Received: September 1, 2022. Revised: September 23, 2022. 526

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

A Weakly Reiterative Patches-Wise Framework for CT Liver and Lesions Segmentation

Youssef Ouassit[1]* Soufiane Ardchir[2]
Mohamed Yassine El Ghoumari[2] Mohamed Azzouazi[1]

[1]*Department of Mathematics, Faculty of Sciences, Ben M'sick, Casablanca, Morocco*
[2]*Department of Mathematics, National School of Marketing and Management, Casablanca, Morocco*
* Corresponding author's Email: ouassit.youssef@gmail.com

Abstract: Automatic Liver and lesions segmentation from volumetric computerized tomography scans has been recently an active research area in images processing field. An accurate automatic segmentation is helpful to make personalized treatment schemes and have a big impact on liver therapy planning. However, it stays a challenging task due to similar pixel intensity of liver lesions with their surrounding tissues, fuzzy borders, diverse densities, and the big variety of size, position, and shape features of liver and lesions. Recently, deep learning achieved the state of art performance in many computers vision tasks. Nevertheless, it's heavy rely on huge amount of labelled data. In medical images semantic segmentation, data annotation is time consuming and expensive to require. In this paper we propose a new framework for Liver and lesions segmentation using a weakly cascaded reiterative patches-wise convolutional neural network. A first model is used to localize object of interest and reduce the scope, the result is feed then as ROI in a second tuning network for final segmentation. To overcome the conventional methods drawbacks and provides greater retention of fine details, a multi-level patches wise training is proposed. Different dilated convolutional kernels sizes with are used in the encoder first layer to derive abundant semantic contextual features from CT scans. We also propose a new multi-level loss function for high precision. The proposed approach achieved a mean IoU score of 0,9511 for liver and 0,9471 for lesions segmentation.

Keywords: Liver segmentation, Liver lesions segmentation, Weakly supervised learning, Deep learning, Computer vision.

1. Introduction

Liver diseases are one the most common deaths cause. Liver is also a common site for secondary lesions. An automatic and fast liver and lesion segmentation from CT volumes is essential for many Liver therapy procedures. Manual segmentation is expensive due to the need of expert, time-consuming because the high number of scans per patient, prone to human error, and impractical for large datasets. Furthermore, the traditional methods based on manual feature extraction used for detection of Liver lesions are time-consuming and require experts to analyze the lesion. Therefore, developing accurate methods for automated lesions segmentation has become a necessity.

Approaches with a relatively low computing cost, such as thresholding, region growing, or clustering methods, are fast and simple to implement. However, they rely on intensity data. As a result, such methods are vulnerable to boundary leakage on blurred lesions and depends on many initialization parameters. Thus, to reduce under-segmentation or over-segmentation some prior knowledges or other algorithms were integrated and combined which make the task more complex. Recently, deep learning has become the state of art in computing vision tasks such as classification, detection, and semantic segmentation. Proposed models in literature produced promising

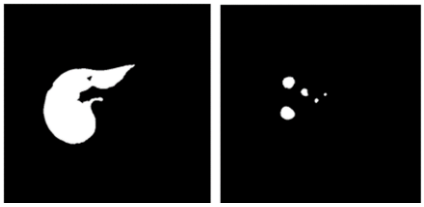
results on medical and biomedical image segmentation. However, methods based using deep models are complex and data hungry and stay limited in semantic segmentation due to the limited training data and the expensive labelling cost because the need of experts. To reduce the labour cost many semi-supervised and weakly supervised methods based especially on incomplete tags and redundant or noisy visual features have been proposed. However, such methods make good results in natural images but not suitable for medical images, for many reasons such as the need of strong annotation to combine with weak labels, and the need Conditional Random Field (CRF) in post-processing step to filter out background pixels, which is not suitable when small regions present same morphology. Furthermore, the CT images of Liver are noisy due to the degree of vascularity and the injected contrast product in acquisition stage. This introduces uncertainty and make the link between the intensity and the type of tissue ambiguous. this uncertainty suggests that the intensity value at one pixel may not accurately characterize the type of tissue contained within this pixel. Additionally, the lesions have a wide range of appearances, and neither methods nor prior knowledge exist to integrate this prior information with every possible appearance of the lesions. Additionally, due to the varied types of lesions, the changes brought on by the injection phases, and the technical variances across imaging machines, healthy and tumoral tissues within the liver have different intensities and appearances.

Liver and liver lesions segmentation challenges can be resumed in:

- The wide variety of liver, lesions and healthy tissues appearances makes difficult to distinguish between tissues.
- Uneven presence
- Fuzzy borders with neighbour organs
- Various liver and lesions densities, shapes, and sizes (see Fig. 2).
- Class unbalance between Liver and lesions pixels in volumes (see Fig. 1).

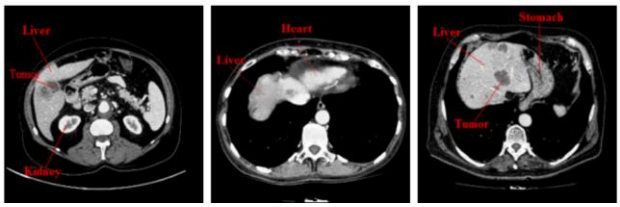
In addition, other clinical constraints must be taken in consideration:

- Robustness and reusability of the method
- Speed and on quality of detection



(a) (b)

Figure. 1 Unbalance between liver and lesions classes



(a) (b) (c)

Figure. 2 Example of challenges in liver and lesions segmentation: (a) Low-intensity difference between nearby organs, (b) Ambiguous boundary, and (c) Similar intensities with stomach

- Diverse CT machines
- Many resolutions levels

This paper, present a new framework for liver and liver lesions semantic segmentation, a cascaded patches-wise learning in a reiterative learning framework to deal with weakly labelled is proposed. The framework is built on two cascaded networks. First an Encoder-Decoder network is trained to extract the ROI, next a second iterative patches-wise model using an area threshold and custom loss to resolve imbalanced data problem and tune the final segmentation. The results show the efficiency of our proposed platform compared with several other methods, the fine details at the boundaries are well distinguished from the neighbour organs and the background due to the proposed tunning network.

The following are the main contributions of this paper:

1. A patch-based learning approach within an encoder-decoder based model to solve weak annotation issue in edges detection.
2. ASPP module with multi-scale dilated convolutional is inserted after the bottleneck of the Encoder-Decoder network to make the network scale invariant and extract context information at different scales
3. A novel multi-level loss function to deal with imbalance pixels between foreground and background pixels joining metrics of binary cross entropy and dice coefficient.
4. A multi-plane training fusion for volume segmentation for a better capture of dependencies within the 3 dimensions
5. Cascaded networks and automatic generating of ROI with architectural integration as bounding box filters architectural prior to tune segmentation reduce false positive and filling holes.

The rest of this paper is organized as follows: in Section 2, we review the relevant related works, the proposed methods and materials are presented in

Sections 3, in Section 4 numerical experiments and discussions are reported. Finally, the conclusion of the research work is represented in Section 6.

2. Related works

Many works have been proposed to segment Liver and Lesions from CT images, they can be categorized by features extraction procedures into: Hand-crafted methods and deep learning methods.

In hand craft methods, researchers focused level set [1], watershed, statistical shape mainly on developing operators such as model [2], region growing [3], active contour model [2], threshold processing [4] and graph cuts [5] and traditional machine learning methods [3, 6] that rely heavily on the quality of extracted Liver and tumors features. As instance [7] employed in semi-automatic strategy with a variety of methods, such as fuzzy C-means, the region- growing algorithm and graph cut, to segment Liver and Liver lesions. The segmentation of liver lesions using dynamic regularization of level set parameters was also proposed by [8]. To segment the liver [8] used super-pixel Simple Linear Iterative Clustering on intensity feature and an AdaBoost algorithm. [11] used the first order statistical features of the liver image to extract the CT liver boundary, and afterwards applied a k-Mean classifier based on distance and color to identify lesions. [12] used a region-growing technique to segment the tumors and identify them as benign or malignant based on the extraction of texture, shape, and kinetic curve parameters.

The Table 1 below list of the most used handcraft methods.

However, the main disadvantage and limits of the discussed hand craft methods is essentially the use of intensity information to segment objects. CT images contain often a noise occurring in acquisition stage and the lesions edges are unclear which leads to segmentation errors. On the other hand, methods based on level set can deal with these difficulties, but they are strongly depending on the good parameters and initialization. In conclusion these methods result good performance, however they can't be applied in wide clinical application because they depend to the operators and time consuming, about 30 minutes for one patient scan with 120 slices.

Table 1. Handcraft methods classified by approaches and algorithms

Segmentation approach	**Algorithms**
Gray Level Based Methods	Region Based
	Active Contour
	Level Set
	Histogram Based
	Graph Cut
	Threshold
	Clustering Based
Texture Based Methods	Wavelets
	Watershed Transform
	Pattern Recognition

Recently, deep learning had high performance in many computing vision tasks such as image detection, classification, and segmentation. Many researchers have been interested on the Liver and lesions segmentation task from CT scans. Contributions in literature can be classified into two categories. In the first category the same model is used in one step to segment Liver and lesions, as instance [9] combine FCN and a deformable models to an automatic segmentation. [15] proposed a Multiscale Combinatorial Grouping, 3D Fractal Residual Network, and Active Contour Model for liver tumors segmentation into CT volumes. [17] used a pre-processing step and a pre-trained CNN to extract a binary segmented image then a smoothing and thresholding post-processing step to refine the result. In the second category researchers propose the segmentation in multiple stages, close to our proposed method [10-13] uses two cascaded deep CNNs. The first network segment the liver and feed it as input for the training of a second network. The second network then segments the lesions in the liver's ROI from the first network. [14] proposed a framework with three models, Segnet to segment liver and lesions, a neural network with a genetic algorithm to detect the slices of the liver that have lesions and a U-Net network for lesion segmentation. Also close to our framework [16] proposed a cascade FCNs models, a localization network to localize the liver, the second network to fine tune the liver segmentation network, and the last for lesions segmentation. [18] used Cascaded deep learning models to segment the liver and lesion in more than one stage by using VGG and Segnet models. For the reason of gap between liver and lesions areas, many approaches like ours prefer to separate liver and liver lesions segmentations steps. As instance [19] segment the liver with a 3D CNN model then provide it as initial prior segmentation to segment the liver lesions.

However, even if the proposed methods reach high accuracy, the dice coefficient for small region must be improved. In addition, this method still struggles to extract fine details at the extremities to reduce the loss of slices that occurs at the boundaries of the lesions, furthermore present results are very sensitive to its input and can easily flow into neighboring organs due to the similar intensity.

Based on the advantages and limitations of the reviewed related works, we propose a new efficient fully end to end automatic framework to improve the segmentation of liver and lesions in 3D CT image scans. Therefore, we propose two cascaded deep learning networks, the first is used to extract the ROI and reduce the scope of the exam to decrease the imbalance between object of interest and background, and the second is capable of more precisely tuning segmentation and increase boundaries and fine details detection.

3. Material and methods

The Fig. 3 show a global view of our framework. Images are feed from different sources in the form of 2D slices or 3D volumes, a preparation layer allows to format all types of images, followed by a pre-processing step to adapt the images for the deep learning model, the next step consists of creating two types of datasets, one containing the images with the original size and the second containing patches of varying sizes. In the inference stage, the case where the input to be segmented is a 3D volume, volume will be cut into slices then we predict the segmentation of each slice before recreating the 3D rendering.

The cascaded models are inspired by U-Net architecture, adapted to patches-wise learning to deal with fine details in images and take in consideration the scale invariance and global context. U-net [20] have been widely used in medical and biomedical images segmentation, it consists of an encoder/decoder architecture, the encoder extract features maps from the input images using convolution and max-pooling layers while the decoder up-sample the obtained feature maps, unlike SegNet [18] where up-sampled max-pooling indices are memorized since shallow layers have maximum responses to extract boundaries, indices extracted from these responses record the location and context information, U-Net feature maps are concatenated with corresponding deep feature maps to reuse contextual and spatial information and improves localization, then at the last layer pixels are classified independently using a SoftMax activation function. However, U-Net is unable to learn fine information in global context and has weak generalization ability when annotated data is limited. To resolve these problems, we propose to cascade two models, the first one is an Encoder/Decoder network trained on full slices, used to generate Grad-CAMs for the localization information and an adapted multi-level patches input network with custom multiple level loss functions to tune the ROI segmentation, we also proposed to add an Atrous Spatial Pyramid Pooling module in the network bottleneck to handle the multiscale features invariance and use dilated convolution in first layer to capture the global context information. The result of the first network (Grad-CAMs) are used as ROI in the second patches-wise model. Patches-wise training consists of dividing the input images on non-overlapping slides, patches are beneficial to extract fine and local information in large scale images and fast in training due to less used memory and help to reduce the imbalanced data in images. In training stage, slices and their corresponding ground-truth segmentation maps are divided into different patches, by using dividing by 2, 4 and 8 (256x256, 128x128, and 64x64) patches sizes, patches then are fed into the model in multiple levels in the network to learn fine details on object edges, 0 stride is used to preserve same input images size. Networks are trained three times on the three different slices planes to extract features from 3D views, then the task of segmentation consists of combining the results models using a soft voting classifier.

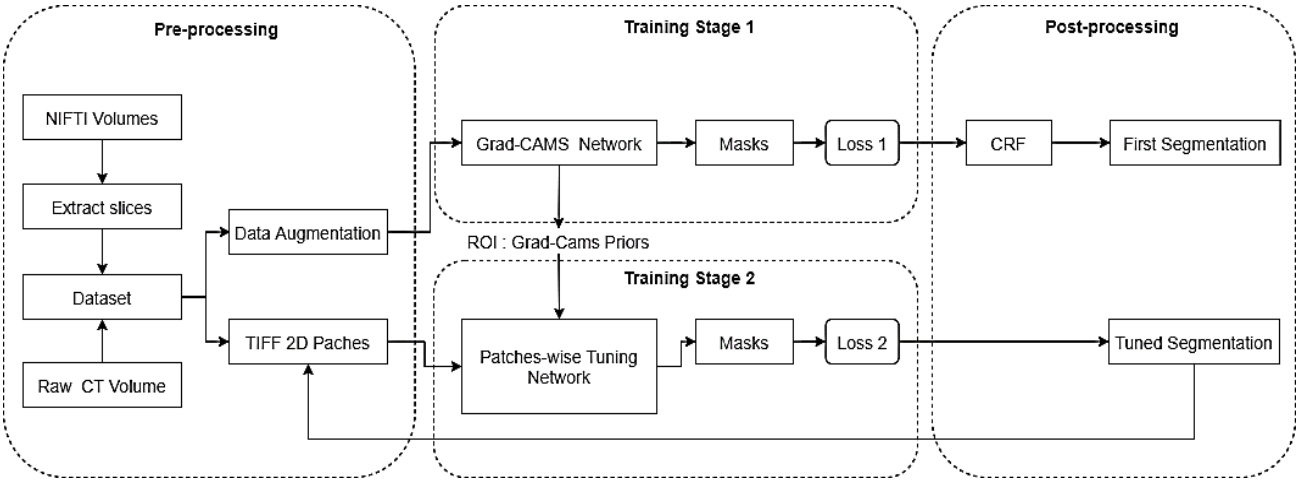


Figure. 3 Framework overview

3.1 Data preparation

The most popular image formats in medical imaging are: DICOM, NIFTI, and less frequently the ANALYZE and NRRD formats. The major difference between the DICOM and NIFTI formats is the information storage. In the DICOM format, the 2D slices made during acquisition are stored in separate files; whereas in the NIFTI format, they are concatenated into a volume that is saved as a single file. In many works or online challenges, a large number of medical images in DICOM format are converted to PNG and JPEG formats, this conversion can be done either to reduce the learning time of the network, or to simplify the data preparation. Unfortunately, converting an image from DICOM format to JPEG or PNG format results in a significant loss of information, invisible to the human eye. This is due to the decrease in the number of gray levels contained in the image: a DICOM image is encoded on 4096 to 65536 gray levels, while a PNG or JPEG image contains only 256.

To preserve all details and prepare our training with a file stream to reduce use of RAM memory, images in created slices are stored in TIFF format.

Next, we divide the dataset into 3 subsets:

Training set: this subset contains most of the initial data (70% in this case). It's used to estimate the parameters of the model during the training phase.

Validation set: this one is used to evaluate the performance of the model and it's fit to the data during the learning phase. It often contains a smaller part of the data (20% in this case).

Test set: it measures the performance of the final model, in particular whether it has a good generalization capacity or, on the contrary, whether it is in an overfitting situation. The test set generally has a small percentage of the initial data (here 10%).

3.2 Preprocessing

Normalizing CT scans is essential to use a common intensity basis. In CT images Hounsfield units (HU), is a measurement of relative densities determined by CT. The HU values fall between -1000 and 1000. We adopted a global windowing preprocessing step to increase contrast or target organs. We set the HU window at the range from -100 to 200 to remove irrelevant organ and tissues. In Fig. 4 we show 3D, coronal, sagittal, and axial plane views. The second rows show the preprocessed volumes with irrelevant organ removed.

After the normalization, all features are in same scale and have equal importance during training. We propose Z-score Eq. (1) :

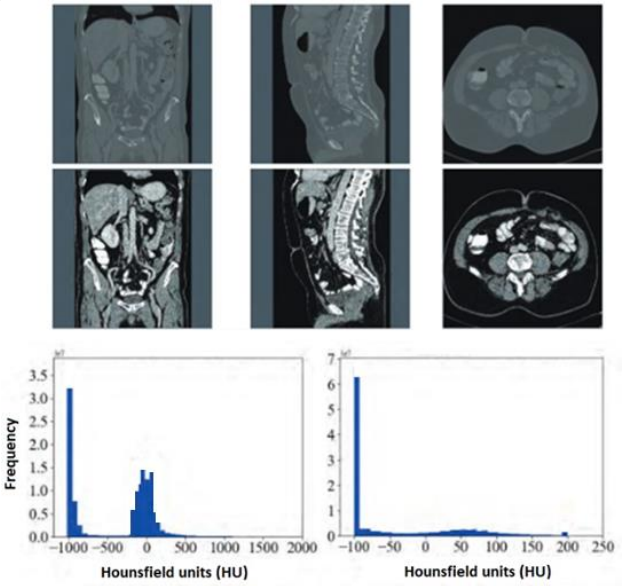


Figure. 4 Image windowing for contrast enhancing

$$Z = \frac{x_i - \mu}{\sigma} \quad (1)$$

where x_i are values, μ is the mean and σ is the standard deviation.

Many artifacts can affect CT images during acquisition process. Noise, beam hardening, motion, scattering, ring, and metal artifacts are the most encountered artifacts. To identify and enhance some of these artifacts, we propose to use the median filter Eq. (2) it is a non-linear filter with good performance in decreasing random, salt-and-pepper, and Gaussian noises. The median filter works well since it keeps the image's edge information while also removing extra noise, such filling up tiny liver holes.

$$I'(u,v) \leftarrow \text{median}\{I(u+i, v+j) \mid (i,j) \in R\} \quad (2)$$

Where I' is the filtered image, I is the image to filter, and R the moving region.

3.3 Localization network

The Fig. 5 describe the first network architecture used to extract Grad-Cams priors (see Fig. 6). The network is an encoder/decoder architecture inspired from U- Net [20] and it consist of many blocks a combination of convolutional, dropout ,max-pooling layers and ReLU activation. Shallow and deep layers are connected through skip connections. At the end

of the network CRF is used as post-processing step to refine the segmentation.

3.4 Tuning network

The localization network provides a good result in almost the whole region of object of interest;

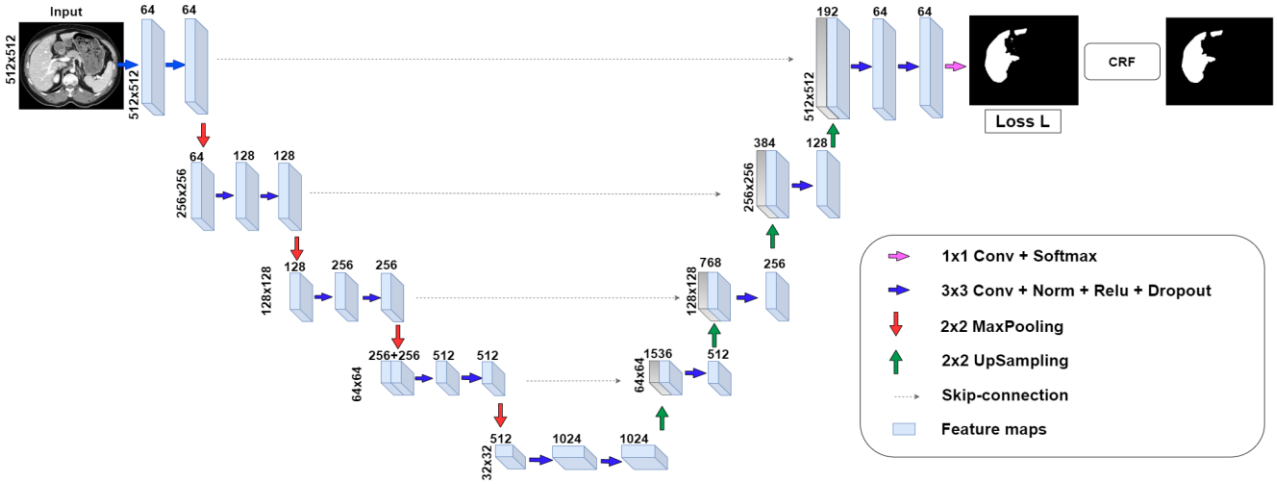


Figure. 5 Localization network

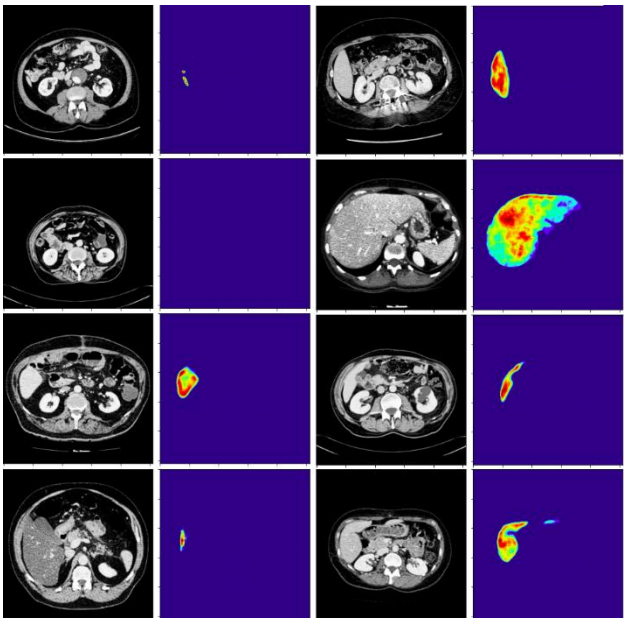


Figure. 6 Samples of grads-cams generated by the first network

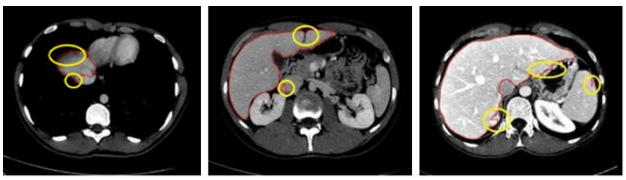


Figure. 7 First segmentation issues

however, it struggles in edges details (see Fig. 7), to resolve this issue we propose to use a cascaded network to refine details in boundaries.

The Fig. 8 describe the tuning network architecture, it's an encoder-decoder model with some concepts inspired from [21]. The model has two inputs layers, the CT images and the Box prior filter result of the first network. The CT image is fed to in the first contracting layer and the patches of this images are fed in next contracting layers. The bounding filter is fed independently to a new block denoted ConvBox. ConvBox is responsible to gather shape and location features. Within each skip connection, the intersection between the unpooled map from a level contracting layer and the location feature map from the ConvBox layer is then obtained. The ConvBox layer indicates the attention area corresponding to the location in the ROI. The output of the model is a segmentation mask derived from learnt relations between the bounding filters as well as the image. The compressed image features are then fed to the bottleneck layer with an ASPP module to make network scale invariant, then the result map is up-sampled and gave to the decoder path on the right side. In the end of each decoder layers, a 1x1 Conv layer is used to flatten the output from each patches layer, the activation function makes the prediction, and the output is then converted to the desired dimension similar to that of the input image. The input of our network combines with the semantic information obtained by down sampling to further link the input in this layer with the advanced feature information closely. Richest and complex features representation appear in the deepest encoding layers. However, with the multiple convolutions and non-linearities, the network tends to lose spatial details in the high-level output maps which make difficult to reduce false detections for small objects with large shape variability. To address this issue, we propose to use multi-level patches to reinforce the signal on fine edges details by concatenating patches features in multiple level in the network to identify relevant

spatial information from low-level feature maps and propagate it to the decoding path. The original image is divided into patches according to the size of the feature map in each encoder layer. We convolve patches with a 1 × 1 convolutional kernel to get the same features dimension as the output of previous layer.

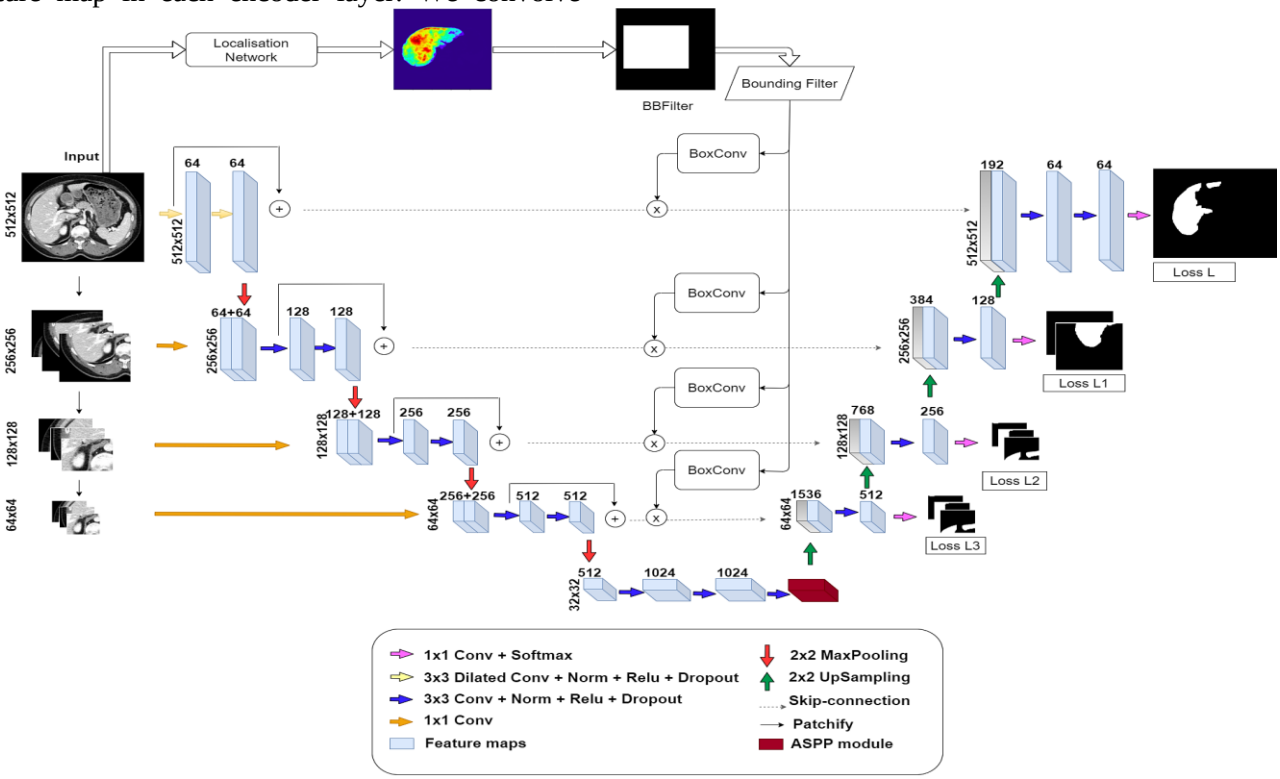


Figure. 8 Tuning network architecture

The summary of the model is:

- Total of params: 38,059,141
- Trainable params: 38,056,069
- Non-trainable params: 3,072

3.5 Integration of ASPP

ASPP is a combination of atrous convolution with different rates and spatial pyramid pooling, the aim of this module is to capture the contextual features at multiple scales by adjusting the receptive field to capture multiscale information. For each pixel i on the output y and filter w, atrous convolution is applied to the input x as shown in Eq. (3):

$$y[i] = \sum_k x[i + r \cdot k]w[k] \quad (3)$$

where r is the atrous rate, it determines the stride of sampling the input image.

We employed the ASPP module that is used in DeepLabv3 [22] to improve the proposed network. Rates of 6, 12, and 18 used. ASPP is applied to the feature map produced by the encoder part Fig. 9, and the resulting feature map is fed into the decoder part, as shown in Fig. 10.

3.6 Loss function

The model outputs the pixel-wise probability map in the top layer. To guarantee the deep layers with

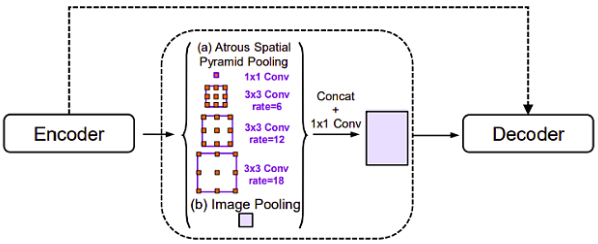


Figure. 9 The ASPP integration

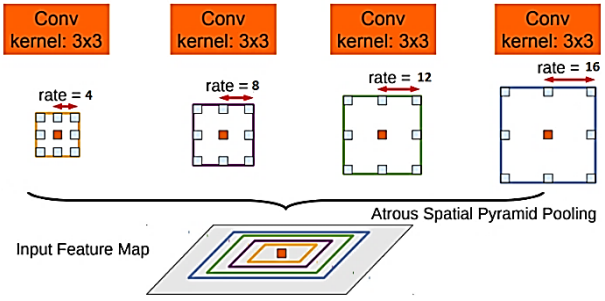


Figure. 10 Multiple scales features using multiple parallel filters with different rates 4,8,12 and 16

correct predictions in deep layers before even reach the top layer, we propose to add labels with corresponding resolution patches to each layer in the decoder path and compare them with the side outputs from deep layers. An effective optimization algorithm is obtained by using calculating loss function in deep levels.

Cross entropy loss Eq. (4) is a widely used pixel-wise loss as it computes on each pixel the entropy value of prediction probability and ground truth, it can well retain boundary information. However, such property might lead to severe sample imbalance since the background has the majority.

$$l_{bce} = -\frac{1}{N}\sum_{n=1}^{N} [y_n \log \hat{y}_n + (1 - y_n) \log(1 - \hat{y}_n)] \quad (4)$$

where $\hat{y}_n$ is the predicted output model and y_n it the ground truth. In cross entropy loss we calculate the average of per-pixel loss discretely, without knowing whether its adjacent pixels are boundaries or not. As a result, cross entropy loss only considers loss in a local sense rather than considering it globally, which is not enough for image level prediction.

The Dice Eq. (5) include also global context and calculates the similarity regions of predicted result and ground truth regardless of the target's relative size.

$$l_{dice} = 1 - \frac{2\hat{y}y+1}{\hat{y}+y+1} \quad (5)$$

Still, training loss would show instability in processing small targets. Therefore, we utilize a combination of Dice loss and binary cross-entropy loss to consider the similarity of local details and global shapes. The higher the value of Dice, the better the segmentation effect is. α and β are used to weight loss in each layer. We believe that dice must be considered in loss function in deep layer to preserve the final segmentation result and avoid unbalanced data in the background. That's why for each layer i the loss is defined as the sum of two losses Eq. (6):

$$l = \alpha\, l_{bce} + \beta\, l_{dice} \quad (6)$$

Where the weights α and β are defined by:

$$\alpha = N - i \text{ and } \beta = i$$

and the top layer loss is defined as: $l = \sum_{i=2}^{N} l_i$ while N is the number of layers in the network.

3.7 Residual connections

The amount of GPU memory needed for training increases with the size of the input image. Additionally, when the architecture uses the complete CT as input, the model has a tendency to overlook information in some portions of the picture, especially when segmenting small regions. To solve

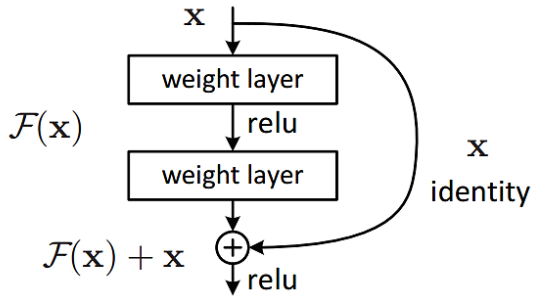


Figure. 11 Residual block

this issue, we suggest a patch-based learning approach, which has the benefit of being more accurate (the network can concentrate on local information at the patch) and requiring less memory for training and inference, hence speeding up computation. Additionally, using 3D patches take advantage of the context in the three directions is a good way to improve the context information provided in the patch (axial, coronal and sagittal).

Furthermore, basic U-Net architecture has only a few layers, to perform better researchers suggest deep structures. Though adding more layers sometimes increases the network performance, it increase remarkably the number of trained parameter and lead to redundant computation due to the rule chain and increase the problem of gradient vanishing during training. Gradient vanishing is the decrease in the learning rate with forward propagation due to the presence of too many hidden layers, which degrades the network's performance.

Residual blocks Fig. 11 shares the same idea of concatenating the input and propagating the low fine details. This enhances the network performance without the need for going deeper. Thus, residual blocks allow to provide deeper networks and reduce the gradient vanishing problem caused by rules chain. Moreover, residual connections make the model learning easier as they learn a function with reference to the input feature map, instead of a referenced function. Therefore, it overcomes the problem of degradation of a deeper network.

3.8 Iterative patches-wise training

Recently, patching learning have been proposed in many works [23-25]. The idea consists of dividing

an image into non-overlapping patches to decrease the need of memory in training step. In our approaches patches are used apply contrastive learning to enforce network to cluster the same instances and push away the distinct instances. This mechanism make network focusing more on local fine-grained features and more fine local features. In our method we utilize an iterative patch learning to manage the large receptive field and make the network mine more local non-discriminative features during optimizing, and thus locate more complete target object regions and fine boundaries. By iterative training on the new self-labelled dataset, we improve the model performance.

The algorithm 1 resume the iterative strategy:

Algorithm 1

Input: X $\{x_1, x_2, ..., x_n\}$: Training dataset
Y $\{y_1, y_2, ..., y_n\}$: Annotations
Output: pixel-wise segmentation maps

Step 1: Choose a threshold area μ.
Step 2: For each level extract patches P $\{p_1, p_{2...}, p_3\}$
Step 3: Train the model on X and Y
Step 4: Obtain segmentation results S {s1,s2..., s3}
Step 5: X = X ⋃ P and $Y = Y \cup S$
Step 6: Fine-tune the network with new images and their patches.
Step 7: Repeat step 3 until the integration model's performance stops improving.

To decrease imbalance due to background. We define an area threshold μ as follows:

$$\mu = \frac{A}{C} \tag{7}$$

where A is the annotated area C is the area of the patch, then patches with an area threshold lower than 0.3 are discarded.

The ConvBox is a succession of max-pooling layer and three convolutional layers to extract the location feature map from the filter input. The output in the correspondent convolved features of each layer for even full images and their patches, a multiply operation is then performed between output and the features from network expanding path as prior integration.

3.9 Post-processing

The output image in the segmentation process is usually not very clear due to the weak features

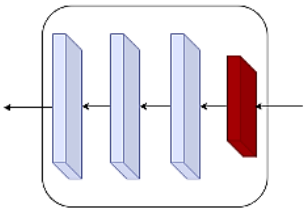
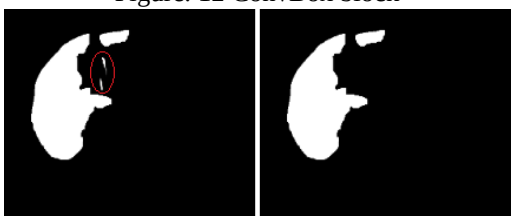


Figure. 12 ConvBox block



(a) (b)

Figure. 13 Removing small artifacts: (a) final probability map and (b) output after removing false positive

Fig. 13. Conditional random field is the most used in postprocessing in image segmentation to improve results and reduce errors, in our framework it is used to optimize the ROI segmentation results in first network. Typically, CRF is a graph-based algorithm where adjacent nodes are coupled to energy terms, to facilitate the assignment of the same labels to the spatial proximal pixels.

3.10 Multi-planes fusion and soft voting inference

In the proposed segmentation method Fig. 14, the models were trained on the three views planes independently. In the inference stage the three masks are combined to compute the final consistence 3D prediction. The motivation of the idea is the 3D context brought by the orthogonal views, each network predicts segmentation in a given slice orientation, and learn a regularity of 2D shapes in slices, but also a certain regularity along the direction orthogonal to the slices. To merge result masks in 3D volume a soft voting is proposed instead of hard voting classifiers (union, intersection, or majority voting) which could lead to over or under-segmentation. In soft voting, we predict the class labels based on the predicted probabilities for classifier and allows one mask to fail without downgrading the final results in the areas where the other two are successful.

Let V ∈ $R^{H\times W\times D}$ be our 3D volume, where H, W, and D are respectively the height, width, and depth of the volume. Furthermore, V(p, r, c) is a single voxel at the location (p, r, c), let P(p, r, c) the predicted value of the voxel V.

$$P(p,r,c) = \frac{1}{3}\sum_{i=1}^{3} P_i(p,r,c) \tag{8}$$

4. Experimental results & discussions

4.1 Datasets

The proposed method was evaluated on two public CT datasets LiTS 2017 and 3D-IRCADb-01.

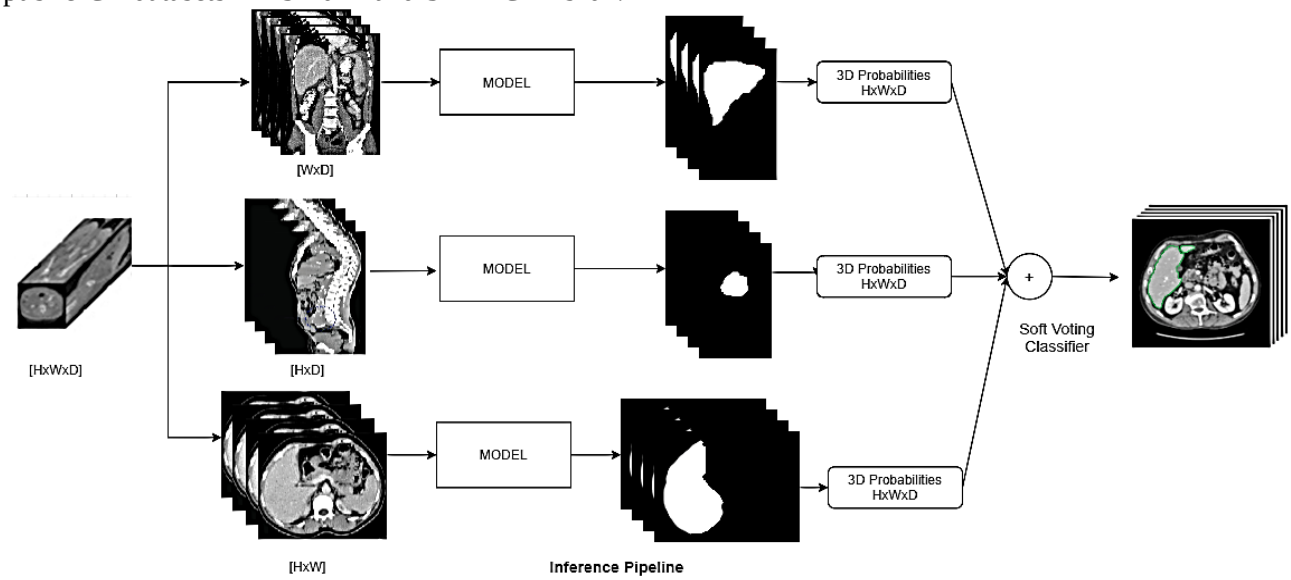


Figure. 14 Models are trained three times with 2D slice. During the inference, the 2D slices are concatenated to obtain 3D volumes and then merged using soft voting

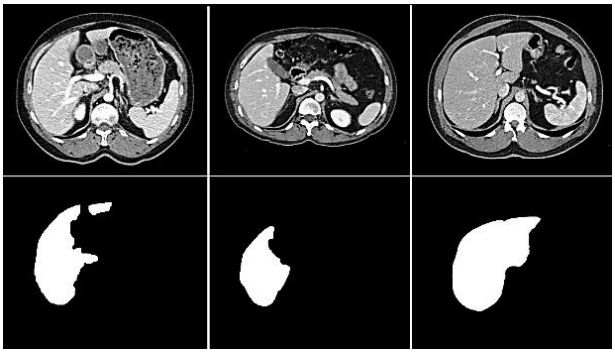


Figure. 15 Example of healthy liver images in datasets in LiTS dataset

of various sizes. The in-plane resolution was always 512x512 pixels, with the pixel spacing, slice thickness, and number of slices varying from 0.56 to 0.87 mm, 1 to 4 mm, and 74 to 260, respectively. Provided tumors are manually segmented by clinical experts and considered as ground truth. The dataset for LiTS was collected from 6 medical centers. The CT scans as well as the segmentations are provided as Nifti .nii files, examples in Fig. 15.

3D-IRCADb-01 dataset contains 3D CT-scans of 20 persons. In 75% of cases hepatic tumours are present. Images are provided in DICOM format in 256x256 size.

The LiTS 2017 dataset includes 3D CT images of individuals who, in 75% of cases, had liver tumours

4.2 Implementation details

The parameters used in our experiments are described in the Table 2. The training done in Google Cloud platform, an instance with a total memory of 128 Go, 16 CPU and 1 Nvidia Tesla 16 GB GPU. Training time was up to 3 hours per model, with an interference time up to 0.01s per slice.

Table 2. Training hyper-parameters

Parameter	Value
Framework	Keras + Tensorflow
Optimizer	Adam
Learning rate	10^{-4}
Dropping rate	0.2
Epochs stopping count	50
Epochs	100
Batch size	4

4.3 Evaluation

We use Accuracy, Dice, and mean Intersection of Union coefficient (IoU) to measure the overlap of the segmentation result and ground truth in order to quantitatively assess the performance of the suggested technique. IoU values vary from 0 to 1, with a value of 0 denoting no overlap and a value of 1 denoting perfectly segmented pixels. Accuracy values range from 0 to 100 and reflect the percentage of correctly predicted pixels.

$$\text{Accuracy} = \frac{TP+\text{TN}}{TP+\text{TN}+\text{FP}+\text{FN}} \quad (9)$$

$$\text{Dice} = \frac{2|X \cap Y|}{|X|+|Y|} \quad (10)$$

$$\text{IoU} = \frac{|X \cap Y|}{|X+Y|} \quad (11)$$

$$\text{mIoU} = \frac{1}{C}\sum_c IoU_c \quad (12)$$

where X is the predicted region and Y the ground truth region, TP denote true positive predictions, TN true negative, FP false positive and FN false negative predictions.

4.4 Experiment results

The statistical results of the proposed method are represented in Table 7 for Liver and in Table 8 for Lesions. Our method achieved a DICE of 0.9511, 0.9501 and 0.9465 respectively in Axial, Coronal and Sagittal plane. And an average IoU of 0.9070, 0.8976 and 0.8988 respectively in Axial, Coronal and Sagittal plane. The Table 3 shows the impact of the proposed patches level method on the final segmentation performance we evaluate integrating patches in each level to measure the impact of this proposed strategy on the final performance, and the Table 4 show the effects of the proposed custom multiple level loss function, the model was trained with one loss function at the end of the model then with the proposed weighted custom loss function , then the Table 5 and Fig. 16 show the impacts of the reiterative learning proposed in the tuning network as described in the framework algorithm, the model is trained iteratively on the self-augmented dataset, iteration stop once we reach the best metrics.

Table 3. Impact of patch level on segmentation performance (Axial plan)

Parameter	**Patch Level**		
	Level-1 256	**Level-2 128**	**Level-3 64**
Average Accuracy	0,9512	0,952	0,9732
Dice	0,9301	0,9102	0,9511
mIoU	0,8810	0,870	0,907

Table 4. Effects of multi-level custom loss function (Axial plan)

Metrics	**Custom Loss Function**	
	Without	**With**
Average Accuracy	0,9020	0,9732
Dice	0,9430	0,9511
mIoU	0,8913	0,907

Table 5. Effects of iterations in the reiterative learning (Axial plan)

Iter	**Metrics**					
	Accuracy		**Dice**		**mIoU**	
	Val	**Test**	**Val**	**Test**	**Val**	**Test**
N = 0	0,9520	0,9410	0,8402	0.8510	0,8210	0,8530
N = 1	0,9572	0,9481	0,8682	0.8964	0,8470	0,8723
N = 2	0,9600	0,9575	0,8953	0.9015	0,8670	0,8870
N = 3	0,9690	0,9591	0,9120	0.9265	0,8810	0,8920
N = 4	),9701	0,9610	0,9420	0.9310	0,8750	0,8910
N = 5	0,9732	0,9621	0,9511	0.9331	0,9700	0,9070
N = 6	0,9730	0,9732	0,9641	0.9244	0,9640	0,9010
N = 7	0,9710	0,9730	0,9672	0.9202	0,9680	0,9020

The framework has been validated on two different datasets, with different input images sizes 512x512 and 256x256, the Table 6 present the obtained results:

Table 6. Performance comparison on two datasets

Category	**Dataset**	**Acc**	**Dice**	**mIoU**
Liver	LiTS	0,9732	0,9511	0,9070
	3D IRCADB	0,9701	0,9330	0,9210
Lesion	LiTS	0,9471	0,9313	0,8718
	3D IRCADB	0,9524	0,9417	0,9249

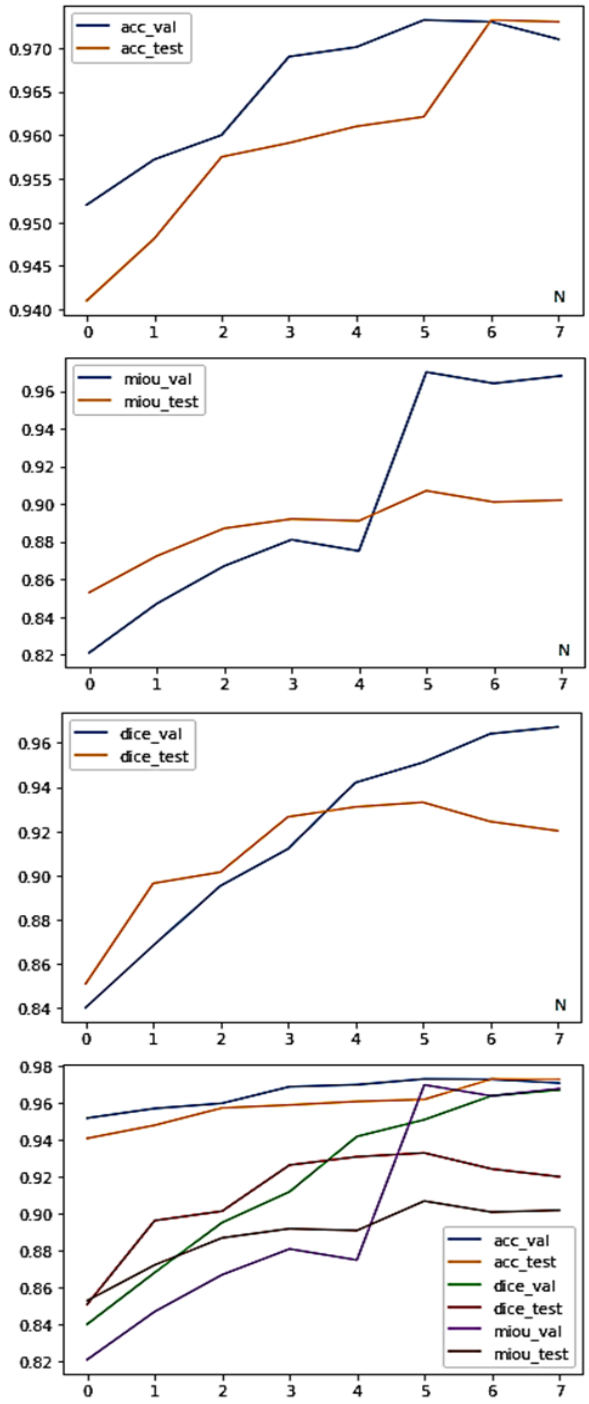
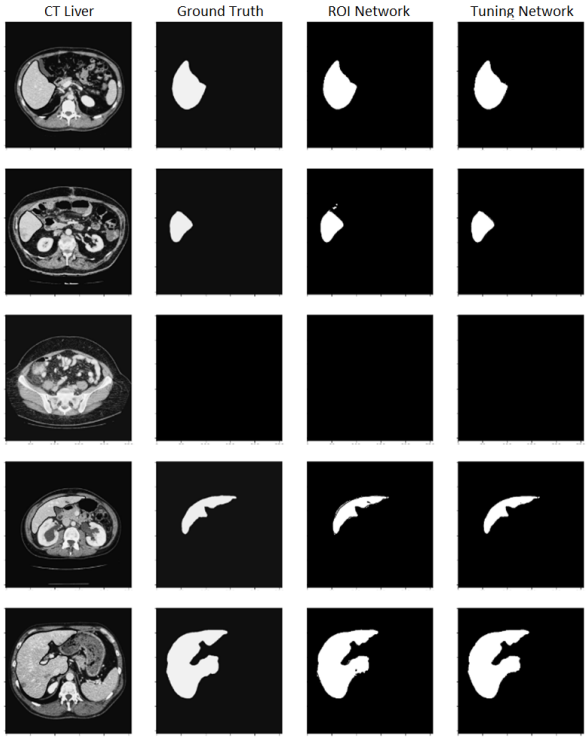
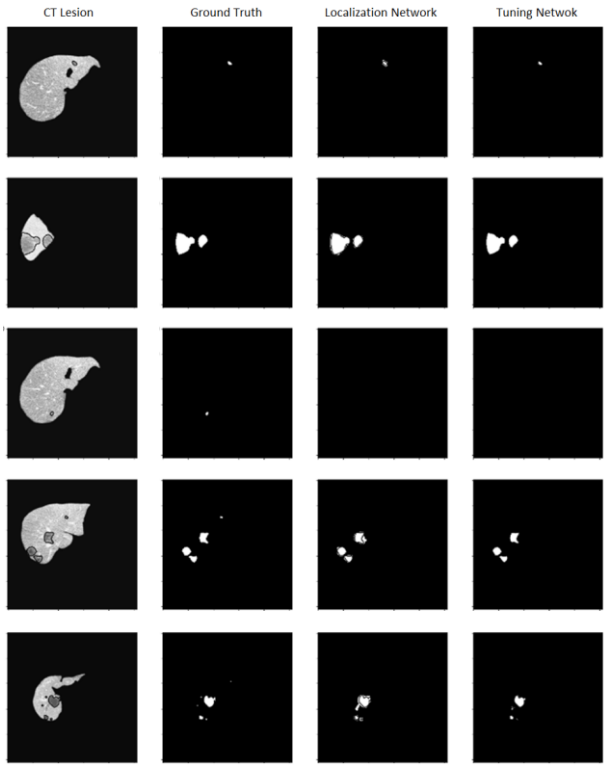


Figure. 16 Metrics evolution per number of iterations



(a) (b) (c) (d)

Figure. 17 Framework results in liver segmentation: (a) the CT images, (b) the ground truth, (c) the ROI extracted with the first network, and (d) the tuned segmentation



(a) (b) (c) (d)

Figure. 18 Framework results in liver lesions segmentation: (a) the CT images, (b) the ground truth, (c) the ROI extracted with the first network, and (d) the tuned segmentation

5. Conclusions

In this paper, we developed a novel framework for weakly semantic segmentation. The framework contains two cascaded and adapted Encoder-Decoder networks. The first one extract the Grad-Cams to localize the region of interest that we feed into a second tuning network using a customized loss function and an iterative and multi-level patches wise learning. The model reaches a dice coefficient of 0.9199 and mIoU of 0.965 on test data. Although iterative learning improves the performance indices of our model, after a certain number of iterations, an overfitting occurs when the accuracy and dice on the validation continues increasing but decrease on test data. The results show that our model performs well on small objects comparing to other models due to our proposed tuning network with the fed patches in deep levels. As post-processing method, we suggest use CRF just in the localization network. CRF shown a good result to remove small artifacts in the ROI. However, in the patches segmentation, and due to errors caused by the weak annotation the data distribution learned by the network is not completely correct, and CRF didn't bring any remarkable improvement.

However, the framework performance relatively depends on some manually defined hyperparameters and taking time to adjust. The contours of the segmentations approximate the specialist's markings, being slightly larger or smaller. We notice also the presence of some wrong classified pixels in the holes lesions that can be improved by using texture features. As future perspective, the framework can be improved also to classify the type of detected lesions.

Conflicts of Interest

The authors declare no conflict of interest

Author Contributions

Conceptualization, software, formal analysis, investigation, writing—original draft preparation, writing—review and editing have been done by 1st author (Youssef OUASSIT). Methodology, validation, supervision have been done by 2nd, 3rd and 4th author (Soufiane ARDCHIR, Mohamed Yassine EL GHOUMARI & Mohamed AZOUAZI).

References

[1] P. Badura and W. Wieclawek, "Calibrating level set approach by granular computing in computed tomography abdominal organs segmentation", *Applied Soft Computing*, Vol. 49, p. 887-900, 2016.

Table 7. Evaluation results of our proposed framework for liver segmentation

Metrics/Models	**SegNet** [18]		**U-Net** [20]		**DeeplabV3** [22]		**Ours**	
	Val	**Test**	**Val**	**Test**	**Val**	**Test**	**Val**	**Test**
	Axial Plane							
Accuracy	0,9578	0,9543	0,9661	0,9525	0,9375	0,9564	0,9732	0,9621
Dice	0,7738	0,7667	0,8293	0,8251	0,8891	0,8807	0,9511	0,9465
mIoU	0,8174	0,8602	0,8566	0,7097	0,8071	0,7964	0,9070	0,8988
	Coronal Plane							
Accuracy	0,9210	0,9213	0,9565	0,9113	0,9432	0,9724	0,973	0,9542
Dice	0,7716	0,7594	0,8285	0,8241	0,8838	0,8798	0,9501	0,9425
mIoU	0,8627	0,8567	0,8434	0,7212	0,799	0,873	0,8976	0,8924
	Sagittal Plane							
Accuracy	0,9340	0,9290	0,9542	0,9662	0,9135	0,9834	0,9631	0,9732
Dice	0,7690	0,7480	0,8280	0,8130	0,8838	0,8793	0,9465	0,9330
mIoU	0,6660	0,6420	0,8411	0,8477	0,7990	0,8956	0,8988	0,8071

Table 8. Evaluation results of our proposed framework for lesions segmentation

Metrics/Models	**SegNet** [18]		**U-Net** [20]		**DeeplabV3** [22]		**Ours**	
	Val	**Test**	**Val**	**Test**	**Val**	**Test**	**Val**	**Test**
	Axial Plane							
Accuracy	0,9631	0,9493	0,9564	0,9410	0,9711	0,9651	0,9471	0,9736
Dice	0,7363	0,5899	0,8125	0,7900	0,8659	0,8615	0,9313	0,9199
mIoU	0,8205	0,7156	0,8467	0,6940	0,8829	0,7743	0,8718	0,9165
	Coronal Plane							
Accuracy	0,7440	0,9444	0,9625	0,9410	0,9233	0,9442	0,9331	0,9725
Dice	0,7280	0,5477	0,8031	0,7850	0,8627	0,8614	0,9225	0,9075
mIoU	0,6220	0,7000	0,8329	0,6880	0,7716	0,7666	0,8562	0,9054
	Sagittal Plane							
Accuracy	0,9582	0,9359	0,9637	0,9431	0,9732	0,9680	0,9314	0,9431

Dice	0,7068	0,4539	0,7952	0,7846	0,8615	0,8334	0,9225	0,9056
mIoU	0,7977	0,6409	0,8330	0,7031	0,7743	0,8620	0,8572	0,9024

[2] T. Heimann, I. Wolf, and H. P. Meinzer, "Active Shape Models for a Fully Automated 3D Segmentation of the Liver – An Evaluation on Clinical Data", In: *Proc. of Medical Image Computing and Computer-Assisted Intervention – MICCAI 2006*, Vol. 4191, pp. 41-48, 2006.

[3] A. Baâzaoui, W. Barhoumi, A. Ahmed, and E. Zagrouba, "Semi-Automated Segmentation of Single and Multiple Tumors in Liver CT Images Using Entropy-Based Fuzzy Region Growing", *IRBM*, Vol. 38, No 2, pp. 98-108, 2017.

[4] J. Han, C. Yang, X. Zhou, and W. Gui, "A new multi-threshold image segmentation approach using state transition algorithm", *Applied Mathematical Modelling*, Vol. 44, pp. 588-601, 2017.

[5] Q. Huang, H. Ding, X. Wang, and G. Wang, "Fully automatic liver segmentation in CT images using modified graph cuts and feature detection", *Computers in Biology and Medicine*, Vol. 95, pp. 198-208, 2018.

[6] Q. Zheng, H. Li, B. Fan, S. Wu, and J. Xu, "Integrating support vector machine and graph cuts for medical image segmentation", *Journal of Visual Communication and Image Representation*, Vol. 55, pp. 157-165, 2018.

[7] W. Wu, S. Wu, Z. Zhou, R. Zhang, and Y. Zhang, "3D Liver Tumor Segmentation in CT Images Using Improved Fuzzy C -Means and Graph Cuts", *BioMed Research International*, Vol. 2017, pp. 1-11, 2017.

[8] M. Barstugan, R. Ceylan, M. Sivri, and H. Erdogan, "Automatic Liver Segmentation in Abdomen CT Images using SLIC and AdaBoost Algorithms", In: *Proc. of the 2018 8th International Conference on Bioscience, Biochemistry and Bioinformatics - ICBBB 2018*, pp. 129-133, 2018.

[9] S. Zheng, B. Fang, L. Li, M. Gao, Y. Wang, and K. Peng, "Automatic liver tumour segmentation in CT combining FCN and NMF-based deformable model", *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization*, Vol. 8, No 5, p. 468-477, 2020.

[10] 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.

[11] J. D. L. Araúj, "An automatic method for segmentation of liver lesions in computed tomography images using deep neural networks", *Expert Systems with Applications*, Vol. 180, p. 115064, 2021.

[12] M. Bellver, K. K. Maninis, J. P. Tuset, X. G. Nieto, J. Torres, and L. V. Gool, "Detection-aided liver lesion segmentation using deep learning", *arXiv*, 2017.

[13] E. Vorontsov, A. Tang, C. Pal, and S. Kadoury, "Liver lesion segmentation informed by joint liver segmentation", *arXiv*, 2018.

[14] N. Nanda, P. Kakkar, and S. Nagpal, "Computer-Aided Segmentation of Liver Lesions in CT Scans Using Cascaded Convolutional Neural Networks and Genetically Optimised Classifier", *Arab J Sci Eng*, Vol. 44, No. 4, pp. 4049-4062, 2019.

[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, p. 82122-82133, 2019,

[16] 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.

[17] Samara National Research University, Samara, Russia et al., "Liver tumor segmentation CT data based on Alexnet-like convolution neural nets", In: *Proc. of International Conference Information Technology and Nanotechnology (ITNT-2016)*, pp. 348-356, 2016.

[18] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation", *arXiv*, 2015,

[19] P. Hu, F. Wu, J. Peng, P. Liang, and D. Kong, "Automatic 3D liver segmentation based on deep learning and globally optimized surface evolution", *Phys. Med. Biol.*, Vol. 61, No. 24, p. 8676-8698, 2016.

[20] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation", *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 234-241, 2015.

[21] E. J. Rosana, C. Petitjean, P. Honeine, and F. Abdallah, "BB-UNet: U-Net With Bounding

Box Prior", *IEEE J. Sel. Top. Signal Process*., Vol. 14, No 6, p. 1189-1198, 2020.
[22] L. Xu, H. Xue, M. Bennamoun, F. Boussaid, and F. Sohel, "Atrous convolutional feature network for weakly supervised semantic segmentation", *Neurocomputing*, Vol. 421, pp. 115-126, 2021.
[23] J. Li, Z. Jie, X. Wang, Y. Zhou, X. Wei, and L. Ma, "Weakly Supervised Semantic Segmentation via Progressive Patch Learning", *IEEE Trans. Multimedia*, p. 1, 2022.
[24] L. Wang, "Automated bone segmentation from dental CBCT images using patch-based sparse representation and convex optimization: Segmentation of CBCT image", *Medical Physics*, Vol. 41, No 4, p. 043503, 2014.
[25] N. Yamanakkanavar and B. Lee, "Using a Patch-Wise M-Net Convolutional Neural Network for Tissue Segmentation in Brain MRI Images", *IEEE Access*, Vol. 8, pp. 120946-120958, 2020.