Automated MRI Segmentation and Feature Extraction for Predicting Liver Function Deploying Advanced Deep Learning Models

Dissertation

zur Erlangung des Doktorgrades
der Naturwissenschaften (Dr. rer. nat.)
der Fakultät für Physik der Universität Regensburg

Vorgelegt von
Florian Raab
aus Oberviechtach
im Jahr 2024

Promotionsgesuch eingereicht am: 28.06.2024
Die Arbeit wurde angeleitet von: Prof. Dr. Elmar W. Lang

Prüfungsausschuss:
Vorsitzender: Prof. Dr. Karsten Rincke
1. Gutachter: Prof. Dr. Elmar W. Lang
2. Gutachter: Prof. Dr. Stefan Solbrig
Weiterer Prüfer: Prof. Dr. Christian Schüller

Datum Promotionskolloquium: 28.10.2024

"The first principle is that you must not fool yourself -
and you are the easiest person to fool."
(Richard P. Feynman)

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Accurate assessment of liver function is crucial for diagnosing liver diseases, determining treatment options, and predicting outcomes in patients with liver disorders. Traditional methods of evaluating liver function rely on blood tests and invasive procedures like biopsies, which have limitations in terms of accuracy, patient comfort, and ability to capture the heterogeneity of liver tissue. There is a growing need for non-invasive techniques that can provide comprehensive and reliable information about liver health and functionality.

Magnetic resonance imaging (MRI) has proven to be a powerful non-invasive tool for visualizing liver anatomy and pathology. Recent advances in MRI technology and protocols have enabled the acquisition of high-resolution, multi-parametric images that can potentially reveal important biomarkers of liver function. However, the manual analysis and interpretation of these complex MRI datasets is time-consuming and subject to inter-observer variability based on examiner experience.

Artificial intelligence (AI) and machine learning techniques offer promising solutions for automating the analysis of medical images and extracting clinically relevant information. In particular, deep learning models have shown remarkable performance in tasks like organ segmentation and feature extraction from radiological images. Applying these advanced AI methods to liver MRI could potentially enable automated, quantitative assessment of liver function and improve clinical decision-making.

This thesis aims to develop and evaluate an automated pipeline for liver function prediction based solely on MRI-derived features, utilizing state-of-the-art deep learning and machine learning approaches.

The key objectives are:

- Implementation and comparison of different deep learning architectures for automated segmentation of liver and related structures from multi-phase MRI sequences.
- Refinement of the automated segmentation due to problems that arise specifically for breath-holding imaging techniques as used in this thesis.
- Extraction of comprehensive quantitative features from the segmented liver regions that may correlate with liver function.
- Development of classification and regression machine learning models that can accurately predict established liver function scores using only the MRI-derived features.
- Assessment and evaluation for the performance of the automated pipeline with comprehensive feature sets against traditional methods.

By achieving these objectives, this work seeks to advance the field of non-invasive liver function assessment and demonstrate the power of AI-driven analysis of medical imaging data. The proposed techniques could potentially improve the accuracy, efficiency, and accessibility of liver function evaluation, leading to better patient care and outcomes in hepatology.

Chapter 2

Preliminary Concepts

2.1 The Liver

This section provides a brief introduction to liver anatomy and physiology, the major liver functions, diseases that can lead to impaired liver functions and relevant scoring systems used for quantification. This foundation helps the reader with the necessary background to understand the deployed methods and the motivation behind this thesis. Unless otherwise stated, references [1, 2, 3] and [4] serve as the primary sources for this section.

2.1.1 Liver anatomy

The liver is located in the upper right quadrant of the abdomen, just below the diaphragm and is a vital organ. The basic anatomy is illustrated in figure 2.1. In adults, the liver weighs approximately 1.5 kg and consists of four lobes that are further subdivided into smaller lobules. These contain hepatic cells, so-called *hepatocytes*, arranged in a hexagonal pattern around a central vein. Specialized capillaries within the lobules, *Liver sinusoids*, facilitate the exchange of substances between the blood and hepatocytes. At any given moment, the liver holds about thirteen percent of the body's blood and is supplied by two main sources, the *hepatic artery* and the *portal vein.* The latter is responsible for $\approx 75\%$ of its blood supply.

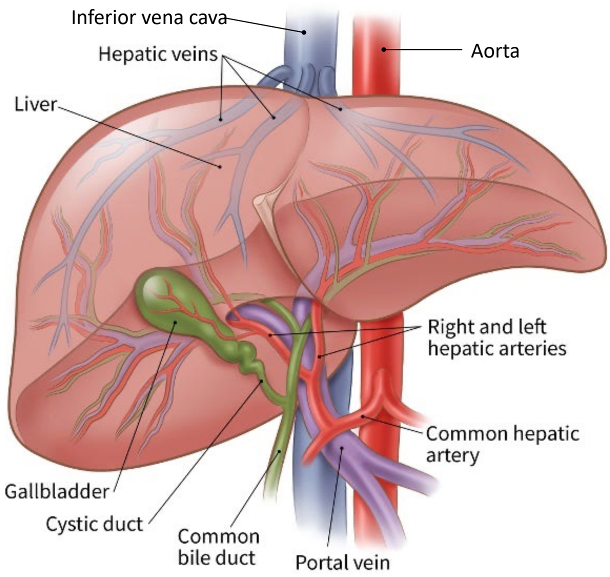


Figure 2.1 This figure illustrates the liver anatomy, including the liver parenchyma, the hepatic vascular system (comprising the portal vein, common hepatic artery, and right and left hepatic arteries), the hepatic veins, the inferior vena cava, the aorta, the gallbladder, the cystic duct, and the common bile duct. Adapted and extended from [5].

Generally, the *aorta* is the main artery that supplies oxygenated blood to the abdominal organs and to the hepatic artery. As blood flows through the gastrointestinal tract, spleen and pancreas, it picks up nutrients and other substances. This nutrient-rich, but oxygen-poor blood collects into smaller veins, converging to form the *portal vein* that carries this nutrient-rich blood directly to the liver. The *portal vein* divides into smaller vessels within the liver lobes, aiding in the processing of nutrients and detoxification prior to the blood's entry into the systemic circulation. The *hepatic veins* in turn collect blood from the liver lobules and carry it to the inferior vena cava, which then transports it back to the heart. The liver is also involved in the production of bile, which is transported via the bile ducts to the gallbladder and to the small intestine, where it aids the digestion and absorbtion of fats. Additionally, resident macrophages of the liver, so-called *Kupffer cells*, play a crucial role in immune function of the human body.

In summary, this complex anatomy supports the major liver functions, including the metabolism of macronutrients, detoxification of harmful substances like alcohol and other drugs, synthesis of plasma proteins as albumin and clotting factors, as well as the production and secretion of bile, which is essential for digestion and absorption of fats. Because of these vital roles, liver disease can lead to a variety of side-effects, like *ascites* (abdominal fluid accumulation), coagulation disorders or even hepatic coma. All of this highlights the liver's critical role in maintaining overall metabolic and digestive health. However, due to its multifunctional role in the body, the liver is susceptible to various diseases and conditions which subsequently can lead to impaired liver function. This can result from mutliple factors, including chronic alcohol abuse, nonalcoholic fatty liver disease, certain medications and viral hepatitis. Those conditions can lead to inflammation of the liver, fibrosis and eventually cirrhosis, which impacts the liver's ability to perform its vital functions [6].

2.2 Key Biomarkers of Liver Function: Albumin, Bilirubin, Creatinine and INR

Clinically, liver function is commonly assessed by four crucial biomarkers: *Bilirubin*, *Creatinine*, the *INR* (International Normalized Ratio) and *Albumin*. Those are integral components of the *MELD*- (Model for End-Stage Liver Disease), *ALBI*- (Albumin-Bilirubin) and *Child-Pugh*-scores. These are widely used to evaluate the severity of chronic liver disease and prioritize patients for liver transplantation [7, 8].

Relevant liver function scores for this thesis and their use cases will be explained in detail in section 2.3.

Albumin is synthesized exclusively by the liver and is the most abundant protein in the blood plasma. Serving as the main controller of plasma oncotic pressure, it also acts as a vital carrier molecule for numerous substances including vitamins, pharmaceuticals, and hormonal compounds. With impaired liver function, low levels of albumin (hypoalbuminemia) in the blood can occur. The albumin levels are a sensitive marker of malnutrition and chronic liver disease and therefore reflect its synthetic capacity and overall health [9, 10].

Bilirubin originates mostly from the breakdown of hemoglobin in red blood cells and is a yellow compound. The liver plays an essential role in processing bilirubin. It is responsible to convert unconjugated bilirubin (insoluble in water) into conjugated bilirubin (soluble and can be excreted). This process happens in the hepatocytes and is required to eliminate bilirubin from the body. Thus, impaired liver function can lead to elevated levels of bilirubin in the blood. Clinically this can manifest as *icterus*, characterized by yellowing of the skin and eyes. Therefore, bilirubin serves as a vital marker of hepatic function that reflects the liver's capacity to detoxify and excrete waste products [11, 12].

Creatinine is a byproduct from muscle metabolism and usually excreted by the kidneys. It is primarily associated with kidney function and muscle mass, but

is also relevant in the context of liver disease and thus included in several liver function scores. With severe liver dysfunction, one can develop a hepatorenal syndrome. This refers to a condition, where advanced liver disease is the reason for degraded kidney function. Therefore, comprised renal function as a result of liver disease can be observed by elevated serum creatinine levels [7, 8, 13].
INR (International Normalized Ratio) is an internationally standardized measure of blood clotting. Several clotting factors (proteins as prothrombin) that are essential for normal blood coagulation are synthesized in the liver. Impaired blood clotting can be observed in case of a damaged liver because of its deteriorated ability to produce those proteins. Due to this, there is an increased risk of bleeding, which is a critical concern in patients with liver disease. Subsequently, the INR is a crucial component of several liver function scores as it provides insight into the livers synthetic function and its ability to maintain hemostasis [14, 15].

2.3 Liver Function Scores

Liver function scores are essential tools used in clinical practice to assess the severity of liver disease and estimate outcomes in patients. They are also used for triage. There is a wide variety of tests and scores to assess liver function, including the CHILD-PUGH, MELD, ALBI, LiMAx, ICG-clearance, and others [16, 14, 17, 18, 19].
For the cohorts in this study there were three available, the **MELD-**, **-** and the **ALBI-score**, which will be described in the following.
The **MELD-score** accurately assesses the risk of death in patients with end-stage liver disease and is suitable to determine organ allocation priorities. It estimates the likelihood of a patient's survival over the next three months [20]. Initially, the MELD was calculated with serum creatinine and bilirubin levels, the INR for prothrombine time and a factor for different liver disease etiologies [14]. This last factor was removed in 2007 after extensive study, hence the current MELD-score formula is [21]:

$$\begin{aligned}\text{MELD score} =&3.78 \times ln(\text{bilirubin}[mg/dL]) + 11.2 \times ln(\text{INR}) + \\ &9.57 \times ln(\text{creatinine}[mg/dL]) + 6,43.\end{aligned} \tag{1}$$

This is commonly used for prioritizing the organ allocation for liver transplantations [21]. The MELD score ranges from six to 40, with lower scores indicating a higher likelihood of survival over the next three months. A MELD≥15 is generally considered an indication for being listed for liver transplantation. However, patients may be listed with a lower score if their quality of life is unacceptable due to the disease or if there are secondary complications of cirrhosis [22, 7, 8].

The **LiMAx-score** (liver maximum function capacity-score) is based on the LiMAx test, a minimal-invasive diagnostic tool to assess liver function. It involves the intravenous administration of ^{13}C-methacetine that is metabolized into $^{13}CO_2$ by the liver and subsequently exhaled by the patient. The increase in $^{13}CO_2$ in the exhaled breath is then compared to a baseline measurement before administration of ^{13}C-methacetine. The results are reported in micrograms of ^{13}C-methacetine metabolized per kilogram of bodyweight per hour (μg/kg/h). Higher values indicate better liver function and for a healthy liver, values above 315 are normal and allow for resection of up to four liver segments in surgery. Intermediate liver function is defined for values between 140-315. For patients with a LiMAx value lower than 140, surgery must be refused, because of a high likelihood to develop complete liver failure after resection. The LiMAx test has been shown to be a reliable tool for predicting postoperative liver failure and mortality [23, 17, 24].

The **ALBI-score** was initially developed as an evidence based approach to assess liver function in patients with hepatocellular carcinoma (HCC). However, it was successfully applied as a general liver function score for various aetiologies, as well [25].

It is based only on total bilirubin and serum albumin, which makes it easy to obtain and free from subjective clinical assessment and was developed as an alternative to the Child-Pugh classification. The Child-Pugh score, which is used for patients with liver cirrhosis, also relies on albumin and bilirubin, but also takes additional parameters into account that are highly subjective. Because of this, the ALBI-score seemed more suitable for the experiments conducted in this thesis [26].

This score is calculated with the following equation:

$$\text{ALBI score} = -0.085 \times (\text{albumin}[g/L]) + 0.66 \times log_{10}(\text{bilirubin}[mol/L]). \quad (2)$$

The result can then be categorized into three grades. For an ALBI-score ≤ -2.60 it is Grade 1. If $-2.60 <$ ALBI-score ≤ -1.39, it is Grade 2 and for values above -1.39, the Grade is 3. Higher grades indicate worse liver function. The ALBI-score has shown comparable or even superior prognostic performance to the more traditional liver function scores as Child-Pugh and MELD in many scenarios and is easy to obtain [26, 27].

2.4 Magnetic Resonance Imaging

This section will provide a short introduction to the physical and physiological concepts of magnetic resonance imaging (MRI). MRI is a non-invasive and nonionizing imaging technology that allows us to collect high-resolution volumetric images of the human anatomy with excellent soft-tissue contrast and is primarily used in medical settings. Unless otherwise specified, this section primarily relies on references [28, 29, 30, 31, 32, 33] and [34].

2.4.1 Magnetic Resonance Imaging Physics

The basic principle of MRI relies on the spin, which is a fundamental property of particles. The nuclear spin is of particular importance and thus the imaging technique was called nuclear magnetic resonance imaging (NMRI) in earlier days. However, because of the negative connotation of the word nuclear, the name was changed to the nowadays more common MRI.

The spin is characterized by the spin quantum number s that possesses values of multiples of $\frac{1}{2}$ and can be positive or negative. In the case of individual unpaired particles like neutrons, electrons and protons, $s = \frac{1}{2}$. The human body consists of about 63% hydrogen atoms and for medical applications the most relevant nuclear magnetic resonance (NMR) signals arise from water and fat, which are the predominant hydrogen containing tissues in human body. These hydrogen atoms contain a nucleus that is composed of a positively charged proton. Since only hydrogen atoms will be taken into consideration for the explanations, those nuclei will be referred to as protons in the following. Those posess the so-called NMR property due to two crucial characteristics. Because of their own nuclear spin, two effects arise. The spin results in an angular momentum $\vec{J}$ due to the odd-numbered atomic mass (i.e. ≈ 1) and it generates an electrical current because of the positive charge of the proton. When placed within a magnetic field, like in a MRI scanner, that current induces a torque which is called the magnetic moment $\vec{\mu}$.

The combination of those effects leads to the phenomenon that protons which are placed in an external magnetic field with flux density $\vec{B}$ (in the following referred to as the magnetic field), can absorb and emit energy in the radio frequency (RF) range of the electromagnetic spectrum. Consequently they produce a NMR signal, which can be captured and transformed into images.

To ensure the magnetic field $\vec{B}_0$ inside the scanner bore is both uniform and sufficiently strong (typically 1.5 to 7 Tesla for humans), a substantial electric current flowing through a coil is required. Therefore, the coil must be superconducting and is cooled down to 4.2K using liquid helium. The direction of $\vec{B}_0$ is pointing from feet to head of the subject. For the following explanations, this is defined as the z-axis, thus:

$$\vec{B}_0 = B_0 \hat{e}_z. \tag{3}$$

Consequently, the plane perpendicular to this axis can be defined as the x-y plane with the x-axis pointing from the left to the right hand of the subject. The y-axis points from the back to the front of the subject.
If a subject is placed into the magnetic field $\vec{B}_0$, $\vec{\mu}_z$ of the protons in its body aligns either parallel or anti-parallel to the magnetic field. However, $\vec{\mu}_x \neq 0$, $\vec{\mu}_y \neq 0$, therefore, the spinning protons exhibit a gyroscopic motion called precession, rather than remaining strictly aligned with the magnetic field (See Fig. 2.2). The angle between the spin axis and the axis defined by the magnetic field is dictated by the angular momentum $\vec{J}$ of the protons. The frequency of precession is called the Larmor-frequency

$$\omega_L = \gamma B_0, \tag{4}$$

with γ being the gyromagnetic factor and $\gamma_{proton} \approx 2.675 \cdot 10^8 \frac{rad}{s \cdot T}$.

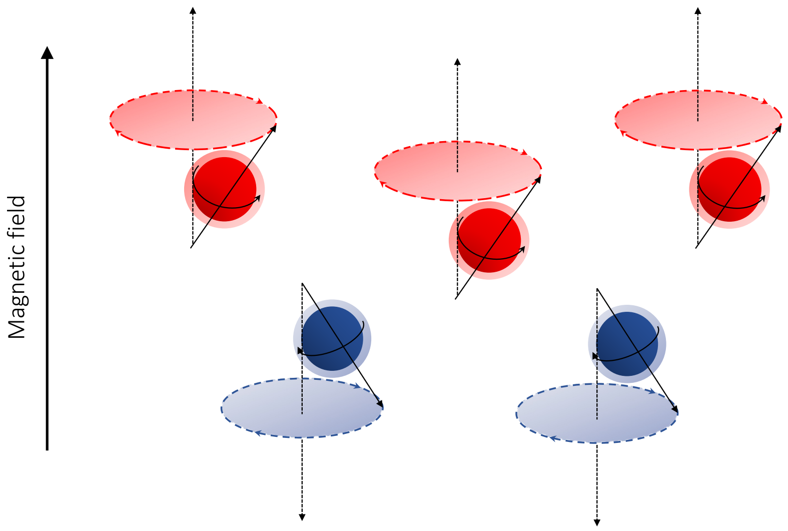


Figure 2.2 This figure illustrates the precession of high- and low-energy states of protons in an external magnetic field. They are either in the parallel state with lower energy (orange) or the antiparallel state with a higher energy level (blue). There are always more protons in the lower energy state, compared to the anti-parallel state.

The ratio of number of particles in parallel N^P and particles in anti-parallel N^{AP} configuration in equilibrium can be described by the Boltzmann statistics [34]:

$$\frac{N^{AP}}{N^P} = e^{-\frac{\Delta E}{k_B T}}. \tag{5}$$

There, ΔE is the difference of energy for both states, $k_B = 1.3806 \cdot 10^{-23} J$ the Boltzmann constant and T the temperature of the system. Since the parallel alignment represents a lower energy state compared to the anti-parallel alignment, there's a greater abundance of particles aligned parallel to the magnetic field. As a consequence, there is an excess of particles with a component pointing in the direction of $\vec{\mu}_z$. This surplus gives rise to the so-called netto magnetization

$$\vec{M}_z^{netto} = (N^P - N^{AP})\mu_z \hat{e}_z \tag{6}$$

within the tissue under consideration. In equilibrium, this magnetization is parallel to $\vec{B}_0$ and its strength is determined by the difference of numbers of protons in each energy state.

For a simpler understanding of MRI, it is helpful to have a more macroscopic view of the process. Therefore, multiples of protons can theoretically be grouped into so-called spin-packets, which are three-dimensional cuboids, containing several nuclei that experience equal magnetic field strength. Then, the net magnetization of each packet can still be described by equation (6). The only thing that changed is the amount of tissue under consideration.

Because the spins within a single spin-packet all precess at identical frequencies yet varied phases, the components perpendicular to the magnetic field, within the xy-plane, collectively nullify to zero on a statistical basis (see Fig. 2.3). Therefore, $\vec{M}_x^{netto} = \vec{M}_y^{netto} = 0$ and consequently $\vec{M}_{equilibrium}^{netto} = \vec{M}_z^{netto}$.

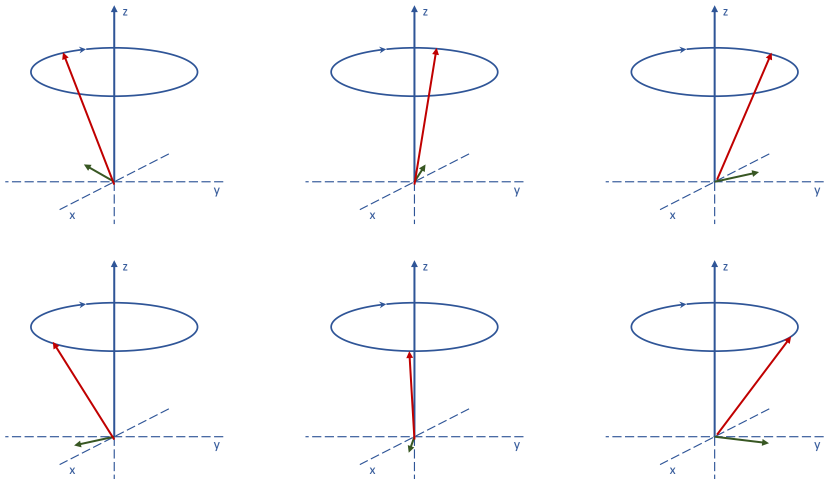


Figure 2.3 This diagram depicts a spin-packet containing six spins, each exhibiting identical precession cones but with distinct phases. Despite sharing the same precession frequency, the spins' varied orientations result in the cancellation of components within the xy-plane (indicated by the blue arrows), leading to zero net magnetization in that plane. Consequently, magnetization predominantly occurs along the z-axis, as it represents the statistical sum of the spin vector's z-components. Inspired by [30].

At this point, there needs to be emphasized that MRI techniques measure those net magnetizations of a spin-packet in a given volume of tissue and are not capable to detect individual nuclei. These signals are then depicted in the final MR image through structures known as voxels, which serve as three-dimensional counterparts to pixels.

The combination of equations (5) and (6) clarifies that lower temperatures and/or higher energy differences between parallel and anti-parallel configuration yield greater measurable net magnetization. However, lower temperatures with a significant effect on the net magnetization are not applicable to humans. Instead, higher magnetic fields are utilized, leveraging the Zeeman effect, where [32]:

$$\Delta E = \frac{h}{2\pi}\gamma B_0. \tag{7}$$

Subsequently, with equations (5) and (6) follows, $\vec{M}_z^{netto} \propto B$.

In order to detect a signal, it's necessary to disrupt the balanced state by administering a RF-pulse to the tissue. Consequently, a proton in a lower energy state can absorb a photon, becoming excited to a higher energy state. These protons in the heightened energy configuration can subsequently emit a photon with energy

equal to ΔE, which can be detected by a RF-receiver coil in the MRI scanner.
The most effective excitation can be achieved by a photon with frequency

$$\nu = \frac{\omega_L}{2\pi} = \gamma \frac{B_0}{2\pi} \tag{8}$$

For a typical MRI scanner with a magnetic field strength of 3T - as used for the acquisition of the images used in this work - this results in $\nu \approx 128$MHz.
The RF-pulse can be modified to flip the net magnetization by an arbitrary angle

$$\Theta = 2\pi\gamma t B_{pulse}, \tag{9}$$

where t is the duration and B_{pulse} the magnetic field component of the pulse.
It is fasctinating that this is achievable, despite the fact that the magnetic moments of each proton are limited to only two discrete values. This is mathematically justified by the Bloch equations, which introduce a semi-classical perspective on these spin systems, thus enabling the utilization of Euler's equations of motion for the precessing net magnetization vectors.
The essential flip-angles for the following explanations are 90° and 180°.
As stated before, the spins precess around the z-axis at the larmor frequency. After flipping the whole net magnetization by 90° into the x-y plane with an RF-pulse, this also starts to precess around the z-axis with ω_L, since the spins are precessing in phase and don't statistically cancel out each other in the x-y plane anymore. This motion of $\vec{M}_{net}$ within the examined tissue can induce an oscillating current in a nearby receiver coil, as per Faraday's law of electromagnetic induction.
This is directly followed by the two relaxation processes, where the energy transferred to the spin system dissipates into the surroundings through spin-lattice or spin-spin interactions and consequently, the spins revert to their original configuration (see Fig. 2.4). This causes the net magnetization to return to its initial state during a spiraling motion, leading to a decline in the signal detected by the receiver coil. This decline in signal is called the free induction decay (FID).

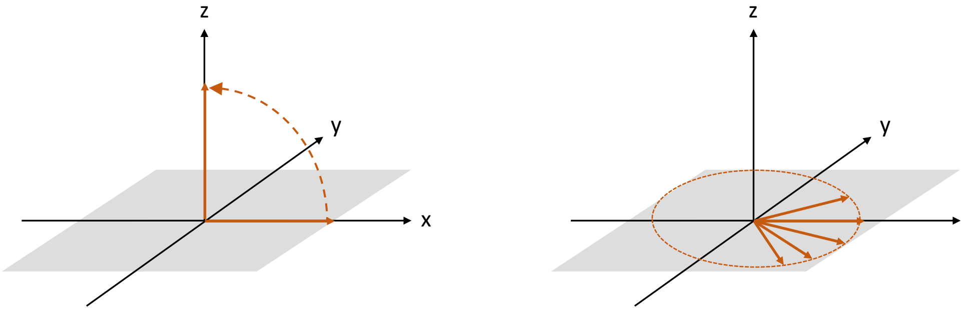


Figure 2.4 This figure is a schematic representation for the longitudinal (left) and transversal (right) relaxation of the net-magnetization for the co-moving coordinate system of the spin. Looking at it from the stationary coordinate system, the left picture would show a spiraling motion. Longitudinal relaxation is primarily influenced by interactions between spins and the lattice, while spin-spin interactions predominantly account for transverse relaxation.

One can focus either on longitudinal relaxation, primarily influenced by spin-lattice interactions, or transverse relaxation, mainly driven by spin-spin interactions, which is crucial for achieving the desired contrast in subsequent MRI images. The former is known as T1-recovery, and the latter as T2-decay.

T1 represents the time constant for longitudinal relaxation, indicating the duration required for the net magnetization to achieve $\frac{1}{1-e}$ of its initial value following the excitation. Formally, the longitudinal magnetization recovery over time is mathematically described as

$$M_z^{net} = M_0(1 - e^{-\frac{t}{T_1}}), \tag{10}$$

where $M_0 = \left|\vec{M}^{net}\right|$ represents the absolute value of the initial net magnetization before excitation.

Immediately after the excitation pulse with $\Theta = 90°$, the transverse net magnetization vectors of all associated spin packets within a specific voxel are in phase, which leads to the maximum transverse component of $\vec{M}^{net}$. The transversal relaxation process describes the decay of the xy-component of the net magnetization. This happens due to dephasing of the transverse net magnetization vectors of the spin-packets in a given voxel, mainly because of spin-spin interactions (see Fig. 2.5).

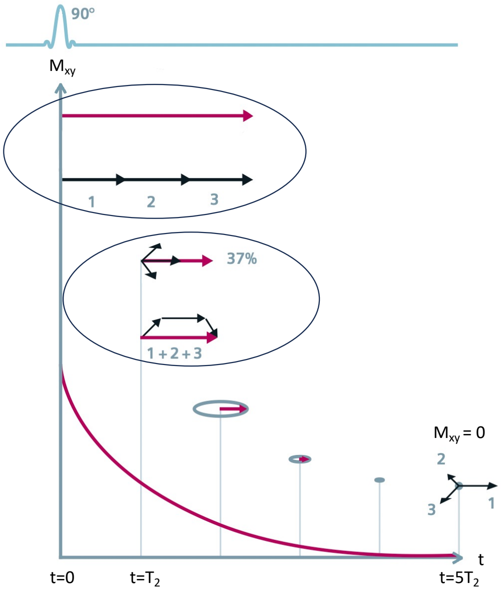


Figure 2.5 This figure is a schematic representation for the transversal relaxation of the net-magnetization of a spin-packet after excitation with a RF-pulse. Adapted and extended from [30], p. 80.

T2 is the duration required for the transverse net magnetization to decay to $\frac{1}{e}$ of its peak value. Its temporal progression of the transverse net magnetization is represented by:

$$M_{xy}^{net} = M_0 e^{-\frac{t}{T_2}}. \tag{11}$$

However, during experiments, the FID diminishes at a faster rate, given by the effective time constant T_2^*. Rather than solely accounting for spin-spin interactions, as done by T2, field inhomogenities are also encompassed. Because of those inhomogenities, ω_L differs for the spins, since it depends on the local magnetic field strength (see (4)) and, in turn, leads to a faster dephasing.

Therefore, the decay after the initial 90° excitation pulse is too fast to serve as the signal for image reconstruction. To capture a meaningful signal, a technique called rephasing is employed. This technique involves realigning the dephased net magnetizations of the spin-packet by applying a second excitation pulse with a flip angle of 180 degrees. By doing this, the dephasing process is reversed and the transverse net magnetization vectors gradually decrease their phase difference again, hence they rephase. This process leads to the generation of another MR Signal, the so-called spin echo [35].

Spin Echo

An excitation pulse with a flip angle of 90 degrees occurs at $t = 0$. Subsequently, another pulse with $\Theta = 180°$ is applied at $t = \tau$. The rephasing process matches the duration of dephasing up to that point. Therefore, the maximum spin echo signal can be detected at $t = 2\tau$, denoted as the echo time (TE). However, the spin echo signal is always weaker than the FID. Its peak strength is determined by the T2-relaxation curve [36, 35]. This phenomenon is illustrated in figure 2.6.

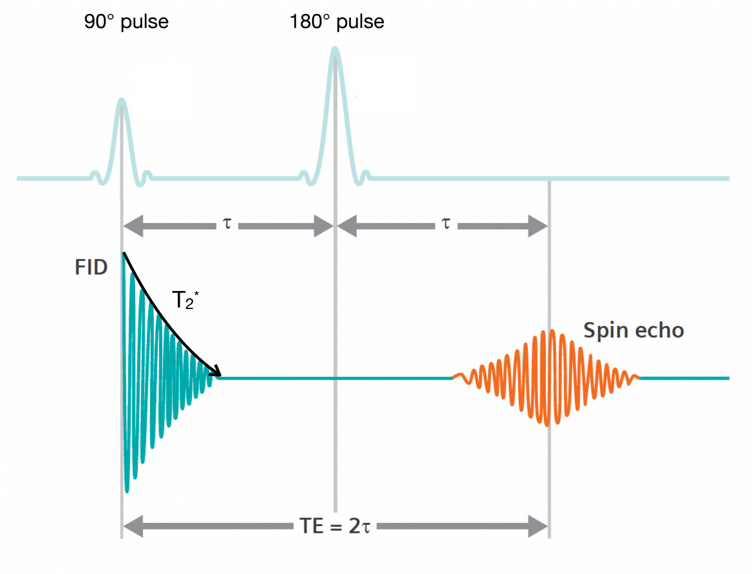


Figure 2.6 This figure illustrates the most basic spin echo sequence. The initial 90° excitation pulse is followed by another with a flip angle of $\Theta = 180°$. Adapted and extended from [31], p. 90.

Furthermore, this principle can be extended to multi-echo sequences. In such sequences, successive 180°-pulses are applied to the tissue, until transverse magnetization is completely decayed according to the T2-relaxation [36, 35]. With this approach, one can obtain several images of the same location with different T1/T2-weighting, all without increasing the overall acquisition time [31].

The outlined theory of MRI physics serves as the foundation for acquiring NMR signals. However, additional steps are needed to be able to create spatially resolved images, because according to the pre-described theory, all spins in the scanner perceive the same B-field, thus they posess the same larmor frequency. Therefore, all spins are excitated with the same RF pulse and afterwards they send an equivalent NMR signal, regardless of their location in the scanner. This uniformity would make it impossible to distinguish between signals from different spatial locations. The first step in resolving this issue is the utilization of gradient fields, which were first mentioned by Lauterbur in 1973 [37].

Gradient Coils

Gradient coils can apply a linear magnetic field along a given axis and overlay with the static field $\vec{B}_0$. Therefore they can induce local variations in the magnetic field and subsequently a local change in ω_L (See eq. (4)) [37].
Having a gradient coil in z-direction allows for spatial resolution of the image in this dimension. The gradient coil must be switched on simultaneously with the RF-pulse for the excitation of a single slice, rather than the whole subject. The slice thickness and thus the resolution can be determined by the increase in intensity of the gradient field along the given axis. For a stronger/steeper increase, the slice will be thinner (see Fig. 2.7).
Therefore, the gradient coil in z-direction is referred to as the slice selecting gradient coil (G_s) [31].

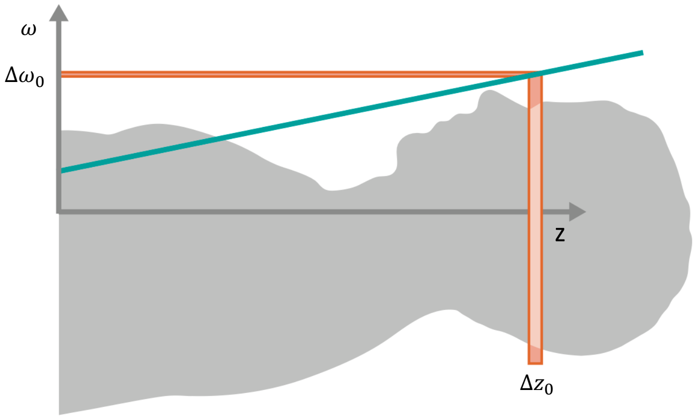


Figure 2.7 This figure illustrates the effect of the slice selection gradient on the perceived linear magnetic field (green line). This leads to an excitation frequency width $\Delta\omega_0$ and the resulting selected slice in z-direction with its thickness Δz_0 in a given subject. Adapted from [31], p. 101.

By utilizing the slice-selecting gradient coil, one of the three desired spatial dimensions is resolved. Consequently, two additional gradient coils are needed to

resolve the remaining dimensions, one for the x-axis and one for the y-axis.
To encode the x-direction, another gradient coil, which also introduces a linear dependency of the Larmor frequency in the x-direction, must be activated. This gradient field is switched on during signal acquisition, giving each voxel in a given row along the x-direction a unique Larmor frequency as its precession frequency. Hence, this gradient is called the frequency encoding gradient (GF). The intensity and spatial information of this row of voxels are encoded by applying a Fourier transformation to the received mixture of signals from all the voxels. This process allows to encode the information for each individual voxel.
For the final dimension, the y-direction, a different method is required to achieve spatial encoding, as otherwise, at least two voxels with the same Larmor frequency would appear and be indistinguishable. Therefore, a phase encoding gradient coil (GP) is utilized. This gradient coil is activated briefly between the pulse and signal acquisition stages. This causes spins to precess at different speeds for a short period. After deactivation of the gradient, spins exhibit varying phase shifts directly corresponding to their positions along the y-direction. Subsequently, these phase shifts can be encoded. In conclusion, Fourier transformation is applied to obtain both the intensity and spatial information from each voxel. These values can then be converted into a series of grayscale images. Finally, those can be stacked to form the conventional MRI volume, which is used in daily medical applications.

2.4.2 MRI Sequences

All images used for clinical diagnostic purposes must display a clear contrast between any pathology and normal anatomical features. Without this contrast, physicians cannot detect abnormalities within the examined tissue.
In MRI images, the contrast is influenced by intrinsic and extrinsic parameters. The intrinsic parameters are fundamental properties of tissues that affect their MR-signal and cannot be influenced. However, the extrinsic parameters, like field strength, imaging sequences or contrast agents can be modified, hence this subsection will focus on those.
One can obtain a high signal of a specific tissue, if there's a large transverse component of coherent magnetization at t=TE, resulting in a bright voxel in the image. T1 and T2 relaxation times vary across different tissues, which can be utilized to create different contrasts in the images (see table 2.1). With higher magnetic fields, T1-relaxation is increasing, whereas T2-relaxation almost stays the same [38, 39, 40, 41, 42].

Table 2.1 This table shows the approximate T1 and T2 relaxation times of different tissues at 3T in milliseconds at body temperature. Data from [43].

Tissue	**T1 time [ms]**	**T2 time [ms]**
Water	4000	2000
Fat	250	70
Muscle	900	50
Liver	500	40

To get the desired contrast - in the following referred to as sequences or modalities - specific values for TR and TE need to be selected for a given pulse sequence. As a result, the image is weighted to emphasize a particular contrast mechanism.

In this study, two fundamental types of contrast will be explained: T1-weighted and T2-weighted contrasts. Following that, contrast enhancement via the injection of a contrast agent and a specific T1-weighted sequence known as the T1-VIBE sequence will be discussed, due to its significance for the main objective of this thesis.

T1-weighted Images

Images in which the contrast primarily relies on the variations in T1 times between different tissues - primarily fat and water - are referred to as T1-weighted. By adjusting the repetition time (TR), it is possible to control the recovery of the transverse component of magnetization for each tissue before the subsequent excitation. Therefore, TR has to be short enough that the net magnetization from fat and water can't completely return to align to $\vec{B}_0$, within that timeframe.

Optimal contrast is achieved at the point where the difference in longitudinal magnetization between the two tissues is greatest. T1-weighted images are characterized by bright signals in fatty tissues, whereas water, lesions such as cysts or tumors and veins appear dark. An example for a standard T1-image is illustrated in figure 2.8 (a).

T2-weighted Images

Given that water possesses a much longer T2 relaxation time compared to fat (see table 2.1), the transverse magnetization component of fat decays more rapidly. This rapid decay results in a diminished signal from fat at later time points. T2 weighting is primarily influenced by the echo time (TE), which determines the extent of T2 decay occurring before the signal acquisition. Thus, TE plays a crucial role in defining the degree of T2 contrast in the resulting image.

T2-weighted images are characterized by bright signals in tissues with high water content, as inflamed liver tissue, cysts and some types of tumors and fat and fibrous or cirrhotic tissues appear darker due to its lower water content compared to healthy liver tissue.

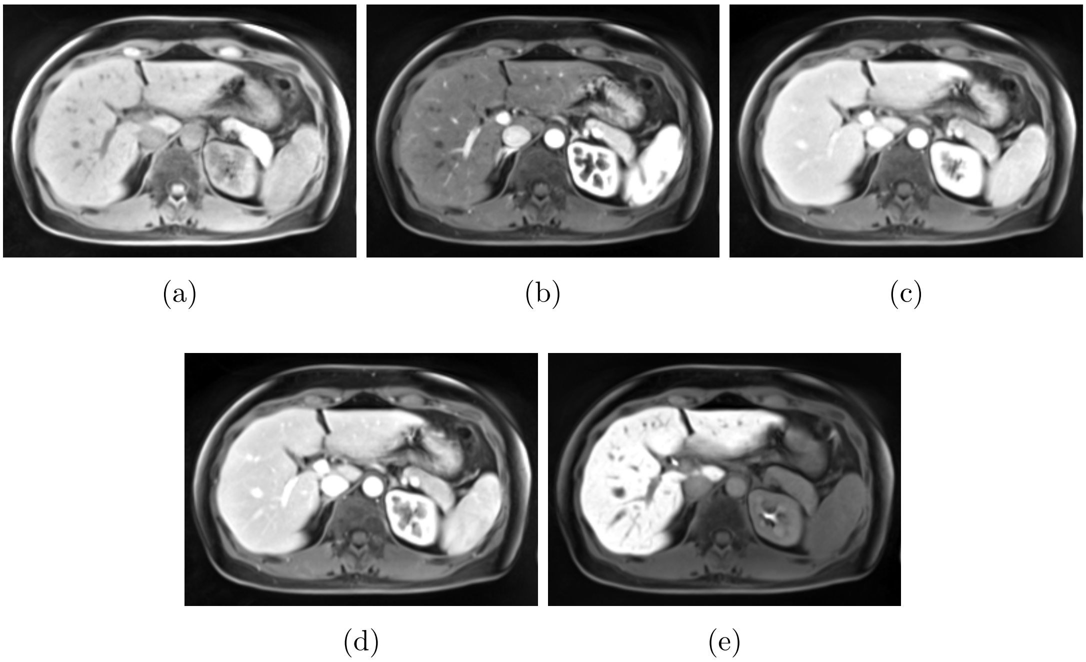
Contrast Enhancement

Contrast enhancement can be achieved by injection of a contrast agent that is capable of shortening the T1-relaxation time in tissues. For the liver, there are mainly two hepatocyte-selective contrast media, namely *gadobenate dimeglumine* (Gd-BOPTA) and *gadoxetic acid* (Gd-EOB-DTPA), also known as Primovist in Europe or as Eovist in the USA [44, 45].
Both of them contain Gadolinium, which is a paramagnetic ion with unpaired electrons, creating a local magnetic field. This can interact with nearby water protons in the body and shorten their T1 relaxation time. Because of this, tissues where it accumulates appear brighter on T1-weighted MRI images [46].
Primovist is directly specialized for liver imaging, with significant hepatocyte uptake and equal renal and biliary excretion [47]. With this, it can provide both morphological and functional liver imaging and thus it the choice for MRI imaging of patients with liver diseases in our clinic. The dynamic contrast enhancement of it can be seen in figure 2.8, when comparing the native T1 image **(a)** to the ones during and after contrast agent injection **(b)** - **(e)**.

VIBE-sequence

The T1-VIBE (T1-Volume Interpolated Breath-hold Examination) sequence is frequently used in MRI for high-resolution liver-specific imaging because of its rapid acquisition and excellent spatial resolution. This makes it ideal for dynamic abdominal imaging, including liver scans using contrast agents like Primovist. Various methods can be used with this sequence to selectively suppress fat signal. For the datasets in this thesis, the spectral fat suppression was used, which utilizes the different resonance frequencies between water- and fat-bound protons. A narrow-band frequency-selective RF-pulse excites mainly protons bound in fat and their transversal magnetization is destroyed with spoiler gradients, thus no fat magnetization can be measured. This helps in optimizing the visualization of the liver parenchyma and improving the detection of lesions and vascular structures during MRI examinations with contrast agents [48, 49, 50].
In T1-VIBE sequences, multiple thin sections of the body are acquired during a single breath-hold. This reduces motion artifacts and enables for detailed visualizations of anatomical structures and dynamic processes such as perfusion.

Five images are acquired sequentially, reflecting the dynamic contrast enhancement of the liver, caused by the Primovist. The five phases are called *native*, *arterial*, *late arterial*, *portalvenous*, *hepatobiliary late phase* (HBP20). All of them are illustrated for one patient of the cohort in figure 2.8 **(a)** - **(e)**. The first phase is acquired prior any application of Primovist and the last one ist typically acquired about 20 minutes after injection of the contrast agent [49, 50].



(a)

(b)

(c)

(d)

(e)

Figure 2.8 This figure demonstrates the dynamic contrast enhancement of a T1-VIBE sequence for one subject in the cohort. From **(a)** to **(e)**, one can see the native, arterial, late arterial, portalvenous and hepatobiliary late (HBP-20) phase.

2.5 Artificial Intelligence

Finding and precisely delineating damaged tissue within the liver and segmenting the whole organ for an estimation of survival is a crucial step in surgical planning. Liver volumetry is done both automated and manually. While manual segmentation in medical images is still considered as the gold standard and almost always used as a baseline for performance evaluation of a segmentation algorithm, there are a lot of downsides to it. It requires a highly experienced physician, is very prone to inter- and intra-rater variability and it's very tedious and time-consuming on three dimensional MRI images [51, 52, 53]. Hence, there's a lot of motivation to build and use automated techniques like neural networks or transformer models for tasks like that. After receiving automatically created segmentation maps of the tissue under observation one can use those to extract meaningful features of the MRIs to train machine learning models for liver function estimation, as deployed in this thesis.

To understand the methods used in this work, the basic principles of conventional machine learning (ML) algorithms and advanced deep learning (DL) models, which are both subcategories of artificial intelligence (AI), will be explained in the following [54].

If not stated differently, the following subsections are mainly obtained from [55, 56] and [57].

2.5.1 Conventional Machine Learning

Conventional machine learning (ML) encompasses a range of algorithms and techniques used for classification, regression, clustering, and other tasks. These methods can be broadly categorized into *supervised* and *unsupervised learning.*

Supervised Learning involves training a model on labeled data, where not only the input features, but also the corresponding output labels are known. The goal is to learn how the inputs can be mapped to the outputs and use this to generate predictions on new, unseen data. Examples include classification algorithms like Random Forest [58] and Gradient Boosting [59, 60], as well as regression techniques [61, 62, 63] and Support Vector Machines [64, 65].

On the other hand, **Unsupervised Learning** deals with unlabeled data. The objective is to identify patterns or structures within the data. Common unsupervised learning techniques include clustering algorithms such as K-means [66] and hierarchical clustering [67], and dimensionality reduction methods like Principal Component Analysis (PCA) [68].

In the context of medical imaging, conventional ML techniques have been widely

used for tasks such as tumor detection, organ segmentation, and disease classification. Commonly, cross-validation is employed to ensure the model performs well on unseen data [69, 70, 71, 72, 73, 74].
For the liver function estimation in this thesis, meaningful features are extracted from the MRI images with automated segmentations, generated by deep learning models, and subsequently those are used to build and train predictive machine learning models with supervised learning. Those employed methods will be explained in detail in the following.

Random Forest

The random forest algorithm is a basic, but powerful, ensemble machine learning method that was introduced by Leo Breiman in 2001 [58]. It can be used for multiple tasks, including classification, regression and clustering. The method operates by building numerous decision trees during the training phase, typically numbering in the hundreds or thousands, hence the name *forest*. Every tree is built on a different randomly selected subset of the training data with a technique called *bagging* [75] (bootstrap aggregating), introducing further diversity among the trees. Random forests further enhance this diversity by randomly selecting a subset of features to consider at each node split, a technique known as *feature bagging* [76]. This combination of bagging and random feature selection helps mitigate overfitting, a common problem in machine learning, where the applied method performs good on the training dataset but poor on unknown data. In other words, the method is not able to generalize well. The final prediction is determined by aggregating the predictions of all trees, either through averaging (for regression) or majority voting (for classification) [58]. A schematic illustration of the Random Forest is given in figure 2.9.

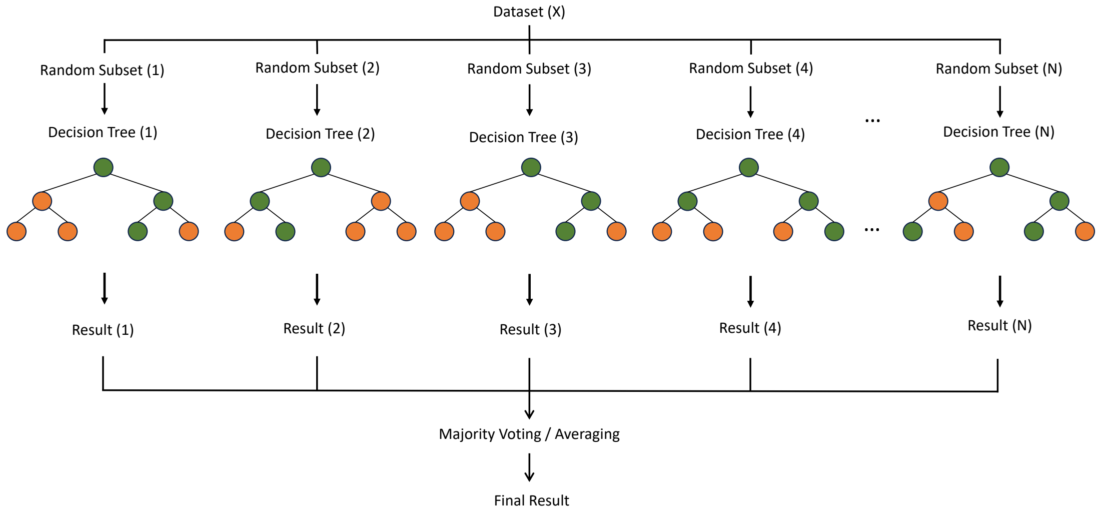


Figure 2.9 This figure illustrates a basic Random Forest. The dataset is split into random subsets, and based on each of those, a decision tree is built and trained. Depending on the task, the results from each of the N decision trees are then either averaged (for regression tasks) or majority voted (for classification tasks) to get the final result.

Random forests are popular because of their ability to handle high-dimensional data and their high accuracy, robustness to outliers and noise. On top of that, they can also provide measures for feature importance, making them suitable for feature selection and interpretation of the results. However, they can be computationally intensive and may struggle with imbalanced datasets [77, 78, 79, 80, 81].

Gradient Boosting Algorithms

Another basic machine learning approach for classification and regression tasks are *Gradient boosting* algorithms. They are based on the foundational work of Freund and Schapire, who introduced the *AdaBoost* algorithm in 1995 [59]. AdaBoost was a significant milestone in machine learning applications and laid the groundwork for subsequent advancements in boosting techniques. Gradient boosting specifically, as further developed by Friedman in 2001, extends these concepts by sequentially building an ensemble of weak learners, typically decision trees, to improve predictive accuracy through gradient descent optimization [82]. A highly scalable and efficient implementation of such an algorithm is XGBoost (eXtreme Gradient Boosting), as introduced by Chen et.al. in 2016 (see Fig. 2.10). This incorporates various optimizations such as *tree pruning* and *regularization*, which makes it particularly effective for competitive machine learning tasks and large datasets [60]. *Tree pruning* is used, which is a method, where the model

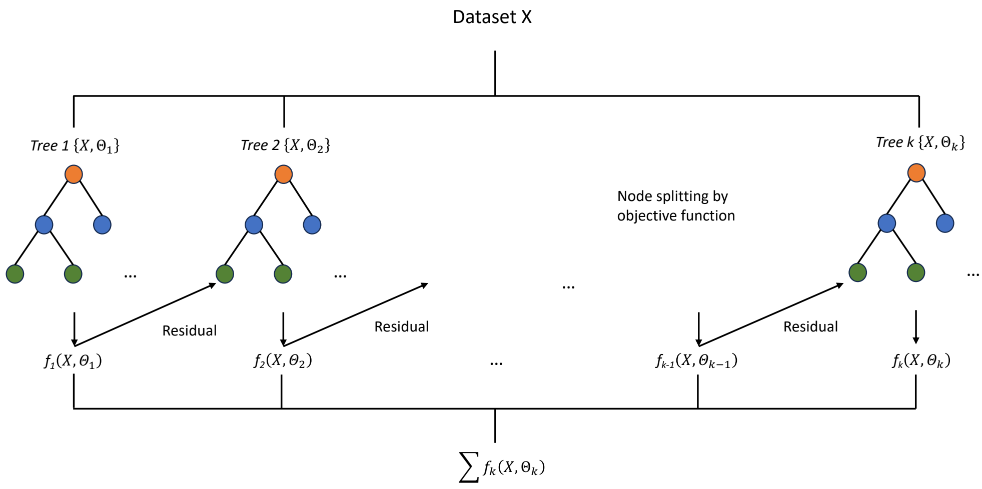


Figure 2.10 This figure is a flow chart of XGBoost. Each tree depends on the residual of its predecessor. The trees work with the input X and they learn a parameter Θ during the training. The final prediction is received by summation over all outputs.

is simplified by removing sections of the tree that are non-critical or redundant for classifying instances [83]. *Regularization* is employed in two ways. L1 (*Lasso* (least absolute shrinkage and selection operator)) regularization is used as well as L2 (*Ridge*) regularization. A more detailed explanation of these techniques can be found in the following paragraph about regression models. On top of those optimizations, parallel processing is also employed in the XGBoost algorithm. During training, a weight parameter Θ for each tree is implicitly learned.
Contrary to the Random Forest, where each tree is constructed independently using bagging, the XGBoost algorithm builds trees sequentially. There, every new tree tries to correct the errors made by the previous ones, and thus enhancing the model's accuracy through gradient boosting (see Fig. 2.10) [60]. With this sequential approach, XGBoost is able to reduce both bias and variance more effectively than Random Forest [84, 85, 86]. XGBoost's advanced optimization features along with its excellent predictive performance, make it a widely used choice for achieving high accuracy in complex tasks, particular in structured or tabular scenarios [87, 88, 89, 90].

Regression Models

In machine learning, regression analysis is a fundamental technique for modeling relationships between variables.

Linear regression is the simplest form of a regression model. It tries to establish a linear relationship between dependent variables $\vec{y}$ (target) and independent variables $\vec{x}$ (predictors) with the equation:

$$\vec{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_n x_n + \epsilon. \tag{12}$$

There, β_0 is the intercept, e.g. the value, where the linear regression line crosses the y-axis and $\beta_1, ..., \beta_n$ are the coefficients of the independent variables. ϵ denotes the error or disturbance term that accounts for differences between the observed values of the dependent variable $\vec{y}$ and the values predicted by the regression model.

Loss functions play a crucial role in measuring how well a model's predictions match the actual data in machine learning. For regression tasks, one common loss function is the Ordinary Least Squares (OLS), which minimizes the sum of squared residuals:

$$\text{Cost function}\left(\vec{y}^{true}, \vec{y}^{pred}\right) = \sum_{i=1}^{p} \left(\vec{y}^{true} - \vec{y}^{pred}\right)^2, \tag{13}$$

where $\vec{y}^{true}$ refers to the actual value and $\vec{y}^{pred}$ is the predicted value. The goal in training regression models is to minimize this loss function, thereby improving the model's accuracy [91, 62, 61].

However, to improve model generalization and adress overfitting, regularized variants of linear regression have been developed.

Lasso Regression (L1 regularization) is one commonly used of those. It adds a penalty term to the linear regression's cost function based on the absolute values of the coefficients $\vec{\beta}$:

$$\text{Lasso cost function}\left(\vec{y}^{true}, \vec{y}^{pred}\right) = \sum_{i=1}^{p} \left(\vec{y}^{true} - \vec{y}^{pred}\right)^2 + \lambda \sum_{j=1}^{n} |\beta_j|, \tag{14}$$

where λ is referred to as the regularization parameter. With this, some coefficients can be shrunk or even set to zero and therefore less important features are eliminated from the model, thus some sort of automatic feature selection is performed. This type of regression is particularly useful for models with high-dimensional data, where automatic feature selection is beneficial [92, 62].

Ridge Regression (L2 regularization) on the other hand adds a penalty term to the linear regression cost function based on the squared magnitude of the coefficients:

$$\text{Ridge cost function}\left(\vec{y}^{true}, \vec{y}^{pred}\right) = \sum_{i=1}^{p}\left(\vec{y}^{true} - \vec{y}^{pred}\right)^2 + \lambda\sum_{j=1}^{n}\beta_j^2. \tag{15}$$

Contrary to *Lasso*, *Ridge* does not set any coefficients to zero but shrinks them toward zero with the regularization parameter λ. This helps to reduce model complexity and multicollinearity. This type of regression is particularly useful for multicollinear data, because it is addressing the issues of highly correlated predictors and thus allows for more stable and interpretable models [93, 62, 61].

Elastic Net Regression combines both L1 and L2 regularizations, thus offering a balance between coefficient shrinkage and feature selection. Mathematically, it can be described by:

$$\text{Elastic Net cost function}\left(\vec{y}^{true}, \vec{y}^{pred}\right) = \sum_{i=1}^{p}\left(\vec{y}^{true} - \vec{y}^{pred}\right)^2 + \lambda\sum_{j=1}^{n}\beta_j^2 + \alpha\sum_{j=1}^{n}|\beta_j|. \tag{16}$$

Those regularization techniques can help in handling mutlicollinearity and high-dimensional data [94].

The choice of regularization often depends on the specific dataset and problem and it is always good practice to test them against each other. The regularization strength of those models depends on the parameters λ or α and can be optimized through cross validation for achieving the best trade-off between predictive performance and model complexity [63, 95].

Logistic Regression is a technique for binary classification problems. Unlike linear regression which predicts continuous values, logistic regression estimates the probability that an instance belongs to a particular class. The logistic function (also called sigmoid function) is used to map predictions to probabilities between 0 and 1:

$$P(y = 1|\vec{x}) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \ldots + \beta_n x_n)}} \tag{17}$$

Where $P(y = 1|\vec{x})$ is the probability that y belongs to class 1 given input $\vec{x}$, and β_i are the model parameters. The cost function used for logistic regression is the log loss or cross-entropy:

$$\text{Logistic Regression Cost function} = -\frac{1}{m}\sum_{i=1}^{m}[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)] \tag{18}$$

Where y_i is the true class and $\hat{y}_i$ is the predicted probability. Like linear regression, logistic regression can also be regularized using L1 and/or L2 penalties to prevent overfitting. Logistic regression is particularly useful for interpretable models and when the relationship between input features and output probabilities is approximately linear. It serves as a foundation for more complex classification algorithms [62].

Support Vector Machines

Support Vector Machines (SVMs) are robust supervised machine learning algorithms used for classification and can be extended to so called Support Vector Regression (SVRs) for regression tasks. They are based on the Vapnik-Chervonenkis theory, which was developed from the 1960s to the 1990s [64, 65]. While SVMs aim to find the optimal hyperplane that maximally separates different classes in the feature space, SVR tries to find a function that deviates from the actual target values by less than a pre-defined margin ϵ [96].
Schematic illustrations of those approaches can be found in figures 2.11 and 2.12, respectively.

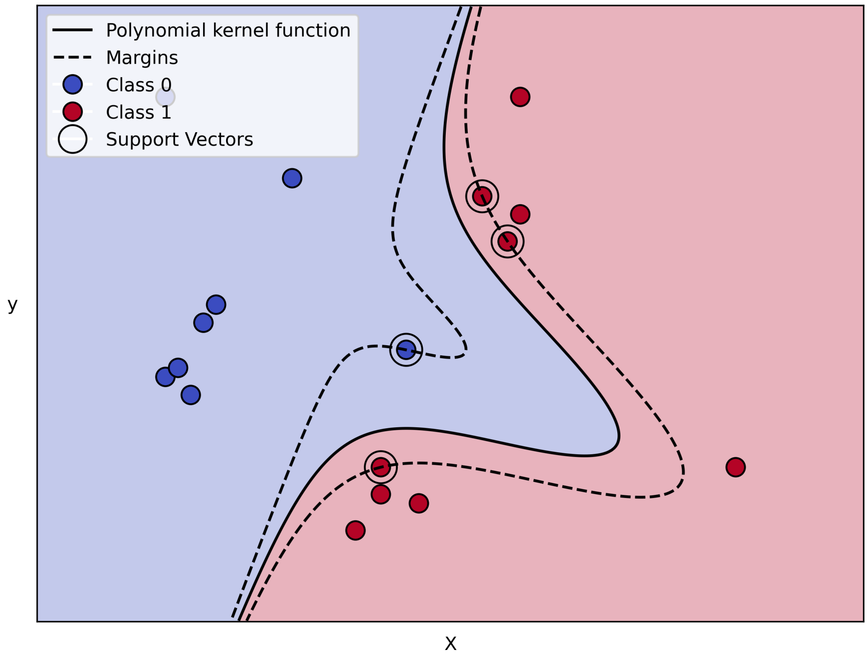


Figure 2.11 Schematic illustration of classification boundaries for a Support Vector Machine with a polynomial kernel of degree 3.

Both techniques employ the so-called *kernel trick*, allowing them efficient computation of dot products in high-dimensional feature spaces without the need to explicitly compute the coordinates. With this, they are able to handle non-linear relationships by implicitly mapping input data into a higher dimensional space. Contrary to explicitly transforming the data, the *kernel trick* approximates inner

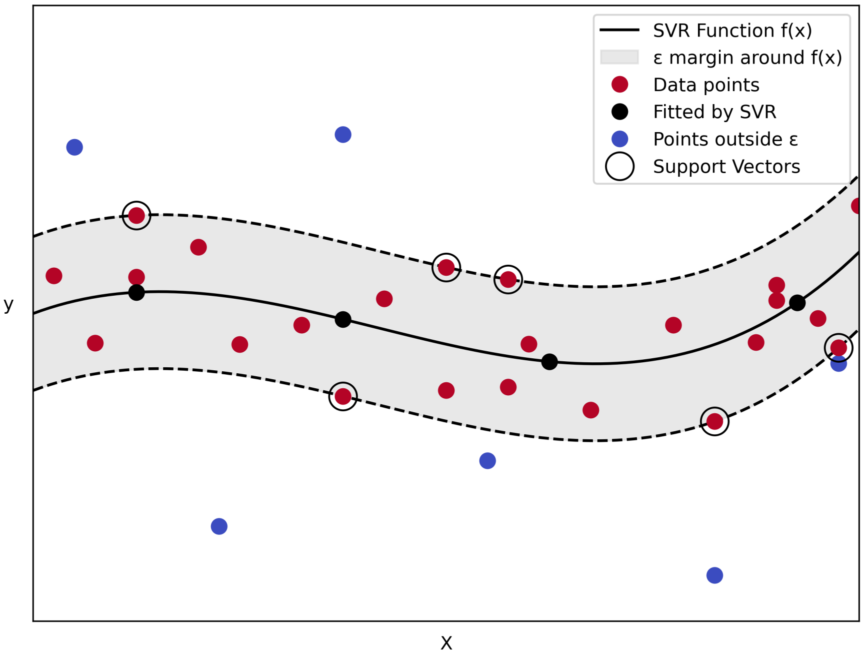


Figure 2.12 Illustration of Support Vector Regression for a polynomial kernel of degree 3.

products in the transformed feature space directly with a *kernel function* $k(x_i, x_j)$, simplifying the computations significantly. A basic example for such a function would be the polynomial kernel [97]:

$$k(\vec{x}_i, \vec{x}_j) = (\vec{x}_i \cdot \vec{x}_j + 1)^d , \tag{19}$$

which allows the SVMs to create a complex decision boundary without the need for explicit mapping, thus making the algorithm computationally efficient.

In its basic form, SVMs are designed to solve binary classification problems, by finding the optimal hyperplane that separates both classes. However, this can be extended to multiclass classification problems with several techniques. The most common are *One-vs-Rest* (OvR) and *One-vs-One* (OvO).

For the **One-vs-Rest** method, a separate binary classifier is trained for each class, where each classifier differentiates between its class and all other classes combined. This results in **K** binary classifiers for **K** classes. The final prediction is chosen based on the highest confidence score for a given class [98, 99].

The **One-vs-One** approach trains a binary classifier for every pairwise combination of classes. This results in $\boldsymbol{\frac{K(K-1)}{2}}$ classifiers. Each classifier is trained just to distinguish between two classes, ignoring all others. During prediction, the final output is received by majority voting of all predictions from all classifiers [100, 99].

Those strategies allow SVMs to handle multiclass-classification problems.

2.5.2 Neural Networks

Neural networks are inspired by the learning processes of the human brain that consists of neurons interconnected by axons and dendrites. Those are then connected by synapses. The strength of the synaptic connections can be altered by external stimuli. Basically, this represents the learning process in living organisms. Artificial neural networks - referred to simply as neural networks (NNs) in this thesis - attempt to mimic these biological mechanisms. The computational units, also called neurons, are connected by weights that resemble the synaptic connection strengths in the human brain. The network processes input data by transmitting values from input nodes to output nodes, effectively calculating a function of the provided inputs.

Unlike for traditional programming, where one has input data and must explicitly code the instructions to achieve the desired output, supervised deep learning (DL) applications use both input and output data for a given training subset, as explained in the previous subsection. Both datasets are fed into the neural network and in an ideal scenario, the network learns to extract features from the input data by itself, to generate the output data [54]. This process is referred to as the training process, where the best values for the weights can be determined and are adjusted accordingly. The input-output pairs used in this process are analogous to the external stimuli required for learning in biological organisms.

For this thesis, those pairs are represented by voxel representations of three-dimensional patches of the abdominal MRI scan and their corresponding label maps. Those maps maintain the same voxel spacing and dimensions as the original MRI images, where voxels overlapping with the liver are assigned a value of one, those overlapping with a lesion have a value of four, and the remaining six tissues of interest are similarly annotated with other values in those volumes. These labels are crucial for the network to know where it has to look for the important features in the MRI image, to be able to extract and learn those.

The patches and labels are fed to the network, where it tries to predict the voxels corresponding to the tissues of interest, using the three-dimensional patches from the original MRI image. The labels provide feedback to the network regarding the accuracy of its predictions. As a result, the network adjusts its weights to better match its predictions more closely with the annotations in the label maps. This process is performed after each iteration, resulting in enhanced model performance, if the training parameters are appropriately configured. With a large and heterogeneous dataset, the network can learn various representations of what it should detect, reducing the difficulty of identifying unknown data. This phenomenon, is known as model generalization. With increasing dataset size, deep learning models are able to achieve much higher accuracy compared to conventional machine learning models.

Perceptrons

The most basic neural network was invented by Rosenblatt in 1958 and is called a perceptron [101]. It consists of a single input layer and an output node (see Fig. 2.13). In the input layer, a series of inputs is directly fed into the network. These inputs are - depending on the network's activation function $\Phi(\cdot)$ and the weights - mapped to an output. Computation occurs only in the output layer without any feedback loop from the output back to the input layer. Such a network is operating in what is referred to as a feed-forward mode. Because of this, the predicted output signal $\vec{y}^{pred}$ can be described by:

$$\vec{y}^{pred} = \Phi(\vec{\nu}) = \Phi\left(\sum_{k,n} w_{k,n} x_n\right) . \tag{20}$$

Here, $\vec{\nu}$ represents the local field, $w_{k,n}$ denotes the edge weights and x_n are the input signals.

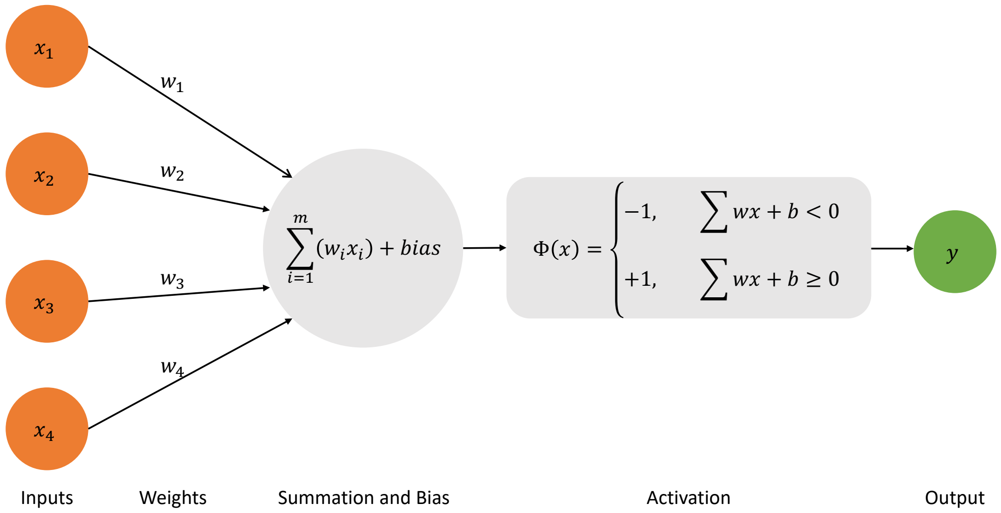

x_1
x_2
x_3
x_4
w_1
w_2
w_3
w_4
$\sum_{i=1}^{m}(w_i x_i) + bias$
$\Phi(x) = \begin{cases} -1, & \sum wx + b < 0 \\ +1, & \sum wx + b \geq 0 \end{cases}$
y
Inputs
Weights
Summation and Bias
Activation
Output

Figure 2.13 Schematic representation of a single-layer perceptron. The orange x's are the input neurons, which are connected to the green output neuron (y) via their corresponding weights and an activation function. There, the sign function was used as activation function, which decides, whether the output neuron will be activated or not, based on the weighted sum of the input neurons.

An simple task like binary classification in combination with the sign activation function (see Fig. 2.13) is well suited for a basic explanation. There, the summation is mapped to the value +1 or -1 and therefore, the weighted sum of the inputs can be converted into a class label. Typically, some part of the prediction remains invariant and is referred to as bias b_n. The combination of this with (20) leads to:

$$\vec{y}^{pred} = sign(\sum_{k,n} w_{x,n} x_n + b_n). \tag{21}$$

For further explanations the bias term will be neglected and hence $b_n = 0$. The prediction's error for this is calculated by

$$\vec{E}(\vec{y}^{true}, \vec{y}^{pred}) = \vec{y}^{true} - \vec{y}^{pred}. \tag{22}$$

Combined with prior explanation, this can possess a value from the set $\{-2; 0; +2\}$. Such a neural network is capable of linearly splitting the hyperspace with its activation function, hence it is able to learn problems that can be linearly separated. To overcome the limitations of a single-layer perceptron and be capable of solving nonlinear problems, multilayer perceptrons are introduced. Basically, those are several serially connected single-layer perceptrons, also operating in the feed-forward mode. The first and last layer are called the input- and the output-layer, respectively. All layers in between are referred to as hidden-layers. The width of a a layer is determined by the number of neurons in it, while the number of layers defines the depth of the neural network. A schematic representation of a multilayer perceptron with three hidden layers, each having a width of four, is shown in figure 2.14. All the layers are fully connected, meaning that each neuron in layer *n+1* is connected to every single neuron in layer *n*. In the following text, matrices are represented by capital letters with arrows, whereas vectors are denoted by lowercase letters with arrows.

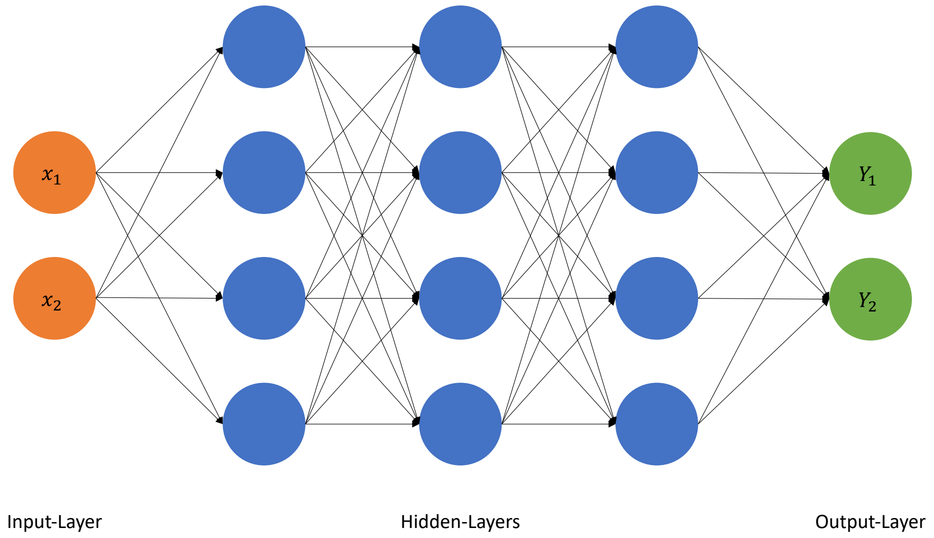


Figure 2.14 Multi-layer perceptron with three hidden layers, two input neurons, and two output neurons. The orange x's are the input neurons, which are fully connected to the blue neurons in the hidden layers via their corresponding weights and an activation function. The arrows indicate the feed-forward process. The hidden layers are fully connected with the green output neurons.

Denoting the k weights between the k neurons in the *m-th* and the n neurons in the *(m-1)-th* layer as a matrix, leads to:

$$\vec{W}^{(m)} = \begin{bmatrix} w_{0,0}^{(m)} & \dots & w_{0,n}^{(m)} \\ w_{1,0}^{(m)} & \dots & w_{1,n}^{(m)} \\ \vdots & \vdots & \vdots \\ w_{k,0}^{(m)} & \dots & w_{k,n}^{(m)} \end{bmatrix}. \tag{23}$$

The same can be done with the inputs. Therefore, the input to the neurons in the first hidden layer can be calculated with:

$$\vec{h}^{(1)} = \begin{bmatrix} h_0^{(1)} \\ h_1^{(1)} \\ \vdots \\ h_n^{(1)} \end{bmatrix} = \vec{W}^{(0)} \cdot \vec{x}^{(0)} = \sum_{k,n} w_{k,n}^{(0)} \cdot x_n^{(0)}. \tag{24}$$

With this, the output of the k neurons in the first hidden layer to the n neurons in the second one is:

$$\vec{a}^{(1)} = \Phi\left(\vec{h}^{(1)}\right) = \begin{bmatrix} \Phi(h_0^{(1)}) \\ \Phi(h_1^{(1)}) \\ \vdots \\ \Phi(h_n^{(1)}) \end{bmatrix} = \Phi_0\left(\sum_{k,n} w_{k,n}^{(0)} \cdot x_n^{(0)}\right), \tag{25}$$

the same as the output of a single layer perceptron (see Eq. (20)).
Generalized, the output of the (i-1)-th layer with k neurons to the j neurons in the i-th layer is:

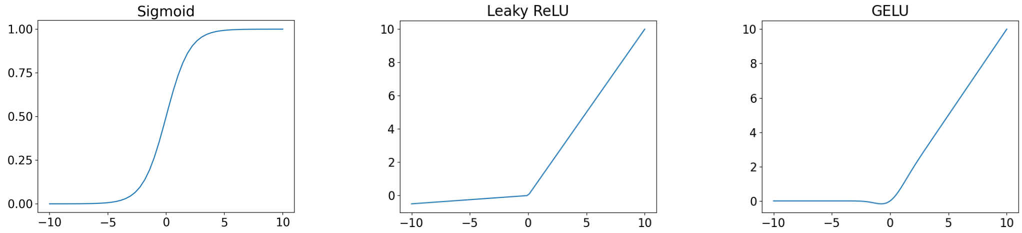
$$\vec{a}^{(i)} = \Phi_{i-1}\left(\vec{W}^{(i-1)} \cdot \vec{a}^{(i-1)}\right) = \Phi_{i-1}\left(\sum_{j,k} w_{j,k}^{(i-1)} \cdot \left(\Phi_{i-2}\left(\sum_{k,n} w_{k,n}^{(i-2)} \cdot a_n^{(i-2)}\right)\right)\right). \tag{26}$$

By concatenating multiple layers, the network becomes capable of solving nonlinear problems. However, to achieve this, the activation function must be nonlinear, because a composition of linear functions would still result in a linear function. Some relevant activation functions are described in the following subsection.

Activation Functions

There are various activation functions to choose from, each playing a crucial role in neural networks by influencing the model's capacity to capture nonlinear relationships within the data. The sigmoid activation function, characterized by its S-shaped curve (see Fig. 2.15 (a)), is particularly effective when employed in the output layer for binary classification tasks. Its output can be directly interpreted as the probability of a given voxel belonging to the desired class.
In the context of multiclass classification, such as in this thesis, the softmax activation function emerges as a natural extension of the sigmoid function and is applied in the output layer of the evaluated architectures. This function transforms raw logits into probabilities across multiple classes, ensuring that the sum of the predicted probabilities for all classes equals 1. Conceptually, the softmax function can be seen as an expansion of the sigmoid function across multiple dimensions, reflecting the probabilistic distribution of inputs into distinct classes. For example, in this thesis involving classification among 7 foreground and 1 background class, the softmax activation function serves as a mechanism to estimate the likelihood of each observation belonging to any of the specified categories [102].
Rectified linear units (ReLU) activation functions were introduced to adress the vanishing gradient problem commonly observed in neural networks during training, while maintaining computational efficiency [103, 104].
However, a drawback arises as inputs below zero are clipped, rendering neurons

dead. This issue prompted the development of the Leaky ReLU variant (see Fig. 2.15 (b)), introducing a small, non-zero gradient for negative inputs to enhance model robustness [105].
In this thesis, Leaky ReLU is used in the hidden layers of the nn-UNet variants (see subsection 3.4.1). For the SwinUNetR architecture (see subsection 3.4.3), the Gaussian Error Linear Unit (GELU) is applied in all layers except the last one. In contrast to Leaky ReLU, it shows a smooth, continuous non-linearity and therefore is differentiable everywhere. This provides a more stable gradient flow during training and can help contributing to faster convergence during training [106].



(a) **(b)** **(c)**

Figure 2.15 Three common activation functions that are typically employed in neural networks. The Sigmoid **(a)**, the Leaky ReLU **(b)** and the GELU **(c)** activation function.

Loss Functions

As introduced in section 2.5.1 **Regression models**, a loss function measures the discrepancy between a model's predictions and the actual data. In neural networks, these functions take on additional importance due to the complexity of the models. The loss functions are an integral part of a neural network and measure, how good the predictions of the network during the training phase are. This process can be illustrated with a very straightforward cost function, the mean squared error (MSE). Given a multilayer perceptron with one in- and output layer and only one hidden layer with N training examples, this loss function would be:

$$\begin{aligned} L\left(\vec{y}^{true}, \vec{y}^{pred}\right) &= \frac{1}{2N}\sum_j \left(y_j^{true} - y_j^{pred}\right)^2 \\ &= \frac{1}{2N}\sum_j \left(y_j^{true} - \Phi_1\left(\vec{W}^1 \cdot \vec{a}^{(1)}\right)\right)^2 \\ &= \frac{1}{2N}\sum_j (y_j^{true} - \Phi_1\left(\sum_k w_{j,k}^{(1)} \cdot \Phi_0\left(\sum_n w_{k,n}^{(0)} \cdot x_n^{(0)}\right)\right)^2 . \end{aligned} \tag{27}$$

The loss function measures the error of the predictions, and the network aims to minimize this error during training by continuously adjusting the weights using a learning rate η until convergence is achieved. This process is known as *learning with gradient descent*, where η dictates the magnitude of changes in the trainable parameters during each adjustment [107, 108]. The gradient of the trainable parameters - the weights in the given example - is computed sequentially from the last layer to the first, a process known as backpropagation and briefly explained in the next subsection (2.5.2) [109].

Backpropagation

Basically the network's operation in each step during training contains two *sweeps* through the whole network. Initially, inputs are presented in mini-batches (several samples simultaneously), to compute neuron activities in the first layer [109].
Using mini-batches offers several advantages in estimating the gradient over the whole training set, compared to the single-sample calculations. Moreover, it can accelerate the training, by enabling parallel computations across a mini-batch of m samples instead of sequentially computing those [110].
This output serves as the input to the next layer and repeats, until the output layer is reached. There, the network produces the components of the vector $\vec{y}^{pred}$, which is the net's estimation of $\vec{y}^{true}$. Following this, every output neuron will be set to its correct output, which starts the second *sweep*. This is known as the *backward phase*, where signals propagate backwards from the output to the input, hence the name *backpropagation*. During this phase, the gradients are calculated [109].
The generalized learning rule for this process is as follows:

$$\begin{array}{cccccc} \text{weight adjustment} & = -\text{learning rate} & \cdot & \text{local gradient} & \cdot & \text{input j to neuron} \\ \Delta w_{k,n}^{(m)} & = - \qquad \eta & \cdot & \delta_k & \cdot & x_n^{(m)} \quad . \end{array} \tag{28}$$

It is important to note that δ_k, depending on the specific layer, can represent a local gradient influenced by other local gradients. For $\delta_k \neq 0$, the weight $\omega_{k,n}^{(m)}$ must be adjusted after the iteration, according to the following rule:

$$w_{k,n}^{new(m)} = w_{k,n}^{old(m)} + \Delta w_{k,n}^{(m)}. \tag{29}$$

After adjusting the weights, the network processes the next mini-batch during the forward phase. Then, the backpropagation algorithm starts again. This cycle continues until a predefined condition is met.

2.5.3 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are specialized types of Artificial Neural Networks (ANNs) that include at least one convolutional layer. They are particularly effective for processing data with a grid-like topology and spatial dependencies, like the 3D MRI volumes used in this thesis. However, the principles of CNNs can also be applied to other types of data, like audio or video. This section introduces the main concepts of CNNs for image processing, which are essential for understanding the used architectures. A typical CNN comprises several convolutional, pooling, and normalization layers, often incorporating residual blocks as well. Nowadays, also deep supervision is commonly employed. All of those building blocks are explained in the following subsection.

Convolution Layers

The most important defining characteristic of a CNN is the convolution operation, which leverages three key ideas that can enhance machine learning systems: *equivariant representations*, *sparse connectivity*, and *parameter sharing*. In the context of CNNs (Convolutional Neural Networks), convolution refers to the process of multiplying a grid-structured set of inputs with a similar grid-structured set of weights, known as kernels or filters. Unlike traditional neural networks, where layers are fully connected, CNNs generally feature *sparse connectivity* (see Fig. 2.16 **(a)**).

This is achieved by the use of a grid-structured weights matrix, or kernel, which is significantly smaller than the entire input volume. Small but meaningful features such as edges and textures can be detected within a small 3D volume of, about 10 × 10 × 10 voxels, even if the entire input volume has a resolution of 512 × 512 × 512 voxels or more. This method reduces the number of operations required to compute the output, thereby significantly lowering the computational cost compared to traditional *fully connected* neural networks (see Fig. 2.16 **(b)**). Additionally, this approach allows 3D CNNs to effectively capture and utilize local patterns within three-dimensional data.

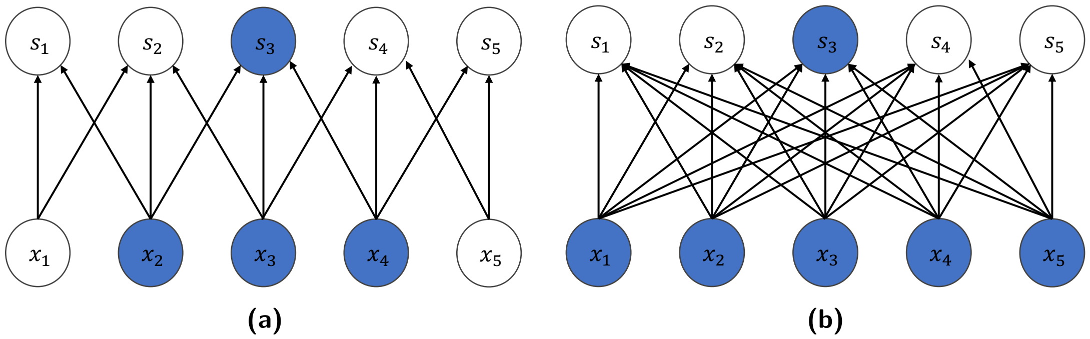


Figure 2.16 Impact of sparse connectivity versus fully connected layers, as seen from a top view. In both subfigures, the output unit s_3 and the influencing input units x_n are highlighted in blue. Those x_n's are referred to as the receptive field of s_3. In **(a)** $\vec{s}$ is formed through convolution with a 3-unit-wide kernel, resulting in a receptive field for s_3 that includes only three input units. Contrary to that, **(b)** shows $\vec{s}$ as fully connected to all of the input units, where all of them affect s_3.

For a layer with m in- and n outputs, the matrix multiplication requires $m \times n$ parameters, resulting in a runtime $\mathcal{O}(m \times n)$. In a sparsely connected approach, if each output has only k connections and $k << m$, the runtime is significantly faster, especially with an increasing amount of layers.

In traditional neural networks, each weight matrix element is used once to compute the layer's output (see Fig. 2.17 **(a)**). In CNNs, *weight sharing* means each distinct kernel element is used multiple times for every input position (see Fig. 2.17 **(b)**), except for edge neurons like x_1 and x_5 which lack adjacent partners in the s-layer.

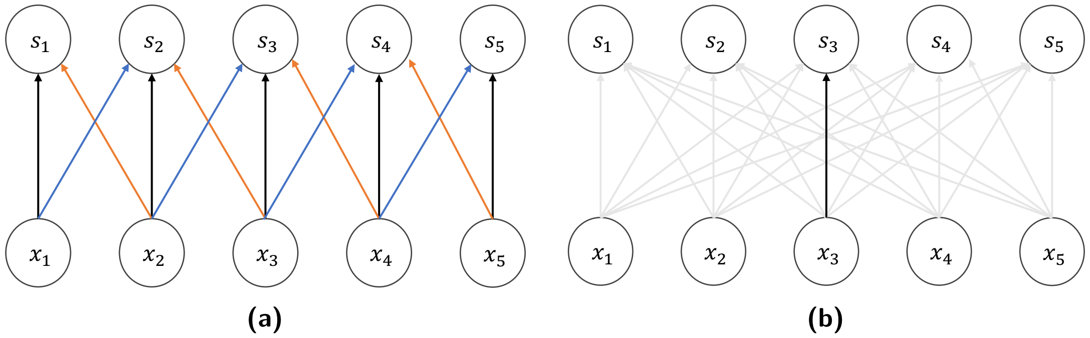


Figure 2.17 This figure demonstrates the impact of parameter sharing in convolutional layers versus fully connected layers. In both subfigures, specific parameters that are equal are highlighted in the same color. In **(a)**, due to parameter sharing, the 3 elements of the kernel are utilized across all input positions. Conversely, in **(b)**, the fully connected model lacks parameter sharing, resulting in each element of the weight matrix being used only once.

Image data exhibits *translation invariance*, where objects are recognized regardless of their position within the volume. 3D CNNs achieve this by using small 3D filters that move through the volume with a defined *stride*, generating similar feature values from local regions with similar patterns. The *stride* determines how far the filter moves in each direction, allowing the 3D CNN to learn features from adjacent volumetric windows. This particular form of parameter sharing makes the layer *equivariant* to translations.

Each layer in a 3D CNN is a 4D grid structure with *height* h, *width* w, *depth* d, and the number of *channels* c. The *height*, *width* and *depth* refer to the spatial dimensions of the input volume (the number of frames in the volumetric image). In this context, the number of *channels* refers to the input channels, the amount of modalities in the input data.

For an input of size $h1 \times w1 \times d1 \times c1$, each 3D kernel ($h2 \times w2 \times d2 \times c1$) in the kernel tensor ($h2 \times w2 \times d2 \times c2$) moves through the volume, computing the dot product between the kernel weights and the *underlying* voxels at each position. This process produces $c2$ output feature maps, forming the output tensor with dimensions $h3 \times w3 \times d3 \times c2$ (see Fig. 2.18).

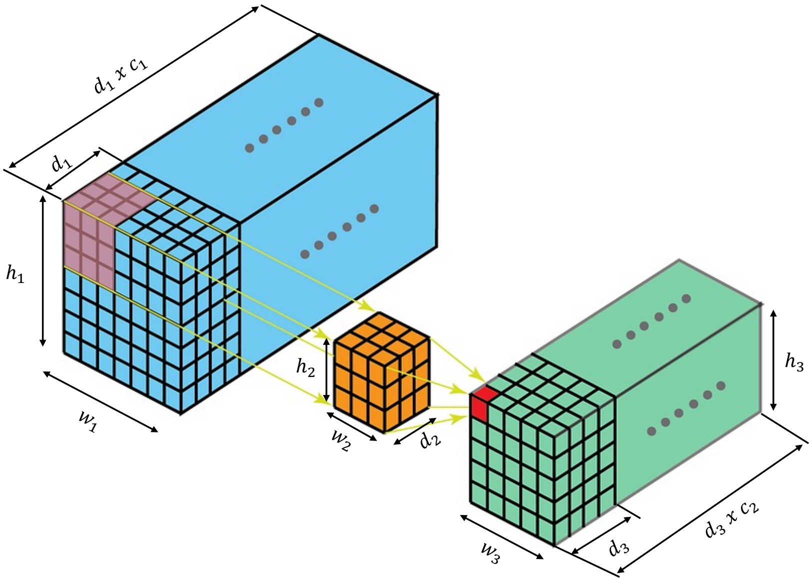


Figure 2.18 Schematic representation of a 3d cconvolution. For clarity, the process is demonstrated for only one of the color channels. The input matrix is shown in blue with the dimensions $h_1 \times w_1 \times d_1 \times c_1$. The kernel (filter) matrices are depicted in orange, with $h_2 \times w_2 \times d_2$ representing the size of the given kernel tensor. Althoug c_2 would be the number of kernels in the convolutional layer, it is not shown for better visibility. The output matrix , shown in green, has the dimensions $h_3 \times w_3 \times d_3 \times c_2$. Adapted and extended from [111].

Those convolutions lead to a size reduction of the (n+1)-th layer, compared with the n-th layer. However, this results in a loss of information at the edges of the input volume, which is generally undesirable. This issue can be resolved by using *padding*, which involves adding zero-value voxels around the edges of the feature map to preserve its spatial dimensions. The number of layers, rows, and columns of voxels needed for padding depends on the kernel size and the stride. Since these added voxels are zeros, they do not contribute to the convolution operation but allow the kernel to extend beyond the border of the layer. It is crucial to note that the kernel size and stride directly affect the receptive field, which indicates the number of output neurons in the previous layer that a specific input in a hidden layer is connected to (see Fig. 2.16 **(a)**). Consequently, each feature in the subsequent layer captures a larger spatial region in the input volume, enabling the recognition of more complex features. For instance, the first layer may learn edges, the second layer shapes, and the third layer more complex patterns.
For the architectures used in this thesis, the leaky ReLU or GELU activation functions are used after the conolutional layers. With this, matrices are generated, which are generally referred to as *activation maps*. Subsequently, those are used as the input for the next layer. Often, those are pooling layers and will be introduced in the following.

Pooling Layers

In 3D networks, pooling layers operate on small cubic regions of size $h_{pool} \times w_{pool} \times d_{pool}$ within each layer. These layers commonly employ max-pooling, which extracts maximum values from each cubic region across the activation volumes. Unlike convolutional layers, pooling layers do not change the number of feature maps. The pooling operation increases the receptive field while reducing the spatial footprint due to larger strides. Max-pooling significantly enhances translation invariance compared to strided convolutions, which reduces computational costs notably in deeper layers.

Residual Blocks

Deeper networks generally provide better performance, but there is a limit to the depth of traditional CNN models, beyond which performance degrades with additional layers. This issue can be mitigated by introducing residual blocks (see Figure 2.19).

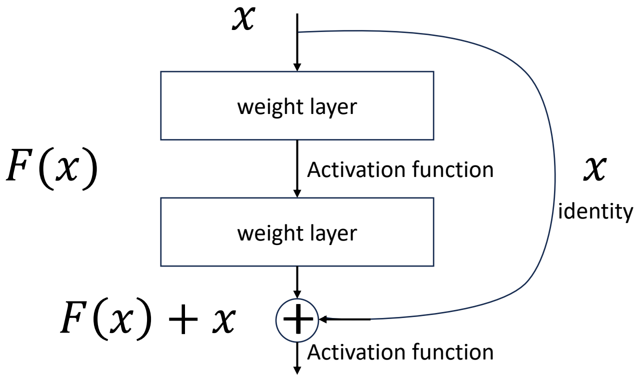


Figure 2.19 This figure shows a schematic representation of a *residual block*. x denotes the input to the residual block and the *residual* $F(x)$ is the output of the second weight layer. Both are combined and then presented as an input to the next ReLU function.

The basic principle of such a block is that it creates shortcut connections, enabling the network to skip one or more layers and allow it to learn residual functions relative to the layer inputs.

Deep Supervision

Deep supervision is a technique used in neural network training to improve learning efficiency and model performace, especially in architectures with convolutional and residual blocks. This involves incorporating additional, smaller neural networks - also referred to as *auxiliary classifiers* - at intermediate network layers. Those generate gradient signals at various depths to ensure that meaningful updates will also reach the early layers. With this, the vanishing gradient problem can be alleviated. Moreover, faster convergence, enhanced generalization and improved model accuracy can be achieved by providing direct feedback throughout the network. Deep supervision can be particularly advantageous in deep architectures, where standard backpropagation struggles to effectively transmit gradient information through many layers [112, 113, 114].
This technique is incorporated into the architectures of the nnU-Net framework that are used for liver segmentation in this thesis [115].

2.5.4 Transformer Networks

Transformer Networks, introduced in 2017 by Vaswani et al. [116], have revolutionized various fields, particularly natural language processing (NLP) and computer vision tasks [117, 118, 119, 120]. This is evident from the success of prominent models like ChatGPT, relying on GPT (Generative Pre-trained Transformer) [119, 121] and Google's BERT [120]. These models have become well-known tools, even among people not deeply involved in machine learning.

Self-Attention Mechanisms

The main innovation of Transformers is the so-called self-attention mechanism, allowing the model to dynamically focus on relevant parts of the input while processing it. This enables the model to capture long-range dependencies and contextual information.
The self-attention mechanism operates on three main components: *Queries* (Q), *Keys* (K) and *Values* (V). Those are derived from the input sequence by learnable weight matrices that resemble linear transformations. With an input matrix X, this derives to:

$$\begin{aligned} Q &= X \cdot W^Q \\ K &= X \cdot W^K \\ V &= X \cdot W^V, \end{aligned} \tag{30}$$

where W^Q, W^K, W^V are the learned weight matrices. With this, attention scores are calculated by taking the dot product of each Query vector with all Key vectors. After scaling them, a softmax activation function is applied to get *attention weights*. Subsequently those are multiplied with the corresponding Value vectors, which can be mathematically expressed as:

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_K}}\right)V. \tag{31}$$

The dimension of the Key vectors, d_K, is used for scaling to improve numerical stability. This approach allows each position in the sequence to attend at all positions, which enables the model to capture global dependencies regardless of their distance in the given sequence.
This can be expanded to the typically employed *multi-head self-attention*, which performs additional self-attention operations multiple times in parallel, but with different learnable functions for every so-called *head*. For n *heads*, this can be expressed as:

$$MultiHead(Q, K, V) = Concat(head_1, ..., head_n)W^0, \tag{32}$$

with

$$head_i = Attention\left(QW_i^Q, KW_i^K, VW_i^V\right) = softmax\left(\frac{(QW_i^Q)(KW_i^K)^T}{\sqrt{d_K}}\right)\left(VW_i^V\right). \tag{33}$$

The key-differences to only the original self-attention mechanism is that every head has its own set of additional learned weight matrices W_i^Q, W_i^K and W_i^V to transform their inputs Q, K and V, which are the derived values obtained from equations (30).
In other words, this becomes a two-step transformation, where in the first step the input sequence X → Q, K, V and then Q, K, V → Q_i, K_i, V_i, the head's specific representations for *Query*, *Key* and *Value*. This allows each head to focus on different aspects of the same input and enhances the model's ability to capture various aspects of the input data simultaneously. This multi-head approach has proven particularly effective in vision transformers and hybrid architectures, enabling the model to process visual information at multiple scales and perspectives, concurrently [116, 118, 122, 123].

Vision Transformers

For computer vision tasks, a powerful alternative to Convolutional Neural Networks has recently emerged, the so-called Vision Transformers (ViTs). They offer several key advantages and were introduced by the prominent Google paper "An image is worth 16x16 words" [122].
ViTs adapt the transformer architecture that has been originally designed for natural language processing tasks to image analysis. The main innovation is their approach to image processing, where an image is divided into a sequence of patches. Those are then linearly embedded and processed using self-attention mechanisms. With this approach, ViTs are able to model long-range dependencies and contextual information more effectively than CNNs, which are limited by their local receptive fields. Recent studies have shown that Vision Transformers (ViTs) are able to outperform Convolutional Neural Networks (CNNs) in certain tasks, particularly when pre-trained on large datasets. ViTs have demonstrated improved scalability and, in some cases, better computational efficiency compared to CNNs [122, 123, 124].

Hybrid Architectures

Because of this success, hybrid architectures like the Swin UNETR were developed, to have the best of both worlds. Those combine the strengths of CNNs and transformers to achieve state-of-the-art performance in complex tasks like 3D medical image segmentation, even with smaller datasets. Those hybrid architectures leverage the global context modeling of transformers while retaining the local feature extraction capabilities of CNNs [125, 118].

For the Swin UNETR specifically, the transformer component employs shifted windows and patch merging to efficiently model both local and global contextual information in volumetric medical images [126]. With this, the architecture is able to capture long-range dependencies while still maintaining computational efficiency. It was able to demonstrate state-of-the-art performance on challenging medical segmentation benchmarks like the Medical Segmentation Decathlon (MSD) [127] and the Beyond the Cranial Vault (BTCV) [128] datasets [118].

Swin UNETR is specifically suited for self-supervised pre-training on large unlabeled datasets which allows it to learn robust feature representations that transfer well to downstream segmentation tasks with limited labeled data. However, the publicly available self-supervised pre-trained weights for Swin UNETR are based on CT images, which presents a challenge for MRI-based tasks due to the significant domain differences between CT and MRI imaging modalities [125]. Despite facing these limitations, experiments were conducted to assess the transfer-learning performance of Swin UNETR on MRI data using the CT-based self-supervised pre-trained weights. However, these tests yielded poorer results compared to the architecture trained from scratch on MRI data. Subsequently, the latter is used for segmentation performance comparison in this work. Given these findings and the focus on MRI-based segmentation for the tasks in this thesis, the self-supervised pre-training aspect of Swin UNETR will not be elaborated at this point.

Chapter 3

Data and Methods

3.1 Datasets

This section provides a detailed description of the datasets used for training and evaluation of the architectures for segmentation as well as the datasets used for liver function estimation.

All MRI images used in this work were acquired on a Siemens MAGNETOM Skyra 3T scanner in the years 2016 to end of 2023. The datasets include a total of 458 adult subjects with liver diseases, who underwent Gd-EOB-DTPA-enhanced T1-weighted volumetric interpolated breath-hold examination (VIBE) MRI sequences with fat suppression at 3T during the native, arterial (AP), late arterial (LAP), portal venous (PVP), and hepatobiliary late phases (HBP20) were included. Additional criteria included current liver function estimation (± 3 days relative to the MRI acquisition), no allergic reactions to the liver-specific MRI contrast agent Gd-EOB-DTPA (Primovist®, Bayer Healthcare, Berlin, Germany) and no general contraindications to MRI.

For the T1-weighted VIBE sequences with fat suppression, the following parameters were used: a repetition time of 3.09 ms, echo times of 1.17 ms and 2.49 ms, a flip angle of 10°, a parallel imaging factor of 2, and 64 slices. The measured voxel size was $1.71 \times 1.25 \times 4.5$ mm^3. Those were reconstructed to a voxel size of $1.25 \times 1.25 \times 3.0$ mm^3 and dimensions of 320 x 320 x 64. All images were obtained during a 14-second breath-hold, both before Gd-EOB-DTPA administration (native phase) and during the dynamic phases. Patients received an intravenous bolus injection of Gd-EOB-DTPA (0.025 mmol/kg body weight) at a flow rate of 1 mL/sec, followed by 20 mL of 0.9% sodium chloride flush for contrast-enhanced MRI.

Initially, T2-weighted images were considered for inclusion in the dataset to improved lesion classification and enhanced segmentation of structures like ascites, due to better water visibility in these modalities. However, the T2-weighted images had a significantly different voxel size for the out-of-plane orientation (6.6 mm) compared to the 3 mm for the T1-VIBE sequences and different FOVs (fields of view). This discrepancy made sufficient coregistration to the T1-VIBE images unfeasible in the performed tests. Consequently, only the T1 sequences necessary for feature extraction in later stages were used.

3.1.1 Dataset for Liver Segmentation

This dataset includes 78 adult subjects, that underwent the ^{13}C-Methacetine breath test about 24 hours before MRI acquisition in 2016.

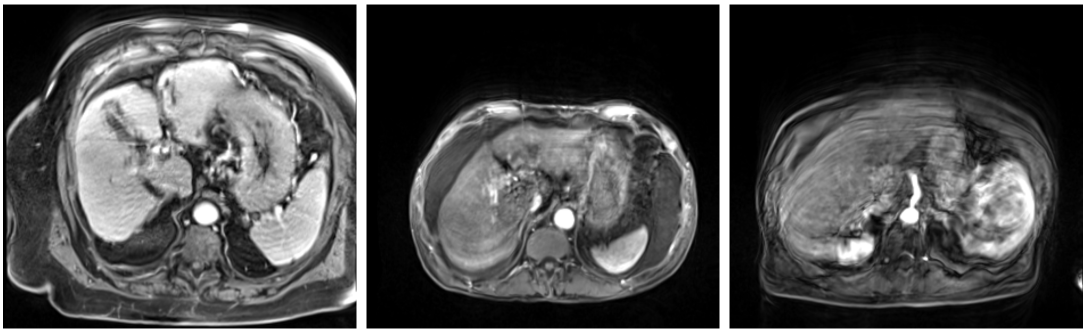
For 19 of the patients, at least one of the five acquired imaging phases was of very poor quality with lots of artifacts that often occur when the patients are not able to remain still or can't hold their breath for 14 seconds without moving. These samples were excluded from the test sets. However, those were still available to the training, making it harder for the network to learn features and likely rendering it more robust during inference. Three of those bad images are illustrated in figure 3.1 and a comparison with the good samples shown in figure 2.8 clearly highlights the problems with those images. Here needs to be emphasized that not all phases from those patients were blurry, hence of course it was possible to manually label those MRI volumes on other phases.

This lead to a total of 59 patients in the test sets, on which the networks were evaluated.

The dataset was split into four different combinations of data available to the training and hold-out data that is only available for testing. Three of those combinations have 14 images in the test set, whereas the fourth combination has 17 subjects for testing.

An experienced radiologist from Universitätsklinikum Regensburg labeled the liver, hepatic veins, portal vein, lesions, abdominal aorta, thoracic aorta, and ascites in the preprocessed MRI images of 78 patients with the software ImFusion Labels v.0.21.5 [129].

Initially, the lesions were classified into categories such as cancerous (*malign*) tumors (hepatocellular carcinoma (HCC) and cholangiocellular carcinoma (CCC)), non-cancerous (*benign*) tumors (focal nodular hyperplasia (FNH), adenoma, hemangioma), cysts, ablation defects, metastases, bilioma, and regenerative nodules. In total, the cohort comprises 21 HCCs, 9 CCCs, 9 FNHs, 5 adenoma, 7 hemangioma, 8 cysts and 10 metastases. However, in most of the given cases, the lesion areas are very small and more often than not showed atypical contrast medium



(a) (b) (c)

Figure 3.1 This figure demonstrates three images with poor quality, due to breathing artifacts, that were excluded from the test dataset. The field of view for all three images is exactly the same. The differences in appearance are there because of drastically different body mass indices. **(a)** depicts the late arterial phase of one patient, **(b)** shows the arterial phase for another subject and **(c)** was acquired during the portalvenous phase.

behavior. Additionally, there was no possibility to get more manually labelled data than those 78 patients, because the radiologist's time for this task was very limited, since it is extremely time-consuming and the daily clinical routine is more important.

Consequently, achieving good classification performance for these lesions was not feasible. The segmentations were acceptable, but classification just didn't work. Therefore, all defective areas in the liver were grouped together into one *lesions* class, resulting in seven remaining classes of interest in the cohort.

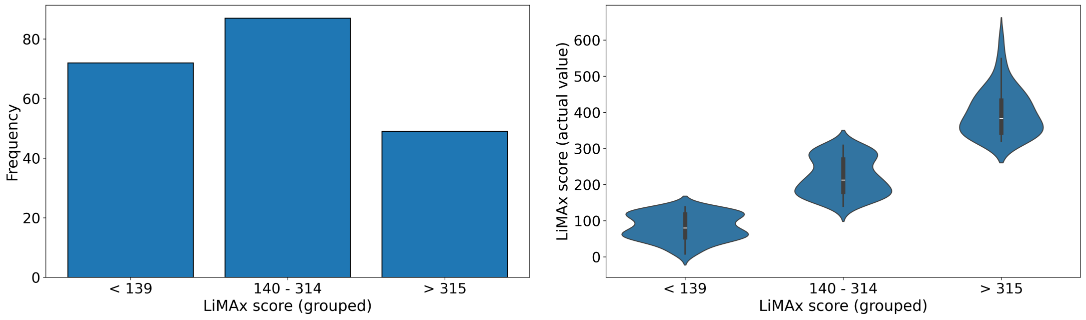
Only 7 patients showed signs of ascites in their MRIs, which is anticipated, as this pathology typically indicates an advanced, very late stage of liver disease, especially in patients with cirrhosis [130].

3.1.2 Datasets for Liver Function Estimation

For every patient in the aforementioned dataset, the LiMAx values are available, and initially, this was planned to be the only score used as ground-truth for liver function estimation in this thesis. However, the LiMAx test is not routinely employed in the daily clinical routine in Regensburg. Additionally, for some patients, the LiMAx values acquired a few days apart showed drastic variance. Therefore, the inclusion of other typically employed scoring systems, such as the MELD score, seemed to significantly improve the quality and robustness of these experiments and led to a much bigger cohort.

Even though, 6871 T1-VIBE MRI scans were acquired at the University Hospital in Regensburg from 2016 to the end of 2023, after deep investigation of the MRIs and laboratory values, only 458 met the previously mentioned inclusion criteria for this experiment.

Subsequently, the datasets used for liver function estimation comprise a total of 458 patients, which are subdivided into multiple cohorts corresponding to three different liver function scores that were used. For some of the patients, there are multiple liver function scores available and therefore they appear in more than one cohort. Specifically, ALBI-scores were measured for 207 subjects, MELD-scores are available for 409 of them and LiMAx values were measured for 208 patients. The dataset for training and evaluation of the neural networks for segmentation (see previous subsection 3.1.1) is completely included in the LiMAx cohort.



(a) (b)

Figure 3.2 Distribution of the LiMAx-scores for the corresponding cohort in this thesis. Because of their wide range, they were grouped to the three cutoff ranges that are used for classification of healthy, intermediate and significant impaired liver function. **(a)** shows the general distribution per group, whereas **(b)** provides a more in-depth view of how the subjects are distributed within the corresponding group.

The grouped distribution of the aforementioned cohort is illustrated in Figure 3.2. For the first group (score lower than 139), second group (LiMAx between 140-314), and third group (values above 315), there are 72, 87, and 49 subjects available in the cohort, respectively. This distribution of the data is relatively balanced, except for the third group, which is slightly underrepresented.

The subjects of the MELD cohort showed MELD-scores from 6 to 31 with an overrepresentation of values between 6 and 9 and only single occurences of scores greater than 20 (see Fig. 3.3). This is slightly contrary to the distribution of the LiMAx cohort, but similar distributions for the MELD-score can be observed in other studies [131, 132] and can be explained by an increasing three-months mortality rate for patients with higher MELD-scores. Because of this, the patients with a lower score are more stable and can wait longer for a transplantation. Some

of them receive a transplant, with an increase in MELD and some of them die before, unfortunately [133, 134].
Those findings highlight the importance of correct liver function evaluation, especially in the early stages of chronic liver diseases.

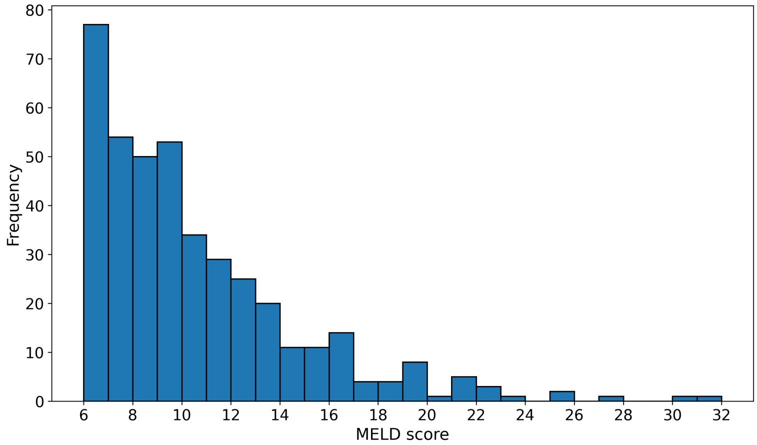
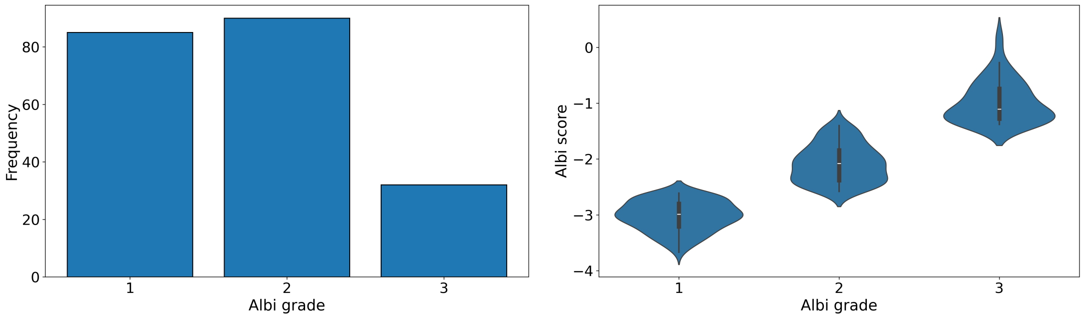


Figure 3.3 Distribution of the MELD-scores for the corresponding cohort in this thesis.

The last cohort (ALBI-score) comprises 85, 90 and 32 patients with albi grades 1, 2 and 3, respectively (see Fig. 3.4). This distribution mirrors that of the MELD-scores, with fewer patients exhibiting severely impaired liver function, a pattern attributable to similar factors. Notably, within the ALBI groups 1 and 3, scores cluster towards the boundaries of grade 2, resulting in a limited number of outliers in the cohort and many patients that are harder to classify accurately.



(a) (b)

Figure 3.4 Distribution of the Albi grades for the corresponding cohort in this thesis. Albi grades 1 and 2 are equally represented, while there are less patients with grade 3. **(a)** shows the general distribution per group, whereas **(b)** provides a more in-depth view of how the subjects are distributed within the corresponding group.

For patients with multiple available scores, pairwise comparisons were conducted to assess the correlation and comparibility between different scoring systems. These results are presented in the following.

Correlation of Different Liver Function Scores

The correlation between MELD- and LiMAx-scores is depicted in Figure 3.5. The analysis reveals a Pearson correlation coefficient of -0.42 and a Spearman's correlation of -0.49, indicating a moderate negative monotonic relationship. This suggests that while the linear relationship is weak, there is a stronger monotonic trend, as captured by the Spearman correlation. Subsequently, as the MELD-score increases, indicating worsening liver function, the LiMAx-score tends to decrease, reflecting reduced liver metabolic capacity. Tis moderate correlation may not be representative for other cohorts, as there are only a small amount of patients with a MELD-score greater than 17 in this one. However, the values are in aggreement with what can be found in other studies [135, 136].

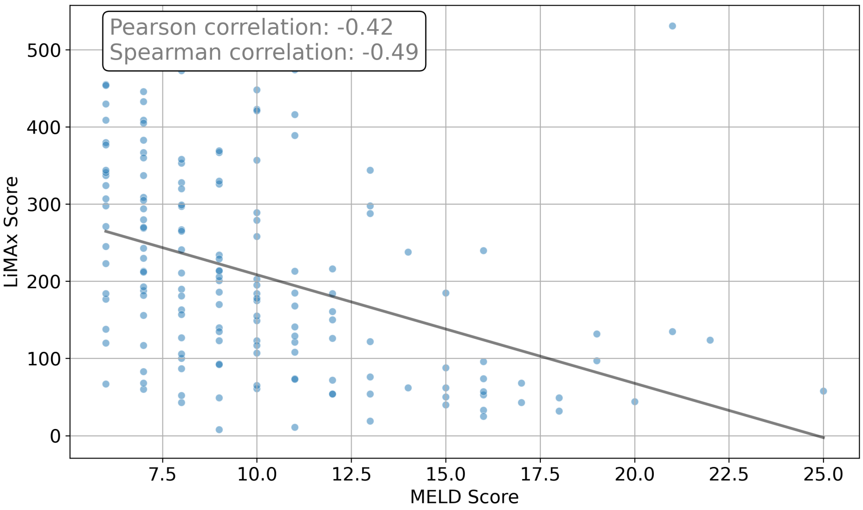


Figure 3.5 Cross correlation of MELD- and LiMAx-score for the given cohort. Moderate Pearson and Spearman correlations of -0.42 and -0.49 can be observed.

In figure 3.6, the correlation between ALBI- and LiMAx-scores is illustrated. Those have Pearson and Spearman correlation coefficients of -0.65 and -0.69, respectively. The stronger negative correlations between ALBI- and LiMAx-scores indicate a more pronounced inverse relationship than for MELD and LiMAx. Again, the higher Spearman correlation indicates a non-linear, but monotonic relationship between both scores. The observed coefficients are higher than found in other studies [136]. One reason for this can be the small amount of patients with highly impaired liver function in the cohort

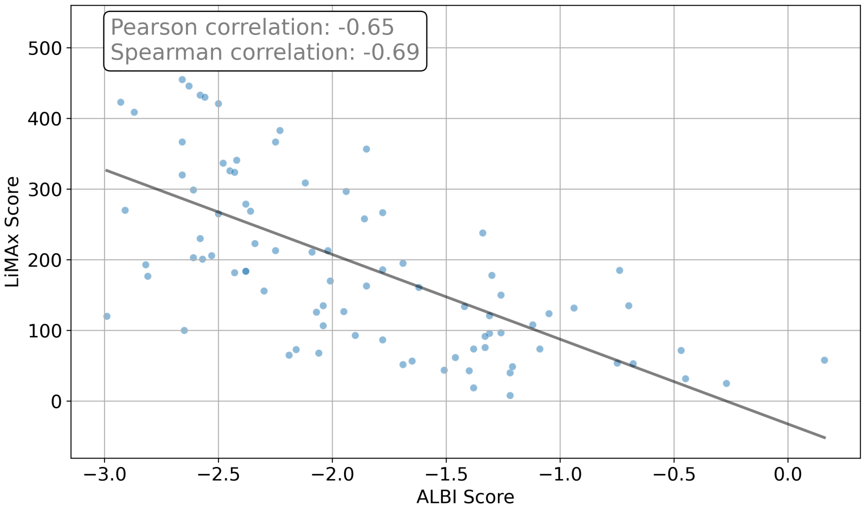


Figure 3.6 Cross correlation of ALBI- and LiMAx-score for the given cohort. Pearson and Spearman correlations of -0.65 and -0.69 can be observed.

Figure 3.7 presents the cross-correlation between the ALBI- and MELD-scores, with a Pearson correlation coefficient of 0.67. For Spearman's correlation, the coefficient is 0.69. Therefore, those scores share the strongest correlation among this comparison, which was to expect since they both share Bilirubin as one main factor for calculation.

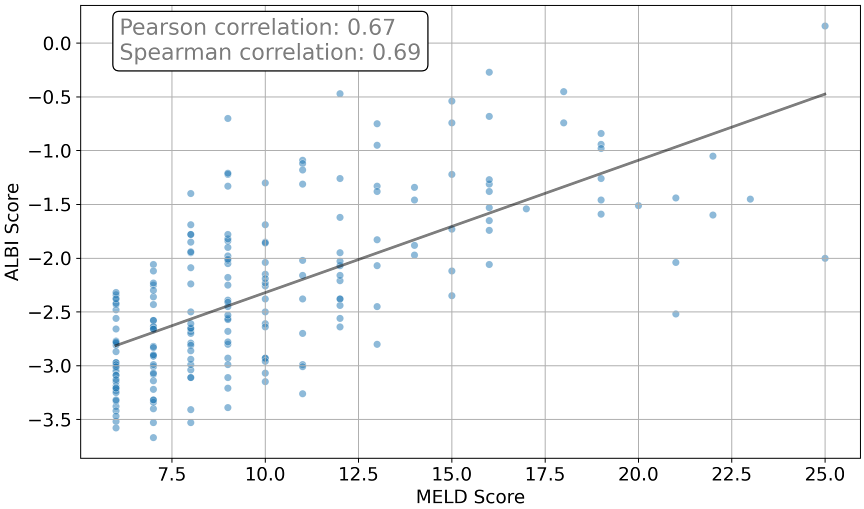


Figure 3.7 Cross correlation of ALBI- and LiMAx-score for the given cohort. Pearson and Spearman correlations of -0.67 and -0.69 can be observed.

Both scores are used to evaluate liver disease severity, but they are derived from different parameters. The positive correlation suggests that higher ALBI-scores are associated with higher MELD-scores. There is a higher linear correlation between ALBI and MELD for patients with a MELD-score smaller than 20, leading to a Pearson coefficient of 0.70, whereas for subjects with a higher MELD-score the correlation drops to only 0.52.
For Spearman's correlation, the coefficients are 0.68 and 0.44, respectively. This indicates non-linear correlation, again. However, patients with a higher MELD-score are underrepresented in this subset, where both scores are available. Therefore, those values may not be representative for other cohorts.

3.2 Preprocessing of MRI Images

All images for the segmentation task have been pre-processed with the following pipeline in a Python v.3.10 environment:

1. Conversion from DICOM to Niftii file format using dcm2niix [137],
2. Bias correction using the N4ITK algorithm [138],
3. Image-wise z-score normalization with numpy [139],
4. Coregistration with nipype interface and ANTs registration using three transformations in the order of *Rigid, Affine* and finally *SyN* [140, 141],
5. Reorientation of the MRI images to the standard radiological orientation with FSL [142],
6. Labeling and reslicing the images to an isotropic voxel spacing of 1.2 mm^3 with ImFusion Labels [129].

The reslicing results in spatial dimensions of (160, 333, 333) for the MRI volumes used in this thesis.
The images for liver function estimation were pre-processed using only the first, second and fourth step of the pre-described pipeline. The decision against intensity normalization is justified in this case, since the majority of the MRI-derived features for the liver function estimation are relative intensity-enhancement indices. Therefore, distorting the correlations of the relative intensities in the liver parenchyma over the five phases would render these measurements unreliable. However, for generation of the segmentation maps of the anatomical structures, all preprocessing steps were applied to the images prior to segmentation inference.

The segmented labels were then applied to the images that underwent the three aforementioned pre-processing steps.

3.3 Metrics for Evaluation

This section outlines the evaluation metrics employed for the various methods used in this thesis, including coregistration, segmentation, and liver function estimation tasks. Accurate evaluation is crucial for determining the effectiveness of these techniques for medical imaging applications.
For coregistration, metrics that assess the alignment quality between images that provide insights into both global and local alignment accuracy are utilized.
For the segmentation tasks, a wide range of metrics is employed to quantify the performance of the tested models and evaluate the accuracy of our segmentation algorithms in delineating specific regions or structures in medical images.
For the regression and classification tasks, suitable metrics are employed to gain insight into the predictive performance of the machine learning methods that are applied for liver function estimation.
With this comprehensive set of metrics, a thorough evaluation of the whole proposed pipeline for liver function estimation can be ensured.
All of them will be explained in the following.

3.3.1 Coregistration

The coregistration process was performed five times for all 458 subjects and all phases. Each time a different phase was picked as the fixed image to which all others were coregistered. The results were evaluated with a combination of visual inspection and the calculation of multiple metrics to find the best suiting approach.
Several metrics are commonly used to assess the quality of image coregistration, where the most prominent ones include mean squared error, mutual information, structural similarity index, and normalized cross-correlation [143, 144, 145].
Each of these metrics provides different insights into the alignment of images, which will be explained in the following.

Mean Squared Error

The Mean Squared Error (MSE) is a simple metric for basic assessment of image coregistration quality. It is a measure for the average squared differences between the intensity values of corresponding voxels in two images. In this context, the mathematic equation is:

$$\mathrm{MSE} = \frac{1}{LNM} \sum_{i=1}^{L} \sum_{j=1}^{N} \sum_{k=1}^{M} (I_2(i,j,k) - I_1(i,j,k))^2, \tag{34}$$

where $I_1(i,j)$ and $I_2(i,j)$ refer to the corresponding intensity values of the n-th voxel in the two images and $L \cdot N \cdot M$ is the total number of voxels. MSE provides a simple and clear measure of the overall difference between the images. However, it has some limitations. It is very sensitive to large differences in intensity values. On top of that, MSE doesn't account for any structural or perceptual differences in the images, thus it is less suitable for multimodal image registration, where intensity values may vary significantly because of different imaging modalities [146, 147, 148].

Despite those limitations, the metric was still used in combination with the other metrics for a better and more comprehensive evaluation of the coregistration performance.

Mutual Information

In the context of image registration, the mutual information (MI) is a measure for how much information one image provides about another. For image coregistration, it quantifies the statistical dependence between the intensity values of corresponding voxels in two images. MI is particularly useful for multimodal image registration, because it doesn't assume any specific functional relationship between image intensities, is robust to intensity variations caused by different modalities and can handle complex relationships between tissue properties and image intensities. However, there are also limitations to it, such as sensitivity to image overlap and potential local maxima in the optimization landscape. The only fixed value for MI is 0, meaning that two images don't share any information and are completely unrelated in terms of their pixel intensities or features. There is no upper limit and the values can't be compared to other published datasets, because the MI highly relies on the way it is calculated (the size of bins used for calculations) [149, 150, 151]. However, for comparing different coregistration approaches on the same dataset with the same calculations of MI, the values can be compared to each other and provide valuable insight.

Structural Similarity Index

The Structural Similarity Index (SSIM) is a suitable metric for the evaluation of multimodal coregistration, because it doesn't rely on intensity based measurements.

Unlike those metrics, SSIM is able to assess the structural similarity between two images or volumes. Therefore, it is well suited to compare coregistered MRI volumes from various modalities with inherently different intensity distributions to each other. With this metric, local patterns of pixel intensities normalized for luminance and contrast, are evaluated. This allows it to capture perceptual and structural similarities even when absolute intensity values differ between modalities. Hence, SSIM is particularly effective for detecting subtle misalignments that can't be quantified by intensity differences alone. Furthermore, it is robust to noise and small variations in intensity. The combination of all those aspects shows that SSIM provides a meaningful measure of alignment quality across modalities, which makes it a valuable tool for quantitative assessment of multimodal MRI coregistration success. Downsides of this metric include a high sensitivity to structural changes as stretching, rotations or other distortions and overestimation of the quality near hard edges in medical images. The metric can have values between 0 and 1, where 1 indicates perfect similarity [152].

Normalized Cross Correlation

Normalized cross correlation (NCC) is a measure for the degree of linear correlation between intensity values of corresponding voxels in two images. It is normalized to account for differences in contrast and image brightness, which makes it suitable for comparing multi-modal image coregistration, where absolute intensity values may vary significantly. NCC is robust to linear intensity transformations, therefore it is effective for detecting global and local misalignments across modalities. It can identify structural similarities, even when the specific intensity mappings are different for each modality. Additionally, NCC is resilient to common artifacts in MRI, such as inhomogeneities in the B-field. Subsequently, it is a reliable and versatile metric for quantitative assessment of the quality of multimodal MRI coregistration, especially in scenarios with relatively consistent intensity relationships between different modalities [153, 154].

3.3.2 Segmentation

There is a wide range of metrics available to quantify the performance of a segmentation model. The most prominent is the Dice-Sørensen-Coefficient (DSC) [155, 156]. However, multiple metrics are required for a comprehensive evaluation of the segmentation performace of an algorithm. Following, there's an expanded explanation of the key metrics used to quantify the segmentation success in this thesis. There, TP, FP, TN, FN refer to true positive, false positive, true negative and false positive, respectively.

- **Positive Prediction Value (PPV):**

$$PPV = \frac{TP}{TP + FP} \tag{35}$$

PPV, also called Precision, quantifies the proportion of correctly identified positive voxels among all voxels classified as positive. It is particularly useful in assessing the model's ability to avoid false positives, which is crucial in medical applications where overdiagnosis can lead to unnecessary treatments.

- **True Positive Rate (TPR):**

$$TPR = \frac{TP}{TP + FN} \tag{36}$$

TPR, also known as Sensitivity or Recall, measures the proportion of actual positive voxels correctly identified by the model. A high TPR indicates that the model is effective at detecting the target structures or lesions, which is essential in scenarios where missing a pathology could have serious consequences.

- **Dice-Sørensen-Coefficient (DSC), F1-Score** [155, 156]:

$$DSC = \frac{2TP}{FN + FP + 2TP} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{37}$$

The DSC (or F1-Score) is a statistical measure of spatial overlap between the predicted segmentation and the ground truth. It ranges from 0 to 1, where 1 indicates perfect overlap. The DSC is sensitive to both false positives and false negatives, making it a balanced measure of segmentation accuracy.

- **Jaccard-Coefficient**[157]:

$$Jaccard = \frac{TP}{TP + FP + FN} \tag{38}$$

Also known as the Intersection over Union (IoU), the Jaccard coefficient is closely related to the DSC, but the true positives are not double weighed. It measures the size of the intersection divided by the size of the union of two segmentation sets. The Jaccard coefficient is always lower than or equal to the DSC for the same segmentation.

- **Lesion-wise False Positive Rate (LFPR):**

$$LFPR = \frac{LFP}{PL} \tag{39}$$

  LFPR evaluates the rate of falsely detected lesions among all predicted lesions. This metric is particularly important in clinical settings where each false positive could lead to unnecessary follow-up procedures.

- **Lesion-wise True Positive Rate (LTPR):**

$$LTPR = \frac{LTP}{RL} \tag{40}$$

  LTPR assesses the proportion of actual lesions correctly identified by the model. This metric is crucial for understanding the model's ability to detect individual lesions, which can be more clinically relevant than voxel-wise metrics in some scenarios.

- **Volume Difference (VD):**

$$VD = \frac{\left|Vol_{pred}^{tissue} - Vol_{true}^{tissue}\right|}{Vol_{true}^{tissue}} = \frac{|(TP + FP) - Vol_{true}^{tissue}|}{Vol_{true}^{tissue}} \tag{41}$$

  VD quantifies the relative difference between the predicted and true lesion volumes. This metric can be particularly important in applications where the size of the segmented region is clinically significant, such as tumor volume assessment for treatment planning or response evaluation.

Each of these metrics is able to provide valuable information about different aspects of segmentation performance. While the DSC offers a good overall measure of spatial overlap, the PPV and TPR provide insights into the balance between false positives and false negatives. The lesion-wise metrics (LFPR and LTPR) are crucial for assessing performance at the level of individual lesions, which can be more clinically relevant in some cases. Finally, the VD offers important information about volumetric accuracy, which is critical in many medical applications.
By using this comprehensive set of metrics, the segmentation performance can be evaluated thoroughly.

3.3.3 Liver Function Estimation

Liver function estimation, encompassing both regression and classification tasks, is essential for diagnosing and managing liver diseases. For the deployed regression models such as linear regression, Elastic Net, Ridge regression, and Lasso regression, the evaluation metrics include Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and R-squared (R^2). These metrics provide insights into the models' accuracy and goodness of fit by quantifying the average prediction error and the proportion of variance explained by the model. To ensure a fair comparison among scores normalized metrics were also used: Normalized Mean Absolute Error (NMAE) and Normalized Root Mean Squared Error (NRMSE). In this approach, MAE and RMSE were simply divided by the range of ground-truth values for the specific score within the given cohort. This normalization process allows for a more equitable assessment of different scales and datasets.

For the classification tasks, where Support Vector Machine (SVM), Random Forest (RF), Multilayer Perceptron (MLP), and XGBoost are deployed, performance is assessed using metrics such as Accuracy, F1-score and Area Under the Receiver Operating Characteristic Curve (AUC-ROC). These metrics thoroughly evaluate the models' ability to correctly classify liver function, balancing the trade-offs between false positives and false negatives.

By employing this comprehensive set of metrics, a thorough evaluation of the models' performance in liver function estimation can be ensured.

3.4 Architectures for Image Segmentation

The first step to derive meaningful features from the preprocessed MRI images for building an automated machine learning model for liver function assessment is to create precise segmentations of the tissues under consideration. To achieve the best possible results for this task, three publicly available models were tested against each other. The first is the standard U-Net [158] within the nnU-Net framework [115], which is well researched and has been successfully applied to a great variety of medical imaging segmentation tasks, where it consistently reached state of the art performance over the last years [159, 115, 160, 161].

The second architecture is a more recent U-Net variant in the same nnU-Net framework, but with residual blocks in the encoder path of the model [162].

With the rising attention to transformer models, it is appropriate to test such an architecture against the well researched CNNs and see, if any improvements can be observed. For this, the Swin UNETR [118] was used, as it achieved promising results in some studies and there is an implementation from MONAI [163] available that can be applied to custom tasks, with minor changes to it.

Those architectures will be explained and illustrated in the following.

3.4.1 3D nnU-Net

The standard 3D nnU-Net architecture, depicted in Figure 3.8, is configured for an input patch size of (80, 192, 160), accommodating five input channels that represent different phases of the T1-VIBE sequence. This configuration allows the model to leverage multi-phase imaging data effectively. The architecture comprises six stages, each with a feature map size progression of 32, 64, 128, 256, 320, and 320, respectively. This progression facilitates an increase in feature complexity and abstraction as the data moves deeper into the network. Each stage employs a consistent kernel size of (3, 3, 3) for the 3D convolutions, except for the final layer, which is optimized for output generation.

Dimensionality reduction within the network is achieved through convolutions with varying strides, as opposed to traditional max pooling layers. Those are configured as follows: (1, 1, 1) for the initial stage to preserve spatial dimensions, followed by (2, 2, 2) for the next four stages to effectively downsample the feature maps. The final stage uses a stride of (1, 2, 2), allowing for selective spatial resolution retention.

Each stage in the encoder and decoder paths contains two 3D convolution layers, ensuring a balance between computational efficiency and feature extraction capability. The use of 3D Instance Normalization throughout the network aids in stabilizing the training process and improving convergence rates.

The model is trained with a batch size of 2, because of the memory constraints given by the framework, due to the VRAM allocation limit of 11 GB for this network configuration.

3.4.2 3D Residual Encoder nnU-Net

The ResEnc nnU-Net is a specialized variant of the standard nnU-Net architecture shown in figure 3.8 that integrates residual connections within its encoder to enhance performance, particularly for medical image segmentation tasks [162].

The model in this thesis is configured with a batch size of 2. The patch size is set to (112,256,256). Each stage employs a kernel size of (3,3,3) across all layers and the strides for the convolution are set as follows: (1,1,1) for the first stage, which preserves the spatial dimensions, followed by (2,2,2) for the next four stages, effectively downsampling the feature maps to reduce dimensionality and increase the receptive field. The final two stages use strides of (1,2,2), allowing for selective downsampling while maintaining some spatial resolution.

Each stage contains a specific number of residual blocks: 1, 3, 4, 6, 6, 6 and 6, respectively. Every residual block contains two 3D convolution layers. In the

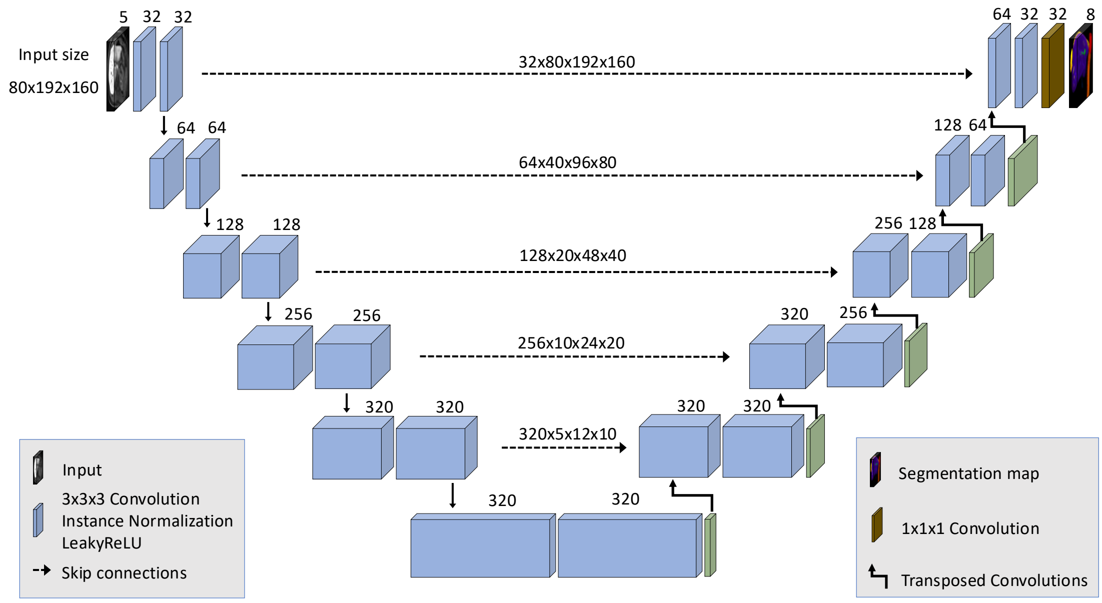


Figure 3.8 3D full resolution nnU-Net architecture used for the segmentations in this thesis.

decoder path, the network uses a single convolution per stage, which helps in reconstructing the segmentation map from the encoded features, while still maintaining efficiency. In the whole network, 3D Instance normalization is utilized.
This architecture is significantly larger and more computationally intensive, requiring substantial VRAM and longer training durations (see section 3.5). It was specifically engineered to take advantage of the advancements in GPU capabilities over recent years. This is emphasized, especially when compared to the standard nnU-Net, which is constrained by a maximum VRAM limit in its self-configuration method [115]. The ResEnc nnU-Net's design allows it to handle more extensive and complex datasets, leveraging modern hardware to achieve superior performance [162].

3.4.3 Swin UNETR

In this thesis, the Swin UNETR is implemented using the MONAI framework [163], which provides a flexible and robust platform for medical imaging applications. This implementation is intended to compare the segmentation performance of a Transformer-CNN hybrid architecture with the well-established nnU-Net CNN variants.
The model is configured with an input image size of (128, 128, 128) and accommodates five input channels. The SwinUNetR model is designed with a feature size

of 48, doubling in each stage. Figure 3.9 shows a schematic representation of this architecture.
Due to memory limitations and to enable a fair comparison with the nnU-Net variants, the model is trained with a batch size of 2. A sliding window inference method with a batch size of 4 is used for efficient prediction.

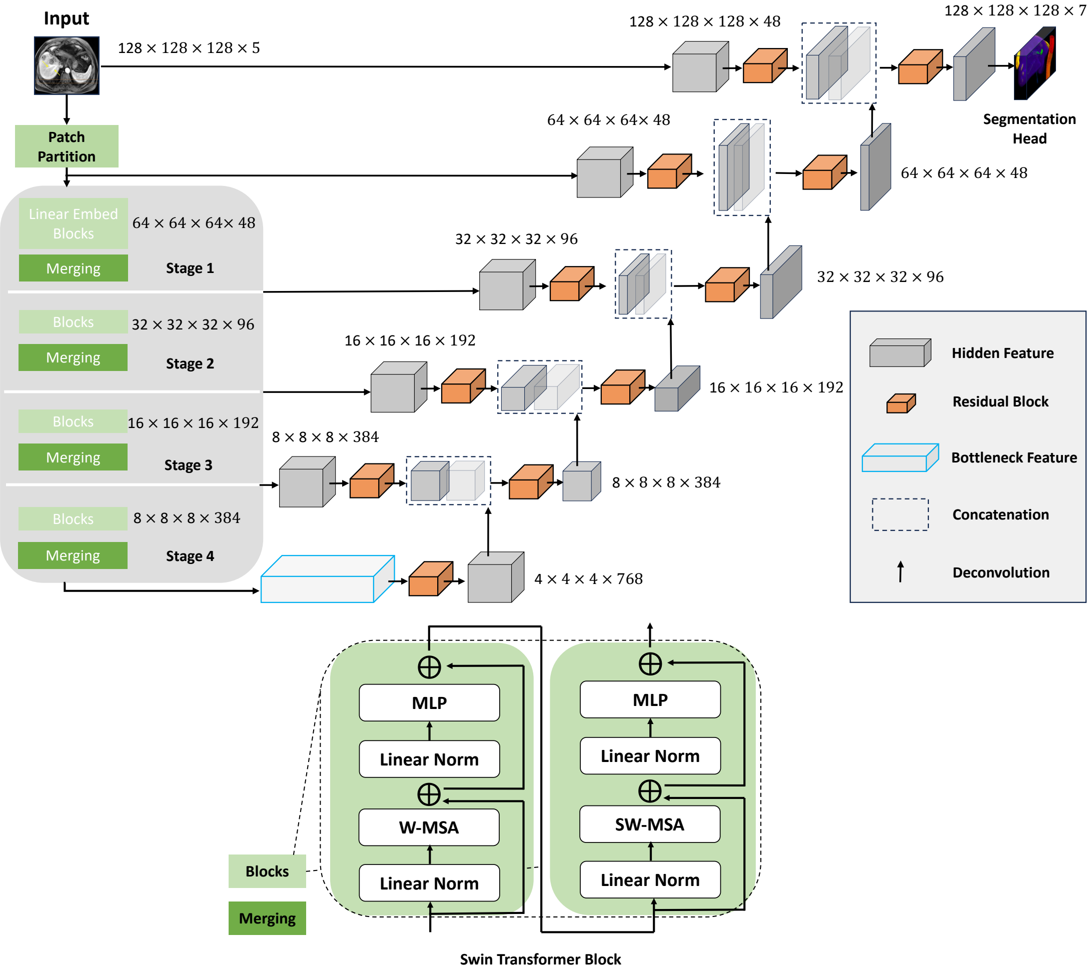


Figure 3.9 **(Top)** 3D Swin-UNetR architecture used for the segmentation performance comparison in this thesis. **(Bottom)** shows a detailed illustration for two serial Swin Transformer Blocks. W-MSA and SW-MSA refer to regular and shifted window multi-head self-attention modules, respectively. Inspired by [118].

3.5 Network Training

The final network training and inference was conducted on a NVIDIA DGX Station A100 with four NVIDIA A100 (40GB VRAM) GPUs, 512 GB RAM and an AMD

EPYC 7742 CPU with 64 cores (128 threads) in a conda environment with Python v.3.10, Pytorch v.2.2.0, Tensorflow v.2.14, MONAI v.1.3.2 and CUDA v.11.8.

All architectures underwent training on four distinct combinations of training and test datasets. For each combination, a five-fold cross-validation was conducted, ensuring consistent training and validation dataset splits across all architectures. All architectures were trained with a DiceCELoss function, which combines Dice loss and cross-entropy loss. Swin UNETR utilizes AdamW optimizer with an initial learning rate of 1e-4, whereas the nnU-Net variants use SGD as an optimizer with an initial learning rate of 0.01.

The nnU-Net framework employs a fixed training procedure of 1000 epochs. For optimal results, the network from the final epoch should be used for inference, even if an earlier stage showed better validation loss. This approach is necessary because the validation process during training only samples 50 batches of patches from the original images, regardless of the total number of possible batches. This method provides a rough estimate of segmentation accuracy on the entire validation dataset, balancing speed and performance overview. It's important to note that nnU-Net defines epochs differently from the conventional understanding. Instead of processing all images from the training dataset, an epoch in nnU-Net consists of a fixed 250 batches, regardless of dataset size or batch size [115].

The SwinUNetR implementation differed in its training and validation process. It evaluated all images from the validation set during each validation stage, selecting the network with the best validation loss for inference. Additionally, one epoch processed all images from the training dataset and the total number of epochs was set to 1300. This approach mitigated potential overfitting issues in later training stages by choosing the network with the lowest validation loss and ensured a training to run long enough to achieve the best results.

The images for the training were augmented to enhance the model's ability to generalize and improve its performance on unseen data. Data augmentation techniques were applied to artificially expand the training dataset and introduce variability. These techniques included random *geometric transformations* (rotations, flips, scaling and translations), *intensity transformations* (brightness and contrast adjustments, gamma corrections) and elastic deformations to simulate realistic tissue deformations that are especially important in abdominal imaging with the problems of breath holding techniques, as explained in section 4.1. These augmentations were applied on-the-fly during training, ensuring that each epoch presented the model with a unique set of transformed images. This approach helped to prevent overfitting and improved the model's ability to handle variations in real-world data.

For both nnU-Net and SwinUNetR, the augmentation strategies were kept as consistent as possible to ensure a fair comparison between the architectures. However, the specific implementation details may have varied slightly due to the differences in their respective frameworks.

During the validation phase and inference, no augmentations were applied to the test images, as the goal in those steps is to evaluate the model's performance on unmodified data. This approach allowed for a realistic assessment of how the trained models would perform in practical applications.

The training process for one fold took 13.5 hours for the standard nnU-Net, 39 hours for the residual encoder nnU-Net and 12 hours for the Swin-UNetR implementation.

Subsequently, for the 20 networks that were fully trained for each architecture, this results in a total training time of 11.25, 32.5 and 10 days, respectively. The history for the training and validation losses for all three architectures is displayed in figure 3.10.

Training the SwinUNetR implementation, the standard 3D U-Net from the nnU-Net framework, and the residual encoder nnU-Net requires approximately 32.5 GB, 8.5 GB, and 28 GB of VRAM, respectively.

Inference for ensembled predictions with the three previously mentioned architectures take approximately 58, 45 and 49 seconds for one subject, in that order.

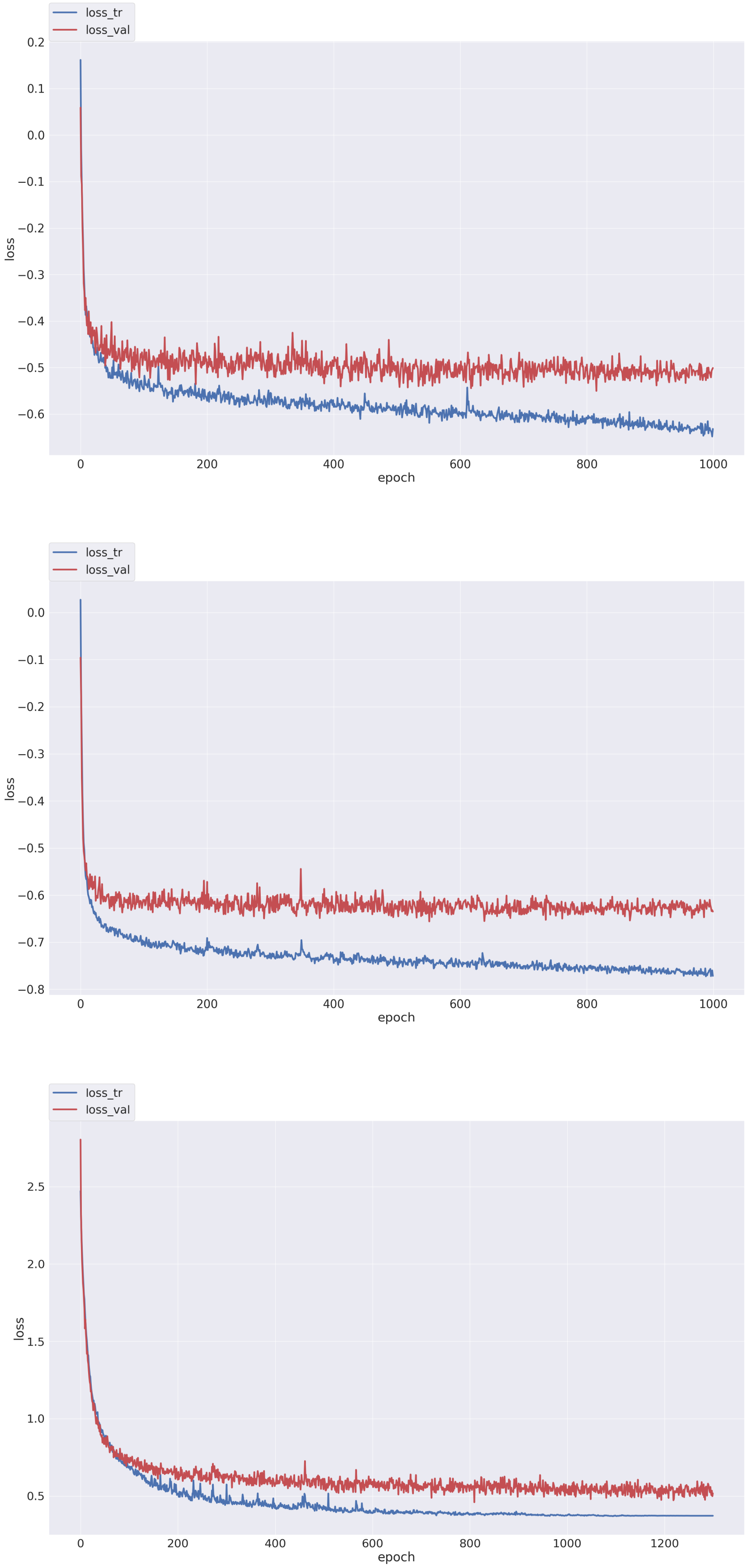


Figure 3.10 Loss history for the standard nnU-Net **(top)**, ResEnc nnU-Net **(mid)** and the SwinUNetR **(bottom)**

3.6 Feature Extraction and Feature Engineering

The segmentations for all 458 MRIs for later feature extraction were created with the nnU-Net architecture, as it achieved the best results overall (see subsection 3.3.2).
As described earlier, the U-Net architecture was trained in five folds within the nnU-Net framework for four different dataset combinations for training and test datasets. This led to a total of twenty different neural networks for whole liver segmentation. Predictions on all 458 MRI acquisitions were made using ensembled predictions. For each dataset combination, the ensembled softmax output probabilities (of the five different folds) for every subject were saved to the disk. This led to 4 different softmax probability outputs for the four different datasets, which were then ensembled to get the final predictions for all patients.
Those segmentation maps were refined with a custom post-processing pipeline, which will be described in this section.

3.6.1 Postprocessing of Segmentations

Due to the challenges in coregistering the T1-VIBE sequences, particularly for fine vascular structures (as described in section 4.1), the segmentation maps did not always align perfectly with all five phases, occasionally aligning with only some of them for small structures. Thus, they were refined with a custom pipeline to ensure reliable results for later feature extraction, relying on multiple phases that are offset against each other (see subsection 3.6.2).
To achieve this, the first step is to split the labelmaps into seven different binary representations of their corresponding labels. This process is illustrated for one subject, containing all pathologies and structures to be segmented, in figure 3.11. If not specified otherwise, the following steps apply to all extracted tissues but will solely be explained using the liver as an example.

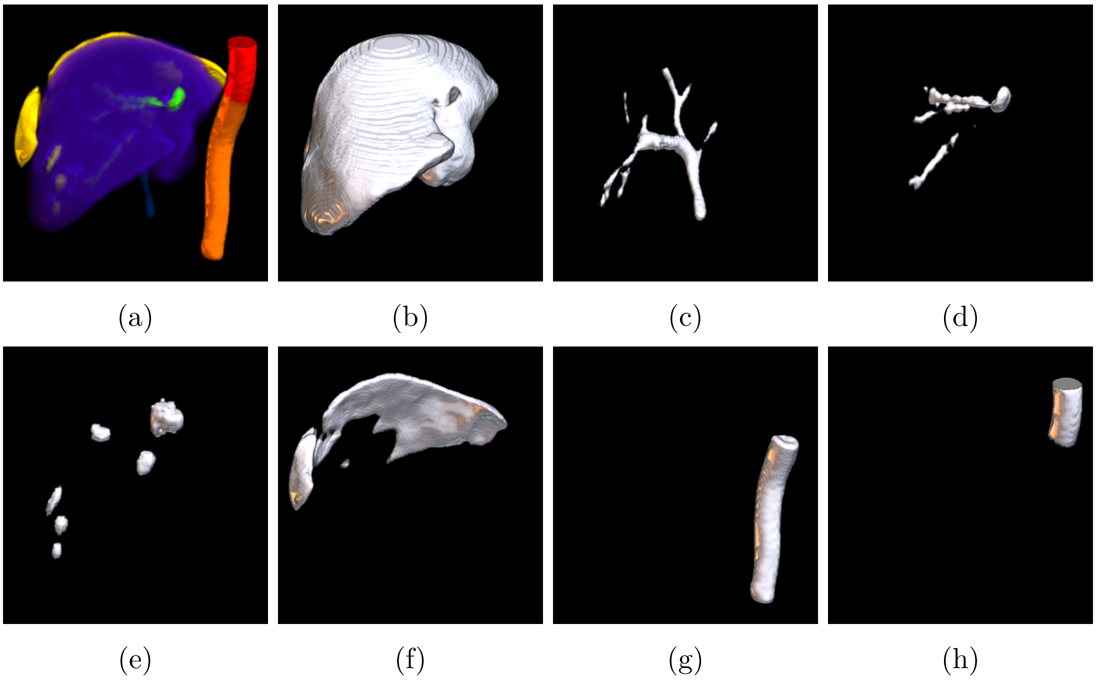


Figure 3.11 Rendered three-dimensional representation of the separation and binarization of the segmentations: the initial labelmap **(a)** with 7 labels is split into 7 binarized labelmaps (**(b)** to **(h)**). The labels are as follows: **(b)** Liver, **(c)** Portal vein, **(d)** Hepatic veins, **(e)** Lesions, **(f)** Ascites, **(g)** Abdominal aorta, and **(h)** Thoracic aorta. All rendered volumes share the same orientation.

For the upcoming visualizations in this subsection, a patient with better overall health than the one shown in figure 3.11 was picked, because of improved visibility for all structures and processing steps.
The binarized label masks are applied to all five phases of the T1-VIBE sequence, yielding five NIfTI files for every label, each containing only the corresponding tissue in its respective phase. This process is illustrated for the liver label of one subject in figure 3.12.

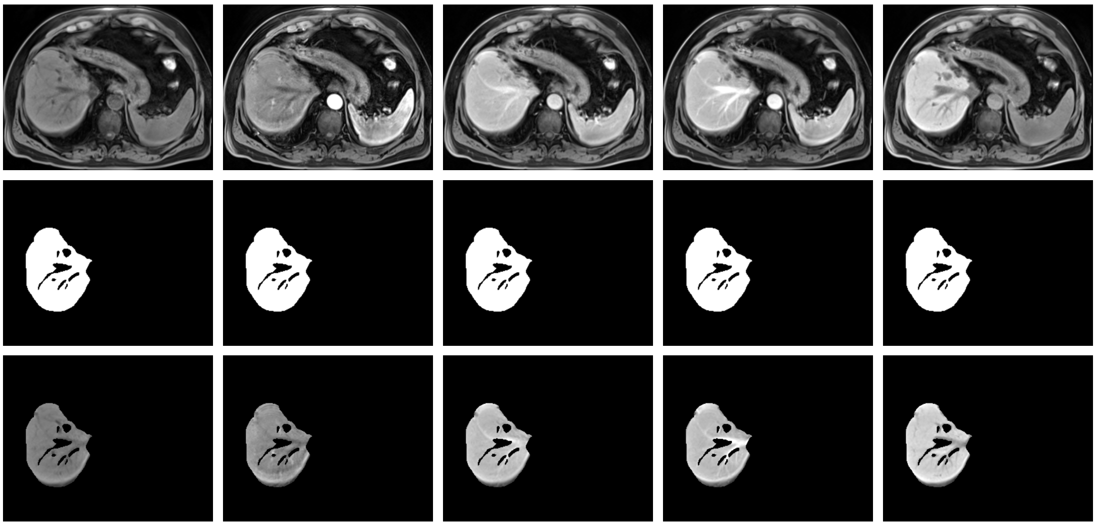


Figure 3.12 All images display axial slices, and the entire figure illustrates the process of liver extraction with the binarized segmentations from the MRIs. The **(top row)** presents all phases of the T1-VIBE sequence for a single patient from the cohort. The **(middle row)** shows the initial binary segmentation map, derived from the nnU-Net, which is multiplied by their corresponding phase above. At this stage of postprocessing, the same segmentation map is applied uniformly across all phases. The **(bottom row)** features axial representations of the results, specifically the extracted liver for each phase. From left to right, the images correspond to the five phases of the T1-VIBE sequences in their acquisition order: native, arterial, late arterial, portal venous, and hepatobiliary late phase.

In the next step, these segmented liver images undergo post-processing to remove intensity outliers. This step is crucial for eliminating remaining non-liver tissues in all phases, such as liver veins or portal veins, that may overlap with the liver parenchyma segmentation in some of the phases, due to problems in coregistration. Two different approaches were tested for this post-processing. One involves a widely used statistical approach for intensity outlier filtering in MRI, where all resulting intensities I' in the image are determined with the condition [164, 165, 166, 167]:

$$I' = \begin{cases} 0 & \text{if } I < \mu - 3\sigma \text{ or } I > \mu + 3\sigma \\ I & \text{otherwise} \end{cases} . \tag{42}$$

There, I is the intensity value of the current voxel, μ is the mean liver intensity within the liver label, and σ is the standard deviation of the liver intensity within the mask.

The second method, known as percentile clipping, is a more aggressive implementation for refinement. It establishes intensity thresholds based on phase-specific percentiles of the intensity distribution in the given label area, thereby excluding extreme values that are likely indicative of e.g. non-liver tissues. For phases like the arterial and late phase, where the liver is brighter than the veins, the boundaries are chosen differently than for phases like late arterial and portalvenous, where the appearance is the other way around (see figure 3.13 **(top row)** or 2.8). For all labels, values were clipped to their phase specific percentiles within their respective masks for each phase of the T1-VIBE sequence, resulting in refinement of the initial labels.

Both of those methods result in five refined extracted liver parenchymas, containing fewer non-liver tissue components that are different from each other. Due to the reason that a majority of the extraced features for liver function estimation in this thesis are based on relative enhancement of signal intensities in the corresponding tissues (see subsection 3.6.2), the extracted livers must be consistent accross all phases from the VIBE sequence. Otherwise, offsetting the phases against each other would result in unreliable measurements. Subsequently, those refined liver images are then binarized again and multiplied together, to ensure that only the regions consistently identified as liver parenchyma across all phases are retained, resulting in a final, robust liver mask. This whole process is depicted in figure 3.13.

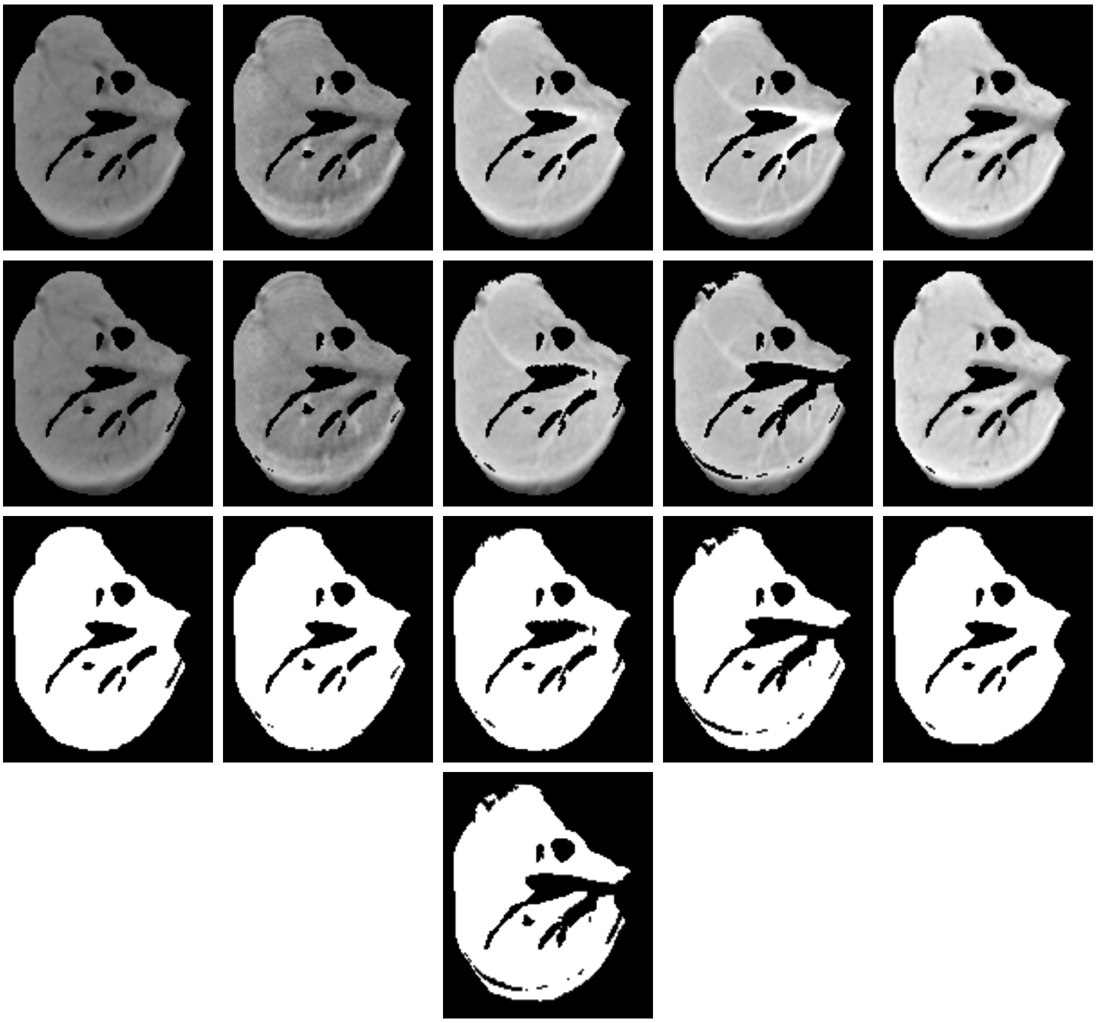


Figure 3.13 The **(first row)** shows the extracted livers with the segmentation map, generated by the nnU-Net. In the **(second row)**, there are the extracted liver parenchymas after percentile clipping of the intensities and the **(third row)** row depicts the binarized versions of the percentile clipped images. From left to right, there are the corresponding axial representations from the five phases of the T1-VIBE sequences in the correct order of acquisition. The slice depicted in the **(last row)**, shows the resulting final binary mask, derived by multiplication of all five phase-specific masks with each other.

A comparison between the initial mask, the one generated by percentile clipping and the one derived from the statistical approach can be found in figure 3.14 in the left, middle and right position, respectively.

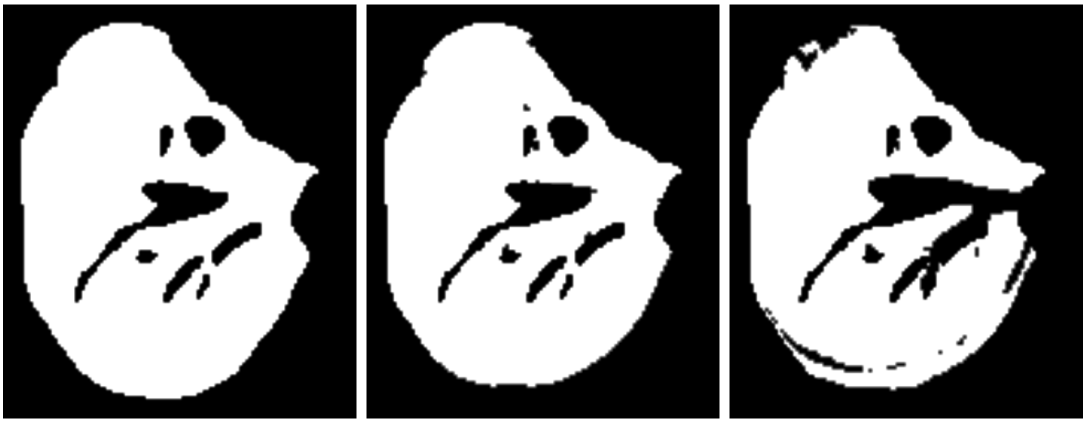


Figure 3.14 This figure depicts axial representations of three distinct liver segmentation label maps. The **(left)** image shows the liver mask before postprocessing. The **(middle)** image is generated using the statistical approach, utilizing the mean and standard deviation of the intensities. The **(right)** image is derived from percentile clipping. This comparison demonstrates that the percentile-clipping method is a more aggressive refinement technique.

In the last step, this final liver mask is applied to the original five phases of the T1-VIBE sequence. By multiplying the final liver mask with every MRI from each phase, the final extracted liver images are obtained. These images now contain a consistent representation of the liver tissue across all phases, mostly free from extraneous tissues and intensity outliers. This is illustrated in figure 3.15, where one can see the refinement in all five phases based on percentile clipping **(bottom)** and with the statistical approach with the mean and standard deviation of the signal intensities in the mask **(mid)**, compared to the initial segmentation, derived directly from the nnU-Net **(top)**.

From this figure, one can observe that in the statistical approach, there is still much left from the hepatic veins and also artifacts that appear hyperintense due to breathing during signal acquisition (especially visible in phase 2, 4 and 5, where there are hyperintense lines at the border of the liver). The approach with percentile clipping removes almost all of the remaining veins and artifacts, leading to much more reliable measurements of the REIs, hence all following steps for liver function estimation were based on the labels, refined with this method.

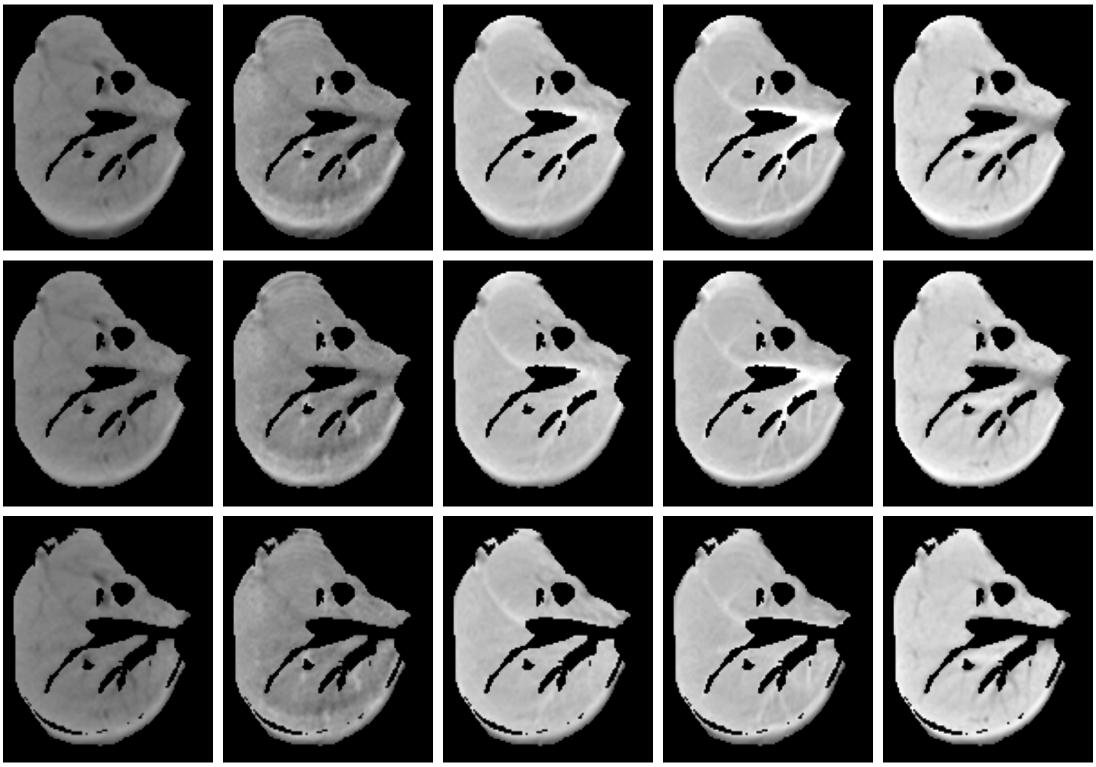


Figure 3.15 This figure depicts the axial representation of the segmented livers: **(Top)** before postprocessing, **(mid)** refined with the statistical approach and **(bottom)** post-processed with percentile clipping of the intensities.

In this whole process, some of the liver parenchyma is also removed due to intensity differences caused by inhomogeneities in the observed B-field and by artifacts due to breathing, as explained before. However, the benefit outweighs the harm, as it is more important not to include other tissue types in the feature extraction than losing some. This gets clearer, when looking at other current studies in this field that solely use a few Regions of Interest (ROIs) to correlate liver function with, for example, the relative intensity enhancement from the native to the hepatobiliary late phase for the liver parenchyma. Losing some of the liver still provides a much more comprehensive feature extraction than the subjective placement of single ROIs (see next subsection 3.6.2).

This meticulous process ensures that the liver segmentation is reliable for the following tasks, facilitating further analyses and more robust feature extraction.

3.6.2 Extraction of Meaningful Features

Studies have shown that the relative enhancement indices of the liver parenchyma, hepatic veins, portal veins and abdominal aorta from comparisons of one phase of the T1-VIBE sequence to another, as well as the liver volume can be strong indicators for liver function assessment [168, 169, 170, 171, 172, 173, 174, 175].

Other studies use so-called T1-maps and based on those, calculate reduction rates of T1 (rrT1) or general changes in the relaxation rate with measurements of T1 relaxation time from before and after contrast administration in the liver parenchyma. In those studies they were able to find strong correlation between liver function and those features [176, 177, 178, 179]. However, T1 maps are typically not acquired in daily clinical routine, but rather for research purposes, because of required long breath-hold time during acquisition of single slices and therefore, even for research purposes, only a small amount of slices will be acquired for abdominal imaging tasks. For some of the patients in this study, T1-maps were available, but only contained three slices per patient. Therefore, coregistration was not possible and thus they wouldn't have fit into the goal of building a fully automated pipeline for liver function assessment and manually placing regions of interest into those slices would have defended the whole purpose of this study, because all those pre-mentioned studies extract features from the images by only manually placing 3-9 ROIs into the corresponding tissues, do their measurements inside those and take this as representative for the whole liver.

Contrary to these approaches, there are studies showing that liver damage and liver function are not necessarily uniform across the entire liver parenchyma and therefore, regional differences can be observed [180, 181, 182].

Consequently, manually placing ROIs into the liver parenchyma and other tissues is prone to inter- and intra-rater variability and can introduce unwanted bias into the measurements.

Subsequently, in this thesis, the feature extraction builds on the whole segmented structures, instead of just small portions in form of ROIs.

For the liver parenchyma, hepatic veins, portal veins, abdominal aorta and the thoracic aorta, the relative enhancement indices (REIs) were calculated with the following formula:

$$REI_n^x = \frac{SI_{phase_n}^x - SI_{native}^x}{SI_{native}^x}. \tag{43}$$

There, x refers to the desired tissue and n to the corresponding phase to which the relative contrast enhancement is calculated. n=1, 2, 3 and 4 refer to arterial, late arterial, portalvenous and hepatobiliary late phase in this order. Mean Signal Intensity (SI) values are calculated over all voxels of the whole tissue under consideration, ensuring that the REI reflects the overall enhancement of the tissue. Differences in intensity from phase n to the native phase are then normalized by dividing through the mean SI values of the given tissue in the native phase. This normalization is crucial because MRI signal intensities are arbitrary and can vary due to different factors such as scanner settings and patient-specific conditions. By normalizing with the native phase, which is acquired using the same scanner and under the same B-field inhomogeneities, the REIs become comparable across different measurements and conditions.

Additional features from the liver have been extracted with Radiomics [183, 184]. For this, the extracted liver from phase n was subtracted by the liver from the one from the native phase and then normalized by the native liver, as described before. The radiomics features were then extracted from these combined and normalized representations of the liver. Features included the *interquartile range* of the intensities in those representations of the liver, median intensity and the values for the 10^{th} and 90^{th} percentile of the intensities found inside the extracted liver. Those have been calculated for each of the 4 phases offset against the native phase. This whole process results in 36 features based on intensity enhancement.

Additionally, liver, lesion, and ascites volumes were used as features, as these can correlate with liver function. Ascites typically appears only in later stages of chronic liver diseases and is part of other liver function scores, such as the Child-Pugh Score [185]. The volumes are calculated by counting the voxels in the initial label map (e.g., the liver) before post-processing and multiplying this number by the corresponding voxel sizes for all three orientations. This ensures compatibility with images that have not been resliced. The rationale behind this is that post-processing was introduced only to address coregistration challenges to ensure, only the tissue under consideration is present in all five phases for feature extraction that relies on multiple images. The low volume difference for the liver parenchyma segmentations compared with their ground truth (see table 4.2) justified the choice of using the original segmentation maps directly for this task.

For the lesions and ascites, the volume differences between manual labels and segmentations are more significant, but post-processing did not improve this. This can be explained by the high Positive Predictive Value (PPV) and low True Positive Rate (TPR) values, indicating that the labels include much less of the desired tissues than the manual annotations but also a low number of false positives. Therefore, the initial segmentations of these structures were used for volumetry. Even if not captured very accurately through the automatic segmentations, these features provide insight into whether these pathologies are present or not.

Additional four features are generated by multiplying the liver Volume with the REIs of the liver for all phases.

This lead to a total of 43 features, derived from the MRI images.

3.6.3 Feature Engineering

Feature engineering is a crucial step in the machine learning pipeline for liver function prediction in this thesis. This ensures that the most relevant and informative characteristics of the data for each liver function score are captured and utilized effectively. The primary goal is to improve the data's quality and relevance as well as the predictive power of those machine learning models by selecting, modifying and creating features that provide additional insights into the data, leading to more accurate and robust models.

The feature engineering includes removing constant, quasi-constant and highly correlated features. For each of the remaining ones, correlation with the target variable is calculated and only features that significantly correlate with the corresponding liver function score, are used for model training and evaluation. Because of the distinct scores and cohorts, this results in a different amount of corresponding features that are used for the models.

For the MELD-, LiMAx- and ALBI-score, the feature engineering methods lead to 15, 20 and 22 features that are used for building the machine learning pipelines, respectively.

Several features consistently remained in all final dataframes after feature engineering. Notably, liver volume ($\mathrm{Vol}^{\mathrm{Liver}}$) was retained for every cohort, underscoring its fundamental role in assessing liver size and capacity, which are crucial indicators of liver health and useful for pre-operative planning in case of resection.

Additionally, the products of the liver volume with enhancement indices, specifically $\mathrm{REI}_2^{\mathrm{Liver}} \cdot \mathrm{Vol}^{\mathrm{Liver}}$, $\mathrm{REI}_3^{\mathrm{Liver}} \cdot \mathrm{Vol}^{\mathrm{Liver}}$, $\mathrm{REI}_4^{\mathrm{Liver}} \cdot \mathrm{Vol}^{\mathrm{Liver}}$ were consistently included. These features highlight the importance of enhancement patterns in different phases: late arterial, portal venous, and hepatobiliary late phase, respectively.

The enhancement index $\mathrm{REI}_4^{\mathrm{Liver}}$ alone, which pertains to the HBP-20 phase, was also a key feature across all models. This indicates the significance of liver enhancement behavior in the hepatobiliary late phase as a predictor of liver function as shown in other studies [170, 168, 131, 169, 172, 173]. The consistent presence of these features across all scoring systems suggests their robust predictive power and their critical role in liver function assessment.

For the distinct cohorts, the feature engineering process identified several additional features of significance. A full overview of all features used for each cohort can be found in the Appendix in table A.1.

3.6.4 Model training and evaluation

For estimating liver function, a diverse set of machine learning models, each with multiple hyperparameters, were trained and compared. Following the initial feature engineering steps, these models were trained and fine-tuned using GridSearchCV from scikit-learn [186] to find the optimal hyperparameter configurations.

Regression models: The models used for regression included Random Forest, Support Vector Regression, XGBoost, Multi-layer Perceptron, Lasso and Ridge Regression, and Elastic Net. Hyperparameter tuning was performed using GridSearchCV with 5-fold cross-validation, focusing on metrics such as Mean Absolute Error (MAE), R-squared (R^2), Root Mean Squared Error (RMSE). The top three models with the lowest average MAE across all folds were selected for further analysis. There, stratified 5-fold cross-validation was performed, where one fold was used as a hold-out test set and the other four as the training data. The three previously chosen top models were then trained and evaluated for each combination, by ensembling their predictions and test those against the single-model-predictions from the top performing model, regarding R^2, MAE, RMSE and normalized versions of MAE and RMSE. Additionally, the Pearson- and Spearman correlations were calculated between the predictions and the ground-truth values. The results were visualized by plotting the corresponding predictions against the ground-truth values. 95%-confidence intervals were calculated for each metric, using bootstrapping.

The regression was performed on the cohorts with LiMAx-, MELD- and ALBI-score.

Classification models: The models used for classification included Random Forest, Support Vector Machine (SVM), XGBoost, Logistic Regression and Multi-layer Perceptron (MLP). Hyperparameter tuning was also done with GridSearchCV with 5-fold cross-validation, but with focus on accuracy and F1 score. The top three models based on their F1-scores were selected for further evaluation. Similar to regression, stratified 5-fold cross validation was performed for training and testing the models. For every run, the training set was oversampled with *BorderlineSMOTE*, focussing on generating synthetic samples for the minority class, particularly for those samples that are near the borderline with the majority class, which are more likely to be misclassified [187, 188]. Additionally to the single-model predictions, a soft-voting ensemble approach was employed to aggregate predicted class-probabilities from these models, and compare the prediction performance on the hold-out test set against each other. 95%-confidence intervals were calculated for accuracy and F1-score, using bootstrapping.

To visualize the classification results, ROC curves were plotted for each class, and a confusion matrix was generated. All metrics, including the F1 scores, accuracy, AUCs and ROCs were calculated across all five test sets to provide an overall performance metric and to ensure a comprehensive evaluation of the classification models' effectiveness.

The MELD-, LiMAx- and ALBI-scores were divided into three groups, indicating normal liver function (NLF), moderate liver disease (MLD) and severe liver disease (SLD). For the ALBI- and LiMAx-score, the official cutoff values for division into three classes were used, as described in section 2.3. For the MELD score, published cutoff values from literature were used, where patients with MELD ≤ 10 represent NLF, MELD between 11 and 18 refer to MLD and severe liver disease is observed for patients with a MELD > 18 [189]. Because of the very small amount of patients with a MELD score above 18, another approach for this cohort was to predict whether a patient has normal liver function or not. In other words, to predict if the MELD is above 10. In the following, those three classes for MELD and LiMAx will be referred to as grades, like for the ALBI-scores.

To evaluate whether the predictive performance of liver function scores was truly enhanced by using a larger feature set than other approaches that only use the relative enhancement index $\mathrm{REI}_4^{\mathrm{Liver}}$ or a combination of this with the liver volume $\mathrm{REI}_4^{\mathrm{Liver}} \cdot \mathrm{Vol}^{\mathrm{Liver}}$ [170, 168, 131, 169, 172, 173], the previously described model training and evaluation was also performed twice, using each of these features individually. This comprehensive comparison was necessary because the published approaches rely solely on regions of interest (ROIs), which can be subjective and liver function can vary across different regions within the liver, as previously described. Additionally, differences in cohort distribution can also significantly affect the results, making direct comparisons across different studies inappropriate. All results for regression and classification are shown in section 4.3.

Chapter 4

Results and Discussion

4.1 Coregistration

The coregistration of MRI volumes across different phases of the T1-VIBE sequence was performed to ensure accurate alignment and facilitate subsequent analyses. The results of this coregistration process were evaluated both visually and quantified by using several metrics: *Mutual Information* (MI), *Normalized Cross Correlation* (NCC), *Mean Absolute Error* (MAE), and *Structural Similarity Index* (SSIM). A detailed explanation of those metrics is provided in subsection 3.3.1.
The results are displayed in table 4.1

Table 4.1 This table illustrates the coregistration performance on the whole dataset for different phases from the T1-VIBE sequence as the fixed image. All MRI Volumes for each patient were coregistered to the given phase as the fixed image. Best and second best results are bold and italic, respectively. Numbers in brackets denote the 95%-confidence intervals for each metric.

Fixed image	MI	NCC	MAE	SSIM
Native	0.90 [0.90, 0.91]	0.91 [0.91, 0.92]	1370 [1305, 1435]	0.83 [0.82, 0.83]
Arterial	0.92 [0.91, 0.93]	0.91 [0.91, 0.91]	1096 [1034, 1158]	0.87 [0.86, 0.87]
Late arterial	**1.03 [1.02, 1.05]**	**0.95 [0.95, 0.95]**	**804 [749, 860]**	**0.87 [0.87, 0.88]**
Portalvenous	*1.02 [1.01, 1.04]*	*0.95 [0.94, 0.95]*	*822 [767, 878]*	*0.87 [0.87, 0.87]*
HBP-20	0.92 [0.91, 0.93]	0.92 [0.92, 0.93]	1095 [1028, 1162]	**0.87 [0.87, 0.88]**

Taking the late arterial and portalvenous phases as the fixed images, consistently demonstrated superior performance across all metrics. The late arterial phase achieved the highest MI (1.03) and NCC (0.95), the lowest MAE (804), and a high SSIM (0.87). Similarly, the portalvenous phase showed strong results with an MI of 1.02, NCC of 0.95, MAE of 822, and SSIM of 0.87. These findings indicate that the late arterial phase provides the best alignment with other phases, making it an ideal candidate for the fixed image in coregistration.

In contrast, the coregistrations to the native phase exhibited lower performance, with an MI of 0.90, NCC of 0.91, MAE of 1370, and SSIM of 0.83. These results suggest that the native phase has less shared information and more significant intensity differences with other phases, potentially due to drastic variations in contrast and intensity.

Those results were in complete aggreement with the visual inspection of the coregistrations, where the late arterial phase showed the best performance. Subsequently, the dataset where all phases were coregistered to the late arterial phase was chosen for all following steps.

However, due to the nature of VIBE sequences, there are issues where not all patients are able to inhale the same amount of air for each phase's acquisition. Unlike brain MRIs, which maintain consistent structures across all modalities but may have orientation differences, the issues in abdominal imaging with breath-hold techniques result in significant deformations of the abdomen and internal organs in all directions from one phase to another. This is compounded by misalignments due to patient movement and is particularly problematic because of the soft, deformable tissues involved. When inhaling, the diaphragm contracts and moves downward to enlarge chest cavity and reduce the pressure inside the lungs compared to the outside atmosphere and thus creating a vacuum, causing air to be drawn into the lungs through the airways. This contraction leads to movement and compression of the abdominal organs. An example with raw images before preprocessing is shown in figure 4.1. There, one can see slices of the sagittal (top) and axial (bottom) representations for the arterial (left) and portal venous (right) phases from the T1-VIBE sequence. Images with same orientation share the exact same FOV. The lines and arrows are there for reference.

The top horizontal lines in the sagittal views show how the heart (top line) and vessels (bottom lines) move up and down, when inhaling more or less air. The vertical lines there show the abdominal border in the arterial (right line) and portal venous (left) line, where one can see how much difference between them can be observed. For the axial views, the horizontal lines indicate borders of the liver (top line) and the spleen (bottom line) in the arterial phase and how they are different from the portal venous phase. The arrows point to the exact same spatial positions in both images. The hepatic veins are visible in the portal venous phase, but not in the arterial phase, due to different amount of air inhaled.

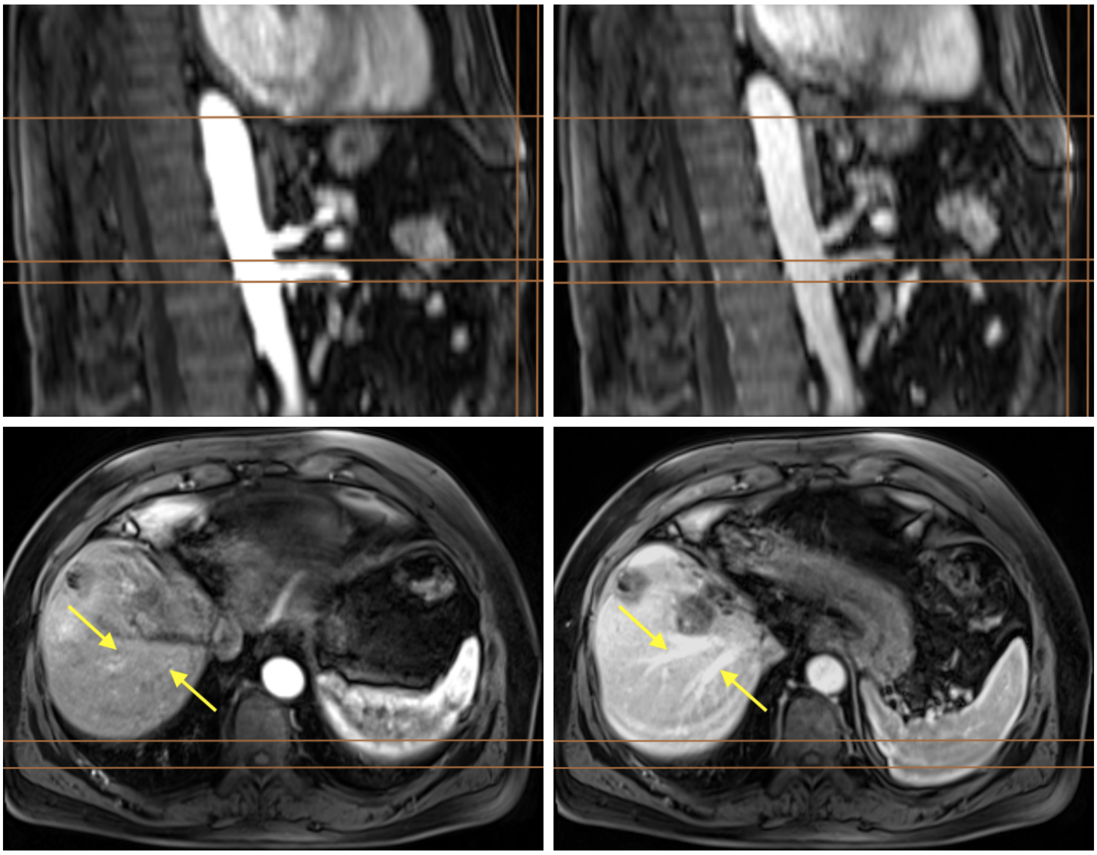


Figure 4.1 Arterial and portal venous phases from the T1-VIBE sequence for one subject before coregistration. The arterial phase is shown in the **(left)** images, and the portal venous phase in the **(right)**. The **(top row)** displays the sagittal view with the patient's back on the left and abdomen on the right. The **(bottom row)** shows the axial orientation of the same MRI. Images in the same row depict the exact same slice. Due to variations in breath intake, the abdominal organs do not remain in a fixed position. Brown reference lines aid visualization. Vertical lines in the top images indicate the moving abdominal border: the right line marks the border in the arterial phase, while the left line marks the border in the portal venous phase, where less air was inhaled. Horizontal lines in these images illustrate the movement of the heart (top line) and vessels (bottom lines) between phases due to breathing. Yellow arrows in the axial images point to identical spatial positions where the hepatic veins are visible in the portal venous phase. The horizontal lines indicate the borders of the liver (left) and spleen (right) during the arterial phase and demonstrate how they shift due to breathing.

Those characteristics made it challenging to get precise coregistration for fine vascular structures as the hepatic veins, hence in a lot of cases it was not as accuracte as desired.

The exact same slices for the given example from figure 4.1, but after coregistration, can be found in figure 4.2 with the reference lines and arrows at the same positions in the images.

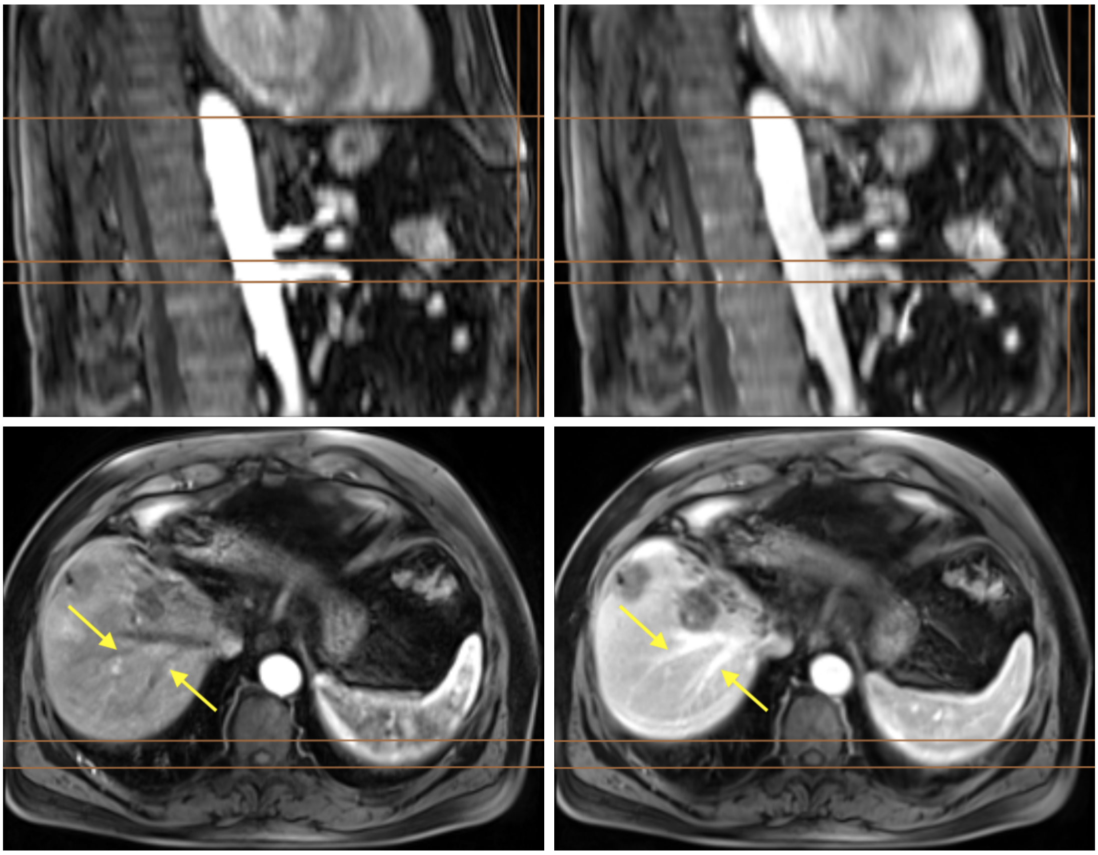


Figure 4.2 Arterial and portal venous phases from the T1-VIBE sequence for the same subject after coregistration. The arterial phase is shown in the left images, and the portal venous phase in the right. The top row displays the sagittal view with the patient's back on the left and abdomen on the right. The bottom row shows the axial orientation of the same MRI. Images in the same row depict the exact same slice. There, the reference lines indicate an overlap of the given structures in both phases and the yellow arrows show the hepatic veins. In contrast to the images before coregistration, they are now visible in the same slice in the arterial phase, but still not exactly at the same positions as in the portal venous one.

There, it can be observed that the differences of heart and vessel positions can't be observed in the sagittal views anymore, between both phases. Additionally, the liver and spleen borders for the axial views now align in both phases. Hepatic veins are also visible in the arterial phase, now. However, they still don't overlap perfectly.

Those challenges make it hard for the network architectures to learn the correct features and is especially problematic for the feature extraction in a later stage of the pipeline, where features from multiple corresponding MRI phases are offset against each other with the help of segmentated structures. This raised the need for an implementation of a custom post-processing pipeline that is applied before feature extraction and was described in subsection 3.6.1.

4.2 Segmentation

In this section, the segmentation performance of three deep learning models - standard nnU-Net [115], ResEnc nnU-Net [162], and Swin UNETR [118] - on the given dataset with seven labels is presented. The models were evaluated on four different dataset combinations (training- and test-dataset) using 5-fold cross-validation, and the predictions were ensembled to ensure robustness. The key metrics used for evaluation are explained in detail in subsection 3.3.2. The averaged results for the four different test datasets are shown in table 4.2. The best and second best results for each metric are highlighted in bold and italic, respectively, and the 95% confidence intervals are provided in brackets.

Both nnU-Net variants (standard and ResEnc) performed exceptionally well on the **liver parenchyma** segmentation task, achieving high DSC, IOU, and TPR scores. The standard nnU-Net slightly outperformed the ResEnc nnU-Net in PPV and VD, indicating a marginally better precision and volume accuracy. The Swin UNETR, while delivering competitive results, lagged behind the nnU-Net variants in all metrics. This suggests that the traditional convolutional architecture of nnU-Net remains highly effective for liver parenchyma segmentation

For the **portal vein** and **hepatic veins**, the standard nnU-Net achieved the best performance across most metrics. The ResEnc nnU-Net showed competitive results but was slightly behind in TPR for hepatic veins. The Swin UNETR's performance was notably lower, particularly in DSC, IOU and TPR, indicating less accurate and complete segmentations.

The segmentation of **lesions** proved challenging for all models, with the standard nnU-Net achieving the highest DSC and IOU. Additionaly, the LFPR was the lowest for the standard nnU-Net. The ResEnc variant was slightly behind, while the Swin UNETR had the lowest scores across most metrics. This suggests that while nnU-Net variants can handle complex structures to some extent, there is still room for improvement, particularly in segmenting irregular and small structures like lesions.

The same suggestions are valid for the segmentation of **ascites**. There, the standard nnU-Net outperformed the other models in DSC and IOU, with the ResEnc nnU-Net following closely. The Swin UNETR showed significantly lower

Table 4.2 This table illustrates the segmentation performance of all tested architectures for all seven labels. All approaches were 5-fold cross validated on the exact same datasets and ensembled predictions were used. Best results are bold and the 95%-confidence intervals are denoted in brackets.

nnU-Net[115]	**DSC**	**IOU**	**PPV**	**TPR**	**LFPR**	**LTPR**	**VD**
Liver parenchyma	**0.97 [0.97, 0.98]**	**0.95 [0.94, 0.96]**	**0.97 [0.96, 0.98]**	**0.98 [0.98, 0.99]**	-	-	**0.03 [0.02, 0.04]**
Portal vein	**0.83 [0.80, 0.87]**	**0.73 [0.69, 0.76]**	**0.86 [0.82, 0.89]**	**0.82 [0.78, 0.86]**	-	-	**0.12 [0.08, 0.16]**
Hepatic veins	**0.78 [0.77, 0.80]**	**0.65 [0.63, 0.68]**	**0.83 [0.80, 0.85]**	0.77 [0.74, 0.80]	-	-	**0.18 [0.14, 0.22]**
Lesions	**0.56 [0.48, 0.64]**	**0.44 [0.36, 0.51]**	**0.78 [0.71, 0.86]**	**0.52 [0.44, 0.61]**	**0.10 [0.04, 0.17]**	**0.65 [0.57, 0.74]**	**0.66 [0.38, 0.93]**
Ascites	**0.44 [0.16, 0.71]**	**0.32 [0.09, 0.55]**	*0.74 [0.42, 1.10]*	**0.36 [0.09, 0.63]**	-	-	*0.69 [0.25, 1.12]*
Abdominal aorta	**0.96 [0.96, 0.97]**	**0.93 [0.92, 0.94]**	**0.96 [0.96, 0.97]**	**0.97 [0.96, 0.98]**	-	-	**0.04 [0.03, 0.05]**
Thoracic aorta	**0.93 [0.91, 0.95]**	**0.87 [0.84, 0.90]**	**0.92 [0.89, 0.95]**	0.94 [0.93, 0.96]	-	-	**0.10 [0.07, 0.14]**
ResEnc nnU-Net[162]							
Liver parenchyma	**0.97 [0.97, 0.98]**	**0.95 [0.94, 0.96]**	*0.96 [0.95, 0.97]*	**0.98 [0.98, 0.99]**	-	-	*0.03 [0.02, 0.05]*
Portal vein	*0.83 [0.79, 0.86]*	*0.72 [0.68, 0.76]*	**0.86 [0.82, 0.89]**	*0.81 [0.77, 0.85]*	-	-	0.13 [0.09, 0.17]
Hepatic veins	*0.78 [0.76, 0.80]*	*0.64 [0.62, 0.67]*	**0.83 [0.80, 0.85]**	0.75 [0.72, 0.78]	-	-	*0.18 [0.15, 0.21]*
Lesions	*0.51 [0.43, 0.60]*	*0.40 [0.32, 0.48]*	*0.77 [0.69, 0.85]*	*0.47 [0.38, 0.56]*	*0.19 [0.10, 0.27]*	0.63 [0.54, 0.72]	*0.73 [0.47, 0.98]*
Ascites	*0.41 [0.18, 0.64]*	*0.28 [0.11, 0.46]*	**0.77 [0.43, 1.11]**	*0.31 [0.09, 0.55]*	-	-	**0.67 [0.42, 0.91]**
Abdominal aorta	**0.96 [0.96, 0.97]**	**0.93 [0.92, 0.94]**	**0.96 [0.96, 0.97]**	0.96 [0.96, 0.97]	-	-	**0.04 [0.03, 0.05]**
Thoracic aorta	**0.93 [0.91, 0.95]**	**0.87 [0.84, 0.90]**	*0.92 [0.89, 0.94]*	**0.95 [0.93, 0.96]**	-	-	*0.12 [0.07, 0.16]*
Swin UNETR[126]							
Liver parenchyma	*0.96 [0.95, 0.97]*	*0.92 [0.90, 0.94]*	0.94 [0.92, 09.6]	*0.98 [0.97, 0.98]*	-	-	0.09 [0.05, 0.13]
Portal vein	0.74 [0.70, 0.78]	0.60 [0.56, 0.64]	*0.82 [0.78, 0.86]*	0.69 [0.64, 0.74]	-	-	0.35 [0.26, 0.43]
Hepatic veins	0.65 [0.61, 0.70]	0.51 [0.46, 0.55]	*0.78 [0.74, 0.81]*	0.61 [0.56, 0.66]	-	-	0.48 [0.38, 0.59]
Lesions	0.47 [0.40, 0.55]	0.35 [0.28, 0.41]	0.65 [0.56, 0.74]	0.44 [0.37, 0.52]	0.55 [0.46, 0.63]	*0.63 [0.54, 0.73]*	1.47 [0.69, 2.25]
Ascites	0.28 [0.08, 0.49]	0.19 [0.04, 0.34]	0.45 [0.15, 0.74]	0.22 [0.05, 0.39]	-	-	3.99 [-2.76, 10.8]
Abdominal aorta	*0.94 [0.94, 0.95]*	*0.89 [0.88, 0.91]*	*0.95 [0.94, 0.96]*	0.94 [0.93, 0.96]	-	-	*0.11 [0.08, 0.14]*
Thoracic aorta	*0.87 [0.83, 0.90]*	*0.78 [0.74, 0.83]*	0.89 [0.86, 0.93]	0.86 [0.82, 0.91]	-	-	0.30 [0.21, 0.39]

performance, indicating its limitations in handling fluid-filled regions and precise boundary delineation. However, the values for this label should be treated with caution as the ground-truth labels were not perfect and visual inspection for all seven patients that showed signs of ascites, tended to give the impression that the segmentations from the standard nnU-Net were more accurate, as shown in figure 4.3.

Both nnU-Net variants performed similarly well on the **abdominal** and **thoracic aorta** segmentation, with the standard nnU-Net slightly ahead in TPR. The Swin UNETR, while competitive, was slightly behind in most metrics. These results suggest that nnU-Net's architecture is well-suited for segmenting tubular structures like the aorta.

A comparison of the model's segmentation with the ground truth annotations and the original image with no annotations for one subject with liver cirrhosis and ascites is shown in figure 4.3. The pink arrows with their direction towards the liver point to the vena cava inferior, which is not part of the original liver labels and is no liver parenchyma. Both nnU-Net variants were able to identify this as non-liver tissue, where the Swin UNETR falls short and classifies it as liver, too. For the pink arrow pointing to the ascites fluid in the images, one can see that the initial labels were not always perfect, showing that some parts were missed by the original label. The standard nnU-Net performs a comprehensive segmentation in this slice, where in the Residual Encoder version some part of the bottom is missing. The Swin UNETR captures this part, but is misses a bigger part that the other approaches were capable of. The yellow arrows show, where the nnU-Net variants were able to precisely segment the liver parenchyma border in this area, but the transformer model did not capture the small gap correctly. Additionally, the red ellipse shows, where the Swin UNETR alone was not sure, whether the given vessel is a hepatic vein or a portal vein and the rectangular box highlights an area, where it misclassified parts of the stomach as liver and lesion. However, the ground-truth annotation shows a small area of the border of a lesion (highlighted with a red ellipse) that none of the architectures was able to capture.

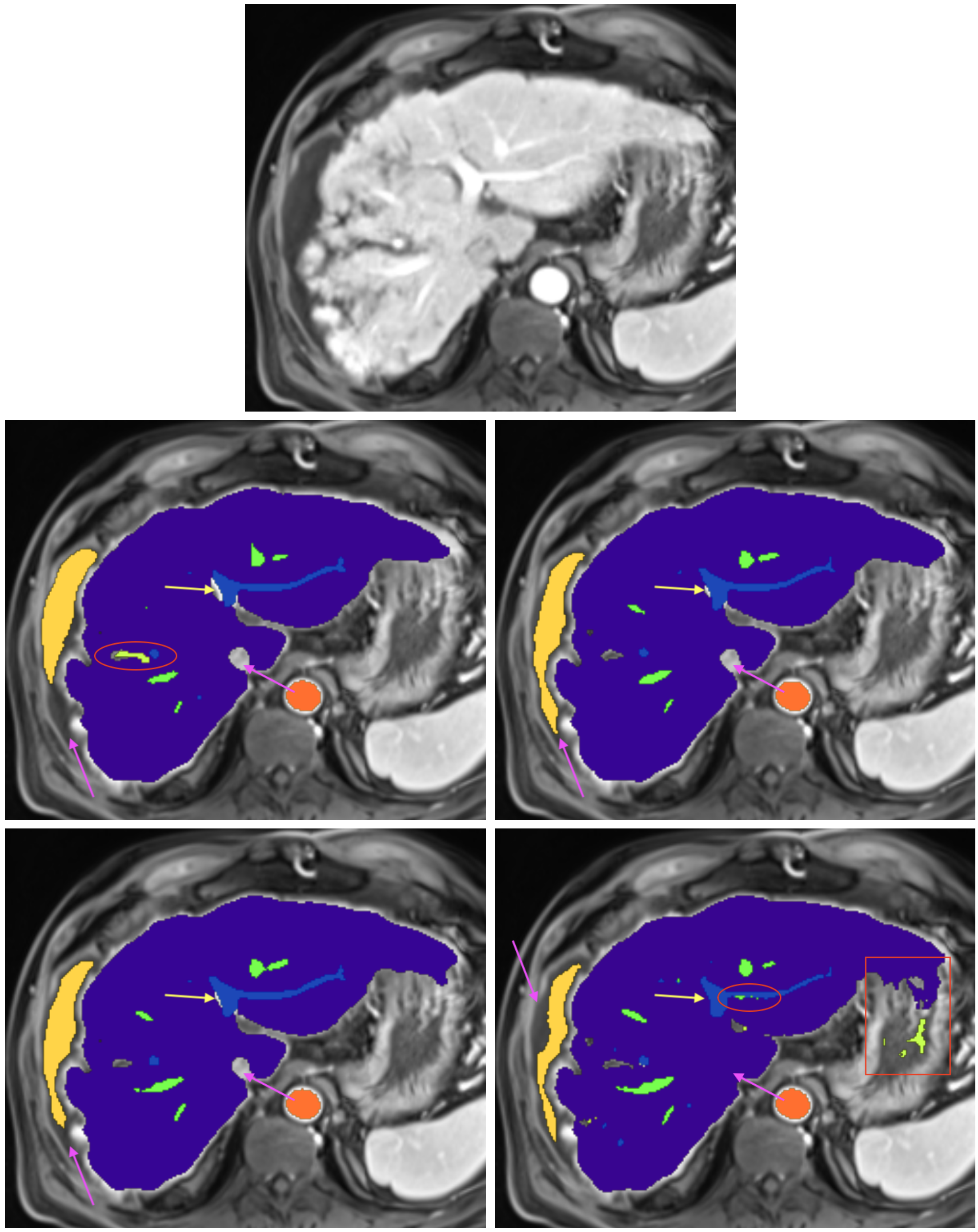


Figure 4.3 All images show the same axial slice of a patient with liver cirrhosis and ascites for the late arterial phase. First row is the given image with no annotations. Labels in the other slices are dark blue, light blue, green, dark yellow, light yellow, and orange for the labels liver, portal vein, hepatic vein, ascites, lesion and abdominal aorta, respectively. Left image in the second row depicts the ground-truth annotations, right one shows the segmentations from the standard nnU-Net. Bottom row illustrates the residual encoder nnU-Net's (left) and the SWIN UNETR's (right) segmentations. Arrows, ellipses and rectangular boxes highlight major differences between the segmentations from the models and the ground-truth annotations.

For another subject with impaired liver function, the model's segmentations are compared to the ground-truth annotations in figure 4.4. The yellow arrow indicates an area, where the nnU-Net models correctly captured the border of the liver parenchyma like in the ground-truth, but the Swin UNETR was not able to distinguish between liver and background. The pink rectangular boxes in the ground-truth annotation show hepatic veins and portal veins that none of the model was able to segment in that slice, correctly. The red ellipses in the residual encoder nnU-Net's and the Swin UNETR's segmentations indicate, where the models were not able to distinguish between portal veins and hepatic veins and therefore misclassified a lot of them, especially the latter.

All the described findings, visually or metric-wise, suggest that the traditional convolutional architecture of nnU-Net remains highly effective for medical imaging segmentation, aligning with previous studies that highlight the robustness of CNN-based models in medical image segmentation. Transformer-based models like Swin UNETR, while promising, often fall short of the performance achieved by CNN-based models, especially with datasets comprising multiple labels [162]. This study's results are consistent with the literature, which frequently reports nnU-Net as one of the top-performing models in various segmentation tasks [161, 162].

Because of those results and the significant longer training time for the ResEnc nnU-Net variant (see subsection 3.5), all segmentations for the following steps were generated with the standard nnU-Net framework that was introduced in 2021 [115].

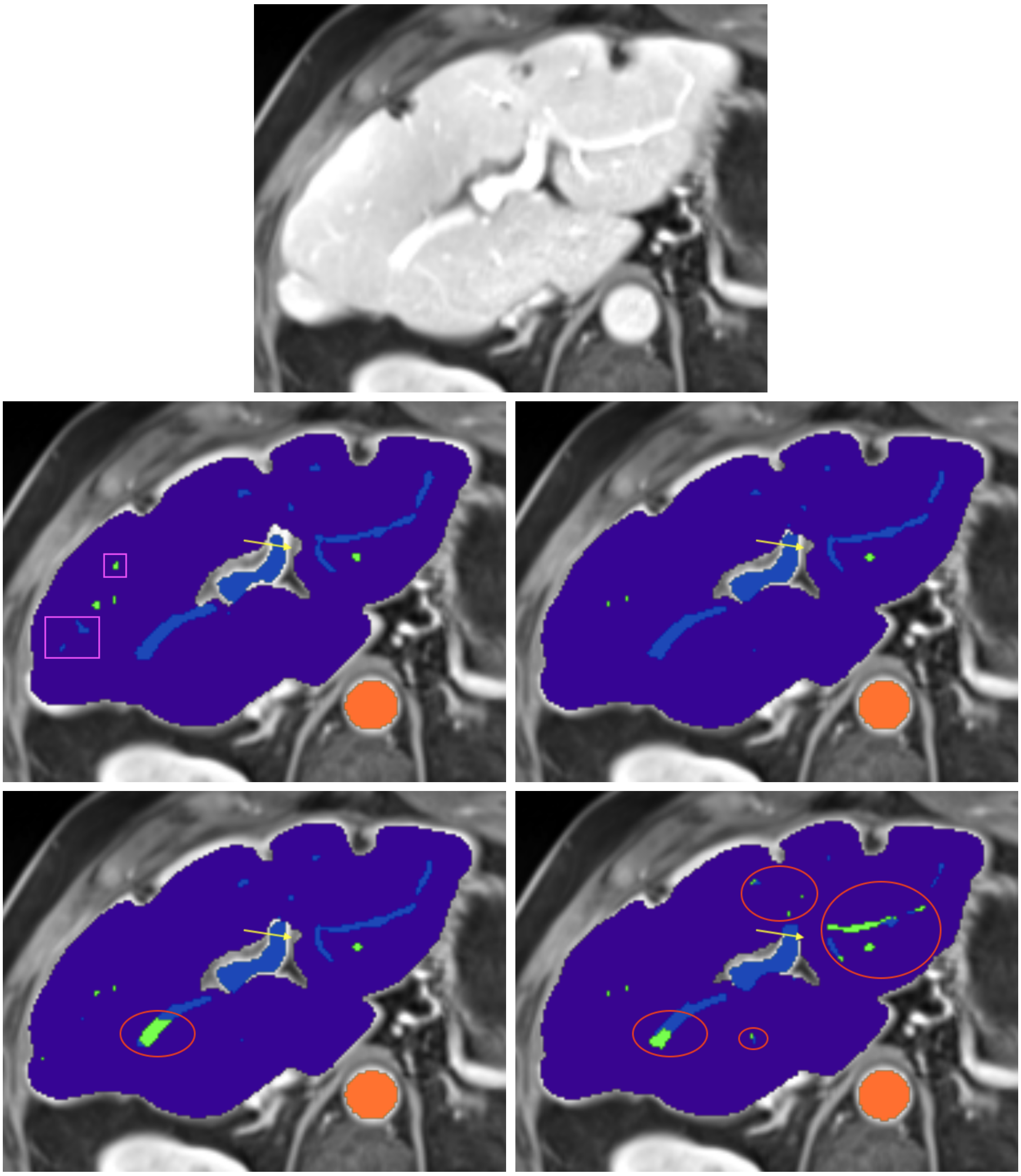


Figure 4.4 All images show the same axial slice of a patient with liver disease for the portalvenous phase. First row is the given image with no annotations. Labels in the other slices are dark blue, light blue, green and orange for the labels liver, portal vein, hepatic vein and abdominal aorta, respectively. Left image in the second row depicts the ground-truth annotations, right one shows the segmentations from the standard nnU-Net. Bottom row illustrates the residual encoder nnU-Net's (left) and the SWIN UNETR's (right) segmentations. Arrows and ellipses highlight major differences between the segmentations from the models and the ground-truth annotations. The pink rectangular boxes show hepatic and portal veins in the ground-truth annotations that all of the architectures missed.

4.3 Liver Function Estimation

This section presents the results for the evaluation of liver function estimation using both regression and classification models. The predictive performance of different models across three key liver function scores, such as ALBI, MELD, and LiMAx, is assessed. Those scores are critical for assessing liver health and guiding clinical decision-making. The analysis aims to determine the effectiveness of various modeling approaches in accurately estimating liver function with MRI derived features.

4.3.1 Regression for Liver Function Score Prediction

Table 4.3 compares the performance of regression models for MELD, LiMAx, and ALBI scores using both single (A) and ensembled (B) prediction approaches for the distinct feature sets derived by the feature engineering (X) (see table A.1), as well as for the models trained with only one feature. Once with $\mathrm{REI}_4^{\mathrm{Liver}}$ (Y) and once for $\mathrm{Vol}^{\mathrm{Liver}} \cdot \mathrm{REI}_4^{\mathrm{Liver}}$ (Z). Since the distinct scores are on different scales, normalized metrics such as Normalized Mean Absolute Error (NMAE) and Normalized Root Mean Square Error (NRMSE) are used to facilitate a fair comparison.
Figure 4.5 illustrates the relationship between the predicted and actual values for each liver function score for single models' predictions when using the corresponding comprehensive feature sets, derived by the feature engineering steps. The regression plots highlight the accuracy of the models, with the dashed red line indicating the ideal scenario where predictions perfectly match the ground truth. The green line represents the Pearson correlation, providing a visual measure of the strength and direction of the linear relationship between predicted and actual values.

MELD-Score

For the MELD-score with the full feature set, the top three performing models were Random Forest, configured with a maximum depth of 10 and 200 estimators; XGBoost, with a maximum depth of 3 and 10 estimators; and ElasticNet, with an alpha value of 0.1 and an L1 ratio of 0.1. The Random Forest model and the ensembled predictions showed similar performance, with an NMAE around 0.09 and an R^2 approximately 0.36-0.37, indicating moderate predictive accuracy.
According to the first plot in Figure 4.5 (generated solely with Random Forest predictions), the best correlation between predicted and actual values is observed for MELD-scores below 15. Beyond this range, where patients exhibit worse liver function, the model struggles to meaningfully predict MELD-scores. Notably, the model never predicts scores higher than 21, despite the cohort containing patients

Table 4.3 Comparison of Regression Model Performance for MELD, LiMAx, and ALBI Scores Using Single **(A)** and Ensembled **(B)** predictions for different feature sets. Numbers in brackets denote the 95%-confidence intervals for each metric.

Feature Set (X)	**MAE**	**R^2**	**RMSE**	**Spearman**	**Pearson**	**NMAE**	**NRMSE**
MELD (A)	2.35 [2.15, 2.60]	0.36 [0.27, 0.45]	3.35 [2.96, 3.77]	0.63 [0.55, 0.69]	0.61 [0.54, 0.67]	0.09 [0.09, 0.10]	0.13 [0.12, 0.15]
MELD (B)	2.35 [2.13, 2.58]	0.37 [0.30, 0.44]	3.33 [2.90, 3.75]	0.64 [0.58, 0.70]	0.61 [0.55, 0.67]	0.09 [0.09, 0.10]	0.13 [0.11, 0.15]
LiMAx (A)	83.36 [74.46, 92.09]	0.31 [0.18, 0.41]	106.86 [94.93, 117.96]	0.59 [0.50, 0.68]	0.57 [0.48, 0.65]	0.14 [0.12, 0.16]	0.18 [0.16, 0.20]
LiMAx (B)	84.29 [75.95, 93.73]	0.31 [0.19, 0.41]	106.61 [95.20, 117.39]	0.58 [0.48, 0.67]	0.56 [0.47, 0.65]	0.14 [0.13, 0.16]	0.18 [0.16, 0.20]
ALBI (A)	0.37 [0.33, 0.41]	0.62 [0.54, 0.69]	0.48 [0.42, 0.53]	0.78 [0.71, 0.83]	0.79 [0.74, 0.84]	0.10 [0.09, 0.11]	0.12 [0.11, 0.14]
ALBI (B)	0.37 [0.33, 0.41]	0.63 [0.54, 0.70]	0.47 [0.42, 0.52]	0.78 [0.72, 0.83]	0.79 [0.73, 0.84]	0.10 [0.09, 0.11]	0.12 [0.11, 0.14]
Feature Set (Y)	**MAE**	**R^2**	**RMSE**	**Spearman**	**Pearson**	**NMAE**	**NRMSE**
MELD (A)	2.62 [2.41, 2.86]	0.26 [0.18, 0.34]	3.61 [3.18, 4.06]	0.56 [0.49, 0.62]	0.51 [0.44, 0.58]	0.10 [0.10, 0.11]	0.14 [0.13, 0.16]
MELD (B)	2.67 [2.44, 2.90]	0.26 [0.18, 0.33]	3.62 [3.18, 4.04]	0.54 [0.46, 0.60]	0.51 [0.43, 0.59]	0.11 [0.10, 0.12]	0.14 [0.13, 0.16]
LiMAx (A)	87.01 [77.90, 95.67]	0.28 [0.14, 0.38]	109.44 [98.64, 120.56]	0.53 [0.42, 0.62]	0.53 [0.42, 0.62]	0.15 [0.13, 0.16]	0.18 [0.16, 0.20]
LiMAx (B)	86.62 [78.39, 95.99]	0.28 [0.16, 0.38]	108.77 [98.73, 118.83]	0.52 [0.41, 0.61]	0.53 [0.44, 0.62]	0.15 [0.13, 0.16]	0.18 [0.16, 0.20]
ALBI (A)	0.42 [0.38, 0.47]	0.52 [0.42, 0.60]	0.54 [0.49, 0.58]	0.71 [0.63, 0.77]	0.72 [0.66, 0.78]	0.11 [0.10, 0.12]	0.14 [0.13, 0.15]
ALBI (B)	0.43 [0.39, 0.48]	0.51 [0.41, 0.58]	0.54 [0.49, 0.59]	0.70 [0.63, 0.76]	0.71 [0.65, 0.77]	0.11 [0.10, 0.12]	0.14 [0.13, 0.16]
Feature Set (Z)	**MAE**	**R^2**	**RMSE**	**Spearman**	**Pearson**	**NMAE**	**NRMSE**
MELD (A)	2.42 [2.21, 2.65]	0.33 [0.26, 0.40]	3.44 [3.01, 3.88]	0.63 [0.57, 0.68]	0.58 [0.52, 0.64]	0.10 [0.09, 0.11]	0.14 [0.12, 0.15]
MELD (B)	2.45 [2.22, 2.69]	0.32 [0.26, 0.38]	3.47 [3.03, 3.91]	0.62 [0.56, 0.67]	0.56 [0.51, 0.62]	0.10 [0.09, 0.11]	0.14 [0.12, 0.16]
LiMAx (A)	86.93 [77.63, 95.90]	0.26 [0.14, 0.36]	110.67 [97.72, 121.47]	0.54 [0.43, 0.64]	0.51 [0.41, 0.61]	0.15 [0.13, 0.16]	0.19 [0.17, 0.20]
LiMAx (B)	86.72 [78.35, 96.42]	0.26 [0.13, 0.36]	110.96 [99.97, 122.35]	0.53 [0.43, 0.62]	0.51 [0.41, 0.60]	0.15 [0.13, 0.16]	0.19 [0.17, 0.21]
ALBI (A)	0.41 [0.37, 0.46]	0.55 [0.46, 0.62]	0.52 [0.47, 0.57]	0.75 [0.69, 0.80]	0.74 [0.68, 0.79]	0.11 [0.10, 0.12]	0.14 [0.12, 0.15]
ALBI (B)	0.41 [0.37, 0.46]	0.56 [0.46, 0.63]	0.52 [0.47, 0.56]	0.75 [0.68, 0.79]	0.75 [0.69, 0.80]	0.11 [0.10, 0.12]	0.13 [0.12, 0.15]

with scores up to 31. This suggests a limitation in the model's ability to generalize to more severe cases of liver dysfunction. This limitation most probably stems from the insufficient representation of higher MELD scores in the whole cohort and subsequently in the training data, leading to a lack of learned patterns for these cases. However, the MELD scores above 20, did not correlate with the other liver function scores, like lower MELD ones did (see Figures 3.7 and 3.5), which could also indicate measurement errors or non-representative samples for the small amount of subjects in this area.
Using only one feature (Y), showed a decrease in performance, with an R^2 of about 0.26 and a slightly higher NMAE of 0.10-0.11, indicating less accuracy. For the combined feature $\mathrm{Vol}^{\mathrm{Liver}} {\cdot} \mathrm{REI}_4^{\mathrm{Liver}}$, the accuracy could be improved upon the relative enhancement index only, achieving an R^2 of 0.32-0.33 and an NMAE similar to the full feature set, suggesting that incorporating liver volume substantially enhances predictive power. There, the best performing models for both of those scenarios were a Multi-Layer Perceptron (MLP), XGBoost and Ridge Regression. However, the models trained on the full feature set still showed the best overall performance in this comparison. For (X), the ensembled models' predictions for the comprehensive feature set slightly improved R^2 and Spearman correlation compared to the single model's predictions, suggesting better predictive consistency, although the differences within the 95%-confidence intervals were negligible. For the small feature sets, performance decreased consistently, when using ensembled predictions.

LiMAx-Score

Regarding the **LiMAx-score** with the comprehensive feature set, the top three performing models were Support Vector Regression (SVR), configured with a C parameter of 100 and gamma set to 'scale'; Random Forest, with a maximum depth of 20 and 500 estimators; and XGBoost, with a maximum depth of 3 and 10 estimators. Regression based on the LiMAx score resulted in an NMAE of about 0.14 and an R^2 of 0.31, reflecting a moderate level of accuracy. The ensemble approach did not enhance the metrics for any of the feature sets, especially when considering the confidence intervals. Using only $\mathrm{REI}_4^{\mathrm{Liver}}$ slightly decreased the R^2, Pearson and Spearman correlation, indicating that this feature alone is less effective. Notably, the combined feature $\mathrm{Vol}^{\mathrm{Liver}} {\cdot}\ \mathrm{REI}_4^{\mathrm{Liver}}$ did not improve any of the metrics at all, suggesting that this combination does not enhance predictive performance for LiMAx as much as for MELD. For feature-set (Y), XGBoost, SVR, and MLP were the top-performing models. These same models also demonstrated the best performance for feature-set (Z), with a slight reordering as MLP and

XGBoost switched positions.
The bottom left plot in figure 4.5 shows a moderate correlation between predictions and ground truth for the model. The Pearson correlation line deviates significantly from the identity line, indicating that while the model captures some trends, it struggles with precise predictions across the full range of scores. This discrepancy suggests that the model may not fully capture the complexity of the LiMAx score, even though the distribution in the dataset is relatively balanced (see subsection 3.1.2).

ALBI-Score

Regression for the **ALBI-score** with the comprehensive feature set demonstrated the highest predictive performance, with an NRMSE of 0.12 and a high R^2 of 0.62-0.63, indicating strong model accuracy and reliability. The top three performing models were a Random Forest, configured with no maximum depth and 500 estimators; XGBoost with a maximum depth of 3 and 10 estimators; and Support Vector Regression (SVR), with a C value of 1 and gamma set to 'scale'. The Pearson and Spearman correlations for predicted and actual values were significantly higher at 0.78 and 0.79, respectively, compared to the other scores. The use of (Y) resulted in a lower R^2 of about 0.51-0.52, with a slightly higher NMAE, showing reduced predictive capability. The $\mathrm{Vol}^{\mathrm{Liver}} \cdot \mathrm{REI}_4^{\mathrm{Liver}}$ feature set achieved an R^2 of 0.55-0.56, which is better than using the relative enhancement index alone but still not quite as effective as the full feature set. For feature set (Y), SVR, MLP and XGBoost were the best performing models. Those also demonstrated the best performance for feature-set (Z), but in the order of MLP, XGBoost and SVR. The bottom right plot in figure 4.5 shows, how the predictions align more closely with the ground truth compared to the other scores, which is in aggreement with the metrics shown in table 4.3. The Pearson correlation line is relatively close to the identity line, suggesting a stronger linear relationship.
However, there are still deviations, particularly at the extremes of the score range, indicating some limitations in the model's predictive capability. One reason for this could be the underrepresentation of patients with a high ALBI-score in the cohort.

Overall, the ensembled models do not significantly enhance the predictive performance within the confidence intervals for regresssion. Consequently, the single-model predictions were chosen for further analysis, as simpler models that achieve similar results are generally preferable due to their ease of interpretation, reduced computational cost, and lower risk of overfitting [62]. Consequently, the plots in

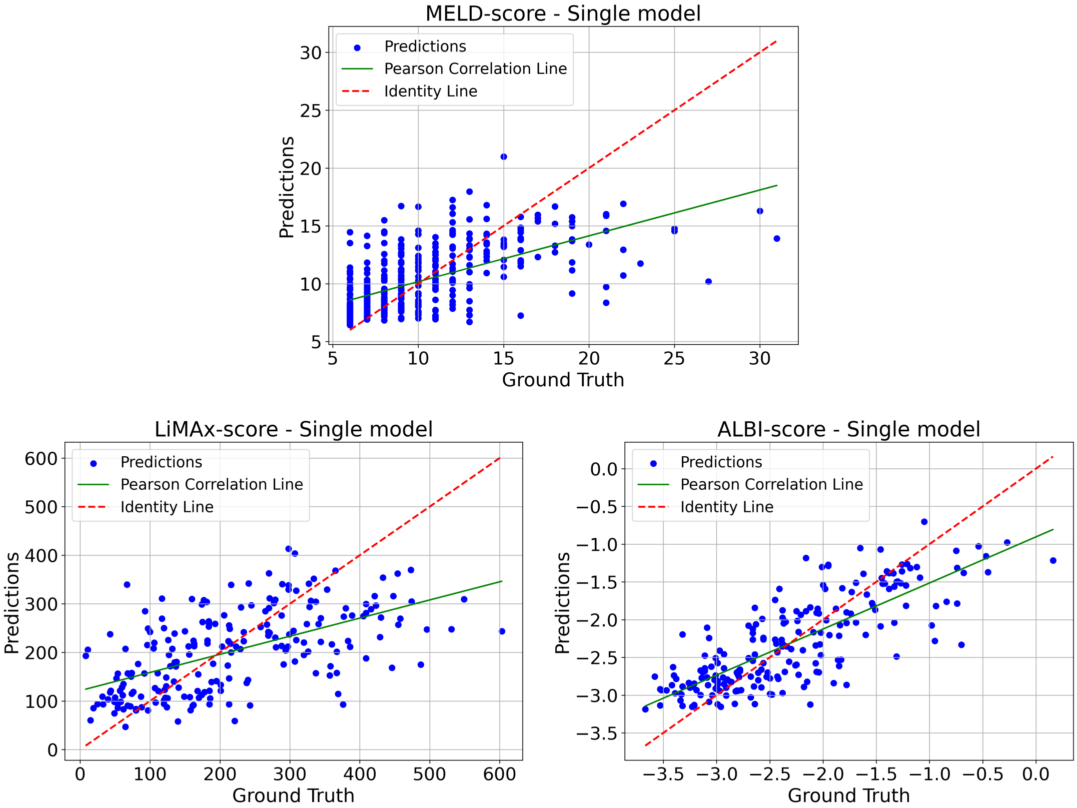


Figure 4.5 Regression results for various liver function scores using the complete feature sets. Predictions are plotted against Ground-Truth values, with the dashed red line representing the identity line, where perfect predictions would lie. The green line illustrates the Pearson correlation between predictions and Ground-Truth. The **top** panel displays the results for the MELD score, while the **bottom right** shows results for the ALBI score. The **bottom left** panel shows the results for the LiMAx score. For all scores, the results shown in the plots were achieved using single model predictions.

figure 4.5 show the results for single model predictions.

The combined feature (Z) improved predictions in comparison with using $\text{REI}_4^{\text{Liver}}$ alone, especially for the MELD score, highlighting the importance of incorporating multiple features for accurate liver function score predictions. However, the comprehensive feature sets consistently outperformed the single-feature models, particularly for the ALBI score, which showed the greatest predictive accuracy overall.

In summary, the regression models demonstrated varying levels of accuracy across the different liver function scores. The MELD model showed limitations, particularly for higher scores, indicating a need for better representation of severe cases in the training data. The LiMAx-score model captured general trends but struggled with precise predictions, while the ALBI-score model showed a stronger correlation, suggesting it may be more reliable for assessing liver function by features derived from Magnetic Resonance Imaging within the tested range.

4.3.2 Classification for Liver Function Prediction

The classification models for liver function prediction were evaluated for MELD-, LiMAx-, and ALBI-scores using both single (A) and ensembled (B) prediction approaches across different feature sets. The notation follows the same form as in the previous subsection, where (X) refers to the distinct feature sets derived by the feature engineering (see table A.1), and (Y) and (Z) for the single features. The results are presented in terms of accuracy and F1-score in Table 4.4. Several ROC curves and confusion matrices for different feature sets and the distinct scores are provided for a comprehensive evaluation, as well.

MELD-Grades

The MELD-score was divided into groups and two distinct classification tasks were performed: binary classification (2 classes) and three-class classification (3 classes). Detailed cutoff value explanations are provided in subsection 3.6.4.

Using the full feature set for the first task, the binary classification achieved the highest accuracy of 0.77 [0.73, 0.81] and F1-score of 0.77 [0.73, 0.81] with ensembled predictions. The ROC curve (top left of Figure 4.6) shows strong performance with an AUC of 0.84. The confusion matrix (top left of Figure 4.7) confirms this with a good overlap of predicted and actual classes. The top 3 models and their hyperparameters were XGBoost with a maximum depth of 3 and 10 estimators; Logistic Regression with a regularization parameter C of 0.1, a maximum of 10.000 iterations and the *liblinear* solver; and Support Vector Classifier (SVC) with a C value of 1 and gamma set to 'scale'.

The three-class classification showed slightly lower performance, with the best accuracy of 0.72 [0.68, 0.77] and F1-score of 0.72 [0.68, 0.77]. The ROC curve (top right of Figure 4.6) shows moderate performance with AUC values of 0.84, 0.76, and 0.78. The confusion matrix (top right of Figure 4.7) reveals more complexity and moderate misclassification, especially for the third class. The top three models were SVC with a regularization parameter of 1 and gamma set to 'scale', Random Forest with no maximum depth and 500 estimators, and XGBoost with a maximum depth of 6 and 1000 estimators. When using only the $\mathrm{REI}_4^{\mathrm{Liver}}$ feature, the performance decreased for both tasks. The binary classification accuracy and AUC dropped to 0.67 [0.62, 0.71] and 0.78 (see Figure A.1 top left), while the three-class classification accuracy fell to 0.56 [0.51, 0.61] with AUCs of 0.78, 0.58

and 0.78 (see Figure A.1 top right).
In comparison to this, the combined feature $\mathrm{Vol}^{\mathrm{Liver}} \cdot \mathrm{REI}_4^{\mathrm{Liver}}$ improved performance, particularly for binary classification, achieving accuracy of 0.71 [0.67, 0.76] and AUC of 0.84 (Figure A.2, top left), comparable to the full feature set. However, for the three-class task, this feature alone underperformed, with an accuracy of 0.54 [0.49, 0.59] and F1-Score of 0.58 [0.55, 0.63]. The ROC curve (Figure A.2, top right) showed moderate performance with AUCs of 0.83, 0.67, 0.75. These results clearly indicate that a more comprehensive feature set significantly enhances predictions for multi-class classification tasks, in this case especially for the second class.

LiMAx-Grades

Classification performance for the LiMAx-grades was generally lower compared to MELD- and ALBI-grades, just like for the regression task. Using the full feature set, the highest accuracy achieved was 0.54 [0.47, 0.61] with a corresponding F1-score of 0.52 [0.45, 0.59]. The ROC curve (bottom left of Figure 4.6) shows lower performance with AUCs of 0.79, 0.69 and 0.70, indicating challenges in precise classification. The confusion matrix (bottom left of Figure 4.7) highlights these challenges with significant misclassification, especially for the second class. The top three models were Logistic Regression with a regularization parameter of 0.1, a maximum of 10,000 iterations, and using the 'liblinear' solver, SVC with a regularization parameter of 1 and gamma set to 'scale', and MLP with hidden layer sizes of 300 and a maximum of 10,000 iterations.
Interestingly, for this cohort, using only $\mathrm{REI}_4^{\mathrm{Liver}}$ showed better performance for Accuracy and F1-score than the combined feature $\mathrm{Vol}^{\mathrm{Liver}} \cdot \mathrm{REI}_4^{\mathrm{Liver}}$. However, the full feature set also did not improve those metrics siginificantly, suggesting that MRI derived features alone may not suffice for accurate LiMAx classification, in general. The ROC-curves for those small feature sets in figures A.1 and A.2, reveal AUCs of 0.79, 0.58, 0.69 and 0.79, 0.59, 0.71 for (Y) and (Z), respectively, showing that the performance does decrease significantly for the second class, but stays about the same for the first and third class in comparison with the comprehensive feature set.

Table 4.4 Comparison of Classification Model Performance for MELD, LiMAx, and ALBI Grades Using Single **(A)** and Ensembled **(B)** predictions for different feature sets. Numbers in brackets denote the 95%-confidence intervals for each metric.

Feature Set (X)	**Accuracy**	**F1-Score**
MELD 2 classes (A)	0.75 [0.71, 0.80]	0.76 [0.72, 0.80]
MELD 2 classes (B)	0.77 [0.73, 0.81]	0.77 [0.73, 0.81]
MELD 3 classes (A)	0.69 [0.65, 0.74]	0.71 [0.67, 0.75]
MELD 3 classes (B)	0.72 [0.68, 0.77]	0.72 [0.68, 0.77]
LiMAx (A)	0.54 [0.47, 0.61]	0.52 [0.45, 0.59]
LiMAx (B)	0.52 [0.45, 0.59]	0.51 [0.44, 0.58]
ALBI (A)	0.68 [0.61, 0.74]	0.67 [0.61, 0.74]
ALBI (B)	0.68 [0.61, 0.73]	0.67 [0.62, 0.74]
Feature Set (Y)	**Accuracy**	**F1-Score**
MELD 2 classes (A)	0.66 [0.61, 0.71]	0.67 [0.62, 0.71]
MELD 2 classes (B)	0.67 [0.62, 0.71]	0.67 [0.63, 0.72]
MELD 3 classes (A)	0.55 [0.50, 0.60]	0.60 [0.55, 0.64]
MELD 3 classes (B)	0.56 [0.51, 0.61]	0.60 [0.55, 0.64]
LiMAx (A)	0.53 [0.46, 0.60]	0.51 [0.44, 0.59]
LiMAx (B)	0.44 [0.37, 0.50]	0.40 [0.34, 0.48]
ALBI (A)	0.54 [0.47, 0.60]	0.51 [0.44, 0.59]
ALBI (B)	0.59 [0.53, 0.66]	0.57 [0.50, 0.64]
Feature Set (Z)	**Accuracy**	**F1-Score**
MELD 2 classes (A)	0.71 [0.67, 0.76]	0.72 [0.68, 0.76]
MELD 2 classes (B)	0.71 [0.67, 0.76]	0.72 [0.68, 0.76]
MELD 3 classes (A)	0.53 [0.49, 0.58]	0.58 [0.55, 0.63]
MELD 3 classes (B)	0.54 [0.49, 0.59]	0.58 [0.54, 0.63]
LiMAx (A)	0.50 [0.44, 0.57]	0.47 [0.40, 0.55]
LiMAx (B)	0.49 [0.43, 0.56]	0.45 [0.39, 0.54]
ALBI (A)	0.57 [0.51, 0.63]	0.56 [0.49, 0.63]
ALBI (B)	0.57 [0.50, 0.63]	0.55 [0.49, 0.62]

ALBI-Grades

The ALBI-grade classification showed moderate performance with the full feature set, achieving an accuracy of 0.68 [0.61, 0.74] and an F1-score of 0.67 [0.61, 0.74]. The ROC curves (bottom right of Figure 4.6) indicate strong performance, with AUC values of 0.88, 0.73, and 0.91. The confusion matrix (bottom right of Figure 4.7) demonstrates strong performance, with moderate misclassifications for class 2 and 3. The top three models were Random Forest with no maximum depth and 100 estimators, XGBoost with a maximum depth of 9 and 10 estimators, and SVC with a regularization parameter of 1 and gamma set to 'scale'.

With the small feature sets, performance decreased significantly, regarding accuracy and F1-score. For (Y), ensembled predictions increased those metrics considerably, compared to the single-model predictions. This observation can not be made with the full feature set (X) and the small feature set (Z). In this scenario, the single-model predictions perform as good as the ensembled models, considering the 95%-confidence intervals. Taking the ROC-curves into account, one can see, AUC values of 0.86, 0.66, 0.88 and 0.88, 0.66, 0.87 for (Y) and (Z), respectively. This highlights the increased challenges for the models to correctly predict ALBI-grade 2, when using too small feature sets.

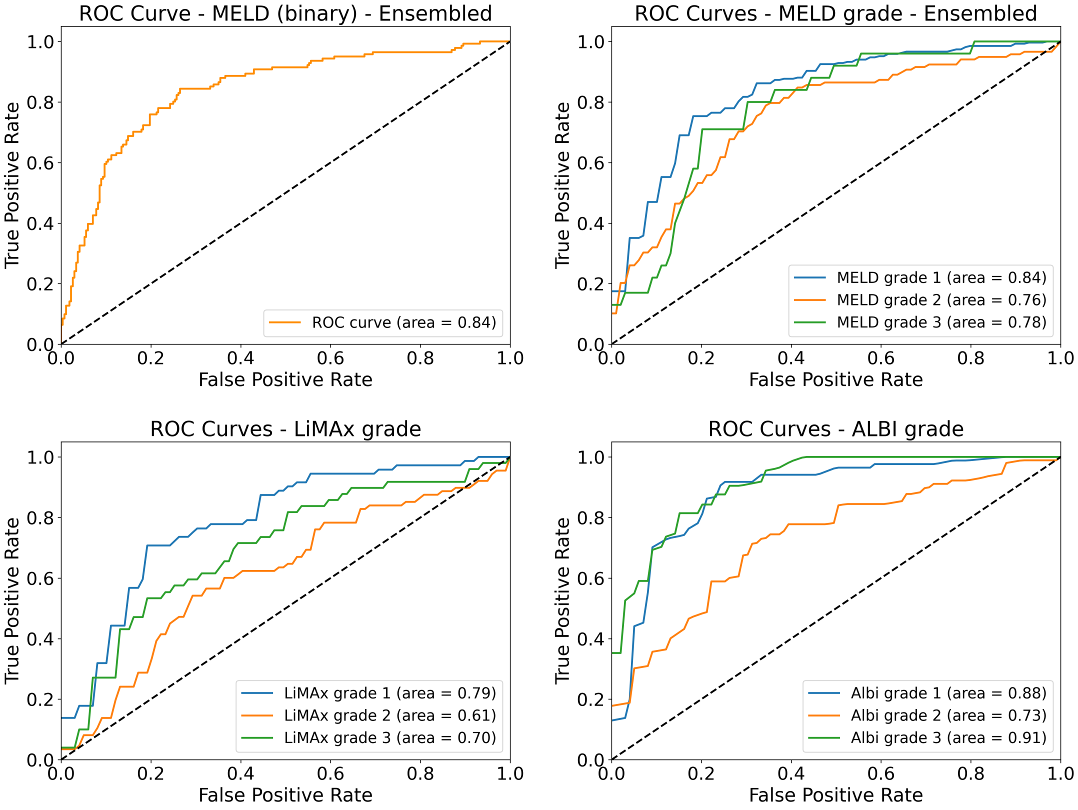


Figure 4.6 Receiver Operating Characteristic (ROC) Curves and Area Under Curve (AUC) for various liver function score classifications using the complete feature sets. The dashed line represents the performance of a random classifier for balanced datasets. The **top left** panel displays results for binary classification of the MELD score, while the **top right** plot illustrates three-class classification results for this score. The **bottom left** panel shows results for LiMAx grades, and the **bottom right** panel presents results for ALBI grades. MELD results were achieved through ensembled predictions, whereas LiMAx and ALBI results were generated using single-model predictions.

In summary, while the MELD score classification showed strong performance, particularly for binary classification, the ALBI grade classification demonstrated the most promising results for distinguishing between normal, medium, and poor liver function. Using the full feature set, ALBI classification achieved moderate accuracy but excellent discrimination ability, with AUC values of 0.88, 0.73, and 0.91 for the three classes. LiMAx grade classification showed the lowest performance across all feature sets, suggesting that MRI-derived features alone may not be sufficient for accurate LiMAx classification. Regarding the MELD score, ensembled predictions improved the performance for binary-classification as well as for classification with three grades, where the LiMAx classification performance consistently decreased with ensembled predictions. For ALBI, the ensembled performance was better only for feature set (Y) and stayed the same for the other feature sets.

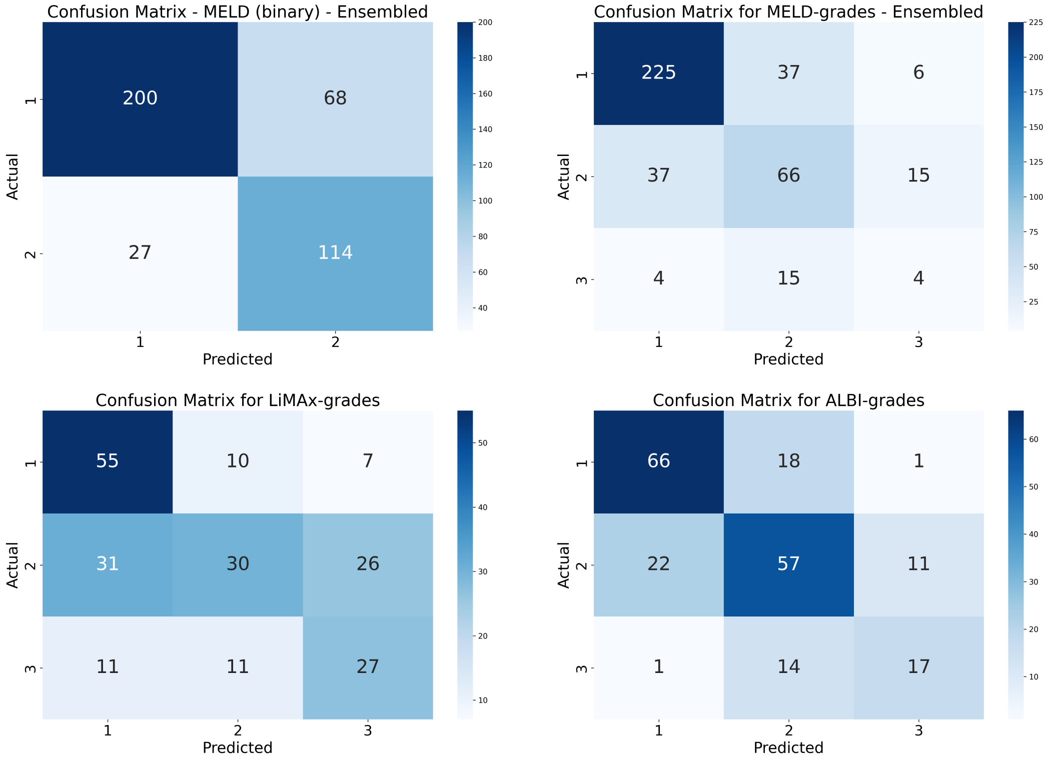


Figure 4.7 Confusion matrices for liver function score classifications using complete feature sets. The **top left** panel shows binary classification results for the MELD score and the **top right** panel displays three-class classification results for this score, both achieved through ensembled predictions. The **bottom left** panel presents results for LiMAx grades, and the **bottom right** panel shows results for ALBI grades, both using single-model predictions.

Overall, the results indicate that comprehensive feature sets generally improve classification performance, especially for multi-class tasks, with ALBI score classification showing the most potential for accurately differentiating between various levels of liver function with MRI-derived features. This is also in aggreement with the regression results from the previous subsection.

Based on the comprehensive analysis of regression and classification models for liver function estimation using MRI-derived features, the ALBI score and grade consistently demonstrated the strongest predictive performance across both regression and classification tasks. The MELD score showed moderate predictive capability for three classes, and very good results for binary classification, while the LiMAx score proved challenging to predict accurately using MRI features alone in both, regression and classification tasks. These findings suggest that automated MRI-derived features, especially when using a comprehensive feature set, have significant potential for non-invasive liver function assessment, with the ALBI score emerging as the most promising target for future clinical applications and research.

Chapter 5

Summary

This thesis presents a novel technique for liver function estimation based solely on MRI imaging features, utilizing machine learning and deep learning approaches. The work is structured in the following way:

Chapter 1 introduces the importance of accurate liver function assessment and the potential of AI-driven methods in combination with medical imaging to enhance this process.

Chapter 2 provides essential background information on liver anatomy, key biomarkers of liver function, liver function scores, principles of MRI, and fundamentals of artificial intelligence including various machine learning models, CNNs and transformer networks.

Chapter 3 provides a comprehensive overview of the study's datasets and methodologies. It describes the MRI datasets used for liver segmentation (78 subjects) and liver function estimation (458 subjects), along with the preprocessing techniques applied to the MRI images. The chapter also outlines the evaluation metrics employed to assess the performance of coregistration, segmentation, and liver function estimation processes. Furthermore, it offers explanations of the various architectures used for image segmentation, the feature extraction methodology, and the procedures for model training. Additionally, the chapter presents an analysis of the cross-correlation between different liver function scores, revealing that the ALBI and MELD scores exhibit the strongest correlation, while the LiMAx and MELD scores show the weakest correlation. A detailed explanation of the meticolous refinement technique for the segmentations can also be found in this chapter.

Chapter 4 offers a comprehensive presentation and analysis of the thesis findings. It examines the coregistration performance of MRI volumes across various phases

and thereby addressing the specific challenges associated with breath-holding imaging techniques. The chapter also evaluates the segmentation performance of different architectures for seven liver-related structures. Furthermore, it presents and discusses the outcomes of liver function estimation using both regression and classification approaches, providing a thorough assessment of the study's key results.

The study utilized three distinct deep learning architectures to comprehensively evaluate segmentations of the liver parenchyma, portal veins, hepatic veins, lesions, ascites, and the thoracic and abdominal aorta. In this comparative analysis, the standard nnU-Net generally outperformed both the residual encoder nnU-Net and the Swin UNETR transformer network across multiple metrics, including DSC, IOU, PPV, TPR, LFPR, LTPR, and Volume Difference for most of the segmentation classes.

Due to coregistration challenges and imperfect segmentations of fine structures such as vessels, the automatically generated labels from the nnU-Net underwent a unique post-processing pipeline, as detailed in subsection 3.6.1. These refined labels were subsequently used to extract meaningful features from the MRIs for liver function estimation in later stages of the pipeline, as elaborated in subsection 3.6.2.

Before training the liver function estimation models, the feature sets were processed through a feature-engineering pipeline to retain only the most relevant features regarding each liver function score. Given the imperfect correlations between scores (see subsection 3.1.2), this resulted in three distinct comprehensive feature sets for the three liver function scores, as illustrated in table A.1. Previous studies have correlated features such as the relative enhancement index of the liver parenchyma, or its combination with liver volume, to various scores. However, these studies typically limited their measurements to a small number of liver ROIs, potentially losing valuable information due to the reported possibility of varying liver health and function across different regions of the organ [180, 181, 182].

In contrast, this thesis employed comprehensive measurements of the entire tissues under consideration with automated segmentations of the whole structures. Consequently, for a fair comparison of the comprehensive approach with the single features, these individual features were also tested against the larger feature sets under the same conditions in this thesis, to investigate whether the predictive performance could be boosted by a larger feature set than from the other studies reported.

The research evaluated both regression and classification approaches on those different feature sets using machine learning models such as Random Forest, XGBoost, Support Vector Machines, Multi-layer perceptrons, logistic regression, Ridge- and Lasso-regression, as well as Elastic Net in terms of MAE, R^2, RMSE, Spearman and Pearson correlation, F1-Score, Accuracy, AUCs, Confusion Matrices and normalized metrics as NMAE and NRMSE. Thorough hyperparamer tuning was performed to find the best possible configurations for each score and feature set. The performance of ensembled predictions vs. the single-model predictions was evaluated, too.

The results demonstrated that using a comprehensive set of features derived from MRI images continuously improved liver function prediction compared to using single features, except for the LiMAx-score, which showed the worst results overall, no matter which feature set was used. This suggests that the use of only MRI derived features is probably not sufficient to build classification models for the LiMAx-score. The results for the ALBI score cohort consistently showed the highest predictive accuracy among the liver function scores evaluated, for the regression and the classification tasks.

Keeping current research in mind, which states that ALBI can show comparable or even superior prognostic performance, compared to more traditional liver function scores as MELD- and Child-Pugh-score [26, 27], the results of this study are very promising regarding non-invasive liver function assessment.

In conclusion, this study is able to demonstrate that fully automated non-invasive liver function prognosis - especially for the ALBI score - based on MRI data is indeed possible, making it a fairly reliable, accurate and objectively reproducible method that could be implemented in routine clinical practice with little effort. Furthermore, this thesis contributes to the growing body of knowledge in AI-driven medical imaging and offers practical solutions for improving the accuracy and efficiency of automated liver function assessments using MRI data, particularly highlighting the potential of comprehensive feature sets to enhance predictive accuracy, except in the case of the LiMAx-score where MRI-derived features alone may be insufficient.

Bibliography

[1] W.Y Lau. *Applied Anatomy in Liver Resection and Liver Transplantation.* eng. 2021.

[2] E.M. Gadzijev, S. Bengmark, and D. Ravnik. *Atlas of Applied Internal Liver Anatomy.* Springer medicine. Springer Vienna, 1996. ISBN: 9783211827932. URL: https://books.google.de/books?id=5OBqAAAAMAAJ.

[3] J. Braun and D. Müller-Wieland. *Basislehrbuch Innere Medizin: kompakt-greifbar-verständlich.* Elsevier Health Sciences, 2017. ISBN: 9783437180118. URL: https://books.google.de/books?id=dFpgDwAAQBAJ.

[4] Michael Schünke et al. *PROMETHEUS Innere Organe: LernAtlas der Anatomie.* Jan. 2018. ISBN: 9783132420878. DOI: 10.1055/b-006-149645.

[5] American Cancer Society. *Embolization Therapy for Liver Cancer.* https://www.cancer.org/cancer/types/liver-cancer/treating/embolization-therapy.html. 2019.

[6] NJ Shah, A Royer, and S John. *Acute Liver Failure.* Updated 2023 Apr 7. Treasure Island (FL): StatPearls Publishing, 2023. URL: https://www.ncbi.nlm.nih.gov/books/NBK482374/.

[7] Guilherme Mariante-Neto et al. "Impact of creatinine values on MELD scores in male and female candidates for liver transplantation." eng. In: *Ann Hepatol* 12.3 (2013), pp. 434–439. ISSN: 1665-2681 (Print); 1665-2681 (Linking).

[8] Florencia I. Aiello et al. "Model for End-stage Liver Disease (MELD) score and liver transplant: benefits and concerns". In: *AME Medical Journal* 2.11 (2017). ISSN: 2520-0518. URL: https://amj.amegroups.org/article/view/4162.

[9] Marcus A. Rothschild, Murray Oratz, and Sidney S. Schreiber. “Serum albumin”. In: *Hepatology* 8.2 (1988), pp. 385–401. DOI: `https://doi.org/10.1002/hep.1840080234`.

[10] Jens van de Wouw and Jaap A Joles. “Albumin is an interface between blood plasma and cell membrane, and not just a sponge.” eng. In: *Clin Kidney J* 15.4 (2022), pp. 624–634. ISSN: 2048-8505 (Print); 2048-8513 (Electronic); 2048-8505 (Linking). DOI: `10.1093/ckj/sfab194`.

[11] Libor Vítek and J Donald Ostrow. “Bilirubin chemistry and metabolism; harmful and protective aspects.” eng. In: *Curr Pharm Des* 15.25 (2009), pp. 2869–2883. ISSN: 1873-4286 (Electronic); 1381-6128 (Linking). DOI: `10.2174/138161209789058237`.

[12] Sadhana Kumbhar, Manish Musale, and Anas Jamsa. “Bilirubin metabolism: delving into the cellular and molecular mechanisms to predict complications”. In: *The Egyptian Journal of Internal Medicine* 36.1 (2024), p. 34. DOI: `10.1186/s43162-024-00298-5`.

[13] Marlies Ostermann, Kianoush Kashani, and Lui G. Forni. “The two sides of creatinine: both as bad as each other?” In: *Journal of Thoracic Disease* 8.7 (2016). ISSN: 2077-6624. URL: `https://jtd.amegroups.org/article/view/7756`.

[14] Patrick S. Kamath et al. “A model to predict survival in patients with end-stage liver disease”. In: *Hepatology* 33.2 (2001), pp. 464–470. DOI: `https://doi.org/10.1053/jhep.2001.22172`.

[15] R J Porte et al. “The International Normalized Ratio (INR) in the MELD score: problems and solutions.” eng. In: *Am J Transplant* 10.6 (2010), pp. 1349–1353. DOI: `10.1111/j.1600-6143.2010.03064.x`.

[16] C G Child and J G Turcotte. “Surgery and portal hypertension.” eng. In: *Major Probl Clin Surg* 1 (1964), pp. 1–85. ISSN: 0025-1062 (Print); 0025-1062 (Linking).

[17] Martin Stockmann et al. “The LiMAx test: a new liver function test for predicting postoperative outcome in liver surgery.” eng. In: *HPB (Oxford)* 12.2 (2010), pp. 139–146. ISSN: 1477-2574 (Electronic); 1365-182X (Print); 1365-182X (Linking). DOI: `10.1111/j.1477-2574.2009.00151.x`.

[18] Philip J Johnson et al. “Assessment of liver function in patients with hepatocellular carcinoma: a new evidence-based approach-the ALBI grade.” eng. In: *J Clin Oncol* 33.6 (2015), pp. 550–558. ISSN: 1527-7755 (Electronic); 0732-183X (Print); 0732-183X (Linking). DOI: `10.1200/JCO.2014.57.9151`.

[19] A. R. Cooke, D. D. Harrison, and A. P. Skyring. “Use of indocyanine green as a test of liver function”. In: *The American Journal of Digestive Diseases* 8.3 (1963), pp. 244–250. DOI: 10.1007/BF02232323.

[20] Alexander Polyak, Alexander Kuo, and Vinay Sundaram. “Evolution of liver transplant organ allocation policy: Current limitations and future directions.” eng. In: *World J Hepatol* 13.8 (2021), pp. 830–839. ISSN: 1948-5182 (Print); 1948-5182 (Electronic). DOI: 10.4254/wjh.v13.i8.830.

[21] Patrick S. Kamath and W. Ray Kim. “The model for end-stage liver disease (MELD)”. In: *Hepatology* 45.3 (2007), pp. 797–805. DOI: https://doi.org/10.1002/hep.21563.

[22] Gabriela Berlakovich. “Indikationen und Kontraindikationen zur Lebertransplantation in Bezug auf aktuelle Leitlinien”. In: *Journal für Gastroenterologische und Hepatologische Erkrankungen* 20.2 (2022), pp. 38–44. DOI: 10.1007/s41971-022-00125-0.

[23] Martin Stockmann et al. “Prediction of Postoperative Outcome After Hepatectomy With a New Bedside Test for Maximal Liver Function Capacity”. In: *Annals of Surgery* 250.1 (2009). URL: https://journals.lww.com/annalsofsurgery/fulltext/2009/07000/prediction_of_postoperative_outcome_after.19.aspx.

[24] Maximilian Jara et al. “Prognostic value of enzymatic liver function for the estimation of short-term survival of liver transplant candidates: a prospective study with the LiMAx test”. In: *Transplant International* 28.1 (2015), pp. 52–58. DOI: https://doi.org/10.1111/tri.12441.

[25] Hidenori Toyoda and Philip J Johnson. “The ALBI score: From liver function in patients with HCC to a general measure of liver function.” eng. In: *JHEP Rep* 4.10 (2022), p. 100557. ISSN: 2589-5559 (Electronic); 2589-5559 (Linking). DOI: 10.1016/j.jhepr.2022.100557.

[26] Maria Fragaki et al. “Comparative evaluation of ALBI, MELD, and Child-Pugh scores in prognosis of cirrhosis: is ALBI the new alternative?” eng. In: *Ann Gastroenterol* 32.6 (2019), pp. 626–632. ISSN: 1108-7471 (Print); 1792-7463 (Electronic); 1108-7471 (Linking). DOI: 10.20524/aog.2019.0417.

[27] Shi-Xue Xu et al. “Role of albumin-bilirubin score in non-malignant liver disease.” eng. In: *World J Gastroenterol* 30.9 (2024), pp. 999–1004. ISSN: 2219-2840 (Electronic); 1007-9327 (Print); 1007-9327 (Linking). DOI: 10.3748/wjg.v30.i9.999.

[28] Joseph P. Hornak. *The Basics of MRI.* https://www.cis.rit.edu/htbooks/mri/inside.html. Last checked: April 16, 2024.

[29] Felix Lugauer and Jens Wetzl. “Magnetic Resonance Imaging.” eng. In: (2018), pp. 91–118.

[30] Siemens Healthcare GmbH. *Magnets, spins and resonances – an introduction to the basics of magnetic resonance.* SIEMENS AG, 2015.

[31] Siemens Healthineers. *Magnets, spins and resonances – an introduction to the basics of magnetic resonance.* SIEMENS AG, 2023.

[32] Yves Gossuin et al. “Physics of magnetic resonance imaging: from spin to pixel”. In: *Journal of Physics D: Applied Physics* 43.21 (2010), p. 213001. DOI: 10.1088/0022-3727/43/21/213001.

[33] Gregory McCarthy Scott A. Huettel Allen W. Song. *Functional Magnetic Resonance Imaging.* Sinauer Associates, Inc, 2004.

[34] Siegfried Stapf and Song I. Han. *NMR Imaging in Chemical Engineering.* English (US). Wiley-VCH, Apr. 2006. ISBN: 352731234X. DOI: 10.1002/3527607560.

[35] E. L. Hahn. “Free nuclear induction”. In: *Physics Today* 6.11 (Nov. 1953), pp. 4–9. ISSN: 0031-9228. DOI: 10.1063/1.3061075.

[36] E. L. Hahn. “Spin Echoes”. In: *Phys. Rev.* 80 (4 1950), pp. 580–594. DOI: 10.1103/PhysRev.80.580. URL: https://link.aps.org/doi/10.1103/PhysRev.80.580.

[37] P. C. Lauterbur. “Image Formation by Induced Local Interactions: Examples Employing Nuclear Magnetic Resonance”. In: *Nature* 242.5394 (1973), pp. 190–191. DOI: 10.1038/242190a0.

[38] Brian J. Soher, Brian M. Dale, and Elmar M. Merkle. “A Review of MR Physics: 3T versus 1.5T”. In: *Magnetic Resonance Imaging Clinics of North America* 15.3 (2007). Body MR Imaging: 1.5T versus 3T, pp. 277–290. ISSN: 1064-9689. DOI: https://doi.org/10.1016/j.mric.2007.06.002.

[39] Sebastian T. Schindera et al. “Abdominal Magnetic Resonance Imaging at 3.0 T: What Is the Ultimate Gain in Signal-to-Noise Ratio?” In: *Academic Radiology* 13.10 (2006), pp. 1236–1243. ISSN: 1076-6332. DOI: https://doi.org/10.1016/j.acra.2006.06.018.

[40] Cedric M. J. de Bazelaire et al. “MR Imaging Relaxation Times of Abdominal and Pelvic Tissues Measured in Vivo at 3.0 T: Preliminary Results”. In: *Radiology* 230.3 (2004). PMID: 14990831, pp. 652–659. DOI: 10.1148/radiol.2303021331. eprint: https://doi.org/10.1148/radiol.2303021331. URL: https://doi.org/10.1148/radiol.2303021331.

[41] Greg J. Stanisz et al. “T1, T2 relaxation and magnetization transfer in tissue at 3T”. In: *Magnetic Resonance in Medicine* 54.3 (2005), pp. 507–512. DOI: `https://doi.org/10.1002/mrm.20605`.

[42] Jorge Zavala Bojorquez et al. “What are normal relaxation times of tissues at 3 T?” eng. In: *Magn Reson Imaging* 35 (2017), pp. 69–80. ISSN: 1873-5894 (Electronic); 0730-725X (Linking). DOI: `10.1016/j.mri.2016.08.021`.

[43] Allen Elster. *Questions and Answers in MRI.* `https://mri-q.com/why-is-t1--t2.html`. Last checked: May 31, 2024. URL: `https://mri-q.com/why-is-t1--t2.html`.

[44] Ahmed Ba-Ssalamah et al. “Clinical value of MRI liver-specific contrast agents: a tailored examination for a confident non-invasive diagnosis of focal liver lesions.” eng. In: *Eur Radiol* 19.2 (2009), pp. 342–357. ISSN: 1432-1084 (Electronic); 0938-7994 (Linking). DOI: `10.1007/s00330-008-1172-x`.

[45] Mark D Goodwin et al. “Diagnostic challenges and pitfalls in MR imaging with hepatocyte-specific contrast agents.” eng. In: *Radiographics* 31.6 (2011), pp. 1547–1568. ISSN: 1527-1323 (Electronic); 0271-5333 (Linking). DOI: `10.1148/rg.316115528`.

[46] D H Lee. “Mechanisms of contrast enhancement in magnetic resonance imaging.” eng. In: *Can Assoc Radiol J* 42.1 (1991), pp. 6–12. ISSN: 0846-5371 (Print); 0846-5371 (Linking).

[47] Cosmin-Nicolae Caraiani et al. “Description of focal liver lesions with Gd-EOB-DTPA enhanced MRI.” eng. In: *Clujul Med* 88.4 (2015), pp. 438–448. ISSN: 1222-2119 (Print); 2066-8872 (Electronic); 1222-2119 (Linking). DOI: `10.15386/cjmed-414`.

[48] N M Rofsky et al. “Abdominal MR imaging with a volumetric interpolated breath-hold examination.” eng. In: *Radiology* 212.3 (1999), pp. 876–884. ISSN: 0033-8419 (Print); 0033-8419 (Linking). DOI: `10.1148/radiology.212.3.r99se34876`.

[49] T J Vogl et al. “Liver tumors: comparison of MR imaging with Gd-EOB-DTPA and Gd-DTPA.” eng. In: *Radiology* 200.1 (1996), pp. 59–67. ISSN: 0033-8419 (Print); 0033-8419 (Linking). DOI: `10.1148/radiology.200.1.8657946`.

[50] Benjamin Kaltenbach et al. “Free-breathing dynamic liver examination using a radial 3D T1-weighted gradient echo sequence with moderate undersampling for patients with limited breath-holding capacity”. In: *European Journal of Radiology* 86 (2017), pp. 26–32. ISSN: 0720-048X. DOI: `https://doi.org/10.1016/j.ejrad.2016.11.003`.

[51] Mohammed Yusuf Ansari et al. “Practical utility of liver segmentation methods in clinical surgeries and interventions.” eng. In: *BMC Med Imaging* 22.1 (2022), p. 97. ISSN: 1471-2342 (Electronic); 1471-2342 (Linking). DOI: 10.1186/s12880-022-00825-2.

[52] Syed M.S. Reza et al. “Deep Learning for Automated Liver Segmentation to Aid in the Study of Infectious Diseases in Nonhuman Primates”. In: *Academic Radiology* 28 (2021). Special Issue: Gastrointestinal Radiology, S37–S44. ISSN: 1076-6332. DOI: https://doi.org/10.1016/j.acra.2020.08.023.

[53] Akshat Gotra et al. “Liver segmentation: indications, techniques and future directions”. In: *Insights into Imaging* 8.4 (2017), pp. 377–392. DOI: 10.1007/s13244-017-0558-1.

[54] Hope Reese. *Understanding the differences between AI, machine learning, and deep learning.* Tech. rep. Tech Republic, Feb. 2017.

[55] Charu C. Aggarwal. *Neural Networks and Deep Learning. A Textbook.* Cham: Springer, 2018, p. 497. ISBN: 978-3-319-94462-3. DOI: 10.1007/978-3-319-94463-0.

[56] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning.* http://www.deeplearningbook.org. MIT Press, 2016.

[57] Stuart J. Russel and Peter Norvig. *Künstliche Intelligenz. Ein moderner Ansatz.* Vol. 3. Pearson Studium, 2004.

[58] Leo Breiman. “Random Forests”. In: *Machine Learning* 45.1 (2001), pp. 5–32. DOI: 10.1023/A:1010933404324.

[59] Yoav Freund and Robert E. Schapire. “A desicion-theoretic generalization of on-line learning and an application to boosting”. In: *Computational Learning Theory.* Ed. by Paul Vitányi. Berlin, Heidelberg: Springer Berlin Heidelberg, 1995, pp. 23–37. ISBN: 978-3-540-49195-8.

[60] Tianqi Chen and Carlos Guestrin. “XGBoost: A Scalable Tree Boosting System”. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* KDD ’16. ACM, Aug. 2016. DOI: 10.1145/2939672.2939785.

[61] Douglas C. Montgomery, Elizabeth A. Peck, and G. Geoffrey Vining. *Introduction to Linear Regression Analysis.* 5th. John Wiley & Sons, 2012.

[62] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* Vol. 2. Springer Series in Statistics, 2009.

[63] Yingjie Tian and Yuqi Zhang. “A comprehensive survey on regularization strategies in machine learning”. In: *Information Fusion* 80 (2022), pp. 146–166. ISSN: 1566-2535. DOI: https://doi.org/10.1016/j.inffus.2021.11.005.

[64] V. N. Vapnik and A. Ya. Chervonenkis. “On the Uniform Convergence of Relative Frequencies of Events to Their Probabilities”. In: *Theory of Probability & Its Applications* 16.2 (1971), pp. 264–280. DOI: 10.1137/1116025.

[65] Xinhua Zhang. “Support Vector Machines”. In: *Encyclopedia of Machine Learning*. Ed. by Claude Sammut and Geoffrey I. Webb. Boston, MA: Springer US, 2010, pp. 941–946. ISBN: 978-0-387-30164-8. DOI: 10.1007/978-0-387-30164-8_804.

[66] Abiodun Ikotun et al. “K-means Clustering Algorithms: A Comprehensive Review, Variants Analysis, and Advances in the Era of Big Data”. In: *Information Sciences* 622 (Dec. 2022). DOI: 10.1016/j.ins.2022.11.139.

[67] Pranav Shetty and Suraj Singh. “Hierarchical Clustering: A Survey”. In: *International Journal of Applied Research* 7 (Apr. 2021), pp. 178–181. DOI: 10.22271/allresearch.2021.v7.i4c.8484.

[68] Ian T Jolliffe and Jorge Cadima. “Principal component analysis: a review and recent developments.” eng. In: *Philos Trans A Math Phys Eng Sci* 374.2065 (2016), p. 20150202. ISSN: 1471-2962 (Electronic); 1364-503X (Print); 1364-503X (Linking). DOI: 10.1098/rsta.2015.0202.

[69] Ron Kohavi. “A study of cross-validation and bootstrap for accuracy estimation and model selection”. In: *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 2*. IJCAI’95. Montreal, Quebec, Canada: Morgan Kaufmann Publishers Inc., 1995, pp. 1137–1143. ISBN: 1558603638.

[70] Sylvain Arlot and Alain Celisse. “A survey of cross-validation procedures for model selection”. In: *Statistics Surveys* 4.none (Jan. 2010), pp. 40–79. DOI: 10.1214/09-SS054.

[71] Dinggang Shen, Guorong Wu, and Heung-Il Suk. “Deep Learning in Medical Image Analysis.” eng. In: *Annu Rev Biomed Eng* 19 (2017), pp. 221–248. DOI: 10.1146/annurev-bioeng-071516-044442.

[72] G. Varoquaux et al. “Machine Learning for Medical Imaging: Methodological Failures and Recommendations for the Future”. In: *npj Digital Medicine* 5 (2022), p. 92. DOI: 10.1038/s41746-022-00592-y.

[73] Heang-Ping Chan et al. “Deep Learning in Medical Image Analysis.” eng. In: *Adv Exp Med Biol* 1213 (2020), pp. 3–21. ISSN: 0065-2598 (Print); 2214-8019 (Electronic); 0065-2598 (Linking). DOI: 10.1007/978-3-030-33128-3{_}1.

[74] Javaria Amin et al. “Brain tumor detection and classification using machine learning: a comprehensive survey”. In: *Complex & Intelligent Systems* 8.4 (2022), pp. 3161–3183. DOI: 10.1007/s40747-021-00563-y.

[75] Leo Breiman. “Bagging predictors”. In: *Machine Learning* 24.2 (1996), pp. 123–140. DOI: 10.1007/BF00058655.

[76] Tin Kam Ho. “The random subspace method for constructing decision forests”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20.8 (1998), pp. 832–844. DOI: 10.1109/34.709601.

[77] Andy Liaw and Matthew Wiener. “Classification and Regression by RandomForest”. In: *Forest* 23 (Nov. 2001).

[78] Nathalie Japkowicz and Shaju Stephen. “The class imbalance problem: A systematic study”. In: *Intelligent Data Analysis* 6.5 (2002), pp. 429–449. DOI: 10.3233/IDA-2002-6504.

[79] Chao Chen and Leo Breiman. “Using Random Forest to Learn Imbalanced Data”. In: *University of California, Berkeley* (Jan. 2004).

[80] Mohammed Khalilia, Sounak Chakraborty, and Mihail Popescu. “Predicting disease risks from highly imbalanced data using random forest”. In: *BMC Medical Informatics and Decision Making* 11.1 (2011), p. 51. DOI: 10.1186/1472-6947-11-51.

[81] Gérard Biau. “Analysis of a Random Forests Model”. In: *Journal of Machine Learning Research* 13.38 (2012), pp. 1063–1095. URL: http://jmlr.org/papers/v13/biau12a.html.

[82] Jerome H. Friedman. “Greedy function approximation: A gradient boosting machine.” In: *The Annals of Statistics* 29.5 (2001), pp. 1189 –1232. DOI: 10.1214/aos/1013203451.

[83] John Mingers. “An Empirical Comparison of Pruning Methods for Decision Tree Induction”. In: *Machine Learning* 4.2 (1989), pp. 227–243. DOI: 10.1023/A:1022604100933.

[84] Manuel Fernández-Delgado et al. “Do we Need Hundreds of Classifiers to Solve Real World Classification Problems?” In: *Journal of Machine Learning Research* 15.90 (2014), pp. 3133–3181.

[85] Candice Bentéjac, Anna Csörgő, and Gonzalo Martínez-Muñoz. “A comparative analysis of gradient boosting algorithms”. In: *Artificial Intelligence Review* 54.3 (Aug. 2020), pp. 1937–1967. ISSN: 1573-7462. DOI: 10.1007/s10462-020-09896-5.

[86] Wandong Hong et al. “A Comparison of XGBoost, Random Forest, and Nomograph for the Prediction of Disease Severity in Patients With COVID-19 Pneumonia: Implications of Cytokine and Immune Cell Profile”. In: *Frontiers in Cellular and Infection Microbiology* 12 (2022). ISSN: 2235-2988. DOI: 10.3389/fcimb.2022.819267.

[87] Wenbing Chang et al. “A Machine-Learning-Based Prediction Method for Hypertension Outcomes Based on Medical Data.” eng. In: *Diagnostics (Basel)* 9.4 (2019). ISSN: 2075-4418 (Print); 2075-4418 (Electronic); 2075-4418 (Linking). DOI: 10.3390/diagnostics9040178.

[88] Zheng-Gang Fang et al. “Application of a data-driven XGBoost model for the prediction of COVID-19 in the USA: a time-series study.” eng. In: *BMJ Open* 12.7 (2022), e056685. ISSN: 2044-6055 (Electronic); 2044-6055 (Linking). DOI: 10.1136/bmjopen-2021-056685.

[89] Richard J. Woodman and Arduino A. Mangoni. “A comprehensive review of machine learning algorithms and their application in geriatric medicine: present and future”. In: *Aging Clinical and Experimental Research* 35.11 (2023), pp. 2363–2397. DOI: 10.1007/s40520-023-02552-2.

[90] Divanshu Singh et al. *Predicting Lung Cancer using XGBoost and other Ensemble Learning Models.* July 2023, pp. 1–6. DOI: 10.1109/ICCCNT56998.2023.10308301.

[91] Sanford Weisberg. *Applied Linear Regression.* 3rd. John Wiley & Sons, 2005.

[92] Robert Tibshirani. “Regression Shrinkage and Selection Via the Lasso”. In: *Journal of the Royal Statistical Society: Series B (Methodological)* 58.1 (1996), pp. 267–288. DOI: https://doi.org/10.1111/j.2517-6161.1996.tb02080.x.

[93] Arthur E. Hoerl and Robert W. Kennard. “Ridge Regression: Biased Estimation for Nonorthogonal Problems”. In: *Technometrics* 12.1 (1970), pp. 55–67.

[94] Hui Zou and Trevor Hastie. “Regularization and Variable Selection via the Elastic Net”. In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 67.2 (2005), pp. 301–320. DOI: 10.1111/j.1467-9868.2005.00503.x.

[95] Jacob Murel and Eda Kavlakoglu. "What Is Regularization?" In: *IBM* (2023). URL: https://www.ibm.com/topics/regularization.

[96] Harris Drucker et al. "Support Vector Regression Machines". In: *Advances in Neural Information Processing Systems*. Ed. by M.C. Mozer, M. Jordan, and T. Petsche. Vol. 9. MIT Press, 1996. URL: https://proceedings.neurips.cc/paper_files/paper/1996/file/d38901788c533e8286cb6400b40b386d-Paper.pdf.

[97] Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik. "A training algorithm for optimal margin classifiers". In: *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*. COLT '92. Pittsburgh, Pennsylvania, USA: Association for Computing Machinery, 1992, pp. 144–152. ISBN: 089791497X. DOI: 10.1145/130385.130401.

[98] Ryan Rifkin and Aldebaro Klautau. "In Defense of One-Vs-All Classification". In: *Journal of Machine Learning Research* 5 (Dec. 2004), pp. 101–141.

[99] Kai-Bo Duan and S. Sathiya Keerthi. "Which Is the Best Multiclass SVM Method? An Empirical Study". In: *Multiple Classifier Systems*. Ed. by Nikunj C. Oza et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 278–285. ISBN: 978-3-540-31578-0.

[100] Chih-Wei Hsu and Chih-Jen Lin. "A comparison of methods for multiclass support vector machines". In: *IEEE Transactions on Neural Networks* 13.2 (2002), pp. 415–425. DOI: 10.1109/72.991427.

[101] F ROSENBLATT. "The perceptron: a probabilistic model for information storage and organization in the brain." eng. In: *Psychol Rev* 65.6 (1958), pp. 386–408. ISSN: 0033-295X (Print); 0033-295X (Linking). DOI: 10.1037/h0042519.

[102] John Scott Bridle. "Probabilistic Interpretation of Feedforward Classification Network Outputs, with Relationships to Statistical Pattern Recognition". In: *NATO Neurocomputing*. 1989.

[103] Kunihiko Fukushima. "Visual Feature Extraction by a Multilayered Network of Analog Threshold Elements". In: *IEEE Transactions on Systems Science and Cybernetics* 5.4 (1969), pp. 322–333. DOI: 10.1109/TSSC.1969.300225.

[104] Vinod Nair and Geoffrey Hinton. "Rectified Linear Units Improve Restricted Boltzmann Machines Vinod Nair". In: vol. 27. June 2010, pp. 807–814.

[105] Andrew L. Maas. "Rectifier Nonlinearities Improve Neural Network Acoustic Models". In: 2013. URL: https://api.semanticscholar.org/CorpusID:16489696.

[106] Dan Hendrycks and Kevin Gimpel. *Gaussian Error Linear Units (GELUs)*. 2023. arXiv: 1606.08415 [cs.LG].

[107] Augustin-Louis Cauchy. "Méthode générale pour la résolution des systèmes d'équations simultanées". In: *Comptes Rendus Hebdomadaires des Séances de l'Académie des Sciences* 25 (1847), pp. 536–538.

[108] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. "Learning representations by back-propagating errors". In: *Nature* 323.6088 (1986), pp. 533–536. DOI: 10.1038/323533a0.

[109] Hecht-Nielsen. "Theory of the backpropagation neural network". In: *International 1989 Joint Conference on Neural Networks*. 1989, 593–605 vol.1. DOI: 10.1109/IJCNN.1989.118638.

[110] Sergey Ioffe and Christian Szegedy. *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift*. 2015. arXiv: 1502.03167.

[111] Shiva Verma. *Understanding 1D and 3D Convolution Neural Network | Keras*. https://medium.com/towards-data-science/understanding-1d-and-3d-convolution-neural-network-keras-9d8f76e29610.

[112] Chen-Yu Lee et al. "Deeply-Supervised Nets". In: *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*. Ed. by Guy Lebanon and S. V. N. Vishwanathan. Vol. 38. Proceedings of Machine Learning Research. San Diego, California, USA: PMLR, 2015, pp. 562–570. URL: https://proceedings.mlr.press/v38/lee15a.html.

[113] Liwei Wang et al. *Training Deeper Convolutional Networks with Deep Supervision*. 2015. arXiv: 1505.02496.

[114] Qi Dou et al. "3D deeply supervised network for automated segmentation of volumetric medical images". In: *Medical Image Analysis* 41 (2017). Special Issue on the 2016 Conference on Medical Image Computing and Computer Assisted Intervention (Analog to MICCAI 2015), pp. 40–54. ISSN: 1361-8415. DOI: https://doi.org/10.1016/j.media.2017.05.001.

[115] Fabian Isensee et al. "nnU-Net: a self-configuring method for deep learning-based biomedical image segmentation". In: *Nature Methods* 18.2 (2021), pp. 203–211. DOI: 10.1038/s41592-020-01008-z.

[116] Ashish Vaswani et al. “Attention is All you Need”. In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc., 2017. URL: `https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf`.

[117] Hu Cao et al. “Swin-Unet: Unet-like Pure Transformer for Medical Image Segmentation”. In: *arXiv preprint arXiv:2105.05537* (2021).

[118] Ali Hatamizadeh et al. “Swin UNETR: Swin Transformers for Semantic Segmentation of Brain Tumors in MRI Images”. In: *International MICCAI Brainlesion Workshop*. Springer. 2022, pp. 272–284.

[119] Alec Radford et al. “Improving language understanding by generative pre-training”. In: *OpenAI Blog* (2018). URL: `https://openai.com/blog/language-unsupervised/`.

[120] Jacob Devlin et al. “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. 2019, pp. 4171–4186.

[121] OpenAI. “ChatGPT: Optimizing Language Models for Dialogue”. In: *OpenAI Blog* (2022). URL: `https://openai.com/blog/chatgpt/`.

[122] Alexey Dosovitskiy et al. “An image is worth 16x16 words: Transformers for image recognition at scale”. In: *International Conference on Learning Representations*. 2021.

[123] Sixiao Zheng et al. “Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 6881–6890.

[124] Luca Deininger et al. *A comparative study between vision transformers and CNNs in digital pathology*. 2022. arXiv: `2206.00389 [eess.IV]`.

[125] Yucheng Tang et al. “Self-Supervised Pre-Training of Swin Transformers for 3D Medical Image Analysis”. In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022, pp. 20698–20708. DOI: `10.1109/CVPR52688.2022.02007`.

[126] Ze Liu et al. “Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 2021, pp. 10012–10022.

[127] Michela Antonelli et al. “The Medical Segmentation Decathlon”. In: *Nature Communications* 13.1 (2022), p. 4128. DOI: 10.1038/s41467-022-30695-9.

[128] Bennett Landman et al. “MICCAI multi-atlas labeling beyond the cranial vault–workshop and challenge”. In: *Proc. MICCAI Multi-Atlas Labeling Beyond Cranial Vault—Workshop Challenge*. Vol. 5. 2015, p. 12. DOI: 10.7303/syn3193805.

[129] *ImFusion Labels (software)*. Retrieved from https://www.imfusion.com/products/imfusion-labels. 2022.

[130] Guruprasad P Aithal et al. “Guidelines on the management of ascites in cirrhosis”. In: *Gut* 70.1 (2021), pp. 9–29. ISSN: 0017-5749. DOI: 10.1136/gutjnl-2020-321790.

[131] Niklas Verloh et al. “Diagnostic performance of Gd-EOB-DTPA-enhanced MRI for evaluation of liver dysfunction: A multivariable analysis of 3T MRI sequences”. In: *Oncotarget* 9 (Nov. 2018). DOI: 10.18632/oncotarget.26368.

[132] Yaacob Yazmin. “A Surrogate for Liver Function: The Usefulness of Liver Enhancement in Hepatobiliary Phase in MRI Liver using Hepatocyte Specific Contrast Agent”. In: *International Journal of Radiology and Imaging Techniques* 7 (Oct. 2021). DOI: 10.23937/2572-3235.1510079.

[133] Tsang Lau and Jawad Ahmad. “Clinical applications of the Model for End-Stage Liver Disease (MELD) in hepatic medicine.” eng. In: *Hepat Med* 5 (2013), pp. 1–10. ISSN: 1179-1535 (Print); 1179-1535 (Electronic); 1179-1535 (Linking). DOI: 10.2147/HMER.S9049.

[134] James F. Trotter and Michael J. Osgood. “MELD Scores of Liver Transplant Recipients According to Size of Waiting ListImpact of Organ Allocation and Patient Outcomes”. In: *JAMA* 291.15 (Apr. 2004), pp. 1871–1874. ISSN: 0098-7484. DOI: 10.1001/jama.291.15.1871.

[135] Jassin Rashidi-Alavijeh et al. “Enzymatic liver function measured by LiMAx is superior to current standard methods in predicting transplant-free survival after TIPS implantation.” eng. In: *Sci Rep* 11.1 (2021), p. 13834. ISSN: 2045-2322 (Electronic); 2045-2322 (Linking). DOI: 10.1038/s41598-021-93392-5.

[136] Janett Fischer et al. “The Liver Maximum Capacity Test (LiMAx) Is Associated with Short-Term Survival in Patients with Early Stage HCC Undergoing Transarterial Treatment.” eng. In: *Cancers (Basel)* 14.21 (2022). ISSN: 2072-6694 (Print); 2072-6694 (Electronic); 2072-6694 (Linking). DOI: 10.3390/cancers14215323.

[137] Xiangrui Li et al. "The first step for neuroimaging data analysis: DICOM to NIfTI conversion." eng. In: *J Neurosci Methods* 264 (2016), pp. 47–56. ISSN: 1872-678X (Electronic); 0165-0270 (Linking). DOI: 10.1016/j.jneumeth.2016.03.001.

[138] Nicholas J. Tustison et al. "N4ITK: Improved N3 Bias Correction". In: *IEEE Transactions on Medical Imaging* 29.6 (2010), pp. 1310–1320. DOI: 10.1109/TMI.2010.2046908.

[139] Charles R. Harris et al. "Array programming with NumPy". In: *Nature* 585.7825 (2020), pp. 357–362. DOI: 10.1038/s41586-020-2649-2.

[140] Brian B Avants et al. "The Advanced Normalization Tools (ANTS)". In: *Neuroimage* 2.3 (2009), pp. 1650–1660.

[141] Krzysztof J Gorgolewski et al. "Nipype: a flexible, lightweight and extensible neuroimaging data processing framework in Python". In: *Frontiers in neuroinformatics* 5 (2011), p. 13.

[142] Stephen M. Smith et al. "Advances in functional and structural MR image analysis and implementation as FSL". In: *NeuroImage* 23 (2004). Mathematics in Brain Imaging, S208–S219. ISSN: 1053-8119. DOI: https://doi.org/10.1016/j.neuroimage.2004.07.051.

[143] International Journal of Advanced Computer Science and Applications. "Cross Correlation versus Mutual Information for Image Mosaicing". In: *Academia.edu* (2013).

[144] C. Steffen et al. "Semantic similarity metrics for image registration". In: *ScienceDirect* (2023).

[145] Tanvi Kulkarni et al. "Registration Quality Evaluation Metric with Self-Supervised Siamese Networks". In: *Medical Imaging with Deep Learning*. 2024. URL: https://openreview.net/forum?id=2wrQLJtygh.

[146] M. Chen, N. J. Tustison, R. Jena, et al. "Image Registration: Fundamentals and Recent Advances Based on Deep Learning". In: *Machine Learning for Brain Disorders*. Ed. by O. Colliot. Available from: https://www.ncbi.nlm.nih.gov/books/NBK597490/. New York, NY: Humana, 2023. Chap. 14. DOI: 10.1007/978-1-0716-3195-9_14.

[147] Steffen Czolbe et al. "Semantic similarity metrics for image registration". In: *Medical Image Analysis* 87 (2023), p. 102830. ISSN: 1361-8415. DOI: https://doi.org/10.1016/j.media.2023.102830.

[148] Shatabdi Basu, Sunita Singhal, and Dilbag Singh. “A Systematic Literature Review on Multimodal Medical Image Fusion”. In: *Multimedia Tools and Applications* 83.6 (2024), pp. 15845–15913. DOI: `10.1007/s11042-023-15913-w`.

[149] F Maes et al. “Multimodality image registration by maximization of mutual information.” eng. In: *IEEE Trans Med Imaging* 16.2 (1997), pp. 187–198. ISSN: 0278-0062 (Print); 0278-0062 (Linking). DOI: `10.1109/42.563664`.

[150] Josien P W Pluim, J B Antoine Maintz, and Max A Viergever. “Mutual-information-based registration of medical images: a survey.” eng. In: *IEEE Trans Med Imaging* 22.8 (2003), pp. 986–1004. ISSN: 0278-0062 (Print); 0278-0062 (Linking). DOI: `10.1109/TMI.2003.815867`.

[151] William M. Wells et al. “Multi-modal volume registration by maximization of mutual information”. In: *Medical Image Analysis* 1.1 (1996), pp. 35–51. ISSN: 1361-8415. DOI: `https://doi.org/10.1016/S1361-8415(01)80004-9`.

[152] Zhou Wang et al. “Image quality assessment: from error visibility to structural similarity”. In: *IEEE transactions on image processing* 13.4 (2004), pp. 600–612.

[153] G.P. Penney et al. “A comparison of similarity measures for use in 2-D-3-D medical image registration”. In: *IEEE Transactions on Medical Imaging* 17.4 (1998), pp. 586–595. DOI: `10.1109/42.730403`.

[154] Shan Liu et al. “2D/3D Multimode Medical Image Registration Based on Normalized Cross-Correlation”. In: *Applied Sciences* 12.6 (2022). ISSN: 2076-3417. DOI: `10.3390/app12062828`.

[155] Lee R. Dice. “Measures of the Amount of Ecologic Association Between Species”. In: *Ecology* 26.3 (1945), pp. 297–302. DOI: `https://doi.org/10.2307/1932409`.

[156] T. A. SORENSEN. “A method of establishing groups of equal amplitude in plant sociology based on similarity of species content and its application to analyses of the vegetation on Danish commons”. In: *Biol. Skar.* 5 (1948), pp. 1–34. URL: `https://ci.nii.ac.jp/naid/10008878962/en/`.

[157] Paul Jaccard. “Lois de distribution florale dans la zone alpine”. In: *Bulletin de la Société vaudoise des sciences naturelles* 38 (Jan. 1902), pp. 69–130. DOI: `10.5169/seals-266762`.

[158] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. *U-Net: Convolutional Networks for Biomedical Image Segmentation.* 2015. arXiv: `1505.04597 [cs.CV]`.

[159] Hao Chen et al. "Deep learning for cardiac image segmentation: A review". In: *Frontiers in Cardiovascular Medicine* 7 (2020), p. 25.

[160] Wei Liu, Xiaoming Zhang, et al. "Automatic segmentation of kidney tumors in CT images based on nnU-Net framework". In: *Journal of Medical Imaging* 9.2 (2022), p. 024001.

[161] Nuno M Rodrigues et al. "A Comparative Study of Automated Deep Learning Segmentation Models for Prostate MRI." eng. In: *Cancers (Basel)* 15.5 (2023). ISSN: 2072-6694 (Print); 2072-6694 (Electronic); 2072-6694 (Linking). DOI: `10.3390/cancers15051467`.

[162] Fabian Isensee et al. *nnU-Net Revisited: A Call for Rigorous Validation in 3D Medical Image Segmentation.* 2024. arXiv: `2404.09556 [cs.CV]`.

[163] MONAI Consortium. *MONAI: Medical Open Network for AI.* Version 1.3.2. June 2024. DOI: `10.5281/zenodo.12542217`.

[164] Alex Zwanenburg et al. "The Image Biomarker Standardization Initiative: Standardized Quantitative Radiomics for High-Throughput Image-based Phenotyping". In: *Radiology* 295.2 (May 2020), pp. 328–338. ISSN: 1527-1315. DOI: `10.1148/radiol.2020191145`.

[165] G. Collewet, M. Strzelecki, and F. Mariette. "Influence of MRI acquisition protocols and image intensity normalization methods on texture classification". In: *Magnetic Resonance Imaging* 22.1 (2004), pp. 81–91. ISSN: 0730-725X. DOI: `https://doi.org/10.1016/j.mri.2003.09.001`.

[166] M Vallières et al. "A radiomics model from joint FDG-PET and MRI texture features for the prediction of lung metastases in soft-tissue sarcomas of the extremities". In: *Physics in Medicine & Biology* 60.14 (2015), p. 5471. DOI: `10.1088/0031-9155/60/14/5471`.

[167] Janita E. van Timmeren et al. "Radiomics in medical imaging—"how-to"guide and critical reflection". In: *Insights into Imaging* 11.1 (2020), p. 91. DOI: `10.1186/s13244-020-00887-2`.

[168] N Verloh et al. "Assessing liver function by liver enhancement during the hepatobiliary phase with Gd-EOB-DTPA-enhanced MRI at 3 Tesla." eng. In: *Eur Radiol* 24.5 (2014), pp. 1013–1019. ISSN: 1432-1084 (Electronic); 0938-7994 (Linking). DOI: `10.1007/s00330-014-3108-y`.

[169] Ute Probst et al. "Efficacy of dynamic enhancement effects on Gd-EOB-DTPA-enhanced MRI for estimation of liver function assessed by 13C-Methacetin breath test". In: *Clinical Hemorheology and Microcirculation* 70.4 (2018), pp. 595–604. DOI: `10.3233/CH-189324`.

[170] Michael Haimerl et al. "Hepatobiliary MRI: Signal intensity based assessment of liver function correlated to 13C-Methacetin breath test". In: *Scientific Reports* 8.1 (2018), p. 9078. DOI: `10.1038/s41598-018-27401-5`.

[171] Sarah Poetter-Lang et al. "Quantification of liver function using gadoxetic acid-enhanced MRI." eng. In: *Abdom Radiol (NY)* 45.11 (2020), pp. 3532–3544. ISSN: 2366-0058 (Electronic); 2366-004X (Print). DOI: `10.1007/s00261-020-02779-x`.

[172] Ming Yang et al. "Evaluation of liver function using liver parenchyma, spleen and portal vein signal intensities during the hepatobiliary phase in Gd-EOB-D TPA-enhanced MRI." eng. In: *BMC Med Imaging* 20.1 (2020), p. 119. ISSN: 1471-2342 (Electronic); 1471-2342 (Linking). DOI: `10.1186/s12880-020-00519-7`.

[173] Chenxia Li et al. "Multiparametric MRI combined with liver volume for quantitative evaluation of liver function in patients with cirrhosis." eng. In: *Diagn Interv Radiol* 28.6 (2022), pp. 547–554. ISSN: 1305-3612 (Electronic); 1305-3825 (Print); 1305-3825 (Linking). DOI: `10.5152/dir.2022.211325`.

[174] Antonio Luis Eiras-Araújo et al. "Relative enhancement index can be used to quantify liver function in cirrhotic patients that undergo gadoxetic acid-enhanced MRI." eng. In: *Eur Radiol* 33.7 (2023), pp. 5142–5149. ISSN: 1432-1084 (Electronic); 0938-7994 (Linking). DOI: `10.1007/s00330-023-09402-9`.

[175] Uluhan Eryuruk et al. "Comparison of the efficacy of the gadoxetic acid MRI-derived relative enhancement index (REI) and functional liver imaging score (FLIS) in predicting liver function: validation with Albumin-Bilirubin (ALBI) grade". In: *Abdominal Radiology* 49.5 (2024), pp. 1456–1466. DOI: `10.1007/s00261-024-04324-6`. URL: `https://doi.org/10.1007/s00261-024-04324-6`.

[176] Takashi Katsube et al. "Estimation of Liver Function Using T1 Mapping on Gd-EOB-DTPA-Enhanced Magnetic Resonance Imaging". In: *Investigative Radiology* 46.4 (2011). URL: `https://journals.lww.com/investigativeradiology/fulltext/2011/04000/estimation_of_liver_function_using_t1_mapping_on.9.aspx`.

[177] Sheng Xie et al. “Assessment of liver function and liver fibrosis with dynamic Gd-EOB-DTPA-enhanced MRI.” eng. In: *Acad Radiol* 22.4 (2015), pp. 460–466. ISSN: 1878-4046 (Electronic); 1076-6332 (Linking). DOI: `10.1016/j.acra.2014.11.006`.

[178] Verena Carola Obmann et al. “T1 reduction rate with Gd-EOB-DTPA determines liver function on both 1.5 T and 3 T MRI”. In: *Scientific Reports* 12.1 (2022), p. 4716. DOI: `10.1038/s41598-022-08659-2`.

[179] Boyang Ma et al. “Evaluation of liver function using Gd-EOB-DTPA-enhanced MRI with T1 mapping”. In: *BMC Medical Imaging* 23.1 (2023), p. 73. DOI: `10.1186/s12880-023-01028-z`.

[180] Tatsuaki Sumiyoshi et al. “CT/99mTc-GSA SPECT fusion images demonstrate functional differences between the liver lobes.” eng. In: *World J Gastroenterol* 19.21 (2013), pp. 3217–3225. ISSN: 2219-2840 (Electronic); 1007-9327 (Print); 1007-9327 (Linking). DOI: `10.3748/wjg.v19.i21.3217`.

[181] Zhi-Peng Zhou et al. “Evaluating segmental liver function using T1 mapping on Gd-EOB-DTPA-enhanced MRI with a 3.0 Tesla.” eng. In: *BMC Med Imaging* 17.1 (2017), p. 20. ISSN: 1471-2342 (Electronic); 1471-2342 (Linking). DOI: `10.1186/s12880-017-0192-x`.

[182] Quirin David Strotzer et al. “Application of A U-Net for Map-like Segmentation and Classification of Discontinuous Fibrosis Distribution in Gd-EOB-DTPA-Enhanced Liver MRI”. In: *Diagnostics* 12.8 (2022). ISSN: 2075-4418. DOI: `10.3390/diagnostics12081938`.

[183] Philippe Lambin et al. “Radiomics: extracting more information from medical images using advanced feature analysis”. In: *European Journal of Cancer* 48.4 (2012), pp. 441–446. DOI: `10.1016/j.ejca.2011.11.036`.

[184] Joost J. M. van Griethuysen et al. “Computational Radiomics System to Decode the Radiographic Phenotype”. In: *Cancer Research* 77.21 (2017). This work was supported in part by the US National Cancer Institute grant 5U24CA194354, QUANTITATIVE RADIOMICS SYSTEM DECODING THE TUMOR PHENOTYPE., e104–e107. DOI: `10.1158/0008-5472.CAN-17-0339`.

[185] Amalia Tsoris and Courtney A. Marlar. *Use Of The Child Pugh Score In Liver Disease*. [Updated 2023 Mar 13]. Treasure Island (FL): StatPearls Publishing, 2024. URL: `https://www.ncbi.nlm.nih.gov/books/NBK542308/`.

[186] F. Pedregosa et al. “Scikit-learn: Machine Learning in Python”. In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[187] Guillaume Lemaître, Fernando Nogueira, and Christos K. Aridas. "Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning". In: *Journal of Machine Learning Research* 18.17 (2017), pp. 1–5. URL: `http://jmlr.org/papers/v18/16-365.html`.

[188] Hui Han, Wen-Yuan Wang, and Bing-Huan Mao. "Borderline-SMOTE: A New Over-Sampling Method in Imbalanced Data Sets Learning". In: *Advances in Intelligent Computing*. Ed. by De-Shuang Huang, Xiao-Ping Zhang, and Guang-Bin Huang. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 878–887. ISBN: 978-3-540-31902-3.

[189] Michael Haimerl et al. "MRI-based estimation of liver function: Gd-EOB-DTPA-enhanced T1 relaxometry of 3T vs. the MELD score". In: *Scientific Reports* 4.1 (2014), p. 5621. DOI: `10.1038/srep05621`.

Appendix

Table A.1 This table illustrates the final feature sets that were used for the models of each cohort after the feature engineering.

MELD	**ALBI**	**LiMAx**
$\mathrm{Vol}^{\mathrm{Liver}}$	$\mathrm{Vol}^{\mathrm{Liver}}$	$\mathrm{Vol}^{\mathrm{Liver}}$
$\mathrm{Vol}^{\mathrm{Liver}} \cdot \mathrm{REI}_2^{\mathrm{Liver}}$	$\mathrm{Vol}^{\mathrm{Liver}} \cdot \mathrm{REI}_1^{\mathrm{Liver}}$	$\mathrm{Vol}^{\mathrm{Liver}} \cdot \mathrm{REI}_2^{\mathrm{Liver}}$
$\mathrm{Vol}^{\mathrm{Liver}} \cdot \mathrm{REI}_3^{\mathrm{Liver}}$	$\mathrm{Vol}^{\mathrm{Liver}} \cdot \mathrm{REI}_2^{\mathrm{Liver}}$	$\mathrm{Vol}^{\mathrm{Liver}} \cdot \mathrm{REI}_3^{\mathrm{Liver}}$
$\mathrm{Vol}^{\mathrm{Liver}} \cdot \mathrm{REI}_4^{\mathrm{Liver}}$	$\mathrm{Vol}^{\mathrm{Liver}} \cdot \mathrm{REI}_3^{\mathrm{Liver}}$	$\mathrm{Vol}^{\mathrm{Liver}} \cdot \mathrm{REI}_4^{\mathrm{Liver}}$
$\mathrm{REI}_4^{\mathrm{Liver}}$	$\mathrm{Vol}^{\mathrm{Liver}} \cdot \mathrm{REI}_4^{\mathrm{Liver}}$	$\mathrm{REI}_2^{\mathrm{Liver}}$
$\mathrm{REI}_4^{\mathrm{Liver}}{}_{\text{(median)}}$	$\mathrm{REI}_2^{\mathrm{Liver}}$	$\mathrm{REI}_4^{\mathrm{Liver}}$
$\mathrm{REI}_4^{\mathrm{Liver}}{}_{(10^{th}\text{-percentile})}$	$\mathrm{REI}_3^{\mathrm{Liver}}$	$\mathrm{REI}_2^{\mathrm{Liver}}{}_{\text{(median)}}$
$\mathrm{REI}_1^{\mathrm{Liver}}{}_{\text{(interquartile)}}$	$\mathrm{REI}_4^{\mathrm{Liver}}$	$\mathrm{REI}_4^{\mathrm{Liver}}{}_{\text{(median)}}$
$\mathrm{REI}_4^{\mathrm{Liver}}{}_{\text{(interquartile)}}$	$\mathrm{REI}_2^{\mathrm{Liver}}{}_{\text{(interquartile)}}$	$\mathrm{REI}_3^{\mathrm{Liver}}{}_{\text{(interquartile)}}$
$\mathrm{REI}_3^{\mathrm{PV}}$	$\mathrm{REI}_3^{\mathrm{Liver}}{}_{\text{(interquartile)}}$	$\mathrm{REI}_1^{\mathrm{PV}}$
$\mathrm{REI}_4^{\mathrm{PV}}$	$\mathrm{REI}_4^{\mathrm{Liver}}{}_{\text{(interquartile)}}$	$\mathrm{REI}_2^{\mathrm{PV}}$
$\mathrm{REI}_2^{\mathrm{HV}}$	$\mathrm{REI}_1^{\mathrm{PV}}$	$\mathrm{REI}_3^{\mathrm{PV}}$
$\mathrm{REI}_3^{\mathrm{HV}}$	$\mathrm{REI}_2^{\mathrm{PV}}$	$\mathrm{REI}_4^{\mathrm{PV}}$
$\mathrm{REI}_4^{\mathrm{HV}}$	$\mathrm{REI}_3^{\mathrm{PV}}$	$\mathrm{REI}_2^{\mathrm{HV}}$
$\mathrm{REI}_1^{\text{Abdominal aorta}}$	$\mathrm{REI}_4^{\mathrm{PV}}$	$\mathrm{REI}_3^{\mathrm{HV}}$
	$\mathrm{REI}_2^{\mathrm{HV}}$	$\mathrm{REI}_1^{\text{Abdominal aorta}}$
	$\mathrm{REI}_3^{\mathrm{HV}}$	$\mathrm{REI}_3^{\text{Abdominal aorta}}$
	$\mathrm{REI}_4^{\mathrm{HV}}$	$\mathrm{REI}_1^{\text{Thoracic aorta}}$
	$\mathrm{REI}_1^{\text{Abdominal aorta}}$	$\mathrm{REI}_2^{\text{Thoracic aorta}}$
	$\mathrm{REI}_2^{\text{Abdominal aorta}}$	$\mathrm{REI}_3^{\text{Thoracic aorta}}$
	$\mathrm{REI}_3^{\text{Abdominal aorta}}$	
	$\mathrm{REI}_4^{\text{Abdominal aorta}}$	

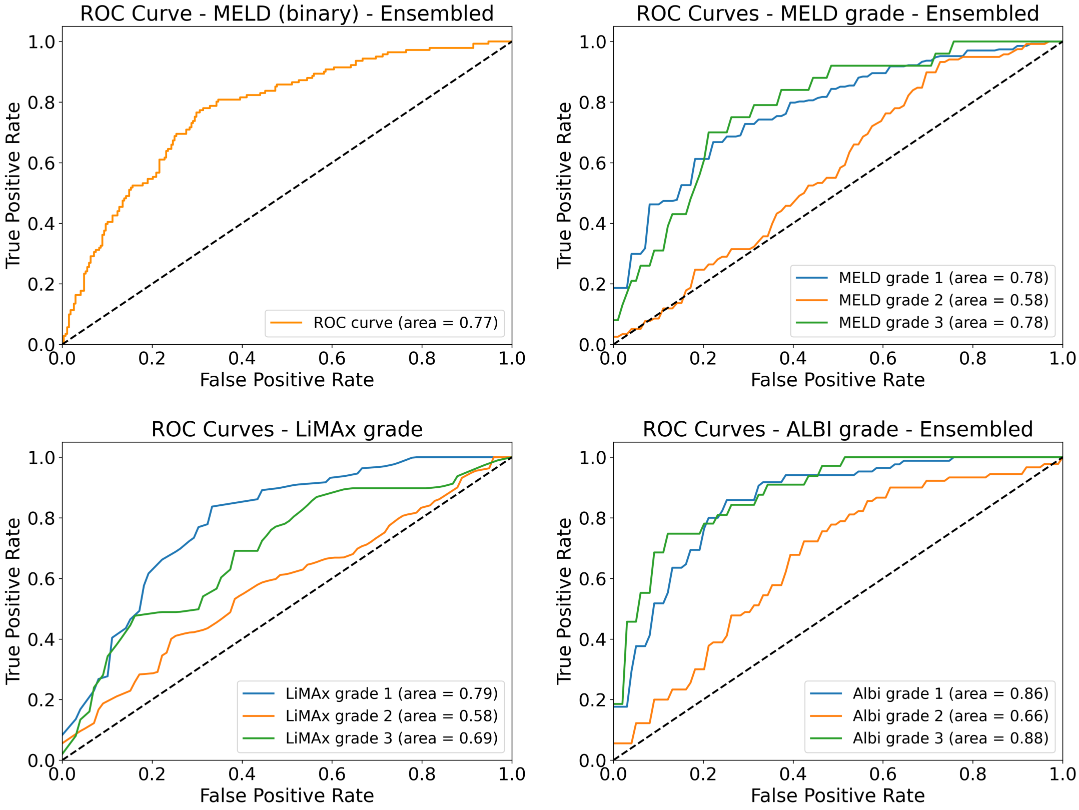


Figure A.1 Receiver Operating Characteristic (ROC) Curves and Area Under Curve (AUC) for various liver function score classifications using only one feature ($\mathrm{REI}_4^{\mathrm{Liver}}$). The dashed line represents the performance of a random classifier for balanced datasets. The **top left** panel displays results for binary classification of the MELD score, while the **top right** plot illustrates three-class classification results for this score. The **bottom left** panel shows results for LiMAx grades, and the **bottom right** panel presents results for ALBI grades. MELD results were achieved through ensembled predictions, whereas LiMAx and ALBI results were generated using single-model predictions.

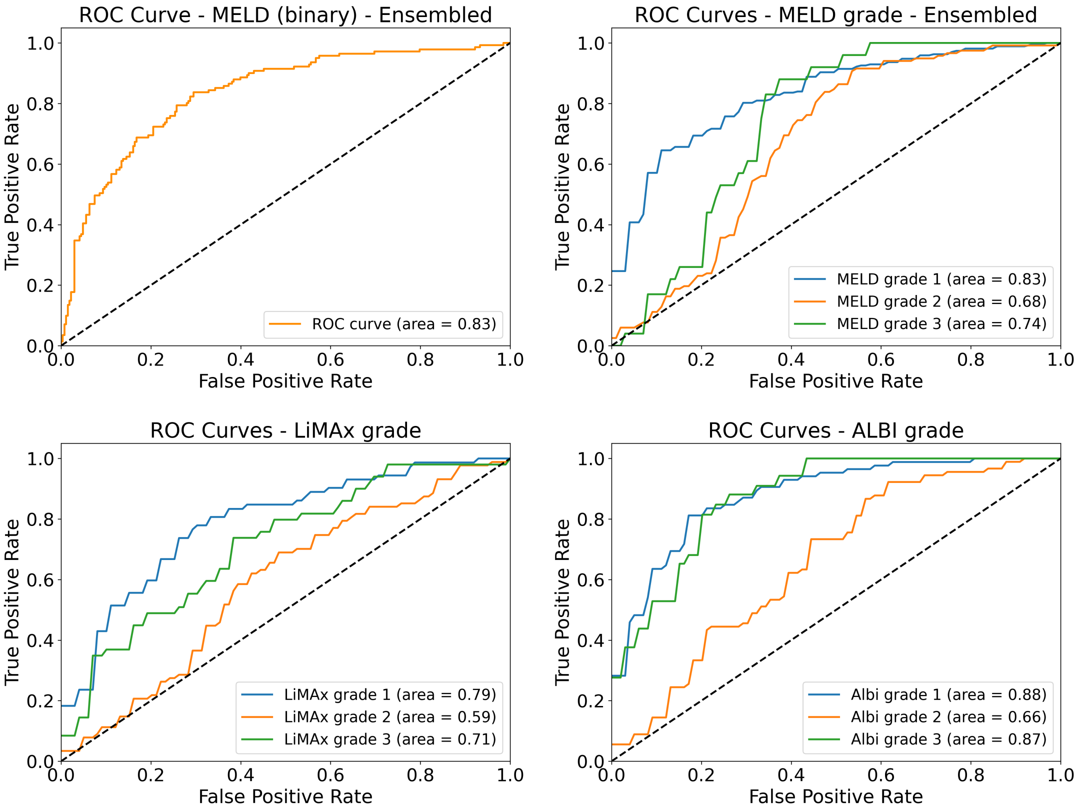


Figure A.2 Receiver Operating Characteristic (ROC) Curves and Area Under Curve (AUC) for various liver function score classifications using only one feature ($\mathrm{Vol}^{\mathrm{Liver}} \cdot \mathrm{REI}_4^{\mathrm{Liver}}$). The dashed line represents the performance of a random classifier for balanced datasets. The **top left** panel displays results for binary classification of the MELD score, while the **top right** plot illustrates three-class classification results for this score. The **bottom left** panel shows results for LiMAx grades, and the **bottom right** panel presents results for ALBI grades. MELD results were achieved through ensembled predictions, whereas LiMAx and ALBI results were generated using single-model predictions.

Acknowledgments

I would like to express my heartfelt gratitude to the following people for their invaluable support during the course of my thesis:

Prof. Dr. Elmar W. Lang for his constant support and guidance throughout my thesis journey during the last few years. His insightful supervision and persistent encouragement have been instrumental in shaping my research experience. I am deeply grateful for the privilege of working under his mentorship, being able to experience his patience, care and trust, which been very impactful on my academic and personal growth.

Prof. Dr. Michael Haimerl for facilitating this interdisciplinary collaboration between the physics faculty and the institute for radiology at the UKR, thereby offering me the great opportunity to conduct research in this field. I would like to thank him for proposing the topic of my thesis on MRI-based liver function estimation. His provision of the segmentation dataset, assistance with manual annotations and medical supervision were very helpful for the success of this research.

Prof. Dr. Ingo Einspieler for taking over the medical supervision of my thesis after Prof. Dr. Haimerl's departure from the UKR. His expertise and support were invaluable in ensuring the continued progress and successful completion of this research.

Dr. Quirin Strotzer for his consistently friendly and helpful support on numerous medical questions. Our frequent discussions and his valuable feedback were really helpful in the success of our collaborative projects. Additionally, he acted as a vital intermediary between the physicians at the UKR and myself as a physicist, playing a crucial role in bridging the gap between our different fields of expertise, aiding the successful completion of this interdisciplinary research.

Prof. Dr. Christian Stroszczynski for providing me with the essential resources, equipment, and office space necessary to conduct this research and offering me a job as a PhD student to finance this work. His support in creating a suitable environment for this study was essential for its completion.

Additionally, I would like to thank Thomas for the numerous entertaining coffee breaks during his time at the UKR that always led to inspiring thoughts. I would like to thank all my other workmates at the radiology's IT department for creating a pleasant working environment and including me into their team. Last but not least, I would like to thank my family and friends for their support over the last few years.