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

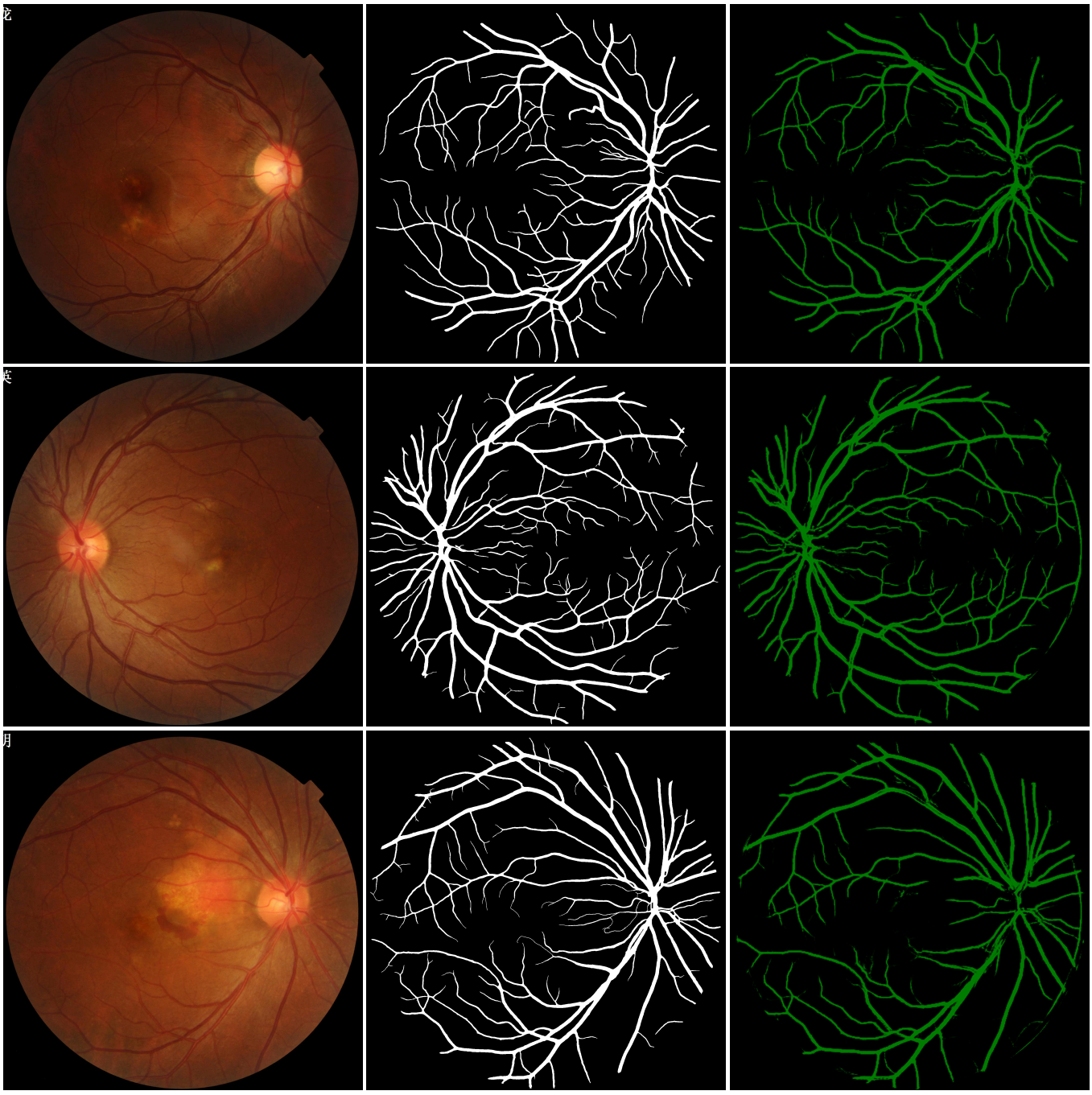
Sarah T. Arai
Software Laboratory antillia.com

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 STructured 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.

Input: image	Mask (ground_truth)	Prediction: inferred_mask



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

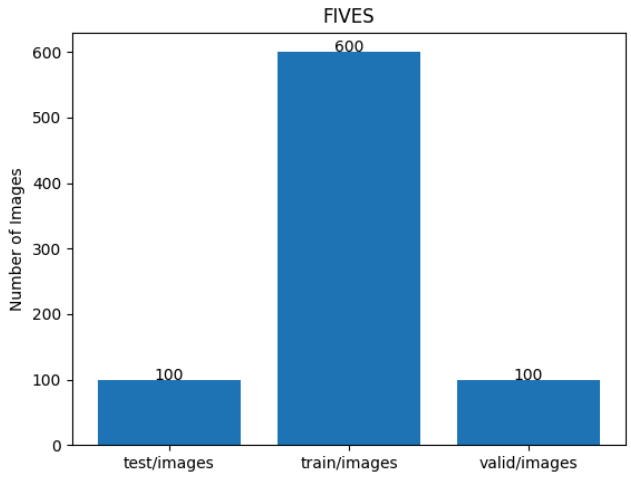
If you would like to train this FIVESSegmentation model by yourself, please download the dataset from **FIVES: A Fundus Image Dataset for AI-based Vessel Segmentation**
The folder structure of the dataset is the following,

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

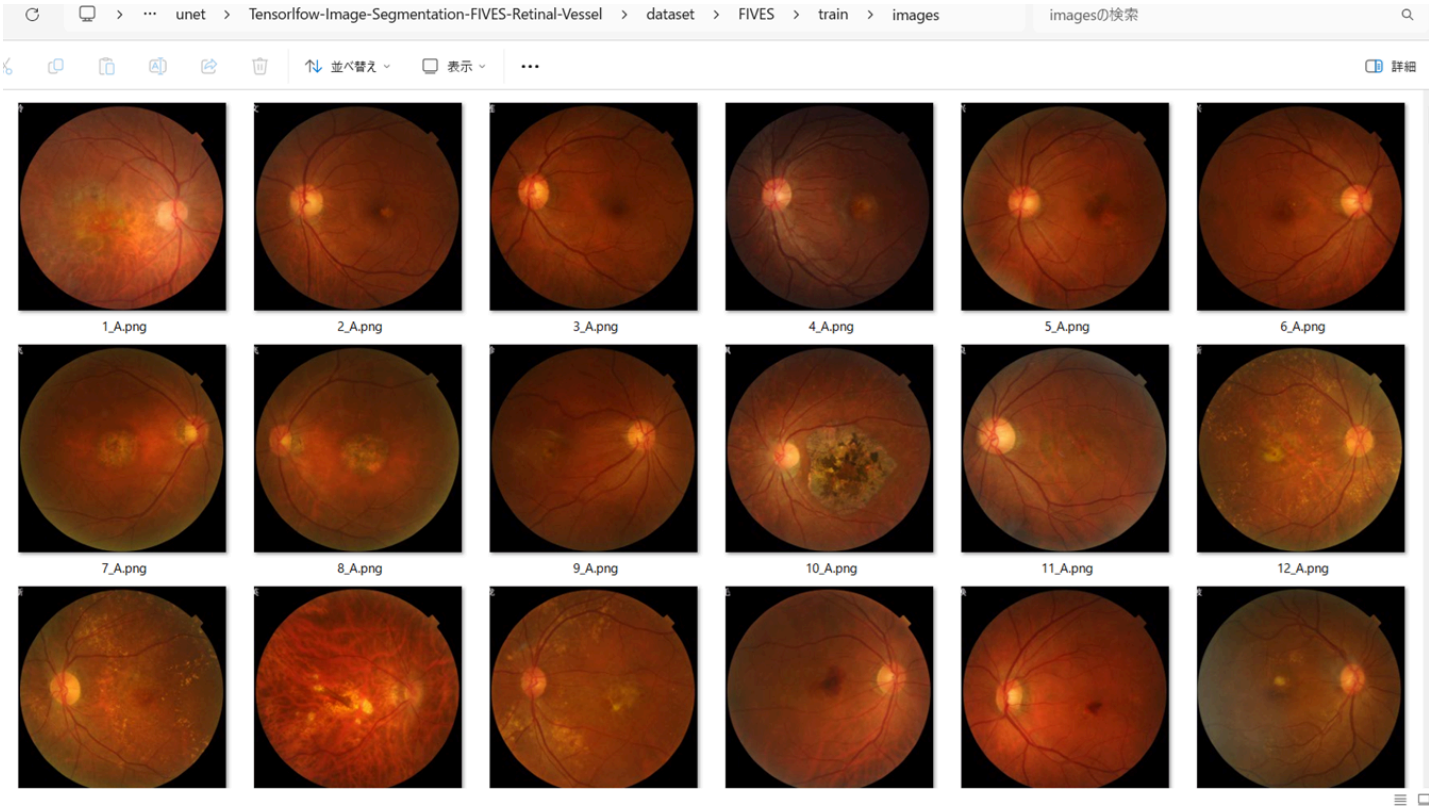
As shown below, we splitted the original dataset into **test**, **train** and **valid** subsets.

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

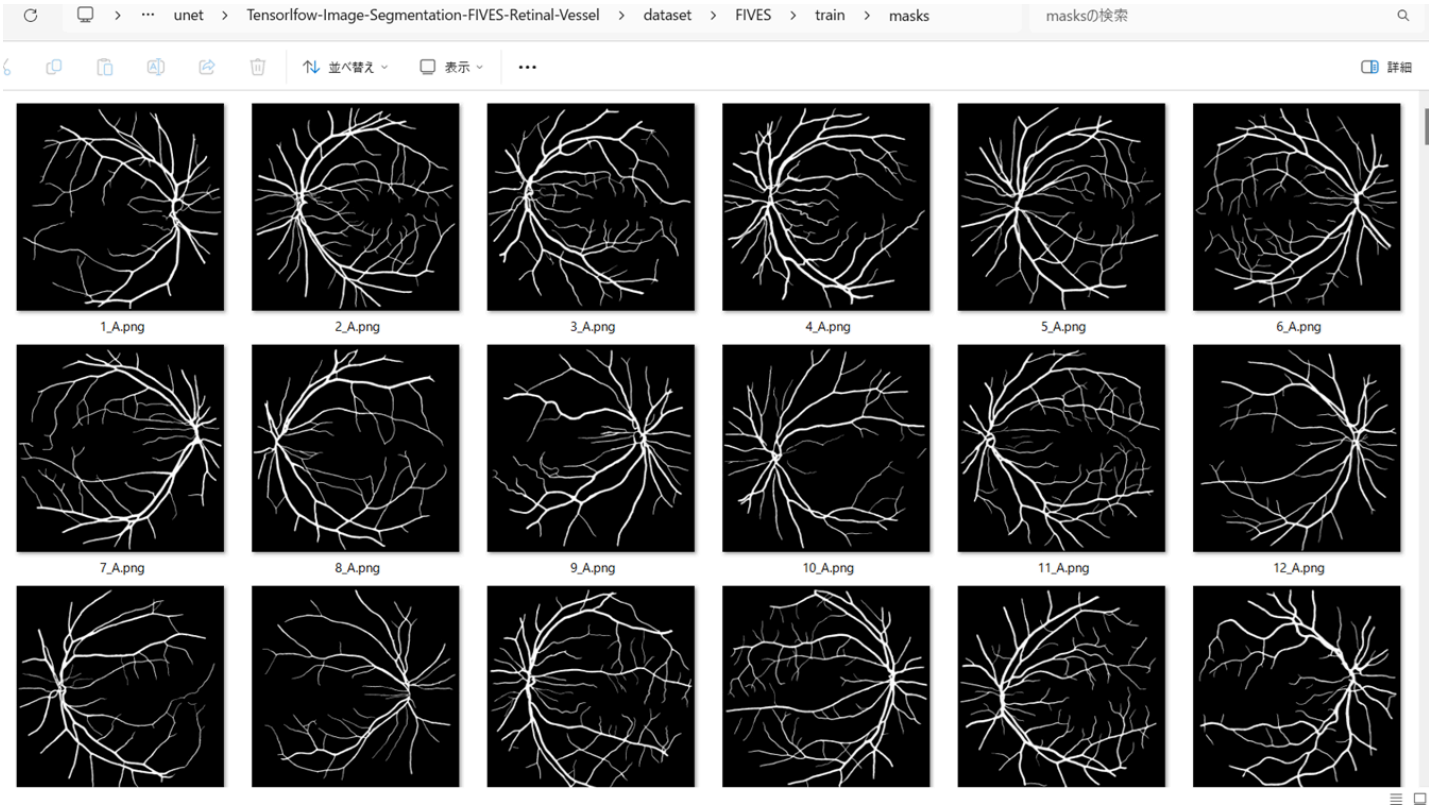
FIVES Dataset Statistics



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 learing_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"
```

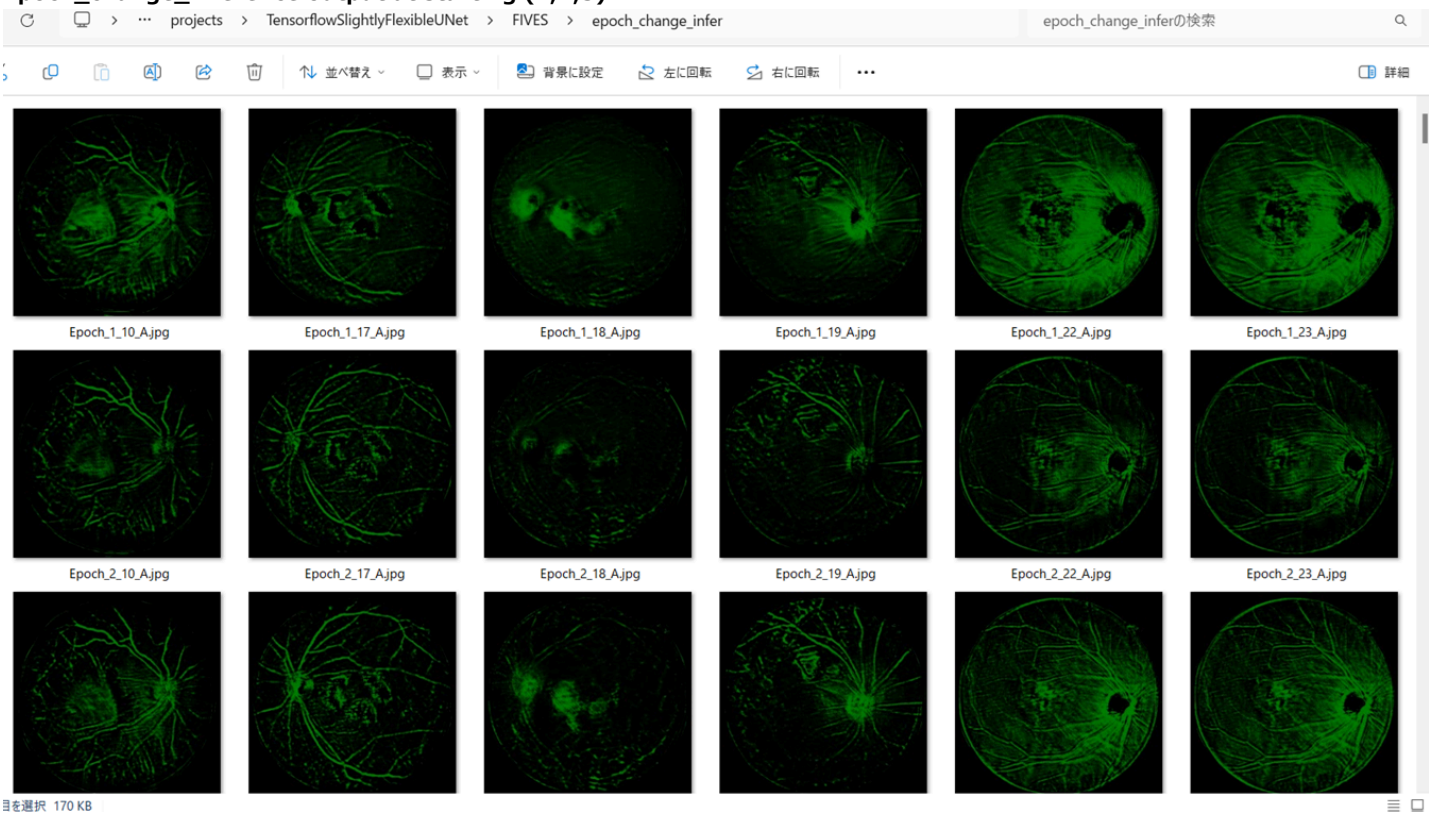
Epoch change inference callbacks

Enabled epoch_change_infer callback.

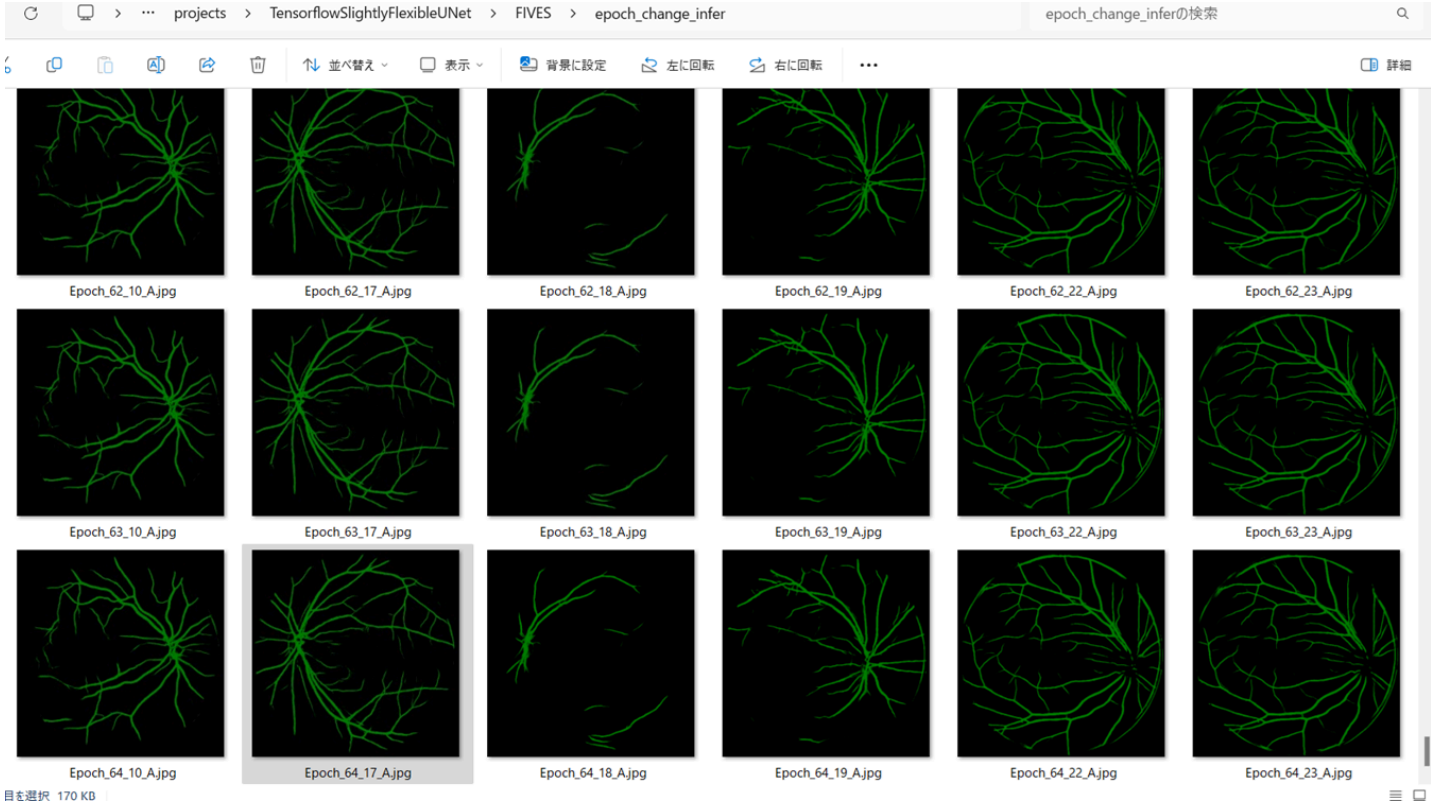
```
[train]
epoch_change_infer       = True
epoch_change_infer_dir   =  "./epoch_change_infer"
epoch_changeinfer        = False
epoch_changeinfer_dir    = "./epoch_changeinfer"
num_infer_images         = 6
```

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.

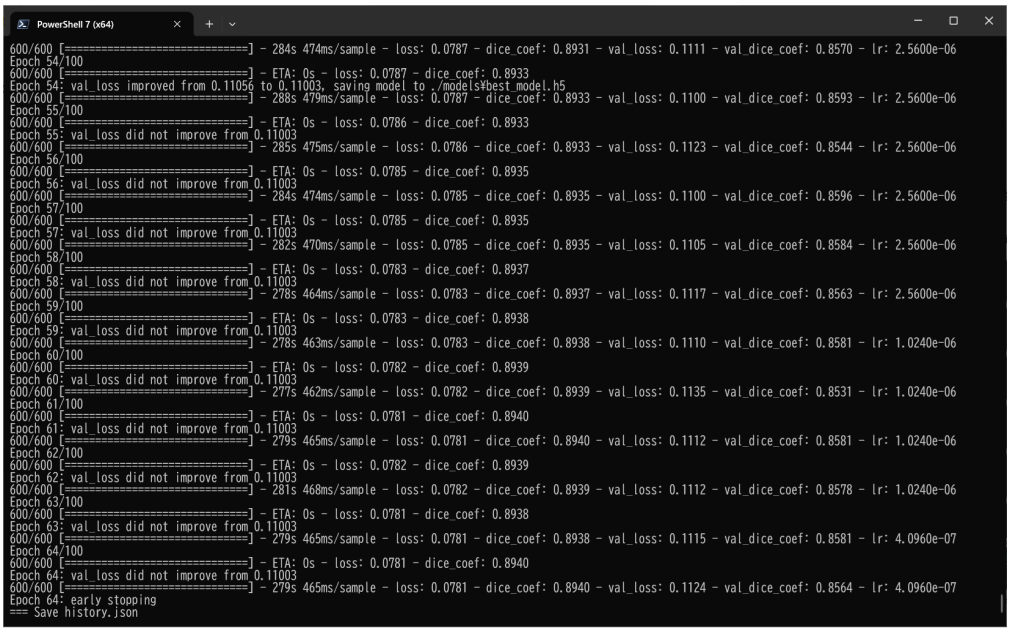
Epoch_change_inference output at starting (1,2,3)



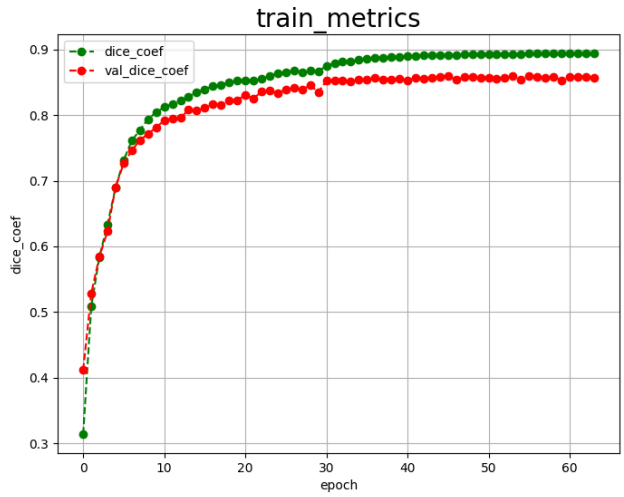
Epoch_change_inference output at ending (62,63,64)



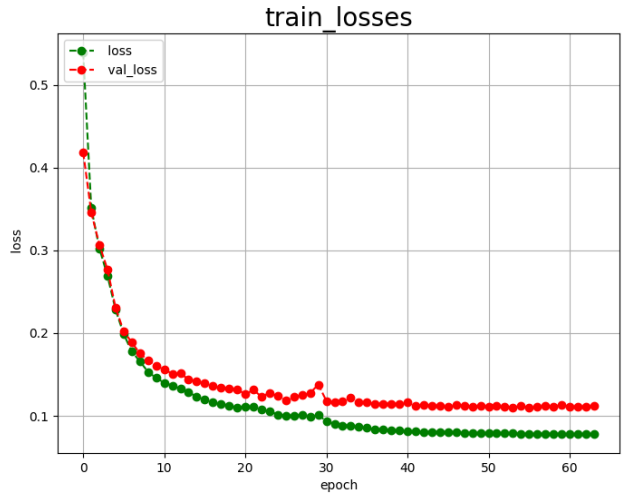
In this experiment, the training process was stopped at epoch 64 by EarlyStopping Callback.



train_metrics.csv



train_losses.csv



4 Evaluation

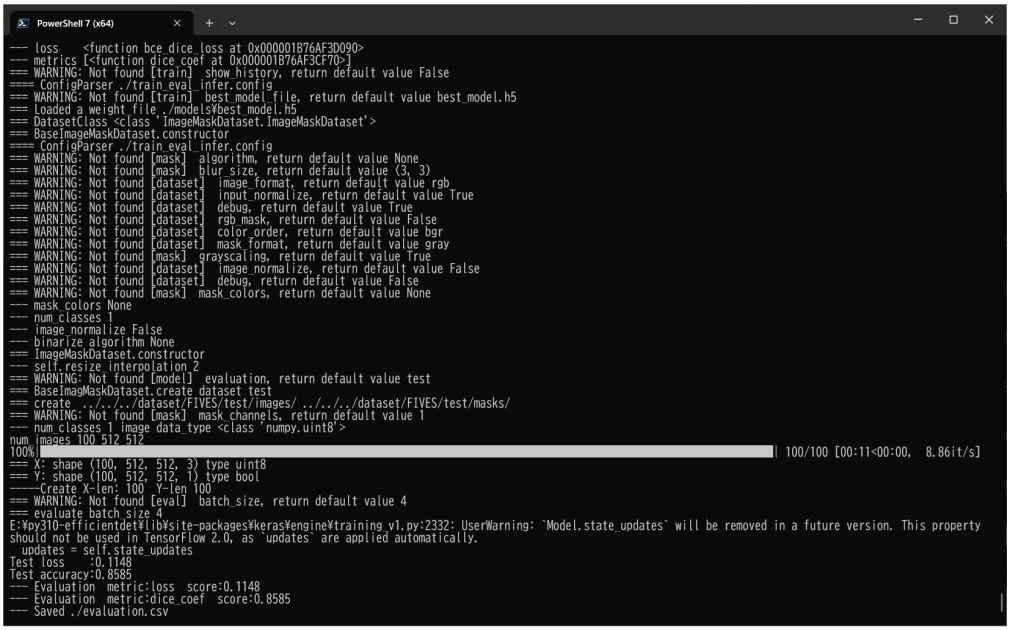
Please move to a **./projects/TensorflowSlightlyFlexibleUNet/FIVES** folder,
and run the following bat file to evaluate TensorflowUNet model for FIVES.

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

This bat file simply runs the following command.

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

Evaluation console output:



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.

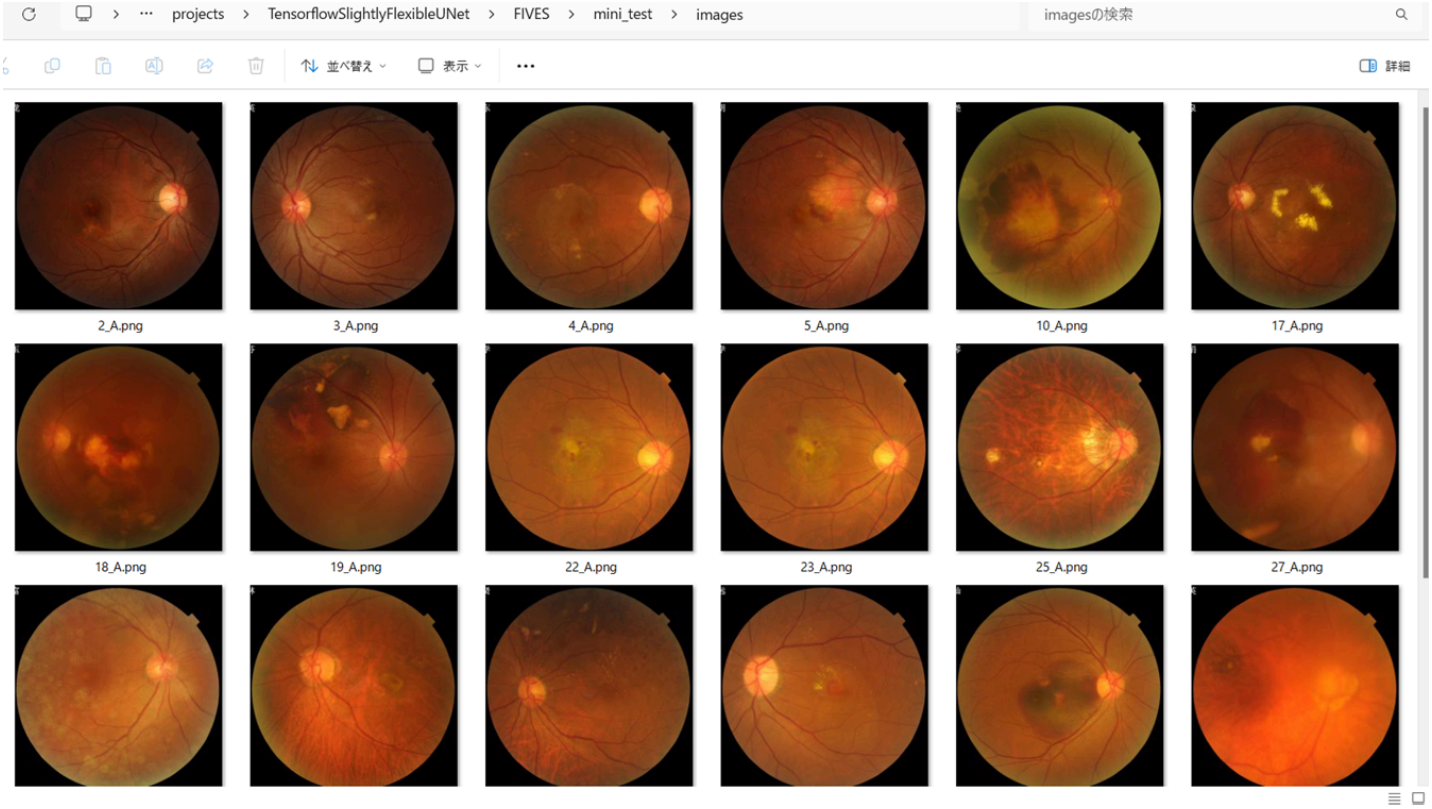
```
./3.infer.bat
```

This simply runs the following command.

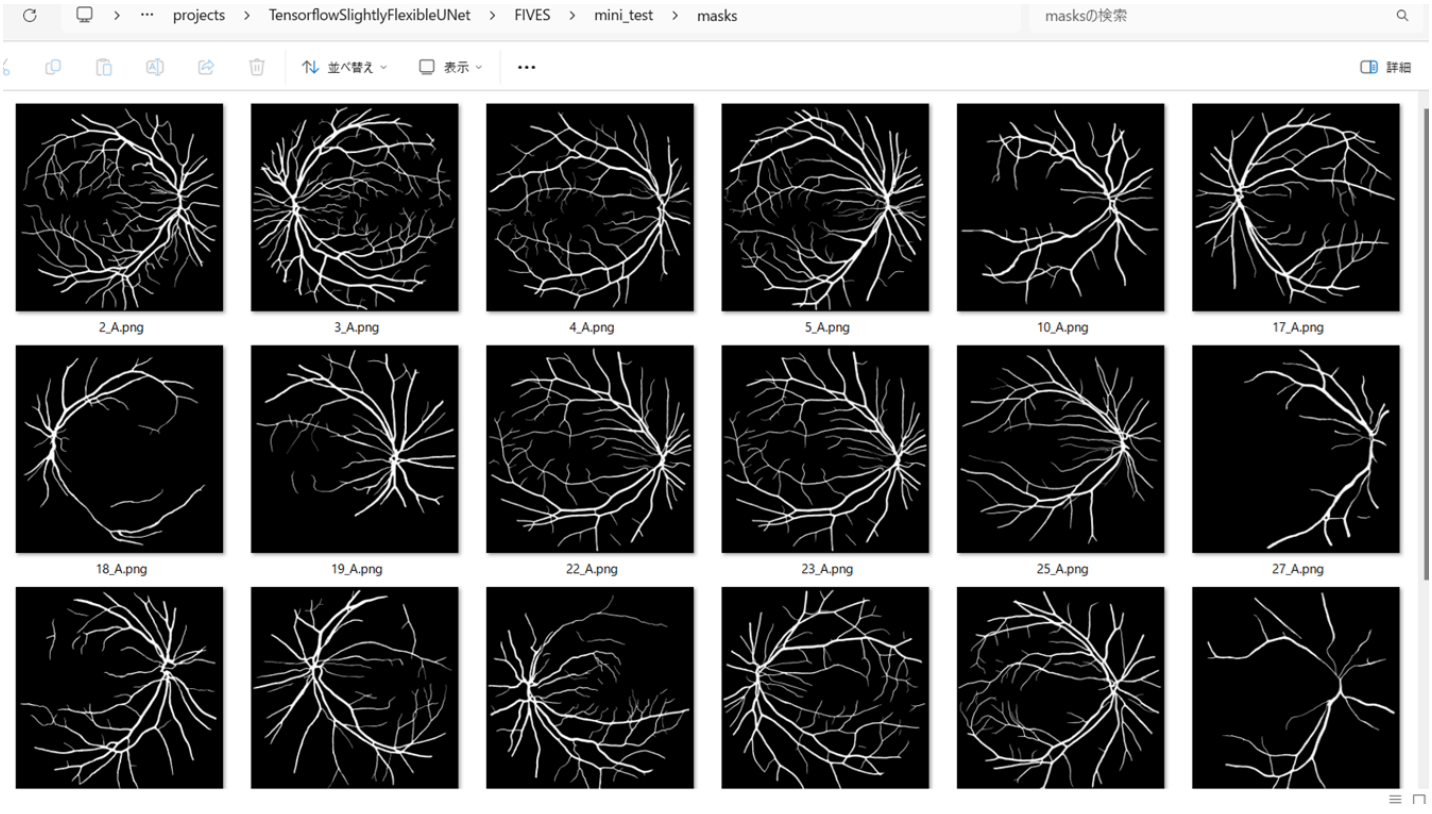
```
python ../../../src/TensorflowUNetInferencer.py ./train_eval_infer_aug.config
```

---

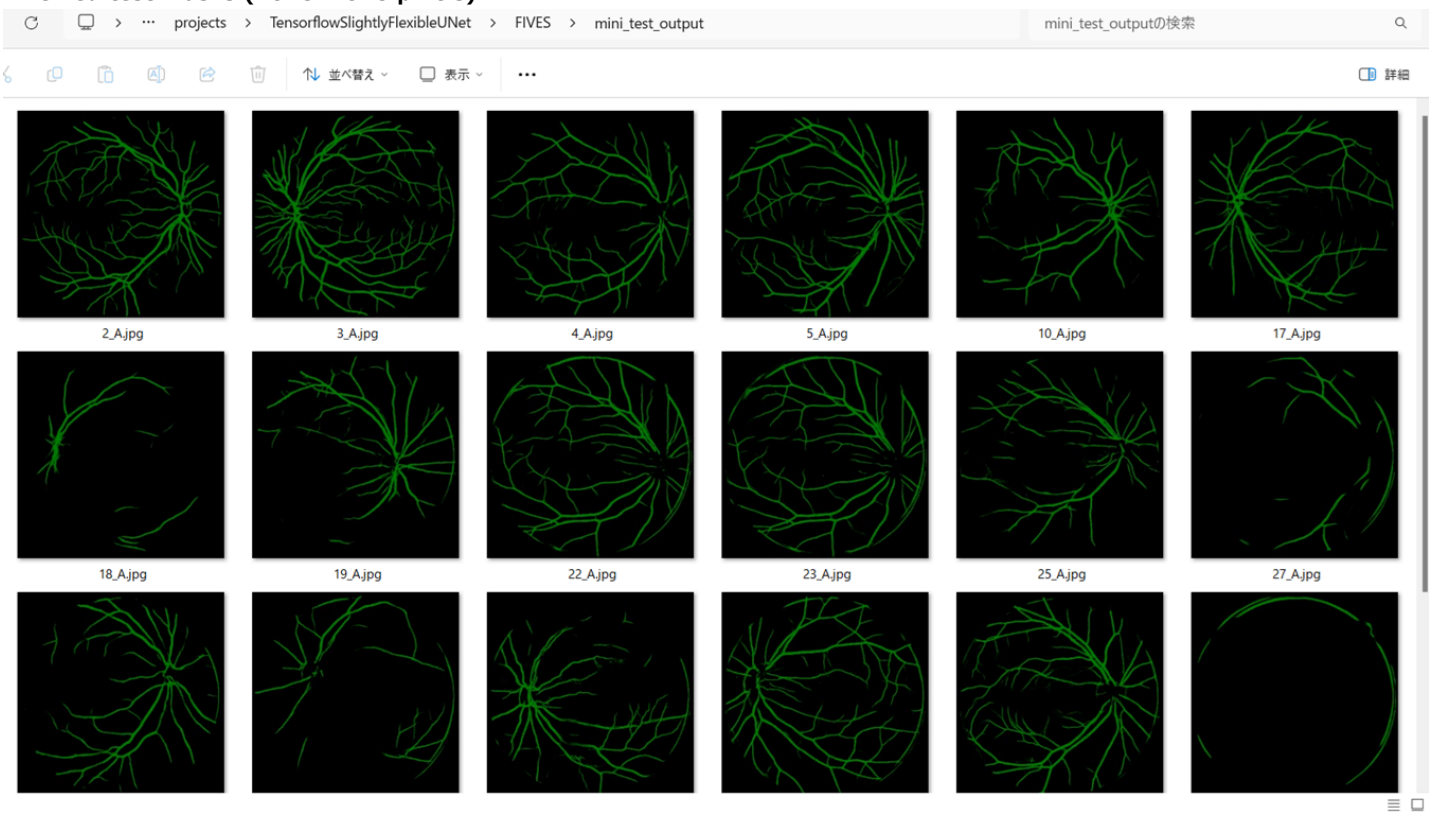
mini_test_images (2048x2048 pixels)



mini_test_mask(ground_truth)

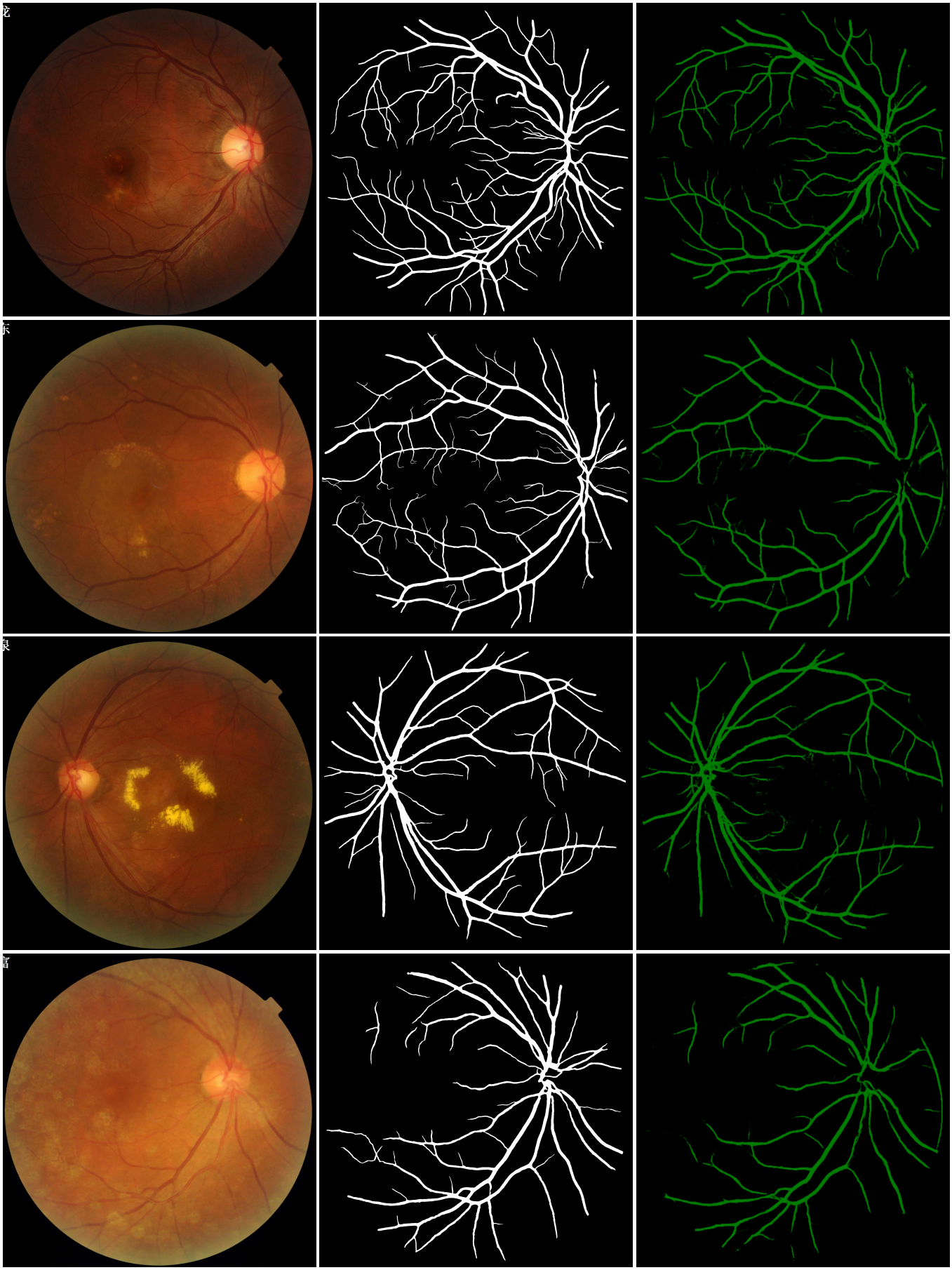


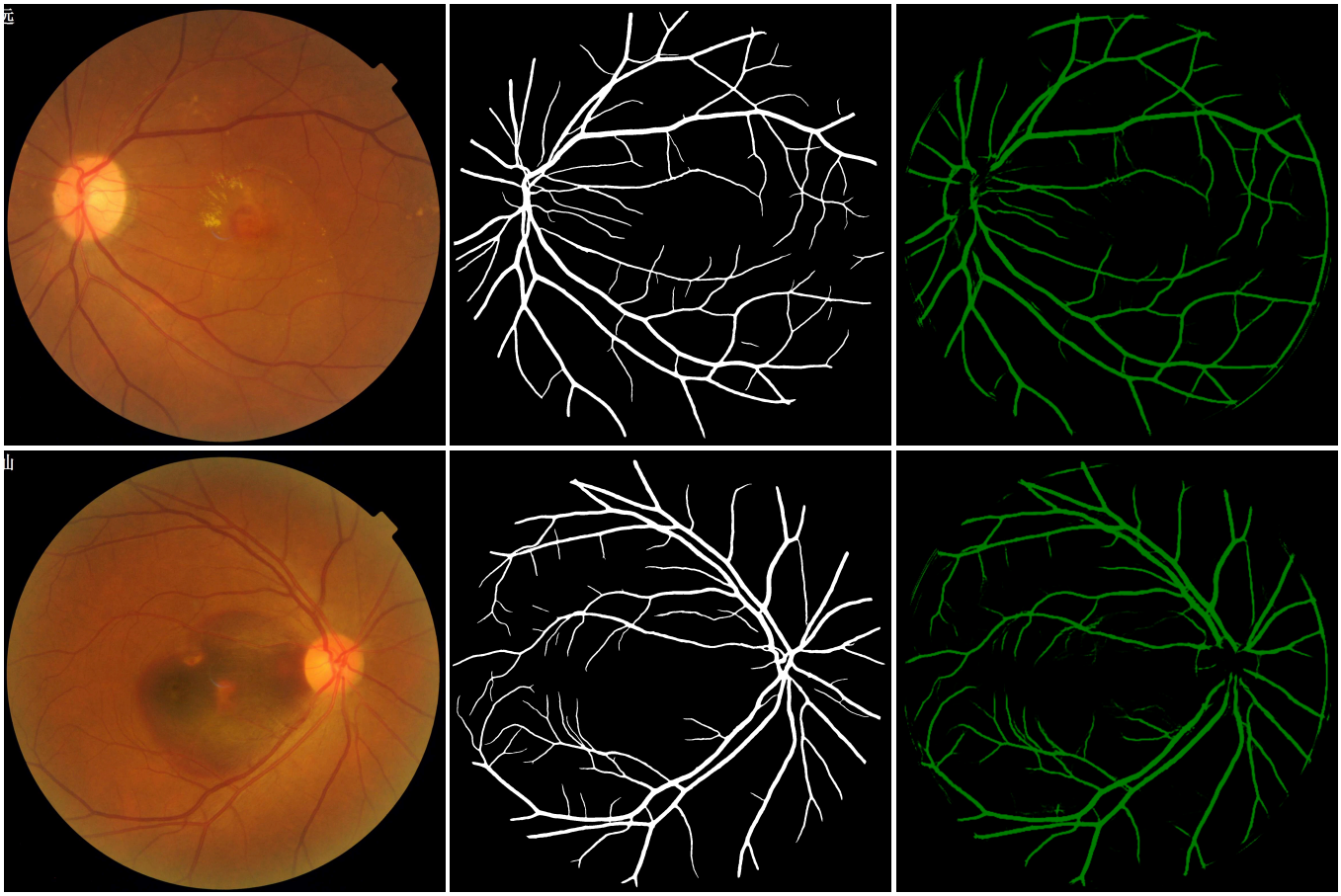
Inferred test masks (2048x2048 pixels)



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