UNIVERSITY OF BOLOGNA

FACULTY OF PHYSICS AND ASTRONOMY

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

IMAGE PROCESSING

REPORT

Liver & Tumor Segmentation

Students
Luca Gardini
Tommaso Carota
Giovanni Costi
Riccardo Giuliani
Leonard Hell
Matteo Paul Giacomelli
Alessandro Pinto
Muhammad Sameer

Lecturer in charge:

Prof. Lanconelli

Submission Date : 08/01/2026

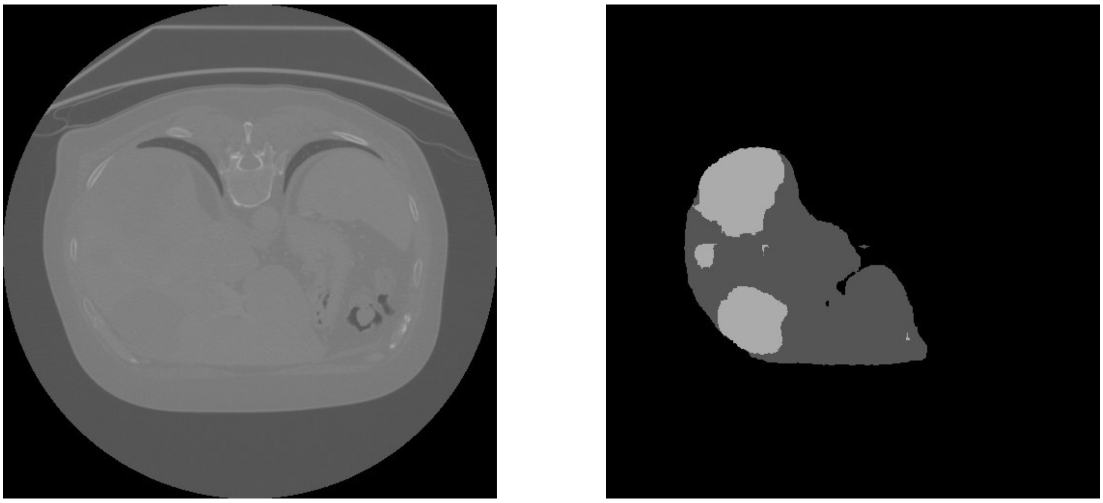
Contents

1 Introduction

The liver is a frequent site of benign and malignant, primary and metastatic tumors, and liver cancer is one of the most common cancer diseases in the world and causes massive deaths every year [1]. Accurate measurements from CT, including tumor volume, shape, location and further functional liver volume, can assist doctors in making accurate tumor evaluation and treatment planning. In clinical routine, the liver and liver lesion are segmented manually or semi-manually by radiologists on a slice-by-slice basis, which is time-consuming and prone to suffer from inter- and intra-observer variabilities. Therefore, automatic liver and liver tumor segmentation methods are highly demanded in the clinical practice.

Automatic liver segmentation is a challenging task due to the low intensity contrast between the liver and other neighboring organs, fuzzy boundaries and highly varying shape. Moreover, radiologists usually enhance CT scans by an injection protocol for clearly observing tumors, which may increase the noise inside the images on the liver region [2]. Compared with liver segmentation, liver tumor segmentation is considered to be an even more challenging task, because it normally suffers from significant variety of appearance in size, shape, location, intensity, textures, as well as the number of occurrences within one patient.

In this project we confronted ourselves with the Liver Tumor Segmentation Challenge [3]. It was set in 2017 by CodaLab and consists of a dataset of 201 CTs of cancerous livers complete with professional segmentations for verification. A slice from a CT scan and its respective ground truth segmentation present in the LiTS dataset are shown in Fig. 1.



(a)

(b)

Figure 1: Example of CT slice (a) and relative ground truth segmentation (b) from the LiTS challenge dataset.

The aim of the challenge is to automate the liver and tumor segmentation. Therefore in this project we tried to address the challenge by employing three different approaches to accurately segment both the liver and its tumors: a manual approach (using ImageJ), a supervised learning approach (using Weka), and an unsupervised training approach (using Neural Networks).

1.1 First Approach: ImageJ

The first approach we developed is based only on ImageJ [4] and its basic operations. The idea is to segment both the liver and its tumors without using any plugins. This approach makes use of operations such as enhance contrast, dilation, erosion, thresholding and many others to segment the liver. The resulting segmentation is then used as a mask to simplify the segmentation of liver tumors, which is done in the same way.

1.2 Second Approach: Weka

The second approach is based on classical machine learning techniques implemented in Fiji [5], a version of ImageJ, using the Weka 3D Segmentation framework [6]. In this approach, the given liver and tumor segmentations are used to train a supervised classification model that assigns each voxel in a CT volume to a semantic class, namely background, liver, or tumor. The segmentation task is formulated as a voxel-wise classification problem.

A critical aspect of this method is the proper definition and representation of the background class. Since the majority of voxels in abdominal CT scans do not belong to the liver or tumor, an insufficient or biased sampling of background voxels can lead to poor generalization and to an increase of false positive rates. Therefore, particular attention is paid to ensuring that background examples include a diverse set of anatomical structures and intensity ranges. Once trained, the Weka model can be applied to unseen CT volumes to automatically generate liver and tumor segmentations based on the learned feature-to-class relationships. This approach relies heavily on the quality of feature engineering and the representativeness of the training data.

1.3 Third Approach: Neural Networks

The third approach employs a deep learning-based method, in which a neural network is trained end-to-end to identify liver and tumor regions directly from CT images. Unlike the ImageJ approach, this method does not rely on manually guided image operations. Instead, the network automatically learns hierarchical feature representations that are optimized for the segmentation task during training. The annotated CT volumes serve as ground truth labels, enabling supervised learning of voxel-wise or pixel-wise predictions.

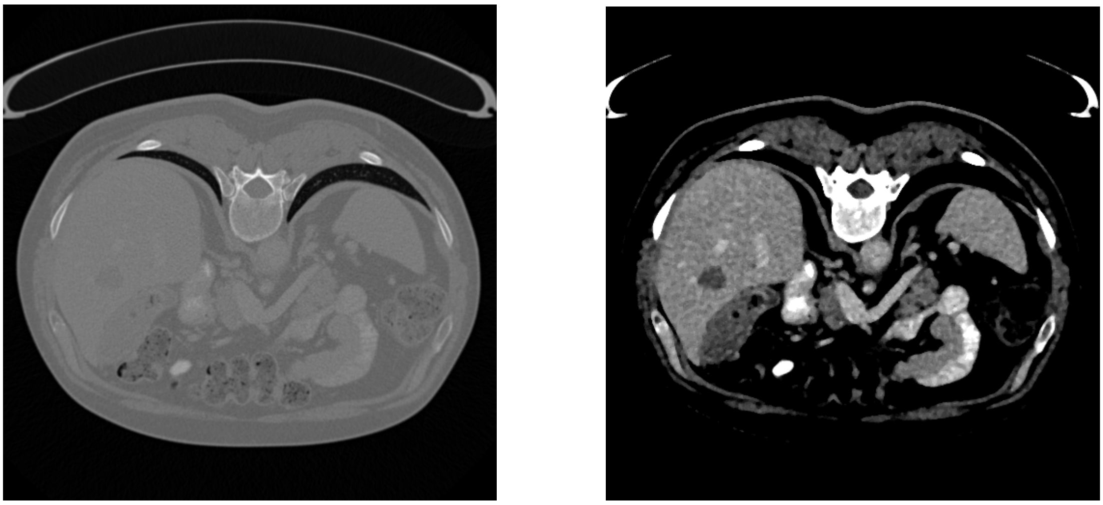
A key factor influencing the performance of this approach is the choice of network architecture. In particular, encoder–decoder architectures have proven effective for medical image segmentation, as they combine local feature extraction with multi-scale contextual information. The depth of the network, the use of skip connections, and the dimensionality (2D versus 3D) all play a crucial role in capturing anatomical structures of varying size and appearance. When properly designed and trained, neural networks are capable of achieving high segmentation accuracy and robustness, especially in complex scenarios such as tumor detection, where shape and intensity variations are significant.

2 Implementation

This section describes the practical implementation of the three segmentation approaches introduced in the previous section. The focus is placed on the realization of the methods using the available software tools and frameworks, rather than on their theoretical foundations.
Firstly, the manual approach that makes use of the most basic functions ImageJ has to offer is explained in order to ensure its repeatability. The operations used are described but the values of the parameters used have not been specified because they are to be determined by the operator on an image by image basis. Secondly, the implementation of the classical machine learning approach using Fiji (successor of ImageJ) and the Weka 3D Segmentation plugin is presented, including the preparation of training data, the handling of background classes and the training process. Lastly, the implementation of the neural network-based approach is outlined, covering data preprocessing, network configuration, and the training procedure. The goal of this section is to provide sufficient detail to ensure reproducibility and to highlight implementation-specific challenges encountered during the project.

2.1 ImageJ

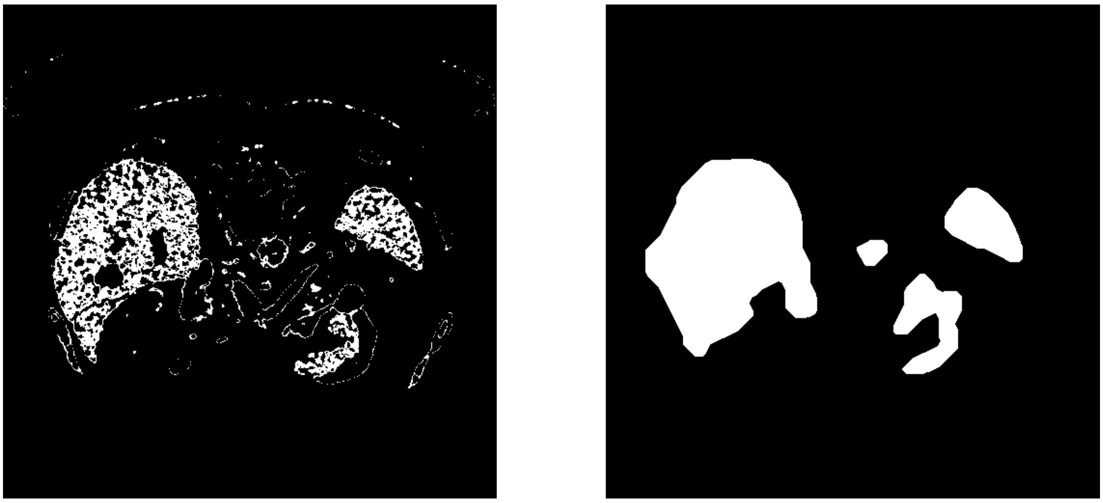
This approach consists of two parts: the segmentation of the liver to create a liver mask and then the application of the liver mask to a slice of the CT scan to repeat the process and segment the tumors.
To find anything in a CT scan, it is necessary to have a basic knowledge of human anatomy in order to tell the difference between different organs and tissues and to find the interesting slices inside a CT scan with hundreds of them. Liver and spleen for example are really similar both in shape and gray level, and CT scans usually start from the knees and end at the level of the heart.
We begin by transforming all the slices from 16 bits to 8 bits (Image → Type → 8-bit), and since we are looking for liver and liver tumors we exclude from our analysis all scans that do not contain the liver. Next, a slice is chosen between the remaining ones in the CT scan and on that slice all our operations will be done.
Firstly an area inside the liver is selected, with this selected area, it is possible to automatically adjust the contrast in order to enhance the visibility of the liver (Image → Adjust → Brightness/Contrast → Auto), which is limited (see Fig. 2a) despite the use of contrast enhancement liquid for liver lesions during the CT scans. It is possible to apply the same procedure again, if the quality of the image is not sufficiently improved. However, a side effect of this process is an excessive enhancement of the noise. To reduce this, it is sufficient to apply a median filter with range 2 pixels (Process → Filters → Median...) as seen in the Fig. 2b.



(a) (b)

Figure 2: Example of slice 380 from patient 10 before contrast enhancement and filtering (a) and after processing with enhanced contrast and median filtering (b).

After all these procedures, the image should be ready for the application of a threshold. The thresholding brightness fully depends on the operator, but a good rule of thumb could be that the segmented liver should be as isolated from other tissues as possible. Of course the process is not perfect, since it usually produces some white areas which should represent the liver, located outside of the liver location, see Fig. 9a. To partially resolve this issue, it is recommended to apply some binary operations. At first, an opening operation, then multiple dilate operations, and lastly a fill holes operation. The idea of these steps is to fill the gaps inside the area of interest (more of this later in Section 4.1). However, by doing so the undesired areas grow bigger, so it is necessary to apply at least two or three erosion operations. This should result in a binary image of the liver like the one shown in Fig. 9b that, when applied as a mask on the original image, allows to restrict the area to search for tumors.



(a) (b)

Figure 3: Result after thresholding on the slice 380 from patient 10 (a) and final liver mask (b).

In order to obtain a successful tumor segmentation, the first thing to do is to apply the liver mask just created (see Fig. 9b) on the whole image, which has to be already reduced to 8-bit, contrast enhanced and median filtered. The obtained image should be black outside the area of interest and ready for thresholding to segment the tumors, see Fig. 4a. The resulting image will be binary, with white areas that represent the tumors, and will still be affected by the same problems seen after the thresholding made to find the liver, see Fig. 4b. To tackle these issues, it is necessary to do an open operation on the image at least once to get rid of small particles or areas that are in the wrong place. Then dilate the image at least once, use the "fill holes" command if necessary, and erode if needed. The obtained binary image (shown in Fig. 4c) should be the segmentation of the tumors inside the liver.

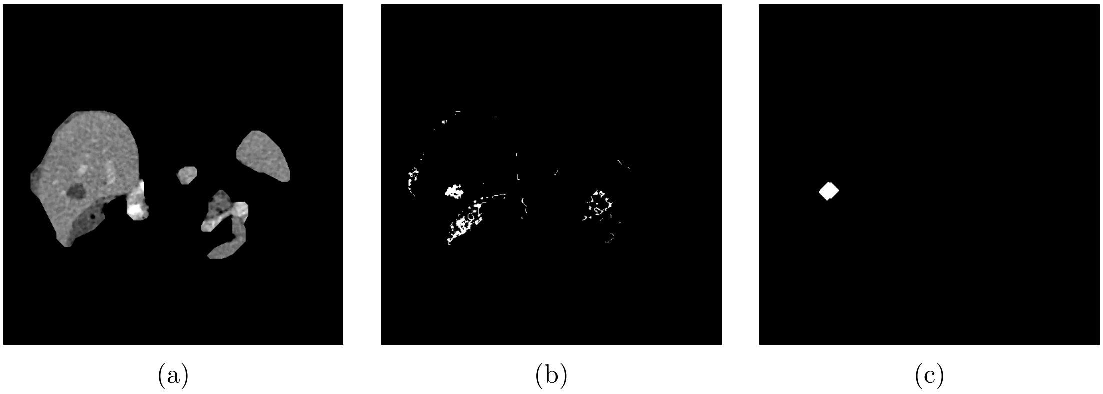

(a) (b) (c)

Figure 4: Tumor segmentation process: application of the liver mask to the whole image (a), thresholding, which reports some areas as tumors even though they are not (b), and final tumor segmentation result (c).

The main problem with this approach is the fact that automatization is very difficult: each scan is different from the others because it is of different patients and made with different machines. Moreover, this method is not time efficient and fully depends on the person working on the segmentation. It also requires knowledge of human anatomy since the operator has to be able to differentiate between liver and other tissues, other than knowing the aspect of a tumor in a CT scan. Lastly it is not a given that both the liver and its tumors will have different gray levels from those of other organs, further complicating thresholding.

2.2 Weka

This approach makes use of the Trainable Weka Segmentation 3D plugin [6] within Fiji [5], which is capable, after training, to produce a model able to detect either the liver or its tumors. Weka (Waikato Environment for Knowledge Analysis) contains a collection of visualization tools and algorithms for data analysis and predictive modeling, together with graphical user interfaces for easy access to this functionality and has as its main goal to work as a bridge between the Machine Learning and the Image Processing fields. To do that Weka offers a variety of selectable features that will be extracted during training (see Fig. 5), with some concentrating on finding edges and some concentrating on extracting texture information.

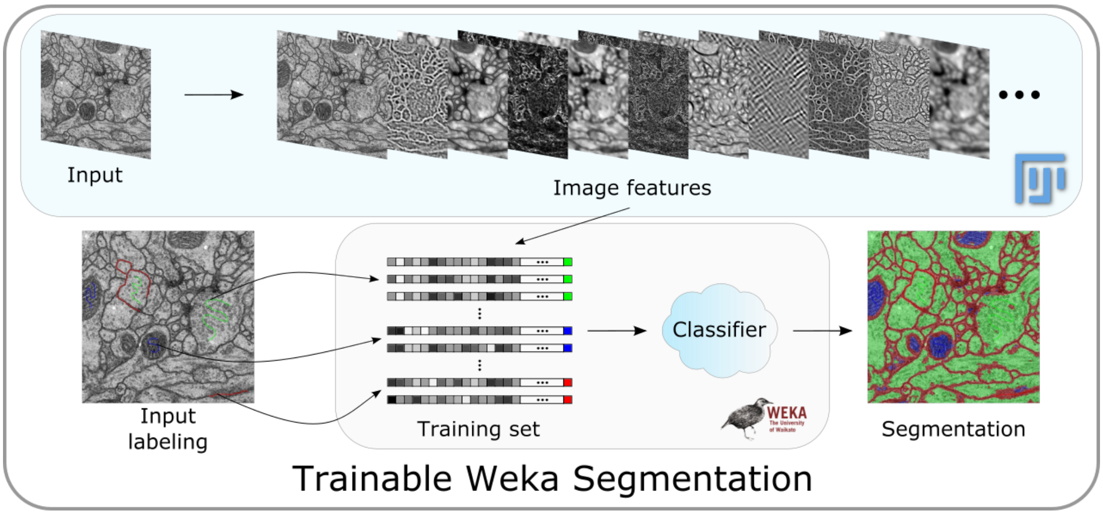


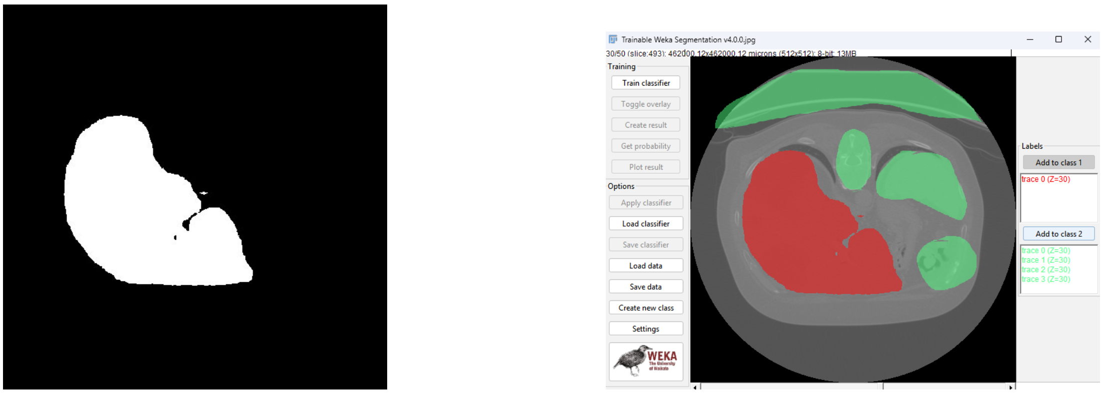
Figure 5: Architecture of Weka.

The Weka approach follows the idea used in the ImageJ approach, meaning that on a test CT scan two different Weka models will be applied: the first (Weka Liver Model) able to segment the liver in order to produce liver masks, the second (Weka Tumor Model) able to find, after the application of liver masks on the CTs, the tumors inside.
For this approach a pipeline was developed using macros in order to make both the preprocessing and the training as fast and as uniform as possible, which permitted the models to be trained on a great number of CT scans with minimal operator input.

Preprocessing For preprocessing in Fiji for the Weka segmentation, all CT volumes were first converted to 8-bit in order to standardize the image format and reduce file size. In addition, the spatial unit of the images was set to pixels, since this is the format returned by Weka and is therefore relevant for subsequent evaluation. To further reduce computational load, all slices that did not contain liver tissue were removed. While this introduces a methodological bias and does not reflect a realistic clinical setting, it was necessary due to limited computational resources in order to retain a sufficient number of relevant slices. Subsequently, the datasets, originally containing approximately 600–900 slices depending on the CT, were resampled to obtain about 20 regularly spaced slices. This resampling procedure was applied both to the CT images and to the provided ground-truth segmentations (GTS). Finally, the GTS were binarized, either into liver versus background or tumor versus background, so that they could be directly used as training labels for the segmentation model.

Weka Liver Model Training For the training process of the Liver Model, the binarized liver GTS were first stored as regions of interest (ROIs) and provided to Weka as class 1 objects (see Fig. 6a). In a next step, a macro was implemented to automatically select background tissue within a defined grayscale range, remove regions containing liver tissue, and pass the resulting structures to Weka as class 2 objects. However, this approach resulted in poorly connected and weakly structured regions, which limited their suitability as meaningful background representations. Although this issue could potentially be addressed with more complex preprocessing, the high computational cost and the impracticality

of training multiple models prevented such an approach. Instead, repeated manual refinement was performed, where incorrectly classified organs were explicitly labeled as background and used to iteratively retrain the model until it consistently and reliably identified liver tissue.
During this phase particular attention was paid to the feature selection and Hessian, Laplacian and Edges were selected as edges features while Minimum, Maximum, Median, Mean and Variance as texture features. It was found that it was not determinant for the model performance the specific features chosen, as long as an equal number from the two types was selected.
The interface used during manual training is illustrated in Fig. 6b.



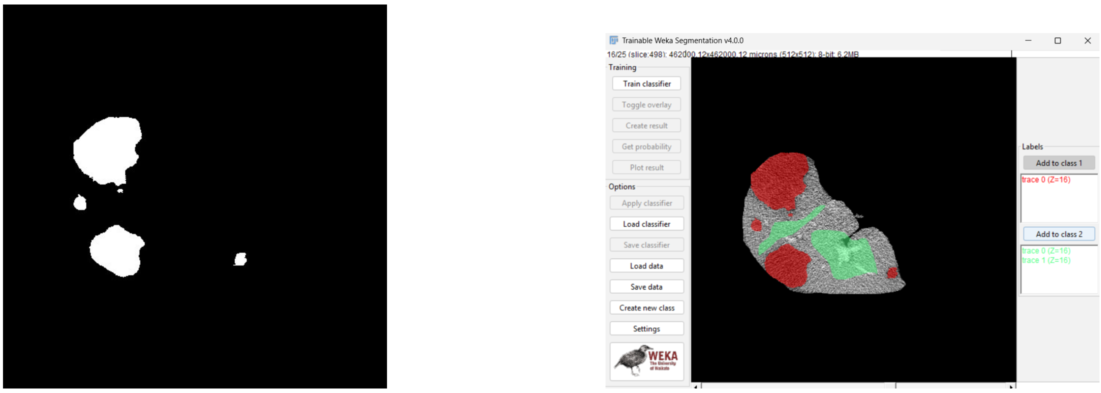
(a)

(b)

Figure 6: Mask slice obtained from the liver GTS used to initialize class 1 objects (a) and Weka interface (b), where the red region is the liver and the green ones are the background selected by the user.

The training took place on the CTs 12–17. The final resulting model was then applied to the CT scans (18–48) for evaluation. Not all CTs were evaluated due to computational and time limitations.

Weka Tumor Model Training For the training process of the Tumor Model, a further preprocessing step was needed: the application of the liver masks to to the CT scans in order to avoid searching for liver tumors outside of the liver. While this should have been done using the segmentations produced by the Weka Liver Model, in order to train both models at the same time, the binarized liver GTS were used instead.
Then the binarized tumor GTS were stored as region of interest (ROIs) and provided to Weka as class 1 objects, while the objects to pass as class 2 were selected manually like during the Weka Liver Model Training (Fig. 7).



(a) (b)

Figure 7: Mask slice obtained from the tumor GTS used to initialize class 1 objects (a) and Weka interface (b), where the red region is the tumor and the green ones are the background selected by the user.

The training took place iteratively on the CTs 4-8-18-26 but unfortunately the resulting model was not able to identify tumor tissues. The final model was therefore not applied on further CT scans, and given the impossibility to evaluate the metrics in Section 3, the model is not discussed in Section 4.

Evaluation For the evaluation of the Weka results from the Weka Liver Model, the obtained segmentations for each CT scan were first binarized and stored as regions of interest (ROIs). These predicted ROIs were then once intersected and once united with the corresponding ground-truth segmentations. For each case, the areas of the predicted segmentation, the ground-truth segmentation, as well as their intersection and union were determined. Based on these values, the evaluation functions introduced in Section 3 were implemented and computed. These metrics quantify the quality of the predictions and are subsequently analyzed and discussed in Section 4.

2.3 Neural Network

In order to solve the automatization issue of the ImageJ approach, another method was implemented, which makes use of neural networks. In particular, a 2D densely-connected UNet was utilized. A UNet is a special kind of convolutional neural network first introduced by Olaf Ronneberger in 2015 [7]. UNets have been widely used in medical image segmentation because of their improved performance compared to other methods in segmentation tasks and quantitative lesion analysis.

The general architecture of a UNet, shown in Fig.8, is composed of two sections: the first one is called encoder and through multiple convolutions and pooling operations it extracts image features, that are stored in the feature maps.

The second section of a UNet architecture is called decoder, it increases the image dimensions back to the original through upsampling. This is performed with a type of convolution called transposed convolution, where the previously generated feature maps are now used as kernels to produce a new image of larger dimensions that retains the details of the original image. At the end of this procedure the final output is an image, with the same dimensions of the original one, where each pixel value corresponds to its probability to belong to a certain class, such as object or background.

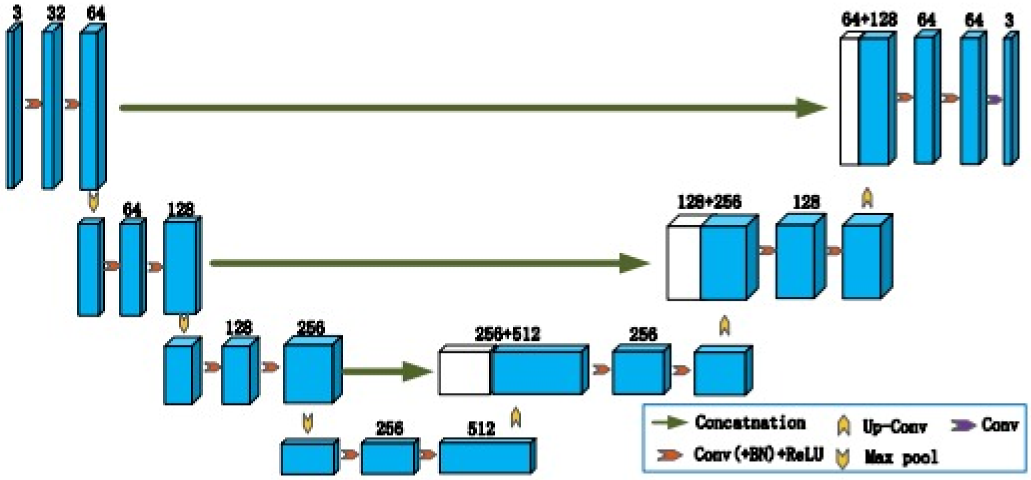


Figure 8: Architecture of a UNet. [8]

In this work, we used the Python's library PyTorch. This library allows to implement the type of neural network mentioned before (2D densely-connected UNet). Moreover, we decided to follow a cascaded approach, meaning that we implemented two models: one to segment the liver and the second to segment the tumor inside the liver based on the segmentation generated by the first model.
Before doing so, we needed to pre-process the three-dimensional scans at our disposal; this is common for deep learning to reduce memory usage and ensure consistency in the numeric type. What we did was to apply a min-max normalization that ensure all image intensities are within the range [0,1]. The utilized formula is (1).

$$\mathrm{I}' = \frac{\mathrm{I} - \min}{\max - \min + \epsilon} \tag{1}$$

In the latter, the minimum value is subtracted from each intensity in order to make the minimum value of the normalized image equal to zero, then the division by the intensity range allows to cap the new highest value to one. The ϵ has a very small value to make sure that in the rare case the maximum and minimum value coincide there won't be any division by zero.
After this preprocessing we built the UNet model using 3×3 kernels for the convolutions and the max pooling as pooling operation. This is one of the most commonly used operations. It reduces the spatial dimensions of the image by dividing the image in small windows, e.g. 2×2, and taking the maximum value inside it, so these windows will now be the pixels of the reduced images, for windows of 2×2 dimension the image width and height are cut in half.
Then we trained the two models on all 131 volume scans; since we are using a cascaded approach the model that will have to segment the liver is trained on the whole scans, while the tumor model is trained only on the liver region, obtained from the ground truth mask. The training ran on twenty epochs and took around fifty minutes for each model using a NVIDIA GeForce RTX 3070 GPU. It is important to notice that inside our model we used two dimensional operations to reduce the computational cost, that otherwise would have been too high to handle.

3 Segmentation Evaluation Metrics

To quantitatively assess the agreement between an automatically generated segmentation and a reference segmentation, several overlap-based and surface-based evaluation metrics are employed. Let $\hat{A}$ denote the automatically generated segmentation and A the reference segmentation. Both segmentations are interpreted as sets of voxels belonging to the object of interest.

The success of a segmentation is evaluated via the following metrics.

3.1 Intersection over Union

The Intersection over Union (IoU), also referred to as Jaccard Index, quantifies the overlap between two sets relative to their union and is defined as

$$\mathrm{IoU}(A, \hat{A}) = \frac{|A \cap \hat{A}|}{|A \cup \hat{A}|}. \tag{2}$$

The IoU also ranges from 0 to 1, with higher values indicating better segmentation quality. Compared to the other metrics such as Dice coefficient, the IoU penalizes mismatches more strongly and therefore provides a stricter measure of overlap.

3.2 Dice Similarity Coefficient

The Dice Similarity Coefficient (DSC) measures the volumetric overlap between two segmentations and is defined as

$$\mathrm{Dice}(A, \hat{A}) = \frac{2|A \cap \hat{A}|}{|A| + |\hat{A}|}. \tag{3}$$

The Dice coefficient takes values in the range $[0, 1]$, where a value of 1 indicates perfect agreement and a value of 0 indicates no overlap. The numerator represents twice the number of correctly overlapping voxels, while the denominator corresponds to the total number of voxels in both segmentations.

3.3 Volume Overlap Error

The Volume Overlap Error (VOE) is derived from the IoU and evaluates the disagreement between two segmentations. It is defined as

$$\mathrm{VOE}(A, \hat{A}) = 1 - \mathrm{IoU}(A, \hat{A}). \tag{4}$$

The VOE takes values in the interval $[0, 1]$, where lower values correspond to better overlap. A VOE of zero indicates perfect agreement between the predicted and reference segmentations.

3.4 Relative Volume Difference

The Relative Volume Difference (RVD) evaluates the relative difference in volume between the automatically generated segmentation and the reference segmentation and is defined as

$$\mathrm{RVD}(A, \hat{A}) = \frac{|A| - |\hat{A}|}{|\hat{A}|}. \tag{5}$$

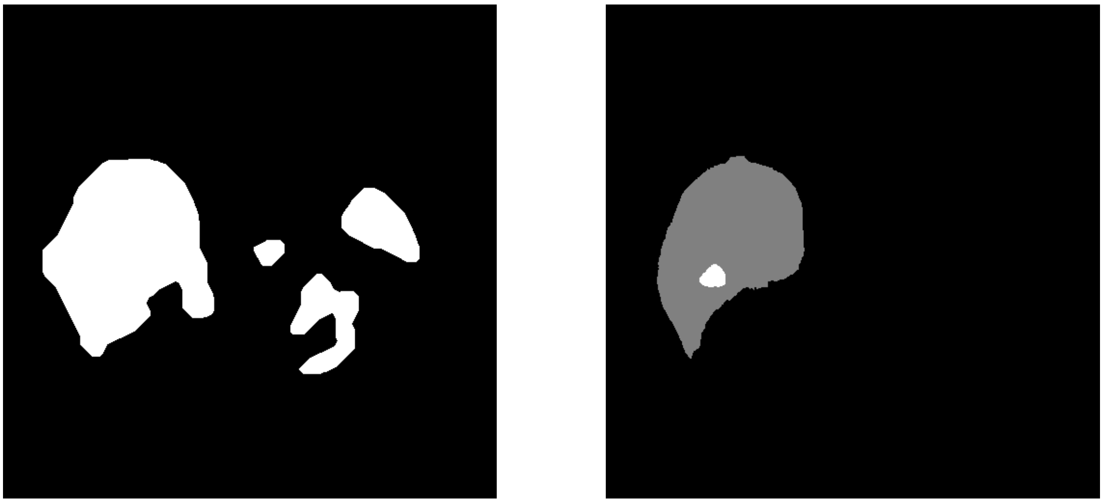
This metric can assume values in the range $(-\infty, \infty)$. A positive RVD indicates under-segmentation, meaning that the automatic segmentation contains less voxels than the reference, whereas a negative RVD indicates over-segmentation. An RVD of zero implies identical volumes, independent of spatial overlap.

4 Results

In this section, the results of the three evaluated segmentation approaches are presented and discussed. First, the ImageJ-only workflow is examined to establish a baseline performance based on classical image processing techniques. Subsequently, the outcomes of the Weka-based machine learning approach are analyzed, highlighting its capabilities and limitations. Finally, the performance of the neural network model is evaluated, enabling a direct comparison between traditional, machine learning, and deep learning methods with respect to segmentation accuracy and robustness which will be discussed in the next section.
The macros used for the Weka preprocessing, the results from the application of the Weka Liver Model on the CT scans 18-48 and the neural network code used for this report are stored in this GitHub repository: `https://github.com/giacomells/LiTS-challenge`

4.1 ImageJ

Given the exploratory nature of the method, the analysis was conducted on a limited number of slices, applying the method to 10 random CT scans of different patients. The number chosen is adequate to evaluate the method's feasibility, offer some statistical relevance and pinpoint any potential critical issues.
The scans chosen from each CT scan were not random since only scans in which the liver was present were selected (again, it is crucial to know human anatomy in order to know where to look at). The metrics are evaluated by comparing the manually obtained segmentations with the ones given by the LiTS challenge. As mentioned earlier this method can sometimes give rise to problems like the one evident in Fig. 9, which consists in wrongly segmenting as liver other organs and tissue that have similar gray levels.

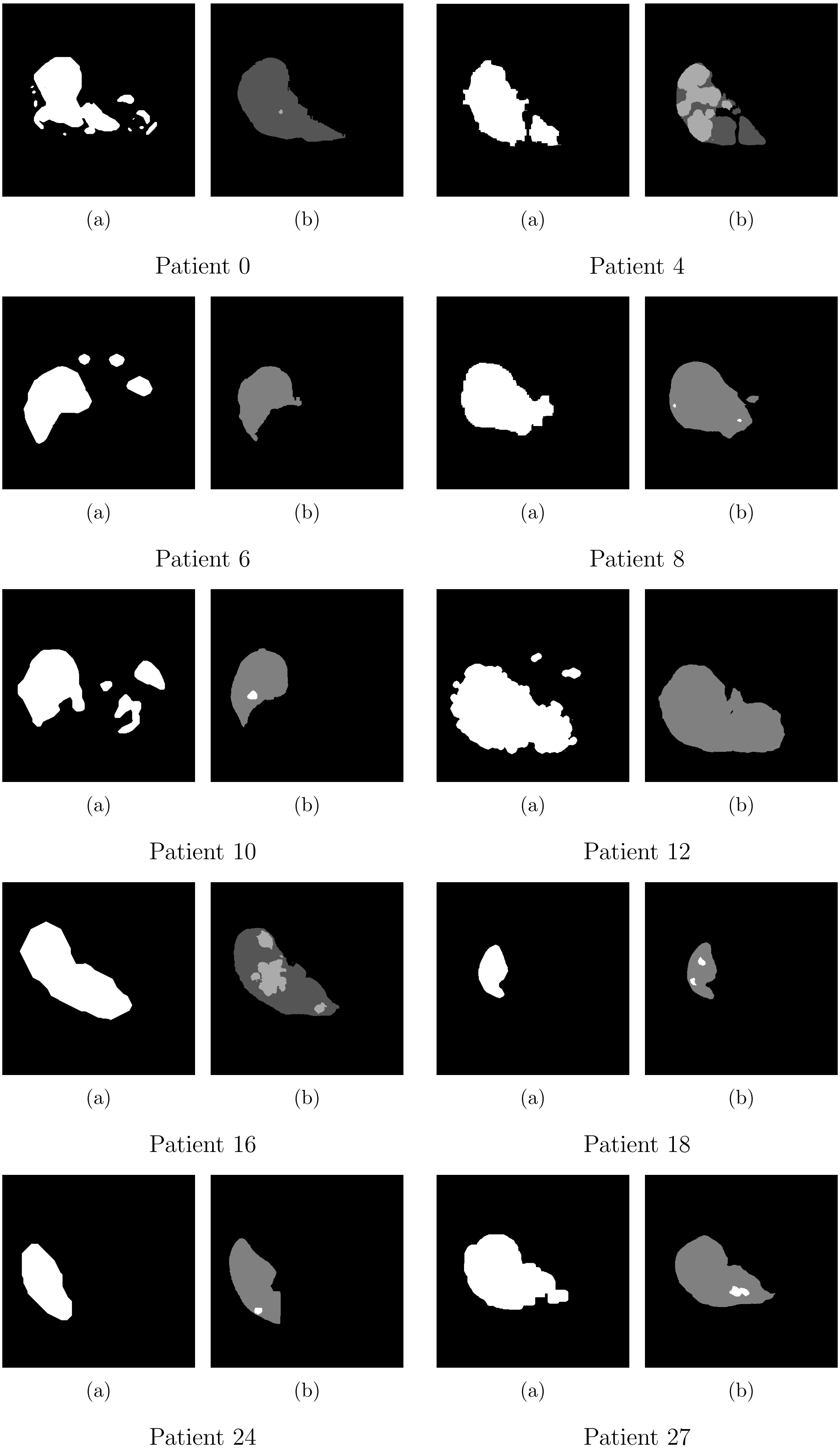


(a) liver in white

(b) liver in gray, tumor in white

Figure 9: Due to the imperfect thresholding other tissues besides the liver can be picked up. By confronting the liver segmentation obtained on slice 380 of patient 10 (a) with the ground truth segmentation (b), it is visible that both spleen and part of the intestines have been wrongly segmented as liver.

This problem arises when in a CT scan many different tissues and organs have similar gray level, which makes hard excluding the ones that are not liver. Since the liver is usually the biggest area after thresholding, a partial solution found to this problem is the opening operation which removes small isolated areas. This operation may be done many times depending on the image.
The 10 liver segmentations obtained with this method are shown in Fig. 10, together with corresponding ground truth segmentations for comparison.



(a) (b) (a) (b)

Patient 0 Patient 4

(a) (b) (a) (b)

Patient 6 Patient 8

(a) (b) (a) (b)

Patient 10 Patient 12

(a) (b) (a) (b)

Patient 16 Patient 18

(a) (b) (a) (b)

Patient 24 Patient 27

Figure 10: Liver segmentation results for all patients: comparison between our liver segmentation in white (a) and liver GTS in dark gray (b).

The metrics calculated to estimate the accuracy of the method for liver segmentation are shown in Table 1.

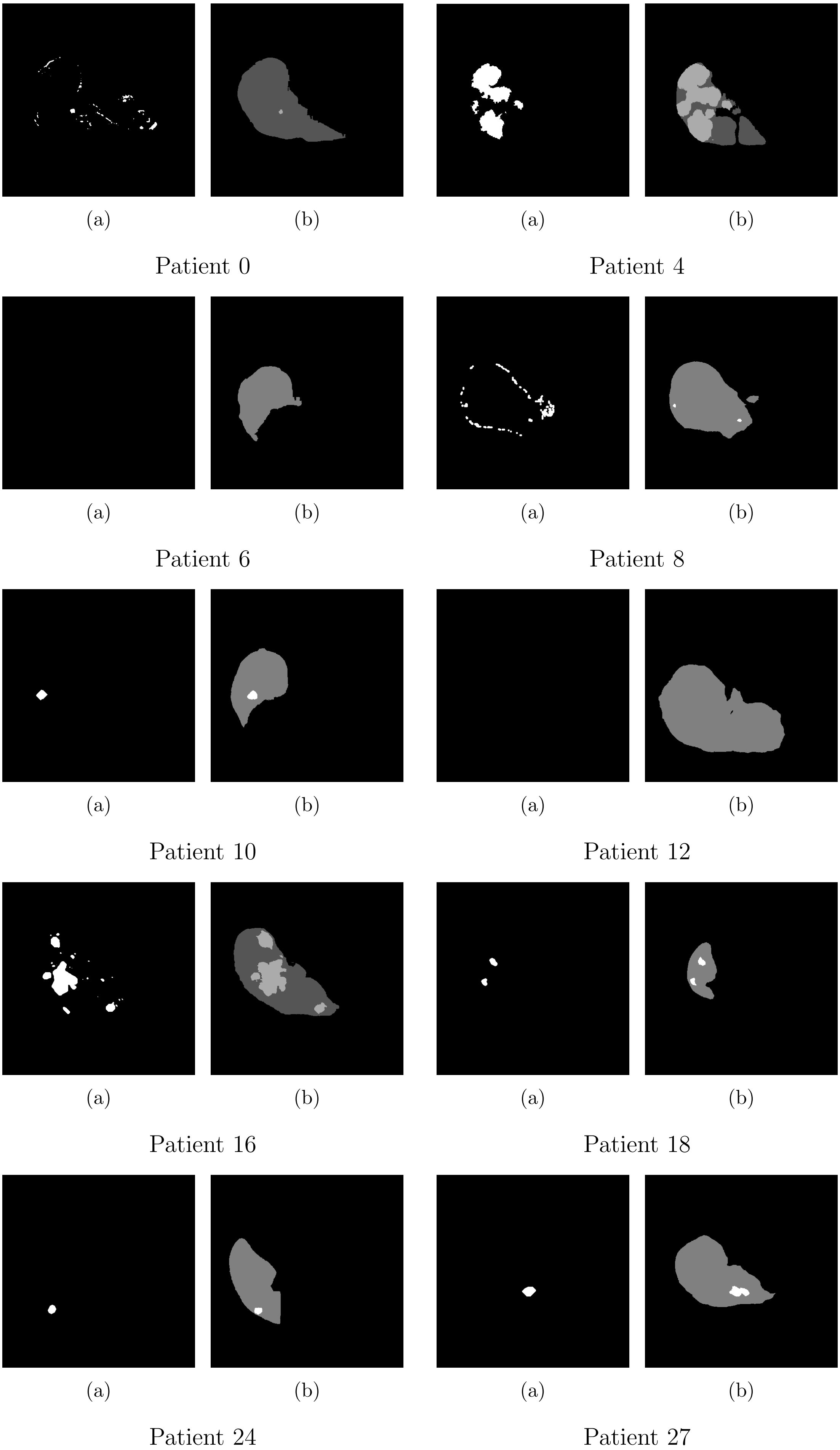
Patient no.	IoU	Dice	VOE	RVD
Patient 0	0.6353	0.7770	0.3647	+0.2461
Patient 4	0.8129	0.8968	0.1871	-0.0307
Patient 6	0.6875	0.8148	0.3125	-0.2844
Patient 8	0.9023	0.9487	0.0977	-0.0213
Patient 10	0.5774	0.7321	0.4226	-0.3947
Patient 12	0.8729	0.9321	0.1271	-0.0140
Patient 16	0.8371	0.9113	0.1629	-0.0367
Patient 18	0.8877	0.9405	0.1123	+0.1084
Patient 24	0.8401	0.9131	0.1599	+0.1774
Patient 27	0.8177	0.8997	0.1823	-0.1465
Average	0.79 ± 0.11	0.88 ± 0.07	0.21 ± 0.11	$+0.0 \pm 0.2$

Table 1: All the liver segmentations are considered successful since the IoU is greater than 0.5

Where for the average IoU, Dice, VOE and RVD metric and the corresponding errors these formulas have been used:

$$\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i, \qquad \sigma_{\bar{x}} = \sqrt{\frac{\sum_{i=1}^{N}(x_i - \bar{x})^2}{N-1}} \tag{6}$$

where $\bar{x}$ is the average, x_i is the i-th value, N is the number of samples (in this case 10) and $\sigma_{\bar{x}}$ is the standard deviation.
The 10 tumors segmentations obtained are instead showed in Fig. 11 together with corresponding ground truth segmentations for comparison.



(a) (b) (a) (b)

Patient 0 Patient 4

(a) (b) (a) (b)

Patient 6 Patient 8

(a) (b) (a) (b)

Patient 10 Patient 12

(a) (b) (a) (b)

Patient 16 Patient 18

(a) (b) (a) (b)

Patient 24 Patient 27

Figure 11: Tumor segmentation results for all patients: comparison between our tumor segmentation in white (a) and tumor GTS in dark gray (b).

The metrics calculated to estimate the accuracy of the method for tumor segmentation are shown in Table 2.

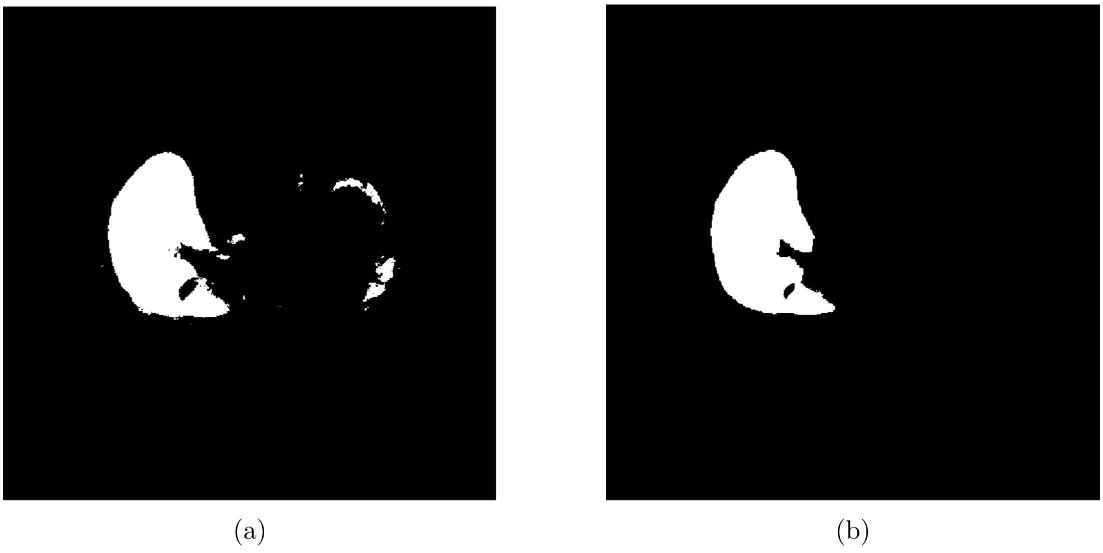
Patient no.	IoU	Dice	VOE	RVD
Patient 0	0.0095	0.0188	0.9905	-0.9905
Patient 4	0.7176	0.8356	0.2824	+0.3883
Patient 6	NaN	NaN	NaN	NaN
Patient 8	0.0494	0.0941	0.9506	-0.9373
Patient 10	0.4320	0.6033	0.5680	-0.1417
Patient 12	NaN	NaN	NaN	NaN
Patient 16	0.6406	0.7809	0.3594	+0.3425
Patient 18	0.7242	0.8401	0.2758	+0.0000
Patient 24	0.3968	0.5881	0.6032	-0.1896
Patient 27	0.6328	0.7751	0.3672	+0.2669
Average	0.5 ± 0.3	0.6 ± 0.3	0.5 ± 0.3	-0.2 ± 0.5

Table 2: Tumor segmentations of patients 0, 8, 10 and 24 are not successful since their IoU is smaller than 0.5. NaN means that the metrics are undefined since there was no tumor to be detected for patients 6 and 12.

Giving us an average IoU of (0.5 ± 0.3). Since we are working on smaller objects and using a liver mask containing errors from the previous liver segmentation the bigger error is justifiable and on average, we can say both quantitatively and qualitatively that liver segmentation is more accurate than tumor segmentation.

4.2 Weka

The evaluation was conducted using CT scans 18–48, and only on the Weka Liver Model. All predictions achieved an Intersection over Union (IoU) greater than 50 %, which qualifies them as successful segmentations according to the performance criterion applied. For illustrative purpose a slice from the CTs resulting from the Weka liver segmentation is presented in Fig. 12.



(a)

(b)

Figure 12: Weka's Liver Model segmentation result (a) is showed beside the liver GTS on (b) for Slice 11 of CT 18.

The liver part is very accurately predicted, however a little bit of other tissue is falsely identified as also being part of the liver and further attempts at refining the model proved unsuccessful since some organs, like the spleen, simply have both similar shape and similar gray level.
The metrics shown in Tab. 3 are averages over all evaluated CTs and the error is the arising standard deviation of measurements.

Metric	Value
IoU	0.82 ± 0.07
Dice	0.90 ± 0.05
VOE	0.18 ± 0.08
RVD	0.09 ± 0.12

Table 3: Weka Results

The obtained metrics indicate strong segmentation performance. The mean IoU of 0.82 and Dice coefficient of 0.90 reflect a high degree of spatial overlap between the predicted and ground-truth segmentations. The Volumetric Overlap Error (VOE) of 0.18 is correspondingly low, confirming limited mismatch between volumes. The Relative Difference in Volume (RVD) of 0.09 shows that the predicted volumes are generally close to the reference volumes, although some variability remains, as reflected by the reported standard deviation. Overall, the results demonstrate reliable and accurate segmentation behavior across the evaluated CTs.

4.3 Neural Network

The NN training ran stably and showed clear convergence: training loss decreased steadily as shown in Fig. 13. It is possible to see that, although the training loss function decreases with the epochs, the validation loss function oscillates indicating that neither UNet is working properly.

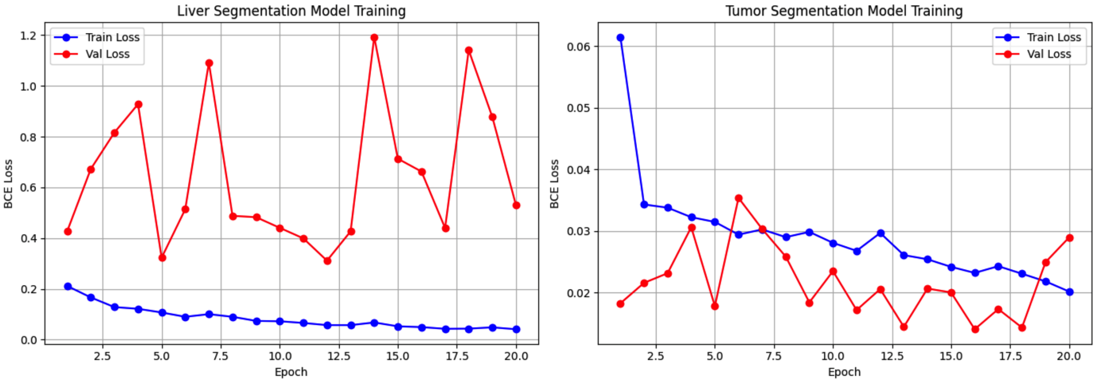


Figure 13: Training loss function and Validation loss function for both liver and tumor models.

As it can be seen in the graph the training converges but the model was not able to maintain the same performance on the validation set; the same goes for the test scans, on which we applied the model, that yielded extremely poor results. This means we couldn't analyze the results quantitavely, even if from a qualitative point of view some predictions align well with the anatomy, capturing organ boundaries and lesions as shown in Fig. 14 but for most of the cases, the model does not work properly.

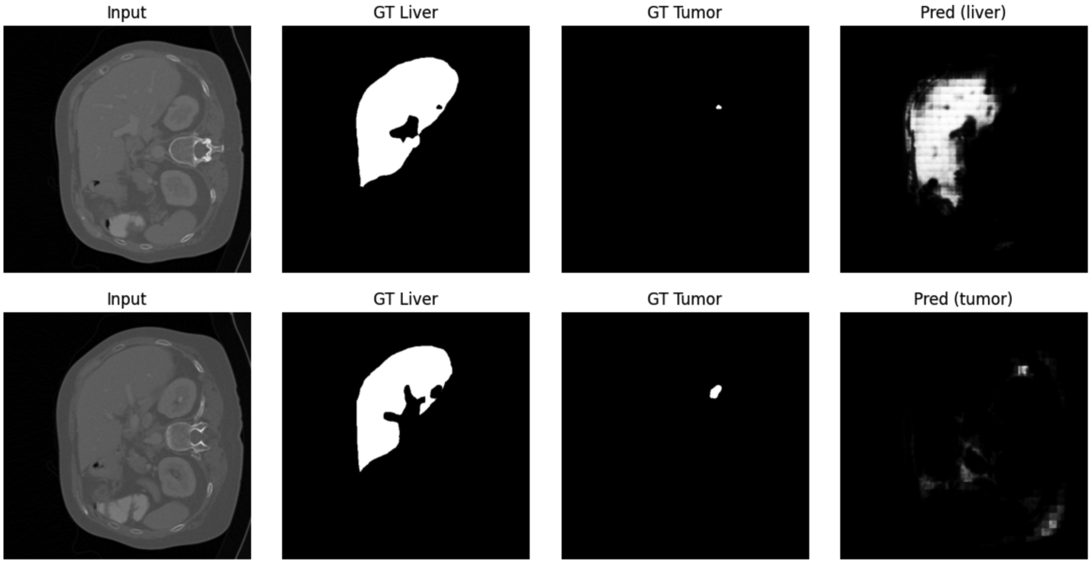


Figure 14: In the first row the prediction of the Liver model is shown and in the second row it's shown the tumor model. This images were obtained from patient number 126.

This is probably due to the fact that the model was structured on a 2D UNet instead of hybrid UNet integrating both 2D and 3D convolutions.

5 Comparison & Conclusion

A qualitative comparison in this work can be conducted exclusively for liver segmentation between the ImageJ and Weka approaches, as tumor segmentations are not available for the Weka method. Based on the evaluated metrics shown in Table 1 and Table 3, Weka consistently shows slightly higher accuracy and reliability. Specifically, the mean IoU for Weka is 0.82 ± 0.07, compared to 0.79 ± 0.11 for ImageJ. The Dice coefficient follows a similar trend with 0.90 ± 0.05 for Weka and 0.88 ± 0.07 for ImageJ. Volumetric Overlap Error (VOE) is lower for Weka (0.18 ± 0.08) than for ImageJ (0.21 ± 0.11), indicating better agreement with the ground truth, while the Relative Volume Difference (RVD) shows a slightly higher value but lower variance for Weka (0.09 ± 0.12) compared to ImageJ (0.0 ± 0.2), but still within acceptable limits. These results suggest that Weka provides a slightly more robust and consistent liver segmentation performance.

Since no quantitative data was collected for the neural network approach, the following discussion focuses on a qualitative comparison between the three methods.

In summary, the three segmentation approaches demonstrate distinctly different strengths and limitations. The pure ImageJ-based method achieved locally good results but lacks robustness and scalability, making it unsuitable for broader application. The Weka-based approach produced some of the best outcomes and was evaluated on the largest dataset, indicating strong potential. Although further optimization is possible, particularly with respect to automated background selection and automated retraining, the manually refined training already resulted in reliable segmentations across a considerable number of 31 CT scans. At this stage, the limitations imposed by computational resources became especially apparent, since the number of CT slices had to be significantly reduced to enable training on standard hardware. Considering these constraints, the obtained performance is notable, even though further improvement remains feasible. By contrast, the deep learning approach using a neural network, initially expected to be the most promising, ultimately failed to achieve satisfactory results. Here, computational limitations were even more critical, as individual training runs could require up to twelve hours. Significantly greater computational capacity and time would be required to develop this approach into a truly applicable solution, which was beyond the scope of this work.

Concluding the following table illustrates qualitatively the advantages and disadvantages of each approach.

Criterion	ImageJ	Weka	Neural Network
Performance	good	very good	bad
Stability	bad	good	good
Scalability	very bad	good	very good
Computational Cost	low	high	very high

Table 4: Comparison of segmentation approaches

6 Further Ideas

In this section will be discussed further ideas that would have been interesting to research if more time and especially computational resources had been given.

6.1 ImageJ

To have a more general idea of the method's accuracy it is desirable to have different people apply the method on more than 10 scans, since this method could give different results depending on the choices made by the person applying it.

6.2 Weka

Future work would focus on training and testing the model on entire CT volumes without clipping and resampling. Indeed, clipping the scans to only include slices containing the liver region introduces a form of bias in the model because it makes use of information about the target anatomy which the model would not have access to. Additionally, resampling removes a considerable part of the liver scans increasing the spacing between slices and depriving the Weka Models of 3D information that would have been used to better track the changes between slices. Both these operations were needed to avoid overloading the system but, with increased computational resources, all slices could be used, which would likely improve the robustness and generalization of the model.

Furthermore, segmentations of surrounding organs could be integrated to provide the model with anatomically meaningful background context during training.

Additional preprocessing steps such as removing irrelevant background structures and trying to make different CTs have the same image contrast and brightness may further improve feature visibility, since the model struggled particularly on unusually bright or dark CT scans.

Data augmentation techniques, including geometric distortions of the CT scans, could be applied to increase dataset variability and expand the training pool. Finally, introducing penalties for small, disconnected prediction volumes may reduce noise and prevent false positive segmentations.

6.3 Neural Network

An improvement of the neural network approach, as discussed in Section 4, can be the implementation of a hybrid network, that combines 2D and 3D convolutions (H-DenseUNet [9]). This type of model provides higher precision because it is able to keep track of intra-slice features, while keeping the training time relatively low. In fact, this method was used by one of the research teams that won the LiTS challenge [3].

References

[1] J. Ferlay et al. “Estimates of worldwide burden of cancer in 2008: GLOBOCAN 2008”. In: *International Journal of Cancer* 127.12 (2010), pp. 2893–2917. DOI: 10.1002/ijc.25516.

[2] M. Moghbel et al. “Review of liver segmentation and computer assisted detection/diagnosis methods in computed tomography”. In: *Artificial Intelligence Review* (2017), pp. 1–41. DOI: 10.1007/s10462-017-9552-z.

[3] CodaLab Competitions. *LiTS - Liver Tumor Segmentation Challenge.* last visited 13. December 2025. 2017. URL: https://competitions.codalab.org/competitions/17094.

[4] Caroline A Schneider, Wayne S Rasband, and Kevin W Eliceiri. “NIH Image to ImageJ: 25 years of image analysis”. In: *Nature methods* 9.7 (2012), pp. 671–675.

[5] Johannes Schindelin et al. “Fiji: an open-source platform for biological-image analysis”. In: *Nature methods* 9.7 (2012), pp. 676–682.

[6] Ignacio Arganda-Carreras et al. “Trainable Weka Segmentation: a machine learning tool for microscopy pixel classification”. In: *Bioinformatics* 33.15 (2017), pp. 2424–2426.

[7] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. *U-Net: Convolutional Networks for Biomedical Image Segmentation.* 2015. arXiv: 1505.04597 [cs.CV]. URL: https://arxiv.org/abs/1505.04597.

[8] Wang Jiangtao, Nur Intan Raihana Ruhaiyem, and Fu Panpan. *A Comprehensive Review of U-Net and Its Variants: Advances and Applications in Medical Image Segmentation.* 2025. arXiv: 2502.06895 [eess.IV]. URL: https://arxiv.org/abs/2502.06895.

[9] Xiaomeng Li et al. “H-DenseUNet: hybrid densely connected UNet for liver and tumor segmentation from CT volumes”. In: *IEEE transactions on medical imaging* 37.12 (2018), pp. 2663–2674. URL: https://arxiv.org/pdf/1709.07330.