mathematics

MDPI

Article

A Lightweight Deep Learning Approach for Liver Segmentation

Smaranda Bogoi * and Andreea Udrea

Department of Automatic Control and Systems Engineering, Faculty of Computer Science and Automatic Control, University "Politehnica" of Bucharest, 060042 Bucharest, Romania
* Correspondence: smaranda.bogoi@stud.acs.upb.ro

Abstract: Liver segmentation is a prerequisite for various hepatic interventions and is a time-consuming manual task performed by radiology experts. Recently, various computationally expensive deep learning architectures tackled this aspect without considering the resource limitations of a real-life clinical setup. In this paper, we investigated the capabilities of a lightweight model, UNeXt, in comparison with the U-Net model. Moreover, we conduct a broad analysis at the micro and macro levels of these architectures by using two training loss functions: soft dice loss and unified focal loss, and by substituting the commonly used ReLU activation function, with the novel Funnel activation function. An automatic post-processing step that increases the overall performance of the models is also proposed. Model training and evaluation were performed on a public database—LiTS. The results show that the UNeXt model (Funnel activation, soft dice loss, post-processing step) achieved a 0.9902 dice similarity coefficient on the whole CT volumes in the test set, with 15× fewer parameters in nearly 4× less inference time, compared to its counterpart, U-Net. Thus, lightweight models can become the new standard in medical segmentation, and when implemented thoroughly can alleviate the computational burden while preserving the capabilities of a parameter-heavy architecture.

Keywords: CT liver segmentation; lightweight neural network; LiTS; automatic post-processing

MSC: 68T07; 68U10; 92C55

Citation: Bogoi, S.; Udrea, A. A Lightweight Deep Learning Approach for Liver Segmentation. *Mathematics* **2023**, *11*, 95. https://doi.org/10.3390/math11010095

Academic Editor: Konstantin Kozlov

Received: 17 November 2022
Revised: 19 December 2022
Accepted: 22 December 2022
Published: 26 December 2022

1. Introduction

Cancer is one of the challenges of modern times as it is one of the prevalent causes of premature death worldwide. According to GLOBOCAN 2020 [1], produced by the International Agency for Research on Cancer (IARC) and available online at the Global Cancer Observatory [2], in 2020, about 19.3 million people were estimated to be diagnosed with cancer and nearly 10 million died. Liver cancer accounted for 4.7% of all new cases, being the sixth most common cancer and the third cause of cancer mortality, with 8.3% of the total new deaths. Furthermore, according to the World Cancer Report 2020 [3], there was a 75% increase in incident liver cancer cases worldwide from 1990 to 2015. This incidence increase is translated in higher health-care costs and accentuates the importance of finding solutions to reduce not only the expenses, but also to improve medical workflow efficiency (e.g., using (semi)automatic algorithms for mundane tasks, such to allow the medical stuff to concentrate on more difficult tasks [4–6]).

1.1. Liver Segmentation and Its Relevance

Liver and liver lesion segmentation is essential in tumor staging and a key prerequisite for many surgical and radiological interventions, such as liver transplant, immunotherapy, embolization therapy, surgical resection, and ablation therapy [5,6]. Furthermore, surgical liver resection and liver transplant planning are complex tasks, and liver volumetry is needed to ensure donor–recipient compatibility. This is obtained through liver segmentation from abdominal CT scans. Moreover, the complexity of this task is of interest to

researchers because the implemented segmentation methods should provide accurate contours despite the organ's unclear boundaries due to adjacent organs or the heterogeneity of its structure (especially in cases presenting tumors or fibrotic areas). The gold standard in clinical applications for liver segmentation is still a manual strategy that is time consuming and prone to subjectivity (large inter-/intra-observer variability) [5]. Thus, automatic or semi-automatic segmentation of the liver could increase the efficiency of the procedure and diminish the time spent by the medical personnel in order to perform this laborious task.

1.2. State of the Art on Neural Networks for Liver Segmentation

In the past years, a plethora of papers concentrated on liver segmentation have been published, involving automatic methods such as deep learning algorithms. In 2017, the Liver Tumor Segmentation Benchmark (LiTS) [7] was provided by the ISBI and MICCAI as a conjoint challenge that addressed the task of automated liver and liver tumor segmentation. For the liver segmentation task, the proposed methods concentrated on using fully convolutional neural networks (FCNs) with U-Net-based resemblance as the core structure. The best result was obtained by a hierarchical framework built on deep fully convolutional–deconvolutional neural networks (CDNNs) that were trained sequentially [8]. Other notable approaches are a hybrid densely connected U-Net (H-DenseUNet) to appropriately extract high-level intra-slice features [9], an ensemble of three orthogonal 2D neural networks with a cascaded U-Net backbone on resampled axial, sagittal, and coronal [10] and similar cascaded approaches considering two stacked FCNs for liver and tumor lesions [11,12].

Since then, different attention mechanisms, convolutional layer types, and other modifications were added to the traditional U-Net baseline [13–19] and, also, generative adversarial network (GAN) approaches were of interest [14]. A light-weight hybrid convolutional network (LW-HCN) is presented in [13]. The authors aim at decreasing the 3D FCNs computational burden by using 2D depthwise convolutions in the bottom layers, a dimension transformation layer to bridge the 2D and 3D convolutions and a depth spatio-temporal separate convolutions (DSTS) factorization for 3D convolutions.

A framework combining the DeepLab-v3 architecture as the backbone for segmentation and the Pix2Pix network for the GAN implementation is presented in [14]. First, the segmentation is performed using the DeepLab-v3 for coarse segmentation. Then, the Pix2Pix architecture is imposed to learn the correlations between the ground-truth mask and the original image by using the Wasserstein adversarial-based loss function. The limitations of the liver and tumor boundary segmentations were considered in [15], concluding that pixels closer to the edges are more difficult to discriminate and it is important to focus more on that specific area. The authors propose a two-stage architecture, the first stage providing a coarse liver segmentation using a Res2Net-50 encoder for extraction of multi-scale features and a U-Net-like decoder.

An improved iRes-UNet framework was proposed in [16] by introducing normalization layers to eliminate internal covariate shift and residual paths to the modules to ease the convergence. The architecture was based on the V-Net and ResUnet and it was furthermore improved by providing a weighting map based on morphological features that can help in a better segmentation of liver edge pixels and of input images with small liver pieces. Drawbacks of U-Net were also presented in [17], where semantic information was enriched by using an EfficientNetB4 as an encoder and a U-Net-like decoder where the convolutional units were replaced with residual ones. Moreover, an attention gate was introduced to every skip connection to have a better focus on the relevant liver parts.

In [18], the proposed RMS-UNet improved the traditional U-Net by incorporating a 2.5D FCN approach with dilated convolutional layers in the convolutional units to enhance multi-slice context, generalized information, and temporal context. These units were followed by residual units for preserving the important semantic information and alleviating the vanishing gradient problem. Another 2.5D approach was conducted by [19] in which the architecture is based on the ResNeXt50 baseline with additional dilated

convolutions that were shown to improve the receptive field without altering the size of the feature maps. At the bottom of the encoder, a transformer approach is implied and with the help of attention blocks, the background information can be filtered, which leads to better global context information.

Even though 3D approaches use a large amount of computational resources, their advantage is the possibility to have temporal and contextual information that 2D neural networks lack. However, in [20], the authors proposed an architecture that tried to leverage the 3D neural networks' computational burden while maintaining the segmentation accuracy. The Res-PAC-UNet advantages are the constant feature maps width and residual connections that limit the memory consumption, while introducing Pyramid Atrous Convolution (PAC) modules through the skip-connections to improve the quality of feature transfer between the encoder and decoder, without duplicating low-level features. Another interesting aspect is that, while this approach is based on a 3D network, the number of parameters is significantly lower than in other 3D approaches, making this architecture a valuable starting point for future research that can concentrate on the trade-off between segmentation performance, memory, and time footprint.

With the aforementioned papers, it can be concluded that the U-Net backbone, namely, the encoder–decoder scheme, has become a standard for liver segmentation with different alterations to the architecture that could improve the network's performance. The majority adopted 2D or 2.5D neural networks due to limited computational burden but with the drawback that such approaches lose the contextual information that can be provided by a 3D approach. However, even in these conditions, the increase in their performance comes with the downside of parameter-heavy architectures.

A more in-depth overview of the results of the recent publication is depicted in Table 1, where the details about the evaluation metric, activation functions, loss functions, datasets, and post-processing are presented. In terms of the network's number of parameters and implicit computational efficiency, only the architectures presented in [13,16,20] have under 4 million parameters.

Table 1. Detailed results of state-of-the-art papers.

References	Dice Similarity Coefficient (Per Case)	Activation Functions Used in the Architecture	Loss Functions Used for Training	Datasets Used for Training	Datasets Used for Testing	Performed Post-Processing
Yuan et al., 2017 [8]	0.9670	ReLU	Function based on Jaccard Index (IoU)	LiTS train set	LiTS TEST set	Yes
Li et al., 2018 [9]	0.9610	ReLU	Weighted cross-entropy loss	LiTS train set	LiTS test set, 3DIRCADb	No
Chlebus et al., 2017 [10]	0.9600	ReLU	Dice loss	Private dataset, LiTS train set	LiTS test set	Yes
Vorontsov et al., 2017 [11]	0.9510	ReLU	Dice loss	LiTS train set	LiTS test set	No
Christ et al., 2017 [12]	0.943	ReLU	Weighted Cross-entropy loss	3DIRCADb, private dataset	3DIRCADb, private dataset	Yes
Zhang et al., 2019 [13]	0.9650	ReLU	Combined loss functions (dice loss, cross-entropy loss)	LiTS train set, 3DIRCADb,	LiTS test set 3DIRCADb,	No
Xia et al., 2019 [14]	0.970	ReLU	Hybrid loss function (region-based, context-based, and adversarial-based)	private dataset, LiTS train set	LiTS test set, private dataset	No

Table 1. *Cont.*

References	Dice Similarity Coefficient (Per Case)	Activation Functions Used in the Architecture	Loss Functions Used for Training	Datasets Used for Training	Datasets Used for Testing	Performed Post-Processing
Tang et al., 2022 [15]	0.968	ReLU	Combined loss (binary cross-entropy, Jaccard index loss)	LiTS train set	LiTS test set, 3DIRCADb	No
Lv et al., 2021 [16]	0.9424	ReLU	Weighted dice loss with morphologically based loss function	LiTS train set, SLIVER07	LiTS train set, SLIVER07	No
Wang et al., 2021 [17]	0.952	ReLU	Combined loss function (dice loss, binary cross-entropy loss)	SLIVER07, LiTS train set	LiTS test set SLIVER07, LiTS train set	No
Khan et al., 2022 [18]	0.9738	ReLU	Combined loss function (dice loss, absolute volumetric difference loss), Binary cross-entropy	LiTS train set, chaos, SLIVER07, 3DIRCADb	LiTS test set, chaos, SLIVER07, 3DIRCADb	No
Li et al., 2022 [19]	0.9338	ReLU	Cross-entropy loss	LiTS train set	LiTS train set	No
Ansari et al., 2022 [20]	0.9580	ReLU	Modified surface loss function	Medical decathlon Train set	Medical decathlon train set	No

From the context presented above, one can extract the two key elements, namely, performance in terms of accurate segmentation and performance in terms of computational efficiency with the obvious tradeoff between them. This paper focuses on casting some light on these two aspects and proposing a fast and accurate procedure for liver segmentation, and the main contributions of this work are as follows:

1. Testing a lightweight architecture, UNeXt, in the case of liver segmentation and comparing the results with the traditional U-Net architecture.
2. Empirical evaluation of two different loss functions while training the models, soft dice loss, and unified focal loss.
3. Modifying the two suggested architectures, U-Net and UNeXt, with respect to activation functions by replacing the commonly used ReLU with a novel function, Funnel.
4. Proposing an automatic post-processing filtering for misclassified non-liver regions.

2. Material and Methods

2.1. Investigated Architectures

In this paper, two architectures were tested on the liver segmentation task, namely, the traditional vanilla U-Net [21], proposed in May 2015, and the novel architecture, UNeXt [21], proposed in March 2022. While both were tested on several medical image segmentation tasks, with U-Net being a baseline in liver segmentation, the latter has not been tested as far as our knowledge on this specific task.

UNeXt [22] was motivated by the idea of point-of-care imaging, where the patient could benefit from accelerated analysis as a bedside procedure, even when there are constraints on computational resources. Thus, it leverages the computational burden of the U-Net architecture by implementing a joint architecture consisting of convolutional and multi-layer perceptron (MLP) blocks while preserving the U-shaped from the original model. Firstly, it decreases the number of filters needed for both the encoder and decoder at all levels of spatial representation. Secondly, the token MLP blocks are included in the latter two stages of the encoder and the former two stages of the decoder. These modules receive

the input feature maps from the prior convolutional stages and are passed through a series of transformations, commonly used in transformer-like networks, such as tokenization and axial shifts for locality introduction (Figure 1).

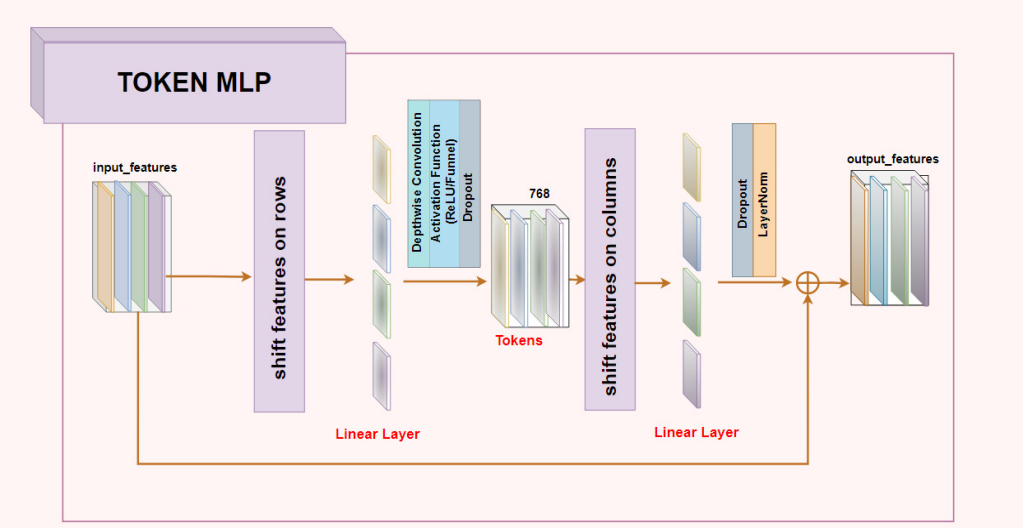


Figure 1. Token MLP block.

Another significant change with regard to the number of learnable parameters is in the process of upsampling. The transposed convolutional layers in the original U-Net are replaced with bilinear interpolation transformations in order to reduce the computational burden. Moreover, instead of concatenating the features as presented in the original U-Net, in UNeXt, they are added.

Furthermore, as seen in Table 1, the main focus of recent publications was on changing the architectures at macro levels by adding or replacing specific modules and less on how micro changes can affect the performance of a neural network, such as changing the activation functions. Currently, the most used activation function is ReLU, presented in Equation (1), due to its simplicity and effectiveness. However, ReLU is not an activation function specifically designed for computer vision tasks, where contextual information might be essential for better performance.

$$relu : \mathbb{R} \rightarrow [0, \infty), \quad relu(x) = \max(0, x) \tag{1}$$

where x is the resulted output of the pixel from the previous layer.

Thus, in the paper [23], a novel function is proposed, Funnel, that tries to diminish the spatial insensitiveness of activation functions. By introducing this function, the authors concluded that it showed significant improvements in image classification, object detection, and semantic segmentation tasks performed on three independent datasets.

The design of the Funnel activation function consists of a spatial condition and a $\max(\cdot)$ non-linearity. The spatial condition is regarded as a depthwise convolutional layer that can capture contextual information from adjacent pixel values while still preserving the positional information. With this spatial condition (funnel condition), boundary pixels, especially those delimiting highly irregularly shaped objects or extremely small objects, are taken into consideration in the final output, due to the extent of the receptive field in deeper layers through the added convolutional layer. Use of the ReLU activation function, on the other hand, results in a smoothing of irregularities on account of its insensitiveness to spatial locality, leading to a poorer segmentation performance when dealing with highly irregular shapes. After the depthwise convolutional layer is applied, normalization is

performed using a BatchNorm Layer. The max(·) non-linearity provides each pixel the option of referring to the spatial surrounding or not, as expressed in Equation (2):

$$funnel : \mathbb{R} \to \mathbb{R}, \quad funnel(x) = \max(x, T(x)) \tag{2}$$

where $T(x)$ is the resulted output of the pixel after performing depthwise convolution and batch normalization.

Having the aforementioned observations, the two architectures can be seen in Figures 2 and 3.

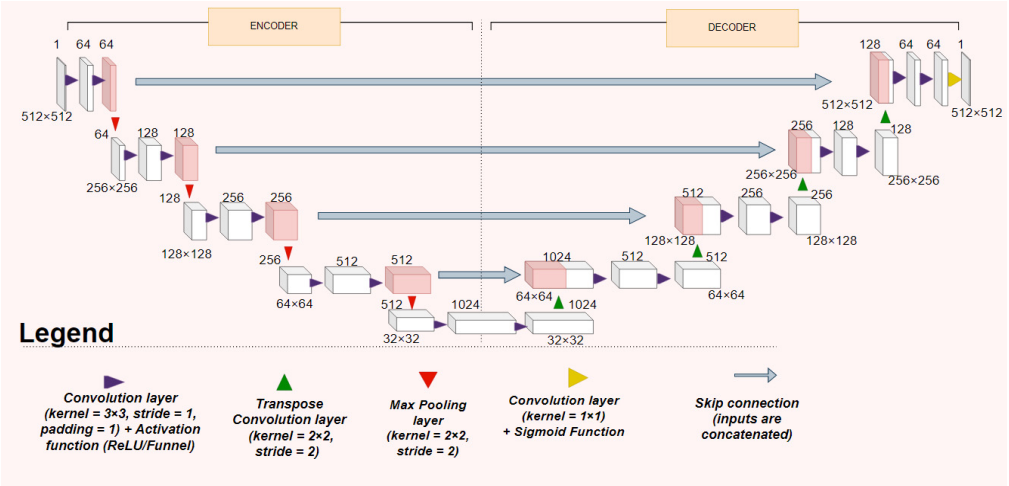


Figure 2. Investigated UNet architecture.

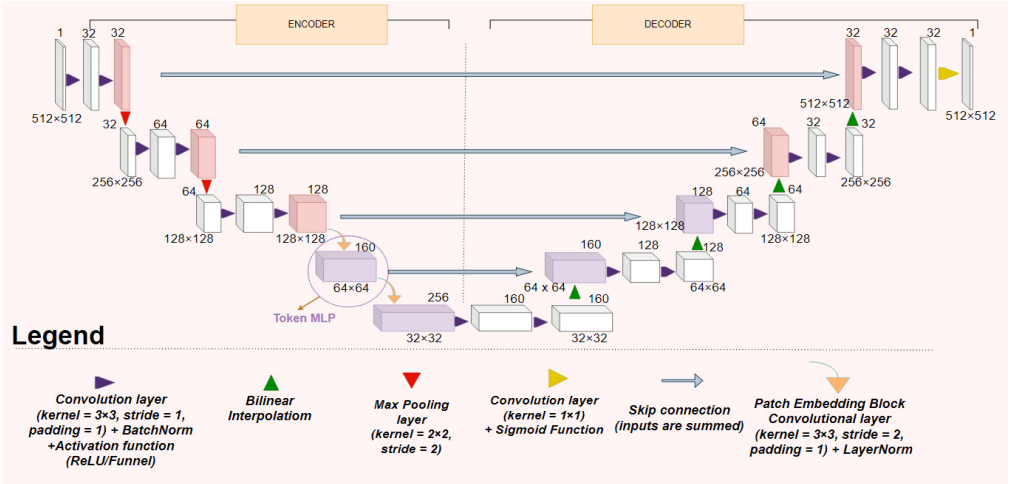


Figure 3. Investigated UNeXt architecture.

The two architectures were trained separately with respect to the two activations functions, ReLU and Funnel, and with respect to the two proposed loss functions explained in Section 2.2. Thus, in total, there were eight combinations of the models that were trained and tested (four models for UNeXt and four models for U-Net).

2.2. Loss Functions

In the context of medical image segmentation, class imbalance and general variability of the input data are prevalent issues that can affect the model's performance. Thus, choosing a loss function is primarily dependent on the type of training data. In a thorough analysis conducted in [24], the authors categorized the existing loss functions into four

categories: distribution-based, region-based, boundary-based, and a mixture of the prior three categories. Distribution-based losses concentrate on suppressing the difference between two data distributions (the predicted segmentation and the ground truth mask). The region-based losses focus on the maximization of the spatial overlap between the prediction and the ground truth, whereas boundary-based losses seek to minimize the differences through a defined distance metric. The latter are instable during training, and therefore they are usually combined with region-based losses The authors also state that it is not a standard loss function that can perform well on all segmentation tasks but a linear combination of more loss functions can help in alleviating the class imbalance problem and the capability to be more stable when training.

Two loss functions are tested in this paper, soft dice loss, and a variant of unified focal loss [25].

Soft dice loss (SDL) function, presented in Equation (5), is based on an adaptation of the dice similarity coefficient, presented in Equation (3), where the denominator contains squared terms (Equation (4)). It is a region-based loss function that concentrates on evaluating the overlapping areas between the predicted segmentation and the ground truth mask.

$$DSC = \frac{2\sum_{i=1}^{H}\sum_{j=1}^{W} p_{i,j} g_{i,j}}{\sum_{i=1}^{H}\sum_{j=1}^{W} p_{i,j} + \sum_{i=1}^{H}\sum_{j=1}^{W} g_{i,j}} \tag{3}$$

$$SoftDSC = \frac{2\sum_{i=1}^{H}\sum_{j=1}^{W} p_{i,j} g_{i,j}}{\sum_{i=1}^{H}\sum_{j=1}^{W} {p_{i,j}}^2 + \sum_{i=1}^{H}\sum_{j=1}^{W} {g_{i,j}}^2} \tag{4}$$

where $p_{i,j}$ represents the predicted raw pixel's value probability (stating if the pixel represents or not a liver region), $g_{i,j}$ is the ground-truth pixel's value (0 and 1 values), and H, W are the dimensions of the CT slice.

$$SoftDiceLoss = 1 - SoftDSC \tag{5}$$

The second loss function, unified focal loss, as stated in [25], is a linear combination between focal loss, a distribution-based loss, and focal Tversky loss, a region-based loss.

Focal loss [26] concentrates more on penalizing the class imbalance by focusing on correcting the misclassified pixels and down-weighting the easily classified ones. It was introduced for object detection on extremely class-imbalanced datasets. The derivation from binary-cross entropy is by introducing a modulation factor with a tunable parameter γ. After ablation tests in the original paper, the tunable parameter was proposed to be considered 2. It can also be expressed as

$$FocalLoss = -\frac{1}{HW}\sum_{i=1}^{H}\sum_{j=1}^{W} (1-p_{i,j}^t)^{\gamma} \log(p_{i,j}^t) \tag{6}$$

where

$$p_{i,j}^t = \begin{cases} p_{i,j}\ , & if\ \ g_{i,j} = 1 \\ 1 - p_{i,j}\ , & otherwise \end{cases} \tag{7}$$

and $p_{i,j}$ represents the predicted raw pixel's value probability (stating if the pixel represents or not a liver region), $g_{i,j}$ is the ground-truth pixel's value (0 and 1 values), and H, W are the dimensions of the CT slice.

The other component of the compound function is represented by the focal Tversky loss. It is derived from the Tversky index (TI) formulated in Equation (8), which is a variant of the dice similarity coefficient. TI, which is presented in [27], considers two

parameters, α *and* β, that control the overall contribution of false positives (FPs) and false negatives (FNs), respectively. Depending on the task, using these changes can improve class imbalances.

$$TI = \frac{\sum_{i=1}^{H}\sum_{j=1}^{W} p_{i,j}g_{i,j}}{\sum_{i=1}^{H}\sum_{j=1}^{W} p_{i,j}g_{i,j} + \beta\sum_{i=1}^{H}\sum_{j=1}^{W}(1-p_{i,j})g_{i,j} + \alpha\sum_{i=1}^{H}\sum_{j=1}^{W} p_{i,j}(1-g_{i,j})} \tag{8}$$

where $p_{i,j}$ represents the predicted raw pixel's value probability (stating if the pixel represents or not a liver region), $g_{i,j}$ is the ground-truth pixel's value (0 and 1 values), and H, W are the dimensions of the CT slice.

On the basis of the aforementioned Tversky index, the focal Tversky loss (FTL) can be formulated as in Equation (9):

$$FTL = \sum_{c=1}^{C}(1-TI)^{\frac{1}{\gamma}} \tag{9}$$

where C is the number of classes, in this case, the foreground (liver class) and the background (non-liver class). For the parameter γ, the proposed optimal value in the original paper [25] is $4/3$, but in this paper, the focal Tversky loss was reduced to simply Tversky loss by choosing $\gamma = 1$.

Thus, the unified focal loss (UFL) can be written as in Equation (10):

$$UFL = \lambda FL + (1-\lambda)FTL \tag{10}$$

where λ is a tunable parameter with values between 0 and 1, depending on which loss should be put more emphasis.

In this paper, the following values for the tunable parameters were determined on the basis of how the unified focal loss function was derived. For the focal loss, it was selected as $\gamma = 2$ in Equation (6), and the focal Tversky loss was transformed into solely Tversky loss by considering the corresponding $\gamma = 1$ in Equation (9). Inside the Tversky index, the parameters had values of $\alpha = 0.4$ and $\beta = 0.6$, respectively. Moreover, for the unified focal loss, it was chosen as $\lambda = 0.5$ in Equation (10).

2.3. Proposed Automatic Post-Processing Filtering

In this paper, the proposed models were trained on CT volumes that were reduced to the CT slices where the liver masks were present. This choice was motivated by the fact that, in general, for an automatic medical segmentation task, the region of interest is a lesser part of the total data volume, and the suggested architecture should be focused more on the relevant foreground. However, from a research perspective, it can be a trade-off between the model's generalization capabilities and overall performance. In this case, having trained on input images that concentrated solely on the liver's presence, the expectation was that when testing on CT slices with non-existent liver, the model's performance would be diminished. Thus, an automated post-processing method that can encapsulate the shortcomings derived from training is further proposed.

The intuition behind the proposed method is to make use of the spatial sequentiality of the slices (not) containing the organ, namely, the fact that on the axial plane, the CT slices are displayed from one end of the abdomen to the other in a sequential manner. Given that the region of interest, the liver, is not present in the whole CT volume, it is desirable to be able to quantify where the CT slices containing the organ may start and end. Thus, if one were to simulate the process of looking manually through a CT volume, one could observe that the liver region through sequential slices is compact, in most cases in the order of small, medium, large, medium, and small liver segments.

In this respect, we consider a function that takes as an argument the slice number and as a result the estimated liver area on that slice, which can be expressed as

$$f : \mathbb{N} \to [0, H \cdot W], \quad f(slice_k) = \sum_{i=1}^{H} \sum_{j=1}^{W} p_{i,j} \tag{11}$$

where $slice_k$ is each index number slice contained inside of one CT volume, $p_{i,j}$ is the predicted thresholded value of the pixel from the inferred segmentation mask (0—not part of the liver region, 1—part of the liver region), and H, W are the dimensions of the CT slice (in this case, 512).

By plotting this function, one can see that the region where the liver is predominant has a bell-shaped curve, whereas the areas where the model misclassifies a non-liver slice region as the liver appears as noise that can be filtered.

The proposed post-processing approach consists in finding the maximum of the bell-shaped curve and descending in both directions, right and left, to determine the first zero-valued points. These points are regarded as indices of the slices that define the coarse region of interest, as seen in Figure 4. Outside of these indices, the filtering procedure marks all slices as liver free. The post-processing results can be seen in Figure 5.

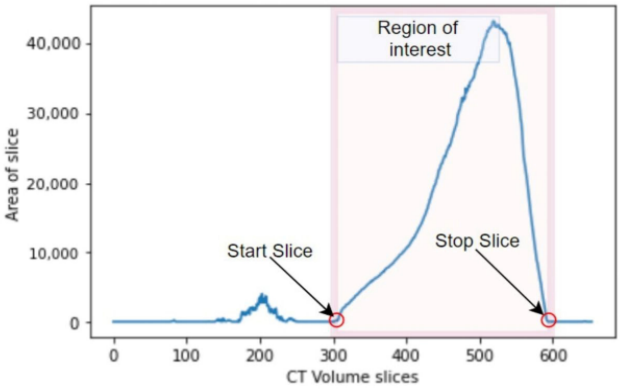


Figure 4. Proposed post-processing filtering.

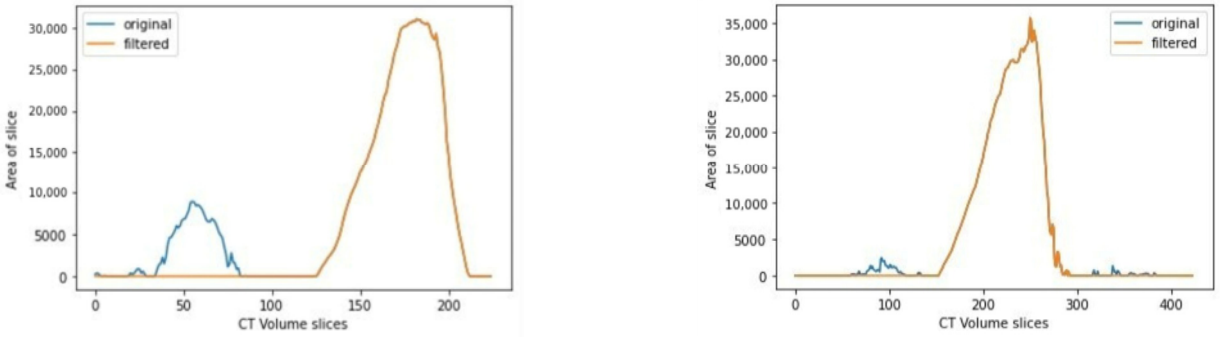


Figure 5. Results of the filtering (post-processing) step on two different (**a**,**b**) CT volumes from the testing set.

It needs to be stated that this post-processing method does not modify the predicted masks but only concentrates on filtering the noise outside the liver volume region. Thus, the models' capacity to segment the liver is still determined by the raw predictions.

3. Experimental Setup

3.1. LiTS Dataset and Pre-Processing

The LiTS (liver tumor segmentation dataset) was created for two specific tasks: liver segmentation and estimation of tumor burden [6]. It includes 201 abdominal CT volumes in the portal vein phase from patients with hepatocellular carcinoma (HCC) as the primary tumor illness. It is also a data acquisition collaboration within seven hospitals and research institutes from across the world, and it is publicly available at [28]. The 201 abdominal CT volumes stated above are divided into a train set of 131 volumes, with the matching ground-truth liver and lesion annotations manually reviewed by expert radiologists. Due to the fact that the test set does not provide publicly available ground truth, the train set CTs were used for the paper's experiments.

As for pre-processing steps, each slice from a CT volume underwent HU clipping in the range $[-100, 400]$ in order to subtract the irrelevant background information. Furthermore, the ground-truth masks were binarized by neglecting the tumor regions and just considering them part of the liver.

For a fair comparison, the suggested architectures were trained and evaluated on the LiTS dataset's original resolution size (512×512).

3.2. Train-Test Procedure

The 131 pre-processed CT volumes from the LiTS train set were randomly split into 111 CT volumes for training and 20 CT volumes for testing. For training purposes, only the slices in the CT volumes where the liver was present were used. The CT volumes for training underwent a 10-fold cross-validation in order to determine the generalization capacity of the proposed models, and the best-trained model was then chosen for testing. Each trained model was tested on the 20 CT volumes, and the best folds were chosen for the visualization of the quantitative and qualitative results. Each architecture was trained with respect to the activation functions, ReLU and Funnel, and the proposed loss functions, soft dice loss and unified focal loss. The trained models were evaluated on the evaluation metrics stated in Section 3.4.

3.3. Implementation Details

The proposed architectures were trained with respect to an early stopping strategy with a patience of 10 epochs to prevent overfitting and reduce the used resources footprint. A variant of Adam optimizer, AdamW [29], was used for training with an initial learning rate of 0.001. Due to restrictions in GPU memory, the batch size for architectures was set to 32 for the UNeXt architecture and 8 for UNet, respectively. The implementation of the architectures and the training and testing process were written in Python 3.9.7 using the PyTorch framework.

3.4. Evaluation Metrics

In [30,31], there are several metrics presented that are commonly used to quantitatively evaluate the model's performance for a medical image segmentation task. The most prevalent indicators used for evaluation are dice similarity coefficient (DSC) formulated in Equation (12), the Jaccard index or intersection over union (IoU) as in Equation (13), specificity presented in Equation (14), and sensitivity as in Equation (15). Each of them is based on the computation of the coefficients from the confusion matrix, namely, true positive (TP), true negative (TN), false positive (FP), and false negative (FN).

$$DSC = \frac{2TP}{2TP + FN + FP} \tag{12}$$

$$IoU = \frac{TP}{TP + FN + FP} \tag{13}$$

$$Specificity = \frac{TN}{TN + FP} \tag{14}$$

$$Sensitivity = \frac{TP}{TP + FN} \tag{15}$$

In this paper, the dice similarity coefficient (DSC) was used as a baseline for choosing the best models, and further result discussions primarily focus on this metric. The DSC score measures the degree of area overlapping between the ground truth segmentation and the model's predicted segmentation. DSC takes values in the interval [0, 1], where 0 denotes no overlapping and 1 denotes a perfect overlapping. This metric is commonly used to evaluate the performance of a medical image segmentation algorithm [30–32]. Moreover, in the LiTS benchmark [7], the DSC is considered as a ranking criterion for the MICCAI liver segmentation challenge, specifically the DSC per case, where the DSC is calculated with respect to each CT volume.

4. Results

4.1. Pre-Processing Results

On the basis of the results provided in Table 2, it can be concluded that the U-Net models (with the Funnel activation included), when trained with the same loss function (soft dice loss and unified focal loss) had a higher DSC and IoU scores than the ones with ReLU. The UNeXt models, nonetheless, had comparable results with their counterparts with less than a 0.02 difference on all evaluation metrics.

Table 2. Evaluation metrics on LiTS test set (20 CT volumes containing solely the liver region) with no post-processing performed—best models from 10-fold cross-validation.

Proposed Model (Activation/Loss Function Used for Training)	Dice Similarity Coefficient	Specificity	Sensitivity	Intersection over Union (IoU) or Jaccard Index
UNeXt (Funnel/unified focal loss)	0.9401 (0.0434) *	0.9565 (0.0448)	0.9663 (0.0135)	0.9044 (0.0623)
UNeXt (ReLU/unified focal loss)	0.9538 (0.0199)	0.9741 (0.0198)	0.9626 (0.0153)	0.9236 (0.0286)
UNeXt (Funnel/soft dice loss)	0.9473 (0.026)	0.9709 (0.0263)	0.9571 (0.0139)	0.915 (0.0359)
UNeXt (ReLU/soft dice loss)	0.9453 (0.019)	0.9761 (0.017)	0.9495 (0.0182)	0.9129 (0.0256)
U-Net (Funnel/unified focal loss)	0.9503 (0.0299)	0.9761 (0.0205)	0.9582 (0.02)	0.9214 (0.04)
U-Net (ReLU/unified focal loss)	0.9435 (0.0423)	0.9789 (0.0207)	0.9466 (0.0361)	0.9143 (0.0582)
U-Net (Funnel/soft dice loss)	0.9606 (0.0263)	0.9772 (0.0204)	0.9702 (0.0179)	0.9358 (0.037)
U-Net (ReLU/soft dice loss)	0.9570 (0.0293)	0.9784 (0.0186)	0.9645 (0.025)	0.9316 (0.041)

* Mean (standard deviation).

However, a more accurate analysis can be depicted in Table 3, where the results are averaged with regard to all trained models on 10-fold cross-validation. In this case, both architectures, U-Net and UNeXt, performed better (with respect to the DSC and specificity score) when trained on soft dice loss and had the Funnel activation included. From this, one can infer that the Funnel activation function has the potential of improving marginally the overall metrics and that loss functions should be considered a hyper-parameter that can be tuned. Moreover, some qualitative results inferred on the trained models are presented in Figure 6. The slices that were chosen highlight the models' capability to segment some corner cases, such as CT slices that contain small liver parts or disconnected liver parts, or CT slices that contain irregular shapes (hence, the segmentations would be much more prone to misclassifications).

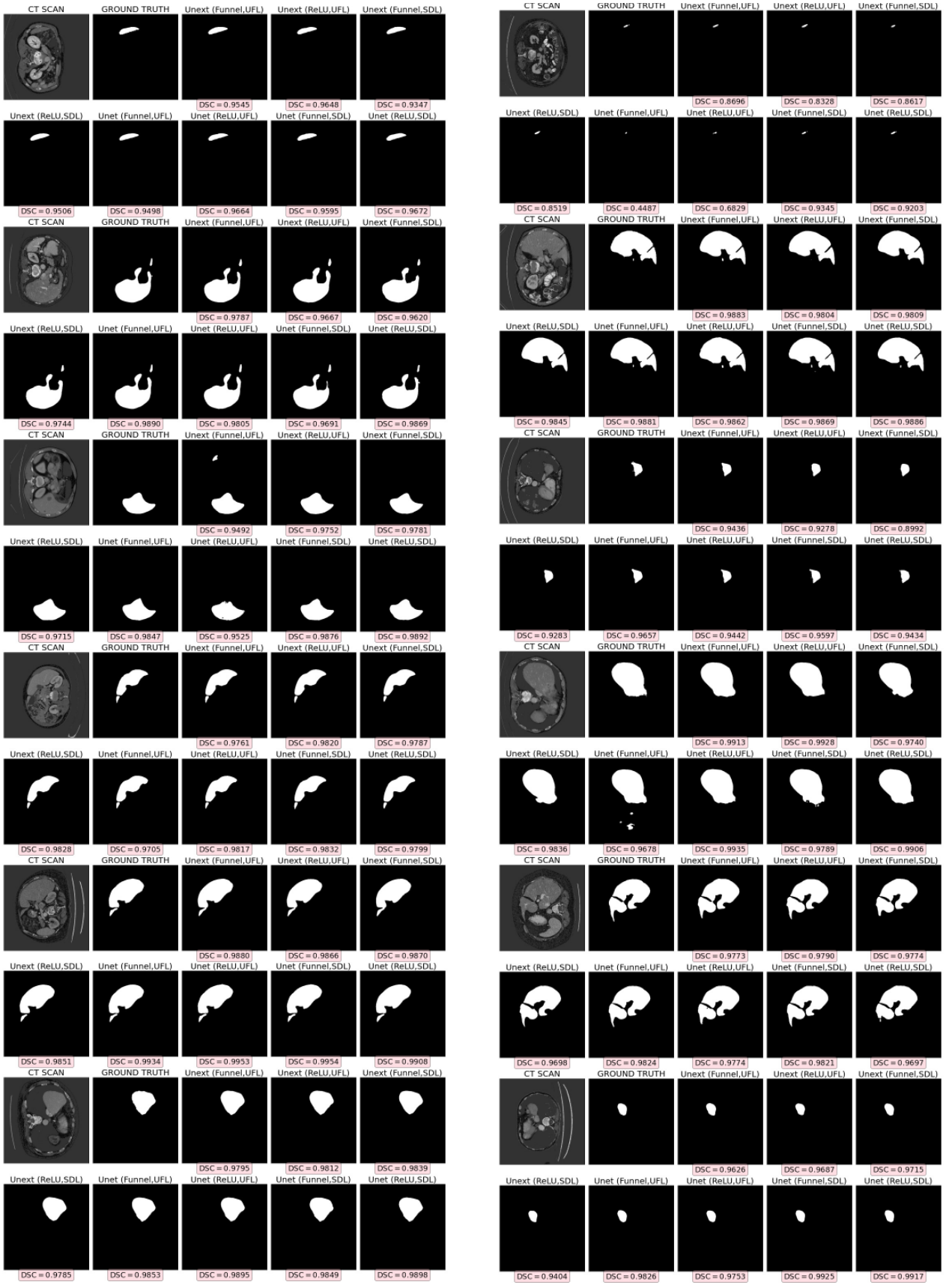


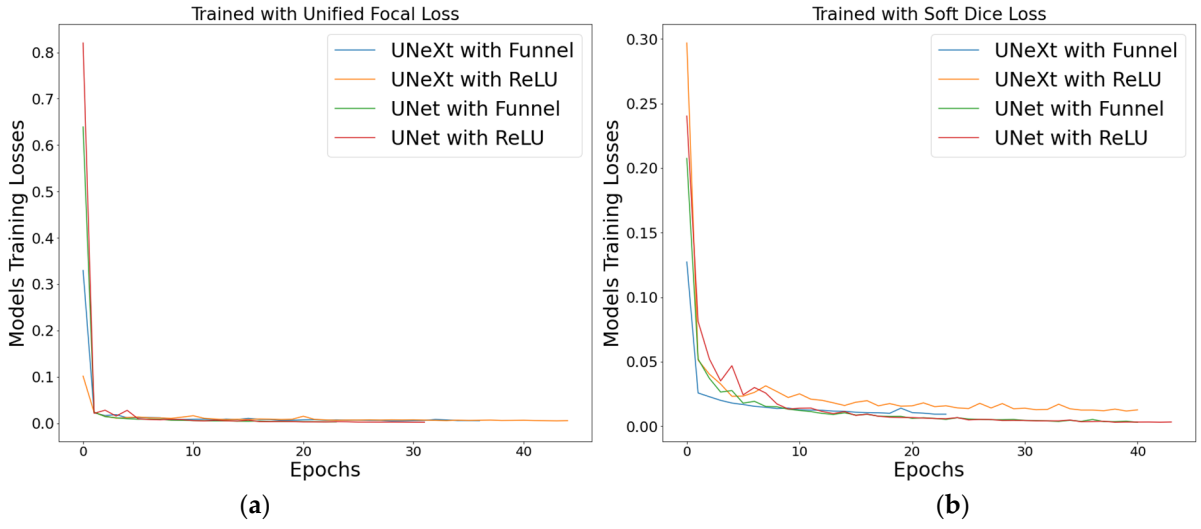
Figure 6. Qualitative results of the trained models from several CT slices from the test set volumes.

Table 3. Evaluation metrics on LiTS test set (20 CT volumes containing solely the liver region) with no post-processing performed—average mean across the 10-fold cross-validation for each trained model.

Proposed Model (Activation/Loss Function Used for Training)	Dice Similarity Coefficient	Specificity	Sensitivity	Intersection over Union (IoU) or Jaccard Index
UNeXt (Funnel/unified focal loss)	0.9251 (0.0136) *	0.9613 (0.0089)	0.9414 (0.0183)	0.8849 (0.0187)
UNeXt (ReLU/unified focal loss)	0.9307 (0.0126)	0.9645 (0.0068)	0.9423 (0.0134)	0.8925 (0.0166)
UNeXt (Funnel/soft dice loss)	0.932 (0.015)	0.9670 (0.0057)	0.9424 (0.0168)	0.8951 (0.0195)
UNeXt (ReLU/soft dice loss)	0.9243 (0.0148)	0.9665 (0.0069)	0.9311 (0.0148)	0.8839 (0.0196)
U-Net (Funnel/unified focal loss)	0.9201 (0.0236)	0.9704 (0.0103)	0.9229 (0.0246)	0.8806 (0.0321)
U-Net (ReLU/unified focal loss)	0.9298 (0.0131)	0.9702 (0.0068)	0.9350 (0.0148)	0.8921 (0.0188)
U-Net (Funnel/soft dice loss)	0.9343 (0.0161)	0.9762 (0.0059)	0.9343 (0.0201)	0.8995 (0.0224)
U-Net (ReLU/soft dice loss)	0.9331 (0.0236)	0.9678 (0.016)	0.9435 (0.0145)	0.9115 (0.0267)

* Mean (standard deviation).

From Figure 7, one observation would be that when trained with unified focal loss, the UNeXt models had a smaller initial learning loss compared to the U-Net models. When trained with soft dice loss, by introducing the Funnel activation function, regardless of the architecture, the model is capable of learning faster even from the first epoch, compared to the models that contain ReLU. Moreover, the UNeXt models trained with soft dice loss converge faster on the basis of the early stopping criteria, but they are more unstable than the U-Net models. This could be explained by the fact that the U-Net architecture, being a heavy-parameter architecture, is more robust to changes in batches during training. However, if we choose to compare based on the training losses, another observation would be that training with soft dice loss is much more unstable compared to Unified focal loss, but both training losses led to a training convergence up to 0 epochs for all models.



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

Figure 7. Training losses for each model: (**a**) models trained with unified focal loss, (**b**) models trained with soft dice loss.

4.2. Post-Processing Results

Having the aforementioned results on the CT volumes that contain just the liver masks, for a fair comparison of the results, the models were tested also on the whole CT volumes. Furthermore, the automatic post-processing step was applied.

From the results in Table 4, it can be observed that the models that made use of Funnel activation had poorer pre-processing performances compared to the models with ReLU activation. Secondly, on an architectural level, the UNeXt was less robust to non-liver CT slices compared to the UNet, with differences in evaluation scores between 0.02 and 0.3 on all metrics, except sensitivity (sensitivity in this context of post-processing remained the same).

The post-processing step led to significant improvements, in spite of its limitations (sensitive to false negatives or false positives at the organ's extremities, which can interfere with the true indexes of start and finish). In order to quantify the error at this step, we extracted from the LiTS segmentations for each CT volume the index of the slices where the liver mask first and last appeared. These were compared to the predicted index slices obtained after the post-processing step. The error was calculated independently for the start index slices and the stop index slices with respect to the root-mean-square error (*RMSE*), as expressed:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} (s_{ind} - \hat{s}_{ind})^2}{N}} \tag{16}$$

where s_{ind} is the true start/stop index slice from the ground-truth masks, $\hat{s}_{ind}$ is the estimated start/stop index slice after the model's inference, and $N = 20$ (the number of the CT volumes present in the test set).

Table 4. Evaluation metrics on the LiTS test set (20 CTs—whole volumes) before and after the post-processing was performed—best models from 10-fold cross-validation.

Proposed Model (Activation/Loss Function Used for Training)	Post-Processing	Dice Similarity Coefficient	Specificity	Sensitivity	Intersection over Union (IoU) or Jaccard Index
UNeXt (Funnel/unified focal loss)	Before	0.6695	0.6757	0.9903	0.6615
	After	0.9883	0.9945	0.9903	0.9803
UNeXt (ReLU/unified focal loss)	Before	0.8786	0.8885	0.9825	0.8692
	After	0.9874	0.9973	0.9825	0.978
UNeXt (Funnel/soft dice loss)	Before	0.7277	0.7317	0.9932	0.7202
	After	0.9902	0.9941	0.9932	0.9827
UNeXt (ReLU/soft dice loss)	Before	0.9602	0.9741	0.9769	0.9501
	After	0.9845	0.9985	0.9769	0.9827
U-Net (Funnel/unified focal loss)	Before	0.9514	0.9583	0.9876	0.9442
	After	0.9912	0.9979	0.9876	0.9839
U-Net (ReLU/unified focal loss)	Before	0.9509	0.9572	0.9889	0.9446
	After	0.9906	0.9969	0.9889	0.9843
U-Net Funnel/soft dice loss	Before	0.908	0.9092	0.9974	0.9056
	After	0.9976	0.9989	0.9974	0.9953
U-Net ReLU/soft dice loss	Before	0.9799	0.981	0.9978	0.9778
	After	0.9978	0.9989	0.9978	0.9957

On the basis of the results in Table 5 and Figure 8, it can be concluded that the training process concentrated solely on the liver region introduced unwanted noise (false positives). Moreover, on the estimations of the stop index slices, the architectures had the tendency to "delay" the liver's presence. This can be explained by the fact that in the data acquisition process, near the end, the pelvic region might also be present, and the HU units can be similar to the liver region with contrast enhancement. Moreover, for better visualization of the models' performance, two CT volumes that had the biggest estimation errors were chosen from the test set and are shown in Figure 8.

Table 5. Error estimations for the automatic post-processing filtering for start/stop index slices—results obtained on the 20 CT volumes test set.

Proposed Model (Activation/Loss Function Used for Training)	RMSE—Start Index Slice	RMSE—Stop Index Slice
UNeXt (Funnel/unified focal loss)	1.6	48.5
UNeXt (ReLU/unified focal loss)	27.1	9.4
UNeXt (Funnel/soft dice loss)	2.75	35.8
UNeXt (ReLU/soft dice loss)	5.75	18.85
U-Net (Funnel/unified focal loss)	3.05	18.15
U-Net (ReLU/unified focal loss)	10.75	20.05
U-Net (Funnel/soft dice loss)	2.5	41.05
U-Net (ReLU/soft dice loss)	5.1	29.25

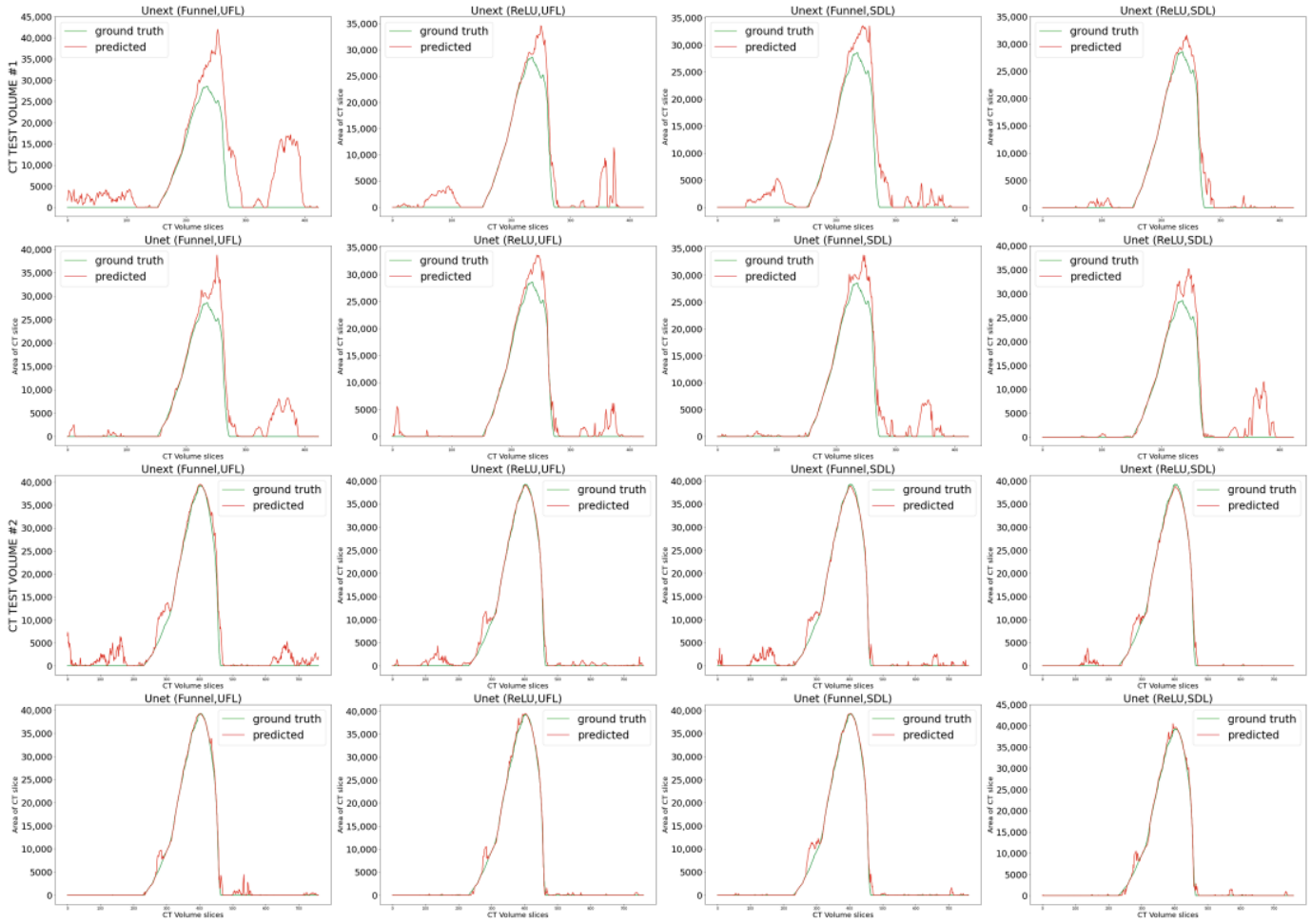


Figure 8. Representation of the ground-truth and the predicted segmentations for two CT test volumes (#1 and #2) on all trained best models, before post-processing was performed.

From Figure 8, it can be seen that in the first case, not only noise was introduced in the non-liver slices, but also the models were prone to introducing more false positives. This representation of a CT volume can also be used as a way to quantify the possible errors that the model introduces and also to detect the tendencies they have for misclassifications. Another thing that can be observed is that, in the second case, the UNeXt architecture was more unstable in predictions than U-Net. This can be justified by the fact that U-Net is, parameter-wise, a much larger architecture, and this contributes to its overall robustness.

However, using the post-processing step, especially for the UNeXt architecture, aids in filtering the additional noise generated, resulting in DSC scores comparable to the U-Net architecture (Table 4), but with superior inference time performance (Table 6).

Table 6. Number of learnable parameters for each proposed architecture and inference time on GPU (average mean on the 20 CT whole test volumes approx. 12 GB data) vs. CPU (largest CT volume approx. 1.2 GB data from the test set).

Proposed Model	Number of Learnable Parameters (M)	Inference Time —GPU(s)		Inference Time —CPU(s)	
		Without Post-Processing	With Post-Processing	Without Post-Processing	With Post-Processing
Proposed UNeXt with ReLU activation	4.09	21.73	24.59	481.98	486.2
Proposed UNeXt with Funnel activation	4.12	28.95	31.95	695.92	700.54
Proposed U-Net with ReLU activation	59.27	86.04	88.71	2601.11	2604.88
Proposed U-Net with Funnel activation	59.4	109.63	112.43	3403.19	3407.23

The post-processing does not lead to a significant increase in the overall inference time (Table 6) because the inference is performed once and then the non-liver partial volumes at the start and the end of the CT volume will have associated non-liver masks that are merged with the original liver volume (determined by the two slice indexes predicted).

4.3. Observations on Parameter Numbers and Inference Time

Time is another asset that must be considered while developing the framework, in addition to the metrics that assess the model's accuracy. A shorter training time allows the researcher to experiment with several versions of the suggested architecture and fine-tune the model. Inference speed, on the other hand, is critical for integrating the model in a healthcare unit, where the doctor may be exempted from time-consuming manual evaluations and deliver a rapid patient screening. As a result, developing an architecture that can have comparable results with other heavy parameter models while still providing efficient time solutions would be preferred. Additionally, it will decrease the hospital's need for costly technological resources required for the application's deployment.

For testing the inference speed of the proposed architectures, the calculations were performed on the testing set used for the models' evaluation, namely, 20 CT volumes. The final results were averaged in order to provide an estimate. Tests with GPU integration were performed on NVIDIA GeForce RTX 3090, whereas solely on CPU, the designated CPU was an Intel® Core™ i9-10980XE CPU @ 3.00 GHz × 36. Due to time constraints, the CPU testing was set to the largest CT volume from the testing dataset. It was done to highlight the capabilities of the UNeXt architecture, even in the absence of a GPU environment. Even though the usage of GPU is always favored due to faster computations, technological resources in healthcare units might be limited, and devices may not supply this processing advantage. Moreover, as stated in [5], a manual segmentation performed by a clinician can lead up to 90 min for one patient volume assessment. The UNeXt architecture is capable of leveraging the time burden, even with just a CPU environment, emphasizing that lightweight models can become an important asset in a real-life clinical setup.

As seen in Table 6, the UNeXt architecture has about 15x fewer learnable parameters than the U-Net. Furthermore, because the Funnel activation function is composed of layers that add a number of learnable parameters, an architecture that contains the aforementioned activation will have additional parameters but an insignificant number in comparison to the entire quantity.

The best result obtained by a lightweight architecture in terms of DSC is 0.96 (Table 1); the model has 3.6 million parameters, and no post-processing steps are made [13]. Compared to it, the best result obtained by the lightweight architecture in this paper is with the UNeXT model (Funnel activation, Soft Dice Loss), with a DSC value of 0.99 (4.12 million parameters), after applying the post-processing step. However, the U-Net model (ReLU activation, Soft Dice Loss), with the post-processing step included, leads to marginal superior results in terms of evaluation metrics, but the inference time was approximately three times greater compared to the UNeXT model (Funnel activation, soft dice loss).

4.4. Discussion

The lack of standardization in recent publications makes a comparative analysis prone to subjectivity, because of the differences in train-test procedure and the limited capabilities of reproducibility since not all steps are described or even mentioned. This issue is gradually becoming more addressed in the AI community, especially in the clinical health setup. In [33], the authors present the current barriers that prevent the state-of-the-art models from having the desired clinical impact, such as dataset biases due to unrepresentative benchmarks, improper evaluation metrics, and ambiguity and subjectivity in terms of testing the model. Moreover, a recurrent problem in the liver segmentation task is choosing the datasets for proposed models' evaluation. Lacking an in-depth description of the publicly available datasets, it creates the possibility of introducing biases in the presented results by, for example, using the same cohort for training and testing. The most striking example is the usage of 3DIRCADb [34] dataset for testing and LiTS for training, because few papers have the a priori knowledge that the former is included, in fact, in the LiTS dataset. Thus, the veracity of the results might be questioned, leading to improper comparison baselines across state-of-the-art papers.

Another aspect that should be tackled is the poor analysis of the models' performance aside from the evaluation metrics. On the liver segmentation task, the majority of papers do not provide information about the architecture's number of parameters or inference time. However, there is a propensity to create complex architectures that are briefly explained, without having a thorough knowledge on the models' computational burden. Of course, the outcomes are spectacular in a controlled environment with appropriate resources, but they are seldom implemented in healthcare units due to high associated costs. Only a few [8,13,16,20] of the research publications covered in Section 1.2. offered explicit details on the number of parameters. Among those, [13,16] implemented a 2D or 2.5D approach with less than 4 million parameters. Even in this scenario, a fair comparison with the results cannot be performed, because in [13] the results are supplied on the original LiTS test set, whereas in [16] the evaluation is inferred solely on eight CT volumes from the LiTS train set. Nonetheless, by including the architecture's number of parameters in the papers, we would create an objective setup for comparing state-of-the-art papers because this criterion is independent of computational resources employed for the train–test procedures, in contrast to training and inference time. In this context, the research should be focused not only on creating frameworks that reduce the computational footprint and preserve the desired outcome, but also on providing a comprehensive analysis of the models' impact in different environments. Moreover, being concerned with implementing lightweight models would enable researchers to reduce the time burden of testing and tuning the models, in addition to the actual expenses associated in employing them in a real-life scenario when time is essential.

Some limitations of this work should be pointed out. First of all, we do not provide an in-depth comparison to the models listed in Table 1, because, as stated above, a direct comparison is difficult. On one hand, the models proposed in the literature are trained and tested on different datasets and, on the other hand, an indirect comparison would be prone to discussions (most of the articles do not provide complete code or enough details to implement the model or pre-processing and post-processing steps to enable us to reproduce

their work). However, we consider that the U-Net model (ReLU activation function, soft dice loss function) serves as a representative state-of-the-art model against which the proposed lightweight architecture can be compared. Secondly, our work is limited in terms of network stability and complexity evaluation with other lightweight models. Another limitation is that we have used only slices containing the organ for training the models and did not compare the results to the models trained on the whole CT volumes.

In spite these limitations, with our work, we expect to clarify at least some of the previously established observations: in-depth discussion on how adjustments at the micro and macro levels of a deep learning architecture can influence its capabilities (using different architectures with different activation and loss functions); on how the model size impacts the performance and computational burden; and how by introducing an adequate post-processing step, it can alleviate training shortcomings and increase the overall model performance metrics.

5. Conclusions and Future Work

According to our findings, the lightweight architecture UNeXt has great potential in outperforming its counterpart U-Net with respect to evaluation metrics, inference performance, and memory footprint. Furthermore, the proposed post-processing method can significantly improve the model's performance, minimally impacting the computational time. Consequently, there is no standard for selecting loss functions since they can be architecture specific and entirely reliant on the input data. Thus, when training a deep learning model, the loss functions should be regarded as an extra hyper-parameter that needs to be tuned. Ultimately, the contribution of the novel activation function, Funnel, shows promise in enhancing the overall model's performance and can serve as a starting point in creating more specific functions that still introduce non-linearity while still preserving more contextual information in computer vision tasks.

All in all, the best trained UNeXt model (with the Funnel activation, trained with soft dice loss and followed by the proposed post-processing step) achieved a 0.9902 DSC score on the whole CT volumes test set with 15$\times$ fewer learnable parameters in nearly 4$\times$ less inference time.

As future work, we would also like to have a broader comparison with other different lightweight models in such way that we could conduct an analysis on the several aspects, such as stability, evaluation metrics, and complexity. We also consider comparing the UNeXt architecture with adaptations of other lightweight models such as MobileNet [35] and ShuffleNet [36] that are designed for image classification but can be adapted as backbones for segmentation tasks. Another future direction is to train the models on the whole CT volume or at least to include in the training set 10% of the non-liver CT slices and ideally extend the database used for training and testing with cases of livers not affected by a disease and also livers that had a part resected.

Author Contributions: S.B. investigated the ideas, implemented the method and the formal analysis, and wrote the original draft of the manuscript. A.U. provided the conceptualization, validation of the method, resources, and funding support, and thoroughly revised the draft of the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: The research leading to these results has received funding from NO Grants 2014-2021, under project EloHyp, contract number 24/2020.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Publicly available datasets were analyzed in this study. These data can be found here: https://competitions.codalab.org/competitions/17094 (accessed on 19 November 2021).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Sung, H.; Ferlay, J.; Siegel, R.L.; Laversanne, M.; Soerjomataram, I.; Jemal, A.; Bray, F. Global Cancer Statistics 2020: GLOBOCAN Estimates of Incidence and Mortality Worldwide for 36 Cancers in 185 Countries. *CA Cancer J. Clin.* **2021**, *71*, 209–249. [CrossRef] [PubMed]
2. Global Cancer Observatory. Available online: https://gco.iarc.fr/ (accessed on 30 March 2021).
3. Wild, C.P.; Weiderpass, E.; Stewart, B.W. (Eds.) *World Cancer Report: Cancer Research for Cancer Prevention*; International Agency for Research on Cancer: Lyon, France, 2020; Available online: http://publications.iarc.fr/586 (accessed on 19 November 2021).
4. Koh, D.-M.; Papanikolaou, N.; Bick, U.; Illing, R.; Kahn, C.E.; Kalpathi-Cramer, J.; Matos, C.; Martí-Bonmatí, L.; Miles, A.; Mun, S.K.; et al. Artificial intelligence and machine learning in cancer imaging. *Commun. Med.* **2022**, *2*, 133. [CrossRef] [PubMed]
5. Gotra, A.; Sivakumaran, L.; Chartrand, G.; Vu, K.-N.; Vandenbroucke-Menu, F.; Kauffmann, C.; Kadoury, S.; Gallix, B.; de Guise, J.A.; Tang, A. Liver segmentation: Indications, techniques and future directions. *Insights Imaging* **2017**, *8*, 377–392. [CrossRef]
6. Ansari, M.Y.; Abdalla, A.; Malluhi, B.; Mohanty, S.; Mishra, S.; Singh, S.S.; Abinahed, J.; Al-Ansari, A.; Balakrishnan, S.; Dakua, S.P. Practical utility of liver segmentation methods in clinical surgeries and interventions. *BMC Med. Imaging* **2022**, *22*, 97. [CrossRef]
7. Bilic, P.; Christ, P.F.; Vorontsov, E.; Chlebus, G.; Chen, H.; Dou, Q.; Fu, C.; Han, X.; Heng, P.; Hesser, J.; et al. The Liver Tumor Segmentation Benchmark (LiTS). *arXiv* **2019**, arXiv:1901.04056. [CrossRef]
8. Yuan, Y. Hierarchical Convolutional-Deconvolutional Neural Networks for Automatic Liver and Tumor Segmentation. *arXiv* **2017**, arXiv:1710.04540.
9. Li, X.; Chen, H.; Qi, X.; Dou, Q.; Fu, C.-W.; Heng, P.-A. H-DenseUNet: Hybrid Densely Connected UNet for Liver and Tumor Segmentation From CT Volumes. *IEEE Trans. Med. Imaging* **2018**, *37*, 2663–2674. [CrossRef]
10. Chlebus, G.; Meine, H.; Moltz, J.H.; Schenk, A. Neural Network-Based Automatic Liver Tumor Segmentation With Random Forest-Based Candidate Filtering. *arXiv* **2017**, arXiv:1706.00842.
11. Vorontsov, E.; Tang, A.; Pal, C.; Kadoury, S. Liver lesion segmentation informed by joint liver segmentation. *arXiv* **2017**, arXiv:1707.07734.
12. Christ, P.F.; Ettlinger, F.; Grün, F.; Elshaera, M.E.A.; Lipkova, J.; Schlecht, S.; Ahmaddy, F.; Tatavarty, S.; Bickel, M.; Bilic, P.; et al. Automatic Liver and Tumor Segmentation of CT and MRI Volumes Using Cascaded Fully Convolutional Neural Networks. *arXiv* **2017**, arXiv:1702.05970.
13. Zhang, J.; Xie, Y.; Zhang, P.; Chen, H.; Xia, Y.; Shen, C. Light-Weight Hybrid Convolutional Network for Liver Tumor Segmentation. In Proceedings of the Twenty-Eighth Internationl Joint Conference on Artificial Intelligence (IJCAI-19); International Joint Conferences on Artificial Intelligence Organization, Macao, China, 10–16 August 2019; pp. 4271–4277. Available online: https://www.ijcai.org/proceedings/2019/0593.pdf (accessed on 27 December 2021).
14. Xia, K.; Yin, H.; Qian, P.; Jiang, Y.; Wang, S. Liver Semantic Segmentation Algorithm Based on Improved Deep Adversarial Networks in Combination of Weighted Loss Function on Abdominal CT Images. *IEEE Access* **2019**, *7*, 96349–96358. [CrossRef]
15. Tang, Y.; Tang, Y.; Zhu, Y.; Xiao, J.; Summers, R.M. E$$ˆ2$$Net: An Edge Enhanced Network for Accurate Liver and Tumor Segmentation on CT Scans. *arXiv* **2020**, arXiv:2007.09791.
16. Lv, P.; Wang, J.; Zhang, X.; Ji, C.; Zhou, L.; Wang, H. An improved residual U-Net with morphological-based loss function for automatic liver segmentation in computed tomography. *Math. Biosci. Eng.* **2021**, *19*, 1426–1447. [CrossRef] [PubMed]
17. Wang, J.; Zhang, X.; Lv, P.; Zhou, L.; Wang, H. EAR-U-Net: EfficientNet and Attention-Based Residual U-Net for Automatic Liver Segmentation in CT. *arXiv* **2021**, arXiv:2110.01014.
18. Khan, R.A.; Luo, Y.; Wu, F.-X. RMS-UNet: Residual multi-scale UNet for liver and lesion segmentation. *Artif. Intell. Med.* **2022**, *124*, 102231. [CrossRef]
19. Li, L.; Ma, H. RDCTrans U-Net: A Hybrid Variable Architecture for Liver CT Image Segmentation. *Sensors* **2022**, *22*, 2452. [CrossRef]
20. Ansari, M.Y.; Yang, Y.; Balakrishnan, S.; Abinahed, J.; Al-Ansari, A.; Warfa, M.; Almokdad, O.; Barah, A.; Omer, A.; Singh, A.V.; et al. A lightweight neural network with multiscale feature enhancement for liver CT segmentation. *Sci. Rep.* **2022**, *12*, 14153. [CrossRef]
21. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional networks for biomedical image segmentation. *arXiv* **2015**, arXiv:1505.04597.
22. Valanarasu, J.M.J.; Patel, V.M. UNeXt: MLP-Based Rapid Medical Image Segmentation Network. *arXiv* **2022**, arXiv:2203.04967.
23. Ma, N.; Zhang, X.; Sun, J. Funnel Activation for Visual Recognition. *arXiv* **2020**, arXiv:2007.11824.
24. Ma, J.; Chen, J.; Ng, M.; Huang, R.; Li, Y.; Li, C.; Yang, X.; Martel, A.L. Loss odyssey in medical image segmentation. *Med. Image Anal.* **2021**, *71*, 102035. [CrossRef] [PubMed]
25. Yeung, M.; Sala, E.; Schönlieb, C.-B.; Rundo, L. Unified Focal loss: Generalising Dice and cross entropy-based losses to handle class imbalanced medical image segmentation. *Comput. Med. Imaging Graph.* **2022**, *95*, 102026. [CrossRef] [PubMed]
26. Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal Loss for Dense Object Detection. *arXiv* **2017**, arXiv:1708.02002.
27. Salehi, S.S.M.; Erdogmus, D.; Gholipour, A. Tversky loss function for image segmentation using 3D fully convolutional deep networks. *arXiv* **2017**, arXiv:1706.05721.
28. LiTS- Liver Tumor Segmentation Challenge. Available online: https://competitions.codalab.org/competitions/17094 (accessed on 19 November 2021).
29. Loshchilov, I.; Hutter, F. Decoupled Weight Decay Regularization. *arXiv* **2017**, arXiv:1711.05101.

30. Müller, D.; Soto-Rey, I.; Kramer, F. Towards a guideline for evaluation metrics in medical image segmentation. *BMC Res. Notes* **2022**, *15*, 210. [CrossRef]
31. Taha, A.A.; Hanbury, A. Metrics for evaluating 3D medical image segmentation: Analysis, selection, and tool. *BMC Med. Imaging* **2015**, *15*, 29. [CrossRef]
32. Zou, K.H.; Warfield, S.K.; Bharatha, A.; Tempany, C.M.C.; Kaus, M.R.; Haker, S.J.; Wells Iii, W.M.; Jolesz, F.A.; Kikinis, R.; St, F. Statistical Validation of Image Segmentation Quality Based on a Spatial Overlap Index 1: Scientific Reports; 2004; Volume 11. Available online: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1415224/ (accessed on 10 December 2022).
33. Varoquaux, G.; Cheplygina, V. Machine learning for medical imaging: Methodological failures and recommendations for the future. *NPJ Digit. Med.* **2022**, *5*, 1–8. [CrossRef]
34. Liver Segmentation–3D-Ircadb-01-IRCAD. Available online: https://www.ircad.fr/research/data-sets/liver-segmentation-3d-ircadb-01/ (accessed on 20 June 2022).
35. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *arXiv* **2017**, arXiv:1704.04861.
36. Zhang, X.; Zhou, X.; Lin, M.; Sun, J. ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices. *arXiv* **2017**. Available online: https://arxiv.org/pdf/1707.01083 (accessed on 12 December 2022).