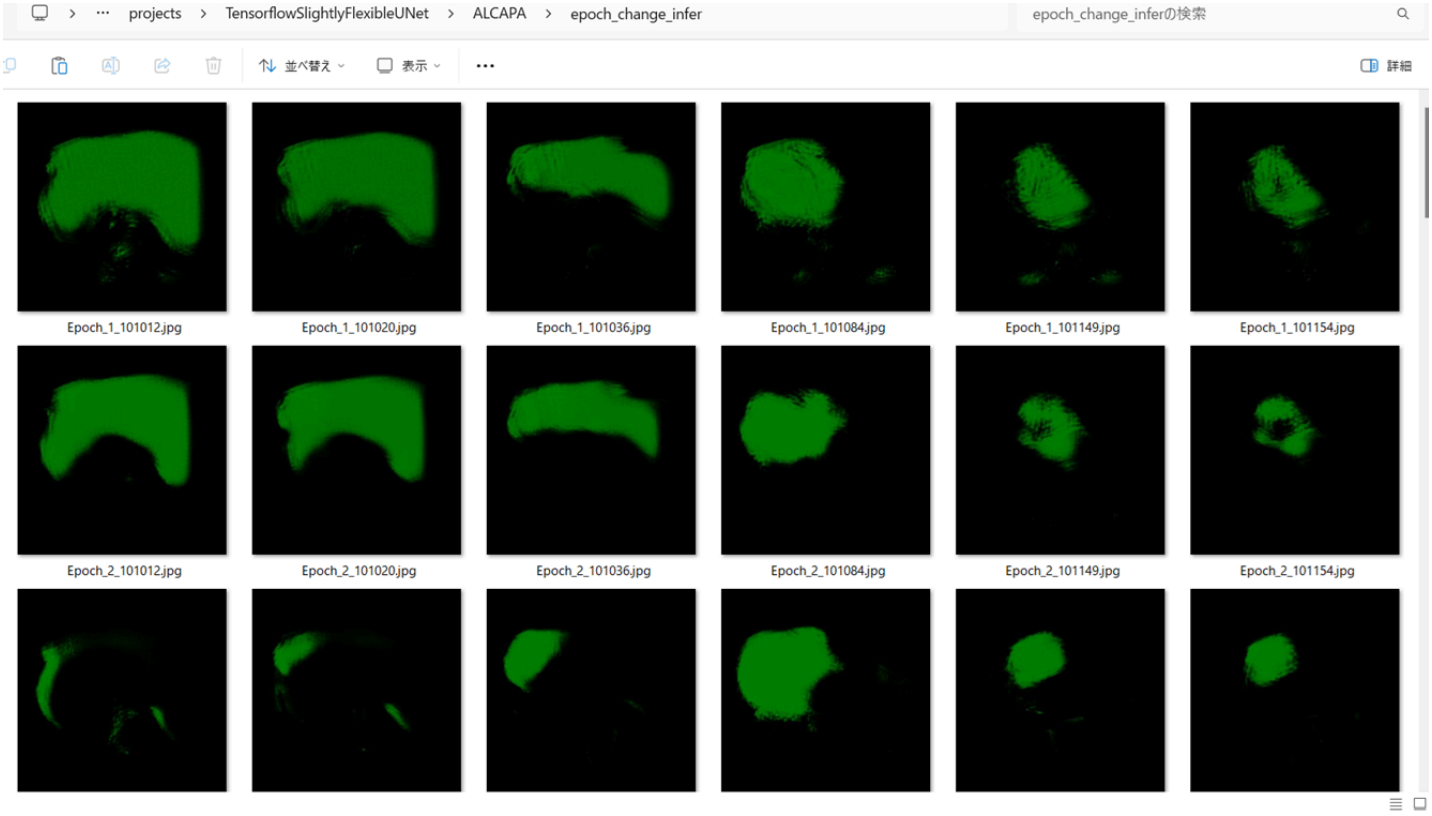
Epoch change inference callbacks

Enabled epoch_change_infer callback.

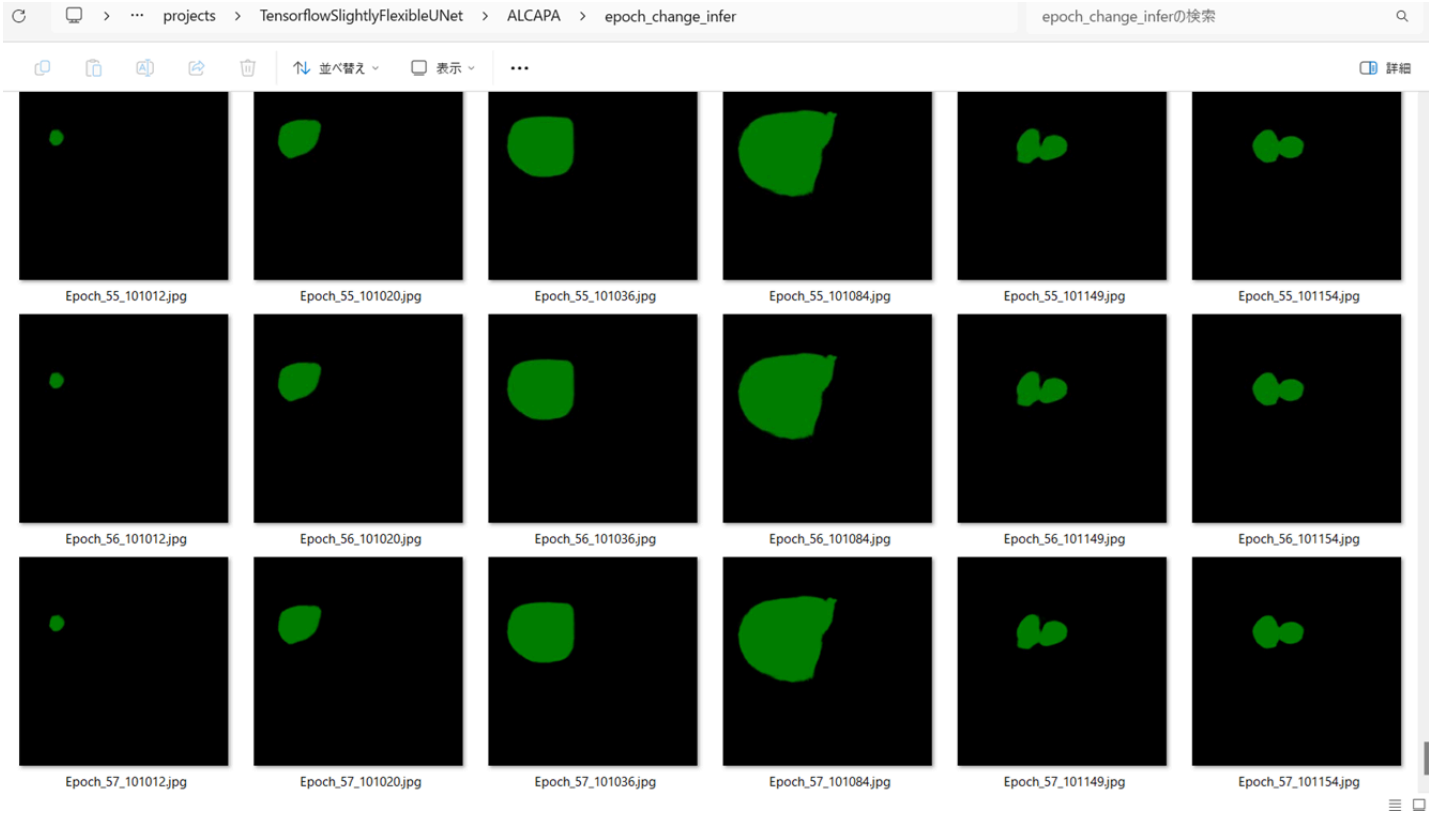
```
[train]
epoch_change_infer       = True
epoch_change_infer_dir   = "./epoch_change_infer"
epoch_changeinfer        = False
epoch_changeinfer_dir    = "./epoch_changeinfer"
num_infer_images         = 6
```

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

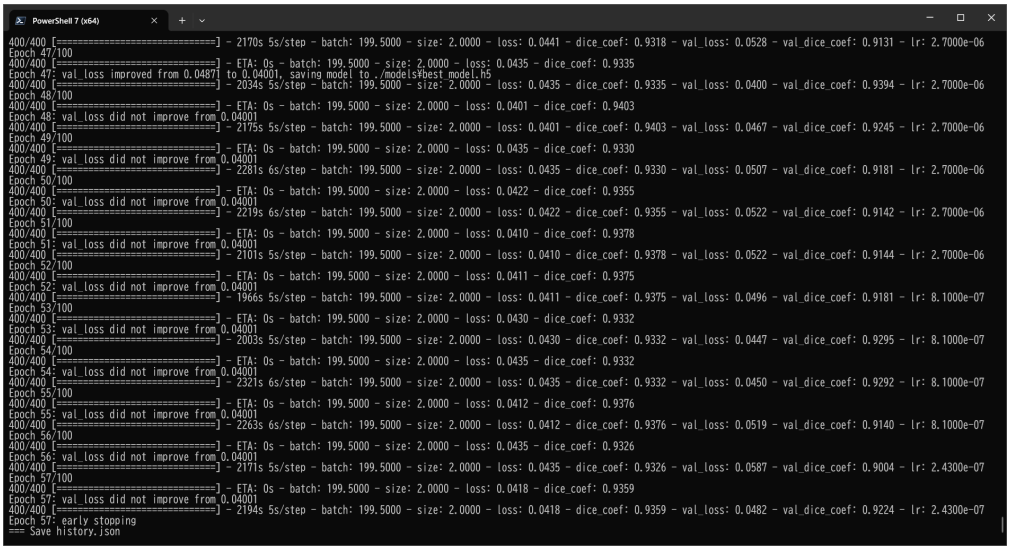
Epoch_change_inference output at starting (1,2,3)



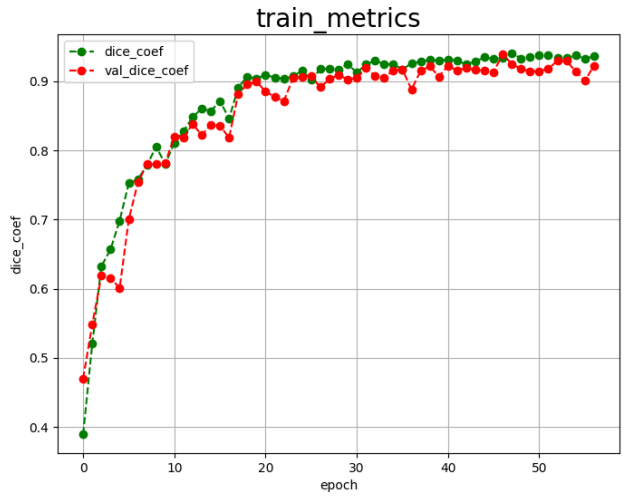
Epoch_change_inference output at ending (55,56,57)



In this experiment, the training process was stopped at epoch 57 by EarlyStopping Callback.



train_metrics.csv



train_losses.csv

4 Evaluation

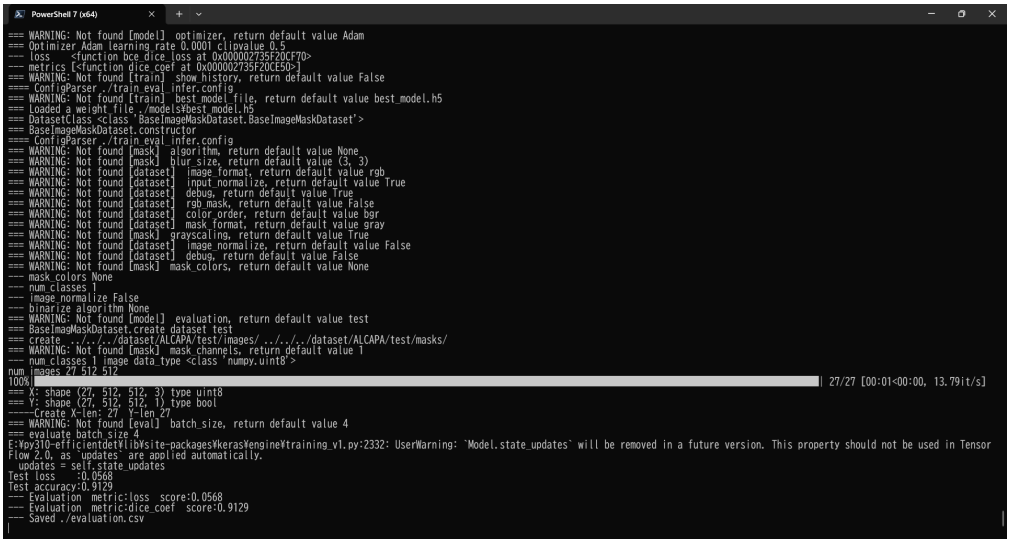
Please move to a **./projects/TensorflowSlightlyFlexibleUNet/ALCAPA** folder,
and run the following bat file to evaluate TensorflowUNet model for ALCAPA.

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

This bat file simply runs the following command.

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

Evaluation console output:



evaluation.csv

The loss (bce_dice_loss) to this ALCAPA/test was low, and dice_coef high as shown below.

```
loss,0.0568
dice_coef,0.9129
```

5 Inference

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

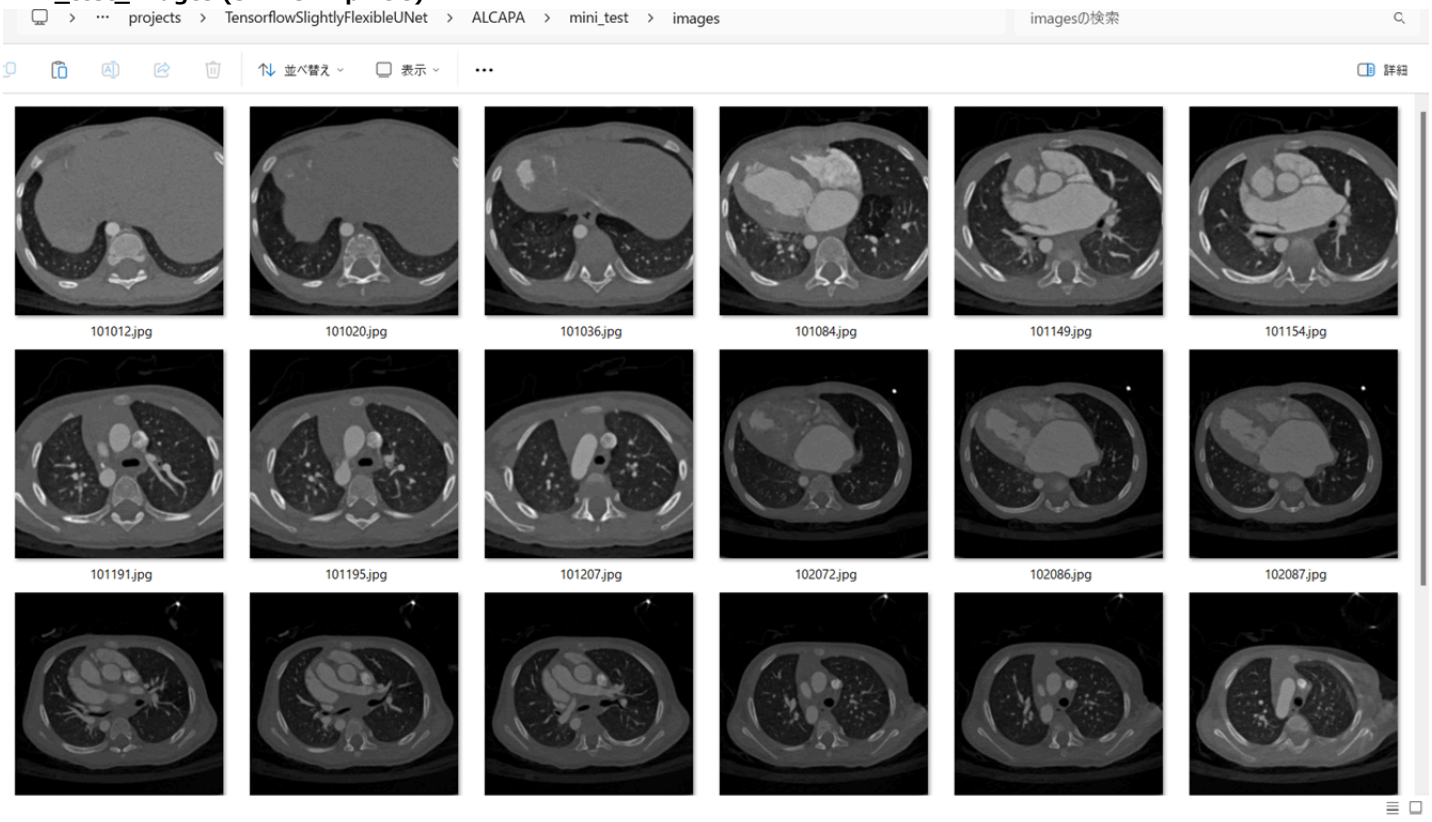
```
./3.infer.bat
```

This simply runs the following command.

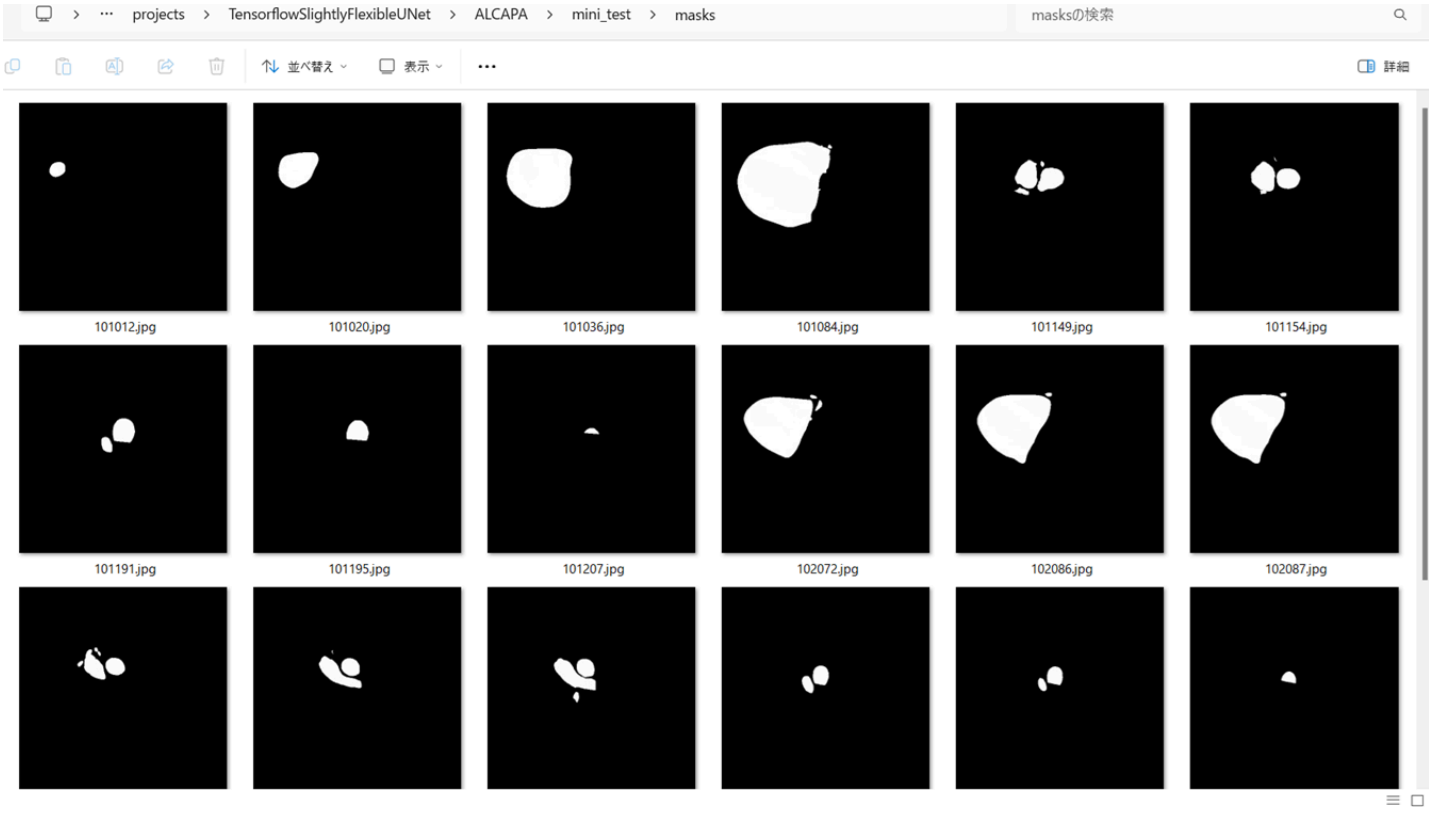
```
python ../../../src/TensorflowUNetInferencer.py ./train_eval_infer_aug.config
```

---

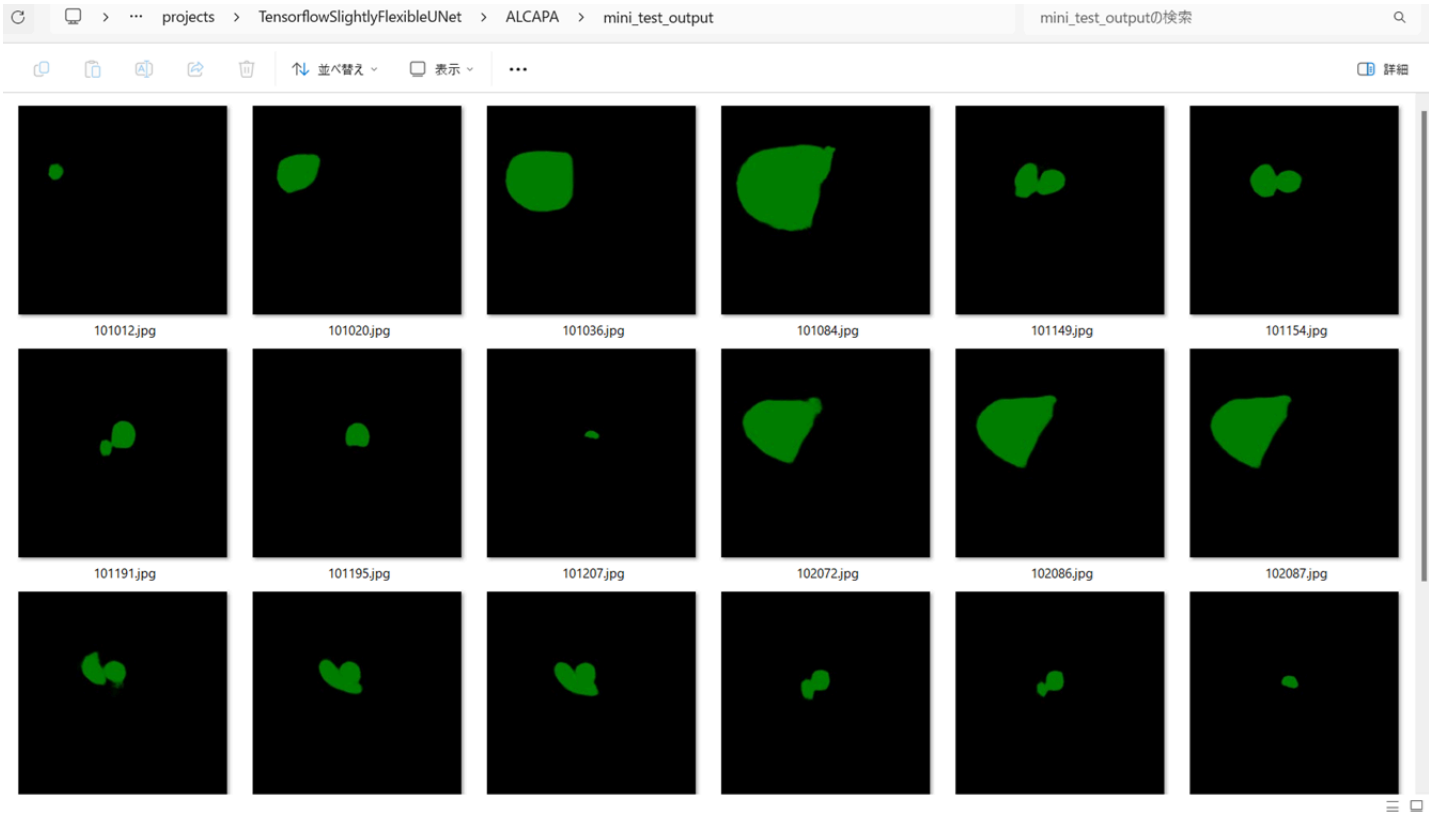
mini_test_images (512x512 pixels)



mini_test_mask(ground_truth)

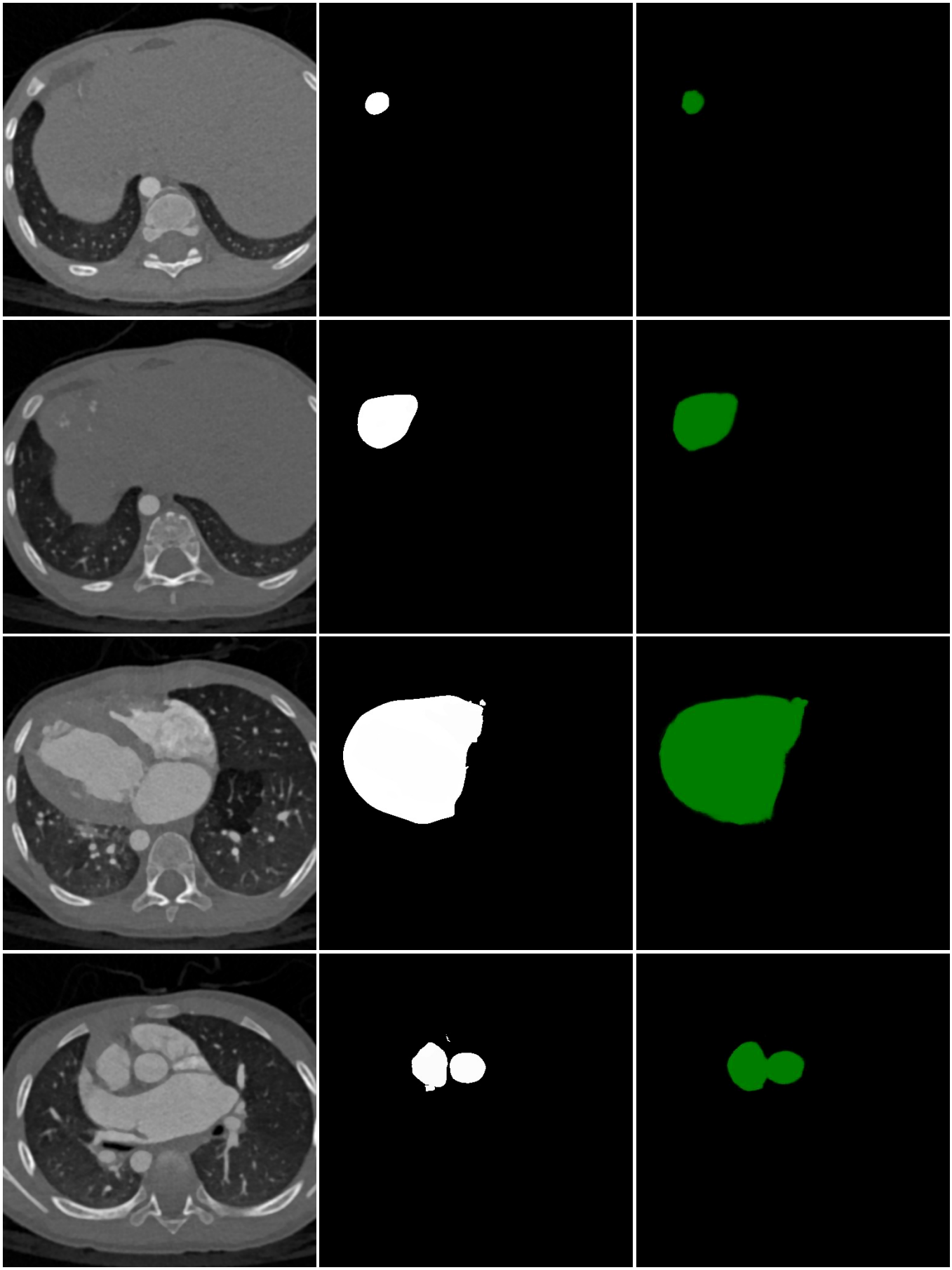


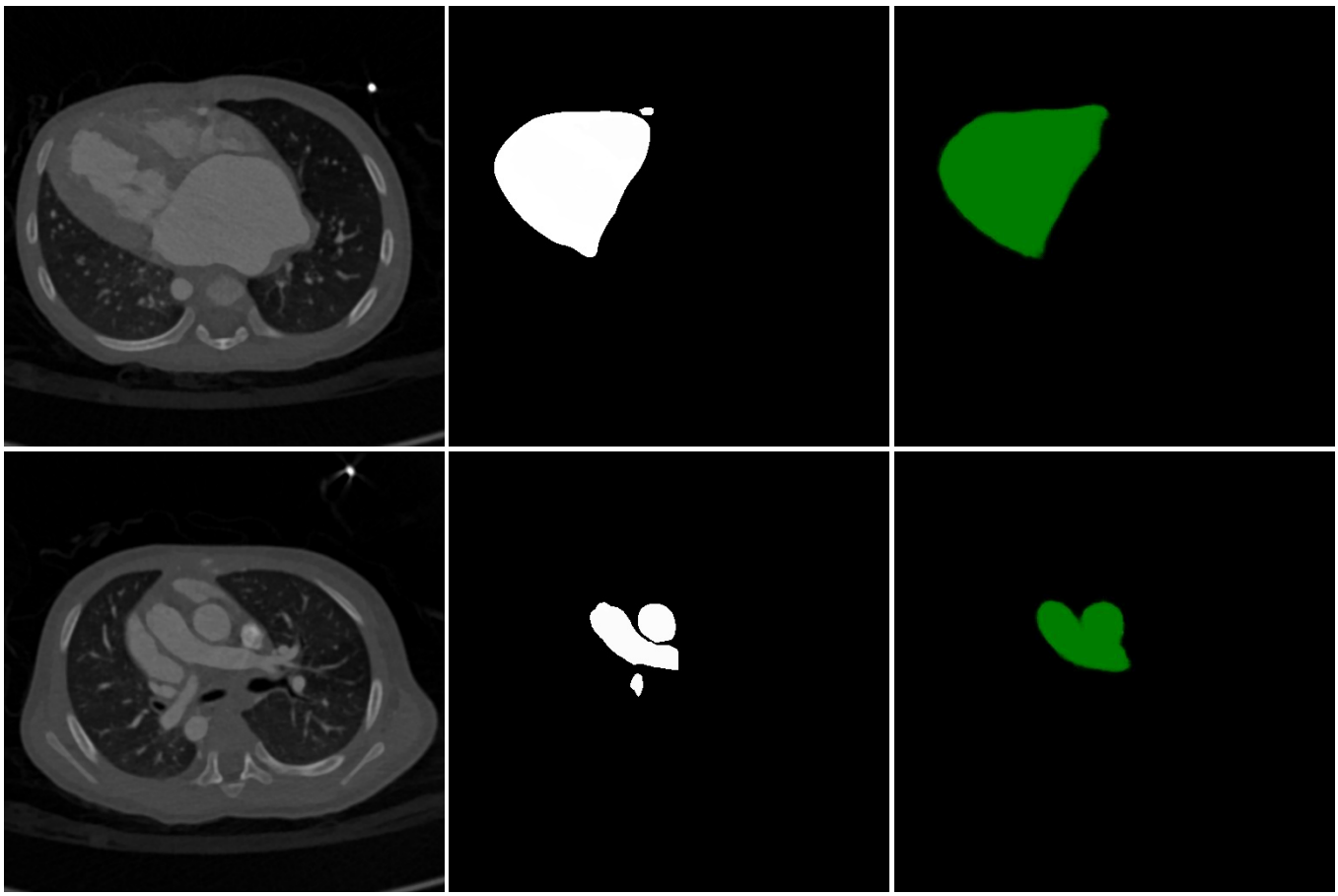
Inferred test masks



Enlarged images and masks (512x512 pixels)

Image	Mask (ground_truth)	Inferred-mask





References

1. ImageALCAPA: A 3D Computed Tomography Image Dataset for Automatic Segmentation of Anomalous Left Coronary Artery from Pulmonary Artery
A. Zeng, C. Mi, D. Pan, Q. Lu and X. Xu,
2022 IEEE International Conference on Bioinformatics and Biomedicine (BIBM),
Las Vegas, NV, USA, 2022, pp. 1800-1803,
doi: 10.1109/BIBM55620.2022.9994951.
https://ieeexplore.ieee.org/document/9994951

2. Anomalous Left Coronary Artery From the Pulmonary Artery
https://www.chop.edu/conditions-diseases/anomalous-left-coronary-artery-pulmonary-artery