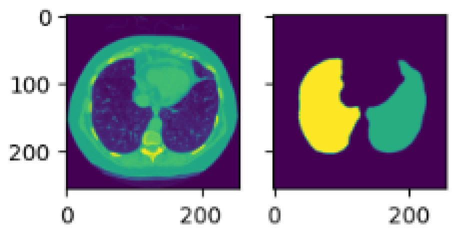
Gnana Spandana Akumalla Serkan Arda Yilal Patrik Svensson

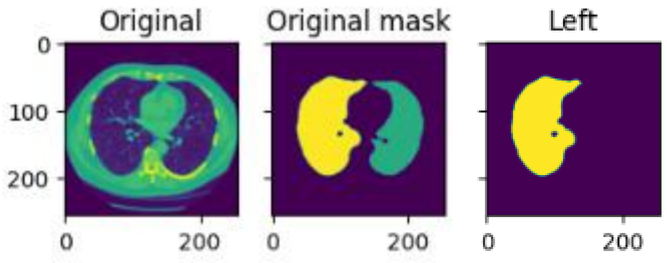
CM2003-Lab 5

Task 1A)

The implemented code for lung segmentation can be seen in the file **Lab5-Task1.ipynb.** Firstly the data input is loaded and also checked to make sure that the images and the masks match up. Then there is a data inspection block to make sure the data is loaded correctly. Left side in the masks (yellow) has the value 0.98431373 and right side (blue) has value 0.61176471. Background is 0:



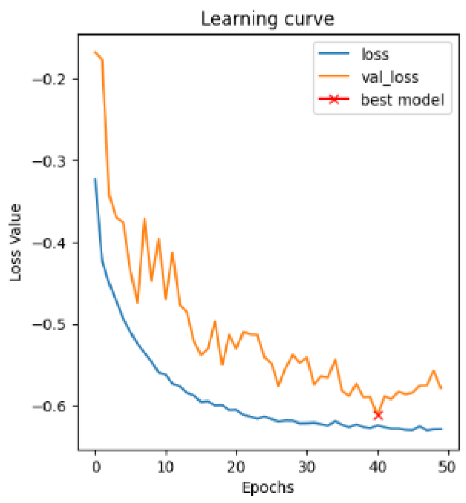
Then there is a block of code to binarize the masks, and also to inspect that the binarization was performed correctly. The binarization works so that all pixels with a value higher than 0.9 becomes 1 and the rest becomes 0:



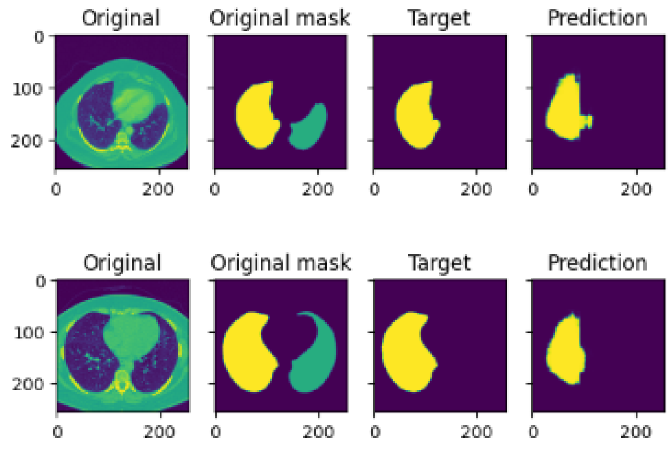
The rightmost image consists only of pixel values 0 or 1, where 1 is the yellow pixels for the lung. Then the model for the left lung is produced and trained for 50 epochs. The loss function used was the dice coefficient loss, learning rate scheduling was employed with initial LR=0.0001. All parameters can be seen in the ipynb-file.

Gnana Spandana Akumalla Serkan Arda Yilal Patrik Svensson

Task 1B) The method of this task was the same as the previous, but augmentation was included in this task as well as precision and recall metrics. The rotation range was set to 10 and both height and width shift range were set to 0.1 while zoom range was set to 0.2. See following images for the learning curve, metrics and the predicted masks.



```
Dice coefficient: 0.709
Precision: 0.803
Recall: 0.644
```



What can be inferred from precision and recall metrics?

Precision tells us what proportion of positive identifications was actually correct, so in this case it tells us how big proportion of predicted yellow pixels were actually part of the target mask. From this we can infer how good the model is at avoiding false positives. A low precision value means more false positives. In our case, with a precision of 80%, it means that 80% of the pixels predicted as yellow were actually part of the left lung. This also means that 20% weren't.

Recall tells us how much of the actual positives that were classified as such. It gives an insight in how good the model is at correctly predicting positive values and avoiding predicting false negatives. A low recall value means more false negatives. A recall of 64% means that 64% of the pixels that were part of the left lung were classified as such.

From our values we can conclude that our model is missing 36% of the left lung, and it is also classifying 20% of non-left lung pixels as belonging to the left lung. Ways to improve this performance would be to:

- Experiment more with parameters as learning rate values and the learning rate decay values, dropout value, L2-regularization value.
- Try the binary_crossentropy loss and compare performances.

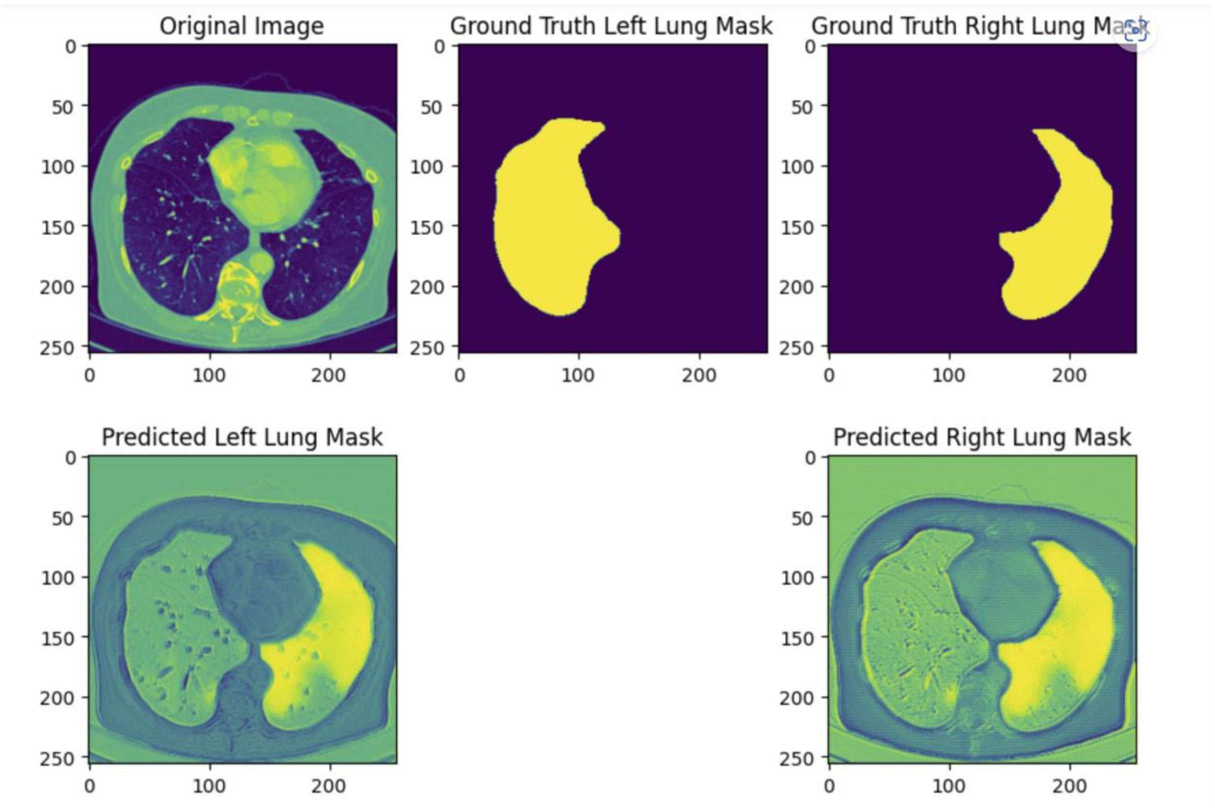
Gnana Spandana Akumalla Serkan Arda Yilal Patrik Svensson

Task 2) Implemented the code for U-net model architecture suitable for multi-organ segmentation for left and right lung images. Defined custom loss function 'multi_organ_loss,' that considers binary cross-entropy for each lung segment. Applied data argumention techniques to the images enhancing the model's ability to handle variation in the data. Also implemented a learning rate scheduling strategy to adjust the learning rate during the training, aiding in convergence and fine-tuning the model. Set the model settings as specified in the task.

The reported metrics provide insights into the model's performance,

```
Multi-Organ Segmentation Metrics on Validation Set:
Dice (Left Lung): 0.729175865650177
Dice (Right Lung): -0.46548697352409363
Precision (Left Lung): 0.8184133768081665
Precision (Right Lung): 1.0
Recall (Left Lung): 0.8140571713447571
Recall (Right Lung): 0.9579192399978638
```

By performing these steps we have successfully adapted a model for multi-organ segmentation focusing on the left and right lungs as below,

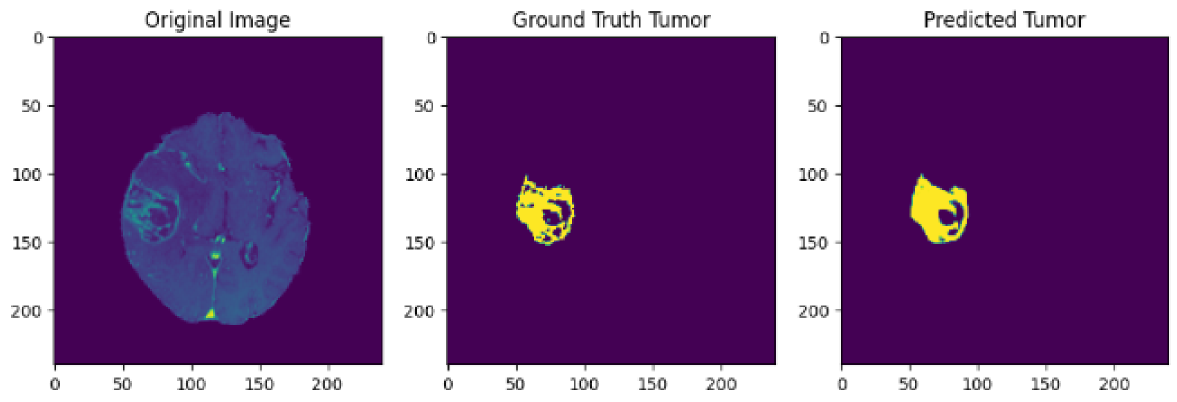

Original Image
Ground Truth Left Lung Mask
Ground Truth Right Lung Mask
Predicted Left Lung Mask
Predicted Right Lung Mask
0
50
100
150
200
250
0
100
200

Gnana Spandana Akumalla Serkan Arda Yilal Patrik Svensson

Bonus Task) This task was about modifying the model to be applied to brain tumours instead. Dice loss was used.

It can be seen that it performs pretty well to predict the tumours.

Possible ways to improve the performance would be to fine-tune the hyperparameters and other parameters like augmentation parameters. It would be interesting to see how well a model based on transfer learning would perform, as it might be possible for it to improve the performance.

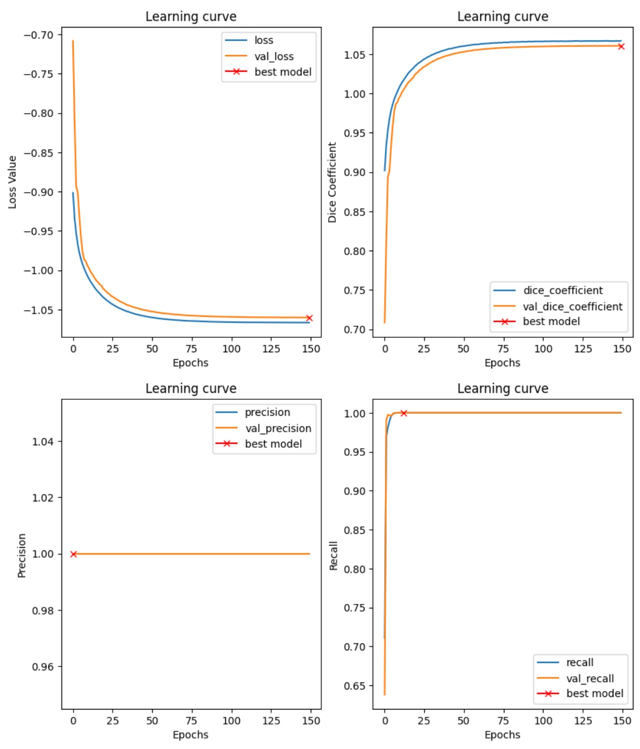


Task 3) In this task, we tried to implement K fold cross validation which separates data into k folds and iterates over it where one of the folds become validation data and the remaining folds become training data. Hence, K different combination of training happens.

For this task, k was given as 3. We ran it on left lungs of CT dataset and calculated metrics such as dice coefficient, precision, and recall. Considering we had trouble on loading all the images, we only took 1000 images for this task. We chose epochs as 150 as it is between 100-200 as it was mentioned. The loss plot and metric plots of each fold can be seen as follows:

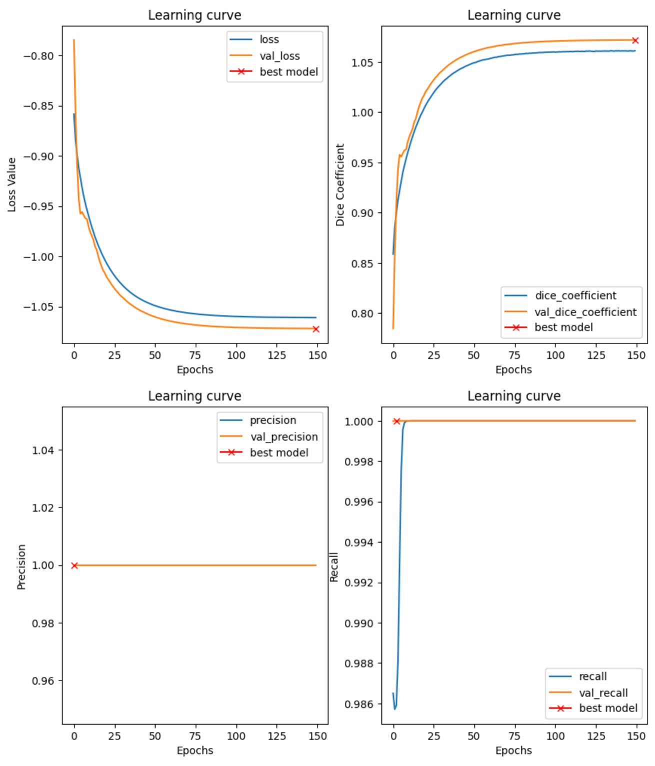
- For Fold 1 = Validation data, Fold 2&3 = Training data:

CT Dataset Left Lung - learning rate = 0.0001, epochs = 150, base = 8, batch_size = 8, dropout = 0.2, FOLD = 1



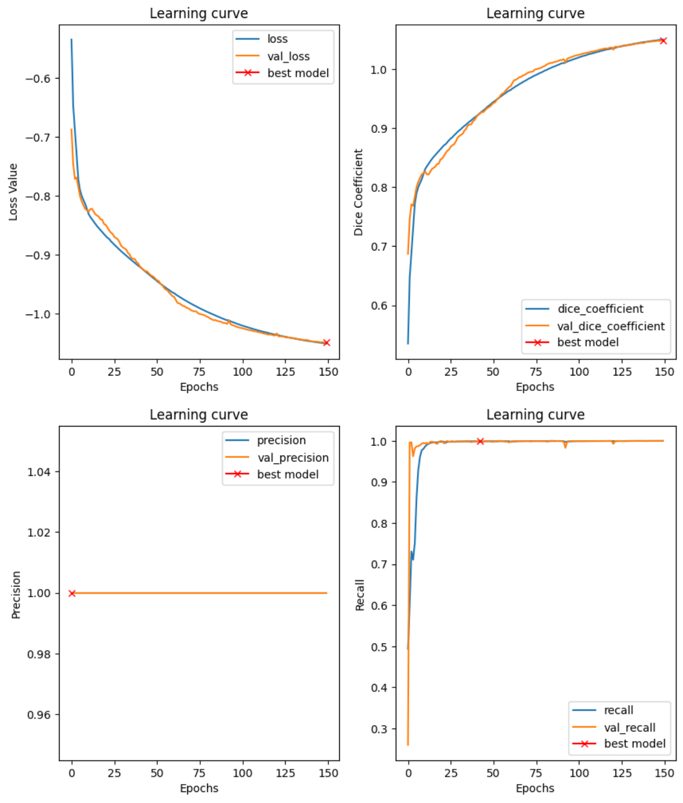
- For Fold 2 = Validation data, Fold 1&3 = Training data:

CT Dataset Left Lung - learning rate = 0.0001, epochs = 150, base = 8, batch_size = 8, dropout = 0.2, FOLD = 2



- For Fold 3 = Validation data, Fold 1&2 = Training data:

CT Dataset Left Lung - learning rate = 0.0001, epochs = 150, base = 8, batch_size = 8, dropout = 0.2, FOLD = 3



When we compare these results, we see that there are small differences on the learning curve of dice coefficient between different folds. For the case where validation data is second fold, the validation loss is lower than training loss while for the other cases, it is the same or higher than training loss. On the other hand, the other metrics do not differ so much. If we were able to call all of the images in the folders instead of only 1000 images, the results might have changed as the diversity might have increased and it might have taken more epochs to achieve the highest score on validation data.