

Recurrent Fully Convolutional Network for 3D Brain Tumor Segmentation

Floris van Hassel
STUDENT NUMBER: 1242117

THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE IN DATA SCIENCE & SOCIETY
DEPARTMENT OF COGNITIVE SCIENCE & ARTIFICIAL INTELLIGENCE
SCHOOL OF HUMANITIES AND DIGITAL SCIENCES
TILBURG UNIVERSITY

Thesis committee:

dr. Sharon Ong
dr. Giovanni Cassani

Tilburg University
School of Humanities and Digital Sciences
Department of Cognitive Science & Artificial Intelligence
Tilburg, The Netherlands
May 2020

Preface

I would like to thank my supervisor Dr. Sharon Ong for her guidance and support during the thesis meetings and Skype calls in these challenging times of COVID-19. I have encountered numerous difficulties during this research and Dr. Ong door was always open for advice. I would also like to thank my second reader Dr. Giovanni Cassani for his constructive feedback. Finally, I would also like to thank Stefano Sipeng Chen for providing a workplace and mentally supporting me during this challenging time.

Recurrent Fully Convolutional Network for 3D Brain Tumor Segmentation

Floris van Hassel

Brain tumor segmentation is a challenging task, with the advances in deep learning it has show much progress over the last decade. The models for brain tumor segmentation use 3D convolutions and these models show very good results, but are computationally very expensive. A solution is to analyse the data via a 2D Fully Convolutional Network, in 2D slice format, however the spatial relations between the 2D slices along the z-axis cannot be leveraged with this method. In this research we propose a RFCN which uses a 2D FCN and a RNN combined for a 3D brain tumor segmentation task. The RNN component is used to leverage inter-slice information to increase segmentation performance. The data used is the BraTS 2019 dataset. The results in this research have shown that the proposed RFCN architecture can leverage inter-slice information to increase the segmentation result compared to a 2D U-net (from 83.89% to 84.37%).

1. Introduction

Cancer diseases have been increasingly impacting people's lives, especially whenever they involve the brain. In fact, after a brain tumor diagnosis, the patients' survival rate beyond five years is 32.9% (National Cancer Institute, 2019). Brain tumors are divided in 2 groups High Grade Gliomas (HGG) and Low Grade Gliomas (LGG), HGG are the more malicious and faster growing type of tumor compared to LGG, and have a significantly lower survival rate than LGG (Schwartzbaum et al. 2006). In 2019, it is estimated that there were 23,820 new cases of brain and other nervous system cancer, accounting for 17,760 deaths in the US (National Cancer Institute, 2019). These statistics imply that when a patient receives treatment for a brain tumor there is still a very high chance the patient does not survive. This makes quality of life post-treatment very important. There is less reason to treat or operate if survival rate is low or if the patient is unable to enjoy life post surgery/treatment. Given the high risk of impacting brain functionality when removing tumors from the brain, patients are often kept awake during surgery. The patients' brain functionality is tested during surgery to assess whether removing certain parts of the brain is impacting its functionality. Hence, it is critical to accurately locate brain tumors.

Less or non-invasive treatments, where medical professionals do not have direct sighting of the tumor, can also benefit from accurately locating brain tumors. An example would be Gamma knife radiosurgery; the procedure uses hundreds of highly focused radiation beams to target tumors in the brain (without any surgical incision). Having algorithms that can properly locate brain tumors can be very beneficial for these kinds of treatments. Therefore, automating tumor localisation tasks can help in minimizing human error and accelerating diagnosis.

Neural networks have improved over the last decade, from classifying images to segmenting pixels in images (classifying every pixel in an image) (Noh, Hong, and

Han 2015). Recently, neural networks have shown promising potential in medical image analysis. Fully convolutional networks (FCN) were among the first convolutional neural networks to make an impact in the medical image segmentation field (Long, Shelhamer, and Darrell 2015). The U-net structure of the FCN greatly improved the results on segmentation tasks (Ronneberger, Fischer, and Brox 2015), yet the segmentation of brain tumors in brain MRI-scans is still challenging due to their irregular shapes, heterogeneous compositions and hard to define tumor boundaries (Akkus et al. 2015).

Medical image data is often in 3D, leading researchers to apply volumetric algorithms. However, this approach requires high amounts of computational power, memory and time (Lai 2015). In order to overcome these issues, the 3D data can be analysed as sequential slices of 2D images (Chen et al. 2016). The 3D segmented image is, then, generated by recombining the segmented 2D slices back into a 3D image. Exchanging the 3D convolutions with 2D convolutions has computational benefits, however the spatial relations between the 2D slices along the z-axis cannot be leveraged. Therefore, resulting in loss of information and less accurate segmentation (Chen et al. 2016).

Recurrent neural networks (RNN) are effective models to process sequential data (Keren and Schuller 2016). In video segmentation, RNNs have been successfully combined with CNNs to sequentially segment 2D video slices, leveraging the spatial relations among the 2D image slices (Siam et al. 2017) (Yao et al. 2019). Biomedical image segmentation can also exploit these properties of the RNN. (Chen et al. 2016) have combined a RNN with a 2D FCN to segment 3D biomedical images of fungi, showing state-of-the-art comparable performance for the architecture. On 3D cardiac segmentation, combining a RNN with 2D FCN have produced near state-of-the-art results, which suggests that inter-slice information is leveraged, resulting in better segmentation results (Poudel, Lamata, and Montana 2016). The Recurrent Fully Convolutional Network (RFCN), proposed by (Poudel, Lamata, and Montana 2016), shows that a RNN combined with a FCN can leverage inter-slice information for cardiac segmentation, meanwhile having clear computational benefits.

As above described, combining a RNN with 2D FCN have shown promising results on 3D segmentation tasks, relative to 3D architectures. Hence, leveraging inter-slices information for brain tumor segmentation is worth exploring.

To the best of my knowledge, little research has been done on implementing a RFCN architectures (RNN combined with a 2D FCN) on 3D brain tumor segmentation tasks. Motivated by the efficacy and efficiency of RFCN architectures, and inspired by the RFCN architectures proposed by (Chen et al. 2016), (Poudel, Lamata, and Montana 2016) and (Siam et al. 2017) a RFCN architecture is proposed to leverage inter-slice information. The aim for this research is to answer the following research questions:

To what extent can a 2D U-Net be improved by combining the U-Net with a Recurrent Neural Network (RNN) on a 3D Brain Tumor segmentation task?

In this research the FCN used will be a U-net. The U-Net is trained and optimised to predict a baseline segmentation score on the brain tumor segmentation task.

The model proposed in this research is going to exploit the ability to leverage inter-slice information via a RNN component. As shown in Figure 1, the proposed model takes 2 inputs: the first is t_0 , a 2D slice from the 3D MRI-scan; the second is a sequence of segmented masks of preceding 2D slices to t_0 of the 3D MRI-scan it is currently being processed (see Fig 1, part b1 & b2). The sequences in this model will be created by a sliding window that precede slice t_0 , and the sequences consist of segmentation masks created by the baseline U-net produces by this research (This process is schematically displayed in Figure 1, part a). The FCN-part in the proposed RFCN will attempt to extract intra-slice information from the 2D slice input. The RNN-part in the proposed

RFCN will attempt to extract inter-slice information from the sequence input. The information extracted by both parts of the RFCN than will be combined and processed in the model to produce a segmentation mask for slice t_0 , as schematically displayed in 1.

To assess whether the RNN component in the proposed RFCN can actually leverage inter-slice information, the predictions from the U-net will be used as baseline.

The RFCN architectures used by (Chen et al. 2016), (Poudel, Lamata, and Montana 2016) and (Siam et al. 2017) have placed the RNN component of the model at different place in the model. (Poudel, Lamata, and Montana 2016) placed the RNN component of the model between the contracting and the expanding path of the model as show in Figure 2. (Chen et al. 2016) and (Siam et al. 2017) placed the RNN component of the model near the end of the network. Both type of RFCN architectures have good results, therefore it is interesting to test whether the placement of the RNN component as an effect on the segmentation result for brain tumor segmentation.

Sub-question 1: What effect does the placement of the RNN component in the RFCN architecture have on the segmentation result? To assess whether the placement of the RNN component has influence on the results, 2 network architectures are proposed. The first architecture with the RNN component in the middle of the network, as seen in Fig 7, and the second architecture with the RNN component in the end of the network, as seen in Fig 8.

Sub-question 2: How well does the RFCN perform compared to the U-Net on segmenting LGG and HGG? Brain tumors consist of 2 types HGG and LGG, these 2 kinds tumors are structured differently in the data and might therefore be harder or easier to detect (Bakas et al. 2018). Additionally, 77% of the training data used in this research are HGG, the different structures and the imbalance in data might an effect of how well the models produces segmentation masks for HGG and LGG (Menze et al. 2014).

The main findings in this research is that adding a RNN component to a U-Net architecture increases the segmentation results produced by the network. The RFCN architecture seems to leverage inter-slice information to improve the segmentation results from 83.9% dice score for the U-Net to 84.4% Dice-score for the best performing RFCN. The results show an increase of approximately 0.6% for the RFCN architecture in the dice score. This seems like a slight increase, however the resolution used in this research is heavily scaled down from the original image, which makes it harder for the network to detect useful structures. The result does indicate that the method can be used to leverage inter-slice information.

This research is structured as follows: Section 2 contains an overview of related work in the domain image segmentation with additional focus on Recurrent fully convolutional models; Section 3 describes the data, prepossessing, methods and models used in this research; Section 4 contains the results, which will be discussed in Section 5; Section 6 closes with a conclusion.

2. Related Work

In this section related papers are going to presented, to give a comprehensive view about relevant segmentation architectures, meanwhile explicating the underling reasoning of this research.

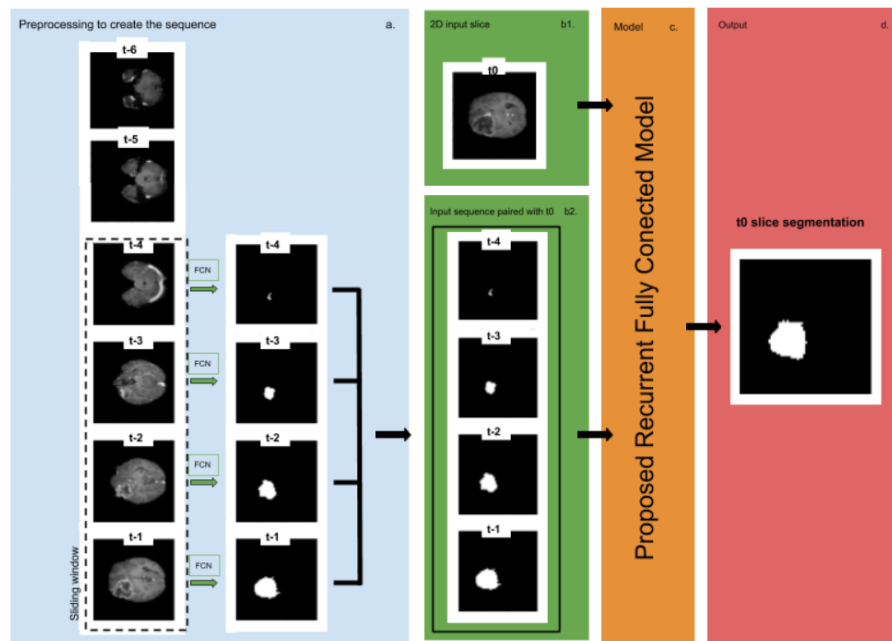


Figure 1

A schematic overview of the proposed RFCN. (part a) The U-net trained in this research is used to predict the segmentation masks for in the sequences that are going to be used by the proposed RFCN. A slicing window is used to look at the preceding slices before slice t0 (part b1) The 2D slice that the RFCN is currently going to process. (part b2) The segmentation masks of the sequence processed in part a. (part c) The proposed RFCN model takes 2 inputs from part b1 and b2 train. (part d) The output of the proposed RFCN model is a segmentation masks for 2D slice t0.

2.1 Brain tumors

Brain tumors have been classified in four different grades (I, II, III, IV) by The World Health Organization (Schwartzbaum et al. 2006). Grades I & II are Low Grade Gliomas (LGG) and grades III & IV are High Grade Gliomas (HGG), LGG are slow growing brain tumors. HGG are faster growing, more malignant tumors and they tend to more frequently develop around the nervous system (Smoll, Schaller, and Gautschi 2013). The removal of these brain tumors is a very difficult process and it starts with accurately locating the tumor. The localisation task is mostly done manually and very time-consuming (Dong et al. 2017). The accuracy of the segmentation is, therefore, highly dependent on the experience and capability of the medical professional examiner. Segmentation algorithms can obviate these human limitations, meanwhile increasing accuracy.

2.2 Neural Networks for Medical Image Segmentation

In the last decade, there has been much progress in the computer vision field. Of particular interest are the developments in segmentation algorithms, whose task is to assign pixels to specific object classes or structures in an image. A wide range of neural network architectures have been developed since the introduction of segmentation

algorithms for medical images e.g. Model Compression based Image Segmentation, Encoder Decoder based Image Segmentation, Attention based Image Segmentation, Adversarial Training based Image Segmentation & Recurrent Neural Network-based Models (Taghanaki et al. 2019). In this research, the focus is on Recurrent Neural Network-based models. These models are a combination of a FCN with a RNN component, making them particularly useful due to their minor computational demand, relative to volumetric models, meanwhile showing near state-of-the-art performances on 3D segmentation tasks (Chen et al. 2016) (Poudel, Lamata, and Montana 2016).

2.3 Fully Convolutional Networks

Before the introduction of the FCN, the most common models used for segmentation were patchwise based models, but those models lack the efficiency of fully convolutional training (Ganin and Lempitsky 2014), (Ning et al. 2005). Patchwise based methods needed to make use of pre- and post-processing complications (Long, Shelhamer, and Darrell 2015). Fully convolutional networks (FCN) were one of the first networks to make an impact in the image segmentation field (Long, Shelhamer, and Darrell 2015). (Long, Shelhamer, and Darrell 2015) introduced the first FCN that could be trained end-to-end for pixelwise prediction, they defined a skip architecture to take advantage of deep feature hierarchical encoded locations and semantics in a nonlinear local-to-global pyramid (Long, Shelhamer, and Darrell 2015). The FCN architecture designed by (Long, Shelhamer, and Darrell 2015) greatly improved state-of-the-art at the time, while simultaneously simplifying and speeding up training times on segmentation tasks.

However, the introduction of U-net structured networks has greatly changed the field of image segmentation (Ronneberger, Fischer, and Brox 2015). Due to the effectiveness of the U-Net, its architecture has been adopted in many other segmentation models (Taghanaki et al. 2019). The U-net's architecture consists of two paths connected by skip connections: a contracting path and an expanding path, which are connected by skip connections (see Figure 6). The contracting path captures contextual information as image structures/patterns, to improve segmentation performance (Ronneberger, Fischer, and Brox 2015). The expanding path is used for locating the object of interest in a pixel wise manner (Ronneberger, Fischer, and Brox 2015). This is followed by 1x1 convolutional layers, thereby producing a class label for every pixel in the output image (Ronneberger, Fischer, and Brox 2015). For medical image segmentation U-Net architectures have been used in a variety of segmentation tasks, and have shown its effectiveness (Taghanaki et al. 2019). The U-net is a popular and an established architecture for segmentation tasks, therefore it is used as a baseline model for in this research.

2.4 3D segmentation models

Traditional approaches to 3D images segmentation include segmenting 2D slices of the image and only then inferring the 3D segmentation result. Models based on this approach consider only the context within the 2D slices and ignore the 3D relations, thereby not fully exploiting the data (Milletari, Navab, and Ahmadi 2016). (Çiçek et al. 2016) and (Milletari, Navab, and Ahmadi 2016) tried to overcome this issue by using 3D convolutions to extrapolate the global context. The V-Net algorithm introduced by (Milletari, Navab, and Ahmadi 2016) and the 3D U-Net proposed by (Çiçek et al. 2016) have shown good results on respectively the MRI prostate volumes and highly variable structures of the Xenopus kidney. Nonetheless, while the V-Net and the 3D U-net show

good results on their respective segmentation tasks, they are very computationally demanding and have high memory usage (Milletari, Navab, and Ahmadi 2016), (Çiçek et al. 2016). For Brain tumor Segmentation, 3D models are state-of-the-art, the winning architecture is using an auto-encoder with 3D convolutions (Myronenko 2018). Models with 3D architectures generally have good performances, but the trade-off is almost always a high demand of computational power and memory usage (Lai 2015).

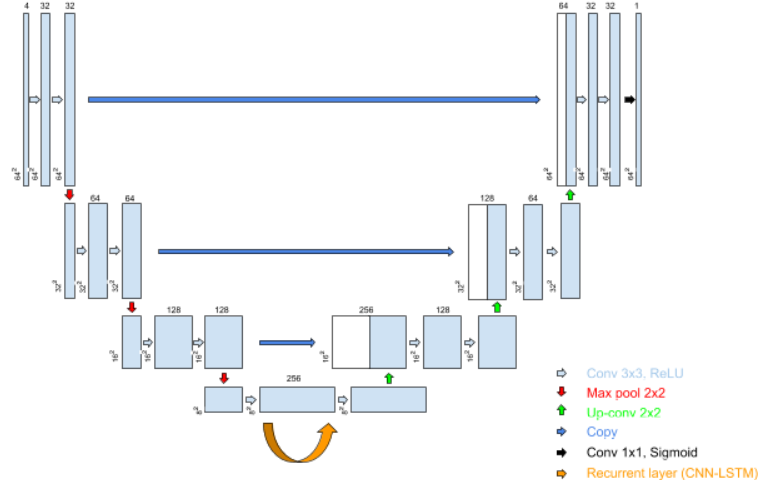
2.5 Combining FCN with RNN

To overcome the high demand of computational power and memory usage, 2D FCNs are combined with an RNN component. RNNs are very effective models to process sequential data (Keren and Schuller 2016). The RNN component is added to exploit the 3D spatial relations between the 2D image slices. In image analysis, RNNs can be used to process the 3D contexts along the z-axes for each 2D slice (Xingjian et al. 2015). (Chen et al. 2016) was among the first to propose a deep learning framework for 3D image segmentation, leveraging inter-slice correlations between 2D slices by combining a FCN combined with a RNN. In this architecture, the FCN component is for 2D feature extraction, efficiently compressing the information from the 2D slices, the RNN component then is used to extract hierarchical features from the 3D contexts (Chen et al. 2016). When met with ambiguous cases, the FCN-RNN model has shown better results than 2D FCNs, and is competitive with state-of-the-art architectures (Chen et al. 2016).

(Siam et al. 2017) embedded a FCN into a 2D RNN for semantic segmentation of online videos. This architecture receives a sequence of consecutive video frames (comparable to 2D slices) and produces a segmentation output for the last video frame. The FCN part of the network is used to extract information from the 2D video frames. The RNN part is used to preserve the spatial relations between the video frames, thereby carrying information forward to the video frame the network is currently segmenting (Siam et al. 2017). This method improved its results compared to a normal 2D FCN, which provides evidence of the effectiveness of implementing RNNs for 3D segmentation tasks (Siam et al. 2017).

(Poudel, Lamata, and Montana 2016) combined a FCN (U-Net) with a RNN for Cardiac segmentation, producing state-of-the-art results. In this architecture, the RNN module was implemented between the expanding and contracting part in the U-net style network, as displayed in Fig 2. The expanding path captures contextual information and compresses those in features maps in the middle of the network and before the contracting part is used for precise localization and RNN component is used to leverage inter-slice information (Poudel, Lamata, and Montana 2016). (Poudel, Lamata, and Montana 2016) further validates leveraging inter-slice information via RFCN-like networks approaches for 3D image segmentation tasks.

The research produced by (Chen et al. 2016), (Siam et al. 2017) and (Poudel, Lamata, and Montana 2016) an RNN component is added to FCNs, and have shown an increase in segmentation results. In the case of (Poudel, Lamata, and Montana 2016) and (Chen et al. 2016) the architectures are competitive with state-of-the-art segmentation models, which mostly used 3D based architectures, while using significantly less computational resources (Taghanaki et al. 2019). Therefore, it is interesting to investigate whether implementing an RNN component in a U-Net will improve the segmentation results for a brain tumor segmentation task.

**Figure 2**

RFCN architecture, this figure was originally created for this thesis and inspired from Figure 2 in the paper produced by (Poudel, Lamata, and Montana 2016)

3. Experimental Setup

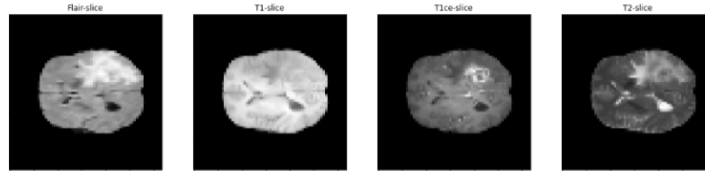
3.1 Data

The dataset is the Multimodal Brain Tumor Segmentation Challenge 2019 (BraTS 2019) dataset. This dataset consists of fMRI images of brain tumors. It includes 259 pre-operative multimodal MRI scans of glioblastoma (GBM/HGG) and 76 lower grade glioma (LGG) with pathologically confirmed diagnosis and available survival data. The data has been labeled with ground truths by expert board-certified neuroradiologists (Bakas et al. 2017).

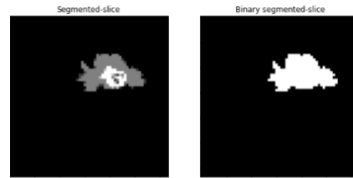
The MRI scans are formatted as NIfTI files (.nii.gz) and consist out of five 3D images containing: native (T1), post-contrast T1-weighted (T1Gd), T2-weighted (T2), T2 Fluid Attenuated Inversion Recovery (T2-FLAIR) volumes and one image with the brain tumor segmented. All the imaging datasets have been segmented manually, by one to four raters, following the same annotation protocol, and their annotations were approved by experienced neuro-radiologists. Annotations comprise the GD-enhancing tumor (ET — label 4), the peritumoral edema (ED — label 2), and the necrotic and non-enhancing tumor core (NCR/NET — label 1) (Menze et al. 2014). The provided data is distributed after pre-processing, i.e. co-registered to the same anatomical template, skull-stripped and interpolated to the same resolution (1 mm³) (Menze et al. 2014).

3.1.1 Preprocessing.

The data from the BraTS dataset are 3D images with (240x240x155) resolution. Due to computational limitations, the data was scaled down to 64x64x41. The 2 tumor types HGG and LGG are combined into 1 dataset, because the network doesn't know if the tumor is a HGG or LGG if it receives an input. Then the data is shuffled and split into: Train (70%), validation (15%) and test (15%).

**Figure 3**

The 4 different type of MRI scans. The images are generated by using BRATS 2019 data (Menze et al. 2014).

**Figure 4**

The segmentation masks provided by the dataset, and the binary mask used in this research. The images are generated by using BRATS 2019 data (Menze et al. 2014).

The data is then normalized by dividing all values by the highest value in the 3D image and converted to a 2D slice format: every 3D image was converted to 41 2D slices of 64x64. Additionally, every 2D slice has 4 channels, each for each of the 4 types of MRI scans (T1, T1Gd, T2 & T2-FLAIR), resulting in 41 4x64x64 2D slices per MRI scan. Before changing the data into slices the data was normalized by dividing all values by the highest value in the 3D image. The same conversion from 3D image to 2D sliced images was applied to the segmentation masks, resulting in a 1x64x64 segmentation mask. Moreover, the segmentation masks were converted to a binary format: 0 for any pixel part of the background and 1 for any pixel belonging to a tumor, as seen in figure 4.

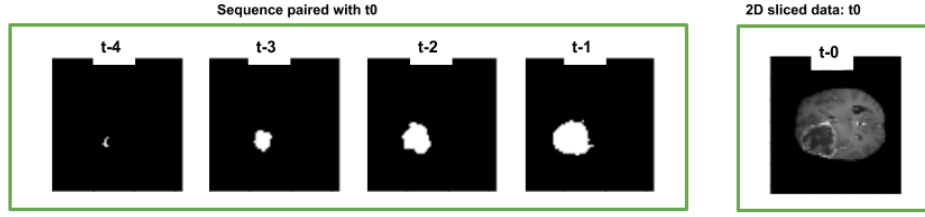
3.1.2 Preprocessing sequences.

Every 2D sliced image in the dataset has 4 shadows, if the sequences were to include the original 2D sliced image than the data would be multiplied by the length of the sequence. Hence, due to computational and memory limitations the sequences used in this research are limited to only a number of segmented masks, which only consist out of 1 2D sliced image without shadows (4 times less memory intensive as the original 2D sliced image). The segmented mask added in this sequence can be seen as a compressed representation of the original 2D slice.

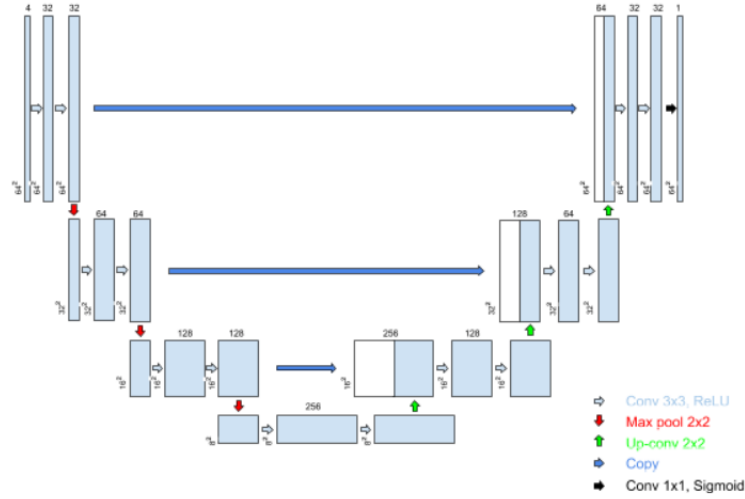
The RNN parts of the proposed RFCN models take as inputs a 2D slice t_0 and a sequence of t_0 - n . The sequence of t_0 - n consists of 2D segmentation masks of n slices that preceded slice t , as show in Figure 5. To indicate the first slice of every 3D MRI-scan, the sequence was initialized with an array of 64x64 with only the values 2. The empty slices in this sequence were padded with arrays of 64x64 with the value 0.

3.2 Models

3.2.1 Baseline: U-net.

**Figure 5**

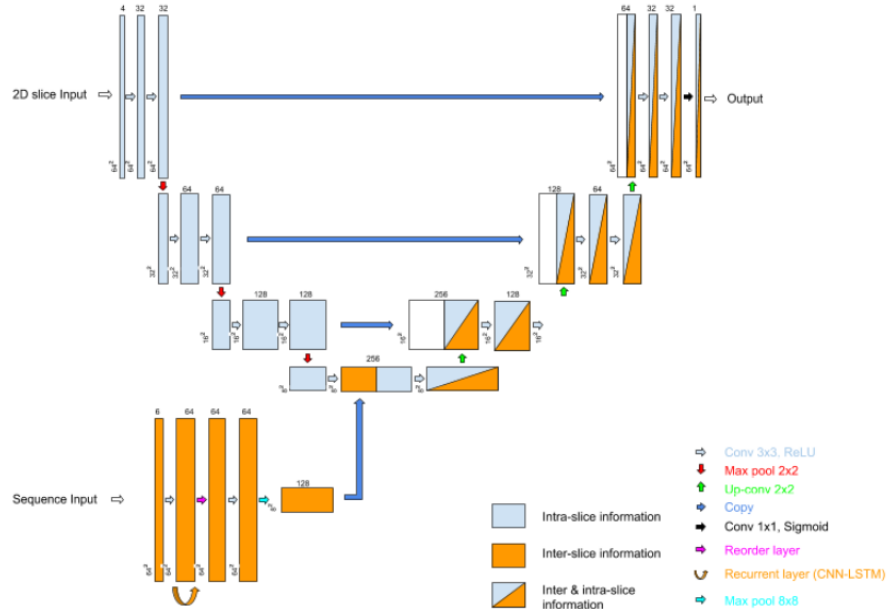
The 2 inputs for the RFCN, the sequence of t_0 - n paired with this slice ($n = 4$ in this sequence) and t_0 the 2D slice (with 4 shadows) that is currently being segmented. The images are generated by using BRATS 2019 data (Menze et al. 2014).

**Figure 6**

An illustration of the U-Net architecture, this figure was originally created for this thesis and inspired from Figure 1 in the paper produced by (Ronneberger, Fischer, and Brox 2015)

The baseline model in this research is a U-Net, which is constructed out of two parts, a contracting path (left side) and an expanding path (right side) connected by skip connections, as displayed in Figure 6. The contracting part consists of blocks: two 3x3 convolutions with stride 1, each followed by a rectified linear unit (ReLU); 2x2 max pooling operation with stride 2 for down-sampling. In the first block of the U-net, 32x64x64 feature maps are extracted out the 4x64x64 input. After each down-sampling step, the number of feature channels is doubled and the resolution of these channels is halved. The process is repeated for 3 blocks. This results in an output of 256x8x8 feature maps in the middle of the network.

The expanding path consists of up-sampling blocks, the exact opposite of the contracting path happens in the expanding path. Each block in the expanding path consists of a 2x2 up-convolution operation with stride 2, concatenated with the corresponding features map outputs from the contracting path, followed by two 3x3 convolutions with stride 1, each followed by a rectified linear unit (ReLU). This process is repeated 3 times.

**Figure 7**

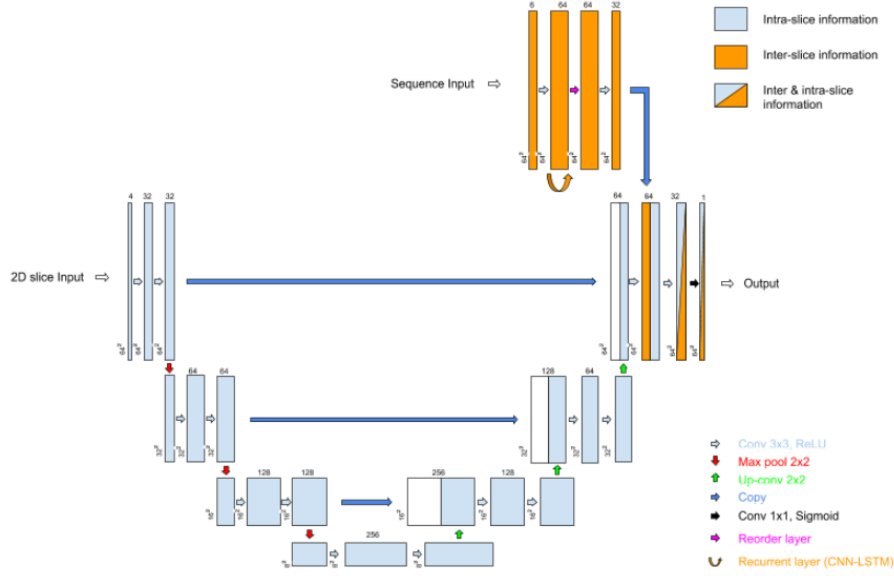
Proposed RFCN1 architecture, the network has 2 inputs, a 2D MRI slice and a sequence paired with this slice of segmentation masks of preceding slices.

The expanding path starts with the 256x8x8 features maps, produced by the contracting path, after each block the feature maps are halved and the resolution of the feature maps is doubled, resulting in a 32x64x64 output. The final layer is a 1x1 convolutional layer, followed by an sigmoid activation, assigning a probability to each pixel, thereby creating a segmentation mask.

3.2.2 Recurrent Fully Convolutional Network 1.

The first RFCN architecture proposed is RFCN-1. RFCN-1 constructed out of three parts: a contracting path (left side), an expanding path (right side) and a RNN component. The RNN component is added between the contracting and expanding path, as proposed and successfully implemented for multi-slice cardiac segmentation (Poudel, Lamata, and Montana 2016). The contracting and expanding path the RFCN network proposed in this research have the same architectural layers as the U-net proposed in this research, with only an added RNN component, as visualized in Figure 7.

In an attempt to capture spatial change between slices, a sequence of a number of slices is added in the RNN component, where t_0 is the slice that is currently being segmented by the network and n is the number of preceding slices before t_0 that are included in the sequence, as schematically displayed in Fig 1. The sequence consists of segmentation masks, which are produced by the baseline U-net, of previous slices preceding slice t_0 . The sequences are processed by a Convolutional-LSTM and the CNN layer, and the features maps produced by this process are then added to the feature maps produced by the contracting path of the RFCN, as shown in Figure 7. This network is tested with sequences length of 2, 4 and 6. The amount of feature maps

**Figure 8**

Proposed RFCN2 architecture, the network has 2 inputs, a 2D MRI slice and a sequence paired with this slice of segmentation masks of preceding slices.

produced by the RNN component is equivalent to the amount of feature maps produced by the U-Net part of the model, which is 128 feature maps in this architecture.

The RNN component starts with an input layer, which is shaped according to the length of the sequence that is inputted, if the sequence length is 6 the shape becomes (None, 6, 64, 64, 1), if the sequence length is 4 it is (None, 4, 64, 64, 1). This input is then processed with a 3x3 2D Convolutional LSTM layer, followed by a ReLU unit. In order to feed this information in the RFCN architecture, the output dimensions of this layer are reordered from (samples, output row, output column, filters) to (samples, filters, output row, output columns). This is followed by 3x3 convolutions with stride 1, followed by ReLU unit. To match the output shape of the features maps produced by the contracting path of the network, a 8x8 max pooling operation with stride 1 is applied to the feature maps produced by the Convolutional LSTM. Finally, the features maps produced by the RNN component and the output from the contracting path are concatenated, and input into the network at the beginning of the expanding path. The architecture of this RNN component is displayed in table 1.

3.2.3 Recurrent Fully Convolutional Network 2.

The second RFCN proposed is RFCN-2. RFCN-2 consists of the same three parts as RFCN-1, a contracting path (left side), an expanding path (right side) and a RNN component. In this model the RNN component is added one layer before the final segmentation output, as inspired by (Siam et al. 2017), as seen in Figure 8. The objective is to create an architecture that better preserves the spatial information in the sequences, relative to the architecture in RFCN-1. The same sequences that were added in RFCN-1 were used in this architecture. The architecture recurrent component can be found in table 2. This network is tested with sequences length of 2, 4 and 6. The amount of

Table 1

Architecture of the global feature component in the Recurrent Fully Convolutional Network 1 (with sequence length 6)

Name	Input	Output	Parameters
Input	(None, 6, 64, 64, 1)	(None, 6, 64, 64, 1)	0
ConvLSTM2D	(None, 6, 64, 64, 1)	(None, 64, 64, 64)	150016
Permute	(None, 64, 64, 64)	(None, 64, 64, 64)	0
Conv	(None, 64, 64, 64)	(None, 128, 64, 64)	73856
Pool	(None, 128, 8, 8)	(None, 128, 8, 8)	0
Concatenate	(None, 128, 8, 8) + (None, 128, 8, 8)	(None, 256, 8, 8)	0

Table 2

Architecture of the global feature component in the Recurrent Fully Convolutional Network 2 (with sequence length 4)

Name	Input	Output	Parameters
Input	(None, 4, 64, 64, 1)	(None, 4, 64, 64, 1)	0
ConvLSTM2D	(None, 4, 64, 64, 1)	(None, 64, 64, 64)	150016
Permute	(None, 64, 64, 64)	(None, 64, 64, 64)	0
Conv	(None, 64, 64, 64)	(None, 32, 64, 64)	18464
Concatenate	(None, 32, 64, 64) + (None, 32, 64, 64)	(None, 64, 64, 64)	0

feature maps produced by the RNN component is 32, which is equivalent to the amount of features maps produced by the U-Net part of this model. The difference in architecture between the 2 RNN component can be seen in Figures 1 and 2.

3.2.4 Loss function.

The loss function used is pixel-wise binary cross-entropy (CE) summed with (1 - dice (DC)) and it can be denoted as:

$$CE(p, \hat{p}) = -(p \log(\hat{p}) + (1 - p) \log(1 - \hat{p}))$$

$$DC = 1 - \frac{2|X \cap Y|}{|X| + |Y|}$$

$$Loss = CE(p, \hat{p}) + DC$$

3.2.5 Implementation.

All the models use Adam as the optimizer. The learning rate is initialised at 0.001 and it is halved whenever the loss does not improve for 5 epochs. The activation units

use ReLU, in all the convolutional layers, and Sigmoid, in the output layer. The batch size used in every model was 64, due to computational limitations: increasing batch size to 128 causes memory issues for the RFCN, thus, for sake of comparability, all batch sizes were kept at 64. Early stopping is applied if the validation loss has not improved for more than 30 epochs. The epoch with the lowest validation loss was selected to predict on test. All the experiments were conducted with a GPU on Google Collaboratory (Bisong 2019).

3.2.6 Evaluation metrics.

The evaluation metrics used are Dice coefficient, sensitivity and specificity. All the evaluations have been performed on the resulting 3D segmentation masks (by recombining all 2D segmentation masks of the sliced data), for every MRI-scan in the test set.

Dice evaluates the overlap between two samples (Taha and Hanbury 2015). Dice scores range from 0 to 1, where 0 means no overlap (no tumor pixels have been classified as tumor) and 1 means complete overlap (every tumor pixel is classified as tumor).

$$Dice = \frac{2TP}{2TP + FP + FN} = \frac{2|X \cap Y|}{|X| + |Y|}$$

Sensitivity evaluates the amount of pixels that have been correctly classified as a tumor out of all the pixel that are classified as tumor (Taha and Hanbury 2015). Sensitivity ranges from 0 to 1, the higher the score the more like it is that when a pixel is classified as tumor that is actually is a tumor. Sensitivity is calculated as:

$$Sensitivity = \frac{TP}{TP + FN}$$

Specificity evaluates the amount of pixels that have been correctly predicted as non-tumor pixels out of all the pixels that were classified as non-tumor (Taha and Hanbury 2015). Specificity ranges from 0 to 1, the higher the score, the less likely a pixel is wrongly classified as belonging to a tumor. Specificity is calculated as:

$$Specificity = \frac{TN}{TN + FP}$$

3.2.7 Programming language & libraries. The coding in this research was done in Python (version 3.6.7). The libraries used were Tensorflow (2.2.0), Keras (2.3.0), tqdm (4.41.1), Numpy (1.18.5), Pandas (1.0.5), SciPy (1.4.1). Nibabel (3.0.2) was used to open the MRI-scans from the BraTS dataset and Matplotlib (3.2.2) was used for visualizing images and results.

4. Results

The models were evaluated with the metrics described in Section 3.2.6. The RFCN models are trained and tested with the sequence length (SL) 2, 4 and 6 as described in Section 3. All the results are reported in Table 3.

Table 3

Best performing models on 3D MRI-scans, with SL for the length of the sequence added to the RFCN. In bold the best performing configurations of RFCN-1 and RFCN-2.

Model	SL	Dice	Sensitivity	Specificity
Baseline: U-net	-	0.8389	0.8284	0.9988
RFCN-1	2	0.8431	0.8354	0.9988
	4	0.8419	0.8280	0.9989
	6	0.8396	0.8187	0.9989
RFCN-2	2	0.8411	0.8248	0.9989
	4	0.8414	0.8322	0.9989
	6	0.8437	0.8216	0.9990

Table 4

difference in performance in HGG and LGG, , with SL for the length of the sequence added to the RFCN

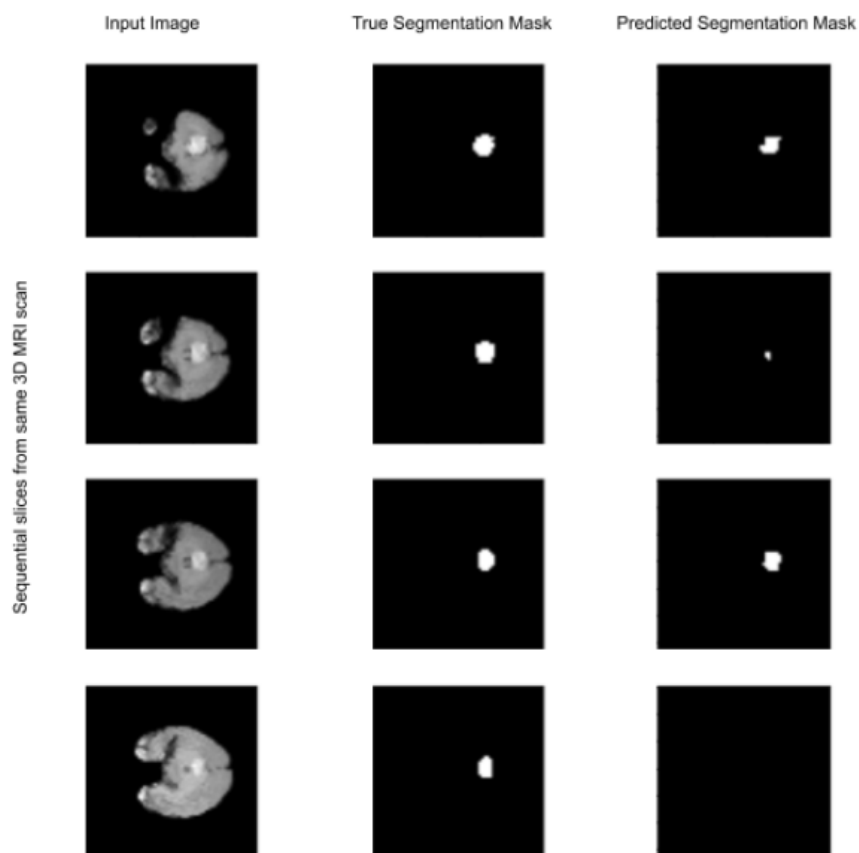
Model	SL	HGG			LGG		
		Dice	Sensitivity	Specificity	Dice	Sensitivity	Specificity
Baseline: U-net	-	0.8611	0.8519	0.9989	0.7870	0.7735	0.9987
RFCN-1	2	0.8649	0.8564	0.9989	0.7923	0.7866	0.9985
	4	0.8612	0.8479	0.9990	0.7968	0.7815	0.9986
	6	0.8591	0.8351	0.9999	0.7939	0.7804	0.9986
RFCN-2	2	0.8584	0.8404	0.9990	0.8001	0.7889	0.9986
	4	0.8574	0.8560	0.9989	0.8038	0.7766	0.9988
	6	0.8637	0.8417	0.9991	0.7971	0.7748	0.9988

The data consists of 259 HGG and 76 LGG. There is considerably less training data on LGG than for HGG, therefore the difference in results for HGG and LGG is also shown in Table 4.

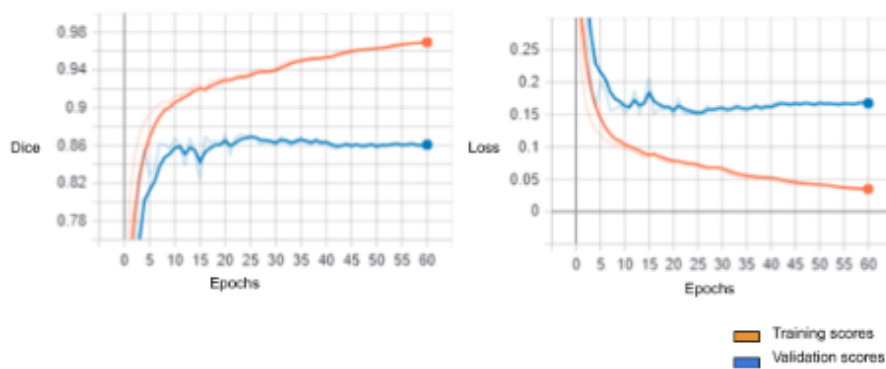
4.1 U-Net results

The U-Net trained for 60 epochs before it was early stopped because the validation loss didn't improve (See Figure 10). Epoch 35 was selected, since that epoch had the lowest validation loss (0.1487), and used for predictions on the test set. The validation dice score (on 2D validation data) for this network is 87.73%. The baseline on the test set produced by the U-Net is a 83.89% dice score with a sensitivity of 82.84% and a specificity of 99.88%.

The segmentation results produced by the U-Net are displayed in Figure 9. The first column consists of the 2D input slices, sequentially presented to the U-Net (from the bottom of the 3D MRI-scan to the top). The second column shows the true segmentation masks and the predicted segmentation masks by the U-net, the dice score for this observation is 55.39%.

**Figure 9**

Results produced by the U-Net. The first column contains 2D image slices from 1 3D MRI that were used as an input for the U-Net, presented sequentially. The second column contains the true segmentation masks and the third column contains the predicted segmentation masks. The images are generated by using BRATS 2019 data ([Menze et al. 2014](#)).

**Figure 10**

U-Net: Train Dice-score and Loss

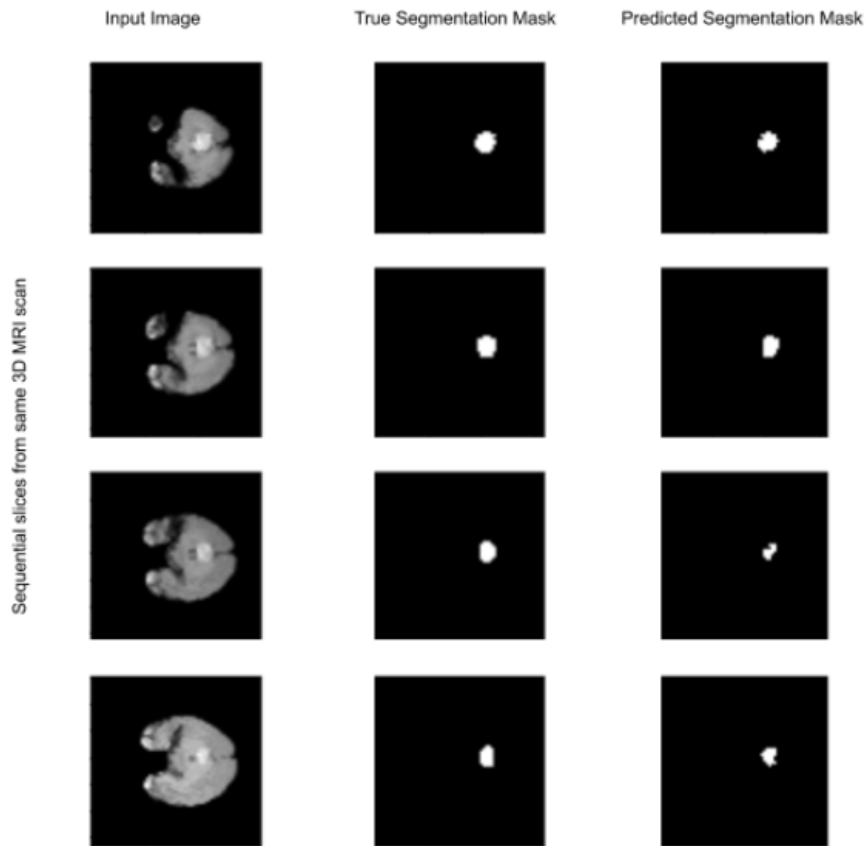


Figure 11

Results produced by the RFCN-1. The first column contains 2D image slices from 1 3D MRI that were used as an input for the RFCN-1, presented sequentially. The second column contains the true segmentation masks and the third column contains the predicted segmentation masks. The images are generated by using BRATS 2019 data (Menze et al. 2014).

4.2 RFCN-1 results

RFCN-1 with sequence length 2 has shown the best results for RFCN-1 and is described in these results. The network was trained for 65 epochs before it was early stopped because the validation loss didn't improve (See Figure 12). Epoch 45 was selected, since that epoch had the lowest validation loss (0.1469), and used for predictions on the test set. The validation dice score (on 2D validation data) for this network is 84.84%. The best performing RFCN-1 (with sequence length 2), outperforms the baseline on test with a 84.31% dice score, 83.54% sensitivity and a specificity of 99.88%.

The segmentation results produced by RFCN-1 are displayed in Figure 9. The first column consists of the 2D input slices, sequentially presented to the RFCN-1 (from the bottom of the 3D MRI-scan to the top). The second column shows the true segmentation masks and the predicted segmentation masks by the RFCN-1, the dice score for this observation is 64.81%.

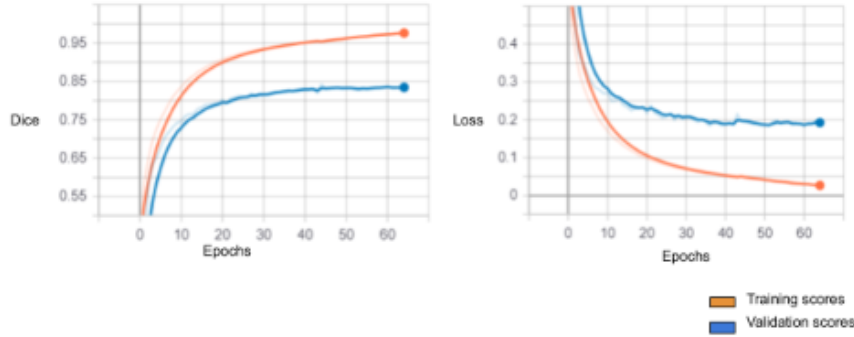


Figure 12
RFCN1: train Dice-score and Loss

4.3 RFCN-2 results

RFCN2 with sequence length 6 has shown the best results for RFCN2 and is described in these results. The network was trained for 56 epochs before it was early stopped because the validation loss didn't improve (See Figure 14). Epoch 30 was selected, since that epoch had the lowest validation loss (0.1440), and used for predictions on the test set. The validation dice score (on 2D validation data) for this network is 84.47%. The best performing RFCN-2 (with sequence length 6), outperforms the baseline on test with a 84.37% dice score with specificity at 99.90%, however sensitivity does not improve and is 82.16%.

The segmentation results produced by RFCN-2 are displayed in Figure 9. The first column consists of the 2D input slices, sequentially presented to the RFCN-2 (from the bottom of the 3D MRI-scan to the top). The second column shows the true segmentation masks and the predicted segmentation masks by the RFCN-2, the dice score for this observation is 63.86%.

4.4 Interpretation of the results

From the results in Table 3, the following interpretations can be made:

First, the proposed RFCN architectures reached higher dice scores on the segmentation task compared to the baseline U-Net, even if it is in some cases only by a small margin. In the case of RFCN-1 paired with a preceding sequence of length 6, the improvement in Dice is only 0.07%, but nevertheless all the RFCN architectures proposed in this research outperform the baseline on the dice score. The RFCN-2 architectures produce overall better results than the RFCN-1 architecture with reaching a dice score of 84.37% on the architecture combined with a sequence of 6 preceding slices.

Second, increasing the sequence input length for RFCN-1 seem to make the information added via the RNN component of the RFCN less useful for the final segmentation, lowering the dice score from 84.31% to 84.19% to eventually 83.96% for a sequence length of 6. In contrast, sequence length has the opposite effect on the dice score for RFCN-2, the longer the sequence length the more useful information the RNN component seems to transfer to the network. Increasing the dice score from 84.11% to 84.14% to finally 84.37% for sequence length 6.

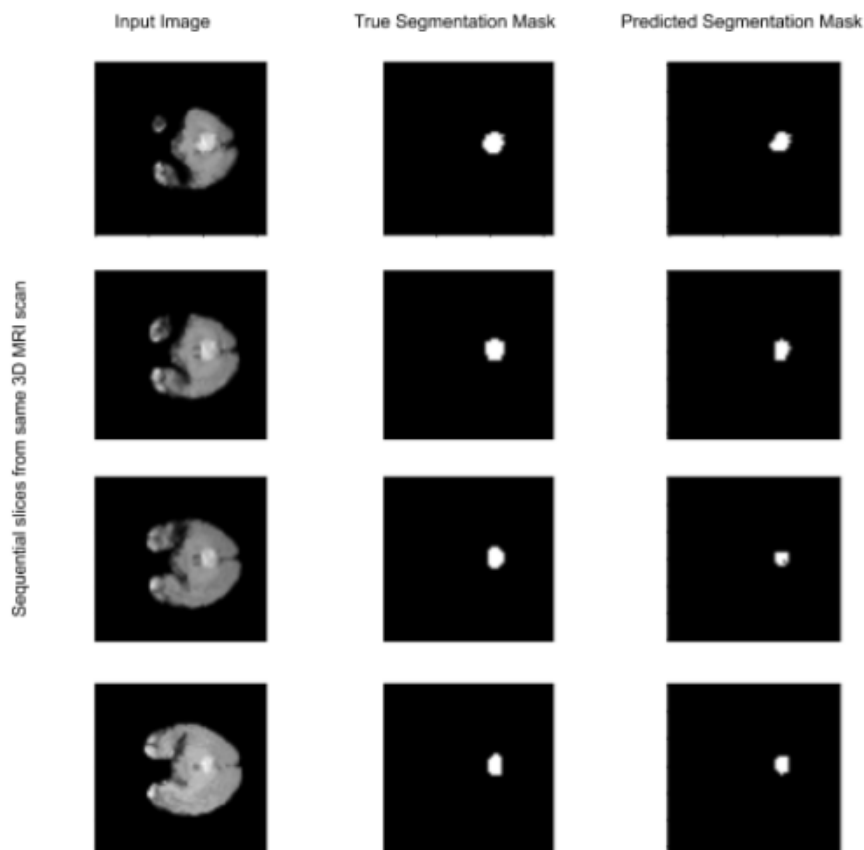


Figure 13
Results produced by the RFCN-2. The first column contains 2D image slices from 1 3D MRI that were used as an input for the RFCN-2, presented sequentially. The second column contains the true segmentation masks and the third column contains the predicted segmentation masks. The images are generated by using BRATS 2019 data (Menze et al. 2014).

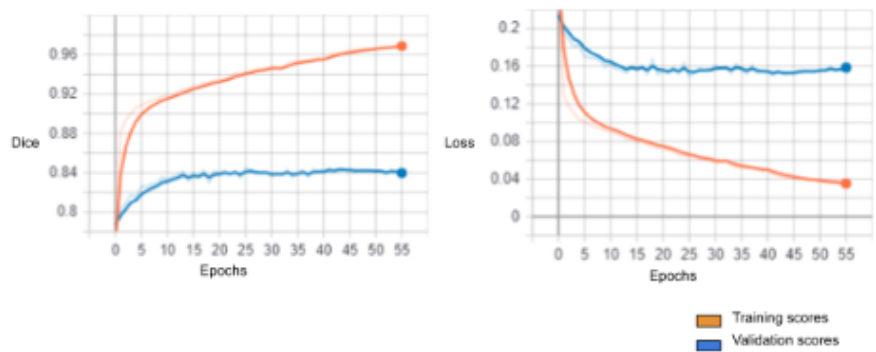


Figure 14
RFCN2: train Dice-score and Loss

Third, all the models tested in this research, reach a specificity of approximately 99.9%, which indicates that the network doesn't make many false positive predictions, while upholding predictive value with a dice score around the 84%. If the network predicted all no tumor, than the specificity would be 100%, but the dice and sensitivity would be 0%, thus only in combination with a relevant dice and sensitivity score, the result on specificity has any meaning. Thus, whenever a pixel is classified as non-tumor there is a very high likelihood of it not containing a tumor, which is very nice for patients (removing healthy brain cells would be unacceptable).

Finally, the sensitivity scores of all the models seem to vary around the 83%, this indicates that false negative predictions on pixels are not predicted very often (but do occur), while maintaining predictive values with the dice score at around the 84% and specificity around 99%. If the network would only predict tumor than sensitivity would be 100%, but specificity would be 0% and the dice score would drop dramatically to below 3%, due to the imbalance in classes (approximately 2% tumor pixels and approximately 98% pixels). Hence, having a fairly high sensitivity score, with specificity at near 100% is favorable for patients, it is better to have a few tumor cells not removed from the brain than to have healthy brain cells removed from the brain. The immune system can destroy malicious cells from the brain, but the brain can not produce new brain cells to replace ones that are removed.

The results in Table 4 show the results on HGG and LGG, these results indicate that: all three models perform worst on segmenting LGG compared to HGG, with an average difference of approximately 9% on dice and sensitivity scores. The specificity scores are around approximately 99.9% on both LGG and HGG, which indicates that the network doesn't make many false positive predictions.

Additionally, both RFCN architectures seem to perform better on LGG compared to the U-Net. The U-net performs the segmentation of LGG at 78.70% dice, and all the RFCN architectures outperform the U-Net on this task, ranging from lowest dice score 79.23% (for the RFCN architectures) to the highest dice score 80.38%. For segmenting HGG the performance of all 3 models seem to follow the same trend as the results from Table 3.

5. Discussion

The goal of this research was to assess whether a RNN component in an RFCN can leverage inter-slice information to improve segmentation results on a brain tumor segmentation task. The U-Net architecture was selected as a baseline for this research, and the proposed RFCN-1 and RFCN-2 architectures are tested on the brain tumor MRI data. All models are evaluated with dice, sensitivity and specificity. The U-Net performed the segmentation task at 83.89% dice. All configurations of the proposed RFCN-1 and RFCN-2 outperformed the U-Net, ranging from a 0.07% to a maximum of 0.48% increase in percentage points on dice (See Table 3). Additionally the results on specificity and sensitivity indicate that the models produced in this research tend to make few false positive predictions (around 99% specificity). False negative predictions do happen, but the distribution between sensitivity and specificity is favorable for patients as false positive predictions have more impact on the brain than false negative predictions. The human brain can remove malicious brain cells that might have been missed by the algorithm, but it can not produce new brain cells if they are destroyed or removed by a surgeon.

5.1 Answering the research questions

The main research question is:

To what extent can a 2D U-Net be improved by combining the U-Net with a Recurrent Neural Network (RNN) on a 3D Brain Tumor segmentation task?

In this research, all configurations of RFCN-1 and RFCN-2 both outperformed the U-Net, yet sometimes only by a small margin. Nevertheless, the results indicate that the RNN component in the RFCN architectures proposed in this research can improve segmentation results. The only additional information added to the RFCN models compared to the U-Net was: a sequence, consisting of segmentation masks of preceding slices from the slice the network is currently segmenting. This additional sequence was built with the baseline U-Net, which predicted at 83.89% dice. Subsequently, any result exceeding 83.89% might indicate that the RFCN is learning from additional information provided by the RNN component in the RFCN architecture. Therefore, these results indicate the leverage of inter-slice information since the best RFCN proposed in this research reached a dice score of 84%.

Results shown in Figures 9, 11, 13 show four 2D MRI slices succeeding each other and their respective true and predicted segmentation masks. Figure 9, shows that the predicted segmentation masks of the U-Net have little inter-slice relation to each other. The true segmentation masks corresponding to the tumor are consistent in shape across the 4 slices, however the predicted segmentation masks are not. The first predicted segmentation mask (top right) and the third predicted segmentation mask are consistent in shape, but the second segmentation mask is much smaller than both of them, and the U-Net doesn't even predict a segmentation mask for the fourth slice (bottom right) (See Figure 9). Both of the RFCN architectures show much better results on the segmentation of this MRI-scan, second predicted segmentation mask is in both models much more visible and both models also produce a accurate segmentation mask for the fourth slice (See Figure 11, 13). Therefore, these results are a strong indication that inter-slice information is being leveraged.

The generalizability of the validation scores to test is further evidence of inter-slice information usage in the RFCN models. The validation data consists of 2D sliced data. When training the U-net, the U-Net achieves a dice score of 87.73% on the validation data. This was translated to 83.89% dice on the 3D test data. For the RFCN models the difference in validation and test scores were significantly lower, the validation dice for RFCN-2 (with sequence length 6) is 84.47%, and the dice score on test is 84.37%. The U-Net has a 3.84% drop in dice from validation to test, where RFCN-2 only has a 0.10% drop. The U-Net is more optimised for predicting on 2D data and that shows whenever you look at the dice score on validation data, but these results don't transfer to 3D data (the test set). The RFCN models are already more optimised for 3D data than the U-Net, because of the RNN component in the models. The RNN component attempts to leverage inter-slice information from the sequences added to the RFCN models. Resulting in a lower dice score on the validation set, but a much more generalizable score for the 3D predictions. Hence, this difference in dice between 2D validation and 3D test data is further evidence of information transfer between 2D slices. Furthermore, to be able to properly assist medical professionals on segmentation tasks it is very important to have a model that generalizes its predictions well. Especially in the case of brain tumor segmentation, considering the impact of a wrongly segmented brain

MRI-scan. Unnecessary procedures can be avoided.

Sub-question 1: What effect does the placement of the RNN component in the RFCN architecture have on the segmentation result?

As shown in research by (Poudel, Lamata, and Montana 2016), (Siam et al. 2017) and (Chen et al. 2016), there are multiple approaches to implement a RNN component combined with a FCN. Inspired by these papers two RFCN architectures were tested in this research.

The two RFCN architectures tested in this research have similar performances, RFCN-2 combined with a sequence of length 6 has a slightly better performance than RFCN-1 with sequence length 2, but is comparable. The difference is the length of the sequence that is added to the two RFCNs to produce the best version of that network, for RFCN-1 it seemed better to add a short sequence and for RFCN-2 a longer sequence improved the result. Whenever the RNN component is added between the contracting path and expanding path (as in RFCN-1) the information added to the network still needs to be processed by the contracting path. There is a higher chance of potentially losing information extracted out of the sequence by the RNN component. Thus, when the information from fewer proceeding slices is stored in the same amount of feature maps, it becomes more likely that this information is going to be used by the RFCN. For RFCN-2 the RNN component is add one layer before the final segmentation, therefore the information added to the network is more likely to be used for the final prediction.

Sub-question 2: How well does the RFCN perform compared to the U-Net on segmenting LGG and HGG?

All models are trained on both HGG and LGG data, whenever a patient has an MRI scan, the medical professional doesn't know what kind of tumor the patient might have. Thus, the model must be able to detect both HGG and LGG. There is less data available on LGG, which makes it harder to train a neural network on LGG. This lack of LGG data results in an imbalanced dataset of 259 HGG and 76 LGG MRI-scans, and subsequently a difference in results on LGG and HGG data.

As mentioned in Section 4.4 all three models perform worst on segmenting LGG compared to HGG, with an average difference of approximately 9% dice. However, The RFCN architectures seem to perform better on the LGG opposed to the U-Net. RFCN-2 paired with sequence length 4, has a 2% increase in dice compared to the U-Net.

The RFCN architecture seems particularly helpful when there is a clearer structure in the tumor shape, as seen in Figures 9, 11, 13. This is a LGG with predicted dice scores of respectively 55.39%, 64.81% and 63.86% by the U-Net, RFCN-1 and RFCN-2. Since the pattern in this MRI-scan was quite clear, it might be possible that the RNN component in the RFCN architecture was able to improve its results because of that (See Figures 9, 11, 13).

5.2 Limitations

Due to the lack of computational power and memory limitations the resolution in this research was scaled down from (240x240x155) to (64x64x41), however it was still possible to extract information about brain structures from these images and produces a segmentation result with 84% dice score. Higher resolution image could possibly detect more and better structures in the brain MRI-scan to increase the segmentation result.

Additionally, due to lack of memory, the sequences in this research were limited to segmentation masks of previous slices of the data, since this is 4 times less memory intensive than adding the full 2D sliced image to the sequence.

5.3 Future research

In this research, the experiments done only used the preceding slices of data. For future research the succeeding slices of data can also be used, since the researchers are already in position of the complete MRI-scan. In addition, the RNN component in this research is only looked at the sequence in one direction, it might be interesting to use a bidirectional RNN component. The complete stack of 2D slices was taken into consideration, however many slices in the data don't contain a tumor in the slice. Hence, preprocessing all the slices in order to crop the z-axes or reduce the dimensions in the x and y-axes might be interesting.

RFCN architectures are very useful for any segmentation task, one implementation this mechanism can be used for is to follow the development of tumors or other malicious diseases that develop over time. Other algorithm can be used to create sequences that hold information, whether the sequence is a image or something else shouldn't matter as long as can combined with the data being processed. Thereby, implementing a RNN components like the sequence used in this research in addition with other algorithms.

6. Conclusion

In this research, two RFCN architectures have been tested on the BraTS 2019 dataset. The RFCN architectures are constructed from a 2D FCN component and a RNN component, respectively; a 2D U-Net and a Convolutional LSTM. The RFCN architectures outperformed the popular U-Net algorithm with a 84.37% dice compared to a 83.89% dice score, indicating that the RNN component in the algorithm successfully managed to leverage inter-slice information across slices. The 2 RFCN architectures tested were one with the RNN component between the contracting and expanding path of the U-Net and one with a RNN component one layer before the final segmentation, the last architecture achieved the best results on segmentation task. The RNN component in both of the RFCN architectures has show the ability to transfer inter-slice information across slices for a brain tumor segmentation task, by analyzing a sequence preceding object of interest. For future research this mechanism could be further analyzed with bidirectional convolutional RNNs and additionally use the succeeding sequences of the object of interest.

References

- Akkus, Zeynettin, Jiri Sedlar, Lucie Coufalova, Panagiotis Korfiatis, Timothy L Kline, Joshua D Warner, Jay Agrawal, and Bradley J Erickson. 2015. Semi-automated segmentation of pre-operative low grade gliomas in magnetic resonance imaging. *Cancer Imaging*, 15(1):12.
- Bakas, Spyridon, Hamed Akbari, Aristeidis Sotiras, Michel Bilello, Martin Rozycki, Justin S Kirby, John B Freymann, Keyvan Farahani, and Christos Davatzikos. 2017. Advancing the cancer genome atlas glioma mri collections with expert segmentation labels and radiomic features. *Scientific data*, 4:170117.
- Bakas, Spyridon, Mauricio Reyes, Andras Jakab, Stefan Bauer, Markus Rempfler, Alessandro Crimi, Russell Takeshi Shinohara, Christoph Berger, Sung Min Ha, Martin Rozycki, et al. 2018. Identifying the best machine learning algorithms for brain tumor segmentation, progression assessment, and overall survival prediction in the brats challenge. *arXiv preprint arXiv:1811.02629*.
- Bisong, Ekaba. 2019. *Building Machine Learning and Deep Learning Models on Google Cloud Platform*. Springer.
- Chen, Jianxu, Lin Yang, Yizhe Zhang, Mark Alber, and Danny Z Chen. 2016. Combining fully convolutional and recurrent neural networks for 3d biomedical image segmentation. In *Advances in neural information processing systems*, pages 3036–3044.
- Çiçek, Özgün, Ahmed Abdulkadir, Soeren S Lienkamp, Thomas Brox, and Olaf Ronneberger. 2016. 3d u-net: learning dense volumetric segmentation from sparse annotation. In *International conference on medical image computing and computer-assisted intervention*, pages 424–432, Springer.
- Dong, Hao, Guang Yang, Fangde Liu, Yuanhan Mo, and Yike Guo. 2017. Automatic brain tumor detection and segmentation using u-net based fully convolutional networks. In *annual conference on medical image understanding and analysis*, pages 506–517, Springer.
- Ganin, Yaroslav and Victor Lempitsky. 2014. N-fields: Neural network nearest neighbor fields for image transforms. In *Asian Conference on Computer Vision*, pages 536–551, Springer.
- Keren, Gil and Björn Schuller. 2016. Convolutional rnn: an enhanced model for extracting features from sequential data. In *2016 International Joint Conference on Neural Networks (IJCNN)*, pages 3412–3419, IEEE.
- Lai, Matthew. 2015. Deep learning for medical image segmentation. *arXiv preprint arXiv:1505.02000*.
- Long, Jonathan, Evan Shelhamer, and Trevor Darrell. 2015. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440.
- Menze, Bjoern H, Andras Jakab, Stefan Bauer, Jayashree Kalpathy-Cramer, Keyvan Farahani, Justin Kirby, Yuliya Burren, Nicole Porz, Johannes Slotboom, Roland Wiest, et al. 2014. The multimodal brain tumor image segmentation benchmark (brats). *IEEE transactions on medical imaging*, 34(10):1993–2024.
- Milletari, Fausto, Nassir Navab, and Seyed-Ahmad Ahmadi. 2016. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *2016 Fourth International Conference on 3D Vision (3DV)*, pages 565–571, IEEE.
- Myronenko, Andriy. 2018. 3d mri brain tumor segmentation using autoencoder regularization. In *International MICCAI Brainlesion Workshop*, pages 311–320, Springer.
- Ning, Feng, Damien Delhomme, Yann LeCun, Fabio Piano, Léon Bottou, and Paolo Emilio Barbano. 2005. Toward automatic phenotyping of developing embryos from videos. *IEEE Transactions on Image Processing*, 14(9):1360–1371.
- Noh, Hyeonwoo, Seunghoon Hong, and Bohyung Han. 2015. Learning deconvolution network for semantic segmentation. In *Proceedings of the IEEE international conference on computer vision*, pages 1520–1528.
- Poudel, Rudra PK, Pablo Lamata, and Giovanni Montana. 2016. Recurrent fully convolutional neural networks for multi-slice mri cardiac segmentation. In *Reconstruction, segmentation, and analysis of medical images*. Springer, pages 83–94.
- Ronneberger, Olaf, Philipp Fischer, and Thomas Brox. 2015. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241, Springer.
- Schwartzbaum, Judith A, James L Fisher, Kenneth D Aldape, and Margaret Wrensch. 2006. Epidemiology and molecular pathology of glioma. *Nature clinical practice Neurology*, 2(9):494–503.

- Siam, Mennatullah, Sepehr Valipour, Martin Jagersand, and Nilanjan Ray. 2017. Convolutional gated recurrent networks for video segmentation. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 3090–3094, IEEE.
- Smoll, Nicolas R, Karl Schaller, and Oliver P Gautschi. 2013. Long-term survival of patients with glioblastoma multiforme (gbm). *Journal of Clinical Neuroscience*, 20(5):670–675.
- Taghanaki, Saeed Asgari, Kumar Abhishek, Joseph Paul Cohen, Julien Cohen-Adad, and Ghassan Hamarneh. 2019. Deep semantic segmentation of natural and medical images: A review. *arXiv preprint arXiv:1910.07655*.
- Taha, Abdel Aziz and Allan Hanbury. 2015. Metrics for evaluating 3d medical image segmentation: analysis, selection, and tool. *BMC medical imaging*, 15(1):29.
- Xingjian, SHI, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. 2015. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *Advances in neural information processing systems*, pages 802–810.
- Yao, Rui, Yikun Zhang, Cunyuan Gao, Yong Zhou, Jiaqi Zhao, and Lina Liang. 2019. Lightweight video object segmentation based on convgru. In *Chinese Conference on Pattern Recognition and Computer Vision (PRCV)*, pages 441–452, Springer.

