lab9

December 10, 2024

1 ECE 5470 Lab 9 Report

1.1 by Mohamed Zakaria Kheder

[6]:

```
from v4 import vd
from IPython.display import display
from PIL import Image
```

1.2 A.1 running the tutorial models

Model convnet1 ——- Epoch 1 ——- Training [160/500 (31%)] Loss: 0.483841 [320/500 (62%)] Loss: 0.306435 [480/500 (94%)] Loss: 0.173066 Loss = 0.3993227462768555 Validation Loss = 0.15031305909156797 ——- Epoch 2 ——- Training [160/500 (31%)] Loss: 0.093907 [320/500 (62%)] Loss: 0.070127 [480/500 (94%)] Loss: 0.070395 Loss = 0.0901779573559761 Validation Loss = 0.0692278477549553 ——- Epoch 3 ——- Training [160/500 (31%)] Loss: 0.064567 [320/500 (62%)] Loss: 0.055768 [480/500 (94%)] Loss: 0.063742 Loss = 0.06490166953206063 Validation Loss = 0.06595850929617882 ——- Epoch 4 ——- Training [160/500 (31%)] Loss: 0.059918 [320/500 (62%)] Loss: 0.069052 [480/500 (94%)] Loss: 0.055536 Loss = 0.06312653046846389 Validation Loss = 0.06454585805535316 ——- Epoch 5 ——- Training [160/500 (31%)] Loss: 0.060235 [320/500 (62%)] Loss: 0.068468 [480/500 (94%)] Loss: 0.058079 Loss = 0.06177570053935051 Validation Loss = 0.06303903982043266 ——- Epoch 6 ——- Training [160/500 (31%)] Loss: 0.062826 [320/500 (62%)] Loss: 0.061102 [480/500 (94%)] Loss: 0.063184 Loss = 0.06037520468235017 Validation Loss = 0.06152056068181 9915 ——- Epoch 7 ——- Training [160/500 (31%)] Loss: 0.065317 [320/500 (62%)] Loss: 0.062589 [480/500 (94%)] Loss: 0.052660 Loss = 0.05896557116508 4864 Validation Loss = 0.05994537323713 3025 ——- Epoch 8 ——- Training [160/500 (31%)] Loss: 0.060217 [320/500 (62%)] Loss: 0.054155 [480/500 (94%)] Loss: 0.060423 Loss = 0.05755597549676897 Validation Loss = 0.05846838310360 9085 ——- Epoch 9 ——- Training [160/500 (31%)] Loss: 0.055098 [320/500 (62%)] Loss: 0.054301 [480/500 (94%)] Loss: 0.055336 Loss = 0.05621037518978 1184 Validation Loss = 0.057081046700477596 ——- Epoch 10 ——- Training [160/500 (31%)] Loss: 0.054981 [320/500 (62%)] Loss: 0.056051 [480/500 (94%)] Loss: 0.052508 Loss = 0.05500720822811126 Validation Loss = 0.05586959823966026 ——- Epoch 11 ——- Training [160/500 (31%)] Loss: 0.055954 [320/500 (62%)] Loss: 0.053379 [480/500 (94%)] Loss: 0.053166 Loss = 0.05392904973030091 Validation Loss = 0.054865711778402326 ——- Epoch 12 ——- Training [160/500 (31%)] Loss: 0.051749 [320/500 (62%)] Loss: 0.055551 [480/500 (94%)] Loss: 0.049907 Loss = 0.05309959599375725 Validation Loss = 0.054063108712434776 ——- Epoch 13 ——- Training [160/500 (31%)] Loss: 0.052612 [320/500 (62%)] Loss: 0.046494 [480/500 (94%)] Loss: 0.052068 Loss = 0.05245991396903991 Validation Loss = 0.05358126357197761 ——- Epoch 14 ——- Training [160/500 (31%)] Loss: 0.053961 [320/500 (62%)] Loss: 0.050086 [480/500 (94%)] Loss: 0.051372 Loss = 0.05206913408637047 Validation Loss = 0.05328110814094543 ——- Epoch 15 ——- Training [160/500 (31%)] Loss: 0.050847 [320/500 (62%)] Loss: 0.049110 [480/500 (94%)] Loss: 0.050942 Loss = 0.05179547330737113 Validation Loss = 0.053062806427478786 Time Elapsed: 148.328017 seconds

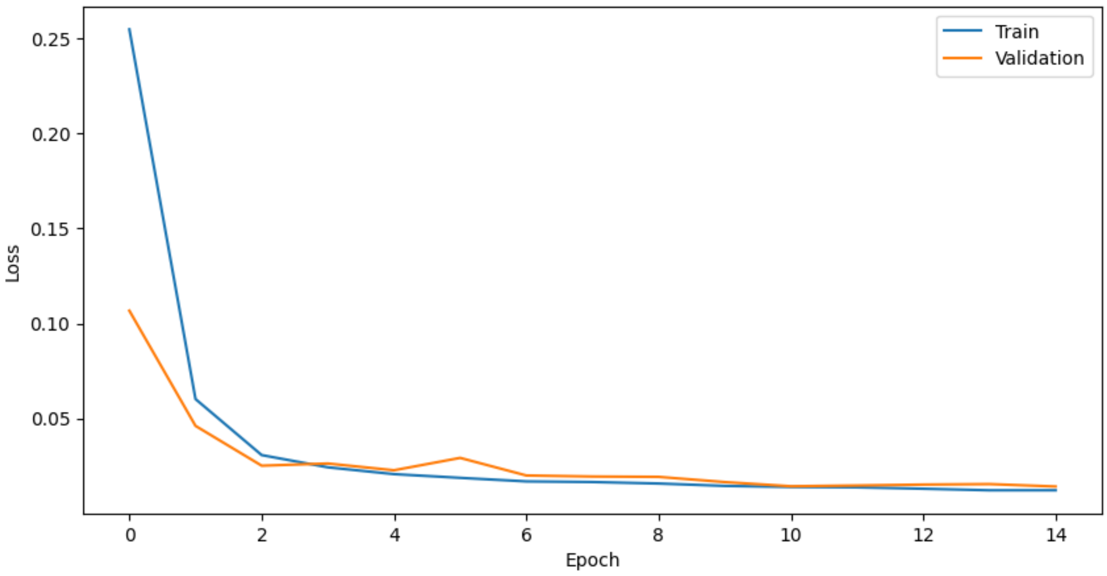
Model unet1 ——- Epoch 1 ——- Training [160/500 (31%)] Loss: 0.541002 [320/500 (62%)] Loss: 0.341874 [480/500 (94%)] Loss: 0.207729 Loss = 0.4563114767074584 Validation Loss = 0.21447854101657865 ——- Epoch 2 ——- Training [160/500 (31%)] Loss: 0.112821 [320/500 (62%)] Loss: 0.076113 [480/500 (94%)] Loss: 0.050758 Loss = 0.10007565313577653 Validation Loss = 0.06028012216091155 ——- Epoch 3 ——- Training [160/500 (31%)] Loss: 0.044538 [320/500 (62%)] Loss: 0.032646 [480/500 (94%)] Loss: 0.033280 Loss = 0.039559854909777634 Validation Loss = 0.039318508058786394 ——- Epoch 4 ——- Training [160/500 (31%)] Loss: 0.035228 [320/500 (62%)] Loss: 0.025473 [480/500 (94%)] Loss: 0.023952 Loss = 0.02674437984824181 Validation Loss = 0.030560681968927382 ——- Epoch 5 ——- Training [160/500 (31%)] Loss: 0.021688 [320/500 (62%)] Loss: 0.019040 [480/500 (94%)] Loss: 0.019648 Loss = 0.02086844703555107 Validation Loss = 0.018918868601322178 ——- Epoch 6 ——- Training [160/500 (31%)] Loss: 0.019820 [320/500 (62%)] Loss: 0.016209 [480/500 (94%)] Loss: 0.015595 Loss = 0.01859696739166975 Validation Loss = 0.02743793532252312 ——- Epoch 7 ——- Training [160/500 (31%)] Loss: 0.016359 [320/500 (62%)] Loss: 0.016228 [480/500 (94%)] Loss: 0.012662 Loss = 0.017059832483530046 Validation Loss = 0.016634457111358643 ——- Epoch 8 ——- Training [160/500 (31%)] Loss: 0.015012 [320/500 (62%)] Loss: 0.015706 [480/500 (94%)] Loss: 0.014862 Loss = 0.014967143960297108 Validation Loss = 0.016177757568657396 ——- Epoch 9 ——- Training [160/500 (31%)] Loss: 0.012811 [320/500 (62%)] Loss: 0.014614 [480/500 (94%)] Loss: 0.011900 Loss = 0.013926139310002326 Validation Loss = 0.01401578828692436 ——- Epoch 10 ——- Training [160/500 (31%)] Loss: 0.013536 [320/500 (62%)] Loss: 0.010962 [480/500 (94%)] Loss: 0.012785 Loss = 0.013397312477231027 Validation Loss = 0.014818869344890118 ——- Epoch 11 ——- Training [160/500 (31%)] Loss: 0.013781 [320/500 (62%)] Loss: 0.012791 [480/500 (94%)] Loss: 0.012741 Loss = 0.012694553941488265 Validation Loss = 0.014038280658423902 ——- Epoch 12 ——- Training [160/500 (31%)] Loss: 0.011948 [320/500 (62%)] Loss: 0.012707 [480/500 (94%)] Loss: 0.010908 Loss = 0.012420875951647757 Validation Loss = 0.014197131283581257 ——- Epoch 13 ——- Training [160/500 (31%)] Loss: 0.012382 [320/500 (62%)] Loss: 0.013841 [480/500 (94%)] Loss: 0.010451 Loss = 0.013549172073602674 Validation Loss = 0.015963094681501387 ——- Epoch 14 ——- Training [160/500 (31%)] Loss: 0.012402 [320/500 (62%)] Loss: 0.010978 [480/500 (94%)] Loss: 0.011596 Loss = 0.011802225790917874 Validation Loss = 0.011506651714444416 ——- Epoch 15 ——- Training [160/500 (31%)] Loss: 0.011631 [320/500 (62%)] Loss: 0.009291 [480/500 (94%)] Loss: 0.009569 Loss = 0.010932175703346732 Validation Loss = 0.014581790193915369 Time Elapsed: 279.75737 seconds

Model unet2 ——- Epoch 1 ——- Training /tmp/ipykernel_1805356/1503528447.py:43: DeprecationWarning: Starting with ImageIO v3 the behavior of this function will switch to that of iio.v3.imread. To keep the current behavior (and make this warning disappear) use `import imageio.v2 as imageio` or call `imageio.v2.imread` directly. input_img = np.asarray(imageio.imread(input_img_path)) /tmp/ipykernel_1805356/1503528447.py:44: DeprecationWarning: Starting with ImageIO v3 the behavior of this function will switch to that of iio.v3.imread. To keep the current behavior (and make this warning disappear) use `import imageio.v2 as imageio` or call `imageio.v2.imread` directly. seg_img = np.asarray(imageio.imread(seg_img_path)) [160/500 (31%)] Loss: 0.351242 [320/500 (62%)] Loss: 0.219544 [480/500 (94%)] Loss: 0.130769 Loss = 0.29853802812099445 Validation Loss = 0.11812746733427047 ——- Epoch 2 ——- Training [160/500 (31%)] Loss: 0.071831 [320/500

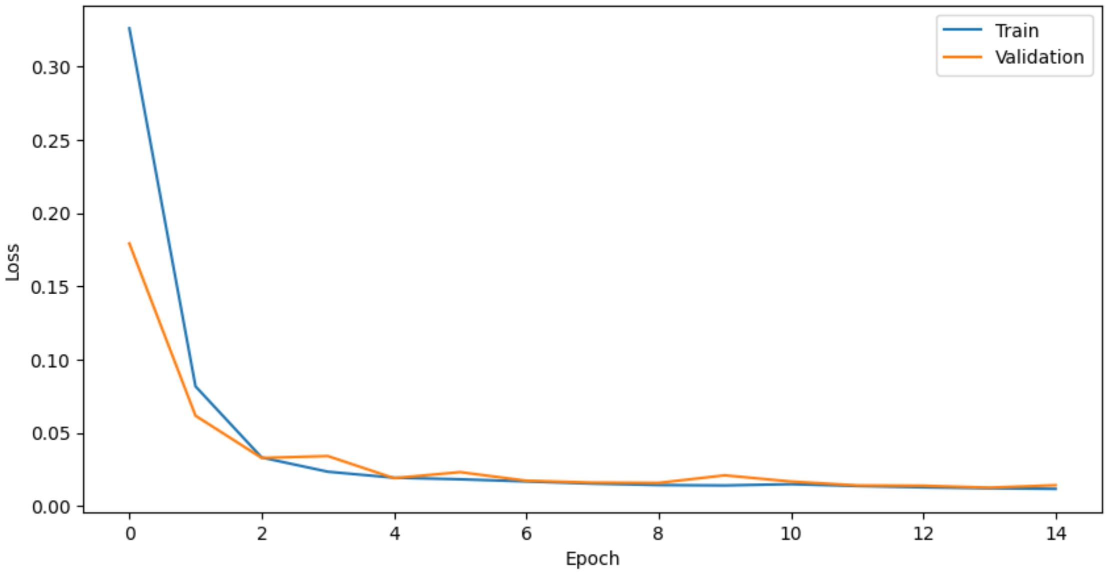
(62%)] Loss: 0.049598 [480/500 (94%)] Loss: 0.039854 Loss = 0.06579984581470488 Validation Loss = 0.07301077187061308 ——- Epoch 3 ——- Training [160/500 (31%)] Loss: 0.030116 [320/500 (62%)] Loss: 0.029837 [480/500 (94%)] Loss: 0.026317 Loss = 0.030201540857553482 Validation Loss = 0.028409226983785632 ——- Epoch 4 ——- Training [160/500 (31%)] Loss: 0.021623 [320/500 (62%)] Loss: 0.019726 [480/500 (94%)] Loss: 0.021222 Loss = 0.02182533159852027 Validation Loss = 0.01960519626736641 ——- Epoch 5 ——- Training [160/500 (31%)] Loss: 0.022952 [320/500 (62%)] Loss: 0.016570 [480/500 (94%)] Loss: 0.018865 Loss = 0.01819968333840371 Validation Loss = 0.018712050467729568 ——- Epoch 6 ——- Training [160/500 (31%)] Loss: 0.016335 [320/500 (62%)] Loss: 0.014721 [480/500 (94%)] Loss: 0.013551 Loss = 0.01624171040952206 Validation Loss = 0.017159371599555016 ——- Epoch 7 ——- Training [160/500 (31%)] Loss: 0.012402 [320/500 (62%)] Loss: 0.012695 [480/500 (94%)] Loss: 0.012964 Loss = 0.013751231342554095 Validation Loss = 0.014211194813251496 ——- Epoch 8 ——- Training [160/500 (31%)] Loss: 0.008781 [320/500 (62%)] Loss: 0.007258 [480/500 (94%)] Loss: 0.006773 Loss = 0.009513318799436092 Validation Loss = 0.026332326382398605 ——- Epoch 9 ——- Training [160/500 (31%)] Loss: 0.008455 [320/500 (62%)] Loss: 0.016871 [480/500 (94%)] Loss: 0.009092 Loss = 0.009678129442036148 Validation Loss = 0.03341002464294433 ——- Epoch 10 ——- Training [160/500 (31%)] Loss: 0.008897 [320/500 (62%)] Loss: 0.004449 [480/500 (94%)] Loss: 0.006177 Loss = 0.008082839656621217 Validation Loss = 0.008207859434187413 ——- Epoch 11 ——- Training [160/500 (31%)] Loss: 0.003717 [320/500 (62%)] Loss: 0.005286 [480/500 (94%)] Loss: 0.003386 Loss = 0.005083623941987752 Validation Loss = 0.005346654523164034 ——- Epoch 12 ——- Training [160/500 (31%)] Loss: 0.002605 [320/500 (62%)] Loss: 0.004493 [480/500 (94%)] Loss: 0.003627 Loss = 0.0038848194908350704 Validation Loss = 0.009582859668880702 ——- Epoch 13 ——- Training [160/500 (31%)] Loss: 0.004453 [320/500 (62%)] Loss: 0.002858 [480/500 (94%)] Loss: 0.002010 Loss = 0.003335556235164404 Validation Loss = 0.0042789016850292684 ——- Epoch 14 ——- Training [160/500 (31%)] Loss: 0.003097 [320/500 (62%)] Loss: 0.003206 [480/500 (94%)] Loss: 0.003922 Loss = 0.003212730635888875 Validation Loss = 0.005679447315633298 ——- Epoch 15 ——- Training [160/500 (31%)] Loss: 0.002215 [320/500 (62%)] Loss: 0.003058 [480/500 (94%)] Loss: 0.003217 Loss = 0.0026228907369077207 Validation Loss = 0.008905640468001366 Time Elapsed: 230.590325 seconds

[7]:
```
## Losses

### convnet1
# im = Image.open("convnet1loss.png")
# print("Loss plot for unet2x3 with Dice Loss")
# display(im)

### unet1
im = Image.open("unet1loss.png")
print("Loss for unet1")
display(im)

### unet2
im = Image.open("unet2loss.png")
print("Loss for unet2")
display(im)
```

```
Loss for unet1
```



```
Loss for unet2
```



1.3 Dice Scores

1.3.1 convnet1

Average Dice score for threshold 0 = 0.018990195978441187 Time Elapsed: 0.435352 seconds Average Dice score for threshold 0.02 = 0.0519058588762l9145 Time Elapsed: 0.895155 seconds Average Dice score for threshold 0.08 = 0.0 Time Elapsed: 1.171124 seconds Average Dice score for threshold

0.13 = 0.0 Time Elapsed: 1.441698 seconds Average Dice score for threshold 0.18 = 0.0 Time Elapsed: 1.687684 seconds Average Dice score for threshold 0.23 = 0.0 Time Elapsed: 1.99926 seconds Average Dice score for threshold 0.28 = 0.0 Time Elapsed: 2.499664 seconds Average Dice score for threshold 0.34 = 0.0 Time Elapsed: 2.954805 seconds Average Dice score for threshold 0.39 = 0.0 Time Elapsed: 3.287261 seconds Average Dice score for threshold 0.44 = 0.0 Time Elapsed: 3.637824 seconds Average Dice score for threshold 0.49 = 0.0 Time Elapsed: 3.920945 seconds Average Dice score for threshold 0.53 = 0.0 Time Elapsed: 4.38681 seconds Average Dice score for threshold 0.58 = 0.0 Time Elapsed: 4.884436 seconds Average Dice score for threshold 0.63 = 0.0 Time Elapsed: 5.329861 seconds Average Dice score for threshold 0.68 = 0.0 Time Elapsed: 5.684441 seconds Average Dice score for threshold 0.73 = 0.0 Time Elapsed: 6.130636 seconds Average Dice score for threshold 0.78 = 0.0 Time Elapsed: 6.484496 seconds Average Dice score for threshold 0.83 = 0.0 Time Elapsed: 6.871082 seconds Average Dice score for threshold 0.88 = 0.0 Time Elapsed: 7.486235 seconds Average Dice score for threshold 0.93 = 0.0 Time Elapsed: 8.110315 seconds Average Dice score for threshold 1 = 0.0 Time Elapsed: 8.694349 seconds

1.3.2 unet1

Average Dice score for threshold 0 = 0.01899019597841187 Time Elapsed: 0.346673 seconds Average Dice score for threshold 0.02 = 0.6611855673260862 Time Elapsed: 0.76396 seconds Average Dice score for threshold 0.08 = 0.8086888378842988 Time Elapsed: 1.181438 seconds Average Dice score for threshold 0.13 = 0.8600695017439864 Time Elapsed: 1.567126 seconds Average Dice score for threshold 0.18 = 0.8843644372827151 Time Elapsed: 2.099805 seconds Average Dice score for threshold 0.23 = 0.9018475393057165 Time Elapsed: 2.616634 seconds Average Dice score for threshold 0.28 = 0.9145380742638348 Time Elapsed: 3.107602 seconds Average Dice score for threshold 0.34 = 0.9141153010050133 Time Elapsed: 3.481116 seconds Average Dice score for threshold 0.39 = 0.0 Time Elapsed: 3.980063 seconds Average Dice score for threshold 0.44 = 0.0 Time Elapsed: 4.415424 seconds Average Dice score for threshold 0.49 = 0.0 Time Elapsed: 4.801767 seconds Average Dice score for threshold 0.53 = 0.0 Time Elapsed: 5.294665 seconds Average Dice score for threshold 0.58 = 0.0 Time Elapsed: 5.701532 seconds Average Dice score for threshold 0.63 = 0.0 Time Elapsed: 6.189852 seconds Average Dice score for threshold 0.68 = 0.0 Time Elapsed: 6.713102 seconds Average Dice score for threshold 0.73 = 0.0 Time Elapsed: 7.096272 seconds Average Dice score for threshold 0.78 = 0.0 Time Elapsed: 7.6041 seconds Average Dice score for threshold 0.83 = 0.0 Time Elapsed: 8.200991 seconds Average Dice score for threshold 0.88 = 0.0 Time Elapsed: 8.816494 seconds Average Dice score for threshold 0.93 = 0.0 Time Elapsed: 9.194436 seconds Average Dice score for threshold 1 = 0.0 Time Elapsed: 9.603837 seconds

1.3.3 unet2

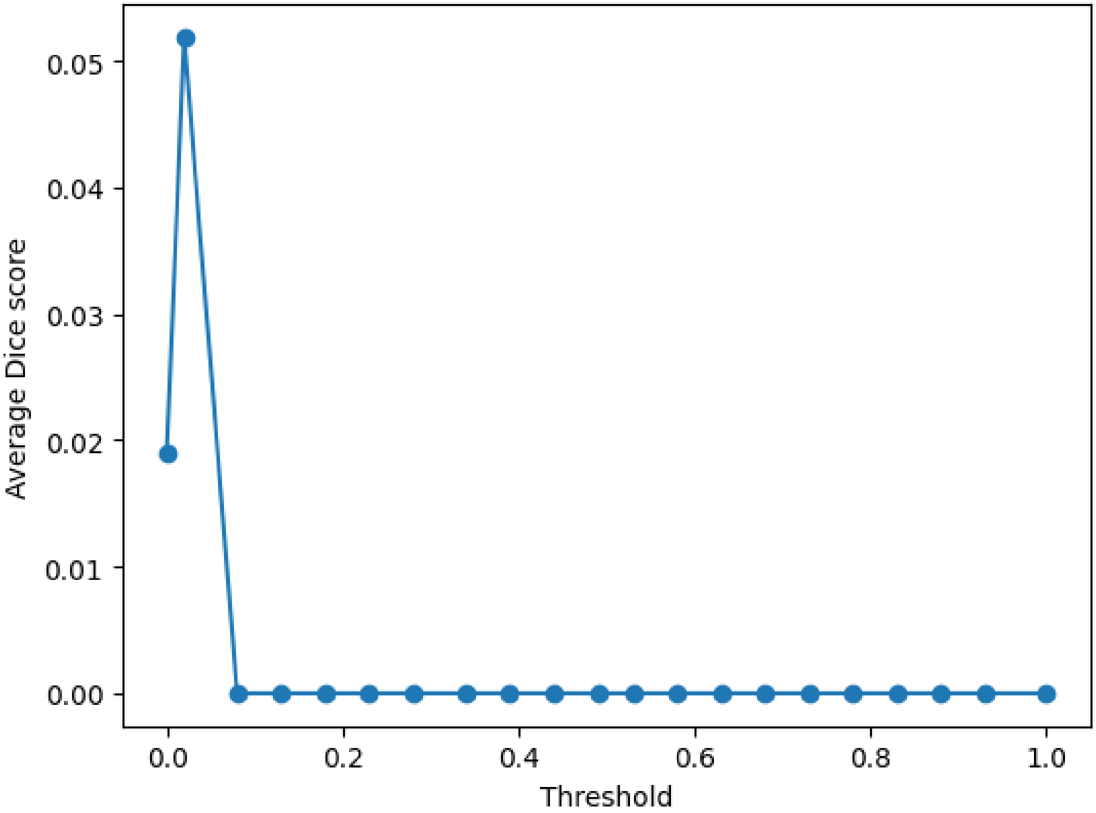
Average Dice score for threshold 0 = 0.01899019597841187 Time Elapsed: 0.382003 seconds Average Dice score for threshold 0.02 = 0.8530226324578648 Time Elapsed: 0.81599 seconds Average Dice score for threshold 0.08 = 0.9336461078875524 Time Elapsed: 1.145219 seconds Average Dice score for threshold 0.13 = 0.9439456901862191 Time Elapsed: 1.875835 seconds Average Dice score for threshold 0.18 = 0.9485967646384225 Time Elapsed: 2.159095 seconds Average Dice score for threshold 0.23 = 0.9486601366270413 Time Elapsed: 2.574651 seconds Average Dice score for threshold 0.28 = 0.9445540231534234 Time Elapsed: 3.019706 seconds Average Dice score for threshold 0.34 = 0.9307405655496741 Time Elapsed: 3.427743 seconds Average Dice score for threshold 0.39 = 0.0 Time Elapsed: 4.226215 seconds Average Dice score for threshold 0.44 = 0.0 Time Elapsed: 4.886487 seconds Average Dice score for threshold 0.49 = 0.0 Time Elapsed:

```
5.255083 seconds Average Dice score for threshold 0.53 = 0.0 Time Elapsed: 5.553261 seconds
Average Dice score for threshold 0.58 = 0.0 Time Elapsed: 5.888603 seconds Average Dice score
for threshold 0.63 = 0.0 Time Elapsed: 6.540022 seconds Average Dice score for threshold 0.68 =
0.0 Time Elapsed: 6.961455 seconds Average Dice score for threshold 0.73 = 0.0 Time Elapsed:
7.322756 seconds Average Dice score for threshold 0.78 = 0.0 Time Elapsed: 7.586936 seconds
Average Dice score for threshold 0.83 = 0.0 Time Elapsed: 8.021103 seconds Average Dice score for
threshold 0.88 = 0.0 Time Elapsed: 8.460054 seconds Average Dice score for threshold 0.93 = 0.0
Time Elapsed: 8.817248 seconds Average Dice score for threshold 1 = 0.0 Time Elapsed: 9.236355
seconds
```

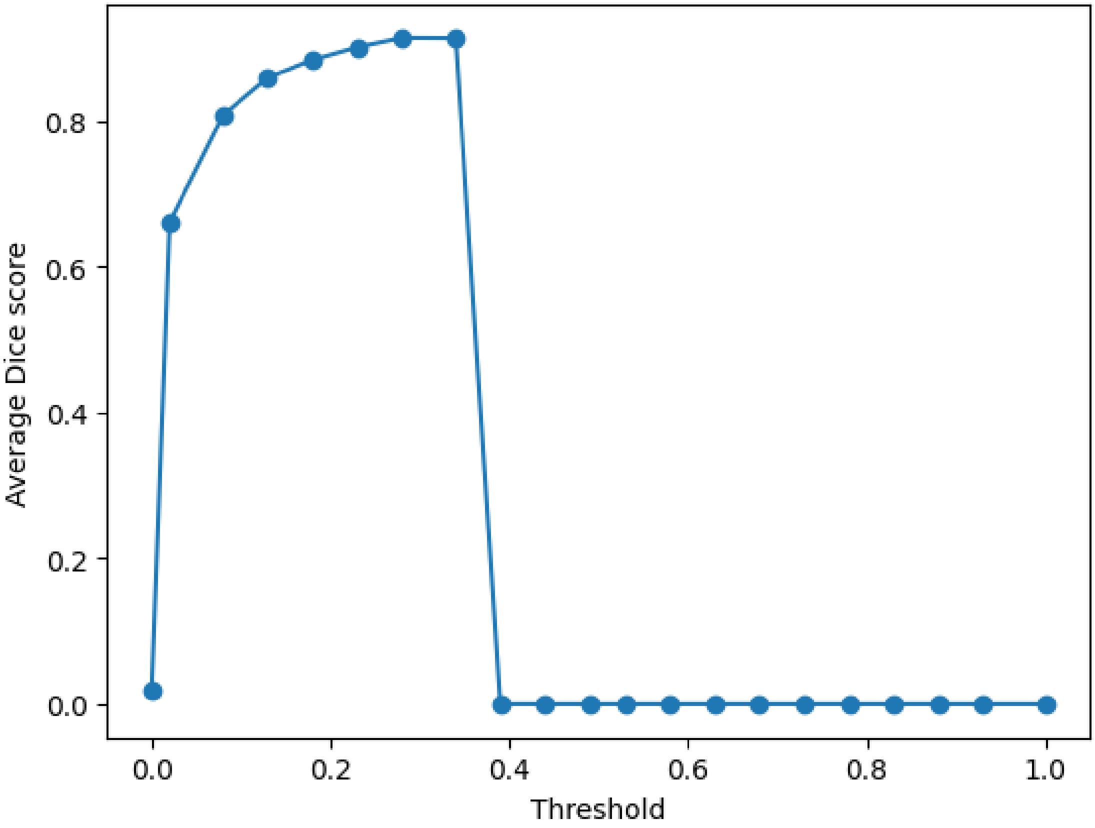
[9]:

```python
#Dice Scores
### convnet1
im = Image.open("convnet1dice.png")
print("Results for convenet1")
display(im)

### unet1
im = Image.open("unet1dice.png")
print("Results for unet1")
display(im)

### unet2
im = Image.open("unet2dice.png")
print("Results for unet2")
display(im)
```

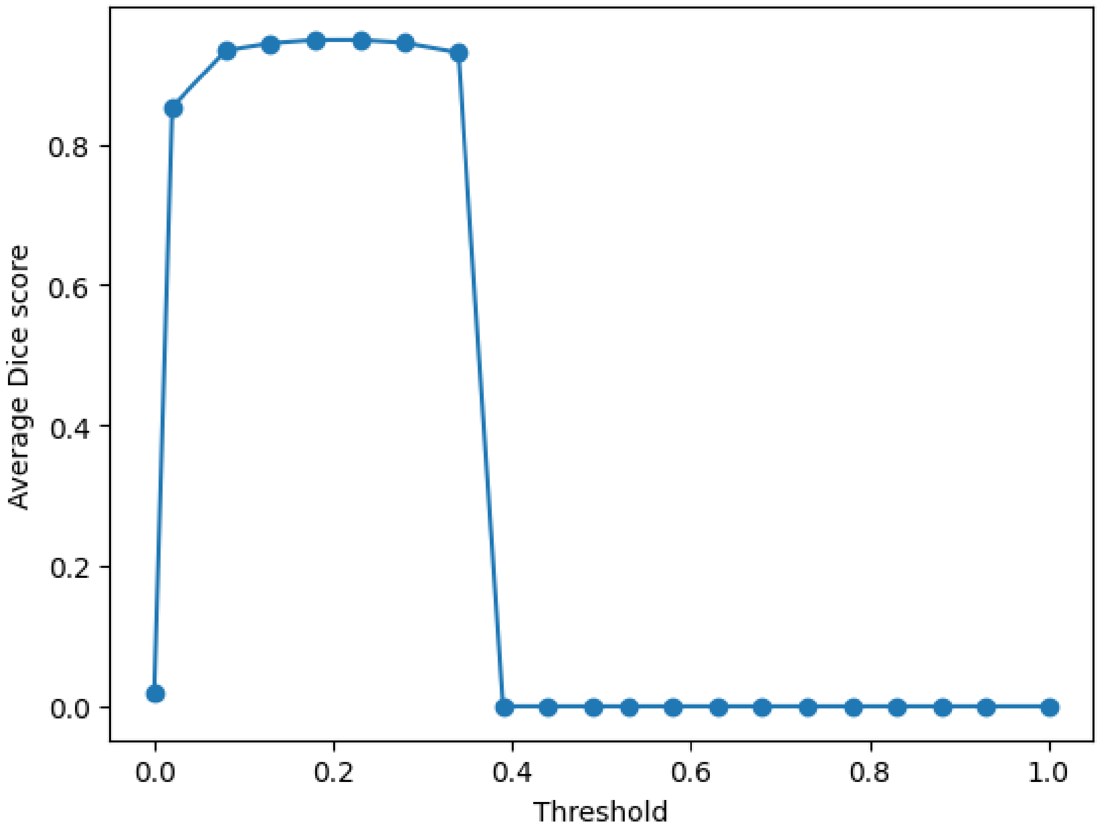
```
Results for convenet1
```



```
Results for unet1
```



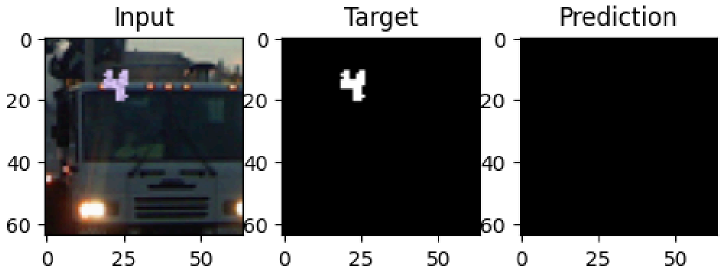
```
Results for unet2
```



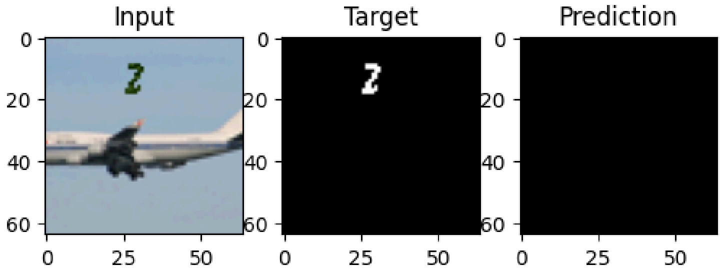
[8]:
```
#Results

### convnet1
im = Image.open("convnet1results.png")
print("Results for convenet1")
display(im)

### unet1
im = Image.open("unet1results.png")
print("Results for unet1")
display(im)

### unet2
im = Image.open("unet2results.png")
print("Results for unet2")
display(im)
```

```
Results for convenet1
```

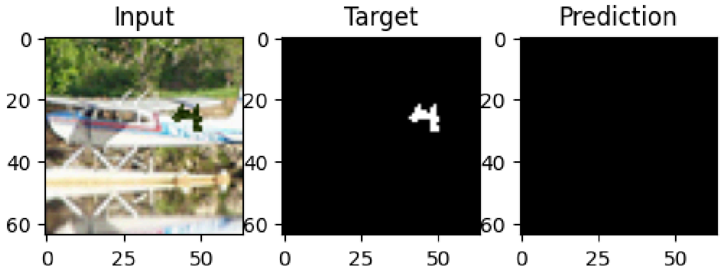
im19.jpg - Dice = 0.00



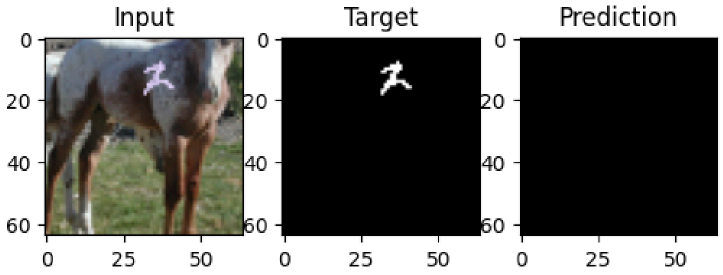
im2.jpg - Dice = 0.00



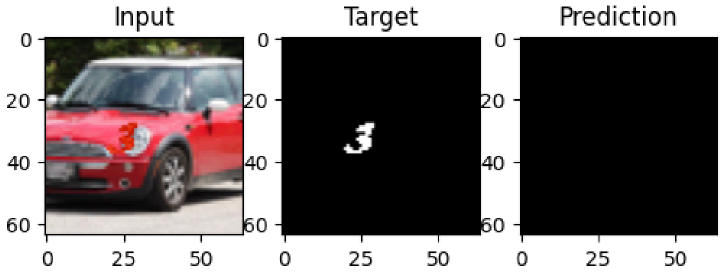
im62.jpg - Dice = 0.00



im44.jpg - Dice = 0.00

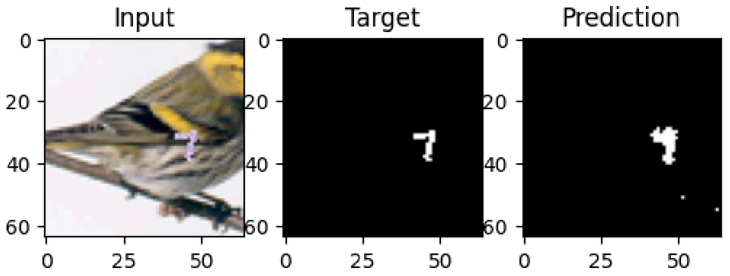


im7.jpg - Dice = 0.00

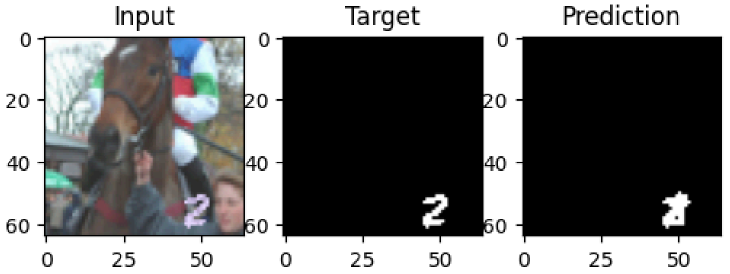


```
Results for unet1
```

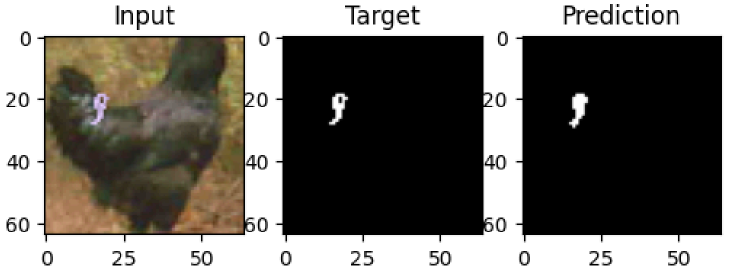
im51.jpg - Dice = 0.68



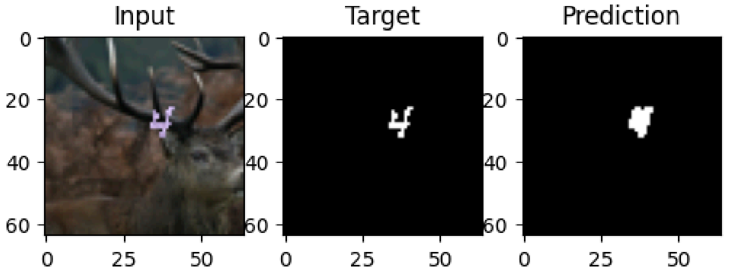
im66.jpg - Dice = 0.84



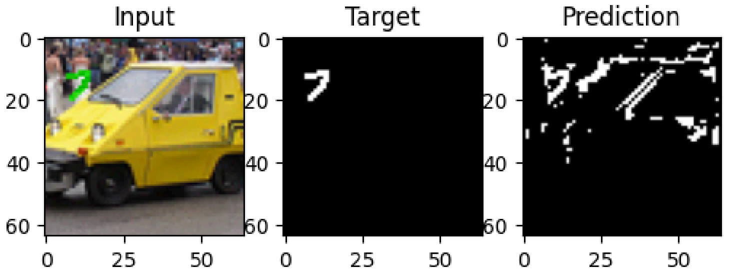
im60.jpg - Dice = 0.89



im82.jpg - Dice = 0.75

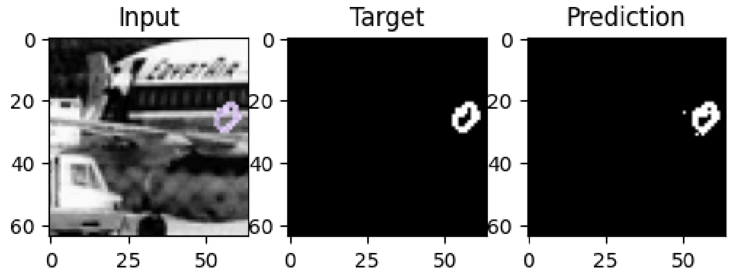


im63.jpg - Dice = 0.22

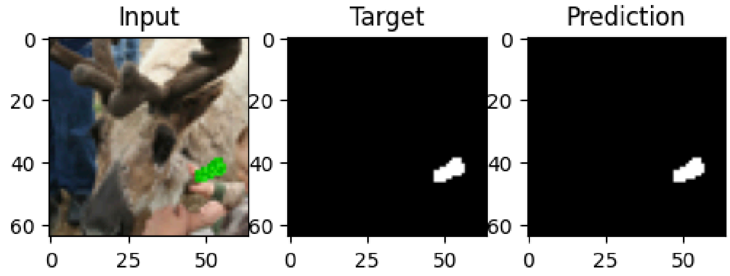


```
Results for unet2
```

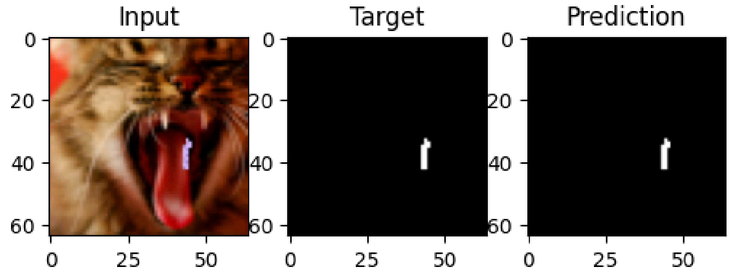
im1.jpg - Dice = 0.94



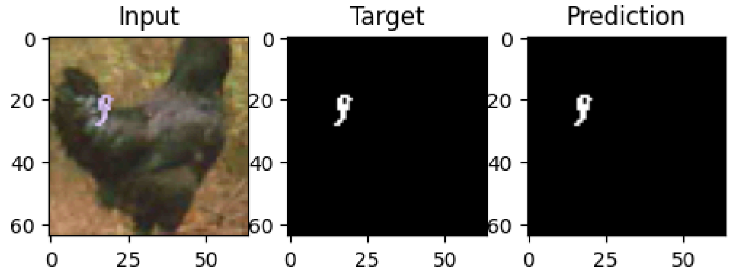
im5.jpg - Dice = 0.99



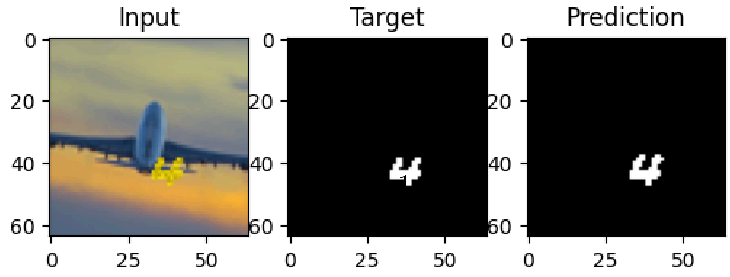
im31.jpg - Dice = 1.00



im60.jpg - Dice = 1.00



im77.jpg - Dice = 0.93



1.4 A.2 Dice training

A. Dice Training

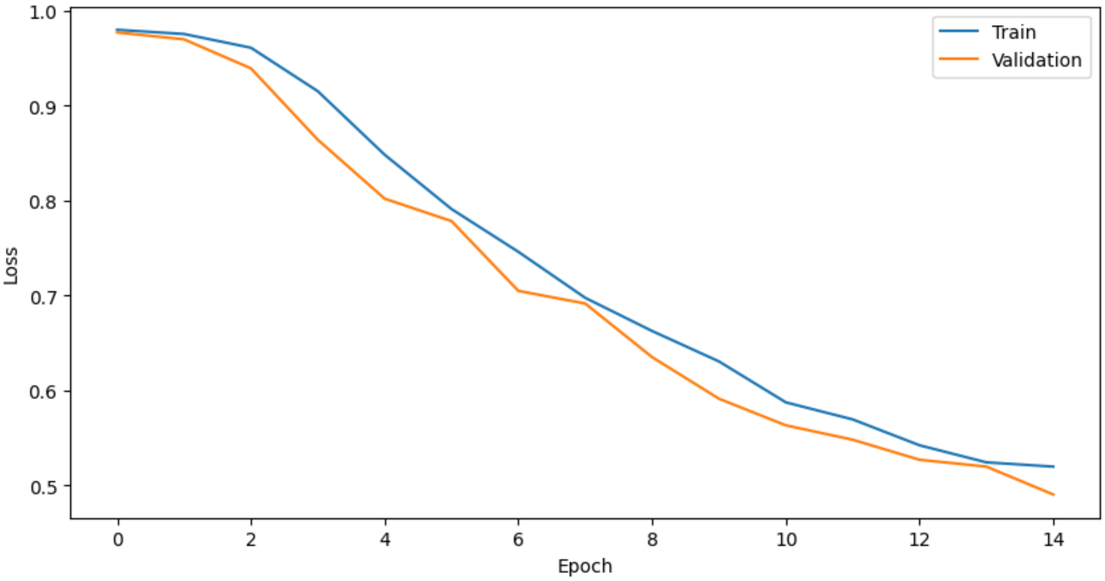
[ ]:
```
# PyTorch
class DiceLoss(nn.Module):
    def __init__(self, weight=None, size_average=True):
        super(DiceLoss, self).__init__()

    def forward(self, inputs, targets, smooth=1):
        # Comment out if your model contains a sigmoid or equivalent activation␣
 ↪layer
        inputs = F.sigmoid(inputs)

        # Flatten label and prediction tensors
        inputs = inputs.view(-1)
        targets = targets.view(-1)

        intersection = (inputs * targets).sum()
        dice = (2. * intersection + smooth) / (inputs.sum() + targets.sum() +␣
 ↪smooth)
        return 1 - dice
```

convnet1 [160/500 (31%)] Loss: 0.979829 [320/500 (62%)] Loss: 0.980024 [480/500 (94%)] Loss: 0.982068 Loss = 0.9797881937026978 Validation Loss = 0.976984281539917 ——- Epoch 2 ——- Training [160/500 (31%)] Loss: 0.976605 [320/500 (62%)] Loss: 0.975146 [480/500 (94%)] Loss: 0.971147 Loss = 0.9753550572395325 Validation Loss = 0.9697327494621276 ——- Epoch 3 ——- Training [160/500 (31%)] Loss: 0.968970 [320/500 (62%)] Loss: 0.949477 [480/500 (94%)] Loss: 0.958182 Loss = 0.9609024329185484 Validation Loss = 0.9392189359664916 ——- Epoch 4 ——- Training [160/500 (31%)] Loss: 0.894148 [320/500 (62%)] Loss: 0.926041 [480/500 (94%)] Loss: 0.898516 Loss = 0.9151358699798584 Validation Loss = 0.8640498733520507 ——- Epoch 5 ——- Training [160/500 (31%)] Loss: 0.905148 [320/500 (62%)] Loss: 0.867190 [480/500 (94%)] Loss: 0.841003 Loss = 0.8482734985351562 Validation Loss = 0.8020081591606141 ——- Epoch 6 ——- Training [160/500 (31%)] Loss: 0.759461 [320/500 (62%)] Loss: 0.856176 [480/500 (94%)] Loss: 0.815827 Loss = 0.7911010355949402 Validation Loss = 0.7784179282188415 ——- Epoch 7 ——- Training [160/500 (31%)] Loss: 0.792351 [320/500 (62%)] Loss: 0.658946 [480/500 (94%)] Loss: 0.813460 Loss = 0.7460345144271849 Validation Loss = 0.7049472570419312 ——- Epoch 8 ——- Training [160/500 (31%)] Loss: 0.667735 [320/500 (62%)] Loss: 0.774076 [480/500 (94%)] Loss: 0.588880 Loss = 0.6976102838516236 Validation Loss = 0.6916141295433045 ——- Epoch 9 ——- Training [160/500 (31%)] Loss: 0.713455 [320/500 (62%)] Loss: 0.565805 [480/500 (94%)] Loss: 0.688973 Loss = 0.6626938018798827 Validation Loss = 0.6352146530151367 ——- Epoch 10 ——- Training [160/500 (31%)] Loss: 0.580728 [320/500 (62%)] Loss: 0.626472 [480/500 (94%)] Loss: 0.628440 Loss = 0.6306771454811095 Validation Loss = 0.5914047956466675 ——- Epoch 11 ——- Training [160/500 (31%)] Loss: 0.520337 [320/500 (62%)] Loss: 0.498523 [480/500 (94%)] Loss: 0.495688 Loss = 0.5875785794258117 Validation Loss = 0.563384087085724 ——- Epoch 12 ——- Training [160/500 (31%)] Loss: 0.561008 [320/500 (62%)] Loss: 0.491720 [480/500 (94%)]

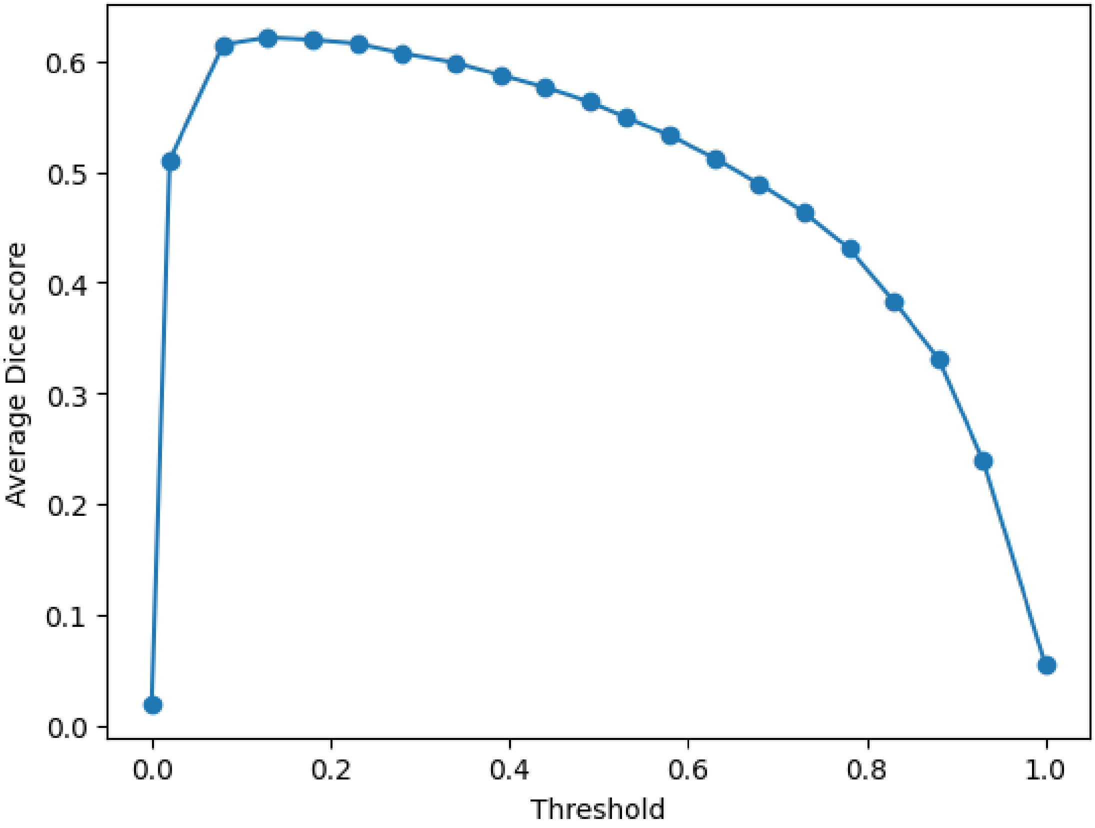
```
Loss: 0.401155 Loss = 0.5696206860542299 Validation Loss = 0.5482477259635925 ——- Epoch 13
——- Training [160/500 (31%)] Loss: 0.532170 [320/500 (62%)] Loss: 0.629674 [480/500 (94%)]
Loss: 0.656899 Loss = 0.5423868160247803 Validation Loss = 0.527144660949707 ——- Epoch 14
——- Training [160/500 (31%)] Loss: 0.585322 [320/500 (62%)] Loss: 0.632129 [480/500 (94%)]
Loss: 0.418641 Loss = 0.5244079308509826 Validation Loss = 0.5199232149124146 ——- Epoch 15
——- Training [160/500 (31%)] Loss: 0.538549 [320/500 (62%)] Loss: 0.674159 [480/500 (94%)]
Loss: 0.573090 Loss = 0.5199228734970093 Validation Loss = 0.4905547332763671 Time Elapsed:
33.487466 seconds
```

[11]:
```python
im = Image.open("dtconvnet1loss.png")
print("Loss for dtconvenet1")
display(im)

im = Image.open("dtconvnet1dice.png")
print("Dice Scores for dt convnet1")
display(im)
```

```
Loss for dtconvenet1
```



```
Dice Scores for dt convnet1
```

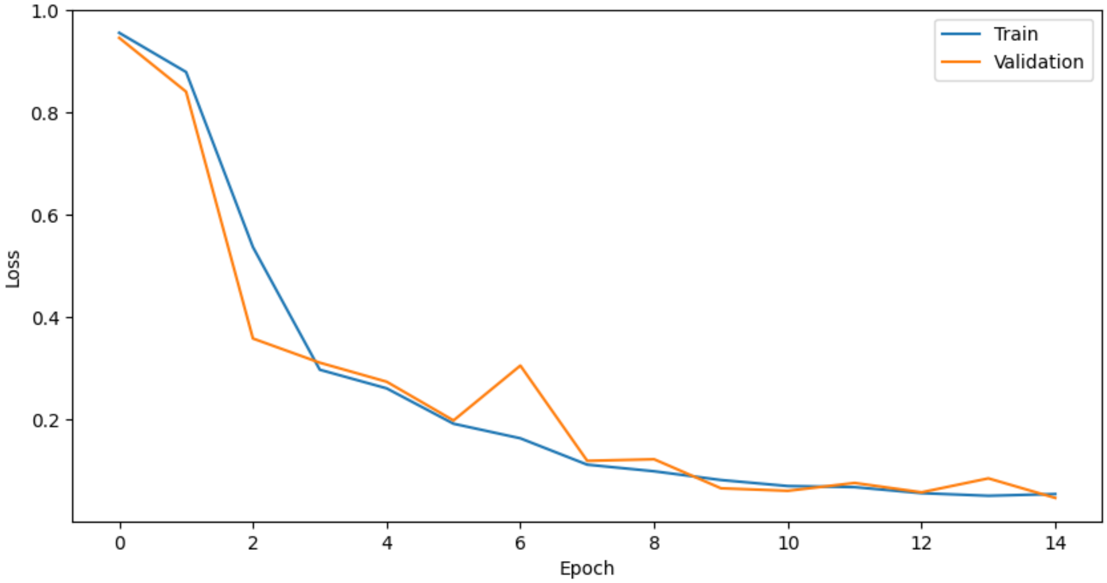


unet2 [160/500 (31%)] Loss: 0.958056 [320/500 (62%)] Loss: 0.953678 [480/500 (94%)] Loss: 0.936715 Loss = 0.9560103998184206 Validation Loss = 0.946160271167755 ——- Epoch 2 ——- Training [160/500 (31%)] Loss: 0.917483 [320/500 (62%)] Loss: 0.863940 [480/500 (94%)] Loss: 0.785416 Loss = 0.8792879257202149 Validation Loss = 0.8408000445365906 ——- Epoch 3 ——- Training [160/500 (31%)] Loss: 0.595686 [320/500 (62%)] Loss: 0.452674 [480/500 (94%)] Loss: 0.420730 Loss = 0.536676932811737 Validation Loss = 0.35777306079864507 ——- Epoch 4 ——- Training [160/500 (31%)] Loss: 0.345628 [320/500 (62%)] Loss: 0.334957 [480/500 (94%)] Loss: 0.293341 Loss = 0.2966760363578797 Validation Loss = 0.3104406952857971 ——- Epoch 5 ——- Training [160/500 (31%)] Loss: 0.326110 [320/500 (62%)] Loss: 0.219451 [480/500 (94%)] Loss: 0.272847 Loss = 0.26020934009552 Validation Loss = 0.2731850862503052 ——- Epoch 6 ——- Training [160/500 (31%)] Loss: 0.122582 [320/500 (62%)] Loss: 0.184855 [480/500 (94%)] Loss: 0.149088 Loss = 0.19099617338180547 Validation Loss = 0.1973887276649475 ——- Epoch 7 ——- Training [160/500 (31%)] Loss: 0.136033 [320/500 (62%)] Loss: 0.192360 [480/500 (94%)] Loss: 0.147689 Loss = 0.1624254670143127 Validation Loss = 0.3047628474235534 ——- Epoch 8 ——- Training [160/500 (31%)] Loss: 0.067044 [320/500 (62%)] Loss: 0.087412 [480/500 (94%)] Loss: 0.115869 Loss = 0.11086428594589233 Validation Loss = 0.1183558988571167 ——- Epoch 9 ——- Training [160/500 (31%)] Loss: 0.058707 [320/500 (62%)] Loss: 0.144275 [480/500 (94%)] Loss: 0.071320 Loss = 0.09793889093399046 Validation Loss = 0.1214747738838196 ——- Epoch 10 ——- Training [160/500 (31%)] Loss: 0.075348 [320/500 (62%)] Loss: 0.088178 [480/500 (94%)] Loss: 0.059632 Loss = 0.08083113145828248 Validation Loss = 0.06465006589889526 ——- Epoch 11 —

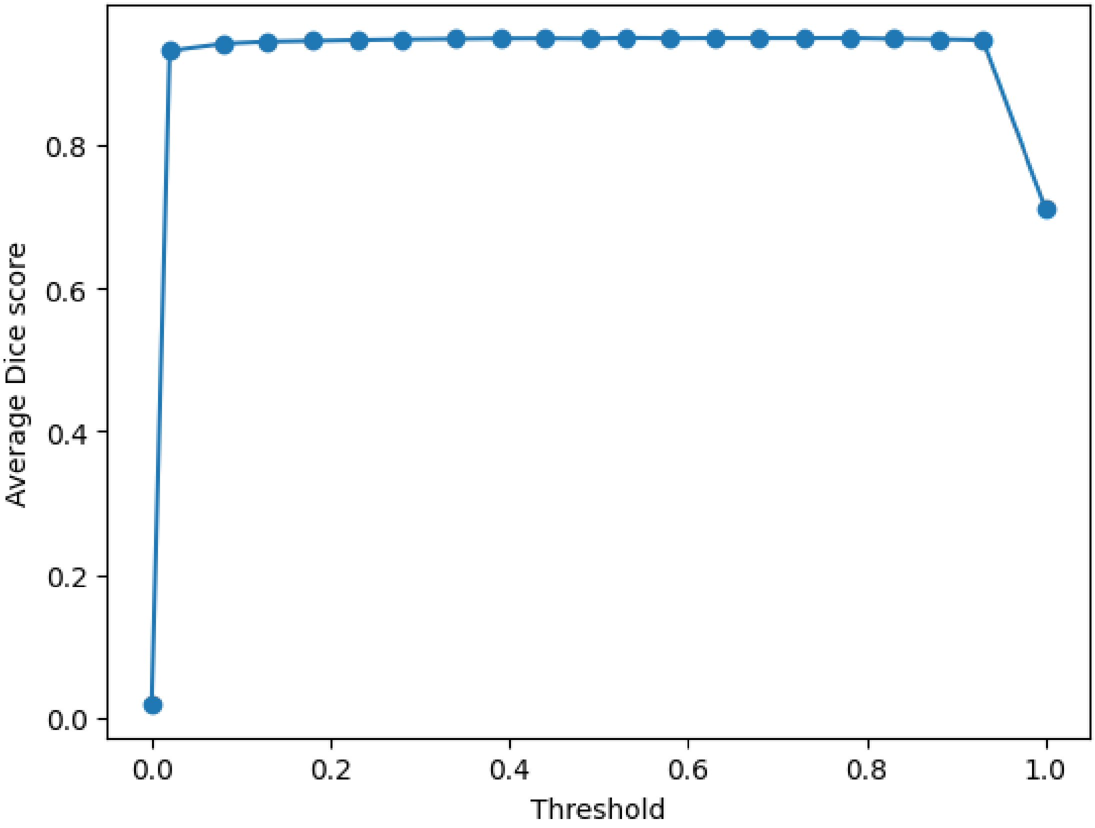
```
—- Training [160/500 (31%)] Loss: 0.068458 [320/500 (62%)] Loss: 0.080857 [480/500 (94%)] Loss:
0.101467 Loss = 0.06903580713272094 Validation Loss = 0.059668593406677246 ——- Epoch 12
——- Training [160/500 (31%)] Loss: 0.103204 [320/500 (62%)] Loss: 0.023041 [480/500 (94%)]
Loss: 0.063764 Loss = 0.06687219762802124 Validation Loss = 0.07510993003845216 ——- Epoch
13 ——- Training [160/500 (31%)] Loss: 0.084688 [320/500 (62%)] Loss: 0.068698 [480/500 (94%)]
Loss: 0.045791 Loss = 0.05509238576889038 Validation Loss = 0.05678844928741455 ——- Epoch
14 ——- Training [160/500 (31%)] Loss: 0.006127 [320/500 (62%)] Loss: 0.021517 [480/500 (94%)]
Loss: 0.031586 Loss = 0.049987637042999265 Validation Loss = 0.0840649938583374 ——- Epoch
15 ——- Training [160/500 (31%)] Loss: 0.013500 [320/500 (62%)] Loss: 0.020657 [480/500 (94%)]
Loss: 0.076565 Loss = 0.05333372497558594 Validation Loss = 0.04574723482131957 Time Elapsed:
49.647085 seconds
```

[12]:
```
im = Image.open("dtunet2loss.png")
print("Loss for dt unet2")
display(im)

im = Image.open("dtunet2dice.png")
print("Dice Scores for dt unet2")
display(im)
```

```
Loss for dt unet2
```



```
Dice Scores for dt unet2
```



B. Dice Training and Balancing

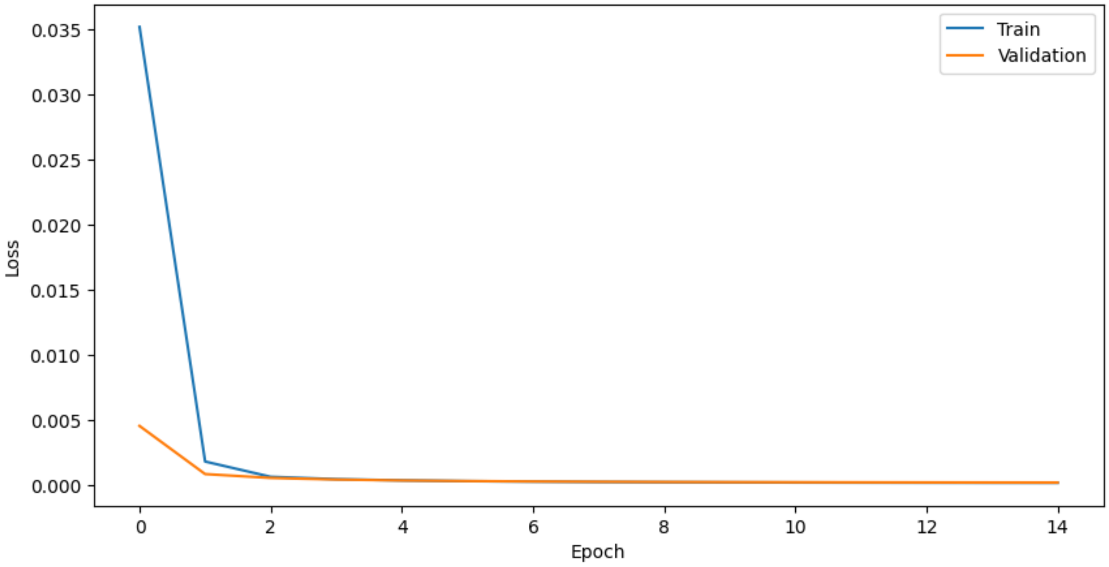
convnet1 [160/500 (31%)] Loss: 0.046818 [320/500 (62%)] Loss: 0.017702 [480/500 (94%)] Loss: 0.005975 Loss = 0.03520873909816147 Validation Loss = 0.004583979453891516 ——- Epoch 2 ——- Training [160/500 (31%)] Loss: 0.002081 [320/500 (62%)] Loss: 0.001160 [480/500 (94%)] Loss: 0.000995 Loss = 0.0018518857578746978 Validation Loss = 0.0008886442077346147 ——- Epoch 3 ——- Training [160/500 (31%)] Loss: 0.000753 [320/500 (62%)] Loss: 0.000746 [480/500 (94%)] Loss: 0.000582 Loss = 0.0006791530665941539 Validation Loss = 0.0005925678764469922 ——- Epoch 4 ——- Training [160/500 (31%)] Loss: 0.000476 [320/500 (62%)] Loss: 0.000549 [480/500 (94%)] Loss: 0.000322 Loss = 0.0004953589984215797 Validation Loss = 0.0004705115454271436 ——- Epoch 5 ——- Training [160/500 (31%)] Loss: 0.000407 [320/500 (62%)] Loss: 0.000339 [480/500 (94%)] Loss: 0.000336 Loss = 0.0004040882363915443 Validation Loss = 0.0003987935977056623 ——- Epoch 6 ——- Training [160/500 (31%)] Loss: 0.000408 [320/500 (62%)] Loss: 0.000299 [480/500 (94%)] Loss: 0.000355 Loss = 0.0003474604708608239 Validation Loss = 0.0003535524974875154 ——- Epoch 7 ——- Training [160/500 (31%)] Loss: 0.000349 [320/500 (62%)] Loss: 0.000252 [480/500 (94%)] Loss: 0.000338 Loss = 0.0003121713059954346 Validation Loss = 0.00032196616870351137 ——- Epoch 8 ——- Training [160/500 (31%)] Loss: 0.000274 [320/500 (62%)] Loss: 0.000337 [480/500 (94%)] Loss: 0.000229 Loss = 0.00028458450012840337 Validation Loss = 0.0002994953014422208 ——- Epoch 9 ——- Training [160/500 (31%)] Loss: 0.000457 [320/500 (62%)] Loss: 0.000209 [480/500 (94%)] Loss: 0.000310 Loss = 0.00026690181111916906

```
Validation Loss = 0.00028284396044909955 ——- Epoch 10 ——- Training [160/500 (31%)]
Loss: 0.000327 [320/500 (62%)] Loss: 0.000253 [480/500 (94%)] Loss: 0.000241 Loss =
0.0002527851429767906 Validation Loss = 0.00027041305846069007 ——- Epoch 11 ——- Training
[160/500 (31%)] Loss: 0.000229 [320/500 (62%)] Loss: 0.000271 [480/500 (94%)] Loss: 0.000243
Loss = 0.00024152004672214385 Validation Loss = 0.0002607164724031463 ——- Epoch 12 ——-
Training [160/500 (31%)] Loss: 0.000237 [320/500 (62%)] Loss: 0.000275 [480/500 (94%)] Loss:
0.000282 Loss = 0.0002332210189197212 Validation Loss = 0.0002531874284613878 ——- Epoch
13 ——- Training [160/500 (31%)] Loss: 0.000222 [320/500 (62%)] Loss: 0.000233 [480/500 (94%)]
Loss: 0.000238 Loss = 0.0002265782072208822 Validation Loss = 0.0002470219059614465 ——-
Epoch 14 ——- Training [160/500 (31%)] Loss: 0.000262 [320/500 (62%)] Loss: 0.000164 [480/500
(94%)] Loss: 0.000211 Loss = 0.00022057966119609773 Validation Loss = 0.0002420657558832318
——- Epoch 15 ——- Training [160/500 (31%)] Loss: 0.000276 [320/500 (62%)] Loss:
0.000255 [480/500 (94%)] Loss: 0.000192 Loss = 0.00021789974515559155 Validation Loss =
0.0002377481124131009 Time Elapsed: 31.493306 seconds
```

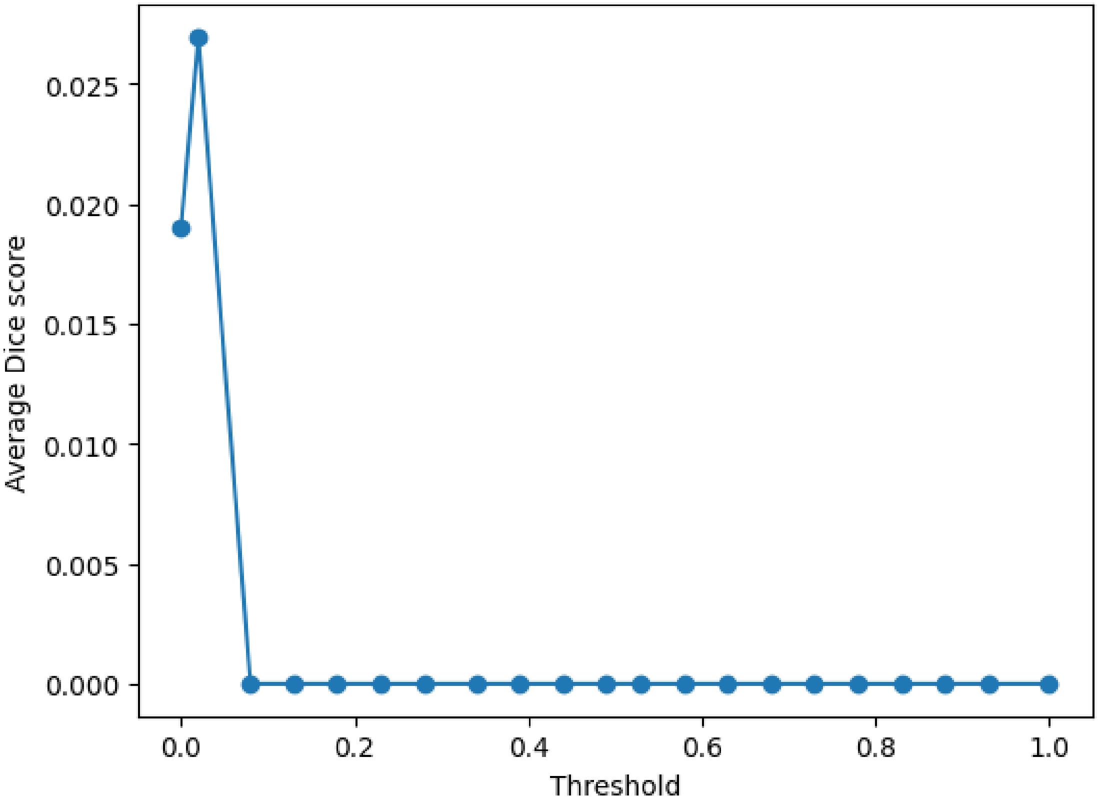
[13]:

```
im = Image.open("bdtconvnet1loss.png")
print("Loss for bdt convnet1")
display(im)

im = Image.open("bdtconvnet1dice.png")
print("Dice Scores for bdt convnet1")
display(im)
```

```
Loss for bdt convnet1
```



```
Dice Scores for bdt convnet1
```



```
[ ]:
```

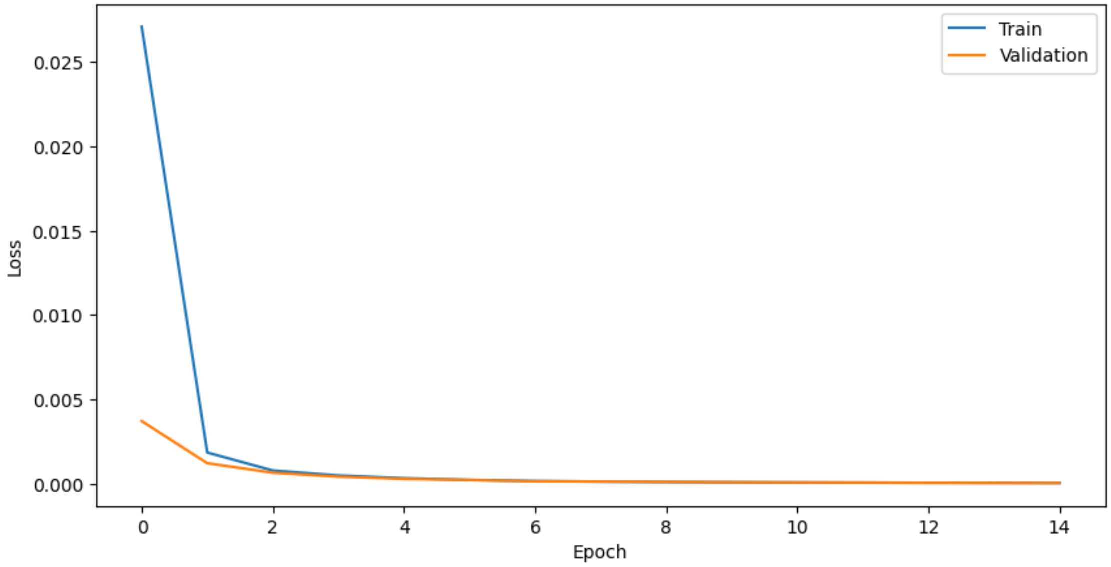
```
[ ]:
```

unet2 [160/500 (31%)] Loss: 0.026632 [320/500 (62%)] Loss: 0.009507 [480/500 (94%)] Loss: 0.004155 Loss = 0.02710374398902058 Validation Loss = 0.0037183952145278453 ——- Epoch 2 ——- Training [160/500 (31%)] Loss: 0.002044 [320/500 (62%)] Loss: 0.001458 [480/500 (94%)] Loss: 0.001029 Loss = 0.0018545807991176841 Validation Loss = 0.00122072272002697 ——- Epoch 3 ——- Training [160/500 (31%)] Loss: 0.000924 [320/500 (62%)] Loss: 0.000684 [480/500 (94%)] Loss: 0.000618 Loss = 0.0007886948613449929 Validation Loss = 0.0006561698159202933 ——- Epoch 4 ——- Training [160/500 (31%)] Loss: 0.000519 [320/500 (62%)] Loss: 0.000480 [480/500 (94%)] Loss: 0.000409 Loss = 0.0004953220298048108 Validation Loss = 0.0004215691064018756 ——- Epoch 5 ——- Training [160/500 (31%)] Loss: 0.000385 [320/500 (62%)] Loss: 0.000349 [480/500 (94%)] Loss: 0.000274 Loss = 0.0003329477987717834 Validation Loss = 0.000292000079061836 ——- Epoch 6 ——- Training [160/500 (31%)] Loss: 0.000238 [320/500 (62%)] Loss: 0.000229 [480/500 (94%)] Loss: 0.000185 Loss = 0.00022791651496663684 Validation Loss = 0.00022665442898869514 ——- Epoch 7 ——- Training [160/500 (31%)] Loss: 0.000169 [320/500 (62%)] Loss: 0.000142 [480/500 (94%)] Loss: 0.000137 Loss = 0.00016652628337033098 Validation Loss = 0.000146902097039856 ——- Epoch 8 ——- Training [160/500 (31%)] Loss: 0.000123 [320/500 (62%)] Loss: 0.000133 [480/500 (94%)] Loss: 0.000141 Loss = 0.00013117628113832325 Validation Loss = 0.00013694590888917445 ——- Epoch 9 ——- Training [160/500 (31%)] Loss: 0.000101

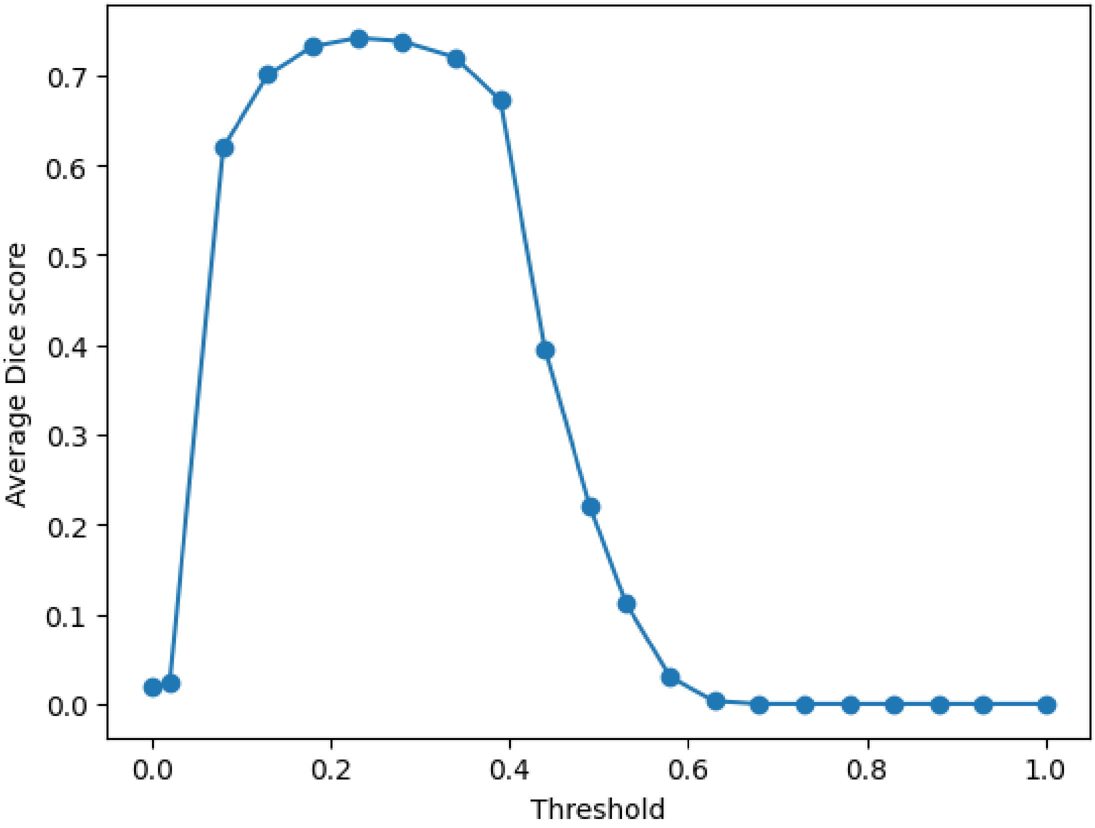
[320/500 (62%)] Loss: 0.000098 [480/500 (94%)] Loss: 0.000096 Loss = 0.00010735255008330569 Validation Loss = 0.00011655587906716392 ——- Epoch 10 ——- Training [160/500 (31%)] Loss: 0.000086 [320/500 (62%)] Loss: 0.000094 [480/500 (94%)] Loss: 0.000083 Loss = 9.03322181547992e-05 Validation Loss = 9.511889918940142e-05 ——- Epoch 11 ——- Training [160/500 (31%)] Loss: 0.000078 [320/500 (62%)] Loss: 0.000075 [480/500 (94%)] Loss: 0.000070 Loss = 7.755001977784558e-05 Validation Loss = 8.166547835571692e-05 ——- Epoch 12 ——- Training [160/500 (31%)] Loss: 0.000079 [320/500 (62%)] Loss: 0.000070 [480/500 (94%)] Loss: 0.000057 Loss = 6.811032741097733e-05 Validation Loss = 7.334564113989472e-05 ——- Epoch 13 ——- Training [160/500 (31%)] Loss: 0.000057 [320/500 (62%)] Loss: 0.000065 [480/500 (94%)] Loss: 0.000054 Loss = 5.969667941099032e-05 Validation Loss = 5.870999011676758e-05 ——- Epoch 14 ——- Training [160/500 (31%)] Loss: 0.000052 [320/500 (62%)] Loss: 0.000051 [480/500 (94%)] Loss: 0.000060 Loss = 5.327554172254168e-05 Validation Loss = 5.427366282674484e-05 ——- Epoch 15 ——- Training [160/500 (31%)] Loss: 0.000047 [320/500 (62%)] Loss: 0.000051 [480/500 (94%)] Loss: 0.000041 Loss = 4.771265460294672e-05 Validation Loss = 4.825429656193591e-05 Time Elapsed: 53.63127 seconds

[14]:
```
im = Image.open("bdtunet2loss.png")
print("Loss for bdt unet2")
display(im)

im = Image.open("bdtunet2dice.png")
print("Dice Scores for bdt unet2")
display(im)
```

```
Loss for bdt unet2
```



```
Dice Scores for bdt unet2
```



[ ]:

[ ]:

[ ]:

[ ]:

1.5 A.3. New Model Design

Modify the tutorial pytorch models to achieve better results.

[ ]:

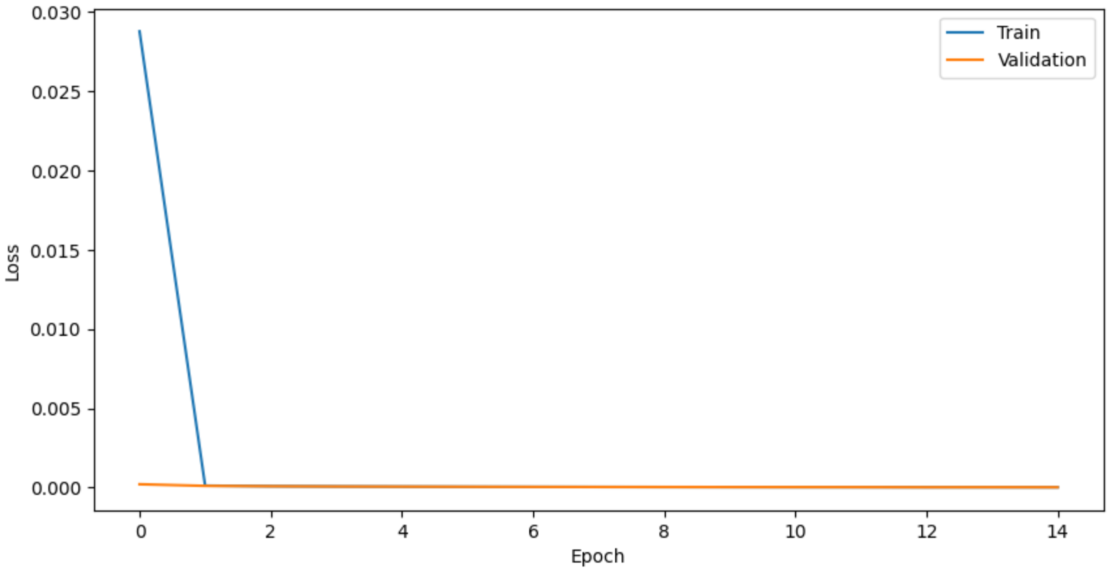
```
#show the new model here
import torch
import torch.nn as nn
import torch.nn.functional as F

class new(nn.Module):
    def __init__(self, in_ch=3, n_channels=8, activation=nn.ReLU):
        super(convnet1, self).__init__()
        self.activation = activation()  # Use a parameterized activation function

        # Convolutional layers
        self.conv1 = nn.Conv2d(in_ch, n_channels, kernel_size=3, padding=1)
        self.bn1 = nn.BatchNorm2d(n_channels)  # Batch normalization after the first convolution
        self.conv2 = nn.Conv2d(n_channels, 1, kernel_size=3, padding=1)
        self.bn2 = nn.BatchNorm2d(1)  # Batch normalization after the second convolution

        # Dropout layer to improve generalization
        self.dropout = nn.Dropout2d(p=0.2)

    def forward(self, x):
        # First convolutional block
        out = self.conv1(x)
        out = self.bn1(out)
        out = self.activation(out)

        # Second convolutional block
        out = self.conv2(out)
        out = self.bn2(out)
        out = self.activation(out)

        # Apply dropout
        out = self.dropout(out)

        return out

class unet2x3(nn.Module):
    def __init__(self, in_ch=3, n_channels=16):
        super(unet2x3, self).__init__()
        # Initial Downsampling block
        self.down1 = down_block(in_ch, n_channels)
        # Second Downsampling block
        self.down2 = down_block(in_ch=n_channels, out_ch=n_channels * 2)
        # Bridge block to connect the encoder and decoder
        self.bridge = down_block(
            in_ch=n_channels * 2,
            out_ch=n_channels * 3,
            max_pooling=False
        )
        # Decoder blocks
        self.up1 = up_block(in_ch=n_channels * 3, out_ch=n_channels * 2)
        self.up2 = up_block(in_ch=n_channels * 2, out_ch=n_channels)
        # Final output layer, reduced kernel size for output adjustment
        self.final_conv = nn.Conv2d(n_channels, 1, kernel_size=1, padding=0)

    def forward(self, x):
        # Encoder: Downsampling stages
        down_output1, skip1 = self.down1(x)  # First down block
        down_output2, skip2 = self.down2(down_output1)  # Second down block

        # Bridge
        bridge_output, _ = self.bridge(down_output2)

        # Decoder: Upsampling stages with skip connections
        up_output1 = self.up1(bridge_output, skip2)  # First up block
        up_output2 = self.up2(up_output1, skip1)  # Second up block

        # Final 1x1 convolution to produce output
        final_output = self.final_conv(up_output2)
        return final_output
```

convnet1x3 [160/500 (31%)] Loss: 0.968150 [320/500 (62%)] Loss: 0.972637 [480/500 (94%)] Loss: 0.971669 Loss = 0.9730825972557067 Validation Loss = 0.9740633511543274 ——- Epoch 2 ——- Training [160/500 (31%)] Loss: 0.970479 [320/500 (62%)] Loss: 0.971219 [480/500 (94%)] Loss: 0.971321 Loss = 0.9702040653228761 Validation Loss = 0.9692761301994324 ——- Epoch 3 ——- Training [160/500 (31%)] Loss: 0.972804 [320/500 (62%)] Loss: 0.969065 [480/500 (94%)] Loss: 0.973208 Loss = 0.9698760833740236 Validation Loss = 0.9689167451858521 ——- Epoch 4 ——- Training [160/500 (31%)] Loss: 0.968652 [320/500 (62%)] Loss: 0.965986 [480/500 (94%)] Loss: 0.972776 Loss = 0.9692247347831726 Validation Loss = 0.965163290500641 ——- Epoch 5 ——- Training [160/500 (31%)] Loss: 0.968763 [320/500 (62%)] Loss: 0.969047 [480/500 (94%)] Loss: 0.965015 Loss = 0.9688313465118408 Validation Loss = 0.9665238070487976 ——- Epoch 6 ——- Training [160/500 (31%)] Loss: 0.964650 [320/500 (62%)] Loss: 0.964974 [480/500 (94%)] Loss: 0.968397 Loss = 0.9681549243927002 Validation Loss = 0.9667536664009094 ——- Epoch 7 ——- Training [160/500 (31%)] Loss: 0.964365 [320/500 (62%)] Loss: 0.969039 [480/500 (94%)] Loss: 0.969592 Loss = 0.9687828216552736 Validation Loss = 0.9697436690330504 ——- Epoch 8 ——- Training [160/500 (31%)] Loss: 0.965906 [320/500 (62%)] Loss: 0.968031 [480/500 (94%)] Loss: 0.969283 Loss = 0.9682415223121644 Validation Loss = 0.9684664058685303 ——- Epoch 9 ——- Training [160/500 (31%)] Loss: 0.966289 [320/500 (62%)] Loss: 0.971051 [480/500 (94%)] Loss: 0.972008 Loss = 0.9680833702087401 Validation Loss = 0.9652195262908936 ——- Epoch 10 ——- Training [160/500 (31%)] Loss: 0.968867 [320/500 (62%)] Loss: 0.967604 [480/500 (94%)] Loss: 0.964595 Loss = 0.9677111654281617 Validation Loss = 0.9642538046836852 ——- Epoch 11 ——- Training [160/500 (31%)] Loss: 0.969538 [320/500 (62%)] Loss: 0.963861 [480/500 (94%)] Loss: 0.968678 Loss = 0.9680660686492919 Validation Loss = 0.9661116266250609 ——- Epoch 12 ——- Training [160/500 (31%)] Loss: 0.967616 [320/500 (62%)] Loss: 0.969298 [480/500 (94%)] Loss: 0.965688 Loss = 0.9675504012107851 Validation Loss = 0.9643642330169677 ——- Epoch 13 ——- Training [160/500 (31%)] Loss: 0.967220 [320/500 (62%)] Loss: 0.969940 [480/500 (94%)] Loss: 0.965645 Loss = 0.9678561139106753 Validation Loss = 0.9631046199798583 ——- Epoch 14 ——- Training [160/500 (31%)] Loss: 0.963787 [320/500 (62%)] Loss: 0.969371 [480/500 (94%)] Loss: 0.969053 Loss = 0.9672271914482115 Validation Loss = 0.9631638455390931 ——- Epoch 15 ——- Training [160/500 (31%)] Loss: 0.967318 [320/500 (62%)] Loss: 0.968674 [480/500 (94%)]

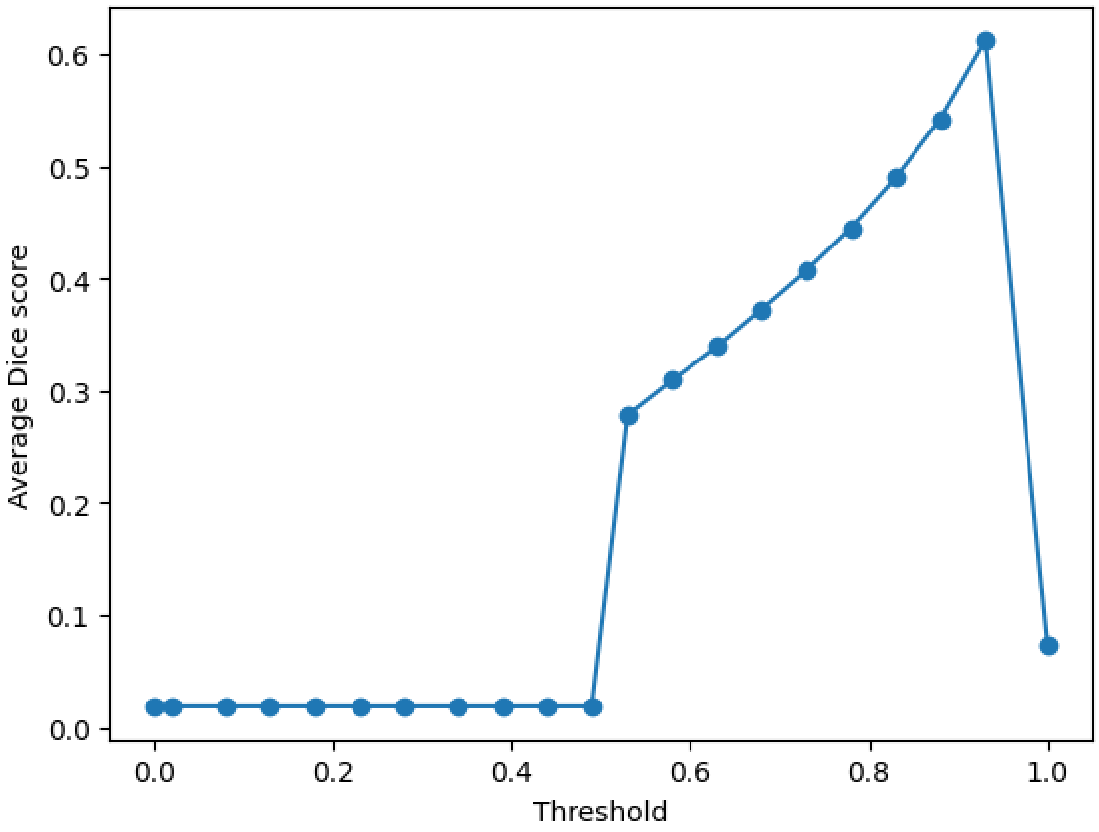
```
Loss: 0.968067 Loss = 0.9677429556846618 Validation Loss = 0.9635839223861694 Time Elapsed: 36.082908 seconds
```

[18]:
```
im = Image.open("unet2x3loss.png")
print("Loss for convnet1x3")
display(im)

im = Image.open("convnet1x3dice.png")
print("Dice Scores for convnet1x3")
display(im)
```

```
Loss for convnet1x3
```



```
Dice Scores for convnet1x3
```



[ ]:

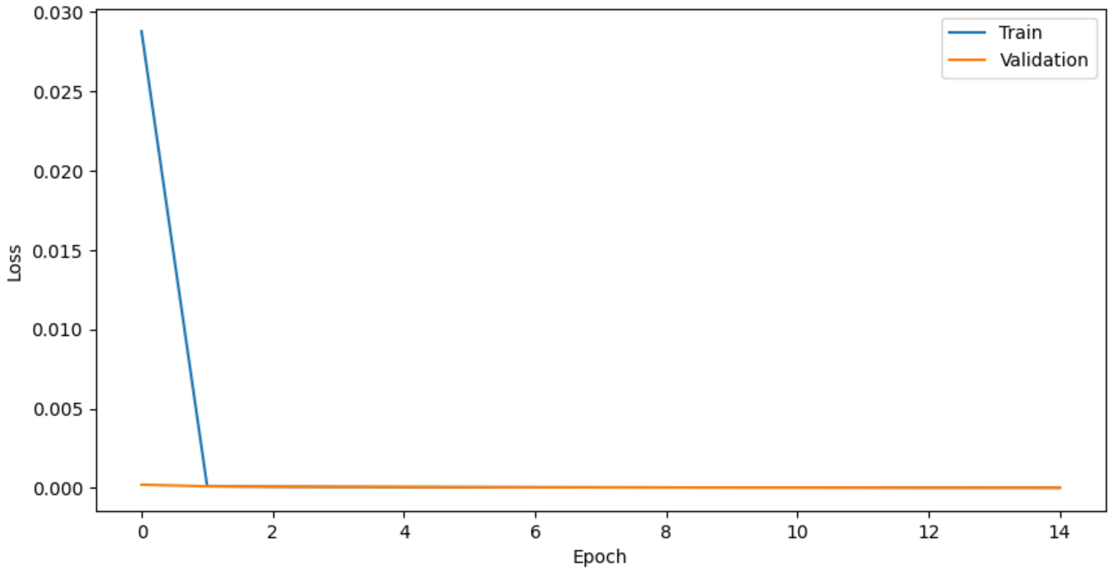
unet2x3 [160/500 (31%)] Loss: 0.005599 [320/500 (62%)] Loss: 0.000381 [480/500 (94%)] Loss: 0.000166 Loss = 0.028777686216402795 Validation Loss = 0.00021134879789315162 ——- Epoch 2 ——- Training [160/500 (31%)] Loss: 0.000122 [320/500 (62%)] Loss: 0.000112 [480/500 (94%)] Loss: 0.000084 Loss = 0.00012536950665526094 Validation Loss = 0.00010951843170914801 ——- Epoch 3 ——- Training [160/500 (31%)] Loss: 0.000094 [320/500 (62%)] Loss: 0.000083 [480/500 (94%)] Loss: 0.000080 Loss = 9.03747298871167e-05 Validation Loss = 8.532053005183117e-05 ——- Epoch 4 ——- Training [160/500 (31%)] Loss: 0.000069 [320/500 (62%)] Loss: 0.000076 [480/500 (94%)] Loss: 0.000057 Loss = 7.356374751543628e-05 Validation Loss = 7.05024620401673e-05 ——- Epoch 5 ——- Training [160/500 (31%)] Loss: 0.000081 [320/500 (62%)] Loss: 0.000058 [480/500 (94%)] Loss: 0.000041 Loss = 6.106562964851037e-05 Validation Loss = 6.25523697817698e-05 ——- Epoch 6 ——- Training [160/500 (31%)] Loss: 0.000046 [320/500 (62%)] Loss: 0.000043 [480/500 (94%)] Loss: 0.000054 Loss = 5.185134758357889e-05 Validation Loss = 5.416892527136951e-05 ——- Epoch 7 ——- Training [160/500 (31%)] Loss: 0.000047 [320/500 (62%)] Loss: 0.000053 [480/500 (94%)] Loss: 0.000045 Loss = 4.525153682334349e-05 Validation Loss = 4.650053437217138e-05 ——- Epoch 8 ——- Training [160/500 (31%)] Loss: 0.000036 [320/500 (62%)] Loss: 0.000058 [480/500 (94%)] Loss: 0.000041 Loss = 3.9803581370506435e-05 Validation Loss = 4.106024745851755e-05 ——- Epoch 9 ——- Training [160/500 (31%)] Loss: 0.000039 [320/500 (62%)] Loss: 0.000041 [480/500 (94%)] Loss: 0.000034 Loss = 3.492948581697419e-05 Validation Loss = 3.729977426701225e-

```
05 ——- Epoch 10 ——- Training [160/500 (31%)] Loss: 0.000025 [320/500 (62%)] Loss:
0.000034 [480/500 (94%)] Loss: 0.000038 Loss = 3.1605061885784386e-05 Validation Loss =
3.28307956806384e-05 ——- Epoch 11 ——- Training [160/500 (31%)] Loss: 0.000032 [320/500
(62%)] Loss: 0.000020 [480/500 (94%)] Loss: 0.000024 Loss = 2.7554560598218814e-05 Validation
Loss = 3.144740083371289e-05 ——- Epoch 12 ——- Training [160/500 (31%)] Loss: 0.000030
[320/500 (62%)] Loss: 0.000021 [480/500 (94%)] Loss: 0.000027 Loss = 2.6384214288555084e-
05 Validation Loss = 2.8929111358593218e-05 ——- Epoch 13 ——- Training [160/500 (31%)]
Loss: 0.000027 [320/500 (62%)] Loss: 0.000021 [480/500 (94%)] Loss: 0.000021 Loss =
2.2898067283676938e-05 Validation Loss = 2.2255088551901285e-05 ——- Epoch 14 ——- Training
[160/500 (31%)] Loss: 0.000020 [320/500 (62%)] Loss: 0.000019 [480/500 (94%)] Loss: 0.000018
Loss = 2.1781092160381378e-05 Validation Loss = 2.564896334661171e-05 ——- Epoch 15 ——-
Training [160/500 (31%)] Loss: 0.000022 [320/500 (62%)] Loss: 0.000018 [480/500 (94%)] Loss:
0.000022 Loss = 1.964850310469046e-05 Validation Loss = 2.1113392504048533e-05 Time Elapsed:
66.755644 seconds
```

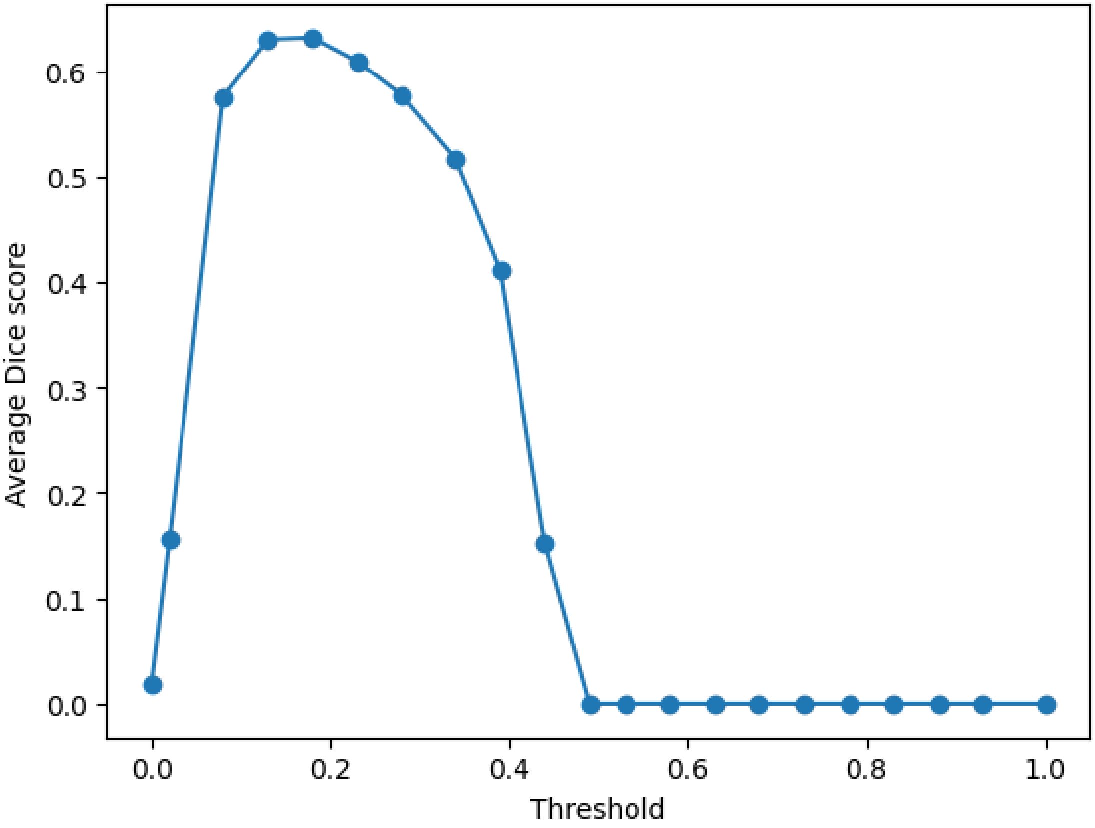
[16]:

```
im = Image.open("unet2x3loss.png")
print("Loss for unet2x3")
display(im)

im = Image.open("unet2x3dice.png")
print("Dice Scores for unet2x3")
display(im)
```

```
Loss for unet2x3
```



```
Dice Scores for unet2x3
```



```
[ ]:
```

```
[ ]:
```

```
[ ]:
```

Discuss any achieved performance improvement. List the changes you have made.

1. convnet1 with 3x3

2. unet2 with 3x3

1.6 A.4. Summary

1.6.1 Comparison of approaches

Model

Avg.DiceScore

TrainLoss

Validationloss

TrainingTime

Epochs

TestingTime

Threshold

convnet1.demo

0.1938

0.0392

0.04091

40.5

15

0.263

0.05

convnet1

unet1

unet2

convnet1+dt

unet2+dt

cnvnet1+bdtl

unet2+bdt

convnet1x3

unet2x3

1.6.2 Discusssion

Make at least three significant observations from the above table #### 1.

2.

3.

[ ]:

```

```