

Tensorflow-Tiled-Image-Segmentation-Pre-Augmented-IDRiD-OpticDisc (2025/03/19)

Sarah T. Arai
Software Laboratory antillia.com

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

Please see also [Tensorflow-Tiled-Image-Segmentation-Pre-Augmented-IDRiD-HardExudates](#)

Experiment Strategies

In this experiment, we employed the following strategies.

1. Create Tiled ImageMask Dataset

We created Tiled-IDRiD-OpticDisc-ImageMask-Dataset, which was tiledly-split 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-OpticDisc train and valid datasets.

3. Evaluate UNet Model performance

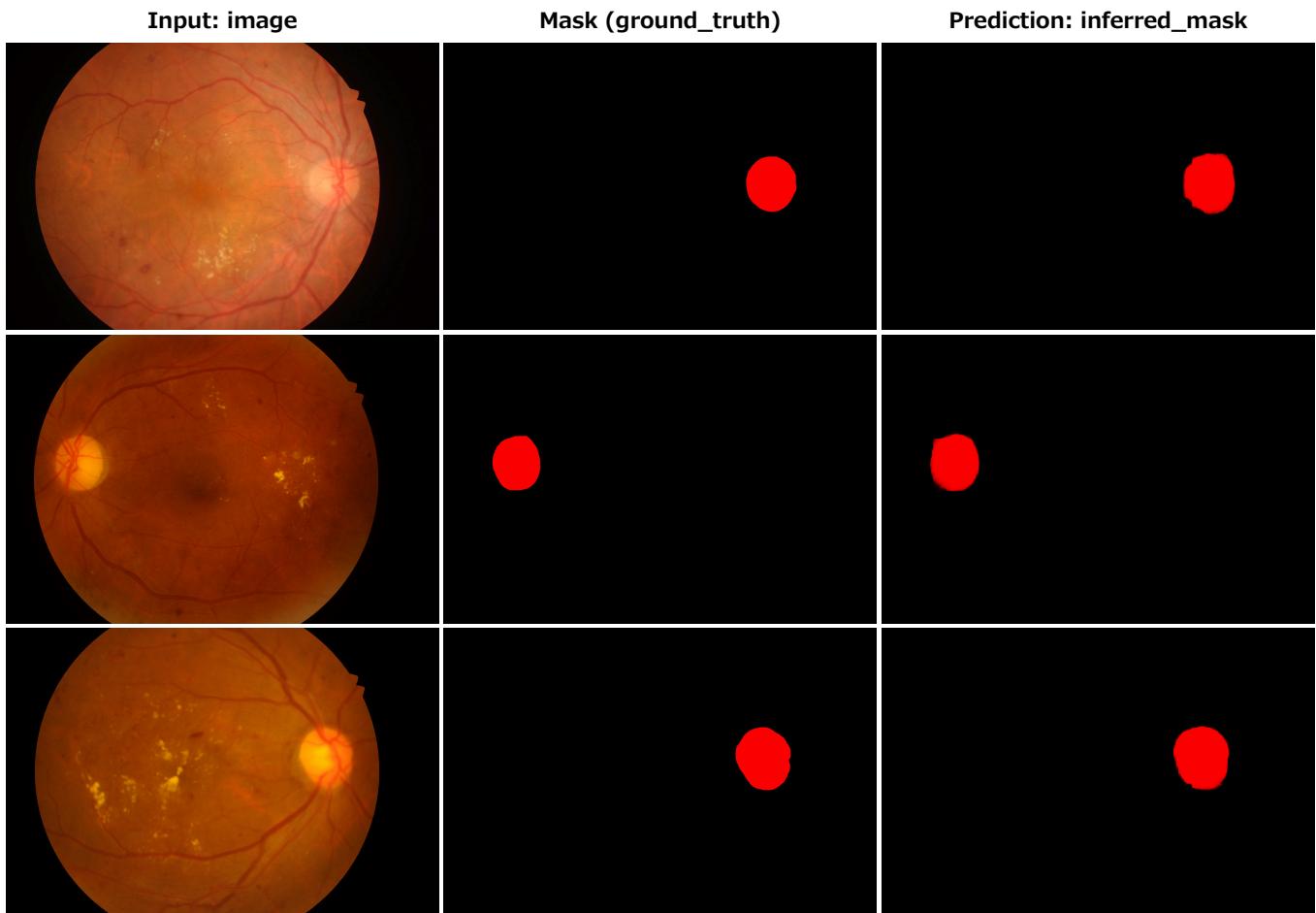
We evaluated a TensorFlow UNet model using the Tiled-IDRiD-OpticDisc test dataset

4. Apply Tiled Image Segmentation

We applied our Tiled-Image Segmentation inference method to predict the Optic Disc 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 inferred masks look quite similar to the ground truth masks.



In this experiment, we used the simple UNet Model [TensorflowSlightlyFlexibleUNet](#) for this IDRiD-OpticDiscSegmentation 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: SEGMENTATION 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,

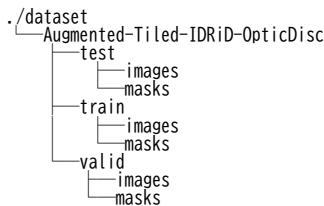
DOI: <https://dx.doi.org/10.21227/H25W98>

License:

[Creative Commons Attribution 4.0 International License.](#)

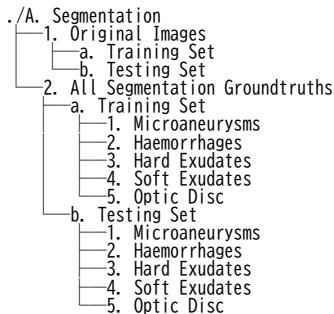
2 Augmented-Tiled-IDRiD-OpticDisc ImageMask Dataset

If you would like to train this IDRiD-OpticDisc Segmentation model by yourself, please download the pre-augmented dataset from the google drive [Augmented-Tiled-IDRiD-OpticDisc-ImageMask-Dataset.zip](#), expand the downloaded ImageMaskDataset and put it under **./dataset** folder to be



This is a 512x512 pixels pre augmented tiles dataset generated from 4288x2848 pixels 54 **Original Images** and their corresponding **Optic Disc GroundTruths** in Training Set.

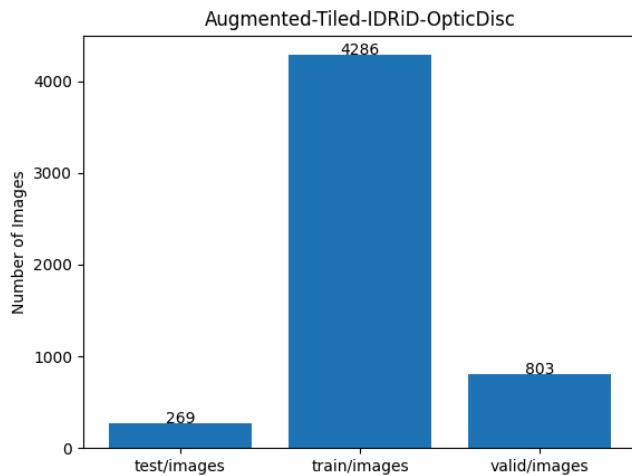
We excluded all black (empty) masks and their corresponding images to generate our dataset from the original one. The folder structure of the original Segmentation data is the following.



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

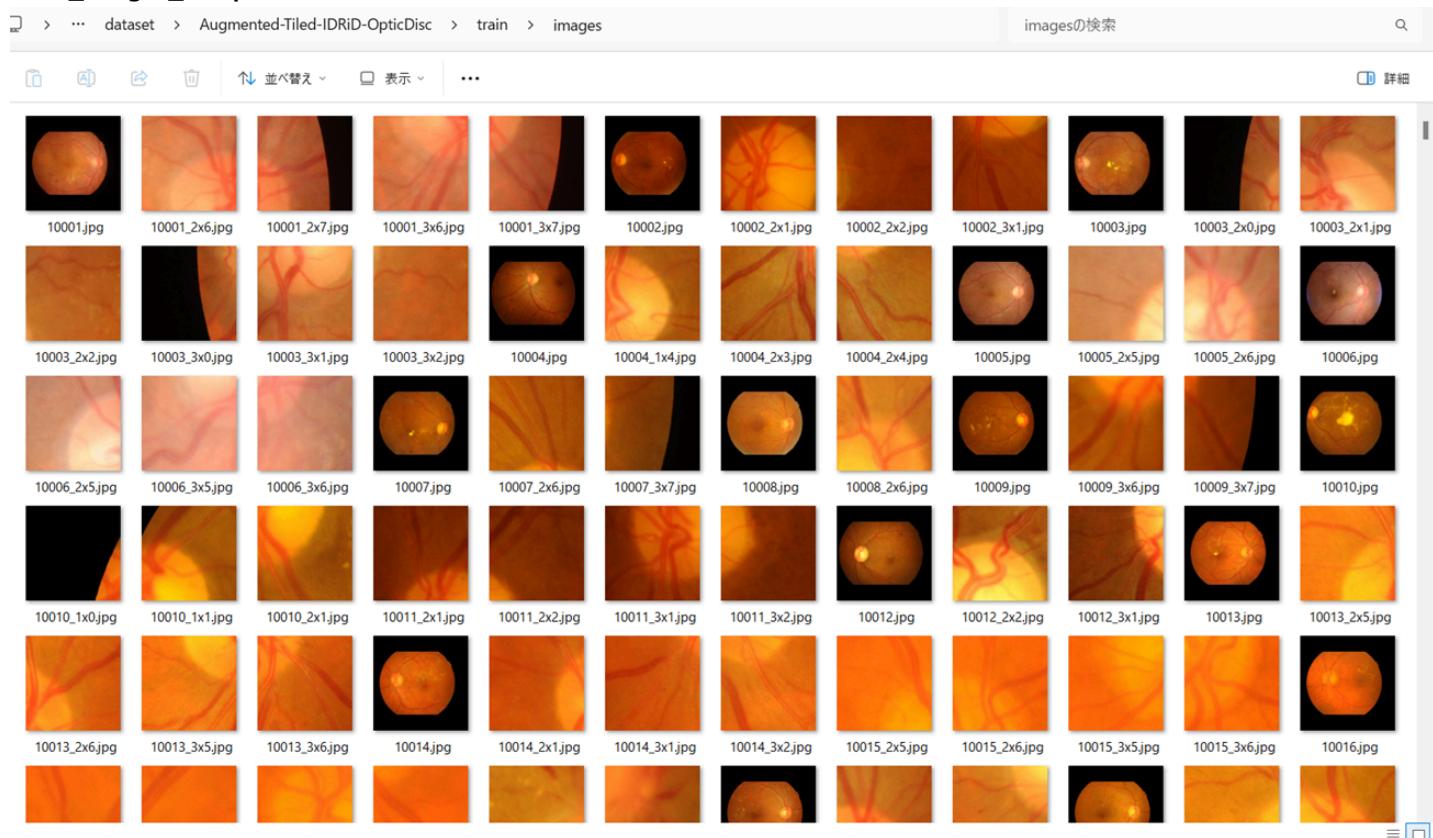
- [TiledImageMaskDatasetGenerator.py](#)
- [split_tiled_master.py](#)

Augmented-Tiled-IDRiD-OpticDisc Statistics

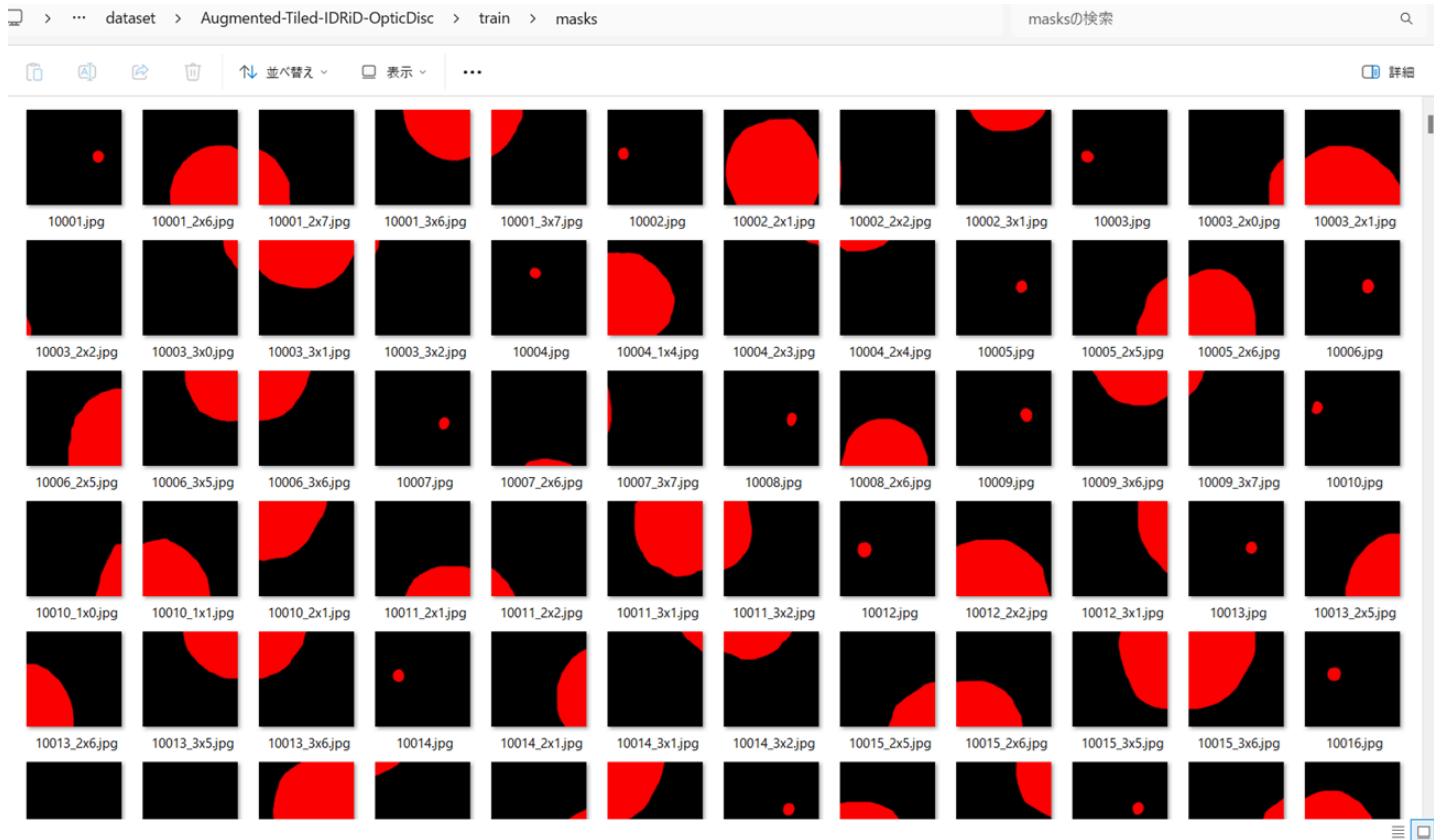


As shown above, the number of images of train and valid datasets is enough to use for a training set of our segmentation model.

Train_images_sample



Train_masks_sample



3 Train TensorflowUNet Model

We have trained IDRiD-OpticDisc TensorflowUNet Model by using the following [train_eval_infer.config](#) file.
Please move to ./projects/TensorflowSlightlyFlexibleUNet/IDRiD-OpticDisc 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=(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 = 0.00007
```

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_CUBIC"
```

Early stopping callback

Enabled early stopping callback with patience parameter.

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

Epoch change inference callbacks

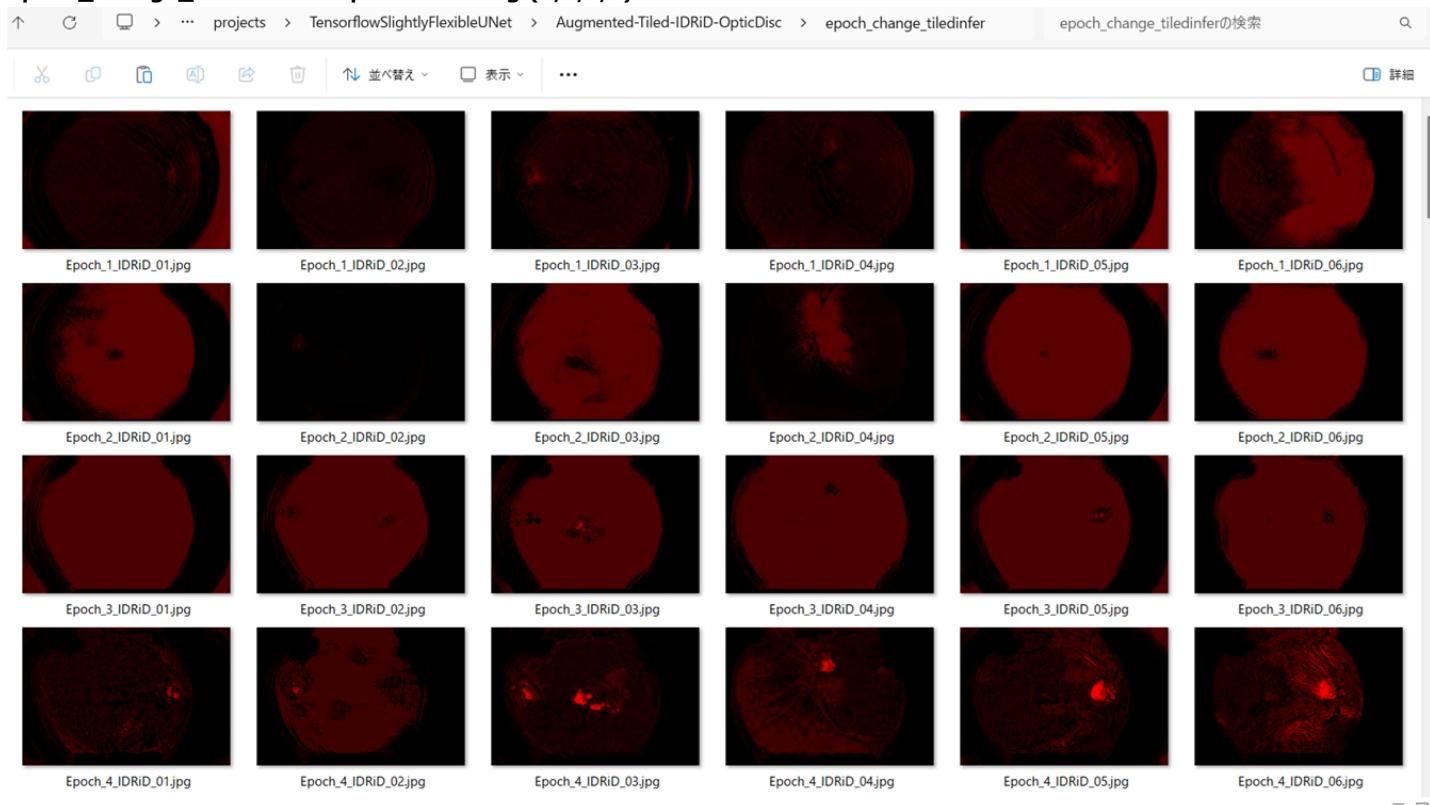
Enabled epoch_change_infer callback.

```
[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 (49,50,51,52)



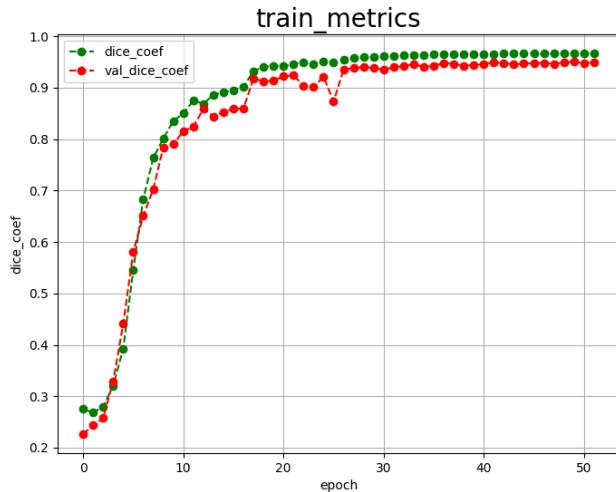
In this experiment, the training process was stopped at epoch 52 by EarlyStopping Callback.

```

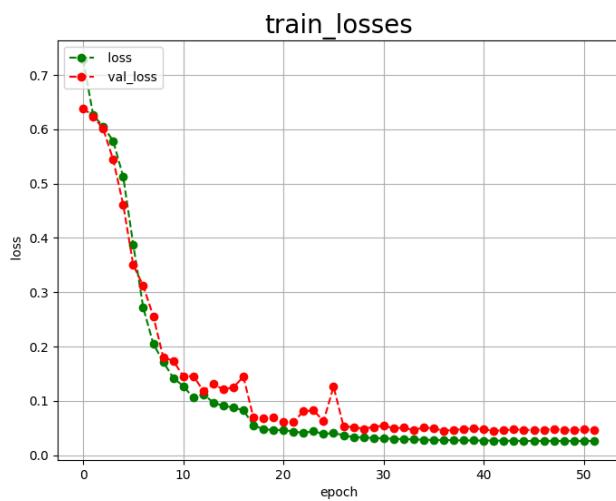
PowerShell 7 (x64) + -
4286/4286 [=====] - ETA: 0s - loss: 0.0268 - dice_coef: 0.9657
Epoch 42: val_loss improved from 0.04497 to 0.04473, saving model to ./models/best_model.h5
4286/4286 [=====] - 1073s 250ms/sample - loss: 0.0268 - dice_coef: 0.9657 - val_loss: 0.0447 - val_dice_coef: 0.9492 - lr: 1.7920e-06
Epoch 43/100
4286/4286 [=====] - ETA: 0s - loss: 0.0265 - dice_coef: 0.9661
Epoch 43: val_loss did not improve from 0.04473
4286/4286 [=====] - ETA: 0s - loss: 0.0265 - dice_coef: 0.9661 - val_loss: 0.0459 - val_dice_coef: 0.9473 - lr: 1.7920e-06
Epoch 44/100
4286/4286 [=====] - ETA: 0s - loss: 0.0263 - dice_coef: 0.9664
Epoch 44: val_loss did not improve from 0.04473
4286/4286 [=====] - 1069s 249ms/sample - loss: 0.0263 - dice_coef: 0.9664 - val_loss: 0.0472 - val_dice_coef: 0.9454 - lr: 1.7920e-06
Epoch 45/100
4286/4286 [=====] - ETA: 0s - loss: 0.0261 - dice_coef: 0.9666
Epoch 45: val_loss did not improve from 0.04473
4286/4286 [=====] - 1071s 250ms/sample - loss: 0.0261 - dice_coef: 0.9666 - val_loss: 0.0465 - val_dice_coef: 0.9468 - lr: 1.7920e-06
Epoch 46/100
4286/4286 [=====] - ETA: 0s - loss: 0.0261 - dice_coef: 0.9666
Epoch 46: val_loss did not improve from 0.04473
4286/4286 [=====] - 1071s 250ms/sample - loss: 0.0261 - dice_coef: 0.9666 - val_loss: 0.0462 - val_dice_coef: 0.9476 - lr: 1.7920e-06
Epoch 47/100
4286/4286 [=====] - ETA: 0s - loss: 0.0263 - dice_coef: 0.9663
Epoch 47: val_loss did not improve from 0.04473
4286/4286 [=====] - 1068s 249ms/sample - loss: 0.0263 - dice_coef: 0.9663 - val_loss: 0.0468 - val_dice_coef: 0.9481 - lr: 7.1680e-07
Epoch 48/100
4286/4286 [=====] - ETA: 0s - loss: 0.0261 - dice_coef: 0.9666
Epoch 48: val_loss did not improve from 0.04473
4286/4286 [=====] - 1055s 246ms/sample - loss: 0.0261 - dice_coef: 0.9666 - val_loss: 0.0479 - val_dice_coef: 0.9464 - lr: 7.1680e-07
Epoch 49/100
4286/4286 [=====] - ETA: 0s - loss: 0.0261 - dice_coef: 0.9666
Epoch 49: val_loss did not improve from 0.04473
4286/4286 [=====] - 1053s 246ms/sample - loss: 0.0261 - dice_coef: 0.9666 - val_loss: 0.0466 - val_dice_coef: 0.9488 - lr: 7.1680e-07
Epoch 50/100
4286/4286 [=====] - ETA: 0s - loss: 0.0260 - dice_coef: 0.9667
Epoch 50: val_loss did not improve from 0.04473
4286/4286 [=====] - 1092s 255ms/sample - loss: 0.0260 - dice_coef: 0.9667 - val_loss: 0.0460 - val_dice_coef: 0.9507 - lr: 7.1680e-07
Epoch 51/100
4286/4286 [=====] - ETA: 0s - loss: 0.0262 - dice_coef: 0.9665
Epoch 51: val_loss did not improve from 0.04473
4286/4286 [=====] - 1098s 256ms/sample - loss: 0.0262 - dice_coef: 0.9665 - val_loss: 0.0472 - val_dice_coef: 0.9473 - lr: 2.8672e-07
Epoch 52/100
4286/4286 [=====] - ETA: 0s - loss: 0.0260 - dice_coef: 0.9667
Epoch 52: val_loss did not improve from 0.04473
4286/4286 [=====] - 1087s 254ms/sample - loss: 0.0260 - dice_coef: 0.9667 - val_loss: 0.0465 - val_dice_coef: 0.9488 - lr: 2.8672e-07
Epoch 52: early stopping
Save history.json

```

[train_metrics.csv](#)



[train_losses.csv](#)



4 Evaluation

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

`./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
--- ConfigParser: Not found [train] show_history, return default value False
--- ConfigParser: /train eval infer.config
--- WARNING: Not found [train] best_model_file, return default value best_model.h5
--- LOADED a weight file: ./models/best_model.h5
Dataset<class 'ImageMaskDataset'>
BaseImageMaskDataset.constructor
--- ConfigParser: /train eval infer.config
--- WARNING: Not found [mask] algorithm, return default value None
--- WARNING: Not found [mask] blur_size, return default value (3, 3)
--- 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] crop, return default value False
--- WARNING: Not found [dataset] color_order, return default value bgr
--- WARNING: Not found [dataset] mask_format, return default value gray
--- 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
binarize_algorithm None
ImageMaskDataset.constructor
self._resize_interpolation_2
self._img_size 256
self._batch_size 1
self._eval_size 1
self._eval_interpolation_2
self._eval_batch_size 1
BaseImageMaskDataset.create_dataset_test
create ./dataset/Augmented-Tiled-IDRiD-OpticDisc/test/images/ ../../dataset/Augmented-Tiled-IDRiD-OpticDisc/test/masks/
--- WARNING: Not found [mask] mask_channels, return default value 1
num_classes 1 image data type <class 'numpy.uint8'>
num_images 269 512 512
100% | 269/269 [00:02<00:00, 119.88it/s]
X: shape (269, 512, 512, 3) type uint8
Y: shape (269, 512, 512, 1) type bool
--- Create X:[len: 269 Y:[len: 269
--- WARNING: Not found [eval] batch_size, return default value 4
--- evaluate batch_size 4
E:\py36\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.
    _updated = self._state_updates
Test loss: 0.0391
Test accuracy: 0.9644
Evaluation metric:bce_loss score:0.0391
Evaluation metric:dice_coef score:0.9644
Saved ./evaluation.csv
```

Image-Segmentation-IDRiD-OpticDisc [evaluation.csv](#)

The loss (bce_dice_loss) to this IDRiD-OpticDisc/test was low, and dice_coef high as shown below.

```
loss, 0.0391
dice_coef, 0.9644
```

5 Tiled inference

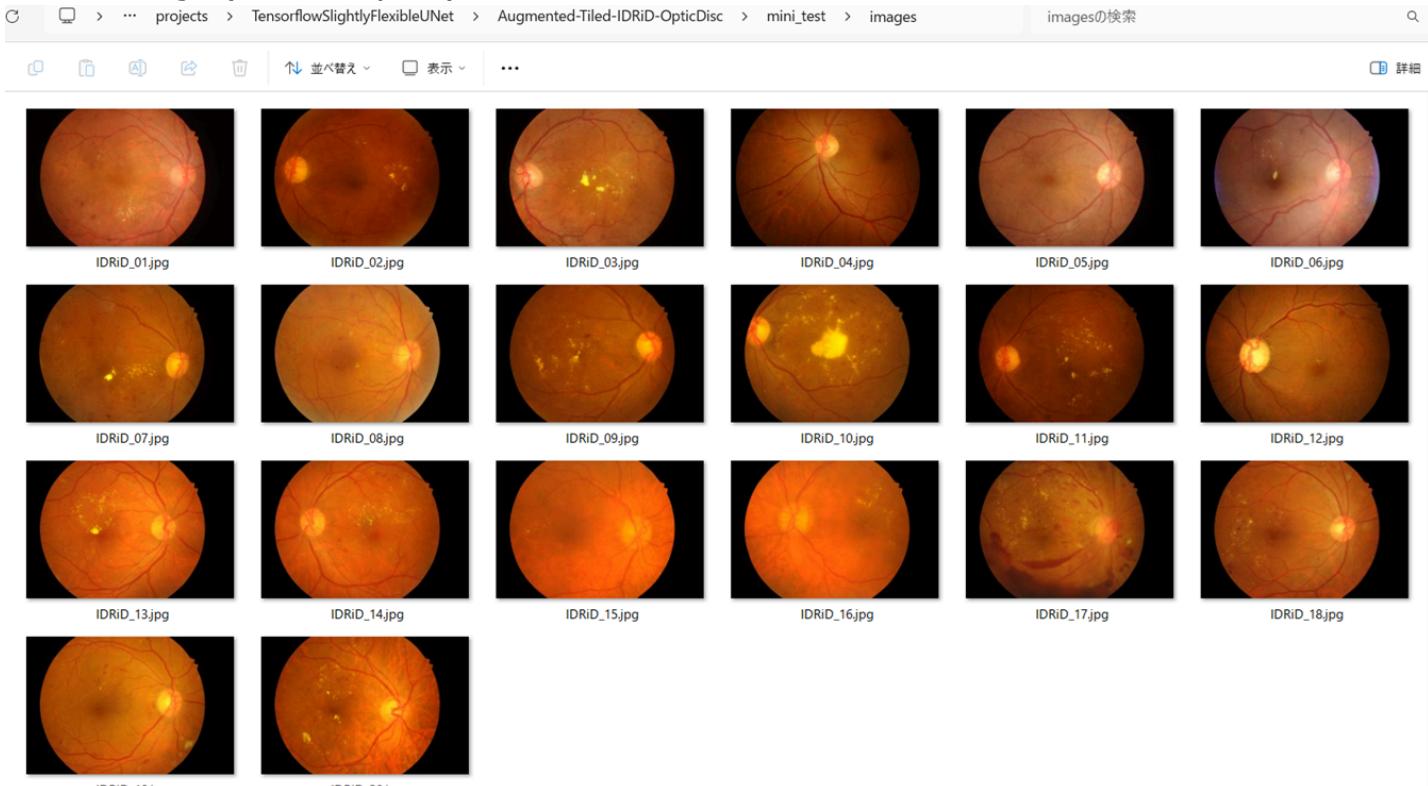
Please move to a [./projects/TensorflowSlightlyFlexibleUNet/IDRiD-OpticDisc](#) folder ,and run the following bat file to infer segmentation regions for images by the Trained-TensorflowUNet model for IDRiD-OpticDisc.

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

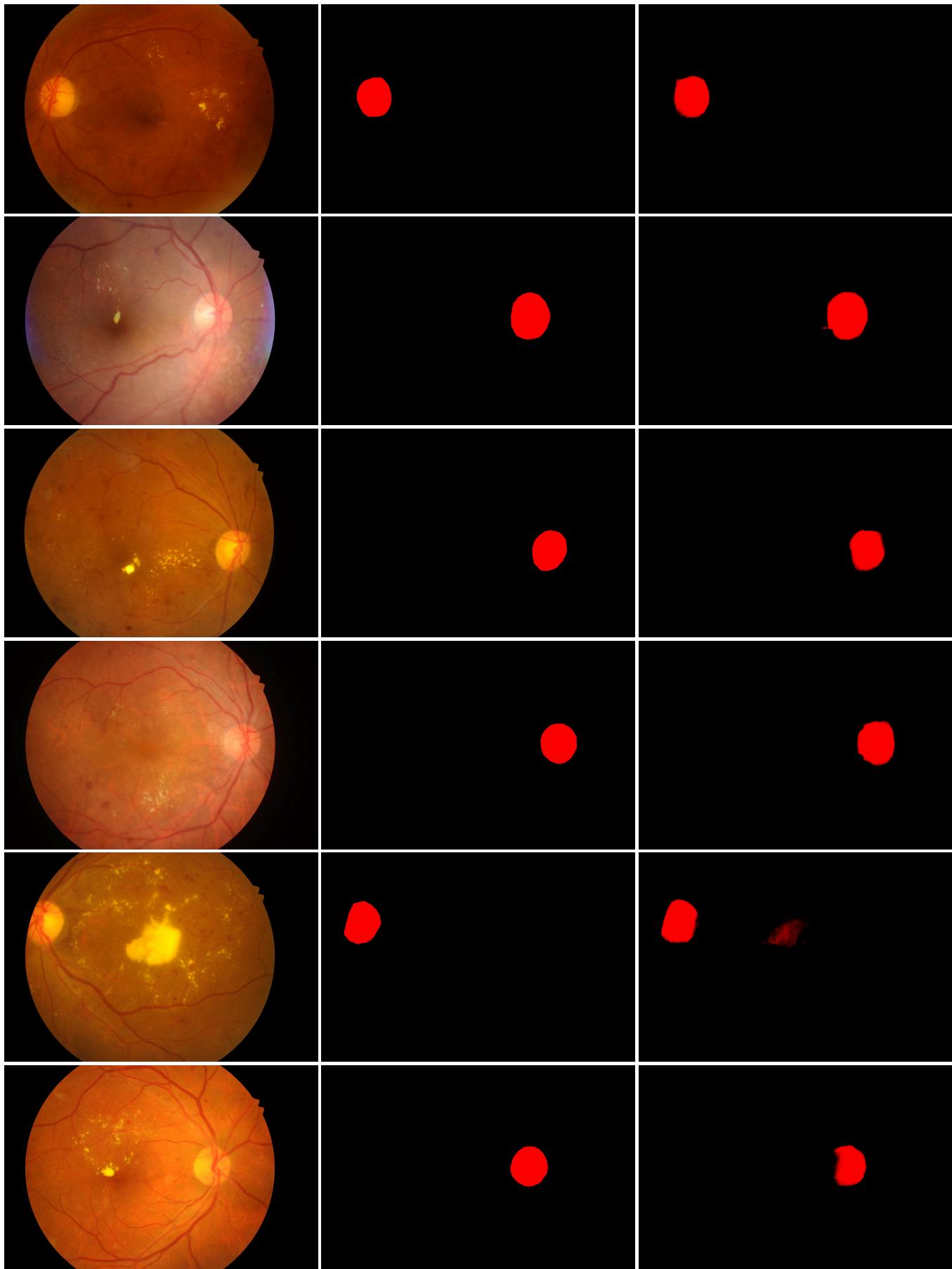
This simply runs the following command.

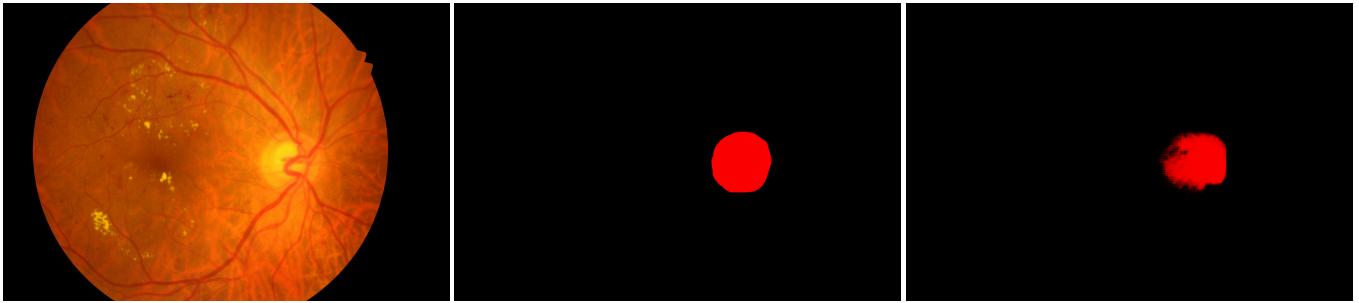
```
python ../../src/TensorflowUNetTiledInferencer.py ./train_eval_infer.config
```

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-inferred-mask**





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, Sang-Hyuk Jung, Fabrice Mériadeau

DOI: <https://doi.org/10.1016/j.media.2019.101561>

<https://www.sciencedirect.com/science/article/abs/pii/S1361841519301033>

2. RMCA U-net: Hard exudates segmentation for retinal fundus images

Yinghua Fu, Ge Zhang, Xin Lu, Honghan Wu, Dawei Zhang

<https://doi.org/10.1016/j.eswa.2023.120987>

<https://www.sciencedirect.com/science/article/abs/pii/S0957417423014896>

3. Tensorflow-Tiled-Image-Segmentation-IDRiD-HardExudates

Toshiyuki Arai @antillia.com

<https://github.com/sarah-antillia/Tensorflow-Tiled-Image-Segmentation-Pre-Augmented-IDRiD-HardExudates>

4. Tensorflow-Tiled-Image-Segmentation-Augmented-Skin-Cancer

Toshiyuki Arai @antillia.com

<https://github.com/sarah-antillia/Tensorflow-Tiled-Image-Segmentation-Augmented-Skin-Cancer>

5. Tensorflow-Tiled-Image-Segmentation-Augmented-MultipleMyeloma

Toshiyuki Arai @antillia.com

<https://github.com/sarah-antillia/Tensorflow-Tiled-Image-Segmentation-Augmented-MultipleMyeloma>

6. Tiled-ImageMask-Dataset-Breast-Cancer

Toshiyuki Arai @antillia.com

<https://github.com/sarah-antillia/Tiled-ImageMask-Dataset-Breast-Cancer>