Task 1 K-fold cross validation

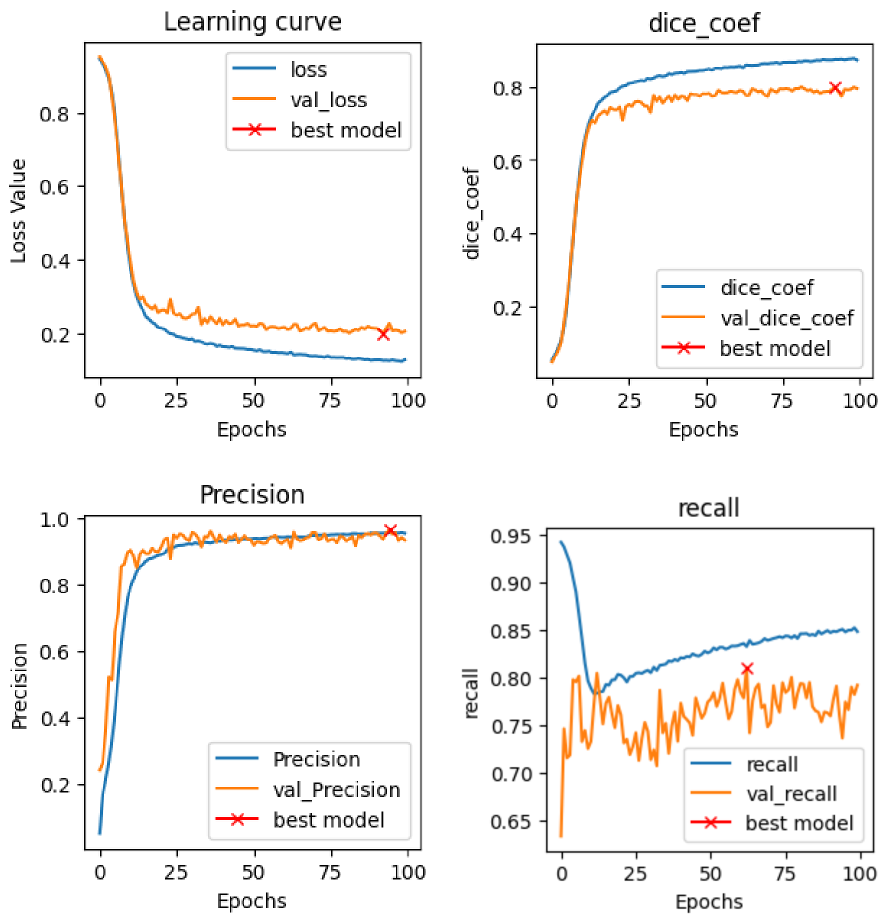
Is the performance consistent across all folds? How would you deal with datasets for which some folds have a different performance than others?

Answer:
The performance from our model is very consistent, especially for dice value and the loss value. For precision and recall, the performance is slightly different between folds, especially in recall.
For the model, we could use this history data in cross validation to get the mean dice of the model as well as its standard deviation.

For the datasets,

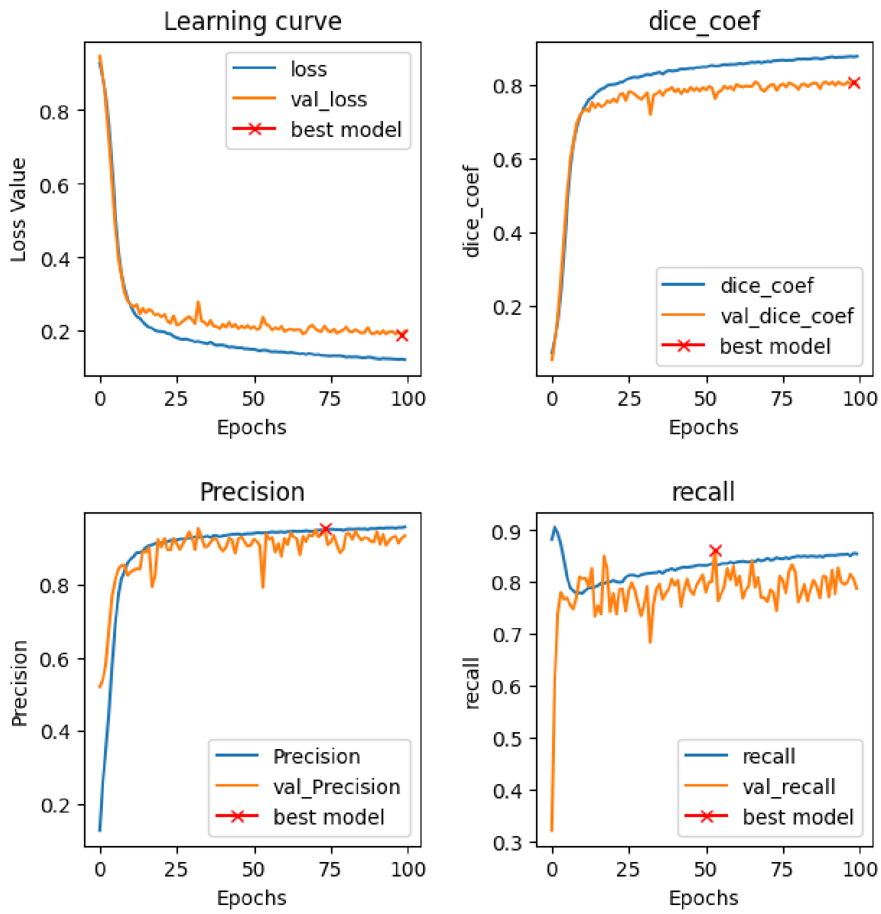
n_folds = 3
epochs = 100

Results from first fold
loss: 0.1275 - dice_coef: 0.8725 - precision: 0.9530 - recall: 0.8484 - val_loss: 0.2047 - val_dice_coef: 0.7953 - val_precision: 0.9324 - val_recall: 0.7920



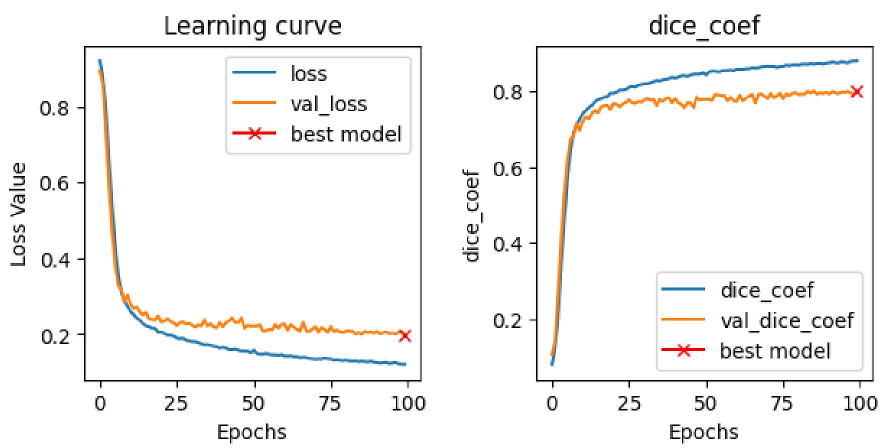
Results from the second fold

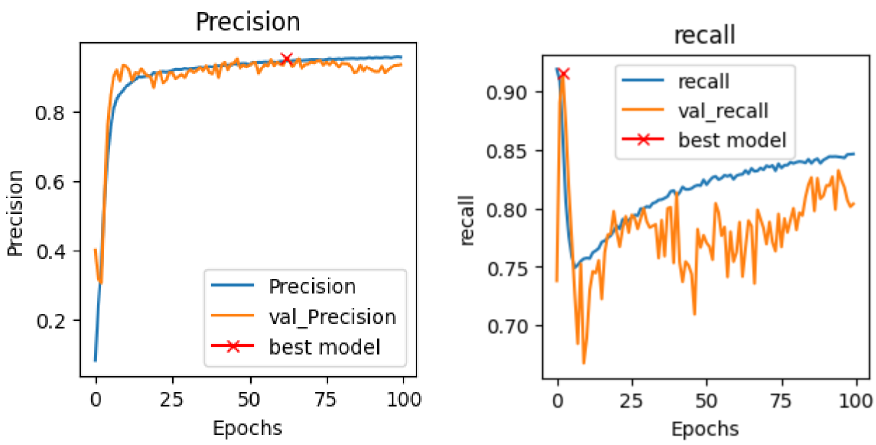
loss: 0.1221 - dice_coef: 0.8779 - precision: 0.9571 - recall: 0.8545 - val_loss: 0.1997 - val_dice_coef: 0.8003 - val_precision: 0.9341 - val_recall: 0.7875



Results from the third fold

loss: 0.1212 - dice_coef: 0.8788 - precision: 0.9546 - recall: 0.8464 - val_loss: 0.1997 - val_dice_coef: 0.8003 - val_precision: 0.9337 - val_recall: 0.8038

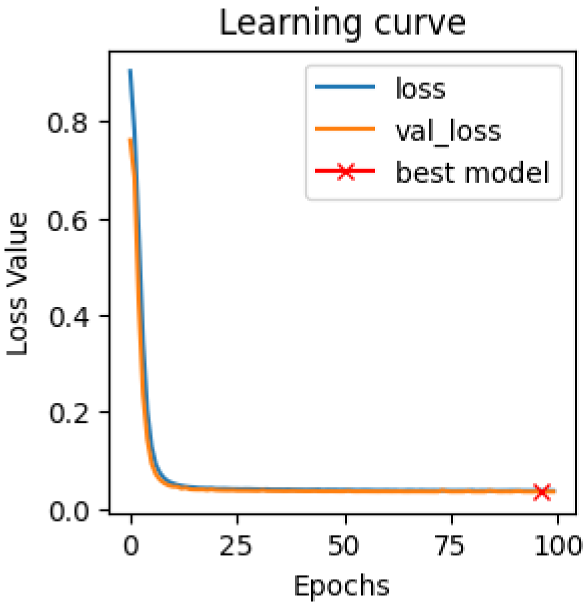




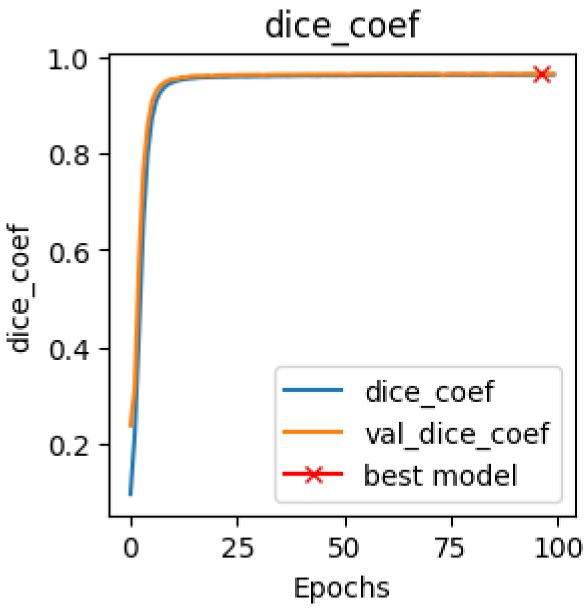
Task 2 Introducing weight maps

You can then finally start training your model. Can you observe a discrepancy between loss function and the traditional (un-weighted) Dice coefficient evaluation metrics? Which accuracy do you achieve?

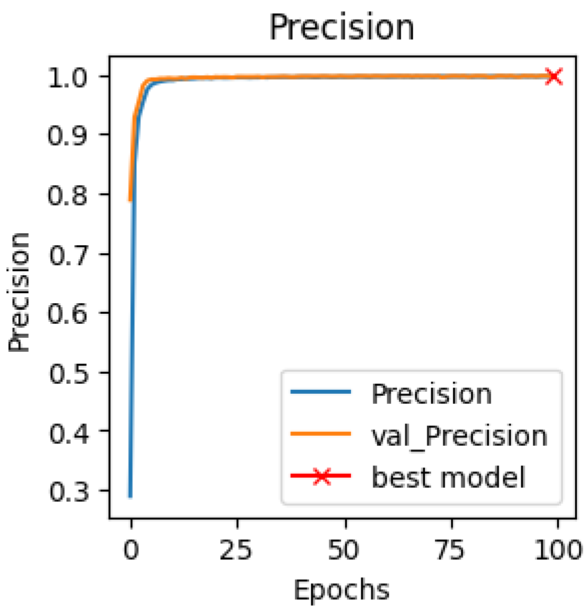
Answer:
From our results, the best model achieved around 0.96 dice value, 0.99 precision and 0.9 recall value.

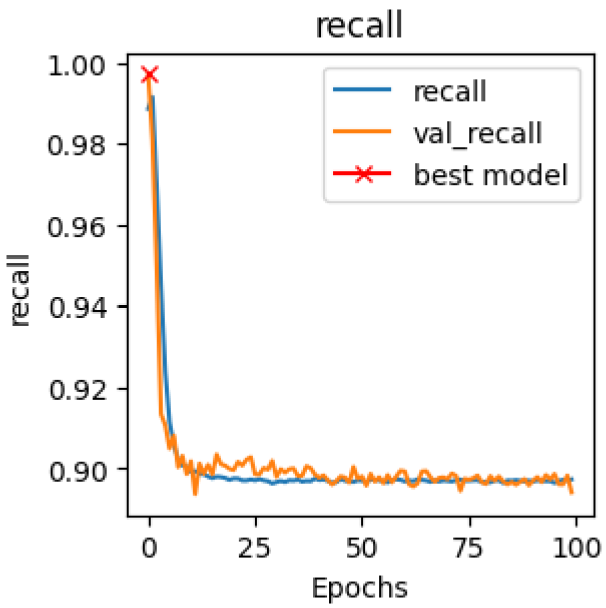


Learning curve
loss
val_loss
best model
0.8
0.6
0.4
0.2
0.0
Loss Value
0
25
50
75
100
Epochs



dice_coef
1.0
0.8
0.6
0.4
0.2
dice_coef
dice_coef
val_dice_coef
best model
0
25
50
75
100
Epochs





Final observations:
Which training modality led to the best performance on your brain tumour dataset (between simple U-Net and U-Net with weighted Dice loss or U-Net with autocontext channel)? What do you think could be additionally tested to improve the performance?

Answer:
From our experience, we conclude that the best performance model is U-Net with weighted Dice loss.(We didn’t implement autocontext with U-Net, so it is unknown to us.)

For additional tests, we would suggest adding new metrics such as volume validation to improve the performance. We checked several prediction from the model and compared it

with the ground truth. Although the performance in dice value is pretty high now, the boundary of the tumour can be slightly different from the ground truth.
So we thought about adding one more dimension to metrics. Now the tumour this model dealing with is mainly in 2D, but in real case they are in 3D. So combining slices of images back to 3D and comparing the volume with ground truth may help improve.