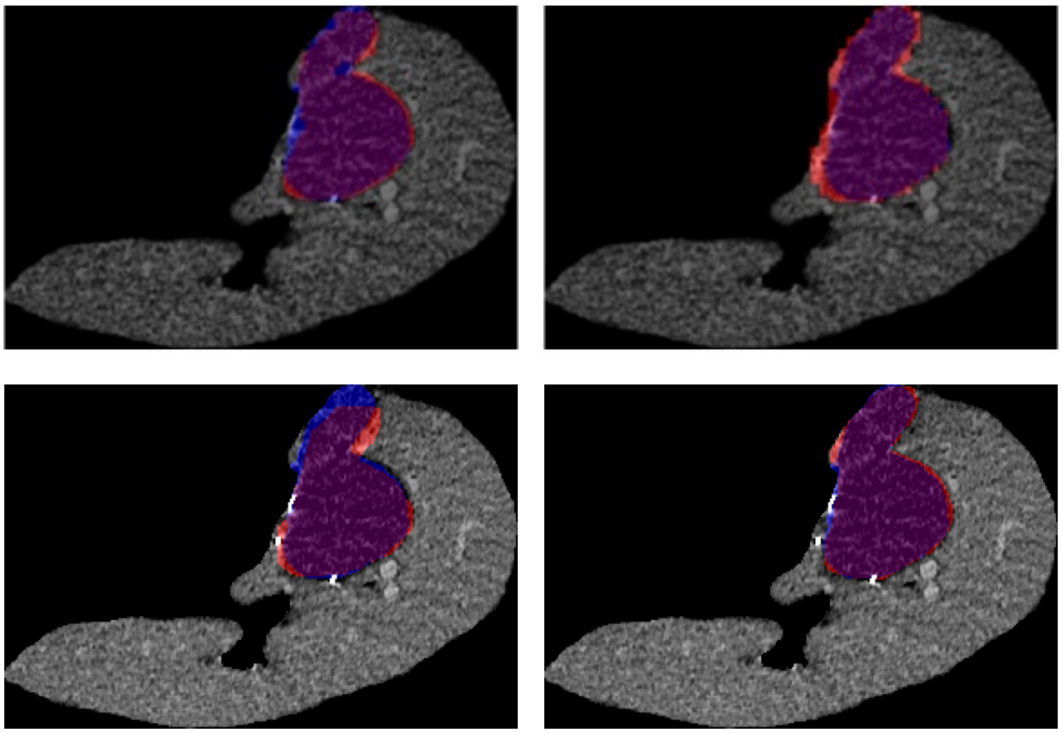
CHALMERS
UNIVERSITY OF TECHNOLOGY



Segmentation of Liver Tumours Using Artificial Intelligence

Master's thesis in Biomedical Engineering

VIRENA NASSIF
HANNA ÅVALL

DEPARTMENT OF MATHEMATICAL SCIENCES

CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2024
www.chalmers.se

Master's thesis 2024

Segmentation of Liver Tumours Using Artificial Intelligence

VIRENA NASSIF, HANNA ÅVALL

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Mathematical Sciences
Chalmers University of Technology
Gothenburg, Sweden 2024

Segmentation of Liver Tumours Using Artificial Intelligence
VIRENA NASSIF, HANNA ÅVALL

Supervisor: Klas Modin, Mathematical Sciences
Examiner: Klas Modin, Mathematical Sciences

Master's Thesis 2024
Department of Mathematical Sciences
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: An example of liver tumour segmentation using U-Net, VGG16, YOLOv8, and SAM. All networks, except SAM, were trained on livers and liver tumours. Purple indicates overlap, red indicates false positives, and blue indicates false negatives.

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2024

Segmentation of Liver Tumours Using Artificial Intelligence
VIRENA NASSIF, HANNA ÅVALL
Department of Mathematical Sciences
Chalmers University of Technology

Abstract

Liver cancer is a serious health condition affecting approximately 800,000 people annually, with around 700,000 deaths worldwide each year from this disease. One of the primary treatment methods for liver cancer is the surgical removal of the tumour. Keyhole surgery, which uses small incisions instead of a large cut, offers several benefits, including shorter hospital stays, reduced risk of complications, and cosmetic advantages such as smaller scars. However, despite its advantages, keyhole surgery is less commonly performed. Navari Surgical, a Medtech startup company founded in 2021, aims to address this challenge by developing a visual aid specifically for keyhole surgeries involving liver tumour removal, benefiting both patients and the healthcare system by improving outcomes and efficiency.

This Master's thesis investigates deep learning networks for semantic segmentation of liver tumours in CT images. This includes comparing the performance of transfer learning networks with the U-Net architecture. This analysis highlights the importance of careful dataset preparation, thoughtful model selection, and hyperparameter tuning to optimise model performance.

In conclusion, U-Net demonstrated the best performance compared to the transfer learning networks, especially when prioritising the Dice score over recall. The recommendation for Navari is to further develop the U-Net architecture for future advancements in tumour segmentation. Alternatively, make small adjustments to the transfer learning networks that may lead to better performance.

Keywords: Liver tumour, Navari, Deep learning, LiTS dataset, U-Net, transfer learning, YOLO, SAM, VGG16

Acknowledgements

We would like to thank our beloved friends and family for their consistent support throughout this project. We also express our deepest gratitude to Carl Bodin and David Löfstrand for their guidance and knowledge during the project. Additionally, we thank our supervisor and examiner, Klas Modin, for his valuable insights and constructive feedback. Lastly, we extend our thanks to Madeleine Gustavsson and Axel Blomé for their positivity and continuous support. Their presence made our time in the office truly enjoyable and inspiring.

Virena Nassif & Hanna Åvall, Gothenburg, June 2024

List of Acronyms

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

Adam	Adaptive Moment Estimation
AI	Artificial Intelligence
AR	Augmented Reality
BCE	Binary Cross Entropy
CBCT	Cone Beam Computed Tomography
CIoU	Complete Intersection over Union
CNN	Convolutional Neural Network
COCO	Common Objects in Context
CT	Computed Tomography
DICOM	Digital Imaging and Communications in Medicine
DFL	Distributional Focal Loss
FN	False Negative
FP	False Positive
HCC	Hepatocellular Carcinoma
HU	Houndsfield Unit
ICC	Intrahepatic Cholangiocarcinoma
IoU	Intersection over Union
LiTS	Liver Tumor Segmentation Benchmark
MRI	Magnetic Resonance Imaging
NIFTI	Neuroimaging Informatics Technology Initiative
ROI	Region of Interest
SAM	Segment Anything Model
TL	Transfer Learning
TP	True Positive
YOLO	You Only Look Once

Contents

List of Acronyms **ix**

List of Figures **xv**

List of Tables **xvii**

1 Introduction **1**
1.1 Background . . . 1
1.2 Aim . . . 2
1.3 Limitations . . . 2
1.4 Prior Work . . . 2

2 Theory **5**
2.1 Medical background . . . 5
2.1.1 Liver Anatomy . . . 5
2.1.2 Liver Cancer . . . 5
2.2 Medical imaging . . . 6
2.2.1 Computed Tomography . . . 6
2.2.2 Cone Beam Computed Tomography . . . 7
2.2.3 Contrast Agent . . . 7
2.2.4 Hounsfield Units . . . 7
2.2.5 Medical File Formats . . . 8
2.3 Dataset for Liver Cancer . . . 8
2.3.1 LiTS dataset . . . 9
2.4 Deep Learning . . . 10
2.4.1 Transfer Learning . . . 10
2.4.2 Basic Concepts of Deep Learning . . . 11
2.4.3 U-Net . . . 12
2.4.3.1 VGG16 . . . 13
2.4.3.2 YOLOv8 . . . 14
2.4.3.3 SAM . . . 16
2.5 Evaluation metrics for liver segmentation . . . 17
2.5.1 Dice Score . . . 17
2.5.2 Recall (sensitivity) . . . 18
2.6 Loss Function . . . 19
2.6.1 Dice Loss . . . 19
2.6.2 Focal Tversky . . . 19

3 Methods **21**
3.1 General Workflow . . . 21
3.2 Datasets . . . 22
3.2.1 Liver . . . 22
3.2.2 Region of Interest . . . 23
3.3 Preparation of dataset . . . 23
3.3.1 Preprocessing . . . 23
3.3.1.1 Cropping . . . 23
3.3.1.2 HU Conversion . . . 24
3.3.1.3 Histogram Equalisation . . . 24
3.3.1.4 Data Augmentation . . . 24
3.4 Training . . . 24
3.4.1 The assembly of VGG16 . . . 24
3.4.2 The assembly of YOLOv8 . . . 25
3.4.3 Model Training Configurations . . . 25
3.4.4 Saving the Best Model . . . 27
3.4.5 SAM . . . 27
3.5 Evaluation . . . 27
3.5.1 Recall . . . 27
3.5.2 Dice Score . . . 28
3.5.3 Weighted Score . . . 28

4 Results **29**
4.1 U-Net . . . 29
4.1.1 LIVER . . . 29
4.1.2 ROI . . . 30
4.2 VGG16 . . . 31
4.2.1 LIVER . . . 32
4.2.2 ROI . . . 33
4.3 YOLOv8 . . . 34
4.3.1 LIVER . . . 34
4.3.2 ROI . . . 35
4.4 SAM . . . 36
4.4.1 LIVER . . . 36
4.4.2 ROI . . . 37
4.5 Comparison between the networks . . . 38
4.5.1 LIVER . . . 38
4.5.2 ROI . . . 40

5 Discussion **43**
5.1 U-Net . . . 43
5.2 VGG16 . . . 44
5.3 YOLOv8 . . . 44
5.4 SAM . . . 45
5.5 Comparison . . . 45
5.5.1 Transfer learning . . . 46
5.5.2 Liver Dataset . . . 46

5.5.3 Region of Interest Dataset 47
5.5.4 General Discussion 47
5.6 Comparison to past work 48
5.7 Future recommendations 49

6 Conclusion **51**

References **53**

List of Figures

2.1 Deep learning is a subset of machine learning which in turn, is a subset of AI [1]. . . . 10
2.2 The network architecture of a U-Net, showing the encoding and decoding layers [2] . . . 13
2.3 The network architecture of VGG16 [3]. . . . 14
2.4 YOLOv8 architecture [4]. . . . 15
2.5 YOLOv8 sizes and its corresponding properties [5]. . . . 16
2.6 The architecture of SAM, including an image encoder, a prompt encoder, and a lightweight mask decoder, as well as the various prompts that can be given to the architecture. There is also an example input image and potential masks for it [6]. . . . 17
2.7 Light blue represent the predicted segmentation and the dark blue the true label or ground truth. FP are pixels wrongfully classified as tumour, FN are pixels wrongfully classified as non-tumour, and TP are pixels correctly classified as tumour. . . . 18

4.1 An example image from the test set evaluated on model 1- 6. Purple indicates overlap, red indicates false positives, and blue indicates false negatives. . . . 30
4.2 An example image from the test set evaluated on model 7- 12. Purple indicates overlap, red indicates false positives, and blue indicates false negatives. . . . 31
4.3 An example image from the test set evaluated on model 13- 18 . Purple indicates overlap, red indicates false positives, and blue indicates false negatives. . . . 32
4.4 An example image from the test set evaluated on model 13- 18 . Purple indicates overlap, red indicates false positives, and blue indicates false negatives. . . . 33
4.5 An example image from the test set evaluated on model 25 - 27. Purple indicates true positive, red indicates false positives, and blue indicates false negatives . . . 35
4.6 An example image from the test set evaluated on model 28 - 30. Purple indicates true positive, red indicates false positives, and blue indicates false negatives . . . 36

4.7 An example image from the test set evaluated on models 31 and 32. Purple indicates true positive, red indicates false positives, and blue indicates false negatives 37
4.8 An example image from the test set evaluated on model 33. Purple indicates true positive, red indicates false positives, and blue indicates false negatives 38
4.9 An example image from the test set evaluated on each network with the highest weighted score for the LIVER dataset. Purple indicates true positives, red indicates false positives, and blue indicates false negatives. 40
4.10 An example image from the test set evaluated on each network with the highest weighted score for the ROI dataset. Purple indicates overlap, red indicates false positives, and blue indicates false negatives. 42

List of Tables

1.1 The evaluation metrics of previous work done in the area of semantic segmentation using the U-Net model. The result presented is the U-Net model with the highest weighted scores from each dataset. . . 3

2.1 Summary of datasets for liver and liver tumour segmentation [7] . . . 8
2.2 Characteristics of LiTS Dataset 9

3.1 The amount of images and respective labels divided into training, validation, and test sets for the two different datasets, LIVER and ROI. 22
3.2 Data augmentation used on the datasets. 24
3.3 Showing the training configuration for each model, including model number, network, dataset, learning rate for freeze and unfreeze as well as loss function where applicable. 26
3.4 Training configurations for the different networks, including image and batch size. Specific configurations vary according to each network's input format and GPU capacity. 26

4.1 Learning rate combinations used for transfer learning networks. *Learning Rate Freeze* refers to the learning rate applied when the backbone is frozen, while *Learning Rate Unfreeze* refers to the learning rate used when the backbone is unfrozen. 29
4.2 Overview of U-Net models for the LIVER dataset running for I+II epochs each where the best models according to the lowest validation loss are saved. The evaluation scores, Dice score, recall and weighted score, are also presented for each model. 30
4.3 Overview of U-Net models for the ROI dataset running for I+II epochs each where the best models according to the lowest validation loss are saved. The evaluation scores, Dice score, recall and weighted score, are also presented for each model. 31
4.4 Overview of VGG16 models for the LIVER dataset running for I+II epochs each where the best models according to the lowest validation loss are saved. The evaluation scores, Dice score, recall and weighted score, are also presented for each model. 32

4.5 Overview of VGG16 models for the ROI dataset running for I+II epochs each where the best models according to the lowest validation loss are saved. The evaluation scores, Dice score, recall and weighted score, are also presented for each model. 33
4.6 Overview of YOLOv8 models for the LIVER dataset running for I epochs with frozen backbone and II epochs with all layers unfrozen, where the best models according to the lowest validation loss are saved. Which learning rate combination, as well as the evaluation scores, Dice score, recall and weighted score, are presented for each model. 34
4.7 Overview of YOLOv8 models for the ROI dataset running for I epochs with frozen backbone and II epochs with all layers unfrozen, where the best models according to the lowest validation loss are saved. Which learning rate combination, as well as the evaluation scores, Dice score, recall and weighted score, are presented for each model. . 35
4.8 Overview of SAM models for the LIVER dataset. The maximum pixel value added from the label to the generated bounding box, as well as the evaluation scores, Dice score, recall and weighted score, are presented for each model. 37
4.9 Overview of SAMs model for the ROI dataset. The evaluation scores Dice score, recall and weighted score, are presented here. 37
4.10 Summary of all models used on the LIVER dataset. Showing the model number, network as well as the evaluation metrics Dice score, recall and weighted score. 39
4.11 The average Dice score and recall and the variation for both for each network on the LIVER dataset. 39
4.12 Summary of all models used on the ROI dataset. Showing the model number, network and evaluation metrics Dice score, recall and weighted score. 41
4.13 The average and variation of the weighted score for the ROI dataset. 41

1
Introduction

In this chapter, the background for this project will be presented, outlining the context and significance of the work undertaken. Furthermore, aim and limitations will be formulated from the background of the project. Additionally, prior work will be presented to understand gaps in the literature and areas where this project can contribute new insights.

1.1 Background

Liver cancer occurs when malignant cells form in the liver tissue, leading to a serious health condition affecting approximately 800,000 people, and causing around 700,000 deaths worldwide annually [8]. One of the primary treatment methods for liver cancer is the surgical removal of the tumour [9]. There are two main surgical approaches for liver tumour removal: open surgery, which involves making a large incision in the abdomen, and minimally invasive surgery, known as keyhole surgery.

Keyhole surgery, which uses small incisions instead of a large cut, offers several benefits, including shorter hospital stays, reduced risk of complications, and cosmetic advantages such as smaller scars [10]. However, a significant drawback of keyhole surgery is the limited visualization it provides. Unlike open surgery, where the surgeon can directly see and feel the tissues, keyhole surgery relies on instruments that restrict the surgeon's view. As a result, despite its advantages, keyhole surgery is less commonly performed.

Navari Surgical, a Medtech startup company founded in 2021, aims to address this challenge by developing a visual aid specifically for keyhole surgeries involving liver tumour removal [10]. Navari's proposed solution leverages Augmented Reality (AR) to project the tumour onto the surgical screen, thereby enhancing navigation and precision[10]. This innovation aims to make it easier and faster for surgeons to locate and remove the tumour, reducing the risk of mistakes, minimising the need for large safety margins, and shortening operation times. Ultimately, Navari hopes that this technology will increase the prevalence of keyhole surgeries for liver tumour removal, benefiting both patients and the healthcare system by improving outcomes and efficiency. Their solution involves an advanced segmentation process to accurately identify the tumour in medical images. Although liver segmentation has nearly achieved human-like precision, robust liver tumour segmentation remains a significant challenge due to the lack of comprehensive datasets and the tedious

process of annotating images [11].

During keyhole surgeries, distinguishing the tumour from healthy liver tissue is particularly challenging [12]. Surgeons currently mitigate this difficulty by studying preoperative medical images, such as computed tomography (CT) scans, to visualise and localise the tumour in a 3D environment during the procedure. This approach is cognitively demanding and leaves a considerable margin for error, posing risks to the patient.

1.2 Aim

This Master's thesis aims to investigate deep learning networks for semantic segmentation of liver tumours in CT images, suitable for Navari Surgical. This includes comparing the performance of transfer learning networks with the U-Net architecture.

1.3 Limitations

This Master's thesis work has several limitations. For the transfer learning models, YOLOv8 (size medium), VGG16, and SAM (size ViT-B) will be investigated, with SAM being used as untrained. These models will be compared to U-Net. Two datasets derived from the dataset used in the LiTS dataset challenge will be used. One dataset is cropped around the liver (LIVER), and the other is cropped as a region of interest (ROI) around the tumour. Only the LIVER dataset contains liver images without tumours, while the ROI dataset contains one tumour per image. During training, all images and their corresponding labels will be augmented randomly with shear, rotation, and scale. Additionally, the neural networks will only be trained in 2D. Although cone beam computed tomography (CBCT) images will be used during surgery, only CT scans will be available for this thesis work. The report will focus on specific loss functions, for VGG16 and U-Net, Dice loss and focal Tversky loss will be used. For the YOLOv8m transfer learning network, the built-in loss function will be used. Evaluation metrics for this report will be limited to Dice score and recall.

1.4 Prior Work

Semantic segmentation of medical images has been the subject of extensive research. Previous thesis work has primarily focused on the application of segmentation algorithms for liver tumour identification [13]. Additionally, studies utilizing artificial intelligence (AI), particularly the original U-Net model, have shown promising potential in this area.

In earlier research, the LiTS dataset was employed, consisting solely of tumour images [13]. Two subsets of this dataset were used: one containing segmented liver images (LIVER) and the other focusing on regions of interest (ROI) around the tumours. The LIVER dataset includes images with multiple tumours, while the ROI dataset typically contains a single tumour, though it sometimes features other tumours nearby. Furthermore, both datasets underwent data augmentation.

The results from prior work using the U-Net model are summarised in Table 1.1. Only the models with the highest weighted scores are presented, along with their Dice score and recall. The methodology for calculating the weighted score is detailed in section 3.5.2.

Dataset	Dice Score	Recall	Weighted Score
LIVER	0.689	0.767	0.720
ROI	0.766	0.796	0.778

Table 1.1: The evaluation metrics of previous work done in the area of semantic segmentation using the U-Net model. The result presented is the U-Net model with the highest weighted scores from each dataset.

2
Theory

In this chapter, the necessary theory for this project is presented. It will begin with the medical background, followed by an overview of medical imaging to explain the basics of liver anatomy and the relevant modalities for this project. Next, the dataset used for this project will be introduced. Then follows the theory behind deep learning, along with the different deep learning network architectures employed in this project. Finally, the evaluation metrics, recall, and Dice score will be explained to determine the model that performs the best.

2.1 Medical background

The medical background needed for the project is presented in this section, starting with the anatomy of the liver and then different forms of liver cancer.

2.1.1 Liver Anatomy

The liver is the largest internal organ in the mammalian body, it also performs a variety of vital and complex functions [14]. These important functions include the maintenance of blood sugar by glycogen storage, protein synthesis, detoxification and production of bile. The liver has a sponge-like texture and weighs around 1300g to 1700g in an adult human. The liver consists of two types of cells only found in the liver, hepatocytes and biliary cells. The hepatocytes are highly polarised cells that are responsible for tasks such as glycogen storage, detoxification and production of bile. The cholangiocytes form biliary channels and modify the composition of the bile by secretion and absorption of compounds. When needed, the liver can regenerate tissue through the proliferation of hepatocytes and cholangiocytes amongst other cell types. The newly generated liver tissue is very similar to the original one, and as much as 70% of the liver's mass can be regenerated. These regenerating properties are essential after surgical resection of a portion of the liver, as the lost tissue might need to be compensated.

2.1.2 Liver Cancer

Cancer can be divided into two categories; primary and secondary. Primary cancer in the liver, originates from the liver, while secondary cancer originates from another part of the body and metastasised to the liver. Primary liver cancer is the fifth most common cancer and is the second most common cause of cancer deaths [15]. Two

of the most common liver cancers are hepatocellular carcinoma (HCC) and intrahepatic cholangiocarcinoma (ICC), combined being responsible for more than 95% of all primary liver cancers. 80-85% of all cases are caused by HCC, making it by far the most common one. HCC often affects people who suffer from chronic liver diseases, such as hepatitis B and C, as it begin in the hepatocytes. ICC, however, forms in the bile ducts of the liver [16].

The preferred treatment is generally liver resection for patients with sufficient liver function [17]. This excludes patients who suffer from cirrhosis, a condition where the liver has been damaged due to liver disease. Due to the regenerative abilities of the liver, it is possible to resect large tumours [15], however, it is important to remove all tumours during the procedure. When the liver function is insufficient or in the case of metastatic liver cancer, the treatment of choice is instead a liver transplant. A shortage of donors and the complexity of the surgery complicates this approach though.

2.2 Medical imaging

Medical imaging refers to various modalities used to visualise anatomical structures and physiological functions within the body, categorised by the type of signals they utilise and their use of ionising radiation [18]. Modalities that use ionising radiation include radiography and computed tomography, both of which employ the transmission of X-ray beams. As these beams pass through the body, they are attenuated by different tissues to form images. Another ionising technique, nuclear medicine, Single Photon Emission Computed Tomography and Positron Emission Tomography scan, involves injecting radioactive compounds into the body; the gamma rays emitted from these radioactive tracers provide diagnostic images based on the local concentration of the compound.

In contrast, non-ionising modalities such as ultrasound and magnetic resonance imaging (MRI) offer safer diagnostic alternatives. Ultrasound utilises high-frequency sound waves sent into the body, capturing the echoes that bounce back from internal structures to create images. MRI employs a strong magnetic field combined with radio frequencies to influence the properties of the proton nuclei in hydrogen atoms, enabling the production of detailed internal body images. Together, these modalities provide critical insights into the body, aiding in diagnosis and treatment planning without the inherent risks of ionising radiation for ultrasound and MRI.

2.2.1 Computed Tomography

Computed Tomography (CT) scan, is a diagnostic imaging exam based on computerised X-ray [19]. CT scans address several limitations of traditional X-rays, including the projection of three-dimensional objects into two-dimensional images, difficulties in capturing low-contrast objects, and the inability to accurately determine the density of an object [20]. The X-ray beams are fan-shaped and aimed at the body while the X-ray source quickly rotates around the patient while the patient

moves slowly, this allows for several different views and more details to be captured when compared to traditional X-ray [19]. Instead of film, the CT scanner uses special digital X-ray detectors, situated opposite of the X-ray source, the scanner transmits a signal to a computer which is then converted into a 2D image. Each rotation the X-ray source conducts results in cross-sectional images or a "slice", which can then be displayed individually or stacked together to generate a 3D image of the selected area. CT scans are used to identify disease or injury and have become a popular tool for detecting tumours or lesions, particularly within the abdomen.

2.2.2 Cone Beam Computed Tomography

There is a specialised form of CT known as Cone Beam Computed Tomography (CBCT). Unlike traditional CT scans that use a fan-shaped beam, CBCT employs a cone-shaped beam [21]. Compared to fan beam CT, CBCT typically exhibits a higher presence of artefacts, more noise, a lower signal-to-noise ratio, and a reduced ability to discriminate low-contrast objects. However, a significant advantage of CBCT is that it exposes the patient to significantly less radiation—about half to one-third of conventional CT scans. This reduction in radiation, though, often results in poorer image quality. Navari's plan is to use CBCT for their solution, however, CT images are used in this master thesis project based on availability and the assumption that they are similar enough to translate any findings.

2.2.3 Contrast Agent

Contrast agents are substances used in medical imaging to enhance the visibility of internal organs, blood vessels, and tissues, making them more distinguishable on scans [22]. These agents work by altering the way imaging technologies interact with the body. For instance, in CT and MRI scans, contrast agents increase the contrast between different tissues, improving the clarity and detail of the images. Common types include iodine- and barium-based compounds for X-rays and CT scans. These agents are crucial for diagnosing conditions that might not be visible on standard scans, such as tumours or vascular diseases.

2.2.4 Hounsfield Units

CT scanners measure the linear attenuation coefficients (μ) of tissues, due to variations in scanner hardware, identical tissues can show different μ values on different scanners or even on the same scanner over time [23]. To standardise readings across different conditions, CT numbers are calculated using the formula:

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

expressed in Hounsfield Units (HU). μ_{water} is set at 0 HU and μ_{air} at -1,000 HU. This standardisation helps in comparing scans by converting raw attenuation data into a consistent scale, where typically, bone reaches around 1,000 HU and metal can exceed 3,000 HU.

2.2.5 Medical File Formats

Medical data is stored in various file formats, with NIfTI (Neuroimaging Informatics Technology Initiative) files being one of the most common. The LiTS dataset, as described in Section 2.3.1, is organised in NIfTI format [24]. NIfTI is known for its simplicity and minimalistic structure, making it a preferred choice for certain types of medical imaging data. In contrast, DICOM (Digital Imaging and Communications in Medicine) is a more complex and comprehensive format. DICOM is widely used for storing data from medical scanners and can include embedded data such as patient notes and audio files. Due to its detailed structure, DICOM is the standard for clinical and diagnostic imaging.

In this project, NIfTI files from the LiTS dataset were converted into DICOM files. This conversion facilitates the integration of the data into clinical workflows, allowing for more comprehensive utilisation of the medical images and associated metadata.

2.3 Dataset for Liver Cancer

There are several datasets available for segmenting liver and liver tumour images in different modalities [7]. Table 2.1 provides an overview of publicly available datasets, detailing liver, tumour and background annotations.

Table 2.1: Summary of datasets for liver and liver tumour segmentation [7]

Dataset	Institution	Liver	Tumour	Annotation	#Volume	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

Most of the datasets do not provide annotated data for both liver and tumour, with the exception of the LiTS and 3Dircadb datasets. For this project, the LiTS dataset will be used as it offers the largest collection of data with annotations for liver, tumour, and background.

2.3.1 LiTS dataset

The Liver Tumour Segmentation Benchmark (LiTS) dataset was collaboratively assembled by seven research institutions and medical centres worldwide [7]. It comprises 201 CT images of the abdomen, 194 of which include liver tumours. This dataset plays a crucial role in the LiTS benchmark challenge, aimed at fostering the development of advanced deep learning models specifically for liver and liver tumour segmentation.

The benchmark challenge associated with the LiTS dataset drives innovation in medical imaging by promoting the development of machine learning models for precise liver and liver tumour segmentation [7]. This initiative enhances technology by providing diverse image sets, facilitating collaborative learning among researchers, and addressing clinical needs through improved diagnostic and surgical planning. It also ensures that the models developed are robust and applicable to the variable conditions found in real-world medical data, enhancing their practical utility in clinical settings.

The LiTS dataset consists of a wide spectrum of liver diseases and stages, which manifest as tumours varying significantly in size and the extent of imaging artefacts [7]. Liver tumours span a wide range of shapes, and sizes and contrast with ambiguous boundaries. There are also differences in the uptake of contrast agents which can introduce variability in the imaging of the tumours. Each image in the LiTS dataset has been meticulously annotated by radiologists. They marked every pixel as either tumour, healthy liver, or background, the latter representing pixels that are neither healthy liver tissue nor tumour. The characterises of the LiTS dataset are summarised in Table 2.2.

Table 2.2: Characteristics of LiTS Dataset

Characteristic	**Range**
Volumes	201
Volumes with tumour	194
Volumes annotated	131
Slice resolution in pixels	512 x 512
Axial slices per volume	42 - 1026
Number of tumours per volume	0 - 12
Slice Thickness per volume	0.45 mm - 6.0 mm
Tumour volume	38 mm^3 - 1231 mm^3
In-plane resolution per pixel	0.56 mm - 1.0 mm
Average tumor HU- value	65 - 59

2.4 Deep Learning

Before introducing different network architectures, it is important first to explain what deep learning is, as well as the basic concepts associated with it. Machine learning is an AI that can automatically learn from data and past experiences to identify patterns and make predictions with little human intervention [25]. Deep learning is a subset of machine learning, which utilises big networks and large amounts of data [26]. For a depiction of how deep learning relates to machine learning and AI, see Figure 2.1.

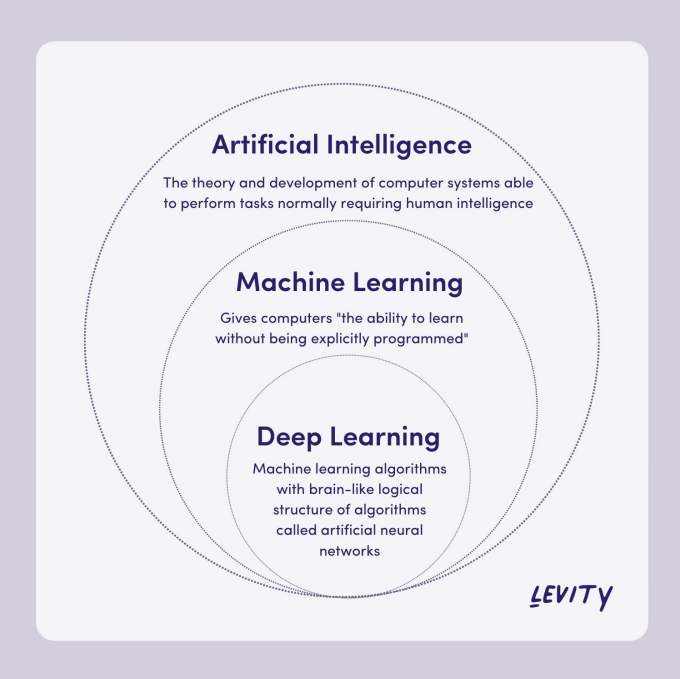


Figure 2.1: Deep learning is a subset of machine learning which in turn, is a subset of AI [1].

Deep learning uses multi-layered neural networks known as deep neural networks, to learn patterns [26]. The idea behind it is to have a significant number of connections to simulate the complexity of the human brain. It is trained on large amounts of data and can identify and classify phenomena, recognise patterns and relationships, evaluate possibilities, and make predictions and decisions. A deep neural network consists of at least three layers, any extra layers help refine and optimise the accuracy of the outcome.

2.4.1 Transfer Learning

In deep learning, it is common that a model starts off with no prior knowledge and learn from the provided dataset by training on it [27]. An advancement to this is transfer learning, where the network is pre-trained on a separate source domain,

which may vary depending on the model. This, in turn, means that the model possesses prior knowledge that it can apply on the target source, the chosen dataset. Transfer learning can enhance the model's performance and make it more accurate, particularly if the source and target sources are similar. However, it can also worsen the model's performance if the domains are too different, compared to the model being trained from scratch.

The benefits of using Transfer learning are not only that it can make the model more accurate, but it can also reduce the amount of data needed to train and speed up the processing time [28].

2.4.2 Basic Concepts of Deep Learning

Most machine learning algorithms rely on hyperparameters, these are predetermined settings external to the learning process itself [29]. At the core of these algorithms lies an optimisation algorithm, a loss function, a model architecture, and a dataset.

In a convolutional neural network (CNN) architecture, several layers play pivotal roles in feature extraction, dimensional reduction, classification, and regularisation [30]. Convolution layers extract essential features from input images using convolution operations, preserving spatial relationships crucial for accurate analysis. Pooling layers follow convolution layers, pooling layers reduce feature map size, aiding in computation efficiency while retaining important features. Fully connected layers connect neurons across layers, facilitating classification tasks and enhancing model performance through multi-layer interactions. To prevent overfitting, dropout layers randomly deactivate neurons during training, simplifying the network and improving generalisation. Activation functions introduce non-linearity, enabling the network to learn complex relationships between variables. Functions like ReLU, Softmax, and Sigmoid determine neuron activation and the significance of input data for prediction. Each of these layers and techniques contributes uniquely to the effectiveness and efficiency of CNNs in various machine-learning tasks.

The network learns through training, where the task is to optimise the initial network weights [31]. The goal is to find accurate weights which map the input to the correct output. The network weights are set to arbitrary values when the model is initialised, but with each epoch, the network will calculate the loss function. The loss function computes the disparity between the network's output predictions and the true labels associated with the input data, producing a quantitative measure of prediction accuracy. The calculated loss is then multiplied by a learning rate, a small number usually ranging from 0.01 to 0.0001, but can vary greatly. The new weight is calculated using Equation 2.2.

$$\text{new weight} = \text{old weight} - (\text{learning rate} \times \text{loss}) \tag{2.2}$$

Once the new weights are calculated, all old weights are replaced by the new ones for the next epoch. The optimisation algorithm plays a pivotal role in deep learning models, aiming to minimise the loss [32]. Gradient descent, a prevalent optimisation

algorithm, iteratively finds local minima of the loss function by computing its gradient and adjusting the model's parameters in the opposite direction of the gradient. An example of this is Adam, Adaptive Moment Estimation, a popular optimisation algorithm for first-order gradient-based optimisation of stochastic objective functions [33]. Training occurs over epochs, with each epoch representing one complete iteration through the entire training dataset [34].

To expedite training, images are typically divided into batches, and processed iteratively within each epoch. Batch size refers to the number of samples used in one forward and backward pass through the network [35]. This poses a trade-off between accuracy, speed and available computational power. Throughout the epoch, the model gathers information to update its current weights, enhancing its predictive capabilities for subsequent iterations [34].

The dataset is usually divided into three sets, train, test and validation set, where the train set is the largest [36]. As the name suggests, the model uses the training set to learn and is evaluated throughout the training process on the validation set. The lack of further improvement on the validation set is a good indicator that the number of optimal epochs has been reached.

Overfitting is a common issue in machine learning where a model learns the training data too well, capturing noise and specific patterns that do not generalise to new data [37]. Instead of learning the general characteristics of the data, the model memorises the training examples, leading to poor performance on unseen data. This is especially problematic in deep learning, where complex models can easily learn intricate details that are not relevant for making predictions on new data. A method to prevent this from happening is data augmentation. This meant to randomly augment the images and their corresponding labels in predecided ways to simulate more data.

Freezing layers in a neural network refers to fixing the weights of certain layers during training, preventing them from being updated [38]. This technique is commonly used in transfer learning, where pre-trained models are used for specific tasks. By freezing early layers that capture generic features and only training later layers, computational resources are conserved, and overfitting is mitigated. Freezing layers enables the efficient adaptation of pre-trained models to new tasks with limited labelled data. You can also fine-tune the model by unfreezing all layers, re-training it and lowering the learning rate, to adapt it more to your chosen data.

2.4.3 U-Net

U-Net is a fully convolutional neural network architecture seen in Figure 2.2 used for image segmentation [2]. U-Net has revolutionised biomedical image analysis by providing a reliable and efficient method for segmenting complex structures in medical images. Its ability to handle small datasets and produce high-quality segmentations has made it a standard tool in many medical imaging applications, contributing to advancements in diagnostic accuracy and treatment planning.

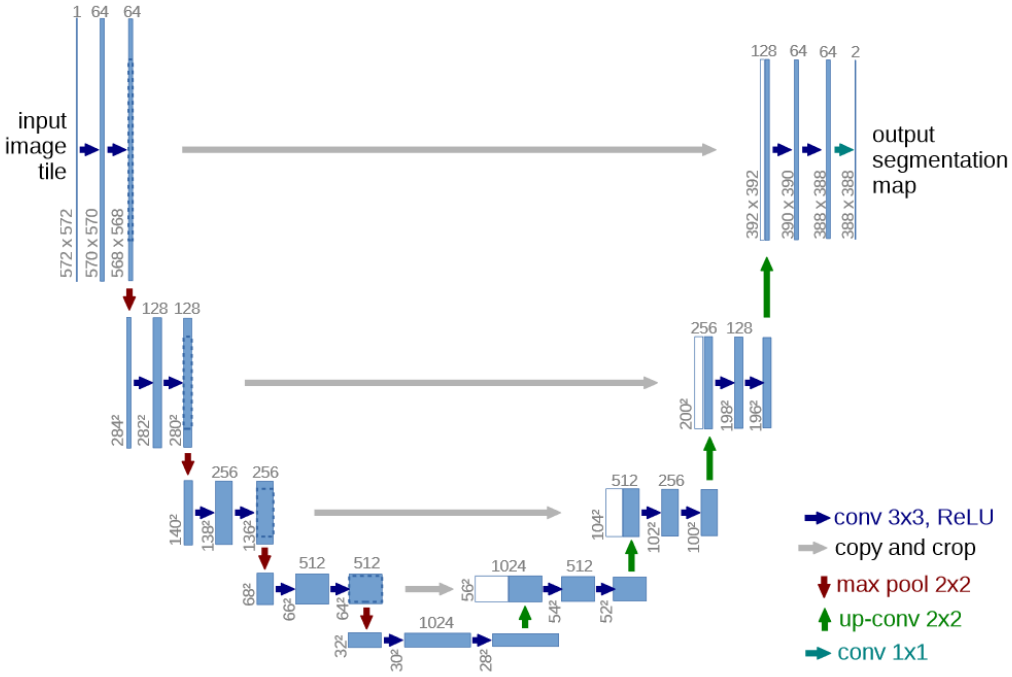


Figure 2.2: The network architecture of a U-Net, showing the encoding and decoding layers [2]

The U-Net structure consists of an encoder- and decoder path, see Figure 2.2 [39]. The encoder captures the context and reduces the spatial dimension, extracting features at multiple levels. In each downsampling step, the image dimensions are halved and the number of feature channels doubled. The main purpose of this is to extract as many features as possible. The resulting compressed image from the encoder path is passed through a bottleneck layer and then continues to the decoder part on the right side, see Figure 2.2. Here the compressing is reversed, the image dimensions are doubled and the feature channels are halved in each layer. Moreover, the decoder involves up-convolutions to increase the spatial dimensions, combined with concatenating features from the encoder path through skip connections. This enables precise localisation and segmentation of the image.

2.4.3.1 VGG16

VGG16, developed by K. Simonyan and A. Zisserman from the University of Oxford, was released in 2014 as a groundbreaking CNN [40]. This model has shown exceptional performance in image classification tasks, particularly notable for its test data accuracy of 92.77% when trained on the ImageNet dataset. ImageNet is a dataset that consists of 14 million images and 1000 classes.
The VGG16 architecture is characterised by its depth and simplicity, comprising 16 layers, including 13 convolutional layers, 5 pooling layers, and 3 fully connected layers, as seen in Figure 2.3 [41]. The network expects an input image of a fixed size of 224x224x3. It employs filters that progressively increase from 64 to 512 through

the network. Each convolutional layer uses a 3x3 filter with a stride of 1, while the max pooling layers use a 2x2 filter with a stride of 2. Due to its robust structure, VGG16 has been widely used as a backbone for various image classification and feature extraction tasks. It is used in applications requiring detailed visual understanding, such as medical image analysis. The network's ability to act as a feature extractor allows it to be fine-tuned for specific tasks, reducing the computational complexity and resource requirements.

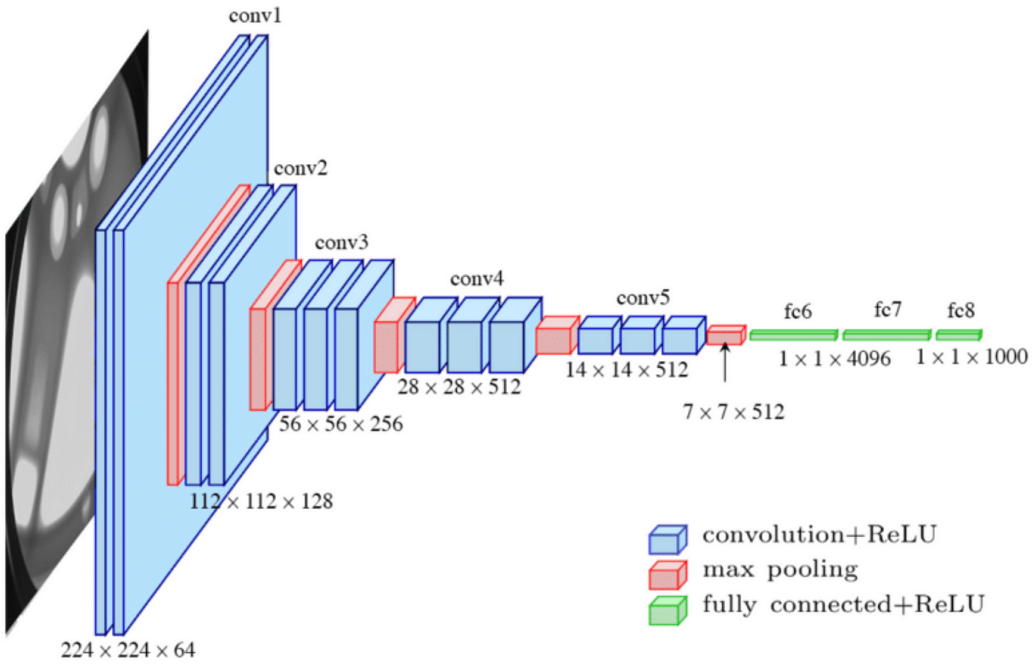


Figure 2.3: The network architecture of VGG16 [3].

The hierarchical structure of VGG16 enables the model to learn robust feature representations, making it suitable for a broad range of image classification tasks [42, 41]. Its extensive use across different domains underscores its versatility and effectiveness.VGG16 remains a cornerstone in the field of deep learning for image processing, illustrating the power of CNN architectures in handling complex visual tasks. Its continued relevance is supported by ongoing research and adaptation in various scientific and industrial applications.

2.4.3.2 YOLOv8

YOLO (You Only Look Once) is an object detection and image segmentation model, which utilises transfer learning [4]. The model became popular due to its exceptional high speed and accuracy in various computer vision applications. The YOLOv8 network was originally presented by Redmon et al. in 2015. It has since then been developed further in 7 versions into YOLOv8, which builds on previous versions and was released in January 2023. This version supports new features and improvements for enhanced performance, flexibility, and efficiency. The segmentation version of YOLOv8 is pre-trained on a dataset called COCO (Common Objects in Context)

[4]. The COCO dataset is an object detection, segmentation and captioning dataset [43]. It is trained 330k images, over 200k of which are labelled, on 80 objects and 91 stuff categories. This diverse dataset enables YOLO to achieve remarkable accuracy across diverse tasks.

The structure of the YOLOv8 network is presented in Figure 2.4, this includes the division of backbone and head. The backbone of the network is made up of a CNN, which plays a crucial role in extracting low, medium and high-level feature maps from an input image [44]. This multi-scale extraction is an important feature for detecting objects of various sizes. These feature maps are then relayed to the head of the structure which then makes the classification prediction of each pixel for segmentation. YOLOv8 operates like a 'black box', where you provide input and commands and receive an output. Making modifications inside the YOLOv8 architecture is therefore very hard to do.

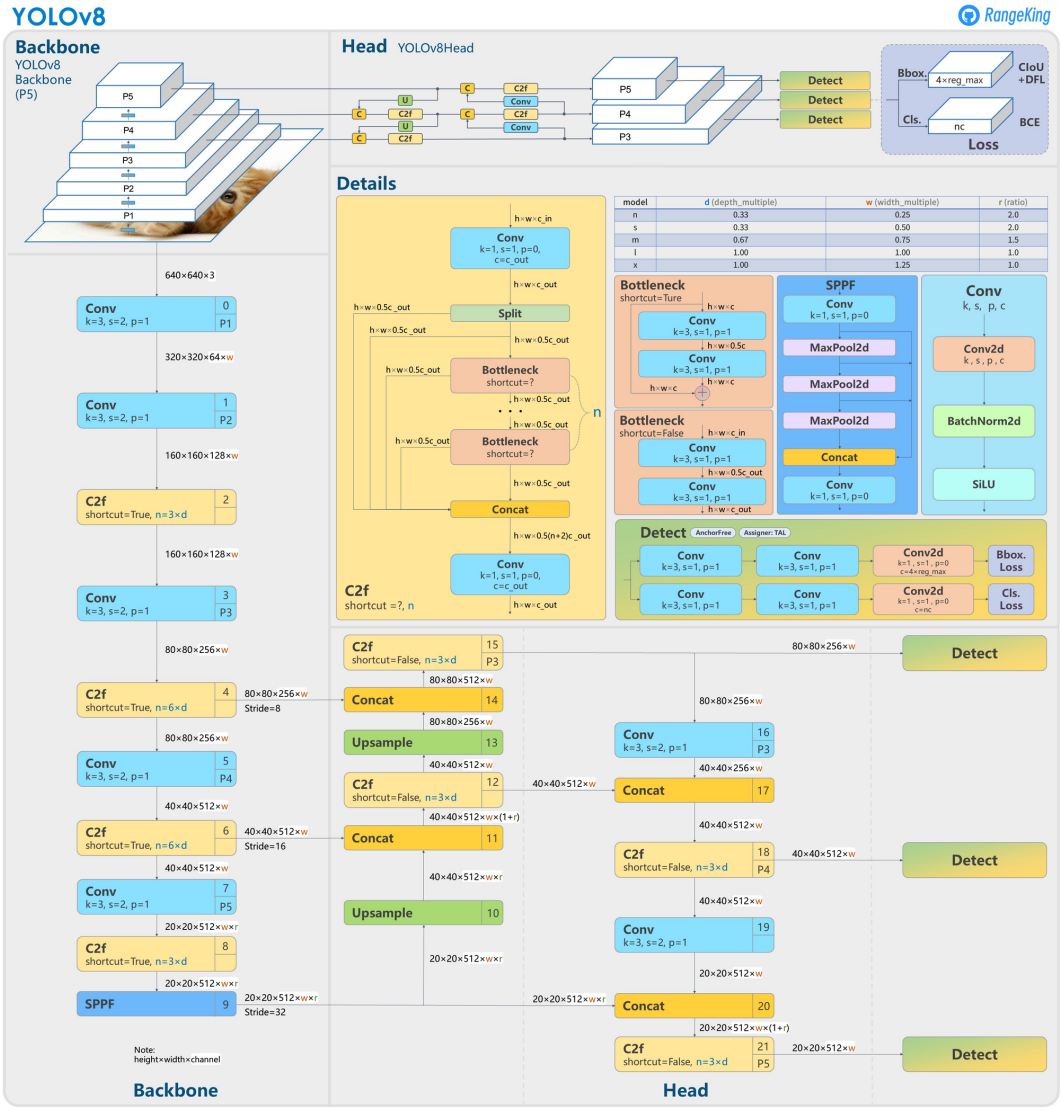


Figure 2.4: YOLOv8 architecture [4].

There are different sizes of YOLOv8, which can be selected depending on the task at hand [5]. There are five variants, nano (n), small (s), medium (m), large (l) and extra large (x). The difference between them is primarily the number of parameters, which in turn affect other properties like speed, this can be seen in Figure 2.5.

Model	size (pixels)	mAP^{box} 50-95	mAP^{mask} 50-95	Speed CPU ONNX (ms)	Speed A100 TensorRT (ms)	params (M)	FLOPs (B)
YOLOv8n-seg	640	36.7	30.5	96.1	1.21	3.4	12.6
YOLOv8s-seg	640	44.6	36.8	155.7	1.47	11.8	42.6
YOLOv8m-seg	640	49.9	40.8	317.0	2.18	27.3	110.2
YOLOv8l-seg	640	52.3	42.6	572.4	2.79	46.0	220.5
YOLOv8x-seg	640	53.4	43.4	712.1	4.02	71.8	344.1

Figure 2.5: YOLOv8 sizes and its corresponding properties [5].

YOLOv8, one of the latest networks in the YOLO series of object detection models, incorporates several innovative enhancements to boost its performance [45]. This technique combines four different images into one composite image to provide the model with a richer context during training. In YOLOv8, this augmentation strategy is notably deactivated during the last 10 training epochs, which is a deviation from its predecessors like YOLOv4, where it was used consistently. The rationale behind this adjustment is to fine-tune the model's performance by focusing on learning from singular, unaltered images towards the end of training but is something that can be turned off altogether. Another property is loss function optimisation, YOLOv8 innovates on the traditional loss function approach by introducing a task alignment score. This score is pivotal in enhancing the model's detection capabilities as it is calculated from classification accuracy and the intersection over union (IoU) metric. The model utilises a combination of binary cross entropy for accurate label prediction, complete IoU for precise bounding box alignment with ground truth, and distributional focal loss to refine the predicted bounding box distribution. Each of these components serves a distinct purpose, contributing to the model's robustness in both classifying and localising objects within an image. Something else that differs in this network compared to others is the labels, the format used for YOLOv8 is a text file. It is one text file per image where each object is a separate row, where an integer represents the class of the object [46].

2.4.3.3 SAM

The Segment Anything Model (SAM) is a new TL segmentation model developed by Meta AI, released in April 2023 [47]. The model is built to identify the precise location of objects in an image and is trained on 1 billion masks on 11M images. SAM is developed to be used as is, meaning it does not necessarily need any additional training. It utilised bounding boxes as input to segment objects in images. However, it can be further trained on new datasets to customise it for specific tasks and datasets.

The architecture is fully supervised and is composed of the following three key components; a powerful image encoder, a prompt encoder, and a lightweight mask decoder, as shown in Figure 2.6 [6]. The image encoder, based on a pre-trained vision transformer, processes each image once and is adaptable to high-resolution inputs. The prompt encoder handles various prompt types, including sparse (points, boxes) and dense (masks), using different encoding techniques like positional encodings and convolutions. The mask decoder integrates these inputs using a transformer-based approach to efficiently generate segmentation masks, adaptable for zero-shot learning across diverse tasks and image types. SAM comes in three different sizes, "ViT-B", "ViT-L", and "ViT-H", each adding more complexity.

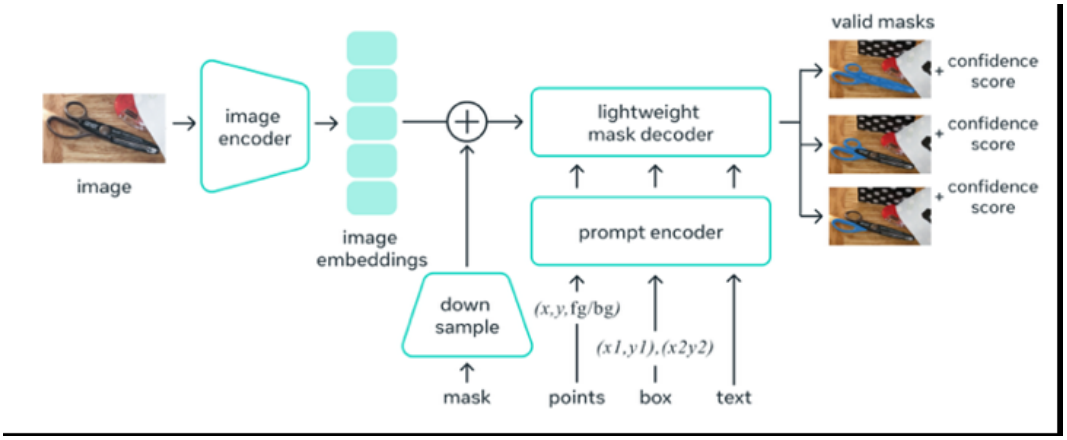


Figure 2.6: The architecture of SAM, including an image encoder, a prompt encoder, and a lightweight mask decoder, as well as the various prompts that can be given to the architecture. There is also an example input image and potential masks for it [6].

2.5 Evaluation metrics for liver segmentation

The evaluation metrics that are used for this project will be presented in this section. In this project, recall and Dice score were used to evaluate the performance of each model and determine the best one. A weighted score of recall and Dice score is also introduced.

2.5.1 Dice Score

Dice score is a evaluation metrics, it is a measure of the overlap between the predicted segmentation and the ground truth. The Dice coefficient or the Dice score, can be visualised in Figure 2.7, showing the segmentation and ground, as well as truth false positive (FP), false negative (FN) and the overlap between the two, true positive (TP).

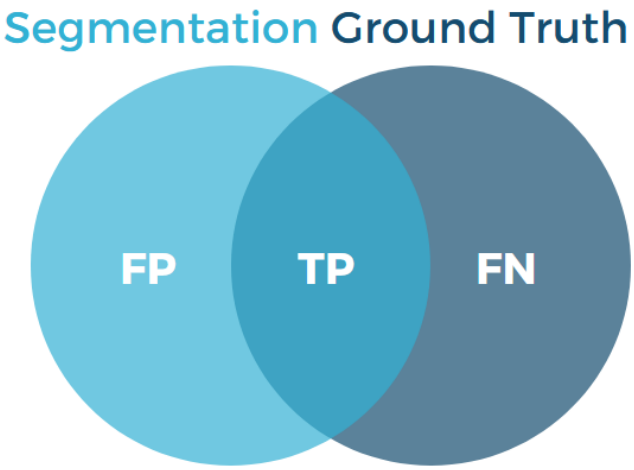


Figure 2.7: Light blue represent the predicted segmentation and the dark blue the true label or ground truth. FP are pixels wrongfully classified as tumour, FN are pixels wrongfully classified as non-tumour, and TP are pixels correctly classified as tumour.

Dice score is a measurement of the overlap between the segmentation and ground truth, and is defined in Equation 2.3. Where TP = true negative, FN = false negative and FP = false positive. When used in the evaluation, the Dice score is applied for every segmentation and then averaged over all cases [7]. The score ranges from 0, indicating no overlap, to 1, signifying perfect agreement between the segmentation and ground truth.

$$\text{Dice Coefficient} = \text{Dice Score} = \frac{2 \times TP}{2 \times TP + FP + FN} \tag{2.3}$$

It is critical for models that segment medical images to accurately identify areas of interest, such as tumours, and the Dice score provides an objective measure of the model's segmentation accuracy.

2.5.2 Recall (sensitivity)

Recall is another way to evaluate a model's performance on a dataset [7]. It is commonly used in segmentation tasks where it is more important to minimise FN than FP. Recall expresses the proportion of correctly identified TP in relation to FN, and the definition can be seen in Equation 2.4. Where TP = true negative, FN = false negative and FP = false positive. A score close to 1 in this case, indicates that the model identifies most of the ROI, which is important for medical purposes, but says nothing about including pixels that are FP.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{2.4}$$

2.6 Loss Function

The loss function is a method used to evaluate how well a network works on a specific dataset, and it is used during the training phase [48]. A higher value from the loss function indicates poor predictions from the model, and vice versa, where lower values indicate better predictions.

A loss function sometimes called a cost or error function, is a mathematical function that quantifies the difference between the predicted value (the output of the model) and the true value (the label) [49]. It is used in the training process to minimise the loss values through optimisation techniques like gradient descent. This value can be used as an indicator when changing parameters in the model to see if they improve performance, i.e. yield a lower training loss.

The choice of loss function should be made with care since the choice of loss function could have a significant impact on the learning process of the model and the results it produces [50]. It is important to choose a loss function suited to your specific dataset and task since different loss functions have different strengths. For example, if you have a lot of background (i.e., the number of non-tumour pixels is far greater than the number of tumour pixels), like in this project, Dice loss is a common choice.

2.6.1 Dice Loss

Dice loss is based on the Dice coefficient shown earlier in Equation 2.3. In image segmentation, the Dice coefficient compares pixels from the predicted mask with those of the ground truth [50]. Contrary to the previous aim of maximising the Dice coefficient and therefore the overlap, the goal here is to minimise the Dice loss formula seen in Equation 2.5. Here TP = true positive, FP = false positive and FN = false negative. Note that the Dice coefficient is slightly modified to ensure the function is defined in all scenarios, like when $TP + FP + FN = 0$.

$$\text{Dice Loss} = 1 - \text{Dice Coefficient} = 1 - \frac{2 \times TP + 1}{2 \times TP + FP + FN + 1} \tag{2.5}$$

2.6.2 Focal Tversky

Focal Tversky loss is a further adaptation of Dice loss [51]. A limitation of the Dice loss function is that it weighs false positive (FP) and false negative (FN) equally. This results in high precision and low recall, to combat this and improve recall, the Tversky index (TI) is presented as a generalisation of the Dice coefficient, shown in Equation 2.6, where TP = true negative, FN = false negative and FP = false positive. α and β are weights for FN and FP against one another. When $\alpha = \beta = 0.5$, the Tversky Index becomes the Dice coefficient, see Figure 2.3.

$$\text{Tversky Index} = \frac{TP}{TP + \alpha FN + \beta FP} \tag{2.6}$$

The TI is converted to a loss function called Tversky loss which is minimised and can be seen in Equation 2.7. Another weakness of the Dice loss is that small ROIs do not contribute significantly to the loss, hence the network struggles to segment them. A parameter γ is added to combat this, creating the final focal Tversky loss as seen in Equation 2.8.

$$\text{Tversky Loss} = (1 - \text{Tversky Index}) \tag{2.7}$$

$$\text{Focal Tversky Loss} = (1 - \text{Tversky Index})^{\gamma} \tag{2.8}$$

This focal Tversky loss function is customised for imbalanced datasets, where there is a significantly larger amount of background data compared to the area of interest, like in the case of this project. It also prioritises false positives (FP) over false negatives (FN) and focuses on harder examples of smaller tumours.

3
Methods

In this chapter, the methods of this project are presented. First, the general workflow of the entire project is summarised. Then, the preparation of the two datasets used throughout the project is described. Finally, the training and evaluation of the deep learning models are explained.

3.1 General Workflow

The work began by familiarisation with previous thesis work from Navari Surgical. This involved reading reports and other documentation, reviewing the code, and attempting to replicate the results. All decisions made during this process were noted for consideration in this new thesis work. A literature study was conducted to gain a better understanding of the problem and potential solutions. Based on this background research, the focus was placed on deep learning networks, specifically TL networks.

The TL networks chosen for further investigation were VGG16, YOLOv8, and SAM. These networks were selected based on their potential as highlighted in the literature. Additionally, the U-net network, which was used in previous years' thesis work, was included to facilitate easier comparison.

To ensure a fair comparison, the TL networks were worked on separately, determining settings through a literature review and a trial-and-error approach. Consistent parameters were maintained across all networks and individual runs to enable direct comparison of results.

It was also decided to create new datasets, inspired by previous thesis work but with improvements. Using the same raw files from the LiTS challenge, the original NIfTI files were reprocessed into PNG format. This allowed for control over all preprocessing steps and addressed certain issues found in previous datasets. The aim was to see if these modifications would positively impact the results. Consequently, two new datasets were created: one focusing on the liver (LIVER) and another simulating a bounding box or region of interest (ROI). During the preprocessing stage, some additional settings and modifications were applied.

The networks were evaluated using various metrics deemed appropriate for this task. The analysis was conducted in stages: first, different settings and networks

were compared individually on each dataset. Then, the networks were compared to each other to draw comprehensive conclusions.

3.2 Datasets

Two datasets were created from the LiTS dataset, named LIVER and ROI. These datasets were designed for use in two different scenarios. The LIVER dataset is intended for direct use, while the ROI dataset is meant to simulate the surgeons' marked regions of interest in the images. Both datasets have been prepared and cropped accordingly.

The datasets were split into training, validation, and test sets in the same way for both LIVER and ROI. A script was used to divide the datasets into 80% for training, 10% for validation, and 10% for testing. The split was based on the total number of images, not the number of patients. The script ensured that images from the same patient would not be placed in different sets. The datasets divided into train, validation and test sets are summarised in Table 3.1.

Table 3.1: The amount of images and respective labels divided into training, validation, and test sets for the two different datasets, LIVER and ROI.

Dataset	Train	Validation	Test
LIVER	10845	1106	1226
ROI	4569	392	490

3.2.1 Liver

The LIVER dataset was cropped to focus on the liver, ensuring the images contained as much of the liver as possible without including excessive background, which could negatively affect the output. Labels with and without liver tumours were used in this dataset to more accurately mimic reality and test the robustness of our models. However, 50% of all the black labels and corresponding images were removed to create a more balanced dataset, as non-tumour labels were overrepresented in the original dataset. In total, 13,177 slices of images and corresponding labels were obtained from the LiTS dataset. The dataset was split into training, validation, and test sets, with the training set consisting of 10,845 images, the validation set of 1,106 images, and the test set of 1,226 images. This results in 82.50% in the training set, 8.63% in the validation set, and 8.79% in the test set. This distribution includes images from 107 patients in the training set and 12 patients in each of the validation and test sets. Before removing 50% of the images not containing a tumour, the dataset consisted of 19,163 images and corresponding labels.

3.2.2 Region of Interest

The ROI dataset consisted of cropped images of tumours. In this dataset, all labels represented tumours. To train the network on a specific task, it was decided to limit the training to images containing only one tumour. This resulted in a total of 5,451 images and corresponding labels. These images and labels were then split into training, validation, and test sets, with the training set consisting of 4,569 images, the validation set of 392 images, and the test set of 490 images. This results in 83.82% in the training set, 7.19% in the validation set, and 8.99% in the test set. This distribution includes images from 107 patients in the training set and 12 patients in each of the validation and test sets.

3.3 Preparation of dataset

The LiTS dataset was sourced from the organisers of the LiTS Challenge, available at medicaldecathlon.com. The dataset was systematically organised into patient-specific folders, each containing image files and corresponding labels in NIfTI format, compressed for efficient transfer. Decompression was performed using 7-zip, followed by the conversion of the NIfTI files to DICOM format using Slicer. This conversion process was done individually for each patient, including a thorough visual inspection of both images and labels to ensure data integrity.

Special attention was given to the segmentation of the liver from the overall anatomical structures within the dataset. Precise segmentation was possible due to specific annotations identifying liver tissue, tumours, and background elements. The liver was isolated using annotations for liver and tumour as input images, while the tumour was isolated using annotations for tumours as labels. This ensured that each image and its corresponding label were consistently linked, enabling the networks to train effectively on segmenting tumours from liver images.

3.3.1 Preprocessing

Preprocessing was a crucial part of the dataset preparation, consisting of cropping, HU conversion, and histogram equalisation. HU conversion and histogram equalisation were applied uniformly across both datasets, while cropping differed between them. Preprocessing was necessary to convert all images and labels from DICOM to PNG format, the required input for network training.

3.3.1.1 Cropping

For the LIVER dataset, images were cropped along the liver, using the annotated liver as a reference. Corresponding labels were cropped identically to ensure alignment with their respective images. For the ROI dataset, images were cropped to isolate a single tumour. To avoid always having the tumour in the centre of the image, the cropping was randomised, extending from 0 to 30 pixels beyond the tumour's size.

3.3.1.2 HU Conversion

The conversion to Hounsfield Units (HU) was carried out to rescale the intensity values, standardising them across all images. This was achieved using Equation 2.1. However, as the constants were not provided in the metadata no differences were made by the conversion. Additionally, no thresholding of HU values was applied as it was not deemed necessary.

3.3.1.3 Histogram Equalisation

Histogram equalisation was applied separately to each image in both the training and validation sets to enhance contrast, using Python's OpenCV library. This process was not applied to the labels, nor was it applied to the test set.

3.3.1.4 Data Augmentation

To prevent overfitting, various data augmentations were randomly applied during the training process. The chosen augmentations—rotation, scaling, and shearing—are detailed in Table 3.2. These were applied consistently to both images and their corresponding masks to maintain mask accuracy. This strategy, based on successful practices from the previous year's thesis, effectively doubled the dataset size and enhanced the model's ability to generalise by retaining all original images and masks.

Table 3.2: Data augmentation used on the datasets.

Augmentation	Span
Rotation	-180 to 180
Shear	- 15 to 15
Scale	0.5 to 1.5

3.4 Training

Training was conducted on U-Net, VGG16, and YOLOv8 models using customised training loops. To maintain consistency, the majority of variables were kept constant across all networks, including the number of epochs, learning rates for layer freezing and unfreezing, and the optimiser. Variables like image size and batch size were adjusted specifically for each network. Each network was trained with two datasets and for U-Net and VGG16, the loss functions varied while for YOLOv8 it was kept the same. This resulted in 12 model variations each for U-Net and VGG16, and 6 for YOLOv8, due to its fixed loss function, resulting in a total of 30 models.

3.4.1 The assembly of VGG16

VGG16 consist of a head and a backbone. The backbone is the same as the original network architect described in section 2.4.3.1. As the layers in the backbone use max-pooling to compress the image, the head consists of upsampling layers to obtain the same size as the input image. The head in this case has not been trained

before.

```
self.upsample = nn.Sequential(
    nn.ConvTranspose2d(512, 256, kernel_size=2, stride=2),
    nn.BatchNorm2d(256),
    nn.ReLU(inplace=True),
    nn.Dropout(p=0.5),
    nn.ConvTranspose2d(256, 128, kernel_size=2, stride=2),
    nn.BatchNorm2d(128),
    nn.ReLU(inplace=True),
    nn.Dropout(p=0.5),
    nn.ConvTranspose2d(128, 64, kernel_size=2, stride=2),
    nn.BatchNorm2d(64),
    nn.ReLU(inplace=True),
    nn.Dropout(p=0.5),
    nn.Conv2d(64, 2, kernel_size=1)
)
```

3.4.2 The assembly of YOLOv8

YOLOv8 requires slightly different input in terms of label format. Unlike the other networks that were trained, YOLOv8 requires a text file as a label. This text file is generated from the masks that served as input for the other networks. However, the output of the prediction is a mask, similar to the outputs of the other networks.

3.4.3 Model Training Configurations

The training involved 30 models, integrating two datasets with various combinations of learning rates for layer freezing and unfreezing. The learning rates for layer freezing were A, B and C and unfreezing C, D and E, as can be seen in Table 3.3. For U-Net and VGG16, two different loss functions were also tested, Dice loss and focal Tversky loss. Key parameters such as the number of epochs, the optimiser Adam, and the two datasets were standardised across all networks. Differences between the networks involved customising the image and batch size for each specific model, depending on input format and GPU capacity, as shown in Table 3.4.

The initial training phase for the TL networks involved freezing the backbone at a predetermined rate, followed by unfreezing all layers and reducing the learning rate (fine-tuning), as shown in Table 3.3. Unlike these networks, U-Net, which is not a TL network, only utilises the first learning rate for comparison purposes, as it does not involve freezing or unfreezing layers. The loss functions used were Dice loss and focal Tversky loss for U-Net and VGG16, while YOLOv8 used a fixed combination of built-in loss functions. Training was conducted using the training subset of the dataset, and performance was monitored using the validation subset. Throughout the training, progress was tracked by observing the plots of the training and validation losses.

Table 3.3: Showing the training configuration for each model, including model number, network, dataset, learning rate for freeze and unfreeze as well as loss function where applicable.

Model nr	Network	Dataset	Lr Freeze	Lr Unfreeze	Lr Combination	Loss Function
1	U-Net	LIVER	A	-	1	Dice
2	U-Net	LIVER	B	-	2	Dice
3	U-Net	LIVER	C	-	3	Dice
4	U-Net	LIVER	A	-	1	Focal Tversky
5	U-Net	LIVER	B	-	2	Focal Tversky
6	U-Net	LIVER	C	-	3	Focal Tversky
7	U-Net	ROI	A	-	1	Dice
8	U-Net	ROI	B	-	2	Dice
9	U-Net	ROI	C	-	3	Dice
10	U-Net	ROI	A	-	1	Focal Tversky
11	U-Net	ROI	B	-	2	Focal Tversky
12	U-Net	ROI	C	-	3	Focal Tversky
13	VGG16	LIVER	A	C	1	Dice
14	VGG16	LIVER	B	D	2	Dice
15	VGG16	LIVER	C	E	3	Dice
16	VGG16	LIVER	A	C	1	Focal Tversky
17	VGG16	LIVER	B	D	2	Focal Tversky
18	VGG16	LIVER	C	E	3	Focal Tversky
19	VGG16	ROI	A	C	1	Dice
20	VGG16	ROI	B	D	2	Dice
21	VGG16	ROI	C	E	3	Dice
22	VGG16	ROI	A	C	1	Focal Tversky
23	VGG16	ROI	B	D	2	Focal Tversky
24	VGG16	ROI	C	E	3	Focal Tversky
25	YOLOv8	LIVER	A	C	1	Built-in
26	YOLOv8	LIVER	B	D	2	Built-in
27	YOLOv8	LIVER	C	E	3	Built-in
28	YOLOv8	ROI	A	C	1	Built-in
29	YOLOv8	ROI	B	D	2	Built-in
30	YOLOv8	ROI	C	E	3	Built-in

Table 3.4: Training configurations for the different networks, including image and batch size. Specific configurations vary according to each network's input format and GPU capacity.

Network	Image Size	Batch Size
U-Net	256	32
VGG16	244	32
YOLOv8	256	16

3.4.4 Saving the Best Model

During the training process, only the weights from the best model of each phase were saved. The selection of the "best model" is based upon the validation loss during training, essentially the epoch where it reaches its minimum value for that run. The training of the TL models consists of two phases: initially, the backbone is frozen, and run for I epochs. However, only the best model from this phase is carried over to the second phase, where all layers are unfrozen and the learning rate is reduced for II epochs, the same procedure applies here where only the best model is saved and is then the one used to predict tumours.

3.4.5 SAM

Unlike other TL networks, the SAM network is designed to be used immediately, without the need for additional training on a custom dataset. An attempt was made to train SAM similarly to other networks to improve its accuracy as much as possible. During this process, it became evident from the training and validation loss data that the network was not learning or improving with the dataset. Consequently, the decision was made to include SAM's untrained version in the project for comparison, rather than excluding it entirely. Both datasets were applied to the SAM network; for the LIVER dataset, bounding boxes were generated from the masks, with random pixel lengths varying between 0-10 in model 31 and 0-20 in model 32. The ROI dataset was used as it is, suitable for SAM's requirement for bounding box inputs, which this dataset replicates, creating model 33.

3.5 Evaluation

To evaluate the performance of each model, predicted masks were generated using the final saved model weights. Approximately 10% of the dataset, designated as the test split, was used to input images into the model, which then predicted each pixel as either tumour or background. The output was a predicted mask, which was compared to the original mask of the corresponding image. We employed evaluation metrics Dice score and recall, where each image receives a score between 0 and 1. The average of these scores for all predictions determines the performance of each model.

3.5.1 Recall

Recall primarily focuses on the TP, assigning a higher value when the proportion of TP is high, as indicated in Equation 2.4. This metric is particularly relevant to our task, where correctly classifying pixels as tumour, high TP and low FN, is critical. However, applying this metric to our LIVER dataset presented challenges with images lacking tumours, as these do not have TP or FP, leading to potential errors. To address this, a special treatment was implemented for images which do not contain a tumour. If the model in that case correctly predicts the absence of

a tumour, the image receives a recall score of 1; if it incorrectly predicts a tumour, the score is 0. This approach penalises false positives effectively.

3.5.2 Dice Score

Dice score is another evaluation metric that was used, which provides a broader image of the model's performance. In accordance with Equation 2.3, the Dice score includes TP, TN, FP and FN, which gives it an overview of the model's overall performance when predicting.

3.5.3 Weighted Score

The weighted score is based on recall from Section 3.5.1 and Dice score from Section 3.5.2. The weighted score is calculated according to equation 3.1 where the evaluation metrics weigh Dice score higher than recall.

$$\text{Weighted Score} = 0.6 \cdot \text{Dice loss} + 0.4 \cdot \text{Recall} \tag{3.1}$$

4
Results

In this chapter, the results from each model described in the training schedule in Section 3.4.3 are presented. As detailed in Section 3.5, the models were evaluated by predicting tumours on the test sets from both datasets using Dice score and recall. From the evaluation scores, a weighted score is calculated. The results are first presented based on the type of network, then collectively to provide an overview, including the average and variation of the evaluation metrics. The learning rate combinations are presented in Table 4.1 for future reference.

Table 4.1: Learning rate combinations used for transfer learning networks. *Learning Rate Freeze* refers to the learning rate applied when the backbone is frozen, while *Learning Rate Unfreeze* refers to the learning rate used when the backbone is unfrozen.

Learning Rate combination	Learning Rate Freeze	Learning Rate Unfreeze
1	A	C
2	B	D
3	C	E

4.1 U-Net

U-Net architecture was used for both the LIVER and ROI datasets, which are presented in Section 4.1.1 for LIVER and Section 4.1.2 for ROI. The U-Net had different settings, resulting in six models for LIVER, summarised in Table 4.2, and six models for ROI, summarised in Table 4.3. All U-Net models were run for I+II epochs, and the best model, determined by the lowest validation loss during training, was the one saved.

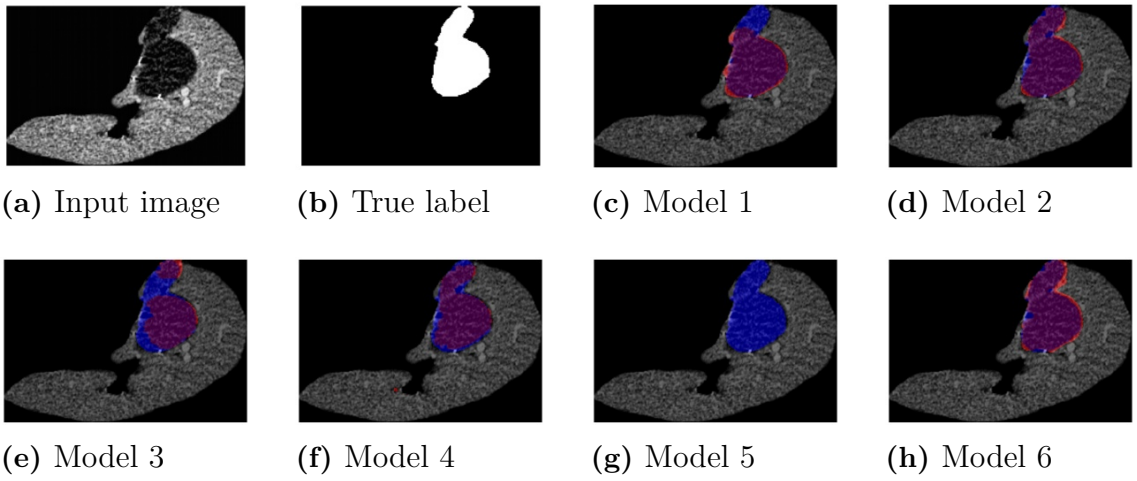
4.1.1 LIVER

The performance of the different U-Net models is summarised in Table 4.2 for the LIVER dataset. Two different loss functions were used for the LIVER dataset: Dice loss and focal Tversky loss, in combination with different learning rates. Model 3 obtained the highest evaluation scores for the U-Net architecture trained on the LIVER dataset.

Table 4.2: Overview of U-Net models for the LIVER dataset running for I+II epochs each where the best models according to the lowest validation loss are saved. The evaluation scores, Dice score, recall and weighted score, are also presented for each model.

Model nr	Learning Rate	Loss Function	Dice Score	Recall	Weighted Score
1	A	Dice Loss	0.699	0.704	0.701
2	B	Dice Loss	0.734	0.738	0.736
3	C	Dice Loss	0.740	0.742	0.741
4	A	Focal Tversky	0.721	0.735	0.727
5	B	Focal Tversky	0.698	0.720	0.707
6	C	Focal Tversky	0.718	0.737	0.726

As seen in Figure 4.1, an example image from the test set has been evaluated on model 1 - model 6.



(a) Input image **(b)** True label **(c)** Model 1 **(d)** Model 2

(e) Model 3 **(f)** Model 4 **(g)** Model 5 **(h)** Model 6

Figure 4.1: An example image from the test set evaluated on model 1- 6. Purple indicates overlap, red indicates false positives, and blue indicates false negatives.

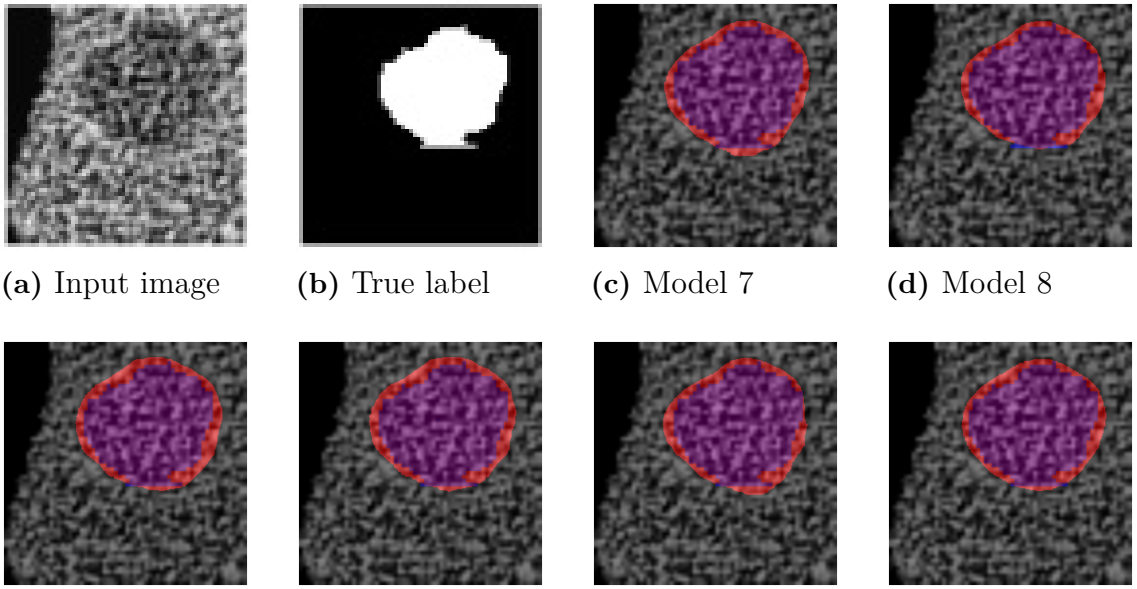
4.1.2 ROI

The performance of the different U-Net models is summarised in Table 4.3 for the ROI dataset. Two different loss functions were used for the ROI dataset: Dice loss and focal Tversky loss, in combination with different learning rates. Model 11 obtained the highest weighted score for the U-Net architecture trained on the ROI dataset.

Table 4.3: Overview of U-Net models for the ROI dataset running for I+II epochs each where the best models according to the lowest validation loss are saved. The evaluation scores, Dice score, recall and weighted score, are also presented for each model.

Model nr	Learning Rate	Loss Function	Dice Score	Recall	Weighted Score
7	A	Dice Loss	0.670	0.685	0.676
8	B	Dice Loss	0.704	0.771	0.731
9	C	Dice Loss	0.717	0.760	0.734
10	A	Focal Tversky	0.708	0.798	0.744
11	B	Focal Tversky	0.702	0.831	0.754
12	C	Focal Tversky	0.716	0.792	0.746

As seen in Figure 4.2, an example image from the test set has been evaluated on model 7 - model 12.



(a) Input image **(b)** True label **(c)** Model 7 **(d)** Model 8

(e) Model 9 **(f)** Model 10 **(g)** Model 11 **(h)** Model 12

Figure 4.2: An example image from the test set evaluated on model 7- 12. Purple indicates overlap, red indicates false positives, and blue indicates false negatives.

4.2 VGG16

The VGG16 architecture was used for both the LIVER and ROI datasets, which are presented in Section 4.2.1 for LIVER and Section 4.2.2 for ROI. The VGG16 had different settings, resulting in six models for LIVER, summarised in Table 4.4, and six models for ROI, summarised in Table 4.5. All VGG16 models were run for a total of I+II epochs. The VGG16 backbone was frozen for the first I epochs, during which only the head of the VGG16 was trained. Subsequently, the backbone was unfrozen and trained for an additional II epochs with a reduced learning rate for

fine-tuning. These learning rate combinations are presented in Table 4.1. The best model, determined by the lowest validation loss during training, was saved during the I+II epochs for each model.

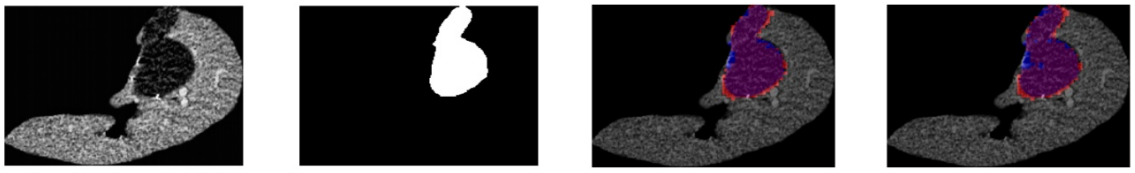
4.2.1 LIVER

The performance of the different VGG16 models is summarised in Table 4.4 for the LIVER dataset. Two different loss functions were used for the LIVER dataset: Dice loss and focal Tversky loss, in combination with different learning rates. Model 13 obtained the highest evaluation scores for the VGG16 architecture trained on the LIVER dataset.

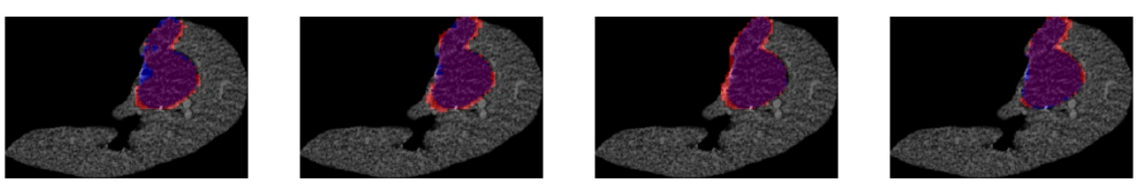
Table 4.4: Overview of VGG16 models for the LIVER dataset running for I+II epochs each where the best models according to the lowest validation loss are saved. The evaluation scores, Dice score, recall and weighted score, are also presented for each model.

Model nr	Learning Rate	Loss Function	Dice Score	Recall	Weighted Score
13	1	Dice Loss	0.712	0.738	0.722
14	2	Dice Loss	0.592	0.623	0.604
15	3	Dice Loss	0.619	0.638	0.627
16	1	Focal Tversky	0.667	0.688	0.675
17	2	Focal Tversky	0.592	0.624	0.605
18	3	Focal Tversky	0.557	0.607	0.589

As seen in Figure 4.3, an example image from the test set has been evaluated on model 13 - model 18.



(a) Input image **(b)** True label **(c)** Model 13 **(d)** Model 14



(e) Model 15 **(f)** Model 16 **(g)** Model 17 **(h)** Model 18

Figure 4.3: An example image from the test set evaluated on model 13- 18 . Purple indicates overlap, red indicates false positives, and blue indicates false negatives.

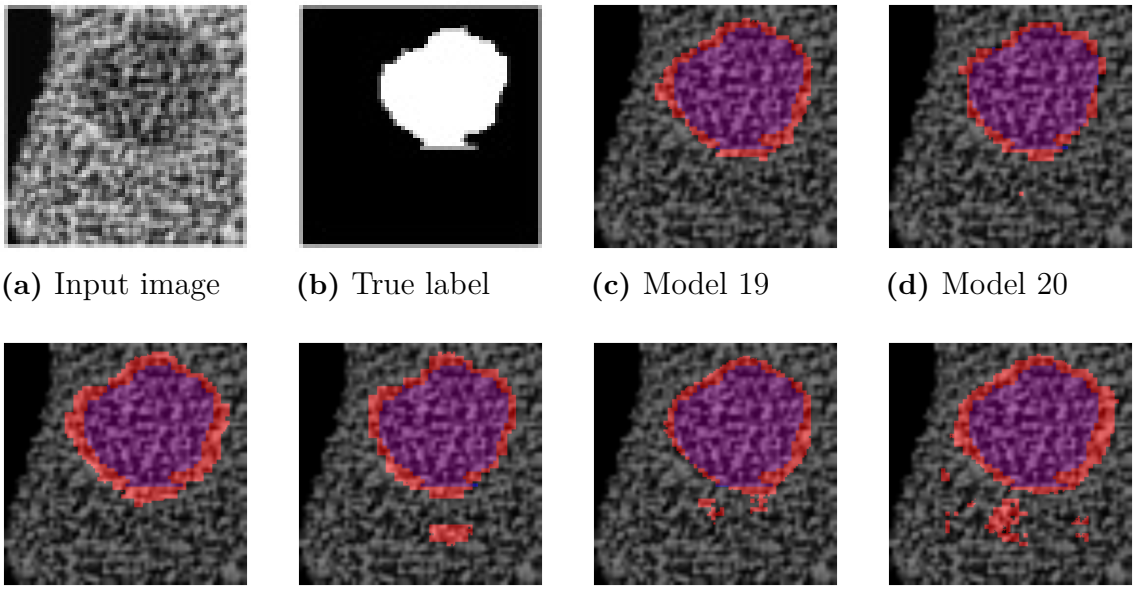
4.2.2 ROI

The performance of the different VGG16 models is summarised in Table 4.5 for the ROI dataset. Two different loss functions were used for the ROI dataset: Dice loss and focal Tversky loss, in combination with different learning rates. Model 11 obtained the highest weighted score for the U-Net architecture trained on the ROI dataset.

Table 4.5: Overview of VGG16 models for the ROI dataset running for I+II epochs each where the best models according to the lowest validation loss are saved. The evaluation scores, Dice score, recall and weighted score, are also presented for each model.

Model nr	Learning Rate	Loss Function	Dice Score	Recall	Weighted Score
19	1	Dice Loss	0.662	0.760	0.701
20	2	Dice Loss	0.653	0.726	0.682
21	3	Dice Loss	0.627	0.715	0.662
22	1	Focal Tversky	0.621	0.838	0.709
23	2	Focal Tversky	0.638	0.813	0.708
24	3	Focal Tversky	0.618	0.779	0.682

As seen in Figure 4.4 , an example image from the test set has been evaluated on model 19 - model 24.



(a) Input image **(b)** True label **(c)** Model 19 **(d)** Model 20

(e) Model 21 **(f)** Model 22 **(g)** Model 23 **(h)** Model 24

Figure 4.4: An example image from the test set evaluated on model 13- 18 . Purple indicates overlap, red indicates false positives, and blue indicates false negatives.

4.3 YOLOv8

The YOLOv8 network was applied on both the LIVER and ROI datasets, their performance is presented in Section 4.3.1 and Section 4.3.2. The three different learning rate combinations seen in Table 4.1, give three models per dataset, which are summarised in Tables 4.7 for the ROI dataset and 4.6 for the LIVER dataset. All models were run for a total of I+II epochs. The YOLOv8 backbone was frozen for the first I epochs, during which the rest of the network was trained. After this, the backbone was unfrozen and trained with a reduced learning rate for another II epochs for fine-tuning. The weights saved from the two different stages (frozen and unfrozen), were determined by the validation loss minimum. The evaluation metrics used were Dice Score and recall.

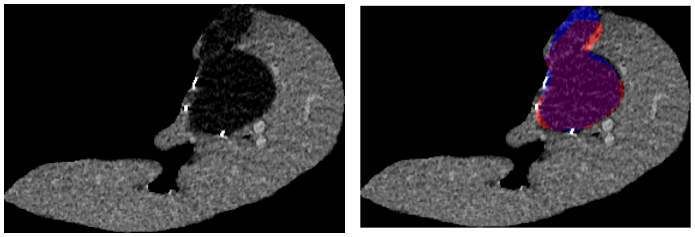
4.3.1 LIVER

The performance of the YOLOv8 models 25 to 27 are presented in Table 4.6 for the LIVER dataset. Model 27 obtained the highest evaluation scores for the YOLOv8 network on the LIVER dataset.

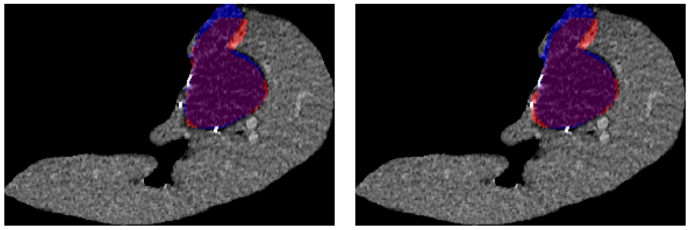
Table 4.6: Overview of YOLOv8 models for the LIVER dataset running for I epochs with frozen backbone and II epochs with all layers unfrozen, where the best models according to the lowest validation loss are saved. Which learning rate combination, as well as the evaluation scores, Dice score, recall and weighted score, are presented for each model.

Model nr	Learning Rate	Loss Function	Dice Score	Recall	Weighted Score
25	1	built-in	0.700	0.686	0.694
26	2	built-in	0.701	0.688	0.696
27	3	built-in	0.720	0.709	0.716

An example image was used by each model from Table 4.6 to predict a tumour and the result can be seen in Figure 4.5.



(a) Input image **(b)** Model 25



(c) Model 26 **(d)** Model 27

Figure 4.5: An example image from the test set evaluated on model 25 - 27. Purple indicates true positive, red indicates false positives, and blue indicates false negatives

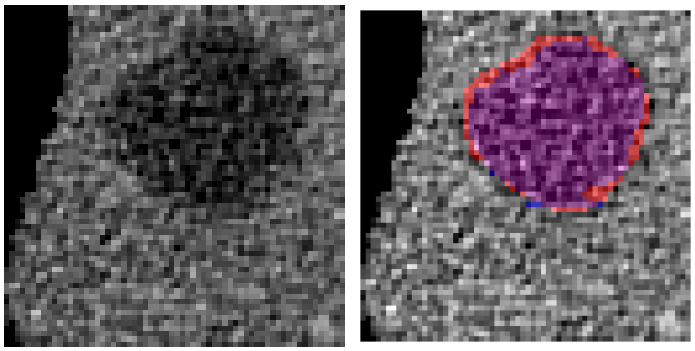
4.3.2 ROI

The performance of the YOLOv8 models 28 to 30 are presented in Table 4.7 for the ROI dataset. Model 30 obtained the highest evaluation scores for the YOLOv8 network on the ROI dataset.

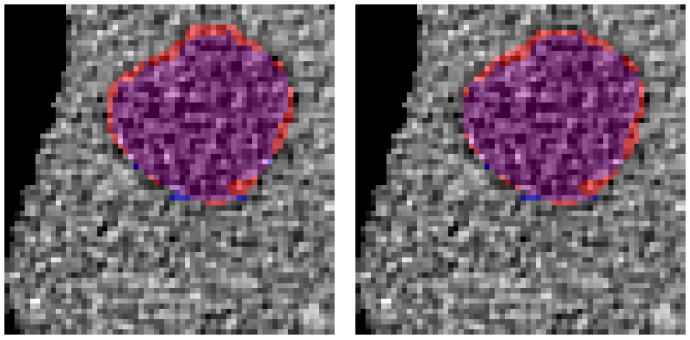
Table 4.7: Overview of YOLOv8 models for the ROI dataset running for I epochs with frozen backbone and II epochs with all layers unfrozen, where the best models according to the lowest validation loss are saved. Which learning rate combination, as well as the evaluation scores, Dice score, recall and weighted score, are presented for each model.

Model nr	Learning Rate	Loss Function	Dice Score	Recall	Weighted Score
28	1	built-in	0.672	0.799	0.723
29	2	built-in	0.689	0.792	0.730
30	3	built-in	0.690	0.809	0.738

An example image was used by each model from Table 4.7 to predict a tumour and the result can be seen in Figure 4.6.



(a) Input image **(b)** Model 28



(c) Model 29 **(d)** Model 30

Figure 4.6: An example image from the test set evaluated on model 28 - 30. Purple indicates true positive, red indicates false positives, and blue indicates false negatives

4.4 SAM

The SAM network was applied on both the LIVER and ROI datasets, their performance is presented in Section 4.8 and Section 4.4.2. There are two SAM models for the LIVER dataset, one with 0-10 pixels added from the label to the bounding box (model 31), and one with 0-20 pixels added (model 32), the performance of these are presented in Table 4.8. The performance of SAM on the ROI dataset is presented in Table 4.4.2. To evaluate the performance, Dice score and recall were used.

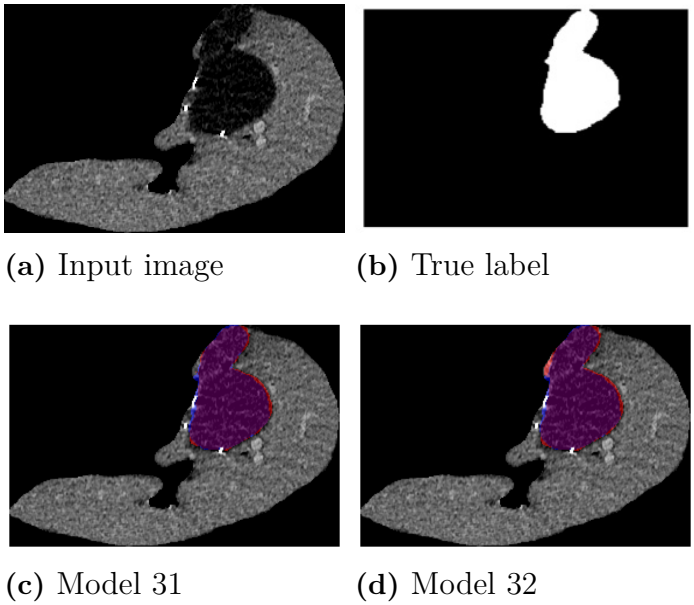
4.4.1 LIVER

There are two models for the LIVER dataset due to the different limits of how many pixels were added to the mask when generating the bounding box, 0-10 or 0-20. The results are summarised in Table 4.8, where the model with fewer pixels added (0-10) exhibited superior performance.

Table 4.8: Overview of SAM models for the LIVER dataset. The maximum pixel value added from the label to the generated bounding box, as well as the evaluation scores, Dice score, recall and weighted score, are presented for each model.

Model nr	Pixel-value	Dice Score	Recall	Weighted Score
31	20	0.410	0.724	0.536
32	10	0.504	0.753	0.604

An example image was used by each model from Table 4.8 to predict a tumour and the result can be seen in Figure 4.7.



(a) Input image **(b)** True label

(c) Model 31 **(d)** Model 32

Figure 4.7: An example image from the test set evaluated on models 31 and 32. Purple indicates true positive, red indicates false positives, and blue indicates false negatives

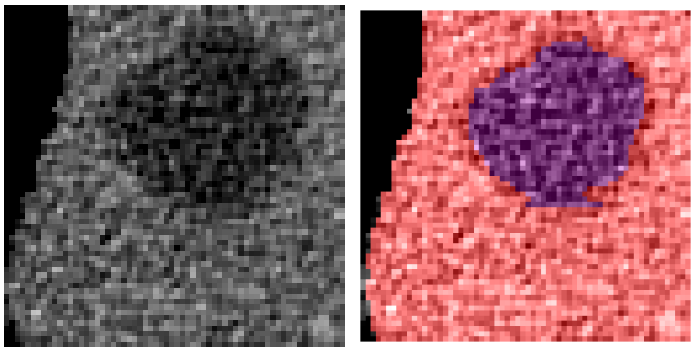
4.4.2 ROI

The performance of the SAM model 33 is presented in Table 4.4.2 for the ROI dataset.

Table 4.9: Overview of SAMs model for the ROI dataset. The evaluation scores Dice score, recall and weighted score, are presented here.

Model nr	Dice Score	Recall	Weighted Score
33	0.196	0.681	0.390

An example image was used by model 33 to predict a tumour and the result is displayed in Figure 4.8.



(a) Input image **(b)** Model 31

Figure 4.8: An example image from the test set evaluated on model 33. Purple indicates true positive, red indicates false positives, and blue indicates false negatives

4.5 Comparison between the networks

In this section, the performance from all networks and models are gathered to provide an overview of the results. The models are presented in terms of the dataset they were applied on, keeping them separate.

4.5.1 LIVER

In Table 4.10, all evaluation scores, Dice scores and recall, are presented for the models applied to the LIVER dataset. The model with the highest Dice score and weighted score is model 3, a U-Net model and the highest recall is model 33 which uses the network SAM.

Table 4.10: Summary of all models used on the LIVER dataset. Showing the model number, network as well as the evaluation metrics Dice score, recall and weighted score.

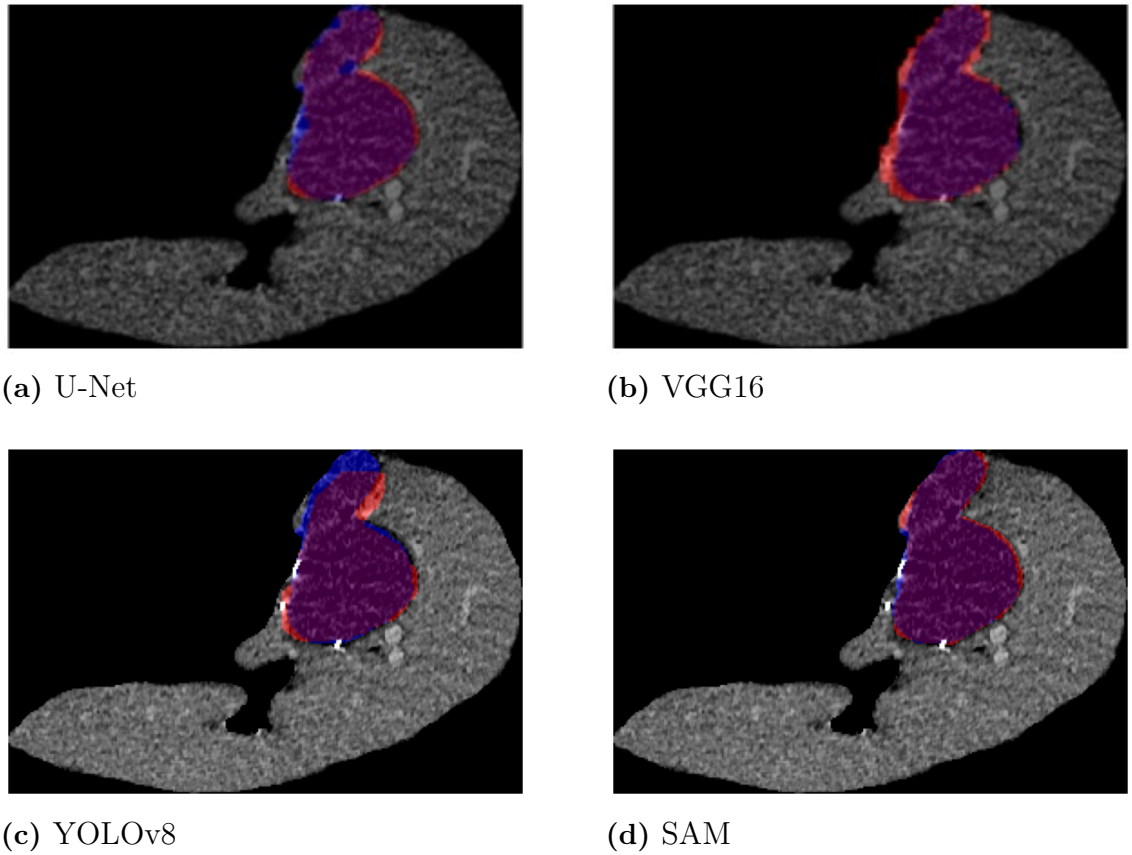
Model number	Network	Dice Score	Recall	Weighted Score
1	U-Net	0.699	0.704	0.701
2	U-Net	0.734	0.738	0.736
3	U-Net	**0.740**	0.742	**0.741**
4	U-Net	0.721	0.735	0.727
5	U-Net	0.698	0.720	0.707
6	U-Net	0.718	0.737	0.726
13	VGG16	0.712	0.738	0.722
14	VGG16	0.592	0.623	0.604
15	VGG16	0.619	0.638	0.627
16	VGG16	0.667	0.688	0.675
17	VGG16	0.592	0.624	0.605
18	VGG16	0.557	0.607	0.589
25	YOLOv8	0.700	0.686	0.694
26	YOLOv8	0.701	0.688	0.696
27	YOLOv8	0.720	0.709	0.716
31	SAM	0.410	0.724	0.536
32	SAM	0.504	**0.753**	0.604

The networks can vary greatly in weighted score, so to get a better understanding of how the network performed, the variation was calculated. This was done by taking the highest weighted score for that model minus the lowest weighted score and is presented in Table 4.11. The average weighted score is also presented here to give a better overview.

Table 4.11: The average Dice score and recall and the variation for both for each network on the LIVER dataset.

Network	Average Weighted Score	Variation Weighted Score
U-Net	0.723	0.04
VGG16	0.637	0.118
YOLOv8	0.702	0.022
SAM	0.570	0.068

The best model from each network with the LIVER dataset is collected based on weighted scores and presented on an example image in Figure 4.9.



(a) U-Net **(b)** VGG16

(c) YOLOv8 **(d)** SAM

Figure 4.9: An example image from the test set evaluated on each network with the highest weighted score for the LIVER dataset. Purple indicates true positives, red indicates false positives, and blue indicates false negatives.

4.5.2 ROI

The collected evaluation scores, Dice score and recall, for the networks applied on the ROI dataset are presented in Table 4.12. The model which had the highest Dice score was the U-Net model 9 with a score of 0.717. Model 22, a VGG16 model, had the highest recall value of 0.838 and the U-Net model 11 had the highest weighted score of 0.754.

Table 4.12: Summary of all models used on the ROI dataset. Showing the model number, network and evaluation metrics Dice score, recall and weighted score.

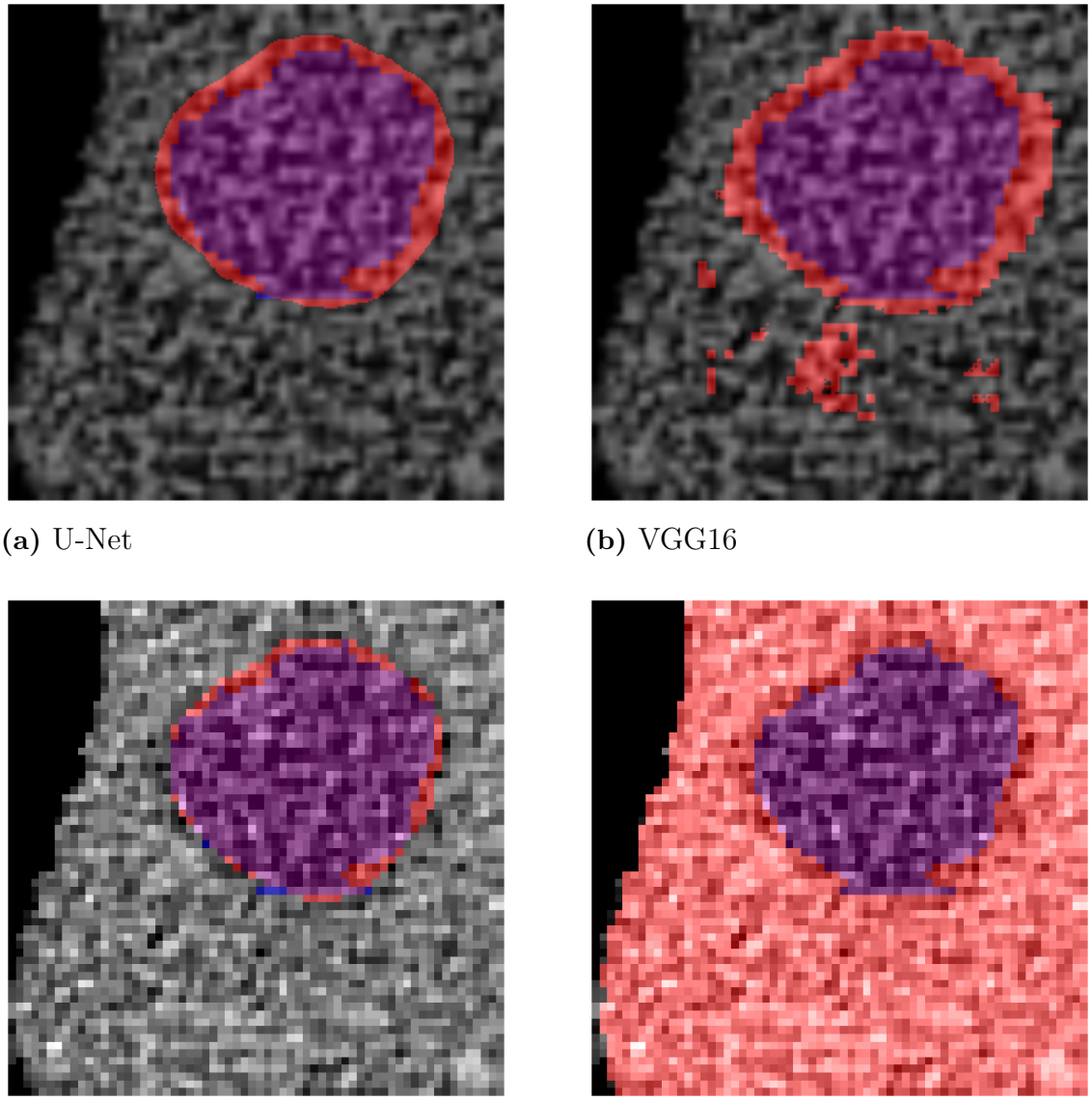
Model number	Network	Dice Score	Recall	Weighted Score
7	U-Net	0.670	0.685	0.676
8	U-Net	0.704	0.771	0.731
9	U-Net	**0.717**	0.760	0.734
10	U-Net	0.708	0.798	0.744
11	U-Net	0.702	0.831	**0.754**
12	U-Net	0.716	0.792	0.746
19	VGG16	0.662	0.760	0.701
20	VGG16	0.653	0.726	0.682
21	VGG16	0.627	0.715	0.662
22	VGG16	0.621	**0.838**	0.709
23	VGG16	0.638	0.813	0.708
24	VGG16	0.618	0.779	0.682
28	YOLOv8	0.672	0.799	0.694
29	YOLOv8	0.689	0.792	0.696
30	YOLOv8	0.690	0.809	0.716
33	SAM	0.196	0.681	0.390

The internal variations (highest - lowest score) in the weighted score for U-Net, VGG16 and YOLOv8, along with the average weighted score, are presented in 4.13.

Table 4.13: The average and variation of the weighted score for the ROI dataset.

Network	Average Weighted Score	Variation Weighted Score
U-Net	0.730	0.078
VGG16	0.691	0.047
YOLOv8	0.702	0.022

An example image was predicted by the best models from each network on ROI, based on the weighted score, and is shown in Figure 4.10.



(a) U-Net **(b)** VGG16

(c) YOLOv8 **(d)** SAM

Figure 4.10: An example image from the test set evaluated on each network with the highest weighted score for the ROI dataset. Purple indicates overlap, red indicates false positives, and blue indicates false negatives.

5

Discussion

In this chapter, results are analysed from the two datasets, similar overall performances was observed with notable internal variations among networks and models. Our discussion will sequentially compare the performances across different networks and datasets, examine the models within each dataset, to then cross-compare the models and datasets to elucidate underlying patterns and discrepancies.

5.1 U-Net

As mentioned in section 2.4.3, the architecture for U-Net is preferred for biomedical images due to its ability to localise features locally and globally due to its encoder (upsamling) and decoder (downsampling) together with the skips connections.

Evaluation of various models trained with the U-Net architecture on the LIVER dataset revealed indications of performance trends related to the choice of loss function and learning rate. Model 3 achieved the highest scores across all evaluation metrics according Table 4.2, indicating efficacy in segmenting liver images. Model 3 scored 0.741 in weighted score. Examination of the impact of loss functions showed that models employing the Dice loss demonstrated higher evaluation metrics with lower learning rates, suggesting that lower learning rates facilitate better model convergence and accuracy. Conversely, models using the focal Tversky loss did not exhibit a consistent pattern regarding the benefits of higher or lower learning rates, as performance varied across different runs. This variability suggests that additional factors may influence the effectiveness of the learning rate in these models. These findings highlight the importance of carefully selecting and tuning model parameters to optimise segmentation performance in medical imaging tasks.

For the ROI dataset trained on the U-Net architecture, model 11 achieved the highest weighted score of 0.754 and highest recall value as presented in Table 4.3. In terms of Dice score, model 9 performed the best. This indicates that model 9 was superior in achieving overlap between predicted and actual segmentation, whereas model 11 was more effective in identifying true tumour pixels. Focal Tversky loss was found to be the most effective loss function for the ROI dataset in conjunction with the U-Net architecture. Additionally, it was determined that a lower learning rate for the ROI dataset leads to higher evaluation scores.

5.2 VGG16

The VGG16 network architecture, comprising multiple convolutional and max pooling layers, effectively captures patterns in image data. An evaluation of various models trained with the VGG16 architecture on the LIVER dataset revealed performance trends associated with the choice of loss function and learning rate. Model 13, in particular, achieved the highest evaluation scores across all metrics. As demonstrated in Table 4.2.1, higher learning rates correlate with improved evaluation scores, and the Dice loss function appears to be the most effective for maximising performance for the LIVER dataset in combination with the VGG16 architecture.

For the ROI dataset, model 22 attained the highest weighted score at 0.709 and recall value 0.838 among the VGG16 models, while model 20 achieved the highest Dice score of 0.653 according to Table 4.2.2. Similar to the LIVER dataset, higher learning rates for the ROI dataset result in better evaluation scores. Additionally, the focal Tversky loss function generally yields superior performance outcomes for the ROI dataset in combination with the VGG16 architecture.

5.3 YOLOv8

Despite YOLOv8 being a "black box" and in some sense harder to modify compared to other networks used in this project, the results indicate that YOLOv8 can still be effectively adapted to this medical imaging task. In this study, a specific combination of learning rates (frozen layers at 10^{-4} and unfrozen layers at 10^{-6}) consistently performed better across different datasets, suggesting that even within the constraints of a "black box" model, there is room for significant optimisation tailored to medical image segmentation. It may also indicate either a potentially optimal learning rate setting for this specific dataset and task combined with the YOLOv8 network, or just that for this task, a lower learning rate is preferable. When looking at the image example of the different models in Figure 4.5 and Figure 4.6, one can see that their visual performance on that image is very similar to one another on both of the datasets.

One factor that could affect performance is the size of the network. As previously stated, YOLOv8 comes in five different sizes, and the middle size, "m," was used in this project. Limited testing was done to choose this size, so it is possible that it might be too complex for the size of the dataset, implying that a smaller size might perform better. Else, the dataset might be sufficient for a larger model.

Another thing to consider is that YOLOv8 takes in the input of the labels as text files, meaning that they have to be converted to such beforehand. This gives a margin of error since it is not the same input as for the other networks, and some information might be lost in the translation.

5.4 SAM

SAM, distinguished by its unique input requirement of bounding boxes, offers an advantage by directly focusing on the general area of the tumour. This approach contrasts with other networks used in this project, which do not differentiate between images with and without tumours in their preprocessing. This potentially simplifies the detection task when tumours are present for this particular network. Also, the same specialisation might limit its application breadth, as it does not process images without tumours, unlike the other networks.

SAM was used in size ViT-B, the least complex of the sizes. Similar to the discussion surrounding the size of the YOLOv8 network, the size of the SAM network can have a great affect on the result and successful training.

Despite SAM's design to operate without extensive prior training, its performance in medical imaging tasks was moderate, see Table 4.8 and Table 4.9. The SAM network, originally configured for general object detection, seems to struggle with the specificity required for medical images. This discrepancy is evident in the Dice scores, which were the lowest among all models tested for both datasets as can be seen in Table 4.10 and Table 4.12. The low Dice scores and high recall values for the LIVER dataset can be due to SAM's tendency to segment extensive areas. Although it seems to perform relatively very well on the LIVER dataset, on the ROI dataset, it seems to have trouble with mistaking the liver edges for the tumour and instead segmenting too large an area, as can be seen in the example prediction of the model 33 in Figure 4.8. This also points to the fact that SAM is not built for ROI images, but for larger images with a bounding box to provide more information when segmenting. The over-segmentation suggests that SAM could significantly benefit from tailored training on medical datasets. It also implies that a tumour might be too small for SAM to focus on effectively, and it might be better utilised for segmenting larger structures like the liver in a cascade network. These results highlight the challenges in utilising untrained models like SAM for complex tasks such as medical image segmentation, where the accuracy of pixel-level predictions is paramount.

5.5 Comparison

This section compares the TL networks to the U-Net network for both datasets separately. The datasets are then compared to each other, and finally, there is a general discussion about the impact of learning rate and loss function.

One important aspect to discuss is the management of recall. As described in more detail in Section 3.5.1, handling recall in images containing no tumour is problematic. The model is either greatly punished or rewarded depending on its prediction, which can make this metric misleading and significantly affect the recall score. Additionally, if the model classifies the entire image as a tumour, it will achieve a recall

score of 1, despite this not being an accurate prediction. Therefore, especially in the ROI dataset, the Dice score is a better indicator of a good model.

This reasoning supports the use of a weighted score and explains why the Dice score is valued higher than recall, as seen in Equation 3.1, where the Dice score is multiplied by a factor of 0.6 and recall by 0.4.

5.5.1 Transfer learning

An important note about the TL networks used in this project is that they are not pre-trained on medical images. VGG16 is trained on ImageNet and YOLOv8 on COCO, neither of which contains biomedical images. The images in these datasets differ significantly from medical images in terms of texture, shape, and contextual cues. This discrepancy can lead to difficulties in feature recognition when these networks are directly applied to medical datasets. The lack of standout performance from the TL networks might be due to the substantial differences between the source domain (pre-trained datasets) and the target domain (LIVER and ROI datasets). However, the TL networks' pre-training includes learning edges and gradients at the initial layers, as they are CNNs. The process of freezing and unfreezing layers is crucial for preserving the weights from this pre-training. When the backbone is frozen, it utilises its learned capacity to detect edges and gradients, while the subsequent layers are trained specifically to recognise the form of tumours in this case. Once the backbone is unfrozen, the entire network undergoes fine-tuning to the medical data, thereby improving its performance on the specific task.

5.5.2 Liver Dataset

In evaluating the LIVER dataset, Dice scores, recall values and weighted scores varied distinctly across different models as detailed in Table 4.10. U-Net model 3 stood out by achieving the highest weighted score of 0.741 and also the highest Dice score of 0.740. The highest recall value, 0.753, was by model 32 which was a SAM model, that model, however, had a low Dice score, 0.504, which also affected the weighted score, 0.604, making it one of the lowest of the investigated models. Overall, the U-Net had majority of the the best performing models both in Dice score, recall and weighted score. This means it outperform the TL networks, making it the best suited for this dataset.

The stability analysis, referenced in Table 4.11, details the average weighted scores. Comparing the TL networks against each other, meaning VGG16, YOLOv8 and SAM, it can be seen that YOLOv8 outperforms the others with the highest average weighted score of 0.702 and the lowest variation of 0.022, making it a good choice out of these. VGG16 on the other hand, had a lower average weighted score and the highest variation, pointing to it not being suitable for this dataset. SAM had a low variation but also the lowest average weighted score out of the TL networks, making

it not trustworthy enough to use in this context as is. Comparing the TL networks to U-Net, it can be seen that U-Net outperforms the best TL model (YOLOv8) in terms of average weighted score and on top of that, has one of the lowest variations on weighted score. This reinforces the notion that it is suitable for this dataset and task.

5.5.3 Region of Interest Dataset

The ROI dataset, when applied to different networks, shows performance variations as detailed in Table 4.12. U-Net model 11 achieved the highest weighted score at 0.754 among all the investigated models. Moreover, U-Net model 9 achieved the highest Dice score at 0.717, while VGG16 model 22 recorded the highest recall value at 0.838. In general, TL networks perform better in recall than U-Net, while U-Net performs better in Dice score than the TL networks. Unlike the LIVER dataset, there is a notable mismatch between high Dice scores and recall values across models, with no single model outperforming the others. This highlights a critical trade-off between sensitivity and precision.

Table 4.13 details the average weighted scores for the ROI dataset. YOLOv8 ranks second-highest in average weighted score, presenting the best overall performance among the three TL networks. Given its consistent performance, YOLOv8 may be a good choice in scenarios requiring stable results, despite not always achieving the highest scores in individual metrics. On the other hand, VGG16 achieves the lowest average weighted score, showing less consistency and making it less ideal for this dataset. Similarly, SAM, with the lowest weighted score according to Table 4.12, is an outlier and not a favourable choice. The highest weighted score is achieved by U-Net, making the U-Net architecture the most favourable choice for ROI. Regarding internal variability, as shown in Table 4.13, the variation in weighted scores across all networks is relatively low, with U-Net showing the highest variation at 0.078, followed by VGG16 at 0.047 and YOLOv8 at 0.022. This suggests that most models maintain consistent segmentation accuracy, with U-Net being the most dependent on the settings.

5.5.4 General Discussion

The highest weighted score for the LIVER dataset was 0.741, achieved by model 3 using the U-Net architecture. Similarly, model 11 using the U-Net architecture achieved the highest weighted score for the ROI dataset at 0.754. The high weighted score for the ROI dataset is due to its high recall value. The ROI dataset generally exhibits higher individual recall scores than the LIVER dataset, indicating a greater ability to identify all relevant areas, though this may result in more false positives. In contrast, the LIVER dataset demonstrates higher individual Dice scores than the ROI dataset, suggesting it is better at accurately isolating regions of interest with fewer false positives.

The observed differences in performance metrics between the datasets are primarily

attributable to the nature of the image cropping specific to each dataset since the settings were the same for each dataset. This highlights how dataset characteristics can significantly influence the performance of the same models across different tasks. As discussed earlier, the network performance varies between the datasets, however, U-Net seems to always be the best choice.

There is not much of a trend when looking at different learning rates, see Table 4.10 and Table 4.12. For instance, when looking at the LIVER dataset, higher learning rates seemed beneficial for VGG16, improving performance across both of the selected loss functions. In contrast, lower learning rates yielded better results for U-Net when paired with the Dice loss function, and YOLO. For the ROI there is even less of a pattern since there is not as clear a combination of high Dice scores and high recalls for the models, as mentioned earlier. This variability indicates that there is no universally optimal learning rate; instead, the effectiveness of learning rate settings may depend on the specific network and task configuration as well as other factors like preprocessing.

There is no clear pattern regarding which loss function provides the best evaluation scores, as performance varies depending on the dataset, learning rate, and other parameters. However, there is an indication that Dice loss performs better on the LIVER dataset, while focal Tversky loss performs better on the ROI dataset. This trend applies to both the U-Net and VGG16 architectures.

Although efforts have been made to make the models as similar to each other as possible, to be able to compare them fairly, they still differ in some key aspects. Apart from YOLOv8 and SAM having different kinds of inputs as text files and bounding boxes compared to the others. YOLOv8 also differs with its built-in loss function, making it impossible to compare them all with the same loss functions. In addition to this, since U-Net is not a transfer learning network, it can not have frozen layers. The decision in this project was to only use the learning rate that the TL have for the frozen layers on U-Net for the total sum of epochs that the TL networks used for both freeze and unfreeze, but it is not certain that this makes for a fair comparison.

There are differences in the performances of the models. That said, apart from SAM, all models achieved good evaluation metric scores, and the analysis focuses on smaller variations. This indicates that the networks are well-suited for both datasets introduced in this report.

5.6 Comparison to past work

In comparing this thesis to previous works, significant deviations are evident despite a similar structural approach. A key difference is the use of TL networks instead of thresholding algorithms and active contour models, although both studies used U-Net as a comparison. Both this and prior studies focused on creating LIVER and

ROI datasets from the LiTS dataset. However, this thesis introduced several key changes. Firstly, the LIVER dataset was cropped directly around the liver rather than merely removing the background, which previously resulted in a significant amount of black space. Additionally, the current LIVER dataset uniquely includes images without tumours, a departure from earlier studies. For the ROI dataset, while the cropping technique around the tumour remained similar, this thesis only included images with a single tumour, unlike the previous inclusion of multiple tumours. Moreover, the preprocessing steps also diverged; notably, no limits were set for HU during image conversion, differing from last year's methods.

The training methodology was modified from a fixed number of epochs for each model to a more flexible 'best model' approach based on performance outcomes. Learning rates were another area of difference. This year's thesis employed the same learning rates (A & B) as the previous year in addition to C. A difference in this year's though was that they were tailored specifically for the 'freezing' phase of model training and reduced them even further for the 'unfreezing' part for the TL networks. Adjustments were also made to the loss functions used; while previous works employed Dice loss and a variation that classified both tumour and background, this project introduced focal Tversky loss, chosen for its properties well-suited to this task and dataset. Furthermore, this year, the 'positive cases' evaluation metric was excluded, as it did not add substantial insight into the results.

Performance comparisons between the years show nuanced results, see Table 1.1, Table 4.10 and Table 4.12. For the LIVER dataset equivalent, the previous highest achieving model scores were 0.689 for Dice score and 0.720 for recall, compared to this year's 0.740 and 0.742, respectively, showing an improvement in Dice score and recall. For the ROI dataset, last year's top model scores were 0.766 for Dice score and 0.796 for recall, whereas this year's achieved 0.702 and 0.831, respectively, marking higher recall but lower Dice score. This result suggests that this year's models are slightly better, although not a huge difference can be seen.

5.7 Future recommendations

VGG16, YOLOv8, and SAM, which represent the TL networks, do not show any significant improvement over U-Net. However, different suggestions are made for further exploration in the area of TL and modifications of the U-Net architecture, as both produced relatively good outcomes for both datasets where U-Net scored the highest evaluation scores.

Regarding the LiTS dataset, its heterogeneity in contrast, brightness, and other aspects necessitates ensuring the network is robust enough to handle these variations. In this report, histogram equalisation was used to address individual image differences. It is worth further developing this technique to improve robustness against dataset variations. However, focusing on the ROI and LIVER datasets used in this report, the ROI dataset, trained to detect tumours, may incorrectly identify tumours in images without them. This limitation restricts its usability. Therefore,

it is advised not to use this ROI dataset in its current form. Adding images of livers without tumours or focusing on the LIVER dataset, which is more suitable for detecting tumours in liver CT scans, is recommended. Since the LiTS dataset is available in NIfTI files, using the dataset in 3D instead of 2D, as done in this report, is a possible improvement.

Despite using dropout and data augmentation, some overfitting occurred. The best model, defined as the one with the lowest validation loss during training, was used. An alternative approach could involve saving the model with the lowest validation loss at any point during training. Additionally, L2 regularisation could be employed to further prevent overfitting. In this report, only one set of data augmentation was included due to time constraints. Future work should include multiple sets of data augmentation to better prevent overfitting.

There are many other aspects to explore, such as different combinations of hyperparameters, loss functions, and optimizers. It can be beneficial to try different learning rates, batch sizes, and image sizes. Since the majority of the networks used in this report were TL networks pre-trained on specific image sizes, different image sizes were used for different networks. For the LIVER dataset, images were cropped around the liver, resulting in varying sizes. This may have caused some images to appear blurry if their size was smaller than the network's input size.

Furthermore, future work should include training the SAM network, as it was not done in this report, to assess its performance. Running the networks with different divisions of the training, validation, and test sets to obtain an average performance value is also recommended to account for statistical variance in the dataset. Combining TL networks to achieve better results is another possibility. Both YOLOv8 and SAM are object detection and segmentation networks, allowing for the creation of a cascade network where one network locates the tumour and another segments it. A cascade network based on U-Net was tested but not fully explored due to work limitations. This is possible as the LiTS dataset, annotated for background, liver, and tumour, is suitable for exploring cascade networks, which showed promising outcomes during the LiTS challenge. Additionally, various modifications of the U-Net should be tested for better performance.

6
Conclusion

In conclusion, this analysis highlights the importance of careful dataset preparation, thoughtful model selection, and hyperparameter tuning to optimise model performance. Key factors such as loss functions, learning rates, and preprocessing strategies play crucial roles, with differences observed when compared to last year's data.

Future research should continue to use the LIVER dataset due to its real-world applicability and immediate usability. Among the models evaluated, the U-Net architecture stands out, especially when prioritising the Dice score over recall, weighted score, for both datasets. In contrast, TL networks do not provide advantages over the U-Net architecture but small adjustments may lead to better performance.

Bibliography

[1] A. Wolfewicz, "Deep Learning vs. Machine Learning – What's The Difference?" 2023. [Online]. Available: https://levity.ai/blog/difference-machine-learning-deep-learning

[2] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," Tech. Rep., 2015. [Online]. Available: https://arxiv.org/abs/1505.04597

[3] Khuyen Le, "An overview of VGG16 and NiN models," 2021. [Online]. Available: https://lekhuyen.medium.com/an-overview-of-vgg16-and-nin-models-96e4bf398484

[4] Ultralytics, "Model Training with Ultralytics YOLO," 2023. [Online]. Available: https://docs.ultralytics.com/

[5] ——, "YOLOv8 Docs," 2023. [Online]. Available: https://docs.ultralytics.com/models/yolov8/#performance-metrics

[6] R. I. Sridhar and R. Kamaleswaran, "Lung Segment Anything Model (LuSAM): A Prompt-integrated Framework for Automated Lung Segmentation on ICU Chest X-Ray Images," *IEEE TRANSACTIONS ON MEDICAL IMAGING*, pp. 1–8, 2023.

[7] P. Bilic *et al.*, "The liver tumor segmentation benchmark (lits)," *Medical Image Analysis*, vol. 84, Feb 2023.

[8] National Cancer Institute, "What Is Liver Cancer? - NCI," *National Institutes of Health*, pp. 1–12, 2022. [Online]. Available: https://www.cancer.gov/types/liver/what-is-liver-cancer

[9] S. Schultz, "Levercancer," 2022. [Online]. Available: https://www.1177.se/sjukdomar--besvar/cancer/cancerformer/levercancer/

[10] "Navari – Revolutionizing keyhole surgery with augmented reality." [Online]. Available: https://navarisurgical.com/

[11] 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, 8 2022.

[12] U. L. Jayarathne, J. Moore, E. C. Chen, S. E. Pautler, and T. M. Peters, "Real-time 3D ultrasound reconstruction and visualization in the context of laparoscopy," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 10434 LNCS, pp. 602–609, 2017. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-319-66185-8_68

[13] S. Allgöwer and S. Ljungdahl, "Liver tumor segmentation using classical algorithms & deep learning," Chalmers University of Technology, Tech. Rep., 2023. [Online]. Available: www.chalmers.se

[14] I. M. Arias, A. W. Wolkoff, S. S. Thorgeirsson, and E. Al., *The Liver: Biology and Pathobiology*. Wiley-Blackwell, 2020.

[15] S. T. Orcutt and D. A. Anaya, "Liver Resection and Surgical Strategies for Management of Primary Liver Cancer," *Cancer control : journal of the Moffitt Cancer Center*, vol. 25, no. 1, pp. 1–15, 1 2018. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/29327594/

[16] S. Buettner, J. L. Van Vugt, J. N. Ijzermans, and B. G. Koerkamp, "Intrahepatic cholangiocarcinoma: current perspectives," *OncoTargets and therapy*, vol. 10, pp. 1131–1142, 2 2017. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/28260927/

[17] I. M. Arias, A. W. Wolkoff, S. S. Thorgeirsson, D. A. Shafritz, D. E. Cohen, J. L. Boyer, and H. J. Alter, *The Liver : Biology and Pathobiology*, 6th ed. Wiley-Blackwell, 2020.

[18] J. L. Prince and J. M. Links, *Medical Imaging Signals and Systems*, 2nd ed. Upper Saddle River, N.J.:: Pearson Education, 2015.

[19] "Computed Tomography (CT)," 2022. [Online]. Available: https://www.nibib.nih.gov/science-education/science-topics/computed-tomography-ct

[20] M. Hultenmo, "Computed Tompgraphy," Chalmers University of Technology, 2023.

[21] L. Lechuga and G. A. Weidlich, "Cone Beam CT vs. Fan Beam CT: A Comparison of Image Quality and Dose Delivered Between Two Differing CT Imaging Modalities," *Cureus*, vol. 8, no. 9, 9 2016. [Online]. Available: /pmc/articles/PMC5063198//pmc/articles/PMC5063198/?report=abstracthttps://www.ncbi.nlm.nih.gov/pmc/articles/PMC5063198/

[22] J. L. Prince and J. M. Links, *Medical Imaging Signals and Systems*, 2nd ed. Upper Saddle River, N.J: Pearson Education, 2015.

[23] ——, *Medical Imaging Signals and Systems*, 2nd ed. Upper Saddle River, NJ: Pearson Education, 2015.

[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, 5 2016.

[25] V. Kanade, "What is machine learning? Understanding types & applications," 2022. [Online]. Available: https://www.spiceworks.com/tech/artificial-intelligence/articles/what-is-ml/

[26] IBM, "What is Deep Learning?" [Online]. Available: https://www.ibm.com/topics/deep-learning

[27] L. Torrey and J. Shavlik, "Transfer Learning," *https://services.igi-global.com/resolvedoi/resolve.aspx?doi=10.4018/978-1-60566-766-9.ch011*, pp. 242–264, 1 2010. [Online]. Available: https://www.igi-global.com/chapter/transfer-learning/36988www.igi-global.com/chapter/transfer-learning/36988

[28] N. Donges, "What Is Transfer Learning? A Guide for Deep Learning," 2022. [Online]. Available: https://builtin.com/data-science/transfer-learning

[29] I. Goodfellow, Y. Bengio, and A. Courville, "Machine learning basics," in *Deep Learning*, 1st ed. MIT Press, 2016, ch. 5, pp. 109–120.

[30] M. Gurucharan, "Basic CNN Architecture: Explaining 5 Layers of Convolutional Neural Network," 2022. [Online]. Available: https://www.upgrad.com/blog/basic-cnn-architecture/

[31] A. Mishra, "Deep Learning Fundamental- Important Concepts," 2019. [Online]. Available: https://medium.datadriveninvestor.com/deep-learning-fundamental-important-concepts-59d7ae90901b

[32] I. Goodfellow, Y. Bengio, and A. Courville, "Machine learning basics," in *Deep Learning*, 1st ed. MIT Press, 2016, ch. 4, pp. 82–83.

[33] D. P. Kingma and J. L. Ba, "Adam: A Method for Stochastic Optimization," *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 12 2014. [Online]. Available: https://arxiv.org/abs/1412.6980v9

[34] A. Upadhyay, "20 Must-Know Topics In Deep Learning For Beginners | Medium," 2023. [Online]. Available: https://medium.com/@aspershupadhyay/mastering-deep-learning-20-key-concepts-explained-ea405aa6603d

[35] "Epochs, Batch Size, Iterations - How they are Important," 2023. [Online]. Available: https://www.sabrepc.com/blog/Deep-Learning-and-AI/Epochs-Batch-Size-Iterations

[36] S. Khanna, "A Comprehensive Guide to Train-Test-Validation Split in 2024," 2024. [Online]. Available: https://www.analyticsvidhya.com/blog/2023/11/train-test-validation-split/

[37] "What is Overfitting?" [Online]. Available: https://www.ibm.com/topics/overfitting

[38] Keras, "Transfer learning & fine-tuning," 2023. [Online]. Available: https://keras.io/guides/transfer_learning/

[39] 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, p. 105620, 8 2022.

[40] S. Mascarenhas and M. Agarwal, "A comparison between VGG16, VGG19 and ResNet50 architecture frameworks for Image Classification," in *Proceedings of IEEE International Conference on Disruptive Technologies for Multi-Disciplinary Research and Applications, CENTCON 2021*. Institute of Electrical and Electronics Engineers Inc., 2021, pp. 96–99.

[41] P. Gayathri, A. Dhavileswarapu, S. Ibrahim, R. Paul, and R. Gupta, "Exploring the Potential of VGG-16 Architecture for Accurate Brain Tumor Detection Using Deep Learning," *Journal of Computers, Mechanical and Management*, vol. 2, no. 2, 6 2023.

[42] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," 9 2014. [Online]. Available: http://arxiv.org/abs/1409.1556

[43] "COCO - Common Objects in Context." [Online]. Available: https://cocodataset.org/#home

[44] G. Boesch, "A Guide to YOLOv8 in 2024," 2024. [Online]. Available: https://viso.ai/deep-learning/yolov8-guide/

[45] Khoa Le, “A review of YOLOv8 ecosystem,” 2024. [Online]. Available: https://vankhoa21991.medium.com/a-review-of-yolov8-ecosystem-58675b386080
[46] Ultralytics, “Instance Segmentation Datasets Overview,” 2024.
[47] Piotr Skalski, “How to Use the Segment Anything Model (SAM),” 2024. [Online]. Available: https://blog.roboflow.com/how-to-use-segment-anything-model-sam/
[48] “Introduction to Loss Functions,” 2018. [Online]. Available: https://www.datarobot.com/blog/introduction-to-loss-functions/
[49] “Introduction to Loss Functions,” 2018. [Online]. Available: https://www.datarobot.com/blog/introduction-to-loss-functions/
[50] S. Jadon, “A survey of loss functions for semantic segmentation.” [Online]. Available: https://github.com/shruti-jadon/
[51] N. Abraham and N. Mefraz Khan, “A Novel Focal Tversky loss function with improved Attention U-Net for lesion segmentation,” 2018. [Online]. Available: https://github.com/nabsabraham/focal-tversky-unet

DEPARTMENT OF SOME SUBJECT OR TECHNOLOGY
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY