

ETH

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich



UNIVERSITAT POLITÈCNICA
DE CATALUNYA
BARCELONATECH



Barcelona
Supercomputing
Center
Centro Nacional de Supercomputación

Detection-aided medical image segmentation using deep learning

A Master's Thesis Submitted to the Faculty of the Escola Tècnica d'Enginyeria de
Telecomunicació de Barcelona Universitat Politècnica de Catalunya by

Míriam Bellver

In partial fulfillment of the requirements for the degree of
MASTER IN TELECOMMUNICATIONS ENGINEERING

Advisors in ETH Zürich: Kevis-Kokitsi Maninis, Dr. Jordi Pont-Tuset
Prof. Dr. Luc van Gool

Advisors in UPC: Prof. Dr. Xavier Giró-i-Nieto, Prof. Dr. Jordi Torres

August 14, 2017

Abstract

A fully automatic technique for segmenting the liver and localizing its unhealthy tissues is a convenient tool in order to diagnose hepatic diseases and also to assess the response to the according treatments. In this thesis we propose a method to segment the liver and its lesions from Computed Tomography (CT) scans, as well as other anatomical structures and organs of the human body. We have used Convolutional Neural Networks (CNNs), that have proven good results in a variety of tasks, including medical imaging. The network to segment the lesions consists of a cascaded architecture, which first focuses on the liver region in order to segment the lesion. Moreover, we train a detector to localize the lesions and just keep those pixels from the output of the segmentation network where a lesion is detected. The segmentation architecture is based on DRU [24], a Fully Convolutional Network (FCN) with side outputs that work at feature maps of different resolutions, to finally benefit from the multi-scale information learned by different stages of the network. Our pipeline is 2.5D, as the input of the network is a stack of consecutive slices of the CT scans. We also study different methods to benefit from the liver segmentation in order to delineate the lesion. The main focus of this work is to use the detector to localize the lesions, as we demonstrate that it helps to remove false positives triggered by the segmentation network. The benefits of using a detector on top of the segmentation is that the detector acquires a more global insight of the healthiness of a liver tissue compared to the segmentation network, whose final output is pixel-wise and is not forced to take a global decision over a whole liver patch. We show experiments with the LiTS dataset for the lesion and liver segmentation. In order to prove the generality of the segmentation network, we also segment several anatomical structures from the Visceral dataset.

Acknowledgements

First of all I wanted to thank Prof. Dr. Luc Van Gool for accepting me to join the Computer Vision Lab in ETH Zürich for this semester as an exchange student, I enjoyed and learned a lot during these six months. I want to specially thank to Jordi P. and Kevis all the advice during the whole project. It has been a pleasure having the chance to work with both of you, I do feel I learned a lot from our conversations during these months. Also, next time I will beat you on kicker!

Thanks also to Xavi and Jordi T., for always being so supportive and motivating since the minute I met both of you. Also my research brother Victor C. and the other part of the team in Barcelona, I am sure we will enjoy the next stage working together, as it has been until this date.

Finally I want to thank my family, who always supported me in an unconditional way, and my partner Victor, that although we were at different continents these past few months, has been my greatest support in all moments, always so caring and attentive. Last but not least, to all the people I have met during these months and have become my little international family, you made my stage in Zürich unforgettable!

Contents

1	Introduction	1
1.1	Motivation and Contributions	1
1.2	Work plan	2
1.2.1	Work Packages	2
1.2.2	Milestones	2
1.2.3	Gantt diagram	3
1.2.4	Deviations from the initial plan	3
2	Related Work and State of the Art	5
3	Detection-aided medical image segmentation using deep learning	7
3.1	Baseline Architecture	7
3.1.1	Cascaded architecture for liver and lesion segmentation	7
3.2	Loss objective	8
3.2.1	Balancing the loss	9
3.3	Using context slices to exploit 3D information	10
3.4	Benefiting from liver segmentation in order to segment the lesion	10
3.4.1	No back-propagation through regions outside liver	11
3.4.2	Multi-task: Segmenting the liver and lesion simultaneously	11
3.4.3	Multi-task with no back-propagation through liver	11
3.5	Lesion Detection	11
3.5.1	Image labeling of images with and without lesion	12
3.5.2	Detection of lesion in a sliding-window fashion	13
3.6	3D - Conditional Random Fields (3D-CRF)	13
4	Experimental Validation	15
4.1	Datasets	15
4.1.1	LiTS	15
4.1.2	Visceral	15
4.2	Metrics	16
4.3	Experiments on LiTS dataset	17
4.3.1	Data Preparation	17
4.3.2	Balancing the Binary Cross Entropy Loss	17
4.3.3	Using context slices to exploit 3D information	18
4.3.4	Using the liver for segmenting the lesion	20

CONTENTS

4.3.5	Training a Lesion Detector	20
4.3.6	Removing False Positives using the Lesion Detector	21
4.3.7	3D Conditional Random Fields	24
4.3.8	LiTS competition	26
4.4	Experiments on the Visceral dataset	27
5	Budget	29
6	Discussion and Conclusions	31
A	Additional material	33
A.1	From predicted liver to bounding box	33
A.2	Visualizations	33
A.2.1	Visualizations of how benefiting from liver helps to segment the lesion	33
A.2.2	Using the detector to remove false positives	33
A.2.3	Visualizations of the final configuration	34

List of Figures

1.1	Gantt diagram	4
3.1	Architecture	8
3.2	Computing 3D bounding box	8
3.3	Stack of slices as input	10
3.4	Scheme of just back propagating trough liver.	11
3.5	Scheme of segmenting the liver and lesion simultaneously	12
3.6	Mixed Scheme	12
3.7	Bounding boxes sampling	13
4.1	Pixel intensity histogram of images from LiTS dataset.	16
4.2	The pixel intensity histogram for the liver.	16
4.3	The pixel intensity histogram for the lesion.	16
4.4	The pixel intensity histogram of images from Visceral dataset.	17
4.5	The pixel intensity histogram of the organs analyzed from the Visceral dataset.	17
4.6	Precision- Recall Curves for different balancing methods for the lesion.	18
4.7	Precision- Recall Curves when the input is a different number of slices for the liver.	19
4.8	Precision- Recall Curves when the input is a different number of slices for the lesion.	19
4.9	Visualizations of strategies that use the liver	21
4.10	Precision-Recall curve for the classification.	22
4.11	Precision-Recall curve for the detection.	22
4.12	Visualizations of detections	23
4.13	Precision Recall Curve after removing negative detections.	24
4.14	Visualizations after applying 3D-CRF	26
4.15	Visceral visualizations	28
A.1	Gaussian fitting of predicted masks for volume 125.	34
A.2	Analysis of selection of sigma.	34
A.3	Zoomed visualizations of strategies that use the liver	35
A.4	Visualizations before applying detector	36
A.5	Visualizations after applying detector	37
A.6	Examples of last configuration.	38

LIST OF FIGURES

List of Tables

1.1	Milestones.	3
3.1	Samples after data augmentation for detection	13
4.1	Average Dice across volumes for the two processing techniques.	17
4.2	Dice for Lesion and Liver using losses.	18
4.3	Comparison of inputting several consecutive slices for the lesion.	19
4.4	Results for the liver segmentation.	19
4.5	Different methods to benefit from liver segmentation.	20
4.6	Performance of the Image Classifier on	22
4.7	Image Window detections performance	22
4.8	Results when filtering the segmentation mask with a lesion detector	24
4.9	Results after applying 3D-CRF	25
4.10	Results when added 3D-CRF to liver	25
4.11	Results obtained at MICCAI LiTS Challenge	25
4.12	Legend of the architecture for each submission of MICCAI LiTS Challenge.	25
4.13	Legend of the different metrics of LiTS Challenge.	27
4.14	Visceral results	27
5.1	Total personal costs	29
5.2	Software Licenses cost	29

Chapter 1

Introduction

1.1 Motivation and Contributions

Segmenting the liver and its lesions on medical images helps oncologists to accurately diagnose liver cancer, as well as to assess the treatment response of patients. Typically, doctors rely on manual segmentation techniques in order to interpret the Contrast Tomography (CT) and Magnetic Resonance Imaging (MRI) images. Automatic tools that is not as subjective and time-consuming has been widely studied in the recent years. Liver lesion segmentation is a challenging task due to the low contrast between liver, lesions, and also nearby organs. Other additional difficulties are the lesion size variability and the noise in CT scans. Building a robust system that is able to beat these difficulties is still an open problem. Recently, methods based on deep Convolutional Neural Networks (CNNs) have demonstrated to be robust to these challenges, and have achieved the state of the art at this task.

In this thesis we study the performance of recent deep learning methods in biomedical image segmentation. Specifically, we adapt DRIU [24] for the challenging task of segmenting both the liver and lesion from CT scans. DRIU is a Fully Convolutional Network (FCN) that has side outputs with supervision at different convolutional stages. The different multi-scale side outputs are combined to generate the final output of the network. This architecture has proven to be successful for the medical task of segmenting the blood vessels and optical disk of eye fundus images, as well as for one-shot video object segmentation (OSVOS)[2] in videos of natural images. We will focus on the liver and lesion segmentation in the framework of the Liver Tumor Segmentation (LiTS) Challenge. The LiTS Challenge provides a dataset with liver and lesion segmentations, but only assesses the lesion segmentation task. This challenge was first opened for a workshop of the International Symposium on Biomedical Imaging (ISBI) 2017 Conference that was held in April 2017. At the beginning of July the challenge was opened again, this time for a workshop of the Medical Image Computing and Computer Assisted Interventions Conference (MICCAI) 2017 Conference. We will also show results for the Visceral dataset to segment more anatomical structures in addition to the liver, as well as multiple organs in one pass of the CNN.

The key aspects of the method are the following:

- We use a cascaded architecture. A first network is trained to segment the liver in order to focus on the liver region at the time of segmenting the lesion.
- In order to deal with the classic problem of imbalanced data for medical applications, we test several methods to balance the Binary Cross Entropy (BCE) loss used to train the network.

- The input of the architecture is a stack of 3 consecutive slices of the CT volumes, which improves accuracy compared to the baseline of considering a single slice.
- We study how to benefit from the liver segmentation in order to segment the lesion, resulting the best method the one that does not back-propagate gradients through regions outside the liver at the time of segmenting the lesion.
- A detector is added on top of the segmentation network in order to remove false positives triggered by the segmentation network. We prove that the combination of using a detector and a segmentation network improves the results over just using the segmentation network.
- As post-processing step, a 3D Conditional Random Field (3D-CRF) is applied to gain spatial coherence for the liver and lesion predictions.
- The generality of the segmentation network is demonstrated by retraining it for several anatomical structures from the Visceral Dataset.

The framework used for this project has been Tensorflow[25], starting from the open-source code of One-shot Video Object Segmentation (OSVOS)[2].

1.2 Work plan

This project is a joint collaboration between the Computer Vision Lab of Eidgenössische Technische Hochschule Zürich (ETH Zürich), the Image Processing Group of Universitat Politècnica de Catalunya (UPC) and the Barcelona Supercomputing Center (BSC). We had regular meetings every week with the team from ETH Zürich, and weekly video calls with the Barcelona team.

1.2.1 Work Packages

The different work packages for the project are defined as follows:

- WP 1: Definition of project
- WP 2: Research about state of the art
- WP 3: Datasets
- WP 4: Adaption of software
- WP 5: Experimentation on LiTS dataset
- WP 6: Experimentation on Visceral dataset
- WP 7: Participation in LiTS Challenge
- WP 8: Final Documentation

1.2.2 Milestones

The milestones of the project are listed in Table 1.1.

WP	Milestone	Date
1	Definition of project	15/02/2017
3	Dataset ready to be used	3/03/2017
4	Run the original code	15/03/2017
5	Do a baseline for our task with the original code	20/03/2017
5	Have a working cascaded architecture for lesion and liver segmentation	25/03/2017
5	Define final loss objective	7/04/2017
5	Use volume of CT scan	18/04/2017
5	Define a strategy that uses liver for lesion segmentation	10/05/2017
5	Have a working classifier for healthy/unhealthy tissues	25/05/2017
5	Have a working detector for healthy/unhealthy tissues	15/06/2017
6	Demonstrate generality of segmentation network with Visceral dataset	5/07/2017
7	Submit to LiTS Challenge	19/07/2017
8	Deliver report to ETH	31/07/2017
8	Oral defense in ETH	31/07/2017
8	Deliver report to UPC	20/08/2017
8	Oral defense in UPC	8/09/2017

Table 1.1: Milestones.

1.2.3 Gantt diagram

The Gantt diagram with the work packages and the different tasks can be seen in Figure 1.1.

1.2.4 Deviations from the initial plan

Throughout the project there had been small deviations from the initial plan, mainly of tasks that took longer than expected. For instance there were some problems training the networks with the Dice loss, as well as some problems when training the classifier for healthy/unhealthy tissues. Another deviation from the initial plan was the date that the Challenge was supposed to end (21th July), as it was extended due to technical problems of the platform. As a consequence, we did not have the last results until the end of July.

CHAPTER 1. INTRODUCTION

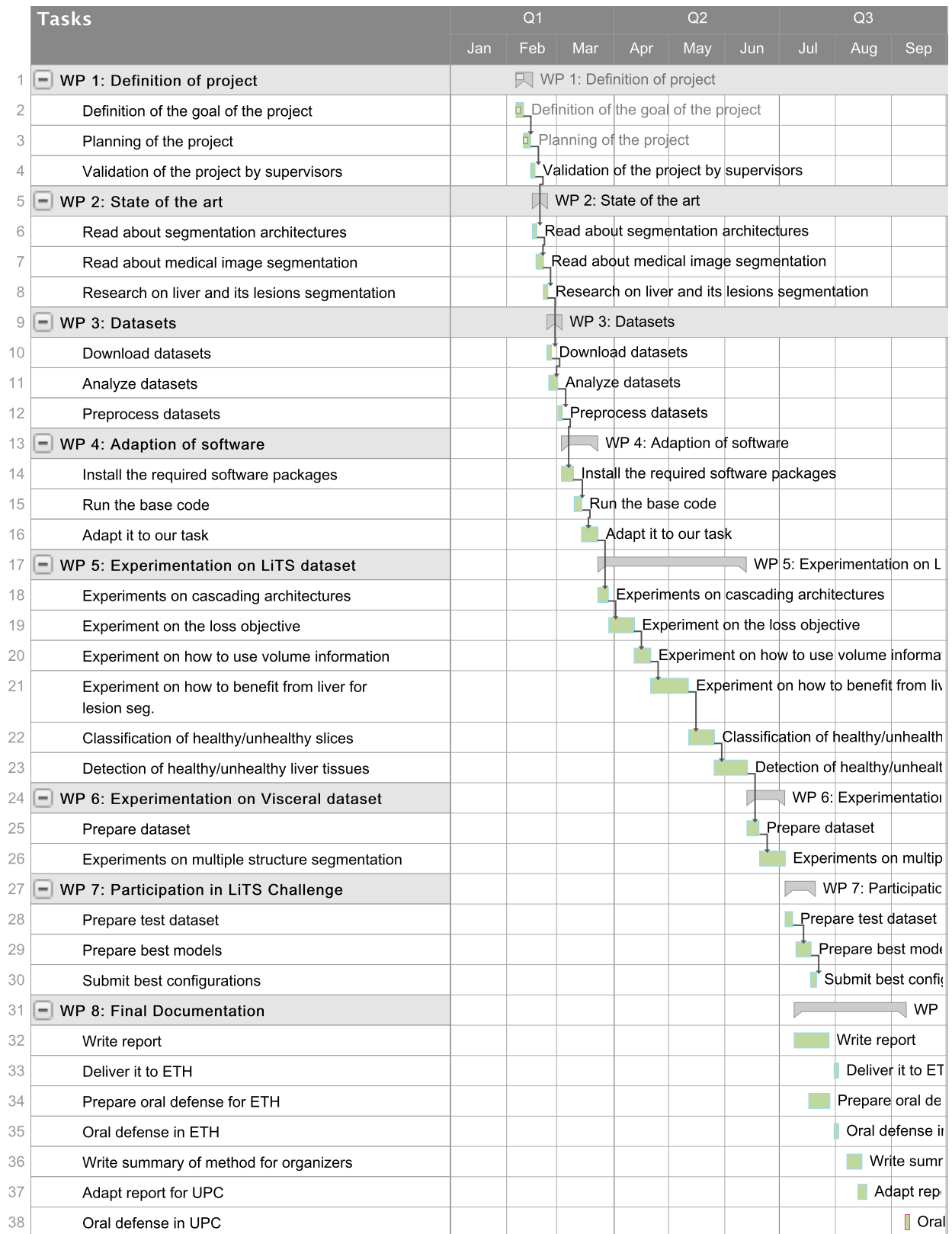


Figure 1.1: Gantt diagram

Chapter 2

Related Work and State of the Art

In the recent years, deep Convolutional Neural Networks (CNNs) have significantly improved the performance of computer vision systems for the tasks of image classification and object detection. Since AlexNet [17], deeper and more complex networks have been proposed, with VGGNet [28] and ResNet [12] being nowadays the reference base networks for multiple computer vision applications. VGGNet is composed by a set of convolutional stages with pooling layers, which decrease the feature maps resolution and increment the receptive field of the neurons. In this contracting path, deeper layers of the network learn more abstract concepts compared to the local features learned at the first convolutional stages. In comparison to VGGNet, ResNet adds shortcuts at each layer, which allows a better flow of the gradient through the network, solving classical problems such as gradient vanishing for deep neural networks.

Starting from Image Classification, advances in deep learning have progressively been transferred to solve other computer vision tasks, as boundary detection and segmentation [29] [23]. State of the art architectures for dense predictions are based on Fully Convolutional Networks (FCN) [21]. Classic CNN architectures used for Image Recognition consist of a contracting path composed by convolutional, pooling and fully connected layers. FCNs just rely on convolutional and pooling layers, which allows to generate segmentation maps for any input size with a single forward pass. In order to output a map with the initial resolution of the image, two alternatives have been studied in the literature. The first one consists of adding an expanding path based on deconvolutional or unpooling layers, which recovers spatial information by merging features from different resolutions of the contracting path. Hence low-level details of major importance in dense prediction can be preserved. DRIU [24], which is the baseline of this work, uses the first alternative, and was designed to segment the optical disc and blood vessels of eye fundus images to assess diagnosis of eye diseases. The second alternative consists of architectures that use dilated/atrous convolutions [30] in order to increment the field of view, preserving a higher resolution of feature maps throughout the network.

Segmentation of medical images is typically exploited to delineate different anatomical structures and also for detecting unhealthy tissues, such as liver lesions. The recent successes of semantic segmentation have also improved the accuracy in the medical field. Nevertheless, there are some characteristics of medical images that are challenging for training CNNs and require of some modifications to the classical pipelines, such as the imbalance of the labels for the data. There are various technologies that produce medical images; computed tomography (CT), ultrasound, X-ray and magnetic resonance Imaging (MRI). These medical images form 3D volumes, that can be processed using several methods. The most straight-forward is using 2D FCNs, such as DRIU [24], U-net [27] or DCAN [3], by considering each slice of the volume independent of each other. Another approach consists on exploiting the 3D data by implementing a 3D-convolutional

network approach, or an hybrid one (2D and 3D convolutions) [18], [19]. Another method is to combine several tri-planar schemes, as [10]. There are also methods that capture the volume information using Recurrent Neural Networks (RNNs) [4].

More concretely about liver and lesion segmentation, in the last years most of the methods were based on statistical shape models, together with intensity distributions models [13] [14]. In fact some of the entries of the LiTS challenge are based on statistical methods, such as [20], that proposes an unsupervised method using phase separation between healthy liver and lesions. There is another group of approaches that rely on classifiers and low-level segmentation, such as [15], that first detects the organ and the lesion is segmented during a second stage using active contours.

Deep CNNs for these tasks have recently been proved successful, as they are robust to varying images, which allows to build fully automatic liver and lesion segmentation systems. For liver segmentation, [8] trains 3D FCNs with Conditional Random Fields as post-processing. In the same direction, [22] use 3D CNNs and Graph Cut to refine segmentations of the liver. Regarding liver and lesion segmentation, the authors of [6] train two FCNs, one to segment the liver, and then another one that works with the mask of the liver in order to segment the lesion. One of the key features of their pipeline is a 3D - Conditional Random Field as post-processing step, to add spatial coherence in all the dimensions of the input volume. They use the formulation proposed in [16].

Some of the top entries of the ISBI - LiTS Challenge were also based on DCNNs. [11] were the winners of the challenge. They trained a DCNN model that takes as input a stack of adjacent channels (2.5D) from scratch, and use both long range (skip connections) and short range connections from residual blocks. They also focus on the liver region by training a first network that outputs an approximate location of the liver. As post-processing step, they do 3D connected component labeling. [1] used also residual networks, forming a cascaded architecture which gradually refines the segmentation of the previous iteration. Another group combines deep learning with classical classifiers, such as [5]. They use 2D U-net architecture with a random forest classifier. First the liver is segmented using an ensemble of CNNs using the LiTS and an external dataset. Then another network is used to segment the lesion, followed by connected components and finally filtering the false positives with a forest classifier. Compared to all these approaches, we instead propose a framework that uses detection to localize the lesions, removing possible false positive pixels triggered by the segmentation network.

Chapter 3

Detection-aided medical image segmentation using deep learning

3.1 Baseline Architecture

Our approach is based on DRIU [24], an architecture for retinal image segmentation that segments the blood vessels and optic disc on fundus images. The architecture uses VGG-16 [28] as the base network, removing the last fully connected layers, so that the network consists on convolutions, activation functions and max poolings. The base network is pre-trained with Imagenet [7] and consists of a set of convolutional stages, each of them working at the same feature map resolution, and separated by the pooling layers. As the network goes deeper, the information is coarser and the learned features are more related to semantics. On the other hand, at the shallowest feature maps that work at a higher resolution, filters capture the most local information. DRIU's goal is to segment the optical disc and the blood vessels from fundus images. The blood vessels are fine structures that can benefit from the shallow layers, as opposed to the optic disc that is a coarser structure. To take advantage from the information learned at feature maps that work at different resolutions, DRIU uses several side outputs with supervision. A side output is a set of convolutional layers that are connected at the end of an specific convolutional stage from the base network. Each of these side outputs specializes on different types of features, depending on the resolution at which the base network is working on at the connection point. In DRIU, the feature maps produced by each side output are resized and combined linearly to output the final result.

3.1.1 Cascaded architecture for liver and lesion segmentation

The medical segmentation network proposed in this work for the liver and lesion segmentation is based on the same architecture as DRIU. In our case there are side outputs after every convolutional stage, and all of them contribute to the multiscale output. The final architecture is depicted in Figure 3.1. It is a cascaded architecture, which first segments the liver to focus on the region of interest in order to segment the lesion. This two-step network leverages the imbalance of positive/negative pixels for the lesion segmentation, as will be discussed in Section 3.2.1. The region of interest results from computing a fixed 3D bounding box of the liver for the whole CT scan volume, and crop each slice with it. Consequently, all cropped ROIs of a volume will have the same dimension. An illustration of this procedure is depicted in Figure 3.2.

We decided to separate the problem into two independent modules: 1) segmenting the liver and 2) segmenting the lesion. For the second task we used the ground truth masks of the liver to work on both

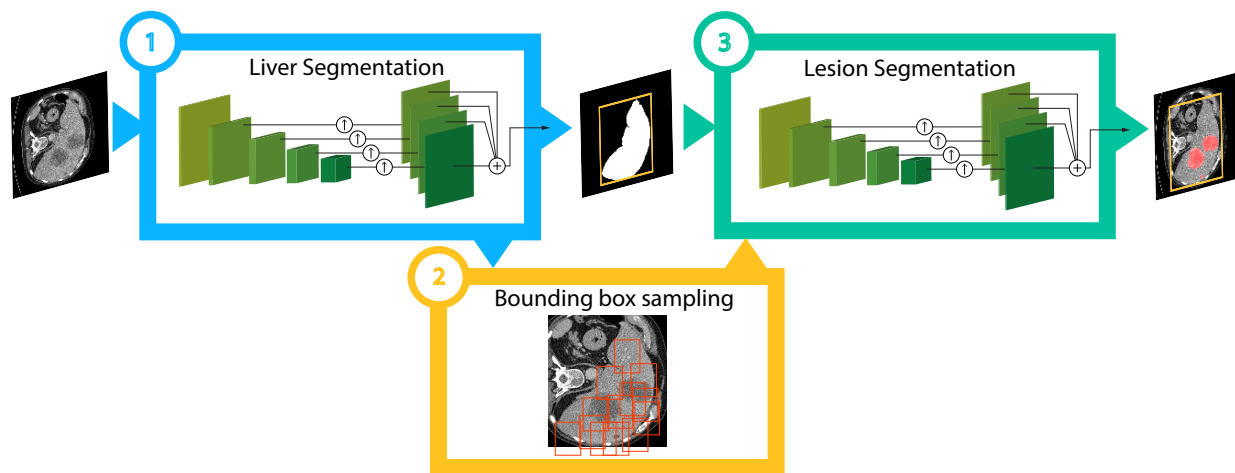


Figure 3.1: Architecture for the detection-aided medical image segmentation. The first stage consists on segmenting the liver. Once we have the liver prediction, we place a 3D-bounding box around the liver, and the different slices cropped by this bounding box are inputted to the lesion segmentation network and also to the detector. The segmentation network for the lesion considers the cropped input image for delineating the lesion. Afterwards, only those localizations that the detector agrees with the segmentation network that there is a lesion are preserved.

modules in parallel.

Regarding the Visceral database, we do not use a cascaded architecture, and the whole image is the input to segment the different organs.

3.2 Loss objective

Regarding the loss objective for training the segmentation network, we worked mainly with the Binary Cross Entropy (BCE) loss. If y is the ground truth and \hat{y} is the predicted value, BCE loss is defined by Eq. 3.1. It computes the binary cross entropy of the output of the network and the ground truth data per-pixel-wise.

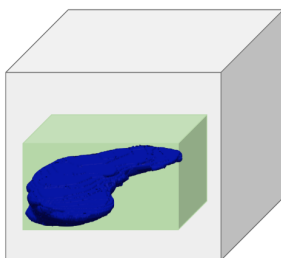


Figure 3.2: This figure illustrates how from the initial 3D volume (gray box) we crop a 3D bounding box around the liver (green box). Each slice of the green bounding box is inputted to the lesion segmentation network, as well as to the detector in order to localize regions with lesions.

$$L(y, \hat{y}) = -y \log \hat{y} - (1 - y) \log(1 - \hat{y}) \quad (3.1)$$

For this loss we used the implementation of [2]. The BCE provides an individual loss per each pixel, and losses coming from positive or negative pixels in the ground-truth can be distinguished. Consequently we can balance differently the positive and negative loss.

3.2.1 Balancing the loss

One of the main challenges of biomedical datasets is that the data is imbalanced; typically more negative samples are provided compared to positive ones. The LiTS dataset is imbalanced in terms of how many images contain lesion, and also it is imbalanced inside an image, since in a positive image, the number of positive lesion pixels compared to negative ones is very small.

In this situation the network parameters could easily fall in the local minimum of outputting all pixels of an image as negative, as this is the majority case. To avoid this, we will balance the Binary Cross Entropy loss (Eq. 3.2) with a variable w in order to give more importance to the loss related to the positive pixels in the ground truth, compared to the one related to the negative pixels.

$$L(y, \hat{y}) = -(1 - w) * y \log \hat{y} - w * (1 - y) \log(1 - \hat{y}) \quad (3.2)$$

We considered two different methods:

- **Per-volume balancing factor** The metric that we use to assess the predicted segmentations is *Dice* (Eq. 3.5). Thus, we aim to maximize this metric, which weights each volume equally. Therefore, it seems logical to weight each slice depending on the volume it belongs to, guaranteeing that the network does not learn more from volumes that are deeper, compared to shallower ones. For each volume V_i , the weighting factor w_i is equal to the number of positive samples divided by the number of negative samples of the volume (Eq. 3.3).

$$w_{i,-} = \frac{|Positive\ Samples\ in\ V_i|}{|Total\ Samples\ in\ V_i|}, \quad w_{i,+} = 1 - w_{i,-} \quad (3.3)$$

- **General balancing factor considering only positive samples of each class** After working with the first weighting strategy, we observed that the weighting factor for a volume that does not contain any lesion is 0, so nothing would be learned from it. This lead us to think of a global weighting factor so that all volumes participated into the learning process. In this new formulation, all slices have the same importance, regardless of the volume they belong to. Moreover, drawing inspiration from the balancing scheme proposed in [9], the different weighting factors for each class will only consider images that actually contain that class. The formulation is in Eq. 3.4.

$$w'_+ = \frac{|Positive\ Samples\ in\ V|}{|Total\ Samples\ in\ positive\ images\ of\ V|}, \quad w'_- = \frac{|Negative\ Samples\ in\ V|}{|Total\ Samples\ in\ all\ images\ of\ V|}$$

$$w_+ = \frac{w'_+}{w'_- + w'_+}, \quad w_- = \frac{w'_-}{w'_- + w'_+} \quad (3.4)$$

In [26], the authors used a methodology to directly optimize for the Dice score (Eq. 3.5), as it is differentiable (Eq. 3.6). It is an intuitive loss because it is the metric that is used to assess the segmentation, and it has been proved to boost performance in their work. However, the loss is not as stable as the Binary Cross Entropy (BCE) loss when there is a small amount of foreground pixels, as it is the case for lesion segmentation. We used the Dice Loss to segment the liver, and although the results are similar, it lead to reduced performance in our experiments.

$$D = \frac{2 \sum_i^N y_i \hat{y}_i}{\sum_i^N y_i^2 + \sum_i^N \hat{y}_i^2} \quad (3.5)$$

$$\frac{\partial D}{\partial y_j} = 2 \left[\frac{\hat{y}_j (\sum_i^N y_i^2 + \sum_i^N \hat{y}_i^2) - 2y_j (\sum_i^N y_i \hat{y}_i)}{(\sum_i^N y_i^2 + \sum_i^N \hat{y}_i^2)^2} \right] \quad (3.6)$$

3.3 Using context slices to exploit 3D information

Until now we have been dealing with the data as if each image was independent from the others, but actually we have volumes of images that have spatial coherence. We could benefit from the redundancy among consecutive images by inputting a volume to the network. Since we are training from the pre-trained weights of Imagenet, the network expects a 3-channel input. We first tried to use these 3 channels to input three consecutive slices and segment all of them simultaneously. At test time, we just keep the central slice from the output volume. This scheme is depicted in Figure 3.3.

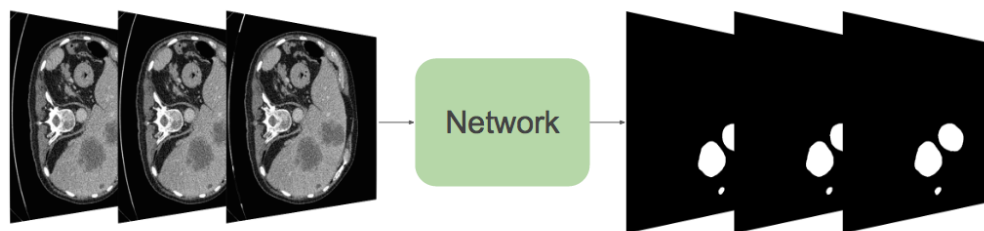


Figure 3.3: In the input of the network 3 consecutive slices are stacked, and at the output, the 3 slices are segmented. This example belongs to the lesion, but the same is applied to the liver.

We also tried to input more slices modifying the filters of the first layer of VGG-16: the original filters are $3 \times 3 \times 64$ and should be modified to be $3 \times n \times 64$, n being the depth of the input volume. In order to initialize the new added filters, we copied the weights of the original pre-trained filter of VGG-16. Our experiments show that inputting 3 consecutive slices yields better results than the baseline. However, inputting more than 3 slices lead to worse results.

3.4 Benefiting from liver segmentation in order to segment the lesion

In this section different strategies that benefit from the liver segmentation to segment the lesion will be commented.

3.4.1 No back-propagation through regions outside liver

The segmentation of the liver allows us to crop the region of interest in order to segment the lesion. Nevertheless, as we know that the lesion is always inside the liver, we can further benefit from the liver segmentation, deciding not to back-propagate through those pixels that are predicted as non-liver. The benefits of this strategy are twofold, 1) the network is just learning from the pixels that actually can belong to the target class, and 2) the positive and negative pixels are more balanced, as the number of negative pixels is greatly reduced. Although the problem becomes more balanced, we still add the balancing term, now just considering the pixels that belong to the liver class. Figure 3.4 depicts the scheme of this configuration. In order to not back-propagate through the pixels outside the liver, we just multiply the output of the lesion segmentation network with the output of the liver segmentation network before back propagating the gradients.

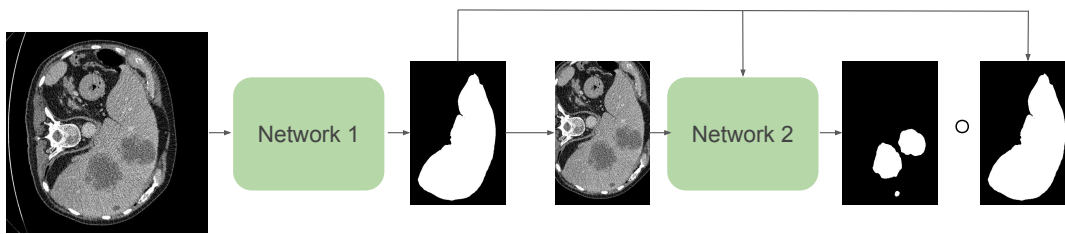


Figure 3.4: Scheme of just back propagating through liver.

3.4.2 Multi-task: Segmenting the liver and lesion simultaneously

Instead of using the fixed liver masks predicted by another network, it could be useful to have a network that learns how to segment the liver and the lesion at the same time. In this way, the segmentation network has the information of the liver while it is segmenting the lesion.

The loss in this case is the sum of the Binary Cross Entropy losses related to the lesion and liver respectively. As both classes are not exclusive, we consider them independent of each other, so the network can decide that a pixel belongs to both classes without any restriction.

3.4.3 Multi-task with no back-propagation through liver

We also tried to use both strategies at the same time. The network can learn to segment the liver and lesion simultaneously, and using the on-line result of the liver to decide through which pixels not to back propagate for segmenting the lesion. In this case the weighting factor for the loss related to the lesion also only considers the pixels that are inside the liver. The final loss is also the sum of the Binary Cross Entropy losses related to each class.

3.5 Lesion Detection

We observed that our network lacked the ability to get a global view of the image, which is helpful to see if there is a lesion or not. As a consequence, some false positives were triggered in almost all images. In order to remove them, we must know in which locations of the image there is an actual lesion, so that we just keep those locations after segmenting the whole input image.

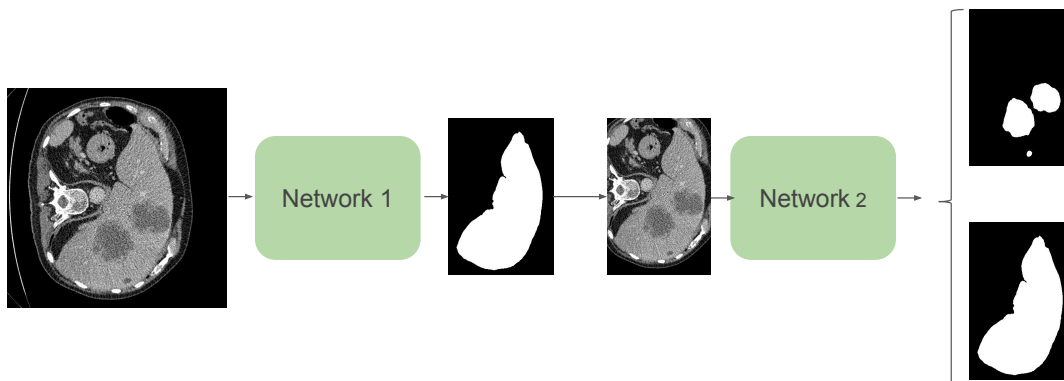


Figure 3.5: Scheme of predicting liver and lesion at the same time. The output of the network are two channels, one for the liver and another for the lesion.

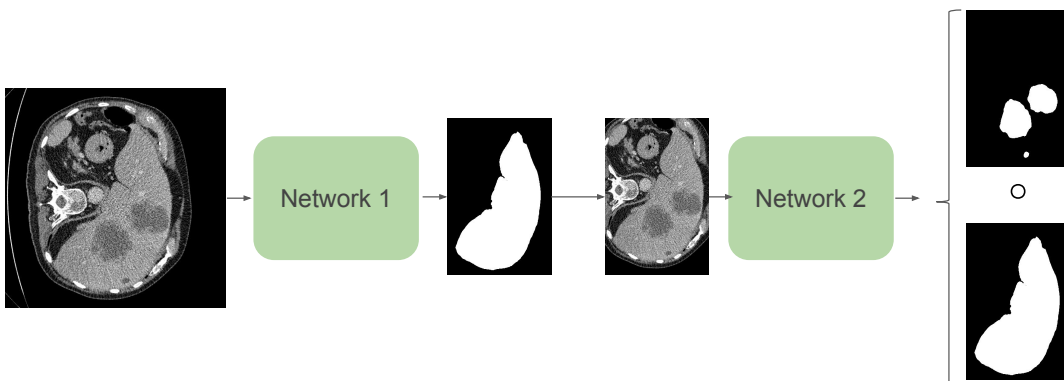


Figure 3.6: Scheme of predicting liver and lesion at the same time. In order to back propagate gradients for the lesion segmentation, first the liver prediction is multiplied by the lesion prediction.

As a first approach, we trained a classifier that labels the whole image as positive or negative. Later, we trained a detector that works at a lower scale and indicates which locations contain a lesion.

3.5.1 Image labeling of images with and without lesion

The first test that we did consists on training an image-level classifier that predicts whether a slice has a lesion or not. This classifier will learn more global features compared to the segmentation pipeline. In order to train such a classifier, we worked with the images cropped around the liver. We have a total of 18576 samples for the 131 volumes. We divide the data into 80/20 training and validation splits. We augment data using two different flips and two different rotations, so we increase our dataset by a factor of 4.

We tried two different configurations, one with VGG-16 and the other with Resnet-50, using the pre-trained weights of Imagenet. In both configurations, we removed the classification layer for Imagenet, and we add our classification layer of a single neuron. The network is trained with the Binary Cross Entropy loss. All images are resized to 224x224 at the input of the network, and the batches are of 32 balanced samples.

	Train	Test
Positive Samples	127K	87K
Negative Samples	9300K	3000K

Table 3.1: Samples after data augmentation for detection

3.5.2 Detection of lesion in a sliding-window fashion

Considering that the classification strategy was a proof of concept to estimate if a CNN could identify a lesion from a global scale, we reduce the scale with a detection pipeline of sliding windows capable of generating finer locations for the liver lesions. In particular, we place bounding boxes in the liver region and then label them as positive or negative, as illustrated in Figure 3.7. The condition in order to place a bounding box is that it overlaps at least 25% with the liver. We use windows of 50x50 pixels, considering a positive observation if there are at least 50 pixels of lesion inside the box. The stride is 50 pixels, and 15 pixels of margin are added to every side of the window to provide additional context, so that each window is finally of size 80x80. The batches are of 64, and are also balanced. The data augmentation in this case is more intensive, as in total we multiply the data by 8, with several flips and rotations. In total the number of samples is depicted in Table 3.1. We consider bounding boxes of a single scale, as in this type of images the concept of scale is not as in natural images, where actually there are big and small objects located at different planes from the camera. In this case, all images are taken at the same global scale, and the lesions by themselves are not instances, but tissues without a defined shape, so defining a scale is not trivial. We chose the dimension of our bounding boxes considering that it should be big enough to cover all lesion tissues with an added context.

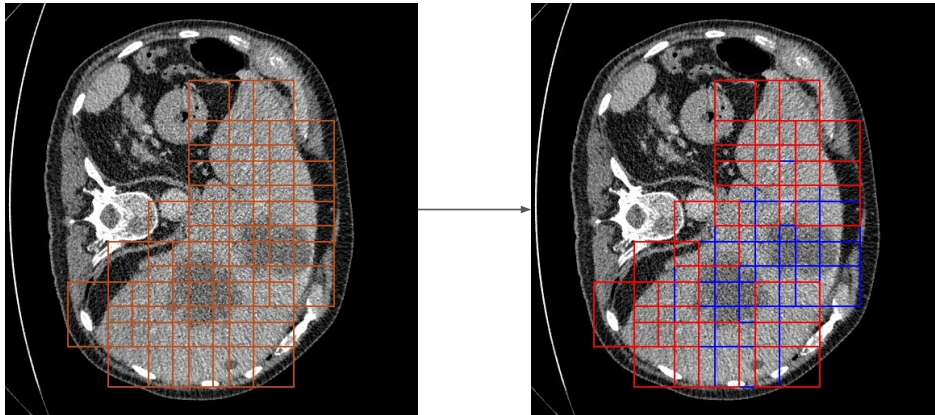


Figure 3.7: Bounding boxes are placed on the liver region, and then labeled as positive or negative for the lesion detection task.

3.6 3D - Conditional Random Fields (3D-CRF)

As a final post-processing step, we add a 3D - Fully Connected Conditional Random Field. A 3D-CRF is a statistical modeling technique applied for structured predictions. CRFs model the conditional distribution of

the output prediction considering all the input at once. The final labels are assigned given the soft predictions outputted by the segmentation network as a maximum a posteriori (MAP) inference in a dense CRF. The model considers both the spatial coherence and also the appearance in terms of the intensity values of the input volume. This 3D-CRF that we use is Fully Connected, so it establishes pairwise potentials on all pairs of pixels in the image, maximizing label agreement between similar pixels. We used the implementation of [6] that uses the 3D-CRF formulation of [16], which states that for a Graph $G = (\mathcal{V}, \mathcal{E})$ with vertices $i \in \mathcal{V}$ for each voxel in the image and edges $e_{ij} \in \mathcal{E} = \{(i, j) \mid \forall i, j \in \mathcal{V} \text{ s.t. } i < j\}$ between all vertices of the graph. The energy function is Eq. 3.7, where x is the label of each vertex. The unary potentials are in Eq 3.8, being I the intensity of the input volume. The pairwise potentials are defined in Eq 3.9, where $\mu(x_i, x_j) = 1(x_i \neq x_j)$ is the Potts function, $|p_i - p_j|$ is the spatial distance between voxels and $|I_i - I_j|$ is the intensity difference.

$$E(x) = \sum_{i \in \mathcal{V}} \phi_i(x_i) + \sum_{(i,j) \in \mathcal{E}} \phi_{ij}(x_i, x_j) \quad (3.7)$$

$$\phi_i(x_i) = -\log P(x_i | I) \quad (3.8)$$

$$\phi_{ij}(x_i, x_j) = \mu(x_i, x_j) \left(w_{pos} \exp\left(-\frac{|p_i - p_j|^2}{2\sigma_{pos}^2}\right) + w_{bil} \exp\left(-\frac{|p_i - p_j|^2}{2\sigma_{bil}^2} - \frac{|p_i - p_j|^2}{2\sigma_{int}^2}\right) \right) \quad (3.9)$$

Depending on the parameters $w_{pos}, w_{bil}, \sigma_{pos}, \sigma_{bil}, \sigma_{int}$ the effect of the pairwise terms and their effective range can be tuned. We tried several combinations for the lesion and liver segmentation, choosing a smaller range for the lesion post-processing, due to the reduced size of lesions compared to liver.

Chapter 4

Experimental Validation

In this section the different experiments and results will be commented on. The first subsection comments on the results of the LiTS dataset, whereas the second one the results of the Visceral Dataset.

4.1 Datasets

4.1.1 LiTS

The database of the LITS challenge consists on 131 CT training scans. Each scan is encoded in a different NifTI (Neuroimaging Informatics Technology Initiative) file, which is a data format typically used for biomedical images encoding. The file extension is .nii, which encodes a single volume, combining data and metadata. There are several libraries to encode and decode this type of files. We used a Matlab library. The test set of the LITS challenge consists of 70 additional CT scans.

The training set of the challenge contains 58,638 images in total. Each volume has a different depth, which ranges from 74 to 987 slices. We did our own partition, keeping 80% from the original training set for our training, and 20% for validation.

First of all we analyzed the data to decide which was the required pre-processing. The histograms of the pixel intensities of the images, the liver and the lesion can be seen in Figures 4.1, 4.2 and 4.3 respectively.

We observed that there were many pixels that had the intensity value of -1024, due to a particularity of the CT image. This value belongs to the background, and does not provide any meaningful information for our segmentation task. We tried two types of processing, both based on min-max normalization (Eq. 4.1), being z_i the normalized volume, and x_i the input volume. The first pre-processing we tried consists in clipping to the maximum/minimum value the pixels outside the range (-1000, 1000), whereas the second type of processing does the same for the range (-150, 250) of the original images, as we observed that liver and lesions belong to this limited range.

$$z_i = \frac{x_i - \min(x)}{\max(x) - \min(x)} \quad (4.1)$$

4.1.2 Visceral

The Visceral dataset is composed of 20 different volumes in different modalities, with annotations for several organs, as the liver, and other structures of the human body, as the aorta artery. We use Visceral dataset to test the generality of the segmentation network. As the number of volumes is lower than in LiTS, we keep 90%

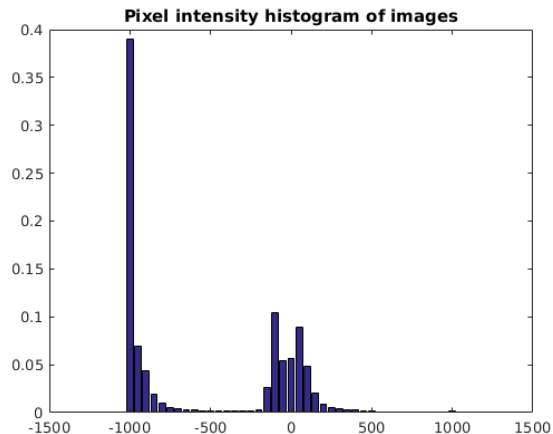


Figure 4.1: Pixel intensity histogram of images from LiTS dataset.

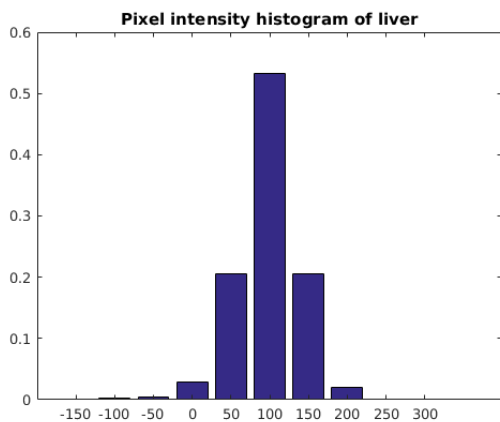


Figure 4.2: The pixel intensity histogram for the liver.

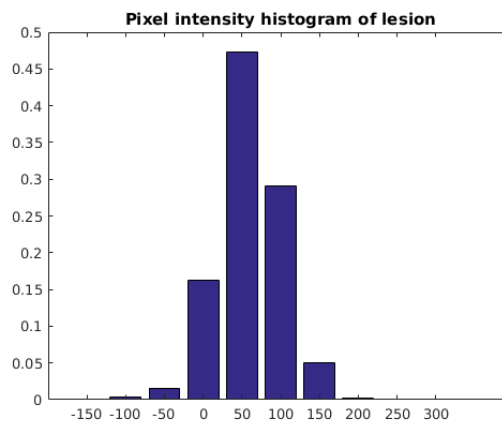


Figure 4.3: The pixel intensity histogram for the lesion.

of the data for training and 10% for validation. There are some volumes that do not have the annotations for all organs. In our experiments, we just worked with the 15 structures that were annotated in all volumes.

Following the same criteria as for the LiTS dataset, we study the histograms of the pixel intensities from the images and also from the different organs (Figures 4.4 and 4.5). In this case, we observe that the organs cover all the range, so we work in the range $(-1000, 1000)$.

4.2 Metrics

The main metric that will be used to assess the results in this work is the Dice, which is the same as the F1-score. For the following experiments, we will refer to two different variants of Dice score. First, the average Dice across volumes, which is the metric assessed in the LiTS challenge. We will refer to this metric as \overline{Dice} . Then, in order to draw the precision recall curves, we worked with the Dice computed from averaging the precisions and recalls. We will refer to this metric as $Dice(\overline{P}, \overline{R})$.

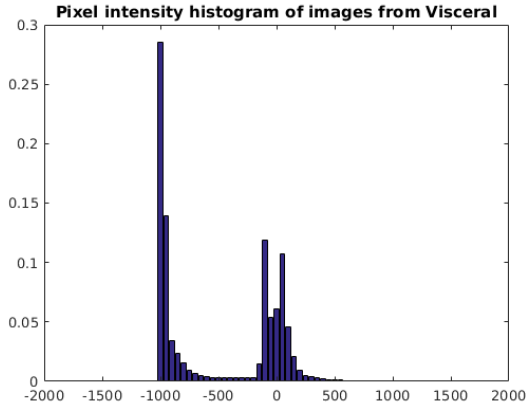


Figure 4.4: The pixel intensity histogram of images from Visceral dataset.

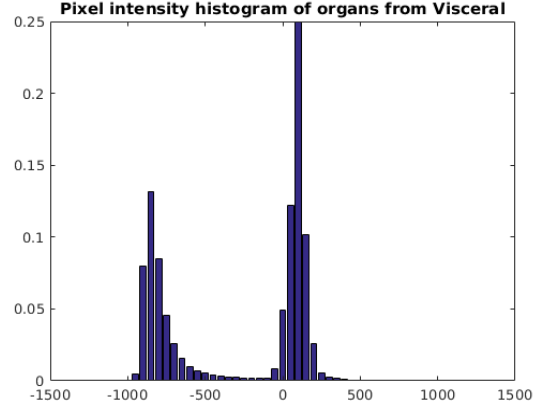


Figure 4.5: The pixel intensity histogram of the organs analyzed from the Visceral dataset.

4.3 Experiments on LiTS dataset

Regarding the LiTS dataset challenge, we present results for the lesion and the liver segmentation. For the configurations of lesion segmentation, we first crop the volume-wise bounding box around the liver from the ground truth to train the different models.

4.3.1 Data Preparation

We first wanted to experiment the implications of working with the range of values that belong to the liver region instead of using a wider range. In Table 4.1 we observe that when limiting the range the performance of the lesion segmentation improves, whereas for the liver the difference is not that significant. This experiment was tested with the per-volume balancing approach to weight the binary cross entropy loss.

4.3.2 Balancing the Binary Cross Entropy Loss

We analyzed different methods to balance the binary cross entropy loss commented in Section 3.2.1 and their performance is compared in Table 4.2. We first experimented with the Per-volume balancing method explained in Section 3.2.1. As it can be seen in the Precision-Recall Curve in Figure 4.6, this approach yielded too many false positives. We tried to tune the formulation of the balanced cross entropy, by doing the following change: $w_+ = w_+ - \alpha$ and $w_- = w_- + \alpha$, just to verify that by giving more importance to the loss related to the negative pixels we could reduce the number of false positives. We empirically defined α to be 0.15, which improved the results. We also tested the General balancing method, which is slice independent and only considers for balancing a certain class the images that contain that class. This

	Lesion	Liver
Processing (-1000,1000)	0.300	0.939
Processing (-250, 150)	0.318	0.942

Table 4.1: Average Dice across volumes for the two processing techniques.

	Lesion		Liver	
	\overline{Dice}	$Dice(\bar{P}, \bar{R})$	\overline{Dice}	$Dice(\bar{P}, \bar{R})$
BCE - Per-Volume balancing	0.3181	0.3610	0.9420	0.9435
BCE - Tuned Per-volume balancing	0.3293	0.3796	0.9542	0.9549
BCE - General balancing	0.3433	0.3824	0.9505	0.9511
Dice Loss	-	-	0.9391	0.9402

Table 4.2: Dice for Lesion and Liver using losses.

is the method that yields the best result for the lesion. As can be seen in Table 4.2, on the liver the best result is achieved with the tuned version of the Per-volume balancing weighting, but still, we prefer using the General balancing method as it does not depend on an empirically chosen parameter dependent on the database. We also trained a network to segment the liver using the Dice Loss, but this resulted in lower Dice. For the lesion we did not achieve a stable training using the Dice Loss.

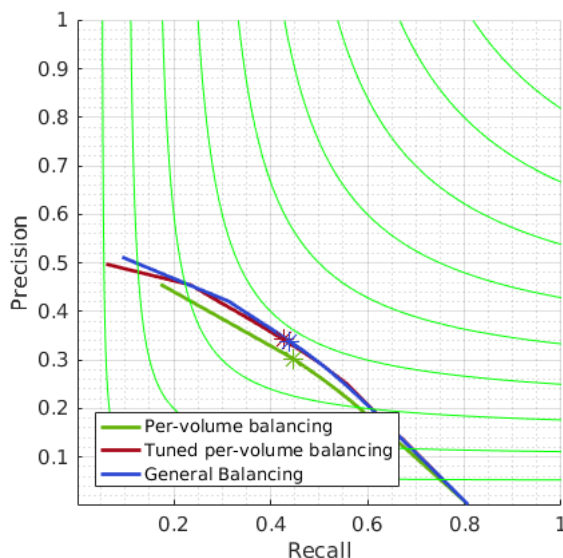


Figure 4.6: Precision- Recall Curves for different balancing methods for the lesion.

4.3.3 Using context slices to exploit 3D information

To benefit from the redundancy between consecutive slices, we segment several consecutive slices at the same time, as commented in Section 3.3. At test time only the central slice is kept. The results for different number of slices can be seen in Table 4.3. The best result is when feeding 3 consecutive slices. Inputting 6 different slices also improves the result compared to the baseline, but 3 slices is still the optimal. As we tested inputting more slices, the results worsened. We also tried to input 3 consecutive slices for the liver segmentation, and it also improved as can be seen in Table 4.4. The Precision-Recall curve for both the liver and the lesion are in Figures 4.7 and 4.8.

	1 slice	3 slices	6 slices	9 slices
\overline{Dice}	0.3433	0.3596	0.3496	0.3259
$Dice(\bar{P}, \bar{R})$	0.3824	0.4070	0.3908	0.3641

Table 4.3: Comparison of inputting several consecutive slices for the lesion.

	Baseline	3 slices
\overline{Dice}	0.9505	0.9574
$Dice(\bar{P}, \bar{R})$	0.9511	0.9579

Table 4.4: Results for the liver segmentation.

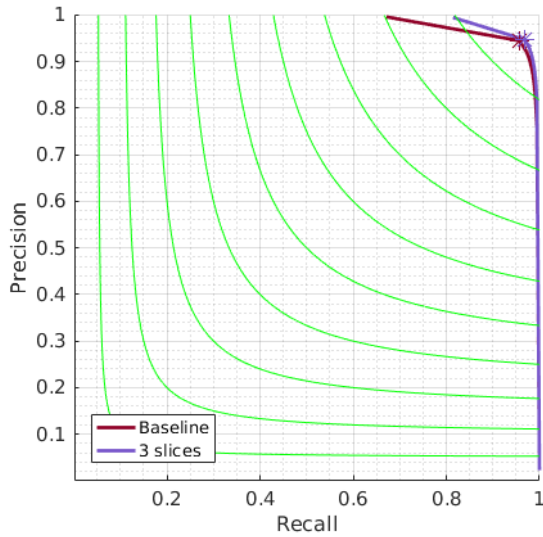


Figure 4.7: Precision- Recall Curves when the input is a different number of slices for the liver.

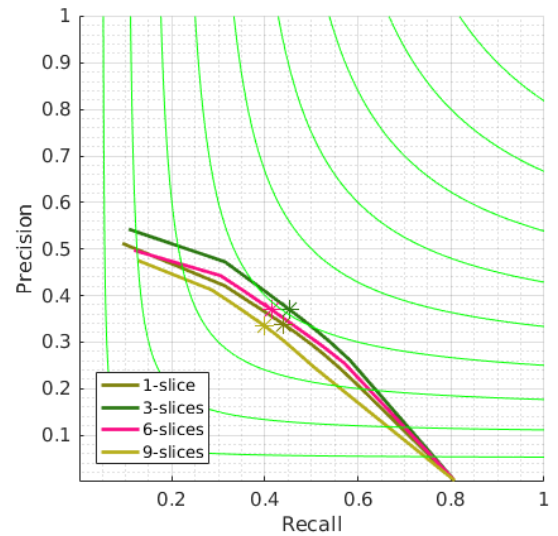


Figure 4.8: Precision- Recall Curves when the input is a different number of slices for the lesion.

	Lesion	
	\overline{Dice}	$Dice(\bar{P}, \bar{R})$
3 slices (baseline)	0.3596	0.4070
3 slices + Masking	0.3600	0.4086
3 slices + Masking + BP trough liver	0.3791	0.4296
3 slices + Multi-task	0.3878	0.4149
3 slices + Masking + Multi-task	0.3858	0.4156
3 slices + Masking + Multi-task + BP trough liver	0.3826	0.4155

Table 4.5: Different methods to benefit from liver segmentation.

4.3.4 Using the liver for segmenting the lesion

In this section we will comment the results on the different strategies presented in Section 3.3. In Table 4.5 we show results for the different methods tested to benefit from the liver information in order to segment the lesion. The baseline consists of using 3 slices at the input.

The *Masking* option refers considering only as candidates for lesion does pixels predicted as belonging to the liver. The *BP through liver* configuration refers to training the network by back-propagating only through pixels that belong to the liver, which also entails that at test time the lesion segmentation will also be masked by the predicted liver. We observe that back-propagating through the liver significantly improves the result to just masking with the liver at test time. We believe this is due to the training being done on relevant pixels of the image.

Another configuration that we tried is the *Multi-task* approach, which consisted on learning to segment the liver and the lesion simultaneously. It also resulted better than the baseline, even without masking, which indicates that what the network learns for segmenting the liver is beneficial for segmenting the lesion as well. Finally we tested imposing no back-propagation trough those pixels that during multi-task were predicted as liver. Although the approach was better than the baseline, the two strategies separately achieve the best results, each of them in a different metric, as can be seen in Table 4.5. In Figure 4.9 there are some examples that depict how both strategies work. We observe that both multi-task and back-propagation through liver strategy are able to detect more lesions compared to the baseline, and that the borders are more precise. The back-propagation strategy shows better results. In the Appendix there are more visualizations for smaller lesions.

4.3.5 Training a Lesion Detector

In this section we will show the results obtained when adding the detection (Section 3.5) on top of the segmentation in order to remove false positives triggered by the segmentation network.

We first train a classifier of lesion/non-lesion image. We tested the two different configurations introduced in Section 3.5, one with VGG-16 and another one with Resnet-50. Table 4.6 and Figure 4.10 depict how Resnet-50 network outperforms the results over the VGG-16 configuration.

The preliminary results on image classification motivated us to train a detector that works at a lower scale, so we train a lesion detector based on Resnet-50. The Precision-Recall Curve we obtain can be seen in Figure 4.11 and the maximum F-score, Precision and Recall, are in Table 4.7.

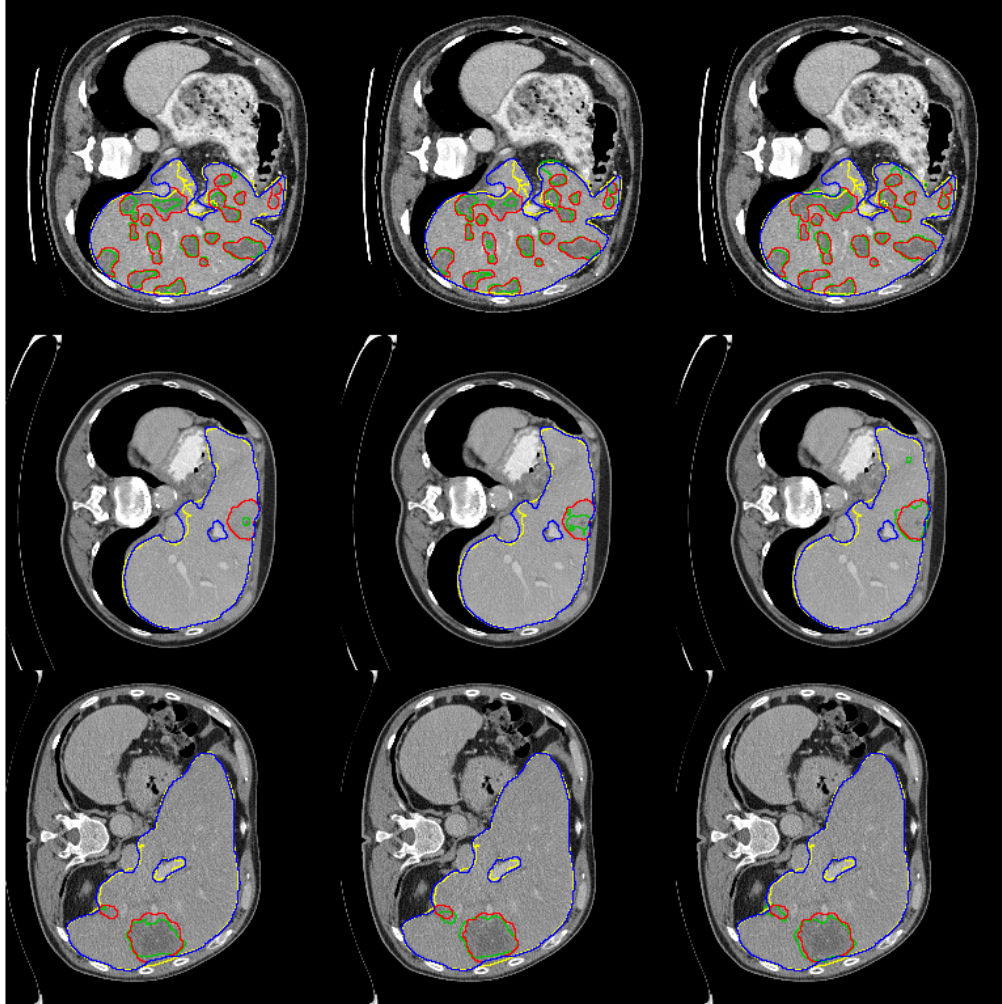


Figure 4.9: The first column is the baseline (3 slices in the input), the second one is doing multi task, and the third is only back propagating through the liver. The blue lines are the liver ground-truth, the red lines are the lesion ground-truth, the yellow lines are the predicted liver and the green lines are the predicted lesion.

4.3.6 Removing False Positives using the Lesion Detector

We want to check how does the detector help the segmentation network. From the best two configurations we have, the one that does the multi-task of segmenting the liver and the lesion simultaneously (Section 3.4.2), and the one that only back-propagates through the liver (Section 3.4.1), we sample bounding boxes around the liver region and predict whether those localizations belong to an unhealthy tissue. From the output of the segmentation network, we simply keep those localizations that are detected as having a lesion in it. In Table 4.8 we can observe the results. Applying the detector improves the dice score for both strategies. The detector indicates which windows of the image contain lesion, so we just keep those windows for the final

	Precision	Recall	F-score	Optimal th
VGG-16	0.6169	0.8259	0.7063	0.014
Resnet-50	0.7807	0.7417	0.7607	0.31

Table 4.6: Performance of the Image Classifier on

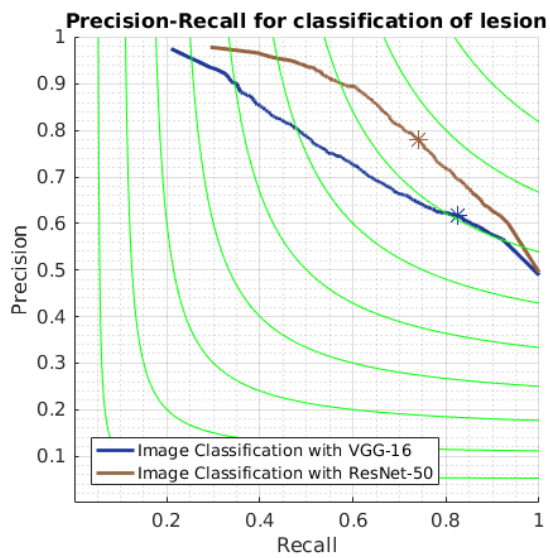


Figure 4.10: Precision-Recall curve for the classification.

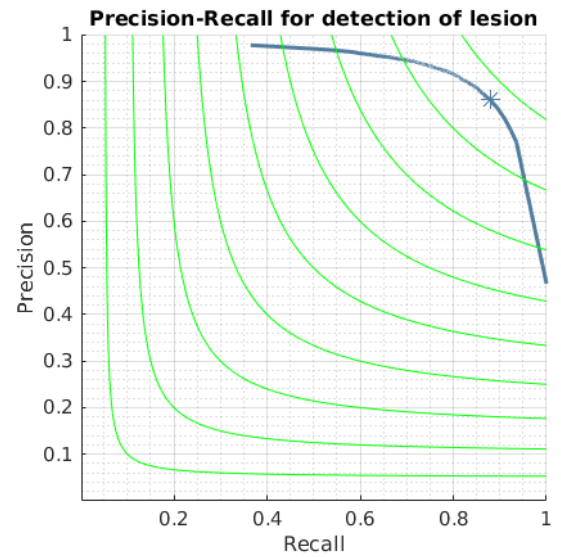


Figure 4.11: Precision-Recall curve for the detection.

Precision	Recall	F-score	Optimal th
0.8698	0.8790	0.8698	0.019

Table 4.7: Image Window detections performance

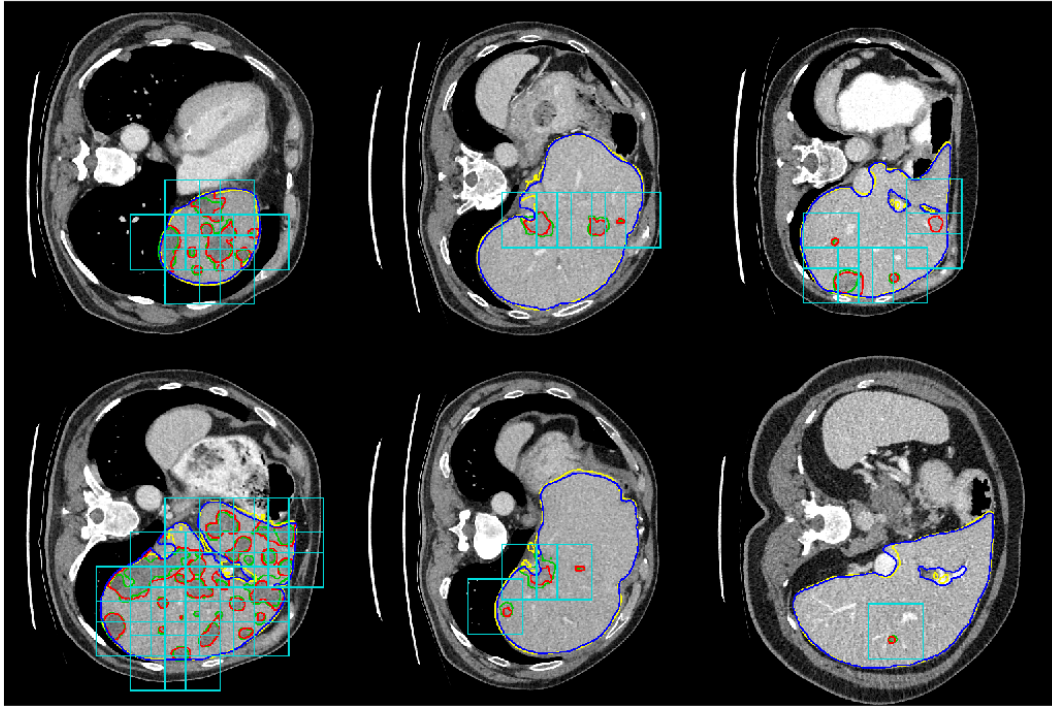


Figure 4.12: In this Figure several CT slices are depicted. The blue bounding boxes are the windows detected as having a lesion. All positive pixels at the output of the segmentation network will be removed if they are not detected for the detector as well.

segmentation. The Precision-Recall Curve is illustrated in Figure 4.13, where it is shown that the detection improves for both configurations. In Figure 4.12 some examples of how the detector performs are depicted. This detector is useful to remove small regions that are false positives as is depicted in several visualizations added in the Appendix.

To conclude, the final best configuration is then the one that only back-propagates through the liver and uses the detector to remove false positives.

Volume-wise crops from predicted liver

All the results until now have been computed using the volume-wise crop from the ground-truth, but we need to calculate the volume-wise crops from the predicted liver, and segment the lesion again. We developed a simple strategy which considers the number of positive pixels obtained at each slice, and fits it into a Gaussian. Details of the algorithm are in the Appendix. In the second column of Table 4.8 there are the results when using the predicted liver for computing the crops. We observe that the performance decreases slightly, as expected.

	Lesion - Crop		Lesion - Crop (test)	
	\overline{Dice}	$Dice(\bar{P}, \bar{R})$	\overline{Dice}	$Dice(\bar{P}, \bar{R})$
Multi-task	0.3878	0.4149	0.3782	0.4078
Multi-task + Masking	0.3858	0.4156	0.3787	0.4092
Multi-task + Detector	0.4046	0.4278	0.3989	0.4226
BP through liver + Masking	0.3791	0.4296	0.3791	0.4296
BP through liver + Masking + Detector	0.4120	0.4474	0.4079	0.4437

Table 4.8: Results when filtering the segmentation mask with a lesion detector

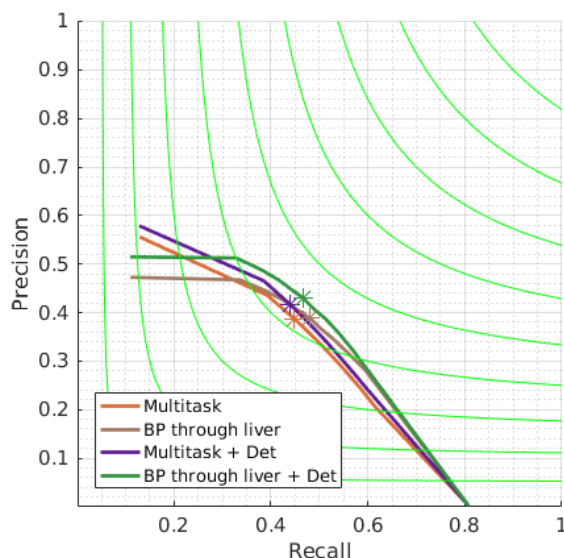


Figure 4.13: Precision Recall Curve after removing negative detections.

4.3.7 3D Conditional Random Fields

At the end of our pipeline we decide to apply a 3D Conditional Random Fields (Section 3.6), using the implementation provided by the authors of [6]. At the input of the 3D-CRF model, we introduce the soft prediction outputted by the network and the preprocessed volume. We work with 3D-CRFs for two reasons. First, the 3D-CRF will help our method to gain spatial coherence. Second, we want to verify if actually our detector adds performance even when applying a heavy post-processing such as a 3D Conditional Random Fields. We take the best configurations we have and apply the Conditional Random Field on top. In Table 4.9 we can see the results. We notice that from both baselines considered in this table, the multi-task and the BP through liver, the 3D-CRF improves results, although not as much as the detector does. Then, we compute how are the results if we use both the detector and the 3D-CRF, and we observe that this leads to a highest Dice score, so both detector and 3D-CRF boost performance in the pipeline. In Table 4.10 the 3D-CRF is applied on the liver segmentation, also yielding better results. In Figure 4.14 some examples of how does the CRF improve the results after the detector are shown.

	\overline{Dice}	$Dice(\bar{P}, \bar{R})$
Multi-task	0.3878	0.4149
Multi-task + 3D-CRF	0.4004	0.4254
Multi-task + Detector	0.4046	0.4278
Multi-task + Detector + 3D-CRF	0.4299	0.4473
BP through liver	0.3791	0.4296
BP through liver + 3D-CRF	0.4027	0.4424
BP through liver + Detector	0.4120	0.4474
BP through liver + Detector + 3D-CRF	0.4278	0.4582

Table 4.9: Results of lesion after applying 3D-CRF

	\overline{Dice}	$Dice(\bar{P}, \bar{R})$
Liver	0.9511	0.9579
Liver + 3D-CRF	0.9620	0.9621

Table 4.10: Results when added 3D-CRF to liver

	Sub 1	Sub 2	Sub 3	Top Entry	Sub Liver	Top entry Liver
Dice per case	0.54	0.57	0.59	0.68	0.94	0.96
Dice global	0.72	0.72	0.74	0.79	0.95	0.96
VOE	0.40	0.39	0.37	0.34	0.11	0.08
RVD	0.37	0.34	0.24	0.16	0.01	0.01
ASSD	1.51	1.43	1.36	1.02	2.90	1.24
MSD	9.63	9.28	8.27	7.12	90.24	26.56
RMSD	2.26	2.17	2.04	1.66	7.90	2.66
Precision	0.02	0.06	0.04	0.12	-	-
Recall	0.19	0.18	0.22	0.30	-	-
Jaccard	0.60	0.61	0.63	-	-	-
Dice	0.75	0.75	0.77	-	-	-

Table 4.11: Results obtained at MICCAI LiTS Challenge. The three first columns are the three different submissions we did during the MICCAI round. The 4th column is the best result in the LiTS Challenge, although we still do not know the identity of the team or the procedure they followed. The top entry gets better results in all the metrics. In bold we marked the best result for the lesion from our different configurations. The Jaccard and the Dice are metrics just given when doing the submission, so we can not know these two metric for the other entries except for ours. In the last two columns there are the results for the liver, with our submission and the top entry. In this case the Challenge platform does not provide the Precision, Recall, Jaccard and Dice, and that is why those cells are not fulfilled.

Sub 1	3-slices, BP through liver
Sub 2	3-slices, BP through liver + Detector
Sub 3	3-slices, BP through liver + Detector + 3D-CRF
Sub Liver	3-slices, 3D-CRF

Table 4.12: Legend of the architecture for each submission of MICCAI LiTS Challenge.

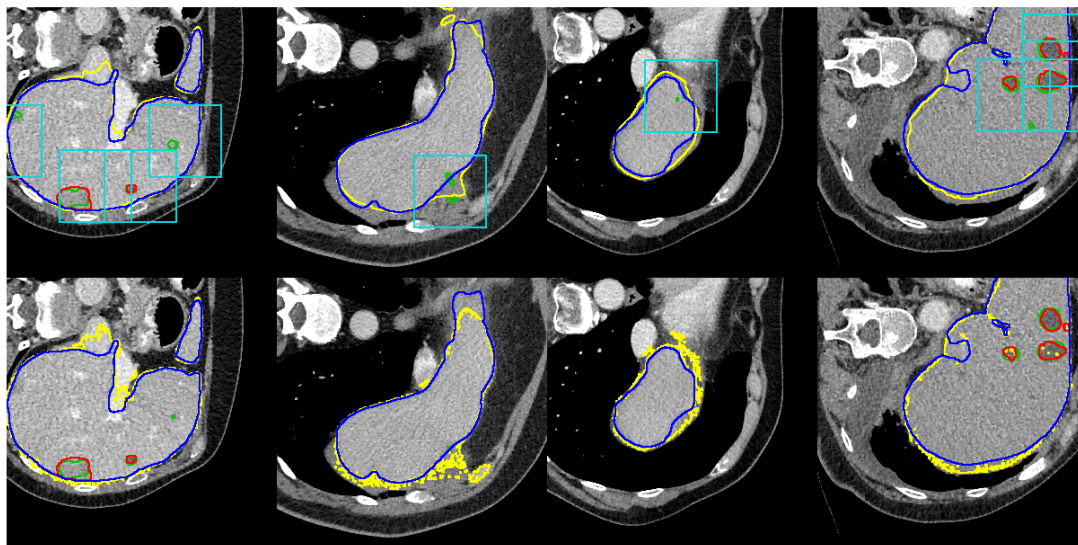


Figure 4.14: First row are the results after the detector. These images show mistakes, where both the detector and the segmentation network fail. 3D-CRF can help to remove false positives considering both the spatial coherence of the volume and the appearance. The second row depicts the same examples, this time applying 3D-CRF for the liver and for the lesion. Some small mistaken lesions are removed by the 3D-CRF.

4.3.8 LiTS competition

This section overviews our participation in LiTS competition. First of all, we should highlight that during the development of this thesis, there have been two different openings of the challenge. The first one was for the workshop of International Symposium on Biomedical Imaging Conference (ISBI) 2017, and then the challenge opened again for a workshop of Medical Image Computing and Computer Assisted Interventions Conference (MICCAI) 2017. The configuration that we submitted for the ISBI-LiTS challenge was the baseline of using a cascaded architecture, the Per-volume balancing scheme and masking the lesion prediction with the predicted liver. We obtained 0.41 of Dice score, far from the top entry of the leader board which had a 0.67 of Dice score.

In the MICCAI Challenge we sent several submissions. The results we obtained are presented in Table 4.11. The challenge will close the submission period on 28th July, 2017. The architecture for each of the submissions is summarized in Table 4.12 and the legend for the metrics is in Table 4.13. The first observation is that the results for the lesion in the on-line validation set are higher than the ones in our own validation set. The opposite happens with the liver segmentation. One first explanation is that our own validation set is very small and maybe it is not enough to actually assess the different algorithms. Another explanation could be the amount of volumes without lesion. In our own validation dataset, there is a considerable number of such volumes that lead to 0 Dice score if there is any false positive pixel. Moreover, in volumes where there is no lesion, segmenting the liver is easier because the texture of the liver is more even. Even with this considerable gap in performance between the on-line validation set and our own partition, there is coherence in the results. The balancing strategy, 2.5D approach, preprocessing of the volumes and only back propagating through liver improved our result from 0.41 to 0.54. Applying the detector improved

Dice per case	Average Dice across volumes
Dice global	The dice if all volumes are considered as a single one
VOE	Volumetric Overlap Error
VOE	Volumetric Overlap Error
RVD	Relative Volume Difference
ASSD	Average Symmetric Surface Distance
MSD	Maximum Symmetric Surface Distance
RMSD	Root Mean Square Symmetric Surface Distance
Precision	Precision at 50% overlap
Recall	Recall at 50% overlap
Jaccard	The Jaccard index slice per slice
Dice	The Dice index slice per slice

Table 4.13: Legend of the different metrics of LiTS Challenge.

	Baseline		Multi-task		3 slices		Multi-task 3 slices	
	\overline{Dice}	$Dice(\bar{P}, \bar{R})$	\overline{Dice}	$Dice(\bar{P}, \bar{R})$	\overline{Dice}	$Dice(\bar{P}, \bar{R})$	\overline{Dice}	$Dice(\bar{P}, \bar{R})$
Liver	0.94207	0.9422	0.93832	0.9386	0.94429	0.9444	0.93152	0.9317
R. Kidney	0.88806	0.8890	0.89002	0.8901	0.88841	0.8890	0.81785	0.8179
Spleen	0.93444	0.9350	0.92784	0.9294	0.92552	0.9267	0.89254	0.9841
R. lung	0.97638	0.9764	0.97628	0.9763	0.97714	0.9772	0.96862	0.9687
Aorta	0.8105	0.8128	0.82177	0.8243	0.84794	0.8500	0.78149	0.7835
Sternum	0.78841	0.7885	0.75305	0.7539	0.80071	0.8009	0.70812	0.7088
R. rectus	0.61692	0.6321	0.56126	0.5816	0.66068	0.6861	0.47048	0.53189

Table 4.14: Visceral Results for several organs. The multi-task approaches are trained for 15 organs, but here we just show results for 7 organs, that are the ones for which we also trained specialized networks.

it up to 0.57, and finally with the 3D-CRF we got the final result of 0.59, so we can say that from our baseline we improved from 0.41 to 0.59, which is a 44% of improvement. The challenge is still open at the time of writing this report, and the top score entry has a 0.68 of Dice score. We notice that there is still a significant gap between our approach and the top entry, but we think that the different techniques that improved our result could also be applicable to other pipelines, more importantly the fact of using the detector in combination to the segmentation network in order to have the best of both worlds, a network that is trained at a local scale, and a detector that works at global scale and learns to generalize from a bigger liver tissue.

4.4 Experiments on the Visceral dataset

We also trained several models on the Visceral dataset in order to test how our segmentation network performs in general for biomedical images.

First, we train networks that are specialized in a single organ. From Table 4.14, the first column shows the results of this configuration. It consists on using the DRU network, inputting a single slice at a time, and balancing with the General balancing method that we defined. Experimenting with the LiTS dataset, we saw that learning where the liver is at the same time as learning the lesion, improved the results. This same

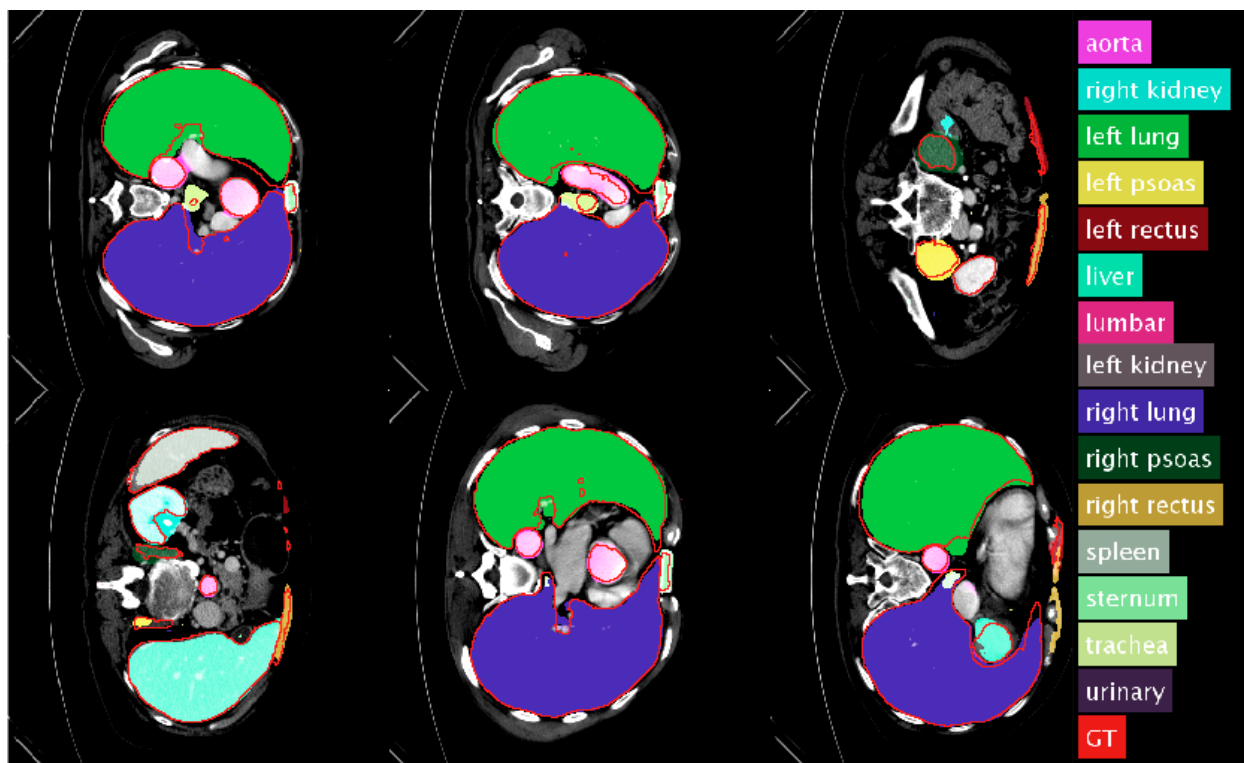


Figure 4.15: These are 6 slice examples. The different organs (15 in total) have their prediction masked in different colors, that can be seen in the legend at the right of the Figure. The ground truth for all the organs is in red.

idea should happen with anatomical structures from CT scans: the more the network understands, the better it will distinguish the different parts.

Also, having a model for each organ is not practical if you seek a multi-organ segmentation, as you need several forwards passes. Then, a single network that is able to delineate multiple structures would be very practical, so this is exactly what we try. Our network will output as many channels as structures we are predicting, a total of 15. The results of this second configuration are in the second column of the Table, and we observe that the performance is quite similar to the first configuration, and even better for some organs such as the right kidney or the aorta.

Then we want to check if segmenting several slices simultaneously also helps in this problem. The results are in the third column, and we observe that indeed for most organs this is the best configuration.

Finally, we train a single network that tries to segment 3 slices of each organ at the same time. As we are working with 15 organs, the final number of channels is 45. This last configuration lead to worse performance, we think it may be because the number of output channels is quite big compared to the number of filters learned at the side outputs, which could act as a bottleneck. In Figure 4.15 some examples of how does the multi-task approach work for segmenting 15 different organs are depicted.

Chapter 5

Budget

In this section the budget to perform this project is estimated. It is important to highlight that this project is software-based, so there is not a final physical product created. Moreover, there is no aim in selling the final outcome of this thesis, so there will be no analysis in this matters.

One important part of the budget comes from the personals costs. There has been a junior engineer working as a full-time worker for the master thesis, and weekly meetings have been held with the supervisors, that will be counted as senior engineers. In Table 5.1 we can see the total personal costs, that are a total of 19440 €. The software has been developed in python and Matlab. Matlab requires of a license, whose amortization for the months that we have used is are in Table 5.2. This sums 250 €.

In order to develop this project the GPU cluster of the ETH has been used. To approximate the cost of usage of this hardware, we can check how much would it cost to get such service from the Amazon Web services (AWS) at their cloud computing service. The time that we have been running jobs in the cluster is approximately of 4 months. During this time, on average 2 jobs have been running all day, each of them in a GPU with 12GB of RAM. The most similar resource that AWS provides is the p2.xlarge, which costs 0.9 USD/hour, a total of 21,6 USD/day. The total cost, estimating that we had two of these jobs continuously running during 4 months (approximately 120 days), would be of 5184 USD. The equivalent is of 4,397.90 €(with the conversion 1 USD = 0.848 €at the date 14/08/2017). Adding the different costs, the total budget for the project results of 24087.9 €.

	Number	Wage	Hours / week	Total weeks	Total
Junior Engineer	1	8 €/ hour	40 h	27	8640 €
Senior Engineer	4	20 €/ hour	5 h	27	10800 €

Table 5.1: Total personal costs

Number of Licenses	Price / Year	Months of project	Amortization/Month	Total
1	500 €	6 months	41.67 €/ month	250 €

Table 5.2: Software Licenses cost

Chapter 6

Discussion and Conclusions

In this work we have proposed an algorithm for segmenting the liver and its lesions using a segmentation network and a detector. We have studied how to exploit the characteristics of the provided data, using the depth information at the input of the network, and taking advantage of the liver segmentation prediction in order to segment the lesion. Also, we have segmented more anatomical structures with the segmentation network, proving that the side outputs with supervision for generating a multi-scale output is an architecture that can be used for segmenting quite diverse structures.

Our experimentation validates that, first, using 2.5D with pre-trained weights is feasible using the same number of channels as the original network, 3 in our case. Additional slices at the network's input lead to worse performance, which indicates that actually the power of the pre-trained weights is important for the architecture, and such kind of modifications really affected the overall performance. This just confirms that pre-trained models are strong and can be used for many different applications. However, in the related work, there are successful 2.5D approaches training a network from scratch. New architectures that are appearing lately do not require as much data as it used to some years ago, hence training from scratch a network that fits more our task is something that should also be taken into consideration in order to gain design flexibility.

With our experimentation it has also been demonstrated that using the liver for learning the lesion is quite beneficial. The best approach is the one that only learns from pixels of the liver, which suggests that limiting the samples from which the algorithm learns to just relevant samples, or difficult ones, is favorable to the problem. This strategy is familiar to using "Attention" mechanisms, as there is a location selected (the liver) from which to learn and this improves the learned representations for the lesion. One of the most challenging issues of the lesion segmentation, is to learn properly from positive and negative pixels, due to the imbalance of samples. Limiting the learning to some samples leverages this problem, but we still think that there is room of improvement of how to make the network learn properly from the pixels, as we noticed that any change in the balancing term of the Binary Cross Entropy loss affected the results significantly.

The most important conclusion is the trade-off between fine localization that the segmentation network can achieve, and the generalization that the detector learns. As the output of the segmentation network is pixel-wise, it tends to trigger false positive pixels, as it is not forced to take more global decisions whether there is lesion or not from a wider perspective. This, on the other hand, is what the detector learns, to just decide if a complete patch is healthy or not, not caring about the possible exact shape of the lesion. Having both techniques analyzing the input image yields a better overall result.

Nevertheless, results in the challenge indicate that our method is still far from state of the art results. Our dice score is of 0.59, what would situate us in the 5th best result with the 4th highest Dice score in the ISBI Challenge, being 0.67 the best result achieved for that round of the challenge. In the future it

would be interesting to have a pipeline that is completely end-to-end and does not require computationally demanding post-processing steps or network ensembles that nowadays seem essential to get competitive results. However, we do believe that the different improvements that we achieved compared to our baseline could be applied to other pipelines.

As future work, an architecture that does detection and segmentation all together would be more efficient, and we are sure that one task will benefit from the other. Inspired by the state of the art detection pipelines for image detection that also perform segmentation, only back propagating from those regions that the detector decides should lead to better performance. It is still challenging how to define bounding boxes or localizations for medical tissues, as there is not a concept of an instance as it happens with natural images. The other challenge would be how to define the balancing terms in order to learn properly. The optimal situation would be to remove any balancing term, and being able to chose the locations of the image whose total distribution is already balanced.

Appendix A

Additional material

This appendix has two different sections. First, the algorithm to define bounding boxes from the predicted liver will be explained. Then, more visualizations for the method are illustrated.

A.1 From predicted liver to bounding box

We explain in this section how to obtain 3D bounding boxes from the predicted liver. First of all, from the predicted liver masks, we compute how many positive pixels there are per each slice of a volume. We plot it, and observe that it resembles a Gaussian function. Then, we do a fitting of a Gaussian, so that an estimated mean and variance is obtained. In Figure A.1 we can see an example of the Gaussian fitting. We can use this fitting to remove false positives, considering that all slices above and below a certain σ may not contain any lesion. In order to decide which sigma to choose, we both consider the liver and the lesion. The false positives and false negatives are plotted in Figure A.2. Finally, considering the number of false negative slices for lesion mainly, which is our target, we decide σ to be 3.0, so at a cost of a few false negatives, we remove a considerable number of false positive slices.

A.2 Visualizations

In this section, additional visualizations to understand the method are depicted.

A.2.1 Visualizations of how benefiting from liver helps to segment the lesion

In Figure A.3 we can observe how the the different methods that benefit from the liver segmentation work. These examples depict small lesions, and how applying these strategies helps to delineate these type of lesions.

A.2.2 Using the detector to remove false positives

In Figure A.4 there are some examples of false positives triggered by the segmentation network that are removed after applying the detector, as it is illustrated in Figure A.5.

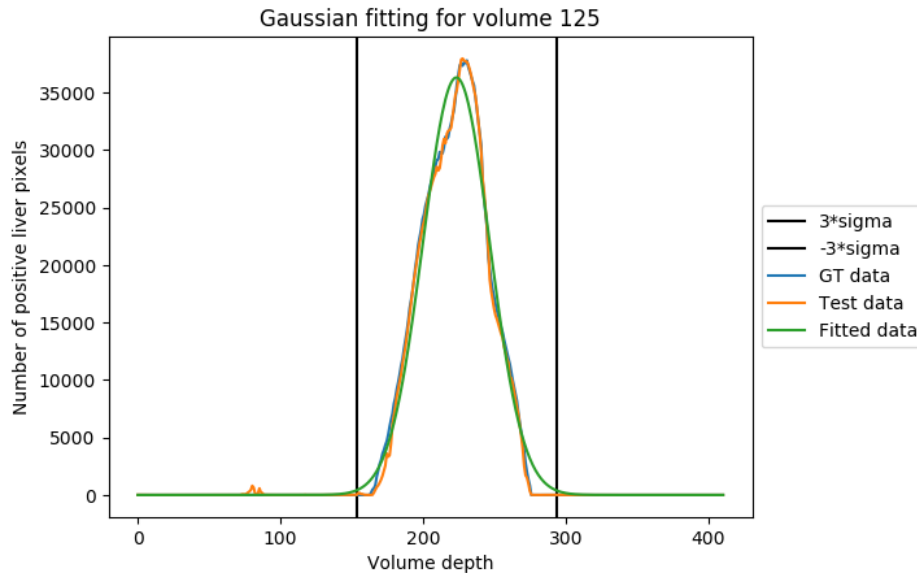


Figure A.1: Gaussian fitting of predicted masks for volume 125.

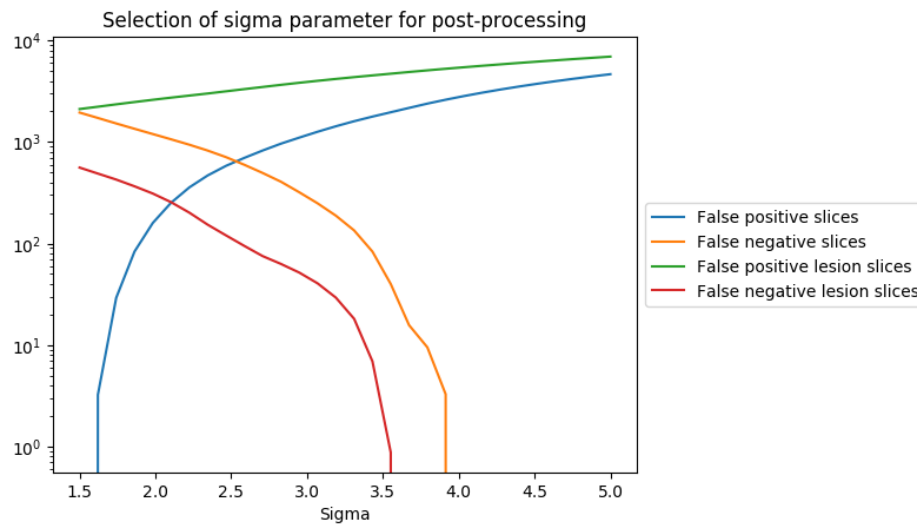


Figure A.2: Analysis of selection of sigma.

A.2.3 Visualizations of the final configuration

Some results for the final configuration are illustrated in A.6.

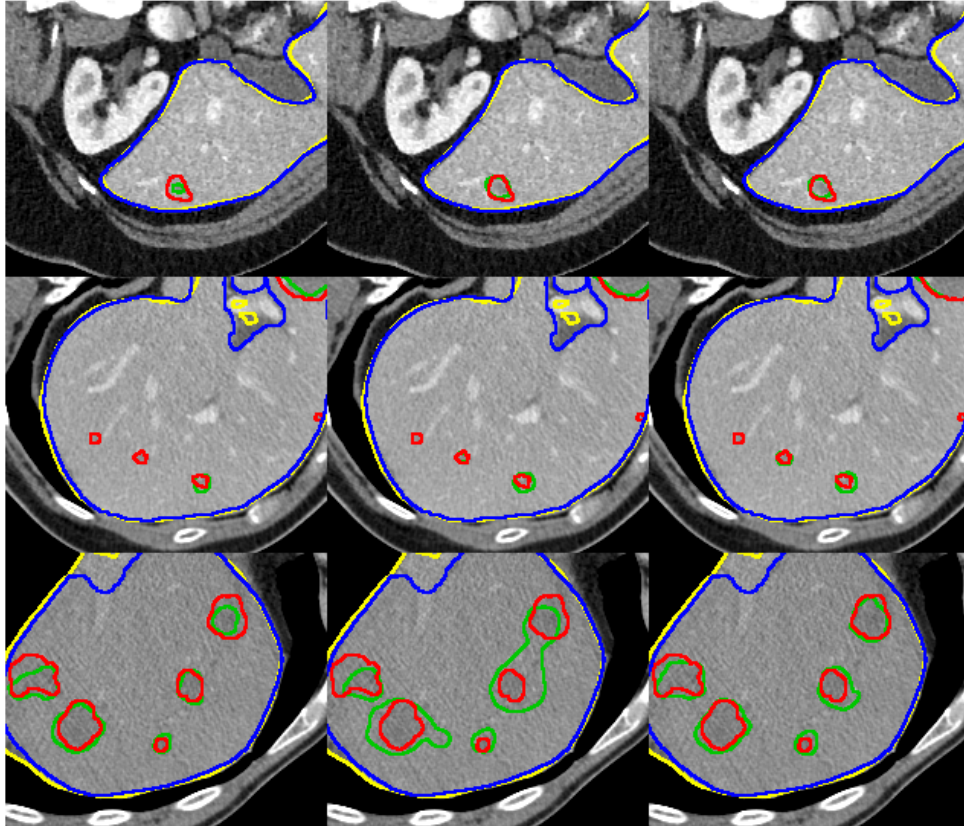


Figure A.3: Zooming to see details in smaller lesions. Each row is a different example from a different volume. Blue lines indicate liver ground-truth, red lines are lesion ground-truth, yellow lines is the predicted liver and green lines is the predicted lesion. The first column is the baseline (3 slices each time in the input), the second column is using the multitask approach, and the third one when there is just back propagation through the liver.

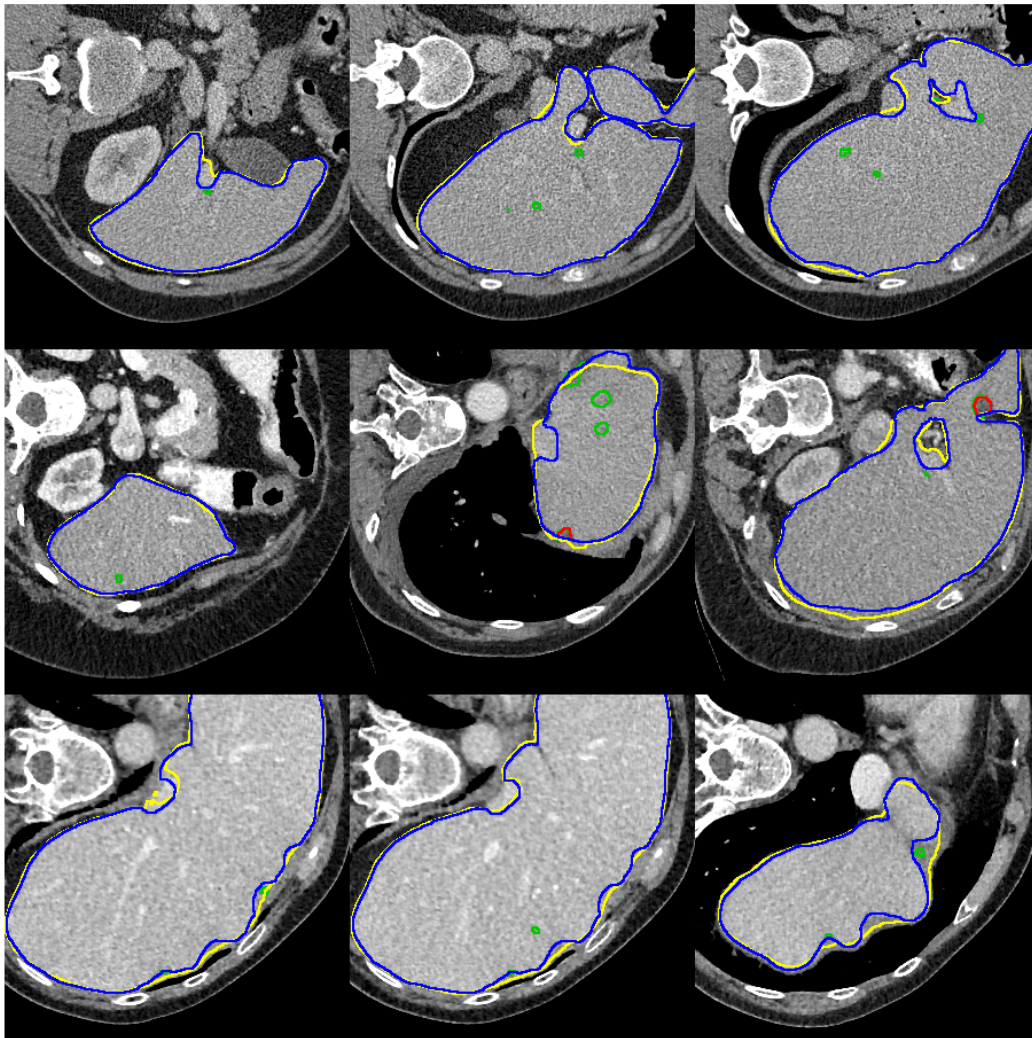


Figure A.4: False positive examples. Each image is a different example, where some fake lesions are predicted by the segmentation network.

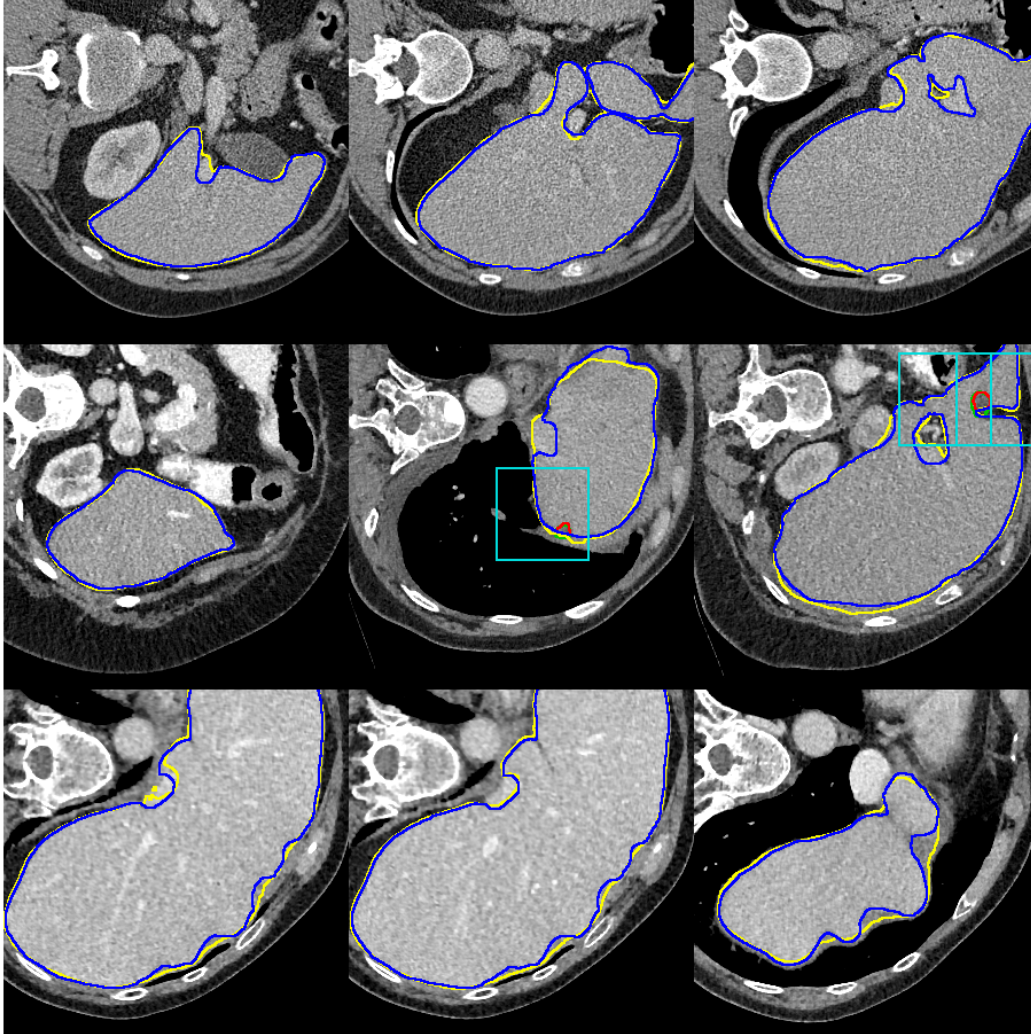


Figure A.5: These examples are the same as in the Figure A.4, but after the detector. Light blue bounding boxes denote the detections of the detector network. We observe that this network does not detect lesions in locations where the segmentation network failed, so false positives are removed.

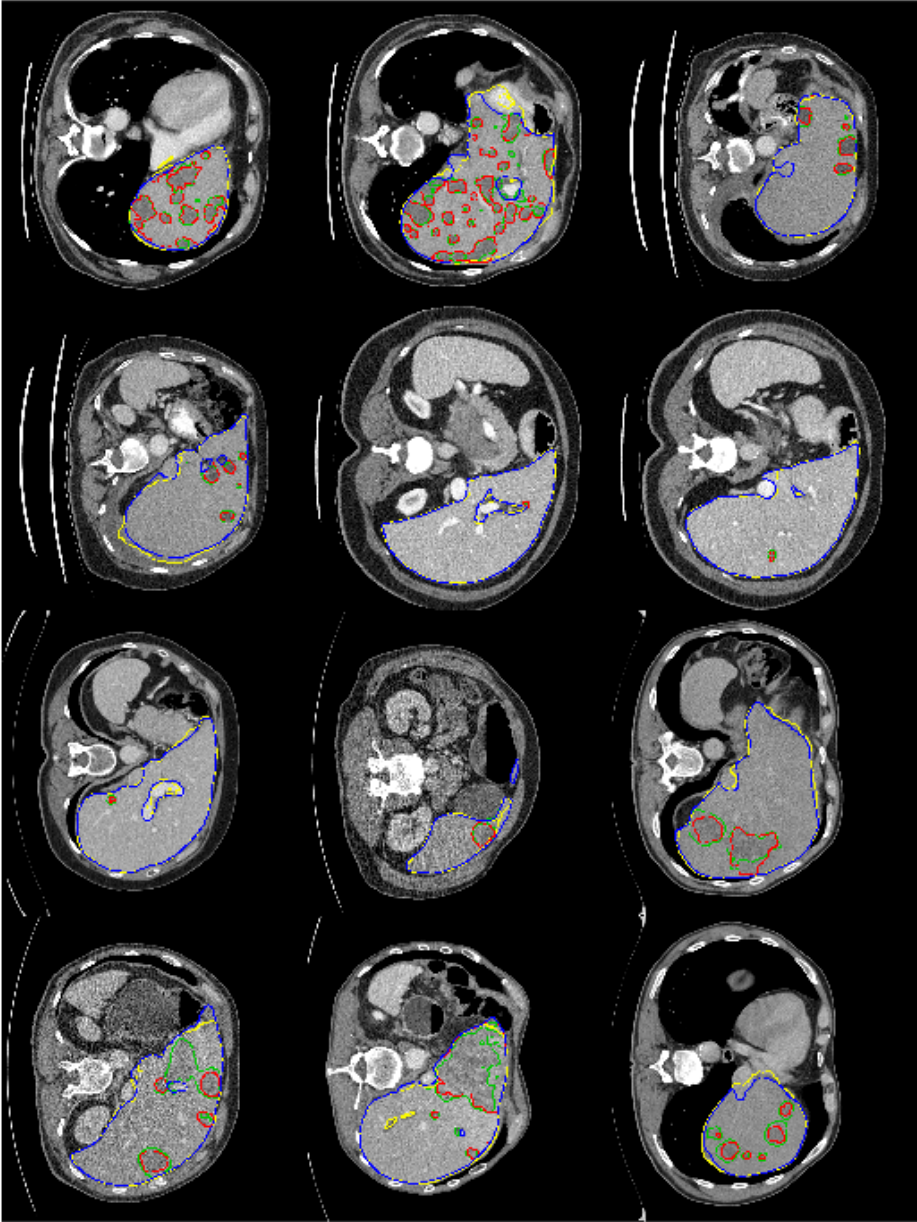


Figure A.6: Examples of last configuration.

Bibliography

- [1] Lei Bi, Jinman Kim, Ashnil Kumar, and Dagan Feng. Automatic liver lesion detection using cascaded deep residual networks. *arXiv preprint arXiv:1704.02703*, 2017.
- [2] Sergi Caelles, Kevis-Kokitsi Maninis, Jordi Pont-Tuset, Laura Leal-Taixé, Daniel Cremers, and Luc Van Gool. One-shot video object segmentation. *arXiv preprint arXiv:1611.05198*, 2016.
- [3] Hao Chen, Xiaojuan Qi, Lequan Yu, and Pheng-Ann Heng. Dcan: Deep contour-aware networks for accurate gland segmentation. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2487–2496, 2016.
- [4] Jianxu Chen, Lin Yang, Yizhe Zhang, Mark Alber, and Danny Z Chen. Combining fully convolutional and recurrent neural networks for 3d biomedical image segmentation. In *Advances in Neural Information Processing Systems*, pages 3036–3044, 2016.
- [5] Grzegorz Chlebus, Hans Meine, Jan Hendrik Moltz, and Andrea Schenk. Neural network-based automatic liver tumor segmentation with random forest-based candidate filtering. *arXiv preprint arXiv:1706.00842*, 2017.
- [6] Patrick Ferdinand Christ, Florian Ettl, Felix Grün, Mohamed Ezzeldin A Elshaera, Jana Lipkova, Sebastian Schlecht, Freba Ahmaddy, Sunil Tatavarty, Marc Bickel, Patrick Bilic, et al. Automatic liver and tumor segmentation of ct and mri volumes using cascaded fully convolutional neural networks. *arXiv preprint arXiv:1702.05970*, 2017.
- [7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009.
- [8] Qi Dou, Hao Chen, Yueming Jin, Lequan Yu, Jing Qin, and Pheng-Ann Heng. 3d deeply supervised network for automatic liver segmentation from ct volumes. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 149–157. Springer, 2016.
- [9] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2650–2658, 2015.
- [10] Krzysztof J Geras, Stacey Wolfson, S Kim, Linda Moy, and Kyunghyun Cho. High-resolution breast cancer screening with multi-view deep convolutional neural networks. *arXiv preprint arXiv:1703.07047*, 2017.

- [11] Xiao Han. Automatic liver lesion segmentation using a deep convolutional neural network method. *arXiv preprint arXiv:1704.07239*, 2017.
- [12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [13] Tobias Heimann, Bram Van Ginneken, Martin A Styner, Yulia Arzhaeva, Volker Aurich, Christian Bauer, Andreas Beck, Christoph Becker, Reinhard Beichel, György Bekes, et al. Comparison and evaluation of methods for liver segmentation from ct datasets. *IEEE transactions on medical imaging*, 28(8):1251–1265, 2009.
- [14] Assaf Hoogi, Christopher F Beaulieu, Guilherme M Cunha, Elhamy Heba, Claude B Sirlin, Sandy Napel, and Daniel L Rubin. Adaptive local window for level set segmentation of ct and mri liver lesions. *Medical image analysis*, 37:46–55, 2017.
- [15] Assaf Hoogi, John W Lambert, Yefeng Zheng, Dorin Comaniciu, and Daniel L Rubin. A fully-automated pipeline for detection and segmentation of liver lesions and pathological lymph nodes. *arXiv preprint arXiv:1703.06418*, 2017.
- [16] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *Advances in neural information processing systems*, pages 109–117, 2011.
- [17] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.
- [18] Matthew Lai. Deep learning for medical image segmentation. *arXiv preprint arXiv:1505.02000*, 2015.
- [19] Kisuk Lee, Aleksandar Zlateski, Vishwanathan Ashwin, and H Sebastian Seung. Recursive training of 2d-3d convolutional networks for neuronal boundary prediction. In *Advances in Neural Information Processing Systems*, pages 3573–3581, 2015.
- [20] Jana Lipková, Markus Rempfler, Patrick Christ, John Lowengrub, and Bjoern H Menze. Automated unsupervised segmentation of liver lesions in ct scans via cahn-hilliard phase separation. *arXiv preprint arXiv:1704.02348*, 2017.
- [21] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015.
- [22] Fang Lu, Fa Wu, Peijun Hu, Zhiyi Peng, and Dexing Kong. Automatic 3d liver location and segmentation via convolutional neural network and graph cut. *International journal of computer assisted radiology and surgery*, 12(2):171–182, 2017.
- [23] Kevis-Kokitsi Maninis, Jordi Pont-Tuset, Pablo Arbeláez, and Luc Van Gool. Convolutional oriented boundaries. In *European Conference on Computer Vision*, pages 580–596. Springer, 2016.
- [24] Kevis-Kokitsi Maninis, Jordi Pont-Tuset, Pablo Arbeláez, and Luc Van Gool. Deep retinal image understanding. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 140–148. Springer, 2016.

- [25] A Mart'in, A Ashish, B Paul, et al. Tensorflow: large-scale machine learning on heterogeneous distributed systems. *Preliminary White Paper*, 2015.
- [26] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *3D Vision (3DV), 2016 Fourth International Conference on*, pages 565–571. IEEE, 2016.
- [27] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 234–241. Springer, 2015.
- [28] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
- [29] Saining Xie and Zhuowen Tu. Holistically-nested edge detection. In *Proceedings of the IEEE international conference on computer vision*, pages 1395–1403, 2015.
- [30] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015.