Comparison of the MultiRes U-Net and the classical U-Net on the performance of kidney and kidney tumor segmentation

Laura Mak

Student number: 5541476

l.e.m.mak@tilburguniversity.edu

Thesis submitted in partial fulfillment of the requirement for the degree of master's in science in Data Science and Society, at the Tilburg University

Thesis Supervisor:

Juan Sebastian Olier Jauregui

Tilburg University

School of Humanities and Digital Sciences

Tilburg, The Netherlands

June 2020

Abstract

Convolutional Neural Networks are the state-of-the-art networks for biomedical image segmentation. With the increasing number of medical images produced, there is a need to automatically detect any lesions or abnormalities in these images. Recently, there have been many adaptations on the popular U-Net for performing these tasks. This paper aims at evaluating one of these adaptations called the MultiRes U-Net. In this paper, the 2D version of the model is tested based on a medical dataset of a different modality, and thereby contributed in extending the evaluation of this model. Other literature claims that pre-processing could be important than architectural modifications. Therefore, this research paper compares the performance of the MultiRes U-Net with a classical U-Net. Additionally, this paper evaluated effect on the U-Net of data pre-processing by pixel value normalization, clipping and different data augmentation techniques. The result shows that the classical U-Net performs better based on the Dice Coefficient, even when there was no pre-processing involved. The MultiRes U-Net shows superiority when the area of interest in the medical images show a high variability in terms of size, location and the presence of noise. The fact that this variability is lacking in the dataset used for these experiments can be a reason why the MultiRes U-Net does not perform better. Future research should conduct experiments to see whether this claim holds.

Introduction

Convolutional neural networks (CNNs) are a type of deep learning algorithm widely used to automate classification tasks and semantic segmentation tasks. Especially the latter has been extensively researched and applied for the use of biomedical image segmentation. In the past five years, there have been many publications on variations in terms of CNN architectures. The U-Net architecture has been taking the lead in image segmentation based on various medical data. This paper is aimed at researching whether an adaption of the U-Net architecture, named MultiRes U-Net, outperforms the classical U-Net architecture in the task of biomedical image segmentation.

Medical imaging, like Magnetic Resonance Imaging (MRI), Computerized Tomography or X-rays, became a very helpful method in diagnosing various diseases or detecting anomalies (Meinzer et al., 2002). The increase in information that needed to be manually processed by specialists, displayed the need for an automation method to shorten this time-consuming process. CNN's have been successfully used for this form of automatization and thereby performing various tasks such as segmenting neuronal membranes (Ciresan et al., 2012), segmenting skin lesions (Adegun & Viriri, 2019), red lesion detection in fundus images (Orlando et al., 2018) and brain tumor segmentation (Havaei et al., 2017). These studies, and many more, have created a level of playing field in which researchers are continuously publishing models that are improving upon the state-of-the-art models for biomedical image segmentation. Within this level of playing field, there have been many "Biomedical Grand Challenges" which provided comparison of the newly proposed architectures with the state-of-the-art architectures. One of these challenges is the 2019 Kidney and Kidney Tumor Segmentation challenge (KiTS 2019). The winning team developed and trained three different U-Net inspired architectures: a 3D 'plain' U-Net, a residual

3D U-Net and a pre-activation residual 3D U-Net (Isensee & Maier-Hein, 2019). In their findings, they stated that architectural modifications do not significantly improve segmentation results. They showed that data pre-processing could have a larger impact than implementing a novel U-Net architecture. The authors claimed that augmenting the U-Net with a lot of 'bells and whistles' does not necessarily lead to better model performance.

On the contrary, Ibtehaz and Rahman (2020) proposed such an architectural modification, called MultiResUNet, and showed promising results in their research paper. The model has been tested by its developers on a variation of datasets with different modalities: Fluorescence Microscopy, Electron Microscopy, Dermoscopy, Endoscopy and MRI. The authors claimed that the MultiRes U-Net model outperformed the classical U-Net. They supported their claim by providing a comparison of model performances on these different datasets. However, they also stated that the MultiRes U-Net should be subjective to further evaluation on different modalities. Therefore, this paper aims to address whether the MultiRes U-Net model outperforms the classical U-Net on the task of binary pixel classification by using a dataset containing Computerized Tomography scans of patients suffering from kidney tumors. The details of this dataset will be set out more extensively in the section *Experimental Setup*. In order to keep the research limited, only this particular dataset will be used.

Research questions

This paper will address the following research question: "To what extent does the use of non-architectural modifications have more significant impact on the semantic segmentation of kidney tumors, than the architectural modification of adding Res paths and MultiRes blocks to the U-Net?"

Subsequently, this research question raised some sub-questions. The following sub-questions related to the research question will be addressed:

1. How does the 2D MultiRes U-Net model perform without any (domain specific) pre-processing?
2. How does a classical 2D U-Net model perform without any (domain specific) pre-processing?
3. How does pre-processing the data by normalizing the pixel values affect the performance of the U-Net?
4. How does pre-processing the data by clipping the Hounsfield unit values affect the performance of the U-Net?
5. How does the process of data augmentation affect the performance of the U-Net?

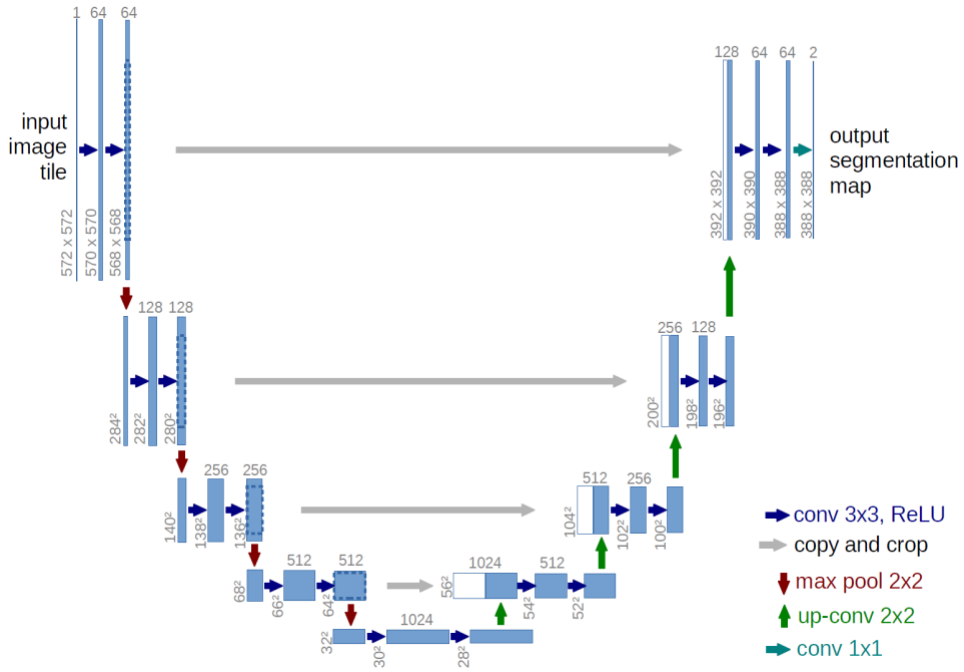
The main finding of this research paper is that the MultiRes U-Net did not outperform the classical U-Net on the task of segmenting kidney and tumor in CT scan data. This could be due to the fact that the dataset is more homogeneous, whereas the datasets used by Ibtehaz et al. (2020) are very heterogeneous in terms of target size, target location and noise.

Related Work

Recent research has been focused on the use of neural networks for biomedical image segmentation (e.g. kidney or brain tumor segmentation) and classification (e.g. normal or abnormal lungs), since manual annotation can be complex and time-consuming. The medical images can have different modalities such as magnetic resonance imaging (MRI), X-ray or computed tomography (CT). An early application of deep neural networks on biomedical image segmentation is reported by Ciresan, Giusti, Gambardella, and Schmidhuber (2012). In this study, they used a 2D convolutional neural network (CNN) for segmentation by performing patchwise image classification in a sliding-window setup (Kayalibay, Jensen, & van der Smagt, 2017; Ciresan, Giusti, Gambardella & Schmidhuber, 2012). This method classified each pixel by extracting a patch (local region) around it. Even though this method yielded good results, it was computationally expensive, as it must be runned separately for each patch. Furthermore, this modeling approach did not exploit the contextual information to its fullest. Ronneberger, Fischer and Brox (2015) introduced the U-Net architecture, built upon the Fully Convolutional Network (FCN) introduced by Long, J., Shelhamer, E., & Darrell, T. (2015). An FCN is built only from locally connected layers, such as convolution, pooling and upsampling. The main differences of the U-Net compared to the FCN are the symmetry of the U-Net and the use of skip connections (Ronneberger et al., 2015). The 2D architecture has a set of convolutional layers to perform the segmentation tasks and consists of a contracting path (the decoder), the bottleneck and an expansive path (the encoder). The decoder captures the context of the input image, which is essential for accurate segmentation. The bottleneck mediates between the decoder and the encoder. Finally, the encoder enables precise localization combined with contextual information from the contracting path (Ronneberger et al., 2015).

Figure 1

U-Net Architecture by Ronneberger et al. (2015)



The network architecture was able to successfully segment 2D images (even of different sizes) with low amounts of data. However, as many biomedical images are in 3D, such as brain MRI or liver CT, an adaptation of the architecture was necessary. One of the challenges posed by a 3D image is the higher memory intensity during training (Heller et al., 2019). Çiçek et al. (2016) introduced a 3D U-Net architecture that could successfully perform volumetric biomedical segmentation tasks.

There have been many more variations of the U-Net since the introduction of the popular architecture. Alom et al. (2018) introduced the R2U-Net using retina blood vessel segmentation, skin cancer lesion segmentation, and lung segmentation. Oktay et al. (2018) proposed the Attention U-Net, which is able to learn to focus on target structures in order to cope with varying target sizes. Another variation called the H-DenseUNet (Li et al., 2018) is able to extract the

features from 3D volumes while maintaining the computational efficiency of having a 2D network. Furthermore, Zhou et al. introduced in 2019 the Unet++ using six different medical image segmentation datasets. In addition to these research papers on architectural variations of the U-Net for biomedical image segmentation, there have been many "Biomedical Grand Challenges". As mentioned before, they provide comparison of the newly proposed solutions to biomedical image analysis problems with the state-of-the-art in a competing manner. One of these challenges was held in 2019: The Kidney and Kidney Tumor Segmentation challenge (KiTS 2019). Since 2014, the new standard in treating kidney cancer has been partial nephrectomy, in which only the tumor is removed instead of the kidney as a whole (Heller et al., 2019). Therefore, segmentation of the tumor related to the kidney is even more important. The KiTS19 challenge aimed to speed up the process of automatic segmentation by providing a dataset of 210 CT images. Participants were ranked by their average Sørensen-Dice coefficient between the kidneys and tumors based on an additional test set of 90 cases. The winning team developed and trained three different U-Net inspired architectures: a 3D 'plain' U-Net, a residual 3D U-Net and a pre-activation residual 3D U-Net (Isensee & Maier-Hein, 2019). The residual 3D U-Net seemed most promising, obtaining the highest average Sørensen-Dice score. They noted that architectural modifications do not significantly improve segmentation results, but their lack of improvement could be due to possible bad implementation of the residual networks or possible bad choice of hyperparameters. Furthermore, they stated: "While this work was designed to achieve maximum segmentation accuracy, future work should be directed towards more extensively evaluating potential differences between the 'plain' U-Net and its architectural variants. Such a comparison should include extensive hyperparameter optimization for each of the architectures and also make a thorough statistical analysis of the results" (Isensee & Maier-

Hein, 2019, p. 7). This statement will be further discussed after the introduction of the MultiResUNet. Another paper emphasized the importance of taking away complicated augmentations of the U-Net and focused on other aspects that influence the model performance and generalizability (Isensee et al., 2018). Isensee et al. (2018) proposed a nnU-Net ("no-new-Net") and tested the performance on seven different datasets (in the context of the Medical Segmentation Decathlon) containing biomedical images. When presented with a new dataset, the nnU-Net will automatically take care of the entire experimental pipeline. Their code has been made publicly available on GitHub.

In contrast with the statement of the winning team in the KiTS19 challenge, Ibtehaz and Rahman (2020) propose a novel architecture. This architecture, called the MultiRes U-Net, builds upon the classical U-Net. Ibtehaz et al. (2020) stated that these improvements upon the U-Net architecture could significantly surpass the state-of-the-art architecture, in the area of biomedical image segmentation. In biomedical image segmentation, one of the challenges is the variation present in the objects of interest, e.g. the size of a brain tumor. To overcome this problem, Ibtehaz and Rahman (2020) replaced the convolutional layers with Inception-like blocks. Within these blocks, they added three convolutional filters in parallel of size 3×3, 5×5 and 7×7 and concatenated the generated feature maps (see Figure A1 in Appendix A). This would increase the U-Nets performance on images at different scales, but likewise increased the memory requirement. To overcome the latter problem, they replaced the demanding 5×5 and 7×7 convolutional layers by a sequence of smaller 3×3 convolutional blocks (see Figure A2 in Appendix A). Even though this modification greatly reduced the memory requirement, they changed the size of the filters in an increasingly from 1 to 3. Furthermore, they added a residual connection because of their proven efficiency in biomedical image segmentation (Drozdzal et al.,

2016). Finally, they added a 1×1 convolutional layer to account for additional spatial information. This modification is called a *MultiRes block* (see Figure A3 in Appendix A) (Ibtehaz and Rahman, 2020) .

Another main attribution of the U-Net is the use of skip connections. A skip connection is a connection that goes around at least one layer (Long, Shelhamer, & Darrell, 2015). The purpose of this connection is to keep information that is captured in the initial layers. This information is required for reconstruction during the up-sampling path. According to Ibtehaz et al. (2020), a drawback of the skip connections is the fact that the first connection bridges the encoder with the decoder before the first pooling and after the last deconvolution operation. They observed a possible semantic gap between the two sets of features being merged, since the features from the decoder contains low level information and the features from the encoder contains high level information. To reduce this semantic gap, they introduced convolutional layers along the skip connections. In addition, they added residual connections to make learning easier. Instead of just concatenating the feature maps from the encoder path to the decoder path, they are passed through a chain of convolutional layers with residual connections. This process is called a *Res path* (see Figure 4 in the Appendix). By incorporating both the MultiRes block and the Res path, they introduce the MultiRes U-Net architecture (see Figure 5 in the Appendix).

In testing their novel model architecture, Ibtehaz et al. used different biomedical image datasets with different modalities (see Table 1 in the Appendix). They aimed to improve upon the classic, state-of-the-art U-Net. Consequently, they based the classical U-Net on the implementation by Ronneberger et al. (2015). The used the U-Net with a five-layer deep encoder and decoder as a baseline. For the 3D U-Net implementation, they used the architecture described by Çiçek et al (2016), previously mentioned, as a baseline. Their final result showed an

improvement on all modalities, based on the Jaccard Index. A second observation worth mentioning was the proven reliability and the robustness of the proposed model, based on the lower variability of the MultiRes U-Net. Furthermore, due to the implementation of the Res paths, the novel model seemed less affected by various types of noises in the case of homologous regions and textures and perturbations in the case of heterogeneous regions. Lastly, the MultiRes U-Net appeared more reliable against outliers. They experimented with the Endoscopy dataset to test the effect of the Res path and the MultiRes blocks on the U-Net separately, and to test the effect of them combined. The combination of both the Res path and the MultiRes blocks yielded the best results (Ibtehaz and Rahman, 2020).

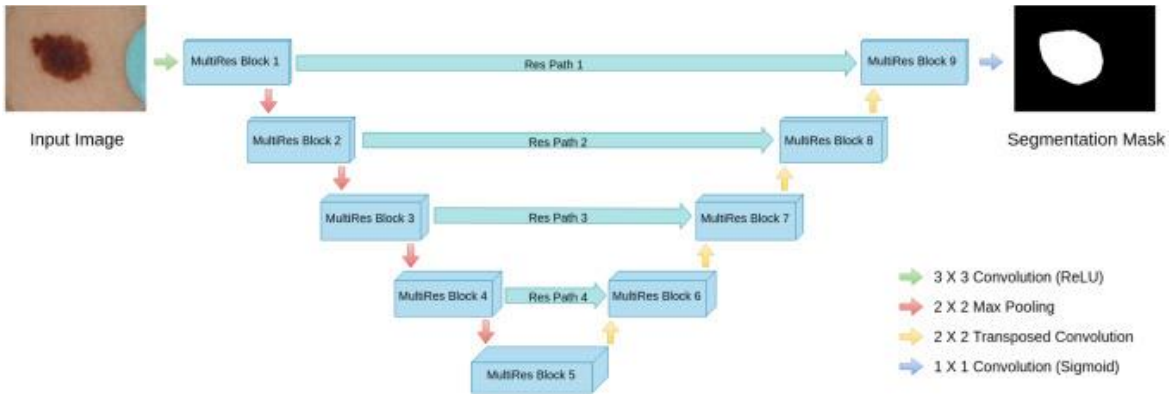
Methods

This research paper aims to test the novel MultiRes U-Net on the performance of binary image segmentation of the KiTS19 dataset. Çiçek et al. (2016) showed the effectiveness of a 3D segmentation U-Net. However, Vu et al. (2019) provided us with a comparison of using a regular 2D CNN, a pseudo 3D CNN and a volumetric 3D CNN on five different datasets for image segmentation, including the KiTS19 dataset. Their result showed that neither the volumetric 3D CNN nor the pseudo 3D CNN outperformed the regular 2D CNN on the task of binary image segmentation. Therefore, in order to save computational costs and due to the limitation in available resources, only the 2D architecture of the MultiRes U-Net and U-Net will be used. The depth of both the networks will be similar, in order to provide accurate comparison. As described above, the MultiRes U-Net should be able to close the semantic gap between the two sets of features being merged, by using convolutional layers along the shortcut connections of the

widely used U-Net. Furthermore, the two consecutive 3×3 convolutional layers are replaced by Inception-like blocks in order to overcome the problem of different scales of input data.

Figure 2

The 2D MultiRes U-Net Architecture



Note. This figure displays the architecture of the model proposed by Ibtehaz et al. (2020). Details of the *MultiRes Block* and the *Res Path* can be found in Appendix A.

The number of parameters W is calculated according to the number of filters in the classical U-Net and a scalar (1).

$$W = \alpha \times U \tag{1}$$

Here, W is the number of parameters, α the scalar coefficient, and U the number of filters corresponding to the classical U-Net. The classical U-Net has five different number of filters along the layers: 32, 64, 128, 256 and 512. In order to keep the number of parameters the same as in the classical U-Net, Ibtehaz et al. (2020) set α at 1.67. This resulted in 7,262,504 parameters (W) which is a bit less than the number of parameters in the classical U-Net (see Figure 3). After each pooling or deconvolution layer, the number of filters get doubled, the same as in the

classical U-Net architecture. The gap between the encoder and the decoder decreases when the model converges to the center of the network. Consequently, the MultiRes U-Net needs fewer convolutional blocks along the Res paths. Figure A3 of Appendix A displays the architecture details. These values are kept from the original paper by Ibtehaz et al. (2020) in order to have an accurate comparison. The convolutional layers and the output layer in both the models are activated by the Rectified Linear Unit function and Sigmoid function, respectively.

Figure 3

Number of Parameters

```
Total params: 7,262,504          Total params: 7,773,249
Trainable params: 7,237,982      Trainable params: 7,773,249
Non-trainable params: 24,522     Non-trainable params: 0
```

Note: output summary of the total parameters of the MultiRes U-Net and the U-Net model. The number of parameters for the MultiRes U-Net are calculated by (1).

Experimental Setup

Dataset description

The dataset used in this experiment, is the KiTS19 challenge dataset. The University of Minnesota and the University of Melbourne organized the KiTS19 challenge, releasing the dataset on March 15, 2019. They selected 554 patients who underwent partial or radical nephrectomy for suspicion of renal cancer by a physician in the urology service of University of Minnesota Health. After this selection, they excluded patients that either had a cyst instead of a tumor or a tumor thrombus, and those for whom imaging in the late arterial phase was not available. The dataset ended up consisting of 300 computerized tomography (CT) scans, collected across more than 50 institutions. The scans were converted from DICOM format to

Nifti using Pydicom and Nibabel. Finally, 210 cases were manually annotated and the other 90 were made available as a separate test set. In addition, the patient characteristics were made available. The guidelines for downloading the dataset have been made publicly available on Github. Every CT scan has a size $N \times 512 \times 512$, with N varying from 29 to 1059 and a segmentation label with corresponding size. The Hounsfield unit (HU) values of the CT scans range typically between -1024.0 until 3000.0 but the dataset has some outliers with a minimum value of -14889.0 and a maximum value of 18558.0. The corresponding segmentation labels have a value of [0, 1, 2], representing the background, kidney and tumor respectively.

Experimental procedure

This paper will compare the performance of the 2D MultiRes U-Net model with a standard 2D U-Net model. The model will be implemented using Keras with Tensorflow backend in Python 3. For computational efficiency, the available Tesla K80 GPU of 12GB on Google Colaboratory will be used. Furthermore, this research will be using the most recent versions of Nibabel, NumPy, Scipy and Scikit-Image.

The data will be loaded with Nibabel and resized with the NumPy resize function to size ($N \times 128 \times 128$) for computational efficiency. As the main goal of this paper is to test the novel architecture on the task of binary pixel classification, the segmentation labels are converted to binary labels. In the converted labels, 0 represents the background and 1 represents kidney and tumor tissue. As both the model architectures are comprised with 2D convolutions, the 3D data needs to be transformed to fit the 2D model. This transformation is done by taking slices of the 3D data and convert them to NumPy arrays. Hereby, the slices are maintaining the original first dimension order (the z-axis) to mimic the volumetric data. Another dimension will be added in order to feed the data to the network.

This paper aims to test the superiority of the proposed MultiRes U-Net over the classical U-Net on CT scan data. Consequently, no domain specific pre- or post-processing techniques were used, and the CT scans are only normalized between range [0, 1]. Faulty cases (case 23, 68, 125 and 133) were detected and confirmed by the KiTS19 challenge organizers (Heller et al., 2019). Therefore, these cases were not part of the final dataset. There was a difficulty in deciding whether to keep the slices that did not contain any target values. On the one hand, it is important for the model to learn being presented with a slice that does not contain neither a kidney nor a tumor. On the other hand, the model is presented with a lot of empty slices which significantly increases the computation time. Due to the constraint on available RAM space on Google Collaboratory, I have decided to take for every case only the slices which contain tumor and/or kidney, plus an additional five slices below and above this range. I want to emphasize that this method is not completely clean, as it makes it for the model unable to learn about the features from bottom and top slices. However, I do not expect a significant difference in model performances when using the full size along the z-axis. In order to have consistency, the test set will be transformed according to this method as well. The final train, validation and test set are of size $13533 \times 128 \times 128 \times 1$, size $1529 \times 128 \times 128 \times 1$ and size $2012 \times 128 \times 128 \times 1$ respectively.

Experiment 1(b)

There are four experiments conducted, each hoping to answer one of the respective sub-questions. Experiment one was conducted to test the U-Net and the MultiRes U-Net on the data without any pre-processing, except for slicing the 3D data and resizing them for computational efficiency. When carrying out the experiment, a first evidence was found opposing the superiority of the MultiRes U-Net, probably due to the homogeneity of the KiTS19 dataset. As a

result, another sub-experiment (experiment 1b) was conducted. In this experiment, the dataset was manually adapted using various data augmentation techniques in order to mimic a more heterogeneous dataset. The data augmentation techniques include translation, horizontal flip, rotation and elastic deformation. The details of these techniques are discussed in the setup of Experiment 4. This new training set was composed of 13000 random sampled images adapted by these augmentation techniques. The first batch of 4000 samples were translated with every 1000 samples shifted either upwards or downwards and either left or right. The second technique consisted of horizontal flipping another 1000 random samples. The third batch of 3000 samples were deformed with sigma of 3 on 4 grid points. The final 3000 samples were rotated by a degree of 30.

Experiment 2(b)

For the second experiment, the U-Net and the MultiRes U-Net were trained on the data for which the pixel values are normalized data according to the min-max normalization technique. The result of this experiment would provide evidence for whether the MultiRes U-Net is indeed superior to the classical U-Net. When carrying out this experiment, is was evident that normalizing the pixel values started to hurt the performance. Therefore, another sub-experiment (Experiment 2b) was added, in which the pixel values were clipped to range [-200, 500]. The details of this pre-processing technique were can be found in subsection *Experiment 3*.

Experiment 3

Experiment three and four aimed to address whether additional pre-processing is indeed more beneficial to the overall performance of the segmentation model, than architectural bells and whistles, as stated by the winning team of the KiTS119 challenge (Isensee & Maier-Hein, 2019). In computerized tomography scans, the value of the pixels correspond to the Hounsfield

unit values, a quantitative scale for describing radiodensity. Clipping these values to a specific range could help getting rid of non-important features, such as air (-1000), lung (-700), bone (-300) and glass (+500) (Cuong et al., 2018). Thus, in experiment three the HU values were clipped to the range of [-200, 500] using NumPy.

Experiment 4

The fourth experiment made use of different data augmentation techniques applied to the training data in order to investigate the effect of data augmentation on the performance of the classical U-Net. Image augmentation is commonly used for medical image segmentation problems as there is usually a limited amount of annotated data available (Buslaev, 2018). Furthermore, Iglovikov et al. (2018) showed that using data augmentation for medical images helps to improve the results of segmentation of hand radiographs. Common data augmentation techniques for biomedical image data are (amongst others) image translation, rotation, flipping, stretching, elastic deformation and contrast augmentation (Tol, 2019). When using translations, one shifts the region of interest of an image up or down and/or left or right. Rotating an image simply means rotating the array of an image by a particular degree size. This results in an image representation of patients scanned under a certain angle.

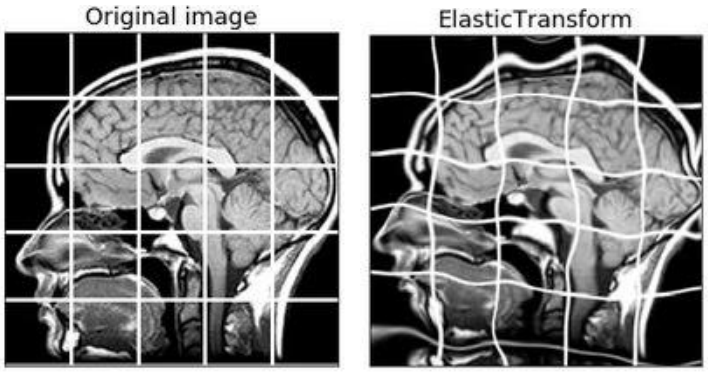
Flipping an image can be done vertically or horizontally (or both), resulting in a mirror image. This technique is especially useful in medical images where the region of interest could have been on the other side of the body, e.g. the kidney tumor on the right side instead of the left side. Training the network with more diverse data in terms of location prevents the network from favoring features present in one side of the body.

When using stretching and elastic deformations, medical images can be transformed to represent different anatomical structures or different body types. Stretching is done by enlarging

an image on either the horizontal or vertical axis. Elastic deformation is a non-linear type of data augmentation in which the image grid gets deformed along a number of grid points with the magnitude of sigma. This type of data augmentation is useful for medical data that can vary much in shape, such as cell nuclei. However, one should carefully consider the level of stretching and elastic deformations on CT scan data. A kidney with the size of a football is not very realistic and could confuse the network. Next to that, a heavily deformed CT scan image does not reflect a realistic body type.

Figure 4

Example of Elastic Transformation for Brain MRI



Note. This example shows how elastic deformation works (Tol, 2018). It is evident that using elastic deformation with a sigma that is very high can result in unrealistic medical images.

In order to keep the augmented images realistic, the augmentation is limited to rotations with a degree of 10, flipping over the horizontal axis and small elastic deformations. The elasticdeform library based on SciPy, NumPy and TensorFlow (Tulder, 2018) is used, with a sigma of 2 on a 4x4 deformation grid. This deformation resulted in images significantly different from the original image, while still realistic. The data augmentation techniques of rotating, flipping and elastic deformation were applied to a random batch of 1000 training samples. This

resulted in an increased training dataset from 13533 samples to 16533 samples. It is interesting to see the impact of both the linear augmentation techniques (i.e. flipping and rotation) and the non-linear augmentation technique (i.e. elastic deformation). Consequently, the model was firstly trained on the dataset augmented with rotations with a degree of 10 and with flipping over the horizontal axis. After this, the model was re-trained on the full augmented dataset.

These four experiments (and its sub-experiments) combined are conducted in order to answer the main research question of whether non-architectural modifications are more important than architectural modifications.

Evaluation method

The task of semantic segmentation is to predict the class of each pixel in an image. Classification accuracy is a widely used metric to measure the performance of a model, defined by dividing the correct predictions by the total number of predictions. In an image, this would be the classification accuracy of every single pixel in the image. However, this so-called pixel accuracy can lead to misleading evaluation results when having an imbalanced dataset. In the case of kidney tumor segmentation, we are confronted with such class imbalance: the kidney and kidney tumor are only a small percentage of the total image. Therefore, pixel accuracy is not a reliable evaluation metric.

Instead of pixel accuracy, the Jaccard Index (or the Intersection-Over-Union) is a widely used alternative metric, used in semantic segmentation tasks. The Jaccard Index (see Equation 1) is calculated by dividing the intersection of the ground truth (A) and the predicted output (B), by the union of the ground truth (A) and the predicted output (B).

$$J(A,B) = \frac{|A \cap B|}{A + B - |A \cup B|} \quad (1)$$

This metric presents us with more information about the actual segmentation performance of the model as it is comparing the segmentations of the output and ground truth. Another metric widely used is the Dice coefficient (2), which is very similar to the Jaccard index. As the Dice Coefficient and the Jaccard Index are related, one can calculate the other using Equation 3.

$$Dice\ Coefficient(A, B) = \frac{2 \times |A \cap B|}{|A| + |B|} \tag{2}$$

$$Jaccard\ Index = \frac{D}{2 - D} \quad \text{and} \quad D = \frac{2J}{J + 1} \tag{3}$$

The Dice coefficient is widely used as a metric to evaluate the performance of a segmentation task. Furthermore, it is used in the KiTS19 challenge, which is the source of the dataset used for the purpose of this research. Therefore, this metric will be used here, in order to evaluate the results of the U-Net and the MultiRes U-Net.

Training setup

As mentioned before, the objective of a segmentation task is binary pixel classification. Therefore, *Cross Entropy* (see Equation 4) is a very common and widely used loss function.

$$Cross\ Entropy(X, Y, \widehat{Y}) = \sum_{px \in X} -(y_{px} log(\hat{y}_{px}) + (1 - y_{px}) log(1 - \hat{y}_{px}) \tag{4}$$

Here, X is the input image, Y is the segmentation label and $\hat{Y}$ is the predicted segmentation. For every pixel *px*, the model predicts $\hat{y}_{px}$, while the true segmentation label corresponds to y_{px}.

Ronneberger et al. (2015) have successfully used the weighted version of the log loss, when the model is presented with a dataset containing unbalanced classes. This unbalanced representation of the classes is very common in biomedical image segmentation datasets, as is the case in the KiTS19 dataset. Therefore, the Cross Entropy loss function will be multiplied by a weight vector (see Equation 5) in order to account for the unbalanced classes.

$$Weight\ vector(w_1, w_0, Y) = Y \times w_1 + (1 - Y) \times w_0 \quad (5)$$

Here, (w_1, w_0) represent the weights given to the pixel value of 1 and 0 respectively. In addition to a weighted Cross Entropy, Milletari et al. (2016) proposed the Dice Coefficient as a loss function. This function, known as the *Dice Loss*, is defined by simply subtracting the Dice Coefficient from one. As a consequence, the final loss function used, is an addition of the Dice Loss to the weighted Cross Entropy in order to obtain the best results.

The model is compiled using an *Adam* optimizer (with an initial learning rate of 1e-4), as previous research has shown this adaptive learning optimizer works best for image segmentation tasks. Both models will be trained for 200 epochs, unless there is no improvement to be found in the validation loss after 30 epochs. The code for these experiments will be made available on GitHub by *Laura1295.*

Results

This research paper will use the results of the MultiRes U-Net described by the Ibtehaz et al. (2020) as a baseline. The authors describe the results of their model and the classical U-Net model for every modality dataset in terms of the Jaccard index in percentages. These results are

presented in Table 1 and are converted to the Dice Coefficient (in percentages) for convenient comparison. The last five rows (CT 1-4) contain the results on the experiments conducted for the purpose of this research. The main goal of this research is to test whether pre-processing is more important than architectural modifications, the MultiRes U-Net model is not part of experiment three and four, as these experiments contain additional pre-processing techniques. The rest of this section is presented as follow. First, it will describe some general results on the performance of both the classical U-Net and the MultiRes U-Net. Then, every experiment will be briefly mentioned and the results for both the models will be presented.

Table 1

Dice Coefficient in Percentages

Modality	Model		Difference
	MultiRes U-Net	U-Net	
Dermoscopy	89.07%	86.65%	+2.42%
Endoscopy	90.15%	85.39%	+4.76%
Fluroscence Microscopy	95.65%	94.35%	+1.30%
Electron Microscopy	93.58%	93.29%	+0.29%
MRI	87.76%	87.08%	+0.68%
CT (1)	68.17%	86.76%	-18.59%
CT (1b)	80.04%	88.47%	-8.43%
CT (2)	15.54%	82.89%	-20.35%
CT (2b)	72.83%	90.05%	-17.22%
CT (3)	-	87.96%	-
CT (4)	-	**93.46%**	-

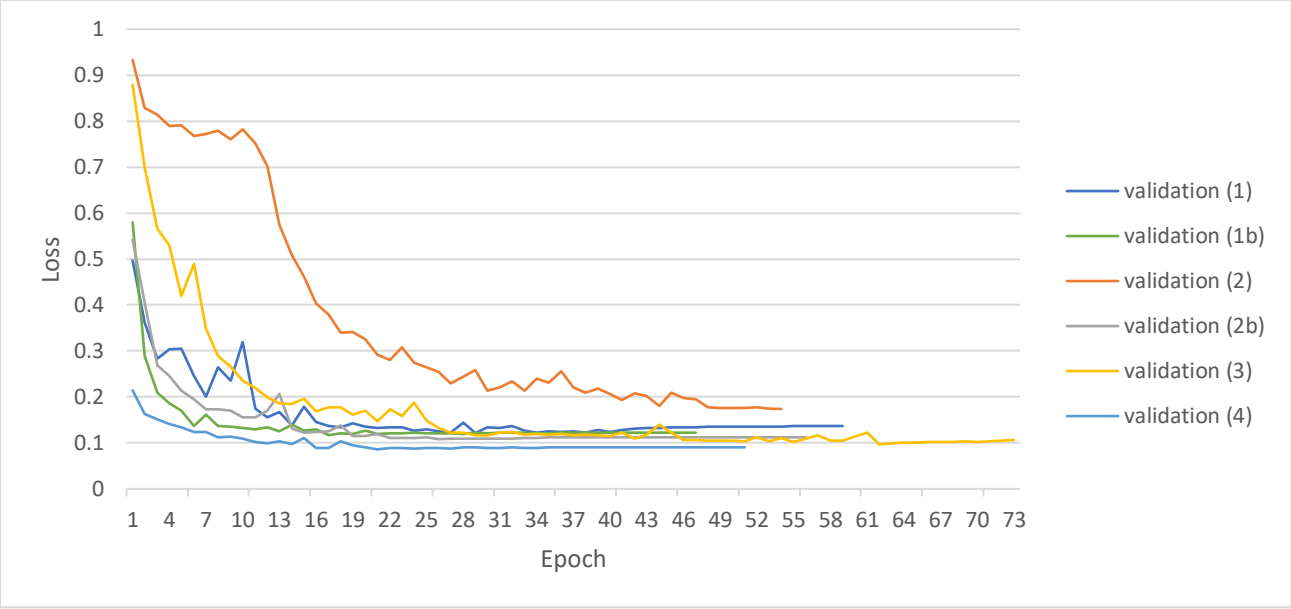
Note. CT (1-4) are the results of the experiments conducted for the purpose of this research. The experiments are performed on the KiTS19 dataset without pre-processing (1), with a transformed training set (1b), with min-max normalization (2), with clipping of the HU values (2b), with both min-max normalization and clipping (3) and with data augmentation (4).

General results

Besides to the Dice Coefficient on the test set, it is important to look at the training and validation Dice Coefficient, the training and validation loss and the time per epoch. The progress of the training and validation loss gives insight into whether the model is training more stable. In addition, it gives an indication of whether the model is overfitting or underfitting during training. Lastly, it is useful to measure these outcomes with respect to efficiency. As the main goal of automizing annotations is efficiency, it is only logic to choose a model that is precise in segmenting but also efficient in segmenting. The Dice Coefficient and loss for both the models can be found in the Appendix B, for all experiments. The time per epoch for the MultiRes U-Net lies around 1220 seconds while the U-Net only takes 210 seconds per epoch. Furthermore, it is evident from Figure 6 that it takes longer for the MultiRes U-Net to converge for all experiments. Figure 5 shows that the validation loss of the U-Net is the highest in experiment two and the lowest in experiment four.

Figure 5

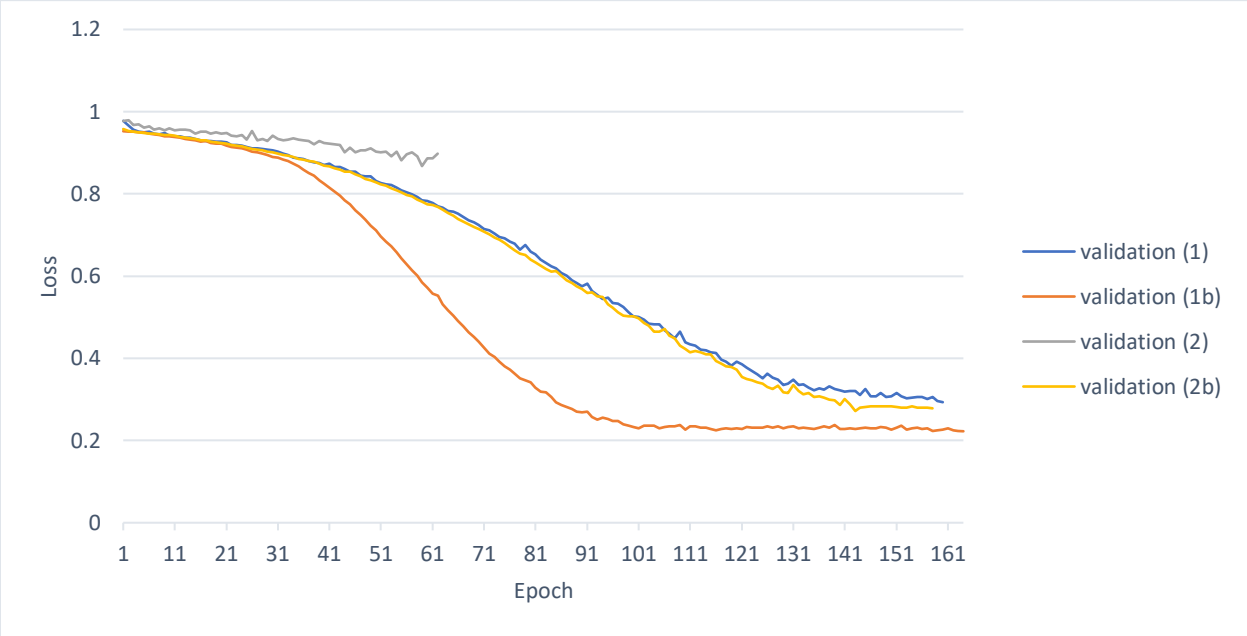
U-Net: Validation Loss of the Experiments



Note. Each number in between brackets in the legend corresponds to the number of the experiment using the KiTS19 dataset: without pre-processing (1), with a transformed training set (1b), with min-max normalization (2), with clipping of the HU values (2b), with both min-max normalization and clipping (3) and with data augmentation techniques (4).

Figure 6

MultiRes U-Net: Validation Loss of the Experiments



Note. Each number in between brackets in the legend corresponds to the number of the experiment using the KiTS19 dataset: without pre-processing (1), with a transformed training set (1b), with min-max normalization (2) and with clipping of the HU values (2b).

Results Experiment 1(b)

In experiment one, both the classical U-Net and the MultiRes U-Net were trained on the data without any pre-processing, except for slicing the 3D data and resizing them for

computational efficiency. From Figure B1 and B5 of Appendix B, it is clear that the classical U-Net converges faster to its final Dice Coefficient than the MultiRes U-Net. When we compare the development of the validation loss, it is evident that the MultiRes U-Net trains more stable and it has less fluctuations in its validation loss. As mentioned before in the *Experimental Setup* section, during experiment one another sub-experiment was conducted. Figure 5 and Figure 6 show that the effect of a more heterogeneous training set is larger for the MultiRes U-Net than for the classical U-Net.

Results Experiment 2(b)

In experiment two, both the classical U-Net and the MultiRes U-Net were trained on the data pre-processed with min-max normalizing the pixel values. When conducting the second experiment, the normalization of the input data actually started to hurt the performance of both the models, the MultiRes U-Net in particular (see Figure 6). Therefore, a sub-experiment (experiment 2b) was conducted, using the data only pre-processed by clipping the HU values to the range of [-200, 500]. This sub-experiment was not part of the initial experiment planning laid out in the *Method* section. It is evident from Table 1 that clipping the HU values is beneficial, as the Dice Coefficient is higher for both the models then in experiment one. Furthermore, Figure 5 shows that the validation loss for the U-Net is more stable and converges faster than in experiment one.

Results Experiment 3

The classical U-Net was trained on the data pre-processed with clipping the HU values, and subsequently min-max normalized. Figure 5 shows that even though it takes longer for the classical U-Net model to converge to its final validation loss, the loss is lower than in experiment

one and two. Nevertheless, the final Dice Coefficient is lower for experiment three than for experiment 2b.

Results Experiment 4

In experiment four only the classical U-Net was trained on the data augmented with several image augmentation techniques. This experiment was conducted in two stages. First the data were augmented by horizontally flipped and slightly rotated random samples Secondly, elastically deformed random samples were added to the data. For every technique, 1000 random samples of the training data were used. The experiment was conducted in these three stages as this would clearly show the effect of different data augmentation techniques, mainly the difference between linear (flip, rotate and shift) and non-linear (elastically deform) image augmentation. Adding augmented images to the training set resulted in the lowest loss, (see Table 1) and the model converges faster to its final loss in this experiment than in all other experiments. Using flipping and rotating as data augmentation techniques resulted in a higher Dice Coefficient on the test set. However, adding shifted and elastically deformed images decreased the model performance on the test set (see Table 2). Therefore, the final result on the test set of experiment four (see Table 1) is based on the model trained on the dataset augmented with only flipped and rotated images.

Table 2

Experiment 4: Dice Coefficient in Percentages per Stage

Stage	Result
Stage 1	**93.46%**
Stage 2	91.15%
Stage 3	90.57%

Note. stage 1 corresponds to the data of experiment two with an additional flipping and rotating of the data, applied to1000 samples for each technique. Stage 2 corresponds the data of stage 1 with an additional 1000 shifted samples. Stage 3 corresponds to the data of stage 2 with an additional 1000 elastically deformed samples.

Discussion

The goal of this research is to investigate whether a novel architecture build upon the U-Net is outperforming the classical U-Net in the task of kidney and kidney tumour segmentation. Furthermore, this research aimed to address whether pre-processing is more important than the model architecture, in the task of biomedical image segmentation. This section will discuss the presented results.

Based on all four experiments, we can conclude that the classical U-Net performed better on the KiTS19 dataset than the MultiRes U-Net. As mentioned before, only pre-processing the images by normalizing the pixel values decreased the performance on both models. This can be due to the fact that the ranges of HU values differ much across the samples. As a consequence, min-max normalization with varying minimum and maximum pixel values leads to difference in pixel value per sample for the same HU value. When the models were presented with an altered training set (Experiment 1b), the Dice Coefficient on the test set of the MultiRes U-Net was over 10% higher than in experiment one (see Table 1). Even though the Dice Coefficient on the test set of the classical U-Net was still higher, it was only increased by a small 2% compared to experiment one. This result supports the idea that the MultiRes U-Net can only create superior results when the dataset is very heterogeneous.

From the result of experiment three it is evident that the performance of the classical U-Net increased when the data were pre-processed by clipping the HU values. Clipping the values

decreased the number of features that need to be processed (e.g. lung and air) and made it easier for the model to focus on the important features (e.g. kidney and kidney tumour tissue). However, after normalizing the pixel values, the performance decreased.

The fourth and final experiment showed the importance of applying data augmentation techniques relevant for medical image data. As shown in Table 2, horizontal flipping of the data and applying slight rotations to the data helped the performance of the model. However, while adding elastic deformations improved the Dice Coefficient on the training set, it did not improve the results on the test set. A reason for this can be the fact that the elastic deformations were not heavily applied in order to keep the image similar to real CT scan images. Possibly, the images were too similar to the original images. As the training Dice Coefficient was almost 4% higher than the validation Dice Coefficient, there were signs of overfitting due to the (almost) duplicate images in the training set. In addition, translating the images by shifting did not contribute to a higher Dice Coefficient either. It could be that the model needed to be trained longer (i.e. with more samples) on shifted data, as this augmentation technique significantly alters the location of the target and needs to be learned by the model.

In contrast to the results obtained by Ibtehaz et al. (2020), the MultiRes U-Net is not outperforming the classical U-Net. First of all, it is much slower due to the fact that the MultiRes U-Net is making use of the 'Res path' in which a feature is passed through a chain of convolutional layers with residual connections. Furthermore, the performance of the MultiRes U-Net on both the min-max normalized data and the HU values clipped data (Experiment 2 and 2b respectively) were much lower than the performance of the U-Net. An explanation for this can the fact that the MultiRes U-Net performs better when the data suffers from noises, perturbations and lack of clear boundaries (Ibtehaz et al., 2020). In the KiTS19 dataset, these noises were less,

or even not at all, present than in the datasets used by Ibtehaz et al. (2020). In addition, the size of the object of interest did not vary much per case. Consequently, the positive impact of the MultiRes blocks was small. A limitation of this research is therefore the lack of variety in the used dataset as the objects of interest (kidney and kidney tumour) are similar in size, location and structure. From experiment 1b, it is evident that increasing the heterogeneity of the training set, leads to a larger increase in performance by the MultiRes U-Net model than by the U-Net model. Therefore, future research could conduct similar experiments using a more heterogeneous dataset. Another limitation to this research could be the fact that the CT scan slices for which the segmentation label was empty, were removed from the training, validation and test set. This method was carefully considered as it would benefit the computational efficiency. However, it also removed some additional variety in the dataset. When one has access to a GPU with more computational power, these slices can be kept in the final dataset.

In recent years, there have been many publications of variations of the famous U-Net architecture. Most of them are outperforming the classical U-Net on a particular task of biomedical image segmentation. Nonetheless, there is a need to test these novel models on their generalizability across different datasets. Ibtehaz et al. (2020) requested such an evaluation of their MultiRes U-Net architecture, namely, testing the model performance on medical images originating from another modality. This research paper provided the basis for the evaluation of the MultiRes U-Net by testing it on CT scan data for kidney and kidney tumour segmentation.

Conclusion

This research paper aimed to address the following question: “To what extent does the use of non-architectural modifications have more significant impact on the semantic

segmentation of kidney tumors, than the architectural modification of adding Res paths and MultiRes blocks to the U-Net?". The different experiments performed on the KiTS19 dataset showed that the Multi-Res U-Net does not outperform the classical U-Net. This can be due to a misalignment between the nature of the dataset and the nature of the datasets on which the model has previously shown to perform significantly better. Furthermore, the U-Net performs increasingly better as more pre-processing techniques are used. From the results of the experiments, one can cautiously state that the classical U-Net is performing better on more homogeneous datasets, and the MultiRes U-Net is performing better on heterogeneous datasets. Directions of future research could evaluate the MultiRes U-Net on more heterogeneous CT scan datasets in order to evaluate whether this statement holds.

Acknowledgements

This research is the final step toward obtaining my master Data Science and Society at the University of Tilburg. I have encountered many obstacles during this journey and would therefore like to thank my thesis supervisor Sebastian Olier for guiding me through these. Furthermore, I would like to thank my friend Erik-Jan Meulenbrugge for contributing with his knowledge on modelling deep neural networks. Finally, I would like to thank my friends and family for their unconditional support.

References

Adegun, A., & Viriri, S. (2019, August). Deep Learning Model for Skin Lesion Segmentation: Fully Convolutional Network. In *International Conference on Image Analysis and Recognition*(pp. 232-242). Springer, Cham.

Alom, M. Z., Hasan, M., Yakopcic, C., Taha, T. M., & Asari, V. K. (2018). Recurrent residual convolutional neural network based on u-net (r2u-net) for medical image segmentation. *arXiv preprint arXiv:1802.06955*.

Buslaev, A., Iglovikov, V. I., Khvedchenya, E., Parinov, A., Druzhinin, M., & Kalinin, A. A. (2018). Albumentations: fast and flexible image augmentations. *Information*, *11*(2), 125.

Çiçek, Ö., Abdulkadir, A., Lienkamp, S. S., Brox, T., & Ronneberger, O. (2016, October). 3D U-Net: learning dense volumetric segmentation from sparse annotation. In *International conference on medical image computing and computer-assisted intervention* (pp. 424-432). Springer, Cham.

Ciresan, D., Giusti, A., Gambardella, L. M., & Schmidhuber, J. (2012). Deep neural networks segment neuronal membranes in electron microscopy images. In *Advances in neural information processing systems* (pp. 2843-2851).

Cuong, N. L. Q., Minh, N. H., Cuong, H. M., Quoc, P. N., Van Anh, N. H., & Van Hieu, N. (2018). Porosity Estimation from High Resolution CT SCAN Images of Rock Samples by Using Housfield Unit. *Open Journal of Geology*, *8*(10), 1019.

Drozdzal, M., Vorontsov, E., Chartrand, G., Kadoury, S., & Pal, C. (2016). The importance of skip connections in biomedical image segmentation. In *Deep Learning and Data Labeling for Medical Applications* (pp. 179-187). Springer, Cham.

Havaei, M., Davy, A., Warde-Farley, D., Biard, A., Courville, A., Bengio, Y., ... & Larochelle, H. (2017). Brain tumor segmentation with deep neural networks. *Medical image analysis*, *35*, 18-31.

Heller, N., Isensee, F., Maier-Hein, K. H., Hou, X., Xie, C., Li, F., ... & Yao, G. (2019). The state of the art in kidney and kidney tumor segmentation in contrast-enhanced CT imaging: Results of the KiTS19 Challenge. *arXiv preprint arXiv:1912.01054*.

Ibtehaz, N., & Rahman, M. S. (2020). MultiResUNet: Rethinking the U-Net architecture for multimodal biomedical image segmentation. *NeuralNetworks*, *121*, 74-87.

Iglovikov, V. I., Rakhlin, A., Kalinin, A. A., & Shvets, A. A. (2018). Paediatric bone age assessment using deep convolutional neural networks. In *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support* (pp. 300-308). Springer, Cham.

Isensee, F., & Maier-Hein, K. H. (2019). An attempt at beating the 3D U-Net. *arXiv preprint arXiv:1908.02182*.

Isensee, F., Petersen, J., Klein, A., Zimmerer, D., Jaeger, P. F., Kohl, S., ... & Maier-Hein, K. H. (2018). nnu-net: Self-adapting framework for u-net-based medical image segmentation. *arXiv preprint arXiv:1809.10486*.

Kayalibay, B., Jensen, G., & van der Smagt, P. (2017). CNN-based segmentation of medical imaging data. *arXiv preprint arXiv:1701.03056*.

Li, X., Chen, H., Qi, X., Dou, Q., Fu, C. W., & Heng, P. A. (2018). H-DenseUNet: hybrid densely connected UNet for liver and tumor segmentation from CT volumes. *IEEE transactions on medical imaging*, *37*(12), 2663-2674.

Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. *In Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3431-3440).

Meinzer, H. P., Thorn, M., Vetter, M., Hassenpflug, P., Hastenteufel, M., & Wolf, I. (2002). Medical imaging: examples of clinical applications. *ISPRS journal of photogrammetry and remote sensing*, *56*(5-6), 311-325.

Milletari, F., Navab, N., & Ahmadi, S. A. (2016, October). V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *2016 Fourth International Conference on 3D Vision (3DV)* (pp. 565-571). IEEE.

Oktay, O., Schlemper, J., Folgoc, L. L., Lee, M., Heinrich, M., Misawa, K., ... & Glocker, B. (2018). Attention u-net: Learning where to look for the pancreas. *arXiv preprint arXiv:1804.03999*.

Orlando, J. I., Prokofyeva, E., del Fresno, M., & Blaschko, M. B. (2018). An ensemble deep learning based approach for red lesion detection in fundus images. *Computer methods and programs in biomedicine*, *153*, 115-127.

Ronneberger, O., Fischer, P., & Brox, T. (2015, October). U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention* (pp. 234-241). Springer, Cham.

Sancinetti, F. (2018, August 29). U-NET ConvNet for CT-Scan segmentation. Retrieved May 20, 2018, from https://medium.com/@fabio.sancinetti/u-net-convnet-for-ct-scan-segmentation-6cc0d465eed3

Tol, J. (2019, June 28). Image Augmentation: How to overcome small radiology datasets? Retrieved June 16, 2020, from https://www.quantib.com/blog/image-augmentation-how-to-overcome-small-radiology-datasets

Tulder, G. V. (2018). Elastic deformations for N-dimensional images (Python, SciPy, NumPy, TensorFlow). Retrieved June 18, 2020, from https://pypi.org/project/elasticdeform/

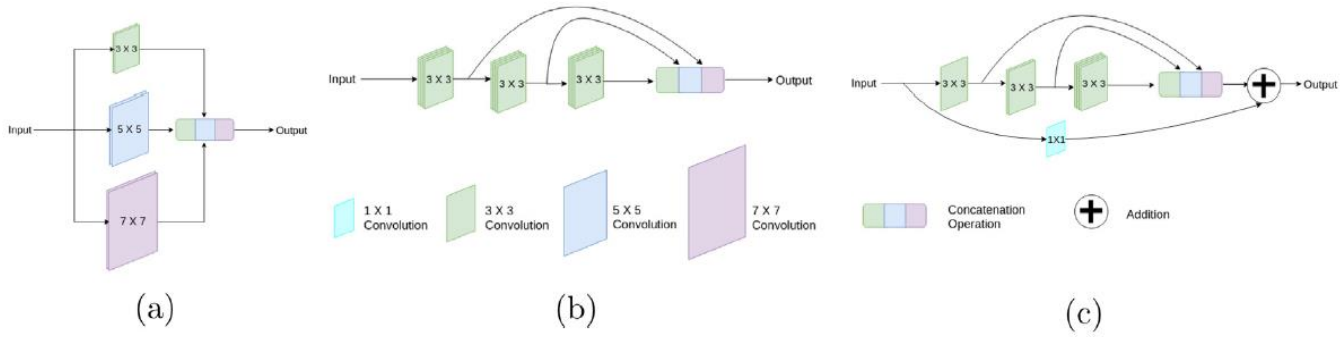
Vu, M. H., Grimbergen, G., Nyholm, T., & Löfstedt, T. (2019). Evaluation of Multi-Slice Inputs to Convolutional Neural Networks for Medical Image Segmentation. *arXiv preprint arXiv:1912.09287*.

Zhou, Z., Siddiquee, M. M. R., Tajbakhsh, N., & Liang, J. (2019). UNet++: Redesigning Skip Connections to Exploit Multiscale Features in Image Segmentation. *IEEE Transactions on Medical Imaging*.

Appendix A

Figure A1

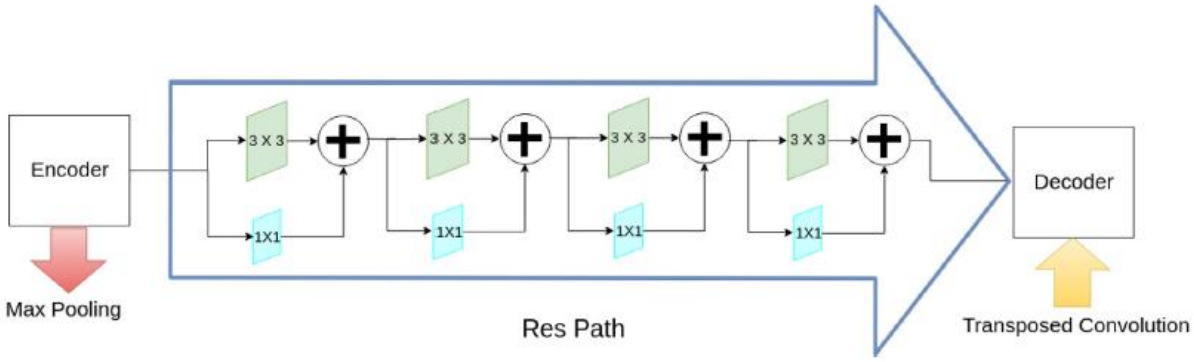
Composition of the MultiRes Block



Note. This figure contains the composition details of the MultiRes Block described by Ibtehaz et al. (2020).

Figure A2

Composition of the MultiRes Path



Note. This figure contains the details of the Res path described by Ibtehaz et al. (2020).

Figure A3

Summary of the Layers of the MultiRes U-Net

MultiResUNet					
Block	Layer (filter size)	#filters	Path	Layer (filter size)	#filters
MultiRes Block 1	Conv2D(3,3)	8	Res Path 1	Conv2D(3,3)	32
	Conv2D(3,3)	17		Conv2D(1,1)	32
MultiRes Block 9	Conv2D(3,3)	26		Conv2D(3,3)	32
	Conv2D(1,1)	51		Conv2D(1,1)	32
MultiRes Block 2	Conv2D(3,3)	17		Conv2D(3,3)	32
	Conv2D(3,3)	35		Conv2D(1,1)	32
MultiRes Block 8	Conv2D(3,3)	53		Conv2D(3,3)	32
	Conv2D(1,1)	105		Conv2D(1,1)	32
MultiRes Block 3	Conv2D(3,3)	35	Res Path 2	Conv2D(3,3)	64
	Conv2D(3,3)	71		Conv2D(1,1)	64
MultiRes Block 7	Conv2D(3,3)	106		Conv2D(3,3)	64
	Conv2D(1,1)	212		Conv2D(1,1)	64
MultiRes Block 4	Conv2D(3,3)	71		Conv2D(3,3)	64
	Conv2D(3,3)	142		Conv2D(1,1)	64
MultiRes Block 6	Conv2D(3,3)	213	ResPath 3	Conv2D(3,3)	128
	Conv2D(1,1)	426		Conv2D(1,1)	128
MultiRes Block 5	Conv2D(3,3)	142		Conv2D(3,3)	128
	Conv2D(3,3)	284		Conv2D(1,1)	128
	Conv2D(3,3)	427	Res Path 4	Conv2D(3,3)	256
	Conv2D(1,1)	853		Conv2D(1,1)	256

Note. This figure contains the details of the MultiRes U-Net architecture as described by Ibtehaz et al. (2020).

Appendix B

Figure B1

Loss MultiRes U-Net - Experiment 1

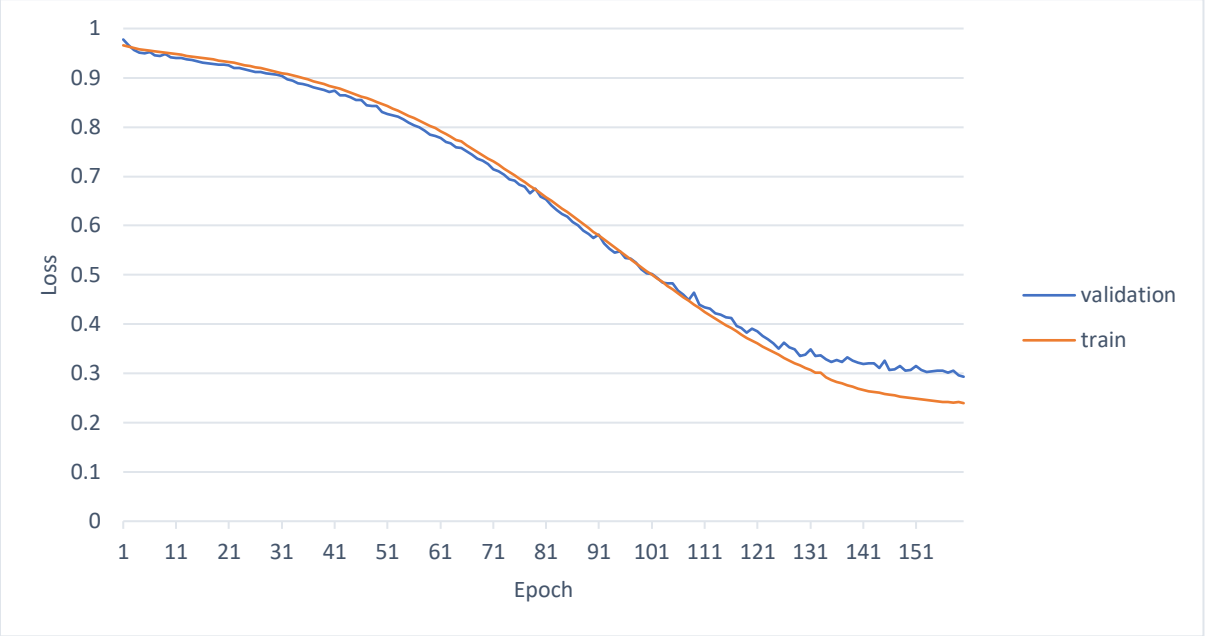


Figure B2

Dice Coefficient MultiRes U-Net - Experiment 1

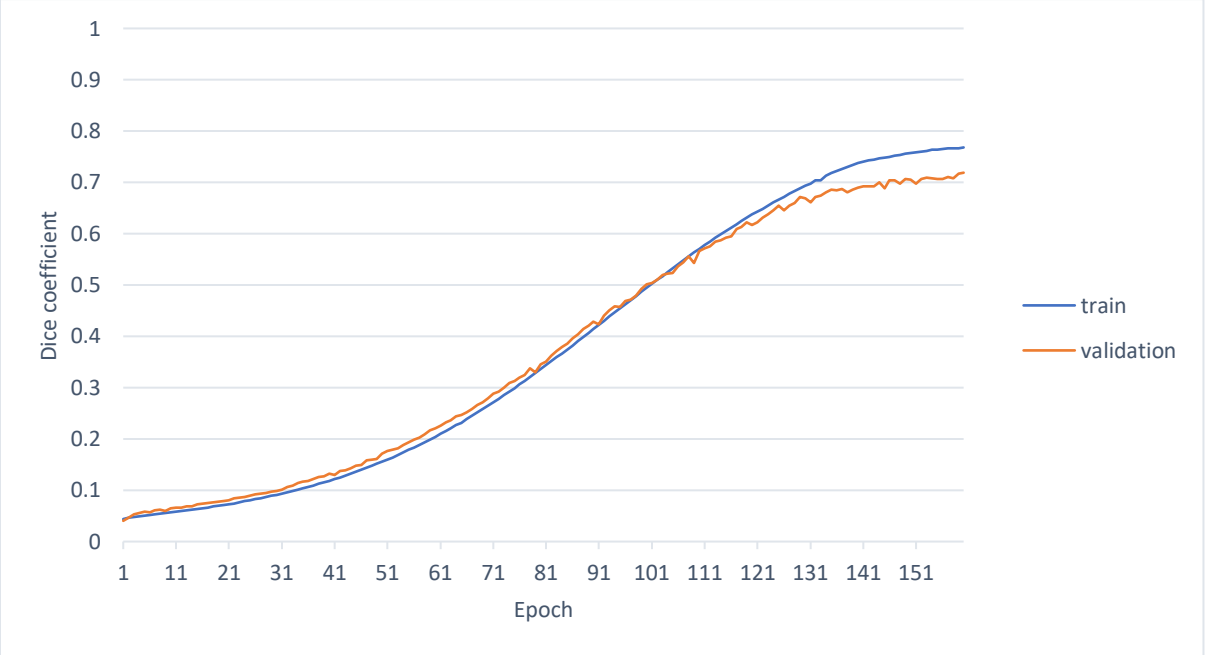


Figure B3

Loss MultiRes U-Net - Experiment 2b

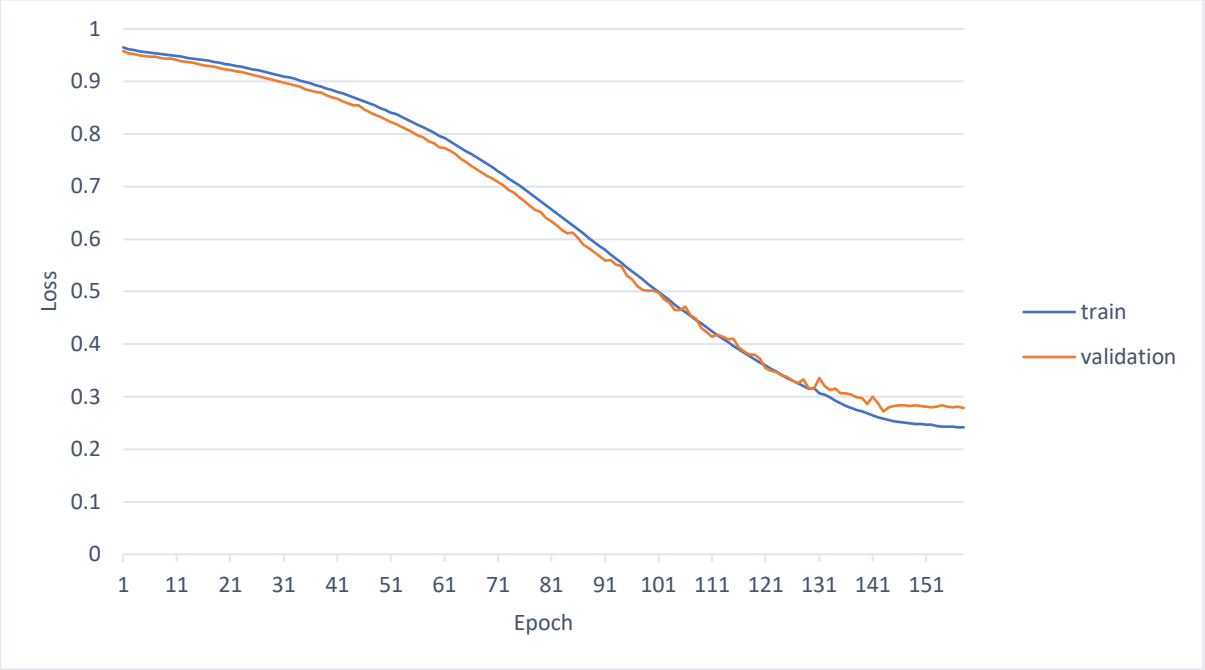


Figure B4

Dice Coefficient MultiRes U-Net - Experiment 2b

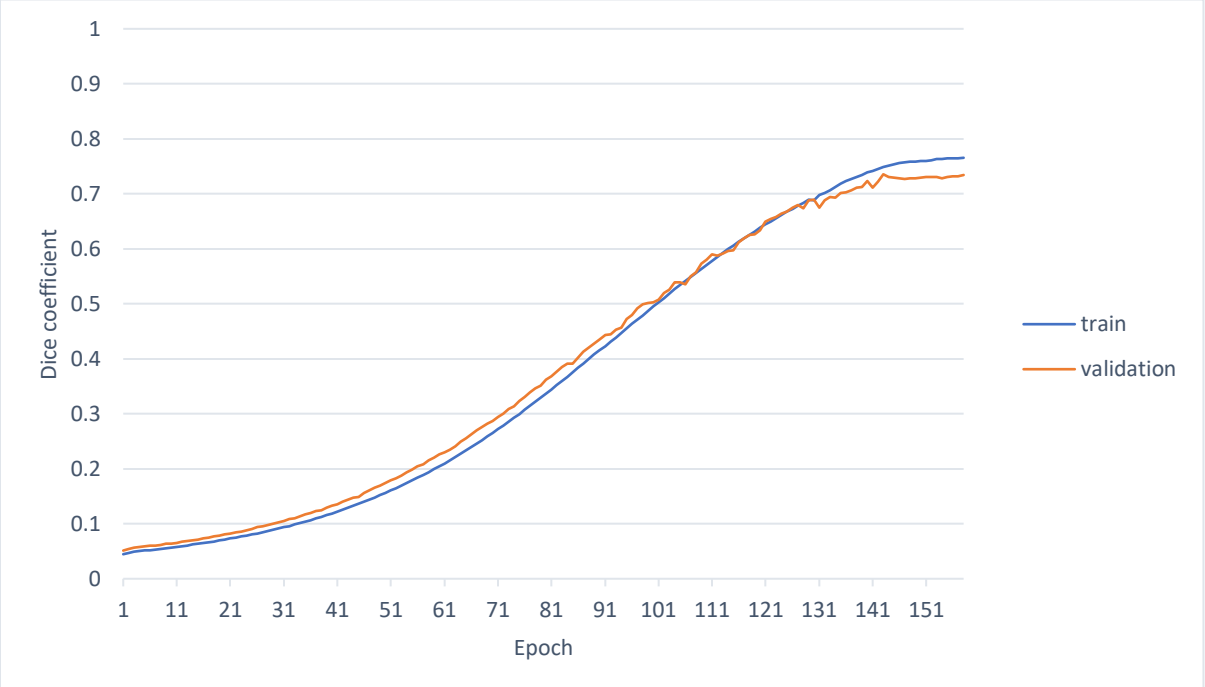


Figure B5

Loss U-Net - Experiment 1

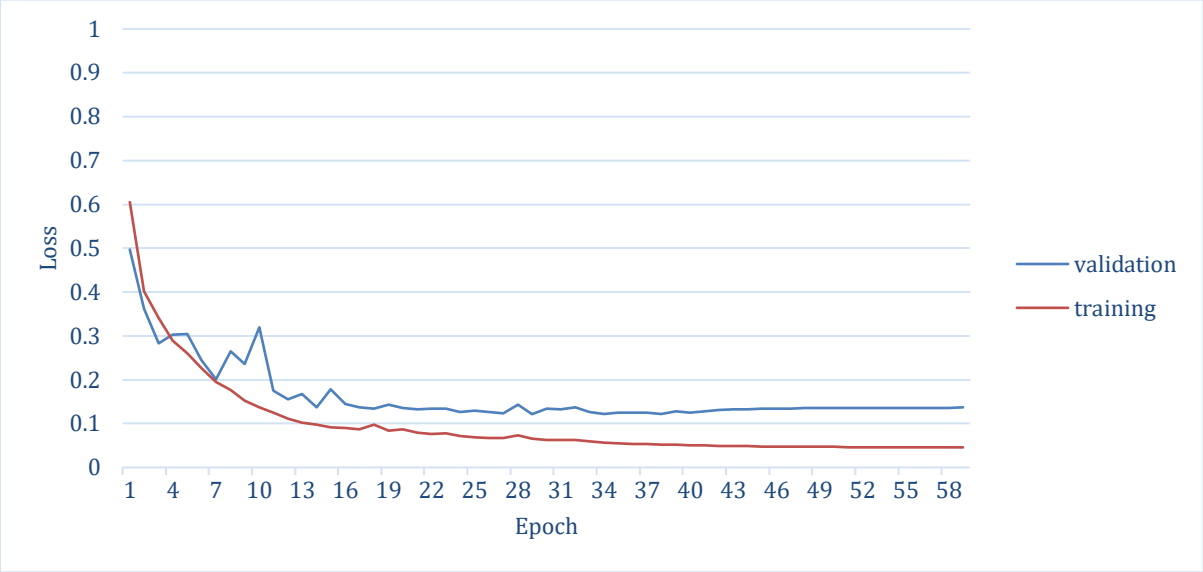


Figure B6

Dice Coefficient U-Net - Experiment 1

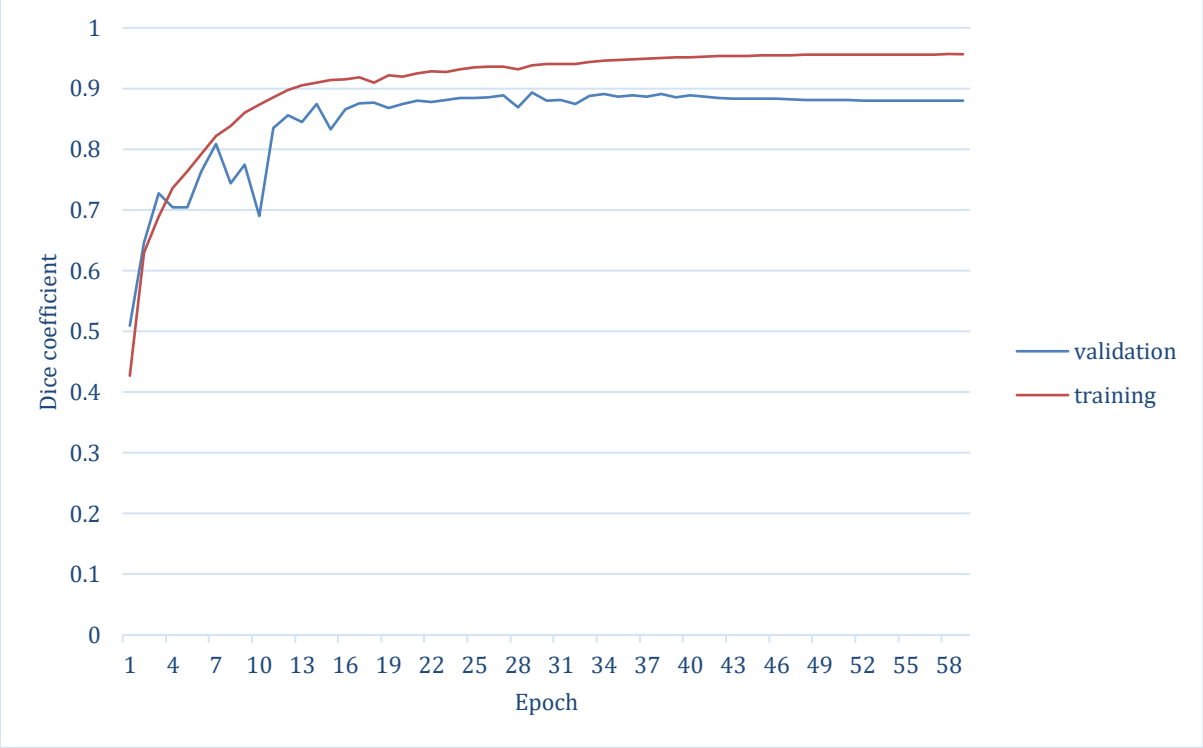


Figure B5

Loss U-Net - Experiment 2

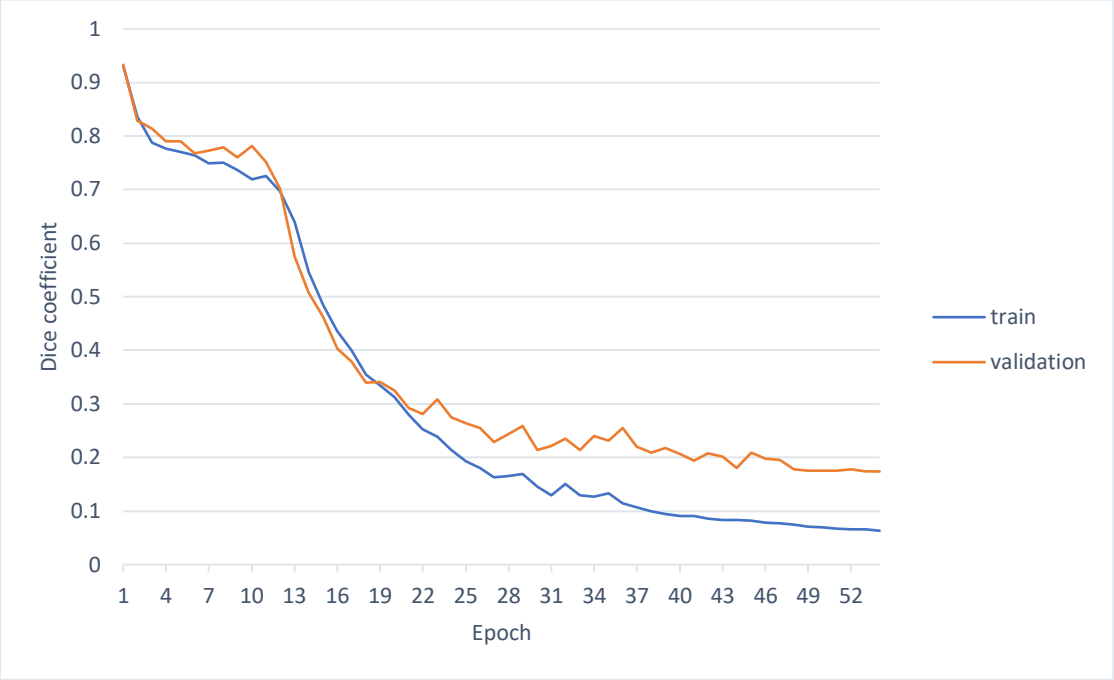


Figure B7

Dice Coefficient U-Net - Experiment 2

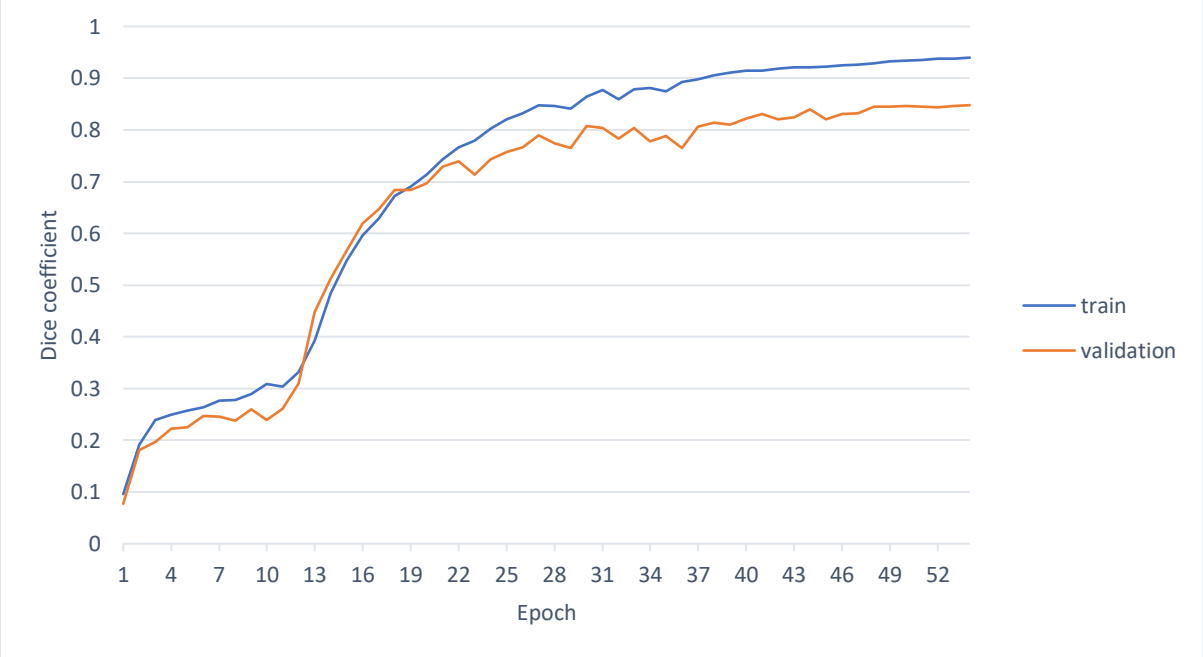


Figure B8

Loss U-Net - Experiment 2B

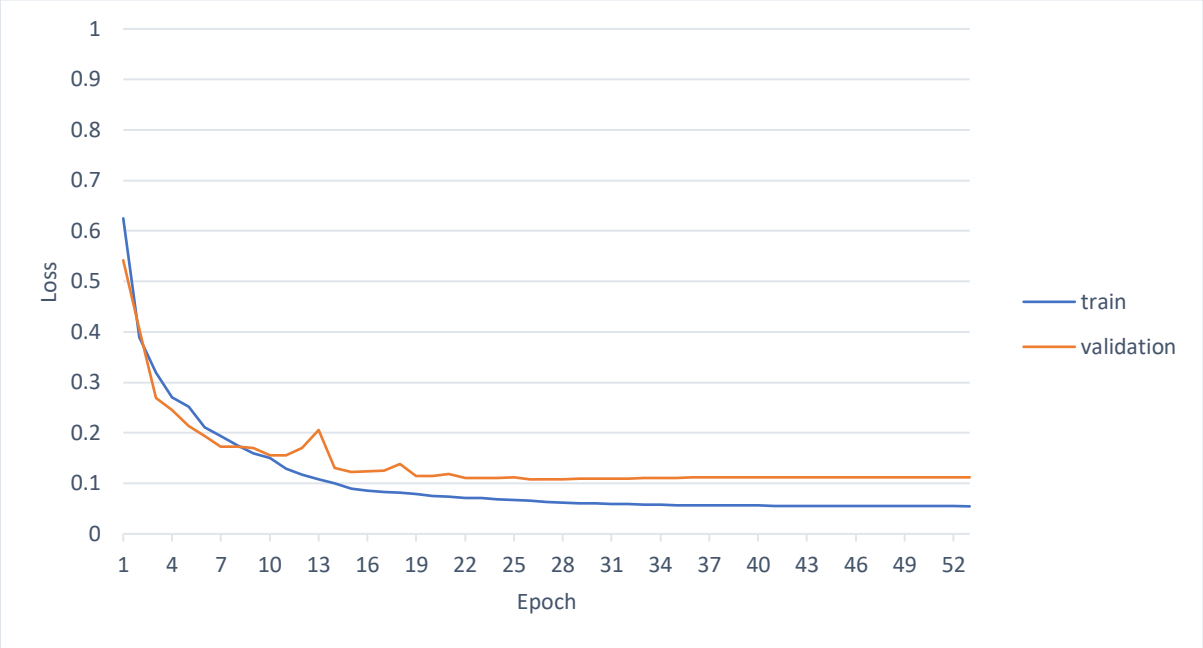


Figure B9

Dice Coefficient U-Net - Experiment 2B

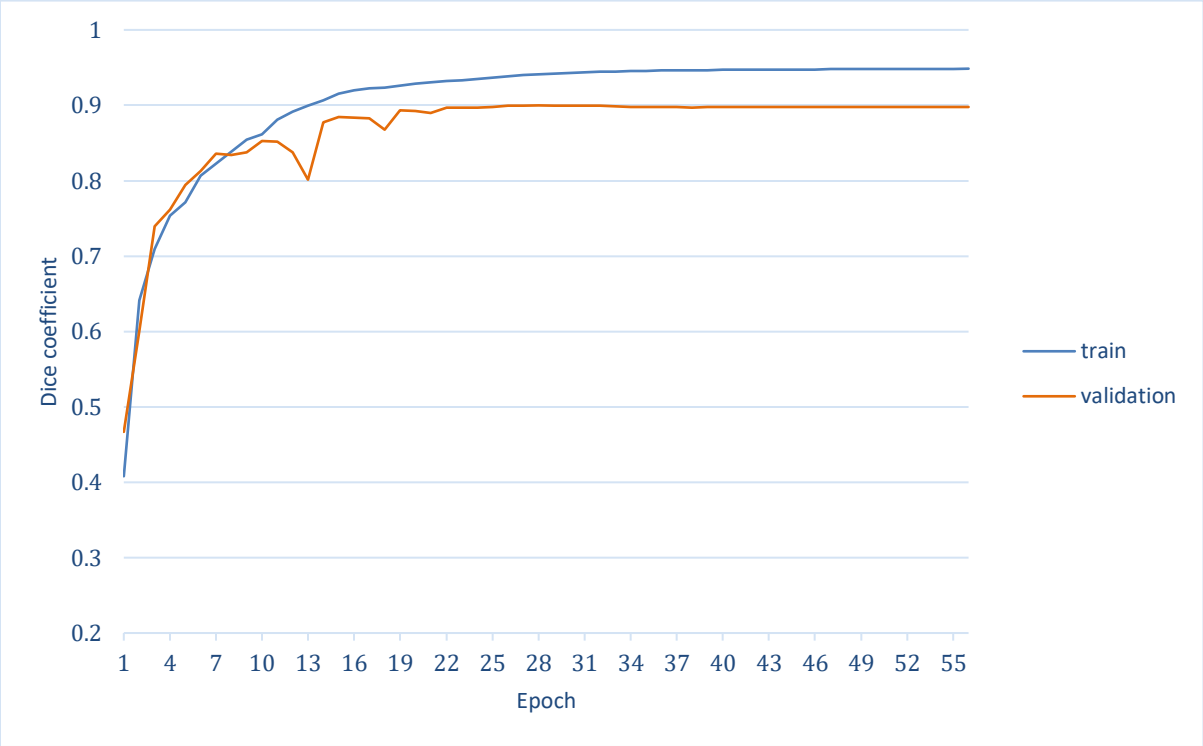


Figure B10

Loss U-Net - Experiment 3

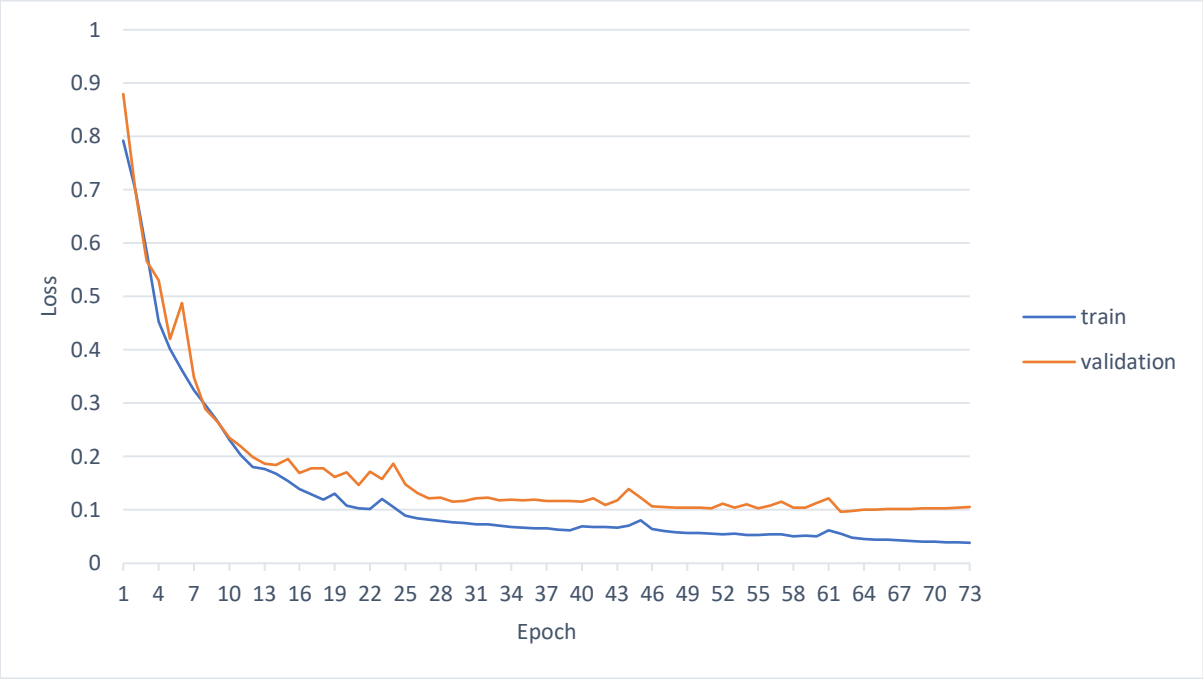


Figure B11

Dice Coefficient U-Net - Experiment 3

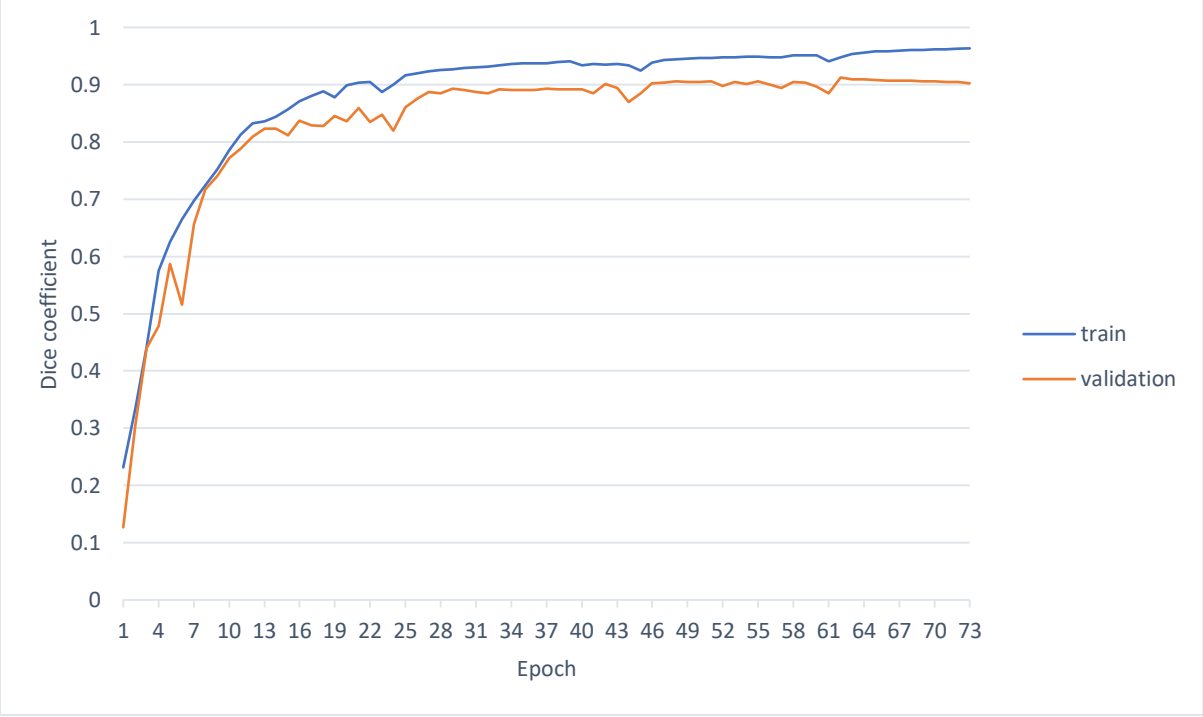


Figure B12

Loss U-Net - Experiment 4

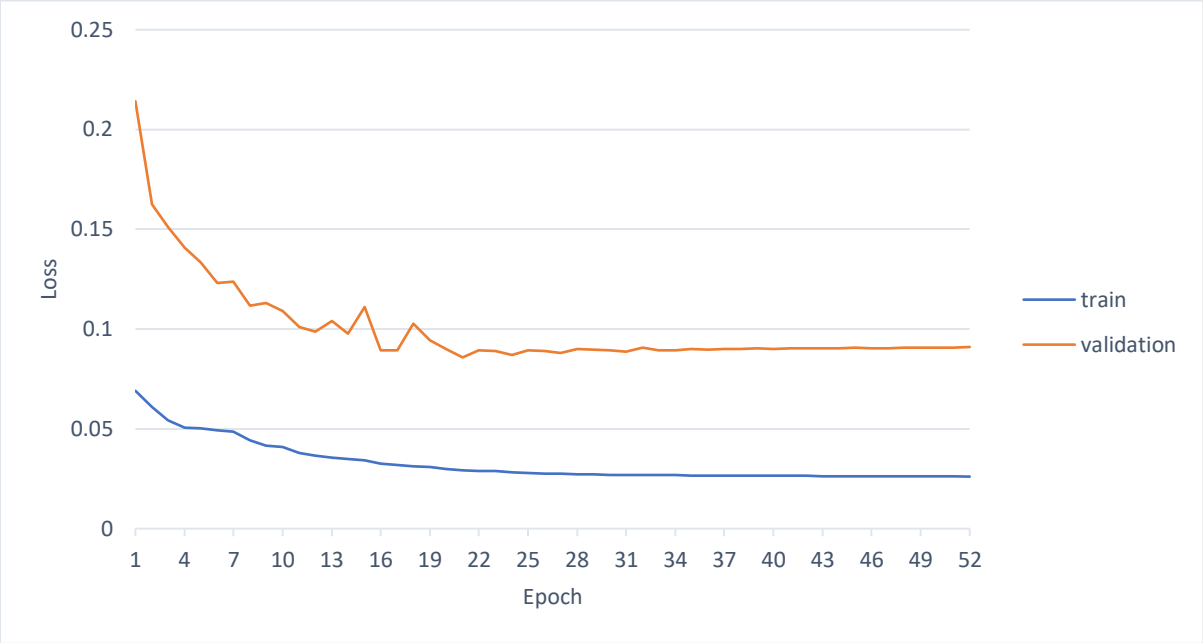


Figure B13

Dice Coefficient U-Net - Experiment 4