Received: March 9, 2022. Revised: May 30, 2022. 

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

# The Efficacy of U-Net in Segmenting Liver Tumors from Abdominal CT Images

**Deepak Jayaprakash Doggalli[1]*** **Sunil Kumar B S[2]**

[1]*Department of Computer Science and Engineering, RV Institute of Technology and Management, Bangalore, India*
[2]*Department of Information Science and Engineering, GM Institute of Technology, Davangere, India*
* Corresponding author's Email: deepakdoggalli@gmail.com

**Abstract:** One of the primary causes of cancer-related mortality is liver cancer. Computed tomography (CT) is a commonly utilised imaging technique for the assessment and staging of hepatic tumors. Manually scanning volumetric CT scans is time intensive and ambiguous. Despite the fact that numerous deep learning models for semantic segmentation have been developed, the U-Net model remains extremely successful. In this work, we propose a method for segmenting liver tumors from abdominal CT images that is entirely based on the U-Net model and demonstrate the model's simplicity and efficacy. The U-Net architecture is employed at two levels. The first level of segmentation is used to segment the liver from CT slices, while the second level is utilised to segment tumours from masked CT images. Using CT scans from the LiTS dataset, the proposed technique achieved a dice global score of 94 percent for liver segmentation and 73 percent for tumour segmentation.

**Keywords:** Liver tumor segmentation, Computed tomography, Deep learning, U-Net.

## 1. Introduction

One of the fastest-spreading cancers in the world is liver cancer. According to a statistical study published in 2019 [1], there were 841080 new cases of liver cancer in 2018, with Asia having the highest incidence and mortality rates. The majority of cases go unreported. The majority of liver cancers go undiagnosed or are detected too late to offer treatment. Early detection of liver tumors is always beneficial in terms of providing effective treatment.

Liver tumor segmentation is a real challenge due to the subtle differences between normal and tumor tissues. Segmentation is impossible when using low contrast Computed Tomography images. As a result, segmentation is performed using contrast enhanced CT scans. Increased contrast also increases image noise due to the anatomical differences between tissues and tumors.

Typically, radiologists delineate tumors manually from CT images. Manual segmentation varies according to the individual and the subtlety of the tumors. Numerous studies are being conducted in the field of liver tumor segmentation. Various semi-automatic and automatic liver tumor segmentation techniques have been proposed to minimize manual intervention. The accuracy of most segmentation methods is enhanced through the use of various pre-processing and post-processing techniques. Many such promising methods are based on deformable models, thresholding models, region growing models, graph cut models, statistical shape models, and neural network models. The majority of traditional segmentation models are based on prior knowledge and tissue characteristics. As a result, these techniques are unreliable for performing accurate segmentation. Deep learning models based on ResNet, AlexNet, FCNs, DenseUnets, MA-Net, and their variants improve tumour detection and segmentation accuracy significantly, but they also require a significant amount of processing power [2].

In this work, we present a simple yet effective deep learning model based on U-Net that can automatically segment liver tumours from CT images while consuming very little computational power. With the proper hyperparameter settings, the U-Net model can be effectively utilised to accurately segment tumours from CT images. The

proposed system is composed of two cascading U-Nets with distinct hyperparameter settings. In order to avoid vanishing gradient or exploding gradient problems, Kiaming initialization is used for initialising the weights in the first U-Net. The second U-Net employs a multi-class cross entropy loss function, which is effective for segmentation tasks. In comparison to other deep learning approaches that use a multiplicity of layers and demand significant processing power with similar accuracy, the suggested method is simple, effective, and consumes minimum resources and time.

The remainder of this work is organized in the following manner. In section 2, we provide a concise overview of existing methods for segmenting the liver and liver tumor from abdominal CT images. The suggested technique is explained in depth in section 3. In section 4, the experimental design and outcomes are presented in detail. The conclusion is presented in section 5.

## 2. Literature review and related work

The U-Net, ResNet, AlexNet, and VGG-net models, as well as versions of these models, account for the vast majority of effective deep learning algorithms for semantic segmentation of the liver and tumor from abdominal CT images.

The first of which was proposed by the Automatic Liver and Lesion Segmentation method by [3] and later in [4] which are based on Cascaded Fully Convolutional Neural Networks and dense 3D Conditional Random Fields. U-Net architecture was used to implement FCNs. Refining was done using 3D CRF that improves both appearance and spatial coherence as part of the post processing step. The method was applied on the medical image database provided by 3DIRCADb dataset (3d Image Reconstruction for Comparison of Algorithm Database).

In [5], the researcher proposed an automated liver tumor segmentation based on DCNN. This model was based on Fully Convolutional neural networks (FCNs). This model uses both U-Net [6] and ResNet [7] architectures based on FCN. Another fully automatic approach based on FCN was proposed by [8]. The lengthy training period and less precise segmentation are the main drawbacks of these techniques.

In [9], the author presented an automatic liver tumor detection method based on cascaded deep residual networks (ResNets). The method overcomes the drawbacks of the traditional FCNs by using ResNets that uses skip connections or residual connections between Convolutional layers to solve the problem of model degradation in deep neural networks. The method proposed by [10] uses CNNs built on the 2D UNet architecture along with random forest classifier to segment liver lesions.

The method proposed by [11] called as the Hybrid Densely Connected Unet (H-DenseUNet) was ranked first in LiTs 2017 segmentation challenge at the time of submission. The proposed method uses a cascade of three layers. The first layer identifies the inter-slice and intra-slice features using 2D DenseUnet followed by a 3D DenseUnet with 65 convolutional layers to extract hybrid features. On two NVIDIA Titan Xp GPUs with 12 GB of RAM, this approach took 60 hours of training to converge. The high computational cost and lengthy training period are also downsides of this approach.

In [12], the researcher has developed a three-dimensional FCN made up of several Attention Hybrid Connection Blocks (AHCBlocks) to segment liver tumors from CT images. The new network model proposed called the AHCBlock contains the Attention Gate (AG) unit and the hybrid connection unit. The AG module highlights the important features in the input image using long skip connections and removes irrelevant regions. The lower level as well as the higher-level features can be combined together using the AG module. Hybrid connection module is used for short skip connections using the lower level and output feature map which improves the flow of information. On the LiTs dataset, the findings reveal a segmentation accuracy of only 59 percent, suggesting the model's low generalisation capabilities.

A model based on multiple deep encoder-decoder CNN (EDCNN) was proposed by [13]. The model architecture was based on SegNet and VGG-16 models which consists of encoder-decoder part to construct a symmetrical structure and a last layer for pixel wise classification. A similar approach was used by [14] using a modified SegNet model. Both algorithms used the 3DIRCADb dataset for training and were unable to obtain the expected tumour segmentation results with many false positives.

Another novel approach based on using the 3D fractal residual network (3D FRN), multi-scale candidate generation method (MCG) and active contour model (ACM) was proposed by [15]. In this approach, liver segmentation was done using a 3D U-Net model, followed by tumor candidate generation using MCG. Classification of tumor candidates was done by combining the fractal structure and residual structure using 3D FRN. The problem with this approach is that tumour boundaries are imprecise, and numerous

neighbouring tumours may be segmented into a single tumour area.

[16] proposed a method based on modified U-Net (mU-Net). The residual path and the skip connections were modified in the standard U-Net model, which improved the feature extraction process. The approach was unable to capture structural similarities sufficiently since it employed mean square error as the loss function. The work in [17] also used a similar approach of modified U-Net for segmentation. The method lowered the number of filters across each convolutional unit and added an additional dropout layer and batch normalization after each convolutional block. Even with additional data augmentation, the method was able to achieve a dice score of only 0.63 in liver tumor segmentation.

In [18], the author has proposed a model that alters the original U-Net encoding process by incorporating a dense module, an inception module, and a dilated convolution to improve information flow and increase the receptive field.

This method additionally takes advantage of the bottleneck feature supervised (BS) U-Net, which bridges the gap by using U-Net as an auto-encoder and collecting additional information from the bottleneck layer for supervision. Despite the fact that training required very little time, the approach could only attain a tumour segmentation accuracy of 56.9% dice per case.

In [19], the researcher has suggested a model based on U-Nets that was refined using the level set technique and fuzzy c-means clustering. The network design was composed of 2D U-Net models and 3D FCN models, each with its own encoder-decoder structure and skip connections. The results of the investigation on the LiTS dataset indicated a Dice per case of 71.8%. The disadvantage of this method is that holes are created during liver segmentation, which may impact tumour segmentation accuracy.

In [20], the author presented a Multi-scale Attention Network (MA-Net), which utilizes a self-attention technique to adaptively integrate local characteristics with their global links. The design included two blocks: a Position-wise Attention Block (PAB) for capturing the spatial relationships between pixels in a global perspective, and a Multi-scale Fusion Attention Block (MFAB) for capturing the channel relationships between any feature maps via multi-scale semantic feature fusion. The problem with this strategy is that the addition of attention blocks complicates the model without significantly improving its precision.

The UV-Net approach suggested by [21] combines U-Net and V-Net with multi-scale feature extraction. V-Net was utilised to prevent inter-layer information loss, which is common in 3D medical picture processing. The V-Net, which is prone to overfitting difficulties, is a downside of this approach.

[22] proposes a model with three layers: Spatial ConvNet for extracting intra-slice features, Temporal ConvNet for obtaining inter-slice features, and a squeeze-and-excitation layer to help direct these features correctly. The addition of these extra layers increases the computational cost without significantly enhancing the accuracy.

In [23], the author proposes a model based on a self-attention module and a deep attention neural network. A high-resolution branch is also used to preserve spatial features. The disadvantage of this method is that it ignores inter-slice contextual information.

[24] suggested the Hybrid Dense X-net model, which consists of a 2D dense U-Net layer followed by a 3D dense U-Net layer cascaded together. Despite the model's ability to attain high segmentation accuracy, its complexity resulted in longer training times.

In [25], a new model called Un-net was created, in which the convolution node was altered to transmit the output characteristics of one layer to the next layer and skip connections. Dilated convolution without a pooling function was also used in this study. The drawback with this method is that using many convolution layers in each encoder and decoder increases the model's complexity and training time.

U-ADenseNet, a model based on U-Net and DenseNet, was described in [26]. Atrous spatial pyramid pooling is used in this method to capture image context at different scales. The HFRU-Net framework was proposed in [27], and it included modified U-Net, Squeeze-and-Excitation networks, and an atrous spatial pyramid pooling module. Both of these strategies involve extensive training.

Almost all of these deep learning approaches have made their structures more complicated by combining models like U-Net, V-Net, AlexNet, ResNet, DenseUnet, MA-Net and others. Furthermore, because these approaches use numerous levels, they may require a substantial amount of computer resources and training time. The major purpose of this research is to find a good balance between model simplicity and training time. This is accomplished through the use of a simplified variant of the basic U-Net model and appropriate hyperparameter settings. The configurations include the uniform application of the Leaky ReLu activation function, Kaiming Initialization [28], L2

regularisation, suitable batch normalisation parameters, and feature dropout.

## 3. Methodology

### 3.1 Overview of the framework

In the first stage, all slices were pre-processed by windowing Hounsfield unit values between -100 and 400 and increasing the contrast by histogram equalization. The proposed framework is based on the application of two tiers of the basic U-Net model [8] as indicated in Fig 1. The first U-Net is utilized to segment the liver from abdominal CT slices, and the second U-Net is utilized to segment tumors from masked CT slices.

### 3.2 U-net model

The U-Net model is utilized at two levels. The network architecture is U-shaped, with a downward moving path followed by an upward-moving path. The downward path is composed of four blocks. Each block has two convolution layers with the Leaky ReLu activation function, succeeded by a max pooling layer with stride 2. In the first block, the two convolution layers use 32 filters each, with a filter size of 3x3. In the upward path, the up sampling was done using 2x2 transposed convolution, and the feature map from the corresponding downward path was concatenated. The upward path also contains four blocks. Each block contains two convolution layers with 3x3 filters and a Leaky ReLu activation function. The Leaky Relu activation function is used because it is inexpensive to compute, converges quickly, and is sparsely activated. If the function receives negative input, it returns 0, but for positive input, it returns the input value as shown in Eq. (1), where x is the input.

$$f(x)=\max(0,x) \tag{1}$$

While the same U-Net model is utilized at both levels, the hyper parameter values are different at each level.

The first U-Net (U-Net-1) uses a filter size of 3X3. The weights are initialized using Kaiming Initialization [28]. Kaiming Initialization is a method for initializing neural networks that takes the nonlinearity of activation functions, such as ReLU activations, into consideration. This approach avoids exponentially lowering or magnifying the size of input signals which might lead to vanishing or exploding gradients problem. A response of each

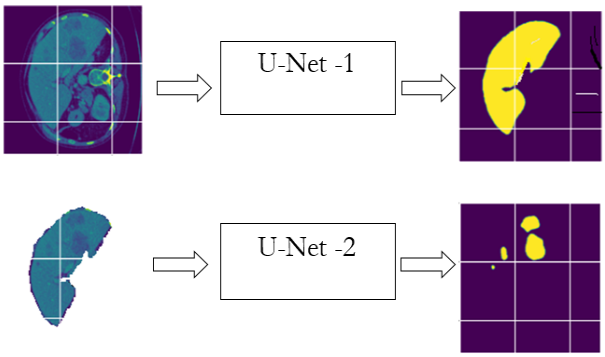


Figure. 1 U-Net based architecture for Liver tumor segmentation

convolution layer can be represented as given in Eq. (2).

$$y_l = W_l\, x_l + \, b_l \tag{2}$$

Here, x is a $k^2c$ x 1 vector that represents co-located k x k pixels in c input channels. k is the spatial filter size of the layer. n = $k^2c$ is the number of connections of a response. W is a d x n matrix, where d is the number of filters and each row of W represents the weights of a filter. b is a vector of biases, and y is the response at a pixel of the output map. $x_l = f(y_{l-1})$ where f is the activation and $l$ is the layer index.

The sufficient condition to avoid gradients problem is given in Eq. (3)

$$\tfrac{1}{2}\, n_l \mathrm{Var}[w_l] = 1 \qquad \forall l \tag{3}$$

Where $w_l$ is random variable of each element in $W_l$ and Var represents variance.

This gives an initialization of the form shown in Eq. (4):

$$w_l \sim \mathcal{N}(0, 2/\, n_l) \tag{4}$$

That is, a zero-centered Gaussian with standard deviation of $\sqrt{2/n_l}$.

The U-Net-1 setup uses L2 regularization to avoid over fitting. Regularization essentially increases the penalty as the complexity of the model increases. Regularization term lambda(λ) penalizes all parameters except the intercept, ensuring that the model adequately generalizes the data and does not over fit. L2 regularization is advantageous when co-dependent features exist. L2 regularization (R) is represented in Eq. (5). Here $W$ indicates the weight matrix with indices $i$ and $j$.

$$R(W) = \sum_i \sum_j W_{i,j}^2 \tag{5}$$

The U-Net-1 also uses drop out of 0.2 to avoid over fitting. Dropout is a regularization technique for neural networks that aids in decreasing interdependent learning among the neurons.

The loss function used is Multi class cross entropy loss which is the commonly used in classification and segmentation tasks. This is used to quantify the performance of a classification model whose output is a probability value ranging from 0 to 1. As the expected probability diverges from the actual label, the cross-entropy loss grows. Multi class cross entropy loss (L) is shown in Eq. (6). Here $y$ is the target value, $\hat{y}$ is the predicted value and $y^{(k)}$ is 0 or 1 based on the correct prediction of label k.

$$L(\hat{y}, y) = -\sum_{k}^{K} y^{(k)} \log \hat{y}^{(k)} \quad (6)$$

Batch normalization layer is used that helps each layer of the network to operate independently of the others. This is used to normalize the preceding layers' output. In U-Net-1, 90% of the dataset is used for training and 10% for validation.

The second U-Net (U-Net-2) also uses L2 regularization and a drop out of 0.2. The loss function used is Multi class cross entropy loss over Multiclass Dice Loss. The setup uses a batch size of 4 and batch normalization is used to avoid over fitting. 90% of the dataset is used for training and 10% for validation. The training dataset is expanded using a variety of data augmentation techniques, including rotation, magnification, horizontal flipping, and vertical flipping.

The detailed illustration of the structure of the U-Net model used is shown in Table 1. The five-level U-Net network consists of a contracting path followed by an expansive path. Each column in Table 1 shows the sequence of operations applied at each level. The table shows in detail the size of the filter, convolution parameter sizes, batch normalization parameter sizes (BatchNorm), activation functions (Leaky Relu, Softmax), dropout, and convolution transpose parameter sizes.

## 4. Experiments and results

### 4.1 Dataset

The model was evaluated using 200 abdominal CT images in NIfTI format from the Liver Tumor Segmentation Challenge (LiTS) of MICCAI 2017 [29]. Among the 200 CT images, 130 were used for training (containing liver and lesion masks) and 70 were used for evaluating the method (without masks). Out of 130, 105 were used for actual training and another 25 were for validation purposes.

### 4.2 Performance evaluation metrics

Typically, the outputs of the segmentation process are validated against ground truth offered by various challenges. Different criteria are used to assess segmentation accuracy and performance. The most widely used statistical metrics are based on several research publications [2], and the various issues are outlined below.

Considering A as the segmented region and B as the ground truth, the definitions of the different metrics are explained below.

Volumetric Overlap Error (VOE) can be calculated by dividing the total number of pixels in an inter-section of a segmented region by the total number of pixels in the ground truth, and the total number of pixels in a union of the segmented region and the ground truth by the total number of pixels in the segmented region. Values close to zero indicate a successful segmentation, while increases in the score indicate differences between segmented images. VOE can be calculated using the formula given below in Eq. (7):

$$VOE = ( ( |A \cap B| / |A \cup B| ) - 1 ) \times 100 \quad (7)$$

The Dice Similarity Coefficient (DSC) measures the segmentation of pixels in a region of interest. A decent segmentation should have values that are relatively close to one, whereas a score close to 0 indicates differences in the segmented picture. DSC can be calculated using the formula given in Eq. (8):

$$DSC = 2 |A \cap B| / ( |A| + |B| ) \quad (8)$$

Relative Volume Difference (RVD) gives the ratio of the total volume of the segmented region and the ground truth. Under segmentation should give negative values and over segmentation should give positive values. RVD is calculated using the following formula given in Eq. (9):

$$\text{Relative Volume Difference} = (( \text{total volume of the segmented region} / \text{total volume of the ground truth}) - 1) \times 100 \quad (9)$$

In Average Symmetric Surface Distance (ASD), the voxels in the borders of segmented region and the ground truth are determined. For every voxel along segmented region border, the closest voxel along the ground truth border is determined. ASD is the average of all these distances calculated

Table 1. Detailed architecture of the proposed U-Net model

| Contracting path (left side) | | | | | Expansive path (right side) | | | |
|---|---|---|---|---|---|---|---|---|
| Level 1 | Level 2 | Level 3 | Level 4 | Level 5 | Level 4 | Level 3 | Level 2 | Level 1 |
| Convolution Param Sizes: (16, 3, 3, 3) (16) | MaxPool2D WithIndices | MaxPool2D WithIndices | MaxPool2D WithIndices | MaxPool2 DWithInd ices | Concat enation | Concat enation | Concat enation | Concat enation |
| BatchNorm Param Sizes:(16) (16) | BatchNorm Param Sizes:(32) (32) | BatchNorm Param Sizes:(64) (64) | BatchNorm Param Sizes:(128) (128) | BatchNorm Param Sizes:(256) (256) | BatchNo rm Param Sizes:(512) (512) | BatchNo rm Param Sizes:(256) (256) | BatchNo rm Param Sizes:(128) (128) | BatchNo rm Param Sizes:(64) (64) |
| Leaky_Relu | Convolution Param Sizes:(64, 32, 3, 3) (64) | Convolution Param Sizes:(128, 64, 3, 3) (128) | Convolution Param Sizes:(256, 128, 3, 3) (256) | Convoluti on Param Sizes:(512, 256, 3, 3) (512) | Convolut ion Param Sizes:(256, 512, 3, 3) (256) | Convolut ion Param Sizes:(128, 256, 3, 3) (128) | Convolut ion Param Sizes:(64, 128, 3, 3) (64) | Convolut ion Param Sizes:(32, 64, 3, 3) (32) |
| BatchNorm Param Sizes: (16) (16) | Leaky_Relu | Leaky_Relu | Leaky_Relu | Leaky_Re lu | Leaky_R elu | Leaky_R elu | Leaky_R elu | Leaky_R elu |
| Convolution Param Sizes:(32, 16, 3, 3) (32) | Feature Dropout | Feature Dropout | Feature Dropout | Feature Dropout | Feature Dropout | Feature Dropout | Feature Dropout | Feature Dropout |
| Leaky_Relu | BatchNorm Param Sizes:(64) (64) | BatchNorm Param Sizes:(128) (128) | BatchNorm Param Sizes:(256) (256) | BatchNorm Param Sizes:(512) (512) | BatchNo rm Param Sizes:(256) (256) | BatchNo rm Param Sizes:(128) (128) | BatchNo rm Param Sizes:(64) (64) | BatchNo rm Param Sizes:(32) (32) |
| Feature Dropout | Convolution Param Sizes:(64, 64, 3, 3) (64) | Convolution Param Sizes:(128, 128, 3, 3) (128) | Convolution Param Sizes:(256, 256, 3, 3) (256) | Convoluti on Param Sizes:(512, 512, 3, 3) (512) | Convolut ion Param Sizes:(256, 256, 3, 3) (256) | Convolut ion Param Sizes:(128, 128, 3, 3) (128) | Convolut ion Param Sizes:(64, 64, 3, 3) (64) | Convolut ion Param Sizes:(32, 32, 3, 3) (32) |
| BatchNorm Param Sizes: (32) (32) | Leaky_Relu | Leaky_Relu | Leaky_Relu | Leaky_Re lu | Leaky_R elu | Leaky_R elu | Leaky_R elu | Leaky_R elu |
| Convolution Param Sizes:(32, 32, 3, 3) (32) | Feature Dropout | Feature Dropout | Feature Dropout | Feature Dropout | Feature Dropout | Feature Dropout | Feature Dropout | FeatureD ropout |
| Leaky_Relu | | | | Convoluti on Transpose Param Sizes:(512, 256, 4, 4) (256) | Convolut ion Transpos e Param Sizes:(256, 128, 4, 4) (128) | Convolut ion Transpos e Param Sizes:(128, 64, 4, 4) (64) | Convolut ion Transpos e Param Sizes:(64, 32, 4, 4) (32) | Convolut ion Param Sizes:(2, 32, 1, 1) (2) |
| Feature Dropout | | | | | | | | Softmax |

using Euclidean distance. Perfect segmentation is represented by a value of 0 mm.

Root Mean Square Symmetric Surface Distance (RMSD) is a variation of ASD where the root value of the ASD is used. Perfect segmentation is represented by a value of 0 mm.

Maximum symmetric surface distance (MSD) is also a variation of ASD where the maximum of all distances between voxels of segmented and ground truth border voxels are used. Perfect segmentation is represented by a value of 0 mm.

### 4.3 Experimental setup

Training was done on Dell R430 Bay Server 2 E5 2603v2, 128 GB RAM with a Tesla p100 16GB GPU. CPU utilization went up to 20 GB and about 8 GB of GPU memory was utilized.

## 5. Experiments and results

### 5.1 Liver segmentation training

Liver segmentation training was done for 20 epochs and took around 25 hours to complete as shown in Fig 2. Table 2 shows the time taken, training loss, training dice and validation dice during the different epochs.

Table 2. Details of liver segmentation training

| Epoch | Time (hours) | Training Loss | Training Dice | Validation Dice |
|---|---|---|---|---|
| 0 | 1.2 | 0.755994 | 0.821975 | 0.902976 |
| 1 | 1.2 | 0.721102 | 0.889706 | 0.911630 |
| 2 | 1.2 | 0.715543 | 0.905051 | 0.9292170 |
| 3 | 1.2 | 0.711219 | 0.915280 | 0.932751 |
| 4 | 1.19 | 0.708147 | 0.922648 | 0.917786 |
| 5 | 1.19 | 0.706152 | 0.928129 | 0.946567 |
| 6 | 1.19 | 0.704345 | 0.933088 | 0.93837 |
| 7 | 1.19 | 0.702488 | 0.936639 | 0.950231 |
| 8 | 1.19 | 0.701034 | 0.940624 | 0.950548 |
| 9 | 1.19 | 0.700426 | 0.943203 | 0.933103 |
| 10 | 1.19 | 0.700206 | 0.944979 | 0.948467 |
| 11 | 1.19 | 0.698753 | 0.947725 | 0.942909 |
| 12 | 1.2 | 0.698252 | 0.948305 | 0.955342 |
| 13 | 1.2 | 0.697388 | 0.950206 | 0.952161 |
| 14 | 1.19 | 0.696774 | 0.950762 | 0.938993 |
| 15 | 1.2 | 0.696675 | 0.951715 | 0.936448 |
| 16 | 1.19 | 0.696622 | 0.952992 | 0.946203 |
| 17 | 1.2 | 0.695863 | 0.952538 | 0.949968 |
| 18 | 1.19 | 0.695617 | 0.954981 | 0.954899 |
| 19 | 1.19 | 0.695447 | 0.955852 | 0.954835 |

### 5.2 Lesion segmentation training

Training was done for 25 epochs and took around 12 hours to complete as shown in Fig 3. Table 3 shows the time taken, training loss, training dice and validation dice during the different epochs.

### 5.3 LiTS results

The validation set provided by the LiTS challenge was checked with the designed architecture. The lesion masks created were submitted to the Coda Lab LiTS Competition website (with Username - Deepak).

The following results were obtained for the liver segmentation: Dice per case of 0.9380, Dice global of 0.9400, VOE of 0.116, RVD of -0.072, ASSD of 2.084, MSD of 32.802 and RMSD of 3.960.

The following results were obtained for the liver tumor segmentation: Dice per case of 0.5840, Dice global of 0.7300, VOE of 0.424, RVD of -0.168, ASSD of 1.387, MSD of 7.782 and RMSD of 2.052.

## 6. Comparison with related work

Table 4 compares our findings to those obtained using state-of-the-art approaches applied to the LiTS dataset and published in other papers. (* indicates results are not submitted to the LiTs competition website. Results with italic font indicate that the method uses variations of U-Net model. -- indicates results not published.) In comparison to other strategies that need a huge number of layers and extensive training, our basic model requires very little training and performs on par with the others. The proposed method's efficacy stems from the U-Net model's simplicity and the use of proper hyperparameter settings. The results illustrate the

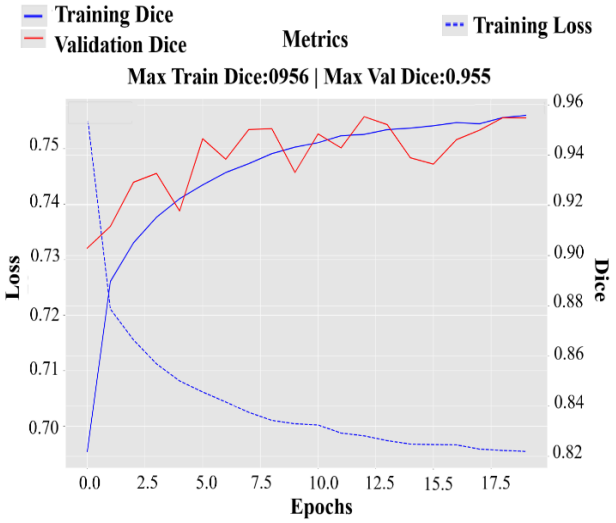


Figure. 2 Liver segmentation training performance

efficacy of the U-Net model in the biomedical image segmentation process. Some of the sample liver and lesion segmentation results are shown in Fig. 4 and 5.

Table 3. Details of Lesion segmentation training

| Epoch | Time (hours) | Training Loss | Training Dice | Validation Dice |
|---|---|---|---|---|
| 0 | 0.434046 | 1.448718 | 0.704702 | 0.649292 |
| 1 | 0.416835 | 1.406488 | 0.758737 | 0.668117 |
| 2 | 0.408057 | 1.391843 | 0.779482 | 0.644747 |
| 3 | 0.406941 | 1.386037 | 0.788345 | 0.660027 |
| 4 | 0.405036 | 1.381469 | 0.791324 | 0.627486 |
| 5 | 0.411827 | 1.378742 | 0.799279 | 0.714896 |
| 6 | 0.405320 | 1.372064 | 0.803000 | 0.706388 |
| 7 | 0.404948 | 1.368929 | 0.805757 | 0.648888 |
| 8 | 0.405751 | 1.367342 | 0.810938 | 0.675198 |
| 9 | 0.405105 | 1.363973 | 0.813065 | 0.687165 |
| 10 | 0.405161 | 1.366202 | 0.812975 | 0.690613 |
| 11 | 1.669217 | 1.362061 | 0.812927 | 0.688881 |
| 12 | 0.412667 | 1.359462 | 0.820076 | 0.667424 |
| 13 | 0.406653 | 1.359249 | 0.818985 | 0.663633 |
| 14 | 0.407001 | 1.357430 | 0.819083 | 0.693245 |
| 15 | 0.406374 | 1.353592 | 0.825494 | 0.659096 |
| 16 | 0.406378 | 1.357945 | 0.820551 | 0.690153 |
| 17 | 0.405654 | 1.353387 | 0.826103 | 0.714149 |
| 18 | 0.404935 | 1.350716 | 0.829779 | 0.648876 |
| 19 | 0.405034 | 1.351276 | 0.824490 | 0.699840 |
| 20 | 0.405233 | 1.350913 | 0.829195 | 0.697623 |
| 21 | 0.405276 | 1.350635 | 0.831178 | 0.676359 |
| 22 | 0.411750 | 1.351804 | 0.827172 | 0.727283 |
| 23 | 0.405151 | 1.347117 | 0.830887 | 0.658026 |
| 24 | 0.405202 | 1.349376 | 0.831404 | 0.697593 |

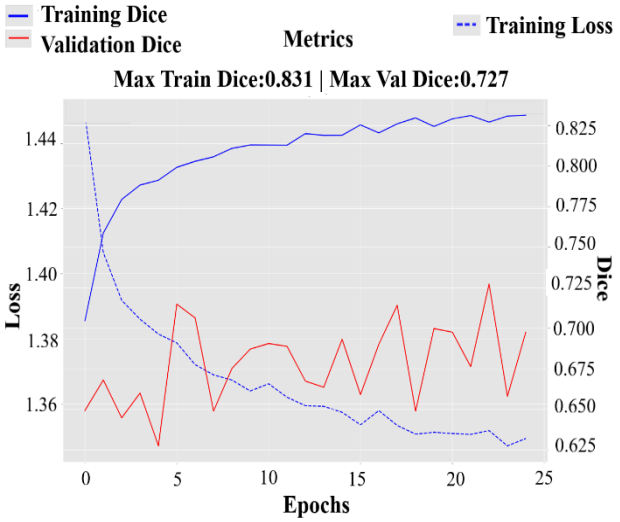


Figure. 3 Lesion segmentation training performance

Table 4. Comparison of liver tumor segmentation results (LiTS) of different techniques

| Approach | Data set | VOE | RVD | ASD (mm) | MSD (mm) | DICE Global (%) |
|---|---|---|---|---|---|---|
| [4] | LiTs | 0.45 | 0.04 | 6.66 | 57.9 | 67 |
| [9] | LiTs | -- | -- | -- | -- | 64 |
| [10] | LiTs | -- | -- | -- | -- | 65 |
| [11] | *LiTs* | -- | -- | -- | -- | *72.2* |
| [12] | 3Dircadb | 1.35 | 0.129 | 1.07 | 6.27 | 73.4 |
| [13] | 3Dircadb | -- | -- | -- | -- | 63.4 |
| [14] | 3Dircadb | -- | -- | -- | -- | 86 |
| [15] | 3Dircadb | 0.32 | 0.194 | 4.40 | 7.11 | 76.4 |
| [17] | *LiTs* | -- | -- | -- | -- | *63* |
| [18] | *LiTs* | -- | -- | -- | -- | *56.9* |
| [19] | LiTs | 0.40 | 0.25 | 3.04 | 4.99 | 72.45 |
| [20] | LiTs | 0.21 | -0.18 | -- | -- | 74.9 |
| [21] | LiTs | -- | *-0.045* | *7.52* | -- | 84.4 * |
| [22] | LiTs | 0.36 | -0.072 | -- | 6.22 | 82.4 * |
| [23] | LiTs | -- | -- | -- | -- | 76.3 |
| [24] | LiTs | 0.35 | -0.369 | 0.96 | 5.40 | 84.3 * |
| [25] | LiTs | *37.8* | *-15.78* | -- | -- | *73.6* |
| [26] | LiTs | 0.35 | -0.124 | -- | -- | 74.5 |
| [27] | LiTs | 0.38 | 0.223 | 1.24 | 7.42 | 61.4 |
| Our method | LiTs | **0.42** | **0.168** | **1.38** | **7.78** | **73** |

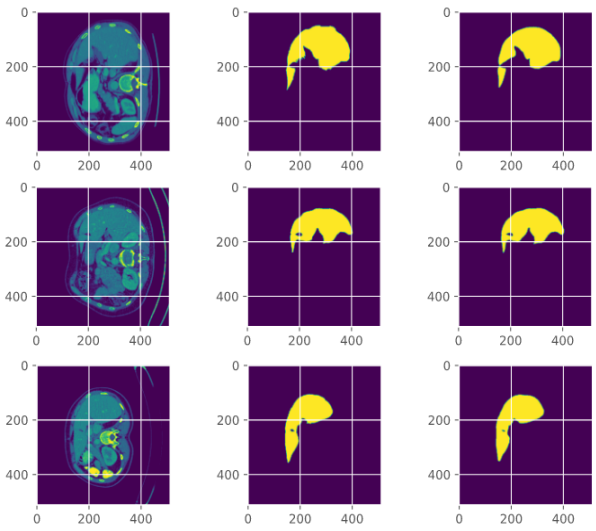

Figure. 4 Liver segmentation sample results

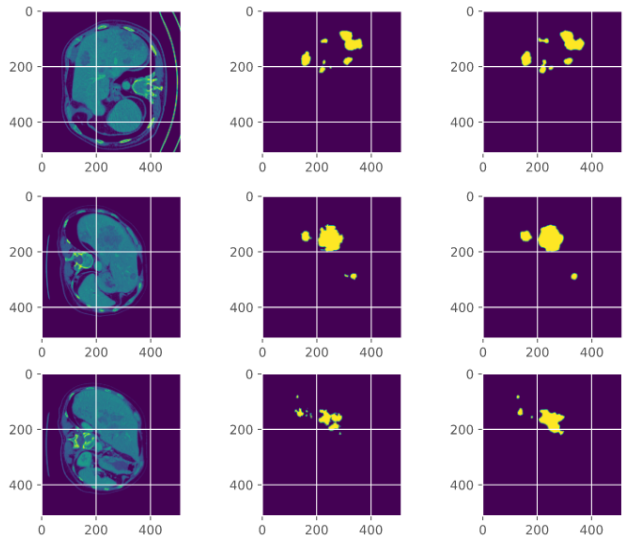

Figure. 5 Liver tumor segmentation sample results

## 7. Conclusion

In this paper, we demonstrated the U-Net model's simplicity and effectiveness in semantic biomedical image segmentation. The proposed architecture based on the variation of the basic U-Net model can be utilized to effectively segment liver and tumors from abdominal CT images. We can achieve comparable accuracy to complex state-of-the-art methods by adjusting the core U-Net model's hyper parameter settings according to the kind of dataset. The model was evaluated using the LiTS dataset and a liver tumor segmentation accuracy of 73 % (Dice Global) was achieved. The segmented results from the validation set were submitted to the LiTS website for verification, and the results demonstrate the efficacy of our approach. Despite of the limited dataset, limited resources and the less training time, the model was able to achieve very good accuracy.

Other medical modalities, such as MRI and ultrasound, can also be segmented using the suggested methodology. As part of our ongoing research, we will continue to investigate new deep learning models in conjunction with the U-Net model in order to further improve segmentation accuracy. Automatic segmentation of tiny liver tumor patches remains a major challenge.

## Conflicts of Interest

The authors declare no conflict of interest.

## Author Contributions

"Conceptualization, Deepak D J and Sunil Kumar B S; methodology, Deepak D J; software, Deepak D J; validation, Deepak D J and Sunil Kumar B S; formal analysis, Deepak D J; investigation, Deepak D J; resources, Deepak D J; data curation, Deepak D J; writing—original draft preparation, Deepak D J; writing—review and editing, Deepak D J and Sunil Kumar B S; visualization, Deepak D J; supervision, Sunil Kumar B S;

## References

[1] E. Goodarzi, F. Ghorat, A. M. Jarrahi, H. A. Adineh, M. Sohrabivafa, and Z. Khazaei, "Global incidence and mortality of liver cancers and its relationship with the human development index (HDI): an ecology study in 2018", *World Cancer Research Journal*, Vol. 6, No. April 2019, Apr. 2019, doi: 10.32113/wcrj_20194_1255.

[2] M. Moghbel, S. Mashohor, R. Mahmud, and M. I. B. Saripan, "Review of liver segmentation and computer assisted detection/diagnosis methods in computed tomography", *Artif Intell Rev*, Vol. 50, No. 4, pp. 497-537, Dec. 2018, doi: 10.1007/s10462-017-9550-x.

[3] P. F. Christ, "Automatic Liver and Lesion Segmentation in CT Using Cascaded Fully Convolutional Neural Networks and 3D Conditional Random Fields", *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2016*, pp. 415-423. doi: 10.1007/978-3-319-46723-8_48.

[4] P. F. Christ, "Automatic Liver and Tumor Segmentation of CT and MRI Volumes using Cascaded Fully Convolutional Neural Networks", *arXiv:1702.05970 [cs]*, 2017, Accessed: Mar. 05, 2022. [Online] Available: http://arxiv.org/abs/1702.05970

[5] X. Han, "Automatic Liver Lesion Segmentation Using A Deep Convolutional Neural Network Method", *Med. Phys.*, Vol. 44, No. 4, pp. 1408-1419, 2017, doi: 10.1002/mp.12155.

[6] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition", In: *Proc. of 2016 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770-778, 2016, doi: 10.1109/CVPR.2016.90.

[7] C. Sun, "Automatic segmentation of liver tumors from multiphase contrast-enhanced CT images based on FCNs", *Artificial Intelligence in Medicine*, Vol. 83, pp. 58-66, 2017, doi: 10.1016/j.artmed.2017.03.008.

[8] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation", *Medical Image Computing and Computer-Assisted Intervention*

– *MICCAI 2015*, pp. 234-241, 2015, doi: 10.1007/978-3-319-24574-4_28.

[9] L. Bi, J. Kim, A. Kumar, and D. Feng, "Automatic Liver Lesion Detection using Cascaded Deep Residual Networks", *arXiv:1704.02703 [cs]*, 2017, Accessed: Mar. 05, 2022. [Online] Available: http://arxiv.org/abs/1704.02703

[10] G. Chlebus, H. Meine, J. H. Moltz, and A. Schenk, "Neural Network-Based Automatic Liver Tumor Segmentation With Random Forest-Based Candidate Filtering", *arXiv, Jun. 27*, 2017. Accessed: Jun. 14, 2022. [Online]. Available: http://arxiv.org/abs/1706.00842

[11] X. Li, H. Chen, X. Qi, Q. Dou, C. W. Fu, and P. A. Heng, "H-DenseUNet: Hybrid Densely Connected UNet for Liver and Tumor Segmentation From CT Volumes", *IEEE Transactions on Medical Imaging*, Vol. 37, No. 12, pp. 2663-2674, 2018, doi: 10.1109/TMI.2018.2845918.

[12] H. Jiang, T. Shi, Z. Bai, and L. Huang, "AHCNet: An Application of Attention Mechanism and Hybrid Connection for Liver Tumor Segmentation in CT Volumes", *IEEE Access*, Vol. 7, pp. 24898-24909, 2019, doi: 10.1109/ACCESS.2019.2899608.

[13] Ü. Budak, Y. Guo, E. Tanyildizi, and A. Şengür, "Cascaded deep convolutional encoder-decoder neural networks for efficient liver tumor segmentation", *Medical Hypotheses*, Vol. 134, p. 109431, 2020, doi: 10.1016/j.mehy.2019.109431.

[14] S. Almotairi, G. Kareem, M. Aouf, B. Almutairi, and M. A. M. Salem, "Liver Tumor Segmentation in CT Scans Using Modified SegNet", *Sensors (Basel)*, Vol. 20, No. 5, p. 1516, 2020, doi: 10.3390/s20051516.

[15] Z. Bai, H. Jiang, S. Li, and Y. D. Yao, "Liver Tumor Segmentation Based on Multi-Scale Candidate Generation and Fractal Residual Network", *IEEE Access*, Vol. 7, pp. 82122-82133, 2019, doi: 10.1109/ACCESS.2019.2923218.

[16] H. Seo, C. Huang, M. Bassenne, R. Xiao, and L. Xing, "Modified U-Net (mU-Net) with Incorporation of Object-Dependent High Level Features for Improved Liver and Liver-Tumor Segmentation in CT Images", *IEEE Trans. Med. Imaging*, Vol. 39, No. 5, pp. 1316-1325, 2020, doi: 10.1109/TMI.2019.2948320.

[17] Y. A. Ayalew, K. A. Fante, and M. A. Mohammed, "Modified U-Net for liver cancer segmentation from computed tomography images with a new class balancing method", *BMC Biomedical Engineering*, Vol. 3, No. 1, p. 4, 2021, doi: 10.1186/s42490-021-00050-y.

[18] S. Li, G. K. F. Tso, and K. He, "Bottleneck feature supervised U-Net for pixel-wise liver and tumor segmentation", *Expert Systems with Applications*, Vol. 145, p. 113131, 2020, doi: 10.1016/j.eswa.2019.113131.

[19] Y. Zhang, "Deep Learning Initialized and Gradient Enhanced Level-Set Based Segmentation for Liver Tumor From CT Images", *IEEE Access*, Vol. 8, pp. 76056-76068, 2020, doi: 10.1109/ACCESS.2020.2988647.

[20] T. Fan, G. Wang, Y. Li, and H. Wang, "MA-Net: A Multi-Scale Attention Network for Liver and Tumor Segmentation", *IEEE Access*, Vol. 8, pp. 179656-179665, 2020, doi: 10.1109/ACCESS.2020.3025372.

[21] C. Zhang, Q. Hua, Y. Chu, and P. Wang, "Liver tumor segmentation using 2.5D UV-Net with multi-scale convolution", *Computers in Biology and Medicine*, Vol. 133, p. 104424, 2021, doi: 10.1016/j.compbiomed.2021.104424.

[22] L. Song, H. Wang, and Z. J. Wang, "Bridging the Gap Between 2D and 3D Contexts in CT Volume for Liver and Tumor Segmentation", *IEEE J. Biomed. Health Inform.*, Vol. 25, No. 9, pp. 3450-3459, 2021, doi: 10.1109/JBHI.2021.3075752.

[23] Y. Li, B. Zou, and Q. Liu, "A deep attention network via high-resolution representation for liver and liver tumor segmentation", *Biocybernetics and Biomedical Engineering*, Vol. 41, No. 4, pp. 1518-1532, 2021, doi: 10.1016/j.bbe.2021.08.010.

[24] J. Chi, X. Han, C. Wu, H. Wang, and P. Ji, "X-Net: Multi-branch UNet-like network for liver and tumor segmentation from 3D abdominal CT scans", *Neurocomputing*, Vol. 459, pp. 81-96, 2021, doi: 10.1016/j.neucom.2021.06.021.

[25] S. T. Tran, C. H. Cheng, and D. G. Liu, "A Multiple Layer U-Net, U $^{n}$ -Net, for Liver and Liver Tumor Segmentation in CT", *IEEE Access*, Vol. 9, pp. 3752-3764, 2021, doi: 10.1109/ACCESS.2020.3047861.

[26] Y. Zhu, A. Yu, H. Rong, D. Wang, Y. Song, Z. Liu, and V. S. Sheng "Multi-Resolution Image Segmentation Based on a Cascaded U-ADenseNet for the Liver and Tumors", *Journal of personalized medicine*, Vol. 11, No. 10, p. 1044, 2021, doi: 10.3390/jpm11101044.

[27] D. T. Kushnure and S. N. Talbar, "HFRU-Net: High-Level Feature Fusion and Recalibration UNet for Automatic Liver and Tumor Segmentation in CT Images", *Computer*

*Methods and Programs in Biomedicine*, Vol. 213, p. 106501, 2022, doi: 10.1016/j.cmpb.2021.106501.

[28] K. He, X. Zhang, S. Ren, and J. Sun, "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification", *arXiv:1502.01852 [cs]*, 2015, Accessed: Mar. 05, 2022. [Online] Available: http://arxiv.org/abs/1502.01852

[29] "CodaLab - Competition." https://competitions.codalab.org/competitions/17094 (accessed Mar. 05, 2022).