HOW TO RE-IMPLEMENT THE PROJECT

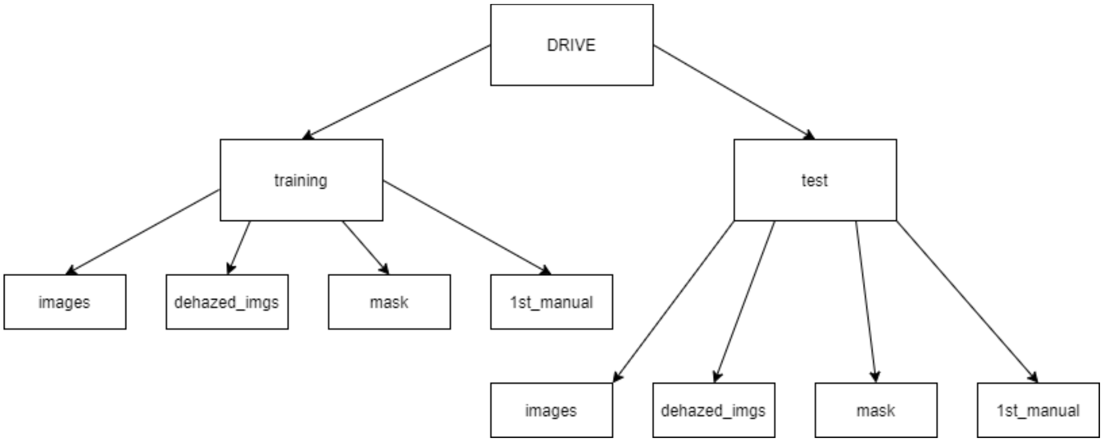
There are three main parts of this project:

1. Implementing the illumination correction to the DRIVE dataset
2. Applying the rest of the pre-processing methods and storing the images in hdf5 files
3. Applying segmentation to the images and getting the results

Implementing Illumination correction:

- Go to DRIVE folder (it contains the original DRIVE dataset and codes for dehazing)
- The training folder contains images (original fundus images), FOV mask (mask) and retinal vessel ground truth (1st_manual)
- Run the DCP_dehaze.py script (the code for applying the Dark Channel Prior was taken from here)
- The illumination corrected images are stored in ./training/dehazed_imgs folder

Once we have the dehazed_imgs folder, we divide the given training set into two halves such that 10 samples are used for training and 10 for testing. So the new dataset has the following structure :



We name this dataset as DRIVE for easy understanding.

Copies of this dataset are stored in each of the different folders corresponding to the different networks: Unet, DUNet and LadderNet.

The rest of the steps can be followed from their respective repositories :

- U-Net : https://github.com/orobix/retina-unet
- LadderNet : https://github.com/juntang-zhuang/LadderNet
- DU-Net : https://github.com/RanSuLab/DUNet-retinal-vessel-detection

For details on how to set up the environment and which packages to install please refer to the above links.

Rest of the pre-processing methods:

Executing the prepare_datasets_DRIVE.py (for Unet and LadderNet) or the prepare_dataset.py (for DUNet) applies the pre-processing methods of RGB to grayscale conversion, intensity normalization, CLAHE and Gamma adjustment to the original image set as well as the illumination corrected image set and stores them in hdf5 format in the DRIVE_datasets_training_testing folder.

U-Net segmentation

Training : while training with the original images, just run run_training.py, while training with corrected images, perform the changes to ./src/retinaNN_training.py mentioned within the script in comments and then run run_training.py

Testing : while testing with the original images, just run run_testing.py, while testing with corrected images, perform the changes to ./src/retinaNN_predict.py mentioned within the script in comments and then run run_testing.py

All results will be stored in a ‘test’ folder.

LadderNet segmentation

Training : Go to the src folder. While training with the original images, just run retinaNN_training.py, while training with corrected images, perform the changes mentioned within the script in comments and then run it

Testing : while testing with the original images, just run run_testing.py from the src folder, while testing with corrected images, perform the changes mentioned inline and then run it.

All results will be stored in the ‘test’ folder.

DU-Net segmentation

Training : while training with the original images, just run pytorch_train.py, while training with corrected images, perform the changes mentioned within the script in comments and then run it.

Testing : while testing with the original images, just run pytorch_predict_fcn.py followed by evaluation.py, while testing with corrected images, perform the changes mentioned within the scripts in comments and then run them in the same order.

All results will be stored in the ./log/experiments folder.