CMPT 733 Assignment 1 Report

Sihui Wang 301474102

1. Major Results

1.1 Configurations and Technical Details:

Hardware and Platform for Training: The Models are trained on CoLab with GPUs.

Size of the images as input: 572×572

Data augmentation methods:

Random Horizontal flip

This is implemented by `torchvision.transforms.RandomHorizontalFlip`. The function will horizontally flip the image and label with 66.7% probability.

Random Vertical flip

This is implemented by `torchvision.transforms.RandomVerticalFlip`. The function will vertically flip the image and label with 66.7% probability.

Zoom

`torchvision.transforms.Resize` and `torchvision.transforms.CenterCrop`. are required to implement *Zoom*. First, the new size is randomly chosen between 1/2 of the original size and 2 times of the original size. Then the image is resized to the new size. Finally, the function `CenterCrop` crops the image if it's larger than desired, and pads the image if it's smaller than desired, which guarantees that the processed image can fit in the network.

Rotation

This is implemented by `torchvision.transforms.RandomRotation`. The function will rotate the image by a random degree between 0° and 180°.

Gamma Correction

This is implemented by `torchvision.transforms.functional.adjust_gamma`. The function will randomly choose the parameter γ between 0.25 and 4.0, and then it will apply gamma correction to the image.

Training Epochs: 50 epochs

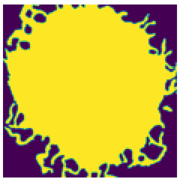
Batch size: 4

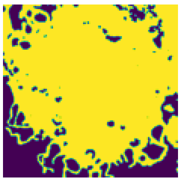
Learning rate: 0.01

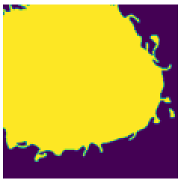
1.2 Results of Segmentation:

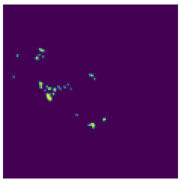
Following are the results on the testing dataset.

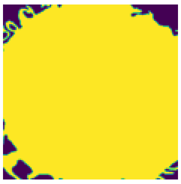
Original Label **UNet Label** **Original Label** **UNet Label**

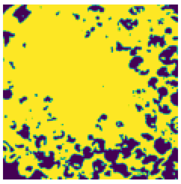


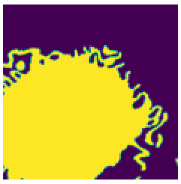


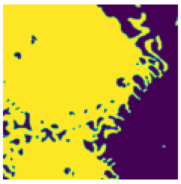




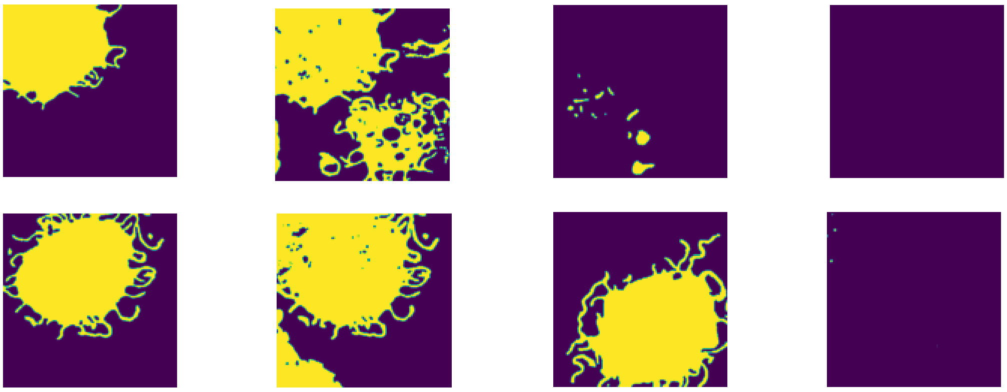






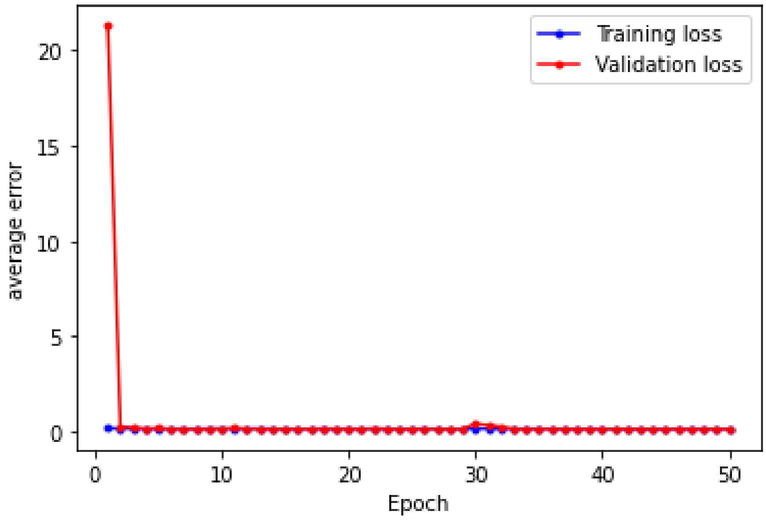


Original Label **UNet Label** **Original Label** **UNet Label**

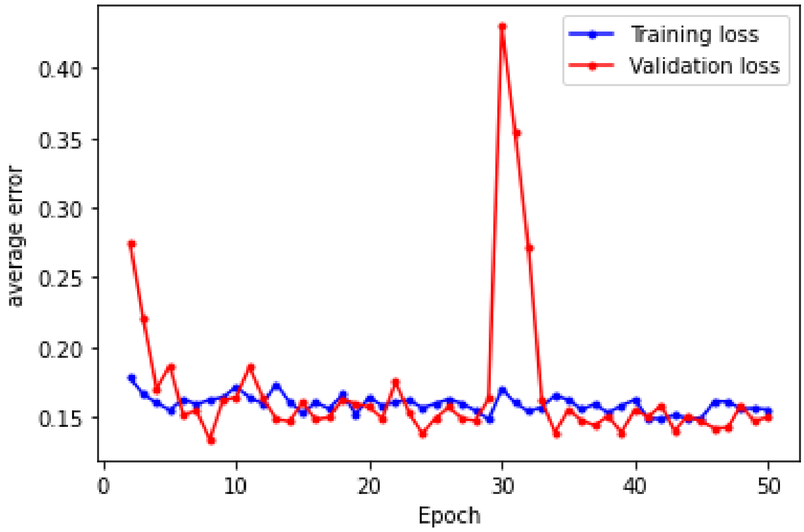


1.3 Results of Running Time, Training and Testing Losses:

The training takes **2 minutes and 48.80 seconds**. The following is the plot of *training and testing losses versus epochs*:



The testing loss for epoch 1 is sometimes extremely large. If we remove the data for epoch 1, then we will obtain the following *plot of losses for epoch 2-50*:



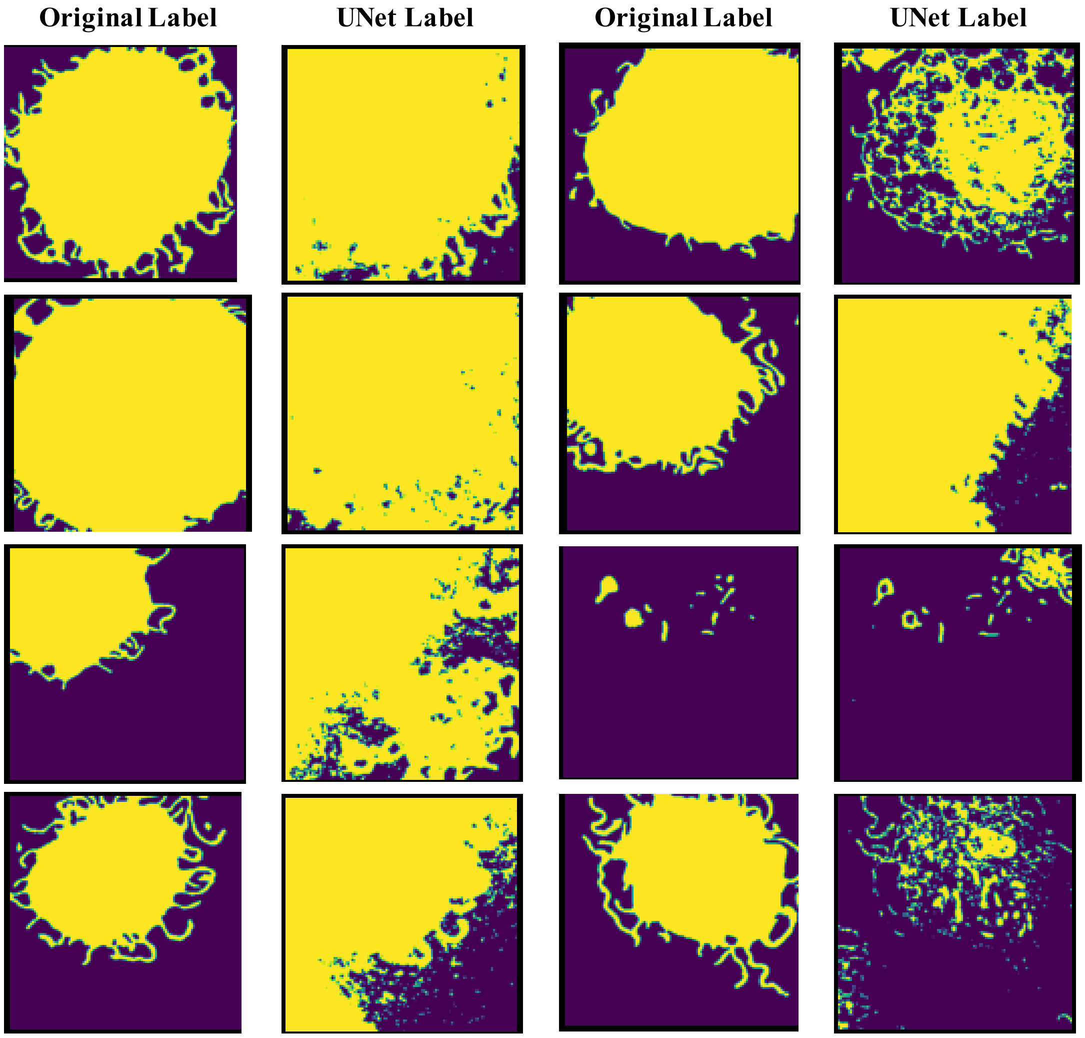
Please refer to: https://colab.research.google.com/drive/1OrMHf6vgV9qUkFfyK-rsCIcnqyQ07zgQ?usp=sharing for more details.

2. Training with different Parameters

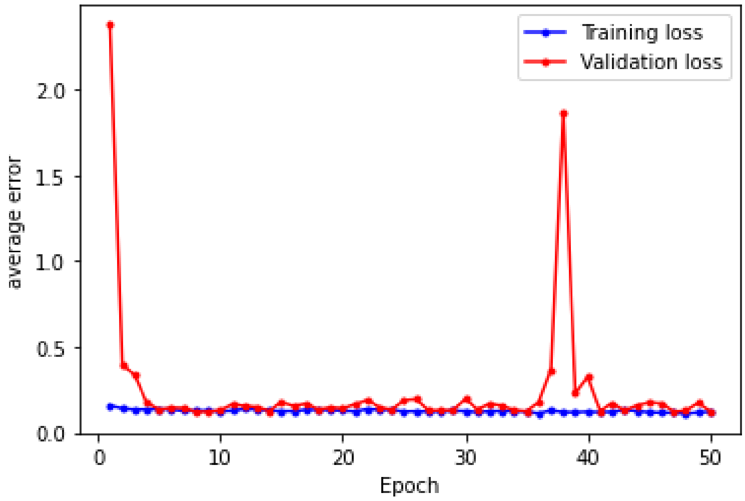
2.1 Data Augmentation

Here are the results of training without data augmentation (the only change is that there is no data augmentation, and all other parameters are identical to 1.1):

https://colab.research.google.com/drive/17rh0JIR1UoA6wgAwijWVzrRxri1l6_2Z?usp=sharing



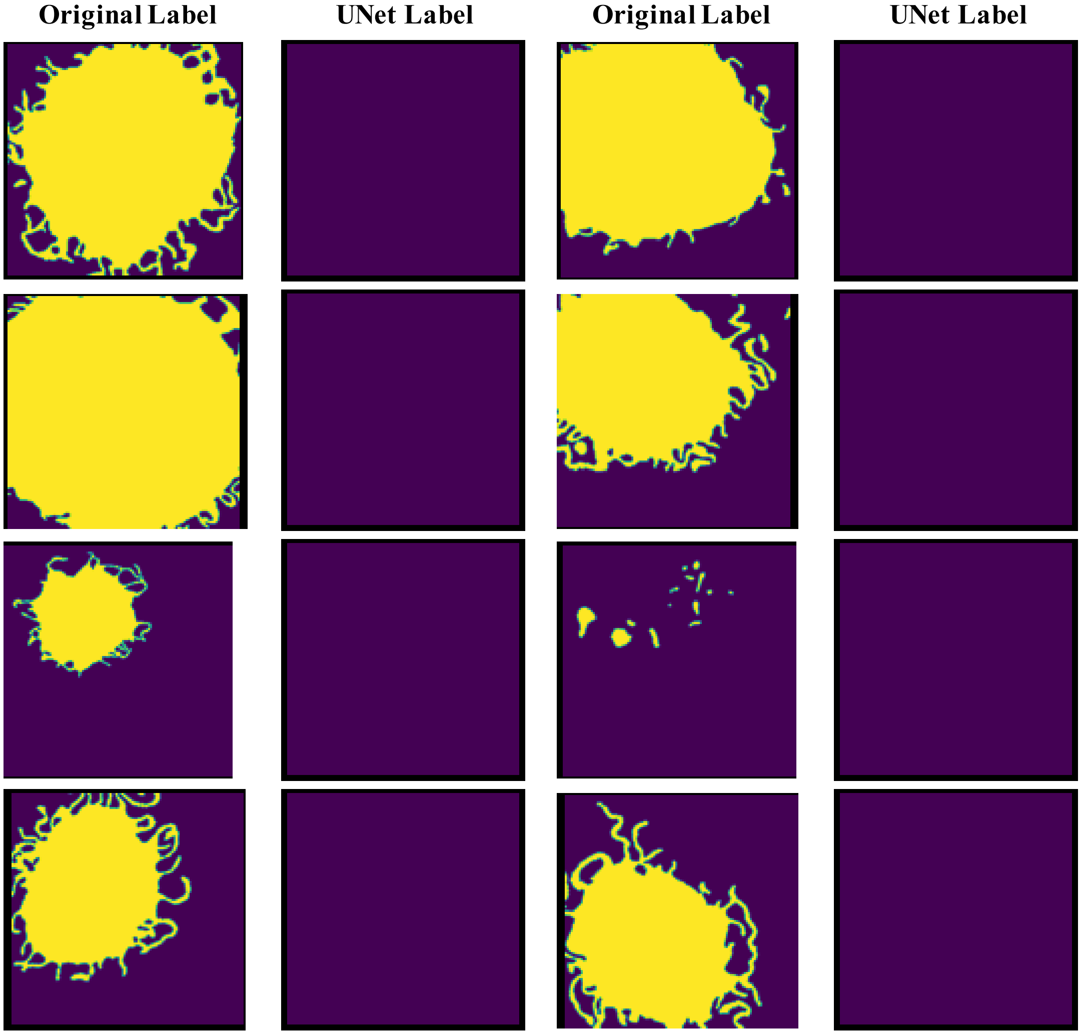
Here is the plot of *training and testing losses versus epochs*:



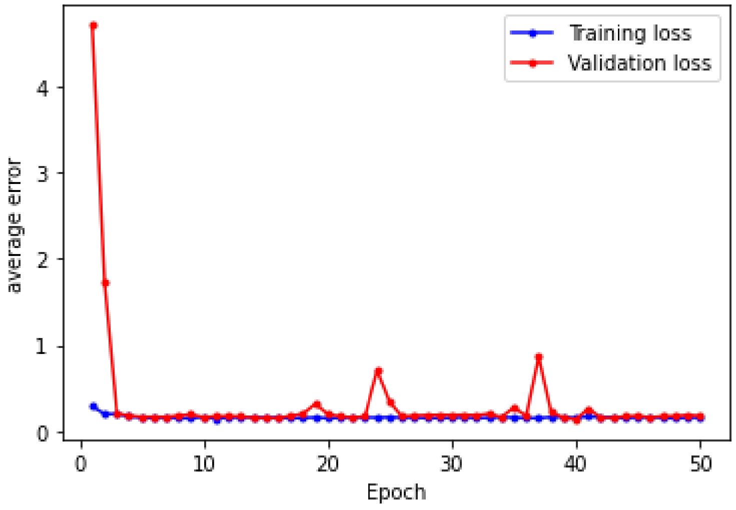
2.2 Learning Rate

(1) The results when $lr = 0.1$:

https://colab.research.google.com/drive/1CXYFQuHd_fYep--9cHP8k_uRbH_9TKW?usp=sharing

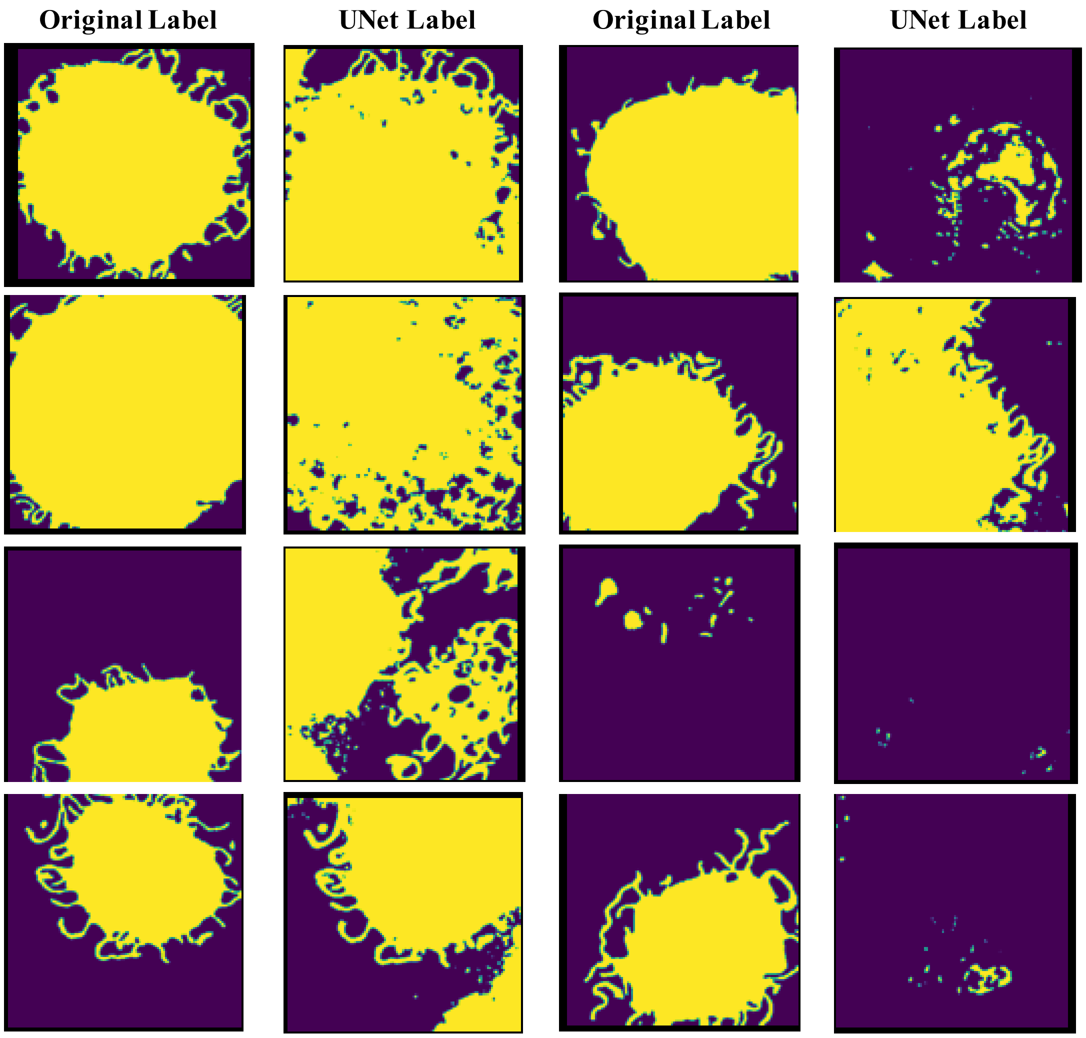


Here is the plot of *training and testing losses versus epochs*:

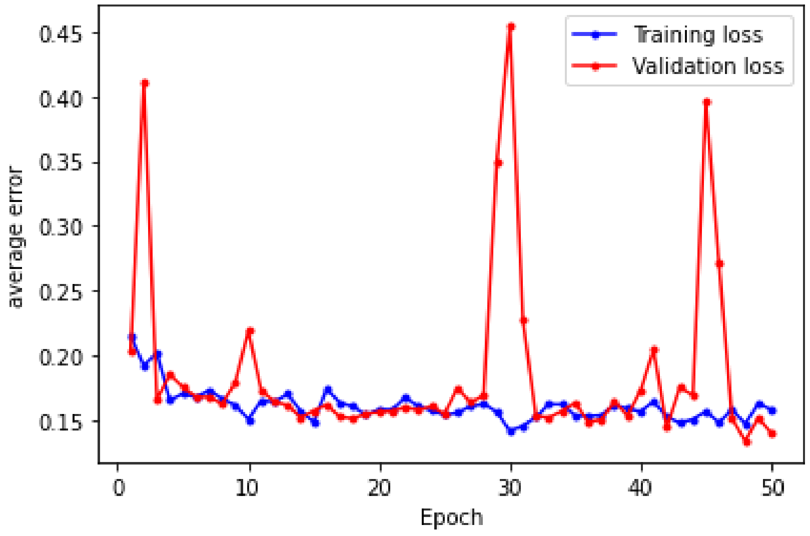


The results when $lr = 0.05$:

https://colab.research.google.com/drive/1TxWd0FIXm9WFrligMRRt71kmShjJrEyB?usp=sharing

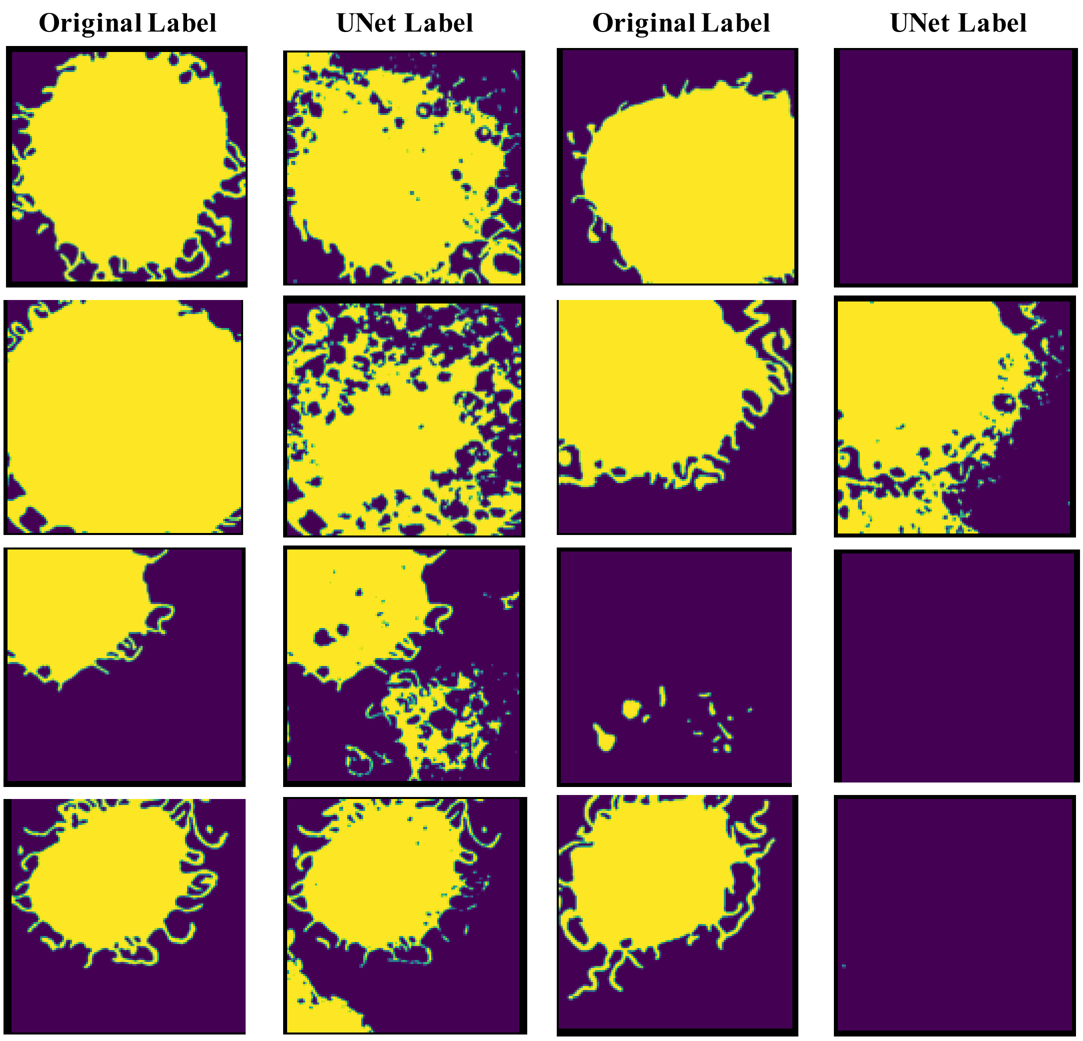


Here is the plot of *training and testing losses versus epochs*:

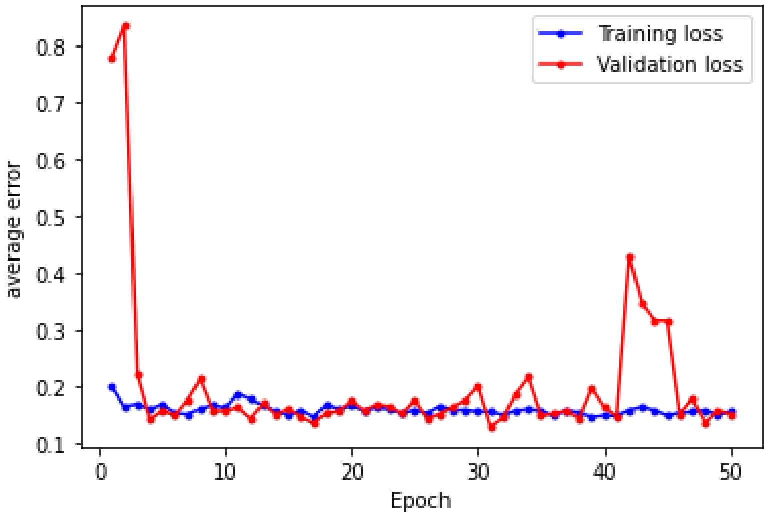


The results when $lr = 0.02$:

https://colab.research.google.com/drive/1ufQC_MtHNtuCsMA-2j2geuGVoutC04_c?usp=sharing

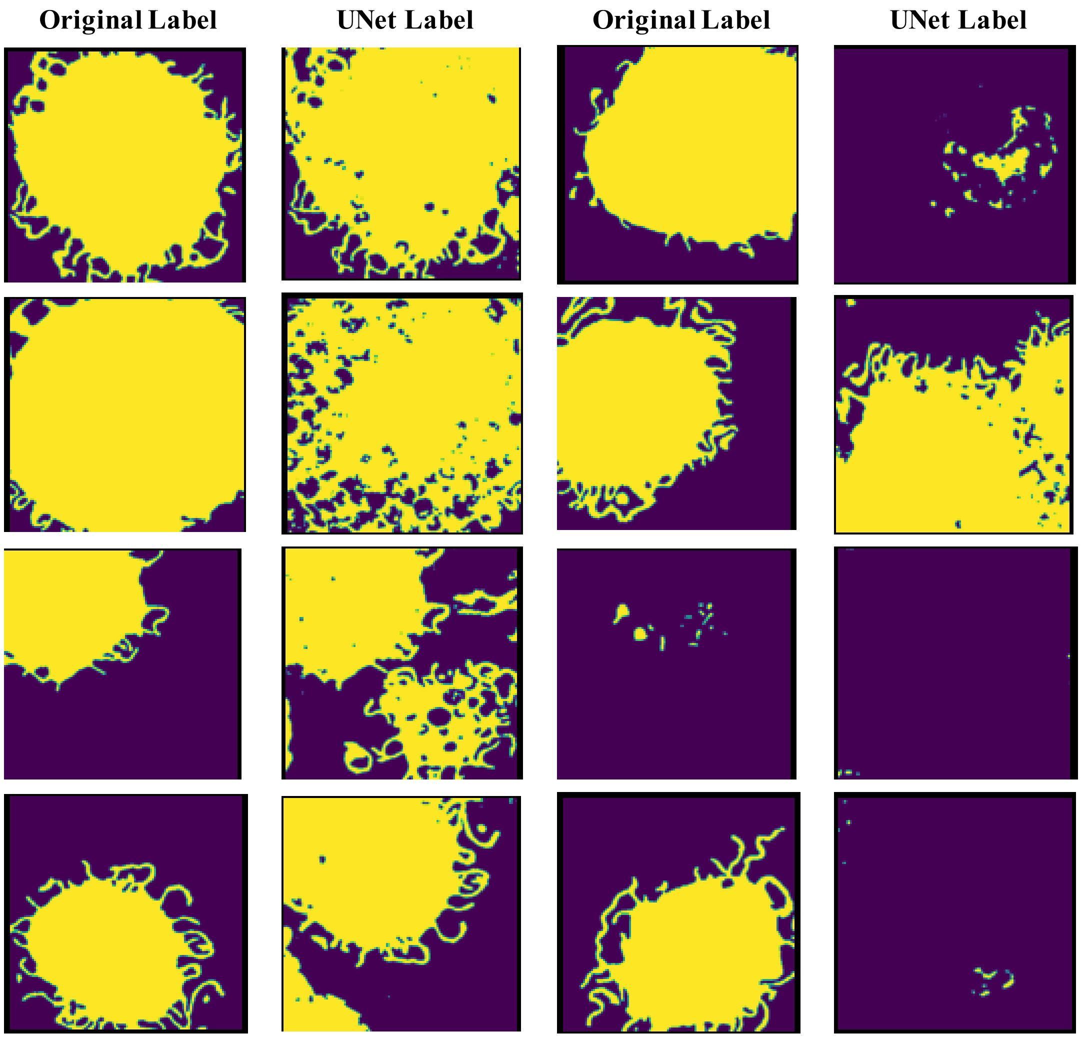


Here is the plot of *training and testing losses versus epochs*:

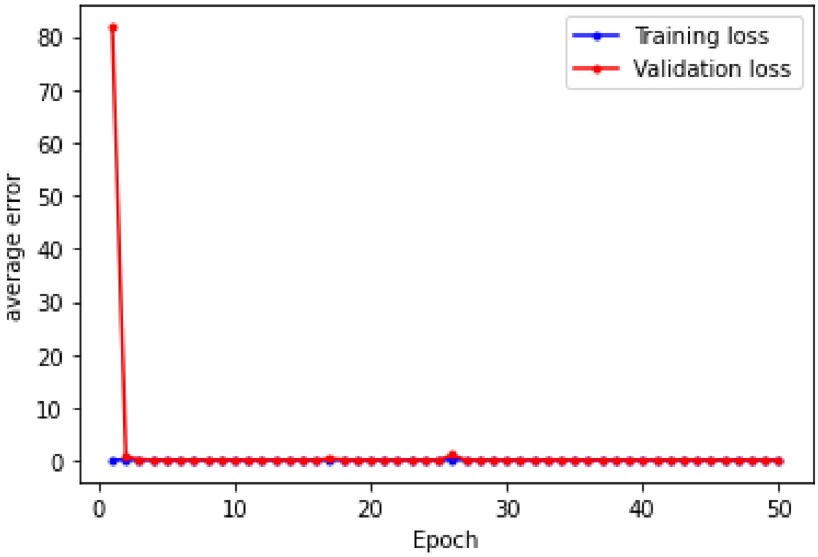


The results when $lr = 0.005$:

https://colab.research.google.com/drive/18jzpLEoA2l1WiUcPuQ9iuOFDYIUsBglv?usp=sharing



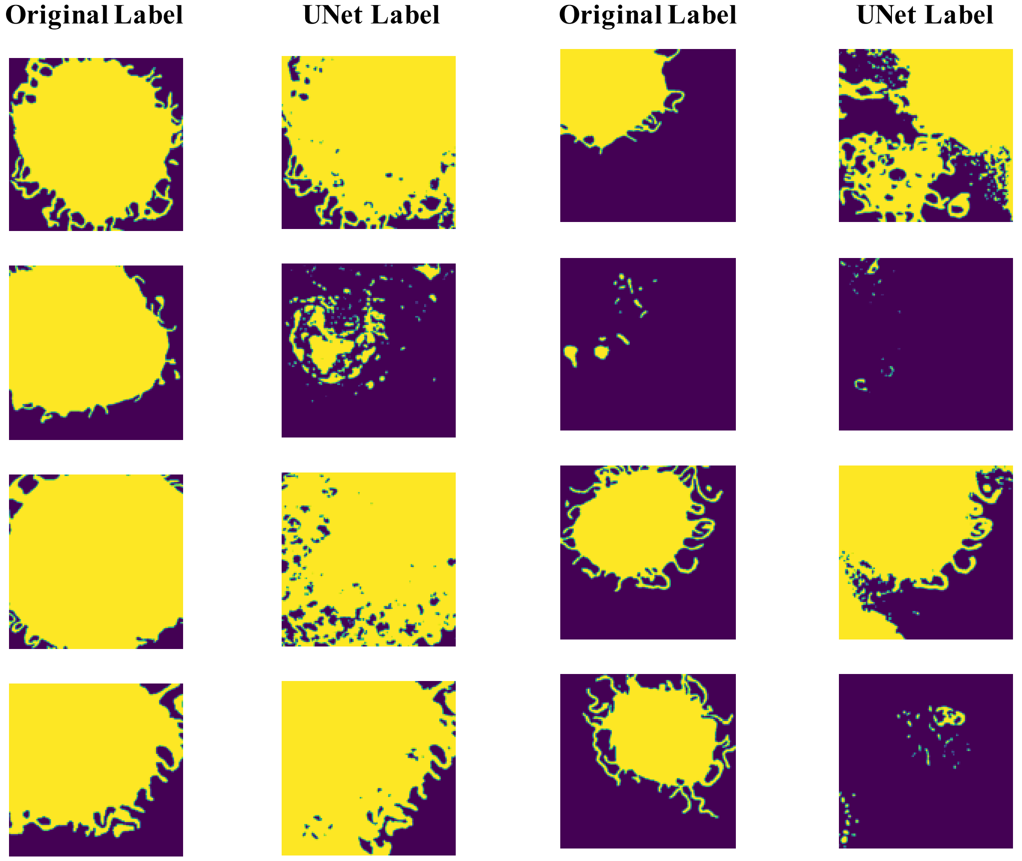
Here is the plot of *training and testing losses versus epochs*:



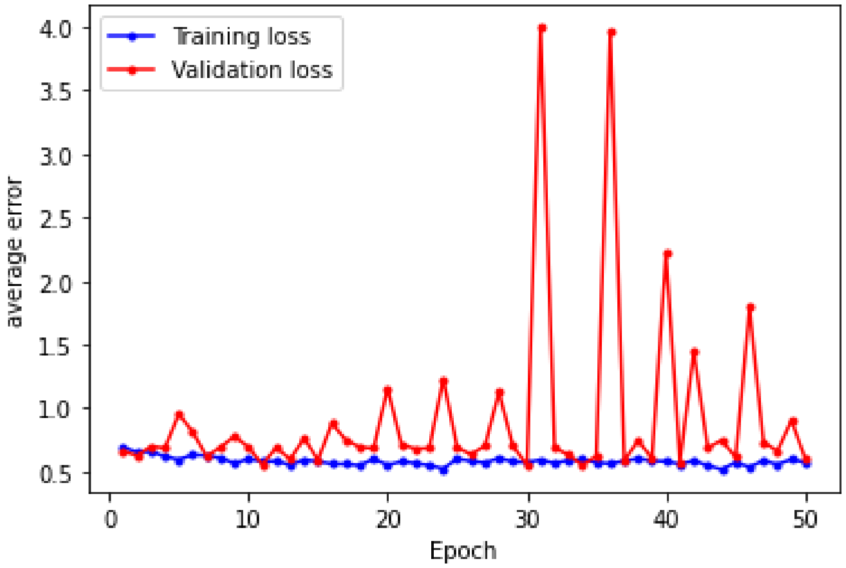
2.3 Batch Size

The result when batch size is 1:

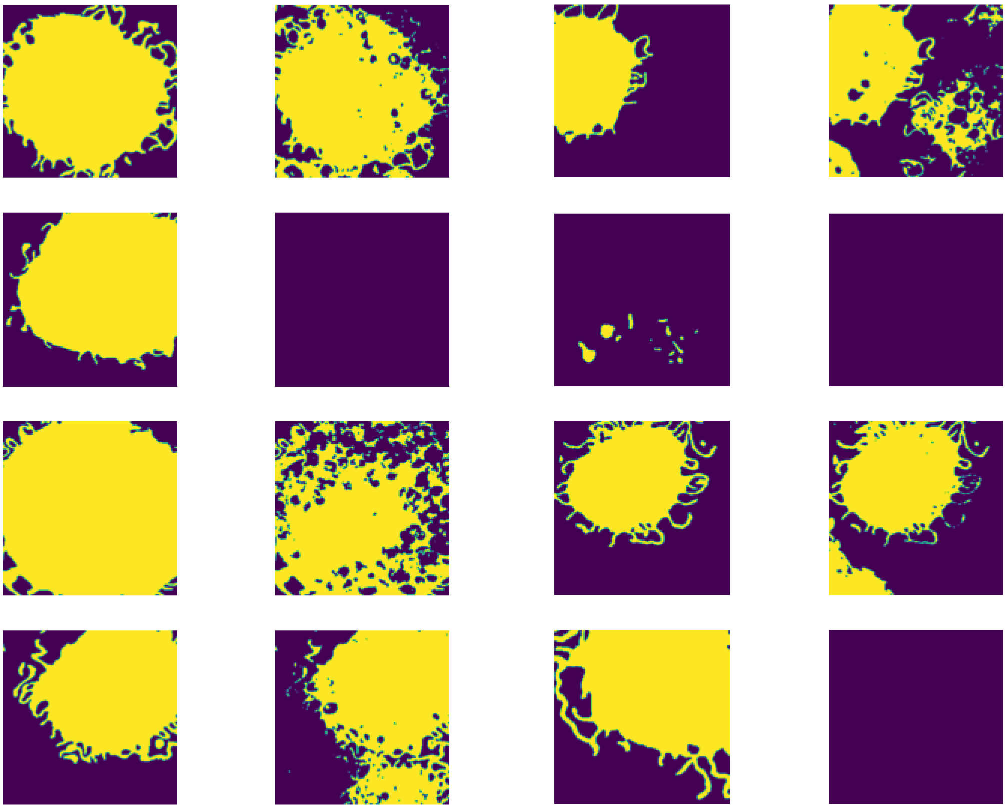
https://colab.research.google.com/drive/1l0bqMQ2u9rKUokUxYp3wT3o57Syrc9q7?usp=sharing



Here is the plot of *training and testing losses versus epochs*:

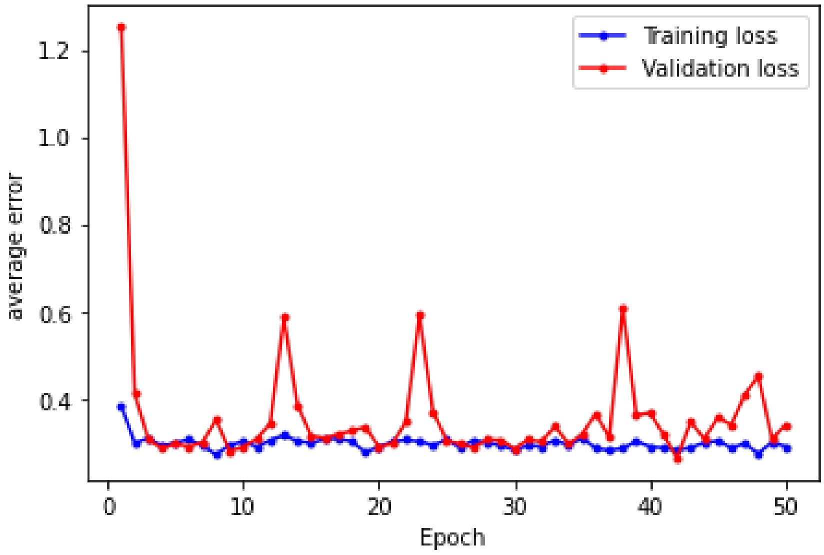


The result when batch size is 2: https://colab.research.google.com/drive/13hwQEOAGq2qawGWVlHq96b-gfmYcG36l?usp=sharing



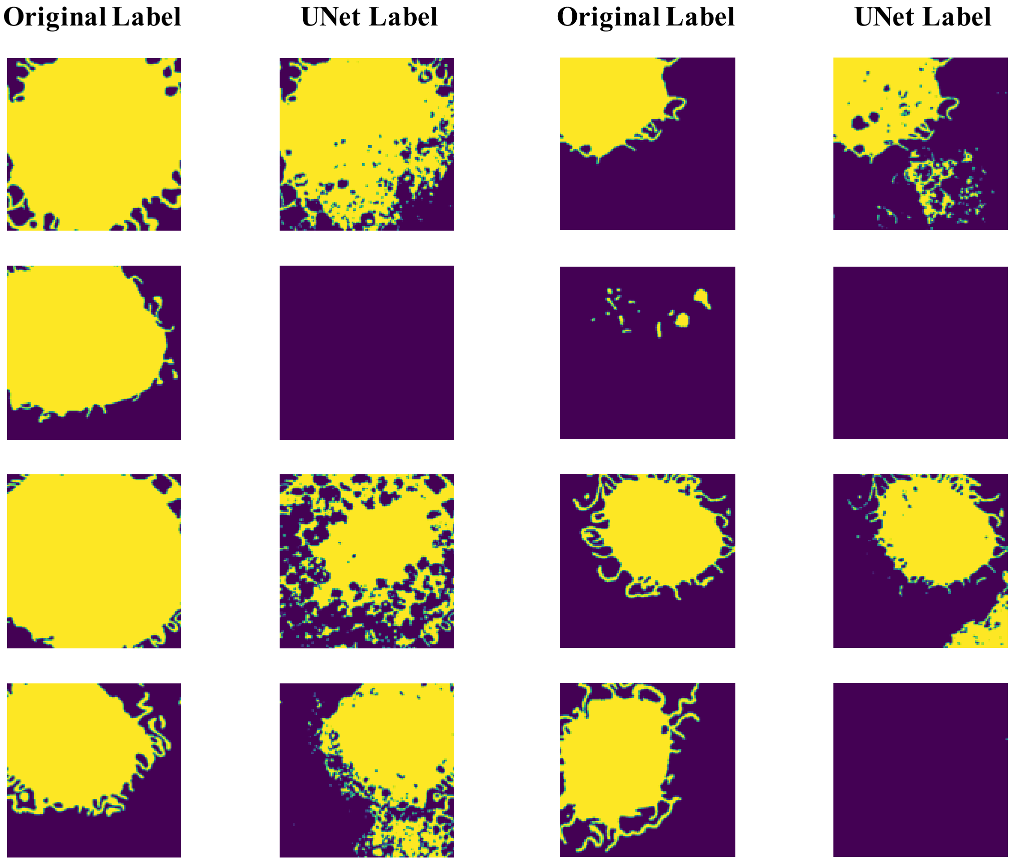
Original Label **UNet Label** **Original Label** **UNet Label**

Here is the plot of *training and testing losses versus epochs*:

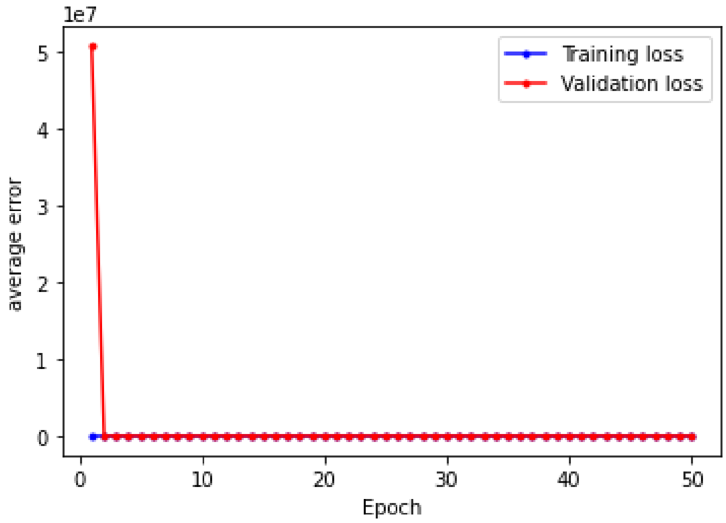


The result when batch size is 8:

https://colab.research.google.com/drive/1PynF7w9my3_RIOa0xioAGd7Vk0Cxdh64?usp=sharing

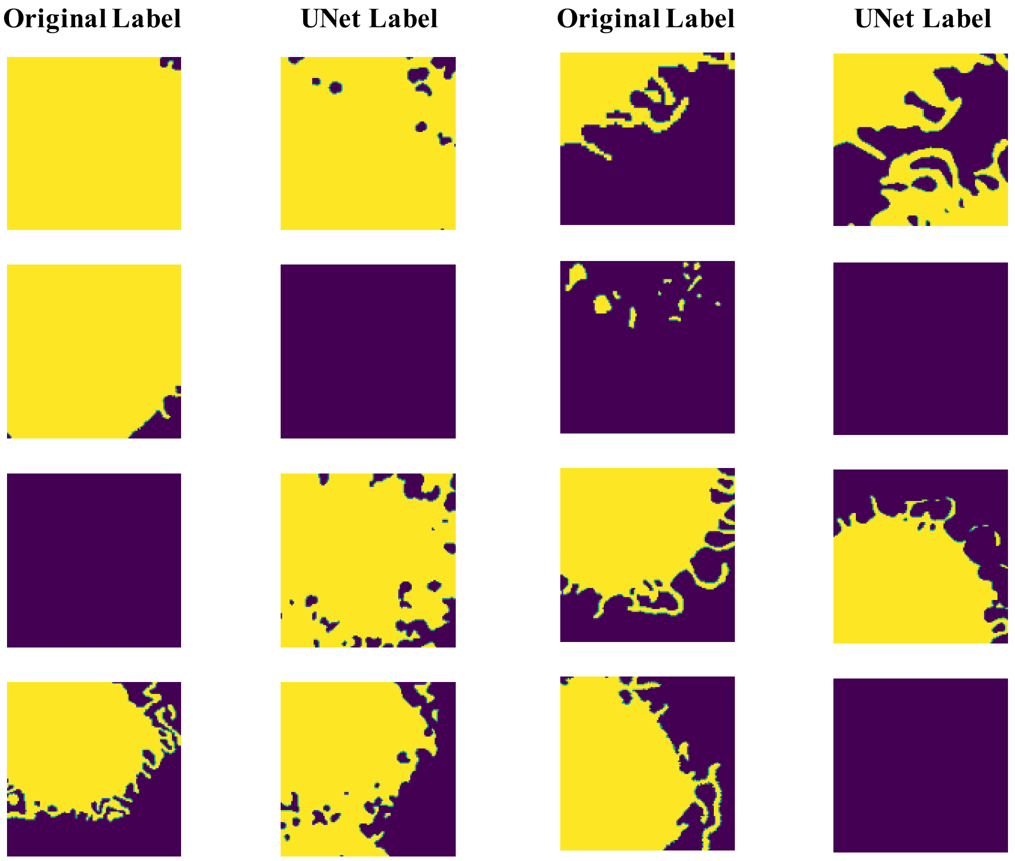


Here is the plot of *training and testing losses versus epochs*:



2.4 Image Size

Image size is 316 × 316: https://colab.research.google.com/drive/1UIBThWumOH8IvCT6AJE4uxNW6okk0niy?usp=sharing



Original Label **UNet Label** **Original Label** **UNet Label**

Here is the plot of *training and testing losses versus epochs*:

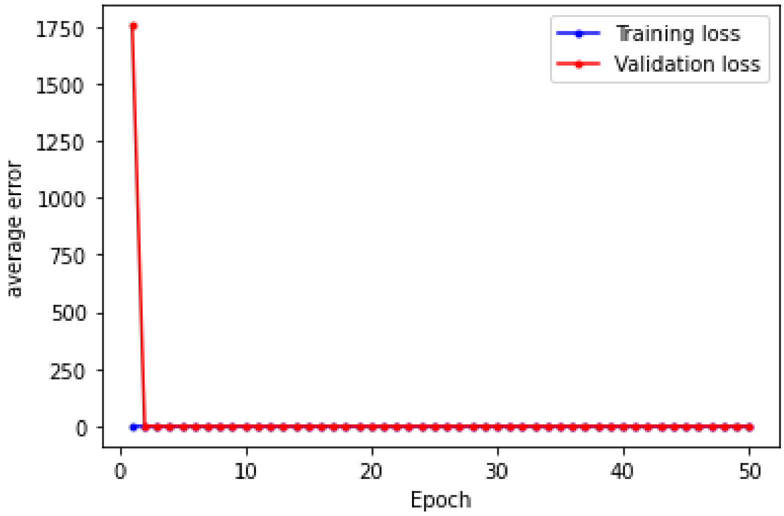
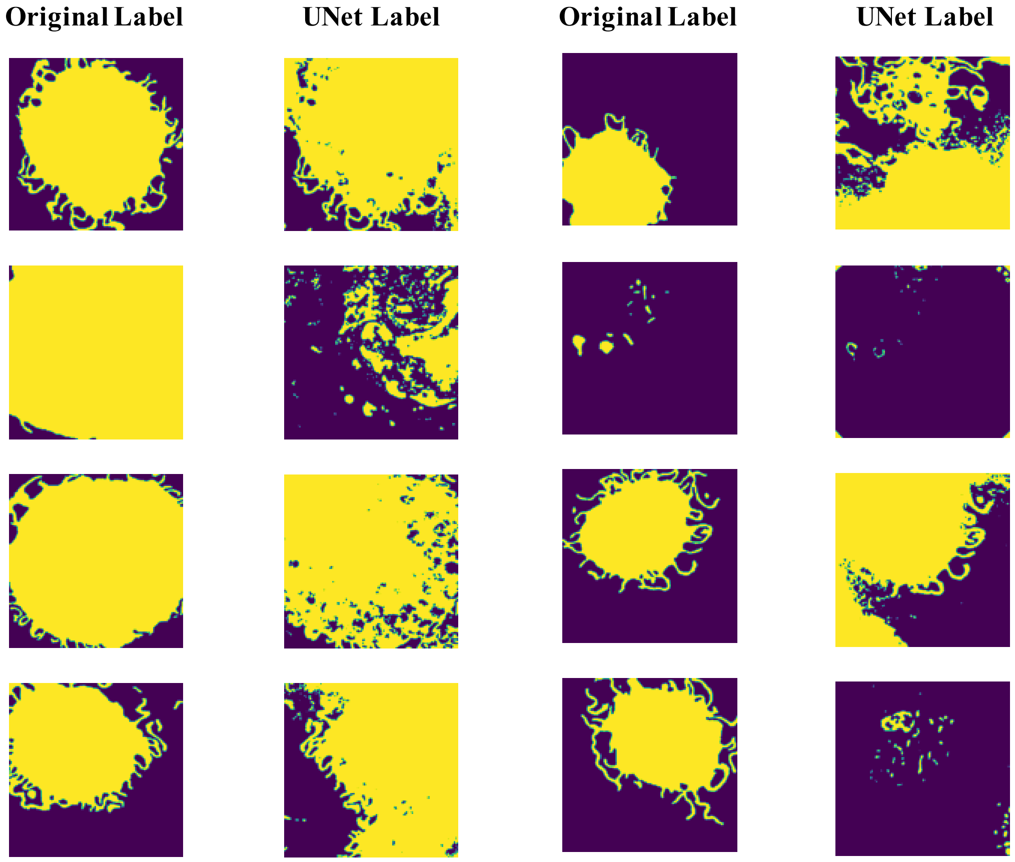
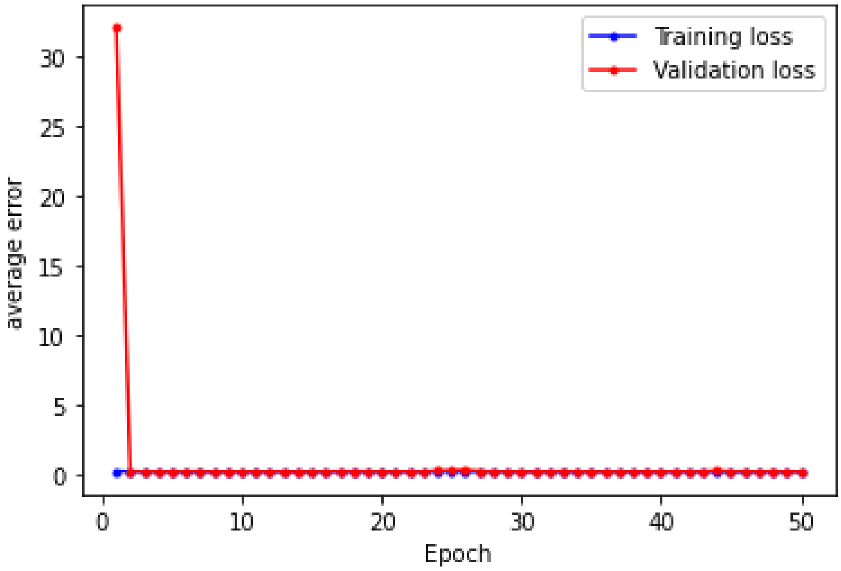


Image size is 700 × 700:

https://colab.research.google.com/drive/1zKMOXB9JNb3YV2pCH8Y_usvRaUbeoz9W?usp=sharing



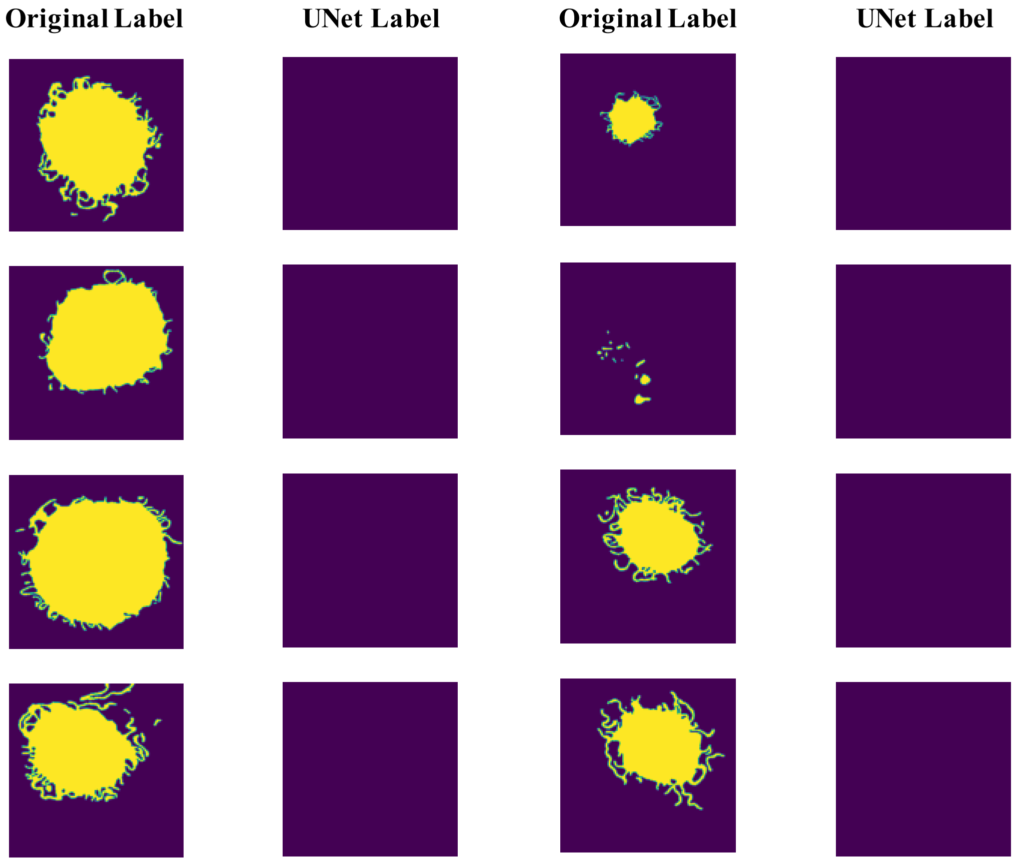
Here is the plot of *training and testing losses versus epochs*:



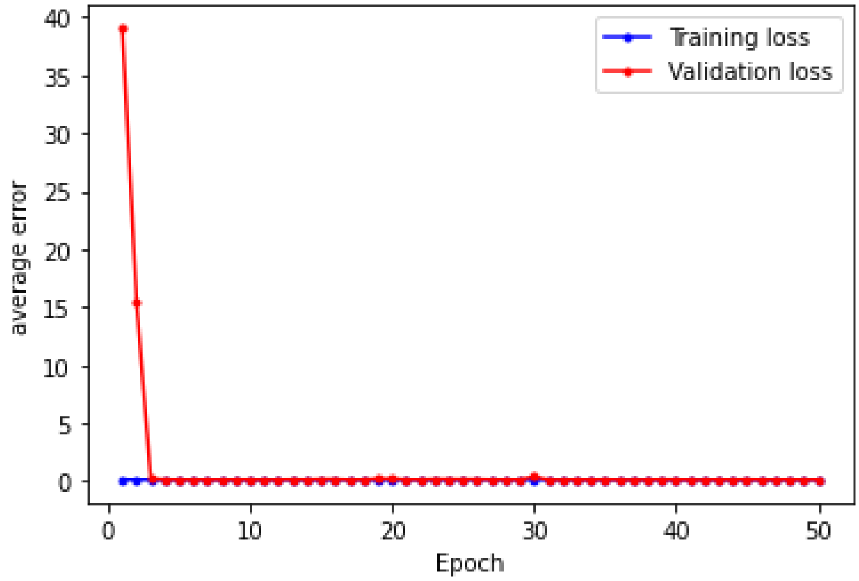
2.5 Padding

If we add "`padding = 1`" to all convolutional layers, the results will be:

https://colab.research.google.com/drive/1r02ngXNLQD2xrzwtuhM5gSdUuEudPFkU?usp=sharing



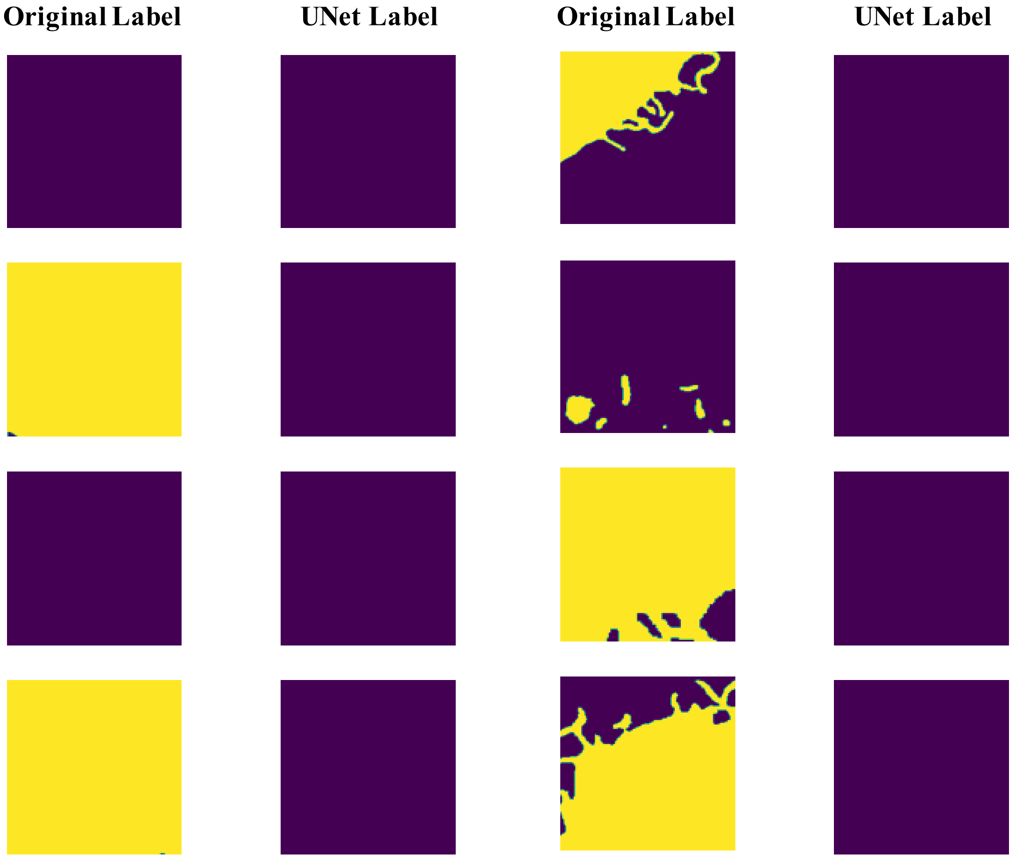
Here is the plot of *training and testing losses versus epochs*:



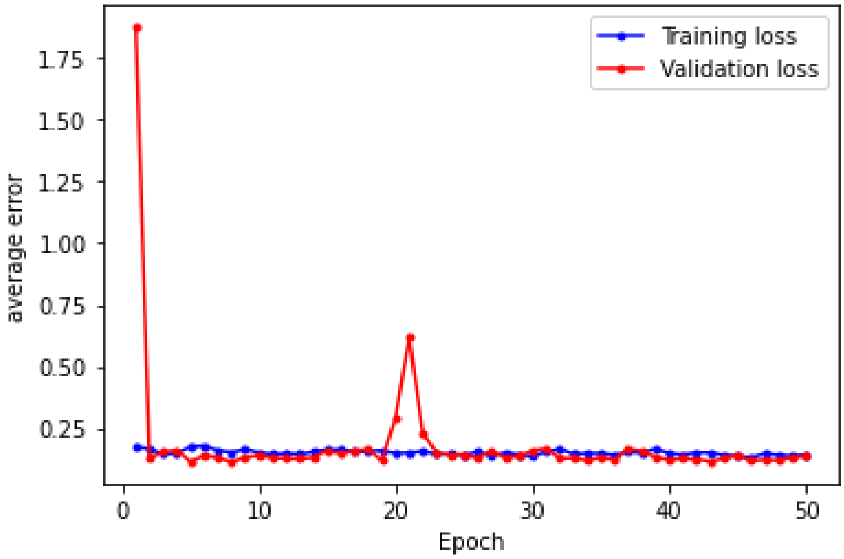
2.6 Depth of the Network

UNet has 5 layers with 4 down steps and 4 up steps. If we add 1 layer with 1 down step and 1 up step, the result will be:

https://colab.research.google.com/drive/13QDhcDey4AgkcfUhJ8n7uxUPeBQZiAPr?usp=sharing

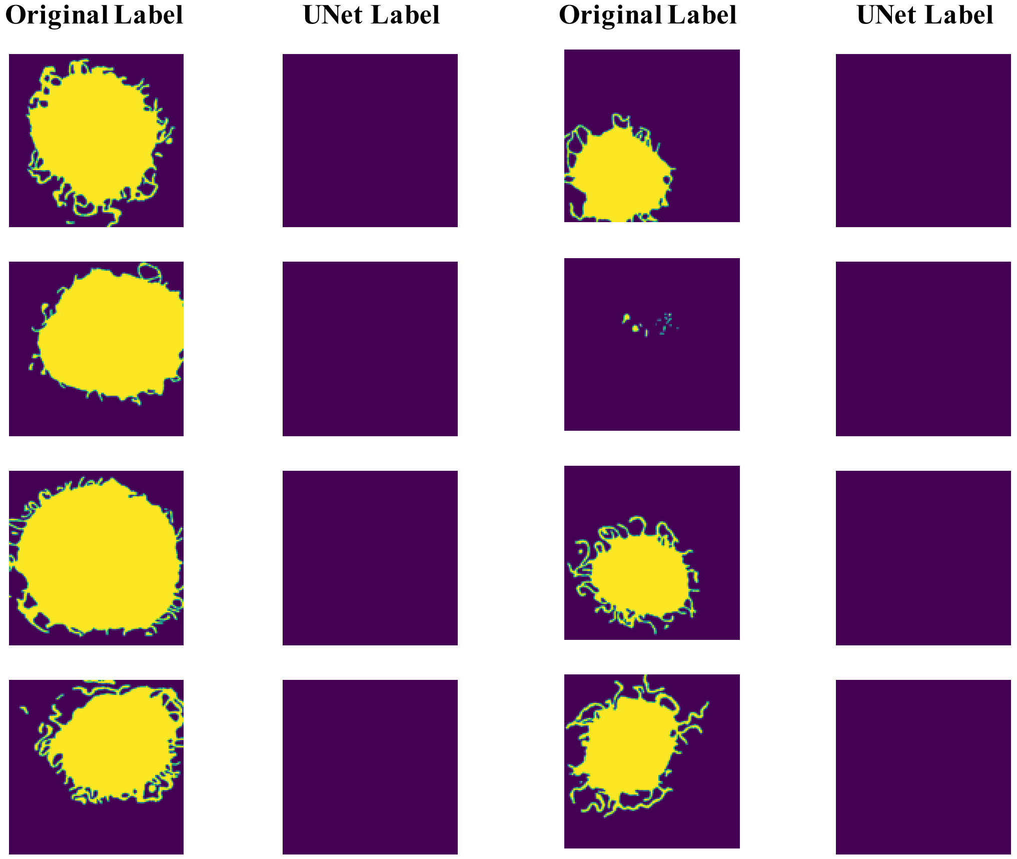


Here is the plot of *training and testing losses versus epochs*:

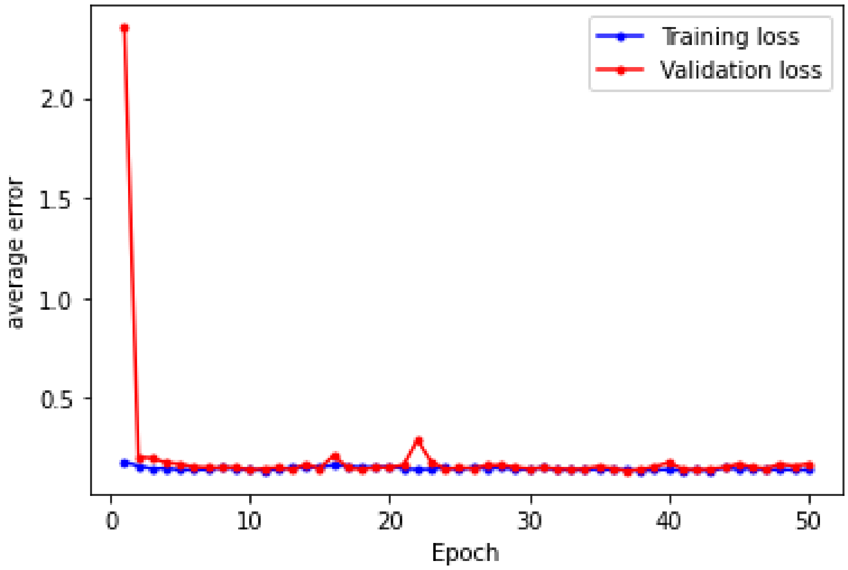


UNet has 5 layers with 4 down steps and 4 up steps. If we remove 1 layer with 1 down step and 1 up step, the result will be:

https://colab.research.google.com/drive/11fb6kZbY2ZnUIANqaK0f2Gvr7sVy37A7?usp=sharing



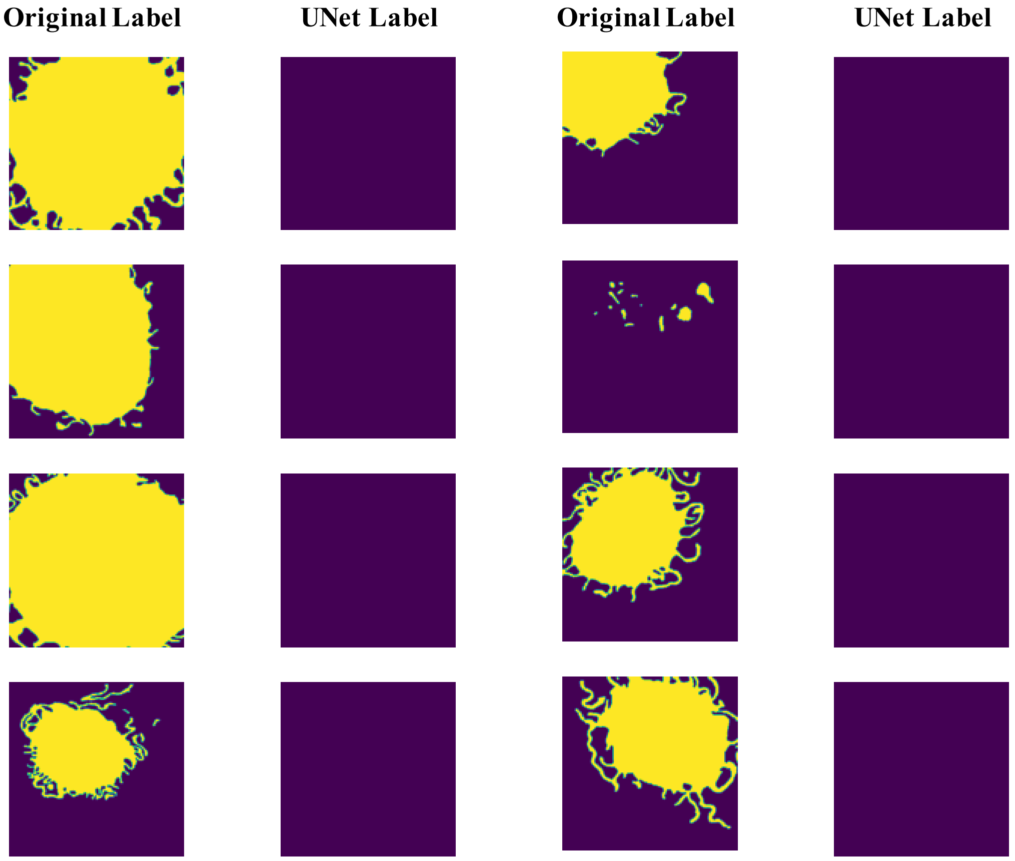
Here is the plot of *training and testing losses versus epochs*:



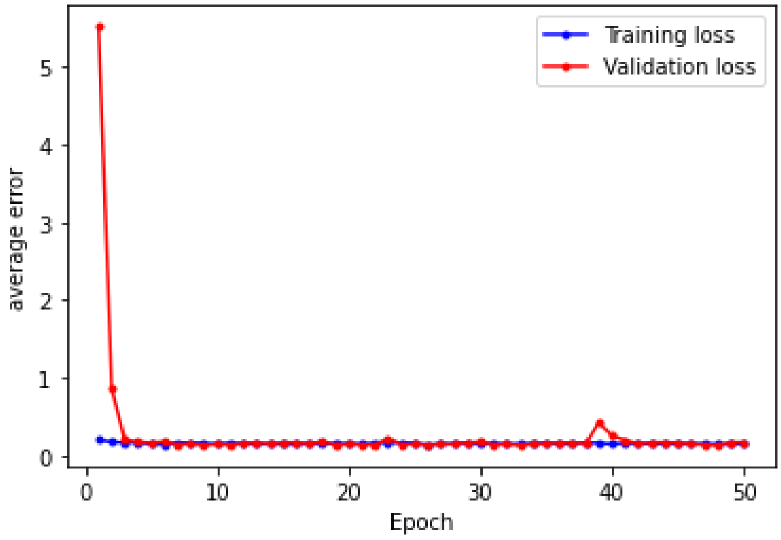
2.7 Number of Convolutional Kernels

In UNet, in each layer there are 64, 128, 256, … channels. If we double the number of channels in each layer, the result will be:

https://colab.research.google.com/drive/1y5wm7He3_VPRLRM4KjYeZU6Jc5zH20YE?usp=sharing

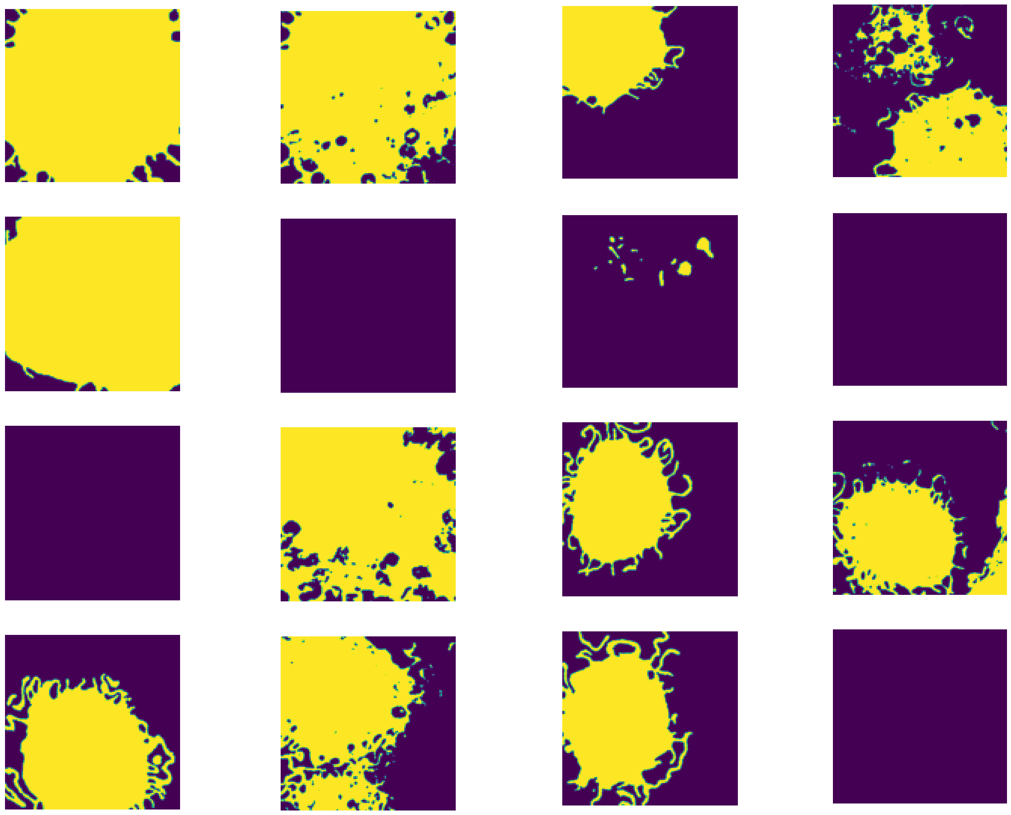


Here is the plot of *training and testing losses versus epochs*:



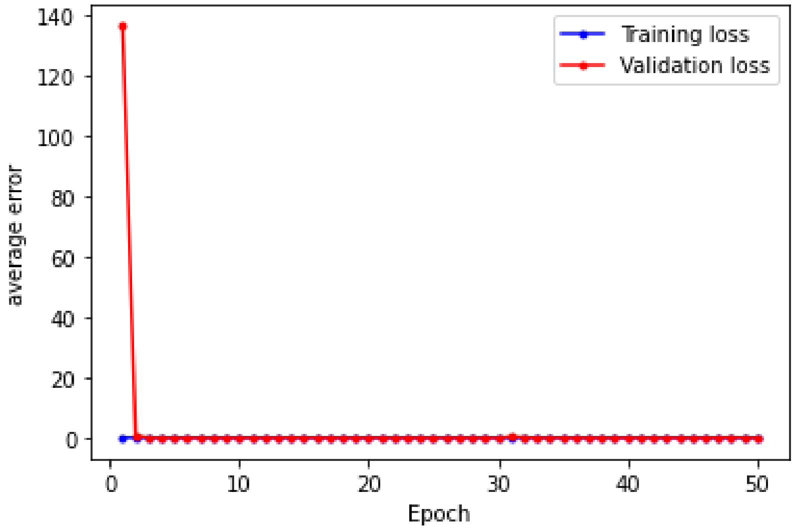
In UNet, in each layer there are 64, 128, 256, … channels. If we halve the number of channels in each layer, the result will be:

https://colab.research.google.com/drive/1BUmknR45V89d8oy_YjSxEK0j9Bgc7khG?usp=sharing



Original Label **UNet Label** **Original Label** **UNet Label**

Here is the plot of *training and testing losses versus epochs*:


Training loss
Validation loss
average error
Epoch