QATAR UNIVERSITY

COLLEGE OF ENGINEERING

DIAGNOSING LIVER'S LESIONS FOR MEDICAL ANALYSIS

BY

AYMAN JAMAL TALEB AL-KABABJI

A Thesis Submitted to

the College of Engineering

in Partial Fulfillment of the Requirements for the Degree of

Masters of Science in Electrical Engineering

January 2022

COMMITTEE PAGE

The members of the Committee approve the Thesis of
Ayman Jamal Taleb Al-Kababji defended on 23/11/2021.

Prof. Faycal Bensaali
Thesis/Dissertation Supervisor

Dr. Sarada Dakua
Thesis/Dissertation Co-Supervisor

Prof. Ridha Hamila
Committee Member

Dr. Fouad Khelifi
Committee Member

Approved:

Khalid Kamal Naji, Dean, College of Engineering

ABSTRACT

AL-KABABJI, AYMAN, J., Masters: January: 2022,

Masters of Science in Electrical Engineering

Title: Diagnosing Liver's Lesions for Medical Analysis

Supervisor of Thesis: Prof. Faycal Bensaali.

Co-Supervisor of Thesis: Dr. Sarada Prasad Dakua.

Machine learning (ML) and computer vision techniques have grown rapidly due to their automation, suitability, and ability to generate astounding results, especially the convolutional neural network (ConvNet). In this thesis, we survey the critical studies published between 2014 and 2020, showcasing the different ML algorithms researchers have used to segment the liver, hepatic-tumors, and hepatic-vasculature structures. Following that, and stemming from the surveyed literature, we propose our methodology that tackles a famous dataset named Medical Segmentation Decathlon Challenge Task 8: Hepatic Vessels (MSDC-T8), which has all the liver tissues manually segmented (liver, tumors, and vessels). This dataset is also considered the largest publicly available dataset for tackling the liver tissues delineation challenge. It encapsulates a total of 443 contrast-enhanced computerized tomography (CE-CT) scans, where the ground-truth liver masks are available for all the volumes, and the tumors and vessels segmentations are known for 303 of them. Correspondingly, this methodology is applied for each tissue of interest (TOI), as we first tackle the liver segmentation, followed by the tumors and vessels segmentation in parallel.

We compare different training environment parameters via famously used percentile and distance metrics. In our results, our liver segmentation ConvNet has surpassed the state-of-the-art performance by scoring a Dice of 98.12% on the

MSDC-T8. Moreover, the tumors and vessels segmentation ConvNets compete with the state-of-the-art, scoring a ~60% Dice for tumors’ segmentation task (with the best model scoring 65.95%) and ~50% for the vessels’ segmentation task (with the best model scoring 51.94%). Finally, when all the masks are segmented, a 3D interpolation is created for the liver (showing its tumors and blood vessels) and is exported into both .obj and .mtl files, which are 3D printing friendly. To manifest the usefulness of our work, we create a user-friendly desktop application that allows clinicians to import CT scans of selected patients. This desktop application's output is the aforementioned 3D interpolated object represented by the .obj and .mtl files.

DEDICATION

For my mom,

Who continuously worked and prayed for my success

ACKNOWLEDGMENTS

All praise and glory are directed towards Allah (God), The Almighty, for bestowing his many blessings upon me and granting me the will, the strength, and the support from people to aid and encourage me towards fulfilling this thesis.

Next, I would like to thank my supervisor Prof. Faycal Bensaali for his continuous support, guidance, and encouragement throughout my BSc and MSc programs periods. I learned a lot from him, and I am still learning. He helped me improve my technical skills and broaden my research knowledge. I would also like to thank Dr. Sarada Prasad Dakua, an academic research scientist from Hamad medical corporation (HMC), for taking the time to co-supervise my work on the thesis, pitching ideas that are insightful, and providing clinical help whenever requested.

I worked with a research team on an NPRP project, led by Prof. Bensaali, albeit not related to this one, but I would like to extend my gratitude to everyone in that team, namely Eng. Abdullah Alsalemi, Dr. Yassine Himeur, and Prof. Abbes Amira, for all the guidance and mentoring they provided me, which indirectly benefited me in my thesis project and allowed me to understand better how high-quality research is conducted.

Special thanks to my friends and brothers Anas Tahir for sharing his expertise in the computer vision world and Anas Rasras for listening to my problems and giving me pragmatic solutions. I thank them both sincerely for their continuous support.

This thesis was made possible by an Award [GSRA6-2-0521-19034] from Qatar National Research Fund (a member of Qatar Foundation). The contents herein are solely the responsibility of the authors. Moreover, the high-performance computer (HPC) resources and services were provided by the Research Computing group at Texas A&M University in Qatar. Research Computing is funded by the Qatar

Foundation for Education, Science, and Community Development (http://www.qf.org.qa).

Finally, words are not enough to express my endless gratitude to my family, especially my mother. I would not have accomplished this without their patience, support, prayers, and advice.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1: INTRODUCTION

1.1 Background

Unfortunately, liver-related diseases reap the lives of a vast number of people yearly around the globe. Approximately two million deaths around the world are credited to hepatic diseases annually [1]. Half of these deaths are related to complications caused by liver cirrhosis, and the other half are due to hepatitis and hepatocellular carcinoma (HCC) [1]. Additionally, it is also a hub for metastasis originating from adjacent organs such as the colon, rectum, pancreas, stomach, esophagus, breasts, lungs, to name some [2]. Regardless of the tumors' origin, the liver and its lesions are routinely analyzed in primary tumor staging [3]. In particular, HCC comprises a genetically and molecularly heterogeneous group of cancers commonly arising in chronically damaged livers [3].

Many imaging modalities have been developed, such as computerized tomography (CT), magnetic resonance imaging (MRI), positron-emitting tomography (PET-CT), and ultrasound (US), to enable clinicians to look inside the human body. Consequently, they have become vital biomedical tools for the liver’s morphological and volumetric analysis, along with the diagnosis of associated diseases [4], [5]. These modalities are deemed valuable, especially for their ability to provide surgeons non-invasively with insights regarding the current state of organs. With such modalities, computer-aided detection and diagnosis (CAD) systems have become significantly important. Furthermore, CT, MRI, and PET-CT can generate 2-dimensional (2D) slices of the human body, which can be combined to create 3D holistic organ volumes for surgeons to analyze. Thus, they bear more advantages than the US modality, especially in providing more informative image slices.

Moreover, the CT scans’ higher signal-to-noise ratio (SNR) and better spatial resolution produce more accurate anatomical information about the visualized structures, making it the imaging technique preferred by diagnosticians [5]. Moreover, relative to MRI, CT scans have a shorter acquisition time [4]. In contrast, the patient is more exposed to radiation in modalities like CT. Additionally, the chances of developing fatal cancers from CT scans are 1 in 2,000, which is relatively small; however, the stakes become higher with the increased number of scans [6].

As previously mentioned, these modalities allow clinicians to have a clear insight of the body organs non-invasively. CT scans, for instance, provide three different anatomical views for the organs from transversal, sagittal and coronal planes, giving clinicians the ability to tackle the organ of interest from various viewing angles, thus, examining organs holistically. Such modalities are utilized extensively by clinicians for countless clinical applications, including organic cancer diagnosis, organ transplantation, and surgical planning [7]. All these procedures are applicable in the case of the liver, where different types of cancerous cells exist, such as HCC, cysts, and metastases. Additionally, such modalities are used for adaptive radiation therapy (ART), a radiation treatment plan that imposes modifications based on the patient’s functional changes during a course of radiation [8]. In another clinical procedure, a pre-procedural CT or MRI scan can help in interventional endoscopy for pancreatic and biliary diseases as image-guidance can be supportive in intra-procedural navigation [9]. Also, medical image registration can aid clinicians in observing intra-patient organs’ motion mid-operatively [10]. Furthermore, tumor burden quantification, which measures the volume of all tumors within the liver, is essential when discussing tumors' progression within the liver. A follow-up CT scan segmenting the liver and its tumors is crucial since disease progression can be documented for further analysis and treatment procedure planning.

However, the downside becomes apparent when many patients are being treated simultaneously, resulting in an enormous amount of CT scans needed to be examined. It is time-consuming for clinicians to extract meaningful information from CT scans, let alone create a holistic delineation of an organ. Currently, the norm in clinical routines is to manually or semi-automatically segment the liver from CT and MRI modalities. In some scenarios, these techniques can be more accurate than fully-automated ones [11]; however, they suffer from subjectivity (i.e., dependency on the radiologists' experience), intra- and inter-radiologist variance, and time-consumption [12], especially for experts whose time are precious. Thus, comes the importance of using automatic methods with high segmentation performance.

Many devised automatic segmentation techniques have been applied in the last two decades. They can be categorized into statistical-based and learning-based approaches. On the one hand, the former can be represented by scans intensities' statistical distribution, including atlases, statistical shape models (SSM), active shape models (ASM), level-set methods (LSM), and graph-cut (GC) methods [13]. Usually, these methods are challenged by boundary leakage and under- or over-segmentation [11]. On the other hand, the latter depends on either hand-crafted features as in conventional machine learning (ML) algorithms or empirically found features like convolutional neural networks (ConvNet) and their deep counterpart. The medical image segmentation field made the most significant leap riding on the wave of deep ConvNets [7], where they reached a state of capability that enabled them to generate expert-like segmentations in minimal time.

However, creating an accurate segmentation of the liver, hepatic tumors, and veins is a challenging task that can be further enhanced. During data acquisition, different scanning protocols with varying voxels densities and scanners resolution and contrasting agents with unstandardized levels of contrast enhancements [3] can lead to avoidable variance within the dataset, affecting the model's performance. On the other hand, from an organs point of view, the low-contrast boundaries exhibited between the liver and surrounding organs create areas of fuzziness that are hard for models to classify, which are translated into over- or under-segmentation [12]. Moreover, the highly varying liver shapes and sizes among people, especially abnormalities introduced by surgical resection [3], make it harder for techniques, particularly SSM and similar predictors [12], to segment the liver. Challenges of segmenting what is within the liver are introduced by the heterogeneity of tumors' sizes and shapes, and intra-hepatic veins (vessels) irregularities, which further complicates the segmentation task [3], [12]. Thus, creating a detailed 3D liver segmentation portraying the exact tissues is one of the most challenging tasks that needs further enhancements.

1.2 Thesis Objectives

The objective behind this thesis can be described on multi-folds:

1) Generate a comprehensive literature review of the fully automated delineation techniques of the liver tissues where ML is fully/partially utilized.
2) Identify all the publicly available datasets that can be used for the liver tissues delineation task.
3) Implement, train, and evaluate ConvNet-based models via a high-performance computer (HPC) for the following tasks:
    a. Segmentation of the liver parenchyma.
    b. Segmentation of the tumors and vessels within the liver.
4) Generate a 3D printable object of the liver with all the delineation masks.
5) Create an extremely friendly and simple-to-use desktop application for clinicians to segment the liver from any CT scan.

1.3 Thesis Organization

The remainder of this thesis is organized as follows: Chapter 2 delivers a comprehensive review on recent studies that deployed automated techniques for liver, tumors, vessels delineation, or any combination of the three, along with varying input shapes and different datasets and challenges. In Chapter 3, a step-by-step reporting of the methodology is detailed, highlighting the utilized dataset, the image pre-processing techniques, the operated network, the evaluation metrics, and the 3D printing aspect. In Chapter 4, the experimental setup is highlighted for each of the liver tissues delineation tasks, where the acquired results are discussed and analyzed. Finally, Chapter 5 outlays the conclusions and suggests future directions related to the delineation of the liver tissues.

CHAPTER 2: LITERATURE REVIEW

This chapter analyzes and discusses the surveyed literature, highlighting possible research gaps in the last section.

2.1 Inclusion and Exclusion Criteria

For conducting this survey, the most significant studies that have been published between 2014 and 2020 have been included. Choosing this period stems from the fact that in 2014, noticeable automated ML techniques were introduced into the medical field, demonstrating their superiority to classical methods. Moreover, the focus was on Q1 and Q2 journals in Elsevier, Springer, and IEEE, along with conference papers published in the Medical Image Computing and Computer Assisted Interventions (MICCAI). Lastly, some articles from other known publishing companies such as Wiley, Hindawi, and arXiv are also included. **It is worth highlighting that the included studies used ML algorithms (partially or fully) to automatically segment one of the liver's tissues (parenchyma, tumors, or vessels)**. The papers that included user interactions anywhere in the model have been excluded as the aim is to automate the whole process.

2.2 Related Works Categorized based on Delineated Tissue

Since it is desired to automatically obtain accurate and real-time results for the liver delineation problem, it is intuitive that ML algorithms are utilized. However, providing a measure for selecting a particular ML algorithm for a specific segmentation task would undoubtedly help, especially with the technological advancements we are currently witnessing. Thus, this section categorizes the unsupervised and supervised ML algorithms based on the segmented tissue, namely liver, tumors, and blood vessels. As an output of the conducted literature survey, Figure 1 (on page 7) and Figure 2 (on page 18) have been created, detailing different ML algorithms, supervised and unsupervised, tackling the issue of liver tissues segmentation. There are notable intersections highlighted (in light brown) among the liver's three tissues of interest (TOI). Moreover, we detail the different input shapes inserted into the ConvNet and the datasets/challenges used for the liver tissues segmentation challenge.

2.2.1 Liver/Liver Parenchyma

This subsection shows all the related works found relevant to the liver parenchyma tissue segmentation. Figure 1 details all the key algorithms found in the literature.

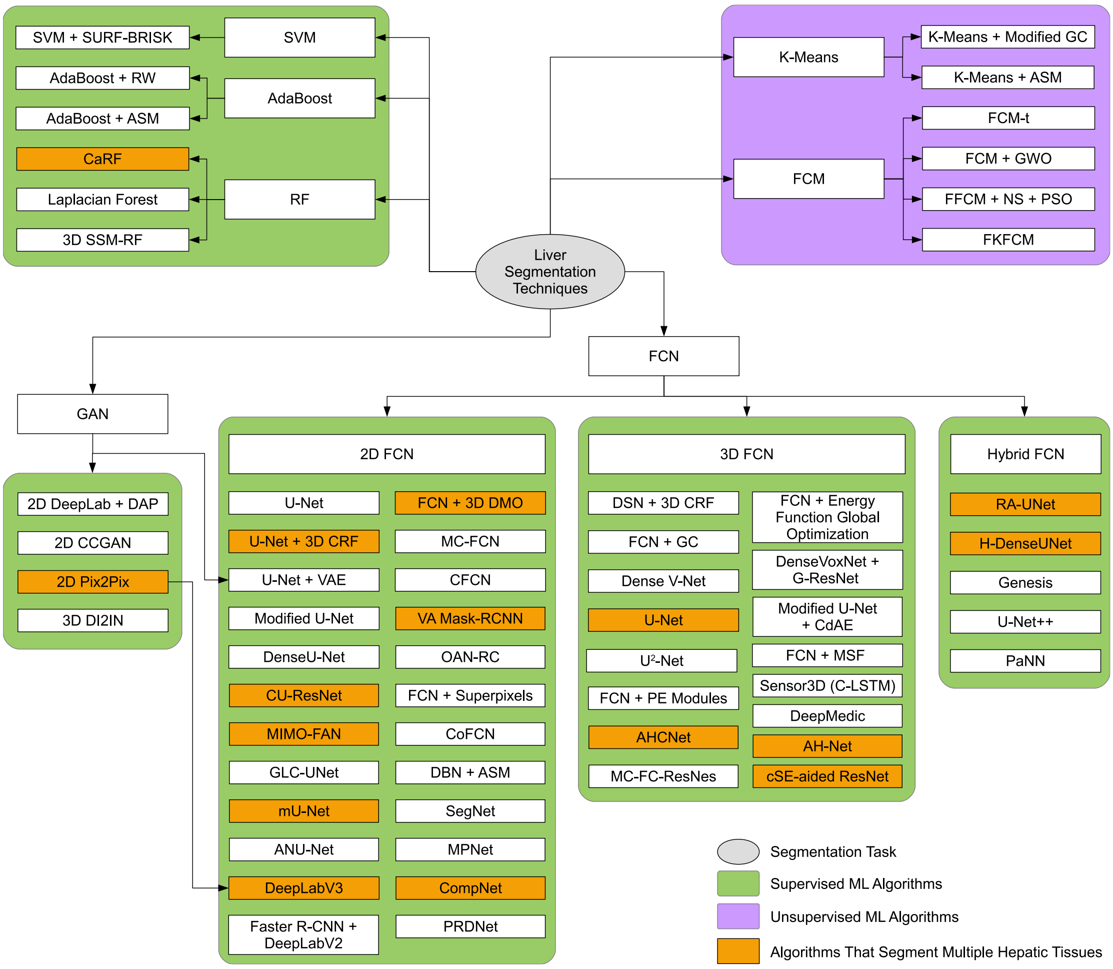


Figure 1. Liver segmentation techniques from the reviewed literature.

2.2.1.1 Miscellaneous Unsupervised

Unsupervised ML algorithms have been reasonably utilized in the liver segmentation task. They work because the liver is a single large continuous organ with relatively similar pixels' intensities in the same CT/MRI volume. The most prominent unsupervised algorithms are the k-means clustering and fuzzy c-means (FCM) clustering. The former only enables hard membership of a particular voxel to a specific clustering centroid. In contrast, the latter allows the voxel to belong to multiple centroids, with a certain probability (soft). In [14], k-means clustering is used for liver localization in CT slices as a basis for thresholding, followed by modified GC segmentation. However, in [15], a 4D k-means on multi-phase MRI volumes for liver segmentation is utilized, aided by the ASM technique. It is worth noting that vessels extraction is implemented through multi-scale vesselness filters.

On the other hand, FCM clustering in [16] calculates the degree of belonging for each voxel to three cluster classes, where one of these centroids represents the liver voxels. In [17], FCM-t provides the choice of an optimum threshold that best determines the degree of belonging a voxel should convey to be considered a liver voxel. Other researchers use FCM, or an enhanced version of it, for the liver delineation task. For instance, FCM is used with the grey wolf optimization (GWO) algorithm in [18]. At the same time, a fast version of FCM (FFCM) is utilized with neutrosophic sets (NS) and particle swarm optimization (PSO) techniques in [19]. Lastly, in [20], another modified version of FCM that is both fast and able to compute a kernel function (FKFCM) is used to segment the liver.

2.2.1.2 Miscellaneous Supervised

Laplacian forest (LF), an improved version of random forests (RF), is used in [21] for the liver segmentation task, among other organs. Other works use RF as a landmark detection tool for 3D SSM [22], while cascaded random forest (CaRF) classifiers are used for liver parenchyma segmentation in [23].

An ensemble of weak classifiers has also been used for the liver segmentation task. Adaptive boosting (AdaBoost) is used to segment the liver with the aid of random walks (RW) algorithm in [24]. A similar work uses the same combination with additional improvements on the RW algorithm in [25], and finally, a three-level ASM is guided by the AdaBoost algorithm in [26].

In [8], speeded up, robust features (SURF) (blob-type) and binary, robust, invariant, scalable key points (BRISK) (corner-type) features are used in a top-bottom flow. Moreover, the model is aided by the support vector machine (SVM) to segment the liver in the bottom-up pathway.

2.2.1.3 GAN

Generative adversarial network (GAN) is also employed for this critical task, wherein [13], a 3D deep image-to-image network (DI2IN) is used to segment the liver. On the other hand, in [27], cascaded conditional GANs (CCGAN) are used for the same task but in 2D form. In [28], a GAN network incorporates the knowledge of a deep atlas prior (DAP), where the generator that is based on DeepLab (ResNet101) is used for the liver segmentation. The discriminator is a simple 2D fully convolutional network (FCN) used to challenge the generator segmentation.

2.2.1.4 2D FCN

FCNs have become the natural choice from the pool of various ML algorithms, especially when the tackled data have a more complex form than the 1D one. FCNs are a sector of ConvNet algorithms, where fully connected layers at the end of the network are replaced by convolutional ones, reducing the number of parameters. Moreover, FCN is more appropriate for the segmentation task as the output's form of such networks is much like the input’s form. Nonetheless, FCNs are used for various problems and purposes.

In [29], a 2D FCN is utilized for liver segmentation, then used for diagnosis report generation. In [7], a 2D FCN-8s training is done via a newly-devised sample selection idea named relaxed upper confident bound (RUCB). In [30], cascaded 2D FCN (CFCN) is used for liver segmentation, where the first FCN coarsely segments the liver and the second one refines it. In [11], a 2D FCN is used for the liver segmentation, followed by a 3D deformable model optimization (3D DMO) that is based on local cumulative spectral histograms and non-negative matrix factorization (NMF).

In [31], a 2D multi-channel FCN (MC-FCN) takes six slices as input from multi-phase MRI imagery, where the used structure outperforms the U-Net on the utilized dataset. In [32], superpixels are computed, forming a map using a simple linear iterative clustering (SLIC) algorithm, and then the map is introduced into the 2D FCN to segment the liver.

An important FCN architecture that revolutionized the biomedical segmentation field is the U-Net [33], playing a similar role to the AlexNet, but for the biomedical field. Thus, it was natural for researchers to use it and inspire by it. In [3], [34], a 2D FCN following the U-Net architecture is utilized along with a 3D conditional random field (3D CRF) for liver segmentation. In [35], a 2D U-Net is used as the primary model, while SegNet is utilized in [36]. In [37], the U-Net acts as a coarse liver segmenter; however, in [38], U-Net is used as a replacement for the finite element method (FEM) to approximate the elastic deformation caused in hyperelastic objects, such as the liver. Interestingly in [39], the 2D U-Net is used to segment the liver, but the work focuses on examining whether a slice can be used to make a diagnostic decision or not.

The U-Net structure inspires other researchers, wherein [40], a 2D FCN (modified U-Net version) is employed for segmenting the liver parenchyma, excluding vessel ducts, from T1-MRI scans. In [41], 16 phases (echoes) of the same slice are generated by employing the multi-echo gradient from the MRI imaging modality. The kernels at the first layer of the 2D U-Net are modified to accept 16 slices as an input. In a similar approach, unenhanced multi-echo spoiled gradient-echo slices from MRI scans are initially used to train a 2D U-Net. It is followed by a transfer learning training step on contrast-enhanced CT scans (CE-CT) and contrast-enhanced MRI scans (CE-MRI) to segment the liver from both modalities [42]. Additionally, in [43], an ensemble of three U-Net-like 2D FCN models is used for the liver segmentation task, and the final mask is the average of those three. Authors in [44] utilized the same segmentation network as in [43], but the interest is in tumor classification. In [45], the skip connections between the encoder and decoder are modified to eliminate the redundant inclusion of low-resolution information, and the network is named modified U-Net (mU-Net). In [46], the semantic segmentation of multiple organs, including the liver, is carried via a 2D ResNet equipped with partially dilated convolutions, multiple concatenations, and fusion stages.

On the one hand, a multi-planar network (MPNet) is employed to segment the liver in any view (transversal, sagittal, or coronal) in [47], [48]. An ensemble of three MPNets is trained to segment the liver from all viewing angles in their work. The outputs of these three MPNets are fused to generate the final mask. On the other hand, in [49], a multi-planar U-Net (MPU-Net) is utilized to capture the organ of interest from different viewing angles (generalizing to more views than the three conventional ones). Similar to [47], [48], the outputs of all planar segmentation are fused to generate the final mask. In [50], a domain adaptation pipeline is created because the authors aim to create an algorithm that achieves excellent results on both CT and MRI scans. The first module is concerned with finding a common space between CT and MRI via variational autoencoders (VAEs) and GANs. The second module takes the common space output from the first module and inserts it into a 2D U-Net to segment the liver, outperforming a CycleGAN-based solution. In [51], a 2D liver extraction residual convolutional network (LER-CN), similar to U-Net architecture, is utilized to segment liver from low-dose CT scans using two main components: noise removal component (NRC) and structural preservation component (SPC).

Other researchers opt to introduce more modifications to the U-Net structure. For instance, in [52], a 2D FCN based on U-Net is equipped with ResNet dense forward connections (U-ResNet) for liver segmentation in digitally reconstructed radiographs (DRR) from X-rays created via a task-driven generative adversarial network (TD-GAN). In [53], cascaded U-ResNet (CU-ResNet) is used for liver segmentation, concatenating the middle outputs from the liver U-ResNet with the corresponding output layers in the lesions' network. The work also compares various loss functions, creating an ensemble of models incorporating different loss functions [53]. In [54], densely connected U-Net (DenseU-Net) is used for the liver segmentation task, and in [55], a comparison between DenseU-Net and atlas-based segmentation models is conducted, proving the efficacy of the former to be used in a future clinical environment. A similar architecture is utilized in [56], where the 2D FCN is based on DenseU-Net, but interestingly, utilized a shallower decoder scheme where they did not witness any reduction in the liver's (and other organs) segmentation performance. In [57], both global and local context U-Net (GLC-UNet) are used to incorporate the global and local context, which also attempts to create Couinaud segmentation of the liver. In another study, a multiple-input and multiple-output feature abstraction network (MIMO-FAN) model adapted the U-Net architecture to generate multi-scale outputs for multi-scale inputs and fuse them to achieve the final segmentation for liver in [58], and on partially labeled datasets for multiple organs in [59].

Sometimes a different backbone architecture is preferred by some of the researchers. For instance, a 2D FCN based on volume attention Mask-RCNN (VA Mask-RCNN) to incorporate volume information is employed for liver segmentation [60]. In [61], a 2D FCN based on DeepLabV3 is used for liver segmentation, followed by Pix2Pix GAN in a two-player game competition to enhance the segmentation mask. In [62], Faster R-CNN is used for liver localization, while a DeepLabV2 network is used for the segmentation. In [63], a complementary network (CompNet) is employed for the segmentation task by attempting to incorporate non-TOI voxels into the learning of TOI ones [64]. A pairwise segmentation technique for sharing supervised segmentation between two paths is investigated by the conjugate FCN (CoFCN) [65]. It takes 2.5D input and explicitly learns from adjacent slices what the segmentation mask should be. In [66], 2D deep belief network (DBN) is deployed to segment the liver, aided by ASM for post-processing refinement.

2.2.1.5 3D FCN

To further involve the volumetric information, some researchers opt for the 3D FCN, bearing in mind that 3D FCN is accompanied by expensive memory usage and high computational complexity.

One of the early works of using 3D FCNs is [67], whereas a 3D FCN is used to segment the liver and is aided by the GC algorithm. In [12], [68], the authors base their implementation over the 3D FCN employed in [67], increasing the kernels' size and modifying some of the activation functions. Then, the 3D FCN output is incorporated in an energy function, which is optimized globally. In [69], [70], a 3D FCN is equipped with a deep supervision mechanism creating a 3D deeply supervised network (3D DSN) aided by 3D CRF to refine the segmentation output as a postprocessing step. In [71], the DeepMedic network, which relies on 3D ConvNet and 3D CRF [72], is used to segment the liver for selective internal radiation therapy (SIRT). In [73], a 3D FCN with dilated convolutional layers is developed for multi-organ segmentation, including the liver. Following their work in [73], another 3D FCN based on Dense V-Net is subsequently built to segment multi-organs, including the liver [9]. The result is extended to create a framework called NiftyNet on the famous Python programming language [74], intended to make it easier to deploy biomedical segmentation algorithms. Another work utilizing Dense V-Net is [75], where a deeply self-supervised scheme based on adaptive contour features is being used for the liver segmentation task alone. In [76], a 3D anisotropic hybrid network (3D AH-Net) transforms 2D weights trained on a 2D encoder into their 3D counterpart and then is used for the liver segmentation task. In a model called Sensor3D [77], a 3D cascaded convolutional long short-term memory (C-LSTM) in a U-Net architecture theme is used for the segmentation task.

To further incorporate the global context information, authors in [78] developed project and excite (PE) modules and employed them within a 3D FCN. A similar approach is utilized in [79], where spatial squeeze and channel excitation (cSE) 3D modules aid a 3D FCN, based on ResNet architecture, in the liver segmentation task. Moreover, in [80], a 3D FCN composed of multiple attention hybrid connection blocks, hence the name (AHCNet), have densely connected long and short skip connections, and soft self-attention modules, where two cascaded AHCNets are used for liver localization and segmentation, respectively. In [81], to take advantage of the multi-phase volumes obtained by MRI scans, a multi-channel 3D FCN based on ResNet (3D MC-FC-ResNet), or 4D FC-ResNet, is used to segment the liver, utilizing the information each phase provides.

It is needless to say that the U-Net architecture also befits in the 3D context [82]. In [83], a 3D U-Net segments the liver organ, while in [84], a 3D U-JAPA-Net model has a generalized 3D U-Net serving as a pipeline for all organs and a specialized one designed for each organ. To segment organs from different modalities, a 3D universal U-net (3D U^2-Net) is built where domain-specific convolution layers are used for each modality, and a single pipeline of convolution layers is shared across the different modalities [85]. In [86], 3D patches are inserted into a 3D U-Net-like network with context-aware units for multi-phase MRI volumes in a multi-scale fashion. In [87], 3D patches were also used; however, some modifications were applied to the 3D U-Net architecture on the pooling layers, activation functions, and channels' depth. Moreover, the work takes advantage of convolutional denoising auto-encoders (CdAE) to create shape prior knowledge and embeds it into a deep data-driven loss (DDL) to enhance the segmentation result. Finally, a 3D U-Net with a multi-scale pyramid-like liver segmentation scheme is employed in [88]. It is extended in [89] to segment 20 organs in total via transfer learning from the original segmented 8 organs in the abdomen.

It is worth mentioning that some studies use FCNs as a complementary part to the core algorithm. In the case of [90], a mean shape fitting (MSF) algorithm, which creates an average shape of the liver, is complemented by a 3D FCN that generates a dense deformation field via the calculation of 3D vector of displacements for each voxel to deform the created prior as necessary. Another work analyzes the inserted CT scans via 3D FCN based on DenseVoxNet. It deforms an initial sphere mesh through

3D graph convolutions-based ResNet (G-ResNet), creating an elegant and smooth 3D mesh representation of the liver [91].

2.2.1.6 Hybrid

In [92], a 2D residual attention-aware U-Net (RA-UNet) coarsely segments the liver, then fed into a 3D RA-UNet counterpart to finely delineate the liver. In [93], both 2D and 3D DenseU-Net models, constituting a hybrid DenseU-Net (H-DenseUNet), are used for liver segmentation. In [94], a redesigned U-Net model, called U-Net++, creates an ensemble mechanism from within the architecture itself, allowing the customizability of having DenseU-Nets at various levels. It is also supported by the deep supervision technique, thus, generating outputs at all levels, serving as ensemble models. Quickly after that, in [95], the U-Net++ model is slightly modified and used for both liver segmentation and registration between pre-operative MRI and intra-operative CT scanning. In [96], attention mechanism and nested U-Net (ANU-Net) build over the 2D version of the U-Net++, where modifications are applied on the loss function and the dense connections between the nested convolutional blocks.

In [97], a prior-aware neural network (PaNN) single-handedly segments the liver, among other organs, trained over partially labeled datasets, similar to the training scheme deployed in [59]. Both 2D and 3D versions of the network are tested and compared with other available networks.

In [98], Models Genesis is a framework that can create a basis for transfer learning to any other organ segmentation via self-supervised training on unlabeled data instead of relying on ImageNet trained weights. The motivation is that the ImageNet dataset is different from the biomedical ones, creating an inappropriate transfer learning process. In their work, 2D and 3D models are initially trained on unlabeled data and then transferred for application-specific biomedical segmentation tasks.

From Figure 1, it is evident that the ML supervised algorithms of favor are the ones utilizing FCN as the main model, where they have become robust to tackle many problems (localization, registration, classification, or segmentation) in many fields. Moreover, most FCN models have utilized 2D network models with inclusion techniques for volume information such as 2.5D inputs. The 2D models are usually preferred over the 3D ones due to expensive computations and heavy memory shortages in the graphical processing units (GPUs).

2.2.2 Tumors/Lesions & Vessels

This subsection mentions all the critical algorithms used for the tumors and vessels segmentation challenges. Figure 2 has been created to highlight all the related vital studies.

2.2.2.1 Tumors/Lesions

To segment tumors, [99] initially uses a Kernelized version of FCM (KFCM), then utilizes spatial-FCM in [100] for the tumor segmentation task, followed by a 4.5C decision tree (DT) algorithm to classify segmented tumors. In [19], the combination of PSO and FCM is used for tumor segmentation, while fast FCM (FFCM) is utilized for the tumor segmentation task in [18], [101].

2D ConvNet with fully connected layers is used in [102] to segment tumors in the form of patches, testing different patch sizes for optimal performance. Patching the slices allows the model to focus on the tumor(s) instead of processing many unrelated voxels simultaneously, as tumors are generally sparse. Also, a comparison is drawn against other ML algorithms such as AdaBoost, RF, and SVM, proving the superiority of ConvNet-based techniques. In [103], a 2D FCN taking a single-phase

CT, and a 2D MC-FCN utilizing three phases of CE-CT scans, are utilized to segment tumors within the liver. Initially, both networks are trained on the liver segmentation task to enable faster convergence when trained on the tumor segmentation challenge. In [104], a DenseU-Net is utilized for the tumor segmentation task, where post-processing for object identification is based on RF to reduce FPs.

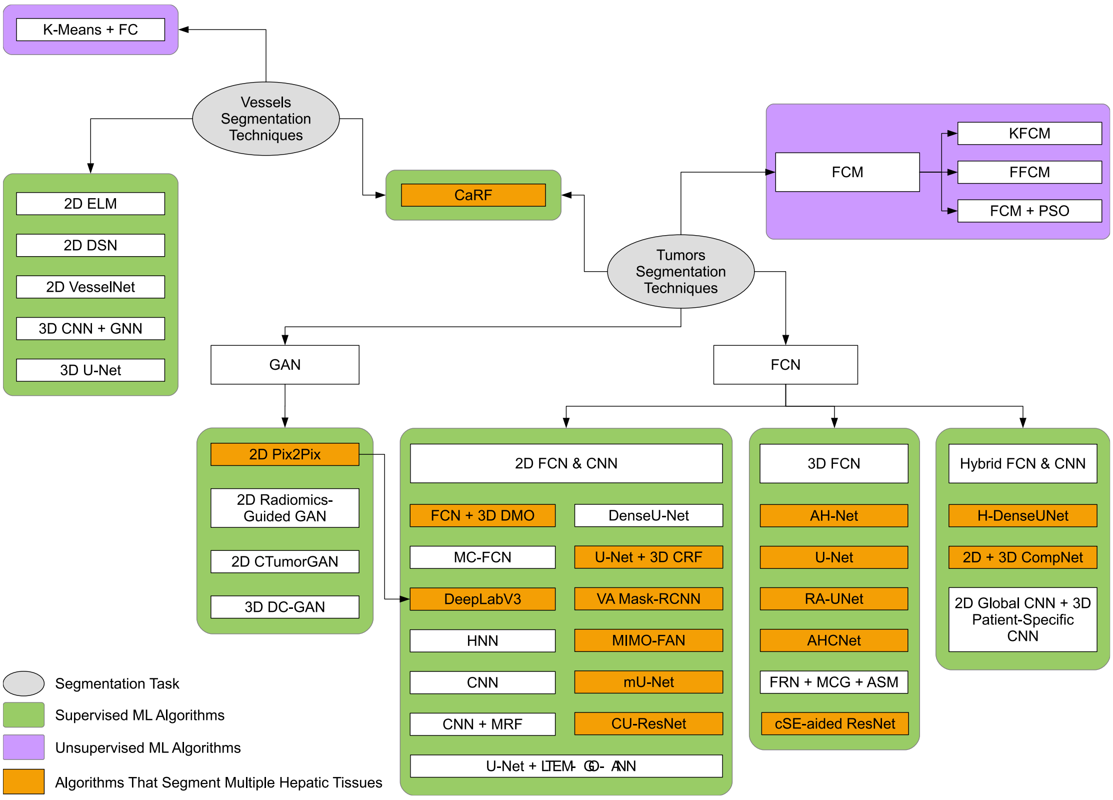


Figure 2. Tumors and vessel segmentation techniques from the reviewed literature.

An interesting approach is investigated on imprecise labeling of tumors, named “response evaluation criteria in solid tumors” (RECIST). Due to the abundance nature of this kind of data, in [105], the authors use such CT slices for the tumor segmentation via a holistic nested network (HNN), which is initially built for edge detection throughout multiple levels within the network [106]. Thus, it utilizes weakly labeled data instead of relying on pixel-wise labeling. In a similar approach in [107], a

2D FCN is used to regress biomarker information (area or volume) on CT slices to segment the tumors, instead of using manually labeled-pixels datasets, which are harder to obtain.

Some of the mentioned above liver segmentation studies use the same model for tumor segmentation as well. For example, the 2D FCN aided by the 3D DMO and NMF in [11], the 2D FCN in [29], the 2D FCN VA Mask-RCNN in [60], the DeepLabV3 followed by Pix2Pix GAN in [61], the ensemble of the three U-Net-like 2D FCN in [43], [44], the mU-Net in [45], the CU-ResNets [53], the 3D AH-Net in [76], the 3D U-Net with context-aware modules in [86], and the H-DenseUNet in [93], all of them segment the tumors along with the liver simultaneously. In [63], the first 2D CompNet, aforementioned in the liver segmentation technique, helps segment large tumors, followed by a 3D CompNet to segment the smaller ones. In [23], the same CaRF used for liver parenchyma segmentation is also used for viable tumor tissue, and necrosis tissues segmentation.

In contrast, some works opt to use an extra network for the tumor segmentation task cascadingly. In [3], [34], and after using 2D FCN to segment the liver, a cascaded 2D FCN is used to delineate the tumors within. Both FCNs are followed by the 3D CRF that refines the output of the CFCN model. Similarly, in [54], another DenseU-Net delineates tumors within the segmented liver from the first DenseU-Net. In [30], a third 2D FCN, following the first two, which segmented the liver, is used for the tumor segmentation task. Moreover, in [108], a third 3D RA-UNet segments tumors from the liver mask outputted from the first two (2D and 3D) RA-UNets networks. Similarly, a third AHCNet is used for tumor segmentation on the segmented liver from the first two AHCNets [80]. On the other hand, in [35], after segmenting the liver with a single U-Net, two other cascaded U-Nets are used for tumor (and its viability) segmentation. In [36], the authors opt for a different network than the SegNet employed for segmenting the liver. Laws texture energy measure (LTEM) features are extracted for tumor detection using a normal ANN optimized by a genetic optimizer algorithm (LTEM-GO-ANN), and then followed by a 2D U-Net performing the tumors segmentation on detected tumor regions. Similarly, in [48], after segmenting the liver using the MPNet, a 3D densely connected GAN (DC-GAN) is used for the tumor segmentation within the segmented liver. In [83], after segmenting the liver using a 3D U-Net, a multi-scale candidate generation (MCG) generates candidate tumor areas based on superpixels, which are inserted into a 3D fractal residual network (FRN), and an ASM algorithm refines the output. In [31], after segmenting the liver using a 2D MC-FCN, another network with dual-pathways is used to segment the tumors using 9-phase slices. In [37], after coarsely segmenting the liver using a 2D U-Net, a 3D FCN is used to segment the tumors within, followed by an LSM algorithm to refine the tumors segmentation. GANs are also used in the tumor segmentation task. For instance, in [109], a radiomics-guided GAN utilizes a dilated DenseU-Net as the generator (segmenter), and a VGG network as the discriminator, whereas the discriminator extracts radiomics features to aid segmenter in tumor segmentation. However, in [110], a 2D CTumorGAN is used for the tumor segmentation task in multiple organs, including the liver, and it incorporates a novel generator scheme that integrates a noise vector with the encoder part to generate segmentation masks.

2.2.2.2 Vessels

Application of unsupervised-based ML algorithms towards segmentation of vessels is rare and only available in [111], where Jerman’s vesselness filter based on K-means clustering is followed by an improved fuzzy connectedness (FC) algorithm to segment the vessels. On the other hand, the only supervised-based ML study that segments all the liver's tissues, i.e., liver parenchyma, tumors, and blood vessels, is in [23], using the same CaRF mentioned above for blood vessels segmentation.

The majority of existing studies employ supervised-based ML algorithms. In [112], an anisotropic filter is used to suppress noise and simultaneously maintain boundary details. Followed using the four filters: 1) Sato; 2) Frangi 3) offset medialness; and 4) strain energy to extract vessel features, that are then normalized. Finally, an extreme learning machine (ELM) is applied to recognize liver vessels from the background. In [113], a more ML-dependent approach is followed. A 2D DSN based on VGG-16 is used on liver vessels segmentation from US imagery. DSN has three types of layers: 1) object boundary definition prediction by fine resolution layers aided by auxiliary losses; 2) coarse resolution layers to discriminate object regions within the boundary; and 3) a trainable fusion layer. In [114], a 2D VesselNet describes an architecture that utilizes three DenseNets aimed for segmenting orthogonal patches, pre-processed by a Frangi filter, from the three planar views (transversal, sagittal, coronal). Thus, vesselness probability maps are inserted into the orthogonal DenseNets, which are then fused to generate the final segmentation mask. To incorporate the 3D context even further, some researchers developed 3D networks. In [115], a 3D U-Net is employed, which is vital in the case of tubular structures traversing narrowly through the slices. The work emphasizes the issue of data imbalance and attempts to solve it by data augmentation schemes and loss function careful design. In [116], a 3D ConvNet for vessel enhancement is used to highlight the vessel centerlines. A 3D tree tracing algorithm initializes the vessel graph tracing with high sensitivity and low specificity. Then, a graph neural network (GNN) equipped with graph attention layers (GAT) is utilized to prune the false-positive branches.

2.3 Related Works Categorized based on Input Shape

When reviewing the literature, slices are inserted into the networks in different shapes and dimensions. ConvNets, by their many forms, can accept inputs with different dimensions. The importance of discussing the input dimensions prevails when we know that these dimensions affect the ConvNet architecture, where filter sizes, convolutional layers, and pooling layers will be designed differently. This section highlights works that utilized different input dimensions, namely, 2D, 3D, 2.5D, 4D, hybrid, patches, or multi-level scaled-down slices.

2.3.1 2D Input

Originally, ConvNets were developed to work with inputs in their 2D form, i.e., images. Many studies utilize 2D inputs, where CT or MRI slices are inserted into the ConvNet slice-by-slice. Each slice is segmented in a single forward pass isolated from adjacent slices as in [34], where a 3D CRF is used to impose 3D context. If the slices were fed sequentially (for the same volume) into the ConvNet model, but with randomizing the records, the model can implicitly understand the existence of a 3D context within. However, the emphasis is weak, and this 3D context quickly dissipates if the training is randomized within volumes.

2.3.2 3D Input

3D inputs are used to involve the volumetric context in the segmentation task. It is helpful to use slices in their 3D form, meaning multiple slices are inserted into a single ConvNet, and the segmentation is carried out in one shot over all of them in the forward pass. In this manner, the network learns the 3D context, and thus, inter-slice information is preserved compared with its 2D counterpart. For instance, the famous 2D U-Net [33] and its 3D form [82] demonstrate the changes in the network architecture that follow the transition from receiving a 3D input instead. However, complications arise when using the 3D volumes; due to the limitations in the GPU memory and the accompanying heavy calculations required by the implementation of 3D ConvNets. Thus, researchers would then have to deal with patches of the original volume or a coarsely down-scaled version of it.

2.3.3 2.5D Input

2.5D is the middle-ground between the 2D and 3D inputs, where it utilizes the 3D context and information while it restrains the segmentation to a single slice in a single forward pass. The input is multiple slices adjacent to one another, creating an odd number of slices $2(k-1)$ inserted into the ConvNet model, but the segmentation is produced for the center slice only. The neighboring input slices serve only as 3D context and spatial information providers. By essence, the ConvNet is built to receive inputs of 2D nature, with multiple channels. The idea in itself is not new, it is clearly stated in [117], but many works opt for this method as it harnesses the benefits of both inputs' dimensions and disposes of their disadvantages as in [28], [43], [54], [56], [65], [93], [104], and others.

2.3.4 4D Input

As controversial as it sounds, a 4D concept comes from the MRI modality since it generates multi-phase 3D volumes of the same shape with a difference in temporal acquisition. They can be grouped when used for segmentation. In [15], the multi-phase volumes are used in 4D K-means clustering aided by active contour refinement. On the other hand, in [81], the 3D volumes are inserted into a 3D ResNet-based ConvNet, where the input is multi-channel (effectively making it 4D). However, this does not reflect architecture change as it can be resolved by channel depth design at the first layer.

2.3.5 2D & 3D Patches

Taking patches entails that the programmer extracts small portions of the slice/volume and then inputs them into the FCN to perform the segmentation. Theoretical background for why it works is provided in [118], and some of the reasons are mentioned here. For the 3D case, GPU and extensive computations are the barriers to evaluating volumes in one shot. Thus, researchers opt to take 3D chunks from the volumetric scan to process, which can have homogeneous dimensions as in [51], [86], [119], or can have heterogeneous dimensions [89]. It is worth noting that in [89], the effect of overlapping patches is also studied against non-overlapping ones. In general, and for 2D and 3D, it helps the model generalize better to unseen real-life scenarios when the segmentation is done over patches instead of the whole slice/volume. Moreover, as in [102], 2D patches are very convenient when the TOI is small, e.g., tumors and vessels within the liver, as processing the whole liver would be redundant when segmenting such tissues.

2.3.6 Multi-Scaled Input

In more recent works, researchers opt to fuse segmentations that are produced at different scales. Such algorithms take multi-scaled inputs, concatenate them, and combine them on many levels of the network to generate the segmentation mask, either in a sequential manner as in [34] or in one shot as in [58].

2.3.7 Hybrid Input (Sequential)

Hybrid input emphasizes the employment of sequential models with different dimensions, whereas the model can utilize an input with distinct dimensions compared to the previous/following one. To increase segmentation accuracy, [92] utilized a 2D network to acquire a coarse liver segmentation, which is sequentially inserted into a 3D network for segmentation refinement. Another approach is jointly using 2D and 3D networks for liver and tumor segmentation and fusing both networks' outputs [93]. Other methodology uses the first 2D ConvNet to segment the liver and the large tumors and the sequential 3D ConvNet to focus on segmenting the small ones [63]. It is worth noting that the training methodology followed in [76] initially relied on developing the weights in a 2D network, which is then extrapolated into their 3D counterpart.

2.4 Related Works Categorized based Datasets/Challenges

This section provides a historical background and a summary of the datasets/challenges' specifications found within the literature. Each dataset’s origins and specifications are discussed and then summarized in Table 2. Some of the mentioned datasets/challenges were hosted by well-known medical conferences such as the Medical Image Computing and Computer Assisted Intervention (MICCAI) and the IEEE International Symposium on Biomedical Imaging (ISBI). The organizers would present the datasets as challenges pushing researchers to participate by creating healthy peer-pressure environments. Other datasets were shared publicly by different research institutions to encourage researchers to develop better algorithms[1].

2.4.1 Historical Background about Datasets/Challenges

This thesis comprehensively highlights and presents datasets that include liver, tumors, or vessels delineations, solely or with other organs' ground-truth masks. However, if the ground-truth labels for either liver, tumors, or blood vessels were not included, we refrain from adding that particular dataset. Lastly, we define ground-truth labels as the delineation (i.e., segmentation) of tissues, not the localization of such tissues.

[1] Some of the datasets were investigated with the aid of ITK-SNAP [150]. Available at: www.itksnap.org

2.4.1.1 Segmentation of the LIVER Competition [2007] (SLIVER07)

In October, the SLIVER07 competition took place in a workshop named “3D Segmentation in the Clinic: A Grand Challenge” in conjunction with MICCAI 2007. The results of that workshop are summarized in [120]. The dataset has 30 CE-CT scans divided into 20 training volumes and 10 testing volumes. The intra-slice resolution varies between 0.54 and 0.86 mm, while the inter-slice space varies between 0.5 and 5 mm. The number of pixels is the same for all slices within all volumes (512×512), with varying slices between 64 and 502.

2.4.1.2 3D Image Reconstruction for Comparison of Algorithm Database (3D-IRCADb) [≤ 2010]

3D-IRCADb is a database gathered by the IRCAD institute in France, where it includes anonymized medical images of patients. In total, the dataset has 22 venous phase CE-CT scans divided into: 1) 3D-IRCADb-01 that contains 10 male and 10 female with 75% having hepatic tumors; 2) 3D-IRCADb-02, which contains 2 CT scans with other abdominal organs segmented. The entire dataset's intra-slice resolution varies between 0.56 and 0.96 mm, while the inter-slice distance varies between 1 and 4 mm. On the other hand, (512×512) pixels are used per slice, while the number of slices ranges between 74 and 260. It is worth noting that the majority of literature focuses on the 3D-IRCADb-01 part and is typically divided into training and testing records accordingly.

2.4.1.3 MIDAS Liver Tumor (MIDAS-LT) Segmentation Dataset [2010]

MIDAS-LT, an acronym we created, is a part of a bigger initiative to provide a collection of archived, analyzed, and publicly accessed datasets called MIDAS [121]. The MIDAS-LT is funded by the National Library of Medicine (NLM) in the USA under the Imaging Methods Assessment and Reporting (IMAR) project. The dataset contains 4 CT scans with (up to) 3 radiologists’ manual segmentation for liver tumors per volume, without a mask of the liver. All the dimensions (inter- and intra-slice) vary between 1.73 and 1.85 mm. On the other hand, pixels in a slice vary between 177 and 189, while the number of slices ranges between 98 and 259. It is worth noting that the original dataset had homogeneous specifications, but the dimensions reported here are for the segmented volumes.

2.4.1.4 Multi-Atlas Labeling Beyond the Cranial Vault (BtCV) - Workshop and Challenge [2015]

This workshop is the last of a series of workshops introduced by Landman et al. from Vanderbilt University [122] hosted by MICCAI. This workshop aims to extend multi-atlas segmentation beyond the skull vault to include the cervix and abdomen segmentation. Thus, the liver and other organs (such as kidneys, gallbladder, esophagus, and stomach) are segmented. The dataset contains many organs' segmentations and registrations. Still, the number of liver segmentation records is 50 venous phase CE-CT scans, divided into 30 training and 20 testing records found under the `RawData' file. The intra-slice resolution is between 0.54 and 0.98 mm, while the inter-slice distance is within the 2.5 to 5 mm range. In contrast, the number of pixels is 512×512, with varying slices between 85 and 198.

2.4.1.5 Pancreas-CT [2015]

Pancreas-CT is a portal venous CE-CT dataset that contains pancreas manual delineations available on The Cancer Imaging Archive (TCIA) website [123]. It originally had 82 records when first published in 2015. However, in 2020, 2 volumes were removed (#25 and #70) as they were duplicates of Record #2 with slight variations. The intra-slice resolution varies between 0.66 and 0.98 mm for the voxels' physical dimensions, while the inter-slice distance ranges between 0.5 to 1 mm. On the other hand, the number of pixels is 512×512, with the number of slices being between 181 and 466. This dataset is included because the liver delineations are created along with other organs for 43 records (42 after removing the duplicate Record #25) from this dataset [9]. It is worth noting that the specifications mentioned in Table 2 are for those records with liver delineation provided in [9]. These records are representative of all the dimensions of the original dataset; however, the minimum number of slices is 186 instead.

2.4.1.6 Visual Concept Extraction Challenge in Radiology (VISCERAL) Anatomy3 [2016]

The challenge took place in three consecutive years (2014 - 2016) in conjunction with the ISBI. The challenge targets multi-organ segmentation by providing ground-truth labels for up to 20 organs (liver, pancreas, spleen, kidneys, lungs, aorta, urinary bladder, gallbladder, to name some). The complete list can be found in [124], and the results of all the workshops are summarized in [125]. The dataset has 120 records, from CT and MRI modalities, with and without contrast-enhancing agents (refer to Table 2 for further details). The intra-slice resolution varies between 0.60 and 1.40 mm for the CT records, while the inter-slice distance is fixed to 3 mm [125]. The slice resolution is between 0.84 and 1.30 mm for the MRI scans, while the inter-slice depth varies between 3 and 8 mm [125].

2.4.1.7 Liver Tumor Segmentation Challenge (LiTS) [2017]

This challenge was conducted in ISBI (18/04/2017) and MICCAI (14/09/2017) to provide researchers with ground-truth labels for the liver and tumors within. The challenge is to automatically segment liver tumors/lesions in CT volumes and estimate tumors' burden, along with the typical liver segmentation challenge. The dataset has 201 CE-CT records in total, divided into 131 training and 70 testing scans. The dataset can be found in [126], and the summary of the challenge results is in [127]. Noting that 3D-IRCADb-01 is part of the training set of LiTS (Records 28 - 47 [80]), care must be taken when both 3D-IRCADb and LiTS datasets are used to train the model. Using them both for training will make the model biased towards the common records since it is trained on the duplicate records twice every epoch. Also, it is inappropriate to train on the entire training set of LiTS and test on 3D-IRCADb, as the testing set would be exposed to the model in the training phase beforehand. For the physical dimensions, the intra-slice resolution varies between 0.55 and 1.00 mm, while the inter-slice distance ranges between 0.45 to 6.0 mm [127]. In contrast, the number of pixels is 512×512, with slices ranging between 42 and 1026.

2.4.1.8 Medical Segmentation Decathlon Challenge (MSDC) [2018]

MSDC was held in MICCAI 2018, where it uniquely focuses on the segmentation generalizability of a model on 10 different biomedical challenges. In this review, we only report the liver-related tasks, namely, Task 3 and Task 8. Nonetheless, details regarding all the tasks are summarized in [128]. We refer to Task 3 and Task 8 datasets as MSDC-T3 and MSDC-T8, respectively.

As a matter of fact, MSDC-T3 is the same as the LiTS dataset, where the training sets are identical, but the testing set in MSDC-T3 is shuffled compared to its counterpart in LiTS. On the other hand, the MSDC-T8 dataset contains 443 portal venous phase CE-CT scans with segmented tumors and vessels only, where 303 are designated as training and the remaining 140 records as testing. The intra-slice resolution varies between 0.56 and 0.98 mm, while the inter-slice distance ranges between 0.80 and 8 mm. The slices have the standard number of pixels for CT scan (512×512), with a varying number of slices between 24 and 251. It is worth noting that the authors in [57] created the liver annotations within the MSDC-T8 443 CT records and shared them publicly. Additionally, Couinaud's segmentation of 193 livers among the 443 records is also shared.

2.4.1.9 CT Volumes with Multiple Organ Segmentations (CT-ORG) Dataset [2019]

CT-ORG is an extension of the LiTS dataset, and it is publicly accessible via the TCIA website [129]. It contains 140 CT scans where the creators [130] added 9 extra PET-CT scans over the LiTS training set and extended the segmentation to multiple organs (lungs, bones, liver, kidneys, bladder, and brain). Most provided segmentations are golden-corpus (manually labeled), while lungs and bones in the training set are silver-corpus (automatically segmented). The intra-slice resolution varies between 0.55 and 1.37 mm for the voxels' physical dimensions, while the inter-slice distance ranges from 0.7 to 5 mm [127]. On the other hand, the number of pixels is 512×512, with varying slices between 74 and 987. The difference between LiTS dimensions mentioned earlier and CT-ORG is contributed to the testing set of LiTS, which is not included in CT-ORG. We have verified this difference by developing a Python code to find the minimum and maximum of each quantity in both datasets.

2.4.1.10 Combined (CT-MR) Healthy Abdominal Organ Segmentation (CHAOS) Challenge [2019]

CHAOS was held in ISBI 2019, aiming to segment abdominal multi-organ tumor-free CT and MRI data. The dataset has both CT and MRI (T1 and T2 weighted) parts, where there is no inter-modality connection (i.e., the CT and MRI data are from random patients, not counterparts for the same patient). The summary of this challenge is reported in [131].

The CT dataset contains 40 CE-CT records for patients with the ground-truth label for healthy livers (potential liver donors) acquired at the portal venous phase. The intra-slice resolution varies between 0.7 and 0.8 mm, while the inter-slice space varies between 3 and 3.2 mm [132]. The resolution is similar to other datasets (512×512), and the number of slices ranges between 77 and 105 [132]. On the other hand, the MRI dataset has ground-truth labeling for the liver, kidneys, and spleen, containing 120 records of both 80 T1-Dual (40 in-phase and 40 out-phase) and 40 T2-SPIR weighted records. The MRI records from different enhancing protocols are for the same patient. For instance, patient 20 has three MRI records falling into the three previously mentioned categories. Intra-slice resolution varies between 1.36 and 1.89 mm, while the inter-slice distance is bounded between 5.5 and 9 mm [132]. It is worth noting that the resolution here is different (256×256), and the number of slices varies between 26 and 50 [132]. The intra-modality acquisition protocol in this dataset is consistent where we see minor variations between different records belonging to the same modality. Further details can be found in [132], including the used acquisition devices and the different contrast-enhancing phases for MRI.

2.4.2 Summary of Challenges/Datasets

Table 1 shows the pros and cons each dataset has relative to other available datasets. Moreover, Table 2 summarizes reviewed challenges and datasets.

Table 1. Advantages and limitations of the available datasets.

Dataset	Advantages	Limitations
SLIVER07	- The earliest publicly available dataset to have liver masks	- Small size - Not challenging anymore - Does not have liver tumors and vessels
3D-IRCADb	- First dataset to include liver tumors - The records metadata are mentioned for each record	- Small size - The majority do not have liver vessels
MIDAS-LT	- N/A	- Minimal size - Very few segmented tumors
BtCV	- Only dataset to have many segmented organs (13 ground-truth organs)	- Medium size
Pancreas-CT	- It has Pancreas segmented as well along with liver masks - High resolution	- Medium size - Does not have liver tumors and vessels

Dataset	Advantages	Limitations
VISCERAL Anatomy3	- Multiple modalities - Large size, the first dataset to have more than 100 ground-truth labeling	- Hard to access - Does not have liver tumors and vessels
LiTS	- Large size - Many tumors are segmented in 131 records	- Does not have liver vessels
MSDC-T8	- Largest abdominal dataset with manual segmentation - Contains all liver tissues (liver, tumors, vessels)	- N/A
CT-ORG	- Builds up over LiTS and adds 9 more records - Has other organs segmented	- Does not have liver vessels
CHAOS	- Multiple modalities - Large size	- Does not have liver tumors and vessels

Table 2 highlights the website where researchers can retrieve the datasets, the type of modality used, the inclusion of contrast enhancement agents, the available masks, the overall dataset size, and the voxels' dimensions.

Table 2. Summary of available datasets.

Dataset	Modality	Available Masks	Size (Train/Test)	Voxels Dimensions (Height × Width × Depth) mm^3
SLIVER07 [133]	CE-CT	Liver	30 (20/10)	(0.54 ~ 0.86) × (0.54 ~ 0.86) × (0.5 ~ 5)
3D-IRCADb [134]	CE-CT	Liver (Tumors, Vessels)	22 (N/A)	(0.56 ~ 0.96) × (0.56 ~ 0.96) × (1 ~ 4)
MIDAS-LT [135]	CT	Tumors in Liver	4 (N/A)	(1.73 ~ 1.85) × (1.73 ~ 1.85) × (1.73 ~ 1.85)
BtCV [122]	CE-CT	Liver & Others	50 (30/20)	(0.54 ~ 0.98) × (0.54 ~ 0.98) × (2.5 ~ 5)
Pancreas-CT [123]	CE-CT	Pancreas [123], Liver & Others [9]	42 (N/A)	(0.66 ~ 0.98) × (0.66 ~ 0.98) × (0.5 ~ 1)
VISCERAL Anatomy3 [124]	CT	Liver & Others	30 (20/10)	(0.97 ~ 1.40) × (0.97 ~ 1.40) × 3
	CE-CT	Liver & Others	30 (20/10)	(0.60 ~ 0.79) × (0.60 ~ 0.79) × 3

Dataset	Modality	Available Masks	Size (Train/Test)	Voxels Dimensions (Height × Width × Depth) mm^3
	MRI	Liver & Others	30 (20/10)	1.25 × 1.25 × 5
	CE-MRI	Liver & Others	30 (20/10)	(0.84 ~ 1.30) × (0.84 ~ 1.30) × (3 ~ 8)
LiTS [126]	CE-CT	Liver (Tumors)	201 (131/70)	(0.55 ~ 1.00) × (0.55 ~ 1.00) × (0.45 ~ 6)
MSDC-T8 [136]	CE-CT	Liver [57] (Tumors & Vessels [128]	443 (303/140)	(0.56 ~ 0.97) × (0.56 ~ 0.97) × (0.8 ~ 8)
CT-ORG [130]	CT, CE-CT, PET-CT	Liver (Tumors) & Others	140 (119/21)	(0.55 ~ 1.37) × (0.55 ~ 1.37) × (0.7 ~ 5)
CHAOS [132]	CE-CT	Liver	40 (20/20)	(0.70 ~ 0.80) × (0.70 ~ 0.80) × (3 ~ 3.2)
	MRI	Liver, Kidneys & Spleen	120 (60/60)	(1.36 ~ 1.89) × (1.36 ~ 1.89) × (5.5 ~ 9)

2.5 Outlook

Many insights can be drawn from the above survey: 1) Most studies in the liver delineation task utilize supervised ML algorithms, especially the 2D and 3D FCN-based models as depicted by Figure 1, the reason behind this phenomenon is the powerfulness of ConvNets in automatically extracting the discriminative features, through the trained filters' coefficients, to optimally reduce the objective function, instead of having to manually extract those features, which include untested assumptions in a trial-and-error process; 2) The vast advancements in the ML field to tackle the biomedical problems, like using 3D or 2.5D instead of 2D to fathom the liver's 3D complex structure; 3) The usage of different algorithms aiming to segment multiple tissues, showing an initiative towards creating a complete algorithm for full liver delineation; 4) The number of studies investigating tumors and vessels delineation, shown in Figure 2, are low when compared with the ones exploring the liver's delineation, especially the vessels segmentation studies that are rare, pointing at a research area worth further investigation; and 5) The severe absence of studies that tackle all liver's different tissues delineation problem.

Moreover, from a computational complexity point of view, unsupervised ML techniques are more expensive than their supervised peers [137]. It is attributed to the fact that unsupervised techniques typically need extensive data to generate significant, meaningful clustering [137]. In contrast, the supervised ones can generate good results from a small dataset if carefully designed [137]. Moreover, with labels' existence finding the optimal solution becomes a guided training process, unlike unsupervised techniques where the learning process is blind [137]. Hence, supervised learning algorithms can generate better results due to the manual labeling applied to the dataset, allowing the machines to learn from the humans' expertise.

Due to the absence of a large number of slices in earlier datasets, some authors opted to augment the data, as in [36], by transitioning the slices 10 pixels in each direction and then horizontally flipping all the slices to generate a new dataset that is 10 times bigger. It is worth noting that they also applied bilateral filters to remove granular noise [36]. Others opt to finely rotate the slices by 5 – 10^{o} to generate new slices to train the algorithm further and allow it to generalize better to unseen data, where the human body is slightly rotated.

CHAPTER 3: METHODOLOGY

In this chapter, the followed methodology is explained with in-depth details, easing the process of reproducibility. It goes through the used datasets, the image pre-processing techniques, the training environment parameters, the input shapes, the neural network models, the evaluation metrics, and the 3D construction and printing.

It is worth mapping out the schema of our system to tackle the liver, tumors, and vessels segmentation challenges in Figure 3. It shows the liver segmentation being tackled by the first ConvNet. The first ConvNet's output is multiplied by the original CT slices to keep the liver's voxels and remove other voxels considered as background. Then, the CT slices, with liver voxels only, are fed into the tumors and vessels ConvNets in parallel to generate these tissues masks. Lastly, the output of the three ConvNets is then combined and interpolated to create the 3D object of the liver, ready to be visualized and printed.

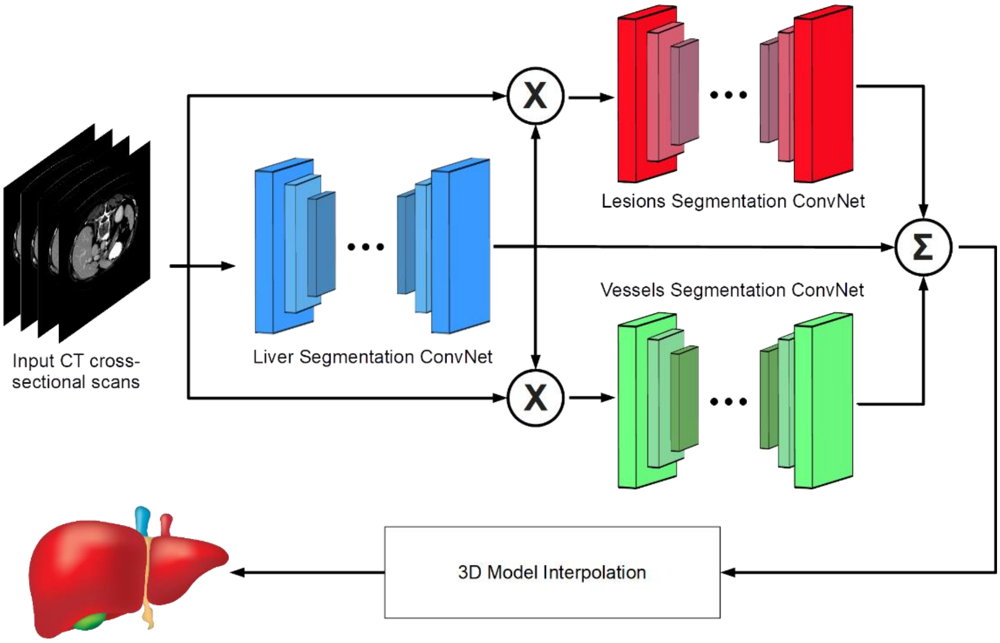

Lesions Segmentation ConvNet
Input CT cross-sectional scans
Liver Segmentation ConvNet
Vessels Segmentation ConvNet
X
X
Σ
3D Model Interpolation

Figure 3. ConvNets sequential training scheme.

One important point to highlight is that the created code used in this thesis was backed up using a GitHub repository, where anyone can freely start a private/public repository. Although it is used primarily by collaborators on the same project, it is also an excellent tool for having cloud-backup code, along with a Version Control feature allowing version reversion.

The followed methodology is summarized in Figure 4, highlighting different aspects, including 1) utilized dataset; 2) data pre-processing; 3) input shape; 4) downloaded network models; 5) loss functions; and 6) evaluation metrics.

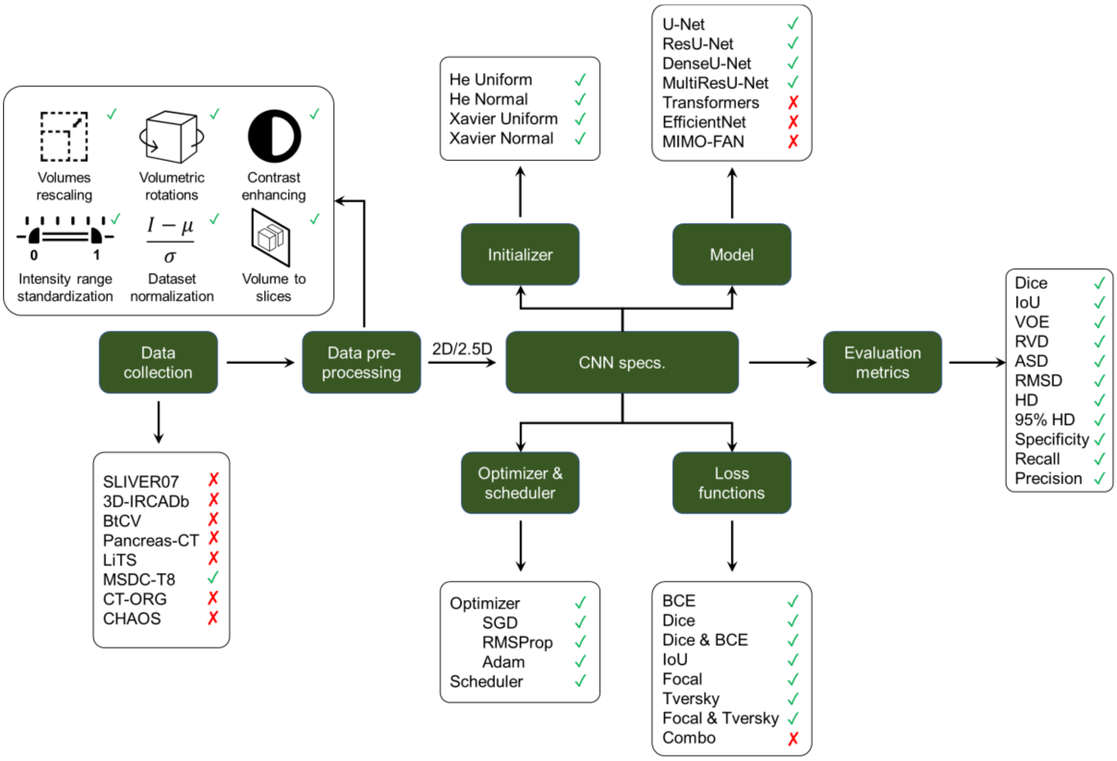


Figure 4. Summary of the followed methodology.

3.1 Utilized Datasets

This thesis aims to segment the liver parenchyma, tumors, and vessels; therefore, a complete dataset containing the manual segmentation for all three tissues must be attained. From the conducted literature review, it is evident that the dataset qualifying for this role is the MSDC-T8 dataset. To re-iterate, it contains 443 CE-CT scans of a varying number of slices and voxels' dimensions. The liver ground-truth labels are provided in [57], while the tumors and vessels manual masks are provided by the challenge organizers in [128]. The liver masks are publicly available for the 443 records; however, the tumors and vessels masks are known for 303 only. Moreover, it is worth noting that the dataset mainly focuses on the human body's abdominal part. The goal of this dataset is primarily to challenge the participants to delineate the tumors and vessels within.

3.2 Data Pre-processing

Once the dataset has been chosen, a set of pre-processing techniques has been selected to boost the deployed convolutional neural network (ConvNets).

3.2.1 Volumetric Rotation

CT scans come in various orientations, given that different hospitals can have different protocols. Fortunately, the orientation information can be found in the Neuroimaging Informatics Technology Initiative (NIfTI) files' metadata. CT scans, along with their masks, are made to face the same direction (to the right) so that avoidable variability is discarded, instead of dealing with the most generic case of having CT scans of patients facing different directions.

3.2.2 Volume Rescaling

A rescaling of ½ is applied to the input CT scans because ConvNets like U-Net have many weight parameters, and after each batch feedforward iteration, a backpropagation process is commenced. Thus, it is highly encouraged to downsample the inputs' sizes to reduce the computational complexity. A 2D input size will be 512×512 (263,680) for every single input if no rescaling is applied, while it is 256×256 (65,536) with rescaling (i.e., a reduction in the input size by a factor of ~¾).

It is worth noting that rescaling is applied to all the training, validation, and testing volumes and masks. However, downsampling is not applied on the testing masks because the resulting segmentation masks are re-scaled to the original shape 512×512. By doing so, we intend to make the comparison as close as possible to the clinical environment. Moreover, the 3D object creation has better resolution when more voxels are involved.

3.2.3 Intensity Clipping

According to Yuan [30], voxels' intensities are clipped to be in the [-100, 400] range to segment the liver and tumors within. Yuan [30] argues that all the liver voxels' intensity values are retained in that range. Clipping is implemented as illustrated by equation (1):

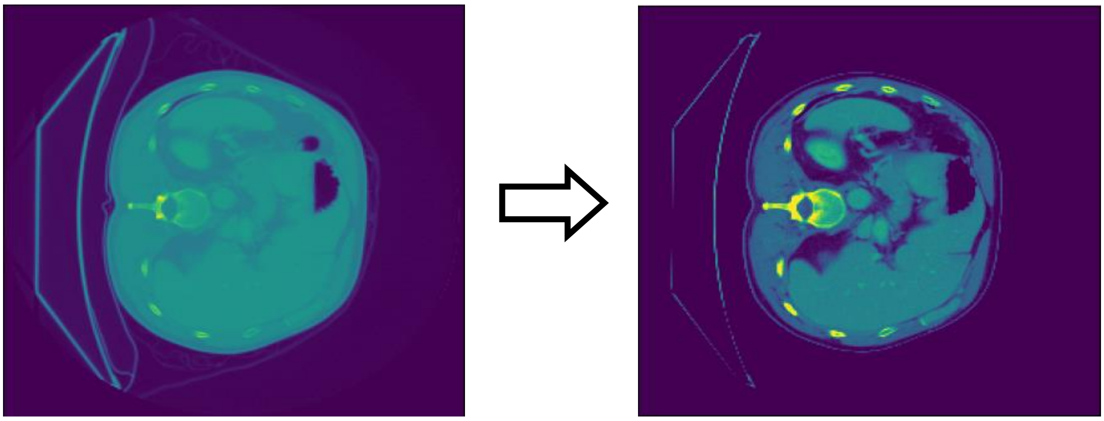
$$V_{(x,y,z)} : \begin{cases} V_{(x,y,z)} = -100 & if\ V_{(x,y,z)} < -100 \\ V_{(x,y,z)} = V_{(x,y,z)} & if -100 < V_{(x,y,z)} < 400 \\ V_{(x,y,z)} = 400 & if\ 400 < V_{(x,y,z)} \end{cases} \tag{1}$$

where $V_{(x,y,z)}$ represents the intensity value for volume V at the x, y, and z coordinates.

Figure 5 shows the effect that clipping has on the CT scan voxels, as it increases the contrast between the liver and its surroundings.

3.2.4 Intensity Range Standardization

Large datasets, like the MSDC-T8, have volumes with varying upper and lower bounds for voxels' intensities. As an example, some of the volumes have voxels' that range between -1100 and 3024, while others can be in the range of -1000 to 1200. This issue can be traced back to the employment of different machines to create these files; hence, they can be using different standards. Nonetheless, all the volumes are standardized to have an identical floating range that varies between 0 and 1, which is achieved by implementing equation (2):



a) Before clipping b) After clipping

Figure 5. Cross-sectional slice from a CT scan during clipping.

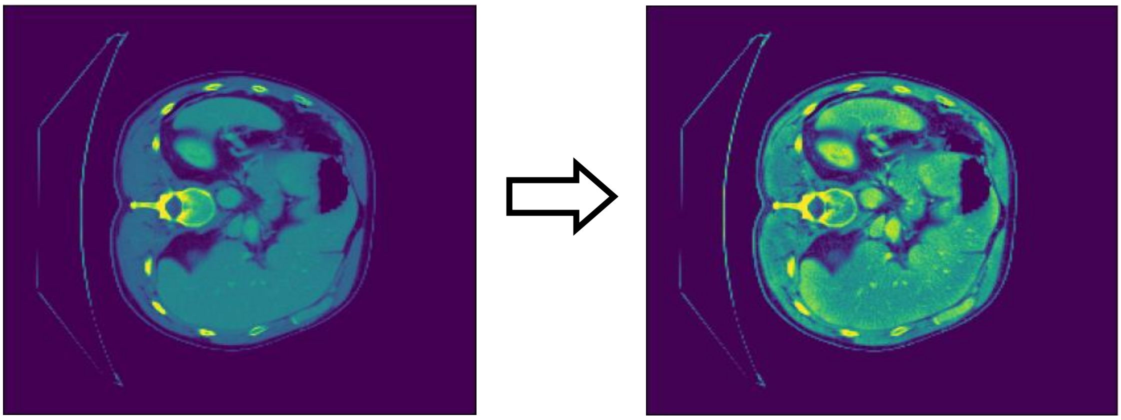
$$V_{(x,y,z)} = \frac{V_{(x,y,z)} - \min(V_{(x,y,z)})}{\max(V_{(x,y,z)}) - \min(V_{(x,y,z)})} \tag{2}$$

However, because clipping is applied earlier, the range of voxels’ values has become within the [-100, 400] range. Thus, it becomes straightforward to subtract -100 and divide by 500 for all the voxels by equation (2).

3.2.5 Contrast Enhancement via CLAHE

Contrast limited adaptive histogram equalization (CLAHE) is used to enhance the inter-organs borders, as well as the intra-liver borders (parenchyma, tumors, and blood vessels). Ordinary histogram equalization works well when the distribution of pixels/voxels' values is similar across the whole image/volume. However, when the image contains significantly lighter or darker regions relative to other parts of the image, CLAHE performs better as the histogram equalization for a pixel/voxel is calculated based on the neighboring pixels/voxels instead of the whole image.

Figure 6 shows the effect CLAHE carries on a CT slice by displaying the slice before and after applying CLAHE on a 3D volume. Figure 6 shows a cross-sectional slice from the actual volume. Moreover, when the CLAHE algorithm is applied, it is applied in a 3D manner, meaning that the neighboring voxels from adjacent slices are considered to apply the local histogram equalization. For documentation purposes, it is worth noting that the slice shown in Figure 5 and Figure 6 is slice 25 in hepaticvessel_001 record from the MSDC-T8 dataset.



a) Before applying CLAHE

b) After applying CLAHE

Figure 6. Cross-sectional slice from a CT scan during applying CLAHE.

3.2.6 Volume Normalization

Moreover, normalization is applied to accelerate the training process. Firstly, the mean and the standard deviation are calculated over all the voxels in all the volumes. Then, the mean is subtracted, and the standard deviation is divided by as shown by equation (3):

$$V_{(x,y,z)} = \frac{V_{(x,y,z)} - \mu_V}{\sigma_V} \tag{3}$$

where μ_V and σ_V are the mean and standard deviation, respectively, of all the voxels found in all the volumes.

3.2.7 Volume Slicing

Lastly, before feeding volumes into the ConvNet, data are transformed from their initial state (3D volumes) into either 2D or 2.5D input. This transformation is done by importing the NIfTI files/volumes and saving them as a Tag Image File

Format (TIFF). The reason for using TIFF instead of other available formats is because it retains the floating points value and saves the files without any loss of information. Moreover, it is possible to save multiple images into the same file (creating a 3D image), making it an essential feature for data preparation, especially for creating 2.5D inputs into the ConvNet.

3.3 Input Shape

After completing the volumes' pre-processing stage, another stage follows, correctly ordering the images/slices to be inserted into the ConvNet. An exciting idea mentioned in the literature regarding the input for 2D ConvNets is stacking the adjacent slices as extra channels (as with RGB pictures) to be inserted into the 2D ConvNet. The reason behind shaping the input in this manner is to provide the ConvNet with volumetric context, which is extremely important in volumes segmentation. The natural alternative is to use 3D ConvNets, which are memory and computationally expensive, to learn the volumetric context. Hence, it is favored to use the 2.5D technique with a 2D ConvNet instead. Figure 7 shows how the stacking of adjacent slices is done.

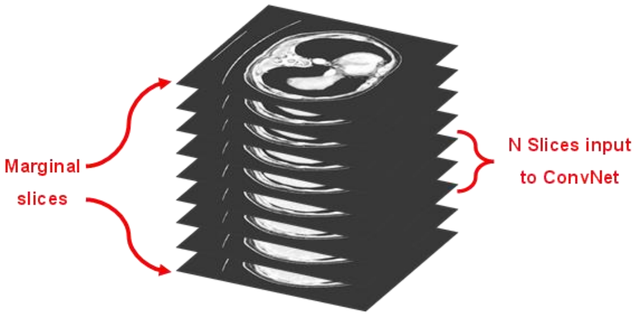


Figure 7. Creating 2.5D input.

Moreover, a question would arise on how marginal slices are being treated. One possible solution is to pad zeros (empty slices) above the top marginal slice and below the bottom marginal slice. Another alternative is to replicate the marginal slices $\frac{k-1}{2}$ times, where k is the number of slices that will be put together into the ConvNet. The second approach is less erroneous as the ConvNet is still fed CT-related slices, unlike the first approach where blank slices are given to the ConvNet, which is further from reality than the second alternative. In this thesis, 5 slices (2.5D input) are inserted into the ConvNet, and hence, the marginal slices must be repeated 2 times to be able to use all the slices within the volume in the training process.

The best file format to save the preprocessed slices in their 2.5D shape to be easily imported during the training phase is the TIFF files. As mentioned earlier, the file format saves 3D arrays into a single file, rendering the data loading of training and testing samples much more effortless. Figure 8 shows how the training samples are saved within the MSDC-T8.

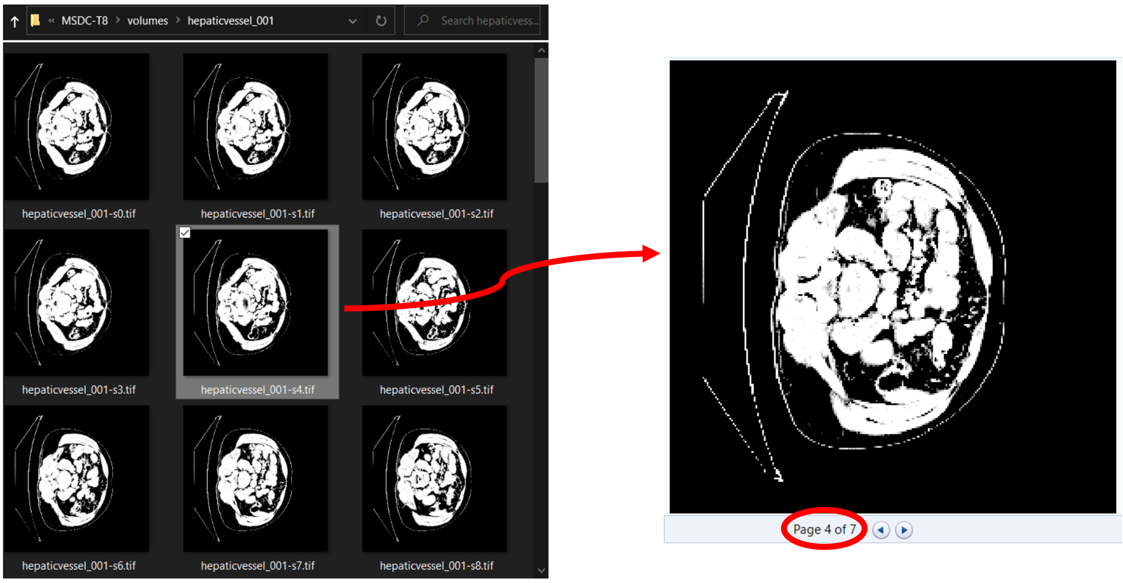


Figure 8. 2.5D training sample storage.

Once opened, the user can see there are 7 adjacent slices/images within each training sample. The reason for making it 7 is to allow the variability for the programmer to choose an input shape that is within {1, 3, 5, 7} without having to do the pre-processing phase again. It is worth noting that the slices look black and white because the software shows them this way; however, the values range in ~[-1, 5] due to the applied normalization from equation (3). The 4th slice (middle slice) is always the targeted training sample, and 1 → 3 and 5 → 7 are the adjacent slices. In the results chapter, a comparison of the most suitable input shape for the liver segmentation is also conducted.

3.4 High-Performance Computer (HPC) Specifications

After defining the CT slices' pre-processing steps, it is vital to mention the platform where these processes occur. This thesis uses an HPC provided by Texas A&M in Qatar (TAMUQ) to pre-process the selected dataset and create ConvNet models trained on the 3 TOIs. The HPC flagship system is called “Raad2” (رعد2 in Arabic), translating into Thunder2 in English. The name is befitting as it contains 4,128 cores from Intel Xeon Gold central processing units (CPUs), distributed over 172 computing nodes (i.e., 24 cores per computing), and 128 Gigabyte (GB) of RAM per computing node [138]. The usable capacity of this HPC is 800 Terabyte, where 500GB are given for each user. Moreover, it uses the SLURM workload manager as the medium to submit jobs for the HPC to process. Additionally, the state-of-the-art NVIDIA Volta V100 16GB VRAM GPU, along with Intel Xeon Skylake CPUs, can be used to run a job, with multiple CUDA versions installed [10.1, 10.2, 11.0, 11.1, 11.2] to be loaded and used based on the specifications of the Python library (TensorFlow/PyTorch). Appendix A further elaborates on how to use the TAMUQ Raad2 HPC.

3.5 Training Parameters

In ConvNets, there are many hyperparameters to be tuned such that the best performance can be attained. In Table 3, the constant hyperparameters and training parameters are mentioned, where later, the changing parameters based on the segmentation task at hand will be mentioned.

Table 3. Hyperparameters and training parameters' values.

Parameter	Value
Train/Val Ratio (number of folds)	80/20 (5 folds)
Image Rescaling factor	½
Batch Size (batch_size)	32
Input Shape (in_channel)	2.5D (5 slices)
Optimizer (optim)	Adam
Learning Rate Scheduler (scheduler)	ReduceLROnPlateau [139] (Patience = 3 Epochs, Factor = $^{1}/_{10}$) / OneCycleLR [140]
Maximum epochs (epochs)	75
Early Stopping (epochs_patience)	Patience = 6 Epochs

The optimal batch size is usually a power of 2 due to the specifically manufactured hardware configurations with the "bit" mentality. It renders any other batch size less optimal even if larger (The maximum batch size used on TAMUQ's Raad2 is 36 with the 2.5D 5 slices input; however, it was slower in execution). Lastly, early stopping is implemented and is critical to ensure that the training will stop if the model has converged where no further performance enhancements can be observed on the validation set. The goal is to reduce wasted time in the training process. Figure 9 illustrates the optimal place to stop, given that no improvements have been witnessed in the model's performance.

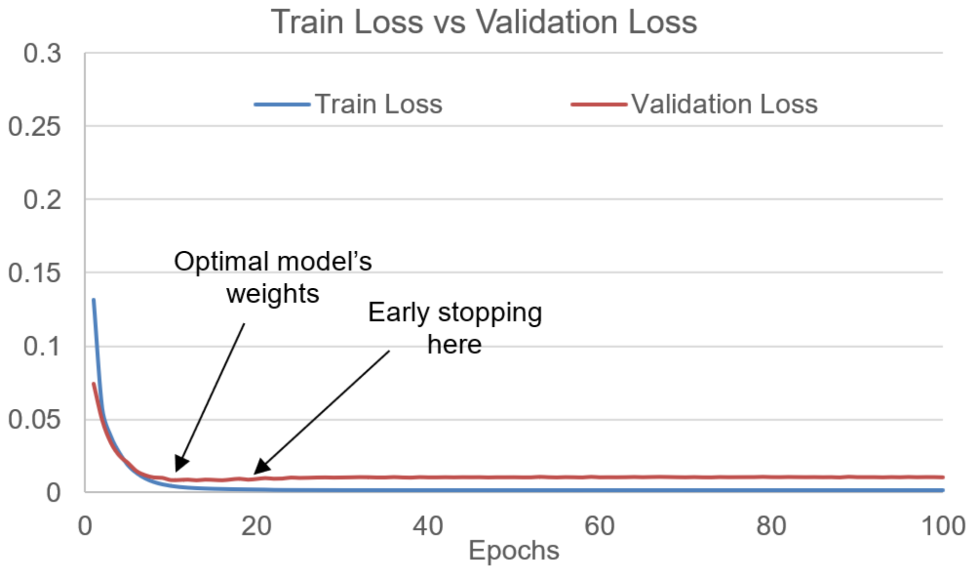


Figure 9. Early stopping reasoning.

3.6 Neural Network Model

As can be noticed from the conducted literature review, many ConvNet architectures have been introduced in the literature and the previous studies. It is cumbersome to try many different ConvNets to achieve similar results to the recent studies. Each study has its add-ons in terms of pre-processing and post-processing (and possibly on different datasets). The authors in [141] demonstrate that the U-Net ConvNet is still a viable solution but only needs fine-tuning to get the optimal results that can compete with the state-of-the-art techniques and the newly devised ConvNets architectures in the literature. Thus, this thesis focuses on the original U-Net, already a significant network architecture deemed powerful for the biomedical segmentation challenge. Figure 10 shows the famous U-Net architecture adapted from their original paper [33]. n portrays the number of slices inserted into the U-Net, and m portrays the number of output channels if a multi-label classification task is being tackled, where the Softmax activation function is used.

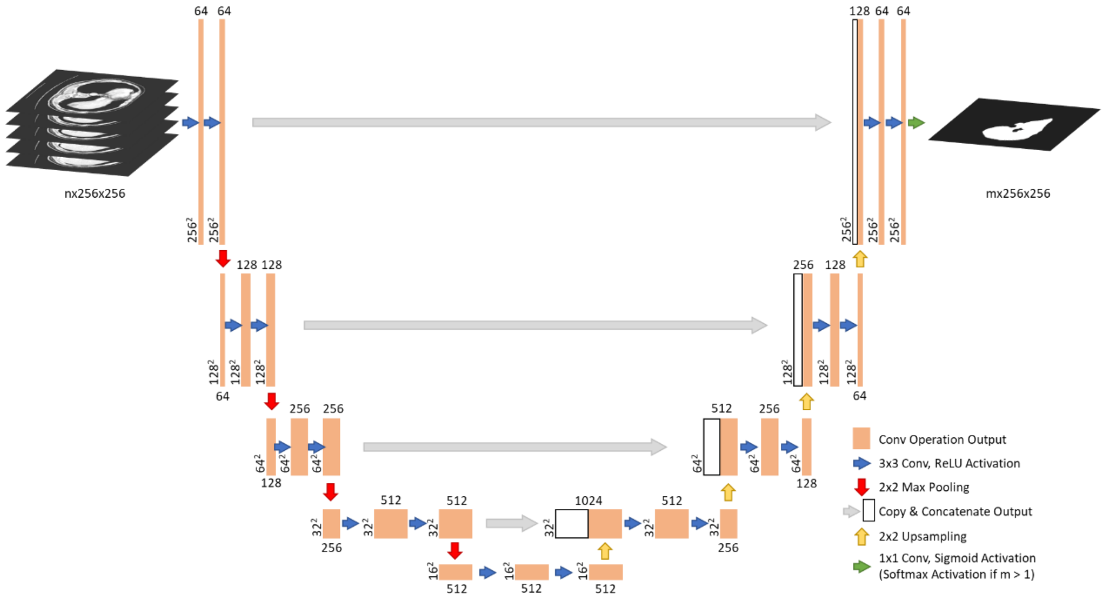


Figure 10. The famous U-Net ConvNet (adapted from [33]).

The developed U-Net is quite similar to the original one in [33]. The numbers "above and below" conv operation output blocks show the number of channels this output has, while the vertical text to the bottom left of the block is the width and height of the output block. For instance, at the first level and first conv output block, the number of channels is 64, and the width and height are both 256.

The following minor changes are applied. Firstly, the input to our U-Net is of size $n \times 256 \times 256$ to reduce the computational complexity (where $n = 5$ slices in our case as we are utilizing 2.5D input), unlike the original one that has an input shape of $1 \times 572 \times 572$. Secondly, we apply padding and stride of 1 throughout the ConvNet to generate output blocks of equal sizes. Applying a stride and padding of 1 is in line with equation (4)

$$W_{new} = \frac{W_{old} - f_w + 2p}{s} + 1, \quad H_{new} = \frac{H_{old} - f_h + 2p}{s} + 1 \qquad (4)$$

where W_{old}, H_{old} is the width and height of the input features, respectively, and W_{new}, H_{new} are the width and height of output features, respectively. $f_w = f_h = f$ is the width and height of the kernels, p is the amount of padding applied, and s is the number of pixels/voxels the filter steps between two adjacent operations. With $f = 3$, $p = 1$, and $s = 1$, we are always guaranteed to have $W_{new} = W_{old}$ and $H_{new} = H_{old}$.

Thirdly, the number of filters the last level has is 512 instead of 1024. The reason is to reduce the number of parameters to tune from 9,438,208 to 2,359,808 per convolutional operation at that level[2]. In the copy and concatenate stage, we do not need to crop any copied outputs because the dimensions of the encoder and decoder sides are identical. For the final output layer, $m = 1$ is used for binary-case classification as either TOI or background.

3.7 Performance Evaluation Metrics

This section mentions the most famous metrics within the liver segmentation literature, highlighting used notations and their significance.

3.7.1 Notations

The following are basic notations that help in understanding the equations for the evaluation metrics:

- A refers to the ground-truth label voxels set
- B refers to the predicted voxels set by the created models
- $|\cdot|$ defines the set cardinality, i.e., number of voxels
- $\|\cdot\|$ represents the Euclidean distance
- $S(\cdot)$ indicates the set of surface voxels
- True positive (TP) is the set of correctly classified TOI pixels/voxels
- True negative (TN) is the set of correctly classified background pixels/voxels, noting that background voxels describe any voxel, which does not belong to the TOI of the study

[2] 3x3x1024x1024+1024 = 9,438,208 parameters ; 3x3x512x512+512 = 2,359,808 parameters

- False positive (FP) is the set of incorrectly classified background pixels/voxels
- False negative (FN) is the set of incorrectly classified TOI pixels/voxels

3.7.2 Percentile Metrics

3.7.2.1 Jaccard Index (JI)

JI is a fundamental metric to understand how closely the generated prediction overlaps with the ground-truth label. It is also known as the Tanimoto index or intersection-over-union (IoU) metric [120]. Equation (5) shows two equivalent definitions of the JI metric.

$$JI = \frac{|A \cap B|}{|A \cup B|} = \frac{TP}{TP + FP + FN} \tag{5}$$

Intuitively, perfect prediction is when JI is equal to 1, meaning that |A ∩ B| is the same as |A ∪ B|. In other words, there are no wrong predictions (i.e., FP and FN = 0), and the volumes are perfectly similar. In contrast, JI equating to 0 means that no intersection whatsoever exists between the ground-truth and prediction, or TP is 0, meaning that the TOI is completely misclassified.

3.7.2.2 Precision/Positive Predictive Value (PPV)

Precision aims to investigate the over-segmentation aspect of the model by dividing the total number of correctly classified TOI voxels over the total positively classified voxels (i.e., true and false) as indicated by equation (6).

$$Precision/PPV = \frac{TP}{TP + FP} \tag{6}$$

Where a value of 1 indicates an ideal segmentation scenario of classifying background voxels correctly. In contrast, a value of 0 is the extreme case of incorrectly classifying all TOI voxels.

3.7.2.3 Recall/Sensitivity/True Positive Rate (TPR)

On the other hand, recall investigates the under-segmentation aspect of the model by dividing the correctly classified TOI voxels over the "actual" number of TOI voxels, as shown by equation (7).

$$Recall/Sensitivity/TPR = \frac{TP}{TP + FN} \tag{7}$$

Recall varies between 0 and 1, where 1 indicates perfect segmentation of all TOI voxels, and 0 indicates the exact opposite.

3.7.2.4 Dice Similarity Coefficient (DSC/Dice)

DSC (or Dice) is the F1 Score counterpart for images, a harmonic mean of precision and recall. In a sense, it measures the similarity between ground-truth set A and generated prediction B. The original Dice for a single image is defined in equation (8).

$$Dice = 2\frac{|A \cap B|}{|A| + |B|} = \frac{2TP}{2TP + FP + FN} \tag{8}$$

Similar to the JI metric, the two extreme cases are 0 and 1, where the former emphasizes the absence of any similarity and the latter shows a perfect similarity between A and B.

3.7.2.5 Specificity/True Negative Rate (TNR)

Depicted in equation (9), specificity investigates the model capability in classifying background voxels correctly; it is the same as the Recall but for the negative class.

$$Specificity/TNR = \frac{TN}{TN + FP} \tag{9}$$

Ranging between 0 and 1, the former denotes a misclassification of all background voxels, and the latter resembles a proper classification for all background voxels.

3.7.2.6 Volumetric Overlap Error (VOE)

VOE is the complementary metric of JI, also known as the Jaccard distance, knowing that VOE is a particular case for volumetric sets. It measures the spatial error represented between the voxels of A and B [120] and is described by equation (10).

$$VOE = 1 - JI = 1 - \frac{|A \cap B|}{|A \cup B|} = \frac{FP + FN}{TP + FP + FN} \tag{10}$$

VOE ranges between 0 and 1, where the former means that the voxels of B are perfectly and correctly lying over A's voxels, where the latter indicates the absence of overlapping voxels between the voxels of A and B.

3.7.2.7 Relative Volume Difference (RVD)

RVD measures the difference between volume A and B, and is an indicator of whether the set of voxels encompassed by B is an under- or over-segmentation by comparing it with A's voxels [120]. Equation (11) highlights this metric.

$$RVD(A, B) = \frac{|B| - |A|}{|A|} \tag{11}$$

This metric can be positive, negative, or zero, whereas positive indicates that B is over-segmenting the original volume, negative indicates an under-segmentation case, and zero as having identical volumes. RVD should not be used alone as it does not necessarily indicate an overlap between A and B [120].

3.7.3 Distance Measurements

The distance measurements extensively used in literature are mentioned here, each measurement captures a specific spatial aspect, and all of them are measured in millimeters [mm].

3.7.3.1 Average Symmetric Surface Distance (ASD)

ASD measures the minimum distance that can be found between a surface voxel in A to another surface voxel in B. Since the metric is symmetric, the same applies to B with respect to A. Then, the average is taken over all the calculated distances. Surface voxel is a name given to a voxel with at least one non-TOI voxel (i.e., background voxel) from its 18-neighboring voxels, as shown in Figure 11.

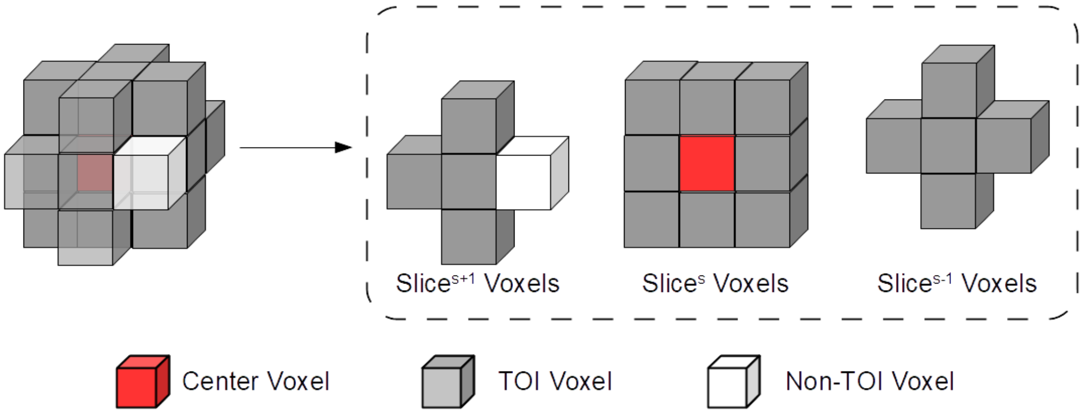


Figure 11. A surface voxel (red) with 1 non-TOI adjacent voxel (white) and 17 TOI voxels (grey).

It is necessary to define the minimum distance between an arbitrary voxel v and the set of surface voxels $S(A)$ to define ASD, as follows:

$$d(v, S(A)) = \min_{s_A \in S(A)} \|v - s_A\| \tag{12}$$

where s_A is a single surface voxel from the surface voxels set $S(A)$. Using equation (12), it is possible now to define ASD as follows:

$$ASD(A, B) = \frac{1}{|S(A)| + |S(B)|} \left(\sum_{s_A \in S(A)} d(s_A, S(B)) + \sum_{s_B \in S(B)} d(s_B, S(A)) \right) \tag{13}$$

From equation (13) and the definition of Euclidean distance in equation (12), it is observed that this metric is always positive. The value converges to 0 when the highest spatial similarity is achieved. However, the larger the value, the worse the overlap between volumes A and B is noticed, and dissimilarity starts to be observed.

3.7.3.2 Root-Mean-Square Symmetric Surface Distance (RMSD)

RMSD is significantly related to the ASD definition described by equation (13). The root-mean-square of ASD is the summation of distances squared under the square root as defined by equation (14).

$$RMSD(A,B) = \sqrt{\frac{1}{|S(A)| + |S(B)|}\left(\sum_{s_A \in S(A)} d^2(s_A, S(B)) + \sum_{s_B \in S(B)} d^2(s_B, S(A))\right)} \quad (14)$$

The benefit of defining such a metric is the amount of weight bestowed over large deviations, making the metric more sensitive to outliers [120]. Equivalently, the best value for this metric should be 0, and the bigger it is, the worse the volumes' overlap is.

3.7.3.3 Maximum Symmetric Surface Distance (MSD)

MSD, famously known as Hausdorff Distance (HD) as well, searches for the maximum distance, defined by equation (15), that can be found between volumes A and B.

$$HD(A,B) = \max\left(\underset{s_A \in S(A)}{max} d\big(s_A, S(B)\big), \underset{s_B \in S(B)}{max} d\big(s_B, S(A)\big)\right) \quad (15)$$

This metric gives the maximum distance error between A and B, and thus, is extremely sensitive to outliers [120]. A 95% HD variant takes the maximum distance on 95% of sorted distances; hence, 95% HD will always be less than HD.

3.8 3D Construction and Printing

Once the models with the highest performance have been developed and identified, it is possible to use them in a clinical environment to aid surgeons and physicians in performing hepatectomy. A script has been implemented to interpolate a 3D object out of the 2D segmented slices from a CT scan to make it smoother and more human-friendly to visualize the liver's tissues. Consequently, it will be possible for the clinicians to see the liver, along with its tumors and vessels, in a 3D form that can be easily accessed over a typical computer or 3D printed for them to conduct a trial operation. The end goal is to have the liver as seen in Figure 12 (the parenchyma is colored as white, but the color has been slightly modified to show in this report).

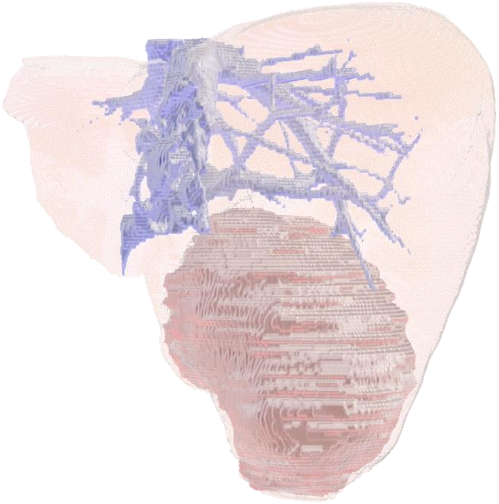
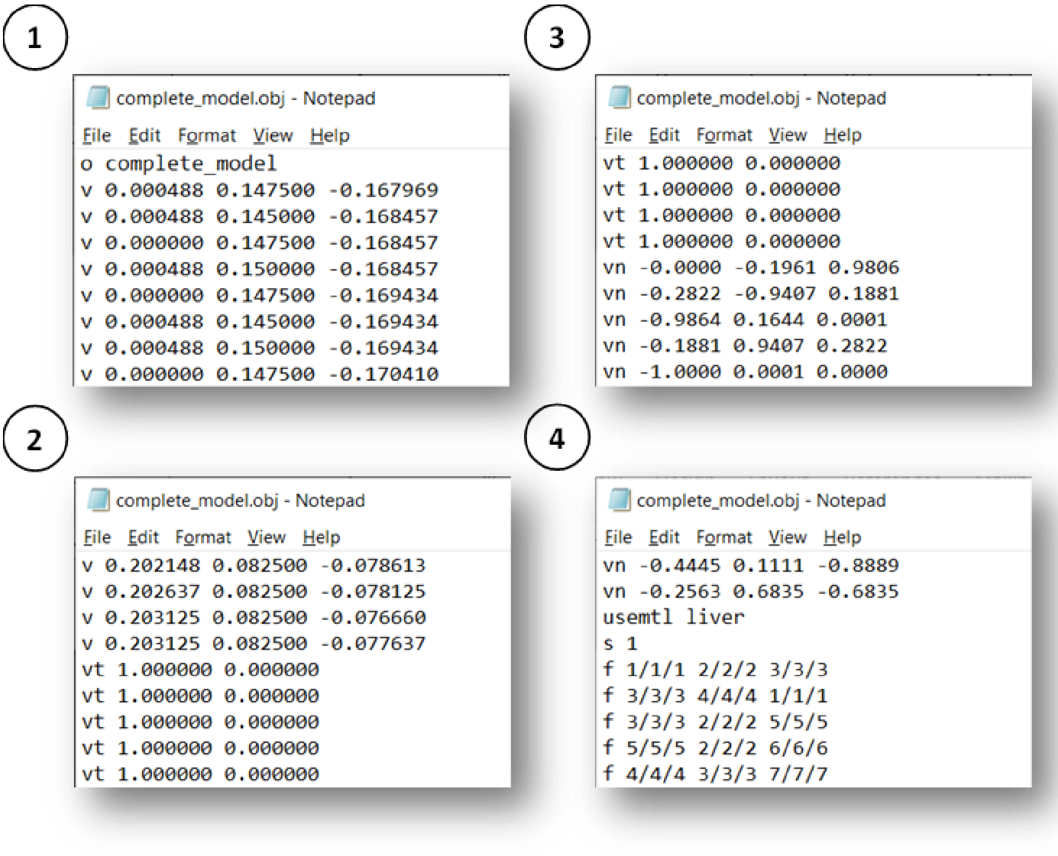


Figure 12. 3D object interpolation of record 294 from the MSDC-T8 dataset.

Figure 12 is the product of the developed 3D interpolation script, where the 2D ground-truth slices, fetched from both the organizers [128] and the researchers [57], are being interpolated.

It is worth noting that an algorithm called “The Marching Cubes” from the scikit-image library on Python [142] is utilized to interpolate the 2D cross-sectional slices into their 3D counterpart. It is deemed suitable for the case of this thesis as our investigation is more towards the ML part implemented on the 2D slices. However, the algorithm can do the 3D interpolation only, but combining them into a single .obj file and utilizing a .mtl file to add the coloring and textures has been implemented on Python from scratch.

Firstly, the 3 tissues’ masks are loaded into the script after being generated by the designated ConvNets, via the nibabel Python library. Then, the 3D Lewiner interpolation using “The Marching Cubes” algorithm, from the scikit-image Python library, is used to create the 3D object of each tissue (the .obj file). It is worth noting that 4 mandatory types of parameters are mentioned in the .obj file, namely, the vertex (v), the vertex texture (vt), the vertex normal (vn), and the face (f). Figure 13 shows the different information needed to construct the .obj file.


1
complete_model.obj - Notepad
File Edit Format View Help
2
complete_model.obj - Notepad
File Edit Format View Help
3
complete_model.obj - Notepad
File Edit Format View Help
4
complete_model.obj - Notepad
File Edit Format View Help

```
o complete_model
v 0.000488 0.147500 -0.167969
v 0.000488 0.145000 -0.168457
v 0.000000 0.147500 -0.168457
v 0.000488 0.150000 -0.168457
v 0.000000 0.147500 -0.169434
v 0.000488 0.145000 -0.169434
v 0.000488 0.150000 -0.169434
v 0.000000 0.147500 -0.170410
```

```
v 0.202148 0.082500 -0.078613
v 0.202637 0.082500 -0.078125
v 0.203125 0.082500 -0.076660
v 0.203125 0.082500 -0.077637
vt 1.000000 0.000000
vt 1.000000 0.000000
vt 1.000000 0.000000
vt 1.000000 0.000000
vt 1.000000 0.000000
```

```
vt 1.000000 0.000000
vt 1.000000 0.000000
vt 1.000000 0.000000
vt 1.000000 0.000000
vn -0.0000 -0.1961 0.9806
vn -0.2822 -0.9407 0.1881
vn -0.9864 0.1644 0.0001
vn -0.1881 0.9407 0.2822
vn -1.0000 0.0001 0.0000
```

```
vn -0.4445 0.1111 -0.8889
vn -0.2563 0.6835 -0.6835
usemtl liver
s 1
f 1/1/1 2/2/2 3/3/3
f 3/3/3 4/4/4 1/1/1
f 3/3/3 2/2/2 5/5/5
f 5/5/5 2/2/2 6/6/6
f 4/4/4 3/3/3 7/7/7
```

Figure 13. Mandatory components of a .obj file.

The (v)s define the coordinates of vertices in the x, y, z 3D space. On the other hand, the (vt)s and (vn)s are not as important, but the (vt) aid in making the 3D object more realistic by mapping a pre-defined texture map on the faces, and the (vn) aid in specifying the direction of the normal vector [143]. Lastly, the (f)s combine the three above components in the following order as shown by equation (16):

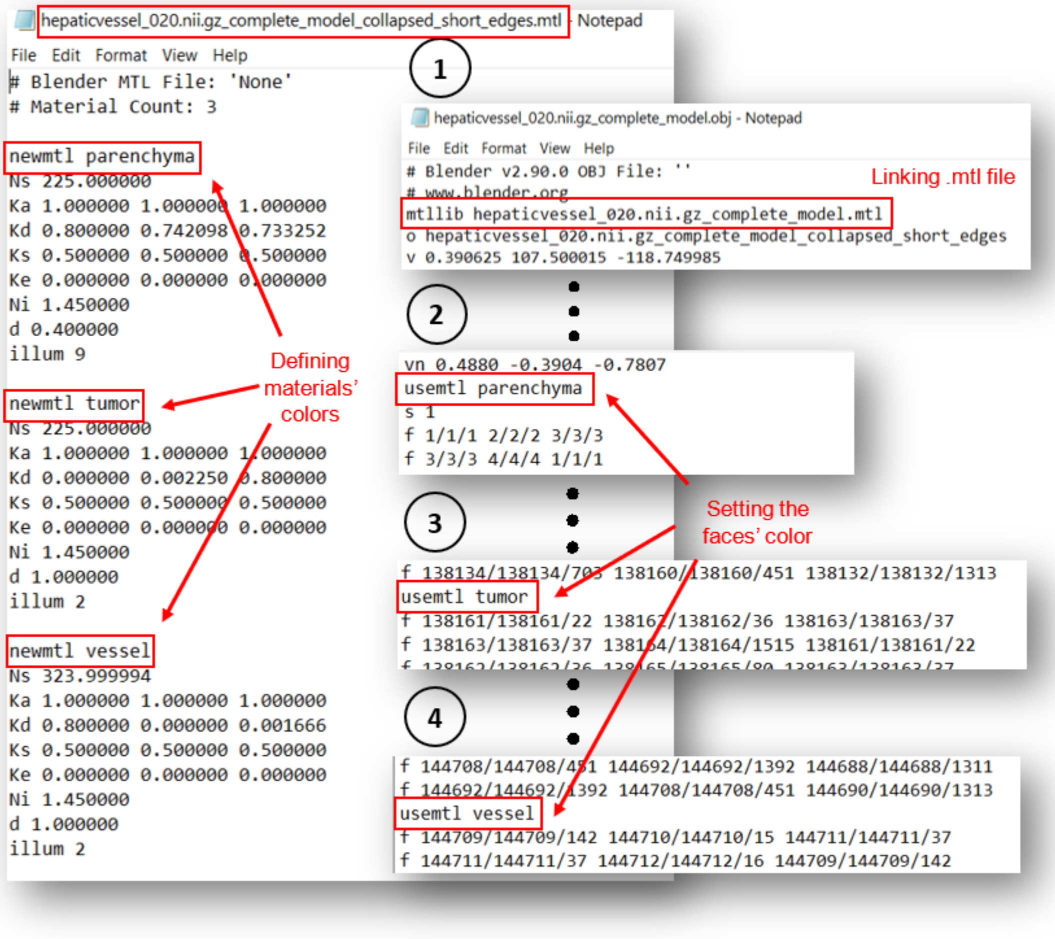
$$f \quad v_1[/vt_1][/vn_1] \quad v_2[/vt_2][/vn_2] \quad v_3[/vt_3][/vn_3] \ldots v_x[/vt_x][/vn_x] \qquad (16)$$

where x is the number of vertices designated for each face f, as it can have as many vertices as needed [144].

The square brackets [ ] in equation (16) indicate that these values are optional. Moreover, because the scikit-image library produces vertices linked generating triangular faces, 3 vertices are needed to build each face.

Once the .obj file has been created for each tissue, the statement "mtllib <material_filename>.mtl" is added, enabling linking with the .mtl file. The linking allows importing the defined texture and color onto different surfaces. Moreover, the statement "usemtl <tissue_material>" precedes the faces of an object in that file, allowing any software to assign the material color/texture to that object. Lastly, .obj files for all tissues are then combined by placing all the (v)s of each .obj at the beginning, followed by all the (vt)s beneath the (v)s, and followed by the (vn)s beneath both (v)s and (vt)s. The same is applied to the (f)s; however, because more (v)s, (vt)s, and (vn)s have been added within the same file, the 2nd, 3rd, 4th,…, x^{th} vertices for the (f)s have to account for the number of vertices that are reserved for the preceding objects. Figure 14 elaborates more on the idea of combining different .obj files into a single one, where it can be noticed that the tumor object uses vertices starting from the 1380161st vertex, as the previous 1380160 are used to construct the liver object.

Lastly, it is well-known that there is available software that can do the 2D to 3D interpolation. However, the reasoning behind implementing it on Python is to have an end-to-end tool for surgeons and clinicians to use such that it eliminates the necessity of third-party software. The end-users interactions are intended to be brought down to the minimum, to feed the CT scan into the desktop application and expect the output to be a 3D interpolated object ready to be visualized and printed. In the upcoming results chapter, we show the developed desktop application to achieve this goal.


hepaticvessel_020.nii.gz_complete_model_collapsed_short_edges.mtl - Notepad
File Edit Format View Help
1
hepaticvessel_020.nii.gz_complete_model.obj - Notepad
File Edit Format View Help
Linking .mtl file
Defining materials' colors
2
3
Setting the faces' color
4

```
# Blender MTL File: 'None'
# Material Count: 3

newmtl parenchyma
Ns 225.000000
Ka 1.000000 1.000000 1.000000
Kd 0.800000 0.742098 0.733252
Ks 0.500000 0.500000 0.500000
Ke 0.000000 0.000000 0.000000
Ni 1.450000
d 0.400000
illum 9

newmtl tumor
Ns 225.000000
Ka 1.000000 1.000000 1.000000
Kd 0.000000 0.002250 0.800000
Ks 0.500000 0.500000 0.500000
Ke 0.000000 0.000000 0.000000
Ni 1.450000
d 1.000000
illum 2

newmtl vessel
Ns 323.999994
Ka 1.000000 1.000000 1.000000
Kd 0.800000 0.000000 0.001666
Ks 0.500000 0.500000 0.500000
Ke 0.000000 0.000000 0.000000
Ni 1.450000
d 1.000000
illum 2
```

```
# Blender v2.90.0 OBJ File: ''
# www.blender.org
mtllib hepaticvessel_020.nii.gz_complete_model.mtl
o hepaticvessel_020.nii.gz_complete_model_collapsed_short_edges
v 0.390625 107.500015 -118.749985
```

```
vn 0.4880 -0.3904 -0.7807
usemtl parenchyma
s 1
f 1/1/1 2/2/2 3/3/3
f 3/3/3 4/4/4 1/1/1
```

```
f 138134/138134/703 138160/138160/451 138132/138132/1313
usemtl tumor
f 138161/138161/22 138162/138162/36 138163/138163/37
f 138163/138163/37 138164/138164/1515 138161/138161/22
f 138162/138162/36 138165/138165/80 138163/138163/37
```

```
f 144708/144708/451 144692/144692/1392 144688/144688/1311
f 144692/144692/1392 144708/144708/451 144690/144690/1313
usemtl vessel
f 144709/144709/142 144710/144710/15 144711/144711/37
f 144711/144711/37 144712/144712/16 144709/144709/142
```

Figure 14. Resulting .obj and .mtl files main components.

To extend our work beyond just a mere visualization tool of liver tissues, it is also possible to have it printed for the end-user to hold physically. Hence, many companies have been contacted to query regarding their 3D printing services within the biomedical field. We contacted the following 5 companies to evaluate their services (see Table 4).

Out of these five companies, two companies were highly responsive and professional, which are i.Materialise and Axial3D. Both have shown interest in our work; however, i.Materialise was much more expensive than what we could afford. Contrastingly, with Axial3D, we were able to negotiate a 50% discount on our order given that we are one of the first customers situated in Qatar, and they would like to expand their market share here. Thus, we decided to work with Axial3D as their services are more within our budget.

Table 4. Companies that are involved in 3D printing and have been contacted.

Company	Ability to Print	Country	Delineate	Printing Quality	Expert	Cost
i.Materialise	Yes (DICOM)	Belgium	Yes	Rubber Parenchyma	Great	$$$$$
Axial3D	Yes (NIfTI)	UK	Yes	Solid Parenchyma	Superb	$$$
Deed3D	Yes (NIfTI)	China	No	Solid Parenchyma	Good	N/A
B9Creations	No	US	N/A	N/A	Bad	N/A
RapidMade	N/A	US	N/A	N/A	N/A	N/A

To conclude, the details behind our implementation are thoroughly discussed in this chapter: the utilized dataset, the data pre-processing techniques, the input shape transformation, the HPC platform, the training parameters, the ConvNet U-Net model, the performance metrics, and the 3D construction and printing. Our detailed explanation should allow researchers to replicate our work and compare their results with ours.

CHAPTER 4: EXPERIMENTAL RESULTS AND DISCUSSION

This chapter depicts and discusses the experimental results generated from the applied preprocessing techniques and the chosen ConvNet. This thesis tackles three main segmentation tasks: liver segmentation, tumors (within liver) segmentation, and vessels (within liver) segmentation. Each segmentation task is further elaborated in a section of its own. It is worth noting that the framework of choice is PyTorch (torch version==1.8.1+cu102), which is responsible for the results that are mentioned below. All the displayed results are the outcomes of a 5-fold cross-validation (80% training/20% validation) procedure. The maximum number of epochs per fold is 75, but early stopping is utilized to avoid unnecessary training time ($epochs\_stop$ = 6). A $batch\_size$ of 32 is used to maximize the learning speed of the model, consequently reducing the training time for the 5-fold cross-validation. Adam optimizer is used in general for all the tasks below. Lastly, it is worth mentioning and praising the Raad2 HPC provided by TAMUQ, as we utilized the NVIDIA Tesla V100 GPU and the Intel Xeon Skylake CPU in this thesis.

4.1 Liver Segmentation

The liver should be initially segmented with high precision and accuracy to segment tumors and vessels successfully. Logically, it makes the liver segmentation a high priority, which will then be used to localize the segmentation area for both tumors and vessels. The liver-related slices and voxels will be the input to both the tumors and vessels ConvNets, as shown previously in Figure 3.

It is vital to explicitly mention the CT records being used in the testing set such that interested researchers can confidently replicate our results and possibly surpass them by testing on the same records. They were selected randomly from the MSDC-T8 dataset, and they sum up to 23 records in total. They are the following:

[003, 012, 045, 072, 090, 105, 117, 129, 141, 153, 169, 178, 193, 205, 220, 236, 246, 258, 268, 280, 294, 304, 320] for the liver segmentation task. When implementing the 5-fold cross-validation, ~24K slices end up in the 4 training folds, and ~6K slices end up in the validation fold. For the tumors and vessels segmentation tasks, the records are less, given that not all the 23 volumes have the tumors and vessels masks, and subsequently, the number of slices for both is different.

We investigate the best model by studying the best input shape inserted into the U-Net ConvNet, then checking different learning rates (LRs) and scheduling techniques that influence the LR throughout the training phase.

4.1.1 Optimal Input Shape Investigation

In this subsection, the optimal ConvNet’s input shape is investigated, as from the previously conducted literature survey, it has been established that different input shapes are fed into the ConvNets. Given that this thesis uses a 2D ConvNet architecture, the 2D and 2.5D input shapes are the shapes of interest in this subsection as they are easily created and input into the ConvNet. Table 5 shows the comparison between 2D (1 slice) and 2.5D (3, 5, and 7 slices) input shapes. It can be observed from the portrayed results that the 2.5D (5 slices) is the optimal input shape for the liver segmentation challenge on the MSDC-T8 dataset.

Moreover, a gradual increase in performance can be observed until the peak performance is reached at 2.5D (5 slices), and then it slightly declines when 7 slices are inserted into the ConvNet. Although the ConvNet performs well for all different input shapes, the optimal input shape is the 2.5D input shape. It provides the ConvNet with the volumetric context without overdoing it, as sometimes the inter-slices distance is significant for the volumetric context to be captured. The effect of capturing volumetric context information via the 2.5D input shape can be observed on the distance measurement, as the 2D version has the highest distance errors.

Table 5. Different input shapes into U-Net performance comparison.

Input shape	Dice (%)	IoU (%)	RVD	ASD (mm)	RMSD (mm)	HD (mm)	95% HD (mm)	Epoch
2D (1 slice)	97.84 (0.21)	95.81 (0.38)	-0.01 (0.003)	0.902 (0.564)	3.16 (1.62)	35.94 (5.65)	6.49 (5.35)	**18.6** **(3)**
2.5D (3 slices)	98.04 (0.07)	96.17 (0.13)	-0.01 (0.003)	0.596 (0.184)	2.34 (0.81)	**26.94** **(3.48)**	**2.66** **(0.46)**	22 (1.7)
2.5D (5 slices)	**98.12** **(0.04)**	**96.33** **(0.07)**	**-0.008** **(0.002)**	**0.624** **(0.443)**	**2.15** **(1.40)**	27.16 (4.53)	4.10 (4.37)	22.0 (1.6)
2.5D (7 slices)	98.02 (0.13)	96.15 (0.24)	-0.008 (0.005)	0.763 (0.639)	2.41 (1.5)	29.73 (1.71)	4.48 (4.38)	20.2 (2.5)

4.1.2 Scheduling Techniques and LRs Variations

In this part, we investigate the different suitable schedulers along with the various LRs, while keeping other variables as constants. The comparison is between two powerful schedulers, ReduceLRonPlateau [139] and OneCycleLR [140]. Each scheduler has a different philosophy behind its creation. On the one hand, the idea behind the ReduceLRonPlateau is to reduce the LR when the model converges to the cost function' local minimum that it is approaching. By doing so, we can squeeze the most out of the model and tune the weight to be as close as we can to the local minimum. On the other hand, OneCycleLR attempts to prevent the model from falling into a local minimum suitable for the training set (overfitting). It prevents the model from overfitting by gradually boosting the LR to get stuck at a "better" cost function's minimum. In a sense, it is considered as a regularization step in the model training [145]. Figure 15 shows the difference in the incurred LR values during the training phase, assuming that the early stopping does not intervene to stop the model's training. Both are implemented on PyTorch, making it easy to "plug and play" both schedulers, testing their efficacy with the liver segmentation task.

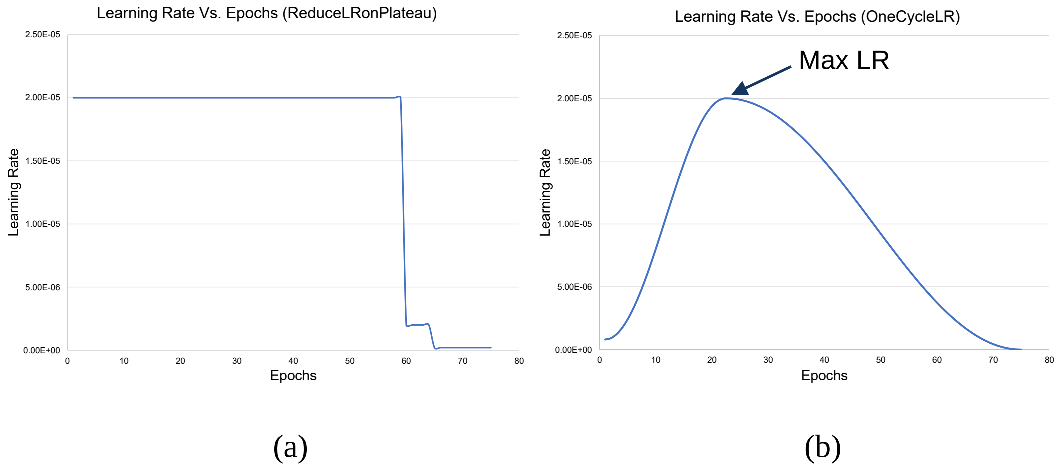


Figure 15. LR change via (a) OneCycleLR and (b) ReduceLRonPlateau.

Table 6 shows the mean and standard deviation from the 5-fold cross-validation implemented per scheduler and LR. Results are reported using the mean (standard deviation) of 5-fold cross-validation.

Table 6. Both schedulers with different LRs are reported for each metric.

Scheduler	LR (10^{-5})	Dice (%)	IoU (%)	RVD	ASD (mm)	RMSD (mm)	HD (mm)	95% HD (mm)	Epochs
OneCycleLR (Max LR)	2	97.67 (0.12)	95.48 (0.23)	-0.008 (0.004)	1.055 (0.509)	3.81 (1.15)	44.00 (4.43)	5.13 (4.03)	42.6 (2.6)
	4	97.72 (0.15)	95.58 (0.28)	-0.009 (0.004)	1.002 (0.522)	3.70 (1.58)	41.58 (7.41)	5.09 (4.12)	34.0 (3.1)
	8	97.93 (0.09)	95.97 (0.17)	-0.009 (0.005)	0.576 (0.089)	2.51 (0.30)	37.78 (1.86)	2.82 (0.70)	29.4 (2.2)
	16	98.01 (0.06)	96.12 (0.11)	-0.008 (0.005)	0.521 (0.110)	2.05 (0.45)	31.51 (6.71)	2.68 (0.67)	29.0 (2.9)
	24	98.07 (0.11)	96.24 (0.21)	-0.005 (0.003)	**0.482** **(0.076)**	**1.88** **(0.29)**	29.25 (5.18)	**2.43** **(0.59)**	30.0 (3.8)
	32	98.03 (0.10)	96.16 (0.19)	**-0.004** **(0.005)**	0.606 (0.147)	2.47 (0.71)	34.31 (3.11)	2.61 (0.45)	25.0 (3.7)
	40	98.04 (0.17)	96.17 (0.32)	-0.007 (0.003)	0.503 (0.131)	1.97 (0.74)	33.17 (8.63)	2.56 (0.82)	25.8 (2.0)
ReduceLRonPlateau	0.4	97.20 (0.13)	94.60 (0.24)	-0.009 (0.004)	1.541 (0.276)	5.59 (0.63)	59.02 (7.91)	10.63 (3.15)	72.8 (3.9)
	0.8	97.40 (0.15)	94.96 (0.27)	-0.010 (0.003)	1.097 (0.238)	4.47 (0.86)	59.47 (7.83)	5.56 (3.35)	48.6 (4.0)
	2	97.69 (0.13)	95.51 (0.24)	-0.011 (0.003)	0.751 (0.130)	3.03 (0.61)	40.43 (1.46)	3.40 (0.76)	30.6 (1.3)
	4	97.87 (0.06)	95.86 (0.11)	-0.011 (0.003)	0.625 (0.142)	2.57 (0.84)	38.26 (7.34)	2.73 (0.12)	23.0 (2.8)
	8	98.04 (0.10)	96.18 (0.19)	-0.009 (0.003)	0.528 (0.104)	1.99 (0.52)	**27.03** **(6.55)**	2.67 (0.73)	20.8 (3.6)
	16	**98.12** **(0.04)**	**96.33** **(0.07)**	-0.008 (0.002)	0.624 (0.443)	2.15 (1.40)	27.16 (4.53)	4.10 (4.37)	22.0 (1.6)
	32	98.02 (0.10)	96.15 (0.19)	-0.006 (0.004)	0.515 (0.062)	2.09 (0.52)	33.87 (10.39)	2.58 (0.42)	**18.8** **(4.2)**

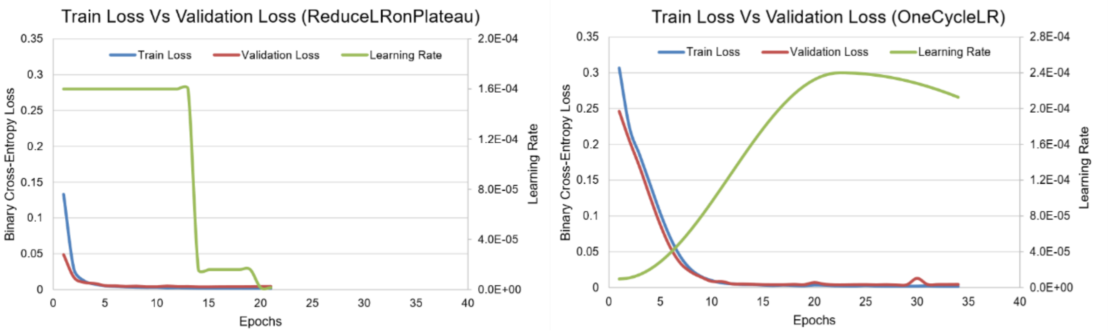
The mentioned LRs for the OneCycleLR scheduler are the maximum LRs that the scheduler will reach (refer to Figure 15b).

4.1.3 Best Model Selection and Discussion

The best results per metric are bolded in Table 6. Firstly, it is worth noting that the achieved results using the original U-Net in 2D and 2.5D modes outperform the ones described in [57]. It achieves a better Dice result than the 2D (with and without convolutional long short-term memory (LSTM)), the 2.5D, and the 3D U-Net counterparts in the liver segmentation task. Moreover, it comes highly close to the proposed GLC-UNet (98.18 ± 0.85)% in [57], while the best run from ReduceLROnPlateau (with LR=16×10^{-5} achieves (98.12 ± 0.04)% in Dice. This outcome shows that it is possible to extract even higher results from their GLC-UNet if they adopt our pre-processing and scheduling techniques. Additionally, the obtained results per fold are close to each other as the reported standard deviation are relatively small.

The chosen LR must not be too small, so the network avoids getting stuck at a local minimum, consequently having prolonged training periods and worse performance on the validation set. This point is evident for both schedulers by observing that higher LRs tend to generate ConvNets that generalize better (more finely tuned) with less training time. In comparing both scheduling techniques in the generated results, the best result from ReduceLROnPlateau (LR = 16×10^{-5} with Dice = 98.12 ± 0.04%) surpasses the best run from OneCycleLR (LR_{max} = 24×10^{-5} with Dice = 98.07 ± 0.11%). Moreover, it converged faster than the OneCycleLR since OneCycleLR initially starts with a small LR that gradually increases and reaches the peak around epoch 23 (refer to Figure 15b). In contrast, ReduceLROnPlateau starts strong with a high LR when compared to the OneCycleLR scheduler.

Figure 16 shows a convergence example from the two best runs from OneCycleLR and ReduceLRonPlateau schedulers, respectively, highlighting the point mentioned above on why ReduceLRonPlateau converges faster in this context.



(a) (b)

Figure 16. Convergence plots for best fold run (a) ReduceLRonPlateau LR=16×10^{-5} and (b) OneCycleLR with LR=24×10^{-5}.

Given that the best model has been found for the liver segmentation challenge, it is possible to tackle the following challenges, represented by the tumors and vessels segmentation. Additionally, it is worth noting that the segmented liver volume can be utilized to localize and remove background voxels from the input to both tumors and vessels ConvNets (as represented by Figure 3). Segmenting the liver beforehand allows both tumors and vessels ConvNets to focus on the critical voxels within the liver. Figure 17 shows an example of performing an element-wise multiplication operation between CT scan slice 128 (record #369) and its ground-truth mask to generate the segmented slices used as inputs to the following ConvNets.

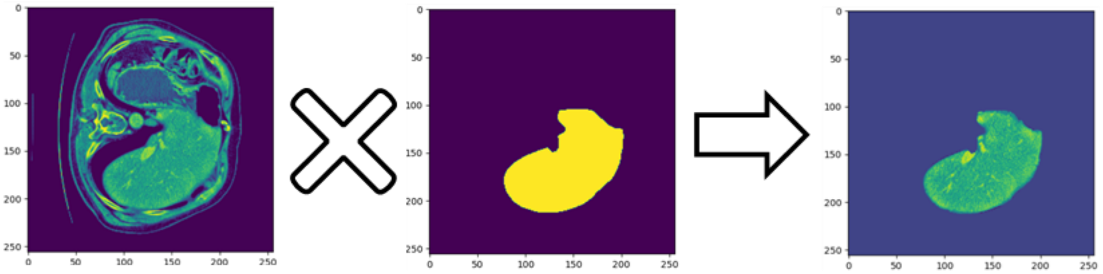


Figure 17. Element-wise multiplication of the CT scan slice with its liver mask.

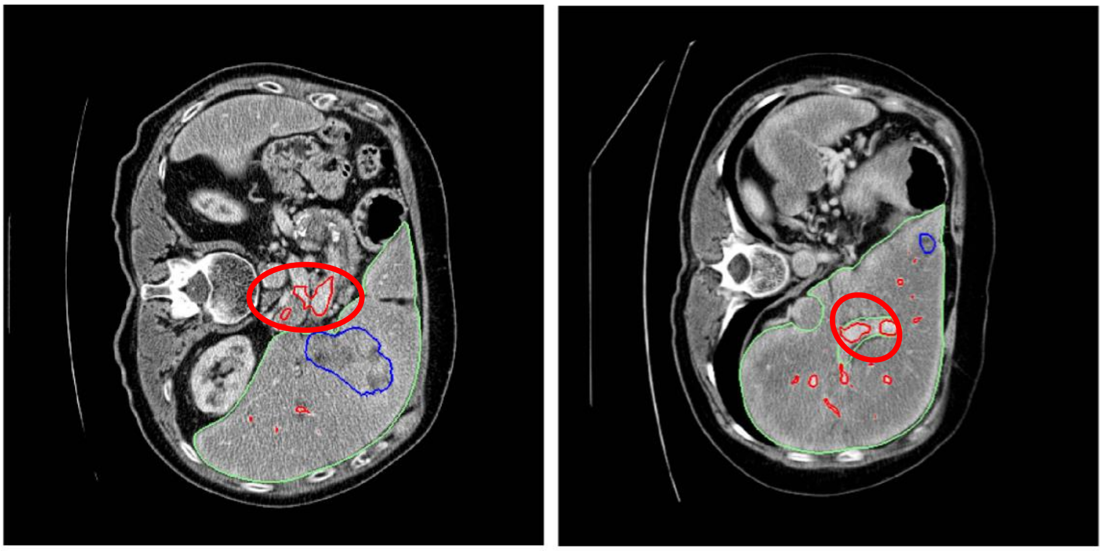
4.2 Tumors and Vessels Segmentation

Out of the 23 records in the testing set, the tumors and vessels' ground-truth masks are available for 14 only, making it a viable solution to test the tumors and vessels segmentation ConvNets. They are the following: [072, 090, 117, 129, 141, 178, 193, 236, 246, 258, 268, 280, 294, 320] for the tumors and vessels segmentation tasks. When implementing the 5-fold cross-validation, ~12K slices end up in the 4 training folds, and ~3K slices end up in the 5th validation fold.

By obtaining the segmented liver via the produced liver masks from the liver ConvNet (as depicted by Figure 17), following tumors and vessels' ConvNets can focus on the liver voxels by disregarding all other voxels as background because they will be equated to zero. However, to show the efficacy of multiplying the liver mask with the CT slices, Table 7 shows the "before and after" effect of applying the ground-truth liver masks for the tumors and vessels segmentation tasks. For the sake of this comparison, the OneCycleLR scheduler has been used, given it is outperforming the ReduceLRonPlateau scheduler in the tumors and vessels segmentation challenges. Table 7 shows that the tumors segmentation has significantly increased when the liver mask has been applied. In contrast, the vessels segmentation did not get improved; instead, the performance slightly worsened. This phenomenon can be attributed to two factors: 1) The liver's vessels structure is unique, so applying the liver mask does not add significant value; however, for the tumors, it did because they are easier to be confused with other organs looking like a tumor, such as the kidneys. Hence, by applying the liver mask, these organs are removed. 2) The method liver masks are created, as some do not include vessels as part of the liver, and when multiplied by the liver mask, the vessels' voxels are discarded. Figure 18 shows two examples of vascular tissues that are discarded when multiplying with the generated liver mask. Green is the ground-truth liver mask, blue is the ground-truth tumors mask, and red is the ground-truth vessels mask.

Table 7. Performance metrics of both tumors' and vessels' ConvNets before and after applying liver masks.

TOI	Stage	LR (10^{-5})	Dice (%)	IoU (%)	RVD	ASD (mm)	RMSD (mm)	HD (mm)	95% HD (mm)
TUMORS	Before	4	38.62 (3.54)	27.66 (2.78)	-0.335 (0.108)	38.018 (12.247)	46.08 (12.5)	105.37 (13.59)	71.46 (16.35)
		28	51.18 (4.08)	39.09 (3.72)	-0.042 (0.247)	41.948 (13.383)	51.53 (12.87)	120.27 (20.09)	76.85 (18.39)
		40	53.24 (4.03)	40.53 (3.88)	0.252 (0.371)	31.109 (11.35)	41.67 (9.68)	120.55 (17.12)	69.11 (10.34)
	After	4	57.73 (1.1)	45.19 (0.86)	**0.040 (0.162)**	25.299 (16.011)	32.15 (15.2)	89.57 (11.13)	51.03 (14.22)
		28	**59.71 (3.91)**	**46.94 (3.68)**	0.251 (0.328)	17.126 (13.208)	23.25 (12.18)	77.39 (8.22)	**38.49 (11.01)**
		40	57.86 (2.41)	45.22 (2.65)	0.312 (0.514)	**11.445 (1.618)**	**18.04 (1.73)**	**68.95 (5.63)**	38.57 (5.92)
VESSELS	Before	4	50.50 (1.18)	34.79 (1.07)	0.052 (0.118)	4.363 (0.336)	9.15 (0.57)	79.64 (10.09)	19.06 (1.46)
		28	**52.68 (1.38)**	**36.85 (1.06)**	0.065 (0.094)	4.563 (0.376)	10.27 (1.98)	72.09 (17.79)	19.81 (0.87)
		40	52.39 (1.84)	36.63 (1.74)	**0.038 (0.115)**	**4.164 (0.163)**	8.22 (0.21)	58.13 (13.10)	18.51 (0.87)
	After	4	48.63 (1.30)	33.18 (1.13)	-0.098 (0.133)	4.230 (0.395)	7.97 (0.54)	49.25 (6.05)	**18.34 (1.33)**
		28	50.34 (1.82)	34.73 (1.52)	-0.045 (0.144)	4.186 (0.350)	**7.95 (0.44)**	44.70 (3.64)	18.96 (1.30)
		40	47.36 (1.85)	32.28 (1.57)	-0.244 (0.065)	4.800 (0.473)	8.61 (0.56)	**42.88 (1.34)**	20.46 (1.23)



(a) Hepaticvessel_294_slice_83 (b) Hepaticvessel_268_slice_27

Figure 18. Examples of vasculature tissues, where example (a) shows vessels completely out of the liver, and example (b) shows vessels inside the liver organ, but outside the liver mask.

After showing the pros and cons of applying the liver delineation, we have applied the masks for both the tumors and vessels tissues to maintain consistency; however, we plan to treat the vessels differently in the future. We now focus on the ConvNets trained on tumors and vessels segmentation after multiplying the liver masks with the CT slices. Both Table 8 and Table 9 show the deployed LRs for the tumors and vessels segmentation tasks, respectively. The portrayed results are the mean (standard deviation) of 5-fold cross-validation runs of these 5 runs. For the tumors' segmentation results shown in Table 8, it is easy to notice that changing LRs did not significantly enhance any of the metrics, albeit their effect is more noticeable in Table 7. Moreover, although the Dice score average value is similar for LR 28x10^{-5} and 150x10^{-5}, the standard deviation is much smaller for the 150x10^{-5} LR. This result shows a less consistent behavior for the trained models under the 28x10^{-5} LR run.

Table 8. Performance evaluation of tumors’ ConvNets with different LRs.

LR (10^{-5})	Dice (%)	IoU (%)	RVD	ASD (mm)	RMSD (mm)	HD (mm)	95% HD (mm)
4	57.73 (1.10)	45.19 (0.86)	**0.040 (0.162)**	25.299 (16.011)	32.15 (15.20)	89.57 (11.13)	51.03 (14.22)
20	59.02 (3.64)	46.79 (3.73)	0.174 (0.231)	24.834 (15.989)	31.23 (15.40)	80.61 (14.21)	50.58 (15.24)
28	**59.71 (3.91)**	46.94 (3.68)	0.251 (0.328)	17.126 (13.208)	23.25 (12.18)	77.39 (8.22)	38.49 (11.01)
40	57.86 (2.41)	45.22 (2.65)	0.312 (0.514)	**11.445 (1.618)**	18.04 (1.73)	**68.95 (5.63)**	38.57 (5.92)
60	57.63 (1.63)	45.18 (2.12)	0.083 (0.228)	23.249 (23.505)	28.98 (23.79)	75.07 (22.5)	44.81 (27.03)
80	58.75 (2.27)	46.38 (1.98)	0.310 (0.184)	17.490 (13.207)	23.94 (12.10)	74.02 (11.71)	42.82 (11.36)
100	57.73 (4.05)	45.60 (3.72)	0.513 (0.436)	11.991 (1.564)	18.92 (2.70)	73.10 (10.13)	39.42 (9.32)
120	59.51 (3.00)	47.08 (3.49)	0.467 (0.431)	11.954 (2.613)	**17.92 (2.99)**	72.46 (7.52)	**34.43 (3.84)**
150	59.70 (0.61)	**47.28 (1.24)**	0.229 (0.263)	11.921 (0.988)	19.04 (2.15)	73.18 (8.83)	38.82 (6.59)

It is worth re-iterating that the results here are after applying the mask. Unlike the tumors’ results, the best performance for most metrics has been achieved with the LR = 28x10^{-5}.

Table 9. Performance evaluation of vessels’ ConvNets with different LRs.

LR (10^{-5})	Dice (%)	IoU (%)	RVD	ASD (mm)	RMSD (mm)	HD (mm)	95% HD (mm)
4	48.63 (1.30)	33.18 (1.13)	-0.098 (0.133)	4.230 (0.395)	7.97 (0.54)	49.25 (6.05)	**18.34 (1.33)**
20	48.39 (2.19)	33.04 (1.81)	-0.095 (0.092)	5.091 (1.157)	9.15 (1.47)	46.59 (2.93)	21.40 (2.83)
28	**50.34 (1.82)**	**34.73 (1.52)**	**-0.045 (0.144)**	**4.186 (0.350)**	**7.95 (0.44)**	44.70 (3.64)	18.96 (1.30)
40	47.36 (1.85)	32.28 (1.57)	-0.244 (0.065)	4.800 (0.473)	8.61 (0.56)	**42.88 (1.34)**	20.46 (1.23)
60	49.72 (2.40)	34.34 (2.01)	-0.196 (0.119)	4.422 (0.732)	8.13 (0.95)	43.67 (2.12)	19.29 (2.01)
80	48.95 (3.01)	33.61 (2.63)	-0.150 (0.144)	4.559 (0.764)	8.35 (0.86)	46.80 (4.10)	19.59 (2.20)
100	47.66 (0.96)	32.39 (0.89)	-0.277 (0.102)	4.900 (0.304)	8.95 (0.50)	44.64 (3.01)	21.33 (1.09)

LR (10^{-5})	Dice (%)	IoU (%)	RVD	ASD (mm)	RMSD (mm)	HD (mm)	95% HD (mm)
120	49.03 (4.02)	33.62 (3.36)	-0.183 (0.259)	4.483 (0.757)	8.23 (0.82)	43.68 (5.04)	19.37 (1.59)
150	48.58 (1.87)	33.30 (1.61)	-0.239 (0.142)	4.843 (0.786)	8.68 (0.95)	44.36 (1.64)	20.57 (1.84)

Overall, the results are more consistent within the vessels challenge, making LR change have minimal effect, even on the standard deviation of models' performances. It is worth noting that we are not far off from the results portrayed on the MSDC challenge website for the Task 8 Hepatic Vessel challenge [146]. However, their models were created to tackle 7 tasks, and then they had the datasets for the Hepatic Vessel, Spleen, and Colon tasks to train over and submit the results once. Moreover, the submitted results appear to be a multi-class challenge, which in retrospect can be easier if it helps the model distinguish tumors from vessels tissues.

4.3 Masks Integration and 3D Interpolation Showcase

This section showcases the segmentation of the best models for each task and visually compares them to the ground-truth counterpart.

Table 10 shows the best-ever trained ConvNets from the previously conducted 5-fold cross-validation process. They are used to compare the performance of our ConvNets with the ground-truth masks throughout this section.

Table 10. Best-ever trained fold ConvNets.

TOI	Sched-uler*	LR (10^{-5})	Dice (%)	IoU (%)	RVD	ASD (mm)	RMSD (mm)	HD (mm)	95% HD (mm)
Liver	RLRP	16	98.2	96.4	-0.0109	0.407	1.44	25.69	2.14
Tumors	OCLR	28	66.0	52.9	0.0913	9.06	14.84	63.68	26.00
Vessels	OCLR	28	51.9	36.1	0.0778	3.85	7.53	45.33	17.90

* RLRP = ReduceLROnPlateau; OCLR = OneCycleLR

Figure 19 shows a comparison between “Our Segmentation” that shows the segmentation of all 3 best-trained ConvNets and the “Ground-Truth” masks. For volume hepaticvessel_268, we can see that the liver segmentation is almost flawless for both slice 20 and slice 27.

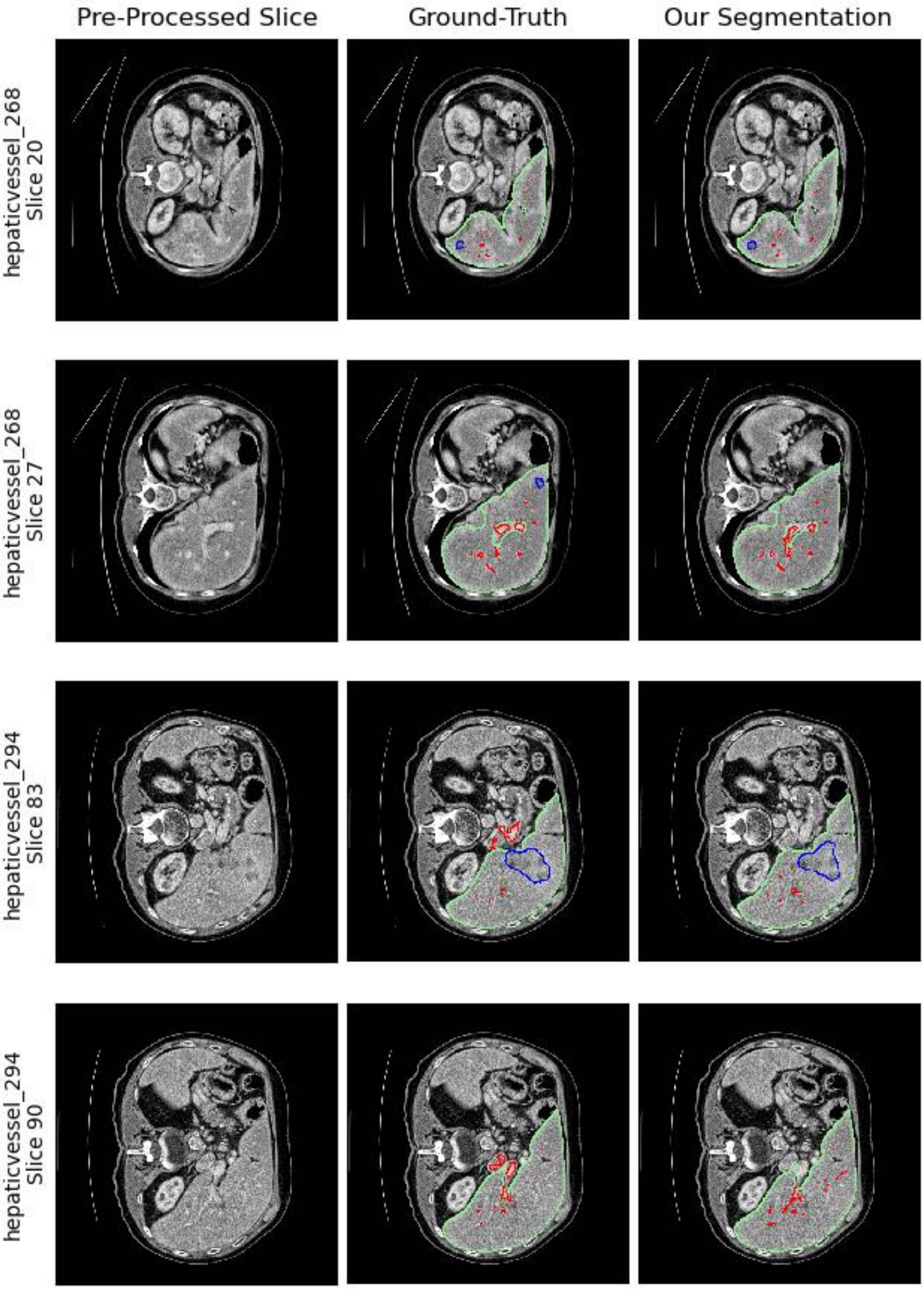
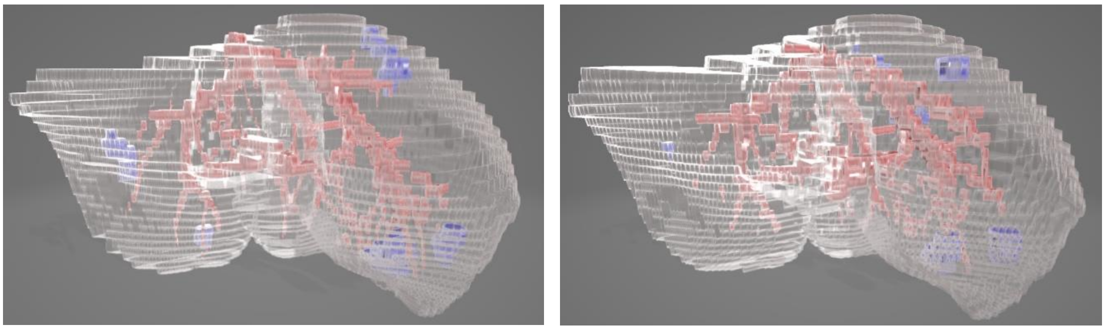


Figure 19. Comparison between our models' performance and the ground-truth masks. Green shows liver mask border, blue shows tumors mask border, and red shows vessels mask border.

For the tumors' segmentation, the ConvNet performs well in slice 20 but poorly on slice 27, missing the small tumor. We are missing some minor vessels for the vessels' segmentation in slice 20; contrastingly, the ConvNet performs significantly well, catching all the vessels, perhaps even better than the ground-truth mask. It is worth mentioning that the vessels ConvNet can catch the vessels outside the liver mask, even though we are multiplying the CT scan with the liver mask (as shown by Figure 17), effectively removing these vessels; however, the ConvNet seems to understand the existence of vessels in that area.

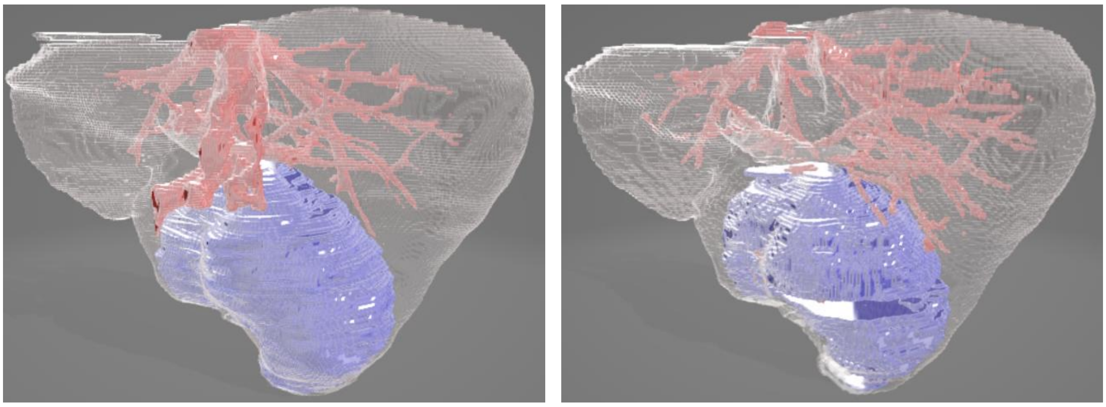
For hepaticvessel_294 volume in both slices 83 and 90, similar findings can be said about the liver segmentation, showing the powerfulness of our developed ConvNet. For tumors' segmentation in slice 83, it captures the existence of the tumor and delineates it well with a high recall. The vessels' ConvNet captures all the vessels tissues within the liver. We argue it is even better than the original mask; however, the vessels outside the liver area have not been captured due to multiplying with the liver mask. This phenomenon is an exciting finding because this tells us that multiplying with the liver mask has a different effect on the vessels depending on their whereabouts to the liver. To elaborate further, the vessels in hepaticvessel_268 are within the liver but outside the liver mask, but they are captured. However, for hepaticvessel_294, the vessels are both outside the liver and the liver mask, and thus, they were missed.

ConvNets with the best performance have been identified for each TOI (liver parenchyma, tumors, and vessels) as shown by Table 10, and the masks are generated for the whole test dataset, as shown by Figure 19. It is possible to integrate them into a 3D printable easy to visualize the object. Figure 20 and Figure 21 show the 3D interpolation of hepaticvessel_268 and herpaticvessel_294, respectively.



(a) Ground-Truth (b) Our Segmentation

Figure 20. 3D interpolation of volume hepaticvessel_268.



(a) Ground-Truth (b) Our Segmentation

Figure 21. 3D interpolation of volume hepaticvessel_294.

The findings we have explored on the 2D slices in Figure 19 are being translated into the 3D object. The vessels have been well captured within hepaticvessel_268 volume, but some tumors have been missed due to their small size. Contrastingly for hepaticvessel_294, significant vessels have not been captured due to the multiplication with the liver mask, as shown by Figure 17. Additionally, there is a discontinuity in the segmented tumor, but most of the tumor has been outlined and captured. We have not applied any post-processing techniques, but this can be easily post-processed to make it a single object and close the gap.

4.4 Clinical Real-Time Implementation

This section outlines the next steps after creating the 3 models developed to segment the 3 TOIs (liver parenchyma, tumors, and vessels). The whole reason behind working on this project is to aid surgeons, doctors, and clinicians diagnose and analyze liver-related diseases. From that perspective, such tools that aid clinicians should be designed with the idea of making it exceptionally end-user friendly and making it require a minimal amount of interactivity to avoid wasting their time. In other words, they need not know about the details of how such a tool works as long as it works accurately and generates expert-like segmentations. Moreover, it should be very intuitive, simple, and self-explanatory, such that anyone can utilize it without any extensive training. Thus, we have taken it upon ourselves to create a desktop application, as we know all the needed details to create such a tool and how it should operate in the most helpful manner.

We specifically used the PyQt5 framework to build the desktop application [147], available in Python. Qt is a comprehensive GUI toolkit that can be used to build cross-platform applications [148]. PyQt5 is the set of bindings that link the Python programming language with Qt5 (Version 5 of Qt), making the Python language a proper replacement for the C++ programming language [147]. It offers a set of classes that can be used as off-the-shelf components in any application. Lastly, a programmer can either utilize the Qt “Designer” tool to create the GUI and then import the GUI into the application along with the components within or create the components programmatically. For our case, and given that the application is reasonably simple in terms of the frontend (GUI) and quite complex in the backend, we have opted to implement it using the latter method.

Figure 22 shows the display that the end-users get prompted with when the application is opened. The application is called “Liver Tissues Segmenter” and has a liver icon for a logo.

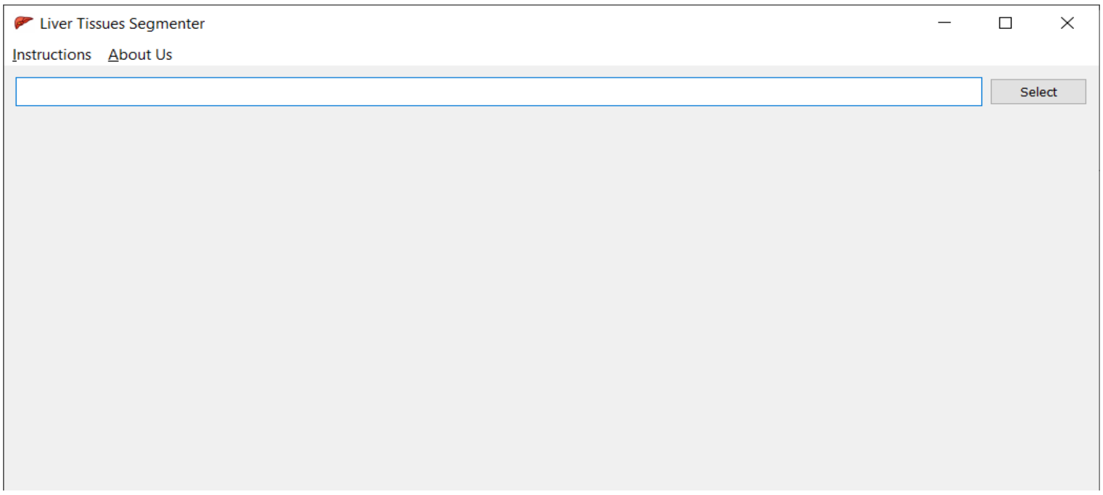
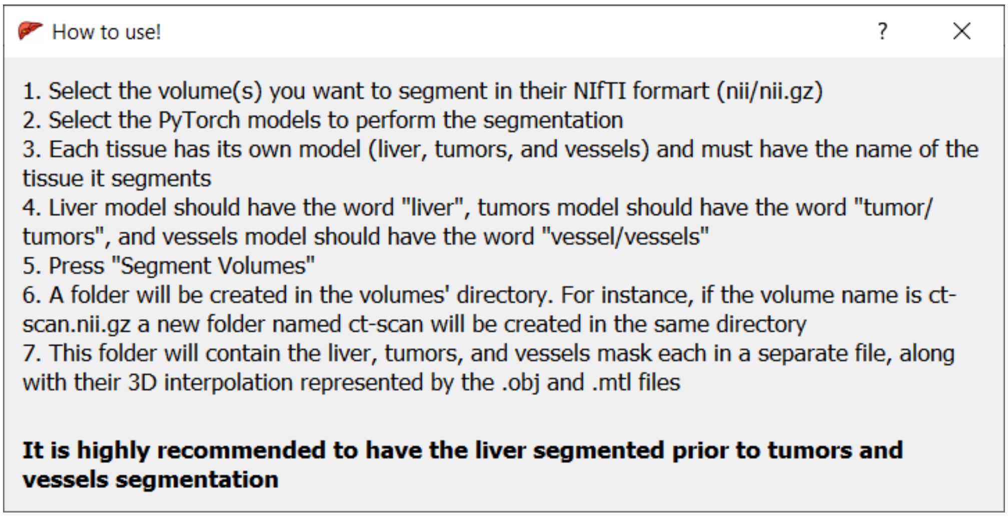


Figure 22. GUI desktop application initial screen.

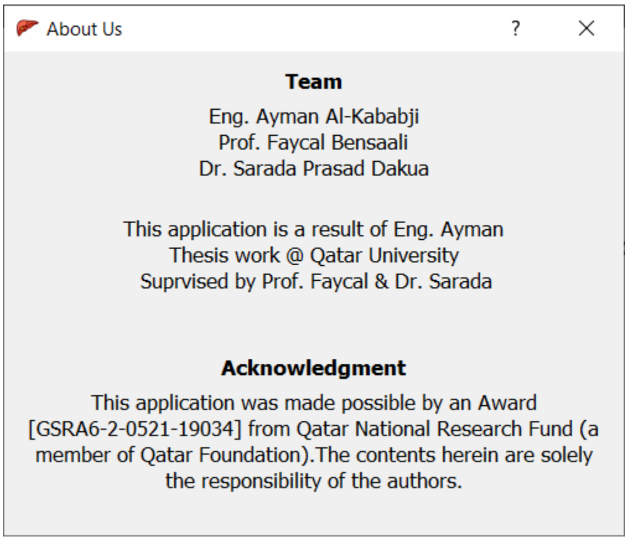
Beneath that exists the Instructions and About Us menu tabs, which the first explains how the application should be used (Figure 23), and the latter mentions the source of the application and the people behind creating it (Figure 24).



1. Select the volume(s) you want to segment in their NIfTI formart (nii/nii.gz)
2. Select the PyTorch models to perform the segmentation
3. Each tissue has its own model (liver, tumors, and vessels) and must have the name of the tissue it segments
4. Liver model should have the word "liver", tumors model should have the word "tumor/tumors", and vessels model should have the word "vessel/vessels"
5. Press "Segment Volumes"
6. A folder will be created in the volumes' directory. For instance, if the volume name is ct-scan.nii.gz a new folder named ct-scan will be created in the same directory
7. This folder will contain the liver, tumors, and vessels mask each in a separate file, along with their 3D interpolation represented by the .obj and .mtl files

It is highly recommended to have the liver segmented prior to tumors and vessels segmentation

Figure 23. Instructions on how to use the desktop application.



Team

Eng. Ayman Al-Kababji
Prof. Faycal Bensaali
Dr. Sarada Prasad Dakua

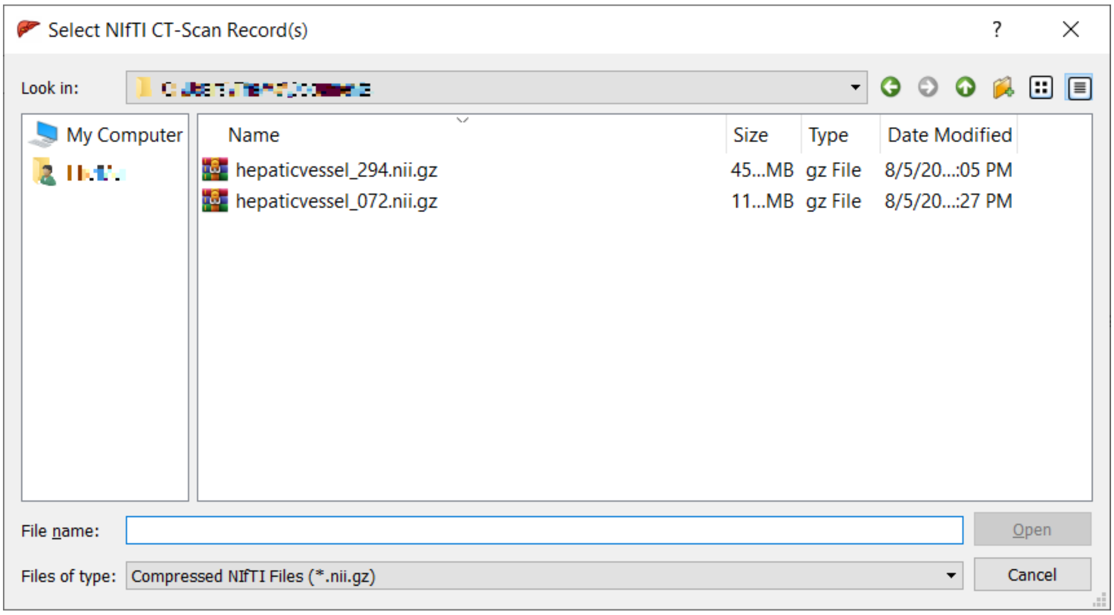
This application is a result of Eng. Ayman
Thesis work @ Qatar University
Suprvised by Prof. Faycal & Dr. Sarada

Acknowledgment

This application was made possible by an Award [GSRA6-2-0521-19034] from Qatar National Research Fund (a member of Qatar Foundation).The contents herein are solely the responsibility of the authors.

Figure 24. The team behind creating this application.

When end-users press on the "Select" button, a prompt window appears, enabling them to select multiple CT scans (saved as NIfTI format). Figure 25 shows the files that are can only be imported, which need to be either nii/nii.gz, where it will not show any other file format.



Look in: [redacted]

[redacted]

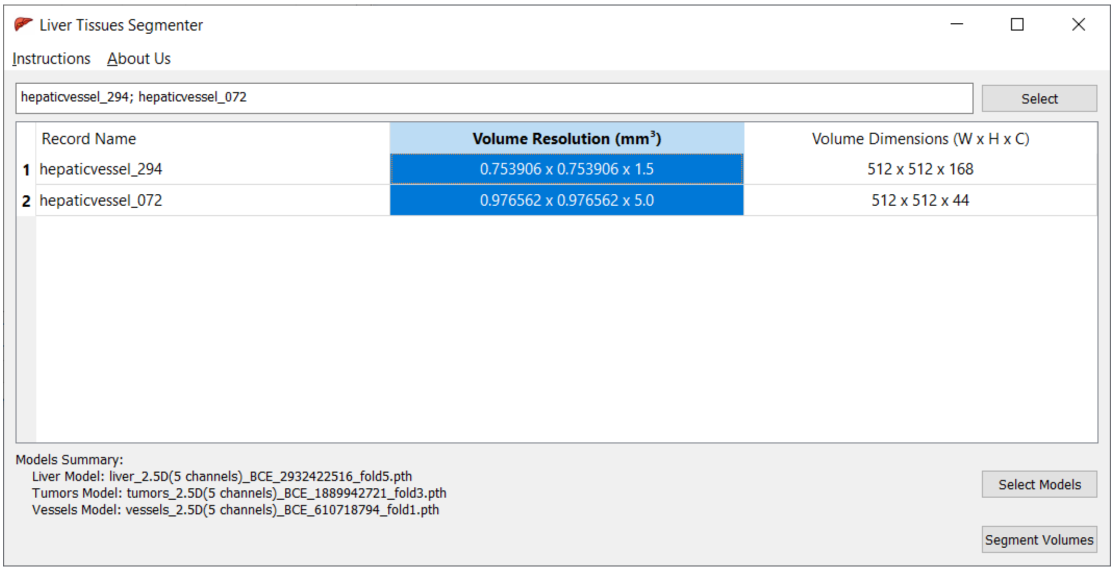
Name	Size	Type	Date Modified
hepaticvessel_294.nii.gz	45...MB	gz File	8/5/20...:05 PM
hepaticvessel_072.nii.gz	11...MB	gz File	8/5/20...:27 PM

File name:

Files of type: Compressed NIfTI Files (*.nii.gz)

Figure 25. A prompt window to select the CT scans of NIfTI file type.

Once CT scans are selected, a table shows some of the records' metadata for the end-user to see. Moreover, the user will have to select the PyTorch models trained to segment the liver, the tumors, and the vessels. Once selected, the application is all set to perform the segmentation. Figure 26 shows the effect of selecting multiple volumes for delineation and the selected models to perform the segmentation. The volumes are shown in a scrollable table such that it is scalable for as many CT records imported to the application.



hepaticvessel_294; hepaticvessel_072

	Record Name	Volume Resolution (mm^3)	Volume Dimensions (W x H x C)
1	hepaticvessel_294	0.753906 x 0.753906 x 1.5	512 x 512 x 168
2	hepaticvessel_072	0.976562 x 0.976562 x 5.0	512 x 512 x 44

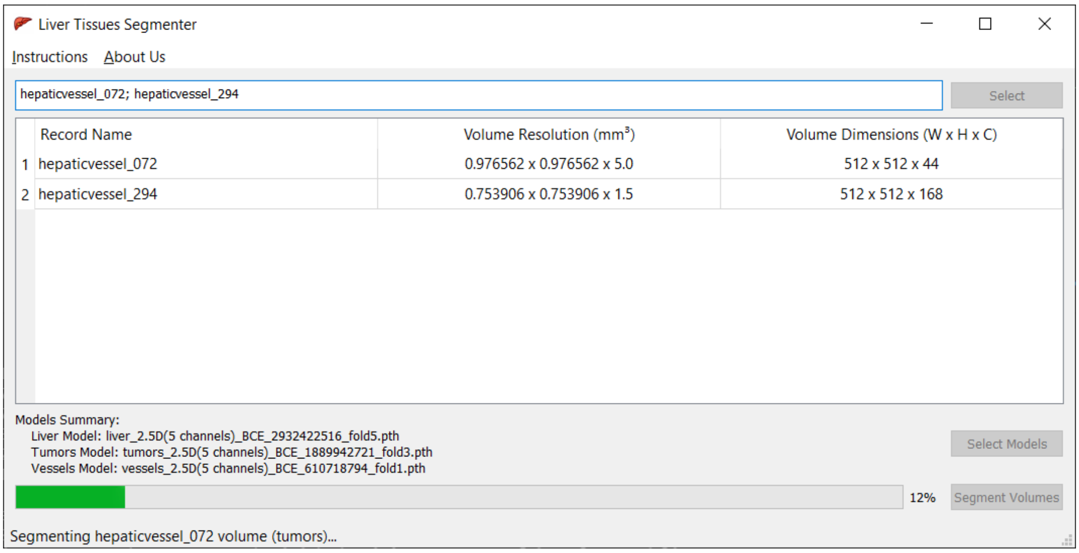
Models Summary:
Liver Model: liver_2.5D(5 channels)_BCE_2932422516_fold5.pth
Tumors Model: tumors_2.5D(5 channels)_BCE_1889942721_fold3.pth
Vessels Model: vessels_2.5D(5 channels)_BCE_610718794_fold1.pth

Figure 26. GUI after selecting to-be-segmented volumes and segmentation models.

The record name is displayed for each selected volume and its resolution and dimensions, which specify the distance between adjacent voxels and the number of available voxels, respectively.

Once the "Segment Volumes" button is pressed, the heavy-backend processes commence. Volumes go through the same preprocessing techniques that were applied to volumes when the models were being trained. This preprocessing stage includes volume rotation, rescaling, clipping, standardization, contrast enhancement via

CLAHE, normalization, and transformation into 2.5D inputs. They are all done in real-time and quite efficiently, especially when using the GPU. It is worth mentioning that cramming all the computations on the main thread is considered bad practice and should be avoided at all costs since it is responsible for all the GUI interactions. Since these processes take a considerate amount of time, it is opted to make all the segmentation computations on another thread to avoid freezing the GUI. Figure 27 illustrates the progress made by the application via the progress bar and the status bar, which show the volume and the phase the application is currently processing. Showing progress enables clinicians to plan their time accordingly.



hepaticvessel_072; hepaticvessel_294

	Record Name	Volume Resolution (mm^3)	Volume Dimensions (W x H x C)
1	hepaticvessel_072	0.976562 x 0.976562 x 5.0	512 x 512 x 44
2	hepaticvessel_294	0.753906 x 0.753906 x 1.5	512 x 512 x 168

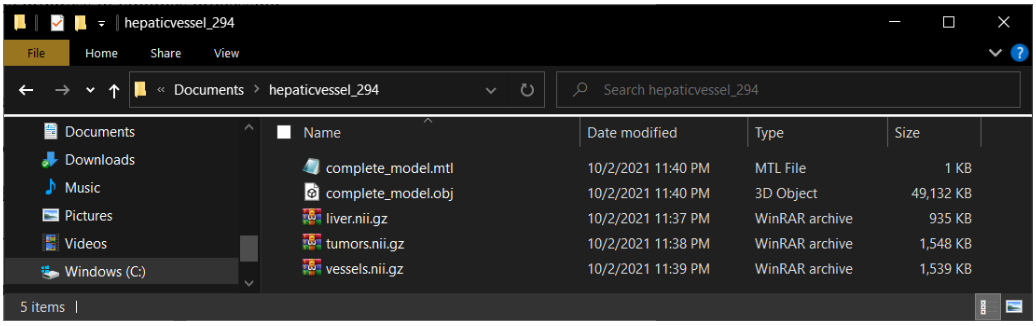
Models Summary:
Liver Model: liver_2.5D(5 channels)_BCE_2932422516_fold5.pth
Tumors Model: tumors_2.5D(5 channels)_BCE_1889942721_fold3.pth
Vessels Model: vessels_2.5D(5 channels)_BCE_610718794_fold1.pth

12%

Segmenting hepaticvessel_072 volume (tumors)...

Figure 27. Showing the progress in volumes processing.

Once the segmented liver, tumors, and vessels of a single volume are completed, the 3D interpolation process for the 3 masks is commenced. The resulting masks and the 3D object are placed in a folder holding the same name as the segmented volume within the same directory. Figure 28 displays the resulting files after processing volume 294 from the MSDC-T8 dataset.



hepaticvessel_294

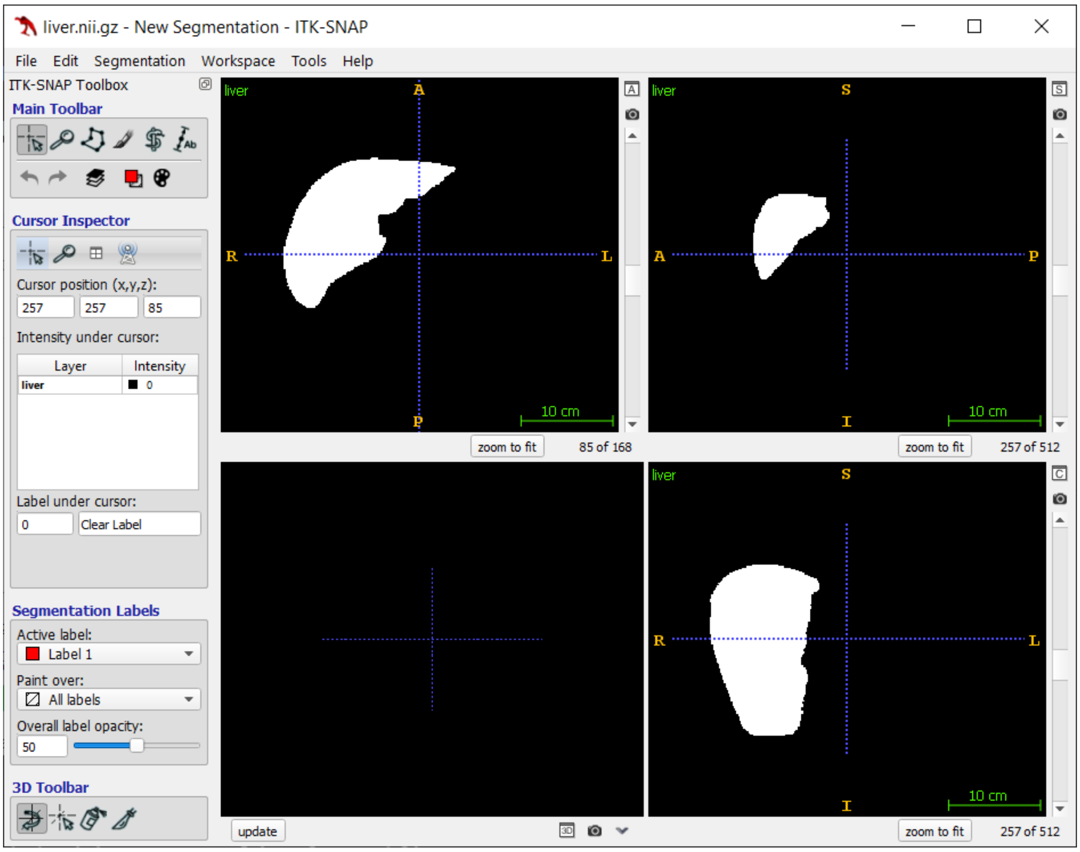
Documents › hepaticvessel_294

Name	Date modified	Type	Size
complete_model.mtl	10/2/2021 11:40 PM	MTL File	1 KB
complete_model.obj	10/2/2021 11:40 PM	3D Object	49,132 KB
liver.nii.gz	10/2/2021 11:37 PM	WinRAR archive	935 KB
tumors.nii.gz	10/2/2021 11:38 PM	WinRAR archive	1,548 KB
vessels.nii.gz	10/2/2021 11:39 PM	WinRAR archive	1,539 KB

5 items

Figure 28. Resulting files for a particular CT scan/volume.

The resulting files can be used for numerous applications. The liver.nii.gz, tumors.nii.gz, and vessels.nii.gz have the segmented tissue in a separate file for easier access, such that any medical software can open these masks such as the ITK-Snap software, as seen in Figure 29.



liver.nii.gz - New Segmentation - ITK-SNAP

Cursor position (x,y,z): 257 257 85

Layer	Intensity
liver	■ 0

Label under cursor: 0 Clear Label

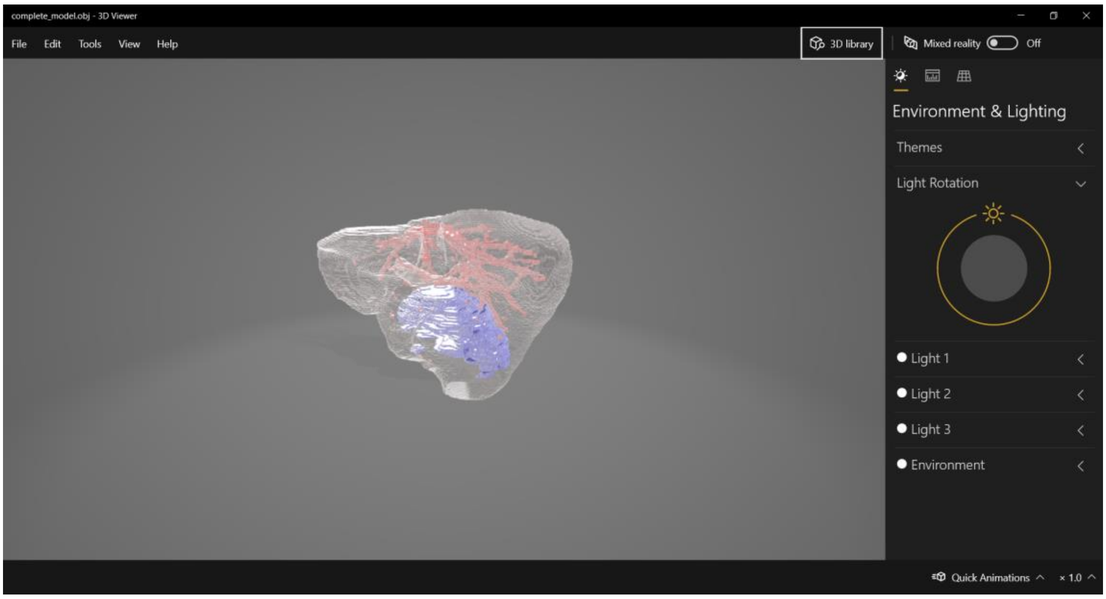
Active label: Label 1

Paint over: All labels

Overall label opacity: 50

Figure 29. Opening segmented liver.nii.gz mask on ITK-Snap software.

Moreover, the complete_model.obj has all the vertices and faces needed to visualize the volume in 3D form. Lastly, the .mtl file adds color to different tissues to easily differentiate between them. Exporting both the .obj file and the .mtl file is the best route for multi-colored 3D printing [143]. The 3D interpolated liver volumes can be immediately put into a 3D printer, and the clinicians can get the liver volume printed. In the absence of such 3D printers, or to reduce wasted plastic, clinicians can open the 3D interpolated object using the "3D Paint" program on any computer to rotate it along any axis (as seen in Figure 30).



Mixed reality Off

× 1.0

Figure 30. Opening 3D interpolated object on 3D Paint software in Windows 10.

To conclude, by creating this application, we provide clinicians an end-to-end user-friendly GUI tool for liver tissues' segmentation that enables them to segment and 3D interpolate multiple records sequentially for medical analysis, with minimal possible human intervention. All they need to do is select the volume(s), select the models, and they will have to wait around 1 – 2 minutes for each volume to be completed. It is worth re-iterating that volumes are done sequentially, meaning the doctors can look into the first processed volume while waiting for the remaining ones.

CHAPTER 5: CONCLUSION AND FUTURE WORK

Segmenting the liver, its tumors, and vessels plays a vital role in saving the lives of patients suffering from hepatic-related diseases. Moreover, it is crucial due to the tremendous fatality rate that accompanies such diseases. To this end, we surveyed the literature to understand the current state-of-the-art techniques and results that have been achieved in 2014 – 2020. From the surveyed studies, it was noticed that there is still room for improving the segmentation performance, especially for the tumors and vessels segmentation tasks. Additionally, there was a lack of studies that segment all those tissues in the same system to generate a complete liver delineation. Fortunately, we have found that a dataset called MSDC-T8 has all the necessary tissues to perform the complete delineation of the liver; thus, we can develop such models.

Our methodology focuses on preprocessing the CT scans and transforming them into 2.5D input to a liver segmentation U-Net ConvNet. Model’s output is used to localize and delineate the liver for the following tumors and vessels ConvNets. The liver segmentation U-Net generated the state-of-the-art performance by scoring an average of 98.12% Dice on a test set containing 23 records from the MSDC-T8 dataset. Moreover, out of those 23 records, the tumors and vessels' ground-truth masks are available for 14, making it a viable solution to test the tumors and vessels segmentation ConvNets. An average of ~60% Dice score is achieved for the tumors’ segmentation, where the best model achieved a 65.95% in the Dice score. Moreover, an average of ~50% Dice score has been achieved for the vessels’ segmentation, where the best model achieved a 51.94% in the Dice score metric.

Once all masks have been acquired, they are combined and then interpolated, using “The Marching Cubes” algorithm, to create a 3D object that is 3D printable by creating both the .obj and .mtl files. Lastly, a desktop application has also been created using the PyQt5 framework to deploy the devised algorithms to help and assist clinicians diagnose hepatic-related diseases. They have to feed the NIfTI volumes to the desktop application, and the output will be the mask of each tissue, along with the 3D object interpolation of said tissues.

Some of the limitations in our methodology include the mistreatment of vessels' voxels by eliminating them when multiplying the liver mask with the CT slices. Moreover, we have not "thoroughly" investigated other ConvNet architecture, which would have added more value to our work. Lastly, it is better to use PySide2 instead of PyQt5, as the license of PySide2 is slightly more flexible.

In the future, we plan to enhance the performance of both the tumors and vessels segmentation algorithms to surpass the state-of-the-art results available in the literature. We plan to optimize the training area for the vessels ConvNet by better methodology than simply multiplying with the liver mask. Moreover, hepaticvessel_072 was always an issue for our tumors ConvNets, and their performance was always low on this record. Hence, understanding why this happens will significantly increase the tumor ConvNets performance (by 4 – 5%). Additionally, we plan to explore ways of smoothening the surface of the interpolated liver, especially for those volumes with significant inter-slice distance (depth), where the interpolated liver has staircase borders.

REFERENCES

[1] S. K. Asrani, H. Devarbhavi, J. Eaton, and P. S. Kamath, “Burden of liver diseases in the world,” *J. Hepatol.*, vol. 70, no. 1, pp. 151–171, 2019, doi: 10.1016/j.jhep.2018.09.014.

[2] U. D. of Surgery, “Liver Metastases,” *UCSF Department of Surgery*. https://surgery.ucsf.edu/conditions--procedures/liver-metastases.aspx#:~:text=After the lymph nodes%2C the,and some less common sites.&text=The majority of liver metastases present as multiple tumors. (accessed Aug. 27, 2020).

[3] P. F. Christ *et al.*, “Automatic Liver and Tumor Segmentation of CT and MRI Volumes using Cascaded Fully Convolutional Neural Networks,” pp. 1–20, 2017, [Online]. Available: http://arxiv.org/abs/1702.05970.

[4] A. Gotra *et al.*, “Liver segmentation: indications, techniques and future directions,” *Insights Imaging*, vol. 8, no. 4, pp. 377–392, Aug. 2017, doi: 10.1007/s13244-017-0558-1.

[5] P. Campadelli, E. Casiraghi, and A. Esposito, “Liver segmentation from computed tomography scans: A survey and a new algorithm,” *Artif. Intell. Med.*, vol. 45, no. 2–3, pp. 185–196, 2009, doi: 10.1016/j.artmed.2008.07.020.

[6] L. Chang, “CT Scan (CAT Scan): Purpose, Procedure, Risks, Side-Effects, Results,” *WebMD*, Dec. 20, 2018. https://www.webmd.com/cancer/what-is-a-ct-scan#2 (accessed Mar. 26, 2020).

[7] Y. Wang, Y. Zhou, P. Tang, W. Shen, E. K. Fishman, and A. L. Yuille, “Training Multi-organ Segmentation Networks with Sample Selection by Relaxed Upper Confident Bound,” in *Medical Image Computing and Computer Assisted Intervention -- MICCAI 2018*, 2018, pp. 434–442, doi: 10.1007/978-3-

030-00937-3.

[8] F. Liang *et al.*, “Abdominal, multi-organ, auto-contouring method for online adaptive magnetic resonance guided radiotherapy: An intelligent, multi-level fusion approach,” *Artif. Intell. Med.*, vol. 90, pp. 34–41, 2018, doi: 10.1016/j.artmed.2018.07.001.

[9] E. Gibson *et al.*, “Automatic Multi-Organ Segmentation on Abdominal CT with Dense V-Networks,” *IEEE Trans. Med. Imaging*, vol. 37, no. 8, pp. 1822–1834, 2018, doi: 10.1109/TMI.2018.2806309.

[10] M. P. Heinrich, “Closing the Gap Between Deep and Conventional Image Registration Using Probabilistic Dense Displacement Networks,” in *Medical Image Computing and Computer Assisted Intervention -- MICCAI 2019*, Oct. 2019, vol. 11769 LNCS, pp. 50–58, doi: 10.1007/978-3-030-32226-7_6.

[11] S. Zheng, B. Fang, L. Li, M. Gao, Y. Wang, and K. Peng, “Automatic Liver Lesion Segmentation in CT Combining Fully Convolutional Networks and Non-negative Matrix Factorization,” in *Imaging for Patient-Customized Simulations and Systems for Point-of-Care Ultrasound, International Workshops, BIVPCS 2017 and POCUS 2017*, 2017, pp. 44–51, doi: 10.1007/978-3-319-67552-7_6.

[12] P. Hu, F. Wu, J. Peng, P. Liang, and D. Kong, “Automatic 3D liver segmentation based on deep learning and globally optimized surface evolution,” *Phys. Med. Biol.*, vol. 61, no. 24, pp. 8676–8698, 2016, doi: 10.1088/1361-6560/61/24/8676.

[13] D. Yang *et al.*, “Automatic Liver Segmentation Using an Adversarial Image-to-Image Network,” in *Medical Image Computing and Computer-Assisted Intervention -- MICCAI 2017*, 2017, pp. 507–515, doi: 10.1007/978-3-319-

66179-7_58.

[14] Q. Huang, H. Ding, X. Wang, and G. Wang, “Fully automatic liver segmentation in CT images using modified graph cuts and feature detection,” *Comput. Biol. Med.*, vol. 95, no. February, pp. 198–208, 2018, doi: 10.1016/j.compbiomed.2018.02.012.

[15] O. V. Ivashchenko *et al.*, “A workflow for automated segmentation of the liver surface, hepatic vasculature and biliary tree anatomy from multiphase MR images,” *Magn. Reson. Imaging*, vol. 68, no. May 2019, pp. 53–65, 2020, doi: 10.1016/j.mri.2019.12.008.

[16] H. Alahmer and A. Ahmed, “Computer-aided Classification of Liver Lesions from CT Images Based on Multiple ROI,” *Procedia Comput. Sci.*, vol. 90, pp. 80–86, 2016, doi: 10.1016/j.procs.2016.07.027.

[17] A.-R. Ali, M. Couceiro, A. M. Anter, A. E. Hassanien, M. F. Tolba, and V. Snášel, “Liver CT Image Segmentation with an Optimum Threshold Using Measure of Fuzziness,” in *Proceedings of the Fifth International Conference on Innovations in Bio-Inspired Computing and Applications IBICA 2014*, 2014, pp. 83–92.

[18] G. I. Sayed, A. E. Hassanien, and G. Schaefer, “An Automated Computer-aided Diagnosis System for Abdominal CT Liver Images,” *Procedia Comput. Sci.*, vol. 90, pp. 68–73, 2016, doi: 10.1016/j.procs.2016.07.012.

[19] A. M. Anter and A. E. Hassenian, “Computational intelligence optimization approach based on particle swarm optimizer and neutrosophic set for abdominal CT liver tumor segmentation,” *J. Comput. Sci.*, vol. 25, pp. 376–387, 2018, doi: 10.1016/j.jocs.2018.01.003.

[20] J. Cai, “Segmentation and Diagnosis of Liver Carcinoma Based on Adaptive

Scale-Kernel Fuzzy Clustering Model for CT Images," *J. Med. Syst.*, vol. 43, no. 11, 2019, doi: 10.1007/s10916-019-1459-2.

[21] H. Lombaert, D. Zikic, A. Criminisi, and N. Ayache, "Laplacian Forests: Semantic Image Segmentation by Guided Bagging," in *Medical Image Computing and Computer-Assisted Intervention -- MICCAI 2014*, 2014, pp. 496–504, doi: 10.1007/978-3-319-10470-6_62.

[22] T. Norajitra and K. H. Maier-Hein, "3D Statistical Shape Models Incorporating Landmark-Wise Random Regression Forests for Omni-Directional Landmark Detection," *IEEE Trans. Med. Imaging*, vol. 36, no. 1, pp. 155–168, 2017, doi: 10.1109/TMI.2016.2600502.

[23] J. Treilhard *et al.*, "Liver Tissue Classification in Patients with Hepatocellular Carcinoma by Fusing Structured and Rotationally Invariant Context Representation," in *Medical Image Computing and Computer-Assisted Intervention -- MICCAI 2017*, 2017, pp. 81–88, doi: 10.1007/978-3-319-66179-7_10.

[24] P. Zhang, J. Yang, D. Ai, Z. Xie, and Y. Liu, "Learning Based Random Walks for Automatic Liver Segmentation in CT Image," in *Chinese Conference on Image and Graphics Technologies*, 2015, pp. 251–259, doi: 10.1007/978-3-662-47791-5.

[25] Y. Zheng *et al.*, "Automatic liver segmentation based on appearance and context information," *Biomed. Eng. Online*, vol. 16, no. 1, pp. 1–12, 2017, doi: 10.1186/s12938-016-0296-5.

[26] B. He *et al.*, "Fast automatic 3D liver segmentation based on a three-level AdaBoost-guided active shape model," *Med. Phys.*, vol. 43, no. 5, pp. 2421–2434, 2016, doi: 10.1118/1.4946817.

[27] Y. Chen, S. Li, S. Yang, and W. Luo, "Liver Segmentation in CT Images with Adversarial Learning," in *International Conference on Intelligent Computing*, 2019, pp. 470–480, doi: 10.1007/978-3-030-26763-6_45.

[28] H. Zheng *et al.*, "Semi-supervised Segmentation of Liver Using Adversarial Learning with Deep Atlas Prior," in *Medical Image Computing and Computer Assisted Intervention -- MICCAI 2019*, 2019, pp. 148–156, doi: 10.1007/978-3-030-32226-7_17.

[29] J. Tian, C. Li, Z. Shi, and F. Xu, "A Diagnostic Report Generator from CT Volumes on Liver Tumor with Semi-supervised Attention Mechanism," in *Medical Image Computing and Computer Assisted Intervention -- MICCAI 2018*, 2018, pp. 702–710, doi: 10.1007/978-3-030-00934-2_78.

[30] Y. Yuan, "Hierarchical Convolutional-Deconvolutional Neural Networks for Automatic Liver and Tumor Segmentation," pp. 1–4, 2017, [Online]. Available: http://arxiv.org/abs/1710.04540.

[31] M. J. A. Jansen *et al.*, "Liver segmentation and metastases detection in MR images using convolutional neural networks," *J. Med. Imaging*, vol. 6, no. 4, p. 1, 2019, doi: 10.1117/1.jmi.6.4.044003.

[32] W. Qin *et al.*, "Superpixel-based and boundary sensitive convolutional neural network for automated liver segmentation," *Phys. Med. Biol.*, vol. 63, no. 9, 2018, doi: 10.1088/1361-6560/aabd19.

[33] 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*, 2015, vol. 9351, pp. 234–241, doi: 10.1007/978-3-319-24574-4_28.

[34] P. F. Christ *et al.*, "Automatic Liver and Lesion Segmentation in CT Using

Cascaded Fully Convolutional Neural Networks and 3D Conditional Random Fields," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2016*, Oct. 2016, pp. 415–423, doi: 10.1007/978-3-319-46723-8_48.

[35] F. Ouhmich, V. Agnus, V. Noblet, F. Heitz, and P. Pessaux, "Liver tissue segmentation in multiphase CT scans using cascaded convolutional neural networks," *Int. J. Comput. Assist. Radiol. Surg.*, vol. 14, no. 8, pp. 1275–1284, 2019, doi: 10.1007/s11548-019-01989-z.

[36] N. Nanda, P. Kakkar, and S. Nagpal, "Computer-Aided Segmentation of Liver Lesions in CT Scans Using Cascaded Convolutional Neural Networks and Genetically Optimised Classifier," *Arab. J. Sci. Eng.*, vol. 44, no. 4, pp. 4049–4062, Apr. 2019, doi: 10.1007/s13369-019-03735-8.

[37] Y. Zhang *et al.*, "Deep Learning Initialized and Gradient Enhanced Level-Set Based Segmentation for Liver Tumor from CT Images," *IEEE Access*, vol. 8, pp. 76056–76068, 2020, doi: 10.1109/ACCESS.2020.2988647.

[38] A. Mendizabal, P. Márquez-Neila, and S. Cotin, "Simulation of hyperelastic materials in real-time using deep learning," *Med. Image Anal.*, vol. 59, p. 101569, 2020, doi: 10.1016/j.media.2019.101569.

[39] Y. Wang *et al.*, "A two-step automated quality assessment for liver MR images based on convolutional neural network," *Eur. J. Radiol.*, vol. 124, no. January, p. 108822, 2020, doi: 10.1016/j.ejrad.2020.108822.

[40] B. Irving *et al.*, "Deep Quantitative Liver Segmentation and Vessel Exclusion to Assist in Liver Assessment," in *Annual Conference on Medical Image Understanding and Analysis*, 2017, pp. 663–673, doi: 10.1007/978-3-319-60964-5_58.

[41] M. Liu *et al.*, "Channel width optimized neural networks for liver and vessel

segmentation in liver iron quantification," *Comput. Biol. Med.*, vol. 122, no. May, p. 103798, 2020, doi: 10.1016/j.compbiomed.2020.103798.

[42] K. Wang *et al.*, "Automated CT and MRI Liver Segmentation and Biometry Using a Generalized Convolutional Neural Network," *Radiol. Artif. Intell.*, vol. 1, no. 2, p. 180022, 2019, doi: 10.1148/ryai.2019180022.

[43] E. Vorontsov, A. Tang, C. Pal, and S. Kadoury, "Liver lesion segmentation informed by joint liver segmentation," in *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*, 2018, pp. 1332–1335, doi: 10.1109/ISBI.2018.8363817.

[44] A. Maaref *et al.*, "Predicting the Response to FOLFOX-Based Chemotherapy Regimen from Untreated Liver Metastases on Baseline CT: a Deep Neural Network Approach," *J. Digit. Imaging*, 2020, doi: 10.1007/s10278-020-00332-2.

[45] H. Seo, C. Huang, M. Bassenne, R. Xiao, and L. Xing, "Modified U-Net (mU-Net) with Incorporation of Object-Dependent High Level Features for Improved Liver and Liver-Tumor Segmentation in CT Images," *IEEE Trans. Med. Imaging*, vol. 39, no. 5, pp. 1316–1325, 2020, doi: 10.1109/TMI.2019.2948320.

[46] H. Guo and D. Yang, "PRDNet: Medical image segmentation based on parallel residual and dilated network," *Measurement*, no. October, 2020, doi: 10.1016/j.measurement.2020.108661.

[47] C. Wang *et al.*, "Automatic Liver Segmentation Using Multi-plane Integrated Fully Convolutional Neural Networks," *Proc. - 2018 IEEE Int. Conf. Bioinforma. Biomed. BIBM 2018*, pp. 518–523, 2019, doi: 10.1109/BIBM.2018.8621257.

[48] L. Chen *et al.*, “Liver tumor segmentation in CT volumes using an adversarial densely connected network,” *BMC Bioinformatics*, vol. 20, no. Suppl 16, pp. 1–13, 2019, doi: 10.1186/s12859-019-3069-x.

[49] M. Perslev, E. B. Dam, A. Pai, and C. Igel, “One Network to Segment Them All: A General, Lightweight System for Accurate 3D Medical Image Segmentation,” in *Medical Image Computing and Computer Assisted Intervention -- MICCAI 2019*, 2019, pp. 30–38, doi: 10.1007/978-3-030-32245-8_4.

[50] J. Yang, N. C. Dvornek, F. Zhang, J. Chapiro, M. De Lin, and J. S. Duncan, “Unsupervised Domain Adaptation via Disentangled Representations: Application to Cross-Modality Liver Segmentation,” in *Medical Image Computing and Computer Assisted Intervention -- MICCAI 2019*, 2019, pp. 255–263, doi: 10.1007/978-3-030-32245-8_29.

[51] M. N. Cheema, A. Nazir, B. Sheng, P. Li, J. Qin, and D. D. Feng, “Liver Extraction Using Residual Convolution Neural Networks From Low-Dose CT Images,” *IEEE Trans. Biomed. Eng.*, vol. 66, no. 9, pp. 2641–2650, 2019.

[52] Y. Zhang, S. Miao, T. Mansi, and R. Liao, “Task Driven Generative Modeling for Unsupervised Domain Adaptation: Application to X-ray Image Segmentation,” in *Medical Image Computing and Computer Assisted Intervention -- MICCAI 2018*, 2018, pp. 599–607, doi: 10.1007/978-3-030-00934-2_67.

[53] X. F. Xi, L. Wang, V. S. Sheng, Z. Cui, B. Fu, and F. Hu, “Cascade U-ResNets for Simultaneous Liver and Lesion Segmentation,” *IEEE Access*, vol. 8, pp. 68944–68952, 2020, doi: 10.1109/ACCESS.2020.2985671.

[54] X. Han, “Automatic Liver Lesion Segmentation using a Deep Convolutional

Neural Network Method," 2017, [Online]. Available: https://arxiv.org/abs/1704.07239.

[55] S. H. Ahn *et al.*, "Comparative clinical evaluation of atlas and deep-learning-based auto-segmentation of organ structures in liver cancer," *Radiat. Oncol.*, vol. 14, no. 1, pp. 1–13, 2019, doi: 10.1186/s13014-019-1392-z.

[56] T. He, J. Hu, Y. Song, J. Guo, and Z. Yi, "Multi-task learning for the segmentation of organs at risk with label dependence," *Med. Image Anal.*, vol. 61, 2020, doi: 10.1016/j.media.2020.101666.

[57] J. Tian, L. Liu, Z. Shi, and F. Xu, "Automatic Couinaud Segmentation from CT Volumes on Liver Using GLC-UNet," in *Medical Image Computing and Computer Assisted Intervention -- MICCAI 2019*, 2019, pp. 274–282, doi: 10.1007/978-3-030-32692-0_32.

[58] X. Fang, S. Xu, B. J. Wood, and P. Yan, "Deep learning-based liver segmentation for fusion-guided intervention," *Int. J. Comput. Assist. Radiol. Surg.*, vol. 15, no. 6, pp. 963–972, 2020, doi: 10.1007/s11548-020-02147-6.

[59] X. Fang and P. Yan, "Multi-Organ Segmentation Over Partially Labeled Datasets With Multi-Scale Feature Abstraction," *IEEE Trans. Med. Imaging*, vol. 39, no. 11, pp. 3619–3629, 2020, doi: 10.1109/TMI.2020.3001036.

[60] X. Wang, S. Han, Y. Chen, D. Gao, and N. Vasconcelos, "Volumetric Attention for 3D Medical Image Segmentation and Detection," in *Medical Image Computing and Computer Assisted Intervention -- MICCAI 2019*, 2019, pp. 175–184, doi: 10.1007/978-3-030-32226-7_20.

[61] K. Xia, H. Yin, P. Qian, Y. Jiang, and S. Wang, "Liver Semantic Segmentation Algorithm Based on Improved Deep Adversarial Networks in Combination of Weighted Loss Function on Abdominal CT Images," *IEEE Access*, vol. 7, pp.

96349–96358, 2019, doi: 10.1109/ACCESS.2019.2929270.

[62] W. Tang, D. Zou, S. Yang, J. Shi, J. Dan, and G. Song, “A two-stage approach for automatic liver segmentation with Faster R-CNN and DeepLab,” *Neural Comput. Appl.*, vol. 32, no. 11, pp. 6769–6778, 2020, doi: 10.1007/s00521-019-04700-0.

[63] R. Dey and Y. Hong, “Hybrid Cascaded Neural Network for Liver Lesion Segmentation,” in *2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI)*, 2020, pp. 1173–1177, doi: 10.1109/ISBI45749.2020.9098656.

[64] R. Dey and Y. Hong, “CompNet: Complementary segmentation network for brain MRI extraction,” in *Medical Image Computing and Computer Assisted Intervention -- MICCAI 2018*, Sep. 2018, vol. 11072 LNCS, pp. 628–636, doi: 10.1007/978-3-030-00931-1_72.

[65] R. Wang, S. Cao, K. Ma, D. Meng, and Y. Zheng, “Pairwise Semantic Segmentation via Conjugate Fully Convolutional Network,” in *Medical Image Computing and Computer Assisted Intervention -- MICCAI 2019*, 2019, pp. 157–165, doi: 10.1007/978-3-030-32226-7_18.

[66] M. Ahmad *et al.*, “Deep Belief Network Modeling for Automatic Liver Segmentation,” *IEEE Access*, vol. 7, pp. 20585–20595, 2019, doi: 10.1109/ACCESS.2019.2896961.

[67] F. Lu, F. Wu, P. Hu, Z. Peng, and D. Kong, “Automatic 3D liver location and segmentation via convolutional neural network and graph cut,” *Int. J. Comput. Assist. Radiol. Surg.*, vol. 12, no. 2, pp. 171–182, 2016, doi: 10.1007/s11548-016-1467-3.

[68] P. Hu, F. Wu, J. Peng, Y. Bao, F. Chen, and D. Kong, “Automatic abdominal multi-organ segmentation using deep convolutional neural network and time-

implicit level sets," *Int. J. Comput. Assist. Radiol. Surg.*, vol. 12, no. 3, pp. 399–411, 2017, doi: 10.1007/s11548-016-1501-5.

[69] Q. Dou, H. Chen, Y. Jin, L. Yu, J. Qin, and P.-A. Heng, "3D Deeply Supervised Network for Automatic Liver Segmentation from CT Volumes," in *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2016*, 2016, pp. 149–157, doi: 10.1007/978-3-319-46723-8_18.

[70] Q. Dou *et al.*, "3D deeply supervised network for automated segmentation of volumetric medical images," *Med. Image Anal.*, vol. 41, pp. 40–54, 2017, doi: 10.1016/j.media.2017.05.001.

[71] X. Tang *et al.*, "Whole liver segmentation based on deep learning and manual adjustment for clinical use in SIRT," *Eur. J. Nucl. Med. Mol. Imaging*, 2020, doi: 10.1007/s00259-020-04800-3.

[72] K. Kamnitsas *et al.*, "Efficient multi-scale 3D CNN with fully connected CRF for accurate brain lesion segmentation," *Med. Image Anal.*, vol. 36, pp. 61–78, Feb. 2017, doi: 10.1016/j.media.2016.10.004.

[73] E. Gibson *et al.*, "Towards Image-Guided Pancreas and Biliary Endoscopy: Automatic Multi-organ Segmentation on Abdominal CT with Dense Dilated Networks," in *Medical Image Computing and Computer-Assisted Intervention -- MICCAI 2017*, 2017, pp. 728–736, doi: 10.1007/978-3-319-66182-7_83.

[74] E. Gibson *et al.*, "NiftyNet: a deep-learning platform for medical imaging," *Comput. Methods Programs Biomed.*, vol. 158, pp. 113–122, 2018, doi: 10.1016/j.cmpb.2018.01.025.

[75] M. Chung, J. Lee, M. Lee, J. Lee, and Y. G. Shin, "Deeply self-supervised contour embedded neural network applied to liver segmentation," *Comput. Methods Programs Biomed.*, vol. 192, p. 105447, 2020, doi:

10.1016/j.cmpb.2020.105447.

[76] S. Liu *et al.*, “3D Anisotropic Hybrid Network: Transferring Convolutional Features from 2D Images to 3D Anisotropic Volumes,” in *Medical Image Computing and Computer Assisted Intervention -- MICCAI 2018,* 2018, pp. 851–858, doi: 10.1007/978-3-030-00934-2_94.

[77] A. A. Novikov, D. Major, M. Wimmer, D. Lenis, and K. Buhler, “Deep Sequential Segmentation of Organs in Volumetric Medical Scans,” *IEEE Trans. Med. Imaging*, vol. 38, no. 5, pp. 1207–1215, 2019, doi: 10.1109/TMI.2018.2881678.

[78] A. M. Rickmann, A. G. Roy, I. Sarasua, N. Navab, and C. Wachinger, “‘Project & Excite’ Modules for Segmentation of Volumetric Medical Scans,” in *Medical Image Computing and Computer Assisted Intervention -- MICCAI 2019*, 2019, pp. 39–47, doi: 10.1007/978-3-030-32245-8_5.

[79] A. Qayyum, A. Lalande, and F. Meriaudeau, “Automatic segmentation of tumors and affected organs in the abdomen using a 3D hybrid model for computed tomography imaging,” *Comput. Biol. Med.*, vol. 127, no. October, p. 104097, 2020, doi: 10.1016/j.compbiomed.2020.104097.

[80] H. Jiang, T. Shi, Z. Bai, and L. Huang, “AHCNet: An Application of Attention Mechanism and Hybrid Connection for Liver Tumor Segmentation in CT Volumes,” *IEEE Access*, vol. 7, pp. 24898–24909, 2019, doi: 10.1109/ACCESS.2019.2899608.

[81] T. Takenaga *et al.*, “Four-dimensional fully convolutional residual network-based liver segmentation in Gd-EOB-DTPA-enhanced MRI,” *Int. J. Comput. Assist. Radiol. Surg.*, vol. 14, no. 8, pp. 1259–1266, 2019, doi: 10.1007/s11548-019-01935-z.

[82] 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*, Oct. 2016, vol. 9901 LNCS, pp. 424–432, doi: 10.1007/978-3-319-46723-8_49.

[83] Z. Bai, H. Jiang, S. Li, and Y. D. Yao, “Liver Tumor Segmentation Based on Multi-Scale Candidate Generation and Fractal Residual Network,” *IEEE Access*, vol. 7, pp. 82122–82133, 2019, doi: 10.1109/ACCESS.2019.2923218.

[84] H. Kakeya, T. Okada, and Y. Oshiro, “3D U-JAPA-Net: Mixture of Convolutional Networks for Abdominal Multi-organ CT Segmentation,” in *Medical Image Computing and Computer Assisted Intervention -- MICCAI 2018*, 2018, pp. 426–433, doi: 10.1007/978-3-030-00937-3_49.

[85] C. Huang, H. Han, Q. Yao, S. Zhu, and K. S. Zhou, “3D U2-Net: A 3D Universal U-Net for Multi-domain Medical Image Segmentation,” in *Medical Image Computing and Computer Assisted Intervention -- MICCAI 2019*, 2019, pp. 291–299, doi: 10.1007/978-3-030-32245-8_33.

[86] F. Zhang *et al.*, “Liver Tissue Classification Using an Auto-context-based Deep Neural Network with a Multi-phase Training Framework,” in *International Workshop on Patch-based Techniques in Medical Imaging*, 2018, pp. 59–66, doi: 10.1007/978-3-030-00500-9_7.

[87] S. Mohagheghi and A. H. Foruzan, “Incorporating prior shape knowledge via data-driven loss model to improve 3D liver segmentation in deep CNNs,” *Int. J. Comput. Assist. Radiol. Surg.*, vol. 15, no. 2, pp. 249–257, 2020, doi: 10.1007/s11548-019-02085-y.

[88] H. R. Roth *et al.*, “A Multi-scale Pyramid of 3D Fully Convolutional Networks

for Abdominal Multi-organ Segmentation," in *Medical Image Computing and Computer Assisted Intervention -- MICCAI 2018*, 2018, pp. 417–425, doi: 10.1007/978-3-030-00937-3_48.

[89] H. R. Roth *et al.*, "An application of cascaded 3D fully convolutional networks for medical image segmentation," *Comput. Med. Imaging Graph.*, vol. 66, no. October 2017, pp. 90–99, 2018, doi: 10.1016/j.compmedimag.2018.03.001.

[90] Q. Zeng *et al.*, "Liver Segmentation in Magnetic Resonance Imaging via Mean Shape Fitting with Fully Convolutional Neural Networks," in *Medical Image Computing and Computer Assisted Intervention -- MICCAI 2019*, 2019, pp. 246–254, doi: 10.1007/978-3-030-32245-8_28.

[91] J. Yao, J. Cai, D. Yang, D. Xu, and J. Huang, "Integrating 3D Geometry of Organ for Improving Medical Image Segmentation," in *Medical Image Computing and Computer Assisted Intervention -- MICCAI 2019*, 2019, pp. 318–326, doi: 10.1007/978-3-030-32254-0_36.

[92] Q. Jin, Z. Meng, C. Sun, H. Cui, and R. Su, "RA-UNet: A Hybrid Deep Attention-Aware Network to Extract Liver and Tumor in CT Scans," *Front. Bioeng. Biotechnol.*, vol. 8, p. 1471, Dec. 2020, doi: 10.3389/fbioe.2020.605132.

[93] 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 Trans. Med. Imaging*, vol. 37, no. 12, pp. 2663–2674, 2018, doi: 10.1109/TMI.2018.2845918.

[94] Z. Zhou, M. M. R. Siddiquee, N. Tajbakhsh, and J. Liang, "UNet++: Redesigning Skip Connections to Exploit Multiscale Features in Image Segmentation," *IEEE Trans. Med. Imaging*, vol. 39, no. 6, pp. 1856–1867,

2020, doi: 10.1109/TMI.2019.2959609.

[95] P. Xu, C. Chen, X. Wang, W. Li, and J. Sun, “ROI-Based Intraoperative MR-CT Registration for Image-Guided Multimode Tumor ablation Therapy in Hepatic Malignant Tumors,” *IEEE Access*, vol. 8, pp. 13613–13619, 2020, doi: 10.1109/aCCESS.2020.2966518.

[96] C. Li *et al.*, “ANU-Net: Attention-based nested U-Net to exploit full resolution features for medical image segmentation,” *Comput. Graph.*, vol. 90, pp. 11–20, 2020, doi: 10.1016/j.cag.2020.05.003.

[97] Y. Zhou *et al.*, “Prior-aware Neural Network for Partially-Supervised Multi-Organ Segmentation,” in *IEEE International Conference on Computer Vision*, 2019, pp. 10671–10680, doi: 10.1109/ICCV.2019.01077.

[98] Z. Zhou *et al.*, “Models Genesis: Generic Autodidactic Models for 3D Medical Image Analysis,” in *Medical Image Computing and Computer Assisted Intervention -- MICCAI 2019*, 2019, pp. 384–393, doi: 10.1007/978-3-030-32251-9_42.

[99] A. Das and S. K. Sabut, “Kernelized Fuzzy C-means Clustering with Adaptive Thresholding for Segmenting Liver Tumors,” *Procedia Comput. Sci.*, vol. 92, pp. 389–395, 2016, doi: 10.1016/j.procs.2016.07.395.

[100] A. Das, P. Das, S. S. Panda, and S. Sabut, “Detection of Liver Cancer Using Modified Fuzzy Clustering and Decision Tree Classifier in CT Images,” *Pattern Recognit. Image Anal.*, vol. 29, no. 2, pp. 201–211, 2019, doi: 10.1134/S1054661819020056.

[101] A. M. Anter and A. E. Hassenian, “CT liver tumor segmentation hybrid approach using neutrosophic sets, fast fuzzy c-means and adaptive watershed algorithm,” *Artif. Intell. Med.*, vol. 97, pp. 105–117, 2019, doi:

10.1016/j.artmed.2018.11.007.

[102] W. Li, F. Jia, and Q. Hu, “Automatic Segmentation of Liver Tumor in CT Images with Deep Convolutional Neural Networks,” *J. Comput. Commun.*, vol. 03, no. 11, pp. 146–151, 2015, doi: 10.4236/jcc.2015.311023.

[103] C. Sun *et al.*, “Automatic segmentation of liver tumors from multiphase contrast-enhanced CT images based on FCNs,” *Artif. Intell. Med.*, vol. 83, pp. 58–66, 2017, doi: 10.1016/j.artmed.2017.03.008.

[104] G. Chlebus, A. Schenk, J. H. Moltz, B. van Ginneken, H. K. Hahn, and H. Meine, “Automatic liver tumor segmentation in CT with fully convolutional neural networks and object-based postprocessing,” *Sci. Rep.*, vol. 8, no. 1, pp. 1–7, 2018, doi: 10.1038/s41598-018-33860-7.

[105] J. Cai *et al.*, “Accurate Weakly-Supervised Deep Lesion Segmentation Using Large-Scale Clinical Annotations: Slice-Propagated 3D Mask Generation from 2D RECIST,” in *Medical Image Computing and Computer-Assisted Intervention -- MICCAI 2018*, 2018, pp. 396–404, doi: 10.1007/978-3-030-00937-3.

[106] S. Xie and Z. Tu, “Holistically-Nested Edge Detection,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1395–1403.

[107] C. Cano-Espinosa, G. Gonzalez, G. R. Washko, M. Cazorla, and R. S. J. Estepar, “Biomarker Localization from Deep Learning Regression Networks,” *IEEE Trans. Med. Imaging*, vol. 39, no. 6, pp. 2121–2132, 2020, doi: 10.1109/TMI.2020.2965486.

[108] Q. Jin, Z. Meng, C. Sun, L. Wei, and R. Su, “RA-UNet: A hybrid deep attention-aware network to extract liver and tumor in CT scans,” 2018, [Online]. Available: http://arxiv.org/abs/1811.01328.

[109] X. Xiao *et al.*, “Radiomics-guided GAN for Segmentation of Liver Tumor Without Contrast Agents,” in *Medical Image Computing and Computer Assisted Intervention -- MICCAI 2019*, 2019, pp. 237–245, doi: 10.1007/978-3-030-32245-8_27.

[110] S. Pang *et al.*, “CTumorGAN: a unified framework for automatic computed tomography tumor segmentation,” *Eur. J. Nucl. Med. Mol. Imaging*, vol. 47, no. 10, pp. 2248–2268, 2020, doi: 10.1007/s00259-020-04781-3.

[111] R. Zhang, Z. Zhou, W. Wu, C. C. Lin, P. H. Tsui, and S. Wu, “An improved fuzzy connectedness method for automatic three-dimensional liver vessel segmentation in CT images,” *J. Healthc. Eng.*, vol. 2018, pp. 1–17, 2018, doi: 10.1155/2018/2376317.

[112] Y. Z. Zeng, Y. Q. Zhao, M. Liao, B. J. Zou, X. F. Wang, and W. Wang, “Liver vessel segmentation based on extreme learning machine,” *Phys. Medica*, vol. 32, no. 5, pp. 709–716, 2016, doi: 10.1016/j.ejmp.2016.04.003.

[113] D. Mishra, S. Chaudhury, M. Sarkar, and A. S. Soin, “Ultrasound Image Segmentation: A Deeply Supervised Network With Attention to Boundaries,” *IEEE Trans. Biomed. Eng.*, vol. 66, no. 6, pp. 1637–1648, 2019, doi: 10.1109/TBME.2018.2877577.

[114] T. Kitrungrotsakul *et al.*, “VesselNet: A deep convolutional neural network with multi pathways for robust hepatic vessel segmentation,” *Comput. Med. Imaging Graph.*, vol. 75, pp. 74–83, 2019, doi: 10.1016/j.compmedimag.2019.05.002.

[115] Q. Huang, J. Sun, H. Ding, X. Wang, and G. Wang, “Robust liver vessel extraction using 3D U-Net with variant dice loss function,” *Comput. Biol. Med.*, vol. 101, no. March, pp. 153–162, 2018, doi:

10.1016/j.compbiomed.2018.08.018.

[116] D. Zhang *et al.*, “Graph Attention Network based Pruning for Reconstructing 3D Liver Vessel Morphology from Contrasted CT Images,” pp. 1–10, 2020, [Online]. Available: http://arxiv.org/abs/2003.07999.

[117] H. R. Roth *et al.*, “A new 2.5D representation for lymph node detection using random sets of deep convolutional neural network observations,” in *Medical Image Computing and Computer Assisted Intervention -- MICCAI 2014*, 2014, vol. 8673 LNCS, no. PART 1, pp. 520–527, doi: 10.1007/978-3-319-10404-1_65.

[118] G. H. Chen, D. Shah, and P. Golland, “A Latent Source Model for Patch-Based Image Segmentation,” in *Medical Image Computing and Computer-Assisted Intervention -- MICCAI 2015*, 2015, pp. 140–148, doi: 10.1007/978-3-319-24574-4_17.

[119] Y. Wang, Y. Zhou, W. Shen, S. Park, E. K. Fishman, and A. L. Yuille, “Abdominal multi-organ segmentation with organ-attention networks and statistical fusion,” *Med. Image Anal.*, vol. 55, pp. 88–102, 2019, doi: 10.1016/j.media.2019.04.005.

[120] T. Heimann *et al.*, “Comparison and Evaluation of Methods for Liver Segmentation from CT Datasets,” *IEEE Trans. Med. Imaging*, vol. 28, no. 8, pp. 1251–1265, 2009, doi: 10.1109/TMI.2009.2013851.

[121] Kitware Inc., “MIDAS - The Digital Archiving System.” https://www.insight-journal.org/midas/ (accessed Sep. 19, 2020).

[122] B. Landman, Z. Xu, J. E. Igelsias, M. Styner, T. R. Langerak, and A. Klein, “2015 MICCAI Multi-Atlas Labeling Beyond the Cranial Vault - Workshop and Challenge,” 2015. .

[123] H. R. Roth, A. Farag, E. B. Turkbey, L. Lu, J. Liu, and R. M. Summers, "Data From Pancreas-CT." The Cancer Imaging Archive (TCIA), 2016, doi: 10.7937/K9/TCIA.2016.tNB1kqBU.

[124] "Visceral - Anatomy3 continuous evaluation." http://www.visceral.eu/benchmarks/anatomy3-open/ (accessed Sep. 16, 2020).

[125] O. Jimenez-Del-Toro *et al.*, "Cloud-Based Evaluation of Anatomical Structure Segmentation and Landmark Detection Algorithms: VISCERAL Anatomy Benchmarks," *IEEE Trans. Med. Imaging*, vol. 35, no. 11, pp. 2459–2475, Nov. 2016, doi: 10.1109/TMI.2016.2578680.

[126] P. Christ, "CodaLab - LiTS - Liver tumor Segmentation Challenge," *CodaLab*, 2017. https://competitions.codalab.org/competitions/17094#learn_the_details-overview (accessed Sep. 16, 2020).

[127] P. Bilic *et al.*, "The Liver Tumor Segmentation Benchmark (LiTS)," no. January, 2019, [Online]. Available: http://arxiv.org/abs/1901.04056.

[128] A. L. Simpson *et al.*, "A large annotated medical image dataset for the development and evaluation of segmentation algorithms," *arXiv*, vol. 12, no. 9, Feb. 2019, Accessed: Jun. 12, 2021. [Online]. Available: http://arxiv.org/abs/1902.09063.

[129] K. Clark *et al.*, "The Cancer Imaging Archive (TCIA): Maintaining and Operating a Public Information Repository," *J. Digit. Imaging*, pp. 1045–1057, 2013, doi: 10.1007/s10278-013-9622-7.

[130] B. Rister, K. Shivakumar, T. Nobashi, and D. L. Rubin, "CT-ORG: CT volumes with multiple organ segmentations [Dataset]." The Cancer Imaging Archive (TCIA), 2019, doi: 10.7937/TCIA.2019.TT7F4V7O.

[131] A. E. Kavur *et al.*, "CHAOS Challenge - combined (CT-MR) healthy

abdominal organ segmentation," *Med. Image Anal.*, vol. 69, p. 101950, Apr. 2021, doi: 10.1016/j.media.2020.101950.

[132] M. A. Selver *et al.*, "CHAOS - Grand Challenge." https://chaos.grand-challenge.org/Combined_Healthy_Abdominal_Organ_Segmentation (accessed Sep. 15, 2020).

[133] T. Heimann, B. van Ginneken, M. A. Styner, and G. Humpire, "SLIVER07 - Grand Challenge." https://sliver07.grand-challenge.org/Home/ (accessed Sep. 15, 2020).

[134] IRCAD, "3Dircadb | IRCAD France." https://www.ircad.fr/research/3dircadb/ (accessed Sep. 13, 2020).

[135] Kitware Inc., "MIDAS - Collection Livers and liver tumors with expert hand segmentations." https://www.insight-journal.org/midas/collection/view/38 (accessed Sep. 19, 2020).

[136] M. J. Cardoso *et al.*, "Medical Segmentation Decathlon Challenge." http://medicaldecathlon.com/index.html (accessed Sep. 19, 2020).

[137] J. Delua, "Supervised vs. Unsupervised Learning: What's the Difference? | IBM," *IBM*, 2021. https://www.ibm.com/cloud/blog/supervised-vs-unsupervised-learning (accessed Nov. 29, 2021).

[138] "System Overview -- RAAD2 - Research Computing @ TAMUQ," 2017. https://rc-docs.qatar.tamu.edu/index.php/Raad2_intro (accessed Oct. 10, 2021).

[139] PyTorch, "ReduceLROnPlateau — PyTorch 1.9.0 documentation." https://pytorch.org/docs/stable/generated/torch.optim.lr_scheduler.ReduceLROnPlateau.html#torch.optim.lr_scheduler.ReduceLROnPlateau (accessed Jun. 26, 2021).

[140] PyTorch, "OneCycleLR — PyTorch 1.9.0 documentation."

https://pytorch.org/docs/stable/generated/torch.optim.lr_scheduler.OneCycleLR.html#torch.optim.lr_scheduler.OneCycleLR (accessed Jun. 26, 2021).

[141] F. Isensee, P. Kickingereder, W. Wick, M. Bendszus, and K. H. Maier-Hein, "No New-Net," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics),* vol. 11384 LNCS, pp. 234–244, Sep. 2018, doi: 10.1007/978-3-030-11726-9_21.

[142] S. van der Walt *et al.*, "scikit-image: image processing in Python," *PeerJ*, vol. 2, p. e453, 2014, doi: 10.7717/peerj.453.

[143] D. Chakravorty, "OBJ File Format – Simply Explained | All3DP." https://all3dp.com/1/obj-file-format-3d-printing-cad/ (accessed Sep. 18, 2021).

[144] "OBJ FILE FORMAT." https://www.cs.cmu.edu/~mbz/personal/graphics/obj.html (accessed Oct. 03, 2021).

[145] L. N. Smith and N. Topin, "Super-convergence: very fast training of neural networks using large learning rates," in *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications*, May 2019, vol. 11006, doi: 10.1117/12.2520589.

[146] "Results." http://medicaldecathlon.com/results/ (accessed Oct. 14, 2021).

[147] "PyQt5 · PyPI." https://pypi.org/project/PyQt5/ (accessed Sep. 18, 2021).

[148] R. Computing, "Riverbank Computing | Introduction." https://www.riverbankcomputing.com/software/pyqt/ (accessed Sep. 18, 2021).

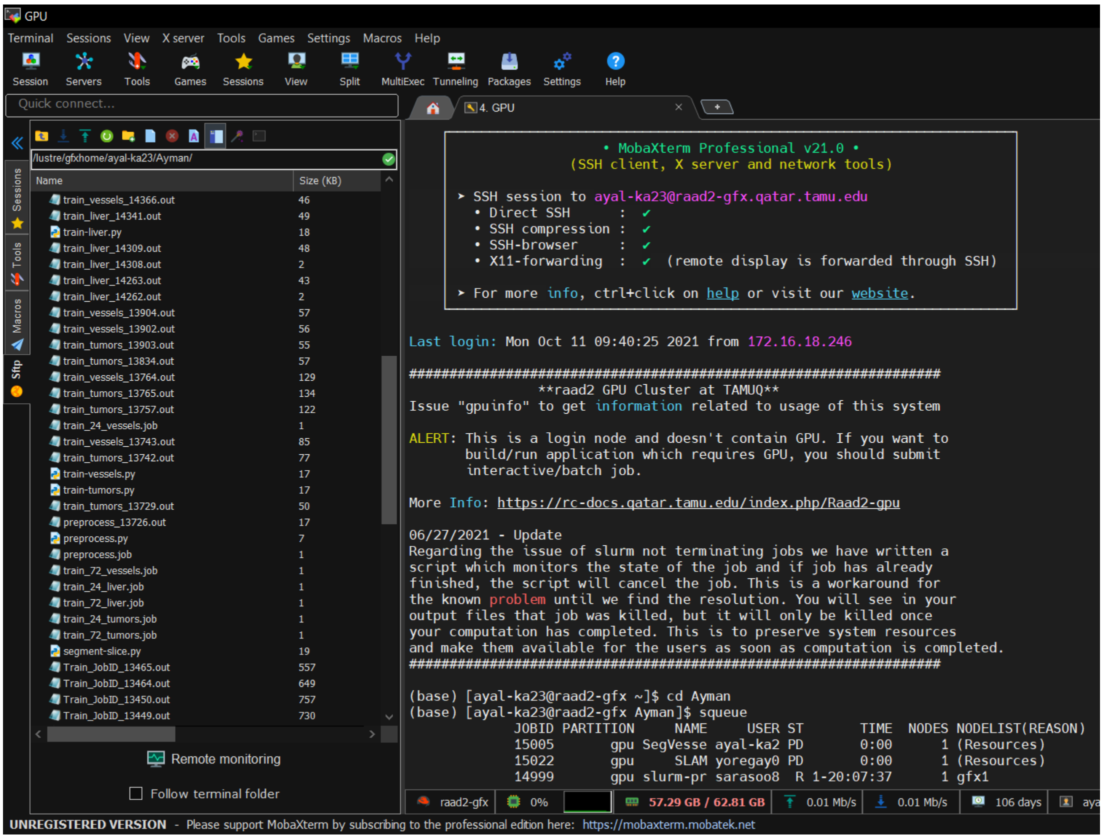
[149] "MobaXterm free Xserver and tabbed SSH client for Windows." https://mobaxterm.mobatek.net/ (accessed Oct. 10, 2021).

[150] P. A. Yushkevich *et al.*, "User-guided 3D active contour segmentation of anatomical structures: Significantly improved efficiency and reliability,"

Neuroimage, vol. 31, no. 3, pp. 1116–1128, Jul. 2006, doi: 10.1016/j.neuroimage.2006.01.015.

APPENDIX A: TAMUQ'S RAAD2 HPC USAGE

In order to obtain access to this HPC, a form must be submitted, where it deems the requesters as eligible if they are affiliated to TAMUQ or any academic institution inside Qatar. Once the application has been processed and accepted, end-users can log into the system and issue computing requests remotely via a Windows 10 toolbox named MobaXterm. The toolbox allows easy access to different remote network tools such as secure shell (SSH) and secure file transfer protocol (SFTP), and from which jobs can be submitted and files can be transferred, respectively [149]. Figure 31 shows the initial graphical user interface (GUI) when logging into the specific node (CPU/GPU).


GPU
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
Quick connect...
4. GPU
Sessions
Tools
Macros
Sftp
Remote monitoring
□ Follow terminal folder

/lustre/gfxhome/ayal-ka23/Ayman/

Name	Size (KB)
train_vessels_14366.out	46
train_liver_14341.out	49
train-liver.py	18
train_liver_14309.out	48
train_liver_14308.out	2
train_liver_14263.out	43
train_liver_14262.out	2
train_vessels_13904.out	57
train_vessels_13902.out	56
train_tumors_13903.out	55
train_tumors_13834.out	57
train_vessels_13764.out	129
train_tumors_13765.out	134
train_tumors_13757.out	122
train_24_vessels.job	1
train_vessels_13743.out	85
train_tumors_13742.out	77
train-vessels.py	17
train-tumors.py	17
train_tumors_13729.out	50
preprocess_13726.out	17
preprocess.py	7
preprocess.job	1
train_72_vessels.job	1
train_24_liver.job	1
train_72_liver.job	1
train_24_tumors.job	1
train_72_tumors.job	1
segment-slice.py	19
Train_JobID_13465.out	557
Train_JobID_13464.out	649
Train_JobID_13450.out	757
Train_JobID_13449.out	730

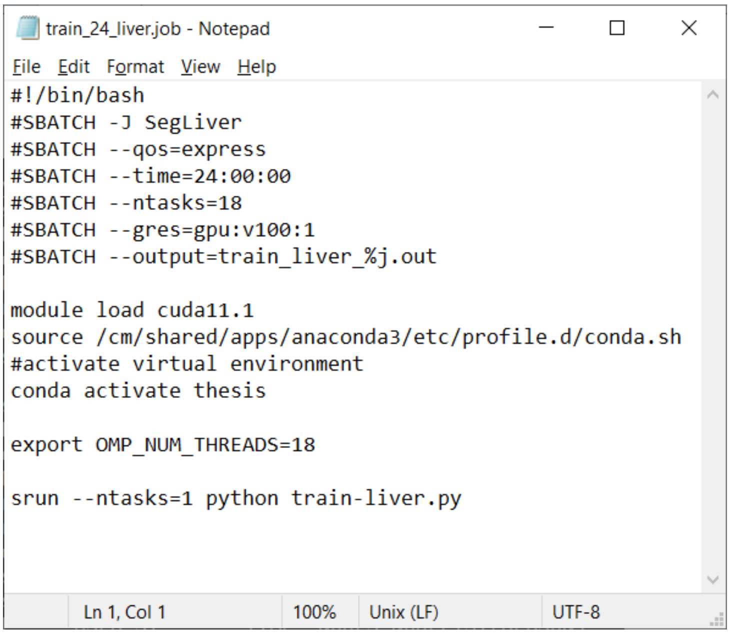
```
                    • MobaXterm Professional v21.0 •
                (SSH client, X server and network tools)

 ➤ SSH session to ayal-ka23@raad2-gfx.qatar.tamu.edu
   • Direct SSH      :  ✓
   • SSH compression :  ✓
   • SSH-browser     :  ✓
   • X11-forwarding  :  ✓  (remote display is forwarded through SSH)

 ➤ For more info, ctrl+click on help or visit our website.

Last login: Mon Oct 11 09:40:25 2021 from 172.16.18.246

###################################################################
                **raad2 GPU Cluster at TAMUQ**
Issue "gpuinfo" to get information related to usage of this system

ALERT: This is a login node and doesn't contain GPU. If you want to
       build/run application which requires GPU, you should submit
       interactive/batch job.

More Info: https://rc-docs.qatar.tamu.edu/index.php/Raad2-gpu

06/27/2021 - Update
Regarding the issue of slurm not terminating jobs we have written a
script which monitors the state of the job and if job has already
finished, the script will cancel the job. This is a workaround for
the known problem until we find the resolution. You will see in your
output files that job was killed, but it will only be killed once
your computation has completed. This is to preserve system resources
and make them available for the users as soon as computation is completed.
###################################################################

(base) [ayal-ka23@raad2-gfx ~]$ cd Ayman
(base) [ayal-ka23@raad2-gfx Ayman]$ squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
             15005       gpu SegVesse ayal-ka2 PD       0:00      1 (Resources)
             15022       gpu     SLAM yoregay0 PD       0:00      1 (Resources)
             14999       gpu slurm-pr sarasoo8  R 1-20:07:37      1 gfx1
```

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net

Figure 31. MobaXterm interface when connecting to Raad2.

However, to use it outside the campus of TAMUQ, it is necessary to access it through a virtual private network (VPN) connection to be linked to the TAMUQ Wi-Fi via “Cisco AnyConnect Secure Mobility Client” software. It is also equipped with a 2-factor authentication for safer use and access to the HPC.

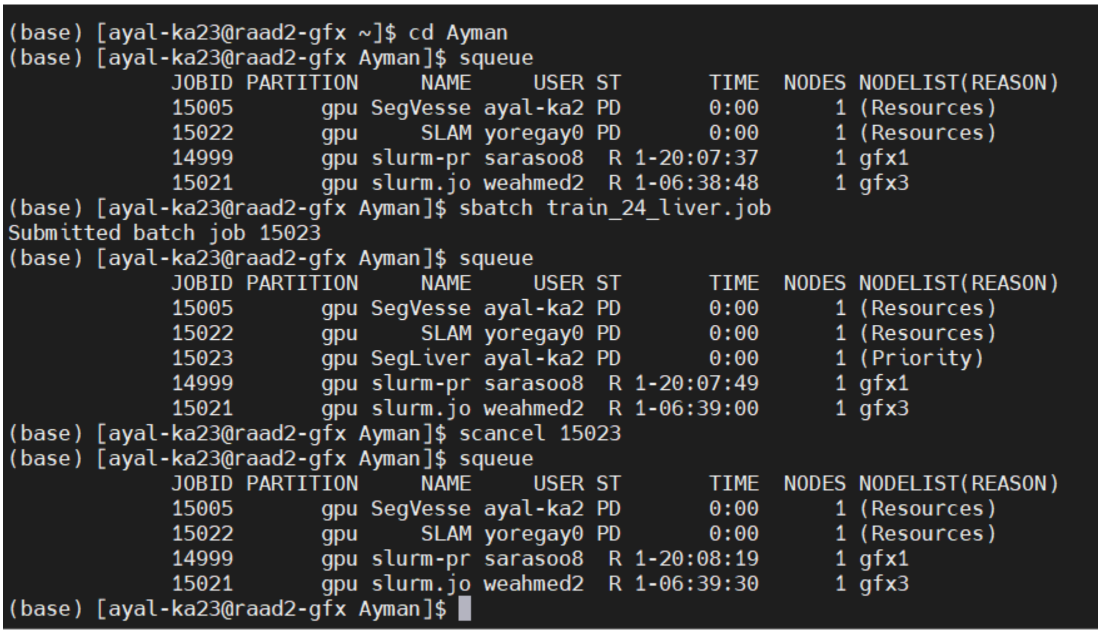
In order to submit a job for the HPC to process, a job file must be submitted to instruct the HPC detailing all the necessary information, as shown in Figure 32.



```
#!/bin/bash
#SBATCH -J SegLiver
#SBATCH --qos=express
#SBATCH --time=24:00:00
#SBATCH --ntasks=18
#SBATCH --gres=gpu:v100:1
#SBATCH --output=train_liver_%j.out

module load cuda11.1
source /cm/shared/apps/anaconda3/etc/profile.d/conda.sh
#activate virtual environment
conda activate thesis

export OMP_NUM_THREADS=18

srun --ntasks=1 python train-liver.py
```

Figure 32. Example of a job file used to execute Python code.

Within this file, parameters are set by specifying the needed time to do the process (24 hours), the GPU to run it on (gpu:v100:1), the name of the output file (train_liver_<job_number>.out), the CUDA version to be used (11.1), the virtual environment to be used (thesis), and the Python script to be run (train-liver.py). Lastly, Figure 33 illustrates how to check the jobs currently running/waiting in the queue, how to submit a new job, and how to cancel a submitted job.



```
(base) [ayal-ka23@raad2-gfx ~]$ cd Ayman
(base) [ayal-ka23@raad2-gfx Ayman]$ squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
             15005       gpu SegVesse ayal-ka2 PD       0:00      1 (Resources)
             15022       gpu     SLAM yoregay0 PD       0:00      1 (Resources)
             14999       gpu slurm-pr sarasoo8  R 1-20:07:37      1 gfx1
             15021       gpu slurm.jo weahmed2  R 1-06:38:48      1 gfx3
(base) [ayal-ka23@raad2-gfx Ayman]$ sbatch train_24_liver.job
Submitted batch job 15023
(base) [ayal-ka23@raad2-gfx Ayman]$ squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
             15005       gpu SegVesse ayal-ka2 PD       0:00      1 (Resources)
             15022       gpu     SLAM yoregay0 PD       0:00      1 (Resources)
             15023       gpu SegLiver ayal-ka2 PD       0:00      1 (Priority)
             14999       gpu slurm-pr sarasoo8  R 1-20:07:49      1 gfx1
             15021       gpu slurm.jo weahmed2  R 1-06:39:00      1 gfx3
(base) [ayal-ka23@raad2-gfx Ayman]$ scancel 15023
(base) [ayal-ka23@raad2-gfx Ayman]$ squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
             15005       gpu SegVesse ayal-ka2 PD       0:00      1 (Resources)
             15022       gpu     SLAM yoregay0 PD       0:00      1 (Resources)
             14999       gpu slurm-pr sarasoo8  R 1-20:08:19      1 gfx1
             15021       gpu slurm.jo weahmed2  R 1-06:39:30      1 gfx3
(base) [ayal-ka23@raad2-gfx Ayman]$ 
```

Figure 33. Submitting a job, checking current jobs, and canceling a job.