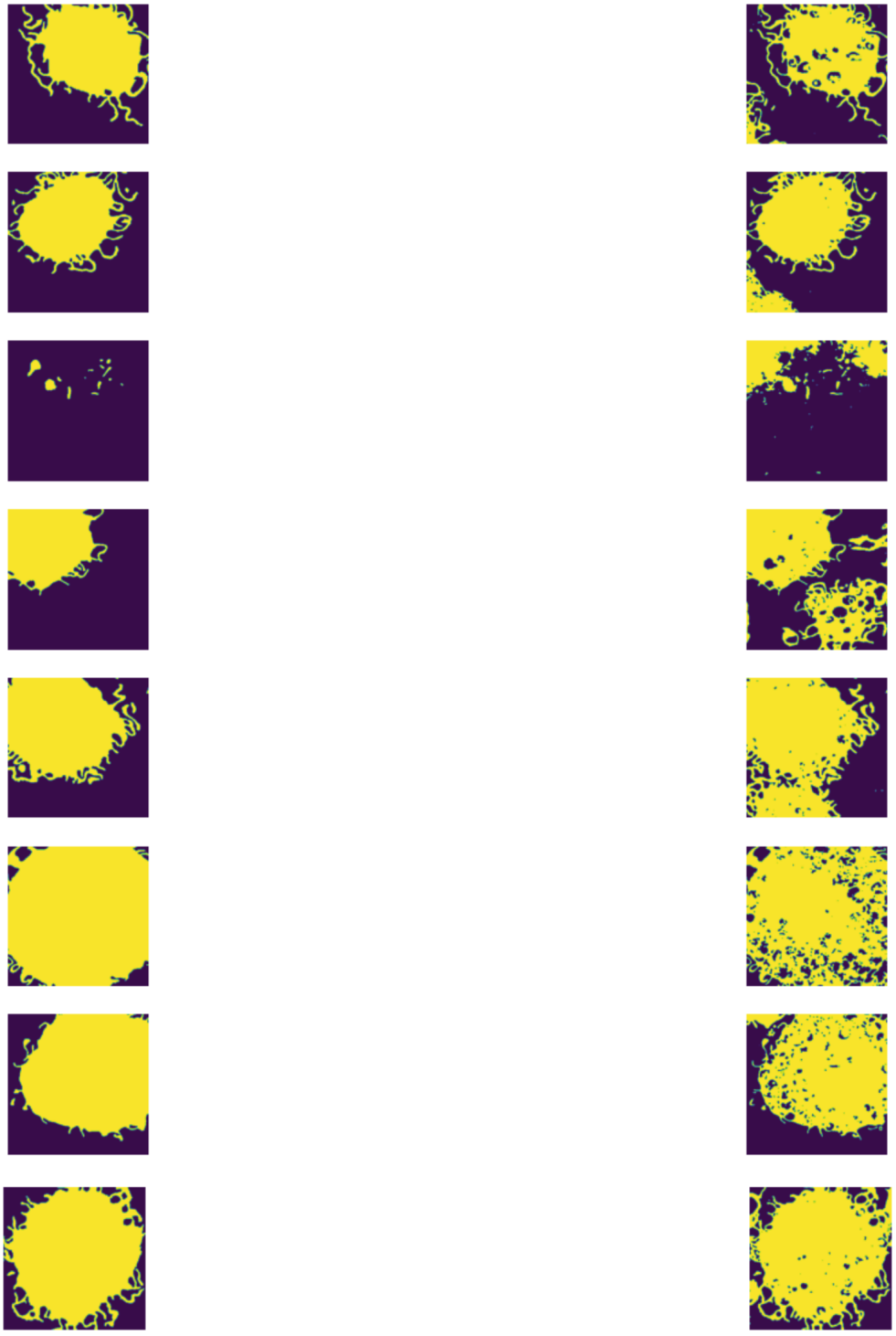
Report

1 Resulting segmentation of the test images (test set accuracy 89.21%):
(left hand side: labels, right hand side: predictions)



2 Size of the images: 572*572
The less pixels are reserved, the less information the model could get. Also, the amount of the data set is not that big. We don't need to worry about the computation ability that much. So I went with an acceptable accuracy and a comparatively larger size of images.

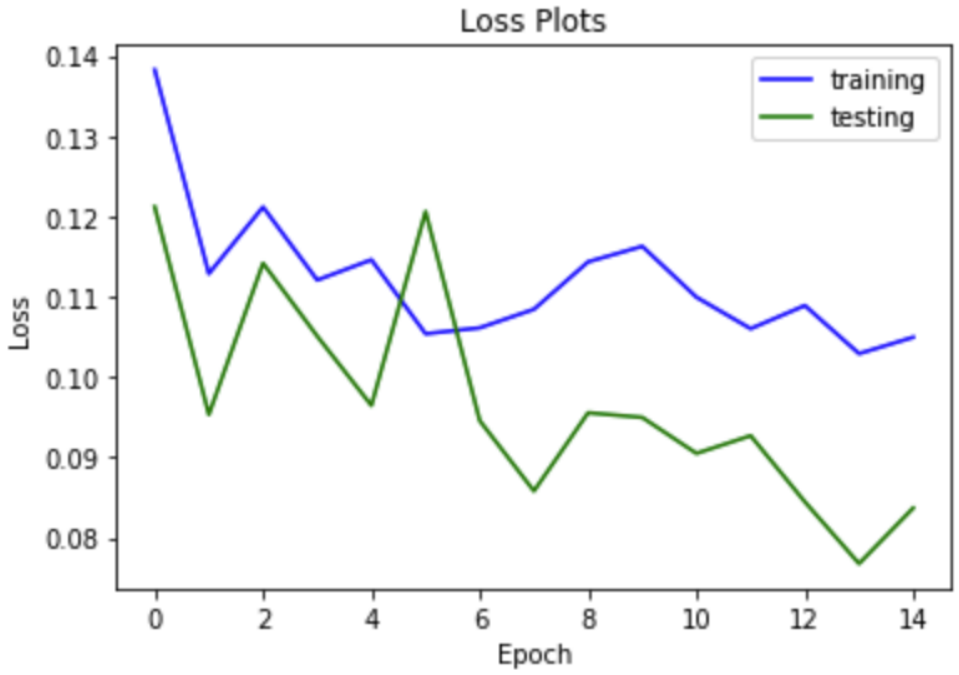
3 Data augmentation strategies:
My data augmentation strategy is: for each image in the training set, randomly choose one augmentation technique from Horizontal/Vertical flip, Zooming, and Rotation(rotation angle: -15 to 15 degree), then apply it. In this way, I doubled the size of the training set.

When I tried not to implement data augmentation,the accuracy of test data would not be as high as when data augmentation was implemented. This is because the raw data could not give the model a more generalized view of the data as the augmented data does.

4 Epochs

(1) Number of epochs: 15
When I tried less epochs, the losses and accuracy of the last epoch could not approach the optimal value. While more epochs led to the plateau or even increase of the loss and decrement of the accuracy. That's why I chose this epoch number at last.

(2) Plots for the train-test losses:



5 Batch size: 4
Usually, the smaller the batch size is, the more noisy the updating of the weights gets. This noise can sometimes help in getting out of local optima. But in other times, it could prevent the

convergence to an optima. Generally speaking, larger batch sizes work for convex errors, while smaller batch sizes are better in situations with many deep local optima.

When I tried larger batch sizes, the loss plot became very fluctuating and was no longer a comparatively steady descent line. That's why I thought a larger batch size would not be suitable for training in this case and chose to stick with the batch size 4.

6 Learning rate: 0.001
Learning rate is the step size when updating weights. Too big learning rate could cause over shooting of the optima point. And too small learning rate could give very slow convergence.

I tried three learning rates–0.01, 0.001, and 0.0001. Learning rate 0.01 gave a lower accuracy and 0.0001 made the model learn too slow. So I decided to use 0.001 as the learning rate.

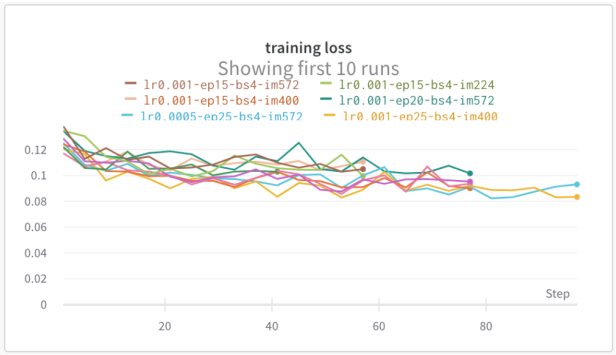
7 The training takes: 213.65 seconds

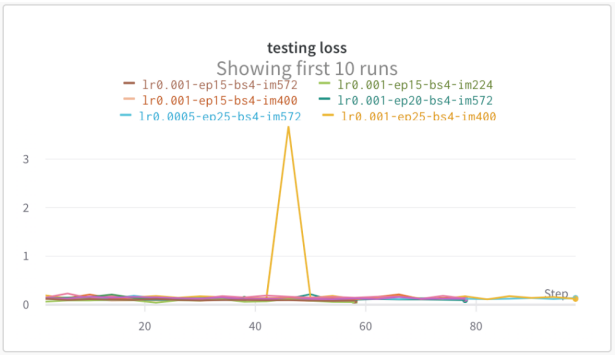
8 Use weights and biases to log the training loss, validation loss, hyperparameters, etc.

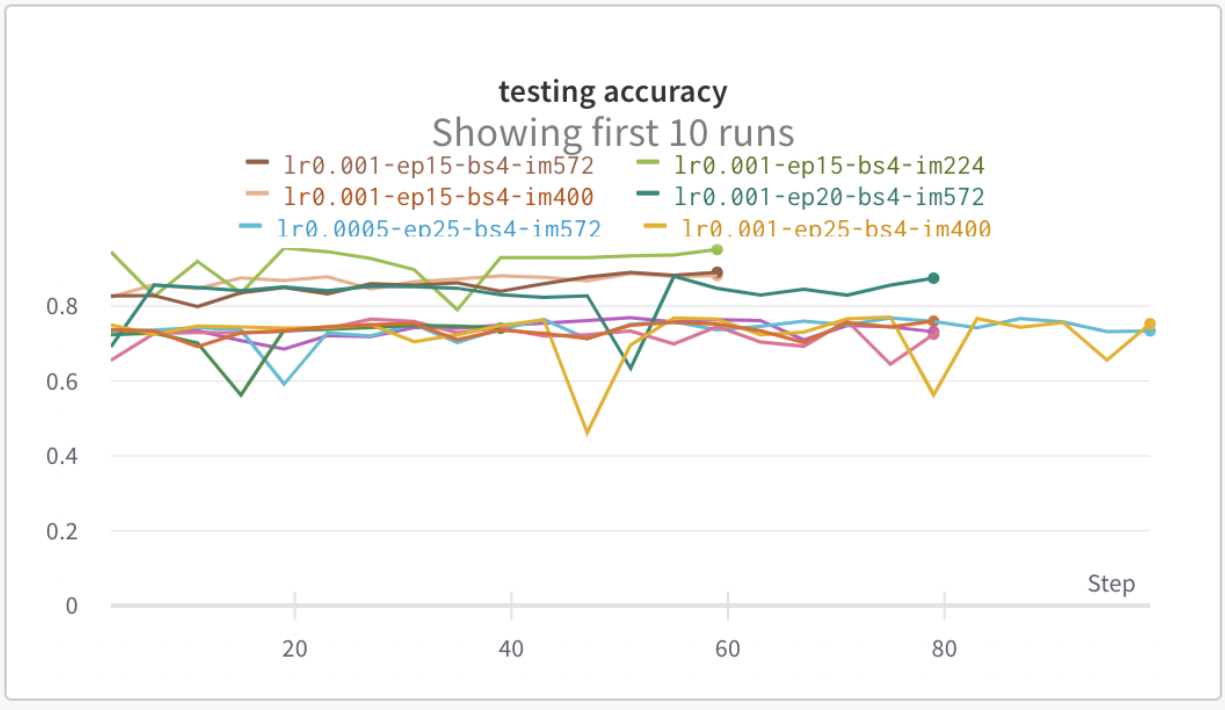
Log of training/test loss, test accuracy, learning rate, batch size, number of epochs, image size:

Name (44 visualized)	State	Notes	User	Tags	Created	Runtime	Sweep	batch_size	epochs	image_size	learning_ra	epoch	testing acc	testing loss	training los
lr0.001-ep15-bs4-im572	finished	Add notes	chenqiao		13m ago	5m 8s	-	4	15	572	0.001	15	0.8905	0.08372	0.105
lr0.001-ep15-bs4-im224	finished	Add notes	chenqiao		22m ago	51s	-	4	15	224	0.001	15	0.9514	0.05352	0.09961
lr0.001-ep15-bs4-im400	finished	Add notes	chenqiao		25m ago	2m 20s	-	4	15	400	0.001	15	0.8818	0.08163	0.1106
lr0.001-ep20-bs4-im572	finished	Add notes	chenqiao		1h ago	5m 41s	-	4	20	572	0.001	20	0.8748	0.09397	0.1016
lr0.0005-ep25-bs4-im572	finished	Add notes	chenqiao		1h ago	7m 15s	-	4	25	572	0.0005	25	0.7339	0.1386	0.0931
lr0.001-ep25-bs4-im400	finished	Add notes	chenqiao		1h ago	3m 26s	-	4	25	400	0.001	25	0.753	0.1185	0.08336
lr0.001-ep20-bs4-im572	finished	Add notes	chenqiao		2h ago	5m 22s	-	4	20	572	0.001	20	0.7316	0.1224	0.09531
lr0.001-ep20-bs4-im576	finished	Add notes	chenqiao		2h ago	15m 21s	-	4	20	576	0.001	20	0.7596	0.1224	0.09005
lr0.001-ep20-bs4-im400	finished	Add notes	chenqiao		2h ago	6m 2s	-	4	20	400	0.001	20	0.725	0.1296	0.09364
lr0.001-ep15-bs4-im572	finished	Add notes	chenqiao		2h ago	2m 55s	-	4	15	572	0.001	11	0.741	0.121	0.1029
lr0.001-ep15-bs4-im572	finished	Add notes	chenqiao		2h ago	8m 36s	-	4	15	572	0.001	15	0.73	0.1463	0.1011
lr0.001-ep15-bs4-im572	finished	Add notes	chenqiao		3h ago	4m 30s	-	4	15	572	0.001	8	0.7571	0.1104	0.09487
lr0.001-ep15-bs4-im572	finished	Add notes	chenqiao		3h ago	3m 57s	-	4	15	572	0.001	15	0.382	0.5155	0.09609
lr0.001-ep15-bs4-im572	finished	Add notes	chenqiao		13h ago	15m 34s	-	4	15	572	0.001	15	0.8302	0.09887	0.1065
lr0.001-ep15-bs4-im572	finished	Add notes	chenqiao		14h ago	1h 7m 23s	-	4	15	572	0.001	15	0.4397	0.2336	0.1129
lr0.001-ep15-bs4-im572	finished	Add notes	chenqiao		14h ago	8m 12s	-	4	15	572	0.001	15	0.8135	0.165	0.1094
lr0.0001-ep15-bs4-im572	finished	Add notes	chenqiao		14h ago	4m 29s	-	4	15	572	0.0001	15	0.7633	0.1064	0.09919
lr0.01-ep15-bs4-im572	finished	Add notes	chenqiao		14h ago	3m 58s	-	4	15	572	0.01	15	0.5834	0.2553	0.1124
lr0.001-ep15-bs12-im572	finished	Add notes	chenqiao		14h ago	7m 48s	-	12	15	572	0.001	15	0.7366	0.05599	0.03196
lr0.001-ep15-bs8-im572	finished	Add notes	chenqiao		15h ago	4m 21s	-	8	15	572	0.001	15	0.7328	0.06051	0.05699

Plot of the log (next page):


training loss
Showing first 10 runs
lr0.001-ep15-bs4-im572
lr0.001-ep15-bs4-im224
lr0.001-ep15-bs4-im400
lr0.001-ep20-bs4-im572
lr0.0005-ep25-bs4-im572
lr0.001-ep25-bs4-im400
0.12
0.1
0.08
0.06
0.04
0.02
0
Step
20
40
60
80


testing loss
Showing first 10 runs
lr0.001-ep15-bs4-im572
lr0.001-ep15-bs4-im224
lr0.001-ep15-bs4-im400
lr0.001-ep20-bs4-im572
lr0.0005-ep25-bs4-im572
lr0.001-ep25-bs4-im400
3
2
1
0
Step
20
40
60
80


testing accuracy
Showing first 10 runs
lr0.001-ep15-bs4-im572
lr0.001-ep15-bs4-im224
lr0.001-ep15-bs4-im400
lr0.001-ep20-bs4-im572
lr0.0005-ep25-bs4-im572
lr0.001-ep25-bs4-im400
0.8
0.6
0.4
0.2
0
Step
20
40
60
80