

Tensorflow-Tiled-Image-Segmentation-Messidor-2-Retinal-Vessel (2025/02/26)

Sarah T. Arai
Software Laboratory antillia.com

This is the first experiment of Image Segmentation for [Messidor-2](#) based on Pretrained HRF Retinal Vessel Model, which was trained by the latest [Tensorflow-Image-Segmentation-API](#), and a **pre-augmented tiled dataset** [Augmented-Tiled-HRF-ImageMask-Dataset.zip](#), which was derived by us from the following dataset:

[Download the whole dataset \(~73 Mb\)](#) in [High-Resolution Fundus \(HRF\) Image Database](#).

Please see also our experiments:

- [Tensorflow-Tiled-Image-Segmentation-Pre-Augmented-HRF-Retinal-Vessel](#) based on [High-Resolution Fundus \(HRF\) Image Database](#).
- [Tensorflow-Tiled-Image-Segmentation-Pre-Augmented-DRIVE-Retinal-Vessel](#) based on [DRIVE: Digital Retinal Images for Vessel Extraction](#)
- [Tensorflow-Tiled-Image-Segmentation-Pre-Augmented-STARE-Retinal-Vessel](#) based on [SStructured Analysis of the Retina](#).
- [Tensorflow-Image-Segmentation-Retinal-Vessel](#) based on [CHASE_DB1 dataset](#).

Experiment Strategies

1. LABELS (antillia ground truth) for Messidor-2 master IMAGES

The Messidor-2 dataset contains no LABELS (ground truth) data. Therefore, we created our own master LABELS (ground truth) from the original Messidor-2 IMAGES by using Tiled Image Segmentation method and Pretrained-HRF-Retinal-Vessel UNet Model which was trained by the latest [Tensorflow-Image-Segmentation-API](#), and a **pre-augmented tiled dataset** [Augmented-Tiled-HRF-ImageMask-Dataset.zip](#).

2. Messidor-2 Dataset

We splitted the Messidor-2-master IMAGES and LABLELES into test, train and valid subsets.

3. Train Messidor-2 Segmentation Model

We trained and validated a TensorFlow UNet model by using the **Messidor-2 train and valid subsets**

4. Evaluate Messidor-2 Segmentation Model

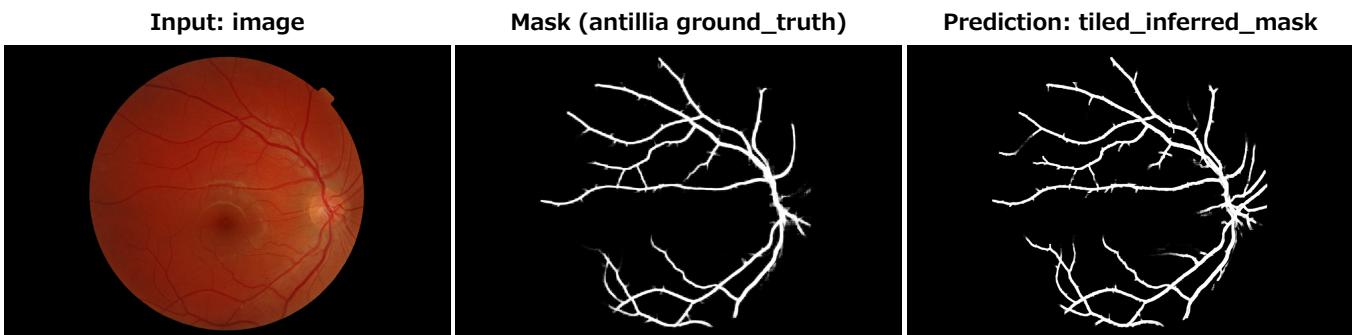
We evaluated the performance of the trained UNet model by using the **Messidor-2 test** dataset by computing the **bce_dice_loss** and **dice_coef**.

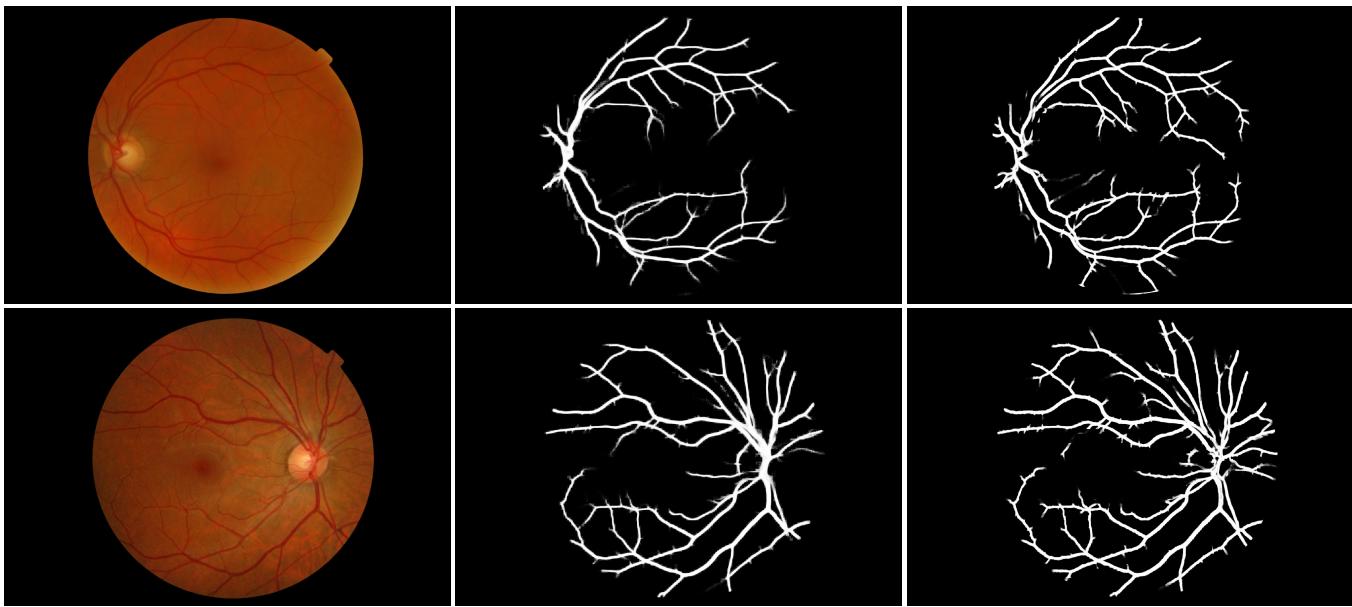
5. Tiled Inference

We applied our Tiled Image Segmentation method to infer the Retinal Vessel for the mini_test images of the original Messidor-2 IMAGES of 2240x1488 pixels.

Actual Tiled Image Segmentation for Messidor-2 IMAGES of 2240x1488 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 HRF 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

1.1 High-Resolution Fundus (HRF) Image Database

The dataset used here has been taken from the dataset [Download the whole dataset \(~73 Mb\)](#) in [High-Resolution Fundus \(HRF\) Image Database](#).

Introduction

This database has been established by a collaborative research group to support comparative studies on automatic segmentation algorithms on retinal fundus images. The database will be iteratively extended and the webpage will be improved. We would like to help researchers in the evaluation of segmentation algorithms. We encourage anyone working with segmentation algorithms who found our database useful to send us their evaluation results with a reference to a paper where it is described. This way we can extend our database of algorithms with the given results to keep it always up-to-date.

The database can be used freely for research purposes. We release it under Creative Commons 4.0 Attribution License.

Citation

[Robust Vessel Segmentation in Fundus Images](#)

Budai, Attila; Bock, Rüdiger; Maier, Andreas; Hornegger, Joachim; Michelson, Georg.

[International Journal of Biomedical Imaging, vol. 2013, 2013](#)

Licence

[Creative Commons 4.0 Attribution License.](#)

1.2 Messidor-2 Dataset

We also used [Messidor-2 Dataset](#),

MESSIDOR stands for **M**ethods to **E**valuate **S**egmentation and **I**ndexing **T**echniques in the field of **R**etinal **O**phthalmology (in French).

Messidor-2 Dataset

The Messidor-2 dataset is a collection of Diabetic Retinopathy (DR) examinations, each consisting of two macula-centered eye

fundus images (one per eye). Part of the dataset (Messidor-Original) was kindly provided by the Messidor program partners. The remainder (Messidor-Extension) consists of never-before-published examinations from Brest University Hospital. In the original Messidor dataset, some fundus images came in pairs, some others were single. Messidor-Original consists of all image pairs from the original Messidor dataset, that is 529 examinations (1058 images, saved in PNG format). In order to populate Messidor-Extension, diabetic patients were recruited in the Ophthalmology department of Brest University Hospital (France) between October 16, 2009 and September 6, 2010. Eye fundi were imaged, without pharmacological dilation, using a Topcon TRC NW6 non-mydiatic fundus camera with a 45 degree field of view. Only macula-centered images were included in the dataset. Messidor-Extension contains 345 examinations (690 images, in JPG format).

Overall, Messidor-2 contains 874 examinations (1748 images). The dataset comes with a spreadsheet containing image pairing.

It does not contain annotations such as a diabetic retinopathy “ground truth”. However, some third-parties proposed such annotations, but these are independent from the official Messidor-2 database, and therefore not handled by our services.

Using the database

Messidor-2 can be used, free of charge, for research and educational purposes. Copy, redistribution, and any unauthorized commercial use are prohibited. Any publication relying on this dataset must acknowledge the [LaTIM laboratory](#) and the Messidor program partners. Please include the following acknowledgment:

Kindly provided by the Messidor program partners (see <https://www.adcis.net/en/third-party/messidor/>).

Decencière et al..

Feedback on a publicly distributed database: the Messidor database.

Image Analysis & Stereology, v. 33, n. 3, p. 231-234, aug. 2014. ISSN 1854-5165.

Available at: <http://www.ias-iss.org/ojs/IAS/article/view/1155> or <http://dx.doi.org/10.5566/ias.1155>. M. D. Abràmoff, J. C. Folk,

D. P. Han, J. D. Walker, D. F. Williams, S. R. Russell, P. Massin, B. Cochener,

P. Gain, L. Tang, M. Lamard, D. C. Moga, G. Quellec, and M. Niemeijer,

Automated analysis of retinal images for detection of referable diabetic retinopathy.

JAMA Ophthalmol, vol. 131, no. 3, Mar. 2013, p. 351–357.

Available at: <https://doi.org/10.1001/jamaophthalmol.2013.1743>.

2. Create LABELS for Messidor-2 IMAGES

2.1 Download Messidor-2 IMAGES Dataset

Please download [Messidor-2 Dataset](#), and place IMAGES under Messidor-2-master folder as shown below.

```
./projects
└── generator
    └── Messidor-2-master
        └── IMAGES
```

2.2 Download Augmented-Tiled-HRF-Pretrained-Model

Please download our [Augmented-Tiled-HRF-Pretrained-Model.zip](#), expand it and place **best_model.h5** under models folder as shown below.

```
./projects
└── TensorflowSlightlyFlexibleUNet
    └── Augmented-Tiled-HRF
        └── models
            └── best_model.h5
```

2.3 Run Tiled Inference method

Please move the folder "../projects/TensorflowSlightlyFlexibleUNet/Augmented-Tiled-HRF". and run the following bat file.

5_tiled_infer-messidor.bat

This will generate our own LABELS (ground truth) for the official Messidor-2 IMAGES by using the HRF-Pretrained-Model, without any human experts.

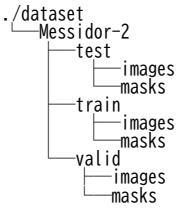
```
./projects
└── generator
    └── Messidor-2-master
        ├── IMAGES
        └── LABELS
```

2.4 Split Messidor-2-master

Please move to "../projects/generator" folder and run the following Python script.

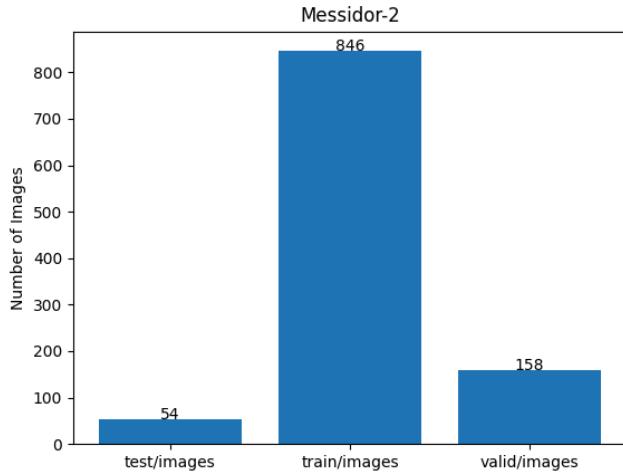
python split_master.py

,by which the following Messidor-2 dataset will be created.



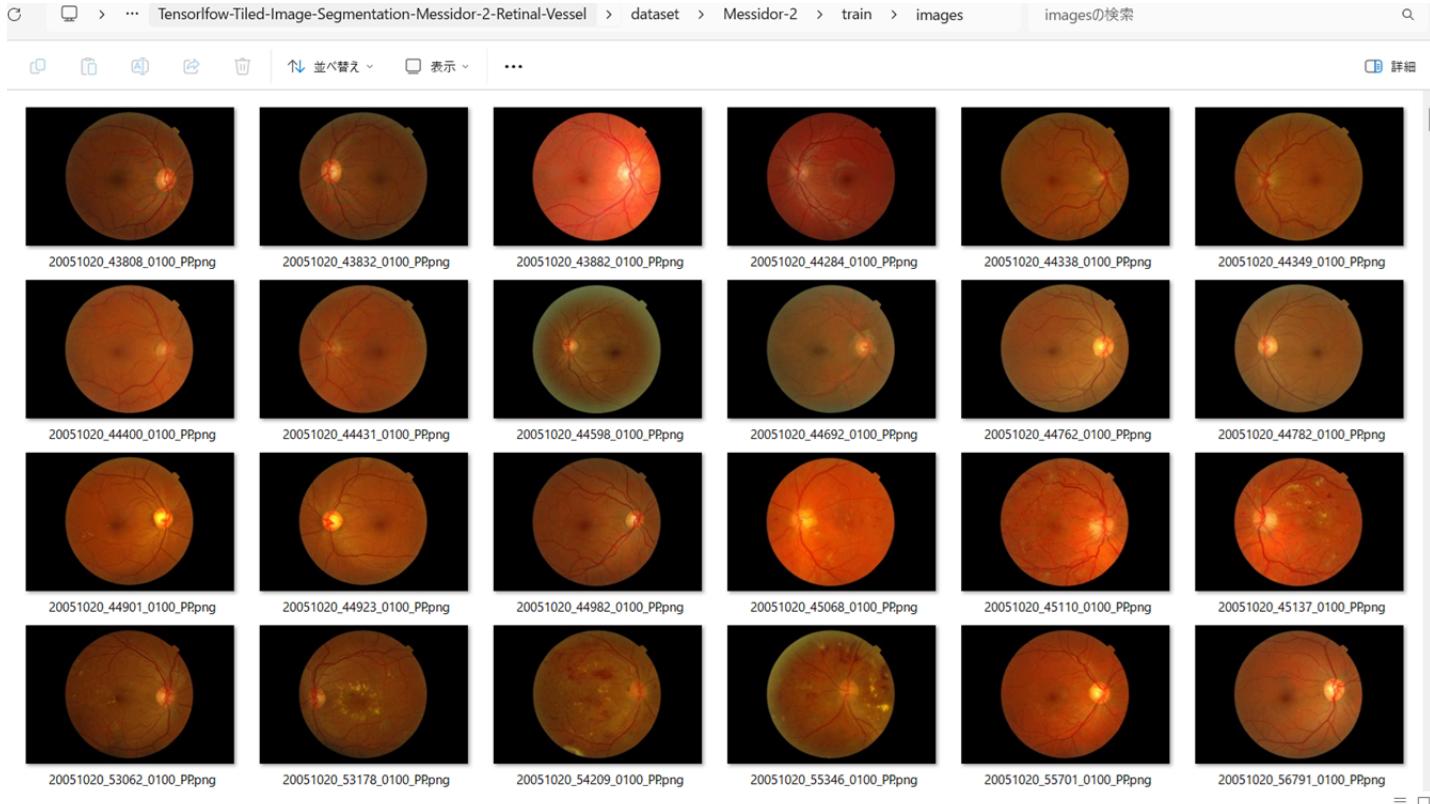
This is a 2240x1488 pixels images and their corresponding masks dataset.

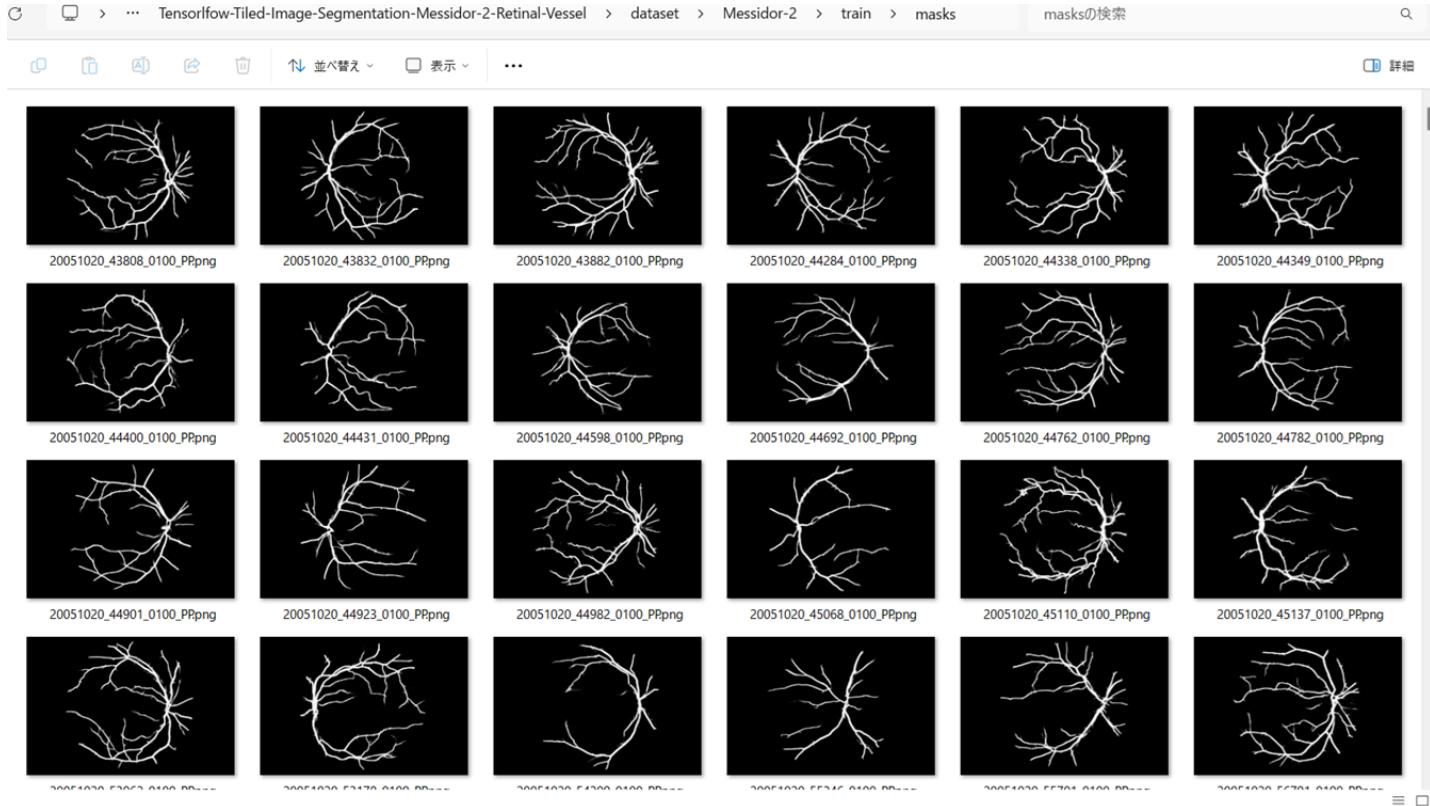
Messidor-2 Statistics



As shown above, the number of images of train and valid datasets is enough to use for a training set of our segmentation model.

Train_images_sample



Train_masks_sample**3 Train TensorflowUNet Model**

We have trained HRF TensorflowUNet Model by using the following [train_eval_infer.config](#) file.
Please move to ./projects/TensorflowSlightlyFlexibleUNet/Messidor-2 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

Enabled Batch Normalization.

Defined a small **base_filters=16** and large **base_kernels=(11,11)** 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 = (11,11)
num_layers = 8
dilation = (1,1)
```

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 learning_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_LINEAR"
```

Early stopping callback

Enabled early stopping callback with patience parameter.

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

Tiled Inference

Used the original Messidor-2 IMAGES as a mini_test dataset for our inference images.

```
[tiledinfer]
images_dir = "./mini_test/images"
output_dir = "./mini_test_output_tiled"
```

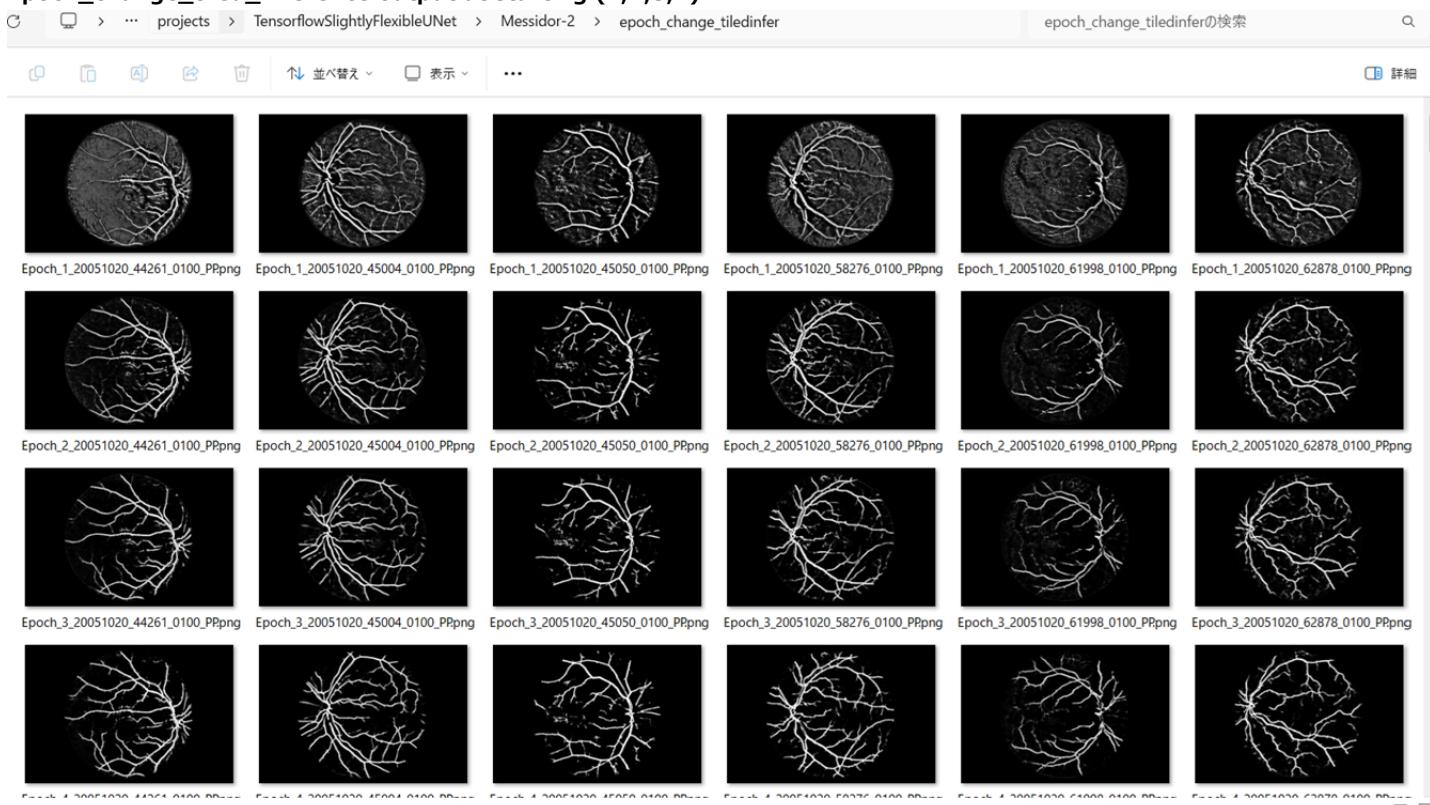
Epoch change inference callbacks

Enabled epoch_change_infer callback.

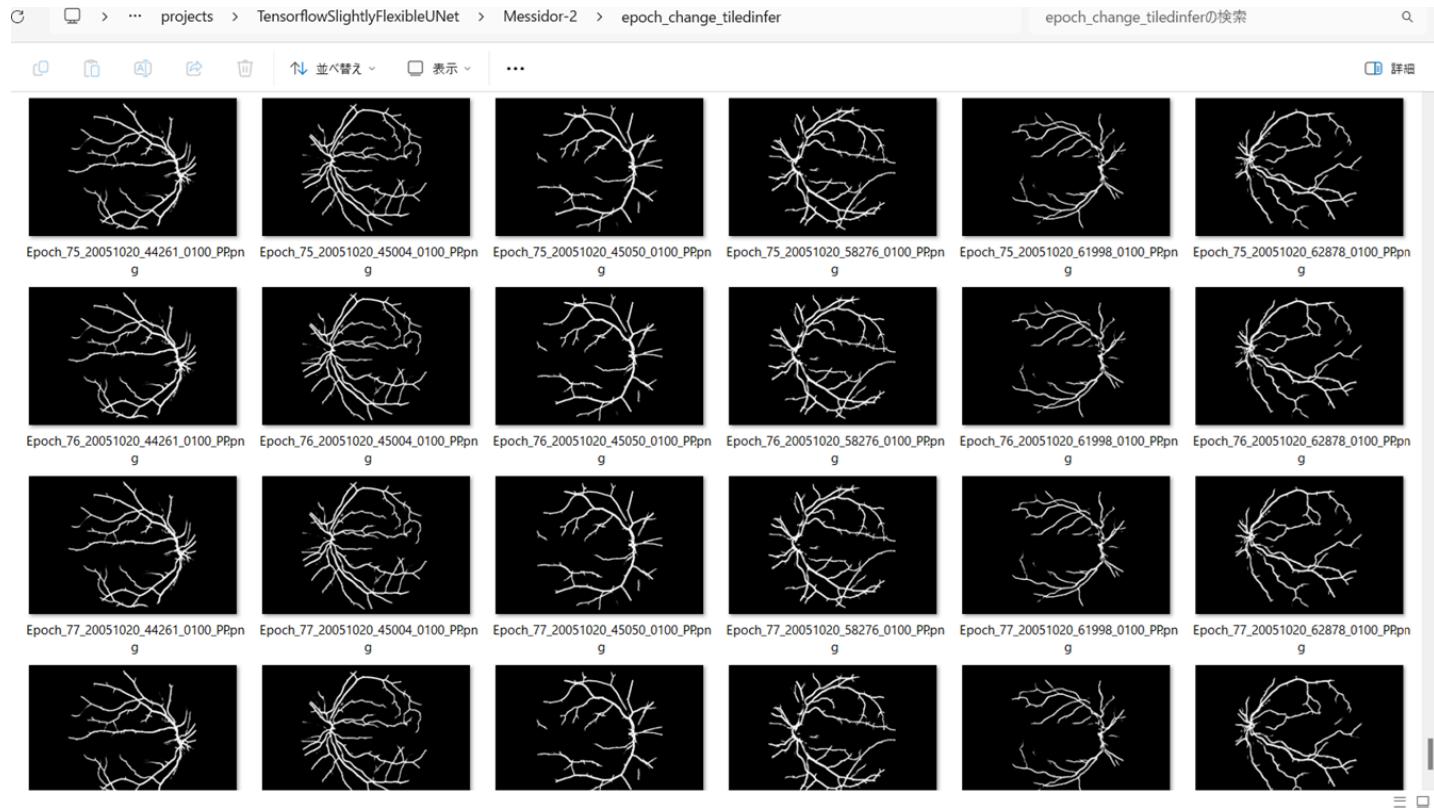
```
[train]
epoch_change_infer = False
epoch_change_infer_dir = "./epoch_change_infer"
epoch_change_tiledinfer = True
epoch_change_tiledinfer_dir = "./epoch_change_tiledinfer"
num_infer_images = 6
```

By using this callback, on every epoch_change, the epoch change tiled 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_tiled_inference output at starting (1,2,3,4)



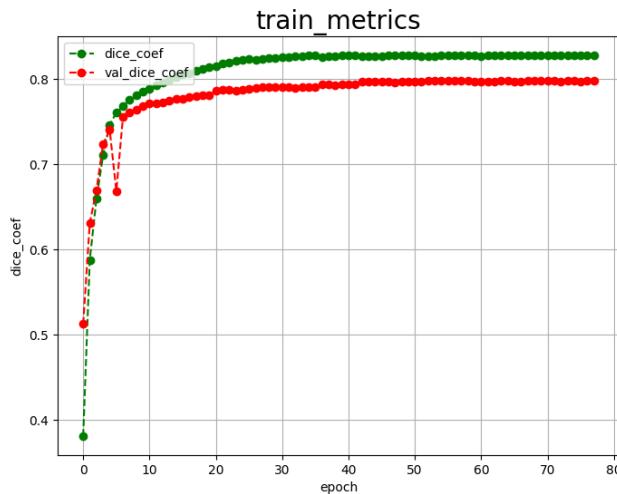
Epoch_change_tiled_inference output at ending (75,76,77,78)



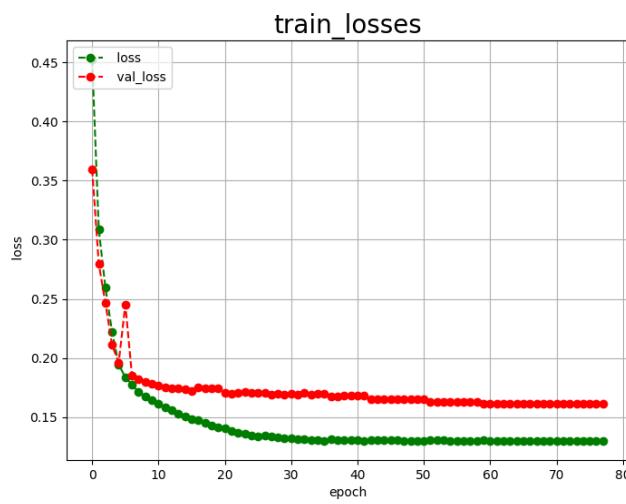
In this experiment, the training process was stopped at epoch 78 by EarlyStopping Callback.

```
PowerShell 7 (x64) - PowerShell 7 (x64)
Epoch 68/100
846/846 [=====] - ETA: 0s - loss: 0.1298 - dice_coef: 0.8279
Epoch 68: val_loss improved from 0.16122 to 0.16118, saving model to ./models/best_model.h5
846/846 [=====] - 246s 290ms/sample - loss: 0.1298 - dice_coef: 0.8279 - val_loss: 0.1612 - val_dice_coef: 0.7981 - lr: 1.6384e-07
Epoch 69/100
846/846 [=====] - ETA: 0s - loss: 0.1298 - dice_coef: 0.8279
Epoch 69: val_loss did not improve from 0.16118
846/846 [=====] - 246s 290ms/sample - loss: 0.1298 - dice_coef: 0.8279 - val_loss: 0.1612 - val_dice_coef: 0.7981 - lr: 1.6384e-07
Epoch 70/100
846/846 [=====] - ETA: 0s - loss: 0.1297 - dice_coef: 0.8279
Epoch 70: val_loss did not improve from 0.16118
846/846 [=====] - 245s 289ms/sample - loss: 0.1297 - dice_coef: 0.8279 - val_loss: 0.1612 - val_dice_coef: 0.7980 - lr: 6.5536e-08
Epoch 71/100
846/846 [=====] - ETA: 0s - loss: 0.1297 - dice_coef: 0.8279
Epoch 71: val_loss did not improve from 0.16118
846/846 [=====] - 244s 288ms/sample - loss: 0.1297 - dice_coef: 0.8279 - val_loss: 0.1613 - val_dice_coef: 0.7979 - lr: 6.5536e-08
Epoch 72/100
846/846 [=====] - ETA: 0s - loss: 0.1297 - dice_coef: 0.8279
Epoch 72: val_loss did not improve from 0.16118
846/846 [=====] - 244s 288ms/sample - loss: 0.1297 - dice_coef: 0.8279 - val_loss: 0.1612 - val_dice_coef: 0.7981 - lr: 6.5536e-08
Epoch 73/100
846/846 [=====] - ETA: 0s - loss: 0.1297 - dice_coef: 0.8280
Epoch 73: val_loss did not improve from 0.16118
846/846 [=====] - 244s 288ms/sample - loss: 0.1297 - dice_coef: 0.8280 - val_loss: 0.1614 - val_dice_coef: 0.7977 - lr: 6.5536e-08
Epoch 74/100
846/846 [=====] - ETA: 0s - loss: 0.1297 - dice_coef: 0.8279
Epoch 74: val_loss did not improve from 0.16118
846/846 [=====] - 244s 288ms/sample - loss: 0.1297 - dice_coef: 0.8279 - val_loss: 0.1613 - val_dice_coef: 0.7980 - lr: 2.6214e-08
Epoch 75/100
846/846 [=====] - ETA: 0s - loss: 0.1297 - dice_coef: 0.8280
Epoch 75: val_loss did not improve from 0.16118
846/846 [=====] - 243s 287ms/sample - loss: 0.1297 - dice_coef: 0.8280 - val_loss: 0.1613 - val_dice_coef: 0.7980 - lr: 2.6214e-08
Epoch 76/100
846/846 [=====] - ETA: 0s - loss: 0.1297 - dice_coef: 0.8280
Epoch 76: val_loss did not improve from 0.16118
846/846 [=====] - 243s 287ms/sample - loss: 0.1297 - dice_coef: 0.8280 - val_loss: 0.1614 - val_dice_coef: 0.7978 - lr: 2.6214e-08
Epoch 77/100
846/846 [=====] - ETA: 0s - loss: 0.1297 - dice_coef: 0.8280
Epoch 77: val_loss did not improve from 0.16118
846/846 [=====] - 243s 287ms/sample - loss: 0.1297 - dice_coef: 0.8280 - val_loss: 0.1613 - val_dice_coef: 0.7981 - lr: 2.6214e-08
Epoch 78/100
846/846 [=====] - ETA: 0s - loss: 0.1297 - dice_coef: 0.8280
Epoch 78: val_loss did not improve from 0.16118
846/846 [=====] - 243s 287ms/sample - loss: 0.1297 - dice_coef: 0.8280 - val_loss: 0.1613 - val_dice_coef: 0.7981 - lr: 1.0486e-08
Epoch 78: early stopping
Save history.json
```

[train_metrics.csv](#)



[train_losses.csv](#)



4 Evaluation

Please move to a **./projects/TensorflowSlightlyFlexibleUNet/Messidor-2** folder, and run the following bat file to evaluate TensorflowUNet model for Messidor-2/test.

`./2.evaluate.bat`

This bat file simply runs the following command.

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

Evaluation console output:

```
PowerShell 7 (x64) + 
--- WARNING: Not found [train] best model file, return default value best.model.h5
--- Loaded a weight file ./models/best model.h5
--- DatasetClass <class 'ImageMaskDataset.ImageMaskDataset'>
--- BaseImageMaskDataset constructor
--- ConfigParser <train eval infer>.config
--- WARNING: Not found [dataset] algorithm, 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 False
--- WARNING: Not found [dataset] mask, return default value False
--- WARNING: Not found [dataset] color order, return default value bgr
--- contrast adjust for False
--- WARNING: Not found [image] contrast alphas, 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 1
--- image normalize False
--- image color space None
--- ImageMaskDataset constructor
--- self._resize_interpolation_2
--- WARNING: Not found [model] evaluation, return default value test
--- BaseImageMaskDataset.create_dataset test
--- create ./dataset/Messidor-2/test/images/ ../../dataset/Messidor-2/test/masks/
--- WARNING: Not found [mask] mask channels, return default value 1
--- num classes 1 image data_type <class 'numpy.uint8'>
num images 54 512 512
100% | 54/54 [00:03<00:00, 15.57it/s]
X: shape (54, 512, 512, 3) type uint8
Y: shape (54, 512, 512, 1) type bool
--- Create X-tensor with len 54
--- WARNING: Not found [model] batch_size, return default value 4
--- evaluate batch size 4
E:\py310-efficientdet\lib\site-packages\keras\engines\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.
  updates = self.state_updates
Test loss: 0.1583
Test accuracy: 0.8027
Evaluation metric:loss score:0.1583
Evaluation metric:dice_coef score:0.8027
--- Saved ./evaluation.csv
```

[evaluation.csv](#)

The loss (bce_dice_loss) to this Messidor-2/test was not low, and dice_coef not high as shown below.

```
loss,0.1583
dice_coef,0.8027
```

5 Tiled Inference

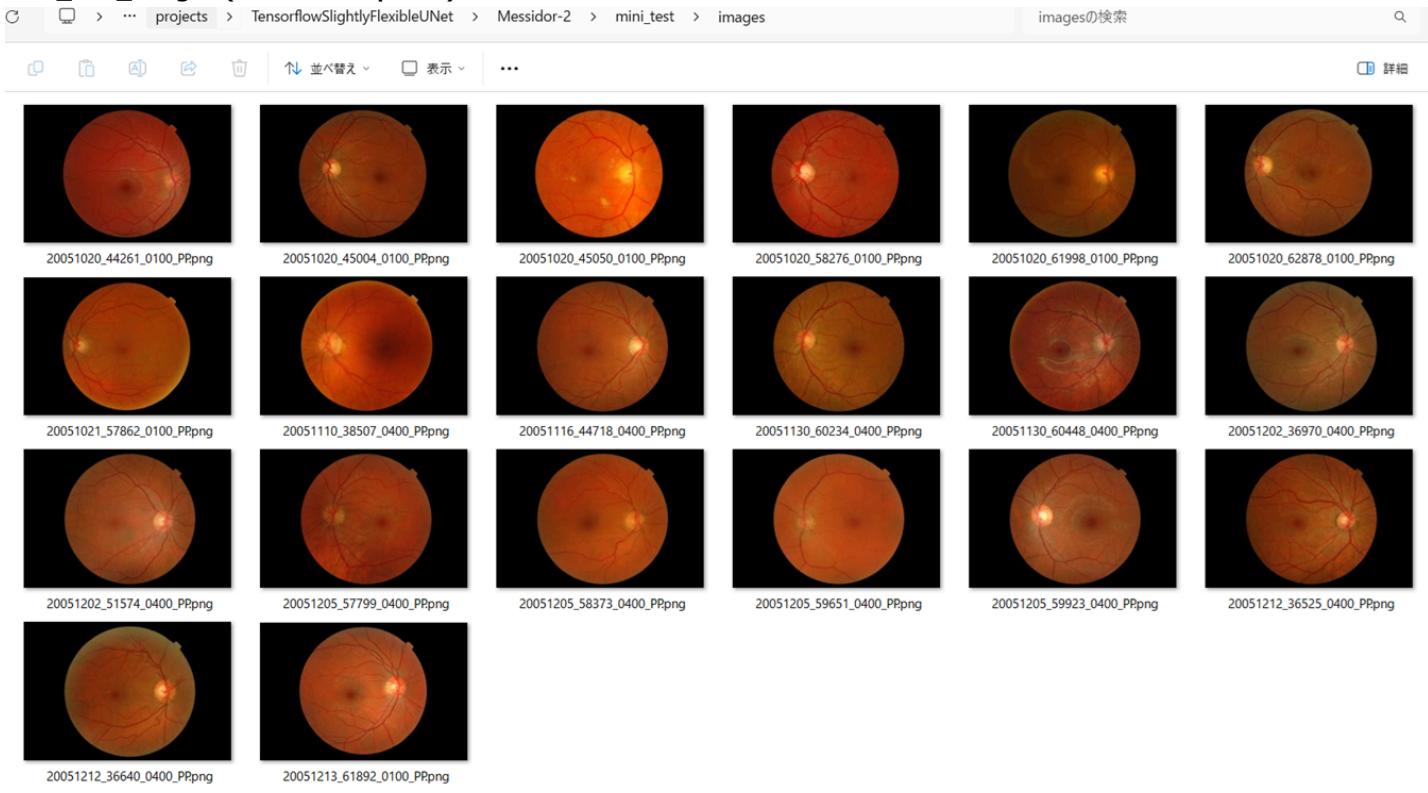
Please move to a **./projects/TensorflowSlightlyFlexibleUNet/Messidor-2** folder
,and run the following bat file to infer segmentation regions for images by the Trained-TensorflowUNet model for Messidor-2.

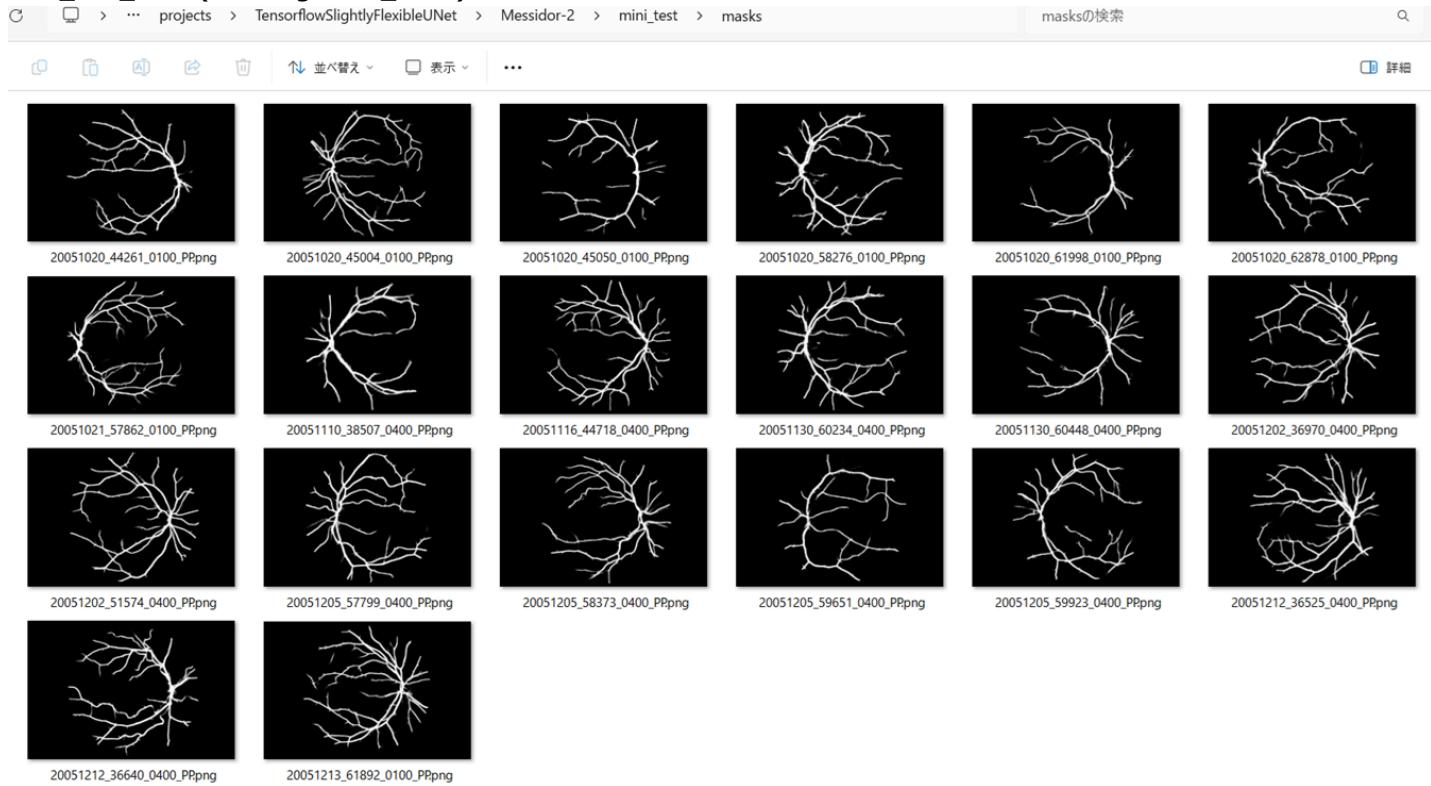
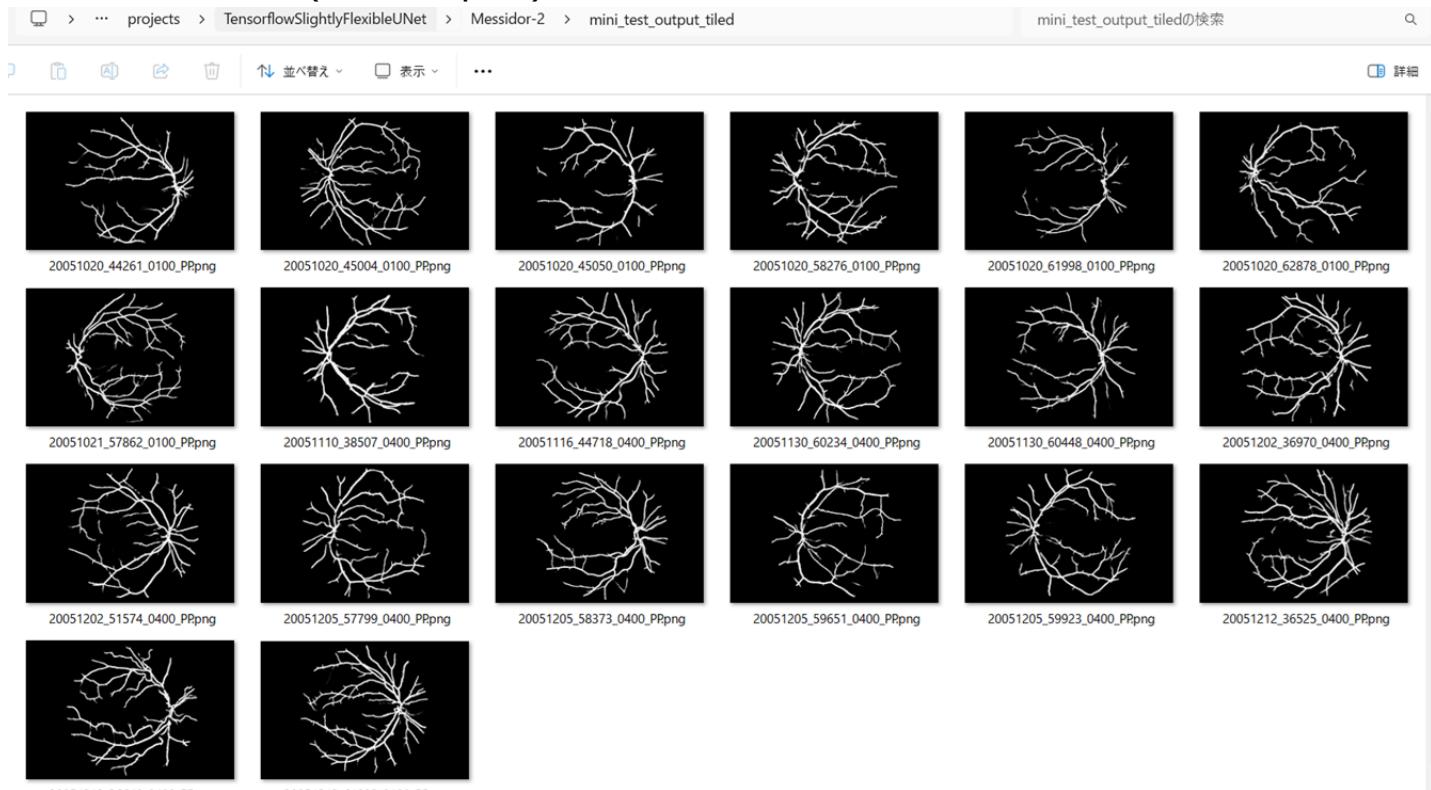
```
./4.tiledinfer.bat
```

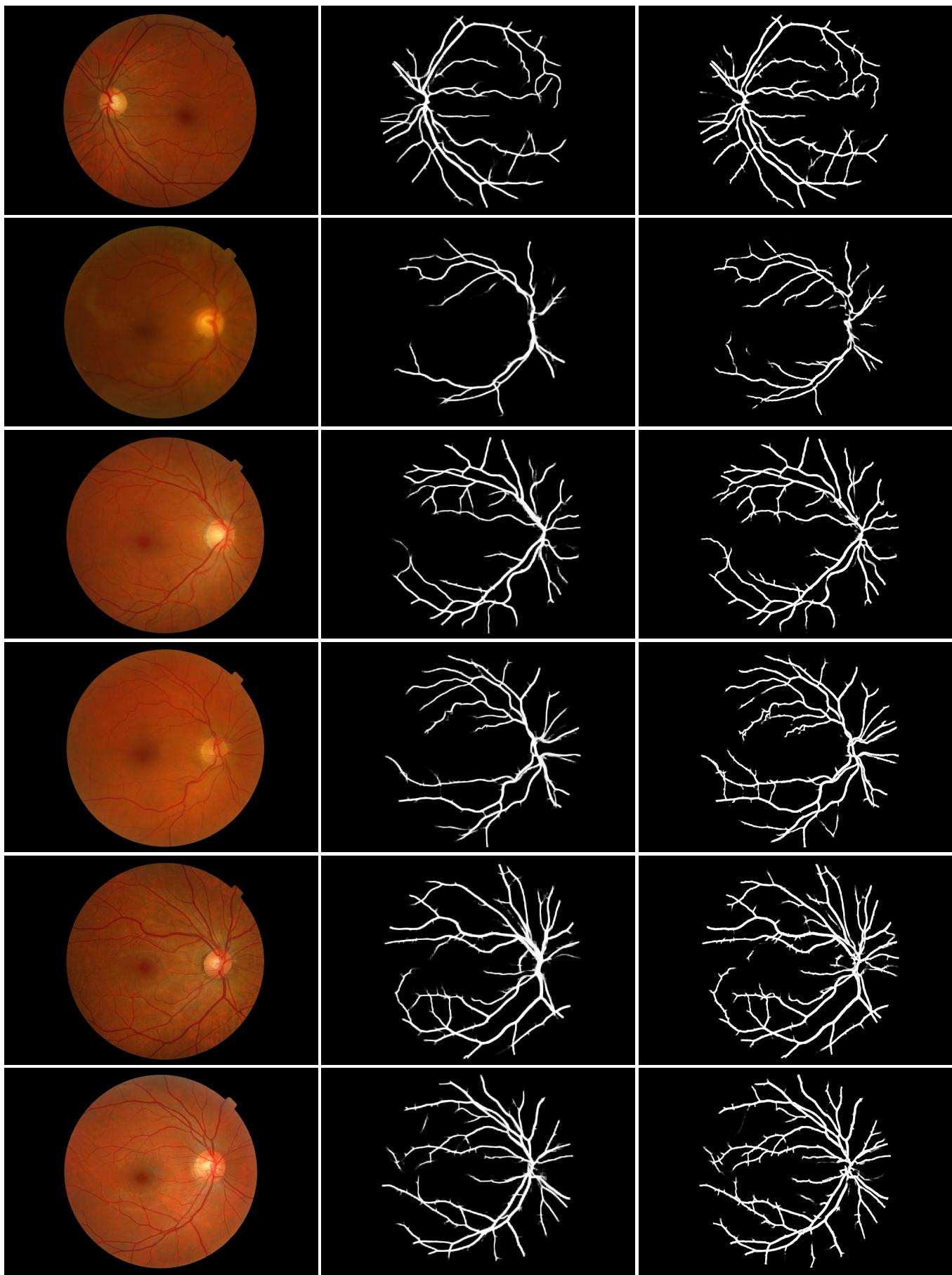
This simply runs the following command.

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

mini_test_images (2240x1488 pixels)



mini_test_mask(antillia ground_truth)**Tiled inferred test masks (2240x1488 pixels)****Enlarged images and masks of 2240x1488 pixels****Image****Mask (antillia ground_truth)****Tiled-inferred-mask**



References

1. Locating Blood Vessels in Retinal Images by Piecewise Threshold Probing of a Matched Filter Response

Adam Hoover, Valentina Kouznetsova, and Michael Goldbaum

<https://www.uhu.es/retinopathy/General/000301IEEETransMedImag.pdf>

2. High-Resolution Fundus (HRF) Image Database

Budai, Attila; Bock, Rüdiger; Maier, Andreas; Hornegger, Joachim; Michelson, Georg.

<https://www5.cs.fau.de/research/data/fundus-images/>.

3. Robust Vessel Segmentation in Fundus Images

Budai, Attila; Bock, Rüdiger; Maier, Andreas; Hornegger, Joachim; Michelson, Georg.

<https://onlinelibrary.wiley.com/doi/10.1155/2013/154860>

4. State-of-the-art retinal vessel segmentation with minimalistic models

Adrian Galdran, André Anjos, José Dolz, Hadi Chakor, Hervé Lombaert & Ismail Ben Ayed

<https://www.nature.com/articles/s41598-022-09675-y>

5. Retinal blood vessel segmentation using a deep learning method based on modified U-NET model

Sanjeeewani, Arun Kumar Yadav, Mohd Akbar, Mohit Kumar, Divakar Yadav

<https://www.semanticscholar.org/reader/f5cb3b1c69a2a7e97d1935be9d706017af8cc1a3>

6. Tensorflow-Tiled-Image-Segmentation-Pre-Augmented-STARE-Retinal-Vessel

Toshiyuki Arai @antillia.com

<https://github.com/sarah-antillia/Tensorflow-Tiled-Image-Segmentation-Pre-Augmented-STARE-Retinal-Vessel>

7, Tensorflow-Tiled-Image-Segmentation-Pre-Augmented-HRF-Retinal-Vessel

Toshiyuki Arai @antillia.com

<https://github.com/sarah-antillia/Tensorflow-Tiled-Image-Segmentation-Pre-Augmented-HRF-Retinal-Vessel>