

Lab 5

Task 1 - K-fold cross validation

Settings:

- n_filters: 8
- Optimizer: Adam
- Learning rate: 0.0001
- Dropout: 0.2
- Batch size: 8
- Eval metrics: Dice coeff, precision, recall
- Loss: dice (non-weighted)
- Batch norm: True
- Number of folds: 3
- Dataset: MRI
- Number of epochs: 100

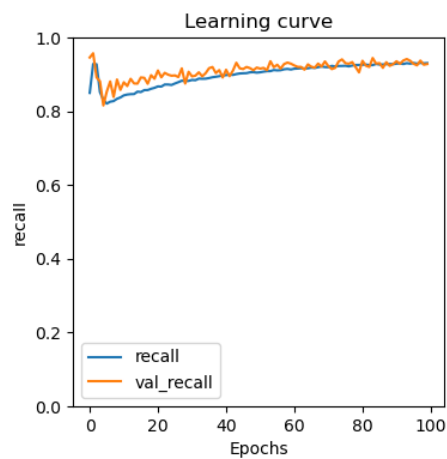
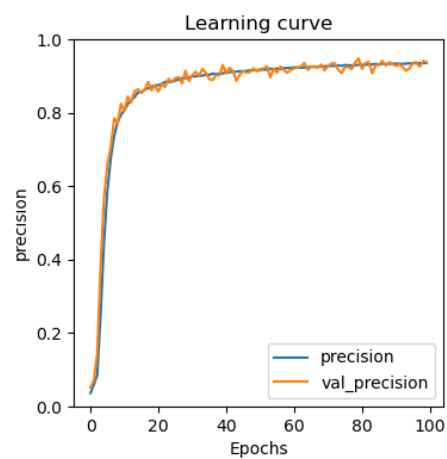
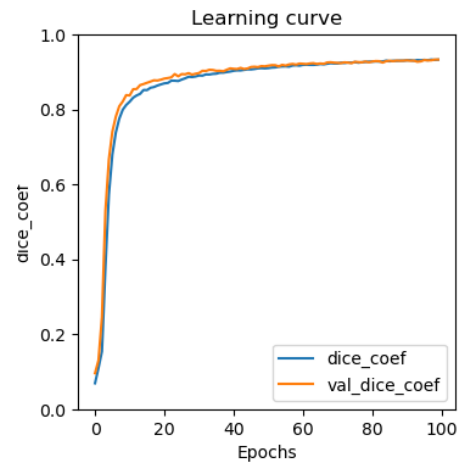
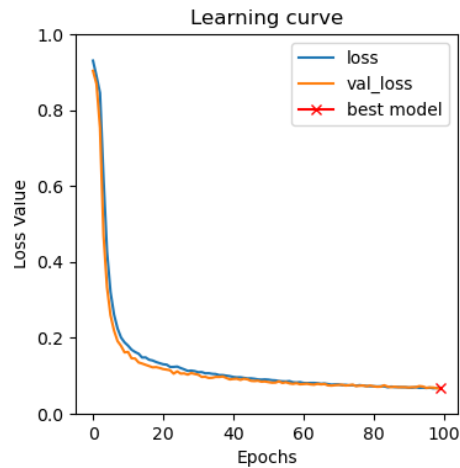
Fold 0

```
dice_coef
dice_coef 0.9326764941215515
val_dice_coef 0.9330580234527588
precision
precision 0.9354605078697205
val_precision 0.9388301372528076
recall
recall 0.9305840730667114
val_recall 0.9280221462249756
```

100th epoch:

loss: 0.0673

val loss: 0.0669



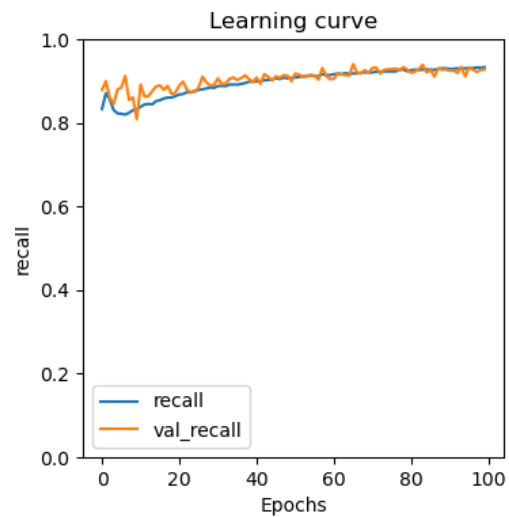
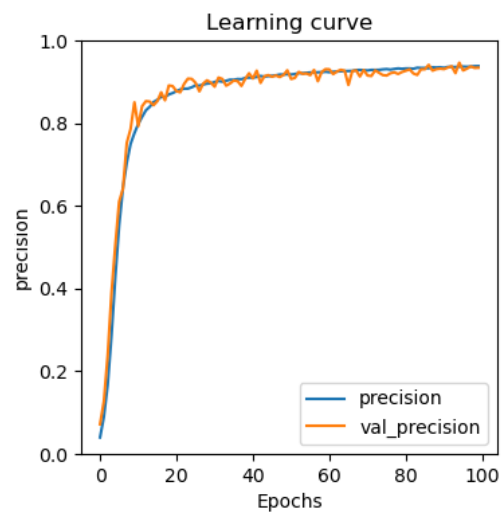
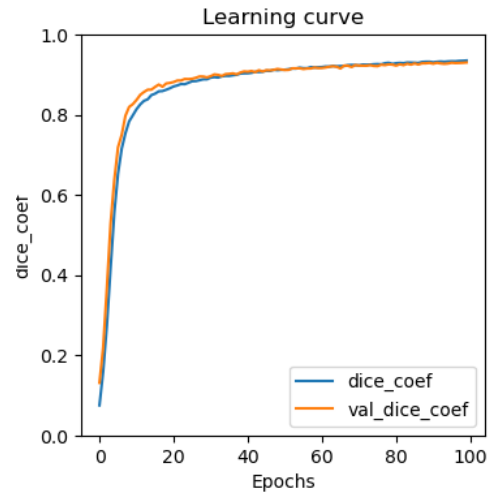
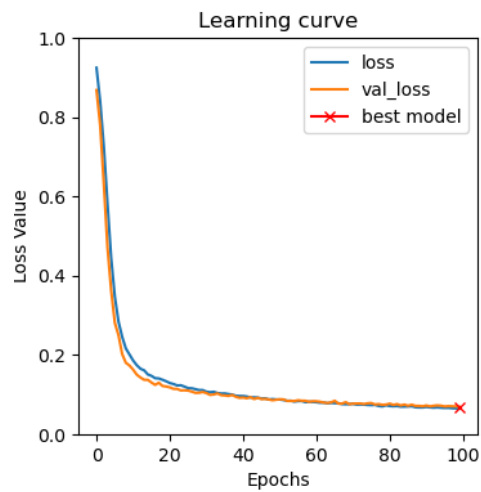
Fold 1

```
dice_coef
dice_coef 0.9351873397827148
val_dice_coef 0.930302619934082
precision
precision 0.9380427002906799
val_precision 0.9340205192565918
recall
recall 0.9330163598060608
val_recall 0.927479088306427
```

100th epoch:

loss: 0.0648

val loss: 0.0697



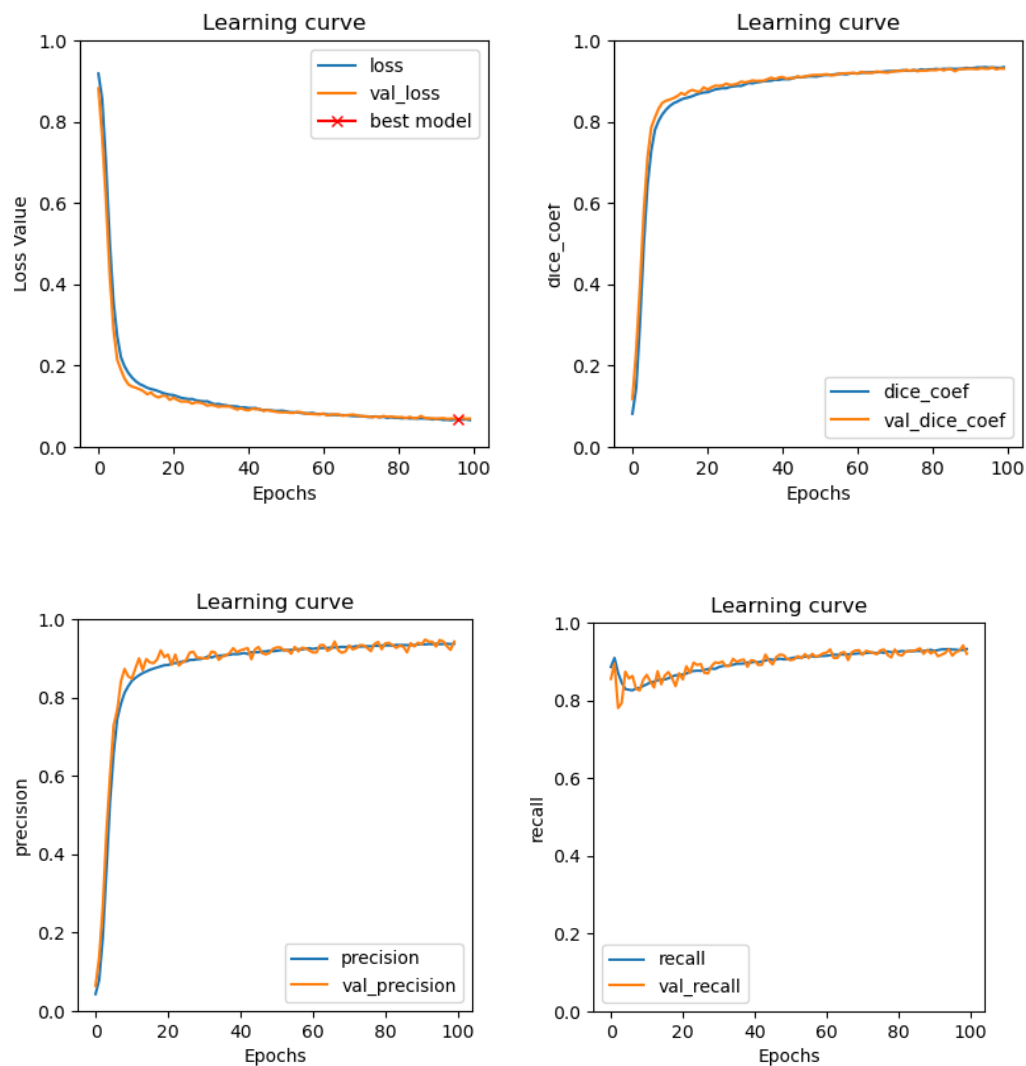
Fold 2

```
dice_coef
dice_coef 0.9341060519218445
val_dice_coef 0.9304676651954651
precision
precision 0.9370064735412598
val_precision 0.9423912167549133
recall
recall 0.9319853782653809
val_recall 0.919845461845398
```

100th epoch:

loss: 0.0659

val loss: 0.0695



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

In 3 K-fold validation for the MRI dataset, the performance of our model is consistent across all 3 folds.

If a dataset has inconsistent performance across folds, that means the dataset could be not evenly distributed (or there's too little of specific classes, or one fold contains more challenging segmentations) and a specific subset of features exists in the train but not in the validation fold. Additional shuffling, data augmentation or addition of weight maps for specific classes might be necessary to resolve this problem.

Task 2 option a)

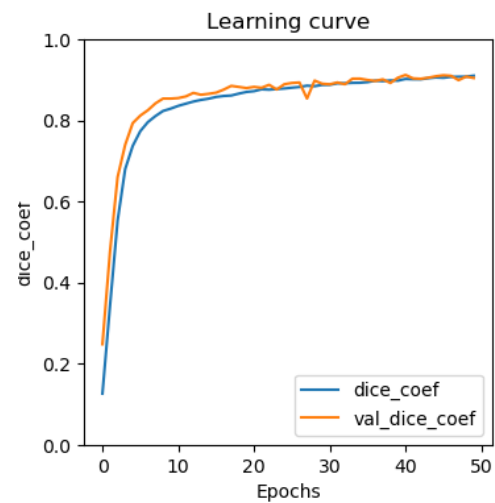
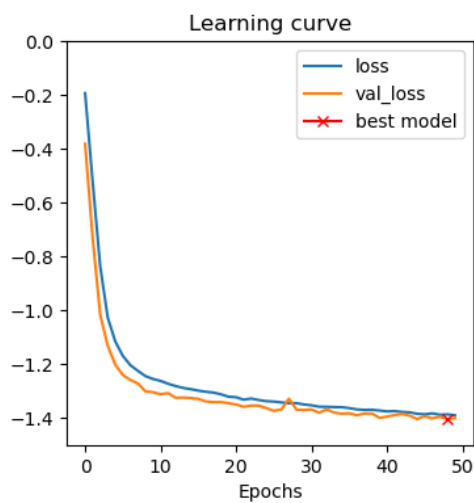
Comparison between using boundary masks. for weighting the loss and using no boundary masks.

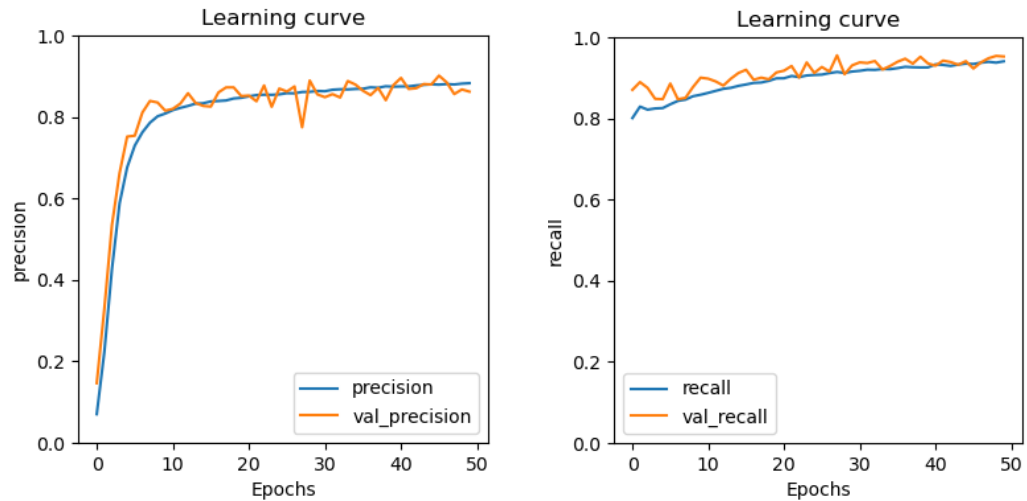
Settings:

No cross validation
 Dropout rate 0.2
 50 epochs
 n_base 8
 validation split 0.2
 learning rate 10e-4
 weight strength 1.5

With boundary masks(weight = mask*1.5):

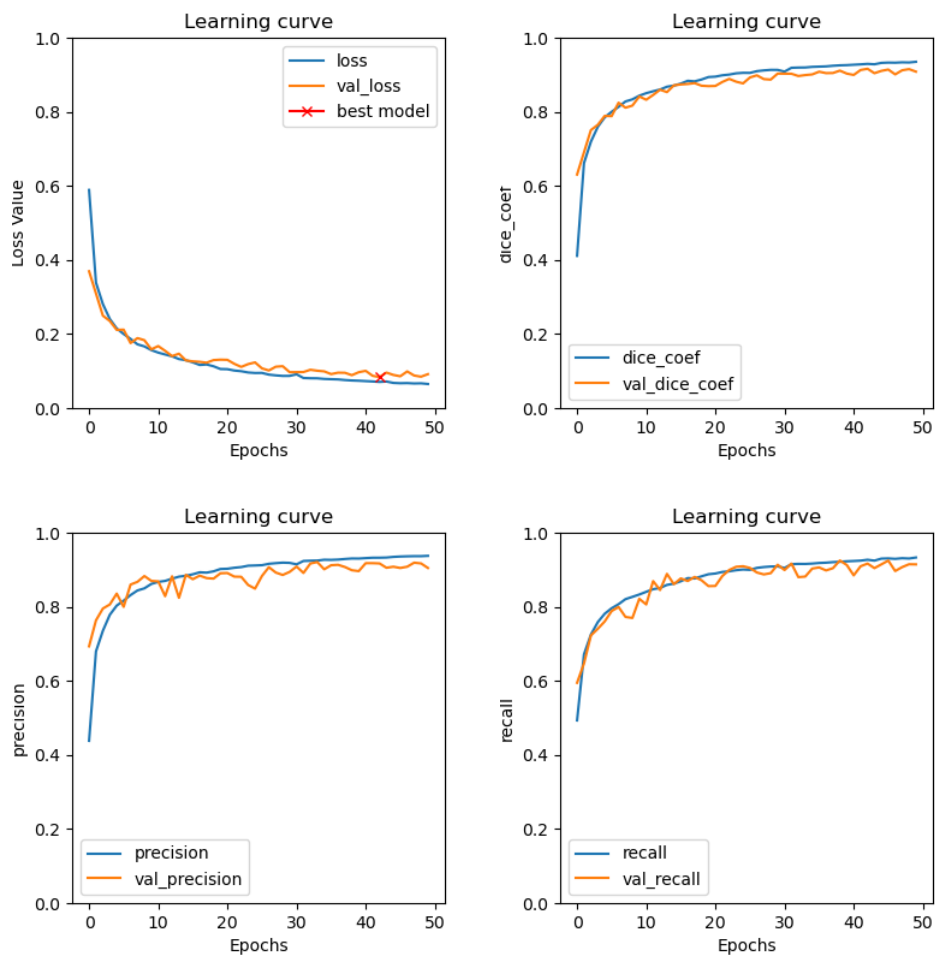
	DSC	precision	recall
Training	0.910	0.883	0.941
Validation	0.904	0.862	0.952





Without boundary masks:

	DSC	precision	recall
Training	0.935	0.938	0.933
Validation	0.905	0.905	0.915



Can you observe a discrepancy between loss function and the traditional (un-weighted) Dice coefficient evaluation metrics? Which accuracy do you achieve?

The validation DSC didn't change substantially. However, the training DSC decreased when using the weighted boundary maps. When using the weighted loss, the validation recall increased substantially from 0.915 to 0.951. However, simultaneously, the validation precision decreased from 0.905 to 0.862. Since we are dealing with tumor segmentations, a higher recall (i.e. True Positive Rate) is more important than a higher precision, as one is interested to find as many of the tumors as possible. Thus, the model with the weighted loss should be preferred. In order to increase the models performance, one could perform a hyperparameter search (e.g. gridsearch or random search) to find more optimal Values for the weight, the learning rate and network architecture parameters.