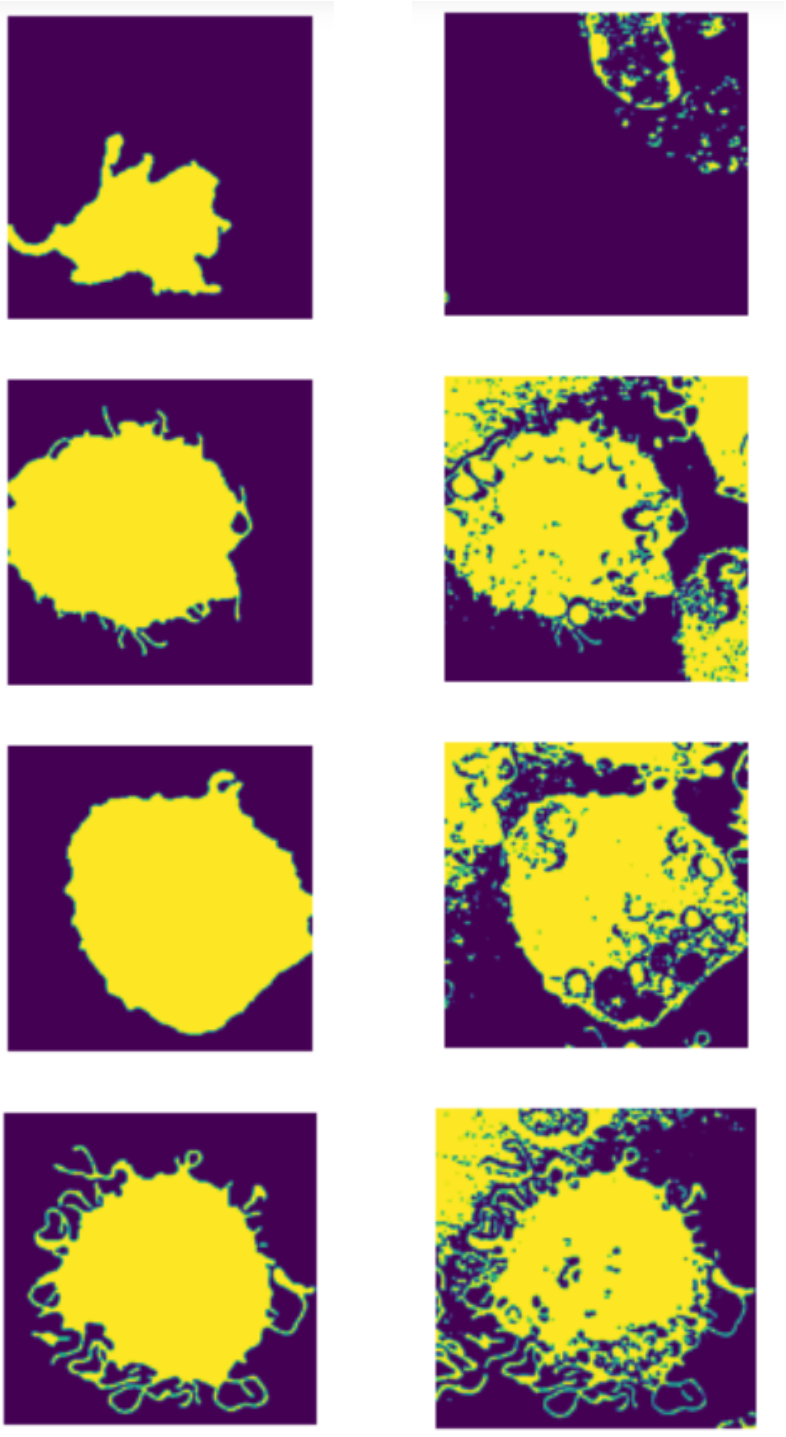
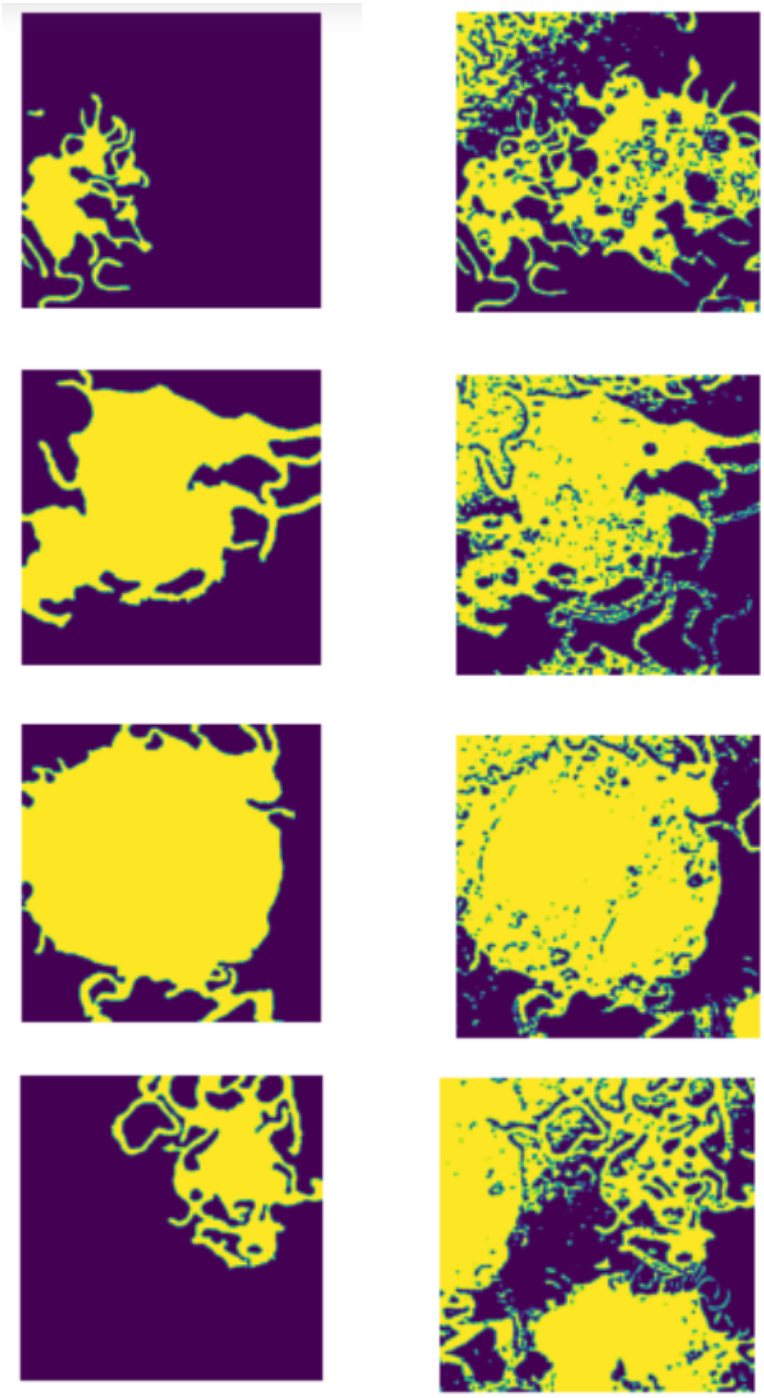
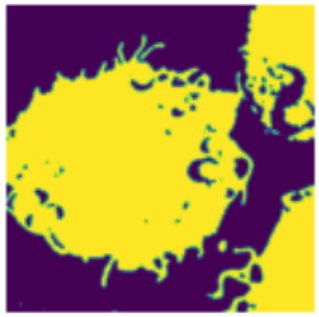
Cmpt 733 assignment 1 report

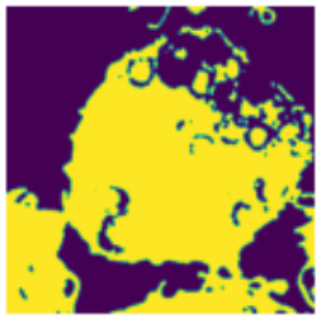
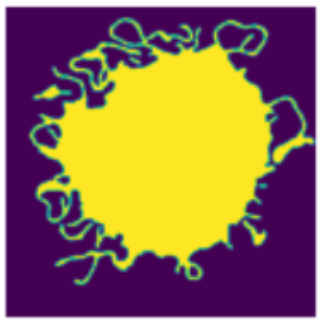
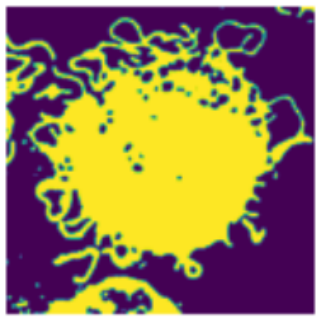
- Result images(left column are original labels and right column are resulting segmentation of the test images):
    - 20 iterations without augmentation:

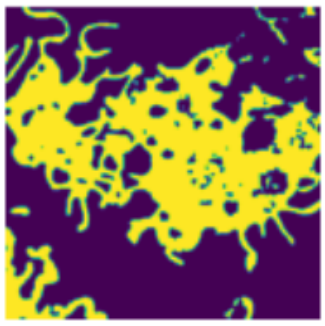


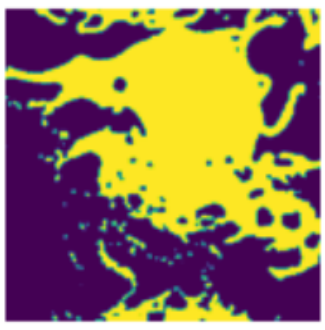


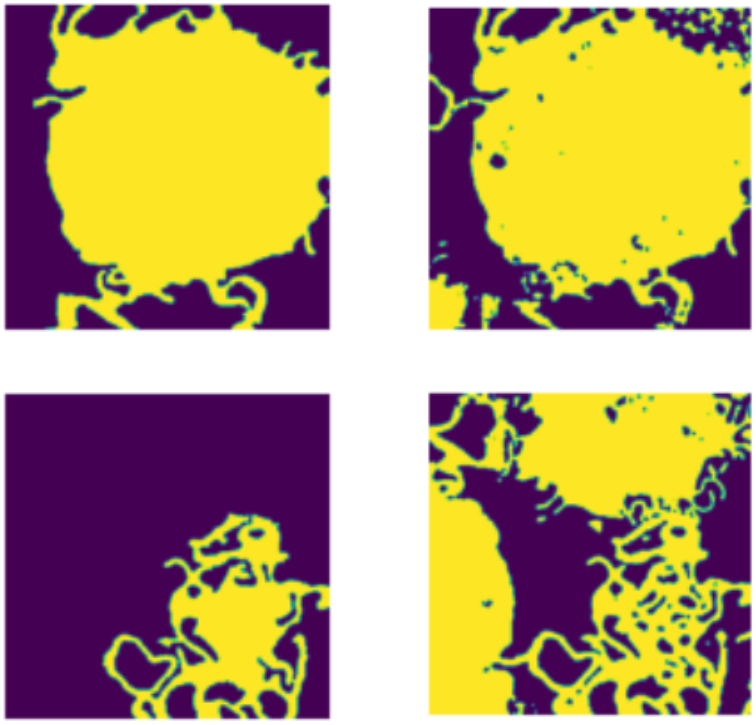
- 20 iterations with augmentation:



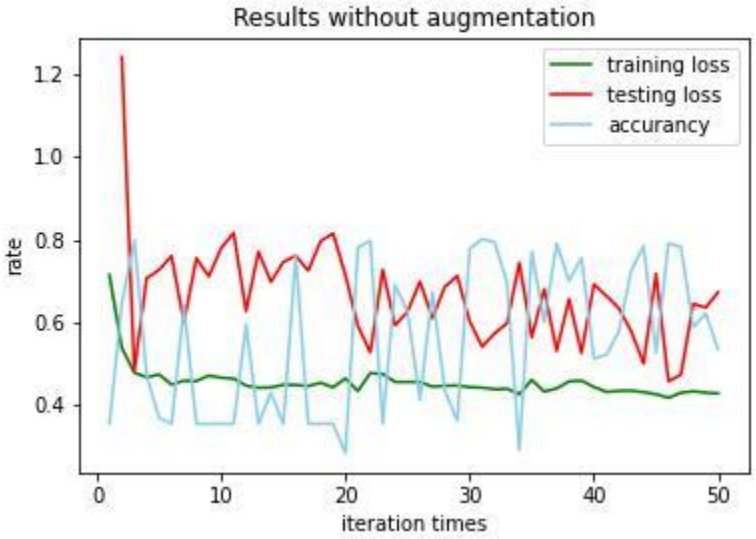


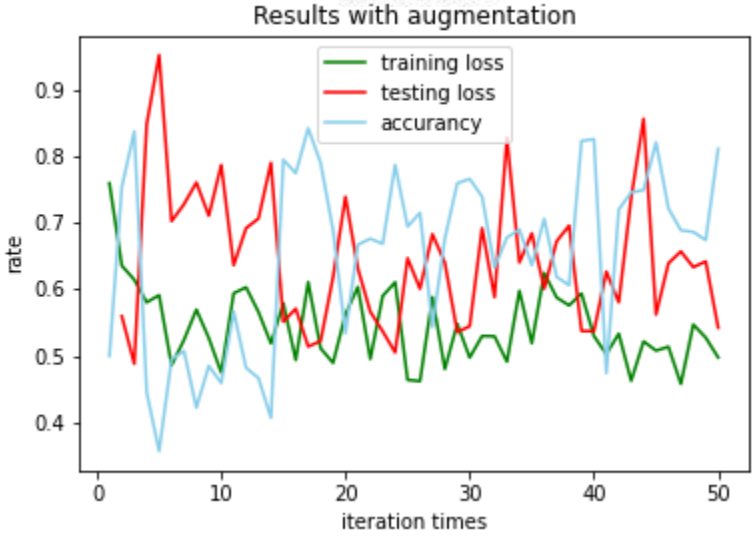


- For all of the training images, I utilized a 572*572 input size, and the output size is 388*388.
- I set a random number in the __getitem__() function, so every time its trains, the read image will undergo a random augmentation. And there are four options for augmenting the image.
    - For the vertical flip, I used the RandomVerticalFlip() function in the transform method of Pytorch and set the probability parameter to 1
    - For the horizontal flip, I did exactly the same as the vertical flip, I used the RandomHorizontalFlip() function and set the probability to 1.
    - Zooming in on an image may be challenging. At first, I used the RandomResizedCrop() function, but I immediately discovered that the augmenting images and labels did not match. This is because the code randomly crops a 572*572 size image from the expanded image. Although the enlarged size is the same, the cropped area is not guaranteed to be consistent. Finally, I used the resize() function and the CenterCrop() function in the transform method to ensure the same image and label scaling.
    - Rotate images are similar to zooming, the RandomRotation() function cannot be used because of its random component. So I first convert tensor to Numpy array form, and then use NumPy.transpose() function flips the

array, and finally flips the Numpy array horizontally to complete the 90-degree clockwise rotation of the image.

- The structure of my network and the size of the input/output of each layer is strictly in accordance with the assignment.
- train-test losses result for 50 iterations:





- I used batch size = 1 and learning rate = 0.01
- The time to run 50 epochs on a Geforce RTX2060 graphics card is about 4 minutes.