DEEP LEARNING FOR AUTOMATIC SEGMENTATION OF PANCREATIC TUMORS IN DIFFUSION-WEIGHTED MR IMAGES

Location, date:	Amsterdam, July 14th, 2022	
Institution:	Amsterdam UMC - Radiology and Nuclear Medicine	
Supervisors:	A. van Schelt	a.vanschelt@amsterdamumc.nl
	N.P.M. Wassenaar	n.p.wassenaar@amsterdamumc.nl
Examiner:	O.J. Gurney-Champion	o.j.gurney-champion@amsterdamumc.nl
2nd reviewer:	M.W.A. Caan	m.w.a.caan@amsterdamumc.nl
Author:	T.L. Doekemeijer	t.l.doekemeijer@amsterdamumc.nl
Student ID:	2714244	
Programme:	Master's degree Biomedical Technology and Physics (BMTP)	
	Faculty of Science, Vrije Universiteit Amsterdam	
Course:	Major Research Project BMTP (CSE)	
Course code:	XM-0065 (30ECTS)	

Abstract/Summary

Pancreatic cancer has an exceptionally high mortality rate, making it one of the most common causes of cancer mortality in devseloped countries. By quantitatively studying the microenvironment of these pancreatic tumors, parameters can be extracted, which can help with diagnosis. An important diagnostic imaging modality to evaluate these tumors is Intravoxel Incoherent Motion (IVIM) MRI, an extension of diffusion-weighted imaging (DWI). Retrieving the quantitative parameters from IVIM MR images necessitates contouring regions of interest (ROIs), a process that is time-consuming, labor-intensive, and prone to observer variation. The aim of this project was to research the use of CNNs, specifically U-Net, to automatically and accurately contour pancreatic tumors on IVIM MR images. The used dataset was retrieved from three separate studies, and consisted of 61 patients, with a total of 119 IVIM MRI volumes. Each volume consisted of 18 axial slices, from which a total of 767 slices were labeled. Leave one out cross-validation (LOOCV) was used to most efficiently split the dataset into training, validation, and testing portions. Multiple loss functions were explored, and hyperparameters of the model were optimized using an automatic software framework for ML, monitoring the mean Dice validation loss. Preprocessing was used to make the dimensions of volumes of equal size and enhance contrast, and data augmentation (rotating, shifting, and shearing) was used to make the model more generalizable. First, experiments were conducted with only foreground slices, after which varying amounts of background slices were added to avoid bias towards predictions on every slice. Additionally, experiments with cropping were applied to counter class imbalance. Model performance was evaluated on a never-before seen part of the dataset, using the Dice Similarity Coefficient (DSC) metric. Algorithms trained and evaluated on foreground slices yield the highest mean DSC with 0.39 (+- 0.22) for zero-padded input images, and 0.40 (+- 0.22) for cropped images, comparable to the mean DSC of anatomical scans in literature, but showing great variance in DSCs between patients. The algorithms trained, validated, and evaluated on input data with multiple added background slices, performed significantly worse than algorithms trained, validated, and evaluated on only foreground slices. In conclusion, U-net is not able to accurately predict pancreatic tumors on IVIM MR volumes from this dataset.

Keywords: Pancreatic tumor, Deep Learning, Convolutional Neural Networks, U-net, Intravoxel Incoherent Motion, Semantic image segmentation, Dice Similarity Coefficient, Hyperparameter optimization, Slice filtering, K-fold cross-validation.

Word count: 367

Contents

Table of figures

Glossary

A list of the most used abbreviations and acronyms is given below.

ADC	Apparent Diffusion Coefficient.
BCE	Binary Cross-Entropy.
CNN	Convolutional Neural Network.
DSC	Dice Similarity Coefficient.
DL	Deep Learning.
DWI	Diffusion Weighted Imaging.
FL	Focal Loss.
IVIM	Intravoxel Incoherent Motion.
NIfTI	Neuroimaging Informatics Technology Initiative.
ROI	Region of Interest.

1. Introduction

1.1 Pancreatic cancer

Pancreatic cancer, despite its low incidence, has an exceptionally high mortality rate, making it one of the most common causes of cancer mortality in developed countries. The most common tumor type of pancreatic cancer is the adenocarcinoma, accounting for approximately 90% of all cases [1, 2]. It is associated with a very poor prognosis, highlighted by the close parallel between disease incidence and mortality. Known risk factors that have been identified include, among others, smoking, obesity, and genetics [3]. Compared to other common cancers, the current 5-year relative survival rate (for all stages combined) is among the lowest for pancreatic cancer, at only 11% [4].

One of the most important factors of the low survival rate is the late stage at which patients are diagnosed. Most patients with pancreatic cancer are asymptomatic or have nonspecific symptoms, until the disease develops to an advanced, metastatic stage [3]. The only treatment leading to long-term survival is a combined procedure of surgical resection and systemic chemotherapy. However, only 10-20% of patients are eligible for such treatment [5]. Hence, early detection and follow-up is of great importance to provide proper treatment and control the disease progression.

1.2 Diagnosis and medical imaging

Classification and characterization of pancreatic lesions can be done through medical imaging or biopsy. Fine needle aspiration (FNA) provides a tissue diagnosis and is often considered the gold standard. However, it can be nondiagnostic due to sampling error, as the tumors tend to be unifocal, and more predominant in the head of the pancreas [6]. Therefore, imaging-based early diagnosis and image-guided treatment response monitoring are emerging potential solutions [7].

Endoscopic ultrasound (EUS) can guide this tissue acquisition, with the use of an echoscope transducer in the stomach [1]. A more widely available imaging modality for imaging patients with pancreatic adenocarcinoma is multi-detector computed tomography (MDCT). It is the best-validated modality in initial diagnosis of a suspected tumor and is additionally able to detect vascular invasion or metastases [1, 8]. Magnetic Resonance Imaging (MRI) is equally sensitive and specific in diagnosing and staging pancreatic cancer as CT [9].

Diffusion-weighted imaging (DWI), a form of MR imaging, has led to promising data towards the introduction of this sequence in routine MRI protocols for several tumors [10]. It provides functional information regarding the free diffusivity of water molecules in and around cells, which largely depends on cell type and density. As each tissue has unique diffusion properties, a distinction can be made between healthy, benign, and malignant tissue.

A promising model for DWI is called Intravoxel Incoherent Motion (IVIM), which refers to "translational movements within a given voxel and during the measurement time, present a distribution of speeds in orientation and/or amplitude" [11]. This model, originally proposed by Le Bihan et al., is an extension of DWI and integrates both the apparent diffusion coefficient (ADC) and perfusion effects. Microscopic perfusion refers to the random collective motion of blood, flowing from one capillary segment to another. This collective motion of randomness is called pseudo-diffusion. IVIM MRI is an important diagnostic imaging modality to evaluate neo-angiogenesis or microvasculature heterogeneity, and shows great potential in the detection and characterization of cancers, as well as for monitoring the response to therapy [12].

The IVIM model can be described using the following bi-exponential model [13]:

$$\frac{S}{S0} = fe^{-b(D+D^{*})} + (1-f)e^{-bD}. \quad (1)$$

In this model, MR signal intensity (S) relative to baseline (So), can be expressed in the following parameters: f is the perfusion fraction, b is a factor reflecting the strength and duration of the pulsed diffusion gradients, D is the apparent diffusion coefficient, and D* is the pseudodiffusion coefficient.

Data analysis requires high-quality data acquisition using multiple b-values, typically ranging between 0 and 1000 s/mm^2, and confidence in the measurements at low b-values [14]. The higher the b-value, the stronger the diffusion effects, and the lower the signal intensity. The optimal combination of b-values depends upon field strength, number of signals averaged, anatomical features, and predicted pathology [15]. By acquiring DWI images at multiple b-values and fitting the data to the above equation, it is possible to estimate D, f, and D* and create parametric maps for each, as can be seen in Figure 1.

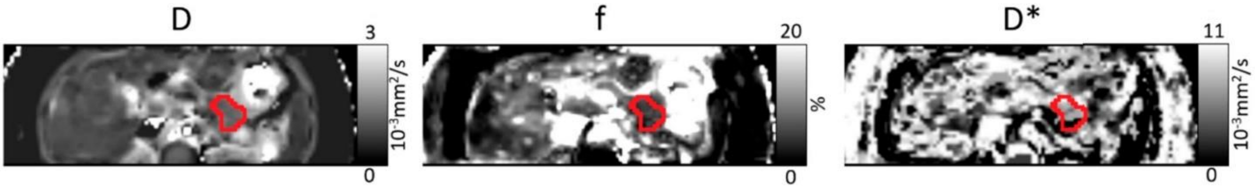


Figure 1. Parameter maps of IVIM-DWI MRI scans. The primary tumor is manually delineated in red. Maps are shown of D, the diffusion coefficient, f, the perfusion fraction, and D the pseudo-diffusion coefficient. Adapted from [16].*

IVIM-based perfusion MRI does not require contrast agents and has gained momentum recently in the field of oncology due to this advantage [11]. It can provide information on tissue microcirculation and blood flow in addition to information on tissue microstructure from diffusion MRI using the same MRI sequence. Although the IVIM model bears much promise as a tool to both visualize and characterize tumors, one major challenge is the relatively noisy parameter maps.

1.3 Artificial Intelligence for medical image segmentation

By quantitatively studying the microenvironment of these pancreatic tumors, parameters can be extracted, which can help with diagnosis. Retrieving such quantitative parameters from MR images necessitates contouring regions of interest (ROIs) from which quantitative values can be assessed. However, contouring ROIs is a time-consuming task for clinicians and there is a considerable interobserver variation in delineation of pancreatic tumors [17].

For automated contouring of ROIs, artificial intelligence (AI) can be used to reduce the workload of clinicians and increase contour consistency. In recent years, certain deep learning techniques have shown promise in solving multiple medical image tasks, with an accuracy close to human performance [18, 19].

1.3.1 Machine and Deep Learning

A subset of AI is Machine Learning (ML), in which a machine or algorithm can learn to recognize patterns and can apply this knowledge to be able to make predictions about new data [20]. Different types of machine learning algorithms can be utilized, depending on the availability of information as input. If the used input is labeled or classified, it is called supervised machine learning, where the algorithm can then correctly compare the output with the intended output [21]. The labeling is often done by a domain expert, i.e., someone who can correctly identify the labels for the input. In the field of oncology, this is done by a clinician.

Deep Learning (DL) is a branch within ML based on Artificial Neural Networks (ANNs), which are inspired by the neural network of the human brain. It consists of layers, each containing many artificial neurons/nodes, see Figure 2. These nodes receive input from nodes from the previous layer, process it using an activation function, and return a weighted sum of outputs for the nodes in the next layer. The links between the nodes are weighted, which means the value of a given node is altered using the weight of the connection (decreased or increased) before it reaches the next node. As the model trains and learns, these weights are adjusted to get the optimal output, i.e., the right prediction according to the label in supervised models. Once trained on many examples, the network is able to make new predictions on unseen data [20].

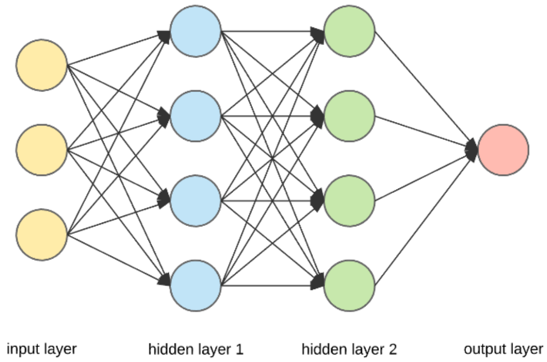


Figure 2. A schematic diagram of the architecture of a neural network, consisting of an input layer, two hidden layers, and an output layer. The nodes, represented as circles, are linked with nodes from the previous and next layer. Each connection has a weight associated with it. Source: TowardsDataScience.

1.3.2 Convolutional Neural Networks

A subtype of a DL model is the Convolutional Neural Network (CNN), most commonly used in image analysis. A CNN architecture consists of an input layer, an output layer, and several hidden layers. The hidden layers are typically a combination of convolutional layers and max pooling layers, with the first extracting the presence of features in an input image, such as edges, and the latter reducing the number of parameters. This reduction has two effects: (1) it lowers the number of trainable parameters to reduce computational cost, and (2) it "simplifies" the features, retaining only the ones that are most relevant.

The U-Net, a CNN proposed by Ronneberger et al. for biomedical image segmentation, can be trained end-to-end from relatively few images, and yields good initial segmentation, outperforming the prior best method (a sliding-window convolutional network) [18]. As opposed to regular classification tasks, where the output to an image is a single class label, the desired output also includes localization, i.e., a class label is assigned to each pixel. This is also called semantic segmentation. This network relies strongly on data augmentation to use the data more efficiently. It consists of a contracting (encoder) path, where features are extracted, and an expansive (decoder) path, where the segmentation mask is generated.

The contracting path has repeated application of convolutions, each followed by a rectified linear unit (ReLU) activation function and a max pooling operation. At each down sampling step, the number of feature channels is doubled. Every step in the expansive path consists of an upsampling of the feature map followed by an up-convolution that halves the number of feature channels, a concatenation with the correspondingly cropped feature map from the contracting path, and two convolutions, each followed by a ReLU. At the final layer, a 1x1 convolution is used to map the final feature vector to a binary segmentation image. In total the network has 23 convolutional layers. See Figure 3 for the U-shaped architecture of the network [7, 18].

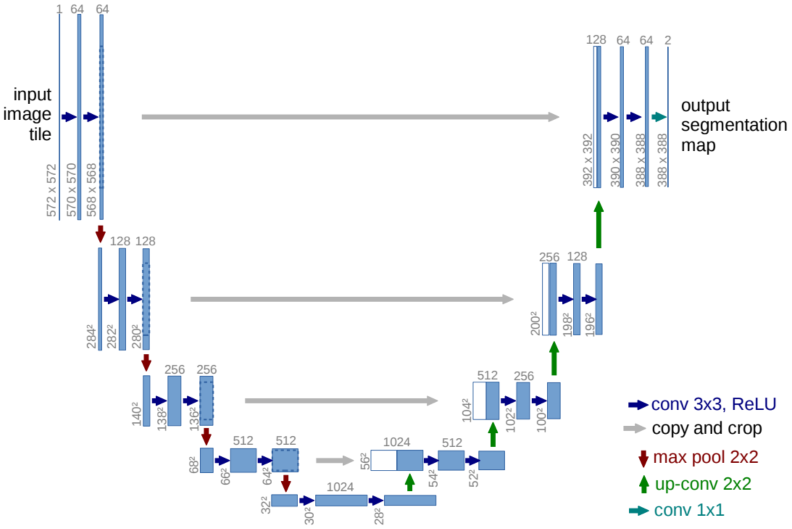


Figure 3. This diagrammatic representation of the U-net shows the contracting (encoder) path, consisting of applications of convolutions, ReLU activation functions, and max pooling operations, and the expansive (decoder) path with a sequence of up-convolutions and concatenations. Adapted from [18].

1.4 Project definition

In current clinical practice, an expert clinician places ROIs manually. Such a process is time-consuming, labor-intensive, and prone to observer variation. Hence, automatic contouring of ROIs is preferred. However, due to poor grey value contrast and complex anatomy, this task can be challenging. Several CNNs have been proposed and evaluated for automatic pancreatic tumor segmentation, yielding average Dice Similarity Coefficients (DSCs) ranging from 0.73 in anatomical MR images to 0.83 in CT scans. The interobserver variations were similar, as they were estimated to be around 0.71% [22, 23]. To my knowledge, the application of U-net for automatic semantic segmentation of the pancreas in quantitative MR images has not been researched.

Therefore, the aim of the present project is to research the use of CNNs, specifically U-Net, to automatically and accurately contour pancreatic tumors on IVIM MR images, so quantitative information can be extracted from the ROI. Its use can reduce bias due to visual inspection and further contribute to pancreatic tumor assessment.

2 Materials and Methods

2.1 Dataset

The dataset consisted of volumes of three studies. The DWI data was fitted to both the IVIM model and an ADC equation, after which the ADC-maps were used for contouring.

MATRIX: This study included patients with one or multiple pancreatic carcinoma(s) who were scanned prior to and after chemotherapy. These patients were divided between phase 1 (consisting of 21 patients) and phase 2 (consisting of 15 patients). Out of these 36 patients, 15 were only scanned before chemotherapy. This led to a total of 57 labeled volumes for the MATRIX study. In phase 1, three patients (four volumes) were scanned with b-values between 0-600 s/mm^2, whereas the remainder of phase 1 and the whole of phase 2 was scanned with b-values in the range of 100-600 s/mm^2 [16].

REMP: This reproducibility study included 14 patients, which were scanned a total of three times during two separate sessions with several days in between (average: 4.5 days apart, range: 1–8 days), leading to 42 labeled volumes. All these images were acquired using b-values between 0-600 s/mm^2. See Appendix A for the sequence parameters during the data acquisition, adapted from [24].

MIPA: 11 patients were scanned at two time points, prior to and after chemoradiation, except two patients who only underwent one scan prior to chemoradiation. Just as the images in the REMP studies, these images were acquired using b-values between 0-600 s/mm^2.

Combined, these studies included 61 patients with 119 corresponding labeled volumes. These 3D-volumes were stored as Neuroimaging Informatics Technology Initiative (NIfTI) files. More detailed information can be found in Appendix B – NIfTI-header of a volume in the MIPA study. As each volume consisted of 18 axial slices, this brought the total axial slices to 2142, divided into 767 foreground slices (with tumor) and 1375 background slices (without tumor). The mean number of axial foreground slices per volume was 6.45, compared to 11.55 background slices. A short overview of the dataset is shown in Table 1.

Table 1. An overview of the used dataset, divided per study, in the number of patients, volumes, and labeled axial slices.

Study	Number of patients	Number of volumes	Number of labeled axial slices
MATRIX	36	57	465
REMP	14	42	183
MIPA	11	20	119
Total	61	119	767

For input, all volumes were sliced in the axial plane to convert the 3D-volumes into 2D-images. Appendix C1 shows the volume of a patient in the MIPA study, acquired before chemoradiation, with Appendix C2 showing the corresponding masks. The highlighted slice in the red rectangular is closer examined in Figure 4 below. As this image shows, there was a class imbalance of foreground and background pixels. The mean percentage of tumor pixels in the labeled slices was 0.83%, whereas this percentage dropped to 0.30% for whole volumes.

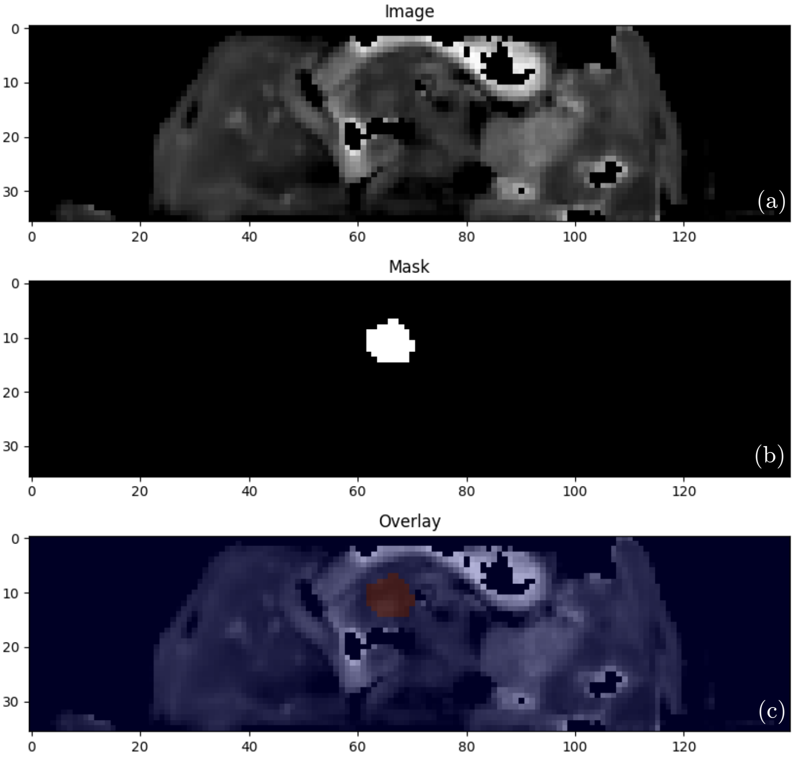


*Figure 4. An axial slice of an IVIM MR volume showing a pancreatic tumor (**a**). Its corresponding label is seen in (**b**) and an overlay of the label, depicted in red, on top of the image is shown in (**c**).*

2.1.1 Dataset division

The dataset was split into a training, validation, and testing portion, using k-fold cross-validation, which is further explained below. The model was trained using only the training and validation portions. During training, its performance was assessed and optimized using the validation data set. Finally, the model's generalization performance was assessed using the test data set, which remained hidden during the training and model validation stage.

Splitting was done per subject level to avoid evaluating the models on images of the same patient they were trained on. As the number of volumes per patient ranged from one to three, these portions slightly differed between tests. The images were shuffled and randomized each epoch to optimize training.

K-fold cross-validation was used to most efficiently train and test on the whole dataset. As the dataset was relatively small, this method ensured that the whole dataset was used for training, while still using the remainder as test set, as can be seen in Figure 5. Note that the training folds consisted of the training and validation portion. Leave one out cross-validation (LOOCV) resulted in a training set of all patients, except for one. As the dataset contained 61 patients, this led to 61 folds and subsequently 61 iterations.

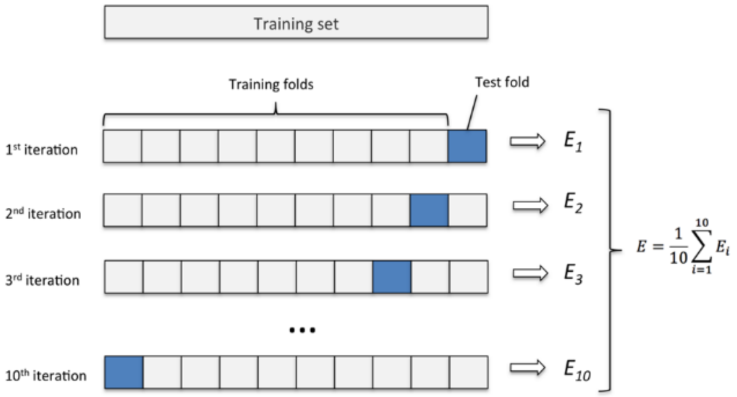


Figure 5. K-fold cross-validation diagram. In this example, the dataset was divided into ten parts, where nine of them were taken as training data. E is the average score of all iterations. Adapted from Niu et al. [25]

2.2 Preprocessing images

2.2.1 Primary tumors and zero-padding

As part of the dataset included multiple other metastases, NIfTI-files were created with only the labeled pancreatic carcinomas. To make the dimensions of each volume of equal size, zero-padding was applied. Figure 6 shows a schematic representation of zero-padding a 3D-volume, from which can be depicted that a layer of black pixels (with value 0) is added around the original volume. It is important to note that resizing was not applied as this would lead to stretching and deformations of images with the smallest dimensions. Therefore, and to make them suitable for the convolution operations of the U-net, the volumes were uniformly rescaled to 160 x 64 x 18 pixels in the xyz-plane. Research has shown that zero-padding does not affect classification accuracy [26]. See Figure 7 for a comparison between the original axial image and the zero-padded image.

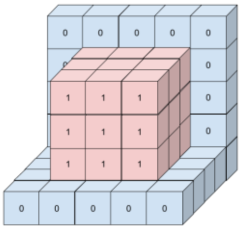


Figure 6 - Schematic overview of zero-padding a 3D matrix. The image contains only voxels with value 1, depicted in red. The padding is done with one layer of 0's. Source: Sparrow Computing.

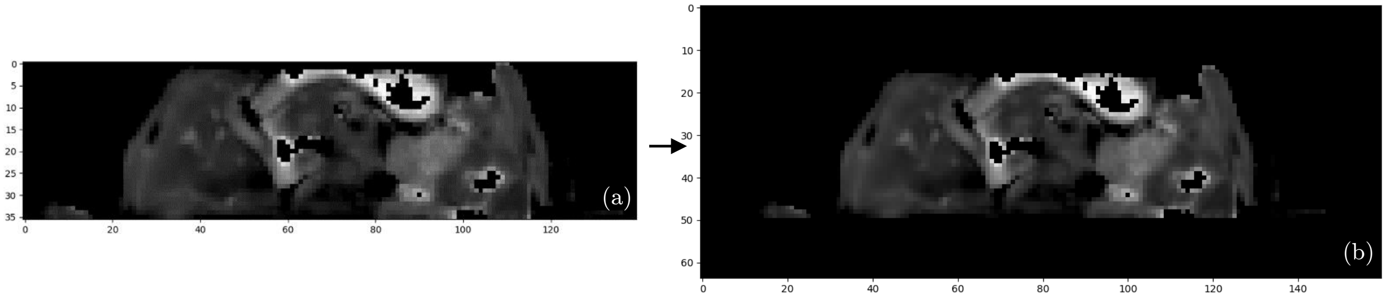


*Figure 7. (**a**) shows an original axial IVIM MR image with dimensions 139 x 36, before applying zero-padding to rescale it to 160 x 64 (**b**).*

2.2.2 Contrast enhancement

Contrast enhancement was used to amplify the edges of the tumor, and thereby help the model with detection. To achieve this, histogram clipping was applied with a minimal percentile of 2.0% and a maximum percentile of 98.0%. All pixels in this lower or upper range were then minimized or maximized, which led to an increase in contrast as can be seen in Figure 8.

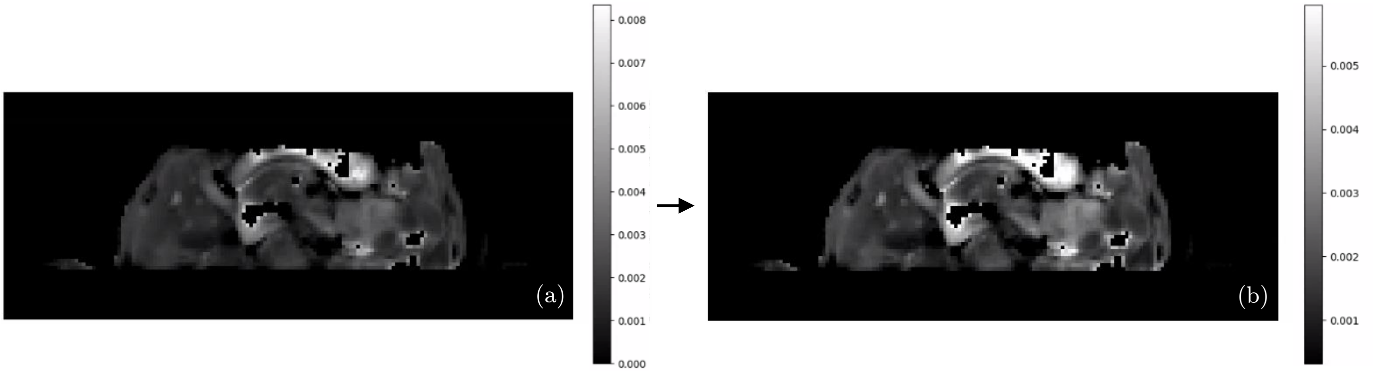


*Figure 8. Comparison between an original image (**a**) and an image on which histogram clipping was applied (**b**). The color bar shows the newly acquired pixel values.*

2.2.3 Data augmentation

To make the model learn more invariant to deformations and hence make it more robust, data augmentation was applied to the training portion. This was done 'on the fly' as the data was loaded into the model, with each image having an 80% probability of undergoing augmentation each iteration. See Table 2 for used transformations.

Table 2. Used transformations on the training portion of the dataset.

Transformations	**Range**
Random rotation	[-10, 10] degrees
Translation	[0.05, 0.05]
Shear	[-5, 5] degrees

These augmentations were chosen according to previous research to closely resemble real scans, but still be unique. The translational range is a fraction of the image width or height. Examples of these augmented images (where the rotation, translation, and shear transforms are visible) can be seen in Figure 9, all originating from the same image.

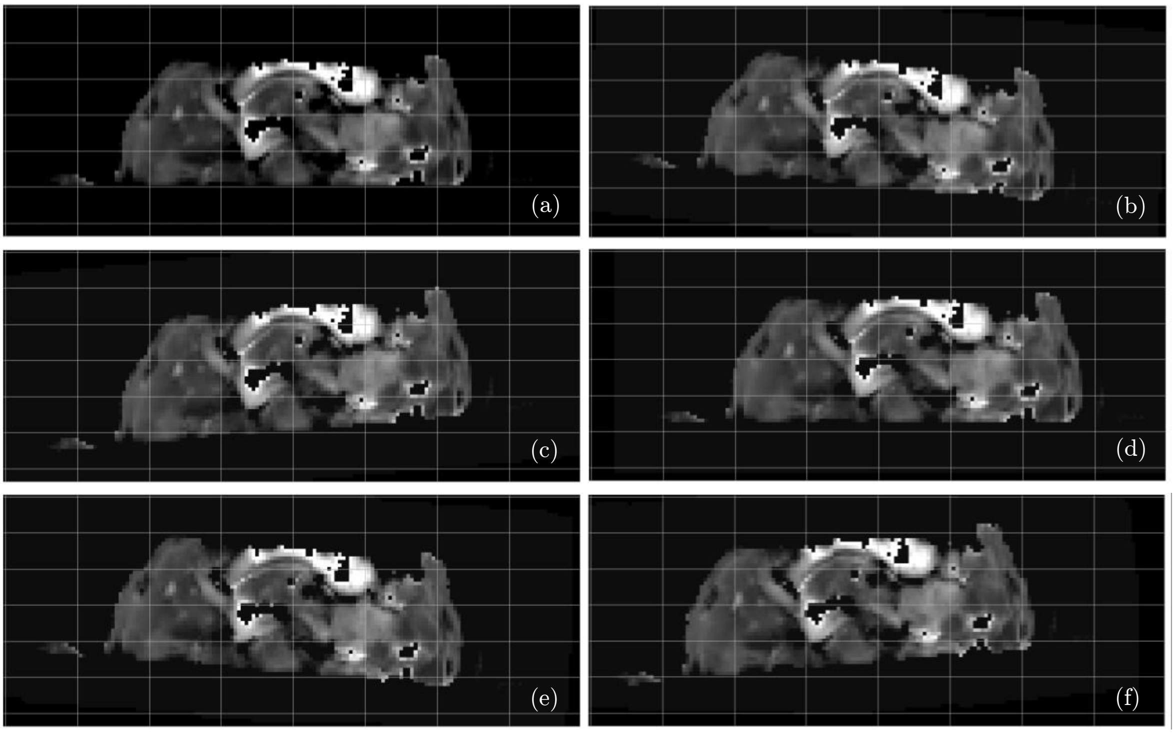


*Figure 9. Visual examples of applied data augmentations with all augmentations applied. (**a**) shows the original image, (**b**) rotation, (**c**) shearing, (**d**) translation, (**e, f**) a mix of all transformations applied.*

2.3 Framework U-net

For this 2D semantic segmentation task, we made use of the U-net [18]. See section 1.3.2 for more information on the general architecture of this CNN. The framework and model summary can be found in Appendix D – U-net model summary. This shows the layers, the output shape of each layer, and the number of trainable parameters. The used hyperparameters are described in 2.4.1.

In comparison to the original U-net [18], a batch normalization layer was added in-between the convolution layer and the ReLU activation function, making the network more stable while training. The dropout rate was set at 0.4 to improve neural network generalization. The dropout technique avoids overfitting, where the model overlearns the details in the training portion before applying this to the testing set. This is avoided by dropping out different random nodes during training.

During training, each epoch consisted of getting the input, zeroing the gradients, performing a forward pass to the network, computing the training loss (error), performing a backward pass, and performing optimization of the weights for the next epoch. The training loss indicates how well the model is fitting the training data, while the validation loss indicates how well the model fits new data. After each iteration, the weights of the trained model were reset so that each fold started with uninitialized weights. Each iteration was trained for 100 epochs with an early stopping of 10, meaning that if the mean validation loss reached a plateau and did not decrease during 10 epochs, the training was automatically stopped, which was the case in approximately 75% of all iterations. The model was saved each time the mean validation loss reached a new minimum value to return the best model after training.

The used computational power for these experiments was an NVIDIA Tesla P100 PCIe 12 GB GPU, operating at a base frequency of 1190 MHz. The model was built using PyTorch, which is an open-source ML framework based on the Python programming language and the Torch library.

2.4 Experiments

2.4.1 Hyperparameter optimization

To assess the performance of the model during training, the Dice mean validation loss (further explained in 2.4.2) was monitored. Optuna, an automatic hyperparameter optimization software framework for ML, was used to optimize the following hyperparameters: the optimizer and the learning rate (LR). The optimizer is used to update network weights iterative based on training data, and the LR is used to change the model in response to the estimated error each time the model weights are updated. Instead of updating the weight with the full amount, it is scaled by the LR. A value too small may therefore have resulted in a long or never-converging training process, whereas a large value may have resulted in a sub-optimal final set of weights or an unstable training process. During training, the LR changed according to a LR scheduler, which decreased the LR by a factor of 10 if the best mean validation loss did not decrease for 5 epochs.

This framework performed 50 trials during 40 epochs, each with a different combination of hyperparameter values within a certain range (see Table 3), to find the combination of values that ensured the lowest mean validation loss. Adam, an extension to the stochastic gradient descent (SGD), was chosen as an option as it has shown good results in the field of deep learning [27]. Adamax, an extension to Adam, was added to the trials, as this optimizer accelerates the optimization process which could result in a more effective optimization.

Table 3. Hyperparameters: optimizer and learning rate with their possible values during the trials.

Hyperparameter	**Settings**
Optimizer	Stochastic gradient descent (SGD), Adam, Adamax
Learning rate	[1e-5, 1e-1]

As there was certain randomness in the initialization of the network, a seed value of 42 was added to bring consistency between runs. This method was used to have the same initialization between runs, so that randomness was eliminated and results between iterations could be compared.

2.4.2 Loss function comparison

To minimize the error (loss) of the model, a fitting loss function was essential. Three loss functions were compared to determine the optimal option. The first was a Dice loss (DL) function with additive smoothing (also known as Laplace smoothing), given by:

$$DL = 1 - \frac{2|A \cap B| + \varepsilon}{|A| + |B| + \varepsilon}. \tag{2}$$

Here, DL is the calculated Dice loss, A is the predicted area, B is the labeled area, and ε is smoothing factor. Smoothing was used to avoid division by zero and was achieved with an additional term alpha while calculating the conditional probability. The additive smoothing was also done for the bias variance tradeoff and to avoid overfitting. With a higher value, variance of the model was reduced, however, bias may have been increased. See Figure 10 for a schematic overview of this metric.

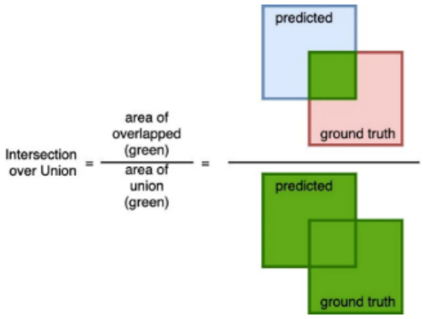


Figure 10. Diagrammatic representation of calculation of the DSC [28]. Source: TowardsDataScience.

Secondly, to account for the imbalance between foreground and background classes, the Focal loss function was applied, given by:

$$FL = -\alpha_t \, (1 - p_t)^{\gamma} * \log(p_t), \tag{3}$$

with p_t defined as:

$$p_t = \begin{cases} \mathrm{p} & if\ y = 1 \\ 1 - \mathrm{p} & otherwise. \end{cases} \tag{4}$$

In the equation, y ∈ [+-1] specifies the ground-truth class and p ∈ [0, 1] is the estimated probability for the class with label y = 1. FL is the Focal loss, α_t ∈ [0, 1] is the weighting factor (defined analogously to p_t), and γ is the modulating term. This function can be seen as a weighted Binary Cross-Entropy (BCE) loss function by adding two hyperparameters α and γ.

These were set to γ = 2, α = 0.25, according to what worked best in literature [29]. While α_t balances the importance of positive/negative examples, it does not differentiate between easy/hard examples. The modulating term γ is added to focus learning on hard misclassified examples, as this factor reduces the loss contribution from easy examples. Setting γ to zero, means FL is equivalent to BCE.

Lastly, to combine the stability of the BCE loss (or log loss) function and the performance of the DSC loss function, these two were combined in a DiceBCE loss function, given by:

$$DiceBCE = -\big(y * log(p) + (1-y)log(1-p)\big) + DSC. \tag{5}$$

DiceBCE is the loss, y_i is the ground-truth, and p is the estimated probability for the class. By adding the DSC and BCE together, the initialization phase of the model benefits from the BCE, after which the influence of the DSC subsequently increases during training.

These loss functions were tested for a duration of 100 epochs with input as explained in 2.4.3. After these shorter tests, the most fitting hyperparameters and loss function were chosen, and the following experiments were performed.

2.4.3 Slice filtering

First, experiments were conducted with only labeled slices, as all background slices of each volume were disregarded. However, to avoid bias in the model towards predictions on every slice for new volumes, the model also needed access to background slices during training.

An addition of two background slices per volume was set, as this led to a ratio of 1/4 background slices and 3/4 labeled slices. As this operation was part of the preprocessing, the specific background slices did not change between epochs. Additional experiments were conducted using zero or five background slices to research the influence of this parameter.

2.4.4 Cropping

Lastly, to counter the foreground/background class imbalance, an additional experiment was done with cropping the volumes to size 128 x 32 x 18. As most pancreatic carcinomas were positioned in the center of the scans (see Figure 11), little information was lost due to cropping, while class imbalance decreased. Cropping resulted in a mean percentage of 0.74% labeled pixels per volume, more than doubling the original ratio. This experiment was done to see if this improved ratio would significantly change the outcome of the model.

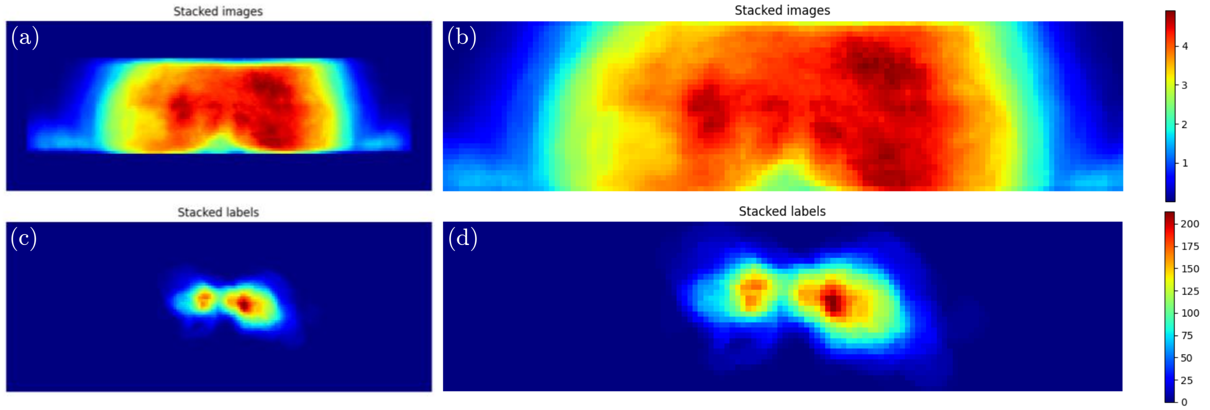


*Figure 11. Heatmaps of the scans (**a**) and labels (**c**) of all zero-padded IVIM MR images. The cropped scans (**b**) and masks (**d**) have a smaller foreground/background class imbalance. The color represents the sum of pixels in a certain location after stacking all axial slices of all volumes of the dataset combined.*

2.5 Performance and statistical analysis

Model performance was evaluated on a never-before seen part of the dataset, using the DSC metric. The DSC of each individual axial slice was calculated, before calculating the mean of the 3D-volume in the testing portion in that fold. After training, the mean and median DSC, as well as the standard deviation (SD) between all folds were calculated. De SD is calculated by:

$$SD = \sqrt{\frac{\sum|x-\mu|^2}{N}}. \tag{5}$$

In this formula, SD is de standard deviation, x is a value in the data set, μ is the mean of the data set, and N is the number of data points. With this calculation, the average amount of variability in the dataset was calculated, and a more accurate analysis could be made of the performance between models, and in comparison to literature.

3 Results

3.1 Hyperparameter optimization

Figure 12 (left) shows the contour plot and (right) shows the intermediate monitoring value (mean Dice validation loss) during the performed trials with Optuna. From this Figure, the optimal optimizer and LR were determined, i.e., optimizer Adamax and a starting LR of 0.028.

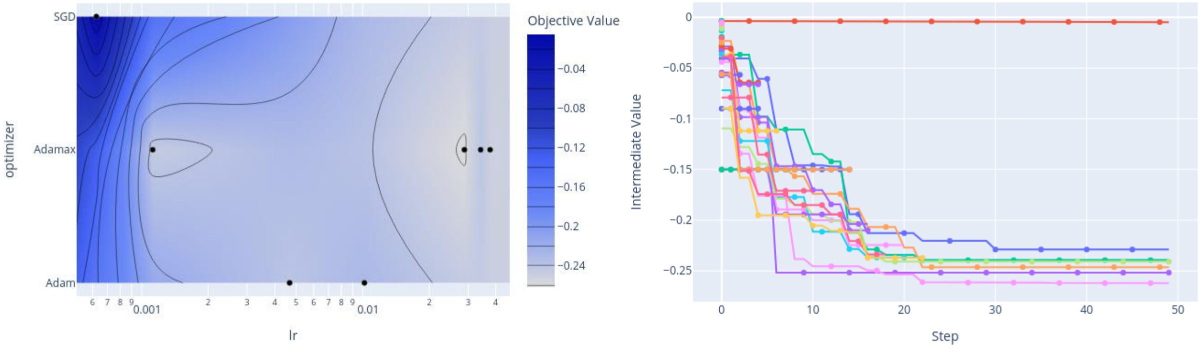


Figure 12. The contour map (left) and intermediate values plot (right) of performed trials with optimizers SGD, Adam, and Adamax and a LR between 1e-5 and 1e-1, monitoring mean validation loss.

3.2 Loss function comparison

Three loss functions were explored, in combination with the optimized hyperparameters. Learning curves of models trained on only foreground slices (as explained in 2.4.3), using (1) Dice loss, (2) Focal loss, and (3) DiceBCE can be seen in Figure 13.

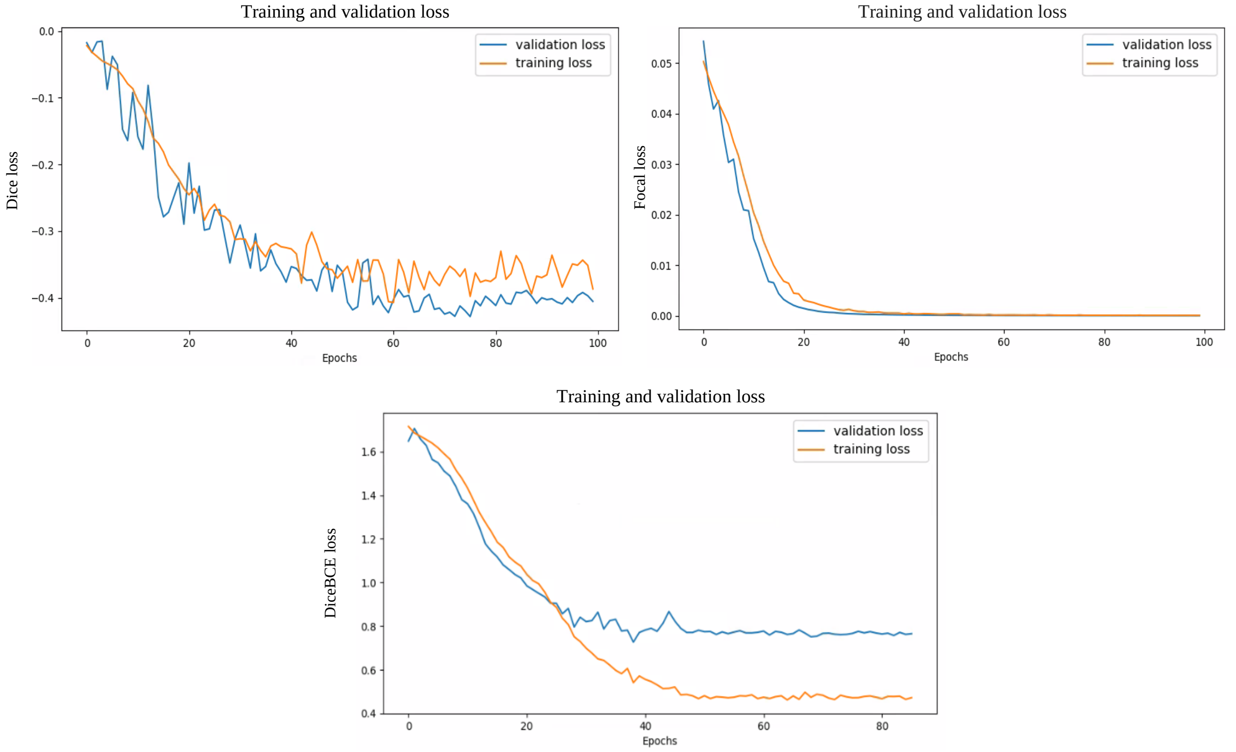


*Figure 13. Representative learning curves of the different loss functions, with training and validation loss values over the number of epochs for the Dice loss (**a**), the Focal loss (**b**) and the DiceBCE loss (**c**).*

These graphs show a quick decrease in training and mean validation loss for all three functions. However, for the FL, the validation and training losses quickly reached a minimum close to 0, whereas the DiceBCE and Dice training loss slowly reached a plateau. For these trials, models trained using the Focal loss yielded a mean DSC of 0.01 (+- 0.01) for the evaluated foreground slices. Altering the values of $\alpha \in [0, 1]$, and $\gamma \in [0, 2]$ did not achieve a significant change in results. The mean DSC for the evaluated foreground slices with the Dice loss function was 0.30 (+- 0.15), just above the mean DSC for the DiceBCE of 0.27 (+- 0.17).

From these experiments, the following hyperparameters were determined for the next experiments: optimizer Adamax, a LR of 0.028 with a LR scheduler and the Dice loss function.

3.3 Slice filtering

3.3.1 Training and evaluation on zero-padded foreground slices: Learning curves

The performances of the best- and worst-performing folds, trained and evaluated on only zero-padded foreground slices, are shown in Figure 14. Although noisy, the mean Dice training and validation losses in both instances decreased over the number of epochs. In both instances, the loss values decreased in a similar pattern.

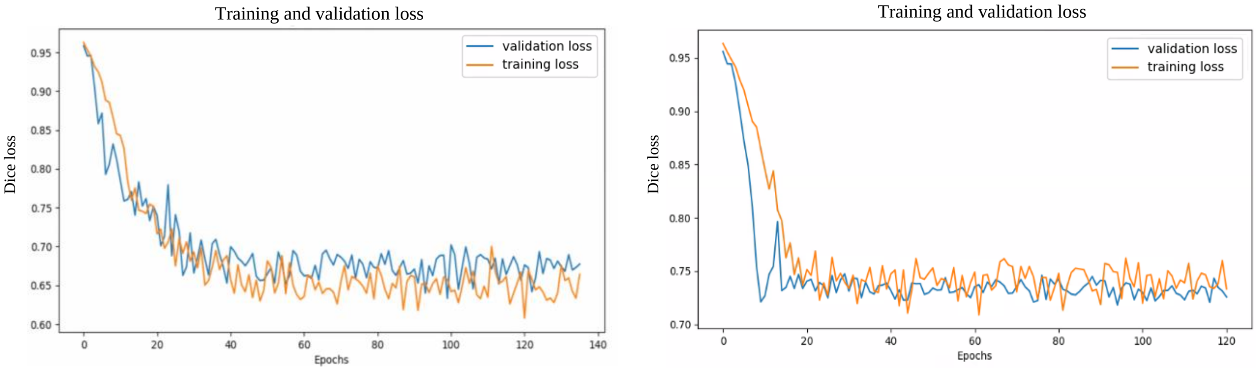


Figure 14. The Dice training and validation loss values over the number of epochs during training of a model with only foreground slices as input, with the left image the best-performing model and the right image the worst-performing model.

3.3.2 Training and evaluation on zero-padded foreground slices: DSCs and predictions

The calculated DSC per patient can be found in Appendix E1 - DSCs of LOOCV folds: foreground slices, resulting in a mean DSC of 0.39 (+- 0.22). Figure 15 shows a slice of the patient for whom the DSC was the highest, i.e., 0.78. The overlap in each axial foreground slice can be found in Appendix E2 – Overlap of ground truth and predictions from the best-performing volume: foreground slices.

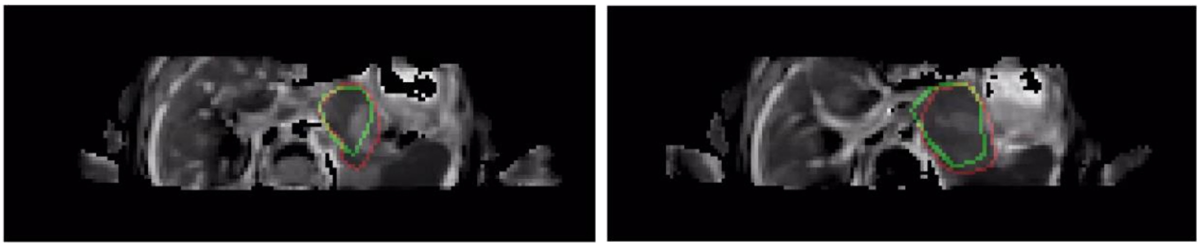


Figure 15. Two slices of the volume with the most accurately predicted tumor. The labeled tumor, or ground truth, is depicted with a red outline, whereas the outline of the predicted tumor is shown in green.

On other volumes, the model predicted less accurately. See Figure 16 for a patient in the MATRIX study with a DSC of 0.37, approximately the mean. Appendix E3 shows the overlap of the ground truth and predictions of each axial foreground slice in this volume.

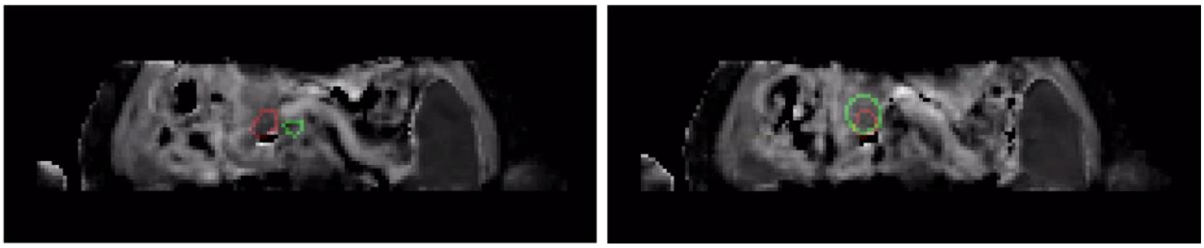


Figure 16. Two slices of a volume with an average DSC close to the mean, with the labeled tumor in red and the predicted tumor in green.

An example of a volume with a DSC of 0 is shown in Figure 17, where the model was not able to produce any predicted ROI.

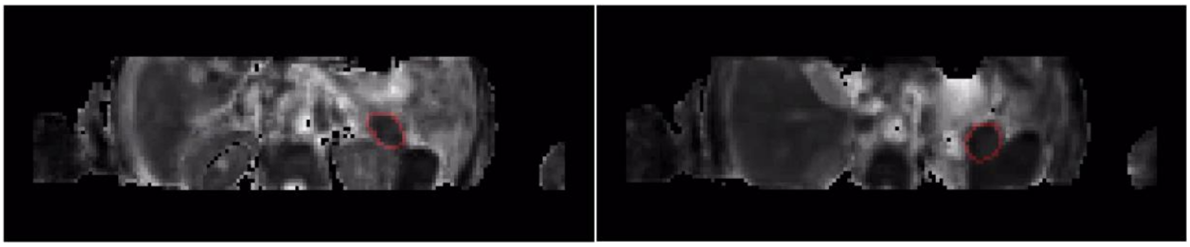


Figure 17. Two slices of a volume for which the model can give no prediction. The region in red depicts the labeled tumor.

3.3.3 Evaluation on zero-padded whole volumes: learning curves

The results in this segment consist of two components: (1) models trained on only foreground slices and tested on whole volumes, and (2) models trained with a varying number of additional background slices and tested on whole volumes. In both instances, the learning curves are as in Figure 19. Although the training loss decreased, the mean validation loss quickly reached a plateau.

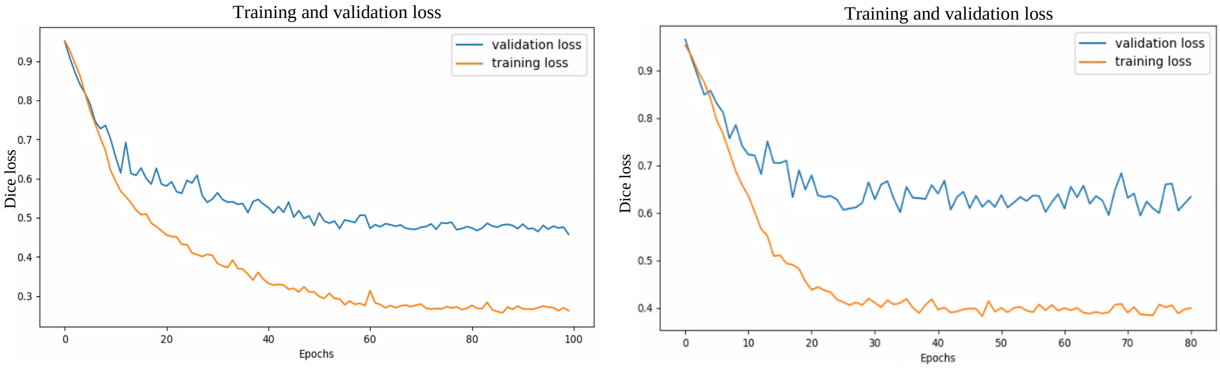


Figure 18. The Dice training and validation loss values over the number of epochs during training of a model with added background images as input. The left image shows the best-performing model, whereas the right image shows the worst-performing model.

3.3.4 Evaluation on zero-padded whole volumes: DSCs and predictions

The first method, training on only foreground slices, yielded a mean DSC of 0.16 (+- 0.14), whereas the second method, varying the number of additional background slices, produced DSCs between 0.15 (+- 0.13) and 0.12 (+- 0.12). The calculated DSCs of all experiments can be seen in Appendix F - DSCs of LOOCV folds: whole volumes.

The results of these varying numbers of input background slices on the DSC for the whole volume are shown in the following table.

Table 4. Mean DSCs, calculated from algorithms with a varying number of added background slices as input. The models were evaluated on whole volumes

Added background slices during training	DSC
0	0.16 (+- 0.14)
2	0.15 (+- 0.13)
5	0.12 (+- 0.12)

Figure 20 shows an evaluation of a whole volume after the first experiment, where the model did not get background images as input (a, b). This figure shows that the model predicted a tumor on a slice where no tumor was present. Moreover, the model predicted a tumor in each of the 18 axial slices. This figure also compares the same slice with a model trained with 2 additional background slices (c, d), and 5 additional background slices (e, f). These images indicate that the prediction on a foreground slice gradually became worse when adding more background slices to the training phase, while still producing a predicted tumor on a background slice.

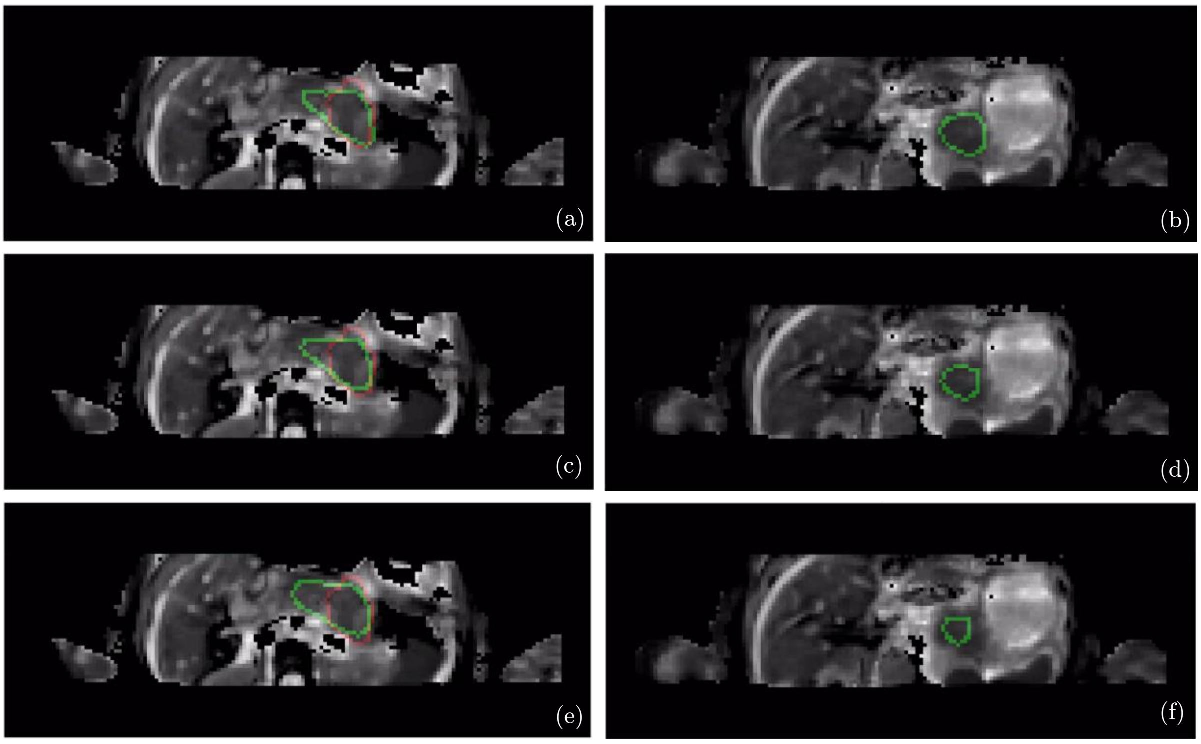


*Figure 19. **(a)** A slice with a well-predicted tumor, **(b)** a slice where there is no tumor, but one is predicted. **(c)** shows the evaluation of the model trained with 2 additional background slices **(d)** no tumor, but one is predicted. **(e, f)** show the model trained with 5 additional background slices.*

3.4 Training and evaluation on cropped foreground slices

Models trained and tested on cropped foreground slices (having more than twice the number of labeled pixels per volume compared to the zero-padded images) yielded a mean DSC of 0.40 (+- 0.22), which is similar to the mean DSC of zero-padded training and testing images. Appendix G shows the DSCs of each fold, having these cropped images as input. Note that these images do miss some information at the borders of the scans. Figure 20 shows the output of this model on a very poorly predicted volume, the same volume as in Figure 17.

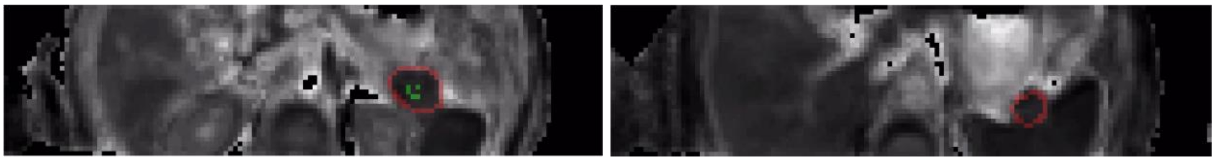


Figure 20. Two slices of a volume for which the DSC is approximately zero. The outline of the labeled tumor is shown in red, the outline of the predicted tumor in green.

3.5 Overall comparison of prediction performance

Figure 21a shows the comparison of all results regarding the prediction experiments. The first prediction experiment shows the DSCs of models trained and evaluated on zero-padded foreground slices. The second shows the DSCs of models trained on zero-padded foreground slices, but evaluated on whole volumes. Subsequently, plot 3 and 4 show DSCs of models trained on zero-padded foreground slices, but with an addition of 2 and 5 background slices, and evaluated on whole volumes. Lastly, experiment five shows the results of models trained and evaluated on cropped foreground slices. This graph shows the decrease in prediction performance when evaluated on whole volumes, opposed to only foreground slices. Furthermore, it shows that DSCs decrease when increasing the number of background slices to training. This also shows that zero-padding and cropping yield similar results for training and evaluating on zero-padded foreground slices.

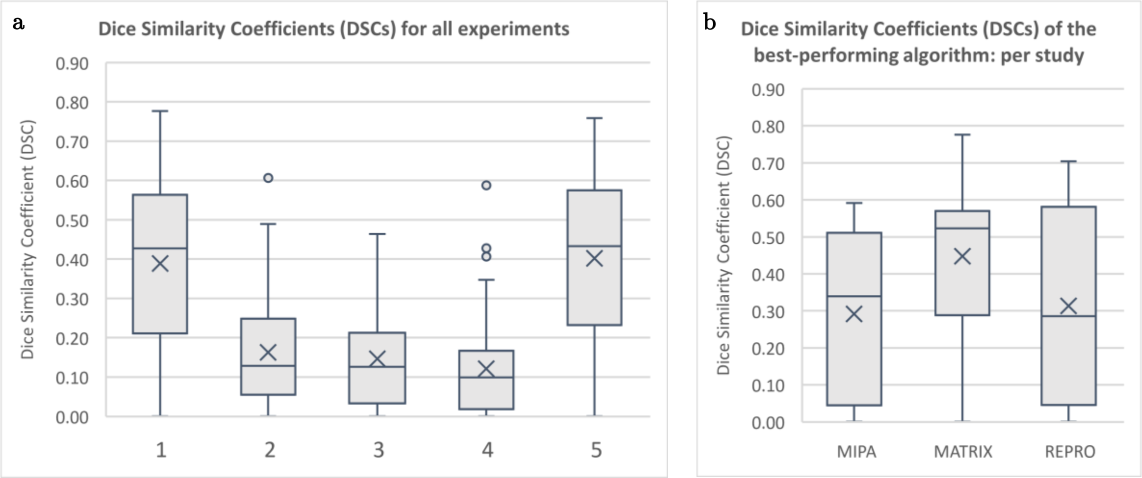


*Figure 21. (**a**) Boxplots for alle performed experiments. (1) shows the DSCs of models trained and evaluated on zero-padded foreground slices. (2) shows the DSCs of models trained on zero-padded foreground slices, but evaluated on whole volumes. (3) and (4) show DSCs of models trained on zero-padded foreground slices with an addition of two and five background slices, and evaluated on whole volumes. (5) shows the results of models trained and evaluated on cropped foreground slices. (**b**) DSCs per study, predicted in experiment (1).*

A comparison of DSCs per study, predicted by the best-performing fold in experiment (1), is shown in Figure 22. Although the means between studies do not show great variations, the interquartile range (IQR) for the MATRIX study is significantly smaller, compared to the IQR of the MIPA or REPRO study.

4 Discussion and conclusions

In current clinical practice contouring pancreatic tumors is done manually, which is time-consuming, labor-intensive, and prone to observer variation. Therefore, the aim of this project was to research whether U-Net can be used to automatically and accurately contour pancreatic tumors on IVIM MR images. The experiments showed that algorithms trained and evaluated on foreground slices yielded the highest mean DSC with 0.39 (+- 0.22) for zero-padded images, and 0.40 (+- 0.22) for cropped images. This was achieved with the Dice loss with smoothing, Adamax optimizer, and an initial LR of 0.028, including a LR scheduler.

Although these experiments yielded the highest DSCs, the results show great variance in DSCs between patients, ranging from 0 to 0.78. This maximum DSC is comparable to the mean DSCs of experiments with anatomical MR (0.73) or CT images (0.83) [22, 23]. However, multiple folds produced a model that was unable to predict a tumor on the evaluation set, in some instances not producing a predicted ROI, at other times contouring the wrong ROI. This could be caused by dissimilarities between the training and evaluation set, although these dissimilarities were counteracted by the data augmentation. However, the added transformations and their values may not have been enough to make the algorithm as robust and universally applicable to be able to predict well on never-before seen scans.

This variance was also seen between the used datasets, although not as severe. The mean DSC for the MIPA study was 0.29, whereas that was 0.45 for the MATRIX study, and 0.31 for the REMP study, as seen in Figure 21b, suggesting that the MIPA study consists of images that are harder to predict the ROI for. However, as the complete dataset consisted of 465 MATRIX, 183 REMP, and 119 MIPA labeled axial slices, the algorithms were better trained towards the first, hence also better predicting masks on those images. This can also be derived from the smaller IQR for this study, compared to the REMP and MIPA study. This seems to indicate there is no correlation between the quality of the scans of the different studies, and the performance of the algorithm.

Additionally, the results showed that cropping the images, hence removing the edges of the images and decreasing the class imbalance, led to a similar performance of the models. This leads to the conclusion that cropping to this extent does not improve the results, therefore, also indicating that zero-padding does not decrease prediction performance, which is consistent with the literature [26]. As the information at the border of the images is lost using cropping, it actually is preferred to use zero-padding. However, a downside of zero-padding is the increased image size, thereby also increasing memory footprint, leading to the requirement of more computational power.

Furthermore, as the application of these predictions is aimed toward whole volumes, the algorithms were trained, validated, and evaluated on input data with multiple added background slices. However, the addition of background slices decreased performance significantly compared to algorithms trained, validated, and evaluated on only foreground slices. Both algorithms predicted tumors on all axial slices, resulting in a significantly decrease in DSCs for evaluation on the whole volume in comparison to the model evaluated on only the foreground slices. To counter this, a variety of background slices were added in the training dataset. However, this did not lead to an increase in accuracy. Little to no difference could be found between results with a varying number of added background slices. Moreover, Figure 19 and 21a show that the predictions get gradually worse by a small margin when adding more of the whole input volume, as opposed to only labeled slices, while still producing ROIs on background slices.

The training processes of models trained on only foreground, as depicted in Figure 14, show that these models do not show over- or underfitting. Although noisy, the Dice mean training and validation losses decreased following the same pattern, reaching a plateau at approximately the same value. However, the validation portion was randomly chosen every fold and often consisted of volumes of multiple studies. Adding background slices caused the model to show signs of an unfit model, as can be seen in Figure 18: both the curves for the training and validation loss showed improvement, but a large gap remained between both curves, meaning that it captured the training data well, but was no able to generalize and apply this to the validation set. This could indicate an unrepresentative train dataset, which does not provide sufficient information to learn probable patterns, relative to the validation dataset.

Lastly, these experiments showed the great influence of a fitting loss function. As can be seen in Figure 13, the Focal loss and DiceBCE loss showed good learning curves, but the DSCs these models yield were subpar. Although the Focal loss should have put more emphasis on hard misclassified examples, and hereby countering class imbalance, its influence was outweighed by the BCE. The hyperparameters were chosen on what worked best according to literature, but altering the hyperparameters of the Focal loss did not lead to improved DSCs. However, this could be due to the combination with the used optimizer, as its loss goes very small quickly, reducing any amount of backpropagation. Models trained with the DiceBCE loss had slightly worse DSCs than the ones using the regular Dice loss with smoothing, meaning that the important factor for right classification in this function was the Dice component. However, both of these loss functions were outperformed by the Dice loss with smoothing.

4.1 Limitations, reliability, and validity

As hyperparameters and loss function were optimized for the experiments using only foreground slices for training and evaluation, it might have been a factor in the poorer prediction performance in other experiments. However, obtaining the optimal combination of hyperparameters and loss functions is an iterative process and may differ depending on the dataset, method, or model used. As time was limited, we decided to determine the optimal combination once, with which the subsequent experiments were performed. However, it would be preferable to perform hyperparameter optimization for every individual experiment.

During preprocessing, histogram clipping was used to enhance contrast and amplify edges of the tumor, thereby helping the model with detection. However, as the used ADC maps were quantitative, these pixel intensities had a physical meaning. By applying a different clipping to each volume, that physical meaning was lost. Although the model performed better with this contrast enhancement, this loss of information could have played a factor in the detection performance.

Another limitation of this project was the relatively small size of the dataset. While the LOOCV maximized the use of the dataset, with only 767 labeled axial slices from 61 patients, the algorithms were trained on a limited range of features and aspects to learn from. Furthermore, as this dataset consisted of patients from three different studies, and the mask sizes (and potentially delineation protocol) may differ per study, it might have been too diverse for the algorithm to discover a pattern.

On average, the REMP dataset study contained fewer slices per volume than the other two studies, and it is unknown if the contouring was done in the same manner. As this study was aimed at test-retest variability, which means only regions that were visible on both repeated scans could be contoured, parts of the tumor may deliberately have been excluded as it was not visible on the repeated scan, leading to much fewer labeled slices per volume than the other two studies.

Considering the calculated mean DSC per volume, some patients consisted of three volumes, while others only had one volume. Therefore, the calculated mean DSC of each patient consisted of a varying number of samples (slices), where the patients with multiple scans had a higher number of samples. However, results showed that there were no significant differences between patients with one volume and patients with multiple volumes.

4.2 Recommendations for future research

Future research in automatic pancreatic tumor segmentation could involve the exploration of several tasks. Firstly, to improve model prediction performance, denoising methods could be used, as proposed by Ashraf et al., by filtering and removing patch-based noise on these images [30]. Furthermore, an occlusion sensitivity map can be made to visualize how parts of the scan affect the output of the model by occluding different parts iteratively. This will lead to an understanding of what parts of the image features the network uses to make the particular semantic segmentation, and to provide insight into the reasons how the model predicts. This could be combined with cropping a smaller ROI, e.g., a bounding box around the pancreas. Research shows that this input would improve performance [31], but this would require first having a correct label of the whole pancreas itself and not just the focal lesion.

Another possibility that could be explored is the added use of other image modalities for anatomical scans, linked to the corresponding IVIM MR images. Having anatomical MR or CT images might enhance the training process, thereby improving the model.

More future research could involve the exploration of other ML models for this task, e.g., a 3D U-net, including the axial, coronal, and sagittal axes [19]. With this volumetric segmentation information, the algorithm might perform better on each volume. It extends the previous U-Net architecture from Ronneberger et al. by replacing all 2D operations with their 3D counterparts. However, this approach will limit data points as these will be limited to only the volumes, instead of each slice. Another similar ML model is the nnU-net, a self-configuring method for deep learning-based biomedical image segmentation [32]. This automatic configuring includes preprocessing, network architecture, training and post-processing, and could enhance performance for this task.

4.3 Conclusion

In conclusion, this project has shown that the U-net was not able to accurately predict pancreatic tumors on IVIM MR images, using this dataset. The DSC of the best-performing volume on only foreground slices was comparable to the mean of DSCs in anatomical scans. When whole patients' volumes were evaluated, the performance decreased dramatically, strengthening the point that contouring the ROI cannot be successfully performed using this CNN.

5 References

[1] L. Zhang, S. Sanagapalli and A. Stoita, ”Challanges in Diagnosis of Pancreatic Cancer,” *World Journal of Gastroenterology,* vol. 24, no. 19, pp. 2047-2060, 2018.

[2] M. Ilic and I. Ilic, ”Epidemiology of pancreatic cancer,” *World Journal of Gastroenterology,* vol. 22, no. 44, pp. 9694-9705, 2016.

[3] T. Kamisawa, L. D. Wood, T. Itoi and K. Takaori, ”Pancreatic cancer,” *The Lancet,* vol. 388, no. 10039, pp. 73-85, 2016.

[4] R. L. Siegel, K. D. Miller and A. Jemal, ”Cancer statistics, 2022,” *CA: A Cancer Journal for Clinicians,* pp. 7-33, 2022.

[5] O. Strobel, J. Neoptolemos, D. Jäger and M. W. Büchler, ”Optimizing the outcomes of pancreatic cancer surgery.,” *Nature reviews Clinical oncology,* vol. 16, no. 1, pp. 11-26, 2019.

[6] M. M. Mortenson, M. H. Katz, E. P. Tamm, M. S. Bhutani,, H. Wang, D. B. Evans and J. B. Fleming, ”Current diagnosis and management of unusual pancreatic tumors,” *The American Journal of Surgery,* vol. Volume 196, no. 1, pp. 100-113, 2008.

[7] T. Boers, Y. Hu, E. Gibson, D. Barratt, E. Bonmati, J. Krdzalic, F. van der Heijden, J. Hermans and H. Huisman, ”Interactive 3D U-net for the segmentation of the pancreas in computed tomography scans,” *Physics in Medicine & Biology,* vol. 65, no. 6, 2020.

[8] M. M. Al-Hawary, I. R. Francis and E. K. Fishman, ”Pancreatic ductal adenocarcinoma radiology reporting template: consensus statement of the Society of Abdominal Radiology and the American Pancreatic Association,” *Radiology,* vol. 270, no. 1, pp. 248-260, 2014.

[9] J. R. Treadwell, H. M. Zafar, M. D. Mitchell, K. Tipton, U. Teitelbaum and J. Jue, ”Imaging Tests for the Diagnosis and Staging of Pancreatic Adenocarcinoma: A Meta-Analysis.,” *Pancreas,* vol. 45, no. 6, p. 789–795, 2016.

[10] C. Messina, R. Bignone, A. Bruno, A. Bruno, F. Bruno, M. Calandri, D. Caruso, P. Coppolino, R. Robertis, F. Gentili, I. Grazzini, R. Natella, P. Scalise, A. Barile, R. Grassi and D. Albano, "Diffusion-Weighted Imaging in Oncology: An Update," *Cancers,* vol. 12, no. 6, p. 1493, 2020.

[11] D. Le Bihan, "What can we see with IVIM MRI?," *NeuroImage,* vol. 187, pp. 56-57, 2019.

[12] A. R. Padhani, G. Liu, D. M. Koh, T. L. Chenevert, H. C. Thoeny, T. Takahara, A. Dzik-Jurasz, B. D. Ross, M. Van Cauteren, D. Collins, D. Hammoud, G. J. Rustin, B. Taouli and P. L. Choyke, "Diffusion-Weighted Magnetic Resonance Imaging as a Cancer Biomarker: Consensus and Recommendations," *Neoplasia,* vol. 11, no. 2, pp. 102-125, 2009.

[13] D. Le Bihan, R. Turner and J. R. Macfall, "Effects of intravoxel incoherent motions (IVIM) in steady-state free precession (SSFP) imaging: application to molecular diffusion imaging.," *Magnetic resonance in medicine,* vol. 10, no. 3, pp. 324-337, 1989.

[14] Y. Wang, H. Huang, C. Zheng, B. Xiao, O. Chevallier and W. Wang, "Diffusion-weighted MRI of the liver: challenges and some solutions for the quantification of apparent diffusion coefficient and intravoxel incoherent motion.," *American journal of nuclear medicine and molecular imaging,* vol. 11, no. 2, pp. 107-142, 2021.

[15] P. Kingsley and W. Monahan, "Selection of the optimum b factor for diffusion-weighted magnetic resonance imaging assessment of ischemic stroke," *Magnetic resonance in medicine,* vol. 51, no. 5, pp. 996-1001, 2004.

[16] E. N. Pijnappel, N. P. Wassenaar, O. J. Gurney-Champion, R. Klaassen, K. van der Lee, M. C. Pleunis-van Empel, D. J. Richel, M. C. Legdeur, A. J. Nederveen, H. W. van Laarhoven and J. W. Wilmink, "Phase I/II Study of LDE225 in Combination with Gemcitabine and Nab-Paclitaxel in Patients with Metastatic Pancreatic Cancer," *Cancers,* vol. 13, no. 19, 2021.

[17] E. Versteijne , O. J. Gurney-Champion, . A. van der Horst, E. Lens, M. Kolff , J. Buijsen, . G. Ebrahimi, . K. J. Neelis, C. R. N. Rasch, J. Stoker, . M. van Herk, A. Bel and G. van Tienhoven, "Considerable interobserver variation in delineation of pancreatic cancer on 3DCT and 4DCT: a multi-institutional study," *Radiation Oncology,* vol. 12, no. 1, p. 58, 2017.

[18] O. Ronneberger, P. Fischer and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015,* pp. 234-241, 2015.

[19] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox and O. Ronneberger, "3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation," *Medical Image Computing and Computer-Assisted Intervention,* pp. 424-432, 2016.

[20] A. Selvikvåg and A. Lundervold, "An overview of deep learning in medical imaging focusing on MRI," *Zeitschrift für Medizinische Physik,* vol. 29, no. 2, pp. 102-127, 2019.

[21] E. Alpaydin, Introduction to Machine Learning, Massachusetts: Massachusetts Institute of Technology, 2020.

[22] Y. Liang, D. Schott, Y. Zhang, Z. Wang, H. Nasief, E. Paulson, W. Hall, P. Knechtges, B. Erickson and X. Allen Li, "Auto-segmentation of pancreatic tumor in multi-parametric MRI using deep convolutional neural networks," *Radiotherapy and Oncology,* vol. 145, pp. 193-200, 2020.

[23] Z. Guo, L. Zhang, L. Lu, M. Bagheri, R. M. Summers, M. Sonka and J. Yao, "Deep LOGISMOS: Deep learning graph-based 3D segmentation of pancreatic tumors on CT

scans," *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018),* pp. 1230-1233, 2018.

[24] O. J. Gurney-Champion, R. Klaassen, M. Froeling, S. Barbieri, J. Stoker, M. Engelbrecht, J. Wilmink, M. Besselink, A. Bel, H. van Laarhoven and A. J. Nederveen, "Comparison of six fit algorithms for the intra-voxel incoherent motion model of diffusion-weighted magnetic resonance imaging data of pancreatic cancer patients," *PloS one,* vol. 13, no. 4, 2018.

[25] M. Niu, Y. Li, C. Wang and K. Han, "RFAmyloid: A Web Server for Predicting Amyloid Proteins," *International journal of molecular sciences,* no. 19, 2018.

[26] M. Hashemi, "Enlarging smaller images before inputting into convolutional neural network: zero-padding vs. interpolation," *Journal of Big Data,* no. 6, 2019.

[27] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *arXiv preprint arXiv:1412.6980,* 2014.

[28] H. Jing, "Biomedical Image Segmentation - U-Net," 14 June 2020. [Online]. Available: https://jinglescode.github.io/2019/11/07/biomedical-image-segmentation-u-net/.

[29] T.-Y. Lin, P. Goyal, R. Girshick, K. He and P. Dollar, "Focal Loss for Dense Object Detection," *IEEE International Conference on Computer Vision (ICCV),* pp. 2999-3007, 2017.

[30] M. Ashraf, W. Quiñones Robles, M. Kim, Y. Ko and M. Yi, "A loss-based patch label denoising method for improving whole-slide image analysis using a convolutional neural network," *Scientific Reports,* vol. 12, 2022.

[31] X. Feng, K. Qing, N. Tustison, C. Meyer and Q. Chen, "Deep convolutional neural network for segmentation of thoracic organs-at-risk using cropped 3D images," *Medical Physics,* vol. 46, no. 5, pp. 2169-2180, 2019.

[32] F. Isensee, . P. F. Jaeger, S. A. A. Kohl,, J. Petersen and K. H. Maier-Hein , "nnU-Net: a self-configuring method for deep learning-based biomedical image segmentation," *Nature Methods volume,* vol. 18, p. 203–211, 2021.

Appendices

Appendix A – Data acquisition sequence parameters for the MATRIX study.

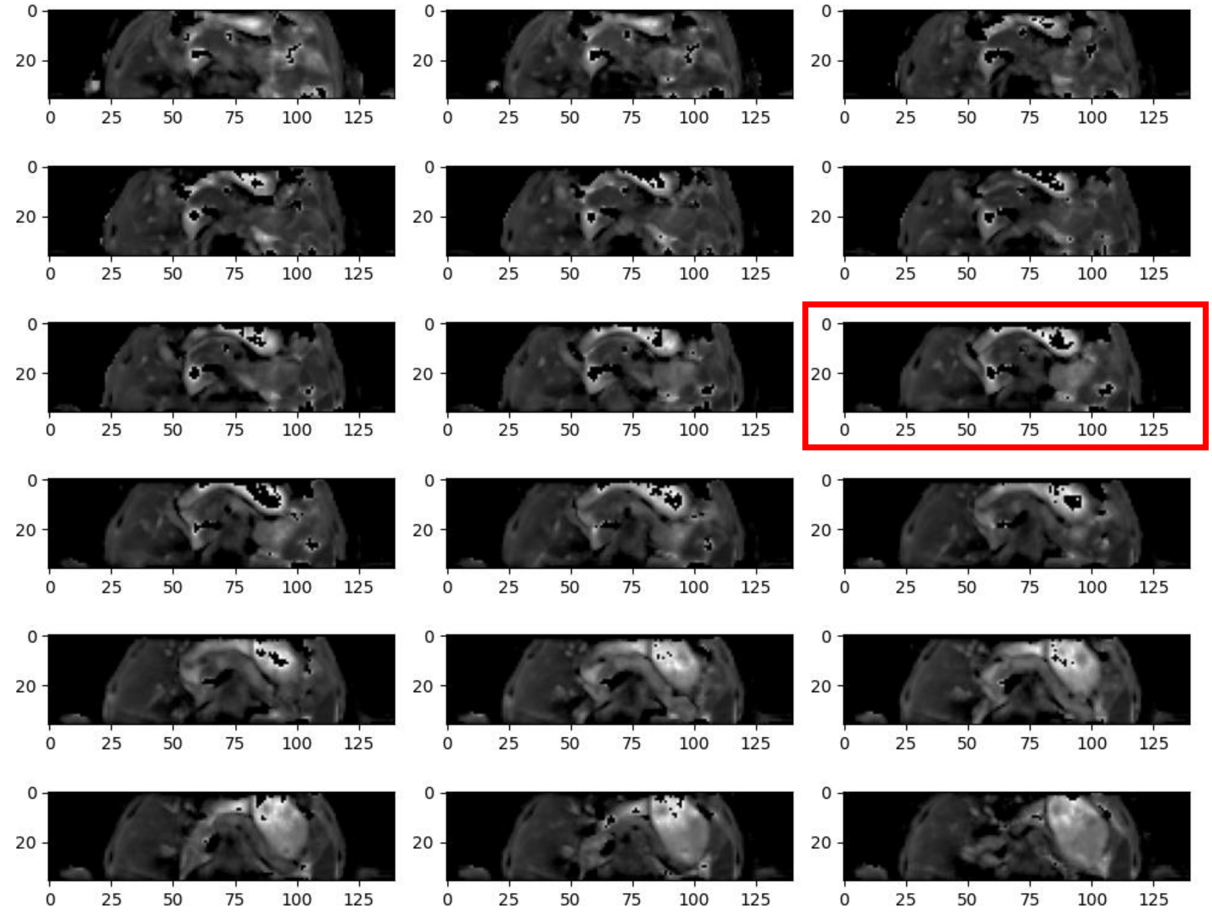
Adapted from [24].

	DWI	T1W GE
FOV (RL × AP) (mm^2)	432 × 108	400 × 353
Acquisition matrix	144 × 34	236 × 208
Slices	18	56
Slice thickness/gap (mm)	3.7/0.3	1.7/-
TR[1]/TE/ ΔTE (ms)	>2200/45/-	4.7/1.15/1.0
FA (°)	90	10
BW (Hz/voxel)	59 (phase direction)	1602 (frequency)
Parallel imaging	1.3 (AP)	2/1.5 (RL/AP)
Partial Fourier	0.8	no
Respiratory compensation	Respiratory trigger (navigator)	1 breath hold
Fat saturation	Gradient reversal during slice selection + SPIR	Dixon reconstruction
b-values (s/mm^2) and directions (between brackets)[2]	0 (15), 10 (9), 20 (9), 30 (9), 40 (9), 50 (9), 75 (4), 100 (12), 150 (4), 250 (4), 400 (4) and 600 (16)	
Diffusion times δ/Δ (ms)	10.1/22.6	

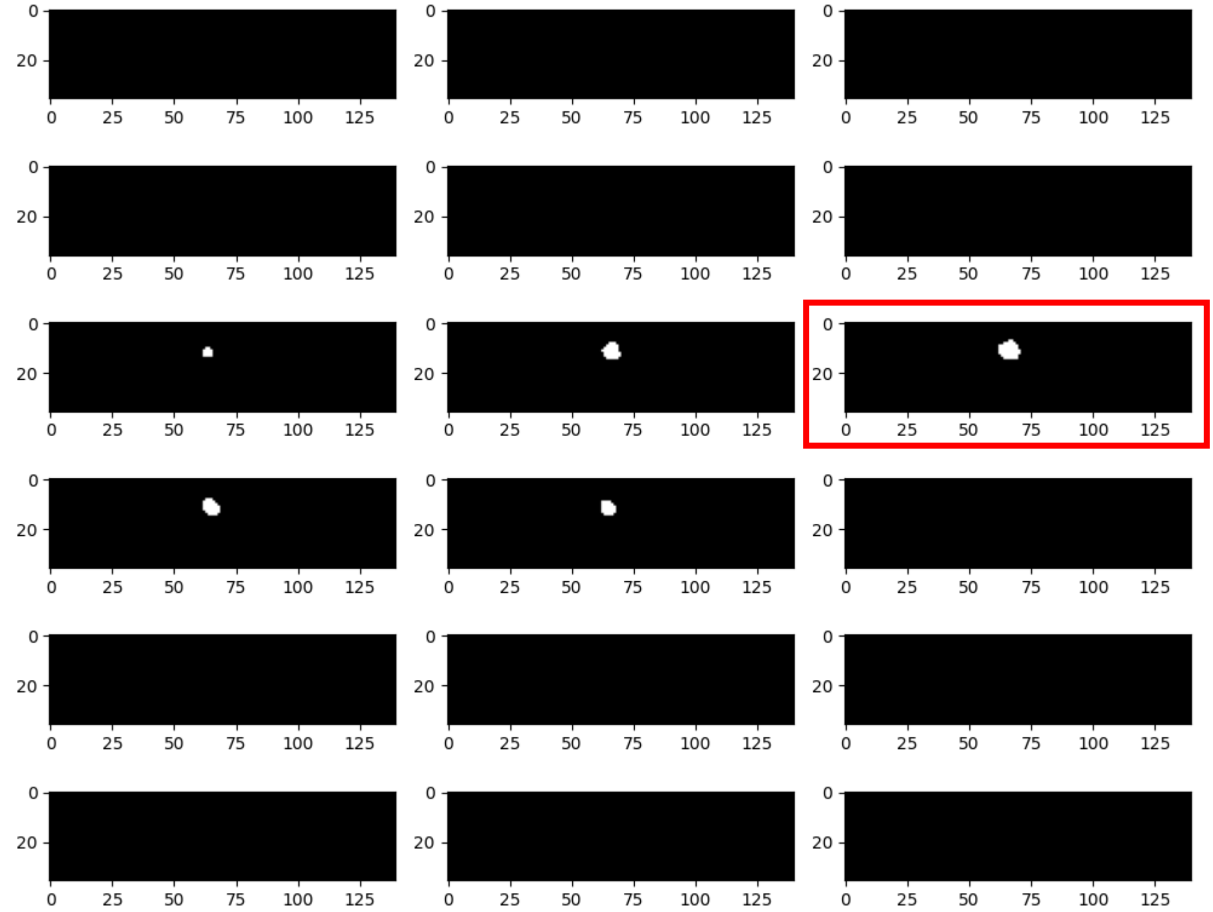
Appendix B – NIfTI-header of a volume in the MIPA study.

```
<class 'nibabel.nifti1.Nifti1Header'> object, endian='<'
sizeof_hdr      : 348
data_type       : b''
db_name         : b''
extents         : 0
session_error   : 0
regular         : b'r'
dim_info        : 0
dim             : [  3 140  36  18   1   1   1   1]
intent_p1       : 0.0
intent_p2       : 0.0
intent_p3       : 0.0
intent_code     : none
datatype        : float32
bitpix          : 32
slice_start     : 0
pixdim          : [-1.  3.  3.  4.  0.  0.  0.  0.]
vox_offset      : 0.0
scl_slope       : nan
scl_inter       : nan
slice_end       : 0
slice_code      : unknown
xyzt_units      : 2
cal_max         : 0.0
cal_min         : 0.0
slice_duration  : 0.0
toffset         : 0.0
glmax           : 0
glmin           : 0
descrip         : b''
aux_file        : b''
qform_code      : aligned
sform_code      : scanner
quatern_b       : -0.0
quatern_c       : 1.0
quatern_d       : 0.0
qoffset_x       :  205.56354
qoffset_y       : -37.24916
qoffset_z       : 8.780899
srow_x          : [ -3.       -0.        0.       205.56354]
srow_y          : [ -0.        3.       -0.       -37.24916]
srow_z          : [0.       0.       4.       8.780899]
intent_name     : b''
magic           : b'n+1'
```

Appendix C1 – Axial slices of a volume of a patient in the MIPA-study.



Appendix C2 – Corresponding axial labels for a volume of a patient in the MIPA-study



Appendix D – U-net model summary

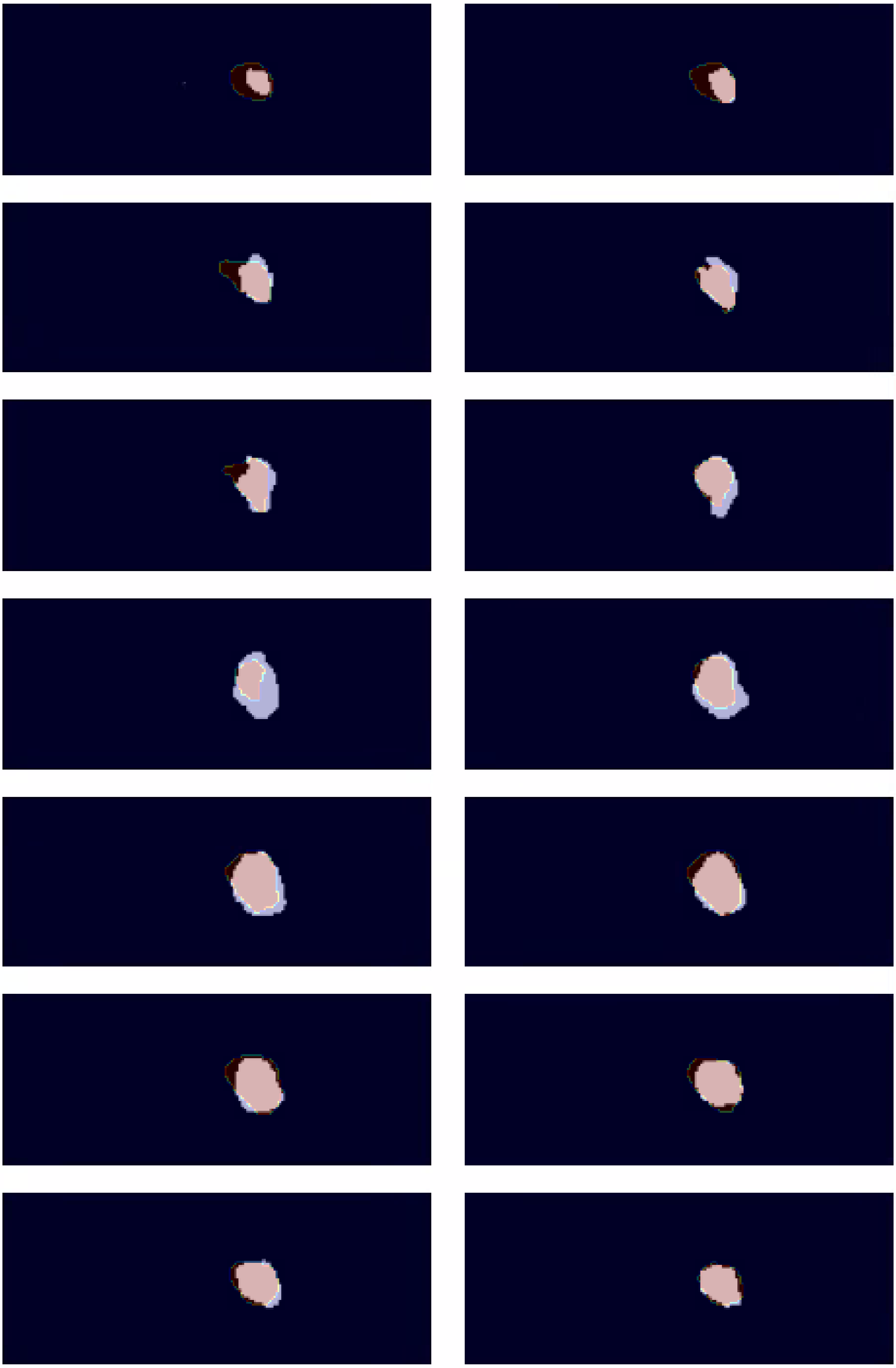
Layer (type)	Output Shape	Param #
Conv2d-1	[-1, 64, 160, 64]	640
BatchNorm2d-2	[-1, 64, 160, 64]	128
Dropout2d-3	[-1, 64, 160, 64]	0
Conv2d-4	[-1, 64, 160, 64]	36,928
BatchNorm2d-5	[-1, 64, 160, 64]	128
Dropout2d-6	[-1, 64, 160, 64]	0
DownConv-7	[-1, 64, 160, 64]	0
MaxPool2d-8	[-1, 64, 80, 32]	0
Conv2d-9	[-1, 128, 80, 32]	73,856
BatchNorm2d-10	[-1, 128, 80, 32]	256
Dropout2d-11	[-1, 128, 80, 32]	0
Conv2d-12	[-1, 128, 80, 32]	147,584
BatchNorm2d-13	[-1, 128, 80, 32]	256
Dropout2d-14	[-1, 128, 80, 32]	0
DownConv-15	[-1, 128, 80, 32]	0
MaxPool2d-16	[-1, 128, 40, 16]	0
Conv2d-17	[-1, 256, 40, 16]	295,168
BatchNorm2d-18	[-1, 256, 40, 16]	512
Dropout2d-19	[-1, 256, 40, 16]	0
Conv2d-20	[-1, 256, 40, 16]	590,080
BatchNorm2d-21	[-1, 256, 40, 16]	512
Dropout2d-22	[-1, 256, 40, 16]	0
DownConv-23	[-1, 256, 40, 16]	0
MaxPool2d-24	[-1, 256, 20, 8]	0
Conv2d-25	[-1, 256, 20, 8]	590,080
BatchNorm2d-26	[-1, 256, 20, 8]	512
Dropout2d-27	[-1, 256, 20, 8]	0
Conv2d-28	[-1, 256, 20, 8]	590,080
BatchNorm2d-29	[-1, 256, 20, 8]	512
Dropout2d-30	[-1, 256, 20, 8]	0
DownConv-31	[-1, 256, 20, 8]	0
Upsample-32	[-1, 256, 40, 16]	0
Conv2d-33	[-1, 256, 40, 16]	1,179,904
BatchNorm2d-34	[-1, 256, 40, 16]	512
Dropout2d-35	[-1, 256, 40, 16]	0
Conv2d-36	[-1, 256, 40, 16]	590,080
BatchNorm2d-37	[-1, 256, 40, 16]	512
Dropout2d-38	[-1, 256, 40, 16]	0
DownConv-39	[-1, 256, 40, 16]	0
UpConv-40	[-1, 256, 40, 16]	0
Upsample-41	[-1, 256, 80, 32]	0
Conv2d-42	[-1, 128, 80, 32]	442,496
BatchNorm2d-43	[-1, 128, 80, 32]	256
Dropout2d-44	[-1, 128, 80, 32]	0
Conv2d-45	[-1, 128, 80, 32]	147,584
BatchNorm2d-46	[-1, 128, 80, 32]	256
Dropout2d-47	[-1, 128, 80, 32]	0
DownConv-48	[-1, 128, 80, 32]	0
UpConv-49	[-1, 128, 80, 32]	0
Upsample-50	[-1, 128, 160, 64]	0
Conv2d-51	[-1, 64, 160, 64]	110,656
BatchNorm2d-52	[-1, 64, 160, 64]	128
Dropout2d-53	[-1, 64, 160, 64]	0
Conv2d-54	[-1, 64, 160, 64]	36,928
BatchNorm2d-55	[-1, 64, 160, 64]	128
Dropout2d-56	[-1, 64, 160, 64]	0
DownConv-57	[-1, 64, 160, 64]	0
UpConv-58	[-1, 64, 160, 64]	0
Conv2d-59	[-1, 1, 160, 64]	577
==		
Total params: 4,837,249		
Trainable params: 4,837,249		
Non-trainable params: 0		

Appendix E1 – DSCs of LOOCV folds: foreground slices

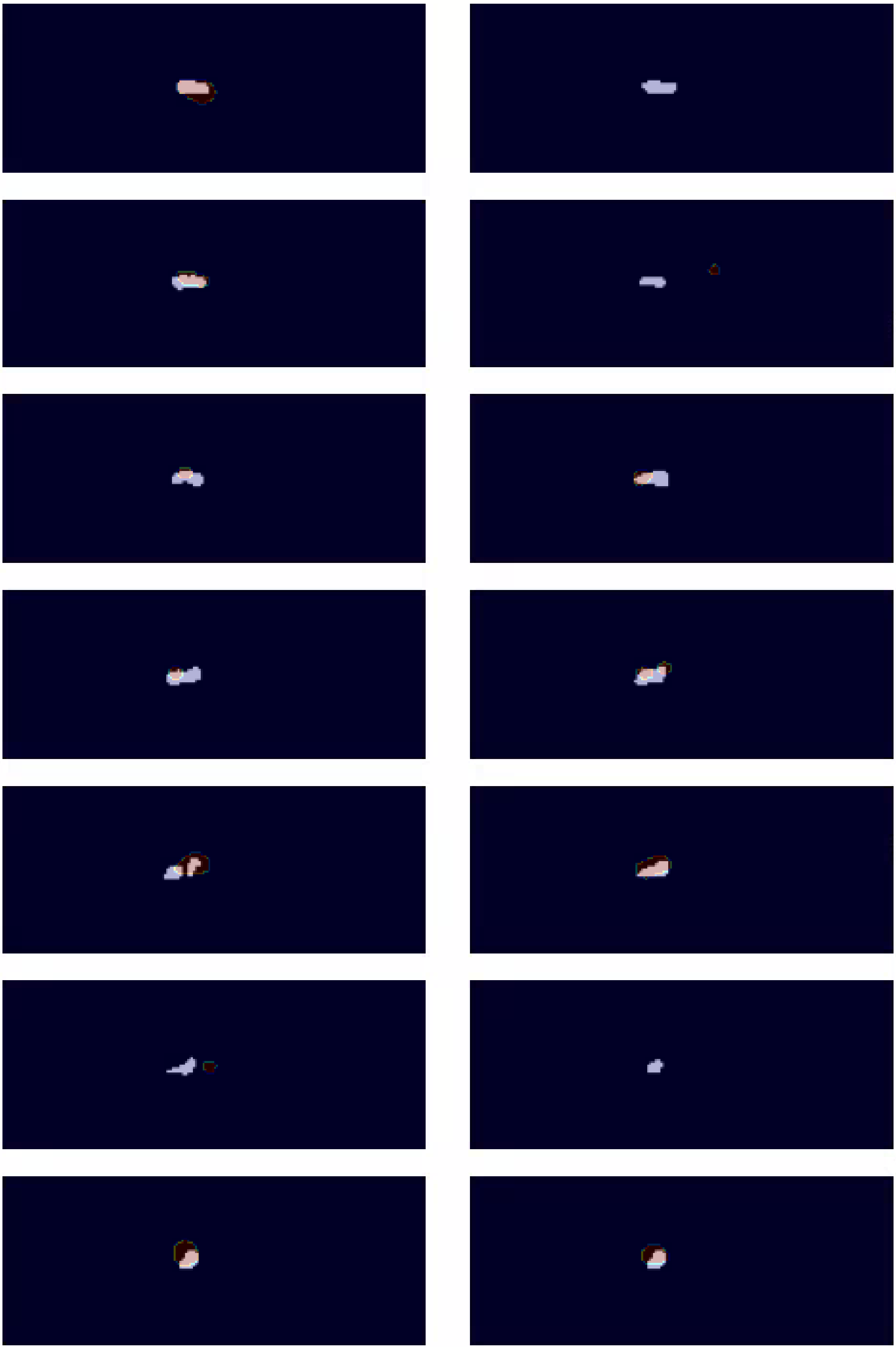
Fold nr.	Patient in evaluation set	Calculated DSCs
1	CR01	0.34
2	CR02	0.26
3	CR03	0.00
4	CR04	0.09
5	CR05	0.59
6	CR06	0.51
7	CR09	0.02
8	CR11	0.43
9	CR12	0.52
10	CR14	0.04
11	CR15	0.40
12	Phase1_10	0.66
13	Phase1_11	0.57
14	Phase1_19	0.27
15	Phase1_20	0.39
16	Phase1_21	0.70
17	Phase1_22	0.78
18	Phase1_23	0.00
19	Phase1_25	0.68
20	Phase1_26	0.68
21	Phase1_28	0.29
22	Phase1_29	0.56
23	Phase1_2	0.00
24	Phase1_30	0.61
25	Phase1_31	0.52
26	Phase1_33	0.26
27	Phase1_34	0.56
28	Phase1_3	0.52
29	Phase1_4	0.53
30	Phase1_5	0.52
31	Phase1_7	0.69
32	Phase1_9	0.57
33	Phase2_36	0.53
34	Phase2_37	0.17
35	Phase2_38	0.13
36	Phase2_40	0.33
37	Phase2_41	0.41
38	Phase2_42	0.56
39	Phase2_43	0.23
40	Phase2_44	0.54

41	Phase2_49	0.37
42	Phase2_51	0.65
43	Phase2_53	0.53
44	Phase2_54	0.51
45	Phase2_56	0.23
46	Phase2_58	0.30
47	Phase2_59	0.29
48	REMP_10	0.00
49	REMP_11	0.00
50	REMP_12	0.06
51	REMP_14	0.57
52	REMP_16	0.61
53	REMP_1	0.44
54	REMP_2	0.70
55	REMP_3	0.38
56	REMP_4	0.45
57	REMP_5	0.19
58	REMP_6	0.00
59	REMP_7	0.16
60	REMP_8	0.16
61	REMP_9	0.66
	Mean	**0.39**
	Median	**0.43**
	Min.	**0.00**
	Max.	**0.78**
	Std	**0.22**

Appendix E2 – Overlap of ground truth and predictions from the best-performing volume: foreground slices



Appendix E3 – Overlap of ground truth and predictions from an average-performing volume: foreground slices



Appendix F – DSCs of LOOCV folds: cropped images

Fold nr.	Patient in evaluation set	Calculated DSCs
1	CR01	0.34
2	CR02	0.29
3	CR03	0.14
4	CR04	0.12
5	CR05	0.54
6	CR06	0.62
7	CR09	0.05
8	CR11	0.41
9	CR12	0.63
10	CR14	0.05
11	CR15	0.21
12	Phase1_10	0.76
13	Phase1_11	0.58
14	Phase1_19	0.29
15	Phase1_20	0.35
16	Phase1_21	0.68
17	Phase1_22	0.71
18	Phase1_23	0.00
19	Phase1_25	0.39
20	Phase1_26	0.72
21	Phase1_28	0.35
22	Phase1_29	0.55
23	Phase1_2	0.00
24	Phase1_30	0.67
25	Phase1_31	0.52
26	Phase1_33	0.40
27	Phase1_34	0.62
28	Phase1_3	0.52
29	Phase1_4	0.45
30	Phase1_5	0.62
31	Phase1_7	0.73
32	Phase1_9	0.70
33	Phase2_36	0.55
34	Phase2_37	0.14
35	Phase2_38	0.01
36	Phase2_40	0.38
37	Phase2_41	0.33
38	Phase2_42	0.46
39	Phase2_43	0.25
40	Phase2_44	0.56

41	Phase2_49	0.44
42	Phase2_51	0.51
43	Phase2_53	0.50
44	Phase2_54	0.54
45	Phase2_56	0.23
46	Phase2_58	0.30
47	Phase2_59	0.29
48	REMP_10	0.10
49	REMP_11	0.03
50	REMP_12	0.26
51	REMP_14	0.69
52	REMP_16	0.57
53	REMP_1	0.47
54	REMP_2	0.58
55	REMP_3	0.52
56	REMP_4	0.43
57	REMP_5	0.24
58	REMP_6	0.00
59	REMP_7	0.20
60	REMP_8	0.22
61	REMP_9	0.70
	Mean	0.40
	Median	0.43
	Min.	0.00
	Max.	0.76
	Std	0.22

Appendix G – DSCs of LOOCV folds: whole volumes

		Calculated DSCs on whole volumes with a varying number of added background slices during training.		
Fold nr.	Patient in evaluation set	0 added slices	2 added slices	5 added slices
1	CR01	0.08	0.05	0.06
2	CR02	0.07	0.07	0.07
3	CR03	0.01	0.00	0.00
4	CR04	0.01	0.00	0.00
5	CR05	0.17	0.18	0.17
6	CR06	0.14	0.13	0.05
7	CR09	0.00	0.00	0.00
8	CR11	0.19	0.18	0.15
9	CR12	0.27	0.31	0.31
10	CR14	0.01	0.00	0.01
11	CR15	0.11	0.00	0.00
12	Phase1_10	0.25	0.16	0.14
13	Phase1_11	0.16	0.17	0.16
14	Phase1_19	0.05	0.06	0.04
15	Phase1_20	0.11	0.08	0.11
16	Phase1_21	0.25	0.23	0.19
17	Phase1_22	0.61	0.60	0.59
18	Phase1_23	0.00	0.00	0.00
19	Phase1_25	0.42	0.45	0.43
20	Phase1_26	0.22	0.21	0.21
21	Phase1_28	0.25	0.19	0.20
22	Phase1_29	0.20	0.17	0.16
23	Phase1_2	0.00	0.00	0.00
24	Phase1_30	0.37	0.38	0.21
25	Phase1_31	0.06	0.06	0.06
26	Phase1_33	0.11	0.03	0.02
27	Phase1_34	0.20	0.14	0.15
28	Phase1_3	0.14	0.17	0.11
29	Phase1_4	0.22	0.27	0.17
30	Phase1_5	0.27	0.24	0.24
31	Phase1_7	0.26	0.24	0.15
32	Phase1_9	0.34	0.33	0.35
33	Phase2_36	0.18	0.18	0.16
34	Phase2_37	0.12	0.09	0.00
35	Phase2_38	0.04	0.01	0.00
36	Phase2_40	0.12	0.09	0.09
37	Phase2_41	0.14	0.15	0.10
38	Phase2_42	0.44	0.32	0.22

39	Phase2_43	0.12	0.11	0.04
40	Phase2_44	0.30	0.22	0.20
41	Phase2_49	0.30	0.30	0.11
42	Phase2_51	0.49	0.46	0.25
43	Phase2_53	0.41	0.39	0.41
44	Phase2_54	0.36	0.33	0.32
45	Phase2_56	0.11	0.09	0.05
46	Phase2_58	0.15	0.14	0.08
47	Phase2_59	0.10	0.12	0.10
48	REMP_10	0.01	0.00	0.00
49	REMP_11	0.01	0.00	0.01
50	REMP_12	0.00	0.03	0.00
51	REMP_14	0.13	0.13	0.09
52	REMP_16	0.13	0.13	0.13
53	REMP_1	0.12	0.09	0.09
54	REMP_2	0.11	0.04	0.02
55	REMP_3	0.16	0.14	0.13
56	REMP_4	0.11	0.10	0.09
57	REMP_5	0.03	0.03	0.01
58	REMP_6	0.01	0.00	0.03
59	REMP_7	0.03	0.04	0.04
60	REMP_8	0.01	0.01	0.01
61	REMP_9	0.13	0.11	0.10
	Mean	0.16	0.15	0.12
	Median	0.13	0.13	0.10
	Min.	0.00	0.00	0.00
	Max.	0.61	0.60	0.59
	Std	0.14	0.13	0.12