

# UC San Diego

## UC San Diego Electronic Theses and Dissertations

### Title

Multi-Stage and Multi-Target Data-Centric Approaches to Object Detection, Localization, and Segmentation in Medical Imaging

### Permalink

<https://escholarship.org/uc/item/2t4613rh>

### Author

Albattal, Abdullah

### Publication Date

2024

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA SAN DIEGO

**Multi-Stage and Multi-Target Data-Centric Approaches to Object Detection, Localization,  
and Segmentation in Medical Imaging**

A dissertation submitted in partial satisfaction of the  
requirements for the degree  
Doctor of Philosophy

in

Electrical Engineering (Signal and Image Processing)

by

Abdullah Albattal

Committee in charge:

Professor Truong Q. Nguyen, Chair  
Professor Cheolhong An, Co-Chair  
Professor Imanuel Lerman  
Professor Elliot McVeigh  
Professor Ramesh Rao

2024

Copyright  
Abdullah Albattal, 2024  
All rights reserved.

The dissertation of Abdullah Albattal is approved, and it is acceptable in quality and form for publication on microfilm and electronically.

University of California San Diego

2024

## DEDICATION

To my **mother**, Eman, a boundless source of unconditional love and tireless devotion; my source of power and guiding compass who taught me that kindness and doing good by others is what we should strive for.

To my **father**, Faisal, my anchor and pillar of strength, and the person who taught me the value of perseverance and the importance of staying true to one's principles.

To my **wife and son**, Reem and Faisal, my heart and my home, whose presence has been my sanctuary throughout this journey, filling my life with joy and purpose.

EPIGRAPH

**The greatest glory in living lies not in never falling, but in rising every time you fall.**

*—Nelson Mandela*

## TABLE OF CONTENTS

Dissertation Approval Page . . . . .	iii
Dedication . . . . .	iv
Epigraph . . . . .	v
Table of Contents . . . . .	vi
List of Figures . . . . .	x
List of Tables . . . . .	xiv
Acknowledgements . . . . .	xvii
Vita . . . . .	xxi
Abstract of the Dissertation . . . . .	xxiii
Chapter 1	
Introduction . . . . .	1
1.1 Background and Motivation . . . . .	3
1.1.1 Medical Imaging . . . . .	3
1.1.2 Object Detection, Tracking and Segmentation in Medical Imaging and The Challenges it Faces . . . . .	7
1.2 Research Objectives and Contribution of Thesis . . . . .	8
1.2.1 Real-Time Object Detection, Localization and Tracking in Ultrasound Scans . . . . .	9
1.2.2 Liver Lesion Detection and Segmentation in CT Scans . . . .	10
1.3 Organization of the Thesis . . . . .	12
Chapter 2	
Object Detection, Segmentation and Tracking in Medical Images . . . . .	14
2.1 Traditional Object Detection, Segmentation, and Tracking Approaches	14
2.2 Object Detection, Segmentation, and Tracking Using Deep Learning	16
2.3 Object Detection and Tracking in Ultrasound Scans . . . . .	17
2.4 CT Scans Registration . . . . .	19
2.5 Liver and Liver Lesion Detection and Segmentation . . . . .	21
Chapter 3	
Real-Time Object Detection and Tracking in Ultrasound Scans . . . . .	24
3.1 Problem Definition and Motivation . . . . .	24
3.1.1 Proposed Approaches and Contribution . . . . .	25
3.2 Methods . . . . .	27
3.2.1 Datasets and Data Preparation . . . . .	27
3.2.2 Data Augmentation . . . . .	29

	3.2.3	Evaluation Metrics . . . . .	31
3.3		Segmentation-Based Framework . . . . .	32
	3.3.1	The Framework and Its Structure . . . . .	32
	3.3.2	Experiments and Results . . . . .	36
3.4		Segmentation and Optical Flow Framework . . . . .	40
	3.4.1	The Framework and Its Structure . . . . .	40
	3.4.2	Experiments and Results . . . . .	41
3.5		Multi-Path Decoder UNet (MD UNet) . . . . .	43
	3.5.1	The Encoder-Multi-Path-Decoder Architecture . . . . .	44
	3.5.2	Experiments and Results . . . . .	46
3.6		Limitations and Future Prospective . . . . .	52
3.7		Conclusion . . . . .	53
Chapter 4		Efficient In-Training Adaptive Compound Loss Function Contribution Control for Medical Image Segmentation . . . . .	55
	4.1	Introduction . . . . .	55
	4.2	Background . . . . .	58
	4.3	Method . . . . .	60
	4.3.1	Adaptive Loss Contribution Control . . . . .	60
	4.3.2	Damped Adaptive Loss Contribution Control . . . . .	62
	4.4	Experiments and Results . . . . .	62
	4.4.1	Datasets . . . . .	62
	4.4.2	Implementation and Setup . . . . .	63
	4.4.3	Evaluation and Results . . . . .	64
	4.4.4	Extension to Other Segmentation Models . . . . .	67
	4.4.5	Sensitivity Analysis . . . . .	67
	4.4.6	Beyond Balancing The Precision and Recall . . . . .	68
	4.5	Conclusion . . . . .	69
Chapter 5		Multi-Phase CT Scan Registration . . . . .	70
	5.1	Introduction . . . . .	70
	5.2	Method . . . . .	73
	5.2.1	Overview of Approach . . . . .	73
	5.2.2	Deformable Registration Architecture and Loss Function . . . . .	74
	5.3	Experiment and Results . . . . .	75
	5.3.1	Dataset and Data Preparation . . . . .	76
	5.3.2	Evaluation Metrics . . . . .	77
	5.3.3	Slice Classifier Training and Results . . . . .	78
	5.3.4	Liver Segmentation Training and Results . . . . .	79
	5.3.5	Deformable Registration . . . . .	79
	5.4	Abnormal Registration Deformations Reduction . . . . .	82
	5.4.1	Method . . . . .	84
	5.4.2	Experiment and Setup . . . . .	86



	5.4.3	Evaluation and Results . . . . .	87
	5.5	Limitations and Future Prospective . . . . .	90
	5.6	Conclusion . . . . .	91
Chapter 6		Multi-Target and Multi-Stage Liver Lesion Segmentation and Detection in Multi-Phase CT Scans . . . . .	92
	6.1	Introduction . . . . .	92
	6.2	Background . . . . .	95
	6.3	Proposed Method . . . . .	97
	6.3.1	Pre-Processing and Augmentation . . . . .	99
	6.3.2	The Liver Lesion Segmentation Framework . . . . .	100
	6.3.3	Model Training and Segmentation Refinement . . . . .	103
	6.4	Experiments And Results . . . . .	106
	6.4.1	Datasets . . . . .	106
	6.4.2	Experimental Setup and Data Preperation . . . . .	107
	6.4.3	Evaluation and Results . . . . .	108
	6.4.4	Limitations and Future Prospective . . . . .	118
	6.5	Conclusion . . . . .	118
Chapter 7		Enhancing Lesion Detection and Segmentation Via Lesion Mask Selection from Multi-Specialized Model Predictions in CT Scans . . . . .	120
	7.1	Introduction . . . . .	120
	7.2	Background . . . . .	122
	7.3	Proposed Method . . . . .	124
	7.3.1	Pre-Processing and Augmentation . . . . .	125
	7.3.2	The Main Segmentation Model . . . . .	126
	7.3.3	The Small Lesion Focused Model . . . . .	128
	7.3.4	The Intensity-Based Features Used for Prediction Comparison . . . . .	129
	7.3.5	Segmentation Mask Prediction . . . . .	131
	7.3.6	Training And Loss Function . . . . .	132
	7.4	Experiments and Results . . . . .	134
	7.4.1	Datasets . . . . .	134
	7.4.2	Experimental Setup and Data Preperation . . . . .	135
	7.4.3	Evaluation and Results . . . . .	136
	7.4.4	Detection Performance . . . . .	141
	7.4.5	Ablation Studies . . . . .	142
	7.5	Extending The Prediction Selection Approach to Multiple Models . . . . .	145
	7.5.1	Multi-Model Corresponding Lesions Selection Algorithm . . . . .	147
	7.5.2	Enhancing The Recall of Small Lesions Via Size-Based Loss Weighting . . . . .	150
	7.5.3	Experiments and Results . . . . .	152
	7.6	Limitations and Future Prospective . . . . .	155
	7.7	Conclusion . . . . .	155

Bibliography . . . . . 157

## LIST OF FIGURES

Figure 1.1:	The rate of medical scans by age group per 1,000 people using computed tomography (CT), magnetic resonance imaging (MRI), ultrasound, and nuclear medicine in the United States (left). The rate of CT scans (middle) and MRI scans (right) per 1,000 people in several OECD countries. . . . .	2
Figure 1.2:	A timeline summarizing some of the major breakthroughs in medical imaging. The first ever X-ray image was taken in 1895, which was of a hand (public domain), the first MRI image taken in 1973 of two water tubes [1], and the first clinical CT scan. . . . .	5
Figure 1.3:	The most commonly used medical imaging modalities with their corresponding imaging method, typical target anatomical structures, diagnostic use, and example scans. Scans are from the National Institute of Health open-access MedPix <sup>®</sup> database. . . . .	6
Figure 1.4:	The overall structure of the multi-stage liver lesion segmentation framework from multi-phase CT scans. . . . .	11
Figure 3.1:	Ultrasound scans of the Vagus nerve with the ground truth (green) and predicted (yellow) bounding boxes using the first proposed framework. . .	28
Figure 3.2:	Example scans from the three datasets with the bounding box enclosing the object of interest. . . . .	29
Figure 3.3:	Overview of the proposed object detection framework with its four stages outlined. . . . .	33
Figure 3.4:	Object localization precision and recall for the 2 <sup>nd</sup> dataset (a) versus the number of subjects used for training (testing on a separate subject), and (b) versus the number of training images (3 subjects for training and 2 for testing. Images chosen randomly from training set). . . . .	38
Figure 3.5:	Heatmap (left) and histograms (right) of the lateral and axial distance offset of the true positive detections from the ground truth in millimeters. . . . .	39
Figure 3.6:	Overview of the five stages of the proposed object detection and tracking framework. . . . .	40
Figure 3.7:	The lateral and axial distance offset of the true positive detections from the ground truth in millimeters. Left: Tracking heatmap of offsets. Right: Detection heatmap of offsets. . . . .	43
Figure 3.8:	The proposed model structure (a) with the different building blocks outlined: The convolutional block in the encoder path (b), the convolutional upsampling block in the main decoder (c), and the multi-path convolutional upsampling block (d). . . . .	46
Figure 3.9:	Two ultrasound scans from the breast dataset. The mask maps are highlighted in green and the bounding box computed from the mask map in red for a benign (left) and malignant (right) lesion. . . . .	47

Figure 3.10:	Boxplots of the proposed architecture’s detection performance (in terms of IoU) compared to the four baseline models on the (a) breast benign lesion, (b) breast malignant lesion, (c) fetal head, and (d) Vagus nerve datasets. . . . .	48
Figure 3.11:	The proposed model IoU performance versus the number of training examples for each dataset. Each dataset’s last data point is for the model trained on the whole training set (vertical axis does not start from 0). . . . .	51
Figure 4.1:	An example slice from each of the five datasets we test the proposed approach on with the boundaries of lesions overlaid on the slices (a)-(e). In (a)-(c), the boundaries of the organ of interest are in yellow, while the boundaries of lesions are in red (a)-(e), overlaid on the slices. . . . .	57
Figure 4.2:	The result of recall and precision balancing for the five datasets using the proposed Algorithm 4.1 (a) and Algorithm 4.2 (b). . . . .	66
Figure 4.3:	The effect of varying the segmentation mask threshold value from 0.05 to 0.95 (after applying the model’s final nonlinearity) on the precision and recall. The results are for the UNet++ and DeepLabV3+ models when trained on the LiTS dataset (a) and (b), and the KiTS dataset (c) and (d). . . . .	66
Figure 5.1:	A CT scan slice from an arterial phase scan (A) and a delayed phase scan (B). Yellow arrows point to the bright arteries in the arterial scan and the lesion boundary in the delayed scan. . . . .	72
Figure 5.2:	The Multi-Phase CT scan registration pipeline is outlined with its three stages for the purpose of liver registration. . . . .	73
Figure 5.3:	The deformable registration model architecture and framework. The model takes two CT volumes as inputs and predicts a deformation field that spatially transform the moving volume onto the fixed volume to register them. The model can be trained in an unsupervised or semi-supervised manner. . . . .	75
Figure 5.4:	Example slices from four different subjects with chessboard overlay to demonstrate registration alignment across phases. In these slices, the fixed volume is from the arterial phase, and the moving volume is from the delayed phase. Top row before registration. Bottom row after registration. . . . .	82
Figure 5.5:	The distribution of the deformation error span for the predicted deformations using VoxelMorph and after correction using the proposed method. . . . .	83
Figure 5.6:	Qualitative results of the proposed error reduction approach on six example slices from the LiTS dataset. . . . .	88
Figure 5.7:	The number of repetitions effect (of model and input variation injection) on the correction algorithm performance. . . . .	89
Figure 5.8:	Qualitative results of the proposed error reduction approach on three example slices from the OASIS dataset. . . . .	90
Figure 6.1:	Three slices from different subjects in the liver lesion dataset (a)-(c). Each slice is acquired at a different phase and cropped to the liver region. The lesion region of interest is highlighted in a yellow bounding box. Within each of the cropped regions, the boundary of the lesion is outlined in yellow. . . . .	94

Figure 6.2:	The proposed framework structure with its different stages. The three outputs of stage 1 are converted to a heatmap that weights the input to stage 2. The outputs of stage 2 are concatenated with the CT image volume from each phase and then fed to stage 3. . . . .	98
Figure 6.3:	The architecture and structure of the proposed main segmentation model in stage 2 of the framework (a) with the different building blocks outlined (b): The convolutional, up-sampling convolutional, attention and feature fusion, stem, and output blocks. . . . .	99
Figure 6.4:	The structure of the Axial Projected Coarse Attention (APCA) module and the Gated Fine Attention module (GFA), which are the two components of the Coarse+Fine Feature Fusion & Attention Module. . . . .	102
Figure 6.5:	The largest axial slice of each liver lesion in the whole dataset (a), the training set (b), and the test set (c) overlaid onto one image at a scale of 1 mm. The boundary of these lesions in the same axial slice for the whole dataset (d), training set (e), and test set (f). . . . .	107
Figure 6.6:	Qualitative comparison of the proposed approach to the other four baseline models. The figure presents a side-by-side assessment of the different approaches' segmentation performance on lesions of different sizes, shapes and intensity characteristics. . . . .	112
Figure 6.7:	The 3D surface of predicted segmentation masks using our proposed approach (red) and ground truth masks (yellow) of lesions of different sizes and shapes. Each row visualizes the masks from 6 different perspectives. . . . .	113
Figure 6.8:	Qualitative demonstration of the proposed framework ability to improve the segmentation outcome versus 3-Phase segmentation models by incorporating leanings from each of the phases individually. . . . .	114
Figure 6.9:	Boxplots of the proposed segmentation model performance by subject (in terms of Dice score) compared to the other four baseline models when trained on the arterial (a), delayed (b) and venous (c) phases individually, and when trained using the three phases as 3-channel inputs (d). . . . .	116
Figure 6.10:	Distribution of segmentation performance by subject across multiple Dice score thresholds. . . . .	117
Figure 6.11:	Relative segmentation performance of models for each subject. The figure outlines the number of instances a model was above versus below average as well as in the top 5 versus bottom 5 when compared to the other models for each subject. . . . .	118
Figure 6.12:	The lesion detection F1 score by Dice score cutoff (a). The average precision (AP), recall (AR), and F1 scores (AF1) across the Dice score cutoff range of 0.1 to 0.9 (a). Model names keys: OS = Ours, MN = MedNext, and NN = nnUNet. The Dice score by lesion versus lesion volume (b). . . . .	119

Figure 7.1:	The proposed prediction selection approach. The 3D image patch is fed to both models where each generates a segmentation prediction map. Corresponding lesion pairs are selected from $Y_1$ and $Y_2$ based on a Dice score overlap of 0.5. . . . .	125
Figure 7.2:	The structure of the models used in the overall approach outlined in Fig. 7.1. In (a), the structure of the main segmentation model is outlined. In (b), the structures of the small lesion focused models. . . . .	127
Figure 7.3:	Qualitative comparison of the proposed approach to the other four benchmark models. For each of the three datasets, two examples are shown. The figure presents a side-by-side assessment of the segmentation performance on lesions of different sizes, shapes and intensity characteristics. . . . .	140
Figure 7.4:	Qualitative demonstration of the proposed approach ability to improve the segmentation outcome of the predictions from both models. The figure presents a side-by-side comparison of the overall approach to the main segmentation model as well as the SVE and HVE results. . . . .	141
Figure 7.5:	The detection F1 score of the proposed approach (SE) compared to SVE and HVE across different Dice score thresholds. The average precision (AP), recall (AR), and F1 score (AF1) across the whole range of Dice score thresholds (0.1 to 0.9) is annotated on each of the plots. . . . .	143
Figure 7.6:	The extended prediction selection approach for multiple models. The 3D image patch is fed to all models where each generates a segmentation prediction map. Corresponding lesion sets are selected from $Y_1$ , $Y_2$ , and $Y_M$ based on a Dice score overlap of 0.5. . . . .	146
Figure 7.7:	The correspondence selection approach demonstrated on a simulated example of lesion predictions from 3 models. The correspondence set from the perspective of each model prediction is shown together with the selected correspondence out of the 3 possible correspondences. . . . .	147
Figure 7.8:	Synthetically generated segmentation map with three lesions (a). The boundaries in the background that separate the lesions' regions (b). The LBWD map for universally chosen decay coefficients (c), and for individually chosen coefficients by lesion (d). . . . .	151
Figure 7.9:	The lesion detection F1 score and recall of the selection (SE - 2 Models) as well as the extended selection (SE - 5 Models) approaches across different Dice score thresholds for all lesions (a) and for small lesions (b). . . . .	154

## LIST OF TABLES

Table 3.1:	The proposed framework localization precision and recall. For experiments 1 and 2, the threshold for a true positive is an $\text{IoU} \geq 0.5$ . For experiment 3, a true positive is a distance error $\leq 2.5$ mm from the nerve center. The model with the best performance is boldfaced. . . . .	37
Table 3.2:	Mean and standard deviation of tracking error in mm for the different ultrasound tracking methods. Autonomous detection of objects and real-time capabilities are highlighted. . . . .	42
Table 3.3:	Computational complexity of the proposed model (MD UNet), UNet and UNet++. Inference time analysis was performed on Nvidia Tesla T4 GPU. In the table, Mul-Add stands for multiplication-addition, M for Millions, G for billions, and ms stands for millisecond. . . . .	44
Table 3.4:	Detection and localization performance results of the proposed model compared to the other four models we used as a benchmark. In the table, (MD) stands for multi-decoder, and (Avg.) stands for average, Std. Dev. stands for standard deviation, and IoU stands for intersection over union. . . . .	49
Table 3.5:	The Percentage of detections below two predetermined thresholds of the proposed model and the four benchmark models for the benign and malignant breast datasets, as well as the Vagus nerve dataset. A lower percentage of detections below the threshold indicates better model robustness. . . . .	50
Table 3.6:	IoU performance of the proposed architecture and UNet++ on the benign and malignant lesion datasets as the size of the architectures is reduced in terms of feature channels. The model with the best performance is boldfaced. . . . .	52
Table 3.7:	IoU performance of the proposed architecture on the benign and malignant lesion dataset as the number of decoders changes. In the table, Mul-Add stands for multiplication-addition, M stands for Millions, and G for billions. . . . .	52
Table 4.1:	The proposed framework precision and recall balancing performance as well as the number of epochs needed for tuning and training (TT Ep.). Pr. stands for precision, Rc. for recall, and Diff. for the absolute difference between the two. . . . .	65
Table 4.2:	The proposed framework precision and recall balancing performance when used with the UNet++, DeepLabV3+, and Attention UNet models on the LiTS and KiTS datasets. Pr. stands for precision, Rc. for recall, and Diff. for the absolute difference between the two. . . . .	67
Table 4.3:	The effect of the adjustment rate ( $\Delta$ ) on the number of epochs required to reach a recall-precision balance with a difference of less than 0.5% and the overall F1 score after using the respective adjustment rate. . . . .	68
Table 5.1:	The liver slice classifiers' performance on the LiTS dataset. . . . .	78
Table 5.2:	The best slice classifier model (ResNet-50) performance on the CT multi-phase registration dataset. . . . .	79

Table 5.3:	The proposed registration approach performance using the pre-trained deformable registration model across the different phases within the test set. $D_c$ stands for the Dice score (higher is better), and HD stands for the Hausdorff distance at the 95 <sup>th</sup> percentile by slice (lower is better). . . . .	80
Table 5.4:	The pre-trained Voxelmorph model performance on the test set under different field of view conditions. Hausdorff distance is thresholded at the 95 <sup>th</sup> percentile by slice. . . . .	81
Table 5.5:	Performance of the model as we increase the contribution of the Dice loss to the overall loss function by a factor of 2, 3, and 5 when compared to the original contribution. Hausdorff distance is thresholded at the 95 <sup>th</sup> percentile by slice. . . . .	81
Table 5.6:	Deformation error reduction performance of the proposed approach. The maximum deformation error and the 99 <sup>th</sup> percentile error are outlined in millimeters (averaged across subjects). The improvement in reducing each of these errors is also outlined as a percentage of the original error. . . . .	89
Table 6.1:	The proposed framework liver lesion segmentation performance on the multi-phase CT dataset. Section I outlines the results of just the segmentation model (stage 2 for our framework). Section II outlines the performance of the overall proposed framework. . . . .	110
Table 6.2:	The proposed framework brain tumor segmentation performance on the BraTS dataset by subject. Similar to Table 6.1, Section I outlines the results of just the segmentation model while Section II outlines the performance of the overall proposed framework. . . . .	115
Table 7.1:	The proposed approach segmentation performance on the three CT datasets. We compare the performance of the two models in our approach: The main segmentation model (Ours) and the small lesion focused model (Ours-HR) to the four benchmark models. . . . .	139
Table 7.2:	The proposed approach lesion detection performance on the three datasets. The precision (Pr.), recall (Rc.) and F1 score are shown for three Dice score thresholds. The average precision (AP), recall (AR), and F1 score (AF1) across the Dice score threshold range of 0.1 to 0.9 are included. . . . .	142
Table 7.3:	The Dice score performance of the proposed approach as the size of the bounding box margin changes for feature extraction of surrounding tissue, and as the Dice score threshold for correspondence changes for lesion predictions pairing in Algorithm 7.1. . . . .	144
Table 7.4:	The effect of varying the patch size and using the adaptive loss outlined in Algorithm 7.2 on the small lesion focused model segmentation performance for the LiTS dataset. . . . .	144



Table 7.5: The extended selection approach (Extended SE) segmentation performance on the clinical multi-phase liver lesion dataset. The performance of the main segmentation model (MSM) used in the 2-model selection approach (SE) is outlined as the threshold of the probability map is varied at inference. . . . . 153

## ACKNOWLEDGEMENTS

My sincerest and deepest gratitude is given to my advisor, Professor Truong Q. Nguyen, for his gracious and unwavering support and guidance throughout my journey as a Ph.D. student. His dedication to his students is truly remarkable, and I feel incredibly fortunate to have had the opportunity to work under his supervision. Whenever I faced challenges or had questions, he was always ready to offer his invaluable insights and advice. His prompt and thoughtful feedback, together with his invaluable experience and knowledge, has been instrumental in shaping my research and helping me navigate the complexities of my journey as a Ph.D. student. Beyond his academic guidance, Professor Nguyen has shown genuine care and concern for his students' well-being. He has created a supportive environment where, as a student in his lab, I felt valued and encouraged to grow both personally and professionally. I am truly grateful to Professor Nguyen for the remarkable journey and unforgettable experience. I am also deeply thankful to Professor Cheolhong An for his immense support, and for providing me with invaluable guidance and insightful feedback. I am truly fortunate to have him as a mentor, and his mentorship has not only enriched my research, but has also helped me grow my research and analytical thinking skills, and for that, I am sincerely grateful. My sincere appreciation and gratitude also go to Professor Imanuel Lerman for his guidance and support during our research collaboration. Professor Lerman's knowledge, enthusiasm, and willingness to engage in thought-provoking discussions have created an environment that fostered my abilities as a researcher. I am incredibly thankful for his mentorship. I would also like to extend my sincere appreciation to my committee members, Professor Elliot McVeigh and Professor Ramesh Rao for their valuable suggestions throughout the course of my work and for supervising me through my Ph.D. exams; their feedback and ideas have always been enlightening.

During my Ph.D., I had the privilege of collaborating with many great researchers and thinkers and I want to extend my appreciation to them. I want to thank Dr. Radhika Madhavan, Dr. Chen Du, Dr. Soan Duong, Dr. Van Ha Tang, Dr. Chanh Nguyen, Dr. Steven QH Truong,

Dr. Chien Phan, Yan Gong, Lu Xu, Timothy Morton, Yifeng Bu, and Quang Duc Tran. I had an exciting internship during my Ph.D. at Golden Set Analytics, where I was able to connect my love for tennis with my passion for computer vision. I am grateful to all my colleagues during my internship for helping me develop algorithms and integrate them into real-world products, expanding my experience and skills beyond academia.

At UCSD, there are many individuals and entities that have made my experience in the past five years truly meaningful and enjoyable. Among those are my lab mates at the video processing lab, where we had many fruitful and engaging discussions, whether it be on research topics or life in general; I am thankful to all of them. I am also grateful to have had the chance to work with the Engaged Teaching Hub, which allowed me to pursue my passion for teaching and mentoring while growing as an instructor by expanding my knowledge and experience on equitable and student-centered teaching. I also want to sincerely thank all my professors and teachers, both at UCSD and at other institutes, for their relentless work and for inspiring me to continue with my studies and reach this stage in my education. To the staff at the ECE department and the international students and programs office, thank you for your hard work and efforts that allow us to pursue our studies.

I also want to express my sincere gratitude and appreciation to the Kingdom of Saudi Arabia, represented by King Fahd University of Petroleum and Minerals (KFUPM), for sponsoring me to pursue my Ph.D. and for their tremendous support, for which I will be forever grateful. May I have the strength and will to use the knowledge, experience, and skills I gained during my Ph.D. to benefit my country, people, and community.

Finally, I want to express my appreciation and gratitude to my family, especially my beloved mother and father, for believing in me and for their unconditional love, support, encouragement, and sacrifices. As a parent myself now, I know how hard it must have been for you to be separated halfway across the world from your son, and I am forever grateful for all the sacrifices you made to enable me to pursue my studies. To my wife, my rock and forever love, you were

there every step of the way; supporting and encouraging me to keep going even when I sometimes could not see the light at the end of the tunnel, so thank you from the bottom of my heart; you complete me. To my son, nothing has brought me more joy than seeing you laugh and be happy, so thank you for filling my life with joy and meaning.

There are many professors, staff members, colleagues, friends, and family members who are not mentioned by name in this acknowledgment, yet they have made a significant impact on my life before and during my Ph.D. that enabled me to embark on this enriching and fruitful journey; to all of them thank you very much.

Chapter 3 is, in full, based on the materials as they appear in the publication of “A CNN segmentation-based approach to object detection and tracking in ultrasound scans with application to the Vagus nerve detection”, Abdullah F. Al-Battal; Yan Gong; Lu Xu; Timothy Morton; Chen Du; Yifeng Bu; Imanuel R Lerman; Radhika Madhavan; Truong Q. Nguyen In Proceedings of International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC), 2021, and “Object detection and tracking in ultrasound scans using an optical flow and semantic segmentation framework based on convolutional neural networks”, Abdullah F. Al-Battal; Imanuel R Lerman; Truong Q. Nguyen In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2022, as well as the material as it appears in “Multi-path decoder U-Net: a weakly trained real-time segmentation network for object detection and localization in ultrasound scans”, Abdullah F. Al-Battal; Imanuel R Lerman; Truong Q. Nguyen, In Computerized Medical Imaging and Graphics journal, 2023. The dissertation author was the primary investigator and author of these papers.

Chapter 4 is, in full, based on the materials as they appear in the publication of “Efficient in-training adaptive compound loss function contribution control for medical image segmentation”, Abdullah F. Al-Battal; Soan T. M. Duong; Chanh D. Tr. Nguyen; Steven Q. H. Truong; Chien Phan; Truong Q. Nguyen; Cheolhong An In International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC), 2024. The dissertation author was the primary investigator

and author of this paper.

Chapter 5 is, in part, based on the materials as they appear in “A Learning-Free Approach to Mitigate Abnormal Deformations in Medical Image Registration”, Abdullah F. Al-Battal; Soan T. M. Duong; Chanh D. Tr. Nguyen; Steven Q. H. Truong; Chien Phan; Truong Q. Nguyen; Cheolhong An, submitted to the Workshop on Biomedical Image Registration of the International Conference on Medical Image Computing and Computed Assisted Intervention (MICCAI), 2024. The dissertation author was the primary investigator and author of this paper.

Chapter 6 is, in part, based on the materials as they appear in “Multi-target and multi-stage liver lesion segmentation and detection in multi-phase computed tomography scans”, Abdullah F. Al-Battal; Soan T. M. Duong; Van Ha Tang; Quang Duc Tran; Steven Q. H. Truong; Chien Phan; Truong Q. Nguyen; Cheolhong An, submitted to the Medical Image Analysis journal, 2024. The dissertation author was the primary investigator and author of this paper.

Chapter 7 is, in part, based on the materials as they appear in “Enhancing lesion detection and segmentation via lesion mask selection from multi-specialized model predictions in CT scans of the liver and kidney”, Abdullah F. Al-Battal; Van Ha Tang; Quang Duc Tran; Steven Q. H. Truong; Chien Phan; Truong Q. Nguyen; Cheolhong An, submitted to the Computers in Biology and Medicine journal, 2024 as well as the material as it may appear in the currently being prepared submission to the Machine Learning in Medical Imaging workshop of the International Conference on Medical Image Computing and Computed Assisted Intervention (MICCAI), 2024. The dissertation author was the primary investigator and author of these papers.

## VITA

### EDUCATION

- 2024 Ph. D. in Electrical Engineering (Signal and Image Processing), University of California San Diego, La Jolla, California, The United States of America
- 2013 M. S. in Electrical Engineering, University of Southern California, Los Angeles, California, The United States of America
- 2010 B. S. in Electrical Engineering *with 1<sup>st</sup> honors distinction*, King Fahd University of Petroleum and Minerals, Dhahran, The Kingdom of Saudi Arabia

### EXPERIENCE

- 2014-present Lecturer, Electrical Engineering Department, King Fahd University of Petroleum and Minerals
- 2011-2014 Graduate Assistant, Electrical Engineering Department, King Fahd University of Petroleum and Minerals

### PUBLICATIONS

**Abdullah F. Al-Battal**, Van Ha Tang, Quang Duc Tran, Steven Q. H. Truong, Chien Phan, Truong Q. Nguyen, and Cheolhong An. “Enhancing lesion detection and segmentation via lesion mask selection from multi-specialized model predictions in CT scans of the liver and kidney”, submitted to *Computers in Biology and Medicine*, 2024.

**Abdullah F. Al-Battal**, Soan T. M. Duong, Van Ha Tang, Quang Duc Tran, Steven Q. H. Truong, Chien Phan, Truong Q. Nguyen, and Cheolhong An. “Multi-target and multi-stage liver lesion segmentation and detection in multi-phase computed tomography scans”, submitted to *Medical Image Analysis*, 2024.

**Abdullah F. Al-Battal**, Soan T. M. Duong, Chanh D. Tr. Nguyen, Steven Q. H. Truong, Chien Phan, Truong Q. Nguyen, and Cheolhong An. “Efficient in-training adaptive compound loss function contribution control for medical image segmentation”, accepted for publication in *2024 46th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, 2024.

**Abdullah F. Al-Battal**, Soan T. M. Duong, Chanh D. Tr. Nguyen, Steven Q. H. Truong, Chien Phan, Truong Q. Nguyen, and Cheolhong An. “A Learning-Free Approach to Mitigate Abnormal Deformations in Medical Image Registration”, submitted to the *Workshop on Biomedical Image Registration of the International Conference on Medical Image Computing and Computed Assisted Intervention (MICCAI)*, 2024.

Jianzhi Long, **Abdullah F. Al-Battal**, Shiwei Jin, Jing Zhang, Dacheng Tao, Imanuel Lerman, and Truong Q. Nguyen. “Localizing scan targets from human pose for autonomous lung ultrasound imaging”, accepted for publication in *Intelligent Systems Conference (IntelliSys)*, 2024.

**Abdullah F. Al-Battal**, Imanuel R. Lerman, and Truong Q. Nguyen. “Multi-path decoder U-Net: a weakly trained real-time segmentation network for object detection and localization in ultrasound scans”, *Computerized Medical Imaging and Graphics*, 107, pp. 102205, 2023.

Jamison Burks, Yifeng Bu, **Abdullah F. Al-Battal**, Truong Q. Nguyen, and Imanuel R. Lerman. “ID: 212135 Noninvasive human immunotyping through machine learning using the autonomic nervous system”, *Neuromodulation*, 26(4), pp. S81, 2023.

Yifeng Bu, **Abdullah F. Al-Battal**, Jamison Burks, Truong Q. Nguyen, and Imanuel R. Lerman. “ID: 211000 Leveraging machine learning for early septicemia detection from noninvasive analysis of the Autonomic nervous system”, *Neuromodulation*, 26(4), pp. S140, 2023.

Sachintha R. Brandigampala, **Abdullah F. Al-Battal**, and Truong Q. Nguyen. “Data augmentation methods for object detection and segmentation in ultrasound scans: An empirical comparative study”, *2022 IEEE 35th International Symposium on Computer-Based Medical Systems (CBMS)*, pp. 288-291, 2022.

**Abdullah F. Al-Battal**, Imanuel R. Lerman, and Truong Q. Nguyen. “Object detection and tracking in ultrasound scans using an optical flow and semantic segmentation framework based on convolutional neural networks”, *2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1096-1100, 2022.

**Abdullah F. Al-Battal**, Yan Gong, Lu Xu, Timothy Morton, Chen Du, Yifeng Bu, Imanuel R. Lerman, Radhika Madhavan, and Truong Q. Nguyen. “A CNN segmentation-based approach to object detection and tracking in ultrasound scans with application to the Vagus nerve detection”, *2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, pp. 3322-3327, 2021.

Wail A. Mousa and **Abdullah F. Al-Battal**. “The design of a homotopy-based 1-D seismic FIR f-x wavefield extrapolation filters”, *Journal of Applied Geophysics*, 173, pp. 103933, 2020.

Wail A. Mousa and **Abdullah F. Al-Battal**. “Computationally Efficient Phase Shift Plus Interpolation Seismic Migration Method”, *IEEE Geoscience and Remote Sensing Letters*, 17(5), pp. 775-778, 2019.

**Abdullah F. Al-Battal** and Wail A. Mousa. “The Design of 2-D Explicit Depth Extrapolators Using the Cauchy Norm”, *IEEE Transactions on Geoscience and Remote Sensing*, 55(5), pp. 3029-3036, 2017.

ABSTRACT OF THE DISSERTATION

**Multi-Stage and Multi-Target Data-Centric Approaches to Object Detection, Localization, and Segmentation in Medical Imaging**

by

Abdullah Albattal

Doctor of Philosophy in Electrical Engineering (Signal and Image Processing)

University of California San Diego, 2024

Professor Truong Q. Nguyen, Chair  
Professor Cheolhong An, Co-Chair

Object detection, localization, and segmentation in medical images are essential in several medical procedures. Identifying abnormalities and anatomical structures of interest within these images remains challenging due to the variability in patient anatomy, imaging conditions, and the inherent complexities of biological structures. To address these challenges, we propose a set of frameworks for real-time object detection and tracking in ultrasound scans and two frameworks for liver lesion detection and segmentation in single and multi-phase computed tomography (CT) scans. The first framework for ultrasound object detection and tracking uses a segmentation



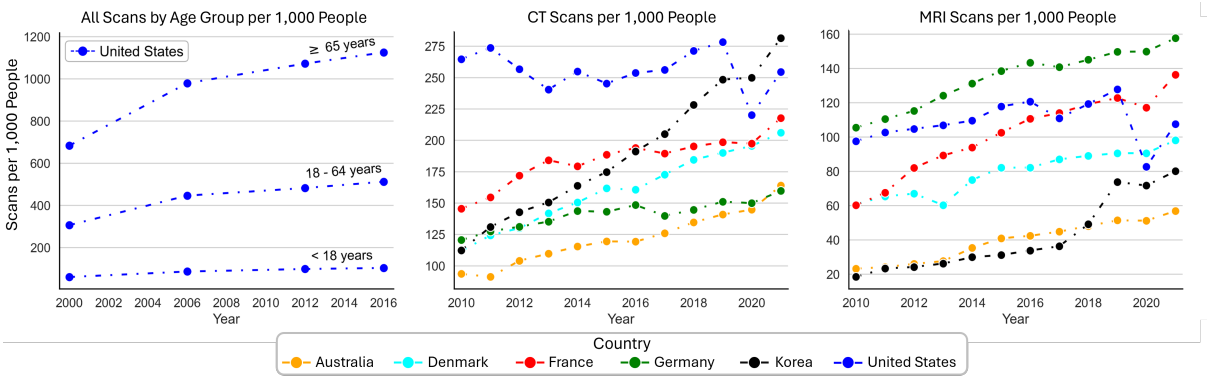
model weakly trained on bounding box labels as the backbone architecture. The framework outperformed state-of-the-art object detection models in detecting the Vagus nerve within scans of the neck. To improve the detection and localization accuracy of the backbone network, we propose a multi-path decoder UNet. Its detection performance is on par with, or slightly better than, the more computationally expensive UNet++, which has 20% more parameters and requires twice the inference time. For liver lesion segmentation and detection in multi-phase CT scans, we propose an approach to first align the liver using liver segmentation masks followed by deformable registration with the VoxelMorph model. We also propose a learning-free framework to estimate and correct abnormal deformations in deformable image registration models. The first framework for liver lesion segmentation is a multi-stage framework designed to incorporate models trained on each of the phases individually in addition to the model trained on all the phases together. The framework uses a segmentation refinement and correction model that combines these models' predictions with the CT image to improve the overall lesion segmentation. The framework improves the subject-wise segmentation performance by 1.6% while reducing performance variability across subjects by 8% and the instances of segmentation failure by 50%. In the second framework, we propose a liver lesion mask selection algorithm that compares the separation of intensity features between the lesion and surrounding tissue from multi-specialized model predictions and selects the mask that maximizes this separation. The selection approach improves the detection rates for small lesions by 15.5% and by 4.3% for lesions overall.

# Chapter 1

## Introduction

Medical imaging is essential in many medical diagnostic and therapeutic procedures, allowing doctors to visualize, examine, and observe various anatomical structures, abnormalities, physiological interactions, and disease progression within the body. In radiology, imaging is key for detecting and diagnosing many critical diseases such as cancer, cardiovascular diseases, lung infections and neurological disorders. Beyond critical diseases, medical imaging is also used for detecting and monitoring bone fractures, tendon sprains, arthritis, and kidney stones. During pregnancy, ultrasound is used for monitoring the growth of the fetus as well as checking for any possible health issues that might require medical interventions. In research fields such as psychology and neuroscience, medical imaging modalities such as functional magnetic resonance imaging (fMRI) are used to examine and analyze brain functionality, and neurological connectivity and interactions. The significance of medical imaging in clinical settings is evident by its continually increased use in medical examinations as shown in Fig. 1.1.

Currently, most medical imaging is implemented using computerized systems and is digitized including acquisition, reconstruction, visualization, and analysis. Digitized medical imaging introduced several improvements to the field overall. Advanced acquisition and reconstruction approaches helped reduce radiation exposure from X-rays and computed tomography



**Figure 1.1:** The rate of medical scans by age group per 1,000 people using computed tomography (CT), magnetic resonance imaging (MRI), ultrasound, and nuclear medicine in the United States (left) [2]. The rate of CT scans (middle) and MRI scans (right) per 1,000 people in several OECD (Organisation for Economic Co-operation and Development) countries [3].

(CT) while improving image quality at the same time [4]<sup>1</sup>. Computer-assisted imaging and image analysis advanced several fields such as radiometrics [7], surgical planning and navigation [8], and drug discovery and analysis [9]. Furthermore, at-scale computer-assisted diagnostic (CAD) systems have become possible with the digitization of medical imaging and are used to support clinicians in examining, analyzing, and interpreting images. They are used to identify anatomical structures of interest in addition to abnormalities and diseases such as the presence of lesions [10, 11]. The clear potential for CAD to improve patient outcomes has made it into one of the major research fields in medical imaging with hundreds of articles published in scientific journals and conferences annually on the subject [12]. Throughout the past decade, artificial intelligence-based medical image analysis and CAD algorithms, specifically deep learning ones, have dominated most of the development in the field, which should come as no surprise due to their powerful learning capabilities in extracting features from images for classification, detection, segmentation, and tracking purposes [13, 14, 15, 16, 17, 18, 19]. These algorithms have the potential to revolutionize medical imaging by providing clinicians with more accurate information about underlining diseases and abnormalities for improved diagnosis and interventions. Another

<sup>1</sup>It is worth noting that computerized imaging can also lead to the overall increased exposure to radiation due to the speed and ease at which the scans can be completed, increasing repetitions of exposure [5, 6].

aspect that might not be mentioned often when these algorithms are in discussion is their ability to increase access to healthcare and unify outcomes across patients of different backgrounds and identities taking healthcare a step forward towards equity and inclusion [20, 21]<sup>2</sup>.

Our objective in this work is on the development of deep learning approaches for the purpose of 1) object detection, localization and tracking in ultrasound scans, and 2) liver lesion detection and segmentation in single- and multi-phase CT scans. However, before we dive into the technical presentation and discussion of our work, we believe a brief review of the background and motivation from the perspective of medical imaging in general, and the perspective of object detection, segmentation and tracking in medical imaging is needed to contextualize and situate our work.

## **1.1 Background and Motivation**

### **1.1.1 Medical Imaging**

Throughout history, studying the internal structure of human bodies while still alive was virtually non-existent. Aside from invasive surgical interventions or dissections, there were no methods available to do so; while the fear of bleeding, pain, and infections as well as cultural barriers limited those interventions [23]. All this changed in the late 19<sup>th</sup> century when medical imaging became possible through the use of direct electromagnetic radiation to create clear, contrasting pictures—although not by today’s standards—between bones and soft tissue [24], a technique that is known to us now as X-ray. Prior to that, clinicians and researchers used external observations and their imaginations to formulate ideas on how the inner parts of our bodies are structured and work together. The invention of X-ray by the German physicist Wilhelm Röntgen in 1895 enabled physicians for the first time to have a peek inside the human body and see the

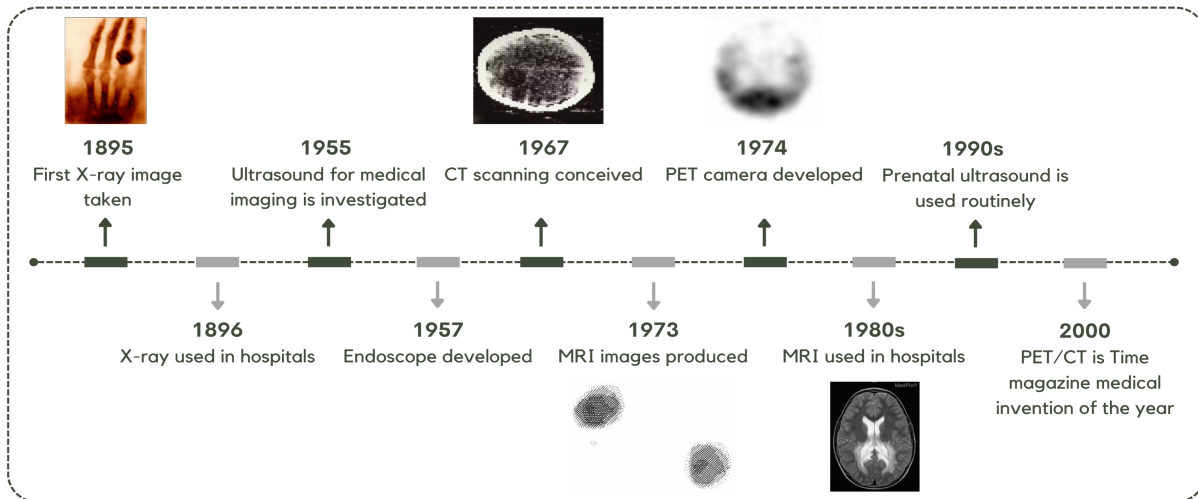
---

<sup>2</sup>Without careful planning of AI methods in healthcare through the assurance of representation, and bias analysis and reduction, AI methods can instead amplify inequality and increase disparities in the healthcare system [22].

different anatomical structures without invasive interventions. This was a significant breakthrough in the medical field and enabled doctors to diagnose a variety of conditions such as broken bones, lung diseases, and gastrointestinal disorders without invasive interventions.

Since the discovery of X-ray, medical imaging has continued to evolve with the development of new imaging modalities such as computed tomography (CT), magnetic resonance imaging (MRI), ultrasound, positron emission tomography (PET) and several others. A timeline summarizing the main breakthroughs in medical imaging is outlined in Fig. 1.2. Each of these modalities scan the body using various and differing techniques to generate a representation of the internal structure of the human body that is different based on the technique used [25]; providing clinicians with a variety of options to assist in diagnostic and therapeutical procedures. These imaging techniques have revolutionized anatomical and physiological studies by allowing doctors to visualize internal structures, detect abnormalities, and monitor disease progression without the need for invasive procedures that would impact patients' health negatively and might not be possible, to begin with.


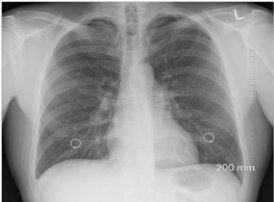
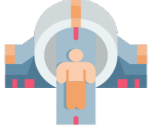


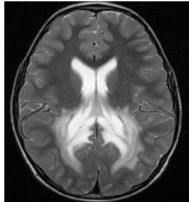
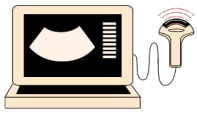
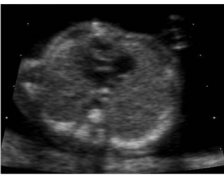
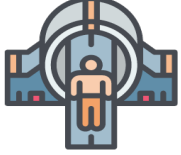
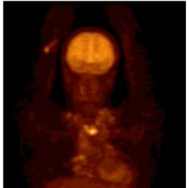
Medical imaging has also led to significant advances in the fields of cardiology, neurology, and oncology, among others. For example, MRI machines use strong magnetic fields and radio waves to generate detailed images based on the magnetic resonance of dipoles in tissues, making them ideal for soft tissue visualization and diagnosis, such as the brain, spinal cord, joints, tendons and muscles [25]. On the other hand, CT scanning employs X-ray to generate cross-sectional projections that are reconstructed to produce images, allowing for an exceptionally high-resolution visualization of bony structures and organs like the lungs, liver, and other anatomical structures in the abdomen [28]. For ultrasound, high-frequency sound waves are used to generate real-time images of organs and tissues, making it particularly useful for examining fetal development and the cardiovascular system. Fig. 1.3 shows examples of X-Ray, CT, MRI, ultrasound, and PET scans together with a summary of their corresponding imaging method, typical target anatomical structures and diagnostic use.



**Figure 1.2:** A timeline summarizing some of the major breakthroughs in medical imaging. The first ever X-ray image was taken in 1895, which was of a hand (public domain), the first MRI image taken in 1973 of two water tubes [1], and the first clinical CT scan, which was of the brain taken in 1971 [26] are shown. An example of a PET scan of the brain taken by the first clinical PET machine (PET III) in 1975, and an example scan of the brain from a modern MRI machine (from the MedPix<sup>®</sup> database [27]).

Medical imaging plays a significant role in clinical settings by empowering clinicians to investigate patients anatomically and physiologically through non-invasive or minimally invasive means. Not only is it used as an essential part of the standard of care for many diagnostic and therapeutic procedures [29, 30], it is also used in many clinical trials and research studies [31]. When coupled with algorithms and methods to assist clinicians in detecting, and tracking objects of interest and abnormalities within scans, medical images have the potential to further improve patient outcomes through consistent diagnostic and intervention performance across patients and clinicians alike.

# Medical Imaging Modalities

<p><b>X-Ray</b></p> 	<p><b>Imaging Method:</b> Ionizing radiation</p> <p><b>Target Anatomical Structures:</b> Bones, chest, abdomen</p> <p><b>Used to Diagnose:</b> Fractures, pneumonia, bowel obstruction</p>	<p><b>Example Scan</b></p> 
<p><b>CT</b></p> 	<p><b>Imaging Method:</b> Cross-Sectional Ionizing radiation</p> <p><b>Target Anatomical Structures:</b> Head, chest, abdomen, pelvis, spine</p> <p><b>Used to Diagnose:</b> Traumatic injuries, cancer staging, stroke</p>	<p><b>Example Scan</b></p> 
<p><b>MRI</b></p> 	<p><b>Imaging Method:</b> Magnetic Resonance</p> <p><b>Target Anatomical Structures:</b> Brain, spine, joints, abdomen, pelvis</p> <p><b>Used to Diagnose:</b> Tumors, herniated discs, ligament injuries</p>	<p><b>Example Scan</b></p> 
<p><b>Ultrasound</b></p> 	<p><b>Imaging Method:</b> High-frequency sound waves</p> <p><b>Target Anatomical Structures:</b> Abdomen, pelvis, heart, blood vessels, Fetus</p> <p><b>Used to Diagnose:</b> Pregnancy, gallstones, vein thrombosis</p>	<p><b>Example Scan</b></p> 
<p><b>Pet</b></p> 	<p><b>Imaging Method:</b> Radioactivity Detection of Tracers</p> <p><b>Target Anatomical Structures:</b> Whole body, heart, brain</p> <p><b>Used to Diagnose:</b> Cancer, heart disease, alzheimer</p>	<p><b>Example Scan</b></p> 

**Figure 1.3:** The most commonly used medical imaging modalities with their corresponding imaging method, typical target anatomical structures, diagnostic use, and example scans. Scans are from the National Institute of Health open-access MedPix® database.

### **1.1.2 Object Detection, Tracking and Segmentation in Medical Imaging and The Challenges it Faces**

Object detection, tracking, and segmentation methods in medical imaging allow for the targeted characterization and representation of objects of interest within a scan. This has many benefits ranging from the identification and detection of lesions and tumors to location and size monitoring of organs and other anatomical structures', which can assist in therapeutic procedures such as radiation therapy or nerve stimulation [32, 33, 34, 35]. In addition, researchers have demonstrated that these methods can be used for many applications ranging from critical ones such as cardiac function analysis in ultrasound and MRI scans [36] to bone fracture detection and quantification in X-rays [37]. In its simplest and earliest forms, these methods started with simple thresholding and edge detection filtering approaches to separate objects within a scan based on their intensity values [38] or the changes in these values [39].

These methods developed over time to be more sophisticated and generalize better to account for variations in images such as intensity variations and noise which hinders the performance of thresholding and edge detection methods. Methods such as active region growing [40], ROI matching [41], and histogram of oriented gradients [42], which we will discuss in more detail in Chapter 2. Although several of these traditional methods (as opposed to deep learning-based approaches) were proposed to overcome the challenge of generalization, they still performed at a level that did not provide significant added value for clinicians and doctors since they could not provide them with information beyond quantitative measurements that experienced doctors could manually find themselves. They also required manual intervention and calibration while in operation and could not operate autonomously.

Most of the current advances over the past decade in the detection, segmentation and tracking of objects in medical images are based on deep learning models that use convolutional neural networks. These models have demonstrated significantly improved abilities in feature



extraction and decision making [43, 14, 15] such as whether a scan contains a lesion or not and if it does, where that lesion is located [44]. Although superior when compared to traditional methods in their ability to generalize and accurately perform the task they were trained on, deep learning methods suffer from a set of drawbacks. First, they require relatively large annotated datasets, which are expensive to generate in the medical field as they require expert annotations and labeling. If trained on smaller datasets, they can suffer from overfitting and the inability to generalize beyond the data they were trained on. These models are also hard to interpret and can give incorrect predictions with high confidence, which is why they are usually referred to as black boxes where users typically do not have much control and understanding of the state of the model beyond its input and the predicted output. Deep learning models learn what they are trained on, and if the data is biased or lacks representation from certain groups, then these trained models will perform poorly when presented with data from these groups. Deep learning models also require extensive computational resources for training and, depending on the model complexity and size, for deployment and inference as well. Finally, One of the major challenges that face deep learning models for detection and segmentation purposes in medical images is class imbalance where the target anatomical structure or abnormality, such as lesions, are severely underrepresented with respect to the overall scan. In many instances, these segmentation and detection targets are orders of magnitude smaller than the overall size of the scan.

## **1.2 Research Objectives and Contribution of Thesis**

The aims and objectives of this dissertation are as follows:

1. A real-time object detection, localization and tracking framework for ultrasound imaging based on convolutional neural networks that can perform well and generalize when trained from a small dataset.
2. A multi-stage and multi-target framework for liver lesion segmentation in multi-phase CT

scans based on convolutional neural networks that can combine learning from individual phases with learning from combined phases to improve the segmentation and detection of lesions as well as reduce performance variability across subjects.

3. A lesion selection approach that compares the predictions from multiple specialized models lesion-wise, evaluating lesions based on intensity features separation between the lesions and surrounding tissue within each of the predictions; selecting the prediction that maximizes feature separation for liver lesion segmentation and detection in both single- and multi-phase CT scans to improve detection rates and reduce the number of missed lesions.

### **1.2.1 Real-Time Object Detection, Localization and Tracking in Ultrasound Scans**

We designed a real-time object detection framework that is designed to autonomously detect, identify, and localize a specific anatomical structure in ultrasound scans. The designed method uses a weakly supervised and modified UNet convolutional neural network (CNN), which is an encoder-decoder architecture designed for segmentation tasks, as its backbone detection and localization algorithm [18]. The proposed approach is designed to autonomously assist sonographers in real-time to enhance their ability to detect and track objects of interest during scanning sessions.

We then designed an improved framework that is also capable of real-time object detection and localization using the same weakly trained segmentation-based UNet accompanied by an optical flow CNN that can track the target object location across frames by estimating its movements. The framework can detect and localize a target anatomical structure within an ultrasound scan field of view with an average localization error of less than 1.25 mm, and a tracking error of less than 0.75 mm. Finally, we designed a weakly supervised backbone segmentation network that incorporates multi-path decoders, which improves object detection accuracy by up to 7%

while maintaining low computational complexity for real-time operation.

The proposed approaches outperform state-of-the-art object detection models in detecting and localizing anatomical structures of interest within ultrasound scans. They do not require a large training set and are trained using bounding box labels, which are easier and significantly cheaper to generate than pixel-wise segmentation annotations.

### **1.2.2 Liver Lesion Detection and Segmentation in CT Scans**

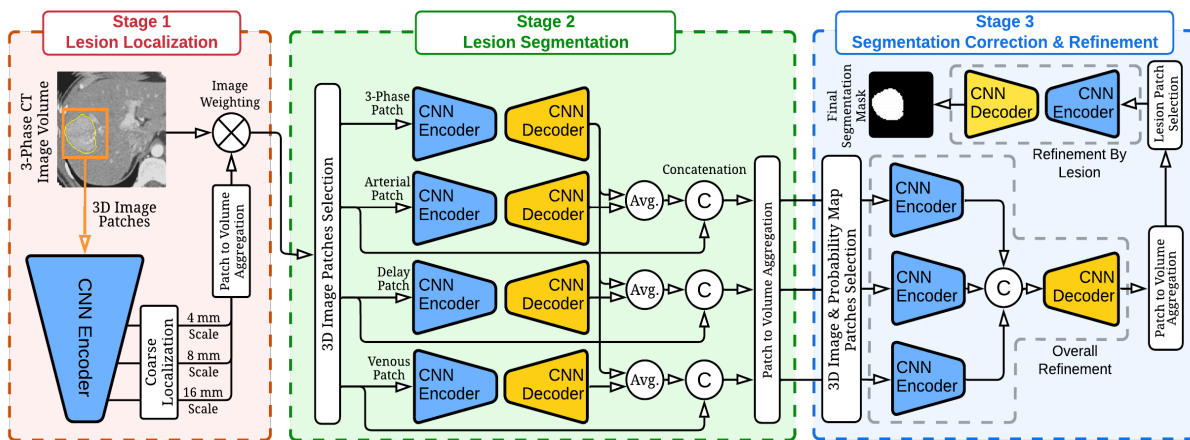
Detecting and segmenting lesions within the liver in CT scans is a challenging problem that is still largely unsolved. Current approaches have average recall and precision that hovers around 40 to 50% [45] for lesion detection and segmentation. These lesions within the liver are of varying sizes, shapes, and locations within the liver. They are also often of similar texture and intensity to the healthy liver tissue that surrounds them adding to the complexity of the problem [45]. Most of the proposed methods that aimed to solve this problem approached it from a segmentation perspective where a segmentation convolutional neural network is designed and trained to segment the lesions directly or in a cascaded framework where the liver is first segmented before lesions to reduce the search space for the lesion segmentation model [46, 47, 48, 49]. Methods that approach the problem with 3D segmentation models achieved on average better segmentation and detection performance when compared with methods that approach the problem on a 2D per-slice matter. This allows contextual information from different slices within the CT volume to be used for the prediction of the segmentation decision [45, 50, 48].

Prior to detecting and segmenting lesions within the liver, we employ three approaches that improve the ability of the lesion segmentation frameworks we propose to identify the lesions within the liver. These approaches are:

1. A slice classifier to identify all the slices of a CT scan that contains the liver with a post-processing stage that utilizes structural information on the continuity of the liver to minimize false negatives.

2. A 3D liver segmentation model that isolates the liver within the CT volume, which is implemented on the slices that were identified by the slice classifier.
3. A deformable liver registration approach that aligns the liver in multi-phase CT scan.

On the liver region of the CT scan, we propose two frameworks for lesion detection and segmentation. The first is A multi-stage and multi-target framework for liver lesion segmentation in multi-phase CT scans based on convolutional neural networks. The framework uses three stages. The first stage identifies the regions within the liver where there might be lesions at three different scales (4, 8, and 16 mm). The second stage is the main segmentation stage and uses four models. The first model is trained on the combined CT volume of all the phases, which are three phases (the arterial, delayed, and venous phases). The remaining three models are trained on each of the phases individually. The third stage is a segmentation refinement and correction stage that incorporates the predictions and CT volume as inputs to refine the predicted segmentation mask. The overall structure of this framework is outlined in Fig. 1.4. The framework enhances feature extraction from multi-phase CT scans and improves the subject-wise segmentation performance by 1.6% while reducing performance variability across subjects by 8% and the instances of segmentation failure by 50%.



**Figure 1.4:** The overall structure of the multi-stage liver lesion segmentation framework from multi-phase CT scans.

The second framework aims to improve detection rates and reduce the number of missed lesions, especially small ones, as they are significantly more challenging to detect. It uses a lesion selection approach that compares the predictions from two specialized models lesion-wise. The models are trained to specialize in detecting lesions based on their size, with the first model focusing on segmenting lesions regardless of their size, while the second specializes in small lesions. The framework evaluates the lesion masks generated from both segmentation models. It then creates corresponding lesions from these masks based on their overlap. For each corresponding lesion pair, we compare the separation between the lesion and surrounding tissue for each lesion individually using intensity-based features and intensity distribution divergence metrics. The lesion mask that maximizes the separation between the lesion and its surrounding tissue is selected out of the two. The framework was tested on both single- and multi-phase CT scans to improve detection rates and reduce the number of missed lesions. It improves the detection rates for small lesions by 15.5% and by 4.3% for lesions overall.

Our approach also focuses on complimenting the segmentation and detection frameworks with data-centric training and inference pipelines by incorporating extensive data augmentation, pre-processing, and sampling. Without careful planning for data pre-processing, sampling, and augmentation, our models' detection and segmentation performance would deteriorate. For example, without proper sampling that accounts for the significant imbalance in the data, stage 1 of the proposed multi-stage approach would not be able to localize lesions.

### **1.3 Organization of the Thesis**

The remainder of this thesis is organized as follows. Chapter 2 discusses the literature on object detection, segmentation, and tracking for images in general, and for medical imaging, specifically ultrasound and CT scans as they are the focus of our proposed approaches. Chapter 3 discusses our approach to detect, localize and track objects of interest within an ultrasound

scan in real-time to assist sonographers while imaging a patient for diagnostic and intervention purposes. In Chapter 4, we introduce our proposed approach to adaptively balance the recall and precision while training 2D lesion segmentation models, which we extend to 3D models and incorporate into the training of the models that are part of our second liver lesion segmentation framework described in Chapter 7. In Chapter 5, we present multi-phase CT scan registration for the purpose of aligning the liver to detect and segment liver lesions using scans from different phases. Finally, Chapter 6 discusses the multi-stage and multi-target liver lesion segmentation framework for multi-phase CT scans.

# Chapter 2

## Object Detection, Segmentation and Tracking in Medical Images

Object detectors are designed to localize objects and identify their underlying category or class within an image [51, 16]. Segmentation, on the other hand, is tasked with providing a detailed pixel-by-pixel representation of either a class of objects within an image (semantic segmentation), or for each instance of an object (instance segmentation). Object tracking across images, such as frames in a video, is tasked with localizing a target object of interest across these images and identifying the direction of its movement.

### 2.1 Traditional Object Detection, Segmentation, and Tracking Approaches

Before the era of deep learning (DL), many traditional object detection algorithms used handcrafted features to detect objects that usually did not generalize well to real-life situations beyond the specific application they were designed for [52, 53]. However, some prominent traditional methods, such as the 1) Viola-Jones Detector, which uses integral images and a cascade

of classifiers that utilize Adaboost to detect face features [54, 51], 2) Histogram of Oriented Gradients (HOG), which uses the histogram of gradients directions at different image locations [55], and 3) Deformable Part-based Model (DPM) [56] were quite successful. Nevertheless, since the introduction of DL-based object detection algorithms, they outperformed traditional methods significantly on all performance metrics [43, 16, 57].

Traditional segmentation approaches such as thresholding an image based on different intensity levels, or edge detection to separate regions based on the change in intensity levels indicating edges between objects are susceptible to many of the issues that are found in real-life scenarios such as noise, brightness and contrast variations. Other segmentation methods such active contours [58] and watershed segmentation [59] are more resilient to noise and intensity distribution variations, but lack the ability to autonomously operate on an image, and need fine refinement and modification at inference.

Several traditional methods prior to deep learning were proposed to track objects across frames as it is an important task for many applications such as augmented reality and sports analytics [60, 61], and once the object is identified by a user at a certain frame and does not move abruptly across frames, this task is usually easier for an algorithm to perform than object detection and segmentation. These approaches used template matching [61], distribution shift of intensity values [62], and optic flow representation of frame elements movement when compared to a previous frame [63]. Similar to object detection and segmentation, traditional object tracking approaches are currently outperformed by deep learning ones in almost every public evaluation metric.



## 2.2 Object Detection, Segmentation, and Tracking Using Deep Learning

Deep learning-based object detection methods, and specifically CNN-based methods, are currently the state-of-the-art methods [64, 65], especially for small objects. These detectors can be categorized into two broad categories, the two-stage detectors, such as Faster R-CNN [66], and the one-stage detectors, such as SSD [67] and YOLO [17]. Two-stage detectors defined the early success of DL-based methods, and are designed to have high identification and localization accuracy, while one stage detectors are designed to be fast and operate in real-time at 30+ frames per seconds (fps). Recently, these real-time methods achieved state-of-the-art object detection accuracy with a performance that is as good as, or better than, two-stage methods [64]. However, both one- and two-stage detectors require large training sets. If one and two-stage detectors are trained on smaller data sets, which is often the case in medical imaging, overfitting and poor generalization can occur [68].

Deep learning-based segmentation methods that use CNN are based on the encoder-decoder architecture. An encoder is designed to extract features from an image using a series of convolutional layers, which are then followed by a decoder to contextualize these features spatially to generate a segmentation map that assigns each pixel the class the model believes it belongs to [18, 69]. One specific approach, the UNet, that uses a contracting path followed by an expansive path, with skip connections between the two paths to preserve localization information has proven to be highly trainable with a small training set, making it very suitable for usage with medical images [18]. Several improvements have been made to the design of UNet, such as utilizing increased skip connections and deep supervision, which have improved the accuracy performance of the network at the expense of inference time [70] as well as the use of attention blocks in the skip connections [71]. Object tracking using deep learning mainly uses similarity matching models [72]. These models use networks where a cutout of the object of interest is used

together with the image containing it as inputs to the network. Localized similarity measurements between the cutout of the subject and the image are then used to identify the location of the object within the image. In addition to using similarity measurements, these models are recently being trained with both a similarity and a dissimilarity metric to not only train the network on matching cutouts, but also on cutouts that are dissimilar, improving the model's ability to maximize the similarity metrics and feature embeddings for similar objects while minimizing them for dissimilar ones [73].

## **2.3 Object Detection and Tracking in Ultrasound Scans**

In ultrasound imaging, object detection problems have been historically tackled using region of interest (ROI) tracking or segmentation-based approaches. For ROI tracking, several methods have been developed such as block matching [74] where exhaustive search-based block matching (ES-BM) is used to track anatomical structures such as arteries across sequential frames [75], elliptical shape fitting to track and localize arteries and veins [76], and deep learning methods using networks that compare similarities between frames [77]. Even though these ROI tracking methods have shown great potential in tracking objects in ultrasound scans, their ability to assist sonographers in detecting and localizing target anatomical structures during scanning sessions is hindered by their slow inference speeds [77, 78], or their dependency on operators to identify the target ROI at the beginning of a scanning session [74], or both.

In ultrasonography, sonographers regularly use traditional computer-aided detection (CAD) methods to assist with image interpretation and detection of structures. These are used to decrease the effect of ultrasound imaging dependency on operators' effectiveness, producing more unified detection and diagnostic capabilities [79]. Traditional CAD platforms that do not utilize deep learning approaches typically use handcrafted features such as histogram slicing and indexing combined with filtering to flag areas that are different from surroundings or have

significant unique representation when compared to surroundings [80, 81]. However, they tend to fail in their ability to detect and identify structures that are not easily identifiable and distinct from their surroundings. In addition, CAD platforms frequently use active contour models for image segmentation in ultrasound scans [82]. These contours are designed to identify salient features and boundaries of an object to separate it from its surroundings. Nevertheless, they also suffer, from their inability to autonomously identify the object of interest and localize it as they need to be initiated by a user in a location within the image that closely surrounds the object of interest or within it.

Autonomous object detection in medical imaging including ultrasound has been historically treated as a segmentation problem. Segmentation-based approaches are used to recover a pixel-wise representation of every part within an image that belongs to an object [18]. Often, the goal of such algorithms is to identify the presence of an object in a medical scan, localize it, and estimate its size. These three goals are achievable through object detection algorithms [57]. Using DL models designed for object detection in natural images, Cao et al. [83] performed a comparative study of their performance on detecting breast tumors in ultrasound scans. They trained and compared the performance of Fast R-CNN [84], Faster R-CNN [66], You Only Look Once (YOLO) [17], and Single-Shot MultiBox Detector (SSD) [67] on a dataset from 1043 cases (579 benign and 464 malignant). Yap et al. [85] used various DL models including the encoder-decode model UNet [18] to segment and identify lesions in ultrasound scans of the breast on two datasets with a total of 469 cases (356 benign and 133 malignant). Further models were proposed for ultrasound image segmentation based on the encoder-decoder [86, 87], or the Fully Convolutional Network (FCN) architecture [88, 89]. Other successful approaches, such as Sono-Net [90], used saliency maps of image classification networks to identify views, and detect anatomical structures of interest in fetal ultrasound scans.

Even though these DL-based approaches were accurate and successful in detecting, and localizing anatomical structures, they faced several challenges that hindered their ability

to be deployed efficiently. Object detection models such as Faster R-CNN and YOLO need large training sets [66, 17] than is usually, and practically, available for many medical imaging applications. Networks using classification and saliency maps also require extensively large datasets [90] for classification networks to develop the saliency maps needed to detect and localize target anatomical structures. Even though segmentation-based networks do not need large datasets to accurately segment anatomical structures of interest [18], they need pixel-wise annotations and labels, which are orders of magnitude more expensive to develop than bounding box and classification labels [91].

## 2.4 CT Scans Registration

Medical image registration techniques can be categorized as rigid, affine, or deformable depending on the spatial transformation degrees of freedom and the scope at which the transformation varies spatially. Rigid and affine registration transformations are parameterized using a set of parameters that defines the rotation and translation operations for rigid registration in addition to scaling and shearing for affine registration. Deformable registration, on the other hand, is defined by a non-parametric dense correspondence, or the shifts in all directions for each pixel in a 2D image, or voxel in 3D volumes, which is usually called the deformation or registration field. This allows the alignment of different anatomical structures using a more realistic local representation of their movement beyond the constrained global rigid and affine transformations [92].

Traditional deformable registration methods such as HAMMER [93], demons [94], ELASTIX [95], and ANTs [96] used iterative optimization algorithms, which are highly accurate but extremely slow, taking hours to align medical image volumes [97]. To improve registration speed while maintaining accuracy, deep-learning-based registration methods use a learning-based approach to train a model on predicting the transformation between two images;

eliminating the need for the time-consuming iterative optimization at inference. These models are trained using different approaches, including unsupervised [98, 99], semi-supervised [100], and fully supervised methods [101, 102]. Unsupervised methods use general similarity metrics such as the mean square error (MSE) [99] or the normalized cross-correlation (NCC) [98], while semi-supervised methods use a segmentation mask of an anatomical structure of interest as guidance through metrics such as the Dice score [100]. Fully supervised models use the error between predicted and synthetic ground truth deformation as a training metric [101, 102].

Early deep-learning methods, such as those proposed by Cao et al. and Yang et al., used key patch-wise alignment to improve computational efficiency [103, 104]. Other methods used both whole image volumes and patches to predict the deformation that aligns images [102]. Patch-wise registration methods require pre-alignment of images using a rigid or affine transformation as the range of predicted deformation is bound by the patch size, which is typically around 15 voxels [103]. VoxelMorph [105] is among the most prominent deep-learning methods for medical image registration that uses an encoder-decoder convolutional neural network (CNN) architecture inspired by the UNet model [18]. Despite performing well, VoxelMorph can produce large distortions [105], which may hinder the use of registered images for diagnostic and therapeutic procedures as these images would not represent the actual shape and location of anatomical structures. Other deep-learning methods such as SynthMorph [106], which is trained to learn contrast-invariant registration, have demonstrated promising registration performance in aligning anatomical structures of interest with high accuracy. To reduce deformation errors and distortions, Mok et al. proposed cLapIRN [107], which learns the conditional features that are correlated with the regularization hyperparameters to select the optimal smoothness regularization during inference. Although accurate in their ability to align objects within medical imaging, all of these models still introduce deformation errors with varying levels.

## 2.5 Liver and Liver Lesion Detection and Segmentation

Detecting and segmenting lesions within the liver in CT scans is a challenging problem that is still largely unsolved where current approaches have average recall and precision that hovers around 50% for lesion detection and segmentation [45]. Detecting and segmenting the liver itself within CT scans is, however, a relatively easy problem with most approaches proposed to solve this problem achieving segmentation Dice scores that are higher than 90%. Most of the proposed methods that aimed to solve this problem approached it from a segmentation perspective [46, 47, 48, 49]. Methods that approach the problem with 3D segmentation models achieved, on average, better segmentation and detection performance than 2D per-slice or 2.5D multi-slice models. This allows contextual information from different slices within the CT volume to be used for the prediction of the segmentation decision [45, 50, 48]. Yuan et al. [108] used fully convolutional neural networks (FCN) to segment both the liver and lesions within the liver achieving a reasonable Dice score segmentation performance of 65.7% for lesions and 94.3% for the liver while Han [46] used a 3D volumetric UNet achieving a lesion Dice score of 67%. Using both 2D and 3D UNet for feature extraction followed by volumetric aggregation of features, Li et al. [109] achieved a lesion segmentation Dice score of 72.2%. Using multiple connections among layers in different convolutional blocks of UNet, Tran et al. proposed work [110] reached a segmentation Dice score of 73.69%. Using a spatially contrasting encoder branch (similar to UNet) and a parallel spatially expansive branch, Valanarasu et al. proposed KiUNet with a Dice score of 77.5%. These approaches mostly report the global Dice score, which aggregates the true positives (TP), false positives (FP), and false negatives (FN) over all subjects and then computes the Dice score as follows:

$$D_c = \frac{2TP}{2TP + FP + FN}. \quad (2.1)$$

The global Dice score is higher than the Dice score computed by subject when there are variations

in lesion sizes across subjects as the score will be heavily influenced by the count of TP, FP, and FN from large lesions, which models consistently perform better on.

Current state-of-the-art liver lesion segmentation networks use the encoder-decoder with skip connections design paradigm based on the UNet [111] architecture. Several developments have been proposed to the UNet architecture to improve its segmentation and efficiency performance. These developments targeted different components of the network. Notably, these improvements included modifications to the convolutional block, such as the integration of residual or bottleneck blocks, which aid in mitigating the vanishing gradient problem and improving feature representation without substantially increasing computational complexity [112, 113]. Additionally, advancements have been made in the architecture's skip connections, with the introduction of nested and multi-stage connections that facilitate the model's ability to capture and integrate multi-scale contextual information more effectively [114]. Moreover, the incorporation of attention mechanisms within the skip connections further refines the model's focus on relevant features, enhancing segmentation precision by selectively emphasizing important spatial features while suppressing less relevant information [71].

Among the improvements that focused on redesigning the skip connections of the UNet architecture, UNet++ [114], Attention UNet [71], and MultiResUNet made large strides in improving the overall segmentation capabilities of the UNet architecture across different medical image segmentation tasks. UNet++ uses a nested architecture that refines skip connections using multiple interconnected convolutional blocks at different depths, enabling an increased mixture of features across the network. Attention UNet incorporates attention gates within its skip connections, focusing the model spatially on relevant image regions by selectively emphasizing important features while suppressing less relevant information. Other models such as the ResUNet, UNetR [115], SwinUNetR [116], and MedNext [117] improved on the UNet architecture by modifying the design of the convolutional block or replacing it with a transformer-based block. The ResUNet architecture replaces the convolutional blocks within the encoder and decoder with

residual convolutional blocks while the MedNext model uses a residual bottleneck convolutional block, which is a 3D version of the ConvNext block [118] that is used in the current state-of-the-art model for natural image classification. The UNetR and SwinUNetR models replaced the encoder with a transformer-based encoder using the ViT-B model and the Swin Transformer, respectively. The MultiResUNet, on the other hand, included modifications to both the convolutional block and the skip connections by incorporating multi-residual paths inspired by DenseNet in the first and successive residual blocks in the second. Transformer encoders, in general, however, struggle with smaller objects and extracting localized dense representations as they encode features using a patch-based manner. The Swin Transformer aimed to mitigate this problem with shifted windows, but it still lags behind in its ability to recover small objects, such as lesions, in their early stages.

Deep learning models based on the UNet architecture are the most widely used and best performing for the task of liver lesion segmentation in CT scans [119]. These models include the current state-of-the-art model, nnUNet [48]. The nnUNet model uses the original UNet architecture and incorporates a self-configuration approach that modifies the network's depth, width, and under-sampling stages, among others, based on the dataset footprint in terms of the target anatomical structure intensity and spatial characteristics. The Model Genesis [120] UNet model, which uses self-supervised learning as a pre-training approach to learn transferable image representations, also performs comparably to the nnUNet model. Transformer models, in general, do not perform on par with CNN models due to the relatively small size of lesions to the overall scan. However, the SwinUNetR model can achieve comparable results due to the use of shifted windows, which improves local and small feature extraction for dense segmentation predictions. All of these methods, however, miss a large number of lesions within the liver, suffering from a large false negative rate. For small lesions, this is more prevalent as they are harder to detect due to the limited distinctive features (because of size) when compared to their surroundings.



# Chapter 3

## Real-Time Object Detection and Tracking in Ultrasound Scans

### 3.1 Problem Definition and Motivation

Ultrasound scanning is an important step in many medical diagnostic and therapeutic workflows due to its well established safety record, its ability to visualize differences among soft tissues, and portability [121, 122]. However, ultrasound scanning is labor intensive where a scanning session can take up to 30 minutes. Ultrasound scans are also sonographer dependent, creating relatively high cross operator variability in accurate anatomical structure identification; novice sonographers demonstrate high diagnostic error rates at up to 52% more than expert sonographers [123]. Ultrasound imaging is primarily used to image soft tissue, which is inherently compressible, creating additional within-subject image variability. Despite these limitations, the trained clinician must carry out accurate and precise ultrasound scanning as it is critical for identification of targeted structures, as well as for precise and accurate therapy administration [121]. An automatic framework tool that assists sonographers in detecting and localizing anatomical structures may radically improve reliable across-subjects scanning for both novice and expert

sonographers.

In medical imaging, object detection problems have been historically tackled using region of interest (ROI) tracking or segmentation-based approaches. For ROI tracking, several methods have been developed such as block matching [74, 75, 76], and deep learning methods using networks that compare similarities between frames [77]. Although capable of tracking objects, ROI tracking methods' ability to assist sonographers in detecting and localizing target anatomical structures during scanning sessions is hindered by their slow inference speeds [77, 78], or their dependency on operators to identify the target ROI at the beginning of a scanning session [74], or both. Segmentation-based approaches can be used instead to autonomously detect, localize, and track anatomical structures of interest. However, they require pixel-wise annotations that are expensive to generate as they require a large number of expert working hours. Deep learning object detection approaches, on the other hand, require extensively large training sets to prevent overfitting and generalize well [66, 17], which is usually not possible in medical imaging and clinical studies. They used bounding box labels and do not require pixel-wise annotations and labels, which are orders of magnitude more expensive to develop [91].

### **3.1.1 Proposed Approaches and Contribution**

Motivated by these constraints and needs, we proposed two different frameworks in progression to solve the problem of object detection and tracking in ultrasound scans. Our first proposed framework to detect, localize and track a specific target anatomical structure in real-time, uses a weakly supervised and modified UNet convolutional neural network (CNN) as its backbone detection and localization algorithm [18]. It is designed to autonomously assist sonographers in real-time to enhance their ability to detect and track objects of interest during scanning sessions. We show that the proposed method outperforms YOLOv4 [124] and EfficientDet [64], which are two state-of-the-art real-time object detection methods, in detecting and tracking a test anatomical structure, which is the Vagus nerve using a dataset that was internally developed [34]. We

use bounding box information to train the backbone network and achieve high detection and localization accuracy even when the model was trained on a small training set. The second proposed framework incorporates a CNN optical flow model in addition to the weakly supervised segmentation model to estimate movements of elements within a scan across frames. The framework can detect and localize a target anatomical structure within the ultrasound scan field of view with an average localization error of less than 1.25 mm, and a tracking error of less than 0.75 mm. Both frameworks have an inference latency of less than 33 ms when operating on a medium-range graphical processing unit (GPU) such as the Nvidia RTX 2080Ti.

To further improve object detection and localization accuracy in ultrasound scans, we designed an improved UNet network that utilizes a multi-path decoder (MD UNet) for deep supervision and early attention (at deeper layers) to the anatomical structure of interest. We designed the model with multi-level-supervision and multi-level spatial attention from the training labels. We weakly trained the network on detecting and localizing anatomical structures of interest using bounding box labels. We tested the performance of the proposed network on two public ultrasound datasets, the breast ultrasound tumor dataset [125], and fetal head dataset [126]. We also tested its performance on our own Vagus nerve dataset [34]. The proposed network outperforms UNet in detecting and localizing anatomical structures of interest in different ultrasound datasets. Its performance is on par with, or slightly better than, the more computationally expensive UNet++. The design of the network improves the localization performance by up to 7% while only increasing the number of parameters of the network by 0.75%. On the other hand, UNet++ [70] requires 20% more parameters and twice the inference time than UNet, and the proposed network. Furthermore, the proposed architecture detects and localizes key anatomical structures with greater consistency and accuracy across different subjects and scans when compared to both UNet and UNet++ as well as Faster R-CNN and YOLOv5.

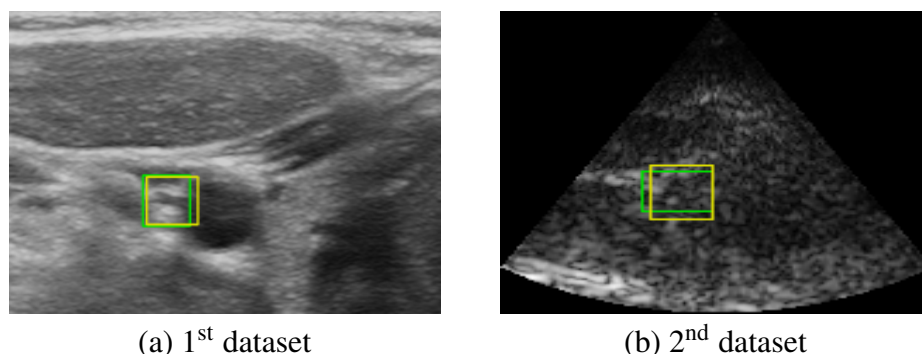
## 3.2 Methods

In this section, we introduce the datasets used to test the performance of the two proposed frameworks and the designed multi-path decoder UNet model. We then outline the set of data augmentation used to promote generalization and prevent overfitting of our models. Finally, we will discuss the evaluation metrics used to judge the performance of the proposed frameworks and models.

### 3.2.1 Datasets and Data Preparation

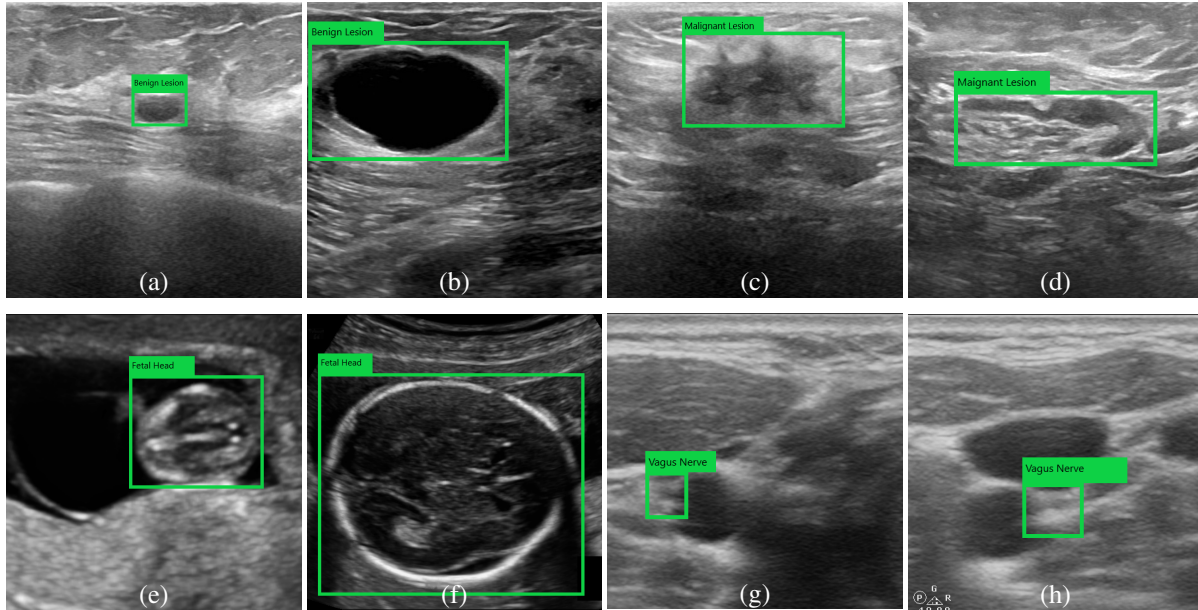
We evaluated the performance of the first framework on two different ultrasound datasets that were created by researchers at UC San Diego (UCSD) Health and Jacobs School of Engineering. The experimental procedures involving human subjects described in this thesis were approved by the Institutional Review Board (IRB) at UCSD (IRB No. 171154). The datasets' scans targeted imaging the Vagus nerve in the mid- and upper-cervical regions of the neck. The scans span different fields of view of the neck to create a variety of scans that would be generated by a sonographer who is looking to image the Vagus nerve within the neck. The two datasets were created using different probes and image reconstruction devices. The 1<sup>st</sup> dataset was created using a probe and device with high-quality diagnostic capabilities. The 2<sup>nd</sup> dataset used a probe that has a small footprint to work alongside non-invasive therapeutic and stimulation devices, and is designed to generate scans at a very rapid pace at the expense of quality. The 1<sup>st</sup> dataset contained 6,368 scans from 3 different subjects, while the 2<sup>nd</sup> contained 26,313 scans from 5 different subjects. Both datasets contained scans from both the left and right sides of the neck. The Vagus nerve shape, location, and surrounding anatomical structures varies greatly within subjects and across subjects. Even a slight movement of the probe can make it challenging for sonographers to re-identify the nerve and its location due to the high variability of neck anatomy visualized with medio-lateral or cephalo-caudal scanning along the cervical neck [127]. In aggregate, nerve

detection with variable anatomy datasets, provides a substantial challenge to test the proposed framework and verify its robustness. Fig. 3.1 shows an example scan from each dataset with the ground truth and predicted bounding boxes. The second framework was evaluated on the 2<sup>nd</sup> dataset.



**Figure 3.1:** Ultrasound scans of the Vagus nerve with the ground truth (green) and predicted (yellow) bounding boxes using the first proposed framework.

The multi-path decoder UNet model was evaluated on 3 different ultrasound datasets. Two public datasets and the 1<sup>st</sup> Vagus nerve dataset. The first dataset is the dataset of breast ultrasound images [125]. This dataset contains a total of 780 scans. Out of those, 210 contain malignant lesions, 437 contain benign lesions, and 133 contain no lesions. The scans were acquired using breast ultrasound scans of women ranging in age from 25 to 75 years. The average scan size in the dataset is  $500 \times 500$  pixels. The second dataset is the fetal head circumference dataset [126]. This dataset contains 999 ultrasound scans of fetal heads at all pregnancy trimesters. The size of the scans in this dataset is the same and is  $800 \times 540$  pixels. The third is the cervical Vagus nerve dataset. From this dataset, we use 1,000 scans from 3 different subjects of size  $680 \times 448$  pixels. Example scans from each of the datasets with the bounding box enclosing the anatomical structure of interest is shown in Fig. 3.2. These datasets have objects of interest that are semantically different, and are of different sizes and shapes. This constitutes a reasonable challenge to test the performance of the proposed model in detecting and localizing objects of interest within the field of view in ultrasound scans.



**Figure 3.2:** Example scans from the three datasets with the bounding box enclosing the object of interest. (a) and (b) showing benign lesions while (c) and (d) showing malignant lesions from the breast ultrasound dataset. (e) and (f) are two examples with different head sizes at different trimesters from the fetal head dataset. (g) and (h) are from the Vagus nerve dataset.

Throughout training and testing of the proposed frameworks, the scans were resized to  $256 \times 256$  pixels for the 1<sup>st</sup> Vagus dataset, and to  $192 \times 192$  for the 2<sup>nd</sup> dataset before being supplied to the models. For the breast ultrasound dataset and the fetal head dataset, the scans were resized to  $256 \times 256$  pixels. In addition to resizing the scans, their intensity values were normalized, and throughout training extensive data augmentation was performed, which is discussed in the next section.

### 3.2.2 Data Augmentation

During training, we used extensive data augmentation to improve our frameworks' ability to overcome overfitting, and generalize to data outside of the training set [128, 129]. The augmentation pipeline includes geometrical transformations and color space randomized contrast and brightness transformations to account for differences in ultrasound signals' energy levels. Most importantly, the pipeline deployed: 1) deformable elastic transformations [130] with ran-

dom Gaussian kernels to elastically deform the grid of an image, which simulates the elastic differences among soft tissues within and across subjects, and 2) mixtures of input scans to enhance the coverage of the probability space while minimizing the risk function during training by implementing vicinal risk minimization instead of empirical risk minimization [131]. While computationally efficient, the empirical risk defined as  $R_e(f) = 1/n \sum_{i=1}^n \ell(f(x_i), y_i)$  only considers the performance of  $f(x)$  on a finite set of training examples for a dataset consisting of training data  $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$  with  $n$  examples of input ( $x_i$ ) and target ( $y_i$ ) pairs, prediction algorithm  $f(x)$ , and a loss function  $\ell(f(x_i), y_i)$ . The empirical risk is used to approximate the expected risk, which is the average of the loss function  $\ell$  over the joint distribution of inputs and targets  $P(X, Y)$ , where the joint distribution is only known at the training examples and can be approximated by the empirical distribution  $P_\delta(x, y) = 1/n \sum_{i=1}^n \delta(x = x_i, y = y_i)$ . However, the distribution can be approximated by  $P_v(\tilde{x}, \tilde{y}) = 1/n \sum_{i=1}^n v(\tilde{x}, \tilde{y} | x_i, y_i)$  where  $v$  is a vicinity distribution that computes the probability of finding the virtual input-target pair  $(\tilde{x}, \tilde{y})$  in the vicinity of the training input-target pair  $(x_i, y_i)$  [131]. The virtual input-target pair  $(\tilde{x}, \tilde{y})$  can be defined as  $\tilde{x} = \lambda x_i + (1 - \lambda)x_j$  and  $\tilde{y} = \lambda y_i + (1 - \lambda)y_j$ , where  $(x_i, y_i)$  and  $(x_j, y_j)$  are two randomly selected input-target pairs from the training set, and  $\lambda$  is sampled from a beta distribution ( $\lambda \sim \text{Beta}(\alpha, \alpha)$ ,  $\alpha = 0.1$ ). This approximation offers a more comprehensive representation and coverage of the joint distribution  $P(X, Y)$ . The use of mixtures of inputs have been implemented in image classification problems to minimize vicinal risk [132]. In our framework, we have built and implemented an approach to use mixtures of inputs to minimize vicinal risk for segmentation-based algorithms.

Prior to training, the masks that are used to train the network are created from bounding box coordinates as images (tensors) of size  $H \times W$  where pixel values within the bounding box are set to 1. This mask will be used to weakly train the segmentation network to detect and localize the presence of target objects within the boundaries of the box.

### 3.2.3 Evaluation Metrics

To evaluate the accuracy of the proposed frameworks, we used the average precision and recall of localization, where a detection is considered a true positive when a certain localization threshold is met, otherwise, that detection is considered a false positive. The main localization metric is the intersection over union, which is the main metric used to evaluate object detection algorithms [133]. We also used the deviation of the predicted center location of detected and tracked objects from the ground truth location in mm. The intersection over union is used to evaluate the overlap between two bounding boxes and is defined as follows:

$$IoU(A, B) = \frac{|A \cap B|}{|A \cup B|}, \quad (3.1)$$

where  $A$  and  $B$  are the areas of the two bounding boxes,  $A \cap B$  is the area of intersection of  $A$  and  $B$ , the region where both bounding boxes overlap, and  $A \cup B$  is the area of union of  $A$  and  $B$ . To compute the precision, we use the following definition:

$$\text{Precision} = \frac{TP}{TP + FP}, \quad (3.2)$$

and to compute the recall, we use the following definition:

$$\text{Recall} = \frac{TP}{TP + FN}, \quad (3.3)$$

where  $TP$  stands for the number of true positives,  $FP$  for the number of false positives, and  $FN$  for the number of false negatives.



### 3.3 Segmentation-Based Framework

In this section, we will explain in detail the structure of our first proposed framework as well as the experimental setup and results of the tests we used to evaluate its performance.

#### 3.3.1 The Framework and Its Structure

The proposed framework consists of 4 stages, as outlined in Fig. 3.3. The 1<sup>st</sup> stage is designed to pre-process the scans, the 2<sup>nd</sup> stage to detect and localize the target object within a scan, the 3<sup>rd</sup> stage to classify whether a scan contains the target object, and the 4<sup>th</sup> and final stage to fine-tune the detection parameters.

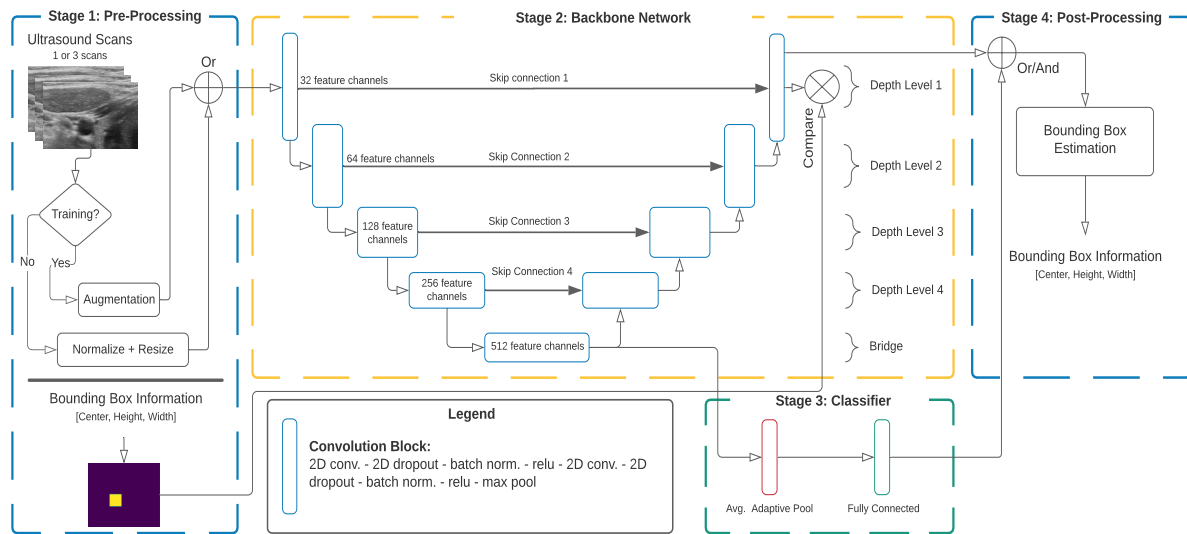
##### Stage 1: Pre-Processing

In this stage, the frames are prepared for the backbone network in stages 2 and 3. The current ultrasound frame is stacked with the previous two frames as a three-channel tensor of size  $H \times W \times 3$ , where  $H$  and  $W$  are the height and width of the scan (frame). Using 3 frames instead of 1 has improved the localization accuracy by 1.7%. The frames' intensity values are then normalized to be in the range  $[0, 1)$ . During training, we used extensive data augmentation to improve our framework's ability to overcome overfitting, and generalize to data outside of the training set [128, 129]. The augmentation pipeline was discussed in detail in section 3.2.2.

In this stage, the masks that are used to train the network are created from bounding box coordinates as images (tensors) of size  $H \times W$  where pixel values within the bounding box are set to 1. This mask will be used to weakly train the network in stage 2 to detect and localize the presence of target objects within the boundaries of the scan.

## Stage 2: Backbone Detection Network

Our framework’s backbone network is designed based on a modified UNet architecture as shown in Fig. 3.3. The proposed network uses 4 depth levels as the standard UNet with 2 convolutional layers in each depth level as well as the bridge of the network. However, in our proposed framework we used 32, 64, 128, 256, and 512 channels in the feature maps at levels 1, 2, 3, 4, and the bridge, respectively. The original method used twice the number of feature maps channels at each of these levels. Reducing the number of channels allows the network to operate in real-time.



**Figure 3.3:** Overview of the proposed object detection framework with its four stages outlined.

Reducing the size of a neural network usually reduces performance. To cope with this, we incorporated several modifications to improve the performance of the network such as two dimensional (2D) dropout layers in addition to original dropout layers. 2D dropout layers regularize the activations more efficiently when high correlation exists among pixels that are close to each other [134]. We also incorporated batch normalization layers and added a localization promoting term to the cost function. The original cost function of UNet is a confidence promoting loss function that computes the binary cross-entropy (BCE) between each pixel of the ground

truth and predicted mask. For each element of the predicted mask with a value  $x$  and ground truth value  $y$  at location  $(i, j)$ , where  $i = 1, 2, \dots, H$  and  $j = 1, 2, \dots, W$ , the BCE cost function can be computed for each training batch as:

$$\mathcal{L}_{bce}(X, Y) = -\frac{1}{N} \sum_{n=1}^N \left[ \frac{1}{M} \sum_{m=1}^M \left[ \ell(x_{m,n}, y_{m,n}) \right] \right], \quad (3.4)$$

where  $N$  is the size of the batch,  $M$  is the number of elements in the mask and is equal to  $H \times W$ ,  $X$  is the predicted mask,  $Y$  is the ground truth mask, and  $\ell(x_{m,n}, y_{m,n})$  is the loss computed element-wise between the ground truth and predictions, and is defined as:

$$\ell(x, y) = w_c y \log \sigma(x) + (1 - y) \log \sigma(x). \quad (3.5)$$

In (3.5), the weight  $w_c$  adjusts the loss function penalization for class  $c$  based on the training set size imbalance for each class, and  $\sigma(x)$  is the sigmoid function defined as  $\sigma(x) = 1/(1 + \exp(-x))$  and it maps the predicted elements into a probability space of predictions where  $\sigma(x)$  constitutes an object if larger than or equal to 0.5, and background otherwise. The weight  $w_c$  and the threshold of  $\sigma(x)$  are used to influence the precision and recall of the network. The loss function promoting localization is based on the Dice coefficient between the predicted and ground truth mask. The Dice coefficient ( $D_c$ ) is defined as [100]:

$$D_c(\hat{Y}, Y) = \frac{2 \sum (\hat{Y} \odot Y)}{\sum_{m=1}^M \hat{y}_m + \sum_{m=1}^M y_m}, \quad (3.6)$$

where  $\odot$  represents the element-wise multiplication and  $\hat{y}_m = \sigma(x_m)$ . The Dice coefficient loss can then be defined to penalize lower  $D_c$  values, which yields lower localization performance, as:

$$\mathcal{L}_{D_c}(X, Y) = 1 - D_c(\hat{Y}, Y). \quad (3.7)$$

The overall object detection loss function is defined as:

$$\mathcal{L}_{obj}(X, Y) = \alpha_{bce} \mathcal{L}_{bce}(X, Y) + \alpha_{dice} \mathcal{L}_{D_c}(X, Y), \quad (3.8)$$

where  $\alpha_{bce}$  and  $\alpha_{dice}$  are coefficients that control the contribution of BCE loss and Dice loss, respectively, to the overall loss function. In our implementation we chose  $\alpha_{bce} = 0.25$  and  $\alpha_{dice} = 1$ .

### Stage 3: Classifier

Stage 2 is designed to localize an object within a scan, but is not optimized to identify the presence of the target object in the scan. Thus, to detect whether the target object is in the scan or not, we use a classifier optimized for this task as can be seen in Fig. 3.3. The classifier adds two extra layers to the framework and uses the output of the last layer of the bridge in stage 2, which contains 512 feature map channels, as input. This input is flattened to a tensor of length 512 using the average global pooling layer that was proposed as part of ResNet [112]. This is then followed by a fully-connected layer and an output layer for the 2 classes activated by a softmax function where the BCE loss is used to train the classifier.

### Stage 4: Post-Processing

The output mask of the backbone network from stage 2 is of size  $H \times W$ . After being threshold by the sigmoid function, the output mask will have elements with values between 0.5 and 1, as well as 0. These elements where the value is higher than 0.5, represent the region in which the network believes the target object is located. The average of the locations of these elements weighted by the confidence, which is the output of the sigmoid function  $\sigma(x, y)$ , is used to estimate the center location of the target object. The center  $(x_c, y_c)$  of the target location can be

then estimated as follows:

$$x_c = \frac{\sum_{k=1}^K \sigma(x_k)x_k}{\sum_{k=1}^K \sigma(x_k)}, \quad y_c = \frac{\sum_{k=1}^K \sigma(y_k)y_k}{\sum_{k=1}^K \sigma(y_k)}. \quad (3.9)$$

$K$  is the number of elements where the confidence  $\sigma(x, y)$  is higher than the threshold, and  $\sigma(x_k) = \sigma(y_k) = \sigma(x_k, y_k)$ . The weighted standard deviation of these elements' locations is used to estimate the width and height of the target object, and can be defined as follows for  $x$ :

$$\sigma_x = \sqrt{\frac{\sum_{k=1}^K \sigma(x_k)(x_k - x_c)^2}{\frac{(K-1)}{K} \sum_{k=1}^K \sigma(x_k)}}, \quad (3.10)$$

where  $\sigma_x$  is the standard deviation in the x direction.  $\sigma_y$  can be calculated using (3.10) by replacing the corresponding variables. The width and height of the bounding box can then be calculated as:  $width = \beta_x \sigma_x$  and  $height = \beta_y \sigma_y$ , where  $\beta_x$  and  $\beta_y$  are factors that are learned during the training of the backbone network. The output of the classifier is then fed together with the output of the backbone network to a decision logic such as an "or" or "and" to decide on the presence of the target object in the scan. Choosing "and" will increase precision at the expense of recall, and vice versa. Controlling this decision logic together with the thresholds ( $\sigma(X)$ ) for the backbone network and classifier can be used in real-time by sonographers as simple methods to control the rate of false positives versus false negatives.

### 3.3.2 Experiments and Results

#### Implementation and Setup

We conducted 3 experiments to test the performance and robustness of our framework. The 1<sup>st</sup> experiment was designed to test the accuracy of the proposed method in detecting and tracking the Vagus nerve on the 1<sup>st</sup> dataset (check section 3.2.1 for details on the datasets used for evaluation). The 2<sup>nd</sup> experiment was designed to test the performance of the proposed method on

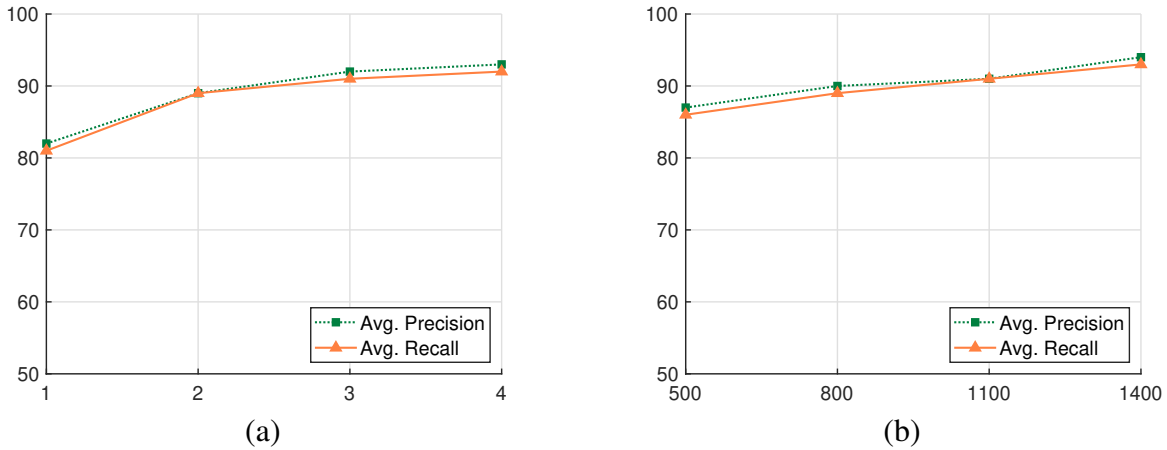
the more challenging 2<sup>nd</sup> dataset and compare it to YOLOv4 and EfficientDet, two state-of-the-art real-time object detectors. In both of these experiments, the datasets were divided into individual scans and split into a 64:16:20 ratio for training, validation, and testing, respectively. The 3<sup>rd</sup> experiment was designed to test and verify the robustness of the framework in accounting for cross-subject variabilities as well as its ability to generalize to new subjects. Hence, in this experiment, we divided our dataset by subject. The framework was trained on scans from 4 subjects and tested on the 5<sup>th</sup> subject. Throughout all three experiments, the scans were resized to 256x256 for the 1<sup>st</sup> dataset, and to 192x192 for the 2<sup>nd</sup> dataset before being supplied to the network in stage 2. The backbone network was optimized using stochastic gradient descent (SGD) with a learning rate =  $10^{-3}$ , momentum = 0.9, weight decay of  $10^{-3}$ , batch size = 16, and trained for 200 epochs.

**Table 3.1:** The proposed framework localization precision and recall. For experiments 1 and 2, the threshold for a true positive is an IoU  $\geq 0.5$ . For experiment 3, a true positive is a distance error  $\leq 2.5$  mm from the nerve center. The model with the best performance is boldfaced.

Method	Avg. Precision	Avg. Recall
Experiment 1 - 1 <sup>st</sup> Dataset		
Ours - Single Frame	94.4%	97.2%
Experiment 2 - 2 <sup>nd</sup> Dataset		
Ours - Single Frame	90.89%	96.01%
Ours - Three Frames	<b>92.67%</b>	<b>97.29%</b>
YOLOv4	90.45%	97.24%
EfficientDet - d3	91.93%	96.35%
Experiment 3 - 2 <sup>nd</sup> Dataset		
Ours - Single Frame	93.5%	91.9%
Ours - Three Frames	<b>95.1%</b>	<b>93.4%</b>

## Evaluation and Results

To evaluate the accuracy and robustness of the proposed framework, we used the average precision and recall of localization, where a detection is considered a true positive when a certain

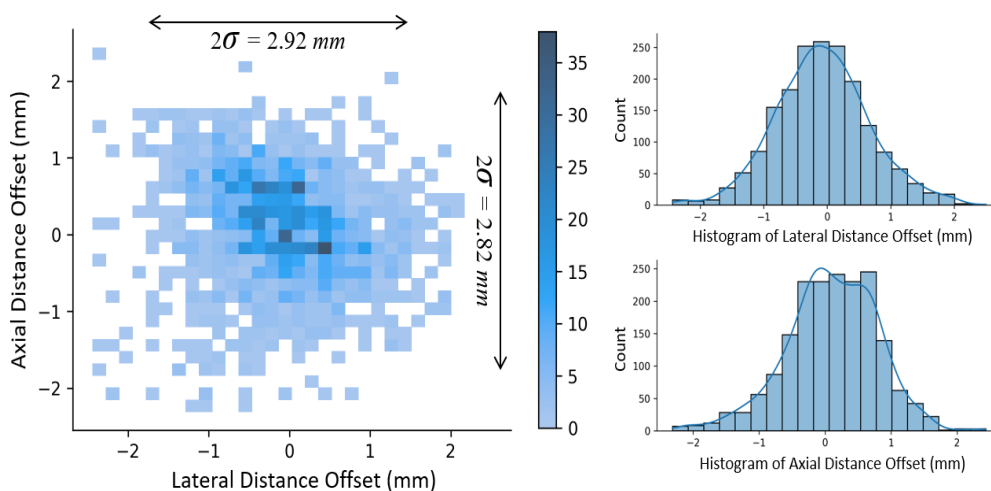


**Figure 3.4:** Object localization precision and recall for the 2<sup>nd</sup> dataset (a) versus the number of subjects used for training (testing on a separate subject), and (b) versus the number of training images (3 subjects for training and 2 for testing. Images chosen randomly from training set).

localization threshold is met, otherwise that detection is considered a false positive. This is the main metric used to evaluate object detection algorithms [133]. For the 1<sup>st</sup> and 2<sup>nd</sup> experiments, the localization threshold is based on the intersection over union (IoU) metric. For the 3<sup>rd</sup> dataset, the threshold is based on a physical distance of 2.5mm from the center of the nerve, which is equal to the radius of a typical Vagus nerve.

Table 3.1 summarizes the performance of the proposed framework for all of the 3 experiments. It can be seen that the proposed framework was able to identify and localize the Vagus nerve in both datasets with a high precision and recall. For the more challenging 2<sup>nd</sup> dataset, the results of the 2<sup>nd</sup> experiment show that the proposed method outperforms YOLOv4 and EfficientDet - d3. For the 3<sup>rd</sup> experiment where the 2<sup>nd</sup> dataset was divided by subjects, 4 subjects for training and 1 subject for testing, the proposed method was still able to generalize to subjects that it did not see during training and achieved high localization precision and recall. This was not the case for YOLOv4 and EfficientDet where the precision and recall dropped below 25%. This loss of accuracy can mainly be attributed to these methods' need for large training datasets with rich visual features to train their backbone and detection networks [124, 64].

To verify the robustness of the proposed framework, we conducted two additional experiments to analyze the proposed framework performance on new subjects while being trained on smaller subsets of the original dataset. The results of these two experiments are shown in Fig. 3.4. In the first experiment, we trained the framework on 1, 2, 3, and 4 subjects then tested on a 5<sup>th</sup> subject and repeated this analysis twice for two different test subjects. We then used three subjects for training and two subjects for testing, randomly sampled scans from the training set, and created training subsets of sizes 500, 800, 110, and 1400. As observed in Fig. 3.4, the proposed framework has a high level of consistency and accuracy even when trained on smaller training sets. The framework produces high localization precision where more than 95% of the true positives predictions are located within 1.5 mm from the ground truth in both the lateral and axial directions, which is shown in the heat map and histograms of the true positive detections offset from the ground truth in Fig. 3.5.

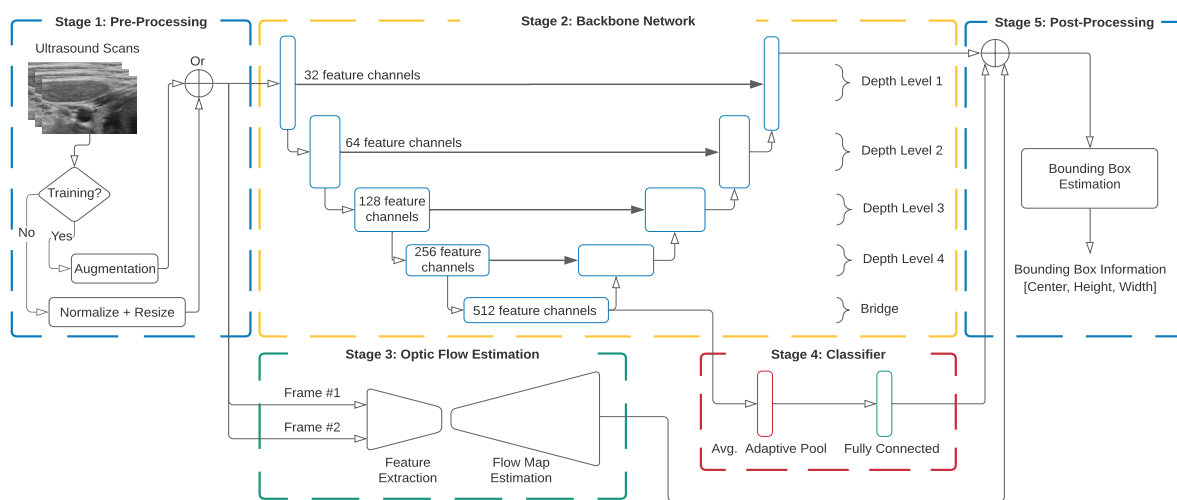


**Figure 3.5:** Heatmap (left) and histograms (right) of the lateral and axial distance offset of the true positive detections from the ground truth in millimeters.



### 3.4 Segmentation and Optical Flow Framework

This framework uses 5 different stages, which are outlined in Fig. 3.6. The pre-processing, backbone network, and classifier stages are adopted from the segmentation-based framework discussed in section 3.3.1. In this section, we will discuss the optical flow stage of the framework and the experiments we used to evaluate its performance together with the results of these experiments.



**Figure 3.6:** Overview of the five stages of the proposed object detection and tracking framework.

#### 3.4.1 The Framework and Its Structure

##### Stage 3: Object Tracking Using Optical Flow Estimation

To track the object detected in the 2<sup>nd</sup> stage as it moves from one frame to another, we incorporated an optical flow estimation network into the design of the framework as the 3<sup>rd</sup> stage. The network is based on the LiteFlowNet architecture [135]. The architecture has 5.4 million parameters. It takes two frames as inputs and generates an optical flow map representing the movement of each element in frame 1 based on its location in frame 2. The flow map has two components, representing the movement in the  $x$  and  $y$  directions, respectively [136]. In

the feature extraction section of the network, there are six depth levels. In each level, there is a series of convolutional layers followed by leaky ReLU. The number of layers in each level is as follows: 1 layer in the first level, 3 in the second, 2 in each of the third and fourth, and 1 in the fifth and sixth layers. The weights of the feature extraction section are tied for both frames, so, the feature maps at each level will be similar for both frames while differences will be mainly attributed to spatial movements. In the flow map estimation section of the network, feature maps of the second frame ( $\mathcal{F}_2$ ) are warped to the ones of the first frame ( $\mathcal{F}_1$ ) at each level. After warping, displacement-based correlation is calculated between the two feature maps:  $\mathcal{F}_1$  and  $\tilde{\mathcal{F}}_2$  (the warped  $\mathcal{F}_2$ ). Warping allows the displacements to span a smaller range since the feature maps have been warped, speeding up inference times. This correlation can be defined as:

$$\begin{aligned} \mathcal{F}_c(x, y, k) &= \mathcal{F}_1(x, y) \cdot \tilde{\mathcal{F}}_2(x + d_x, y + d_y) \\ \forall d_x, d_y &\in [-d, d], \quad d \in \mathbb{Z} \end{aligned} \tag{3.11}$$

In (3.11),  $d$  represents the maximum displacement used to calculate correlation, and  $k$  goes from 1 to  $(2d + 1)^2$ , which is the total number of displacements; assuming we are using a stride of 1. The output flow-map generated from stage 3 is represented by a tensor of size  $H \times W \times 2$ , where at any location  $(x, y)$  in the flow map, 2 numbers represent the movement in the  $x$  and  $y$  directions. The output of this stage is used to track the object as it moves from one frame to another, by averaging the movement that is estimated within the predicted bounding box estimated from the segmentation map output.

## 3.4.2 Experiments and Results

### Implementation and Setup

We conducted 2 experiments to test the performance and robustness of our framework. The 1<sup>st</sup> experiment was designed to test the accuracy of the proposed framework in detecting and

localizing the Vagus nerve. It was conducted on the 1<sup>st</sup> dataset (check section 3.2.1 for details on the datasets used for evaluation). The dataset was divided into individual scans and split into a 64:16:20 ratio for training, validation, and testing, respectively (the scans were resized to 256x256). The backbone network in stage 2 was optimized using stochastic gradient descent (SGD) with a learning rate =  $10^{-3}$ , momentum = 0.9, weight decay of  $10^{-3}$ , batch size = 16, and trained for 200 epochs. The optical flow network in stage 3 was trained stage-wise [135] on the Chairs dataset [136], then on the Things3D dataset [137]. The training used the Adam optimizer and learning rates 1e-4, 5e-5, and 4e-5. The 2<sup>nd</sup> experiment was designed to test the framework’s ability to use the optical flow network in stage 3 effectively to track the movement of the Vagus nerve accurately.

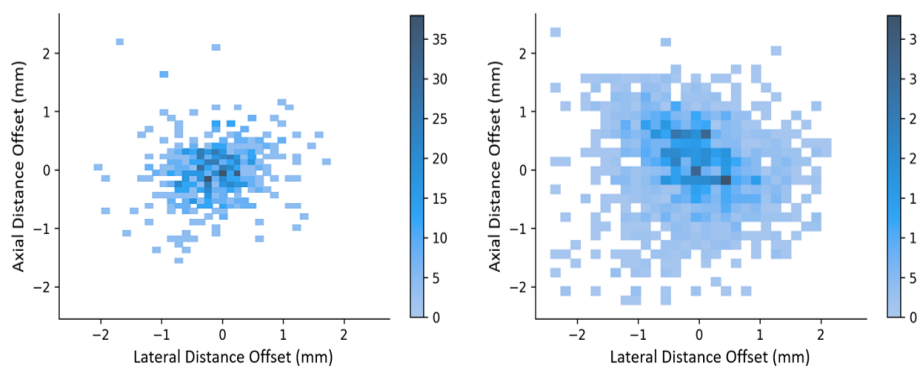
**Table 3.2:** Mean and standard deviation of tracking error in mm for the different ultrasound tracking methods. Autonomous detection of objects and real-time capabilities are highlighted.

Method	Mean	$\sigma$	Detection?	Real-Time?
Shepard [138]	0.72	1.25	No	No
Williamson [139]	0.74	1.03	No	Semi (8 fps)
Gomariz [140]	1.34	2.57	No	Yes (105 fps)
Makhinya [141]	1.44	2.8	No	Yes (20 fps)
Proposed	0.73	1.23	Yes	Yes (30 fps)

## Evaluation and Results

To evaluate the detection and localization accuracy of the proposed framework, we used the average precision and recall of localization. This evaluation metric considers a detection as true positive when a certain localization threshold is met, otherwise that detection is considered a false positive. This is the main metric used to evaluate object detection algorithms [124, 64]. The localization threshold is based on the intersection over union (IoU) metric. The proposed framework achieved an average precision of 94.4% and average recall of 97.2% for an IoU threshold of 0.5. This surpasses the average precision and recall performance of both YOLOv4

and EfficientDet - d3. For YOLOv4, the average precision was 93.2% and the average recall was 97.2%, while for EfficientDet - d3 the average precision was 94.1% and the average recall was 96.8%. For tracking, the framework achieved a mean error of less than 0.75 mm at real-time speeds of 30 fps when tracking the Vagus nerve across frames. Tracking results are summarized in Table 3.2, where the performance of the proposed framework is presented together with several prominent tracking algorithms. These algorithms were designed and trained to track objects in ultrasound and their performance was tested on the Liver Ultrasound Tracking (CLUST) challenge [142]. Finally, the framework achieves high localization precision where more than 95% of the true positive detections are within 1.5 mm from the ground truth in both the lateral and axial directions. Predicted location offset from the ground truth is shown as a heatmap in Fig. 3.7 for both the object detection and tracking predictions. To contextualize the framework localization effectiveness, it is worth noting that the axis span in Fig. 3.7 is smaller than the sides of the bounding boxes enclosing the object of interest.



**Figure 3.7:** The lateral and axial distance offset of the true positive detections from the ground truth in millimeters. Left: Tracking heatmap of offsets. Right: Detection heatmap of offsets.

### 3.5 Multi-Path Decoder UNet (MD UNet)

To increase the accuracy of object detection of the backbone network while maintaining computational efficiency, we designed an improved UNet network that utilizes a multi-path

decoder (MD UNet) for deep supervision and early attention (at deeper layers) to the anatomical structure of interest. We designed the model with multi-level-supervision and multi-level spatial attention from the training labels. We weakly trained the network on detecting and localizing anatomical structures of interest using bounding box labels without the need for expensive-to-generate pixel-wise annotations. We tested the performance of the proposed network on two public ultrasound datasets, the breast ultrasound tumor dataset [125], and fetal head dataset [126]. We also tested its performance on our own internally developed Vagus nerve dataset [34]. The proposed network outperforms UNet in detecting and localizing anatomical structures of interest in different ultrasound datasets. Its performance is on par with, or slightly better than, the more computationally expensive UNet++ (Table 3.4). The design of the network improves the localization performance by up to 7% while only increasing the number of parameters of the network by 0.75%. On the other hand, UNet++ [70] requires 20% more parameters and twice the inference time than UNet, and the proposed network (Table 3.3). Furthermore, the proposed architecture detects and localizes key anatomical structures with greater consistency and accuracy across different subjects and scans when compared to both UNet and UNet++ as well as Faster R-CNN and YOLOv5 (Fig. 3.10).

**Table 3.3:** Computational complexity of the proposed model (MD UNet), UNet and UNet++. Inference time analysis was performed on Nvidia Tesla T4 GPU. In the table, Mul-Add stands for multiplication-addition, M for Millions, G for billions, and ms stands for millisecond.

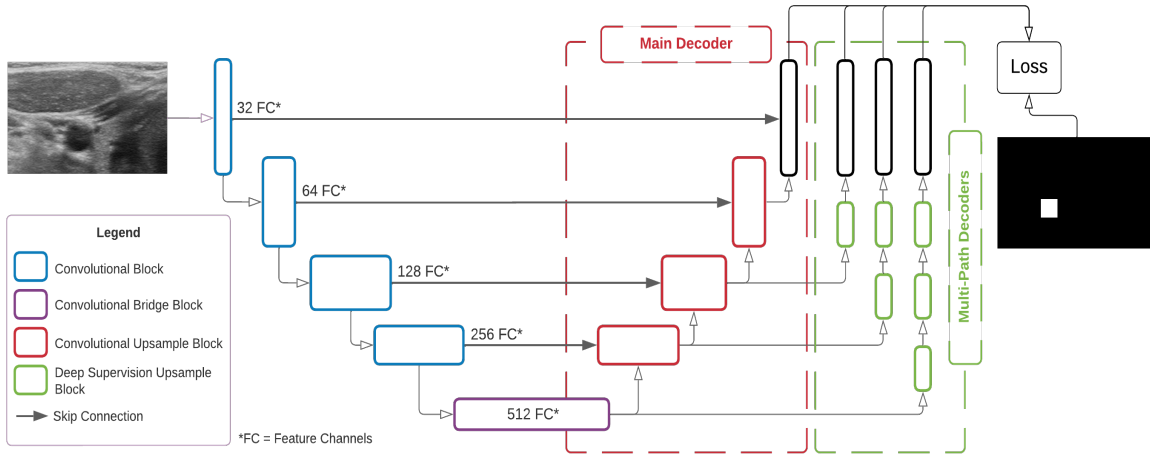
Model	Number of Parameters	Mul-Add Operations	Inference Time
UNet	7.86 M	14.0 G	8.1 ms
UNet++	9.16 M	34.5 G	19.3 ms
MD UNet	7.91 M	14.2 G	8.7 ms

### 3.5.1 The Encoder-Multi-Path-Decoder Architecture

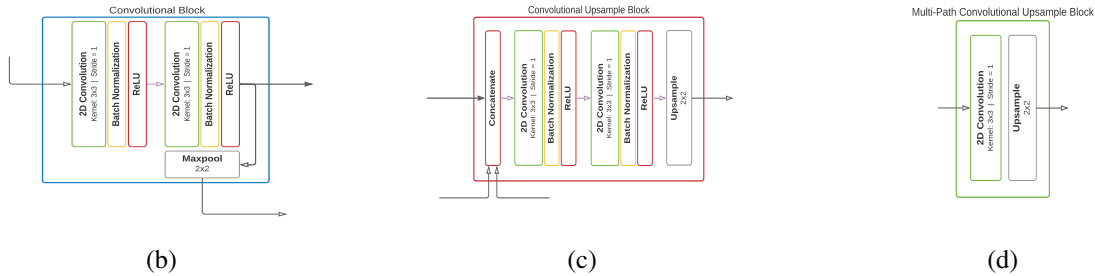
The proposed architecture is composed of an encoder, a main decoder and a multi-path decoder. The encoder path of the proposed model contains four convolutional blocks, with one

block at each depth level. In each of the convolutional blocks, which are shown in Fig. 3.8 (b), there are two convolution layers followed by batch normalization, and ReLU activation. The output of these two layers is then taken as input to two different locations of the model. The first location is the next depth level convolutional block where it first passes through a maxpooling layer to reduce the size of feature map by two in the spatial space. The second location is the convolutional upsampling block that is at the same level through the skip connection. Finally, the bridge block connects the encoder and decoder paths in the network. The design of the bridge is the same as the convolutional blocks. The feature maps have 32, 64, 128, 256, and 512 feature channels at depth levels 1, 2, 3, 4, and the bridge. The number of feature channels is reduced by half when compared to the original UNet encoder.

In the main decoder path of the network there are four convolutional upsampling blocks. These convolutional upsampling blocks start with concatenating feature maps from the skip connection and the previous convolutional upsampling block, or the bridge for the first convolutional upsampling block. The concatenated input is then fed to two consecutive sequences of a convolutional layer followed by batch normalization and ReLU activation. Finally, the output is upsampled in the spatial space by 2 and used as the input to the next convolutional upsampling block. The output of the bridge and the first two convolutional upsampling blocks are used as the inputs to the multi-path decoders. Each of these outputs passes through a sequence of the multi-path convolutional upsampling blocks shown in Fig. 3.8 (d). The number of these blocks in the sequence are 3, 2, and 1 for the outputs from the bridge, first, and second convolutional upsampling blocks, respectively. Our proposed multi-path convolutional upsampling blocks serve two purposes: First, they enable more direct propagation of the loss from feature maps to deeper layers and the encoder, when compared to the main decoder path. Second, they facilitate the spatial upsampling of feature maps to preserve high-resolution information by propagating the loss from the same mask map used in the main decoder path.



(a) Multi-path decoder UNet network



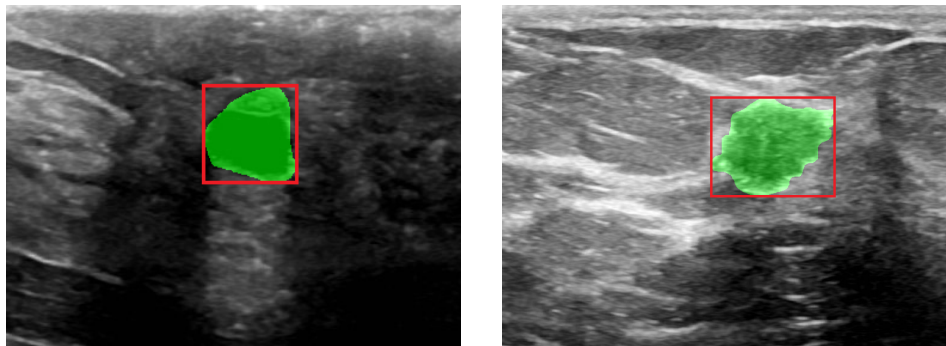
**Figure 3.8:** The proposed model structure (a) with the different building blocks outlined: The convolutional block in the encoder path (b), the convolutional upsampling block in the main decoder (c), and the multi-path convolutional upsampling block (d).

### 3.5.2 Experiments and Results

#### Implementation and Setup

For each of the three datasets described in section 3.2.1, we trained the proposed model from randomly initiated weights. We chose to follow this approach to evaluate the robustness of the proposed architecture and its generalization ability without any pre-training on any prior datasets. For each of the datasets, the scans were split into 80% for training and validation, and 20% for testing. The scans were resized to  $256 \times 256$  pixels. We compared the model performance to two encoder-decoder networks; UNet as a baseline benchmark and to UNet++

as the current state-of-the-art segmentation fully convolutional network. We also compared the proposed model to two benchmark object detection networks; Faster R-CNN and YOLO v5. All models, have been trained and fine-tuned on the three datasets. The breast and fetal head datasets have segmentation maps as ground truth. For these two datasets, we calculate the bounding box based on the span of the segmentation map for each target object. For the Vagus nerve dataset, the dataset is already labeled using bounding box information. Two examples from the breast dataset with mask maps and corresponding bounding boxes are shown in Fig. 3.9.



**Figure 3.9:** Two ultrasound scans from the breast dataset. The mask maps are highlighted in green and the bounding box computed from the mask map in red for a benign (left) and malignant (right) lesion.

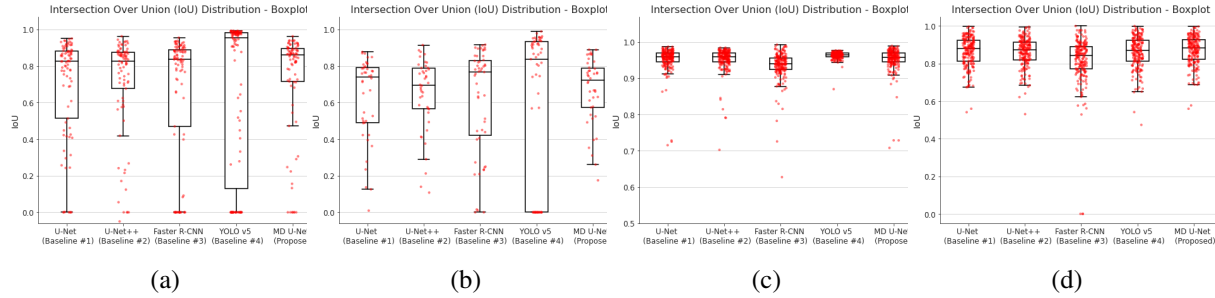
## Evaluation and Results

To evaluate the performance of our proposed architecture, we use the intersection over union (IoU) metric between the predicted bounding box and the ground truth one. The IoU metric, which is also known as the Jaccard index, is the benchmark metric used to evaluate object detection, segmentation, and localization methods [91]. This metric is calculated as follows:

$$IoU(\hat{Y}, Y) = \frac{\sum(\hat{Y} \odot Y)}{\sum_{i,j=1}^{H,W} (\hat{y}_{ij} | y_{ij})}, \quad (3.12)$$

where  $\odot$  is element-wise multiplication,  $|$  is the element-wise *or* logical operator,  $\hat{y}_{ij}$  is the predicted mask map, and  $y_{ij}$  is the ground truth at location  $(i, j)$ . Using the IoU value to evaluate





**Figure 3.10:** Boxplots of the proposed architecture’s detection performance (in terms of IoU) compared to the four baseline models on the (a) breast benign lesion, (b) breast malignant lesion, (c) fetal head, and (d) Vagus nerve datasets. For each model, the IoU of each sample from the test set is also overlaid on top of the boxplot.

whether a detection is considered a proper detection or not, we compute the average precision and recall based on a certain IoU threshold. That means if that detection has an IoU value higher than the threshold, it will be considered a true positive, otherwise, it is a false positive. The average precision is defined as  $TP/(TP + FP)$ , while the average recall as  $TP/(TP + FN)$  where  $TP$  stands for true positives,  $FP$  for false positives, and  $FN$  for false negatives.

The performance of the proposed multi-decoder architecture using these metrics is summarized in Table 3.4, and is compared to the other 4 benchmark models. For the challenging breast lesions dataset, we also added the multi-decoder architecture to the UNet++ model to showcase its added value in improving detection and localization. The proposed architecture constantly outperformed UNet, and was either on par, or outperformed, UNet++ with an inference time that is less than half of what the UNet++ architecture needs, which is essential for real-time guided therapeutic applications. The performance of the proposed architecture was evaluated by conducting a statistical test on its IoU scores using the bootstrapped one-sided Mann-Whitney U test. The test determines whether the distribution of IoU scores of the proposed architecture is significantly greater than the distributions of the other architectures it was compared against. Table 3.4 presents the p-values obtained from these tests. The results show that the multi-decoder architecture significantly improves the object detection performance of UNet in ultrasound scans, with only a minimal increase in computational complexity.

**Table 3.4:** Detection and localization performance results of the proposed model compared to the other four models we used as a benchmark. In the table, (MD) stands for multi-decoder, and (Avg.) stands for average, Std. Dev. stands for standard deviation, and IoU stands for intersection over union.

Dataset Method	Avg. IoU ( $\pm$ Std. Dev.)	Precision @IoU = 0.5 @ max recall	Recall @IoU = 0.5 @ max precision	Precision @IoU = 0.75 @ max recall	Recall @IoU = 0.75 @ max precision
<b>Breast Dataset - Benign Lesion</b>					
Faster R-CNN	65.4% ( $\pm$ 35.6)	94.5%	77.5%	86.3%	75.9%
YOLO v5	67.2% ( $\pm$ 42.2)	93.7%	73.5%	86.3%	71.8%
UNet	66.9% ( $\pm$ 28.3)	82.1%	90.1%	70.5%	87.7%
UNet++	69.7% ( $\pm$ 27.1)	92.5%	90%	71.4%	87.3%
Ours - MD UNet	70.6% ( $\pm$ 27.3)	92.1%	90.1%	78.9%	87%
Ours - MD UNet++	72.6% ( $\pm$ 26.1)	93.51%	90.1%	78.9%	87.7%
<b>Breast Dataset - Malignant Lesion</b>					
Faster R-CNN	61.6% ( $\pm$ 28.7)	72.3%	68%	< 60%	< 60%
YOLO v5	59.3% ( $\pm$ 42.5)	89.7%	67.9%	< 60%	< 60%
UNet	64.8% ( $\pm$ 23.0)	73.7%	77.4%	< 60%	< 60%
UNet++	66.9% ( $\pm$ 20.2)	82.5%	94.3%	< 60%	< 60%
Ours - MD UNet	67.1% ( $\pm$ 18.3)	85.4%	97.2%	< 60%	< 60%
Ours - MD UNet++	70.1% ( $\pm$ 18.7)	87.3%	97.5%	< 60%	< 60%
<b>Fetal Head Dataset</b>					
Faster R-CNN	92.2% ( $\pm$ 10.6)	98.5%	98.5%	97.5%	97.5%
YOLO v5	95.8% ( $\pm$ 6.9)	100%	99.5%	100%	99.5%
UNet	95.2% ( $\pm$ 3.5)	100%	100%	99.5%	100%
UNet++	95.5% ( $\pm$ 3.3)	100%	100%	99.5%	100%
Ours - MD UNet	95.4% ( $\pm$ 3.4)	100%	100%	99.5%	100%
<b>Vagus Nerve Dataset</b>					
Faster R-CNN	81.5% ( $\pm$ 13.4)	100%	98.5%	83.3%	89.2%
YOLO v5	85.9% ( $\pm$ 8.6)	99.5%	100%	91.5%	95.3%
UNet	86.4% ( $\pm$ 8)	100%	100%	92%	95.8%
UNet++	86.9% ( $\pm$ 7.6)	100%	100%	91%	95.8%
Ours - MD UNet	86.9% ( $\pm$ 7.7)	100%	100%	95.8%	100%

To test the performance of the proposed model across variable scans, we analyzed its performance distribution and compared that to the four benchmark models as shown in Fig. 3.10. We observe that the performance of the proposed architecture is more consistent across scans. For the Vagus nerve and fetal head datasets, the IoU score difference among the five models is minimal due to performance saturation from training the networks on the full training set. Nevertheless, performance variability for the proposed architecture is still lower than the other models for the

Vagus nerve dataset as shown in Fig. 3.10 (d). We conducted an analysis to assess the proposed architecture’s robustness to data variability and compared it against four benchmark models. The analysis involved setting two IoU thresholds and computing the percentage of detections below these thresholds for each model. We repeated this process to evaluate each model’s performance on all the datasets, except the fetal head dataset due to performance saturation for this dataset. The first threshold was set to an IoU of 0.5, which is the standard threshold for false positives in most object detection applications. The second threshold was set to the 25<sup>th</sup> percentile of all detections from all models for the corresponding dataset. A lower percentage of detections below these thresholds indicates higher model robustness, as it implies a smaller number of inaccurate detections. The results of this analysis are presented in Table 3.5.

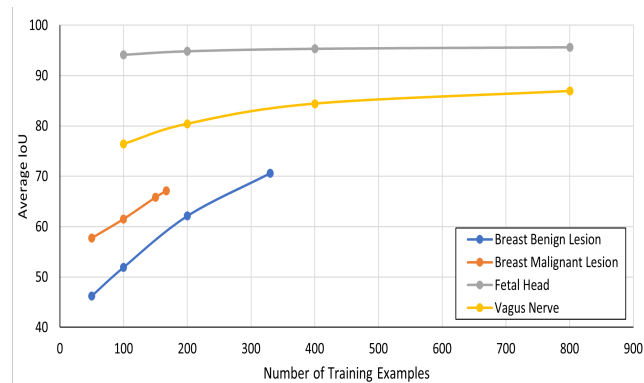
**Table 3.5:** The Percentage of detections below two predetermined thresholds of the proposed model and the four benchmark models for the benign and malignant breast datasets, as well as the Vagus nerve dataset. A lower percentage of detections below the threshold indicates a greater degree of model robustness to data variability. The model with the best performance is boldfaced.

Threshold	Model	Dataset		
		Breast Benign	Breast Malignant	Vagus
IoU = 0.5	Faster R-CNN	25.8%	33.3%	1.5%
	YOLO v5	29.9%	33.3%	0.5%
	UNet	24.7%	31.0%	<b>0.0%</b>
	UNet++	<b>15.3%</b>	21.4%	<b>0.0%</b>
	MD UNet	17.6%	<b>16.7%</b>	<b>0.0%</b>
25 <sup>th</sup> Percentile	Faster R-CNN	25.8%	31.4%	39.8%
	YOLO v5	32.7%	33.3%	23.5%
	UNet	27.1%	23.8%	22.6%
	UNet++	<b>18.8%</b>	19.0%	<b>19.6%</b>
	MD UNet	<b>18.8%</b>	<b>14.3%</b>	<b>19.6%</b>

## Ablation Studies

We performed two ablations studies to test the ability of the network to perform under more challenging conditions, as well as another study to examine the effect of the number of decoder paths on performance. The first ablation study focused on dataset size, while the second

focused on architecture size. For the first study, we reduced the size of the training set multiple times for each of the datasets and tested the performance of the proposed architecture on the same testing set. We observe from Fig. 3.11, that the performance of the network is reliable even when trained on datasets of only 150 to 200 examples. The detection and localization performance, when trained on half the size of the original dataset, stayed within a 10% tolerance from the original performance (when trained on the complete dataset) across all three datasets.



**Figure 3.11:** The proposed model IoU performance versus the number of training examples for each dataset. Each dataset’s last data point is for the model trained on the whole training set (vertical axis does not start from 0).

For the second ablation study, we reduced the number of feature channels by half then by one fourth, and tested the performance of the architectures on detecting the benign and malignant lesions, which are the two most challenging tasks out of all the datasets. Table 3.6 shows the results of this study. As the size of the network is reduced, the performance is also reduced. However, even with one quarter the original number of feature channels, the proposed architecture performance was still within a 4% tolerance of the original architecture performance.

In the third ablation study, we wanted to analyze the effect of the number of decoders on the model performance. We tested different variations of the proposed model with a single (UNet), two, three, and four (MD UNet) decoders. Table 3.7 shows the results of this study. We can see clearly that as we increase the number of decoders, the model’s ability to detect and localize benign and malignant lesions in breast scans improves. This is a further indication that

**Table 3.6:** IoU performance of the proposed architecture and UNet++ on the benign and malignant lesion datasets as the size of the architectures is reduced in terms of feature channels. The model with the best performance is boldfaced.

Size	Model	Dataset	
		Benign	Malignant
Original	UNet++ (Baseline)	69.7%	66.9%
	MD UNet (Proposed)	<b>70.6%</b>	<b>67.1%</b>
Half	UNet++ (Baseline)	67.2%	<b>65.5%</b>
	MD UNet (Proposed)	<b>68.2%</b>	65.4%
Quarter	UNet++ (Baseline)	66.2%	64.7%
	MD UNet (Proposed)	<b>66.7%</b>	<b>64.8%</b>

using multi decoders for object detection and localization produces significantly better results than single decoders with just a minimal increase in computation complexity.

**Table 3.7:** IoU performance of the proposed architecture on the benign and malignant lesion dataset as the number of decoders changes. In the table, Mul-Add stands for multiplication-addition, M stands for Millions, and G for billions.

Number of Decoders	Number of Parameters	Mul-Add Operations	Dataset	
			Benign	Malignant
1 (UNet)	7.86 M	13.97 G	66.9%	64.8%
2	7.91 M	14 G	67.4%	65.3%
3	7.91 M	14.07 G	69.1%	65.9%
4 (MD UNet)	7.91 M	14.18 G	70.6%	67.1%

### 3.6 Limitations and Future Prospective

The multi-decoder architecture we propose in this thesis together with the two multi-stage frameworks are for object detection and localization in ultrasound scans. The proposed approaches are limited in the sense that they have not been tested on other medical imaging modalities, and have not been tested on segmentation tasks, which are closely related to object detection and localization in medical imaging. Therefore, a logical future extension for the proposed work in this chapter is to train and test the approaches we discussed on detecting anatomical structures of

interest from other medical imaging modalities. Another limitation we observed is that precise localization performance of the proposed architecture at an IoU higher than 0.75 degrades for extremely small datasets such as the malignant lesion dataset; this, however, is a limitation of all the models we tested as shown in Table 3.4.

### **3.7 Conclusion**

In this work, we proposed two frameworks for accurate real-time object detection and tracking in ultrasound scans that do not require large training sets, and although they use a segmentation model as their backbone architecture, they do not require pixel-wise annotations for training as they are trained using bounding box labels. We also proposed a weakly trained multi-path decoder segmentation-based architecture for real-time object detection and localization in ultrasound scans. The proposed architecture enables the loss and optimization algorithm to influence deeper layers more prominently through the multiple decoder paths improving the network’s overall detection and localization performance. To evaluate the architecture’s effectiveness, we tested it on three different ultrasound scans, focusing on the detection and localization of different objects of interest. The proposed architecture outperformed UNet while having only 0.75% more parameters. Its performance is on par with UNet++, a state-of-the-art architecture, which contains 20% more parameters than the original UNet and requires more than twice as much time for inference when compared to both the proposed architecture and UNet. Thus, the proposed architecture offers a more computationally efficient and accurate alternative for real-time object detection and localization in ultrasound scans.

Chapter 3 is, in full, based on the materials as they appear in the publication of “A CNN segmentation-based approach to object detection and tracking in ultrasound scans with application to the Vagus nerve detection”, Abdullah F. Al-Battal; Yan Gong; Lu Xu; Timothy Morton; Chen Du; Yifeng Bu; Imanuel R Lerman; Radhika Madhavan; Truong Q. Nguyen In

Proceedings of International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC), 2021, and “Object detection and tracking in ultrasound scans using an optical flow and semantic segmentation framework based on convolutional neural networks”, Abdullah F. Al-Battal; Imanuel R Lerman; Truong Q. Nguyen In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2022, as well as the material as it appears in “Multi-path decoder U-Net: a weakly trained real-time segmentation network for object detection and localization in ultrasound scans”, Abdullah F. Al-Battal; Imanuel R Lerman; Truong Q. Nguyen, In Computerized Medical Imaging and Graphics journal, 2023. The dissertation author was the primary investigator and author of these papers.

# Chapter 4

## Efficient In-Training Adaptive Compound Loss Function Contribution Control for Medical Image Segmentation

### 4.1 Introduction

Medical image segmentation is essential in many clinical applications. Segmentation is used to detect anatomical structures of interest as well as anomalies in the scanned body such as lesions for cancer detection, staging, and treatment planning [143]. Deep learning models are currently state-of-the-art in medical image segmentation. These models are trained by minimizing a loss function representative of the segmentation objective. One of the major challenges that these models encounter is class imbalance, where the target object is significantly underrepresented in many applications, including lesion segmentation. The compound loss function that uses binary cross-entropy (BCE), and Dice loss is one of the most significant approaches to address this issue. However, determining the contribution of each the BCE and Dice loss to the overall compound loss function is a tedious process. It requires multiple iterations of training for hyper-parameter

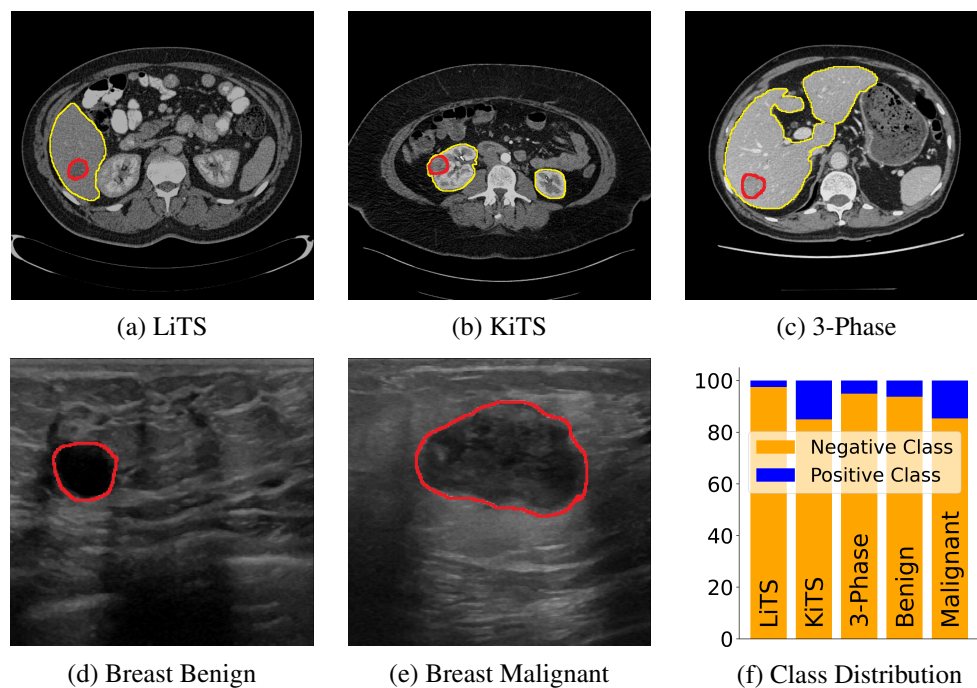


fine-tuning, which is highly inefficient in terms of time and energy consumption. To address this issue, we propose an approach that adaptively controls the contribution of each of these individual loss functions during training. This eliminates the need for multiple fine-tuning iterations to achieve the desired precision and recall for segmentation models.

For medical images, the current state-of-the-art deep learning segmentation models are based on the UNet encoder-decoder architecture with skip connections [111, 114, 144, 145]. Skip connections between the encoder and decoder improve the performance of segmentation models by propagating high-resolution details to the predicted segmentation masks. These deep learning models are trained on the target segmentation task using an iterative optimization approach, where the model weights are updated based on the gradient descent of the loss between the predicted and ground-truth segmentation masks [146]. The cross-entropy loss is the most widely used loss function in image classification tasks [15, 112] and early segmentation models [111]. In addition to the cross-entropy loss, the Dice loss is currently used to promote region-based optimization and improve localization [147]. A compound loss of the Dice and cross-entropy losses is used by state-of-the-art models to combat some of the inherent issues found in medical imaging datasets such as the large imbalance in classes, which is the case for lesion segmentation, where the lesions are severely underrepresented [119, 148].

Without selecting the proper loss functions and their hyper-parameters, such as the contribution factor for compound loss, the large imbalance in the dataset can cause significant deviations in the model predictions in terms of false positive and false negative rates, leading to large differences between precision and recall. Having balanced precision and recall is essential to optimize the model's ability to identify all positive cases while minimizing false positives and false negatives. This balance holds particular significance in medical image segmentation, where both over-diagnosis (resulting from false positives) and under-diagnosis (resulting from false negatives) can produce inappropriate treatment plans, potentially jeopardizing patient health and well-being. Manual selection of the individual loss contributions to the overall loss function [34]

and randomized grid search for hyper-parameter optimization [149] can be employed to balance the model precision and recall performance. Both of these approaches, however, require multiple training repetitions to find the most suitable hyper-parameters, rendering them highly inefficient in terms of time and energy consumption. Alternatively, weighting the loss function due to class imbalance, as is the case in weighted binary cross-entropy (WBCE) and Tversky loss [150], lacks real-time model performance monitoring and adjustment capabilities [147]; also necessitating repeated training iterations to identify the optimal weights.



**Figure 4.1:** An example slice from each of the five datasets we test the proposed approach on (a)-(e). In (a)-(c), the boundaries of the organ of interest are in yellow, while the boundaries of lesions are in red (a)-(e), overlaid on the slices. The class distribution within each dataset is represented as a percentage of the overall distribution, with the positive and negative classes corresponding to lesions and healthy tissue, respectively (f).

Therefore, we propose an approach that adaptively controls the contribution of the binary cross-entropy and Dice loss functions to the overall compound loss function during the training of segmentation models. The proposed approach eliminates the need for repeatedly training the model to find the optimal hyper-parameters. It continuously adjusts the contribution factor for the

BCE and Dice loss during training to balance the precision and recall performance of the model. We test the proposed approach on five lesion segmentation datasets. Three computed tomography (CT) datasets of the liver and kidney, and two ultrasound datasets of the breast. An example slice from each of these datasets is shown in Fig. 4.1 as well as the class distribution within each dataset. We also compare the proposed approach performance to manual hyper-parameter selection, randomized grid search, weighted binary cross-entropy, and Tversky loss approaches for the task of 2D lesion segmentation. Using significantly fewer training iterations, the proposed approach effectively balances precision and recall across the five datasets while either matching or surpassing the F1 scores of other methods.

## 4.2 Background

The compound loss function is composed of the weighted sum of the binary cross-entropy (BCE) loss and the Dice loss:

$$\mathcal{L}_{comp}(\hat{Y}, Y) = \alpha \mathcal{L}_{bce}(\hat{Y}, Y) + \beta \mathcal{L}_{D_c}(\hat{Y}, Y), \quad (4.1)$$

where  $\alpha$  and  $\beta$  are the coefficients that control the contribution of BCE loss ( $\mathcal{L}_{bce}$ ) and Dice loss ( $\mathcal{L}_{D_c}$ ), respectively, to the overall loss function.  $\hat{Y}$  and  $Y$  represent the predicted and target (ground truth) segmentation masks. For each element (pixel) of the predicted mask with a value  $\hat{y}$  and ground truth value  $y$  at location  $(i, j)$ , where  $i = 1, 2, \dots, H$  and  $j = 1, 2, \dots, W$ , for a given image width,  $W$  and height,  $H$ , the BCE loss function can be computed for each training example as:

$$\mathcal{L}_{bce}(\hat{Y}, Y) = -\frac{1}{HW} \sum_{i=1}^H \sum_{j=1}^W \ell(\hat{y}_{i,j}, y_{i,j}), \quad (4.2)$$

where  $\ell(\hat{y}_{i,j}, y_{i,j})$  is the loss computed element-wise between the ground truth and predictions, and is defined as:

$$\ell(x, y) = w_1 y \log \sigma(x) + w_0 (1 - y) \log \sigma(x). \quad (4.3)$$

In (4.3),  $\sigma(x)$  is the sigmoid function that defines  $\hat{y}$  and is defined as  $\hat{y} = \sigma(x) = 1/(1 + \exp(-x))$ .  $\sigma(x)$  maps the predicted elements into a probability space of predictions where  $\sigma(x)$  represents an object if larger than or equal to 0.5, and background otherwise.  $w_1$  and  $w_0$  are the class weights. They are both 1 for the binary cross-entropy loss and are varied for the weighted binary cross-entropy loss. The Dice loss is based on the Dice coefficient between the predicted and ground truth mask. The Dice coefficient ( $D_c$ ) is defined as [100]:

$$D_c(\hat{Y}, Y) = \frac{2 \sum(\hat{Y} \odot Y)}{\sum \hat{y}_{i,j} + \sum y_{i,j}}, \quad (4.4)$$

where  $\odot$  represents the element-wise multiplication. The Dice loss can be defined to penalize lower  $D_c$  values, which yields a lower segmentation performance, as:

$$\mathcal{L}_{D_c}(X, Y) = 1 - D_c(\hat{Y}, Y). \quad (4.5)$$

The Dice loss promotes recall more than precision because if the model fails to predict a region that is present in the ground truth (leading to a low recall), the term in the numerator of the Dice score would be small. Thus, a decrease in the numerator of the Dice Coefficient would significantly increase the Dice loss. Conversely, if the model over-predicts regions not in the ground truth (affecting precision more than recall), the increase in the denominator due to the  $\sum \hat{y}_{i,j}$  term would not be as penalizing as missing out a region entirely, which affects both the numerator and the denominator. While the BCE loss penalizes both false positives and false negatives, in many practical scenarios, especially with imbalanced data, the model tries to maximize the class with more data (often the negative or background class). This means that the

model might end up predicting fewer positives to avoid the heavy penalty of being wrong on the negative (which would reduce its precision). Thus, the BCE loss, in such scenarios, promotes precision. Hence, increasing  $\alpha$  in (4.1) relative to  $\beta$  increases the precision compared to the recall and vice versa. The precision and recall can be defined as:

$$\text{Precision} = \frac{TP}{TP + FP}, \quad \text{Recall} = \frac{TP}{TP + FN}, \quad (4.6)$$

where TP, TN, FP and FN denote true positives, true negatives, false positives and false negatives, respectively. Thus, maximizing precision and recall would minimize the false positive (FPR) and false negative rates (FNR), and minimizing the trade-off between the two metrics would minimize the trade-off between the FPR and FNR. Using TP, FP, and FN, the Tversky loss can be defined as:

$$\mathcal{L}_{Tv} = 1 - \frac{TP}{TP + \alpha_t \cdot FP + \beta_t \cdot FN} \quad (4.7)$$

In (4.7), increasing  $\alpha_t$  relative to  $\beta_t$  increases the precision compared to recall, and vice versa.

## 4.3 Method

### 4.3.1 Adaptive Loss Contribution Control

Our approach computes the precision and recall for each training batch and calculates the overall precision and recall by taking the average of each over all the batches. At the end of each training epoch, the average precision and recall are used to either increase or decrease the  $\alpha$  and  $\beta$  hyper-parameters used in the overall compound loss defined in (4.1). If the recall is higher than the precision, the approach increases  $\alpha$  and decreases  $\beta$ , but if precision is higher, it decreases  $\alpha$

and increases  $\beta$ . The approach is summarized in Algorithm 4.1.

---

**Algorithm 4.1** Adaptive Loss Contribution Control

---

```
1: procedure AT END OF TRAINING EPOCH
2:   if Recall < Precision then
3:      $\beta \leftarrow \beta + \Delta$ 
4:      $\alpha \leftarrow \alpha - \Delta$ 
5:   end if
6:   if Recall > Precision then
7:      $\alpha \leftarrow \alpha + \Delta$ 
8:      $\beta \leftarrow \beta - \Delta$ 
9:   end if
10:   $\alpha \leftarrow \max(0.1, \min(5.0, \alpha))$ 
11:   $\beta \leftarrow \max(0.1, \min(5.0, \beta))$ 
12:  return  $\alpha, \beta$ 
13: end procedure
```

---

In Algorithm 4.1,  $\Delta = 0.05$ . We chose this value as it ensures that the hyper-parameters can adapt within a few tens of epochs. If the adjustment was too small (e.g., 0.001), it would take hundreds to thousands of epochs to see a significant change in the hyper-parameters. Conversely, if the adjustment was too large (e.g., 0.5), it might lead to unstable training as the hyper-parameters would change too abruptly from one epoch to the next. Setting the maximum and minimum bounds prevents suppression by ensuring that neither  $\alpha$  nor  $\beta$  becomes overwhelmingly large compared to the other. Without these bounds, over several epochs, one of the hyper-parameters could grow or reduce significantly, thereby suppressing the effect of the other. The range of [0.1, 5.0] is wide enough to allow the model to prioritize either precision or recall as required by the task, but not too wide that it causes suppression. Therefore, the choice of  $\Delta$  and the bounds [0.1, 5.0] are designed to ensure that the algorithm can adapt the hyper-parameters to their target values efficiently, without causing instability in the training process. They also help in maintaining a balance between  $\alpha$  and  $\beta$ , ensuring one does not completely overshadow the other.

### 4.3.2 Damped Adaptive Loss Contribution Control

We also introduce a damping factor based on the difference between precision and recall. This rate is then applied to modify how aggressively  $\alpha$  and  $\beta$  are adjusted. This approach is summarized in Algorithm 4.2. It provides smoother adjustments to potentially prevent rapid oscillations or over-adjustments, but is relatively slower to adapt than the first approach summarized in Algorithm 4.1. Given the damping mechanism, the rate of 0.05 from the previous algorithm is not sufficient in scenarios where the difference is small. Therefore, to ensure a reasonable adjustment even for small imbalances, the rate is increased to 0.2 ( $\gamma$ ). Essentially, the larger rate compensates for potential reductions due to the damping effect, ensuring the adjustments remain impactful.

---

**Algorithm 4.2** Damped Adaptive Loss Contribution Control

---

```
1: procedure AT END OF TRAINING EPOCH
2:   diff  $\leftarrow$  |Recall - Precision|
3:   if Recall < Precision then
4:      $\beta \leftarrow \beta + \gamma \times \text{diff}$ 
5:      $\alpha \leftarrow \alpha - \gamma \times \text{diff}$ 
6:   end if
7:   if Recall > Precision then
8:      $\alpha \leftarrow \alpha + \gamma \times \text{diff}$ 
9:      $\beta \leftarrow \beta - \gamma \times \text{diff}$ 
10:  end if
11:   $\alpha \leftarrow \max(0.1, \min(5.0, \alpha))$ 
12:   $\beta \leftarrow \max(0.1, \min(5.0, \beta))$ 
13:  return  $\alpha, \beta$ 
14: end procedure
```

---

## 4.4 Experiments and Results

### 4.4.1 Datasets

We evaluated the proposed approach on five different medical image segmentation datasets. The first dataset is the Liver Tumor segmentation Benchmark (LiTS) dataset [119]. This dataset contains CT scans of the liver from 131 subjects for the purpose of lesion segmentation. The

second is the Kidney Tumor Segmentation Challenge (KiTS) dataset [148], which contains CT scans of the kidney from 489 subjects. The third is an internal dataset that was created by researchers at VinBrain, JSC and the University Medical Center at Ho Chi Minh City. This dataset contains contrast-enhanced 3-phase (arterial, delayed, and venous [151]) CT scans of the liver from 354 subjects. The fourth and fifth datasets are breast ultrasound datasets [125] with benign and malignant lesions from 437 and 210 scans, respectively. For each of the CT datasets the organ of interest (the liver and kidneys) was isolated through segmentation from the surrounding anatomical structures and axial 2D slices containing lesions were extracted. The Hounsfield unit range was clipped to  $[-200, 200]$ , to enhance the contrast of anatomical structures in the abdomen. The slices from all the datasets were resized to  $256 \times 256$  pixels.

#### 4.4.2 Implementation and Setup

We used a 2D UNet [111] segmentation model with a ResNet-101 [112] as its backbone network to evaluate the performance of our approach on the five datasets. We compared the ability of the proposed approach to balance the precision and recall for segmentation models with manual  $\alpha$  and  $\beta$  selection based on prior domain knowledge and the literature [34, 152, 71]. We also compared it to a randomized grid search for hyper-parameter tuning that uses the Tree-structured Parzen Estimator (TSPE) algorithm [153]. For both algorithms in our approach, we initiated  $\alpha$  and  $\beta$  with a value of 1. For the TSPE grid search approach, we used the same  $\alpha$  and  $\beta$  limits of 0.1 and 5 that we used in our approach. We trained the model for 100 epochs using our approach and the manual selection approach. For the randomized grid search, we conducted 20 trials with 20 epochs each to find the optimum  $\alpha$  and  $\beta$  hyper-parameters that would balance the precision and recall, then trained the model for 100 epochs using these hyper-parameters. We tested four different weight ratios for each of WBCE and Tversky losses by training the model for 50 epochs, then used the best weight ratios to train the model for 100 epochs. A common practice when training models on imbalanced datasets is to use weights that are inversely proportional to class



frequency distribution [154, 155], which we refer to in our analysis as WBCE (CDW). Although widely used [155], it is not effective for highly imbalanced datasets as it equates the probability of making an error in both classes. This leads to a significantly higher recall than precision for cases where the positive class is severely underrepresented as shown in Table 4.1. In our experiments, we trained the segmentation models using the AdamW optimizer [156] and Reduce on Plateau scheduler with a patience period of 10 and a reduction factor of 0.5. Our initial learning rate used for training is  $10^{-4}$  and the minimum learning rate the scheduler can use is  $10^{-6}$ .

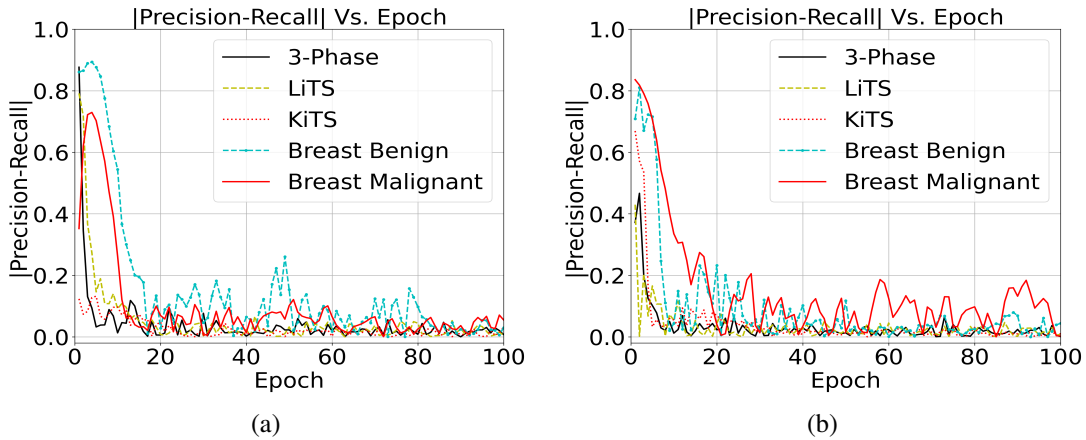
### 4.4.3 Evaluation and Results

To evaluate the performance of the proposed approach we used the average precision and recall balance, the amount of time and energy (computational power consumption over the training period), and the F1 score. We refrained from using the receiver operating characteristic (ROC) curve or the area under the ROC curve (AUC ROC) score as they are both unreliable metrics for imbalanced datasets [157]. Fig. 4.2 shows the results of the precision-recall balancing of the proposed approach for the five datasets as the model progresses during training. The proposed approach achieved a precision-recall balance with a difference of less than 0.6% consistently across the five datasets as shown in Table 4.1. It was also able to achieve this in a fifth of the time that is needed for the TSPE grid search and a third of the time needed for the WBCE and Tversky losses allowing for loss function contribution tuning during training. Since the proposed approach requires either two additions for Algorithm 4.1 or two additions and multiplications for Algorithm 4.2 per epoch, its computational complexity is negligible compared to a training iteration of the UNet model, which requires billions of multiplication-addition operations per image. Furthermore, the proposed approach achieved higher F1 scores across the five datasets improving the ability of the model to reduce both, the FPR and FNR.

During Inference, the recall and precision can be controlled and balanced by modifying the threshold value after the model’s final nonlinearity (the sigmoid function in binary cases). This,

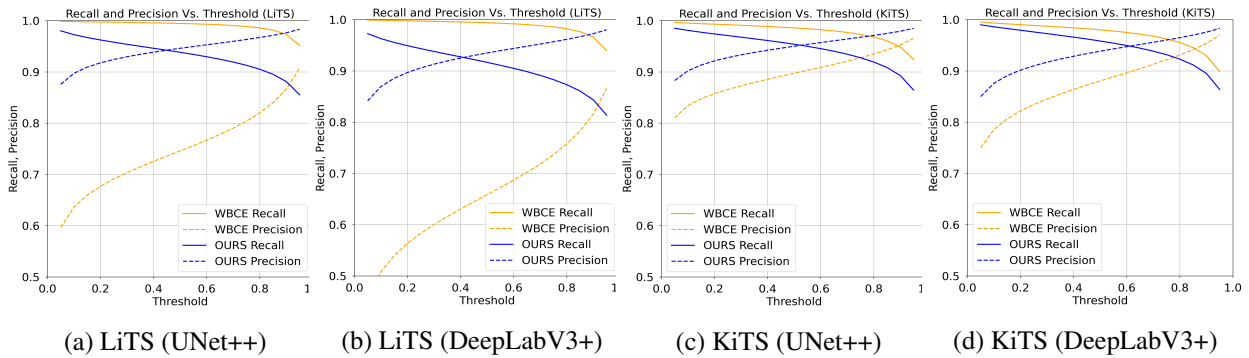
**Table 4.1:** The proposed framework precision and recall balancing performance as well as the number of epochs needed for tuning and training (TT Ep.). Pr. stands for precision, Rc. for recall, and Diff. for the absolute difference between the two. For precision and recall, higher is better while for TT Ep. and Diff. lower is better. For each of the datasets, the approach with the best balancing performance (lowest Diff.) is boldfaced. The approach producing the best F1 score is highlighted in gray while the second best is in light gray.

Approach	TT Ep.	LiTS			KiTS			3-Phase			Breast Benign			Breast Malignant		
		Pr.	Rc.	Diff.	Pr.	Rc.	Diff.	Pr.	Rc.	Diff.	Pr.	Rc.	Diff.	Pr.	Rc.	Diff.
Ours-Algorithm 4.1	100	91.5	91.2	0.3	92.8	93.4	0.6	91.9	91.7	<b>0.2</b>	78.3	77.8	0.5	78.9	79.0	<b>0.1</b>
Ours-Algorithm 4.2	100	91.8	91.5	<b>0.2</b>	93.5	93.1	<b>0.4</b>	92.5	92.3	<b>0.2</b>	80.0	79.9	<b>0.1</b>	77.8	78.2	0.4
Manual Selection	100	91.9	90.2	1.7	91.7	93.2	1.5	89.6	90.9	1.3	78.5	76.4	2.1	80.6	75.7	4.9
TSPE Grid Search	500	92.3	90.9	1.4	93.5	92.7	0.9	92.6	91.3	1.3	77.9	79.1	1.2	74.7	80.5	5.8
WBCE	300	89.9	92.2	2.3	91.9	94.4	2.5	91.3	93.0	1.7	83.3	76.7	6.6	76.6	77.1	0.5
WBCE (CDW)	100	73.1	98.8	25.7	88.8	97.7	8.9	73.7	98.7	25.0	61.3	85.5	24.2	65.6	85.9	20.3
Tversky Loss	300	89.8	92.7	2.9	91.5	94.7	2.8	90.4	91.5	1.1	73.8	82.7	8.9	75.5	81.7	1.2



**Figure 4.2:** The result of recall and precision balancing for the five datasets using the proposed Algorithm 4.1 (a) and Algorithm 4.2 (b).

however, is challenging in imbalanced datasets as shown in Fig. 4.3, and in severely imbalanced datasets might not even be possible unless the model was trained using an approach that balances these two metrics, which can be observed in Fig. 4.3 (a) and (b). This ability to control and trade-off recall and precision is important for clinicians and practitioners as it allows them to investigate the regions within a scan with respect to the model’s predictive confidence.



**Figure 4.3:** The effect of varying the segmentation mask threshold value from 0.05 to 0.95 (after applying the model’s final nonlinearity) on the precision and recall. The results are for the UNet++ and DeepLabV3+ models when trained on the LiTS dataset (a) and (b), and the KiTS dataset (c) and (d).

#### 4.4.4 Extension to Other Segmentation Models

In addition to testing our proposed approach on the UNet model, we incorporated it into training three other segmentation models that are used for medical image segmentation, which are the UNet++ [114], DeepLabV3+ [158, 159] and Attention UNet models [71]. The proposed approach was able to achieve a precision-recall balance with a difference of less than 0.4% for all the three models on both the LiTS and KiTS datasets as shown in Table 4.2. In addition to balancing the precision and recall, the proposed approach achieved higher F1 scores when compared to the baseline models trained using the WBCE loss with the inverse class distributions as weights.

**Table 4.2:** The proposed framework precision and recall balancing performance when used with the UNet++, DeepLabV3+, and Attention UNet models on the LiTS and KiTS datasets. Pr. stands for precision, Rc. for recall, and Diff. for the absolute difference between the two.

Model (Approach)	LiTS			KiTS		
	Pr.	Rc.	Diff.	Pr.	Rc.	Diff.
UNet++ (Ours)	93.9	94.0	0.1	95.1	95.1	0.0
UNet++ (WBCE)	77.9	99.1	21.2	90.3	97.4	7.1
DeepLabV3+ (Ours)	92.6	92.5	0.1	94.7	94.7	0.0
DeepLabV3+ (WBCE)	71.2	99.0	27.8	89.6	97.4	7.9
Att. UNet (Ours)	92.3	91.9	0.4	93.4	93.7	0.3
Att. UNet (WBCE)	66.5	99.2	32.7	87.1	96.2	9.1

#### 4.4.5 Sensitivity Analysis

We conducted a sensitivity analysis on the adjustment rate ( $\Delta$ ) effect on the proposed approach’s ability to balance the recall and precision, which is summarized in Table 4.3. We varied the value of  $\Delta$  outlined in Algorithm 4.1 from 0.01 to 0.1 and measured the number of epochs the proposed approach needs to achieve a recall-precision balance with a difference of less than 0.5%. Although the initial choice of  $\Delta = 0.05$  followed logical reasoning, which is explained in Sec. 4.3, we show that even deviations that span an order of magnitude are acceptable and would still achieve the desired outcome.

**Table 4.3:** The effect of the adjustment rate ( $\Delta$ ) on the number of epochs required to reach a recall-precision balance with a difference of less than 0.5% and the overall F1 score after using the respective adjustment rate.

Dataset		Adjustment Rate ( $\Delta$ )					
		0.01	0.02	0.04	0.06	0.08	0.1
LiTS	Epochs	77	51	43	21	11	13
	F1 Score	91.1	91.2	91.4	91.5	91.1	91.1
KiTs	Epochs	73	43	36	18	16	15
	F1 Score	93.2	93.3	93.2	93.1	93.1	93.1

The sensitivity of model performance to  $\Delta$  is significantly less than the sensitivity to the  $\alpha$  and  $\beta$  parameters that control the BCE and Dice loss contribution to the overall loss, which need to be fine-tuned and selected carefully to achieve reasonable segmentation results. The model’s effectiveness hinges on appropriately selecting these parameters, as they dictate the relative impact of each loss function throughout the training process. In contrast, our proposed approach involves an adaptive adjustment of the  $\alpha$  and  $\beta$  parameters during training. This adaptive adjustment is designed to optimize these parameters’ values to enhance the model’s overall segmentation performance, even if the rate at which  $\alpha$  and  $\beta$  are adjusted varies.

#### 4.4.6 Beyond Balancing The Precision and Recall

The WBCE, Tversky, and randomized grid search approaches can, in principle, perform better than the proposed approach if the hyper-parameter fine-tuning iterations were extended. However, under the time constraints, and limited availability of resources inherent in real-world scenarios, this excessive use of resources is generally undesirable.

In certain applications, prioritizing either recall or precision may be more critical, and thus, striving for a balance between these two metrics might not always yield the most desired outcome. Therefore, a direct extension to the approach we presented, is to include a margin that aims at maintaining a higher recall, or a higher precision, in both Algorithms 4.1 and 4.2. We conducted two preliminary experiments on the LiTS dataset where we conditioned the change in

the  $\alpha$  and  $\beta$  parameters on a margin of 10% between the recall and precision. For the experiment aiming at maintaining higher recall, the recall and precision were 92.9% and 84.3%, while for the experiment aiming at maintaining higher precision, they were 87.3% and 95.1%.

In addition, we extend the proposed approach by using the exponential moving average of recall and precision to modify the contribution of the loss functions; an extension of the damped approach outlined in Algorithm 4.2. This extension is particularly useful in scenarios with significant variations in training data, such as patch-wise sampling in 3D medical image segmentation in Chapter 7.

## 4.5 Conclusion

In this chapter, we presented an approach that adaptively controls the contribution of the BCE and Dice loss to the overall compound loss without the need for multiple fine-tuning iterations to achieve the desired precision and recall for segmentation models. The proposed approach was able to balance the precision and recall for five different segmentation tasks during training. It achieved comparable F1 scores to fine-tuning using randomized grid search while only requiring a fifth of the training time and computational resources to achieve these results.

Chapter 4 is, in full, based on the materials as they appear in the publication of “Efficient in-training adaptive compound loss function contribution control for medical image segmentation”, Abdullah F. Al-Battal; Soan T. M. Duong; Chanh D. Tr. Nguyen; Steven Q. H. Truong; Chien Phan; Truong Q. Nguyen; Cheolhong An In International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC), 2024. The dissertation author was the primary investigator and author of this paper.

# Chapter 5

## Multi-Phase CT Scan Registration

### 5.1 Introduction

Image registration, also known as image fusion, matching, or warping, is the process of aligning one or more images to the space of a fixed (reference) image [160]. In medical imaging, registration is a critical process in several diagnostic and therapeutic procedures, where images from multiple scans contain useful, but different information about the anatomical structure of interest [160, 161]. By aligning multiple medical images acquired at different times or ones that were acquired using different modalities, information from different images can be integrated, improving diagnostic accuracy. Consequently, image registration is an active research area within the medical image analysis field [162].

Image registration techniques can be categorized as rigid, affine, or deformable depending on the spatial transformation degrees of freedom and the scope at which the transformation varies spatially. Rigid and affine registration transformations are parameterized using a set of parameters that defines the rotation and translation operations for rigid registration in addition to scaling and shearing for affine registration. Deformable registration, on the other hand, is defined by a non-parametric dense correspondence, or the shifts in all directions for each pixel in a 2D image,

or voxel in 3D volumes, which is usually called the deformation or registration field ( $\phi$ ). This allows the alignment of different anatomical structures using a more realistic local representation of their movement beyond the constrained global rigid and affine transformations [92]. Regardless of their classification, registration methods seek to solve the problem defined as:

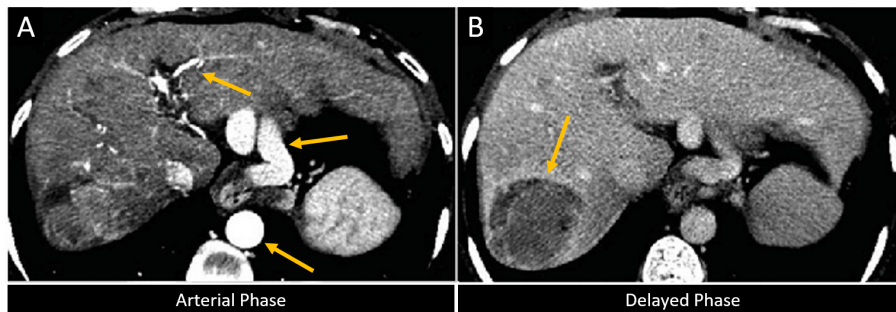
$$\hat{T} = \underset{T}{\operatorname{argmin}} \mathcal{S}(I_f, T(I_m)), \quad (5.1)$$

where  $T$  represents the registration transformation,  $\mathcal{S}$ , the similarity measure, which is correlated with the level of alignment between the image being transformed ( $I_m$ ), usually referred to as the moving image, and the reference image ( $I_f$ ), referred to as the fixed image.

In addition to aligning scans from different imaging modalities and across time, image registration is used extensively for the registration of CT scans from different phases. Multi-phase CT scans are used to highlight different anatomical structures and abnormalities within the body due to the progress of contrast agents that is injected into the body. These agents are radioactive and depending on the time the CT scan is acquired after these agents are injected, different anatomical structures appear brighter in the scan [163]. The arterial phase CT scan is acquired right after the injection of the contrast agent within a few seconds, and hence arteries in the images appear brighter than usual. For the purpose of lesion detection, this phase can highlight blood flow through the arteries to lesions. The venous phase CT scan is acquired around a minute after the injection of the contrast agent and highlights veins' interaction with tissues as well as liver parenchyma. Finally, the delayed phase is acquired several minutes after the injection of the contrast agent and is usually used to highlight various types of lesions, especially in the liver as well as fibrosis. Fig. 5.1 shows two example slices from an arterial phase scan and a delayed phase scan to demonstrate the different anatomical structures these scans highlight. These three phases are of great importance specifically for the analysis of lesions within the liver. Using scans from these three phases improves the ability of clinicians to detect liver lesions. However, due to the need for them to be acquired at different times, these scans are usually not aligned



spatially, and differences in positions of key anatomical structures such as the liver can be off by as much as several centimeters<sup>1</sup>. This misalignment can degrade clinicians' ability to detect and identify lesions within the liver, and is certainly a significant issue when designing detection or segmentation algorithms due to the misplacement of spatial features from one scan to the other. Therefore the registration of these CT scans, which we refer to as volumes too, is an essential step in the process of using multi-phase CT scans for lesion detection and segmentation.



**Figure 5.1:** A CT scan slice from an arterial phase scan (A) and a delayed phase scan (B). Yellow arrows point to the bright arteries in the arterial scan and the lesion boundary in the delayed scan.

Several methods have been proposed to register medical images and align them spatially. We discussed the major methods that have been developed in Section 2.4. In our approach, we employed multiple stages to align the scans with the purpose of aligning the liver for lesion detection and segmentation. The approach produced accurate registration where the Dice score of liver masks after registration reached an average of 95.28% ( $\pm 2.04$ ). We also proposed a learning-free approach to correct abnormal registration deformations and reduces maximum registration deformation errors by up to 6.1% when used on registration models that are based on deep convolutional networks.

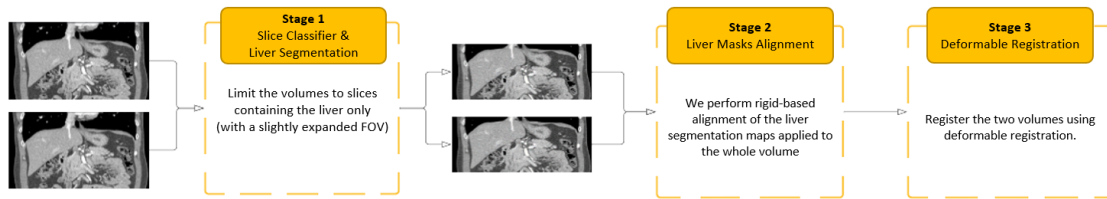
---

<sup>1</sup>These measurements are based on the analysis of an internal multi-phase dataset of the liver.

## 5.2 Method

### 5.2.1 Overview of Approach

Since our goal of registration is to align the liver across scans, our approach relies on first isolating the slices that contain the liver within the CT volume and then expanding the field of view to ensure the liver is present within the new spatially constrained volume before and after registration. Using pair-wise registration, we then perform rigid alignment of the center mass of the liver. This is achieved by segmenting the liver and computing the center location of all the voxels that belong to the liver mask. Once the center location of both the liver masks (the liver mask of the moving and fixed volume) is known, the moving volume is shifted in space so that the center location of the liver masks is aligned. After alignment, the volumes are registered using the deformable registration model, Voxelmorph [105]. The pipeline of this approach is shown in Fig. 5.2.



**Figure 5.2:** The Multi-Phase CT scan registration pipeline is outlined with its three stages for the purpose of liver registration.

We retrained the Voxelmorph model for abdominal CT registration using a semi-supervised approach. Our training scheme used randomized pairing of CT volumes from the Liver Tumor Segmentation Challenge (LiTS) dataset [45]. At each iteration, two pairs from the dataset are randomly selected and the model is trained to estimate the deformation field that would map the moving volume to the fixed volume.

## 5.2.2 Deformable Registration Architecture and Loss Function

The model was trained using a loss function composed of three terms as follows:

$$\mathcal{L}(\phi) = \mathcal{L}_{\text{NCC}}(I_f, I_m, \phi) + \beta \cdot \mathcal{L}_{\text{Dice}}(I_f, I_m, \phi) + \alpha \cdot \mathcal{L}_{\text{reg}}(\phi), \quad (5.2)$$

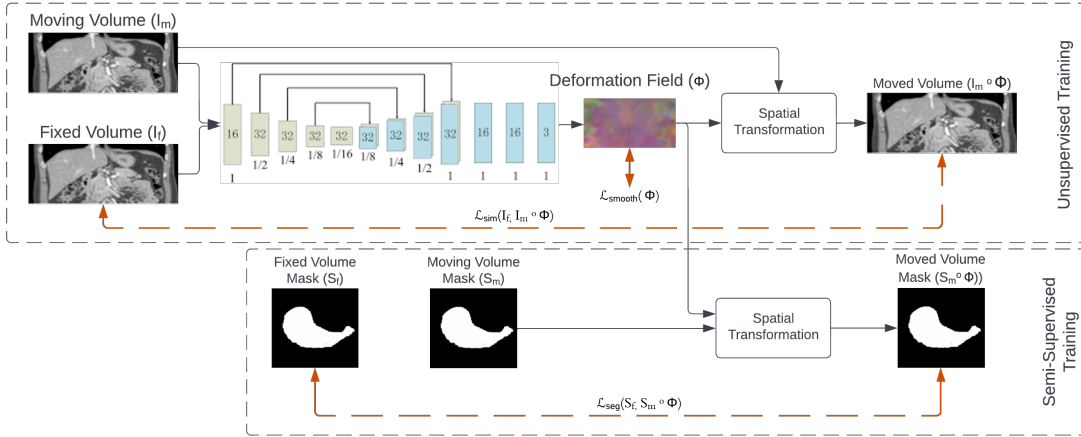
where  $\mathcal{L}_{\text{NCC}}$  represents the similarity loss between the transformed moving volume and fixed volume and is based on the normalized cross correlation,  $\mathcal{L}_{\text{Dice}}$ , the Dice loss between the segmentation map of the transformed moving volume and fixed volume,  $\mathcal{L}_{\text{reg}}$  the smoothness regularizing loss, and  $\phi$  the estimated deformation field.  $I_f$  represents the fixed volume,  $I_m$  the moving volume, and  $\alpha$  and  $\beta$  are parameters that control the contribution of the loss functions to the overall loss function. These three loss functions are defined as:

$$\mathcal{L}_{\text{NCC}}(I_f, I_m, \phi) = -\frac{\sum_x (I_f(x) - \mu_f)(I_m(\phi(x)) - \mu_m)}{\sqrt{\sum_x (I_f(x) - \mu_f)^2 \sum_x (I_m(\phi(x)) - \mu_m)^2}} \quad (5.3)$$

$$\mathcal{L}_{\text{Dice}}(S_f, S_m, \phi) = 1 - \frac{2 \sum_x S_f(x) S_m(\phi(x))}{\sum_x S_f(x) + \sum_x S_m(\phi(x))} \quad (5.4)$$

$$\mathcal{L}_{\text{reg}}(\phi) = \sum_x \|\nabla \phi(x)\|^2, \quad (5.5)$$

where  $\mu_f$  and  $\mu_m$  are the mean intensity of the fixed and moving volumes respectively.  $S_f$  and  $S_m$  are the segmentation map of the fixed and moving volumes respectively,  $x$  the spatial coordinates, and  $\nabla$  the gradient operator. The model architecture is shown in Fig. 5.3. The model takes the two volumes as inputs, they together pass through a series of convolutional blocks that follows the encoder-decoder architecture of UNet. The encoder-decoder architecture predicts a deformation field that is used to apply a spatial transformation on the moving volume to register it onto the fixed volume. The similarity loss is computed between these two volumes. The spatial transformation is applied on the segmentation map (if it is used to train the model) and the Dice loss is then computed between the moved segmentation map and the fixed one.



**Figure 5.3:** The deformable registration model architecture and framework. The model takes two CT volumes as inputs and predicts a deformation field that spatially transform the moving volume onto the fixed volume to register them. The model can be trained in an unsupervised or semi-supervised manner by using the segmentation map alignment as an additional form of supervision through the Dice loss function.

### 5.3 Experiment and Results

To achieve our goal of registering the liver from two different CT volumes we conducted a series of experiments to train and test the proposed approach on registering CT volumes for the purpose of liver registration. We started by using a pre-trained Voxelmorph model that was trained on the Open Access Series of Imaging Studies (OASIS) dataset [164] and an unsupervised rigid registration model [165] for pre-alignment based on keypoints detection using Superpoint [166], keypoint matching to estimate the transformation affine matrix by using two-way-nearest-neighborhood (TWNN) and random sample consensus (RANSAC) [167]. We observed that this approach can be unstable for certain CT volumes by producing affine transformation matrices that do not align the volumes. So, instead of using the pre-alignment model described earlier, we proposed the approach based on liver mask alignment that works on aligning the volumes within a reasonably close margin for the deformable registration model to converge. In addition to that, we retrained the deformable registration model to improve its registration performance and reduce volumes' misalignments.

### 5.3.1 Dataset and Data Preparation

For training the slice classifier, liver segmentation model, and deformable registration model, we used the LiTS dataset [45]. This dataset is originally intended for liver and liver tumor segmentation tasks, however, due to the limited number of datasets that are available for liver segmentation, we modified the dataset usage to train a registration model. For the liver deformable registration network, we trained and implemented the model on the dataset by creating 2,162 different volume combinations for training and used the segmentation map of the liver as guidance for the semi-supervised optimization of Voxelmorph. The CT volumes were resized to be on a unified physical sampling grid of 1mm in all 3 directions. The volumes were then padded with zeros to be of  $512 \times 512 \times 256$  voxels. The volumes were then resized to be  $256 \times 256 \times 128$  voxels for computational purposes before being used as inputs to the model. The volumes' intensity values (Hounsfield units) were clipped to the range  $[-250, 250]$ . The intensity values of the scans were then normalized before feeding them to the network.

#### **Slice Classifier Data Preparation and Model**

For the slice classifier, we used the ResNet-50 model [112]. We tested a set of models and compared their performance on the task and the ResNet-50 model performed best. We trained the model on slices from the LiTS dataset that were resized to  $224 \times 224$  pixels of varying physical resolutions. Physical resolutions of the slices in the LiTS dataset vary in range from 0.5 mm to approximately 1 mm. The slices' intensity values were clipped to the range  $[-250, 250]$ . The intensity values of the scans were then normalized to a range of 0 to 255, and then normalized again before feeding them to the network.

#### **Liver Segmentation Data Preparation and Model**

For the liver segmentation network, we trained and implemented the 3D ResUNet, which is a modified version of the 3D UNet model with residual connections in the convolutional blocks.

We trained the model on volumes from the LiTS dataset that were resized to  $256 \times 256 \times D$ , where  $D$  is the dimension of the volume across the z-axis and is varying depending on the span of the liver. The slices' intensity values were clipped to the range  $[-250, 250]$ . The intensity values of the scans were then normalized before feeding them to the network.

### Target Dataset for Testing

The target dataset we tested the performance of our approach on is an internal dataset and is not publicly available<sup>2</sup>. The dataset contains scans from 13 subjects. Each subject has scans from the arterial, venous, and delayed phases together with liver and lesion masks that were manually generated by specialized clinicians. Example slices from the dataset and from two different phases are shown in Fig. 5.1.

### 5.3.2 Evaluation Metrics

To evaluate the accuracy of the proposed approach, we use different evaluation metrics for the different stages within the registration framework. We use accuracy, precision, and recall for the slice classifier to evaluate its performance. These metrics are defined as follows:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}, \quad \text{Precision} = \frac{TP}{TP + FP}, \quad \text{Recall} = \frac{TP}{TP + FN}, \quad (5.6)$$

where  $TP$  stands for the number of true positives,  $TN$  the number of true negatives,  $FP$  the number of false positives, and  $FN$  for the number of false negatives. For the segmentation and registration models, we use the liver masks Dice score ( $D_c$ ), which is defined as:

$$D_c(\hat{Y}, Y) = \frac{2 \sum (\hat{Y} \odot Y)}{\sum_{m=1}^M \hat{y}_m + \sum_{m=1}^M y_m} \quad (5.7)$$

---

<sup>2</sup>The dataset is provided by VinBrain JSC. as part of a collaborative project for liver lesion detection and segmentation in multi-phase CT scans.

where  $\odot$  represents the element-wise multiplication,  $\hat{Y}$  the predicted or transformed mask, and  $Y$  the ground truth or fixed mask, and  $M$  the number of elements in the mask. In addition to the Dice score, we use the Hausdorff distance, which measures the maximum distance of all the closest distances among points from two sets. The sets in our case are the boundaries of the liver segmentation mask after registration.

**Table 5.1:** The liver slice classifiers’ performance on the LiTS dataset.

Model	Accuracy	Precision	Recall
ResNet-18	97.38%	97.37%	93.38%
ResNet-50	97.73%	97.37%	94.66%
EfficientNet-B5	96.83%	95.17%	93.75%
EfficientNet-B7	96.43%	95.54%	91.80%
ResNet-50 (with post-processing)	97.8%	97.62%	94.74%

### 5.3.3 Slice Classifier Training and Results

We used the slice classifier to detect the slices that contain liver in a CT volume. We split the dataset by volume where we used 101 volumes (subjects) for training and 30 volumes for testing. We trained the ResNet-50 model by starting our training from the pre-trained model available through the Pytorch [168] model zoo. We used a stochastic gradient descent as an optimizer with a learning rate of 0.001 and a momentum of 0.9. We also used a step-based scheduler where the learning rate is reduced by a factor of 0.2 every 5 iterations. The batch size was set to 32 and the model was trained for 100 epochs. Furthermore, we used the known anatomical structure of the liver and its continuity to post-process the classifier output by assuming that the liver is continuous and that if there are slices classified as not containing a liver within a series of slices classified as containing a liver, we know that this is a false negative and we change the classification. In addition to ResNet-50, we also trained and tested other models including ResNet-18, EfficientNet-B5 [169] and EfficientNet-B7. The performance of these models on the

test set from the LiTS dataset is summarized in Table 5.1 while the performance of the best model (ResNet-50) on the internal registration dataset is summarized in Table 5.2.

**Table 5.2:** The best slice classifier model (ResNet-50) performance on the CT multi-phase registration dataset.

Phase	Accuracy	Precision	Recall
Arterial	94.22%	95.29%	96.51%
Delayed	94.47%	95.26%	96.6%
Venous	97.3%	96.1%	96.6%

### 5.3.4 Liver Segmentation Training and Results

We trained and tested multiple segmentation models to identify, localize and estimate the boundary of the liver within a CT scan volume. We used two approaches. The first one is using 2D segmentation networks on a randomly split dataset by slices rather than volumes. The 2<sup>nd</sup> approach is using 3D segmentation models on a dataset split by volume. 3D models for liver segmentation performed better than their 2D counterparts. For the 3D segmentation model, the best-performing model that we trained on the LiTS dataset is the 3D ResUNet. For this model, the CT volumes were split into 80% training and 20% testing, and the model was trained using a learning rate of 0.0001 for 200 epochs with an early stop threshold of 30 epochs if results did not improve. The volumes were resized to  $256 \times 256$ , and 48 consecutive slices were selected at a time for each volume loaded as input to the model. Overall the model archived a Dice Score performance of 93.7%.

### 5.3.5 Deformable Registration

After aligning the volumes using the center location of segmentation maps, we performed deformable registration on the moving volume to align it with the fixed volume. We started by



testing the performance of the pre-trained Voxelmorph model. The results using this model are summarized in Table 5.3. The table shows the Dice score ( $D_c$ ) and the Hausdorff distance (HD) thresholded at the 95<sup>th</sup> percentile by slice. From the table, we can observe the importance of the alignment process using the liver segmentation map to ensure close correspondence of the two volumes spatially before deformable registration. Although not shown here, deep learning-based deformable registration would fail without close proximity between the two volumes [105].

**Table 5.3:** The proposed registration approach performance using the pre-trained deformable registration model across the different phases within the test set.  $D_c$  stands for the Dice score (higher is better), and HD stands for the Hausdorff distance at the 95<sup>th</sup> percentile by slice (lower is better).

Fixed Volume	Moving Volume	Before Alignment		After Alignment		After Registration	
		$D_c$ (%)	HD (mm)	$D_c$ (%)	HD (mm)	$D_c$ (%)	HD (mm)
Arterial	Delayed	77.8	-	90.6	-	94.8	-
Arterial	Venous	1.4	-	89.8	-	94.24	-
Delayed	Arterial	77.8	-	90.6	-	94.7	-
Delayed	Venous	1.17	-	86.7	-	93.6	-
Venous	Arterial	1.4	-	89.8	-	95.1	-
Venous	Delayed	1.17	-	86.7	-	93.2	-
Overall		25.2	38.6	89.1	10.4	94.4	7.7

We also tested the performance of the pre-trained Voxelmorph model under different conditions of slice resolution, field of view, and initial alignment. However, this pre-trained model did not perform as well as required and lacked consistency across subjects and across phases. The performance of the pre-trained model is outlined in Table 5.4 under three different conditions. The first is for a CT volume that has the slices containing the liver with additional 10 slices before and after, the second with additional 20 slices before and after, and the third with additional 30 slices before and after the liver.

To improve the performance and consistency of the model, we trained it on the LiTS dataset in a semi-supervised manner where we used the segmentation mask of the liver as guidance for the network. This is done by incorporating the Dice loss, which is computed as the negative

**Table 5.4:** The pre-trained Voxelmorph model performance on the test set under different field of view conditions. Hausdorff distance is thresholded at the 95<sup>th</sup> percentile by slice.

Condition	Dice Score	Hausdorff Distance (mm)
1 (10 slices margin)	94.4% ( $\pm 3.5$ )	7.7 ( $\pm 7.2$ )
2 (20 slices margin)	94.5% ( $\pm 3.4$ )	7.1 ( $\pm 7.6$ )
3 (30 slices margin)	94.5% ( $\pm 3.4$ )	7.5 ( $\pm 6.7$ )

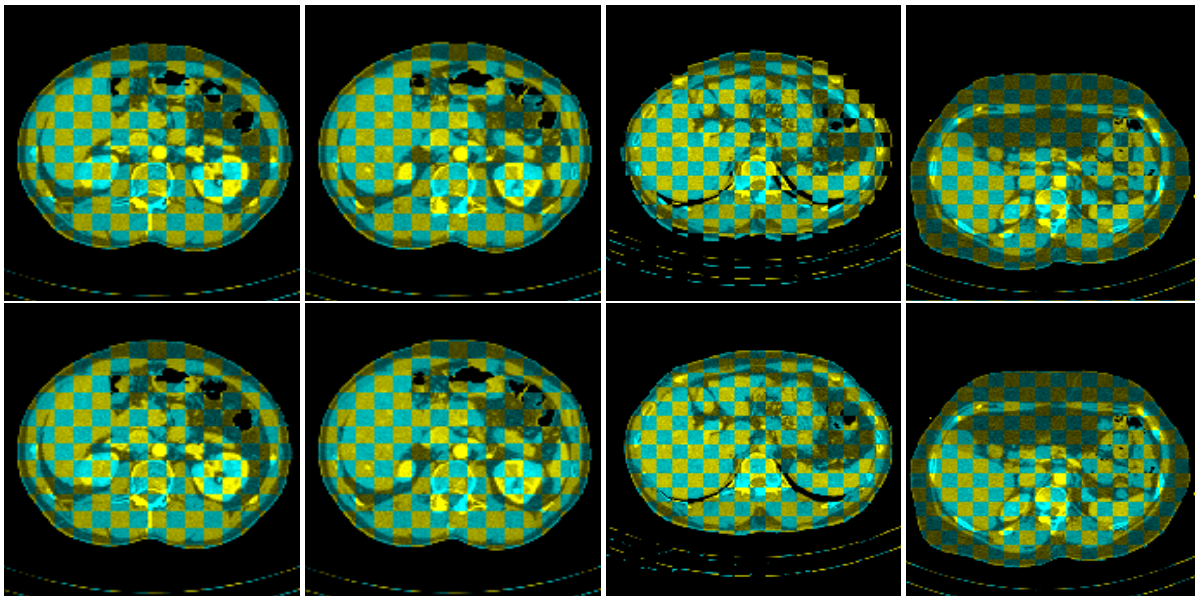
of the Dice score between the warped segmentation map and the fixed segmentation map, into the overall loss function of the model defined in (5.2). All three losses in the overall loss are weighted by a factor and then summed to form the overall semi-supervised registration loss function. In our training approach, we kept exploring the effect of increasing the Dice loss contribution to the overall loss function. As we increased the Dice loss contribution, the performance of the model in its ability to align the livers improved. Table 5.5 outlines the performance improvement of the model as we increase the contribution of the Dice loss to the overall loss function by a factor of 2, 3, and 5 when compared to the original contribution.

**Table 5.5:** Performance of the model as we increase the contribution of the Dice loss to the overall loss function by a factor of 2, 3, and 5 when compared to the original contribution. Hausdorff distance is thresholded at the 95<sup>th</sup> percentile by slice.

Dice Loss Contribution Factor	Dice Score	Hausdorff Distance (mm)
2	94.1% ( $\pm 3.6$ )	7.6 ( $\pm 4.3$ )
3	94.8% ( $\pm 2.7$ )	6.7 ( $\pm 7.6$ )
5	95.28% ( $\pm 2.04$ )	5.6 ( $\pm 2.8$ )

To train this model, the volumes were all resized to a new unified size which is  $1 \times 1 \times 1$  mm. The volumes are then padded with zeros to reach a size of  $512 \times 512 \times 256$ . They are then resized to  $256 \times 256 \times 128$  to not exceed GPU memory capacity during training. The target is to train the model for 1,500 epochs where in each epoch 100 random combinations of the volumes are used for training.

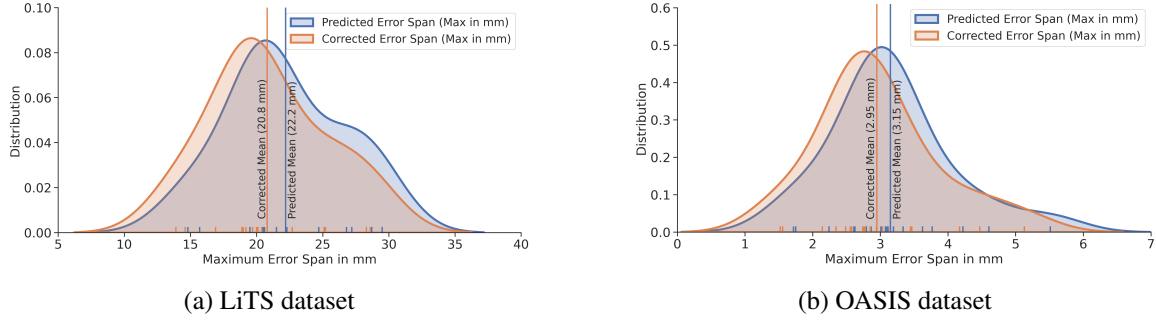
Finally, when the Hausdorff distance is thresholded by slice, the results are more restrictive and represent a closer distance to the maximum Hausdorff distance. If the distance is thresholded over the whole volume, it would be more representative of the true distance as we allow the points to be matched across the 3D volume and not by slice, and we allow the thresholding to occur over the whole volume as well. When we compute the Hausdorff distance using this approach, the distances are as follows: 13.52 ( $\pm 5.98$ ), 6.02 ( $\pm 3.47$ ), and 3.42 ( $\pm 1.62$ ) mm for the maximum, 99<sup>th</sup> percentile, and 95<sup>th</sup> percentile, respectively. Example axial slices before and after registration can be found in Fig. 5.4.



**Figure 5.4:** Example slices from four different subjects with chessboard overlay to demonstrate registration alignment across phases. In these slices, the fixed volume is from the arterial phase, and the moving volume is from the delayed phase. Top row before registration. Bottom row after registration.

## 5.4 Abnormal Registration Deformations Reduction

Regardless of their accuracy and ability to register images among different modalities, phases, or across time, deformable registration models still induce deformation errors and may



**Figure 5.5:** The distribution of the deformation error span for the predicted deformations using VoxelMorph and after correction using the proposed method.

cause artifacts in the images they transform, causing apparent abnormalities in these images or obscuring important elements that might be needed for diagnostic purposes. To help mitigate the effect of these abnormal deformations, we proposed a learning-free statistical framework based on randomized variation injection into inputs and model weights at inference to identify and reduce distortions. Fig. 5.5 shows the distribution of deformation error span with and without using our proposed approach on two different datasets. Our approach was tested on the VoxelMorph [105], SynthMorph [106], and cLapIRN [107] models, as well as two datasets: a computed tomography (CT) abdomen dataset [45] (LiTS dataset) and an MRI brain dataset [164] (OASIS dataset). Our main contributions are:

1. A learning-free framework to identify large distortions at inference for deformable registration deep-learning models.
2. A post-processing method for selective correction of deformations based on the identified large distortions locations.
3. A method that is model agnostic as it can be used with CNN deformable registration models in general, which we tested on three different models.

### 5.4.1 Method

The proposed approach aims to identify abnormal distortions in medical image registration and minimize their effect by injecting randomized variations into both the input images and model weights repeatedly. Algorithm 5.1 outlines the proposed framework. First, a set of variations is chosen randomly to be applied to the input intensity values and the model weights. Next, the registration model is used to estimate the deformation field. These two steps are repeated multiple times, and the estimated deformation field at each repetition is stored. The sample standard deviation is used to calculate the spread of estimated deformations at each voxel, and deformations with the largest spread are replaced with interpolated deformations using third-order b-spline interpolation.

#### Deformation Abnormalities Estimation and Correction

During inference, we apply the deformable registration model several times on the two image volumes. Each time we inject a randomized set of variations into the input, the model weights, or both as outlined in Algorithm 5.1. For the input volumes, the first of these variations is noise, where we add Gaussian noise with mean 0 and a standard deviation equal to 5% of the maximum volume intensity value (the volume intensity values are normalized to the range [0, 1]). The second is gamma correction, which is used to vary the contrast and luminance of different parts of the volume to simulate the use of contrast agents in CT scans. We randomly varied the gamma coefficient at each iteration between 0.95 and 1.05. The third input variation that we incorporated is the zeroing of intensity values at random. We randomly selected a number of voxels between 2.5% and 7.5% and replaced their intensity values with zero.

For model weights, the variations injected are noise and weights zeroing. For the first, we added Gaussian random noise with a standard deviation equal to 0.5% of the weight we are adding noise to. As for zeroing the model weights, we randomly selected a subset of weights ranging in size from 2.5% to 7.5% of the total set of weights at each convolutional layer. The

---

**Algorithm 5.1** Abnormal deformation estimation & correction

---

**Input:** Fixed image volume, moving image volume, synthetic ground truth deformation field, trained deformable registration model,  $N$

**Output:** Deformation field spread (standard deviation) and corrected deformation field

- 1: Align volumes using rigid transformation
  - 2: **for**  $n = 0$  **to**  $N$  **do**
  - 3:     Initialize copy of input volumes
  - 4:     **for each** input variation **in** set of input variations **do**  
       // Set of input variations: {random noise, gamma transformation,  
       random zeroing of intensity values}
  - 5:         **if** random number  $\geq 0.5$  **then**  
           // Random numbers in the range  $[0, 1]$
  - 6:             Apply input variation to the copy of the input volumes
  - 7:         **end if**
  - 8:     **end for**
  - 9:     **for each** model variation **in** set of model variations **do**  
       // Set of model variations are: {random noise, random zeroing of  
       weights (dropout at inference)}
  - 10:         **if** random number  $\geq 0.5$  **then**  
           // Random numbers in the range  $[0, 1]$
  - 11:             Apply model variation to the model weights
  - 12:         **end if**
  - 13:     **end for**
  - 14:     Compute deformation field ( $\phi_n$ ) using modified input volumes and modified model
  - 15: **end for**
  - 16: Compute the standard deviation of  $\phi_n$  at each voxel
  - 17: Replace the deformations with the largest set of standard deviations with interpolated deformations using third order B-spline interpolation
-

generated deformation fields from each iteration are then aggregated, and the standard deviation of the deformation at each voxel is computed. This is the metric we use to estimate the possibility a deformation is abnormal and not accurate. An average correlation coefficient of 0.61 is found to be present between the standard deviation and deformation error in the experiments we conducted.

Once the aggregate deformation fields are generated, and the standard deviation of the deformations at each voxel is computed, we use them to correct for abnormal deformations. At locations with high standard deviations, we replace the deformations with the interpolated deformation using third-order b-spline interpolation. The deformations selected for replacement are based on the distribution of standard deviations for that specific volume and are selected based on experimental testing of performance. We found that replacing the deformations with a standard deviation higher than the 75<sup>th</sup> percentile works best. This approach provides selective regularization on the value of deformation, smoothing the deformation field at specific locations rather than globally.

## 5.4.2 Experiment and Setup

### Dataset and Implementation

We tested the proposed approach using 3D CT scans from the Liver Tumor Segmentation (LiTS) dataset [45] and 3D MRI brain scans from the Open Access Series of Imaging Studies (OASIS) dataset [164]. The LiTS dataset contains scans of the abdomen from 131 subjects. The goal of this dataset is to benchmark liver and liver lesion segmentation models, and we adapted it to our registration work due to the extensive complexities present in abdominal scans of the body. We trained the VoxelMorph model on this dataset using 47 subjects, and we randomly selected pairs from these subjects as inputs to the model in training; there are 2,162 such pairs. The 3D CT volumes were resized so that each voxel size is  $1 \times 1 \times 1$  mm. They are then resized to  $256 \times 256 \times 128$  voxels as inputs to the model. For the OASIS dataset, we used pre-trained

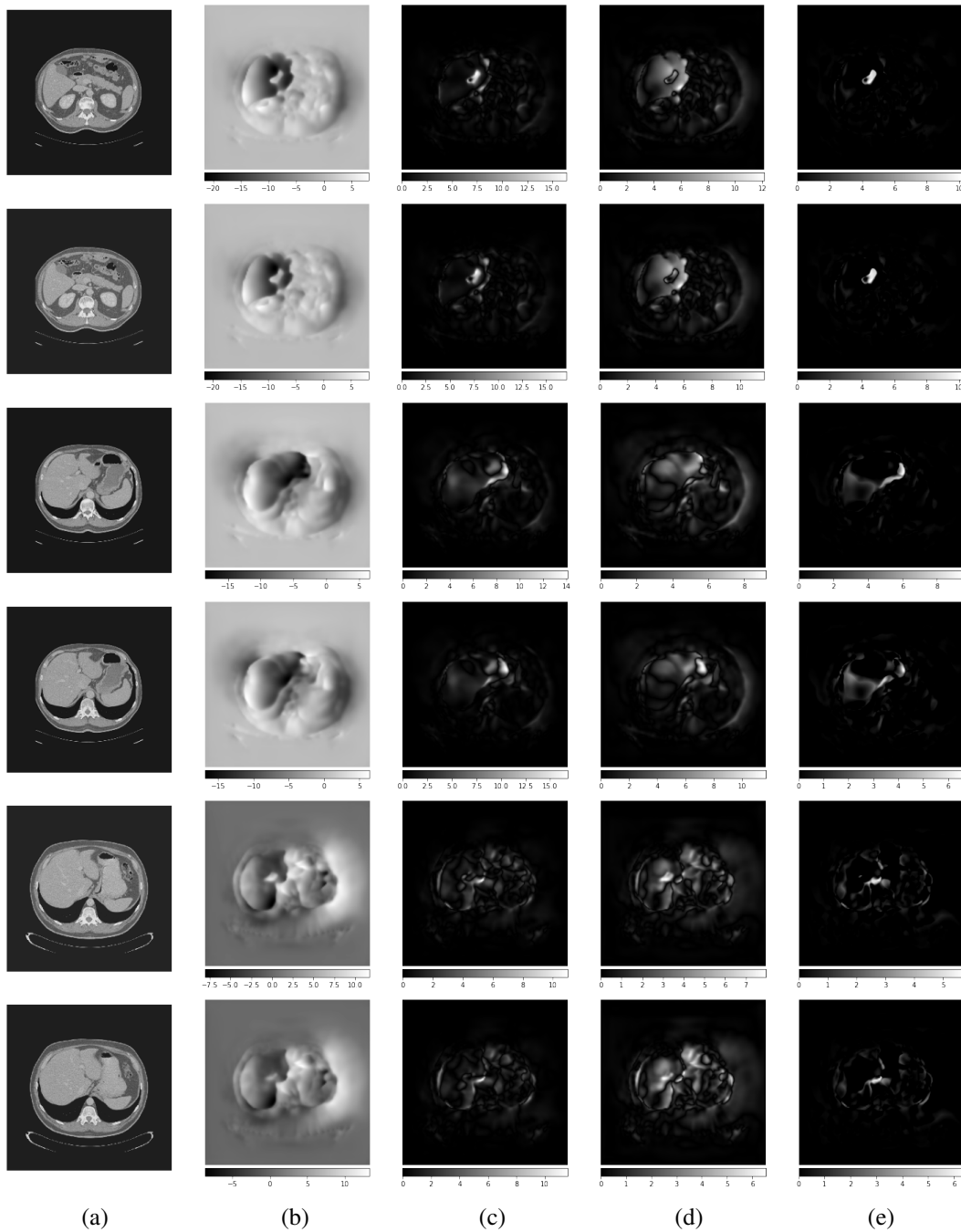
models that were made available by the authors of the three registration models we evaluated (VoxelMorph, SynthMorph and cLapIRN) on the original scan size of  $176 \times 208 \times 176$  voxels.

For both datasets, we generated ground truth deformations and applied these deformations to image volumes. We then used the registration models to predict the deformation field. Using these ground truth deformations allows us to calculate the exact difference between the predicted and actual deformation fields in spatial units, which is millimeters in our analysis. It also allows us to quantify our approach’s ability to estimate and reduce abnormal deformations. For the LiTS dataset, we used fifteen subjects, while for the OASIS dataset, we used twenty five subjects to evaluate our approach’s performance. For each of the subjects, the predicted deformation field using the registration model and the corrected one using our approach are compared to the ground truth deformation field; using the absolute error in millimeters as the evaluation metric.

### **5.4.3 Evaluation and Results**

We evaluated the proposed approach from two perspectives. The first is its ability to estimate where the predicted deformation fields would be abnormal and cause distortions by deviating significantly from the ground truth. The second is its ability to use this information to mitigate and correct these rogue deformations. Using these two perspectives, we tested the approach’s ability to estimate abnormal deformations based on a single form of variation injection first. However, we observed that using multiple types of randomized variations improved the correlation coefficient between the deformations’ standard deviation and deformation error by approximately 20% from 0.5 to 0.61. Using multiple types of randomized variations, our approach outlined in Algorithm 5.1 reduced the maximum deformation error by up to 6.1% as shown in Table 5.6. Our approach was also consistent in reducing the maximum deformation error and the abnormal deformation errors at the 99<sup>th</sup> percentile across the different datasets and models. Fig. 5.6 provides a qualitative representation of the proposed approach’s ability to reduce deformation error on the LiTS dataset and Fig. 5.8 on the OASIS dataset.



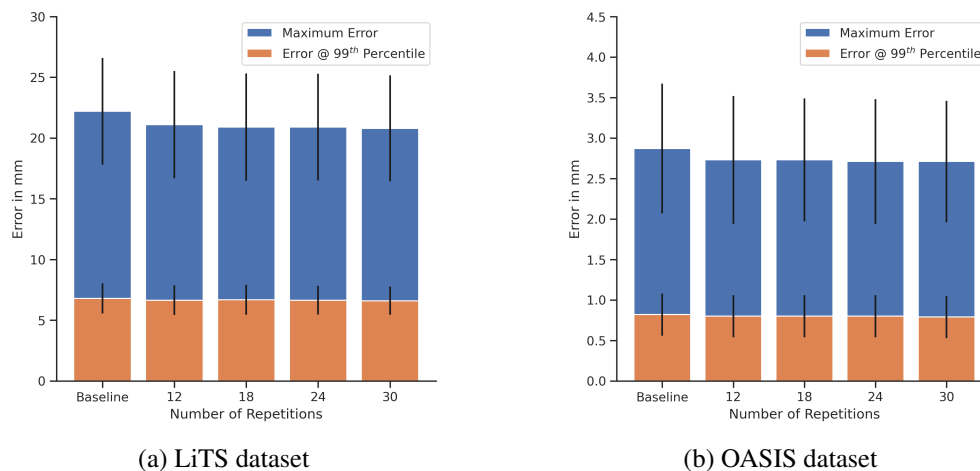


**Figure 5.6:** Qualitative results of the proposed error reduction approach on six example slices from the LiTS dataset. (a) The example slices from the LiTS dataset, (b) the ground truth registration deformation field, (c) the predicted deformation field error using VoxelMorph, (d) the corrected deformation field error using our approach, and (e) the error difference between the corrected and predicted deformation error. Deformation fields and errors colormap is in millimeters. The colormap scale for the deformations and errors is different for each image, and depends on the maximum value within that image.

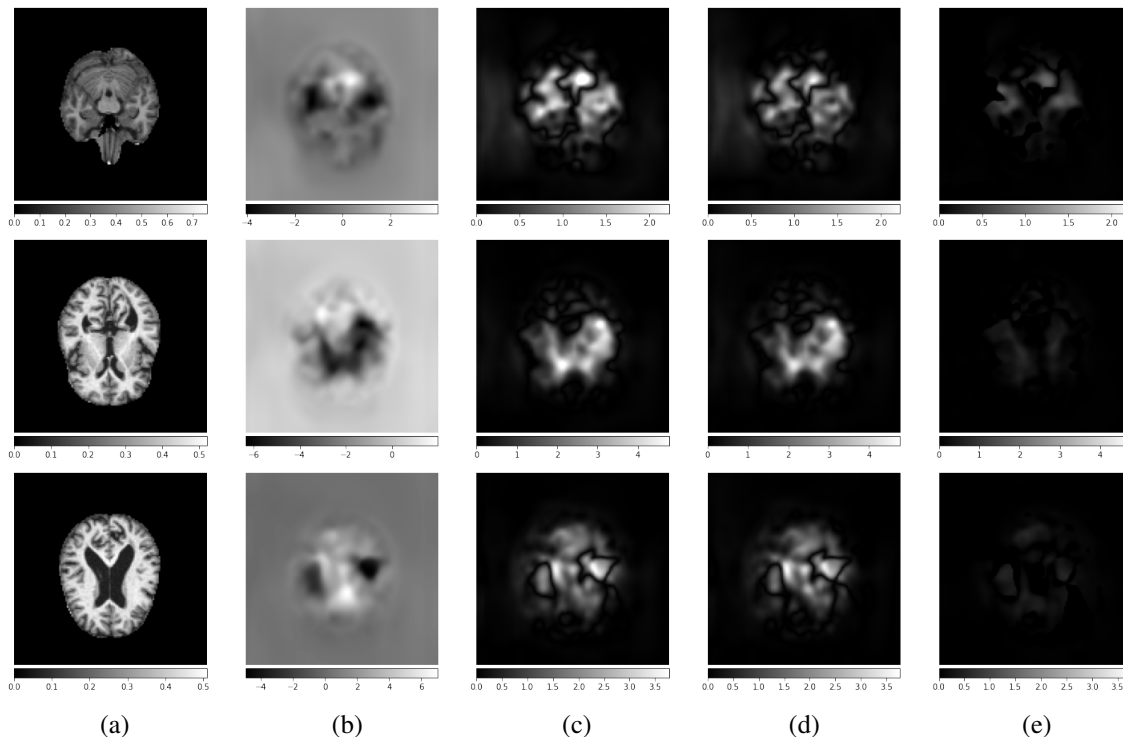
**Table 5.6:** Deformation error reduction performance of the proposed approach. The maximum deformation error and the 99<sup>th</sup> percentile error are outlined in millimeters (averaged across subjects). The improvement in reducing each of these errors is also outlined as a percentage of the original error.

Dataset	Approach	Avg. Max. Error	Avg. Error @ 99 <sup>th</sup> Perc.	Improvement (%)	
				Max. Err.	99 <sup>th</sup> % Err.
LiTS	VoxelMorph	22.2	6.8	-	-
	Proposed	20.8	6.6	6.1%	2.5%
OASIS	VoxelMorph	3.1	0.82	-	-
	Proposed	2.95	0.8	4.6%	2.3%
	SynthMorph	2.08	0.81	-	-
	Proposed	1.98	0.8	4.9%	1.2%
	cLapIRN	1.75	0.76	-	-
	Proposed	1.64	0.76	5.7%	0.3%

As the number of repetitions increases, the performance of the proposed approach tends to saturate (see Fig. 5.7). 12 to 15 repetitions were sufficient in our tests to achieve the results outlined in Table 5.6. Although the proposed approach is successful at consistently reducing the predicted deformation error without the need for training, there are still future prospects for improvement as the deformation error extends beyond 20 mm for some volumes, even after correction.



**Figure 5.7:** The number of repetitions effect (of model and input variation injection) on the correction algorithm performance.



**Figure 5.8:** Qualitative results of the proposed error reduction approach on three example slices from the OASIS dataset. (a) The example slices from the OASIS dataset, (b) the ground truth registration deformation field, (c) the predicted deformation field error using VoxelMorph, (d) the corrected deformation field error using our approach, and (e) the error difference between the corrected and predicted deformation error. Deformation fields and errors are in millimeters.

## 5.5 Limitations and Future Prospective

The registration framework proposed in this chapter focuses on solving the problem of liver registration in multi-phase CT scans. Our tests demonstrate that the framework can accurately and robustly register and align relatively small anatomical structures with respect to the image volume field of view, such as the liver in CT scans. However, the framework might not provide significant improvements when applied to the registration of larger anatomical structures with respect to the image volume field of view, such as brain scans. Although the proposed abnormal deformations reduction approach consistently reduces the predicted deformation error without the need for training, there are still opportunities for future improvements. The logical next step is to address and overcome the saturation of error reduction performance as the number of repetitions

increases. This can be achieved by modifying or expanding the types of variations injected into the input images and model weights. Another possible extension is to examine the effect of other variations on the correction performance. We observed that intensity-based variations, such as contrast modifications, produced better results than noise-based corrections, such as adding noise or randomly zeroing intensity values. We expect that these possible extensions can further enhance the capabilities of our proposed approach in mitigating large deformation errors.

## 5.6 Conclusion

In this work, we designed, implemented, and tested a framework for accurate registration of multi-phase CT scans of the abdomen with the goal of aligning the liver. The proposed approach was both accurate and robust across scans from different phases and allowed for pre-alignment before deformable registration to promote the success of these models as they need close spatial correspondence for them to accurately register the anatomical structures of interest. Furthermore, we proposed a learning-free statistical-based framework to estimate and correct abnormal deformations for deformable image registration models. We tested the framework by estimating and mitigating abnormal deformations generated by three different deformable registration models for 3D abdominal CT and 3D brain MRI scan registration. With its initial success in mitigating abnormal deformations, we hope that the proposed framework would encourage more research in this area of learning-free registration evaluation and correction.

Chapter 5 is, in part, based on the materials as they appear in “A Learning-Free Approach to Mitigate Abnormal Deformations in Medical Image Registration”, Abdullah F. Al-Battal; Soan T. M. Duong; Chanh D. Tr. Nguyen; Steven Q. H. Truong; Chien Phan; Truong Q. Nguyen; Cheolhong An, submitted to the Workshop on Biomedical Image Registration of the International Conference on Medical Image Computing and Computed Assisted Intervention (MICCAI), 2024. The dissertation author was the primary investigator and author of this paper.

# Chapter 6

## Multi-Target and Multi-Stage Liver Lesion Segmentation and Detection in Multi-Phase CT Scans

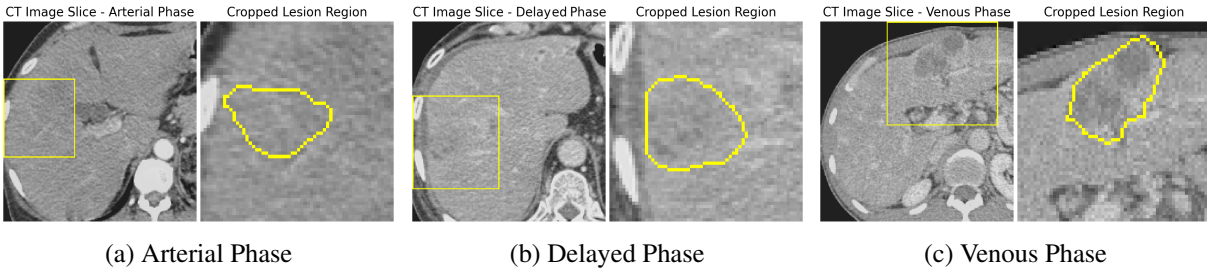
### 6.1 Introduction

Multi-phase computed tomography (CT) scans use contrast agents to highlight different anatomical structures within the body to enhance their visibility. Clinicians use these multi-phase scans to improve the chances of identifying and detecting anatomical structures of interest and abnormalities such as liver lesions. Detecting and identifying lesions within the liver is a crucial step in the diagnosis, staging, and treatment planning of liver cancer, a highly fatal form of cancer, as it is the second most common cause of premature death from cancer [170]. Nevertheless, detecting and segmenting these lesions is still a challenging task due to a multitude of reasons. They vary significantly in their size, shape, texture, and contrast with respect to surrounding tissue. Therefore, radiologists need to have extensive experience to be able to identify and detect these lesions. Yet, even radiologists with over five years of experience are still challenged by

this task, where the recall rate for liver lesions can be as low as 72% for lesions ranging in size from 10-20 mm and 16% for lesions smaller than 10 mm [171, 172]. Segmentation-based neural networks can assist radiologists with this task.

Convolutional neural networks (CNNs) are the best-performing models for medical image segmentation, especially when objects of interest are small in size such as lesions [48, 71, 111, 114]. CNNs utilize convolutional kernels within the encoder to extract features and spatially contextualize these features in the decoder. Besides CNNs, transformers have also shown significant promise in medical image segmentation by processing images on a patch-wise basis in the encoder instead of relying on convolutional kernels [115, 116]. Both CNNs and transformers have their own advantages and disadvantages. CNNs excel at capturing spatial hierarchies and local features within images, making them particularly suited for detailed and texture-rich medical images, but they struggle with long-range dependencies. On the other hand, transformers handle global context well while less capable of capturing local features without excessively large datasets and higher computational resources [173, 174].

Current state-of-the-art lesion segmentation networks use the encoder-decoder with skip connections design paradigm based on the UNet [111] architecture. In these networks, the multi-phase CT scans are fed to the network as a multi-channel input [48, 117]. Although this approach utilizes information from all the phases and outperforms single-phase segmentation networks, we demonstrate that their performance is not optimal and can be further improved by incorporating the learning from models trained on each single-phase individually. Our approach comprises three stages. The first stage identifies the regions within the liver where there might be lesions at three coarse scales (4, 8, and 16 mm). The second stage includes the main segmentation model trained using all the phases as well as a segmentation model trained on each of the phases individually. The third stage uses the multi-phase CT volumes together with the predictions from each of the segmentation models to generate the final segmentation map. We test our approach's segmentation and detection performance on a clinical three-phase CT dataset. The



**Figure 6.1:** Three slices from different subjects in the liver lesion dataset (a)-(c). Each slice is acquired at a different phase and cropped to the liver region. The lesion region of interest is highlighted in a yellow bounding box. Within each of the cropped regions, the boundary of the lesion is outlined in yellow.

dataset contains scans from 354 subjects annotated with liver lesion segmentation labels. Each subject has 3 contrast-enhanced scans at three different phases, which are the arterial, delayed, and venous phases. Example scans from this dataset showing slices from each of the phases are shown in Fig. 6.1. Overall, our approach improves relative liver lesion segmentation performance by 1.6% while reducing performance variability across subjects by 8% when compared to the current state-of-the-art models. Overall, the major contributions of our work are:

1. A multi-stage multi-target segmentation framework for liver lesions in multi-phase CT scans that use fully convolutional neural networks; improving liver lesion segmentation and detection.
2. A segmentation strategy that incorporates features learned from individual phases in both the encoder and decoder in addition to the multi-phase segmentation model for improved feature extraction and spatial contextualization.
3. A feature fusion and attention (FF&A) module in the skip connections of the main segmentation model to integrate and emphasize relevant features from both the encoder and decoder paths for an enhanced segmentation spatial focus.

## 6.2 Background

Many deep learning methods aimed to solve the challenging problem of liver lesion segmentation. These models perform better when trained on 3D image volumes rather than 2D image slices or 2.5D multi-slice images [119]. Fully convolutional networks (FCN) based on the UNet architecture are the most successful at this problem so far [111, 119]. The UNet architecture uses a symmetric structure with a contracting path (the encoder) to capture context and extract features coupled with an expanding path to localize features (the decoder) with skip connections between the encoder and decoder to combine features at different stages of the network. Several developments have been proposed to the UNet architecture to improve its segmentation and efficiency performance. These developments targeted different components of the network. Notably, these improvements included modifications to the convolutional block, such as the integration of residual or bottleneck blocks, which aid in mitigating the vanishing gradient problem and improving feature representation without substantially increasing computational complexity [112, 113]. Additionally, advancements have been made in the architecture's skip connections, with the introduction of nested and multi-stage connections that facilitate the model's ability to capture and integrate multi-scale contextual information more effectively [114]. Moreover, the incorporation of attention mechanisms within the skip connections further refines the model's focus on relevant features, enhancing segmentation precision by selectively emphasizing important spatial features while suppressing less relevant information [71].

Among the improvements that focused on redesigning the skip connections of the UNet architecture, UNet++ [114], Attention UNet [71], and MultiResUNet made large strides in improving the overall segmentation capabilities of the UNet architecture across different medical image segmentation tasks. UNet++ uses a nested architecture that refines skip connections using multiple interconnected convolutional networks at different depths, enabling an increased mixture of features across the network. Attention UNet incorporates attention gates within



its skip connections, focusing the model spatially on relevant image regions by selectively emphasizing important features while suppressing less relevant information. Other models such as the ResUNet, UNetR [115], SwinUNetR [116], and MedNext [117] improved on the UNet architecture by modifying the design of the convolutional block or replacing it with a transformer-based block. The ResUNet architecture replaces the convolutional blocks within the encoder and decoder with residual convolutional blocks while the MedNext model uses a residual bottleneck convolutional block, which is a 3D version of the ConvNext block proposed by [118]. The UNetR and SwinUNetR models replaced the encoder with a transformer-based encoder using the ViT-B model and the Swin Transformer, respectively. The MultiResUNet, on the other hand, included modifications to both the convolutional block as well as the skip connections, by incorporating multi-residual paths inspired by DenseNet in the first and successive residual blocks in the second. Transformer encoders, in general, however, struggle with smaller objects and extracting localized dense representations as they encode features using a patch-based manner. The Swin Transformer aimed to mitigate this problem with shifted windows, but it still lags behind in its ability to recover small objects, such as lesions, in their early stages.

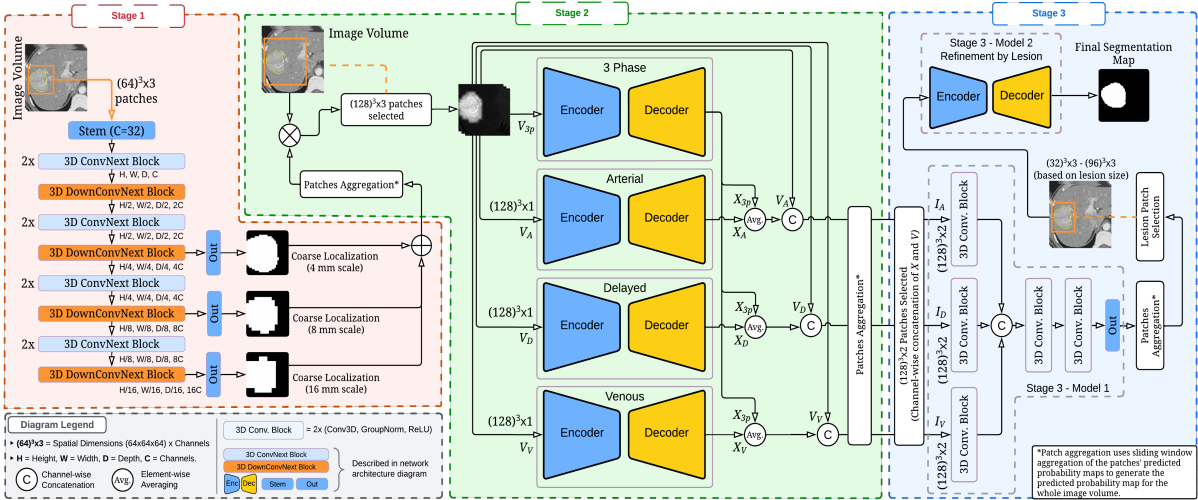
Deep learning models based on the UNet architecture are the most widely used and best performing for the task of liver lesion segmentation in single-phase CT scans [119]. These models include the current state-of-the-art model, nnUNet [48]. The nnUNet model uses the original UNet architecture and incorporates a self-configuration approach that modifies the network's depth, width, and under-sampling stages, among others, based on the dataset footprint in terms of the target anatomical structure intensity and spatial characteristics. The Model Genesis [120] UNet model, which uses self-supervised learning as a pre-training approach to learn transferable image representations, also performs comparably to the nnUNet model. Transformer models, in general, do not perform on par with CNN models due to the relatively small size of lesions to the overall scan. However, the SwinUNetR model can achieve comparable results due to the use of shifted windows, which improves local and small feature extraction for dense segmentation predictions.

chen et al. [175] proposed OctopusNet, which is a CNN-based architecture that uses separate encoder branches and a single decoder branch for multi-phase medical image segmentation. In their tests, they found that this approach improves performance when compared to using the different phases as input channels [175]. Nevertheless, current state-of-the-art approaches use the latter network design where the different phases are used as input channels.

Although these models were tested on multi-phase and multi-contrast datasets such as the Brain tumor dataset (BraTS), they were not tested on lesion segmentation in multi-phase CT scans prior to the work we present in this chapter. At the time of this chapter writing, there are currently no publicly available multi-phase liver lesion segmentation datasets. Therefore in our approach we use an internally created three-phase dataset that was annotated and rated by two different radiologists for 354 subjects with primary and secondary liver tumors. Each subject was scanned at the arterial, delayed, and venous phases of contrast injection. Multi-phase and multi-contrast models use each of the phases or contrast images as a separate input channel to the overall model while maintaining the same structure of the overall segmentation model. For example, a model trained on segmenting a three-phase CT scan will have a single input of three channels, each channel representing each of the phases individually. Although this approach incorporates information from all of the phases, we demonstrate in our work that this is not optimal and that incorporating models trained on each of the phases individually enhances the segmentation performance significantly.

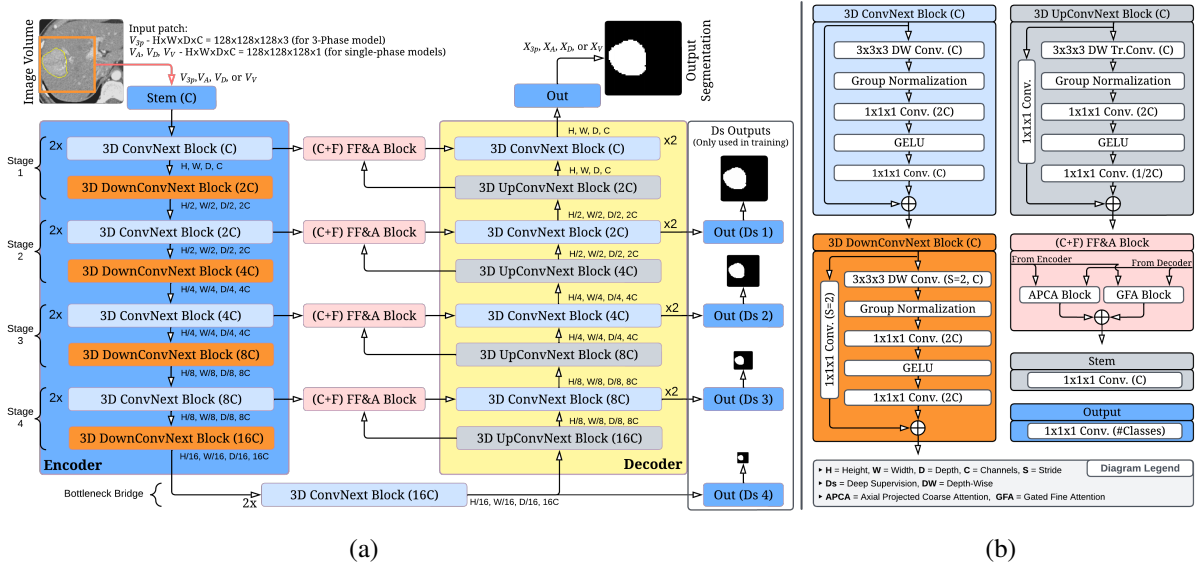
### **6.3 Proposed Method**

Effective identification and segmentation of lesions in the liver benefits significantly from imaging at different phases post contrast injection as the response of these tumors to the contrast agent at different times allow them to be more distinguishable from surrounding tissue. Hence, it is important to be able to extract features from these scans in a manner that allows the segmentation



**Figure 6.2:** The proposed framework structure with its different stages. The three outputs of stage 1 are converted to a heatmap that weights the input to stage 2. The outputs of stage 2 are concatenated with the CT image volume from each phase and then fed to stage 3. The structure of the models in stage 2 and model 2 in stage 3, as well as the structure of the Stem, Output, 3D ConvNext, and 3D DownConvNext blocks are outlined in Fig. 6.3.

model to segment them accurately. Therefore, we design our framework, which is composed of three stages as shown in Fig. 6.2, to incorporate a segmentation model trained using all the phases as multi-channel input as well as a segmentation model trained on each of the phases individually. The outcomes of these models together with the CT volumes are fed into the third stage, which is a segmentation correction and refinement stage, to generate the final segmentation map. Prior to these two stages we feed the CT volume to the first stage, which identifies the areas within the liver where there might be lesions at three different scales (4, 8, and 16 mm). In the main segmentation model we design a feature fusion and attention (FF&A) module that improves the ability of the model to combine features from the decoder and encoder in the skip connections. In this section, we explain our approach in detail, starting with the augmentation and pre-processing techniques that are used during training to promote generalization and robustness.



**Figure 6.3:** The architecture and structure of the proposed main segmentation model in stage 2 of the framework (a) with the different building blocks outlined (b): The convolutional block in the encoder and decoder, the up-sampling convolutional block, the attention and feature fusion block, the stem and output blocks. The spatial dimension and number of feature channels at the output of each of these blocks are outlined in (a). The structure of model 2 in stage 3 of the framework is the same as the main model in (a), but uses only 3 stages in the encoder and decoder (stages 1, 2, and 3) in addition to the bottleneck bridge instead of 4 stages.

### 6.3.1 Pre-Processing and Augmentation

Pre-processing and augmentation while training neural networks is an essential step to promote robustness and generalization. We use them in our proposed approach to induce variabilities into the training set stemming from the variabilities inherent in scans that would be present when the model is deployed. The proposed augmentation methods can be categorized into three main categories: a) intensity-based to account for different imaging devices and b) geometrically rigid and c) elastic transformations to account for variations in anatomical structures' shape, size and elasticity. Prior to augmentation, we pre-process the CT volume images and prepare them for our segmentation framework by clipping and normalizing their intensity values and randomly selecting a patch of size  $128 \times 128 \times 128$  voxels, which are the dimensions of the model input image. All the CT volumes are resampled to an isotropic spatial spacing of  $1 \times 1 \times 1$  mm.

For intensity-based augmentation, we randomly choose a set of options to change the visual properties of CT scans during training. We adjust properties like brightness, contrast, and noise levels. These changes teach the model to identify important features even when scans are created using different CT devices or reconstruction settings. Additionally, we simulate how a patient might be positioned slightly differently for each scan. We do this by rotating, shifting, cropping, resizing, and flipping the images. This helps the model become more reliable, ensuring it can accurately analyze scans even when there are variations in how the patient was situated during the imaging process. Additionally, we use affine transformations that include shearing and scaling to mimic the way soft tissues might stretch or compress. We also introduce elastic deformations that simulate the natural variations in shape and position of internal organs. These variations can be caused by factors like breathing, differences in patient body size, or even the presence of tumors or other abnormalities. By simulating these tissue variabilities, we make the model more robust in recognizing important anatomical features despite the inevitable differences between individual patients.

### **6.3.2 The Liver Lesion Segmentation Framework**

#### **Stage 1: Lesion localization and Patch Flagging Model**

This stage is tasked with identifying the areas within the liver that might contain lesions at three different scales (4, 8, and 16 mm). This stage’s model is outlined in Fig. 6.2. The first component of the model is a stem that expands the input patch feature width from 3 (a channel for each phase) to 32 channels. The model uses five convolutional blocks. Each convolutional block contains two 3D ConvNext blocks followed by a 3D down-sampling ConvNext block, apart from the last convolutional block, which does not use a down-sampling block. The outputs from the third (prior to down-sampling), fourth (also, prior to down-sampling), and fifth convolutional blocks are then fed to an output convolutional layer to generate a segmentation map at each of

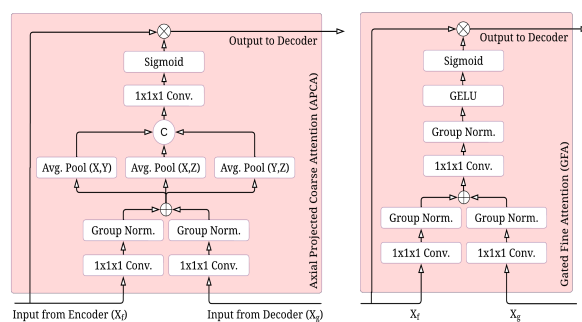
the scales, namely 4, 8, and 16 mm. These segmentation maps work as an area or patch flagging mechanism where patches at each of the scales are either flagged (containing lesion) or not. The patches are then combined to form a compound heatmap that highlight areas within the liver where the model believes there are lesions.

## **Stage 2: Lesion Segmentation Model**

In this stage we train the encoder-decoder segmentation models outlined in Fig. 6.3. We train a main model that takes all the phases as input channels as well as a model for each of the phases individually. The first component of this model is also a stem that expands the input patch feature width from 3 (a channel for each phase) to 32 channels. The model uses four convolutional blocks in the encoder and four convolutional blocks in the decoder with a bridge bottleneck after the last encoder block to connect the encoder with decoder. Each convolutional block contains two 3D ConvNext blocks followed by a 3D down-sampling ConvNext block while the bridge bottleneck contains only two 3D ConvNext blocks. The structure of the ConvNext and down-sampling ConvNext blocks are outlined in Fig. 6.3 (b). ConvNeXt blocks improve upon prior convolutional blocks design by offering greater efficiency through depth-wise separable convolutions, and incorporating transformer-inspired design paradigms such as the inverted bottleneck design. This enables ConvNext-based models to perform on par with transformer-based models on coarse computer vision tasks such as classification while outperforming them on dense prediction tasks such as segmentation, especially for small objects. To train the model, we incorporate deep supervision where the output of each convolutional block in the decoder at each depth level ( $N$ ) is fed to an output linear projection layer that contracts the channel space from  $2^{(N-1)}C$  to 2, which represents the number of classes (background versus lesion).

To improve feature mixing in the skip connection between the encoder and decoder, we incorporate a Coarse+Fine Feature Fusion & Attention Module (C+F FFA). This module includes two feature fusion and attention mechanisms; the first is Axial Projected Coarse Attention (APCA)

while the second is Gated Fine Attention (GFA), to refine feature maps by emphasizing spatially relevant information. The APCA mechanism processes features from the encoder ( $X_f$ ) and decoder ( $X_g$ ) layers, using  $1 \times 1 \times 1$  convolutions and group normalization to create a compact representations in the feature space. These are mixed and then projected across each of the three spatial dimensions using adaptive average pooling, concatenation, and further feature extraction using convolutional layers to generate coarse and smooth attention maps that is less susceptible to noise due to axial based projection and averaging. These maps are used to spatially weight the feature map from the encoder ( $X_f$ ), to enhance relevant features. The GFA module, in parallel, processes the same feature sets with  $1 \times 1 \times 1$  convolutions and group normalization, combining them without axial projection to maintain fine details of representations within the feature map. A subsequent convolution produces a spatial attention map, modulating ( $X_f$ ) to focus on important spatial regions. Together, these mechanisms enable focusing on salient features of interest, which enhances the segmentation accuracy by acknowledging spatial relationships within the image. The detailed structure of both modules is outlined in Fig. 6.4.



**Figure 6.4:** The structure of the Axial Projected Coarse Attention (APCA) module and the Gated Fine Attention module (GFA), which are the two components of the Coarse+Fine Feature Fusion & Attention Module.

### Stage 3: Segmentation Correction and Refinement

This stage is composed of two individual models. The first model incorporates the segmentation probability map from the models trained in stage 2 as well as the CT volume from

each of the phases as inputs. This model is trained on patches of the same size as the patches used in stage 2. The model has three individual encoder branches for each of the three phases followed by feature fusion through concatenation and a convolutional block, which is then followed by a single decoder branch to generate the overall segmentation map. Each of the encoder branches has two input channels. The first channel is the CT volume patch from one of the phases. The second input channel is the mean of the segmentation probability map output of the model trained on the same phase and the model trained on all phases from stage 2. This model structure is shown in Fig. 6.2.

The second model works on refining the segmentation map by lesion. For each of the lesions predicted by the first model in stage 3, this model uses an encoder-decoder structure shown in Fig. 6.3 to refine the segmentation for that lesion. To identify and separate lesions, we use morphological connectivity to identify each of the lesions, then create a bounding box around this lesion with a margin of 20% as input to this model. The segmentation map is then updated based on the segmentation output for each of the lesions individually to create the overall final segmentation map.

### 6.3.3 Model Training and Segmentation Refinement

All models in the different stages of the proposed framework are trained using a compound loss function of two loss functions. The first is the binary cross-entropy (BCE) loss function, which promotes matching the predicted mask map with the ground truth on a voxel by voxel basis globally over the whole mask without explicit consideration for overlap or object-based penalization. For a predicted mask ( $X$ ) and ground truth mask ( $Y$ ), both of size  $H \times W \times D$ , the loss is defined for each training example as:

$$\mathcal{L}_{bce}(X, Y) = -\frac{1}{H \times W \times D} \sum_{i=1}^H \sum_{j=1}^W \sum_{k=1}^D \ell(x_{ijk}, y_{ijk}), \quad (6.1)$$



where the predicted mask  $x$  and ground truth  $y$  at location  $(i, j, k)$  are used to calculate the loss  $\ell(x_{ijk}, y_{ijk})$  at each of the voxels:

$$\ell(x_{ijk}, y_{ijk}) = w_c y_{ijk} \log \sigma(x_{ijk}) + (1 - y_{ijk}) \log \sigma(x_{ijk}). \quad (6.2)$$

In (6.2),  $\sigma(x)$  is the sigmoid function defined as  $\sigma(x) = 1/(1 + \exp(-x))$ , and  $i = 1, 2, \dots, H$ ,  $j = 1, 2, \dots, W$ , and  $k = 1, 2, \dots, D$ .  $\sigma(x)$  maps the predicted voxels onto a probability space of predictions; its value indicates an object if it is larger than or equal to a threshold, and background otherwise (this threshold is usually set to 0.5, the median between 0 and 1). The second loss function is the Dice loss, which uses the Dice coefficient between the predicted and ground truth mask. The Dice coefficient promotes overlap matching and provides an explicit localized and object-based penalization. The Dice coefficient ( $D_c$ ) is defined as [100]:

$$D_c(\hat{Y}, Y) = \frac{2 \sum_{i,j,k=1}^{H,W,D} \hat{y}_{ijk} \odot y_{ijk}}{\sum_{i,j,k=1}^{H,W,D} \hat{y}_{ijk} + \sum_{i,j,k=1}^{H,W,D} y_{ijk}}, \quad (6.3)$$

where  $\odot$  is element-wise multiplication and  $\hat{y}_{ijk} = \sigma(x_{ijk})$ . The loss based on the Dice coefficient ( $D_c$ ) promotes higher  $D_c$  values while countering lower ones. It is defined as:

$$\mathcal{L}_{D_c}(X, Y) = 1 - D_c(\hat{Y}, Y). \quad (6.4)$$

The total compound loss function is defined as the weighted sum of the two losses:

$$\mathcal{L}_{obj}(X, Y) = \alpha_b \mathcal{L}_{bce}(X, Y) + \alpha_d \mathcal{L}_{D_c}(X, Y), \quad (6.5)$$

where the coefficients  $\alpha_b$  and  $\alpha_d$  control the contribution of each loss to the total loss function. In our implementation we chose  $\alpha_b = \alpha_d = 1$ .

### Stage 3 Model Training

In stage 3, the first model processes inputs from the three phases of CT scans and the segmentation probability maps to generate an initial segmentation map. Each encoder branch takes a CT volume patch and the averaged segmentation probability maps as input. For each phase  $p \in \{A, D, V\}$ , where  $A$  stands for arterial,  $D$  for delayed, and  $V$  for venous. The input to the model is formed as follows:

$$I_p = \mathfrak{C}_C(V_p, \frac{1}{2}(X_p + X_{3p})), \quad (6.6)$$

where  $\mathfrak{C}_c$  represents channel-wise concatenation,  $V_p$  the CT volume patch from each of the phases, and  $X_p$  and  $X_{3p}$  are the segmentation probability maps from the models trained on this phase and three phases in stage 2, respectively. Each of the encoders generates a feature map ( $E_p$ ) from each of the phases individually, which we define as  $E_p = \mathfrak{F}_E(I_p)$  where  $\mathfrak{F}_E$  represents the encoder feature extraction operations outlined in Fig. 6.2. The overall output segmentation map from this model is then computed as follows:

$$X_R = \mathfrak{F}_D \mathfrak{C}_C(E_A, E_D, E_V). \quad (6.7)$$

In (6.7),  $\mathfrak{F}_D$  represents the decoder operations, and  $\mathfrak{C}_c$  channel-wise concatenation.  $E_A$ ,  $E_D$  and  $E_V$  are the encoder feature outputs of the arterial, delayed, and venous phases. For the second model in stage 3, the input to the model is formed by isolating each of the lesions in  $X_R$  using morphological connectivity and identifying the width, height, and depth of the lesion. We then crop a bounding box that is 20% larger than the lesion span in each of these dimensions from the CT volume  $V$ . Finally we iteratively update  $X_R$  with the refined segmentation map of each of the lesions as follows:

$$X_R = \sum_{l=1}^L \mathfrak{F}_{M2}(V_l), \quad (6.8)$$

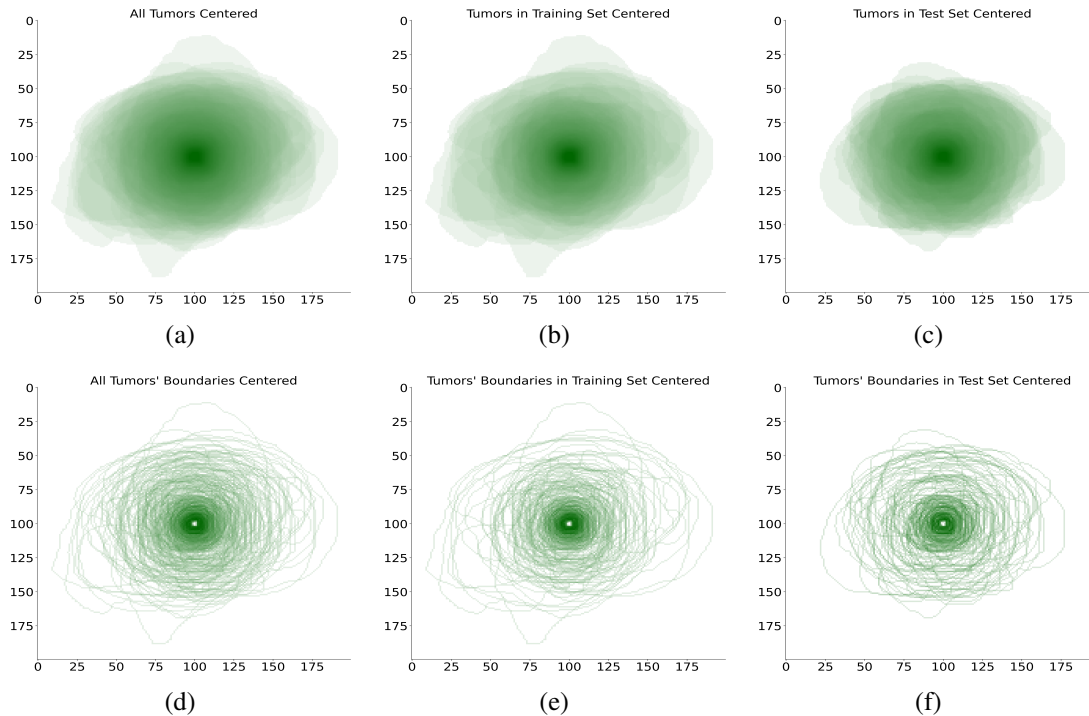
where  $\mathfrak{F}_{M2}$  represents the second model operations and  $V_l$  the cropped region from the three-phase CT volume for each of the lesions  $l = 1, 2, \dots, L$ .

## 6.4 Experiments And Results

### 6.4.1 Datasets

We evaluated our model on 2 different datasets. The first dataset is the main dataset we used to test our approach. It is a clinical dataset that was developed by researchers and clinicians at VinBrain, JSC and the University Medical Center at Ho Chi Minh City. This dataset contains contrast-enhanced 3-phase (arterial, delayed, and venous) CT scans of the liver from 354 subjects. The dataset was annotated and rated by two radiologists with extensive experience in liver oncology. Each of the axial scans in the dataset has a resolution of  $512 \times 512$  pixels at a physical spacing that ranges from 0.5 mm to 0.84 mm with an average of 0.66mm. Slice thickness in the dataset ranges from 0.5mm to 1mm with an average of 0.9 mm. The average number of lesions in each of the scans is 2.2 while the maximum is 11 and the minimum is 1. Lesions are of significantly varying sizes with lesions as large as 129 mm and as small as 2.7 mm in diameter present within the dataset scans. Masks of these lesions and their boundaries overlaid using their largest axial slice are shown in Fig. 6.5 to demonstrate the variability in lesion sizes and boundaries within the dataset. The scans from the arterial phase and delayed phase were registered on the venous phase for each of the subjects. This dataset have lesions with varying shapes, sizes, and semantics with respect to healthy liver tissue, which forms a reasonable challenge to test the performance of the proposed framework.

In addition to the main dataset, we test our approach on the BraTS19 [176, 177] dataset to evaluate its ability to extend its performance improvements to other multi-phase and multi-contrast datasets beyond liver lesion segmentation. The BraTS19 dataset contains MRI scans of the brain from 484 subjects for the purpose of brain tumor segmentation. For each subject, the



**Figure 6.5:** The largest axial slice of each liver lesion in the whole dataset (a), the training set (b), and the test set (c) overlaid onto one image at a scale of 1 mm. The boundary of these lesions in the same axial slice for the whole dataset (d), training set (e), and test set (f).

dataset contains 4 multi-contrast scans, which are: a) native T1, b) post-contrast T1-weighted, c) T2-weighted, and d) T2 Fluid Attenuated Inversion Recovery (FLAIR). The dataset was annotated by 1 to 4 raters and these annotations were approved by experienced neuroradiologists. All the scans are distributed after they have been pre-processed through co-registration to the same anatomical template atlas, interpolated to the same spatial resolution of 1 mm, and skull-stripped.

## 6.4.2 Experimental Setup and Data Preparation

For each of the datasets, we trained the models in the proposed framework from randomly initiated weights. For the liver lesion dataset the scans were split into 200 for training and 154 for testing while for the BraTS dataset the split was 387 for training and 97 for testing. For the liver lesion dataset, the liver was segmented using a model trained on the LiTS dataset and scans with only the liver region were used for the training and testing of the proposed lesion segmentation

approach. The BraTS dataset is already pre-processed with the organ of interest (brain) isolated. We used a spatial resolution of  $1 \text{ mm}^3$  for both datasets and patches of size  $128 \times 128 \times 128$  voxels for the models in stage 2 and 3, and patches of size  $64 \times 64 \times 64$  voxels for the model in stage 1. The CT scans intensity values which are represented by the Hounsfield units were clipped to the range  $[-200, 200]$  before normalization. We compared the proposed approach performance to four 3D segmentation networks, which are the current leading models across different medical segmentation tasks. These models are the SwinUnetR [116], Model Genesis [120], nnUNet [48], and MedNext [117]. We also compared our approach to OctopusNet [175], which is specifically designed for multi-phase and multi-contrast medical images. Furthermore, we incorporate the two best performing models out of these five, which are the nnUNet and MedNext models, as the stage 2 models in our framework to demonstrate the ability of the overall framework we propose to improve the segmentation output of these models. The models were trained for 800 to 1200 epochs depending on the learning rate that is suitable for the model, which ranged from  $1e^{-4}$  to  $1e^{-2}$ . All the models were trained using the AdamW [156] optimizer and loss function defined in (6.5), except for the nnUNet and Model Genesis models, which are trained using the stochastic gradient descent optimizer as it is the recommended optimizer for both models. At each iteration within an epoch, we select two patches randomly from the image volume using a weighted sampling scheme that gives a 50% higher weight to the probability of sampling a patch containing a lesion. The second model in stage 3 was trained using a single patch at a time as the patches were of varying size depending on the lesion size.

### **6.4.3 Evaluation and Results**

#### **Evaluation Metrics**

To evaluate the performance of our proposed architecture, we use multiple segmentation, detection and localization metrics. For segmentation, we use the Dice score and the intersection

over union (IoU) metrics. Both metrics are considered the benchmark metric used to evaluate detection, and segmentation methods [91]. The Dice score is defined in (6.3) while the IoU metric is calculated as follows:

$$IoU(\hat{Y}, Y) = \frac{\sum(\hat{Y} \odot Y)}{\sum(\hat{Y} | Y)}, \quad (6.9)$$

where  $\odot$  is element-wise multiplication,  $|$  is the element-wise *or* logical operator,  $\hat{Y}$  is the predicted mask map, and  $Y$  is the ground truth. For the Dice score, we measure it globally by aggregating true positives (TP), false positives (FP), and false negatives (FN) over all subjects and then computing the Dice score as follows:

$$D_c = \frac{2TP}{2TP + FP + FN} \quad (6.10)$$

We also measure the Dice score by subject to capture the variability in model performance across subjects. Furthermore, we evaluate the segmentation recall ( $TP/(TP + FN)$ ) and precision ( $TP/(TP + FP)$ ) by subject as well as the surface Dice score at a tolerance of 1.5mm.

### Overall Segmentation Performance

The segmentation performance of the proposed approach is summarized in Table 6.1, and is compared to the other 5 benchmark models. In section I of Table 6.1 we compare the performance of segmentation models only (stage 2 for our framework) without the use of the whole framework. In these models, the input is either a single channel 3D image patch for individual phases or a 3-channel 3D image patch for the 3-phase case. In section II of the table we summarize the performance of the whole framework. The proposed approach constantly outperformed the 5 benchmark models, and was able to improve the feature extraction and segmentation of lesions from the different phases leading to a 1.6% relative Dice score improvement (76.3% versus 75.1%) when compared to the best performing benchmark model. This performance improvement translates to better segmentation maps of lesions with better detection and boundaries as shown

**Table 6.1:** The proposed framework liver lesion segmentation performance on the multi-phase CT dataset. Section I outlines the results of just the segmentation model (stage 2 for our framework). For each of the arterial, delayed, and venous phases, the input number of channels is 1 while for 3-Phase, input channels are 3 (arterial, delayed and venous stacked). Section II outlines the performance of the overall proposed framework with our proposed segmentation model in stage 2 as well as the nnUNet and MedNext models. Best and 2<sup>nd</sup> best results are boldfaced for each category in Section I while only the best is boldfaced in Section II. All metrics are in the range 0 to 100. Values in parentheses represent the standard deviation across subjects.

Phase	Model	Global	By Subject				Surface
		Dice	Dice	IoU	Recall	Precision	Dice
Section I: Segmentation Model Results							
Arterial	SwinUNetR	71.5	60.3 ( $\pm$ 28.5)	48.6 ( $\pm$ 26.8)	62.4 ( $\pm$ 30.3)	68.6 ( $\pm$ 30.3)	43.8
	Model Genesis	75.8	63.1 ( $\pm$ 26.1)	50.8 ( $\pm$ 25.0)	72.4 ( $\pm$ 26.0)	63.2 ( $\pm$ 28.0)	45.5
	nnUNet	77.2	65.7 ( $\pm$ 26.5)	54.0 ( $\pm$ 25.9)	65.1 ( $\pm$ 29.0)	<b>76.7</b> ( $\pm$ 25.8)	51.1
	MedNext	<b>80.7</b>	<b>69.5</b> ( $\pm$ 23.7)	<b>57.6</b> ( $\pm$ 24.2)	<b>75.9</b> ( $\pm$ 23.0)	71.6 ( $\pm$ 26.7)	<b>54.0</b>
	Ours	<b>80.8</b>	<b>69.8</b> ( $\pm$ 24.0)	<b>58.0</b> ( $\pm$ 24.2)	<b>75.7</b> ( $\pm$ 23.4)	<b>72.1</b> ( $\pm$ 26.8)	<b>54.3</b>
Delayed	SwinUNetR	79.5	61.6 ( $\pm$ 28.6)	50.0 ( $\pm$ 26.8)	66.5 ( $\pm$ 30.8)	66.6 ( $\pm$ 30.4)	44.1
	Model Genesis	78.5	62.4 ( $\pm$ 28.2)	50.8 ( $\pm$ 26.7)	<b>67.6</b> ( $\pm$ 29.4)	66.4 ( $\pm$ 30.2)	45.7
	nnUNet	81.8	64.3 ( $\pm$ 29.5)	53.3 ( $\pm$ 27.6)	64.4 ( $\pm$ 31.4)	73.2 ( $\pm$ 29.1)	48.8
	MedNext	<b>82.7</b>	<b>67.3</b> ( $\pm$ 27.1)	<b>56.0</b> ( $\pm$ 26.1)	66.7 ( $\pm$ 28.6)	<b>76.8</b> ( $\pm$ 25.6)	<b>52.6</b>
	Ours	<b>83.0</b>	<b>67.4</b> ( $\pm$ 27.0)	<b>56.1</b> ( $\pm$ 26.1)	<b>66.9</b> ( $\pm$ 28.8)	<b>76.8</b> ( $\pm$ 25.5)	<b>52.8</b>
Venous	SwinUNetR	80.0	63.6 ( $\pm$ 26.6)	51.6 ( $\pm$ 25.7)	66.3 ( $\pm$ 28.3)	69.9 ( $\pm$ 29.0)	47.5
	Model Genesis	79.4	63.3 ( $\pm$ 29.7)	52.3 ( $\pm$ 27.8)	68.0 ( $\pm$ 30.8)	67.4 ( $\pm$ 30.7)	47.8
	nnUNet	<b>82.9</b>	65.5 ( $\pm$ 29.2)	54.6 ( $\pm$ 27.8)	65.0 ( $\pm$ 30.8)	<b>73.6</b> ( $\pm$ 29.6)	51.5
	MedNext	82.1	<b>67.8</b> ( $\pm$ 26.5)	<b>56.3</b> ( $\pm$ 26.2)	<b>69.9</b> ( $\pm$ 27.6)	73.4 ( $\pm$ 28.2)	<b>54.0</b>
	Ours	<b>82.2</b>	<b>67.8</b> ( $\pm$ 26.4)	<b>56.5</b> ( $\pm$ 26.1)	<b>69.8</b> ( $\pm$ 27.4)	<b>73.5</b> ( $\pm$ 28.0)	<b>54.1</b>
3-Phase	SwinUNetR	81.6	68.2 ( $\pm$ 23.2)	55.8 ( $\pm$ 23.5)	68.1 ( $\pm$ 25.0)	78.0 ( $\pm$ 23.6)	53.1
	Model Genesis	80.8	70.7 ( $\pm$ 22.4)	58.7 ( $\pm$ 22.8)	73.4 ( $\pm$ 23.5)	76.6 ( $\pm$ 23.2)	57.6
	OctopusNet	75.5	67.2 ( $\pm$ 22.7)	54.4 ( $\pm$ 22.5)	64.2 ( $\pm$ 24.3)	<b>81.2</b> ( $\pm$ 23.5)	52.4
	nnUNet	83.7	73.4 ( $\pm$ 22.4)	62.0 ( $\pm$ 23.0)	74.8 ( $\pm$ 24.2)	79.4 ( $\pm$ 22.0)	61.8
	MedNext	<b>83.9</b>	<b>75.1</b> ( $\pm$ 20.1)	<b>63.6</b> ( $\pm$ 21.1)	<b>76.8</b> ( $\pm$ 22.6)	80.3 ( $\pm$ 19.1)	<b>64.1</b>
	Ours	<b>84.1</b>	<b>75.5</b> ( $\pm$ 19.8)	<b>63.9</b> ( $\pm$ 20.8)	<b>76.8</b> ( $\pm$ 22.1)	<b>80.9</b> ( $\pm$ 18.9)	<b>64.4</b>
Section II: Overall Framework Results							
Overall	Ours (nnUNet)	84.2	74.1 ( $\pm$ 22.1)	62.8 ( $\pm$ 22.7)	77.7 ( $\pm$ 22.8)	77.3 ( $\pm$ 22.2)	62.7
	Ours (MedNext)	84.6	75.8 ( $\pm$ 18.9)	63.8 ( $\pm$ 20.9)	79.1 ( $\pm$ 19.5)	78.6 ( $\pm$ 20.8)	64.5
	Ours (Ours)	<b>85.1</b>	<b>76.3</b> ( $\pm$ 18.5)	<b>64.6</b> ( $\pm$ 20.3)	<b>79.1</b> ( $\pm$ 20.0)	<b>80.0</b> ( $\pm$ 19.4)	<b>65.0</b>

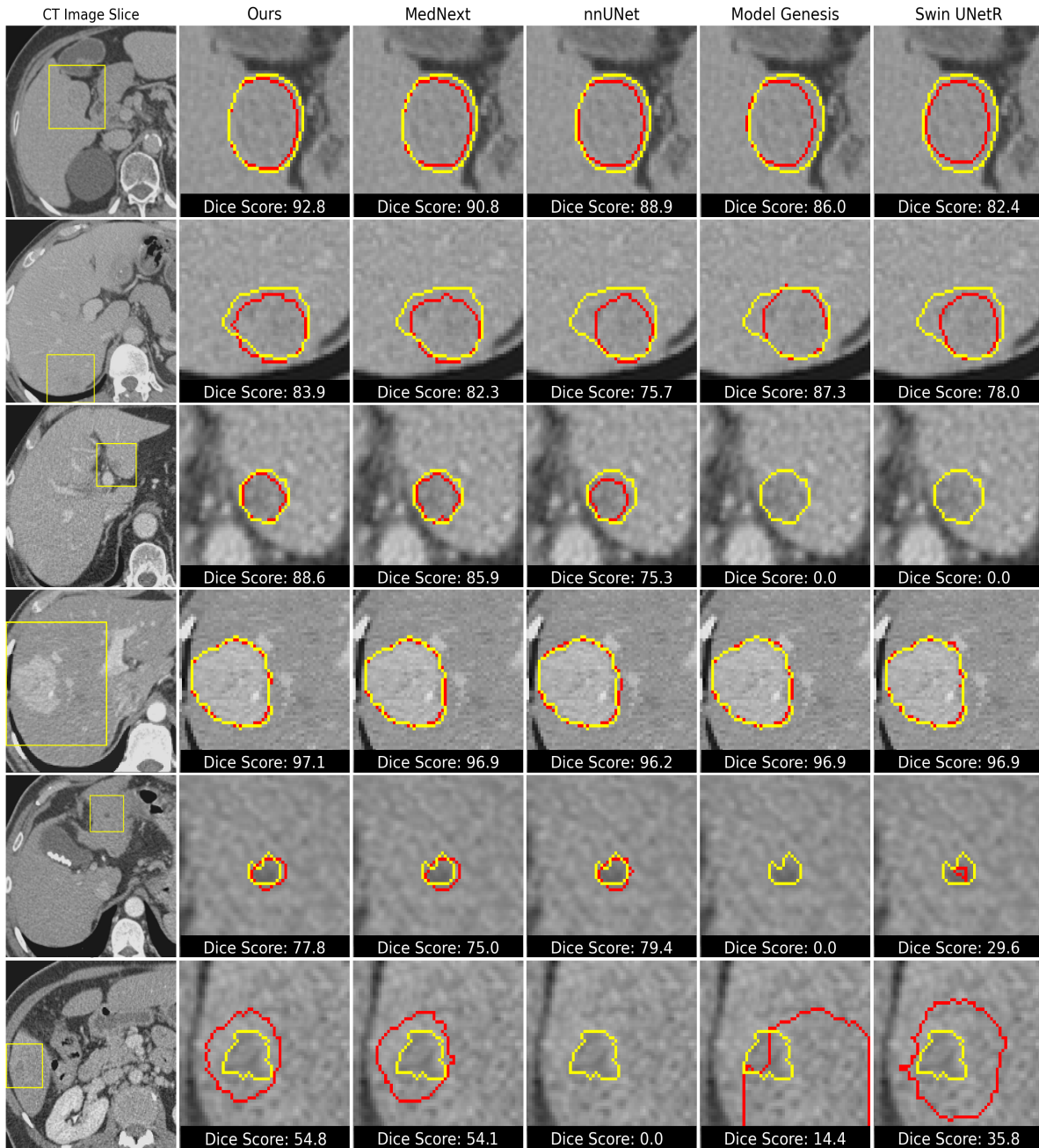
in Fig. 6.6, Fig. 6.7 and Fig. 6.8. In Fig. 6.6, lesions with different characteristics from multiple subjects are shown together with the ground truth and predicted segmentation boundaries from our proposed framework and other baseline models. In Fig. 6.8, we demonstrate the ability

of the proposed framework to improve lesions' segmentation when compared to the 3-Phase segmentation model. The 3D surface of predicted segmentation masks using our proposed framework overlaid on the surface of ground truth masks for lesions of different sizes are outlined in Fig. 6.7, demonstrating the ability of the proposed framework to segment lesions accurately. This improvement in performance is attributed to all the stages of the framework, where the first stage improves the segmentation model Dice score performance across subjects by 0.1% while the third stage improves the overall performance by 0.7%. The overall framework can also integrate with other segmentation models that can be used instead of our proposed model in stage 2 of the framework. We conducted two experiments with the best two benchmark models. A relative Dice score performance improvements of 1% were observed for both the nnUNet and MedNext models as shown in Section II of Table 6.1.

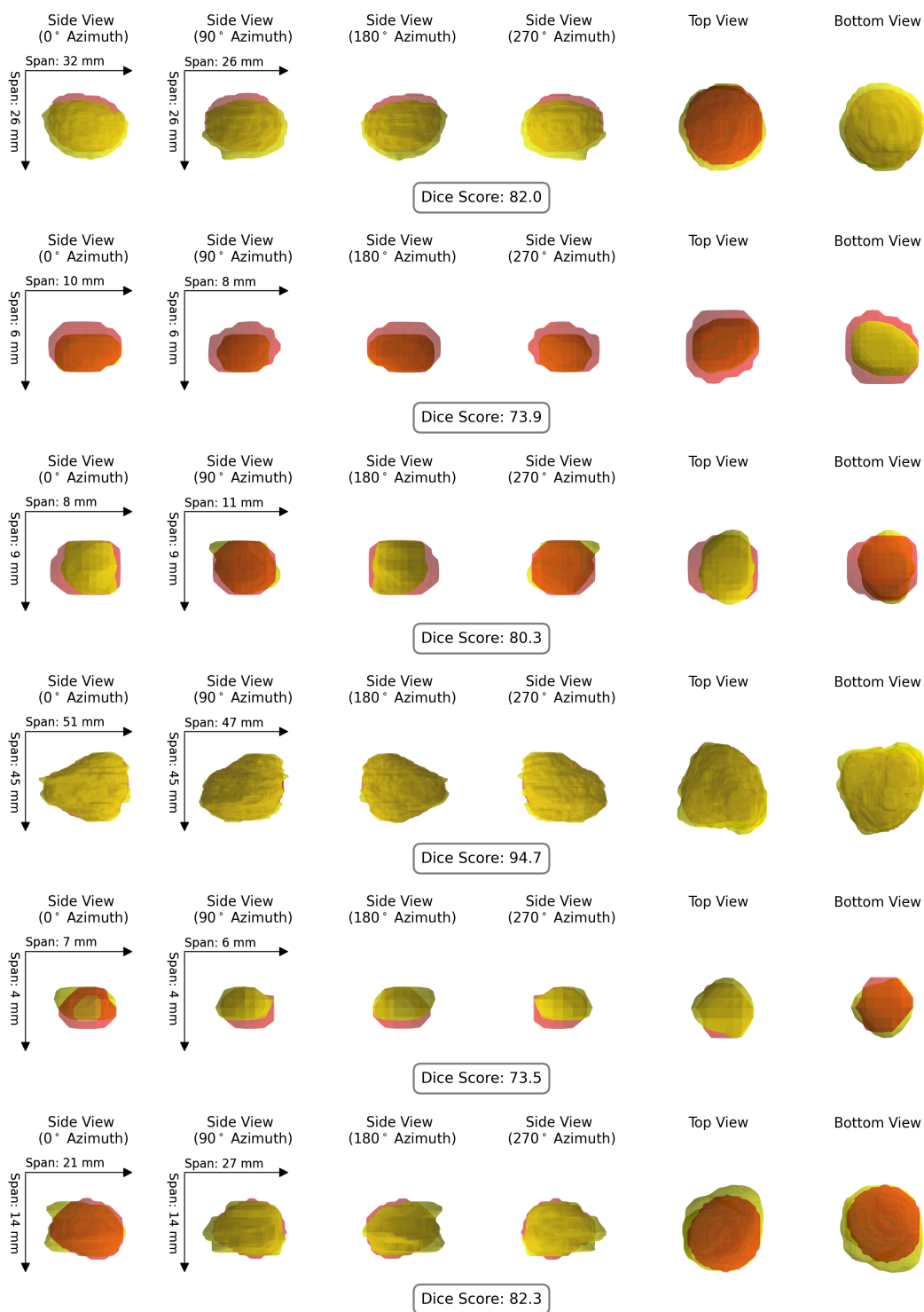
### **Extension to The BraTS Dataset**

On the BraTS dataset, we evaluated the ability of the proposed approach to improve the segmentation of brain tumors when tested on a different imaging modality and a different anatomical structure of interest. We can observe from Table 6.2 that the proposed approach outperforms the current state-of-the-art model and further reduces segmentation variability across subjects. The performance improvement on the BraTS dataset is not as significant when compared to the liver lesion dataset due to the proximity in performance between models trained on each of the individual contrast images and the model trained using all four contrast images together as a 4-channel input. Nevertheless, The proposed framework reduced the relative Dice score variability across subjects, which is represented by the standard deviation, by 5.2% when compared to the benchmark model.

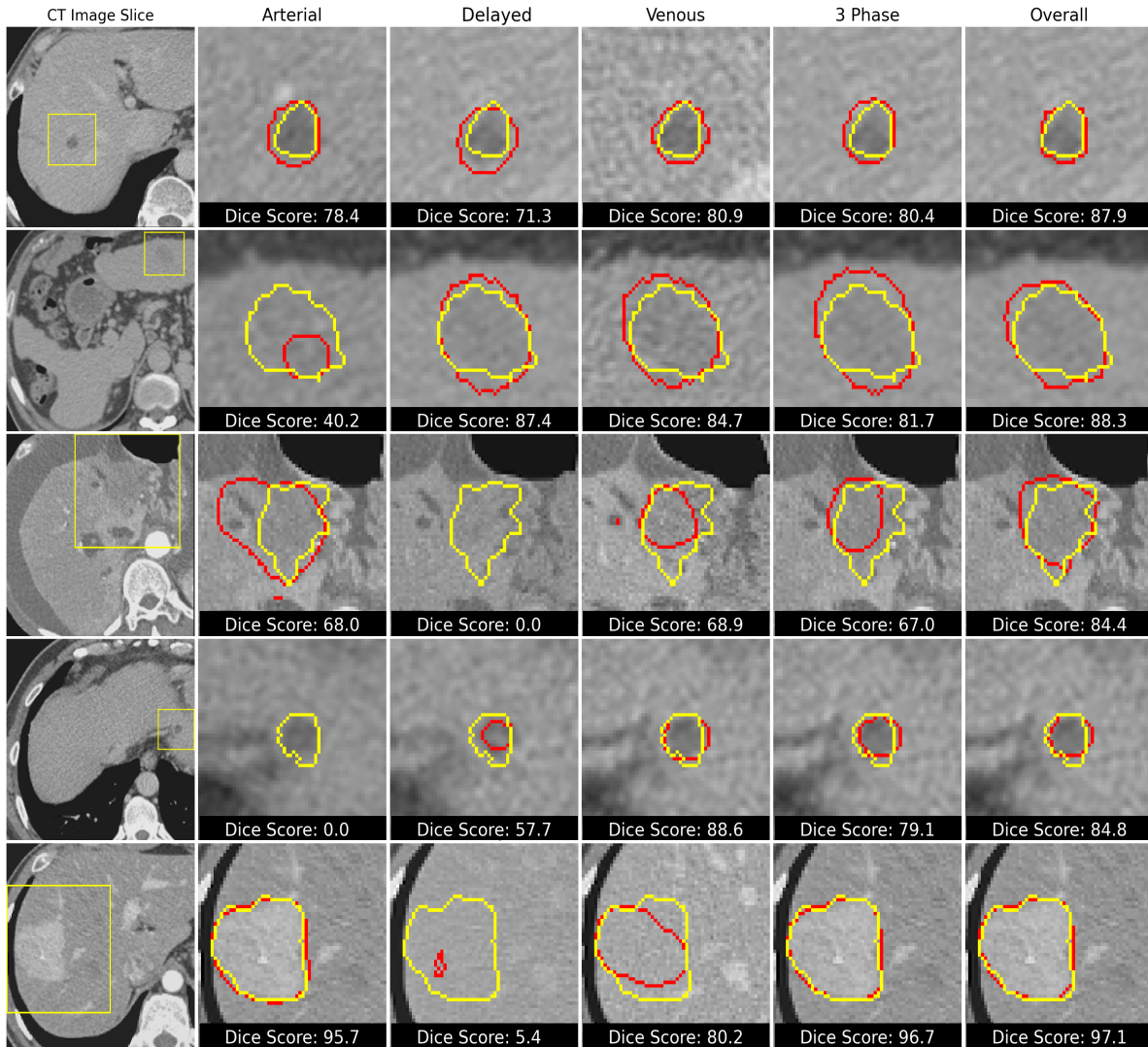




**Figure 6.6:** Qualitative comparison of the proposed approach to the other four baseline models. The figure presents a side-by-side assessment of the different approaches' segmentation performance on lesions of different sizes, shapes and intensity characteristics. The first column shows the original CT image slices cropped to the liver region with the region of interest highlighted in a yellow bounding box. The subsequent columns show the cropped bounding box region with the outline of segmentation results from the different approaches (in red) and the ground truth segmentation outline (in yellow). The Dice score for each of the predictions outlined is listed below each image.



**Figure 6.7:** The 3D surface of predicted segmentation masks using our proposed approach (red) and ground truth masks (yellow) of lesions of different sizes and shapes. Each row visualizes the masks from 6 different perspectives. The lesion span in all three dimensions is outlined in the first two visualizations. The Dice score for each of the predictions is listed below each row.



**Figure 6.8:** Qualitative demonstration of the proposed framework ability to improve the segmentation outcome versus 3-Phase segmentation models by incorporating learnings from each of the phases individually. The figure presents a side-by-side comparison of the overall framework to the different models trained on each of the phases individually (arterial, delayed, and venous) as well as the 3-Phase model. The first column shows the original CT image slices cropped to the liver region with the region of interest highlighted in a yellow bounding box. The subsequent columns show the cropped bounding box region with the outline of segmentation predictions (in red) and the ground truth segmentation outline (in yellow). For each of the models trained on individual phases, the ground truth and predictions are shown on a cropped box region from the corresponding phase slice. The Dice score for each of the predictions outlined is listed below each image.

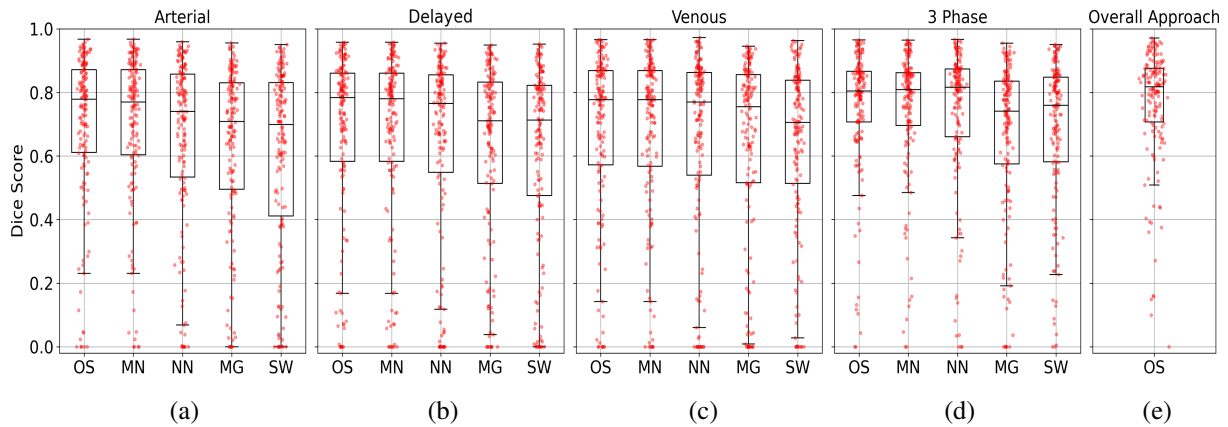
**Table 6.2:** The proposed framework brain tumor segmentation performance on the BraTS dataset by subject. Similar to Table 6.1, Section I outlines the results of just the segmentation model while Section II outlines the performance of the overall proposed framework. All metrics are in the range 0 to 100. Values in parentheses represent the standard deviation across subjects. Best results are boldfaced.

Phase	Model	Dice	IoU	Recall	Precision
Section I: Segmentation Model Results					
T1	MedNext	86.6 ( $\pm$ 8.0)	77.1 ( $\pm$ 11.2)	86.4 ( $\pm$ 10.4)	88.0 ( $\pm$ 9.5)
	Ours	86.7 ( $\pm$ 7.9)	77.1 ( $\pm$ 11.2)	86.6 ( $\pm$ 10.4)	87.9 ( $\pm$ 9.4)
T2	MedNext	88.7 ( $\pm$ 7.8)	80.4 ( $\pm$ 10.9)	87.8 ( $\pm$ 9.7)	90.6 ( $\pm$ 9.2)
	Ours	88.7 ( $\pm$ 7.5)	80.5 ( $\pm$ 10.8)	88.1 ( $\pm$ 9.5)	90.4 ( $\pm$ 9.3)
T1Gd	MedNext	86.5 ( $\pm$ 7.8)	77.0 ( $\pm$ 11.3)	85.7 ( $\pm$ 10.3)	88.6 ( $\pm$ 9.4)
	Ours	86.6 ( $\pm$ 7.9)	77.1 ( $\pm$ 11.5)	85.7 ( $\pm$ 10.5)	88.7 ( $\pm$ 9.2)
FLAIR	MedNext	89.6 ( $\pm$ 6.9)	81.9 ( $\pm$ 10.0)	87.8 ( $\pm$ 10.5)	92.7 ( $\pm$ 6.7)
	Ours	89.8 ( $\pm$ 6.6)	82.1 ( $\pm$ 10.0)	88.4 ( $\pm$ 10.0)	92.3 ( $\pm$ 6.8)
4-Phase	MedNext	90.8 ( $\pm$ 5.8)	83.7 ( $\pm$ 9.1)	89.6 ( $\pm$ 8.7)	<b>92.8</b> ( $\pm$ 6.8)
	Ours	90.9 ( $\pm$ 5.8)	83.8 ( $\pm$ 9.0)	90.0 ( $\pm$ 8.4)	92.7 ( $\pm$ 6.9)
Section II: Overall Framework Results					
Overall	Ours	<b>91.1</b> ( $\pm$ 5.5)	<b>84.0</b> ( $\pm$ 8.7)	<b>90.3</b> ( $\pm$ 7.9)	92.7 ( $\pm$ 6.8)

### Performance Variability Across Subjects

Performance consistency when it comes to lesion segmentation and detection is crucial to maintain similar levels of patient care and reduce bias. We designed our framework to improve the recovery and segmentation of lesions by incorporating learning from individual phase models. This also reduces performance variability across subjects as shown in Table 6.1. The proposed framework reduced the relative Dice score variability across subjects, which is represented by the standard deviation, by 8% when compared to the best performing benchmark model (18.5% versus 20.1%).

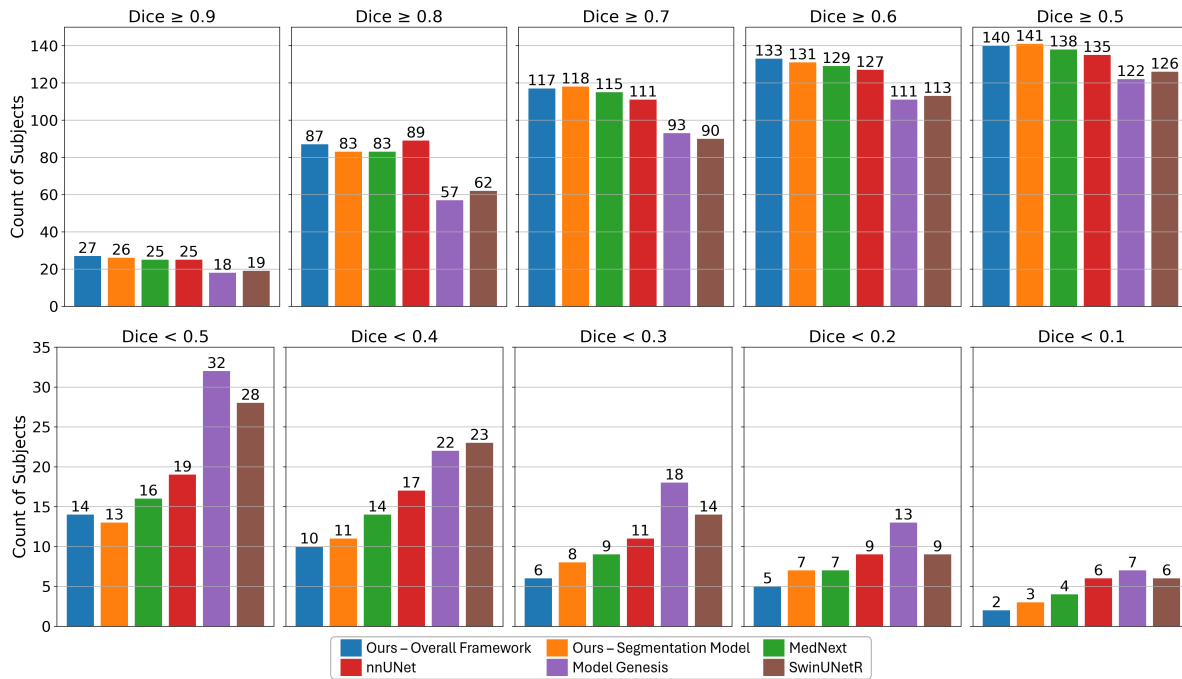
This consistency is also demonstrated in Fig. 6.9 and Fig. 6.10. In Fig. 6.9, the distribution of the Dice score across subjects using box plots is shown. The interquartile region of the proposed model and framework is consistently lower demonstrating a reduction in variability and increased consistency. Furthermore, the proposed approach reduces the number of subjects with low quality



**Figure 6.9:** Boxplots of the proposed segmentation model performance by subject (in terms of Dice score) compared to the other four baseline models when trained on the arterial (a), delayed (b) and venous (c) phases individually, and when trained using the three phases as 3-channel inputs (d). The performance of the overall framework is in (e). For each model as well as the overall framework, the Dice score of each sample from the test set is overlaid on top of the boxplot as a red circle at the vertical location of the corresponding Dice score. Model names keys: OS = Ours, MN = MedNext, NN = nnUNet, MG = Model Genesis, and SW = SwinUNetR.

segmentation by 50%, 28.5%, 33.3%, 28.5%, and 12.5% for low quality segmentation with Dice score thresholds of 0.1, 0.2, 0.3, 0.4, and 0.5 respectively as shown in Fig. 6.10. On the other hand, the proposed approach increases the number of subjects with high quality segmentation by 1.4%, 3.1%, and 1.7% for high quality segmentation with Dice score thresholds of 0.5, 0.6, and 0.7.

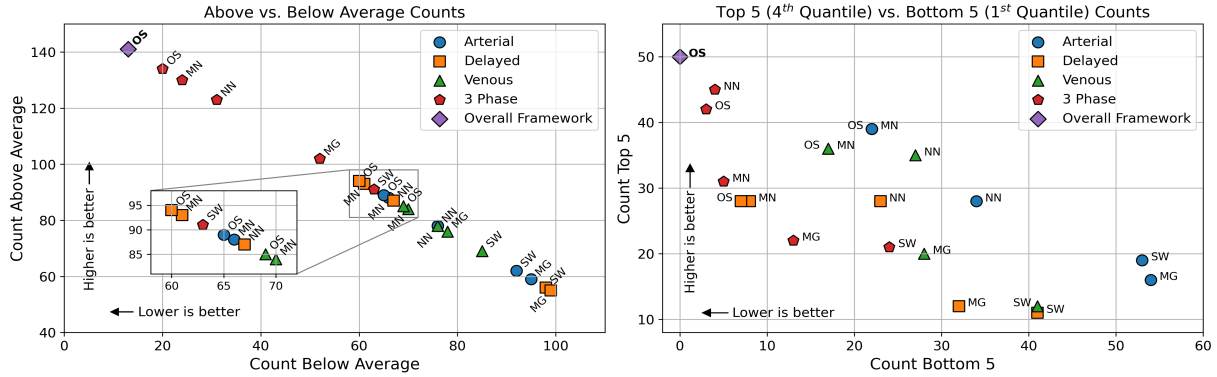
To evaluate the proposed approach’s performance from each subject’s perspective, we compared its segmentation results to the other models on a subject-by-subject basis. Specifically, we counted the number of instances where the proposed approach performed better than the average performance of models for each subject versus the number of instances it performed worse than the average as shown in Fig. 6.11. In 141 subjects out of a total of 154, the proposed approach performed better (highest among all models), and in only 13 subjects it performed worse (lowest among all models). In Fig. 6.11, we also show the relative performance by subject when the top and bottom quantiles are considered. The proposed approach was in the top quantile more than any other model and was never in the bottom quantile (lowest among all models).



**Figure 6.10:** Distribution of segmentation performance by subject across multiple Dice score thresholds. The upper panel (a higher count is better) shows the performance at higher Dice score thresholds ( $\geq 0.5$ ), indicating the number of subjects for each model achieving a better segmentation score than the threshold at the top of the graph. The lower panel (a lower count is better) shows the performance at lower Dice coefficient thresholds ( $< 0.5$ ), indicating the number of subjects for each model achieving a lower segmentation score than the threshold at the top of the graph. For the upper panel, higher counts are better while for the lower panel, lower counts are better. This provides insight into the reliability and consistency of each model in clinical settings. Higher bars at the upper panel suggest superior segmentation capabilities, while lower bars at the lower panel suggest reduced failure rates.

## Detection and Localization By Lesion

To evaluate the proposed approach lesion detection performance, we computed the precision, recall, and F1 score at different Dice score thresholds (0.1 to 0.9) and calculated the average precision, recall, and F1 scores across all the thresholds as shown in Fig. 6.12 (a). Overall, the average F1 score by lesion of the proposed approach is 62.8% versus 61.3% and 62.0% for the MedNext and nnUNet models. In terms of localization, we used the average Euclidean distance between the center of ground truth and predicted segmentation by lesion. The difference in localization performance of our proposed approach with respect to the best performing benchmark



**Figure 6.11:** Relative segmentation performance of models for each subject. The figure outlines the number of instances a model was above versus below average as well as in the top 5 versus bottom 5 when compared to the other models for each subject. Model names keys: OS = Ours, MN = MedNext, NN = nnUNet, MG = Model Genesis, and SW = SwinUNetR.

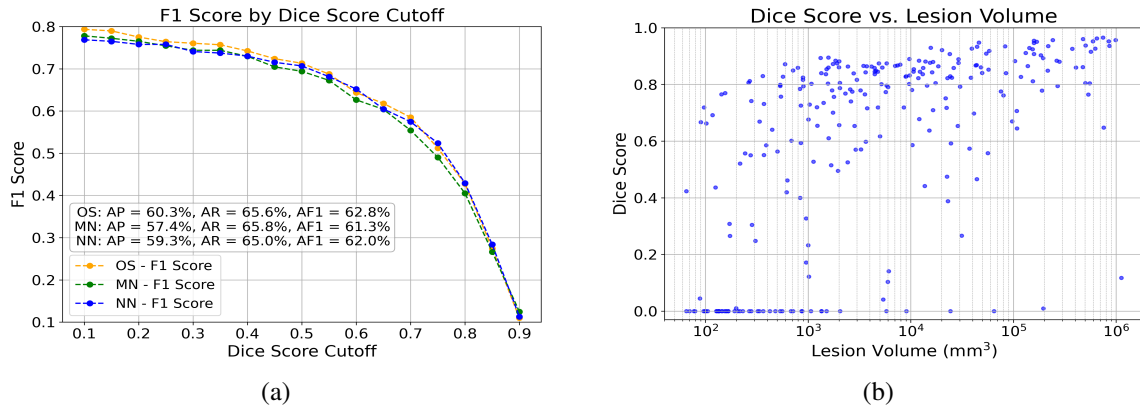
model was relatively small with an average localization error for detected lesions of 3.44 mm versus 3.50 mm and a standard deviation of 5.40 mm versus 5.45 mm.

### 6.4.4 Limitations and Future Prospective

The proposed framework is designed to improve the recovery and segmentation of lesions from multi-phase CT scans of the liver when compared to multi-channel segmentation models, which are the current state-of-the-art as shown in Table 6.1. Nevertheless, a major challenge we observed in our proposed framework, as well as other models, is recovering and segmenting small lesions from the image volume, which remains an open and challenging problem for lesion segmentation approaches, as shown in Fig. 6.12 (b). To address this challenge, we propose a lesion mask selection approach from the predictions of multi-specialized models’ in the next chapter, which improves the recovery and detection of lesions, including small ones.

## 6.5 Conclusion

We proposed a multi-stage segmentation framework for liver lesions in multi-phase CT scans. The proposed framework improves feature extraction from each of the phases when



**Figure 6.12:** The lesion detection F1 score by Dice score cutoff (a). The average precision (AP), recall (AR), and F1 scores (AF1) across the Dice score cutoff range of 0.1 to 0.9 (a). Model names keys: OS = Ours, MN = MedNext, and NN = nnUNet. The Dice score by lesion versus lesion volume (b).

compared to the current state-of-the-art segmentation models. This enables the framework to improve the overall segmentation performance by 1.6% while reducing performance variability across subjects by 8%. As the backbone segmentation model of the framework, we proposed a UNet-like architecture that uses the ConvNext convolutional block in both the encoder and the decoder. In the skip connections, we proposed the Coarse+Fine Feature Fusion & Attention Module (C+F FFA) to enhance feature fusion and attention between the encoder and decoder, which improved segmentation accuracy by 0.4% and reduced performance variability across subjects by 1.5% when compared to the current-state-of-the-art model.

Chapter 6 is, in part, based on the materials as they appear in “Multi-target and multi-stage liver lesion segmentation and detection in multi-phase computed tomography scans”, Abdullah F. Al-Battal; Soan T. M. Duong; Van Ha Tang; Quang Duc Tran; Steven Q. H. Truong; Chien Phan; Truong Q. Nguyen; Cheolhong An, submitted to the Medical Image Analysis journal, 2024. The dissertation author was the primary investigator and author of this paper.



# Chapter 7

## Enhancing Lesion Detection and Segmentation Via Lesion Mask Selection from Multi-Specialized Model Predictions in CT Scans

### 7.1 Introduction

Automated segmentation and detection of tumors in CT scans have a significant potential to assist clinicians with cancer diagnosis and treatment planning. However, current approaches, including state-of-the-art deep learning ones, still face many challenges. Many tumors are not detected by these approaches when tested on public datasets for tumor detection and segmentation such the Kidney Tumor Segmentation Challenge (KiTS) [148, 178, 179] and the Liver Tumor Segmentation Challenge (LiTS) [119]. False negative rates by lesion as high as 50% are commonly observed, and this rate is even higher for smaller lesions as they exhibit a high degree of variability (heterogeneity) among themselves. Additionally, in numerous instances, these lesions share

similarities (homogeneity) in intensity, size, and shape with other anatomical structures as well as blurriness and blending with surrounding tissue. This task is even challenging for experienced radiologists where in the liver, accurate detection of tumors can be as low as 72% for 10-20 mm lesions, and 16% for lesions smaller than 10 mm [171, 172]. In the kidney, a similar detection accuracy of 79% has been reported in the literature for lesions in general [180], where small lesions are often more challenging to detect [181, 182].

To improve the detection and segmentation accuracy of lesions in CT scans, we propose a lesion mask selection approach that uses the predictions of two models to select the best possible mask for lesions. Both models are based on the UNet architecture and use the ConvNext convolutional block in both the encoder and decoder. The first model is trained on lesion segmentation regardless of size, while the second is designed and fine-tuned to segment and detect small lesions. Once the segmentation mask is predicted from both models, we extract intensity-based features from within the lesion, contrast them with features from surrounding tissue, and select the mask that maximizes features' separation between the two. We test our approach on three different datasets for lesion segmentation in CT scans of the kidney and liver. The first dataset is a clinical three-phase CT dataset. The dataset contains scans from 354 subjects annotated with liver lesion segmentation labels. Each subject has 3 contrast-enhanced scans at three different phases, which are the arterial, delayed, and venous phases. This is the same dataset we used to test our segmentation approach in Chapter 6. We also test the proposed approach on two publicly available datasets for liver and kidney lesions, which are the Liver Tumor Segmentation Challenge (LiTS) [119] and the Kidney Tumor Segmentation Challenge (KiTS) [148, 178, 179]. The LiTS dataset contains CT scans of the liver from 131 subjects with segmentation annotations of both the liver and lesions. For the KiTS dataset, we use the most recent version of the dataset (KiTS<sub>23</sub>), which contains 489 scans annotated with masks for both the kidneys and lesions. We compare the proposed approach to the current state-of-the-art segmentation models, which are the MedNext [117], nnUnet [48], SwinUNetR [116], and Model

Genesis [120] models as well as soft voting ensembling (SVE) and hard voting ensembling (HVE). The proposed approach improved the segmentation and detection performance in a global, by-subject, and by-lesion manner, improving the overall segmentation accuracy as well as the detection and segmentation of small lesions. Overall, the major contributions of the proposed approach are:

1. A selection approach that combines the predictions of two models and selects the best prediction based on lesion feature separation with respect to surrounding tissue; improving the overall and small lesion segmentation and detection.
2. A segmentation model that is specifically designed to improve small lesion segmentation by incorporating an increased number of high resolution features at different model stages.
3. A feature extraction and comparison approach that provides a measure of separation between intensity-based features of lesions and surrounding tissue.
4. A high resolution feature fusion (HR-FF) module that integrate features from high resolution stages into the skip connections of deeper stages of the model designed for small lesion segmentation.

## 7.2 Background

Segmentation models that performs best on liver and kidney lesion segmentation such as the MedNext [117], nnUNet [48], and Model Genesis [120] models incorporate the spatially contracting design paradigm of the UNet architecture. These models use the spatially contracting encoder and spatially expanding decoder architecture with skip connections. This allows the architecture to encode spatial information in an image into a high dimensional (feature-wise) latent space then contextualize these features spatially to generate a mask map where each pixel, or voxel for 3D images, is represented individually based on the class it belongs to [69]. Skip connections

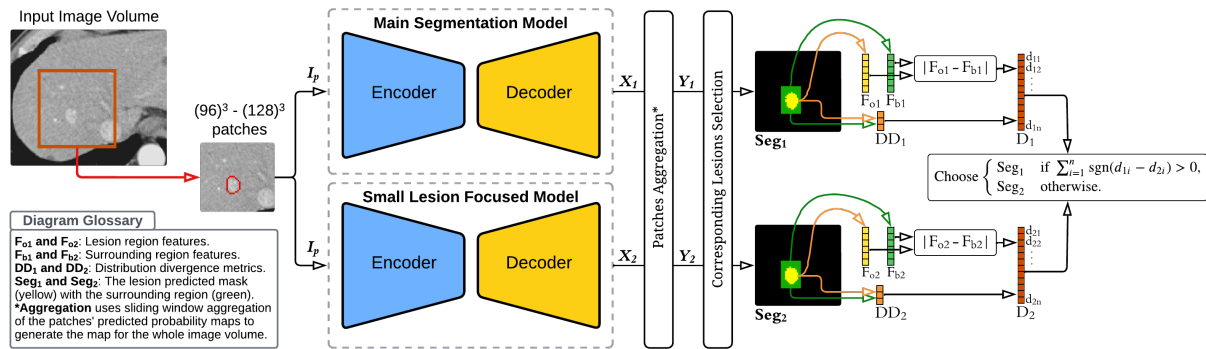
transfer features from the encoder layers to the decoder at each depth level, which improves the overall performance of these architectures significantly [111]. These skip-connections transfer the feature map at each level and either concatenate it with or add it to the feature map in the decoder, which defines the UNet architecture [111]. Several modifications have been proposed to the UNet architecture that improved its overall segmentation performance. The most significant ones focused primarily on the design of convolutional blocks within the encoder and decoder [183, 113], skip connections [114, 71] or self-configurability of the network depth and width [48]. Other improvements targeted network training using deep supervision, [114, 184], and were able to improve the segmentation performance of UNet on medical images.

Although this spatially contracting design paradigm has demonstrated superior performance in segmentation tasks [111, 48], the reduced spatial dimensions in the later stages of the encoder and early stages of the decoder reduce the ability of the model to achieve the same accuracy for small lesions as it does for larger ones. On the other hand, this spatial dimension reduction is important for extending the receptive field of the model to improve the learning of long-range dependencies [185]. This spatial reduction also allows the model to incorporate a higher number of features in deeper layers without exceeding the available memory capacity on a GPU. Therefore, we propose an approach that trains two models to counter and balance these limitations. The first model uses a complete contracting encoder and expanding decoder for general lesion segmentation. This model works best for large lesions and is, on average, worse for small lesions than the second model. The second model removes the spatial reduction step in the last stage of the model and incorporates high resolution features in each of the skip connections that occur after down-sampling; promoting an increased incorporation of high resolution features. We then use the predictions from both models to select the best possible mask for each predicted lesion. For overlapping lesions, we contrastively compare intensity features between the lesion and surrounding tissue in both predictions. We then select the prediction that maximizes the difference in intensity features between the lesion and its surrounding tissue. The proposed

approach improves the overall lesion segmentation and detection performance as well as enhances the discovery and segmentation of small lesions within CT scans.

### **7.3 Proposed Method**

Effective identification and segmentation of small lesions is challenging for current segmentation models. Modifying the model architecture or training parameters and approaches such as loss function weighting to enhance small lesion segmentation and detection reduces the overall performance of the model on larger lesions. Therefore, our approach, which is composed of two models as shown in Fig. 7.1, aims to mitigate this issue by combining the predictions of both models by comparing lesion intensity features with surrounding tissue in both predictions and selecting the prediction that maximizes the difference between the two sets of features. In our approach, both models are based on the UNet architecture with a spatially contracting encoder and a spatially expanding decoder. The first model uses a complete contracting encoder and expanding decoder with four down-sampling and up-sampling stages for general lesion segmentation. This model works best for large lesions and is on average worse for small lesions than the second model. The second model removes the spatial reduction step in the last stage of the model and incorporates high resolution features in each of the skip connections that occur after down-sampling to incorporate more high resolution features at each stage. Both models' structures are outlined in Fig. 7.2 (a) and (b), respectively, and labeled as the "Main Segmentation Model" and "Small Lesion Focused Model 1", respectively. In this section, we explain our approach in detail, starting with the augmentation and pre-processing approaches to promote generalization and robustness.



**Figure 7.1:** The proposed prediction selection approach. The 3D image patch is fed to both models where each generates a segmentation prediction map. Corresponding lesion pairs are selected from  $Y_1$  and  $Y_2$  based on a Dice score overlap of 0.5. For corresponding lesions in each model prediction, intensity-based features, which are described in Section 7.3.4, are extracted from the region of the lesions ( $F_o$ ) and surrounding tissue ( $F_b$ ) in addition to intensity distribution divergence metrics ( $DD$ ). The prediction with the highest count of largest differences is selected. The detailed structure of both models is shown in Fig. 7.2.

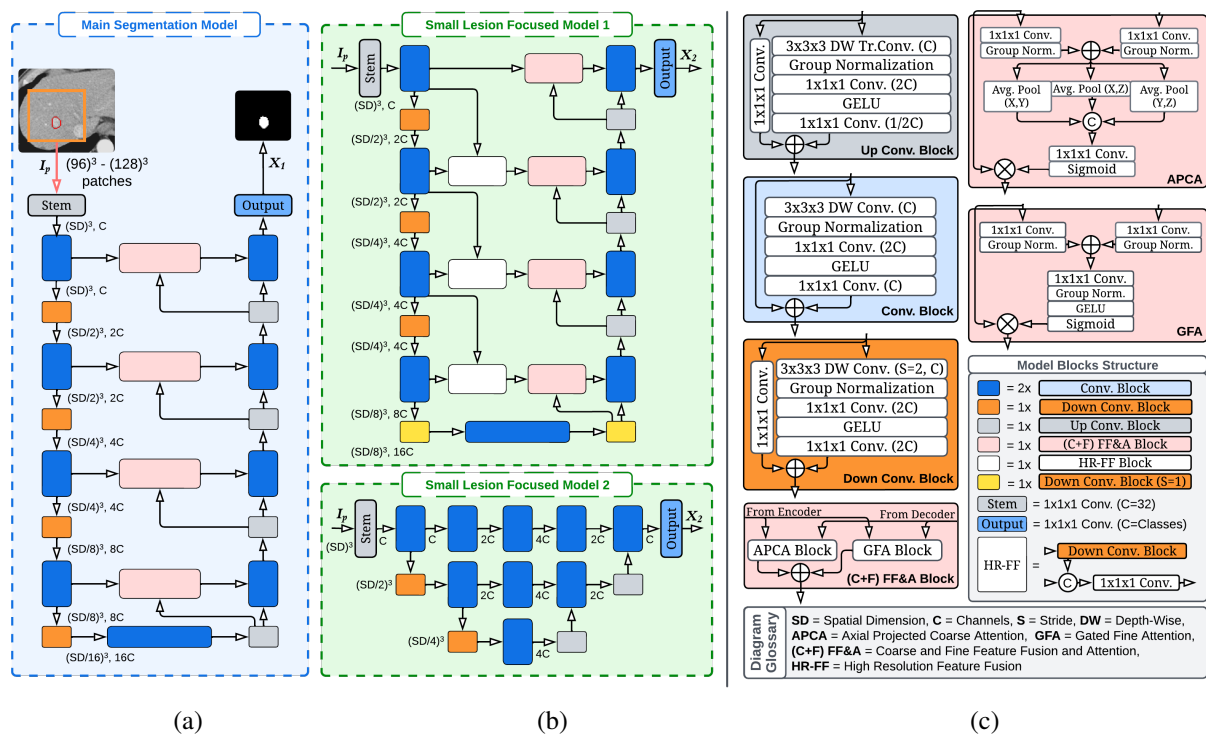
### 7.3.1 Pre-Processing and Augmentation

Proper pre-processing and data augmentation is essential while training neural networks to promote robustness and allow the network to generalize well beyond the training set. Our proposed models use a set of randomized augmentation operations to introduce variabilities into the training dataset to simulate the variability present in actual scans during the model's deployment. The augmentation methods are categorized into three main categories: a) intensity-based to account for different imaging devices, b) geometrically rigid, and c) elastic transformations to account for variations in anatomical structures' shape, size and elasticity. Prior to augmentation in training and inference in testing, we pre-process the CT 3D images and prepare them for our segmentation framework by clipping and normalizing their intensity values. Then, we randomly select a patch of size  $128 \times 128 \times 128$  voxels for the clinical liver lesion dataset and  $96 \times 96 \times 96$  for the LiTS and KiTS datasets, which are the dimensions of the model input image. The patch size for the clinical dataset is larger in scale due to the presence of large lesions in the dataset that exceed 96 mm in diameter. All the CT volumes are resampled to an isotropic spatial spacing of  $1 \times 1 \times 1$  mm.

For intensity-based augmentation, we use a random set of options in each training iteration to induce variations in the scans' color (intensity) space. These transformations change brightness, contrast, and noise levels to improve the model's ability to account for different imaging devices and image reconstruction settings. To account for different positional and spatial variabilities, we incorporated rigid geometrical transformations in the form of rotations, translations, flipping, cropping, and resizing as part of the augmentation pipeline. We also incorporate affine transformations that include shearing and scaling as well as elastic deformations to simulate soft tissue stretching and compression and the natural variations in the shape and position of internal organs. These variations can be caused by factors such as breathing, differences in patient body size, or even the presence of tumors and other abnormalities.

### 7.3.2 The Main Segmentation Model

The first stage of this model is a stem that expands the input patch feature space to 32 channels. The model uses four stages in the encoder and four stages in the decoder with a bridge bottleneck after the last encoder stage to connect the encoder with the decoder. Each stage contains two 3D ConvNext blocks and a 3D down-sampling ConvNext block, which expands the number of features and reduces the spatial dimension by 2. The bridge bottleneck contains only two 3D ConvNext blocks. In the decoder, each stage contains two 3D ConvNext blocks and a 3D up-sampling ConvNext block that contracts the number of features by 2 and expands the spatial dimension by 2. The structure of the model, ConvNext and down-sampling ConvNext blocks are outlined in Fig. 7.2 (a) and (c). ConvNeXt blocks improve upon prior convolutional blocks design by offering greater efficiency through depth-wise separable convolutions and the inverted bottleneck design. To train the model, we incorporate deep supervision where the output of each convolutional block in the decoder at each depth level ( $N$ ) is fed to an output linear projection layer that contracts the channel space from  $2^{(N-1)}C$  to 2, which represents the number of classes (background versus lesion).



**Figure 7.2:** The structure of the models used in the overall approach outlined in Fig. 7.1. In (a), the structure of the main segmentation model is outlined. In (b), the structures of the small lesion focused models. The small lesion focused model 1 is the main small lesion focused model in our approach and experiments in Section 7.4. The small lesions focused model 2 is a proof of concept model that we tested on the clinical dataset to evaluate the ability of its architecture with primarily high resolution features to detect and segment small lesions. The structure of the modules and blocks for all models is in (c) as well as the diagram glossary of abbreviations. The diagrams in (a) and (b) are annotated with the spatial dimension and the number of feature channels at the output of each stage.

In the skip connection between the encoder and decoder, we use a Coarse+Fine Feature Fusion & Attention Module (C+F FFA) that we proposed in Chapter 6. This module integrates two mechanisms for feature fusion and attention: Axial Projected Coarse Attention (APCA) and Gated Fine Attention (GFA). The APCA module refines feature maps by highlighting important information spatially using mixtures of the encoder and decoder features ( $X_f$  and  $X_g$ ) to modulate  $X_f$ . It uses  $1 \times 1 \times 1$  convolutions and group normalization to create a compact representation in the feature space, which are mixed and then projected across each of the three spatial dimensions using adaptive average pooling to create a coarse and smoothed attention map



that is less susceptible to noise due to axial based projection and averaging. The GFA module, in parallel, processes the same feature sets ( $X_f$  and  $X_g$ ) with  $1 \times 1 \times 1$  convolutions, combining them without axial projection to maintain spatially fine details of representations within the feature map. A subsequent convolution layer produces a spatial attention map, modulating ( $X_f$ ). The detailed structure of both modules is outlined in Fig. 7.2 (c).

### 7.3.3 The Small Lesion Focused Model

We designed two models to function as agents to improve small lesion segmentation. The first model labeled as "Small Lesion Segmentation Model 1" in Fig. 7.2 (b) is the main small lesions focused model in our proposed approach. The second model is a proof of concept model that we tested to evaluate the validity of models that do not follow the spatially contracting encoder and expanding decoder design paradigm. The second model in initial preliminary tests on the clinical dataset demonstrated promising performance, approaching the performance of the first model with a limited number of parameters. However, due to the use of multiple stages at full spatial resolution, scaling in the feature space beyond 128 feature channels is currently not possible due to GPU memory constraints.

#### Model 1

The first stage of this model is a stem that expands the input patch feature space to 32 channels. The model uses four stages in the encoder and four stages in the decoder with a bridge bottleneck after the last encoder stage to connect the encoder with the decoder. Stages 1, 2, and 3 in the encoder contain two 3D ConvNext blocks and a 3D down-sampling ConvNext block, which expands the number of features by 2 and reduces the spatial dimension by 2. Stage 4, on the other hand, uses a 3D down-sampling ConvNext with a stride of 1 that maintains the spatial resolution while expanding the number of features by 2. This design choice is to maintain a higher resolution in the bridge, which contains the largest number of feature channels. The bridge

bottleneck contains only two 3D ConvNext blocks. In the decoder, each stage contains two 3D ConvNext blocks and a 3D up-sampling ConvNext block that contracts the number of features by 2 and expands the spatial dimension by 2 except for the stage after the bridge, which does not expand the spatial dimension. The structure of the model, ConvNext and down-sampling ConvNext blocks are outlined in Fig. 7.2 (b) and (c). In the skip connection between the encoder and decoder, we use the Coarse+Fine Feature Fusion & Attention Module (C+F FFA). We also use a High Resolution Feature Fusion (HR-FF) module that incorporates features from the encoder’s previous stage, which is at a higher resolution, into the skip connection of stages 2, 3, and 4. The HR-FF module is outlined in Fig. 7.2 (c). To train this model, we also incorporate deep supervision as we did for the main segmentation model.

## Model 2

In this model, we use 5 stages at full spatial resolution (level 1), 3 stages at half the resolution (level 2), and 1 stage at a quarter of the resolution (level 3). Before the features are fed to the stages in level 2 and 3, we use the 3D down-sampling ConvNext block. At the end of each of levels 2 and 3, the features are passed through a 3D up-sampling ConvNext block and added to the features from the second to last stage in the previous level before being passed to the last stage in the previous level as shown in Fig. 7.2 (b). To train this model, we also incorporate deep supervision where the output of each level ( $N$ ) is fed to an output linear projection layer that contracts the channel space from  $2^{(N-1)}C$  to 2.

### 7.3.4 The Intensity-Based Features Used for Prediction Comparison

To compare the predictions of the two models, we use a set of intensity-based features that are computed from within the lesion region and surrounding tissue for each of the two predicted masks. These features are the mean, standard deviation, median, 5<sup>th</sup> and 95<sup>th</sup> percentiles, skewness, kurtosis, and Shannon entropy of the distribution of intensity values. For the Shannon

entropy, we use the following definition [186]:

$$H_R = - \sum_i p(i) \log_2 p(i), \quad (7.1)$$

where  $R$  represents the region of interest (either inside the lesion or surrounding tissue),  $i$  the intensity values, and  $p(i)$  the intensity value probability, which is computed from the normalized histogram distribution. In addition to these features, we use three symmetric distribution divergence metrics to evaluate the dissimilarity between the intensity values distribution within the lesion and surrounding tissue. The first metric is the absolute Standardized Mean Difference (ASMD), which is a symmetric modified version of the Cohen's coefficient [187]:

$$\text{ASMD} = \frac{|\mu_L - \mu_{ST}|}{\sqrt{\frac{\sigma_L^2 + \sigma_{ST}^2}{2}}}, \quad (7.2)$$

where  $\mu$  and  $\sigma$  refer to the mean and standard deviation of intensity values, respectively. The  $L$  and  $ST$  subscripts in (7.2) refer to the lesion (L) and surrounding tissue (ST). In addition to the ASMD, which is a parametric measure, we use the Kolmogorov-Smirnov statistic (KS) [188, 189] and Jensen-Shannon divergence (JSD) [190]; both are non-parametric measures that evaluate the divergence of intensity values distribution. KS, which quantifies the maximum discrepancy between the cumulative distribution functions ( $F$ ) is defined as:

$$\text{KS} = \sup_i |F_L(i) - F_{ST}(i)|. \quad (7.3)$$

The JSD, which measures the distance between two probability distribution functions (PDF) is defined as:

$$\text{JSD}(L, ST) = \frac{1}{2} D_{KL}(p_L || p_M) + \frac{1}{2} D_{KL}(p_{ST} || p_M), \quad (7.4)$$

where  $D_{KL}(p||q)$  is the Kullback-Leibler divergence between two distributions  $p$  and  $q$ , and is

defined as:

$$D_{KL}(p||q) = \sum_i p(i) \log \frac{p(i)}{q(i)}, \quad (7.5)$$

and  $p_M = \frac{1}{2}(p_L + p_{ST})$  is the average of the two PDFs.

### 7.3.5 Segmentation Mask Prediction

Once the features outlined in Section 7.3.4 are extracted from the lesion region and surrounding tissue, we follow the algorithm shown in Fig. 7.1 and outlined in details in Algorithm 7.1 to select the prediction mask that maximizes intensity features' separation between the lesion region and surrounding tissue. From the predicted mask of both models ( $Seg_1$  and  $Seg_2$ ) we identify corresponding lesion pairs based on a Dice score overlap threshold of 0.5. For each of the corresponding pairs, we compare the eight intensity-based features ( $|F_{o1} - F_{b1}|$  and  $|F_{o2} - F_{b2}|$ ), and the three distribution divergence metrics ( $DD_1$  and  $DD_2$ ) from within the lesion region and surrounding tissue for both predictions as well as the average of the two; selecting the prediction with the majority of greater separations across metrics. In regions where there are no correspondences, we use the average of the two predicted probability maps ( $Y_1$  from the main segmentation model and  $Y_2$  from the small lesion focused model).

---

**Algorithm 7.1** Lesion Segmentation Map Selection

---

1: **input:** Image Volume (I)  
2: **output:** Final Segmentation Mask (Seg<sub>F</sub>)  
3: **procedure** AFTER GENERATING THE PROBABILITY MAP AND SEGMENTATION MASK FROM THE MAIN SEGMENTATION MODEL ( $Y_1$  AND SEG<sub>1</sub>) AND SMALL LESION FOCUSED MODEL ( $Y_2$  AND SEG<sub>2</sub>).  
4: Initiate Seg<sub>F</sub>: Seg<sub>F</sub> = [0]<sub>W×H×D</sub>  
5: Identify corresponding lesion pairs from Seg<sub>1</sub> and Seg<sub>2</sub> based on Dice score overlap  
6:  $S_{PL} \leftarrow$  set of lesion pairs.  
7: **for** each lesion pair in  $S_{PL}$  **do**  
8: Create a bounding box (BBox) with a margin around both lesions  
9:  $F_{o1} \leftarrow$  Seg<sub>1</sub> lesion intensity features:  $F_{o1} \in \mathbb{R}^8$   
10:  $F_{o2} \leftarrow$  Seg<sub>2</sub> lesion intensity features:  $F_{o2} \in \mathbb{R}^8$   
11:  $F_{b1} \leftarrow$  Seg<sub>1</sub> surrounding region features  $\in$  BBox:  $F_{b1} \in \mathbb{R}^8$   
12:  $F_{b2} \leftarrow$  Seg<sub>2</sub> surrounding region features  $\in$  BBox:  $F_{b2} \in \mathbb{R}^8$   
13:  $DD_1 \leftarrow$  Seg<sub>1</sub> distribution divergence metrics:  $DD_1 \in \mathbb{R}^3$   
14:  $DD_2 \leftarrow$  Seg<sub>2</sub> distribution divergence metrics:  $DD_2 \in \mathbb{R}^3$   
15:  $D_1 \leftarrow [|F_{o1} - F_{b1}|, DD_1] : D_1 = [d_{11}, d_{12}, \dots, d_{1N}] \in \mathbb{R}^{11}$   
16:  $D_2 \leftarrow [|F_{o2} - F_{b2}|, DD_2] : D_2 = [d_{21}, d_{22}, \dots, d_{2N}] \in \mathbb{R}^{11}$   
17: **choose**  
$$\text{Seg}_c = \begin{cases} \text{Lesion mask from Seg}_1 & \text{if } \sum_{i=1}^N \text{sgn}(d_{1i} - d_{2i}) > 0, \\ \text{Lesion mask from Seg}_2 & \text{otherwise.} \end{cases}$$
  
18: Seg<sub>F</sub> = Seg<sub>F</sub> + Seg<sub>c</sub>  
19: **end for**  
20:  $S_{IL} \leftarrow$  set of individual lesions from Seg<sub>1</sub> and Seg<sub>2</sub>  $\notin S_{PL}$ .  
21: Initiate  $Y_{avg}$ :  $Y_{avg} = [0]_{W \times H \times D}$   
22:  $Y_{avg} \leftarrow \frac{1}{2}(Y_1 + Y_2) \forall Y_{avg}(i, j, k) \in S_{IL} : i, j, k = 1, 2, \dots, W, H, D$   
23:  $Y_{avg}(i, j, k) \leftarrow 1 \forall Y_{avg}(i, j, k) \geq 0.5$   
24: Seg<sub>F</sub> = Seg<sub>F</sub> +  $Y_{avg}$   
25: **return** Final Segmentation Mask (Seg<sub>F</sub>)  
26: **end procedure**

---

### 7.3.6 Training And Loss Function

To train both models, we used a compound loss of two loss functions. The compound loss is composed of the weighted sum of the binary cross-entropy (BCE) loss and the Dice loss:

$$\mathcal{L}_{comp}(\hat{Y}, Y) = \alpha \mathcal{L}_{bce}(\hat{Y}, Y) + \beta \mathcal{L}_{D_c}(\hat{Y}, Y), \quad (7.6)$$

where  $\alpha$  and  $\beta$  are the coefficients that control the contribution of BCE loss ( $\mathcal{L}_{bce}$ ) and Dice loss ( $\mathcal{L}_{D_c}$ ), respectively.  $\hat{Y}$  and  $Y$  represent the predicted and target (ground truth) segmentation masks.

For each element (voxel) of the predicted mask with a value  $\hat{y}$  and ground truth value  $y$  at location  $(i, j, k)$ , where  $i = 1, 2, \dots, W$ ,  $j = 1, 2, \dots, H$ , and  $k = 1, 2, \dots, D$  for a given segmentation mask with width =  $W$ , height =  $H$ , and depth =  $D$ , the BCE loss function is defined as:

$$\mathcal{L}_{bce}(\hat{Y}, Y) = -\frac{1}{WHD} \sum_{i,j,k=1}^{W,H,D} \ell(\hat{y}_{i,j,k}, y_{i,j,k}), \quad (7.7)$$

where  $\ell(\hat{y}_{i,j,k}, y_{i,j,k})$  is the loss computed element-wise between the ground truth and predicted mask, and is defined as:

$$\ell(x, y) = w_1 y \log \sigma(x) + w_0 (1 - y) \log \sigma(x). \quad (7.8)$$

In (7.8),  $\sigma(x)$  is the sigmoid function that defines  $\hat{y}$  and is defined as  $\hat{y} = \sigma(x) = 1/(1 + \exp(-x))$ .  $\sigma(x)$  converts the predicted output of the model into a probability space with values between 0 and 1. The parameters  $w_1$  and  $w_0$  are weights assigned to each class. For the binary cross-entropy loss, they are both 1 while for the weighted binary cross-entropy loss they are adjusted to increase the loss function attention to one class versus the other. The Dice loss is based on the Dice coefficient between the predicted and ground truth mask. The Dice coefficient ( $D_c$ ) is defined as [100]:

$$D_c(\hat{Y}, Y) = \frac{2 \sum(\hat{Y} \odot Y)}{\sum \hat{y}_{i,j,k} + \sum y_{i,j,k}}, \quad (7.9)$$

where  $\odot$  represents the element-wise multiplication. The Dice loss can be defined to penalize lower  $D_c$  values, which yields a lower segmentation performance, as:

$$\mathcal{L}_{D_c}(X, Y) = -D_c(\hat{Y}, Y). \quad (7.10)$$

When training the Small Lesion Focused Model, we incorporated an adaptive weighting of the BCE loss function to ensure a recall value that is at least 5% higher than precision in order to recover small lesions effectively due to the large class imbalance, especially for small lesions.

This adaptive algorithm is an extension of the approach we proposed in Chapter 4, where we proposed an algorithm to balance the recall and precision adaptively during training without the need of multiple hyperparameter finetuning iterations for 2D medical image segmentation. The algorithm monitors the exponential moving average (EMA) of the recall and precision over epochs and adjusts the weights  $w_1$  and  $w_0$  in the BCE loss function after each epoch to either increase recall versus precision (increasing  $w_1$  and lowering  $w_0$ ), or vice versa. This process is outlined in detail in Algorithm 7.2. At the beginning of training, we initiate both weights with 1 and adjust them by 2% ( $\Delta$  of 0.02) at the end of each epoch when needed. Finally, we limit the maximum and minimum possible values of  $w_1$  and  $w_0$  to be between 0.1 and 5 to ensure stability during training.

---

**Algorithm 7.2** Adaptive BCE Loss Weights Control

---

```

1: procedure AT END OF TRAINING EPOCH
2:   if Recall EMA < (Precision EMA  $\times$  1.05) then
3:      $w_1 \leftarrow w_1 \times (1 + \Delta)$ 
4:      $w_0 \leftarrow w_0 \times (1 - \Delta)$ 
5:   else
6:      $w_1 \leftarrow w_1 \times (1 - \Delta)$ 
7:      $w_0 \leftarrow w_0 \times (1 + \Delta)$ 
8:   end if
9:    $w_1 \leftarrow \max(0.1, \min(5.0, w_1))$ 
10:   $w_0 \leftarrow \max(0.1, \min(5.0, w_0))$ 
11:  return  $w_0, w_1$ 
12: end procedure

```

---

## 7.4 Experiments and Results

### 7.4.1 Datasets

We evaluated the segmentation and detection performance of our approach on three datasets. The first dataset is a clinical three-phase CT dataset, which contains scans from 354 subjects annotated with liver lesion segmentation labels. Each subject has 3 contrast-enhanced scans at three different phases, which are the arterial, delayed, and venous phases. It is the same dataset we used to test our segmentation approach in Chapter 6. It was developed by researchers

and clinicians at VinBrain, JSC and the University Medical Center at Ho Chi Minh City. The dataset was annotated and rated by two radiologists who are experienced in liver oncology. The axial slice resolution of the dataset is  $512 \times 512$  pixels with physical spacing in the range 0.5 mm to 0.84 mm, and an average of 0.66 mm. The average slice thickness is 0.9 mm and ranges from 0.5 mm to 1 mm. The average number of lesions per scan is 2.2 with a maximum of 11 and a minimum of 1. Lesions in the scans cover a wide range of sizes with lesions as large as 129 mm and as small as 2.7 mm in diameter. Masks of these lesions and their boundaries overlaid on the largest axial slice are shown in Fig. 6.5 to demonstrate the variability in lesion sizes and shapes within the dataset. We also test the proposed approach on two publicly available CT datasets for liver and kidney lesions, which are the Liver tumor segmentation challenge (LiTS) [119] and the Kidney Tumor Segmentation Challenge (KiTS) [148, 178, 179]. The LiTS dataset contains CT scans of the liver from 131 subjects with segmentation annotations of both the liver and lesions. For the KiTS dataset, we use the most recent version of the dataset (KiTS<sub>23</sub>), which contains 489 scans annotated with masks for both the kidneys and lesions.

## 7.4.2 Experimental Setup and Data Preparation

For all three datasets, we trained both models in the proposed approach from randomly initiated weights. For the clinical dataset, the scans were split into 200 for training and 154 for testing while for the LiTS dataset the split was 100 for training and 31 for testing. For the KiTS dataset, it was 380 for training and 109 for testing. For the clinical dataset, the liver was segmented using a model trained on the LiTS dataset and scans with only the liver region were used for the training and testing of the proposed lesion segmentation approach. The same approach was followed for the LiTS and KiTS dataset where scans with only the liver and kidney region were used for the training and testing of the proposed approach. For all the datasets, we used a spatial resolution of  $1 \text{ mm}^3$ , and patches of size  $128 \times 128 \times 128$  voxels for clinical dataset and  $96 \times 96 \times 96$  voxels for the LiTS and KiTS datasets as outlined in Section 7.3.1.



The CT scans Hounsfield units were clipped to the range  $[-200, 200]$  before normalization. We compared the proposed approach segmentation performance to four 3D segmentation networks, which are the current leading models across different medical segmentation tasks. These models are the SwinUnetR [116], Model Genesis [120], nnUNet [48], and MedNext [117]. We also compared the proposed selection approach to soft voting ensembling (SVE) and hard voting ensembling (HVE). For hard voting ensembling, since it requires majority voting, we incorporate the predictions from the best performing segmentation model in addition to the predictions of the proposed two models. The models were trained for 800 to 1200 epochs depending on the learning rate that is suitable for the model, which ranged from  $1e^{-4}$  to  $1e^{-2}$ . All the models were trained using the AdamW [156] optimizer and the loss function defined in (7.6), except for the nnUNet and Model Genesis models, which are trained using the stochastic gradient descent optimizer as it is the recommended optimizer for both models. At each iteration within an epoch, we randomly select two patches from the 3D image using a weighted sampling approach; giving a 50% higher probability weight to selecting a patch containing a lesion.

### 7.4.3 Evaluation and Results

#### Evaluation Metrics

To evaluate the performance of our proposed architecture, we use multiple segmentation and detection metrics. For segmentation, we use the Dice score and the intersection over union (IoU) metrics, which are the benchmark metrics used to evaluate detection, and segmentation methods [91]. We defined the Dice score in (7.9), and the IoU score is defined as:

$$IoU(\hat{Y}, Y) = \frac{\sum(\hat{Y} \odot Y)}{\sum(\hat{Y} | Y)}, \quad (7.11)$$

where  $\odot$  is element-wise multiplication,  $|$  is the element-wise *or* logical operator,  $\hat{Y}$  is the predicted mask map, and  $Y$  is the ground truth mask. For the Dice and IoU scores, we evaluate

them by subject to highlight how the performance of each model differs across subjects. We also evaluate the recall ( $TP/(TP + FN)$ ) and precision ( $TP/(TP + FP)$ ) by subject. Furthermore, we evaluate the Dice score globally by accumulating true positives (TP), false positives (FP), and false negatives (FN) over all subjects and then computing the Dice score as follows:

$$D_c = \frac{2TP}{2TP + FP + FN} \quad (7.12)$$

For detection, we evaluate the precision, recall, and F1 scores at different Dice score thresholds of detection ranging from 0.1 to 0.9. For each threshold, detections that have a higher Dice score than the threshold are considered true positives, while detections with lower Dice scores than the threshold are considered false positives. Lesions with no predicted segmentation maps (Dice score of 0) are considered false negatives. We also evaluate the average precision, recall, and F1 scores across the range of Dice score thresholds (0.1 to 0.9).

### **Overall Segmentation Performance**

The segmentation performance of the proposed approach and each of the two models comprising the approach is summarized in Table 7.1, and is compared to the other 4 benchmark models. The results outlined in Table 7.1 are achieved using model 1 shown in Fig. 7.2 (b) and explained in Section 7.3.3 as the small lesion focused model in our approach. The small lesion focused model 2 is a proof of concept model that we tested on the clinical dataset to evaluate the validity of models that do not use the spatially contracting encoder and expanding decoder architecture. This model achieved a 73.1% Dice score by subject and an average F1 score for lesion detection that is 1.6% lower than model 1. This was achieved with only 128 feature channels in its deepest layers, which is a fourth of the number of feature channels in the main segmentation model and small lesion focused model 1. However, due to the use of multiple stages at full spatial resolution, scaling in the feature space beyond 128 feature channels is currently not possible due to GPU memory constraints. This initial test of the model demonstrates that

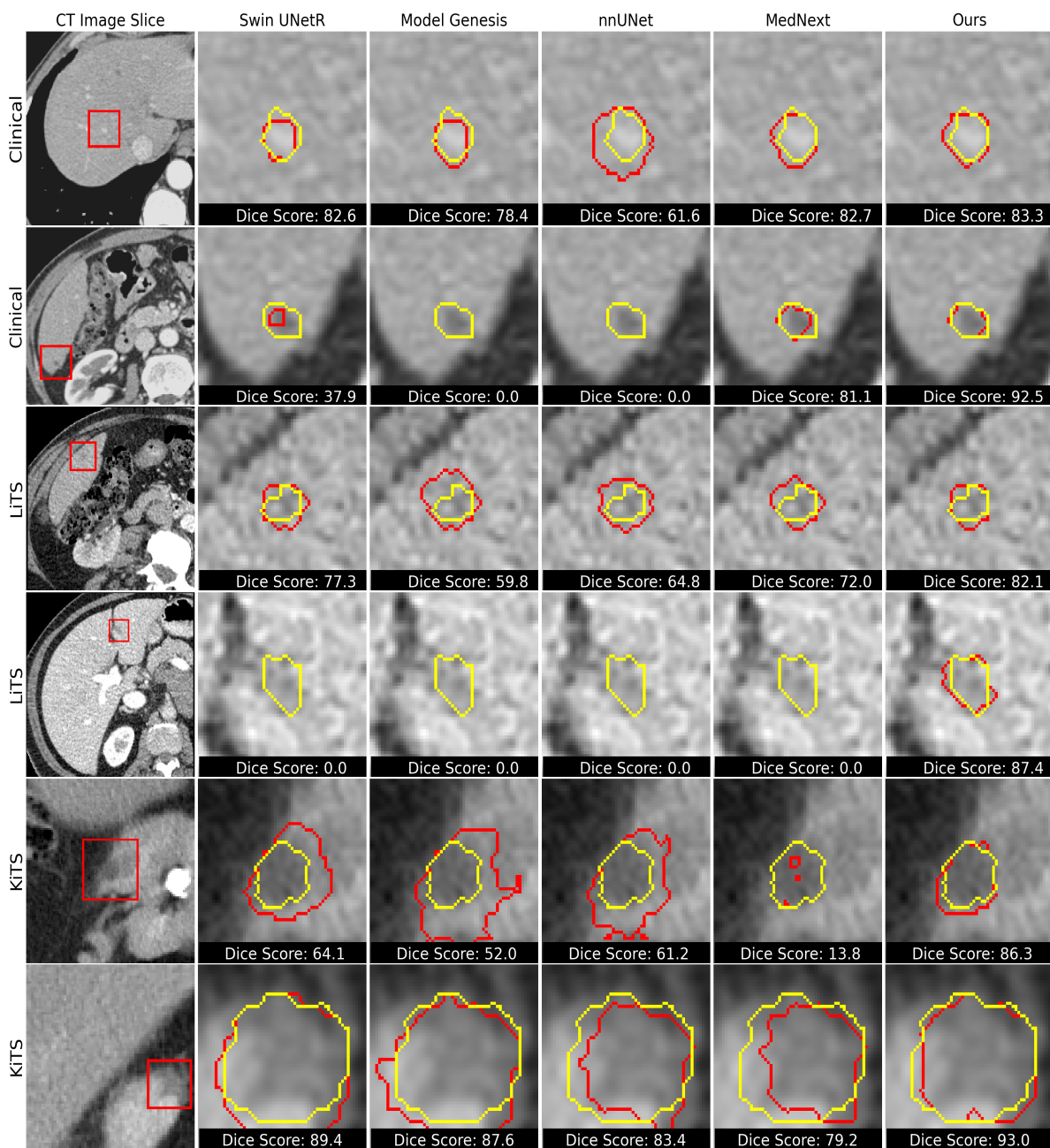
with future memory expansions in GPUs, this model architecture has the potential to improve the detection and segmentation of small lesions due to its increased usage of high resolution features.

In Table 7.1, we also compare the selection method of our proposed approach (Ours-SE) to soft voting ensembling (SVE) and hard voting ensembling (HVE) for all three datasets. The proposed approach constantly outperformed the 4 benchmark models on the three datasets improving the overall relative segmentation performance by 0.9%, 2.9%, and 1.8% for the clinical, LiTS, and KiTS datasets, respectively. This performance improvement results in improved segmentation maps of lesions with better detection and boundaries as shown in Fig. 7.3 and Fig. 7.4. In Fig. 7.3, different lesions from multiple subjects within each of the three datasets are shown together with the ground truth and predicted segmentation boundaries from our proposed approach as well as the 4 benchmark models. In Fig. 7.4, we demonstrate the ability of the proposed selection approach to improve segmentation accuracy when compared to the main segmentation model as well as SVE and HVE predictions.

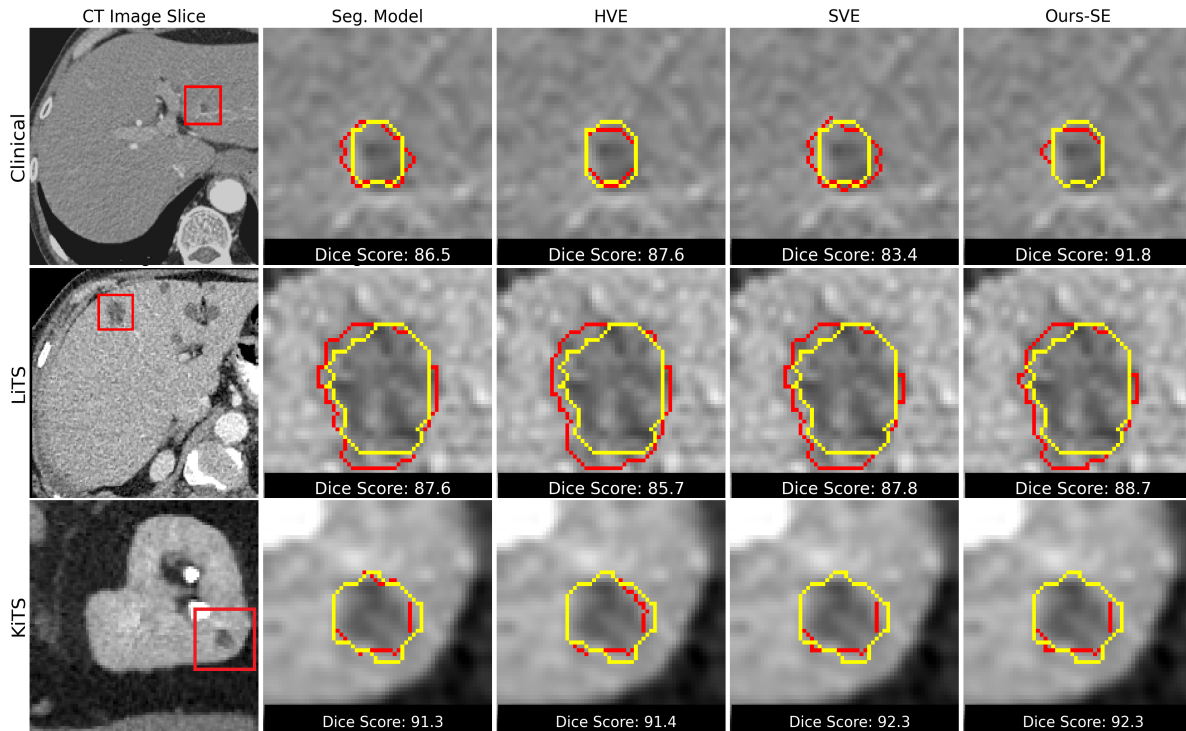
Our approach also reduces performance variability across subjects where the standard deviation of Dice scores across subjects are reduced by 1.5%, 7.9%, and 2.2% for the clinical, LiTS, and KiTS datasets, respectively. Performance improvements for the LiTS dataset are more prevalent than the other two, which is further validation of the proposed approach ability to improve detection rates and segmentation performance for small lesions as the LiTS dataset contains a large number of small lesions when compared to the other two datasets. The Small Lesion Focused model constantly achieves the best recall while maintaining high precision, demonstrating its ability to achieve the overall objective, which is to improve the recovery and detection of lesions, especially small ones. Finally, the overall design paradigm of using two models-one of which is a general segmentation model while the other is focused on small lesions-and combining their predictions whether through our proposed selection approach or other ensembling approaches results in an improved segmentation and detection performance, especially for applications where small lesions are more prevalent.

**Table 7.1:** The proposed approach segmentation performance on the three CT datasets. We compare the performance of the two models in our approach: The main segmentation model (Ours) and the small lesion focused model (Ours-HR) to the four benchmark models. We also compare the proposed selection approach (Ours-SE) to soft voting (SVE) and hard voting (HVE) ensembling. Best results are boldfaced while 2<sup>nd</sup> best are underlined for each dataset in the single model sections. Only the best is boldfaced in the ensemble sections. All metrics are in the range 0 to 100. Values in parentheses represent the standard deviation across subjects.

Dataset	Approach	Model	Global		By Subject		
			Dice	Dice	IoU	Recall	Precision
Clinical	Single Model	SwinUNetR	81.6	68.2 (23.2)	55.8 (23.5)	68.1 (25.0)	78.0 (23.6)
		Model Genesis	80.8	70.7 (22.4)	58.7 (22.8)	73.4 (23.5)	76.6 (23.2)
		nnUNet	83.7	73.4 (22.4)	62.0 (23.0)	74.8 (24.2)	79.4 (22.0)
		MedNext	<u>83.9</u>	<u>75.1</u> (20.1)	<u>63.6</u> (21.1)	<u>76.8</u> (22.6)	<u>80.3</u> (19.1)
		Ours	<b>84.1</b>	<b>75.5</b> (19.8)	<b>63.9</b> (20.8)	<u>76.8</u> (22.2)	<b>80.9</b> (18.9)
		Ours-HR	82.4	74.7 (20.9)	63.2 (21.6)	<b>77.7</b> (21.8)	79.0 (20.5)
	Ensemble	HVE	84.0	75.3 (20.2)	63.7 (21.1)	76.9 (22.6)	<b>81.0</b> (18.2)
		SVE	83.7	75.0 (20.4)	63.4 (21.4)	77.4 (22.3)	79.8 (19.5)
		Ours-SE	<b>84.5</b>	<b>75.8</b> (19.8)	<b>64.4</b> (20.7)	<b>78.4</b> (21.7)	80.1 (18.9)
	LiTS	Single Model	SwinUNetR	79.7	68.4 (28.3)	58.1 (29.5)	60.0 (28.9)
Model Genesis			83.1	73.1 (28.0)	63.4 (27.4)	71.4 (28.6)	75.8 (16.3)
nnUNet			<u>83.5</u>	74.3 (27.6)	63.8 (28.2)	71.4 (30.2)	<u>78.1</u> (15.7)
MedNext			82.5	73.5 (27.2)	63.8 (27.3)	74.4 (30.8)	72.2 (18.7)
Ours			<b>83.7</b>	<u>74.4</u> (27.1)	<b>64.3</b> (27.0)	<u>77.2</u> (27.2)	74.9 (19.2)
Ours-HR			81.3	<b>74.5</b> (25.8)	<b>64.3</b> (26.7)	<b>79.7</b> (27.6)	70.2 (16.6)
Ensemble		HVE	82.5	73.4 (27.3)	63.8 (27.3)	74.5 (30.8)	72.2 (18.7)
		SVE	83.3	74.9 (26.1)	64.5 (26.8)	77.7 (29.6)	73.5 (14.4)
		Ours-SE	<b>83.6</b>	<b>75.6</b> (24.8)	<b>64.9</b> (25.6)	<b>79.1</b> (27.2)	<b>73.8</b> (14.4)
KiTS <sub>23</sub>		Single Model	SwinUNetR	<b>89.5</b>	73.3 (33.1)	66.4 (33.4)	76.9 (27.8)
	Model Genesis		<u>88.1</u>	73.9 (33.2)	67.2 (33.1)	<u>77.6</u> (26.0)	74.2 (33.8)
	nnUNet		87.9	73.1 (33.5)	66.2 (33.3)	77.4 (26.1)	73.5 (33.8)
	MedNext		87.7	<u>75.5</u> (31.9)	68.7 (32.0)	74.9 (24.9)	<u>78.5</u> (32.3)
	Ours		87.9	<b>76.3</b> (30.7)	<b>69.9</b> (31.0)	76.0 (23.3)	<b>78.9</b> (32.2)
	Ours-HR		87.1	75.4 (32.3)	<u>68.8</u> (32.7)	<b>78.7</b> (22.9)	76.2 (33.5)
	Ensemble	HVE	87.6	75.5 (32.0)	68.7 (31.9)	74.9 (24.9)	78.5 (32.3)
		SVE	87.9	76.6 (31.2)	70.0 (31.6)	76.7 (23.1)	<b>78.6</b> (32.7)
		Ours-SE	<b>88.4</b>	<b>76.9</b> (31.2)	<b>70.4</b> (31.6)	<b>77.8</b> (23.0)	77.9 (32.6)



**Figure 7.3:** Qualitative comparison of the proposed approach to the other four benchmark models. For each of the three datasets, two examples are shown. The figure presents a side-by-side assessment of the segmentation performance on lesions of different sizes, shapes and intensity characteristics. The first column shows the original CT image slices cropped to the organ region with the region of interest highlighted in a red bounding box. The subsequent columns show the cropped bounding box region with the outline of segmentation results from the different approaches (in red) and the ground truth segmentation outline (in yellow). The Dice score is listed below each image.



**Figure 7.4:** Qualitative demonstration of the proposed approach ability to improve the segmentation outcome of the predictions from both models. The figure presents a side-by-side comparison of the overall approach to the main segmentation model as well as the SVE and HVE results. A slice from each of the three datasets is shown. The first column shows the original CT image slices cropped to the organ region with the region of interest highlighted in a red bounding box. The subsequent columns show the cropped bounding box region with the outline of segmentation predictions (in red) and the ground truth segmentation outline (in yellow). The Dice score is listed below each image.

#### 7.4.4 Detection Performance

To evaluate the proposed approach lesion detection performance, we computed the precision, recall, and F1 score at different Dice score thresholds (0.1 to 0.9) and calculated the average precision, recall, and F1 scores across all the thresholds as shown in Fig. 7.5. In our analysis we evaluated the performance for lesions of 4 mm in diameter or larger. Lesions with a diameter less than 5 mm are not conclusively investigated using CT scans in clinical settings. We also investigate the precision, recall, and F1 score at Dice score thresholds of 0.25, 0.5, and 0.75, which is summarized in Table 7.2. Overall the proposed approach improves the detection of

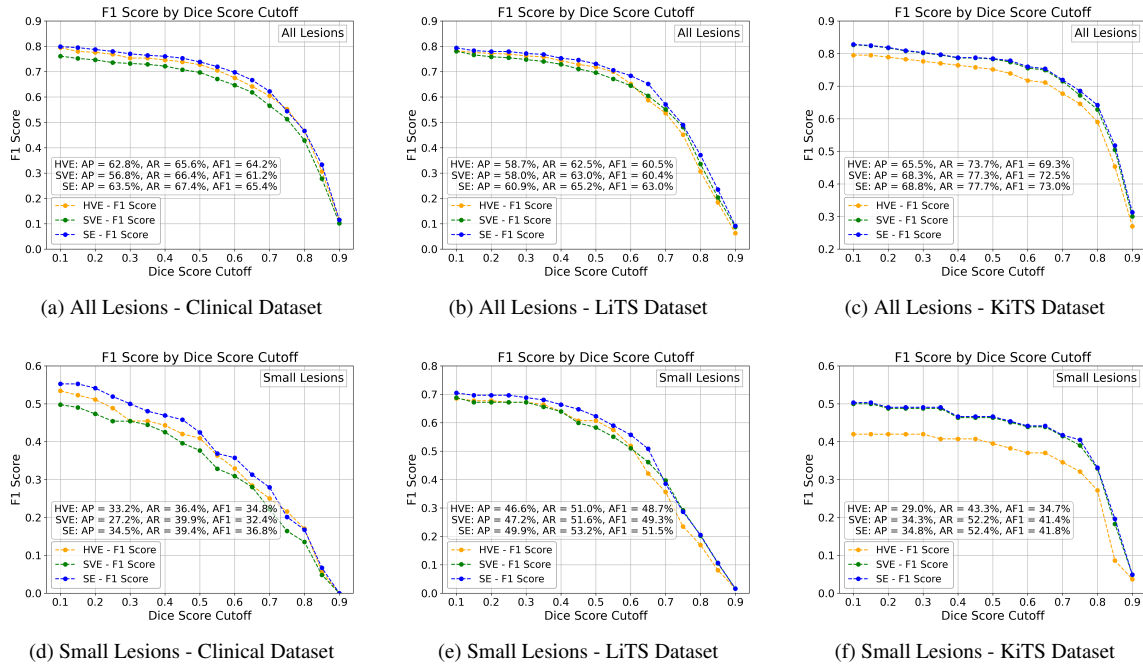
**Table 7.2:** The proposed approach lesion detection performance on the three datasets. The precision (Pr.), recall (Rc.) and F1 score are shown for three Dice score thresholds. The average precision (AP), recall (AR), and F1 score (AF1) across the Dice score threshold range of 0.1 to 0.9 are included. Best results are boldfaced while 2<sup>nd</sup> best are underlined for each dataset. All metrics are in the range 0 to 100.

Dice Score Thr. =		0.25			0.5			0.75			0.1 - 0.9		
Dataset	Approach	Rc.	Pr.	F1	Rc.	Pr.	F1	Rc.	Pr.	F1	AR	AP	AF1
Clinical	HVE	78.6	<u>75.1</u>	<u>76.8</u>	74.4	<b>71.1</b>	<u>72.7</u>	<b>56.4</b>	<b>54.0</b>	<b>55.2</b>	65.6	<u>62.8</u>	<u>64.2</u>
	SVE	<u>79.8</u>	68.3	73.6	<u>75.6</u>	62.2	69.7	55.7	47.6	51.3	<u>66.4</u>	56.8	61.2
	Ours-SE	<b>80.5</b>	<b>75.6</b>	<b>78.0</b>	<b>76.0</b>	<u>70.0</u>	<b>73.8</b>	<u>56.1</u>	<u>52.8</u>	<u>54.4</u>	<b>67.4</b>	<b>63.5</b>	<b>65.4</b>
LiTS	HVE	<u>79.3</u>	<u>74.7</u>	<u>77.2</u>	<u>74.3</u>	<u>69.5</u>	<u>71.9</u>	46.7	43.6	45.1	62.5	<u>58.7</u>	<u>60.5</u>
	SVE	78.5	72.7	75.9	72.4	67.0	69.6	<u>50.5</u>	<u>46.0</u>	<u>48.2</u>	<u>63.0</u>	58.0	60.4
	Ours-SE	<b>80.5</b>	<b>75.5</b>	<b>77.9</b>	<b>75.4</b>	<b>70.0</b>	<b>73.1</b>	<b>50.9</b>	<b>47.2</b>	<b>48.9</b>	<b>65.2</b>	<b>60.9</b>	<b>63.0</b>
KiTS <sub>23</sub>	HVE	83.2	73.9	78.3	79.9	71.0	75.1	68.7	61.0	64.6	73.7	65.5	69.3
	SVE	<b>86.1</b>	<u>76.1</u>	<u>80.8</u>	<u>83.5</u>	<u>73.8</u>	<u>78.3</u>	<u>71.6</u>	<u>63.3</u>	<u>67.2</u>	<u>77.3</u>	<u>68.3</u>	<u>72.5</u>
	Ours-SE	<b>86.1</b>	<b>76.3</b>	<b>80.9</b>	<b>83.6</b>	<b>74.0</b>	<b>78.5</b>	<b>72.9</b>	<b>64.6</b>	<b>68.5</b>	<b>77.7</b>	<b>68.8</b>	<b>73.0</b>

lesions significantly across the three datasets. For small lesions of diameters 4 to 10 mm, the proposed approach significantly improved lesion detection across different Dice score thresholds, which is demonstrated in Fig. 7.5 (d)-(f). The relative improvement in the average F1 score for all lesions in the clinical dataset is 1.8% while for the LiTS and KiTS datasets, the improvement is 4.1% and 0.7%. The improvement in detection performance is also more prevalent in the LiTS dataset similar to the segmentation performance due to the larger ratio of small lesions in the dataset.

### 7.4.5 Ablation Studies

We performed two ablations studies to test the effect of the different design components of our proposed approach. The first ablation study tested the effect of the Dice score threshold and bounding box margin size when evaluating correspondences between lesion segmentation maps in Algorithm 7.1. We observed that the choice of both has a minimal impact on the overall performance of the proposed approach. A Dice score of 0.5 between lesions, which is a logical correspondence threshold, seems to work best when compared to thresholds of 0.6 and 0.7 as



**Figure 7.5:** The detection F1 score of the proposed approach (SE) compared to SVE and HVE across different Dice score thresholds. The average precision (AP), recall (AR), and F1 score (AF1) across the whole range of Dice score thresholds (0.1 to 0.9) is annotated on each of the plots.

shown in Table 7.3. As we increase the threshold, a reduced number of correspondences are created and the proposed selection approach is applied on a smaller number of lesions. Bounding box margins of 10% and 15% work well when creating a region surrounding the lesion to extract features from the surrounding tissue. On average a margin of 15% works slightly better as it allows better sampling for surrounding tissue. As we expand the margin beyond 20%, we observed performance reductions that indicate the importance of localized comparison rather than comparing the lesion to large segments of surrounding tissues.

In the second ablation study, we investigated the effect of reducing the patch size from 96 to 64 voxels, and the impact of the adaptive loss function outlined in Algorithm 7.2 on the LiTS dataset when training and testing the Small Lesion Focused Model. We conducted this study on the LiTS dataset as it contains a large number of small lesions compared to the other two datasets. As shown in Table 7.4, the adaptive loss function improves the segmentation performance of the



**Table 7.3:** The Dice score performance of the proposed approach as the size of the bounding box margin changes for feature extraction of surrounding tissue, and as the Dice score threshold for correspondence changes for lesion predictions pairing in Algorithm 7.1.

Dataset	BBox	Dice Thr.		
	Margin	0.5	0.6	0.7
Clinical	10%	<b>75.89</b>	75.82	75.80
	15%	<u>75.85</u>	75.77	75.75
LiTS	10%	75.54	75.51	75.52
	15%	<b>75.58</b>	<u>75.56</u>	75.53
KiTS <sub>23</sub>	10%	76.89	76.89	76.78
	15%	<b>76.91</b>	<u>76.91</u>	76.85

model for both patch sizes. On the other hand, reducing the patch size to 64 from 96 with the aim of increasing the relative size of small lesions within the field of view of the input patch, did not improve the model performance, but rather degraded it significantly. We believe this can be attributed to one of two reasons, or both. The first reason is the importance of contextual information when segmenting lesions. The second is that the extensively small spatial resolution at the deeper stages of the model when using a small patch size reduces the effectiveness of convolutional kernels to extract and contextualize features spatially for segmentation purposes.

**Table 7.4:** The effect of varying the patch size and using the adaptive loss outlined in Algorithm 7.2 on the small lesion focused model segmentation performance for the LiTS dataset.

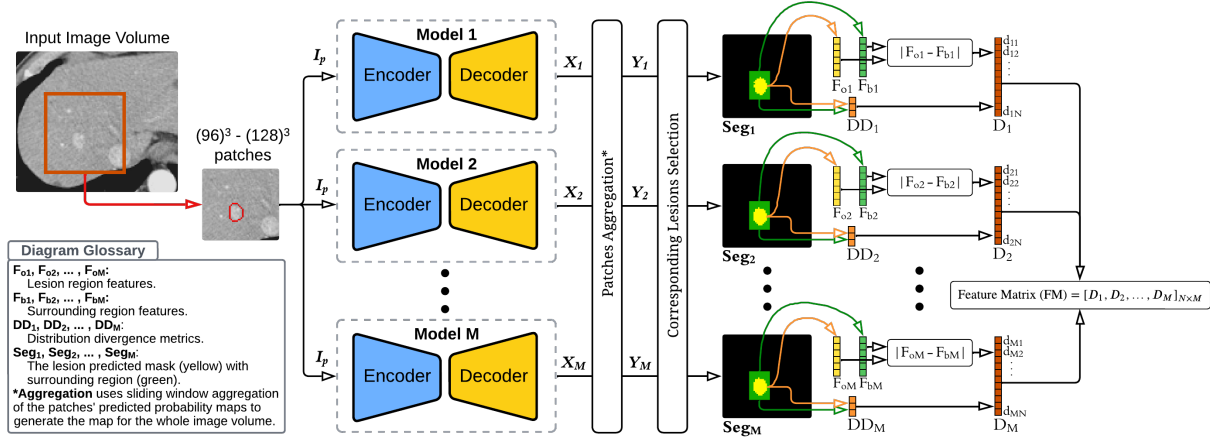
Patch Size		Loss Configuration		Dice Score
64	96	Static	Adaptive	
✓		✓		72.91
✓			✓	73.14
	✓	✓		<u>73.61</u>
	✓		✓	<b>74.52</b>

## 7.5 Extending The Prediction Selection Approach to Multiple Models

To extend the proposed two-model selection approach to accommodate multiple models, we reformulate the correspondence and prediction selection process to handle multiple models simultaneously. The initial approach used the predictions from two models ( $\text{Seg}_1$  and  $\text{Seg}_2$ ) for each correspondence in the set of lesion correspondences, from which a decision was made based on the vector of separation features ( $D_1$  and  $D_2$ ). When extending the approach to  $M$  models, the correspondence of lesions can be challenging as each lesion could be part of multiple correspondences. To overcome this challenge, we designed the lesion correspondence selection approach outlined in Section 7.5.1. Once the set of lesion correspondences is identified for  $M$  models, the segmentation mask ( $\text{Seg}_1, \text{Seg}_2, \dots, \text{Seg}_M$ ) produced by each of the  $M$  models is used for each lesion in the correspondence. The prediction selection approach, which is outlined in Fig. 7.6, uses the separation features vectors ( $D_1, D_2, \dots, D_M$ ) to identify the best prediction mask for each of these lesions.  $D_1, D_2, \dots, D_M$  are generated by concatenating the lesion versus surrounding tissue separation features  $|F_{oi} - F_{bi}|$  and the distribution divergence metrics  $DD_i$  for  $i = 1, 2, \dots, M$  such that  $D_i = [|F_{oi} - F_{bi}|, DD_i] : D_i = [d_{11}, d_{12}, \dots, d_{1N}]$ . The feature matrix (FM) containing these vectors is then created as follows:

$$\text{FM}_{N \times M} = \begin{bmatrix} d_{11} & d_{21} & \cdots & d_{M1} \\ d_{12} & d_{22} & \cdots & d_{M2} \\ \vdots & \vdots & \ddots & \vdots \\ d_{1N} & d_{2N} & \cdots & d_{MN} \end{bmatrix}, \quad (7.13)$$

where  $N$  is the number of features and  $M$  is the number of models. Once the feature matrix  $FM$  is generated, the selection process follows based on the count of features with the largest values across all models. For each feature  $j$  (where  $j = 1, 2, \dots, N$ ) in the FM matrix, we determine



**Figure 7.6:** The extended prediction selection approach for multiple models. The 3D image patch is fed to all models where each generates a segmentation prediction map. Corresponding lesion sets are selected from  $Y_1, Y_2,$  and  $Y_M$  based on a Dice score overlap of 0.5. For corresponding lesions in each model prediction, intensity-based features, which are described in Section 7.3.4, are extracted from the region of the lesions ( $F_o$ ) and surrounding tissue ( $F_b$ ) in addition to intensity distribution divergence metrics ( $DD$ ) to form the lesion versus surrounding tissue separation feature vector ( $D$ ). These vectors are stacked as columns of the feature matrix ( $FM$ ) where the prediction with the highest count of largest separations is chosen.

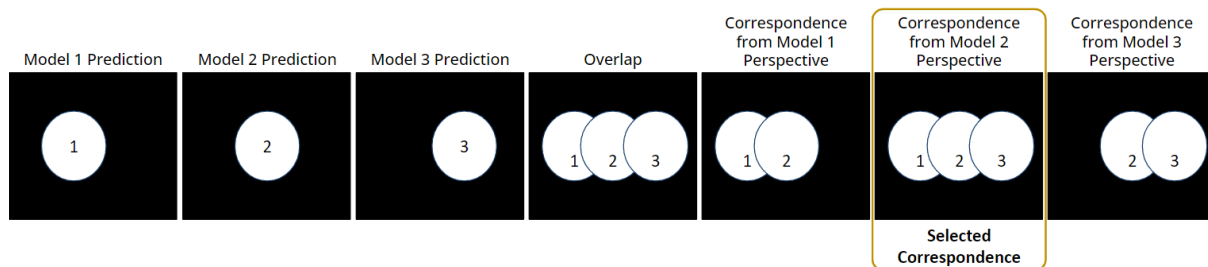
which model  $i$  has the maximum value:

$$i_{j,\max} = \underset{i}{\operatorname{argmax}} FM_{ji}, \forall j \in 1, 2, \dots, N \quad (7.14)$$

The model  $i$  with the maximum count across all features is selected as the best-performing model for the correspondence set, and its predicted mask is selected for the lesion. Once the mask for each correspondence set is selected, the final segmentation map is generated following the algorithm outlined in Algorithm 7.1. We evaluate the extended selection approach on the clinical multi-phase liver lesion dataset we used to test our segmentation approach in Chapter 6 and in this chapter using five models. The dataset details are outlined in section 7.4.1 while the models' details are in section 7.5.2.

### 7.5.1 Multi-Model Corresponding Lesions Selection Algorithm

To identify the sets of corresponding lesions across segmentation masks produced by multiple models, we designed the lesion correspondence set creation algorithm defined in Algorithm 7.3. The most significant challenge encountered when identifying these correspondences is the possibility of a lesion belonging to multiple correspondences depending on the model prediction perspective used to initiate the correspondence set. This is demonstrated in Fig. 7.7. To overcome this challenge, we start the algorithm by initializing an empty set  $S_{LC}$  to store the final lesion correspondence sets, and an empty collection  $C$  to keep track of intermediate correspondence sets. The algorithm iterates over the segmentation masks of each model (denoted as  $\text{Seg}_1, \text{Seg}_2, \dots, \text{Seg}_M$ ), considering the lesions within them. For each model  $\text{Seg}_m$ , it counts the number of lesions, denoted as  $I_m$ . Then, for each lesion  $l_i^m$  where  $i \in \{1, \dots, I_m\}$ , it initializes a set  $C[m, i]$  with the tuple  $(m, i)$ , representing the model index  $m$  and lesion index  $i$ . The algorithm then proceeds to find corresponding lesions in other models by comparing the current lesion  $l_i^m$  against each lesion  $l_k^j$  in other segmentation maps  $\text{Seg}_j$  (where  $j \in \{1, \dots, M\}$  and  $j \neq m$ ). The number of lesions in each model is counted as  $K_j$ . The similarity between lesions  $l_i^m$  and  $l_k^j$  is measured using the Dice score ( $D_c$ ). If  $D_c(l_i^m, l_k^j) \geq 0.5$ , the algorithm considers these lesions to be corresponding and adds  $(j, k)$  to the set  $C[m, i]$ .



**Figure 7.7:** The correspondence selection approach demonstrated on a simulated example of lesion predictions from 3 models. The correspondence set from the perspective of each model prediction is shown together with the selected correspondence out of the 3 possible correspondences.

After the correspondence sets are identified, the algorithm resolves any duplicate corre-

spondences by sorting  $C$  in descending order based on the size (number of lesions) of correspondence sets within it ( $C_{\text{sorted}}$ ). For each correspondence set ( $C_n$ ), it iterates over the pairs  $(i, j)$  and checks if any of these pairs are present in subsequent sets that are smaller in size  $C_k$  (where  $k \in \{i + 1, \dots, N\}$  and  $N$  is the number of correspondence sets in  $C$ ). If so, these smaller sets are removed from  $C$ , ensuring no duplicate correspondences exist from the perspective of each lesion. Finally, the algorithm assigns  $S_{LC}$  to the filtered set  $C$  and returns it as the final set of correspondence sets.

---

**Algorithm 7.3** Lesion Correspondence Set Creation

---

```
1: require: Segmentation masks from  $M$  models:  $\text{Seg}_1, \text{Seg}_2, \dots, \text{Seg}_M$ 
2: output: Final set of lesion correspondences sets  $S_{LC}$ 
3:  $S_{LC} \leftarrow \{\}$ 
4:  $C \leftarrow \{\}$  / empty collection of correspondences sets
   // Identify corresponding lesions across the segmentation masks
    $\text{Seg}_1, \text{Seg}_2, \dots, \text{Seg}_M$ 
5: for each model segmentation map  $\text{Seg}_m$  in  $\text{Seg}_1, \text{Seg}_2, \dots, \text{Seg}_M$  do
6:    $I_m \leftarrow$  number of lesions in  $\text{Seg}_m$ 
7:   for each lesion  $l_i^m$  in  $\text{Seg}_m$  where  $i \in \{1, \dots, I_m\}$  do
8:      $C[m, i] \leftarrow \{(m, i)\}$  /  $(m, i)$  are the model and lesion indices
9:     for  $\text{Seg}_j$  where  $j \in \{1, \dots, M\}$  and  $j \neq m$  do
10:       $K_j \leftarrow$  number of lesions in  $\text{Seg}_j$ 
11:      for each lesion  $l_k^j$  in  $\text{Seg}_j$  where  $k \in \{1, \dots, K_j\}$  do
12:        if  $D_c(l_i^m, l_k^j)^\dagger \geq 0.5$  then
13:           $C[m, i] \leftarrow C[m, i] \cup \{(j, k)\}$ 
14:        end if
15:      end for
16:    end for
17:  end for
18: end for
   // Discard duplicate correspondence sets with a smaller number of
   // lesions for each lesion belonging to a correspondence set with
   // a larger number of lesions
19:  $C_{\text{sorted}} = \{C_1, C_2, \dots, C_N\}: N = |C|$  and  $|C_n| \geq |C_{n+1}| \forall n = 1, 2, \dots, N-1$ 
20: for  $C_n$  where  $n \in \{1, \dots, N\}$  do
21:   for  $(i, j)$  in  $C_n$  do /  $(i, j)$  are the model and lesion indices in  $C_n$ 
22:     if  $(i, j) \in C_k \forall k \in \{i+1, \dots, N\}$  then
23:        $C = C \setminus C_k$  / remove  $C_k$  from  $C$ 
24:     end if
25:   end for
26: end for
27:  $S_{LC} \leftarrow C$ 
28: return:  $S_{LC}$ 
    $^\dagger D_c$  refers to the Dice score.
```

---

## 7.5.2 Enhancing The Recall of Small Lesions Via Size-Based Loss Weighting

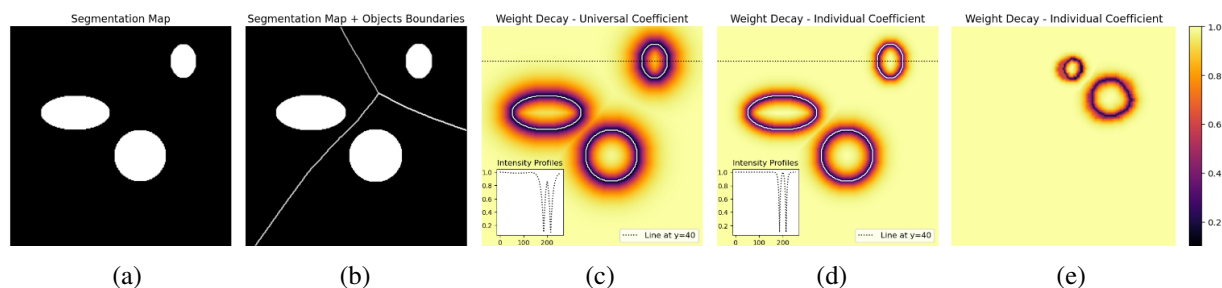
Segmenting lesions is a highly imbalanced task where the lesion class is severely under-represented compared to other tissue, which is considered the background class. When training models on imbalanced datasets, a standard approach assigns loss weights inversely proportional to the class frequency distribution [154, 155]. Although widely used [155], it is ineffective for highly imbalanced datasets because it equates the probability of making an error in both classes. This results in a substantially higher recall than precision when the positive class is significantly underrepresented. We demonstrated this issue in Chapter 4. Another issue with this approach is that it does not address size imbalance within the target class itself. For example, the sizes of liver lesions can differ by up to four orders of magnitude. This leads segmentation models to perform better on larger lesions than smaller ones due to their increased representation in the loss function during training.

To overcome this challenge, we design a size-based loss weighting approach that weights both the BCE and Dice losses in the overall loss function defined in (7.6) for each lesion individually. These weights are chosen to increase as the size of the lesion decreases. In addition to varying the weights based on lesion size, we incorporate a weight decay near the borders of lesions to minimize the impact of annotation and registration errors, as the lesions' borders would be the most impacted by such errors. This decay, which we name the Lesion Border Weight Decay (LBWD), is defined as follows:

$$\text{LBWD} = \begin{cases} 1 - e^{-c_i d_i} & \text{if } \text{Seg}_{gt} \geq 0.5 \\ 1 - e^{-c_o d_o} & \text{otherwise,} \end{cases} \quad (7.15)$$

where  $c_i$  and  $c_o$  are coefficients controlling the rate at which the exponential function approaches 1 for voxels inside and outside lesions, respectively.  $d_i$  and  $d_o$  are the normalized distances

from the lesion border inside and outside lesions, respectively.  $\text{Seg}_{gt}$  refers to the ground truth segmentation map. This weight decay weighs the loss at each voxel inside and outside lesions with respect to that voxel's distance from the lesion border. If  $c_i$  and  $c_o$  are chosen universally across lesions, the loss weights within the lesion will vary significantly based on the lesion shape and size as shown in Fig. 7.8 (c). Therefore, we individually choose the coefficient for each lesion to ensure that at least 80% of the lesion volume is higher than 0.8 (the maximum LBWD value is 1). This is achieved iteratively, where for each lesion, the coefficient  $c_i$  is increased until the condition is met. Higher thresholds than 80% can cause excessively steep decays at the borders for small lesions. To allow the LBWD to recover from the lesion border decay in regions between lesions that are close to each other, we identify the boundaries in the background that separate each of the lesions as shown in Fig. 7.8 (b). The coefficient  $c_o$  is then iteratively chosen to reach a value of 1 at the point on the border closest to the lesion, which is implemented for all the regions surrounding each lesion.



**Figure 7.8:** Synthetically generated segmentation map with three lesions (a). The boundaries in the background that separate the lesions' regions (b). The LBWD map for universally chosen decay coefficients (c), and for individually chosen coefficients by lesion (d). In (e) we show an example of the LBWD map for a slice from the multi-phase liver lesion dataset. The lesions' borders are represented by white contours in (c) and (d).

The final weight map is generated by multiplying the LBWD map by the lesion size-based weight map. This final combined weight map is used to weight the loss function voxel-wise as it is defined spatially rather than by class. For the lesion size-based weight map, we found that a weighting factor that linearly decays from 20 for lesions with a span of 4 mm to 1 for lesions with a span of 45 mm performs well in its ability to maintain a reasonable balance between precision



and recall. The weight remains constant for lesions smaller than 4 mm in diameter and lesions larger than 45 mm in diameter. The maximum span was set to 45 mm as lesions larger than 45 mm are never missed, with the majority of false negatives laying in the range of 4 to 10 mm. To identify these weight ranges, we trained the main segmentation model outlined in Fig. 7.2 (a) for 50 epochs and observed the precision and recall across lesions of all sizes in the dataset; starting with a maximum weight of 100 and reducing the weight by 10 at each trial.

Using this loss weighting approach together with the training setup described in sections 7.3.6 and 7.4.2, we trained three models based on the architecture in Fig. 7.2 (a) in addition to the two models we used in the original two-model selection approach we proposed (refer to section 7.3). For the first and second models, we used the adaptive loss control algorithm outlined in Algorithm 7.2 to maintain a recall higher than precision by 10% and 15% respectively. This allows for an increased recovery of lesions while preserving precision. For the third model, we fine-tuned the main segmentation model from the two-model selection approach using the size-based loss weighting to improve its performance in segmenting small lesions.

### **7.5.3 Experiments and Results**

We evaluated the proposed extended selection approach using five models on the clinical multi-phase liver lesion dataset. Once the segmentation map for each subject is predicted from each model, we generate the set of correspondences. We limit the minimum number of lesions for correspondences to 3, which is the midpoint for the possible number of predictions using 5 models. The segmentation performance of the proposed approach is summarized in Table 7.5, and is compared to the selection approach using two models. The proposed approach was able to improve the segmentation performance by subject (Dice score) and recall while reducing performance variability across subjects, demonstrating the ability of the proposed selection approach to extend beyond two models. In Table 7.5, we also compare the proposed approach segmentation performance to the main segmentation model as we vary the threshold of the

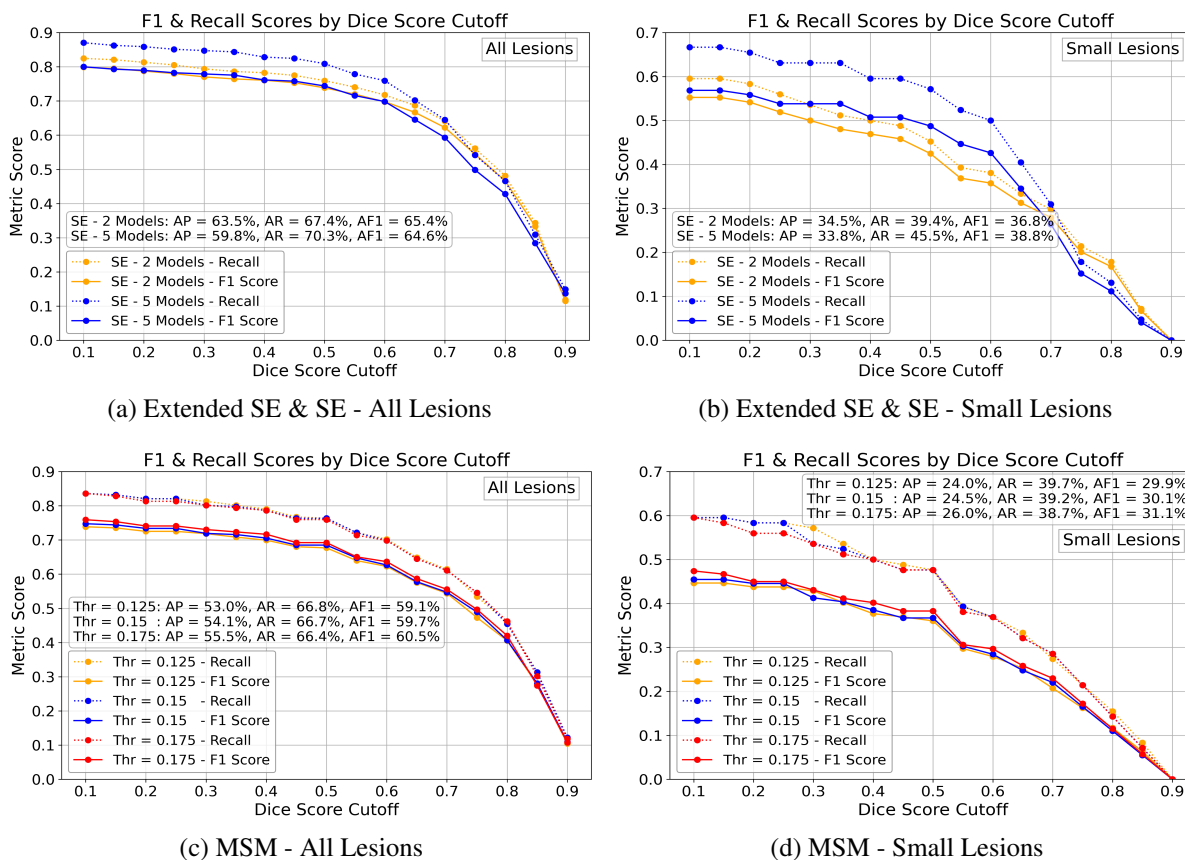
probability map used to generate the segmentation map. This threshold is usually set at 0.5 (the probability map output’s midpoint between 0 and 1). Reducing this threshold increases recall and vice versa. At a threshold of 0.125, the main segmentation model was able to match the recall of the proposed approach. However, the Dice score and precision were both lower by 1% and 1.2%, respectively. This increased recall from reducing the threshold of the model is also attributed to lesions already detected by the model, as we can observe from the large gap in detection performance outlined in Fig. 7.9 between the selection approach and the segmentation model.

**Table 7.5:** The extended selection approach (Extended SE) segmentation performance on the clinical multi-phase liver lesion dataset. The performance of the main segmentation model (MSM) used in the 2-model selection approach (SE) is outlined as the threshold of the probability map is varied at inference. The model at the threshold that matches the recall of the Extended SE approach is highlighted in gray. The best results by subject are boldfaced. All metrics are in the range 0 to 100. Values in parentheses represent the standard deviation across subjects.

Approach	Global		By Subject		
	Dice	Dice	IoU	Recall	Precision
SE - 2 Models	84.5	75.8 (19.8)	<b>64.4</b> (20.7)	78.4 (21.7)	<b>80.1</b> (18.9)
Extended SE - 5 Models	84.0	<b>76.0</b> (19.4)	<b>64.4</b> (20.6)	<b>82.9</b> (18.6)	75.5 (21.5)
MSM - Thr. = 0.1	84.4	74.5 (19.6)	62.5 (20.8)	83.8 (19.9)	72.1 (20.9)
MSM - Thr. = 0.125	84.5	75.0 (19.5)	63.2 (20.7)	83.0 (20.3)	74.3 (19.9)
MSM - Thr. = 0.15	84.6	75.2 (19.5)	63.4 (20.7)	82.4 (20.6)	75.2 (19.7)
MSM - Thr. = 0.175	84.7	75.3 (19.5)	63.6 (20.7)	82.0 (20.6)	75.7 (19.7)
MSM - Thr. = 0.2	84.7	75.4 (19.5)	63.7 (20.7)	81.6 (20.7)	76.2 (19.6)
MSM - Thr. = 0.3	84.6	75.5 (19.7)	63.9 (20.7)	80.0 (21.2)	78.2 (19.0)

As one of the major challenges the extended selection approach aims to overcome is the detection and recovery of lesions, especially small ones, we evaluated its ability to do so by calculating the precision, recall, and F1 score at different Dice score thresholds (0.1 to 0.9). We then computed the average precision, recall, and F1 scores across all the thresholds as shown in Fig. 7.9. We evaluated the performance for lesions of 4 mm in diameter or larger. Lesions with a diameter less than 5 mm are not conclusively investigated using CT scans in clinical settings. We compared the proposed extended selection approach to the 2-model selection approach as well as

the main segmentation model (at probability map thresholds that allow it to achieve comparable recall by subject). The results in Fig. 7.9 (c) and (d) show that even though modifying this threshold to increase recall improves the recall by subject, it does not improve the detection recall of lesions. Overall, the proposed approach improves the detection recall of lesions significantly while maintaining a higher precision performance when compared to the main segmentation model that matches its recall by subject performance (at a threshold = 0.125). For small lesions of diameters 4 to 10 mm, the proposed approach also significantly improved lesion detection across different Dice score thresholds, which is demonstrated in Fig. 7.9 (b) and (d).



**Figure 7.9:** The lesion detection F1 score and recall of the selection (SE - 2 Models) as well as the extended selection (SE - 5 Models) approaches across different Dice score thresholds for all lesions (a) and for small lesions (b). The lesion detection performance of the main segmentation model (MSM) at different probability map thresholds for all lesions (c) and for small lesions (d). The average precision (AP), recall (AR), and F1 score (AF1) across the whole range of Dice score thresholds (0.1 to 0.9) is annotated on each of the plots.

## 7.6 Limitations and Future Prospective

The proposed approach improves the recovery and segmentation of lesions overall in CT scans of the liver and kidney when compared to the current state-of-the-art models as shown in Table 7.1 with a focus on improving small lesions detection and segmentation. The proposed extended selection approach further improves the performance of the selection approach on a by-subject basis as well as lesion-wise, especially for small lesions, as shown in Table 7.5 and Fig. 7.9. However, there are still several instances of small lesions where the proposed approach, and the state-of-the-art models we compare it to, are not able to recover and identify. Hence, the task of detecting and segmenting small lesions can still be further improved and remains an open and challenging task. Based on our analysis of predictions, we observed that some lesions are correctly predicted by some models while missed by other models. Therefore, a logical extension of the work presented in this chapter is developing and investigating a selection approach that can identify predictions from different models to increase the detection and recovery of lesions. Future research could also explore the development of a configurable attention mechanism that adapts to the specific purpose of the segmentation model. This configurable attention mechanism could be designed to aggregate features on different spatial scales using feature extraction kernels of different sizes depending on whether the model aims to segment large or small lesions. By introducing configurable purpose-driven attention modules, the model's ability to capture and delineate objects of various sizes could be enhanced.

## 7.7 Conclusion

We proposed a prediction selection approach for lesion segmentation and detection in CT scans of the liver and kidney that uses the predictions from two models. The first model is a general lesion segmentation model that is designed and trained to segment lesions regardless of their size. The second model is a small lesion focused model that is designed and trained to improve

the segmentation and detection of small lesions while maintaining comparable performance for larger lesions. The selection approach extracts intensity-based features from the lesion region and surrounding tissue, and estimates the best of the two predictions by comparing the lesion and surrounding tissue features' divergence of both predictions. We tested the proposed approach on three different datasets, a clinical dataset of liver lesions and two public datasets, which are the LiTS and KiTS datasets. Our proposed approach was able to improve the detection of lesions by 1.8%, 4.1% and 0.7% for the clinical, LiTS and KiTS datasets, respectively. Segmentation performance by subject was also improved where the proposed approach achieved a segmentation Dice score improvement of 0.9%, 2.9%, and 1.8% for the clinical, LiTS and KiTS datasets, respectively. We also extended the proposed selection approach to apply to multiple models. We tested it on the clinical liver lesion dataset, focusing on improving the detection of lesions, especially small ones. The extended approach improved the segmentation performance by subject and increased the detection rate of all lesions by 4.3% and of small lesions by 15.5% when compared to the 2-model selection approach. Our results demonstrate the approach's ability to improve the detection and segmentation of lesions, both large and small, when compared to the current state-of-the-art segmentation models.

Chapter 7 is, in part, based on the materials as they appear in "Enhancing lesion detection and segmentation via lesion mask selection from multi-specialized model predictions in CT scans of the liver and kidney", Abdullah F. Al-Battal; Van Ha Tang; Quang Duc Tran; Steven Q. H. Truong; Chien Phan; Truong Q. Nguyen; Cheolhong An, submitted to the Computers in Biology and Medicine journal, 2024 as well as the material as it may appear in the currently being prepared submission to the Machine Learning in Medical Imaging workshop of the International Conference on Medical Image Computing and Computed Assisted Intervention (MICCAI), 2024. The dissertation author was the primary investigator and author of these papers.

# Bibliography

- [1] P. C. Lauterbur, “Image formation by induced local interactions: examples employing nuclear magnetic resonance,” *nature*, vol. 242, no. 5394, pp. 190–191, 1973.
- [2] R. Smith-Bindman, M. L. Kwan, E. C. Marlow, M. K. Theis, W. Bolch, S. Y. Cheng, E. J. A. Bowles, J. R. Duncan, R. T. Greenlee, L. H. Kushi, J. D. Pole, A. K. Rahm, N. K. Stout, S. Weinmann, and D. L. Miglioretti, “Trends in use of medical imaging in us health care systems and in ontario, canada, 2000-2016,” *Jama*, vol. 322, no. 9, pp. 843–856, 2019.
- [3] Organisation for Economic Co-operation and Development (OECD), “Diagnostic exams,” 2024, accessed on 2024-04-25. [Online]. Available: <https://data-explorer.oecd.org/?lc=en>
- [4] L. Yu, X. Liu, S. Leng, J. M. Kofler, J. C. Ramirez-Giraldo, M. Qu, J. Christner, J. G. Fletcher, and C. H. McCollough, “Radiation dose reduction in computed tomography: techniques and future perspective,” *Imaging in medicine*, vol. 1, no. 1, p. 65, 2009.
- [5] K.-H. Ng and M. M. Rehani, “X ray imaging goes digital,” pp. 765–766, 2006.
- [6] A. J. Einstein, “Medical imaging: the radiation issue,” *Nature Reviews Cardiology*, vol. 6, no. 6, pp. 436–438, 2009.
- [7] P. Lambin, R. T. Leijenaar, T. M. Deist, J. Peerlings, E. E. de Jong, J. van Timmeren, S. Sanduleanu, R. T. Larue, A. J. Even, A. Jochems, Y. van Wijk, H. Woodruff, J. van Soest, T. Lustberg, E. Roelofs, W. van Elmpt, A. Dekker, F. M. Mottaghy, J. E. Wildberger, and S. Walsh, “Radiomics: the bridge between medical imaging and personalized medicine,” *Nature reviews Clinical oncology*, vol. 14, no. 12, pp. 749–762, 2017.
- [8] S. Lavallée, *Computer-integrated surgery: technology and clinical applications*. MIT Press, 1996.
- [9] A. J. Sinskey, S. N. Finkelstein, and S. M. Cooper, “Medical imaging in drug discovery, part i,” *PharmaGenomics. March/April*, vol. 2025, 2004.
- [10] H.-P. Chan, L. Hadjiiski, C. Zhou, and B. Sahiner, “Computer-aided diagnosis of lung cancer and pulmonary embolism in computed tomography—a review,” *Academic radiology*, vol. 15, no. 5, pp. 535–555, 2008.

- [11] A. Malich, D. R. Fischer, and J. Böttcher, “Cad for mammography: the technique, results, current role and further developments,” *European radiology*, vol. 16, pp. 1449–1460, 2006.
- [12] K. Doi, “Computer-aided diagnosis in medical imaging: historical review, current status and future potential,” *Computerized medical imaging and graphics*, vol. 31, no. 4-5, pp. 198–211, 2007.
- [13] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, “Improving neural networks by preventing co-adaptation of feature detectors,” *arXiv preprint arXiv:1207.0580*, 2012.
- [14] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.
- [15] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014.
- [16] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.
- [17] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.
- [18] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*. Springer, 2015, pp. 234–241.
- [19] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger, “3d u-net: learning dense volumetric segmentation from sparse annotation,” in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2016: 19th International Conference, Athens, Greece, October 17-21, 2016, Proceedings, Part II 19*. Springer, 2016, pp. 424–432.
- [20] M. S. C. of Computing, “The potential of artificial intelligence to bring equity in health care,” accessed on 2023-03-07. [Online]. Available: <https://news.mit.edu/2021/potential-artificial-intelligence-bring-equity-health-care-0601>
- [21] M. Matheny, S. T. Israni, M. Ahmed, and D. Whicher, “Artificial intelligence in health care: The hope, the hype, the promise, the peril,” *Washington, DC: National Academy of Medicine*, 2019.
- [22] L. A. Celi, J. Cellini, M.-L. Charpignon, E. C. Dee, F. DERNONCOURT, R. Eber, W. G. Mitchell, L. Moukheiber, J. Schirmer, J. Situ, J. Paguio, J. Park, J. G. Wawira, and S. Yao, “Sources

- of bias in artificial intelligence that perpetuate healthcare disparities—a global review,” *PLOS Digital Health*, vol. 1, no. 3, p. e0000022, 2022.
- [23] J. G. Betts, K. A. Young, J. A. Wise, E. Johnson, B. Poe, D. H. Kruse, O. Korol, J. E. Johnson, M. Womble, and P. DeSaix, *Anatomy and physiology*. OpenStax College, Rice University, 2013.
- [24] E. Bercovich and M. C. Javitt, “Medical imaging: from roentgen to the digital revolution, and beyond,” *Rambam Maimonides medical journal*, vol. 9, no. 4, 2018.
- [25] E. Samei and D. J. Peck, *Hendee’s physics of medical imaging*. John Wiley & Sons, 2019.
- [26] A. Maier, S. Steidl, V. Christlein, and J. Hornegger, *Medical imaging systems: An introductory guide*. Springer, 2018.
- [27] American College of Radiology, “Medpix™: medical image database,” 2019, accessed on 2023-04-25. [Online]. Available: <https://medpix.nlm.nih.gov>
- [28] I. Cunningham and J. Dobbins III, “Handbook of medical imaging physics and psychophysics,” in *SPIE*, 2000, pp. 79–222.
- [29] O. F. Erondy, *Medical imaging in clinical practice*. BoD–Books on Demand, 2013.
- [30] K. Doi, “Diagnostic imaging over the last 50 years: research and development in medical imaging science and technology,” *Physics in Medicine & Biology*, vol. 51, no. 13, p. R5, 2006.
- [31] P. Murphy and D.-M. Koh, “Imaging in clinical trials,” *Cancer Imaging*, vol. 10, no. 1A, p. S74, 2010.
- [32] P. J. Keall, G. S. Mageras, J. M. Balter, R. S. Emery, K. M. Forster, S. B. Jiang, J. M. Kapatoes, D. A. Low, M. J. Murphy, B. R. Murray, C. R. Ramsey, M. B. Van Herk, S. S. Vedam, J. W. Wong, and E. Yorke, “The management of respiratory motion in radiation oncology report of aapm task group 76 a,” *Medical physics*, vol. 33, no. 10, 2006.
- [33] I. Lerman, R. Hauger, L. Sorkin, J. Proudfoot, B. Davis, A. Huang, K. Lam, B. Simon, and D. G. Baker, “Noninvasive transcutaneous vagus nerve stimulation decreases whole blood culture-derived cytokines and chemokines: a randomized, blinded, healthy control pilot trial,” *Neuromodulation: Technology at the Neural Interface*, vol. 19, no. 3, pp. 283–290, 2016.
- [34] A. F. Al-Battal, Y. Gong, L. Xu, T. Morton, C. Du, Y. Bu, I. R. Lerman, R. Madhavan, and T. Q. Nguyen, “A cnn segmentation-based approach to object detection and tracking in ultrasound scans with application to the vagus nerve detection,” in *2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*. IEEE, 2021, pp. 3322–3327.



- [35] A. F. Al-Battal, I. R. Lerman, and T. Q. Nguyen, "Object detection and tracking in ultrasound scans using an optical flow and semantic segmentation framework based on convolutional neural networks," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 1096–1100.
- [36] T. Leiner, D. Rueckert, A. Suinesiaputra, B. Baeßler, R. Nezafat, I. Išgum, and A. A. Young, "Machine learning in cardiovascular magnetic resonance: basic concepts and applications," *Journal of Cardiovascular Magnetic Resonance*, vol. 21, pp. 1–14, 2019.
- [37] H. Y. Chai, L. K. Wee, T. T. Swee, and S. Hussain, "Gray-level co-occurrence matrix bone fracture detection," *WSEAS Transactions on Systems*, vol. 10, no. 1, pp. 7–16, 2011.
- [38] J. K. Leader, B. Zheng, R. M. Rogers, F. C. Sciurba, A. Perez, B. E. Chapman, S. Patel, C. R. Fuhrman, and D. Gur, "Automated lung segmentation in x-ray computed tomography: development and evaluation of a heuristic threshold-based scheme1," *Academic radiology*, vol. 10, no. 11, pp. 1224–1236, 2003.
- [39] H. Trichili, M.-S. Bouhlel, N. Derbel, and L. Kamoun, "A survey and evaluation of edge detection operators application to medical images," in *IEEE international conference on systems, man and cybernetics*, vol. 4. IEEE, 2002, p. 4.
- [40] R. Pohle and K. D. Toennies, "Segmentation of medical images using adaptive region growing," in *Medical Imaging 2001: Image Processing*, vol. 4322. SPIE, 2001, pp. 1337–1346.
- [41] S. Golemati, A. Sassano, M. J. Lever, A. A. Bharath, S. Dhanjil, and A. N. Nicolaidis, "Carotid artery wall motion estimated from b-mode ultrasound using region tracking and block matching," *Ultrasound in medicine & biology*, vol. 29, no. 3, pp. 387–399, 2003.
- [42] H. M. Harb, A. S. Desuky, A. Mohammed, and R. Jennane, "Histogram of oriented gradients and texture features for bone texture characterization," *Int. J. Comput. Appl*, vol. 975, p. 8887, 2005.
- [43] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
- [44] H. Dong, G. Yang, F. Liu, Y. Mo, and Y. Guo, "Automatic brain tumor detection and segmentation using u-net based fully convolutional networks," in *Medical Image Understanding and Analysis: 21st Annual Conference, MIUA 2017, Edinburgh, UK, July 11–13, 2017, Proceedings 21*. Springer, 2017, pp. 506–517.
- [45] P. Bilic, P. F. Christ, E. Vorontsov, G. Chlebus, H. Chen, Q. Dou, C. Fu, X. Han, P. Heng, J. Hesser, S. Kadoury, T. K. Konopczynski, M. Le, C. Li, X. Li, J. Lipková, J. S. Lowengrub, H. Meine, J. H. Moltz, C. Pal, M. Piraud, X. Qi, J. Qi, M. Rempfler, K. Roth, A. Schenk, A. Sekuboyina, P. Zhou, C. Hülsemeyer, M. Beetz, F. Ettliger, F. Grün, G. Kaissis, F. Lohöfer, R. Braren, J. Holch, F. Hofmann, W. H. Sommer, V. Heinemann, C. Jacobs,

- G. E. H. Mamani, B. van Ginneken, G. Chartrand, A. Tang, M. Drozdal, A. Ben-Cohen, E. Klang, M. M. Amitai, E. Konen, H. Greenspan, J. Moreau, A. Hostettler, L. Soler, R. Vivanti, A. Szeskin, N. Lev-Cohain, J. Sosna, L. Joskowicz, and B. H. Menze, “The liver tumor segmentation benchmark (lits),” *arXiv preprint arXiv:1901.04056*, 2019.
- [46] X. Han, “Automatic liver lesion segmentation using a deep convolutional neural network method,” *arXiv preprint arXiv:1704.07239*, 2017.
- [47] F. Ouhmich, V. Agnus, V. Noblet, F. Heitz, and P. Pessaux, “Liver tissue segmentation in multiphase ct scans using cascaded convolutional neural networks,” *International journal of computer assisted radiology and surgery*, vol. 14, pp. 1275–1284, 2019.
- [48] 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, no. 2, pp. 203–211, 2021.
- [49] S. Gul, M. S. Khan, A. Bibi, A. Khandakar, M. A. Ayari, and M. E. Chowdhury, “Deep learning techniques for liver and liver tumor segmentation: A review,” *Computers in Biology and Medicine*, p. 105620, 2022.
- [50] J. M. J. Valanarasu, V. A. Sindagi, I. Hacihaliloglu, and V. M. Patel, “Kiu-net: Over-complete convolutional architectures for biomedical image and volumetric segmentation,” *IEEE Transactions on Medical Imaging*, vol. 41, no. 4, pp. 965–976, 2021.
- [51] P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features,” in *Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001*, vol. 1. IEEE, 2001, pp. I–I.
- [52] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *International journal of computer vision*, vol. 60, pp. 91–110, 2004.
- [53] H. Bay, T. Tuytelaars, and L. Van Gool, “Surf: Speeded up robust features,” *Lecture notes in computer science*, vol. 3951, pp. 404–417, 2006.
- [54] Y. Freund, R. Schapire, and N. Abe, “A short introduction to boosting,” *Journal-Japanese Society For Artificial Intelligence*, vol. 14, no. 771-780, p. 1612, 1999.
- [55] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *2005 IEEE computer society conference on computer vision and pattern recognition*, vol. 1. IEEE, 2005, pp. 886–893.
- [56] P. Felzenszwalb, D. McAllester, and D. Ramanan, “A discriminatively trained, multi-scale, deformable part model,” in *2008 IEEE conference on computer vision and pattern recognition*. IEEE, 2008, pp. 1–8.

- [57] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, “Overfeat: Integrated recognition, localization and detection using convolutional networks,” in *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014*, 2014.
- [58] T. F. Chan and L. A. Vese, “Active contours without edges,” *IEEE Transactions on image processing*, vol. 10, no. 2, pp. 266–277, 2001.
- [59] S. Beucher, “The watershed transformation applied to image segmentation,” *Scanning Microscopy*, vol. 1992, no. 6, p. 28, 1992.
- [60] D. Michel, C. Panagiotakis, and A. A. Argyros, “Tracking the articulated motion of the human body with two rgbd cameras,” *Machine Vision and Applications*, vol. 26, pp. 41–54, 2015.
- [61] J. W. Davis and A. F. Bobick, “The representation and recognition of human movement using temporal templates,” in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE, 1997, pp. 928–934.
- [62] D. Comaniciu, V. Ramesh, and P. Meer, “Kernel-based object tracking,” *IEEE Transactions on pattern analysis and machine intelligence*, vol. 25, no. 5, pp. 564–577, 2003.
- [63] B. K. Horn and B. G. Schunck, “Determining optical flow,” *Artificial intelligence*, vol. 17, no. 1-3, pp. 185–203, 1981.
- [64] M. Tan, R. Pang, and Q. V. Le, “Efficientdet: Scalable and efficient object detection,” in *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2020.
- [65] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, “Yolox: Exceeding yolo series in 2021,” *arXiv preprint arXiv:2107.08430*, 2021.
- [66] S. Ren, K. He, R. Girshick, and J. Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks,” in *Advances in neural information processing systems*, 2015, pp. 91–99.
- [67] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, and A. C. Berg, “SSD: Single shot multibox detector,” in *European conference on computer vision*. Springer, 2016, pp. 21–37.
- [68] S. G. Lee, J. S. Bae, H. Kim, J. H. Kim, and S. Yoon, “Liver lesion detection from weakly-labeled multi-phase ct volumes with a grouped single shot multibox detector,” in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2018, pp. 693–701.
- [69] V. Badrinarayanan, A. Kendall, and R. Cipolla, “Segnet: A deep convolutional encoder-decoder architecture for image segmentation,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.

- [70] Z. Zhou, M. M. R. Siddiquee, N. Tajbakhsh, and J. Liang, "Unet++: A nested u-net architecture for medical image segmentation," in *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*. Springer, 2018, pp. 3–11.
- [71] O. Oktay, J. Schlemper, L. L. Folgoc, M. C. H. Lee, M. P. Heinrich, K. Misawa, K. Mori, S. G. McDonagh, N. Y. Hammerla, B. Kainz, B. Glocker, and D. Rueckert, "Attention u-net: Learning where to look for the pancreas," *arXiv preprint arXiv:1804.03999*, 2018.
- [72] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. Torr, "Fully-convolutional siamese networks for object tracking," in *Computer Vision—ECCV 2016 Workshops: Amsterdam, The Netherlands, October 8–10 and 15–16, 2016, Proceedings, Part II 14*. Springer, 2016, pp. 850–865.
- [73] X. Dong and J. Shen, "Triplet loss in siamese network for object tracking," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 459–474.
- [74] A. Giachetti, "Matching techniques to compute image motion," *Image and Vision Computing*, vol. 18, no. 3, pp. 247–260, 2000.
- [75] M. K. Almekkawy, Y. Adibi, F. Zheng, E. Ebbini, and M. Chirala, "Two-dimensional speckle tracking using zero phase crossing with riesz transform," in *Proceedings of Meetings on Acoustics 168ASA*, vol. 22, no. 1. Acoustical Society of America, 2014, p. 020004.
- [76] D. C. Wang, R. Klatzky, B. Wu, G. Weller, A. R. Sampson, and G. D. Stetten, "Fully automated common carotid artery and internal jugular vein identification and tracking using b-mode ultrasound," *IEEE Transactions on Biomedical Engineering*, vol. 56, no. 6, pp. 1691–1699, 2009.
- [77] S. Bharadwaj and M. Almekkawy, "Deep learning based motion tracking of ultrasound image sequences," in *2020 IEEE International Ultrasonics Symposium (IUS)*, 2020, pp. 1–4.
- [78] S. Bharadwaj and M. Almekkawy, "Faster search algorithm for speckle tracking in ultrasound images," in *2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*. IEEE, 2020, pp. 2142–2146.
- [79] A. Jalalian, S. B. Mashohor, H. R. Mahmud, M. I. B. Saripan, A. R. B. Ramli, and B. Karasfi, "Computer-aided detection/diagnosis of breast cancer in mammography and ultrasound: a review," *Clinical imaging*, vol. 37, no. 3, pp. 420–426, 2013.
- [80] K. Drukker, M. L. Giger, K. Horsch, M. A. Kupinski, C. J. Vyborny, and E. B. Mendelson, "Computerized lesion detection on breast ultrasound," *Medical Physics*, vol. 29, no. 7, pp. 1438–1446, 2002.
- [81] K. Horsch, M. L. Giger, L. A. Venta, and C. J. Vyborny, "Computerized diagnosis of breast lesions on ultrasound," *Medical physics*, vol. 29, no. 2, pp. 157–164, 2002.

- [82] B. Liu, H.-D. Cheng, J. Huang, J. Tian, X. Tang, and J. Liu, "Probability density difference-based active contour for ultrasound image segmentation," *Pattern Recognition*, vol. 43, no. 6, pp. 2028–2042, 2010.
- [83] Z. Cao, L. Duan, G. Yang, T. Yue, Q. Chen, H. Fu, and Y. Xu, "Breast tumor detection in ultrasound images using deep learning," in *International Workshop on Patch-based Techniques in Medical Imaging*. Springer, 2017, pp. 121–128.
- [84] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.
- [85] M. H. Yap, G. Pons, J. Marti, S. Ganau, M. Sentis, R. Zwigelaar, A. K. Davison, and R. Marti, "Automated breast ultrasound lesions detection using convolutional neural networks," *IEEE journal of biomedical and health informatics*, vol. 22, no. 4, pp. 1218–1226, 2017.
- [86] M. Byra, P. Jarosik, A. Szubert, M. Galperin, H. Ojeda-Fournier, L. Olson, M. O'Boyle, C. Comstock, and M. Andre, "Breast mass segmentation in ultrasound with selective kernel u-net convolutional neural network," *Biomedical Signal Processing and Control*, vol. 61, p. 102027, 2020.
- [87] Y. Wang, C. Qin, C. Lin, D. Lin, M. Xu, X. Luo, T. Wang, A. Li, and D. Ni, "3d inception u-net with asymmetric loss for cancer detection in automated breast ultrasound," *Medical Physics*, vol. 47, no. 11, pp. 5582–5591, 2020.
- [88] M. Villa, G. Dardenne, M. Nasan, H. Letissier, C. Hamitouche, and E. Stindel, "Fcn-based approach for the automatic segmentation of bone surfaces in ultrasound images," *International journal of computer assisted radiology and surgery*, vol. 13, no. 11, pp. 1707–1716, 2018.
- [89] H. Yang, C. Shan, A. F. Kolen, and P. H. de With, "Efficient catheter segmentation in 3d cardiac ultrasound using slice-based fcn with deep supervision and f-score loss," in *2019 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2019, pp. 260–264.
- [90] C. F. Baumgartner, K. Kamnitsas, J. Matthew, T. P. Fletcher, S. Smith, L. M. Koch, B. Kainz, and D. Rueckert, "Sononet: real-time detection and localisation of fetal standard scan planes in freehand ultrasound," *IEEE transactions on medical imaging*, vol. 36, no. 11, pp. 2204–2215, 2017.
- [91] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*. Springer, 2014, pp. 740–755.
- [92] A. Sotiras, C. Davatzikos, and N. Paragios, "Deformable medical image registration: A survey," *IEEE transactions on medical imaging*, vol. 32, no. 7, pp. 1153–1190, 2013.

- [93] D. Shen, “Image registration by local histogram matching,” *Pattern Recognition*, vol. 40, no. 4, pp. 1161–1172, 2007.
- [94] T. Vercauteren, X. Pennec, A. Perchant, and N. Ayache, “Diffeomorphic demons: Efficient non-parametric image registration,” *NeuroImage*, vol. 45, no. 1, pp. S61–S72, 2009.
- [95] S. Klein, M. Staring, K. Murphy, M. A. Viergever, and J. P. Pluim, “Elastix: a toolbox for intensity-based medical image registration,” *IEEE Transactions on Medical Imaging*, vol. 29, no. 1, pp. 196–205, 2009.
- [96] B. B. Avants, N. J. Tustison, G. Song, P. A. Cook, A. Klein, and J. C. Gee, “A reproducible evaluation of ants similarity metric performance in brain image registration,” *NeuroImage*, vol. 54, no. 3, pp. 2033–2044, 2011.
- [97] A. Klein, J. Andersson, B. A. Ardekani, J. Ashburner, B. Avants, M.-C. Chiang, G. E. Christensen, D. L. Collins, J. Gee, P. Hellier, J. H. Song, M. Jenkinson, C. Lepage, D. Rueckert, P. Thompson, T. Vercauteren, R. P. Woods, J. J. Mann, and R. V. Parsey, “Evaluation of 14 nonlinear deformation algorithms applied to human brain mri registration,” *Neuroimage*, vol. 46, no. 3, pp. 786–802, 2009.
- [98] B. B. Avants, C. L. Epstein, M. Grossman, and J. C. Gee, “Symmetric diffeomorphic image registration with cross-correlation: evaluating automated labeling of elderly and neurodegenerative brain,” *Medical Image Analysis*, vol. 12, no. 1, pp. 26–41, 2008.
- [99] P. Viola and W. M. Wells III, “Alignment by maximization of mutual information,” *International Journal of Computer Vision*, vol. 24, no. 2, pp. 137–154, 1997.
- [100] A. P. Zijdenbos, B. M. Dawant, R. A. Margolin, and A. C. Palmer, “Morphometric analysis of white matter lesions in mr images: method and validation,” *IEEE Transactions on Medical Imaging*, vol. 13, no. 4, pp. 716–724, 1994.
- [101] K. A. Eppenhof and J. P. Pluim, “Pulmonary ct registration through supervised learning with convolutional neural networks,” *IEEE Transactions on Medical Imaging*, vol. 38, no. 5, pp. 1097–1105, 2018.
- [102] H. Sokooti, B. de Vos, F. Berendsen, M. Ghafourian, S. Yousefi, B. P. Lelieveldt, I. Isgum, and M. Staring, “3d convolutional neural networks image registration based on efficient supervised learning from artificial deformations,” *arXiv preprint arXiv:1908.10235*, 2019.
- [103] X. Cao, J. Yang, J. Zhang, D. Nie, M. Kim, Q. Wang, and D. Shen, “Deformable image registration based on similarity-steered cnn regression,” in *Proc. International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2017, pp. 300–308.
- [104] X. Yang, R. Kwitt, M. Styner, and M. Niethammer, “Quicksilver: Fast predictive image registration—a deep learning approach,” *NeuroImage*, vol. 158, pp. 378–396, 2017.

- [105] G. Balakrishnan, A. Zhao, M. R. Sabuncu, J. Guttag, and A. V. Dalca, “Voxelmorph: a learning framework for deformable medical image registration,” *IEEE Transactions on Medical Imaging*, vol. 38, no. 8, pp. 1788–1800, 2019.
- [106] M. Hoffmann, B. Billot, D. N. Greve, J. E. Iglesias, B. Fischl, and A. V. Dalca, “Synthmorph: learning contrast-invariant registration without acquired images,” *IEEE transactions on medical imaging*, vol. 41, no. 3, pp. 543–558, 2021.
- [107] T. C. Mok and A. C. Chung, “Conditional deformable image registration with convolutional neural network,” in *Medical Image Computing and Computer Assisted Intervention–MICCAI 2021: 24th International Conference, Strasbourg, France, September 27–October 1, 2021, Proceedings, Part IV 24*. Springer, 2021, pp. 35–45.
- [108] Y. Yuan, “Hierarchical convolutional-deconvolutional neural networks for automatic liver and tumor segmentation,” *arXiv preprint arXiv:1710.04540*, 2017.
- [109] 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, no. 12, pp. 2663–2674, 2018.
- [110] S.-T. Tran, C.-H. Cheng, and D.-G. Liu, “A multiple layer u-net, u n-net, for liver and liver tumor segmentation in ct,” *IEEE Access*, vol. 9, pp. 3752–3764, 2020.
- [111] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.
- [112] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
- [113] A. Khanna, N. D. Londhe, and S. Gupta, “A deep residual u-net convolutional neural network for automated lung segmentation in computed tomography images,” *Biocybernetics and Biomedical Engineering*, vol. 40, no. 3, pp. 1314–1327, 2020.
- [114] Z. Zhou, M. M. R. Siddiquee, N. Tajbakhsh, and J. Liang, “Unet++: Redesigning skip connections to exploit multiscale features in image segmentation,” *IEEE transactions on medical imaging*, vol. 39, no. 6, pp. 1856–1867, 2019.
- [115] A. Hatamizadeh, Y. Tang, V. Nath, D. Yang, A. Myronenko, B. Landman, H. R. Roth, and D. Xu, “Unetr: Transformers for 3d medical image segmentation,” in *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, 2022, pp. 574–584.
- [116] A. Hatamizadeh, V. Nath, Y. Tang, D. Yang, H. R. Roth, and D. Xu, “Swin unetr: Swin transformers for semantic segmentation of brain tumors in mri images,” in *International MICCAI Brainlesion Workshop*. Springer, 2021, pp. 272–284.

- [117] S. Roy, G. Koehler, C. Ulrich, M. Baumgartner, J. Petersen, F. Isensee, P. F. Jaeger, and K. H. Maier-Hein, “Mednext: transformer-driven scaling of convnets for medical image segmentation,” in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2023, pp. 405–415.
- [118] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, “A convnet for the 2020s,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 11 976–11 986.
- [119] P. Bilic, P. Christ, H. B. Li, E. Vorontsov, A. Ben-Cohen, G. Kaissis, A. Szeskin, C. Jacobs, G. E. H. Mamani, G. Chartrand, F. Lohöfer, J. W. Holch, W. Sommer, F. Hofmann, A. Hostettler, N. Lev-Cohain, M. Drozdal, M. M. Amitai, R. Vivanti, J. Sosna, I. Ezhov, A. Sekuboyina, F. Navarro, F. Kofler, J. C. Paetzold, S. Shit, X. Hu, J. Lipková, M. Rempfler, M. Piraud, J. Kirschke, B. Wiestler, Z. Zhang, C. Hülsemeyer, M. Beetz, F. Ettliger, M. Antonelli, W. Bae, M. Bellver, L. Bi, H. Chen, G. Chlebus, E. B. Dam, Q. Dou, C.-W. Fu, B. Georgescu, X. G. i Nieto, F. Gruen, X. Han, P.-A. Heng, J. Hesser, J. H. Moltz, C. Igel, F. Isensee, P. Jäger, F. Jia, K. C. Kaluva, M. Khened, I. Kim, J.-H. Kim, S. Kim, S. Kohl, T. Konopczynski, A. Kori, G. Krishnamurthi, F. Li, H. Li, J. Li, X. Li, J. Lowengrub, J. Ma, K. Maier-Hein, K.-K. Maninis, H. Meine, D. Merhof, A. Pai, M. Perslev, J. Petersen, J. Pont-Tuset, J. Qi, X. Qi, O. Rippel, K. Roth, I. Sarasua, A. Schenk, Z. Shen, J. Torres, C. Wachinger, C. Wang, L. Weninger, J. Wu, D. Xu, X. Yang, S. C.-H. Yu, Y. Yuan, M. Yue, L. Zhang, J. Cardoso, S. Bakas, R. Braren, V. Heinemann, C. Pal, A. Tang, S. Kadoury, L. Soler, B. van Ginneken, H. Greenspan, L. Joskowicz, and B. Menze, “The liver tumor segmentation benchmark (lits),” *Medical Image Analysis*, vol. 84, p. 102680, 2023.
- [120] Z. Zhou, V. Sodha, J. Pang, M. B. Gotway, and J. Liang, “Models genesis,” *Medical image analysis*, vol. 67, p. 101840, 2021.
- [121] A. L. Klibanov and J. A. Hossack, “Ultrasound in radiology: from anatomic, functional, molecular imaging to drug delivery and image-guided therapy,” *Investigative radiology*, vol. 50, no. 9, p. 657, 2015.
- [122] D. L. Miller, N. B. Smith, M. R. Bailey, G. J. Czarnota, K. Hynynen, I. R. S. Makin, and Bioeffects Committee of the American Institute of Ultrasound in Medicine, “Overview of therapeutic ultrasound applications and safety considerations,” *Journal of ultrasound in medicine*, vol. 31, no. 4, pp. 623–634, 2012.
- [123] E. Tegnander and S. Eik-Nes, “The examiner’s ultrasound experience has a significant impact on the detection rate of congenital heart defects at the second-trimester fetal examination,” *Ultrasound in Obstetrics and Gynecology*, vol. 28, no. 1, pp. 8–14, 2006.
- [124] A. Bochkovskiy, C. Wang, and H. M. Liao, “Yolov4: Optimal speed and accuracy of object detection,” *arXiv preprint arXiv:2004.10934*, 2020.
- [125] W. Al-Dhabyani, M. Gomaa, H. Khaled, and A. Fahmy, “Dataset of breast ultrasound images,” *Data in brief*, vol. 28, 2020.



- [126] T. L. Van den Heuvel, D. de Bruijn, C. L. de Korte, and B. Ginneken, “Automated measurement of fetal head circumference using 2d ultrasound images,” *PloS one*, vol. 13, no. 8, 2018.
- [127] A. T. Ahuja, *Diagnostic Ultrasound: Head and Neck*, ser. Diagnostic Ultrasound. W. B. Saunders, 2019.
- [128] A. Dosovitskiy, J. T. Springenberg, M. Riedmiller, and T. Brox, “Discriminative unsupervised feature learning with convolutional neural networks,” in *Advances in Neural Information Processing Systems 27*. Curran Associates, Inc., 2014, pp. 766–774.
- [129] J. Wang and L. Perez, “The effectiveness of data augmentation in image classification using deep learning,” *Convolutional Neural Networks Vis. Recognit*, vol. 11, 2017.
- [130] P. Simard, D. Steinkraus, and J. Platt, “Best practices for convolutional neural networks applied to visual document analysis,” in *Seventh International Conference on Document Analysis and Recognition, 2003. Proceedings.*, vol. 2, 2003, pp. 958–958.
- [131] O. Chapelle, J. Weston, L. Bottou, and V. Vapnik, “Vicinal risk minimization,” in *Proc. of the 13th International Conference on Neural Information Processing Systems*, ser. NIPS’00. Cambridge, MA, USA: MIT Press, 2000, p. 395–401.
- [132] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, “mixup: Beyond empirical risk minimization,” in *International Conference on Learning Representations*, 2018. [Online]. Available: <https://openreview.net/forum?id=r1Ddp1-Rb>
- [133] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, “The pascal visual object classes (voc) challenge,” *International journal of computer vision*, vol. 88, no. 2, pp. 303–338, 2010.
- [134] J. Tompson, R. Goroshin, A. Jain, Y. LeCun, and C. Bregler, “Efficient object localization using convolutional networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 648–656.
- [135] T.-W. Hui, X. Tang, and C. C. Loy, “Liteflownet: A lightweight convolutional neural network for optical flow estimation,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8981–8989.
- [136] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. Van Der Smagt, D. Cremers, and T. Brox, “Flownet: Learning optical flow with convolutional networks,” in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 2758–2766.
- [137] N. Mayer, E. Ilg, P. Hausser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox, “A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 4040–4048.

- [138] A. J. Shepard, B. Wang, T. K. Foo, and B. P. Bednarz, "A block matching based approach with multiple simultaneous templates for the real-time 2d ultrasound tracking of liver vessels," *Medical physics*, vol. 44, no. 11, pp. 5889–5900, 2017.
- [139] T. Williamson, W. Cheung, S. K. Roberts, and S. Chauhan, "Ultrasound-based liver tracking utilizing a hybrid template/optical flow approach," *International journal of computer assisted radiology and surgery*, vol. 13, no. 10, pp. 1605–1615, 2018.
- [140] A. Gomariz, W. Li, E. Ozkan, C. Tanner, and O. Goksel, "Siamese networks with location prior for landmark tracking in liver ultrasound sequences," in *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*. IEEE, 2019, pp. 1757–1760.
- [141] M. Makhinya and O. Goksel, "Motion tracking in 2d ultrasound using vessel models and robust optic-flow," *Proceedings of MICCAI CLUST*, vol. 20, pp. 20–27, 2015.
- [142] V. De Luca, J. Banerjee, A. Hallack, S. Kondo, M. Makhinya, D. Nouri, L. Royer, A. Cifor, G. Dardenne, O. Goksel, M. J. Gooding, C. Klink, A. Krupa, A. Le Bras, M. Marchal, A. Moelker, W. J. Niessen, B. W. Papiez, A. Rothberg, J. Schnabel, T. van Walsum, E. Harris, M. A. Lediju Bell, and C. Tanner, "Evaluation of 2d and 3d ultrasound tracking algorithms and impact on ultrasound-guided liver radiotherapy margins," *Medical physics*, vol. 45, no. 11, 2018.
- [143] L. K. Lee, S. C. Liew, and W. J. Thong, "A review of image segmentation methodologies in medical image," in *Advanced Computer and Communication Engineering Technology: Proceedings of the 1st International Conference on Communication and Computer Engineering*. Springer, 2015, pp. 1069–1080.
- [144] D. Jha, M. A. Riegler, D. Johansen, P. Halvorsen, and H. D. Johansen, "Doubleu-net: A deep convolutional neural network for medical image segmentation," in *2020 IEEE 33rd International symposium on computer-based medical systems (CBMS)*. IEEE, 2020, pp. 558–564.
- [145] D. Jha, P. H. Smedsrud, D. Johansen, T. De Lange, H. D. Johansen, P. Halvorsen, and M. A. Riegler, "A comprehensive study on colorectal polyp segmentation with resunet++, conditional random field and test-time augmentation," *IEEE journal of biomedical and health informatics*, vol. 25, no. 6, pp. 2029–2040, 2021.
- [146] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.
- [147] S. Jadon, "A survey of loss functions for semantic segmentation," in *2020 IEEE conference on computational intelligence in bioinformatics and computational biology (CIBCB)*. IEEE, 2020, pp. 1–7.
- [148] N. Heller, N. J. Sathianathen, A. A. Kalapara, E. Walczak, K. Moore, H. Kaluzniak, J. E. Rosenberg, P. Blake, Z. Rengel, M. Oestreich, J. Dean, M. B. Tradewell, A. Shah, R. Tejpaul, Z. Edgerton, M. Peterson, S. Raza, S. K. Regmi, N. Papanikolopoulos, and

- C. J. Weight, “The kits19 challenge data: 300 kidney tumor cases with clinical context, ct semantic segmentations, and surgical outcomes,” *arXiv preprint arXiv:1904.00445*, 2019.
- [149] J. Bergstra and Y. Bengio, “Random search for hyper-parameter optimization.” *Journal of machine learning research*, vol. 13, no. 2, 2012.
- [150] S. S. M. Salehi, D. Erdogmus, and A. Gholipour, “Tversky loss function for image segmentation using 3d fully convolutional deep networks,” in *International workshop on machine learning in medical imaging*. Springer, 2017, pp. 379–387.
- [151] K. T. Bae, “Intravenous contrast medium administration and scan timing at ct: considerations and approaches,” *Radiology*, vol. 256, no. 1, pp. 32–61, 2010.
- [152] A. F. Al-Battal, I. R. Lerman, and T. Q. Nguyen, “Multi-path decoder u-net: A weakly trained real-time segmentation network for object detection and localization in ultrasound scans,” *Computerized Medical Imaging and Graphics*, vol. 107, p. 102205, 2023.
- [153] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, “Algorithms for hyper-parameter optimization,” *Advances in neural information processing systems*, vol. 24, 2011.
- [154] C. H. Sudre, W. Li, T. Vercauteren, S. Ourselin, and M. Jorge Cardoso, “Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations,” in *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support: Third International Workshop, DLMIA 2017, and 7th International Workshop, ML-CDS 2017*. Springer, 2017, pp. 240–248.
- [155] T. Sugino, T. Kawase, S. Onogi, T. Kin, N. Saito, and Y. Nakajima, “Loss weightings for improving imbalanced brain structure segmentation using fully convolutional networks,” in *Healthcare*, vol. 9, no. 8. MDPI, 2021, p. 938.
- [156] I. Loshchilov and F. Hutter, “Decoupled weight decay regularization,” *arXiv preprint arXiv:1711.05101*, 2017.
- [157] A. Fernández, S. García, M. Galar, R. C. Prati, B. Krawczyk, and F. Herrera, *Learning from imbalanced data sets*. Springer, 2018, vol. 10.
- [158] L. C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, “Encoder-decoder with atrous separable convolution for semantic image segmentation,” in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 801–818.
- [159] J. Wang and X. Liu, “Medical image recognition and segmentation of pathological slices of gastric cancer based on deeplab v3+ neural network,” *Computer Methods and Programs in Biomedicine*, vol. 207, p. 106210, 2021.
- [160] D. Hill, P. Batchelor, and M. Holden, “Hawkes dj. medical image registration,” *Phys Med Biol*, vol. 46, pp. R1–R45, 2001.

- [161] S. Oh and S. Kim, “Deformable image registration in radiation therapy,” *Radiation oncology journal*, vol. 35, no. 2, p. 101, 2017.
- [162] J. P. Pluim and J. M. Fitzpatrick, “Image registration,” *IEEE Transactions on Medical Imaging*, vol. 22, no. 11, pp. 1341–1343, 2003.
- [163] J. Hsieh, *Computed tomography: principles, design, artifacts, and recent advances*. SPIE press, 2003.
- [164] D. S. Marcus, A. F. Fotenos, J. G. Csernansky, J. C. Morris, and R. L. Buckner, “Open access series of imaging studies: longitudinal mri data in nondemented and demented older adults,” *Journal of cognitive neuroscience*, vol. 22, no. 12, pp. 2677–2684, 2010.
- [165] C.-J. Ho, S. T. Duong, Y. Wang, C. D. T. Nguyen, B. Q. Bui, S. Q. Truong, T. Q. Nguyen, and C. An, “An unsupervised learning approach to 3d rectal mri volume registration,” *IEEE Access*, vol. 10, pp. 87 650–87 660, 2022.
- [166] D. DeTone, T. Malisiewicz, and A. Rabinovich, “Superpoint: Self-supervised interest point detection and description,” in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2018, pp. 224–236.
- [167] M. A. Fischler and R. C. Bolles, “Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography,” *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.
- [168] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, “Pytorch: An imperative style, high-performance deep learning library,” in *Advances in Neural Information Processing Systems 32*, 2019, pp. 8024–8035.
- [169] M. Tan and Q. Le, “Efficientnet: Rethinking model scaling for convolutional neural networks,” in *International conference on machine learning*. PMLR, 2019, pp. 6105–6114.
- [170] H. Rungay, M. Arnold, J. Ferlay, O. Lesi, C. J. Cabasag, J. Vignat, M. Laversanne, K. A. McGlynn, and I. Soerjomataram, “Global burden of primary liver cancer in 2020 and predictions to 2040,” *Journal of Hepatology*, vol. 77, no. 6, pp. 1598–1606, 2022.
- [171] B. Wiering, T. J. Ruers, P. F. Krabbe, H. M. Dekker, and W. J. Oyen, “Comparison of multiphase ct, fdg-pet and intra-operative ultrasound in patients with colorectal liver metastases selected for surgery,” *Annals of surgical oncology*, vol. 14, pp. 818–826, 2007.
- [172] P. S. Freitas, C. Janicas, J. Veiga, A. P. Matos, V. Herédia, and M. Ramalho, “Imaging evaluation of the liver in oncology patients: A comparison of techniques,” *World Journal of Hepatology*, vol. 13, no. 12, p. 1936, 2021.

- [173] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, “An image is worth 16x16 words: Transformers for image recognition at scale,” in *International Conference on Learning Representations*, 2020.
- [174] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, “Swin transformer: Hierarchical vision transformer using shifted windows,” in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 10 012–10 022.
- [175] Y. Chen, J. Chen, D. Wei, Y. Li, and Y. Zheng, “Octopusnet: a deep learning segmentation network for multi-modal medical images,” in *Multiscale Multimodal Medical Imaging: First International Workshop, MMMI 2019, Held in Conjunction with MICCAI 2019, Shenzhen, China, October 13, 2019, Proceedings 1*. Springer, 2020, pp. 17–25.
- [176] B. H. Menze, A. Jakab, S. Bauer, J. Kalpathy-Cramer, K. Farahani, J. Kirby, Y. Burren, N. Porz, J. Slotboom, R. Wiest, L. Lanczi, E. Gerstner, M.-A. Weber, T. Arbel, B. B. Avants, N. Ayache, P. Buendia, D. L. Collins, N. Cordier, J. J. Corso, A. Criminisi, T. Das, H. Delingette, Demiralp, C. R. Durst, M. Dojat, S. Doyle, J. Festa, F. Forbes, E. Geremia, B. Glocker, P. Golland, X. Guo, A. Hamamci, K. M. Iftekharuddin, R. Jena, N. M. John, E. Konukoglu, D. Lashkari, J. A. Mariz, R. Meier, S. Pereira, D. Precup, S. J. Price, T. R. Raviv, S. M. S. Reza, M. Ryan, D. Sarikaya, L. Schwartz, H.-C. Shin, J. Shotton, C. A. Silva, N. Sousa, N. K. Subbanna, G. Szekely, T. J. Taylor, O. M. Thomas, N. J. Tustison, G. Unal, F. Vasseur, M. Wintermark, D. H. Ye, L. Zhao, B. Zhao, D. Zikic, M. Prastawa, M. Reyes, and K. Van Leemput, “The multimodal brain tumor image segmentation benchmark (brats),” *IEEE transactions on medical imaging*, vol. 34, no. 10, pp. 1993–2024, 2014.
- [177] S. Bakas, H. Akbari, A. Sotiras, M. Bilello, M. Rozycki, J. S. Kirby, J. B. Freymann, K. Farahani, and C. Davatzikos, “Advancing the cancer genome atlas glioma mri collections with expert segmentation labels and radiomic features,” *Scientific data*, vol. 4, no. 1, pp. 1–13, 2017.
- [178] N. Heller, F. Isensee, D. Trofimova, R. Tejpaul, Z. Zhao, H. Chen, L. Wang, A. Golts, D. Khapun, D. Shats, Y. Shoshan, F. Gilboa-Solomon, Y. George, X. Yang, J. Zhang, J. Zhang, Y. Xia, M. Wu, Z. Liu, E. Walczak, S. McSweeney, R. Vasdev, C. Hornung, R. Solaiman, J. Schoepfoerster, B. Abernathy, D. Wu, S. Abdulkadir, B. Byun, J. Spriggs, G. Struyk, A. Austin, B. Simpson, M. Hagstrom, S. Virnig, J. French, N. Venkatesh, S. Chan, K. Moore, A. Jacobsen, S. Austin, M. Austin, S. Regmi, N. Papanikolopoulos, and C. Weight, “The kits21 challenge: Automatic segmentation of kidneys, renal tumors, and renal cysts in corticomedullary-phase ct,” 2023.
- [179] N. Heller, F. Isensee, K. H. Maier-Hein, X. Hou, C. Xie, F. Li, Y. Nan, G. Mu, Z. Lin, M. Han, G. Yao, Y. Gao, Y. Zhang, Y. Wang, F. Hou, J. Yang, G. Xiong, J. Tian, C. Zhong, J. Ma, J. Rickman, J. Dean, B. Stai, R. Tejpaul, M. Oestreich, P. Blake, H. Kaluzniak, S. Raza, J. Rosenberg, K. Moore, E. Walczak, Z. Rengel, Z. Edgerton, R. Vasdev, M. Peterson, S. McSweeney, S. Peterson, A. Kalapara, N. Sathianathen, N. Papanikolopoulos,

- and C. Weight, “The state of the art in kidney and kidney tumor segmentation in contrast-enhanced ct imaging: Results of the kits19 challenge,” *Medical image analysis*, vol. 67, p. 101821, 2021.
- [180] J. H. Kim, H. Y. Sun, J. Hwang, S. S. Hong, Y. J. Cho, S. W. Doo, W. J. Yang, and Y. S. Song, “Diagnostic accuracy of contrast-enhanced computed tomography and contrast-enhanced magnetic resonance imaging of small renal masses in real practice: sensitivity and specificity according to subjective radiologic interpretation,” *World Journal of Surgical Oncology*, vol. 14, pp. 1–8, 2016.
- [181] R. H. Cohan, L. S. Sherman, M. Korobkin, J. C. Bass, and I. R. Francis, “Renal masses: assessment of corticomedullary-phase and nephrographic-phase ct scans.” *Radiology*, vol. 196, no. 2, pp. 445–451, 1995.
- [182] B. A. Birnbaum, J. E. Jacobs, and P. Ramchandani, “Multiphasic renal ct: comparison of renal mass enhancement during the corticomedullary and nephrographic phases.” *Radiology*, vol. 200, no. 3, pp. 753–758, 1996.
- [183] M. Z. Alom, C. Yakopcic, M. Hasan, T. M. Taha, and V. K. Asari, “Recurrent residual u-net for medical image segmentation,” *Journal of Medical Imaging*, vol. 6, no. 1, 2019.
- [184] Q. Zhu, B. Du, B. Turkbey, P. L. Choyke, and P. Yan, “Deeply-supervised cnn for prostate segmentation,” in *2017 international joint conference on neural networks*. IEEE, 2017, pp. 178–184.
- [185] W. Luo, Y. Li, R. Urtasun, and R. Zemel, “Understanding the effective receptive field in deep convolutional neural networks,” *Advances in neural information processing systems*, vol. 29, 2016.
- [186] C. E. Shannon, “A mathematical theory of communication,” *The Bell system technical journal*, vol. 27, no. 3, pp. 379–423, 1948.
- [187] J. Cohen, *Statistical power analysis for the behavioral sciences*. Routledge, 2013.
- [188] K. An, “Sulla determinazione empirica di una legge di distribuzione,” *Giorn Dell’inst Ital Degli Att*, vol. 4, pp. 89–91, 1933.
- [189] N. Smirnov, “Table for estimating the goodness of fit of empirical distributions,” *The annals of mathematical statistics*, vol. 19, no. 2, pp. 279–281, 1948.
- [190] J. Lin, “Divergence measures based on the shannon entropy,” *IEEE Transactions on Information theory*, vol. 37, no. 1, pp. 145–151, 1991.