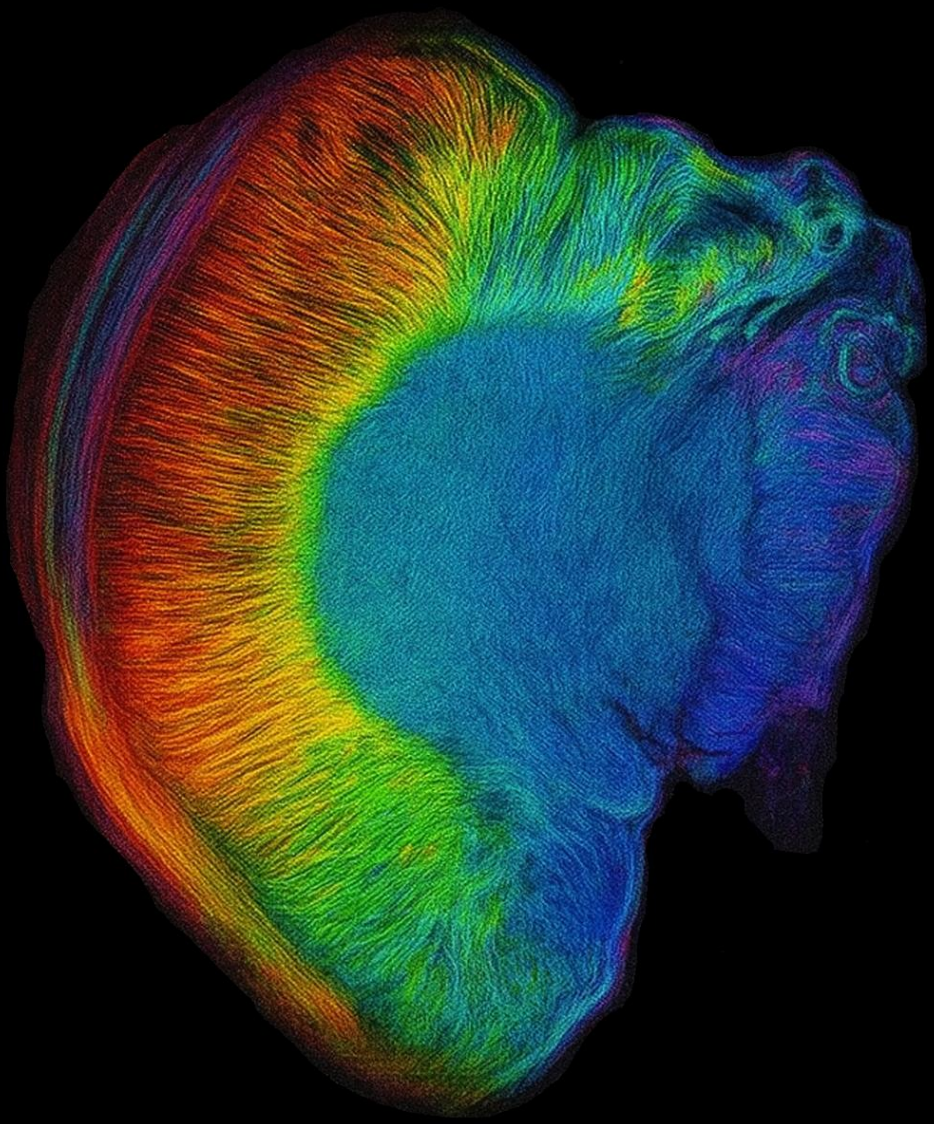
Revealing the margins: Towards ex vivo image-guided surgery in oral cancer resection

Guus Versteeg



Master Thesis

Technical Medicine

11 July 2025

Facilitating intraoperative assessment of resection margins during oral squamous cell carcinoma surgery using automatic segmentation and 3D visualisation on ex vivo 7 tesla magnetic resonance imaging

by

Guus Versteeg

s2102749

UNIVERSITY
OF TWENTE.

Master thesis
Submitted to the Faculty of Science and Technology
University of Twente
In partial fulfilment of the requirements
for the degree of Master of Science

in

Technical Medicine
Track Medical Imaging and Interventions

July 2025

Graduation committee

Dr. C.O. Tan	Chair & university technical supervisor Faculty of Electrical Engineering, Mathematics and Computer Science University of Twente
Dr. M. de Ridder, MD.	Medical supervisor Radiotherapy department Division Imaging & Oncology University Medical Centre Utrecht
H.C. Nguyen, MSc.	Technical Medicine supervisor 3D lab Division Surgical Specialties University Medical Centre Utrecht
N. Cramer Bornemann, MSc.	Process supervisor Communication and professional behaviour Faculty of Science and Technology University of Twente
Dr. ir. W.M. Brink	External member Faculty of Science and Technology University of Twente

Additional supervisors

Dr. ir. M.E.P. Philippens	Hospital technical supervisor Radiotherapy department Division Imaging & Oncology University Medical Centre Utrecht
R. van der Woude, MSc.	Daily supervisor Radiotherapy department Division Imaging & Oncology University Medical Centre Utrecht

Contents

Acknowledgements i
Abbreviations ii
General introduction 1
Chapter 1: Automatic tumour segmentation 5
Introduction 5
Methods 7
Results 11
Discussion 14
Chapter 2: 3D model 16
Introduction 16
Methods 17
Results 23
Discussion 24
Chapter 3: 3D model evaluation 26
Introduction 26
Methods 27
Results 29
Discussion 30
Chapter 4: Super-resolution 32
Introduction 32
Methods 33
Results 34
Discussion 36
General discussion 37
Conclusion 38
References 39
Declaration of AI and AI-assisted technologies in the writing process 42
Appendix A: Probability map merging script 43
Appendix B: 3D model script 45
Appendix C: Clinical cases for 3D model demonstrations 59
Appendix D: Super-resolution script 61

Acknowledgements

I would like to express my gratitude to everyone who has supported me throughout the development of this thesis. First to Chien Nguyen for his extensive scientific guidance and critical insights, and to Can Ozan Tan for his expertise and the enjoyable technical discussions during this project. I am particularly grateful to Rémi van der Woude for her day-to-day support, which played a vital role in the realisation of this thesis. I also thank Mischa de Ridder for his supervision of my clinical development, and Marielle Philippens for overseeing the project and providing valuable feedback. Special recognition is reserved for Nicole Cramer Bornemann for her continued support and coaching in my personal development.

Finally, I would like to thank my fellow students at the 3D lab for making this year the most enjoyable out of my six Technical Medicine years.

Abbreviations

2D	Two-dimensional
3D	Three-dimensional
5-FCV	5-fold cross-validation
7T	7 tesla
95HD	95th percentile Hausdorff distance
ANTs	Advanced Normalization Tools
CNR	Contrast-to-noise ratio
CT	Computed tomography
DICOM	Digital Imaging and Communications in Medicine
DSC	Dice similarity coefficient
DWI	Diffusion-weighted imaging
GPU	Graphics processing unit
HE	Haematoxylin and eosin
IOARM	Intraoperative assessment of resection margins
MR	Magnetic resonance
MSD	Mean surface distance
NIfTI	Neuroimaging Informatics Technology Initiative
nnU-Net	no-new-U-Net
OR	Operating Room
OSCC	Oral squamous cell carcinoma
PC	Personal computer
PET/CT	Positron emission tomography/computed tomography
RMSE	Root mean squared error
SD	Standard deviation
SNR	Signal-to-noise ratio
SSIM	Structural similarity index measure
SUS	System Usability Scale
T2w	T2-weighted
TSE	Turbo spin echo
UMC Utrecht	University Medical Centre Utrecht

General introduction

Of all cancers worldwide, head and neck cancer was determined to be the sixth most common in 2018, with 890,000 new cases and 450.000 deaths[1]. Oral squamous cell carcinoma (OSCC) is diagnosed most frequently in the head and neck region, accounting for nearly half of all cancers[2]. Most prevalent tumour locations include the tongue and the floor of the mouth, while the gingivae, palate, and buccal and labial mucosa are affected less commonly (see Figure 1)[3].

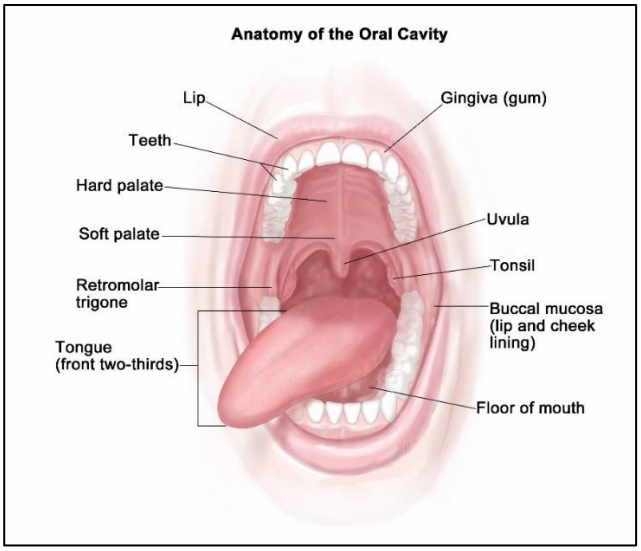


Figure 1: Anatomy of the oral cavity[4]

The primary choice of treatment for OSCC is complete surgical removal of the tumour[5]. Wide margins around the resected tumour are crucial to maximise clinical outcome[6]. However, there is a high incidence of inadequate margins (< 5 mm on histopathology, see also Table 1), with numbers ranging from 40% to 85%[7], [8], [9]. Inadequate surgical margins increase the likelihood of local recurrence, regional recurrence, distant metastasis, and decrease overall survival[7]. When post-surgical histopathological examination of the tumour shows inadequate margins, the patient is therefore subjected to additional treatment including (chemo-)radiotherapy (27%) and/or secondary surgery (12%)[10], [11], [12]. These additional treatment options contribute to higher healthcare costs and significantly impact the patients' quality of life[13], [14], [15]. Furthermore, adequate margin control is complicated in secondary surgery as it becomes more difficult to assess the anatomical orientation between the original and secondary specimen[5].

Table 1: OSCC resection margin indication[16]

Resection margin	Indication	Description
> 5 mm	Low risk	Negative, clear, radical, adequate
1 – 5 mm	Intermediate risk	Close, inadequate
< 1 mm	High risk	Positive, involved, irradical, inadequate

To overcome inadequate margins during OSCC resections, the surgeon can perform intraoperative assessment of resection margins (IOARM). To date, numerous IOARM methods have been proposed, for example Raman spectroscopy, ultrasound, autofluorescence, and optical coherence tomography to electromagnetically tracked ultrasound systems[17], [18], [19], [20].

Magnetic resonance (MR) imaging has recently been reported to enable accurate localisation of OSCC in resection specimens due to its strong ability to distinguish tumour from healthy tissue[8, 21]. Currently at the University Medical Centre Utrecht (UMC Utrecht), the 7Tex study is being conducted to examine the feasibility of ex vivo high-resolution MR imaging as a means of IOARM. This MR-based IOARM workflow is being set up in which resected OSCC specimens are prepared and sent to a preclinical, small-bore 7 tesla (7T) MR scanner during surgery. High-resolution T2-weighted (T2w) scans are acquired of the specimens, which are then used to visualise the resection margins intraoperatively.

The overall goal of the 7Tex study is to perform this intraoperative workflow within a time limit of 45 minutes, as set by surgeons for a minimal disruption of the surgical process. Implementing such a workflow could improve margin control and thereby reduce the amount of adjuvant therapy. This imaging workflow includes specimen preparation, transport to the MR scanner, scan acquisition, and scan processing. Within these 45 minutes, the surgeon could perform a neck dissection or sentinel lymph node procedure, minimising the impact on the surgical process and operating time.

A second goal of the 7Tex study is to provide surgeons with an intuitive, automatic visualisation model of the resected tumour and its resection margins. In 2024, a study by De Koning et al. already explored the possibility to create three-dimensional (3D) visualisations of the resected specimen using 7T MR scans[5]. The proposed workflow was however still too labour- and time-intensive for intraoperative implementation due to its manual nature. Spacing of slices in the 3D visualisation was also irregular and ranged from 1 to 10 mm, which could cause inaccuracies in measuring resection margins. A new and improved visualisation method should be produced quickly enough for intraoperative feasibility with adequate source scan resolution in all directions.

Problem statement

An automatic pipeline for tumour delineation and resection margin visualisation is required to reach IOARM within 45 minutes after specimen resection. After tumour delineation, a usable and intuitive designated automatic visualisation method must be developed. The main research question of this study is:

How can intraoperative resection margin assessment be facilitated using ex vivo 7T MRI of oral squamous cell carcinoma resection specimens?

This study proposes an IOARM method which maintains the high-resolution benefit of the preclinical 7T MR scanner also used by De Koning et al., while eliminating the need for manual 3D visualisation generation. To contribute to the larger goal of executing specimen preparation and transport, scan acquisition, and resection margin assessment within 45 minutes, this study aims to set up a clinical pipeline which generates an intuitive IOARM method as quickly as possible. This pipeline will include automatic segmentation of the tumour using deep-learning networks, and an automatic 3D visualisation model of the tumour and resection margins. The total pipeline contains several essential components which will be thoroughly covered in this study (Figure 2).

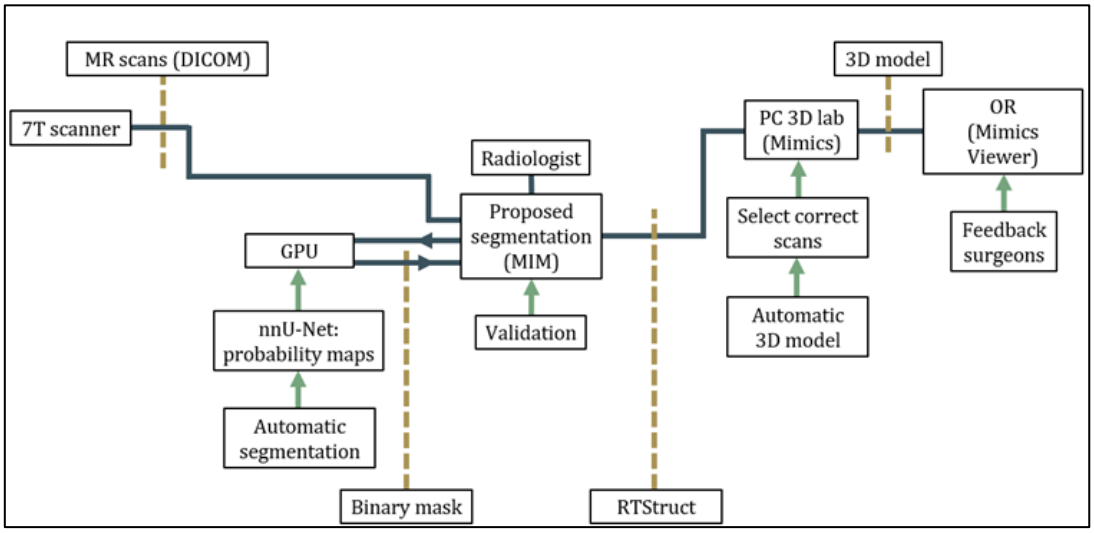


Figure 2: Clinical pipeline. MR scans are acquired by a 7T scanner and imported into MIM in DICOM format. From here, deep-learning networks can be called to perform OSCC segmentation on a graphics processing unit (GPU), after which this segmentation is sent back to MIM as a binary mask. Within MIM, a radiologist reviews the segmentation, makes adjustments and confirms the final segmentation. This final segmentation is sent through along with the MR scans in RTStruct format to a personal computer (PC) at the 3D lab. Here, the 3D model is automatically generated based on the imported RTStruct, after which the model is uploaded to Mimics Viewer so that it can be presented in the operating room (OR).

This study is divided into multiple subsections of which each focuses on a different component of the overall objective. First, automatic tumour segmentation is required to generate a viable and accurate 3D model as quickly as possible. A workflow containing deep learning networks is set up, of which the segmentation performance is evaluated. The subquestion for this section is:

How can automatic segmentation of OSCC be performed on 7T MR imaging using nnU-Net networks?

Secondly, the focus of this study is to develop the 3D model for IOARM. It is essential that this 3D model generation is robust, intuitive, and automatic. The goal is to generate the model within five minutes. The subquestion for this section is:

How can an intuitive, three-dimensional model be created within five minutes, which enables intraoperative assessment of resection margins for surgeons performing OSCC resections?

Thirdly, it is imperative that this 3D model is objectively validated to gain an understanding of its usability before it is clinically implemented. The third section focuses on the following subquestion:

What is the usability of the automatic, intraoperative three-dimensional model, according to all surgeons performing OSCC resections at the UMC Utrecht?

Lastly, high-resolution MR imaging enables superior in-plane resolution but is regularly held back by poor relative resolution in the through-plane direction. Isotropic 3D MR scans may solve this issue, however clinically implementing such scanning sequences is hampered by relatively long scanning times, sub-optimal contrast and blurring. To improve clinical applicability of high-resolution MR imaging, the technique of super-resolution is investigated to gauge its feasibility in reproducing isotropic, 3D scans of transverse and sagittal T2w scans. As voxel anisotropy is encountered frequently in high-resolution MR imaging, a solution is suggested for resampling difficulties originating from highly anisotropic voxel dimensions. The subquestion of this section is:

To what extent can 3D T2-weighted scans of OSCC specimens be recovered by means of super-resolution upsampling based on anisotropic transverse and sagittal T2-weighted MR scans with high in-plane resolution?

Chapter 1: Automatic tumour segmentation

Introduction

The first part of the pipeline considers the MR scan processing and the automatic segmentation of the OSCC (Figure 1.1). The MR scans are imported into software at the Radiotherapy department which enables automatic calling of deep-learning networks and manual adaptation of segmentations.

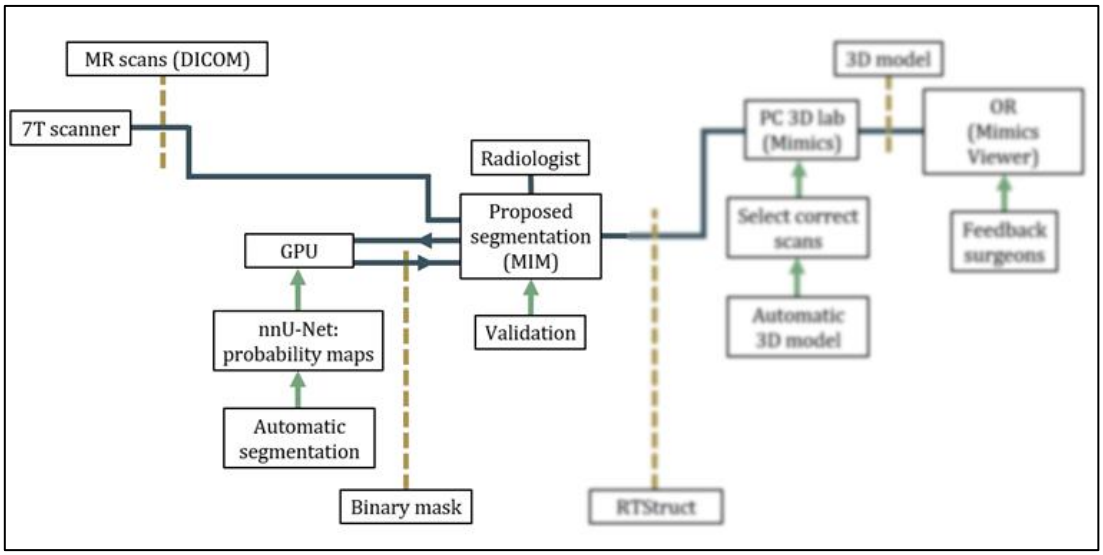


Figure 1.1: Clinical pipeline: first section

MR scan handling and segmentation adaptation is performed in MIM (MIM Software, Beachwood, OH, USA), which is imaging software frequently used by radiation oncologists at the Radiotherapy department. Within MIM, pretrained deep-learning networks can be called which run on graphics processing units (GPUs) and return segmentations of tumour tissue. Radiologists can visually check the proposed segmentations and make alterations slice-by-slice if necessary. The final segmentation can then be sent further onwards in the pipeline.

Background

Deep-learning is an ever-evolving technique which has become an indispensable part of medical image processing and segmentation. To achieve robust deep-learning networks which produce reliable segmentations for a clinical environment, large databases are required[22]. To ensure optimal segmentation performance in clinical practice, it is important that the training data reflect the characteristics of the scans encountered in the target clinical workflow. Deep-learning segmentation tasks in head and neck cancers are abundant in literature, focusing on positron emission tomography/computed tomography (PET/CT) scans and CT scans, as well as fluorescent confocal microscopy[23], [24], [25], [26]. Less prevalent are deep-learning networks trained on ex vivo MR imaging, especially for OSSC segmentation. Databases on which a deep-learning network could be trained to segment OSCC tissue on 7T ex vivo MR scans are therefore not readily accessible, which is an important factor when determining the most appropriate deep-learning network.

The no-new-U-net (nnU-Net) is a deep-learning, segmentation-based method which has self-configuration capabilities, especially designed for atypical segmentation tasks[27]. The method is designed to adapt to a given dataset, automatically generating a fitting U-net-based segmentation pipeline. The clinical pipeline in this study requires a deep-learning network which can be trained from scratch on ex vivo 7T MR scans of OSCC resection specimens; a highly non-standard image modality

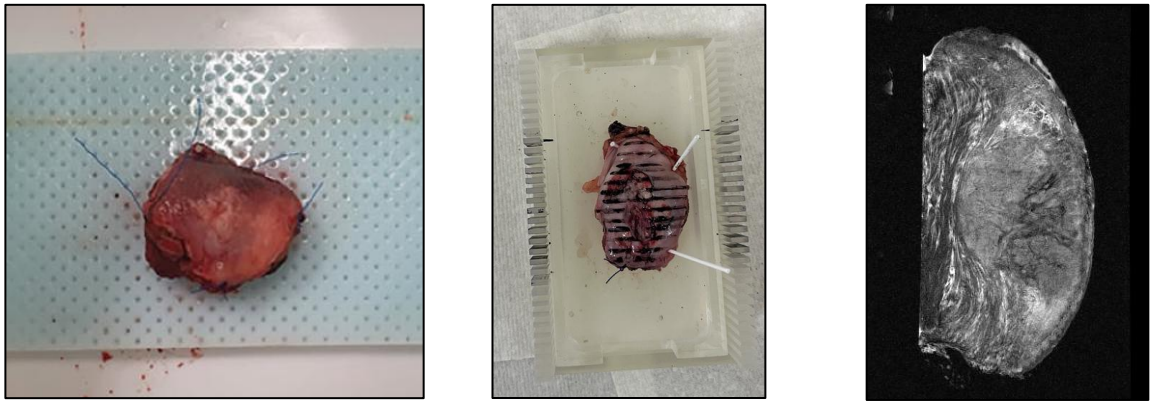
for which no pretrained networks or existing segmentation pipelines are available. This makes the nnU-Net highly suitable for this task. Automatic OSCC segmentation is essential to generate an IOARM method as quickly as possible.

Building upon this, the segmentation accuracy of the deep learning networks will be assessed. In a paper by Wahid et al., deep-learning based segmentation of oropharyngeal primary tumours on multiparametric MR scans resulted in an average Dice similarity coefficient (DSC) score of 0.73, which was considered reasonable[28]. Therefore, the accuracy goal for this section is a DSC of ≥ 0.7. This also ensures correct tumour localisation despite potential boundary inaccuracies which can be accounted for by radiologists further down the clinical pipeline. A second focus is the segmentation performance of the nnU-Net networks.

Methods

Specimen scanning

Patients undergoing curative surgery for OSCC resection (cT1-T4) were included in the 7Tex study workflow. Once the OSCC was resected, it was retrieved from the operating room (OR) and transported to a utility room, where the specimen was prepared (Figure 1.2a). Here, the specimen was placed on a plastic holder and fixated with intravenous cannulas. Parallel lines were painted on the specimen, perpendicular to the longest dimension of the specimen (unless indicated otherwise by the Pathology department). These lines indicated the transverse scanning direction and the direction in which pathological sections should be made to allow validation of the MR images with histology (Figure 1.2b). The specimen was placed inside a plastic cylinder, which was then filled with perfluoropolyether (Galden, Solvay Solexis, Thorofare, NJ, USA) to account for susceptibility artifacts on the MR images. The cylinder was then transported to the Gemeenschappelijk Dierenlaboratorium (Universiteit Utrecht, Utrecht, The Netherlands) for scanning (Figure 1.2c).



A **B** **C**

Figure 1.2: Resected (A), fixated (B) & scanned (C) specimen

Specimen scanning was performed using a preclinical, small-bore 7T MR scanner (Biospec 7T, Bruker, Ettlingen, Germany), interfaced with a Philips console (Philips Medical Systems, Best, The Netherlands). Orientation and size of the field of view were manually chosen during MR scout scanning and differed based on specimen dimensions. To save time, T2w Turbo Spin Echo (TSE) scans were acquired in only two directions for clinical implementation: transverse and sagittal. These scans had an in-plane resolution of 0.125 mm^2 and a slice thickness of 1 mm (Figure 1.2c).

Study population

A total of 77 OSCC resection specimens were included in the 7Tex study, of which 55 were suitable for training of the nnU-Net. Exclusion criteria were:

- Larger artifacts on the MR scans
- Patients with a pT1 tumour, unless the tumour was clearly distinguishable on the MR scans

The median age of this patient group was 63 years old (range 27 – 87). More patient group details are stated in Table 1.1.

Table 1.1: Patient group details

Sex		**Pathological tumour staging (pTNM)**		**OSCC origin**	
Male	28	T1	15	Tongue	31
Female	27	T2	23	Bucca	16
		T3	11	Floor of mouth	7
Total	55	T4	6	Gingiva	1

MR scans

Of the 55 included specimens, 55 transverse and 47 sagittal T2w scans were available. The scans were acquired in Digital Imaging and Communications in Medicine (DICOM) format and converted to Neuroimaging Informatics Technology Initiative (NIfTI) format for further processing.

Ground truth

The ground truth consisted of manual tumour delineations in the transverse T2w scans. All segmentations were performed by two researchers of the 7Tex study. First, the tumour was delineated on the corresponding haematoxylin and eosin (HE) histopathological slices and approved by a dedicated pathologist. The histopathological slices were then registered to the corresponding transverse MR slices via point-based registration. Subsequently, the histological delineations could be propagated to the transverse T2w scans. The delineations were then manually adjusted to reach the correct tumour segmentation on MR. The delineations could also be propagated directly to the sagittal scans since the transverse and sagittal scans were registered to the same image space. The delineations were exported as binary masks in GIPL format from in-house contouring software (Volumetool, version 1.30.64)[29] and converted to NIfTI format using the SimpleITK Python library. Each mask was resampled to its corresponding T2w scan using nearest-neighbour interpolation.

nnU-Net architecture

In the current study, transverse and sagittal T2w scans were available for tumour segmentation using the nnU-Net. One nnU-Net network required resampling of these two scan types to the same image space and dimensions in order to properly process both scanning directions. An inherent disadvantage to 7T MR imaging with a high in-plane resolution was the relatively large slice thickness compared to in-plane pixel size, which caused a large difference in spatial resolution between scanning directions. Resampling of transverse and sagittal scans to the same image space was challenging as their voxels were highly anisotropic. Alignment of voxel sizes across axes would require substantial interpolation or extrapolation which likely would introduce artifacts or lead to loss of anatomical detail[30]. A study by Mulder et al. explored the effects of voxel anisotropy on volume estimation in MR scans and found that 3-fold voxel anisotropy consistently caused a deviation to volume ground truth of more than 50%[31]. As the voxel dimensions in this study showed 8-fold anisotropy (0.125x0.125x1 mm), resampling the sagittal scans to the transverse scans for use in the nnU-Net was not a viable option.

To overcome this resampling problem, one network was trained separately on the transverse scans and another on the sagittal scans. Both networks were configured to output tumour predictions in the form of probability maps rather than binary segmentations. This approach enabled fusion of OSCC tumour predictions while still preserving the anatomical information of the high in-plane resolution of both the transverse and sagittal acquisitions (see also Figure 1.3).

nnU-Net hyperparameters

All nnU-Net networks were trained using a combination of dice loss and cross-entropy loss over 1000 epochs. Because transverse and sagittal datasets were trained on separate networks, all networks were trained in the two-dimensional (2D) configuration[27].

nnU-Net training

nnU-Net networks were trained separately on the transverse and sagittal datasets. Table 1.2 shows the composition of each train and test set. 80% of each dataset was used for training, the remaining 20% was used for testing. Seven ground truth delineations were not checked by a pathologist, which is why they were spread over the datasets in the same 80/20 ratio. All eight patient cases missing a sagittal T2w scan were placed in the training set of the transverse networks. For each scanning direction, two networks were trained. One was trained using 5-fold cross-validation (5-FCV) to estimate segmentation performance and assess network variability across different data subsets. A second network was trained using the entire training set (80%) to exploit all available data for inference on the held-out test set, following standard practice to balance robust evaluation and optimal final performance.

Table 1.2: Train/test splits

Network	**Training (%)**	**Test (%)**	**Network**	**Training (%)**	**Test (%)**
Transverse nnU-Net	44 (80)	11 (20)	**Sagittal nnU-Net**	38 (81)	9 (19)
Non-validated scans	6 (86)	1 (14)	**Non-validated scans**	6 (86)	1 (14)
Missing sagittal T2w	8 (100)	0 (0)	**Missing sagittal T2w**	0 (0)	0 (0)

Table 1.3: Trained networks

Transverse 5-fold cross-validation	Transverse 80/20
Sagittal 5-fold cross-validation	Sagittal 80/20

Merging of network output

The outputs from the transverse and sagittal deep-learning networks were probability maps representing predicted segmentations. These maps were saved in NPZ format and merged using a dedicated Python script (see Figure 1.3 for a schematic overview, and Appendix A for the script). The sagittal map was first resampled to match the transverse image space using linear interpolation.

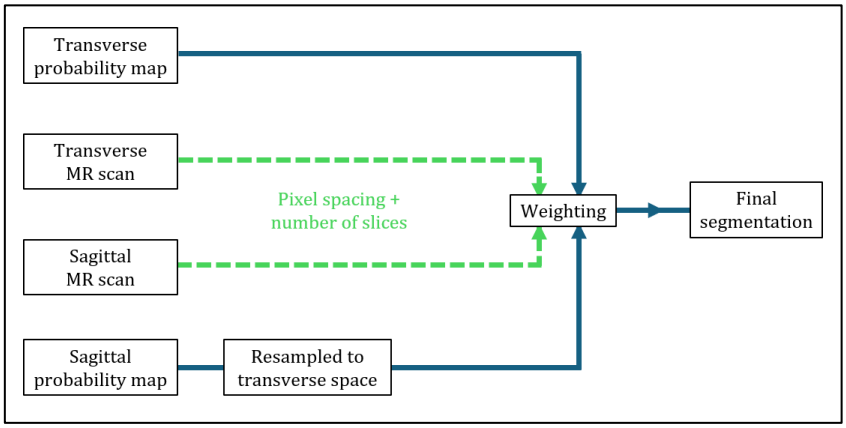


Figure 1.3: Probability map merging workflow. After resampling of the sagittal probability map to the transverse image space, both maps are merged using weighting. As each probability map prediction is based on its corresponding MR scan, one prediction could be based on more anatomical information than the other, creating an imbalance in prediction certainty between the transverse and sagittal prediction. To account for this imbalance, weighting is applied which is based on the pixel spacing and number of slices of the corresponding transverse and sagittal MR scans.

When merging probability maps, it is important to note that one map may hold more information regarding the predicted tumour than the other map, which means that this map should be assigned a higher weight to reflect its greater relevance in the final segmentation.

Weighting was based on the physical depth of the corresponding transverse and sagittal MR scans because this depth directly correlated to the amount of information available in each scan. Weighting was applied as two factors together summing to one. This weighting was determined by multiplying the voxel spacing with the number of slices per MR scan. By multiplying these parameters, the physical depth of each scan was determined. Weighting of the transverse probability map was determined by dividing the physical depth of the transverse scan by the sum of the physical depths of both scans:

$$W_T = \frac{N_T * P_T}{N_T * P_T + N_S * P_S}$$

Where W is the weighting, N is the number of slices per scan, and P is the pixel spacing of each scan (with T for transverse and S for sagittal). Weighting of the sagittal probability map was determined by subtracting the transverse weighting from one:

$$W_S = 1 - W_T$$

Merging of probability maps was then achieved by applying each weighting:

$$combined\ prob\ map = transversal\ prob\ map * W_T + sagittal\ prob\ map * W_S$$

The final binary segmentation was created by assigning each voxel to the class with the highest probability in the combined probability maps.

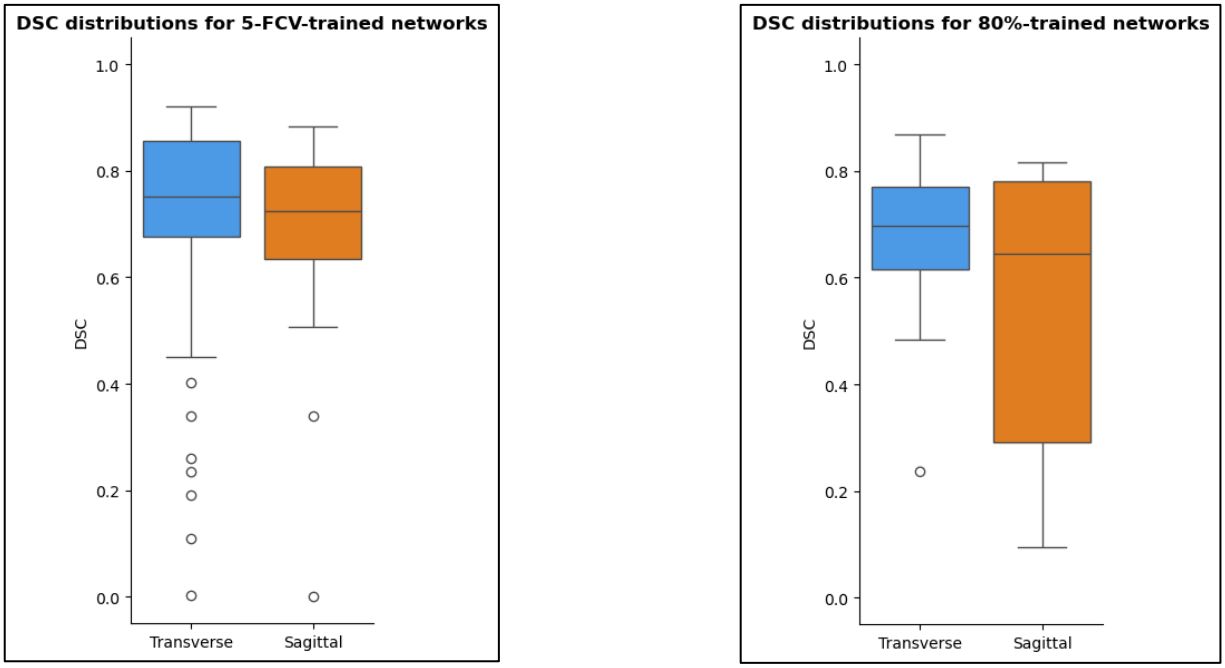
Network performance evaluation

Predicted segmentations were measured against ground truth delineations using the DSC, sensitivity, specificity, 95th percentile Hausdorff distance (95HD), and the mean surface distance (MSD). Sensitivity and specificity were calculated within a rectangular bounding box enclosing the ground truth tumour volume, defined by its outer dimensions, to ensure class balance and relevance to the tumour region.

Results

DSC, sensitivity, specificity, 95HD, and MSD metrics are presented of the transverse and sagittal networks. The average DSC for transverse networks was 0.695 ± 0.231 for the 5-FCV network and 0.659 ± 0.170 for the 80%-trained network. For the sagittal networks, average DSC scores were 0.698 ± 0.164 and 0.478 ± 0.307 for 5-FCV and 80%-trained networks respectively. Figure 1.4 shows the interquartile range of DSC distributions of both networks for each scanning direction dataset, including outliers.

Seven outliers were observed in the transverse 5-FCV-trained network with DSC scores of 0.4 or lower. For the sagittal 5-FCV-trained network, there were two outliers at DSC scores of 0 and 0.34. In the 80%-trained networks only one outlier was found in the transverse network at a DSC score of 0.24.



A **B**

Figure 1.4: DSC score distributions for 5-FCV (A) and 80% (B) trained networks

Average 95HD scores were the lowest for the sagittal 5-FCV network and highest for the sagittal 80%-trained network. Table 1.4 shows the averaged metrics and corresponding standard deviation (SD) for each network, including the median 95HD.

Table 1.4: Segmentation metrics

Network	Average DSC (SD)	Average sensitivity (SD)	Average specificity (SD)	Average 95HD in mm (SD)	Median 95HD in mm	Average MSD in mm (SD)
Transverse 5-FCV	0.695 (0.231)	0.676 (0.238)	0.959 (0.0356)	4.16 (3.00)	3.00	0.765 (0.601)
Sagittal 5-FCV	0.698 (0.164)	0.655 (0.190)	0.952 (0.0364)	3.34 (1.62)	3.09	0.638 (0.258)
Transverse 80%	0.659 (0.170)	0.658 (0.169)	0.959 (0.0357)	6.27 (8.05)	3.07	1.29 (2.02)
Sagittal 80%	0.478 (0.307)	0.444 (0.314)	0.970 (0.0286)	7.38 (9.49)	3.42	2.10 (3.57)
Merged segmentations	0.622 (0.218)	0.588 (0.227)	0.976 (0.0150)	6.75 (8.57)	3.34	1.48 (2.40)

Two examples of OSCC predictions (red) are given in Figures 1.5 and 1.6 against their ground truth (green), displaying a good segmentation (DSC = 0.91) and a poor segmentation (DSC = 0.32) respectively.

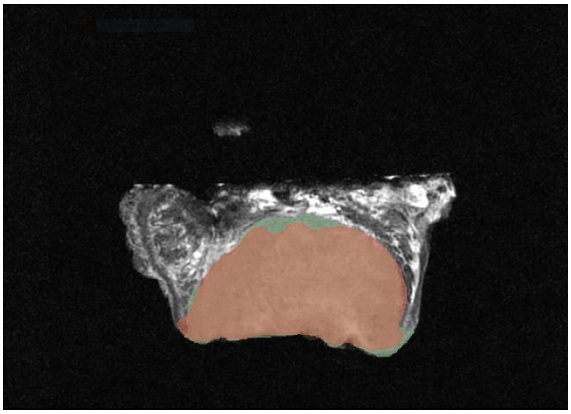


Figure 1.5: 5-FCV OSCC prediction (red) on ground truth (green) on transverse scan, with a high DSC score of 0.91

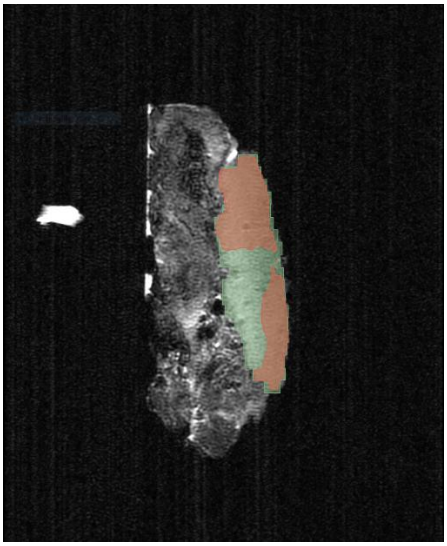


Figure 1.6: 80% OSCC prediction (red) on ground truth (green) on sagittal scan, with a low DSC score of 0.32

Figure 1.7 shows the learning curves for one fold of the transverse 5-FCV-trained network, with pseudo Dice in green, validation loss in red, and training loss in blue. These curves were measured over the full training duration of 1000 epochs.

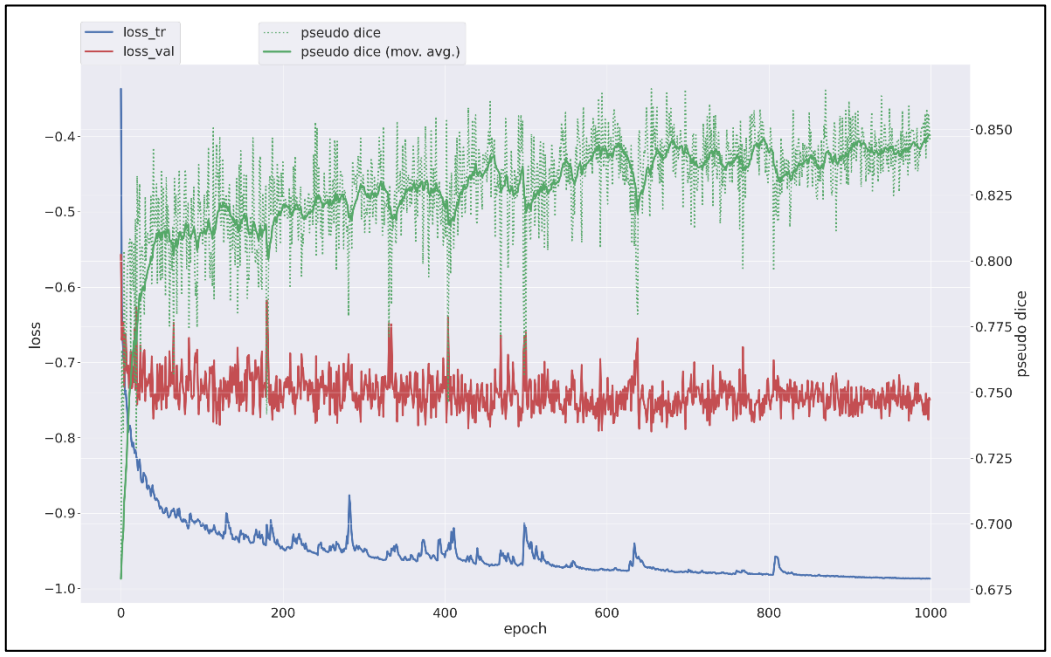


Figure 1.7: Training log of learning curves of the transverse 5-FCV network. The training loss is presented in blue and the validation loss in red.

Figure 1.8 displays a merged OSCC segmentation (green), generated from a transverse (red) and sagittal (blue) prediction. Overlay of all three segmentations illustrates how the merge script determined the final segmentation boundaries.

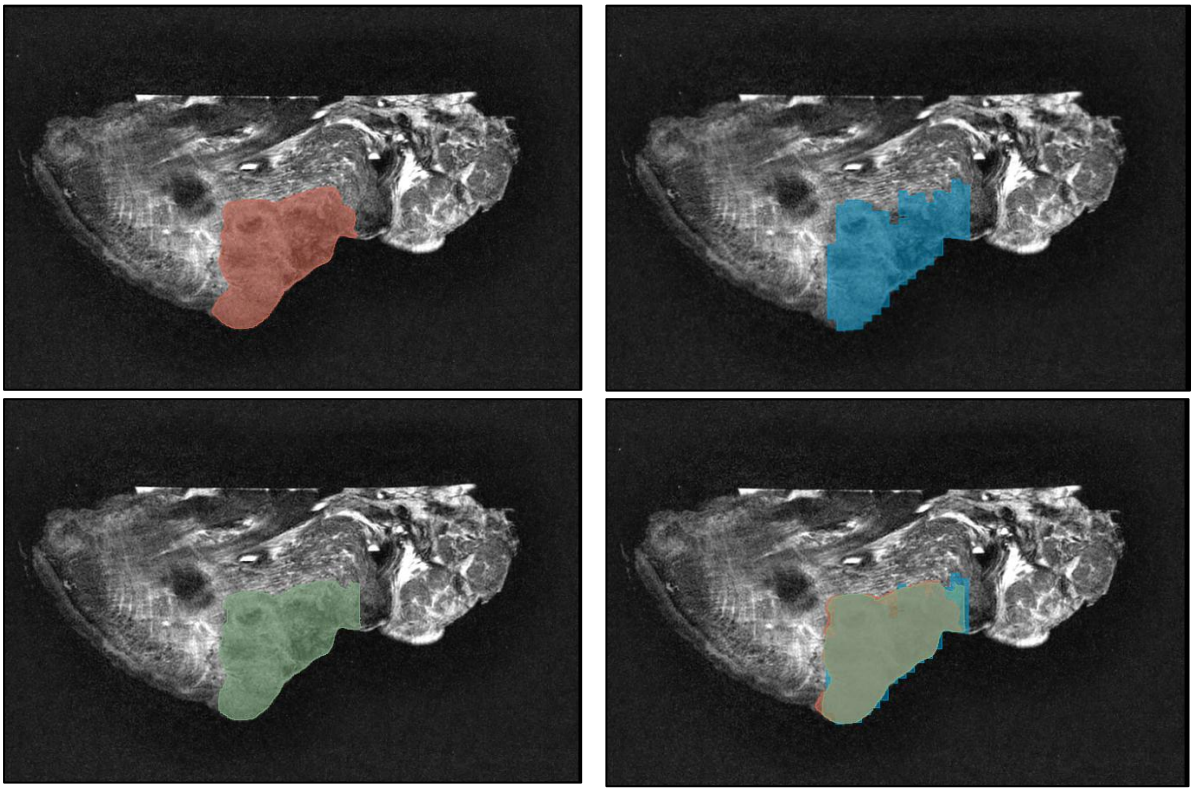


Figure 1.8: Transverse (red), sagittal (blue), and merged (green) tumour predictions on a transverse MR scan

Discussion

Automatic segmentation of OSCC on transverse and sagittal 7T MR scans was achieved by training two deep-learning networks and merging their probability map output using adaptive weighting. The observed segmentation accuracy was reasonable, considering the dataset size. The findings of this section show that automatic segmentation is applicable in the proposed clinical pipeline, which contributes to the IOARM method being performed as quickly as possible.

Judging from the results, the average segmentation performance of the deep-learning networks is decent, apart from some outliers (Figure 1.4). Of all outliers in the predictions of 5-FCV-trained networks, 67% were predictions on pT1 tumours, while 22% were pT2 and 11% were pT3 tumours. This suggests that the network can find relative difficulty in predicting small tumours. Figure 1.9 shows DSC distributions for the 5-FCV-trained networks where all pT1 tumours are removed. Average DSC jumped from 0.695 (± 0.231) to 0.747 (± 0.172) for the transverse and from 0.698 (± 0.164) to 0.733 (± 0.121) for the sagittal network, signifying a respective 7.5% and 5.0% increase compared to when pT1 tumours were included. This suggests that the networks perform significantly better on tumour sizes bigger than pT1.

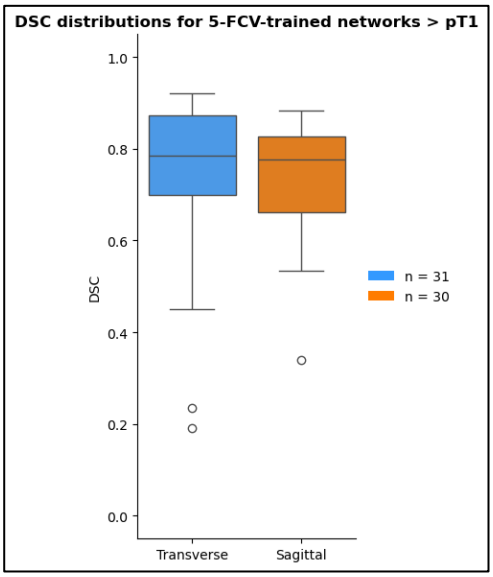


Figure 1.9: DSC distributions of all tumours > pT1

The distribution of the DSC scores of the network trained on 80% of the data is notably different to the other three DSC score distributions, most likely due to the low sample size (n = 9), in combination with a case where no OSCC tissue was predicted. The high specificity scores can be explained by the imbalance between tumour and non-tumour pixels in the analysed bounding box. The median 95HD values of 3 to 3.09 mm and the average MSD values of 0.765 and 0.638 mm of the two 5-FCV-trained networks illustrate how the networks do not perfectly delineate OSCC edges on the scans. However, as the main purpose of the networks is to accelerate the tumour segmentation process, the networks are still effective in the clinical pipeline as the radiologist can correct for such imperfections. The performance of the final merged segmentations falls in between that of the transverse and sagittal 80%-trained networks, which is to be expected as they form a combination of both network's output.

Visual analysis of predicted segmentations by the 5-FCV-trained networks revealed that the networks usually predicted the correct tumour location in the scans. In some cases, the tumour volume was underestimated, while in other cases an overestimation of tumour tissue was observed. Specific anatomical structures like salivary glands did not cause misprediction of tumour tissue by the

networks. This suggests that the networks have a decent understanding of tumour tissue characteristics.

To assess potential overfitting in the trained nnU-Net networks, training and validation loss trajectories were monitored during training over all folds for each network. The trajectories of the training and validation loss functions of fold 1 of the transverse 5-FCV network in Figure 1.7 are representative of all trained networks. The training loss exhibited a gradual decline while validation loss either steadily decreased or plateaued, with no sudden increases or deviations signifying overfitting. This indicates a healthy convergence and suggests the absence of classic overfitting behaviour[32]. The average DSC scores from all five folds was 0.696 (± 0.077) for the transverse and 0.698 (± 0.051) for the sagittal network. This acceptable variability in standard deviations was another sign that the networks are stable and that overfitting was not evidently present. The average DSC scores of the networks on the held-out test set were 0.659 for transverse and 0.478 for sagittal (Table 1.4). While the average transverse DSC score is a promising sign of robust performance on unseen data, the relatively low average sagittal DSC score could mean that the sagittal network indeed struggles on unseen data. However, these results should be interpreted with caution due to the small test set size (n = 9), which limits statistical reliability and increases sensitivity to outlier cases. Also, while the training and validation loss learning curves showed persistent improvement, a large gap remained between the two (Figure 1.7). This may signify a shortage of data for the network to sufficiently carry out the segmentation task[33]. While suspicion of overfitting is low, further testing on a larger dataset will be necessary to more conclusively assess the network's susceptibility to overfitting and its true generalisation capacity.

Furthermore, in terms of the ground truth, discrepancies could occur during the point-based registration of histopathological slices to the transverse MR scans. When adjustment of the ground truth delineations was required as a result, this was performed manually by two researchers of the 7Tex study. Ideally, the delineations should be performed by radiologists, however this has not been done due to two reasons. Firstly, the radiologists are not trained to delineate OSCC tissue on ex vivo MR scans, while the researchers had significantly more experience delineating OSCC tissue on the MR scans. Secondly, validation of all delineations by a radiologist was not logistically possible.

The ground truth delineations were also based on the transverse scans because they corresponded to the histopathological slices and because the tumour was best visible in this direction. The delineations were therefore slightly pixelated on the sagittal scans, which will have impacted the segmentation performance of the deep-learning networks trained on sagittal slices. This will explain some of the difference in DSC scores between the transverse and sagittal networks (Figure 1.4), together with the fact that the training set for the sagittal network was slightly smaller.

Looking ahead, given the significant impact of tumour size on segmentation performance, extra attention by the radiologist should be given to cases where a small tumour is delineated in the clinical pipeline. Nevertheless, the clinical impact of this will be limited as pT1 tumours will be excluded from the MARGIN study, in which the proposed pipeline will be clinically tested.

Conclusion

This study demonstrated that automatic segmentation of OSCC on ex vivo 7T MR scans is feasible using a dual-network approach with merged probability maps. The integrates well into the proposed clinical pipeline, supporting quick intraoperative assessment of resection margins.

Chapter 2: 3D model

Introduction

The second part of the pipeline considers the automatic 3D visualisation model of the resected OSCC specimens including the resection margins (Figure 2.1). Once the tumour segmentations are completed, they are imported along with the corresponding transverse MR scans to a computer which automatically generates a 3D model based on these data.

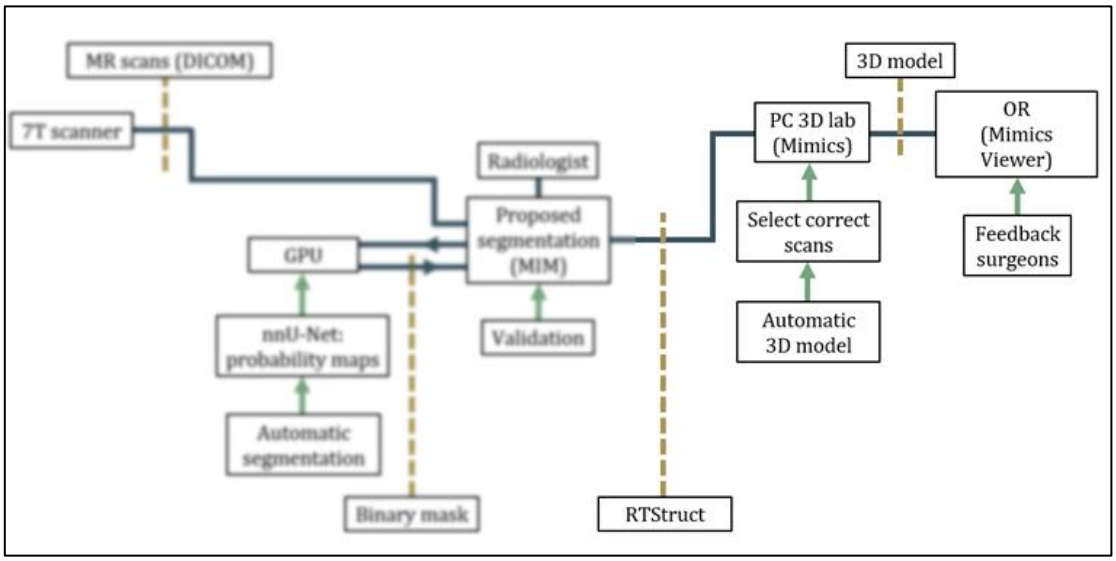


Figure 2.1: Clinical pipeline: second section

Background

To date, little is known in literature about the use of ex vivo MR imaging in intraoperative assessment of resection margins during OSCC resections. In a literature review by Barroso et al. (2021), of the eighteen found IOARM methods, none of them involved the use of MR imaging[34]. By 2023, a systematic review analysing IOARM methods by Carnicelli et al. found four studies describing the use of ex vivo MR imaging during OSCC surgery[35]. Of these four studies, none featured a logistically viable 3D visualisation method for resection margins, while all of them questioned the feasibility of ex vivo IOARM during OSCC resections[8], [36], [37], [38]. In the cases where 3D visualisation was in fact proposed, it was either based on pathological slices, or it was a manual method[5], [39]. Therefore, an automatic 3D visualisation method for OSCC specimen and resection margins is yet to be developed.

The goal of this section is to develop an intuitive 3D visualisation model of OSCC specimen and resection margins which can be automatically generated from ex vivo T2w MR scans. This automatic 3D model should be created within five minutes to support its intraoperative use. The model should also be robust, ensuring it can work with any given MR scan of an OSCC specimen.

Methods

Materials

The 3D model was developed using Mimics 26.0 and 3-matic 18.0 (Materialise NV, Leuven, Belgium). Scripting was performed in Python 3.10 using the Mimics Innovation Suite API (see Appendix B), and the final visualisation of the model was realised with Materialise Mimics Viewer (Materialise NV, Leuven, Belgium).

The script was tested on ten randomly selected transverse T2w MR scans from the dataset described in Chapter 1, ensuring unbiased selection. The tumour segmentations used for testing the 3D model were created manually in Mimics and were processed in STL format. In the actual clinical pipeline, these tumour segmentations will be generated by the nnU-Net networks.

Runtime duration

To obtain an accurate estimation of the average runtime of the model, the script was executed on ten different patient scans while the runtime was recorded using the time library in Python.

3D model process

Importing necessary files

The script initialises directories and selects and stores the relevant scans. The input scans for the model are transverse T2w MR scans, saved in DICOM format and acquired by the 7T MR scanner (Figure 2.2). These scans are extracted by means of DICOM tag criteria, separating them from other scans present in the same directory (see Table 2.1). Next, the tumour segmentation is imported, which was created manually in this study but will be created by the nnU-Net network and validated by a radiologist when clinically implemented (see also Figure 2.3). This segmentation is turned into a 3D part.

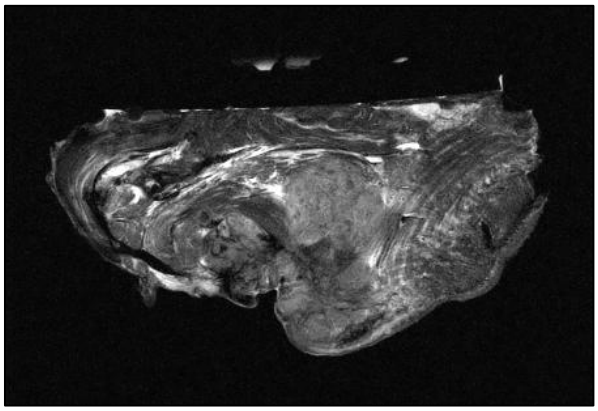


Figure 2.2: Transverse T2w MR scan of resected specimen

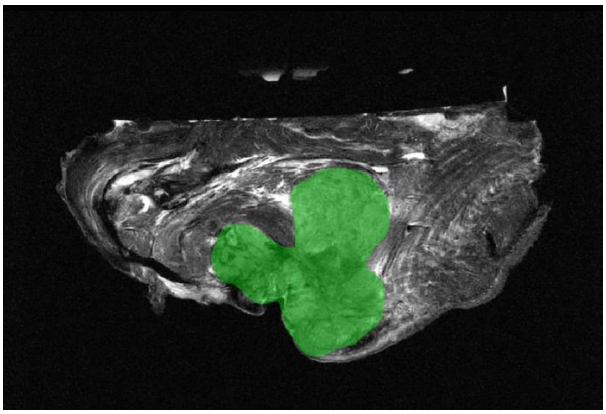


Figure 2.3: Tumour segmentation

Table 2.1: DICOM tag criteria

DICOM tag	Tag description	Required value
0002,0060	Modality	'MR'
0018,0020	Scanning Sequence	'SE'
0018,0021	Sequence Variant	'SK'
0018,0022	Scan Options	'SP' or 'OTHER'
2001,100B	Acquisition Plane	'TRANSVERSAL'

Segmentation of specimen

To enable visualisation of the resection margins relative to the resection plane, the entire resection specimen must be segmented from the scan. The automatic segmentation of the specimen is achieved using K-means clustering, which is a classical unsupervised learning algorithm that partitions data into *k* clusters based on feature similarity. This method is well-suited for this segmentation task because it effectively separates foreground structures from the background. Voxel intensity data is extracted from the scan and reduced to a single dimension before being clustered using K-means. The number of clusters is determined by the intensity range of the specific MR scan. To identify the appropriate number of clusters per scan, the script analyses the standard deviation of the voxel intensities in grayscale units. If the standard deviation is below 38, n=4 clusters are used; if the standard deviation is 38 or above, n=3 clusters are applied. This prevents inclusion of the background in the segmentation mask. The initialisation step, which controls the number of algorithm iterations, is set at ten. This enables reliable segmentation performance without significantly increasing runtime. Finally, the top two clusters are selected to define the threshold values for segmentation.

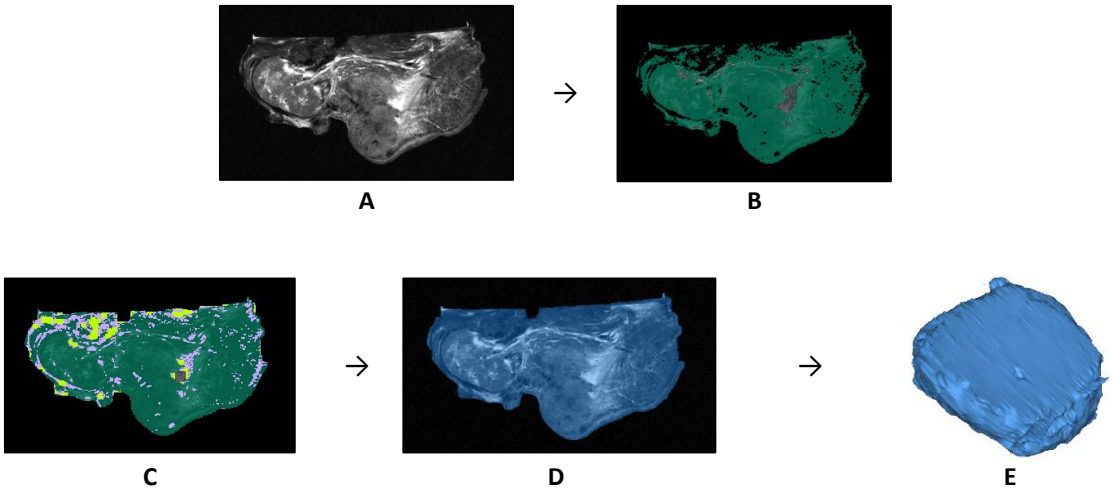


Figure 2.4: Specimen segmentation using K-means. The specimen in the original scan (A) is segmented using K-means (B). Morphological closing (C) and filling of holes is performed to reach the final specimen segmentation (D). This segmentation is then turned into a 3D part (E).

Once the K-means clustering is completed, the resulting threshold values are used to create a segmentation mask (Figure 2.4b). The morphological operation of closing for ten pixels and a connectivity of eight is then executed, filling small cavities in the mask (Figure 2.4c). Afterwards, any leftover holes in the mask are filled (Figure 2.4d) and the mask is turned into a 3D specimen part (Figure 2.4e). A transparency of 20% is added to the specimen part, so that the tumour part is visible inside the specimen part in the final 3D model. Smoothing is applied to the specimen part during 500 iterations. Shrinkage of the part is compensated using a specialised function so that the outer edges of the part are not affected. This is essential to preserve the exact edges of the specimen, as this will directly determine the resection margin status in the final 3D model.

Exporting parts from Mimics and importing them into 3-matic

The tumour and specimen parts are exported from Mimics and imported into 3-matic. Here, a validation step assures that exactly two parts were provided by the script.

Orientation (3-matic)

Clear visualisation of the orientation of the specimen is an essential part of the final 3D model to enable adequate assessment of resection margins. The visualised components in the 3D model should be easily translatable to the real-life situation in the OR. As the orientation of the specimen in relation to the patient may differ from the orientation of the specimen in the 7T scanner, an interactive console is built into the script to account for these potential discrepancies. A dialog window appears in which the researchers link each scan direction to the anatomical direction inside the patient (Figure 2.5).

Table 2.2: Anatomical and scan directions

Anatomical directions in patient					
Medial		Cranial		Anterior	
Scan directions					
X	-X	Y	-Y	Z	-Z

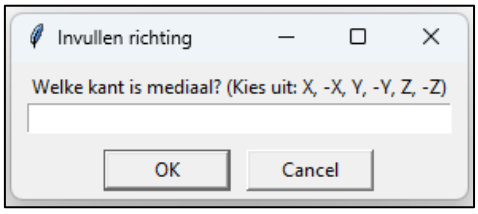


Figure 2.5: Dialog window for direction input

The direction input is used to create orientation arrows, which point in each of the three anatomical directions (Figure 2.6). In the final 3D model, the red arrow points towards medial, the green arrow points towards cranial and the blue arrow points towards anterior. These colour and direction combinations are always kept the same so that the surgeon can quickly determine each direction by means of the colours of the arrows. Nevertheless, the anatomical direction is also printed on the side of each arrow for added clarity.

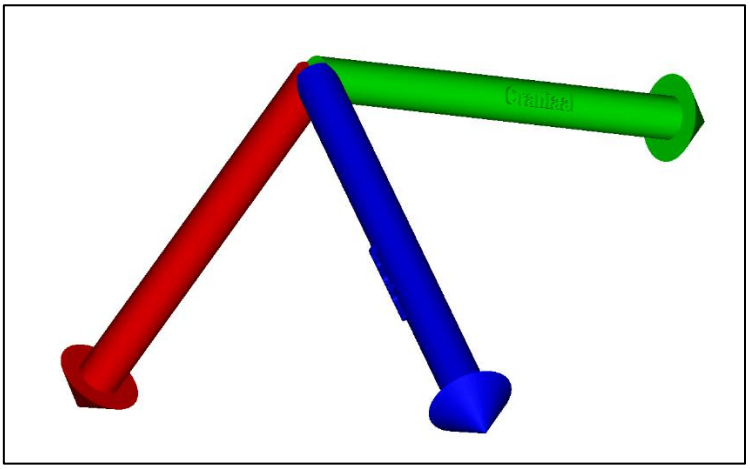


Figure 2.6: Orientation arrows

Remeshing and 5 mm margin creation (3-matic)

Before geometric operations can be applied to the tumour part, the tumour part is uniformly remeshed to create a more robust surface mesh. During remeshing, a target triangle edge length of 0.2 mm is

applied. This value is chosen because it serves as an acceptable trade-off between surface mesh precision and efficient script runtime. Additionally, surface contours are preserved so that geometry along surface contours is unchanged.

The 5 mm resection margin surface mesh is created using a wrapping function on the tumour part. This function applies a 5 mm margin around the tumour and forms the resection margin part (Figure 2.7). Within this function, a gap closing distance of 0.5 mm is applied while the smallest detail is also set at 0.5 mm. These values are based on the slice thickness of the original MR scan and prevent irregularities in the created resection margin.

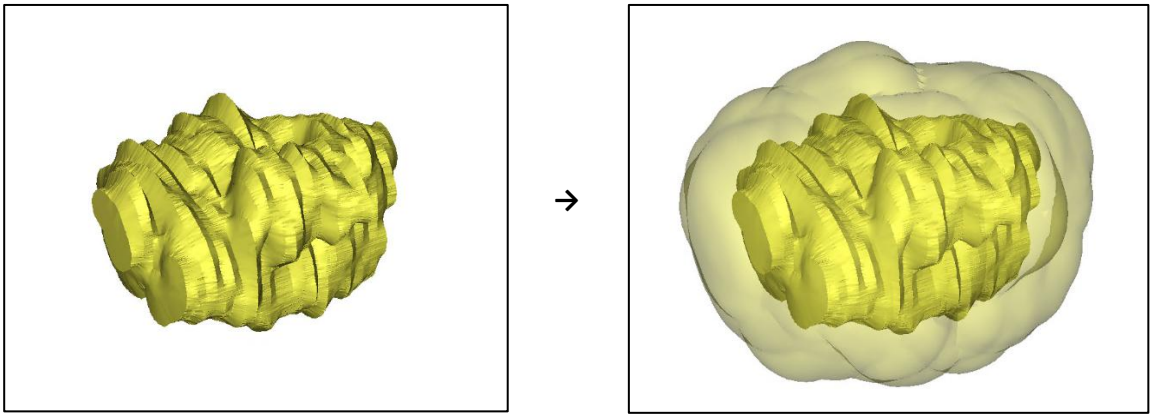


Figure 2.7: 5 mm margin around tumour part

Detection of irradical or close margins (3-matic)

To determine whether inadequate margins are present, the specimen part is subtracted from the resection margin part. This operation returns all sections of the resection margin part which were situated outside of the specimen part and could therefore be classed as inadequate margins (Figure 2.8). Each of these sections is saved as an individual part. The first, largest inadequate margin is assumed to always be the side of the tumour closest to the mucosal layer, which is where OSCC originates. While the tumour will always be within 5 mm of the tissue edge on this side, it is not part of the resection plane and will therefore not be classed as inadequate. This 'inadequate' margin is singled out and saved as 'mucosal margin', while the other remaining margins are saved as inadequate.

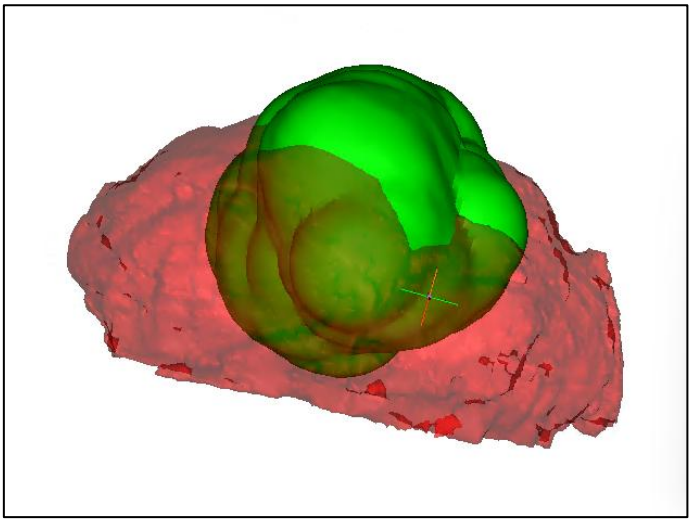


Figure 2.8: Subtracting specimen (red) from 5 mm resection margin (green)

Max distance calculation (3-matic)

The maximum depth of each inadequate margin is quantified and displayed in the 3D model, to inform surgeons about how much tissue should still be removed to reach an adequate margin. To calculate this maximum depth, the triangle vertex data are first extracted from the specimen and the vertices are structured using a k-d tree for fast processing. Next, the triangle vertex data of each inadequate margin is retrieved, and its corresponding triangle centroids are calculated to be used as sample points per triangle. The distance between each specimen point and each individual inadequate margin point is then calculated to find the maximum distance per inadequate margin in mm. This indicates the maximum depth of each inadequate margin and therefore the minimum thickness of tissue which should be removed during re-resection. The maximum distance is visualised with a label displaying the distance in mm and a red line within each inadequate margin (Figure 2.9).

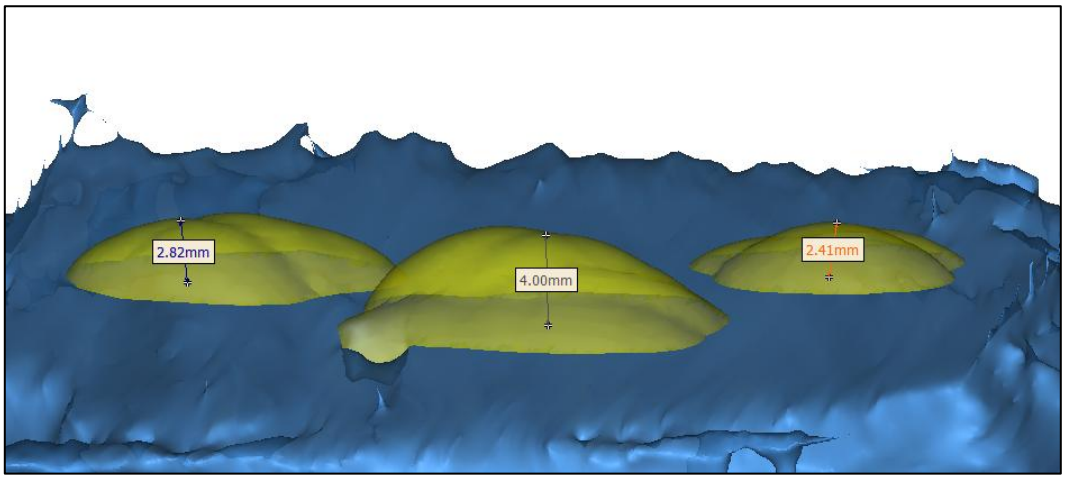


Figure 2.9: Depth per inadequate margin

Exporting parts from 3-matic and importing them into Mimics

The inadequate margins are exported back to Mimics (if they exist), together with the orientation arrows, margin thickness visualisation parts and a duplicate part displaying the complete resection margin.

Final visualisation

The maximum distance visualisation and orientation arrow parts are processed and edited to enable similar visualisation in Mimics. Inadequate margins (if present) are given a yellow colour. The tumour part is given an orange colour so that it is easily distinguishable from the margin parts. If no inadequate margins are found, the tumour is given a green colour, and a message will be displayed saying that all margins were adequate. The specimen part is given a blue colour and maintains its 20% transparency set during its segmentation (Figure 2.10).

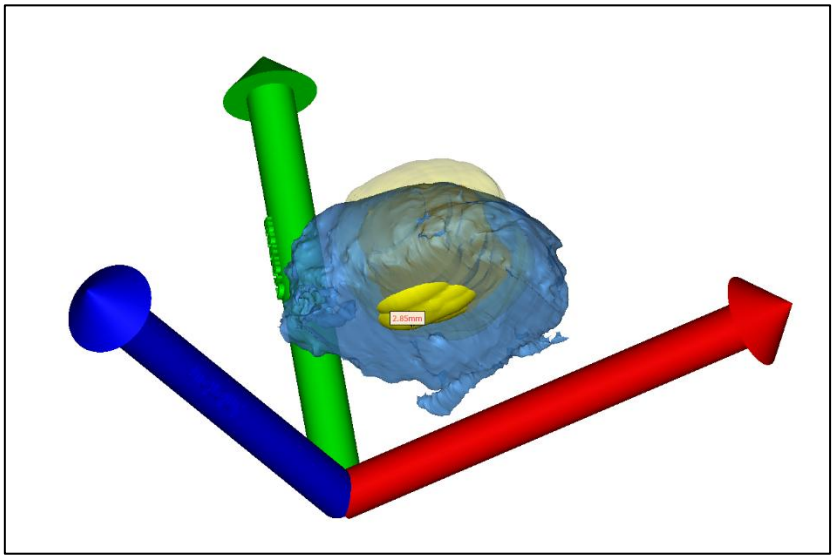


Figure 2.10: 3D model with specimen (blue), tumour (orange), inadequate margin (yellow), & complete resection margin (transparent yellow)

In the final 3D model, an additional part is created of the complete resection margin to provide extra clarification for the surgeons if needed. This part is given a yellow colour, and its transparency is set to 80% so that the other 3D parts are still visible. Lastly, all parts are given Dutch names for maximum clarity for the surgeons.

Results

The average script runtime was 4.2 minutes, with a median of 4.3 minutes (range 2.5 – 6.3 minutes). The 3D models of only two out of the ten patients exceeded the five-minute goal, which was due to larger inadequate margin volumes requiring more computational power. These cases had runtimes of 5.3 and 6.3 minutes.

Figure 2.11 provides an overview of what the final model looks like when presented in the OR in Mimics Viewer. As well as the 3D model, all 2D views of the specimen are presented with each 3D volume superimposed as a mask. On the right-hand side, each anatomical direction can be found with each corresponding colour for convenience. Also, each separate component of the 3D model can be shown, hidden or made transparent to the user's liking.

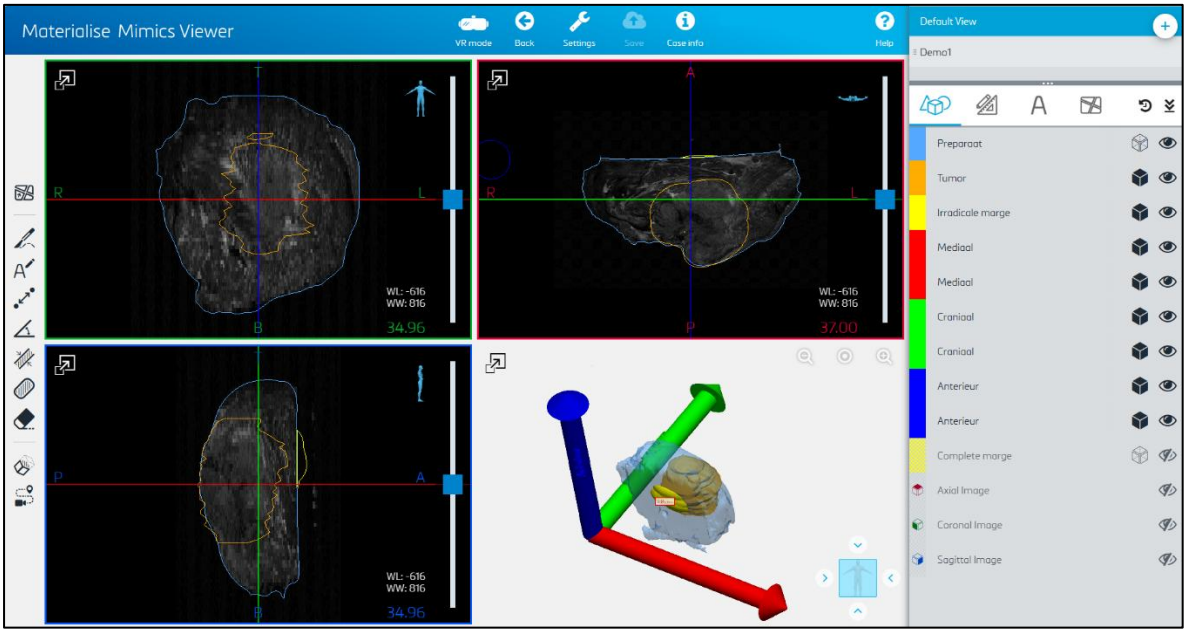


Figure 2.11: 3D model presentation as seen in the OR

Discussion

This study proposes an automatic, intuitive 3D visualisation method of OSCC specimens and resection margins which can generally be created within five minutes. This model can be used intraoperatively by surgeons to determine the location of potential re-resections for proper margin control. The model only requires an OSCC segmentation and MR scan and is set up using imaging software with CE Marking approval, which makes it feasible for clinical implementation.

Looking at the results, the average script runtime of 4.2 minutes remained well within the predefined five-minute goal to ensure clinical applicability of the 3D model. The two cases out of then that exceeded the target threshold (5.4 and 6.3 minutes), involved specimens with relatively large volumes of inadequate margins. This likely increased required computational power and thereby prolonged the runtime. Even though they exceeded the threshold, the two cases still fell within an acceptable range for clinical applicability and would still be a dramatic increase in efficiency compared to manually created 3D models. Future perspectives could focus on finding more efficient maximum margin depth calculations than the method presented in this section, the further accelerate 3D model creation.

An added value of use of the model is not only where margins were inadequate, but also where too much healthy tissue might have been resected. This is not reversible during re-resections, however it may help the surgeons to be more conservative in certain anatomical regions for future patients.

Though the proposed 3D model seems promising for clinical implementation, there are still several limitations that should be considered. First, as the inadequate margins in the model are determined by subtracting the specimen from the 5 mm resection margin, the first, largest margin near the mucosal layer (mucosal margin) will always be a factor. At the mucosa, the 5 mm margin will always extend past the specimen. However, this location is not part of the resection plane, meaning that the mucosal margin cannot be ‘inadequate’. In the model, it is assumed that this mucosal margin will never involve the resection plane. Though not encountered during this study, in theory it could be possible that this mucosal margin extends through the resection plane. In the current model, these inadequate margins would be missed. To overcome this, a future model would indicate the transition of mucosa to resection plane and solve the problem of assuming that the largest margin is always located near the mucosal layer.

Currently, it is assumed that a 5 mm margin on MR is sufficient to achieve the 5 mm histopathological margin needed for an adequate resection. However, the specimen can shrink and deform during pathological fixation and cutting. Consequently, it could be possible that a larger margin should be applied on MR to compensate for this tissue shrinkage. Both the degree of tissue shrinkage and the optimal margin to be applied on MR will still be quantified in the 7Tex study. An advantage of the current 3D model is that the margin is easily adjustable if necessary.

Between specimen scanning and resection, deformation of the specimen is negligible. The specimen is fixated in the cylinder with careful attention as to not stretch the specimen in any way. Deformation during scanning could happen if the specimen is too large to fit properly in the cylinder and the top of the specimen presses again the top of the cylinder. This too is carefully checked before including the specimen in the study.

The anatomical direction input is the only manual process required in this 3D model pipeline, which introduces a risk of error and possible increase in runtime. This step is however crucial for reliable orientation preservation. As the scanned OSCC specimens consist of irregularly sized, soft tissue masses without any anatomical landmarks, it is not feasible to determine specimen orientation based on its dimensions. For example, the longest side of the specimen might usually be anterior/posterior,

however for some cases it will be medial/lateral, depending on indication of the Pathology department. The specimen shape is also not a reliable factor, as OSCC specimens can have highly irregular shapes, while other specimens may appear completely circular. These factors impede any form of automatic recognition of specimen orientation, which is why the decision is made to complete this step manually.

Future perspectives

3-matic offers functionality which visualises the depth of the inadequate margins using a colour map (see Figure 2.12). This would grant the surgeons a more intuitive and complete picture of how much tissue needs to be removed during re-resection, compared to the current workflow where only the maximum depth is calculated. This functionality is however not translatable to Mimics Viewer, which means that it can not be used intraoperatively. This issue has been communicated to Materialise and future updates might enable this approach. Implementation of this functionality would result in a more robust, intuitive and detailed 3D model and give surgeons an even better understanding of how to approach their re-resections.

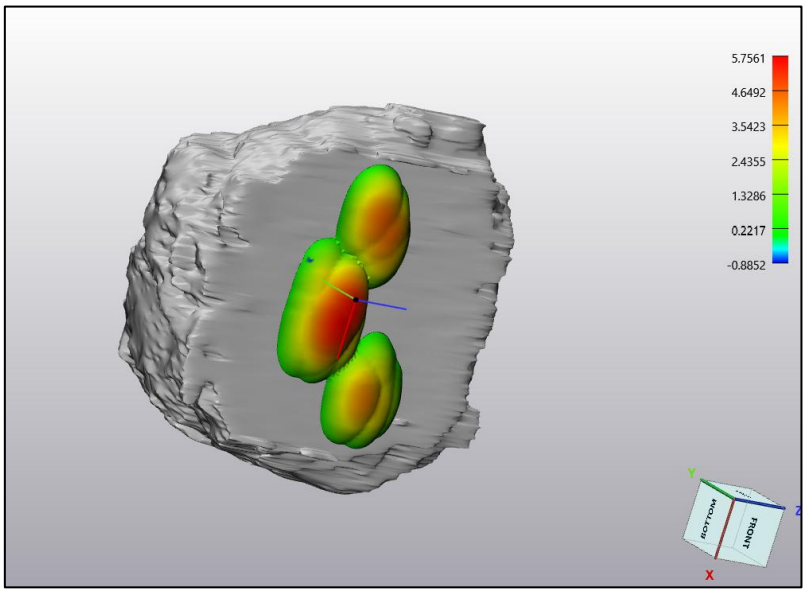


Figure 2.12: Colour map functionality in 3-matic

Adding a HoloLens (Microsoft Corp., Redmond, WA, USA) to the clinical workflow could improve resection margin feedback to the surgeons in the future. Instead of gauging their resection margins via a screen in the OR, the surgeons could visualise the 3D model near the resection plane with real-life scaling of the specimen. This could further improve the usability of the model as surgeons would get a more realistic representation of the exact resection margins, contributing to more accurate re-resections.

Conclusion

This study presents an intuitive and clinically feasible method for 3D visualisation of OSCC specimens and resection margins. By enabling automatic 3D model generation of resection margins based on 7T MR scans, it contributes directly to the development of a clinically implementable IOARM pipeline.

Chapter 3: 3D model evaluation

Introduction

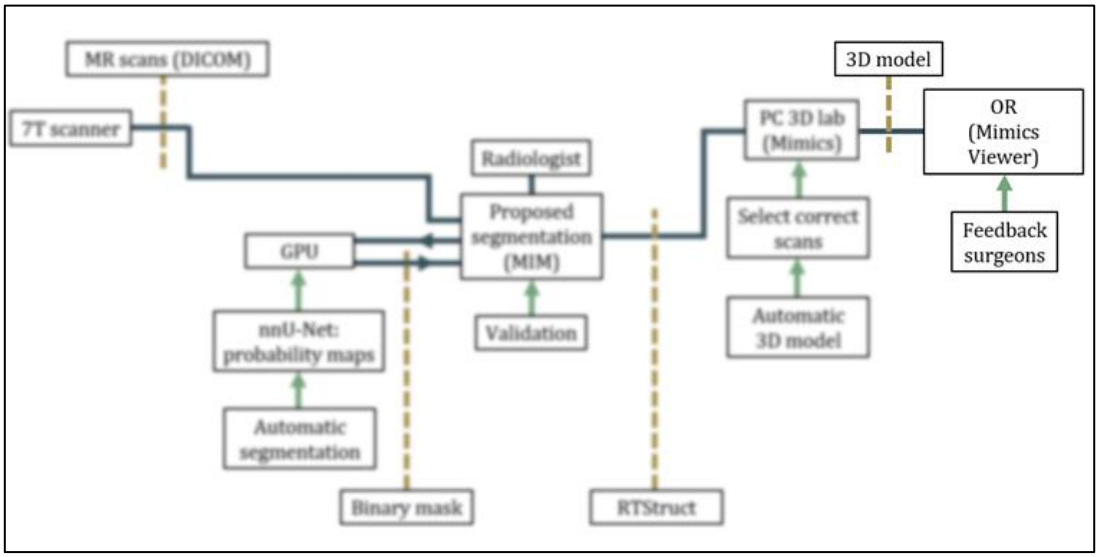


Figure 3.1: Clinical pipeline: 3D model usage

The 3D model presented in the previous chapter will be integrated into the clinical pipeline proposed in this study. Because the model is developed from the ground up and lacks comparable tools currently used during OSCC resections, surgeon's perceptions of its usability remain unknown. Evaluating the model's usability from the end-user's perspective is crucial as it may highlight potential shortcomings or practical implications before clinical implementation. Additionally, such an evaluation helps surgeons to familiarise themselves with the model prior to clinical implementation, thereby facilitating a smoother implementation process.

The goal of this section is to establish an objective benchmark of the model's usability. This is achieved through consultation of all surgeons performing OSCC resections at the UMC Utrecht. Their feedback enables a reliable assessment of the model prior to its clinical implementation.

Methods

The 3D model's usability was evaluated through demonstrations. Each demonstration was performed one-on-one with each surgeon and started with a short presentation about the project and the essential components in the 3D model.

Two clinical cases were created to simulate real situations in the OR, where the 3D model was used to determine the re-resection (see Appendix C). Both cases supplied tumour location and dimensions. In a schematic drawing of the tongue and mouth, different wound beds were presented for each case (Figure 3.2). The tumours and wound beds differed in dimensions to simulate a realistic variety in case characteristics. Usage instructions like zooming and hiding components in Mimics Viewer were also included in the demonstration.

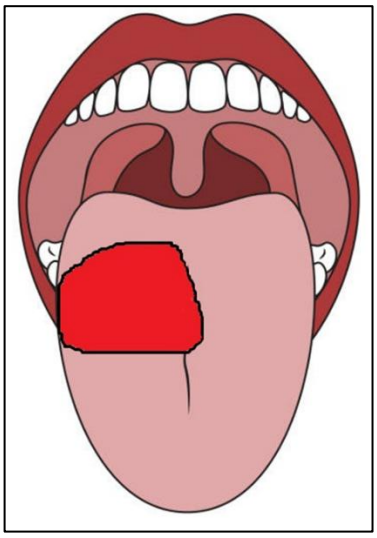


Figure 3.2: Schematic drawing of tongue and wound bed (red) of Case 1

The surgeon was instructed to assess the resection margins using the 3D model, after which they had to mark on the drawing where they would perform the re-resection. For both cases, the surgeon filled in the depth they would apply during this re-resection. After the surgeon completed both cases, they filled in the System Usability Scale (SUS) based on their experience with the 3D model.

The SUS has been a popular measurement of perceived usability for more than forty years[40]. It is the most widely used standardised questionnaire to assess usability of a great number of applications. The SUS features ten statements, five positive and five negative, on which the user rates their rate of agreement (Table 3.1).

Table 3.1: System Usability Scale

			Strongly disagree 1	2	3	4	Strongly agree 5
1	I think that I would like to use this system frequently.		◯	◯	◯	◯	◯
2	I found the system unnecessarily complex.		◯	◯	◯	◯	◯
3	I thought the system was easy to use.		◯	◯	◯	◯	◯
4	I think that I would need the support of a technical person to be able to use this system.		◯	◯	◯	◯	◯
5	I found the various functions in this system were well integrated.		◯	◯	◯	◯	◯
6	I thought there was too much inconsistency in this system.		◯	◯	◯	◯	◯
7	I would imagine that most people would learn to use this system very quickly.		◯	◯	◯	◯	◯
8	I found the system very cumbersome to use.		◯	◯	◯	◯	◯
9	I felt very confident using the system.		◯	◯	◯	◯	◯
10	I needed to learn a lot of things before I could get going with this system.		◯	◯	◯	◯	◯

The SUS score is calculated by the following formula:

$$SUS\ score = ((X - 5) + (25 - Y)) * 2.5$$

Where X equals the sum of points for all odd-numbered questions and Y equals the sum of points for all even-numbered questions. The result is a score between 0 and 100 which denotes the perceived usability.

After the SUS was filled in by the surgeon, additional feedback was asked about what improvements could be made to the 3D model.

Results

The SUS scores of all eight demonstrations are presented in Figure 3.3. The average was 74.4 (dashed line) (range 60 – 90). Table 3.2 features each individual scoring per statement.

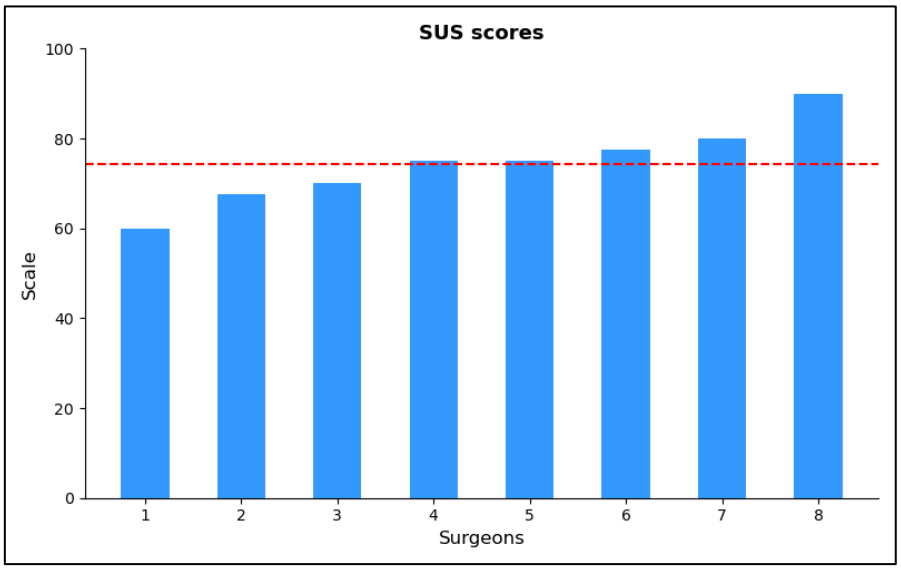


Figure 3.3: SUS scores of all eight surgeons

Table 3.2: Individual SUS statement results

Statement	Surgeon 1	Surgeon 2	Surgeon 3	Surgeon 4	Surgeon 5	Surgeon 6	Surgeon 7	Surgeon 8
1	4	4	4	4	5	5	4	5
2	2	2	2	2	2	2	2	1
3	2	4	4	3	4	4	4	5
4	3	2	3	3	2	2	1	1
5	4	3	4	4	4	4	4	3
6	4	2	2	1	2	2	2	1
7	3	4	4	5	4	4	4	4
8	2	2	2	2	4	2	2	1
9	4	2	3	4	5	4	4	5
10	2	2	2	2	2	2	1	2

Discussion

The usability of the 3D model was positively evaluated by all surgeons performing OSCC resections at the UMC Utrecht. None of the surgeons suggested adjustments which would improve the 3D model. The findings in this study indicate that there are no major shortcomings in the 3D model hampering implementation in the clinical pipeline.

To interpret SUS scores like those observed in this section, Bangor et al. proposed an adjective rating scale to give meaning to and enable interpretation of individual SUS scores[41]. The adjective rating to the SUS score of the 3D model (74.4) is classed as 'good', nearing the category of 'excellent' (see Figure 3.4). This suggests that the 3D model is suitable for immediate, practical use in the intraoperative setting.

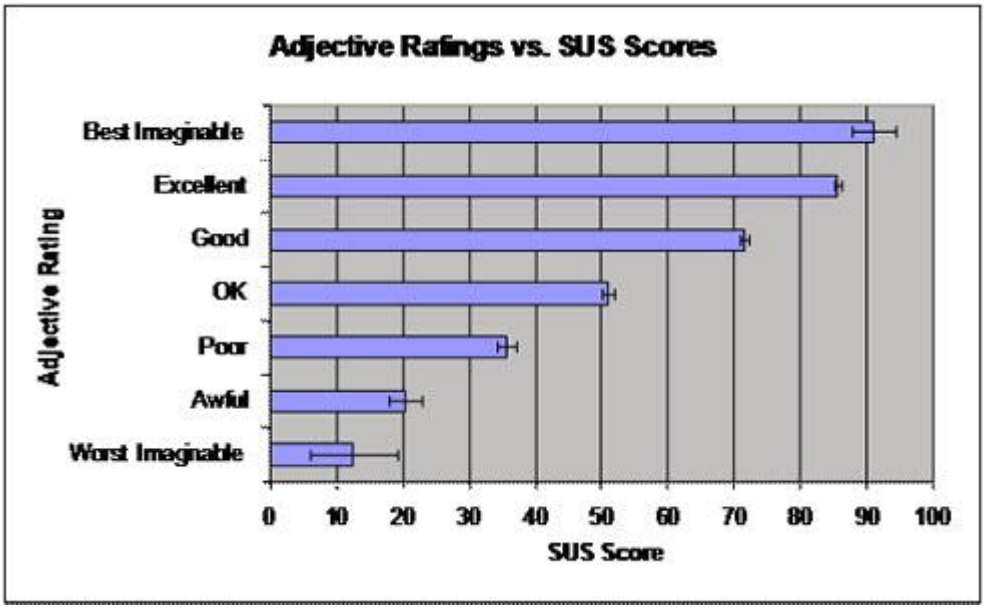


Figure 3.4: Adjective ratings vs. SUS scores[41]

SUS scores of computer-assisted surgical navigation ranging from 43 to 64 are reported in literature[42], [43]. A direct comparison is however challenging as evaluation of 3D visualisation or resection margins for OSCC surgery is not described in literature.

The most agreement was found in the second and tenth statement of the SUS (see Table 3.2). This shows that the model is not considered complex and that the model can be used without requiring much training. The first statement about whether the surgeons would like to use the 3D model frequently received the strongest agreement. This result shows that the surgeons value the 3D model as a valid addition to their practice.

A point of feedback which was offered by all of the eight questioned surgeons, was that usability would increase significantly after they gained more experience with it. The orientation arrows were most often described as the aspect which needed the most time to get accustomed to.

To continue, data was gathered from all surgeons performing OSCC resections at the UMC Utrecht, which means that the SUS score of the 3D model can be viewed as a comprehensive and reliable judgement of its usability. Also, the demonstrations were designed to simulate the real-life, intraoperative situation as closely as possible. However, the representation of the wound bed in 2D on a piece of paper made it more difficult to approach it as a 3D patient. This disconnect between the 3D model and the 2D wound bed could have accounted for slightly lower SUS scores, compared to if the

demonstrations were performed in the OR instead with for example a 3D-printed tongue. This was also confirmed by some of the surgeons.

The findings of the study show that surgeons are positive about the use of the 3D model for OSCC resections. This serves as validation that using an automatic 3D visualisation model is a feasible method to perform IOARM. In the MARGIN study, the 3D model will be used in the clinical setting to test the viability of the 7Tex clinical workflow. Performing a second evaluation of the 3D model using the SUS when this study finishes will objectively quantify how the usability will have improved as a result of the surgeons having got fully accustomed to the 3D model.

Conclusion

The automatic 3D model created for intraoperative assessment of resection margins during OSCC resections has been evaluated as 'good' by all surgeons performing OSCC resections at the UMC Utrecht. Continued use of the 3D model in the clinical setting will increase this usability rating even further.

Chapter 4: Super-resolution

Introduction

High-resolution MR imaging is clinically relevant because it enables detailed assessment of anatomical structures[44]. MR scanning using high magnetic field strengths (such as 7T) provides enhanced signal-to-noise ratio (SNR), contrast-to-noise ratio (CNR), and excellent spatial resolution[45]. A major drawback to high resolution MR imaging is the increase in slice thickness relative to the in-plane resolution. Increased resolution is therefore only provided in two out of three scanning directions, which can complicate visualisation of small anatomical details[30]. To overcome this, isotropic 3D T2w scans can be acquired which feature high resolution in all scanning directions. However, their acquisition time is significantly greater than that of anisotropic multislice T2w scans, and therefore not suitable for the clinical pipeline in this project. To save time, the current clinical pipeline relies on multislice scans in only two orientations (transverse and sagittal).

To enable high resolution in all scanning directions, using super-resolution algorithms might be a solution[46]. Therefore, this section explores a workflow in which isotropic 3D T2w scans are reconstructed from transverse and sagittal multislice T2w scans using super-resolution.

Such a workflow is clinically relevant as it surpasses the need for time-intensive acquisition of 3D T2w scans. In particular, it may improve deep-learning-based tumour segmentation. The clinical pipeline proposed in Chapter 1 employs two deep-learning networks, one for transverse and one for sagittal scans, trained in a 2D configuration. This means each network cannot learn from 3D anatomical context. By incorporating isotropic 3D scans, the segmentation network can better learn complex anatomical features of OSCC and surrounding healthy tissue.

This section evaluates the feasibility of reconstructing 3D T2w scans from anisotropic input using super-resolution methods.

Methods

An overview of the super-resolution workflow is presented in Figure 4.1. The Python script can be found in Appendix D. To simulate the reconstruction of isotropic 3D T2w scans, acquired coronal multislice T2w scans were used as reference. These coronal scans had high in-plane resolution in the coronal direction, making them suitable for evaluating the quality of super-resolution reconstructions. The coronal scans served as a benchmark to assess how well the low-resolution coronal information from the transverse and sagittal scans could be used to recover the true anatomical detail in the acquired coronal slices.

Transverse, sagittal, and coronal scans from the 7Tex dataset described in Chapter 1 were used for this section. Raw DICOM files were transformed into NIfTI format using the dcm2niix Python library. Further image processing steps were performed using the Advanced Normalization Tools (ANTs) library, a state-of-the-art medical image processing toolbox with suitable functionality for super-resolution upsampling of MR image data[47].

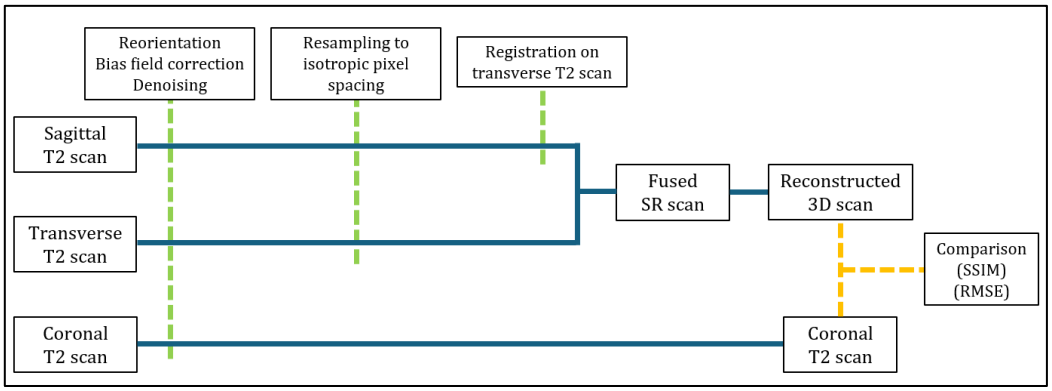


Figure 4.1: Super-resolution workflow. Sagittal, transverse and coronal MR scans were preprocessed, after which the sagittal and transverse scans were resampled to an isotropic pixel spacing based on the in-plane resolution of the coronal scan. After registering the sagittal scan to the transverse scan, both scans were fused and the end result was compared to the coronal scan using the structural similarity index measure and the root mean squared error.

Transverse, sagittal and coronal scans were preprocessed by scan orientation alignment, N4 Bias field correction and denoising[48]. The transverse and sagittal scans were resampled to match isotropic pixel spacing based on the coronal scan. If the coronal scan for example had dimensions of 0.08x0.08x1 mm, the transverse and sagittal scans were resampled to pixel spacing of 0.08x0.08x0.08 mm. After resampling, the sagittal scan was registered to the transverse scan using rigid registration. Fusion of both scans was performed by averaging the pixel values:

$$fused\ scan = \frac{(transverse\ scan + sagittal\ scan)}{2}$$

The reconstructed voxels of the transverse and sagittal scan were compared to the reference voxels of the coronal scan to quantify to what extent the coronal slices could be recovered. The reconstructed 3D scan and the coronal scan were compared using the structural similarity index measure (SSIM) and the root mean squared error (RMSE) to quantify structural and absolute differences respectively. The RMSE was also expressed as the percentage of the signal intensity of the coronal scan to also obtain a normalised metric:

$$RMSE\% = \frac{RMSE}{max\ signal\ intensity - min\ signal\ intensity} * 100$$

Results

55 MR patient scan sets were used for analysis in this section. The average SSIM between the original coronal scans and the reconstructed 3D scans was 0.766 (± 0.0842), with a median of 0.781 (see also Figure 4.2). The SSIM score distribution showed no statistical outliers.

Distribution of absolute RMSE scores is presented in Figure 4.3. The apparent outlier at RMSE = 141 corresponds to a RMSE% of 5.85%, indicating that this case does not represent a true outlier when normalised to the scan's overall intensity.

The average RMSE% of the reconstructed 3D scans and the original coronal scans was 4.16% (± 1.54), with a median of 3.84% (see also Figure 4.4). One outlier with an RMSE% of 8.56% was found, corresponding to a case in which the sagittal and transverse scans had a mismatched field of view. The SSIM for this case was 0.702, which was below the average but within the normal distribution range.

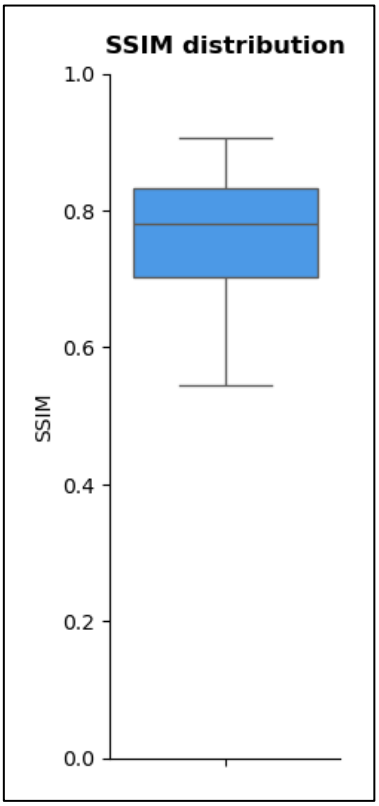


Figure 4.2: SSIM distribution

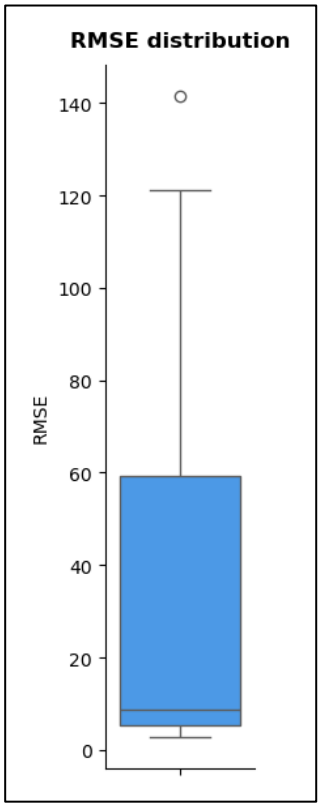


Figure 4.3: Absolute RMSE distribution

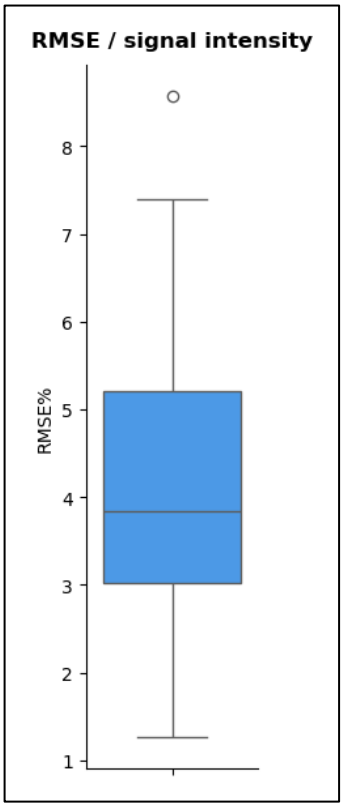


Figure 4.4: RMSE% distribution

These results indicate a decent structural similarity between the reconstructed and original coronal scans, with SSIM scores generally exceeding 0.75. The low average RMSE% further supports the accuracy of reconstruction, suggesting minimal voxel-wise intensity deviation. The single outlier is likely a result of scan acquisition mismatch rather than a methodological error.

Figure 4.5 shows the reference coronal scan (A), the reconstructed 3D scan of this same slice (D), and the transverse and sagittal scans on which the reconstructed 3D scan was based (B and C). The SSIM and RMSE scores of this patient scan set were 0.854 and 3.36% respectively.

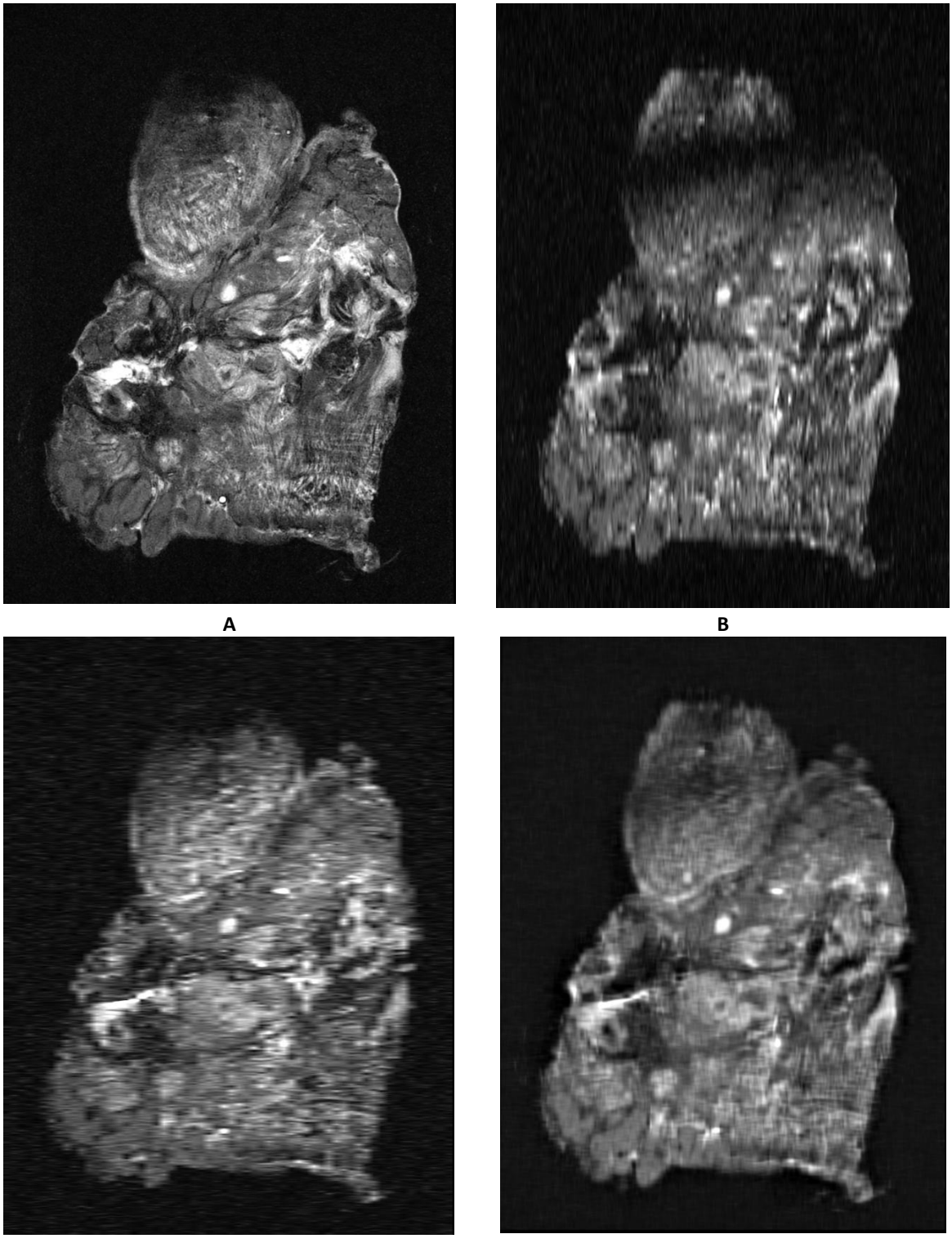


Figure 4.5: Coronal planes scanned in coronal direction (A), transverse direction (B), sagittal direction (C) and reconstructed using super-resolution (D). A notable difference in resolution is seen between the coronal scan and the transverse and sagittal scans. The reconstructed scan shows recovery of several anatomical structures which were hardly visible on the transverse and sagittal scans.

Discussion

This section demonstrated that isotropic 3D T2w scans can be adequately recovered through super-resolution upsampling of high-resolution transverse and sagittal multislice T2w scans. Using the ANTs Python library, promising SSIM and RMSE results were achieved on resected OSCC specimens scanned at 7T, suggesting meaningful anatomical detail can be recovered without the need for dedicated 3D acquisitions.

To the best of our knowledge, super-resolution reconstruction metrics on 7T MR scans of resected OSCC tumour specimens are not found in literature. Papers by Du et al. and Jiang et al. investigated super-resolution-based reconstruction of 3 tesla brain MR scans[49], [50]. The observed SSIM scores in these papers are particularly higher than those found in this section. This could however be explained by their more advanced and deep-learning based algorithms. Nevertheless, although our SSIM scores are lower, they still reflect substantial image consistency and contrast recovery.

The SSIM metric measures image similarity based on image luminance, contrast and correlation, and is considered a consistent metric with human visual system and perception[49]. The observed average SSIM score in this section (0.766 ± 0.0842) illustrates how image contrast is decently recovered through the proposed super-resolution method. This is supported by visual assessments (see Figure 4.5), where improved contrast is evident compared to the individual transverse and sagittal inputs. The RMSE% scores paint a similar picture with an average of 4.16% (± 1.54), and only one case exceeding 8%, likely due to a mismatch in fields of view between transverse and sagittal scans. These observed findings highlight the potential of super-resolution techniques to reduce scanning time and increase clinical, intraoperative feasibility.

With the transverse, sagittal, and reconstructed 3D scans, high resolution is provided in all three scanning directions. These three scan types must be merged to reach a definitive isotropic 3D scan which is fit for clinical use.

The perceived increased contrast in the reconstructed 3D scans means that radiologists will be able to delineate OSCC tissue more confidently in all three directions, compared to if only the transverse and sagittal scans would be visible. Also, isotropic 3D scans of OSCC specimens will enable deep-learning networks to predict tumour tissue more accurately than if only the transverse and sagittal scans are provided, as such networks then have anatomical depth information available to learn 3D patterns as well as 2D. Future studies must explore the effect of reconstructed 3D scans on tumour delineation accuracy, both for radiologists and for deep-learning networks.

The super-resolution methodology applied in this section is based on fundamental image processing tools from the ANTs library. This library also features deep-learning functionalities which may yield improved performance. Future studies should explore these capabilities to approach the quality of acquired 3D T2w scans, as the extensive scanning time of these scans impedes clinical applicability.

Conclusion

This study demonstrates that isotropic 3D T2w scans can be reconstructed from high-resolution multislice MR scans using super-resolution techniques. These findings highlight the potential of super-resolution to enhance MR image quality for IOARM applications without relying on time-intensive 3D acquisitions.

General discussion

This study presents a clinical pipeline for intraoperative assessment of resection margins using automatic tumour segmentation and 3D-visualisation based on ex vivo 7T MR scans of OSCC resection specimens. This pipeline provides a technical foundation for the 7Tex study leading up to the MARGIN study, where this pipeline will be evaluated in the clinical setting.

Application of nnU-Net deep-learning networks produced acceptable segmentation results for the proposed clinical pipeline. These automatic segmentations will support the radiologist to quickly delineate OSCC tissue on MR. Also, the automatic 3D model for resection margin visualisation can on average be generated within five minutes to further speed up the clinical pipeline. The 3D model is considered decently usable by all surgeons performing OSCC resections at the UMC Utrecht, which further improves the potential of the clinical pipeline. Lastly, super-resolution is proven to be a promising technique to reconstruct isotropic 3D scans from transverse and sagittal MR scans. This highlights the potential to increase OSCC delineation performance by a radiologist as well as deep-learning networks, while overcoming the long scanning duration associated with isotropic 3D MR scans.

The use of 7T MR scanning comes with logistical challenges, as can be seen in the Methods section of Chapter 1. An important consideration is whether using a 3T MR scanner would be a better fit to the pipeline as it eliminates the need to transport the specimen outside the hospital for scanning. For large tumours, this could be the preferred option, however smaller tumours likely require the improved resolution and SNR achieved through the 7T scanner for proper assessment of resection margins.

Before the pipeline can be properly tested clinically, some logistical challenges must still be resolved. First, a secure connection must be established to send the MR scans from the scanner to the hospital network. Next, adjustments must be made within the MIM software so that the scans can be easily imported and the nnU-Net framework can be accessed to perform the segmentation tasks. Once the segmentation is reviewed and finalised by the radiologist, an automatic workflow is required to import the scan and segmentation into the 3D model generation script. Once these technical steps are completed, the full pipeline can be tested. This will show whether the whole workflow meets the time constraints necessary for intraoperative use.

Along with the T2w scans, diffusion-weighted (DWI) MR scans of the specimen are also acquired in the current setup. Studies by Carcinelli et al. and Heidkamp et al. have shown how these DWI scans provided the highest contrast between OSCC and healthy tissue on ex vivo 3T scans[8], [35]. This suggests that incorporating DWI scans into the nnU-Net framework may significantly enhance segmentation accuracy in the future. This is particularly relevant given the limited size of the 7Tex dataset, as it will be difficult to improve the observed segmentation accuracy in this study with just the T2w scans alone.

Lastly, to properly evaluate the clinical benefit of the proposed pipeline in this study, a randomised controlled trial would ultimately be required. Such a study could assess the effect on OSCC recurrence rates and adjuvant therapy for patients having undergone OSCC resection thereby providing evidence for its true clinical benefit.

Conclusion

This study presents a clinical pipeline for IOARM in OSCC surgery, combining automatic segmentation and intuitive 3D visualisation of ex vivo 7T MR scans. While not yet part of the core pipeline, the positive evaluation of the 3D visualisation by head and neck surgeons along with the promising results from super-resolution reconstruction further support the clinical potential and future implementation of the IOARM workflow.

References

[1] D. E. Johnson, B. Burtness, C. R. Leemans, V. W. Y. Lui, J. E. Bauman, and J. R. Grandis, 'Head and neck squamous cell carcinoma', *Nat. Rev. Dis. Primer*, vol. 6, no. 1, p. 92, Nov. 2020, doi: 10.1038/s41572-020-00224-3.

[2] A. W. Y. Chai, K. P. Lim, and S. C. Cheong, 'Translational genomics and recent advances in oral squamous cell carcinoma', *Semin. Cancer Biol.*, vol. 61, pp. 71–83, Apr. 2020, doi: 10.1016/j.semcancer.2019.09.011.

[3] L. Feller and J. Lemmer, 'Oral Squamous Cell Carcinoma: Epidemiology, Clinical Presentation and Treatment', *J. Cancer Ther.*, vol. 03, no. 04, pp. 263–268, 2012, doi: 10.4236/jct.2012.34037.

[4] National Cancer Institute, 'Oral cavity'. 2024. [Online]. Available: https://www.cancer.gov/publications/dictionaries/cancer-terms/def/oral-cavity

[5] K. J. De Koning *et al.*, 'Feasibility of an MR-based digital specimen for tongue cancer resection specimens: a novel approach for margin evaluation', *Front. Oncol.*, vol. 14, p. 1342857, Mar. 2024, doi: 10.3389/fonc.2024.1342857.

[6] C.-T. Liao *et al.*, 'Analysis of Risk Factors of Predictive Local Tumor Control in Oral Cavity Cancer', *Ann. Surg. Oncol.*, vol. 15, no. 3, pp. 915–922, Mar. 2008, doi: 10.1245/s10434-007-9761-5.

[7] R. W. H. Smits *et al.*, 'Resection margins in oral cancer surgery: Room for improvement', *Head Neck*, vol. 38, no. S1, Apr. 2016, doi: 10.1002/hed.24075.

[8] J. Heidkamp *et al.*, 'Assessment of surgical tumor-free resection margins in fresh squamous-cell carcinoma resection specimens of the tongue using a clinical MRI system', *Head Neck*, vol. 42, no. 8, pp. 2039–2049, Aug. 2020, doi: 10.1002/hed.26125.

[9] K. J. De Koning *et al.*, 'Feasibility study of ultrasound-guided resection of tongue cancer with immediate specimen examination to improve margin control – Comparison with conventional treatment', *Oral Oncol.*, vol. 116, p. 105249, May 2021, doi: 10.1016/j.oraloncology.2021.105249.

[10] W. M. Mendenhall *et al.*, 'Postoperative Radiotherapy for Squamous Cell Carcinoma of the Head and Neck', *Clin. Med. Res.*, vol. 4, no. 3, pp. 200–208, Sep. 2006, doi: 10.3121/cmr.4.3.200.

[11] J. Solomon *et al.*, 'The impact of close surgical margins on recurrence in oral squamous cell carcinoma', *J. Otolaryngol. - Head Neck Surg.*, vol. 50, no. 1, p. 9, Jan. 2021, doi: 10.1186/s40463-020-00483-w.

[12] E. A. Dik, S. M. Willems, N. A. Ipenburg, S. O. Adriaansens, A. J. W. P. Rosenberg, and R. J. J. Van Es, 'Resection of early oral squamous cell carcinoma with positive or close margins: Relevance of adjuvant treatment in relation to local recurrence', *Oral Oncol.*, vol. 50, no. 6, pp. 611–615, Jun. 2014, doi: 10.1016/j.oraloncology.2014.02.014.

[13] P. Jehn *et al.*, 'Physical and Psychological Impairments Associated with Mucositis after Oral Cancer Treatment and Their Impact on Quality of Life', *Oncol. Res. Treat.*, vol. 42, no. 6, pp. 342–349, 2019, doi: 10.1159/000499720.

[14] Z. Yang, W. Chen, H. Huang, C. Pan, and J. Li, 'Quality of Life of Patients With Tongue Cancer 1 Year After Surgery', *J. Oral Maxillofac. Surg.*, vol. 68, no. 9, pp. 2164–2168, Sep. 2010, doi: 10.1016/j.joms.2009.09.048.

[15] R. W. H. Smits *et al.*, 'Intraoperative Assessment of the Resection Specimen Facilitates Achievement of Adequate Margins in Oral Carcinoma', *Front. Oncol.*, vol. 10, p. 614593, Dec. 2020, doi: 10.3389/fonc.2020.614593.

[16] 'Definitie resectiemarge mondholtecarcinomen', Federatie Medisch Specialisten, Sep. 2023. Accessed: Jun. 11, 2025. [Online]. Available: https://richtlijnendatabase.nl/richtlijn/hoofd-halstumoren_sept_2023/behandeling_bij_hht_2023/mondholtecarcinoom/definitie_resectiemarge_mondholtecarcinomen.html

[17] Y. Aaboubout *et al.*, 'INTRAOPERATIVE ASSESSMENT OF RESECTION MARGINS BASED ON RAMAN SPECTROSCOPY IN OCSCC SURGERY', *Oral Surg. Oral Med. Oral Pathol. Oral Radiol.*, vol. 132, no. 1, pp. e35–e36, Jul. 2021, doi: 10.1016/j.oooo.2021.03.132.

[18] N. M. Bekedam *et al.*, 'Intra-operative resection margin model of tongue carcinoma using 3D reconstructed ultrasound', *Adv. Oral Maxillofac. Surg.*, vol. 4, p. 100154, Oct. 2021, doi: 10.1016/j.adoms.2021.100154.
[19] K. J. De Koning, C. M. E. M. Adriaansens, R. Noorlag, R. De Bree, and R. J. J. Van Es, 'Intraoperative Techniques That Define the Mucosal Margins of Oral Cancer In-Vivo: A Systematic Review', *Cancers*, vol. 16, no. 6, p. 1148, Mar. 2024, doi: 10.3390/cancers16061148.
[20] V. Panzarella *et al.*, 'Site-Coded Oral Squamous Cell Carcinoma Evaluation by Optical Coherence Tomography (OCT): A Descriptive Pilot Study', *Cancers*, vol. 14, no. 23, p. 5916, Nov. 2022, doi: 10.3390/cancers14235916.
[21] J. Heidkamp *et al.*, 'MRI evaluation of vulvar squamous-cell carcinoma in fresh radical local excision specimens for cancer localization and prediction of surgical tumor-free margins', *NMR Biomed.*, vol. 32, no. 1, Jan. 2019, doi: 10.1002/nbm.4025.
[22] H.-P. Chan, R. K. Samala, L. M. Hadjiiski, and C. Zhou, 'Deep Learning in Medical Image Analysis', in *Deep Learning in Medical Image Analysis*, vol. 1213, G. Lee and H. Fujita, Eds., in Advances in Experimental Medicine and Biology, vol. 1213. , Cham: Springer International Publishing, 2020, pp. 3–21. doi: 10.1007/978-3-030-33128-3_1.
[23] Y. M. Moe *et al.*, 'Deep learning for automatic tumour segmentation in PET/CT images of patients with head and neck cancers', 2019, *arXiv*. doi: 10.48550/ARXIV.1908.00841.
[24] V. Shavlokhova *et al.*, 'Deep Learning on Oral Squamous Cell Carcinoma Ex Vivo Fluorescent Confocal Microscopy Data: A Feasibility Study', *J. Clin. Med.*, vol. 10, no. 22, p. 5326, Nov. 2021, doi: 10.3390/jcm10225326.
[25] A. Diamant, A. Chatterjee, M. Vallières, G. Shenouda, and J. Seuntjens, 'Deep learning in head & neck cancer outcome prediction', *Sci. Rep.*, vol. 9, no. 1, p. 2764, Feb. 2019, doi: 10.1038/s41598-019-39206-1.
[26] A. A. K. Abdel Razek, R. Khaled, E. Helmy, A. Naglah, A. AbdelKhalek, and A. El-Baz, 'Artificial Intelligence and Deep Learning of Head and Neck Cancer', *Magn. Reson. Imaging Clin. N. Am.*, vol. 30, no. 1, pp. 81–94, Feb. 2022, doi: 10.1016/j.mric.2021.06.016.
[27] F. Isensee, P. F. Jaeger, S. A. A. Kohl, J. Petersen, and K. H. Maier-Hein, 'nnU-Net: a self-configuring method for deep learning-based biomedical image segmentation', *Nat. Methods*, vol. 18, no. 2, pp. 203–211, Feb. 2021, doi: 10.1038/s41592-020-01008-z.
[28] K. A. Wahid *et al.*, 'Evaluation of deep learning-based multiparametric MRI oropharyngeal primary tumor auto-segmentation and investigation of input channel effects: Results from a prospective imaging registry', *Clin. Transl. Radiat. Oncol.*, vol. 32, pp. 6–14, Jan. 2022, doi: 10.1016/j.ctro.2021.10.003.
[29] G. H. Bol, A. N. T. J. Kotte, U. A. Van Der Heide, and J. J. W. Lagendijk, 'Simultaneous multi-modality ROI delineation in clinical practice', *Comput. Methods Programs Biomed.*, vol. 96, no. 2, pp. 133–140, Nov. 2009, doi: 10.1016/j.cmpb.2009.04.008.
[30] E. M. McMaster *et al.*, 'Sensitivity of quantitative diffusion MRI tractography and microstructure to anisotropic spatial sampling', 2024, *arXiv*. doi: 10.48550/ARXIV.2409.18255.
[31] M. J. Mulder, M. C. Keuken, P.-L. Bazin, A. Alkemade, and B. U. Forstmann, 'Size and shape matter: The impact of voxel geometry on the identification of small nuclei', *PLOS ONE*, vol. 14, no. 4, p. e0215382, Apr. 2019, doi: 10.1371/journal.pone.0215382.
[32] S. Mieruch, S. Demirel, S. Simoncelli, R. Schlitzer, and S. Seitz, 'SalaciaML: A Deep Learning Approach for Supporting Ocean Data Quality Control', *Front. Mar. Sci.*, vol. 8, p. 611742, Apr. 2021, doi: 10.3389/fmars.2021.611742.
[33] Jason Brownlee, 'How to use Learning Curves to Diagnose Machine Learning Model Performance', Aug. 2019. Accessed: Jun. 26, 2025. [Online]. Available: https://machinelearningmastery.com/learning-curves-for-diagnosing-machine-learning-model-performance/
[34] E. M. Barroso *et al.*, 'Performance of Intraoperative Assessment of Resection Margins in Oral Cancer Surgery: A Review of Literature', *Front. Oncol.*, vol. 11, p. 628297, Mar. 2021, doi: 10.3389/fonc.2021.628297.

[35] G. Carnicelli *et al.*, 'Image-Guided Intraoperative Assessment of Surgical Margins in Oral Cavity Squamous Cell Cancer: A Diagnostic Test Accuracy Review', *Diagnostics*, vol. 13, no. 11, p. 1846, May 2023, doi: 10.3390/diagnostics13111846.
[36] C. Giannitto *et al.*, 'Frozen Section Analysis and Real-Time Magnetic Resonance Imaging of Surgical Specimen Oriented on 3D Printed Tongue Model to Assess Surgical Margins in Oral Tongue Carcinoma: Preliminary Results', *Front. Oncol.*, vol. 11, p. 735002, Dec. 2021, doi: 10.3389/fonc.2021.735002.
[37] S. C. A. Steens *et al.*, 'Evaluation of tongue squamous cell carcinoma resection margins using ex-vivo MR', *Int. J. Comput. Assist. Radiol. Surg.*, vol. 12, no. 5, pp. 821–828, May 2017, doi: 10.1007/s11548-017-1524-6.
[38] Y.-Y. Zhang *et al.*, 'The role of magnetic resonance imaging in assessing the extent of tongue squamous cell carcinoma: A prospective cohort study', *J. Stomatol. Oral Maxillofac. Surg.*, vol. 123, no. 6, pp. e822–e827, Nov. 2022, doi: 10.1016/j.jormas.2022.03.006.
[39] Y. Shimazu, T. Kudo, H. Yagishita, and T. Aoba, 'Three-dimensional visualization and quantification for the growth and invasion of oral squamous cell carcinoma', *Jpn. Dent. Sci. Rev.*, vol. 46, no. 1, pp. 17–25, Feb. 2010, doi: 10.1016/j.jdsr.2009.09.002.
[40] J. R. Lewis, 'The System Usability Scale: Past, Present, and Future', *Int. J. Human–Computer Interact.*, vol. 34, no. 7, pp. 577–590, Jul. 2018, doi: 10.1080/10447318.2018.1455307.
[41] Aaron Bangor, Philip Kortum, and James Miller, 'Determining What Individual SUS Scores Mean: Adding an Adjective Rating Scale', *JUS*, vol. 4, no. 3, pp. 114–123, May 2009.
[42] R. A. Hestehave, P. H. Gundtoft, C. L. Nielsen, O. Brink, and J. D. Rölfing, 'Poor usability of computer-assisted navigation for hip fracture surgery', *Arch. Orthop. Trauma Surg.*, vol. 144, no. 1, pp. 251–257, Oct. 2023, doi: 10.1007/s00402-023-05096-2.
[43] H. Choi, S. Kim, and W. Jang, 'User Experience Evaluation of a Spinal Surgery Robot: Workload, Usability, and Satisfaction Study', *JMIR Hum. Factors*, vol. 11, p. e54425, Apr. 2024, doi: 10.2196/54425.
[44] H. Do, P. Bourdon, D. Helbert, M. Naudin, and R. Guillevin, '7T MRI super-resolution with Generative Adversarial Network', *Electron. Imaging*, vol. 33, no. 18, pp. 106-1-106–7, Jan. 2021, doi: 10.2352/ISSN.2470-1173.2021.18.3DIA-106.
[45] G. Barisano *et al.*, 'Clinical 7 T MRI: Are we there yet? A review about magnetic resonance imaging at ultra-high field', *Br. J. Radiol.*, vol. 92, no. 1094, p. 20180492, Feb. 2019, doi: 10.1259/bjr.20180492.
[46] E. Van Reeth, I. W. K. Tham, C. H. Tan, and C. L. Poh, 'Super-resolution in magnetic resonance imaging: A review', *Concepts Magn. Reson. Part A*, vol. 40A, no. 6, pp. 306–325, Nov. 2012, doi: 10.1002/cmr.a.21249.
[47] N. J. Tustison *et al.*, 'The ANTsX ecosystem for quantitative biological and medical imaging', *Sci. Rep.*, vol. 11, no. 1, p. 9068, Apr. 2021, doi: 10.1038/s41598-021-87564-6.
[48] N. J. Tustison *et al.*, 'Large-scale evaluation of ANTs and FreeSurfer cortical thickness measurements', *NeuroImage*, vol. 99, pp. 166–179, Oct. 2014, doi: 10.1016/j.neuroimage.2014.05.044.
[49] J. Du *et al.*, 'Super-resolution reconstruction of single anisotropic 3D MR images using residual convolutional neural network', *Neurocomputing*, vol. 392, pp. 209–220, Jun. 2020, doi: 10.1016/j.neucom.2018.10.102.
[50] D. Jiang, W. Dou, L. Vosters, X. Xu, Y. Sun, and T. Tan, 'Denoising of 3D magnetic resonance images with multi-channel residual learning of convolutional neural network', *Jpn. J. Radiol.*, vol. 36, no. 9, pp. 566–574, Sep. 2018, doi: 10.1007/s11604-018-0758-8.

Declaration of AI and AI-assisted technologies in the writing process

The Artificial Intelligence (AI) technology ChatGPT (OpenAI, San Francisco, CA, USA) has been used to provide suggestions during the writing of this thesis to improve text clarity, and during the writing of the Python scripts provided in the appendices of this thesis. Contents of each output was carefully checked and was only used as inspiration.

Appendix A: Probability map merging script

```
## Guus Versteeg - TM MII M3 3D lab/Radiotherapy - UMC Utrecht

import numpy as np
import SimpleITK as sitk
import os

def load_npz_prob_map(npz_path):
    data = np.load(npz_path)
    return data['probabilities']


def resample_to_reference(input_np, input_ref_nii, output_ref_nii):
    input_itk = sitk.GetImageFromArray(input_np.astype(np.float32))
    input_itk.CopyInformation(input_ref_nii)

    resample = sitk.ResampleImageFilter()
    resample.SetReferenceImage(output_ref_nii)
    resample.SetInterpolator(sitk.sitkLinear)
    return sitk.GetArrayFromImage(resample.Execute(input_itk))


def save_segmentation(pred, reference_nii_path, output_path):
    ref_nii = sitk.ReadImage(reference_nii_path)
    pred_itk = sitk.GetImageFromArray(pred.astype(np.uint8))
    pred_itk.CopyInformation(ref_nii)
    sitk.WriteImage(pred_itk, output_path)


def combine_probability_maps(
        probmap_transversal_path, probmap_sagittal_path,
        img_transversal_path, img_sagittal_path,
        output_segmentation_path
):
    # Load probability maps
    probs_trans = load_npz_prob_map(probmap_transversal_path)
    probs_sag = load_npz_prob_map(probmap_sagittal_path)

    # Retrieve number of slices for weight determination
    slices_trans = probs_trans.shape[1]
    print(f"Number of slices in transversal: {slices_trans}")
    slices_sag = probs_sag.shape[3]
    print(f"Number of slices in sagittal: {slices_sag}")

    weight_trans = slices_trans / (slices_trans + slices_sag)
    weight_sag = 1.0 - weight_trans
    print(f"Weighting: Transversal = {weight_trans:.2f}, Sagittal =
{weight_sag:.2f}")
```

```
    # Import MR scans
    img_trans = sitk.ReadImage(img_transversal_path)
    img_sag = sitk.ReadImage(img_sagittal_path)
    spacing_trans = img_trans.GetSpacing()
    spacing_sag = img_sag.GetSpacing()

    # Confirm transversal probability map matches image shape
    trans_shape = sitk.GetArrayFromImage(img_trans).shape  # (Z, Y, X)
    assert probs_trans.shape[1:] == trans_shape, (
        f"Mismatch between transversal prob map {probs_trans.shape[1:]} and
image {trans_shape}"
    )

    # Resample sagittal prob map to match transversal space
    resampled_probs_sag = np.zeros_like(probs_trans)
    for c in range(probs_trans.shape[0]):
        resampled_probs_sag[c] = resample_to_reference(
            probs_sag[c], img_sag, img_trans
        )

    # Apply weighting
    combined_probs = weight_trans * probs_trans + weight_sag *
resampled_probs_sag

    # Create final segmentation
    seg_final = np.argmax(combined_probs, axis=0)

    # Save segmentation
    save_segmentation(seg_final, img_transversal_path,
output_segmentation_path)
    print(f"Saved combined segmentation to: {output_segmentation_path}")


# Usage
combine_probability_maps(
    probmap_transversal_path=r"/local_scratch/gverste2/Probability
maps/Data/transv_OSCC_051_IS_P031.npz",
    probmap_sagittal_path=r"/local_scratch/gverste2/Probability
maps/Data/sagitt_OSCC_051_IS_P031.npz",
    img_transversal_path=r"/local_scratch/gverste2/Probability
maps/Data/transv_scan_P031.nii.gz",
    img_sagittal_path=r"/local_scratch/gverste2/Probability
maps/Data/sagitt_scan_P031.nii.gz",
    output_segmentation_path=r"/local_scratch/gverste2/Probability
maps/Output/final_combined_segmentation_P031X.nii.gz"
)
```

Appendix B: 3D model script

```
## Guus Versteeg - TM MII M3 3D lab/Radiotherapy - UMC Utrecht


######## Import required libraries ########
import time
import json
import shutil
import pydicom
import numpy as np
from typing import Dict, Tuple, Union
from sklearn.cluster import KMeans

start_time = time.time()

######## Part names to share between Mimics and 3-matic ########
SHARED_OBJS = ["Tumour part", "Specimen part"]



######## Environment check (Mimics vs. 3-matic) ########
try:
    import trimatic
except ImportError:
    in_3matic = False
else:
    in_3matic = True



######## Full processing in Mimics ########
def run_mimics_pipeline():
    # Import required libraries
    import mimics
    import os
    import subprocess
    import json


    #### Setting up ####
    # Initialise directories
    input_dir =
r'\\ds.umcutrecht.nl\DATA\BENO\Radiotherapie\Research\Project\hoofdhals\7Tex\G
uus\Data\P032\MR'
    output_dir = r'C:\Users\gverste2\OneDrive - UMC Utrecht\Guus Versteeg - 3D
Lab Stages\M3\Mimics\PatientdXXdyeseXdX\Output'
    os.makedirs(output_dir, exist_ok=True)
```

```
    # Set predefined answers for dialog windows during import
    mimics.dialogs.set_predefined_answer("ChangeOrientation", "default")
    mimics.dialogs.set_predefined_answer("SelectPixelSize", "X")
    mimics.dialogs.set_predefined_answer("VariableSliceDistanceCorrection.Appl
y", "Yes")

    # Create function which uses DICOM tag criteria to find the transversal T2
scan from the MRI dataset
    def check_dicom_tags_MRI(dicom_file, criteria: Dict[Tuple[int, int],
Union[str, set, list, tuple]]) -> bool:
        try:
            dicom_data = pydicom.dcmread(dicom_file)
            for tag, expected_value in criteria.items():
                if tag not in dicom_data:
                    return False
                actual_value = dicom_data[tag].value

                # If expected value is list/set/tuple, check if actual value
matches any
                if isinstance(expected_value, (set, list, tuple)):
                    # Handle multi-valued DICOM elements
                    if isinstance(actual_value, (list, tuple,
pydicom.multival.MultiValue)):
                        if not any(val in expected_value for val in
actual_value):
                            return False
                    else:
                        if actual_value not in expected_value:
                            return False
                else:
                    if actual_value != expected_value:
                        return False
            return True
        except Exception as e:
            print(f"Error reading {dicom_file}: {e}")
            return False

    # Set DICOM tag criteria to select the right MRI scans
    tag_criteria_MRI = {
        (0x0008,0x0060): 'MR',
        (0x0018,0x0020): 'SE',
        (0x0018,0x0021): 'SK',
        (0x0018,0x0022): {'SP', 'OTHER'},
        (0x2001,0x100B): 'TRANSVERSAL'
    }

    # Create variable to check the number of selected MRI DICOM files
    selected_MRI = []
```

```
    # Iterate through database to find all MRI scans meeting the tag criteria
    for root, dirs, files in os.walk(input_dir):
        for file in files:
            file_path = os.path.join(root, file)
            if check_dicom_tags_MRI(file_path, tag_criteria_MRI):
                shutil.copy(file_path, output_dir)
                selected_MRI.append(file_path)
                print(f"Selected MRI file: {file_path}")
    print(f"Selected {len(selected_MRI)} DICOM files based on the specified
criteria.")

    # Import selected scans
    try:
        mimics.file.import_dicom_images(source_folder=output_dir)
        print(f"Loaded {len(selected_MRI)} DICOM files into Mimics from
{output_dir}.")
    except Exception as e:
        print(f"Error loading DICOM files into Mimics: {e}")

    # Define function for naming the image series and execute it
    def rename_image_series():
        for image_series in mimics.data.images:
            try:
                new_name = 'Unknown_series'
                dicom_tags = image_series.get_dicom_tags()

                modality_tag = (0x2001, 0x100B)
                if modality_tag in dicom_tags:
                    modality = dicom_tags[modality_tag].value

                    if modality == 'TRANSVERSAL':
                        new_name = 'Transversal T2 7T'

                    elif modality == 'SAGITTAL':
                        new_name = 'Sagittal T2 7T'

                    elif modality == 'CORONAL':
                        new_name = 'Coronal T2 7T'

                image_series.name = new_name
                print(f"Renamed series to {new_name}")

            except Exception as e:
                print(f"Error processing series {image_series.name}: {e}")

    rename_image_series()

    # Import tumour part (TEMPORARY)
```

```
    tumour_part_path = r'C:\Users\gverste2\OneDrive - UMC Utrecht\Guus
Versteeg - 3D Lab Stages\M3\Mimics\Tumour_Guus_demo_1.stl'
    tumour_part = mimics.file.import_stl(filename=tumour_part_path)
    tumour_part.name = "Tumour part"

    #### Specimen segmentation ####
    # Create segmentation function for specimen using K-means
    def specimen_segmentation(image_data: mimics.ImageData, n_clusters: int =
4):
        # Get voxel intensity data
        voxel_buffer = np.array(image_data.get_voxel_buffer(), dtype=np.uint8)
        voxel_data = voxel_buffer.flatten().reshape(-1, 1)

        # Apply K-means clustering
        kmeans = KMeans(n_clusters=n_clusters, random_state=42, n_init=10)
        labels = kmeans.fit_predict(voxel_data)

        # Sort clusters by intensity and select top two
        cluster_means = kmeans.cluster_centers_.flatten()
        sorted_clusters = np.argsort(cluster_means)
        specimen_clusters = sorted_clusters[-2:]

        # Determine threshold values for segmentation by combining selected
clusters
        specimen_voxels = voxel_data[np.isin(labels, specimen_clusters)]
        lower_threshold = np.min(specimen_voxels)
        upper_threshold = np.max(specimen_voxels)

        # Create a mask using the identified cluster
        mask = mimics.segment.create_mask()
        mimics.segment.threshold(mask=mask, threshold_min=lower_threshold,
threshold_max=upper_threshold, \
        bounding_box=None)
        mimics.segment.morphology_operations(input_mask=mask,
operation='Close', number_of_pixels=10, connectivity=8, \
        target_mask_name="Target mask", limited_to_mask=None)
        target_mask = mimics.data.masks.find("Target mask")
        final_mask = mimics.segment.keep_largest(mask=target_mask)
        mimics.segment.fill_holes(mask=final_mask)
        final_mask.color = (0.33, 0.66, 1.0)
        final_mask.name = "Specimen mask"
        mimics.data.masks.delete(mask)

        # Convert mask to a part for further processing
        part = mimics.segment.calculate_part(mask=final_mask,
quality='Optimal')
        part.color = (0.33, 0.66, 1.0)
        part.transparency = 0.20
```

```
        final_part = mimics.tools.smooth(part=part, smooth_factor=1,
iterations=500, compensate_shrinkage=True, \
        keep_originals=False)
        final_part.name = "Specimen part"

        print(f"Segmentation complete using K-means. Created part:
{final_part.name}")
        return final_part

    # Perform segmentation function
    IMAGE_DATA = mimics.data.images.find("Transversal T2 7T")
    # Determine standard deviation of voxel values for appropriate number of
clusters
    voxel_buffer = np.array(IMAGE_DATA.get_voxel_buffer(), dtype=np.uint8)
    voxel_data = voxel_buffer.flatten().reshape(-1, 1)
    print(f"STD DEV IS {np.std(voxel_data)}")
    if np.std(voxel_data) >= 38:
        NUM_CLUSTERS = 3
    else:
        NUM_CLUSTERS = 4
    print(f"Chosen number of clusters for segmentation: {NUM_CLUSTERS}")
    specimen_segmentation(IMAGE_DATA, NUM_CLUSTERS)

    ## Enable transparency view    # Only in Mimics 26.0
    mimics.view.enable_transparency()

    ## Show reference planes (optional)
    #mimics.view.show_reference_planes()


    #### Exporting to and launching 3-matic ####
    # Define script path & temporary communication file
    root_path_of_script = os.path.split(os.path.abspath(__file__))[0]
    temp_file_path = os.path.join(root_path_of_script, "my_temp.txt")
    json_path = os.path.join(root_path_of_script, "annotations.json")

    # Write STL export paths to the temp file and export each part
    with open(temp_file_path, "w") as f:
        f.write("STL export paths:\n")
        for part_name in SHARED_OBJS:
            part = mimics.data.parts.find(part_name)
            if not part:
                raise ValueError(f"Part '{part_name}' not found")
            path_of_stl = os.path.join(root_path_of_script, part.name +
".stl")
            mimics.file.export_part(object_to_convert=part,
file_name=path_of_stl, output_format='STL_ASCII') # Mimics 26.0
            #mimics.file.export_objects_as_parts(objects=[part], format="ASCII
STL", filename=path_of_stl)    # Mimics 27.0
```

```
            f.write(path_of_stl + "\n")

    # Launch 3-matic with correpsonding arguments
    trimatic = mimics.file.get_path_to_3matic()
    command = trimatic
    args = ("-run_script", __file__, temp_file_path)
    process = subprocess.Popen((command,) + args, shell=False,
stdout=subprocess.PIPE)
    process.wait()


    #### Importing 3-matic results and finalising visualisation ####
    # Read and import result STL files exported from 3-matic
    with open(temp_file_path, "r") as f:
        output_paths = [line.strip() for line in f.readlines() if
line.strip()]
    os.remove(temp_file_path)

    for path in output_paths:
        if os.path.isfile(path):
            mimics.file.import_stl(path)

    # Create irradical margin mask for 2D view
    irradical_margin = mimics.data.parts.find("Irradical margin")
    if irradical_margin:
        masks = mimics.segment.calculate_mask_from_part(part=irradical_margin,
target_mask=None)  # Mimics 26.0
        #masks =
mimics.segment.calculate_masks_from_objects(objects=[irradical_margin])
   # Mimics 27.0
        irradical_margin_mask = masks
        irradical_margin_mask.name = "Irradical margin"
        irradical_margin_mask.color = (1.0, 1.0, 0.0)
    else:
        print("No irradical margin found!")
        irradical_margin_mask = None

    with open(json_path, "r") as f:
        annotations = json.load(f)

    # Make annotations visible in Mimics
    for ann in annotations:
        name = ann["name"]
        max_point = ann["max_point"]
        closest_point = ann["closest_point"]
        distance = ann["distance_mm"]

        # Create a distance measurement (this also includes a text label)
```

```
        pt1 = mimics.analyze.create_point(point=max_point,
name=f"{name}_max_point", color=(1.0, 0.0, 0.0))
        pt2 = mimics.analyze.create_point(point=closest_point,
name=f"{name}_closest_point", color=(0.0, 0.0, 1.0))
        measurement =
mimics.measure.create_distance_measurement(point1=max_point,
point2=closest_point)
        print(f"Created distance annotation for {name}: {ann['distance_mm']}
mm")

        # Clean up unnecessary measurement parts
        mimics.data.parts.delete(pt1)
        mimics.data.parts.delete(pt2)

    # Set complete margin colour and transparency
    complete_margin = mimics.data.parts.find("Complete margin")
    complete_margin.color = (1.0, 1.0, 0)
    complete_margin.transparency = 0.8

    # Define colour mapping based on directional keywords
    colour_map = {
        "anterieur":  (0,0,1),    # Blue
        "craniaal":   (0,1,0),    # Green
        "mediaal":    (1,0,0)     # Red
    }

    # Apply colours based on name content
    for part in mimics.data.parts:
        for direction, colour in colour_map.items():
            if direction in part.name:
                part.color = colour
                break

    # Change colour for tumour and irradical margins
    if irradical_margin:
        mimics.data.parts.find("Irradical margin").color = (1.0, 1.0, 0.0)
    mimics.data.parts.find("Tumour part").color = (1.0, 0.68, 0.0)

    # Clean up temporary visualisation parts after annotation
    for part in mimics.data.parts:
        if any(keyword in part.name for keyword in ["MaxMarginPoint",
"ClosestSpecimenPoint", "MaxDistanceVector"]):
            mimics.data.parts.delete(part)

    # Name parts in Dutch for maximum clarity
    mimics.data.parts.find("Specimen part").name = "Preparaat"
    mimics.data.parts.find("Tumour part").name = "Tumor"
    if irradical_margin:
        mimics.data.parts.find("Irradical margin").name = "Irradicale marge"
```

```
    mimics.data.parts.find("Complete margin").name = "Complete marge"
    mimics.data.parts.find("mediaal_arrow_cylinder").name = "Mediaal"
    mimics.data.parts.find("mediaal_arrow_cone").name = "Mediaal"
    mimics.data.parts.find("craniaal_arrow_cylinder").name = "Craniaal"
    mimics.data.parts.find("craniaal_arrow_cone").name = "Craniaal"
    mimics.data.parts.find("anterieur_arrow_cylinder").name = "Anterieur"
    mimics.data.parts.find("anterieur_arrow_cone").name = "Anterieur"

    if not irradical_margin:
        mimics.data.parts.find("Tumor").color = (0.0, 1.0, 0.0)
        print("All margins are radical!")



######## Full processing in 3-matic ########
def run_3matic_pipeline():
    # Import required libraries
    import trimatic
    import os
    import json
    import tkinter as tk
    from tkinter import simpledialog
    from scipy.spatial import KDTree


    #### Part import from Mimics ####
    # Read STL paths passed from Mimics
    temp_file_path = sys.argv[1]
    with open(temp_file_path, "r") as f:
        lines = [line.strip() for line in f.readlines()]
    stl_paths = lines[1:]

    # Validate that exactly two parts were provided
    if len(stl_paths) != 2:
        raise ValueError("Expected exactly 2 STL paths, but got: " +
str(len(stl_paths)))

    # Import and assign parts
    part1 = trimatic.import_part_stl(stl_paths[0]) # Tumour part
    part2 = trimatic.import_part_stl(stl_paths[1]) # Specimen part


    #### Orientation arrow creation ####
    # Create GUI prompt for direction mapping
    root = tk.Tk()
    root.withdraw() # Hide main window
    dir_map = {}
    for label in ["mediaal", "craniaal", "anterieur"]:
```

```
        direction = simpledialog.askstring("Invullen richting", f"Welke kant
is {label}? (Kies uit: X, -X, Y, -Y, Z, -Z)")
        if direction not in ["X", "x", "-X", "-x", "Y", "y", "-Y", "-y", "Z",
"z", "-Z", "-z"]:
            raise ValueError(f"Incorrecte input voor richting {label}:
{direction}")
        dir_map[label] = direction.upper()

    # Dictionary to hold all arrow parts with their axis
    vector_map = {
        "X":  (1,0,0),
        "x":  (1,0,0),
        "-X": (-1,0,0),
        "-x": (-1,0,0),
        "Y":  (0,1,0),
        "y":  (0,1,0),
        "-Y": (0,-1,0),
        "-y": (0,-1,0),
        "Z":  (0,0,1),
        "z":  (0,0,1),
        "-Z": (0,0,-1),
        "-z": (0,0,-1)
    }

    # Arrow dimensions
    origin = (-50,0,-50)
    arrow_length = 100
    cone_length = 10

    # Build arrows and assign colours
    arrow_parts = []
    for label, direction in dir_map.items():
        vec = vector_map[direction]

        end = tuple(origin[i] + arrow_length * vec[i] for i in range(3))
        cone_tip = end
        mid_point = tuple((origin[i] + end[i]) / 2 for i in range(3))

        cyl = trimatic.create_cylinder_part(point_1=origin, point_2=end,
radius=5, tolerance=0.01)
        cyl.name = f"{label}_arrow_cylinder"
        arrow_parts.append(cyl)

        cone = trimatic.create_cone_part(origin=cone_tip, direction=vec,
height=cone_length, bottom_radius=10, top_radius=0,\
        tolerance=0.01, target_edge_length=0.1)
        cone.name = f"{label}_arrow_cone"
        arrow_parts.append(cone)
```

```
        # Add label in the same direction
        try:
            trimatic.quick_label(
                entity=cyl,
                text=label.capitalize(),
                point=mid_point,
                direction=vec,
                follow_surface=False,
                font='Arial',
                font_height=5,
                label_height=1,
                bold=True,
                italic=False
            )
        except RuntimeError as e:
            print(f"Failed to label {label} arrow: {e}")
    print("Orientation arrows created succesfully!")


    #### Part remesh and resection margin creation ####
    # Remesh tumour part
    remeshed = trimatic.uniform_remesh(entities=part1,
target_triangle_edge_length=0.2, preserve_sharp_edge_angle=None,\
    preserve_surface_contours=True, skip_bad_edges=False)

    # Create 5 mm resection margin
    margin = trimatic.wrap(entities=remeshed, gap_closing_distance=0.5,
smallest_detail=0.5, protect_thin_walls=False,\
    resulting_offset=5, reduce=False, preserve_sharp_features=False,
preserve_surface_structure=True)
    margin_duplicate = trimatic.duplicate(margin)
    margin_duplicate.name = "Complete margin"

    #### Subtracting specimen from resection margin and part handling ####
    # Duplicate specimen part for analysis step
    specimen_part = part2
    specimen_duplicate = trimatic.duplicate(specimen_part)


    # Boolean subtract specimen from margin to isolate tumour overlap
    subtracted_margin = trimatic.design.boolean_subtraction(margin,
specimen_part)
    subtracted_margin.name = "Subtracted margin"

    # Split into shells and label mucosa & irradical margins
    parts = trimatic.shells_to_parts(subtracted_margin)
    if not isinstance(parts, (list, tuple)):
        parts = [parts]
    mucosa_part = parts[0]
```

```
    mucosa_part.name = "Mucosa margin"
    irradical_parts = []
    for index, part in enumerate(parts[1:], start=1):
        part.name = f"Irradical margin {index}"
        irradical_parts.append(part)

    # Check if there is an irradical margin
    if not irradical_parts:
        print("\nNo irradical margins found! Only mucosa margin exists.")
    else:
        print(f"\n{len(irradical_parts)} irradical margin(s) found.")


    #### Maximum margin calculation ####
    # Assign duplicate specimen for max margin calculation
    specimen_max = specimen_duplicate
    if not specimen_max:
        raise Exception("Specimen not found.")

    # Get triangle vertices and build KD-tree for fast nearest-point lookup
    specimen_points, specimen_tris = specimen_max.get_triangles()
    specimen_vertices = [tuple(pt) for pt in specimen_points]
    kdtree = KDTree(specimen_vertices)

    # Create data lists
    visualisation_parts = []
    annotation_data = []

    # Loop through each margin part
    for part in irradical_parts:
        print(f"\nProcessing margin: {part.name}")

        # Get triangle data for margin
        margin_points, margin_tris = part.get_triangles()

        # Compute triangle centroids as sample points
        sampled_margin_centroids = []
        for tri in margin_tris:
            p1, p2, p3 = margin_points[tri[0]], margin_points[tri[1]],
margin_points[tri[2]]
            centroid = (
                (p1[0] + p2[0] + p3[0]) / 3.0,
                (p1[1] + p2[1] + p3[1]) / 3.0,
                (p1[2] + p2[2] + p3[2]) / 3.0
            )
            sampled_margin_centroids.append(centroid)

        # Find maximum distance from margin to specimen
        max_dist = -1
```

```
        max_point = None
        closest_point = None

        for margin_pt in sampled_margin_centroids:
            dist, idx = kdtree.query(margin_pt)
            specimen_pt = specimen_vertices[idx]

            if dist > max_dist:
                max_dist = dist
                max_point = margin_pt
                closest_point = specimen_pt

        print(f"Max distance: {max_dist:.2f} mm")

       # Visualise result
        if max_point and closest_point:
            # Create red spheres at the max and closest points
            max_margin_point =
trimatic.design.create_sphere_part(point_center=max_point, radius=0.3,
tolerance=0.01)
            max_margin_point.name = f"MaxMarginPoint_{part.name}"
            closest_specimen_point =
trimatic.design.create_sphere_part(point_center=closest_point, radius=0.3,
tolerance=0.01)
            closest_specimen_point.name = f"ClosestSpecimenPoint_{part.name}"

            # Create a red line (cylinder) between the points
            biggest_distance =
trimatic.design.create_cylinder_part(point_1=max_point, point_2=closest_point,
radius=0.2, tolerance=0.01)
            biggest_distance.name = f"MaxDistanceVector_{part.name}"

            # Create annotation showing the value
            anchor_point = trimatic.analyze.create_point(max_point)
            text_point = trimatic.analyze.create_point((
                (max_point[0] + closest_point[0]) / 2,
                (max_point[1] + closest_point[1]) / 2,
                (max_point[2] + closest_point[2]) / 2 + 3,
            ))

            annotation = trimatic.measure.create_annotation(
                point_anchor=anchor_point,
                point_text=text_point,
                text=f"{max_dist:.2f} mm",
                alignment=trimatic.TextAlignment.Center
            )
            trimatic.data.delete(anchor_point)
            trimatic.data.delete(text_point)
```

```
            print("Visualisation created: spheres, cylinder and annotation!")

            visualisation_parts.append(max_margin_point)
            visualisation_parts.append(closest_specimen_point)
            visualisation_parts.append(biggest_distance)
            annotation_data.append({
                "name": part.name,
                "distance_mm": round(max_dist, 2),
                "max_point": max_point,
                "closest_point": closest_point
            })

    print("\nFinished processing all margins.")


    #### Part combination and export to Mimics ####
    # Combine irradical margins into one unified part (if they exist)
    if not irradical_parts:
        parts_to_export = visualisation_parts + arrow_parts
    else:
        if len(irradical_parts) > 1:
            unified_irradical = trimatic.boolean_union(irradical_parts)
            unified_irradical.name = "Irradical margin"
        elif len(irradical_parts) == 1:
            irradical_parts[0].name = "Irradical margin"
            unified_irradical = trimatic.data.find_part("Irradical margin")
        parts_to_export = [unified_irradical] + visualisation_parts +
arrow_parts

    # Add complete margin for export
    parts_to_export.append(margin_duplicate)

    # Export analysis geometry back to Mimics
    exp = trimatic.export_stl_ascii(parts_to_export,
os.path.split(os.path.abspath(__file__))[0])

    with open(temp_file_path, "w") as f:
        for path in exp:
            f.write(path + "\n")

    json_path = os.path.join(os.path.split(os.path.abspath(__file__))[0],
"annotations.json")

    with open(json_path, "w") as json_file:
        json.dump(annotation_data, json_file, indent=4)
    print("Process complete. To continue, please close 3-matic")
```

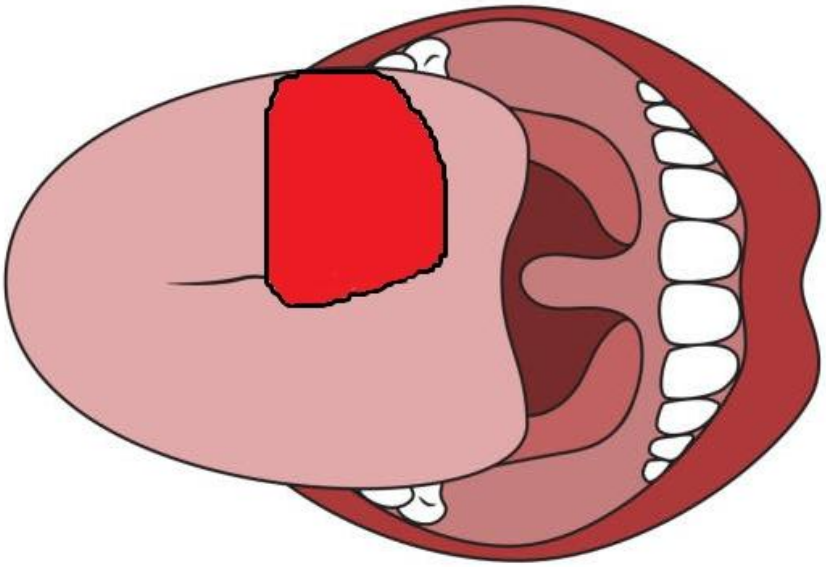
```
######## Script execution ########
if __name__ == "__main__":
    if in_3matic:
        run_3matic_pipeline()
    else:
        run_mimics_pipeline()



end_time = time.time()
total_time = end_time - start_time
print(f"\nTotal runtime: {total_time:.2f} seconds")
```

Appendix C: Clinical cases for 3D model demonstrations

Casus 1

Op de preoperatieve MRI van deze patiënt is een cilindervormige OSCC in de tong te zien van 3.5 cm lang (A-P), 2.7 cm breed (L-R) en 2.4 cm diep (C-C). Deze tumor bevond zich vlak onder het dorsale oppervlak van de tong, aan de rechter kant lateraal vanaf het mediale vlak. De gereseceerde tumor is inmiddels gescand in de 7T-MRI en het 3D-model op basis van deze scan verschijnt op het scherm voor u op OK.

In de figuur hieronder ziet u een schematische weergave van de tong, met daarop in het rood het wondbed weergegeven. Geef op dit figuur aan waar u, met behulp van het 3D-model, de naresectie toe zou passen. Noteer hierbij welke diepte u bij het snijden aan zou houden.



Naresectiediepte:	

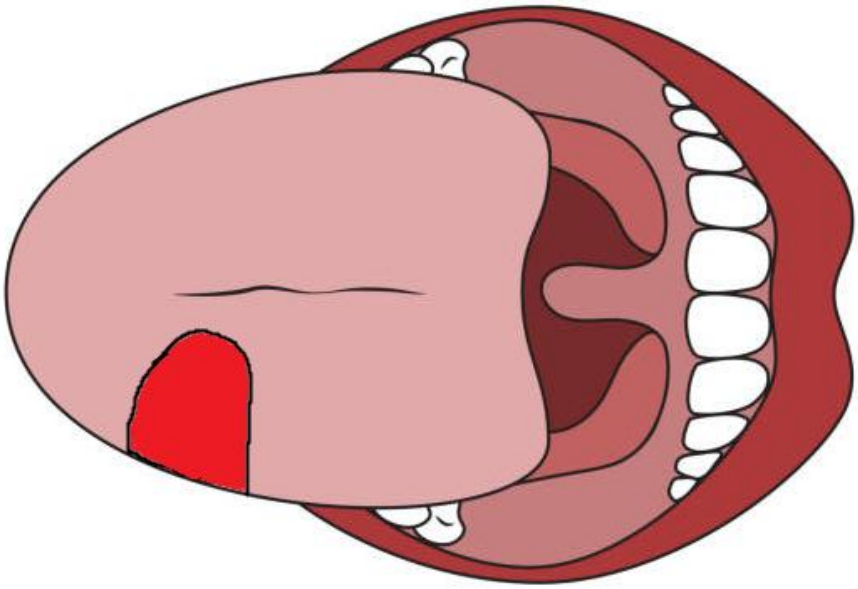
Instructies Mimics Viewer

3D-model roteren	->	Vasthouden rechter muisknop
3D-model vergroten	->	Scrollen
Model/scans op groot scherm	->	Muis op beeld naar keuze houden + spatiebalk
Doorlopen slices in scans	->	Scrollen
Contrast aanpassen scans	->	Vasthouden rechter muisknop + naar rechts/links
Contrast resetten	->	‘Reset contrast’ knop linksonder -> 
Locatie aanpassen	->	Klikken rechter muisknop
Onderdelen verbergen/tonen	->	Oogje naast onderdeel rechter menu -> 

Casus 2

De preoperatieve MRI en echo van deze patiënt laten een langwerpige tumor zien met een lengte van 1.0 cm (A-P), een breedte van 0.7 cm (L-R) en een diepte van 0.6 cm (C-C). De tumor bevond zich lateraal links op de tong (zie ook de figuur hieronder). U bent begonnen met de resectie van deze tumor, om vervolgens verder te gaan met een halsklierdissectie. In de tussentijd is het gereseceerde preparaat naar de 7T-scanner gebracht en wordt het 3D-model inclusief resectiemarges getoond op het scherm voor u.

In de figuur hieronder ziet u wederom een schematische weergave van de tong, met daarop in het rood het wondbed weergegeven. Geef op dit figuur aan waar u, met behulp van het 3D-model, de naresectie toe zou passen. Noteer hierbij welke diepte u bij het snijden aan zou houden.



Naresectiediepte:	

Instructies Mimics Viewer

3D-model roteren	->	Vasthouden rechter muisknop
3D-model vergroten	->	Scrollen
Model/scans op groot scherm	->	Muis op beeld naar keuze houden + spatiebalk
Doorlopen slices in scans	->	Scrollen
Contrast aanpassen scans	->	Vasthouden rechter muisknop + naar rechts/links
Contrast resetten	->	‘Reset contrast’ knop linksonder -> 
Locatie aanpassen	->	Klikken rechter muisknop
Onderdelen verbergen/tonen	->	Oogje naast onderdeel rechter menu -> 

Appendix D: Super-resolution script

```
## Guus Versteeg - TM MII M3 3D lab/Radiotherapy - UMC Utrecht

import ants
import os
import numpy as np
import matplotlib.pyplot as plt
import csv
from skimage.metrics import structural_similarity as ssim

# Configuration
root_dir = r"/local_scratch/gverste2//Super-resolution/Super-resolution data"
output_csv = os.path.join(root_dir, "metrics_summary.csv")

# Visualisation helper function
def show_slice_pair(img_before, img_after, title_before, title_after,
slice_index=None):
    fig, axes = plt.subplots(1, 2, figsize=(10, 5))
    arr_b = img_before.numpy()
    arr_a = img_after.numpy()
    if slice_index is None:
        slice_index = arr_b.shape[2] // 2
    axes[0].imshow(arr_b[:, :, slice_index], cmap='gray')
    axes[0].set_title(title_before)
    axes[0].axis('off')
    axes[1].imshow(arr_a[:, :, slice_index], cmap='gray')
    axes[1].set_title(title_after)
    axes[1].axis('off')
    plt.show()

## Step 1: Preprocessing
def preprocess_scan(path, outpath):
    print(f"Preprocessing: {path}")
    img = ants.image_read(path)
    img = ants.reorient_image2(img, orientation="RAI")
    img_bc = ants.n4_bias_field_correction(img)
    img_dn = ants.denoise_image(img_bc)
    ants.image_write(img_dn, outpath)
    #show_slice_pair(img, img_dn, "Original", "Bias corrected + denoised")
    return img_dn

## Step 2: Resample to match coronal ground truth spacing
def resample_to_spacing(img, target_spacing, outpath):
    print(f"Resampling image to spacing: {target_spacing}")
    img_resampled = ants.resample_image(img, target_spacing, use_voxels=False,
interp_type=3)
    ants.image_write(img_resampled, outpath)
```

```
    #show_slice_pair(img, img_resampled, "Original preprocessed", f"Resampled
to {target_spacing}")
    return img_resampled

# Loop over all subjects
with open(output_csv, 'w', newline='') as csvfile:
    writer = csv.writer(csvfile)
    writer.writerow(['Subject', 'SSIM', 'RMSE'])

    for subject_id in os.listdir(root_dir):
        subject_dir = os.path.join(root_dir, subject_id)
        raw_dir = os.path.join(subject_dir, "NIfTI")
        pre_dir = os.path.join(subject_dir, "Preprocessed")
        recon_dir = os.path.join(subject_dir, "Reconstructed")

        if not os.path.isdir(raw_dir):
            continue

        os.makedirs(pre_dir, exist_ok=True)
        os.makedirs(recon_dir, exist_ok=True)

        print(f"\n=== Processing {subject_id} ===")

        try:
            trans = preprocess_scan(os.path.join(raw_dir, "T2_t.nii.gz"),
os.path.join(pre_dir, "T2_t_pre.nii.gz"))
            sag = preprocess_scan(os.path.join(raw_dir, "T2_s.nii.gz"),
os.path.join(pre_dir,"T2_s_pre.nii.gz"))
            cor = preprocess_scan(os.path.join(raw_dir, "T2_c.nii.gz"),
os.path.join(pre_dir,"T2_c_pre.nii.gz"))

            # Define isotropic spacing based on coronal in-plane resolution
            cor_spacing = cor.spacing
            iso_spacing = (cor_spacing[0], cor_spacing[0], cor_spacing[0])
            #print(f"Target isotropic spacing from coronal scan: {iso_spacing}
(from coronal spacing {cor_spacing})")

            trans_iso = resample_to_spacing(trans, iso_spacing,
os.path.join(pre_dir, "T2_t_iso.nii.gz"))
            sag_iso = resample_to_spacing(sag, iso_spacing,
os.path.join(pre_dir, "T2_s_iso.nii.gz"))

            ## Step 3: Register sagittal scan to transversal
            print("\n--- Registering sagittal to transversal ---")
            reg = ants.registration(fixed=trans_iso, moving=sag_iso,
type_of_transform='Rigid')
            sag_reg = reg['warpedmovout']
            ants.image_write(sag_reg, os.path.join(recon_dir,
"T2_s_registered_rigid.nii.gz"))
```

```
            #show_slice_pair(sag_iso, sag_reg, "Sagittal Iso", "Registered to
Transversal")

            ## Step 4: Fuse volumes
            print("\n--- Fusing registered scans ---")
            fused_raw = (trans_iso + sag_reg) / 2
            #fused = ants.denoise_image(fused_raw)
            fused_aligned = ants.resample_image_to_target(fused_raw, cor)
            ants.image_write(fused_aligned, os.path.join(recon_dir,
"fused_iso_aligned_to_coronal.nii.gz"))

            print("--- Image Metadata ---")
            #print("trans_iso shape:", trans_iso.shape, "spacing:",
trans_iso.spacing)
            #print("sag_reg shape:  ", sag_reg.shape, "spacing:",
sag_reg.spacing)
            #print("cor shape:   ", cor.shape, "spacing:", cor.spacing)
            #show_slice_pair(fused, fused_aligned, "Fused raw", "Fused
aligned")

            ## Step 5: Compare to ground truth
            fused_np = fused_aligned.numpy()
            cor_np = cor.numpy()

            common_shape = tuple(min(fused_np.shape[i], cor_np.shape[i]) for i
in range(3))
            fused_np_cropped = fused_np[:common_shape[0], :common_shape[1],
:common_shape[2]]
            cor_np_cropped = cor_np[:common_shape[0], :common_shape[1],
:common_shape[2]]

            ssim_val = ssim(cor_np_cropped, fused_np_cropped,
data_range=cor_np_cropped.max() - cor_np_cropped.min())
            rmse_val = np.sqrt(np.mean((cor_np_cropped - fused_np_cropped) **
2))

            print("\n--- Evaluation Metrics ---")
            writer.writerow([subject_id, f"{ssim_val:.4f}",
f"{rmse_val:.4f}"])
            #print(f"SSIM: {ssim_val:.4f}")
            #print(f"RMSE: {rmse_val:.4f}")
            #show_slice_pair(cor, fused_aligned, "Ground truth", "Fused
result")

        except Exception as e:
            print(f"Failed to process {subject_id}: {e}")
```