Interactive Medical Image Segmentation - Towards Integrating Human Guidance and Deep Learning

Michael Johann Trimpl

Wadham College
University of Oxford

A thesis submitted for the degree of Doctor of Philosophy

Hilary 2023

Abstract

Medical image segmentation is an essential step in many clinical workflows involving diagnostics and patient treatment planning. Deep learning has advanced the field of medical image segmentation, particularly with respect to automating contouring. However, some anatomical structures, such as tumours, are challenging for fully automated methods. When automatic methods fail, manual contouring is required. In such cases, semi-automatic tools can support clinicians in contouring tasks. The objective of this thesis was to leverage clinicians' expert knowledge when performing segmentation tasks, allowing for interactions along the segmentation workflow and improving deep learning predictions.

In this thesis, a deep learning approach is proposed that produces a 3D segmentation of a structure of interest based on a user-provided input. If trained on a diverse set of structures, state-of-the-art performance was achieved for structures included in the training set. More importantly, the model was also able to generalize and make predictions for unseen structures that were not represented in the training set. Various avenues to guide user interaction and leverage multiple user inputs more effectively were also investigated. These further improved the segmentation performance and demonstrated the ability to accurately segment a broad range of anatomical structures.

An evaluation by clinicians demonstrated that time spent contouring was reduced when using the contextual deep learning tool as compared to conventional contouring tools. This evaluation also revealed that the majority of contouring time is observation time, which is only indirectly affected by the segmentation approach. This suggests, that user interface design and guiding the user's attention to critical areas can have a large impact on time taken on the contouring task.

Overall, this thesis proposes an interactive deep learning segmentation method, demonstrates its clinical impact, and highlights the potential synergies between clinicians and artificial intelligence.

Interactive Medical Image Segmentation - Towards Integrating Human Guidance and Deep Learning



Michael Johann Trimpl

Wadham College

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

Hilary 2023

Supervisors

Dr. Mark J. Gooding

Prof. Eleanor P. J. Stride

Prof. Katherine A. Vallis

Acknowledgements

I would like to thank my supervisors Dr. Mark J. Gooding, Prof. Eleanor P. J. Stride, and Prof. Katherine A. Vallis for giving me the opportunity to work on my DPhil project, for all their support, their innovative mindsets, and their guidance.

I want to thank all the people who have supported and helped me throughout my DPhil. For the never ending cake supply, the sharing of dog photos, the post-meeting pool games, the talks on medical imaging as well as discussions on how to make elderflower cordial, UK prime ministers, or how to best grow your own oak tree.

Thank you to all my family and friends in Germany, the UK and wherever they may live, for their understanding when I was too busy to meet them and equally when they dragged me out of my room to Port Meadow or invited me to a writing retreat in Scotland.

Finally, I would like to thank the PREDICT Marie Sklodovska Curie Action and Mirada Medical for providing my DPhil funding, enabling me to go for a secondment to Maastricht, and connect to the other groups within the Innovative Training Network. My thanks also go to the Google Cloud Research Credit programme that allowed me to conduct the evaluation with clinicians.

Abstract

Medical image segmentation is an essential step in many clinical workflows involving diagnostics and patient treatment planning. Deep learning has advanced the field of medical image segmentation, particularly with respect to automating contouring. However, some anatomical structures, such as tumours, are challenging for fully automated methods. When automatic methods fail, manual contouring is required. In such cases, semi-automatic tools can support clinicians in contouring tasks. The objective of this thesis was to leverage clinicians' expert knowledge when performing segmentation tasks, allowing for interactions along the segmentation workflow and improving deep learning predictions.

In this thesis, a deep learning approach is proposed that produces a 3D segmentation of a structure of interest based on a user-provided input. If trained on a diverse set of structures, state-of-the-art performance was achieved for structures included in the training set. More importantly, the model was also able to generalize and make predictions for unseen structures that were not represented in the training set. Various avenues to guide user interaction and leverage multiple user inputs more effectively were also investigated. These further improved the segmentation performance and demonstrated the ability to accurately segment a broad range of anatomical structures.

An evaluation by clinicians demonstrated that time spent contouring was reduced when using the contextual deep learning tool as compared to conventional contouring tools. This evaluation also revealed that the majority of contouring time is observation time, which is only indirectly affected by the segmentation approach. This suggests, that user interface design and guiding the user's attention to critical areas can have a large impact on time taken on the contouring task.

Overall, this thesis proposes an interactive deep learning segmentation method, demonstrates its clinical impact, and highlights the potential synergies between clinicians and artificial intelligence.

Contents

List of Research Outputs **xiii**

List of Figures **xv**

List of Tables **xxi**

List of Abbreviations **xxiii**

1 Introduction **1**

1.1 Statement of Problem 1

1.2 Aims and Objectives 3

1.3 Thesis Outline 4

2 Background **5**

2.1 Introduction 5

2.2 Deep Learning in Medical Image Segmentation 6

2.3 Few-shot Learning, Transfer Learning and Fine-tuning 11

2.3.1 Few-shot learning methods in medical imaging segmentation 12

2.3.2 Transfer learning 14

2.4 Interactive Methods in Medical Image Segmentation 17

2.4.1 Interactive methods in 2D 18

2.4.1.1 Traditional methods 19

2.4.1.2 Deep learning methods 20

2.4.2 Interactive methods in 3D 22

2.4.2.1 Contour propagation 23

2.4.2.2 Other deep learning methods in 3D 23

2.4.3 Active Learning 25

2.5 Guiding User Interaction using ML Feedback 27

2.6 Discussion 29

3 Contextual Deep Learning **35**
3.1 Introduction . . . 35
3.2 Using Context Information in Deep Learning . . . 36
3.2.1 Materials and Methods . . . 36
3.2.1.1 Contextual deep learning . . . 36
3.2.1.2 Network architecture . . . 38
3.2.1.3 Data . . . 39
3.2.1.4 Preprocessing . . . 39
3.2.1.5 Experimental details . . . 39
3.2.1.6 Evaluation measures . . . 42
3.2.2 Results . . . 43
3.2.2.1 Single- and multi-structure models . . . 43
3.2.2.2 Input variants . . . 44
3.2.2.3 Network architectures . . . 47
3.2.3 Discussion . . . 48
3.2.3.1 Single- and multi-structure models . . . 48
3.2.3.2 Input variants . . . 49
3.2.3.3 Network architectures . . . 50
3.2.4 Conclusion . . . 52
3.3 Generalizable Contextual Deep Learning . . . 53
3.3.1 Materials and Methods . . . 53
3.3.1.1 Contextual deep Learning for 3D segmentation . . . 53
3.3.1.2 Network architecture . . . 54
3.3.1.3 Data . . . 54
3.3.1.4 Preprocessing . . . 56
3.3.1.5 Experimental details . . . 56
3.3.1.6 Evaluation measures . . . 59
3.3.2 Results . . . 60
3.3.2.1 Training set diversity . . . 61
3.3.2.2 Iterative and direct prediction . . . 62
3.3.2.3 Propagation using image registration . . . 63
3.3.3 Discussion . . . 65
3.3.3.1 Training set diversity . . . 65
3.3.3.2 Iterative and direct prediction . . . 67
3.3.3.3 Propagation using image registration . . . 68
3.3.4 Conclusion . . . 69

4 Evaluation by Clinicians **71**
4.1 Introduction . . . 71
4.2 Preliminary Evaluation . . . 72
4.2.1 Materials and Methods . . . 73
4.2.1.1 Standard contouring workflow . . . 73
4.2.1.2 GUI design . . . 74
4.2.1.3 Data . . . 76
4.2.1.4 Experimental details . . . 76
4.2.2 Results . . . 77
4.2.2.1 Impact on contouring time . . . 77
4.2.2.2 Impact on contouring workflow . . . 77
4.2.2.3 User feedback . . . 79
4.2.3 Discussion . . . 80
4.2.3.1 Impact on contouring time . . . 80
4.2.3.2 Impact on the contouring workflow . . . 81
4.2.3.3 User feedback . . . 83
4.3 Extended Evaluation . . . 84
4.3.1 Materials and Methods . . . 84
4.3.1.1 Changes in GUI design . . . 84
4.3.1.2 Data . . . 85
4.3.1.3 Local evaluation of contouring differences . . . 86
4.3.1.4 Experimental details . . . 86
4.3.2 Results . . . 87
4.3.2.1 User interaction tracking . . . 87
4.3.2.2 Impact on contouring time . . . 88
4.3.2.3 Impact on intra- and interobserver variability . . . 90
4.3.2.4 Local evaluation of inter-observer variability . . . 91
4.3.3 Discussion . . . 92
4.3.3.1 User interaction tracking . . . 92
4.3.3.2 Impact on contouring time . . . 94
4.3.3.3 Impact on intra- and inter-observer variability . . . 96
4.3.3.4 Local evaluation of inter-observer variability . . . 97
4.4 Conclusion . . . 97

5 Task-specific Fine-tuning for Contextual Deep Learning **101**
5.1 Introduction . . . 102
5.2 Materials and Methods . . . 103
5.2.1 Fine-tuning of deep learning models . . . 103
5.2.2 Data . . . 103

5.2.3 Experimental details . . . 104
5.2.3.1 Contextual and automatic deep learning model . . 104
5.2.3.2 Single and multiple inputs . . . 105
5.2.4 Evaluation measures . . . 106
5.3 Results . . . 107
5.3.1 Interactive contouring through fine-tuning of contextual deep learning model . . . 107
5.3.1.1 Single input . . . 107
5.3.1.2 Multiple user inputs . . . 107
5.3.2 Impact of inter-observer variability on model predictions . . 109
5.3.3 Fully automatic segmentation given small dataset . . . 110
5.4 Discussion . . . 110
5.4.1 Interactive contouring through fine-tuning of contextual deep learning model . . . 110
5.4.2 Impact of inter-observer variability on model predictions . . 112
5.4.3 Fully automatic segmentation given small dataset . . . 113
5.5 Conclusion . . . 114

6 Contextual Deep Learning and Multiple User Interactions 117
6.1 Introduction . . . 117
6.2 Guiding User Interaction . . . 118
6.2.1 Materials and Methods . . . 120
6.2.1.1 Model uncertainty . . . 120
6.2.1.2 Representativeness of inputs - Mutual information 122
6.2.1.3 Auxiliary model . . . 124
6.2.1.4 Data and preprocessing . . . 125
6.2.1.5 Experimental details . . . 125
6.2.1.6 Evaluation measures . . . 126
6.2.2 Results . . . 127
6.2.2.1 Model performance based on input position . . . 127
6.2.2.2 Single slice performance as a proxy for overall performance . . . 129
6.2.2.3 Predicting the next slice to contour . . . 131
6.2.3 Discussion . . . 133
6.2.3.1 Model performance based on input position . . . 133
6.2.3.2 Single slice performance as a proxy for overall performance . . . 137
6.2.3.3 Predicting the next slice to contour . . . 137
6.2.4 Conclusion . . . 139

6.3 Optimizing Contextual Deep Learning for Multiple Inputs 141
6.3.1 Materials and Methods 141
6.3.1.1 Data and preprocessing 141
6.3.1.2 Experimental details 141
6.3.1.3 Evaluation measures 143
6.3.2 Results 143
6.3.2.1 Iterative prediction - Choosing the boundary between two user inputs 143
6.3.2.2 Direct prediction - Choosing between two user inputs 144
6.3.3 Discussion 146
6.3.3.1 Iterative prediction - Choosing the boundary between two user inputs 146
6.3.3.2 Direct prediction - Choosing between two user inputs 149
6.3.4 Conclusion 149

7 Deep Learning Models for Multiple User Interactions 151
7.1 Introduction 151
7.2 Double-context Deep Learning Model 152
7.2.1 Materials and Methods 152
7.2.1.1 Double-context model 152
7.2.1.2 Data and preprocessing 154
7.2.1.3 Experimental details 154
7.2.1.4 Evaluation measures 154
7.2.2 Results 155
7.2.2.1 Interpolation and extrapolation 155
7.2.2.2 Hybrid-context model 155
7.2.3 Discussion 157
7.2.3.1 Interpolation and extrapolation 157
7.2.3.2 Hybrid-context model 159
7.2.4 Conclusion 160
7.3 Memory Networks 161
7.3.1 Materials and Methods 161
7.3.1.1 Memory networks 161
7.3.1.2 Data and preprocessing 163
7.3.1.3 Training strategy 164
7.3.1.4 Experimental details 165
7.3.2 Results 165
7.3.2.1 Comparison of memory network architectures . . . 165
7.3.2.2 Impact of multiple user inputs 167

7.3.3 Discussion . . . 167
7.3.3.1 Comparison of memory network architectures . . . 167
7.3.3.2 Impact of multiple user inputs . . . 171
7.3.4 Conclusion . . . 172

8 Conclusion and Future Outlook **175**
8.1 Conclusions . . . 175
8.2 Future Outlook . . . 179

Data Sources and Material Transfer Agreement **183**

Statement of Originality **185**

References **187**

Appendices

A Data Overview **203**

B Learning from Context **205**
B.1 Impact of training set diversity on generalizability from context - Additional structures and evaluation measures . . . 205
B.2 Comparison of contour prediction approaches - Additional structures and evaluation measures . . . 210
B.3 Models trained on all but one structure . . . 213
B.4 Models trained on a single structure . . . 217
B.5 Models trained using different number of input channels . . . 219

C Evaluation by Clinicians **221**

D Task-specific Fine-tuning for Contextual Deep Learning **223**

List of Research Outputs

Patent inventorship filed WO2021175992A1
System and Method for Interactive Contouring of Medical Images
M. Gooding, D. Boukerroui, M. J. Trimpl
March 2020

Finalist at MedPhys Slam 2020
The sidekick health care heroes deserve
International 3 minute thesis competition, AAPM 2020
M. J. Trimpl
July 2020

Poster presented at AAPM 2020 [1]
Interactive Contouring Through Contextual Deep Learning
M. J. Trimpl, D. Boukerroui, E. Stride, K. Vallis, M. Gooding
July 2020

Research article published in Medical Physics [2]
Interactive Contouring Through Contextual Deep Learning
M. J. Trimpl, D. Boukerroui, E. Stride, K. Vallis, M. Gooding
January 2021

Poster presented at ESTRO 2021 [3]
Clinical Evaluation of an Interactive Deep Learning Contouring Tool for Target Contouring
M. J. Trimpl, P. Charlton, H. Zeng, S. Teoh, E. Stride, K. Vallis, M. Gooding
August 2021

Oral presentation at AAPM 2022 [4]
Task-Specific Fine-Tuning for Interactive Deep Learning Segmentation for Lung Fibrosis On CT Post Radiotherapy
M. J. Trimpl, P. Salome, D. Walz, J. Hoerner-Rieber, S. Regnery, E. Stride, K. Vallis, J. Debus, A. Abdollahi, M. Gooding, M. Knoll
June 2022

Review article published in Physics in Medicine and Biology [5]
Beyond Automatic Medical Image Segmentation - The spectrum between fully manual and fully automatic delineation
M. J. Trimpl, S. Primakov, D. Boukerroui, E. Stride, K. Vallis, M. Gooding
June 2022

Manuscript in preparation: **Research article to be submitted to Physics and Imaging in Radiation Oncology** [6] (expected submission by 15.04.2023)
The Impact of Interactive Deep Learning on the Clinical Contouring Workflow
M. J. Trimpl, G. Higgins, E. Stride, K. Vallis, M. Gooding

Manuscript in preparation: **Research article to be submitted to Medical Image Analysis** [7] (expected submission by 31.05.2023)
Task-Specific Fine-Tuning from Limited Data: Optimizing contextual deep learning to segment radiation induced lung fibrosis
M. J. Trimpl, P. Salome, D. Walz, J. Hoerner-Rieber, S. Regnery, E. Stride, K. Vallis, J. Debus, A. Abdollahi, M. Gooding, M. Knoll

List of Figures

2.1 Medical image segmentation on a spectrum from manual to fully automatic tools. 6
2.2 Fully automatic segmentation using deep learning. 7
2.3 Neural networks, convolutional layers, and the U-Net architecture for medical image segmentation. 8
2.4 Schematic illustration of different network building blocks. 10
2.5 Few-shot learning (n-way k-shot learning). 11
2.6 Comparison of few-shot learning and fully supervised methods when trained on small datasets. 14
2.7 Traditional machine learning vs. Transfer Learning. 15
2.8 Interactive contouring workflow. 17
2.9 Example of interactive 2D segmentation via clicks. 18
2.10 3D interactive segmentation. 22
2.11 Guiding user interaction based on model uncertainty. 28

3.1 Inputs to contextual deep learning model and model architecture. . 37
3.2 Illustration of single-structure and multi-structure model. 40
3.3 Exploration of input variants for contextual deep learning model. . 41
3.4 Schematic of dual-encoder network to incorporate contextual input. 42
3.5 Model performance and illustration of segmentation for different epochs during model training, shown for single-structure and multi-structure model. 45
3.6 Model performance when using different input variants. 46
3.7 Illustration of 3D segmentation using contextual deep learning. . . . 54
3.8 Illustration of a series of models trained on successively expanding training sets. 57
3.9 Example of contour prediction using 2D image registration. 59
3.10 Performance of models with successively increasing training set on different test datasets using iterative prediction. 61
3.11 Boxplots of the performance of different models on selected structures. 62
3.12 Comparison of direct and iterative prediction using Model ABCDEF on different structures. 63

3.13 Comparison the registration approach and Model ABCDEF for different datasets. 64
3.14 Visualization of segmentation outputs of a randomly selected head and neck test case from Dataset F. 67

4.1 Screenshot of GUI for the preliminary evaluation. 73
4.2 Standard contouring workflow as described by clinicians. 74
4.3 Experimental set-up for preliminary evaluation. 77
4.4 Relative change in active contouring time and observation time when using the deep learning tool compared to manual segmentation. . . 78
4.5 Correlation of active time and observation time for preliminary evaluation. 79
4.6 Example of user interaction times for one non-small cell lung cancer case. 80
4.7 Screenshot of GUI for the evaluation by clinicians used in the extended evaluation. 85
4.8 Histogram of time intervals between tracked mouse movement. . . . 88
4.9 Composition of active contouring time and observation time by annotator. 89
4.10 Impact on contouring time by clinician. 90
4.11 Impact on contouring time by case. 91
4.12 Correlation of active and observation time for extended evaluation. 92
4.13 Comparison of inter-observer variability for the deep learning tool and when using manual tools only, as well as the intra-observer variability by case. 93
4.14 Spatial difference to consensus contour showing median, 10, 30, 70 and 90 percentiles for all annotators per case. 94
4.15 Contouring variation between DL and manual consensus contour. . 99
4.16 Spatial variation showing 10th to 90th percentile range projected on the consensus shape for two example cases. 100

5.1 Overview of the fine-tuning experiments for contextual deep learning and fully automatic deep learning methods. 105
5.2 Input placement when using one single or multiple user inputs for fine-tuning experiments. 106
5.3 Segmentation performance when using original weights and after fine-tuning the model on the lung fibrosis dataset. 108
5.4 Visualization of segmentation results using original model weights and after fine-tuning. 108

5.5 Comparison of model performance when trained from random weights or using pre-trained weights for an automatic deep learning model and the contextual deep learning model. 108
5.6 Performance comparison between fine-tuned model and the original model with multiple inputs at every n-th slice. 109
5.7 Inter-observer variability compared to model performance when provided with one user input at the central slice of the structure. . . 110
5.8 Segmentation performance of a fully automatic deep learning model trained from random weights, and the contextual deep learning model fine-tuned. 111

6.1 Flowchart to illustrate contouring workflow. 119
6.2 Sample selection process for Active Learning. 120
6.3 Illustration of query pipeline for selecting next slice and the pipeline for query selection using a deep learning regression model based on uncertainty inputs. 122
6.4 Illustration of the pipeline for query selection using an auxiliary model. 125
6.5 Guiding user interaction. Predict which slice to contour next to minimize APL. 127
6.6 Relative APL per slice throughout the structure for user inputs at different positions in the structure. 128
6.7 Relative APL for 3D segmentation (full structure) if given a user input at different positions relative to the structure size. 129
6.8 Contextual deep learning segmentation performance with multiple user inputs. 130
6.9 Example for how a boundary slice provides a poor initial input due to error propagation 131
6.10 Illustration of correlation between segmentation performance and a suitable next user input for four example structures. 132
6.11 Predicting the next slice to contour using entropy. 134
6.12 Model performance when selecting subsequent inputs using different query methods. 135
6.13 How to optimize contextual deep learning: Choose which propagation boundary is best for iterative prediction. Choose which user input to use for direct prediction. 142
6.14 Comparison of empirical performance for a given propagation boundary and combined mutual information if that propagation boundary is chosen. 145
6.15 Illustration of optimally chosen propagation boundaries that may cluster together. 146

6.16 Selecting which of two user inputs to use for direct predictions. . . . 147

7.1 Illustration of inputs for double-context model. 153
7.2 Input placement when using one single or multiple user inputs for double-context model. 155
7.3 Performance of double-context and single-context model when using two user inputs for interpolation and extrapolation. 156
7.4 Performance of hybrid and single-context approach when using 2, 3, 4 or 5 user inputs. 157
7.5 Qualitative comparison of single-context and hybrid model when given multiple user inputs. 158
7.6 Memory network architecture. 163
7.7 Illustration of memory read operation. 163
7.8 Experiment to evaluate the effect of far-away inputs on memory networks. 166
7.9 Performance of memory network and double-context approach when using 2, 3, 4 or 5 user inputs. 168
7.10 Comparison of the segmentation throughout the structure if the memory network is given a single central input or a second input in one half of the structure. 169
7.11 Comparison of the segmentation performance of the memory network when either one or two user inputs are given to evaluate the impact of user inputs in memory that are far from a slice to be segmented. 170

B.1 Performance of models with successively increasing training set on different test datasets. 206
B.2 Boxplots of the performance (relative APL) of different models for different structures. 207
B.3 Boxplots of the performance (DSC) of different models for different structures. 208
B.4 Boxplots of the performance (95% HD) of different models for different structures. 209
B.5 Comparison of direct and iterative prediction using the model ABCDEF on different structures (relative APL). 210
B.6 Comparison of direct and iterative prediction using the model ABCDEF on different structures (DSC). 211
B.7 Comparison of direct and iterative prediction using the model ABCDEF on different structures (95% HD). 212
B.8 Multi-structure models trained on all but one structure. 214

B.9 Multi-structure models trained on all but one structure. (additional structures) 215
B.10 Multi-structure models trained on all but 1 structure are evaluated for the excluded structure. 216
B.11 Single-structure models trained on one structure. 218
B.12 Models trained with different number of input channels, evaluated for different structures using iterative prediction. 220

C.1 Contouring workflow in 3D tracked on the axial plane. 222

List of Tables

3.1 Overview of the data used for exploring contextual deep learning: training strategy, input variants, network architecture. 39
3.2 Model performance when using different network architectures. . . . 47
3.3 Overview of the data used for the in-depth investigation of contextual deep learning: training set diversity, iterative and direct prediction. 55

6.1 Model performance for different query approaches. 133
6.2 Model performance when choosing propagation boundary using different selection methods for iterative prediction. 144
6.3 Model performance when using different methods to select the user input to use for direct prediction 147

7.1 Model performance for single-context, double-context and a combination of these two approaches. 158
7.2 Model performance when using memory networks with ResNet50 backbone and U-Net-like architecture. 166

A.1 Overview of the individual datasets used in the thesis showing number of patients and structure included. 203

B.1 Reference table to link main text figures to supplementary figures which show all structures and evaluation measures. 205

D.1 Comments with regards to inter-observer variability by 3rd clinician. 224
D.2 Comments with regards to interobserver variability by 3rd clinician (Table D.1 continued). 225

List of Abbreviations

2D 2-dimensional

3D 3-dimensional

APL . . . . . Added Path Length

AU . . . . . Arbitrary Units

CNN . . . . . Convolutional Neural Network

CRF . . . . . Conditional Random Field

CT . . . . . Computed Tomography

CTAC . . . . . CT Attenuation Correction

CTN . . . . . Contour Transformer Network

CV . . . . . Cross-Validation

DL . . . . . Deep learning

FCN . . . . . Fully Convolutional Network

GAN . . . . . Generative Adversarial Network

GPU . . . . . Graphics Processing Unit

GT . . . . . Ground Truth

GTV . . . . . Gross Tumour Volume

GUI . . . . . Graphical User Interface

HD . . . . . Hausdorff Distance

HU . . . . . Hounsfield Unit

MI . . . . . Mutual Information

ML . . . . . Machine Learning

MR . . . . . Magnetic Resonance

MC . . . . . Monte-Carlo

NLP . . . . . Natural Language Processing

NSCLC . . . . Non-Small Cell Lung Cancer

OAR . . . . . Organ at Risk

PET . . . . . Positron Emission Tomography

R2U-Net . . . Residual-Recurrent U-Net

RIL . . . . . Radiology Informatics Laboratory

RT . . . . . Radiation Therapy

RU-Net . . . . Residual U-Net

SABR . . . . . Stereotactic Ablative Radiation Therapy

SBRT . . . . . Stereotactic Body Radiation Therapy

SSD . . . . . Sum of Squares of Intensities

STAPLE . . . Simultaneous Truth And Performance Level Estimation

TCIA . . . . . The Cancer Imaging Archive

WB . . . . . Whole Body

WEKA . . . . Waikato Environment for Knowledge Analysis

ZSL . . . . . Zero-Shot Learning

1 Introduction

Many clinical procedures and research rely on accurate contouring of anatomical structures in medical images. For example, image segmentation is extensively used in radiotherapy planning to identify healthy and cancerous regions [8]. Accurate segmentation of both the tumour and the healthy tissue in medical images is essential to maximize the dose of radiation delivered to the tumour while minimizing the dose delivered to healthy tissue. Segmentation is also often the first step in radiomics to extract image features from medical images that are then used, for example, to predict treatment outcomes. This introduction describes the problems of manual and automated medical image segmentation and states the need for interactive contouring approaches. This thesis aims to address this need by developing, evaluating and improving an interactive deep learning contouring tool.

1.1 Statement of Problem

Manual contouring is time consuming and subject to significant intra- and inter-observer variability. As a result, semi-automatic and fully automatic image segmentation tools have been developed to reduce contouring time and to improve the consistency of image segmentation. Machine Learning (ML)-based and especially Deep Learning (DL)-based approaches have been employed with great success

to automatically contour medical images [9], [10]. When supervised algorithm training is performed using a sufficiently large and representative dataset, ML-based approaches have been shown to generate contours that are indistinguishable from manually drawn contours [11]. However, automatic image segmentation techniques fail when the postulated assumptions inherent to the ML model are violated [12], [13]. For example, the high variability in size, location and appearance of tumours makes it challenging to build a representative training set, limiting the performance of ML methods [13]. Human input is thus still required for segmenting structures whose appearance and location exhibit large variability.

Semi-automatic methods can be employed to overcome the drawbacks of fully automatic image segmentation techniques. Such approaches can improve segmentation accuracy by integrating user inputs to guide the segmentation process. Compared to fully manual segmentation, interactive techniques have been shown to improve repeatability and consistency across observers [14]–[17]. Such models utilize user inputs as additional prior information to the system, improving contouring results [18]–[20]. For example, if a user draws a contour on one image slice in a CT or MR dataset, this contour can be propagated through the remaining slices based on the context between the contour on the delineated slice and the remaining image slices [18], [19].

Fully automatic ML techniques have successfully been employed in clinical workflows, significantly improving contouring time and consistency. However, manual edits to segmentations are often still required. Interactive semi-automatic techniques are necessary to help with this task. Furthermore, automatic solutions only exist for some structures of interest. Therefore, there is a need for versatile models that can be applied to any given structure. Interactive methods using ML or DL have not yet been extensively studied but are crucial to further help with segmentation tasks.

1.2 Aims and Objectives

The overall goal of this DPhil project is to develop an interactive contouring method using deep learning and evaluate its suitability for clinical application. Furthermore, the interplay of human input and DL will be studied. The key objectives of the present work include investigation of the following:

(1) **Identifying a structure of interest based on user input** - Whether an interactive deep learning model can be trained to understand what the user intends to contour when provided with a single user input, and then produce a 3D segmentation of that structure.

(2) **Generalizable interactive deep learning** - Whether an interactive deep learning model can generalize beyond the structures included in the training set of the model. Deep learning models typically only perform well on their training set; therefore the model training must be designed to make predictions based on the user inputs, rather than the structures used in training.

(3) **Impact on segmentation quality and contouring workflow** - The impact on contouring time, contour consistency and the contouring workflow in clinical practice. Assessing model performance compared to the labelled ground truth can be important, but ultimately contouring time and final contour quality are more important than other measures when it comes to evaluating a contouring tool for clinical suitability.

(4) **Guiding user interaction** - Anticipating beneficial future user inputs. Can model uncertainty or input representativeness be used to suggest which slice to contour next? This can be beneficial to focus a user's attention on areas in the scan or specific image slices, where the model anticipates it may have failed segmenting or where additional user inputs may improve overall segmentation performance.

(5) **Effectively utilizing multiple user inputs** - The role of human interaction with the model and how to optimally use multiple user inputs. Ultimately, in the contour review process, more than one user input will inevitably be required to correct the deep learning generated contours. This thesis will also address how multiple inputs can be leveraged effectively.

1.3 Thesis Outline

In Chapter 2, a detailed review of previous work in the field of deep learning for medical image segmentation is given, with a focus on semi-automatic and specifically interactive methods. The spectrum from manual to fully automatic contouring methods is very diverse, with each method suited for different applications. The review of the variety of contouring approaches will allow to set the contextual deep learning method that is developed and validated in this thesis into context.

Chapters 3, 4, 5, 6 and 7 describe the experimental work that addresses the key objectives outlined above. Chapter 3 introduces a deep learning segmentation method that uses a user input on one 2D slice to create a 3D segmentation of the user-indicated structure. Specifically, this chapter will address if the contextual deep learning method is able to segment structures outside the training set and, therefore, can generalize to new structures. The contextual deep learning method is then evaluated by expert volunteers in Chapter 4, focusing on the impact on contouring time, as well as contouring consistency among different users. In Chapter 5, the deep learning method is tested on a small dataset for lung fibrosis segmentation and further improved by task-specific fine-tuning. Chapter 6 investigates the impact of the human input location and the potential to anticipate the best next user input to improve segmentation performance. In Chapter 7, alternative approaches to handling multiple user inputs are investigated.

The overall conclusions of this work are presented in Chapter 8, together with an outlook to future work specific to the methods investigated in the thesis and the wider research field of automatic and interactive deep learning methods for medical image segmentation.

2 Background

Contents

2.1 Introduction 5
2.2 Deep Learning in Medical Image Segmentation 6
2.3 Few-shot Learning, Transfer Learning and Fine-tuning 11
2.3.1 Few-shot learning methods in medical imaging segmentation 12
2.3.2 Transfer learning 14
2.4 Interactive Methods in Medical Image Segmentation . 17
2.4.1 Interactive methods in 2D 18
2.4.2 Interactive methods in 3D 22
2.4.3 Active Learning 25
2.5 Guiding User Interaction using ML Feedback 27
2.6 Discussion 29

2.1 Introduction

Initially published as a review article as Ref. [5].

In this chapter, a brief overview of different segmentation methods in medical imaging is given and set within a wider historical and technical context, focusing on the impact deep learning has had on medical image segmentation. The chapter reviews different segmentation methods and their applications, ranging from manual to fully automatic segmentation and weighs their advantages and disadvantages. The spectrum from manual to fully automatic segmentation is illustrated in Figure

2.1. On this spectrum, the need for the interactive deep learning method proposed in this thesis will be demonstrated.

First, deep learning methods in medical imaging and fully automatic methods are introduced, followed by few-shot learning, which can produce segmentations given only few labelled examples. Finally, interactive methods to which the proposed contextual deep learning method belongs are introduced.

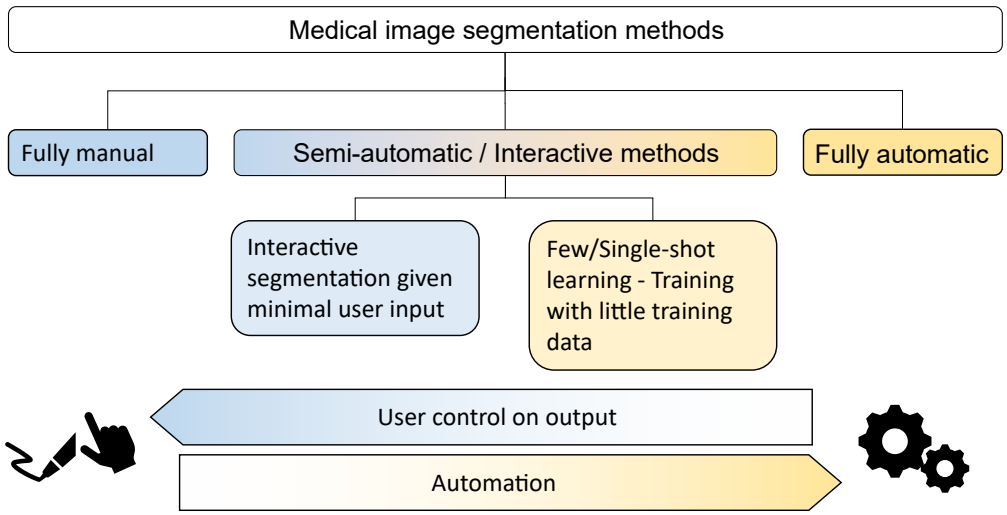


Figure 2.1: Medical image segmentation methods can be arranged on a spectrum from manual to fully automatic tools. Semi-automatic and interactive methods can be found between the extremes.

2.2 Deep Learning in Medical Image Segmentation

Deep learning has become established as the method of choice for fully automated contouring [21]–[24]. The process from creation to application of an automatic deep learning contouring tool is shown in Figure 2.2. This section will give a brief overview of the development of deep learning methods for medical imaging, and the challenges that remain. A wider introduction to deep learning applied to medical imaging can be found in [25]–[29].

Deep learning, as a sub-field of ML, has led to many breakthroughs in computer vision tasks [30], [31]. Fundamentally, it uses neural networks to extract features

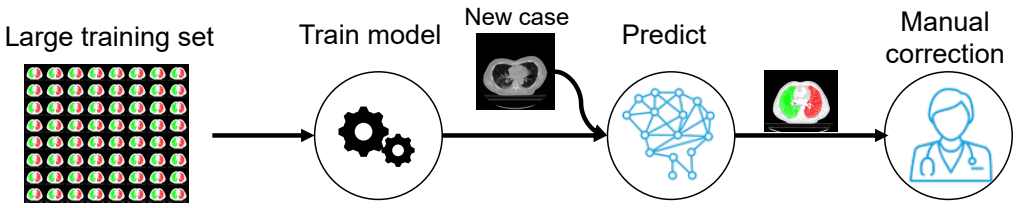


Figure 2.2: Fully automatic contouring using deep learning. A large training set is required to train the deep learning model. This model can then be used to predict the contours of new cases. The predicted segmentation can then be manually corrected by a clinician.

from the provided input data and map these to an output. A neural network consists of information processing units called neurons, that are connected to each other to form the neural network, as shown in Figure 2.3a. If many layers of neurons are stacked, it is called deep learning. The recent success of deep learning has only been made possible by the availability of increased computing power that can handle the computationally expensive task of training deep neural networks [32].

For computer vision tasks, including medical image segmentation, convolutional neural networks (CNN) are the most successful network architectures. CNNs are inspired by the hierarchical receptive field model of the visual cortex of the human brain [33]. The key features of CNNs are convolutional layers that apply a filter to the input to extract features. The appearance of an object is recognized independently of its location in an image. Thus, detection can be performed using convolution across the image. A given neuron gets a weighted input from the units in the previous layer within a small receptive field. In deep learning, multiple layers are stacked to achieve an increasingly wider receptive field. By sharing the weights of the feature mapping in different positions for each layer, the number of parameters can be decreased compared to other types of neural networks. An illustration of a convolutional operation is shown in Figure 2.3b.

One of the first implementations of CNNs for image segmentation was a fully convolutional network (FCN) [34]. This enabled the use of non-fixed input sizes, by using exclusively convolutional layers, as well as the output of a spatial segmentation map. Skip connections were used to merge upsampled feature maps from the final layers with feature maps of earlier layers. The disadvantage of this type of FCN

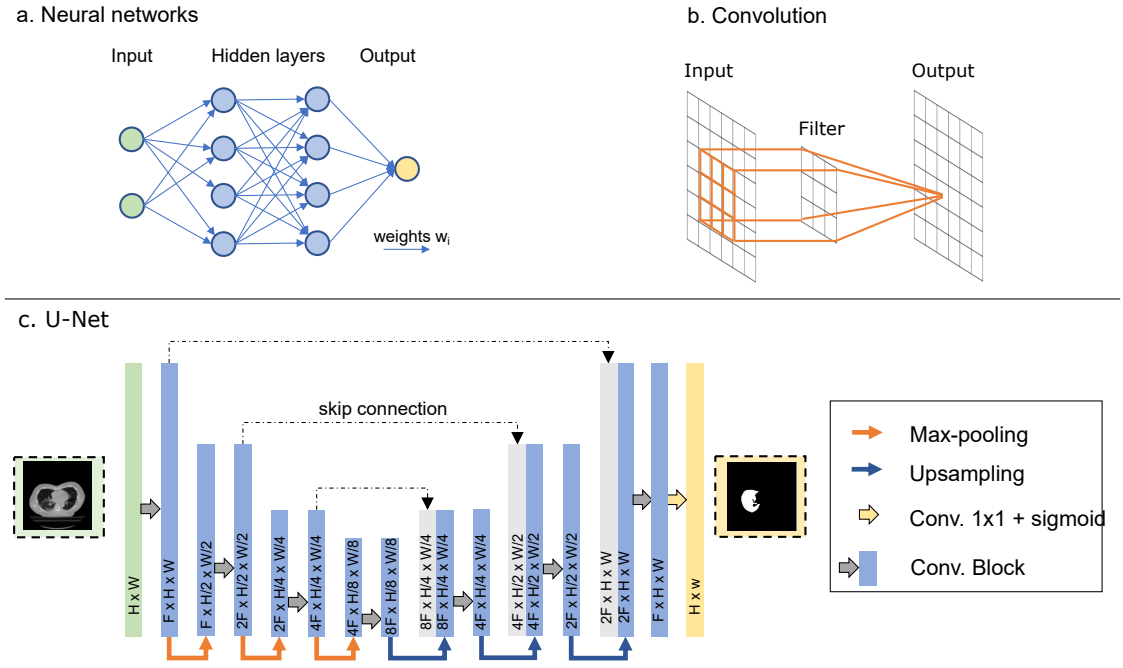


Figure 2.3: a. Schematic representation of a neural network. Neurons are connected to each other by weights W_I. Multiple layers form a neural net. b. Convolutional operation. A filter, also called a kernel, is applied to the input layer. The same filter is applied across the entire image to produce an output, called a feature map c. U-Net. The U-Net is a convolutional neural network architecture that is well suited to medical image segmentation. It consists of a contracting and expanding path with neurons connected by skip connections. The layers in the neural net vary in their number of feature maps F and spatial resolution (height H and width W).

is that the upsampling is crude and not sensitive to the details of the image. The resulting segmentation is therefore of low resolution. Additionally, in FCN, pixels are classified without fully considering spatial consistency between them.

Shortcomings of FCNs were addressed by the U-Net architecture [35], which has since been widely used in medical image segmentation. In principle, the U-Net architecture consists of a contracting path to capture the spatial context and an expanding path for localization (Figure 2.3c). Both pathways are connected by skip connections. Skip connections provide an alternative path for the gradient, which helps solve the vanishing gradient problem often faced in deep neural networks [36]. A vanishing gradient during backpropagation can prevent the neural network from updating the weights successfully during training. Furthermore, skip connections allow the U-Net to combine high-level and low-level image information and localize

these [36]. The U-Net architecture was first applied to 2D microscope images. Many variants of U-Net architecture have emerged to improve performance of specific tasks. For example, for 3D volumetric images, the 3D-U-Net and V-Net architectures [37], [38] replace 2D convolutional blocks with 3D convolutions. This enables information from neighbouring slices to be used in the generation of a contour on a particular slice, which is often critical in medical applications. Other modified neural networks include the Res-U-Net [39]–[41], which includes residual layers to mitigate the vanishing gradient problem (see Figure 2.4b), recurrent layers [41] to widen the receptive field of the network without increasing the number of network parameters (see Figure 2.4c), Attention U-Net [42], [43] which replaces skip connections with self-learned attention gates to determine which image regions matter most (see Figure 2.4d), or the hybrid densely connected U-Net [44] which fuses features from a 2D and 3D U-Net to combine intra-slice representations and inter-slice features, respectively.

Many highly specialized networks have been proposed for various medical image segmentation tasks. However, it is time-consuming to develop such tailored networks to new datasets and often the improvements to common benchmarks such as the standard U-Net are very small. The nn-U-Net was proposed to automate the configuration of preprocessing, network architecture, training and post-processing for any new segmentation task. It uses 2D and 3D U-Nets and an ensemble method to select the best segmentation from the various networks that are automatically trained [45]. Using the various architectures simultaneously and learning which to use, rather than fine-tuning the network manually, the framework has outperformed many specialized networks on over 20 datasets and medical image segmentation competition [45].

The above are a few examples deep learning architectures that have been successfully applied to fully automatic segmentation in medical imaging. These methods still face challenges that arise from working with deep learning, such as integrating interactivity into the workflow or learning on small datasets. One option to overcome the need for large quantities of data is to use alternative deep learning frameworks. For example, few-shot learning is a type of machine learning

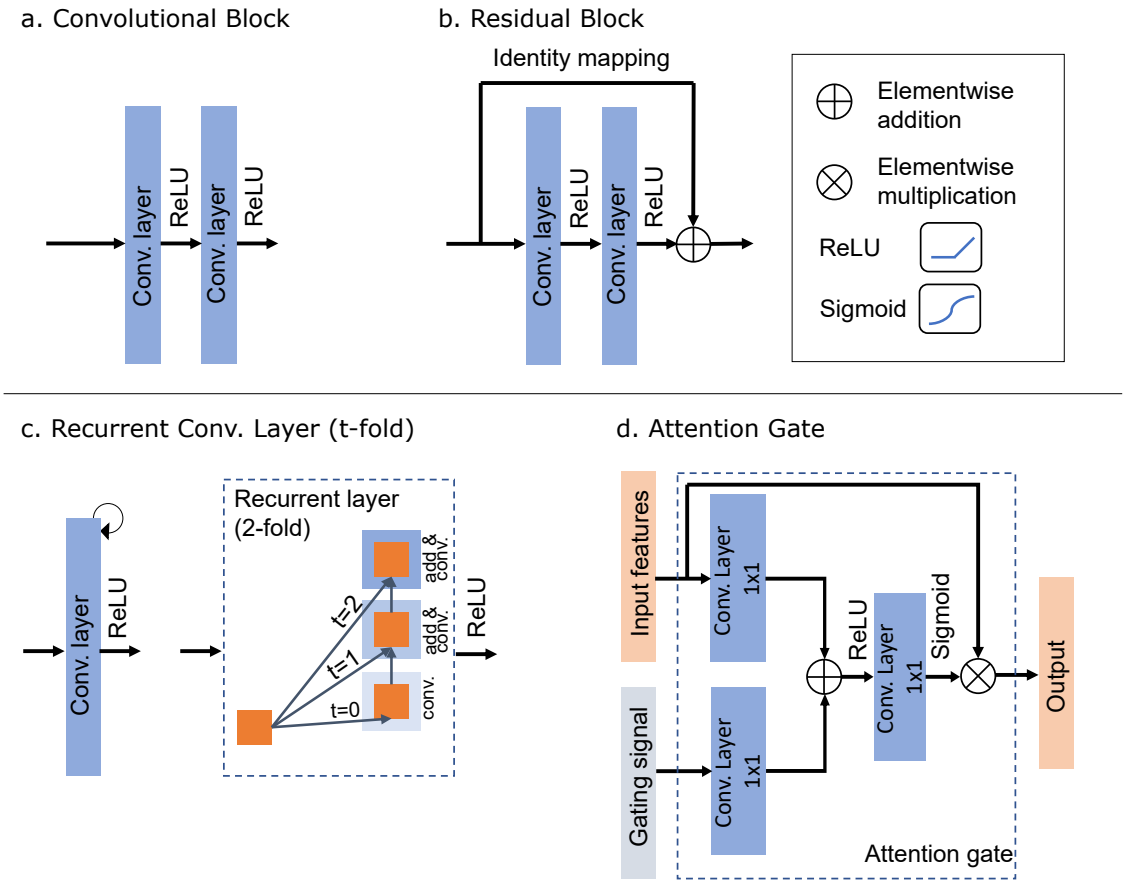


Figure 2.4: Schematic illustration of different network building blocks. (a) Standard convolutional block, consisting of two consecutive convolutions. (b) Residual layer, adding the original input to the output of the convolutional block. (c) Recurrent Block, replaces the convolutional layer with successive convolutions. To each convolution output the original input is added before convolution. Weights are shared. (d) Attention gate, replaces the skip connection and adds soft attention.

method where the training dataset contains limited information. The task of few-shot learning is to predict the class of samples for which only a few examples are available in the training set. The task of zero-shot learning is to predict the class of samples for which no examples of the class were where included in the training set. With transfer learning, existing networks can be adapted to new tasks, reducing the amount of data needed. These methods will be discussed in more detail in the next section.

2.3 Few-shot Learning, Transfer Learning and Fine-tuning

Currently, the majority of the research on automatic segmentation for medical imaging utilizes fully supervised ML models [44], [46], [47]. There is, however, an increasing number of articles proposing weakly-supervised methods, including few-, one- and zero-shot learning approaches. An in-depth review of these methods can be found in [48], [49]. Fully supervised DL methods require large and representative datasets to maximize their performance. However, in the medical imaging field, acquiring such datasets and corresponding segmentation labels can be problematic, due to restrictive personal data regulations, heterogenous clinical protocols and labelling complexity. In cases when there are only a few training images available, fully supervised methods struggle to generate correct predictions. To tackle this problem, few-shot learning methods have been proposed. Transfer learning and fine-tuning frameworks can also help when dealing with small annotated datasets of medical images [50]–[52].

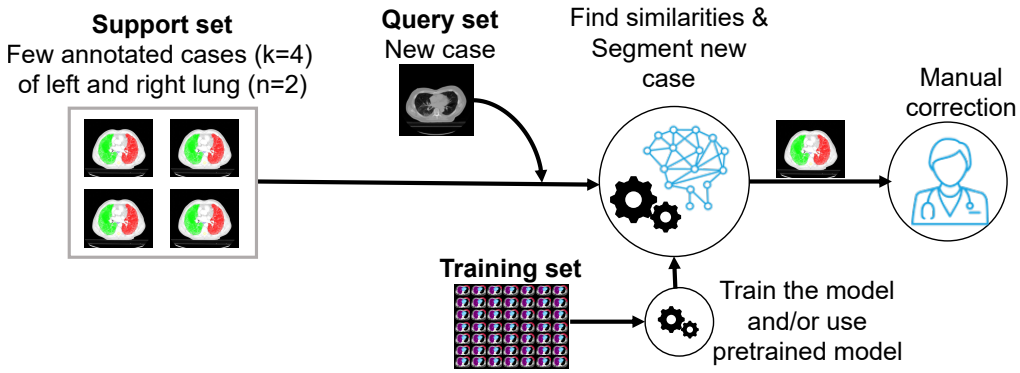


Figure 2.5: Few-shot learning (n-way k-shot learning). The structures in the training set can be different to the ones provided in the support set, but should be related, e.g. the training set should also be made up of anatomical structures. When presented with a few cases (k) of unseen class – the support set – the corresponding labels (n) are predicted for the new case – the query set. Manual correction may follow to refine segmentation. By applying transfer learning, a pre-trained model can be used as an initialization of the model.

2.3.1 Few-shot learning methods in medical imaging segmentation

The name few-shot usually refers to methods that use a small number of labelled images, called the *support set*, to assist in solving the segmentation task. Training the model still requires a large training set from which the support set is sampled to simulate the few-shot problem [53]. The goal of training is not to know what a specific structure is, as is the case in the fully automatic approach discussed in the previous section. Instead, the goal is to learn the similarity and difference between structures. The training set may have a variety of structures in it, expanding the pool of available data for training, even if only a few cases are available for the specific structure to be segmented.

The *query set* contains the images that are to be segmented. A method that uses a support set with k labelled images and n semantic classes would be called a n-way k-shot learning approach [53]. The term zero-shot learning refers to methods where the target class is not present in the support set [54]. A few annotated cases are all that are needed to achieve the corresponding segmentation on new cases, as illustrated in Figure 2.5.

Early studies on few-shot learning were mainly focused on image classification tasks [55]–[58]. However, this approach was soon adopted in natural image computer vision for segmentation tasks [59], [60] due to its reduced demand for supporting data. To perform one-shot semantic segmentation, Shaban et al. proposed using a model with two branches: conditioning and segmentation [53]. The conditioning branch is used to extract the parameters from the masked support set image, and the segmentation branch extracts the features from the query set image. The final segmentation mask is then obtained by performing pixel-level logistic regression on the query set features using parameters from the conditioning branch [53]. This approach has since been improved in several ways. For example, instead of using separate feature extractors for the support and query set, it has been proposed to use shared network weights to extract the features from both sets, reducing the number of parameters in the model [61].

In the medical imaging field, several investigators have tried to adopt few-shot learning, proposing various approaches. Mondal et al. [62] pioneered few-shot medical imaging segmentation and argued that the methods suggested by Shaban et al. have limitations, due to the heterogeneity of medical images [53], [61], [62]. Therefore, a modified 3D U-Net was proposed as a discriminator in a generative adversarial network (GAN) setting to perform few-shot infant brain MRI segmentation. This enabled the use of the unlabelled and synthetic image patches to boost the performance of the 3D U-Net [62]. As an alternative approach, the one-shot medical imaging segmentation task can be treated as a classical atlas-based segmentation problem. For this, a VoxelMorph framework was used with a GAN for additional supervision [63]. The proposed method takes the atlas and target images as input and predicts the correspondence map which can be applied to transfer the segmentation label [63]. More recently, Lu et al. presented a one-shot anatomical structure segmentation method (Contour transformer network, CTN) incorporating user intervention [64]. The framework takes only one labelled image and a set of unlabelled images as input to generate the predicted contour. To bring this solution into clinical application user corrections were incorporated to improve segmentation performance [64]. Training the CTN model requires only one labeled image and leverages additional unlabeled data through loss functions that measure the global shape and appearance consistency of contours. The CTN uses a pre-trained network trained on ImageNet [65] as the backbone of the encoding block. Optionally, additional labeled data or user annotation on mislabeled outputs of the network can be used to fine-tune the model in order to improve segmentation performance. An example segmentation of this approach is shown in Figure 2.6 and compared to a fully supervised approach. The one-shot learning approach can successfully segment the structures, whereas the fully supervised approach fails unless a sufficiently large training set is provided.

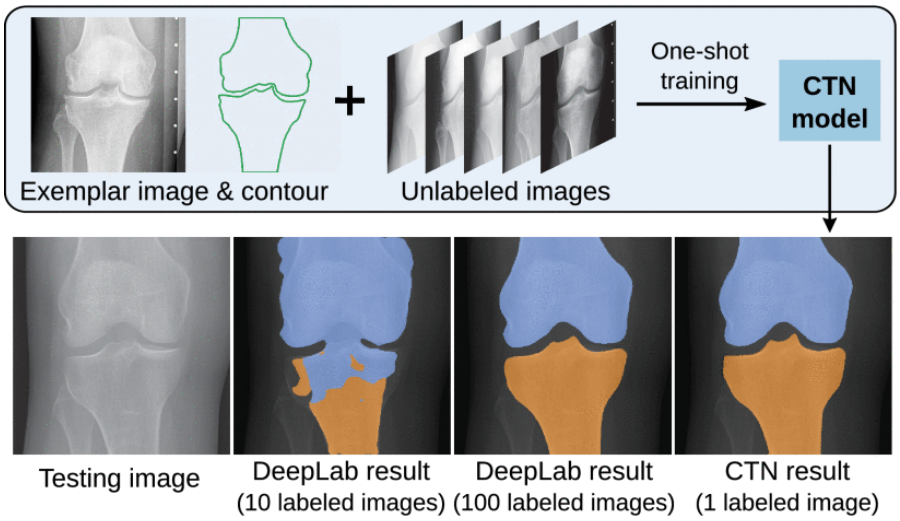


Figure 2.6: Comparison of few-shot learning and fully supervised methods. The few-shot learning model (Contour transformer network, CTN) learnt to segment the anatomical structure accurately from only one example. In contrast, fully supervised methods such as for example DeepLab [66] fail when training with insufficient labelled images. Figure obtained with permission from [64] (Copyright © 2021, IEEE).

2.3.2 Transfer learning

Deep learning has been used for medical image segmentation with a specific task in mind. Consequently, these methods are built and trained from scratch using a task-specific dataset. However, the human brain - the inspiration for modern neural networks - does not need to be retrained in this same way. By learning how to recognize specific shapes our brain can transfer this knowledge and reuse it for solving more complex tasks [67]. Transfer learning is the idea that previously learned neuron interaction coefficients, i.e. image features, can be used to solve new tasks instead of starting from scratch [50].

Transfer learning can help in tackling the limited data problem by transferring knowledge from models that were trained previously on large datasets. This can reduce the amount of data needed to train a model for the new task. Moreover, in some settings, it can drastically reduce the model training time. Compared to few-shot learning, however, a larger training set is required.

Transfer learning can be applied in a variety of settings that depend on the problem domain, model selection, and available data. Pan et al. distinguish

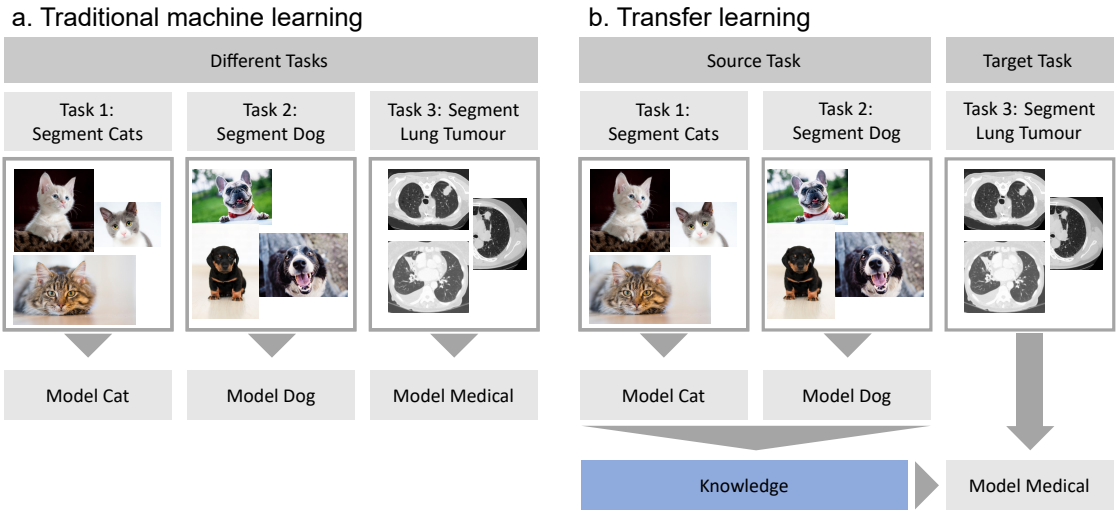


Figure 2.7: Traditional machine learning vs. Transfer Learning. (a) Traditional machine learning methods. Models are trained separately for each task. (b) Transfer learning. Datasets and the trained models are used as a knowledge source to improve the model of the target task.

at least three settings based on the availability of the labelled data: Inductive transfer learning, transductive transfer learning, and unsupervised transfer learning [50]. Inductive transfer learning characterizes the condition when both target and source tasks are different but related, and source and target domains are the same. Examples in medical imaging include the segmentation of a different organ in the same image modality and using a model trained for lung tumour classification to initialize a lung tumour segmentation model. In transductive transfer learning, tasks are the same, but the source and target domains are different, e.g. segmentation of the same organ on different image modalities. In unsupervised transfer learning, both tasks and domains are different but related, and there are no labels available in both domains during training; for example, using clustering to identify distinguishing characteristics in lung tumour patient CTs and utilize these clusters as features in a cancer classification model [50].

In most cases, deep learning models utilize an inductive transfer learning strategy, where previously trained weights could be used in two ways, i.e. without retraining as a feature extractor or being fine-tuned for a target dataset. When the pre-trained model is used as a feature extractor, the pre-trained weights are used without

being updated when trained on a new task. During training on target data, only the last layer gets trained. If additional layers of the networks are retrained on the new data, it is called fine-tuning.

Fine-tuning is frequently used together with the transfer learning approach; it can include several techniques such as selective layers retraining or pruning. After transferring the weights from a pre-trained network, users can decide to retrain some of the deep layers together with a model's final fully-connected layer [68]. These deep layers may contain very specific features that are irrelevant to a new domain or task. By retraining them on the target data these neurons will learn features specific to the target domain, therefore contributing to the performance of the model [69]. Pruning is another technique to deal with irrelevant features. Network pruning methods allow redundant neurons to be omitted during inference, resulting in reduced computing costs [70], [71].

Transfer learning with fine-tuning has been widely adopted in the medical imaging field [22], [51], [52], [72]. In many of these works, a deep learning model that was pre-trained on a publicly available imaging dataset such as ImageNet [65] is used. This model is then fine-tuned on the target medical imaging dataset [69], [73]–[75]. Several studies have reported that transfer learning-based pipelines were able to achieve performance comparable to a human reader [76], [77]. For example, the pre-trained architecture (InceptionV3) fine-tuned on a PET brain dataset to predict Alzheimer's disease [77] outperformed the user's performance on the independent test set [77]. Using a similar approach, a classification of skin lesions, equivalent to that defined by dermatologists, was achieved on a test set consisting of 1942 images [76]. To compare transfer learning with a fully supervised training model with no knowledge transfer, Van Opbroek showed that when there are few data available, transfer learning can outperform common supervised methods and reduce classification errors by up to 60% [78]. Despite the success of transfer learning in medical imaging, some studies have drawn attention to the possibility of overparameterization and suggested the use of more flexible hybrid approaches to transfer learning for medical imaging tasks [72].

2.4 Interactive Methods in Medical Image Segmentation

In contrast to few-shot learning approaches, which move along the spectrum shown in Figure 2.1 by seeking to reduce the need for training data for automated segmentation, interactive methods start with fully manual contouring and seek to reduce the need for manual intervention. Such interactive tools can be applied both when contouring manually and when editing a pre-existing segmented image. User interaction can be introduced in different ways for semi-automatic processing: 2D contours can be generated by using clicks [17], [79], scribbles [80]–[82] or bounding boxes [15], [83], [84]; a contour may also be adjusted automatically in real-time while the clinician is drawing [85]; or some 2D contours can be used to generate 3D contours of a structure [2], [18]. The key to the success of these interactive methods is to try and find a balance between human interaction and automation. Often this results in an iterative workflow between user interaction and processing, as illustrated in Figure 2.8. Interactive tools have the potential to save the clinician time and effort when contouring, compared to manual annotation alone or reviewing and editing the results of fully automated segmentation.

This section discusses interactive methods for 2D or 3D segmentation. For this, approaches that do not rely on deep learning will be introduced and contrasted with deep learning methods.

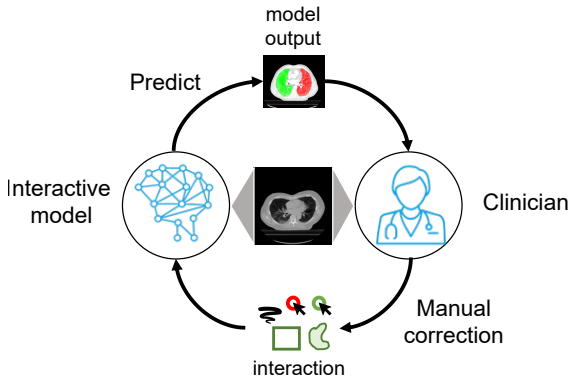


Figure 2.8: Interactive contouring workflow. The interactive model uses the first user interaction to make an initial prediction. Further user interactions are used to refine the prediction until the clinician is satisfied with the contour.

2.4.1 Interactive methods in 2D

In this section, different methods for generating contours in 2D are discussed. An example of how user interactions can result in a contour is illustrated in Figure 2.9. The figure shows a series of clicks that result in the segmentation of the user-indicated structure. To achieve such interactive segmentation, a wide range of interactive image segmentation approaches exist that do not rely on deep learning. Here, a selection of these methods and how they affect the development of deep learning methods will be discussed. A fuller description of non-deep learning methods may be found in [86], [87].

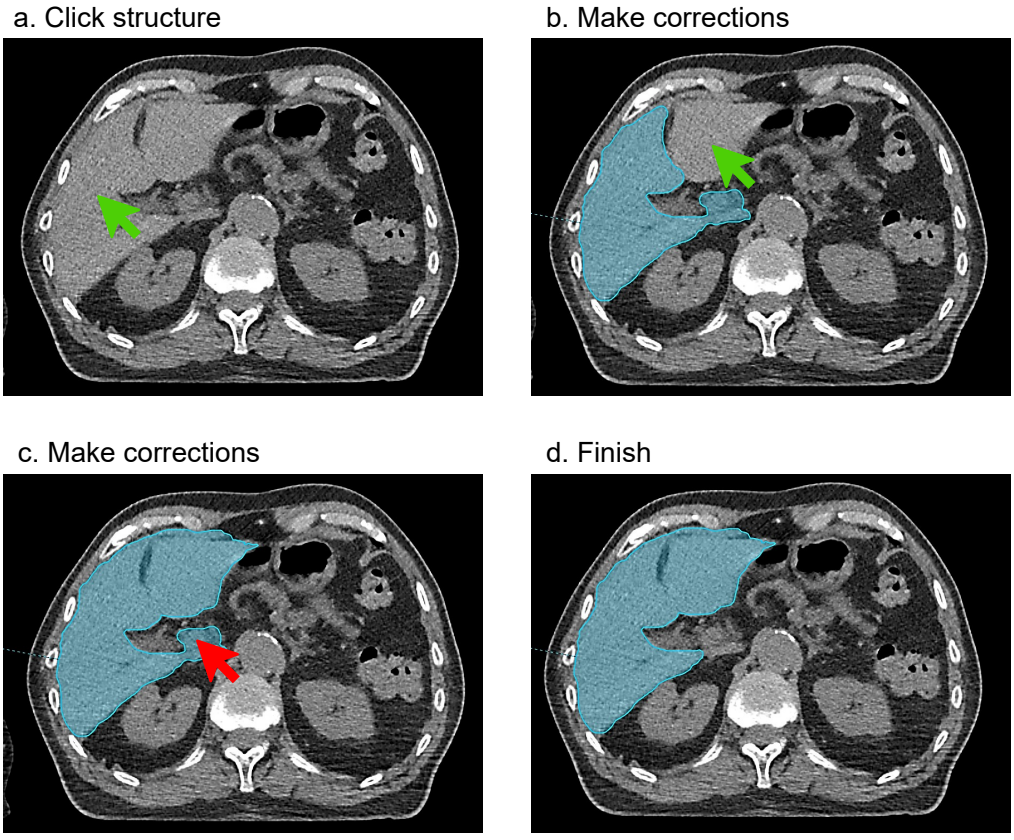


Figure 2.9: Example of interactive 2D segmentation via clicks. a. With a first click the structure to be segmented is selected. This resulting initial segmentation may need further corrections which can be edited by clicking on areas that need to be (b) included (c) or excluded. Further clicks can be used until arriving at the (d) final segmentation.

2.4.1.1 Traditional methods

Graphical energy minimisation techniques have been a popular method for image segmentation [88]–[91] particularly for interactive segmentation [46], [89], [92], [93]. For example, GrabCut sought to maximize the separation of the foreground and background classes, as initially indicated by the user, by modelling these classes using Gaussian Mixture Models. [94]. The segmentation was then produced by applying the graph cut method from the user-provided annotation. The model and segmentation are then iteratively refined based on additional user feedback. The strength of this method is the speed of the graph cut segmentation allowing rapid feedback to the user such that the segmentation could be refined easily by providing additional annotation. On the downside, GrabCut often resulted in shrunken structures. This is because GrapCut tries to minimize an energy function that consists of a data term (likelihood of foreground and background) and a spatial coherence term (length of the boundary modulated by image contrast). The latter term results in a bias towards shorter boundaries. [95] This was prevented by adding a topological prior [96]. Furthermore, the performance was improved further by including a Conditional Random Field (CRF) [96], [97] to encourage segmentation homogeneity. Instead of bounding boxes, geodesic distance transforms have been introduced to ensure spatial regularization and contrast-sensitivity. This method is called GeoS [98].

Another example of a framework for segmentation is Livewire, a contouring tool that quickly updates the contour based on user interaction. After beginning contouring, the optimal path between the starting point and the current cursor location is calculated using a lowest cost path algorithm [99]. The contour changes as the user moves the mouse. Various algorithms may be used. For example, edge detection may be implemented using a Sobel filter. In that case, the lowest cost path will be that along the edges. Livewire has been extended to the Intelligent Scissor tool [85]. On-the-fly training enables the contour to be applied to the specific type of edge being traced rather than just to the strongest edge in the image area. Furthermore, the Intelligent Scissors tool automatically freezes unchanging segments

and inserts additional seed points to increase contouring efficiency and the accuracy of the generated contours. The cost calculation can be adjusted to fit the specific needs of a given image modality. For example, ultrasound images are still difficult to process using fully automatic segmentation due to the presence of speckle and other artefacts – which are inherent to this imaging modality [100]. Livewire was adapted for ultrasound images by introducing two sets of costs: First, feature asymmetry to improve edge localization, and second, a weak shape constraint cost to improve boundary selection in the presence of missing information or artefacts [100]. As a result, the fuzzy boundaries in ultrasound images can be detected by identifying structural relevance rather than intensity gradients.

Other examples of non-deep learning based methods include level-set segmentation [101], random walks and random forest [102]. For example, SlicSeg uses online random forests to segment fetal MRI images [81]. Additionally, SlicSeg allows for contour estimation in the remaining image volume and interactivity for refinement following an adapted GraphCut method.

Open-source tool kits have been created based on these methods, for example ITK-SNAP [103] and trainable WEKA (Waikato Environment for Knowledge Analysis) [104]. Both provide an intuitive graphical user interface and are able to process various image modalities and formats. ITK-SNAP uses active contour methods to produce segmentation given an initial user input in 2D, 3D and multi-modality medical images. The trainable WEKA allows the user to train a model on a given contoured image set. The framework uses and compares any available classifier to perform image segmentation based on pixel classification. Such semi-automatic and interactive tool kits can help minimize the time and effort required for manual contouring tasks.

2.4.1.2 Deep learning methods

Deep learning has greatly improved fully automatic segmentation performance in medical imaging. Adoption of interactive contouring methods has been slower,

but various approaches to incorporate user interaction into deep learning have already shown promising results.

Rajchl et al. introduced DeepCut, which uses a CRF to update the parameters of a CNN model to favour segmentation homogeneity – as has been done with GrabCut before. DeepIGeoS combines CNNs, CRFs, and geodesic distance transforms [16], bringing deep learning to the GeoS method. This approach uses two models. The first model is used to obtain an initial segmentation. The user can then interact with the result to identify misclassification. A second network is used to refine the result given the user interaction. When applied to placenta and brain tumour segmentation of fetal MRI images, DeepIGeoS reduced user time by 66% compared with segmentation using GeoS. DeepCut performs segmentation using a bounding box input by users [83]. To improve performance, Deep Extreme Cut uses extreme points (edges) of the structure as an input to the CNN [105]. The CNN learns to match the segmentation to the extreme points of the object resulting in improved performance compared to bounding boxes. To further improve the performance of this method, other deep neural network frameworks, such as recurrent neural networks, have been proposed [106].

Most deep learning based methods that allow for user interaction build on the popular U-Net architecture. For example, Amrhen et al. proposed UI-Net for interactive segmentation [107]. In addition to allowing each image slice to be contoured, as in the standard U-Net, it also allows for user "scribbles" to be included in the model as an input. These scribbles indicate which areas should and should not be included in the segmentation. During the segmentation process, the user can continue to provide input to achieve a precise segmentation result. This system has shown superior results compared to networks without the user input channel component when applied to liver lesion segmentation. This is attributed to consistent improvement in segmentation with each user interaction. To make better use of the provided scribbles, Lin et. al. proposed ScribbleSup [80]. Instead of using the scribbles directly or applying a geodesic distance transform, a graphical

model propagates the information from the scribbles to the unmarked pixels, based on spatial constraints and appearance.

Open-source tool kits, similar to ITK-SNAP and trainable WEKA, also exist for deep learning approaches. For example, RIL-Contour [108] (Radiology Informatics Laboratory) supports medical image annotation using fully automated deep-learning, semi-automated methods, and manual methods. It also enables workflows for continual learning on newly annotated data provided by multiple annotators.

2.4.2 Interactive methods in 3D

Clinicians frequently need to contour multiple slices, for example, in 3D image sets. Manual segmentation of 3D images on a slice-by-slice basis can be very time consuming. A workflow for segmenting 3D images is illustrated in Figure 2.10. Some of the methods discussed in the section above have been extended into 3D. For example, Livewire in 2D uses two points to define a curve. In 3D, a user needs to indicate one or more closed curves (contours), and the algorithm finds the corresponding minimum cost surface [102]. Bounding box based approaches have been extended by either using individual bounding boxes on various 2D slices or by using a bounding cuboid [84].

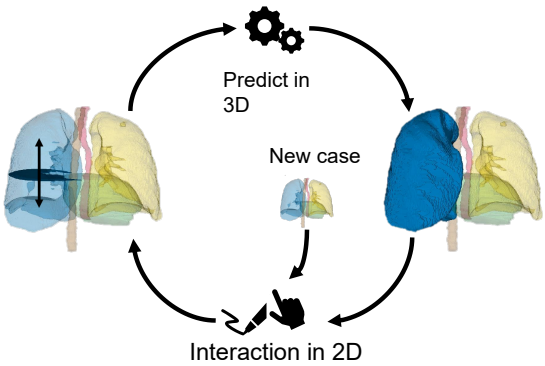


Figure 2.10: 3D interactive segmentation. 2D segmentation methods can be easily extended to 3D by, for example, propagating the contour through the image. The predicted contours can be reviewed and individual slices edited until a satisfactory segmentation is achieved.

2.4.2.1 Contour propagation

If the direct extension of a 2D to 3D contouring method is not possible, then there are a number of alternative approaches. These include automatically initializing the segmentation on the adjacent slice via a seed input and performing a segmentation on this slice. Alternatively, the contour can be iteratively propagated throughout consecutive images of a scan. China et al. introduced a volumetric segmentation approach that relies on subsequent initialization of the segmentation on adjacent image slices in ultrasound volumes. A gradient vector flow based propagation technique was used to provide an initial segmentation for the next image slice, whereas iterative random walks were used to correct the contours in subsequent steps of the algorithm [109]. Contour propagation and interpolation has also been achieved by using slice-to-slice registration of contours [110]. For this, the deformation between two image slices is calculated and that transformation is then applied to the contour. These propagation approaches can be applied in addition to any of the 2D methods mentioned before to produce 3D segmentation based on 2D input. Alternatively, if a few slices in a 3D image have been contoured, interpolation (e.g., linear interpolation) between these contours can provide an estimate for the full 3D volume of a structure.

Using CNNs, contour propagation has been applied to bladder segmentation using a single contoured image slice as the input [18]. A similar propagation approach has been used for multi-class image segmentation of the cardiac system [19]. These methods have been shown to offer better segmentation performance than fully automatic methods. However, they are only able to segment structures included in the training set and require retraining if used for other structures. This problem is often faced in supervised deep learning, as it is highly dependent on the training set provided. This problem will be addressed in Chapter 3.

2.4.2.2 Other deep learning methods in 3D

Sparse user annotations, by providing selected 2D contours, can be used to predict the remaining slices of a 3D structure. A 3D U-Net [37] has been proposed, that

uses multiple sparse annotations in a 3D volume. This method can learn from just a few contoured slices of a 3D scan and complete the segmentation by using on-the-fly elastic deformations for efficient data augmentation. The image slices with a user-defined contour are used to fine-tune a deep learning model to segment the remaining non-contoured image slices of the specific scan. This approach in effect applies transfer learning on a case-by-case basis (as discussed in Section 2.3.2) to enable an interactive contouring approach.

Again, as for 2D methods, user control on the segmentation output through image-specific fine-tuning can be increased by providing scribbles to the network. These scribbles are used to create a weighted loss function. User-provided scribbles are associated with higher confidence than the other pixels and are therefore assigned heavier weighting. Additionally, during fine-tuning, pixels with low confidence in the test image are given a lower weighting. Model based uncertainty has been associated with the softmax output, where low confidence corresponds to a value close to 0.5 [111]. This is discussed further in the next section. Using scribble-based and model uncertainty-based loss functions for fine-tuning has been shown to improve interactive segmentation performance in medical imaging for several structures, including the placenta, brain, fetal lungs and maternal kidneys [15]. As an alternative to scribbles, seed points have been used to indicate the structure to be segmented into a 2D or 3D U-Net structure [17], [112].

Memory networks have emerged as a promising approach for video segmentation tasks, enabling models to retain and utilize temporal context effectively [113]. In the context of medical image segmentation, specifically for 3D images, memory networks can have similar impact, but have not yet been extensively explored. By capturing the sequential information across multiple slices in a volumetric image, memory networks may be able to better understand the 3D structure of anatomical regions and improve segmentation accuracy [114]–[116]. The ability to store and retrieve information from previous slices would allow the model to maintain spatial consistency and coherence, which is crucial in medical applications where accurate delineation of organs and lesions is essential for diagnosis and treatment planning.

Additionally, memory networks can handle varying slice thickness and irregular spacing in medical images, making them adaptable to different scanning protocols. As a result, memory networks offer a promising avenue for advancing the state-of-the-art in medical image segmentation, particularly for 3D medical images.

The interactive 2D and 3D segmentation methods discussed in this section can synergize well with fully automatic segmentation methods. Fully automatic methods can produce a quick first estimate for a contour. Subsequently, interactive methods can be used to get the final increment in accuracy that is needed for medical image segmentation. Thus, interactive methods are an excellent tool during the contour review stage.

2.4.3 Active Learning

Active learning addresses the challenge of labelled data scarcity, common in medical imaging, from a different angle by involving human feedback via annotations in the training process [117]. The process begins with an initial, relatively small labelled dataset, and a machine learning model is trained on this limited data. Once the model is trained, it is evaluated on an unlabelled pool of data. Instead of passively waiting for human annotation, active learning algorithms select a subset of the most informative or uncertain data points for human labelling [118], [119].

The key principle behind active learning is to focus on those data points that are likely to improve the model's performance the most. This is achieved by measuring the model's uncertainty about its predictions on unlabelled data, and actively seeking instances where the model's confidence is low. These uncertain instances are deemed valuable for learning, as they represent cases where the model is less certain and can potentially benefit from additional information [120].

Various uncertainty sampling strategies are commonly used in active learning, such as selecting instances with the highest entropy (i.e., the most unpredictable cases) [118], querying for the least confident predictions, or identifying data points lying close to the decision boundary [121]. Additionally, there are diversity-based

sampling methods that aim to ensure the model receives a diverse set of samples to improve its generalization [122].

Once the most informative samples are chosen, they are sent for annotation by human experts, and the newly labelled data is added to the training set. The model is then retrained on this augmented dataset, and the active learning cycle continues iteratively, each time selecting new informative samples, refining the model, and incrementally improving its performance.

Despite its advantages, active learning also presents challenges, such as the selection of appropriate sampling strategies, managing the trade-off between uncertainty and diversity, and designing efficient selection methods to avoid unnecessary queries. Nevertheless, ongoing research in this field continues to push the boundaries of active learning, making it an increasingly indispensable tool for training accurate and efficient machine learning models.

Active learning query strategies will be used in Chapter 6 for the proposed interactive deep learning model to make decisions on which additional slices would be most advantageous to contour next.

Active Learning addresses a different aspect of user interaction with deep learning. Instead of taking into account user input to improve predictions, Active Learning tries to sample example cases to be annotated by a user to make the model learning more efficient. Therefore, Active Learning is a machine learning paradigm that aims to optimize the learning process by intelligently selecting informative samples for annotation, reducing the need for a vast amount of labeled data. Instead of passively relying on pre-labeled data, active learning algorithms actively engage in selecting the most valuable instances for human annotation, typically those that are uncertain or challenging for the current model. By iteratively selecting and labeling these samples, the model's performance improves, making it a cost-effective and time-efficient approach, especially in scenarios where acquiring labeled data is expensive or time-consuming. Active learning finds applications in various domains, including natural language processing, computer vision, and bioinformatics, among others, enabling the development of more accurate and robust machine learning models.

2.5 Guiding User Interaction using ML Feedback

Up to this point, it has been discussed how user interaction can be interpreted by a computer to assist segmentation. In this section, the possibility of a computer giving feedback to the clinician is discussed. As the performance of computer aided segmentation methods improves, the human role moves from one of an active agent to a more supervisory role. In this capacity, the user's role is to check and edit the segmentation where required. Even when using interactive segmentation methods, this checking process can still be time-consuming since the image and segmentation must still be loaded and displayed, and reviewing requires the user's attention and interaction [2]. If a model could estimate on its own where it is certain and uncertain about a prediction, or better yet where it is accurate or likely to be inaccurate, a clinician might not have to review a full image scan but could focus on the few critical regions of the scan [123].

To effectively design a feedback system, the measure of uncertainty obtained from the model must correlate with the accuracy of the model. More generally, transparency in how the model came to make a prediction could help clinicians in their decision-making process. It has been suggested that model uncertainty could be used to estimate areas to edit for interactive segmentation [124]. For example, an Uncertainty-Guided Refinement Framework has been applied to segmentation in motion-corrupted fetal MRI [123]. In this framework, the clinician is asked to edit slices with the highest segmentation uncertainty following the automatic processing step. This uncertainty estimation was shown to correlate well with mis-segmentations. By guiding the clinician's attention, contouring time was reduced by 30% when compared to using the DeepIGeoS method, while achieving similar final accuracy [123]. An example of this is shown in Figure 2.11, where model uncertainty is determined for different image slices (Figure 2.11a). Based on the model uncertainty, selected image slices or image regions may be prompted to the clinician for review (Figure 2.11b). Here, a brief overview of different approaches for determining uncertainty in deep learning prediction is given. It should be noted that

determining the uncertainty of deep learning is still a subject of ongoing research, and a more extensive discussion may be found in [125].

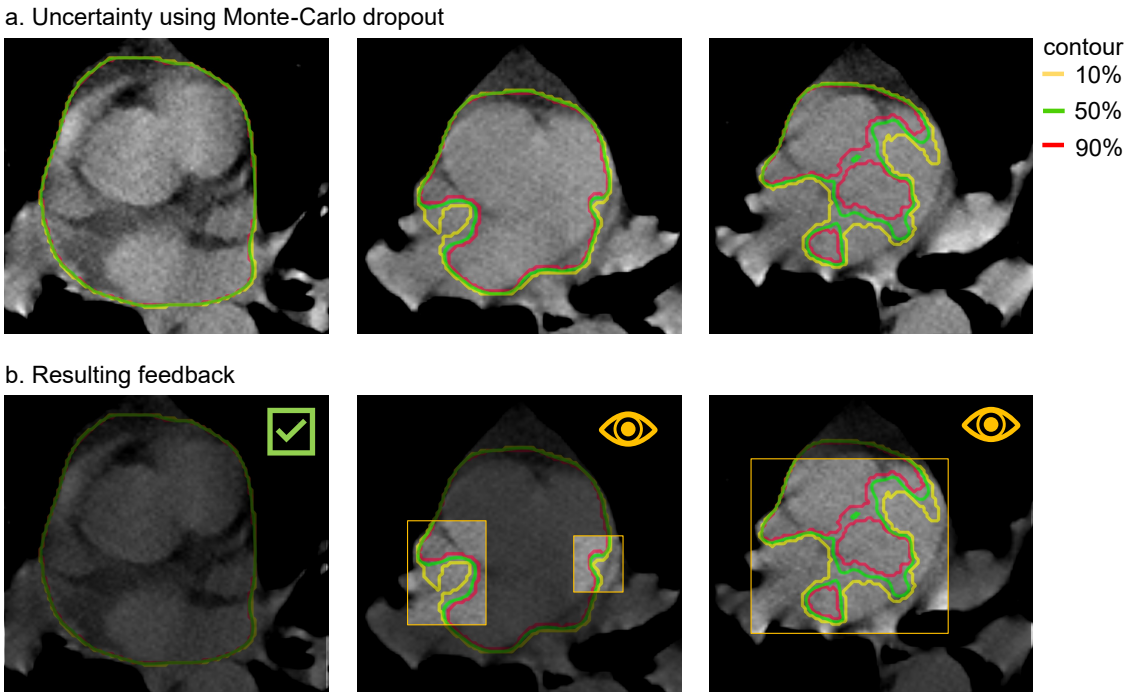


Figure 2.11: Guiding user interaction based on model uncertainty. a. Obtaining uncertainty from multiple outcomes via MC dropout or ensembling. The predictions are summarized by contours that surround the area which a certain fraction of the model includes. For example, the 10% contour (yellow) indicates that 10% of outcomes included this area. 50% corresponds to the average contour. b. Illustration of how model uncertainty could be visualized to the clinician. Image slices with little uncertainty (left image) may not need to be prompted to the clinician for review. Note, that model uncertainty is still a subject of active research. Deep learning models may also be confidently wrong in their predictions. For example, if the model has overfitted to the training data, or instead of generalizing to it overfitted to image artifacts. The approach illustrated here only illustrates the uncertainty that stems from the model itself.

The most straightforward way to estimate uncertainty is to use the final activation layer of a deep learning model as a proxy for confidence in a prediction. The activation layer (e.g. softmax) may output a value between 0 and 1 to predict if a pixel belongs to the background or foreground, respectively. If the output is close to 0.5, the model is not sure to which class to assign the pixel. An uncertainty can then be attributed to the prediction using least confidence, marginal confidence or entropy [118], [126]. However, these estimates on model uncertainty suffer from a tendency, common across deep learning methods, to be overconfident in the

accuracy of the model predictions.

To estimate the uncertainty of a model, Gal et al. introduced dropout in their neural networks as a Bayesian approximation to uncertainty [127]. Using dropout means that different connections within the neural network are randomly dropped. This results in a slightly different model each time. Originally, dropout was used during model training to reduce overfitting and increase the robustness of the model [128]. Applying dropout at test time can be used to calculate the outcome of the resulting slightly different models. The resulting distribution of outcomes can then be used as a measure of uncertainty of the model predictions [111], [121], [127].

Similarly, the uncertainty of a prediction can be estimated if an ensemble of deep learning models is used. Ensembling describes the method of using various models to make a prediction by consensus. Additionally, disagreement between the individual predictions can indicate a region of uncertainty [118], illustrated in Figure 2.11. A way to create an ensemble of models is by using convolutions in multiple groups [123]. Thereby, multiple models following the same architecture can be trained in parallel. The disagreement in the prediction can then be used to obtain an uncertainty measure, which has been shown to be superior to Monte-Carlo dropout when estimating uncertainty [123]. As above, determining uncertainty in a deep learning model remains a matter of ongoing research. However, focusing a clinician's attention on the most crucial areas and so guiding their user interaction can help make contouring more efficient and accurate [123].

2.6 Discussion

Deep learning has become the state-of-the-art approach to medical image processing, and the resulting tools are being adopted into the clinical workflow. However, the clinically implemented approaches have largely been restricted to fully automatic segmentation models. Yet, as outlined in this chapter, there are many approaches beyond fully automatic tools that could aid clinicians in performing their work.

Fully automatic methods are excellent when presented with a clearly defined task and where a large, labelled training dataset is available. In such a scenario, these

methods can potentially eliminate the need for editing by clinicians. However, while the task of segmentation might appear to be clearly defined, in reality, it is difficult to find a gold standard contour for many structures. Clinicians may disagree on what should and shouldn't be included in a contour; indeed if an individual clinician is asked to outline the same case twice, they will likely produce two different sets of contours [129]. A deep learning model trained by a single user or set of users will be biased towards the style of the clinician or centre providing the cases for the training set [130]. Conversely, if the cases in the training set are representative of a wide range of institutions and clinicians, the model may average the different opinions in a way that satisfies no one's requirements [131]. Furthermore, common deep learning methods require large amounts of data to be able to create robust models. The confidential nature of medical imaging data makes sharing and creating large datasets difficult. While there are more and more open access image datasets available [132]–[135], they are restricted often to datasets that were acquired under certain conditions for a specific study. Consequently, fully automatic methods are very rigid and not always suitable for application to a general problem.

This chapter set out to explore possible alternatives to give users greater control over deep learning model output, whilst retaining efficiency. The first step to changing to a more user-controlled model is to provide a minimal training set for a deep learning model. The question: "How little data can we get away with?" is essential here, and this in itself is a large area of ongoing research in computer vision, covered by k-shot learning, transfer learning and model fine-tuning. These methods can produce segmentations requiring only a small number of annotated images of the target class. The performance of these methods strongly depends on the problem domain and annotations in the training/support set, and the performance is typically lower than the performance of the fully supervised state of the art methods. Yet, this could give the clinician the option to – for example – define a completely new structure that a model should learn to segment. All that is needed for this is a few prior annotated cases – the support set. From these few cases, an automatic contouring pipeline can be established that can then be

applied to other cases. Overall, these methods take conventional deep learning and lower the barrier for model training by allowing for small dataset sizes. By doing so, a support set can easily be defined by a single person who can then determine what (and how) they want a certain structure to be contoured, given them a greater amount of control on the output.

Despite the promising results of the k-shot learning application in the medical imaging domain reported in several studies [49], [53]–[62], [64], challenges remain. As these methods use one or few annotated images to learn how to produce the segmentation, they are more sensitive to the variation in the medical images than fully supervised methods. To overcome this problem, some methods use unlabelled images or/and GANs to produce more training samples artificially. Another challenge for using the k-shot learning methods for medical imaging segmentation arises from the segmentation task itself. For example, the segmentation of tumours is an arduous task even for fully supervised methods, as tumours are very heterogeneous. Thus, tumour segmentation is very challenging for k-shot learning methods as they cannot capture tumour heterogeneity from a small number of samples.

Interactive methods for many applications can provide a good middle ground in terms of the trade-off between automation and user control. They are designed to interpret what the user wants to segment and suggest a segmentation based on that. The interactive methods discussed in this chapter highlight the diverse ways that this can be done, ranging from bounding boxes and scribbles to clicks and individual contours. Regardless of the specific interaction type, these methods try to interpret the user interaction and complete the segmentation that the clinician requires. In the case of energy minimization techniques, user interaction is transformed into something the computer understands by including the user interaction as a constraint to the minimization problem. This problem is generally well understood and therefore it is possible to introduce the user interaction directly into the equation. For deep learning, this becomes more complicated. The loss function, by which the model is optimized is not directly constrained by the user interaction. The

loss function compares how well the prediction matches the ground truth (GT) of the training set. The parameters in the neural network are updated based on the discrepancy. With deep learning, the loss function is effectively optimized at training time, when the user interaction on a specific patient cannot be known. It remains a challenge to express the user interaction at interaction time as a clearly defined constraint, similarly as it is done in energy optimization. To make a more effective constraint when training a model, it is necessary to better understand deep learning models themselves.

Model interpretability is an ongoing research topic. A lack of understanding of how user interaction affects a model presents a challenge to its usability. If the clinician does not know what to expect as an output from any interaction, they cannot easily control the output. This chapter has briefly addressed how a model can guide a clinician's attention to the image regions where user input is likely to be most needed. Such regions are where confidence in the prediction is low and consequently, the prediction is most likely to be incorrect.

While, to date, it remains challenging to determine uncertainty, there are other ways to improve understanding of a model and guide the clinician when contouring. Given the large number of parameters in a deep learning model, it is hard to intuitively understand what is going on throughout a neural network and therefore, it is intrinsically difficult to interpret deep learning models [136]. This understanding can be beneficial to be able to effectively use deep learning in a feedback loop, through which certain regions of a scan are brought to the attention of the clinician for review. What goes on in the model can be visualized by highlighting image-specific saliency maps - image regions that drive a model to its prediction [137].

Another way of making deep learning more accessible to human interpretation is by using "attention" in neural networks [42], [43]. Attention in a network with image input, refers to the regions in the image that receive a larger weighting due to their importance in making a prediction. They also give a more intuitive insight into what is going on inside the neural network. While convolutional filters and the resulting feature maps are very abstract, the deeper the network attention maps are

easily interpreted as they simply highlight which regions of the body are weighted highly to make a prediction. While model uncertainty and interpretability are just beginning to be understood, they can be an avenue to a more fully interactive contouring system, where the model not only interprets user inputs but can guide future user interaction where it is most needed.

3

Contextual Deep Learning

Contents

3.1 Introduction 35
3.2 Using Context Information in Deep Learning 36
3.2.1 Materials and Methods 36
3.2.2 Results 43
3.2.3 Discussion 48
3.2.4 Conclusion 52
3.3 Generalizable Contextual Deep Learning 53
3.3.1 Materials and Methods 53
3.3.2 Results 60
3.3.3 Discussion 65
3.3.4 Conclusion 69

3.1 Introduction

In Chapter 2 the wide spectrum from manual to automatic medical image segmentation was discussed. Fully automatic segmentation is already used in a clinical setting, but automatically generated contours still require human review and in many cases need to be adjusted manually. Deep learning has improved the fully automatic segmentation of this process; however, manual intervention has yet often been neglected.

In this chapter, an interactive deep learning model is introduced that helps address this issue. On the one hand side, an interactive contouring tool can be used to adjust existing contours, but also to update contours following human interaction. The first section of the chapter investigates if the contextual deep learning approach is able to distinguish between different structures of interest given a user-provided input. The second part of the chapter develops a training strategy that allows the model to generalize to arbitrary structures of interest.

3.2 Using Context Information in Deep Learning

Initially presented at AAPM 2020 [1].

This section sets out to explore the feasibility of generalizing the works of Leger et al. [18] and Zhang et al. [19]. Both of these works use contextual information to increase segmentation accuracy, but are limited to a predefined structure that is used in model training. To address this limitation, in this chapter different training set compositions, input variants and network architectures are explored. The best performing selection of these is then used in the subsequent section to develop a training strategy for generalizable contextual learning on a larger dataset.

3.2.1 Materials and Methods

3.2.1.1 Contextual deep learning

Contextual deep learning describes a type of deep learning model that makes predictions based on user-provided context information. In this section, the contextual segmentation task was defined as segmenting a selected structure on a 2D image slice (target image), given a segmentation of this structure on another image slice (contextual information) in the same Computed Tomography (CT) scan. For the contextual deep learning model to learn and make predictions based on user-provided context, a relevant set of inputs to the network needs to be chosen. First, the slice to be segmented (target slice) is essential as an input, as it holds all the specific shape and location information. Second, the inputs need to contain relevant information to indicate what is to be contoured on the target slice, referred to as

the contextual information. An example is shown in Figure 3.1a. A user-provided contour (contextual information) of a nearby slice can serve as an initial estimate of the target contour, assuming small variations between adjacent slices. Leger et al. used a prior contour as an additional input to a deep learning model to improve segmentation performance [18]. Here, it was hypothesized that the combination of the image and contour information of the prior slice can benefit segmentation performance. The contour information was passed to the model as a binary mask that does not label which structure was present, such as whether the structure was, for example, a heart, a tumor or any other structure. This information was deliberately omitted, to generalize the model to its context. This generalization has not yet been applied by prior arts [18], [19].

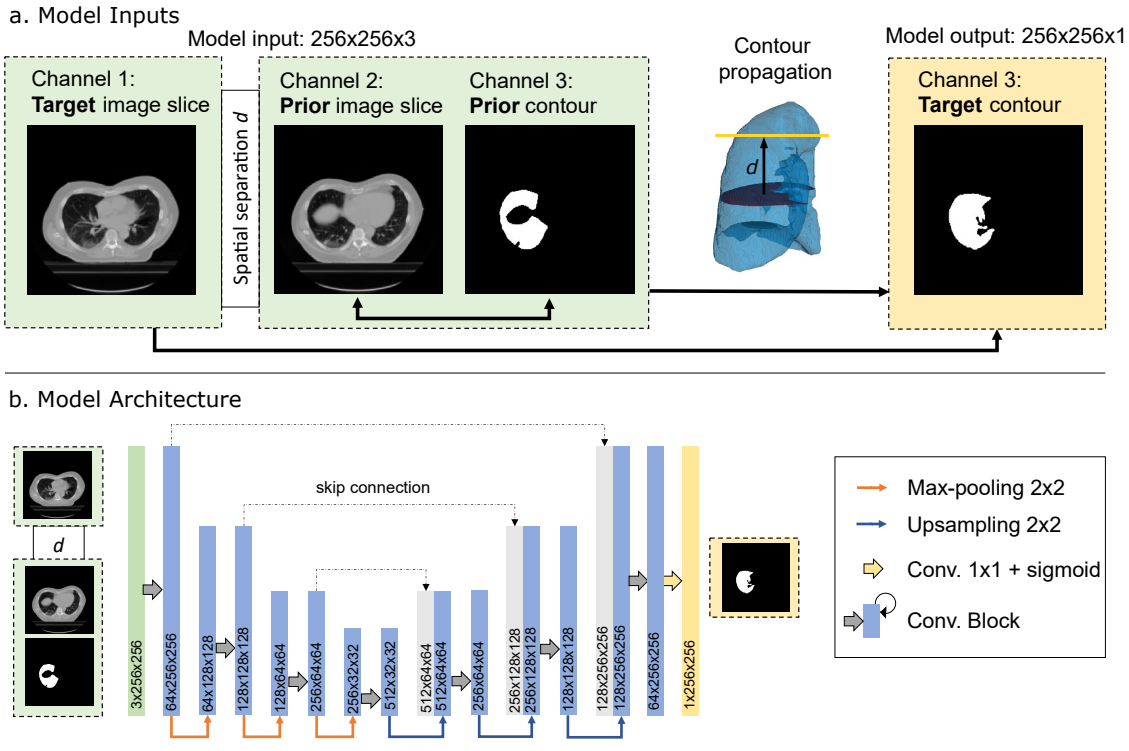


Figure 3.1: (a) Inputs to the deep learning model. Target image slice and a contextual input which consists of an image slice and its corresponding binary mask at a slice distance d from the target image slice position. The predicted target contour corresponds to the target image slice (b) U-Net architecture with illustrations of inputs and outputs. The number of features F and layer dimension (width W and height H) is indicated in the network: F × W × H.

3.2.1.2 Network architecture

Deep learning has proven to greatly improve segmentation performance in the past and major milestones have been discussed in Chapter 2. The U-Net architecture [35] has established itself as one of the most robust architectures and is often used as a benchmark for newly developed architectures. The standard U-Net is shown in Figure 3.1b. Different modifications to the standard architecture were explored in this section. These include: Residual blocks [41], [138], recurrent convolution blocks [41], [138], attention gates for skip connection [40], [42], [43] (see Figure 2.4), a U-Net with 2-separate encoding pathways [19] (see Figure 3.4), as well as combinations thereof.

Residual blocks have been shown to mitigate the vanishing gradient problem and improve training stability and convergence [41], [138]. Recurrent convolutional blocks increase the receptive field of the network, without increasing the number of network parameters [139]. To achieve a larger receptive field, a larger convolutional kernel or deeper network can be chosen but this increases the number of network parameters. As small training sets are common in medical imaging, overfitting due to an over-parameterized networks is even a greater risk. Therefore, recurrent networks can provide an alternative avenue to achieve a wider receptive field. Attention gates are incorporated to connect the features from the contracting path with the expanding path. The soft self-attention from attention gates highlights salient image regions implicitly [42], [43] thereby increasing segmentation performance.

For all architectures investigated, a sigmoid activation was applied in the last layer to obtain the probability map of the object segmentation. The number of input channels varies with the chosen number of inputs as contextual information. The networks were trained using pixelwise binary cross entropy loss [140]. The Adam optimizer [141] was used with an initial learning rate of 1×10^{-5} and a batch size of 4. Data augmentation was applied on-the-fly during training using random affine matrix transformations consisting of translation ($\pm 10\,\mathrm{mm}$), rotation (± 10 degrees) and scaling ($\pm 5\%$).

Each model presented in this study was trained from randomly initialized weights. Therefore, model performance may vary with initialization. To mitigate the variability and achieve the best possible performance for each model, 8 models were initialized for training. After 5 epochs, the worst 50% of the models were discarded. This was repeated each time the number of epochs doubles until 1 model remains. Then the remaining model was trained for up to 20 more epochs using early stopping.

3.2.1.3 Data

This study included CT images and delineations of 5 different structures of interest from the AAPM 2017 Thoracic dataset. An overview of the structures used is shown in Table 3.1. The 3D CT volumes and corresponding contouring data were randomly split into a training, validation and test set using a 70:15:15 split, respectively.

Table 3.1: Overview of number of slices used for training, validation and testing for different anatomical structures, as well as the total number of contoured slices available in the AAPM 2017 Thoracic dataset [132].

Structure	Train	Val.	Test	Total
Heart	1565	213	318	2096
Oesophagus	3830	615	802	5247
Lung (left)	3543	704	805	5057
Lung (right)	3780	711	814	5305
Spinal cord	5492	1100	1315	7907

3.2.1.4 Preprocessing

CT pixel intensities were clamped to the range from -1000 HU (air) to 2000 HU (cortical bone) to avoid artifacts in the high and low HU range. Intensities outside this range are typically irrelevant to segmenting biological tissue. The CT image slices of 512x512 pixels were rescaled to the network input size of 256x256 pixels. The rescaling was used to decrease computation time and reduce memory consumption.

3.2.1.5 Experimental details

Single- and multi-structure models First, it was investigated whether a single contextual deep learning model can make predictions for different structures without

retraining and use context to decide what structure to predict. While contextual deep learning has been shown to improve segmentation [18], [19], it has not been determined whether contextual inputs can indicate what structures to segment without explicit labelling the different structures.

A deep learning segmentation model, using a standard U-Net architecture, was trained using two alternative approaches, a single-structure model and multi-structure model. The models were trained either including contours from a single organ or from a variety of structures. A single-structure model was trained for the heart, lungs, spinal cord and oesophagus. The multi-structure model was trained on all structures as illustrated in Figure 3.2. Contextual information was provided to the model, using a prior contoured slice as an input, in addition to the slice to be contoured. Here, slice distance between target slice and contextual input slice in training and evaluation was fixed at 24 mm. The AAPM 2017 Thoracic contouring challenge dataset was used. Each case contained five organs at risk (OAR): heart, left and right lung, oesophagus and spinal cord. The models were evaluated on all OARs, regardless of the training set used. Each model was evaluated at different epochs during model training.

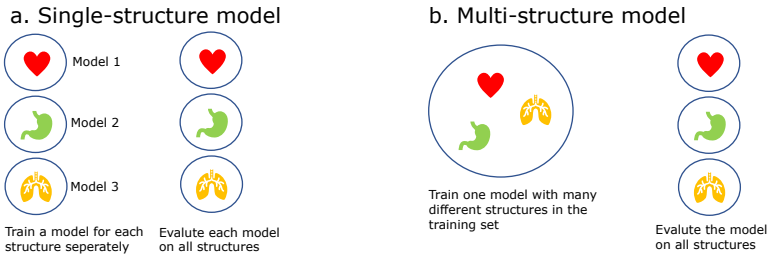


Figure 3.2: Illustration of single-structure and multi-structure model. (a) The single-structure is trained on just one structure (e.g. the heart), but will be evaluated on predicted other structures as well. (b) The multi-structure model is trained on a variety of structures.

Input variants In the literature, it has been shown that providing prior contour information increases segmentation performance [18], [19]. Here, different input variants were investigated to determine if prior image information has additional benefits to segmentation accuracy and generalizability. The question of whether

additional inputs to both sides of the structure benefit segmentation is discussed in more detail in Chapter 6. Models were trained, similar as in the previous section, for single- and multi-structure models using the AAPM 2017 Thoracic dataset. A set of deep learning models was trained using a standard U-Net architecture using different input variants. The inputs are illustrated in Figure 3.3. The target image slice always needs to be included, as this is the image slice to be segmented. All other inputs provide additional information to the model. The first variant included only a prior contour of an image slice at an inter-slice distance d from the target image slice (Figure 3.3a). The second variant additionally included the image slice corresponding to the prior contour (Figure 3.3b).

For each of the input variants, a deep learning model was trained for three slice distances d between the target and prior image slice ($d = 3\,\text{mm}$, $d = 9\,\text{mm}$, $d = 24\,\text{mm}$) and using either a single-structure or multi-structure training set, resulting in 36 different models. The models were evaluated at the slice distances used in training, respectively.

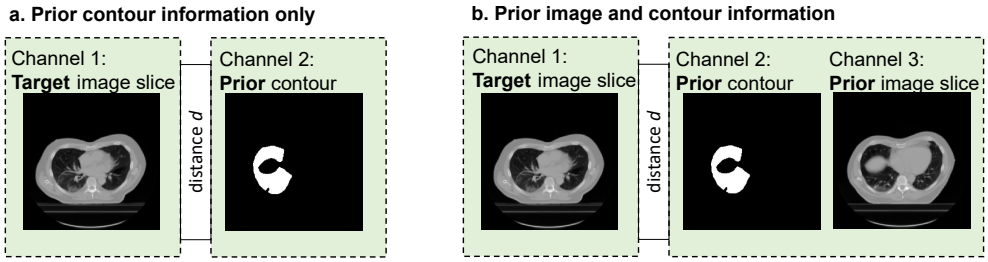


Figure 3.3: Illustration of different contextual input variants to the deep learning model: (a) GT+CT. Target image slice and ground truth (GT) of a prior slice at an inter-slice distance d from the target image slice. (b) (CT/GT)+CT. Target image slice and GT with corresponding image slice at a distance d from the target image slice.

Network architectures Different architectures based on the U-Net were explored by training a multi-structure models for three slice distances ($d = 3\,\text{mm}$, $d = 9\,\text{mm}$, $d = 24\,\text{mm}$), using the input variation (CT/GT)+CT. The models investigated were: a standard U-Net architecture, a U-Net with Residual blocks (RU-Net), a Residual U-Net with recurrent blocks (R2U-Net), and a R2U-Net with attention gates instead of standard skip connections (Attention U-Net). Additionally, a dual-encoder U-Net

that separates encoding for target slice and contextual information following the works of Zhang et al. was explored. The Dual-encoder architecture is illustrated in Figure 3.4. The models were evaluated at the respective slice distance d used in training and on all structures included in the AAPM 2017 Thoracic dataset.

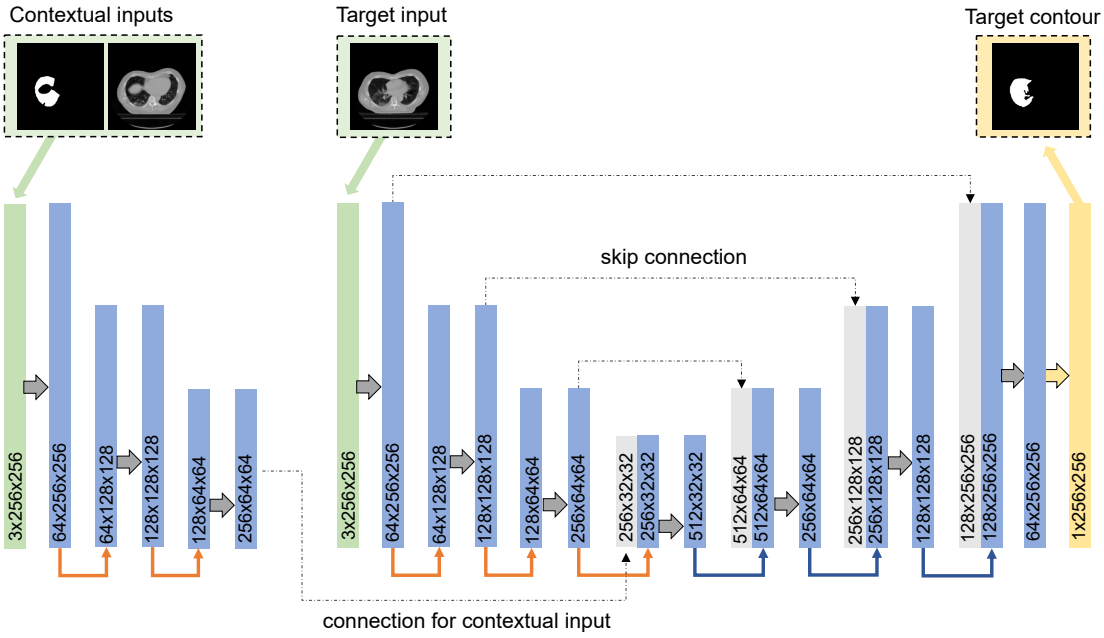


Figure 3.4: Dual-Encoder U-Net [19]. One Encoding path takes the target image as an input, the second encoding path uses the prior image slice and its corresponding segmentation as an input. The number of features F and layer dimension (width W and height H) is indicated in the individual blocks of the network: FxWxH.

3.2.1.6 Evaluation measures

The quality of the segmentation was quantified using the 3D Dice Similarity Coefficient (DSC) [142]. DSC is one of the most common evaluation measures for medical image segmentation and was chosen for the first experiment to better benchmark against existing methods. 3D DSC is strongly dependent on the volume of the structure that is being segmented. The 3D DSC of a larger structure tends to be higher than for smaller structures, due to the ratio of surface area to volume of the structure – or ratio of contour length to segmented area for 2D DSC. DSC can still provide a good reference for comparison with literature because it is still

wildly used. However, measures such as 95% HD or APL have been shown to offer a better representation of clinical impact.

The DSC measures the overlap between two areas or two volumes, A and B. The DSC is defined as

$$\mathrm{DSC}(A, B) = 2\frac{|A \cap B|}{|A| + |B|} \tag{3.1}$$

To compute the DSC, the masks that were generated were upsampled to the original 512x512 imaging resolution using bi-linear interpolation. It is important to evaluate the performance after upsampling because a clinician would review these contours at the original image resolution.

3.2.2 Results

3.2.2.1 Single- and multi-structure models

In Figure 3.5 (a, b) the segmentation performance of a single-structure model trained on hearts is compared at different epochs during training with the segmentation performance of a multi-structure model.

For the single-structure, heart-only model, the segmentation performance of the heart increased from a DSC of 0.79 ± 0.10 to 0.94 ± 0.02 during training. The other structures started with a DSC ranging from 0.20 to 0.65, but their performance decreased to a DSC of nearly 0 for the fully trained model. The multi-structure model showed an increasing DSC throughout training for all structures considered.

The DSC for heart segmentation was 0.91 for the single-organ model that was trained using a heart-only training set. However, the mean DSC for the other 4 OARs was 0.025. For a multi-organ model, the DSC on heart segmentation was 0.92 with a mean DSC of 0.76 for the other OARs. Note, that in the training process, the single-organ model did not completely fail for the first iteration of training, but then got worse for all organs but the heart. In contrast, the multi-organ model showed improvements in all organs tested throughout training. In addition, the heart segmentation performance of the multi-organ model remained similar to the performance of the single-organ model.

Figure 3.5 visualizes the model output at different epochs. At iteration 1, the predicted contours of both models appear to be shifted upward from the Ground Truth (GT), indicated by the white arrow. The model effectively outputs a copy of the input contour that is intended to provide context. However, the predicted contour is associated with an image slice that is above the input contour slice, resulting in the observed upshift of the contours. At iteration 40, the single-organ model trained on hearts, predicted a heart even when presented with a spinal cord contour as context.

The oesophagus and spinal cord model were difficult to train. Despite the 8 randomly initialized trained models, and the selection of the best performing one, these models did not converge. Upon any input image slice, these models would output an empty segmentation mask. This may be attributed to an inadequate loss function for these specific structures when trained isolated from other training examples. The structures were very small and therefore are severely imbalanced for the binary crossentropy loss function, resulting in a prediction of the background rather than any structure. Therefore, we adjusted the training parameters by increasing the class weight of these structures for the first 10 iterations (only for these two single-structure models). This gives the model a chance to find appropriate weights despite the class imbalance.

3.2.2.2 Input variants

The performance of a single-structure, lung-only model and multi-structure model trained with contextual inputs at different slice distances d is shown in Figure 3.6 for the prediction of heart and spinal cord.

The multi-structure model showed equal or better performance for all predicted structures, input variants and slice distances, as compared to the single-structure model. For prediction of the lungs, the DSC was high and similar in value for all slice distances d and input variants. With increasing slice distance d, the DSC decreased for all other structures.

The image slice and contour contextual input (CT/GT+CT) achieved higher DSC than the contour only contextual input (GT+CT). Particularly for the single-

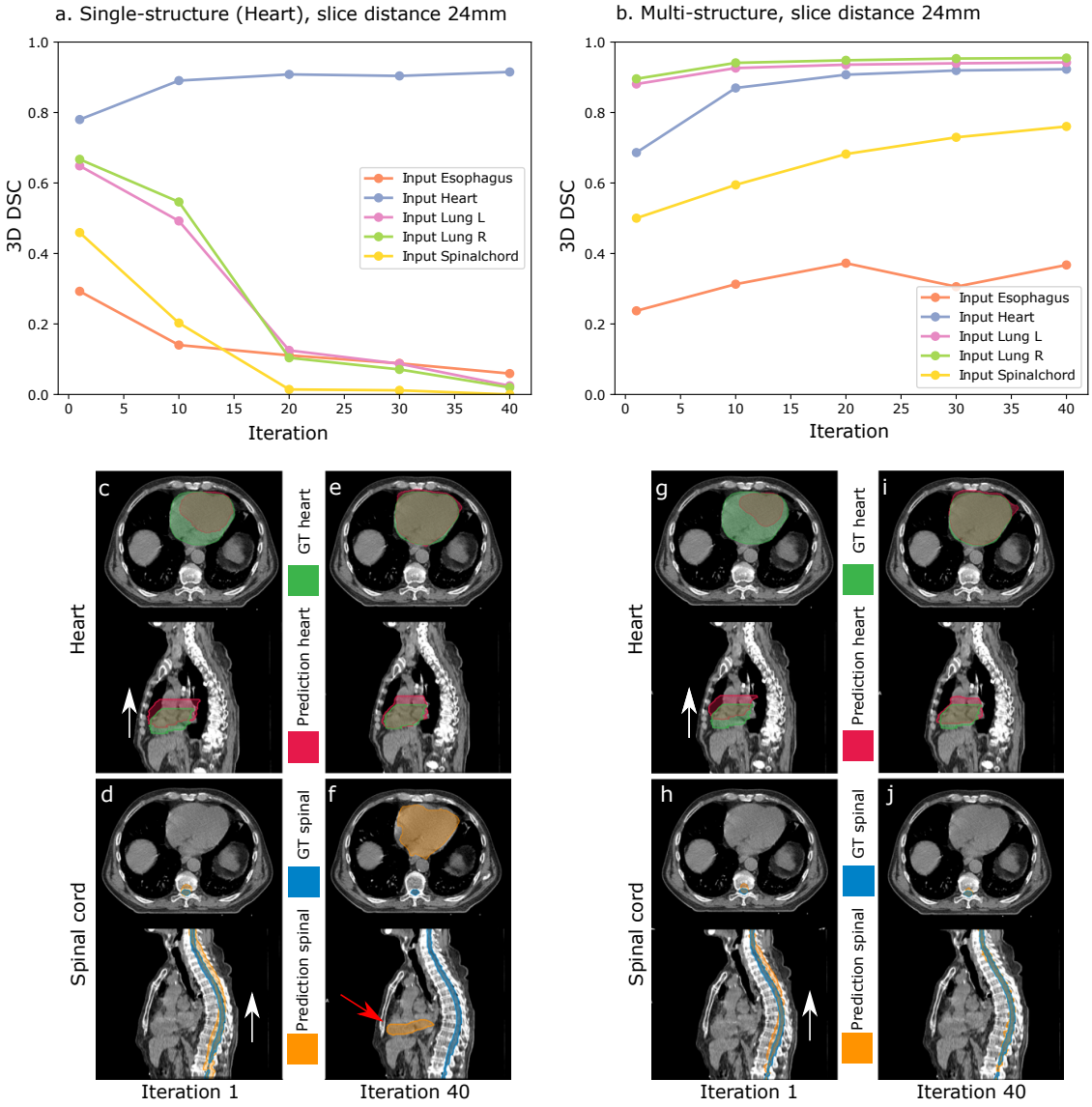


Figure 3.5: Model performance and illustrated output for different structures at selected epochs in training for a single- and multi-structure model at slice distances $d = 24\,\text{mm}$. (a) Single-structure, heart-only model. (b) Multi-structure model. Illustrations: (c) Heart-only model, Iteration 1: Prediction of Heart. (d) Heart-only model, Iteration 1: Predicting Spinal cord. (e) Heart-only model, Iteration 40: Predicting Heart. (f) Heart-only model, Iteration 40: Predicting Spinal cord. (g) Multi-structure model, Iteration 1: Predicting Heart. (h) Multi-structure model, Iteration 1: Predicting Spinal cord. (i) Multi-structure model, Iteration 40: Predicting Heart. (j) Multi-structure model, Iteration 40: Predicting Spinal cord. At iteration 1, a *copy-paste* behavior of the model is observed, where the prior contour becomes the output of the model. For iteration 40, heart specific segmentation is observed for the heart-only model and segmentation based on context for the multi-structure model.

structure model on structures not included in the training set, the additional image information improved segmentation. For example, the contour-only model trained exclusively on lung has a DSC of 0 for the heart, whereas if the image information is included as context a DSC of 0.4 is achieved. In contrast, the multi-structure model achieved a DSC of 0.92 and 0.95 for the CT/GT+CT and GT+CT model respectively.

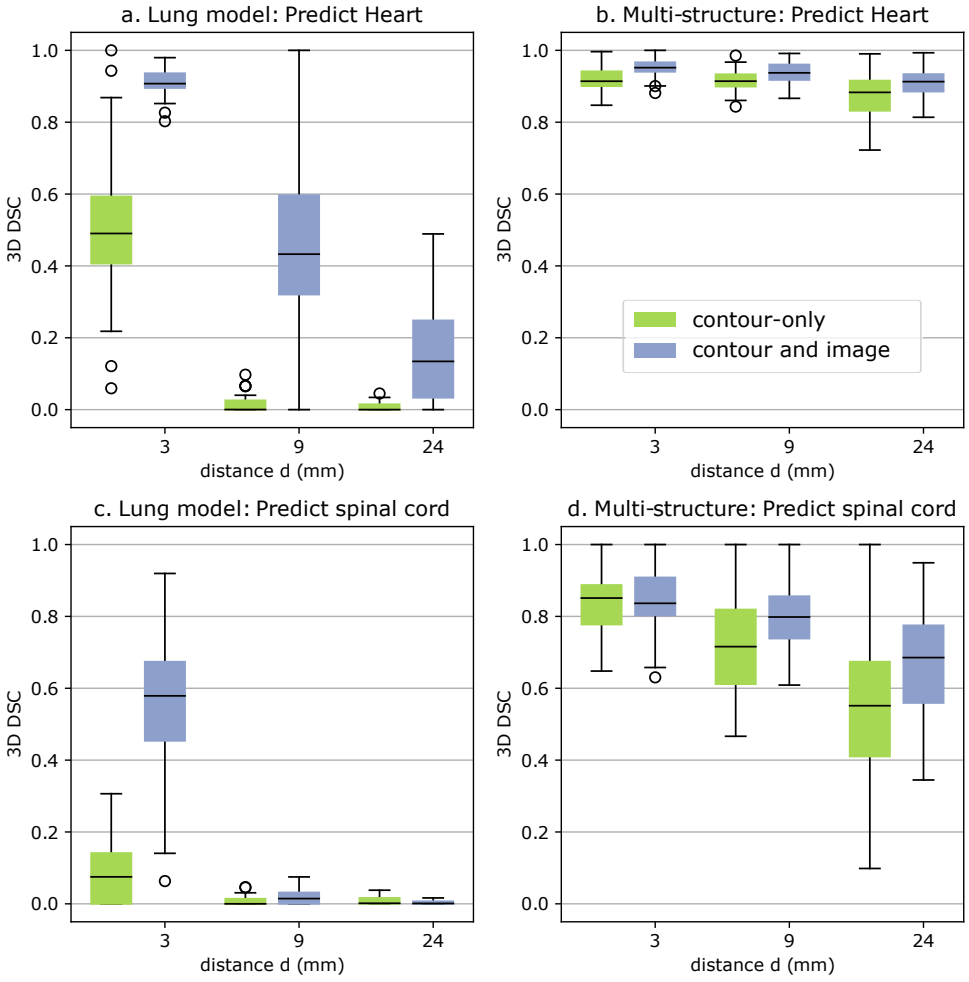


Figure 3.6: DSC for predicting heart and spinal cord using a single-structure, lung-only model (a, b) and a multi-structure model (c, d). The models are trained and tested using a fixed slice distance d and using a fixed set of input: (GT)+CT (green) and (CT/GT)+CT (purple). The input variants are illustrated in Figure 3.3.

3.2.2.3 Network architectures

The performance of the standard U-Net, RU-Net, R2U-Net, Attention U-Net and dual-encoder U-Net are summarized in Table 3.2 for the multi-structure model.

All networks showed an overall similar performance across all structures and slice distances d. The heart and lung structures all showed similarly high performance for all structures at all distances. Of these, the R2U-Net and Attention U-Net marginally outperformed the other networks.

The performance between the networks varied strongest for the oesophagus and spinal cord. For the oesophagus at slice distance $d = 3\,\text{mm}$, the standard U-Net had the highest performance with a DSC of 0.67 ± 0.09 and the dual encoder the lowest with a DSC of 0.52 ± 0.13. At larger slice distance d both of these networks had a DSC of about 0.3 at $d = 24\,\text{mm}$, whereas the R2U-Net and Attention U-Net achieved a DSC of approximately 0.5. For the spinal cord the standard U-Net was the only network with a significantly lower DSC of 0.73 for all slice distances, while the modified U-Net architectures have a DSC of nearly 0.8 for all distances.

Table 3.2: DSC for oesophagus, heart, lung and spinal cord, using different network architectures trained as a multi-structure model. Results are shown for a standard U-Net architecture, Attention R2U-Net and Dual-Encoder U-Net. Each model is trained on the multi-structure training set. Individual models are trained for each architecture at three slice distances and evaluated on the distance used for training ($d = 3\,\text{mm}$, $d = 9\,\text{mm}$, $d = 24\,\text{mm}$). Significantly better performing networks using Wilcoxon signed-rank test ($p<0.01$) are highlighted in bold.

Organ	d	U-Net	Dual-Enc.	RU-Net	R2U-Net	Att. R2U
Oesoph.	3	$\mathbf{0.67 \pm 0.09}$	0.52 ± 0.13	0.53 ± 0.11	0.56 ± 0.12	0.56 ± 0.15
	9	0.57 ± 0.10	0.50 ± 0.13	0.55 ± 0.14	0.57 ± 0.13	$\mathbf{0.58 \pm 0.15}$
	24	0.33 ± 0.11	0.39 ± 0.15	0.44 ± 0.12	$\mathbf{0.50 \pm 0.13}$	0.49 ± 0.12
Heart	3	0.92 ± 0.02	0.93 ± 0.02	0.93 ± 0.02	$\mathbf{0.95 \pm 0.01}$	$\mathbf{0.95 \pm 0.02}$
	9	0.91 ± 0.02	0.93 ± 0.01	0.92 ± 0.02	$\mathbf{0.95 \pm 0.02}$	$\mathbf{0.95 \pm 0.01}$
	24	0.92 ± 0.03	0.92 ± 0.05	0.92 ± 0.01	0.94 ± 0.02	$\mathbf{0.96 \pm 0.01}$
Lung	3	0.97 ± 0.01	0.97 ± 0.01	0.96 ± 0.01	0.97 ± 0.01	0.97 ± 0.01
	9	0.96 ± 0.01	0.97 ± 0.01	0.96 ± 0.01	0.97 ± 0.01	0.97 ± 0.01
	24	0.95 ± 0.02	0.96 ± 0.02	0.96 ± 0.01	0.97 ± 0.01	0.97 ± 0.01
Spinal c.	3	0.73 ± 0.13	$\mathbf{0.80 \pm 0.12}$	0.78 ± 0.11	0.79 ± 0.09	$\mathbf{0.80 \pm 0.10}$
	9	0.74 ± 0.10	$\mathbf{0.80 \pm 0.11}$	0.75 ± 0.12	$\mathbf{0.80 \pm 0.12}$	0.78 ± 0.11
	24	0.72 ± 0.11	0.75 ± 0.11	0.76 ± 0.10	$\mathbf{0.78 \pm 0.11}$	0.77 ± 0.12

3.2.3 Discussion

3.2.3.1 Single- and multi-structure models

The comparison of single-structure and multi-structure models in Figure 3.5 shows that a single-structure model is not able to generalize the contextual information to organs different from itself. A model that has only been trained on one structure (e.g. the heart) learns to contour exclusively that structure, regardless of the provided context, as shown in Figure 3.5f, where a spinal cord contour was provided for context. Instead of predicting the spinal cord, the model falsely segmented the heart. For multi-structure training (Figure 3.5j), the models learned to predict different organs based on the spatial context input without explicit labelling of the organ.

While single-structure models have previously been suggested to improve the segmentation performance [18], [19], the contextual information in single-structure model is not used to identify what structure is to be segmented. The single-structure models could be useful in a very narrow range of applications. It would require a separate model for each structure to be contoured and an additional step of selecting which model to use depending on what you want to contour. More importantly, it can be difficult to find a representative training set for some structures (such as tumours) and it can be time consuming to create a high quality dataset.

Note, that if contextual information is available to the single-structure model (Figure 3.5a), the initial segmentation at epoch 1 was better than at epoch 40 for organs that were not included in the training set. This is attributed to an initial learning of "copy pasting" of the GT contours to the current slice. Model outputs are visualized after the first epoch in Figure 3.5. What the model initially learned is to output the provided prior segmentation, but not make any alterations to it. After few epochs, model prediction improved for the single organ included in the training set but decreased for other organs due to learning of specific features of the structure in the training set. As the model learned to predict the structure in the training set. As there were no alternative structures in the training set with the same label, the model did not need to rely on the prior contour information to locate which structure to segment. In contrast, the multi-structure model

increased in performance throughout the model training for all structures tested. While all structures had the same label, given the binary mask during training for each structure, the model needed to learn which structure to predict, given a particular input. This is particularly important in body regions, where multiple structures are present on the same axial slice. For example, an axial slice across the chest, may contain structures of heart, lung, oesophagus, and spinal cord. Without the prior contour information the model would fail to distinguish which structure to segment on this given slice.

This experiment has demonstrated, that given a user-provided contour on a single slice, the model can distinguish between several structures that may be present on the same slice and segment the structure of interest. This has not yet shown that contextual information can be leveraged to segment previously unseen structures. Currently, the contextual information may indicate that a specific structure used in training is to be segmented on the given slice. However, the interactive model should also be able to do this on a structure not included in the training set. The generalizability of the contextual learning approach is investigated in Section 3.3

3.2.3.2 Input variants

Only small differences in performance between the different input variants were observed for structures included in the training set. However, for structures excluded from the training set (Figure 3.6a,b) additional image information increased the DSC.

The improvement in segmentation is attributed to the context information between the CT and contour on this slice. Providing only the GT as a contextual input may focus the network on this image region or enable the network to learn the features of the contour that identify different structures (e.g. it identifies based on the contour that the heart is intended to be segmented by the user). For this contour-only contextual input, the network may not segment based on the provided context but based on structure-specific features. The model does not understand the user intends to contour, because it does not see the corresponding image slice to the contour. In contrast, if additionally the CT slice corresponding to the contour

is added as an input, the network can understand the relation between the prior image slice and contour and apply this to the CT slice. This may result in the model learning features that are related to the boundaries of the contour or features within the contour relating to the image slice. What it is that the model learns is inherently difficult to visualize however.

As a result of this understanding how CT and contour relate, structures that were not included in training improve for the single-structure model (see all distances d for the Lung model predicting heart, as well as the Lung model predicting the spinal cord for $d = 3\,\text{mm}$). This is qualitatively explained by the model guiding the model towards the image region in which the contour was provided. For the multi-structure model, the additional image information also improved segmentation performance, particularly at larger slice distances d. At small slice distances the relationship between image slice and contour information may be less relevant to the segmentation, as simple copying of the prior contour may result in a good enough segmentation, as mentioned above. However, at larger slice distances the contour provided for context and the corresponding structure on the target image slice may have shifted to a different image region. Therefore, it becomes more important for the model to interpret and learn what the user intended segment, rather than to focus on the image region indicated by the prior contour.

Taking the heart segmentation as an example (Figure 3.6a) the input variant using only contour information for context, had a DSC of 0.5 at slice distance $d = 3\,\text{mm}$, whereas the addition of the corresponding image slice achieved a DSC of nearly 0.9. At larger slice distances d, the single-structure model failed to segment the heart completely, whereas the image- and contour information model had a DSC of 0.5 $d = 9\,\text{mm}$ and 0.15 $d = 24\,\text{mm}$.

3.2.3.3 Network architectures

The five architectures were tested for multi-structure training and the results are shown in Table 3.2. All models achieved robust segmentation for all structures investigated. With all but the RU-Net achieving the highest segmentation performance

for a given scenario. For the lung, all model achieved very high DSC, as this is a very simple structure to segment and therefore, only marginal differences were observed.

When using this model in an interactive contouring application, evaluation at larger distance from the original input is more important and also is expected to be more challenging, as the target contour likely differs more from the user-provided contextual input. As pointed out before, models with low slice distance d have a tendency to be overconfident in the prior input and are prone to copying rather than predicting based on context and therefore, are expected to be less robust to segmentation of previously unseen structures.

At a slice distance of $d = 24\,\text{mm}$ between target image slice and contextual input, the R2U-Net and Attention U-Net achieved the best performance for the different structures. Both achieved considerably higher DSC for the oesophagus which is particularly challenging. The inter-observer variability for clinicians on the oesophagus in the AAPM 20217 dataset in terms of DSC was 0.8 [132]. This shows that even for clinicians the oesophagus is difficult to contour consistently. Additionally, it is a very small structure. The down-sampling used for inference (512 to 256), results in an even smaller number of foreground pixels in training. This results in an underrepresentation in the training set and thus may contribute to the lower segmentation performance of this structure. The increased receptive field of the R2U-Net may help with the segmentation of this more challenging structure. Additionally, the residual recurrent layers allow for learning of deeper features without increasing the number of model parameters. The Attention U-Net's attention gate has a soft-attention mechanism that can focus the model on the image region.

The RU-Net slightly outperformed the standard U-Net for most structures and slice distances. Residual layers may have contributed to more stable model training and faster convergence. Thus, accounting for better segmentation.

The Dual-Encoder network that was investigated in analogy to Zheng et al. [19] did not show better performance than the R2U-Net and Attention U-Net. The Dual-Encoder architecture has a larger number of parameters than the remaining networks

due to the separate encoding pathways. However, the additional computational complexity did not improve segmentation performance. The Dual-Encoder network may also suffer from not including feature maps created when processing the prior information into the skip connection, as the contextual information is only provided at the very bottom of the U-Net.

Overall, the investigated networks achieved similar DSC as fully automatic segmentation approaches for all structures investigated except the oesophagus [143], [144].

3.2.4 Conclusion

This section investigated whether contextual information can be used to segment different structures of interest, using just a single model. A single-structure model is not able to segment different structures. However, if different structures are present in training, these structures can be segmented by providing an individual contour, without specifying explicitly which structure to segment.

Additionally, it was shown that if a contoured image slice is available, the input of both contour and image slice is better than just the contour input. This is attributed to the information gained when relating the contour to the image and thus, being able to learn features specific to the structure to be segmented. The contour alone instead is only able to indicate to the model in which region the structure of interest may be found on the target image slice. But at larger slice distances between context and target, this is not enough information to successfully segment a structure.

In the end, different network architectures were explored and compared to the standard U-Net architecture. Residual, recurrent and Attention blocks all proved to be beneficial to the segmentation. However, it remains unclear if the attention gates provided additional benefits, as the performance on this datasets was very similar.

The next section builds on this work and explores whether structures previously unseen to the model can be segmented. Generalizing to unseen objects requires the model to rely more on context information rather than learning structure-specific information.

3.3 Generalizable Contextual Deep Learning

Initially published as Ref. [2].

This section builds on the previous exploration of contextual deep learning and focuses on the generalizability of this approach to structures that were not included in the training set. This will show if the contextual information is sufficient to segment new structures, rather than being limited to a specific structure defined by the training set. This section introduces a novel training approach that is required to achieve generalizable contextual deep learning.

3.3.1 Materials and Methods

3.3.1.1 Contextual deep Learning for 3D segmentation

In this section, the task of predicting a 3D segmentation given a user-provided contour on a 2D slice is investigated. The principles of contextual deep learning were introduced in Section 3.2 and it was investigated if a prediction of a contour at a distant target slice is possible. Here, the prediction of the entire 3D structure is investigated more closely. In clinical practice, it is expected that a user would select an image slice and delineate a structure on that slice. Then, the structure would be segmented by the model on the remaining image slices given the initial user input. Given the contextual model introduced before, two different contour prediction methods can achieve segmentation of the 3D image when one input contour is given, here these are referred to as iterative and direct prediction.

In the first approach, the initial input image slice (the central slice) was chosen as the contextual input for the segmentation of all the remaining slices. This method is referred to as *direct prediction*. In the second approach, the expert contour at the central slice was used as the contextual input and the adjacent image slice as the target slice. The predicted target contour becomes the contextual input for the next slice. The process was repeated until all slices are contoured. This method is referred to as *iterative prediction*. The process of the iterative prediction is illustrated in Figure 3.7.

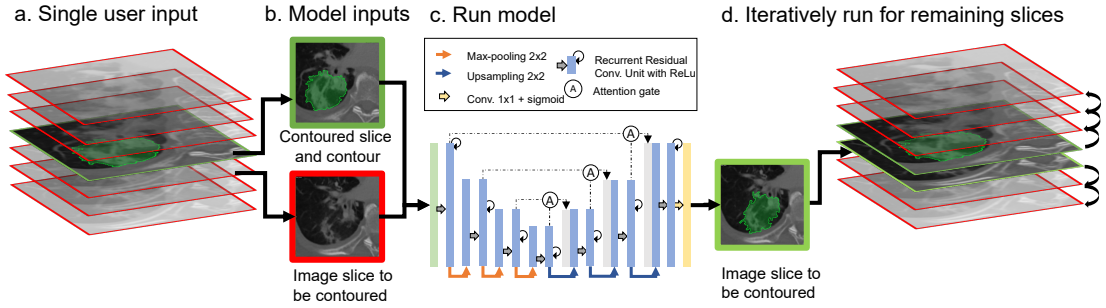


Figure 3.7: Illustration of 3D segmentation using contextual deep learning. (a) After contouring the first slice, (b) the contoured slice and the slice to be contoured are used to predict the segmentation by (c) running the model. (d) The predicted contour is used as an input to predict the next adjacent slice. This is repeated for all remaining slices.

3.3.1.2 Network architecture

The Attention U-Net and R2U-Net were identified as the most promising architectures for contextual segmentation model (see Section 3.2). When applied to this problem the Attention U-Net showed a better performance. The Attention U-Net architecture is visualized in Figure 3.7c.

The networks were trained using pixelwise binary cross entropy loss [140]. The Adam optimizer [141] was used with an initial learning rate of 1×10^{-5} and a batch size of 4. Data augmentation was applied on-the-fly during training using random affine matrix transformations consisting of translation (± 10 mm), rotation (± 10 degrees) and scaling ($\pm 5\%$).

Each model presented in this study was trained from randomly initialized weights. Therefore, model performance may vary with initialization. To mitigate the variability and achieve the best possible performance for each model, 8 models were initialized for training. After 5 epochs the worst 50% of the models are discarded. This was repeated each time the number of epochs doubles until 1 model remains. Then the remaining model was trained for up to 20 more epochs using early stopping.

3.3.1.3 Data

This study included CT images and delineations of 19 different structures of interest from various openly available datasets. An overview of all the used structures is shown in Table 3.3.

Table 3.3: Overview of number of slices used for training, validation and testing for different anatomical structures, as well as the total number of contoured slices available in the respective dataset. Note, that there are several other structures within the original datasets that were excluded due to their relatively small size or sparse annotations.

Set	Structure	Train	Val.	Test	Total	Ref
A	Heart	1000	200	318	2096	[132]
	Heart	1000	200	404	3741	[133]
B	Oesophagus	2000	200	802	5247	[132]
	Lung (left)	2000	200	805	5057	
	Lung (right)	2000	200	814	5305	
	Spinal cord	2000	200	1315	7907	
C	Lung tumour	2000	200	1219	10961	[133]
D	Pancreas	2000	200	987	8783	[134]
	Pancreas tumour	2000	200	273	2537	
E	Liver	2000	200	5052	27449	[134]
	Liver tumour	2000	200	2172	10972	
F	lymph node (level II-IV, left)	2000	200	450	3136	[135]
	lymph node (level II-IV, right)	2000	200	405	3099	
	Submandib. (right)	976	127	130	1233	
	Submandib. (left)	934	130	162	1266	
	Parotid (left)	1359	200	210	1806	
	Parotid (right)	1387	200	222	1832	
	Brain	2000	200	523	3454	
	Brain tumour	2000	200	260	2878	
G	Spleen	0	0	1363	1363	[134]

The 3D CT volumes and corresponding contouring data were randomly split into a training, validation and test set using a 60:20:20 split, respectively. 2000 image slices per structure were randomly selected from the training data in order to limit the computational resources required and to create a balanced representation of each structure in the training set. Structures that were delineated on only very few patients or that extend over fewer than 10 slices were excluded from this study. 2000 slices were chosen, as this was roughly the number of suitable lung tumour segmentations. While the total number of tumour segmentations far exceeds this number, a lot of the segmentations were gross tumour volumes (GTV) of tumours extending only over a few slices.

3.3.1.4 Preprocessing

CT pixel intensities were clamped to the range from -1000 HU (air) to 2000 HU (cortical bone) to avoid artifacts in the high and low HU range. The CT image slices of 512x512 pixels were rescaled to the network input size of 256x256 pixels. The rescaling was used to decrease computation time and reduce memory consumption.

3.3.1.5 Experimental details

Training set diversity Following the optimization based on the previous experiments, the ability of the model to segment previously unseen structures was investigated. For this, a series of models was trained, for which the size of the training set was successively increased. Figure 3.8 shows the various models, together with the respective set of structures included in the training set. All models were evaluated on all structures. This permits the assessment of whether or not the ML model generalizes to unseen structures. The order in which the datasets were added to the training set was chosen arbitrarily. A combinatorial investigation would be prohibitively computationally expensive. A single GeForce GTX 1080 GPU was used for all experiments. The inference time is approximately 0.1 seconds per image slice.

In the training set, the previously contoured and target slices were chosen carefully to support learning from context. The inter-slice distance between the contoured and target slice was particularly important. If the previously contoured image slice and the target image slice were adjacent image slices, there was insufficient change between the image slices for the model to learn the context and, in this case, the model learnt to copy the existing contour rather than generate a segmentation mask for the new slice. Target slices far away from the contoured slice are less relevant for the application of this model, as the target image slices corresponding to large inter-slice distances often lie outside of the structure to be segmented. The slice distance was sampled from a range of distances to balance between the two extremes and create a more diverse training set that was not biased towards a specific slice distance. It was important to sample the distance from a range of distances relative to the size of the structure and not use one predefined set of

distances for all structures. Thus, different structures with a variety of size (e.g. lungs, heart, tumour) were equally well represented. The previously contoured image slice must be chosen such that it lies within the structure, otherwise no contour exists from which context can be determined. However, the target image slice may lie outside the structure volume to allow the model to learn not only where to contour but also when to stop contouring.

In this experiment, target image slices and contextual input slices were selected at a random distance L_D from each other following a draw from a Gaussian distribution ($\mu = 0.25L_S$, $\sigma = 0.1L_S$) relative to length L_S of a structure S. The length L_S of the structure was determined by the inferior and superior edge of the structure. A larger distance d (e.g. $d > L_S$) would mean that the target slice often is outside the area of interest (structure to be segmented) and thus only predicting an empty mask. A small distance would incentivize the model to learn copying, as discussed in the previous section. Additionally, no absolute value for all samples may be determined, as the structure length L_S varies between patients among the same structure, as well as between different types of structures.

Evaluation was performed using iterative prediction from the central slice of the ground truth contour.

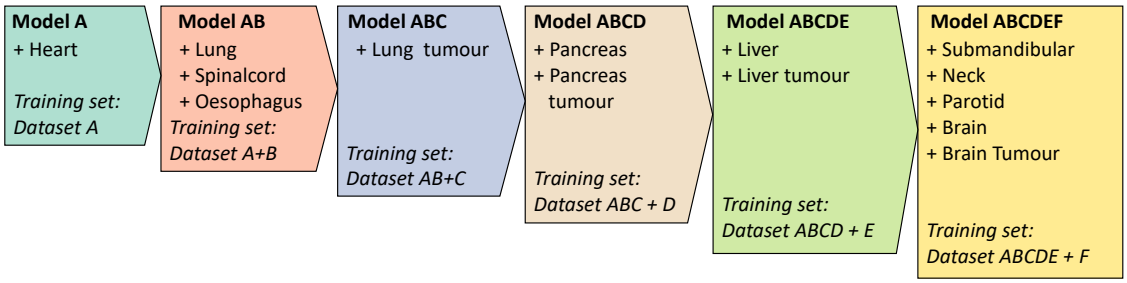


Figure 3.8: Series of models trained on successively expanding training sets. Each new model uses the training set of the prior model and includes new structures previously excluded from the training set.

Iterative and direct prediction The models that have been trained with the various training sets (see Figure 3.8) were evaluated using direct and iterative prediction methods. The two approaches are compared for all structures investigated

on the final Model ABCDEF. The slice distance from the provided contextual slice will be the focus of this comparison. The experiment will show which affects 3D segmentation performance more: the little errors from a small slice distance that may accumulate via iterative prediction, or the big error from one large slice distance between target and context.

Propagation using image registration To compare the deep learning model with another approach, propagation via 2D image registration was implemented. To propagate a contour using image registration, the prior input image slice and the target image slice are aligned via a displacement field. If the same displacement field is applied to the contour, the contour corresponds to the deformed prior image slice, which also is an estimate to the contour of the target image slice.

Here, a diffeomorphic demons algorithm was applied for 2D image registration, for more details see [145]. The optimization was performed for 20 iterations. The sum of squares of intensity distance (SSD) was used as the similarity metric to compute pixel based similarity between pixels.

For a more intuitive understanding of the registration process, the registration process is shown for an example in Figure 3.9. It shows how the moving image is aligned with the fixed image. In terms of the contour propagation problem discussed before, the fixed image corresponds to the target image slice whose contour is to be predicted, and the moving image corresponds to the prior input image whose contour is given by a user input.

The displacement field (Figure 3.9d) between target image (a) and prior image (c) is calculated using the Demons algorithm [145]. The deformation field is then applied to the prior contour, to obtain an estimate of the target contour. One can visually see that the deformation was successful by applying the deformation field to the prior image, which looks visually similar to the target image.

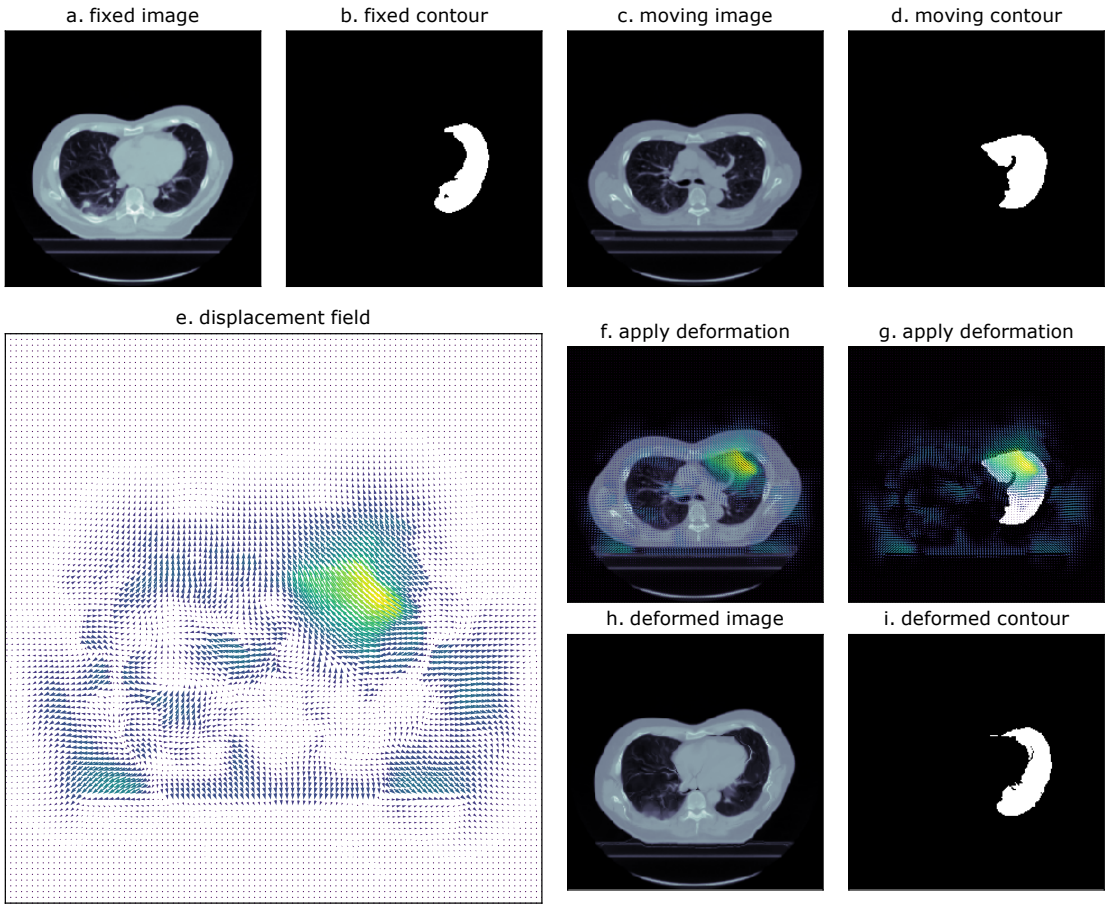


Figure 3.9: Example of contour prediction using 2D image registration. (a) fixed image, (b) fixed contour, (c) moving image, (d) moving contour, (e) displacement field, larger displacement vectors are indicated in yellow (f) displacement field is overlayed to the moving image, (g) displacement field is overlayed to the moving contour, (h) deformed image, (i) deformed contour. (a) The fixed image (target image slice) and (c) the moving image (prior image slice) are used to calculate (e) the displacement field. The displacement field is applied to the prior contour, as shown in (g). Through this, an estimate for the target contour is obtained. The estimate is shown in (i), whereas the ground truth target contour is shown in (b).

3.3.1.6 Evaluation measures

The quality of the segmentation was quantified using the 2D Dice Similarity Coefficient (DSC) [142], 90% Hausdorff Distance (HD) [146] and relative Added Path Length (APL) [147].

The DSC measures the overlap between two areas or two volumes, A and B. The DSC is defined as

$$\mathrm{DSC}(A,B) = 2\frac{|A \cap B|}{|A| + |B|} \tag{3.2}$$

The HD measures how far two subsets of a metric space are from each other. The HD $d_{\mathrm{HD}}(A,B)$ is defined by

$$\mathrm{d_{HD}}(A,B) = \max\left\{\sup_{a\in A}\inf_{b\in B}\mathrm{d}(a,b), \sup_{b\in B}\inf_{a\in A}\mathrm{d}(a,b)\right\} \tag{3.3}$$

where d denotes the Euclidian distance.

The APL is the length of contour drawn when editing a segmentation. Because the absolute length of a contour varies between patients and different structures, the APL is reported relative to the ground truth contour length. A tolerance of 2 mm between predicted result and ground truth was used for APL. A low relative APL means that few edits are necessary, whereas at a relative APL of the full contour needs to be drawn by the user. [147]

The DSC and HD are widely used to quantify the performance of segmentation tasks. However, these measures are not necessarily the best choice in a clinical context, where the contouring speed is more important. The time needed for an expert to review and adjust the predict contours shows better correlation with the APL than with DSC and HD [147].

To compute the performance measures, the masks that were generated were upsampled to the original 512x512 imaging resolution using bi-linear interpolation. It is important to evaluate the performance after upsampling because a clinician would review these contours at the original image resolution.

3.3.2 Results

Results for all evaluation measures and structures can be found in Appendix B.1 and B.2, and show similar results. In the following, the relative APL or DSC are reported.

3.3.2.1 Training set diversity

Figure 3.10 shows the evolution of the model performance as new structures from an additional dataset are added. Here, results for iterative prediction are shown, averaging the slice-wise evaluation measures across all structures for each dataset. Model A, with only the heart in the training set, showed the worst performance with a DSC of 0.86, 0.03, 0.24, 0.21, 0.40, 0.13, 0.33 and a relative APL of 0.45, 0.96, 0.86, 0.88, 0.91, 0.95, 0.85 for the test datasets A, B, C, D, E, F, G respectively. The successive models included additional structures in the training set and showed a progressive overall improvement in performance compared to model A. The best performing model was the ABCDEF model with a DSC of 0.87, 0.77, 0.62, 0.49, 0.66, 0.69, 0.80 and a relative APL of 0.44, 0.26, 0.60, 0.62, 0.56 0.43, 0.35 across the different datasets.

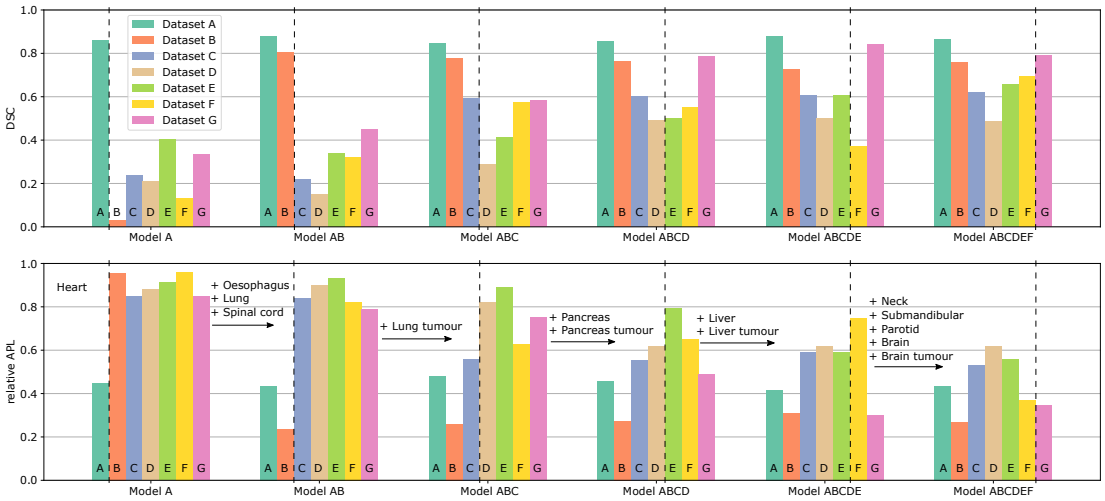


Figure 3.10: Performance of models with successively increasing training set on different test datasets using iterative prediction. Datasets on the left of the dashed line, indicate that this dataset was included in the training set of the respective model, whereas datasets to the right are excluded from the training set.

The boxplots in Figure 3.11 show the relative APL for the heart, lung tumour, pancreatic tumour, liver, submandibular gland and spleen. An equivalent figure for the rest of the structures and evaluation measures is provided in Appendix B.1. The most significant decrease in the relative APL for a given structure was observed

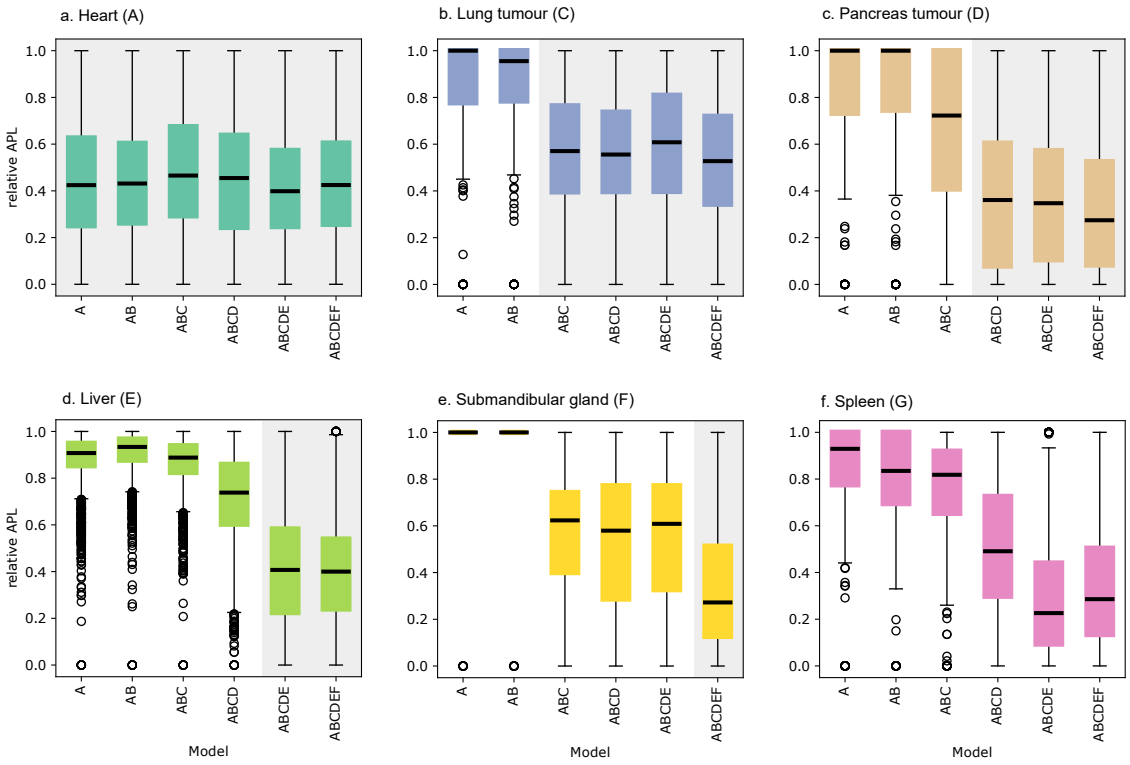


Figure 3.11: Boxplots of the performance of different models on selected structures: (a) heart, (b) lung tumour, (c) pancreatic tumour, (d) liver, (c) submandibular gland and (f) spleen. The grey shaded background indicates the models that include the structure in their respective training set.

when that structure was included in the training set of a model. However, observe that including additional structures in the training set also resulted in segmentation improvements to structures that were not included in the training set. For example, the relative APL for pancreas tumour, liver, submandibular gland and spleen decreased on adding the lung tumour in the training set (Model AB to Model ABC). The spleen was excluded from all models trained. Yet, the relative APL of the spleen decreased from 0.85 ± 0.22 for model A to 0.35 ± 0.21 for the final ABCDEF model.

3.3.2.2 Iterative and direct prediction

In Figure 3.12 the performance of model ABCDEF for direct and iterative prediction is compared at different inter-slice distances from the central slice. Figures for the remaining structures and evaluation measures can be found in Appendix B.2.

Predictions closer to the central slice had a lower relative APL, whereas

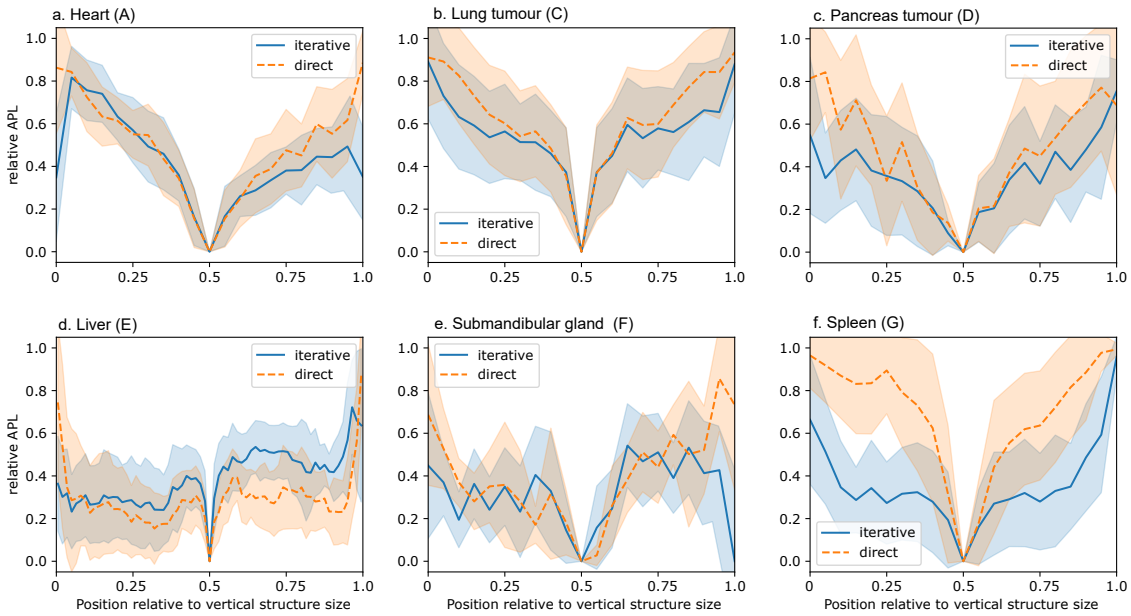


Figure 3.12: Comparison of direct (orange-dashed line) and iterative (blue line) prediction using Model ABCDEF on different structures: (a) heart, (b) lung tumour, (c) pancreatic tumour, (d) liver, (e) submandibular gland and (f) spleen. The shaded area indicates the standard deviation of the APL for different test samples.

the relative APL increased towards the superior and inferior most slice of each structure. The structures included in the training set (Figure 3.12a-e) showed similar performance between the direct and iterative prediction methods. In contrast, the spleen (Figure 3.12f) had a lower relative APL, when using iterative prediction as compared to direct prediction.

3.3.2.3 Propagation using image registration

So far, it has been shown that contextual deep learning can generalize and segment previously unseen organs. In this section, a baseline is established to compare the contour prediction using contextual deep learning to existing methods. 2D image registration is used in some commercial applications [148] for contour prediction in medical imaging, and has previously been used as a baseline for deep learning prediction approaches [18].

Figure 3.13 summarizes the performance using registration and compares them to the performance of model ABCDEF. This is shown for direct and iterative prediction. This shows that Model ABCDEF (iterative) achieved higher DSC

and lower APL for datasets A, B, E, F compared to registration, and a similar performance for datasets C, D and G. The same holds true for direct prediction, for all but dataset G. Direct prediction of dataset G (spleen, unseen structure) performed worse using Model ABCDEF as compared to registration.

Registration using iterative prediction achieved higher DSC and lower APL than registration using direct prediction for all datasets.

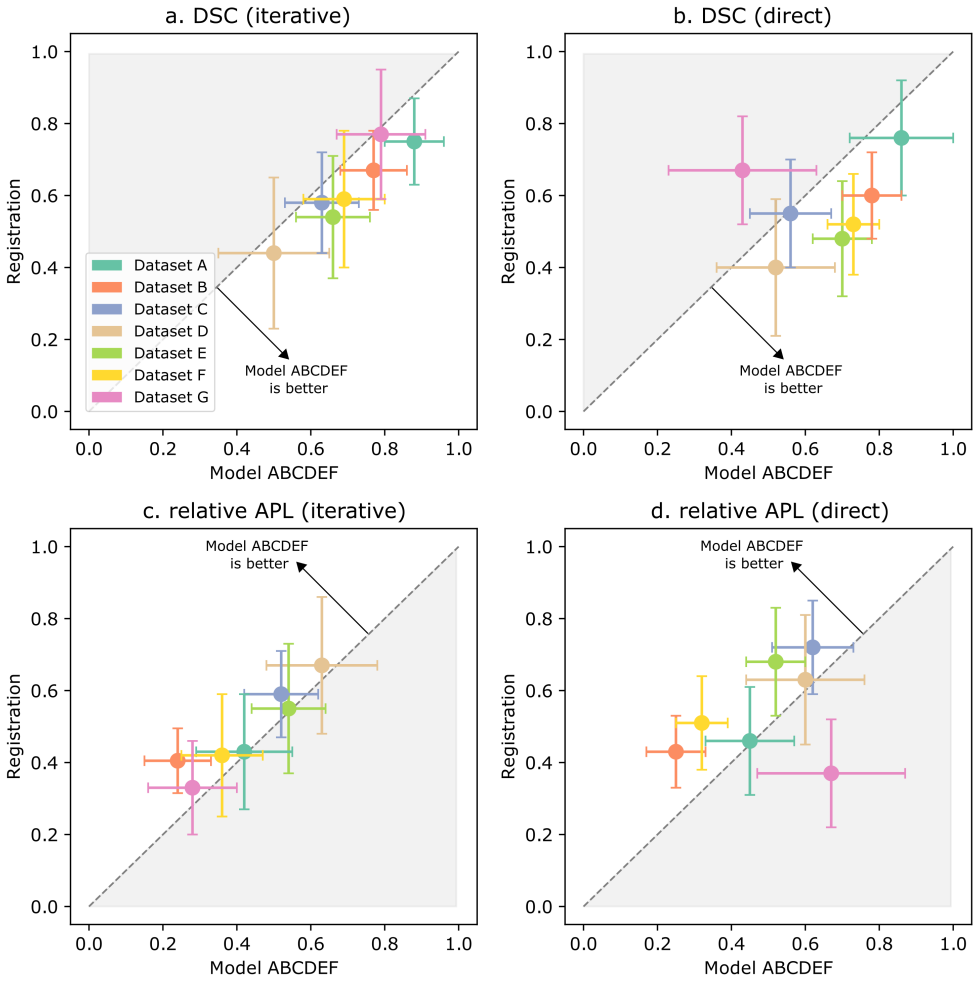


Figure 3.13: Comparison the registration approach and Model ABCDEF for different datasets in terms of 2D DSC and relative APL. Error bars indicate the standard deviation.

3.3.3 Discussion

3.3.3.1 Training set diversity

The addition of training examples of structures would be expected to improve the performance of a model in delineating those structures. This can be observed in Figures 3.10 and 3.11. However, with subsequent addition of structures to the training set, not only did the performance improve for the structures newly included in the training set, but also for the structures excluded from the training set, as seen in Figure 3.11. For example, when adding lung tumours to the training set (going from model AB to model ABC), the performance improved for lung tumours, pancreatic tumours, liver, submandibular gland and spleen. This improved performance cannot originate from learning any structure-specific features within the training set, but must be attributed to learning contextual features and making predictions based on the context from the provided input contour and image slice. For example, Model ABCDEF achieved a relative APL of 0.35 for the spleen, despite not having been trained on this structure before. This suggests that user interaction and thus time spent contouring can be decreased for de-novo contouring, when using such a model. Additionally, it is shown in Appendix B.3, that most structures can achieve a high segmentation accuracy, even if that structure was excluded from training set and the training set consisted of all remaining structures. In contrast, a model trained on only a single structure fails to generalize as shown in Appendix B.4.

The ability to generalize to unseen structures distinguishes this approach from previous work, in which previously contoured image slices are used to assist in segmentation [18], [19].

The gradual incremental improvement of the segmentation when adding additional structures is visualized in Figure 3.14 for a randomly selected head and neck case (Dataset F). Models A to Model ABCDE never had these structures included in training, but some of these already can contour the structures, given an initial user input. The segmentation improves, the larger the training set is expanded, with the exception of Dataset E (Liver).

Certain structures appear to have a larger impact on the model's ability to generalize to unseen structures. For example, four new structures were added to the training set between model A and model AB, but the improvement in segmentation performance is almost exclusively seen in the newly added structures. In contrast, between model AB and ABC, the segmentation performance improved for all structures, although only lung tumours were added to the training set. The improvement in segmentation of unseen structures and thus, the ability to generalize, could potentially be attributable to the high diversity in the tumour appearance.

At the same time, the structures that have already been included in previous models showed further improvement upon expanding the training set by new structures, see Figure 3.11b and c. This suggests that the quality of segmentation of an individual structure benefits from an increased diversity in the training set. This might be attributed to a refinement of the contextual features used in the segmentation over the shape-specific features of each structure. In this work, it has not been tested if the incremental changes between subsequent models are statistically significant due to the small and variable size of the test dataset for the different structures. This needs to be addressed in future work.

The interactive contouring approach proposed here achieved similar or higher DSC than fully automatic segmentation approaches reported for the heart [143], lung [143], spinal cord [143], lung tumour [144] and pancreas tumour [46], but lower performance for the oesophagus [143] and spleen[46]. This comparison is limited to studies in which fully automatic solutions were reported for the same dataset as used in this section. It is noted that the current approach benefits from input of an initial manual contour.

The contextual deep learning approach used here did not achieve a DSC comparable to the best performing fully-automatic models for unseen structures such as the spleen [46]. Note, however, that those fully-automatic models were trained on a dataset of images that include the spleen, whereas the model proposed in this work, was trained on images that did not include the spleen. Therefore, this indicates that the current approach can be applied to unseen structures and

may decrease delineation times for such structures for which no automatic solutions exist. Furthermore, contextual deep learning achieved high DSC values for pancreas and lung tumours compared to fully-automated solutions, highlighting the benefits of this approach for segmentation of highly variable structures.

The proposed method has not been evaluated on non-anatomically defined structures (e.g. CTV or PTV). The model is expected to perform poorly on such structures because it has been trained on anatomically defined structures and as such, the model likely makes predictions based on image intensity.

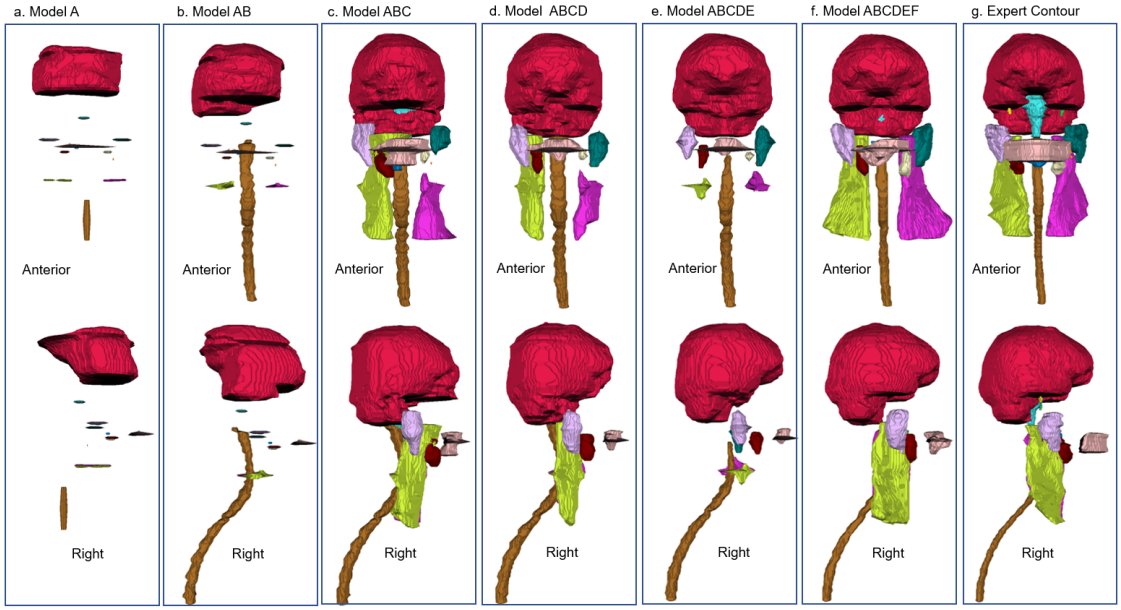


Figure 3.14: Visualization of segmentation outputs of a randomly selected head and neck test case from Dataset F. Figure (a)-(f) show the segmentation result of different models using iterative prediction. Figure (g) shows the expert contour.

3.3.3.2 Iterative and direct prediction

While direct prediction showed similar performance as iterative prediction for most structures, iterative prediction performed significantly better for an anatomical structure that was omitted from the training set, i.e. the spleen.

In the direct prediction approach, the inter-slice information is not used and the slice distance to the user input slice varies. This may increase the error of the direct prediction approach. In the iterative approach, an initial error or

error in a subsequent prediction can accumulate throughout the CT scan. The improved performance of iterative over direct prediction suggests that the cumulative segmentation error through iterative prediction is smaller than the error of direct prediction when compared over the same inter-slice distance. For direct prediction far from the initial input slice there may be large differences between the target and input image slices. If these differences are too large, the model will fail to relate the two slices based on context and thus hinder accurate segmentation. In practice, if the error of either prediction approach becomes too large at a certain distance from the initial contoured slice, a user can adjust the predicted contour. These predicted contours in return can then be used to make a prediction.

The difference between iterative and direct prediction was less marked for structures included in the training set. This may be because the model still learns some features specific to the structures in the training set. These learnt features may account for the improved performance of iterative prediction.

Future work needs to study the impact of the proposed interactive contouring tool on inter- and intra-observer variability. In this chapter, the middle axial slice of a structure was used as a user input. In practice, users may choose different slices to contour for a first user input or provide a slightly different initial contour. This will affect the segmentation output of the model. Future work needs to investigate, the impact of different initial user inputs on the segmentation performance. But even more importantly, it needs to be investigated if such differences in user behaviour have an impact on the final output of the structure after clinicians have made edits to the segmentation outputs. The deep learning tool may bias clinicians towards a certain segmentation outcome based on the data included in training. This will be discussed in Chapter 4, as part of the evaluation by clinicians.

3.3.3.3 Propagation using image registration

The contextual deep learning approach achieved similar or better segmentation results than the registration approach for structures included in the training set and similar segmentation results for unseen structures. The better performance

of the deep learning model over registration on structures included in training can be attributed to the model learning structure-specific features. These features help with refining the segmentation even if the contextual input alone may not be sufficient to make an adequate prediction. In the registration approach all structures are unseen and it can only rely on the contextual input.

Registration performed better using the iterative prediction than the direct prediction. This signifies that for registration the accumulated small errors at each prediction step are less significant than the larger errors from a big step. This may indicate that the registration approach is not robust to large changes between target and moving image. Iterative prediction was also the better performing prediction approach for Model ABCDEF.

Generally, it is difficult to draw a fair comparison between those two approaches, as each approach has a lot of space for individual fine-tuning. Registration and deep learning approach are both limited by the chosen registration approach or network architecture, respectively, and depend on the chosen hyper-parameters. The visualization of the model output (Figure 3.14) further demonstrated that the deep learning model is heavily dependent on the training set provided. But a deep learning model with a larger and more diverse training set has the potential for highly-accurate contour prediction.

3.3.4 Conclusion

This section investigated the generalizability of contextual information by a deep learning model using three input channels. The first channel was used for the image to be segmented. The second and third were used to provide context from an image slice and corresponding manual delineation of the structure to be contoured. The experiments demonstrated that a model trained using multiple distinct structures can accurately make predictions for structures within the training set and achieve performance similar to or higher than fully automatic approaches, but more importantly it could generalize and predict structures excluded from the training set based on the provided context alone.

Two different prediction procedures (direct and iterative prediction) were introduced and compared. For unseen structures, it was found that the error propagated using iterative prediction remains small when compared to the direct prediction approach.

The proposed contextual deep learning approach allows prior context to be incorporated into deep learning contouring to facilitate interactive contouring of CT imaging. Such an approach may enable faster de-novo contouring in clinical practice, where manual contouring is required and there is insufficient training data to train a specific model.

4

Evaluation by Clinicians

Contents

4.1 Introduction 71
4.2 Preliminary Evaluation 72
4.2.1 Materials and Methods 73
4.2.2 Results 77
4.2.3 Discussion 80
4.3 Extended Evaluation 84
4.3.1 Materials and Methods 84
4.3.2 Results 87
4.3.3 Discussion 92
4.4 Conclusion 97

4.1 Introduction

The clinical applicability of automatic and interactive contouring software has been evaluated in the literature for many body regions, such as abdomen [149], head and neck [150], [151], breast [152] or lung [9]. For many of these, only the segmentation accuracy of the method is evaluated, without direct clinical implications such as impact on contouring time [9]. However, clinical applicability is difficult to demonstrate by only evaluating model accuracy by common evaluation measures, as geometric similarity does not necessarily predict impact on the clinical

workflow [153].

For example, auto-contouring methods can reduce contouring workload by providing an initial contour. The question remains how much additional time is needed to edit the automatically generated contours compared to manual contouring. For an auto-contouring tool to be viable clinically, the editing of automatically generated contours must save time compared to standard contouring tools - but this is often not investigated [153]. While some evaluation measures, such as Added Path Length correlate better with contouring time than others, such as Dice Similarity Coefficient, they are only a surrogate for real clinical impact.

In Chapter 3, the contextual deep learning method was shown to perform well according to common geometric evaluation measures. In this chapter, this is evaluated with respect to its clinical impact. To evaluate the clinical usefulness of the contouring method, the proposed method is compared to manual contouring directly when used by clinicians to contour lung tumours. The goal of this evaluation by clinicians is to investigate the impact on target contouring when using the interactive deep-learning assisted contouring tool, focusing on contouring time, contouring workflow, and inter- and intra-observer variability. First, a preliminary evaluation was conducted for an initial assessment of the tool, as well as to receive feedback from participating clinicians, as user experience and interface design can have a great impact on contouring time. Then, more clinicians took part in an extended evaluation. The results of these evaluations are presented in this chapter.

4.2 Preliminary Evaluation

Initially presented at ESTRO 2021 [3].

A Graphical User Interface (GUI) for interactive contouring was developed in close collaboration with clinicians to suit their needs when contouring. It incorporated the contextual deep learning model, as well as typical manual contouring tools. A screenshot of the application used is shown in Figure 4.1. The preliminary clinical validation was conducted to improve the GUI and optimize the model as needed, before involving a larger number of clinicians.

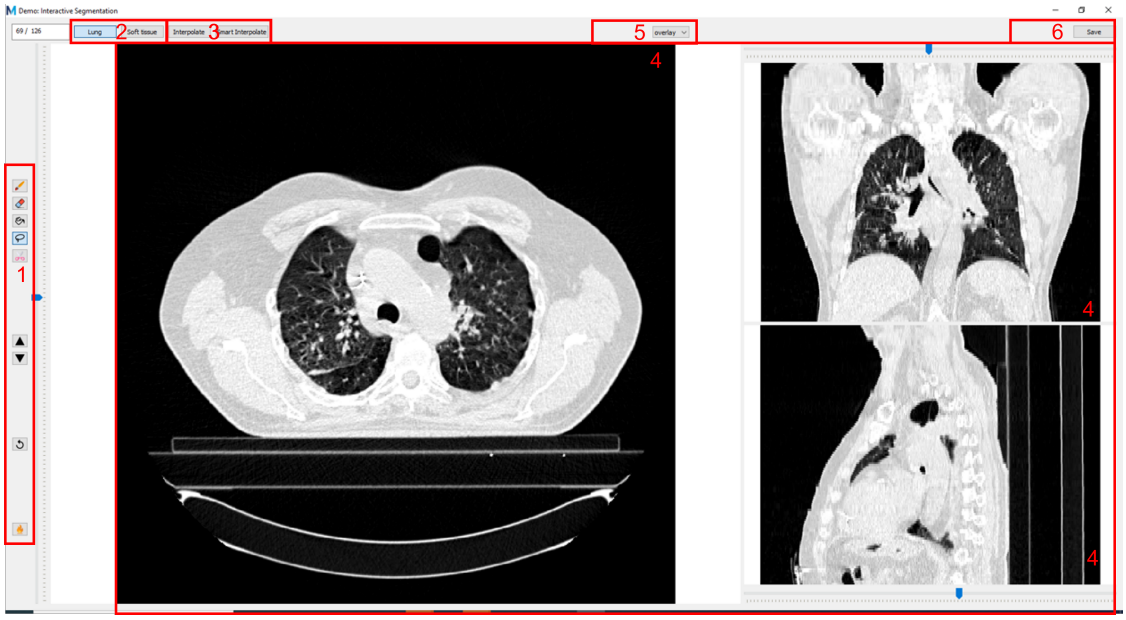


Figure 4.1: Screenshot of GUI for the preliminary evaluation by clinicians. The GUI contains: (1) Basic tools for editing: brush, eraser, lasso, scissor, delete one slice (undo) and delete all (fire). Navigation tool: up/down. (2) Two window level settings: Lung and soft tissue. (3) Two interpolation settings: Linear interpolation and Smart interpolation (Smart interpolation corresponds to predictions using the deep learning tool), (4) Axial, coronal and sagittal view of the CT. (5) Different contour visualization settings. (6) Save button.

4.2.1 Materials and Methods

4.2.1.1 Standard contouring workflow

Before designing the user interface of the GUI, four clinicians were asked for the workflow they normally follow when contouring lung tumour cases, as well as which tools are required to do the task. The standard contouring workflow as described by clinicians is shown in Figure 4.2. First, clinicians familiarize themselves with the scan and contour the image slices. Then, the contours are revised and contours are edited as seen fit. When familiarizing and contouring, different clinicians reported different preferred workflows. Most clinicians stated, they contoured every other slice (or every n-th, depending on the structure) and use an interpolate function to receive an initial contour for intermediate slices. One clinician noted, that they typically do not use any interpolate function when contouring tumours, but use it only for OAR contouring. A final revision of the scan is then made to ensure consistent contouring.

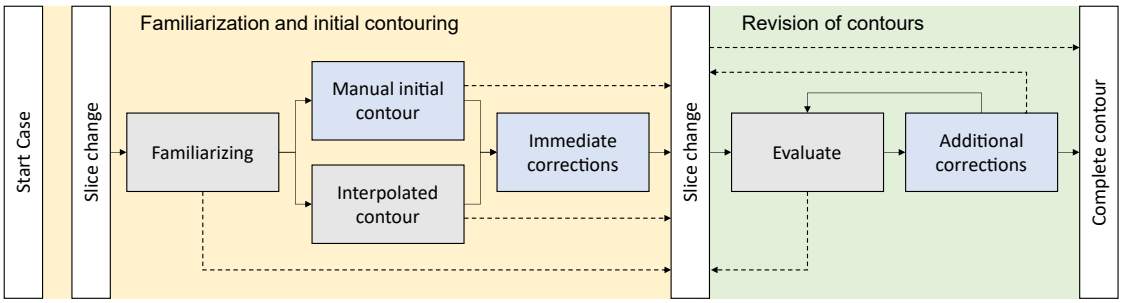


Figure 4.2: Standard contouring workflow as described by clinicians. After opening the case, clinicians familiarize themselves with the scan, and begin contouring individual slices. Some clinicians contour every n-th slice and then use an interpolate function to receive an interpolated contour for intermediate slices. The resulting contours are then corrected as necessary. At the end, the contours are evaluated and additional corrections made. This process can be iterative until an adequate contour is achieved.

The clinicians were based at different hospitals and therefore the contouring software they used differed. For the experiment, they said they expected standard editing tools (brush, eraser) and displaying options (window level, navigation, zoom).

Additionally, depending on the treatment plan and hospital site, the image modalities available and contouring guidelines vary. For example, for Lung SABR (Stereotactic Ablative Radiotherapy) treatment, various OARs need contouring and tumour motion is tracked using 4DCT. This requires contouring of the tumour volumes at multiple time points. Initially, it was planned to reproduce a SABR contouring workflow of multiple cases. However, it was not feasible to have clinicians contour 10 such cases due to the time spent on a full planning scan and the clinicians' large day-to-day workloads. Instead in the experiment, each clinician only contours the lung GTV per case.

4.2.1.2 GUI design

The GUI only contained the tools strictly necessary for performing the segmentation task and to fit the workflow as described in Figure 4.2. The aim was to make it easy to use for clinicians regardless of what contouring software they were accustomed to.

The GUI contained axial, sagittal and coronal views of the patient, as well as a set of buttons to change between contouring tools and different display and window level settings. The user could choose between a *lung* (-1350HU to +150HU) and *soft*

tissue (-125HU to +225HU) window level. The participating clinicians reported that these are the only two window levels they needed to contour the primary tumour. Only these two options were provided to simplify the GUI. By scrolling the mouse wheel while hovering over the corresponding view, the scan could be reviewed. It was possible to zoom in and out of the various view planes by pressing the Z key and moving the mouse. The *overlay* drop-down menu, allowed the user to choose between different sets of contour visualizations: contour outline only, fill only, outline and fill.

The editing tools included in the GUI were located at the left hand side: a brush and eraser tool, as well as a lasso and scissor tool. These could be used to create and edit the contour on the current displayed axial slice. The current mask could be deleted using the *undo* button. The masks on all slices could be deleted using the *fire* button.

Above the CT views, there was an *Interpolate* and *Smart Interpolate* option. During the experiment, the clinicians needed to contour each scan using either manual (*Interpolate*) or deep learning functionality (*Smart Interpolate*). The respective other button was disabled. The *Interpolate* required there to be at least two contours present in the scan. Linear interpolation was used to fill in the masks between the contoured slices. The *Smart Interpolate* button only required one contour in the scan. It used the method described in Chapter 3 to predict the contours for the remaining slices without a contour [2]. If more than one slice was contoured, the closest of two inputs was used to make a prediction on a given slice that had not yet been contoured. The up and down arrow buttons could be used to go along the axial slices. If the deep learning tool was enabled, and the next slice was not yet contoured, the slice was contoured by the deep learning tool.

The current state of the GUI was saved using the *Save* button. User interaction was automatically tracked by the GUI. The drawing and editing of contours by dragging the cursor was recorded as *active* contouring time. All other behaviour was logged as *observation* time.

4.2.1.3 Data

Ten non-small cell lung cancer (NSCLC) cases from the LUNG1 dataset [133] were selected to represent a wide range of tumour types (size, location, shape etc.). The original dataset contains manual delineation of the GTV by radiation oncologists for a total of 422 patients. The preprocessing of the data was the same as in Chapter 3.

4.2.1.4 Experimental details

The time spent contouring the GTV was compared for a manual and deep learning assisted contouring tool applied to CT imaging datasets of 10 NSCLC, each contoured by three clinicians. The manual and deep learning tools shared the same user interface and a typical basic tool set for contouring (see Figure 4.1). After contouring a case using one method, the same case was contoured at least one week later using the other method. The one week break was introduced so that clinicians are less likely to recognize a case they have already contoured which could impact the contouring behaviour and impact contouring time. The order in which manual and deep learning tools were used was randomized per case and per clinician to mitigate the impact of order on the overall timing statistics. The experiment design is shown in Figure 4.3.

The tool was made accessible to the clinicians through a virtual machine with a NVIDIA Tesla M60-GPU (8GB), that hosted the GUI and data on Microsoft Azure. After login on to the platform and opening the GUI, the selected case would open with the views set at the tumour, and indicating the primary tumour to contour via a previously provided contour with the original dataset. The indicated contour disappeared when starting contouring.

User feedback on the experimental design and GUI was collected to make improvements for the extended study.

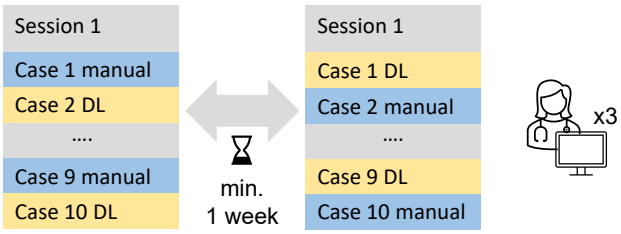


Figure 4.3: Experimental set-up for preliminary evaluation by clinicians. Three clinicians contour 10 cases in one session, alternating between using deep learning tools and manual tools only. After a minimum of 1 week, the clinicians contour the same cases, but using the other contouring tool for each case.

4.2.2 Results

4.2.2.1 Impact on contouring time

Use of the interactive deep learning tool led to a significant decrease in mean active contouring time of 33% relative to the standard manual segmentation, as shown in Figure 4.4. For two clinicians active contouring time was reduced by nearly 50%, whereas for the third clinician the decrease was 22%. The mean decrease in observation time was 26%. The two clinicians for which active contouring time was nearly halved showed a larger reduction in observation time (32% and 41%) than the other clinician (14%). The reduction in contouring time was significant for both active contouring time and observation time ($p<0.01$, one-sided paired Wilcoxon).

Observation time made up nearly 80% of interaction time independent of the contouring approach. Active and observation time appear to be correlated to each other, as shown in Figure 4.5.

The total contouring time varied greatly for each tumour volume. On average total time spent contouring per case was reduced from 13 minutes to 9 minutes when using the deep learning tool. Yet, for some of the cases contouring time was longer using the deep learning contouring tool.

4.2.2.2 Impact on contouring workflow

The contouring workflow and time spent contouring per slice for an example case are shown in Figure 4.6. When contouring, the clinician scrolled up and down, while editing the contours as they progressed through the scan (Figure 4.6a, c).

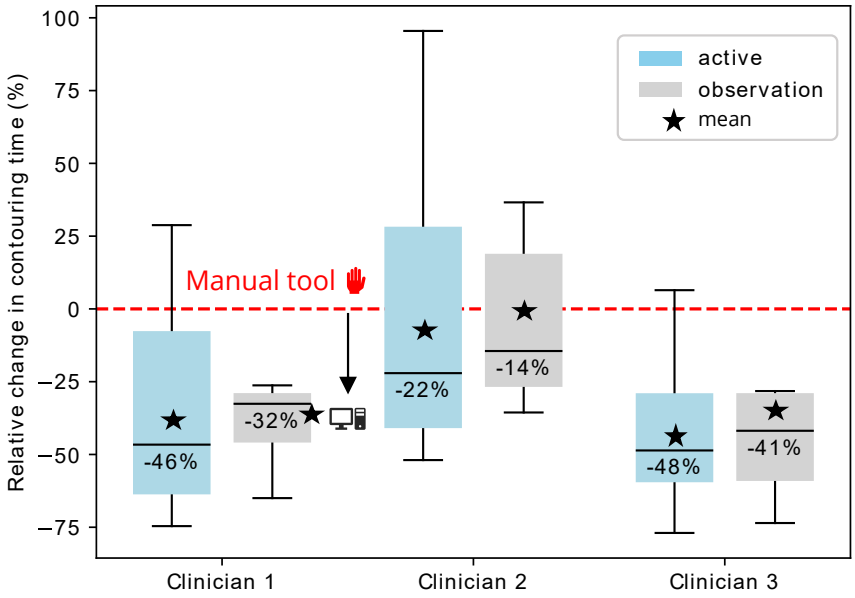


Figure 4.4: Relative change in active contouring time and observation time when using the deep learning tool compared to manual segmentation. The boxplot shows the relative change in contouring time for each clinician. The mean relative change in contouring time is indicated by a star icon.

When using the manual tool, approximately every second slice was contoured (Figure 4.6a). Then, the linear interpolation tool was used to fill the gaps and subsequently make edits where necessary. Despite only employing simplistic linear interpolation, no edits were made for four of the slices produced through interpolation (Figure 4.6b).

When using the deep learning assisted tool, one slice was contoured and the remaining slices were automatically predicted. The clinician subsequently revised the scan and made edits where necessary (Figure 4.6c). After using the deep learning propagation tool, most propagated contours required editing, and only 6 required no edits (see Figure 4.6d). Using the manual tools, the active contouring time per slice was more than 10 s for many slices, while for the deep learning tool only the initial slice took more than 10 s and for most other slices the interaction time was less than 5 s.

For both contouring approaches, approximately 80% of the slices were edited by the user. For each case, the clinician scrolls through the full scan for a final review.

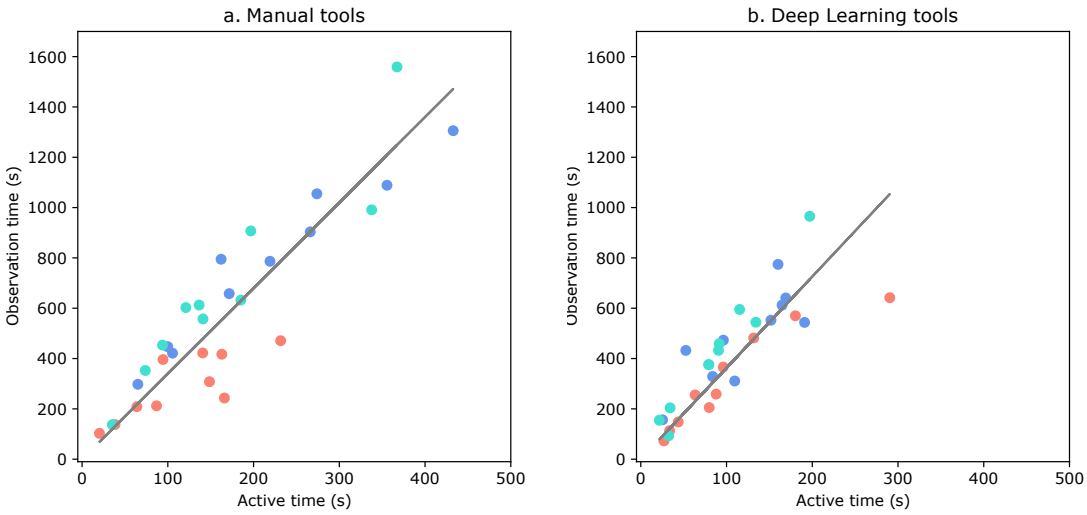


Figure 4.5: Correlation of active and observation time for (a) using manual tools only and (b) using the deep learning contouring tool for the preliminary evaluation. Each data point corresponds to a single case contoured by one clinician. Each color corresponds to a different clinician.

4.2.2.3 User feedback

All three participating clinicians stated they found the deep learning helpful when contouring and were pleased with the results of the tool when given just one user input.

Two participants noted, that in the absence of a Positron Emission Tomography (PET) scan, they struggled to identify the extent of the tumour, particularly, at the boundary region to soft tissue. But this would have been a bigger issue if they had to identify lymph node metastases or other metastases.

The way clinicians interacted with the tool varied by clinician. Two clinicians preferred to contour one slice, use the smart interpolate button and then manually edit the predicted contours; or to contour every other slice and then use the smart interpolate button once, when multiple inputs are given. One clinician reported that they intended to use the smart interpolation tool more than once, after making some edits to some of the interpolated slices. They expected that the tool would then update whatever has not yet been edited. In the smart interpolation implementation for this experiment, the tool does not override any slice that already contains a

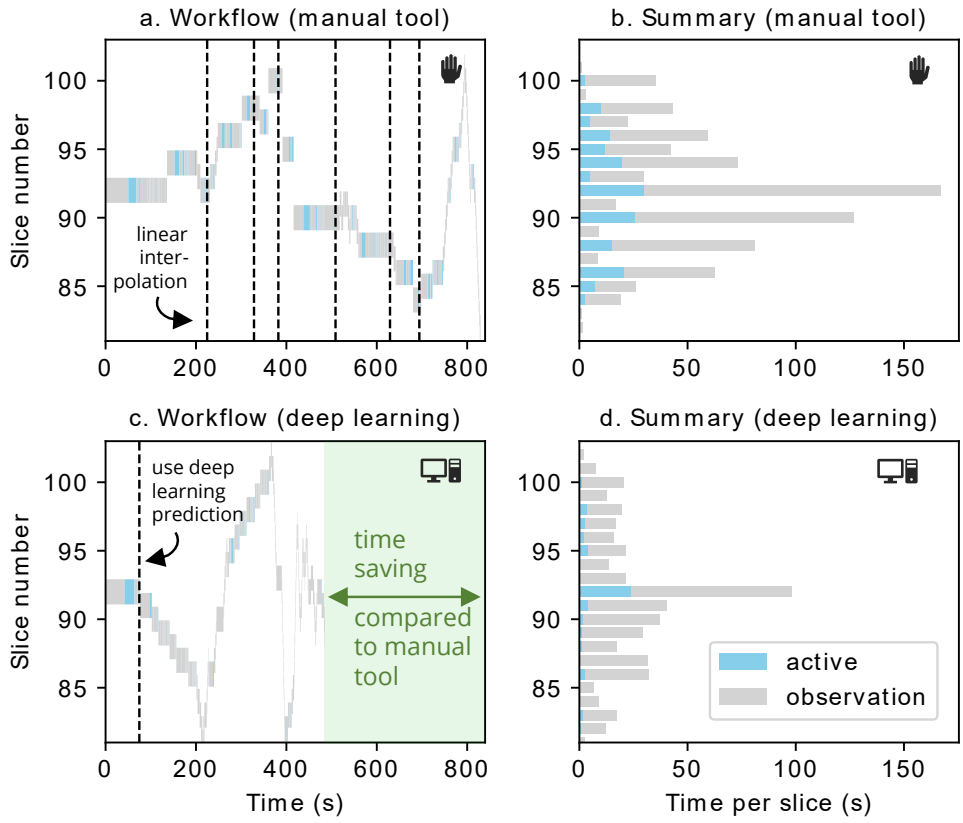


Figure 4.6: Example of user interaction times for one NSCLC case using a manual (a,b) and deep-learning assisted (c, d) contouring tool. Interaction times are shown as a workflow (a, c) and an aggregated slice-by-slice summary (b, d). The vertical lines in (a) and (c) indicate when the prediction function of the deep learning tool and the linear interpolation of the manual tool were used, respectively.

contour. Therefore, if the tool already predicted most slices and unless the user deleted inadequate slices and manually edited some slices, the tool would not make a new prediction, as the propagation algorithm only tries to predict slices that do not yet contain a contour.

4.2.3 Discussion

4.2.3.1 Impact on contouring time

The deep learning assisted contouring tool decreased active contouring time and observation time by 33% and 26% respectively. The change in active contouring time directly reflects the impact of the deep learning tool, in terms of the number and magnitude of edits that were necessary to correct the predicted contours. While

80% of the slices required editing for both the manual and deep learning tool, the mean decrease of active contouring time by 33% shows that the edits per slice may not be less frequent but fewer edits are needed per slice.

The decrease in observation time can be attributed to multiple factors, such as the time it takes to identify the tumour, navigation of the GUI and to do a final review of a scan. However, most of these tasks are required when contouring using the manual tools only as well. Furthermore, the order in which clinicians use the manual or deep learning tool was randomized and a week passed between contouring the same case again. As a result, the time to familiarize with the scan should not be affected by the contouring methods. Active contouring time is only tracked when the mouse is pressed and dragged. The time between clicks or change tools is counted towards observation time, which occurs more often if more and/or larger edits are necessary. This causes the correlation in change of active and observation time and as a result, decreasing observation time when active time is reduced.

Additionally, a large difference in time saving was observed between the different observers. While contouring time nearly halved for two clinicians, it only decreased by 25% for the other clinician. The reason for this is unclear. The variance between observers shows that the interactive tool (or any tool provided to clinicians) may support individual users to a different extent. Either, because the users are already very efficient using the tools at hand, or because the new tools do not fit their workflow. Furthermore, some people may pick up how to use new tools faster.

4.2.3.2 Impact on the contouring workflow

The contouring workflow varied greatly between the clinicians and each case they contoured. In general, when using the deep learning tool, fewer slices were contoured before using the deep learning prediction. Often only a single slice was contoured and then the smart interpolate function was used. When using manual tools only, the user inputs were close together and approximately every 3rd slice was contoured before using linear interpolation. The workflow analysis showed that 80% of the slices for both contouring methods required editing. So even linear interpolation

was accepted for entire slices for 20% of the slices, even though it does not take into account any image specific information.

It was also found that observation time far outweighs active contouring time. As pointed out above, changes in active time and observation time correlate, because observation time partially consists of actions such as changing contouring tools. If more edits are necessary, contouring tools may need to be changed more often. Therefore, active time and observation time increase. If the segmentation performance of the deep learning tool improves further, active and observation time may be impacted.

A large reduction in observation time may also be achieved by optimizing the user's interaction with the user interface. These may be simple adjustments such as the placement of the various buttons, or replacing buttons with keyboard shortcuts. Additionally, practice with the software might help the clinician navigate the tool faster. In the GUI used for the experiment, the mouse travel path to the different buttons in the GUI cannot be reduced much more, as they are already placed directly beside the axial viewing panel. For the extended study, keyboard shortcuts were introduced for switching the lung window (pressing the W key), execute interpolation and smart interpolate (pressing the I key), as well as for toggling the PET overlay (pressing the P key) which was included in the extended study. However, it was found that these shortcuts were not used much. This may change for contouring software used on a daily basis however. As in this case, clinicians only contoured a few cases and they would have had to look up keyboard shortcuts and therefore saw only a marginal benefit.

Observation time additionally is made up of decision-making time. Time that the clinicians spends reviewing the case and evaluating different areas of the scan. This time may be reduced, if the tool is able to highlight areas of uncertainty or which slices require further user interaction, as will be discussed in Chapter 6.

4.2.3.3 User feedback

The clinicians reported that a PET scan is needed to simulate a more realistic contouring experience. The absence of PET scans may affect the observation time, as clinicians may spend more time to evaluate the tumour location. As only the primary tumour was contoured and the PET is more necessary for contouring metastases. This will be assessed in the extended evaluation by clinicians.

Furthermore, there was confusion on how the propagation works when multiple inputs were given, specifically when the smart interpolation tool is used more than once. User expectation for one clinician was that the tool would update slices that were not yet contoured after a few edits had been made. However, this was not implemented in the GUI. To be able to indicate to clinicians what will be updated, an overlay is needed that indicates which slices have been interacted with, and those that have not and therefore are updated if the smart interpolation tool is used again. This may make the GUI interface more complex and may result in additional interactions with the user interface to allow use of use this functionality.

The user feedback will be incorporated in the extended study which is discussed in the following section.

4.3 Extended Evaluation

In preparation as a manuscript to be submitted as Ref. [6].
Following the preliminary evaluation by clinicians, the participants' feedback was used to update the GUI and a PET/CT dataset was chosen for the extended experiment. In addition to analysing the impact and contouring time and workflow, the extended study investigates the inter- and intra-observer variability for the nine participants.

4.3.1 Materials and Methods

4.3.1.1 Changes in GUI design

The revised GUI incorporated three main changes compared to the preliminary study.

First, an option was included to overlay the CT with a registered PET scan while pressing the control key. This was requested by the clinicians, as they would typically have this available when contouring lung tumours. It allows them to be more certain about the location of the tumour. Using PET/CT data better resembles current clinical practice.

Second, the updated GUI highlights which contours have and haven't been edited or confirmed by the clinician. This makes it easier for clinicians to focus on areas they have not yet addressed. A screenshot of the GUI showing this feature is shown in Figure 4.7. If a contour was drawn on a slice, this contour was displayed as fully blue on the axial, sagittal and coronal view. If smart or linear interpolation was used, then the computer predicted slices were displayed as blue, with a red outline on the axial view, and as a red slice on the sagittal and coronal views. Once a clinician edited (via any contouring tool) or approved (lock icon) the contour on a slice, the contour was displayed as blue. All contours could be marked as accepted using the green tick icon, or rejected using the red cross icon. The display of edited and non-edited slices was introduced to help clinicians visualize their progress and which areas still needed their attention. Additionally, it helped them decide whether a further user input may help improve the overall segmentation if the interpolation tools were subsequently used.

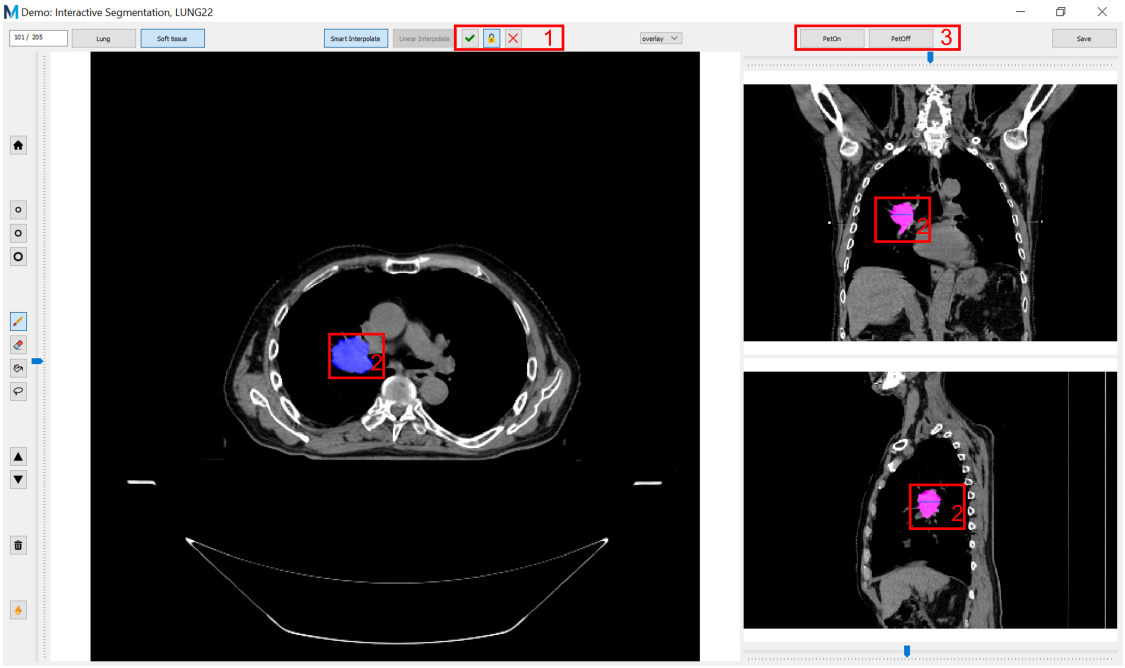


Figure 4.7: Screenshot of GUI for the evaluation by clinicians used in the extended evaluation. Changes to the GUI in Figure 4.1: (1) Buttons are available to accept (tick) or reject (red cross) all contours, or just accept/reject the current slice. (2) In the views, a contour surrounded by red on the axial slice, or a red highlighted slice on the sagittal and coronal views indicate that this slice has not yet been edited or accepted by the clinician. Here, an accepted or edited slice is shown on the axial view and therefore is missing the red border. (3) The *PetOn* and *PetOff* button overlay the registered PET scan to the CT.

Third, the previous evaluation was not sufficient to determine the contribution of different actions to total observation time. Previously, all clicks and mouse dragging while pressed were tracked. However, the mouse motion was not tracked while it was not clicked. In this extended evaluation, all mouse movement was recorded.

4.3.1.2 Data

50 cases of patients treated for NSCLC were received from LeCentre Hospitalier Universitaire de Liege. The data included radiotherapy planning CT with manual delineations of OAR and tumour, PET whole-body computerised tomography attenuation correction (WB-CTAC), and body low dose CT with manual delineations of OAR and tumour. Of the 50 cases, 7 cases were selected by an independent clinician to showcase a broad variety of tumour types (size, location, shape, etc.).

Additionally, three cases were selected from a large scale CT and PET/CT dataset for lung cancer diagnosis from The Cancer Imaging Archive (TCIA) [154]. These were specifically chosen to include larger tumours, as the Liege dataset had only smaller tumours. Unlike the data provided from Liege, the TCIA data did not include planning CT, but diagnostic CT scans.

The preprocessing of the data was the same as in Chapter 3.

4.3.1.3 Local evaluation of contouring differences

A consensus contour was created from all user created contours to compare the contours of the individual clinicians to each other. For this, the STAPLE (Simultaneous truth and performance level estimation) algorithm [155] was used, which was developed for the validation of image segmentation methods. The algorithm considers a collection of segmentations and computes a probabilistic estimate of the true segmentation and a measure of the performance level represented by each segmentation using an expectation-maximization algorithm [155].

To identify and visualize the anatomical regions of adjustments between clinicians, each user contour was aligned to the consensus contour and the amount of deviation from the consensus contour was quantified. Statistics of deviation were reported as the median and 10th to 90th percentile range of difference between individual clinicians and the consensus contour.

4.3.1.4 Experimental details

The contouring workflow, impact on contouring time and contouring consistency were compared for manual and deep learning assisted contouring tools applied to CT imaging datasets of 10 NSCLC, each contoured independently by 9 clinicians. The manual and deep learning tools shared the same user interface and a typical basic tool set for contouring, shown in Figure 4.7. When working on a case to be contoured with the deep learning tool the linear interpolation tool was disabled and vice versa. In the first session, the 10 cases were contoured by the clinicians alternating between deep learning and manual contouring. After contouring a case using one method, the same case was contoured at least one week later using the

other method. The case order was the same for all clinicians. User interaction was automatically tracked by the GUI.

Seven clinicians were experienced consultants, two trainees. There was not enough data for a statistically significant analysis to distinguish the impact depending on experience level.

The tool was made accessible to the clinicians through a virtual machine with a NVIDIA T4-GPU (16GB), that hosted the GUI and data on Google Cloud. After login on to the platform and opening the GUI, the selected case would open with the views set at the center of the patient, the clinicians were instructed to locate and contour the primary tumour. Note, in the preliminary study the case opened with the views being set at the primary tumour.

4.3.2 Results

4.3.2.1 User interaction tracking

During each contouring session, all mouse movement was tracked. A histogram of the time intervals between mouse movements is shown in Figure 4.8. For active time, these time differences were very short (almost all $< 2\,\mathrm{s}$). As expected, because the mouse button was pressed and as contours were being actively edited, the mouse location was changing. The time intervals for observation time varied much more. 9 time intervals larger than 100 s with no mouse movement were attributed to breaks or interruptions during contouring. 13 large time intervals that lasted 85 s to 100 s occurred, of which 8 occurred for one clinician in one session. In this session, the clinician reported the GUI to be non-responsive several times.

Because these large time intervals were likely due to interruption and not contouring specific tasks, they were excluded for all further analysis of contouring time. These exclusions were small compared to the total of approximately 61 hours of contouring time and more than one million interaction time intervals logged. The data were excluded blindly and based on the overall timing statistics, irrespective of manual and deep learning contouring method used.

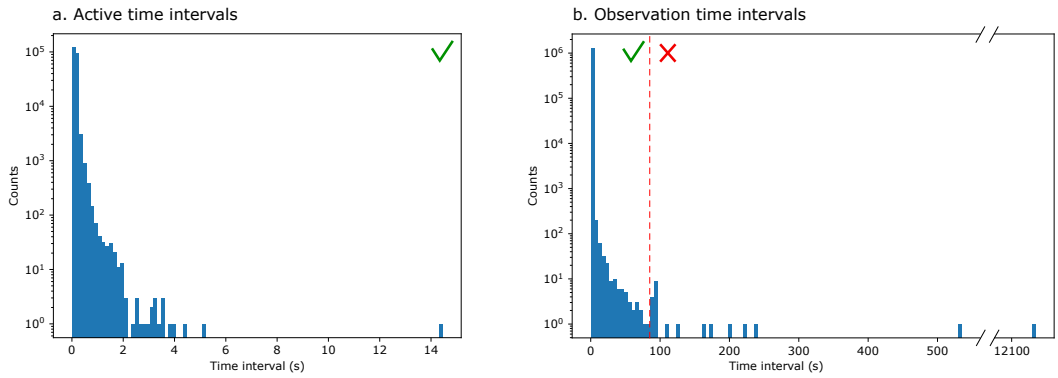


Figure 4.8: Histogram of time intervals between tracked mouse movement for (a) active and (b) observation time. Note, these do not correspond to individual interaction, but to each logged time point when mouse was moved, when mouse was pressed (active) or not pressed (observation).

The relative breakdown of active time and observation time per clinician is shown in Figure 4.9. For active time, the lasso tool was most commonly used, whereas only one clinician did not use the lasso tool at all but only brush and eraser. The scissor tool was not used by any clinician. Across all clinicians, the lasso tool made up 51%, the brush 32% and the eraser 17% of active time.

Observation time can not be directly broken down into different user activities. 42% of all observation time was below 1 s across all users, whereas intervals of more than 50 s made up 19%. Note, that 2 annotators had no time attributed to interruptions larger than 50 s.

4.3.2.2 Impact on contouring time

The interactive deep learning tool reduced mean active contouring time by 23% relative to the standard manual segmentation, whereas mean observation time did not change between the different contouring approaches across all annotators and cases. The decrease in contouring time was significant for active contouring time ($p<0.01$, one-sided paired Wilcoxon). On average the time spent contouring per case was reduced from 22 minutes to 19 minutes per case when using the deep learning tool.

The change in contouring time per clinician across the different cases is shown in Figure 4.10. The clinician with the largest median decrease in active contouring time

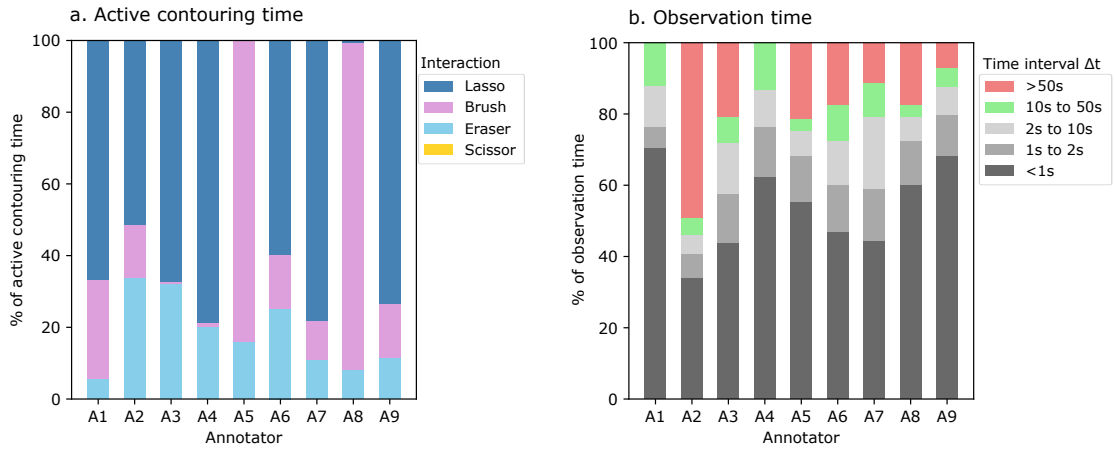


Figure 4.9: Relative composition of (a) active contouring time and (b) observation time by annotator. Active time is split into the different contouring tools used. Observation time cannot be easily categorized into different types, therefore the contouring times are grouped by time intervals between mouse movements.

was annotator A7 with a reduction of 63%. Other annotators did not experience a benefit using the deep learning contouring tool, but on average they spent a similar time contouring using either tool set.

The change in contouring time per case varied greatly as shown in Figure 4.11. All but one case, showed a reduction in active contouring time. For case C4, there was no change in active contouring time between the manual and deep learning tool. In contrast, for case C2 the deep learning tool reduced active contouring time by more than 40%. Median change in observation time by case ranged from an increase of nearly 50% to a decrease by nearly 50% compared to manual contouring.

Observation time made up 67% of contouring time when using manual tools and 74% when using deep learning contouring tools. Absolute observation time was approximately the same between both contouring approaches. The relationship between the two is shown in Figure 4.12.

Note, that the observation time excluded tracking data during which the mouse was idle for more than 85 s, as shown in Figure 4.8. Without the aid of camera tracking of the participants it is not possible to differentiate between observation of the tumour and decision-making time and the clinician taking a break.

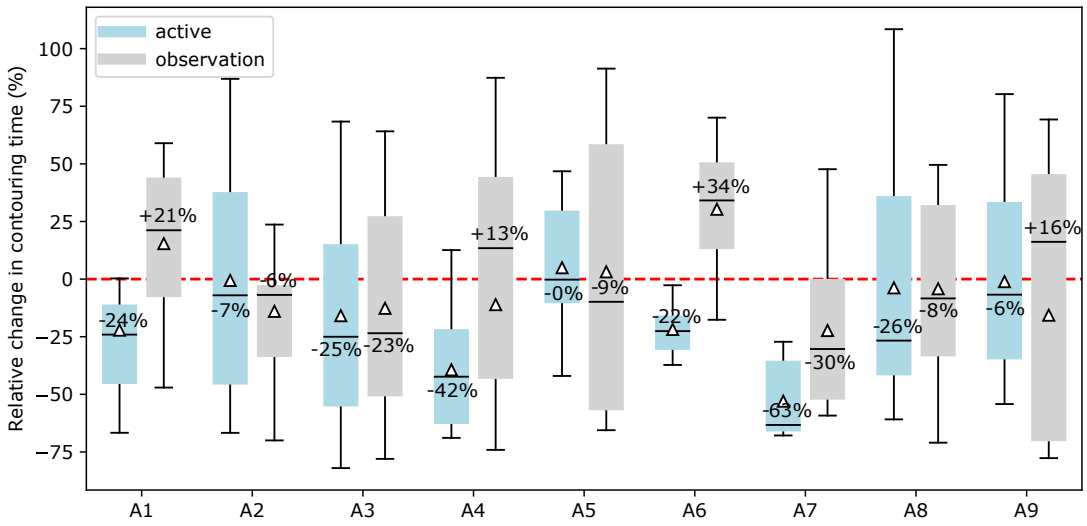


Figure 4.10: Impact on contouring time by annotator. Relative change in active contouring time and observation time when using the deep learning tool compared to manual segmentation. The boxplot lists the median relative change in contouring time for each clinician. The triangle indicates the mean relative change in contouring time.

4.3.2.3 Impact on intra- and interobserver variability

To investigate the inter-observer variability, each clinician's contour was compared to the consensus contour, shown case by case in Figure 4.13. This was done separately for the cases contoured using the deep learning or manual tool. The inter-observer variability for manual contouring tools in terms of DSC was 0.73 ± 0.10, whereas the inter-observer variability for the deep learning tool was 0.77 ± 0.08. The inter-observer variability in terms of relative APL was 0.41 ± 0.11 for the manual only tools and 0.39 ± 0.10 for the deep learning tool. The difference between manual and deep learning tools was not significant in terms of APL and DSC.

The intra-observer variability when comparing a clinician's manual to deep learning contour was 0.83 ± 0.07 across all cases, shown as a boxplot for each case in Figure 4.13 next to the inter-observer variability. The inter- and intra-observer variability were similar for most cases. A significantly lower intra-observer variability compared to inter-observer variability was observed for cases C2, C4, C6 and C8. For these cases, an independent clinician stated that the variability may be caused due to collapsed lung, which is of similar density as the tumour,

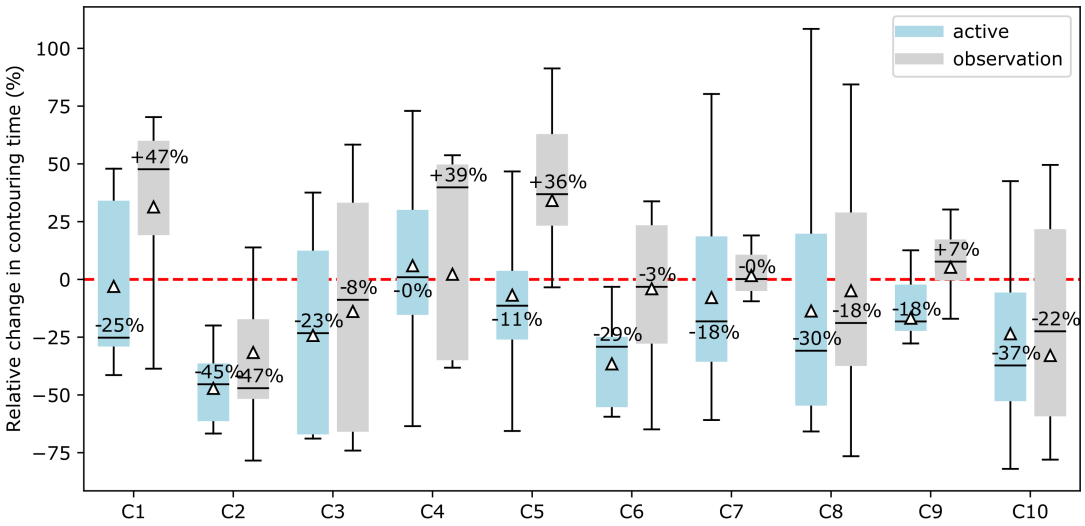


Figure 4.11: Impact on contouring time by case. Relative change in active contouring time and observation time when using the deep learning tool compared to manual segmentation. The boxplot lists the median relative change in contouring time for each case. The triangle indicates the mean relative change in contouring time.

and therefore, difficult to distinguish from each other. Further difficulties may arise where the tumour abuts the mediastinum.

The inter-observer variability overall was not significantly higher or lower for manual or deep learning tools. However, a significantly higher inter-observer variability was observed for case C4, when using manual tools only.

Additionally, a consensus contour was created for the deep learning and manually created contours. The difference between the consensus contours is illustrated Figure 4.15, which shows the distance of the deep learning consensus contour from the manually created consensus contour for two example cases. The difference between the two consensus contours were small for each case, not exceeding 3 mm for either case shown and showing little difference in most areas.

4.3.2.4 Local evaluation of inter-observer variability

Between the manually and deep learning created contours there was no significant difference (see Figure 4.15), therefore a consensus contour for all user created contours (manual and using the deep learning tool) was created.

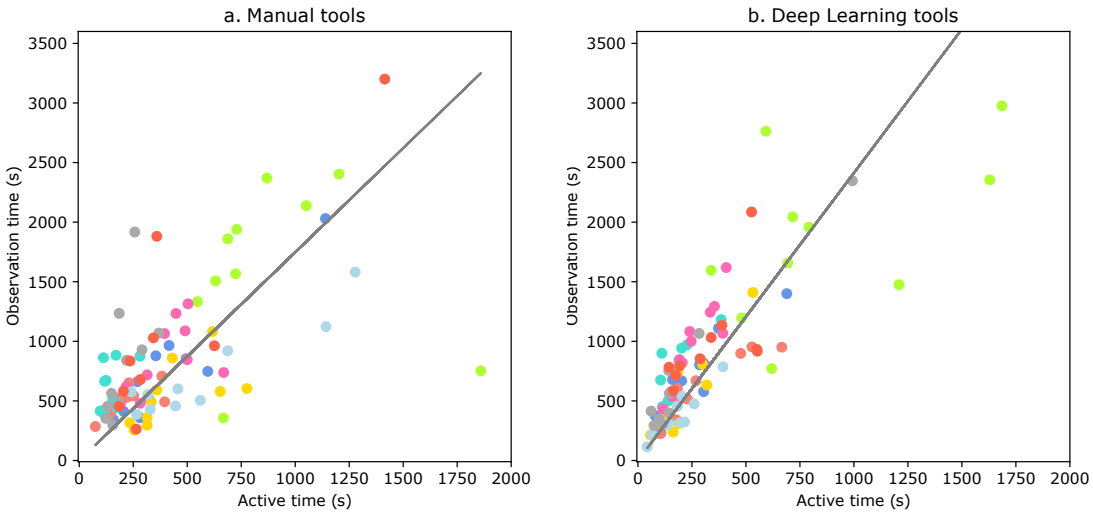


Figure 4.12: Correlation of active and observation time for (a) using manual tools only and (b) using the deep learning contouring tool for the extended evaluation. Each data point corresponds to a single case contoured by one clinician. Each color corresponds to a different clinician.

The spatial distances of the individual contours to the consensus contour are shown in Figure 4.16, illustrating the 10th to 90th percentile range of the distance between individual annotator and consensus contour for two selected tumour cases. The 10th to 90th percentile can be viewed as a measure of variability.

For example case C1, the variability was negligible for most of the tumour, except for the inferior part of the tumour where the 10th to 90th percentile range was 18 mm. For the second case C2, large variations were observed on the inferior and superior borders of the tumour with overall very little disagreement.

Spatial variation showing median, 10, 30, 70 and 90 percentiles over all users per case are shown in Figure 4.14, for each case. Most structures had a 10th to 90th percentile range of about ±10 mm, cases C4 and C5 showing the largest variation with a range of −30 mm to 40 mm for the manual tool.

4.3.3 Discussion

4.3.3.1 User interaction tracking

Active time Active contouring time consisted of many very short user interactions. The tools preferred for contouring varied greatly between the observers,

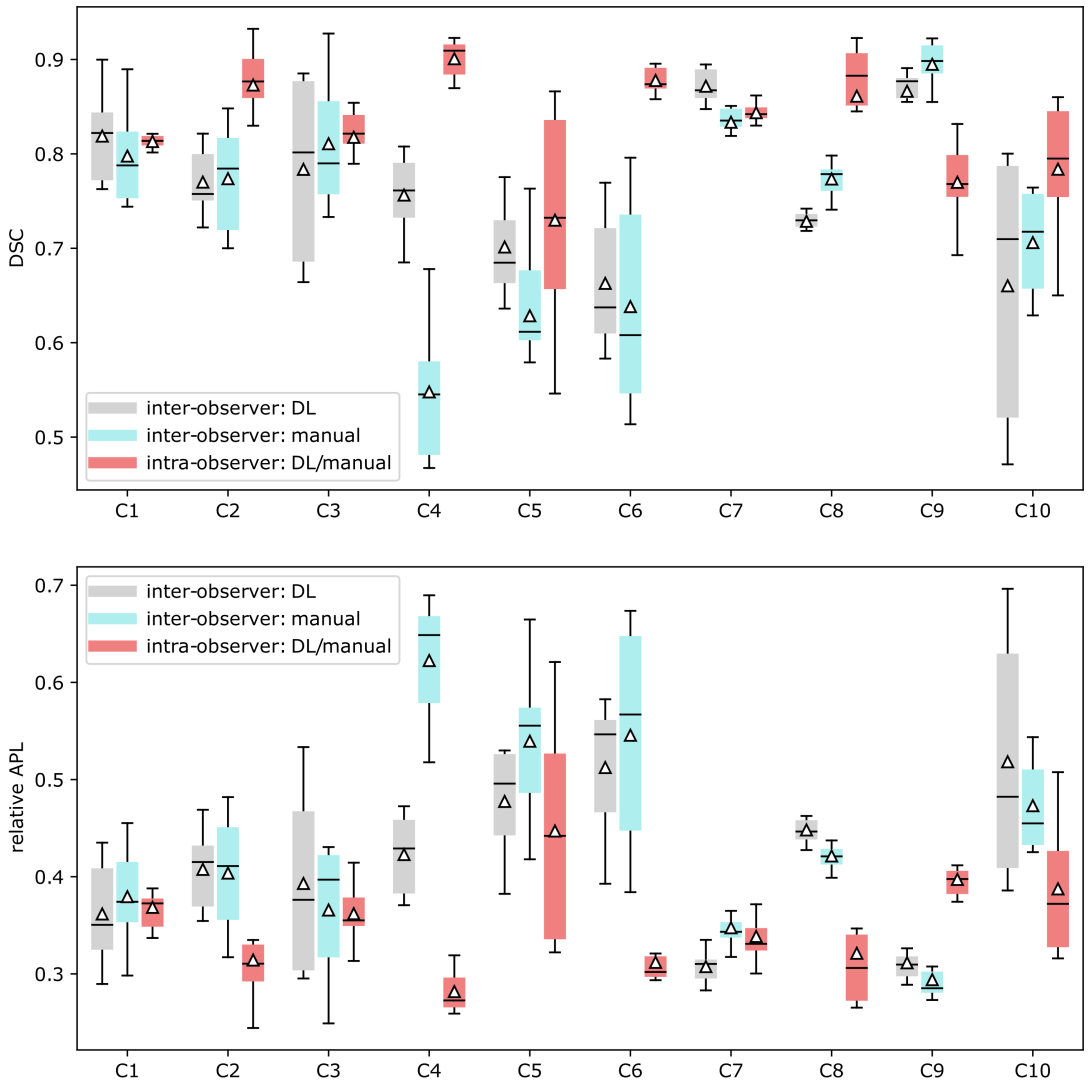


Figure 4.13: Comparison of inter-observer variability for the deep learning tool and when using manual tools only, as well as the intra-observer variability by case. The inter-observer variability compares each clinician's contour with the consensus contour, whereas the intra-observer variability compares the manual and deep learning tool segmentations by the same clinician.

demonstrating that even for such small things as tool preference (brush vs. lasso) user preference and therefore usability of the tool for the users may vary. The Scissor tool, which could be used in place of an eraser, was not used at all by any clinician. Feedback from the users indicated that they preferred the lasso tool for making large changes and particularly for manually contouring an initial slice, whereas they used the brush and eraser tools to make smaller edits. The scissor tool

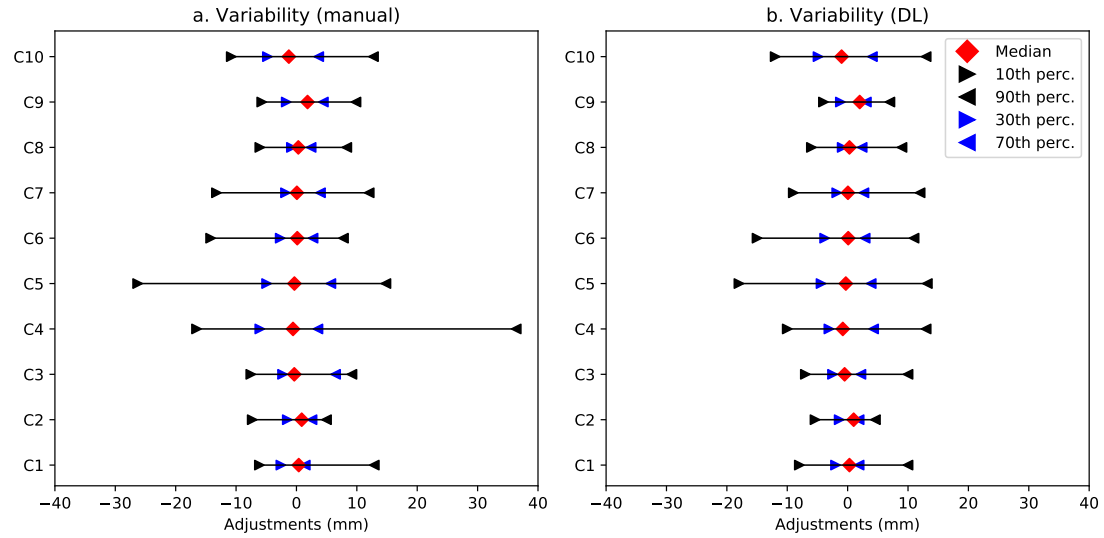


Figure 4.14: Spatial difference to consensus contour showing median, 10, 30, 70 and 90 percentiles for all annotators per case.

would be used mostly, when a large bit of structure is cut out, but as the tumour shapes were largely convex and without holes in the structure, it was not required.

Observation time Observation time with time intervals below 1 s made up 42% of contouring time. These small time intervals mean that the mouse was constantly moving and therefore these time intervals were likely due to interactions with the user interface, such as for example changing contouring tools, navigating through the scan or zooming. Larger observation time intervals may be attributed to studying the case and decision-making time. But time intervals larger than 10 s and particularly larger than 50 s likely occurred due to small distractions, as for example, checking messages, emails or brief interactions with colleagues.

A more detailed breakdown would require visual tracking of the clinicians via a camera or eye tracking, in order to understand what they are focusing on.

4.3.3.2 Impact on contouring time

As with the preliminary study, the extended study revealed that active contouring time was reduced. But the magnitude of the reduction varied greatly by annotator and case.

Active time and observation time were again correlated. This is attributed to the observation time that occurs whenever contouring is necessary - including, for example, changing contouring tools. More edits result in larger active time which increase the time spent interacting with the user interface (observation time).

Observation time made up a large fraction of the total contouring effort. Observation time was about twice as long (70% of total time spent contouring) as the active time. For the preliminary study observation time was nearly 4 times as long (80% of total time spent contouring). Additionally, during the preliminary study the reduction in active contouring time correlated with a reduction in observation time. This was not observed for the extended study in which observation time did not change between both studies but varied strongly between each annotator and case.

The longer observation time compared to active time in the preliminary study may have been due to the lack of PET images. The availability of PET scans in the extended study may have helped in speeding up the decision-making process. The added functionality of the smart interpolation tool that highlights where user interaction has occurred in the past, could also have influenced the difference in observation time and active time between the two studies.

The total time spent contouring was longer for the extended study. This may be due to the size and difficulty of the tumour cases. Additionally, the slice spacing for the preliminary study was 3 mm whereas it was 2 mm for the extended evaluation, which results in a larger contouring load.

The experimental changes from the preliminary to extended study, may also have contributed to the differences in the results of the two studies. In the preliminary study, when starting the software for each case the axial, sagittal and coronal view were centred at the tumour to be contoured and until interaction with the suer interface started, the ground truth for these slices were shown. On the other hand, for the extended study, this was not the case. Instead, the clinicians had to identify and navigate to the tumour themselves."

4.3.3.3 Impact on intra- and inter-observer variability

The inter-observer variability overall was not significantly different between the manual and deep learning tools. However, a significantly higher inter-observer variability was observed for case C4, when using manual tools only. Furthermore, it was found that the consensus contour created from the deep learning tool and the consensus contour from the manual tool were very similar with only very small variance between the two. Yet, when comparing the local difference in contouring (see Figure 4.14), while having similar median and 30th to 70th percentile variance, the 10th to 90th percentile of adjustments was much larger for the manual tool than for the deep learning tool. This suggests, that while for the largest part of the tumour, clinicians agree with each other, on some local areas they do not. These were the areas of high variation in both manual and deep learning contoured cases. But this variation was higher in manually contoured cases. This indicates that the deep learning tools guides the user in their contouring, resulting in smaller variation. However, it does not influence what they contour on average, as the consensus contours are still similar.

Previous studies have shown that automatic and interactive contouring tools can decrease intra- and inter-observer variability compared to manual contouring [14]–[17], [156]. More consistent and repeatable contours can improve the treatment given to a patient. The contextual deep learning model that is investigated here shows no significant impact on the inter-observer variability compared to manual contouring, when evaluated on the full structure. However, if the local variance is considered, it does reduce the inter-observer variability at places where clinicians disagree most.

Generally, decreasing variability between observers is desirable in clinical practice, in order to be able to standardize treatment. However, the studies reporting on reducing inter- and intra-observer variability mostly refer to OAR contouring. OARs are more similar between patients and it is far easier for a deep learning model to be able to learn the rules on how to segment these structures. Disagreement between clinicians may largely be based on variations in contouring practices and guidelines. For tumours, on the other hand, it is much more difficult to find a

representative training set and in clinical practice the cases may be different to those used to train the deep learning tool. This may be the reason why the tool did not show a decrease in variability.

This investigation also shows that care needs to be taken over how inter-observer variability is evaluated. Geometric measures that relate to the full structure such as 3D DSC are not very sensitive to local variations between observers, while a local evaluation of difference to a consensus contour revealed a difference of variability between manual and deep learning for selected areas of a structure.

4.3.3.4 Local evaluation of inter-observer variability

It was not possible to analyse the manual edits to a predicted contour across all observers, as the observers followed different contouring approaches, and therefore the initial prediction of the interactive tool varied. E.g. Some clinicians only edited one slice, used smart interpolation, and then reviewed and edited the predictions using manual tools; others contoured multiple slices before using the smart interpolate functionality and used manual tools to edit the prediction. Therefore what they needed to edit changed and it is unclear where edits were necessary given the original position. The instruction for the clinician was to use the tool as they see fit, as this would give insights in their contouring behaviour and would help understand what level of interactivity would be most beneficial to the clinician. Unfortunately, this makes it impossible to analyse comprehensively where edits are needed following propagation using the deep learning tool, as has for example been done in the literature for edits required to automatically generated contours [157].

4.4 Conclusion

The contextual deep learning contouring approach was evaluated and shown to decrease active contouring time when used to delineate lung cancer GTVs. Although observation time is only indirectly affected by the contouring method, it was positively impacted by the deep learning tool in the preliminary study. In contrast,

observation time was found to remain constant compared to manual contouring tools in the extended study.

Observation time was found to make up the majority of the contouring time in both the preliminary and extended evaluation. The mouse tracking during observation time showed that for nearly half of the observation time, the mouse is constantly moving ($< 1\,\mathrm{s}$ time interval), indicating this is time spent navigating the GUI. Regardless of the tools used for contouring or just for correcting automatically generated contours, interaction with the user interface occupies a lot of time. Focusing on the user interface design may have a large impact on the time spent contouring.

An analysis of local variability between contours demonstrated that the deep learning tool helps reduce inter-observer variability at locations where clinicians tend to disagree, while the consensus contour does not change significantly depending on the contouring approach. Thus, the tool helps make contours consistent in critical areas, while still segmenting what the user intends to contour.

Such an interactive tool could be integrated into the clinical workflow to assist clinicians in contouring tasks and to improve contouring efficiency, as well as consistency.

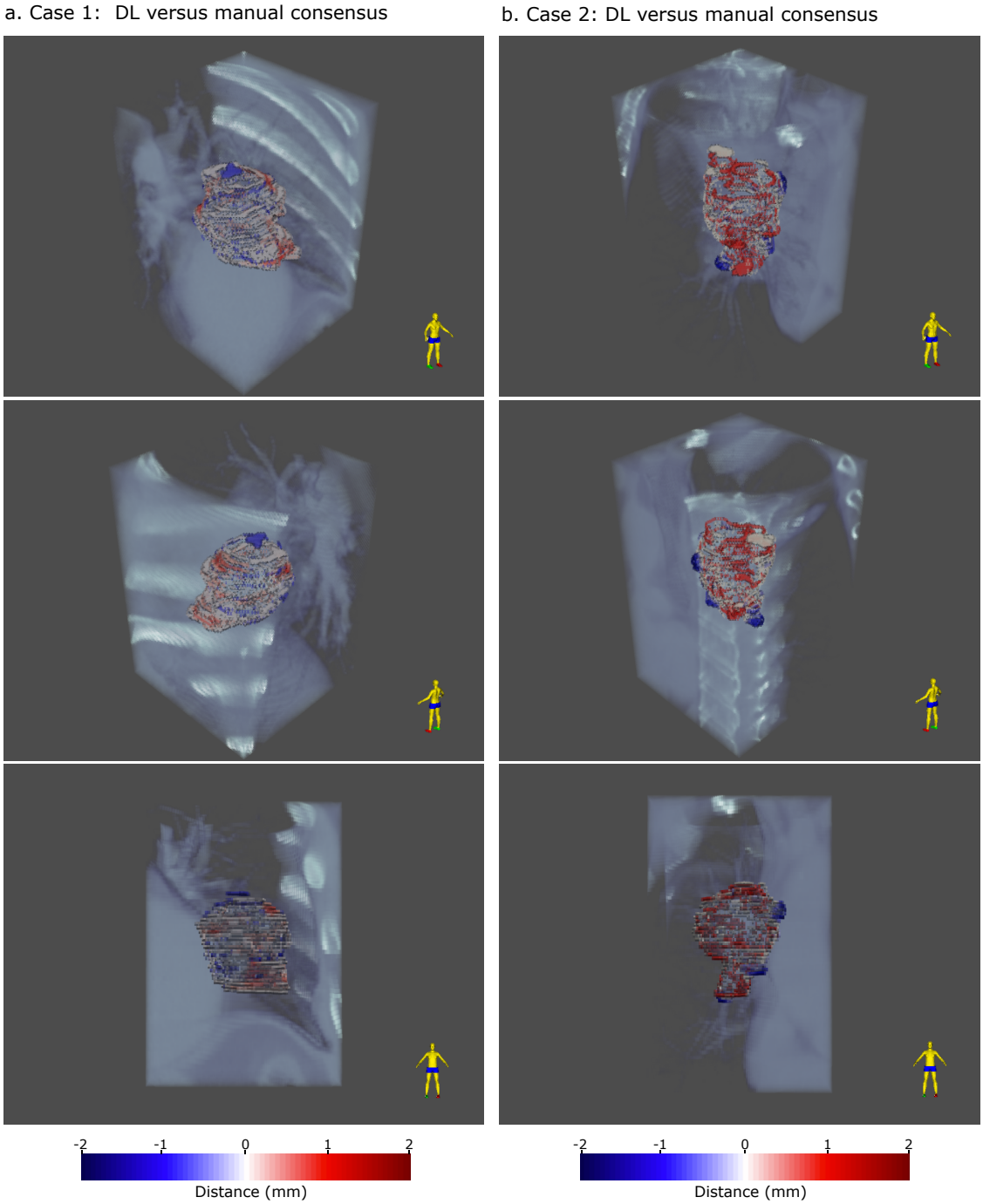


Figure 4.15: Contouring variation between DL and manual consensus contour - shown as the distance of the DL consensus contour projected onto the manual consensus shape for two example cases C1 and C2. A positive distance corresponds to an enlargement of the DL contour outwards with respect to the manual consensus contour.

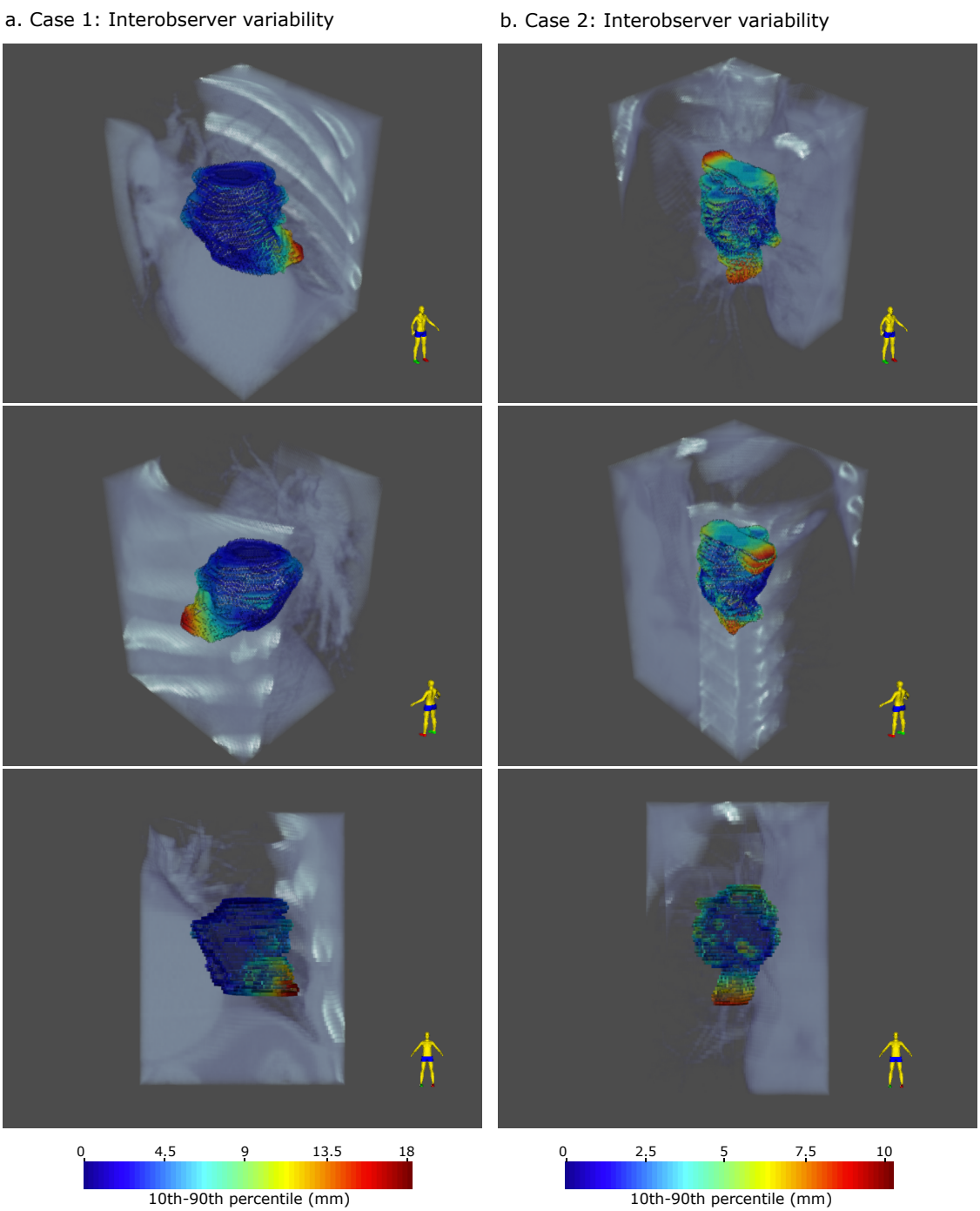


Figure 4.16: Spatial variation showing 10th to 90th percentile range projected on the consensus shape for two example cases. The 10th to 90th percentile range represents the variability in disagreement between clinicians.

5

Task-specific Fine-tuning for Contextual Deep Learning

Contents

5.1 Introduction 102
5.2 Materials and Methods 103
5.2.1 Fine-tuning of deep learning models 103
5.2.2 Data 103
5.2.3 Experimental details 104
5.2.4 Evaluation measures 106
5.3 Results 107
5.3.1 Interactive contouring through fine-tuning of contextual deep learning model 107
5.3.2 Impact of inter-observer variability on model predictions 109
5.3.3 Fully automatic segmentation given small dataset 110
5.4 Discussion 110
5.4.1 Interactive contouring through fine-tuning of contextual deep learning model 110
5.4.2 Impact of inter-observer variability on model predictions 112
5.4.3 Fully automatic segmentation given small dataset 113
5.5 Conclusion 114

5.1 Introduction

Partially presented at AAPM 2020 [4]. In preparation as a manuscript to be submitted as Ref. [7].

Radiation therapy is one of the most common and effective treatments for cancer. However, radiation-induced injuries, including lung fibrosis, affect approximately half of the treated patients [158]. In lung cancer patients, lung fibrosis is the most prevalent injury [159]. It consists of inflammation and subsequent scarring of lung tissue due to radiation exposure during treatment. This can result in breathing problems, and even in fatal respiratory insufficiency [160]. Prevention and treatment of radiation-induced injuries can greatly improve the quality of life of patients. CT imaging is often used to diagnose and monitor lung fibrosis [161]. Correct segmentation and distinction from other entities such as local tumor recurrences is important but fibrosis segmentation is not currently a regular part of the clinical decision-making process.

A few studies have performed and examined automated lung fibrosis segmentations [162]–[164]. At first glance, automatic deep learning tools may seem an ideal solution. Through CNNs, many segmentation tasks in the medical field have been successfully automated and user interactions for editing have been greatly reduced, as well as reducing inter-observer variability and reducing time spent contouring. But deep learning models struggle with the task of lung fibrosis segmentation due to the high variability in shape, size, structure and similarity to other tissue, making it difficult to create a representative dataset. In addition, fully automatic deep learning tools rely on a large amount of annotated training data, which for fibrosis are very limited.

Instead, semi-automatic tools that produce 3D segmentations given minimal user interaction can help accelerate annotation of lung fibrosis or other structures, which are difficult to contour or lack sufficient training data for fully automatic deep learning models. In the case of lung fibrosis segmentation, there is no publicly available large dataset in the literature. Transfer learning (introduced in detail in Section 2.3.2) has previously been successfully employed to train models on small

training sets. Transfer learning has been applied to learn deep representations that are then re-purposed for other tasks [165] or few-shot learning [166]. For segmentation applications, the most common approach is to train a network on a source domain and task and then fine-tune the model parameters on the target domain or task [51].

This chapter investigates whether the contextual deep learning approach proposed in Chapter 3 can help with the task of lung fibrosis segmentation. The impact of fine-tuning the contextual deep learning method to specific segmentation tasks will be investigated. Additionally, the contextual deep learning model will be compared to fine-tuning of automatic models on the same dataset.

5.2 Materials and Methods

5.2.1 Fine-tuning of deep learning models

For fine-tuning, the learning rate was halved (from 1×10^{-5} to 5×10^{-6}) and all model layers were fine-tuned. This has previously been referred to as deep fine-tuning [167]. Other studies (for example [168]) have reported fine-tuning only selected layers and freezing the remaining weights. In this application, partial fine-tuning did not result in an improvement. Five-fold Cross Validation (CV) was used to make better use of the limited lung fibrosis data available. For each of the five models trained as part of the five-fold CV, three folds were used as a training set, one as a validation set to be used for early stopping, and one for testing. All testing folds together represented the full dataset. Extensive data augmentation was used to avoid overfitting. In addition to translation (±10 mm), rotation (±10 degrees) and scaling ($\pm5\%$), random intensity augmentation was used to increase the training set diversity. For the contextual deep learning model, the remaining training strategies, such as slice sampling during training, were followed as in Chapter 3.

5.2.2 Data

211 non-small cell lung cancer (NSCLC) patients that were treated with stereotactic body radiation therapy (SBRT) between 2005-2019 were retrospectively identified.

Median follow-up time was 18 months (range, 1-96). 38 (18%) patients were diagnosed with fibrosis based on follow-up CTs. These patients were included in this study. Median age at time of treatment was 70 years (range, 47-85). Median follow-up time was 28 months (range, 3-96). Median time between SBRT and fibrosis diagnosis was 9.5 months (range, 1-87).

Segmentations of lung fibrosis were provided by 2 clinicians. Based on the radiologists' follow-up description, manual fibrosis segmentation was performed using 3D slicer [169] (version 4.11, www.slicer.org). As far as assessable, the tumour residuum was explicitly excluded from the segmentation to merely include fibrotic lung tissue.

The preprocessing of the data was the same as in Chapter 3.

5.2.3 Experimental details

5.2.3.1 Contextual and automatic deep learning model

In this chapter, a set of automatic and contextual deep learning models were trained and evaluated on lung fibrosis segmentation. An overview of the models trained is shown in Figure 5.1.

Contextual deep learning First, the contextual deep learning model was evaluated on the lung fibrosis dataset after having been trained on 19 structures, as described in Chapter 3. It was assessed whether the contextual deep learning model could generalize to this structure. Then, using the pretrained weights, the model was fine-tuned to determine the impact of task-specific fine-tuning for the contextual deep learning model. Moreover, the contextual deep learning model was trained from random weights using the lung fibrosis dataset to investigate how the pre-trained weights affect performance. If starting with random weights an initial learning rate of 1×10^{-5} was chosen.

Automatic deep learning Second, an automatic deep learning model was trained, to investigate if the interactive deep learning model provides benefits compared to a fully automatic solution. The network architecture was identical to the contextual deep learning model except for the model inputs. Instead of the image slice to be segmented and the contextual inputs, the image slice to be contoured was the only input. The automatic model was evaluated for lung fibrosis, when fine-tuned using pre-existing weights and if trained from random weights. The pre-trained model was created by training a model for heart segmentation, tumour segmentation and as a multi-task model on the LUNG1-dataset [133]. If starting with random weights an initial learning rate of 1×10^{-5} was chosen.

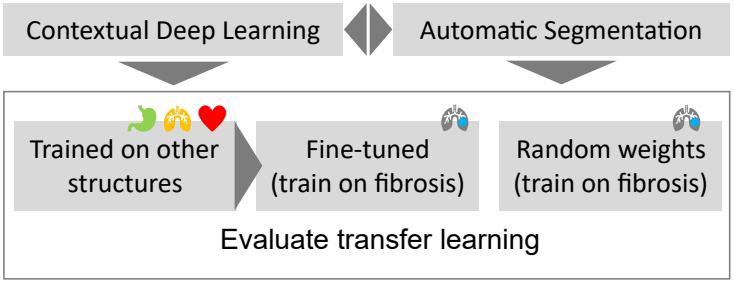


Figure 5.1: Overview of the experiments in this chapter. Transfer learning was evaluated when applied to the contextual deep learning model, as well as to an automatic deep learning segmentation method. For each approach, a model was trained including different structures (for the contextual deep learning model this is the model described in Chapter 3). This model was then used as a pretrained model to fine-tune using the lung fibrosis data. Additionally, a model was trained from random weights as a reference.

5.2.3.2 Single and multiple inputs

The evaluation by clinicians of the contouring tool (Chapter 4) has shown that multiple user inputs are often needed to reach a clinically acceptable contour. Therefore, the effect of multiple user inputs on segmentation performance was evaluated. Clinical workflow showed that clinicians often set additional inputs to the model at different slice distances. This varied strongly by case and user. Here, equal-distant user inputs were chosen for evaluation.

If only a single input was given, the central slice of the structure was chosen as an input, as illustrated in Figure 5.2a. The adjacent slices were predicted iteratively (for details on propagation, see Chapter 3).

Multiple user inputs were assessed given a fixed slice distance from each other, rather than comparing a set number of user inputs per fibrosis case. This was done to compare structures of different sizes fairly. If each case had the same number of user inputs, smaller structures would have a much higher frequency of user inputs. The input scenarios and propagation method are illustrated in Figure 5.2. Iterative propagation was used for prediction with multiple inputs. If a slice has not yet been segmented and has user inputs to both sides, the closer of the two was used to make a prediction.

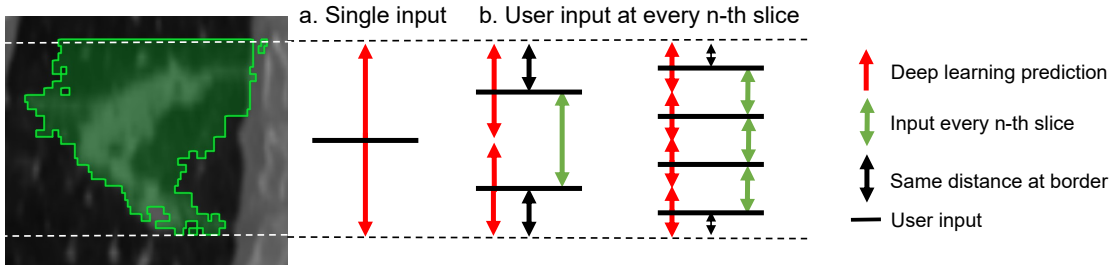


Figure 5.2: Segmentation using (a) single or (b) multiple user inputs. Inputs are provided at given positions. Based on the contours on these slices the remaining contours are being calculated.

5.2.4 Evaluation measures

The quality of the segmentation was quantified using 3D DSC [142], 95% HD [146] and relative APL [147]. As described in detail in Section 3.3.1.6. To compute the performance measures, the masks that were generated were upsampled to the original 512x512 imaging resolution using bi-linear interpolation.

One-sided paired Wilcoxon signed-rank test was used to determine significant differences ($p<0.01$) between the deep learning models investigated.

5.3 Results

5.3.1 Interactive contouring through fine-tuning of contextual deep learning model

5.3.1.1 Single input

The deep learning tool was evaluated using the original weights and after fine-tuning, shown in Figure 5.3. Fine-tuning resulted in a significant improvement in segmentation performance (p-value<0.01, paired Wilcoxon), when given a single input; increasing DSC from 0.56±0.14 to 0.66±0.18, and reducing 95% HD from (40 ± 22) mm to (31 ± 30) mm and relative APL from 0.66±0.12 to 0.58±0.20.

Figure 5.4 illustrates an example of the segmentation using a single input using the pre-trained model before and after fine-tuning. The original weights failed to accurately segment the lung fibrosis as intended by the user. Specifically, the model using original weights only segmented the high density areas of fibrosis with confluent consolidation (i.e., mass-like fibrosis), but did not include low density areas of fibrosis with patchy consolidation (i.e., modified-conventional and scar-like fibrosis patterns), respectively the air-filled volumes within those regions. After fine-tuning, the predicted segmentations also included the low-density fibrotic regions.

Please note that when the interactive contouring model was trained from scratch with random weights assigned, the model performed worse (presumably due to failing to converge). Results for the lung-fibrosis contextual deep learning model that was trained from random weights, instead of fine-tuning are shown in Figure 5.5b.

5.3.1.2 Multiple user inputs

The segmentation performance of the contextual deep learning model if given multiple inputs is shown in Figure 5.6, when contoured at every n-th slice. Additional user inputs steadily increased the segmentation performance for the model both before and after fine-tuning. For the fine-tuned model, the relative APL was reduced from 0.58±0.20 (single input) to 0.41±0.18 (every 12th), 0.38±0.17 (8th), 0.28±0.11 (4th), 0.18±0.09 (2nd). The model using original weights improved less when given multiple user inputs: 0.66±0.12 (central), 0.64±0.16 (every 12th),

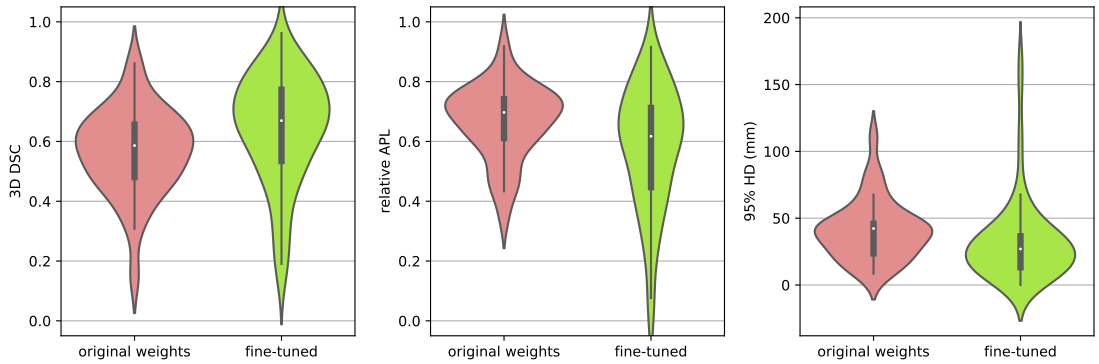


Figure 5.3: Segmentation performance when using original weights and after fine-tuning the model on the lung fibrosis dataset, shown for 3D DSC, relative APL and 95% HD.

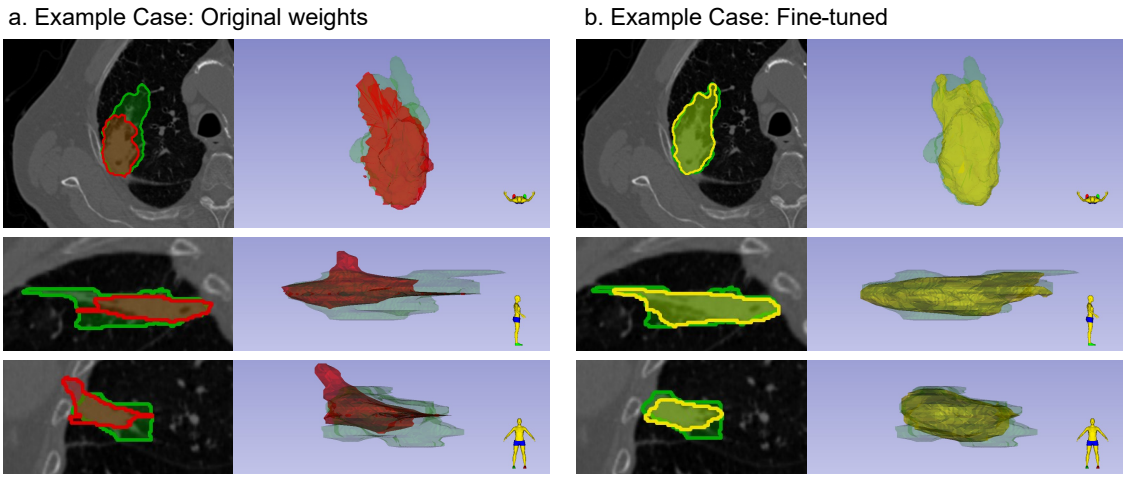


Figure 5.4: Example for segmentation using (a) original model weights in red and (b) weights after fine-tuning in yellow, after a single user input at the center of the structure. The original model often under- or overshoots the ground truth segmentation (green), while after fine-tuning the lesion is successfully segmented.

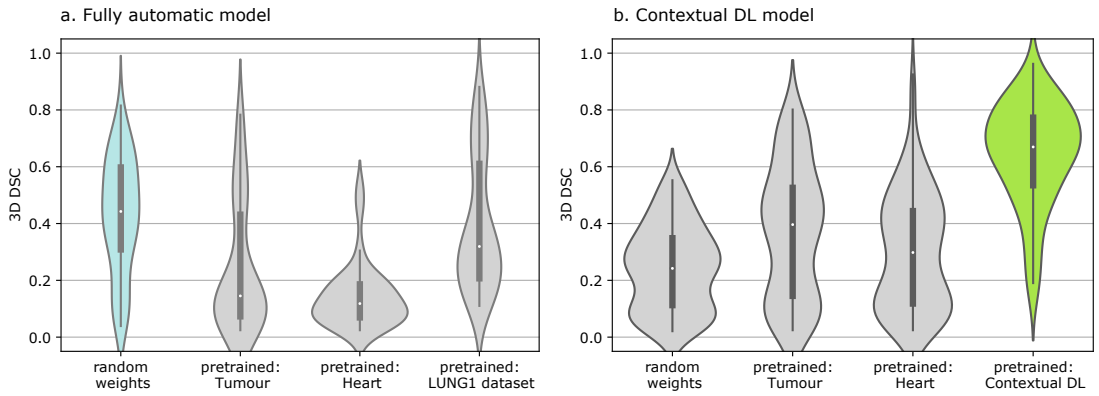


Figure 5.5: Comparison of model performance when trained from random weights or using pre-trained weights for (a) an automatic deep learning model and (b) the contextual deep learning model.

0.63±0.17 (8th), 0.57±0.25 (4th), 0.39±0.16 (2nd) for relative APL. The fine-tuned model performed statistically better than before fine-tuning at any given fixed input distance d (p-value<0.01, paired Wilcoxon).

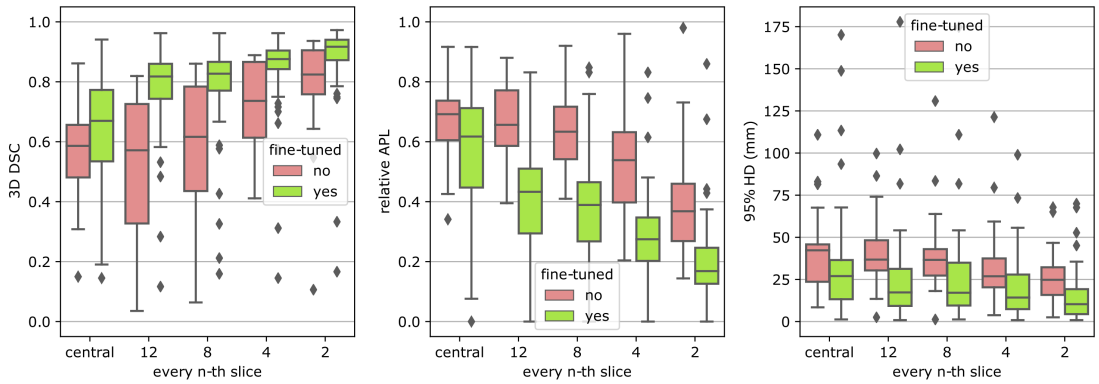


Figure 5.6: Performance comparison between fine-tuned model and the original model with multiple inputs at every n-th slice, shown for 3D DSC, relative APL and 95% HD.

5.3.2 Impact of inter-observer variability on model predictions

The dataset was annotated by two observers. The annotations showed discrepancies in terms of segmentation. The relative APL for Annotator 2 using Annotator 1 as Ground Truth (GT) was 0.86±0.09, and from Annotator 1 to Annotator 2 0.90±0.06, with a 3D DSC of 0.25±0.17, and 95% HD of (47 ± 41) mm. In contrast, the fine-tuned model given a single user input achieved a relative APL of 0.56±0.22 for Annotator 1 and 0.60±0.17 for Annotator 2, as well as a DSC of 0.64±0.19 and 0.67±0.16, and 95% HD of (38 ± 46) mm and (29 ± 25) mm, respectively. The inter-observer variability compared to the segmentation output by the DL model is shown in Figure 5.7. The annotations of both clinicians with respect to the differences were reviewed by a third clinician. The findings for the individual cases and comments on potential origins for the differences are included in Appendix D. The largest difference occurred in fibrosis for which different parts of the lung was only contoured by one of the two annotators.

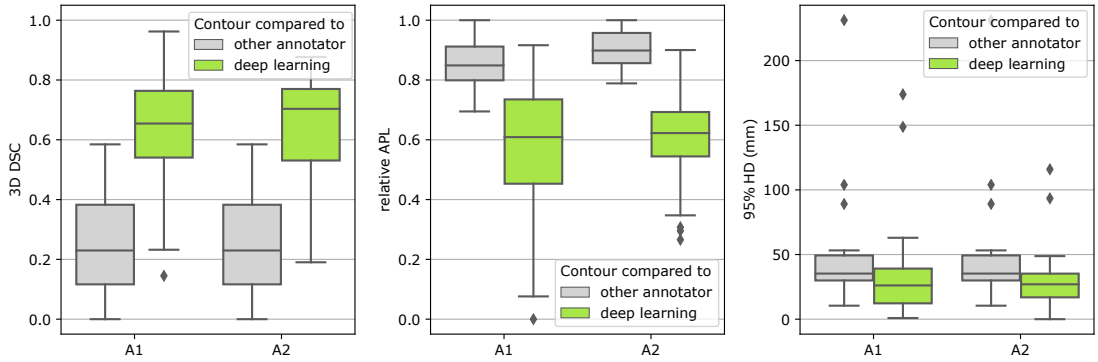


Figure 5.7: Inter-observer variability compared to model performance when provided with one user input at the central slice of the structure, shown for 3D DSC, relative APL and 95% HD. Results are shown using Annotator 1 (A1) or Annotator 2 (A2) as ground truth; and comparing the 2nd annotator's segmentation (grey) and the deep learning output given one input (green) compared to the respective ground truth.

5.3.3 Fully automatic segmentation given small dataset

In Figure 5.8, fully automatic segmentation models were compared to the interactive contouring model given one input after fine-tuning. The fully automatic model was trained either using randomly initialized weights - or after having been trained for heart segmentation or lung tumor segmentation. The automatic model trained from random weights achieved a 3D DSC of 0.42±0.20, relative APL of 0.78±0.15, and 95% HD of (38 ± 35) mm. The fully automatic model only adequately learned to segment lung fibrosis when trained from random weights. The results for fine-tuning an automatic model that was previously trained to segment hearts, or trained to segment lung tumour is shown in Figure 5.5a.

5.4 Discussion

5.4.1 Interactive contouring through fine-tuning of contextual deep learning model

The results show that given a single input, the contextual deep learning model can segment lung fibrosis. After fine-tuning, the performance of the model improved significantly.

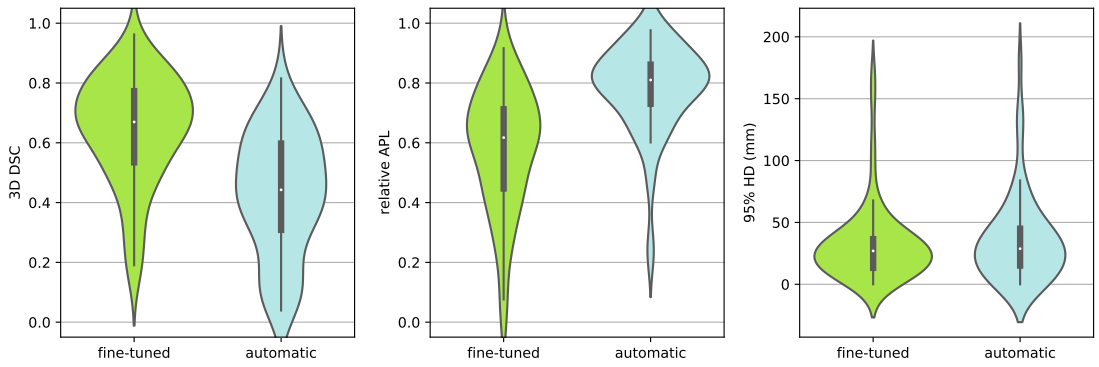


Figure 5.8: Segmentation performance of a fully automatic deep learning model (blue) trained from random weights, and the contextual deep learning model fine-tuned (green), shown for 3D DSC, relative APL and 95% HD.

Figure 5.4a shows that the pre-trained model included the high-density mass-like fibrosis pattern while excluding low-density fibrosis. This may be due to the visual similarity of low-density fibrosis to lung primary tumours which were included in the training set. Furthermore, the pre-trained model seemed unable to determine where to stop segmentation when propagating. This may be due to visually similar tumour structures that extend to the rib cage. The difference in segmentation performance before and after fine-tuning can be attributed to biases in the training set of the pre-existing model (19 structures).

If random weights were used for initialization, the contextual deep learning model failed to converge if only trained on lung fibrosis data. This also occurred when trying to train a contextual deep learning model for the oesophagus and spinal cord. For these structures, this has previously been attributed to the class imbalance and the difficulty to segment this structure in general (see Section 3.2.2.1 for details). This was solved by adjusting the label weighting for smaller structures. In the case of lung fibrosis, the model may fail to converge because it hasn't learnt to leverage context and the training data does not provide sufficient information to learn to make predictions based on context or the training dataset is too small. Training models to perform lung fibrosis segmentation from scratch is made more challenging by the difficulty of defining the structure even for human annotators resulting in a highly variable, but small, dataset which makes it difficult to train

the deep learning model. In contrast, the contextual deep learning model that was pre-trained on a variety of structures had already been trained to learn from context. Fine-tuning then enhanced its ability to specifically segment lung fibrosis.

The pre-trained model and fine-tuned model both improved with additional user inputs. Slices adjacent to a user input tend to be more similar to the input and are therefore easier to predict from context, whereas the model performs worse for more distant slices. If an input is available to both sides of a qualitative structure change, the propagation error can be mitigated.

5.4.2 Impact of inter-observer variability on model predictions

Both annotators, though given the same task, provided very different annotations for lung fibrosis. Despite the potential benefits of segmentation for follow-ups and treatment outlook etc, fibrosis is still rarely segmented in clinical practice, therefore lacking clear clinical guidelines for segmentation. Moreover, radiation-induced changes of lung tissue are complex and comprise a variety of different appearances on the CT (i.e. modified conventional pattern, mass-like pattern, scar-like pattern, and no changes), thus making its diagnosis challenging [170]. Additionally, recognizing fibrosis and its distinction from fibrosis-mimicking appearances (i.e. infection, lymphangitic carcinomatosis, lobar collapse and tumor recurrence) may be difficult [171]. This is especially the case, since margins of the residual tumour and surrounding fibrosis blend into each other, while the outer margins of fibrosis and their transition into normal tissue may not be clearly definable.

The lack of training and clinical standardization can contribute to the great discrepancies. Variable size, shape, structure, and scattered location make it difficult to segment lung fibrosis and potential miss-classification of similar tissue types may further contribute to large inter-observer variability.

The segmentation tool has shown that it can adapt to the annotations of different people and segment what the user intends to contour rather than a predefined structure in training. Thus, the tool may not result in reduced inter-observer

variability, which is generally desirable. However, in the case of lung fibrosis, this may be an advantage, as in the clinical routine the tool can adapt according to patient and treatment specific fibrosis appearances, house-specific contouring guidelines, and task-specific needs. If the tool is fine-tuned on a large set of annotations from different users, a learned average may guide future user annotations. This could reduce inter-observer variability given by the already learned training bias whilst preserving the flexibility of the tool needed for the heterogeneous nature of fibrosis appearances in different treatment cohorts.

The flexibility of the contextual deep learning model can also be beneficial when employed across different sites. When training a fully automatic deep learning model using data from just one facility, the data used for training can be assumed to be sampled from the same distribution [172]. If this model is applied to another site differences in contouring guidelines, imaging device or imaging protocol (particularly an issue for MR images), the model's performance may drop significantly if the data distribution between the two sites are different [173], [174]. Contextual deep learning can help, as it does not rely on the data it has been trained on but can make predictions of what the user wants to segment given the contextual input.

More broadly, there are still structures, such as lung fibrosis, that will remain challenging for automatic tools. For these structures, interactive versatile tools that predict what the user intends to contour are the best approach to reduce the contouring workload for clinicians. These tools can aid in enabling large training sets faster by reducing contouring time. Large training sets in return may result in successful fully automatic tools. Additionally, these interactive tools could be employed to assist in editing the errors that automatic tools still make.

5.4.3 Fully automatic segmentation given small dataset

Contextual deep learning was shown to outperform fully automatic segmentation when trained on the same dataset. The automatic deep learning tool struggled to converge and even after adjusting the label weights, did not achieve a performance similar to the fine-tuned contextual deep learning model. This is largely attributed

to the large inter-observer variability of the 2 annotators. This results in receiving different GTs for the same patient, making it very difficult for an automatic model to decide what is correct. The contextual deep learning model on the other hand can deal with different annotators, because it is not bound by just the target image slice as an input, but takes into account the corresponding user annotation.

The automatic model also performed worse if pre-trained weights and subsequent fine-tuning were used instead of random weights. This was the opposite for the contextual deep learning model. This may be attributed to the automatic model learning structure specific features. The features learnt when segmenting hearts and lung fibrosis may not be relevant for fibrosis segmentation and result in failure of the model. However, some pre-training of the network layers would be expected to be beneficial for lung fibrosis segmentation, as CNNs tend to learn high and low level features. Layers that specialize in e.g. edge detection may have been of benefit for lung fibrosis segmentation as well. It may be that upon model training with pre-trained weights, the initial predictions were far off from the ground truth and the model fell in a local minimum that did not result in convergence later on. In contrast, the contextual deep learning model does not learn structure specific features, but how the user input slice and contour relate to the target image. Therefore, when fine-tuning the contextual deep learning model, it will already output a roughly accepted contour, making the pre-trained model a better starting point than random weights. In early epochs of training the general contextual model, it even has been observed that a contextual deep learning model learns to "copy" structures from the initial user input rather than altering them. (see Chapter 3, Figure 3.5) Later on in training, the contours are refined. This is similar to what happens here, when fine-tuning the model using just one structure as an input.

5.5 Conclusion

This chapter investigated whether a contextual deep learning model can assist clinicians in lung fibrosis segmentation. Due to a lack of training data, and high inter-observer variability, fully automatic tools are not suitable for segmenting lung

fibrosis. Given two very different annotator inputs, the contextual deep learning model was found to be able to adjust to what the clinician intended to segment, even though it had not been trained on lung fibrosis before. After fine-tuning on a small fibrosis dataset the model performance improved significantly. This opens up the opportunity for continual learning when subsequently annotating more cases of the same structure. The impact of fine-tuning on the ability of the contextual deep learning model to predict various structures of interest has not been considered in this chapter. If fine-tuning does affect the ability of the contextual deep learning model to generalize to other structures, an application of the fine-tuning approach could be to use the contextual deep learning model as a starting point if new structures need to be segmented for which no automatic tools exist, and to apply fine-tuning to continually increase performance and minimize user interactions by specializing the model to a given task. Overall, the performance of the fine-tuned model with a single interactive input is comparable to the original model's performance with 4 inputs. This demonstrates the potential time savings provided by task-specific fine-tuning and continual learning for structures which still require manual segmentation. While this analysis focused on lung fibrosis, the framework presented could be applied to other structures of interest. Having the potential to assist clinicians at any given contouring task. It has the potential to reduce user interactions, while still being able to segment the structure the clinician intended to segment. Alternatively, such a tool could also be applied to edit the output of automatic solutions, propagating the edits on the segmentation output to adjacent slices and achieving faster review of medical image scans.

6

Contextual Deep Learning and Multiple User Interactions

Contents

6.1 Introduction 117
6.2 Guiding User Interaction 118
6.2.1 Materials and Methods 120
6.2.2 Results 127
6.2.3 Discussion 133
6.2.4 Conclusion 139
6.3 Optimizing Contextual Deep Learning for Multiple Inputs 141
6.3.1 Materials and Methods 141
6.3.2 Results 143
6.3.3 Discussion 146
6.3.4 Conclusion 149

6.1 Introduction

The evaluation by clinicians of the contextual deep learning tool showed that, to reach a clinically acceptable contour, edits to the initial segmentation by the deep learning tool are required. If multiple user inputs are used as starting points for propagation, fewer edits may be required, because the predictions of the deep learning model tend to be more accurate closer to a user input. However, the contextual

deep learning model has not been optimized to handle multiple user inputs.

The workflow for using the tool after the first contoured slice and the associated problems are illustrated in Figure 6.1. After contouring the first slice, the model proposes a segmentation to the user, which can be accepted or rejected. If the contours are rejected, any edits to the contours by the clinician can be used by the model to update the segmentation. The choice of the next slice to contour can be significant in determining how much the subsequent prediction of the model improves segmentation performance. For this, the model needs to anticipate where future input will be most beneficial. Once multiple inputs are given, a new prediction is made. The cycle of proposing the optimal area to edit, contouring by the user and presentation of the new segmentation will continue until a satisfactory segmentation is reached.

The first section of this chapter explores how user interaction can be directed to the most impactful areas of the structure being contoured (Question 1 in Figure 6.1). In the second section, the contextual deep learning model is analysed to see how multiple inputs can be used most effectively, by changing the propagation method or choice of input slice used to make a prediction (Question 2 in Figure 6.1).

6.2 Guiding User Interaction

With the contextual deep learning tool from Chapter 3, a user chooses the next slice to edit after the tool has predicted the segmentation for the remaining unlabelled image slices. However, this might not provide new information to the model. Instead, the contouring tool could suggest to the user which slice to contour next. This would be the slice of which the tool believes it would have the greatest impact on segmentation performance. This could guide the user's attention and reduce time spent on subsequent interactions.

The task of selecting the optimal next user input is similar to the challenges faced in the research field of *Active Learning* [117]. The goal of Active Learning is to obtain the best model, with the fewest labelled instances [118]–[120]. The problem Active Learning is trying to solve is selecting the samples that will be most

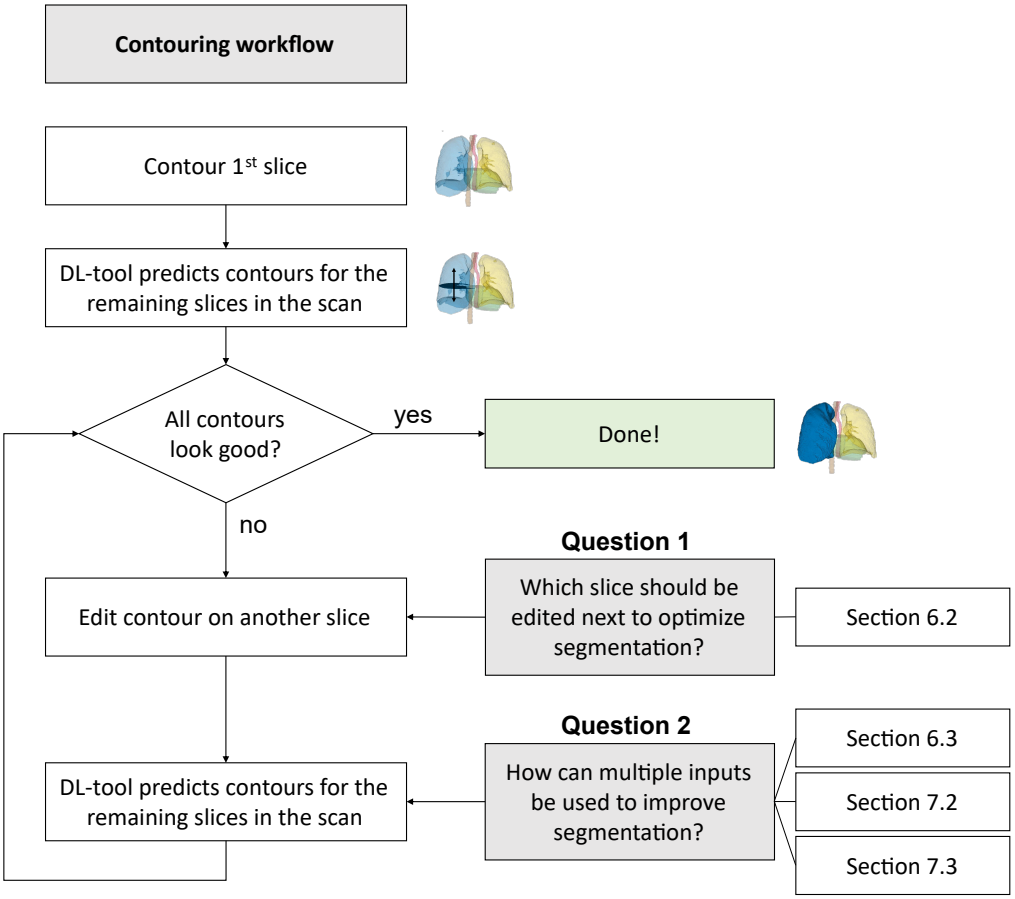


Figure 6.1: Flowchart to illustrate contouring workflow. If after contouring one slice the deep learning prediction is not good enough, another slice can be contoured to improve the segmentation. By suggesting the best next slice to contour, fewer user interactions may be needed (Question 1). Once multiple user inputs are made, then it is important to find the optimal way to leverage these multiple inputs (Question 2). To be able to determine the next slice to contour, it should also be considered how this slice is used to contour adjacent slices. The two problems depend on each other.

informative for the model. The cyclical selection process for Active Learning is illustrated in Figure 6.2. Whereas in Active Learning the training data is expanded by annotated samples to increase overall model performance, here, the next slice to contour is selected to complement the already annotated slices for the structure being contoured. For example, the next slice to contour could be selected based on high or low uncertainty attributed to the initial prediction, or sampling based on representativeness of additional input and subsequent anticipated information gain for the model compared to the original input alone.

This section investigates whether guided user interaction can help decrease

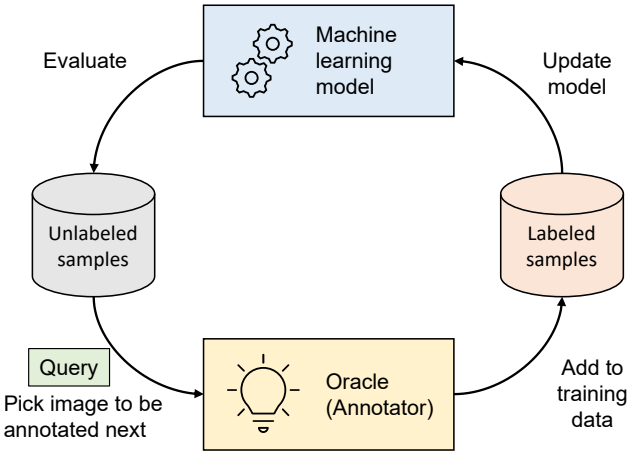


Figure 6.2: Sample selection process for Active Learning. A selection strategy is used to query a sample from unlabelled pool and present it to the oracle (here, the clinician), then the queried sample is added to the labelled dataset. Typically, in Active Learning the model is re-trained given the newly labelled training data to improve model performance with the least number of labelled samples. In the application here, the next user input is chosen to improve the segmentation performance of the model for a single case given the minimal number of user inputs.

the time spent contouring by reducing the number of user interactions needed to achieve a clinically acceptable contour.

6.2.1 Materials and Methods

The methods described in the following originate from works on Active Learning [118], [120], because the characteristics of the samples selected in the Active Learning process are the same as when looking for the next slice to contour. Three selection methods were evaluated: Model uncertainty (Entropy and MC dropout), sample representativeness (Mutual information), and an auxiliary network.

6.2.1.1 Model uncertainty

Intuitively, a slice that is associated with a high uncertainty should be reviewed and if necessary be edited by a clinician. Various methods have been proposed to assess model uncertainty: Using the softmax output [118], [126], [175], Monte-Carlo (MC) Dropout [176], Bayesian Inference [121] or using model ensembles [176].

Entropy The softmax output from the last layer of the network corresponds to the class probability for each pixel. The predictive entropy H is often used to determine the uncertainty for a given prediction, given by

$$H = -\sum_{c} p(y = c|X)\log\left(p(y = c|X)\right), \tag{6.1}$$

for the probability $p(y = c|X)$ of a sample X with classes c. The predictive entropy is maximal when all classes have equal uniform probability and is minimal, when one class has a probability of 1 and all others a probability of 0. Here, there are two classes to be considered: the structure of interest and background pixels. The entropy is calculated per pixel and results in a pixel-wise uncertainty map for the image slice.

Monte-Carlo (MC) Dropout MC Dropout introduces stochasticity in the prediction by enabling dropout regularization at inference time [177]. Each prediction results in a slightly different segmentation. An ensemble could produce a segmentation by averaging many uncertain results. By not ensembling, the difference in the prediction becomes a measure of uncertainty. The contextual deep learning model introduced in Chapter 3 did not include dropout regularization because it did not result in any model performance improvement. To use MC dropout, dropout layers (p=0.2) were added in the neural network and multiple forward passes through the network were made. Then, the variance of the predictions was calculated to measure their disagreement.

Next slice selection based on uncertainty maps The uncertainty estimating methods mentioned above produce spatial uncertainty maps, which visually represent a model's confidence for each pixel [178]. This gives the model some amount of interpretability. A clinician may also use the uncertainty map to guide their contouring. However, for the problem investigated in this section, the segmentation performance is to be determined for an entire slice, where uncertainty functions as a proxy [179]–[181], and therefore pixel-wise uncertainty is not enough.

Segmentation quality estimation has implicitly been employed in adversarial training, where performance is improved by using an auxiliary discriminator network

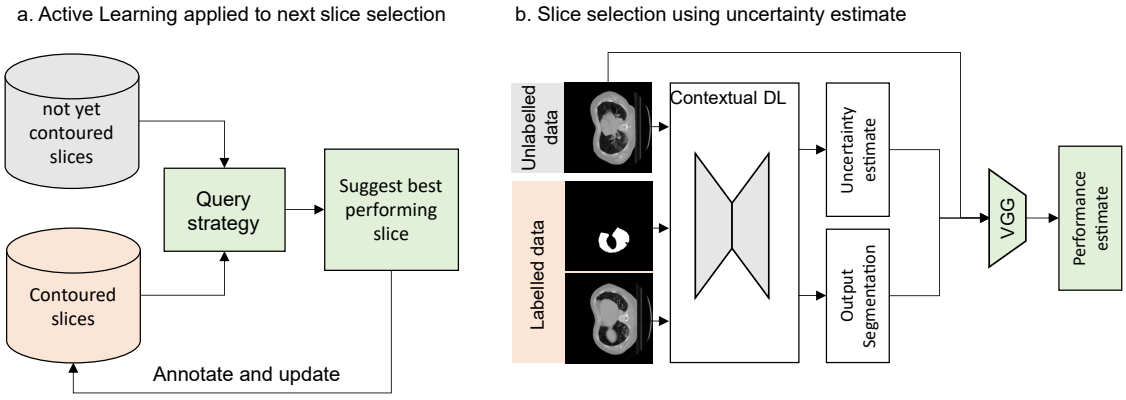


Figure 6.3: Illustration of (a) query pipeline for selecting next slice based on Active Learning and (b) the pipeline for query selection using uncertainty estimates and deep learning regression model. The segmentation prediction model takes the unlabelled image slice, uncertainty estimate and predicted segmentation as an input. By processing all slices in a scan, those slices can be ranked. The best scoring slice will be suggested to the user.

to predict the quality of the segmentation [178], [182]. Specifically, the discriminator was trained to distinguish between the ground truth segmentation and predicted segmentation. This has led to improvements in prostate cancer and brain MRI segmentation [183], [184]. A more direct approach was employed by QualityNet [185] which uses convolutional neural networks to predict segmentation quality for object segmentation.

Similarly, a segmentation quality prediction network with a VGG-backbone has been proposed to estimate segmentation performance given the input image, predicted segmentation mask and uncertainty map [178]. This approach is used here to train a network to estimate APL for a given prediction, illustrated in Figure 6.3. The segmentation quality prediction network was trained for 30 epochs with the Adam optimizer using early stopping. The slice with the highest predicted relative APL was selected as the next input.

6.2.1.2 Representativeness of inputs - Mutual information

Methods concerned only with uncertainty have the potential to focus on small difficult regions of the sample distribution [122], [186], [187]. In Active Learning, training on examples from the same area of the distribution may introduce redundancy

into the selection strategy, or may skew the model towards a particular area of the distribution [186]. Selecting the next slice to contour may bias the model towards few outlier cases which are not representative of the structure to be contoured as a whole.

Mutual information between image slices was used as a measure to determine representativeness. This assumes that slices more similar to the input slices are easier to predict for the model. Mutual information I_{MI} as a similarity measure of two images X and Y has previously been used for this purpose

$$I_{\mathrm{MI}}(X,Y) = \sum_{y\in Y}\sum_{x\in X} p(x,y)\log\left(\frac{p(x,y)}{p(x)p(y)}\right), \tag{6.2}$$

where image intensities are interpreted as probability distributions. The probability distribution $p(x,y)$ corresponds to the 2D-Histogram between X and Y; $p(x)$ and $p(y)$ are the marginal probability distributions.

When determining the next slice to contour, it needs to be considered how well the initially contoured slice and the second contoured slice represent their neighbouring slices. The most representative sample for the overall segmentation is the one that maximizes the total mutual information I_{MI} together with the initial input slice. It is not enough for the second contoured slice to be different to the initial contoured slice; the second contoured slice should be similar to the remaining image slices.

In the segmentation approach described in Chapter 3 the nearest contoured slice is used for segmentation. Therefore, the representativeness for an initial contour on the image slice X and a second image slice Y is given by

$$I_{\mathrm{total}}(X,Y) = \sum_{n=n_{\mathrm{low}}}^{n_{\mathrm{mid}}} I\left(X,S_n\right) + \sum_{n=n_{\mathrm{mid}}}^{n_{\mathrm{up}}} I\left(Y,S_n\right), \tag{6.3}$$

where n_{low} and n_{up} are the superior and inferior boundary slices of the structure. S_n is the image slice at slice number n. The middle slice n_{mid} is the slice between the two user input slices, up to which propagation is used, starting from the respective user input.

For the mutual information approach, a 3D bounding box of the structure is estimated following the initial segmentation using the initial user input. Mutual

information is only calculated for the patches within the bounding box, so that potentially irrelevant information can be excluded from the image comparison.

The most representative slice is the one that maximizes I_{total}. If more than one user input is given and a subsequent slice is predicted, iterative prediction, as described in Chapter 3, is used up to the middle between two user inputs.

6.2.1.3 Auxiliary model

The uncertainty-based approaches introduced above use a convolutional neural network to transfer the uncertainty map to a segmentation quality prediction, to then be able to query the next slice to segment.

A more direct approach to segmentation quality estimation is used by ConfidNet [188]. The network uses feature maps at the end of the segmentation pipeline to predict the model confidence. These feature maps contain information about the segmentation, as well as the image information, and therefore, the image segmentation does not need to be provided separately to the network. Because the model already has extracted the features from image and segmentation, ConfidNet only uses a small classification network instead of the VGG-like architecture of the segmentation quality prediction network.

Here, the network aims to predict the next input that will improve overall segmentation performance the most. The auxiliary network was trained using the output of the penultimate layer (before 1x1 convolutional layer) of the segmentation model as an input to the auxiliary model. The auxiliary model consists of four 3x3 convolutional layers (f=128,64,32,1 respectively). ConfidNet used a final sigmoid layer [188]; here, a final linear activation function was chosen for the regression task. The network architecture is illustrated in Figure 6.4.

The contextual learning model has learnt to extract complex features to produce the segmentation of a given structure. To benefit from these deep features, they were fed to the regression model that predicts the segmentation performance of the model. As proposed by Corbiere et al. the squared error loss was used to

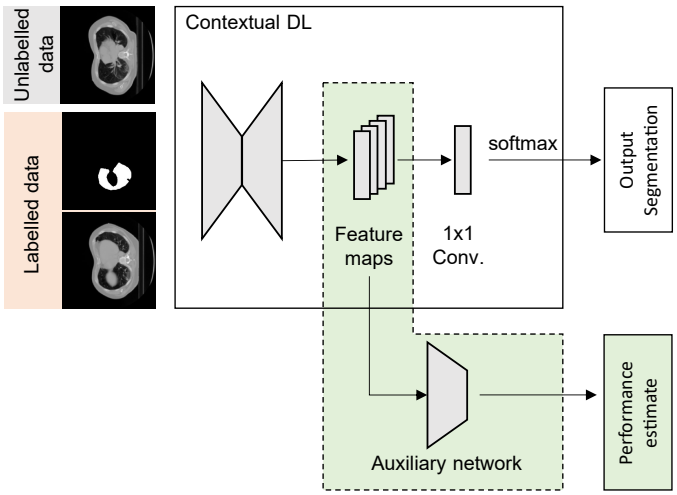


Figure 6.4: Illustration of the pipeline for query selection using a deep learning auxiliary regression model. The auxiliary model takes the feature maps from the penultimate layer to determine a score for the current target image slice. By processing all slices in a scan, those slices can be ranked. The best scoring slice will be suggested to the user.

train the auxiliary network [188]. The auxiliary network was trained for 30 epochs with the Adam optimizer using early stopping.

The segmentation performance used for regression was the 3D relative APL if that slice is chosen as an input and normalized across all potential second inputs.

The next slice to contour selected by this method was the slice with the lowest predicted score, as this corresponds to the lowest overall APL.

6.2.1.4 Data and preprocessing

The same set of data and preprocessing was used as in Section 3.3. The data includes CT images and delineations of 19 different structures of interest from various openly available datasets [132]–[135].

6.2.1.5 Experimental details

The experiments described in the following section serve two purposes. First, to determine the performance given different user inputs, in particular to understand which additional inputs may result in improved segmentation. Second, to evaluate different selection methods to query the next slice to contour, as illustrated in Figure 6.5.

Performance of multiple user inputs at different input positions First, all test cases were evaluated when given a single user input, performing iterative prediction originating from the single user input. In Chapter 3 the central slice was selected as an input. Here, the model was evaluated for each possible user input to evaluate the relevance of input position on model performance.

Second, all test cases were evaluated for all input combinations when given two user inputs to analyse how well they complement each other. Comparison to the single user input allowed assessment of how much the segmentation is improved following two user inputs. When two user inputs were given the closest input to a slice was used for the starting slice of propagation.

Predicting the best slices to contour next Four different approaches to next slice selection were investigated: (1) Entropy, (2) MC Dropout, (3) Mutual information, and (4) Auxiliary network. These were compared to the baseline of a randomly drawn user input from all remaining unlabelled slices, and the optimal next input which was determined after calculating all possible input combinations.

The first slice to contour was chosen as the central slice in the structure, then the different approaches were used to select the second, third, fourth and fifth user input. The model performance was then evaluated using iterative prediction. The selection process was described for selecting the second input for the varying query methods in Sections 6.2.1.1 to 6.2.1.3. Subsequent inputs were selected by calculating the respective selection score analogously, but using two, three or four previously labelled input slices. The best ranking slice was then chosen as the next input.

6.2.1.6 Evaluation measures

The quality of the segmentation was quantified using the 3D DSC [142], 95% HD [146] and relative APL [147], as described in detail in Section 3.3.1.6. On inference, the generated masks were upsampled to the original 512x512 imaging resolution using bi-linear interpolation.

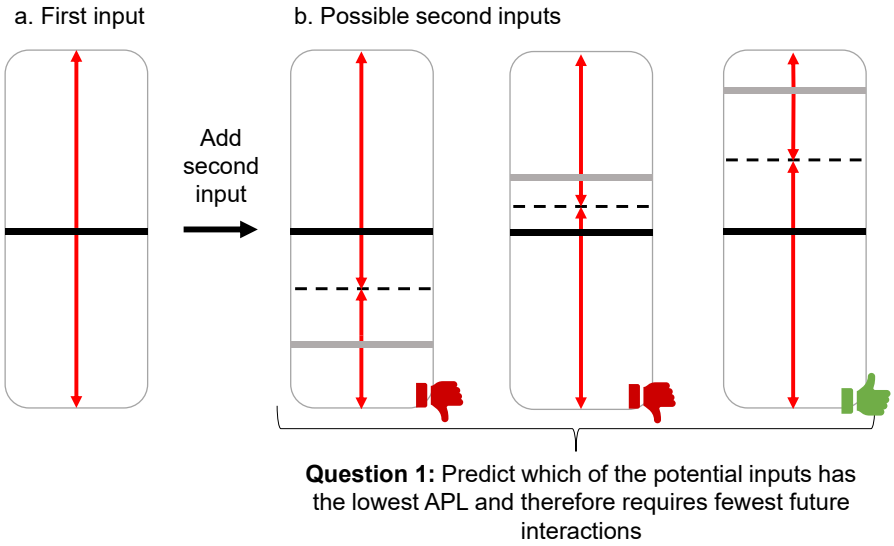


Figure 6.5: Guiding user interaction. (a) Contour first slice. (b) Predict which second slice minimizes APL in order to improve segmentation performance faster.

Additionally, when scoring the different slices for the next best slice to contour, the slice score was ranked from the least recommended to the most recommended slice and compared to the ranked worst and best performing slices.

6.2.2 Results

6.2.2.1 Model performance based on input position

Single input The segmentation performance of the model depends on which slice is used as an initial input. Examples of how slicewise (2D) segmentation performance varied throughout the structure given a certain input position are shown in Figure 6.6. For a given input position, the segmentation quality was better in the direct vicinity of the user input and tended to worsen towards the inferior and superior edge of the structure. As shown in Figure 6.6a, the segmentation quality for the heart did not change much for different inputs, except when the input slices were at the boundary of the structure. For the spleen (Figure 6.6b), the segmentation deteriorated if the input was selected from the inferior half of the structure. In that case, the relative APL reached 0.8 for slices in the superior half of the structure, while the segmentation performance was high towards the inferior end with a relative APL of 0.2.

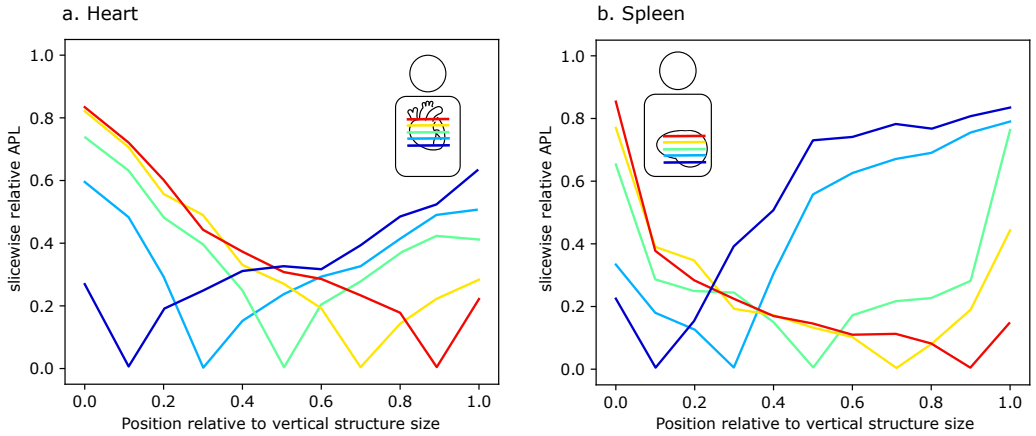


Figure 6.6: Relative APL throughout the structure when using iterative prediction for user inputs at different positions in the (a) Heart and (b) Spleen. Relative APL tends to be lower closer to the respective user input and higher towards the inferior and superior edges of the structure.

The 3D segmentation performance for input positions throughout the structure is shown in Figure 6.7 for example structures. For the heart (Figure 6.7a), the model performance was constant with a relative APL of approximately 0.4 regardless of the chosen input slice, dropping sharply if the input slice was chosen at either the inferior or superior edge of the structure, each with a relative APL of about 0.6. For the spleen (Figure 6.7b), the overall segmentation performance was poorer if the input was chosen in the inferior half of the structure, for which the relative APL goes up to 0.8.

Two inputs Figure 6.8 (left column) shows the APL of the 3D segmentation for different structures if two inputs are given for all possible user input positions. The model performance varied depending on the combination of two input positions. For example, the input slices 8 and 15 of the spleen resulted in an APL of approximately 1500 mm. The input slices 3 and 18 of the spleen resulted in an APL of 2800 mm. The optimal input position combinations varied for each structure.

The change in APL for a given first input after providing a second input is shown in Figure 6.8 (right column). Not all additional inputs resulted in improved segmentation as compared to the first single input alone. The second slices that did not improve segmentation performance tended to be located at the edges of

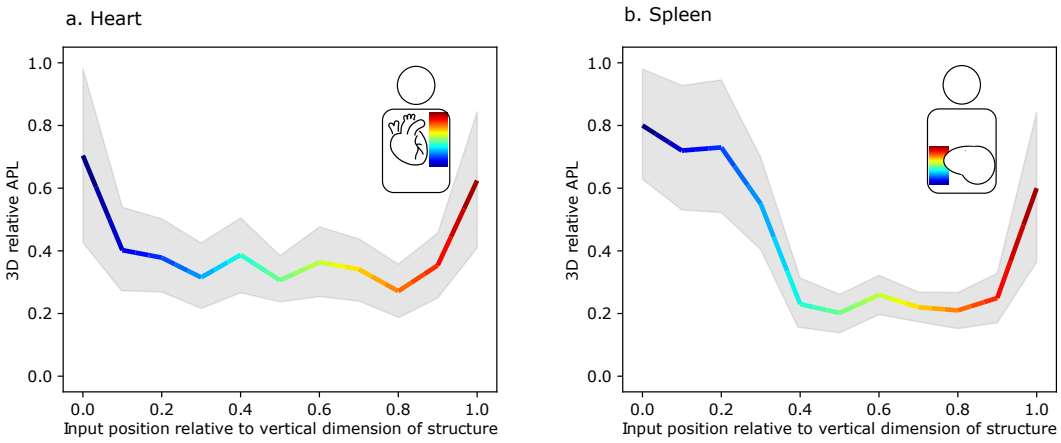


Figure 6.7: Relative APL for 3D segmentation given a user input at different positions relative to the structure size, shown for (a) Heart and (b) Spleen. The 3D relative APL is lower if the user input is at the middle of the structure. User inputs at the inferior or superior edge tend to result in poorer segmentation performance. The shaded area indicates the standard deviation of the APL for different test samples.

the structure, as seen for example: heart (slices 0 and 1) and spleen (slices 18 and 19). For the spleen, as seen in Figure 6.9, a user input at the edge of the structure resulted in poor segmentation, because a segmentation error at a nearby slice is propagated throughout the rest of the structure.

6.2.2.2 Single slice performance as a proxy for overall performance

The relationship between slicewise segmentation performance given one input, and 3D segmentation performance of the structure if a second input is given is shown in Figure 6.10. Specifically, the figure shows the correlation between the relative APL of a 2D slice if one input was provided as a single input to the model (x-axis), and the relative APL of the full structure if those two slices were given as an input to the model (y-axis). This is shown explicitly for selected example structures for individual cases (6.10, left column), as well as after ranking the 3D and 2D APL pairs and showing a histogram for the pairs 6.10 (right column) for all test cases.

The correlation between 3D APL and slicewise 2D APL is not always directly visible for individual cases, but after evaluating all cases a general negative correlation is observed. Qualitatively, higher APL of one slice leads to lower global APL after that slice is given as a second input.

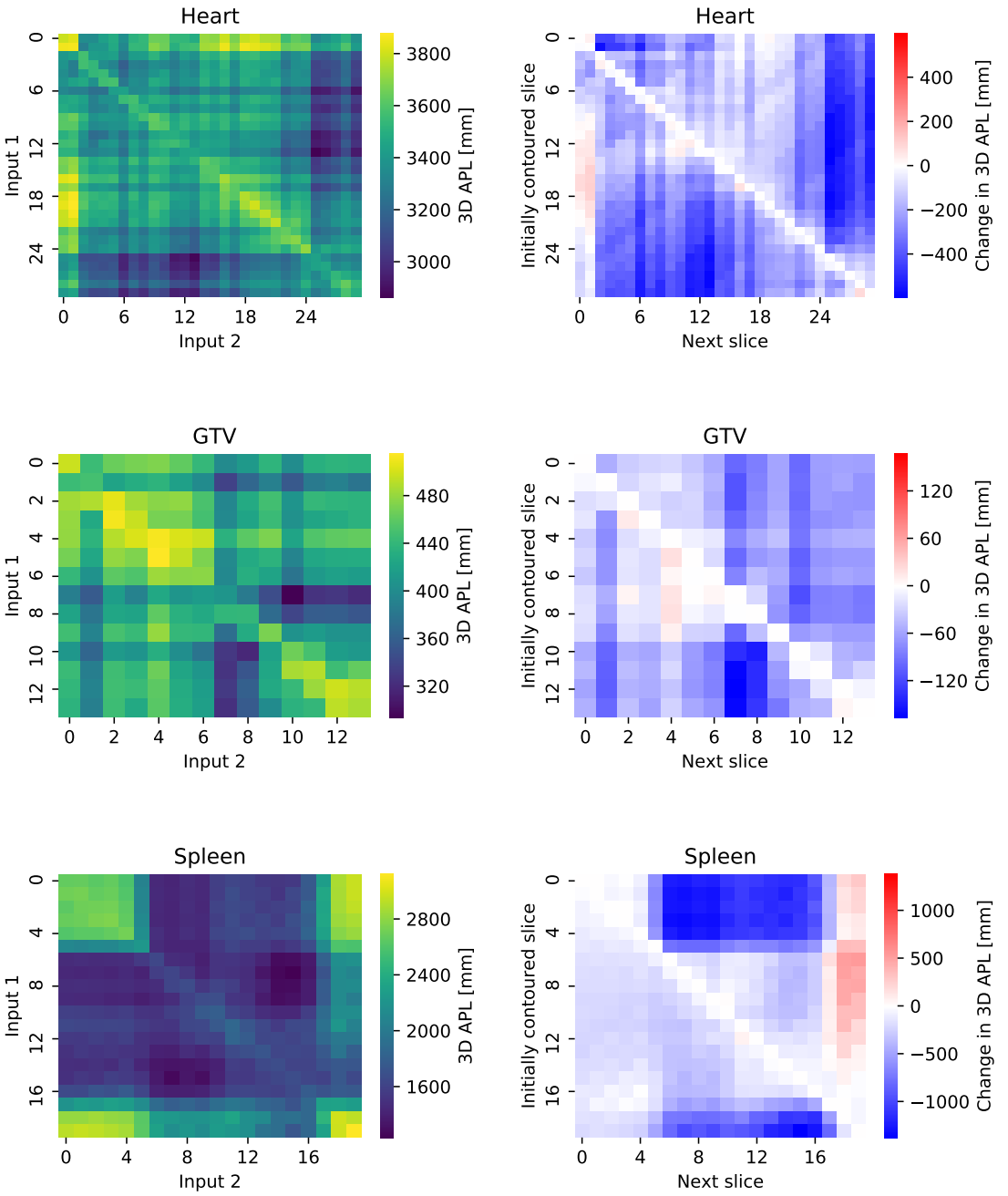


Figure 6.8: Contextual deep learning with multiple user inputs. Left column: Total APL for the 3D segmentation using the interactive deep learning model [2] with two inputs. Right column: Change in APL for a given structure, when the second input is given in addition to the initially contoured slice. This illustrates how much a contour can be improved upon a second input, given a specified first contour. Results are shown for the Heart, Lung GTV and Spleen.

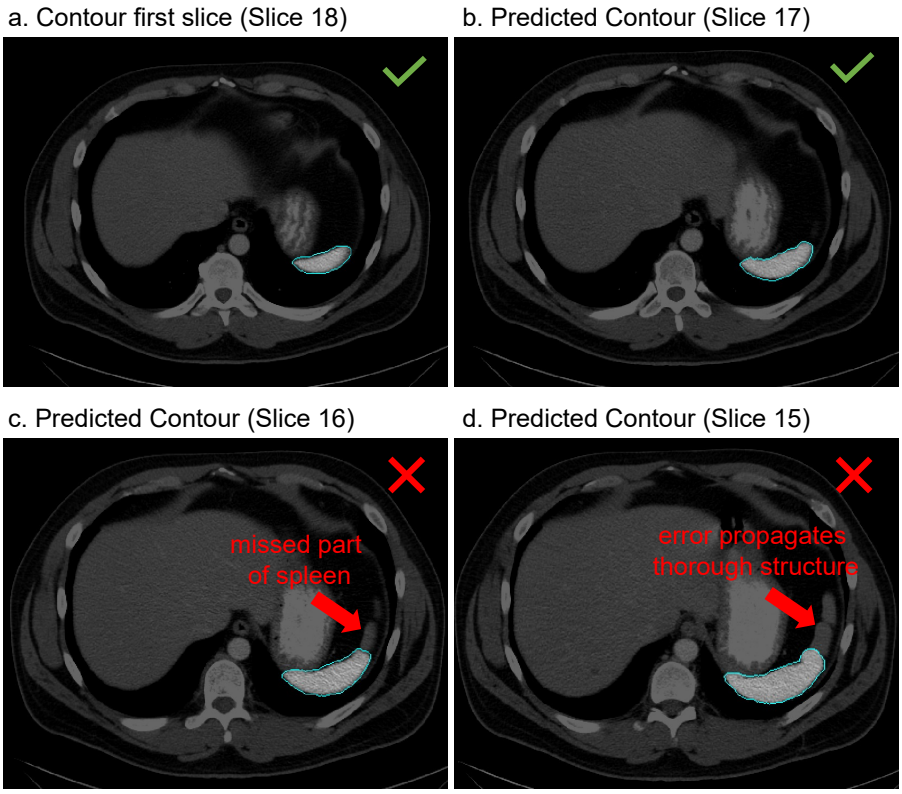


Figure 6.9: Example of a poor performing initial user input for the spleen. (a) Initial user input at slice at the superior edge of the structure (slice 18) and subsequent propagation of contour through the structure. (b) The adjacent slice 17 is correctly segmented. For the subsequent slices (c) 16 and (d) 15, the contour fails to contour a segment of the spleen. This error accumulates throughout the structure. This is the spleen shown in Figure 6.8 and explains why slices 17-19 are poorly chosen user inputs, resulting in increased APL.

6.2.2.3 Predicting the next slice to contour

The model performance when given two user inputs following the different query strategies is summarized in Table 6.1. The mutual information-based approach was the best performing method for the structures included in training, whereas MC Dropout performed better for the spleen (excluded in training). The auxiliary network performed better than random but was among the worst performing approaches for nearly all datasets. Overall, the predictions did not vary strongly from each other for selecting only one extra slice. Larger differences were observed for the subsequent slice prediction.

The performance for subsequent user inputs is shown in Figure 6.12. All prediction approaches lie between randomly and optimally chosen slice. The

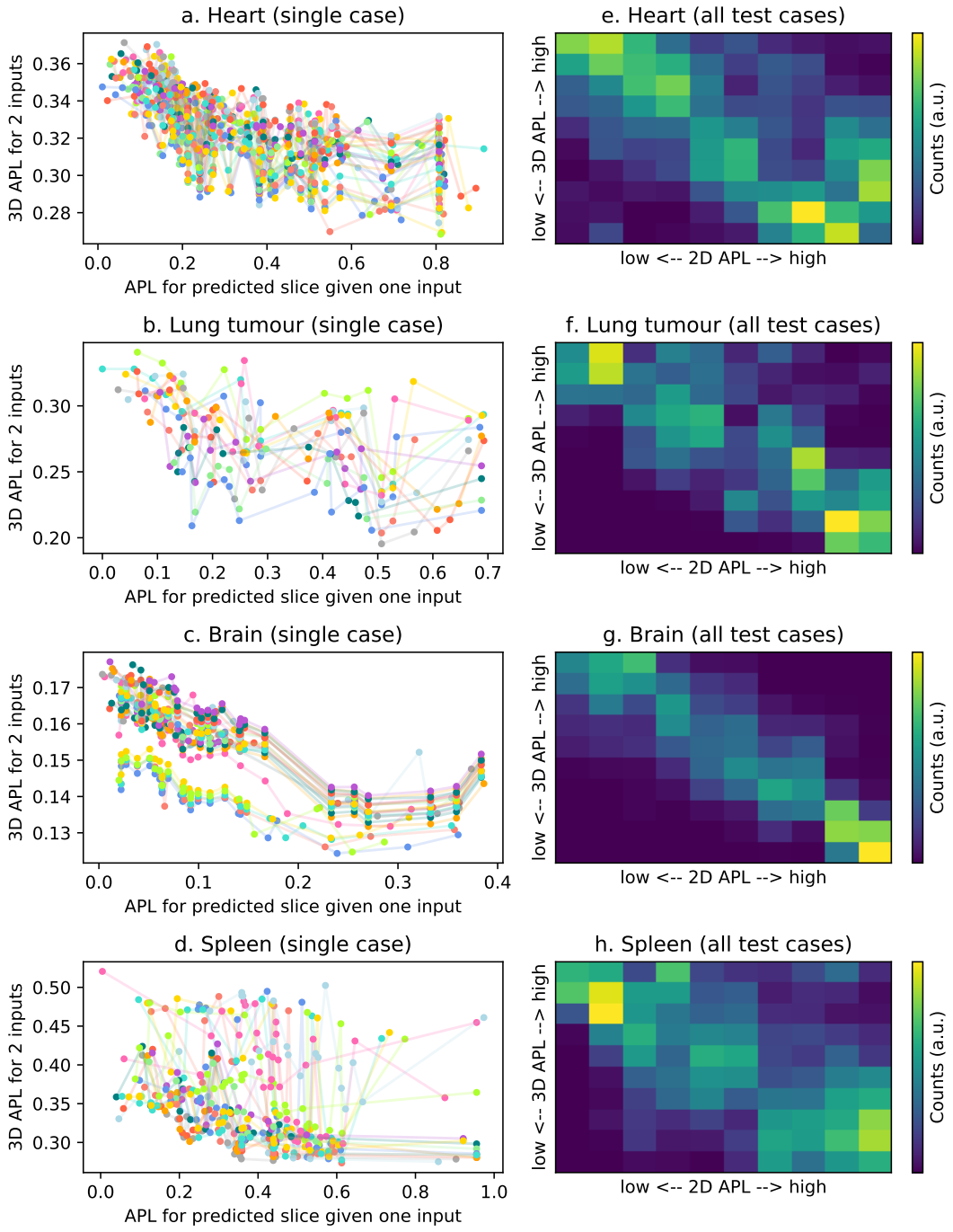


Figure 6.10: Illustration of correlation between segmentation performance and a suitable next user input for four example structures. Left: The scatter plot shows the segmentation performance of a given 2D slice vs the overall segmentation performance if that slice were given as an additional input to the model. Each line corresponds to one first input at a given slice position. Right: The 2D histogram shows the pairs of ranked 2D and 3D segmentation performance for all test cases.

auxiliary network performed well for the first next slice as an input, but did not perform well when predicting subsequent slices.

A 2D histogram for the entropy-based approach and optimal chosen slice is shown in Figure 6.11 (left column). This shows that the prediction scores correlated with the optimally chosen slices for all structures but the Heart. However, as seen in the histogram for the next best slice (Figure 6.11, right column), the entropy approach can make severe errors in the next best slice prediction. While the majority of predictions show good agreement with good next slice choices for most structures, the approach fails to predict the heart correctly for the next best slice.

Table 6.1: Segmentation performance (relative APL) for different datasets, when the next slice is chosen by each query approach. Significantly better performing query approaches are highlighted in bold. Wilcoxon signed-rank test ($p<0.01$) was used.

Data	Random	Entropy	MC	MI	Aux.	Optimal
A	0.42	0.42	0.41	**0.38**	**0.38**	0.36
B	0.25	0.25	0.24	**0.23**	0.24	0.22
C	0.49	0.44	0.44	**0.40**	0.43	0.37
D	0.46	0.45	0.46	**0.42**	0.45	0.40
E	0.47	0.45	0.46	0.45	0.45	0.43
F	0.33	0.32	0.33	**0.31**	0.32	0.28
G	0.30	**0.25**	0.26	0.29	0.27	0.22

6.2.3 Discussion

6.2.3.1 Model performance based on input position

Single input Model performance when given one user input varied based on the input position. For most structures (e.g. Heart, shown in Figure 6.6) the model performance only varied slightly for user inputs at input positions close to the center of the structure. However, inputs at the edge of the structure nearly always failed to produce good segmentation (see Figure 6.6). Edge inputs often corresponded to slices where only a very small part of the structure is visible, sometimes only a few pixels. This results in a small number of the labelled class as an input - particularly after down-sampling. This may not be enough for the model to detect which structure is to be segmented. The very edge of the structure

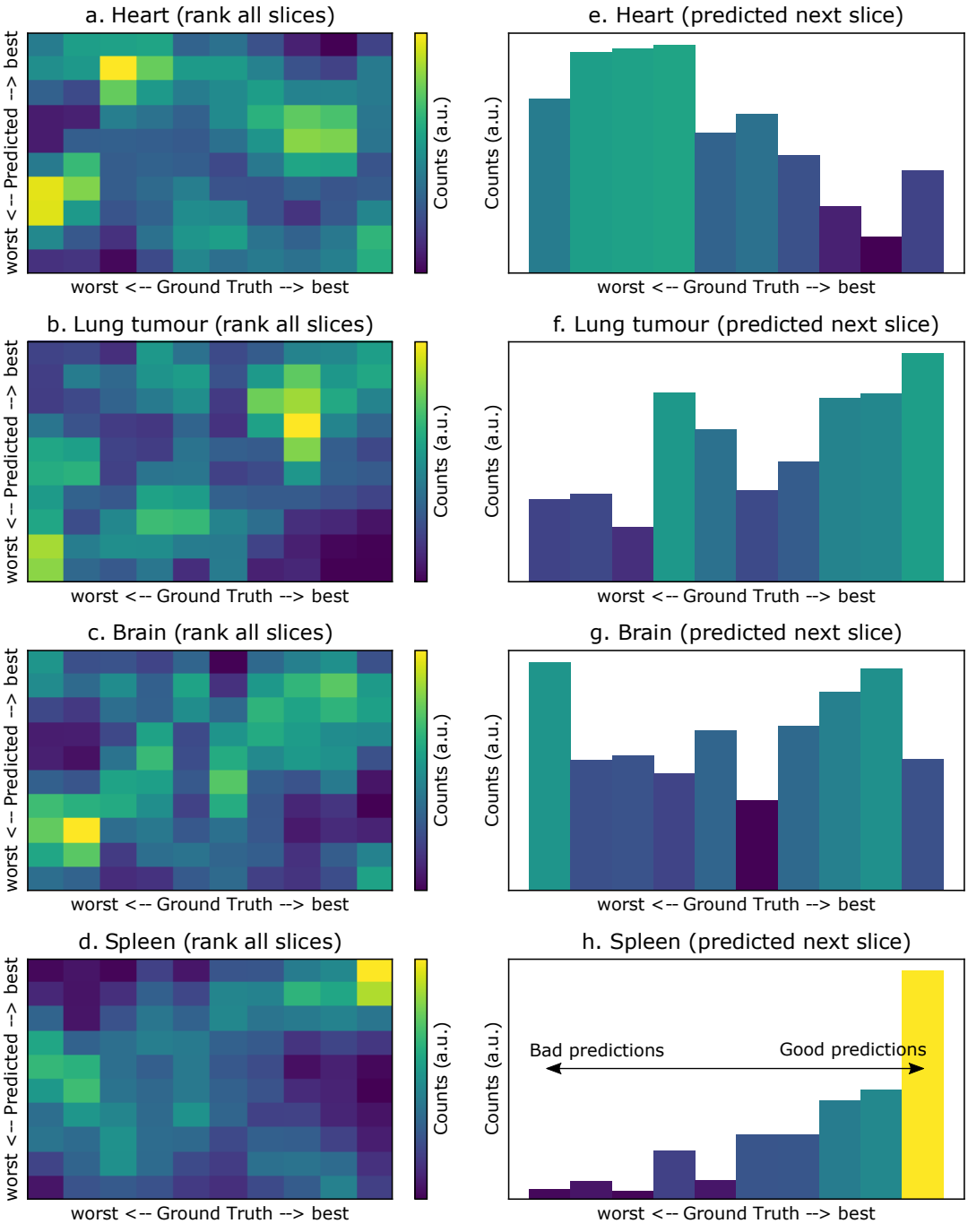


Figure 6.11: Next slice prediction using entropy. Left: 2D histogram showing pairs of predicted and optimal slice selection. Right: Histogram for the predicted best slice and how well it matches the ground truth, corresponding to the top row of the 2D histogram.

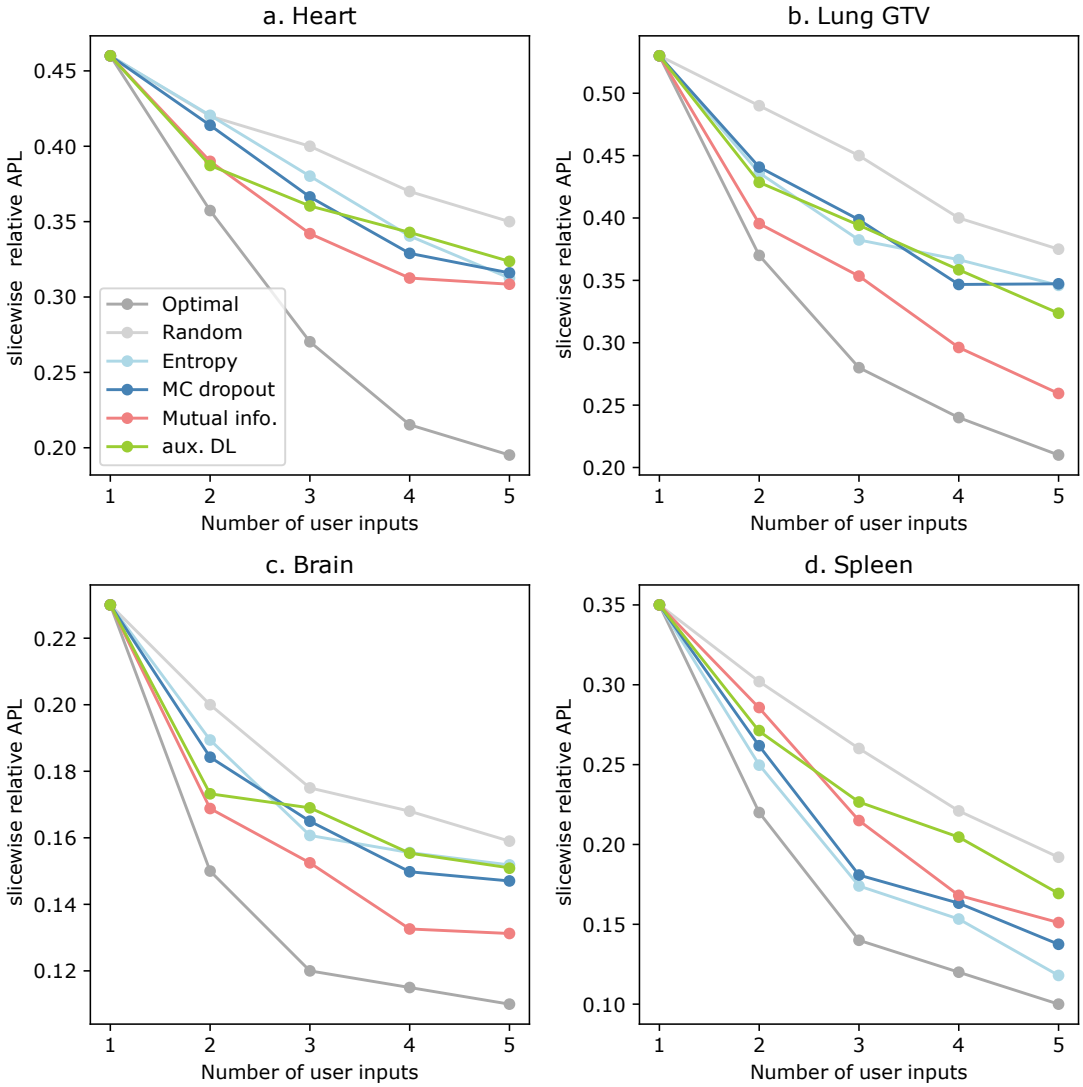


Figure 6.12: Selecting subsequent next inputs. Model performance when contouring the central slice of a structure and then contouring selecting the next best slice to contour using various query approaches. The sampling is repeated for up to 5 user inputs for each approach.

is also underrepresented in sampling during training, as most randomly selected training samples will originate from a slice somewhere between the edge slices, while structural changes from slice to slice may be more marked at the very edge of the structure. Ultimately, poor performance at the edges is not a great disadvantage to the model, as it is unlikely to be the slices that the user will choose as they start contouring. Instead, a user is more likely to initially contour a central slice of a structure and then use the propagation approach, as was observed in the evaluation

by clinicians (Chapter 4). Edge slices of a structure tend to only have few pixels and are also short in terms of the length of the contour to be drawn. Therefore, these slices will require less time to edit if the model fails to predict them correctly.

As shown in Figure 6.6 the segmentation performance tended to be better in the proximity of the initial input slice, but the overall performance varied little for the full structure for any given input slice (Figure 6.7). Therefore, a spatially distant second user input complements an existing user input, as each input benefits its immediate proximity. The small variance in 3D segmentation performance based on input position may indicate that the model learns structure-specific features that help it segment what it has seen before. This is true for all structures, except the spleen, where the model performance for the overall structure varied depending on user input position. This is likely because the spleen was not included during training and therefore is more difficult for the model to predict from context.

Two inputs Figure 6.8 showed that the optimal two user input positions vary for different structures. Overall, a second input slice spatially at a greater distance from the initial input tended to improve the 3D segmentation the most. A larger slice separation on average will correspond to an image slice that diverges more from the initial slice. Indicating slice dissimilarity can result in an information gain for the model which agrees with the mutual information hypothesis.

Figure 6.8 (right) showed that not all additional inputs improve model performance. This is attributed to the way the overall segmentation performance is calculated. Each user-provided input is used to predict the slices closest to it. This means that two user inputs are used to predict up to the slice midway between them. However, the performance may be better if this boundary - up to which point a certain user input is used to predict a slice - is shifted. This is because one of the two user inputs may result in a poor starting point for prediction and propagate segmentation errors up to the propagation boundary at the midway point. When the additional user inputs were at the edge of the structure the resulting performance was often poor and even sometimes lead to an increased APL. As

shown in Figure 6.6, edge slices resulted in poor quality segmentation of the rest of the structure. A qualitative example was shown for the spleen in Figure 6.9.

Figure 6.8 also shows that certain slices nearly always improved the model for a given case if used as a second input regardless of the initially contoured slice. For an example case showing the heart, slice 25 to 27 of the heart resulted in the lowest APL for the initially contoured slices 0 to 23. Similarly, slice 7 as the next input slice for the lung GTV minimizes the 3D APL for nearly all initially contoured slices. These slices are either particularly informative for the model to predict other slices, or they are very distinct from the remaining slices and are therefore hard to predict if they are not contoured manually. On visual inspection, these slices do not seem to show very distinctive characteristics compared to other slices. However, the features that the model extracts from them may provide it with valuable information, allowing it to make better predictions.

6.2.3.2 Single slice performance as a proxy for overall performance

The auxiliary network introduced in this section processes the target slice separately. Yet, to choose the next best slice the score relative to the other slices is important, but the model has no information about this at inference time. Therefore, it must be asked: how can the model be expected to make such a prediction if it has no information about the remaining slices? Figure 6.10 shows that the 3D segmentation performance correlates with the segmentation performance for a 2D images slice. This indicates that if the model is able to predict its segmentation performance for a given slice it may be able to predict which slice to contour next. If the model is able to predict this with sufficient accuracy for each slice, the performance score of the individual slices can be ranked and the next slice to be contoured can be predicted. This justifies that the auxiliary network investigated in this structure calculates a score per slice, rather than evaluating all slices simultaneously.

6.2.3.3 Predicting the next slice to contour

The comparison of the different slice selection methods showed that the overall representativeness-based query approaches performed best for most datasets. Only

for Dataset G (spleen) did the uncertainty-based approaches outperform representativeness. A possible explanation for this may be that the Datasets A to F were included in training, and therefore, the uncertainty associated with these structures may be lower, as the model may accept these as correct in nearly all instances. For structures that the model was not trained on, such as the spleen, there is likely to be lower confidence in the predictions. Overconfidence of the model likely contributed to the poor performance of the model in terms of next slice selection based on uncertainty. In the literature it has been noted, that the softmax probabilities of a neural network are poorly calibrated and should not be viewed as a direct representation of confidence [189], but here, the entropy and MC dropout approach performed similarly.

The correlation between predicted and ideal rank was poor for the heart. A possible reason for this may be, that difference in performance based on input position and the relative improvement when a second input is given is low (see Figure 6.8). This is likely due to the heart being an easy structure to segment and so the model shows a similar performance regardless of input position. Although the ranked histogram for the heart indicates that a majority of predictions are worse than random, the overall performance for this structure is as good as random because the effect the chosen input has is relatively small compared to other structures, such as the spleen (see Figure 6.8).

One problem with the uncertainty-based approach and the auxiliary model is that they do not compare one slice to all other slices at inference time, as is done using the mutual information approach, where the overall representative is taken into account. Instead, each slice is assigned a score by the model and subsequently ranked. Memory limitations prevent storing all these inputs at the same time, much less training a model on this. However, as model performance on a 2D slice correlates with a suitable next input, the model is still able to perform better than random slice selection.

Finally, computational costs need to be considered when evaluating the suitability of the various approaches. MC Dropout required multiple forward passes through

the network, which would result in long waiting times for the clinician. The auxiliary network is small in terms of network parameters compared to the segmentation model and can be calculated alongside the model predictions, and therefore is able to provide user feedback in real time.

6.2.4 Conclusion

In this section, a framework for guided interactive segmentation of medical images is proposed. It demonstrated that APL is reduced if an Active Learning-based query approach for the next slice to contour is chosen. This may reduce contouring effort and time spent contouring for clinicians. All but the MC dropout approach can be easily implemented to give user feedback in real time.

Uncertainty-based next slice selection did not provide consistent results for the structures investigated. The auxiliary network performed well when predicting the next slice to contour for the second slice, but performed poorly for each subsequent prediction. Overall, user input representativeness was the most consistent approach for predicting the next best slice. This was attributed to the approach not only considering the impact on a given slice, but calculating the representativeness with respect to all other slices. Additionally, it explicitly took into account the iterative prediction approach that uses the closest of two user inputs.

Each query approach still made bad predictions for individual slices, even suggesting the worst possible next input slice in some cases. All approaches performed better than random slice selection. However, a clinician intuitively may select slices as the next input in a better than random way. The evaluation by clinicians in Chapter 4 was designed to measure impact on contouring time and interaction behaviour of clinicians. It did not include specific guidelines on how to use the contouring tool. Sometimes clinicians would contour only the first slice and use the propagation tool once, followed by manual edits. At other times, they contoured several slices and then used the deep learning tool. However, in this chapter when user-guidance was investigated, it was assumed that after each user

interaction the deep learning prediction tool is applied. Additional evaluation by clinician is needed to conclusively determine the benefit of user-guidance.

6.3 Optimizing Contextual Deep Learning for Multiple Inputs

Following the flowchart in Figure 6.1 - after a user has chosen and interacted with an additional slice - the best possible prediction using those inputs needs to be made. So far, the contextual deep learning tool has used the closer of two user inputs to make a prediction for two slices. This is a reasonable assumption, as a closer slice is likely more similar than a more distant slice, and the error propagated is likely smaller when a slice closer to the initial one is chosen. But different structures may have significant unique differences in shape or size, that make a different input slice a better choice. Slice distance may not always be a suitable indicator of which of the user-provided contours should be used to predict the contour on another slice, nor is the slice at the midway point necessarily the optimal propagation boundary.

To improve the contextual deep learning model, the iterative and direct propagation approaches were investigated to determine how much segmentation quality can be improved when a different prediction method is used.

6.3.1 Materials and Methods

The methods for this section are generally the same as for the prediction of the next best slice. Model uncertainty and image representativeness are used to optimize the iterative and direct prediction method. Figure 6.13 illustrates how the contextual deep learning tool handles multiple inputs (baseline), as well as how the prediction may be improved (optimized), by changing the propagation or prediction approach.

6.3.1.1 Data and preprocessing

This section uses the same data and preprocessing as the previous section and was introduced in detail in Section 3.3.

6.3.1.2 Experimental details

Model performance baseline The model performance for different input positions for iterative prediction when using the middle between two user inputs as

the propagation boundary was determined as in Section 6.2. For direct prediction, the performance for all user input combinations for the test cases was determined when using the closest user input, as illustrated in Figure 6.13b.

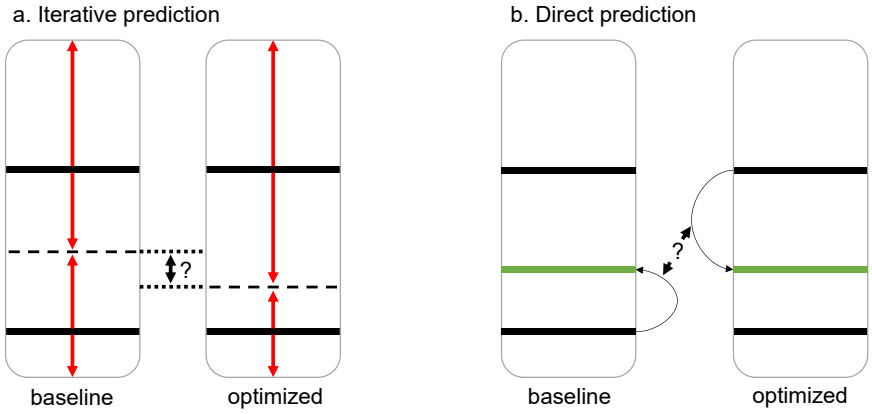


Figure 6.13: Optimizing contextual deep learning. (a) For iterative prediction, choose which propagation boundary is best. (b) For direct prediction, choose on a slice-by-slice basis which input is best.

Iterative prediction - choosing the boundary between two user inputs

Three different approaches to selecting the propagation boundary were investigated: (1) Entropy, (2) MC Dropout and (3) Mutual information. These are compared to the baseline when the propagation boundary is chosen as the middle of two user inputs, as well as the optimally chosen propagation boundary which is determined after calculating all possible input combinations.

To obtain the optimal propagation boundary using representativeness, mutual information and feature space distance were calculated following Equation 6.3, but here the propagation boundary N_{mid} was varied and the slice with the highest overall mutual information was selected.

For model uncertainty, the segmentation quality prediction network was evaluated using both user inputs, then the propagation boundary was selected that minimizes predicted APL for the structure. Note, that unlike in the previous section where the next slice was selected, both input slices are known, so the overall performance can be optimized, rather than evaluating which slice is poorly segmented.

Direct prediction - choosing between two user inputs Three different approaches to select which input slice should be chosen for a given unlabelled slice were investigated: (1) Entropy, (2) MC Dropout and (3) Mutual information. These were compared to the baseline of choosing the closest user annotated slice as an input, as well as the optimally chosen input slice which was determined after calculating all possible input combinations.

For the direct approach, all unlabelled slices were compared to all labelled samples. The input slice with the highest mutual information or lowest predicted APL (for uncertainty methods) was chosen, depending on the selection method.

6.3.1.3 Evaluation measures

The segmentation quality was quantified using the 3D DSC [142], 95% HD [146] and relative APL [147], as described in detail in Section 3.3.1.6.

6.3.2 Results

6.3.2.1 Iterative prediction - Choosing the boundary between two user inputs

Figure 6.14 shows the segmentation performance for all possible propagation boundaries (left column) and the input representativeness score for the respective propagation boundary (right column) for example cases. The choice of propagation boundary for iterative prediction can vary greatly within a given structure. For example, in case of the heart, if input 1 was slice 57 and the second input was at slice 47, the ideal propagation boundary was at slice 55 with an APL of approximately 3300 mm, whereas if the propagation boundary was chosen at slice 48, the APL was 3600 mm. If the midpoint between the two inputs was chosen (slice 52) then the APL was 3400 mm.

For most input combinations, the optimal propagation boundary was close to the middle between the two slices. However, in some cases the optimal performance was achieved when the boundary was closer to one input than the other. This is shown as an extreme case for the spleen: When input 1 was at slice 27, and

input 2 was 32, 33, 34 or 35, the best performance was achieved when input 1 is used to propagate all the way up to these other slices.

Overall, empirical performance was not smooth with various local minima along the shifting boundaries. Interestingly, the global minima for the propagation boundary were clustered along a few slices, as illustrated in Figure 6.15. Input representativeness (right column, Figure 6.14) was very smooth and quite often peaked somewhere in the middle between two user inputs. But not in all cases, as for the spleen for example, if input 1 was at 27 and if input 2 was at the most caudal end of the superior part of the structure, then mutual information was highest, if the boundary was chosen such, that input 1 was used for propagation - showing good agreement with the empirical performance.

The change in segmentation performance for the different propagation boundary selection approaches is summarized in Table 6.14 for various datasets. Mutual information performed better than uncertainty-based measures for all but dataset D for which entropy-based selection achieved better results. Even the best selection criteria achieved only small reductions in relative APL.

Table 6.2: Relative APL for datasets A to G, if the boundary for propagation is chosen using different selection methods. Significantly better performing boundary selection methods are highlighted in bold. Wilcoxon signed-rank test ($p<0.01$) was used.

Dataset	Middle	Entropy	MC Dropout	MI	Optimal
A	0.42	0.43	0.42	**0.40**	0.39
B	0.25	0.27	0.25	**0.24**	0.23
C	0.49	0.50	0.52	**0.47**	0.44
D	0.46	**0.45**	0.46	0.46	0.42
E	0.47	**0.46**	0.47	**0.46**	0.45
F	0.33	0.35	0.34	**0.32**	0.31
G	0.31	0.29	0.31	**0.28**	0.26

6.3.2.2 Direct prediction - Choosing between two user inputs

Figure 6.16 shows the empirical result for which input was better when using direct prediction for two example cases (left column) and the prediction for which slice

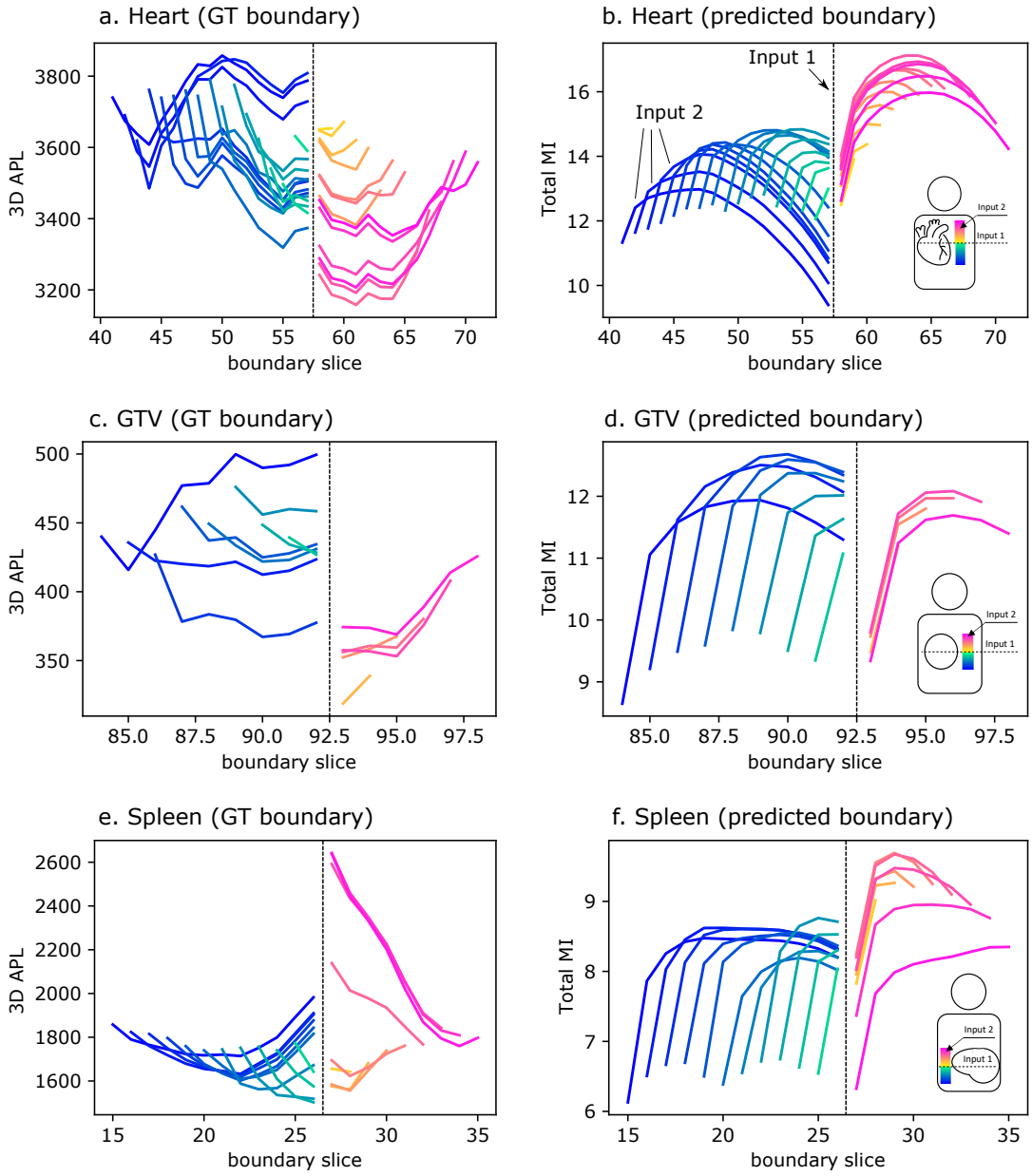


Figure 6.14: Comparison of empirical performance for a given propagation boundary and combined mutual information (MI) if that propagation boundary is chosen. Left: APL for full structure for different input combinations. Each coloured line represents a different second input to the model. The points along the line show the performance for a given chosen boundary for the respective Input 1 and Input 2. The location of Input 1 and Input 2 within the structure is illustrated in the insets. For example, in case of the heart, if input 1 was slice 57 and the second input was at slice 47, the ideal propagation boundary was at slice 55 with an APL of approximately 3300mm, whereas if the propagation boundary was chosen at slice 48, the APL was 3600mm. If the midpoint between the two inputs was chosen (slice 52) then the APL was 3400mm. Right: Mutual information calculated using Equation 6.3.

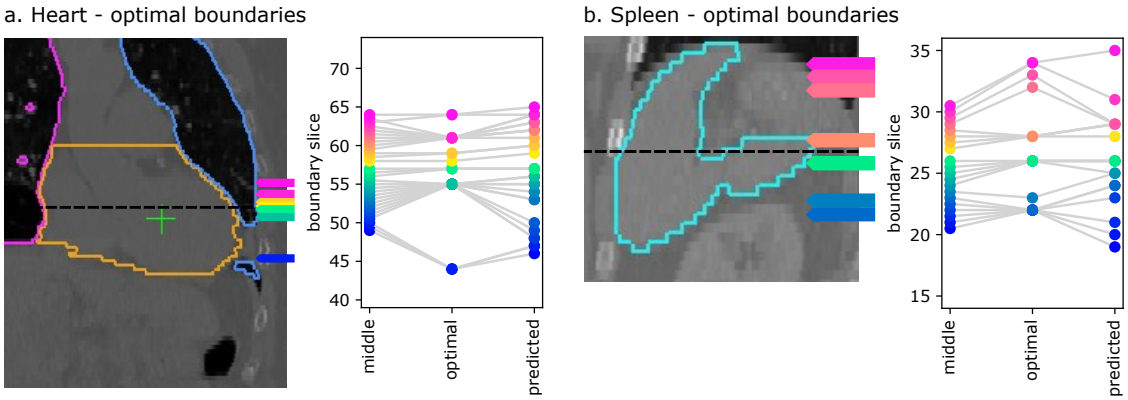


Figure 6.15: Coronal view of (a) Heart and (b) Spleen, illustrating that optimally selected boundaries for propagation may cluster together.

was better when using mutual information for comparing the slice to be contoured with input 1 and input 2 (right column).

The empirical result (ground truth) shows that the optimal slice to use for prediction between two input slices tended to be the slice closer to the slice to be segmented. However, this was not always the case. For example, for the lung GTV for input 1 at the inferior edge of the structure (Slice 0 and 1), and Slice 2 in the superior half of the structure (Slice 30), the inferior 6 slices of the structure (Slices 0 to 6) achieved better results if input 2 from the upper half of the structure was used to make a prediction, although that user input is further away.

The better input slice as determined by mutual information is visually similar to the empirical results, as illustrated in Figure 6.16 (right column).

The quantitative evaluation of all test cases is summarized in Table 6.3. This shows that mutual information only performed slightly better than the closest slice chosen. Entropy and MC dropout performed similarly well, achieving a comparable reduction in relative APL for each dataset.

6.3.3 Discussion

6.3.3.1 Iterative prediction - Choosing the boundary between two user inputs

The empirical analysis showed that the ideal boundary was, in many instances, close to the middle slice. A lot of exceptions to this behaviour occurred if one of

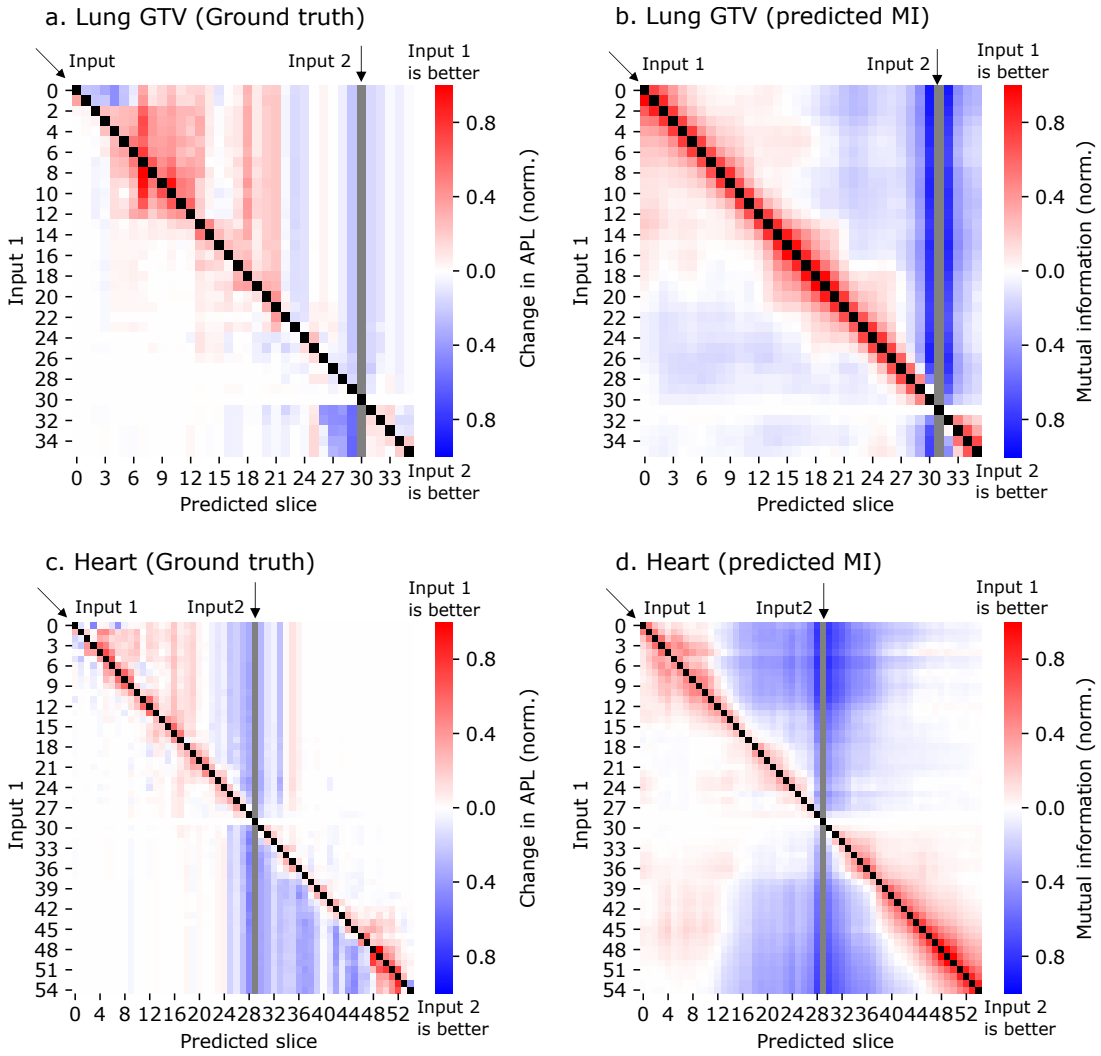


Figure 6.16: Selecting which of two user inputs to use for direct predictions, shown for fixed input 2 and varying input 1. Left: Model performance for different input combinations. Each row corresponds to one input combination. Right: predicted better performance using MI.

Table 6.3: Relative APL for a given datasets for different input selection methods. Significantly better performing input selection methods are highlighted in bold. Wilcoxon signed-rank test ($p<0.01$) was used.

Dataset	Closest	Entropy	MC Dropout	MI	Optimal
A	0.44	0.44	0.43	**0.42**	0.40
B	0.28	**0.27**	0.28	**0.27**	0.25
C	0.51	**0.48**	0.50	0.49	0.46
D	0.49	**0.47**	0.48	**0.47**	0.45
E	**0.48**	0.49	0.49	**0.48**	0.47
F	**0.35**	**0.35**	0.36	**0.35**	0.33
G	0.40	0.37	0.39	**0.36**	0.32

the inputs was close to the inferior or superior edge of the structure, as was shown for the spleen in Figure 6.14e. In that case, the input at the edge of the structure should not be used to propagate through the rest of the structure, therefore the propagation boundary should be close to this input. This reaffirms the observation from Section 6.2, suggesting that edge slices make poor inputs for the contextual deep learning model. This is likely because they often only contain few pixels. The inferior and superior image slices also showed larger inter-observer variability in the case of lung GTVs in the evaluation by clinicians reported in Chapter 4.

Global minima for the empirically best boundary for propagation were clustered to very few slices, as shown in the right column of Figure 6.15. This indicates that no matter the inputs to either side of that boundary there is a relatively significant change in the structure with respect to the two sides of the boundary. An example of this is shown in Figure 6.15. Visually these propagation boundaries do not seem very different to the human eye or could not be made out as distinct boundaries. The selected coronal view of the spleen in Figure 6.15b indicates that the propagation boundaries are at positions where the contour may change more drastically.

Mutual information did not show such clear propagation boundary clusters, also indicating that the difference may not directly rely on image information but the feature space produced by the model.

Input representativeness can adequately capture the magnitude of importance of the boundary position. For example, the spleen had several propagation boundaries with similar resulting APL. For these slices, mutual information attributed a similar score. While the global minima in those cases may not be accurately predicted, a slight variation around the minima would not have a big impact on APL.

Uncertainty-based approaches performed similarly to selecting the middle slice as a boundary for propagation. The middle slice was likely a very good baseline. Mutual information made predictions similar to the middle slice, but may be more sensitive to significant changes in the structure and thus resulting in improved model performance.

6.3.3.2 Direct prediction - Choosing between two user inputs

The direct prediction approach performed less well than the iterative prediction. This is true for choosing the baseline approach (closest input) or the optimal select inputs. Previously in Chapter 3, it was found that the performance of direct prediction was worse than that of iterative prediction. This was attributed to the large distance between the target and user input slice, which makes it more difficult for the model to make predictions based on context, as compared to iterative slice-by-slice prediction. Here, it was shown, that even if the best possible of two user inputs was chosen, direct prediction did not perform better than the iterative approach.

For iterative prediction mutual information improved segmentation performance for all datasets. For direct prediction, it did not improve segmentation in the case of datasets E and F. For the iterative approach, mutual information calculated the representativeness compared to all other slices. In direct prediction, it only compared the respective target slice with both user inputs. Small errors when comparing single slices using MI may be smoothed out when the total representativeness is calculated, as is done for iterative prediction.

6.3.4 Conclusion

In this chapter, different methods to make better predictions given two user inputs were investigated by modifying the contextual deep learning model.

It has been shown that iterative prediction performed better than direct prediction. Selecting the optimal boundary for contour propagation was most successful using image similarity, whereas uncertainty-based approaches performed about as well as when the middle slice was selected.

Image representativeness was found to be the most promising approach to estimate the next best slice to contour as well as to predict the more informative of two user inputs. This suggests that the contextual deep learning model was better at predicting slices similar to the input image slice that is provided as contextual input. Large variations from the contextual input resulted in worse segmentation.

None of the proposed approaches improved the model drastically. Deep learning approaches dedicated to handling multiple user inputs may be able to achieve better segmentation performance using the same user inputs. Such models will be discussed in the next chapter.

7

Deep Learning Models for Multiple User Interactions

Contents

7.1 Introduction 151
7.2 Double-context Deep Learning Model 152
7.2.1 Materials and Methods 152
7.2.2 Results 155
7.2.3 Discussion 157
7.2.4 Conclusion 160
7.3 Memory Networks 161
7.3.1 Materials and Methods 161
7.3.2 Results 165
7.3.3 Discussion 167
7.3.4 Conclusion 172

7.1 Introduction

The previous chapter investigated how the Contextual Deep Learning model from Chapter 3 can better make use of multiple user inputs. Selecting the propagation boundary or choosing the right user inputs proved to be challenging. Fundamentally, these approaches are limited because they only use one of multiple user inputs to make a prediction. More than one user input may contain relevant information

for the segmentation of a target slice.

This chapter explores deep learning methods that use multiple user inputs simultaneously and do not rely on hand-crafted methods to determine which user input should be selected. First, a double-context model is proposed that uses contextual information from both sides of the slice to be contoured. Second, a memory network is introduced that stores relevant information in external memory from where it can be retrieved for each slice to be segmented. Both proposed networks answer Question 2 in the workflow of Figure 6.1, which is how to use multiple user inputs more effectively.

7.2 Double-context Deep Learning Model

In this section, the contextual deep learning model is adapted to use the two closest contextual inputs to make predictions. Here, it is evaluated whether such an approach can improve segmentation performance compared to the iterative prediction approach discussed in Chapter 3.

7.2.1 Materials and Methods

7.2.1.1 Double-context model

To extend the concept of contextual deep learning to multiple user inputs, the model was adapted to take an additional contextual input to make a prediction. In this chapter, the contextual deep learning model from Chapter 3 will be referred to as *single-context model*, a contextual deep learning model that considers the two closest user inputs will be referred to as *double-context model*. Later, a combination of the two models will be introduced and referred to as *hybrid model*.

For the double-context model, five input channels were provided to the model, which are illustrated in Figure 7.1: the target image slice, the image slice and contour of the first user input, and the image slice and contour of the second user input. The same network architecture as the contextual deep learning model in Chapter 3 was used.

Similar to the training strategy for the single-context model, the previously contoured and target slices had to be chosen carefully. A too small inter-slice distance could incentivise the model to copy the user-provided contour rather than generating a segmentation specific to the target image. Large inter-slice distances would result in inputs outside the structure. To train the interpolation model, the user input slices were sampled randomly without replacement from image slices that contain a contour of the structure of interest. Then, the target image slice was sampled from the remaining slices. Note, that this included target slices not only from in-between the user inputs, but also, where the target image slice lies on the same side of two user inputs. This is not interpolation, but extrapolation which increased the training set diversity. If the two user inputs were sampled with the target image slice restricted to between the user inputs, the slice distances to the target input slice d_1 and d_2 tend to be biased to smaller distances and therefore, prone to copying. Additionally, by sampling from outside the interpolation region, the double-context model could still be applied when only one user input is given. In that case, the single user input would be used for both input channels.

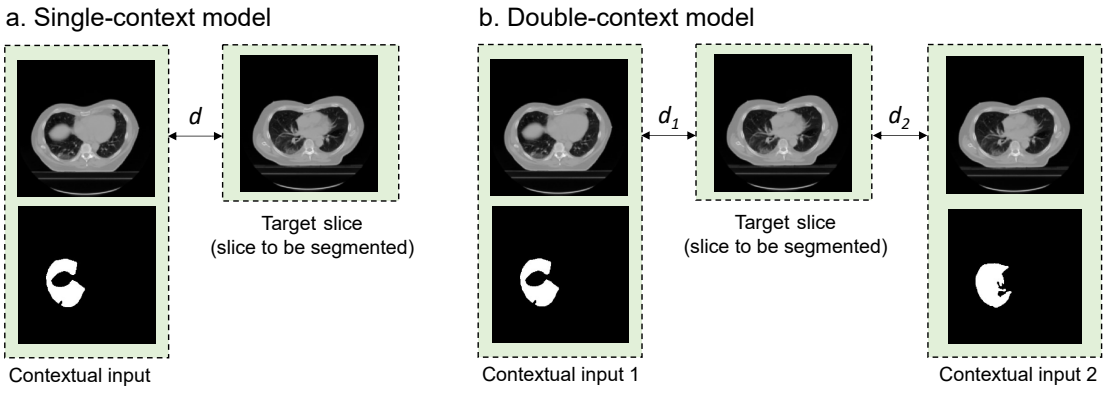


Figure 7.1: Illustration of different contextual input variants to the deep learning model. (a) Single-context deep learning: Target image slice and GT with corresponding image slice at a distance d from the target image slice (used for iterative prediction, Chapter 3 (b) Double-context model: two user inputs (contour and image slice) are provided to the deep learning model at a distance d_1 and d_2.

7.2.1.2 Data and preprocessing

All models were trained using the data introduced in Section 3.3, consisting of CT images and delineations of 19 different structures of interest from various openly available datasets [132]–[135].

7.2.1.3 Experimental details

Interpolation and extrapolation The model was evaluated for two user inputs: the first at 0.25 of the structure size and the second at 0.75 of the structure size, as illustrated in Figure 7.2a. The target slices were predicted using direct prediction. For evaluation, either the full structure was evaluated, or it was distinguished between the slices between two user inputs (interpolation region) and those that were outside of the two user inputs (extrapolation region). The double-context model was compared to the single-context model (Model ABCDEF, Chapter 3) when the same user inputs were used for iterative prediction with the propagation boundary at the middle between two user inputs. The extrapolation and interpolation region of the structure are illustrated in Figure 7.2.

Multiple input selection for interpolation model To test how well the double-context model works compared to the single-context model, all structures were evaluated using a different number of inputs. The inputs were selected such that the user inputs were the same distance d from each other and that the distance to the border of the structure was $d/2$, as illustrated in Figure 7.2. For the interpolation model, the closest two user inputs were selected to predict a target slice.

7.2.1.4 Evaluation measures

The quality of the segmentation was quantified using the 3D DSC [142], 95% HD [146] and relative APL [147], as described in detail in Section 3.3.1.6. To compute the performance measures, the masks that were generated were upsampled to the original 512x512 imaging resolution using bi-linear interpolation.

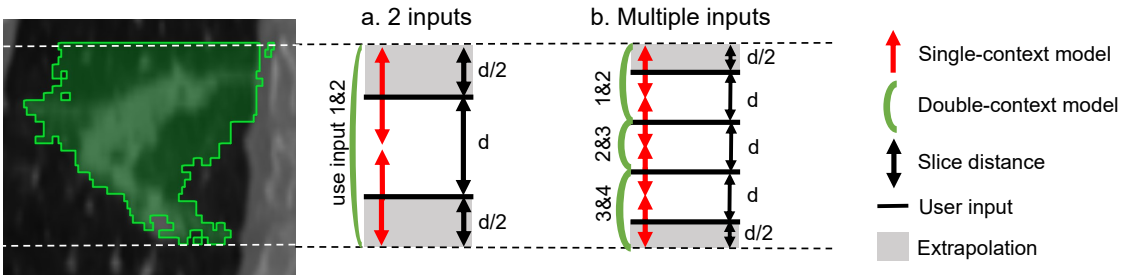


Figure 7.2: Segmentation using multiple user inputs: (a) using two inputs (b) using more than two inputs. For the single-context model (see Chapter 3 for details) the closest user input will be used as the starting slice for iterative prediction. For the double-context model the 2 closest user inputs are used for prediction. The image region between two user inputs is referred to as interpolation region, the region outside two user inputs is referred to as extrapolation region.

7.2.2 Results

7.2.2.1 Interpolation and extrapolation

The double-context and single-context model are compared in Figure 7.3 for the interpolation and extrapolation region of the structure. For all structures, the double-context model achieved significantly better results for interpolation, whereas it performed worse for the extrapolation region. For example, the relative APL for dataset G (Spleen) improved from 0.11 to 0.07 for interpolation, but increased from 0.43 to 0.46 for extrapolation. For all datasets, the interpolation region resulted in a lower APL compared to extrapolation for both segmentation approaches.

7.2.2.2 Hybrid-context model

The segmentation performance of the full structure across all datasets is compared in Table 7.1. In addition to the segmentation performance of the single-context and double-context approach, the table shows the performance of the hybrid model, which used the single-context approach for the extrapolation region, and the double-context model for the interpolation region. The hybrid model was best for all datasets investigated. It achieved an APL of 0.37 (A), 0.21 (B), 0.41 (C), 0.39 (D), 0.44 (E), 0.30 (F) and 0.25 (G) for the full structure.

Figure 7.4 compares the double-context hybrid model to the iterative prediction method, when given 2, 3, 4 or 5 equidistant inputs. The hybrid model performs

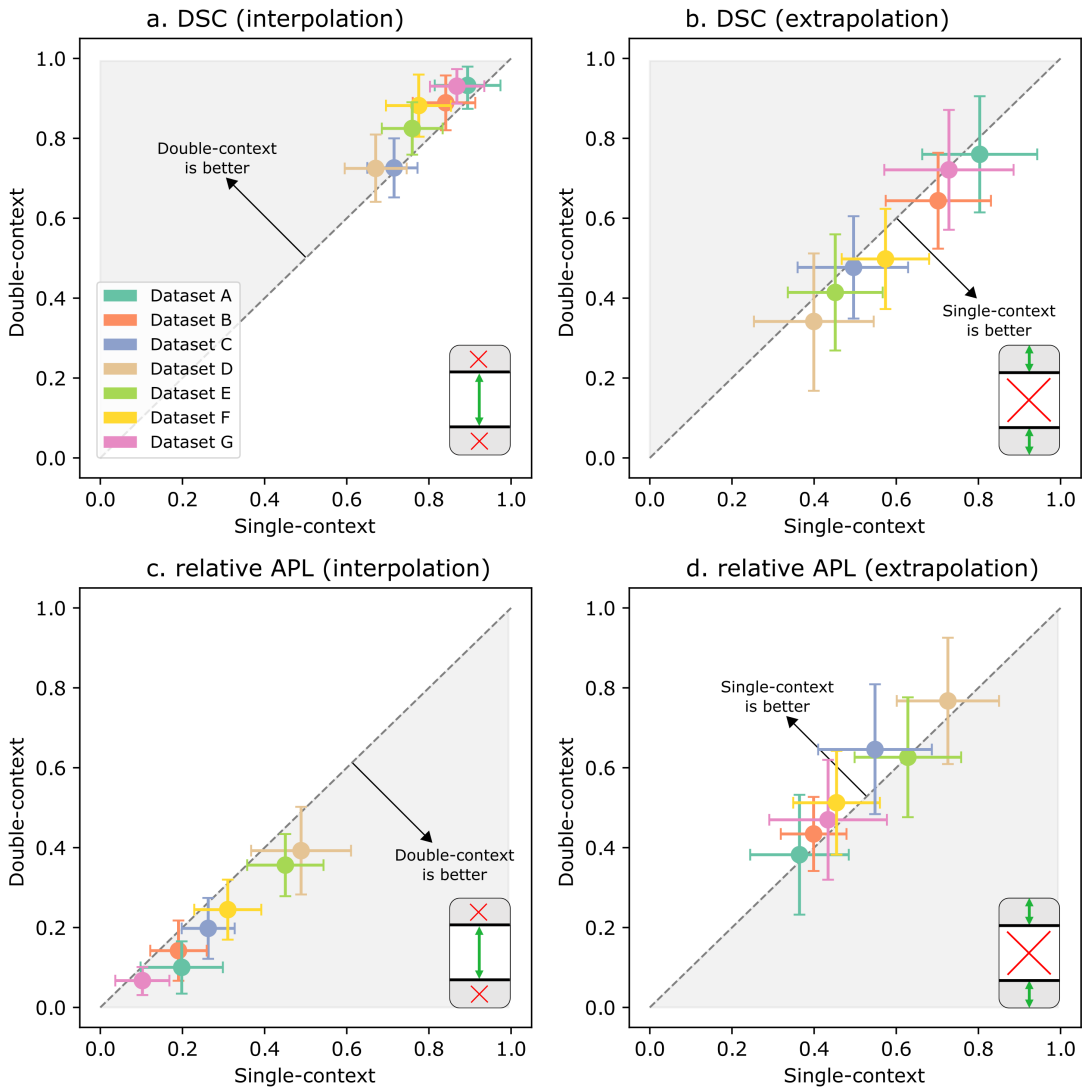


Figure 7.3: Performance of double-context and single-context model when using two user inputs. The *interpolation* region (a, c) corresponds to prediction of slices between two user inputs. The *extrapolation* region (b, d) refers to slices not bounded by 2 user inputs. Error bars indicate the standard deviation.

better for all structures evaluated.

A qualitative example of the single-context and hybrid approach is visualized in Figure 7.5, the single-context method with multiple inputs was sometimes unsatisfactory because it resulted in a discontinuous contour on the non-axial planes, where the propagations from two different input slices meet. But the hybrid model could adequately segment the structure.

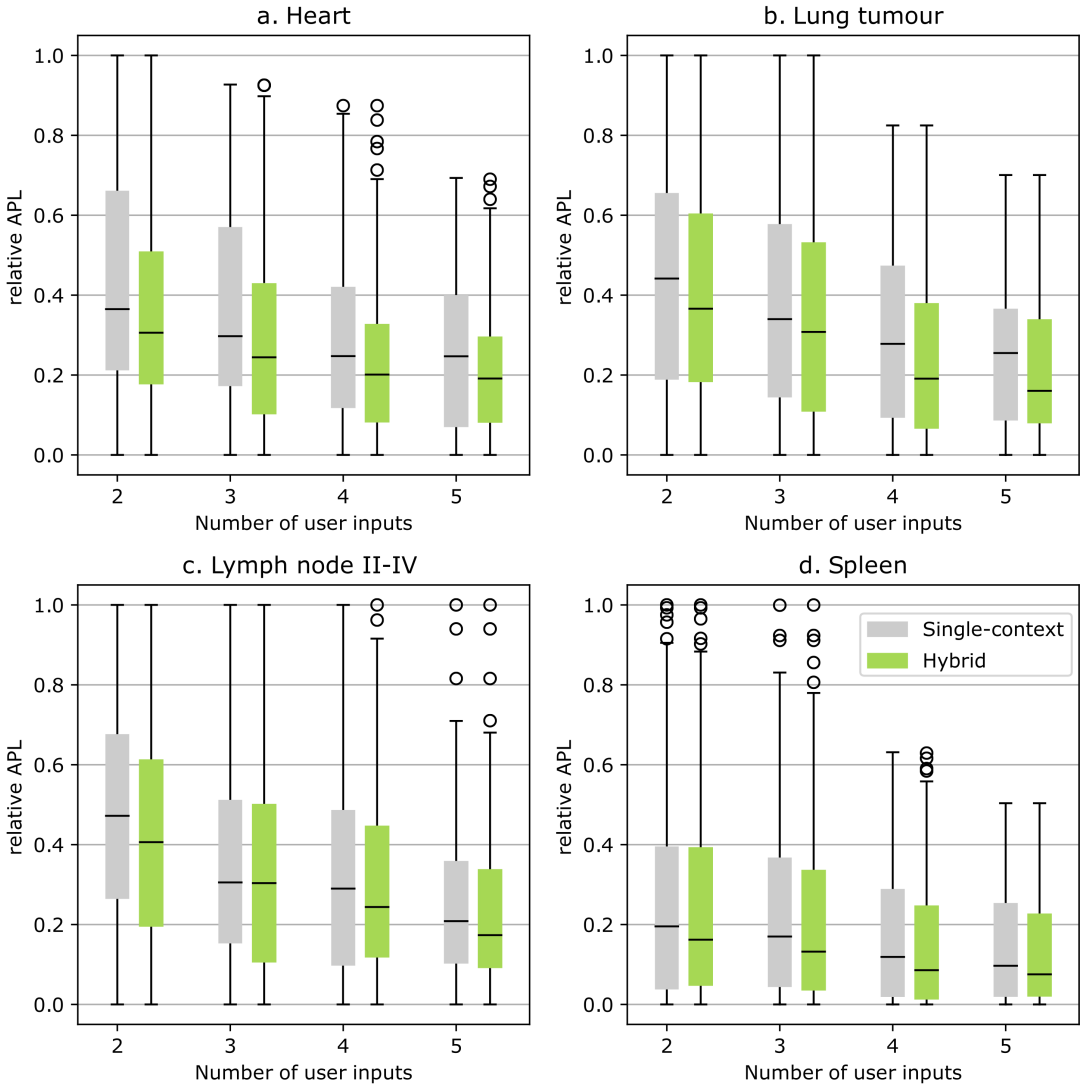


Figure 7.4: Slicewise relative APL of hybrid (green) and single-context approach (grey) when using 2, 3, 4 or 5 user inputs, shown for (a) heart, (b) lung tumour, (c) lymph node II-IV (left) and (d) spleen.

7.2.3 Discussion

7.2.3.1 Interpolation and extrapolation

The double-context model benefited from two user inputs if they were on either side of the slice to be segmented. In contrast, if the model did not interpolate but extrapolated, it performed worse. These results could be biased by the slice sampling during training compared to the single-context approach. For instance, the single-context approach used a different slice distance selection from the double-

Table 7.1: Comparison of segmentation performance (relative APL) of the whole structure (interpolation+extrapolation) when given 2 user inputs for 3 different prediction approaches. (1) Single-context (2) Double-context (3) Hybrid: propagation (extrapolation) + double-context (interpolation). Significantly better performing methods are highlighted in bold. Wilcoxon signed-rank test ($p<0.01$) was used.

Dataset	Single-context	Double-context	Hybrid
A	0.42	0.41	**0.37**
B	0.25	0.25	**0.21**
C	0.49	0.49	**0.41**
D	0.46	0.45	**0.39**
E	0.47	0.48	**0.44**
F	0.33	0.34	**0.30**
G	0.31	0.26	**0.25**

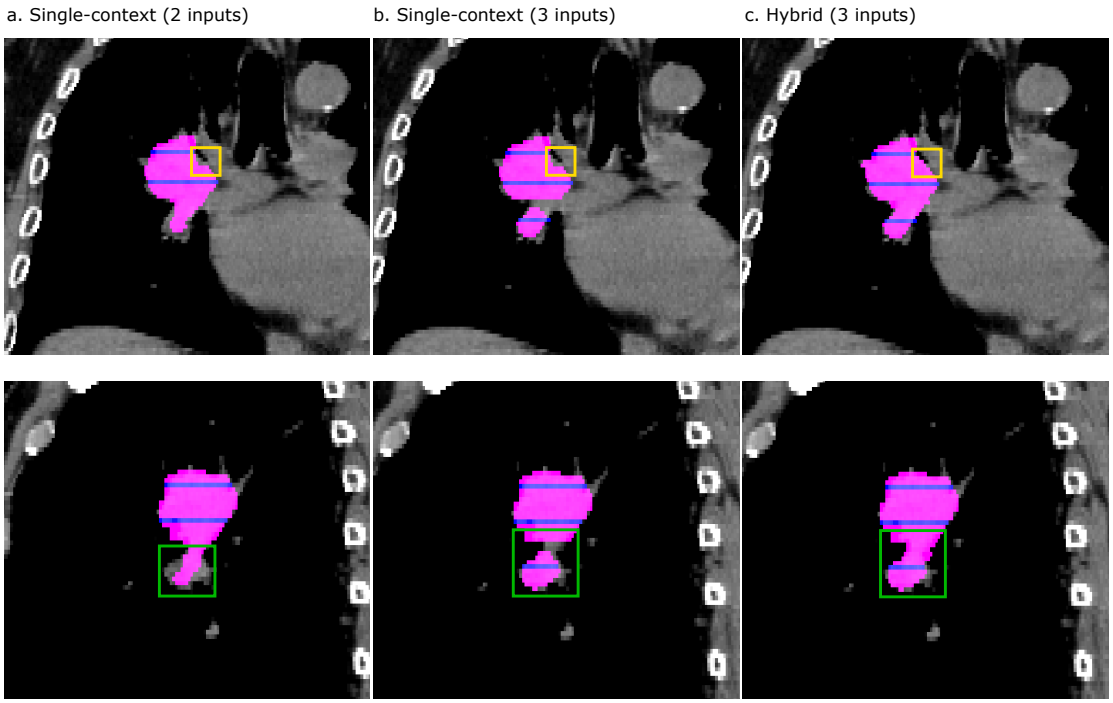


Figure 7.5: Qualitative comparison of single-context model and hybrid model when given multiple user inputs. (a) Single-context model when given two user inputs. The yellow box indicates a slight discontinuity in the segmentation, where the propagations from two different inputs meet. The inferior end of the tumour is not segmented correctly, as indicated by the green box. (b) If given a third input at the inferior end of the structure to correct for the segmentation error, the model fails to propagate towards the superior end and doesn't meet with the segmentation propagated from the other user input. (c) For the hybrid model, the slight discontinuity vanishes (yellow), and the area between two inputs on the inferior end is segmented correctly (green).

context model. For the double-context model, slices from the interpolation and extrapolation region were included in training and therefore, the model should not only be biased towards interpolation. The interpolation model did not however include slices outside of the structure, which previously was included in training for the single-context approach to ensure that the model learns when to stop contouring. This may have affected the model performance at the edge of the structure. However, the lower segmentation performance in the extrapolation region cannot only be attributed to the model not knowing when to stop segmenting, because the worse performance occurred in terms of DSC and APL. APL would not be affected by the model overshooting compared to the ground truth, as deleting contours on axial slices does affect APL.

In this thesis, all structures included in training are single connected structures and has not been evaluated for structures that are disconnected. The contextual deep learning model using iterative prediction will stop making predictions if the model encounters a slice that has no predicted contour. Therefore, it cannot pick up disconnected structures, if intermediate axial slices do not contain the structures of interest. This issue can be addressed using direct prediction for the contextual deep learning model. Direct prediction was also used for memory networks and the double-context model. Disconnected structures on the axial plane have also not been included in training and therefore may cause issues in practice. The structures in training were all connected and therefore, the location of the structure on the contextual input tends to be in close proximity on the axial plane on adjacent slices. If there is large shifts on the location of the structure between adjacent slices this may result in a poor segmentation output. This issue may be solved by employing translation for data augmentation between target slice and contextual input, as well as by introducing structures in training that are disconnected.“

7.2.3.2 Hybrid-context model

The single-context and double-context approaches performed similarly if evaluated on the whole structure. But each of the models consistently outperformed the other

for either interpolation or extrapolation. Therefore, they can easily be used for specific tasks and combined into one model, for which the extrapolation is performed using single-context and interpolation between user inputs using the double-context model. This resulted in an overall superior model compared to the single-context and double-context approaches when multiple user inputs are considered. The hybrid model was consistently better than iterative prediction or direct prediction even when considering the maximal improvement by propagation boundary selection or input selection which was investigated in the previous chapter (see Section 6.3.2).

A qualitative example for both segmentation approaches is visualized in Figure 7.5. The single-context method with multiple inputs sometimes offered unsatisfactory results because it resulted in a discontinuous contour on the non-axial planes, where the propagation from two different input slices meet. But the interpolation model can adequately segment the structure if multiple inputs were given. This example was chosen as an extreme case and in many cases the propagation from two sides leads to a non-obvious boundary. This may depend on the structure, similarity of the inter-slice region to the inputs on both sides and the error accumulation when making predictions. However, as was investigated in the previous chapter, there is no easy solution to deciding which side of the propagation is right or what propagation boundary should be chosen. Averaging of the contours is not an adequate solution, as the propagation error may be due to a topological change in the structure along one of the propagation directions.

7.2.4 Conclusion

The double-context model achieved better performance than the single-context model when multiple user inputs are given, but only for the slices in-between two user inputs. In this interpolation scenario, the model can leverage both inputs successfully. For extrapolation, the single-context model performed better. For this reason, both approaches were combined. The double-context approach is used for the region between user inputs, whereas the single-context approach is used otherwise, resulting in consistently better segmentation results.

7.3 Memory Networks

The hybrid model of the previous section successfully improved segmentation performance if multiple user inputs are used. That method does not directly weigh the user input's relevance for the target slice to be predicted, though some network layers may implicitly learn such a weighting. Factors such as distance to the slice to be predicted and image similarities to either input slice are only indirectly taken into account by the model. Weighting multiple inputs on how beneficial they are and making a prediction based on that may improve the contouring performance. This is possible using memory networks. Memory networks have been very successful in interactive video segmentation tasks. For this, one or more video frames of the video are manually annotated. Then the remaining frames are segmented by the model - a task similar to segmenting 3D medical images given a few contoured image slices.

In this section, the framework of memory networks is applied to medical imaging. First, an architecture successful in video segmentation is applied to the medical image segmentation task using fine-tuning. Second, a memory network with a U-Net-like architecture is proposed. It is hypothesized that the memory read operation can selectively retrieve information from nearby and spatially distant user interactions and thereby improve network performance if multiple user inputs are provided.

7.3.1 Materials and Methods

7.3.1.1 Memory networks

Memory networks refer to neural networks with external memory where information can be written and read to perform a specific task [113]. Memory networks were first used in Natural Language Processing [190]–[192] and have recently been successful in computer vision tasks [114]–[116].

The memory network architecture is illustrated in Figure 7.6 and is based on the works of Oh et al [113]. It consists of a memory encoder, query encoder, query decoder, and external memory. The memory encoder takes two inputs: the contoured image slice as well as the contour information. User input is 0 (background) and 1 (structure). The memory encoder may also store intermediate predictions of

the deep learning model. In that case, the contour information is represented as a single channel probability map between 0 and 1, given by the softmax output of the network prediction. The query encoder takes one input: the image slice to be contoured. Each encoder outputs two feature maps: key and value. These result from two parallel convolutional layers following the bottleneck network. They reduce the feature dimensionality for the key by a factor of 8 and the value by a factor 2. The network used in video segmentation uses a ResNet50 backbone [113].

The external memory stores memorable information, separately embedded, into key and value feature vectors [113]. Information relevant to a specific query is addressed by the key. The corresponding value is then returned. Here, image slices that have a user-created contour were placed in memory. The image slice to be segmented acts as the query. To predict a sequence of image slices, the probability maps of the predicted masks can also be added to memory to help with appearance changes and error accumulation. For pixel-wise prediction of the segmentation on each image slice, the query slice needs to access memorable information originating from different image slices. The information is accessed following the memory read operation proposed by Oh et al. [113]. This operation searches for relevant information in the external memory and weighs the relevance of memorable information from the different image slices and mask in memory. The memory read measures the similarities between all pixels of the query key map and the memory key map $f\left(k_i^Q, k_j^M\right)$, as proposed by Oh et al. [113] This is used to weight the memory values v_j^M. The weighted memory values and query values v_i^Q are then concatenated and fed to the decoder. This is given by

$$y_i = \left[v_i^Q, \frac{1}{Z} \sum_j f\left(k_i^Q, k_j^M\right) v_j^M \right], \tag{7.1}$$

where $Z = \sum_j f\left(k_i^Q, k_j^M\right)$ is a normalizing factor and $[\cdot, \cdot]$ denotes concatenation. The memory read operation using tensor operations is illustrated in Figure 7.7. If there are multiple image slices with associated segmentation masks (T user inputs), they are separately encoded in key and values and stacked, forming the external

memory.

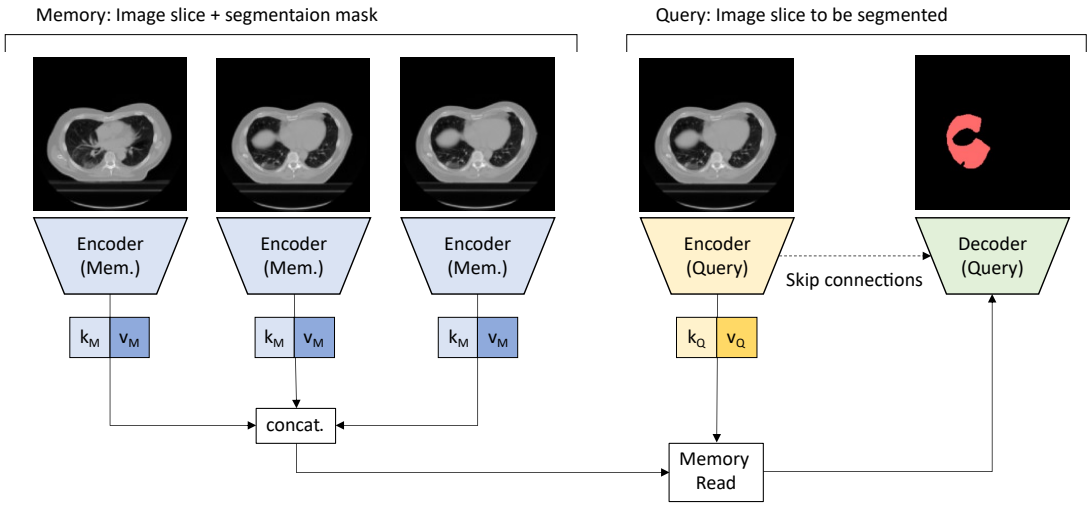


Figure 7.6: Memory network architecture. The query encoder (yellow) and decoder (green) are connected via skip connections. Additionally, The query decoder uses the memory read operation to retrieve information from the memory (memory encoder, blue).

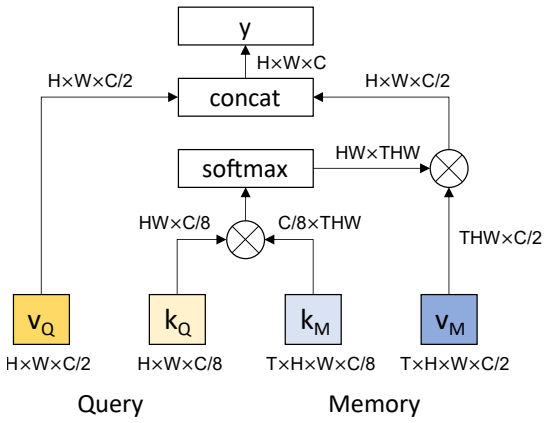


Figure 7.7: Illustration of memory read operation using tensor operations. $\otimes$ denotes the matrix inner-product. First, the inner-product of the query k_Q and memory keys k_M identifies local information for each memory input T with respect to the query key. The resulting weighting of relevant input (softmax) is applied to the memory values v_M. Concatenating of query values v_Q and weighted memory values concludes the read operation.

7.3.1.2 Data and preprocessing

The memory networks were trained and evaluated using the same set of data and preprocessing as in Section 3.3, including CT images and delineations of 19 different

structures of interest from various openly available datasets [132]–[135].

7.3.1.3 Training strategy

The network was trained in two stages: First on individual slices that were augmented, and second on a sequence of CT image slices.

Stage 1 The model was trained for up to 50 epochs using a randomly sampled image slice for 18 different structures and using early stopping. The image slice and corresponding segmentation mask were augmented using random affine matrix transformations consisting of translation (± 20 mm), rotation (± 20 degrees) and scaling ($\pm 10\%$). This generated a series of similar image slices and masks that simulated a series of slices. This was used to initialize the memory and query encoder. Similarly, previous works have used static images to simulate video sequences [193], [194].

Stage 2 For the main training, a series of 5 image slices was randomly sampled from the training case. The first image slice was randomly sampled from any slice that contains an annotation. The remaining 4 slices were sampled at random from the remaining image slices that contain contours and a small adjacent region that does not contain contours. The region not containing contours was restricted to the slices $\pm 10\%$ of the structure size on the inferior and superior regions of the structure. These unlabelled samples were included for the model to also learn when not to output a segmentation mask.

During training, the predicted segmentation mask at a previous step was added to the external memory. The probability map of the segmentation was used in the memory encoding to provide an uncertainty estimation of the prediction.

Downsampled 256x256 image slices were used for training. The batch size is set to 4. The Adam optimizer [141] was used to minimize the cross-entropy loss. The memory networks were trained on all structures, except the spleen, to determine if the architecture is able to perform denovo segmentation on structures excluded from training.

7.3.1.4 Experimental details

Different memory network architectures Two different architectures of memory networks were investigated: First, the network architecture proposed by Oh et al. [113] was used and fine-tuned on the medical imaging dataset. This architecture used a ResNet50 backbone for the query and memory encoder, and a refinement module [195] for a decoder. Second, a U-Net-like encoder and decoder was adopted for the memory networks, as this has shown to be successful for medical images in other applications. Therefore, the query encoder and decoder were used as described in Chapter 3, including attention gates for skip connections. These models were evaluated for one user input, using iterative prediction.

Evaluation with multiple user inputs The model performance was evaluated for two to five user interactions. User interactions provided can be stored into memory from the start. Here, it was assumed that the first interaction was in the middle of the structure. If more than one input was given, they were selected equidistant to each other. This is the same user input selection as for the double-context model discussed in the previous section and as was illustrated in Figure 7.2. On inference, only the provided user inputs were stored in the external memory. Intermediate predictions were not stored in memory.

Impact of user input on far away slices To evaluate if an extra input has additional effects on memory networks, two inputs were considered as illustrated in Figure 7.8. It was assessed whether the input of the second slice affects the performance in the half of the structure with no additional user input.

7.3.2 Results

7.3.2.1 Comparison of memory network architectures

In Table 7.2 the performance of the memory network with different architectures is compared to the double-context model. The memory network model performed

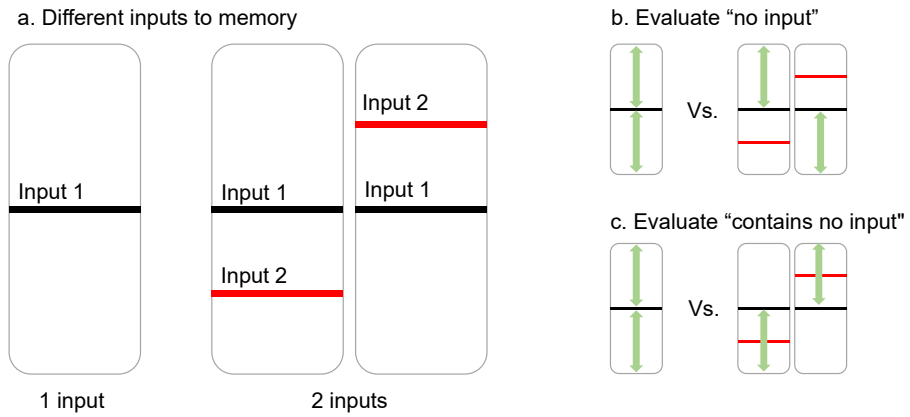


Figure 7.8: Experiment to evaluate the effect of far-away inputs on memory networks. (a) The first input is always at the center of the structure. The model is evaluated with just one or two inputs in memory. The second input is either in the upper or lower half of the structure. The model is evaluated for two scenarios: (b) Comparing the model with 1 input to the one with 2 inputs, for the halves where no additional inputs are given. (c) Comparing the model with 1 input to the areas where a second input was present.

worse than the double-context model for all datasets but dataset E (Liver). The hybrid approach performed best for all datasets.

Between the ResNet50 and U-Net memory network, the U-Net model achieved the same or better results compared to the fine-tuned ResNet50, for all but dataset D (Pancreas) and E (Liver).

Table 7.2: Comparison of segmentation performance (relative APL) for memory networks with ResNet50 backbone and U-Net-like architecture, to the double-context model and hybrid model introduced in Section 7.2. Significantly better performing methods are highlighted in bold. Wilcoxon signed-rank test ($p<0.01$) was used.

	Contextual Deep Learning		**Memory Networks**	
Dataset	Double-context	Hybrid	Resnet50	UNet
A	0.41	**0.37**	0.44	0.43
B	0.25	**0.21**	0.30	0.27
C	0.49	**0.41**	0.51	0.50
D	0.45	**0.39**	0.47	0.48
E	0.48	**0.44**	0.48	0.49
F	0.34	**0.30**	0.35	0.35
G	0.26	**0.25**	0.33	0.32

7.3.2.2 Impact of multiple user inputs

The performance of the memory network (U-Net) given equidistant multiple user inputs is shown in Figure 7.9 for the heart, lung tumour, lymph node (level II-IV) and spleen. The performance was compared to the performance of the double-context approach. The memory network performed worse than the hybrid model from the previous section for all structures for all number of inputs. The APL for the spleen after five user inputs was about twice as high for the memory network (0.21) as it was for the interpolation model (0.10) with the same number of user inputs.

The performance of the memory network slice-by-slice for selected input positions is illustrated in Figure 7.10 for two example structures. If a user input was placed at the center of the structure, the model performance was better close to the user input and dropped towards the edge of the structure. If an additional input was placed in the inferior half of the structure, the model performance increased in the vicinity of the second user input. But the additional user input did not affect segmentation performance beyond the first input, in the superior half of the structure.

The impact of an additional user input to one side of a first user input is shown in Figure 7.11. This shows that across all datasets, the model performance was improved in the area where the additional input is placed. However, in the area beyond the first user input, which did not include a second user input, the model performance did not significantly change compared to the prediction with just the initial user input.

7.3.3 Discussion

7.3.3.1 Comparison of memory network architectures

Neither the fine-tuned ResNet50 memory network nor the U-Net-based memory network achieved better segmentation performance than the double-context or hybrid approach for two user inputs.

Between the two memory network architectures the U-Net architecture achieved a lower APL than ResNet50. U-Net-like architectures have been among the best performing architectures in many medical applications because they are well suited

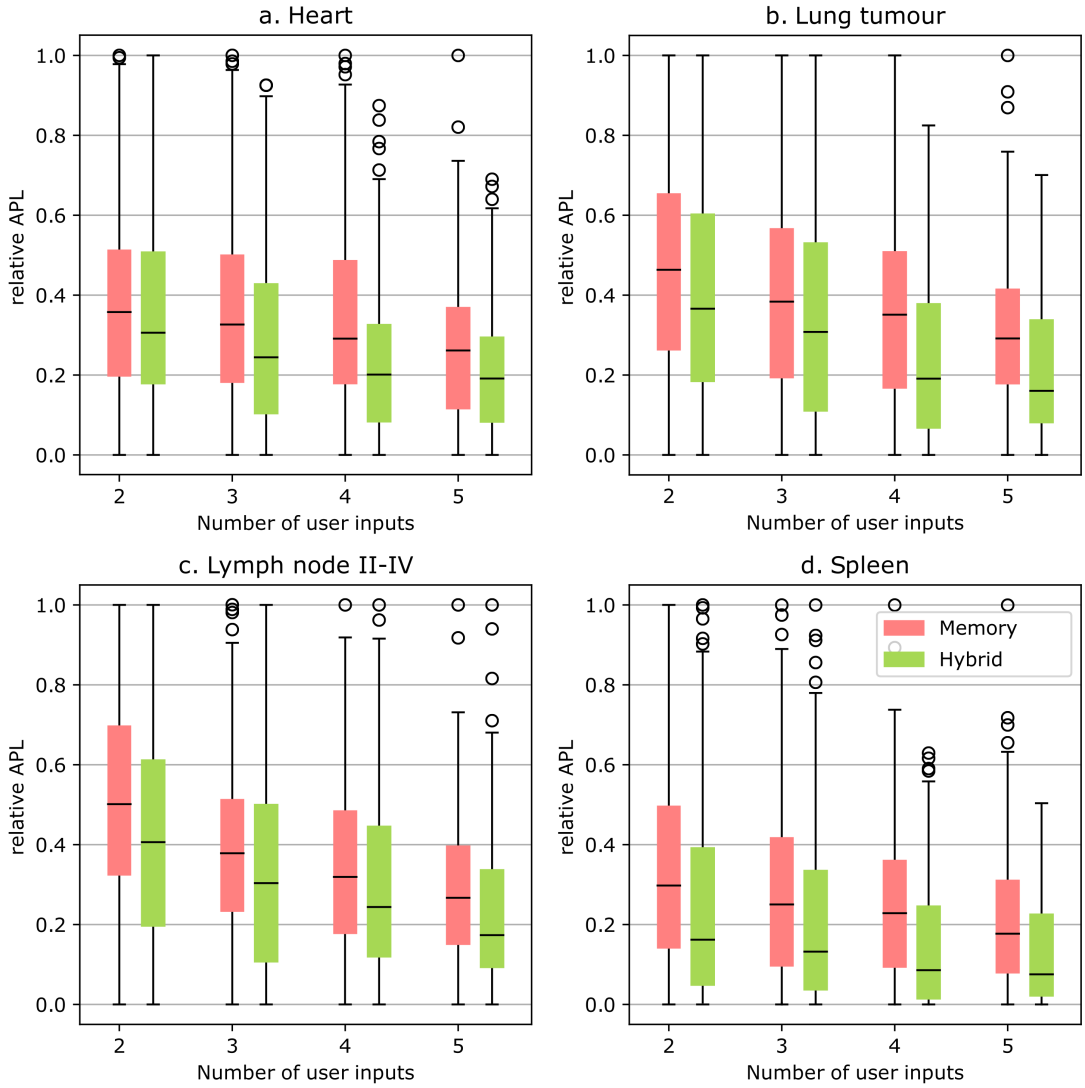


Figure 7.9: Slicewise relative APL of memory network (red) and double-context (green) when using 2, 3, 4 or 5 user inputs, shown for (a) heart, (b) lung tumour, (c) lymph node II-IV (left) and (d) spleen.

for medical image segmentation tasks. The down- and up-sampling of image features combined with the skip connections allow the network to combine high and low level features without loosing spatial information that is important for the segmentation task.

The proposed U-Net-like architecture also had fewer parameters than the ResNet50 reported in literature [113] which may make it more robust to overfitting when provided with fewer labelled images. Training data for medical images is typically more scarce than for natural images. The memory network with U-

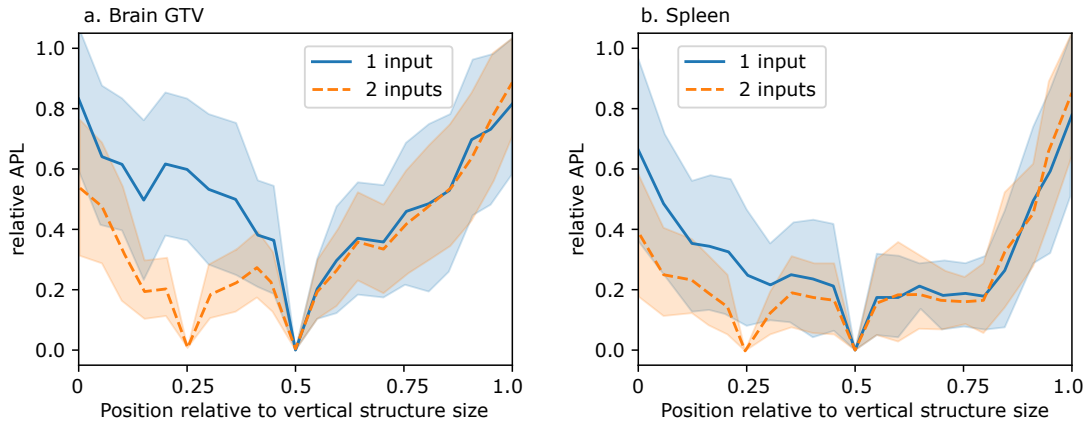


Figure 7.10: Comparison of the segmentation throughout the structure if the memory network is given a single central input (blue line) or a second input in one half of the structure (orange-dashed line): (a) Brain GTV and (b) Spleen. The shaded area indicates the standard deviation of the APL for different test samples.

Net-like encoder had 12 Mio. parameters, whereas the original ResNet50-based architecture had 17 Mio. parameters.

Another downside of the original ResNet50 architecture was that the refinement model of the decoder estimated the mask in 1/4 scale of the input image. The bilinear upsampling to match the input image size may result in additional error. The hybrid model used bilinear upsampling, but the segmentation masks were only downscaled half from the original image size. Furthermore, the ResNet50 architecture was deeper, using more downsampling in the feature space (32-fold, compared to 8-fold in the U-Net), this may result in too small a feature space for the model to retrieve useful information from the external memory. This is an even greater problem when segmenting small structures such as the spinal cord and esophagus, where the ResNet50 architecture performed particularly poorly (these structures are included in dataset B), but the network still managed to segment larger structures. The lower downsampling in the U-Net approach may have contributed to the improved performance.

The memory network with U-Net-like architecture (12 Mio.) still had more parameters than the double-context model (8 Mio.), due to the separation of memory and query encoder. The larger number of parameters may lead to over-fitting and poorer performance for the spleen compared to the double-context model. The

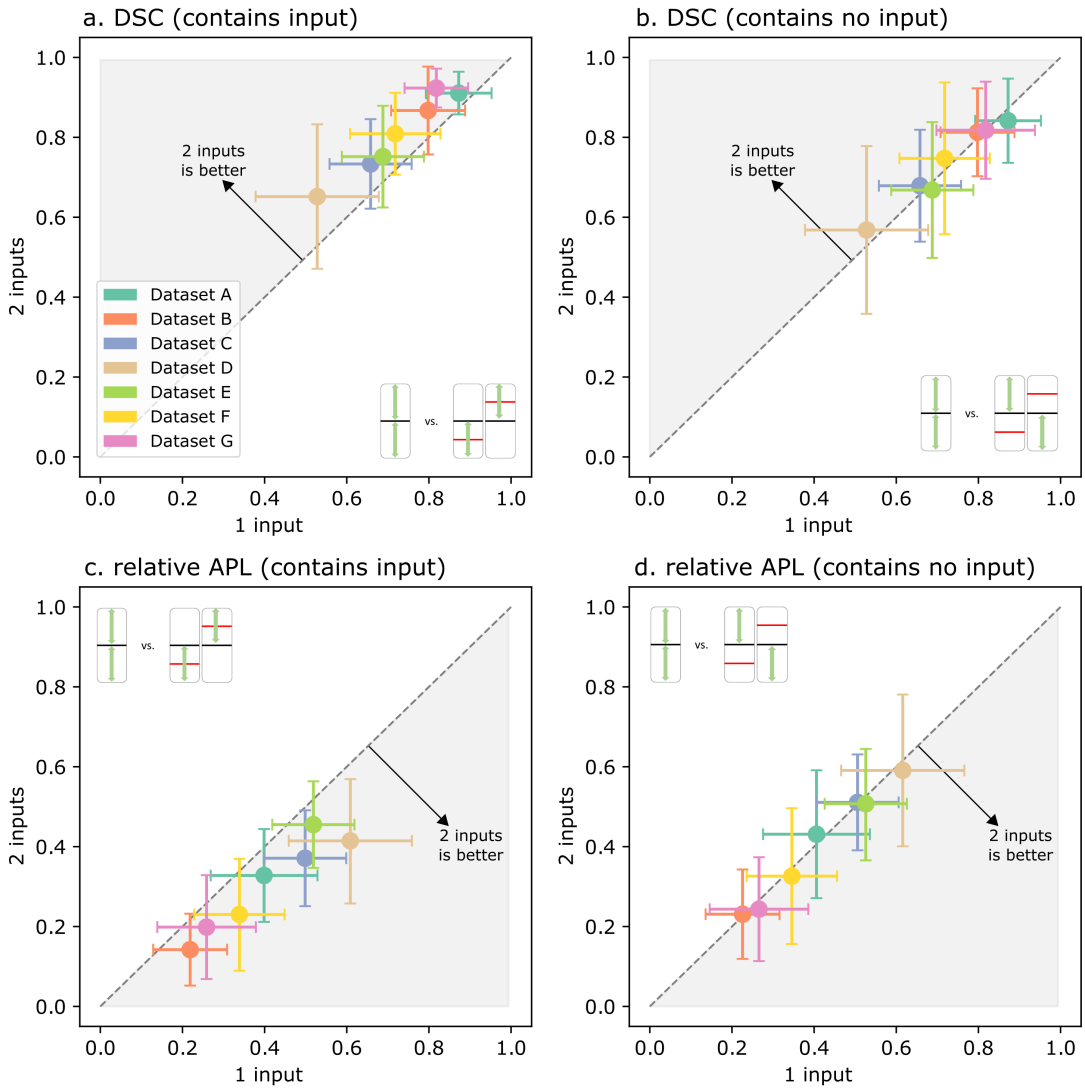


Figure 7.11: Comparison of the segmentation performance of the memory network when either one or two user inputs are given, shown for DSC (a, b) and relative APL (c,d). In the left column, only the halves where an additional input was present are evaluated if two inputs are given. On the right, the opposite halves are evaluated, meaning the half of the structure that did not contain the second user input. Error bars indicate the standard deviation.

memory network also performed worse than the double-context model for structures included in training, although to a lesser extent. The number of parameters may be reduced by sharing weights between the memory and query encoder in the memory network structure. Sharing weights would reduce the number of parameters, but the memory and query encoder serve different purposes. Therefore, sharing weights can help against over-fitting but may result in poorer segmentation performance. The memory encoder is trained to recognize features that identify what to segment and stores these in keys and value pairs, whereas the query encoder focuses on retrieval from memory. Studies of parameter sharing between memory and query encoder for interactive video segmentation have shown that sharing weights reduces model performance slightly, but the number of parameters required in the model can almost be halved [196]. This may have a positive impact on applications in the medical field, where less training data is available.

The training set itself was less diverse when trained on medical images. The video challenge dataset has millions of videos and thousands of object classes to train on. Several different medical imaging datasets and structures were used in this work. However, it was not possible to match the diversity of the training set in the video segmentation dataset. The lack of data diversity was mitigated by employing data augmentation. Still, the reduced training set diversity compared to natural images may impede the performance of memory networks when applied to interactive medical image segmentation.

7.3.3.2 Impact of multiple user inputs

The memory networks performed worse than the interpolation model even if multiple inputs (>2) were given. The appeal of the memory networks was, that the memory read operation would result in the model being able to retrieve relevant user input information from any user interaction when it is relevant to the segmentation. However, even when more than two inputs were considered, the interpolation model performed better than the memory network model. A reason for this may be that far away inputs, compared to a closer one, do not help with segmentation. This

was also observed for the double-context model. Two user inputs were beneficial if the slice to be segmented was between those two inputs. But two inputs performed worse than just one input, if the model was required to predict image slices when the target slice was in the extrapolation region of the structure.

For the memory network, extra inputs did not affect the segmentation performance beyond the closest input slice. This either means that the memory retrieval is not successful or that the far away image slices have little effect on segmentation performance. It was shown previously, that for direct prediction using contextual deep learning, sometimes far away input slices can be better inputs than nearby slices for certain input combinations.

Another reason why the model did not perform well with multiple inputs may be because the model was not trained with multiple inputs in memory. It was trained by using an initial input, and storing the predictions for adjacent slices in memory as well. This simulated a user input and also made the model learn confidence in its prediction.

7.3.4 Conclusion

Memory networks have been very successful when applied to sequential tasks such as natural language processing and video object segmentation. In this study, however, they did not perform better than the hybrid-context model which was introduced in the first section of this chapter. The anticipated benefit of memory networks was to use multiple user inputs more efficiently; but here it was shown, that the segmentation only improved in the vicinity of a user input in memory. If an additional user input is given, then it does not improve segmentation performance beyond another user input that is closer to the slice to be segmented.

Overall, this network performed worse than the double-context model (Section 7.2) and about as well as the single-context model. The memory network performance was improved by replacing the ResNet50 backbone (encoder) and refinement module (decoder) used in video segmentation with a U-Net-like architecture, which is more suited for medical image segmentation.

New advances in memory network research, such as using spatial priors or weight sharing between memory and query encoder [196], [197], could also improve this network type for the task of medical image segmentation.

In future work, focusing on interactive deep learning networks that can leverage multiple user inputs remains important. It has been shown that clinicians interact a lot with the produced segmentation of the contextual deep learning network (see Chapter 4), but also for structures produced by fully automatic segmentation tools [157]. Using all user interactions to update the segmentation in real time could further reduce segmentation time.

8 Conclusion and Future Outlook

The aim of this thesis was to investigate how interactive deep learning can help in the clinical contouring workflow. In this chapter, the overall conclusions of the work are summarised and compared to the aims of the work outlined in the Introduction, Section 1.2:

(1) Identifying a structure of interest based on user input

(2) Generalizable interactive deep learning

(3) Impact on segmentation quality and contouring workflow

(4) Guiding user interaction

(5) Effectively utilizing multiple user inputs

In addition, various areas of future work, which were identified over the duration of the research, are summarized.

8.1 Conclusions

In Chapter 2, previous work relevant to interactive contouring and deep learning was reviewed. Deep learning has already had a profound impact on medical image segmentation tasks. However, this impact is mostly restricted to fully automatic contouring approaches. Research on how deep learning can benefit from user interaction has received less attention. Few-shot learning is one approach in which

user input and deep learning can be combined to create segmentations for structures where no automatic solutions exist. These methods to date, however, have fallen short compared to other segmentation methods, though they are very versatile and segmentation of a structure given just one fully labelled example is possible.

It remains challenging to create a generalizable interactive deep learning solution. For instance, different users may interact with the model differently, and this variance in input may inhibit a model from making an appropriate prediction.

To date, neither the users nor the trainers of deep learning models understand fully how a model makes its predictions or learns how to make predictions. Understandable and interpretable AI has received much attention recently, particularly with respect to the decision-making processes that come with using AI in medicine. A few avenues to interpreting deep learning models have been reviewed. For example, the model's attention can be visualized in several ways. This may help indicate which image region is deemed important and can help spot whether the model is making sensible decisions.

The experimental work of this thesis was contained in Chapters 3, 4, 5, 6 and 7. In summary, a deep learning approach for interactive contouring was introduced, and an evaluation by clinicians was conducted. After that, the model was applied to a specific dataset and investigated with respect to multiple user inputs.

Chapter 3 introduced a model that can contour arbitrary structures of interest, even if the model has not been trained on this specific structure. First, it was shown, that a deep learning model can segment a structure of interest if adequate contextual information is provided (aim 1). Next, a novel training strategy was devised to enable such a generalizable interactive model (aim 2). The contextual deep learning model used a user-provided contour at one slice of a 3D medical image, which was then propagated throughout the remaining image slices. Iterative prediction demonstrated lower error accumulation than direct prediction when used for unseen structures in the training set.

In Chapter 4, the contextual deep learning model was tested by clinicians in form of a preliminary and extended evaluation (aim 3). The deep learning tool was found to decrease active contouring time by 33% (preliminary evaluation) and 23% (extended evaluation) compared to manual contouring. Edits were still necessary on most slices even when using the deep learning prediction tool. These were smaller than when using the linear interpolation tool that is normally available when manually contouring. It was also found that observation time - decision-making time and time spent interacting with the user interface - made up approximately 80% (preliminary evaluation) and 70% (extended evaluation) of the total time spent contouring. This indicates that much emphasis should be put on providing an intuitive user interface and potentially guiding the user's attention and interactions when developing clinical tools.

In Chapter 5, the contextual deep learning tool was tested on lung fibrosis lesions, which are a difficult structure to segment and subject to very high inter-observer variability. The model demonstrated the ability to make predictions based on a user interaction given by the clinician (aim 2). For highly variable structures, such as lung fibrosis, this may mean higher inter-observer variability. However, the tool was shown to be able to adapt to the structure the user intended to segment. The high inter-observer variability may largely be due to the lack of guidelines with respect to lung fibrosis segmentation. Task-specific fine-tuning of the contextual deep learning model showed that, given even a small training set, the segmentation performance can be improved.

In Chapter 6, different aspects of how to handle multiple user inputs in an interactive deep learning model were addressed. First, a framework to guide user interaction for contextual deep learning was proposed (aim 4). It was shown that APL can be reduced if an active learning-based query approach is used to select the next slice to contour, as compared to selecting a random next slice. The query approaches investigated were based on model uncertainty (entropy, MC dropout), sample representativeness (mutual information) and a self-learned query

approach (auxiliary network). Particularly, sample representativeness (mutual information) demonstrated to be a good indicator for a suitable next slice to contour to maximize the segmentation performance. A correlation was found between a slice of bad segmentation and a good slice to contour next. This means that predicting poor segmentation and predicting where to contour next go hand-in-hand. While most query approaches performed better than random selection, they were still inconsistent for given structures in predicting the next best slice to contour. Therefore, they should not be implemented to prompt the user as to which slice should be contoured next, but they could be used to highlight areas in the scan which might require the clinician's attention.

Second, the iterative and direct prediction approaches were optimized for multiple user inputs by selecting the propagation boundary between inputs or selecting which user input to use to make a prediction. Using sample representativeness, the contextual deep learning performance was slightly improved when multiple user inputs were given. However, these improvements were not large compared to the baseline performance.

Chapter 7 proposes two approaches to handling multiple user inputs using deep learning. A double-context model was introduced that is an extension of the contextual deep learning model using two instead of one contextual input. This model successfully improved segmentation performance for the region between two user inputs but performed slightly worse compared to the contextual deep learning model when used to extrapolate outside the user inputs. Therefore, a combined model was proposed that uses the single-context model for extrapolation, and the double-context model in-between two user inputs, resulting in an improved segmentation performance for all datasets investigated (aim 5). Qualitatively, the double-context model solved the problem of occasionally discontinuous segmentations on the sagittal and coronal plane, which occurred at the propagation boundary when using iterative prediction.

Finally, memory networks were explored. Through an external memory and a memory read operation, they have the potential to select areas of each user input which are most important for the segmentation of a target slice. While they did not perform better than the double-context model, a significant improvement was possible by adjusting the original architecture for video segmentation to a U-Net-like architecture, which was more suitable for medical image segmentation (aim 5).

Interactive deep learning models, such as the ones investigated in this thesis, can result in faster de-novo delineation of structures for which no automatic tools exist, or can provide a tool for editing the contours received from an auto contouring solution. This makes them valuable tools to improve the healthcare given to patients.

8.2 Future Outlook

As indicated throughout the thesis, various areas for future work were identified over the duration of the research which are summarised below.

The evaluation by clinicians showed that observation time makes up a very large part of overall contouring time. Observation time, as defined in this thesis, can be split into two core components: (1) Time interacting with the user interface and (2) time making decisions on where to contour. Separating these two is not possible without a detailed visual analysis of the participants (e.g. eye-tracking), which was not part of the analysis in Chapter 4. Interaction time with the user interface may be reduced by closely studying how clinicians interact with the user interface and developing tools that naturally integrate into their contouring practices. This is difficult as clinicians and clinical sites may follow different protocols and guidelines. In research, user interface design is often neglected, and therefore, optimizing user experience should receive a lot of attention when developing an application used in clinical practice. The decision-making time was addressed in this thesis by attempting to predict which slice should be contoured next to minimize future interactions. While this was only partially successful, the large amount of

observation time that was recorded in the clinical study shows the potential of tools that guide the user's attention. Helping to focus the clinician's attention can save time, lessen the mental effort needed for contouring, or be used to help train new doctors in an interactive way on where to look when contouring difficult structures.

Chapter 5 investigated how fine-tuning affects the contextual deep learning model. It has been demonstrated that model performance significantly increases even if trained only on a small dataset and retaining the ability to predict what the user intends to contour based on a given user input rather than segmenting based on specific learnt features only. This still distinguishes it from fully automatic solutions.

Fine-tuning also opens up the opportunity to use the contextual model as a continual learning method. Continual learning may lead to a gradual improvement in segmentation performance, if more and more structures are segmented and the labelled dataset for fine-tuning increases. However, it needs to be investigated if the model is still able to generalize to other structures after fine-tuning or can then only be employed for specific tasks. Alternatively, the contextual deep learning model could be retrained using the new dataset, but including training examples from previously seen structures to avoid the model forgetting how to segment other structures and importantly to remain able to generalize.

In Chapter 6, user-guidance was proposed to decrease the number of interactions needed for contouring a structure of interest. Better than random selection was achieved if any of the proposed query approaches were used to predict which slice to contour next. However, to evaluate the clinical impact of such a feature, the simulated impact of user-guidance is not enough. A designated evaluation by clinicians would be necessary to study clinicians' contouring behaviour with such a user-guided system.

Memory networks (Chapter 7) were hypothesized to be able to leverage multiple user inputs simultaneously by learning from different user inputs, as well as selected image regions. While this was not successful in this application to medical images,

the framework holds potential for other aspects of medical imaging. The external ledger that uses key and value pairs to retrieve relevant information from memory could also be a useful tool to evaluate sample representativeness. Instead of assessing representativeness of the pure image or hand-crafted features, the feature space can be used to evaluate the information gain for an additional user input, by comparing all remaining image slices with the information stored in memory. Thus, the diversity of images stored in memory could be maximized, and the predicted segmentations for the remaining unlabelled image slices could be improved. Additionally, memory networks hold opportunities when it comes to estimating or learning model uncertainty. During model training, model predictions may be stored in memory with their softmax output rather than a binary mask; the softmax output functioning as a proxy for model uncertainty. By training the model to retrieve information from memory that is associated with uncertainty, it may be able to better judge the uncertainty in its own prediction. If the model is able to produce accurate uncertainty estimates, these estimates could be used to guide the clinician to slices associated with high uncertainty and potentially reduce time spent contouring.

Future research should also focus on new methods to combine artificial intelligence and clinical expertise, instead of focusing on one or the other. A clinician can point out segmentation errors to the model, but the model may also communicate areas of high and low uncertainty to the clinician. By combining the strength of artificial intelligence and clinical expertise, patient care can be improved; elevating contour consistency and quality, and reducing the time-spent contouring thus freeing up clinicians to focus on direct patient care.

Data Sources and Material Transfer Agreement

All data used in this thesis was fully anonymised. The data originating from *The Cancer Imaging Archive* (TCIA) was used under Creative Commons licences: Yang et al. (CC BY 3.0) [132], Aerts et al. (CC BY 3.0) [133], Simpson et al. (CC-BY-SA 4.0) [134], Wee et al. (CC BY-NC 3.0) [135], and Li et al. (CC BY 4.0) [154]. The PET/CT data for the evaluation by clinicians was provided by the *Centre Hospitalier Universitaire de Liege* and used under a Material Transfer Agreement between the Centre Hospitalier Universitaire de Liege and Mirada Medical Ltd.

Statement of Originality

I hereby declare that this submission is my own work and, to the best of my knowledge, it contains no materials previously published or written by another person, or substantial proportions of material which have been accepted for the award of any other degree or diploma at the University of Oxford or any other educational institution, except where due acknowledgement is made in the thesis.

Any contribution made to the research by others, with whom I have worked at the University of Oxford or elsewhere, is explicitly acknowledged in the thesis.

I also declare that the intellectual content of this thesis is the product of my own work, except to the extent that assistance from others in the project's design and conception or in style, presentation and linguistic expression is acknowledged.

Michael Johann Trimpl, March 2023

References

[1] M. J. Trimpl, D. Boukerroui, E. P. J. Stride, K Vallis, and M. J. Gooding, "Interactive contouring through contextual deep learning," *Annual meeting of the American Association of Physicists in Medicine (AAPM)*, vol. PO-GeP-M-263, 2020.

[2] M. J. Trimpl, D. Boukerroui, E. P. Stride, K. A. Vallis, and M. J. Gooding, "Interactive contouring through contextual deep learning," *Medical Physics*, vol. 48, pp. 2951–2959, 6 2021.

[3] M. J. Trimpl, P. Charlton, S. Teoh, *et al.*, "Clinical evaluation of an interactive deep-learning assisted contouring method for target contouring," *Annual meeting of the European Society for Radiotherapy and Oncology (ESTRO)*, vol. PO-1164, 2021.

[4] M. J. Trimpl, P. Salome, D. Walz, *et al.*, "Task-specific fine-tuning for interactive deep learning segmentation for lung fibrosis on ct post radiotherapy," *Annual meeting of the American Association of Physicists in Medicine (AAPM)*, vol. SU-E-BRB-8, 2022.

[5] M. J. Trimpl, S. Primakov, P. Lambin, E. P. Stride, K. A. Vallis, and M. J. Gooding, "Beyond automatic medical image segmentation-the spectrum between fully manual and fully automatic delineation," *Physics in medicine and biology*, vol. 67, 12 Jun. 2022.

[6] M. J. Trimpl, G. Higgins, E. P. J. Stride, K. A. Vallis, and M. J. Gooding, "In preparation: The impact of interactive deep learning on the clinical contouring workflow," *Physics and Imaging in Radiation Oncology*, 2023.

[7] M. J. Trimpl, P. Salome, D. Walz, *et al.*, "In preparation: Task-specific fine-tuning from limited data - optimizing contextual deep learning to segment radiation induced lung fibrosis," *Medical Image Analysis*, 2023.

[8] A. Ramkumar, J. Dolz, H. A. Kirisli, *et al.*, "User interaction in semi-automatic segmentation of organs at risk: A case study in radiotherapy," *Journal of Digital Imaging*, vol. 29, pp. 264–277, 2 Apr. 2016.

[9] T. Lustberg, J. van Soest, M. Gooding, *et al.*, "Clinical evaluation of atlas and deep learning based automatic contouring for lung cancer," *Radiotherapy and Oncology*, vol. 126, pp. 312–317, 2 Feb. 2018.

[10] D. Jarrett, E. Stride, K. Vallis, and M. J. Gooding, "Applications and limitations of machine learning in radiation oncology," *British Journal of Radiology*, vol. 92, p. 20 190 001, 1100 Jun. 2019.

[11] M Gooding, A Smith, D. Peressutti, *et al.*, "Pv-0531: Multi-centre evaluation of atlas-based and deep learning contouring using a modified turing test," *Radiotherapy and Oncology*, vol. 127, S282–S283, 2018.

[12] C. S. Perone and J. Cohen-Adad, "Promises and limitations of deep learning for medical image segmentation," *Journal of Medical Artificial Intelligence*, vol. 2, 0 2019.

[13] A. S. Lundervold and A. Lundervold, "An overview of deep learning in medical imaging focusing on mri," *Zeitschrift fur Medizinische Physik*, vol. 29, pp. 102–127, 2 May 2019.

[14] S. D. Olabarriaga and A. W. M. Smeulders, "Interaction in the segmentation of medical images: A survey," *Medical Image Analysis*, vol. 5, pp. 127–142, 2 2001.

[15] G. Wang, W. Li, M. A. Zuluaga, *et al.*, "Interactive medical image segmentation using deep learning with image-specific fine tuning," *IEEE Transactions on Medical Imaging*, vol. 37, pp. 1562 –1573, 7 2018.

[16] G. Wang, M. A. Zuluaga, W. Li, *et al.*, "Deepigeos: A deep interactive geodesic framework for medical image segmentation," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 41, pp. 1559–1572, 7 2019.

[17] T. Sakinis, F. Milletari, H. Roth, *et al.*, "Interactive segmentation of medical images through fully convolutional neural networks," *ArXiv*, vol. abs/1903.0, 2019.

[18] J. Léger, E. Brion, U. Javaid, J. Lee, C. D. Vleeschouwer, and B. Macq, "Contour propagation in ct scans with convolutional neural networks," *International Conference on Advanced Concepts for Intelligent Vision Systems ACIVS 2018: Advanced Concepts for Intelligent Vision Systems*, pp. 380–391, 2018.

[19] Q. Zheng, H. Delingette, N. Duchateau, and N. Ayache, "3d consistent robust segmentation of cardiac images by deep learning with spatial propagation," *IEEE Transactions on Medical Imaging*, vol. 37, pp. 2137–2148, 2018.

[20] A. Novikov, D. Major, M. Wimmer, D. Lenis, and K. Bühler, "Deep sequential segmentation of organs in volumetric medical scans," *IEEE Transactions on Medical Imaging*, vol. 38, pp. 1207–1215, 5 Jul. 2018.

[21] M. J. Willemink, W. A. Koszek, C. Hardell, *et al.*, "Preparing medical imaging data for machine learning," *Radiology*, vol. 295, pp. 4–15, 1 2020.

[22] R. Wang, T. Lei, R. Cui, B. Zhang, H. Meng, and A. K. Nandi, "Medical image segmentation using deep learning: A survey," *IET Image Processing*, vol. 16, pp. 1243–1267, 5 Sep. 2022.

[23] H. Seo, M. B. Khuzani, V. Vasudevan, *et al.*, "Machine learning techniques for biomedical image segmentation: An overview of technical aspects and introduction to state-of-art applications," *Medical physics*, vol. 47, pp. 148–167, 5 Jun. 2020.

[24] N. Tajbakhsh, L. Jeyaseelan, Q. Li, J. N. Chiang, Z. Wu, and X. Ding, "Embracing imperfect datasets: A review of deep learning solutions for medical image segmentation," *Medical Image Analysis*, vol. 63, p. 101 693, Jul. 2020.

[25] D. Shen, G. Wu, and H. I. Suk, "Deep learning in medical image analysis," *Annu Rev Biomed Eng*, vol. 19, pp. 221–248, Jun. 2017.

[26] K. Suzuki, "Overview of deep learning in medical imaging," *Radiological physics and technology*, vol. 10, pp. 257–273, 3 Sep. 2017.

[27] S. M. Anwar, M. Majid, A. Qayyum, M. Awais, M. Alnowami, and M. K. Khan, "Medical image analysis using convolutional neural networks: A review," *Journal of Medical Systems*, vol. 42, pp. 1–13, 11 Oct. 2018.

[28] Z. Guo, X. Li, H. Huang, N. Guo, and Q. Li, "Deep learning-based image segmentation on multimodal medical imaging," *IEEE transactions on radiation and plasma medical sciences*, vol. 3, pp. 162–169, 2 Mar. 2019.

[29] M. Kim, J. Yun, Y. Cho, *et al.*, "Deep learning in medical imaging," *Neurospine*, vol. 16, pp. 657–668, 4 Dec. 2019.

[30] B. J. Erickson, P. Korfiatis, Z. Akkus, and T. L. Kline, "Machine learning for medical imaging," *Radiographics*, vol. 37, pp. 505–515, 2 Mar. 2017.

[31] G. Litjens, T. Kooi, B. E. Bejnordi, *et al.*, "A survey on deep learning in medical image analysis," *Medical image analysis*, vol. 42, pp. 60–88, Dec. 2017.

[32] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85–117, Jan. 2015.

[33] D. H. Hubel and T. N. Wiesel, "Receptive fields of single neurones in the cat's striate cortex," *The Journal of Physiology*, vol. 148, pp. 574–591, 3 Oct. 1959.

[34] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3431–3440, Oct. 2015.

[35] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," *Medical Image Computing and Computer-Assisted Intervention*, vol. 9351, pp. 234–241, 2015.

[36] M. Drozdzal, E. Vorontsov, G. Chartrand, S. Kadoury, and C. Pal, "The importance of skip connections in biomedical image segmentation," *Lecture Notes in Computer Science*, vol. 10008, pp. 179–187, Oct. 2016.

[37] Özgün Ciçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger, "3d u-net: Learning dense volumetric segmentation from sparse annotation," *MICCAI*, pp. 424–432, 2016.

[38] F. Milletari, N. Navab, and S. A. Ahmadi, "V-net: Fully convolutional neural networks for volumetric medical image segmentation," *International Conference on 3D Vision*, pp. 565–571, Jun. 2016.

[39] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *2016 IEEE conference on computer vision and pattern recognition (CVPR)*, pp. 770–778, 2016.

[40] F. Wang, M. Jiang, C. Qian, *et al.*, "Residual attention network for image classification," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6450–6458, Apr. 2017.

[41] M. Z. Alom, C. Yakopcic, T. T. M., and V. Asari, "Nuclei segmentation with recurrent residual convolutional neural networks based u-net (r2u-net)," *2018 IEEE National Aerospace and Electronics Conference (NAECON)*, pp. 228–233, 2018.

[42] O. Oktay, J. Schlemper, L. L. Folgoc, *et al.*, *Attention u-net: Learning where to look for the pancreas*, 2018.

[43] J. Schlemper, O. Oktay, M. Schaap, *et al.*, "Attention gated networks: Learning to leverage salient regions in medical images," *Medical Image Analysis*, vol. 53, pp. 197–207, Apr. 2019.

[44] X. Li, H. Chen, X. Qi, Q. Dou, C. W. Fu, and P. A. Heng, "H-denseunet: Hybrid densely connected unet for liver and tumor segmentation from ct volumes," *IEEE Transactions on Medical Imaging*, vol. 37, pp. 2663–2674, 12 Dec. 2018.

[45] F. Isensee, P. F. Jaeger, S. A. Kohl, J. Petersen, and K. H. Maier-Hein, "Nnu-net: A self-configuring method for deep learning-based biomedical image segmentation," *Nature methods*, vol. 18, pp. 203–211, 2 Feb. 2021.

[46] F. Isensee, J. Petersen, A. Klein, *et al.*, "Nnu-net: Self-adapting framework for u-net-based medical image segmentation," *Bildverarbeitung für die Medizin*, pp. 22–22, 2019.

[47] Z. Zhou, M. M. R. Siddiquee, N. Tajbakhsh, and J. Liang, "Unet++: A nested u-net architecture for medical image segmentation," *2018 Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support (DLMIA)*, vol. 11045, pp. 3–11, 2018.

[48] S. Kadam and V. Vaidya, "Review and analysis of zero, one and few shot learning approaches," *Advances in Intelligent Systems and Computing*, vol. 940, pp. 100–112, Dec. 2018.

[49] Y. Wang, Q. Yao, J. T. Kwok, and L. M. Ni, "Generalizing from a few examples," *ACM Computing Surveys (CSUR)*, vol. 53, 3 Jun. 2020.

[50] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, pp. 1345–1359, 10 2010.

[51] D. Karimi, S. K. Warfield, A. Gholipour, and S. Member, "Critical assessment of transfer learning for medical image segmentation with fully convolutional neural networks," *Computing Research Repository (CoRR)*, vol. abs/2006.00356, May 2020.

[52] D. Karimi, S. K. Warfield, and A. Gholipour, "Transfer learning in medical image segmentation: New insights from analysis of the dynamics of model parameters and learned representations," *Artificial Intelligence in Medicine*, vol. 116, p. 102 078, Jun. 2021.

[53] A. Shaban, S. Bansal, Z. Liu, I. Essa, and B. Boots, "One-shot learning for semantic segmentation," *2017 British Machine Vision Conference (BMVC)*, vol. 167, pp. 1–13, 2017.

[54] M. Bucher, M. Cord, T. H. Vu, and P. Pérez, "Zero-shot semantic segmentation," *Advances in Neural Information Processing Systems*, vol. 32, Jun. 2019.

[55] L. Fei-Fei, R. Fergus, and P. Perona, "A bayesian approach to unsupervised one-shot learning of object categories," *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2, pp. 1134–1141, 2003.

[56] L. Fei-Fei, R. Fergus, and P. Perona, "One-shot learning of object categories," *IEEE transactions on pattern analysis and machine intelligence*, vol. 28, pp. 594–611, 4 Apr. 2006.

[57] A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra, T. Lillicrap, and G. Deepmind, *Meta-learning with memory-augmented neural networks*, Jun. 2016.

[58] J. Snell, K. Swersky, and T. R. Zemel, "Prototypical networks for few-shot learning," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[59] N. Dong and E. P. Xing, "Few-shot semantic segmentation with prototype learning," *2018 British Machine Vision Conference (BMVC)*, 2018.

[60] X. Zhang, Y. Wei, Y. Yang, and T. S. Huang, "Sg-one: Similarity guidance network for one-shot semantic segmentation," *IEEE Transactions on Cybernetics*, vol. 50, pp. 3855–3865, 9 Oct. 2018.

[61] K. Wang, J. H. Liew, Y. Zou, D. Zhou, and J. Feng, "Panet: Few-shot image semantic segmentation with prototype alignment," *2019 IEEE International Conference on Computer Vision (ICCV)*, pp. 9196–9205, Aug. 2019.

[62] A. K. Mondal, J. Dolz, and C. Desrosiers, "Few-shot 3d multi-modal medical image segmentation using generative adversarial learning," *ArXiv*, vol. abs/1810.12241, Oct. 2018.

[63] S. Wang, S. Cao, D. Wei, *et al.*, "Lt-net: Label transfer by learning reversible voxel-wise correspondence for one-shot medical image segmentation," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 9159–9168, Mar. 2020.

[64] Y. Lu, K. Zheng, W. Li, *et al.*, "Contour transformer network for one-shot segmentation of anatomical structures," *IEEE transactions on medical imaging*, vol. 40, pp. 2672–2684, 10 Oct. 2021.

[65] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 248–255, Mar. 2010.

[66] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," *2018 European Conference on Computer Vision (ECCV)*, pp. 801–818, 2018.

[67] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, and S. Wermter, "Continual lifelong learning with neural networks: A review," *Neural Networks*, vol. 113, pp. 54–71, May 2019.

[68] P. Peng and J. Wang, "How to fine-tune deep neural networks in few-shot learning?" *CoRR*, vol. abs/2012.00204, Dec. 2020.

[69] K. Wang, X. Gao, Y. Zhao, X. Li, D. Dou, and C.-Z. Xu, "Pay attention to features, transfer learn faster cnns | openreview," *2020 International Conference of Learning Representation*, 2020.

[70] J. H. Luo, J. Wu, and W. Lin, "Thinet: A filter level pruning method for deep neural network compression," *IEEE International Conference on Computer Vision (ICCV)*, pp. 5068–5076, Jul. 2017.

[71] Z. Liu, M. Sun, T. Zhou, G. Huang, and T. Darrell, "Rethinking the value of network pruning," *7th International Conference on Learning Representations, ICLR 2019*, Oct. 2018.

[72] M. Raghu, C. Zhang, J. Kleinberg, and S. Bengio, "Transfusion: Understanding transfer learning for medical imaging," *Advances in Neural Information Processing Systems*, vol. 32, Feb. 2019.

[73] P. Rajpurkar, J. Irvin, K. Zhu, *et al.*, "Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning," *CoRR*, vol. abs/1711.05225, Nov. 2017.

[74] V. Gulshan, R. P. Rajan, K. Widner, *et al.*, "Performance of a deep-learning algorithm vs manual grading for detecting diabetic retinopathy in india," *JAMA Ophthalmology*, vol. 137, pp. 987–993, 9 Sep. 2019.

[75] A. W. Liu and J. H. Chan, "An evaluation of transfer learning with chexnet on lung opacity detection in covid-19 and pneumonia chest radiographs," *2021 13th International Conference on Information Technology and Electrical Engineering (ICITEE)*, pp. 137–142, Oct. 2021.

[76] A. Esteva, B. Kuprel, R. A. Novoa, *et al.*, "Dermatologist-level classification of skin cancer with deep neural networks," *Nature*, vol. 542, pp. 115–118, 7639 Jan. 2017.

[77] Y. Ding, J. H. Sohn, M. G. Kawczynski, *et al.*, "A deep learning model to predict a diagnosis of alzheimer disease by using 18 f-fdg pet of the brain," *Radiology*, vol. 290, pp. 456–464, 2 Mar. 2019.

[78] A. V. Opbroek, M. A. Ikram, M. W. Vernooij, and M. D. Bruijne, "Transfer learning improves supervised image segmentation across imaging protocols," *IEEE transactions on medical imaging*, vol. 34, pp. 1018–1030, 5 May 2015.

[79] N. A. Koohbanani, M. Jahanifar, N. Z. Tajadin, and N. Rajpoot, "Nuclick: A deep learning framework for interactive segmentation of microscopic images," *Medical Image Analysis*, vol. 65, 101771 2020.

[80] D. Lin, J. Dai, J. Jia, K. He, and J. Sun, "Scribblesup: Scribble-supervised convolutional networks for semantic segmentation," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3159–3167, 2016.

[81] G. Wang, M. A. Zuluaga, R. Pratt, *et al.*, "Slic-seg: A minimally interactive segmentation of the placenta from sparse and motion-corrupted fetal mri in multiple views," *Medical Image Analysis*, vol. 34, pp. 137–147, Dec. 2016.

[82] T. G. Boers, Y. Hu, E. Gibson, *et al.*, "Interactive 3d u-net for the segmentation of the pancreas in computed tomography scans," *Physics in Medicine and Biology*, vol. 65, p. 065 002, 6 2020.

[83] M. Rajchl, M. C. Lee, O. Oktay, *et al.*, "Deepcut: Object segmentation from bounding box annotations using convolutional neural networks," *IEEE Transactions on Medical Imaging*, vol. 36, pp. 674–683, 2 2017.

[84] E. Redekop and A. Chernyavskiy, "Medical image segmentation with imperfect 3d bounding boxes," *Lecture Notes in Computer Science*, vol. 13003, pp. 193–200, Aug. 2021.

[85] W. A. Barrett and E. N. Mortensen, "Interactive live-wire boundary extraction," *Medical Image Analysis*, vol. 1, pp. 331–341, 4 Sep. 1997.

[86] K. S. Camilus and V. K. Govindan, "A review on graph based segmentation," *Graphics and Signal Processing*, vol. 4, p. 1, 5 Jun. 2012.

[87] F. Zhao and X. Xie, "An overview of interactive medical image segmentation," *Annals of the BMVA*, vol. 2013, pp. 1–22, 7 2013.

[88] C. Xu and J. L. Prince, "Snakes, shapes, and gradient vector flow," *IEEE Transactions on Image Processing*, vol. 7, pp. 359–369, 3 1998.

[89] Y. Y. Boykov and M.-P. Jolly, "Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images," *2001 International Conference on Computer Vision (ICCV)*, vol. 1, pp. 105–112, 2001.

[90] N. Komodakis, G. Tziritas, and N. Paragios, "Fast, approximately optimal solutions for single and dynamic mrfs," *2007 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–8, 2007.

[91] S. Zhou, J. Wang, S. Zhang, Y. Liang, and Y. Gong, "Active contour model based on local and global intensity information for medical image segmentation," *Neurocomputing*, vol. 186, pp. 107–118, Apr. 2016.

[92] D. Freedman and T. Zhang, "Interactive graph cut based segmentation with shape priors," *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, pp. 755–762, 2005.

[93] B. L. Price, B. Morse, and S. Cohen, "Geodesic graph cut for interactive image segmentation," *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3161–3168, 2010.

[94] C. Rother, V. Kolmogorov, and A. Blake, ""grabcut"- interactive foreground extraction using iterated graph cuts," *ACM Transactions on Graphics*, vol. 23, pp. 309–314, 3 2004.

[95] S. Vicente, V. Kolmogorov, and C. Rother, "Graph cut based image segmentation with connectivity priors," *2008 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–8, 2008.

[96] V. Lempitsky, P. Kohli, C. Rother, and T. Sharp, "Image segmentation with a bounding box prior," *2009 IEEE International Conference on Computer Vision*, pp. 277–284, 2009.

[97] M. M. Cheng, V. A. Prisacariu, S. Zheng, P. H. Torr, and C. Rother, "Densecut: Densely connected crfs for realtime grabcut," *Computer Graphics Forum*, vol. 34, pp. 193–201, 7 Oct. 2015.

[98] A. Criminisi, T. Sharp, and A. Blake, "Geos: Geodesic image segmentation," *2008 European Conference on Computer Vision (ECCV)*, pp. 99–112, 2008.

[99] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, pp. 269–271, 1 Dec. 1959.

[100] T. M. Rackham, S. Rueda, C. L. Knight, and J. A. Noble, "Ultrasound image segmentation using feature asymmetry and shape guided live wire," *Medical Imaging 2013: Image Processing*, vol. 8669, 86690P, Mar. 2013.

[101] W. Qiu, J. Yuan, E. Ukwatta, D. Tessier, and A. Fenster, "Three-dimensional prostate segmentation using level set with shape constraint based on rotational slices for 3d end-firing trus guided biopsy," *Medical physics*, vol. 40, 7 2013.

[102] L. Grady, T. Schiwietz, S. Aharon, and R. Westermann, "Random walks for interactive organ segmentation in two and three dimensions: Implementation and validation," *Lecture Notes in Computer Science*, vol. 3750, pp. 773–780, Oct. 2005.

[103] P. A. Yushkevich, Y. Gao, and G. Gerig, "Itk-snap: An interactive tool for semi-automatic segmentation of multi-modality biomedical images," *2016 IEEE Engineering in Medicine and Biology Society*, pp. 3342–3345, Oct. 2016.

[104] I. Arganda-Carreras, V. Kaynig, C. Rueden, *et al.*, "Trainable weka segmentation: A machine learning tool for microscopy pixel classification," *Bioinformatics*, vol. 33, pp. 2424–2426, 15 Aug. 2017.

[105] K. K. Maninis, S. Caelles, J. Pont-Tuset, and L. V. Gool, "Deep extreme cut: From extreme points to object segmentation," *20017 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 616–625, Nov. 2017.

[106] S. Zheng, S. Jayasumana, B. Romera-Paredes, *et al.*, "Conditional random fields as recurrent neural networks," *CoRR*, vol. abs/1502.03240, 2015.

[107] M. Amrehn, S. Gaube, M. Unberath, *et al.*, "Ui-net: Interactive artificial neural networks for iterative image segmentation based on a user model," *Eurographics Workshop on Visual Computing for Biology and Medicine*, 2017.

[108] K. A. Philbrick, A. D. Weston, Z. Akkus, *et al.*, "Ril-contour: A medical imaging dataset annotation tool for and with deep learning," *Journal of Digital Imaging*, vol. 32, pp. 571–581, 4 Aug. 2019.

[109] D. China, A. Illanes, P. Poudel, M. Friebe, P. Mitra, and D. Sheet, "Anatomical structure segmentation in ultrasound volumes using cross frame belief propagating iterative random walks," *IEEE Journal of Biomedical and Health Informatics*, vol. 23, pp. 1110–1118, 3 May 2019.

[110] G. P. Penney, J. A. Schnabel, D. Rueckert, M. A. Viergever, and W. J. Niessen, "Registration-based interpolation," *IEEE Transactions on Medical Imaging*, vol. 23, pp. 922–926, 7 Jul. 2004.

[111] G. Wang, W. Li, M. Aertsen, J. Deprest, S. Ourselin, and T. Vercauteren, "Aleatoric uncertainty estimation with test-time augmentation for medical image segmentation with convolutional neural networks," *Neurocomputing*, vol. 338, pp. 34–45, Apr. 2019.

[112] A. Pepe, R. Schussnig, J. Li, *et al.*, "Iris: Interactive real-time feedback image segmentation with deep learning," *SPIE Medical Imaging*, 2020.

[113] S. W. Oh, J.-Y. Lee, N. Xu, and S. J. Kim, "Video object segmentation using space-time memory networks," *IEEE/CVF International Conference on Computer Vision*, pp. 9225–9234, Apr. 2019.

[114] C. C. Park, B. Kim, and G. Kim, "Attend to you: Personalized image captioning with context sequence memory networks," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6432–6440, 2017.

[115] T. Yang and A. B. Chan, "Learning dynamic memory networks for object tracking," *2018 European Conference on Computer Vision (ECCV)*, vol. 11213, pp. 153–169, 2018.

[116] S. Lee, J. Sung, Y. Yu, and G. Kim, "A memory network approach for story-based temporal summarization of 360° videos," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1410–1419, 2018.

[117] A. Top, G. Hamarneh, and R. Abugharbieh, "Active learning for interactive 3d image segmentation," *2011 Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, vol. 6893, pp. 603–610, 2011.

[118] B. Settles, "Active learning literature survey. computer sciences technical report 1648," University of Wisconsin-Madison, 2009.

[119] A. Smailagic, P. Costa, H. Y. Noh, *et al.*, "Medal: Accurate and robust deep active learning for medical image analysis," *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 481–488, 2018.

[120] S. Budd, E. C. Robinson, and B. Kainz, "A survey on active learning and human-in-the-loop deep learning for medical image analysis," *Medical Image Analysis*, vol. 71, p. 102 062, Jul. 2021.

[121] Y. Gal, R. Islam, and Z. Ghahramani, "Deep bayesian active learning with image data," *34th International Conference on Machine Learning (ICML)*, vol. 70, pp. 1183–1192, 2017.

[122] C. Murray, J. U. Allingham, J. Antorán, and J. M. Hernández-Lobato, "Addressing bias in active learning with depth uncertainty networks... or not," *Proceedings of Machine Learning Research, NeurIPS 2021 Workshops*, vol. 163, pp. 59–63, 2022.

[123] G. Wang, M. Aertsen, J. Deprest, S. Ourselin, T. Vercauteren, and S. Zhang, "Uncertainty-guided efficient interactive refinement of fetal brain segmentation from stacks of mri slices," *2020 Medical Image Computing and Computer Assisted Intervention (MICCAI)*, vol. 12264, pp. 279–288, 2020.

[124] E. Zheng, Q. Yu, R. Li, P. Shi, and A. Haake, "A continual learning framework for uncertainty-aware interactive image segmentation," *AAAI Conference on Artificial Intelligence*, vol. 35, 7 2021.

[125] M. Abdar, F. Pourpanah, S. Hussain, *et al.*, "A review of uncertainty quantification in deep learning: Techniques, applications and challenges," *Information Fusion*, vol. 76, pp. 243–297, Dec. 2021.

[126] C. E. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*, 1948.

[127] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," *33rd International Conference on Machine Learning (ICML)*, vol. 48, pp. 1050–1059, Jun. 2016.

[128] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 56 2014.

[129] G. Sharp, K. D. Fritscher, V. Pekar, *et al.*, "Vision 20/20: Perspectives on automated image segmentation for radiotherapy," *Medical Physics*, vol. 41, 5 2014.

[130] U. Mahmood, "Deep learning for medical imaging in a biased environment," Chester F. Carlson Center for Imaging Science (COS), Aug. 2022.

[131] K. V. Sarma, S. Harmon, T. Sanford, *et al.*, "Federated learning improves site performance in multicenter deep learning without data sharing," *Journal of the American Medical Informatics Association*, vol. 28, pp. 1259–1264, 6 Jun. 2021.

[132] J. Yang, G. Sharp, H. Veeraraghavan, *et al.*, "Data from lung ct segmentation challenge.," *The Cancer Imaging Archive*, 2017.

[133] H. J. W. L. Aerts, L. Wee, E. R. Velazquez, *et al.*, "Data from nsclc-radiomics lung1," *The Cancer Imaging Archive*, 2019.

[134] A. L. Simpson, M. Antonelli, S. Bakas, *et al.*, "A large annotated medical image dataset for the development and evaluation of segmentation algorithms," *The Cancer Imaging Archive*, Jul. 2019.

[135] L. Wee and A. Dekker, "Data from head-neck-radiomics-hn1," *The Cancer Imaging Archive*, 2019.

[136] M. Reyes, R. Meier, S. Pereira, *et al.*, "On the interpretability of artificial intelligence in radiology: Challenges and opportunities," *Radiology: Artificial Intelligence*, vol. 2, e190043, 3 May 2020.

[137] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 618–626, 2017.

[138] M. Z. Alom, M. Hasan, C. Yakopcic, T. M. Taha, and V. K. Asari, "Recurrent residual convolutional neural network based on u-net (r2u-net) for medical image segmentation," *ArXiv*, vol. abs/1802.0, 2018.

[139] C. J. Spoerer, P. McClure, and N. Kriegeskorte, "Recurrent convolutional neural networks: A better model of biological object recognition," *Frontiers in Psychology*, vol. 8, Sep. 2017.

[140] K. Janocha and W. M. Czarnecki, "On loss functions for deep neural networks in classification," *Schedae Informaticae*, vol. 25, pp. 49–59, 2016.

[141] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," 2015.

[142] L. R. Dice, "Dice lr: Measures of the amount of ecologic association between species.," *Ecology*, vol. 26, pp. 297–302, 3 1945.

[143] X. Feng, K. Qing, N. J. Tustison, C. H. Meyer, and Q. Chen, "Deep convolutional neural network for segmentation of thoracic organs at risk using cropped 3d images," *Medical Physics*, vol. 46, pp. 2169–2180, 5 May 2019.

[144] S. Hossain, S. Najeeb, A. Shahriyar, Z. R. Abdullah, and M. A. Haque, "A pipeline for lung tumor detection and segmentation from ct scans using dilated convolutional neural networks," *2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1348–1352, May 2019.

[145] T. Vercauteren, X. Pennec, A. Perchant, and N. Ayache, "Non-parametric diffeomorphic image registration with the demons algorithm," *Med Image Comput Comput Assist Interv.*, vol. 10, pp. 319–326, 2007.

[146] D. P. Huttenlocher, G. A. Klanderman, and W. J. Rucklidge, "Comparing images using the hausdorff distance," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, pp. 850–863, 9 1993.

[147] F. Vaassen, C. Hazelaar, A. Vaniqui, *et al.*, "Evaluation of measures for assessing time-saving of automatic organ-at-risk segmentation in radiotherapy," *Physics and Imaging in Radiation Oncology*, vol. 13, pp. 1–6, Jan. 2020.

[148] J. W. Piper, "Systems and methods for contouring a set of medical images," pat., 2017.

[149] T. Heimann, B. V. Ginneken, M. A. Styner, *et al.*, "Comparison and evaluation of methods for liver segmentation from ct datasets," *IEEE transactions on medical imaging*, vol. 28, pp. 1251–1265, 8 Aug. 2009.

[150] L. J. Stapleford, J. D. Lawson, C. Perkins, *et al.*, "Evaluation of automatic atlas-based lymph node segmentation for head-and-neck cancer," *International journal of radiation oncology, biology, physics*, vol. 77, pp. 959–966, 3 2010.

[151] A. K. H. Duc, G. Eminowicz, R. Mendes, *et al.*, "Validation of clinical acceptability of an atlas-based segmentation algorithm for the delineation of organs at risk in head and neck cancer," *Medical physics*, vol. 42, 9 Sep. 2015.

[152] V. K. Reed, W. A. Woodward, L. Zhang, *et al.*, "Automatic segmentation of whole breast using atlas approach and deformable image registration," *International journal of radiation oncology, biology, physics*, vol. 73, pp. 1493–1500, 5 Apr. 2009.

[153] W. J. Zabel, J. L. Conway, A. Gladwish, *et al.*, "Clinical evaluation of deep learning and atlas-based auto-contouring of bladder and rectum for prostate radiation therapy," *Practical Radiation Oncology*, vol. 11, e80–e89, 1 Jan. 2021.

[154] P Li, S Wang, T Li, J Lu, Y HuangFu, and D Wang, "A large-scale ct and pet/ct dataset for lung cancer diagnosis," *The Cancer Imaging Archive*, 2020.

[155] S. K. Warfield, K. H. Zou, and W. M. Wells, "Simultaneous truth and performance level estimation (staple): An algorithm for the validation of image segmentation," *IEEE transactions on medical imaging*, vol. 23, pp. 903–921, 7 Jul. 2004.

[156] G. Wang, W. Li, S. Ourselin, and T. Vercauteren, "Automatic brain tumor segmentation using cascaded anisotropic convolutional neural networks," *BrainLes 2017: Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries*, vol. 10670, pp. 178–190, 2018.

[157] F. Vaassen, D. Boukerroui, P. Looney, *et al.*, "Real-world analysis of manual editing of deep learning contouring in the thorax region," *Physics and Imaging in Radiation Oncology*, vol. 22, pp. 104–110, Apr. 2022.

[158] R. Baskar, K. A. Lee, R. Yeo, and K. W. Yeoh, "Cancer and radiation therapy: Current advances and future directions," *International journal of medical sciences*, vol. 9, pp. 193–199, 3 Feb. 2012.

[159] M. S. Wilson and T. A. Wynn, "Pulmonary fibrosis: Pathogenesis, etiology and regulation," *Mucosal immunology*, vol. 2, p. 103, 2 2009.

[160] L. Giuranno, J. Ient, D. D. Ruysscher, and M. A. Vooijs, "Radiation-induced lung injury (rili)," *Frontiers in Oncology*, vol. 9, p. 877, Sep. 2019.

[161] A. Christe, A. A. Peters, D. Drakopoulos, *et al.*, "Computer-aided diagnosis of pulmonary fibrosis using deep learning and ct images.," *Investigative Radiology*, vol. 54, pp. 627–632, 10 Oct. 2019.

[162] J. Ley-Zaporozhan, A. Giannakis, T. Norajitra, *et al.*, "Fully automated segmentation of pulmonary fibrosis using different software tools," *Respiration; international review of thoracic diseases*, vol. 100, pp. 580–587, 7 Jul. 2021.

[163] G. Wang, S. Zhang, S. Zhai, *et al.*, "Semi-supervised segmentation of radiation-induced pulmonary fibrosis from lung ct scans with multi-scale guided dense attention," *IEEE transactions on medical imaging*, vol. 41, pp. 531–542, 3 Mar. 2022.

[164] D. Yang, G. Lasio, B. Zhang, *et al.*, "Automated pulmonary fibrosis segmentation using a 3d multi-scale convolutional encoder-decoder approach in thoracic ct for the rhesus macaque with radiation-induced lung damage," *Journal of Signal Processing Systems*, vol. 94, pp. 473–483, 5 Oct. 2020.

[165] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" *27th International Conference on Neural Information Processing Systems (NIPS)*, vol. 2, pp. 3320–3328, 2014.

[166] Z. Zhang and V. Saligrama, "Zero-shot learning via semantic similarity embedding," *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 4166–4174, Sep. 2015.

[167] N. Tajbakhsh, J. Y. Shin, S. R. Gurudu, *et al.*, "Convolutional neural networks for medical image analysis: Full training or fine tuning?" *IEEE Transactions on Medical Imaging*, vol. 35, pp. 1299–1312, 5 Jun. 2017.

[168] M. Ghafoorian, A. Mehrtash, T. Kapur, *et al.*, "Transfer learning for domain adaptation in mri: Application in brain lesion segmentation," *Lecture Notes in Computer Science*, vol. 10435, pp. 516–524, 2017.

[169] A. Fedorov, R. Beichel, J. Kalpathy-Cramer, *et al.*, "3d slicer as an image computing platform for the quantitative imaging network," *Magnetic resonance imaging*, vol. 30, pp. 1323–1341, 9 Nov. 2012.

[170] T. Yamamoto, N. Kadoya, Y. Morishita, *et al.*, "Assessment and agreement of the ct appearance pattern and its severity grading of radiation-induced lung injury after stereotactic body radiotherapy for lung cancer," *PloS one*, vol. 13, 10 Jul. 2018.

[171] A. Linda, M. Trovo, and J. D. Bradley, "Radiation injury of the lung after stereotactic body radiation therapy (sbrt) for lung cancer: A timeline and pattern of ct changes," *European journal of radiology*, vol. 79, pp. 147–154, 1 Jul. 2011.

[172] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, 7553 May 2015.

[173] G. Mårtensson, D. Ferreira, T. Granberg, *et al.*, "The reliability of a deep learning model in clinical out-of-distribution mri data: A multicohort study," *Medical Image Analysis*, vol. 66, p. 101 714, Dec. 2020.

[174] S. Takahashi, M. Takahashi, M. Kinoshita, *et al.*, "Fine-tuning approach for segmentation of gliomas in brain magnetic resonance images with a machine learning method to normalize image differences among facilities," *Cancers*, vol. 13, pp. 1–15, 6 Mar. 2021.

[175] B. Settles and M. Craven, "An analysis of active learning strategies for sequence labeling tasks," *2008 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference (EMNLP)*, pp. 1070–1079, 2008.

[176] W. H. Beluch, T. Genewein, A. Nürnberger, and J. M. Köhler, "The power of ensembles for active learning in image classification," *2018 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9368–9377, 2018.

[177] Y. Gal and Z. A. Uk, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," *33rd International Conference on Machine Learning (ICML)*, vol. 48, pp. 1050–1059, 2016.

[178] T. DeVries and G. W. Taylor, "Leveraging uncertainty estimates for predicting segmentation quality," *ArXiv*, vol. abs/1807.00502, Jul. 2018.

[179] M. Kampffmeyer, A.-B. Salberg, and R. Jenssen, "Semantic segmentation of small objects and modeling of uncertainty in urban remote sensing images using deep convolutional neural networks," *2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 680–688, 2016.

[180] A. Kendall, V. Badrinarayanan, and R. Cipolla, "Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding," *2017 British Machine Vision Conference (BMVC)*, 2017.

[181] A. Kendall and Y. Gal, "What uncertainties do we need in bayesian deep learning for computer vision?" *2017 31st International Conference on Neural Information Processing Systems (NIPS)*, pp. 5580–5590, 2017.

[182] G. Sohaliya and K. Sharma, "Semantic segmentation using generative adversarial networks with a feature reconstruction loss," *2021 Asian Conference on Innovation in Technology (ASIANCON)*, pp. 1–7, Aug. 2021.

[183] S. Kohl, D. Bonekamp, H.-P. Schlemmer, *et al.*, "Adversarial networks for the detection of aggressive prostate cancer," *CoRR*, vol. abs/1702.08014, Feb. 2017.

[184] P. Moeskops, M. Veta, M. W. Lafarge, K. A. Eppenhof, and J. P. Pluim, "Adversarial training and dilated convolutions for brain mri segmentation," *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, vol. 10553, pp. 56–64, 2017.

[185] C. Huang, Q. Wu, and F. Meng, "Qualitynet: Segmentation quality evaluation with deep convolutional networks," *2016 Visual Communications and Image Processing (VCIP)*, pp. 1–4, Jan. 2016.

[186] S. Budd, "Symbiotic deep learning for medical image analysis with applications in real-time diagnosis for fetal ultrasound screening," Imperial College of Science, Technology and Medicine, Sep. 2021, pp. 43–44.

[187] V. L. Nguyen, M. H. Shaker, and E. Hüllermeier, "How to measure uncertainty in uncertainty sampling for active learning," *Machine Learning*, vol. 111, pp. 89–122, 1 Jan. 2022.

[188] C. Corbière, N. Thome, A. Bar-Hen, M. Cord, and P. Pérez, "Addressing failure prediction by learning model confidence," *Proceedings of the 33rd International Conference on Neural Information Processing Systems (NIPS)*, vol. 261, 2019.

[189] D. Hendrycks and K. Gimpel, "A baseline for detecting misclassified and out-of-distribution examples in neural networks," *2017 5th International Conference on Learning Representations (ICLR)*, Oct. 2017.

[190] S. Sukhbaatar, A. Szlam, J. Weston, and R. Fergus, "End-to-end memory networks," *2015 Proceedings of the 28th International Conference on Neural Information Processing Systems (NIPS)*, vol. 2, pp. 2440–2448, 2015.

[191] A. H. Miller, A. Fisch, J. Dodge, A. H. Karimi, A. Bordes, and J. Weston, "Key-value memory networks for directly reading documents," *2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1400–1409, 2016.

[192] A. Kumar, O. Irsoy, P. Ondruska, *et al.*, "Ask me anything: Dynamic memory networks for natural language processing," *33rd International Conference on Machine Learning (ICML)*, vol. 48, pp. 1378–1387, 2016.

[193] A. Khoreva, F. Perazzi, R. Benenson, B. Schiele, and A. Sorkine-Hornung, "Learning video object segmentation from static images," *017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3491–3500, 2016.

[194] S. W. Oh, J.-Y. Lee, A. Research, N. Xu, and S. J. Kim, "Fast user-guided video object segmentation by interaction-and-propagation networks," *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5242–5251, 2019.

[195] S. W. Oh, J. Y. Lee, K. Sunkavalli, and S. J. Kim, "Fast video object segmentation by reference-guided mask propagation," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7376–7385, Dec. 2018.

[196] S. W. Oh, J. Y. Lee, N. Xu, and S. J. Kim, "Space-time memory networks for video object segmentation with user guidance," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, pp. 442–455, 1 Jan. 2022.

[197] P. Zhang, L. Hu, B. Zhang, and P. Pan, "Spatial consistent memory network for semi-supervised video object segmentation," *The 2020 DAVIS Challenge on Video Object Segmentation (CVPR 2020 Workshops)*, 2020.

Appendices

A
Data Overview

This section gives an overview of the different datasets used in this thesis. The experimental chapters of the thesis describe the number of images used for training, validation and testing of specific models, whereas here the overall characteristics of the individual datasets are summarized in Table A.1.

Table A.1: Overview of the individual datasets used in the thesis showing number of patients and structure included.

Dataset	**# Patients**	**Structures**	**Chapter**
AAPM 2017 [132]	60	Heart, oesophagus, lung (left, right), spinal cord	3, 6, 7
Lung1 [133]	422	Heart, NSCLC	3, 6, 7
HN1 [135]	137	Lymph nodes, submandibular glands, parotid, oral cavity, medulla oblongata, brain, brain tumour submandibular glands, parotid, oral cavity, medulla oblongata, brain, brain tumour	3, 6, 7
MSD Pancreas [134]	420	Pancreas, Pancreas tumour	3, 6, 7
MSD Liver [134]	210	Liver, Liver tumour	3, 6, 7
MSD Spleen [134]	61	Spleen	3, 6, 7
Lung Fibrosis (MTA Heidelberg)	38	Lung fibrosis	5
CT/PET Lung tumour (MTA Liege)	50	Lung tumour	4

B
Learning from Context

B.1 Impact of training set diversity on generalizability from context - Additional structures and evaluation measures

The figures provided in this section show the results relating to the main work for the relative APL, HD and DSC measure and all 19 structures investigated. The supporting figures are summarized in Table B.1.

Table B.1: Reference of main text figure to supplementary figure. The supplementary figures include all 19 structures investigated and are reported for the relative APL, DSC and HD.

Main text figure	APL	DSC	HD
Figure 3.10	Figure B.1	Figure B.1	Figure B.1
Figure 3.11	Figure B.2	Figure B.3,	Figure B.4
Figure 3.12	Figure B.5	Figure B.6	Figure B.7

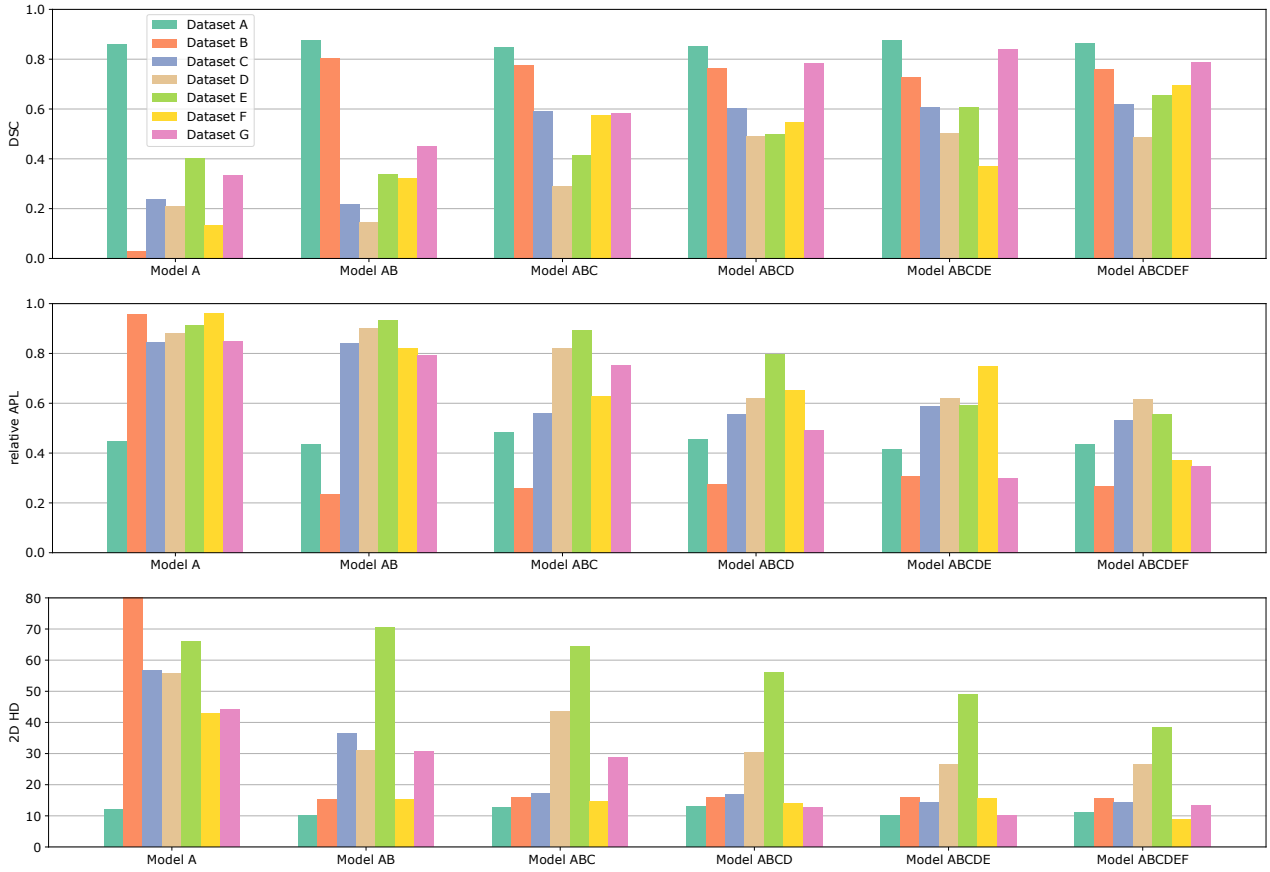


Figure B.1: Performance of models with successively increasing training set on different test datasets. This Figure is the equivalent of Figure 3.10, but considers the HD measure in addition to the DSC and relative APL measure.

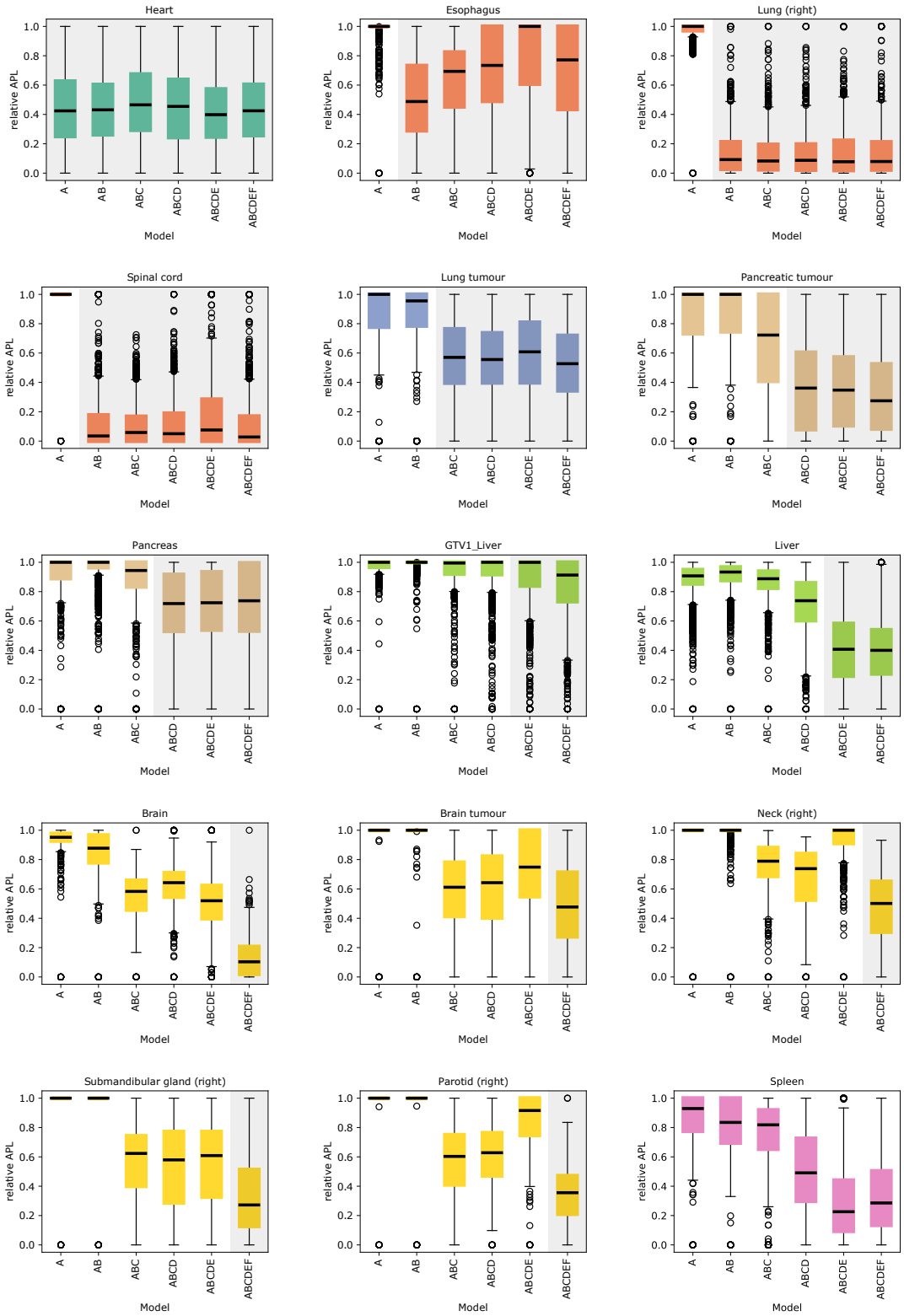


Figure B.2: Boxplots of the performance (relative APL) of different models for different structures. This figure is an extension of Figure 3.11 in the main part of the thesis to include all structures. The grey shaded area indicates the models that include the structure in their respective training set. Structures that have a left/right counter part are only shown for the right side because they are qualitatively similar.

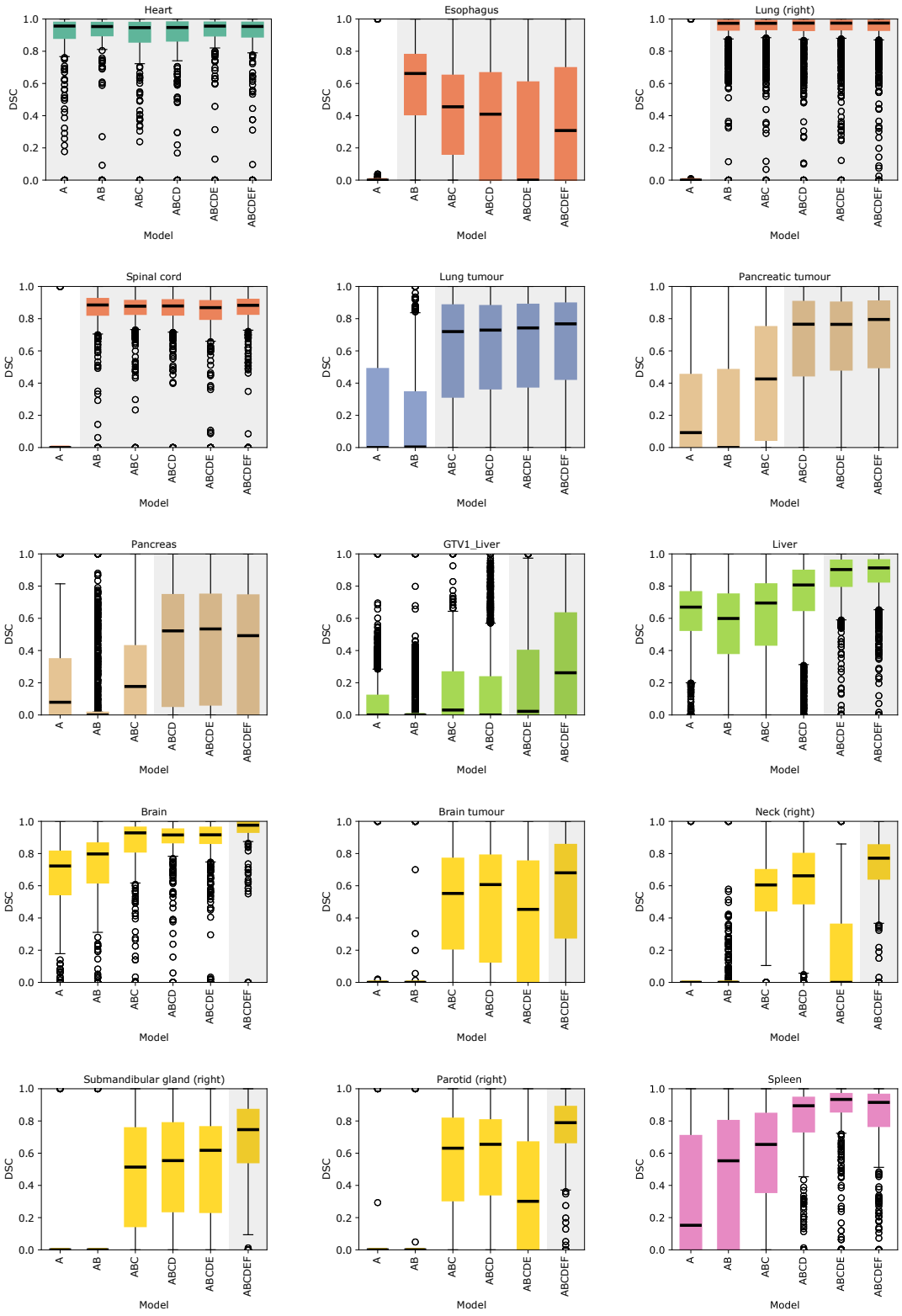


Figure B.3: Boxplots of the performance (DSC) of different models for different structures. This figure is an extension of Figure 3.11 in the main part of the thesis to include all structures. The grey shaded area indicates the models that include the structure in their respective training set. Structures that have a left/right counter part are only shown for the right side because they are qualitatively similar.

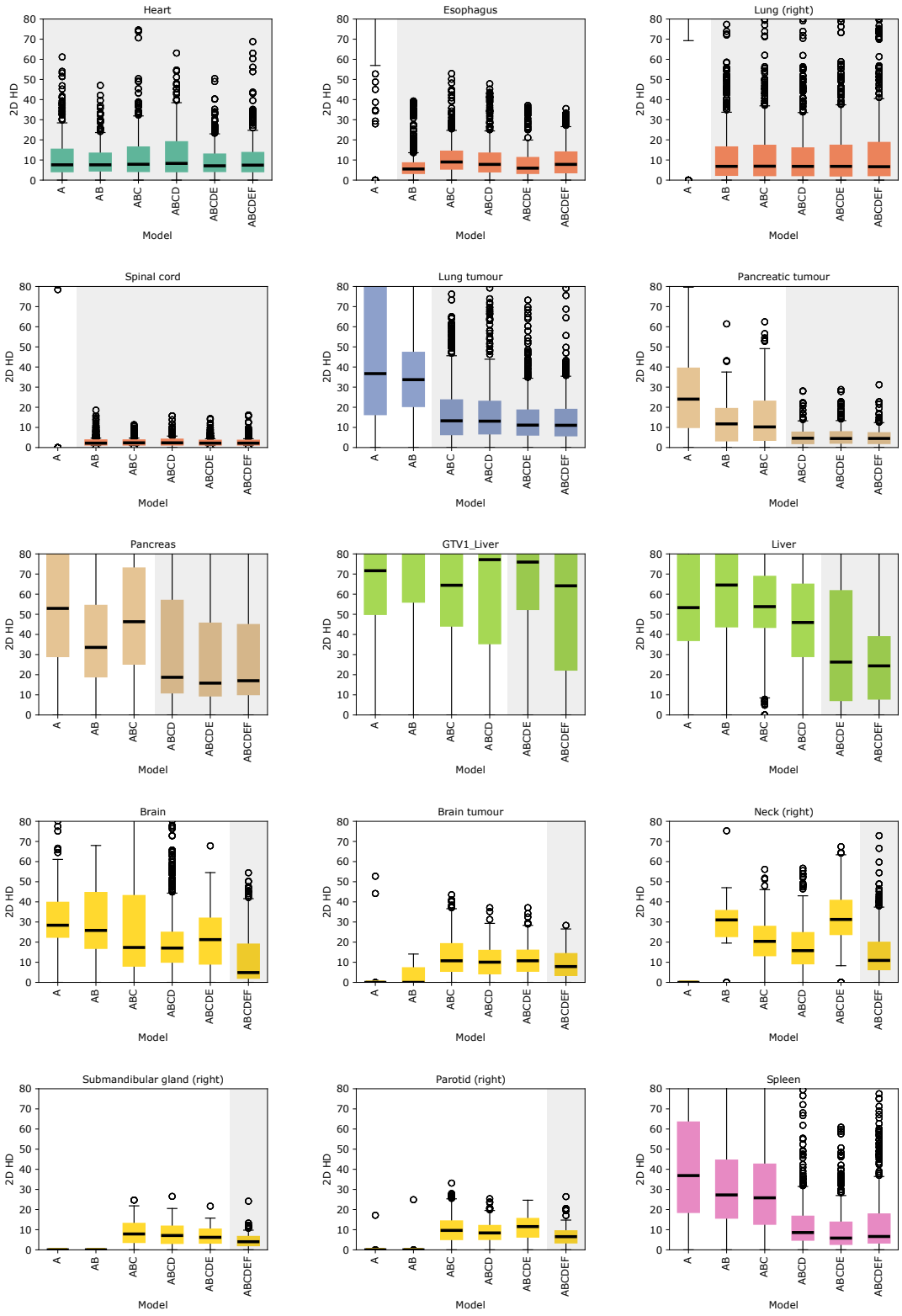


Figure B.4: Boxplots of the performance (95% HD) of different models for different structures. This figure is an extension of Figure 3.11 in the main part of the thesis to include all structures. The grey shaded area indicates the models that include the structure in their respective training set. Structures that have a left/right counter part are only shown for the right side because they are qualitatively similar.

B.2 Comparison of contour prediction approaches - Additional structures and evaluation measures

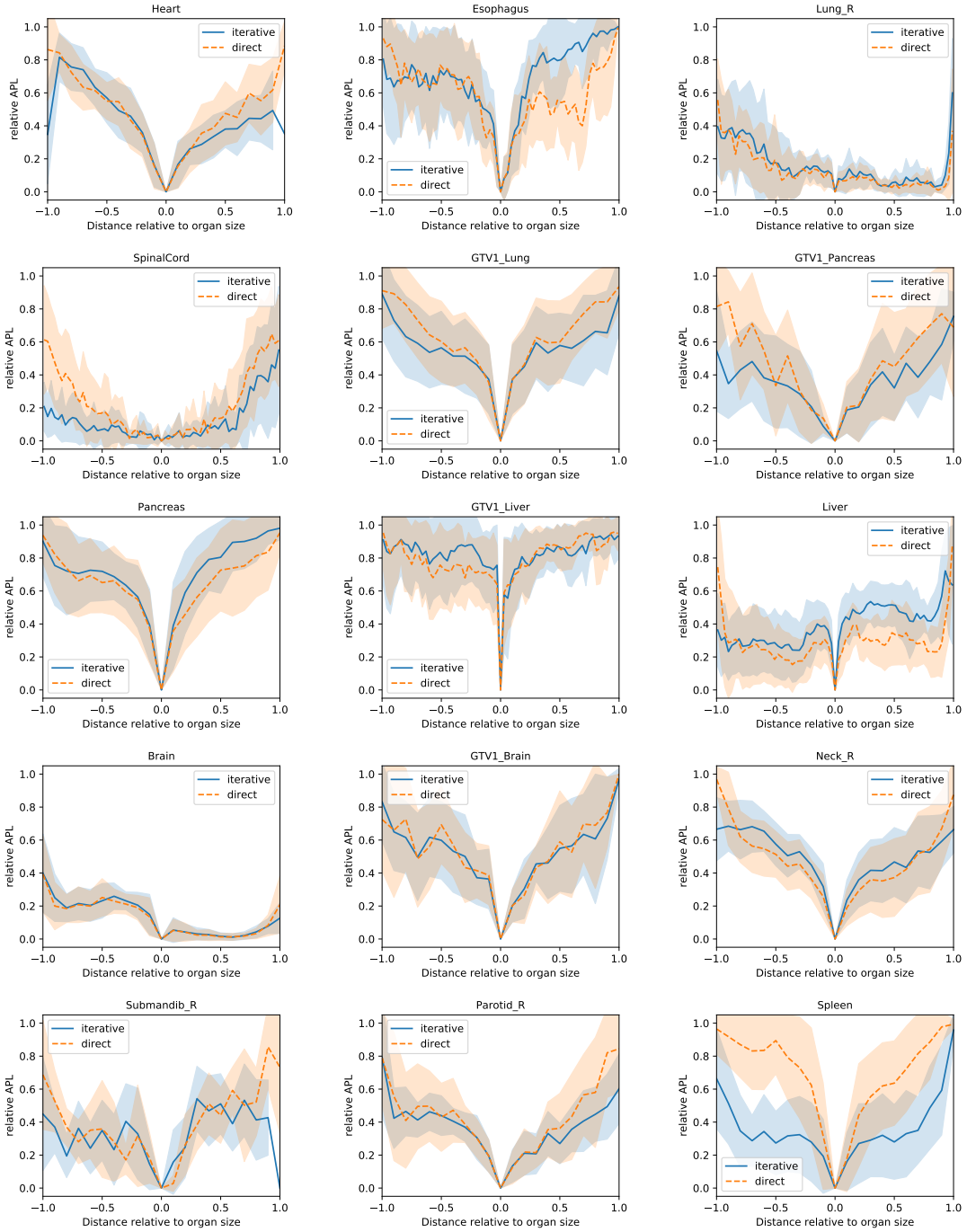


Figure B.5: Comparison of direct (orange-dashed line) and iterative (blue line) prediction using the model ABCDEF on different structures. This Figure is an extension of Figure 3.12 in the main part of the thesis to include all structures, assessed using relative APL.

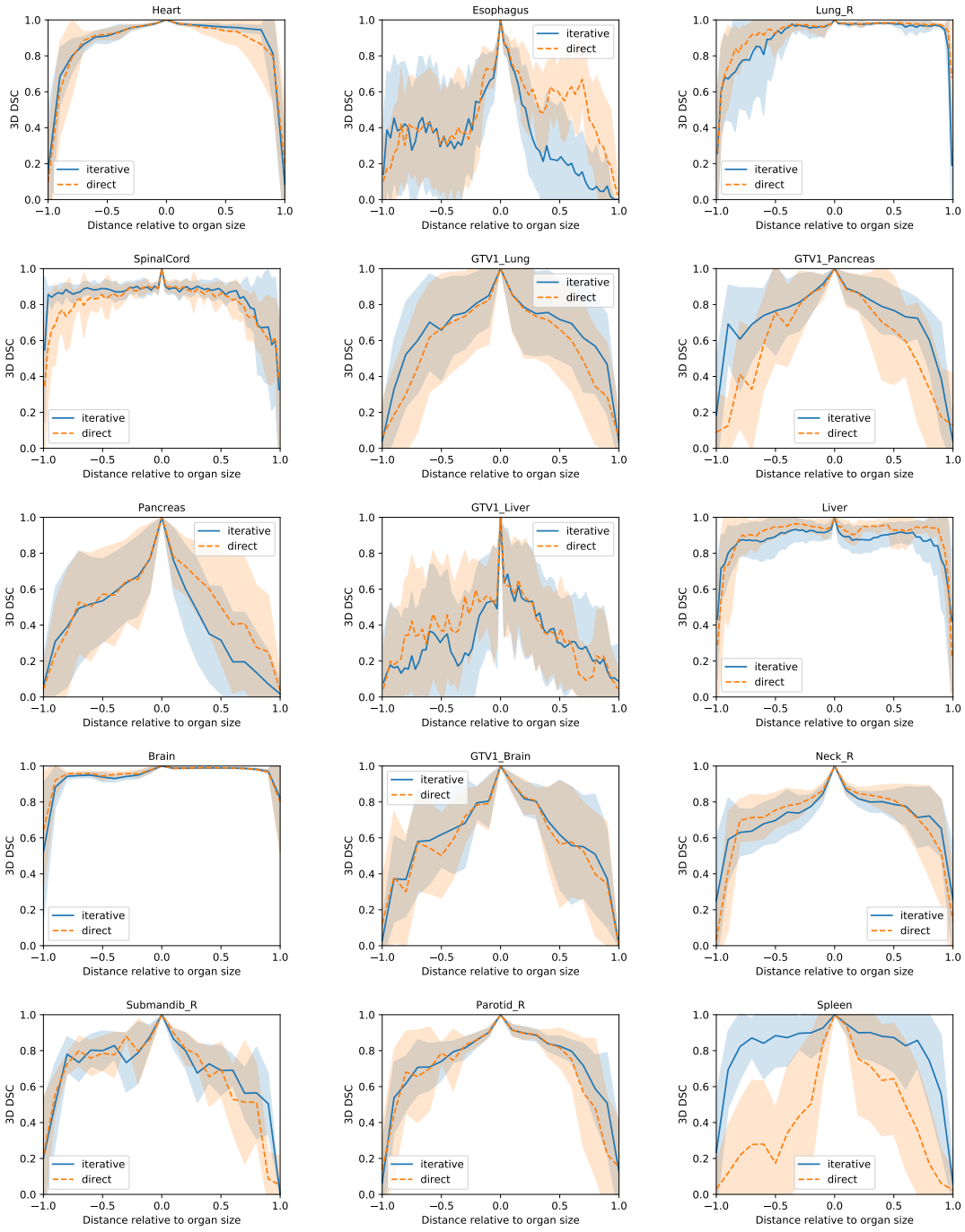


Figure B.6: Comparison of direct (orange-dashed line) and iterative (blue line) prediction using the model ABCDEF on different structures. This Figure is an extension of Figure 3.12 in the main part of the thesis to include all structures, assessed using DSC.

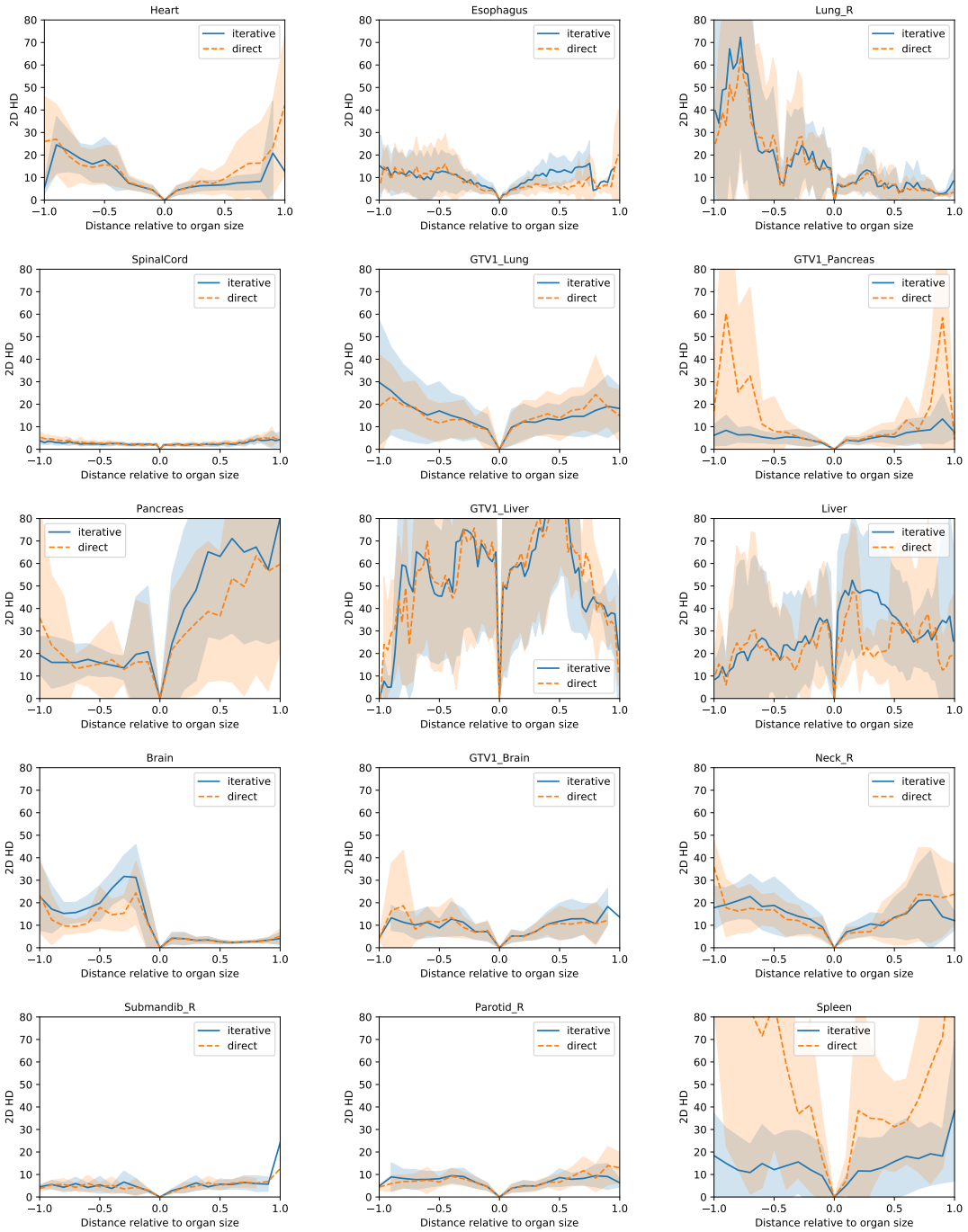


Figure B.7: Comparison of direct (orange-dashed line) and iterative (blue line) prediction using the model ABCDEF on different structures. This Figure is an extension of Figure 3.12 in the main part of the thesis to include all structures, assessed using 95% HD.

B.3 Models trained on all but one structure

In this section a series of models is presented, that was trained on all, but one structure. Note, additionally to the single excluded structure the spleen was never included in any training set, as the size of that dataset did not allow for an adequate split into training, validation and test set as set out in our methods. The results are shown in Figures B.8 to B.10.
Models that were trained excluding one particular structure for training have a higher relative APL as compared to models that include that structure. This is not surprising, as one would expect higher performance if the model was trained for a specific structure. However, this shows that even if the structure was not included in the training set, it can segment the structure given only one user interaction.

In the following figures the different structures are referred to by the following numbers:

1 Esophagus
2 Heart
3 Lung (left)
4 Lung (right)
5 Spinal cord
6 Lung tumour
7 Neck (left)
8 Neck (right)
9 Submandibular gland (left)
10 Submandibular gland (right)
11 Parotid (left)
12 Parotid (right)
13 Brain
14 Brain tumour
15 Liver
16 Liver tumour
17 Pancreas
18 Pancreatic tumour
19 Spleen

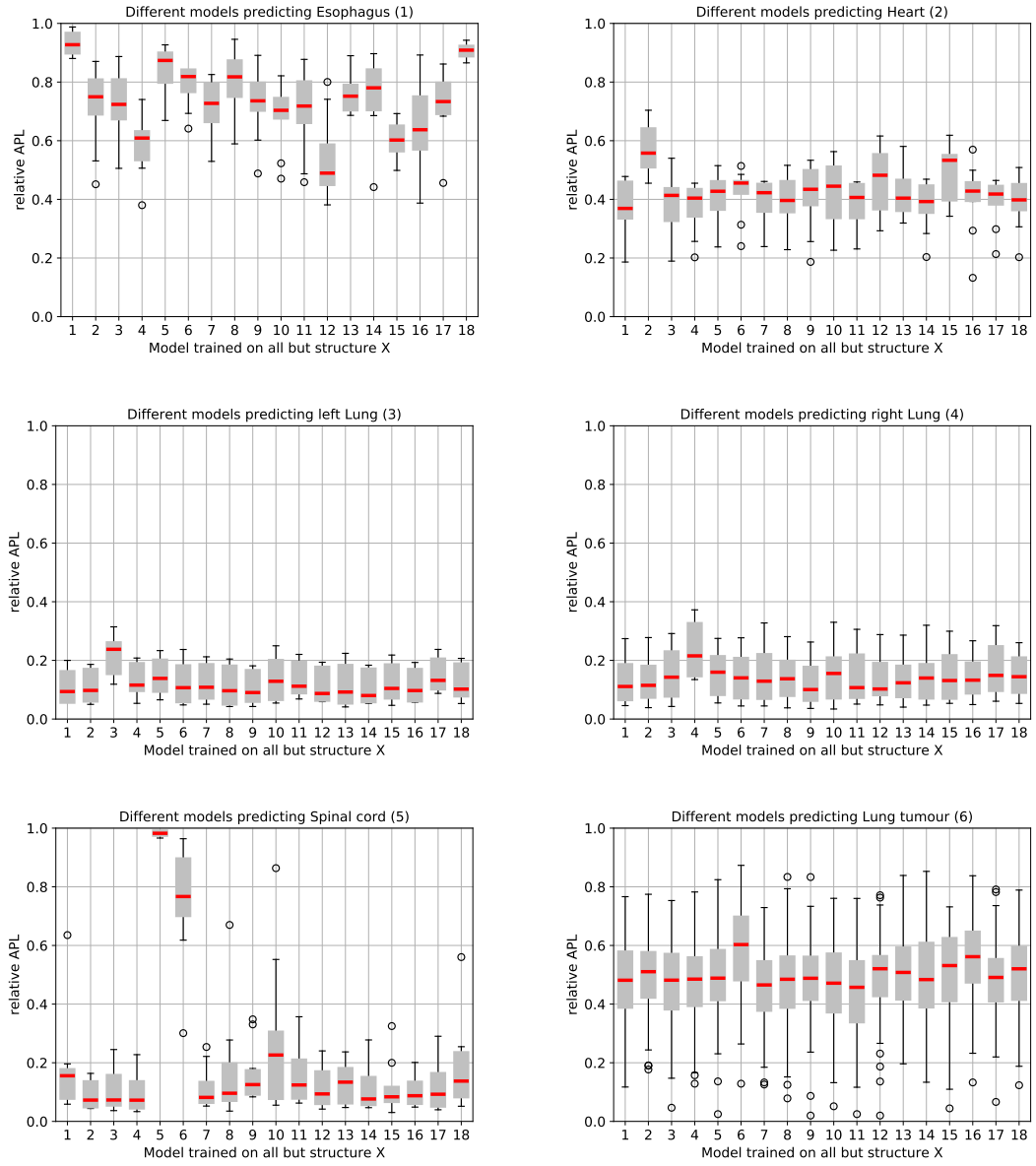


Figure B.8: Multi-structure models trained on all but one structure. For most structures, the one model excluding that structure from training performs worse than other structures. This is not surprising, as it lacks prior training on that structure. Importantly, the model still manages to achieve good segmentation results for structures excluded from the training set.

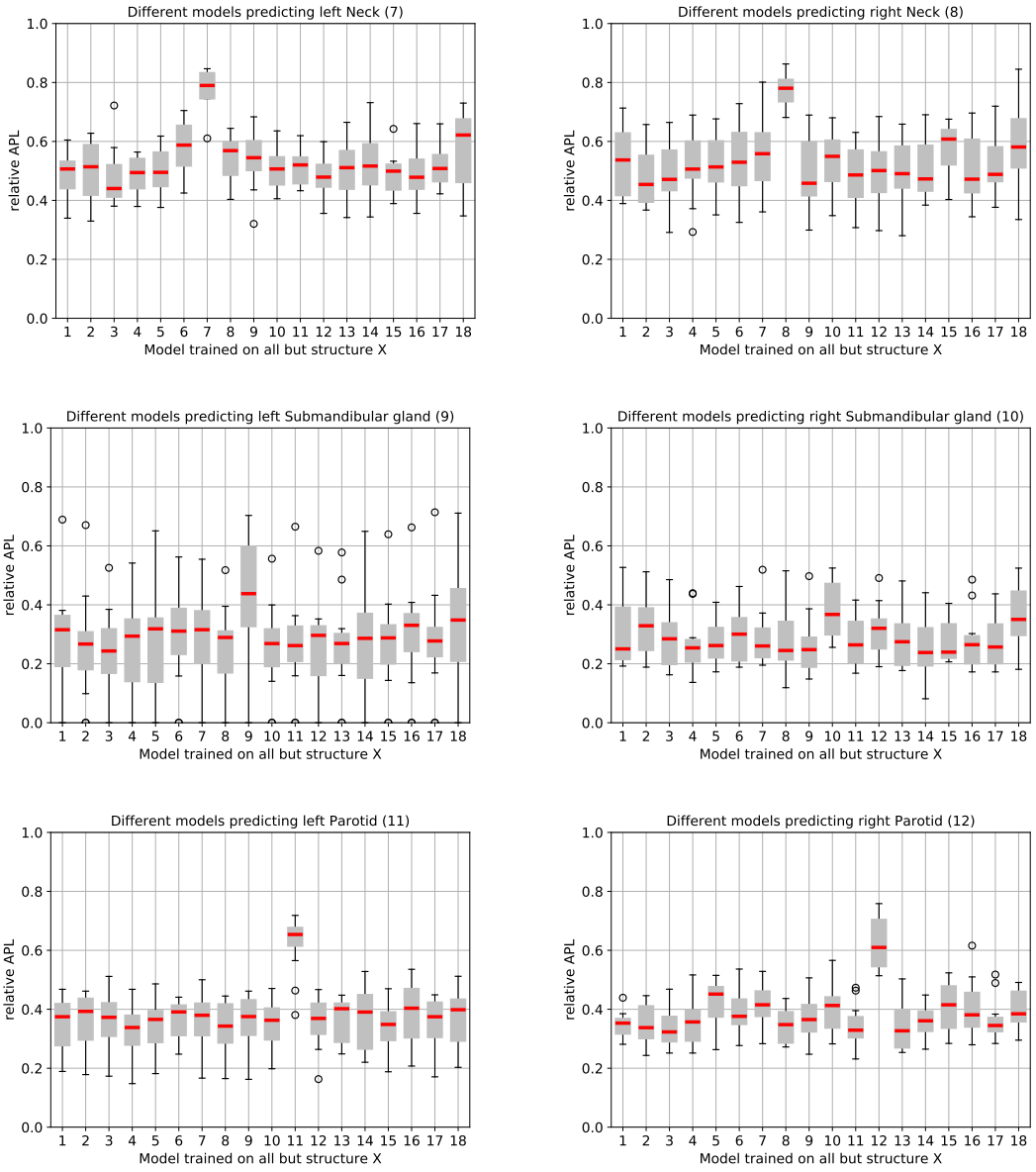


Figure B.9: Equivalent to Figure B.8 for the different structures. Multi-structure models trained on all but one structure. For most structures, the one model excluding that structure from training performs worse than other structures. This is not surprising, as it lacks prior training on that structure. Importantly, the model still manages to achieve good segmentation results for structures excluded from the training set.

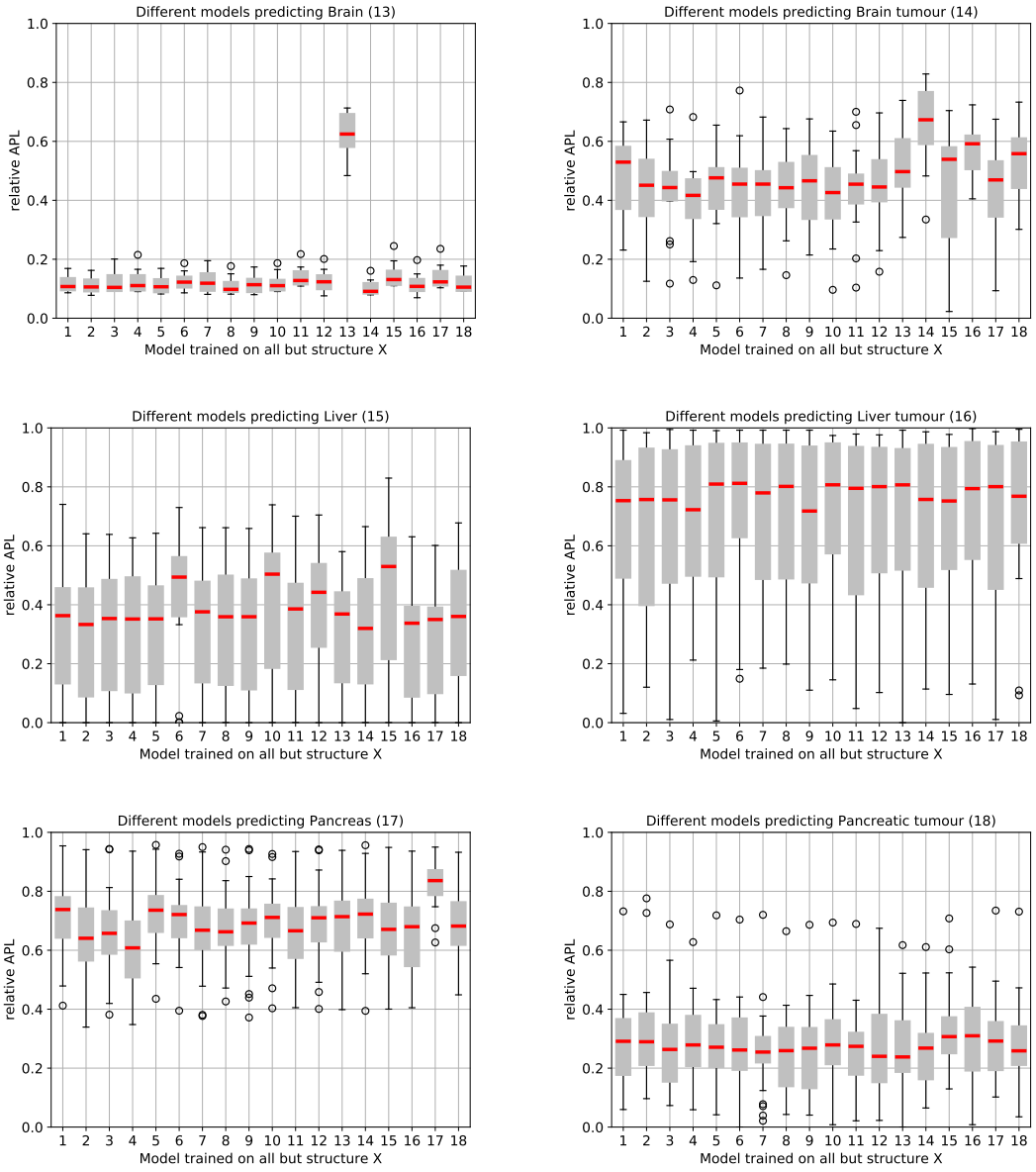


Figure B.10: Equivalent to Figure B.8 for the different structures. Multi-structure models trained on all but 1 structure are evaluated for the excluded structure. For most structures, the one model excluding that structure from training performs worse than other structures. This is not surprising, as it lacks prior training on that structure. Importantly, the model still manages to achieve good segmentation results for structures excluded from the training set.

B.4 Models trained on a single structure

In this section a series of models is presented, that was trained on a single structure and evaluated for predicting a wide range of structures. The results are shown in Figure B.11. Note, that only models trained for structures 2, 3, 4, 13, 15, and 16 converged. Specifically single-structure models for small anatomical structures tended to fail. This is attributed to the bias given through the binomial cross entropy loss function. Given different randomly initialized weights the other structures should be able to converge as well. Due to computational limitations, models that failed to converge were not trained using different initial weights multiple times. Each of single structure models excels at predicting the structure it has been trained on, but fails for most other structures. For example, The model trained on the brain, still shows a DSC of 0.5 for the heart, while the lung models have a DSC of 0 for the heart.

In the following figures the different structures are referred to by the following numbers:

1 Esophagus
2 Heart
3 Lung (left)
4 Lung (right)
5 Spinal cord
6 Lung tumour
7 Neck (left)
8 Neck (right)
9 Submandibular gland (left)
10 Submandibular gland (right)
11 Parotid (left)
12 Parotid (right)
13 Brain
14 Brain tumour
15 Liver
16 Liver tumour
17 Pancreas
18 Pancreatic tumour
19 Spleen

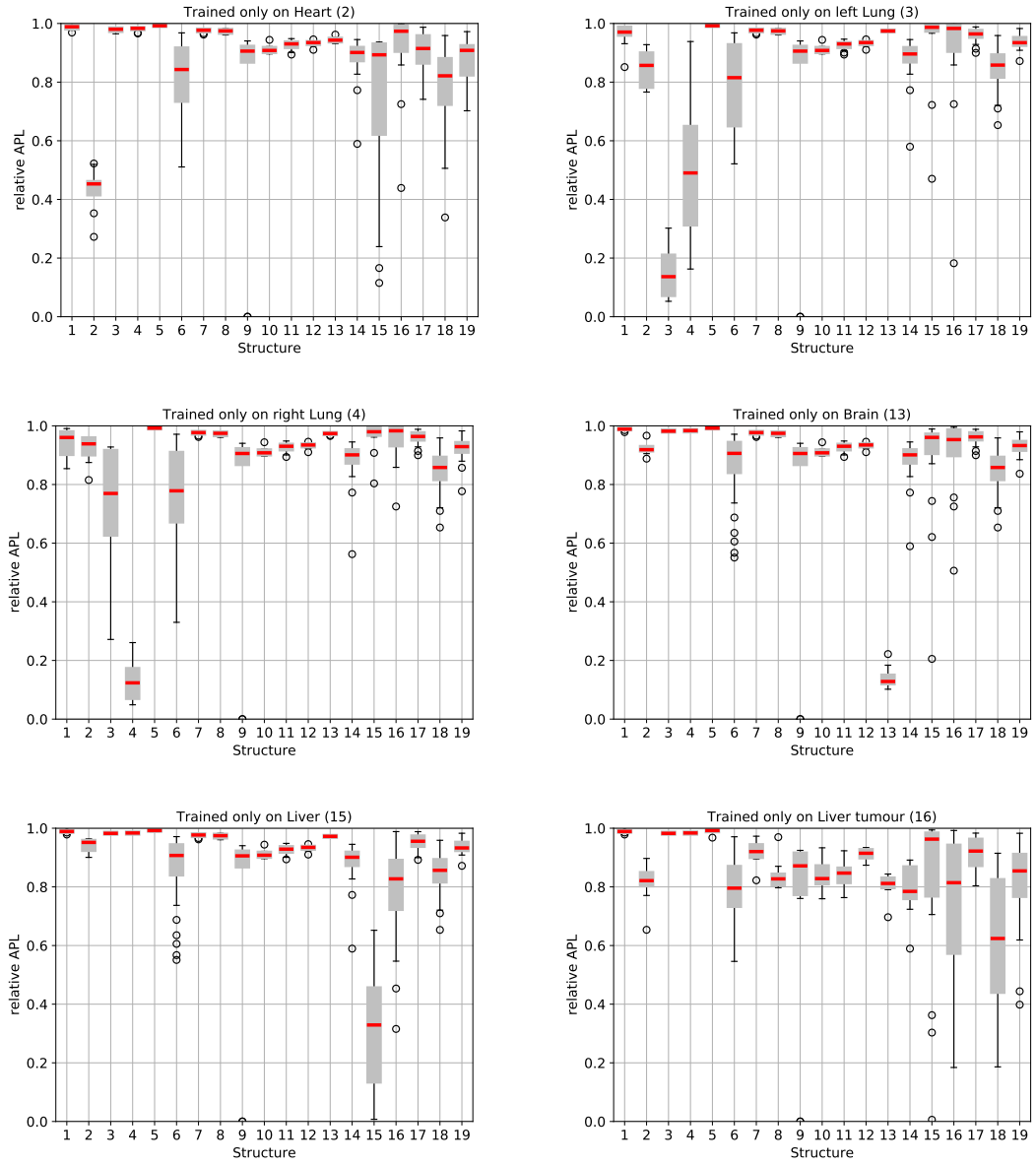


Figure B.11: Single-structure models were trained with a training set of only a single structure. Yet, these models were evaluated on their ability to predict other structures given a single contoured slice for context. Each of these models excels at the structure it has been trained on, but fails for most other structures.

B.5 Models trained using different number of input channels

In this section different input configurations to the contextual deep learning model are compared. This is to highlight why the three selected inputs as proposed in this work are necessary. Figure B.12 shows the performance of each model evaluated on 19 structures for 3D DSC and relative APL, respectively.
One input channel (target image only) fails, because it cannot identify which structure should be segmented. Note, that some structures (e.g. the brain) show a relatively high DSC. This is attributed to the way they appear in the training set relative to other structures. There are hardly any other structures in the training set that need to be contoured on the same slice as the brain itself, the model therefore learns structure specific segmentation rather than to segment based on given context. For 2 input channels, a relatively high DSC across all structures is observed. However, the spleen fails to be segmented by this model because it can only segment what has been included in the training set. With all 3 inputs the model is able to segment even structures excluded from the training set. This is attributed to the spatial link between the segmentation mask and the image slice that can be incorporated to identify the structure of interest on the target image.

In the following figures the different structures are referred to by the following numbers:

1 Esophagus
2 Heart
3 Lung (left)
4 Lung (right)
5 Spinal cord
6 Lung tumour
7 Neck (left)
8 Neck (right)
9 Submandibular gland (left)
10 Submandibular gland (right)
11 Parotid (left)
12 Parotid (right)
13 Brain
14 Brain tumour
15 Liver
16 Liver tumour
17 Pancreas
18 Pancreatic tumour
19 Spleen

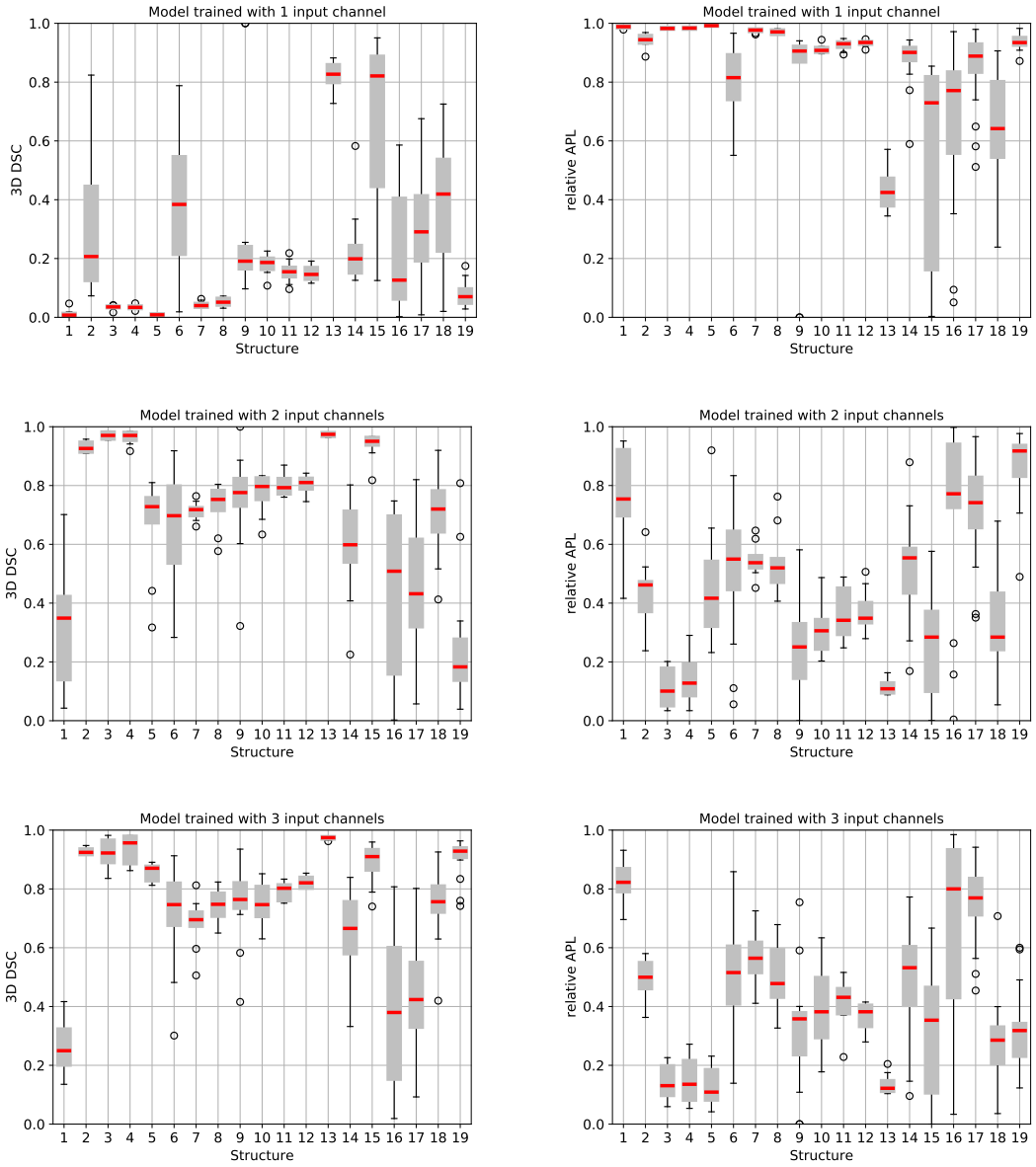


Figure B.12: Models trained with different number of input channels, evaluated for different structures using iterative prediction: a) 1 input channel: target image slice, b) 2 input channels: target image slice and mask of previously contoured image slice, c) 3 input channels: target image slice, mask of previously contoured image slice and the corresponding image slice. (This is the method proposed in Chapter 3). The spleen was excluded from the training set for all models. Performance shown for 3D DSC and relative APL.

C
Evaluation by Clinicians

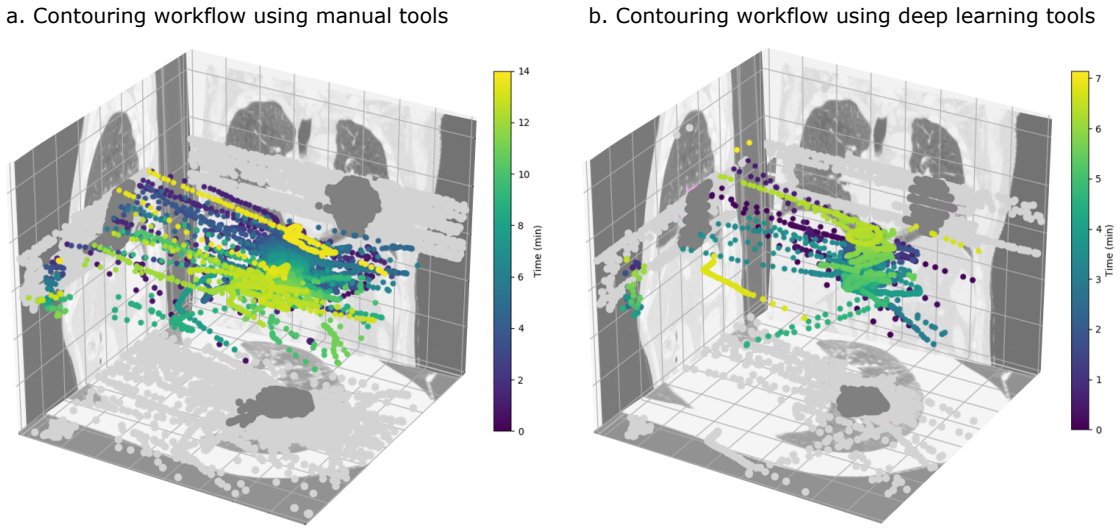


Figure C.1: Contouring workflow in 3D tracked on the axial plane (a) using manual tools and (b) the deep learning tool. The mouse movements are projected on the x-y (axial), y-z (sagittal) and x-z (coronal) plane. Light grey dots indicate regular mouse movement, whereas dark grey indicate mouse movement while the mouse button is pressed down (active contouring). The 3D scatter plot illustrates the workflow, with the dot color indicating the time after starting contouring for a given mouse position. The lines at the edge of the plot would extend to the things such as button or the different axial views.

D

Task-specific Fine-tuning for Contextual Deep Learning

Table D.1: Comments with regards to inter-observer variability by 3rd clinician.

Case	3D DSC	relative APL	Comment
1	0.00	1.00	ok, Clinician 1 and 2 contoured different lesions
2	0.54	0.85	ok, mostly overlapping, Clinician segmentation in opposite lung probably due to beam direction
3	0.20	0.91	ok
4	0.13	0.87	Right upper lobe: radiation induced fibrosis; right lower lobe: radiofrequency ablation; =>2 lesions, 2 therapies at different times, only the apical lesion is of interest here (SBRT)
5	0.53	0.83	ok
6	0.33	0.91	ok
7	0.14	0.79	ok
8	0.22	0.94	ok, other round foci known (on the right upper lobe), another lesion on the right middle lobe
9	0.11	0.95	ok, two targets, nodule (NSCLC) + hilar lymph nodes, both captured
10	0.00	1.00	ok, contoured different lesions
11	0.34	0.90	ok
12	0.27	0.96	ok, various irradiations; both have the same region of interest segmented. Further abnormalities in the left lower lobe, right middle lobe, right upper lobe
13	0.00	1.00	ok, contoured different lesions
14	0.38	0.90	ok
15	0.36	0.89	ok, Clinician 1: rather conservative, only compact, fibrostic tissue
16	0.00	1.00	ok
17	0.35	0.88	ok
18	0.29	0.86	ok, second carcinoma right upper lobe

Table D.2: Comments with regards to interobserver variability by 3rd clinician (Table D.1 continued).

Case	**3D DSC**	**relative APL**	**Comment**
19	0.22	0.85	ok, probably only parts of the ROI segmented (left upper lobe); further abnormalities in the right lower lobe
20	0.53	0.85	ok
21	0.21	0.94	ok
22	0.41	0.82	ok, two target volumes, both segmented
23	0.08	0.93	ok, further filling right lower lobe (soft tissue metastasis)
24	0.46	0.89	ok, further irradiation: right upper lobe later
25	0.00	1.00	ok, contoured different lesions
25	0.00	1.00	ok, contoured different lesions
26	0.12	0.99	ok
27	0.22	0.87	ok
28	0.24	0.96	ok
29	0.00	1.00	ok, second abnormality in the left upper lobe (2nd carcinoma)
30	0.40	0.89	Clinician 1 contoured the area of the SBRT, clinician 2 in the area of the primary tumor (right upper lobe: metastasis)
31	0.38	0.82	ok
32	0.58	0.79	ok
33	0.49	0.92	ok
34	0	1.0	ok
35	0.43	0.89	ok
36	0.55	0.79	ok
37	0	1.0	ok
38	0.31	0.95	ok