

 sarah-antillia / [Tensorflow-Image-Segmentation-Pre-Augmented-Coronary-Artery-Disease](#)

[Code](#) [Issues](#) [Pull requests](#) [Actions](#) [Projects](#) [Wiki](#) [Security](#) [Insights](#) [Settings](#)

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 4894134 · 6 hours ago

generator	Initial commit	6 hours ago
projects/TensorflowSlightlyFlexibleUNet/...	Initial commit	6 hours ago
src	Initial commit	6 hours ago
.gitignore	Initial commit	6 hours ago
LICENSE	Initial commit	7 hours ago
README.md	Initial commit	6 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 a train dataset of ARCADE [ARCADE](#).

Dataset Augmentation Strategy

To address the limited size of ARCADE stenosis train segmentation dataset, which contains 1000 images and their corresponding binary masks in human oesophagus of NulnsSeg 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

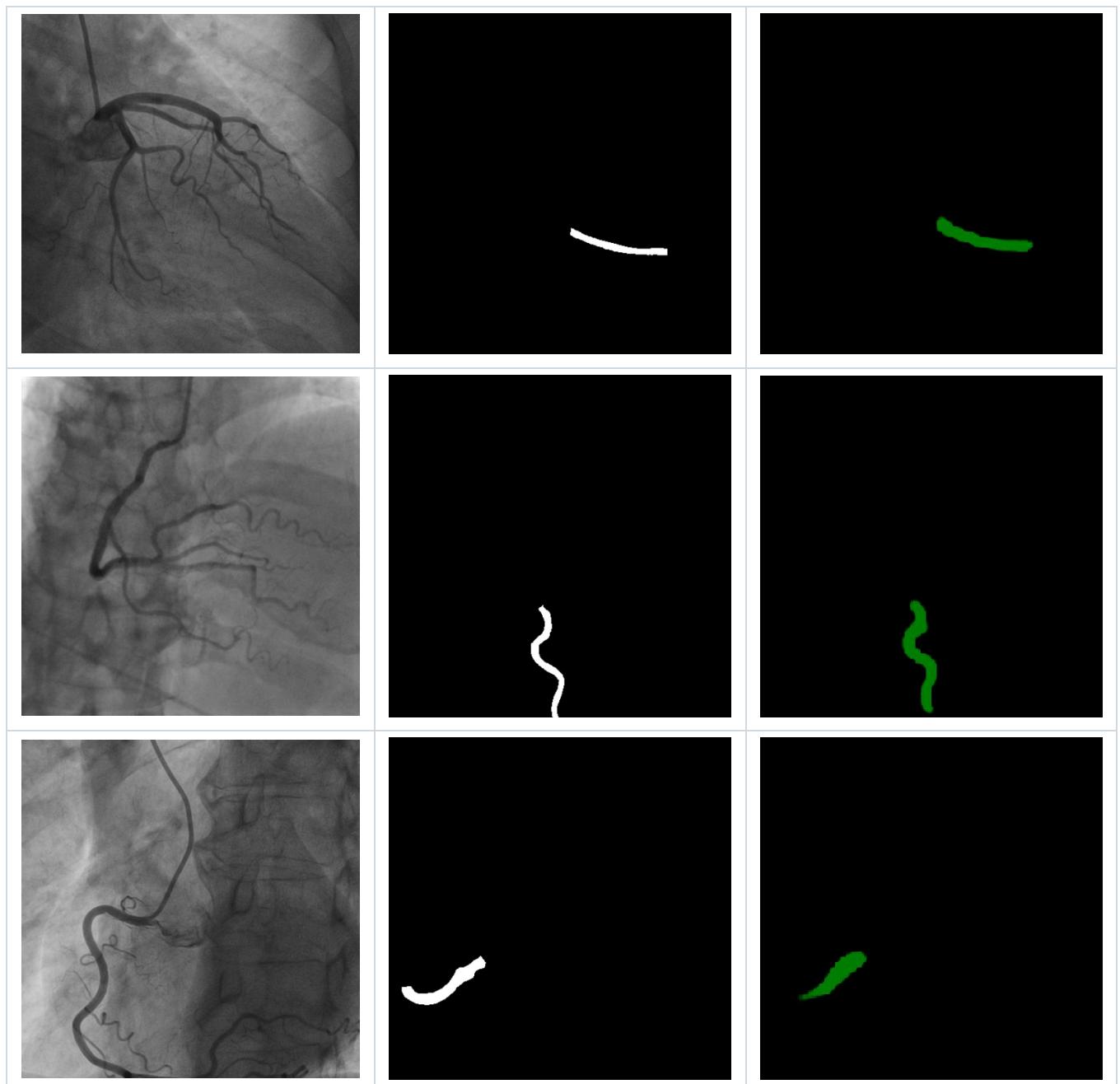
Please see also the following tools

- [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 dataset 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, Akmara Am anturdieva, Nuren Zhaksylyk, Alsabir Alkanov, Adilbek Saniyazbekov, Temirgali Aimyshev, Eldar Ismailov, Ablay Bulegenov, Alexey Kolesnikov, Aizhan Kulambayeva, Arystan Kuzhukayev, 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
```

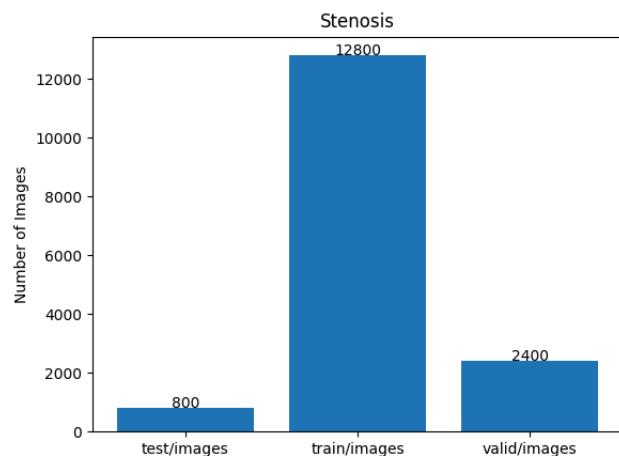
The foloder structure of the original **dataset_phase_1** of ARCADE is the following.

```
./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**.

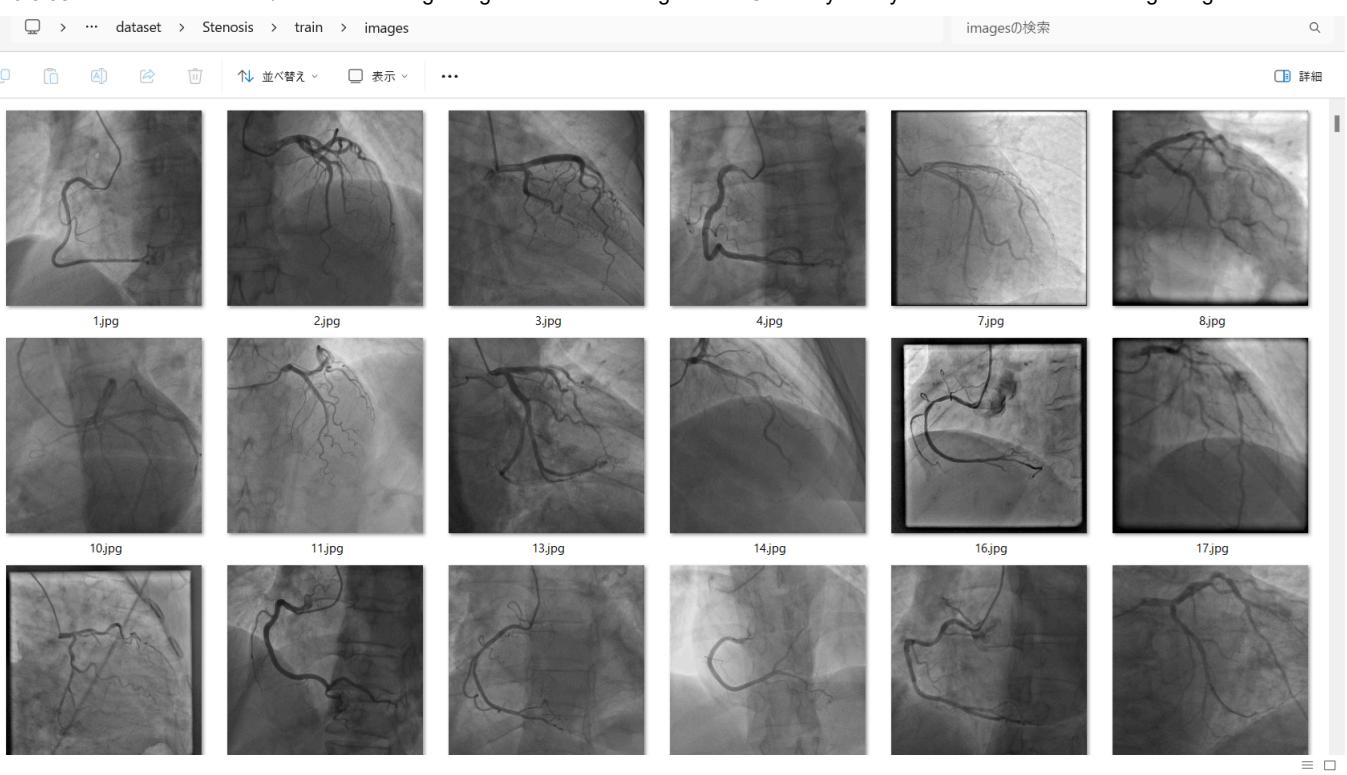
- [ImageMaskDatasetGenerator.py](#)
- [split_master.py](#)

Stenosis Dataset Statistics

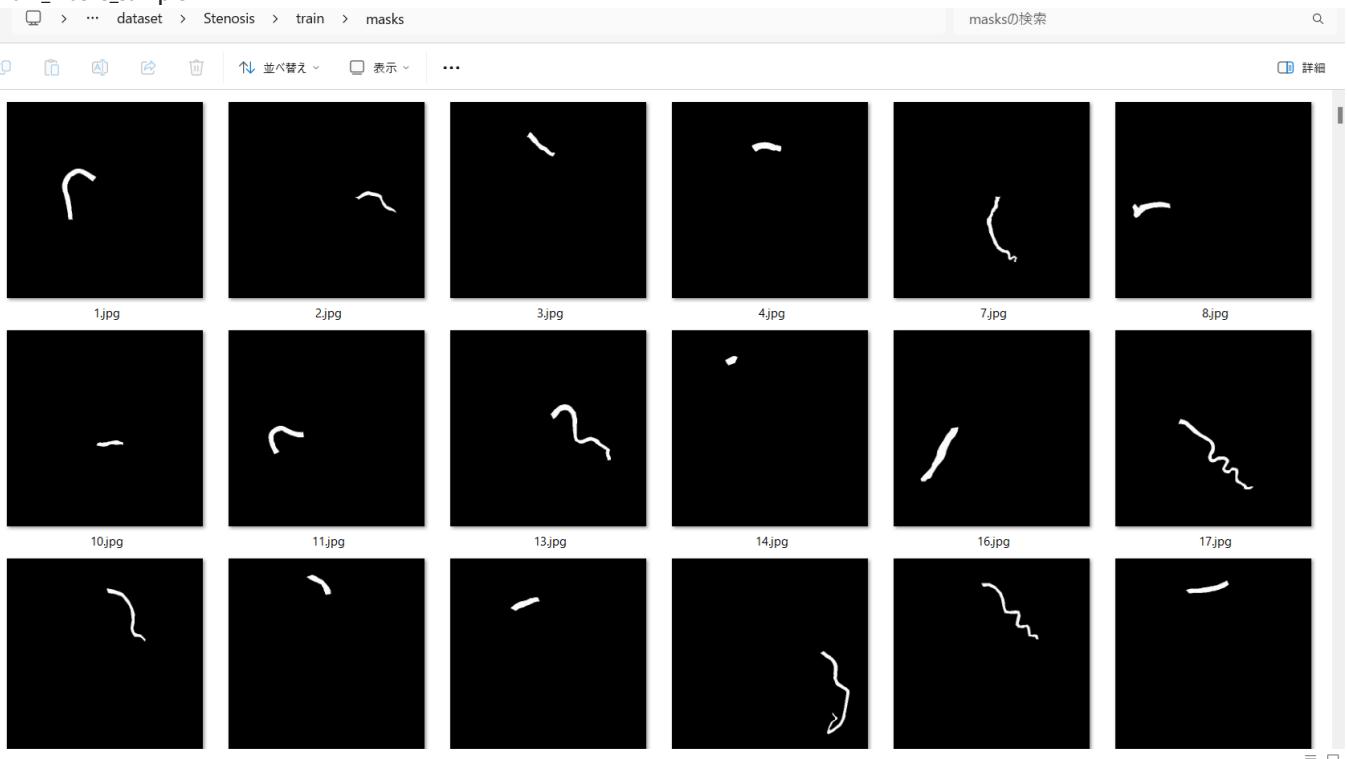


As shown above, the number of images of train and valid dataset is enough large to use for a training dataset of our segmentaion model.

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 learning_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

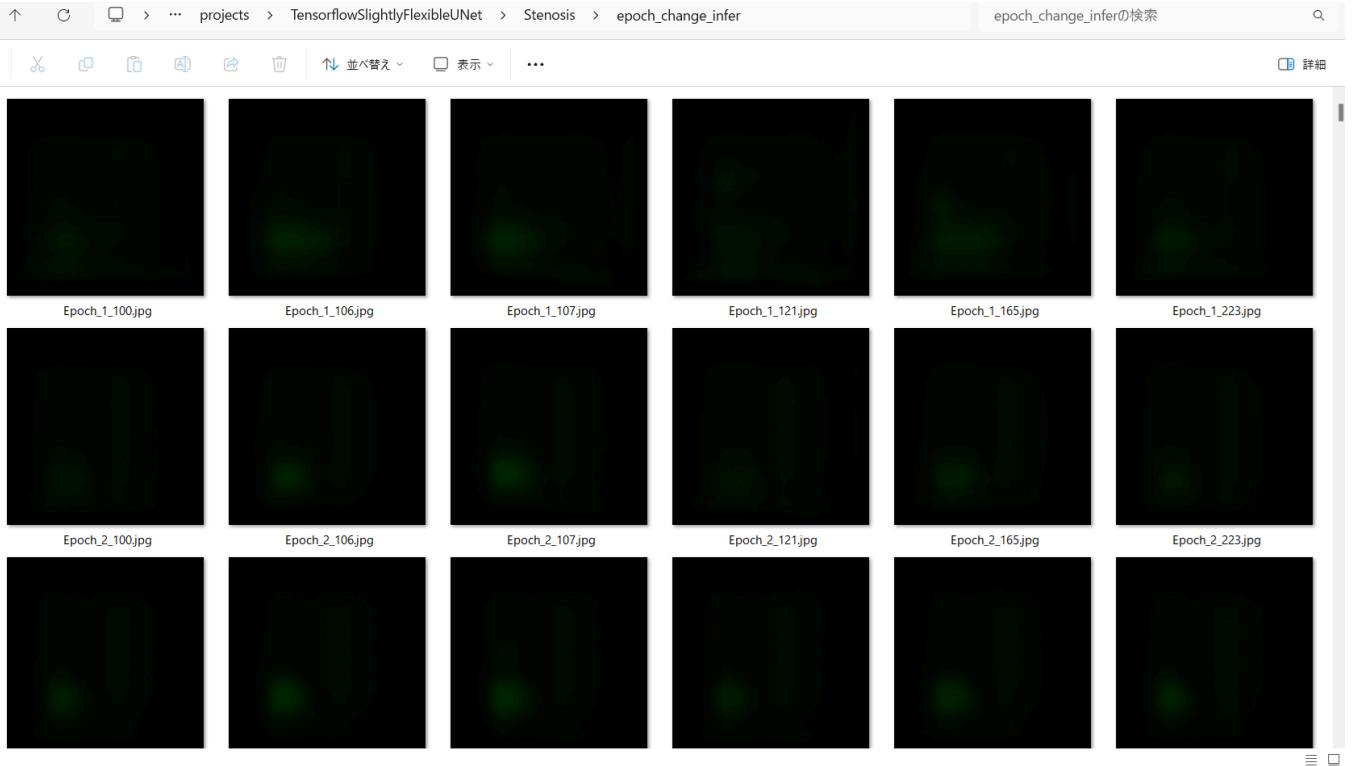
Enabled EpochChange infer callback.

Enabled color space converter

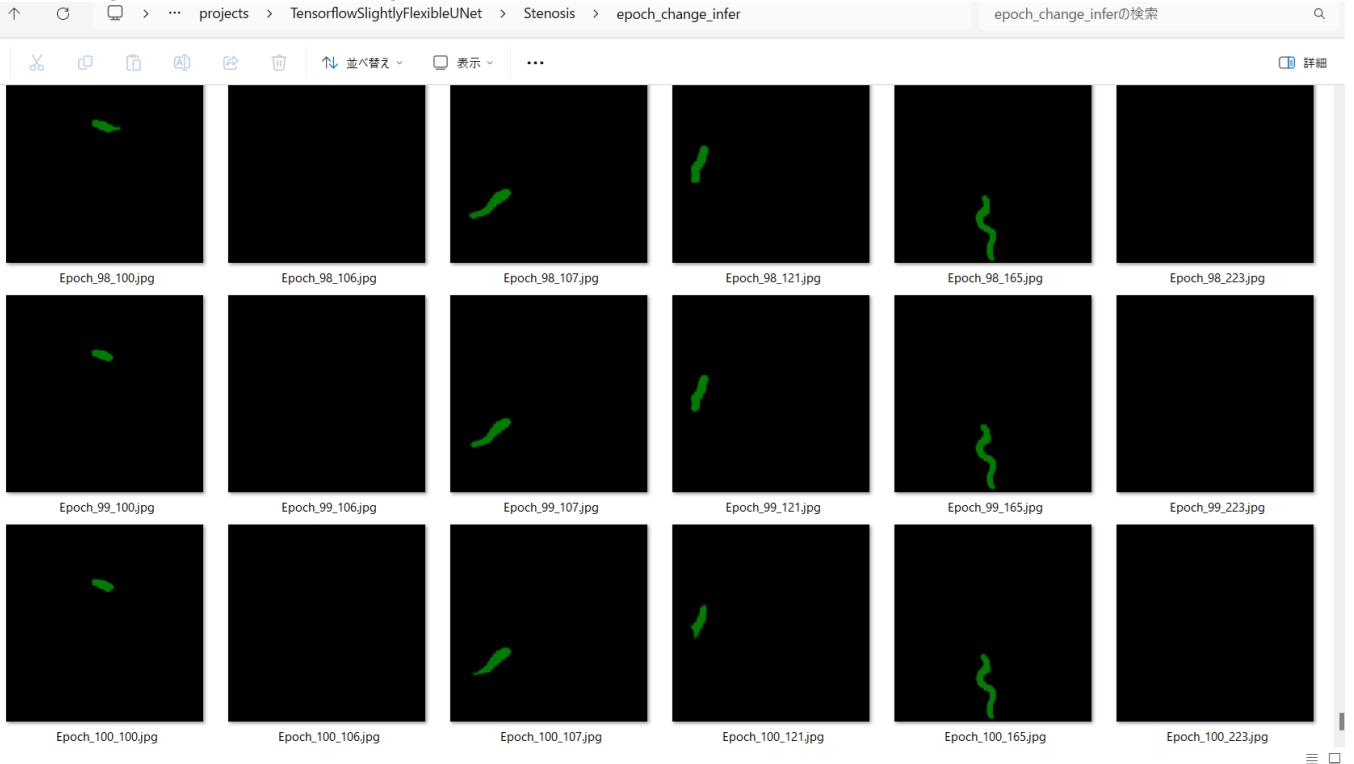
```
[train]
epoch_change_infer    = True
epoch_change_infer_dir = "./epoch_change_infer"
num_infer_images       = 6
```

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..

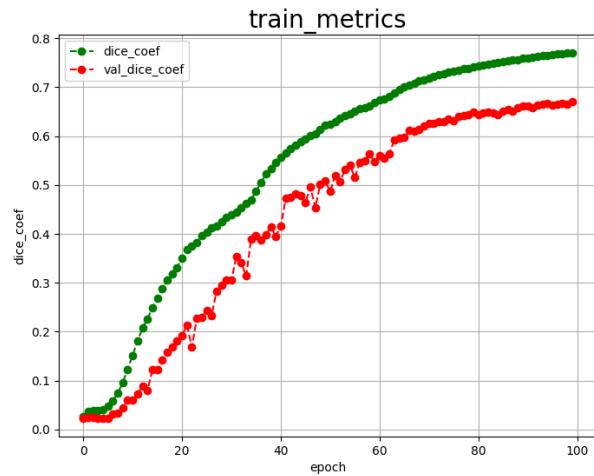
Training console output

```

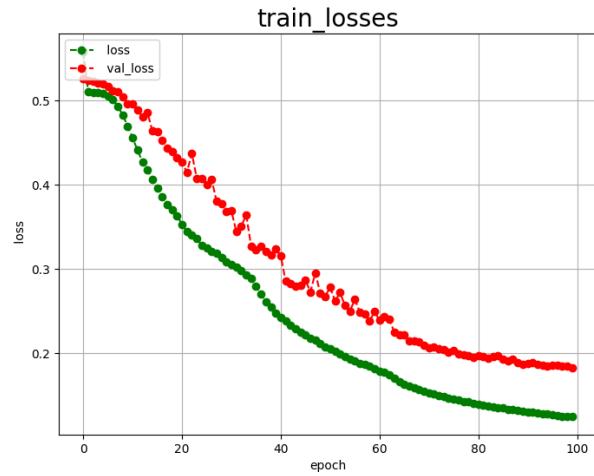
PowerShell 7 (x64)      +  -  x
Epoch 90/100
[2800/12800 [=====]] - ETA: 0s - loss: 0.1313 - dice_coef: 0.7585
Epoch 90/100 val_loss improved from 0.18905 to 0.18706, saving model to ./models/best_model.h5
[2800/12800 [=====]] - 2219s 173ms/sample - loss: 0.1313 - dice_coef: 0.7585 - val_loss: 0.1871 - val_dice_coef: 0.6615 - lr: 2.8000e-05
Epoch 91/100
[2800/12800 [=====]] - ETA: 0s - loss: 0.1305 - dice_coef: 0.7599
Epoch 91: val_loss did not improve from 0.18706
[2800/12800 [=====]] - loss: 0.1305 - dice_coef: 0.7599 - val_loss: 0.1877 - val_dice_coef: 0.6609 - lr: 2.8000e-05
Epoch 92/100
[2800/12800 [=====]] - ETA: 0s - loss: 0.1298 - dice_coef: 0.7612
Epoch 92: val_loss did not improve from 0.18706
[2800/12800 [=====]] - 2238s 175ms/sample - loss: 0.1298 - dice_coef: 0.7612 - val_loss: 0.1890 - val_dice_coef: 0.6584 - lr: 2.8000e-05
Epoch 93/100
[2800/12800 [=====]] - ETA: 0s - loss: 0.1288 - dice_coef: 0.7630
Epoch 93: val_loss did not improve from 0.18706
[2800/12800 [=====]] - 2207s 172ms/sample - loss: 0.1288 - dice_coef: 0.7630 - val_loss: 0.1873 - val_dice_coef: 0.6626 - lr: 2.8000e-05
Epoch 94/100
[2800/12800 [=====]] - ETA: 0s - loss: 0.1282 - dice_coef: 0.7641
Epoch 94: val_loss improved from 0.18706 to 0.18633, saving model to ./models/best_model.h5
[2800/12800 [=====]] - 2232s 174ms/sample - loss: 0.1282 - dice_coef: 0.7641 - val_loss: 0.1863 - val_dice_coef: 0.6646 - lr: 2.8000e-05
Epoch 95/100
[2800/12800 [=====]] - ETA: 0s - loss: 0.1277 - dice_coef: 0.7651
Epoch 95: val_loss improved from 0.18633 to 0.18493, saving model to ./models/best_model.h5
[2800/12800 [=====]] - 2190s 171ms/sample - loss: 0.1277 - dice_coef: 0.7651 - val_loss: 0.1848 - val_dice_coef: 0.6669 - lr: 2.8000e-05
Epoch 96/100
[2800/12800 [=====]] - ETA: 0s - loss: 0.1272 - dice_coef: 0.7659
Epoch 96: val_loss did not improve from 0.18483
[2800/12800 [=====]] - 2229s 174ms/sample - loss: 0.1272 - dice_coef: 0.7659 - val_loss: 0.1861 - val_dice_coef: 0.6637 - lr: 2.8000e-05
Epoch 97/100
[2800/12800 [=====]] - ETA: 0s - loss: 0.1264 - dice_coef: 0.7675
Epoch 97: val_loss did not improve from 0.18483
[2800/12800 [=====]] - 2173s 170ms/sample - loss: 0.1264 - dice_coef: 0.7675 - val_loss: 0.1861 - val_dice_coef: 0.6645 - lr: 2.8000e-05
Epoch 98/100
[2800/12800 [=====]] - ETA: 0s - loss: 0.1255 - dice_coef: 0.7689
Epoch 98: val_loss did not improve from 0.18483
[2800/12800 [=====]] - 2204s 172ms/sample - loss: 0.1255 - dice_coef: 0.7689 - val_loss: 0.1849 - val_dice_coef: 0.6668 - lr: 2.8000e-05
Epoch 99/100
[2800/12800 [=====]] - ETA: 0s - loss: 0.1250 - dice_coef: 0.7698
Epoch 99: val_loss did not improve from 0.18483
[2800/12800 [=====]] - 2300s 180ms/sample - loss: 0.1250 - dice_coef: 0.7698 - val_loss: 0.1853 - val_dice_coef: 0.6656 - lr: 2.8000e-05
Epoch 100/100
[2800/12800 [=====]] - ETA: 0s - loss: 0.1251 - dice_coef: 0.7697
Epoch 100: val_loss improved from 0.18483 to 0.18283, saving model to ./models/best_model.h5
[2800/12800 [=====]] - 2325s 182ms/sample - loss: 0.1251 - dice_coef: 0.7697 - val_loss: 0.1828 - val_dice_coef: 0.6710 - lr: 1.1200e-05
== Save history.json

```

[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:

```
PowerShell 7 (x64) + - x
==== DatasetClass <class 'ImageMaskDataset.ImageMaskDataset'>
==== BaseImageMaskDataset.constructor
==== ConfigParser ./train_eval_infer config
==== WARNING: Not found [mask] algorithm, return default value None
==== WARNING: Not found [image] color_converter, return default value None
==== WARNING: Not found [dataset] image_format, return default value rgb
==== WARNING: Not found [dataset] image_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
==== contrast_adjuster False
==== WARNING: Not found [image] contrast alph, return default value 1.5
==== WARNING: Not found [image] contrast best, return default value 40
==== WARNING: Not found [dataset] mask format, return default value gray
==== WARNING: Not found [mask] binarize, return default value False
==== 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
==== image_normalize False
==== image_colorspace None
==== ImageMaskDataset.constructor
==== self.resize_interpolation 2
==== WARNING: Not Found [model] evaluation, return default value test
==== BaseImageMaskDataset.create_dataset test
==== Configuration ./dataset/Stenosis/test/images/..././dataset/Stenosis/test/masks/
==== WARNING: Not found [mask] mask channels, return default value 1
==== num_classes 1 image data type <class 'numpy.uint8'>
num_images 800 512 512
100% | 800/800 [00:07<00:00, 104.98it/s]
X: shape (800, 512, 512, 3) type uint8
Y: shape (800, 512, 512, 1) type bool
Create X-Y ten 800 Y-ten 800
==== WARNING: Not Found [eval] batch_size, return default value 4
==== evaluate batch_size 4
E:\PyXY3D\efficientdet\lib\site-packages\keras\engine\training_v1.py:232: UserWarning: 'Model.state_updates' will be removed in a future version. This property should not be used in Tensorflow 2.0, as 'updates' are applied automatically.
Test loss :0.1992
Test accuracy:0.6378
==== Evaluation metric:loss score:0.1992
==== Evaluation metric:dice coeff score:0.6378
==== Saved ./evaluation.csv
```

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
```

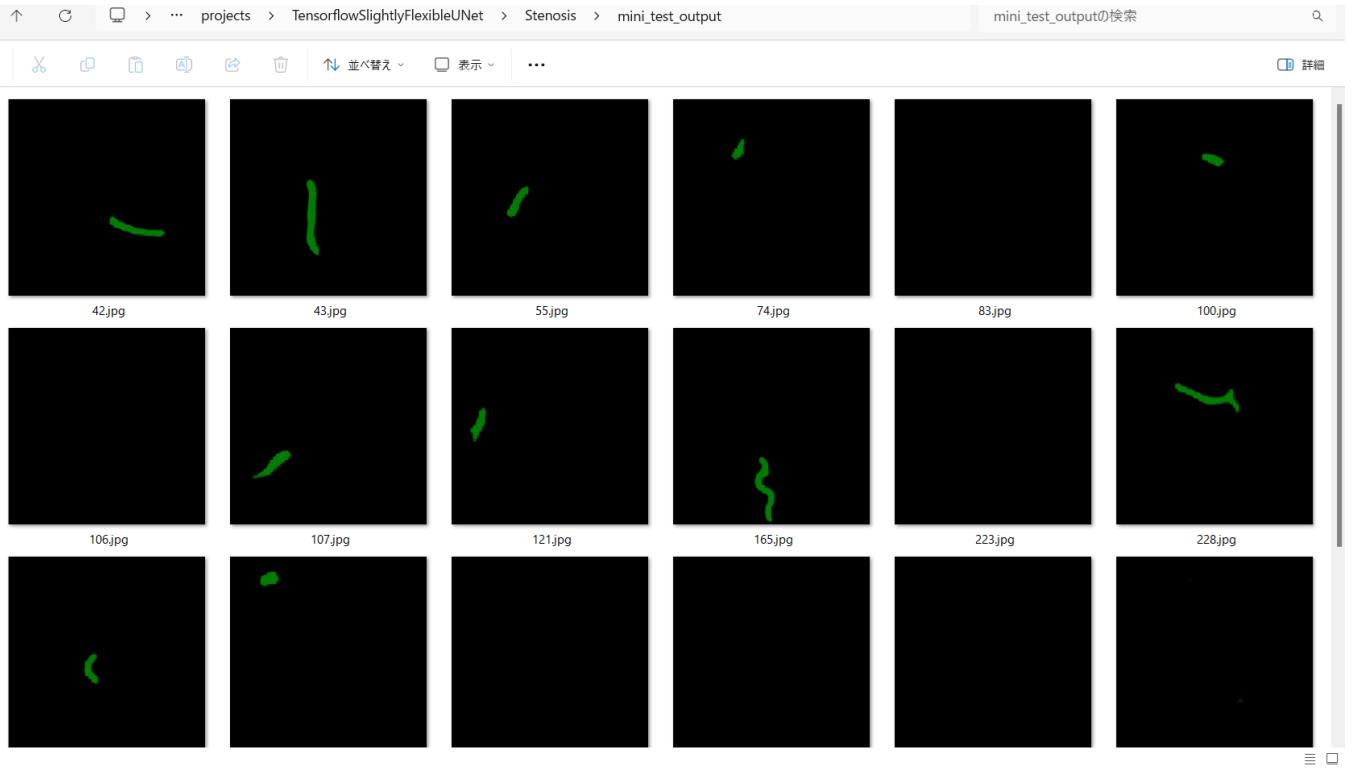
mini_test_images



mini_test_image(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

Pylint	Configure
Lint a Python application with pylint.	
Python application	Configure
Create and test a Python application.	
Django	Configure
Build and Test a Django Project	

[More workflows](#)

[Dismiss suggestions](#)