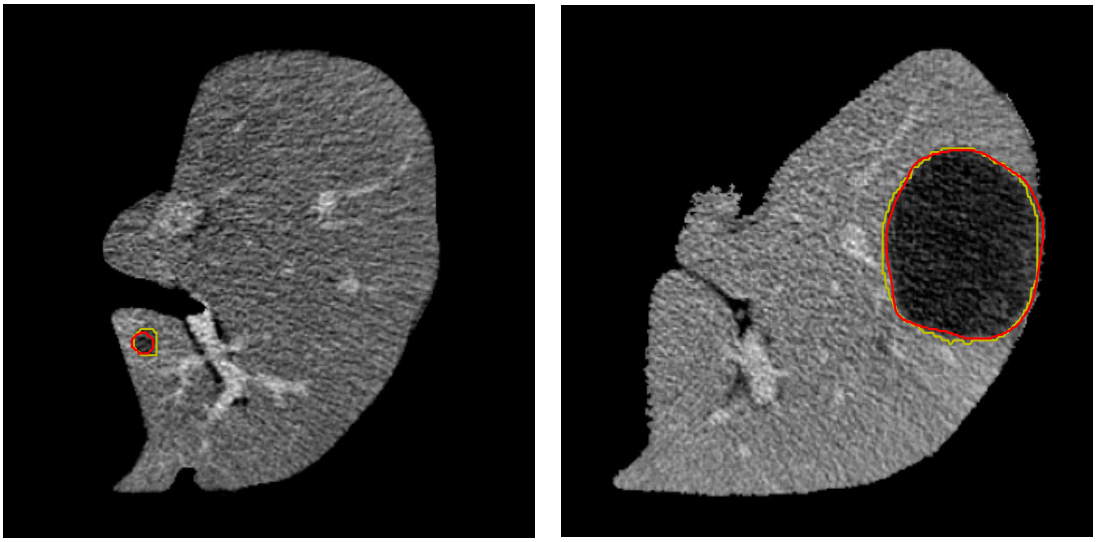

Liver Tumor Segmentation Using Classical Algorithms & Deep Learning

Master's thesis in Biomedical Engineering

SOFIE ALLGÖWER, SOFIA LJUNGDAHL

DEPARTMENT OF MATHEMATICAL SCIENCES

CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2023
www.chalmers.se

Master's thesis 2023

Liver Tumor Segmentation Using Classical Algorithms & Deep Learning

SOFIE ALLGÖWER, SOFIA LJUNGDAHL

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Mathematical Sciences
Chalmers University of Technology
Gothenburg, Sweden 2023

Liver Tumor Segmentation Using Classical Algorithms & Deep Learning
SOFIE ALLGÖWER, SOFIA LJUNGDAHL

Supervisor: Carl Bodin, Navari Surgical
Examiner: Klas Modin, Mathematical Sciences

Master's Thesis 2023
Department of Mathematical Sciences
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Tumor segmentation of two different scans predicted by a U-Net model trained on cropped images with two-channel dice loss. The yellow lines are the predicted segmentation and the red lines are the ground truth. For more results, see Chapter 4.

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2023

Liver Tumor Segmentation Using Classical Algorithms & Deep Learning
SOFIE ALLGÖWER, SOFIA LJUNGDAHL
Department of Mathematical Sciences
Chalmers University of Technology

Abstract

Liver cancer is a common condition that traditionally required open surgery, posing a high risk of complications. Laparoscopic surgery has become increasingly popular, but comes with navigation challenges. The MedTech start-up *Navari Surgical* has developed a visualization solution using augmented reality, and this project aims to suggest a tumor segmentation method to support this solution.

Previous studies have inspired this work to explore tumor segmentation utilizing different approaches, such as thresholding algorithms, active contour models, and a deep learning model utilizing the U-Net architecture. Thresholding methods uses pixel intensities, active contour models focuses on minimizing image energy, and U-Net models learn image features through training. For the U-Net models, variations in the learning rate, augmented data quantity, and loss functions are explored. The study utilizes the open-source LiTS dataset. The methods employ either liver-segmented or cropped tumor images as inputs. Evaluation metrics include dice's similarity coefficient (DSC) and recall, with a dataset of 107 images for evaluation of the classical algorithms, and 696 test images for the U-Net models.

The obtained results demonstrate that thresholding algorithms with cropped input yield the highest DSC and recall values for the classical algorithms. The best performance was observed with cropped Multi Otsu (DSC: 0.435, recall: 0.605). For the U-Net models, increased augmented data, reduced learning rate, and more epochs resulted in improved performance. The best U-Net model achieved a DSC of 0.766 and a recall of 0.796.

The discussion highlights challenges with algorithms designed for single tumor detection when evaluating datasets containing multiple tumors per image. Classical algorithms show a need for individualization for each scan, impacting automation and efficiency. Overfitting is a concern for the U-Net models, suggesting room for improvement. Further enhancements include pre and post-processing techniques, parameter variation, exploration of modified architectures, and utilization of 3D input data.

In conclusion, U-Net demonstrated the best performance among the methods explored. However, its performance is not yet suitable for practical use, requiring further improvements. The recommendation for *Navari* is to continue to explore deep learning and U-Net for future advancements in tumor segmentation.

Keywords: liver tumor, augmented reality, image segmentation, LiTS dataset, thresholding, active contour models, U-Net, dice similarity coefficient, recall value.

Acknowledgements

We would like to express our deepest gratitude to our supervisor, Carl Bodin, for his unwavering support, invaluable guidance, and constant encouragement throughout this project. His expertise, dedication, and mentorship have been instrumental in shaping our research and academic growth. We would also like to thank our examiner, Klas Modin, for his valuable insights and constructive feedback that greatly contributed to the refinement of our work. Additionally, we extend our heartfelt appreciation to Madeleine Gustavsson and Axel Blomé, for their camaraderie, positivity, and continuous support. Their presence made our time in the office truly enjoyable and inspiring. Lastly, we would like to thank Ellen Arnholm and Lisa Månsson for their kind guidance. Their belief in us and willingness to lend a helping hand were deeply appreciated.

Sofie Allgöwer & Sofia Ljungdahl, Gothenburg, June 2023

List of Acronyms

Below is the list of acronyms that have been used throughout this thesis listed in alphabetical order:

Adam	Adaptive Moment Estimation
CBCT	Cone Beam Computed Tomography
CNN	Convolutional Neural Network
CT	Computed Tomography
DICOM	Digital Imaging and Communications in Medicine
DSC	Dice's Similarity Coefficient (Dice Score)
HCC	Hepatocellular Carcinoma
HU	Hounsfield Units
ICC	Intrahepatic Cholangiocarcinoma
LiTS	Liver Tumor Segmentation
MRI	Magnetic Resonance Imaging
NIfTI	Neuroimaging Informatics Technology Initiative
ReLU	Rectified Linear Unit
SGD	Stochastic gradient descent

Contents

List of Figures

List of Tables

1
Introduction

In this chapter, the background for this project is presented. From the background, a clear gap is understood, from which an aim with limitations is formulated. Before starting the project, some prior work is investigated for inspiration and gathering of knowledge.

1.1 Background

The standard treatment of liver cancer in acute cases is surgical resection [4]. Surgical resection of liver tumors is performed by either open surgery, where the abdomen is cut open, or laparoscopic surgery, a minimally invasive procedure using only small incisions. Open surgery is the traditional alternative but comes with a high risk of complications after surgery and a long hospital stay. Laparoscopic surgery is getting more common as it decreases these risks [5]. However, a major challenge with laparoscopic surgery is the difficulty for the surgeon to navigate the liver tissue and locate the tumor.

The course of action to navigate during laparoscopic liver surgery today is that the surgeon switches between looking at the laparoscopic camera view and preoperational images and creates a mental image of where the tumor is located in the liver [6]. To make sure that the entire tumor is removed the surgeon removes it by a large enough margin. This approach increases the probability of total resection of the tumor but also increases the recovery time [5]. To support the surgeon in navigating and locating the tumor in the liver tissue, a solution using augmented reality (AR) could be used that displays the tumor on the laparoscopic camera screen [6]. This might also help the surgeon resect the tumor with a smaller margin, which shortens the recovery time. *Navari Surgical AB* is a MedTech start-up company that develops such an AR solution, and this Master's thesis is performed in conjunction with them.

A vital step to make the AR solution work in practice is to find the tumor in the intraoperative image, by segmenting the image. To ensure a good outcome of the surgery it is essential that the result of the segmentation is reliable. It is also important that the segmentation is fast since this is a new task added to the workflow of liver surgery and thus need to be introduced to the medical team. Therefore, the segmentation must be both fast and correct in order to be useful in practice.

In a previous Master's thesis performed by two master students together with

Navari, different algorithms to perform semantic segmentation were examined and discussed [7]. The results have potential but need to be confirmed on more data to be granted as trustworthy. Furthermore, solutions based on neural networks have been proposed by *Navari*. From this, the aim of this Master's thesis work was composed.

1.2 Aim

This Master's thesis aims to suggest a solution for semantic segmentation of liver tumors in computed tomography images suitable for *Navari Surgical*. This includes investigating and evaluating the performance of the algorithms from previous thesis work on a larger amount of data, as well as improving the algorithms by tuning parameters and making them more automated. It also includes developing a deep learning model to perform this task. For the solution to be suitable for the purpose, it needs to be automated and achieve a high performance for tumor segmentation. By evaluating all models, a conclusion about which approach is most likely to be a good fit for the company will be drawn.

1.3 Limitations

For this Master's thesis work, there are some limitations. Since the aim is to develop tumor segmentation methods, liver segmentation will not be further investigated. The algorithms examined will be limited to the ones proven interesting in a previous thesis work [7]. For the deep learning part, limitations will be related to the amount of data, the training time and the computer capacity. Furthermore, the neural networks will only be trained on 2D input. The data type available will also constitute a limitation. In practice, cone beam computed tomography images will be used during surgery, but during this thesis work, only computed tomography scans are available. Because of the similarities, the result will only be evaluated and discussed from the performance on the computed tomography images, and how the methods will work with cone beam computed tomography images will not be considered.

1.4 Prior Work

Semantic segmentation of medical images is not a new subject and there is already a lot of research executed in the field. This includes both classical algorithm solutions and more automated methods such as deep learning approaches. This thesis work will start from a previous master thesis report concluded in 2022, investigating classical algorithms such as thresholding methods and active contour models [7].

In the previous work, the methods of global thresholding, Multi Otsu thresholding, k-means clustering, and multiple active contour models were explored and evaluated on a limited dataset of only 97 scans from four different patients. Throughout their work, it was concluded that k-means clustering and basic contour models did not

yield any acceptable results. It was also concluded that of the remainder of the methods, only the method called Morphological Snake with thresholding explained in Subsection 2.4.2 resulted in less than half of the images being insufficient. This can be observed in the report's result section and Figure 4.29 in the report [7]. The result was evaluated on the 97 image slices from four different volumes. However, the evaluation method is not transparent since it is not evident what measures as insufficient, acceptable, or sufficient results. This thesis work will further develop the segmentation using the same methods, but since the evaluation in prior work is unclear the comparison with the current work will be affected. Another approach that will be investigated in this work is the effect of using cropped images. This since it is reasonable to expect the surgeon to be able to mark a region of interest around the tumor in a computed tomography scan. A smaller input image could affect the segmentation results in a positive way.

Besides the classical algorithm methods, the alternative using deep learning will also be investigated in this thesis work. Concerning this, the work will start from the results of a competition called Liver Tumor Segmentation Benchmark (LiTS) [3]. The competition has been organized on three different occasions. The first one was held in conjunction with the IEEE International Symposium on Biomedical Imaging (ISBI) in 2017, and then again in 2017 and 2018, with the International Conferences on Medical Image Computing and Computer-Assisted Intervention (MICCAI). A larger dataset, which will be explained in more detail in Subsection 2.8.2, was developed to be used in the competition. Before the LiTS benchmark, deep learning was rarely used for liver tumor segmentation, and more traditional machine learning methods were more common [3]. In 2016, Christ et al. were the first to use a 3D U-Net for liver and liver tumor segmentation using a cascaded segmentation strategy [8]. The diverse LiTS dataset opened up new possibilities, and many new deep-learning solutions were proposed in conjunction with the benchmark.

The contributions to the LiTS benchmark were most often modified U-Net based architectures [3]. Another common feature was cascading networks to be able to perform liver and liver tumor segmentation separately. Both full images and patches were used for training, and networks taking in 2D and 3D input were common throughout the different competitions. While 3D input was more common for the last occasions, 2.5D input utilising multichannel 2D input was more common for the early contributions. Pre- and postprocessing were also common for most of the submissions. Standardization, normalisation, and geometric data augmentation were among the most used for preprocessing. For the postprocessing step overlaying the liver mask on the tumor segmentation to discard tumors outside the liver was a common approach. Popular optimizers such as Adam (Adaptive Moment Estimation) and SDG (Stochastic Gradient Descent) with momentum were utilized together with multiple loss functions.

The solution that got the best result from the last competition was the nnU-Net created by Isensee et al. in 2018 [9]. The network is a self-configuring method for deep learning-based biomedical image segmentation with 3D input. The network

achieved a dice score of 0.739 and a recall value of 0.554. For a more detailed description of the nnU-Net and other modifications of the classical U-Net model, see Subsection 2.5.3.

2
Theory

In this chapter, the theory needed for the project is presented. First, some basic knowledge about the liver and liver cancer is conferred, followed by information about medical imaging and image segmentation. To be able to perform the tumor segmentation, four classical methods and corresponding theories are presented. Then, the theory behind the deep learning approach and further development are explored. Lastly, the evaluation metrics as well as the datasets investigated are introduced.

2.1 Medical Background

In this section, an overview of liver anatomy and physiology is given. Furthermore, the most common forms of liver cancer are introduced, as well as different treatment strategies for liver cancer.

2.1.1 Liver Anatomy and Physiology

The liver is the largest internal organ in the human body and performs a variety of vital and complex functions [10]. Some examples of important functions are the maintenance of blood sugar by glycogen storage, protein synthesis, detoxification and production of bile. It is located below the diaphragm and the rib cage in the upper right quadrant of the abdomen [10]. The weight of the organ in an adult human is about 1300-1700 g and it has a continuous sponge-like structure [10]. The liver contains only two cell types that are unique for the organ, hepatocytes and cholangiocytes [10]. Hepatocytes are the most common cells in the liver and are responsible for several important tasks such as glycogen storage, detoxification and production of bile [10]. The cholangiocytes on the other hand play an important role in the drainage of bile, as they form the biliary channels and modify the composition of the bile by secreting and absorbing different compounds [10]. In addition to these, the liver also contains several other specialized cells necessary for liver function and structure.

One special thing about the liver is its unique vascular structure. Blood is supplied to the liver in two ways, 75 % through the portal vein and 25 % through the hepatic artery [10]. The portal vein carries already deoxygenated blood rich in nutrients and possible toxins from the gastrointestinal tract to the liver [10]. This ensures that absorbed substances are processed in the liver before entering the systemic

circulation. The hepatic artery on the other hand carries oxygenated blood from the systemic circulation to the liver [10]. These two vessels branch out into smaller vessels called arterioles and venules that in turn empties into a capillary network made up of a special kind of capillaries called sinusoids [10]. The blood from the sinusoids is then collected into small veins and carried from the liver through the hepatic veins to the inferior vena cava, where it enters the systemic circulation. In total approximately 22 % of the liver mass/volume is made up of blood vessels [10]. The liver is constructed to take large blood volumes at high flow rates and at physiological conditions, about 12 % of the total blood volume is contained in the liver [10].

The liver can be divided into eight functional segments based on its vascular supply. Each segment is an individual unit that is supplied by its own branches in the vascular network as well as drained by its own biliary channels [10, 11]. This makes it possible to resect one or several neighbouring segments without damaging the remaining ones [10]. The liver can also regenerate lost tissue by the proliferation of hepatocytes, cholangiocytes and other liver cells [10]. This creates a tissue that is very similar to the unharmed liver. As much as 70 % of the liver can be regenerated. This regenerative capacity is particularly important after surgical resection of a portion of the liver, as the remaining segments grow in size to compensate for tissue loss.

2.1.2 Liver Cancer

Liver cancer can be divided into primary and secondary cancer. Primary liver cancer originates in the liver whereas secondary liver cancer originates in another part of the body and metastasizes to the liver. Primary liver cancer is the fifth most common cancer worldwide and is the second most common cause of cancer deaths [12]. Hepatocellular carcinoma (HCC) and intrahepatic cholangiocarcinoma (ICC) together account for more than 95 % of all primary liver cancers [12]. HCC is by far the most common one and alone accounts for approximately 80-85 % of all cases. HCC arises from the hepatocytes and often affects people that suffer from chronic liver diseases, such as hepatitis B and C [12]. ICC, on the other hand, arises from the bile ducts inside the liver [13].

There exist several treatment options for patients suffering from HCC and the treatment strategy should be individualized based on factors such as disease stage, degree of liver dysfunction and the general status and age of the patient [12, 14]. For patients without cirrhosis, a condition where the liver has been damaged due to liver disease, and with sufficiently good liver function resection of the tumor is the treatment of choice [14]. Thanks to the regenerative ability of the liver tissue it is possible also to resect large tumors. During resection, it is important that all of the tumors are removed and that the function of the remaining liver tissue is sufficient [12]. As a consequence, resection is not a suitable strategy for patients that suffer from severe cases of chronic liver disease and cirrhosis, as this increases the risk for postoperative complications [14]. If the patient suffers from chronic liver disease the functionality of the liver must be evaluated prior to a possible resection [12]. Resection is also

unsuitable for patients suffering from metastatic liver cancer [12].

For patients with advanced liver cirrhosis, the treatment of choice is liver transplantation, since this is the only treatment option that also improves or cures the chronic liver disease [12]. However, the shortage of donor organs and the complexity of the surgery complicates the use of this method. Another group of treatment methods is percutaneous ablation techniques, for example, radiofrequency ablation [14]. During treatment with radiofrequency ablation, a needle that generates an electrical current on its tip is inserted into the tumor and the heat generated causes the tumor cells to die. This is a possible treatment choice for patients that are not suitable for resection and in some cases, the results are comparable to surgical resection [14]. There also exist a number of other tumor ablation techniques, as well as other types of treatment options such as different kinds of drugs [14].

2.2 Medical Imaging

To visualize anatomical structures and physiological functions inside the body there are a variety of different image modalities to choose from. An imaging modality is a technique used to create a medical image. The different kinds are split into two categories of ionizing and non-ionizing modalities [15]. The first category contains Computed Tomography (CT) and nuclear medicine, while the second contains Ultrasound and Magnetic Resonance imaging (MRI). In nuclear medicine, tracers of radioactive compounds are injected into the body and the radioactive decay inside the body is then measured and registered to form an image. Ultrasound, which is commonly used to monitor a fetus during pregnancy, uses high-frequency waves and registers the reflection from different structures inside the body. MRI makes use of a strong magnetic field to generate images visualizing the anatomy and processes inside the body. Lastly, CT transmits X-rays through the body and measures the attenuation. CT is the imaging modality used for this work and will be further explained in more detail below.

2.2.1 Computed Tomography

In CT, X-rays are used as the source for the imaging [16]. From X-ray tubes, pulses shaped like cones are passed through the body. The intensity of these pulses is then measured by an X-ray detector, creating shadow-like features when passed through very dense tissue and attenuated inside the body. Unlike in projection radiography where overlapping tissues are visible, CT collects multiple projections from different angles of the same tissue. This is done by moving the X-ray tubes around the body being imaged. From these signals, it is possible to reconstruct multiple cross-sections of the body, forming the 3D volume. These cross-sections are called slices and are reconstructed to have better resolution and no overlap compared to projection radiography [16].

The attenuation of the X-rays depends on how radio dense the tissue being passed is. Different tissue types have different radiodensity which can be quantified by

the Hounsfield units (HU), after Sir Godfrey Hounsfield who helped develop the technique for CT imaging [17]. The HU value indicates the ability of the specific tissue to attenuate X-rays. It is beneficial to convert the CT images to HU for more than this reason. Different scanners have different effective energies from the X-ray tubes and do hence produce images with different pixel values [18]. To be able to compare one CT scan to another from a different scanner, they are converted into images expressed in HU according to

$$\mathrm{HU} = 1,000 \times \frac{\mu - \mu_{\mathrm{water}}}{\mu_{\mathrm{water}}} \tag{2.1}$$

where μ is the pixel value. From Equation (2.1) it is clear that the HU value for water is 0. Other interesting HU values for different tissues can be seen in Table 2.1.

Table 2.1: Differnt HU values for different tissue types [1, 2].

Substance	**HU**
Air	-1000
Fat	-90
Water	0
Liver	+60
Compact bone	+1000

2.2.2 Cone Beam CT

Cone Beam Computed Tomography (CBCT) is a quicker image modality option to normal CT, utilising rapid coverage and reconstructions of 3D volumes [18]. CBCT uses a cone beam radiation pattern together with an area detector, unlike regular CT which used a fan-beam and linear detector [18]. The X-ray source and area detector are placed on a C-arm able to rotate around the patient [19]. The scanner can be placed inside an operating room because of its slimmer design allowing for imaging to be performed during surgery without having to move the patient [20]. A CBCT scanner achieves results of the same type as a regular fan-beam CT, with better spatial resolution thanks to the flat detector, allowing for the same kind of reconstruction [18]. The rapid coverage achieved with just one rotation around the patient's body increases the risk of motion artifacts because of the slower rotation [19]. Other artifacts from noise, scattering, beam hardening, and more may also occur as a result of the rotating arm. Compared to CT, CBCT has a limited field of view that need to be considered to include the entire liver when imaging. For the purpose of *Navari Surgical*'s solution, CBCT will be used during surgery, but since CT images are more widely available in open-source, CT images will be used for this thesis work.

2.2.3 Contrast Imaging

To improve the look of a CT image to easier distinguish between different tissues, for example, liver and tumor, it is possible to enhance the contrast [21]. Contrast

is referred to as the intensity difference between pixels in an image, distinguishing between the object being imaged and the background. From Subsection 2.2.1, it is known that body tissues have different radiodensity, attenuating different amounts of X-rays. It is this difference in radiodensity that gives rise to contrast in the image [22]. From Table 2.1, it can be observed that the contrast is evident between air, water, and bone, but that is not always the case between different soft tissues. This complicates the imaging. To manage this problem, contrast agents can be introduced into the body. Contrast agents are chemical compounds that increase the absorption of X-rays, for the anatomical structure where it is introduced. By placing this agent cleverly, contrast can be enhanced and image segmentation could therefore be executed more easily.

2.2.4 Image Artifacts

When considering image quality, there are no escaping image artifacts. Artifacts in images do not represent any anatomical structures or functional objects inside the patient but represent only false information [23]. When it comes to CT, common artifacts are among others aliasing, beam hardening, X-ray scattering, and motion artifacts. The first three all depend on the scanner and X-ray source and are almost impossible to avoid and appear as streaks, rings, or stars in the CT images. Motion artifacts are due to the patient moving during imaging, which is almost inevitable since the heart will continue beating and the patient needs to breathe during a longer scanning time. Hence, image artifacts are hard to eliminate and need to be considered, since they can affect the segmentation.

2.2.5 DICOM and NIfTI

When working with medical data, it is important to be familiar with the different file formats. The LiTS dataset described in Subsection 2.8.2 is stored in a NIfTI (Neuroimaging Informatics Technology Initiative) format, which is most commonly used for neurological images. It is also common that a lot of image analysis tools require NIfTI data [24]. In the previous thesis work, the segmentation was performed on 2D images from DICOM (Digital Imaging and Communications in Medicine) files. The DICOM format is commonly used when storing or exporting data from scanners. DICOM is a complex standard that includes a lot of embedded data beyond the images. Example of this is scanned patient notes, audio files and tags such as size, the device used, imaging specifics, and image dimensions [24]. Unlike DICOM data, NIfTI files are very simple and minimalistic, where the data consists of just a header with information as well as uncompressed image data. For this project, the NIfTI files are converted into DICOM files.

2.2.6 Image Segmentation

To distinguish a specific object in an image, the object must be separated from the rest. In the case of liver tumor segmentation, it is desirable to separate the liver tumor from the liver tissues. This is what image segmentation is within the medical field: separating different tissues or anatomical structures from each other [25]. To

obtain a segmented image there are many available methods, both automatic and semi-automatic. Most methods involve extracting and classifying features, one being pixel classification which results in an image divided into regions. These regions are then labelled and the image is segmented into different regions of interest. Some of the available semi-automated techniques for segmentation include region-based methods such as thresholding, or edge-based segmentation methods such as active contour models [25]. For automatic segmentation, machine learning is the dominant method, where neural networks are trained to perform segmentation on the images. All these methods will be described in more detail in the following sections.

2.3 Thresholding Algorithms

Thresholding techniques have a central role in image segmentation because of their simplicity and computational speed. The result of the segmentation is solely dependent on the intensity of the pixels in the image [26].

2.3.1 Global Thresholding

If an image contains light objects on a dark background, such that the image histogram has two peaks with a clear valley between them, thresholding can be used to segment the image into foreground and background parts [26]. The threshold is a chosen intensity value used to classify each pixel as either foreground or background. A pixel is classified as foreground if its intensity exceeds the threshold, and as background if it is below the threshold. For a chosen threshold, k, the segmented image $g(x, y)$ is thus given by

$$g(x,y) = \begin{cases} \text{foreground} & \text{if } f(x,y) > k \\ \text{background} & \text{if } f(x,y) \leq k \end{cases} \tag{2.2}$$

where $f(x, y)$ is the intensity value at position (x, y) [25]. In global thresholding the chosen threshold k is constant over the whole image. Multiple thresholds can also be used [26].

Since the quality of the segmentation is dependent on the choice of suitable thresholds, it is ideal that the image histogram has peaks separated by sharp and deep valleys [27]. It is often hard to find the bottom of the valleys exactly, for example when the valleys are wide and flat and the peaks have different heights. This also makes thresholding sensitive to noise, that corrupts the image histogram [27]. The quality of the segmentation is also limited in cases where more than two thresholds are needed to successfully separate the objects in the image. Thresholding works best if one or two thresholds are chosen, and the image pixels accordingly are segmented into two or three classes [26].

The choice of suitable thresholds is a necessity for the method to yield satisfactory results. Choosing the threshold manually can be a challenge, as images often con-

tain multiple objects with similar intensities as well as intensity variations in the background. Thus, it is not certain that the image histogram is divided into clear peaks and valleys that represent different features in the image. To avoid choosing the threshold based on visual inspection, some automatic methods for threshold selection have been proposed, one such method was presented by Otsu in 1979 [27].

2.3.2 Otsu's Method

In Otsu's method, the goal is to automatically find the optimal threshold that best separates the different classes based on their pixel intensities [27]. In the method, it is assumed that the image has a bimodal histogram, a histogram with two peaks, but the method can also be extended to multi-class problems, see Subsection 2.3.3.

Let the pixel intensities in an image be represented by L gray levels in the range $[0, L-1]$. Also, let the number of pixels with intensity i be denoted as n_i and the total number of pixels in the image be N. The probability that a pixel has intensity i can then be computed as

$$p_i = \frac{n_i}{N} \tag{2.3}$$

and it holds that

$$\sum_{i=0}^{L-1} p_i = 1. \tag{2.4}$$

If the pixels are separated into two classes C_0 and C_1 based on their intensity value by a threshold k the probabilities of occurrence of each class can be computed as the cumulative sums

$$\omega_0(k) = \mathrm{P}\left(C_0\right) = \sum_{i=0}^{k} p_i \tag{2.5}$$

$$\omega_1(k) = \mathrm{P}\left(C_1\right) = \sum_{i=k+1}^{L-1} p_i = 1 - \omega_0(k) \tag{2.6}$$

where C_0 denotes the background class with intensities in the range $[0,\, k]$ and C_1 denotes the foreground objects with intensities in the range $[k+1,\, L-1]$ [27]. Using the occurrence probabilities in Equation (2.5) and (2.6) the mean pixel intensity in each class can be calculated as

$$\mu_0(k) = \sum_{i=0}^{k} i\frac{p_i}{\omega_0(k)} \tag{2.7}$$

$$\mu_1(k) = \sum_{i=k+1}^{L-1} i\frac{p_i}{\omega_1(k)} = \sum_{i=k+1}^{L-1} i\frac{p_i}{1-\omega_0(k)}. \tag{2.8}$$

Furthermore, the mean intensity of the whole image can be obtained as

$$\mu_T = \sum_{i=0}^{L-1} i p_i. \tag{2.9}$$

The between-class variance can then be computed as

$$\begin{aligned}\sigma_B^2(k) &= \omega_0(k)\left(\mu_0(k)-\mu_T\right)^2 + \omega_1(k)\left(\mu_1(k)-\mu_T\right)^2 \\ &= \omega_0(k)\omega_1(k)\left(\mu_1(k)-\mu_0(k)\right)^2.\end{aligned} \tag{2.10}$$

The optimal threshold is the value k that maximizes σ_B^2 and best separates the classes based on their intensity values [27]. The fact that the between-class variance is a measure of separability between the classes can be seen in Equation (2.10) where a larger σ_B^2 is obtained by a larger distance between the two mean intensities $\mu_0(k)$ and $\mu_1(k)$ [26]. Thus, the optimal threshold k^* is given by

$$k^* = \max_{0<k<L-1} \sigma_B^2(k). \tag{2.11}$$

Once the optimal threshold k^* has been found thresholding is performed as before in Equation (2.2) [26].

2.3.3 Multi Otsu

The Otsu method can be extended to multi-threshold problems with an arbitrary number of classes, the method is then referred to as Multi Otsu [27]. If an image contains 3 classes C_0, C_1, and C_2 the algorithm needs to find two thresholds, k_1 and k_2, where $0 < k_1 < k_2 < L-1$. Class C_0 contains intensities in the interval $[0,\, k_1]$, C_1 contains intensities in the interval $[k_1+1,\, k_2]$ and C_2 contains intensities in the interval $[k_2+1,\, L-1]$. In this case, the between-class variance that should be maximized becomes a function of k_1 and k_2

$$\sigma_B^2(k_1,k_2) = \omega_0\left(\mu_0-\mu_T\right)^2 + \omega_1\left(\mu_1-\mu_T\right)^2 + \omega_2\left(\mu_2-\mu_T\right)^2 \tag{2.12}$$

where

$$\omega_0 = \sum_{i=0}^{k_1} p_i \qquad \omega_1 = \sum_{i=k_1+1}^{k_2} p_i \qquad \omega_2 = \sum_{i=k_2+1}^{L-1} p_i \tag{2.13}$$

and

$$\mu_0 = \sum_{i=0}^{k_1} i\frac{p_i}{\omega_0} \qquad \mu_1 = \sum_{i=k_1+1}^{k_2} i\frac{p_i}{\omega_1} \qquad \mu_2 = \sum_{i=k_2+1}^{L-1} i\frac{p_i}{\omega_2}. \tag{2.14}$$

For a problem with M classes, C_0, C_1, …, C_{M-1}, the algorithm requires $M-1$ thresholds, k_1, k_2, …, k_{M-1}, and the between-class variance is given by

$$\sigma_B^2\left(k_1,k_2,\ldots,k_{M-1}\right) = \sum_{m=0}^{M-1} \omega_m\left(\mu_m-\mu_T\right)^2 \tag{2.15}$$

where ω_m and μ_m, for $m = 0,\ldots,M-1$, depend on the thresholds in the same way as in Equation (2.13) and (2.14) [26]. The optimal thresholds are the intensity values that maximize the between-class variance and can be obtained from

$$\sigma_B^2\left(k_1^*,k_2^*,\ldots,k_{M-1}^*\right) = \max_{0<k_1<k_2<\ldots<k_{M-1}<L-1} \sigma_B^2\left(k_1,k_2,\ldots,k_{M-1}\right). \tag{2.16}$$

Even though it is possible to use an arbitrary number of classes in the Multi Otsu method, it is not recommended in practice since the credibility of the detected thresholds decreases when the amount of classes increases. The method is feasible for up to three classes, but if more classes should be distinguished, other methods that use more information than just the pixel intensities are more suitable [26].

2.4 Active Contour Models (Snakes)

Active contour models use deforming splines, also known as snakes, to fit to an object in an image [28]. The snake acts on internal and external forces to minimize the composed energy dynamically. This is done by iteratively solving a partial differential equation (PDE) which deforms the curve to fit the nearest contour, being the encircled object in the image. The curve is under the influence of image forces pushing the curve towards the contour, consisting of lines and edges, and some external constraint forces. The Basic Snake algorithms depend on these external forces to put the snake close to the desired object, also known as the desired local minimum [28].

The energy being minimized is the internal and the external energy which composes the energy function [28]. If the position of the snake is represented by $\mathbf{v}(s) = (x(s), y(s))$, then the energy function can be written as

$$E^*_{\text{snake}} = \int_0^1 E_{\text{snake}}(\mathbf{v}(s))ds = \int_0^1 E_{\text{internal}}(\mathbf{v}(s)) + E_{\text{image}}(\mathbf{v}(s)) + E_{\text{con}}(\mathbf{v}(s))ds \quad (2.17)$$

where E_{internal} is the internal energy, E_{image} is the forces due to the image itself, and E_{con} represents the constraint forces introduced by the user. To minimize the energy is to minimize Equation (2.17). The internal energy arises from the bending of the snake and can be written as

$$E_{\text{internal}} = \left(\alpha(s)\left|\mathbf{v}_s(s)\right|^2 + \beta(s)\left|\mathbf{v}_{ss}(s)\right|^2\right)/2 \quad (2.18)$$

where $\alpha(s)$ and $\beta(s)$ are adjustable weights, or parameters, to control the amount of stretch and curvature of the snake. These terms control the spacing of the points in the contour, and the oscillations of the contour, respectively. To ensure that the snake is attracted to the object of interest, it is crucial that the snake is attracted to salient features [28]. These features are the lines, edges, and terminations, and these energy functionals together make up the total image energy as

$$E_{\text{image}} = w_{\text{line}}\, E_{\text{line}} + w_{\text{edge}}\, E_{\text{edge}} + w_{\text{term}}\, E_{\text{term}}. \quad (2.19)$$

Each energy function has its own designated weight that could be altered to change the snakes' behaviour. A higher weight indicates that this feature will contribute more to the image force. The energy function E_{line} from Equation (2.19), which is the line function, is defined as the image intensity as $E_{\text{line}} = I(x, y)$. It attracts the snake to align with the nearest light or dark lines, depending on the sign of w_{line}. To detect edges, $E_{\text{edge}} = -|\nabla I(x, y)|^2$ which will affect the snake to be attracted to large gradients in the image. The last energy function is utilized to find corners and terminations. If $C(x, y) = G_\sigma(x, y) * I(x, y)$ is a smoothed version of the original image smoothed with a Gaussian filter and with the gradient angle $\theta = \tan^{-1}(C_y/C_x)$, the unit vector along and perpendicular to the gradient angle can be written as $\mathbf{n} = (\cos\theta, \sin\theta)$ and $\mathbf{n}_\perp = (-\sin\theta, \cos\theta)$. From this, the termination function energy can be represented as

$$E_{\text{term}} = \frac{\partial\theta}{\partial\mathbf{n}_\perp} = \frac{\partial^2 C/\partial\mathbf{n}_\perp^2}{\partial C/\partial\mathbf{n}_\perp} = \frac{C_{yy}C_x^2 - 2C_{xy}C_xC_y + C_{xx}C_y^2}{(C_x^2 + C_y^2)^{3/2}}. \quad (2.20)$$

The last functional energy from Equation (2.17) is E_{con}, which is energy from the user. The user can use this term to guide the snake toward the desired object or away from local minima. A visualization of the evolution of the Basic Snake method can be seen in Figure 2.1.

Figure 2.1: An illustration of the evolution of an active contour model fitting a snake to a star shape.

2.4.1 Morphological Operations

Morphology is a term from biology used to describe form and structure. In the context of image processing, mathematical morphology is implemented [29]. It can be used to extract components in an image. Such components can be used for the representation and description of shapes in the image. Examples of these are boundaries, skeletons and the convex hull. Morphology can also be used in the pre-and postprocessing steps as filtering or thinning.

There are two fundamental morphological operations called erosion and dilation which all other operations are based on [30]. To explain the two operations A is a binary image in Euclidean space E and B is a kernel. Erosion of A by B can be defined as

$$A \ominus B = \{z|(B)_z \subseteq A\} \tag{2.21}$$

where z is the vector such that B translated by z is contained in A. This operation can be observed in Figure 2.2 as a thinning or shrinking operation. In this example, the image is binary, meaning that the centre pixel of B is kept as 1 (true) if all pixels covered by B are true, which is the same as B not sharing any pixels with the background. Otherwise, the centre pixel is set to zero. Dilation, on the other hand, is an enlarging operation defined as

$$A \oplus B = \{z|(B)_z \cap A \neq \varnothing\}. \tag{2.22}$$

Dilation of A by B can be seen in Figure 2.3 where the A has been enlarged. This example is of a binary image, and when the kernel is placed such that at least one pixel coincides with one pixel in A, the centre pixel of B is set to 1 (true) [30].

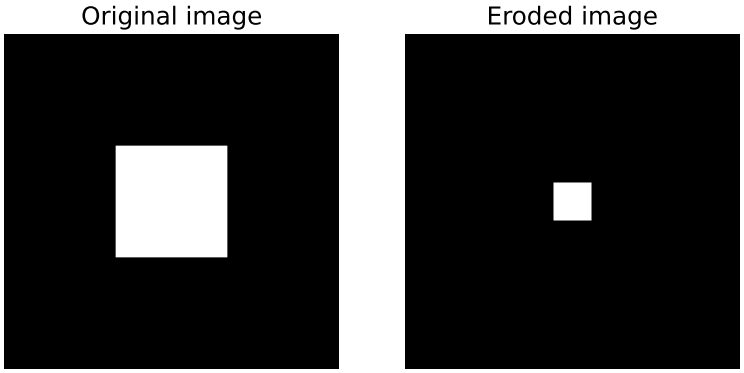

Figure 2.2: Example of an eroded binary image.

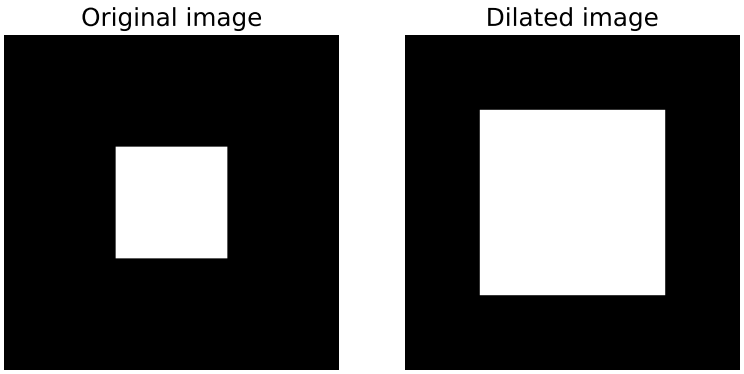

Figure 2.3: Example of a dilated binary image.

From erosion and dilation comes the closing and opening operations [31]. The opening operation is used for smoothing and filtering and is defined as $A \circ B = (A \ominus B) \oplus B$. This operation is simply the erosion of A by B followed by the dilation by B and can be observed in Figure 2.4. If applied the other way around, the closing operation is defined as $A \bullet B = (A \oplus B) \ominus B$. First, the dilation of A by B is followed by the erosion of B. The closing operation could also be used for smoothing but it is more effective for closing holes and filling gaps as can be observed in Figure 2.5.

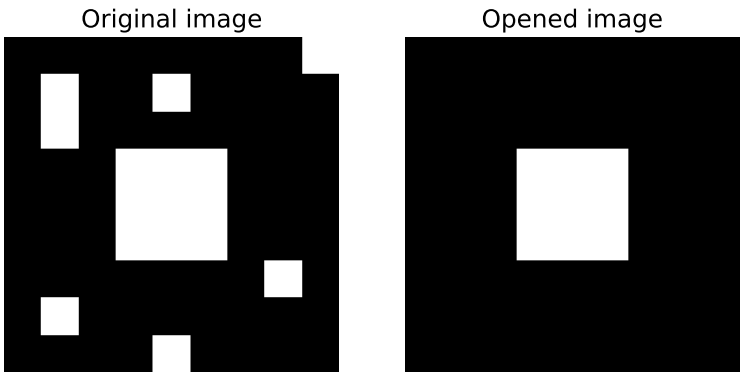

Figure 2.4: Example of the opening operation on a binary image.

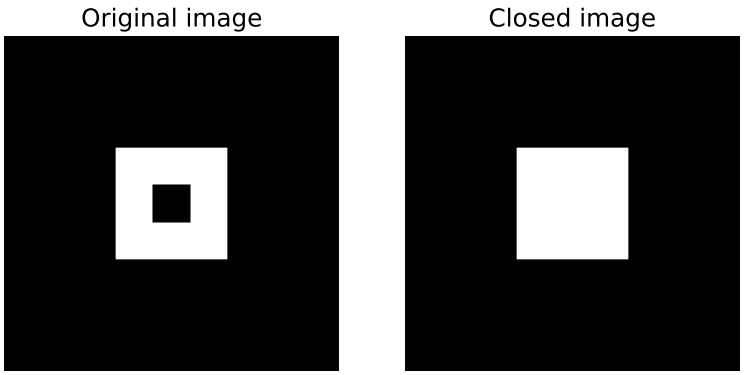

Figure 2.5: Example of the closing operation on a binary image.

2.4.2 Morphological Snake

The Basic Snake algorithm has its flaws. The numerical calculations made in the algorithm make the approach slow and complex, which can cause instability problems. In addition to this, the model is sensitive to user input, such as the initialization and parameter selection [32]. From the original Basic Snake algorithm. progress has been made to develop the model in a few different directions. One model that has come from this is the Morphological Snake [33]. The Morphological Snake algorithm uses a morphological approach when solving the PDE in Equation (2.17) in the Basic Snake model. This is accomplished by approximating the solution to a composition of morphological operators. Eliminating the need for a finite difference numerical implementation of the differential operator has the advantage of being a faster algorithm. Moreover, there is no need for re-initialization making it a fast and stable curve evolution algorithm. For a more detailed description of Morphological Snake and the algorithm used, see the original article from 2010 by Álvarez et al. [33].

2.5 Deep Learning

Traditionally, CT scans of the liver were analyzed manually, slice by slice by a physician, an approach that is both time-consuming and demands high expertise [34]. The result is solely dependent on the skill and experience of the physician [34]. This kind of segmentation has not been commonly used since semi-automatic methods were introduced [35]. In semi-automatic methods, a physician performs the segmentation with the help of computer algorithms. The method thus requires user input but improves efficiency and saves time compared to the manual method [34, 35]. However, the end goal when developing tumor segmentation methods is to find automatic methods that perform the segmentation fast without the need for user input. This both saves time and makes the result of the segmentation independent of the experience of the physician [35].

The developments in machine learning, especially within deep learning, have opened up more efficient ways to perform tumor segmentation [36]. Convolutional neural

networks have been developed and used to perform a variety of different tasks in medical image analysis, for example, classification and segmentation of medical images. The idea behind these deep learning models is that they learn image features directly from the data during training. The trained models can then be used to perform the segmentation automatically without the need for user input, thus saving time for the physician [36]. During training on, for example, liver images with tumors the model learns features such as textures, shape and intensities of the liver and tumor tissues [36]. Deep learning can be extra useful for identifying small tumors that are hard to spot for humans and thus has the potential to improve the accuracy of tumor segmentation [35]. The use of deep learning models in tumor segmentation has increased significantly since the successful implementation of the U-Net model in 2015 [36].

2.5.1 Basic Concepts of Deep Learning

Before introducing the U-Net architecture some basic concepts of deep learning will be explained. Deep learning is a special kind of machine learning that is based on artificial neural networks [37]. Feedforward neural networks are typical deep learning models. They consist of at least three layers, an input layer, one or more hidden layers, and an output layer. Information is fed through the network from the input to the output layer [38]. Almost all deep learning algorithms are based on a combination of a model, a loss function, an optimization algorithm, and a dataset [37].

Optimization is a central part of most deep learning algorithms and is often achieved by minimization of an objective function, commonly called a **loss function** [39]. More information about loss functions is presented in Section 2.7. A common optimization algorithm in deep learning is **gradient descent** [39]. It works by computing the gradient of the loss function and then taking a small step in the negative gradient direction. In this way, the algorithm uses the gradients of the function to follow it downhill to a minimum. The step length is determined by a parameter called the **learning rate**, which is thus an important parameter for the result of the optimization [39]. Two common optimization algorithms in deep learning, both based on the gradient descent algorithm, are SGD with momentum and **Adam** [40].

It is essential that the deep learning algorithm achieves a high performance not only on data it has been trained on but also on new data that has previously not been seen by the algorithm [37]. To measure how well the model performs on new data the available dataset is often split into two parts; training data and test data. The performance on the **test set** should be a measure of how well the model performs in a real scenario, therefore it is important that the test set is kept separate from the training data and that it is never used to make any choices about the model [37]. The training data on the other hand is used during training and is often split into two parts; a **training set** and a **validation set**. The training set is used to learn the parameters, and the validation set is used to estimate the performance of the model during or after training [37]. It is common to use 80 % of the training data for training and 20 % for validation [37]. Dividing the data into fixed training

and test sets can be problematic if the dataset is small. This is because there exists statistical uncertainty around the mean test error when the model has been evaluated on a small test set [37]. The statistical uncertainty makes it difficult to draw conclusions about the results and compare the performance of different models on the test set. There are ways to decrease the uncertainty at the expense of a higher computational cost, for example by using a method called k-fold cross-validation [37].

The number of times the algorithm iterates through the entire training set during training is called the number of **epochs** [41]. Optimization algorithms can process the training set in different ways during training, either all samples in the training set at once, a smaller part of the samples at a time, or only one sample at a time [42]. Optimizers that process the entire set at once are called **batch** gradient methods, while those that process a smaller part are called **minibatch** gradient methods, and those that process one sample at a time are called **stochastic** gradient methods [42]. Most optimization algorithms in deep learning use some kind of minibatch gradient method. The size of the minibatch, the number of samples processed simultaneously by the optimizer, is often called the **batch size**. Typical batch sizes are 32, 64, 128 and 256 [42].

Large models with much capacity often have a tendency to adapt too much to the training set [41]. This can be seen by studying how the training error and the validation error change over time during training. If the training error continues to decrease while the validation error starts to increase after some time, it suggests that the model becomes less general and **overfitting** occurs [41]. There exist several strategies to prevent overfitting. The best way to do this is to add more training data, but this can be hard to achieve since the availability of data is often limited [41]. One way to increase the amount of data is to create ”fake” data from the existing data, a process called **data augmentation** [41]. Data augmentation is a very efficient way to prevent overfitting in, for example, image classification tasks. Transformations such as translating, rotating, and scaling can be used to change the appearance of the images without changing the content and hence produce new images [41].

Another way to prevent overfitting is by using **early stopping**, a simple and effective strategy based on returning the model that achieved the lowest error on the validation set [41]. In the training process, the model parameters are periodically saved whenever there is an improvement in the error on the validation set. The training continues until a predefined number of epochs pass without any further improvement in the validation error. At that point, the model parameters associated with the best validation error are saved, rather than the most recent ones [41]. Another strategy to prevent overfitting is **dropout** [41]. Dropout involves randomly deactivating certain connections or nodes in the neural network during training, preventing all information from being carried throughout the entire network [43]. This technique encourages the model to learn more robust and independent features, thus reducing overreliance on specific patterns present in the training data. One more alternative that can be used to prevent overfitting is **L2 regularization**, also called

weight decay, in which a penalty term is added to the loss function that pushes the weights to become smaller [41]. This reduces the complexity of the model and makes it less likely to overfit.

Convolutional neural networks (CNNs) are a special kind of feed-forward neural networks commonly used with for example image data [38]. As the name suggests, CNNs contain at least one layer that uses the mathematical operation **convolution** instead of general matrix multiplication that otherwise is standard in neural networks. These layers are called **convolutional layers** and are used to extract image features such as lines and corners. In a typical CNN, each convolutional layer is followed by a **nonlinear activation function** and a **pooling layer** [38]. The nonlinear activation function introduces nonlinearity to the network, and the standard activation function in a CNN is the **rectified linear unit** (ReLU) [38]. The purpose of the pooling layer is to downsample the feature maps, which makes the network more computationally efficient. A common form of pooling is **max-pooling** [38].

During the development and training of a deep learning model, there are a large number of choices that need to be done, for example regarding which model, optimizer and loss function to use [40]. Depending on how the training proceeds, it is also important to make decisions about how to change the deep learning algorithm to achieve better performance in the end. Such decisions can for example be to increase or decrease the model capacity, to add more data, to add methods to prevent overfitting and to tune hyperparameters [40]. The values of the hyperparameters are chosen before the training starts, and they affect the performance of the trained model as well as the runtime and memory cost of the training. Hyperparameters can both be chosen manually, which demands more understanding about how they affect training, and automatically, which comes with a higher computational cost [40]. One of the most important hyperparameters is the learning rate, which has a high impact on the optimization algorithm [40]. If the learning rate is too large, the training error can increase and if it is too small, the training error can get stuck at a high level. Other examples of hyperparameters include weight decay coefficient and dropout rate [40].

2.5.2 Original U-Net

The original U-Net was presented in 2015 by Ronneberger et al. [44]. Since then, the network has been used in many different applications of medical image segmentation [35]. Several variations of U-Net have also been implemented, some described in Subsection 2.5.3. U-Net is an example of a fully convolutional neural network that does not contain any fully connected layers. The network consists of a contracting part, where the feature map is downsampled, and an expansive part, where the feature map is upsampled [44]. The contracting part is also called an encoder, whereas the expansive part is called a decoder [36]. The encoder and decoder are almost symmetric to each other and contain the same number of steps down or up, which allows the network to be visualized in a U-shape, see Figure 2.6. The input image is fed into the network on the left side. The task of the encoder is then to

extract features and contextual information from the input image. Then the task of the decoder is to learn where the features are located [36].

The encoder has a structure that is typical for a convolutional neural network. Each block consists of two convolutional layers with kernel size 3×3 and no padding followed by ReLU activations, and a max-pooling layer with kernel size 2×2 and stride 2 [44]. The application of the max-pooling layer results in a downsampling of the feature map, where the dimensions are halved. However, as the feature map progress through the encoder, the number of channels in the feature map is doubled for each step down [44]. The decoder then restores the image to its original dimensions by upsampling the feature maps [36].

Each block in the decoder starts with a transposed convolution with a 2×2 kernel which results in a duplication of the dimensions of the feature map and a halving of the number of feature channels [36]. Then the feature map is concatenated with the cropped feature map from the corresponding block in the encoder, which is fed to the decoder block via skip connections [44]. Skip connections are, as the name suggests, connections from an earlier layer in the network directly to a later layer in the network that skips some intermediate layers. A skip connection is thus a shortcut in the network [36]. The reason why skip connections are useful in U-Net is to localize features in the upsampled feature map by combining the upsampled feature map with the high-resolution feature map from the encoder [36, 44]. After the concatenation follows two convolutional layers with kernel size 3×3 and ReLU activations, similar to the contracting part [44]. As a final block in the network, a convolution with kernel size 1×1 is applied to map all 64 feature vectors to the number of output classes, in this case, 2 [44].

Different kinds of U-Net architectures are commonly used in liver tumor segmentation tasks [3, 34]. Almost all architectures in the LiTS challenge were based on U-Net, only two submissions used other architecture types. To improve the performance different modifications of the original network are made, for example by using different kinds of skip connections [3, 36]. It is also common to use the U-Net architecture in a cascaded form, where the first U-Net model segments the liver and passes its output as input to the second U-Net model, which segments the tumors [3, 34]. Most submissions in the LiTS challenge used the approach with cascaded U-Nets [3].

2.5.3 Modified U-Nets

In the field of deep learning for medical image segmentation, the U-Net architecture is widely used. Modified versions of U-Net have also emerged, particularly in the LiTS benchmark mentioned in Section 1.4. Various submissions to the benchmark introduced altered U-shaped networks. The 2018 winner was Isensee et al., who presented their nnU-Net model [9]. This model, referred to as the ”no new” U-Net, leverages existing manual method configuration processes to achieve a self-configuring model. It encompasses preprocessing, network architecture, training,

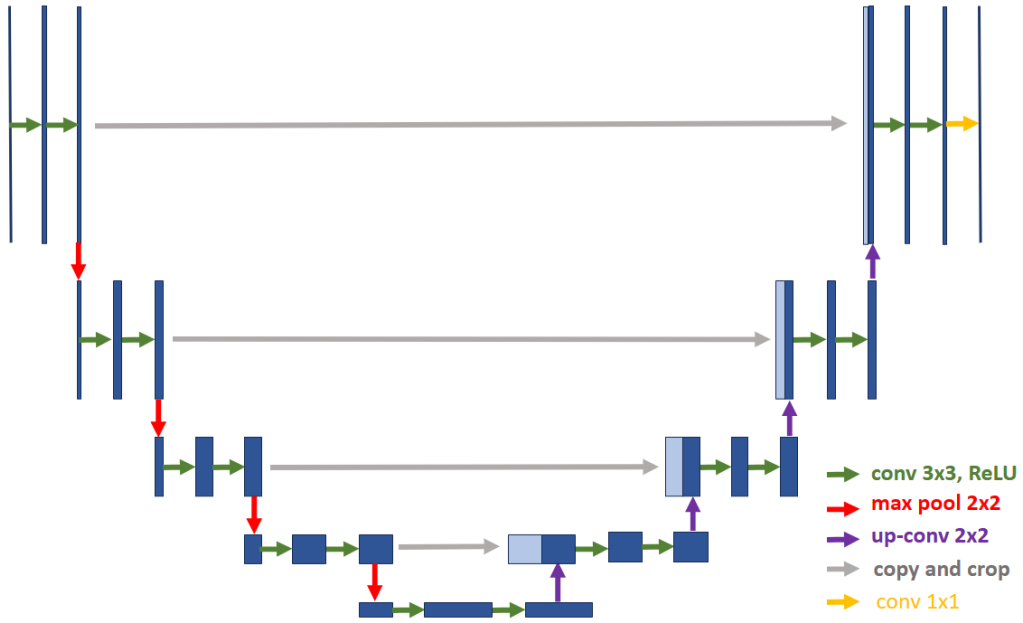

Figure 2.6: Overview of the U-Net architecture. The input image is fed into the network on the left side and the output segmentation map is returned on the right side.

and postprocessing steps, applicable to any type of medical image input. The method was developed and tested using all ten datasets of the Medical Segmentation Decathlon, where LiTS focuses on liver and liver tumors [45].

The self-configuring nature of nnU-Net is based on dividing parameters into three categories: fixed, rule-based, and empirical [9]. Design choices that remain consistent across datasets, such as the U-Net structure, are considered fixed. Dependencies are then formed to establish a standardized dataset representation, including image size and voxel spacing. Interdependent heuristic rules are used to create these dependencies, enabling nearly instantaneous execution. For a more detailed description and examples of these rules, please refer to the original article [9]. The remaining design choices are empirically determined based on the training data. Consequently, when the network is applied to a new dataset, it configures itself automatically without requiring manual input. When trained and tested on the LiTS dataset, described in Subsection 2.8.2, nnU-Net achieved superior results compared to previous LiTS benchmark challenges. The evaluation methods used in the benchmark and in this thesis work are presented in Section 2.6. Currently, nnU-Net has achieved a dice score of 0.739 and a recall value of 0.554, where higher values close to 1 indicate better results for both metrics.

Another modified U-Net model developed and evaluated on the LiTS dataset is the U^n-Net by Tran et al. [35]. U^n-Net is a multi-layered version of the original network that introduces changes to the skip connection path, max-pooling, and upsampling. In this modified architecture, all convolutional units in the encoder nodes serve as input to the next layer, as well as to the same level decoder node. This modification

allows for greater information flow within the network. Figure 2.7 provides a visualization of the original U-Net structure (top) and the U^n-Net version (bottom). Tran et al. explored U^2-Net and U^3-Net, where the power indices represent the number of convolution nodes in that network.

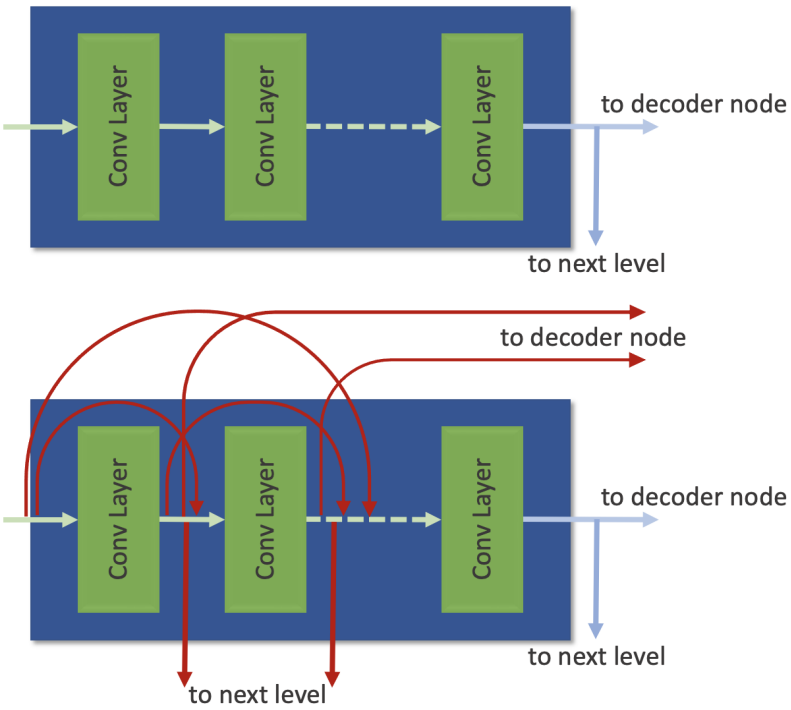

Figure 2.7: One level of the original U-Net structure on the top, and the modified structure for Un-Net on the bottom, where all the red lines are the new features.

The U^n-Net suggested by Tran et al. included additional modifications [35]. One being dilated convolution, which extends the convolution region without pooling, was employed to cover a wider range of information while preserving spatial details, potentially enhancing the segmentation[35, 46]. Another feature introduced in the network was a dense structure for the nodes, addressing the problem of vanishing gradients and improving feature propagation efficiency [35, 47]. Although this network was not part of the LiTS benchmark challenge, it was evaluated using the dataset. It's important to note that the dataset used for U^n-Net evaluation was not complete, as some data (volumes 28 to 47) had been removed, and there was no access to the labelled test volumes. With this in mind, for tumor segmentation, U^2-Net and U^3-Net achieved dice scores of 0.706 and 0.737 respectively with dilated convolution. However, no recall value was presented for these methods. The impressive results obtained from both nnU-Net and U^n-Net suggest that both models are valid choices for further investigation, making them recommended options for future thesis work

2.6 Evaluation

For this work to be as objective and fair as possible, standardized evaluation metrics are used. The first one is Dice's Similarity Coefficient (DSC), also known as dice score, defined as

$$\mathrm{DSC}(G, S) = \frac{2|G \cap S|}{|G| + |S|} \tag{2.23}$$

where G and S denote two masks, the ground truth and the segmentation, respectively [3]. The DSC is a measurement of the overlap between the two masks, and a value of 1 indicates a perfect segmentation, whereas a value of 0 means that there is no overlap [35]. The DSC can also be defined in terms of true positives (TP), false positives (FP), and false negatives (FN) as

$$\mathrm{DSC} = \frac{2 \cdot \mathrm{TP}}{2 \cdot \mathrm{TP} + \mathrm{FP} + \mathrm{FN}}. \tag{2.24}$$

Figure 2.8 shows that the true positives represent the same pixels as the intersection of the segmentation and the ground truth and that the sum of all pixels in the two masks can be calculated as $2 \cdot \mathrm{TP} + \mathrm{FP} + \mathrm{FN}$.

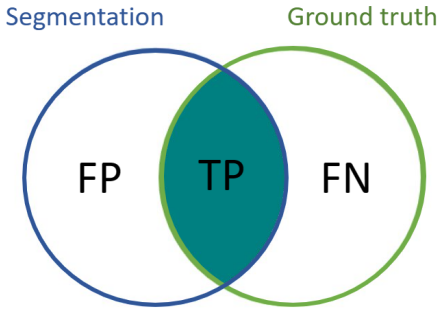

Figure 2.8: The true positives (TP) are pixels correctly classified as tumor and can be computed as the intersection of the segmentation and the ground truth, the false positives (FP) are pixels incorrectly classified as tumor and the false negatives (FN) are pixels that are incorrectly classified as background.

The second metric used is recall, which relates the true positives to the false negatives as

$$\mathrm{recall} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{2.25}$$

and measures what proportion of the tumor pixels are found by the model [3].

2.7 Loss Functions for Image Segmentation

The choice of the loss function is essential for the performance of the deep learning model and the result of the segmentation. Some examples of loss functions used in semantic segmentation tasks are binary cross-entropy, cross-entropy, dice loss and tversky loss [48]. When there exists class imbalance, which is often the case in tumor segmentation where the number of tumor pixels is far less than the number

of background pixels, the loss function should be chosen with this in mind. Loss functions that are common to use in cases with class imbalanced data are dice loss and weighted cross-entropy [49].

2.7.1 The Dice Loss

The dice loss is based on the dice score presented in Section 2.6, and it is defined in Equation (2.26) [48]. In its adaptatione to a loss function, the goal is to minimize the loss $1 - \text{DSC}$ rather than maximizing the overlap, DSC. The ones have been added to the numerator and denominator to avoid division by zero if $|G| = |S| = 0$.

$$\text{dice loss}(G, S) = 1 - \frac{2|G \cap S| + 1}{|G| + |S| + 1} \tag{2.26}$$

During training, the loss is calculated for each minibatch iteration. The input to the dice loss function is thus on the form (`batch_size`, `n_channels`, `img_size`, `img_size`), where `batch_size` is the size of the minibatch, `n_channels` is the number of predicted maps and `img_size` is the image size. In the case when the network should predict two classes (background and foreground) two predictions are outputted from the model. The first map predicts the probability that each pixel is a background pixel and the second map predicts the probability that each pixel is a foreground pixel. An example of what the two predictions can look like for an image is shown in Figure 2.9, with the corresponding test image and final prediction shown in Figure 2.10. Note that the background map shows high probabilities for the background pixels whereas the foreground map shows high probabilities for the tumor pixels. The final prediction for the tumor is obtained by classifying each pixel as belonging to the class with the highest probability in the two prediction maps.

In this work, the dice loss function is used in two ways during training. In the first variation both the background and foreground channels are used to calculate the loss, which means that correctly classifying a background pixel will be equally rewarding as correctly classifying a tumor pixel. From here on this variation is called the *two-channel dice loss*. In the other variation only the foreground channel is used to calculate the loss, which means that only the tumor predictions will be used to compute the loss. From here on this variation is called the *one-channel dice loss*. The reason why both variations are used is that in unbalanced segmentation tasks, where the foreground segmentations are very small compared to the background segmentations, the background segmentation could have a too big impact on the loss. This is the case in the liver-segmented input images, but not necessarily the case in the cropped region-of-interest input images. Thus both variations are interesting to explore.

2.8 Datasets

There exist open-source datasets containing both liver and tumor, available for use when developing an image segmentation method. In Table 2.2 an overview of the available datasets can be observed. When choosing among datasets, there are certain

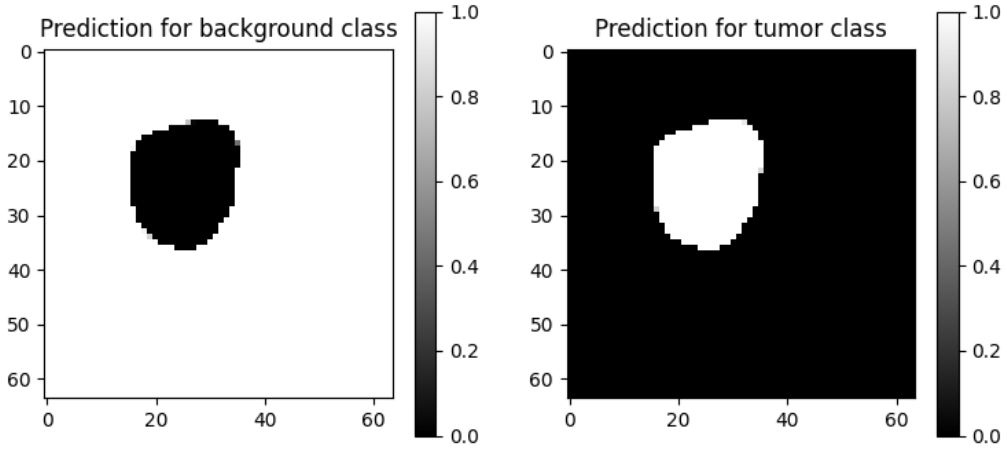

Figure 2.9: An example of a prediction for the background class and the foreground class of the test image in Figure 2.10.

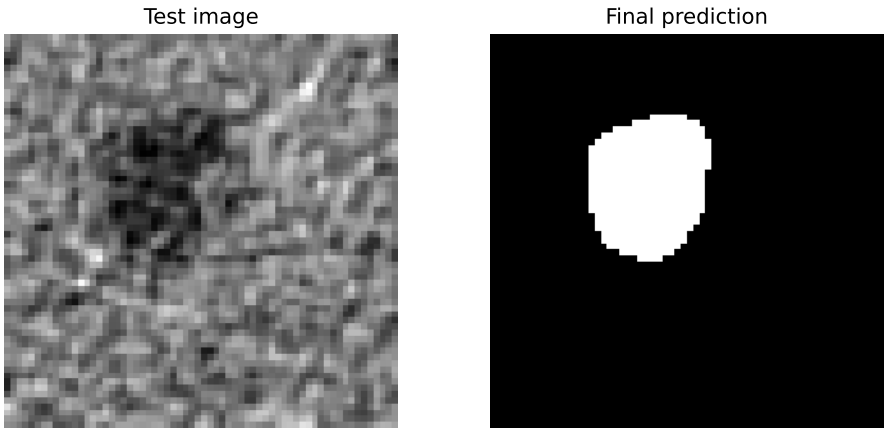

Figure 2.10: A test image and its final prediction, based on both the prediction for the background class and the foreground class.

aspects that need to be considered. For this project, it is desirable for the images to contain both liver and tumor, as well as to come from the imaging modality CT. It is also beneficial if the data is annotated since it makes the evaluation process, and the training of the network simpler. In the following Sections, the dataset used in the previous thesis work as well as the dataset chosen for this project will be discussed in detail. Other datasets seen in Table 2.2 have not been considered due to the lack of annotation, tumor or liver information, or an insufficient size.

2.8.1 TCGA-LIHC

The Cancer Genome Atlas Liver Hepatocellular Carcinoma Collection (TCGA-LIHC) is a part of The Cancer Genome Atlas Program (TCGA)[1] and it is available through the Cancer Imaging Archive[2]. The TCGA researchers have collected data

[1]https://www.cancer.gov/ccg/research/genome-sequencing/tcga
[2]https://wiki.cancerimagingarchive.net/pages/viewpage.action?pageId=6885436

Table 2.2: Summation of the available datasets containing liver images [3].

Dataset	Institution	Liver	Tumor	Annotation	#Volumes	Modality
TCGA-LIHC	TCIA	✓	✓	×	1688	CT, MR, PT
MIDAS	IMAR	✓	✓	×	4	CT
3Dircadb-01, 3Dircadb-02	IRCAD	✓	✓	✓	22	CT
SLIVER'07	DKFZ	✓	×	✓	30	CT
LTSC'08	Siemens	×	✓	✓	30	CT
ImageCLEF'15	Bogazici Uni.	✓	×	✓	30	CT
VISCERAL'16	Uni. of Geneva	✓	×	✓	60/60	CT/MRI
CHAOS'19	Dokuz Eylul Uni.	✓	×	✓	40/120	CT/MRI
LiTS	TUM	✓	✓	✓	201	CT

for 20 different cancers in multiple parts of the body, for example in liver, breast and brain [50]. The dataset contains data from 97 patients that suffer from hepatocellular carcinoma [51]. The data has been collected from many different locations around the world, mostly during routine care and not as a part of clinical trials or research studies. As a result, the data is heavily mixed in terms of image modalities used (CT, MRI, and PET images are present in the dataset) and also in terms of manufacturers of scanners and acquisition protocols. The dataset does not contain any annotated data [51].

Furthermore, the dataset does not follow a clear structure and to find useful data for a specific task it is necessary to manually go through the dataset patient by patient. The dataset contains folders of DICOM files for 97 patients and for one patient the number of scans can vary a lot. Things that differentiate the patients are the number of scans available, whether the scans are with or without contrast, in which anatomical planes the scans have been taken, whether the scans are taken in arterial or venous phase and whether the scans contain artifacts. In some cases there exist multiple scans obtained at the same time but with different settings, in other cases there exist scans that are taken during different times with the same settings and in other cases, both settings and times vary a lot. Data from TCGA-LIHC was used in the previous Master's thesis [7].

2.8.2 LiTS

The Liver Tumor Segmentation (LiTS) dataset is a dataset intended for the LiTS benchmark challenge. The challenge has been held a number of times and the competition is intended to promote development of deep learning models for liver and liver tumor segmentation [3]. The contributions have been included both multichannel 2D input as well as 3D input.

The dataset is open-source and is available for anyone to download and use on the LiTS challenge website[3]. Before the LiTS dataset, the problem with other open-source datasets of the liver is that they are very small in size and they lack annota-

[3]www.lits-challenge.com

tion. These problems affect the ability to train deep networks since there is a need for both larger datasets, to make sure the network is not biased and works in many cases, and the annotation is crucial in the training and evaluation process. Because of this, the LiTS dataset is a great resource. The dataset is the result of a collaboration with seven hospitals and research institutions where the data were annotated manually. The annotation was performed slice-wise by radiologists with more than three years of experience in oncologic imaging. The labels were assigned as healthy liver, tumor, or background. The segmentation was then verified by another three radiologists and the most senior one had the decisive vote in the case of labelling conflict.

The dataset itself consists of both training and test sets, but because it is developed with the competition in mind, the labels for the test set are only available for the organizers. Hence, the dataset consists of 201 CT volumes, where 131 are annotated training volumes. These volumes contain CT scans of the abdomen and the only processing done is the anonymizing step as well as the transformation into NIfTI files. The data is diverse and contains both primary liver cancer as well as secondary liver cancer with metastases from colorectal, breast, and lung primary cancers. The data varies from no tumor up to 12 tumors, where both size and location are varying as well. Also, variations in lesion-to-background levels occur and the scans are taken both pre and post-therapy and are acquired with different CT scanners. The latter causes the resolution in the scans to differ as well. The in-plane resolution varies from 0.56 to 1.0 mm, and the slice thickness varies from 0.45 to 6.0 mm. The number of slices per volume also varies, and some images have artifacts, for example, caused by metal, which is commonly found in patients. The properties of the dataset are summarized in Table 2.3.

Table 2.3: A summary of the characteristics of the LiTS dataset. The table displays the value ranges for the different parameters.

Characteristic	**Range**
Volumes	201
Volumes annotated	131
Scan size	512 x 512
Axial slices	42 to 1026
Number of tumors	0 to 12
Tumor volume	38 mm^3 to 1231 mm^3
In-plane resolution	0.56 mm to 1.0 mm
Slice thickness	0.45 mm to 6.0 mm

3
Methods

In this chapter, the methods of the thesis are presented. First, the general workflow of the entire project is summarized. Then the tumor segmentation methods are presented in more detail, both for the algorithms and the deep learning models.

3.1 General Workflow

The thesis work was divided into two main parts that were partly conducted in parallel. The first part was to perform tumor segmentation using different thresholding and active contour model algorithms. The second part was to perform tumor segmentation by training a deep neural network to perform the same task. Before starting work on any of the parts, appropriate data had to be collected. The first task was to search for an available open-source dataset containing both the liver and tumor while being annotated. The available datasets are described in more detail in Section 2.8.

Before starting the process of developing the different segmentation methods, the Python IDE (Integrated Development Environment) PyCharm was installed, and a suitable environment was created for the project. Appropriate Python libraries was used. For version control, Git was used. To familiarize with the different Python libraries fitting for the task, code from the previous project was examined. The starting point for the programming part of the project was to rewrite some of the already existing code from the prior thesis work. To be able to write new relevant code for the project, theoretical research was conducted about the different segmentation algorithms. From the research, segmentation scripts performing tumor segmentation with thresholding methods and active contour models were written. Each method is further explained under Section 3.2 and 3.3.

Since no prior work had been carried out using deep learning, this part of the project started with researching scientific articles. To find relevant information search tools such as *PubMed*, *Google Schoolar*, and *Chalmers library* were used. From an extensive collection of information, limitations and choices had to be made to continue the work with one network. For more information about the implementation of the deep learning model, see Section 3.4.

To be able to evaluate and compare the performance of the different methods an evaluation method was decided, see Section 2.6. The algorithms were evaluated us-

ing 107 images, with each randomly selected from the 107 patients with at least one liver tumor. The deep learning models were evaluated using the test set, consisting of all image slices with tumors from 11 patients. To be able to compare the result, 11 images were randomly picked from the test set, one image from each patient. These were then used to evaluate each method and the result could then be compared. From the comprehensive results, some conclusions could be drawn.

3.1.1 Preparation of Data

From the research on the different datasets, it was decided to focus on the LiTS dataset from Subsection 2.8.2 because of the advantages of having a larger and annotated dataset. Using open-source datasets also ensure that there are no ethical considerations to make. Since only the training set was annotated only these volumes were used. This was decided since it would require both a lot of time and expertise, not possessed by the authors, to annotate the data. The first step when preparing the data was to make sure it all looked as expected. Visual inspection of the data resulted in a dataset of 120 volumes appropriate for this project work. Volumes that did not display as CT scans or contained empty files were removed, concluding that the data had been corrupted. The volumes left were then inspected, and some notes about the number of tumors were taken. Lastly, the NIfTI files were converted to DICOM files for both the CT scans and the labels. All DICOM files corresponding to one NIfTI, or one patient, were placed in one folder. From these folders, DICOM files could then be read into Python, where the preprocessing took place.

When the DICOM files were read into Python, they were converted into arrays. This made it possible to work with and plot the images. After this conversion was made, the preprocessing took place. First, the background of the scans was set to zero. Then, the images were converted to HU. The importance of this comes from that the CT volumes imaged using different scanners results in scans with different value ranges. To convert the scans to HU, information stored in the metadata in the DICOM files was used. The relation between the stored values and the output values is as follows

$$\text{Output units} = \text{Rescale Slope} \cdot \text{Stored Values} + \text{Intercept} \tag{3.1}$$

where the rescale slope and intercept are values stored in the DICOM files [52].

After converting to HU it was desirable to perform some kind of thresholding to get rid of redundant information in the scans, such as bone and air. Since the Hounsfield values for air and bone are known, thresholding could be performed to get rid of them. To make sure that no liver or tumor tissue was removed by the thresholding, the range of values to keep was chosen to be between -20 to 150 HU.

For the classical algorithms, which are developed to only find one tumor per scan, the last step of the pre-processing was to remove every tumor but the largest from the label to be able to compare the prediction to the label.

3.2 Thresholding Segmentation

The input to the thresholding methods global thresholding and Multi Otsu thresholding was a preprocessed image slice. The segmentation was then performed in two steps; first by choosing a region of interest and then by performing tumor segmentation. Two different approaches to selecting a region of interest were tested, the first one was to choose the entire liver as the region of interest and the second one was to choose a small square around the tumor as the region of interest. Then, the chosen region of interest was used as input to the tumor segmentation algorithm.

3.2.1 Choice of Region of Interest

An image only containing the liver could be obtained by performing automatic liver segmentation. Since the project is limited to not including further development of the liver segmentation algorithm, the method from the prior project was utilized. The first step in this liver segmentation was to perform binary thresholding, using a specific pixel intensity as a threshold. This threshold was chosen to get a binary mask of all image structures with intensity values above the chosen threshold, which contains the intensity of the liver, see Table 2.1. The result was that all pixel intensities above the threshold were set to 255 and all intensities below the threshold were set to 0. After that morphological operations were performed on the binary mask. Those include opening to get rid of noise, closing to close holes in the structures in the image, and erosion to shrink and separate objects in the image. See Subsection 2.4.1 for a more detailed description of the different morphological operations.

Then all connected components, all remaining areas, in the image were labelled and the label corresponding to the second largest area was chosen. Connected components labelling consists of assigning a unique label to all pixels in an area of pixels - or a connected component [53]. These labels can then be used to distinguish between the different areas in the image, and in this way, it is possible to choose which area to keep. The unique labels assigned in this method depend on how large the area is, or how many pixels it includes, where the largest area is assigned label 0, the second largest label 1, and so on. Saving the second largest area, or the connected component with label 1, is desirable because the largest area should correspond to the background and the second largest area to the liver. The last step to create the liver mask was to fill in any holes in the mask by finding the internal contours and drawing them. An image containing only the liver (and the tumor) could then be created by using the liver mask on top of the preprocessed image.

An alternative method to obtain a liver-segmented image is to utilize the available liver label, the ground truth, for each image. The liver label could then simply be used on top of the preprocessed CT image to obtain an image only containing the liver. Making use of this method ensures that the tumor segmentation method can be evaluated without the effect of the performance of the liver segmentation method.

The second way to choose the region of interest was to select a small square around

the tumor. An image only containing this could be obtained by cropping the pre-processed CT scan according to an approximate tumor location. In this project, these images were obtained by cropping the stack of image slices containing tumor based on the information on tumor location from the tumor labels. The dimensions of the region of interest, the side length of the square, were selected to be three times the tumor diameter. In practice, the approximate location of the tumor could be marked by the surgeon.

3.2.2 Global Thresholding

During liver segmentation, as part of the global thresholding for tumor segmentation, an interval called the liver range, containing most of the pixel values in the liver was also saved. The liver range was then used in the tumor segmentation, described in Section 3.2.2. This range of liver values was obtained by selecting an interval in which a given percentage of the pixel intensities in the liver was included. To do this, all zero-intensity pixels (background) were removed and then the rest of the intensities were sorted from largest to smallest and each pixel value, even though not unique, was saved. Then depending on what percentage of pixels were to be included, the highest and the lowest pixel values were excluded from the range.

The obtained liver-segmented image or the cropped region of interest was then used as a starting point to segment the tumor. When a liver-segmented image was used, the first step in the tumor segmentation was to remove all intensity values within the previously obtained liver range. The idea behind this is that the healthy liver tissue and the tumor have different intensities, therefore deleting most of the liver range should not correspond to deleting the tumor. Then the rest of the tumor segmentation was carried out in a very similar way to the liver segmentation. The same morphological operations to get rid of noise and close holes were performed, connected components were used to choose the second largest area, and holes in the tumor mask were filled by finding and drawing the contours. The final step in the tumor segmentation was to dilate the tumor mask. The last step was taken since it was observed that most of the time a smaller area than the actual tumor area was being segmented.

For the second case, when a cropped image was used as the region of interest, the segmentation process was very similar to when the entire liver was used. First, the liver tissue in the cropped image was segmented, then the liver range was deleted, and then the tumor was segmented.

A problem that could occur when using thresholding and connected components to keep the second largest area is that the area that is being saved does not correspond to the tumor. As an added step to avoid this problem a user input option was introduced to the workflow of the method. This step was added after the connected components had labelled all areas. In an image displayed to the user, all the labelled areas were displayed next to the CT scan, and assuming that the user knows approximately where the tumor is located, the user can select which area to keep. After the

selection, this area was kept as the tumor and the same steps as before were then performed in the same way. With the user input option, some more morphological operations were performed to be able to distinguish the tumor area. This difference could mean that in the cases where ordinary thresholding works well, the user input option is not the best, and vice versa. User input was a repeated addition to most of the following tumor segmentation methods.

3.2.3 Multi Otsu

The segmentation algorithm based on Multi Otsu thresholding is very similar to the one based on global thresholding, see Subsection 3.2.2. The input to the algorithm was either the whole image or a cropped image. The images were obtained and preprocessed in the same way as in the case of thresholding. The liver segmentation was also performed in the exact same way in both methods. What mainly separates thresholding from Multi Otsu thresholding is the tumor segmentation part.

The first thing that happens in the tumor segmentation is that the image is smoothed with an averaging filter. Then two optimal thresholds were found. These could be used to divide the pixels into three classes, with maximal separability between them. The class that contains the least amount of pixels was then kept, as that should correspond to the tumor. The other two classes should correspond to the liver tissue and the background. Next, the morphological operation erosion was used to separate joined areas in the remaining class. Then connected components were used to select the second largest area that should correspond to the tumor. Then dilation is performed and any holes in the mask were filled, creating the final tumor mask. The addition of a user input feature was added to the Multi Otsu methods too. As before, the user was able to select which area was being kept after connected components, both for the segmented liver and the cropped image as input.

3.3 Active Contour Model Segmentation

The input to the active contour models was similar to the one for the thresholding algorithms. The images were preprocessed in the same way, and then a region of interest was picked around the tumor. For the Basic Snake method, the region was an initial snake around the tumor, and for the Morphological Snake model, the same cropped images were used as for the thresholding methods.

3.3.1 Basic Snake

The Basic Snake algorithm demands user input to work, including an initial snake, but no other region of interest. Therefore, the Basic Snake method had only one approach since cropping the image or segmenting out the liver would not have any effect. All other preprocessing steps were still performed on the images used for Basic Snake. First, the initial snake was developed as a circle around the tumor, and centred based on the label. The centre point could be shifted to place the initial

snake with some offset from the tumor. After the initialization of the first snake, the parameters α and β were experimented with to achieve different results. Before evaluation, α and β were chosen based on observations. Then the Basic Snake algorithm presented in Section 2.4 was implemented.

3.3.2 Morphological Snake

Unlike the Basic Snake algorithm, Morphological Snake does not require user input. Hence, before applying the Morphological Snake algorithm to the image, preprocessing was performed. This included cropping the image to include the tumor. After the preprocessing, the Morphological Snake operation was applied to perform a set number of iterations. This step was also applied after first performing some thresholding. This thresholding was performed by calculating the average pixel value (excluding all zero-value pixels) and deleting all pixels under this value. Adding this step to the Morphological Snake resulted in two different methods: Morphological Snake with and without thresholding.

The morphological algorithm applied to the images is called Active Contours Without Edges (ACWE), MorphACWE for short. The motivation behind using MorphAWCE over other alternatives was that this method does not require well-defined borders but instead detects differences in pixel values for different regions [54]. When the algorithm detected edges in the image, connected components were used to pick out the largest area. Then the area was filled in as in previous methods. The last step for the Morphological Snake method was to once again add the user input feature, making the user able to select the area, after the Morphological Snake operation, to keep.

3.4 Deep Learning Segmentation

The U-Net architecture was chosen to perform the tumor segmentation since it is a common architecture used for biomedical image segmentation. As stated in Section 1.3, the focus in this project was on tumor segmentation rather than liver segmentation, and therefore only one U-Net with the purpose of segmenting tumors was trained. Furthermore, the input to the network was in 2D, which was also stated in Section 1.3. The U-Net model was trained using two different types of input. The first one was an entire image with the liver segmented in the same way as described in Subsection 3.2.1. The second one was cropped input images where a region of interest around the tumor had been saved. In this case, a small square was selected around the tumor, similar to in Subsection 3.2.1, but in this case, the positioning of the square around the tumor was randomized. To ensure that all tumor pixels remained within the frame, the square was positioned accordingly. However, the tumor's location within the square was deliberately varied to prevent the neural network from learning that the tumor is consistently situated in the centre of the image.

Since training with 2D input, only images with tumor was kept to reduce training time. This resulted in 6367 images from the 120 CT volumes. Considering the size

of the data, it was decided to split it into 80/10/10 % for the train/validation/test sets. To avoid creating a biased model, data from the same patients were not split between any datasets. Furthermore, the decision of which patients ended up in which dataset was randomized. The training and validation dataset was used for training and continuous evaluation purposes, whereas the network did not see the test dataset before the final evaluation described in Subsection 3.5.3. More information about the division of data is presented in Table 3.1.

Table 3.1: Overview of the division of data into training, validation and test sets.

Dataset	# patients	# images
Training	101	4976
Validation	8	695
Test	11	696

3.4.1 The Training Loop

To proceed with training, a network needs to be defined and the datasets created. The desired network was the U-Net described in Subsection 2.5.2. A class was defined inheriting from a machine learning framework. To create the datasets, a class was defined. Three datasets were created, one for training, one for validation, and one for testing. The test dataset was never seen by the network before the evaluation part. When creating the datasets some transformations were applied to the data. The image size was reduced to a set number of pixels for all liver-segmented input images and another set number of pixels for all cropped input images, in order to save training time. During training and validation (not test) random flip transformations were also applied, both horizontally and vertically.

After the datasets had been created, dataloaders were created. The dataloaders were used as input to the network. One dataloader was created from the training set and one from the validation set. In the dataloaders, the batch size, described in Subsection 2.5.1, was specified. First, a batch size was decided. After that, the U-Net model was defined by using the U-Net class. Next, the optimizer was chosen to be Adam and a learning rate was selected, based on the research from prior work presented in Section 1.4. The loss function was then chosen to be the dice loss, described in Subsection 2.7.1. The dice loss function was created as a customized loss function with a new forward function that computes the dice loss. Two variations of the dice loss function were created, a one-channel dice loss and a two-channel dice loss, both described in Subsection 2.7.1.

To use the dataloaders to train the network, a training function was defined that takes the dataloaders, the model, the loss function, and the optimizer with the learning rate and trains the model for a given number of epochs. The validation dataloader was used to validate the model during training. To determine when training was finished and had reached the desired performance, the progress was visualized during training. The visualization consisted of prints and graphs of the

mean training and validation loss for each epoch, computed as a mean of the losses for all mini-batches.

3.4.2 Training Schedule

The performance of a trained neural network depends, among other things, on the choice of model, optimizer, and loss function, as described in Section 2.5.1. To limit the scope of the project, the U-Net model was trained with the Adam optimizer on two different kinds of input and two variations of the dice loss function. The inputs used were liver-segmented images and cropped images where only a square region of interest was kept. The reason for using both the one-channel and the two-channel dice loss was that there existed doubts about whether the class imbalance between the tumor pixels and the background pixels in the liver-segmented images was too big to use a two-channel loss function. The four combinations of inputs and loss functions are presented in Table 3.2.

Table 3.2: The four different combinations of loss functions and inputs used during training.

Combination
1: Two-channel dice loss + Cropped input
2: One-channel dice loss + Cropped input
3: Two-channel dice loss + Liver segmented input
4: One-channel dice loss + Liver segmented input

Furthermore, it was chosen to only investigate the performance of the U-Net model with respect to the learning rate, the number of epochs, and the use of data augmentations during training. Thus, there remain great opportunities to experiment with parameters such as batch size, image size, the utilization of early stopping and dropout, the choice of optimizer, and other loss functions in the future. The tested values of the explored parameters are summarized in Table 3.3, and the seven steps in the training scheme are presented in the following list.

Table 3.3: Different parameters and their values explored during training.

Parameter	**Variations**
Learning rates	`A`, `B`
Data augmentations	0, 2, 4
Number of epochs	`A`, `B`

1. Learning rate = `A`, no augmentations, trained for `A` number of epochs.
2. Learning rate = `B`, no augmentations, trained for `A` number of epochs.
3. The best learning rate from 1 and 2, 2 augmentations, trained for `A` number of epochs.
4. The best learning rate from 1 and 2, 4 augmentations, trained for `A` number of epochs.

5. The best learning rate from 1 and 2, no augmentations, trained for `B` number of epochs.
6. The best learning rate from 1 and 2, 2 augmentations, trained for `B` number of epochs.
7. The best learning rate from 1 and 2, 4 augmentations, trained for `B` number of epochs.

In steps 1 and 2, the effect of the training using two different learning rates (`A` and `B`) was tested. The learning rate that resulted in the best behaviour of the training was chosen as the standard learning rate for the following models. The best behaviour was decided based on several factors, more particularly the training and validation loss, the level of overfitting, and the stability of the loss over the epochs.

In steps 3 and 4, the best learning rate and the number of epochs used in steps 1 and 2 were kept, but data augmentation was used to increase the amount of training data and, hopefully, to decrease the level of overfitting. Four different combinations of transformations were used, which are presented in Table 3.4. The amount of each transformation (e.g. rotation) applied to each sample (both the image and the label) in the dataset were randomized from a given range. The range for each transformation was decided by inspecting the effect of applying the extreme values of the transformation on multiple images since too great a difference between the original and the augmented image needed to be avoided. "No augmentations" means that only the original augmentations horizontal and vertical flip was used during training and thus that the training dataset had its original size (4976 samples). "2 augmentations" means that the first and the second combination of transformations in Table 3.4 was used to increase the size of the dataset. A concatenated dataset was created which consisted of the original 4976 samples, 4976 samples augmented with the first combination of transforms, and 4976 samples augmented with the second combination of transforms. The total size of the dataset thus became 14928 samples, three times the original size. Finally, "4 augmentations" means that all combinations of transformations in Table 3.4 were used to increase the size of the dataset. The concatenated dataset thus consisted of 4976 samples and 4976 additional ones for each combination of transforms. The total size of the dataset thus became 24880 samples, five times the original size.

Table 3.4: Four different combinations of transformations used for data augmentation.

Transformations
1: Rotation, Scaling, Gaussian blur
2: Rotation, Translation, Shear, Brightness
3: Rotation, Scaling, Shear, Gaussian blur
4: Scaling, Translation, Shear, Gaussian blur, Brightness

In steps 5 to 7, the number of epochs was increased to `B` number of epochs, the learning rate was kept constant and each augmentation case was tested.

This scheme was repeated for all four combinations of input and loss functions shown in Table 3.2. Since the combination of the two-channel dice loss and the liver-segmented input images resulted in failure, only 21 successful models were trained.

3.5 Evaluation

Because of the significant difference between the methods applied for tumor segmentation, the evaluation had to be divided into different stages for the methods. Below are the evaluation methods for the algorithm part, the deep learning part, and the comparison presented.

3.5.1 Thresholding

To evaluate the thresholding methods, one image was randomly selected from each of the 107 patients containing at least one tumor in the LiTS dataset. This resulted in 107 images that could be used for evaluation. Since the methods were designed to only segment out one tumor, all the associated tumor labels were modified only to contain one tumor, where the largest one was kept. Hence, the evaluation is a measure of the performance of segmenting out the only tumor or the largest tumor for each scan.

The first step of the evaluation was to measure, out of the 107 images, in how many cases the methods segmented out at least one true positive tumor pixel. This metric is referred to as ”Positive Cases” in the result chapter. When no tumor pixel is correctly segmented, the DSC and recall value will be zero. For all of the 107 images, DSC and recall were calculated as a mean for the entire set of 107 images. For global thresholding, the evaluation was performed for four different methods. With the entire liver as the region of interest with and without user input, and with the cropped region of interest with and without user input. For Multi Otsu thresholding, the same four different methods were evaluated using the same method.

3.5.2 Active Contour Models

The same 107 images were used for the evaluation of the active contour models. The snakes obtained were converted into labels and the DSC and recall value were calculated. For the Basic Snake model, three different initial snake offsets were evaluated. For Morphological Snake, four different methods were evaluated: with and without thresholding first, as well as with and without the user input feature. Since the same dataset, the 107 images, was used for the evaluation of all the methods, the result could be compared in a correct way.

3.5.3 Deep Learning

For the evaluation of the U-Net models, the 21 different U-Net models were tested using the test set of 696 images from 11 patients, described in Section 3.4. Two versions of this test set were used, one with cropped region-of-interest images for the networks trained on cropped input and one with liver-segmented images for the networks trained on liver-segmented input. When a dataset was created for the liver-segmented test images, as described in Subsection 3.4.1, the resize transformation caused all existing tumor pixels to vanish in three of the original images. These three images originally contained one or very few scattered pixels. When computing the DSC this resulted in division by zero and undefined values, since the DSC does not contain a smoothing term (as the ones in the nominator and denominator of the dice loss). To avoid this problem these three images were removed from the test set, which means that the networks trained on liver-segmented input were evaluated on 693 images instead of 696 images. During the evaluation of a network, the test images were fed to the network, which outputted a prediction. Then the predicted tumor segmentation was evaluated by using the DSC and recall value.

3.5.4 Comparison

To be able to compare the classical algorithm methods to the performance of the U-Nets, they needed to be evaluated on the same data. Since the U-Net models are biased toward the data used during training, 11 images were selected, one from each patient in the test set. To be able to compare the methods, the images were selected to only contain one tumor and hence no modification had to be implemented to the labels. The performance of the different methods on this small dataset of 11 images made it possible to compare them, at least their performance on this particular dataset.

4
Results

In this chapter, the results of the thesis work will be presented. Firstly, the performance of the different segmentation algorithms; Global thresholding, Multi Otsu thresholding, Basic Snake, and Morphological Snake will be presented. Secondly, the performance of the U-Net models will be presented. Lastly, the results of the segmentation algorithms and the best-performing U-Net models on the 11 comparison images will be presented.

4.1 Liver Segmentation

The first part of the segmentation was to segment out the liver, as described in Subsection 3.2.2. After evaluating the liver segmentation on the 107 images, it could be determined that in only 72 cases, more than 60 % of the liver was segmented out correctly. It was decided that this would have too large of an effect on the tumor segmentation evaluation, which would be the next step after liver segmentation, and hence could not be used without affecting the results. From this, it was decided to use the liver label for each image to segment out the liver. This was done for the global thresholding and Multi Otsu methods as well as the input to the U-Net models.

4.2 Thresholding

First, the different thresholding methods, global thresholding and Multi Otsu thresholding, were evaluated on 107 images.

4.2.1 Global Thresholding

Performing tumor segmentation using global thresholding was evaluated using the 107 images with four different methods. First, the entire liver was used as input, and thresholding was performed according to the method described in Subsection 3.2.2. Then, the feature of user input was added to the method, and the 107 images were again used as input for evaluation. For the last two methods, the liver images were cropped with a region of interest around the tumor and were again used with and without user input for evaluation of the thresholding method. In the following tables, a ”positive case” is defined as an image slice where the model correctly classifies at least one tumor pixel. The mean DSC is computed as the mean of the individual DSC values obtained for all 107 test images. The mean recall is computed in the

same way, more particularly as the mean of the recall values obtained for the same 107 images. An overview of the results from the four different global thresholding methods can be observed in Table 4.1. Here it can be noticed that *Thresholding Cropped with User Input* achieved the highest mean DSC, and *Thresholding Cropped* achieved the highest mean recall value.

Table 4.1: Overview of the results for the different thresholding methods. "Positive Cases" points out in how many cases (of the 107 images) at least one pixel was correctly classified as tumor, shown in percentage. The "Mean DSC" shows a mean DSC of all 107 images. The "Mean Recall" value is also the mean value for all 107 images. The best results in each category are highlighted in bold font.

Method	Positive Cases	Mean DSC	Mean Recall
Thresholding	48.6 %	0.257	0.302
Thresholding, User Input	46.7 %	0.245	0.228
Thresholding Cropped	72.0 %	0.384	**0.605**
Thresholding Cropped, User input	**80.4 %**	**0.405**	0.489

4.2.2 Multi Otsu

For the Multi Otsu algorithm, the same four methods as for global thresholding were evaluated. First, with the entire liver as input, with and without user input, and then, with cropped input images, with and without user input. The results can be observed in Table 4.2. Here it can be observed that *Multi Otsu Cropped with User Input* achieved the highest mean DSC, and *Multi Otsu Cropped* achieved the highest mean recall value.

Table 4.2: Overview of the results for four different Multi Otsu methods.

Method	Positive Cases	Mean DSC	Mean Recall
Multi Otsu	49.5 %	0.346	0.364
Multi Otsu, User Input	46.7 %	0.311	0.358
Multi Otsu Cropped	**67.3 %**	0.408	**0.520**
Multi Otsu Cropped, User Input	**67.3 %**	**0.435**	0.517

4.3 Active Contour Models

Secondly, the active contour models Basic Snake and Morphological Snake were evaluated on the same 107 images.

4.3.1 Basic Snake

The Basic Snake method was evaluated in three different ways, with the initial snake being centred around the tumor according to the label, and offset by either `A` or `B`

% of the tumor diameter. The parameters for the algorithms were α and β. The results can be observed in Table 4.3. Here it can be observed that the method with the centred initial snake performed best in all categories.

Table 4.3: Overview of the results for the Basic Snake algorithm with the different initial snakes.

Method	Positive Cases	Mean DSC	Mean Recall
Basic Snake, Centered Initial Snake	**83.2 %**	**0.393**	**0.644**
Basic Snake, `A` % Offset Initial Snake	45.8 %	0.134	0.188
Basic Snake, `B` % Offset Initial Snake	22.4 %	0.017	0.017

4.3.2 Morphological Snake

The Morphological Snake algorithm was evaluated using four methods. First, the algorithm was used on the entire liver-segmented image, where a region of interest was selected for each image. The Morphological Snake algorithms were then performed for a set number of iterations. The same method was also used with the added feature user input. For the last two methods, the liver-segmented images were thresholded before the morphological operations were performed, both with and without user input. The results for the four different methods can be observed in Table 4.4. Here it can be observed that *Morphological Snake with Thresholding and User Input* achieved the highest mean DSC as well as the highest mean recall value.

Table 4.4: Overview of the results for the different Morphological Snake methods.

Method	Positive Cases	Mean DSC	Mean Recall
Morph Snake	70.1 %	0.104	0.554
Morph Snake, User Input	86.0 %	0.194	0.709
Morph Snake, Thresholding	65.4 %	0.198	0.477
Morph Snake, Thresholding & User Input	**95.3 %**	**0.340**	**0.820**

4.4 Deep Learning

The 21 U-Net models were evaluated on a test set consisting of 696 image slices from 11 patients, as described in Section 3.4. Below the percentage of positive cases, the mean DSC and the mean recall value are presented for each model. The mean values are calculated as the mean of the results obtained from the 696 test images.

The results for the seven models trained on the cropped images with the two-channel dice loss are presented in Table 4.5. Here it can be noted that the highest percentage of positive cases is obtained by *Model 4* (4 augmentations, `A` number of epochs, learning rate `A`), that the highest DSC is obtained by *Model 6* (2 augmentations, `B` number of epochs, learning rate `A`), even though the DSC for *Model 4* and *Model*

7 (4 augmentations, `B` number of epochs, learning rate `A`) is only negligibly lower, and finally that the highest recall is obtained by *Model 4* and *Model 7*. The results for the seven models trained on the cropped images with the one-channel dice loss are presented in Table 4.6. In this case, it can be observed that *Model 12* (no augmentations, `B` number of epochs, learning rate `A`) achieved the highest percentage of positive cases, whereas *Model 14* (4 augmentations, `B` number of epochs, learning rate `A`) achieved both the highest DSC and recall.

Table 4.5: Overview of the results for the U-Net models trained on the cropped images with the two-channel dice loss. "Positive Cases" points out in how many cases (of the 696 images) at least one pixel was correctly classified as tumor, shown in percentage. The "Mean DSC" shows a mean DSC of all 696 images. The "Mean Recall" shows the mean recall value for all 696 images. The information after the model name specifies the choice of training parameters; "aug" is the number of further augmentations used, "epochs" is the number of epochs the model is trained for and "lr" is the chosen learning rate. The best results in each category are highlighted in bold font.

Method	Positive Cases	Mean DSC	Mean Recall
Model 1: no aug, epochs = `A`, lr = `A`	99.0 %	0.738	0.745
Model 2: no aug, epochs = `A`, lr = `B`	97.8 %	0.706	0.701
Model 3: 2 aug, epochs = `A`, lr = `A`	98.3 %	0.736	0.735
Model 4: 4 aug, epochs = `A`, lr = `A`	**99.1 %**	0.757	**0.773**
Model 5: no aug, epochs = `B`, lr = `A`	98.0 %	0.750	0.740
Model 6: 2 aug, epochs = `B`, lr = `A`	98.3 %	**0.758**	0.766
Model 7: 4 aug, epochs = `B`, lr = `A`	98.9 %	0.757	**0.773**

Table 4.6: Overview of the results for the neural network trained on the cropped images with the one-channel dice loss.

Method	Positive Cases	Mean DSC	Mean Recall
Model 8: no aug, epochs = `A`, lr = `A`	98.9 %	0.751	0.782
Model 9: no aug, epochs = `A`, lr = `B`	98.1 %	0.722	0.712
Model 10: 2 aug, epochs = `A`, lr = `A`	97.8 %	0.755	0.769
Model 11: 4 aug, epochs = `A`, lr = `A`	98.6 %	0.758	0.784
Model 12: no aug, epochs = `B`, lr = `A`	**99.1 %**	0.751	0.758
Model 13: 2 aug, epochs = `B`, lr = `A`	98.9 %	0.759	0.780
Model 14: 4 aug, epochs = `B`, lr = `A`	99.0 %	**0.766**	**0.796**

The results for the seven models trained on the liver segmented images with the one-channel dice loss are presented in Table 4.7. Here it can be seen that the highest percentage of positive cases is detected by *Model 18* (4 augmentations, `A` number of epochs, learning rate `A`), the highest DSC is obtained by *Model 21* (4 augmentations, `B` number of epochs, learning rate `A`) and the highest recall by *Model 20* (2 augmentations, `B` number of epochs, learning rate `A`). No results for the model trained on the liver-segmented images with the two-channel dice loss are presented

since the training was unsatisfactory, and the trained model always classified all pixels as background.

Table 4.7: Overview of the results for the neural network trained on the liver segmented images with the one-channel dice loss.

Method	Positive Cases	Mean DSC	Mean Recall
Model 15: no aug, epochs = `A`, lr = `A`	85.4 %	0.643	0.647
Model 16: no aug, epochs = `A`, lr = `B`	89.2 %	0.667	0.711
Model 17: 2 aug, epochs = `A`, lr = `A`	84.6 %	0.638	0.610
Model 18: 4 aug, epochs = `A`, lr = `A`	**92.8 %**	0.692	0.748
Model 19: no aug, epochs = `B`, lr = `A`	89.5 %	0.681	0.710
Model 20: 2 aug, epochs = `B`, lr = `A`	92.5 %	0.689	**0.767**
Model 21: 4 aug, epochs = `B`, lr = `A`	92.2 %	**0.696**	0.729

4.5 Comparison

To compare the U-Net models and the algorithm methods, a small dataset of 11 images was selected for evaluation of all different methods. In Table 4.8, the results can be observed. Results for all 21 U-Net models are not presented, instead, the two models that achieve the highest DSC in each setup shown in Tables 4.5 to 4.7 are selected. In the comparison table, Table 4.8, it can be seen that U-Net *Model 4* achieves the highest mean DSC and that U-Net *Model 14* achieves the highest mean recall value. The highest percentage of positive cases is obtained by all U-Net models with cropped images as input, all of them find at least one tumor pixel in all of the test images.

To better visualize the performance of the different methods, two test images are used to compare the tumor segmentation with the ground truth. The images are picked from the 11 images selected for evaluation and comparison purposes. In the images, the border of the tumor segmentation is shown as a yellow line, and the border of the ground truth is a red line. Note that these images have been created for visualization purposes and do not give a comprehensive view of the performance of the different methods. For a more accurate view of the result, see Tables 4.1 to 4.8, and the discussion Chapter. From now on, the two images will be referred to as scan 1 and scan 2.

For all methods using cropped images, the scans have been cropped prior to the evaluation, and for the other methods, the liver has been segmented out from the original images. In Figure 4.1 scan 1 can be observed and the performance of the methods global thresholding (with and without user input), Multi Otsu (with and without user input), and Basic Snake with the initial snake being centred and with an offset of `B` % of the tumor diameter is visualized. In Figure 4.2 the same methods can be observed for scan 2.

Table 4.8: Overview of the results for the 11 images selected for evaluation purposes. The best results in each category are highlighted in bold font. The information following the model name for the U-Net models states the number of augmentations used, the number of epochs the model has been trained for, the used loss function (one-channel or two-channel dice loss) and whether or not the input images are cropped. All models were trained with a learning rate of `A`.

Method	**Positive Cases**	**Mean DSC**	**Mean Recall**
Thresholding	54.5 %	0.226	0.267
Thresholding, User Input	36.4 %	0.170	0.166
Thresholding Crop	72.7 %	0.294	0.551
Thresholding Crop, User Input	45.5 %	0.277	0.324
Multi Otsu	54.5 %	0.308	0.481
Multi Otsu, User Input	81.8 %	0.458	0.654
Multi Otsu Crop	54.5 %	0.329	0.492
Multi Otsu Crop, User Input	45.4 %	0.227	0.336
Basic Snake	90.9 %	0.458	0.624
Basic Snake, Offset `A` %	36.4 %	0.108	0.079
Basic Snake, Offset `B` %	18.2 %	0.012	0.006
Morph Snake	72.7 %	0.215	0.624
Morph Snake, User Input	72.7 %	0.209	0.616
Morph Snake, Thresholding	63.6 %	0.239	0.484
Morph Snake, Thresholding and User Input	90.9 %	0.314	0.763
Model 4: 4 aug, epochs = `A`, 2-chan dice, crop	**100 %**	**0.723**	0.858
Model 6: 2 aug, epochs = `B`, 2-chan dice, crop	**100 %**	0.703	0.855
Model 13: 2 aug, epochs = `B`, 1-chan dice, crop	**100 %**	0.721	0.883
Model 14: 4 aug, epochs = `B`, 1-chan dice, crop	**100 %**	0.699	**0.890**
Model 18: 4 aug, epochs = `A`, 1-chan dice	72.7 %	0.415	0.530
Model 21: 4 aug, epochs = `B`, 1-chan dice	72.7 %	0.445	0.548

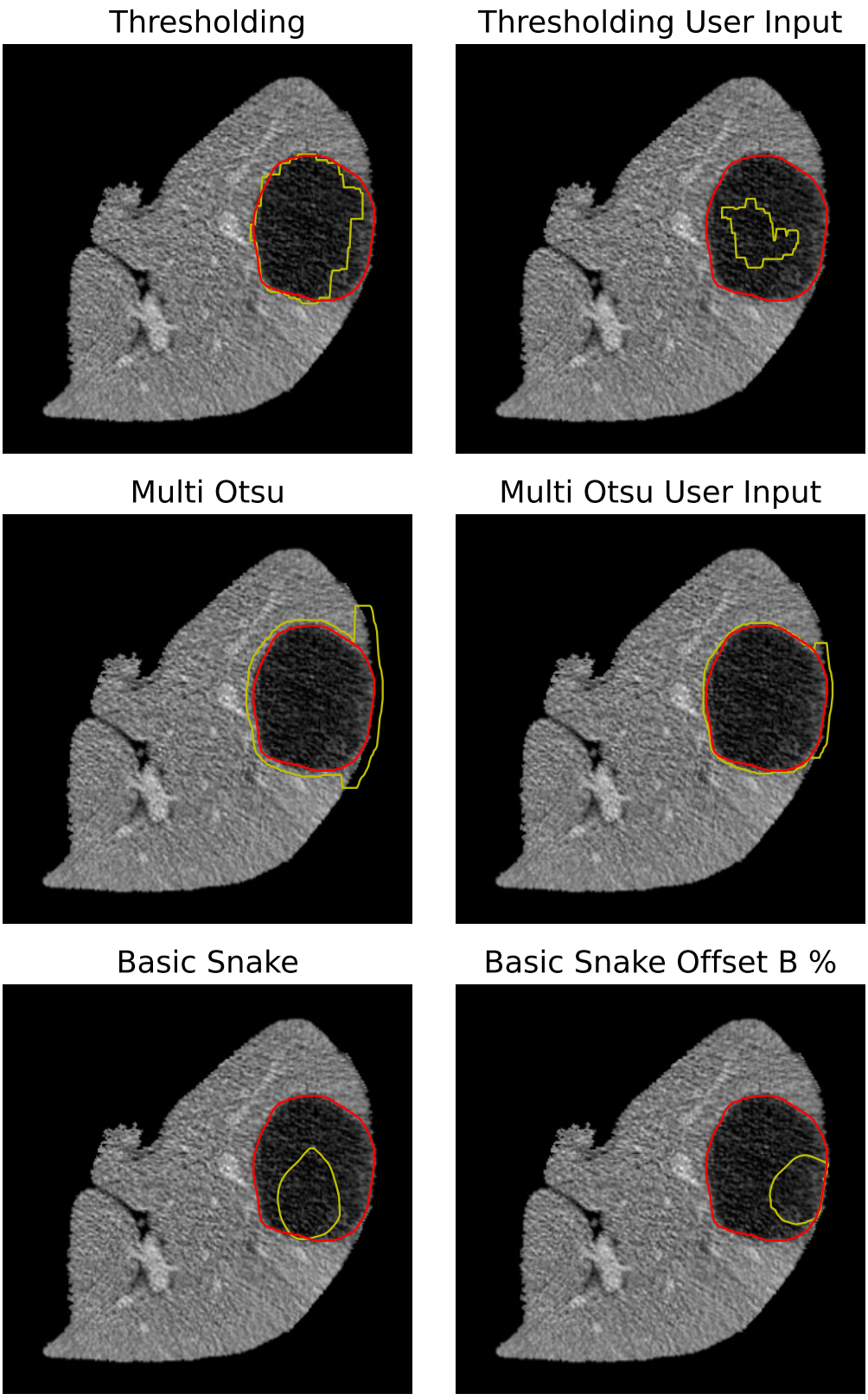

Figure 4.1: Scan 1 and the performance of thresholding, Multi Otsu and basic snake. The red outline is the ground truth, and the yellow outline is the prediction of the method.

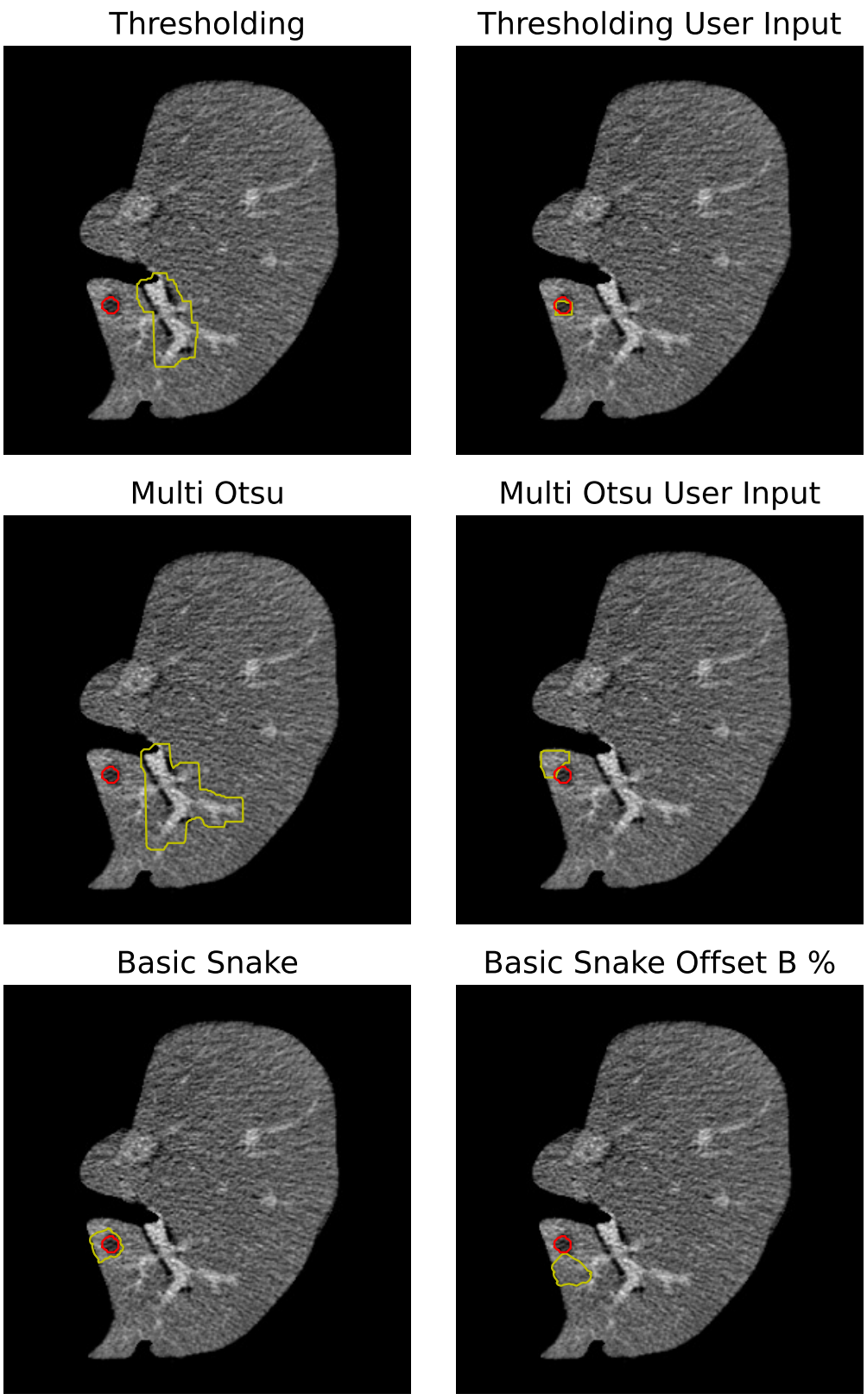

Figure 4.2: Scan 2 and the performance of thresholding, Multi Otsu and basic snake. The red outline is the ground truth, and the yellow outline is the prediction of the method.

In Figures 4.3 and 4.4, the performance of the cropped global thresholding and Multi Otsu methods can be observed on scans 1 and 2, respectively. It can be noted that the global thresholding method on scans 1 and 2 does not find the tumor, and hence the segmentation is somewhere else in the image. The user input methods, both thresholding and Multi Otsu, do not find any relevant tumor area for scan 2 and hence do not show any yellow lines at all.

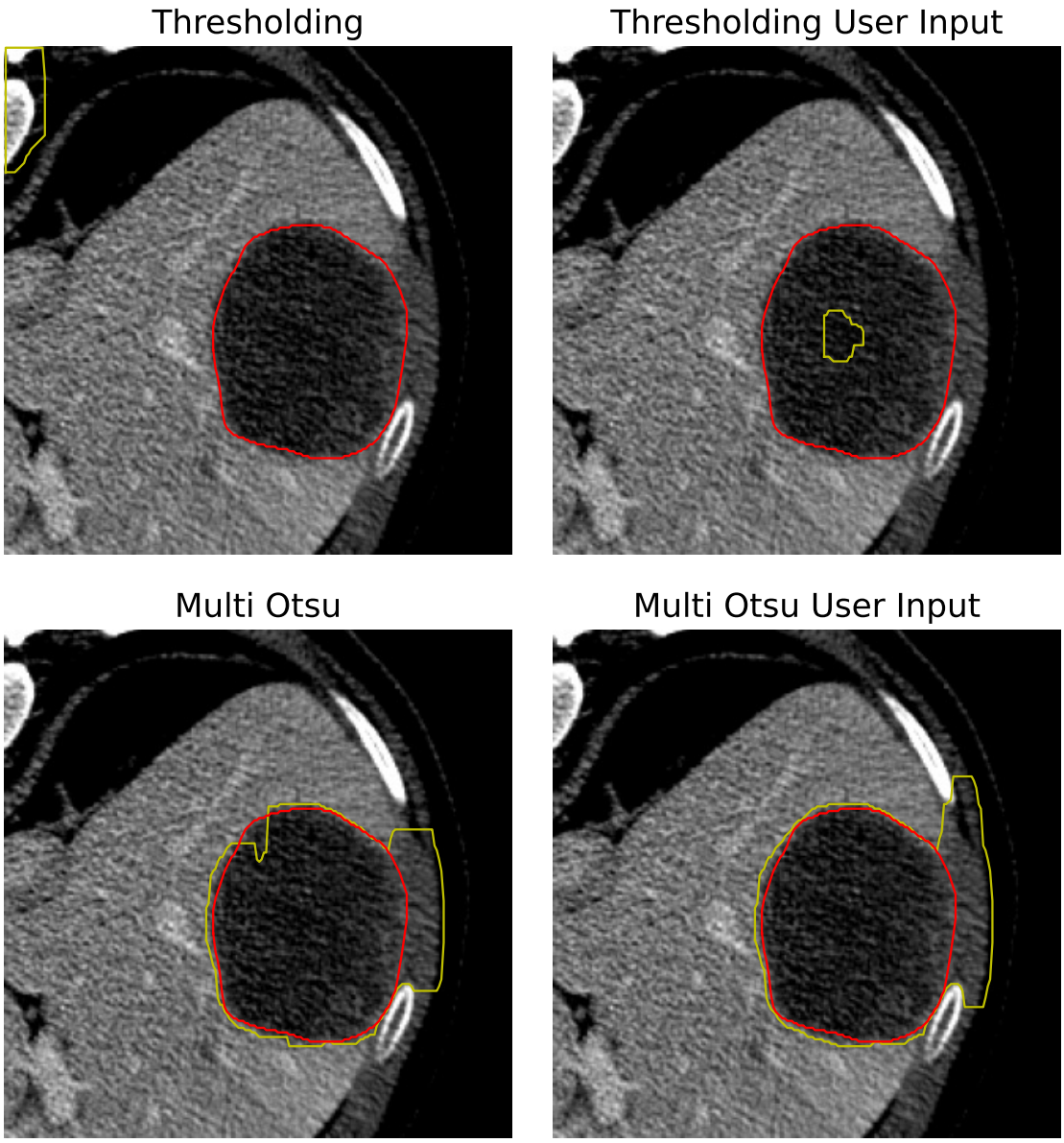

Figure 4.3: Scan 1 and the performance of cropped thresholding and cropped Multi Otsu. The red outline is the ground truth, and the yellow outline is the prediction of the method.

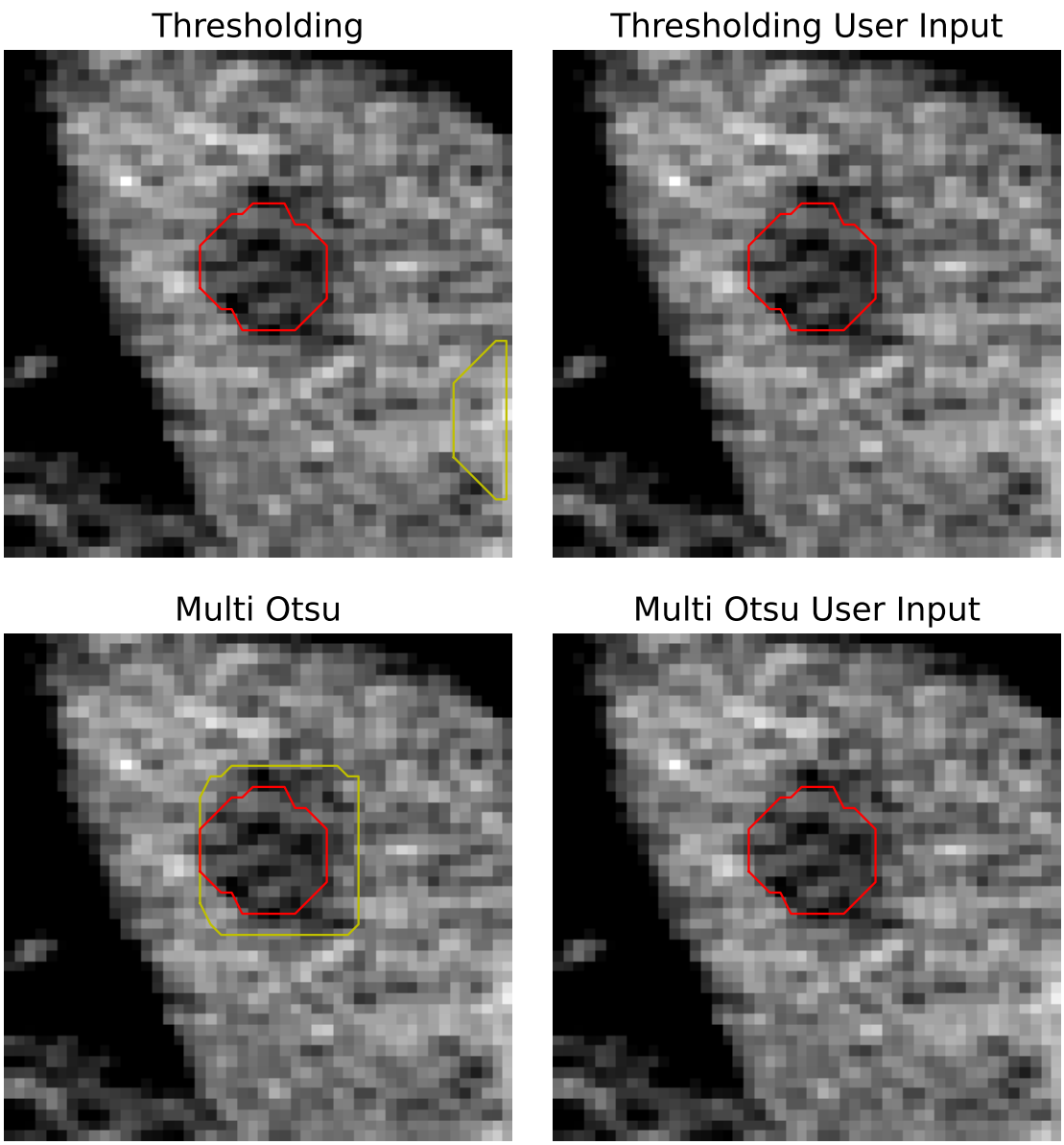

Figure 4.4: Scan 2 and the performance of cropped thresholding and cropped Multi Otsu. The red outline is the ground truth, and the yellow outline is the prediction of the method. Here it can be observed that the cropped thresholding and Multi Otsu with user input do not find any tumor area and hence do not display any yellow lines.

In Figures 4.5 and 4.6, the performance of the four different Morphological Snake methods are shown for scans 1 and 2, respectively. Here it can be observed, that for both scans 1 and 2, the methods that do not involve thresholding predict a larger area than any other method. This means that the entire tumor is still found, even though the segmentation does not predict the edges of the tumor.

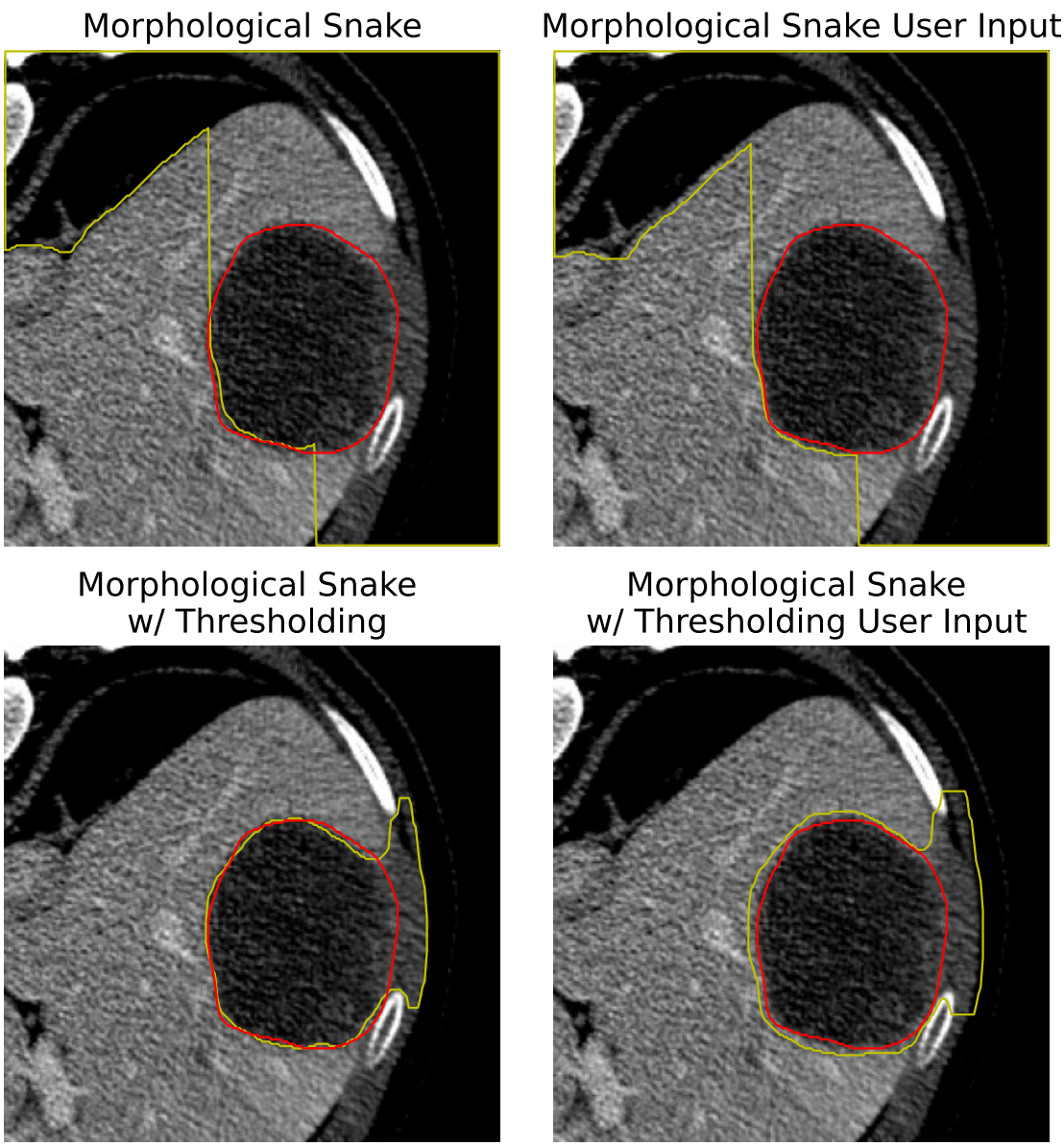

Figure 4.5: Scan 1 and the performance of the different morphological snake methods. The red outline is the ground truth, and the yellow outline is the prediction of the method.

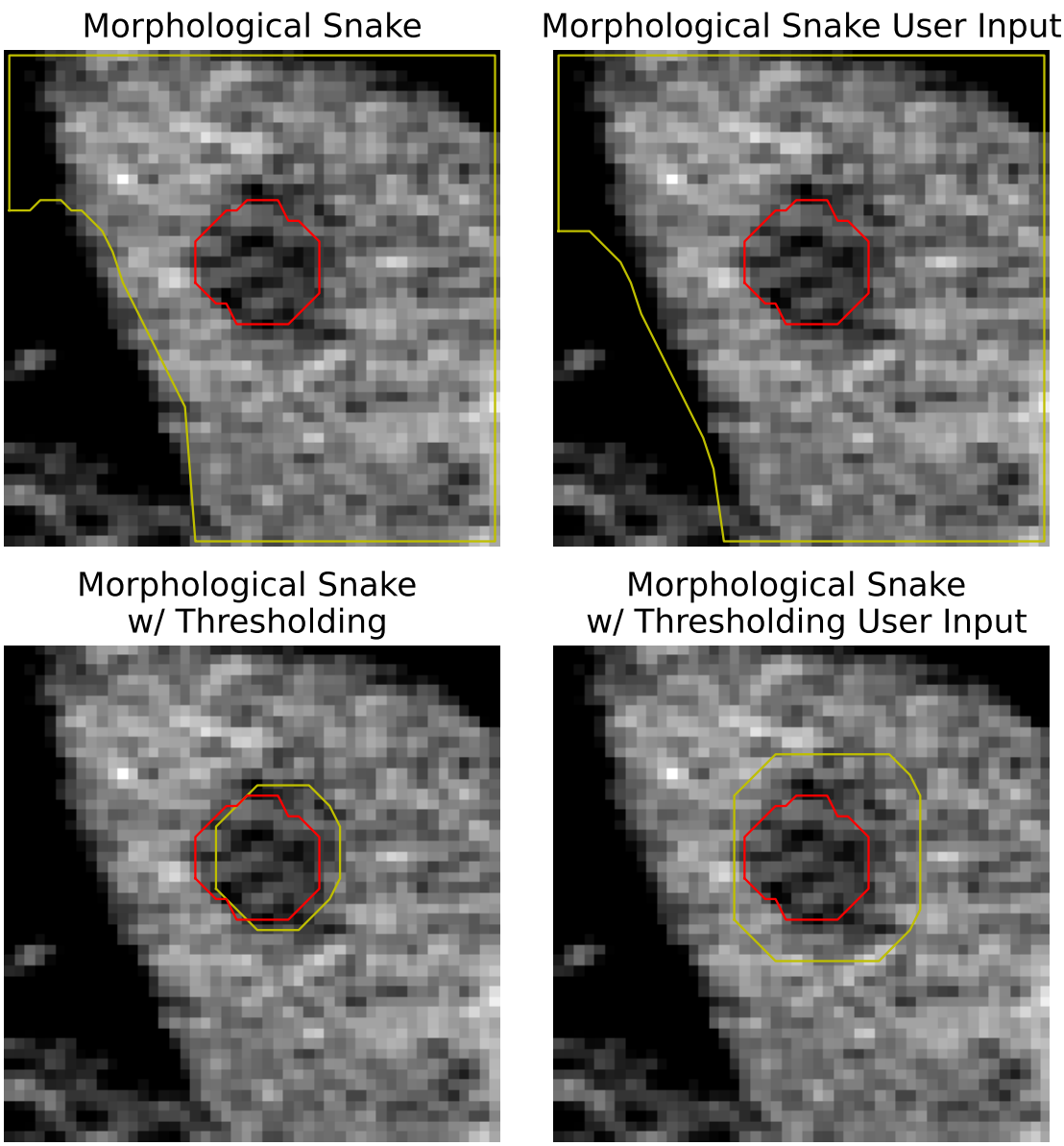

Figure 4.6: Scan 2 and the performance of the different morphological snake methods. The red outline is the ground truth, and the yellow outline is the prediction of the method.

Figures 4.7 and 4.8 show the performance of the two best-performing U-Net models, based on the mean DSC, trained on liver-segmented input data. Both models find the tumor and predict that there only exists one tumor in both scans.

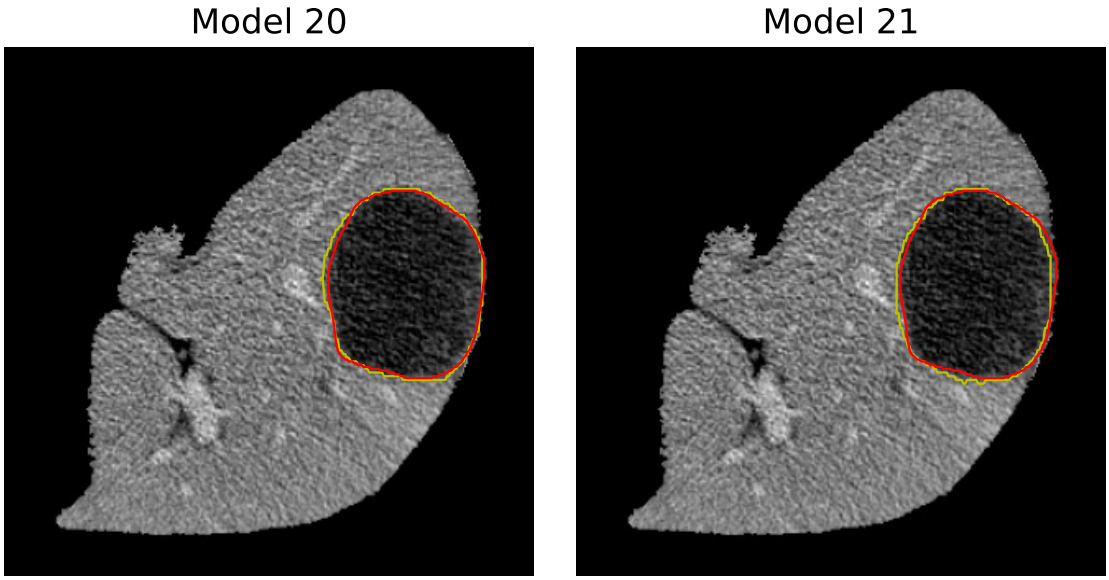

Figure 4.7: Scan 1 and the performance of the best U-Net models (based on DSC) trained on liver-segmented input images. The red outline is the ground truth, and the yellow outline is the prediction of the model.

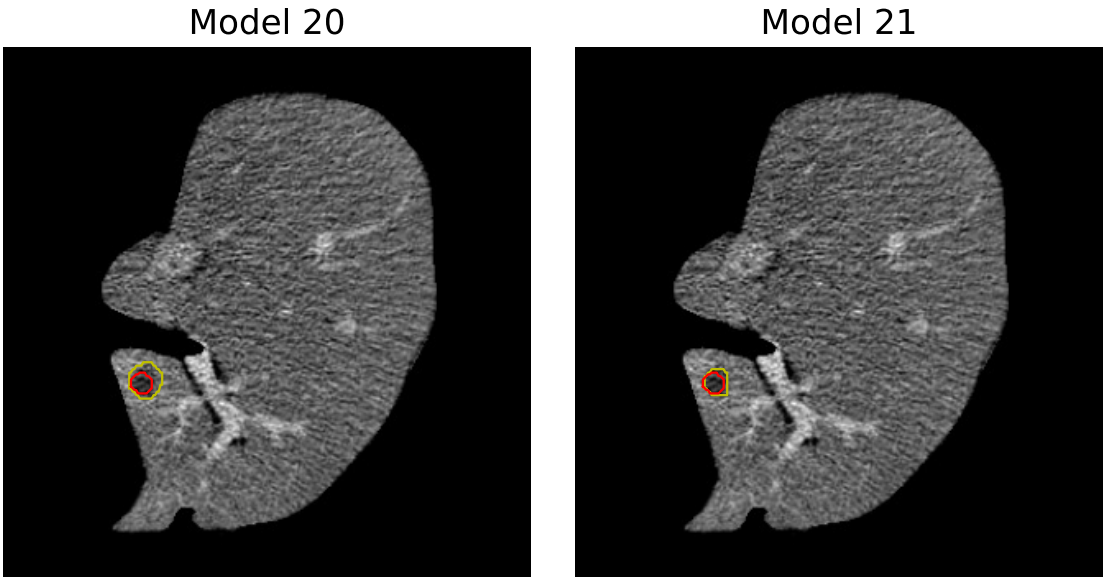

Figure 4.8: Scan 2 and the performance of the best U-Net models (based on DSC) trained on liver segmented input images. The red outline is the ground truth, and the yellow outline is the prediction of the model.

Figure 4.9 and 4.10 show the performance of the four best-performing U-Net models, based on the mean DSC, trained on cropped input data. All models find the tumor in both scans. Note that Model 4 and Model 14 predict that there exists more than one tumor in the scan in Figure 4.9.

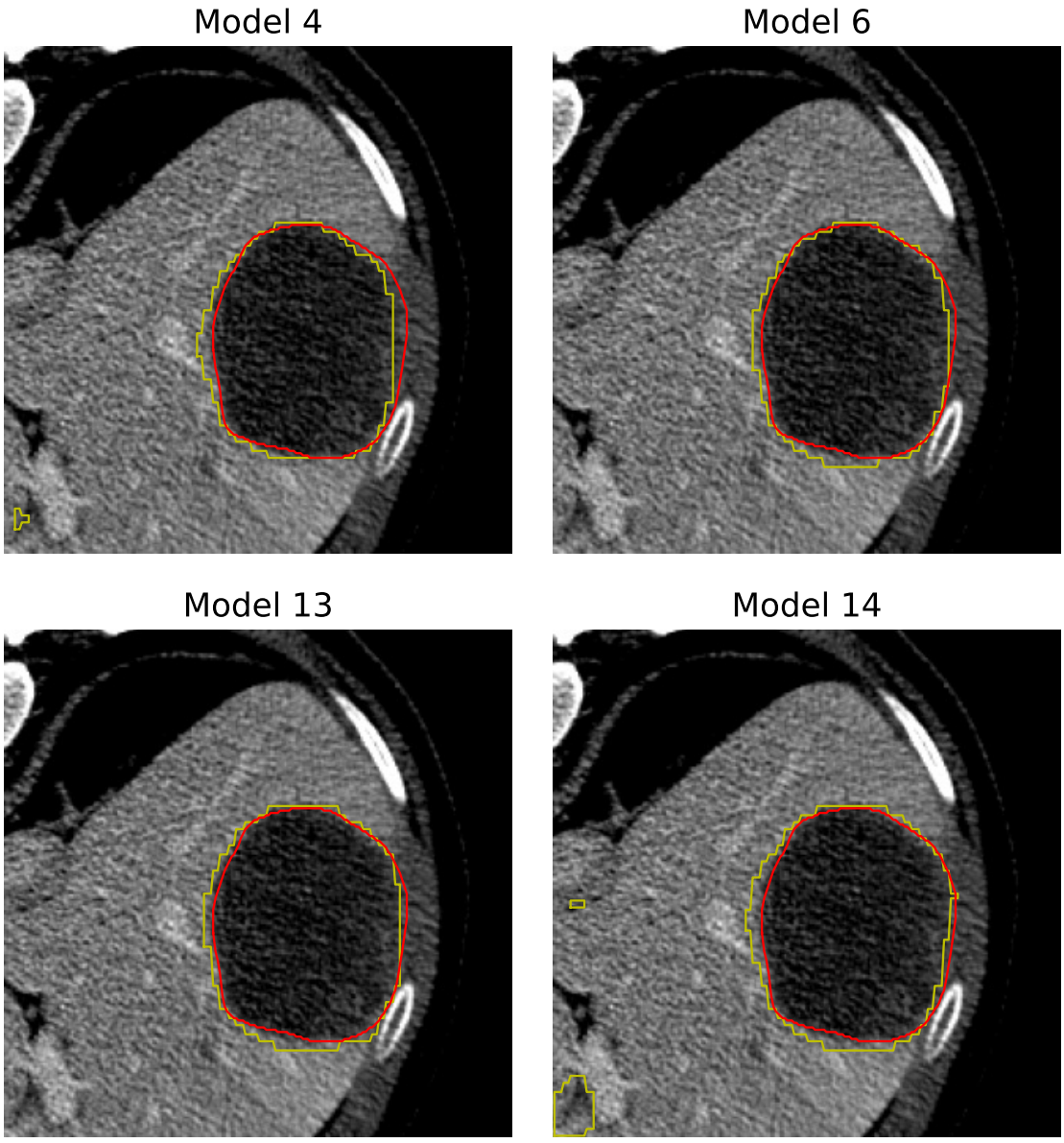

Figure 4.9: Scan 1 and the performance of the best U-Net models (based on DSC) trained on cropped input images. The red outline is the ground truth, and the yellow outline is the prediction of the model. Note that Model 4 and Model 14 predict that there exists more than one tumor in the scan.

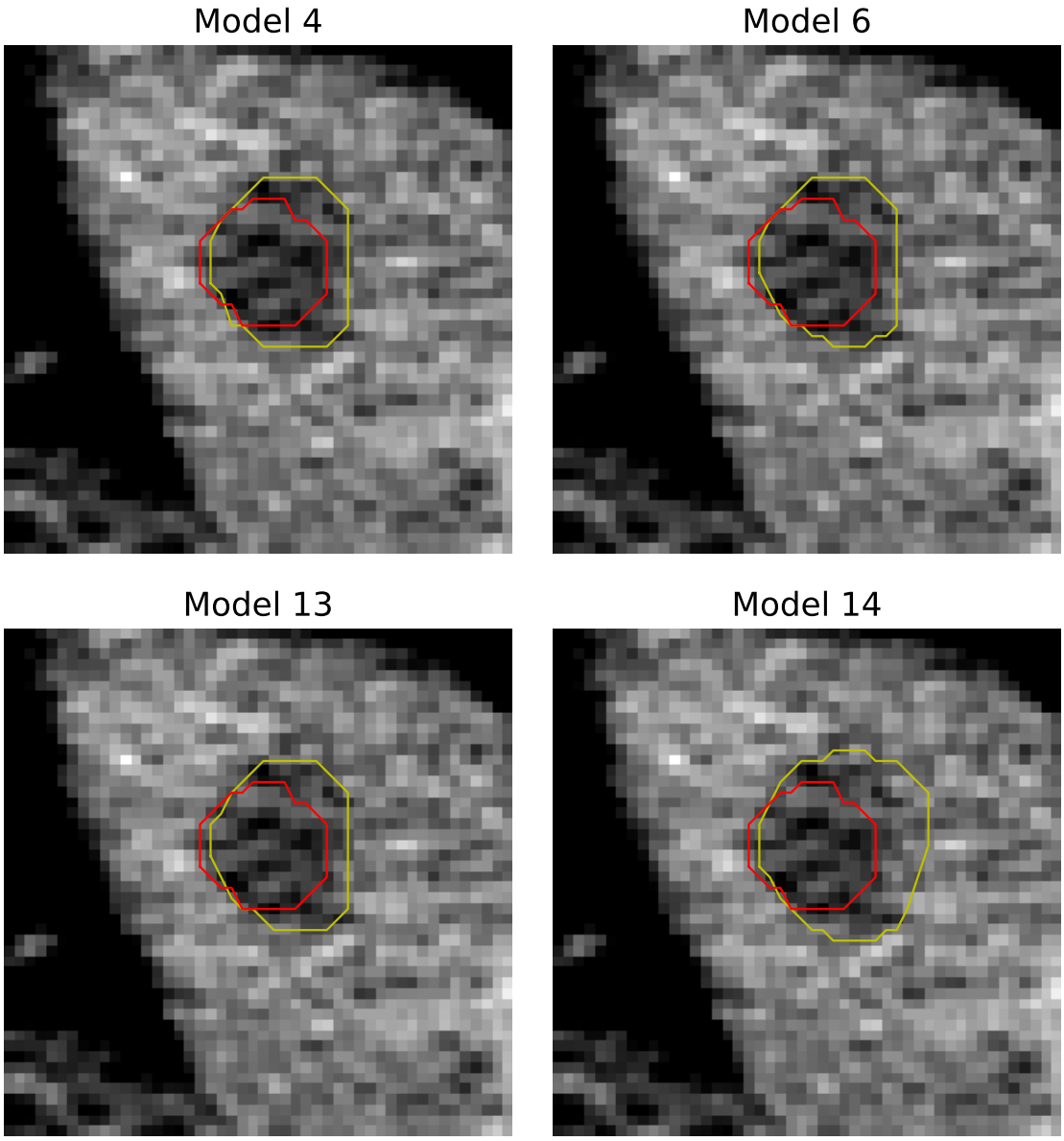

Figure 4.10: Scan 2 and the performance of the best U-Net models (based on DSC) trained on cropped input images. The red outline is the ground truth, and the yellow outline is the prediction of the model.

5
Discussion

In this chapter, the results presented in Chapter 4 will be analysed and discussed. First, it needs to be noted what is essential to look for in the results. Since working with medical images that contain tumors, the most important metric to look at is the mean recall value. This metric clarifies how much of the tumor tissue is identified. The mean DSC is a metric of overlap and a great addition to the mean recall value that provides more information about how accurate the segmentation is.

5.1 Thresholding Methods

For all the algorithms, thresholding and active contour models, it is assumed that there is only one tumor present in the liver. This is a problem since that is not the case for all patients in the used dataset. To be able to evaluate the performance, the labels needed to be altered to only contain one tumor, as described in Subsection 3.5.1. Because of this, the evaluation of the algorithms might not be completely accurate, since the possibility that another tumor is found by the algorithm, which does not correspond to the kept tumor label. However, since the methods still need to be evaluated, this course of action is used. Therefore it is encouraged to keep this problem in mind when reading the results and the discussion.

5.1.1 Global Thresholding

From the results presented in Table 4.1 about global thresholding, it can be observed that none of the thresholding methods manage to detect the tumor in all 107 images. The best-performing method in this regard is thresholding with a cropped image as input and added user input. For the mean DSC, this method performed best as well, with a score of 0.405, and for the mean recall value thresholding with cropped input and no user input performed the best. A mean recall value of 0.605 for 107 images is equivalent to about 60 % of all tumor pixels being correctly identified as tumor pixels. The result would need to be higher to have the potential to be used in the desired setting.

User input was added to the method to improve the results since observation showed that a common fault was that the tumor did not constitute the largest area after thresholding. Another common fault was that two areas were still connected after the morphological operations and therefore the area chosen as the tumor area was way too large. Hence, the user input function included some more iterations for

the morphological operations. This means that the method of user input might not work as well as the option without user input in some cases, and could instead be seen as a way to deal with the cases where the original method does not perform. Looking at thresholding with the entire liver as input, 52 images were positive cases and adding user input another 5 images were positive cases. For the cropped input, 77 images were positive cases and with the added user input 86 were positive cases, out of these 17 were new images. This means that in 8 images that were positive cases without user input, no tumor pixel was correctly classified with the user input method. This, in turn, means that if the two methods were to be combined, 94 images would have been classified as positive cases.

During the evaluation, the most common problem observed with global thresholding was that the pixel range selected to be deleted sometimes included the tumor intensities. This led to the tumor being thresholded away and the results being negative. An example of this can be seen in Figure 4.2 on the top left, where another structure in the image is being segmented instead of the tumor. The method of thresholding also assumes that the liver covers more of the region of interest than the tumor, which is not always the case for larger tumors. It would be desirable to keep the method automated while being able to adjust it to each individual image. Changing the thresholding value, or the range deleted, could yield different results and could hence be worth investigating.

5.1.2 Multi Otsu

From the results presented in Table 4.2 for the Multi Otsu methods, it can be observed that both of the methods with cropped input achieved the same amount of positive cases. Multi Otsu with cropped input and the addition of user input achieved the highest mean DSC of 0.435 and the same method without user input achieved the highest mean recall value of 0.520. Comparing the results of the latter method to the ones achieved by global thresholding with cropped input, the mean recall value is lower and the mean DSC is higher for the same set of 107 images. This means that the global thresholding method segmented more tumor pixels correctly but the Multi Otsu method had greater overlap between the prediction and the ground truth. This indicates that the thresholding method most likely classified surrounding liver pixels as tumor pixels in some cases, which is not necessarily a bad thing. As mentioned, it is of greater importance that all tumor pixels are correctly segmented than some surrounding healthy liver tissue being classified as tumor.

The same problem regarding introducing user input to the thresholding methods is also present in the Multi Otsu methods. That is, if the method works well on an image, it is not certain that the user input version of the same method works equally well. For the method with the entire liver as input, 52 images were positive cases, and when adding the user input option, 7 additional images were also classified as positive cases. For the cropped input with 72 positive cases adding user input resulted in 11 more positive cases.

During observation of the evaluation process, the most common problem was that the area containing the tumor was thresholded away after Multi Otsu thresholding. In these cases, the addition of user input made no difference since the tumor was already thresholded away. The problem is based on the fact that the method assumes that the tumor covers less of the input image than the liver does. Since this is not always the case, the tumor can get thresholded away. The same solution as for global thresholding would be to develop an automated way to make the method adjust for each individual image.

5.2 Active Contour Models

For the evaluation of the active contour models the same 107 images were used. The models required some parameters such as α and β for the Basic Snake model and a number of iterations for the Morphological Snake. All of these parameters were kept constant during the evaluation.

5.2.1 Basic Snake

From the results presented in Table 4.3 for the Basic Snake algorithm, it can be observed that the best results were achieved with a centred initial snake, with a mean DSC of 0.393 and a mean recall value of 0.644. This method reached the best mean recall value so far for all methods. One needs to be cautious with this positive outcome since the performance changes drastically with just a slight offset of the initial snake. With an offset of `A` or `B` % of the tumor diameter, the performance falls to be worse than the other methods evaluated so far. From this, it is clear that the Basic Snake algorithm is not robust.

The fragile state of the method bear concern. It can not be guaranteed that a surgeon will be able to initiate the first snake centred around the tumor in an exact way. From the result, it can be observed that this would cause huge tumor segmentation problems. Hence, the method is not robust enough to be used in a real scenario. Another problem with the method is the need for user input to choose both the initial snake as well as the parameters α and β. During this project, α and β have been kept constant during evaluation and have hence been the same for all images. The outcome could have been more satisfactory if the parameters had been chosen individually for each image but that would have been very time-consuming. Alike with the thresholding methods, it could be beneficial to develop an automated method to adjust the algorithm for each individual image.

5.2.2 Morphological Snake

From the results in Table 4.4 for the Morphological Snake algorithm, it can be observed that the method using both thresholding and user input achieved both the highest mean DSC of 0.340 as well as the highest mean recall value of 0.820. The mean recall value is the highest so far, while the mean DSC is not proportional compared to the other methods. This indicates that most of the tumor pixels are

correctly classified as tumor, but so are also a lot of surrounding tissue. As mentioned, this is not necessarily a problem since the most crucial aspect is to find all the tumor pixels, but when observing the results it is noted that the approach more or less segments out the entire region of interest instead of actually finding the tumor. An example of this can be seen in Figure 4.6 on the top images, where most of the pixels in the images are segmented as tumor. This explains the low mean DSC compared to the relatively high mean recall value for the Morphological Snake methods.

As for the thresholding methods utilising user input, the Morphological Snake could also receive better results for positive cases if the user input was added to the images where the regular method did not perform well. For the Morphological Snake without thresholding 75 images were positive cases and when adding user input another 18 images were classified as positive cases. For Morphological Snake with thresholding 70 images were positive cases and with user input another 33 ones were added. By changing the threshold value, the results could differ, and better performance could potentially be achieved.

5.2.3 Algorithm Comparison

Comparing all the different algorithm methods for tumor segmentation, it can be observed that Multi Otsu with cropped input and user input achieves the highest mean DSC (0.435) and that the Morphological Snake with thresholding and user input achieved the highest mean recall value (0.820). From what has been discussed, it is clear that the Morphological Snake method segments out the region of interest rather than the actual tumor. Therefore, it can be recognized that this method is not robust. The same goes for the Basic Snake. Therefore the best-performing algorithm methods are considered to be the thresholding methods, where cropped thresholding and cropped Multi Otsu achieved the greatest results.

5.3 U-Net

The evaluation was performed on 21 U-Net models with different parameters and inputs. In Table 4.5, the results for seven models trained with a setup of two-channel dice loss and cropped images as input, but with varying parameters, are presented. Among these models, Model 6 achieved the highest mean DSC of 0.758, while Models 4 and 7 obtained the highest mean recall value of 0.773. Moving on to models trained with a one-channel dice loss on the same input images, as shown in Table 4.6, Model 14 achieved the highest mean DSC of 0.766 and the highest mean recall value of 0.796. Finally, in the case of models utilizing liver-segmented images with a one-channel dice loss, Table 4.7 illustrates that Model 21 attained the highest mean DSC of 0.696, while Model 20 achieved the highest mean recall value of 0.767.

5.3.1 Deep Learning Comparison

To compare the performance of different model types, it is necessary to start by comparing similar models within each type. Starting with the cropped image input and two-channel dice loss, it can be observed in Table 4.5 that the outcome is quite similar for all models. The most significant difference in results for the mean DSC is between Model 2 with a value of 0.706 and Model 7 with a mean DSC of 0.758. Likewise, the mean recall value is the poorest for Model 2 and highest for Models 4 and 7, with the mean DSC of Model 7 being close to Model 6 at 0.757. These findings suggest that Model 2 performs the worst, while Models 4 and 7 perform the best within this group. When comparing the parameters of Models 4 and 7, the only distinction is the number of epochs trained. On the other hand, when contrasting Model 7 with Model 2, several differences emerge, including more augmented data, a higher number of training epochs (although with minimal impact compared to Model 4), and a smaller learning rate for Model 7. Based on these observations, it can be inferred that a smaller learning rate and increased use of augmented data could potentially lead to improved performance.

The results in Table 4.6 show a consistent pattern when examining models trained with cropped images and a one-channel dice loss. The most significant difference in mean DSC values is observed between Model 9, with a mean DSC of 0.722, and Model 14, with a mean DSC of 0.766. A similar pattern holds true for the mean recall value, with Model 9 achieving 0.712 and Model 14 achieving 0.796. These Models correspond to Models 2 and 7, respectively, from the first type of models discussed earlier. Notably, Model 9 lacks augmented data and employs a larger learning rate of `B`, similar to Model 2. On the other hand, Model 14, like Model 7, has been trained with augmented data with four variations and adopts a smaller learning rate of `A`. These findings suggest that similar conclusions can be drawn from this type of model as observed in the first type.

For the third type of models, with the segmented liver as input and a one-channel dice loss, see Table 4.7, the result stays consistent for the mean DSC but is lower than for the first two types. The mean DSC ranges from 0.638 for Model 17, to 0.696 for Model 21. The mean recall value covers a larger range of values, from 0.647 for Model 15, up to 0.767 for Model 20. The only thing consistent for the higher-performance models here is the larger amount of epochs. For Models 20 and 21, the training has been executed for `B` number of epochs, while Models 15 and 17 have only been trained for `A` number of epochs. Comparing the three models trained for `B` number of epochs, Models 19 - 21, it can be observed that the best performance is achieved with some added augmented data. This is also true for the models trained for `A` number of epochs with a learning rate of `A`. However, it should be noted that with no augmentation and `A` number of epochs, a learning rate of `B` showed the best result and might be a better approach when having the segmented liver as input. Hence evaluating the later models with a learning rate of `B` could be of interest. The reason why this was not performed during training was that the training scheme was based on the result and behaviour of the training and validation loss, as described in Subsection 3.4.2, and not on the result of the test set.

Comparing the different types of models, some things can be noted. First, there is a trend showing that better results are achieved with added augmented data and training for more epochs, while the learning rate is dependent on what type of input is being used. The best overall result is achieved by Model 14, suggesting that a one-channel dice loss is beneficial compared to a two-channel dice loss. Model 7 is trained with the same settings as Model 14 but with a two-channel dice loss and achieves a mean DSC of 0.757 compared to Model 14, with a mean DSC of 0.766. The results also suggest that a cropped input yields better results than a full-segmented liver image. However, it might be interesting to try another learning rate for the liver-segmented images.

As briefly mentioned in Section 4.4, no satisfactory results were generated for the two-channel dice loss combined with liver-segmented input. During training, the training and the validation loss did not converge and during evaluation, the trained models predicted that all pixels were background. This is likely caused by the class imbalance between the background and the tumor class. The fact that the majority of the pixels are background gives the dice loss computed from the background channel too big of an impact on the total dice loss, as described in Subsection 2.7.1.

5.3.2 Potential Areas of Improvement

During the training process, it has been observed that overfitting to the training data occurs. To mitigate this issue, several measures can be implemented. One common approach is to increase the amount of data available for training. This can be accomplished by incorporating additional datasets or enhancing the augmentation techniques to generate more diverse data from the existing set. By introducing more varied examples, the model can learn to generalize better and reduce overfitting. Other strategies that could be used to reduce overfitting are to use early stopping, dropout and L2 regularization, as described in Subsection 2.5.1. By employing a combination of these strategies, it is possible to address the overfitting issue and enhance the model's performance on unseen data.

From the research about U-Net explained in Subsection 2.5.1, there are plenty of opportunities to experiment with different factors that could increase the performance of the models even more, besides dealing with the overfitting problem. These include trying out different optimizers and loss functions. From the information gathered from the benchmark using the LiTS dataset, other optimizers to try could be SGD and SGD with momentum. Some other loss functions used in the LiTS benchmark were cross-entropy, weighted cross-entropy, and a combination of dice loss together with binary cross-entropy. The cross-entropy loss was used in the early parts of this project, but with poor results, probably due to the evident class imbalance, causing the switch to the dice loss. The use of weighted cross-entropy and a combination of dice loss together with binary cross-entropy was considered but never implemented. Improvements could also be made by changing parameters such as batch size, im-

age size, the learning rate as already mentioned, and of course, by implementing 3D input, which research showed promising results for, see Section 1.4. Changing the network architecture could also improve the results, and a few suggestions are presented in Subsection 2.5.3.

Further changes that could increase the performance of the models could be to improve the pre- and postprocessing methods. Currently, the preprocessing step is minimal with only converting the images to HU. This step could be expanded to involve some light thresholding or the application of different filters. Light thresholding could help the segmentation process and lead to improved models. The addition of a filter during the preprocessing step could reduce noise or aid feature enhancement. Some examples of techniques that can be utilized to refine the input images are median filtering, Gaussian smoothing, and morphological operations.

At present, the implemented models lack any postprocessing steps. However, considering the specific objective of segmenting a single tumor, postprocessing techniques can be introduced to enhance the results. A suggested postprocessing approach involves selecting and preserving the largest identified area exclusively, thereby discarding smaller regions that may be irrelevant. Furthermore, to ensure improved coverage and shape consistency, postprocessing operations such as convex hull or dilation can be employed. The convex hull operation generates a boundary that encapsulates the tumor, resulting in a smoother and more regular representation. Additionally, applying dilation allows for slight expansion of the segmented region, addressing potential under-segmentation and filling in any minor gaps or irregularities. By incorporating any of these pre- or postprocessing techniques, the models could be improved.

5.3.3 Comparison with Prior Work

When comparing the results of this study with the results of previous studies, caution should be exercised. First, it is important to acknowledge that a direct comparison between this study and the previous thesis work is not feasible due to differences in the datasets used and the absence of metrics in the earlier study. Second, it is also essential to state that a fair comparison between the results in this work and in previous works that have utilized the LiTS dataset, outlined in Section 1.4 and Subsection 2.5.3, cannot be made due to several reasons. One reason is that the models were based on 3D input, whereas the present work employs 2D input. Another reason is that different test sets have been used to obtain the results, see again Subsection 2.5.3. Furthermore, the rather small test set of 11 patients used in this study may imply statistical uncertainty around the mean test error, an effect described in Section 2.5.1. That is, if another random selection of 11 patients would be made, it is likely that the results and the performance of the models would be altered. This statistical uncertainty around the result of the U-Net models on the test set is important to keep in mind.

In addition to differences in input dimensionality and test set, a few more things should also be noted. Firstly, the contestants in the competition were required to accurately segment the liver before proceeding with tumor segmentation, a step omitted in this work. Secondly, it is important to recognize that the utilization of region of interest cropped input, which is employed in this work, would not meet the benchmark requirements. These two factors likely have a substantial impact on the results achieved in this work. The winners of the LiTS competition nnU-Net and the promising U^n-Net achieved a mean DSC value of 0.737 and 0.739, respectively, with a recall value of 0.554 for the nnU-Net model. The best results obtained in this thesis work comprise a DSC of 0.796 and a recall value of 0.766, both higher than what was achieved in the prior works mentioned. However, since the conditions for this study were not as challenging, cropped input could be utilised, and a functioning liver segmentation step is lacking in this work it is reasonable to conclude that further improvement of the results could be achieved by incorporating features from the models presented in Subsection 2.5.3.

5.4 Comparison Discussion

Since all classical algorithms share the same task of segmenting a single tumor and were evaluated using the same 107 test images, a comparison among them is feasible, see Subsection 5.2.3. The same holds for the U-Net models. Since they were developed to segment out all tumors in the images and were evaluated on the same test set consisting of 700 images from 11 patients, it is possible to compare different U-Net models with each other as in Subsection 5.3.1. However, due to the differences between the evaluation processes for the classical algorithms and the deep learning algorithms, comparing results across the two algorithm groups proves challenging. This is due to the different methods being evaluated on different datasets. Although evaluation of the classical algorithms on the test set could technically have been carried out, the inclusion of user input would have required more time than was available.

To enable a comprehensive comparison among the different algorithms and U-Net models, a smaller dataset containing 11 images was created for comparison purposes, as explained in Subsection 3.5.4. The results of this evaluation can be found in Table 4.8. Before analysing the results, it needs to be noted that the mean DSC and mean recall value could be greatly affected by which 11 images are selected to be in this very small dataset. Thus, this approach is not suitable for evaluating individual methods but rather serves as an opportunity to compare them based on this particular subset of 11 images.

The comparison results reveal that U-Net Model 4 achieved the highest mean DSC of 0.723 (and a mean recall value of 0.858), while U-Net Model 14 obtained the highest mean recall value of 0.890 (and mean DSC of 0.699). Model 4 also achieved a mean DSC of 0.757 and a mean recall value of 0.773 on the test set, while Model 14 obtained a mean DSC of 0.766 and a mean recall value of 0.796. Although some significant differences exist between the results on the test set and the comparison

dataset, these two models consistently performed the best in both scenarios. These results can be compared to the results of the algorithms that performed the best in individual evaluations, namely cropped thresholding and cropped Multi Otsu (with and without user input). The mean DSC values for thresholding were 0.294 and 0.277 and for Multi Otsu 0.329 and 0.227. The corresponding mean recall values for thresholding were 0.551 and 0.324 and for Multi Otsu 0.492 and 0.336. The Basic Snake method with a centred initial snake also demonstrated promising results, but its lack of robustness made the offset initial snake the worst-performing method, with a mean DSC of 0.012 and a mean recall value of 0.006. The Morphological Snake method with thresholding and user input followed the trend with a high mean recall value of 0.763 but a comparatively low mean DSC of 0.314.

Comparing the performance of the deep learning algorithms to the performance of the classical algorithms makes it clear that the best U-Net models performed almost twice as well as the best algorithm methods, the cropped thresholding methods. However, it is important to note that this comparison is specific to these particular 11 images and should not be generalized beyond that scope.

6
Conclusion

In this Master's thesis, a variety of different methods for semantic segmentation of liver tumors in CT images are investigated and evaluated with the purpose of finding a suitable segmentation algorithm for the company *Navari Surgical*. The investigated methods include variations of the classical segmentation algorithms global thresholding, Multi Otsu thresholding, Basic Snake, and Morphological Snake as well as a deep learning algorithm consisting of the U-Net architecture, the Adam optimizer and the dice loss function. Based on the results, the current best method is a U-Net model trained on cropped region-of-interest input with a one-channel dice loss. However, more research is needed before selecting a particular approach, and many areas of improvement exist. The current best model does not achieve a good enough performance to be used in practice.

Bibliography

[1] A. Toga and J. Mazziotta, *Brain Mapping: The Methods*, 2nd ed. Academic Press, 2002, ch. 17: CT Angiography and CT Perfusion Imaging.

[2] E. Kuntz and H.-D. Kuntz, *Hepatology, Principles and Practice: History, Morphology, Biochemistry, Diagnostics, Clinic, Therapy*, 2nd ed. Springer Science & Business Media, 2006, ch. 1.4, p. 171.

[3] P. Bilic, P. Christ, H. B. Li, E. Vorontsov, A. Ben-Cohen *et al.*, "The liver tumor segmentation benchmark (LiTS)," *Medical Image Analysis*, vol. 84, no. 102680, 2023.

[4] P.-J. C. Chun-Yu Liu, Kuen-Feng Chen, "Treatment of liver cancer," *Cold Spring Harb Perspect Med*, vol. 5, no. 9, 2015.

[5] E. Braunwarth, S. Stättner, M. Fodor, B. Cardini, T. Resch, R. Oberhuber, D. Putzer, R. Bale, M. Maglione, C. Margreiter, S. Schneeberger, D. Öfner, and F. Primavesi, "Surgical techniques and strategies for the treatment of primary liver tumours: hepatocellular and cholangiocellular carcinoma," *European Surgery*, vol. 50, no. 3, pp. 100–112, 2018.

[6] H. Tranchart and I. Dagher, "Laparoscopic liver resection: A review," *Journal of Visceral Surgery*, vol. 151, no. 2, pp. 107–115, 2014.

[7] E. Arnholm and J. Bergendahl, "Image segmentation of liver tumors in computed tomography scans," June 2022.

[8] P. F. Christ, M. E. A. Elshaer, F. Ettlinger, S. Tatavarty, M. Bickel, P. Bilic, M. Rempfler, M. Armbruster, F. Hofmann, M. D'Anastasi, W. H. Sommer, S. Ahmadi, and B. H. Menze, "Automatic liver and lesion segmentation in CT using cascaded fully convolutional neural networks and 3D conditional random fields," *CoRR*, vol. abs/1610.02177, 2016. [Online]. Available: http://arxiv.org/abs/1610.02177

[9] F. Isensee, P. F. Jaeger, S. A. A. Kohl, J. Petersen, and K. H. Maier-Hein, "nnU-Net: a self-configuring method for deep learning-based biomedical image segmentation," *Nature Methods*, vol. 18, no. 2, pp. 203–211, 2021.

[10] I. M. Arias, H. J. Alter, J. L. Boyer, D. E. Cohen, D. A. Shafritz, S. S. Thorgeirsson, and A. W. Wolkoff, *The Liver : Biology and Pathobiology*, 6th ed. John Wiley & Sons, Incorporated, 2020, ch. 1, pp. 3–12.

[11] M. Sharma, P. Somani, C. S. Rameshbabu, T. Sunkara, and P. Rai, "Stepwise evaluation of liver sectors and liver segments by endoscopic ultrasound," *World journal of gastrointestinal endoscopy*, vol. 10, no. 11, p. 326–339, 2018.

[12] S. T. Orcutt and D. A. Anaya, "Liver resection and surgical strategies for management of primary liver cancer," *Cancer Control*, vol. 25, no. 1, pp. 1–15, 2018.

[13] S. Buettner, J. L. van Vugt, J. N. IJzermans, and B. G. Koerkamp, "Intrahepatic cholangiocarcinoma: current perspectives," *Onco Targets Ther*, vol. 10, p. 1131–1142, 2017.

[14] I. M. Arias, H. J. Alter, J. L. Boyer, D. E. Cohen, D. A. Shafritz, S. S. Thorgeirsson, and A. W. Wolkoff, *The Liver : Biology and Pathobiology*, 6th ed. John Wiley & Sons, Incorporated, 2020, ch. 61, pp. 782–789.

[15] J. L. Prince and J. M. Links, *Medical imaging: signal and systems*, 2nd ed. Upper Saddle River, N.J.: Pearson Education, 2014, ch. 1.3, p. 7.

[16] ——, *Medical imaging: signal and systems*, 2nd ed. Upper Saddle River, N.J.: Pearson Education, 2014, ch. 1.4-5, pp. 7–9.

[17] ——, *Medical imaging: signal and systems*, 2nd ed. Upper Saddle River, N.J.: Pearson Education, 2014, ch. 1.1, p. 5.

[18] ——, *Medical imaging: signal and systems*, 2nd ed. Upper Saddle River, N.J.: Pearson Education, 2014, ch. 6.3, pp. 197–213.

[19] B. Blanche, L. Matthieu, B. Romain, V. Valérie, and R. Maxime, "Cone Beam Computed Tomography (CBCT) in the Field of Interventional Oncology of the Liver," *Cardiovasc Intervent Radiol*, vol. 39, no. 1, pp. 8–20, 2016.

[20] C. Bodin, "Robust augmented reality navigation algorithm for laparoscopic liver tumor resection," Master thesis, Chalmers University of Technology, 2022.

[21] J. L. Prince and J. M. Links, *Medical imaging: signal and systems*, 2nd ed. Upper Saddle River, N.J.: Pearson Education, 2014, ch. 3.2, pp. 55–60.

[22] ——, *Medical imaging: signal and systems*, 2nd ed. Upper Saddle River, N.J.: Pearson Education, 2014, ch. 6.2.3, pp. 141–143.

[23] ——, *Medical imaging: signal and systems*, 2nd ed. Upper Saddle River, N.J.: Pearson Education, 2014, ch. 6.4.3, pp. 221–223.

[24] X. Li, P. S. Morgan, J. Ashburner, J. Smith, and C. Rorden, "The first step for neuroimaging data analysis: Dicom to nifti conversion," *Journal of Neuroscience Methods*, vol. 264, pp. 47–56, 2016.

[25] A. Norouzi, M. S. M. Rahim, A. Altameem, T. Saba, A. E. Rad, A. Rehman, and M. Uddin, "Medical image segmentation methods, algorithms, and applications," *IETE Technical Review*, vol. 31, no. 3, pp. 199–213, 2014.

[26] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 4th ed. 330 Hudson Street, New York, NY 10013: Pearson, 2018, ch. 10.3, pp. 742–760.

[27] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE transactions on systems, man, and cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.

[28] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *International journal of computer vision*, vol. 1, no. 4, pp. 321–331, 1988.

[29] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 3rd ed. Upper Saddle River, N.J.: Pearson Prentice Hall, 2008, ch. 9-9.1, pp. 649–652.

[30] ——, *Digital Image Processing*, 3rd ed. Upper Saddle River, N.J.: Pearson Prentice Hall, 2008, ch. 9.2, pp. 652–657.

[31] ——, *Digital Image Processing*, 3rd ed. Upper Saddle River, N.J.: Pearson Prentice Hall, 2008, ch. 9.3, pp. 657–661.

[32] S. Kichenassamy, A. Kumar, P. Olver, A. Tannenbaum, and A. Yezzi, "Gradient flows and geometric active contour models," in *Proceedings of IEEE International Conference on Computer Vision*, 1995, pp. 810–815.

[33] L. Álvarez, L. Baumela, P. Henríquez, and P. Márquez-Neila, "Morphological snakes," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2010, pp. 2197–2202.

[34] A. A. Albishri, S. J. H. Shah, and Y. Lee, "Cu-net: Cascaded u-net model for automated liver and lesion segmentation and summarization," in *2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, 2019, pp. 1416–1423.

[35] S.-T. Tran, C.-H. Cheng, and D.-G. Liu, "A multiple layer u-net, un-net, for liver and liver tumor segmentation in ct," *IEEE Access*, vol. 9, pp. 3752–3764, 2021.

[36] S. Gul, M. S. Khan, A. Bibi, A. Khandakar, M. A. Ayari, and M. E. Chowdhury, "Deep learning techniques for liver and liver tumor segmentation: A review," *Computers in Biology and Medicine*, vol. 147, 2022.

[37] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, 1st ed. MIT Press, 2016, ch. 5, pp. 109–159.

[38] ——, *Deep Learning*, 1st ed. MIT Press, 2016, ch. 6, pp. 170–200.

[39] ——, *Deep Learning*, 1st ed. MIT Press, 2016, ch. 4, pp. 95–98.

[40] ——, *Deep Learning*, 1st ed. MIT Press, 2016, ch. 11, pp. 392–405.

[41] ——, *Deep Learning*, 1st ed. MIT Press, 2016, ch. 7, pp. 222–258.

[42] ——, *Deep Learning*, 1st ed. MIT Press, 2016, ch. 8, pp. 263–312.

[43] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014.

[44] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," 2015, arXiv:1505.04597. [Online]. Available: https://arxiv.org/abs/1505.04597

[45] M. Antonelli, A. Reinke, S. Bakas *et al.*, "The medical segmentation decathlon," *Nature Communications*, vol. 13, no. 4128, 2022. [Online]. Available: https://doi.org/10.1038/s41467-022-30695-9

[46] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," 2015. [Online]. Available: https://arxiv.org/abs/1511.07122

[47] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," 2016. [Online]. Available: https://arxiv.org/abs/1608.06993

[48] S. Jadon, "A survey of loss functions for semantic segmentation," in *2020 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*. IEEE, 2020.

[49] C. H. Sudre, W. Li, T. Vercauteren, S. Ourselin, and M. J. Cardoso, "Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations," in *Deep learning in medical image analysis and multimodal learning for clinical decision support : Third International Workshop, DLMIA 2017, and 7th International Workshop, ML-CDS 2017, held in conjunction with MICCAI 2017 Quebec City.* , 2017, p. 240–248.

[50] K. Clark, B. Vendt, K. Smith, J. Freymann, J. Kirby, P. Koppel, S. Moore, S. Phillips, D. Maffitt, L. T. Michael Pringle, and F. Prior, "The cancer imaging archive (TCIA): Maintaining and operating a public information repository," *Journal of Digit Imaging*, vol. 26, p. 1045–1057, 2013.

[51] B. J. Erickson, J. K. Smith, S. Kirk, O. Bathe, M. Kearns, J. Lemmerman, and K. Reiger-Christ, "The cancer genome atlas liver hepatocellular carcinoma collection (TCGA-LIHC) (Version 5) [Data set]," *The Cancer Imaging Archive.*, 2016.

[52] Innolitics, "Rescale Intercept Attribute - DICOM Standard Browser," Accessed 13rd of February 2023. [Online]. Available: https://dicom.innolitics.com/ciods/ct-image/ct-image/00281052

[53] L. He, X. Ren, Q. Gao, X. Zhao, B. Yao, and Y. Chao, "The connected-component labeling problem: A review of state-of-the-art algorithms," *Pattern Recognition*, vol. 70, pp. 25–43, 2017.

[54] P. Márquez-Neila, L. Baumela, and L. Alvarez, "A morphological approach to curvature-based evolution of curves and surfaces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 1, pp. 2–17, 2014.

DEPARTMENT OF MATHEMATICAL SCIENCES
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY