Chen Siyuan

Darçot Benjamin

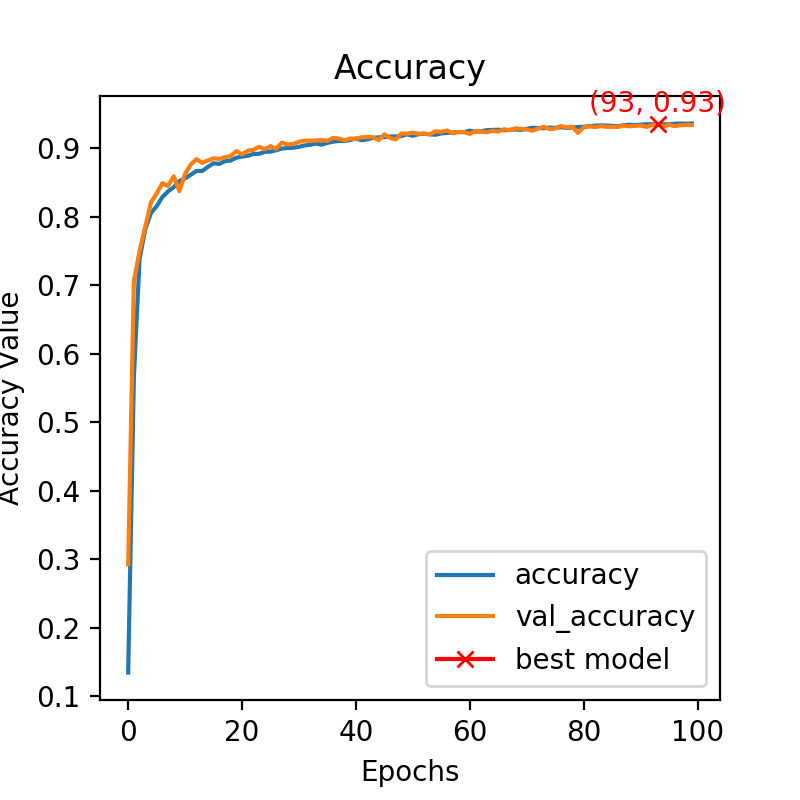
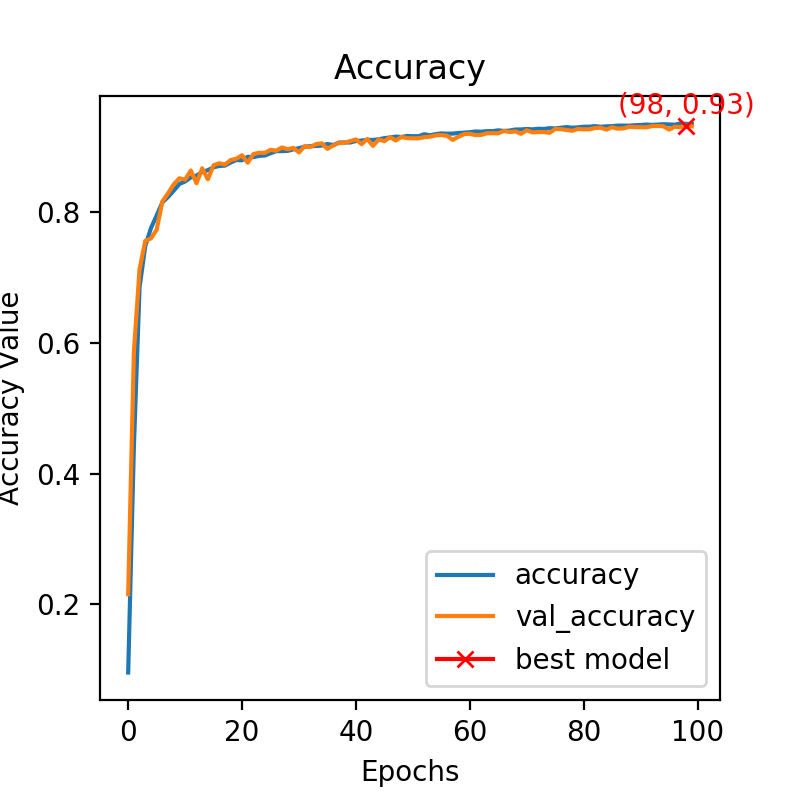
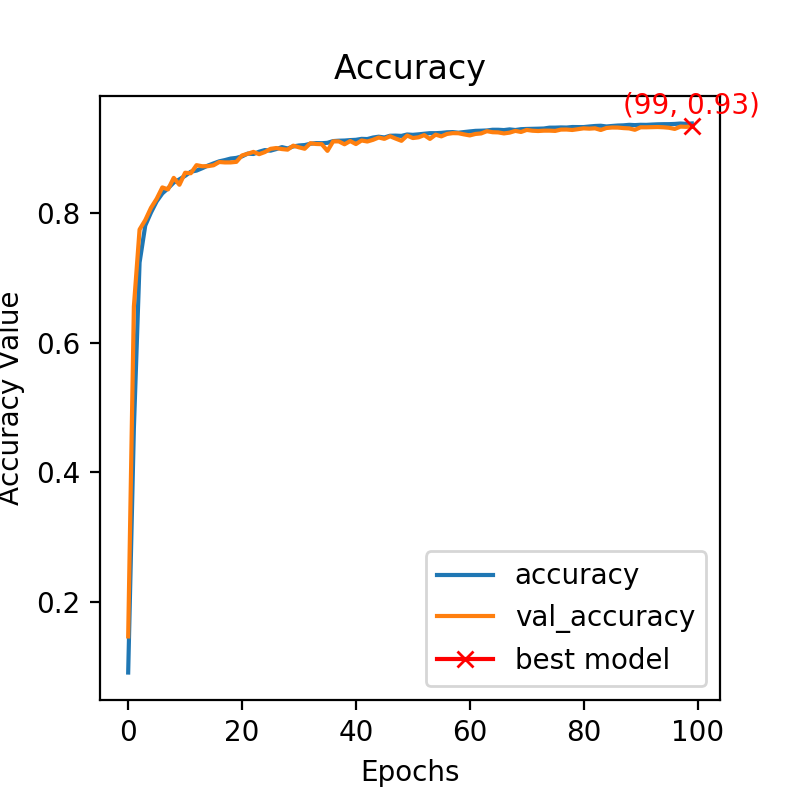
**Lab 5 : Report**

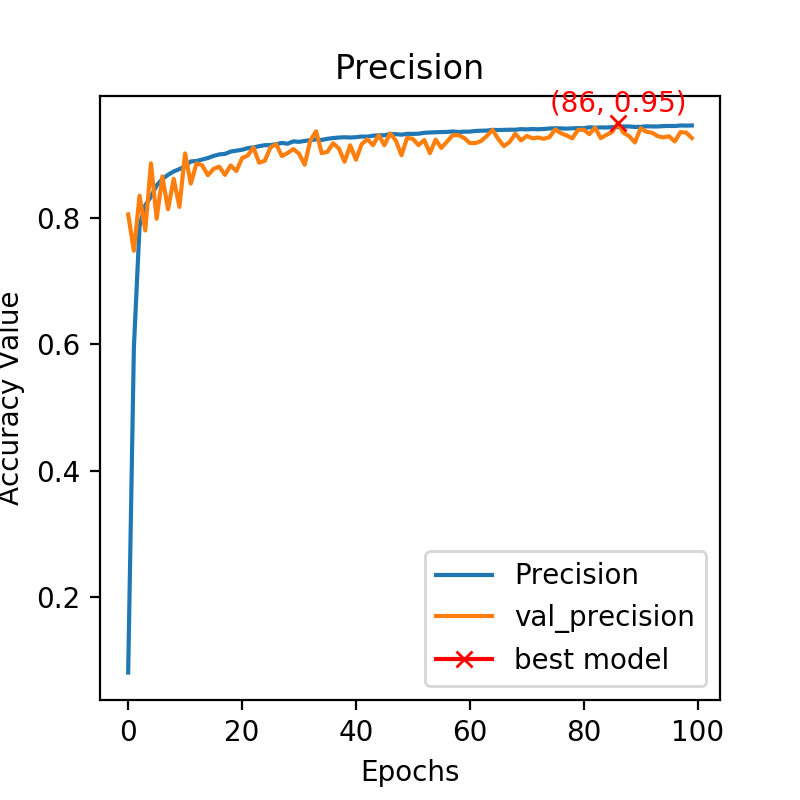
**Task 1**

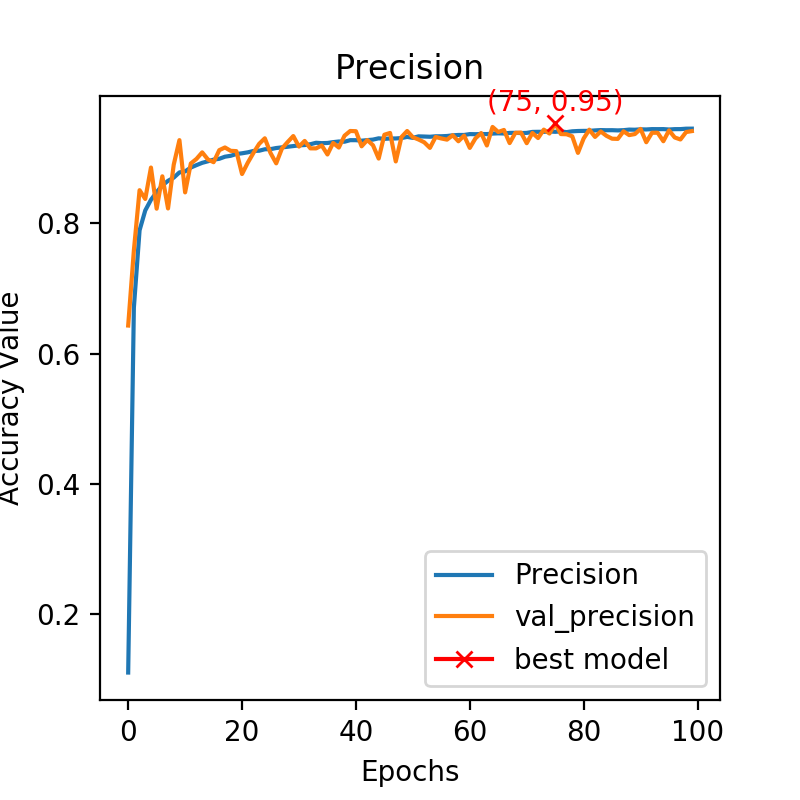
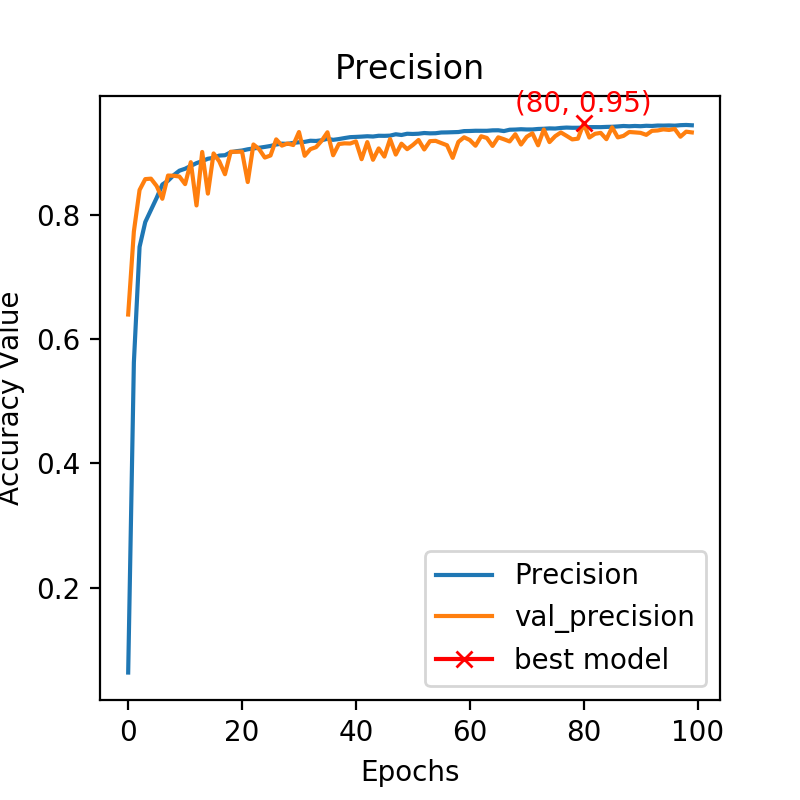
In this first task, the U-net architecture with K-fold cross validation is used to segment brain tumors in MR images. There are three fold for this cross-validation, meaning that the dataset will be splitted three different times : the first fold the first n/3 images will be used as validation set and the remaining 2\*n/3 as training set. Instead, for the second fold, the first n/3 images and the last n/3 images will be the training set. For the last fold, the first 2\*n/3 images of the list will be used as training set.

The parameters for this study are : a base number of 8, a learning rate of 1e-4, a batch size of 8 and a dropout rate of 0,2.

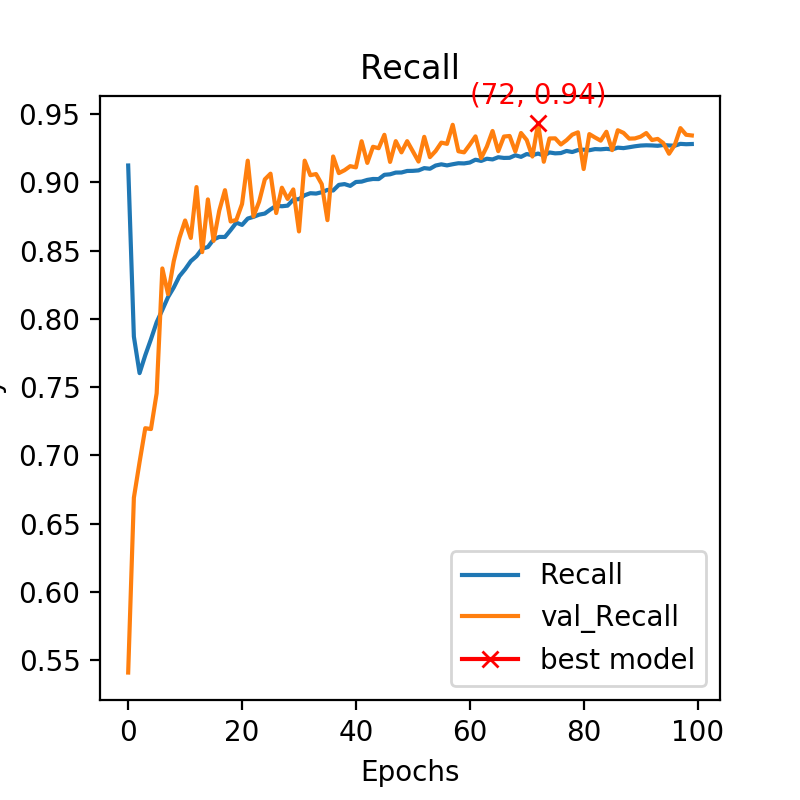
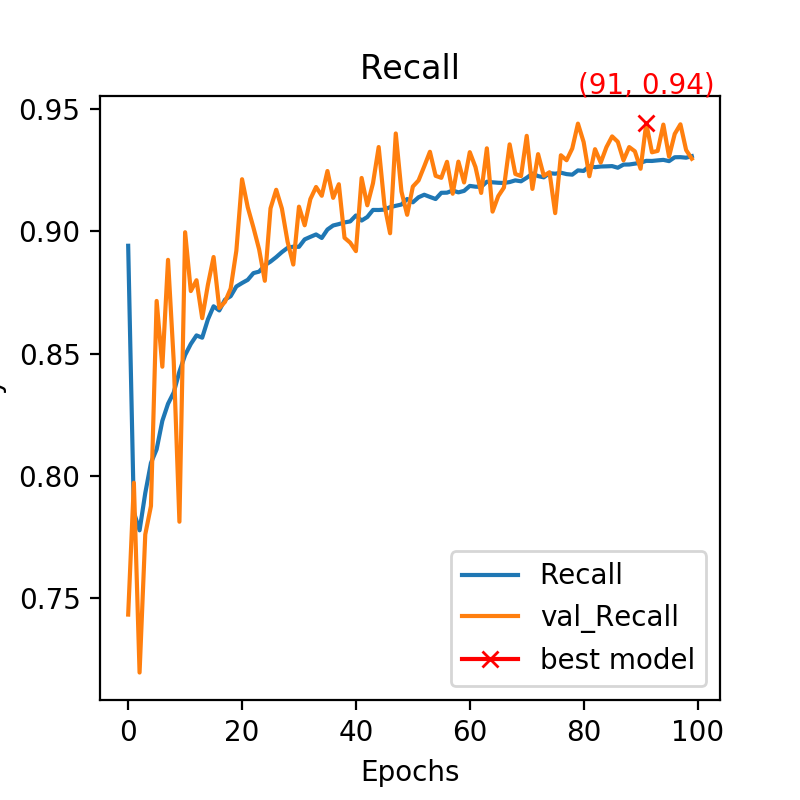
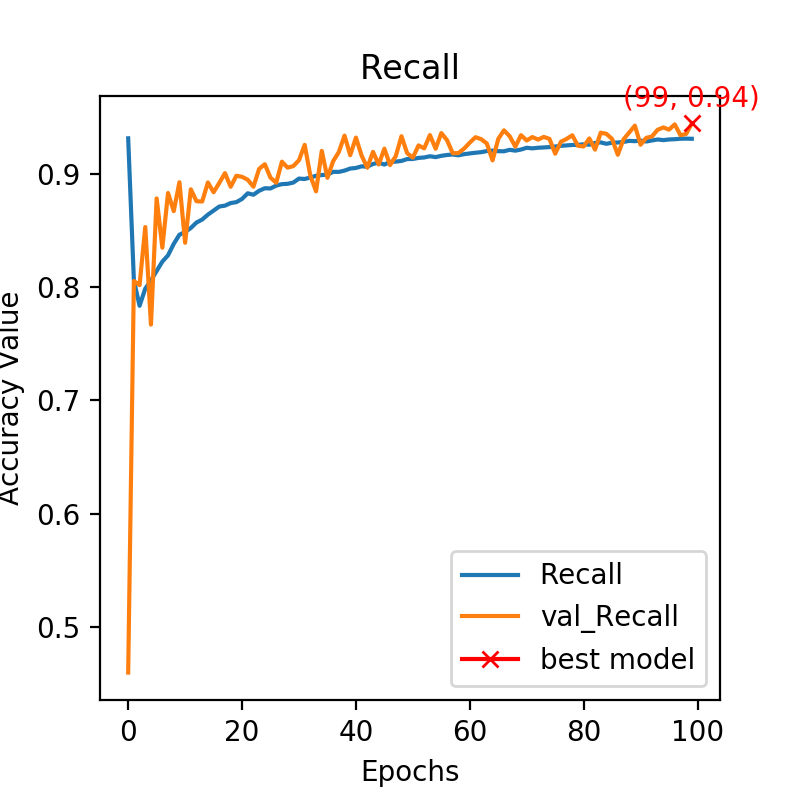
Below are the results for each fold (left = fold 1, middle = fold 2, right = fold3).

**Accuracy**



**Precision**

**Recall**



We can therefore see that the performances are really good for each fold : an accuracy around 93%, a precision of 95% and a recall of 94%.

Furthermore, the results are really consistent across all folds which means that there is a balanced classification. If it wasn’t the case, instead of splitting the dataset completely randomly, we could have splitted it still randomly but in a way that class distribution is the same in every fold, this is called stratification.

**Task 2B**

Unfortunately this task couldn’t be run due to a GPU memory issue in our code. However with autocontext, we expect the results to be better. Indeed, an autocontext layer helps giving more information to the model and therefore having better results. Maybe in this segmentation task, it would not help a lot because the results are already really good, but in general, autocontext is a good way to increase performances.