

Tensorflow-Image-Segmentation-FIVES-Retinal-Vessel (2025/02/19)

Sarah T. Arai
Software Laboratory antillia.com

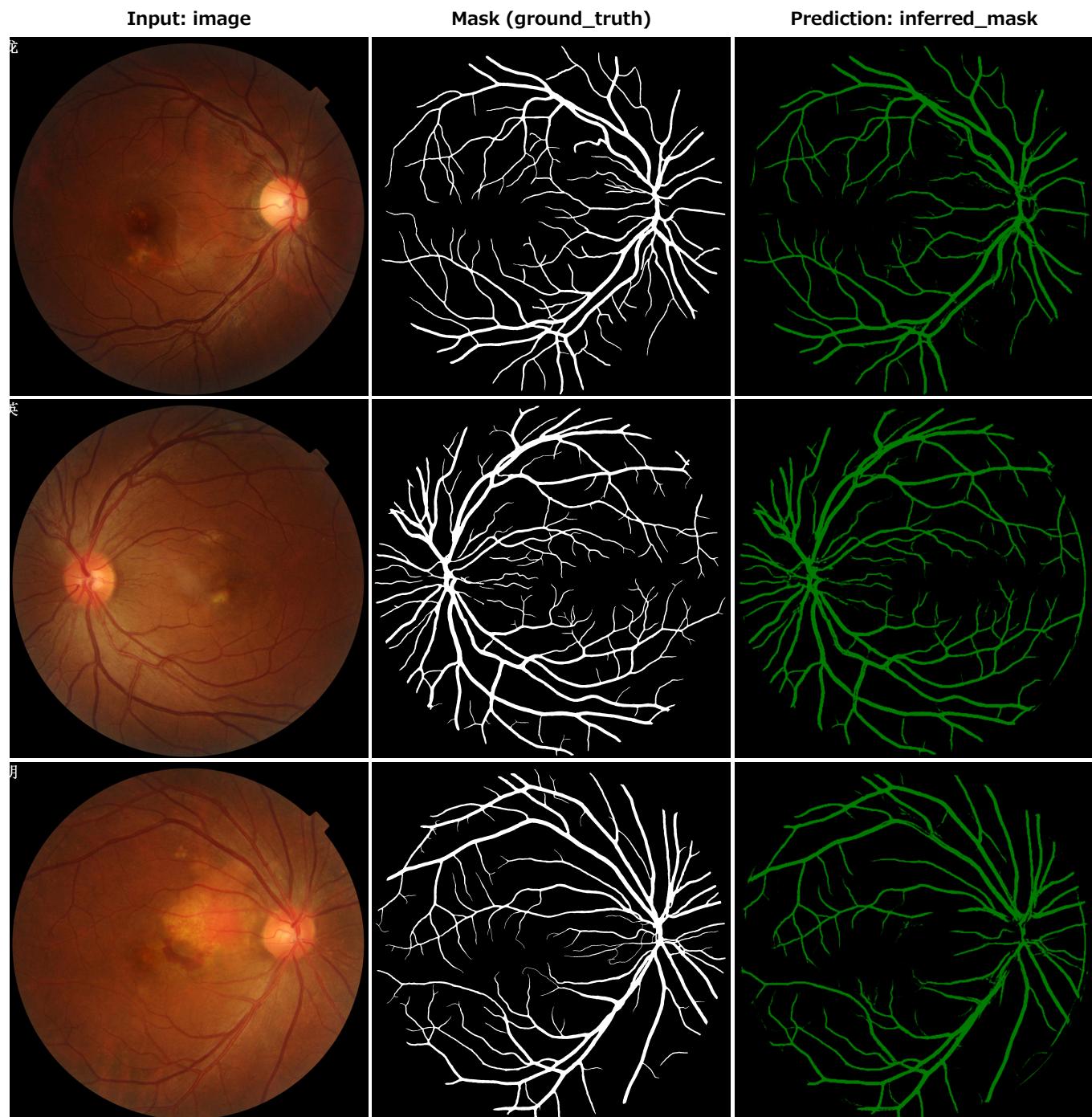
This is the first experiment of Image Segmentation for **FIVES** Retinal Vessel based on the latest [Tensorflow-Image-Segmentation-API](#), and [FIVES: A Fundus Image Dataset for AI-based Vessel Segmentation](#)

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](#) baased on [SStructured Analysis of the Retina](#).
- [Tensorflow-Image-Segmentation-Retinal-Vessel](#) based on [CHASE DB1 dataset](#).

Actual Image Segmentation for Images of 2048x2048 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 FIVES 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 the following figshare web-site:

[FIVES: A Fundus Image Dataset for AI-based Vessel Segmentation](#)

Kai Jin, Xingru Huang, Jingxin Zhou, Yunxiang Li, Yan Yan, Yibao Sun, Qianni Zhang, Yaqi Wang, Juan Ye

FIVES dataset consists of 800 high-resolution multi-disease color fundus photographs with pixel-wise manual annotation. The annotation process was standardized through crowdsourcing of a group of medical experts. The quality of each image was evaluated, including illumination and color distortion, blur, and low contrast distortion, based on which the data splitting was conducted to make sure the balanced distribution of image features.

Detailed descriptions can be found in the original paper, and please cite it if utilizing any part of the dataset:

Jin, K., Huang, X., Zhou, J. et al. FIVES: A Fundus Image Dataset for Artificial Intelligence based Vessel Segmentation. Sci Data 9, 475 (2022). <https://doi.org/10.1038/s41597-022-01564-3>

2 FIVES ImageMask Dataset

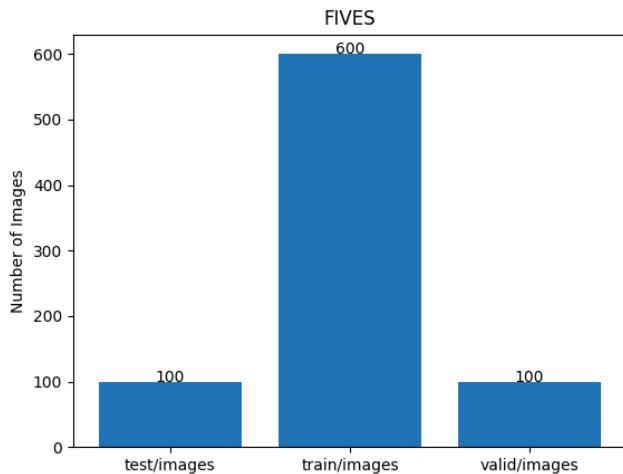
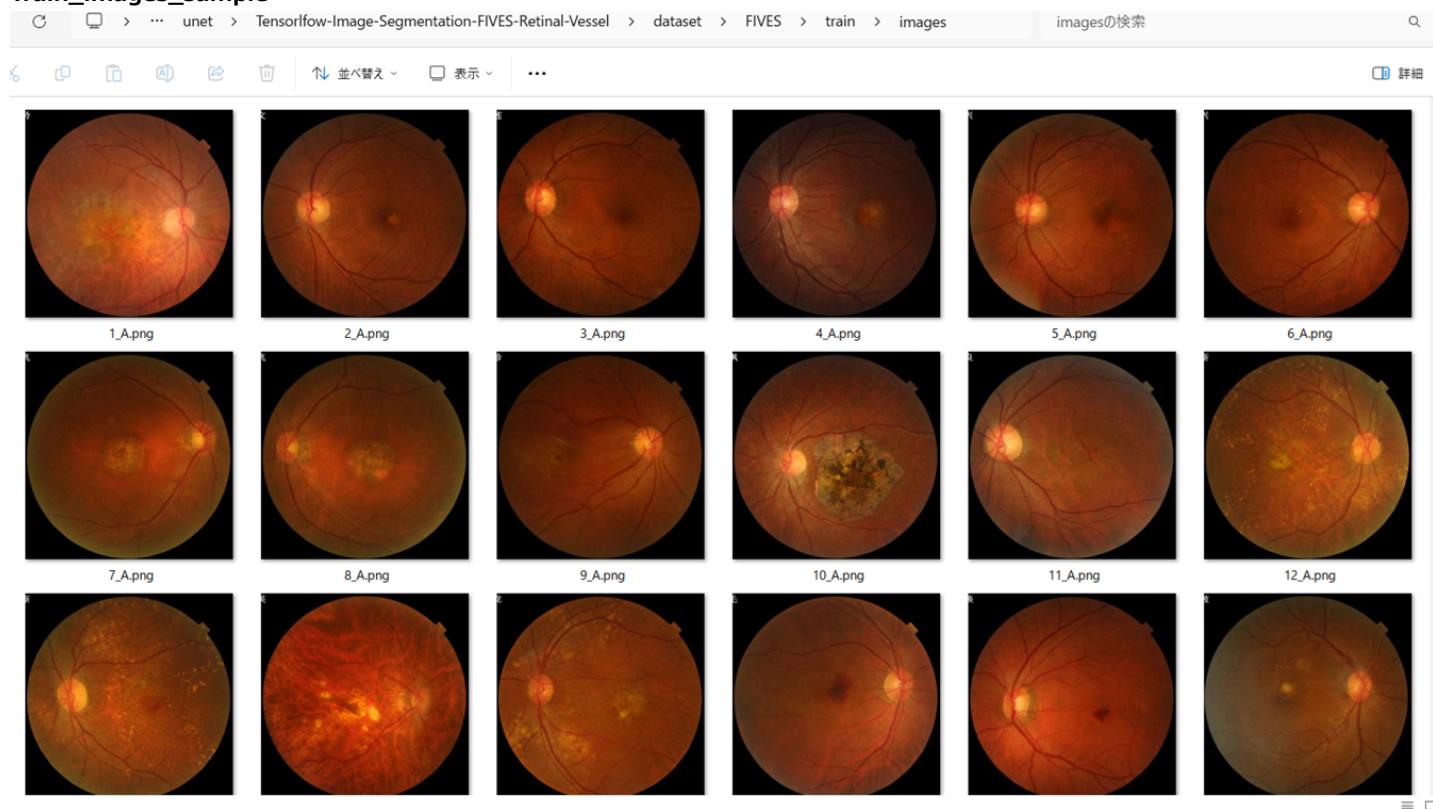
If you would like to train this FIVESSegmentation model by yourself, please download the dataset from [FIVES: A Fundus Image Dataset for AI-based Vessel Segmentation](#)

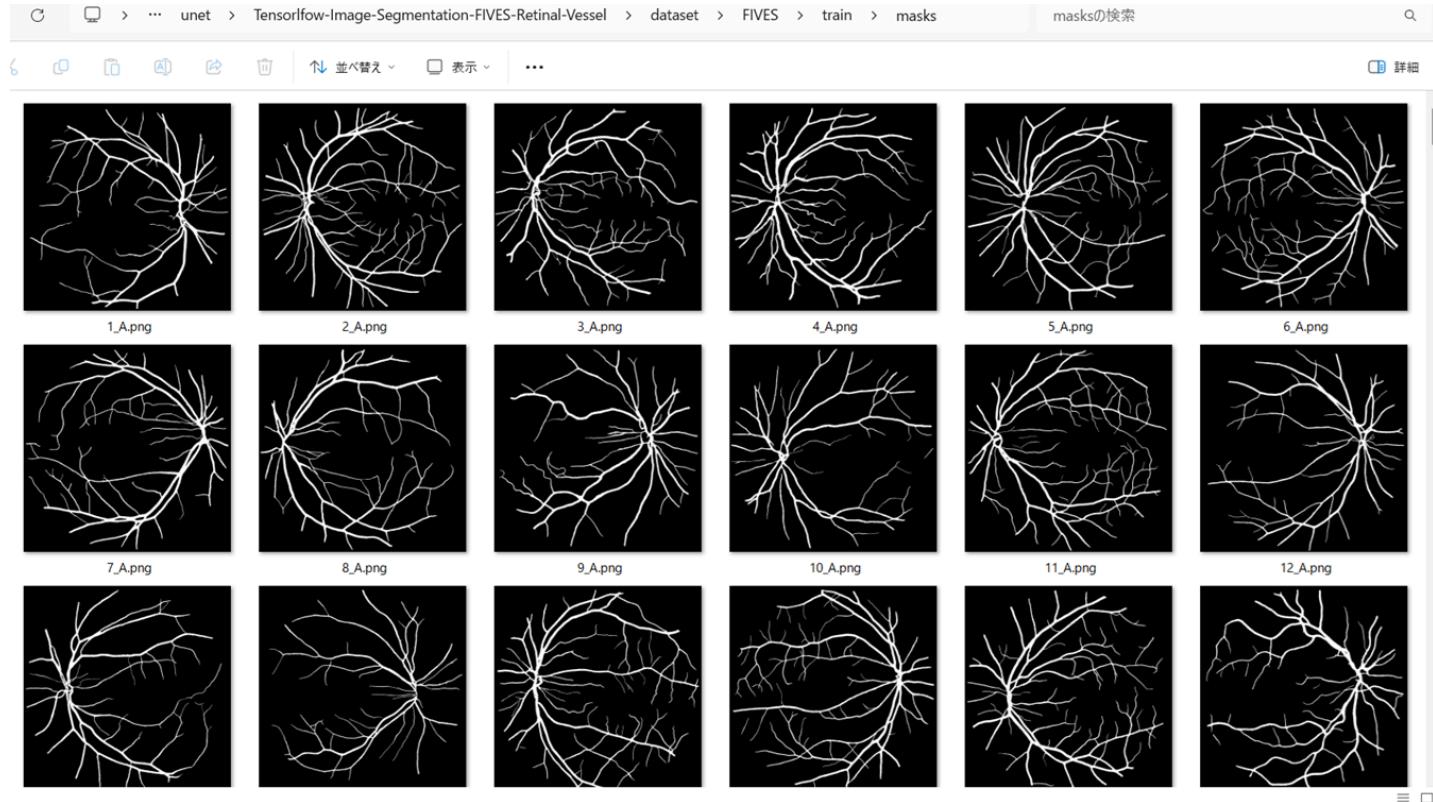
The folder structure of the dataset is the following,

```
./FIVES A Fundus Image Dataset for AI-based Vessel Segmentation
└── test
    ├── Ground truth
    └── Original
└── train
    ├── Ground truth
    └── Original
```

As shown below, we splitted the original dataset into **test**, **train** and **valid** subsets.

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

FIVES Dataset Statistics**Train_images_sample**

Train_masks_sample**3 Train TensorflowUNet Model**

We have trained FIVESTensorflowUNet Model by using the following [train_eval_infer.config](#) file.

Please move to ./projects/TensorflowSlightlyFlexibleUNet/FIVES 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 = (11, 11)
num_layers = 8
```

Learning rate

Defined a small learning rate.

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

Online augmentation

Disabled our online augmentation.

```
[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
```

Early stopping callback

Enabled early stopping callback with patience parameter.

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

Color space conversion

Used COLOR_BGR2Luv color space converter..

```
[image]
color_converter = "cv2.COLOR_BGR2Luv"
```

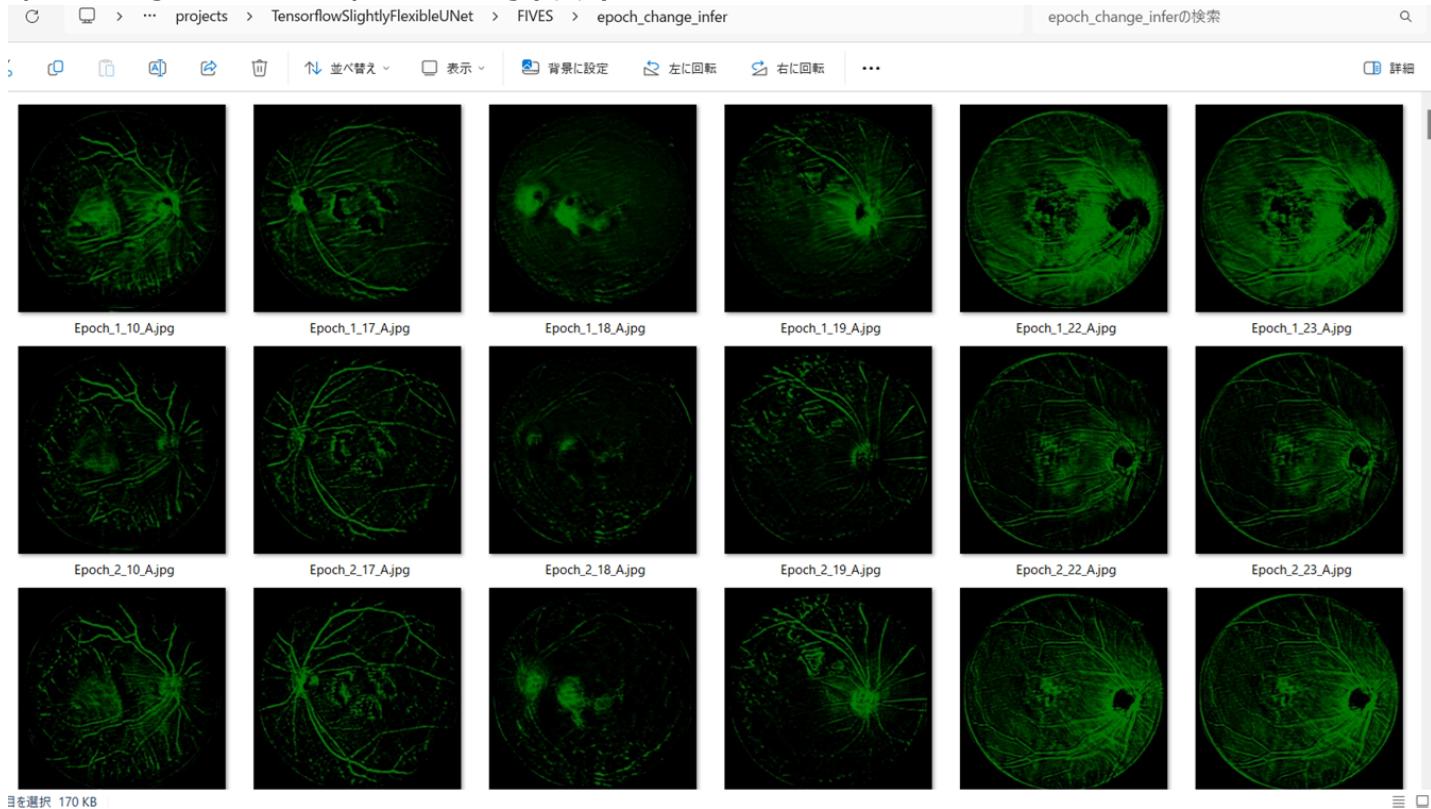
Epoch change inference callbacks

Enabled epoch_change_infer callback.

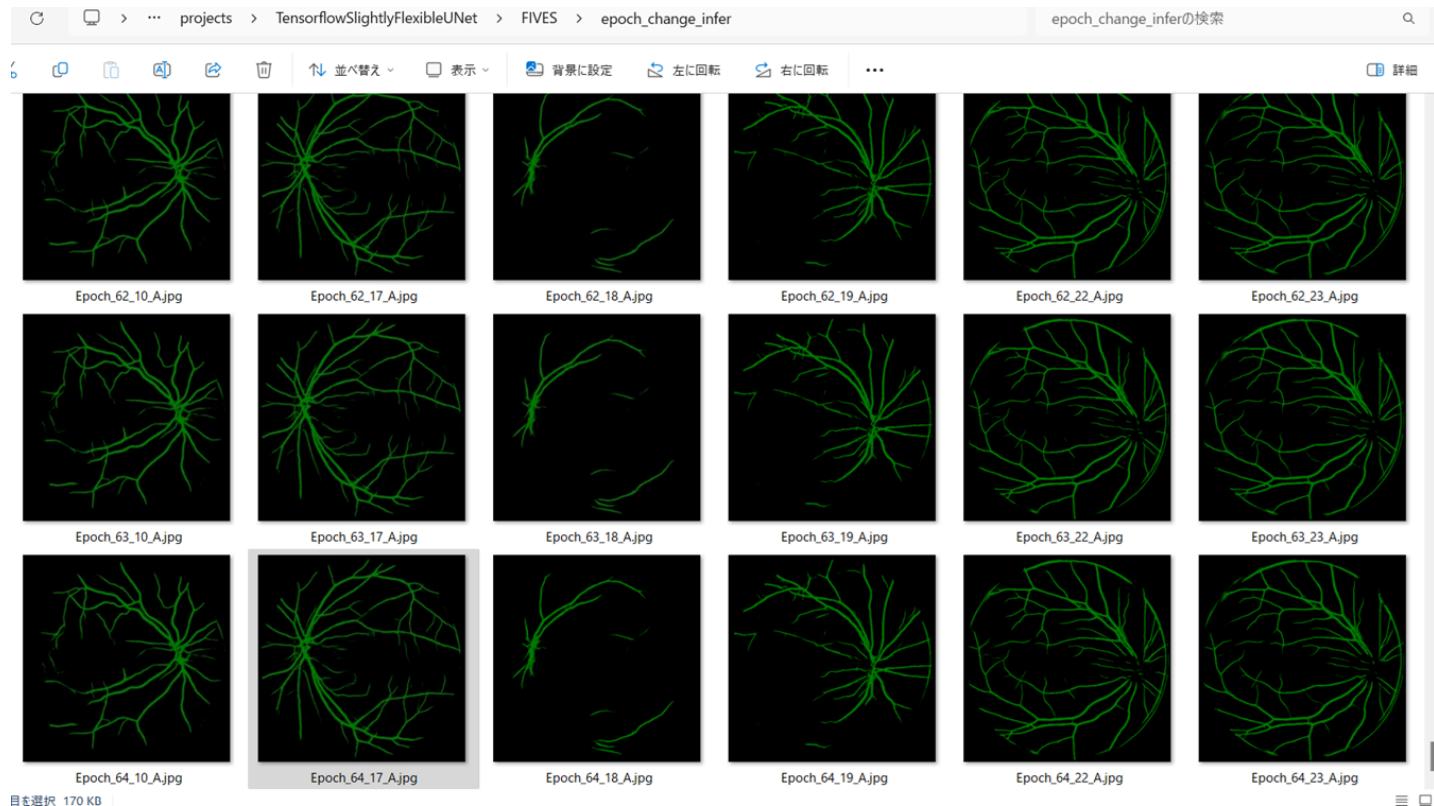
```
[train]
epoch_change_infer      = True
epoch_change_infer_dir   = "./epoch_change_infer"
epoch_changeinfer        = False
epoch_changeinfer_dir    = "./epoch_changeinfer"
num_infer_images         = 6
```

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 (62,63,64)

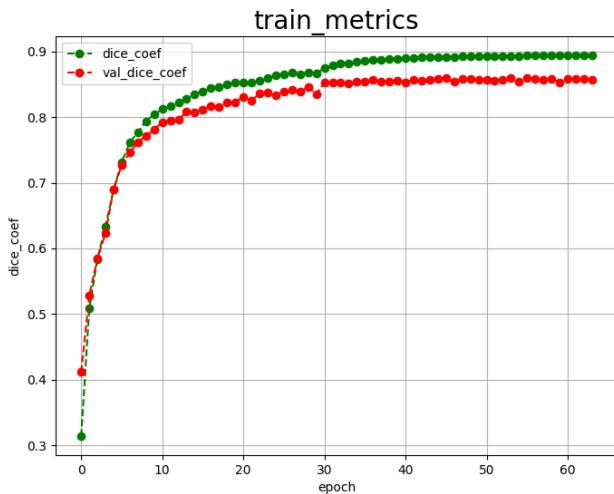


目を選択 170 KB

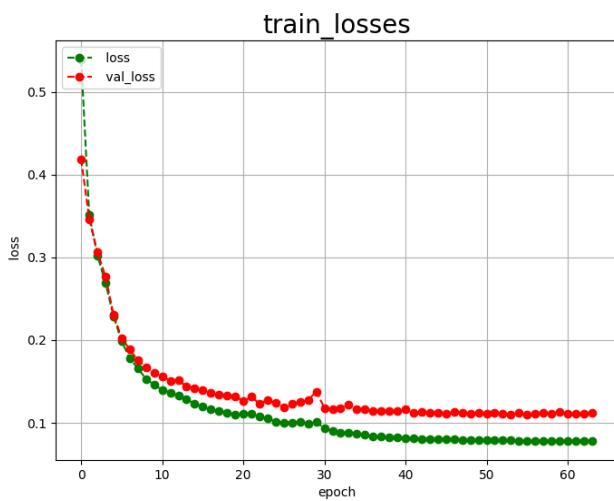
In this experiment, the training process was stopped at epoch 64 by EarlyStopping Callback.

```
PowerShell 7 (x64) - + - x
600/600 [=====] - 284s 474ms/sample - loss: 0.0787 - dice_coef: 0.8931 - val_loss: 0.1111 - val_dice_coef: 0.8570 - lr: 2.5600e-06
Epoch 54/100
600/600 [=====] - ETA: 0s - loss: 0.0787 - dice_coef: 0.8933
Epoch 54: val_loss improved from 0.11056 to 0.11003, saving model to ./model/best_model.h5
600/600 [=====] - 288s 479ms/sample - loss: 0.0787 - dice_coef: 0.8933 - val_loss: 0.1100 - val_dice_coef: 0.8593 - lr: 2.5600e-06
Epoch 55/100
600/600 [=====] - ETA: 0s - loss: 0.0786 - dice_coef: 0.8933
Epoch 55: val_loss did not improve from 0.11003
600/600 [=====] - 285s 475ms/sample - loss: 0.0786 - dice_coef: 0.8933 - val_loss: 0.1123 - val_dice_coef: 0.8544 - lr: 2.5600e-06
Epoch 56/100
600/600 [=====] - ETA: 0s - loss: 0.0785 - dice_coef: 0.8935
Epoch 56: val_loss did not improve from 0.11003
600/600 [=====] - 284s 474ms/sample - loss: 0.0785 - dice_coef: 0.8935 - val_loss: 0.1100 - val_dice_coef: 0.8596 - lr: 2.5600e-06
Epoch 57/100
600/600 [=====] - ETA: 0s - loss: 0.0785 - dice_coef: 0.8935
Epoch 57: val_loss did not improve from 0.11003
600/600 [=====] - 282s 470ms/sample - loss: 0.0785 - dice_coef: 0.8935 - val_loss: 0.1105 - val_dice_coef: 0.8584 - lr: 2.5600e-06
Epoch 58/100
600/600 [=====] - ETA: 0s - loss: 0.0783 - dice_coef: 0.8937
Epoch 58: val_loss did not improve from 0.11003
600/600 [=====] - 278s 464ms/sample - loss: 0.0783 - dice_coef: 0.8937 - val_loss: 0.1117 - val_dice_coef: 0.8563 - lr: 2.5600e-06
Epoch 59/100
600/600 [=====] - ETA: 0s - loss: 0.0783 - dice_coef: 0.8938
Epoch 59: val_loss did not improve from 0.11003
600/600 [=====] - 278s 463ms/sample - loss: 0.0783 - dice_coef: 0.8938 - val_loss: 0.1110 - val_dice_coef: 0.8581 - lr: 1.0240e-06
Epoch 60/100
600/600 [=====] - ETA: 0s - loss: 0.0782 - dice_coef: 0.8939
Epoch 60: val_loss did not improve from 0.11003
600/600 [=====] - 277s 462ms/sample - loss: 0.0782 - dice_coef: 0.8939 - val_loss: 0.1135 - val_dice_coef: 0.8531 - lr: 1.0240e-06
Epoch 61/100
600/600 [=====] - ETA: 0s - loss: 0.0782 - dice_coef: 0.8940
Epoch 61: val_loss did not improve from 0.11003
600/600 [=====] - 279s 465ms/sample - loss: 0.0781 - dice_coef: 0.8940 - val_loss: 0.1112 - val_dice_coef: 0.8581 - lr: 1.0240e-06
Epoch 62/100
600/600 [=====] - ETA: 0s - loss: 0.0782 - dice_coef: 0.8939
Epoch 62: val_loss did not improve from 0.11003
600/600 [=====] - 281s 468ms/sample - loss: 0.0782 - dice_coef: 0.8939 - val_loss: 0.1112 - val_dice_coef: 0.8578 - lr: 1.0240e-06
Epoch 63/100
600/600 [=====] - ETA: 0s - loss: 0.0781 - dice_coef: 0.8938
Epoch 63: val_loss did not improve from 0.11003
600/600 [=====] - 279s 465ms/sample - loss: 0.0781 - dice_coef: 0.8938 - val_loss: 0.1115 - val_dice_coef: 0.8581 - lr: 4.0960e-07
Epoch 64/100
600/600 [=====] - ETA: 0s - loss: 0.0781 - dice_coef: 0.8940
Epoch 64: val_loss did not improve from 0.11003
600/600 [=====] - 279s 465ms/sample - loss: 0.0781 - dice_coef: 0.8940 - val_loss: 0.1124 - val_dice_coef: 0.8564 - lr: 4.0960e-07
Epoch 64: early stopping
Save history.json
```

[train_metrics.csv](#)



[train_losses.csv](#)



4 Evaluation

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

`./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
--- loss <function bce_dice_loss at 0x000001B76AF3D090>
--- metrics [

```

[evaluation.csv](#)

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

```

loss,0.1148
dice_coef,0.8585

```

5 Inference

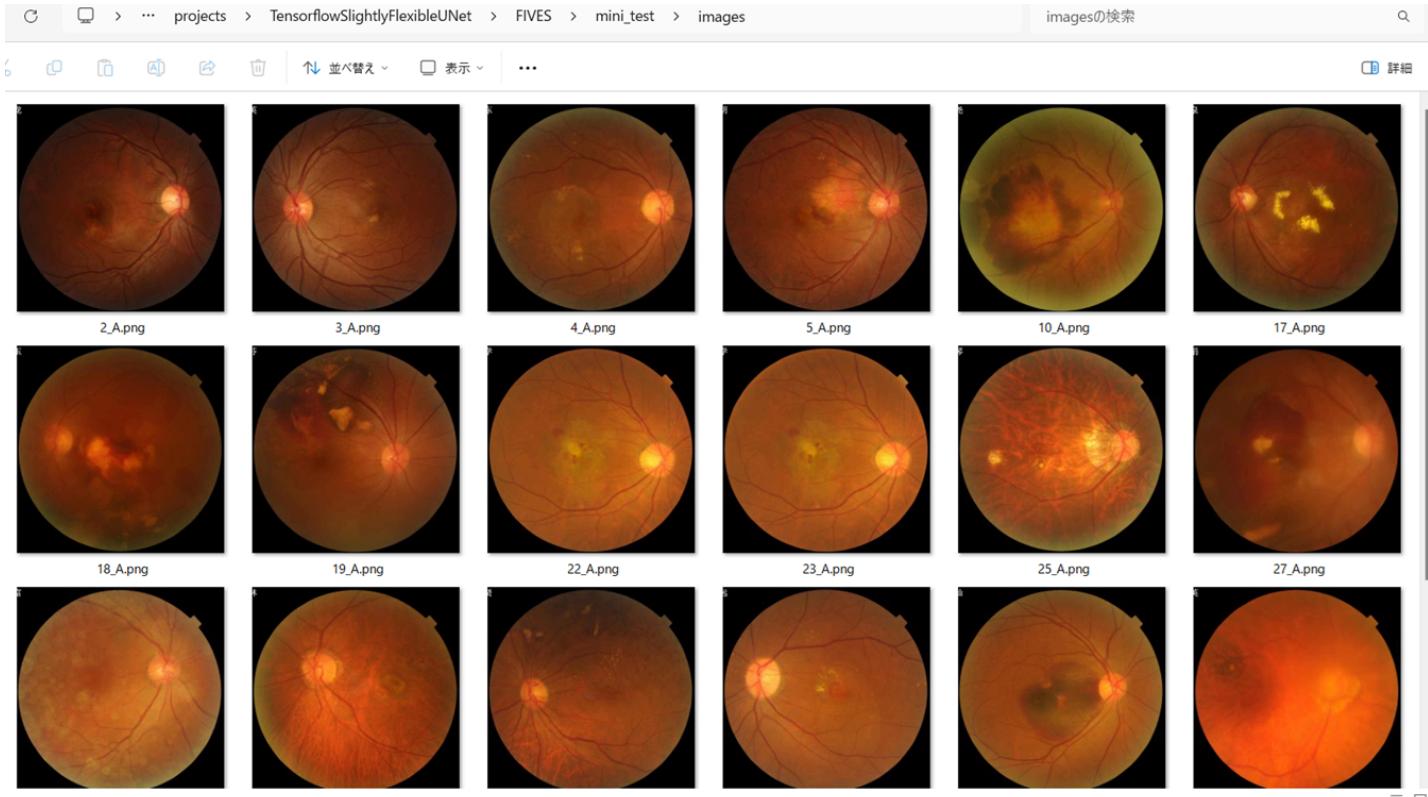
Please move to a **./projects/TensorflowSlightlyFlexibleUNet/FIVES** folder
and run the following bat file to infer segmentation regions for images by the Trained-TensorflowUNet model for FIVES.

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

This simply runs the following command.

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

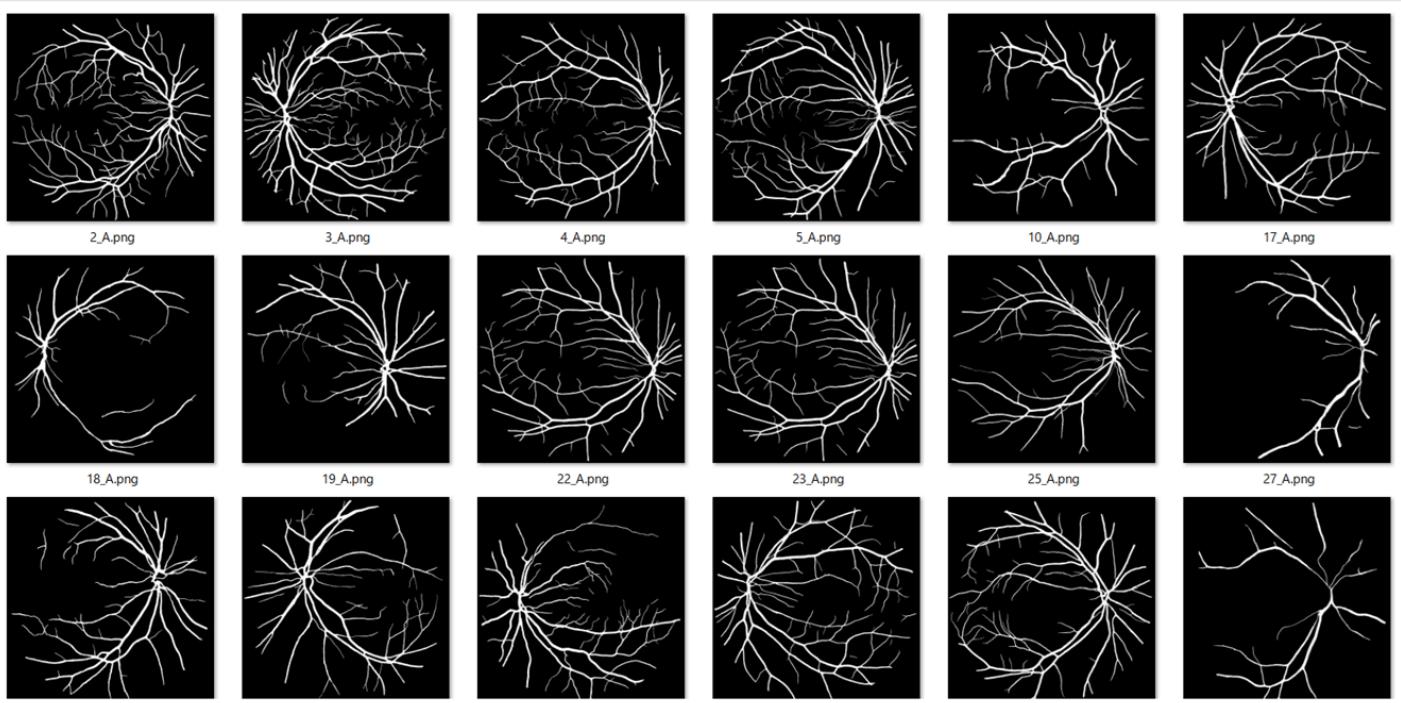
mini_test_images (2048x2048 pixels)



mini_test_mask(ground_truth)

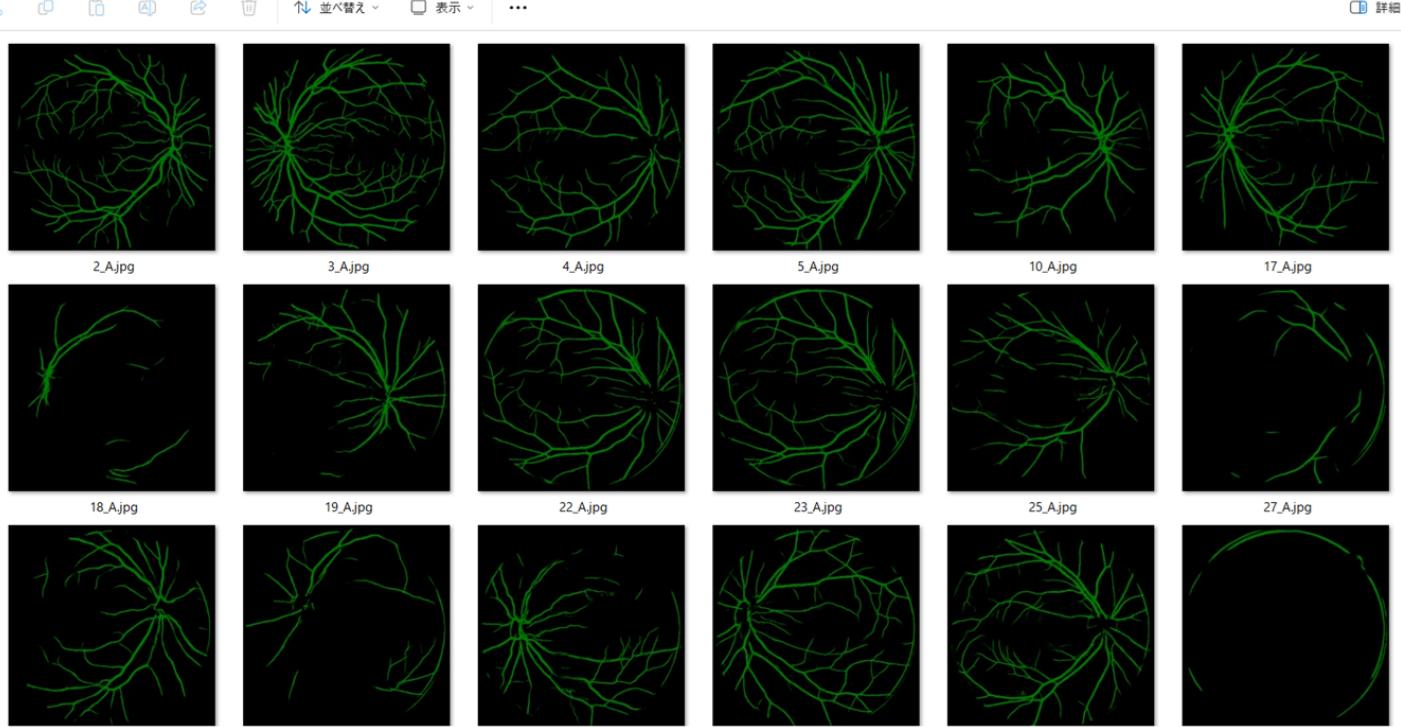
C projects > TensorflowSlightlyFlexibleUNet > FIVES > mini_test > masks

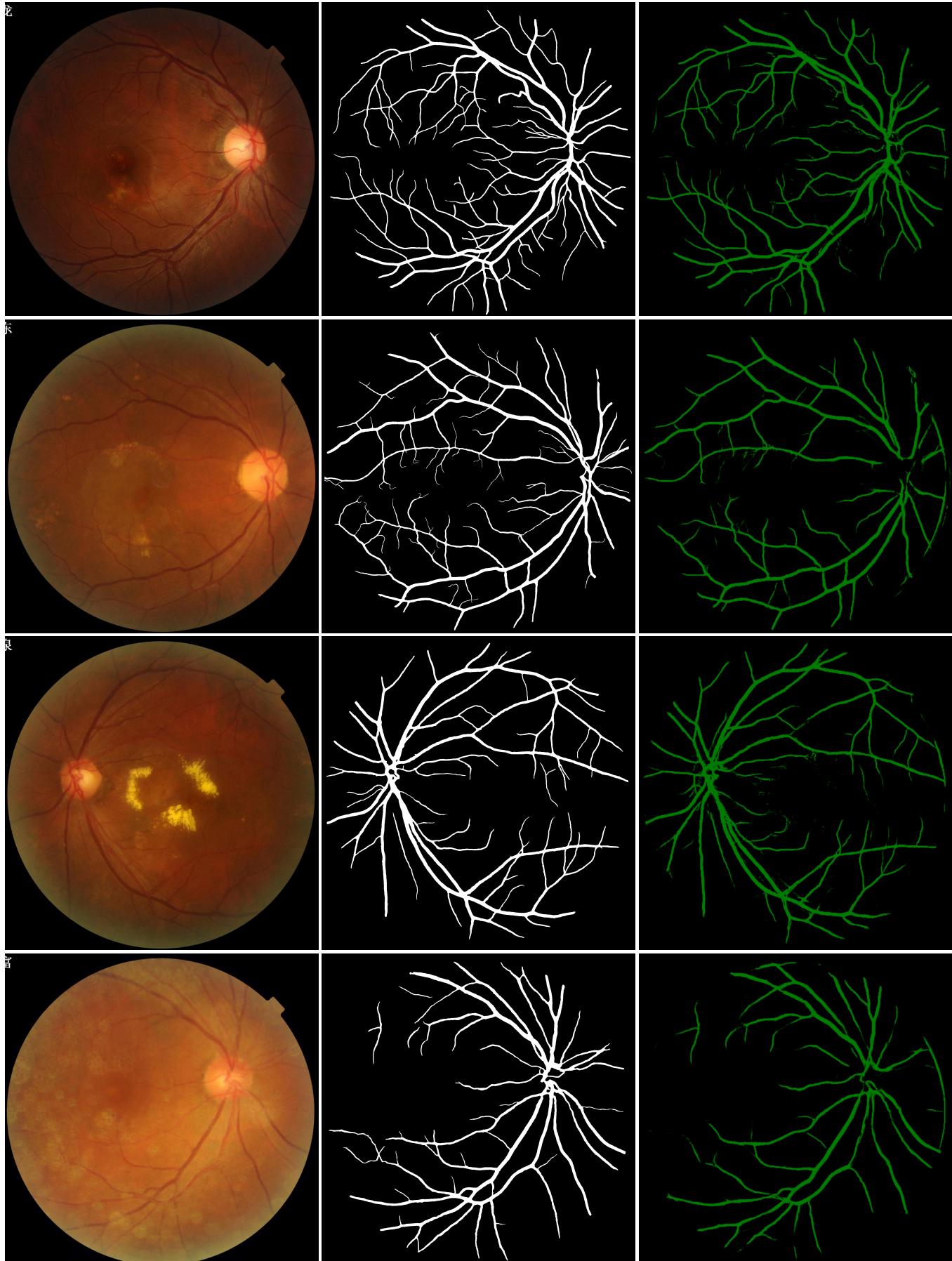
masksの検索

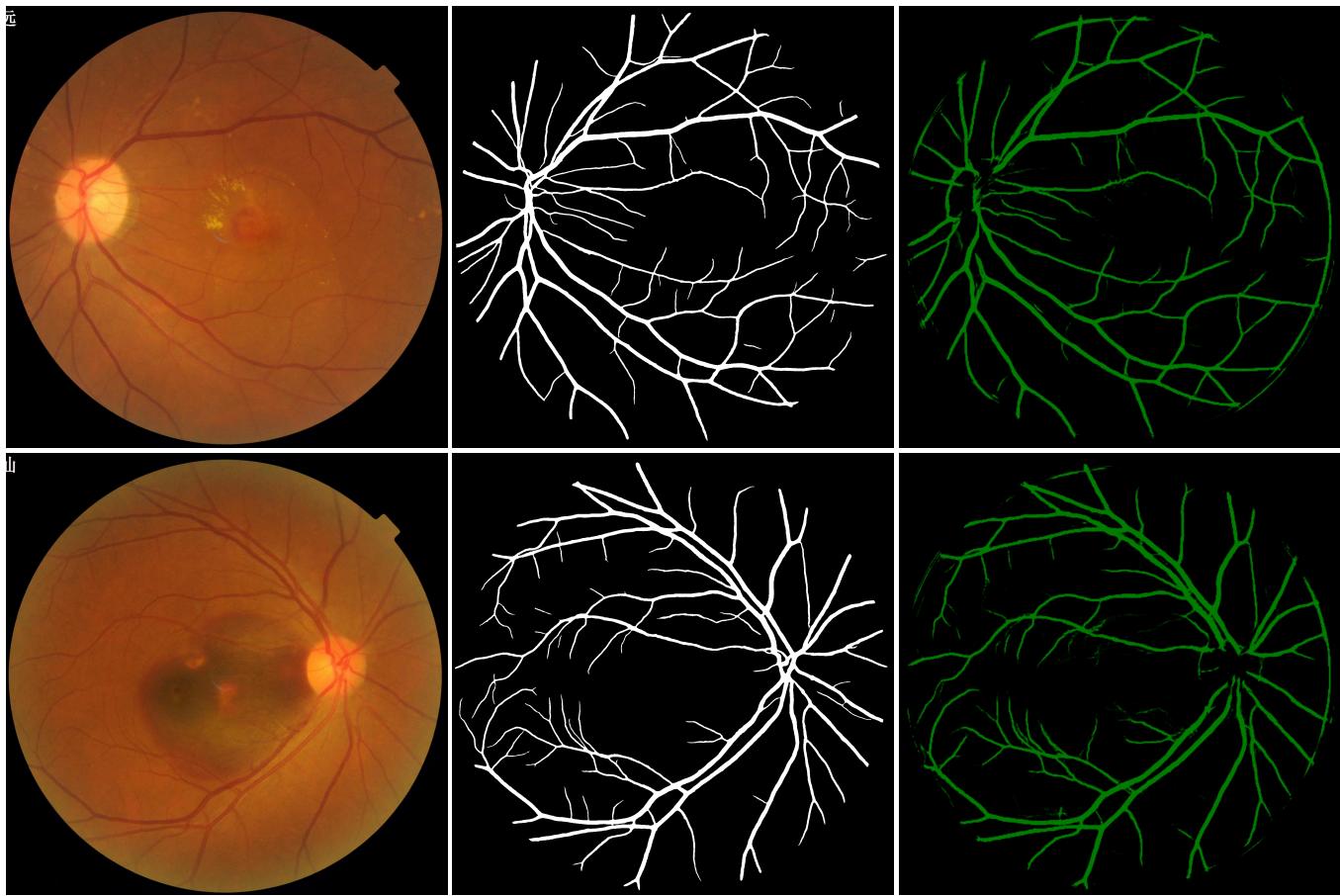
**Inferred test masks (2048x2048 pixels)**

C projects > TensorflowSlightlyFlexibleUNet > FIVES > mini_test_output

mini_test_outputの検索

**Enlarged images and masks (2048x2048 pixels)****Image****Mask (ground_truth)****Inferred-mask**





References

1. 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>

2. FIVES: A Fundus Image Dataset for Artificial Intelligence based Vessel Segmentation

Kai Jin, Xingru Huang, Jingxing Zhou, Yunxiang Li, Yan Yan, Yibao Sun, Qianni Zhang, Yaqi Wang & Juan Ye

<https://www.nature.com/articles/s41597-022-01564-3>