Github link: https://github.com/DebadityaQU/UNet_assignment3_picai

Model architecture:

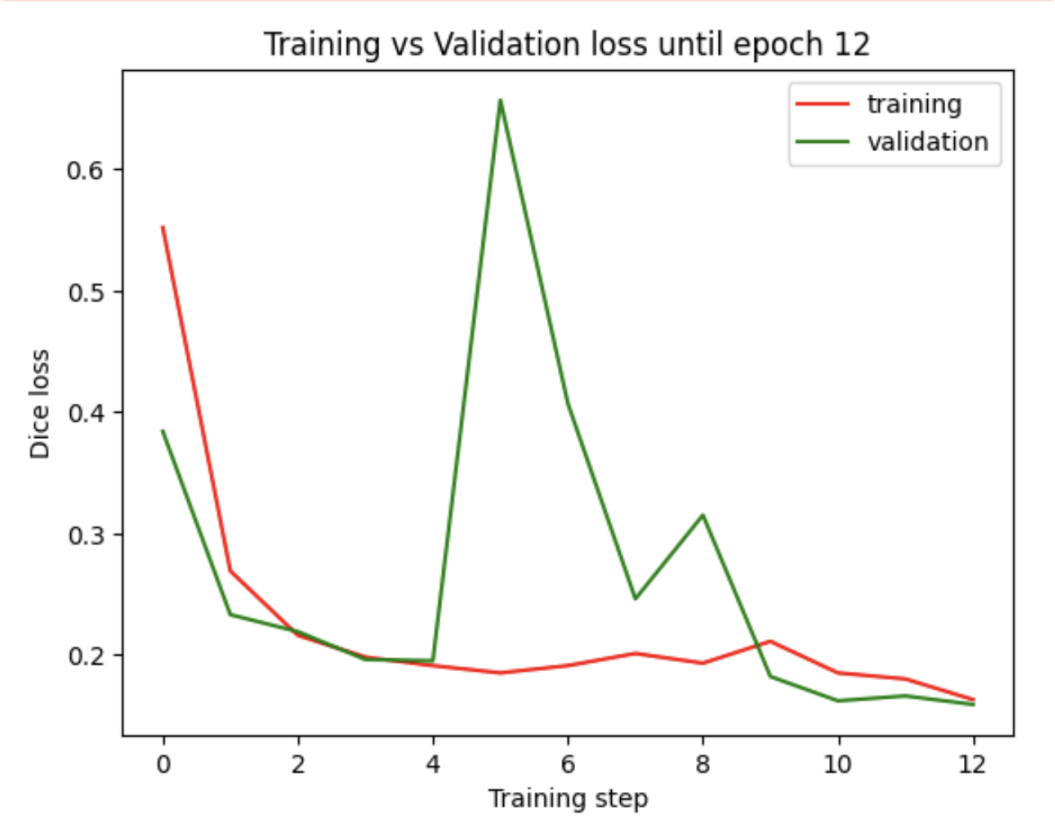
I've chosen the standard UNet architecture (https://arxiv.org/pdf/1505.04597.pdf) and experimented with 2, 3, and 4 level compression/expansion in the Ablation study. The difference between the standard UNet mentioned in the paper and my architecture is the use of ConvTranspose2D (has learnable params) in my architecture instead of Upsampling (no learnable params) in the original paper.

Stratification:

I've used the folds (1, 2, 4) as training, fold 3 as validation, and fold 0 as test.

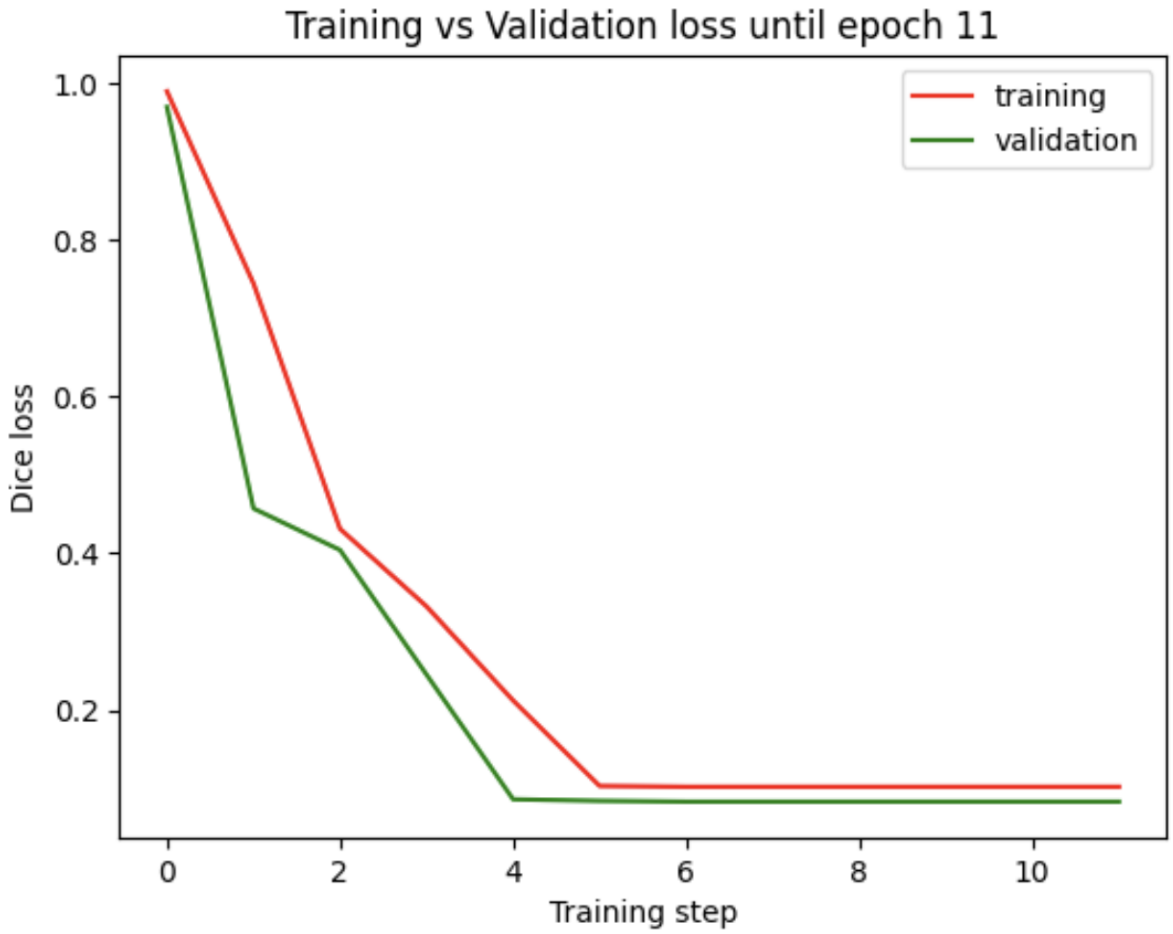
Training performance monitoring:

1) **Task A:**



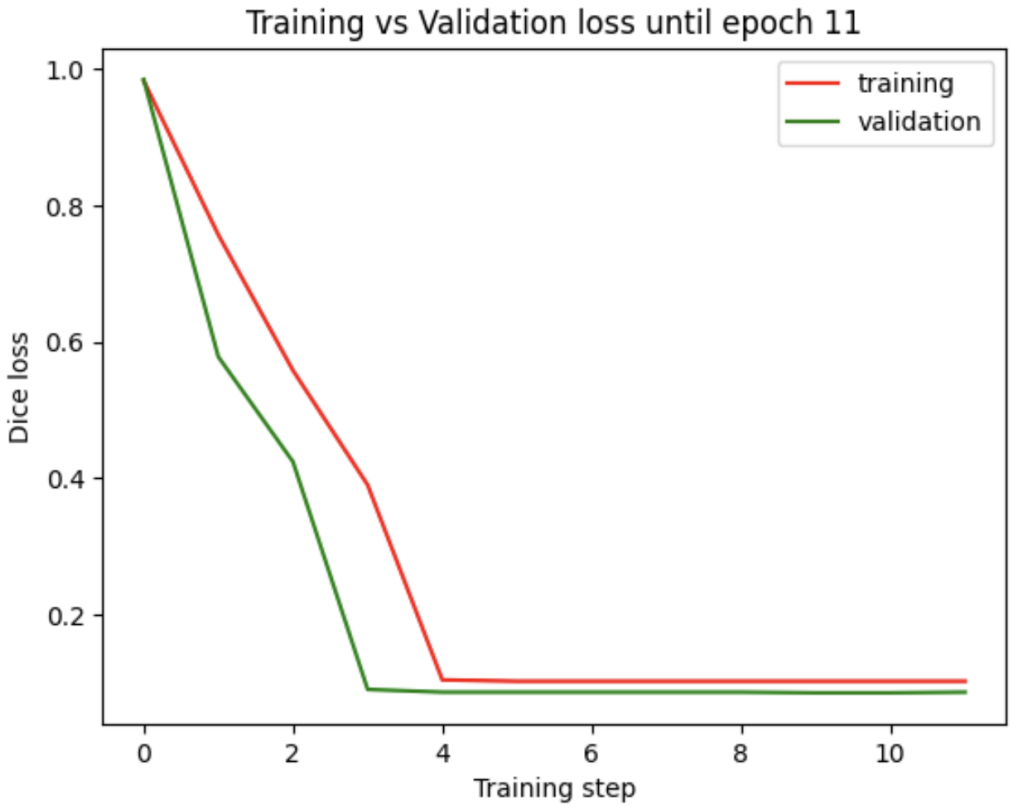

```
Validation loss degraded after patience limit! Stopping training
```

2) **Task B:**




```
Validation loss degraded after patience limit! Stopping training
```

3) **Task C:**




```
Validation loss degraded after patience limit! Stopping training
```

Performance Evaluation after training:

1) **Task A:**

```
*************** STARTING EVALUATION ***************

Evaluation on TEST set!

100%|████████████████████████████████████████████████████████████████| 19
7/197 [00:32<00:00,  6.11it/s]

 Dice score on test set: 0.8348


Evaluation on VALIDATION set!

100%|████████████████████████████████████████████████████████████████| 19
7/197 [00:32<00:00,  6.11it/s]

 Dice score on validation set: 0.8410122881686022


Evaluation on TRAIN set!

100%|████████████████████████████████████████████████████████████████| 59
1/591 [01:35<00:00,  6.20it/s]

 Dice score on train set: 0.8486810104697691
```

2) Task B:

```
*************** STARTING EVALUATION ***************

Evaluation on TEST set!

100%|████████████████████████████████████████████████████████████████| 19
7/197 [00:32<00:00,  6.10it/s]

 Dice score on test set: 0.8986


Evaluation on VALIDATION set!

100%|████████████████████████████████████████████████████████████████| 19
7/197 [00:32<00:00,  6.09it/s]

 Dice score on validation set: 0.9169428200286052


Evaluation on TRAIN set!

100%|████████████████████████████████████████████████████████████████| 59
1/591 [01:36<00:00,  6.10it/s]

 Dice score on train set: 0.8983736051319012
```

3) Task C:

```
********** STARTING EVALUATION **********

Evaluation on TEST set!

100%|| 19
7/197 [00:32<00:00,  6.10it/s]

 Dice score on test set: 0.9020


Evaluation on VALIDATION set!

100%|| 19
7/197 [00:32<00:00,  6.12it/s]

 Dice score on validation set: 0.9145555284422666


Evaluation on TRAIN set!

100%|| 59
1/591 [01:36<00:00,  6.10it/s]

 Dice score on train set: 0.8982177993972894
```

Ablation results:

(i) Ablation 1: UNet compression level (2 vs 3 vs 4)

Task A -> 4 Level UNET performs best (3 level is almost equal)
Task B -> 3 Level and 4 Level UNET performs best (equal)
Task C -> All three level UNETs perform equally

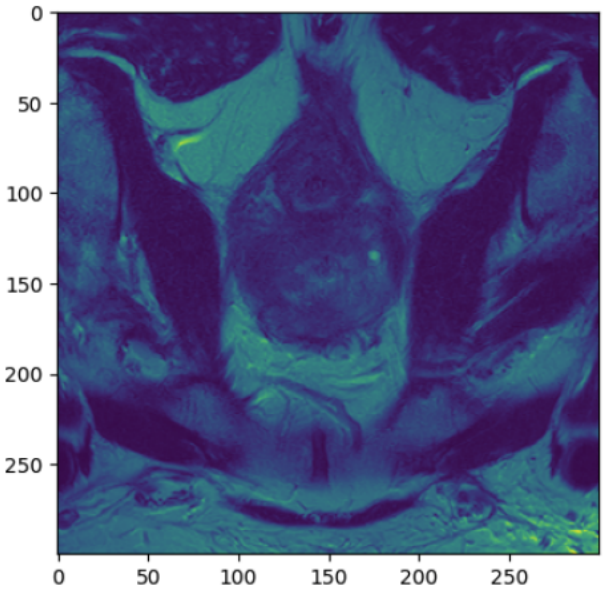
(ii) Ablation 2: Learning rate (2e-4 vs 2e-6)

This experiment would help us check if our initially chosen 2e-4 learning rate was optimal or not. We find that training a 4-level UNet with learning rate 2e-6 results in a dice score of 0.44 on TASK A test set, whereas we earlier got a dice score of 0.83 on a test set with the same model trained using a learning rate of 2e-4. This shows that we chose a pretty-optimal estimate of learning rate.

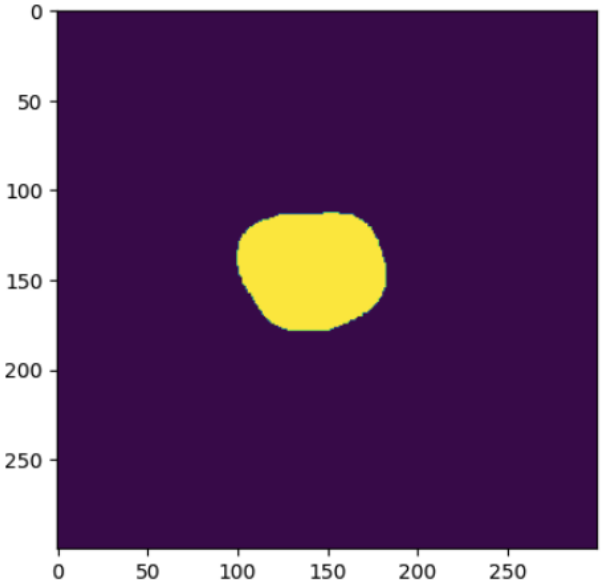
Predictions of best model on:

1) **Task A**

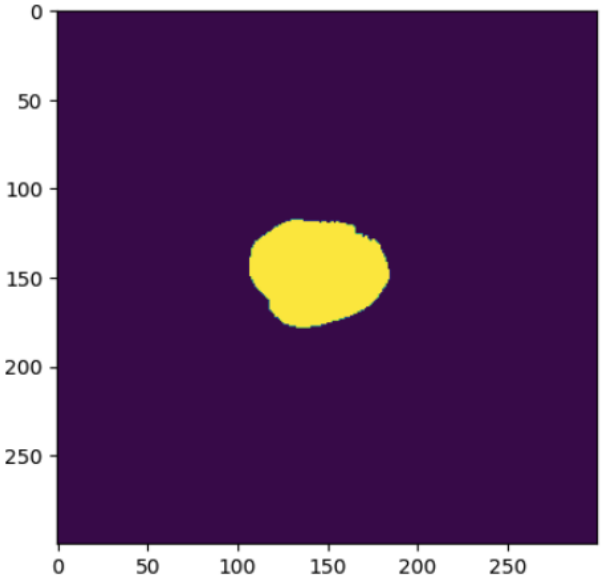
   Input image:



True segmentation mask:

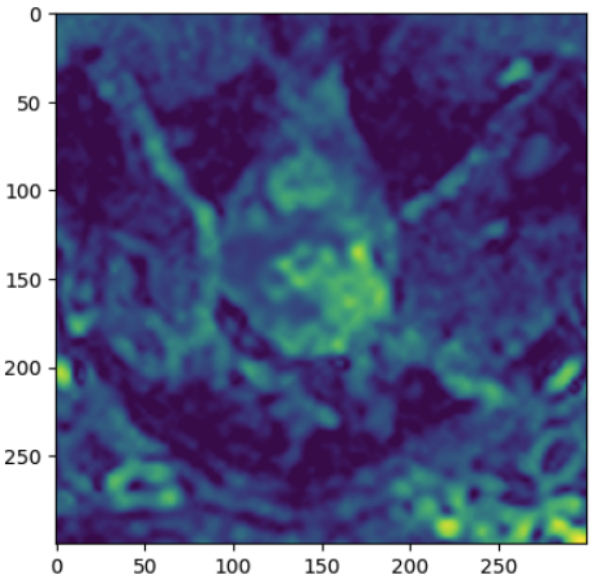


Predicted segmentation mask:

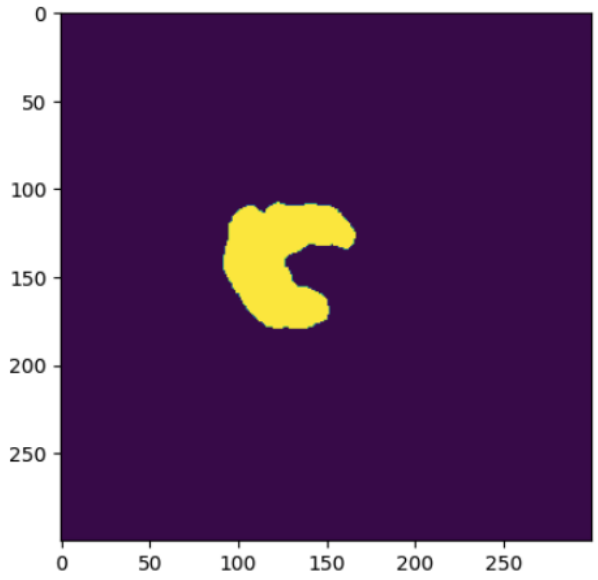


2) Task B

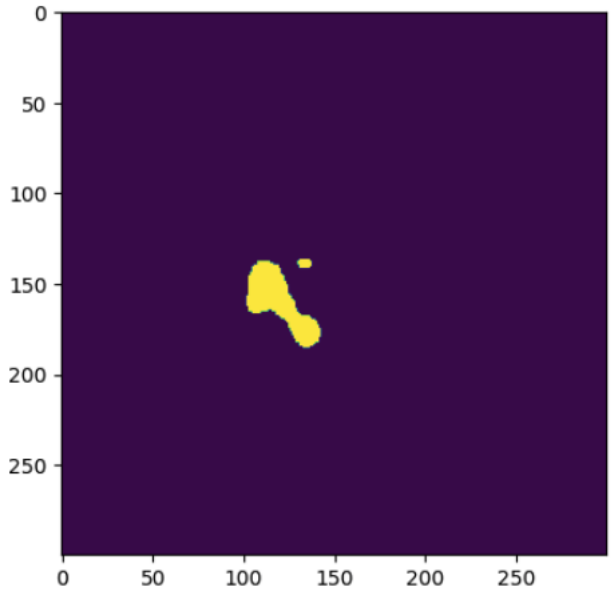
Input image:



True segmentation mask:

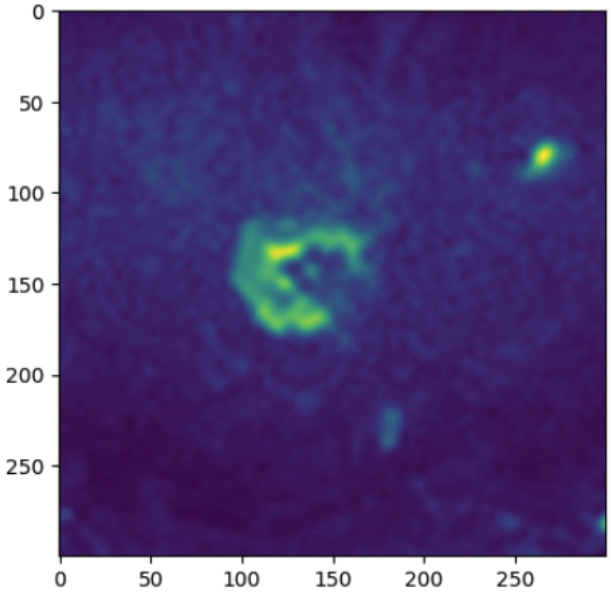


Predicted segmentation mask:

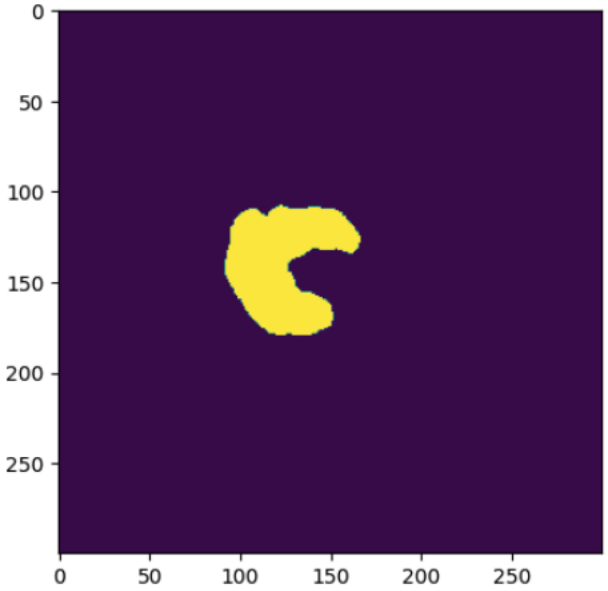


3) Task C:

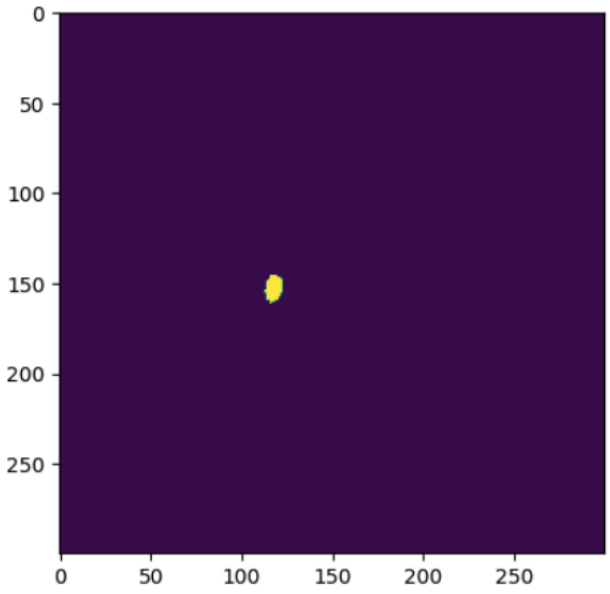
Input image:



True segmentation mask:



Predicted segmentation mask:


0
50
100
150
200
250
0
50
100
150
200
250