Tensorflow-Tiled-Image-Segmentation-IDRiD-Haemorrhages (2025/04/08)

Sarah A. Arai
Software Laboratory antillia.com

This is the first experiment of Tiled Image Segmentation for IDRiD-Haemorrhages based on the latest Tensorflow-Image-Segmentation-API, and a tiled dataset Tiled-IDRiD-Haemorrhages-ImageMask-Dataset.zip, which was derived by us from Segmentation dataset of **Indian Diabetic Retinopathy Image Dataset (IDRiD)**

Please see also our experiment Tensorflow-Tiled-Image-Segmentation-Pre-Augmented-IDRiD-HardExudates

Tiled Image Segmentation
This is one of the divide and conquer algorithms in UNet segmentation model.
The pixel size of the images of IDRiD is 4288x2848, which is too large to use for a training dataset of a simple UNet model. For segmentation of large images using UNet, employing the algorithm is a good idea. Since large images cannot be processed all at once, they are divided into small tiles. Segmentation processing is then applied to these divided small images, and finally, they are reassembled to their original size.

Experiment Strategies
In this experiment, we employed the following strategies.
1. Create Tiled ImageMask Dataset
We created Tiled-IDRiD-Haemorrhages-ImageMask-Dataset, which was tiledly-splitted to 512x512 pixels and reduced to 512x512 pixels image and mask dataset from the original 4288x2848 pixels images and mask files.

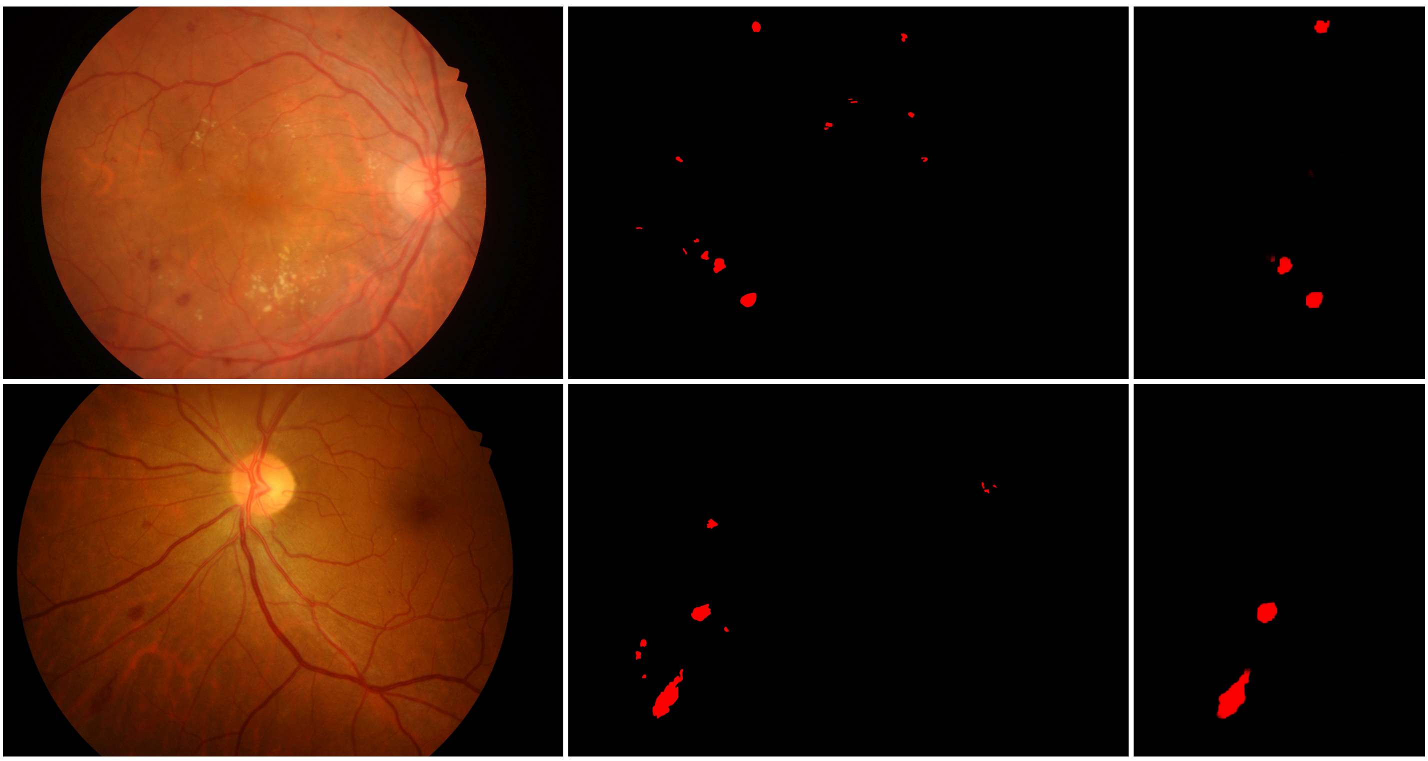
2. Train UNet Model by Tiled ImageMask Dataset
We trained and validated a TensorFlow UNet model using the Tiled IDRiD Haemorrhages train and valid datasets, which was tiledly-splitted to 512x512 pixels and reduced to 512x512 pixels image and mask dataset from the original 4288x2848 pixels images and mask files.

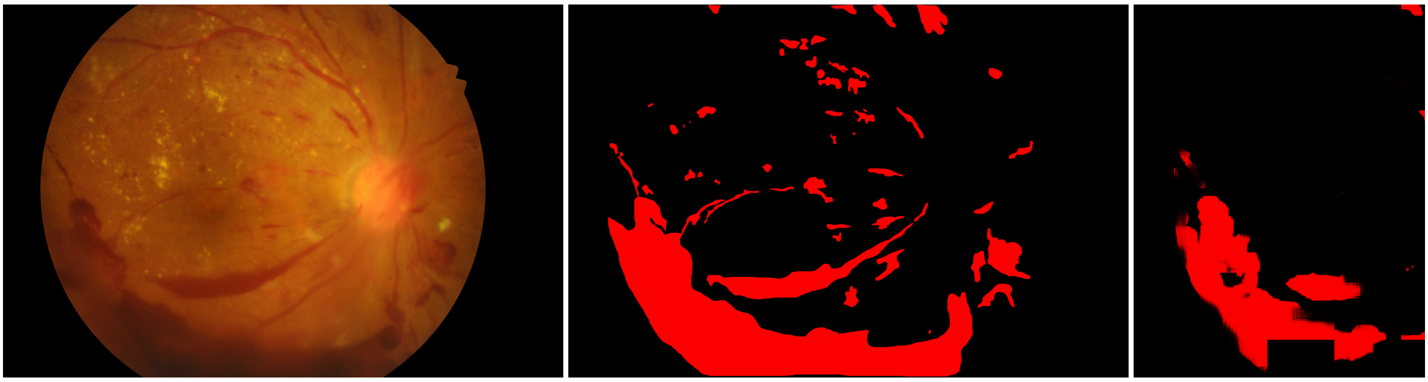
3. Evaluate UNet Model performance
We evaluated the performance of the trained TensorFlow UNet model by using the Tiled IDRiD Haemorrhages test dataset

4. Apply Tiled Image Segmentation
We applied our Tiled-Image Segmentation inference method to predict the Haemorrhages regions for the mini_test images with a resolution of 4288x2848 pixels.

Actual Tiled Image Segmentation for Images of 4288x2848 pixels
As shown below, the segmentation result is unsatisfactory, the inferred masks look somewhat similar to the ground truth masks, but they lack precision in certain areas.

Input: image	Mask (ground_truth)	Prediction: Tiled_





In this experiment, we used the simple UNet Model TensorflowSlightlyFlexibleUNet for this IDRiD-Haemorrhages 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 take from the following **IEEE DataPort** web site
Indian Diabetic Retinopathy Image Dataset (IDRiD)

Please see also **DIABETIC RETINOPATHY: SEGMENNTATION AND GRAND CHALLENGE**

Citation Author(s):
Prasanna Porwal, Samiksha Pachade, Ravi Kamble, Manesh Kokare, Girish Deshmukh,
Vivek Sahasrabuddhe, Fabrice Meriaudeau,
April 24, 2018, "Indian Diabetic Retinopathy Image Dataset (IDRiD)", IEEE Dataport,

DOI: https://dx.doi.org/10.21227/H25W98

License:

2 Tiled-IDRiD-Haemorrhages ImageMask Dataset

If you would like to train this IDRiD-Haemorrhages Segmentation model by yourself, please download the dataset from the google drive Tiled-IDRiD-Haemorrhages-ImageMask-Dataset.zip, expand the downloaded ImageMaskDataset and put it under **./dataset** folder to be

```
./dataset
└─Tiled-IDRiD-Haemorrhages
    ├─test
    │  ├─images
    │  └─masks
    ├─train
    │  ├─images
    │  └─masks
    └─valid
        ├─images
        └─masks
```

This is a 512x512 pixels tiles dataset generated from 4288x2848 pixels 53 **Original Images** and their corresponding **Hard Exudates GroundTruths** in Training Set.

.

We excluded all black (empty) masks and their corresponding images to generate our dataset from the original one.
The folder structure of the original Segmentation data is the following.

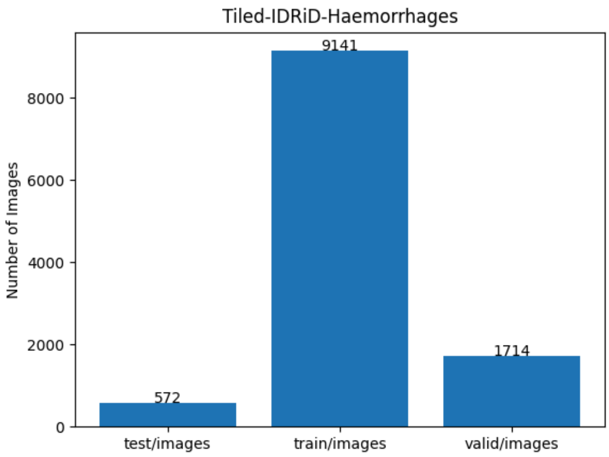
```
./A. Segmentation
 ├─1. Original Images
 │   ├─a. Training Set
 │   └─b. Testing Set
 └─2. All Segmentation Groundtruths
     ├─a. Training Set
     │  ├─1. Microaneurysms
     │  ├─2. Haemorrhages
     │  ├─3. Hard Exudates
     │  ├─4. Soft Exudates
     │  └─5. Optic Disc
     └─b. Testing Set
```

```
    ├─1. Microaneurysms
    ├─2. Haemorrhages
    ├─3. Hard Exudates
    ├─4. Soft Exudates
    └─5. Optic Disc
```

On the derivation of this tiled dataset, please refer to the following Python scripts.

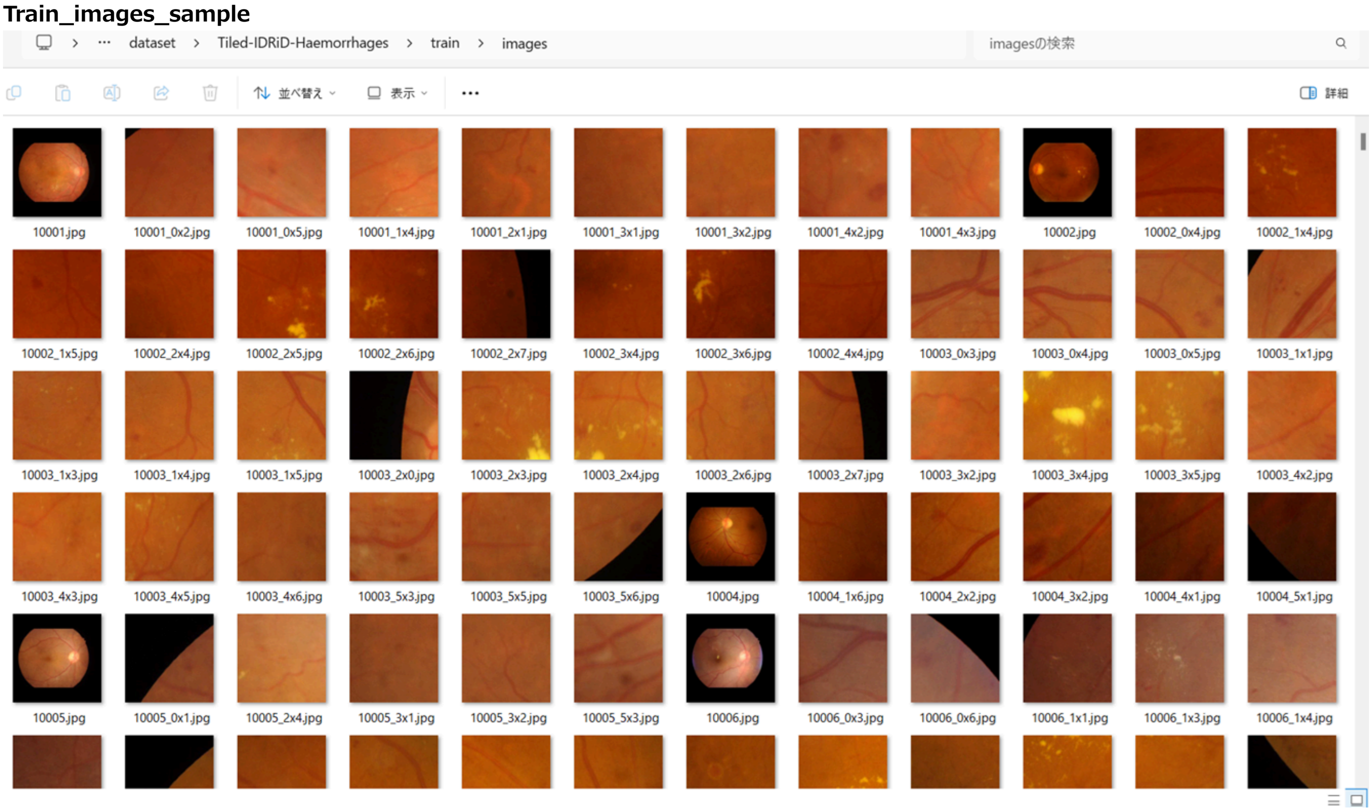
- [TiledImageMaskDatasetGenerator.py]
- [split_tiled_master.py]

Tiled-IDRiD-Haemorrhages Statistics

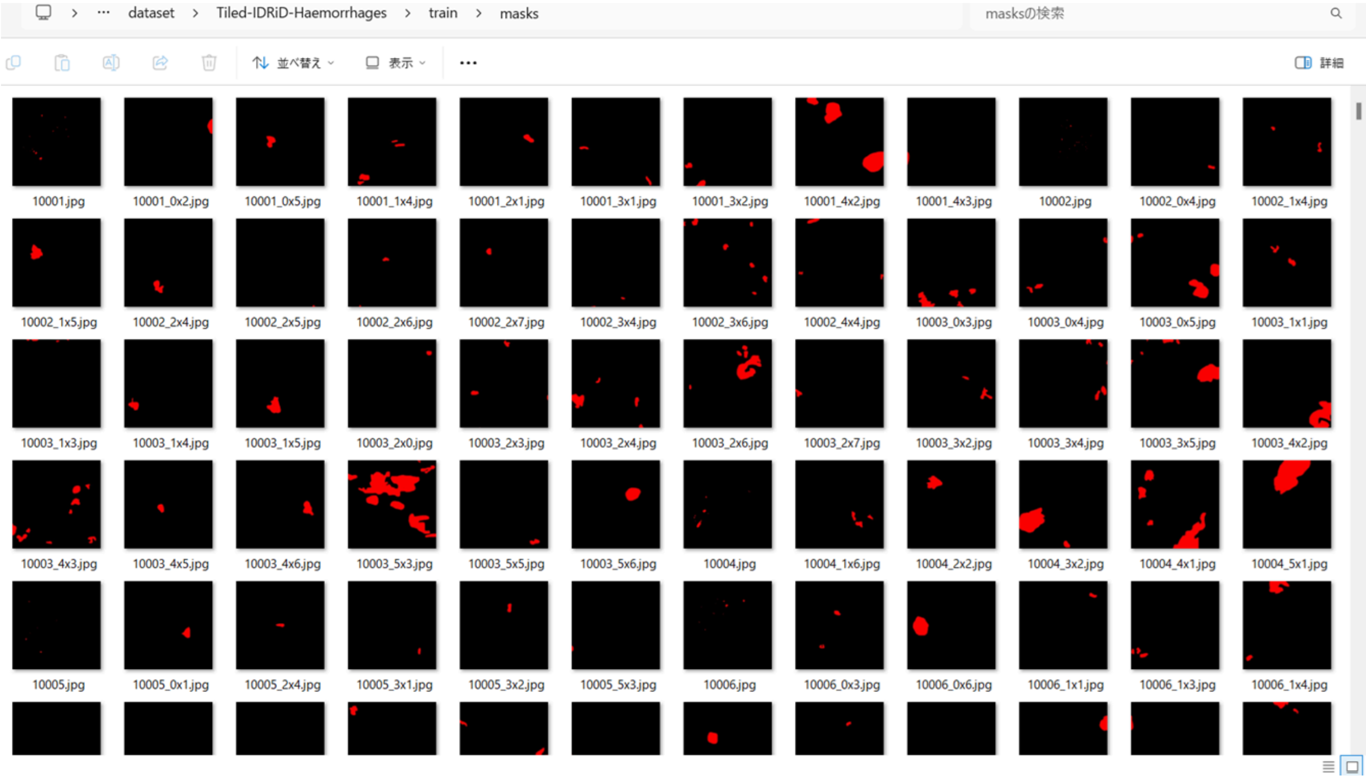


As shown above, the number of images of train and valid datasets is enough to use for a training set of our segmentation model.

Train_images_sample



Train_masks_sample



3 Train TensorflowUNet Model

We trained IDRiD-Haemorrhages TensorflowUNet Model by using the following **train_eval_infer.config** file.
Please move to ./projects/TensorflowSlightlyFlexibleUNet/IDRiD-Haemorrhagesand 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=16** and large **base_kernels=(9,9)** 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   = (9,9)
num_layers     = 8
dilation       = (3,3)
```

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 learing_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_CUBIC"
```

Early stopping callback

Enabled early stopping callback with patience parameter.

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

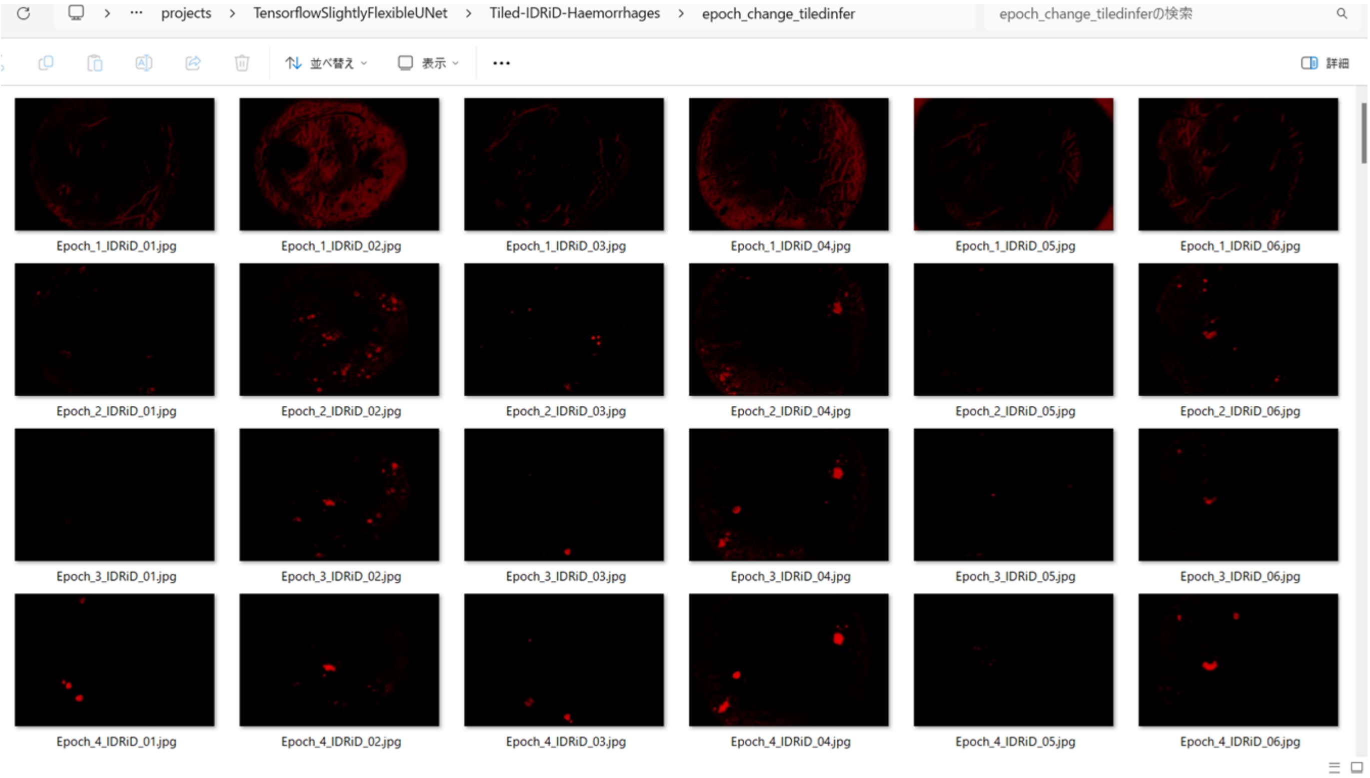
Epoch change inference callbacks

Enabled epoch_change_infer callback.

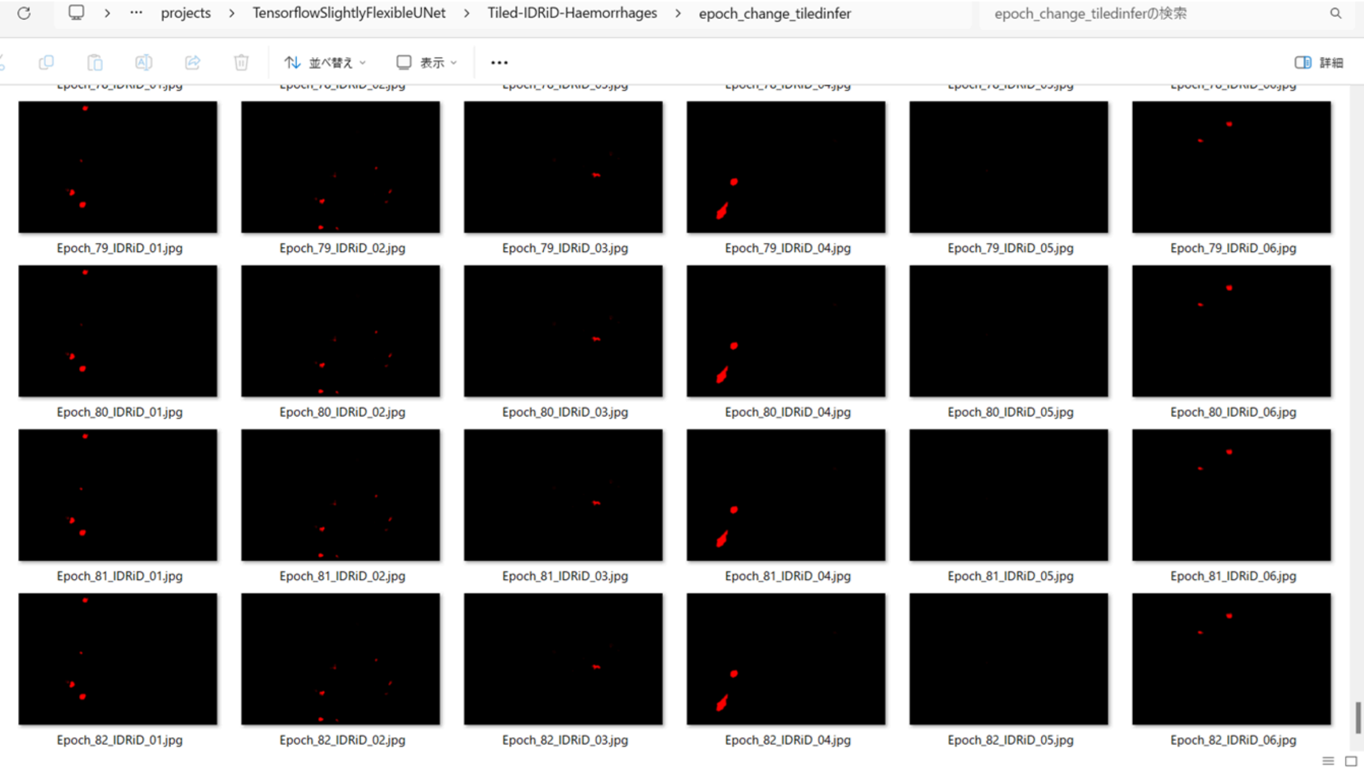
```
[train]

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 tiledinfer procedure can be called for 6 image in **mini_test** folder. This will help you confirm how the predicted mask changes at each epoch during your training process.

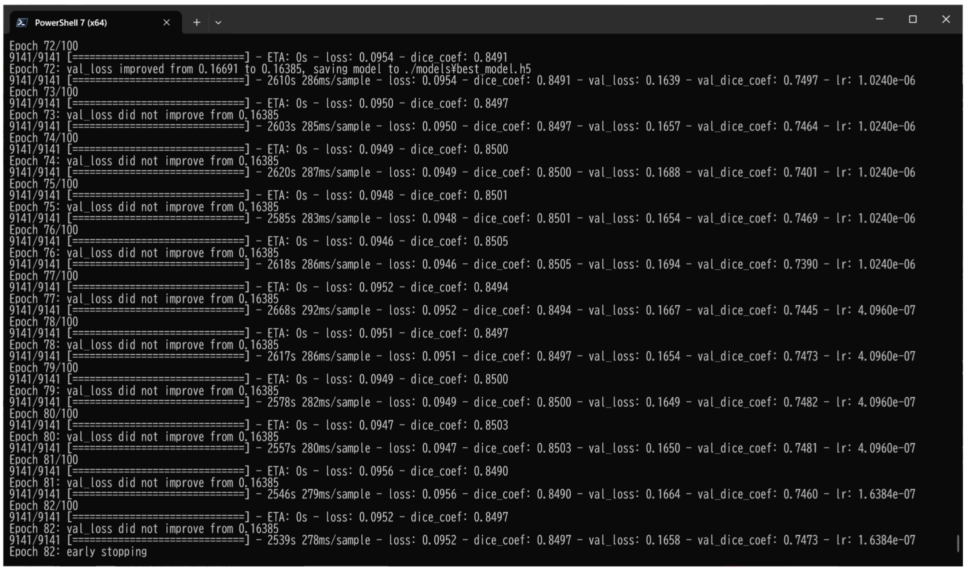
Epoch_change_inference output at starting (1,2,3,4)



Epoch_change_inference output at ending (79,80,81,82)



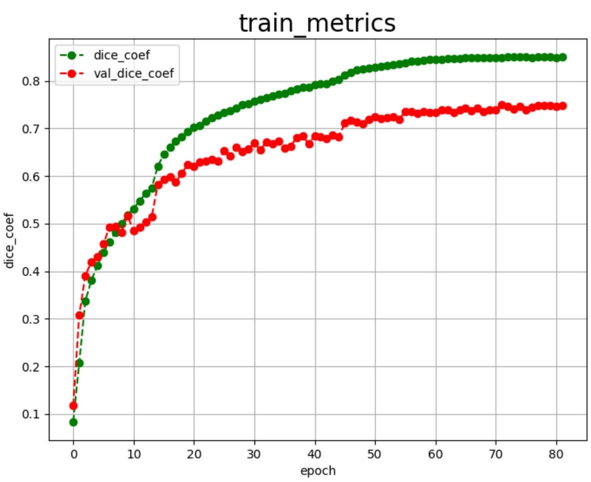
In this experiment, the training process was stopped at epoch 82 by EarlyStopping Callback.



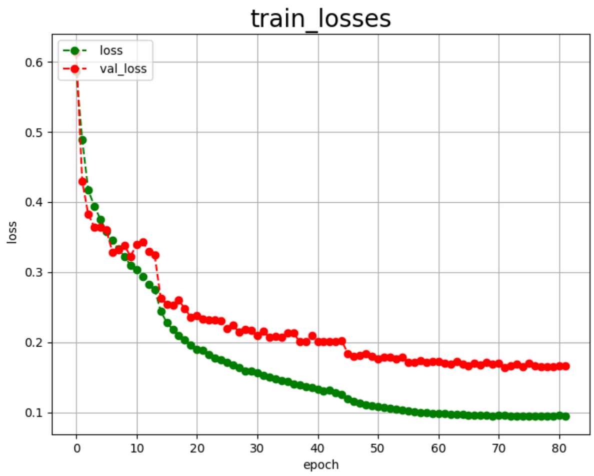
PowerShell 7 (x64)

```
Epoch 72/100
9141/9141 [==============================] - ETA: 0s - loss: 0.0954 - dice_coef: 0.8491
Epoch 72: val_loss improved from 0.16691 to 0.16385, saving model to ./models¥best_model.h5
9141/9141 [==============================] - 2610s 286ms/sample - loss: 0.0954 - dice_coef: 0.8491 - val_loss: 0.1639 - val_dice_coef: 0.7497 - lr: 1.0240e-06
Epoch 73/100
9141/9141 [==============================] - ETA: 0s - loss: 0.0950 - dice_coef: 0.8497
Epoch 73: val_loss did not improve from 0.16385
9141/9141 [==============================] - 2603s 285ms/sample - loss: 0.0950 - dice_coef: 0.8497 - val_loss: 0.1657 - val_dice_coef: 0.7464 - lr: 1.0240e-06
Epoch 74/100
9141/9141 [==============================] - ETA: 0s - loss: 0.0949 - dice_coef: 0.8500
Epoch 74: val_loss did not improve from 0.16385
9141/9141 [==============================] - 2620s 287ms/sample - loss: 0.0949 - dice_coef: 0.8500 - val_loss: 0.1688 - val_dice_coef: 0.7401 - lr: 1.0240e-06
Epoch 75/100
9141/9141 [==============================] - ETA: 0s - loss: 0.0948 - dice_coef: 0.8501
Epoch 75: val_loss did not improve from 0.16385
9141/9141 [==============================] - 2585s 283ms/sample - loss: 0.0948 - dice_coef: 0.8501 - val_loss: 0.1654 - val_dice_coef: 0.7469 - lr: 1.0240e-06
Epoch 76/100
9141/9141 [==============================] - ETA: 0s - loss: 0.0946 - dice_coef: 0.8505
Epoch 76: val_loss did not improve from 0.16385
9141/9141 [==============================] - 2618s 286ms/sample - loss: 0.0946 - dice_coef: 0.8505 - val_loss: 0.1694 - val_dice_coef: 0.7390 - lr: 1.0240e-06
Epoch 77/100
9141/9141 [==============================] - ETA: 0s - loss: 0.0952 - dice_coef: 0.8494
Epoch 77: val_loss did not improve from 0.16385
9141/9141 [==============================] - 2668s 292ms/sample - loss: 0.0952 - dice_coef: 0.8494 - val_loss: 0.1667 - val_dice_coef: 0.7445 - lr: 4.0960e-07
Epoch 78/100
9141/9141 [==============================] - ETA: 0s - loss: 0.0951 - dice_coef: 0.8497
Epoch 78: val_loss did not improve from 0.16385
9141/9141 [==============================] - 2617s 286ms/sample - loss: 0.0951 - dice_coef: 0.8497 - val_loss: 0.1654 - val_dice_coef: 0.7473 - lr: 4.0960e-07
Epoch 79/100
9141/9141 [==============================] - ETA: 0s - loss: 0.0949 - dice_coef: 0.8500
Epoch 79: val_loss did not improve from 0.16385
9141/9141 [==============================] - 2578s 282ms/sample - loss: 0.0949 - dice_coef: 0.8500 - val_loss: 0.1649 - val_dice_coef: 0.7482 - lr: 4.0960e-07
Epoch 80/100
9141/9141 [==============================] - ETA: 0s - loss: 0.0947 - dice_coef: 0.8503
Epoch 80: val_loss did not improve from 0.16385
9141/9141 [==============================] - 2557s 280ms/sample - loss: 0.0947 - dice_coef: 0.8503 - val_loss: 0.1650 - val_dice_coef: 0.7481 - lr: 4.0960e-07
Epoch 81/100
9141/9141 [==============================] - ETA: 0s - loss: 0.0956 - dice_coef: 0.8490
Epoch 81: val_loss did not improve from 0.16385
9141/9141 [==============================] - 2546s 279ms/sample - loss: 0.0956 - dice_coef: 0.8490 - val_loss: 0.1664 - val_dice_coef: 0.7460 - lr: 1.6384e-07
Epoch 82/100
9141/9141 [==============================] - ETA: 0s - loss: 0.0952 - dice_coef: 0.8497
Epoch 82: val_loss did not improve from 0.16385
9141/9141 [==============================] - 2539s 278ms/sample - loss: 0.0952 - dice_coef: 0.8497 - val_loss: 0.1658 - val_dice_coef: 0.7473 - lr: 1.6384e-07
Epoch 82: early stopping
```

train_metrics.csv



train_losses.csv



4 Evaluation

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

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

This bat file simply runs the following command.

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

Evaluation console output:

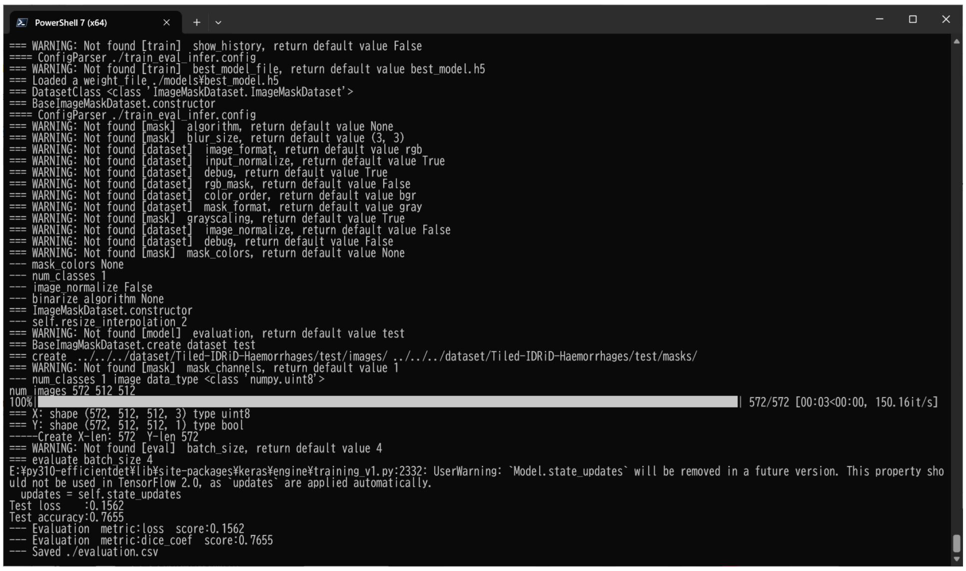


Image-Segmentation-IDRiD-Haemorrhages evaluation.csv
The loss (bce_dice_loss) to this IDRiD-Haemorrhages/test was not low, and dice_coef not high as shown below.

```
loss,0.1562
dice_coef,0.7655
```

5 Tiled inference

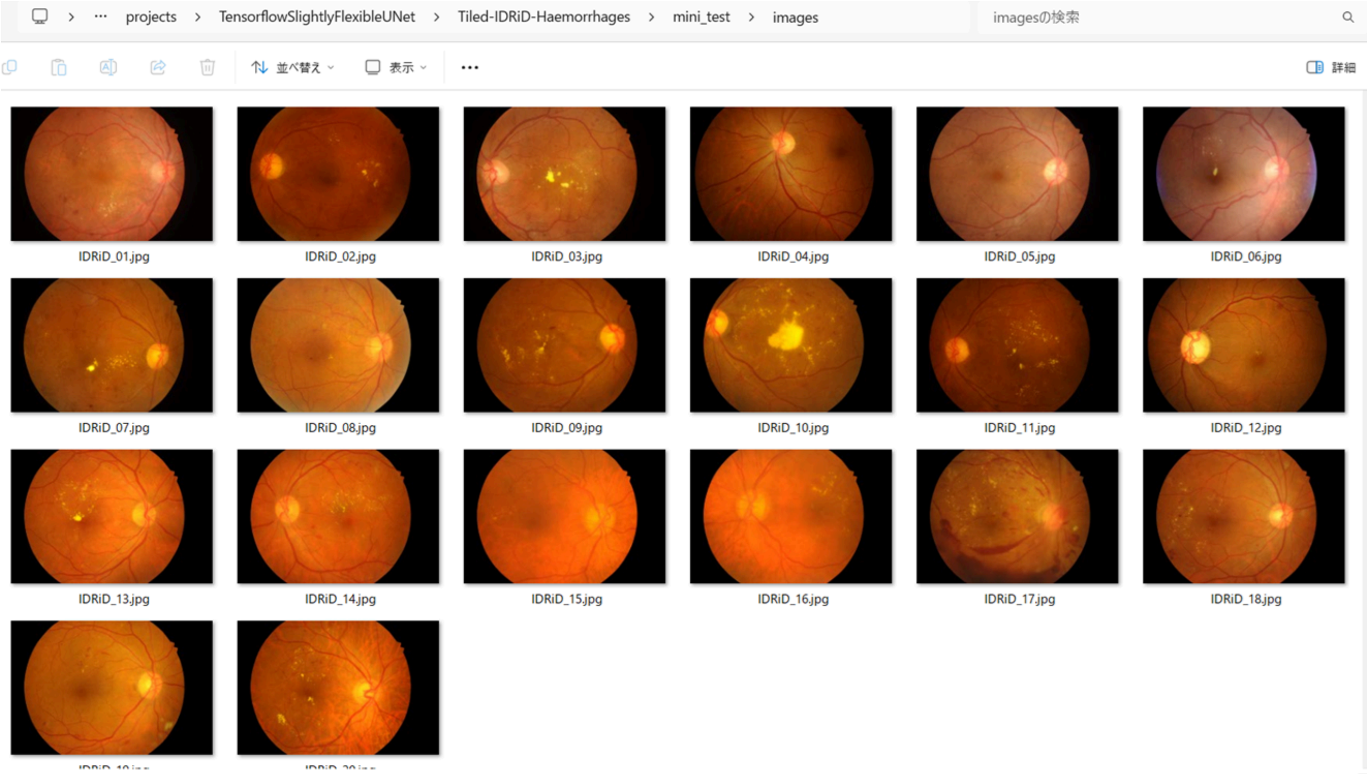
Please move to a **./projects/TensorflowSlightlyFlexibleUNet/IDRiD-Haemorrhages** folder
,and run the following bat file to infer segmentation regions for images by the Trained-TensorflowUNet model for IDRiD-Haemorrhages.

```
./4.tiled_infer.bat
```

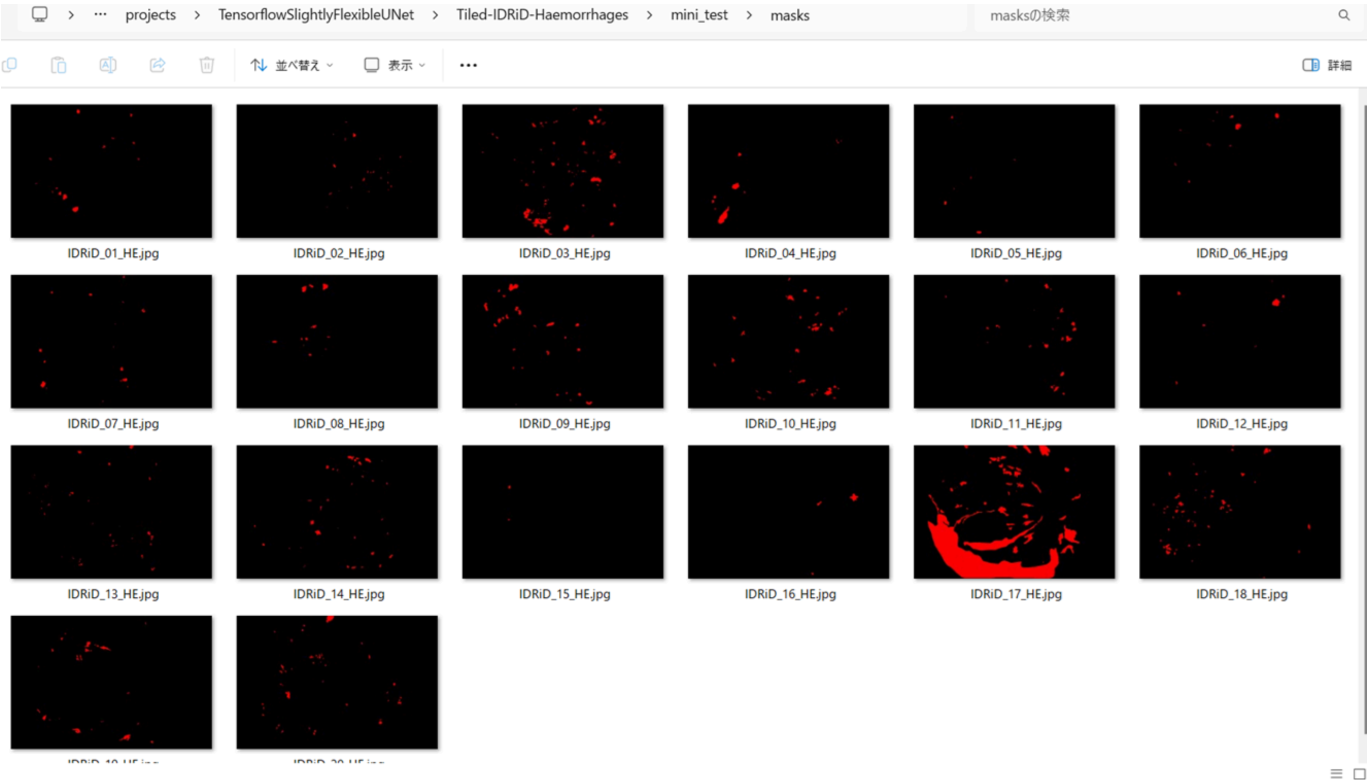
This simply runs the following command.

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

mini_test_images (4288x2848 pixels)



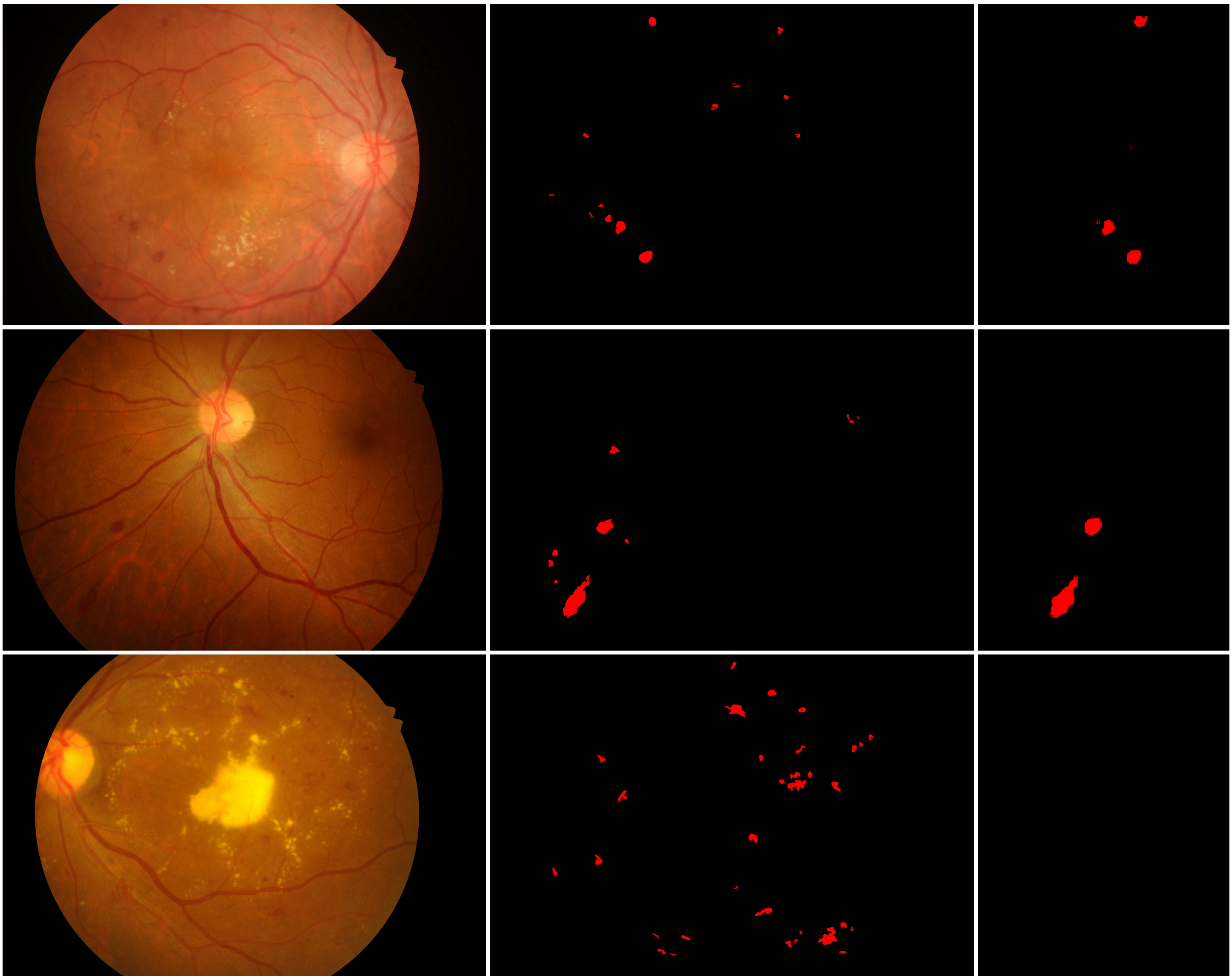
mini_test_mask(ground_truth)

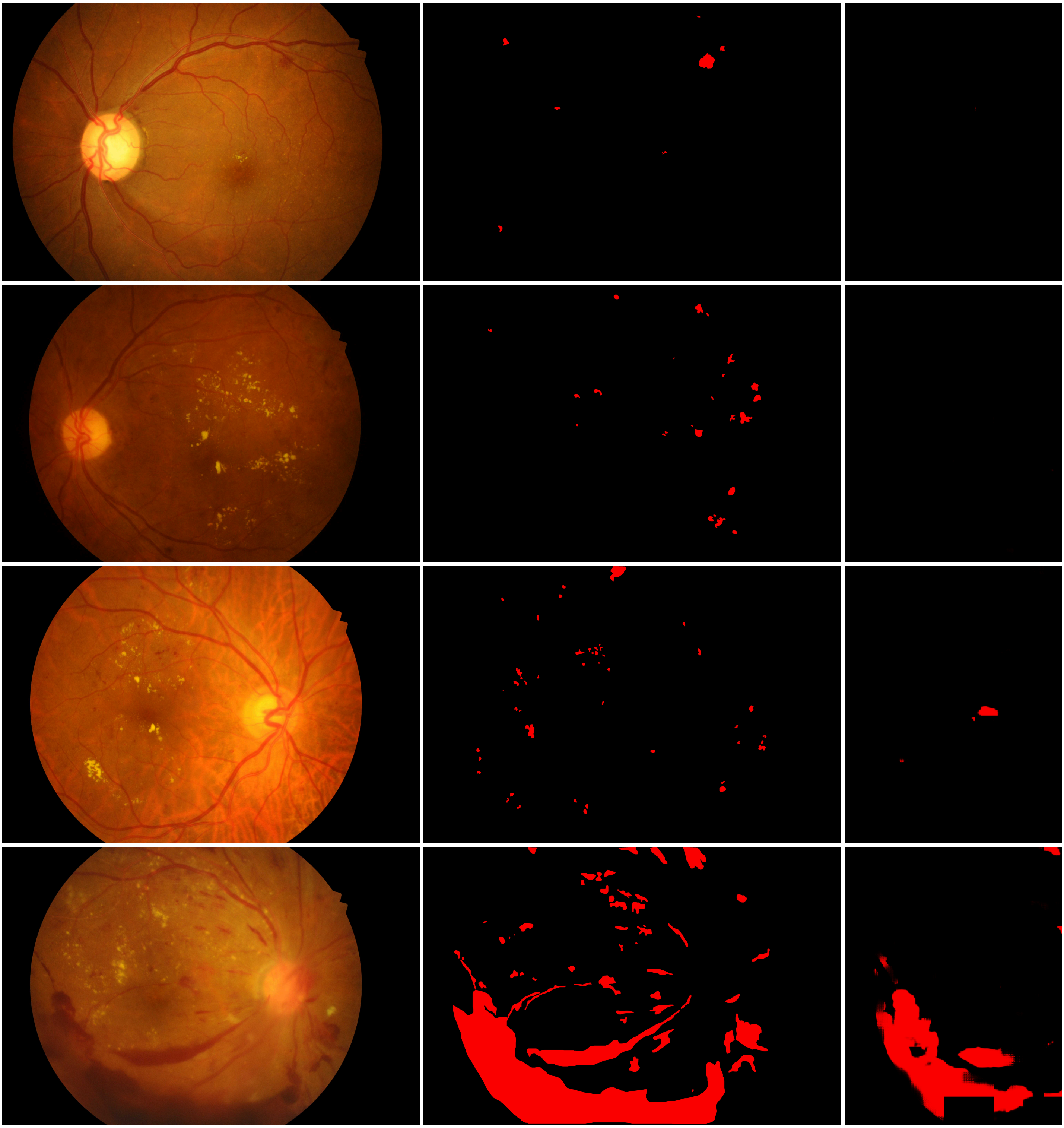


Tiled inferred test masks (4288x2848 pixels)

Enlarged images and masks of 4288x2848 pixels

Image | Mask (ground_truth) | Tiled-inferr





References

1. IDRiD: Diabetic Retinopathy – Segmentation and Grading Challenge
Prasanna Porwal , Samiksha Pachade, Manesh Kokare, Girish Deshmukh, Jaemin Son, Woong Bae, Lihong Liu
, Jianzong Wang, Xinhui Liu, Liangxin Gao, TianBo Wu, Jing Xiao, Fengyan Wang
, Baocai Yin, Yunzhi Wang, Gopichandh Danala, Linsheng He, Yoon Ho Choi, Yeong Chan Lee
, Sang-Hyuk Jung,Fabrice Mériaudeau

DOI:https://doi.org/10.1016/j.media.2019.101561
https://www.sciencedirect.com/science/article/abs/pii/S1361841519301033

2. Tensorflow-Tiled-Image-Segmentation-Pre-Augmented-IDRiD-HardExudates
Toshiyuki Arai @antillia.com
https://github.com/sarah-antillia/Tensorflow-Tiled-Image-Segmentation-Pre-Augmented-IDRiD-HardExudates