TensorFlow Image Segmentation for Pre Augmented Coronary Artery Disease based on Tensorflow-Image-Segmentation-API

Apache-2.0 license

0 stars 0 forks 2 watching Branches Activity
Tags

Public repository

main 1 Branch 0 Tags Go to file t Go to file Add file Code

sarah-antillia Initial commit 8525e9d · 12 minutes ago

generator	Initial commit	10 hours ago
projects/TensorflowSlightlyFlexibleUNet/...	Initial commit	10 hours ago
src	Initial commit	10 hours ago
.gitignore	Initial commit	10 hours ago
LICENSE	Initial commit	10 hours ago
README.md	Initial commit	12 minutes ago
sarah-antillia_Tensorflow-Image-Segment...	Initial commit	3 hours ago

README Apache-2.0 license

Tensorflow-Image-Segmentation-Pre-Augmented-Coronary-Artery-Disease (2025/01/20)

This is the first experiment of Image Segmentation for Coronary Artery Disease Stenosis based on the Tensorflow-Image-Segmentation-API, and a pre-augmented Stenosis-ImageMaskDataset.zip, which is derived by us from the stenosis train dataset of **ARCADE: Automatic Region-based Coronary Artery Disease diagnostics using x-ray angiography imagEs Dataset Phase 1**

Dataset Augmentation Strategy
To address the limited size of ARCADE stenosis train segmentation dataset, which contains 1000 images and their corresponding binary masks in **dataset_phase_1/stenosis_dataset**, we employed an offline augmentation tool to generate a pre-augmented dataset, which supports the following augmentation methods.

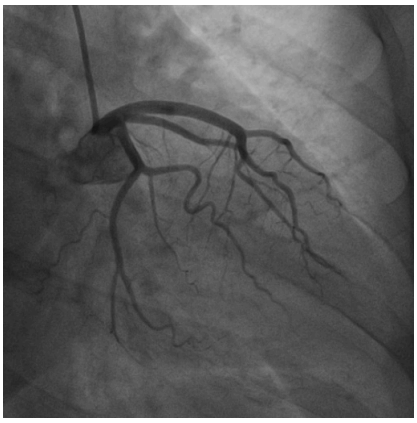
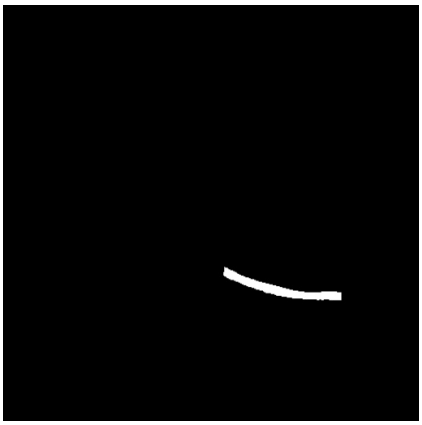
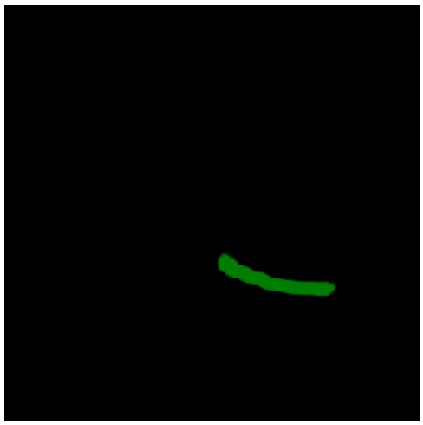
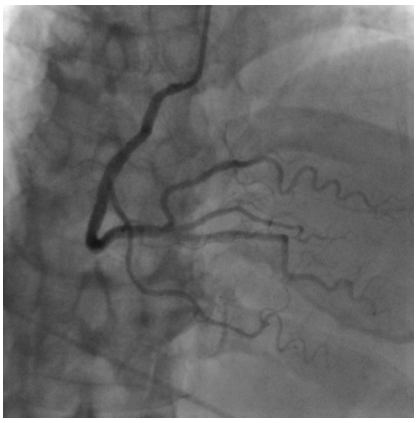
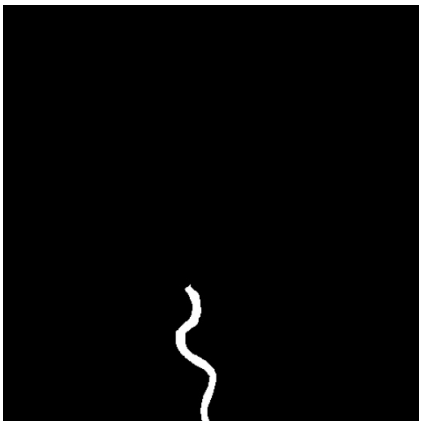
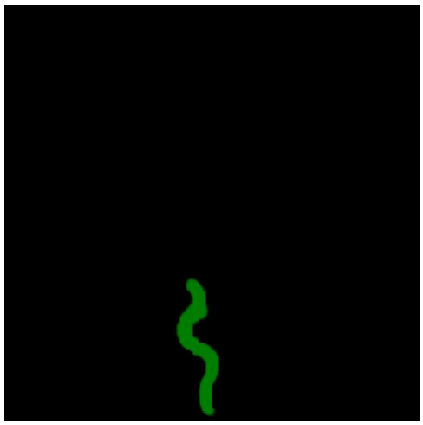
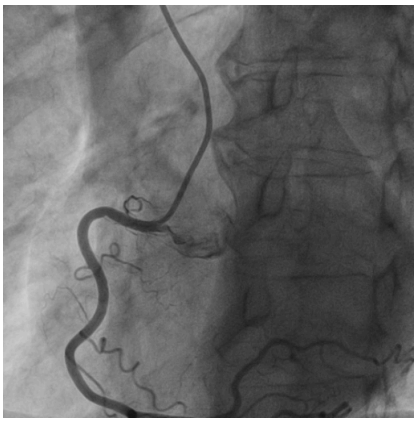
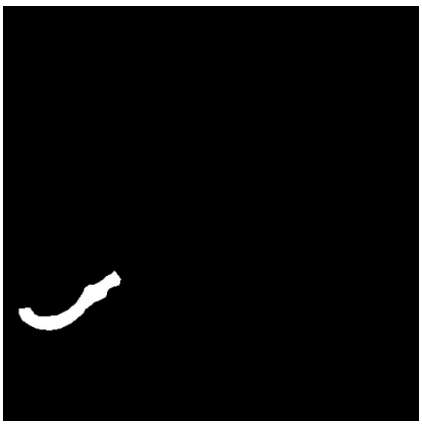
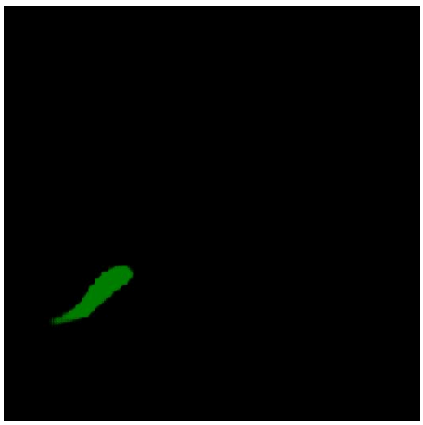
- Vertical flip
- Horizontal flip
- Rotation
- Shrinks
- Shears
- Deformation
- Distortion
- Barrel distortion
- Pincushion distortion

Please see also the following tools

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

Actual Image Segmentation for Images of 512x512 pixels

The inferred colorized masks predicted by our segmentation model appear 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 Stenosis Segmentation.
As shown in Tensorflow-Image-Segmentation-API. you may try other more advanced TensorFlow UNet Models to get better segmentation models:

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

1. Dataset Citation

The datase used here has been taken from the following zenodo.org web site.

ARCADE: Automatic Region-based Coronary Artery Disease diagnostics using x-ray angiography imagEs Dataset Phase 1
Maxim Popov, Akmaral Am anturdieva, Nuren Zhaksylyk, Alsabir Alkanov, Adilbek Saniyazbekov,
Temirgali Aimyshev, Eldar Ismailov, Ablay Bulegenov, Alexey Kolesnikov, Aizhan Kulanbayeva,
Arystan Kuzhukeyev, Orazbek Sakhov,Almat Kalzhanov, Nurzhan Temenov, Siamac Fazli1
https://zenodo.org/records/7981245

2 ImageMask Dataset

If you would like to train this Stenosis Segmentation model by yourself, please download the dataset from the google drive Stenosis-ImageMaskDataset.zip,
, which is a pre-augmented dataset from the original **dataset_phase_1/stenosis_dataset**.
Please expand the downloaded ImageMaskDataset and place it under **./dataset** folder to be

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

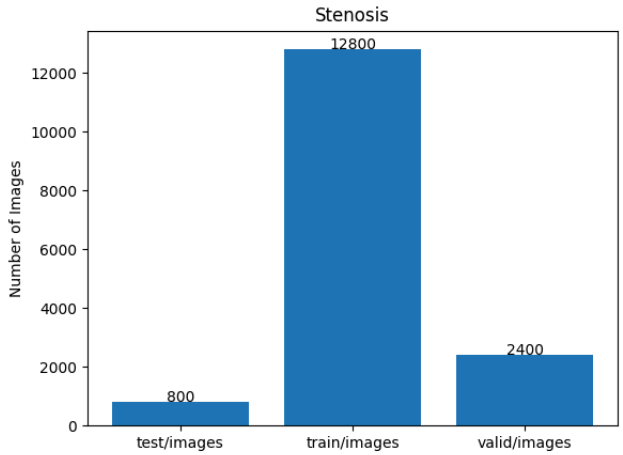
The foloder structure of the original **dataset_phase_1** of ARCADE is the following.

```
./dataset_phase_1
 └--stenosis_dataset
    ├─sten_train
    ├─annotations
    │ └─images
    └─sten_val
      ├─annotations
      └─images
```

We used the two Python scripts to generate our ImageMask-Dataset from **dataset_phase_1**.

- ImageMaskDatasetGenerator.py
- split_master.py

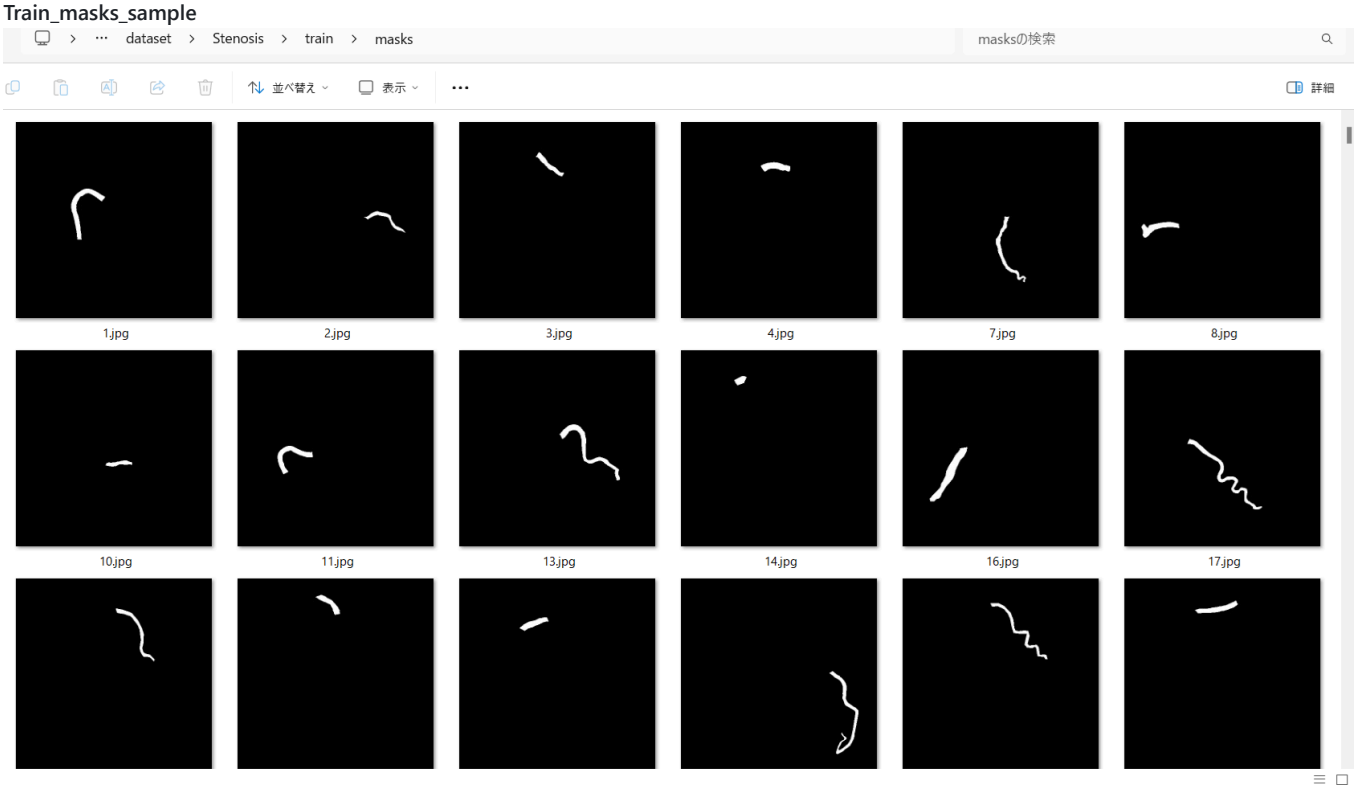
Stenosis Dataset Statistics



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

Train_images_sample

Train_masks_sample



4 Train TensorflowUNet Model

We trained Stenosis TensorflowUNet Model by using the following **train_eval_infer.config** file.
Please move to ./projects/Stenosis and run the following bat file for Python script TensorflowUNetTrainer.py.

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

, which simply runs the following command.

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

Model parameters

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

```
base_filters   = 16
base_kernels   = (7,7)
num_layers     = 7
dilation       = (3,3)
```

Online augmentation

Disabled our online augmentation. To enable the augmentation, set generator parameter to True.

```
[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.

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

Early stopping callback

Enabled early stopping callback with patience parameter.

```
[train]
patience      = 20
```

Epoch change inference callback

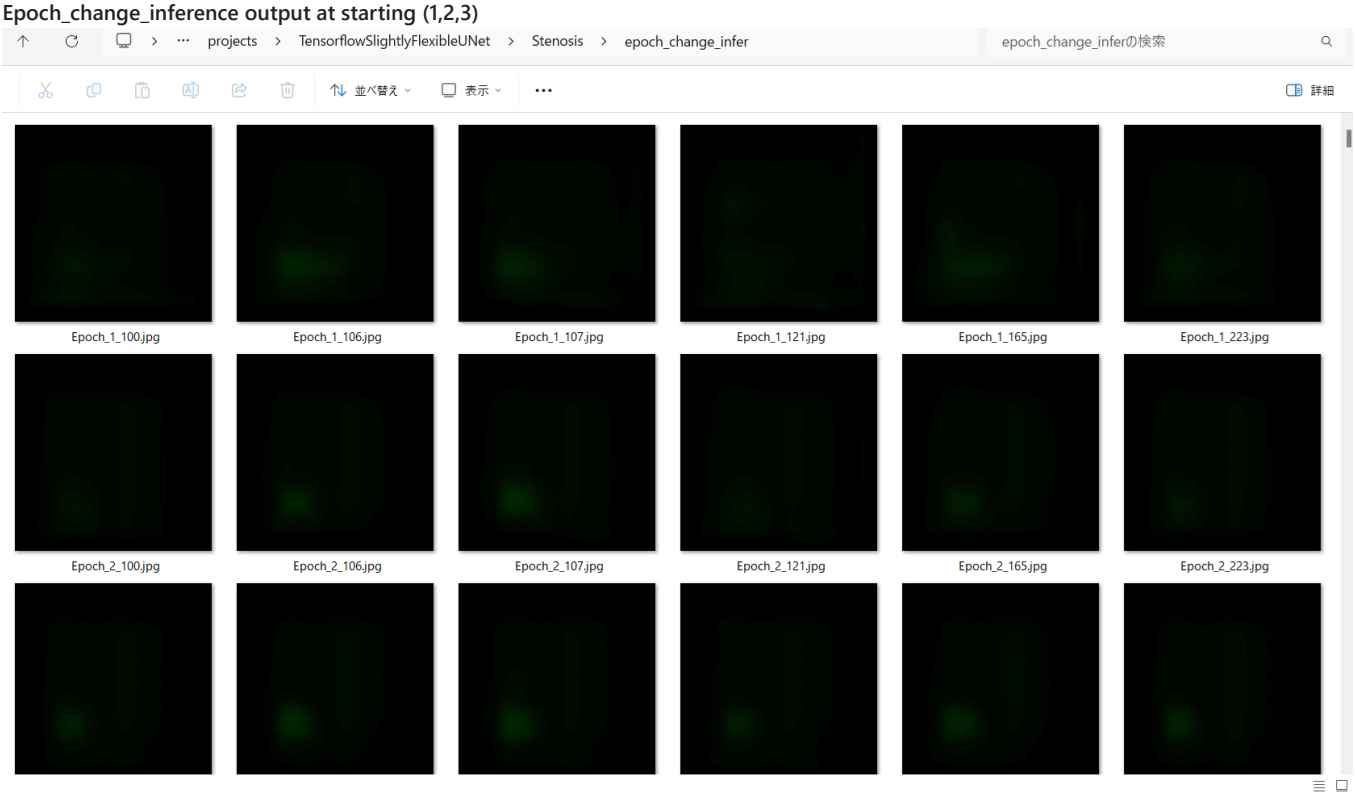
Enabled EpochChange infer callback.

```
Enabled color space converter
```

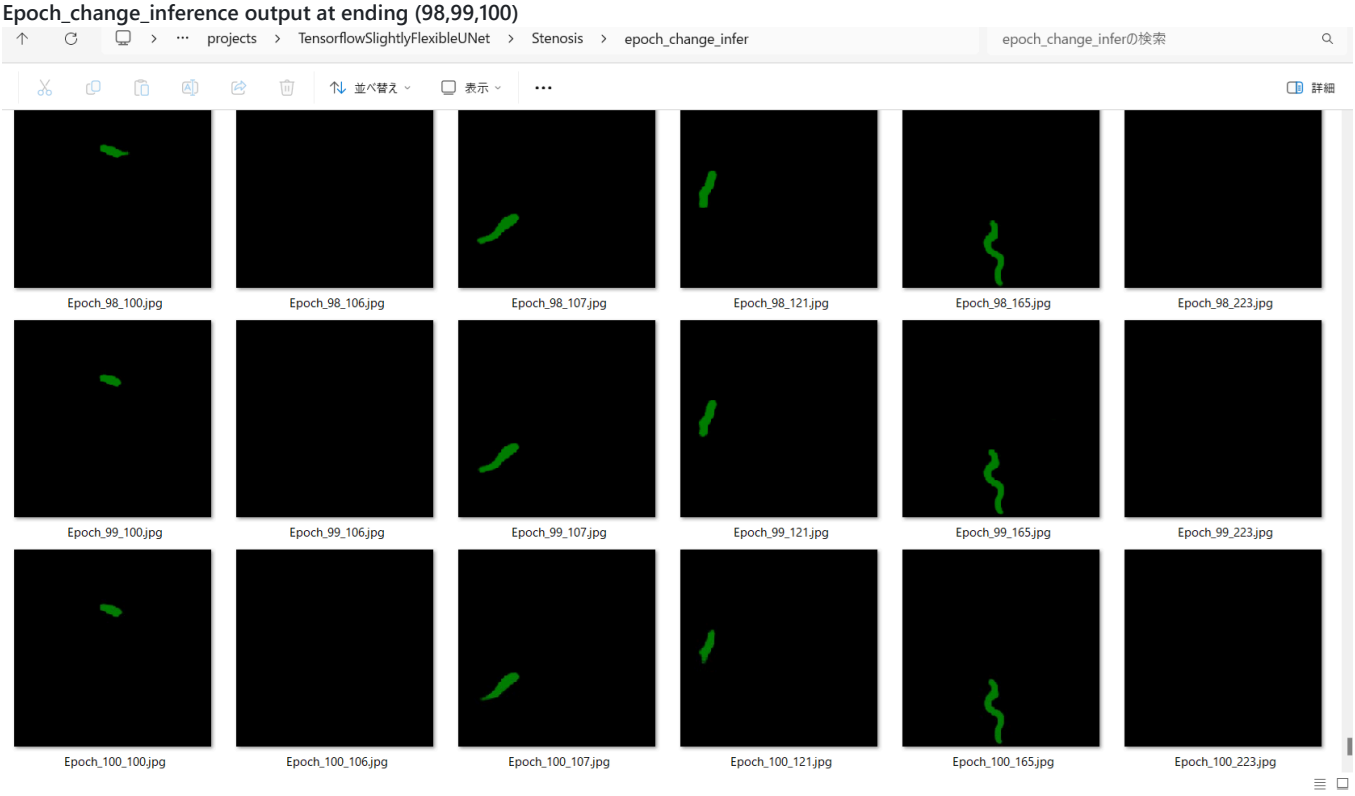
```
[train]
epoch_change_infer     = True
epoch_change_infer_dir =  "./epoch_change_infer"
num_infer_images       = 6
```

By using this EpochChangeInference 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 (1,2,3)

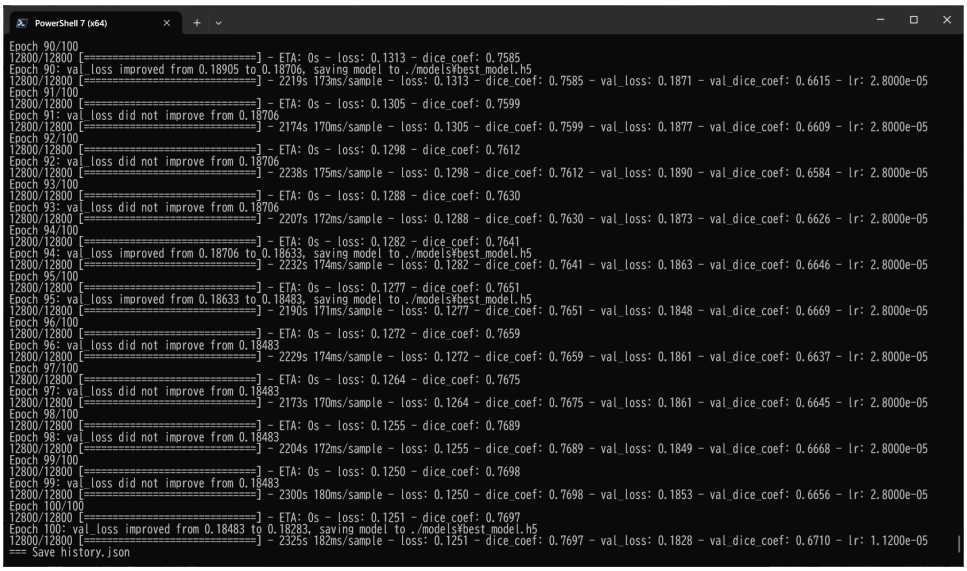


Epoch_change_inference output at ending (98,99,100)

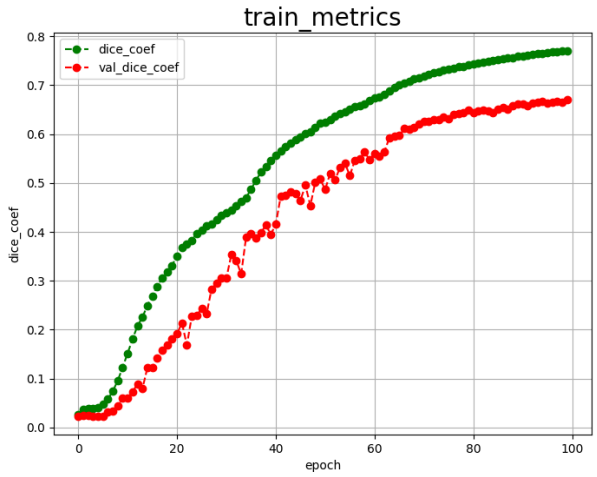


In this case, the training process was terminated at epoch 100..

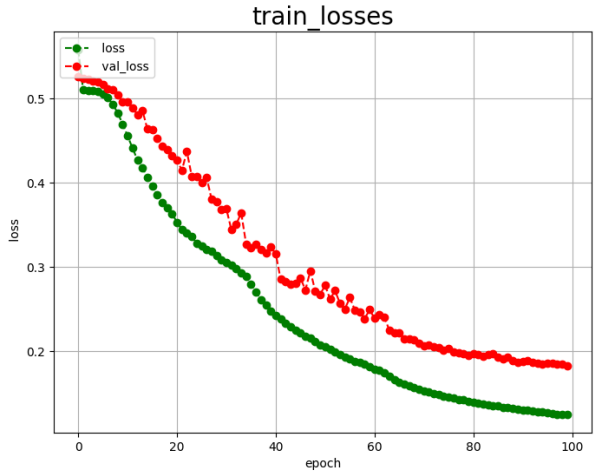
Training console output



[train_metrics.csv](#)



[train_losses.csv](#)



5 Evaluation

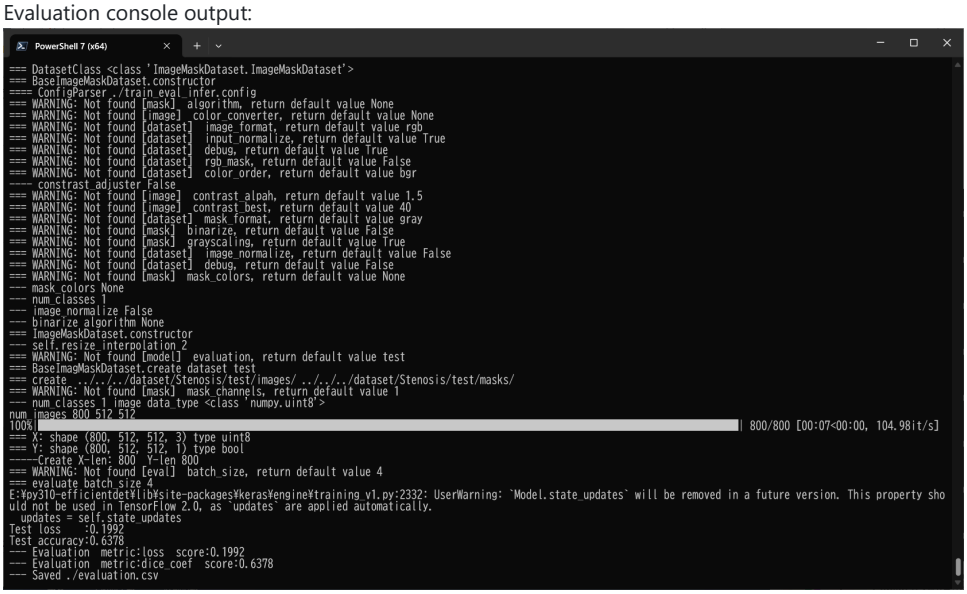
Please move to a **./projects/TensorflowSlightlyFlexibleUNet/Stenosis** folder,
and run the following bat file to evaluate TensorflowUNet model for Stenosis.

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

This bat file simply runs the following command.

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

Evaluation console output:



The loss (bce_dice_loss) score for this test dataset was not low, and dice_coef not high as shown below.

```
loss,0.1992
dice_coef,0.6378
```

6 Inference

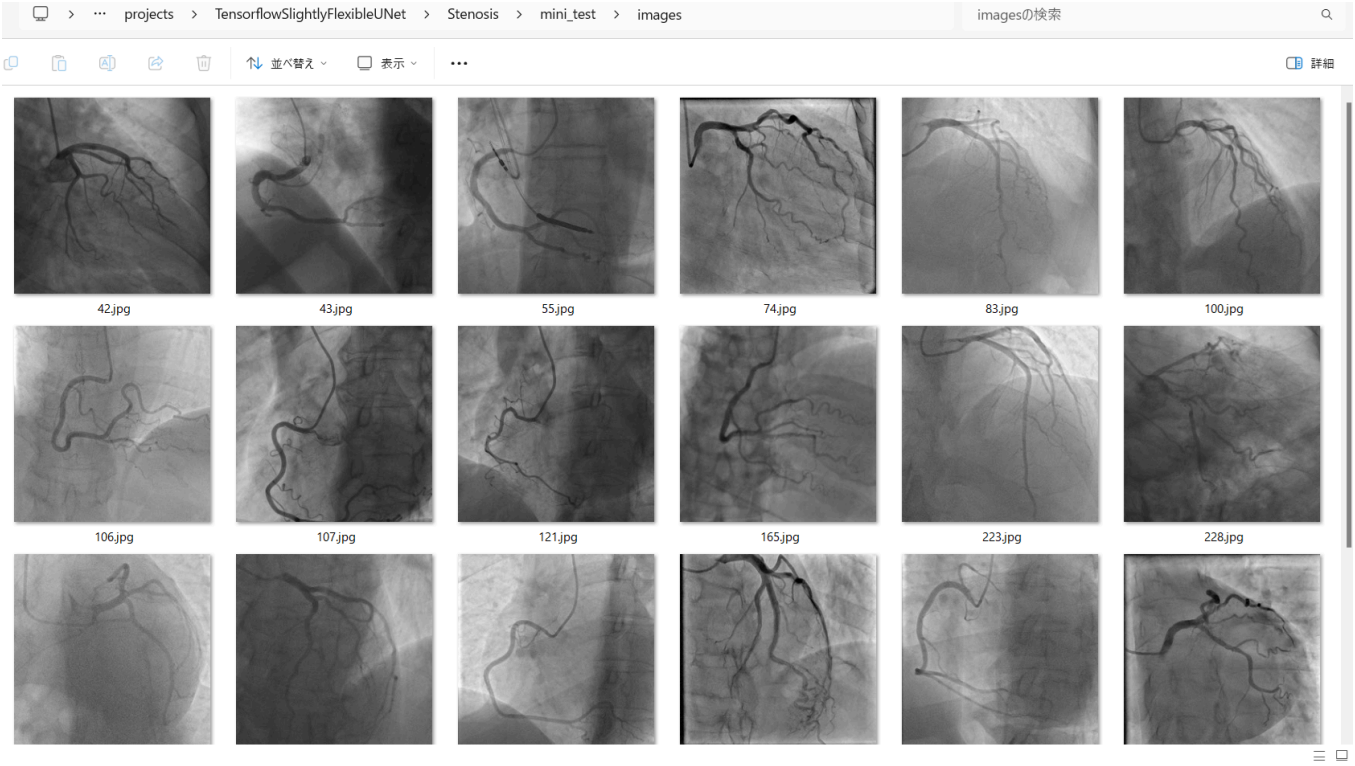
Please move to a **./projects/TensorflowSlightlyFlexibleUNet/Stenosis** folder
,and run the following bat file to infer segmentation regions for images by the Trained-TensorflowUNet model for Stenosis.

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

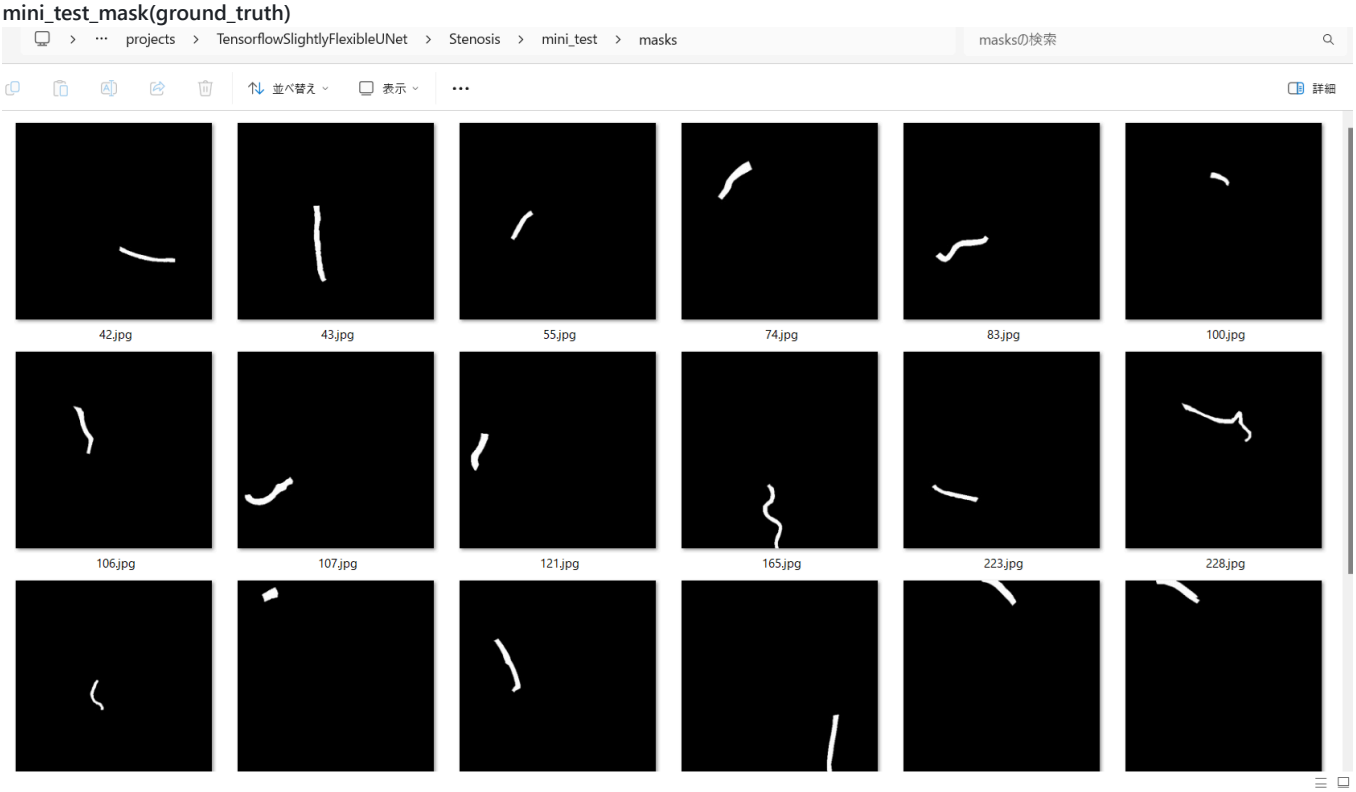
This simply runs the following command.

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

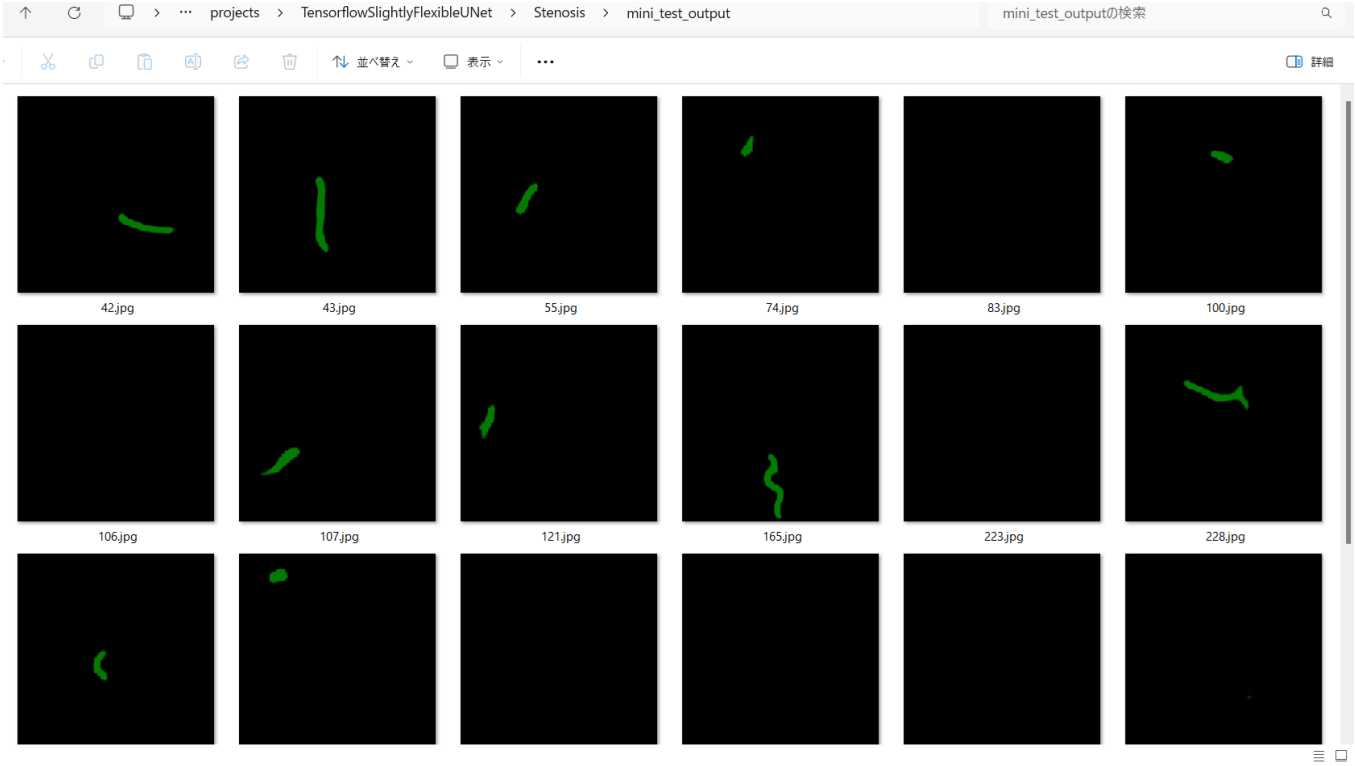
mini_test_images



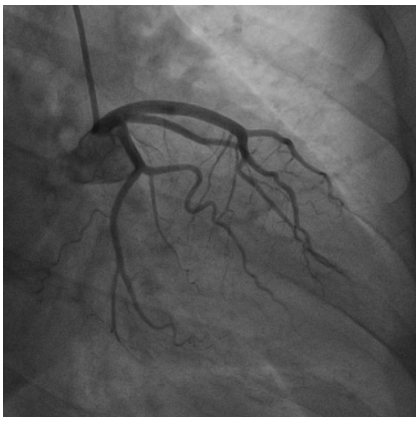
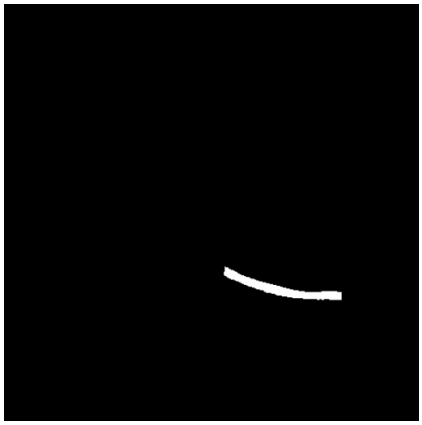
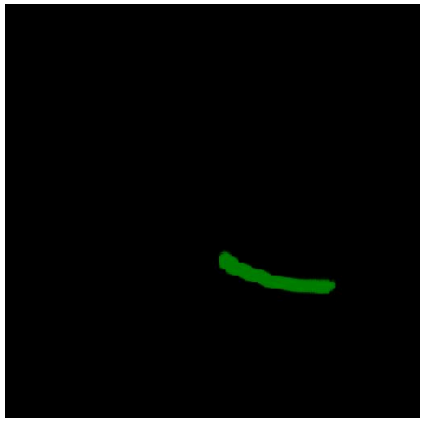
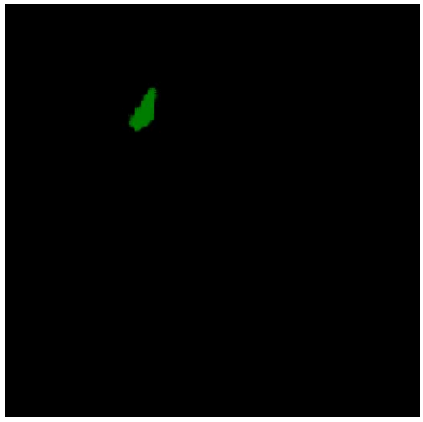
mini_test_mask(ground_truth)



Inferred test masks



Enlarged images and masks

Image	Mask (ground_truth)	Inferred-mask
		
		

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

Languages

Python 99.9% Batchfile 0.1%

Suggested workflows

Based on your tech stack

Django
Build and Test a Django Project
Configure

Python application
Create and test a Python application.
Configure

Pylint
Lint a Python application with pylint.
Configure

More workflows
Dismiss suggestions