

  [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

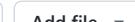
 Apache-2.0 license

 0 stars  0 forks  2 watching  Branches  Activity

 Tags

 Public repository

---

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

 <a href="#">sarah-antillia</a>	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.

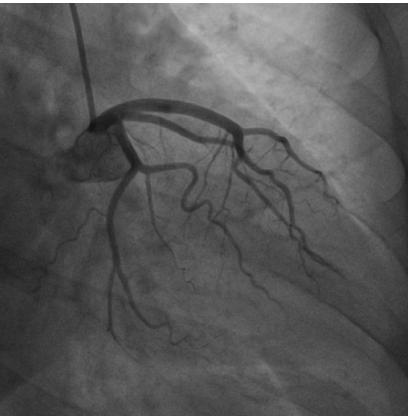
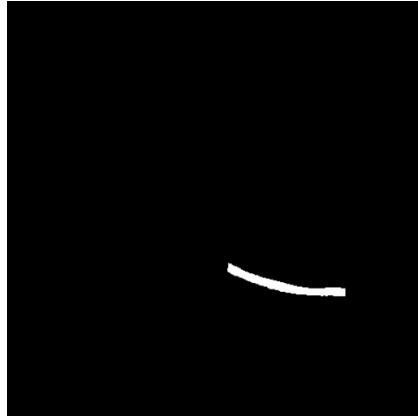
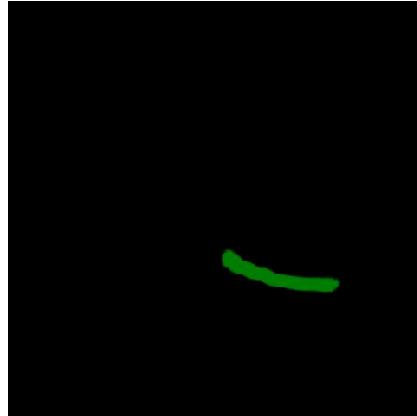
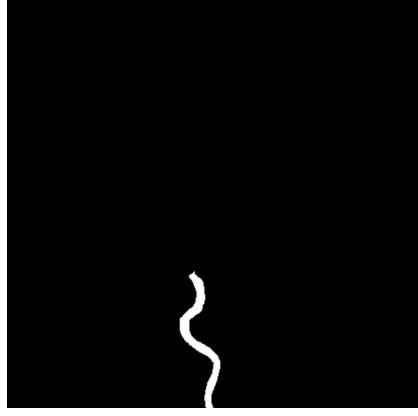
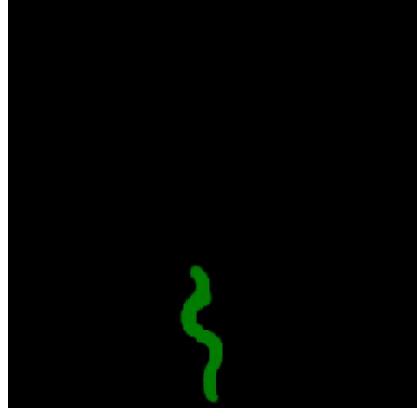
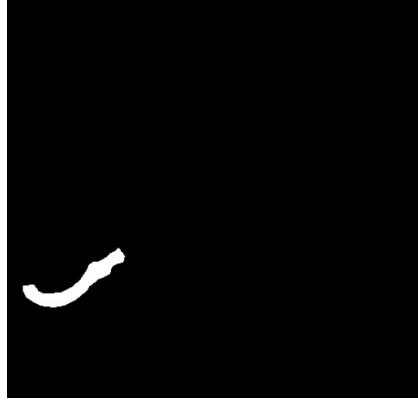
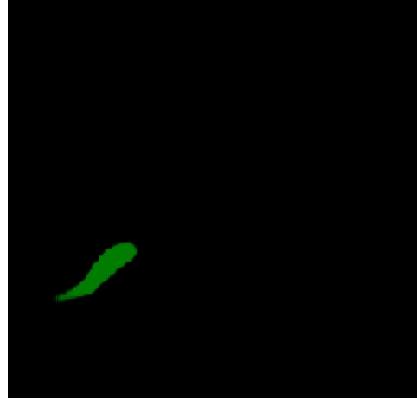
- Vertical flip
- Horizontal flip
- Rotation
- Shrinks
- Shears
- Deformation
- Distortion
- Barrel distortion
- Pincushion distortion

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 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 Kulabayeva,

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
```

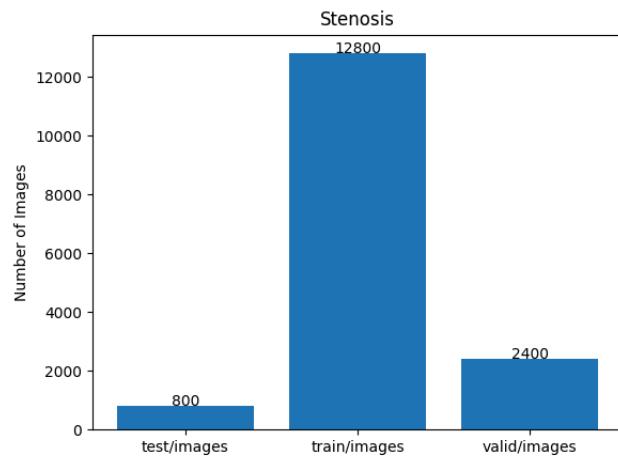
The foloder structure of the original **dataset\_phase\_1** of ARCADE is the following.

```
./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**

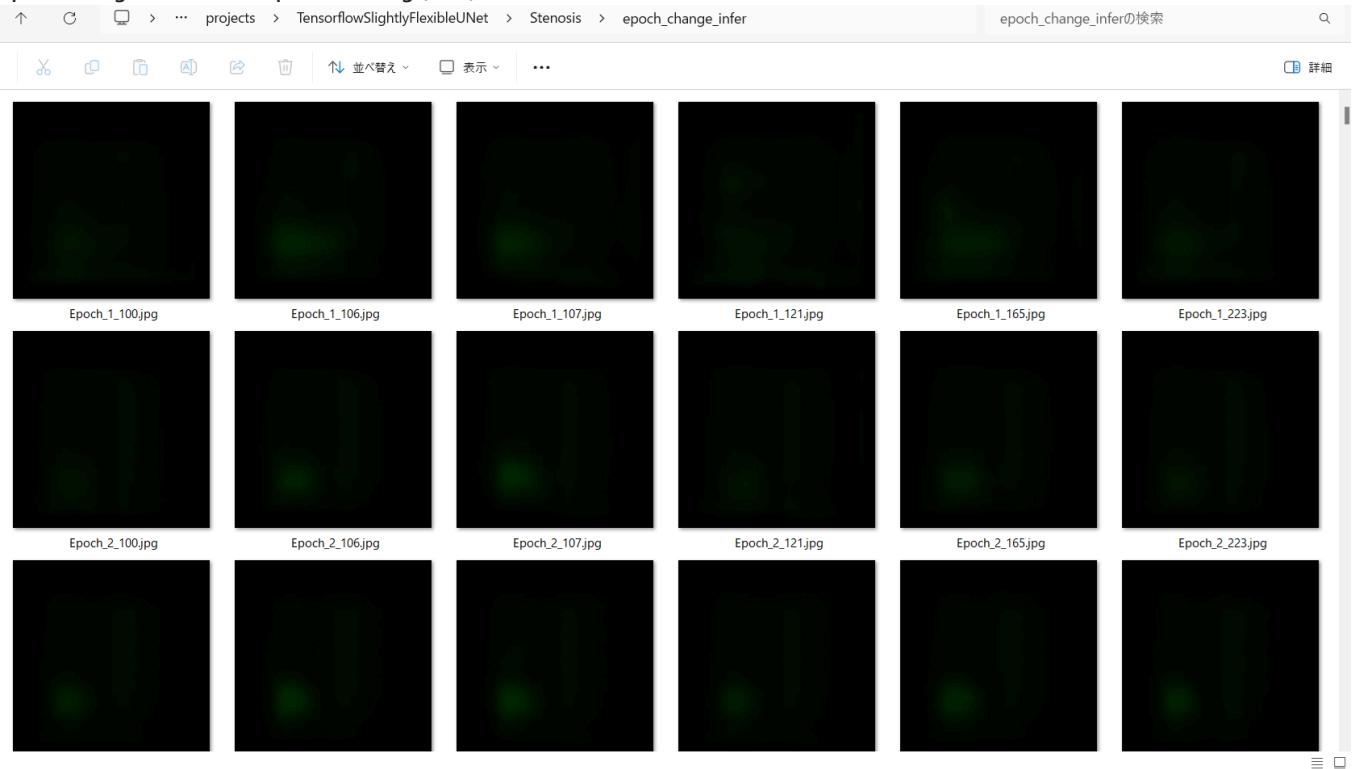
Enabled EpochChange infer callback.

**Enabled color space converter**

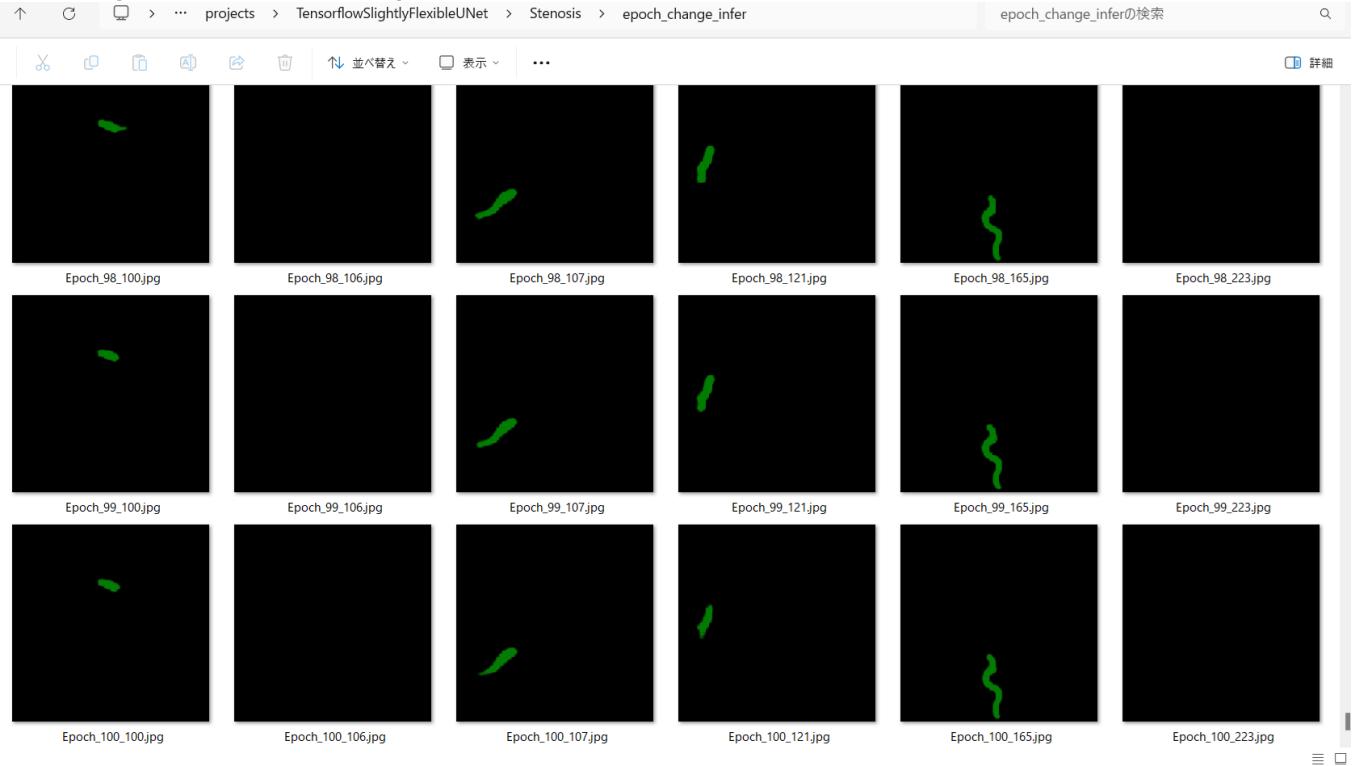
```
[train]
epoch_change_infer     = True
epoch_change_infer_dir = "./epoch_change_infer"
num_infer_images       = 6
```

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)



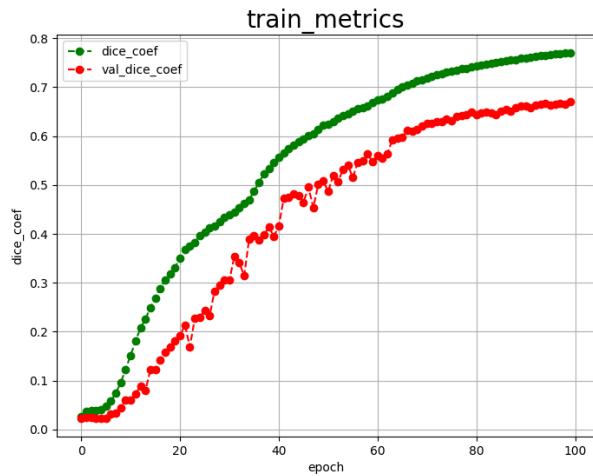
#### Epoch\_change\_inference output at ending (98,99,100)



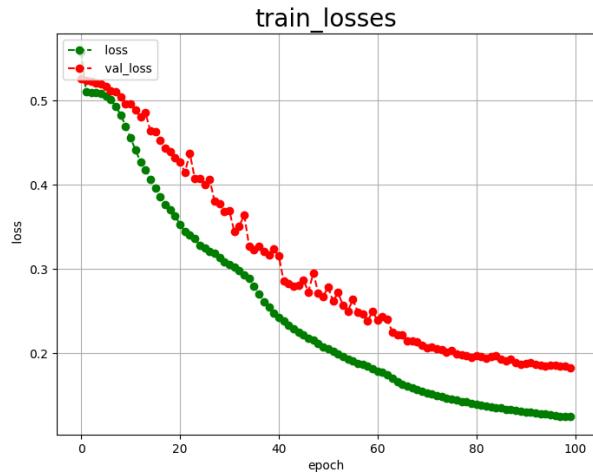
In this case, the training process was terminated at epoch 100..

Training console output

## 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 [dataset] color_space, return default value None
== WARNING: Not found [dataset] image_format, return default value rgb
== WARNING: Not found [dataset] input_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
== WARNING: Not found [dataset] mask_color
== WARNING: Not found [image] contrast_alpha, 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 [mask] invert, 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 1
image normalize False
image algorithm None
ImageMaskDataset.constructor
self.resize_interpolation 2
== WARNING: Not found [model] evaluation, return default value test
== BaseImageMaskDataset.create_dataset test
== WARNING: Not found [dataset] Stenosis/test/images/ ./../dataset/Stenosis/test/masks/
== WARNING: Not found [dataset] mask_channels, return default value 1
num classes 1 image data_type <class 'numpy.uint8'>
num images 800 512
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-tensor: 800 Y-tensor: 800
== WARNING: Not found [eval] batch_size, return default value 4
== evaluate batch_size 4
E:\PyXY3D-EfficientNet\lib\site-packages\keras\engine\training_v1.py:2332: 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.
  warnings.warn('Model.state_updates' + warning)
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

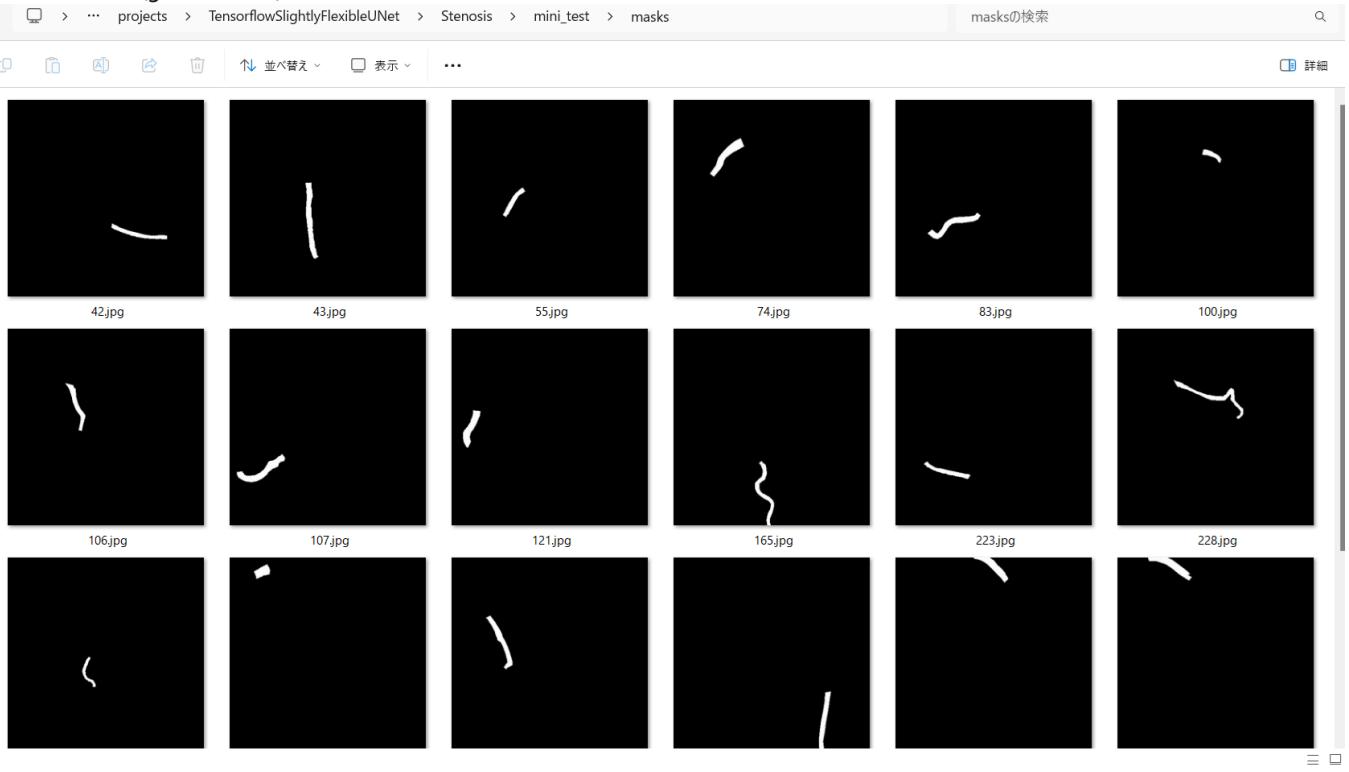
This simply runs the following command.

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

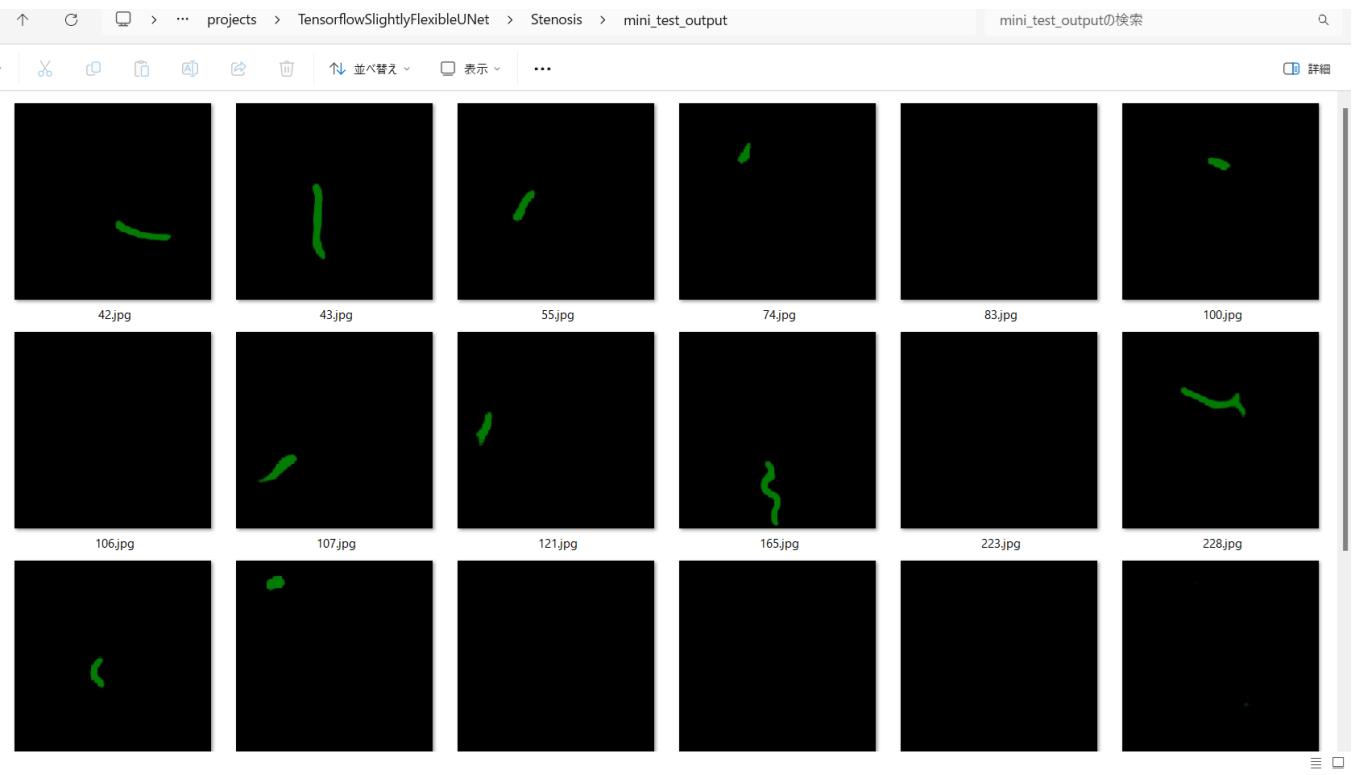
## mini\_test\_images



mini\_test\_image(ground\_truth)

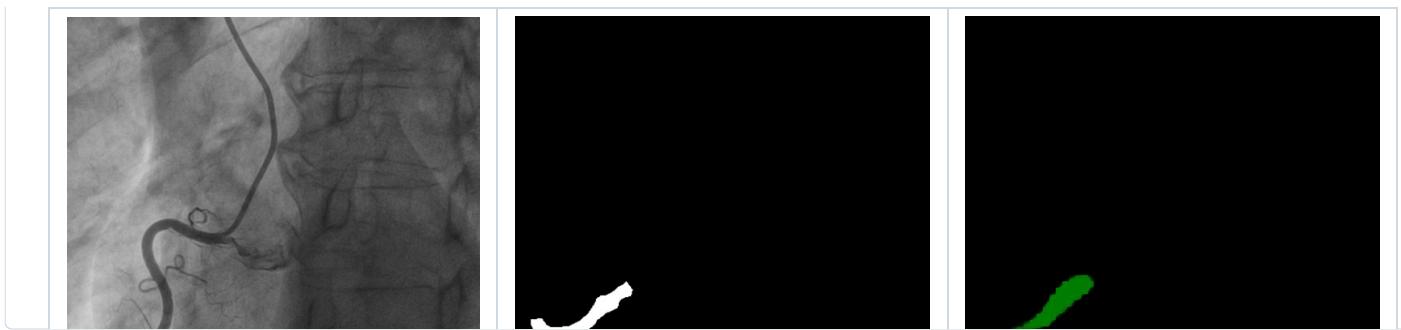


Inferred test masks



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

Django	<a href="#">Configure</a>
Build and Test a Django Project	
Python application	<a href="#">Configure</a>
Create and test a Python application.	
Pylint	<a href="#">Configure</a>
Lint a Python application with pylint.	

[More workflows](#)

[Dismiss suggestions](#)