**Lab 4 - Report**

***(Note: When we are talking about loss & accuracy, we are referring to the validation values)***

**Task 0**

**Answer)** See submitted code**.**

**Task 1a**

**Answer)** Final result were, BCE loss & DICE accuracy of 0.0522 and 0.9572.

**Task 1b**

**Answer)** We could observe an improvement in the final results which were, BCE loss & DICE accuracy of 0.0347 and 0.9652.

If the target size is smaller than the image size, the dice accuracy became small (far from 1) and would thus negatively impact the training (?). In this case we would use binary cross entropy.

**Task 1c**

**Answer)** **With binary cross entropy**, the model performance decreases. The final results were BCE loss & DICE accuracy of 0.0568 and 0.9494. **With dice loss** the model performance improved. The final results were the DICE loss & DICE accuracy of 0.0325 and 0.9675. The learning curves looks good in both cases. No overfitting occurred and both learning curves converged.

**Task 1d**

**Answer) With binary cross entropy,** the model performance decreases. Similar to Task 1c. The results we got were BCE loss & DICE accuracy of 0.0523 and 0.9701. **With dice loss,** the model performance improved again. The results we got were DICE loss & DICE accuracy of 0.0271 and 0.9728. The learning curves could be seen to overfit slightly in both cases.

**Task 1e**

**Answer)** When decreasing the number of feature maps, applying batch-normalization and data augmentation techniques, the results are worsened for both loss functions. The learning curve indicates overfitting. The final results **for binary cross entropy** were loss & DICE accuracy of 0.3236 and 0.6500. The final results **for dice loss** were DICE loss & DICE accuracy of 0.2465 and 0.7535.

**When training the network with data augmentation and no augmentation on the validation data we. Then we get pretty decent results with. Final results: loss: 0.1635 - dice\_coef: 0.8365 - val\_loss: 0.2883 - val\_dice\_coef: 0.7117 (???)**

**Task 2a**

**Answer)** As a result of memory problem we could unfortunately only use 4000 images. Any more than that and the kernel would die mid-training. The problem was probably due to a lot of student groups not shutting down their kernels after running experiments.

The final results we got were a DICE loss & DICE accuracy of 0.5826 and 0.4174. The learning curve indicated heavy overfitting.

#val\_loss: 0.5493 - val\_dice\_coef: 0.4507

**#val\_loss: 0.7934 - val\_dice\_coef: 0.2066 (extreme overfitting)**

**Task 2b**

**Answer)** The after final epoch were loss & DICE accuracy of 0.3761 and 0.6239. The precision & recall were 0.9003 and 0.6540.

Precision refer to the number of true positives divided by the sum of true positives and false positives. Thus, it gives us an idea of how many pixels we predict have a value of 1, actually have the value of 1 in the mask. We may well have predicted that a pixel has a value of 1, whilst in reality the mask has a value of 0 in that corresponding pixel.

Recall refer to the number of true positives divided by the sum of true positives and false negatives. Thus, this metric gives us a sense of how many pixels with value 1 in the mask we have missed. We may well have predicted that a pixel has a value of 0, whilst in reality the mask has a value of 1 in that corresponding pixel.

**Task 3**

**Answer)** For the same reasons as in task 2, only 4000 images were used. 3200 of these images were used for training and 800 images were used for validation, a so-called 80/20 split.

The results after final epoch, **using categorical cross-entropy loss**, were loss & DICE accuracy of 0.5503 and 0.7445. The precision & recall were 0.7350 and 0.8490