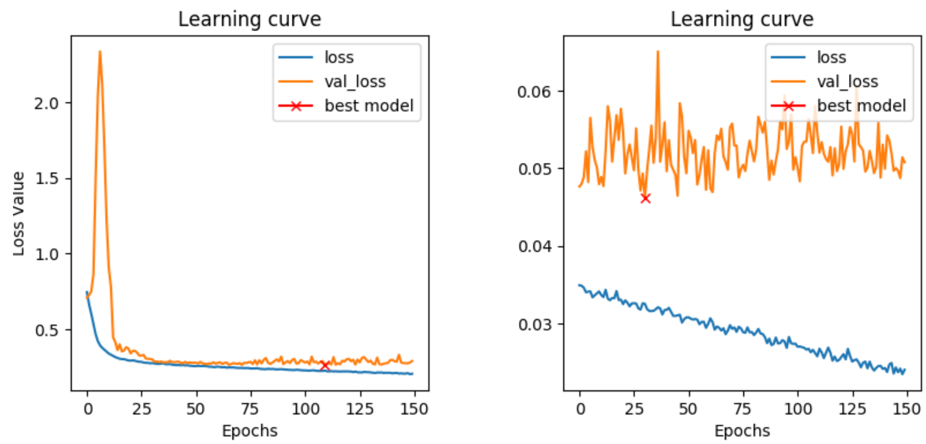
CM2003 Lab3

Md Rajibul Islam

Task1:
By considering the given setting

number of filters	image size	batch size	dropout	batch normalization	learning rate (Adam)	metics
16	256	8	0.5	True	0.0001	Dice Coefficient

the following learning curves are achieved.

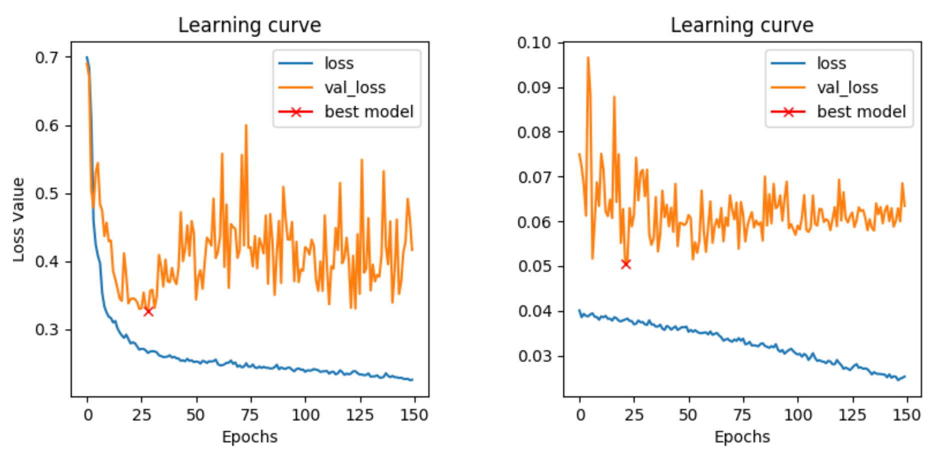


(a) (b)

Figure 1: Task 1a (a: loss function *binary cross entropy*) vs Task 1b(b: loss function *Dice loss*)

The learning curve with loss function *binary cross entropy* (1a) is more stable concerning the over fitting. But with the loss function *Dice loss* (1b), the validation loss is much smaller.

Task 2



(a) (b)

Figure 2: Task 1a (left: loss function *binary cross entropy*) Task 1b(right: loss function *Dice loss*)

In task 2 with batch normalization = False, we can see that the left learning curve shows more over fitting compared with the task 1. On the other hand, the right learning curves shown almost similar features for task 1 and task 2. But with the batch normalization = True, the validation loss is a bit smaller.

By comparing among the four pictures, we can conclude that the learning curves with loss function *binary cross entropy* shown better performances concerning with the over fitting. On the other hand, by considering with the validation loss, I would say the right picture in task 1 (with loss function = Dice loss and batch normalization = True) provided better result.

Task 3
For this task, I have considered the setting with the validation loss is lower (with loss function = Dice loss and batch normalization = True). Similar setting with task 1b but with the # of filters at the first layer as 32. Then I have achieved the following learning curve.

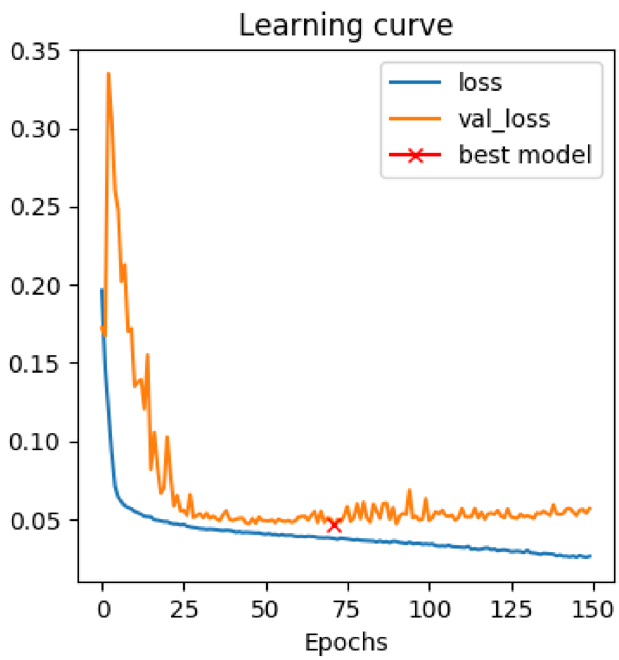


Figure 3

By comparing this figure with the figure in task1b, we can say that the higher base size improved the result. But still some over fitting properties exits.

The features map worked as a input of the of the system. The high number of features map represents the high number of input. Moreover, for the first hidden layer, the feature map will be connected with the different regions of the input images. Therefore, high number of filters in the first layer might provides the better outcomes. But at the same time it increases the cost of the system.

Task 4
In this task by considering the given parameters the augmentation technique has been applied. The following learning curve are achieved.

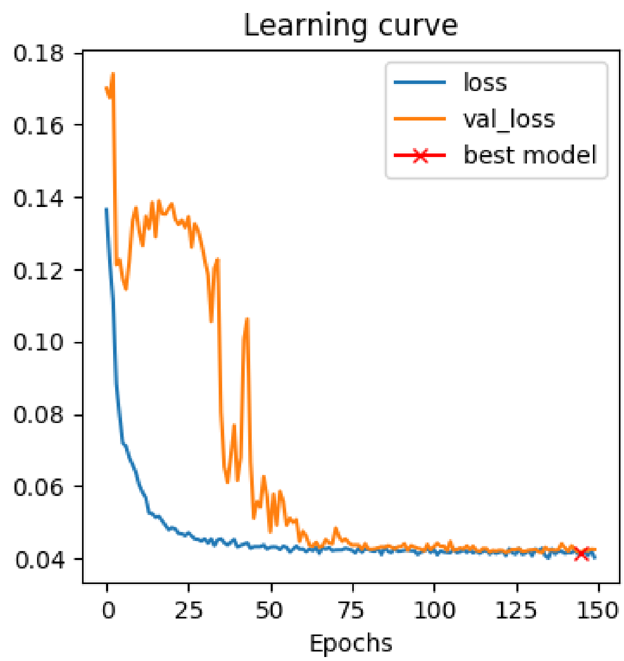
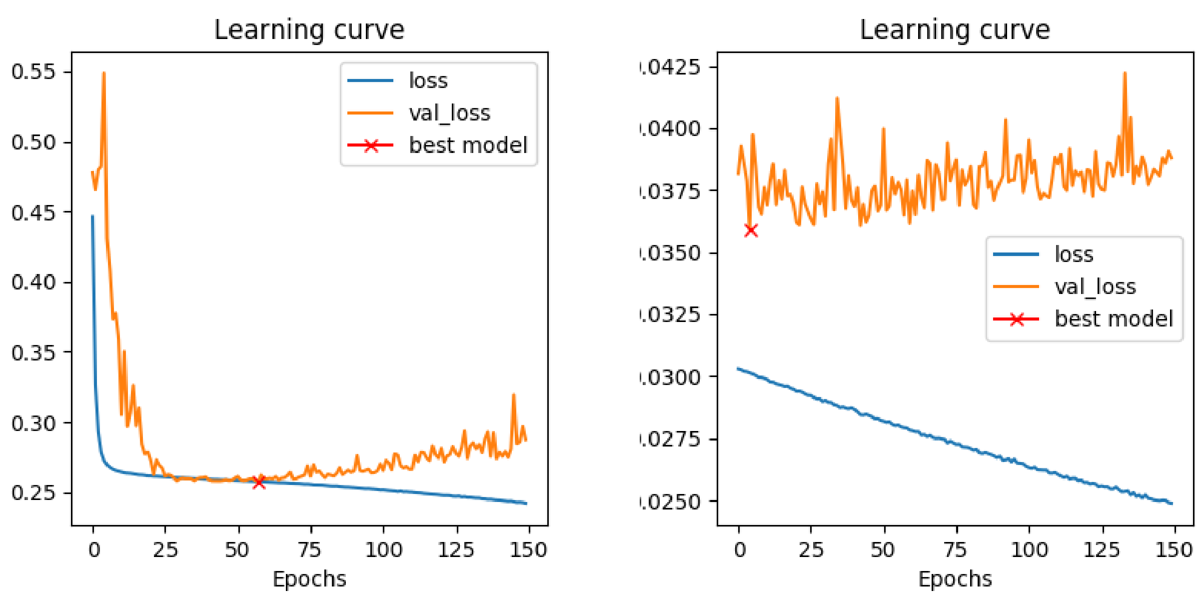


Figure 4

From the Figure 4 , it can be seen that the over fitting has been removed and improved the result, validation loss also a bit smaller. The generalization power can be said quite a good.

Task 5a
For this task, similar setting like task 1 has been used. But instead of X-ray images, the experiment has been performed with CT images.



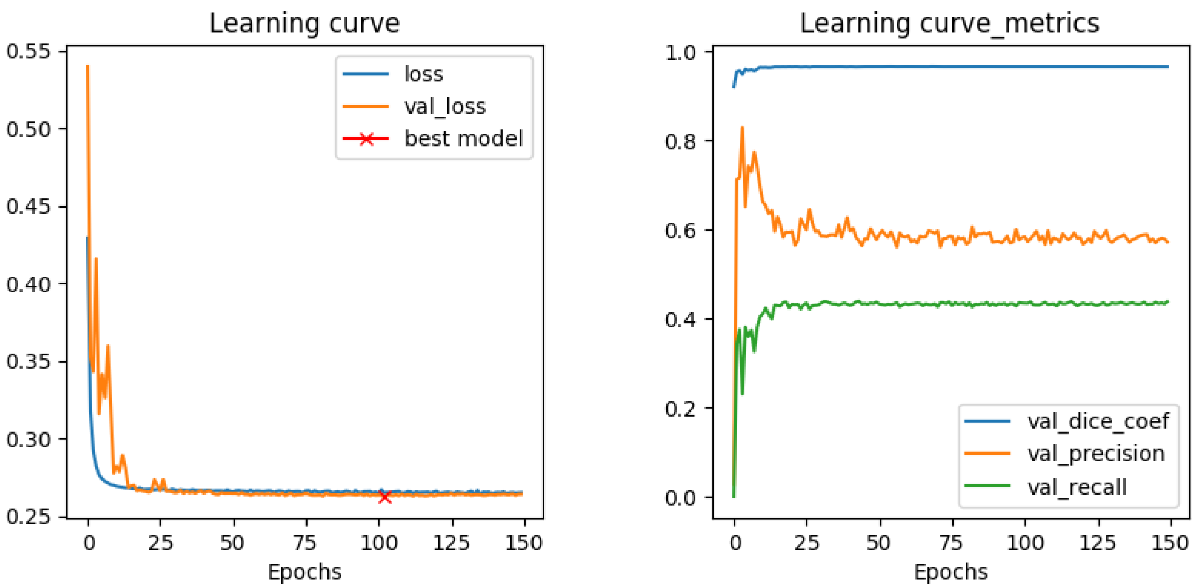
(a) (b)

Figure 5: Left: loss function *binary cross entropy*, Right: loss function *Dice loss*

If we compared learning curves in Figure 5 with the learning curves in Figure 1, more clearly with the loss function *binary cross entropy*, it can be said the the model with the X-ray images (Task 1) performed better than the CT images. It might be one of the causes of the number of output neuron of the mode. Which is one. But CT images contains images for lung segmentation and the lung has two parts (left and right lung) which has different label values.

Task 5b

In this part the data augmentation technique has been used. Then the model performances has been observed by considering the properties like *precision, recall* of the confusion matrix. The *recall* of the confusion matrix provides us an idea about the accuracy of the prediction results. More precisely, when the actual value is yes, how often does it predict yes. On the other hand, the *precision* tells us about when it predicts yes, how often is it correct. Anyway, from the experiment the following results are achieved.

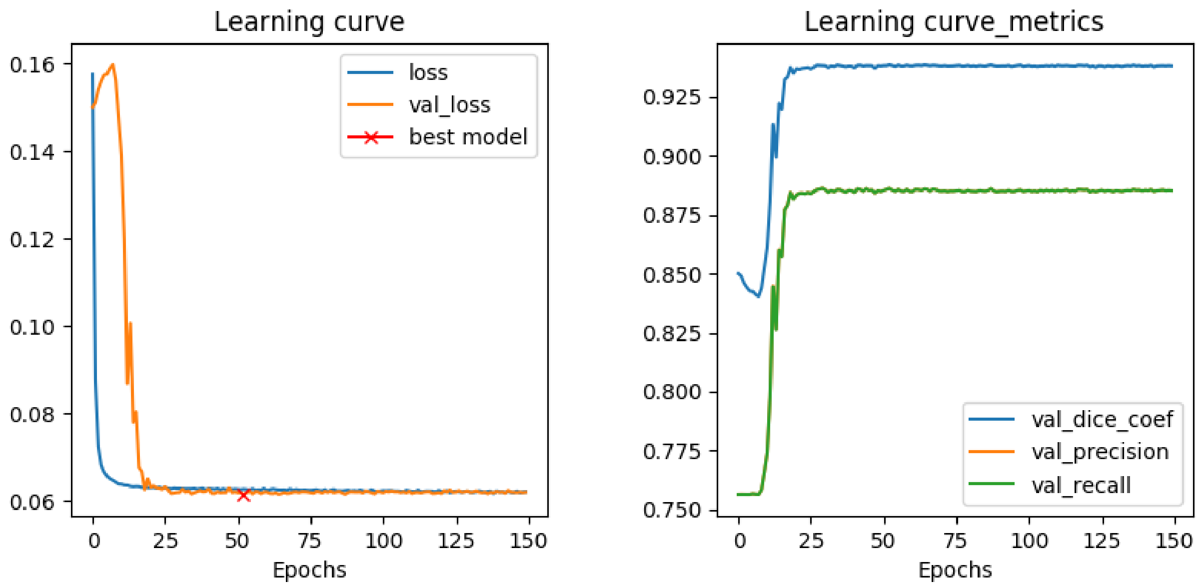


(a) (b)

Figure 6: Learning curve vs metrices with loss function *binary cross entropy*

If we look the picture b concerning the metrics, the model provides quite a good dice coefficient, which is about 95%. But the precision value is around 0.6 & the recall value is just over 0.4. Therefore based on the precision and recall, it can be said that the experiment has not provide good accuracy. On the other hand, the learning curves in picture (a) of Figure 6, provides better result.

Task 6 For this task, the unet model has been modified a bit. The number of output neuron has been change to 3 and the activation function *sigmoid* is replaced by *softmax*. Moreover, to improve the results, one hot encoding has been considered for the mask images.The experimental results are presented in Figure 7 below.



(a) (b)

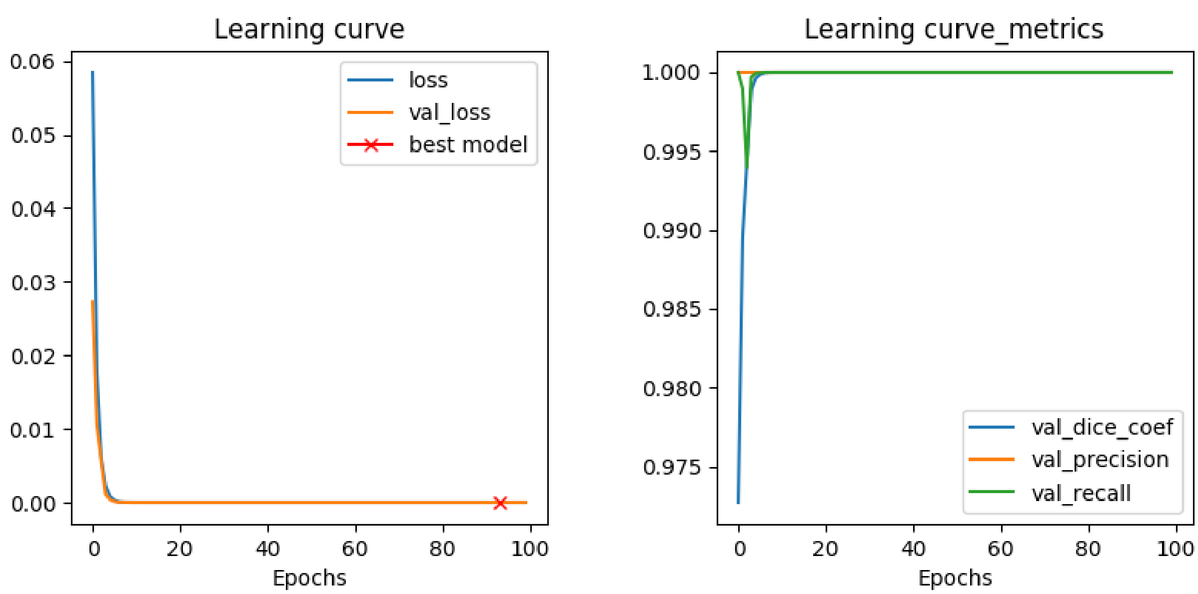
Figure 7: Learning curve vs metrices with loss function *Dice loss*

By comparing the *precision* & *recall* values with the previous task, it can be observed that the model modification has significantly improved the accuracy. Both the recall & precision values more than 88% after a certain number of epochs (23). [Note that the orange curve (precision) is under the green curve (recall)].

Task 7

For this task the *one hot encoding* and the unet model for single neuron has been used. Then the results are presented in below by considering the following setting.

number of filters	image size	batch size	dropout	batch normalization	learning rate (Adam)	metics
16	240	4	0.5	True	0.0001	Dice Coefficient



(a) (b)

Figure 8: Learning curve vs metrices with loss function *Dice loss*

Done !