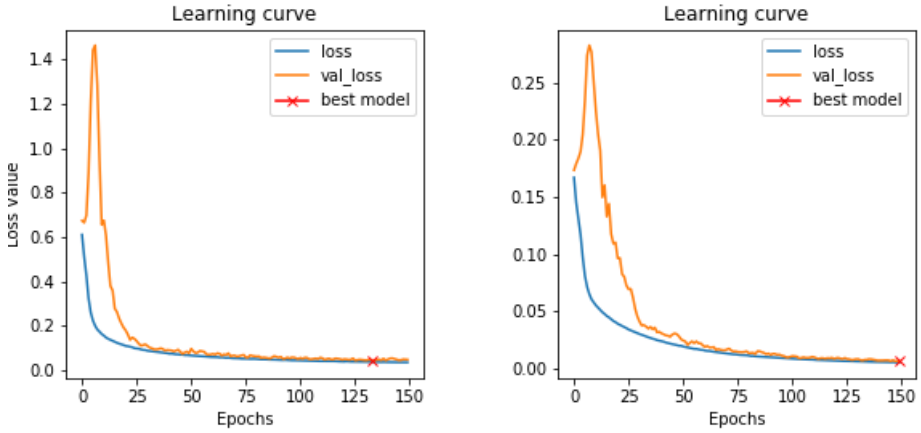
CM2003 Lab3

Libo Xu and Md Rajibul Islam

Task1:

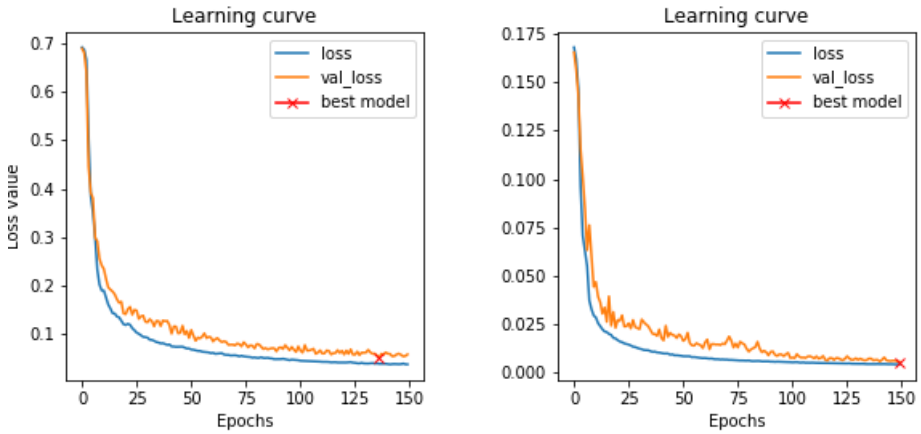


(a) (b)

Figure 1: Task 1a (left) Task 1b(right)

Loss and metrices:
Binary cross entropy 1(a), val_loss : 0.0431 - val_dice_coef :0.9954
Dice lose 1(b), val_loss : 0.0059 - val_dice_coef :0.9941
Model performs better when we change the loss function to *Dice loss*, because the validation loss is much lower after changing the loss function.

Task2:



(a) (b)

Figure 2: Task 2a (left) Task 2b(right)

Loss and metrices:
Binary cross entropy 2(a), val_loss : 0.0497 - val_dice_coef :0.9937

Dice lose 2(b), val_loss : 0.0054 - val_dice_coef :0.9946
It does not affect the model performance much, because the validation loss and accuracy did not change much compared to task 1. But the learning curves are better because the big fluctuations in the beginning of the learning process disappeared.

Task3:

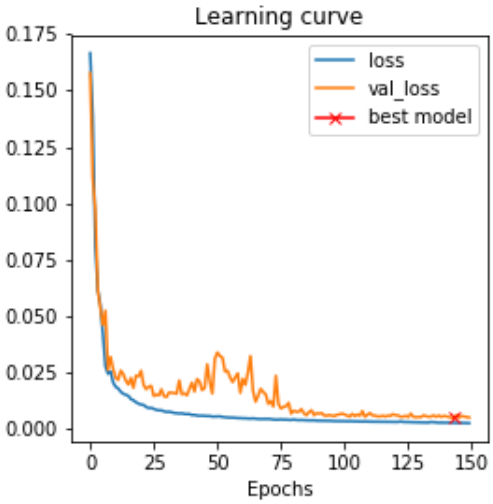


Figure 3

Loss and metrices:
The best setting is batchnormalization = False, loss function = 'Dice loss'
val_loss : 0.0048 - val_dice_coef :0.9952
The segmentation accuracy is higher. Large number of features maps is better in many cases.

Task4

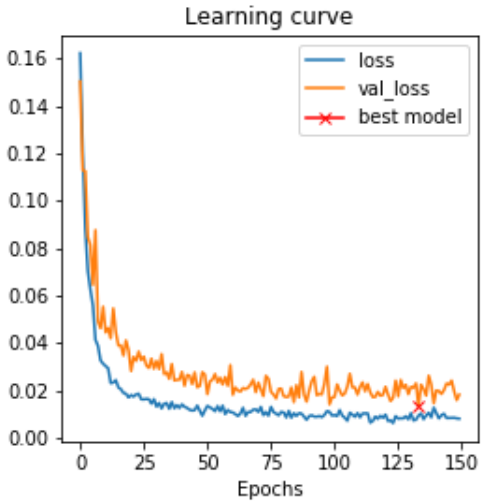
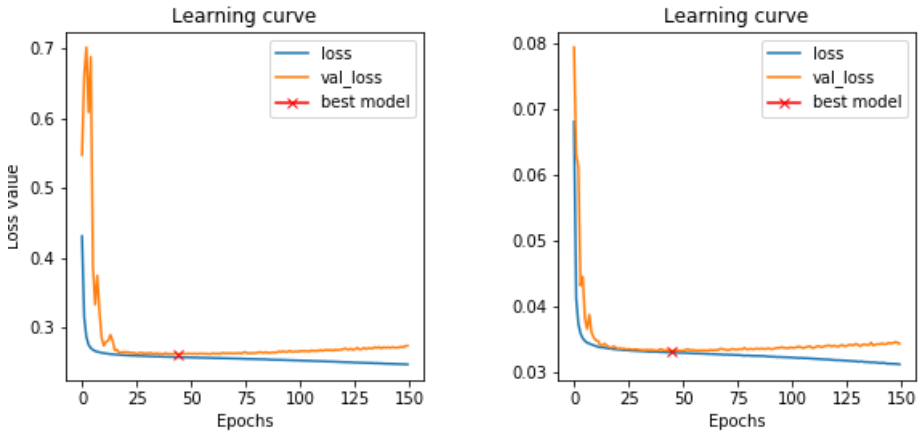


Figure 4

Loss and metrices:
val_loss : 0.0135 - val_dice_coef :0.9863
From the validation and the validation dice coefficient we got, data augmentation didn't improve the model performance in this case. The generalization power is lower.

Task 5a

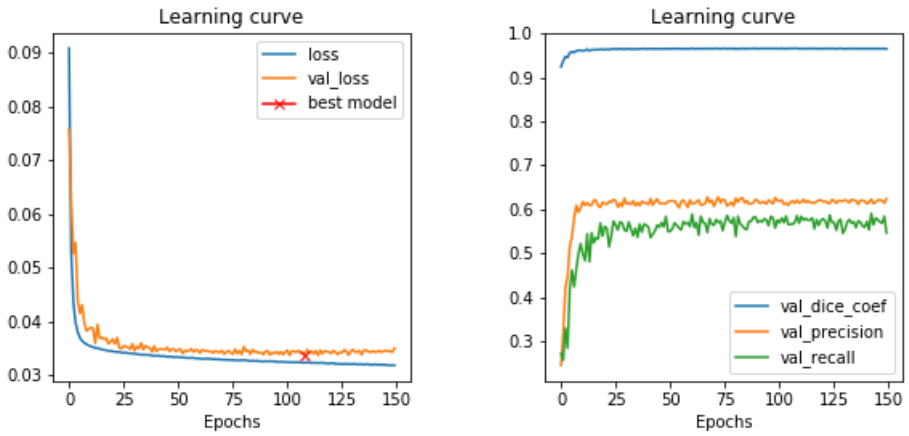


(a) (b)

Figure 5: Task 5a_1 (left) Task 5a_2 (right)

Loss and metrics:
Binary cross entropy (5a_1), val_loss : 0.2626 - val_dice_coef :0.9662
Dice loss (5a_2), val_loss : 0.0333 - val_dice_coef :0.9667
X-ray yielded more accurate segmentation results. Because of the labels of the left and right lung in the CT masks are not the same values, which makes the classes to be 3. But the last layer of the network is still *Conv2D(1, (1,1), activation= 'sigmoid')(n)*. So it works well on binary problems.

Task 5b

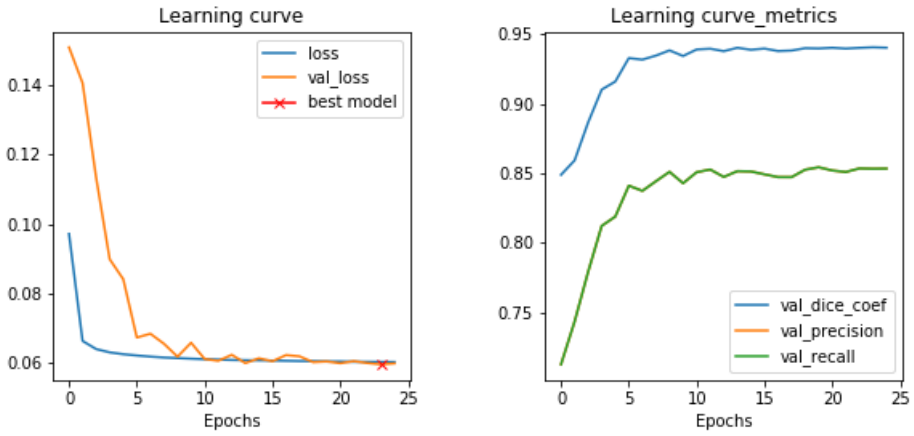


(a) (b)

Figure 6: Loss and metrics)

As the 'Dice loss' performs better in the previous task, we have only used 'Dice loss' as the loss function for this task. We can see that the values of the dice coefficient over the validation is very high and converges to 0.9660. However the values of the precision and recall are quite low compared to dice coefficient, which converge to around 0.63 and 0.56. But in the later tasks, the values precision and recall over validation set are closed to 1.0. So, may be the augmentation affects the precision and recall.

Task 6

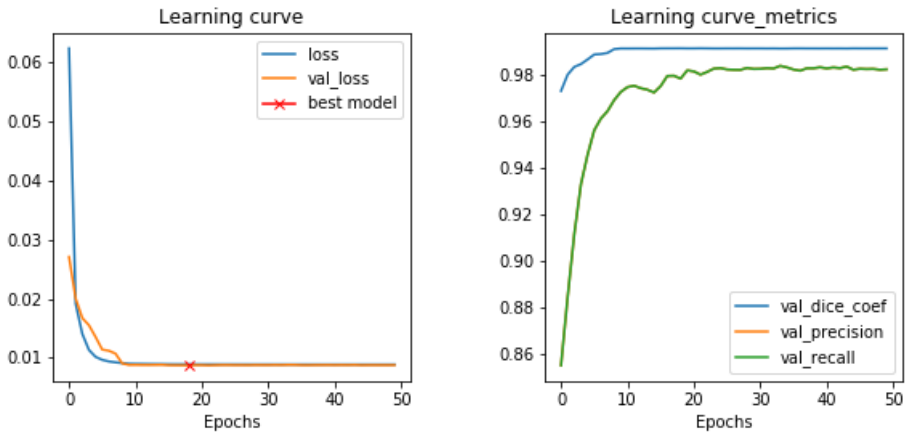


(a) (b)

Figure 7: Loss and metrics)

Loss and metrics:
val_dice_coef :0.9406 - val_precission: 0.8533 - val_recall :0.8533

Task 7



(a) (b)

Figure 8: Loss and metrics)

Loss and metrics:
val_dice_coef :0.9931 - val_precission: 0.9782 - val_recall :0.9782
Best setting:
Base = 32, dropout = 0.5 batchnorm = True, lr=0.00001, batch_size=4, epochs=50