Slovak University of Technology in Bratislava

Faculty of Informatics and Information Technologies

Reg. No.: FIIT-182905-97014

Bc. Jakub Povinec

Segmentation of Brain Magnetic Resonance Images using Deep Neural Networks

Master's Thesis

Thesis supervisor: Prof. Vanda Benešová

May 2024

Slovak University of Technology in Bratislava

Faculty of Informatics and Information Technologies

Reg. No.: FIIT-182905-97014

Bc. Jakub Povinec

Segmentation of Brain Magnetic Resonance Images using Deep Neural Networks

Master's Thesis

Study programme: Intelligent Software Systems

Study field: Computer Science

Training workplace: Institute of Computer Engineering and Applied Informatics

Thesis supervisor: Prof. Vanda Benešová

May 2024

Slovak University of Technology in Bratislava
Ústav počítačového inžinierstva a aplikovanej informatiky

Faculty of Informatics and Information Technologies
Academic year: 2023/2024
Reg. No.: FIIT-182905-97014

MASTER THESIS TOPIC

Student:	**Bc. Jakub Povinec**
Student's ID:	97014
Study programme:	Intelligent Software Systems
Study field:	Computer Science
Thesis supervisor:	prof. Ing. Vanda Benešová, PhD.
Head of department:	Ing. Katarína Jelemenská, PhD.

Topic: **Segmentation of brain magnetic resonance images using deep neural networks**

Language of thesis: English

Specification of Assignment:

Segmentácia anatomických štruktúr v snímkach magnetickej rezonancie mozgu je dôležitá pre diagnostiku a prípadnú, následnú prípravu liečby pacienta. Keďže sa jedná o volumetrické dáta, ich manuálna segmentácia je vo všeobecnosti veľmi zložitá a časovo náročná. V posledných rokoch zaznamenal výskum segmentačných prístupov výrazný pokrok najmä vzhľadom k rozvoju metód počítačového videnia založených na použití hlbokých neurónových sietí. Pri vytváraní úspešných modelov určených na segmentáciu medicínskych dát, je jedným z kľúčových problémov získanie dostatočne rozsiahleho dátového súboru s kvalitnými anotáciami. Z tohto hľadiska sa zdajú byť perspektívne hlavne metódy využívajúce učenie s čiastočným učiteľom (semi-supervised learning), ktoré sú efektívne aj pri menšom dátovom súbore. Analyzujte súčasný stav problematiky spracovania neurologických snímok mozgu z magnetickej rezonancie metódami počítačového videnia. Zamerajte sa predovšetkým na metódy segmentácie využívajúce hlboké neurónové siete s čiastočným učiteľom. Navrhnite a implementujte vlastnú metódu určenú na segmentáciu nádorov alebo rôznych anatomických častí mozgu zo snímok magnetickej rezonancie založenú na hlbokých neurónových sieťach a využívajúcu učenie s čiastočným učiteľom. Vyhodnoďte úspešnosť vašej metódy pomocou zaužívaných metrík v danej oblasti a porovnajte ju s inými relevantnými riešeniami.

Deadline for submission of Master thesis:	17. 05. 2024
Approval of assignment of Master thesis:	18. 04. 2024
Assignment of Master thesis approved by:	prof. Ing. Vanda Benešová, PhD. – study programme supervisor

Declaration of honour

I, Jakub Povinec, honestly declare that this thesis is my independent work under the supervision of prof. Ing. Vanda Benešová, PhD., except where I acknowledged the work of other people in cited literature.

In Bratislava, 17.5 .2024 .....

Bc. Jakub Povinec

Special thanks

I would like to thank my supervisor Professor Vanda Benešová for guidance and help during the work on my Master's thesis. I would also like to thank Siemens Healthineers for sponsoring the Azure Machine Learning services, which we used to train and evaluate our models.

Annotation

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Degree Course: Intelligent Software Systems

Author: Bc. Jakub Povinec
Master's Thesis: Segmentation of Brain Magnetic Resonance Images using Deep Neural Networks
Supervisor: Prof. Vanda Benešová
May 2024

In general, successful neural network models require extensive datasets for their training. This issue is even more prevalent in medicine, where obtaining large, high-quality datasets with a sufficient number of annotated samples is particularly difficult because of their expensive acquisition. In this regard, approaches which can be sufficiently trained on easily obtainable labels are more promising.

In our work, we developed a method which utilised user-defined clicks for segmentation of Vestibular Schwannomas from T1ce and T2 MRI sequences. The main idea behind the proposed method was to use an additional correction network to refine initial (imperfect) segmentations, based on the provided clicks. The refinement is done on a local level, where the clicks denote the area of the initial segmentation that needs to be corrected.

Our work can therefore be split into two main parts. In the first part, we implemented an initial segmentation model based on the 3D U-Net architecture that was trained on a few fully-annotated images. In the second part of the work we explored different architectures and methods for implementing the correction network, where we eventually settled on a multi-encoder U-Net.

Anotácia

Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií
Študijný program: Inteligentné softvérové systémy

Autor: Bc. Jakub Povinec
Diplomová práca: Segmentácia snímok magnetickej rezonancie mozgu pomocou hlbokých neurónových sietí
Vedúci projektu: Prof. Vanda Benešová
May 2024

Vo všeobecnosti je pre natrénovanie úspešných neurónových sietí potrebné veľké množstvo dát. Tento problém je ešte výraznejší v medicíne, kde je získanie veľkých a kvalitných datasetov s dostatočným počtom anotovaných vzoriek obzvlášť náročné. V tomto ohľade sú preto sľubnejšie prístupy, ktoré môžu byť dostatočne natrénované aj na jednoduchších anotáciach.

V našej práci sme vyvinuli metódu na segmentáciu vestibulárnych schwannómov z T1ce a T2 MRI sekvencií, ktorá využíva používateľom definované kliky. Hlavnou myšlienkou navrhovanej metódy bolo využitie dodatočnej korekčnej siete na vylepšenie počiatočných (nedokonalých) segmentácií na základe poskytnutých klikov. Toto vylepšovanie sa vykonáva na lokálnej úrovni, kde kliknutia označujú oblasť počiatočnej segmentácie, ktorú je potrebné upraviť.

Našu prácu je teda možné rozdeliť na dve hlavné časti. V prvej časti sme implementovali počiatočný segmentačný model založený na architektúre 3D U-Net, ktorý bol trénovaný na niekoľkých plne anotovaných obrázkoch. V druhej časti práce sme skúmali rôzne architektúry a metódy implementácie korekčnej siete, kde sme sa nakoniec rozhodli použiť U-Net architektúru s viacerými enkódermi.

Table of contents

List of used abbreviations

GPU	Graphics Processing Unit
CPU	Central Processing Unit
MRI	Magnetic Resonance Imaging
CT	Computer Tomography
BraTS	Brain Tumour Segmentation challenge
T1	T1-weighted (MRI sequence)
T1ce	post-contrast T1-weighted (MRI sequence)
T2	T2-weighted (MRI sequence)
FLAIR	T2 Fluid Attenuated Inversion Recover (MRI sequence)
ANN	Artificial Neural Network
CNN	Convolutional Neural Network
GAN	Generative Adversarial Network
ReLU	Rectified Linear Unit
PReLU	Parametric Rectified Linear Unit
DSC	Dice Score Coefficient
BCE	Binary Cross Entropy

Chapter 1

Introduction

The use of computer vision is becoming increasingly more popular in solving medical imaging tasks. This rise in popularity came with the development of more complex deep learning methods commonly based on the use of convolutional neural networks (CNNs). These methods are able to solve different tasks that are crucial for diagnostics or treatment planning, such as segmentation or classification from different modalities [1]. Previously, these tasks had to be performed manually by clinical experts, which was especially demanding due to the nature of the data.

Training neural network models for tasks involving medical image data poses several challenges that need to be addressed [2]. One of the common problems faced in this domain is the acquisition of high-quality datasets with a sufficient number of samples. In medical imaging, obtaining large annotated datasets is often challenging due to the requirement for expert annotations, which are time-consuming and often subjective, but also because of privacy concerns and the overall complexity of acquiring and curating these datasets. In this regard, methods which can be sufficiently trained on smaller datasets or with easily obtainable labels can

therefore be viewed as more promising.

In our work, we developed a method which utilises weakly-annotated data (clicks) for image segmentation. The main idea behind the method is to utilise the provided clicks to refine an initial imperfect segmentation. In general, our work can be divided into two main parts; generation of an initial segmentation and subsequent correction of this segmentation with an auxiliary correction network.

In the first part, we trained a simple segmentation model. This segmentation model was trained in a fully-supervised manner on a limited amount of training data. In the second part, we trained a correction network that based on user-defined clicks extracted small areas of the segmentation, and subsequently refined them.

For training and evaluation of our method, we used the publicly available Vestibular Schwannoma (VS) dataset[1], that consist of brain MRI scans in both T1ce and T2 sequences [3]. Vestibular Schwannoma is a benign tumour that originates from the eighth cranial nerve, most commonly in the vestibular part [4]. It is a relatively small tumour, where the majority of diagnosed VS are smaller than 20mm [5]. Vestibular Schwannomas are primarily assessed on MRI, where the contrast-enhanced T1-weighted sequences are considered to be the most telling, but additionally, T2-weighted sequences may be utilised in order to rule out other structures and correctly classify the tumour [4].

[1] https://www.cancerimagingarchive.net/collection/vestibular-schwannoma-seg/

Chapter 2

Analysis

2.1 Popular segmentation architectures

There are several different CNN architectures which have been proven to be successful for tasks related to the processing of medical data, in particular the segmentation of medical images.

2.1.1 U-Net based architectures

Recently, the U-Net architecture proposed by Ronneberger et al. [6] has emerged as one of the most popular and successful architectures used for medical image segmentation.

This architecture consists of a downsampling (encoder) and a symmetrical upsampling (decoder) path that forms a U shape-like architecture. Both the encoder and decoder are composed of multiple convolutional layers with kernel size 3 and ReLU activation functions. The downsampling path includes 2x2 max-pooling operations which reduce the size of the output feature maps. Alternatively, the expanding

path consists of 2x2 up-convolution operations with stride 2 that increase the size of the feature maps. The last layer of the network consist of a 1x1 convolution followed by a softmax activation function. The architecture of the original U-Net is provided in the Figure (2.1).

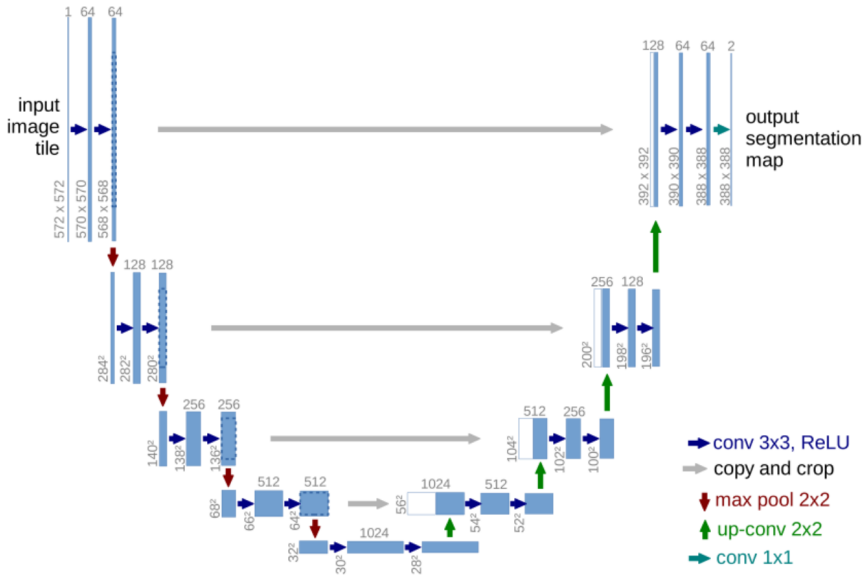

Figure 2.1: Architecture of the original U-Net [6].

The biggest advantage of U-Net is that opposed to other architectures, U-Net effectively captures both the local and global contextual information [7]. This advantage comes from the concatenation of the last feature map from the down-sampling path with the corresponding upsampled feature map from the upsampling path. This concatenation operation produces so-called "skip connections", which provide additional information to the network, aiding in more precise localisation of the identified features [6].

While the original U-Net architecture uses 2D convolutions, there are also modifications of this architecture for the segmentation of volumetric medical images, which utilise 3D convolutional layers, such as V-Net, 3D U-Net or UNet++ [8, 9,

10].

2.1.1.1 3D U-Net

The 3D U-Net architecture proposed by Çiçek et al. [9] is a modification of the U-Net architecture suitable for segmenting volumetric medical images.

The overall architecture follows the same structure as the original U-Net, where the most notable changes of this architecture compared to the original U-Net are the use of 3D convolutional, maxpooling and upsampling (up-convolutional) layers as opposed to the 2D equivalents used in the original U-Net. The authors also double the number of channels before the downsampling operation (in the last convolution layer of each step as opposed to the first convolutional layer after the downsampling in the original U-Net). This change is supposed to avoid the representational bottleneck, which can result in an overall less expressive network [11]. Another notable change is the addition of batch normalisation before each activation function.

The authors also argue that training of such architecture on volumetric medical images requires fewer training samples since 3D images often contain numerous recurring structures and shapes. In their paper "3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation" [9], the authors provide a demonstration that the 3D U-Net is, with specialised loss function, also able to be effectively trained for segmentation of volumetric medical images from sparse labels.

2.1.1.2 V-Net

V-Net is an U-Net based architecture proposed by Milletari et al. [8], which is also suitable for segmentation of volumetric medical data. The V-Net architecture can be seen in the Figure (2.2).

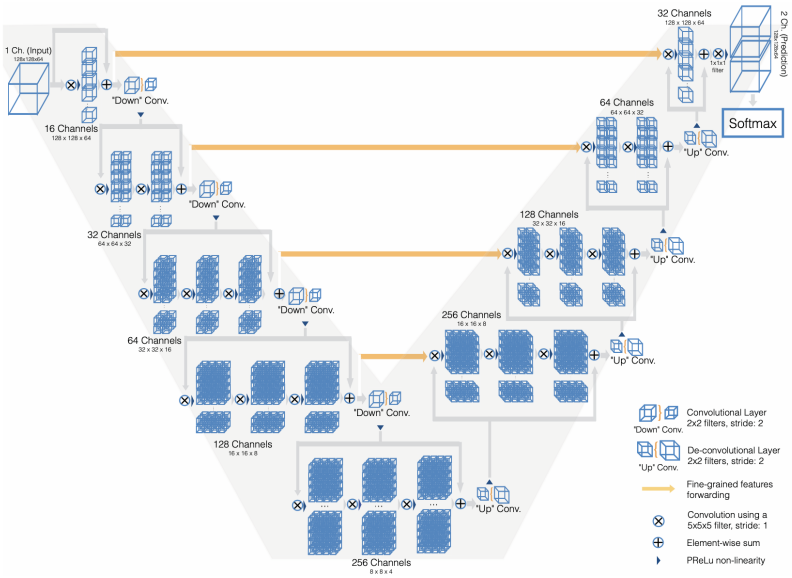

Figure 2.2: Architecture of the V-Net [8].

Similarly to the original U-Net the V-Net consists of a downsampling and an upsampling paths which include mulitple stages with 3D convolutional, downsampling and upsampling operations. In contrast with both the 3D U-Net and the original U-Net the authors used kernels of size 5x5x5 instead of 3x3x3 (3x3) for the convolutional layers. The authors also used strided convolutions as downsampling operations, as opposed to maxpooling which is used in both the original U-Net and the 3D U-Net. An another notable change is the use of PReLU activation function instead of the commonly used ReLU. The major difference however is that the downsampled (or upsampled) output from each stage is also added to the last convolutional layer of the given stage forming additional skip connections on each stage of both the upsampling and downsampling paths in addition to the skip connections between the both paths. The authors argue that the use of these residual skip connections is beneficial for both the quality of the results and the convergence time of the network [8].

The V-Net architecture has also been used in different works to successfully seg-

ment brain MRI scans [12, 13].

2.1.1.3 Attention U-Net

In the Attention U-Net proposed by Oktay et al. [14], the authors expanded upon the original U-Net architecture by utilising additive attention gates in the decoder part of the network. An overview of the proposed network architecture as well as the architecture of the proposed attention mechanism can be seen in the Figures (2.3) and (2.4) respectively.

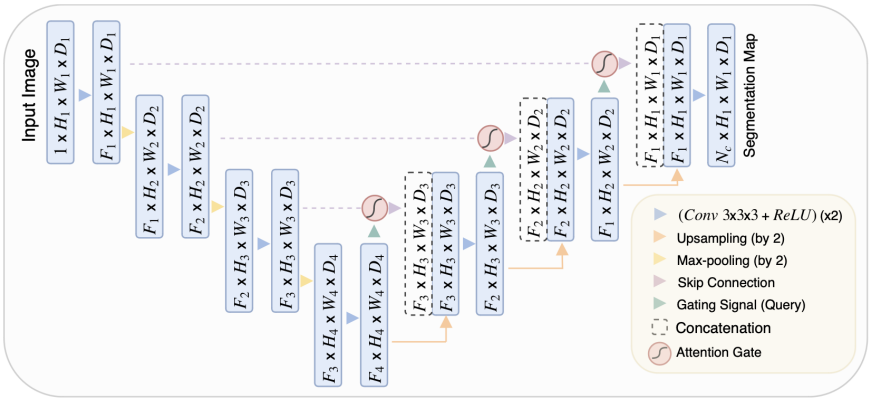

Figure 2.3: Architecture of the Attention U-Net proposed by Oktay et al. [14].

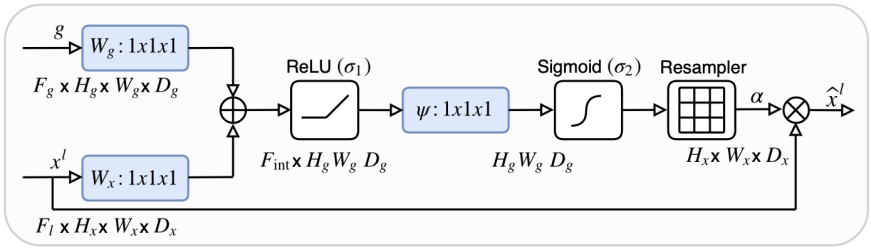

Figure 2.4: Architecture of the attention gate as proposed by Oktay et al. [14].

The attention gates are placed right before each skip connection and help the network extract relevant features from the input feature maps with the use of attention coefficients α, which weight the feature maps from the encoder path (skip

connection) before they are concatenated with the feature maps in the decoder path.

The attention gates essentially scale the input features x by multiplying them with the computed attention coefficient α. The α coefficient is computed by applying a 1x1x1 convolution ψ with one output channel to the combined feature maps of the input and the gating signal [14]. The result is then run through the sigmoid activation function, which squashes its values to the range [0, 1].

Mathematically the computation of α can be described by the equations (2.1, 2.2), where the x represents the input feature maps (the skip connection), g represents the gating signal (features from the previous stage) and Θ represents the parameters of the attention gate, i.e. the weights (W_x, W_g) of both convolutional layers and biases (b_g, b_ψ) [14].

$$q_{att}^l = \psi^T(\sigma_1(W_x^T x_i^l + W_g^T g_i + b_g)) + b_\psi \tag{2.1}$$

$$\alpha_i^l = \sigma_2(q_{att}^l(x_i^l, g_i; \Theta_{att})) \tag{2.2}$$

Apart from the attention gates in the upsampling path, the architecture follows the standard 3D U-Net architecture with 3x3x3 convolutions and ReLU activation functions.

The authors argue that the Attention U-Net is particularly beneficial for the segmentation of objects of smaller sizes. Similar methods utilising the attention mechanism have also been used in the segmentation of gliomas from MRI scans [15, 16] and also proved to be helpful in the segmentation of unbalanced datasets as well as in various weakly-supervised methods [17, 18, 19].

2.1.1.4 UNet++

The UNet++ architecture proposed by Zhou et al. [10] is a more complex modification of the original U-Net architecture, which aims to improve the quality of segmentations for medical images.

The main modification that the UNet++ introduced are to the skip connections between both paths, which as the authors argue, ensure that the network is able to capture "fine-grained details" of the target object more effectively and therefore produce more accurate segmentations [20]. Compared to the original U-Net architecture, the UNet++ incorporates convolution layers in the skip connections, where each convolutional layer is connected with all previous convolutional layers on the same stage as well as with the upsampled output from lower layer [7]. These connections create dense convolutional blocks, which help to ensure a better understanding of the image's content throughout the network. Lastly, UNet++ also utilise deep supervision, which enables the possibility of model pruning and allows for more accurate segmentations [10]. The UNet++ architecture can be seen in the Figure (2.5).

The authors tested their architecture on various different medical datasets, including brain MRI scans, histological images and liver CT scans and achieved significant improvement compared to the original U-Net architecture [20].

2.1.2 Generative Adversarial Networks

Methods based on the Generative Adversarial Networks (GANs) architecture have been quite popular in various fields of computer vision and have proven to be also effective for medical image segmentation, where in some cases showed even better results than more traditional approaches based on fully convolutional neural networks [21, 22].

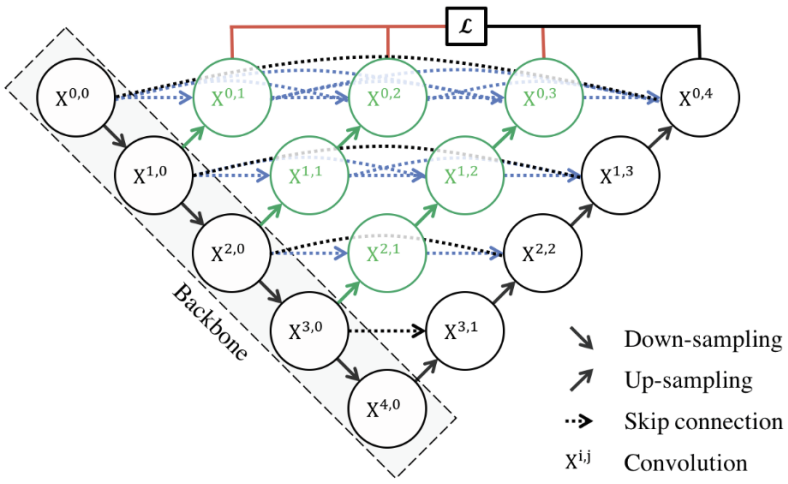

Figure 2.5: Architecture of the UNet++, where the black components represent the original U-Net and the green and blue represent dense convolutional blocks [10].

GANs typically consist of two networks, one that is generating new, synthetic images and the other, which is trying to differentiate the synthetic images from the real ones, where both gradually improve for their respective tasks [23]. Both the generator and discriminator network generally consists of a CNN. GANs can also be utilised in multiple ways for medical image segmentation. One way is by generating new synthetic images together with their corresponding segmentations [24], an example of this architecture can be seen in Figure (2.6). Similarly, they can also be used for image segmentation in a way where the generator only creates the new segmentations from the original images and the output from the discriminator is used to further refine the generator (final segmentations) [25]. An example of this type of architecture can be seen in Figure (2.7).

Besides image segmentation GANs have been used for various different tasks in medical imaging such as generation of new synthetic images, for converting medical images from one modality to another or they can aid in image registration [21].

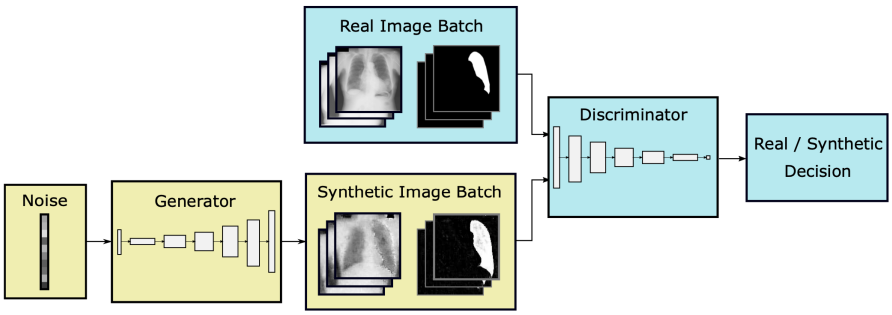

Figure 2.6: GAN architecture for synthesis and segmentation of thorax X-ray images as proposed by Neff et al. [24].

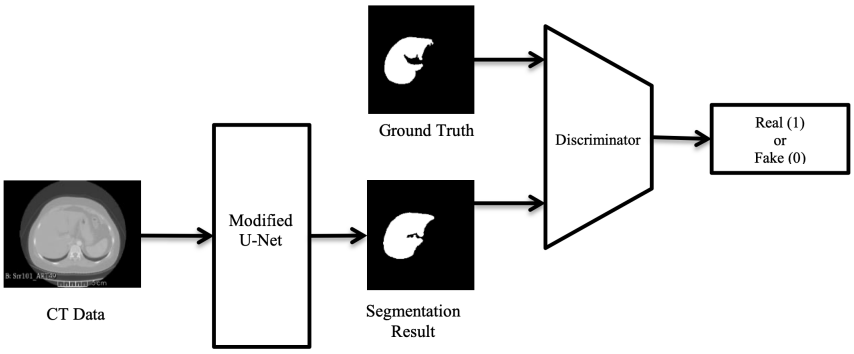

Figure 2.7: GAN architecture for segmentation of liver CT images as proposed by Enokiya et al. [25].

2.2 Common challenges

The use of neural networks for medical diagnosis has the potential to be very promising, but there are several challenges that must be overcome when training neural network models for tasks involving medical image data. These challenges, among others, include a limited number of available annotated datasets, class imbalance and high distribution differences between datasets [2].

2.2.1 Limited number of available annotations

Neural network models need to be trained on a large dataset of annotated images in order to be successful. It is often difficult and time-consuming to obtain such

datasets, especially for medical images, where the data is often subjected to privacy regulations. One way to address this issue is to expand the annotated dataset with the use of different data augmentations or to create entirely new synthetic data with GANs [26, 21].

2.2.1.1 Semi-supervised learning

Semi-supervised learning can be beneficial in cases, where only a small amount of images have their corresponding annotations. The goal of this method is to use the unlabelled data to train a better-performing model than what could be achieved by using only the labelled data and to achieve a performance which is similar to the one of a fully annotated dataset [27].

There are several different ways in which semi-supervised segmentation methods can be designed and trained. One approach is to use fully supervised networks as a starting point and then fine-tune the models on the unlabelled data in order to improve their performance [28].

Another approach is to use a self-training method, where the model is first trained on the labelled data and then used to make predictions on a few unlabelled samples [27]. These new annotations are added to the training set, and the model is trained again on the now expanded labelled dataset, where this process is repeated multiple times.

2.2.1.2 Weakly-supervised learning

Similarly to semi-supervised learning, in weakly-supervised learning the models are mainly trained by using partially labelled, **weakly-annotated data** with only a small amount of samples with full annotations. This technique is particularly beneficial in medical image segmentation, where obtaining pixel-level annotations

is more challenging [27]. The commonly used types of weakly-annotated data, which are used as a substitution for the high-quality pixel-level annotations are image-level labels or tags [29, 30], bounding boxes [31], extreme points [32, 33], user-defined (generated) clicks or points [34, 35, 36, 37] and scribbles [38, 39]. These simple annotations then can be used to generate pseudo-masks or region proposals [40, 41].

2.2.1.3 Active learning

Active learning is another approach which aims to minimise the need for large amounts of fully annotated datasets. In active learning, the model is first trained by using a large set of unlabelled samples. Subsequently after each iteration, a small subset of samples is chosen by some heuristic for annotating, often based on the representativeness of the unlabelled samples [27]. The selection of only the most representative samples, allows the network to be trained by using only a few fully annotated samples while achieving high accuracy.

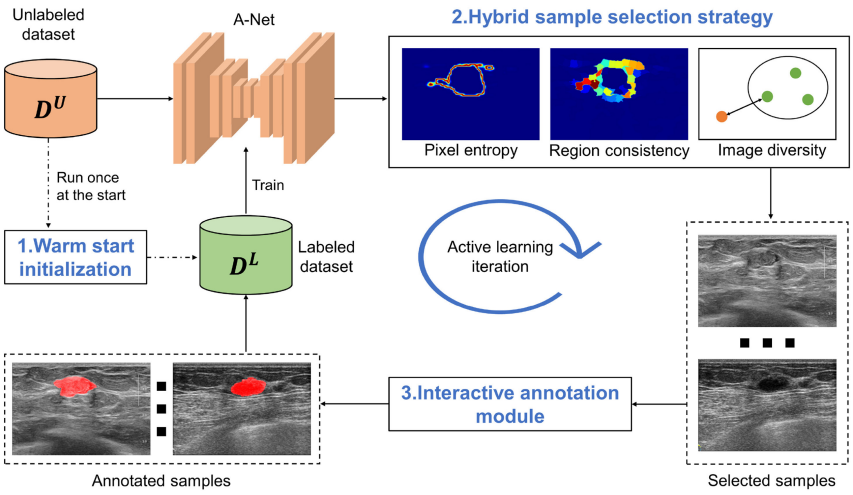

Figure 2.8: Overview of an active learning framework for medical image segmentation proposed by Li et al. [42], which utilises pixel entropy and regional consistency to select samples with high diversity and uncertainty for annotation.

Active learning has proven to be very effective in medical image segmentation and the right technique for selecting the optimal samples for annotation can greatly reduce the needed amount of fully annotated data [42, 43, 44].

2.2.2 Class imbalance

One issue that is particularly frequent in medical datasets is a class imbalance, where it is often the case that certain classes are heavily underrepresented in comparison to others. This primarily occurs because the relevant objects typically occupy only a small region within the image. It is especially common in radiology where, for example, tumours are often much smaller than the overall scan and the background voxels are therefore much more prevalent. Models trained on unbalanced datasets can become biased towards the majority class, which may lead to lower performance and accuracy.

Sampling techniques such as oversampling or undersampling are often effective methods to address the class imbalance in classification tasks [45]. However, when it comes to medical image segmentation, these methods are not directly applicable and require specialised approaches beyond simple data sampling.

A common technique to address the class imbalance in medical image segmentation is to employ **specialised loss functions**. These loss functions either assign higher weights to the minority class or use adaptive strategies to balance the contribution of different classes during training [46]. Among the popular loss functions used for imbalanced segmentation of medical images are the distribution loss functions, such as weighted cross-entropy or Focal loss, region-based loss functions, such as Dice loss or Tversky loss and combined loss functions, such as DiceFocal loss [47].

Based on the comparisons of different specialised loss functions on several unbal-

anced datasets by Ma et al. [47] and Yeung et al. [48] the selection of the right loss function has a significant influence on the performance of the model. Among the best-performing loss functions were the Dice loss, FocalTversky loss and DiceFocal loss.

2.2.2.1 Dice loss

The Dice loss directly optimises the Dice Similarity Coefficient (DSC), which measures the overlap between two regions and is a widely used metric for evaluating segmentation performance [8]. The dice loss is therefore defined as:

$$DSC(A, B) = \frac{2(A \cap B)}{|A| + |B|} \tag{2.3}$$

$$L_{Dice} = 1 - DSC \tag{2.4}$$

The generally defined Dice loss is however very unstable for highly unbalanced datasets [49] and more commonly is used either in combination with other loss functions [50, 51] or with weights as a weighted version of the Dice loss [52].

2.2.2.2 FocalTversky loss

The FocalTversky loss is a combination of Focal loss and Tversky index [47]. Focal loss is a modification of the cross-entropy loss functions that aims at preventing a large number of easy (negative) examples from dominating the gradient by down-weighting them [53]. The Tversky index (TI) is a modification of the DSC that adds weights to the false positives and false negatives, thus making it more suitable for highly unbalanced datasets [48].

$$L_{FocalTversky} = \sum_{c}(1 - TI_c)^{\gamma} \tag{2.5}$$

The Focal Tversky loss modifies the Tversky index by incorporating a focal factor (γ). The focal factor helps to adjust the contribution of each sample based on its difficulty and gives more weight to challenging samples that are hard to segment accurately [54].

2.2.2.3 DiceFocal loss

The DiceFocal loss combines benefits from both Dice and Focal loss functions and is therefore effective on imbalanced datasets [48]. The loss function is defined as a simple combination of Dice loss and Focal loss.

$$L_{DiceFocal} = L_{Dice} + \gamma L_{Focal} \tag{2.6}$$

The Dice loss helps the model learn the class distribution, which helps with class imbalanced and on the other hand, the focal loss forces the model to focus on learning poorly classified voxels (pixels) more effectively [51]. The γ parameter modifies the trade-off between Dice loss and Focal loss [51].

2.2.3 Domain shift

On the other hand, the domain shift problem occurs when there is a difference between the training dataset, also called the source and the testing dataset, also called the target [55]. In the case of medical images, the domain shift can occur when, for example, the dataset is built up from MRI scans across different scanners, settings or consists of different sequences. One way to solve this problem is by utilising transfer learning, in particular domain adaptation.

Domain adaptation is a technique used to adapt a model trained on one domain to work in another domain. Overall, the goal of domain adaptation is to improve the

performance of a model on the target domain by leveraging the knowledge gained from the source domain. It is often used when there is a lack of labelled data in the target domain or when the data distribution between the source and target domains is significantly different.

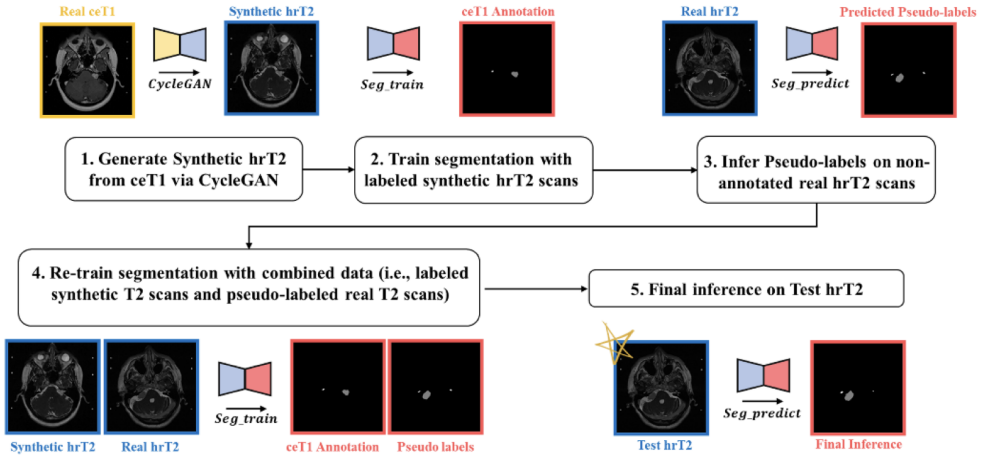

Figure 2.9: Overview of an unsupervised domain adaptation method for segmenting Vestibular Schwannomas from unannotated hrT2 MRI sequences proposed by Shin et al. [56].

Domain adaptation can also be divided into supervised, semi-supervised and unsupervised based on the availability of annotations in the target domain [55]. Unsupervised domain adaptation in particular, can be done in a number of ways, for example by aligning the distributions of the source and target domains [40] or by using feature and image alignment to learn domain-invariant features [57]. Image alignment is frequently used with GANs, more specifically Cycle GAN, which was proposed by Zhu et al. [58] and its advantage lies in the fact that it can translate images from one domain to another. An example of unsupervised domain adaptation can be seen in the Figure (4.7), where the authors tried to segment Vestibular Schwannomas from unannotated hrT2 sequences.

Chapter 3

Related work

In this chapter, we provide a short overview of couple works that focus on the segmentation from medical images. We looked into the state-of-the-art methods for segmentation from unbalanced medical datasets (Section 3.1), different approaches of segmentation from weakly-annotated data (Section 3.2) and interactive segmentation methods (Section 3.3), particularly those which utilise user-defined clicks to correct predicted segmentations.

3.1 Segmentation from medical image data

In this section, we reviewed some of the recent works which focused on segmentation of medical images. In particularly we looked for works that segmented Vestibular Schwannomas.

The majority of recent approaches utilise the U-Net architecture, which is additionally enhanced by the use of residual blocks or by another form of supervision, mostly through the use of additional attention modules in the decoder path.

3.1.1 Automatic Segmentation of Vestibular Schwannoma from contrast-enhanced T1 and high resolution T2 sequences [18]

Shapey et al. [18] proposed an attention-based 2.5D CNN for segmenting Vestibular Schwannoma (VS) from T1ce and T2 MRI sequences. The proposed network used a combination of 2D and 3D convolutions and had a U-Net like encoder-decoder architecture with 5 levels of convolutions. The first 2 levels on both sides of the network used 2D convolutions, whereas the following levels employed 3D convolutions. The authors argue that the variations between these levels are beneficial because of the difference in the resolution of VS tumours from different directions, where the in-plane resolution is 2–3 times bigger than the through-plane resolution. The proposed architecture can be seen in the Figure (3.1).

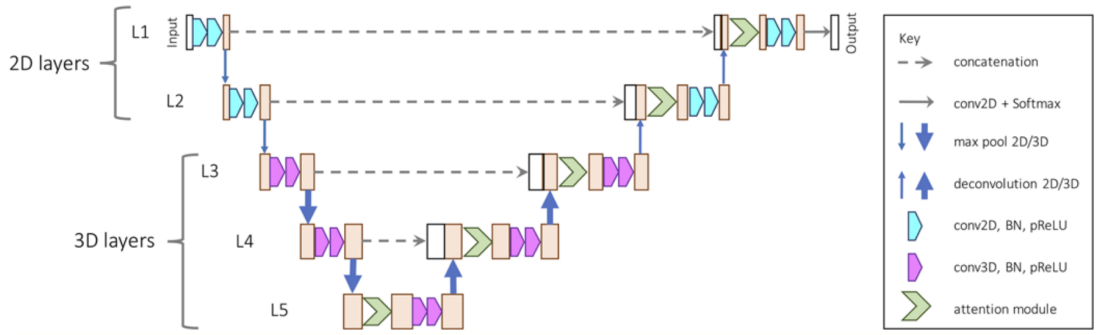

Figure 3.1: The 2.5D U-Net architecture proposed by Shapey et al. [18].

Since the segmented tumours typically cover only a small area, the authors added spatial attention to the decoder. Spatial attention allows the network to focus more on the tumour while ignoring the background. The attention module follows similar architecture as in the Attention U-Net [14] and consists of two convolutional layers and a sigmoid activation function [18]. As the loss function, the authors used weighted dice loss which gave higher weights to voxels misclassified by the

network.

Based on the evaluation made by the authors, the proposed method achieved a 0.93 dice coefficient for the segmentation of the VS tumours [18].

3.1.2 Automatic segmentation of vestibular schwannomas from T1-weighted MRI with a deep neural network [17]

In their work Wang et al. [17] proposed a method for segmenting VS from T1-weighted MRI scans. The proposed method used similar U-Net architecture to the one proposed by Shapey et al. [18], where both architectures incorporated attention blocks in the decoder part. On the other hand, the architecture by Wang et al. only consisted of 3D layers as opposed to the architecture by Shapey et al. which used 2D operations on the upper layers of the network. Another innovation of the proposed architecture was the use of residual blocks in each stage of the network, as opposed to the commonly used combination of two convolutional layers. The authors argue that the addition of another skip connection in each stage helps with exploding and vanishing gradients [17].

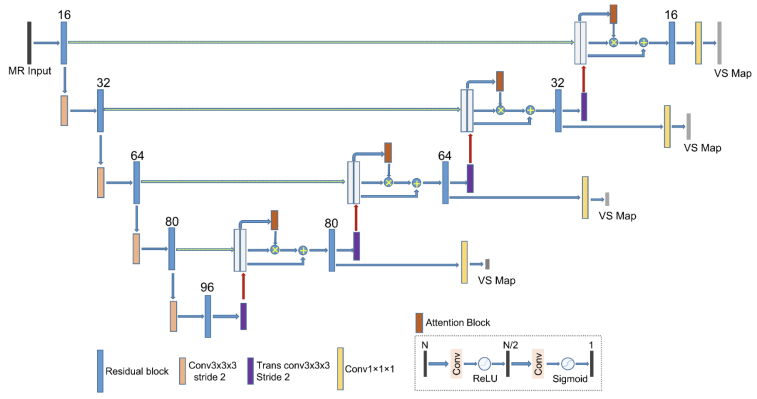

Figure 3.2: The architecture proposed by Wang et al. [17].

Another interesting addition is the use of deep supervision at each stage of the decoder, where the segmentation maps were generated using a 1x1x1 convolution and sigmoid activation function applied to the feature output from the residual block. These output maps were added to the loss function and compared to the ground truth of a corresponding resolution [17]. The proposed architecture can be seen in the Figure (3.2).

Similarly to the method proposed by Shapey et al. the authors used the conventional dice loss function for the spatial attention loss and also for the deep supervision loss. The total loss was therefore computed as a sum of these losses at the multiple levels of the decoder.

Based on the evaluation made by the authors, the proposed method achieved a dice coefficient of 0.917 for the segmentation of VS from the T1-weighted MRI scans [17]. The authors also tried to use both T1 and T2 weighted MRI scans, however, the use of another sequence did not have a significant impact on the final results.

3.1.3 Multimodal segmentation network

In [59], the authors used a "multimodal" U-Net to segment gliomas from four different MRI sequences. Their architecture differed from the original U-Net by having 4 separated encoders (one for each MRI sequence). The authors incorporated "hyper-dense connectivity" between the encoders, resulting in feature maps that are not only interconnected within each encoder, but also across different encoders. According to the authors, this approach allows the network to capture more complex relationships between the different modalities.

In [60], the authors used a multi-encoder architecture to segment different structures from histological images of hearth tissue. An interesting part of their work

was the mechanism they used to merge the resulting feature maps from the individual encoders. For this the authors utilised a custom attention module, which consisted of a spatial attention module and a channel-wise attention, which according to the authors, helped the network to balance the importance of the given modalities for certain regions. Based on their analysis the multi-encoder architecture and the attention mechanism improved the networks performance on certain problematic structures and overall, the model produced finer segmentations [60].

3.2 Segmentation from weakly-annotated data

In this chapter we focused primarily on the popular segmentation approaches, which utilise user defined or automatically generated click or scribble based annotations.

3.2.1 Scribble-based annotations

One notable work, which utilises scribble-based annotations was proposed by Wang et al. [61]. In this work, the authors focused on refining the generated segmentations through user-defined scribbles. The interesting part of this work is the weighted loss function, which assigns different weights to foreground and background scribbles.

Similarly Can et al. [62] also based their segmentation method on user-defined scribbles. In their approach, they utilised scribble-based annotations and the random walker algorithm to generate pseudo-masks. These pseudo-masks were then used in training with a combination of the network's prediction to both train the network and update (generate) annotations in the training set [62].

The authors evaluated their method on the segmentation of prostate from CT scans and achieved comparable results with a fully-supervised approaches.

3.2.2 Going to Extremes: Weakly Supervised Medical Image Segmentation [33]

Similarly to the previous work by Can et al. [62], in the paper by Roth et al. [33] the authors proposed a segmentation method which utilised extreme points to generate an initial pseudo-mask by using the random walker algorithm. This initial segmentation served as a noisy supervision signal and was used to train the segmentation model. An overview of their method can be seen in the Figure (3.3).

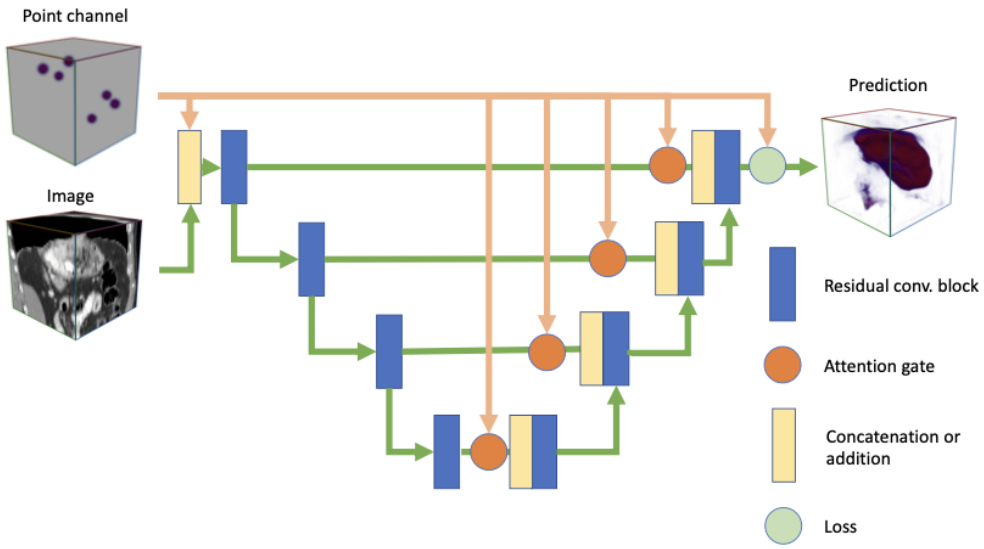

Figure 3.3: Overview of the weakly-supervised segmentation method utilising extreme points proposed by Roth et al. [33].

Their method used U-Net like architecture with residual blocks that consisted of two 3D convolutions followed by group normalisation and ReLU activation function. The authors argue that the use of group normalisation is beneficial in cases

where the network is trained with a lower batch size [33]. The authors also included attention gates in the decoder part of the network, in which they also utilised the extreme points annotation to further guide the network. The network was trained by using a combination of the dice loss with point loss, which was defined to penalise the distance between the boundary of the predicted segmentation with the location of the extreme points [33].

They evaluated their model on six segmentation datasets on which they achieved similar performance to the fully-supervised approaches.

3.2.3 Deep learning with mixed supervision for brain tumor segmentation [29]

Mlynarski et al. [29] proposed a method for training segmentation models with a limited number of annotated brain MRI scans. In their work, they used both fully annotated and weakly annotated images (image-level tags indicating whether the image contains a tumour) during training. For training and testing of their method, the authors used the BraTS dataset, which contains volumetric brain MRI scans in 4 different sequences, however, the authors decided to use only the individual slices to demonstrate the use of weakly annotated images since in the BraTS dataset all of the scans contain the tumour.

The main idea behind their approach was to use a U-Net based architecture with added sub-network for image classification, where both of these networks are trained jointly. The objective of the proposed architecture was to leverage the information contained in weakly annotated images using the classification network while keeping the training process supervised using fully annotated images, which helped to learn features that are beneficial for the segmentation task [29].

The architecture of the segmentation sub-network complies with the standard

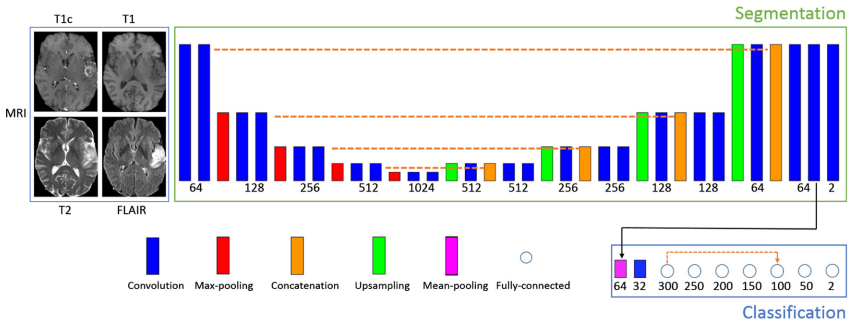

Figure 3.4: Architecture of the network proposed by Mlynarski et al. [29].

U-Net architecture. The second to last convolutional layer in the segmentation network is used as an input to the classification sub-network. The classification network consists of one mean pooling layer that reduces the size of the input feature maps, one convolutional layer and 7 fully connected layers that outputs the class (with or without tumour) of the input image [29].

Both networks were trained with the cross-entropy loss function, where the segmentation loss was also additionally weighted for individual labels in order to deal with class imbalance [29]. The total loss was therefore defined as a combination of these two losses.

Based on the author's evaluation, models trained with both the fully and weakly annotated images achieved significantly better results compared to the models trained with only a limited number of fully annotated images [29].

3.2.4 Click-supervised multiclass segmentation

In the context of multiclass segmentation, En et al. [36] proposed a method which was able to yield accurate segmentations by requiring only one click per class (organ). Similarly to other works the authors also used U-Net like architecture. An overview of their method can be seen in the Figure (3.5).

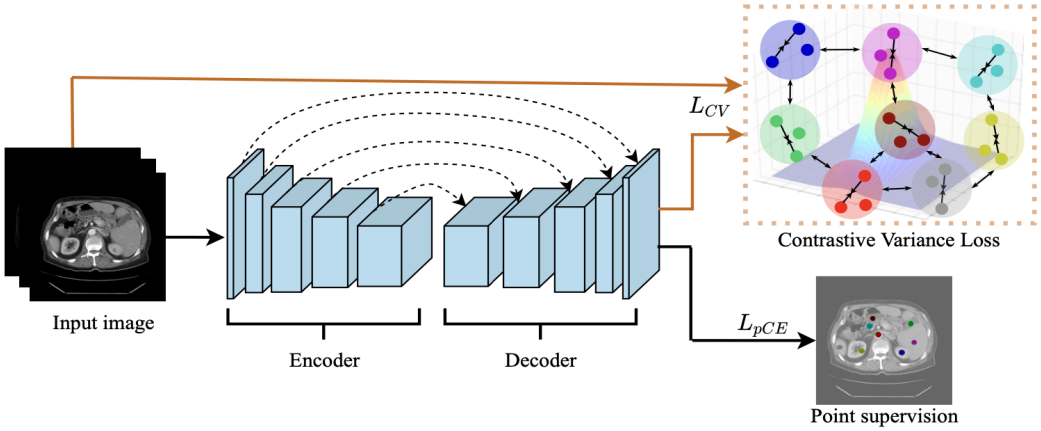

Figure 3.5: Click-supervised multiclass segmentation method proposed by [36].

The most interesting part of their approach, however, is the used loss function. The network was trained by using a combination of the cross-entropy loss and a contrastive variance loss that leverages contrastive variance, which can differentiate the spatial distribution properties of the different organs in the scan [36].

The cross-entropy loss is computed from the point annotations and the predicted segmentation. On the other hand, the contrastive variance loss is computed from the input images and the final segmentation, so it utilises the unlabelled pixels [36]. Specifically, they calculate the variance distribution map for each class, which is used as the appearance representation of that class and can therefore be compared with the same class from different image.

Based on their experiments on multiple datasets their method also outperforms state-of-the-art methods, where it achieved 0.841 average dice score on the ACDC Dataset, which contains 100 cardiac MRI scans [36].

3.3 Interactive segmentation and correction-based approaches

In this chapter, we explored interactive segmentation methods and approaches that generate the final segmentation by iteratively correcting previous rough predictions. Most of these methods utilise used-defined clicks to refine and enhance the accuracy of segmentation results.

3.3.1 Efficient Mask Correction for Click-Based Interactive Image Segmentation [63]

Du et al. [63] in their work proposed an approach for interactive segmentation which utilised a lightweight mask correction network for further refinement of the predicted segmentation. Overall their method consisted of two main parts, a base segmentation network and a mask correction network. A diagram depicting their method can be seen in the Figure (3.6).

The segmentation network was used to generate rough segmentation of the object and extracted target-aware features. The input for the network consisted of the image and map of the first click. For encoding of the input, the authors used the same method which was proposed by Sofiiuk et al. [64] and consists of first adjusting the channels of both components and subsequently summing them element-wise.

In the second part, the correction network updates the generated mask whenever a new click is made. It takes as input the original image, the new click's map, the previous mask and the extracted features. To effectively exploit the information from the clicks, the network incorporates a click-guided self-attention and a correlation module [63]. Firstly the network maps the encoded click on the fea-

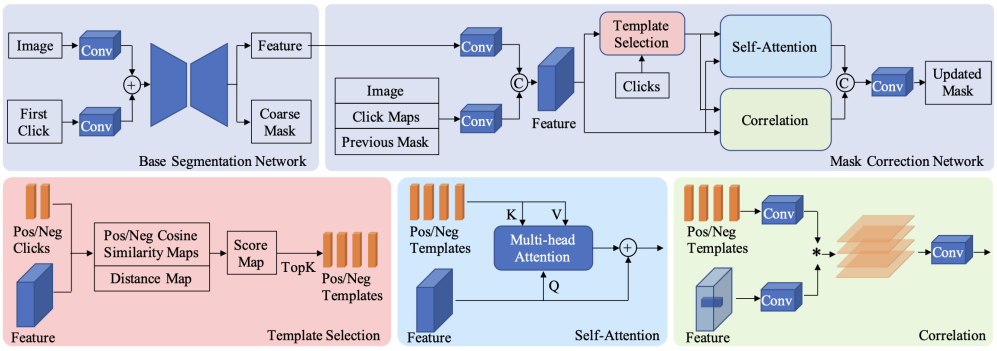

Figure 3.6: Overview of the correction method proposed by Du et al. [63].

ture maps from which they select couple template features. These templates are selected based on the similarity of pixels to the click features. The selected templates are further used in the self-attention module, which consists of multi-head attention that helps with the identification of the target. The templates are also used in the pixel-wise correlation module that aids with denoting the outlines of the object and therefore, improves the segmentation [63].

During testing, the authors measured the number of clicks that were required by the annotator in order to achieve the desirable quality of the final segmentation. Their method achieved comparable results with state-of-the-art methods while being less computationally demanding [63].

3.3.2 FocalClick: Towards Practical Interactive Image Segmentation [65]

Both the method proposed by Du et al. [63] and the approach used in this method share a similar architecture for generating rough segmentations. The main difference is that the approach by Chen et al. [65] utilised smaller crops for both the initial correction and subsequent refinement of the segmentation. These crops were selected from both the original image and the rough segmentation based on

the selected clicks. In particular the cropping of the rough segmentation, "Focus Crop", was done by calculating the "max connected region" from the difference of the rough segmentation and the previous mask [65].

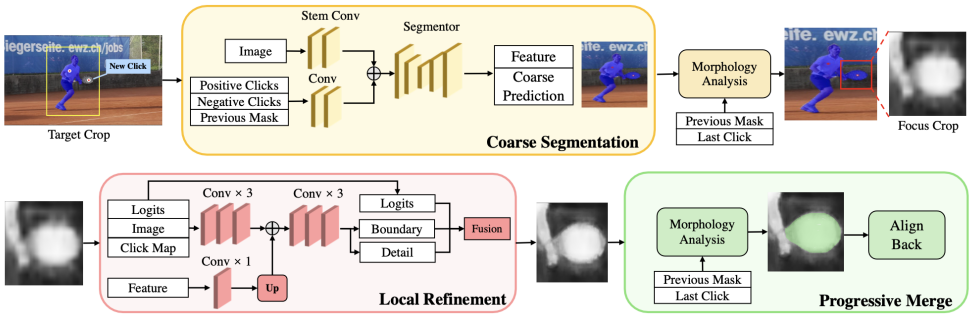

Figure 3.7: Overview of the correction method proposed by Chen et al. [65].

For the refinement of the selected crops, the authors utilised the Xception architecture [66] to predict a detail map and a boundary map from which they calculated the improved segmentation [65].

The authors used a combination of the binary cross-entropy loss with a normalised focal loss (NFL) [67] and a boundary-weighted NFL to supervise the training of the network.

Similarly to the method proposed by Du et al. [63] as an evaluation the authors measured the total number of clicks required to achieve segmentation of desirable quality. They tested their method on popular image datasets such as Berkeley and DAVIS [68, 69] and achieved state-of-the-art results. As the authors mention, however, the method is insufficient for segmenting tiny structures [65].

3.3.3 Cascade-Forward Refinement with Iterative Click Loss for Interactive Image Segmentation [70]

In this work, the authors proposed an interactive segmentation approach that involves two loops; an outer loop for user interaction and an inner loop for segmentation refinement. The outer loop is used to generate rough segmentations based on the user-defined clicks. On the other hand, the inner loop iteratively refines the segmentation by repeatedly feeding the segmentation model with the same input image and user clicks. The inner loop is terminated when the number of changed pixels between the current and previous masks falls below a set threshold [70]. An overview of the proposed method is provided in Figure (3.8).

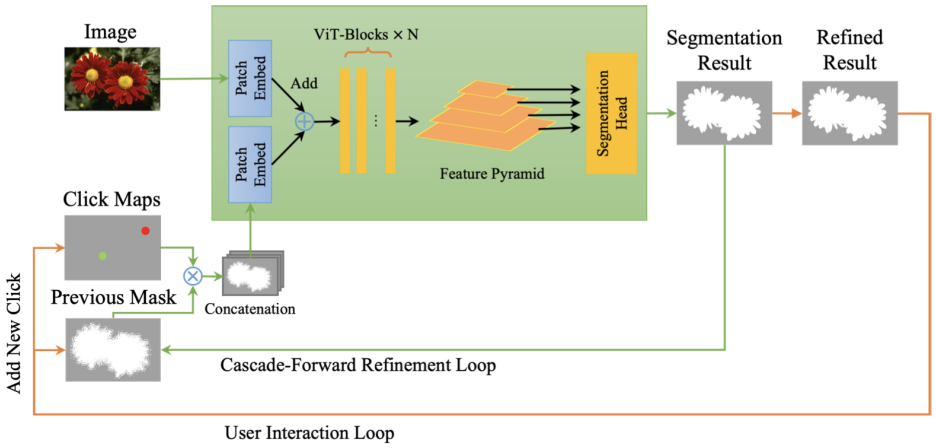

Figure 3.8: Overview of the correction method proposed by Sun et al. [70].

The authors used a state-of-the-art segmentation model to generate rough segmentations. Specifically, they experimented with SimpleClick's ViT-Base and ViT-Huge models [71], based on vision transformers, and the HRNet [72].

The network was trained with Iterative Click Loss (ICL). The loss was defined as the normalised focal loss (NFL) between the ground truth and the predicted segmentation, where in order to minimise the required amount of clicks the loss

was weighted by a weighting parameter β which increased with higher number of clicks.

The proposed method yields better performance than the current state-of-the-art models by reducing the number of required clicks (by 33.2% on the Berkeley and 15.5% on the DAVIS datasets) for 0.95 IoU [70].

3.4 Summary of related work

In this chapter, we explored various methods for segmentation of medical images. In Section 3.1 we analysed different segmentation approaches suited for the segmentation from unbalanced datasets. Unsurprisingly all of the analysed works based the architecture used in their methods around U-Net with a couple of modifications, namely the addition of attention modules as used in [17, 18]. The use of the attention mechanism in the decoder part of the network seems to be especially beneficial in cases where the segmented object is very small. In addition, in [17] the authors also used residual blocks instead of the standard convolutional blocks. As their loss function, most of the works also utilised a combination of dice loss with some kind of specialised loss function. In Section 3.1.3 we analysed two segmentation methods which utilised multi-encoder architecture to provide the network with additional information in form of different modalities. The different modalities utilised separated encoders, where this design helped the network to better extract information from the provided modalities.

In Section 3.2 we particularly focused on approaches which utilised some kind of weak supervision for the segmentation of medical images. Some of the works used the obtained labels (points or scribbles) to generate pseudo-masks, which were subsequently used to train the network [61, 62, 41, 73]. In particular, in [62] the authors used scribbles to generate pseudo-masks, which they iteratively refined by

combining them with the predicted segmentation. In other cases, the point labels were directly utilised in the loss function. In [33], the authors used a combination of the dice loss function with a specialised point loss, which penalised the network based on the distance between the generated segmentations and specified extreme points. Similarly, most of the works in this section also utilised U-Net like architecture.

In Section 3.3 we analysed methods which focused on refining the rough segmentation predicted by a segmentation network. We focused primarily on methods which utilise clicks, as that is the subject of our work, but there are also methods that use other types of weakly-annotated data such as the method proposed by Ibrahim et al. [74], which utilised bounding boxes. Most of the works we analysed used some kind of additional correction module to iteratively refine the rough segmentation from the base segmentation network. In both approaches by Du et al. [63] and Chen et al. [65] the authors utilised a similar strategy for generating the initial segmentation, but the former opted for refining selected local crops rather than improving the whole volume. In contrast, the method by Sun et al. [70] did not utilise any additional modules, where their method relied on repeated forwarding of the mask and on a customised normalised focal loss which penalised a larger number of clicks. Additionally, visual transformers can be also used for interactive image segmentation as they proved to be also usable in this field [70, 75, 76].

Chapter 4

Our work

The main objective of our work was to develop an interactive segmentation method, which utilised weakly annotated data. We decided to adopt a similar approach to some of the works analysed in Section 3.3, which were based on an auxiliary correction network that refined the rough segmentations predicted from a segmentation network.

In general, our work can be split into two main parts. In the first part, we implemented a simple segmentation model, which was trained in a fully-supervised manner on a small portion of the dataset. In the second part, we trained a correction model to correct local errors from small cuts. These cuts are extracted from the initial segmentation based on user-defined clicks. To make the training of the correction network feasible, we also developed a method to simulate user clicks and to generate erroneous segmentations. Lastly, we employed the pre-trained correction network to refine the wrongly segmented regions from the segmentation network. A diagram describing our method can be seen in Figure (4.1).

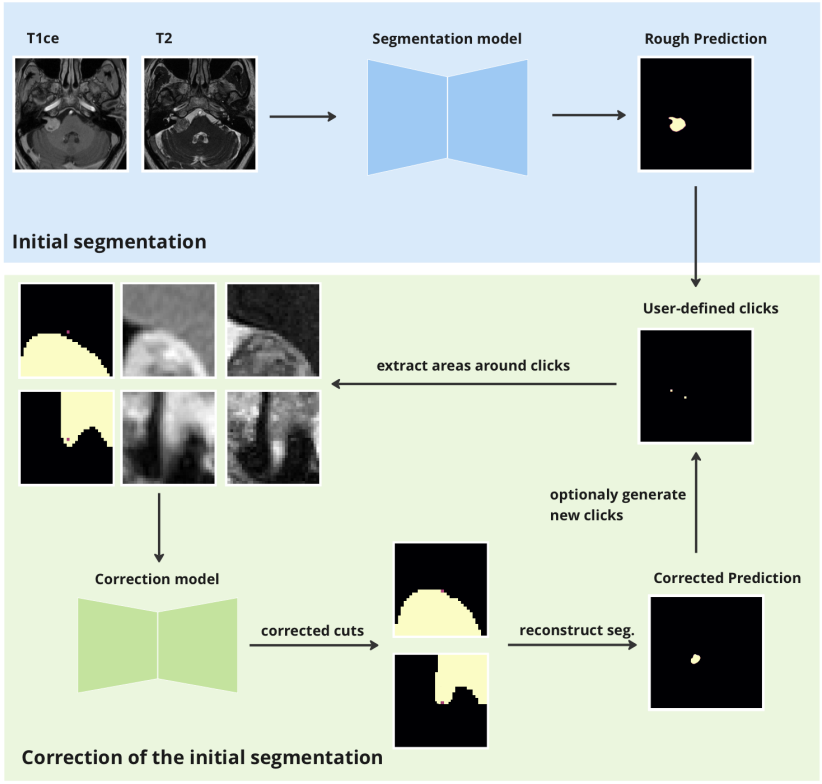

Figure 4.1: Diagram of our proposed method consisting of two main parts. First part generates initial segmentation and the second part employs a auxiliary correction network which refines the segmentation based on user-defined clicks.

4.1 Dataset

In our work we utilise the publicly available Vestibular Schwannoma (VS) dataset[1], which consists of 242 brain MRI scans in T1ce and T2 sequences [3]. Additionally to the scans, the dataset also includes annotations of the VS for each scan. An example of one slice in each sequence and the corresponding annotations can be seen in Figure (4.2). Originally the scans come in the DICOM format and the two sequences are not co-registered, where the T1ce sequence usually has more slices than the T2. The curators of the dataset provide pre-processing scripts[2] to

[1] https://www.cancerimagingarchive.net/collection/vestibular-schwannoma-seg/
[2] https://github.com/KCL-BMEIS/VS_Seg/tree/master/preprocessing

convert the data into the NIfTI format and co-register the sequences.

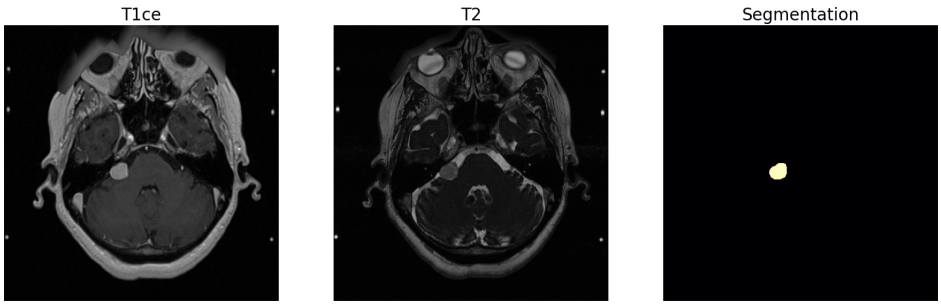

Figure 4.2: Sample T1ce, T2 sequences and the corresponding Vestibular Schwannoma segmentation from the dataset.

While analysing the dataset, we came across a couple of issues that needed to be addressed. First of all, we found out that the final volumes come in different shapes. In particular, there are 191 scans with shape 448x448x80, 48 scans with shape 384x384x40, one with 384x384x80, one with 448x448x70, and one with 384x384x20. Another issue we came across is that the data are highly unbalanced, where the tumour voxels only take up 0.006 of the total space.

4.2 Data preprocessing

As part of data pre-processing, we cropped and resized the sequences from their original shapes to 40x256x256 to deal with the shape variety and also to make training feasible due to high memory requirements. The cropping removed most of the background from the scans and we also decided to remove some slices from the sequences since the average depth of the tumour was just around 10 slices with the maximal depth being 20. The depth cropping was done in such a way that the tumour was somewhere in the middle of the new volume. However, one scan has a depth smaller (20) than the selected depth (40), where in this case we padded the first and last 10 slices with zeros.

The adjusted sequences were further normalised using a Min-Max scaler and stacked together, so the final tensors had a shape of `(b,2,40,256,256)` for the input images and `(b,2,40,256,256)` for segmentation masks, where `b` represents the batch size. An example of the preprocessed sequences and the corresponding segmentation can be seen in Figure (4.3).

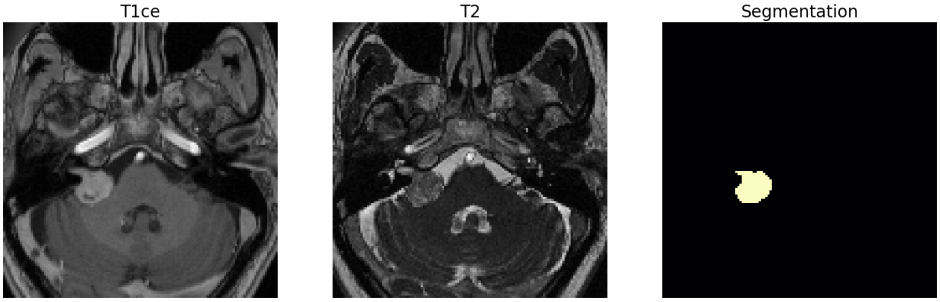

Figure 4.3: An example of the preprocessed sequences and segmentation.

Additionally, to verify that preprocessing did not intervene with the tumour, we summed all preprocessed segmentations. The visualisation can be seen in the Figure (4.4), where we verified that the tumours remained intact.

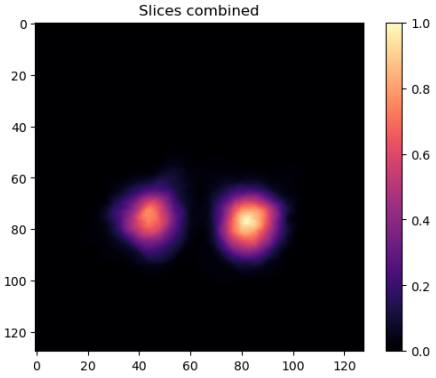

Figure 4.4: Combined segmentations.

4.3 Proposed method

As we have already mentioned, our method consists of two main parts, the generation of an initial segmentation and the subsequent correction of this segmentation in the second part, as can be seen in the diagram in Figure (4.1).

The first part is composed of a pre-trained segmentation model that generates initial rough segmentations. In this part, we experimented with different loss functions, in order to find a loss function that would allow us to use as little fully-annotated images as possible, while producing segmentations that are still satisfactory.

In the second part, the initial segmentation is cut on the basis of user-defined clicks into small cutouts. These are then sent to the pre-trained auxiliary correction model that refines them. For the training of the correction network we used erroneous segmentation cuts that we generated from the ground-truth labels. In this part, we experimented with different network architectures, loss functions, and different modifications of the generated training cuts.

Finally, the extracted cuts are reconstructed back into the initial volume, where the user can optionally generate new clicks to further improve the segmentation.

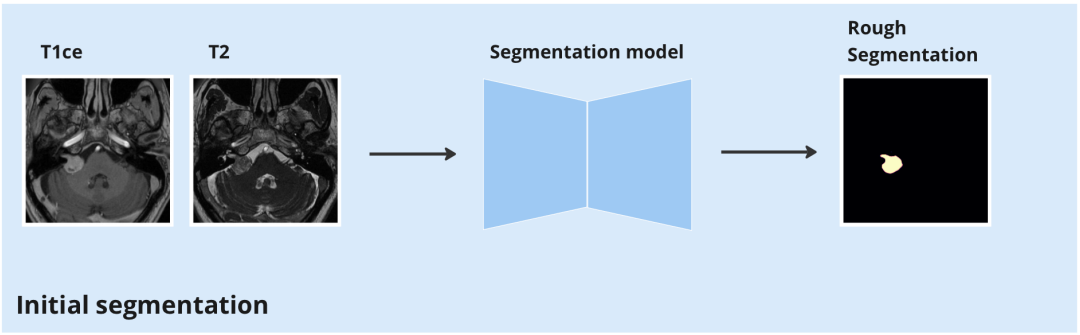

Figure 4.5: Generation of the initial segmentation.

4.4 Initial segmentation model

In this part we designed a method for segmentation of the vestibular schwannoma from both available MRI sequences. The model was trained in a fully-supervised manner, where the purpose of the segmentation model was to generate initial segmentations. The general process for this part is shown in Figure (4.5).

4.4.1 Architecture

We chose to use the 3D U-Net architecture [9], as it has proven to be very successful in this domain [17, 18, 19]. A diagram showing our architecture can be seen in the Figure (4.6).

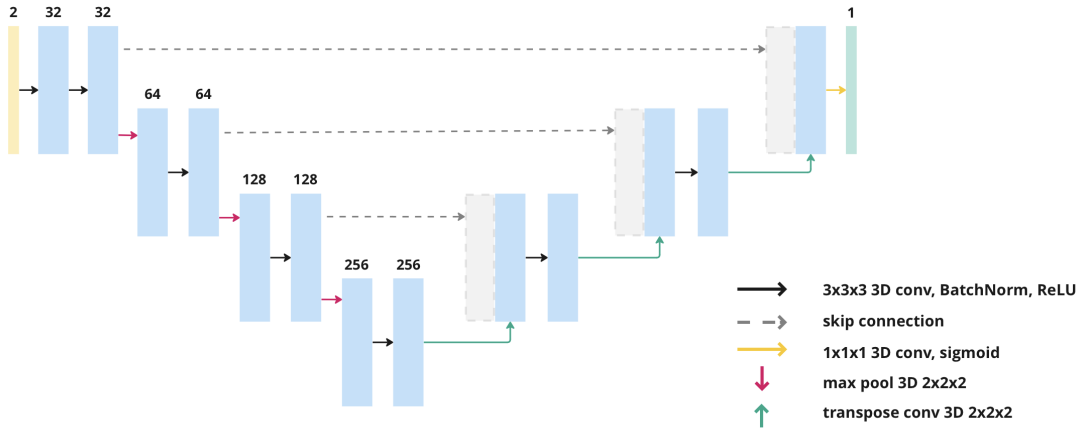

Figure 4.6: U-Net architecture of the segmentation model.

The layout of our architecture follows the standard U-Net, with three stages in both the encoder and the decoder. Each stage consists of two convolutional layers with 3x3x3 3D convolutions and zero padding followed by BatchNorm and the ReLU activation function.

The downsampling operation is performed by 3D max-pooling layers with pooling size 2x2x2 and stride 2. On the contrary, the upsampling of the feature maps is

achieved by transposed 3D convolutions with kernel size 2x2x2, stride 2 and zero padding.

The input of the network has 2 channels and consists of stacked T1ce and T2 sequences. Subsequently, the channels of the feature maps double through the downsampling path, starting from 32 and ending with 256 in the bottleneck. In the upsampling path, the number of feature channels is halved. Since we are performing binary segmentation, the last layer is a 1x1x1 convolution with one channel followed by the sigmoid activation function, which outputs the segmentation.

4.4.2 Training of the segmentation model

For model training, we randomly split the data into three sets, where 170 samples were in the training set and 36 samples were in the validation set. The remaining 36 images were left for testing of our models. We used both T1ce and T2 sequences for training of the network. These sequences were stacked together and used as input to the segmentation network.

We trained the baseline model for 50 epochs with batch size 2 and Adam optimiser with an initial learning rate set to 0.001, which decreased on plateau by a factor of 0.5.

As for the loss function, we decided to base our experiments in this part of the work around finding the most suitable loss function for this task. We in particular paid attention to loss functions that are suitable for segmentation from unbalanced datasets.

For monitoring the training process, as well as for the final evaluation of the model, we decided to use the dice score coefficient (DSC), which measures the overlap between the computed segmentation and the ground truth. The equation

describing the DSC can be found in the analytical part of the document (eq. 2.3).

4.4.3 Results

Since the VS dataset is fairly unbalanced, we decided to first experiment with multiple loss functions to find the one most suitable for our task. In this experiment, we used the whole training dataset. The results from both training and testing can be seen in Table (4.1).

Overall we experimented with weighted binary cross-entropy (bce), dice loss, the combination of dice loss and bce, focal loss, tversky loss and the combination of focal and tversky loss.

	Training	Validation	Testing
Weighted binary cross-entropy	0.7546	0.7232	0.7091
Dice loss	0.8720	0.8408	0.8329
Combination of dice loss and bce	0.8901	0.8378	0.8432
Tversky loss	0.8667	0.8248	0.8150
FocalTversky loss	**0.9243**	**0.8838**	**0.8702**

Table 4.1: Training, validation and testing DSC for models trained with different loss functions.

The **FocalTversky loss** performed the best in this experiment, where the model trained with this loss function achieved a test DSC of **0.8702**. Surprisingly, the Dice loss function also performed relatively well and achieved the test DSC of 0.8329. An example segmentation by the model trained with FocalTversky loss can be seen in Figure (4.7).

In our case, however, we rather wanted the model to use fewer fully-annotated images, so in another experiment we employed the two best performing loss functions on smaller training sizes. The purpose of this experiment was to find the

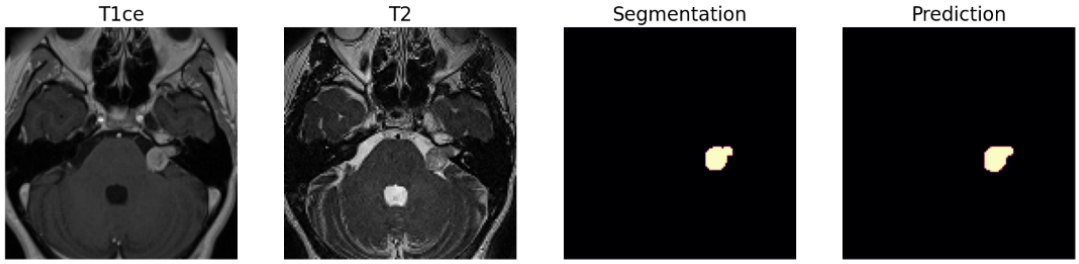

Figure 4.7: An example segmentation from the segmentation model. On this particular scan the model achieved DSC of 0.9034.

least amount of data that we can use for the model to produce segmentations that are still satisfactory (above 0.75 DSC).

	Training Size	Training	Validation	Testing
Dice loss	32	0.9071	0.6553	0.6313
FocalTversky loss	32	0.8895	0.7125	0.7032
Dice loss	64	0.9209	0.7765	0.7783
FocalTversky loss	64	0.9204	0.8017	0.8030
Dice loss	128	0.9291	0.8607	0.8643
FocalTversky loss	**128**	**0.9333**	**0.8736**	**0.8612**

Table 4.2: Training, validation and testing DSC for models trained with Dice and FocalTversky loss functions on different training sizes.

As can be seen in Table (4.2), the best model was trained using the FocalTversky loss on a training set of 128 images. We decided that 128 images represented the maximum number of images we wanted to use, as this already exceeded 75% of our training set. Graphs showing the training of these models are provided in the Appendix **??**.

Models trained on a smaller amount of training data were visibly prone to overfitting, as indicated by the high differences between training and validation DSC. This did not pose a problem for us, as the use case of our method is to use an additional network to correct inaccurate segmentations. With an overfitted model, we can more accurately test the correction network capabilities. As a result, we

decided to use the model trained on 64 images and with the FocalTversky loss function as **the segmentation model**.

4.5 Correction model

In the second part of our work, we implemented a correction network, which was designed to correct the segmentation errors of the given segmentation mask. The process of correcting the initial segmentation can be seen in Figure (4.8).

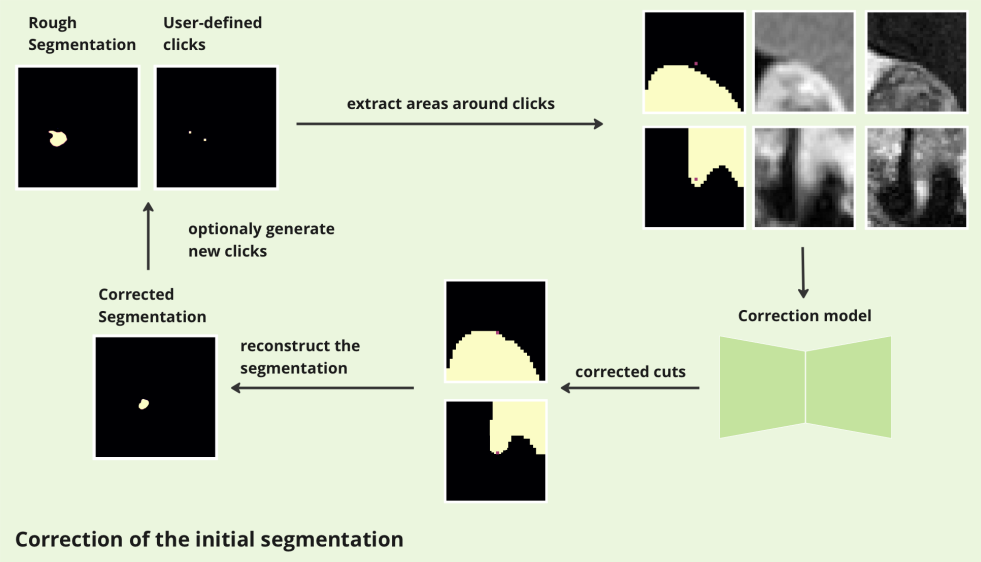

Figure 4.8: Diagram depicting the correction of the initial segmentation.

Our strategy for this part involved extracting small areas from the segmentation and training the network to correct the defects in these small cutouts. This approach allowed the network to focus solely on the local area of the faulty segmentation, which proved beneficial for the networks overall performance.

These cutouts are generated after the user clicks on an area of the segmentation that they want to correct. After that, a small area around the click is extracted and sent to the correction network.

4.5.1 Training of the correction model

For the training of the network to be feasible, we first needed to design a method to simulate user clicks and to automatically generate faulty segmentations. The process of training the correction network is depicted in Figure (4.9). The process consists of utilising the preprocessed data to generate a set of training cuts and true cuts. The training cuts are used as input to the network, while the true cuts are used to evaluate corrected cuts returned by the network.

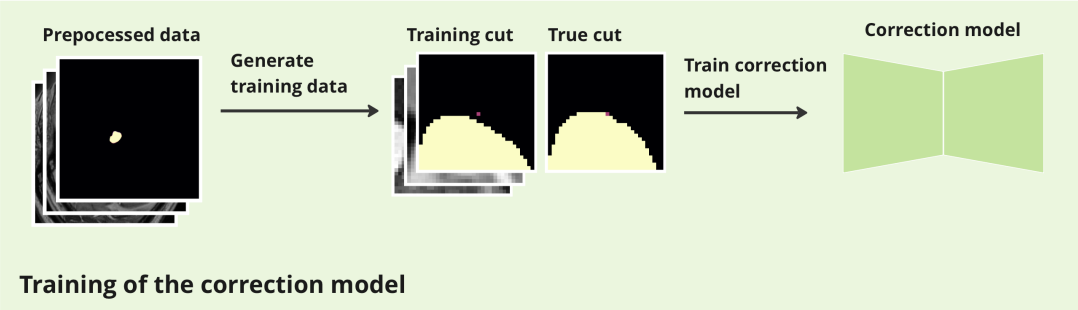

Figure 4.9: Diagram depicting the training of the correction network.

4.5.2 Generating training data

The generation of data for the training of the correction network consists of four main steps, which can be seen in the diagram in Figure (4.10). In the first place, the input volumes are preprocessed, where we make use of the preprocessing pipeline as described in Section 4.2.

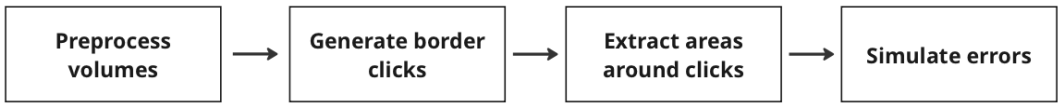

Figure 4.10: Diagram of the pipeline for generating training data.

The preprocessed volumes are used to generate border clicks (the process of gen-

erating border clicks is described in 4.5.2.1). These clicks are then used to cut an area from both the MRI sequences and the segmentation mask. The cut is made around the click, with the click being in the middle of the acquired cutout. We tried different sizes of the cutouts, from which cuts with size 48 (width and height) performed the best (Table 4.4). Afterwards, we create copies of the cuts and modify them to resemble faulty segmentations (the whole process is described in 4.5.2.2). These unmodified and erroneous cuts form the dataset used for training of the correction network.

4.5.2.1 Generating clicks

The clicks are generated from the provided segmentation mask by first using morphological erosion to get a border around the segmented object (the original segmentation - erosion). Subsequently, a set number of points are randomly selected from the acquired border. To ensure that the points are not generated too close to each other, they are selected based on the distance between them. This distance is calculated using the Euclidean distance.

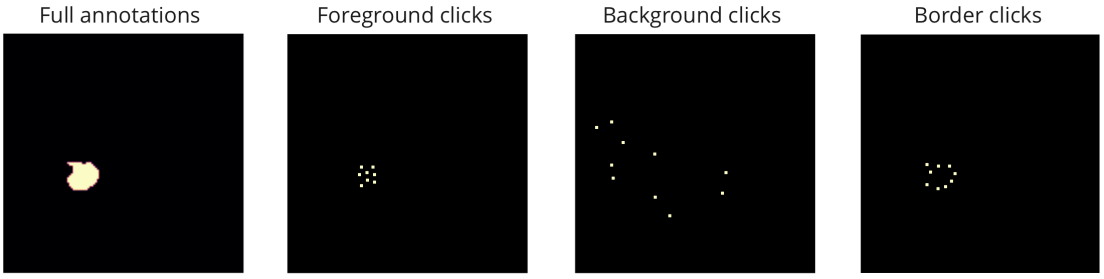

Figure 4.11: An example of generated background, foreground and border clicks.

Similarly, we can also generate other types of clicks, such as background (dilatation - the original segmentation) and foreground (selected from the original segmentation). An example of generated clicks can be seen in Figure (4.11).

4.5.2.2 Simulating errors

For simulating segmentation errors, we applied morphological dilatation and erosion on acquired segmentation cutouts. To mimic oversegmentation, we used dilation; on the other hand, to simulate insufficient segmentation, we used erosion. To prevent the destruction of the segmentation, we reduced the number of iterations for erosion when applied on smaller segmentations. An example of generated erroneous segmentations can be seen in Figure (4.12).

Figure 4.12: An example of generated cuts (top row) and cuts with generated errors (bottom row).

To ensure sparsity between the generated errors, the morphological operations were selected randomly. As a form of regularisation we also left some of the cuts unchanged, where with 20% probability, the error generation would be skipped for the given cut.

4.5.3 Simulating user interaction

In order to make the testing of the correction network faster, we developed a method to automatically generate clicks in the areas of the segmentation that need to be refined. The clicks are generally added in the area where there is the biggest

difference between the original (ground truth) segmentation and the predicted segmentation. The process can be described by the following pseudocode:

Algorithm 1: Simulating user clicks

Data: y_true, y_pred
Result: $clicks$
$border \leftarrow y\_true - errosion(y\_true)$;
if $y\_true[y\_true == 1] > y\_pred[y\_pred == 1]$ **then**
  $dst \leftarrow euclidean\_distance\_transform(1 - y\_pred)$;
else
  $dst \leftarrow euclidean\_distance\_transform(y\_pred)$;
end
$weighted\_border \leftarrow dst \times border$;
$click\_coordinates \leftarrow where(weighted\_border == weighted\_border.max())$;
$clicks \leftarrow select\_clicks(click\_coordinates)$;

4.5.4 Loss function

For this task, we also designed a simple loss function. The loss consisted of a dice coefficient that is scaled by a weight map (w). The weight map ensures that the middle pixels have higher weights, so the network would therefore be more penalised for segmenting them incorrectly. Since the cuts are made around the middle pixel, weighting the dice coefficient in this way should improve the accuracy of the final segmentation compared to normal dice loss. This "correction" loss (L_{corr}) can be defined as:

$$L_{corr} = 1 - 2\frac{\sum_{i=1}^{W}\sum_{j=1}^{H}(p_{i,j} \cdot y_{i,j} \cdot w_{i,j})}{\sum_{i=1}^{W}\sum_{j=1}^{H}((p_{i,j} + y_{i,j}) \cdot w_{i,j})} \tag{4.1}$$

where the numerator computes the intersection between the two volumes, and the denominator computes the union. Both intersection and union are performed by multiplying or summing the volumes along the spatial dimensions.

4.5.5 Architecture

For the correction network, we chose to use a 2D Attention U-Net with attention blocks in the decoder. The use of the attention mechanism proved to be beneficial for the performance of the network (as can be seen in Table 4.3). The architecture of the used network is shown in Figure (4.13).

Apart from the additional attention blocks, the network follows the standard U-Net architecture. In this case, we used 3-channel input, which consisted of stacked cuts of the T1ce and T2 sequences and the incorrect segmentation. The output of the network has 1 channel and represents the corrected segmentation cut.

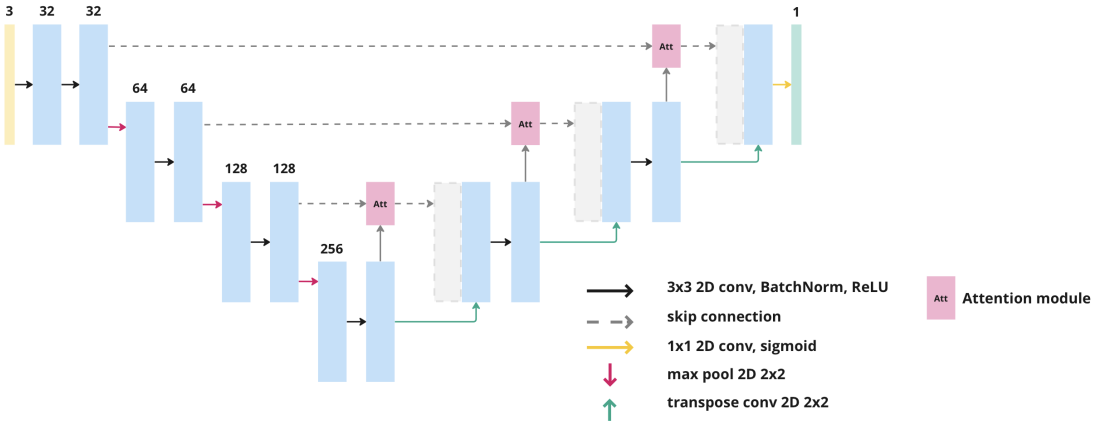

Figure 4.13: Attention U-Net architecture used for the correction model.

For the attention mechanism, we used spatial attention as described by Oktay et al. [14] and its architecture can be seen in Figure (4.14).

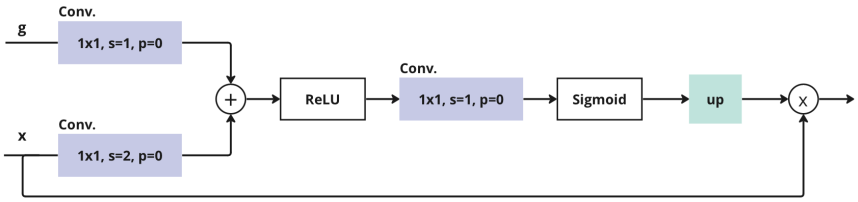

Figure 4.14: Architecture of the attention block used in the correction model.

The first two convolutions in the attention module adjust the number of channels of both inputs. The `x` convolution also downsamples the input since the skip connection has bigger dimensions than the gating signal. These adjusted feature maps are then added together and run through the ReLU activation function. Subsequently, we apply another convolution, which reduces the number of channels to 1 and scales the output values to a range between 1 and 0 using the Sigmoid function. The resulting coefficients are upsampled to match the dimensions of the skip connection and then multiplied by the input skip connection.

To help with regularisation, we also added dropout layers with 40 % change after each ReLU in the convolutional blocks.

4.5.6 Training setup

For the training of the correction network we kept the same data split as for the training of the segmentation network (i.e., 170 for training, 36 for validation and 36 for testing). In this part, we used the incorrect segmentation and both T1ce and T2 sequences as input to the network. For generating the dataset, we used all of the training and validation scans. Since the scans are volumetric and there can be multiple clicks on each slice, the final number of training images was; 8128 for training and 2212 for validation.

All experiments were trained for 50 epochs with a batch size of 4. We used the same optimiser settings as in the segmentation network training (4.4.2).

In this part of the work we performed first correction trainings and experimented with data generator settings, where we tried different cut sizes. Furthermore, we also tried to verify the benefits of the attention blocks. In addition, we performed experiments on both dice and correction loss functions.

4.5.7 Testing of the correction network

The testing of the trained correction models was done on cuts created from scans in the testing set. At first, we generated segmentation for the selected scan. For this, we utilised the trained segmentation model. First, we generate clicks for the acquired prediction (explained in 4.5.3). Then we used the clicks to cut the generated segmentation and MRI sequences. These acquired cuts represent the test dataset.

To ensure that all models will have the same conditions, we set a custom seed for the click selections, so it will always return the same set of clicks.

4.5.8 Results

At first we wanted to confirm that the use of spatial attention and redefined dice loss have beneficial effects on the training of the model. For this we conducted 4 trainings with alternating the dice and correction losses as well as architecture with or without the spatial attention. The results of these experiments are shown in Table 4.3. For these experiments, we used the same cut size, which was set to 48.

	Testing
dice loss, attention blocks	0.8328
correction loss, attention blocks	**0.8417**
dice loss	0.8072
correction loss	0.8251

Table 4.3: Testing DSC for models with / without the attention blocks and trained on either dice or attention losses.

We were able to confirm that the inclusion of the attention blocks had a positive effect on the performance of the model, where the testing DSC slightly improved in both cases. Similarly, the correction loss also performed better than the basic

dice loss function. The best model was therefore trained with these settings and achieved testing DSC of 0.8417.

In the following experiments, we tried to use different cut sizes to find the most suitable one. We experimented with four different cut sizes; 16, 32, 40 and 48. We did not include cutouts with sizes over 48 as then the cutout would not capture the local area of the tumour, as the cutout would usually include the whole tumour. Testing DSC for different cut sizes can be seen in Table 4.4.

	Testing
correction loss, 16-sized cuts	0.7042
correction loss, 32-sized cuts	0.8042
correction loss, 40-sized cuts	0.8350
correction loss, 48-sized cuts	**0.8417**

Table 4.4: Testing DSC for models trained on cutouts of different sizes.

Based on the results, the model trained on 48-sized cuts performed the best. Overall, the performance was improving with larger cut sizes. The model trained on 16-sized cuts performed significantly worse, which can be attributed to the fact that the sequence cutouts were, in this case, too small and the details were too distorted.

With these experiments, we were able to find the base configuration for the correction model. Subsequently, all of the following models were trained with the correction loss function, attention modules, and 48-sized cuts. An example of a corrected segmentation of the best trained model in this stage can be seen in Figure (4.15). On the particular scan, the model achieved a DSC of 0.7228. Compared to the segmentation predicted by the segmentation model (0.5727 DSC) the correction improved the segmentation by 0.1501.

Figure 4.15: An example of corrected cuts produced by the attention U-Net (bottom row) compared to the original segmentation (top row). In this particular case, the corrected cuts achieved DSC of 0.7228.

4.6 Multi-encoder U-Net

From the results of the previous experiments, we hypothesised that the network did not adequately utilise the information from the MRI sequences. In [59, 60], the authors utilised multi-encoder segmentation architectures to better incorporate different modalities into the training of their network.

Inspirited by their work, we decided to try this modification as well, since the additional encoder might help the network to better extract the information from these modalities. In our case, we decided to use two separated encoders. One for the incorrect cut and the other for the stacked MRI sequences.

Apart from the modification of the network architecture, we used the same training process as in the previous experiments. For the training of the network, we utilised the correction loss and 48-sized cuts, as this combination performed the best in previous experiments (Table 4.4).

4.6.1 Architecture

The multi-encoder U-Net architecture mainly differs from the standard U-Net in the downsampling path. In this case, the downsampling path of the network includes a supplementary encoder for the additional modalities. In our case, we decided to use one encoder for the incorrect segmentation cut and one additional encoder for the stacked MRI sequences. The last downsampled feature maps from these two encoders are concatenated before the bottleneck.

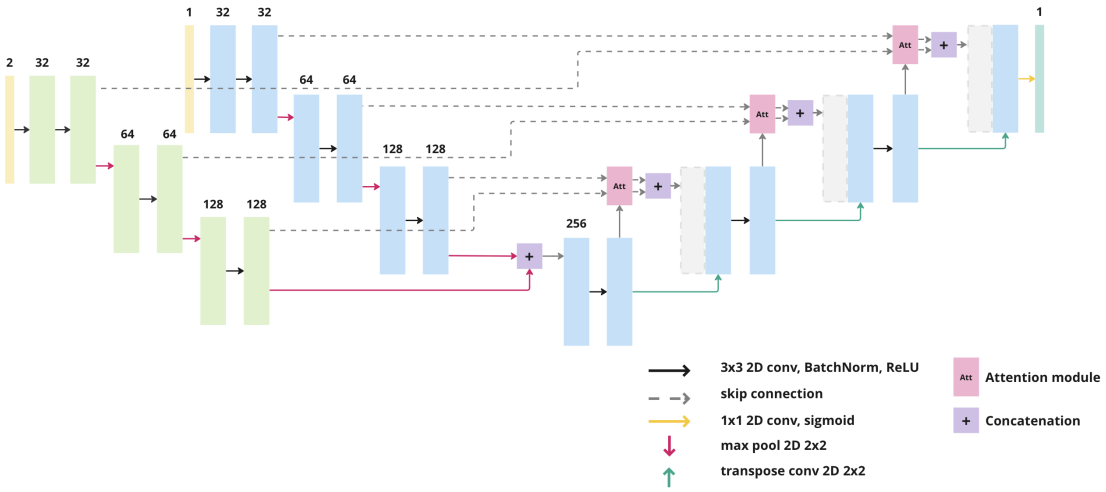

Figure 4.16: Modified correction network architecture with two encoders.

In the upsampling path the skip connections are first inputted into separate attention modules (in the figure shown as attention module with two inputs). Opposed to [60], we only compute spatial attention for each skip connection and concatenate the resulting feature maps. The concatenated feature maps then continue to the corresponding upsampling block, where we first adjust the number of channels before concatenating them with the upsampled feature maps. In this case, we use the same spatial attention module as described 4.6.1.

4.6.2 Results

In this experiment, we wanted to compare whether the use of an additional encoder helped the model to better utilise the information provided from the MRI sequences. The results of this experiment can be seen in Table 4.5. In this case, we compared the trained model with the best model we have trained so far (trained with the correction loss on 48-sized cuts).

	Testing
initial cuts	0.7630
multi-encoder U-Net	**0.8687**

Table 4.5: Testing DSC for the best attention U-Net based model and for the model trained with the multi-encoder U-Net architecture.

As can be seen from the results, the additional encoder slightly improved the models performance and achieved testing DSC of 0.8687, compared to the 0.8417 achieved by the previous best model. A similar improvement can also be seen in the example of corrected cuts in Figure (4.17). Where the correction slightly improved in comparison to the attention U-Net based model (from 0.7228 to 0.7884).

4.7 Using volumetric cuts

Furthermore, to improve the networks performance, we attempted to also utilise volumetric cuts, as we believed that the additional information could be beneficial to the accuracy of the network.

To generate the training cuts, we followed the same pipeline as described in Section 4.5.2. The only difference was that, except for the width and height of the volume, we also cut the depth. With this approach, only one slice of the resulting cutout correctly matched with the provided click. To account for the additional slices, we made some modifications to the correction loss. The errors were generated for

Figure 4.17: An example of corrected cuts produced by the multi-encoder (bottom row) compared to the original segmentation (top row). In this particular case, the corrected cuts achieved DSC of 0.7884.

all of the slices in the given cut. An example of generated volumetric cuts can be seen in Figure (4.18).

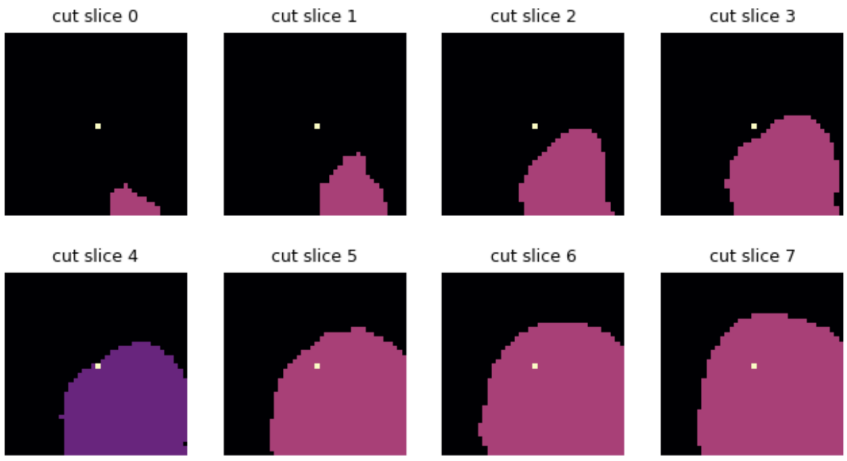

Figure 4.18: Example of generated volumetric cuts with depth=8, where the middle slice is denoted by darker colour.

For training, we used the multi-encoder architecture as described in 4.6.1 and only

changed the dimensions of the individual operations (convolutions, downsampling and upsampling). As for the experiments, we tried to train the network with different cut depths.

4.7.1 Loss function for volumetric cuts

Since the click position only correctly matches with the middle slice of the volume and the additional slices were used as supplementary information for the network, the correction loss needed to be adjusted.

For volumetric cuts, the loss function was composed of two components. The first component was the correction loss (defined in Equation 4.1) which was only computed for the middle slice, since that is the volume that needs to be corrected. The other component of the loss was the dice loss (defined in Equation 2.4), which was calculated for the whole volume. The total loss can be therefore defined by the following equation:

$$L_{total} = L_{corr} + \alpha L_{dice} \tag{4.2}$$

where α is a given constant. During experiments, α was set to 0.75 to underline the correction loss.

4.7.2 Results

In this part, we experimented with two different cut depths. In this case, we did not want to go below 8, as in this case, we would need to change the number of downsampling operations in the network architecture, which we view as not beneficial. On the other hand, we also did not want to use a higher depth because the tumour itself is relatively small (on average only occupies 10 slices). The resulting test DSC for the individual cut depths can be seen in Table 4.6.

	Testing
multi-encoder U-Net, volumetric correction loss, 8 cut depth	0.8010
multi-encoder U-Net, volumetric correction loss, 16 cut depth	0.8152
previous best model (multi-encoder U-Net)	**0.8687**

Table 4.6: Testing DSC for models trained on volumetric cuts of different depths and the best multi-encoder model.

From the results, the model trained with cuts of higher depth (16) performed slightly better than the model trained on cuts with depth 8. However, the added additional dimension to the training data did not improve the networks performance, and the best model trained with the volumetric cuts still underperformed in comparison to the model trained on 2D cuts.

4.8 Model fine-tuning

Since the performance of the correction model was still not satisfactory, we decided to fine-tune the best correction model that we had trained so far on real segmentations. We hypothesise that the method, which we used to imitate segmentation errors for training was perhaps too trivial and that the model simply learnt how to revert the used morphological operations, instead of truly correcting the imperfections. In this regard, we decided to additionally fine-tune the model on the segmentations generated by the segmentation model, trained in Section 4.4.

For this, we applied the following pipeline, which can be seen in Figure (4.19).

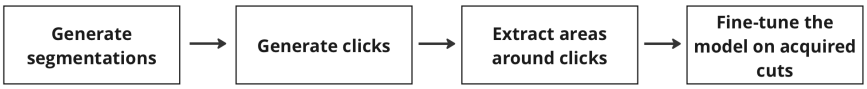

Figure 4.19: Diagram describing the fine-tuning pipeline.

The fine-tuning pipeline consists of 4 main steps. At first, we generate a segmenta-

tion by inputting the given image into the segmentation model. Then we generate clicks that mark the areas where the segmentation is insufficient. The whole process of simulating user clicks is described in Section "Simulating user interaction". After the clicks are generated, we cut the given volume. Here we use the same process as in the generation of training data (Section 4.5.2). Finally, we use the generated cuts to further train the correction model. In the end, we acquired 2375 training cuts and 594 validation cuts.

For the fine-tuning of the model, we used the same optimiser settings and the same overall training process as in the previous experiments.

4.8.1 Results

With this experiment, we wanted to further improve the best multi-encoder model by additionally training it on cuts generated from inaccurate segmentations. The results of this experiment can be seen in Table 4.7, where we compared the fine-tuned model with the best multi-encoder U-Net based model.

	Testing
previous best model (multi-encoder U-Net)	0.8687
fine-tuned multi-encoder U-Net	**0.8938**

Table 4.7: Testing DSC for the fine-tuned model and the "original" model.

As can be seen from the table and the example segmentation in Figure (4.20), fine-tuning has indeed improved the performance of the model, where the DSC for the corrected segmentation improved from 0.7884 to 0.9265.

Figure 4.20: An example of corrected cuts produced by the fine-tuned multi-encoder U-Net (bottom row) compared to the original segmentation (top row). In this particular case, the corrected cuts achieved DSC of 0.9265

4.9 Reconstruction of the segmentation

Final part of our method is the reconstruction of the segmentation, from the corrected cuts. For this we used a simple algorithm that iterated over the provided clicks and replaced the area of the segmentation denoted by the click with the corrected cut. The pseudocode for this algorithm is provided bellow.

Algorithm 2: Reconstruction of the segmentation

Data: pred, click_coords, cuts, cut_size
Result: Reconstructed volume
for $click\_idx$ **to** $len(click\_coords)$ **do**
    $coords \leftarrow click\_coords[click\_idx]$;
    $cut \leftarrow cuts[click\_idx]$;
    $z, y, x \leftarrow coords[0], coords[1], coords[2]$;
    $pred[0, z, y - cut\_size : y + cut\_size, x - cut\_size : x + cut\_size] \leftarrow cut$;
end

4.9.1 Results

In this part, we combined the previously explored methods and tested the full proposed pipeline.

As first, we used the segmentation model to generate an initial segmentation, for which we simulated the users interaction and generated clicks on regions we wanted to refine. Subsequently, these regions were extracted from the initial segmentation and the acquired cuts were sent to the correction model, which corrected them. Finally, we replaced the old regions in the initial segmentation with the corrected cuts, for which we used the method described in this section. In this case, we only generated one set of clicks for each segmentation. The results from the testing are in Table 4.8.

	Testing
initial segmentation model	0.8030
after correction	**0.8630**

Table 4.8: Testing DSC for the whole pipeline.

Additionally, Table 4.9 shows the DSC for the initial segmentation, reconstructed segmentation and the reconstructed segmentation after 2nd iteration (where we generated a new set of clicks) for a selected volume.

	DSC
initial segmentation	0.7342
corrected segmentation	0.8606
corrected segmentation, 2nd iteration	0.8908

Table 4.9: DSC of the initial segmentation and reconstructed segmentation after 1st and 2nd iteration for selected volume.

Figure (4.21) shows the first four slices of the segmented tumour. The whole segmentation masks are in the Appendix **??**.

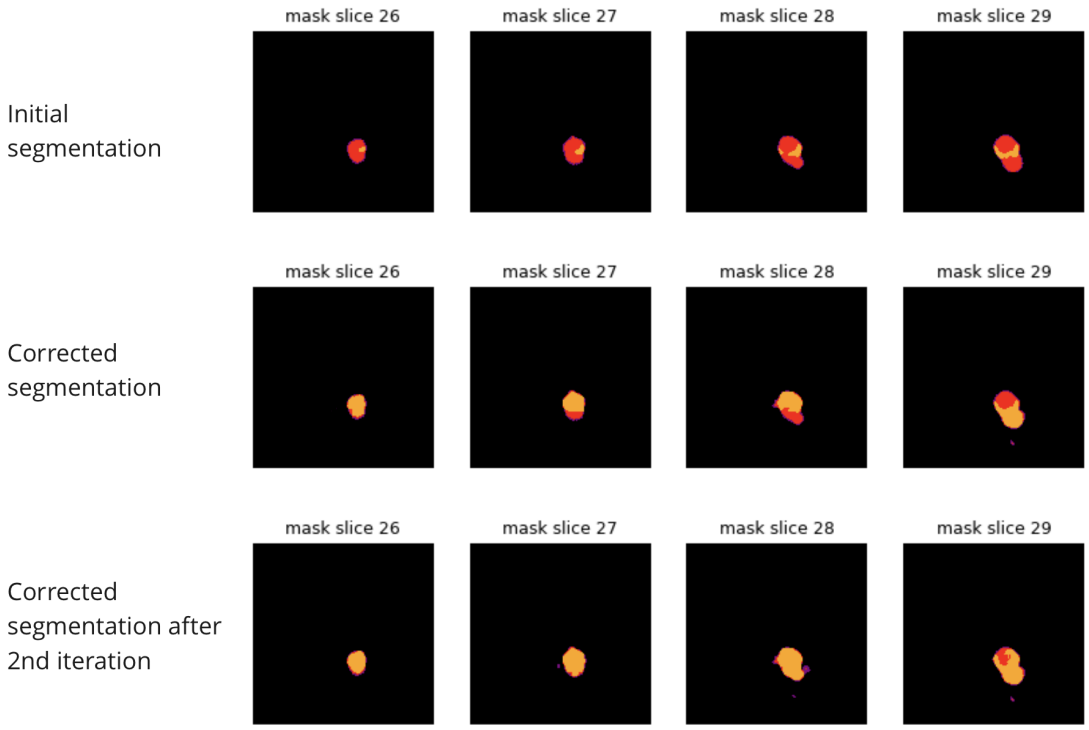

Figure 4.21: First four slices of a sample segmentation. The red colour denotes the ground truth label, the purple the generated segmentation and the orange the overlap between them.

Overall, the proposed reconstruction method is very naive and primarily serves as a demonstration of the process. In the future, it would be more appropriate to replace this method with a more complex process. One of the issues with this method is that it does not take into account the previous segmentations when replacing the cuts. This means that in some cases the correct segmentation might get overwritten by an incorrect one. A similar issue that might occur is that multiple cuts can sometimes involve the same region. Since the cuts are processed sequentially, later cuts might therefore overwrite regions that were corrected by the previous ones.

In this way it could be beneficial to expand the proposed reconstruction method with a strategy to handle the overlapping of certain regions.

Chapter 5

Conclusion

The objective of our work was to design a method that segmented Vestibular Schwannomas from T1ce and T2 MRI sequences, by utilising weakly-annotated data, in our case represented by clicks. The used approach was based on two networks; an initial segmentation network and a correction network. The proposed pipeline therefore consisted of the segmentation network, which generated the initial segmentation and the correction network, that based on the defined clicks extracted small regions from the initial segmentation and refined them. These refined regions replaced the original ones, resulting in an improved segmentation.

At first, we focused on the initial segmentation network. As the network architecture, we utilised the standard 3D U-Net. In this part, we based our experiments around finding the appropriate loss function that would allow us to use the least amount of data for training of the network, while still producing segmentations that are acceptable. Based on our experiments, we settled for a model trained on 64 images and **FocalTversky loss**, which achieved **0.8030** testing DSC. The fact that the model was over-fitted did not pose a problem for us as we could better test the abilities of our correction network.

Since the correction network uses small cutouts of the initial segmentation, we first developed a process to generate training data for this network, where we simulated faulty segmentations by using different morphological operations. At first, we based the architecture of the correction network on the Attention U-Net and experimented with different sizes of the cutouts as well as different loss functions. Eventually, the best-performing model was trained on 48-sized cuts and our custom modification of the dice loss function. This model achieved testing DSC of **0.8417**.

In the next stage, we hypothesise that the previous model was not fully utilising the additional information provided by the MRI sequences. To solve this issue, we employed a multi-encoder U-Net architecture, which consisted of two separated encoders, one for the faulty cut and one for stacked cutouts of the T1ce and T2 sequences. The separation of encoders slightly improved the performance of the model, which achieved testing DSC of **0.8687**.

In addition to the additional encoder, we also tried to train the network on volumetric cuts, where we experimented with different depths of the training volumes. The model however, did not benefit from the use of the additional dimension.

In order to further improve the performance of the multi-encoder model we decided to fine-tune it on "real" faulty segmentations, as opposed to the ones created by morphological operations. In this case, we utilised cuts extracted from a coarse segmentation generated by the initial segmentation network. These cuts were then used to fine-tune the model. The fine-tuning improved the models performance significantly and achieved testing DSC of **0.8938**.

In the end, we used a simple reconstruction method to demonstrate the complete pipeline, where corrected segmentations achieved testing DSC of **0.8630**, where in comparison the initial segmentations achieved DSC of 0.7538.

5.1 Future work

Our work could potentially be further improved by exploring different merging strategies in the multi-encoder correction model as the fusion of feature maps between the two modalities can have a major effect on the performance of the model.

Another possible improvement could be made in the segmentation reconstruction algorithm, which in its current form is too simplistic. A possible refinement of the method would be an addition of a strategy to handle the overlapping of certain regions, as we view this as a major step back.

Additionally, an interesting extension of the proposed method would be to use the correction network to also improve the initial segmentation model.

Resumé

6.2 Úvod

Používanie počítačového videnia sa stáva čoraz obľúbenejším pri riešení úloh medicínskeho zobrazovania. Tento nárast popularity prišiel s vývojom komplexnejších metód hlbokého učenia, ktoré sú založené na použití konvolučných neurónových sietí (CNN). Tieto metódy sú schopné riešiť rôzne úlohy, ktoré sú kľúčové pre diagnostiku alebo plánovanie liečby, ako je segmentácia alebo klasifikácia z rôznych modalít [1]. Predtým museli tieto úlohy vykonávať manuálne klinickí experti, čo bolo obzvlášť náročné vzhľadom na charakter týchto dát.

6.3 Populárne segmentačné architektúry

Existuje niekoľko rôznych architektúr CNN, ktoré sa ukázali ako úspešné pri úlohách súvisiacich so spracovaním medicínskych údajov, najmä segmentáciou lekárskych obrazových dát.

6.3.1 Architektúry založené na U-Net

V súčastnej dobe sú najpopulárnejšie riešenia založené na U-Net architektúre. Táto architektúra pozostáva z dvoch symetrických častí; enkódera a dekódera. Enkodér

aj dekodér sú zložené z viacerých konvolučných vrstiev, kde enkóder zahŕňa aj 'downsampling' vrstvy a dekóder zas 'upsampling' vrstvy.

Najväčšou výhodou U-Net architektúry je, že na rozdiel od iných architektúr, U-Net efektívne zachytáva lokálne aj globálne kontextové informácie [6]. Zatiaľ čo pôvodná U-Net architektúra využíva 2D konvolúcie, existujú aj modifikácie tejto architektúry na napr. segmentáciu volumetrických lekárskych dát, ktoré využívajú 3D konvolučné vrstvy, ako napríklad V-Net, 3D U-Net alebo UNet++ [8, 10].

6.3.2 Attention U-Net

Ďalšou významnou modifikáciou U-Net architektúry je Attention U-Net. V tejto architektúre autori pridali do pôvodnej U-Net architektúry, takzvané 'attention' brány, ktoré pomáhajú sieti extrahovať relevantné prvky zo vstupných máp. Mechanizmus funguje tak, že hodnoty týchto máp váhujú pred tým ako sú spojené s mapami v dekodéri.

6.4 Časté problémy pri trénovaní neurónových sietí

Medzi hlavné výzvy pri trénovaní neurónových sietí na medicínskych dátach patria: obmedzený počet dostupných anotovaných dát a nerovnováha tried [2].

6.4.1 Učenie s čiastočným učiteľom

Učenie s čiastočným učiteľom môže byť prospešné v prípadoch, keď len malé množstvo obrázkov má zodpovedajúce anotácie. Cieľom tejto metódy je využiť najmä neanotované dáta na trénovanie modelu [27].

6.4.2 Učenie so slabým učiteľom

Podobne ako pri učení s čiastočným učiteľom, v učení so slabým učiteľom sú modely trénované hlavne použitím čiastočne anotovaných, **slabo anotovaných dát** a len s malým množstvom vzoriek s úplnými anotáciami. Bežne používanými typ slabo anotovaných dát, sú obrazové označenia, 'bounding boxy', extrémne body, používateľom definované (generované) kliky.

6.5 Naša práca

Hlavným cieľom našej práce bolo vyvinúť segmentačnú metódu, ktorá využívala slabo anotované dáta. Naše riešenie bolo založené na korekčnej sieti, ktorá vylepšovala prvotné segmentácie vytvorené natrénovanou segmentačnou sieťou.

Vo všeobecnosti možno našu prácu rozdeliť na dve hlavné časti. V prvej časti sme implementovali jednoduchý segmentačný model, ktorý bol trénovaný na malej časti dát, ale s plnými anotáciami. V druhej časti sme trénovali korekčný model na opravu lokálnych chýb z malých rezov.

V práci sme využívali verejne dostupný dataset, ktorý obsahuje T1ce a T2 MRI sekvencie pre pacientov s vestibulárnym schwannomom (VS)[1]. Dataset obsahoval 242 skenov spolu s prislušnou segmentačnou maskou.

6.5.1 Počiatočný segmentačný model

V tejto časti práce sme implementovali jednoduchý segmentačný model na segmentovanie vestibulárneho schwannómu z oboch dostupných MRI sekvencií. Architektúra tohto modelu pozostávala z 3D U-net architektúry. V tomto prípade sme sa snažili nájsť takú stratovú funkciu, pomocou ktorej by sa nám podarilo

[1] `https://www.cancerimagingarchive.net/collection/vestibular-schwannoma-seg/`

natrénovať model na čo najmenšom počte plne anotovaných dát. Na koniec sme použili 'FocalTversky' loss funkciu a model natrénovali na 64 obrázkoch. Tento model dosiahol dice skóre 0.8612 na testovacom datasete.

Modely natrénované na takomto malom počte obrázkov sú samozrejme značne pretrénované. V našom prípade to ale nebol až taký problém, keďže sme aspoň mohli lepšie otestovať korečnú metódu.

6.5.2 Korekčná sieť

V druhej časti našej práce sme implementovali korekčnú sieť, ktorá bola navrhnutá na opravu segmentačných chýb na danej segmentačnej maske.

Naša stratégia pre túto časť zahŕňala extrakciu malých oblastí z danej segmentácie a trénovanie siete na opravu defektov v týchto malých výrezoch. Tento prístup umožnil sieti zamerať sa výlučne na lokálnu oblasť chybnej segmentácie, čo sa ukázalo ako prospešné pre celkový výkon siete. Tieto výrezy sa generujú po kliknutí používateľa na oblasť v segmentácií, ktorú chce opraviť. Následne sa extrahuje malá oblasť okolo kliknutia, ktorá sa odošle do korekčnej siete.

Korekčný model sme trénovali na datasete umelo vytvorených chybných výrezov. Na simuláciu chýb v týchto výrezoch sme využili morphologické operácie, kokrétne dilatáciu a eróziu.

Ako architektúru korekčnej siete sme využili Attention U-Net.

6.5.3 Vlastná stratová funkcia

Na trénovanie korekčnej siete, sme si vytvorili modifikovanú verziu dice stratovej funkcie. Modifikácie tejto stratovej funkcie spočívala vo váhovaní stredových pixelov výrezu. Keďže výrezy sa robia okolo stredného pixelu, váhovanie takýmto

spôsobom by malo zlepšiť presnosť konečnej segmentácie v porovnaní s normálnou dice stratovou funkciou.

6.5.4 Výsledky z trénovania korekčnej siete

V tomto prípade sme si chceli overiť účinnosť našej stratovej funkcie ako aj nájsť správnu veľkosť trénovacích výrezov. Ako najlepšie dopadol model, ktorý bol trénovaný na výrezoch velkosti 48 a s našou modifikáciou dice stratovej funkcie. Tento model dosiahol DSC 0.8417.

6.5.5 U-Net architektúra s viacerími enkódermi

Z výsledkov predchádzajúcich experimentov sme predpokladali, že sieť dostatočne nevyužívala informácie z MRI sekvencií. Rozhodli sme sa teda rozšíriť našu architektúru o ďalší separátny enkóder pre jednotlivé MRI sekvencie.

Dodatočný enkóder mierne zlepšil výkon modelov a dosiahol testovací DSC 0,8687 v porovnaní s 0,8417, ktorý dosiahol predchádzajúci model.

6.5.6 Dolaďovanie modelu

Keďže výkon korekčného modelu stále nebol uspokojivý, rozhodli sme sa doladiť najlepší korekčný model, aký sme doteraz natrénovali na reálnych segmentáciách. V tomto prípade sme sa domnievali, že metóda, ktorú sme použili na napodobňovanie segmentačných chýb, bola možno príliš triviálna a model sa jednoducho naučil vrátiť použité morfologické operácie namiesto ich skutočnej opravy. V tejto súvislosti sme sa rozhodli dodatočne doladiť model na výrezoch vytvorených zo segmentácií generovaných prvotným segmentačným modelom.

Dolaďovanie nakoniec naozaj zlepšilo výkon modelu, kde takto dotrénovaný model dosiahol testovací DSC 0,9265.

6.5.7 Rekonštrukcia segmentácie

Záverečnou časťou našej metódy je rekonštrukcia segmentácie z korigovaných rezov na čo sme využili jednoduchý algoritmus. Tento algoritmus prechádzal cez jednotlivé kliky a nahradil oblasť segmentácie označenú kliknutím opravenými rezmi.

Na testovacích dátach dosiahli upravené snímky priemerný DSC 0.8630. Priemerný DSC pre neupravené snímky bol 0.7538.

6.5.8 Zhodnotenie

Cieľom našej práce bolo navrhnúť metódu, ktorá by segmentovala vestibulárne schwannómy z dostupných MRI sekvencií s použitím slabo anotovaných dát, v našom prípade reprezentovaných klikmi. Navrhovaný 'pipeline' pozostával zo segmentačnej siete, ktorá vygenerovala počiatočnú segmentáciu a korekčnej siete, ktorá na základe definovaných kliknutí extrahovala malé oblasti z počiatočnej segmentácie a vylepšila ich. Tieto spresnené oblasti nahradili pôvodné, čo viedlo k zlepšeniu segmentácie.

Najprv sme sa zamerali na počiatočnú segmentačnú sieť. Ako sieťovú architektúru sme použili štandardný 3D U-Net. V tejto časti sme naše experimenty zamerali na nájdení vhodnej stratovej funkcie, ktorá by nám umožnila použiť čo najmenšie množstvo dát na trénovanie siete a zároveň produkovať segmentácie, ktoré sú prijateľné. V tejto časti bol najlepší model natrénovaný na 64 obrázkoch a **FocalTversky loss**, ktorý dosiahol **0,8030** testovacie DSC.

Keďže korekčná sieť využíva malé výrezy počiatočnej segmentácie, najprv sme vyvinuli proces na generovanie trénovacích údajov pre túto sieť, kde sme simulovali chybné segmentácie pomocou rôznych morfologických operácií.

Najprv sme založili architektúru korekčnej siete na Attention U-Net a experimentovali s rôznymi veľkosťami výrezov, ako aj rôznymi stratovými funkciami, kde bol najvýkonnejší model trénovaný na 48-veľkostných rezoch a našej vlastnej úprave dice stratovej funkcie. Tento model dosiahol testovací DSC **0.8417**.

V ďalšej časti sme vyskúšali použiť architektúru U-Net s viacerými enkódermi. Oddelenie enkóderov mierne zlepšilo výkon modelu, čím sme dosiahli testovací DSC **0.8687**.

Aby sme ešte viac zlepšili výkon multi-enkóderového modelu, rozhodli sme sa ho doladiť na 'skutočných' chybných segmentáciách, na rozdiel od tých, ktoré vznikajú morfologickými operáciami. V tomto prípade sme použili rezy extrahované z hrubej segmentácie vygenerovanej počiatočnou segmentačnou sieťou. Tieto výrezy potom slúžili na doladenie modelu. Toto dotrénovanie modelu výrazne zlepšilo jeho úspešnosť a dosiahol testovací DSC **0.8938**.

Nakoniec sme použili jednoduchú rekonštrukčnú metódu na demonštráciu celého postupu, kde opravené segmentácie dosiahli testovací DSC **0,8630**, pričom počiatočné segmentácie dosiahli DSC 0,7538.

References

[1] Andre Esteva et al. "Deep learning-enabled medical computer vision". In: *NPJ digital medicine* 4.1 (2021), p. 5.

[2] S. Kevin Zhou et al. "A Review of Deep Learning in Medical Imaging: Imaging Traits, Technology Trends, Case Studies With Progress Highlights, and Future Promises". In: *Proceedings of the IEEE* 109.5 (2021), pp. 820–838. DOI: 10.1109/JPROC.2021.3054390.

[3] Shapey Jonathan et al. *Segmentation of Vestibular Schwannoma from Magnetic Resonance Imaging: An Open Annotated Dataset and Baseline Algorithm*. [Data set]. 2021. DOI: doi.org/10.7937/TCIA.9YTJ-5Q73.

[4] Roland Goldbrunner et al. "EANO guideline on the diagnosis and treatment of vestibular schwannoma". In: *Neuro-Oncology* 22.1 (Oct. 2019), pp. 31–45. ISSN: 1522-8517. DOI: 10.1093/neuonc/noz153. eprint: https://academic.oup.com/neuro-oncology/article-pdf/22/1/31/31789998/noz153.pdf. URL: https://doi.org/10.1093/neuonc/noz153.

[5] Sven-Eric Stangerup et al. "The natural history of vestibular schwannoma". In: *Otology & Neurotology* 27.4 (2006), pp. 547–552.

[6] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. *U-Net: Convolutional Networks for Biomedical Image Segmentation*. 2015. arXiv: 1505.04597 [cs.CV].

[7] Nahian Siddique et al. "U-Net and Its Variants for Medical Image Segmentation: A Review of Theory and Applications". In: *IEEE Access* 9 (2021), pp. 82031–82057. DOI: 10.1109/ACCESS.2021.3086020.

[8] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. "V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation". In: *2016 Fourth International Conference on 3D Vision (3DV)*. 2016, pp. 565–571. DOI: 10.1109/3DV.2016.79.

[9] Özgün Çiçek et al. *3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation*. 2016. arXiv: 1606.06650 [cs.CV].

[10] Zongwei Zhou et al. "Unet++: A nested u-net architecture for medical image segmentation". In: *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support: 4th International Workshop, DLMIA 2018, and 8th International Workshop, ML-CDS 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 20, 2018, Proceedings 4*. Springer. 2018, pp. 3–11.

[11] Christian Szegedy et al. *Rethinking the Inception Architecture for Computer Vision*. 2015. arXiv: 1512.00567 [cs.CV].

[12] Siddhartha Chandra et al. "Context aware 3D CNNs for brain tumor segmentation". In: *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries: 4th International Workshop, BrainLes 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 16, 2018, Revised Selected Papers, Part II 4*. Springer. 2019, pp. 299–310.

[13] Rui Hua et al. "Multimodal brain tumor segmentation using cascaded V-Nets". In: *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries: 4th International Workshop, BrainLes 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 16, 2018, Revised Selected Papers, Part II 4*. Springer. 2019, pp. 49–60.

[14] Ozan Oktay et al. *Attention U-Net: Learning Where to Look for the Pancreas.* 2018. arXiv: 1804.03999 [cs.CV].

[15] Mobarakol Islam et al. "Brain tumor segmentation and survival prediction using 3D attention UNet". In: *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries: 5th International Workshop, BrainLes 2019, Held in Conjunction with MICCAI 2019, Shenzhen, China, October 17, 2019, Revised Selected Papers, Part I 5*. Springer. 2020, pp. 262–272.

[16] Dhiraj Maji, Prarthana Sigedar, and Munendra Singh. "Attention Res-UNet with Guided Decoder for semantic segmentation of brain tumors". In: *Biomedical Signal Processing and Control* 71 (2022), p. 103077.

[17] Hesheng Wang et al. "Automatic segmentation of vestibular schwannomas from T1-weighted MRI with a deep neural network". In: *Radiation Oncology* 18.1 (2023), pp. 1–9.

[18] Jonathan Shapey et al. "An artificial intelligence framework for automatic segmentation and volumetry of vestibular schwannomas from contrast-enhanced T1-weighted and high-resolution T2-weighted MRI". In: *Journal of neurosurgery* 134.1 (2019), pp. 171–179.

[19] Holger Roth et al. "Weakly Supervised Segmentation from Extreme Points". In: *Large-Scale Annotation of Biomedical Data and Expert Label Synthesis and Hardware Aware Learning for Medical Imaging and Computer Assisted Intervention*. Ed. by Luping Zhou et al. Cham: Springer International Publishing, 2019, pp. 42–50. ISBN: 978-3-030-33642-4.

[20] Zongwei Zhou et al. "UNet++: Redesigning Skip Connections to Exploit Multiscale Features in Image Segmentation". In: *IEEE Transactions on Medical Imaging* 39.6 (2020), pp. 1856–1867. DOI: 10.1109/TMI.2019.2959609.

[21] Jiwoong J Jeong et al. "Systematic review of generative adversarial networks (gans) for medical image classification and segmentation". In: *Journal of Digital Imaging* (2022), pp. 1–16.

[22] Salome Kazeminia et al. "GANs for medical image analysis". In: *Artificial Intelligence in Medicine* 109 (2020), p. 101938. ISSN: 0933-3657. DOI: https://doi.org/10.1016/j.artmed.2020.101938. URL: https://www.sciencedirect.com/science/article/pii/S0933365719311510.

[23] Andreas Maier et al. "A gentle introduction to deep learning in medical image processing". In: *Zeitschrift für Medizinische Physik* 29.2 (2019), pp. 86–101.

[24] Thomas Neff et al. "Generative adversarial network based synthesis for supervised medical image segmentation". In: *Proc. OAGM and ARW joint Workshop*. Vol. 3. 2017, p. 4.

[25] Yuki Enokiya et al. "Automatic liver segmentation using U-Net with Wasserstein GANs". In: *Journal of Image and Graphics* 6.2 (2018), pp. 152–159.

[26] Saeid Asgari Taghanaki et al. "Deep semantic segmentation of natural and medical images: a review". In: *Artificial Intelligence Review* 54.1 (2021), pp. 137–178.

[27] Yassine Ouali, Céline Hudelot, and Myriam Tami. "An overview of deep semi-supervised learning". In: *arXiv preprint arXiv:2006.05278* (2020).

[28] Xiangli Yang et al. "A survey on deep semi-supervised learning". In: *IEEE Transactions on Knowledge and Data Engineering* (2022).

[29] Pawel Mlynarski et al. "Deep learning with mixed supervision for brain tumor segmentation". In: *Journal of Medical Imaging* 6.3 (2019), pp. 034002–034002.

[30] Longlong Jing, Yucheng Chen, and Yingli Tian. "Coarse-to-Fine Semantic Segmentation From Image-Level Labels". In: *IEEE Transactions on Image*

Processing 29 (2020), pp. 225–236. ISSN: 1941-0042. DOI: 10.1109/TIP.2019.2926748.

[31] Hoel Kervadec et al. "Bounding boxes for weakly supervised segmentation: Global constraints get close to full supervision". In: *Proceedings of the Third Conference on Medical Imaging with Deep Learning*. Ed. by Tal Arbel et al. Vol. 121. Proceedings of Machine Learning Research. PMLR, 2020, pp. 365–381. URL: https://proceedings.mlr.press/v121/kervadec20a.html.

[32] Kevis-Kokitsi Maninis et al. "Deep extreme cut: From extreme points to object segmentation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 616–625.

[33] Holger R Roth et al. "Going to extremes: weakly supervised medical image segmentation". In: *Machine Learning and Knowledge Extraction* 3.2 (2021), pp. 507–524.

[34] Tianyi Zhao and Zhaozheng Yin. "Weakly Supervised Cell Segmentation by Point Annotation". In: *IEEE Transactions on Medical Imaging* 40.10 (2021), pp. 2736–2747. DOI: 10.1109/TMI.2020.3046292.

[35] Tomas Sakinis et al. "Interactive segmentation of medical images through fully convolutional neural networks". In: *CoRR* abs/1903.08205 (2019). arXiv: 1903.08205. URL: http://arxiv.org/abs/1903.08205.

[36] Qing En and Yuhong Guo. *Annotation by Clicks: A Point-Supervised Contrastive Variance Method for Medical Semantic Segmentation*. 2022. arXiv: 2212.08774 [cs.CV].

[37] Ning Xu et al. "Deep Interactive Object Selection". In: *CoRR* abs/1603.04042 (2016). arXiv: 1603.04042. URL: http://arxiv.org/abs/1603.04042.

[38] Mostafa Jahanifar et al. "Robust interactive semantic segmentation of pathology images with minimal user input". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 674–683.

[39] Zhanghexuan Ji et al. “Scribble-based hierarchical weakly supervised learning for brain tumor segmentation”. In: *Medical Image Computing and Computer Assisted Intervention–MICCAI 2019: 22nd International Conference, Shenzhen, China, October 13–17, 2019, Proceedings, Part III 22*. Springer. 2019, pp. 175–183.

[40] Han Liu et al. “Unsupervised Domain Adaptation for Vestibular Schwannoma and Cochlea Segmentation via Semi-supervised Learning and Label Fusion”. In: *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries*. Springer International Publishing, 2022, pp. 529–539. DOI: 10.1007/978-3-031-09002-8_46.

[41] Kibrom Berihu Girum et al. “Fast interactive medical image segmentation with weakly supervised deep learning method”. In: *International Journal of Computer Assisted Radiology and Surgery* 15 (2020), pp. 1437–1444.

[42] Xiaokang Li et al. “HAL-IA: A Hybrid Active Learning framework using Interactive Annotation for medical image segmentation”. In: *Medical Image Analysis* (2023), p. 102862.

[43] Dhruv Sharma et al. “Active learning technique for multimodal brain tumor segmentation using limited labeled images”. In: *Domain Adaptation and Representation Transfer and Medical Image Learning with Less Labels and Imperfect Data: First MICCAI Workshop, DART 2019, and First International Workshop, MIL3ID 2019, Shenzhen, Held in Conjunction with MICCAI 2019, Shenzhen, China, October 13 and 17, 2019, Proceedings 1*. Springer. 2019, pp. 148–156.

[44] Mélanie Gaillochet, Christian Desrosiers, and Hervé Lombaert. “TAAL: Test-time augmentation for active learning in medical image segmentation”. In: *MICCAI Workshop on Data Augmentation, Labelling, and Imperfections*. Springer. 2022, pp. 43–53.

[45] Alessandro Bria, Claudio Marrocco, and Francesco Tortorella. "Addressing class imbalance in deep learning for small lesion detection on medical images". In: *Computers in Biology and Medicine* 120 (2020), p. 103735. ISSN: 0010-4825. DOI: https://doi.org/10.1016/j.compbiomed.2020.103735.

[46] Shruti Jadon. "A survey of loss functions for semantic segmentation". In: *2020 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*. 2020, pp. 1–7. DOI: 10.1109/CIBCB48159.2020.9277638.

[47] Jun Ma et al. "Loss odyssey in medical image segmentation". In: *Medical Image Analysis* 71 (2021), p. 102035. ISSN: 1361-8415. DOI: https://doi.org/10.1016/j.media.2021.102035.

[48] Michael Yeung et al. "Unified Focal loss: Generalising Dice and cross entropy-based losses to handle class imbalanced medical image segmentation". In: *Computerized Medical Imaging and Graphics* 95 (2022), p. 102026. ISSN: 0895-6111. DOI: https://doi.org/10.1016/j.compmedimag.2021.102026.

[49] Jeroen Bertels et al. "Optimization with soft dice can lead to a volumetric bias". In: *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries: 5th International Workshop, BrainLes 2019, Held in Conjunction with MICCAI 2019, Shenzhen, China, October 17, 2019, Revised Selected Papers, Part I 5*. Springer. 2020, pp. 89–97.

[50] Saeid Asgari Taghanaki et al. "Combo loss: Handling input and output imbalance in multi-organ segmentation". In: *Computerized Medical Imaging and Graphics* 75 (2019), pp. 24–33.

[51] Wentao Zhu et al. "AnatomyNet: deep learning for fast and fully automated whole-volume segmentation of head and neck anatomy". In: *Medical physics* 46.2 (2019), pp. 576–589.

[52] Carole H Sudre et al. "Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations". In: *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support: Third International Workshop, DLMIA 2017, and 7th International Workshop, ML-CDS 2017, Held in Conjunction with MICCAI 2017, Québec City, QC, Canada, September 14, Proceedings 3*. Springer. 2017, pp. 240–248.

[53] Shruti Jadon. "A survey of loss functions for semantic segmentation". In: *2020 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*. 2020, pp. 1–7. DOI: 10.1109/CIBCB48159.2020.9277638.

[54] Nabila Abraham and Naimul Mefraz Khan. "A Novel Focal Tversky Loss Function With Improved Attention U-Net for Lesion Segmentation". In: *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*. 2019, pp. 683–687. DOI: 10.1109/ISBI.2019.8759329.

[55] Hao Guan and Mingxia Liu. "Domain adaptation for medical image analysis: a survey". In: *IEEE Transactions on Biomedical Engineering* 69.3 (2021), pp. 1173–1185.

[56] Hyungseob Shin et al. *Self-Training Based Unsupervised Cross-Modality Domain Adaptation for Vestibular Schwannoma and Cochlea Segmentation*. 2021. DOI: 10.48550/ARXIV.2109.10674. URL: https://arxiv.org/abs/2109.10674.

[57] Jae Won Choi. *Using Out-of-the-Box Frameworks for Contrastive Unpaired Image Translation for Vestibular Schwannoma and Cochlea Segmentation: An approach for the crossMoDA Challenge*. 2021. DOI: 10.48550/ARXIV.2110.01607. URL: https://arxiv.org/abs/2110.01607.

[58] Jun-Yan Zhu et al. *Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks*. 2017. DOI: 10.48550/ARXIV.1703.10593.

[59] Fan Xu et al. “LSTM Multi-modal UNet for Brain Tumor Segmentation”. In: *2019 IEEE 4th International Conference on Image, Vision and Computing (ICIVC)*. 2019, pp. 236–240. DOI: 10.1109/ICIVC47709.2019.8981027.

[60] Matej Halinkovič. “Interpretability and Explainability of Deep Learning Systems in the Domain of Microscopic Medical Imaging”. pp. 58-60. Master’s Thesis. Bratislava: FIIT STU, 2023.

[61] Guotai Wang et al. “Interactive Medical Image Segmentation Using Deep Learning With Image-Specific Fine Tuning”. In: *IEEE Transactions on Medical Imaging* 37.7 (2018), pp. 1562–1573. DOI: 10.1109/TMI.2018.2791721.

[62] Yigit B. Can et al. “Learning to Segment Medical Images with Scribble-Supervision Alone”. In: *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*. Ed. by Danail Stoyanov et al. Cham: Springer International Publishing, 2018, pp. 236–244. ISBN: 978-3-030-00889-5.

[63] Fei Du et al. “Efficient mask correction for click-based interactive image segmentation”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 22773–22782.

[64] Konstantin Sofiiuk, Ilya A Petrov, and Anton Konushin. “Reviving iterative training with mask guidance for interactive segmentation”. In: *2022 IEEE International Conference on Image Processing (ICIP)*. IEEE. 2022, pp. 3141–3145.

[65] Xi Chen et al. “Focalclick: Towards practical interactive image segmentation”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 1300–1309.

[66] François Chollet. “Xception: Deep learning with depthwise separable convolutions”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 1251–1258.

[67] Konstantin Sofiiuk, Olga Barinova, and Anton Konushin. "Adaptis: Adaptive instance selection network". In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2019, pp. 7355–7363.

[68] David Martin et al. "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics". In: *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*. Vol. 2. IEEE. 2001, pp. 416–423.

[69] Federico Perazzi et al. "A Benchmark Dataset and Evaluation Methodology for Video Object Segmentation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016.

[70] Shoukun Sun et al. "Cfr-icl: Cascade-forward refinement with iterative click loss for interactive image segmentation". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 38. 5. 2024, pp. 5017–5024.

[71] Qin Liu et al. "SimpleClick: Interactive Image Segmentation with Simple Vision Transformers". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 2023, pp. 22290–22300.

[72] Ke Sun et al. "Deep High-Resolution Representation Learning for Human Pose Estimation". In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 5686–5696. DOI: 10.1109/CVPR.2019.00584.

[73] Xiaoming Liu et al. "Weakly supervised segmentation of COVID19 infection with scribble annotation on CT images". In: *Pattern recognition* 122 (2022), p. 108341.

[74] Mostafa S Ibrahim et al. "Semi-supervised semantic image segmentation with self-correcting networks". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 12715–12725.

[75] Kun Li, George Vosselman, and Michael Ying Yang. “Interactive Image Segmentation with Cross-Modality Vision Transformers”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023, pp. 762–772.

[76] Jiacheng Lin et al. “AdaptiveClick: Click-Aware Transformer With Adaptive Focal Loss for Interactive Image Segmentation”. In: *IEEE Transactions on Neural Networks and Learning Systems* (2024).