Antwoorden vragen week 4

**Number of parameters**

The first convolutional layer in the MNIST example has 320 parameters. The first fully connected layer has 1179,776 parameters. What do these parameters correspond to?

***The number of parameters are the learnable elements for a filter, in general they are the weights that are learnt during training.***

What is the general expression for the number of parameters of 1) a convolutional layer and 2) a fully-connected layer?

***Convolutional: ((kernel\_size)\*stride+1)\*filters)***

***Fully-connected: with kernel\_size = n, m 🡪 (n+1)\*m***

**Fully-convolutional MNIST model**

Modify the model in the MNIST example in such a way that it only contains convolutional layers while keeping the same number of parameters. If you do the modification correctly, the two models will have the same behaviour (i.e. they will represent the same model, only with different implementation). Show this experimentally.

**U-Net architecture**

What is the role of the skip connections in the U-Net neural network architecture? Will it be possible to train the exact same architecture with the skip connections omitted? If yes, what would be the expected result? If no, what would be the cause of the error?

***The skip-connections make it possible to recover fine-grained details in the final prediction.***

***If we would omit the skip connections we can train the architecture, but it would not be a ‘ u-net’ anymore, it then looks like a normal convolutional neural network. This causes the fine-grained details to be gone in the final prediction (??)***

**Data augmentation**

Why does data augmentation result in less overfitting? Can data augmentation be applied to the test samples? If yes, towards what goal? If no, what is preventing that?

***Data augmentation result in less overfitting, since the model is trained on data that are all a little bit different. This causes the model to learn features more robust.***

***Applying data augmentation to the test samples is possible, but it will not add any value. You don’t train the model with the test dataset, you only use it to evaluate the model. Evaluating the model on the augmented test data set only gives you more evaluation metrics. These metrics might be more robust since there are differences in the test data itself, but this is only useful when the augmented images also make sense in real-life. The image after data augmentation applied should thus still reflect real-life images.***

Implement random brightness augmentation of the image data by adding a random offset to the image intensity before passing them trough the network at training time. Train a model with random brightness augmentation and compare it to the baseline above.

Implement data augmentation procedure that in addition to brightness augmentation also performs b-spline geometric augmentation using the [gryds](https://github.com/tueimage/gryds) package (you can look at the documentation of the package for an example on how to do that). Compare the new model with the baseline and the model that only performs brightness augmentation.