Constrained Deep Networks for Medical Image Segmentation

by

Hoel KERVADEC

MANUSCRIPT-BASED THESIS PRESENTED TO ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
IN PARTIAL FULFILLMENT FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY
Ph.D.

MONTREAL, JANUARY 29 2021

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

BOARD OF EXAMINERS

THIS THESIS HAS BEEN EVALUATED

BY THE FOLLOWING BOARD OF EXAMINERS

Mr. Ismail Ben Ayed, thesis supervisor
Département de génie des systèmes, ÉTS Montréal, Canada

Mr. Jose Dolz, co-supervisor
Département de génie logiciel et des TI, ÉTS Montréal, Canada

Mr. Eric Granger, co-supervisor
Département de génie des systèmes, ÉTS Montréal, Canada

Mr. Jean-Marc Lina, president of the board of examiners
Département de génie électrique, ÉTS Montréal, Canada

Mr. Matthew Toews, member of the jury
Département de génie des systèmes, ÉTS Montréal, Canada

Mrs. Marleen de Bruijne, external examiner
Erasmus MC, Rotterdam, Netherlands

THIS THESIS WAS PRESENTED AND DEFENDED

IN THE PRESENCE OF A BOARD OF EXAMINERS AND THE PUBLIC

ON DECEMBER 14 2020

AT ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

Réseau profonds contraints appliqué à la segmentation d'imagerie médicale

Hoel KERVADEC

RÉSUMÉ

La segmentation sémantique faiblement supervisée, prenant la forme d'images partiellement annotées, fait l'object d'une grande attention académie, puisqu'elle peut limiter le besoin d'annotations (chère à produire) requises par les modèles de réseaux profonds. Imposer des contraintes globales et non linéaires (sous la forme d'inégalités) aux prédictions d'un réseau de neurone peut ainsi guider l'entrainement vers des solutions atanomiquement possibles, et ainsi permettre d'utiliser des informations *à priori* sur la tâche. Les inégalités sont très flexibles, puisqu'elles ne requiert pas une information précise et parfaite. L'optimisation Lagrangienne standard a très peu été utilisée dans le cadre des réseaux de neurone, principalement à cause du coût de calcul très élevé dû à l'alternance des mises-à-jour explicites entre paramètres et multiplicateurs. Au cours de cette thèse, nous avons testé différents méthodes – pénalités naïves et extension de *log-barrier* plus formelles – afin de contourner les limitations de l'optimisation Lagrangienne. Les deux méthodes ont produit des résultats significativement meilleurs que les quelques méthodes existantes (limitées quant à elles à de simples contraintes linéaires), ainsi qu'un entraineemnt plus stable avec une meilleure convergence. L'extension des *log-barrier*, plus puissante, a permis l'utilisation de fonctions plus complexes, et plus compétitives entre-elles. Nous présentons des expériences robustes et variées, sur une multitude de tâches de segmentation sémantique ; démontrant à la fois l'efficacité de nos méthodes et la pertinence de l'entrainement sous contrainte dans le contexte de l'imagerie médicale. Tout le code produit par cette thèse est disponible en ligne, et peut être réutilisé et modifié librement.

Mots-clés: optimisation sous constraintes, apprentissage profond, imagerie médicale, faible supervision

Constrained Deep Networks for Medical Image Segmentation

Hoel KERVADEC

ABSTRACT

Weakly supervised image segmentation, in the form of partially labeled images, is attracting significant research attention as it can mitigate the need for laborious pixel annotations required by deep learning models. Enforcing high-order, global inequality constraints on the network outputs can leverage unlabeled data by guiding the training with prior knowledge, restricting the search space during training to anatomically feasible solutions. A range of possible values (such as a lower/upper bounds on the size of a organ) can be very valuable to guide training. However, in the context of deep neural networks, standard Lagrangian optimization has been largely avoided, mainly due to the instability and computational complexity ensuing from alternating explicit dual updates and stochastic optimization. Interior point methods, despite their popularity in convex optimization, are not applicable neither, as they require a feasible starting point, which is itself a difficult constrained problem for deep neural networks. In this thesis, we investigate hard inequality constraints in the context of deep networks with both quadratic penalties and more principled *log-barrier* extensions. We also investigate methods to mitigate class-imbalance in segmentation problems, such as in brain lesions dataset, by constraining the boundary of the predicted segmentation to match the ground-truth boundary. This thesis produced five different publications as first author, and four papers as co-author. Our papers received several awards, and we were invited to publish extended versions of our works in two special issues of Medical Image Analysis (MedIA).

In our *first* contribution, we propose to introduce a differentiable penalty, which enforces inequality constraints directly in the loss function, avoiding expensive Lagrangian dual iterates and proposal generation. From constrained-optimization perspective, our simple penalty-based approach is not optimal as there is no guarantee that the constraints are satisfied. However, surprisingly, it yields *substantially* better results than Lagrangian-based constrained convolutional neural networks, while reducing the computational demand for training. By annotating only a small fraction of the pixels, our approach reaches performances comparable to full supervision, on three separate tasks. While our experiments focused on basic linear constraints such as the target-region size and image tags, our framework can be easily extended to other non-linear constraints, e.g., invariant shape moments and other region statistics.

In our *second* contribution, we propose *log-barrier extensions*, which approximate Lagrangian optimization of constrained-CNN problems with a sequence of unconstrained losses. Unlike standard interior-point and log-barrier methods, our formulation does not need an initial feasible solution. We report comprehensive weakly supervised segmentation experiments, with various constraints, showing that our formulation outperforms substantially the existing constrained-CNN methods, both in terms of accuracy, constraint satisfaction and training stability.

In our *third* contribution, we enforce constraints on the boundary of predicted segmentation. Widely used loss functions for CNN segmentation, such as Dice or cross-entropy, are based on

integrals over the segmentation regions. Unfortunately, for highly unbalanced segmentations, such regional summations have values that differ by several orders of magnitude across classes, which affects training performance and stability. We propose a *boundary* loss, which takes the form of a distance metric on the space of contours, not regions. This can mitigate the difficulties of highly unbalanced problems because it uses integrals over the interface between regions instead of unbalanced integrals over the regions. Furthermore, a boundary loss complements regional information. Inspired by graph-based optimization techniques for computing active-contour flows, we express a non-symmetric L_2 distance on the space of contours as a regional integral, which avoids completely local differential computations involving contour points. This yields a boundary loss expressed with the regional softmax probability outputs of the network, which can be easily combined with standard regional losses and implemented with any existing deep network architecture for N-D segmentation. We report comprehensive evaluations on different unbalanced problems, showing that our boundary loss can yield significant increases in performances while improving training stability.

In a *fourth* contribution, we investigates a curriculum-style strategy for semi-supervised CNN segmentation, which devises a regression network to learn image-level information such as the size of the target region. These regressions are used to effectively regularize the segmentation network, constraining the softmax predictions of the unlabeled images to match the inferred label distributions. Our framework is based on inequality constraints, which tolerate uncertainties in the inferred knowledge, e.g., regressed region size. It can be used for a large variety of region attributes. We evaluated our approach for left ventricle segmentation in magnetic resonance images (MRI), and compared it to standard proposal-based semi-supervision strategies. Our method achieves competitive results, leveraging unlabeled data in a more efficient manner and approaching full-supervision performance.

In our *fifth* and last contribution, we propose a novel weakly supervised framework based on several global constraints derived from box annotations. Particularly, we leverage a classical tightness prior to a deep learning setting via imposing a set of constraints on the network outputs. Such a powerful topological prior prevents solutions from excessive shrinking by enforcing any horizontal or vertical line within the bounding box to contain, at least, one pixel of the target region. Furthermore, we integrate our deep tightness prior with a global background emptiness constraint, guiding training with information outside the bounding box. We demonstrate experimentally that such a global constraint is much more powerful than standard cross-entropy for the background class. The ensuing optimization problem is challenging as it takes the form of a large set of inequality constraints on the network outputs. We solve it with a sequence of unconstrained losses based on our log-barrier extensions. This accommodates standard stochastic gradient descent, while avoiding computationally expensive and unstable Lagrangian dual steps and projections. Extensive experiments over two different public data sets and applications (prostate and brain lesions) demonstrate that the synergy between our global tightness and emptiness priors yield competitive performances, approaching full-supervision performances.

All the codes ensuing from this thesis are publicly available, and free to reuse and modify. The functional programming style used makes it easy to integrate new loss functions and constraints, with little-to-no additional coding efforts.

Keywords: constrained optimization, deep learning, medical imaging, weak supervision

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

Page

LIST OF SYMBOLS AND UNITS OF MEASUREMENTS

$\mathbb{R}_+$	Set of real number ≥ 0
$\mathbb{R}_{++}$	Set of real number > 0
$\Omega \subset \mathbb{R}^{2,3}$	Image space
M	Number of modalities (channels) in an image
$\mathcal{K} = \{1, \ldots, K\}$	Discrete set of labels
$\mathcal{N}(\cdot; \boldsymbol{\theta})$	Neural network with parameters $\boldsymbol{\theta}$
$\mathcal{D} = \{(x^n, y^n)\}_{n=1}^N$	Dataset
$x^n : \Omega \rightarrow \mathbb{R}^M$	Input image of sample n
$Y^n : \Omega \rightarrow \mathcal{K}$	Label for sample n
$y^n : \Omega \rightarrow \{0, 1\}^K$	One-hot encoded label for sample n
P	Number of constraints
$\boldsymbol{\lambda}$	Lagrangian dual-multipliers
$\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$	Graph of Random Field (MRF or CRF)
$C_{\mathcal{G}}$	Set of cliques of graph $\mathcal{G}$

LIST OF ABREVIATIONS

wrt.	with respect to
GD	Gradient Descent
SGD	Stochastic Gradient Descent
CNN	Convolutional neural network
DNN	Deep neural network
CRF	Conditional random field
GC	Graph-Cut (not to be confused with Gra**b**-cut)
CT	Computed tomography (scan)
MRI	Magnetic resonance imaging
FCN	Fully convolutional (neural) network
DL	Deep Learning

INTRODUCTION

Computer vision (CV) is an interdisciplinary field that aims to enable computers to "see", not simply retrieving and encoding the signal of a photographic sensor (CMOS), but rather processing and interpreting automatically its content. In other words, computer vision attempts to get a higher-level understanding of the image, and to mimic the capabilities of the human visual system. This branch of artificial intelligence [1] has many real-world applications, such as video-surveillance, autonomous driving, healthcare, industrial processes, image search and retrieval, and so on. While there might be significant overlap, computer vision does not necessarily imply or involve machine learning, although most of the recent literature also belongs to that second category.

Due to the recent advances in deep learning, the past few years have witnessed an unprecedented progress in the performances of computer vision systems, while lowering the barrier of entry for new practitioners. Neural networks are not new LeCun, Bottou, Bengio & Haffner (1998), but their surge in popularity and performance improvements have been enabled by other factors:

- Available computing power, often in the form of Graphical Processing Units (GPUs), reached a tipping point enabling larger and more complex models to be trained in a reduced amount of time.

- The multiplication of large public annotated datasets (e.g. ImageNet Russakovsky, Deng, Su, Krause, Satheesh, Ma, Huang, Karpathy, Khosla, Bernstein et al. (2015), PascalVOC Everingham, Van Gool, Williams, Winn & Zisserman (2010)), have facilitated (pre-)training of deep models. Availability of these public benchmarks also allows a fair comparison between methods.

[1] It was thought in 1966 that it could be done in a single summer project Papert (1966).

If we focus on recognition, there exist three main tasks (from easier to more difficult): image classification, object detection and semantic segmentation—all of them illustrated in Figure 0.1.

- *Image classification (Fig. 0.1).* It consists of predicting a label, or a class, for the whole image: is it a cat, a dog, a boat? Is there a tumor in this medical image, or not? The label can be a simple binary choice (yes/no question), or take a discrete value (from a set of possible values). When several objects of interest are present in the same image (a cat next to a dog, for instance), it becomes difficult to assign a single label to the whole image.

- *Object detection (Fig. 0.1).* It goes beyond image classification by locating roughly (in the form of a bounding box) the object. This enables the prediction of multiple (overlapping) bounding boxes per image, each one with an independent class.

- *Semantic segmentation (Fig.* **??***).* It generalizes object detection, being more fine grained, with the goal of locating the object exactly—it amounts to pixel-wise classification. This gives a complete delineation of each object, as well as exact shape and size attributes, which may play a crucial role to precisely interpret the image content. **This is the application we focus on in this PhD dissertation.**

For image semantic segmentation, decades of research have produced a wealth of methods to tackle this challenging task in multiple scenarios, including image pre-processing, thresholding, graphical models Rother, Kolmogorov & Blake (2004), active shape models Cootes, Taylor, Cooper & Graham (1995), level-sets Boykov & Funka-Lea (2006) or atlases methods Dolz, Desrosiers & Ayed (2017); Koch, Rajchl, Bai, Baumgartner, Tong, Passerat-Palmbach, Aljabar & Rueckert (2018). Although efficient in some specific settings, those classical approaches might be imprecise and/or too slow when dealing with more general and difficult tasks. As in many other fields, deep learning has in the past few years pushed performances to

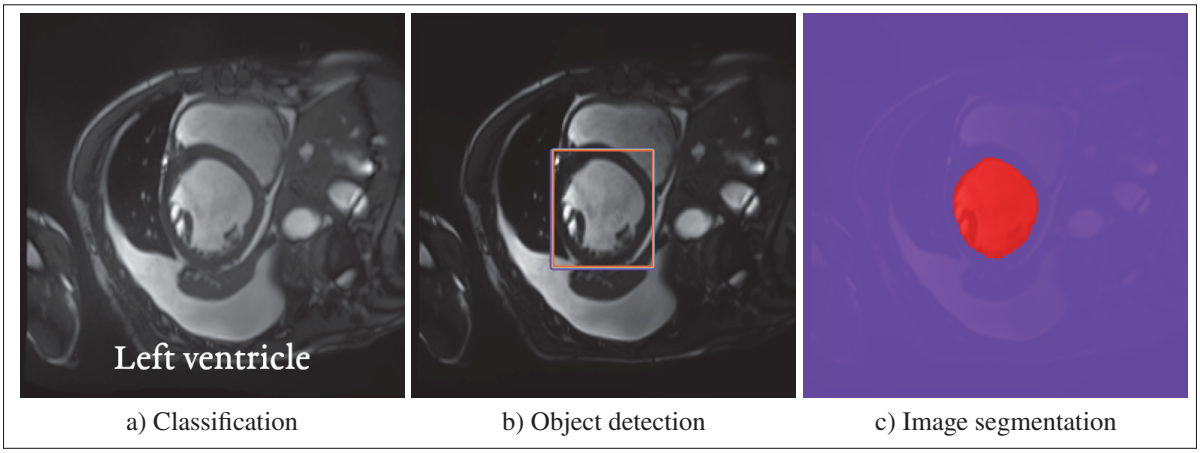


Figure 0.1 Illustration of the difference between the different tasks

new heights[2]. Fully Convolutional Networks (FCNs) are at the core of every state-of-the-art method, relying on now standard architectures such as Chen, Papandreou, Kokkinos, Murphy & Yuille (2015); Long, Shelhamer & Darrell (2015); Ronneberger, Fischer & Brox (2015). Despite some astonishing results of deep learning methods, their main bottleneck remains the need for huge datasets of labeled data—series of examples, with the corresponding inputs and correct answers. Those examples have to be produced and assembled by humans, and the time needed to do so varies between tasks: from a few seconds to annotate an image for classification to several minutes for segmentation Bearman, Russakovsky, Ferrari & Li (2016). More complex scenes, such as high-resolution street views, may take more than one hour to annotate, while annotating a 3D medical scan may take several hours or days. Once a dataset of labeled data is assembled (often referred to as the *training set*), the model parameters are tuned automatically until its predictions match the examples as closely as possible. A change in the training set (be it addition, omission, label noise, or shuffling) will produce a different model, which in turn will have different predictions once put into production.

[2]Natural language processing also had its deep learning revolution, and humans were finally beaten in the game of Go, to a large extent due to deep learning methods.

Figure 0.2 https://xkcd.com/1838/

Medical imaging is the field that seeks to produce images of the human body and make it available to the clinical setting. For the most part of history, medicine has been limited by the (in)ability to see *inside* the human body, which makes harder to diagnose, understand, and treat disease and injuries. A major breakthrough came in the late 19th century, with the discovery of the X-rays and their medical applications by Wilhelm Röntgen—for which he was awarded the first Nobel prize of Physics. Noticing that different parts of the human body absorbed (or did not) different waves-lengths of radiations, it allowed to take a *picture* of the inside, as showed by Figure 0.3.

With decades of refinements and cutting-edge research, contemporary imaging methods include:

- *Computer tomography (CT).* A rotating X-Ray machine performs a series of 2D scans at different angles, followed by reconstructing a 3D model of the body. Because of the higher dose of radiation the patient receives, there is safety limits on the frequencies that can be performed. Naturally, CT-scans can image the same objects as X-rays scans—organs with

a) *Hand mit Ringen*

b) Frontal view of a knee

Figure 0.3 Comparison of the first X-ray taken by Wilhem Röntgen depicting his wife's hand, and a modern X-ray

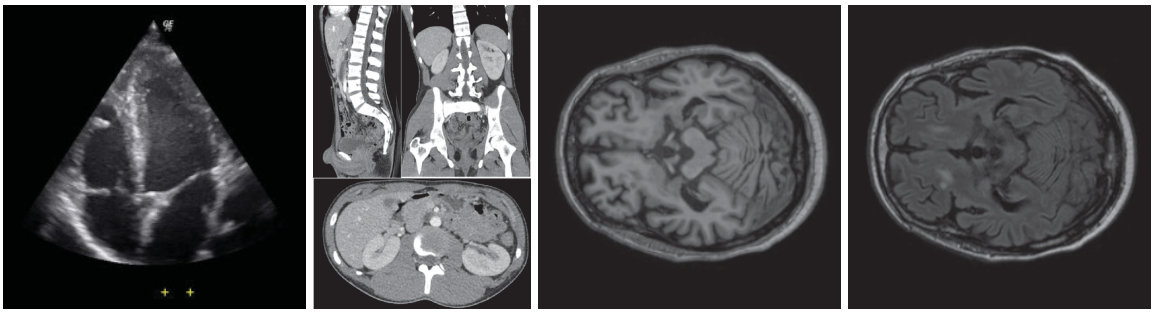
a high-water content, such as the brain, remains virtually invisible with it—limiting its applications.

- *Magnetic Resonance Imaging (MRI).* The basic concept is that a powerful magnetic field will excite, with pulses at the correct resonating frequency, nucleis of specific molecules[3] (most of the time, the H nuclei of the H_2O molecules). By measuring the change of response, and the time it takes for the molecules to relax, one is able to deduce the composition of the scanned area. By changing the targeted molecules, measuring a different signal, or using a contrastive agent, one is able to measure different features/modalities. As illustrated in Fig. 0.4, different modalities (such as T1 and FLAIR) react differently to white and gray matter. Performing a scan remains long, and 3D images (made by stitching a series of 2D scans) are very sensitive to motion, especially at higher resolutions. Considerable research efforts continue to improve speed, precision and patient comfort.

- *Ultrasounds.* As sound waves are (partly) reflected at the accoustic boundary between tissues (e.g., between different organs), it is possible to emit sound waves and then measure

[3] As Arthur C. Clarke famously stated, «Any sufficiently advanced technology is indistinguishable from magic ».

the round-trip time of the reflected waves, making it possible to deduce the tissue layouts[4]. This very portable and real-time imaging method remains noisy, and it is difficult to get sharp images with it.



a) Ultrasound b) CT scan c) MRI (T1 modality) d) MRI (FLAIR modality)

Figure 0.4 Illustration of different modern imaging methods

Selecting the best suited modality will depend on several factors: the information required by the doctors, portability (some patients have limitations to be moved to the room containing the scan), cost and availability of the machines. Safety can also come into play, due to radiation exposure or the presence of non-removable metallic implants.

Notice that, contrary to natural images, medical imaging methods work in an indirect and reactive way. A CMOS sensor will “simply" measure the light-waves emitted and reflected by the object of interest. On the contrary, the methods we previously described first emit waves or particules toward the body and, from the response, deduce an image.

When computer vision meets medical imaging

Nowadays, radiologists spend a considerable amount of their time looking at and annotating medical images (often 3D volumes). Not only time consuming—reducing the experts availabil-

[4]What is interesting is the scalability of this method, as one can perform the same to image the center of the Earth.

ity for patients care or clinical research—those repetitive tasks can be error-prone. Automating parts of their workflow could facilitate their work, and ultimately improve patients' outcome.

Semantic image segmentation, as introduced earlier, is of crucial importance in the medical setting, as it serves the diagnosis, treatment and follow-up of many diseases. For instance, the segmentation of the left-ventricle in a Cine-MRI (MRI scan over time) can be used to compute the ventricle volume over time, helping to diagnos cardiac arythmia. In oncology, it can help target areas to radiate and organs to spare during radiotherapy. A complete segmentation of a scan can help to design custom-made implants. Moreover, an automatic segmentation is easier to interpret/understand and useful for quality control. Those applications and advantages explain the considerable attention that image segmentation receives in the research community, which translates in numerous publications in conferences in the field (Figure 0.5).

This PhD will focus on two major difficulties of image segmentation: annotations cost and data imbalance. While natural image segmentation is similarly affected, those difficulties are much more pronounced in the medical field.

Annotations are expensive to make

As mentioned earlier, training a deep learning model requires to assemble a curated and annotated set of data; this is often the most expensive step of a machine learning pipeline and its main bottleneck. For natural images, annotations can be crowd-sourced, with tools such as ReCaptcha (Fig 0.6), Amazon Mechanical Turk, or other forms of cheap labor. In the medical field, annotations require high expertise, restricting greatly the pool of capable annotators. As most of the time medical images are 3D volumes, they take even longer to annotate. Proper tooling might help, but the expert might still be required to go manually through all 2D slices. As such, for some tasks, it may take up to one week to annotate a single image (Figure 0.7). Additionally, the difficulty is further accrued by the diversity in acquisition settings

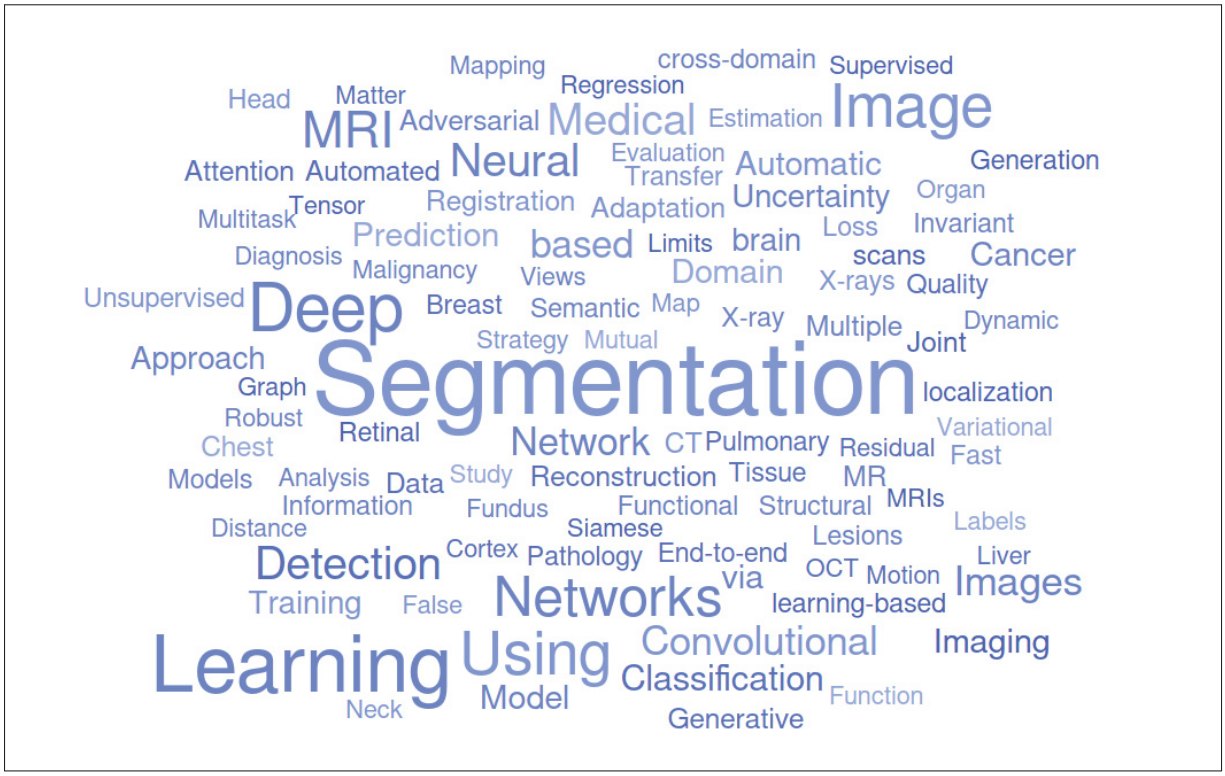
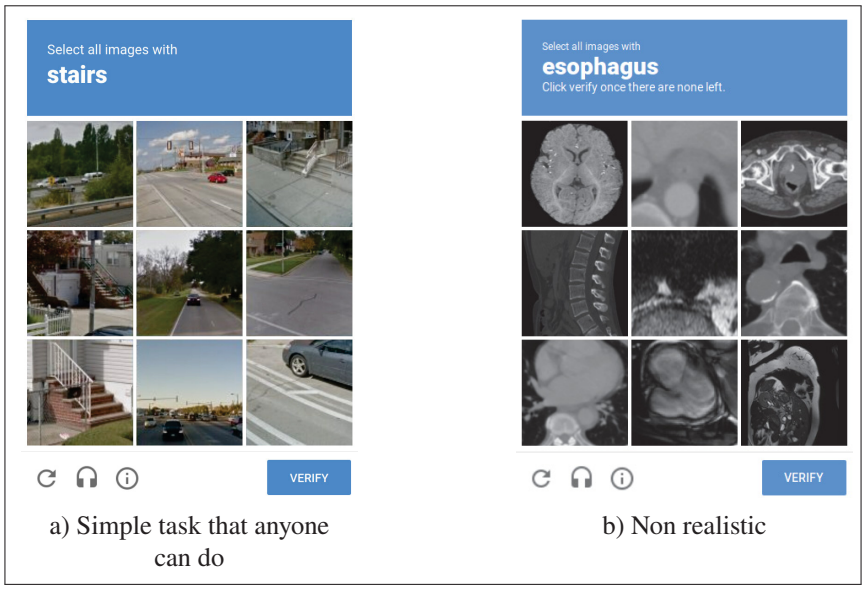


Figure 0.5 Word cloud of the paper titles from the Medical Imaging with Deep Learning (MIDL) 2020 conference

(manufacturer of the MRI, and its settings), which directly affects generalization performances: subtle differences between settings might make a network trained on setting `A` unable to predict correctly on setting `B`. A good dataset must, therefore, cover not only many patients but also several sites, vendors and settings—which adds administrative and regulatory hurdles, as the sharing of medical data is strictly regulated.

Data imbalance

Data imbalance refers to big discrepancy in distribution between classes of a dataset, with one class several orders of magnitude more frequent than another. In medical semantic segmentation, it often happens on brain lesions dataset (but not limited to), where most of the brain is healthy, as showed in Figure 0.8. If training methods are not modified to mitigate this imbalance, the



a) Simple task that anyone can do

b) Non realistic

Figure 0.6 The second case is not realistic to ask to humans

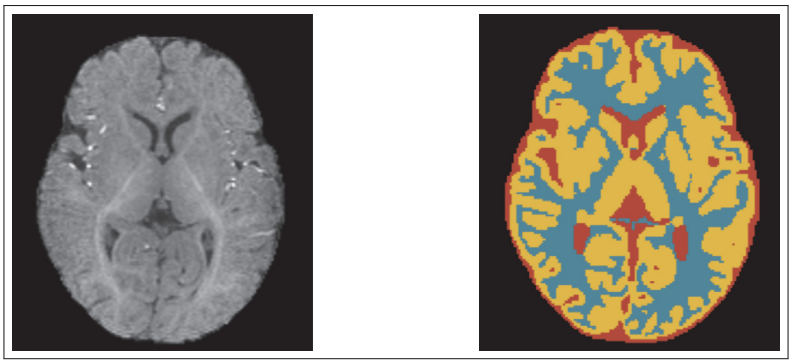


Figure 0.7 Annotating such a scan can take up to one week for a medical doctor

resulting predictions will over-predict the majority class and completely skip the rest—when the minority class (the lesions) is the most important one to detect.

Motivations and objectives

While in the natural world, for instance, it is easy to know a car size and shape in great details from its blueprints—which *could* guide a segmentation algorithm—it is difficult to use that

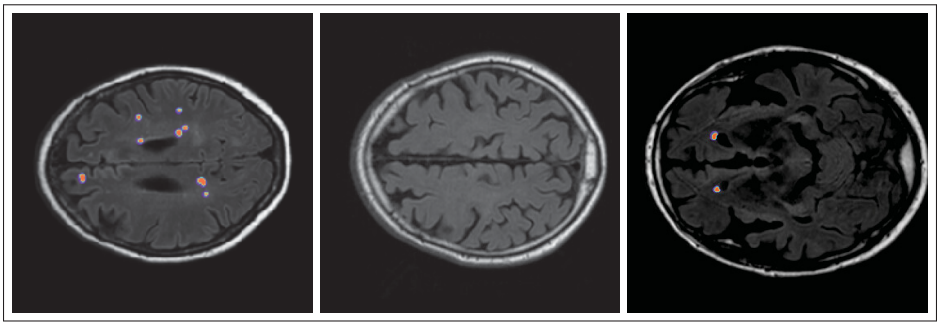


Figure 0.8 Example of a very imbalanced dataset, the White Matter Hyperintensities (WMH) MICCAI 2017 challenge. Brain lesions make up for only 0.05% of the total number voxels, with many slices without any lesion

information in practice. The variety of points of view (orientation, distance) and potential occlusion creates a lot of variance in the way the object appears in a 2D image. In medical imaging, acquisition parameters are controlled and consistent across scans: the point of view, distance and orientation of the patient are all available information—making it easier to translate textbook knowledge about an organ (approximate size, shape, location, ...) to a 3D scan. The motivation to use prior knowledge directly into the training is strong: Why re-learn (through expensive, annotated data) our expensive text-book? Radiological text reports, another existing source of prior information, could also be used. Managing to embed those priors at training would reduce the required amount of newly annotated data, although it is not clear how to achieve it in the context of deep learning.

Our approach was to formulate the training of a deep network with priors *as a constrained optimization problem*: the usual error minimization between labels and predictions remains unchanged, but inequality constraints restrict the search space to only anatomically feasible solutions. While elegant theoretically, it is actually very difficult to solve constrained optimization problems when dealing with deep neural networks. As we will show in Chapter 1, despite the extensive literature on how to handle constraints in classical convex optimization, deep learning

brings new difficulties that are not easily manageable, for computational and memory reasons[5]. Novel methods need to be developed with deep learning in mind to be able to effectively use our prior information.

We can regroup the contributions of this dissertation in two parts, each composed of several articles published in different conferences and journals. While they can be read independently, their research and development was intertwined: interesting constraints (such as the ones in Chapter 6) became apparent only once previous works proved that inequality constraints were a useful and effective way to embed prior information. Conversely, such complex constraints were needed to benchmark and push different optimization methods to their limits (as in Chapter 3).

Thesis outline

Background

We start with a dedicated chapter to introduce useful notions for the understanding of this thesis and its context. Optimization basics and notations are layed down, with an emphasis on constrained optimization and standard Lagrangian dual methods. We connect this optimization framework to deep learning and standard stochastic gradient descent, highlighting the optimization difficulties of deep learning and explaining the lack of theoretical guarantees on convergence and optimality. While we do not cover neural network architectures in great details, we will introduce some standard training losses and discuss their effects. A connection to classical Random Fields methods—and how they can be used as post-processing in the deep learning era—is made. We then present the most common forms of weak labels and supervision. More specifically, we describe the major differences between *proposal* based methods (which attempt to mimic full supervision) and *direct-loss* methods (which embrace the weak nature of

[5]As for several areas, it would not be an issue had we infinite time to train our networks.

the labels). We argue that direct losses, while more difficult to formulate and adapt, are more suited for weak labels. At last, we discuss the few related methods for constrained optimization in the context of deep neural networks.

First part: Constraining deep neural networks

Constrained-CNN Losses for Weakly Supervised Segmentation

H. Kervadec, J. Dolz, M Tang, E. Granger, Y Boykov, I. Ben Ayed. MIDL 2018 (Selected for oral presentation), journal extension in MEDIA, volume 54, 2019.

Weakly-supervised learning based on partially labelled images or image-tags is currently attracting significant attention in CNN segmentation, as it can mitigate the need for full and laborious pixel/voxel annotations. Enforcing high-order (global) inequality constraints on the network output (for instance, to constrain the size of the target region) can leverage unlabeled data, guiding the training process with domain-specific knowledge. Inequality constraints are very flexible because they do not assume exact prior knowledge. We propose to introduce a differentiable penalty, which enforces inequality constraints directly in the loss function, avoiding expensive Lagrangian dual iterates and proposal generation. From constrained-optimization perspective, our simple penalty-based approach is not optimal as there is no guarantee that the constraints are satisfied. However, surprisingly, it yields *substantially* better results than the Lagrangian-based constrained CNNs in Pathak *et al.* (2015a), while reducing the computational demand for training. By annotating only a small fraction of the pixels, the proposed approach can reach performances comparable to full supervision, on three separate tasks. While our experiments focused on basic linear constraints such as the target-region size and image tags, our framework can be easily extended to other non-linear constraints, e.g., invariant shape moments Klodt & Cremers (2011) or other region statistics Lim, Jung & Kohli (2014).

Constrained Deep Networks: Lagrangian Optimization via Log-Barrier Extensions

H. Kervadec, J. Dolz, J. Yuan, C. Desrosiers, E. Granger, I. Ben Ayed. Pre-print.

This study investigates the optimization aspects of imposing hard inequality constraints on the outputs of CNNs. In the context of deep networks, constraints are commonly handled with *penalties* for their simplicity, despite their well-known limitations. Lagrangian-dual optimization has been largely avoided, except for a few recent works, mainly due to the computational complexity and stability/convergence issues caused by alternating *explicit* dual updates/projections and stochastic optimization. Several studies showed that, surprisingly for deep CNNs, the theoretical and practical advantages of Lagrangian optimization over penalties do not materialize in practice. We propose a *log-barrier extensions*, which approximate Lagrangian optimization of constrained-CNN problems with a sequence of unconstrained losses. Unlike standard interior-point and log-barrier methods, our formulation does not need an initial feasible solution. We report comprehensive weakly supervised segmentation experiments, with various constraints, showing that our formulation outperforms substantially the existing constrained-CNN methods, both in terms of accuracy, constraint satisfaction and training stability.

Boundary loss for highly unbalanced segmentation

H. Kervadec, J. Bouchtiba, C. Desrosiers, E. Granger, J. Dolz, I. Ben Ayed. MIDL 2019 (Runner-up for best-paper award), journal extension in MEDIA, volume 67, 2020.

Widely used loss functions for CNN segmentation, such as Dice or cross-entropy, are based on integrals over the segmentation regions. Unfortunately, for highly unbalanced segmentations, such regional summations have values that differ by several orders of magnitude across classes, which affects training performance and stability. We propose a *boundary* loss, which takes the form of a distance metric on the space of contours, not regions. This can mitigate the difficulties

of highly unbalanced problems because it uses integrals over the interface between regions instead of unbalanced integrals over the regions. Furthermore, a boundary loss complements regional information. Inspired by graph-based optimization techniques for computing active-contour flows, we express a non-symmetric L_2 distance on the space of contours as a regional integral, which avoids completely local differential computations involving contour points. This yields a boundary loss expressed with the regional softmax probability outputs of the network, which can be easily combined with standard regional losses and implemented with any existing deep network architecture for N-D segmentation. We report comprehensive evaluations and comparisons on different unbalanced problems, showing that our boundary loss can yield significant increases in performances while improving training stability.

Second part: Constraints for medical image segmentation

Curriculum semi-supervised segmentation

H. Kervadec, J. Dolz, E. Granger, I. Ben Ayed. MICCAI 2019.

This study investigates a curriculum-style strategy for semi-supervised CNN segmentation, which devises a regression network to learn image-level information such as the size of the target region. These regressions are used to effectively regularize the segmentation network, constraining the softmax predictions of the unlabeled images to match the inferred label distributions. Our framework is based on inequality constraints, which tolerate uncertainties in the inferred knowledge, e.g., regressed region size. It can be used for a large variety of region attributes. We evaluated our approach for left ventricle segmentation in magnetic resonance images (MRI), and compared it to standard proposal-based semi-supervision strategies. Our method achieves competitive results, leveraging unlabeled data in a more efficient manner and approaching full-supervision performance.

Bounding boxes for weakly supervised segmentation: Global constraints get close to full supervision

H. Kervadec, J. Dolz, S. Wang, E. Granger, I. Ben Ayed. MIDL 2020 (Selected for oral presentation).

We propose a novel weakly supervised learning segmentation based on several global constraints derived from box annotations. Particularly, we leverage a classical tightness prior to a deep learning setting via imposing a set of constraints on the network outputs. Such a powerful topological prior prevents solutions from excessive shrinking by enforcing any horizontal or vertical line within the bounding box to contain, at least, one pixel of the foreground region. Furthermore, we integrate our deep tightness prior with a global background emptiness constraint, guiding training with information outside the bounding box. We demonstrate experimentally that such a global constraint is much more powerful than standard cross-entropy for the background class. The resulting optimization problem is challenging as it takes the form of a large set of inequality constraints on the outputs of deep networks. We solve it with a sequence of unconstrained losses based on a recent powerful extension of the log-barrier method, which is well-known in the context of interior-point methods. This accommodates standard stochastic gradient descent (SGD) for training deep networks, while avoiding computationally expensive and unstable Lagrangian dual steps and projections. Extensive experiments over two different public data sets and applications (prostate and brain lesions) demonstrate that the synergy between our global tightness and emptiness priors yield very competitive performances, approaching full supervision and outperforming significantly DeepCut. Furthermore, our approach removes the need for computationally expensive proposal generation.

Code and open-source

The code of all papers is available, free to reuse/modify. While split in different repositories, all code stem from the same (private) codebase, that expanded over the years of this PhD.

- **Constrained-CNN Losses for Weakly Supervised Segmentation** https://github.com/LIVIAETS/SizeLoss_WSS
- **Constrained Deep Networks: Lagrangian Optimization via Log-Barrier Extensions** https://github.com/LIVIAETS/extended_logbarrier
- **Boundary loss for highly unbalanced segmentation** https://github.com/LIVIAETS/surface-loss
- **Curriculum semi-supervised segmentation** https://github.com/LIVIAETS/semi_curriculum
- **Bounding boxes for weakly supervised segmentation: Global constraints get close to full supervision** https://github.com/LIVIAETS/boxes_tightness_prior

Co-authored publications

In addition to the aforementioned first-author publications, this thesis has also led to the following co-authored publications.

- **Constrained domain adaptation for segmentation** *M. Bateson, J. Dolz, H. Kervadec, H. Lombaert, I. Ben Ayed.* MICCAI 2019.
- **Source-Relaxed Domain Adaptation for Image Segmentation** *M. Bateson, H. Kervadec, J. Dolz, H. Lombaert, I. Ben Ayed.* MICCAI 2020.

- **Discretely-constrained deep network for weakly supervised segmentation** *J. Peng, H. Kervadec, J. Dolz, I. Ben Ayed, M. Pedersoli, C. Desrosiers.* Neural Networks, volume 130, 2020.

- **Laplacian pyramid-based complex neural network learning for fast MR imaging** *H. Liang, Y. Gong, H. Kervadec, J. Yuan, H. Zheng, S. Wang.* MIDL 2020.

CHAPTER 1

BACKGROUND

1.1 Optimization - Background and notations

1.1.1 Unconstrained optimization

Optimization , in very broad terms, consists of finding the *ideal* value (either a maximum or minimum) of a function $f_0 : \mathbf{dom} f \subseteq \mathbb{R}^D \rightarrow \mathbb{R}$ with respect to its input $x \mapsto f_0(x)$. It has widespread real-world applications: many decision making or system design problems can be formulated as an optimization problem. A minimization problem is denoted as[1]:

$$\min_{x \in \mathbf{dom} f} f_0(x).$$

An optimal solution $p^* := f_0(x^*)$ will verify:

$$\forall y \in \mathbf{dom} f : p^* \leq f_0(y),$$

while an ϵ sub-optimal solution $\tilde{x}$ will verify:

$$f_0(\tilde{x}) \leq p^* + \epsilon.$$

Apart form trivial functions, finding the optimal p^* (or the corresponding optimal input x^*) is usually very difficult, and cannot be solved analytically. Trying exhaustively all different $x \in \mathbf{dom} f$ is not feasible, more so when dealing with continuous domains and/or high dimensions. As finding a *global* solution x^* might not be feasible, one can instead settle for a *local* approximation $\tilde{x}$: a solution that is minimal in its own local neighborhood. A very crude approach consists of starting with an initial guess x^0, and then refining it with an existing

[1] A maximization problem can simply be transformed into a minimization problem by putting a minus sign $(-)$ in front of f_0.

algorithm—for instance a gradient descent[2]. As the gradient $\nabla f(x)$ is the *slope* of the function at that point, a simple method is to follow it: go up if we maximize f_0, go down if we minimize it. The procedure is described in more details in Algorithm 9. When f_0 is convex[3] we have the guarantee that $\hat{x} = x^*$: a *global* optimum—see Figure **??**. If not (as in Fig. **??**), the slope pushes back toward $\hat{x}$; it is *stuck* in that local minima and more complex optimization methods are needed.

Algorithm 1.1 Overview of the gradient descent algorithm.

Input: *Given step update γ, Given stopping criterion η*
Output: *Current solution $\hat{x} := x^t$*
Init x^0 to some value in $\mathbf{dom} f$,
Init $t \leftarrow 0$
while *η is not met* **do**
    $\Delta x \leftarrow \nabla f_0(x^t)$
    $x^{t+1} \leftarrow x^t - \gamma \Delta x$
    $t \leftarrow t + 1$
end

Optimization methods for convex problems are quite mature and robust. Convex problems are, typically, quite straightforward to optimize. However, as highlighted by Boyd Boyd & Vandenberghe (2004), optimizing a non-convex problem is more akin to *art* than technology; there is no standard method that works for everything. On the contrary, for convex problems, the art resides in the difficulty to identify or reformulate the problem as convex. Since there exist much more numerically efficient algorithms for convex problems, a decent strategy for some non-convex problems might be to minimize a convex upper bound, and then refine locally the convex upper bound.

[2]There is a whole subset of optimization research that focused on gradient-free optimization, but this is both out of scope, and not applicable to deep neural networks. Here, we assume that f_0 is derivable (at least once) over its domain.

[3]A function is *convex* when, $\forall x, y \in \mathbf{dom} f, \alpha + \beta = 1 : f_0(\alpha x + \beta y) \leq \alpha f_0(x) + \beta f_0(y)$. In other words, the line (chord) between $f_0(x)$ and $f_0(y)$ is always above f_0.

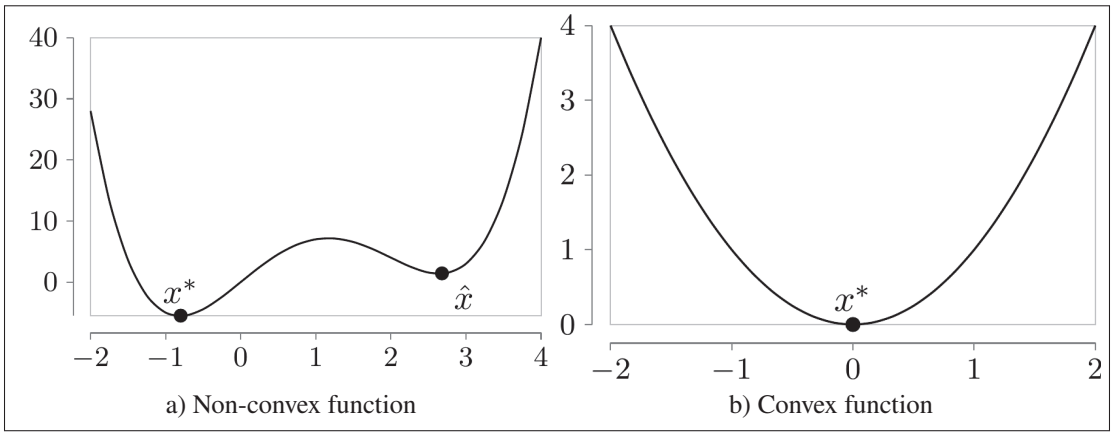


Figure 1.1 Depending on the starting point, a gradient descent will find a different local minima for non-convex functions.

1.1.2 Constrained optimization: Lagrangian and duality

In some situations, we want not only to minimize f_0, but also to enforce some conditions on the solution; those are constraints on the optimization process, which the solution should satisfy. Formally, constraints can be written as follows[4]:

$$\begin{aligned} \min_x \quad & f_0(x) \\ \text{subject to} \quad & f_1(x) \leq 0 \\ & \dots \\ & f_P(x) \leq 0. \end{aligned} \tag{1.1}$$

A basic algorithm for constrained optimization comes from Joseph-Louis Lagrange (1736-1813), who introduced the Lagrangian-dual problem. A simplistic way to reformulate Equation (1.1) into a unconstrained optimization problem would be to use a infinite penalty function when the constraints are not satisfied:

$$\min_x f_0(x) + \sum_{i=1}^{P} \infty_{[f_i(x)>0]}, \tag{1.2}$$

[4]An equality constraint $f_n(x) = 0$ can be written with two inequality constraints: $f_n(x) \leq 0$ and $f_n(x) \geq 0$.

where $\infty_{[a]}$ takes the value 0 when axioms a is `False`, and the value $+\infty$ when a is `True`. It will not come as a surprise that such a discontinuous function is horrible to optimize. But first, we can notice that:

$$\forall i : \infty_{[f_i(x)>0]} = \max_{\boldsymbol{\lambda}\geq 0} \lambda_i f_i(x).$$

When maximizing over $\boldsymbol{\lambda} \geq 0$ for some $f_i(x)$, the optimal solution is 0 when $f_i(x) < 0$. When $f_i(x)$ is positive, the optimal λ value is $+\infty$. We can plug this reformulation back into our poorly conditioned minimization problem (1.2):

$$\min_x \max_{\boldsymbol{\lambda}\geq 0} \quad f_0(x) + \sum_{i=1}^{P} \lambda_i f_i(x). \tag{1.3}$$

This problem is still difficult to optimize, but less so if we swap the minimization and maximization:

$$\max_{\boldsymbol{\lambda}\geq 0} \min_x \quad f_0(x) + \sum_{i=1}^{P} \lambda_i f_i(x). \tag{1.4}$$

While easier to solve, it does not have the same optimum as the original Eq (1.3) (more on that shortly). For a fixed $\boldsymbol{\lambda}$, we can optimize x, this is the Lagrangian dual function:

$$\mathcal{L}(\boldsymbol{\lambda}) = \min_x f_0(x) + \sum_{i=1}^{P} \lambda_i f_i(x). \tag{1.5}$$

We can easily show that $\mathcal{L}(\boldsymbol{\lambda}) \leq p^*$. Indeed, for a feasible solution $\tilde{x}$, $\forall i \in \{1, ..., P\} : \lambda_i f_i(\tilde{x}) \leq 0$. Therefore,

$$L(\boldsymbol{\lambda}) \leq f_0(\tilde{x}) + \sum_{i=1}^{P} \lambda_i f_i(\tilde{x}) \leq f_0(\tilde{x}) \leq f_0(x^*) = p^*.$$

The non-negative difference $p^* - \mathcal{L}(\boldsymbol{\lambda})$ is called the *duality gap*. By alternating optimization with respect to $\boldsymbol{\lambda}$ and x, we decrease the duality gap, and eventually reach a gap of zero—if the *Karush-Kuhn-Tucker* (KKT) conditions are met Boyd & Vandenberghe (2004). If so, strong-duality holds and $\hat{x} = x^*$. This is generally not the case, and not always true even for convex settings.

1.2 Deep neural networks

1.2.1 High-level overview

Deep neural networks (and their ancestors the perceptron and multi-layer perceptron) are originally inspired from a very simplified model of biological neurons. At their core, neural networks are parametric functions, performing matrix multiplications between their inputs and their parameters—their *weights*. It is those weights that we want to optimize[5]. Modern networks are usually regrouped in *layers*, which are composed together. To increase expressiveness and ability for the network to model more complex functions, non-linearity are added on top of the core dot products. As such, a high-level description[6] of a neural network $\mathcal{N}$ can be:

$$\mathcal{N}(x;\boldsymbol{\theta}) := l_p \circ ... \circ l_0(x), \tag{1.6}$$

where each $l_i(x)$ involves different operations and weights. The overall structure of a single layer is often in the form:

$$w_i, b_i := \boldsymbol{\theta}_i$$
$$l_i\left(x;(w_i,b_i)\right) := g(xw_i + b_i),$$

where g is a non-linear, derivable function. Some layers might be much simpler, whose purpose is to reduce the dimensionality of the data (by averaging, max-pooling, or other methods).

In the context of computer vision, the convolution operation—inspired from signal processing, where the same operation is performed on a subset of the input, in a sliding window fashion—is at the core of many architectures. Convolutional Neural networks proved to be very effective in a breadth of difficult computer vision tasks. Defining the network architecture (the final function

[5]We often read in the (scientific) literature that neural networks are *trained*. This terminology is actually very close to what Alan Turing describes in his seminal paper Turing (1950) on the imitation game, where he discusses the idea of creating an artificial kid and teaching it to be adult. While being a very interesting read, it remains a though experiment. Using the word *training* for an optimization problem might be perceived as *anthropomorphism*, and not necessarily very scientific.

[6]For the sake of simplicity here, we do not model skip connections, without any loss of generality.

composition) is, however, not enough, as the weights $\boldsymbol{\theta}$ need to be tuned for the network to perform well. This cannot be done by hand: the weights are in a very high dimensionional space, and the network function is quite impervious to mathematical analysis. The current preferred method consist of first initializing the parameters randomly and then tune them over a training set via a descent-based optimization scheme.

Let $\mathcal{D} = \{(x^n, Y^n)\}_{n=1}^N$ be a set of `(input, label)` pairs, where the input is fed to the network and label is the desired output of the network. A loss function is designed in such a way that it is minimized when the network predictions match perfectly the labels. This loss is then optimized with respect to the network parameters $\boldsymbol{\theta}$:

$$\arg\min_{\boldsymbol{\theta}} \sum_{(x^n, Y^n) \in \mathcal{D}} \mathcal{L}(\mathcal{N}(x^n; \boldsymbol{\theta}), y^n).$$

Once *"trained"* (i.e., when the optimization procedure cannot find a better solution), we can use those parameters for inference/deployment.

The resulting optimization problem is highly non-convex and very difficult to optimize. Using a standard gradient descent might work decently in theory, but the resulting optimization problem is still beyond computing capabilities of modern hardware:

$$\min_{\boldsymbol{\theta}} \sum_{(x^n, y^n) \in \mathcal{D}} \mathcal{L}(\mathcal{N}(x^n; \boldsymbol{\theta}), Y^n).$$

Performing the updates of Alg. 9 would require to store all the gradients for each data point *at the same time*, which would exhaust the memory of most computers. Instead, a slight modification of the algorithm perform a similar update, but on a different random subset of the dataset (a batch) at each iteration. This is the *Stochastic* Gradient Descent (SGD): sub-batches $\mathcal{B} \subset \mathcal{D}$ are sampled, and we do one update with respect to that batch. The algorithm is succinctly described in Algorithm 11.

For classification tasks, Y take a value among a set of discrete labels $\mathcal{K} = \{1, ..., K\}$, and the network is designed in such a way that $\mathcal{N}(\cdot; \boldsymbol{\theta}) \in \mathbb{R}^K$. The class predicted by the network is the

Algorithm 1.2 Overview of the **stochastic** gradient descent algorithm.

Input: *Given step update* γ*, Given batch size B, Given stopping criterion* η *(convergence, or quality of the result), Given distribution* Π*, Given uniform distribution U*
Output: *Current solution* $\hat{\theta} := \theta^t$
Init $\theta \sim \Pi$
Init $t := 0$
while η *is not met* **do**
    Sample $\mathcal{B} \sim U(0, N)^B$
    $L = \sum_{b \in \mathcal{B}} \mathcal{L}(\mathcal{N}(x^b; \theta), Y^b)$
    $\Delta\theta = \nabla L$
    $\theta^{t+1} := \theta^t - \gamma\Delta\theta$
    $t = t + 1$
end

index of the output vector with the highest value:

$$\hat{Y}^n = \arg\max_{k \in \mathcal{K}} \mathcal{N}(x^n; \boldsymbol{\theta})_k .$$

This works at inference. However, during training, the arg max function is not derivable, which makes it incompatible with gradient descent. Instead of having discrete network outputs, during training, we use continuous probabilities: the network outputs are vectors in $\mathbb{R}^K$ and within the probability simplex (all the values of a simplex vector are between 0 and 1, and sum to 1). It is easy to obtain a vector of probabilities from the raw network outputs, with the popular softmax function:

$$s^n_{\boldsymbol{\theta}} := \frac{1}{Z} \mathrm{e}^{\mathcal{N}(x^n;\boldsymbol{\theta})^k},$$

where $Z = \sum_{k' \in \mathcal{K}} \mathrm{e}^{\mathcal{N}(x^n;\boldsymbol{\theta})^{k'}}$ is a normalizing constant. The final result $s^n_{\boldsymbol{\theta}} : \Omega \to [0, 1]^K$ is a vector of continuous probabilities, within the simplex ($\sum_{k=1}^K s^n_{\boldsymbol{\theta}}(k) = 1$), but not necessarily on its vertices. An exact solution will predict a probability of 1 for the class Y, and 0 for all others.

The labels Y can similarly be re-encoded as a K length one-hot vector, such as:

$$y^n(k) = \begin{cases} 1 & \text{if } k = Y^n \\ 0 & \text{otherwise.} \end{cases}$$

While not required explicitly, representing the label in this way will make the operations much simpler to define. To summarize, $y^n : \Omega \to \{0,1\}^K$ is the one-hot encoded label, and $s^n_{\boldsymbol{\theta}} : \Omega \to [0,1]^K$ is the softmax output of the network, both vectors summing to 1.

1.2.2 Neural networks for image segmentation

Image semantic segmentation is in essence pixel-wise segmentation. Let us first define $\Omega \subset \mathbb{R}^{2,3}$ the image spacial domain of our dataset $\mathcal{D} = \{(x^n, y^n)\}_{n=1}^N$, x^n being the input images and y^n their corresponding one-hot encoded ground truth. For semantic segmentation, the network architecture is designed in such a way that its output matches the dimension of the inputs:

$$\begin{aligned} x^n &: \Omega \to \mathbb{R}^M \\ y^n &: \Omega \to \{0,1\}^K \\ s^n_{\boldsymbol{\theta}} &: \Omega \to [0,1]^K, \end{aligned}$$

where M represent the number of modalities[7] of the input.

This thesis does not focus on networks architectures, and all formulations presented are architecture-agnostic. For readability reasons, we will simply denote onwards $s^n_{\boldsymbol{\theta}}$ for the softmax predictions, without referring to $\mathcal{N}(\cdot;\boldsymbol{\theta})$ each time. We will describe losses dedicated for semantic segmentation shortly, in Section 1.2.3. Once trained, the predicted segmentation

[7]Often called *channels* for natural images—3 in the case of RGB images, 1 for grayscale.

can be drawn (as in the case of classification) with the arg max function:

$$\hat{Y}^n := [\hat{Y}^n(p) \; \forall p \in \Omega] \tag{1.7}$$

$$\hat{Y}^n(p) := \arg\max_{k \in \mathcal{K}} s^n_{\boldsymbol{\theta}}(p, k) \tag{1.8}$$

1.2.3 Training losses for segmentation

In the previous section, we discussed how the SGD algorithm will minimize a loss. There is several standard losses used in image segmentation. As all of them are averaged over the current batch b, for readibility reasons we will denote in this section $s^n_{\boldsymbol{\theta}}(p)_k$ as $s^{p,k}_{\boldsymbol{\theta}}$ and $y^n(p)_k$ as $y^{p,k}$.

1.2.3.1 Common losses

L2 Loss is one of the simplest choices, where one minimizes the L2 norm between the one-hot vector encoding the ground-truth and the network-predicted probability vector:

$$\mathcal{L}_{\text{L2}}(s_{\boldsymbol{\theta}}, y) = \sum_{k=1}^{K} \sum_{p \in \Omega} |s^{p,k}_{\boldsymbol{\theta}} - y^{p,k}|^2$$

Cross-entropy loss takes the following form, and could be viewed as the KL divergence between the label distribution and the predicted distribution:

$$\mathcal{L}_{\text{CE}}(s_{\boldsymbol{\theta}}, y) = -\sum_{k} \sum_{p \in \Omega} y^{p,k} \log(s^{p,k}_{\boldsymbol{\theta}})$$

It reaches its minimum at 0, when $y^{p,k}$ matches $s^{p,k}_{\boldsymbol{\theta}}$

Dice loss is a modification of the the common DSC index, used to measure the overlap between two segmentations (usually the ground-truth segmentation and the predicted one). The

formulation is relaxed to use the predicted continuous probabilities $s_{\boldsymbol{\theta}}$ instead of binary labels:

$$\mathcal{L}_{DSC}(s_{\boldsymbol{\theta}}, y) = \sum_{k=1}^{K} -\frac{2\sum_{p\in\Omega} s_{\boldsymbol{\theta}}^{p,k} y^{p,k}}{\sum_{p\in\Omega} s_{\boldsymbol{\theta}}^{p,k} + \sum_{p\in\Omega} y^{p,k}}.$$

As we want to maximize the DSC in a minimization setting, we simply add a minus sign in front of the formula.

Notice that $\mathcal{L}_{L2}$ and $\mathcal{L}_{CE}$ treat semantic segmentation as a purely independent pixel-wise classification problem—the formulation is exactly the same as in other settings. $\mathcal{L}_{DSC}$ is slightly different in that aspect, as the loss takes into account the predictions over the whole image.

1.2.3.2 Losses gradients

As we perform a gradient descent on those losses to train our neural network, it is interesting to compare the range of values that the gradients (wrt. the softmax probabilities $s_{\boldsymbol{\theta}}$) can take—as it can influence the training and behavior in major ways:

$$\frac{\partial \mathcal{L}_{L2}}{\partial s_{\boldsymbol{\theta}}^{p,k}} = 2(s_{\boldsymbol{\theta}}^{p,k} - y^{p,k}) \tag{1.9}$$

$$\frac{\partial \mathcal{L}_{CE}}{\partial s_{\boldsymbol{\theta}}^{p,k}} = -\frac{y^{p,k}}{s_{\boldsymbol{\theta}}^{p,k}} \tag{1.10}$$

$$\frac{\partial \mathcal{L}_{DSC}}{\partial s_{\boldsymbol{\theta}}^{p,k}} = -\frac{2(y^{p,k}\mathrm{U}^k - \mathrm{I}^k)}{{\mathrm{U}^k}^2}, \tag{1.11}$$

where $\mathrm{I}^k = \sum_{p\in\Omega} s_{\boldsymbol{\theta}}^{p,k} y^{p,k}$ and $\mathrm{U}^k = \sum_{p\in\Omega} s_{\boldsymbol{\theta}}^{p,k} + \sum_{p\in\Omega} y^{p,k}$, corresponding to the intersection and union of the two segmentations, respectively. The gradients for some ground truth and softmax predictions are plotted in Figure 1.2.

We can easily see that $-2 \leq \frac{\partial \mathcal{L}_{L2}}{\partial s_{\boldsymbol{\theta}}^{p,k}} \leq 2$ and $-\infty \leq \frac{\partial \mathcal{L}_{CE}}{\partial s_{\boldsymbol{\theta}}^{p,k}} \leq 0$. The ranges of values are different from one loss to the other, and it is interesting to see how each loss will push a probability

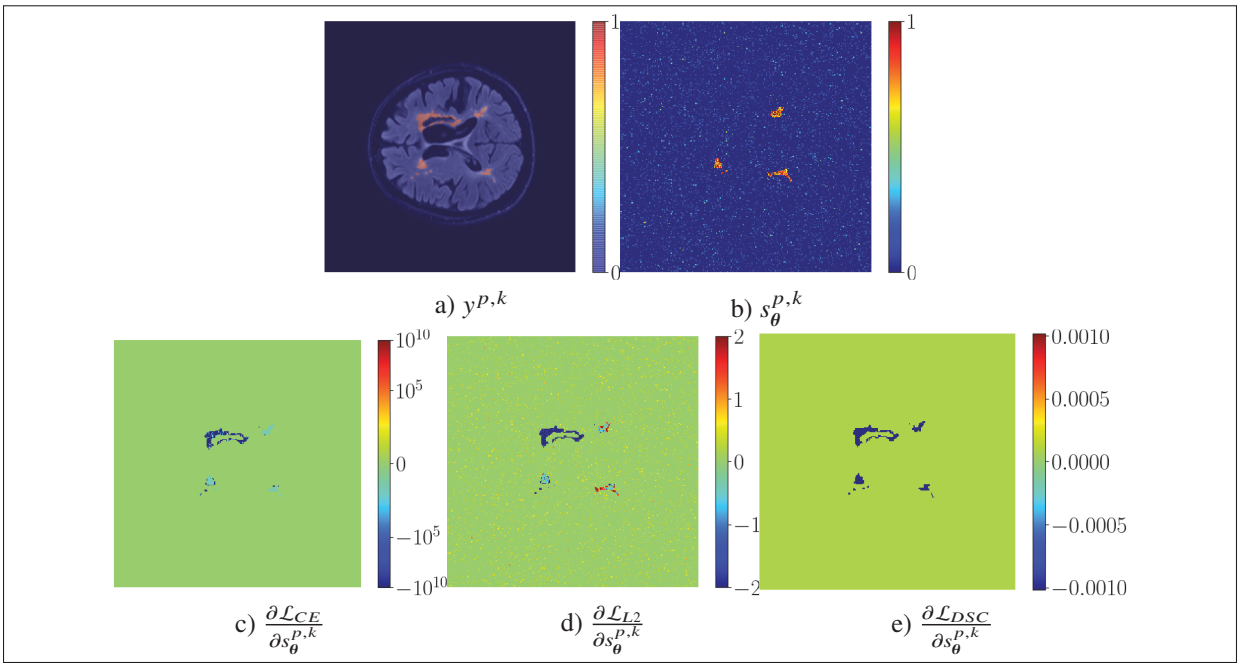


Figure 1.2 Partial derivatives of commons losses wrt. $s_{\theta}^{p,k}$. Notice the variation in the scale of the gradients in (c), (d), (e). Best viewed in colors at high DPI.

"down" if needed[8]: $\mathcal{L}_{L2}$ will push it directly down with its positive gradient. On the contrary, $\mathcal{L}_{CE}$ will do it in an indirect way, by pushing *up* the probabilities for $k' \neq k$. One can also notice that, for each pixel, the gradient depends solely on the pixel, and is not influenced in any manner by its neighbors or other pixels in the image.

The case of $\frac{\partial \mathcal{L}_{DSC}}{\partial s_{\theta}^{p,k}}$ is quite different, and we can quickly see that $\frac{\partial \mathcal{L}_{DSC}}{\partial s_{\theta}^{p,k}} \in \left\{\frac{-2}{\mathrm{U}^k}, \frac{2\mathrm{I}^k}{\mathrm{U}^{k^2}}\right\}$. While the values of U^k and I^k will vary between images, we can notice that the gradient boils down to a weighted negative of the ground truth y. Furthermore, it can be shown easily (notice that I^k and U^k are bounded by $\sum_{p \in \Omega} y^{p,k}$ and $|\Omega|$) that while $-2 \leq \frac{\partial \mathcal{L}_{DSC}}{\partial s_{\theta}^{p,k}} \leq 2$, values in practice will be much closer to 0 than -2 and 2 (see the scale of values in Figure 1.2). Those small gradients might, therefore, require to use a higher learning rate than for $\mathcal{L}_{L2}$ or $\mathcal{L}_{CE}$ if we want to achieve a similar convergence speed.

[8] As we perform a gradient *descent*, the gradient needs to be positive to push the probability down.

1.2.3.3 Modified losses for imbalanced tasks

For tasks with a big data imbalance (where there is orders of magnitude more background than foreground pixels, for instance), using an unmodified standard loss can make the training unstable, or produce a network predicting everything as background. As the vast majority of gradients will push the predicted probabilities *down*, $s_{\boldsymbol{\theta}}$ will naturally remain very close to 0 over the whole image. This can cause the cross-entropy, for instance, to produce values going to infinity for the few foreground pixels, as $-\log(0) = +\infty$.

Some modified losses have been proposed Milletari, Navab & Ahmadi (2016); Ronneberger *et al.* (2015); Sudre, Li, Vercauteren, Ourselin & Cardoso (2017) to deal with this problem, often weighting the components of the losses to give a higher priority to the few foreground pixels. As an example, the often used Generalized Dice Loss (GDL) Sudre *et al.* (2017):

$$\mathcal{L}_{GDL} = \sum_{k \in \mathcal{K}} -2 \frac{w_F^k \sum_{p \in \Omega} s_{\boldsymbol{\theta}}^{p,k} y^{p,k} + w_F^k \sum_{p \in \Omega} (1 - s_{\boldsymbol{\theta}}^{p,k})(1 - y^{p,k})}{w_F^k \left(\sum_{p \in \Omega} s_{\boldsymbol{\theta}}^{p,k} y^{p,k} \right) + w_B^k \left(\sum_{p \in \Omega} (1 - s_{\boldsymbol{\theta}}^{p,k})(1 - y^{p,k}) \right)},$$

where $w_F^k = \frac{1}{\left(\sum_{p \in \Omega} y^{p,k}\right)^2}$ and $w_B^k = \frac{1}{\left(\sum_{p \in \Omega} (1 - y^{p,k})\right)^2}$.

1.3 Regularization and random fields in classical computer vision

1.3.1 Random fields: basics

Discrete random fields—also known as Markov random field (MRF) or conditional random field (CRF)—have been very popular in computer vision for a long time Blake, Kohli & Rother (2011), as they can be applied to a variety of applications and easily embed prior knowledge. A discrete Random Field is a weighted graph $\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$ representing the segmentation of an image: each node correspond to a single pixel, with an associated hidden variable (the labels, $y(p)$), and weighted undirected edges modelling the relationship between pixels. $\mathcal{E}_p = \{q \in \mathcal{V} | (p, q) \in \mathcal{E}\}$ represent the set of *neighbors* of p—its adjacent nodes. $C_{\mathcal{G}}$ represents the set of cliques of $\mathcal{G}$.

Each node has an inherent weight, its *unary* potentials Φ_u, representing the likelihood for a pixel to belongs to a specific target class, and can be derived from another algorithm (such as the output of a neural network, as we will see later in Section 1.3.2).

The edges weights, called *pairwise* potentials Φ_p, can model different relationships between pixels: "how much" they look alike, if their labels are compatible, or how far are they located from each others—it can be used to model priors that we have about the problem. For instance, a common prior in computer vision is about proximity: close pixels *tend* to have the same class. Another one is that boundaries between classes tends to be smooth.

With those potentials, we can compute the *probability* of a segmentation Y (taking the form of a Gibbs distribution):

$$P(Y|I) = \frac{1}{Z(I)} e^{\sum_{c \in C_{\mathcal{G}}} \phi(Y_c|I)}, \tag{1.12}$$

where $\phi(Y_c|I)$ is the potential of clique c conditionned over the image I, and $\frac{1}{Z(I)}$ is a normalizing constant[9]. Similarly, we can define the graph energy such as:

$$E(Y|I) = \sum_{c \in C_{\mathcal{G}}} \phi_c(Y_c|I) \tag{1.13}$$

$$= \sum_{i \in \mathcal{V}} \phi_u(Y(i)|I) + \sum_{(i,j) \in \mathcal{E}} \phi_p(Y(i), Y(j)|I). \tag{1.14}$$

In other words, the graph energy represent the *cost* of a segmentation: assigning an unlikely label is still possible, but it is not free. By balancing this cost across all pixels with respect to all unary and pairwise potentials, we obtain the most likely segmentation *for this specific graph*. If properly defined, the label assignment Y^* that maximize $P(Y|I)$ will produce the most desirable output—for instance by fitting the image edges snuggingly, or having a smooth contour. An example is given in Figure 1.3, comparing a plain segmentation (which uses unary potentials only) and a regularized segmentation with a MRF that minimizes the length of the segmentation. The process of finding this optimal segmentation is called the *Maximum A Posteriori* (MAP),

[9]In this section, for clarity and avoid confusion, we will refer as $I : \Omega \to \mathbb{R}^M$ as an input image. Similarly, we will note: $\phi(Y_c) := \phi(Y_c|I)$. The same will applies for ϕ_u and ϕ_p.

which is an NP-hard problem:

$$Y^* = \arg\max_Y P(Y|I) \tag{1.15}$$

$$= \arg\min_Y E(Y|I). \tag{1.16}$$

a) Input image b) Using unary potentials only c) Final segmentation

Figure 1.3 Illustration of the regularizing effect that a MRF can have, by removing noisy areas in the segmentation. S^1 is the foreground to segment, while S^0 the background to remove.

Depending on how the potentials are defined, and the graph topology, different methods exist. For binary assignments and *sparse* graphs ($|\mathcal{E}_i| << |\Omega|$)—most commonly a Grid CRF—methods such a Graph Cut Boykov & Funka-Lea (2006) are proved to find a global optimum at a reasonable polynomial complexity. For fully connected graphs ($|\mathcal{E}_i| = |\Omega| - 1$)—which can model more complex relationships between pixels at longer spatial distances—solving Equation (1.15) exactly becomes intractable. DenseCRFKrähenbühl & Koltun (2011b) presents an efficient solution to compute an approximation of the solution, with convergence guarantee of the algorithm. We detail it in the next section.

1.3.2 Random fields as post-processing: the case of DenseCRF

CNNs for segmentation tend to have coarse outputs, due to the way the final segmentation map $\hat{Y}$ is obtained. As showed in Equation (1.7), each pixel is maximized independently and spatial consistency is completely ignored. This can be mitigated with additional post-processing. We present here the method of Krahenbühl et al. Krähenbühl & Koltun (2011b), which, as a post-processing, was popularized by DeepLab Chen *et al.* (2015). While not solving the MAP exactly, it computes an approximation of the problem, with convergence guarantees. First, let us present the potentials that they use. When re-using existing softmax probabilities, the unary potentials are simply:

$$\phi_p(Y_p; s_{\boldsymbol{\theta}}) := -\log\left(s_{\boldsymbol{\theta}}(p, Y_p)\right).$$

The pairwise potentials take into account pixel appearances and their spatial distances:

$$\begin{aligned}
\phi_{p,q}(Y(p), Y(q); x) &:= \mathbf{1}_{[Y(p)=Y(q)]} \sum_{m=1}^{2} w^{(m)} \kappa^{(m)}(p, q) \\
\kappa^{(1)}(p, q) &:= w^{(1)} e^{-\frac{|\Omega_p - \Omega_q|^2}{2\omega_\alpha^2} - \frac{|x(p) - x(q)|^2}{2\omega_\beta^2}} \\
\kappa^{(2)}(p, q) &:= w^{(2)} e^{-\frac{|\Omega_p - \Omega_q|^2}{2\omega_\gamma^2}},
\end{aligned}$$

where Ω_p are the pixel coordinates and $x(p)$ their intensity, $w^{(m)}$ are hyper-parameters balancing the kernels. $\kappa^{(1)}$ is an appearance kernel: it ensures local consistency for similarly looking pixels. $\kappa^{(2)}$ is a smoothness kernel, suppressing small isolated regions that are due to noise. Informally, the final segmentation boundary has to align with the image edges, while being smooth.

Since solving the equation (1.15) is an NP-Hard problem, a standard method to facilitate it is to introduce an approximate distribution Q and to minimize it's KL divergence with P:

$$\begin{aligned} \tilde{Y} &:= \arg\min_{Y} \mathrm{KL}(Q(Y)||P(Y)) \\ Q(Y) &:= \prod_{i \in \nu} Q_i(Y_i) \\ Q_i(Y_i = k) &:= \frac{1}{Z_i} e^{-\phi_u(Y_i=k) - \sum_{k' \in \mathcal{K}} \sum_{j \in \epsilon_i} \phi_p(Y_i=k, Y_j=k') Q_j(Y_j=k')}. \end{aligned} \tag{1.17}$$

As Q is the product of independent components, each Q_i can be maximized in parallel, with a message passing algorithm such as Algorithm 1.3. The $\tilde{Q}_i^{(m)}$ update can be accelerated even further, by first downsampling the graph using gaussian filtering, performing the update, and then upsampling it again.

Algorithm 1.3 Overview of Krähenbühl & Koltun (2011b) main algorithm.

Input: *Init* $Q_i(Y_i) \leftarrow \frac{1}{Z_i} e^{-\phi_u(Y_i)} \quad \forall i$
Output: *Current solution* Q_i
while Q_i *not converged* **do**
  $\tilde{Q}_i^{(m)}(k) \leftarrow \sum_{j \in \epsilon_i} \kappa^{(m)}(i,j) Q_j(k) \quad \forall i \in \Omega, \forall (m), \forall k \in \mathcal{K}$
  $\hat{Q}_i(k) \leftarrow \sum_{k' \in \mathcal{K}} 1_{[Y_i = k']} \sum_{m=1}^{2} w^{(m)} \tilde{Q}_i^{(m)}(k) \quad \forall i, \forall k$
  $Q_i(k) \leftarrow \frac{1}{Z_i} e^{-\phi_u(Y_i) - \hat{Q}_i(Yi)} \quad \forall i, \forall k$
end

Because of its good results and rather fast runtime, DenseCRF is now very popular and is used in many current methods as either post-processing or internal regularizer Arnab, Zheng, Jayasumana, Romera-Paredes, Larsson, Kirillov, Savchynskyy, Rother, Kahl & Torr (2018); Chen *et al.* (2015); Rajchl *et al.* (2017). Some limitations remains, as the implementation is CPU bound (which makes it slower to interract with GPU-based deep learning methods), and the complete procedure is not derivable. The high number of hyper-parameters (such as ω_α and ω_β controling the appearance kernel, ω_γ controlling the distance kernel, and $w^{(1)}$ and $w^{(2)}$

balancing the two) requires careful tuning on a new datasets, especially when there is a big discrepancy in contrast and edge sharpness between samples.

1.4 Weakly supervised image segmentation

1.4.1 Partial annotations

The losses that we described in Section 1.2.3 are defined for fully annotated images, i.e., y^p is known $\forall p \in \Omega$. As those labels are very time consuming to produce, some faster (though imperfect) alternatives can be envisioned, as illustrated in Figure 1.4. They can be regrouped in two broad categories: *semi*-annotations and *weak*-annotations. Let us denote $\Omega_L \subseteq \Omega$ the set of labeled pixels, and $\Omega_U \subseteq \Omega$ the set of unlabeled pixels, such as $\Omega_L \cup \Omega_U = \Omega$ and $\Omega_L \cap \Omega_U = \{\emptyset\}$.

Semi-annotations

With those annotations, only a subset Ω_L of pixels is annotated, but there is certainty about those. Examples of such annotations include scribbles and points annotations. The rest is unknown. Notice that $\Omega_L = \Omega$ correspond to the full annotation case.

Weak-annotations

In those cases, the information provided is uncertain, and often correspond to a Multiple Instance Learning setting. This is the case, for instance, of an image-tags or bounding-boxe annotation. For the latter, no pixels outside the bounding box belong to the object, but *some* pixels inside do, although we are not certain about which ones. Other forms of weak labels may include higher-level information, such as size, shape moments, or information derived from radiological reports.

Certainty and uncertainty must be taken into account when designing a method that uses weak labels.

1.4.2 Training with partial labels

Losses from section 1.2.3 cannot be used directly, even on the restrained subset of annotated pixels Ω_L (doing so simply gives very poor results). Methods designed to use semi- and weak-labels can be split in two broad categories: *proposal* based methods and *direct losses* methods.

Full supervision

The optimization model in full supervision takes the following general form:

$$\min_{\boldsymbol{\theta}} \quad \sum_{p \in \Omega} \mathcal{L}(y(p), s_{\boldsymbol{\theta}}(p)),$$

where $\mathcal{L}$ will simply one or a combination of the standard supervised-learning losses introduced earlier.

Proposal based methods

As they attempt to *mimic* full supervision, proposal based methods takes the following general form:

$$\min_{\boldsymbol{\theta}, \tilde{y}} \quad \sum_{p \in \Omega} \mathcal{L}(\tilde{y}(p), s_{\boldsymbol{\theta}}(p)),$$

where $\tilde{y}$ are *pseudo-labels* or *proposals*. These methods attempt to generate a full mask, and then train on it—often alternating between the two. Methods will vary in the way they generate the proposals, and how often they update it—it quickly involve a high number of heuristics and hyper-parameters that must be very carefully tuned. For instance, DeepCut and other methods Dai, He & Sun (2015); Khoreva, Benenson, Hosang, Hein & Schiele (2017); Papandreou, Chen, Murphy & Yuille (2015); Rajchl *et al.* (2017) use either DenseCRF or GraphCut to update their proposals, and might initialize the proposals with GrabCut Rother *et al.* (2004). Simple-does-it Khoreva *et al.* (2017) adds additional heuristics, by discarding proposals where the segmentation size goes below a certain threshold. In those methods, the proposals are updated every few epochs. Pathak et al. Pathak *et al.* (2015a) introduce a proposal to enforce

linear constraints on it, and simustaneously update network predictions and proposals at each iteration.

Direct loss methods

The general form is simpler compared to the previous one, as there is only one variable to optimize:

$$\min_{\theta} \quad \sum_{p \in \Omega_L} \mathcal{L}(y(p), s_\theta(p)) + \alpha \mathcal{R}(s_\theta),$$

where $\mathcal{R}$ is a regularizer and α a scalar balancing the two objectives. Regularizers might take different forms, such as a CRF Tang, Djelouah, Perazzi, Boykov & Schroers (2018a); Tang, Perazzi, Djelouah, Ben Ayed, Schroers & Boykov (2018b); Zheng, Jayasumana, Romera-Paredes, Vineet, Su, Du, Huang & Torr (2015), which can rectify erroneous segmentations when training from scribbles only. ScribbleSup Lin, Dai, Jia, He & Sun (2016) uses super-pixels to enforce consistency over patches of similar-looking pixels, while enabling them to directly supervise a higher fraction of the image. The authors of Qu, Wu, Huang, Yi, Riedlinger, De & Metaxas (2019) leverage point annotations in the context of histopathology images. From labeled points, they derived additional information in the form of a voronoi diagram to generate coarse labels for nuclei segmentation. Their objective function integrated the cross-entropy with coarse labels and the conditional random field (CRF) loss in Tang *et al.* (2018b).

Advantages of a direct loss

We argue that proposal based methods are inherently more unstable than direct loss methods, as early mistakes in $\tilde{y}$ can reinforce themselves by training the network with contradicting information. As convolutional layers are designed to have the same activation for similar looking patches of images, it is implicitely expected they have the same label. When this is not the case, at the back-propagation step, the layer will be updated with two contradicting informations—cancelling each others. On the contrary, direct losses might supervise *less pixels* directly, but in a more reliable way. In this context, *less is more*. Dealing with those inherent

limitations can be done only with many heuristics, ad-hoc regularizers and careful tuning of the hyper-parameters. This is not just hypothetical, but verified experimentally in several of our papers, where comparing to proposal based methods showed the same pattern of instability and collapsing predictions, as showed in Figure 1.5.

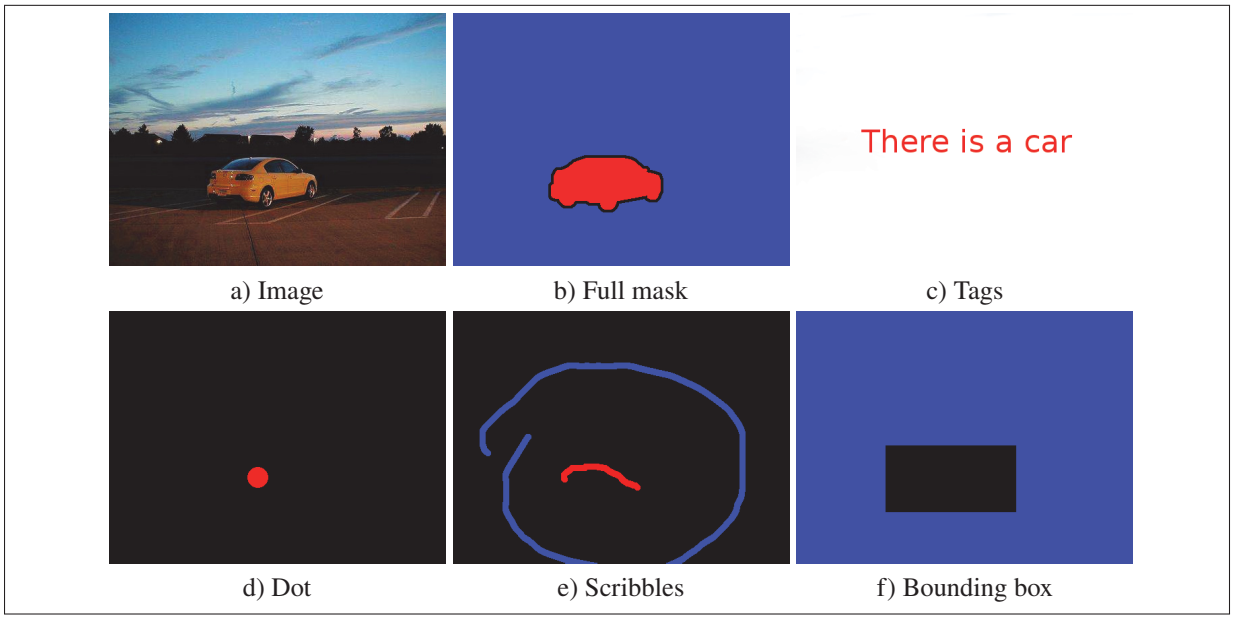


Figure 1.4 Illustration and comparison of different semi- and weak-annotations. Blue represents the background class, red the foreground class, and black is undetermined.

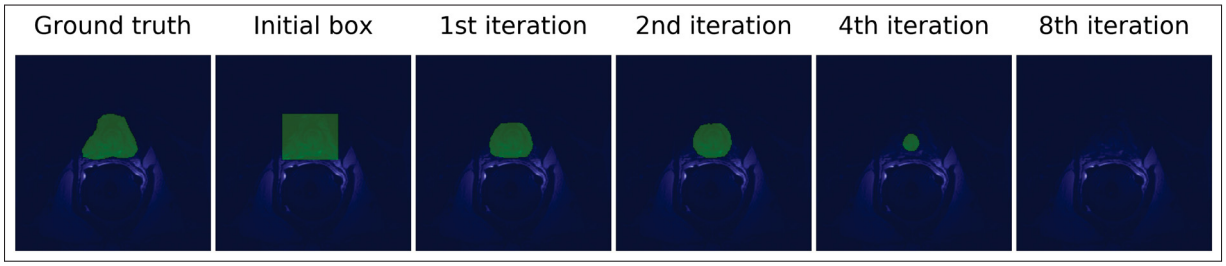


Figure 1.5 Evolution of the proposals from DeepCut Rajchl *et al.* (2017) on the PROMISE12 dataset Litjens *et al.* (2014): the prostate segmentation gradually disappears over time.

1.5 Constrained deep networks

1.5.1 Challenges of standard Lagrangian optimization

Let us first remember the general formulation of constrained optimization in Equation (1.1), and adapt it to a deep learning setting:

$$\begin{aligned} \min_{\boldsymbol{\theta}} \quad & \sum_{n \in \mathcal{D}} \mathcal{L}(s^n_{\boldsymbol{\theta}}, y^n) && && (1.18) \\ \text{subject to} \quad & f_1(s^n_{\boldsymbol{\theta}}) && \forall n \in \mathcal{D} \\ & \dots \\ & f_P(s^n_{\boldsymbol{\theta}}) && \forall n \in \mathcal{D}. \end{aligned}$$

In this case, we have P constraints to enforce *on every single sample*. The Lagrangian corresponding to (1.18) would be:

$$\max_{\boldsymbol{\lambda} \geq 0} \min_{\boldsymbol{\theta}} \quad \sum_{n \in \mathcal{D}} \mathcal{L}(s^n_{\boldsymbol{\theta}}, y^n) + \sum_{i=1}^{P} \sum_{n=1}^{N} \lambda^n_i f_i(s^n_{\boldsymbol{\theta}}) \tag{1.19}$$

where $\boldsymbol{\lambda} \in \mathbb{R}_+^{P \times N}$ is the dual variable (or Lagrange-multiplier) vector, with λ^n_i the multiplier associated with constraint $f_i(s^n_{\boldsymbol{\theta}}) \leq 0$. A standard Lagrangian would alternatively optimize with respect to network parameters $\boldsymbol{\theta}$ and dual variable $\boldsymbol{\lambda}$.

Lagrangian optimization has several well-known theoretical and practical advantages over penalty methods Fletcher (1987); Gill, Murray & Wright (1981): it finds automatically the optimal weights of the constraints, acts as a barrier for satisfied constraints and guarantees constraint satisfaction when feasible solutions exist. Unfortunately, in the case of deep networks, solving exactly the Lagrangian would require to retrain completely the neural network twice *at each iteration*, alternating the optimization of a CNN for the primal with SGD, and projected gradient-ascent iterates for the dual. Due to the time-scales already at play to train a neural network once (from a few hours to several days), it is simply not feasible. Another

important difficulty in Lagrangian optimization is the interplay between stochastic optimization (e.g., SGD) for the primal and the iterates/projections for the dual. Basic gradient methods have well-known issues with deep networks, e.g., they are sensitive to the learning rate and prone to weak local minima. Therefore, the dual part in Lagrangian optimization might obstruct the practical and theoretical benefits of stochastic optimization (e.g., speed and strong generalization performance), which are widely established for unconstrained deep network losses Hardt, Recht & Singer (2016). This is in line with the results reported by the authors of Márquez-Neila, Salzmann & Fua (2017) in the context of 3D human pose estimation. In their case, replacing the equality constraints with simple quadratic penalties yielded better results than Lagrangian optimization.

1.5.2 Challenges of interior point methods

Interior point methods, such as the barrier methods Boyd & Vandenberghe (2004); Fiacco & McCormick (1990), gained a lot of popularity in 60s, as they can bypass the expensive dual-updates of Lagrangian optimization—while still providing convergence and optimality guarantees. The requirement for *interior-point* methods is to start with a feasible point ($\boldsymbol{\theta}$ in the case of deep learning), such as all constraints are satisfied. Then, the original problem can be optimized with an added *barrier* that gets close to infinity when the constraints approach their upper bound (as shown in Figure 1.6). To update the example from Equation (1.18), the new optimization problem will take the following form:

$$\min_{\boldsymbol{\theta}} \quad \sum_{n \in \mathcal{D}} \left[\mathcal{L}(s^n_{\boldsymbol{\theta}}, y^n) + \sum_{p=1}^{P} \psi_t \left(f_p(s^n_{\boldsymbol{\theta}}) \right) \right] \tag{1.20}$$

$$\psi_t(z) := -\frac{1}{t} \log(-z), \tag{1.21}$$

where $t > 0$ is a scalar value the starts small and is gradually increased over time. Notice that $\lim_{t \to +\infty} \phi_t = +\infty_{[z<0]}$. The reader will quickly notice that ϕ_t is undefined for $z \geq 0$, and the infinite penalty as $z \to 0$ prevents the optimization procedure to ever go out of bounds. Depending on the problem to be solved, finding a strictly feasible starting point might not be

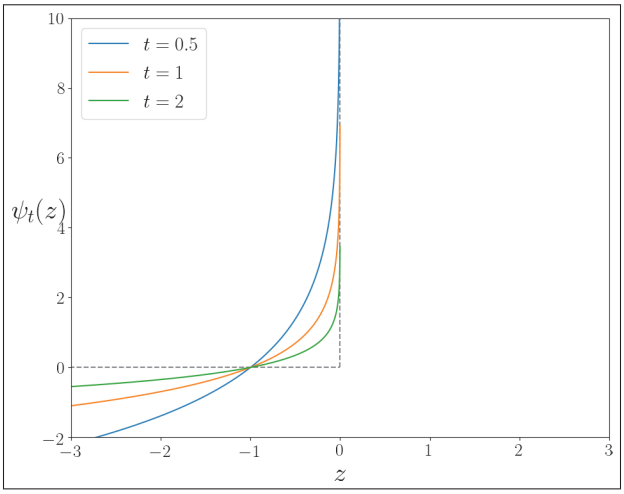


Figure 1.6 Parameterized log-*barrier*, for different t values.

doable analytically. In classical optimization, a first step called Phase I is required, and consists of finding a starting point that satisfies the constraints, without considerations for optimality. Then, a Phase II optimization will refine this starting point, using the most adapted optimization algorithm for the task.

When dealing with deep networks, where initial weights have to be randomaly initialized, a feasible starting point cannot be found easily. Moreover, solving the Phase I problem for deep network is as difficult as solving Phase II. The interior point method becomes self-defeating for deep networks: in order to solve this constrained optimization problem, one has to first solve it.

1.5.3 ReLU Lagrangian modification Nandwani *et al.* (2019)

To accelerate training with a Lagrangian setting, one might decide to relax the alternating updates: for instance, re-using the parameters from the previous iteration, and updating λ less frequently—every few epochs. This not only adds arbitrary cut-offs, but also introduces some new instabilities. As λ_i^n is updated less frequently, it can remain positive even when the constrained function f_i is satisfied ($f_i(s_{\boldsymbol{\theta}}^n) \leq 0$). Because of this, the SGD on $\boldsymbol{\theta}$ will continue to minize that term, even though is should be "out of the way".

To manage this new problem, Nandwani et al. Nandwani *et al.* (2019) proposed a ReLu modification of the Lagrangian term, avoiding completely the projection steps for the dual variables. The dual variables $\lambda_i^n \; \forall n \in \mathcal{D}$ are also regrouped into a single λ_i for each constrained function f_i, in an attempt to save memory:

$$\max_{\boldsymbol{\lambda}} \min_{\boldsymbol{\theta}} \quad \mathcal{L}(S_{\boldsymbol{\theta}}, \boldsymbol{\lambda}) = \mathcal{E}(\boldsymbol{\theta}) + \sum_{i=1}^{P} \lambda_i \sum_{n=1}^{N} \max(0, f_i(s_{\boldsymbol{\theta}}^n)). \tag{1.22}$$

Since the gradient $\nabla\boldsymbol{\lambda}$ is always positive, $\boldsymbol{\lambda}$ can only increase over time. This, we argue, can make updates and training unstable, especially when there is a high number of competing constraints to satisfy.

1.5.4 Lagrangian with proposals Pathak *et al.* (2015a)

Another approach by Pathak et al. was introduced in the context of weakly supervised image segmentation, to constraint the size of the predicted segmentation. The problem they are trying to solve is therefore:

$$\begin{aligned} \min_{\boldsymbol{\theta}} \quad & \sum_n \mathcal{L}_{CE}(y^n, s_{\boldsymbol{\theta}}^n) && && (1.23) \\ \text{s.t.} \quad & {s_{\boldsymbol{\theta}}^n}^\top a_1 - b_1 \leq 0 && \forall n \in \mathcal{D} \\ & \dots \\ & {s_{\boldsymbol{\theta}}^n}^\top a_P - b_P \leq 0 && \forall n \in \mathcal{D}, \end{aligned}$$

where y^n is partially or completely unknown, $a_1, ..., a_P \in \mathbb{R}^{|\Omega|}$ and $b_1, ..., b_P \in \mathbb{R}^K$. The authors first insight was to rewrite y^n as a continuous variable ($\in [0, 1]^{K \times |\Omega|}$), and to introduce a latent

variable $\tilde{y}^n \in [0,1]^{K\times|\Omega|}$, on which they imposed the linear constraints:

$$\begin{aligned} \min_{\tilde{y},\boldsymbol{\theta}} \quad & \sum_n \mathrm{KL}(\tilde{y}^n||s^n_{\boldsymbol{\theta}}) && (1.24)\\ s.t. \quad & \tilde{y}^{n\top} a_1 - b_1 \preceq 0 && \forall n \in \mathcal{D} \\ & \dots \\ & \tilde{y}^{n\top} a_P - b_P \preceq 0 && \forall n \in \mathcal{D} \\ & \mathbf{1}^\top \tilde{y}^n_p = 1 && \forall n \in \mathcal{D}, \forall p \in |\Omega|. \end{aligned}$$

where KL denotes the KullbackLeibler divergence.

The corresponding Lagrangian is:

$$\begin{aligned} \max_{\boldsymbol{\lambda},\boldsymbol{\nu}} \min_{\tilde{y},\boldsymbol{\theta}} \quad & \sum_n \left(\mathrm{KL}(\tilde{y}^n||s^n_{\boldsymbol{\theta}}) + \sum_{i=1}^{P} \lambda_i^n (\tilde{y}^{n\top} a_i - b_i) + \sum_{p\in\Omega} \nu_p^n (\mathbf{1}^\top \tilde{y}^n_p - 1) \right) && (1.25)\\ s.t. \quad & \boldsymbol{\lambda} \succeq 0, \end{aligned}$$

where $\boldsymbol{\lambda} \in \mathbb{R}_+^{P\times|\mathcal{D}|}$ and $\boldsymbol{\nu} \in \mathbb{R}^{|\mathcal{D}|\times|\Omega|}$ are the Lagrangian dual variables. Minimizing $\tilde{y}$, for constant $\boldsymbol{\theta}, \boldsymbol{\lambda}, \boldsymbol{\nu}$, can be solved analytically. Updating $\boldsymbol{\lambda}$ and $\boldsymbol{\nu}$ requires to perform a projected gradient ascent[10]. Pathak *et al.* (2015a) concluded that is was best to perform this at each minibatch, for the corresponding samples. While limited to linear functions, it could in theory be extended to any function f_i. However, minimizing $\tilde{y}$ would not be analytically solvable anymore, and would (in most cases) requires a dedicated descent procedure. The introduction of latent variable $\tilde{y}^n$ makes Pathak *et al.* (2015a) a *proposal* based method, with the same limitations and caveats: early mistakes in the training process can reinforce themselves, or make the training unstable when partial labels (such as scribbles) are available.

[10]We detail the whole algorithm and equations in the Supplemental material 4.

CHAPTER 2

CONSTRAINED-CNN LOSSES FOR WEAKLY SUPERVISED SEGMENTATION

Hoel Kervadec[a], Jose Dolz[b], Meng Tang[c], Eric Granger[a], Yuri Boykov[c], Ismail Ben Ayed[a]

[a] Département de génie des systèmes, ÉTS Montréal, QC, Canada,
[b] Département de génie logiciel et des TI, ÉTS Montréal, QC, Canada,
[c] Department of computer science, University of Waterloo, ON, Canada

Oral presentation at Medical Imaging with Deep Learning (MIDL) 2018, Amsterdam.
CIFAR student travel award.
Invited for a deep learning special issue in Medical Image Analysis (MEDIA), vol 54.

Abstract

Weakly-supervised learning based on, e.g., partially labelled images or image-tags, is currently attracting significant attention in CNN segmentation as it can mitigate the need for full and laborious pixel/voxel annotations. Enforcing high-order (global) inequality constraints on the network output (for instance, to constrain the size of the target region) can leverage unlabeled data, guiding the training process with domain-specific knowledge. Inequality constraints are very flexible because they do not assume exact prior knowledge. However, constrained Lagrangian dual optimization has been largely avoided in deep networks, mainly for computational tractability reasons. To the best of our knowledge, the method of Pathak et al. Pathak *et al.* (2015a) is the only prior work that addresses deep CNNs with linear constraints in weakly supervised segmentation. It uses the constraints to synthesize fully-labeled training masks (proposals) from weak labels, mimicking full supervision and facilitating dual optimization.

We propose to introduce a differentiable penalty, which enforces inequality constraints directly in the loss function, avoiding expensive Lagrangian dual iterates and proposal generation. From constrained-optimization perspective, our simple penalty-based approach is not optimal as there is no guarantee that the constraints are satisfied. However, surprisingly, it yields *substantially* better results than the Lagrangian-based constrained CNNs in Pathak *et al.* (2015a), while reducing the computational demand for training. By annotating only a small fraction of the

pixels, the proposed approach can reach a level of segmentation performance that is comparable to full supervision on three separate tasks. While our experiments focused on basic linear constraints such as the target-region size and image tags, our framework can be easily extended to other non-linear constraints, e.g., invariant shape moments Klodt & Cremers (2011) and other region statistics Lim *et al.* (2014). Therefore, it has the potential to close the gap between weakly and fully supervised learning in semantic medical image segmentation. Our code is publicly available.

2.1 Introduction

In the recent years , deep convolutional neural networks (CNNs) have been dominating semantic segmentation problems, both in computer vision and medical imaging, achieving ground-breaking performances when full-supervision is available Dolz, Desrosiers & Ben Ayed (2018); Litjens, Kooi, Bejnordi, Setio, Ciompi, Ghafoorian, van der Laak, van Ginneken & Sánchez (2017); Long *et al.* (2015). In semantic segmentation, full supervision requires laborious pixel/voxel annotations, which may not be available in a breadth of applications, more so when dealing with volumetric data. Furthermore, pixel/voxel level annotations become a serious impediment for scaling deep segmentation networks to new object categories or target domains.

To reduce the burden of pixel-level annotations, weak supervision in the form partial or uncertain labels, for instance, bounding boxes Dai *et al.* (2015), points Bearman *et al.* (2016), scribbles Lin *et al.* (2016); Tang *et al.* (2018a), or image tags Pinheiro & Collobert (2015); Wei, Liang, Chen, Shen, Cheng, Feng, Zhao & Yan (2017), is attracting significant research attention. Imposing prior knowledge on the networks output in the form of unsupervised loss terms is a well-established approach in machine learning Goodfellow, Bengio & Courville (2016); Weston, Ratle, Mobahi & Collobert (2012). Such priors can be viewed as regularization terms that leverage unlabeled data, embedding domain-specific knowledge. For instance, the recent studies in Tang *et al.* (2018a,1) showed that direct regularization losses, e.g., dense conditional random field (CRF) or pairwise clustering, can yield outstanding results in weakly supervised segmentation, reaching almost full-supervision performances in natural image segmentation.

Surprisingly, such a principled direct-loss approach is not common in weakly supervised segmentation. In fact, most of the existing techniques synthesize fully-labeled training masks (proposals) from the available partial labels, mimicking full supervision Kolesnikov & Lampert (2016); Lin *et al.* (2016); Papandreou *et al.* (2015); Rajchl *et al.* (2017). Typically, such proposal-based techniques iterate two steps: CNN learning and proposal generation facilitated by dense CRFs and fast mean-field inference Krähenbühl & Koltun (2011a), which are now the de-facto choice for pairwise regularization in semantic segmentation algorithms.

Our purpose here is to embed high-order (global) inequality constraints on the network outputs directly in the loss function, so as to guide learning. For instance, assume that we have some prior knowledge on the size (or volume) of the target region, e.g., in the form of lower and upper bounds on size, a common scenario in medical image segmentation Gorelick, Schmidt & Boykov (2013); Niethammer & Zach (2013). Let $I : \Omega \subset \mathbb{R}^{2,3} \rightarrow \mathbb{R}$ denotes a given training image, with Ω a discrete image domain and $|\Omega|$ the number of pixels/voxels in the image. $\Omega_L \subseteq \Omega$ is a weak (partial) ground-truth segmentation of the image, taking the form of a partial annotation of the target region, e.g., a few points (see Figure 2.2). In this case, one can optimize a *partial* cross-entropy loss subject to inequality constraints on the network outputs Pathak *et al.* (2015a):

$$\min_{\boldsymbol{\theta}} \ \mathcal{H}(S) \quad \text{s.t} \quad a \leq \sum_{p \in \Omega} S_p \leq b \tag{2.1}$$

where $\mathrm{S} = (S_1, \ldots, S_{|\Omega|}) \in [0,1]^{|\Omega|}$ is a vector of softmax probabilities[1] generated by the network at each pixel p and $\mathcal{H}(S) = -\sum_{p \in \Omega_L} \log(S_p)$. Priors a and b denote the given upper and lower bounds on the size (or cardinality) of the target region. Inequality constraints of the form in (2.1) are very flexible because they do not assume exact knowledge of the target size, unlike Boykov, Isack, Olsson & Ayed (2015); Jia, Huang, Eric, Chang & Xu (2017); Zhang, David & Gong (2017a). Also, multiple instance learning (MIL) constraints Pathak *et al.* (2015a), which enforce image-tag priors, can be handled by constrained model (2.1). Image

[1]The softmax probabilities take the form: $S_p(\boldsymbol{\theta}, I) \propto \exp f_p(\boldsymbol{\theta}, I)$, where $f_p(\boldsymbol{\theta}, I)$ is a real scalar function representing the output of the network for pixel p. For notation simplicity, we omit the dependence of S_p on $\boldsymbol{\theta}$ and I as this does not result in any ambiguity in the presentation.

tags are a form of weak supervision, which enforce the constraints that a target region is present or absent in a given training image Pathak *et al.* (2015a). They can be viewed as particular cases of the inequality constraints in (2.1). For instance, a suppression constraint, which takes the form $\sum_{p\in\Omega} S_p \leq 0$, enforces that the target region is not in the image. $\sum_{p\in\Omega} S_p \geq 1$ enforces the presence of the region.

Even though constraints of the form (2.1) are linear (and hence convex) with respect to the network outputs, constrained problem (2.1) is very challenging due to the non-convexity of CNNs. One possibility would be to minimize the corresponding Lagrangian dual. However, as pointed out in Márquez-Neila *et al.* (2017); Pathak *et al.* (2015a), this is computationally intractable for semantic segmentation networks involving millions of parameters; one has to optimize a CNN within each dual iteration. In fact, constrained optimization has been largely avoided in deep networks Ravi, Dinh, Lokhande & Singh (2019), even thought some Lagrangian techniques were applied to neural networks a long time before the deep learning era Platt & Barr (1988); Zhang & Constantinides (1992). These constrained optimization techniques are not applicable to deep CNNs as they solve large linear systems of equations. The numerical solvers underlying these constrained techniques would have to deal with matrices of very large dimensions in the case of deep networks Márquez-Neila *et al.* (2017).

To the best of our knowledge, the method of Pathak et al. Pathak *et al.* (2015a) is the only prior work that addresses inequality constraints in deep weakly supervised CNN segmentation. It uses the constraints to synthesize fully-labeled training masks (proposals) from the available partial labels, mimicking full supervision, which avoids intractable dual optimization of the constraints when minimizing the loss function. The main idea of Pathak *et al.* (2015a) is to model the proposals via a latent distribution. Then, it minimize a KL divergence, encouraging the softmax output of the CNN to match the latent distribution as closely as possible. Therefore, they impose constraints on the latent distribution rather than on the network output, which facilitates Lagrangian dual optimization. This decouples stochastic gradient descent learning of the network parameters and constrained optimization: The authors of Pathak *et al.* (2015a) alternate between optimizing w.r.t the latent distribution, which corresponds to proposal generation

subject to the constraints[2], and standard stochastic gradient descent for optimizing w.r.t the network parameters.

We propose to introduce a differentiable term, which enforces inequality constraints (2.1) directly in the loss function, avoiding expensive Lagrangian dual iterates and proposal generation. From constrained optimization perspective, our simple approach is not optimal as there is no guarantee that the constraints are satisfied. However, surprisingly, it yields *substantially* better results than the Lagrangian-based constrained CNNs in Pathak *et al.* (2015a), while reducing the computational demand for training. In the context of cardiac image segmentation, we reached a performance close to full supervision while using a fraction of the full ground-truth labels (0.1%). Our framework can be easily extended to non-linear inequality constraints, e.g., invariant shape moments Klodt & Cremers (2011) or other region statistics Lim *et al.* (2014). Therefore, it has the potential to close the gap between weakly and fully supervised learning in semantic medical image segmentation. Our code is publicly available [3].

2.2 Related work

2.2.1 Weak supervision for semantic image segmentation

Training segmentation models with partial and/or uncertain annotations is a challenging problem Buhmann, Ferrari & Vezhnevets (2012); Vezhnevets, Ferrari & Buhmann (2011). Due to the relatively easy task of providing global, image-level information about the presence or absence of objects in an image, many weakly supervised approaches used image tags to learn a segmentation model Verbeek & Triggs (2007); Vezhnevets & Buhmann (2010). For example, in Verbeek & Triggs (2007), a probabilistic latent semantic analysis (PLSA) model was learned from image-level keywords. This model was later employed as a unary potential in a Markov random field (MRF) to capture the spatial 2D relationships between neighbours. Also, bounding boxes have become very popular as weak annotations due, in part, to the wide use of classical

[2]This sub-problem is convex when the constraints are convex.

[3]The code can be found at https://github.com/LIVIAETS/SizeLoss_WSS

interactive segmentation approaches such as the very popular GrabCut Rother *et al.* (2004). This method learns two Gaussian mixture models (GMM) to model the foreground and background regions defined by the bounding box. To segment the image, appearance and smoothness are encoded in a binary MRF, for which exact inference via graph-cuts is possible, as the energies are sub-modular. Another popular form of weak supervision is the use of scribbles, which might be performed interactively by an annotator so as to correct the segmentation outcome.

GrabCut is a notable example in a wide body of "shallow" interactive segmentation works that used weak supervision before the deep learning era. More recently, within the computer vision community, there has been a substantial interest in leveraging weak annotations to train deep CNNs for color image segmentation using, for instance, image tags Papandreou *et al.* (2015); Pathak *et al.* (2015a); Pathak, Shelhamer, Long & Darrell (2015b); Pinheiro & Collobert (2015); Wei *et al.* (2017); Xu, Schwing & Urtasun (2014), bounding boxes Dai *et al.* (2015); Khoreva *et al.* (2017); Rajchl *et al.* (2017), scribbles Lin *et al.* (2016); Tang *et al.* (2018a,1); Vernaza & Chandraker (2017); Xu, Schwing & Urtasun (2015) or points Bearman *et al.* (2016). Most of these weakly supervised semantic segmentation techniques mimic full supervision by generating full training masks (segmentation proposals) from the weak labels. The proposals can be viewed as synthesized ground-truth used to train a CNN. In general, these techniques follow an iterative process that alternates two steps: (1) standard stochastic gradient descent for training a CNN from the proposals; and (2) standard regularization-based segmentation, which yields the proposals. This second step typically uses a standard optimizer such mean-field inference Papandreou *et al.* (2015); Rajchl *et al.* (2017) or graph cuts Lin *et al.* (2016). In particular, the dense CRF regularizer of Krähenbühl and Koltun Krähenbühl & Koltun (2011a), facilitated by fast parallel mean-field inference, has become very popular in semantic segmentation, both in the fully Arnab *et al.* (2018); Chen *et al.* (2015) and weakly Papandreou *et al.* (2015); Rajchl *et al.* (2017) supervised settings. This followed from the great success of DeepLab Chen *et al.* (2015), which popularized the use of dense CRF and mean-field inference as a post-processing step in the context fully supervised CNN segmentation.

An important drawback of these proposal strategies is that they are vulnerable to errors in the proposals, which might reinforce themselves in such self-taught learning schemes Chapelle, Schölkopf & Zien (2006), undermining convergence guarantee. The recent approaches in Tang *et al.* (2018a,1) have integrated standard regularizers such as dense CRF or pairwise graph clustering directly into the loss functions, avoiding extra inference steps or proposal generation. Such direct regularization losses achieved state-of-the-art performances for weakly supervised color segmentation, reaching near full-supervision accuracy. While these approaches encourage pairwise consistencies between pixels during training, they do not explicitly impose global constraint as in (2.1).

2.2.2 Medical image segmentation with weak supervision

Despite the increasing amount of works focusing on weakly supervised deep CNNs in semantic segmentation of color images, leveraging weak annotations in medical imaging settings is not simple. To our knowledge, the literature on this matter is still scarce, which makes weak-supervision approaches appealing in medical image segmentation. As in color images, common settings for weak annotations are bounding boxes. For instance, DeepCut Rajchl *et al.* (2017) follows a similar setting as Papandreou *et al.* (2015). It generates image proposals, which are refined by a dense CRF before being re-used as "fake" labels to train the CNN. Using the bounding boxes as initializations for the Grab-cut algorithm, the authors showed that, by this iterative optimization scheme, one can obtain a performance better than the shallow counterpart, i.e., GrabCut. In another weakly supervised scenario Rajchl, Lee, Schrans, Davidson, Passerat-Palmbach, Tarroni, Alansary, Oktay, Kainz & Rueckert (2016), images were segmented in an unsupervised manner, generating a set of super-pixels Achanta, Shaji, Smith, Lucchi, Fua, Süsstrunk et al. (2012), among which users had to select the regions belonging to the object of interest. Then, these masks generated from the super-pixels were employed to train a CNN. Nevertheless, as proposals are generated in an unsupervised manner, and due to the poor contrast and challenging targets typically present in medical images, these "fake" labels are likely prone to errors, which can be propagated during training, as stated before.

2.2.3 Constrained CNNs

To the best of our knowledge, there are only a few recent works Jia *et al.* (2017); Márquez-Neila *et al.* (2017); Pathak *et al.* (2015a) that addressed imposing global constraints on deep CNNs. In fact, standard Lagrangian-dual optimization has been completely avoided in modern deep networks involving millions of parameters. As pointed out recently in Márquez-Neila *et al.* (2017); Pathak *et al.* (2015a), there is a consensus within the community that imposing constraints on the outputs of deep CNNs that are common in modern computer vision and medical image analysis problems is impractical: the direct use of Lagrangian-dual optimization for networks with millions of parameters requires training a whole CNN after each iterative dual step Pathak *et al.* (2015a). To avoid computationally intractable dual optimization, Pathak et al. Pathak *et al.* (2015a) imposed inequality constraints on a latent distribution instead of the network output. This latent distribution describes a "fake" ground truth (or segmentation proposal). Then, they trained a single CNN so as to minimize the KL divergence between the network probability outputs and the latent distribution. This prior-art work is the most closely related to our study and, to our knowledge, is the only work that addressed inequality constraints in weakly supervised CNN segmentation. The work in Márquez-Neila *et al.* (2017) imposed hard equality constraints on 3D human pose estimation. To tackle the computational difficulty, they used a Kyrlov sub-space approach and limited the solver to only a randomly selected sub-set of the constraints within each iteration. Therefore, constraints that are satisfied at one iteration may not be satisfied at the next, which might explain the negative results in Márquez-Neila *et al.* (2017). A surprising result in Márquez-Neila *et al.* (2017) is that replacing the equality constraints with simple L_2 penalties yields better results than Lagrangian optimization, although such a simple penalty-based formulation does not guarantee constraint satisfaction. A similar L_2 penalty was used in Jia *et al.* (2017) to impose equality constraints on the size of the target regions in the context of histopathology segmentation. While the equality-constrained formulations in Jia *et al.* (2017); Márquez-Neila *et al.* (2017) are very interesting, they assume exact knowledge of the target function (e.g., region size), unlike the

inequality-constraint formulation in (2.1), which allows much more flexibility as to the required prior domain-specific knowledge.

2.3 Proposed loss function

We propose the following loss for weakly supervised segmentation:

$$\mathcal{H}(S) + \lambda\, C\,(V_S), \tag{2.2}$$

where $V_S = \sum_{p\in\Omega} S_p$, λ is a positive constant that weighs the importance of constraints, and function C is given by (See the illustration in Fig. **??**):

$$C(V_S) = \begin{cases} (V_S - a)^2\,, & \text{if } V_S < a \\ (V_S - b)^2\,, & \text{if } V_S > b \\ 0, & \text{otherwise} \end{cases} \tag{2.3}$$

Now, our differentiable term C accommodates standard stochastic gradient descent. During back-propagation, the term of gradient-descent update corresponding to C can be written as follows:

$$-\frac{\partial C(V_S)}{\partial \boldsymbol{\theta}} \propto \begin{cases} (a - V_S)\,\frac{\partial S_p}{\partial \boldsymbol{\theta}}, & \text{if } V_S < a \\ (b - V_S)\,\frac{\partial S_p}{\partial \boldsymbol{\theta}}, & \text{if } V_S > b \\ 0, & \text{otherwise} \end{cases} \tag{2.4}$$

where $\frac{\partial S_p}{\partial \boldsymbol{\theta}}$ denotes the standard derivative of the softmax outputs of the network. The gradient in (2.4) has a clear interpretation. During back-propagation, when the current constraints are satisfied, i.e., $a \le V_S \le b$, observe that $\frac{\partial C(V_S)}{\partial \boldsymbol{\theta}} = 0$. Therefore, in this case, the gradient stemming from our term has no effect on the current update of the network parameters. Now, suppose without loss of generality that the current set of parameters $\boldsymbol{\theta}$ corresponds to $V_S < a$, which means the current target region is smaller than its lower bound a. In this case of constraint violation, term $(a - V_S)$ is positive and, therefore, the first line of (2.4) performs a gradient

ascent step on softmax outputs, increasing S_p. This makes sense because it increases the size of the current region, V_S, so as to satisfy the constraint. The case $V_S > b$ has a similar interpretation.

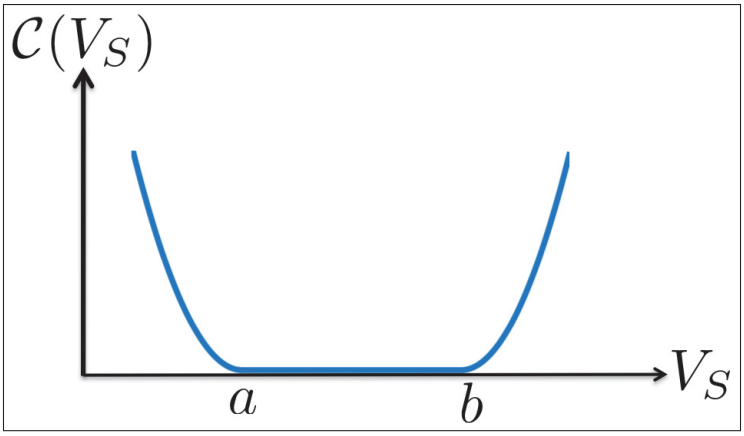


Figure 2.1 Illustration of our differentiable loss for imposing soft size constraints on the target region.

The next section details the dataset, the weak annotations and our implementation. Then, we report comprehensive evaluations of the effect of our constrained-CNN losses on segmentation performance. We also report comparisons to the Lagrangian-based constrained CNN method in Pathak *et al.* (2015a) and to the fully supervised setting.

2.4 Experiments

2.4.1 Medical Image Data

In this section, the proposed loss function is evaluated on three publicly available datasets, each corresponding to a different application—cardiac, vertebral body and prostate segmentation. Below are additional details of these data sets.

Left-ventricle (LV) on cine MRI

A part of our experiments focused on left ventricular endocardium segmentation. We used the training set from the publicly available data of the 2017 ACDC Challenge[4]. This set consists of 100 cine magnetic resonance (MR) exams covering well defined pathologies: dilated cardiomyopathy, hypertrophic cardiomyopathy, myocardial infarction with altered left ventricular ejection fraction and abnormal right ventricle. It also included normal subjects. Each exam contains acquisitions only at the diastolic and systolic phases. The exams were acquired in breath-hold with a retrospective or prospective gating and a SSFP sequence in 2-chambers, 4-chambers and in short-axis orientations. A series of short-axis slices cover the LV from the base to the apex, with a thickness of 5 to 8 mm and an inter-slice gap of 5 mm. The spatial resolution goes from 0.83 to 1.75 mm^2/pixel. For all the experiments, we employed the same 75 exams for training and the remaining 25 for validation.

Vertebral body (VB) on MR-T2

This dataset contains 23 3D T2-weighted turbo spin echo MR images from 23 patients and the associated ground-truth segmentation, and is freely available from [5]. Each patient was scanned with 1.5 Tesla MRI Siemens scanner (Siemens Healthcare, Erlangen, Germany) to generate T2-weighted sagittal images. All the images are sampled to have the same sizes of 39×305×305 voxels, with a voxel spacing of 2×1.25×1.25 mm^3. In each image, 7 vertebral bodies, from T11 to L5, were manually identified and segmented, resulting in 161 labeled regions in total. For this dataset, we employed 15 scans for training and the remaining 5 for validation.

Prostate segmentation on MR-T2

The third dataset was made available at the MICCAI 2012 prostate MR segmentation challenge[6]. It contains the transversal T2-weighted MR images of 50 patients acquired at different centers

[4]https://www.creatis.insa-lyon.fr/Challenge/acdc/
[5]http://dx.doi.org/10.5281/zenodo.22304
[6]https://promise12.grand-challenge.org

with multiple MRI vendors and different scanning protocols. It is comprised of various diseases, i.e., benign and prostate cancers. The images resolution ranges from $15 \times 256 \times 256$ to $54 \times 512 \times 512$ voxels with a spacing ranging from $2 \times 0.27 \times 0.27$ to $4 \times 0.75 \times 0.75 \text{mm}^3$. We employed 40 patients for training and 10 for validation.

2.4.2 Weak annotations

To show that the proposed approach is robust to the strategy for generating the weak labels, as well as to their location, we consider two different strategies generating weak annotations from fully labeled images. Figure 2.2 depicts some examples of fully annotated images and the corresponding weak labels.

Erosion

For the left-ventricle dataset, we employed binary erosion on the fully annotations with a kernel of size 10×10. If the resulted label disappeared, we repeated the operation with a smaller kernel (i.e., 7×7) until we get a small contour. Thus, the total number of annotated pixels represented the 0.1% of the labeled pixels in the fully supervised scenario. This correspond to the second row in Figure 2.2.

Random point

The weak labels for the vertebral body and prostate datasets were generated by randomly selecting a point within the ground-truth mask and creating a circle around it with a maximum radius of 4 pixels (fourth and sixth row in Fig. 2.2), while ensuring there is no overlap with the background. With these weak annotations, only 0.02% of the pixels in the dataset have ground-truth labels.

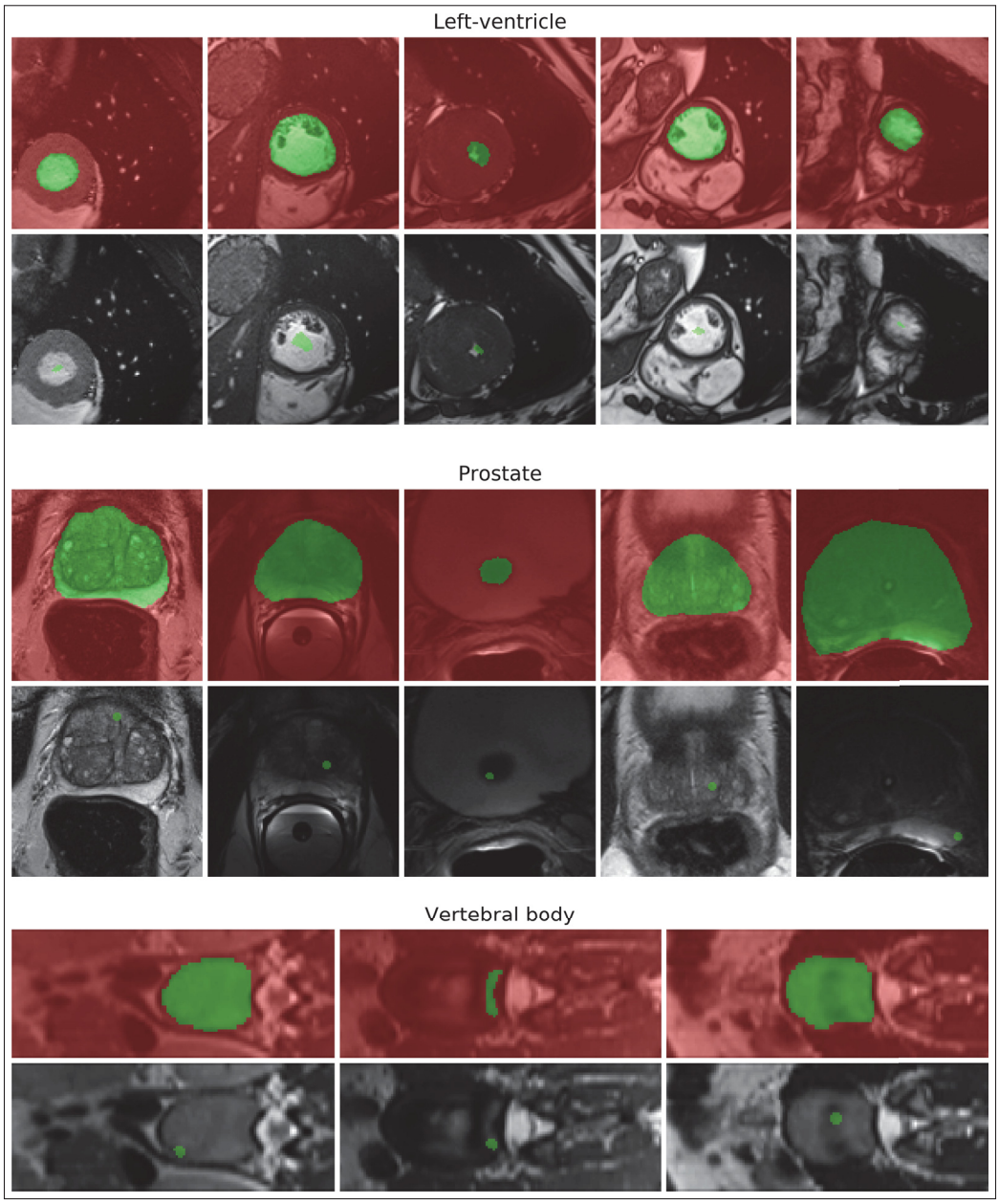


Figure 2.2 Examples of different levels of supervision. In the fully labeled images (*top*), all pixels are annotated, with red depicting the background and green the region of interest. In the weakly supervised cases (*bottom*), only the labels of the green pixels are known. The images were cropped for a better visualization of the weak labels. The original images are of size 256 $\times$ 256 pixels.

2.4.3 Different levels of supervision

Training models with diverse levels of supervision requires that appropriate objectives be defined for each case. In this section, we introduce the different models, each with different levels of supervision.

2.4.3.1 Baselines

We trained a segmentation network from weakly annotated images with no additional information, which served as a lower baseline. Training this model relies on minimizing the cross-entropy corresponding to the fraction of labeled pixels: $\mathcal{H}(S) = -\sum_{p\in\Omega_L} \log(S_p)$. In the following discussion of the experiments, we refer to this model as *partial cross-entropy (CE)*.

As an upper baseline, we resort to the fully-supervised setting, where class labels (foreground and background) are known for every pixel during training ($\Omega_L = \Omega$). This model is referred to as *fully-supervised*.

2.4.3.2 Size constraints

We incorporated information about the size of the target region during training, and optimized the partial cross-entropy loss subject to inequality constraints of the general form in Eq. (2.1). We trained several models using the same weakly annotated images but different constraint values.

Image tags bounds

Similar to MIL scenarios, we first used image-tag priors by enforcing the presence or absence of a the target in a given training image, as introduced earlier. This reduces to enforcing that the size of the predicted region is less or equal to 0 if the target is absent from the image, or

larger than 0 otherwise. To simplify the implementation, we can represent the constraints as:

$$a, b = \begin{cases} 1, |\Omega| & \text{if target is present } (\Omega_L \neq \emptyset) \\ 0, 0 & \text{otherwise} \end{cases}. \tag{2.5}$$

While being very coarse, these constraints convey relevant information about the target regions, which may be used to find common patterns in the case of region absence or presence.

Common bounds

The next level of supervision consists of using tighter bounds for the positive cases, instead of $(1, |\Omega|)$. To this end, the complete segmentation of a *single* patient is employed to compute the minimum and maximum size of the target region across all the slices. Then, we multiplied these minimum and maximum values by 0.9 and 1.1, respectively, to account for inter-patient variability. In this case, all the images containing the object of interest have the same lower and upper bounds. As an example, this results in the following values for the ACDC dataset:

$$a, b = \begin{cases} 60, 2000 & \text{if target is present } (\Omega_L \neq \emptyset) \\ 0, 0 & \text{otherwise} \end{cases}. \tag{2.6}$$

Individual bounds

With common bounds, the range of values for a given target may be very large. To investigate whether a more precise knowledge of the target is helpful, we also consider the use of individual bounds for each slice, based on the true size of the region:

$$\tau_Y = \sum_{p \in \Omega} Y_p,$$

with $Y = (Y_1, ..., Y_{|\Omega|}) \in \{0,1\}^{|\Omega|}$ denoting the full annotation of image I. As before, we introduce some uncertainty on the target size, and multiply τ_Y by the same lower and upper factors, resulting in the following bounds:

$$a, b = \begin{cases} 0.9\tau_Y, 1.1\tau_Y & \text{if target is present } (\Omega_L \neq \emptyset) \\ 0, 0 & \text{otherwise} \end{cases} . \tag{2.7}$$

2.4.3.3 Hybrid training

We also investigate whether combining our proposed weak supervision approach with fully annotated images during the training leads to performance improvements. For this purpose, considering we have a training set of m weakly annotated images, we replace n ($n < m$) among these by their fully annotated counterparts. Thus, the training amounts to minimizing the cross-entropy loss for the n fully annotated images, along with the partial cross-entropy constrained with common size bounds for the remaining $m - n$ weakly labeled images. To examine the positive effect of size constraints in this scenario (referred to as *Hybrid*), we compare the results to a network trained with the n fully annotated images (without constraints).

2.4.4 Constraining a 3D volume

We can extend our formulation to constrain a 3D volume as follows:

$$\sum_{S \in \mathrm{B}} \mathcal{H}(S) + \lambda C(V_\mathrm{B}), \qquad \text{with} \quad V_\mathrm{B} = \sum_{S \in \mathrm{B}} V_S$$

where V_{B} denotes the target-region volume, $\mathrm{B} = ((Y^1, S^1), ..., (Y^{|B|}, S^{|B|}))$ denotes a training batch containing all the 2D slices of the 3D volume[7], and the 3D constraints are now given by:

$$a, b = 0.9\tau_{\mathrm{B}}, 1.1\tau_{\mathrm{B}}, \qquad \text{with} \quad \tau_{\mathrm{B}} = \sum_{Y \in \mathrm{B}} \tau_Y$$

Notice that, with constraints on the whole 3D volume, we have less supervision than the 2D scenarios from 2.4.3.2, where all the 2D slices have independent supervision (e.g., the image tags).

2.4.5 Training and implementation details

For the experiments on the left-ventricle and vertebral-body datasets, we used ENet Paszke, Chaurasia, Kim & Culurciello (2016), as it has shown a good trade-off between accuracy and inference time. Due to the higher difficulty of the prostate segmentation task, we employed a fully residual version of U-Net Ronneberger *et al.* (2015), similar to Quan, Hildebrand & Jeong (2016).

For the three datasets, we trained the networks from scratch using the Adam optimizer and an initial learning rate of 5×10^{-4} that we decreased by a factor of 2 if the performances on the validation set did not improve over 20 epochs. All the 3D volumes were sliced into 256×256 pixels images, and zero-padded when needed. Batch sizes were equal to 1, 4, and 20 for the left-ventricle, prostate and vertebral body, respectively. Those values were not tuned for optimal performances, but to speed-up experiments when enough data were available. The weight of our loss in (2.2) was empirically set to 1×10^{-2}. Due to the difficulty of the task, data augmentation was used for the prostate dataset, where we generated 4 copies of each training image using random mirroring, flipping and rotation.

[7]For readability, we simplify a batch as a list of labels Y and associated predictions S.

All our tests were implemented in Pytorch Paszke, Gross, Chintala, Chanan, Yang, DeVito, Lin, Desmaison, Antiga & Lerer (2017). We ran the experiments on a machine equipped with a NVIDIA GTX 1080 Ti GPU (11GBs of video memory), AMD Ryzen 1700X CPU and 32GBs of memory. The code is available at https://github.com/LIVIAETS/SizeLoss_WSS. We used the common Dice similarity coefficient (DSC) to evaluate the segmentation performance of trained models.

Modification and tweaks for Lagrangian proposals

For a fair comparison, we re-implemented the Lagrangian-proposal method of Pathak et al. Pathak *et al.* (2015a) in PyTorch, to take advantage of GPU capabilities and avoid costly transfers between GPU and CPU. Lagrangian proposals reuse the same network and loss function as the fully-supervised setting. At each iteration, the method alternates between two steps. First, it synthesizes a ground truth $\tilde{Y}$ with projected gradient ascent (PGA) over the dual variables, with the network parameters fixed. Then, for fixed $\tilde{Y}$, the cross-entropy between $\tilde{Y}$ and S is optimized as in standard fully-supervised CNN training. The learning rate used for this PGA was set experimentally to 5×10^{-5}, as sub-optimal values lead to numerical errors. We found that limiting the number of iterations for the PGA to 500 (instead of the original 3000) saved time without affecting the results. We also introduced an early stopping mechanism into the PGA in the case of convergence, to improve speed without impacting the results (a comparison can be found in Table 2.5). The constraints of the form $0 \leq V_S \leq 0$ required specific care, as the formulation from Pathak *et al.* (2015a) is not designed to work on equalities, unlike our penalty approach, which systematically handles equality constraints when $a = b$. In this case, the bounds for Pathak *et al.* (2015a) were modified to $-1 \leq V_S \leq 0$.

2.5 Results

To validate the proposed approach, we first performed a series of experiments focusing on LV segmentation. In Sec. 2.5.1, the impact of including size constraints is evaluated using our direct penalty. We further compare to the Lagrangian-proposal method in Pathak *et al.*

(2015a), showing that our simple method yields substantial improvements over Pathak *et al.* (2015a) in the same weakly supervised settings. We also provide the results for several degrees of supervision, including hybrid and fully supervised learning in Sec. 2.5.2. Then, to show the wide applicability of the proposed constrained loss, results are reported for two other applications in Sec. 2.5.3: MR-T2 vertebral body segmentation and prostate segmentation task. We further provide qualitative results for the three applications in Sec. 2.5.4. In Sec. 2.5.5, we investigate the sensitivity of the proposed loss to both the lower and upper bounds. Finally, the efficiency of different learning strategies are compared (Sec. 2.5.6), showing that our direct constrained-CNN loss does not add to the training time, unlike the Lagrangian-proposal method in Pathak *et al.* (2015a).

2.5.1 Weakly supervised segmentation with size constraints

2D segmentation

Table 2.1 reports the results on the left-ventricle validation set for all the models trained with both the Lagrangian proposals in Pathak *et al.* (2015a) and our direct loss. As expected, using the partial cross entropy with a fraction of the labeled pixels yielded poor results, with a mean DSC less than 15%. Enforcing the image-tag constraints, as in the MIL scenarios, increased substantially the DSC to a value of 0.7924. Using common bounds increased the results marginally in this case, slightly increasing the mean Dice value by 1%. The Lagrangian proposal Pathak *et al.* (2015a) reaches similar results, albeit slightly lower and much more unstable than our penalty approach (see Figure 2.3).

The difference in performance is more pronounced when we employ individual bounds instead. In this setting, our method achieves a DSC of 0.8708, only 2% lower than full supervision. However, the Lagrangian-proposal method achieves a performance similar to using common (loose) bounds, suggesting that it is not able to make use of this extra, more precise information. This can be explained by its proposal-generation method, which tends to reinforce early mistakes (especially when training from scratch): the network is trained with conflicting information—

i.e., similar-looking patches are both foreground and background according the the synthetic ground truth—and is not able to recover from those initial mis-classifications.

3D segmentation

Constraining the size of the 3D volume of the target region also shows the benefit of our penalty approach, yielding a mean DSC of 0.8580. Recall that, here, we are using less supervision than the 2D case. Since we do not use tag information in this case, these results suggest that only a fraction of all the slices may be used when creating the labels, allowing annotators to scribble the 3D image directly instead of going through all the 2D slices one by one.

Table 2.1 Left-ventricle segmentation results with different levels of supervision. Bold font highlights the best weakly supervised setting.

	Model	Method	DSC (Val)
Weakly supervised	Partial CE		0.1497
	CE + Tags	Lagrangian Proposals Pathak *et al.* (2015a)	0.7707
	Partial CE + Tags	Direct loss (Ours)	0.7924
	CE + Tags + Size*	Lagrangian Proposals Pathak *et al.* (2015a)	0.7854
	Partial CE + Tags + Size*	Direct loss (Ours)	0.8004
	CE + Tags + Size**	Lagrangian Proposals Pathak *et al.* (2015a)	0.7900
	Partial CE + Tags + Size**	Direct loss (Ours)	**0.8708**
	CE + 3D Size**	Lagrangian Proposals Pathak *et al.* (2015a)	N/A
	Partial CE + 3D Size**	Direct loss (Ours)	0.8580
Fully supervised	Cross-entropy		0.8872

*Common bounds / ** Individual bounds

2.5.2 Hybrid training: mixing fully and weakly annotated images

Table 2.2 and Figure 2.4 summarize the results obtained when combining weak and full supervision. First, and as expected, we can observe that adding n fully annotated images to the training set (Hybrid_n) improves the performances in comparison to the model trained solely with the weakly annotated images, i.e., Weak_All. Particularly, the DSC increases by 4%,5% and 6%

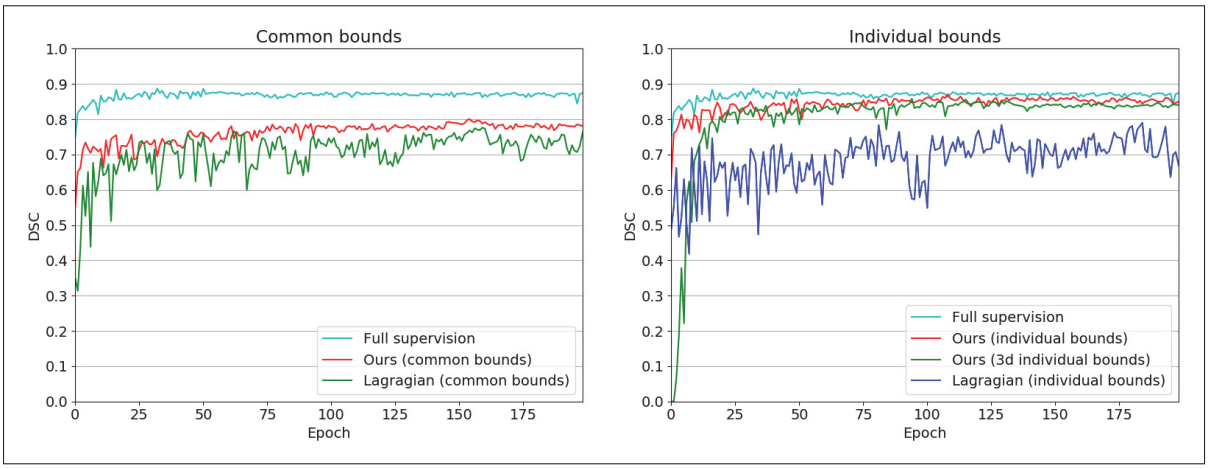


Figure 2.3 Evolution of the DSC during training for the left-ventricle validation set, including the weakly supervised learning models and different strategies analyzed, with also the full-supervision setting. As tags and common bounds achieve similar results, we plot only common bounds for better readability.

when n is equal to 5,10 and 25, respectively, approaching the full-supervision performance with only 25% of the fully labeled images.

Nevertheless, it is more interesting to see the impact of adding weakly annotated images (i.e., Hybrid_n) to a model trained solely with fully labeled images (i.e., Full_n). From the results, we can observe that adding weakly annotated images to the training set significantly increases the performance, particularly when the amount of fully annotated images (i.e., n) is limited. For instance, in the case of n equal to 5, adding weakly annotated images enhanced the performance by more than 30% in comparison to full supervision with n equal to 5. Despite the fact that this gap decreases with the number of fully annotated images, the difference between both settings (i.e., Full and Hybrid) remains significant. More interestingly, training the same model with a high amount of weakly annotated images and no or a very reduced set of fully labeled images (for example Weak_All or Hybrid_5) achieves better performances than employing datasets with much higher numbers of fully labeled images, e.g, Full_25.

These results suggest that a good strategy when annotating a new dataset might be to start with weak labels for all the images, and progressively complete full annotations, should ressources become available.

Table 2.2 Ablation study on the amounts of fully and weakly labeled data. We report the mean DSC of all the testing cases, for all the settings and using the same architecture.

Name	Training approach	# Fully/Weakly annotated images	DSC
Weak_All	Weak supervision*	0/150	0.8004
Full_5	Full supervision	5/0	0.5434
Hybrid_5	Full + weak supervision*	5/145	0.8386
Full_10	Full supervision	10/0	0.6004
Hybrid_10	Full + weak supervision*	10/140	0.8475
Full_25	Full supervision	25/0	0.7680
Hybrid_25	Full + weak supervision*	25/125	0.8641
Full_All	Full supervision	150/0	0.8872

*Common bounds

2.5.3 MR-T2 vertebral body and prostate segmentation

The results obtained for the vertebral-body dataset (Table 2.3) highlight well the differences in the performances of different levels of supervision. Using tag bounds produces a network that roughly locates the object of interest (DSC of 0.5597), but fails to identify its boundaries (as seen in Figure 2.6, *third column*). Employing the common size strategy achieves satisfactory results for the slices containing objects with a regular shape but still fails when more difficult/irregular targets are present, resulting in an overall improvement of DSC (0.7900). However, when using individual bounds, the network is able to satisfactory segment even the most difficult cases, obtaining a DSC of 0.8604, only 3% lower than full supervision.

For the prostate dataset, one can observe that common bounds still improve the results obtained with tags (+3%), but the difference is much smaller than the case of vertebral-body segmentation. Using individual bounds increases the DSC value by 10%, reaching 0.8298, a behaviour similar

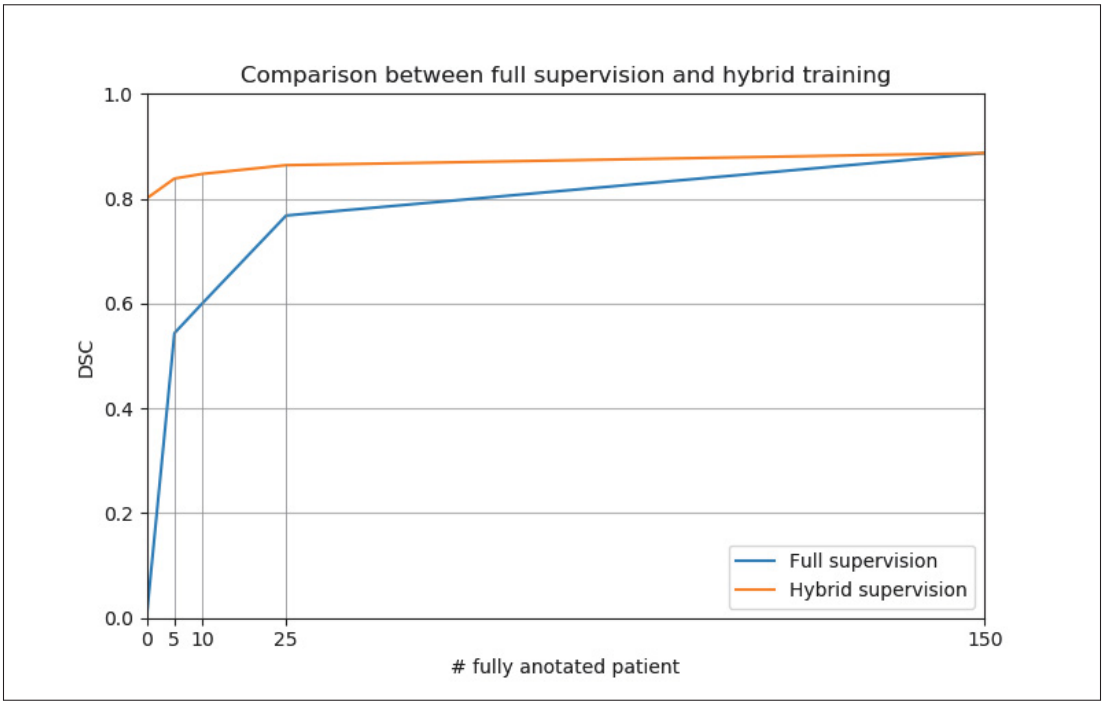


Figure 2.4 Mean DSC values over the number of fully annotated patients employed for training.

Table 2.3 Mean Dice scores (DSC) for several degrees of supervision, using the vertebral-body and prostate validation sets. Bold font indicates the best weakly supervised setting for each data set.

Method	Vertebral body DSC	Prostate DSC
Partial CE	0.1155	0.0320
Partial CE + Tags	0.5597	0.6911
Partial CE + Tags + Common size	0.7900	0.7214
Partial CE + Tags + Individual size	**0.8604**	**0.8298**
Fully supervised	0.8999	0.8911

to what we observed earlier for the other datasets. Nevertheless, in this case, the gap between full and weak supervision with individual bounds constraints is larger than what we obtained for the other datasets.

2.5.4 Qualitative results

To gain some intuition on different learning strategies and their impact on the segmentation, we visualize some results sampled from the validation sets in Fig. 2.5, 2.6 and 2.7 for LV, VB and prostate, respectively.

LV segmentation task

We compare 4 methods to the ground truth: full supervision, Lagrangian proposals Pathak *et al.* (2015a) with common bounds, direct loss with common bounds and direct loss with individual bounds. We can see that, for the easy cases containing regular shapes and visible borders, all methods obtain similar results. However, the methods employing common bounds can easily over-segment the object, especially when their size is considerably smaller; see for example the last row in Figure 2.5. Since individual bounds are specific to each image, a model trained with these bounds will not suffer in such cases, as shown in the figure.

Vertebral-body segmentation task

In this case, we visualize the results of full supervision, tag bounds, common bounds and individual bounds. In line with results reported in Table 2.3, we can visually observe the gap in performances between each setting, which clearly highlights the impact of the different values of the bounds during the optimization process. Using only tags, the network learn to roughly locate the object. When size bounds are included as common size information, the network is able to somehow learn the boundaries, but only for object shapes that are within the standard variability of a typical vertebral body shape. As it can be observed, the model fails to segment the unusual shapes (last three rows in Figure 2.6). Lastly, a network trained with individual sizes is able to better handle those cases, while still being imprecise on some regions.

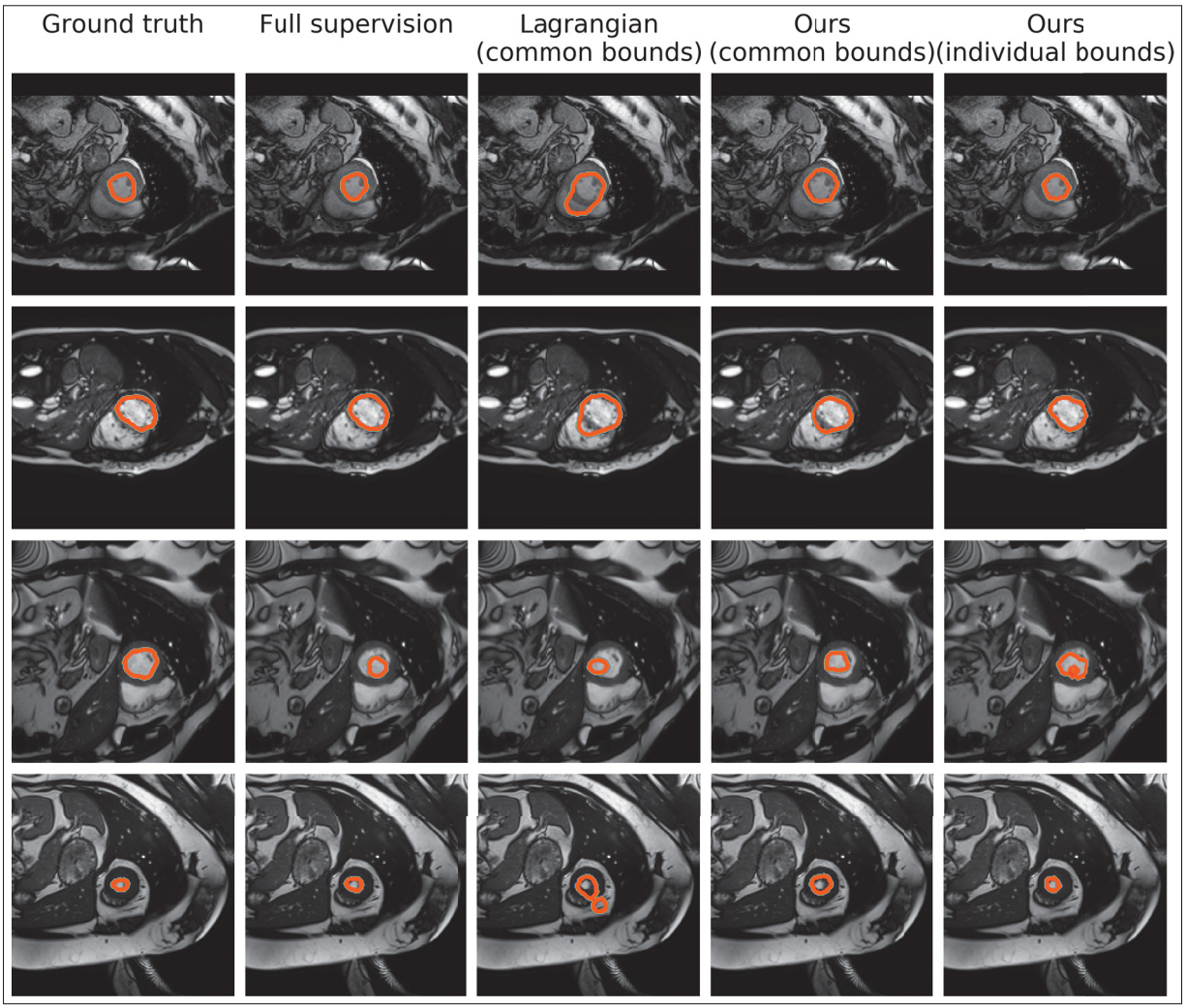


Figure 2.5 Qualitative comparison of the different methods using examples from the LV dataset. Each column depicts segmentations obtained by different methods, whereas each row represents a 2D slice from different scans (Best viewed in colors).

Prostate segmentation task

As in the previous case, we depict the results of full supervision, tag bounds, common bounds and individual bounds. Both the tags and common bounds locate the object in a similar fashion, but both have difficulties finding a precise contour, typically over-segmenting the target region. This is easily explained by the variability of the organ and the very low contrast on some images. As shown in the last column, using individual bounds greatly improves the results.

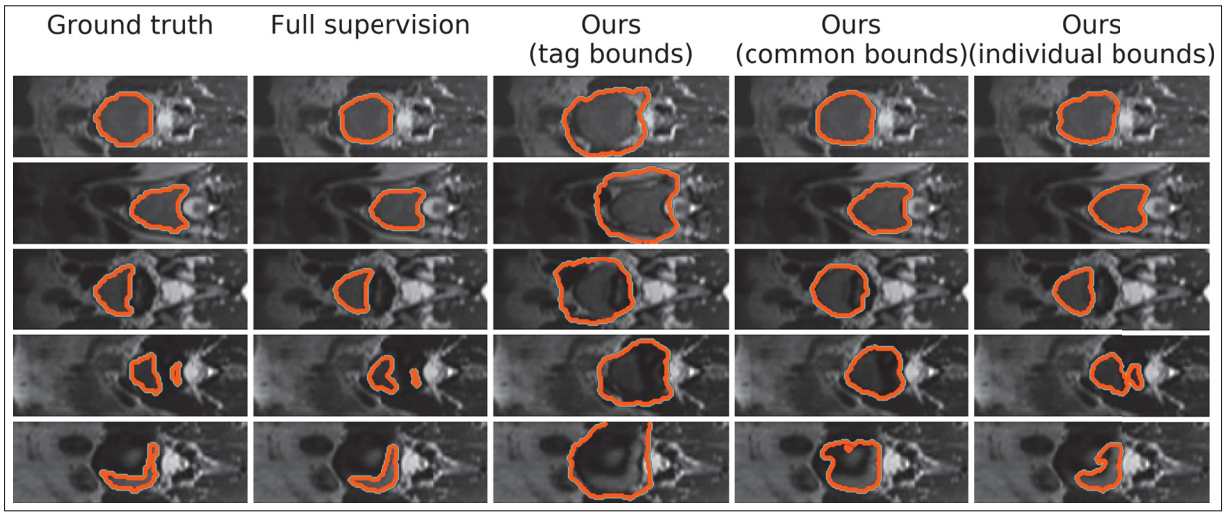


Figure 2.6 Qualitative comparison using examples from the VB dataset. Each column depicts segmentations obtained by different levels of supervision, whereas each row represents a 2D slice from different scans (Best viewed in colors).

2.5.5 Sensitivity to the constraint boundaries

In this section, an ablation study is performed on the lower and upper bounds when using common bounds, and investigate their effect on the performance on the vertebral-body segmentation task. Results for different bounds are reported in Table 2.4. It can be observed that progressively increasing the value of the upper bound decreases the performance. For example, the DSC drops by nearly 12% and 16% when the upper bound is increased by a factor of 5 and 10, respectively. Decreasing the lower bound from 80 to 0 has a much smaller impact than the upper bound, with a constant drop of less than 1%. These findings are aligned with visual predictions illustrated in Figure 2.6. While a network trained only with tag bounds tends to over-segment, adding an upper bound easily fixes the over-segmentation, correcting most of the mistakes. Nevertheless, for the same reason, i.e., over-segmentation, very few slices benefit from a lower bound.

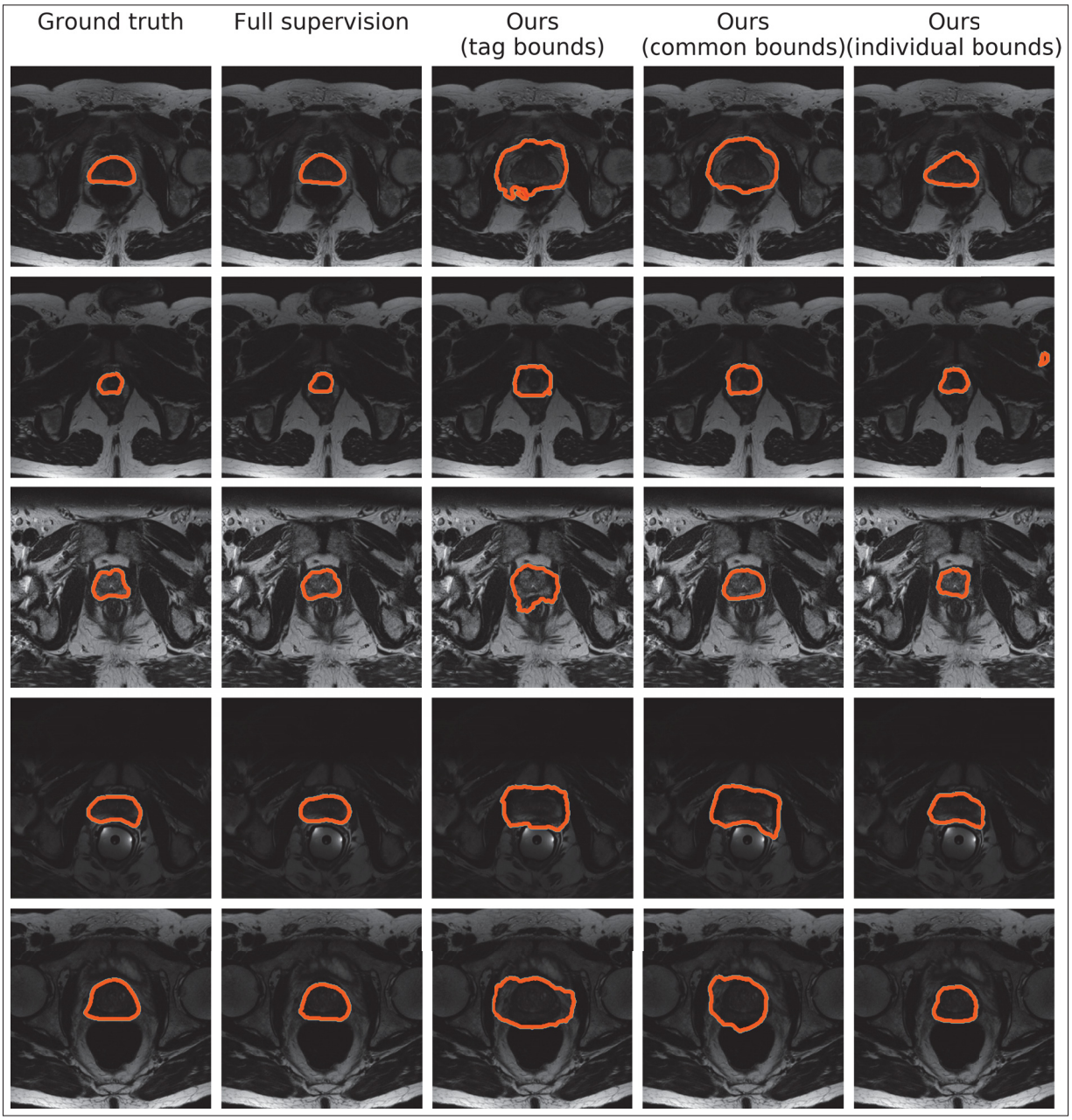


Figure 2.7 Qualitative comparison of the different levels of supervision. Each row represents a 2D slice from different scans. (Best viewed in colors)

2.5.6 Efficiency

In this section, we compare the several learning approaches in terms of efficiency (Table 2.5). Both the weakly supervised partial cross-entropy and the fully supervised model need

Table 2.4 Ablation study on the lower and upper bounds of the size constraint using the vertebral body dataset.

	Bounds		Mean DSC
Model	Lower (a)	Upper (b)	
Weak Sup. w/ direct loss	$0.9\tau_Y$	$1.1\tau_Y$	0.8604
Weak Sup. w/ direct loss	80	1100	0.7900
Weak Sup. w/ direct loss	80	5000	0.6704
Weak Sup. w/ direct loss	80	10000	0.6349
Weak Sup. w/ direct loss	0	1100	0.7820
Weak Sup. w/ direct loss	0	5000	0.6694
Weak Sup. w/ direct loss	0	10000	0.6255
Weak Sup. w/ direct loss	0	65536	0.5597

to compute only one loss per pass. This is reflected in the lowest training times reported in the table. Including the size loss does not add to the computational time, as can be seen in these results. As expected, the iterative process introduced by Pathak *et al.* (2015a) at each forward pass adds a significant overhead during training. To generate their synthetic ground truth, they need to optimize the Lagrangian function with respect to its dual variables (Lagrange multipliers of the constraints), which requires alternating between training a CNN and Lagrangian-dual optimization. Even in the simplest optimization case (with only one constraint), where optimization over the dual variable converges rapidly, their method remains two times slower than ours. Without the early stopping criteria that we introduced, the overhead is much worse with a six-fold slowdown. In addition, their method also slows down when more constraints are added. This is particularly significant when there is many classes to constrain/supervise.

Generating the proposals at each iteration also makes it much more difficult to build an efficient implementation for larger batch sizes. One either needs to generate them one by one (so the overhead grows linearly with the batch size) or try to perform it in parallel. However, due to the nature of GPU design, the parallel Lagrangian optimizations will slow each other down, meaning that there may be limited improvements over a sequential generation. In some cases it may be faster to perform it on CPU (where the cores can truly perform independent tasks

in parallel), at the cost of slow transfers between GPU and CPU. The optimal strategy would depend on the batch size and the host machine, especially its available GPU, number of CPU cores and bus frequency.

Table 2.5 Training times for the diverse supervised learning strategies with a batch size of 1, using tags and size constraints.

Method	Training time (ms/batch)
Partial CE	112
Direct loss (1 bound)	113
Direct loss (2 bounds)	113
Lagrangian proposals (1 bound)	610
Lagrangian proposals (2 bounds)	675
Lagrangian proposals (1 bound), w/ early stop	221
Lagrangian proposals (2 bounds), w/ early stop	220
Fully supervised	112

2.6 Discussion

We have presented a method to train deep CNNs with linear constraints in weakly supervised segmentation. To this end, we introduce a differentiable term, which enforces inequality constraints directly in the loss function, avoiding expensive Lagrangian dual iterates and proposal generation.

Results have demonstrated that leveraging the power of weakly annotated data with the proposed direct size loss is highly beneficial, particularly when limited full annotated data is available. This could be explained by the fact that the network is already trained properly when a large fully annotated training set is available, which is in line with the values reported in Table 2.2. Similar findings were reported in Bai, Oktay, Sinclair, Suzuki, Rajchl, Tarroni, Glocker, King, Matthews & Rueckert (2017); Zhou, Wang, Tang, Bai, Shen, Fishman & Yuille (2019c), where authors exhibited an increased of performance when including non-annotated images in a semi-supervised setting. This suggests that including more unlabelled or weakly labelled data can potentially lead to significantly improvements in performance.

Findings from experiments across different segmentation tasks indicate that highly competitive performance can be obtained with a rough estimation of the target size. This is especially the case on well structured problems where the size and/or shape of the object remains consistent across subjects. If more precise size bounds are provided, the proposed approach is able to reach performances close to full supervision, even when the size and shape variability across subjects is large. For difficult tasks, where the gap between our approach and full supervision is larger, such as prostate segmentation, including an unsupervised regularization loss Tang *et al.* (2018a,1) to encourage pairwise consistencies between pixels may boost the performance of the proposed strategy. A noteworthy point is the robustness of our method to the weak-label generation. While the weak labels were generated from a ground-truth erosion for the first dataset, with seeds always in the center of the target region, they were randomly generated and placed for the other two datasets. Thus, the results showed consistency in the behaviour of the different methods, regardless of the strategy used.

Even though the proposed method has been shown to provide good generalization capabilities across three different applications, the segmentation of images with severe abnormalities, whose sizes largely differ from those seen in the training set, has not been assessed. Nevertheless, the ablation study performed on the values of the size bounds, and the results obtained with common bound sizes suggest that the proposed approach may perform satisfactorily in the presence of these severe abnormalities, by simply increasing the upper bound value. In addition, if a greater 'precise' estimation of the abnormality size is given, our proposed loss may improve segmentation performance, as demonstrated by the results achieved by the individual bounds strategy. It is important to note that, even in the case of full supervision, if a new testing image contains a severe abnormality much larger than the objects seen during the training phase, the network will likely to poorly segment the region of interest.

Our framework can be easily extended to other non-linear (fractional) constraints, e.g., invariant shape moments Klodt & Cremers (2011) or other statistics such as the mean of intensities within the target regions Lim *et al.* (2014). For instance, a normalized (scale invariant) shape moment of a target region can be directly expressed in term of network outputs using the following

general fractional form:

$$F_S = \frac{\sum_{p\in\Omega} f_p S_p}{\sum_{p\in\Omega} S_p} \tag{2.8}$$

where f_p is a unary potential expressed in term of exponents of pixel/voxel coordinates. For example, the coordinates of the center of mass of the target region are particular cases of (2.8) and correspond to first-order scale-invariant shape moments. In this case, potentials f_p correspond to pixel coordinates. Now, assume a weak-supervision scenario in which we have a rough localization of the centroid of the target region. In this case, instead of a constraint on size representation V_S as in Eq. (2.3), one can use a cue on centroid as follows: $a \leq F_S \leq b$. This can be embedded as a direct loss using differentiable penalty $C(F_S)$. Of course, here, F_S is a non-linear fractional term unlike region size. Therefore, in future work, it would be interesting to examine the behaviour of such fractional terms for constraining deep CNNs with a penalty approach. Finally, it is worth noting that the general form in Eq. (2.8) is not confined to shape moments. For instance, the image (intensity) statistics within the target region, such as the mean[8], follow the same general form in (2.8). Therefore, a similar approach could be used in cases where we have prior knowledge on such image statistics.

Our direct penalty-based approach for inequality constraints yields a considerable increase in performance with respect to to Lagrangian-dual optimization Pathak *et al.* (2015a), while being faster and more stable. We hypothesize that this is due, in part, to the *interplay* between stochastic optimization (e.g., stochastic gradient descent) for the primal and the iterates/projections for the Lagrangian dual[9]. Such dual iterates/projections are basic (non-stochastic) gradient methods for handling the constraints. Basic gradient methods have well-known issues with deep networks, e.g., they are sensitive to the learning rate and prone to weak local minima. Therefore, the dual part in Lagrangian optimization might obstruct the practical and theoretical benefits of stochastic optimization (e.g., speed and strong generalization performance), which are widely established for unconstrained deep network losses Hardt *et al.* (2016). Our penalty-

[8]Notice that the mean of intensity within the target region can be represented with network output using general form (2.8), with f_p corresponding to the intensity of pixel p

[9]In fact, a similar hypothesis was made in Márquez-Neila *et al.* (2017) to explain the negative results of Lagrangian optimization in the case of equality constraints.

based approach transforms a constrained problem into an unconstrained loss, thereby handling the constraints fully within stochastic optimization and avoiding completely the dual steps. While penalty-based approaches do not guarantee constraint satisfaction, our work showed that they can be extremely useful in the context of constrained CNN segmentation.

2.7 Conclusion

In this paper, a novel loss function is present for weakly supervised image segmentation, which, despite its simplicity, performs significantly better than Lagrangian optimization for this task. We achieve results close to full supervision by annotating only a small fraction of the pixels, across three different tasks, and with negligible computation overhead. While our experiments focused on basic linear constraints such as the target-region size and image tags, our direct constrained-CNN loss can be easily extended to other non-linear constraints, e.g., invariant shape moments Klodt & Cremers (2011) or other region statistics Lim *et al.* (2014). Therefore, it has the potential to close the gap between weakly and fully supervised learning in semantic medical image segmentation.

CHAPTER 3

CONSTRAINED DEEP NETWORKS: LAGRANGIAN OPTIMIZATION VIA LOG-BARRIER EXTENSIONS

Hoel Kervadec[a], Jose Dolz[b], Jing Yuan[c], Christian Desrosiers[b], Eric Granger[a], Ismail Ben Ayed[a]

[a] Département de génie des systèmes, ÉTS Montréal, QC, Canada,
[b] Département de génie logiciel et des TI, ÉTS Montréal, QC, Canada,
[c] School of Math. and Stat., Xidian University, China

Paper submitted to Transactions of Pattern Analysis and Machine Intelligence (TPAMI).

Abstract

This study investigates the optimization aspects of imposing hard inequality constraints on the outputs of CNNs. In the context of deep networks, constraints are commonly handled with *penalties* for their simplicity, and despite their well-known limitations. Lagrangian-dual optimization has been largely avoided, except for a few recent works, mainly due to the computational complexity and stability/convergence issues caused by alternating *explicit* dual updates/projections and stochastic optimization. Several studies showed that, surprisingly for deep CNNs, the theoretical and practical advantages of Lagrangian optimization over penalties do not materialize in practice. We propose *log-barrier extensions*, which approximate Lagrangian optimization of constrained-CNN problems with a sequence of unconstrained losses. Unlike standard interior-point and log-barrier methods, our formulation does not need an initial feasible solution. Furthermore, we provide a new technical result, which shows that the proposed extensions yield an upper bound on the duality gap. This generalizes the duality-gap result of standard log-barriers, yielding sub-optimality certificates for feasible solutions. While sub-optimality is not guaranteed for non-convex problems, our result shows that log-barrier extensions are a principled way to approximate Lagrangian optimization for constrained CNNs via *implicit* dual variables. We report comprehensive weakly supervised segmentation experiments, with various constraints, showing that our formulation outperforms substantially the existing constrained-CNN methods,

both in terms of accuracy, constraint satisfaction and training stability, more so when dealing with a large number of constraints.

3.1 Introduction

Deep convolutional neural networks (CNNs) are dominating in most visual recognition problems and applications, including semantic segmentation, action recognition, object detection and pose estimation, among many others. In a standard setting, CNNs are trained with abundant labeled data without any additional prior knowledge about the task (apart from model architecture and loss). However, in a breadth of learning problems, for example, semi- and weakly-supervised learning, structured prediction or multi-task learning, a set of natural prior-knowledge constraints is available. Such additional knowledge can come, for example, from domain experts.

In the semi-supervised setting, for instance, several recent works Kervadec, Dolz, Tang, Granger, Boykov & Ayed (2019b); Nandwani *et al.* (2019); Zhou, Li, Bai, Wang, Chen, Han, Fishman & Yuille (2019a) showed that imposing domain-specific knowledge on the network's predictions at unlableled data points acts as a powerful regularizer, boosting significantly the performances when the amount of labeled data is limited. For instance, the recent semi-supervised semantic segmentation works in Kervadec *et al.* (2019b); Zhou *et al.* (2019a) added priors on the sizes of the target regions, achieving good performances with only fractions of full-supervision labels. Such prior-knowledge constraints are highly relevant in medical imaging Litjens *et al.* (2017), and can mitigate the lack of full annotations[1]. Similar experimental observations were made in other application areas of semi-supervised learning. For example, in natural language processing, the authors of Nandwani *et al.* (2019), among other recent studies, showed that embedding prior-knowledge constraints on unlabled data can yield significant boosts in performances. 3D human pose estimation from a single view Márquez-Neila *et al.*

[1]In semantic segmentation, for instance, full supervision involves annotating all the pixels in each training image, a problem further amplified when such annotations require expert knowledge or involves volumetric data, as is the case in medical imaging.

(2017) is another application example where task-specific prior constraints arise naturally, e.g., symmetry constraints encode the prior that the two arms should have the same length.

Imposing prior knowledge in the form of hard constraints on the output of modern deep CNNs with large numbers of parameters is still in a nascent stage, despite its clear benefits and breadth of applications. As discussed in several recent works Kervadec *et al.* (2019b); Márquez-Neila *et al.* (2017); Nandwani *et al.* (2019); Pathak *et al.* (2015a); Ravi *et al.* (2019); Zhou *et al.* (2019a), there are several challenges that arise from optimization perspectives, particularly when dealing with deep networks involving millions of parameters.

3.1.1 General Constrained Formulation

Let $\mathcal{D} = \{I^1, ..., I^N\}$ denotes a partially labeled set of N training images, and $S_{\boldsymbol{\theta}} = \{s^1_{\boldsymbol{\theta}}, ..., s^N_{\boldsymbol{\theta}}\}$ denotes the associated predicted networks outputs in the form of softmax probabilities, for both unlabeled and labeled data points, with $\boldsymbol{\theta}$ the neural-network weights. These could be class probabilities or dense pixel-wise probabilities in the case of semantic image segmentation. We address constrained problems of the following general form:

$$\begin{aligned} \min_{\boldsymbol{\theta}} \quad & \mathcal{E}(\boldsymbol{\theta}) \\ s.t. \quad & f_1(s^n_{\boldsymbol{\theta}}) \leq 0, \quad n = 1, \dots N \\ & \dots \\ & f_P(s^n_{\boldsymbol{\theta}}) \leq 0, \quad n = 1, \dots N \end{aligned} \tag{3.1}$$

where $\mathcal{E}(\boldsymbol{\theta})$ is some standard loss over the set of labeled data points, e.g., cross-entropy, and f_i are a series of derivable function whose output we want to constraint for each data point n. Inequality constraints of the general form in (3.1) can embed very useful prior knowledge on the network's predictions for unlabeled pixels. Assume, for instance, in the case of image segmentation, that we have prior knowledge about the size of the target region (i.e., class) k. Such a knowledge can be in the form of lower or upper bounds on region size, which is common in medical image segmentation problems Gorelick *et al.* (2013); Kervadec *et al.*

(2019b); Niethammer & Zach (2013). In this case, $I^n : \Omega \subset \mathbb{R}^2 \rightarrow \mathbb{R}$ could a partially labeled or unlabeled image, with Ω the spatial support of the image, and $s^n_{\boldsymbol{\theta}} \in [0,1]^{K \times |\Omega|}$ is the predicted mask. This matrix contains the softmax probabilities for each pixel $p \in \Omega$ and each class k, which we denotes $s^n_{k,p,\boldsymbol{\theta}}$. A constraint in the form of $f_i(s^n_{\boldsymbol{\theta}}) = \sum_{p \in \Omega} s^n_{k,p,\boldsymbol{\theta}} - a$ enforces an upper limit a on the size of target region k. Such constraints could be also very useful for imposing tightness priors in the context of box-based weakly supervised segmentation Hsu, Hsu, Tsai, Lin & Chuang (2019). There exist many other application areas where constraints arise naturally, including in natural language processing (NLP), where prior knowledge on the language structure exists and could be incorporated into the training with constraints on network softmax predictions Nandwani *et al.* (2019).

3.1.2 Related Works and Challenges in Constrained CNN Optimization

As pointed out in several recent studies Kervadec *et al.* (2019b); Márquez-Neila *et al.* (2017); Nandwani *et al.* (2019); Pathak *et al.* (2015a); Ravi *et al.* (2019); Zhou *et al.* (2019a), imposing hard constraints on deep CNNs involving millions of trainable parameters is challenging. This is the case of problem (3.1), even when the constraints are convex with respect to the outputs of the network. In optimization, a standard way to handle constraints is to solve the Lagrangian primal and dual problems in an alternating scheme Boyd & Vandenberghe (2004). For (3.1), this corresponds to alternating the optimization of a CNN for the primal with stochastic optimization, e.g., SGD, and projected gradient-ascent iterates for the dual. However, despite the clear benefits of imposing global constraints on CNNs, such a standard Lagrangian-dual optimization is mostly avoided in modern deep networks. As discussed recently in Márquez-Neila *et al.* (2017); Pathak *et al.* (2015a); Ravi *et al.* (2019), this might be explained by the computational complexity and stability/convergence issues caused by alternating between stochastic optimization and dual updates/projections.

In standard Lagrangian-dual optimization, an unconstrained problem needs to be solved after each iterative dual step. This is not feasible for deep CNNs, however, as it would require re-training the network at each step. To avoid this problem, Pathak et al. Pathak *et al.* (2015a)

introduced a latent distribution, and minimized a KL divergence so that the CNN output matches this distribution as closely as possible. Since the network's output is not directly coupled with constraints, its parameters can be optimized using standard techniques like SGD. While this strategy enabled adding inequality constraints in weakly supervised segmentation, it is limited to linear constraints. Moreover, the work in Márquez-Neila *et al.* (2017) imposed hard equality constraints on 3D human pose estimation. To alleviate the ensuing computational complexity, they used a Kyrlov sub-space approach, limiting the solver to a randomly selected subset of constraints within each iteration. Therefore, constraints that are satisfied at one iteration may not be satisfied at the next, which might explain the negative results obtained in Márquez-Neila *et al.* (2017). In general, updating the network parameters and dual variables in an alternating fashion leads to a higher computational complexity than solving a loss function directly.

Another important difficulty in Lagrangian optimization is the interplay between stochastic optimization (e.g., SGD) for the primal and the iterates/projections for the dual. Basic gradient methods have well-known issues with deep networks, e.g., they are sensitive to the learning rate and prone to weak local minima. Therefore, the dual part in Lagrangian optimization might obstruct the practical and theoretical benefits of stochastic optimization (e.g., speed and strong generalization performance), which are widely established for unconstrained deep network losses Hardt *et al.* (2016). More importantly, solving the primal and dual separately may lead to instability during training or slow convergence, as shown recently in Kervadec *et al.* (2019b). To alleviate the instability caused by the dual part in Lagrangian optimization, Nandwani et al. Nandwani *et al.* (2019) introduced a ReLu modification of the Lagrangian term, avoiding completely the projection steps for the dual variables.

3.1.2.1 Penalty approaches

In the context of deep networks, "hard" inequality or equality constraints are typically handled in a "soft" manner by augmenting the loss with a *penalty* function He, Liu, Schwing & Peng (2017); Jia *et al.* (2017); Kervadec *et al.* (2019b). Such a penalty approach is a simple alternative to Lagrangian optimization, and is well-known in the general context of constrained

optimization; see Bertsekas (1995), Chapter 4. In general, penalty-based methods approximate a constrained minimization problem with an unconstrained one by adding a term (penalty) $\mathcal{P}(f_i(s_{\boldsymbol{\theta}}))$, which increases when constraint $f_i(s_{\boldsymbol{\theta}}) \leq 0$ is violated. By definition, a penalty $\mathcal{P}$ is a non-negative, continuous and differentiable function, which verifies: $\mathcal{P}(f_i(s_{\boldsymbol{\theta}})) = 0$ if and only if constraint $f_i(s_{\boldsymbol{\theta}}) \leq 0$ is satisfied. In semantic segmentation Kervadec *et al.* (2019b) and, more generally, in deep learning He *et al.* (2017), it is common to use a quadratic penalty for imposing an inequality constraint: $\mathcal{P}(f_i(s_{\boldsymbol{\theta}})) = [f_i(s_{\boldsymbol{\theta}})]_+^2$, where $[x]_+ = \max(0, x)$ denotes the rectifier function. Penalties are convenient for deep networks because they remove the requirement for explicit Lagrangian-dual optimization. The inequality constraints are fully handled within stochastic optimization, as in standard unconstrained losses, avoiding gradient ascent iterates/projections over the dual variables and reducing the computational load for training Kervadec *et al.* (2019b). However, this simplicity of penalty methods comes at a price. In fact, it is well known that penalties do not guarantee constraint satisfaction and require careful and *ad hoc* tuning of the relative importance (or weight) of each penalty term in the overall function. More importantly, in the case of several competing constraints, penalties do not act as *barriers* at the boundary of the feasible set (i.e., a satisfied constraint yields a null penalty and null gradient). As a result, a subset of constraints that are satisfied at one iteration may not be satisfied at the next. Take the case of two competing constraints f_1 and f_2 at the current iteration (assuming gradient-descent optimization), and suppose that f_1 is satisfied but f_2 is not. The gradient of a penalty $\mathcal{P}$ w.r.t the term of satisfied constraint f_1 is null, and the the penalty approach will focus solely on satisfying f_2. Therefore, due to a null gradient, there is nothing that prevents satisfied constraint f_1 from being violated. This could lead to oscillations between competing constraints during iterations, making the training unstable (we will give examples in the experiments).

3.1.2.2 Lagrangian approaches

3.1.2.2.1 Standard Lagrangian-dual optimization

Let us first examine standard Lagrangian optimization for problem (3.1):

$$\max_{\boldsymbol{\lambda}} \min_{\boldsymbol{\theta}} \quad \mathcal{L}(S_{\boldsymbol{\theta}}, \boldsymbol{\lambda}) = \mathcal{E}(\boldsymbol{\theta}) + \sum_{i=1}^{P} \sum_{n=1}^{N} \lambda_i^n f_i(s_{\boldsymbol{\theta}}^n) \tag{3.2}$$

$$s.t. \quad \boldsymbol{\lambda} \geq 0$$

where $\boldsymbol{\lambda} \in \mathbb{R}_+^{P \times N}$ is the dual variable (or Lagrange-multiplier) vector, with λ_i^n the multiplier associated with constraint $f_i(s_{\boldsymbol{\theta}}^n) \leq 0$. The dual function is the minimum value of Lagrangian (3.2) over $\boldsymbol{\theta}$: $g(\boldsymbol{\lambda}) = \min_{\boldsymbol{\theta}} \mathcal{L}(S_{\boldsymbol{\theta}}, \boldsymbol{\lambda})$. A standard Lagrangian would alternatively optimize w.r.t the network parameters $\boldsymbol{\theta}$ and dual variable λ.

Lagrangian optimization can deal with the difficulties of penalty methods, and has several well-known theoretical and practical advantages over penalty methods Fletcher (1987); Gill *et al.* (1981): it finds automatically the optimal weights of the constraints, acts as a barrier for satisfied constraints and guarantees constraint satisfaction when feasible solutions exist. Unfortunately, as pointed out recently in Kervadec *et al.* (2019b); Márquez-Neila *et al.* (2017), these advantages of Lagrangian optimization do not materialize in practice in the context of deep CNNs. Apart from the computational-feasibility aspects, which the recent works in Márquez-Neila *et al.* (2017); Pathak *et al.* (2015a) address to some extent with approximations, the performances of Lagrangian optimization are, surprisingly, below those obtained with simple, much less computationally intensive penalties Kervadec *et al.* (2019b); Márquez-Neila *et al.* (2017). This is, for instance, the case of the recent weakly supervised CNN semantic segmentation results in Kervadec *et al.* (2019b), which showed that a simple quadratic-penalty formulation of inequality constraints outperforms substantially the Lagrangian method in Pathak *et al.* (2015a). Also, the authors of Márquez-Neila *et al.* (2017) reported surprising results in the context of 3D human

pose estimation. In their case, replacing the equality constraints with simple quadratic penalties yielded better results than Lagrangian optimization.

3.1.2.2.2 ReLU Lagrangian modification Nandwani *et al.* (2019)

One of the main problems of the standard Lagrangian-dual optimization in deep CNNs is its instability due, in part to dual variables λ_i^n, which could remain positive while the constrained function f_i is satisfied ($f_i(s_{\boldsymbol{\theta}}^n) \leq 0$). Because of this, the SGD on $\boldsymbol{\theta}$ keeps minimizing the constrained term, although no modification should be made anymore. Nandwani et al. Nandwani *et al.* (2019) devised a trick to alleviate this problem, by putting the constrained function into a rectified linear unit first. They also regroup the constraints by function f_i, sharing the same λ_i for all samples of dataset $\mathcal{D}$[2]:

$$\max_{\boldsymbol{\lambda}} \min_{\boldsymbol{\theta}} \quad \mathcal{L}(S_{\boldsymbol{\theta}}, \boldsymbol{\lambda}) = \mathcal{E}(\boldsymbol{\theta}) + \sum_{i=1}^{P} \lambda_i \sum_{n=1}^{N} \max(0, f_i(s_{\boldsymbol{\theta}}^n)). \tag{3.3}$$

Since the gradient $\nabla\boldsymbol{\lambda}$ is always positive, $\boldsymbol{\lambda}$ can only increase over time.

3.1.2.2.3 Lagrangian with proposals Pathak *et al.* (2015a)

This is another approach, introduced by Pathak et al., to deal with the limitations of standard Lagrangian with deep neural networks, in the context of weakly supervised image segmentation Pathak *et al.* (2015a). We want the softmax probabilities $s_{\boldsymbol{\theta}}^n$ to match as closely as possible some binary labels $y^n \in \{0,1\}^{K\times|\Omega|}$ such as $\sum_k y_{k,p}^n = 1 \; \forall p \in \Omega$. Their first insight was to rewrite y^n as a continuous variable ($\in [0,1]^{K\times|\Omega|}$), and to introduce a latent variable $\tilde{y}^n \in [0,1]^{K\times|\Omega|}$ on

[2]This, we argue that this is unwarranted and may introduce several problems, as we explain in great details in our supplemental material 3.

which they imposed linear constraints:

$$\begin{aligned} \min_{\tilde{y},\boldsymbol{\theta}} \quad & \sum_n KL(\tilde{y}^n || s^n_{\boldsymbol{\theta}}) && \\ s.t. \quad & \tilde{y}^{n\top} a_1 - b_1 \preceq 0 && \forall n \in \mathcal{D}, \\ & \dots && \\ & \tilde{y}^{n\top} a_P - b_P \preceq 0 && \forall n \in \mathcal{D}, \\ & \mathbf{1}^\top \tilde{y}^n_p = 1 && \forall n \in \mathcal{D}, \forall p \in |\Omega|, \end{aligned} \tag{3.4}$$

where $a_1, ..., a_P \in \mathbb{R}^{|\Omega|}$ and $b_1, ..., b_P \in \mathbb{R}^K$. The corresponding Lagrangian is:

$$\begin{aligned} \max_{\boldsymbol{\lambda},\boldsymbol{\nu}} \min_{\tilde{y},\boldsymbol{\theta}} \quad & \sum_n \left(KL(\tilde{y}^n || s^n_{\boldsymbol{\theta}}) + \sum_{i=1}^{P} \lambda^n_i (\tilde{y}^{n\top} a_i - b_i) + \sum_{p \in \Omega} \nu^n_p (\mathbf{1}^\top \tilde{y}^n_p - 1) \right) \\ s.t. \quad & \boldsymbol{\lambda} \succeq 0, \end{aligned} \tag{3.5}$$

where $\boldsymbol{\lambda} \in \mathbb{R}_+^{P \times |\mathcal{D}|}$ and $\boldsymbol{\nu} \in \mathbb{R}^{|\mathcal{D}| \times |\Omega|}$ are the Lagrangian dual variables. Minimizing $\tilde{y}$, for constant $\boldsymbol{\theta}, \boldsymbol{\lambda}, \boldsymbol{\nu}$, can be solved analytically. Updating $\boldsymbol{\lambda}$ and $\boldsymbol{\nu}$ requires to perform a projected gradient ascent[3]. Pathak *et al.* (2015a) concluded that is was best to perform this at each minibatch, for the corresponding samples. While limited to linear functions, it could in theory be extended to any function f_i. However, minimizing $\tilde{y}$ would not be analytically solvable anymore, and would probably requires a dedicated stochastic gradient descent.

3.1.3 Contributions

Interior-point and log-barrier methods can approximate Lagrangian optimization by starting from a feasible solution and solving unconstrained problems, while completely avoiding explicit dual steps and projections. Unfortunately, despite their well-established advantages over penalties, such standard log-barriers were not used before in deep CNNs because finding a feasible set of initial network parameters is not trivial, and is itself a challenging constrained-CNN

[3]We detail the whole algorithm and equations in the Supplemental material 4.

problem. We propose *log-barrier extensions*, which approximate Lagrangian optimization of constrained-CNN problems with a sequence of unconstrained losses, without the need for an initial feasible set of network parameters. Furthermore, we provide a new theoretical result, which shows that the proposed extensions yield a duality-gap bound. This generalizes the standard duality-gap result of log-barriers, yielding sub-optimality certificates for feasible solutions in the case of convex losses. While sub-optimality is not guaranteed for non-convex problems, our result shows that log-barrier extensions are a principled way to approximate Lagrangian optimization for constrained CNNs via *implicit* dual variables. Our approach addresses the well-known limitations of penalty methods and, at the same time, removes the explicit dual updates of Lagrangian optimization. We report comprehensive weakly supervised segmentation experiments, with various constraints, showing that our formulation outperforms substantially the existing constrained-CNN methods, both in terms of accuracy, constraint satisfaction and training stability, more so when dealing with a large number of constraints.

3.2 Background on Duality and the Standard Log-barrier

This section reviews both standard Lagrangian-dual optimization and the log-barrier method for constrained problems Boyd & Vandenberghe (2004). We also present basic concepts of duality theory, namely the *duality gap* and ϵ-suboptimality, which will be needed when introducing our log-barrier extension and the corresponding duality-gap bound. We also discuss the limitations of standard constrained optimization methods in the context of deep CNNs.

Let us consider again the Lagrangian-dual problem in Eq. (3.2). A dual feasible $\boldsymbol{\lambda} \geq 0$ yields a lower bound on the optimal value of constrained problem (3.1), which we denote $\mathcal{E}^*$: $g(\boldsymbol{\lambda}) \leq \mathcal{E}^*$. This important inequality can be easily verified, even when the problem (3.1) is not convex; see Boyd & Vandenberghe (2004), p. 216. It follows that a dual feasible $\boldsymbol{\lambda}$ gives a sub-optimality certificate for a given feasible point $\boldsymbol{\theta}$, without knowing the exact value of $\mathcal{E}^*$: $\mathcal{E}(\boldsymbol{\theta}) - \mathcal{E}^* \leq \mathcal{E}(\boldsymbol{\theta}) - g(\boldsymbol{\lambda})$. Nonnegative quantity $\mathcal{E}(\boldsymbol{\theta}) - g(\boldsymbol{\lambda})$ is the duality gap for primal-dual pair $(\boldsymbol{\theta}, \boldsymbol{\lambda})$. If we manage to find a feasible primal-dual pair $(\boldsymbol{\theta}, \boldsymbol{\lambda})$ such that the duality gap is less or equal than a certain ϵ, then primal feasible $\boldsymbol{\theta}$ is ϵ-suboptimal.

Definition 1. *A primal feasible point* $\boldsymbol{\theta}$ *is* ϵ-suboptimal *when it verifies:* $\mathcal{E}(\boldsymbol{\theta}) - \mathcal{E}^* \leq \epsilon$.

This provides a non-heuristic stopping criterion for Lagrangian optimization, which alternates two iterative steps, one primal and one dual, each decreasing the duality gap until a given accuracy ϵ is attained[4]. In the context of CNNs Pathak *et al.* (2015a), the primal step minimizes the Lagrangian w.r.t. $\boldsymbol{\theta}$, which corresponds to training a deep network with stochastic optimization, e.g., SGD: $\arg\min_{\boldsymbol{\theta}} \mathcal{L}(S_{\boldsymbol{\theta}}, \boldsymbol{\lambda})$. The dual step is a constrained maximization of the dual function[5] via projected gradient ascent: $\max_{\boldsymbol{\lambda}} g(\boldsymbol{\lambda})$ s.t $\boldsymbol{\lambda} \geq 0$. As mentioned before, direct use of Lagrangian optimization for deep CNNs increases computational complexity and can lead to instability or poor convergence due to the interplay between stochastic optimization for the primal and the iterates/projections for the dual. Our work approximates Lagrangian-dual optimization with a sequence of unconstrained log-barrier-extension losses, in which the dual variables are *implicit*, avoiding explicit dual iterates/projections. Let us first review the standard log-barrier method.

3.2.1 The standard log-barrier

The log-barrier method is widely used for inequality-constrained optimization, and belongs to the family of *interior-point* techniques Boyd & Vandenberghe (2004). To solve our constrained CNN problem (3.1) with this method, we need to find a strictly feasible set of network parameters $\boldsymbol{\theta}$ as a starting point, which can then be used in an unconstrained problem via the standard log-barrier function. In the general context of optimization, log-barrier methods proceed in two steps. The first, often called *phase I* Boyd & Vandenberghe (2004), computes a feasible point by Lagrangian minimization of a constrained problem, which in the case of (3.1) is:

$$\begin{aligned} \min_{x,\boldsymbol{\theta}} \quad & x \\ \text{s.t.} \quad & f_i(s^n_{\boldsymbol{\theta}}) \leq x \qquad\qquad \forall i \in \{1, \ldots, P\}, \forall n \in \mathcal{D} \end{aligned} \tag{3.6}$$

[4]Strong duality should hold if we want to achieve arbitrarily small tolerance ϵ. Of course, strong duality does not hold in the case of CNNs as the primal problem is not convex.

[5]Notice that the dual function is always concave as it is the minimum of a family of affine functions, even when the original (or primal) problem is not convex, as is the case for CNNs.

For deep CNNs with millions of parameters, Lagrangian optimization of problem (3.6) has the same difficulties as with the initial constrained problem in (3.1). To find a feasible set of network parameters, one needs to alternate CNN training and projected gradient ascent for the dual variables. This might explain why such interior-point methods, despite their substantial impact in optimization Boyd & Vandenberghe (2004), are mostly overlooked in modern deep networks[6], as is generally the case for other Lagrangian-dual optimization methods.

The second step, often referred to as *phase II*, approximates (3.1) as an unconstrained problem:

$$\min_{\boldsymbol{\theta}} \quad \mathcal{E}(\boldsymbol{\theta}) + \sum_{i=1}^{P} \sum_{n=1}^{N} \psi_t \left(f_i(s_{\boldsymbol{\theta}}^n) \right) \tag{3.7}$$

where ψ_t is the log-barrier function: $\psi_t(z) = -\frac{1}{t}\log(-z)$. When $t \to +\infty$, this convex, continuous and twice-differentiable function approaches a hard indicator for the constraints: $H(z) = 0$ if $z \leq 0$ and $+\infty$ otherwise. The domain of the function is the set of feasible points. The higher t, the better the quality of the approximation. This suggest that large t yields a good approximation of the initial constrained problem in (3.1). This is, indeed, confirmed with the following standard duality-gap result for the log-barrier method Boyd & Vandenberghe (2004), which shows that optimizing (3.7) yields a solution that is PN/t-suboptimal.

Proposition 1. *Let $\boldsymbol{\theta}^*$ be the feasible solution of unconstrained problem* (3.7) *and $\boldsymbol{\lambda}^* \in \mathbb{R}^{P \times N}$, with $\lambda_{i,n}^* = -1/(t\, f_i(s_{\boldsymbol{\theta}}^n))$. Then, the duality gap associated with primal feasible $\boldsymbol{\theta}^*$ and dual feasible $\boldsymbol{\lambda}^*$ for the initial constrained problem in* (3.1) *is:*

$$\mathcal{E}(\boldsymbol{\theta}^*) - g(\boldsymbol{\lambda}^*) = PN/t$$

Proof: The proof can be found in Boyd & Vandenberghe (2004), p. 566. □

An important implication that follows immediately from proposition (1) is that a feasible solution of approximation (3.7) is PN/t-suboptimal: $\mathcal{E}(\boldsymbol{\theta}^*) - \mathcal{E}^* \leq PN/t$. This suggests a

[6]Interior-point methods were investigated for artificial neural networks before the deep learning era Trafalis, Tutunji & Couellan (1997).

simple way for solving the initial constrained problem with a guaranteed ϵ-suboptimality: We simply choose large $t = PN/\epsilon$ and solve unconstrained problem (3.7). However, for large t, the log-barrier function is difficult to minimize because its gradient varies rapidly near the boundary of the feasible set. In practice, log-barrier methods solve a sequence of problems of the form (3.7) with an increasing value t. The solution of a problem is used as a starting point for the next, until a specified ϵ-suboptimality is reached.

3.3 Log-barrier Extensions

We propose the following unconstrained loss for approximating Lagrangian optimization of constrained problem (3.1):

$$\min_{\boldsymbol{\theta}} \quad \mathcal{E}(\boldsymbol{\theta}) + \sum_{i=1}^{P} \sum_{n=1}^{N} \tilde{\psi}_t \left(f_i(s^n_{\boldsymbol{\theta}}) \right) \tag{3.8}$$

where $\tilde{\psi}_t$ is our *log-barrier extension*, which is convex, continuous and twice-differentiable:

$$\tilde{\psi}_t(z) = \begin{cases} -\frac{1}{t} \log(-z) & \text{if } z \leq -\frac{1}{t^2} \\ tz - \frac{1}{t} \log(\frac{1}{t^2}) + \frac{1}{t} & \text{otherwise} \end{cases} \tag{3.9}$$

Similarly to the standard log-barrier, when $t \to +\infty$, our extension (3.9) can be viewed a smooth approximation of hard indicator function H. However, a very important difference is that the domain of our extension $\tilde{\psi}_t$ is not restricted to feasible points $\boldsymbol{\theta}$. Therefore, our approximation (3.8) removes completely the requirement for explicit Lagrangian-dual optimization for finding a feasible set of network parameters. In our case, the inequality constraints are fully handled within stochastic optimization, as in standard unconstrained losses, avoiding completely gradient ascent iterates and projections over *explicit* dual variables. As we will see in the experiments, our formulation yields better results in terms of accuracy and stability than the recent penalty constrained CNN method in Kervadec *et al.* (2019b).

In our approximation in (3.8), the Lagrangian dual variables for the initial inequality-constrained problem of (3.1) are *implicit*. We prove the following duality-gap bound, which yields sub-optimality certificates for feasible solutions of our approximation in (3.8). Our result[7] can be viewed as an extension of the standard result in proposition 1, which expresses the duality-gap as a function of t for the log-barrier function.

Proposition 2. *Let $\boldsymbol{\theta}^*$ be the solution of problem* (3.8) *and $\boldsymbol{\lambda}^* \in \mathbb{R}^{P\times N}$ the corresponding vector of implicit Lagrangian dual variables given by:*

$$\lambda^*_{i,n} = \begin{cases} -\frac{1}{t f_i(s^n_{\boldsymbol{\theta}^*})} & \text{if } f_i(s^n_{\boldsymbol{\theta}^*}) \leq -\frac{1}{t^2} \\ t & \text{otherwise} \end{cases} . \tag{3.10}$$

Then, we have the following upper bound on the duality gap associated with primal $\boldsymbol{\theta}^*$ and implicit dual feasible $\boldsymbol{\lambda}^*$ for the initial inequality-constrained problem (3.1)*:*

$$\mathcal{E}(\boldsymbol{\theta}^*) - g(\boldsymbol{\lambda}^*) \leq PN/t$$

Proof: We give a detailed proof of Prop. 2 in the Supplemental material. □

From proposition 2, the following important fact follows immediately: If the solution $\boldsymbol{\theta}^*$ that we obtain from unconstrained problem (3.8) is feasible and global, then it is PN/t-suboptimal for constrained problem (3.1): $\mathcal{E}(\boldsymbol{\theta}^*) - \mathcal{E}^* \leq PN/t$.

Finally, we arrive to our constrained CNN learning algorithm, which is fully based on SGD. Similarly to the standard log-barrier algorithm, we use a varying parameter t. We optimize a sequence of losses of the form (3.8) and increase gradually the value t by a factor μ. The network parameters obtained for the current t and epoch are used as a starting point for the next t and epoch. We can summarize the fundamental differences between our log-barrier extension and a standard penalty function as follows:

[7]Our result applies to the general context of convex optimization. In deep CNNs, of course, a feasible solution of our approximation may not be unique and is not guaranteed to be a global optimum as $\mathcal{E}$ and the constraints are not convex.

a) A penalty does not act as a barrier near the boundary of the feasible set, i.e., a satisfied constraint yields null penalty and gradient. Therefore, at a given gradient update, there is nothing that prevents a satisfied constraint from being violated, causing oscillations between competing constraints and making the training unstable. On the contrary, the strictly positive gradient of our log-barrier extension gets higher when a satisfied constraint approaches violation during optimization, pushing it back towards the feasible set.

b) Another fundamental difference is that the derivatives of our log-barrier extensions yield the implicit dual variables in Eq. (3.10), with sub-optimality and duality-gap guarantees, which is not the case for penalties. Therefore, our log-barrier extension mimics Lagrangian optimization, but with implicit rather than explicit dual variables. The detailed proof of Prop. 2 in the Supplemental material clarifies how the $\lambda^*_{i,n}$'s in Eq. (3.10) can be viewed as implicit dual variables.

3.4 Experiments

Most of the existing methods[8]—and the proposed log-barrier – are compatible with any differentiable function f_i, including non-linear and fractional terms, as in Eqs. (3.11) and (3.12) introduced further in the paper. However, we hypothesize that our log-barrier extension is better for handling the interplay between multiple competing constraints. To validate this hypothesis, we compare all strategies on the joint optimization of joint segmentation constraints related to region size and centroid. We will test the Lagrangian with proposals from Pathak *et al.* (2015a) when the experiments allows it, i.e., when the functions constrained are linear and the number of total constraints per image is not too high[9].

[8]The only and notable exception being Pathak *et al.* (2015a).

[9]As their complexity is linear to the number of constraints, their method quickly becomes intolerably slow with high number of constraints, making it not feasible to train a neural network in a timely fashion.

Region-size constraint

We define the size (or volume) of a segmentation for class k as the sum of its softmax predictions over the image domain:

$$\mathcal{V}^n_{k,\boldsymbol{\theta}} = \sum_{p \in \Omega} s^n_{k,p,\boldsymbol{\theta}} \tag{3.11}$$

We use the following inequality constraints on region size: $0.9\tau_{\mathcal{V}^n_k} \leq \mathcal{V}^n_{k,\boldsymbol{\theta}} \leq 1.1\tau_{\mathcal{V}^n_k}$, where, similarly to the experiments in Kervadec *et al.* (2019b), $\tau_{\mathcal{V}^n_k} = \sum_{p \in \Omega} y^n_{k,p}$ is determined from the ground truth y^n of each image.

Region-centroid constraints

The centroid of the predicted region can be computed as a weighted average of the pixel coordinates:

$$C^n_{k,\boldsymbol{\theta}} = \frac{\sum_{p \in \Omega} s^n_{k,p,\boldsymbol{\theta}} c_p}{\sum_{p \in \Omega} s^n_{k,p,\boldsymbol{\theta}}}, \tag{3.12}$$

where $c_p \in \mathbb{N}^2$ are the pixel coordinates on a 2D grid. We constrain the position of the centroid in a box around the ground-truth centroid: $\tau_{C^n_k} - 20 \leq C^n_{k,\boldsymbol{\theta}} \leq \tau_{C^n_k} + 20$, with $\tau_{C^n_k} = \frac{\sum_{p \in \Omega} y^n_{k,p} c_p}{\sum_{p \in \Omega} y^n_{k,p}}$ corresponding to the bound values associated with each image.

Bounding box tightness prior

This prior Hsu *et al.* (2019); Lempitsky, Kohli, Rother & Sharp (2009) assumes that any horizontal or vertical line inside the bounding box of an object of class k will eventually cross the object. This can be generalized with segments of width w inside the box, that will cross at least w times the object. This prior can be easily reformulated as constraints. If $\mathcal{S}^n_L := \{s^n_l\}$ denotes the set of parallel segments to the sides of the bounding box for sample n, the following set of inequality constraints is trivial to define:

$$\sum_{p \in s^n_l} y^n_{k,p} \geq w \qquad \forall s^n_l \in \mathcal{S}^n_L, \forall n \in \mathcal{D}. \tag{3.13}$$

If we define the inside of the bounding box as Ω_F, and the outside as Ω_B (such as $\Omega = \Omega_F \cup \Omega_B$ and $\Omega_F \cap \Omega_B = \{\emptyset\}$), we can define two other useful constraints for each image:

$$\sum_{p \in \Omega_B} s^n_{k,p,\boldsymbol{\theta}} \leq 0 \qquad \forall n \in \mathcal{D}, \tag{3.14}$$

$$\sum_{p \in \Omega} s^n_{k,p,\boldsymbol{\theta}} \leq |\Omega_F| \qquad \forall n \in \mathcal{D}. \tag{3.15}$$

There is some interplay between constraint (3.13) and constraint (3.14), as they have competing trivial solutions: $s^n_{k,p,\boldsymbol{\theta}} = 1 \; \forall p$ would satisfy constraint (3.13) perfectly, whereas $s^n_{k,p,\boldsymbol{\theta}} = 0 \; \forall p$ would satisfy (3.14). While constraint (3.15) is there to balance the two and limit the shift to extremes, this setting remain a good benchmark to evaluate the interplay of multiple, competing constraints simultaneously.

3.4.1 Datasets and Evaluation Metrics

Our evaluations and comparisons were performed on three different segmentation scenarios using synthetic and medical images. The data sets used in each of these problems are detailed below.

Synthetic images

We generated a synthetic dataset composed of 1100 images with two different circles of the same size but different intensity values, where the darker circle is the target region (Fig. 3.1, first column). Furthermore, different levels of Gaussian noise were added to the images. We employed 1000 images for training and 100 for validation. The objective of this simple dataset is to compare our log-barrier extension to other methods when several functions are constrained, e.g., size and centroid. Imposing these constraints individually is not sufficient to learn which circle is the target, since no pixel annotation is used during training. However, if the two constraints are combined, it *should* be enough to identify the correct circle.

This setting will evaluate how different methods behave when there exist interplay between two different constraints.

Medical images

We use the dataset from the MICCAI 2012 prostate segmentation challenge Litjens *et al.* (2014). This dataset contains Magnetic Resonance (MR) images from 50 patients, from which we employ 10 patients for validation and use the rest for training. We investigate two different settings on this dataset. Setting I) we test the combinations of constraints (3.13), (3.14) and (3.15), with bounding boxes derived from the ground truth. Setting II) we test the setting of Kervadec *et al.* (2019b), where weak labels derived from the ground truth by placing random dots inside the object of interest (see Figure in appendix 2) and a region-size constraints in the form of (3.11) is imposed.

Evaluation

We resort to the common Dice index (DSC) = $\frac{2|S \bigcap Y|}{|S|+|Y|}$ to evaluate the performance of tested methods. Furthermore, we evaluate the effectiveness and stability of the constrained optimization methods. To this end, we first compute at each epoch the percentage of constraints that are satisfied. Second, we measure the stability of the constraints, i.e., the percentage of constraints satisfied at epoch t that are still satisfied at epoch $t + 1$. And last, we simply measure the time taken to run a single epoch for each method, including the proposal generation of Pathak *et al.* (2015a) and the $\boldsymbol{\lambda}$ update for the Standard Lagrangian and the ReLU Lagrangian Nandwani *et al.* (2019).

3.4.2 Training and implementation details

Since the two datasets have very different characteristics, we considered a specific network architecture and training strategy for each of them.

For the dataset of synthetic images, we used the ENet network Paszke *et al.* (2016), as it has shown a good trade-off between accuracy and inference time. The network was trained from scratch using the Adam optimizer and a batch size of 1. The initial learning rate was set to 5×10^{-4} and decreased by half if validation performance did not improve for 20 epochs. Softmax temperature value was set to 5. To segment the prostate, we used the same settings as in Kervadec *et al.* (2019b), reporting their results for the penalty-based baselines.

For the standard and ReLU Lagrangian, we alternate between one epoch (one update for each sample of the dataset) of SGD to optimize $\boldsymbol{\theta}$, then one epoch to update $\boldsymbol{\lambda}$. We set to 5 the initial t value of our extended log-barrier. We increased it by a factor of $\mu = 1.1$ after each epoch. This strategy relaxes constraints in the first epochs so that the network can focus on learning from images, and then gradually makes these constraints harder as optimization progresses. All experiments were implemented in Python 3.8 with PyTorch 1.4 Paszke *et al.* (2017). All the experiments were carried out on a server equipped with a NVIDIA Titan RTX. The code is publicly available[10].

3.4.3 Results

Quantitative results

Results in terms of DSC are reported in Table 3.1. The first thing we can observe is that the standard Lagrangian, despite the introduction of a dedicated learning rate for its $\boldsymbol{\lambda}$ update, is not able to learn when multiple constraints enter in competition, i.e, DSC of 0.005 in the synthetic example. In addition, the ReLU Lagrangian approach proposed by Nandwani *et al.* (2019) can better handle multiple constraints than a simple penalty He *et al.* (2017); Kervadec *et al.* (2019b), but it performs similarly if only one constraint is enforced, such as in the case of the size constraint on the PROMISE12 dataset. On the other hand, with the high number of constraints and trivial solutions to balance, the proposed log-barrier extension learns successfully based on the information given by the constraints, compared to the other methods,

[10]https://github.com/LIVIAETS/extended_logbarrier

achieving the best DSC across the three settings. The behavior of the ReLU Lagrangian is very interesting, which highlights one of the drawbacks of the ReLU introduced in their Lagrangian formulation. As $\boldsymbol{\lambda}$ can only increase—which happens when a constraint is not satisfied—when trying to balance each constraint, all $\boldsymbol{\lambda}$ keep increasing, making the subtle balance more and more difficult to achieve. For instance, the constraint (3.14) started with $\lambda = 0$ and after 200 epochs it had reached an average value of 350 million across the whole dataset (PROMISE12), despite the introduction of a learning rate to slow down its increase. The lower performance of penalty-based methods can be explained by the high-gradients generated when constraints are not satisfied, which leads to big and simplistic updates.

Table 3.1 Mean DSC and standard deviation of the last 10 epochs on the validation on the toy example and PROMISE12 datasets.

Method	Synthetic dataset	PROMISE12	
		Setting I	Setting II
Lagrangian proposal Pathak *et al.* (2015a)	NA	NA	0.740 (0.018)
Standard Lagrangian	0.005 (0.014)	0.000 (0.000)	0.752 (0.007)
ReLU Lagrangian Nandwani *et al.* (2019)	0.798 (0.006)	0.000 (0.000)	0.790 (0.007)
Penalty He *et al.* (2017); Kervadec *et al.* (2019b)	0.712 (0.022)	0.000 (0.000)	0.817 (0.006)
Log-barrier extensions (ours)	**0.945** (0.001)	**0.813** (0.024)	**0.823** (0.003)
Full supervision	0.998 (0.000)	0.880 (0.001)	

Qualitative results

A visual comparison of the predicted results on the toy example is depicted in Figure 3.1. In this figure we can first observe that standard Lagrangian generates noisy segmentations, which is in line with the quantitative results reported in Table 3.1. Both ReLU Lagrangian Nandwani *et al.* (2019) and penalty-based methods obtain better target segmentations. Nevertheless, as observed in the case of penalties, they cannot handle efficiently the interplay between multiple constraints. While the size constraint is apparently satisfied, the centroid constraint is not properly enforced (e.g., the non-target circle contains segmented regions). Last, the proposed extended log-barrier demonstrates a strong ability to handle several constraints simultaneously, which is reflected in the circle segmentation close to the ground truth.

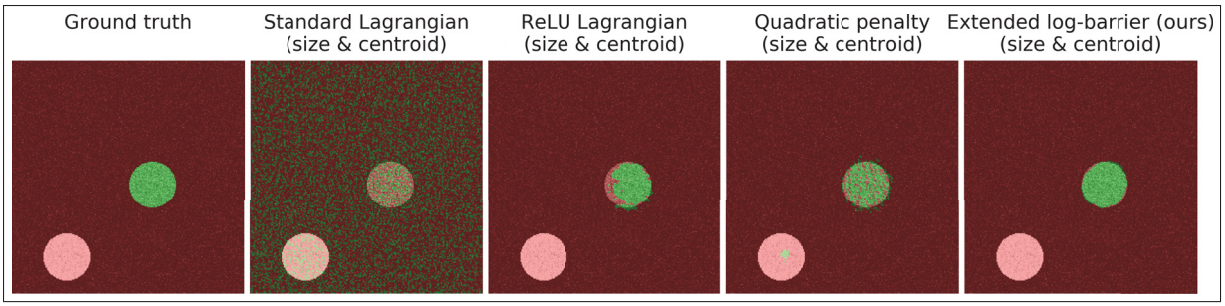


Figure 3.1 Results on the synthetic dataset (background in red and foreground in green).

Constraints satisfaction and stability

We further evaluated our method in terms of how the constraints are satisfied across epochs and the stability during training, whose results are shown in Fig. 3.2. We can first notice that on top of the better absolute performances, the proposed log-barrier extension is also more stable during training, both in performance and constraints satisfaction. The gap between the proposed approach and prior work is more significant on the synthetic dataset, where multiple constraints are enforced simultaneously. Other methods that perform satisfactorily in terms of DSC metric, i.e., quadratic penalty or ReLU Lagrangian, tend to present a higher variance across epochs when the constraints satisfaction and stability is evaluated. This indicates that our method not only achieves the best segmentation performance, but also satisfies the constraints better than known approaches.

Computational cost and efficiency

Penalties and the proposed log-barrier extension have negligible cost compared to optimizing the base-loss $\mathcal{E}(\boldsymbol{\theta})$ alone (up to 5% slowdown when the number of constraints becomes very high). In contrast, Lagrangian methods incur in higher computational cost. For example, in the standard and ReLU Lagrangian, it amounts to nearly a 25% slowdown (due to the extra loop over the training set to perform the $\boldsymbol{\lambda}$ update). The Lagrangian with proposals in Pathak *et al.* (2015a) is much slower (about three times slower in the studied setting).

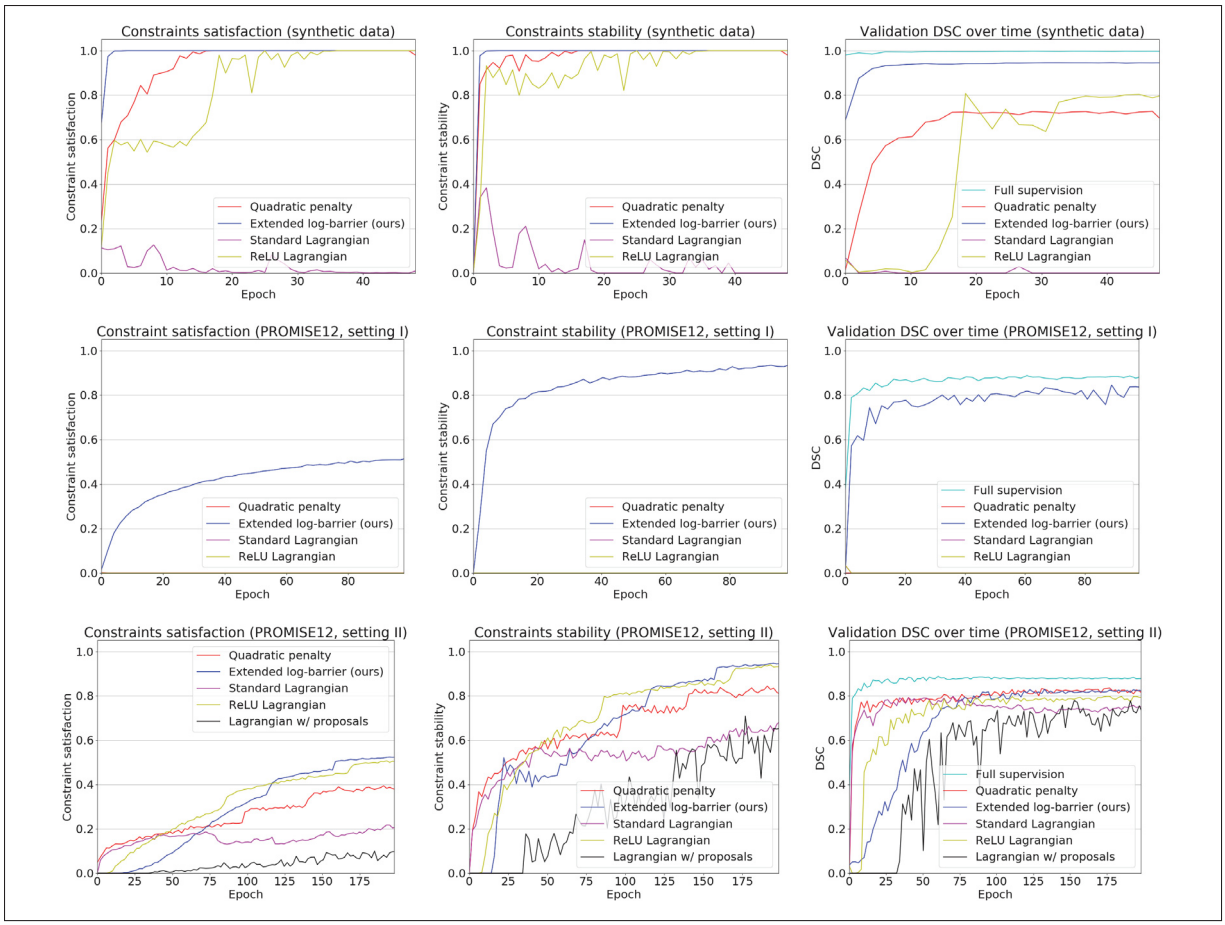


Figure 3.2 Constraints satisfaction, stability and DSC evolution on different settings.

3.5 Conclusion

We proposed log-barrier extensions, which approximate Lagrangian optimization of constrained-CNN problems with a sequence of unconstrained losses. Our formulation relaxes the need for an initial feasible solution, unlike standard interior-point and log-barrier methods. This makes it convenient for deep networks. We also provided an upper bound on the duality gap for our proposed extensions, thereby generalizing the duality-gap result of standard log-barriers and showing that our formulation has dual variables that mimic implicitly (without dual projections/steps) Lagrangian optimization. Therefore, our implicit Lagrangian formulation can be fully handled with SGD, the workhorse of deep networks. We reported comprehensive

constrained-CNN experiments, showing that log-barrier extensions outperform several other types of Lagrangian methods and penalties, in terms of accuracy and training stability.

While we evaluated our approach in the context of weakly supervised segmentation, log-barrier extensions can be useful in breadth of problems in vision and learning, where constraints occur naturally. This include, for instance, adversarial robustness Rony, Hafemann, Oliveira, Ben Ayed, Sabourin & Granger (2019), stabilizing the training of GANs Gulrajani, Ahmed, Arjovsky, Dumoulin & Courville (2017), domain adaptation for segmentation Zhang, David, Foroosh & Gong (2019), pose-constrained image generation Hu, Yang, Salakhutdinov, Qin, Liang, Dong & Xing (2018), 3D human pose estimation Márquez-Neila *et al.* (2017), deep reinforcement learning He *et al.* (2017) and natural language processing Nandwani *et al.* (2019). To our knowledge, constraints (either equality[11] or inequality) in these problems, among others in the context of deep networks, are typically handled with basic penalties. Therefore, it will be interesting to investigate log-barrier extensions in these diverse contexts.

[11]Note that our framework can also be used for equality constraints as one can transform an equality constraint into two inequality constraints.

CHAPTER 4

BOUNDARY LOSS FOR HIGHLY UNBALANCED SEGMENTATION

Hoel Kervadec[a], Jihene Bouchtiba[a], Christian Desrosiers[b], Eric Granger[a], Jose Dolz[b], Ismail Ben Ayed[a]

[a]Département de génie des systèmes, ÉTS Montréal, QC, Canada,
[b]Département de génie logiciel et des TI, ÉTS Montréal, QC, Canada,

Oral presentation at Medical Imaging with Deep Learning (MIDL) 2019, London.
Runner-up for best paper award.
Invited for a deep learning special issue in Medical Image Analysis (MEDIA), volume 67.

Abstract

Widely used loss functions for CNN segmentation, e.g., Dice or cross-entropy, are based on integrals over the segmentation regions. Unfortunately, for highly unbalanced segmentations, such regional summations have values that differ by several orders of magnitude across classes, which affects training performance and stability. We propose a *boundary* loss, which takes the form of a distance metric on the space of contours, not regions. This can mitigate the difficulties of highly unbalanced problems because it uses integrals over the interface between regions instead of unbalanced integrals over the regions. Furthermore, a boundary loss complements regional information. Inspired by graph-based optimization techniques for computing active-contour flows, we express a non-symmetric L_2 distance on the space of contours as a regional integral, which avoids completely local differential computations involving contour points. This yields a boundary loss expressed with the regional softmax probability outputs of the network, which can be easily combined with standard regional losses and implemented with any existing deep network architecture for N-D segmentation. We report comprehensive evaluations and comparisons on different unbalanced problems, showing that our boundary loss can yield significant increases in performances while improving training stability. Our code is publicly available.

4.1 Introduction

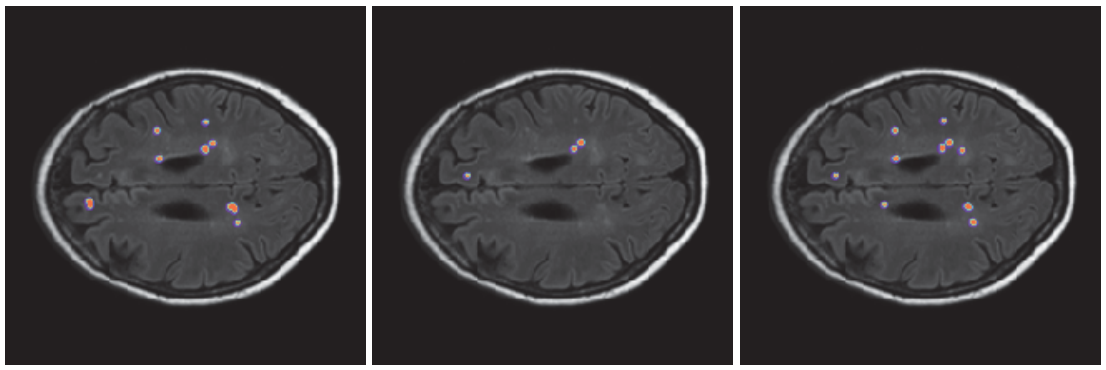
Recent years have witnessed a substantial growth in the number of deep learning methods for medical image segmentation Dolz *et al.* (2018); Ker, Wang, Rao & Lim (2018); Litjens *et al.* (2017); Shen, Wu & Suk (2017). Widely used loss functions for segmentation, e.g., Dice or cross-entropy, are based on *regional* integrals, which are convenient for training deep neural networks. In practice, these regional integrals are summations over the segmentation regions of differentiable functions, each directly invoking the softmax probability outputs of the network. Therefore, standard stochastic optimizers such as SGD are directly applicable. Unfortunately, difficulties occur for highly unbalanced segmentations, for instance, when the size of target foreground region is several orders of magnitude less than the background size. For example, in the characterization of white matter hyperintensities (WMH) of presumed vascular origin, the foreground composed of WMH regions may be 500 times smaller than the background (see the typical example in Fig. 4.1). In such cases, quite common in medical image analysis, standard regional losses contain foreground and background terms with values that differ considerably, typically by several orders of magnitude, potentially affecting performance and training stability Milletari *et al.* (2016); Sudre *et al.* (2017).

Segmentation approaches based on convolutional neural networks (CNN) are typically trained by minimizing the cross-entropy (CE), which measures an affinity between the regions defined by probability softmax outputs of the network and the corresponding ground-truth regions. The standard regional CE has well-known drawbacks in the context of highly unbalanced problems. It assumes identical importance distribution of all the samples and classes. To achieve good generalization, it requires a large training set with balanced classes. For unbalanced data, CE typically results in unstable training and leads to decision boundaries biased towards the majority classes. Class-imbalanced learning aims to mitigate learning bias by promoting the importance of infrequent labels. In medical image segmentation, a common strategy is to re-balance class prior distributions by down-sampling frequent labels Havaei, Davy, Warde-Farley, Biard, Courville, Bengio, Pal, Jodoin & Larochelle (2017); Valverde, Cabezas, Roura, González-Villà, Pareto, Vilanova, Ramio-Torrenta, Rovira, Oliver & Lladó (2017). Nevertheless, this strategy

limits the information of the images used for training. Another common practice is to assign weights to the different classes that are inversely proportional to the frequency of the corresponding labels Brosch, Yoo, Tang, Li, Traboulsee & Tam (2015); Kamnitsas, Ledig, Newcombe, Simpson, Kane, Menon, Rueckert & Glocker (2017); Long *et al.* (2015); Ronneberger *et al.* (2015); Yu, Yang, Chen, Qin & Heng (2017). In this scenario, the standard cross-entropy (CE) loss is modified so as to assign more importance to the rare labels. Although effective for some unbalanced problems, such weighting methods may undergo serious difficulties when dealing with highly unbalanced datasets, as seen with WMH segmentation. The CE gradient computed over the few pixels of infrequent labels is typically noisy, and amplifying this noise with a high class weight may lead to instability.

The well-known Dice overlap coefficient was also adopted as a regional loss function, typically outperforming CE in unbalanced medical image segmentation problems Milletari *et al.* (2016); Milletari, Ahmadi, Kroll, Plate, Rozanski, Maiostre, Levin, Dietrich, Ertl-Wagner, Bötzel et al. (2017); Wong, Moradi, Tang & Syeda-Mahmood (2018). Sudre et al. Sudre *et al.* (2017) generalized the Dice loss Milletari *et al.* (2016) by weighting according to the squared inverse of class-label frequency. Despite these improvements over CE Milletari *et al.* (2016); Sudre *et al.* (2017), regional Dice losses may encounter difficulties when dealing with very small structures. In such highly unbalanced scenarios, mis-classified pixels may lead to large decreases of the loss, resulting in unstable optimization. Furthermore, Dice corresponds to the harmonic mean between precision and recall, implicitly using the arithmetic mean of false positives and false negatives. False positives and false negatives are, therefore, equally important when the true positives remain the same, making this loss mainly appropriate when both types of errors are equally high. The recent research in Abraham & Khan (2019); Salehi, Erdogmus & Gholipour (2017) investigated losses based on the Tversky similarity index in order to provide a better trade-off between precision and recall. It introduced two parameters that control the importance of false positives and false negatives. Other recent advances in class-imbalanced learning for computer vision problems have been adopted in medical image segmentation. For example, inspired by the concept of focal loss Lin, Goyal, Girshick, He & Dollár (2018), Dice and

Tvserky losses have been extended to integrate a focal term, which is parameterized by a value that controls the importance of easy and hard training samples Abraham & Khan (2019); Wong *et al.* (2018). Furthermore, the combination of several of these regional losses has been further investigated Zhu, Huang, Zeng, Chen, Liu, Qian, Du, Fan & Xie (2019). The main objective of these losses is to balance the classes not only in terms of their relative class sizes, but also by the level of segmentation difficulty.



a) Ground truth b) GDL c) GDL + boundary loss

Figure 4.1 A visual comparison that shows the positive effect of our boundary loss on a validation data from the WMH dataset. Our boundary loss helped to recover small regions that were otherwise missed by the model trained with the generalized Dice loss (GDL). Best viewed in colors.

More recently, Karimi et al. Karimi & Salcudean (2019) proposed a novel loss function that attempts to directly reduce the Hausdorff distance (HD). This relaxed loss based on the HD is shown to bring improvements when combined with the DSC loss. Nevertheless, its main drawback is the high computational cost of computing the distance transforms. Particularly, at each training epoch, the new distance maps have to be recomputed for all the images, which incurs in a computationally costly process. This issue is further magnified in the case of 3D volumes, which heavily increases the computational burden.

4.1.1 Contributions

In this paper, we propose a *boundary* loss that takes the form of a distance metric on the space of contours (or shapes), not regions. We argue that a boundary loss can mitigate the issues related to regional losses in highly unbalanced segmentation problems. Rather than using unbalanced integrals over the regions, a boundary loss uses integrals over the boundary (interface) between the regions. Furthermore, it provides information that is complementary to regional losses. It is, however, challenging to represent the boundary points corresponding to the regional softmax outputs of a CNN. This difficulty may explain why boundary losses have been avoided in the context of deep segmentation networks. Our boundary loss is inspired by techniques in discrete graph-based optimization for computing gradient flows of curve evolution Boykov, Kolmogorov, Cremers & Delong (2006). Following an integral approach for computing boundary variations, we express a non-symmetric L_2 distance on the space of shapes (or contours) as a regional integral, which avoids completely local differential computations involving contour points. This yields a boundary loss expressed as the sum of linear functions of the regional softmax probability outputs of the network. Therefore, it can be easily combined with standard regional losses and implemented with any existing deep network architecture for N-D segmentation.

We evaluated our boundary loss in conjunction with various region-based losses on two challenging and highly unbalanced segmentation problems—the Ischemic Stroke Lesion (ISLES) and the White Matter Hyperintensities (WMH) benchmark datasets. The results indicate that the proposed boundary loss yields a more stable learning process, and can bring significant gains in performances, in terms of Dice and Hausdorff scores.

4.2 Formulation

Let $I : \Omega \subset \mathbb{R}^{2,3} \rightarrow \mathbb{R}$ denotes a training image with spatial domain Ω, and $g : \Omega \rightarrow \{0, 1\}$ a binary ground-truth segmentation of the image: $g(p) = 1$ if pixel/voxel p belongs to the target region $G \subset \Omega$ (foreground region) and 0 otherwise, i.e., $p \in \Omega \setminus G$ (background region)[1]. Let

[1]We focus on two-region segmentation to simplify the presentation. However, our formulation extends to the multi-region case in a straightforward manner.

$s_\theta : \Omega \rightarrow [0, 1]$ denotes the softmax probability output of a deep segmentation network, and $S_\theta \subset \Omega$ the corresponding segmentation region: $S_\theta = \{p \in \Omega \,|\, s_\theta(p) \geq \delta\}$ for some threshold δ. Widely used segmentation loss functions involve a *regional integral* for each segmentation region in Ω, which measures some similarity (or overlap) between the region defined by the probability outputs of the network and the corresponding ground-truth. In the two-region case, we have an integral of the general form $\int_\Omega g(p) f(s_\theta(p)) dp$ for the foreground, and of the form $\int_\Omega (1 - g(p)) f(1 - s_\theta(p)) dp$ for the background. For instance, the standard two-region cross-entropy loss corresponds to a summation of these two terms for $f = -\log(\cdot)$. Similarly, the generalized Dice loss (GDL) Sudre *et al.* (2017) involves regional integrals with $f = 1$, subject to some normalization, and is given as follows for the two-region case:

$$\mathcal{L}_{GDL}(\theta) = 1 - 2\frac{w_G \int_{p\in\Omega} g(p) s_\theta(p) dp + w_B \int_{p\in\Omega} (1 - g(p))(1 - s_\theta(p)) dp}{w_G \int_\Omega [s_\theta(p) + g(p)] dp + w_B \int_\Omega [2 - s_\theta(p) - g(p)] dp} \tag{4.1}$$

where coefficients $w_G = 1/\left(\int_{p\in\Omega} g(p) dp\right)^2$ and $w_B = 1/\left(\int_\Omega (1 - g(p)) dp\right)^2$ are introduced to reduce the well-known correlation between the Dice overlap and region size.

Regional integrals are widely used because they are convenient for training deep segmentation networks. In practice, these regional integrals are summations of differentiable functions, each invoking directly the softmax probability outputs of the network, $s_\theta(p)$. Therefore, standard stochastic optimizers such SGD are directly applicable. Unfortunately, extremely unbalanced segmentations are quite common in medical image analysis, where, e.g., the size of the target foreground region is several orders of magnitude smaller than the background size. This represents challenging cases because the foreground and background terms have substantial differences in their values, which affects segmentation performance and training stability Milletari *et al.* (2016); Sudre *et al.* (2017).

Our purpose is to build a boundary loss $\text{Dist}(\partial G, \partial S_\theta)$, which takes the form of a distance metric on the space of contours (or region boundaries) in Ω, with ∂G denoting a representation of the boundary of ground-truth region G (e.g., the set of points of G, which have a spatial neighbor in background $\Omega \setminus G$) and ∂S_θ denoting the boundary of the segmentation region

defined by the network output. On the one hand, a boundary loss should be able to mitigate the above-mentioned difficulties for unbalanced segmentations: rather than using unbalanced integrals within the regions, it uses integrals over the boundary (interface) between the regions. Furthermore, a boundary loss provides information that is different from and, therefore, complimentary to regional losses. On the other hand, it is not clear how to represent boundary points on ∂S_θ as a differentiable function of regional network outputs s_θ. This difficulty might explain why boundary losses have been mostly avoided in the context of deep segmentation networks.

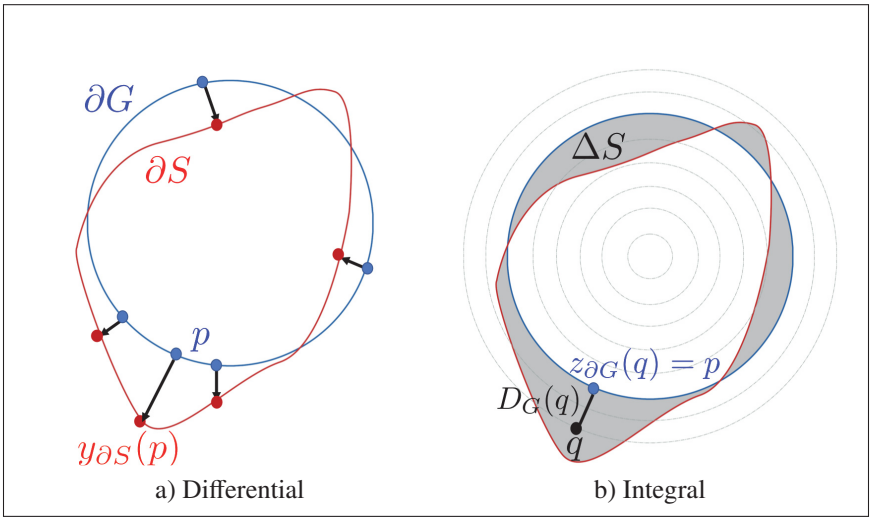


Figure 4.2 The relationship between *differential* and *integral* approaches for evaluating boundary change (variation).

Our boundary loss is inspired from discrete (graph-based) optimization techniques for computing gradient flows of curve evolution Boykov *et al.* (2006). Similarly to our problem, curve evolution methods require a measure for evaluating boundary changes (or variations). Consider the following non-symmetric L_2 distance on the space of shapes, which evaluates the change between two nearby boundaries ∂S and ∂G Boykov *et al.* (2006):

$$\text{Dist}(\partial G, \partial S) = \int_{\partial G} \|y_{\partial S}(p) - p\|^2 dp \tag{4.2}$$

where $p \in \Omega$ is a point on boundary ∂G and $y_{\partial S}(p)$ denotes the corresponding point on boundary ∂S, along the direction normal to ∂G, i.e., $y_{\partial S}(p)$ is the intersection of ∂S and the line that is normal to ∂G at p (See Fig. 4.2.a for an illustration) $\|.\|$ denotes the L_2 norm. In fact, this *differential* framework for evaluating boundary change is in line with standard variational curve evolution methods Mitiche & Ben Ayed (2011), which compute the motion of each point p on the evolving curve as a velocity along the normal to the curve at point p. Similarly to any contour distance invoking directly points on contour ∂S, expression (4.2) cannot be used directly as a loss for $\partial S = \partial S_\theta$. However, it is easy to show that the differential boundary variation in (4.2) can be approximated using an *integral* approach Boykov *et al.* (2006), which avoids completely local differential computations involving contour points and represents boundary change as a regional integral:

$$\text{Dist}(\partial G, \partial S) \approx 2 \int_{\Delta S} D_G(q) dq \tag{4.3}$$

where ΔS denotes the region between the two contours and $D_G : \Omega \rightarrow \mathbb{R}^+$ is a *distance map* with respect to boundary ∂G, i.e., $D_G(q)$ evaluates the distance between point $q \in \Omega$ and the nearest point $z_{\partial G}(q)$ on contour ∂G: $D_G(q) = \|q - z_{\partial G}(q)\|$. Fig. 4.2.b illustrates this integral framework for evaluating the boundary distance in Eq. (4.2). To clarify approximation (4.3), notice that integrating the distance map $2D_G(q)$ over the normal segment connecting a point p on ∂G and $y_{\partial S}(p)$ yields $\|y_{\partial S}(p) - p\|^2$, via the following variable change:

$$\int_p^{y_{\partial S}(p)} 2D_G(q) dq = \int_0^{\|y_{\partial S}(p) - p\|} 2D_G dD_G = \|y_{\partial S}(p) - p\|^2$$

Thus, from approximation (4.3), the non-symmetric L_2 distance between contours in Eq. (4.2) can be expressed as a sum of regional integrals based on a *level set* representation of boundary ∂G:

$$\frac{1}{2}\text{Dist}(\partial G, \partial S) = \int_S \phi_G(q) dq - \int_G \phi_G(q) dq = \int_\Omega \phi_G(q) s(q) dq - \int_\Omega \phi_G(q) g(q) dq \tag{4.4}$$

where $s : \Omega \rightarrow \{0, 1\}$ is binary indicator function of region S: $s(q) = 1$ if $q \in S$ belongs to the target and 0 otherwise. $\phi_G : \Omega \rightarrow \mathbb{R}$ denotes the level set representation of boundary ∂G: $\phi_G(q) = -D_G(q)$ if $q \in G$ and $\phi_G(q) = D_G(q)$ otherwise. Now, for $S = S_\theta$, i.e., replacing binary variables $s(q)$ in Eq. (4.4) by the softmax probability outputs of the network $s_\theta(q)$, we obtain the following boundary loss which, up to a constant independent of θ, approximates boundary distance $\text{Dist}(\partial G, \partial S_\theta)$:

$$\mathcal{L}_B(\theta) = \int_\Omega \phi_G(q) s_\theta(q) dq \tag{4.5}$$

Notice that we omitted the last term in Eq. (4.4) as it is independent of network parameters. The level set function ϕ_G is pre-computed directly from the ground-truth region G. In practice, our boundary loss in Eq. (4.5) is the sum of linear functions of the regional softmax probability outputs of the network. Therefore, it can be easily combined with standard regional losses ($\mathcal{L}_R$) and implemented with any existing deep network architecture for N-D segmentation:

$$\mathcal{L}_R(\theta) + \alpha \mathcal{L}_B(\theta), \tag{4.6}$$

where $\alpha \in \mathbb{R}$ is a parameter balancing the two losses.

It is worth noting that our boundary loss uses ground-truth boundary information via pre-computed level-set function $\phi_G(q)$, which encodes the distance between each point q and ∂G. In Eq. (4.5), the softmax for each point q is weighted by the distance function. Such distance-to-boundary information is omitted in widely used regional losses, where all the points within a given region are treated equally, independently of their distances from the boundary.

Notice that the global minimum (smallest possible value) of our boundary loss (4.5) is reached when all the negative values in the distance function are included in the sum (i.e., the softmax predictions for the pixels within the ground-truth foreground are equal to 1) and all the positive values are omitted (i.e., the softmax predictions within the background are equal to zero). This means that the global optimum is reached for softmax predictions that correspond exactly to the ground truth, which confirms the meaningfulness of our boundary loss. It is also worth

noticing that the gradient of our loss is ϕ_G multiplied the gradient of the softmax predictions. This results in negative factors for the pixels in G, which encourages s_θ to increase during SGD, with the magnitude (strength) of the factors depending on the distance between the pixel and the ground-truth boundary (the further the pixel from the boundary, the higher the magnitude of the factor). Positive factors for pixels within the background ($\Omega \setminus G$) encourage the softmax predictions to decrease.

As we will see in our experiments, it is important to use our boundary loss in conjunction with a regional loss for the following technical facts. As discussed earlier, the global optimum of our boundary loss corresponds to a strictly negative value, with the softmax probabilities yielding a non-empty foreground region. However, an empty foreground, with approximately null values of the softmax probabilities almost everywhere, corresponds to very low gradients. Therefore, this trivial solution is close to a local minimum or a saddle point. This is why we integrate our boundary loss with a regional loss: the regional loss guides training during the first epochs and avoids getting stuck in such trivial solutions. In the next section, we will discuss various scheduling strategies for updating the weight of the boundary loss during training, with the boundary loss becoming very dominant, almost acting alone, towards the end of the training process. It is also worth noting that this behaviour of boundary terms is conceptually similar to the behaviour of classical and popular contour-based energies for segmentation, e.g., level set Geodesic Active Contours (GAC) Caselles, Kimmel & Sapiro (1997) or discrete Markov Random Fields (MRFs) for boundary regularization and edge alignment Boykov & Funka-Lea (2006), which require additional regional terms to avoid trivial empty-region solutions.

4.3 Experiments

In this section, we perform two sets of experiments. First, we perform comprehensive evaluations demonstrating to positive effect of integrating our boundary loss with different regional losses $\mathcal{L}_R$. Then, we perform a study on the different strategies for selecting and scheduling weight α in (4.6), showing its impact on performances and good default values for new applications.

4.3.1 Datasets

To evaluate the proposed boundary loss, we selected two challenging brain lesion segmentation tasks, each corresponding to highly unbalanced classes.

ISLES

The training dataset provided by the ISLES organizers is composed of 94 ischemic stroke lesion multi-modal scans. In our experiments, we split this dataset into training and validation sets containing 74 and 20 examples, respectively. Each scan contains Diffusion maps (DWI) and Perfusion maps (CBF, MTT, CBV, Tmax and CTP source data), as well as the manual ground-truth segmentation. The spatial resolution goes from $0.8mm \times 0.8mm \times 4mm$ to $1mm \times 1mm \times 12mm$. More details can be found in the ISLES website[2].

WMH

The public dataset of the White Matter Hyperintensities (WMH)[3] MICCAI 2017 challenge contains 60 3D T1-weighted scans and 2D multi-slice FLAIR acquired from multiple vendors and scanners in three different hospitals. The spatial resolution goes from $0.95mm \times 0.95mm \times 3mm$ to $1.21mm \times 1mm \times 3mm$ for each volume. In addition, the ground truth for the 60 scans is provided. From the whole set, 50 scans were used for training, and the remaining 10 for validation.

4.3.2 Compared losses

As stated previously, our proposed boundary loss can be combined with any standard regional loss. In the following experiments, we evaluated different popular ones:

[2]http://www.isles-challenge.org
[3]http://wmh.isi.uu.nl

GDL Sudre *et al.* (2017)

We use the binary case of this loss, described in Equation (4.1). This is also the baseline loss that we use for the experiments on the selection of α. An important advantage of this loss is that it is hyper-parameter free.

Distance weighted cross-entropy Ronneberger *et al.* (2015)

UNet original paper proposed this loss as a way to integrate spatial information during the training. It is a modified weighted cross-entropy loss, where the weight for each pixel depends both on the class distribution, and its distance to the two cells closest boundaries. We adapted it for our case, where we take into account only one distance:

$$\mathcal{L}_{\text{UNET}}(\theta) = -\int_C \int_\Omega u_c(p) \log s_\theta^c(p) dp dc,$$

where C is the set of classes and $s_\theta^c(p)$ are the network predictions for class c. $u_c(p)$ is defined as:

$$u_c(p) = g_c(p) \left[w_c + w_0 e^{\frac{-D_G(p)^2}{2\sigma^2}} \right],$$

where $w_c = \frac{\int_\Omega g_c(p) dp}{|\Omega|}$, and $w_0 = 10$ and $\sigma = 5$ are two hyper-parameters. We kept the paper's default values.

Focal loss Lin *et al.* (2018)

The idea of this loss is to give hard examples a more important weight:

$$\mathcal{L}_{\text{FOCAL}} = -\int_C \int_\Omega (1 - s_\theta^c(p))^\gamma g_c(p) \log s_\theta^c(p) dp dc,$$

with $\gamma = 2$ as default hyper-parameter. Therefore, during training, pixels correctly classified with a high confidence will have little to no influence.

Hausdorff loss Karimi & Salcudean (2019)

This closely related loss is also designed to minimize some distance between the two boundaries, but through a different path. We refer to this loss as $\mathcal{L}_{HD}$.

$$\mathcal{L}_{HD} = \int_{\Omega} (g(p) - s_\theta(p))^2 (D_G(p)^\beta + D_S(p)^\beta) dp,$$

where D_S denotes the distance function from predicted boundary S, after thresholding s_θ. β is a hyper-parameter, which the authors of Karimi & Salcudean (2019) set to 2 following a grid search. Unlike our boundary loss, computing D_S cannot be done in a single step before training. The distance needs to be re-computed at each epoch during training, for all the images. It also requires to store the whole volume Ω in memory, as we cannot compute the distance map for only a subset of Ω. These might be important computational and memory limitations, more so when dealing with large images, as is the case for 3D distance maps.

4.3.3 2D and 3D distance maps

While the main experiments resort to a distance map computed from each individual 2D slice, we evaluate the proposed boundary loss with a distance map computed from the whole initial 3D segmentation mask. Equation (4.5) enables us to have only a subset of Ω at each update, making it possible to use a 3D distance map with mini-batches of 2D slices.

4.3.4 Selection of alpha

We study several strategies for selecting α, and its effect on the performances. On top of a constant pre-selected α, we evaluated simple scheduling strategies to update it during the training.

Constant α

The simplest method would be to use a constant during the whole training, but this might require careful tuning of its value.

Increase α

We start with a low value of $\alpha > 0$, and increase it gradually at the end of each epoch. The weight of the regional loss $\mathcal{L}_R$ remains constant over time. At the end of the training, the two losses have the same weight.

Rebalance α

First we rewrite our combined loss as $(1 - \alpha)\mathcal{L}_R + \alpha\mathcal{L}_B$. As for the increase strategy, we start with a low $\alpha > 0$, and increase it over time. In this way, we give more importance to the regional loss term at the beginning while gradually increasing the impact of the boundary loss term. Note that we make sure that the weight for $\mathcal{L}_R$ never reaches 0; the two losses are used at all times during training.

4.3.5 Implementation details

Data pre-processing

While the scans are provided as 3D images, we process them as a stack of independent 2D images, which are fed into the network. In fact, the scans in some datasets, such as ISLES, contain between 2 and 16 slices, making them ill-suited for 3D convolutions in those cases. The scans were normalized between 0 and 1 before being saved as a set of 2D matrices, and re-scaled to 256×256 pixels if needed. When several modalities were available, all of them were concatenated before being used as input to the network. We did not use any data augmentation in our experiments.

Architecture and training

We employed UNet Ronneberger *et al.* (2015) as deep learning architecture in our experiments. To train our model, we employed Adam optimizer, with a learning rate of 0.001 and a batch size equal to 8. The learning rate is halved if the validation performances do not improve during 20 epochs. We did not use early stopping.

To compute the level set function ϕ_G in Eq. (4.5), we used standard SciPy functions[4]. Note that, for slices containing only the background region, we used a zero-distance map, assuming that the regional loss is sufficient in those cases. For the increase and rebalance α scheduling strategies, we start with $\alpha = 0.01$ and increase it by 0.01 at the end of each epoch. For all the experiments comparing different losses, we use the same rebalance strategy, with the same hyper-parameters.

In addition, we evaluated the performance when the boundary loss is the only objective, i.e., $\alpha = 0$.

For our implementation, we used PyTorch Paszke *et al.* (2017), and ran the experiments on a machine equipped with an NVIDIA GTX 1080 Ti GPU with 11GBs of memory. Our code (data pre-processing, training and testing scripts) is publicly available[5]. As Karimi & Salcudean (2019) did not release their code, we relied on the re-implementation from Ma, Wei, Zhang, Wang, Lv, Zhu, Chen, Liu, Peng, Wang et al. (2020)[6].

Evaluation

For evaluation purposes, we employ the common Dice Similarity Coefficient (DSC) and modified Hausdorff Distance[7] (HD95) metrics.

[4]https://docs.scipy.org/doc/scipy-0.14.0/reference/generated/scipy.ndimage.morphology.distance_transform_edt.html

[5]https://github.com/LIVIAETS/surface-loss

[6]https://github.com/JunMa11/SegWithDistMap

[7]We report the 95th percentile distance value instead of the maximum-distance value.

4.3.6 Results

4.3.6.1 Comparison of regional losses

In this section, we detail the results that we obtained when using different regional losses $\mathcal{L}_R$.

Quantative evaluation

Table 4.1 reports the DSC and HD performances for our experiments using four different choices of $\mathcal{L}_R$, with each regional term used either alone or in conjunction with our boundary loss in Eq. (4.6), on the ISLES and WMH datasets. In most of the settings, adding the boundary loss during training improves the performances, as reflected in the significantly better DSC and HD values. For instance, on the ISLES segmentation task, adding the boundary loss yielded about 13% improvement in DSC over using Generalized Dice loss alone, and about 3% improvement over using UNet cross-entropy or focal loss alone. The discrepancy of the improvements the boundary loss brings might be due to the difference in the difficulty of the tackled tasks. The more difficult the tasks (i.e., when regional terms have difficulty achieving good performances), the larger the gain boundary loss brings (as it complements regional information). GDL/ISLES is a noticeable case, where boundary loss corrected substantially the performance of the GDL regional loss, making it the winning competitor (although, without boundary information, it is the worse-performing regional loss).

The mixed results with the UNet cross-entropy (improvement on ISLES, but stall on WMH), and the difference on the HD95 metrics can potentially be explained by a toxic interplay between the two losses: both of them are trying to use the distance from the boundary information, potentially counter-acting each others, and introducing instability.

Computing the distance map from the 3D volume rather than from the 2D slices gives a small boost in performance (about 1% DSC), and is more noticeable on the training curve for WMH (Figure 4.3). This difference could be explained by the spacing between the slices on the z axis: they are quite close (and correlated) in the case of WMH. However, in the case of ISLES, the

big spacing (around 1cm) makes slices quite independent. Adding 3D information in this case is less helpful.

While the Hausdorff loss Karimi & Salcudean (2019) also improves the results over the GDL alone (around 7% on ISLES), its performance is not always at the same level as boundary loss (similar performance on WMH, but lower on ISLES). This is consistent with the findings of Ma *et al.* (2020), which found that the differences in performances are dataset dependent.

Table 4.1 Average DSC and HD95 values (and standard deviation over three independent runs) achieved on the validation subset. Best results highlighted in bold.

Loss	ISLES		WMH	
	DSC	HD95 (mm)	DSC	HD95 (mm)
$\mathcal{L}_B$	NA	NA	NA	NA
$\mathcal{L}_{HD}$	NA	NA	0.638 (NA)	4.578 (NA)
GDL	0.511 (0.016)	5.320 (1.742)	0.768 (0.051)	3.634 (2.570)
w/ $\mathcal{L}_B$ (2D)	0.644 (0.026)	4.795 (3.712)	0.793 (0.006)	2.039 (1.834)
w/ $\mathcal{L}_B$ (3D)	**0.659 (0.001)**	2.725 (2.196)	**0.818 (0.003)**	**1.702 (1.982)**
w/ $\mathcal{L}_{HD}$	0.582 (0.015)	4.126 (1.634)	0.805 (0.015)	2.151 (2.100)
UNet cross-entropy Ronneberger *et al.* (2015)	0.608 (0.025)	4.572 (0.675)	0.757 (0.015)	4.355 (3.388)
w/ $\mathcal{L}_B$ (2D)	0.631 (0.016)	5.961 (2.291)	0.756 (0.022)	2.887 (2.629)
Focal loss Lin *et al.* (2018)	0.631 (0.046)	4.989 (2.775)	0.808 (0.026)	1.816 (1.370)
w/ $\mathcal{L}_B$ (2D)	0.650 (0.019)	**1.770 (0.549)**	0.786 (0.031)	2.258 (2.513)

Using the boundary loss alone does not yield the same competitive results as a joint loss (i.e., boundary and region), making the network collapse quickly into empty foreground regions, i.e., softmax predictions close to zero[8]. We believe that this is due to the following technical facts. In theory, the global optimum of the boundary loss corresponds to a negative value, as a perfect overlap sums only over the negative values of the distance map. In this case, the softmax probabilities correspond to a non-empty foreground. However, an empty foreground (null values of the softmax probabilities almost everywhere) corresponds to low gradients. Therefore, this trivial solution is close a local minimum or a saddle point. This is not the case when we use our boundary loss in conjunction with a regional loss, which guides the training during the first epochs and avoids getting stuck in such a trivial solution. The scheduling method then increases the weight of the boundary loss, with the latter becoming very dominant towards

[8]For this reason, we do not report metrics in this case, as it would be meaningless.

the end of the training process. This behaviour of boundary terms is conceptually similar to the behaviour of classical and popular contour-based energies for level set segmentation, e.g., geodesic active contours Caselles *et al.* (1997), which also require additional regional terms to avoid trivial solutions (i.e., empty foreground regions).

The learning curves depicted in Figure 4.3 show the gap in performances between using a regional loss $\mathcal{L}_R$ alone and when augmented with our boundary loss, for different choices of $\mathcal{L}_R$. In most of the settings, the difference becomes significant at convergence. This behaviour is most visible when $\mathcal{L}_R = \mathcal{L}_{\text{GDL}}$, and is consistent for both metrics and both dataset, which clearly shows the benefits of employing the proposed boundary loss term.

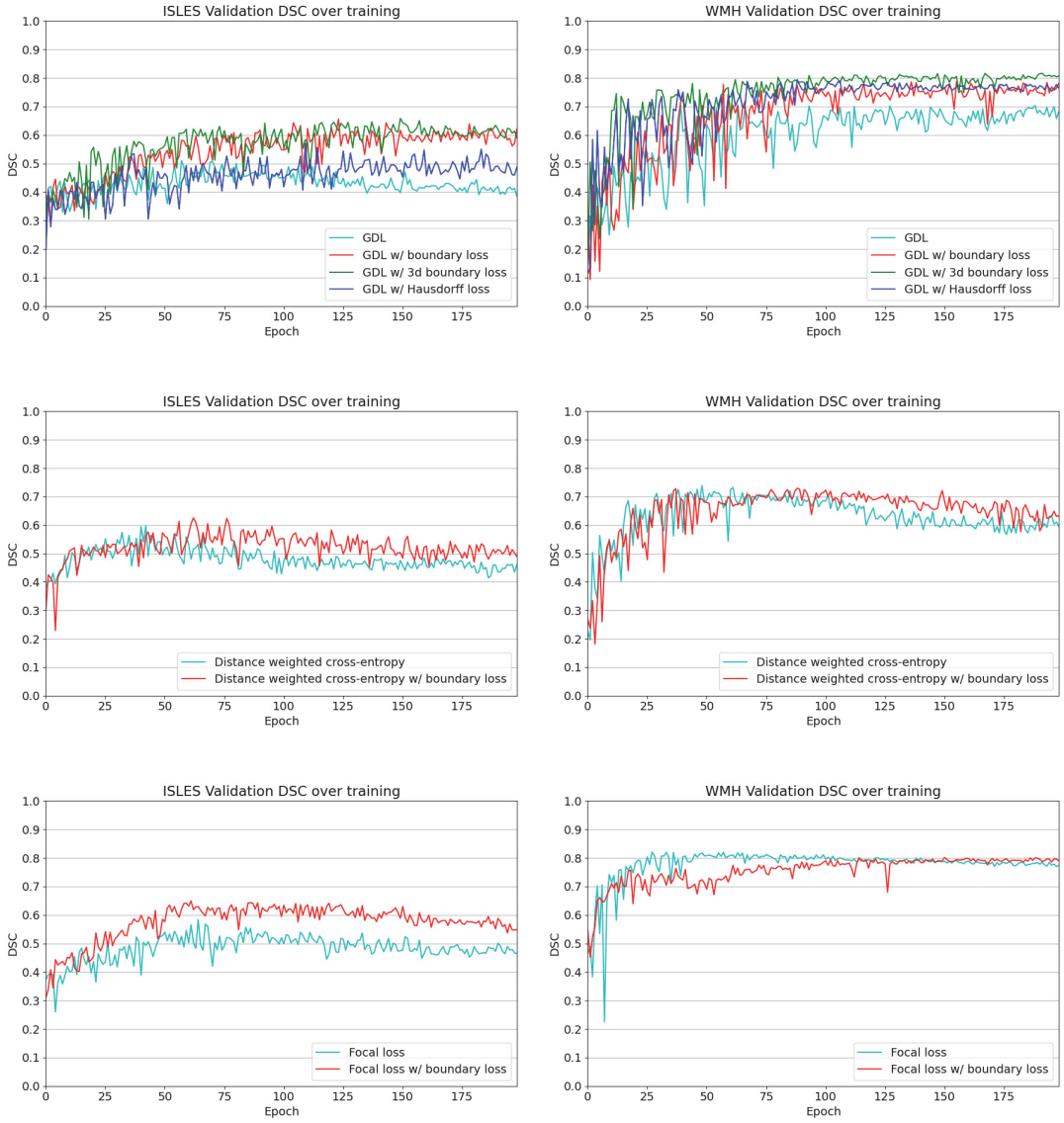


Figure 4.3 Evolution of DSC values on validation subsets, for different base losses, on both ISLES and WMH. Best viewed in colors.

4.3.6.1.1 Qualitative evaluation

Qualitative results are depicted in Fig. 4.4. Inspecting these results visually, we can observe that there are two major types of improvements when employing the proposed boundary loss. First, as the methods based on DSC losses, such as GDL, do not use spatial information, prediction errors are treated equally. This means that the errors for pixels/voxels in an already detected object have the same importance as the errors produced in completely missed objects. On the contrary, as our boundary loss is based on the distance map from the ground-truth boundary ∂G, it will penalize much more such cases, helping to recover small and far regions. This effect is best illustrated in Fig. 4.1 and Fig. 4.4 (third row). False positives (first row in Fig. 4.4) will be far away from the closest foreground, getting a much higher penalty than with the GDL alone. This helps in reducing the number of false positives. Additional qualitative results for other base losses, and their combination with the proposed boundary loss, are depicted in Figures 4.5, 4.6. These figures also show failure cases (*last column*) of the boundary loss.

4.3.6.1.2 Computational complexity

It is worth mentioning that, as the proposed boundary loss term involves an element-wise product between two matrices—i.e., the pre-computed level-set function ϕ_G and the softmax output $s_\theta(p)$—the complexity that it adds is negligible as showed in Table 4.2. Contrary, the Hausdorff loss Karimi & Salcudean (2019) introduces around 10% of slowdown in the training process. This will be further magnified if we generalize to multi-class problems, where an individual distance map should be computed for each class.

4.3.6.2 Selection of alpha

Table 4.3 reports the performances of the proposed approach on the ISLES segmentation task for different α values and scheduling techniques. Figure 4.3 shows a subset of the learning curves related to α selection strategies in Table 4.3. This is an indication that, while our boundary loss can benefit from a tuned balance between the two losses, even a sub-optimal

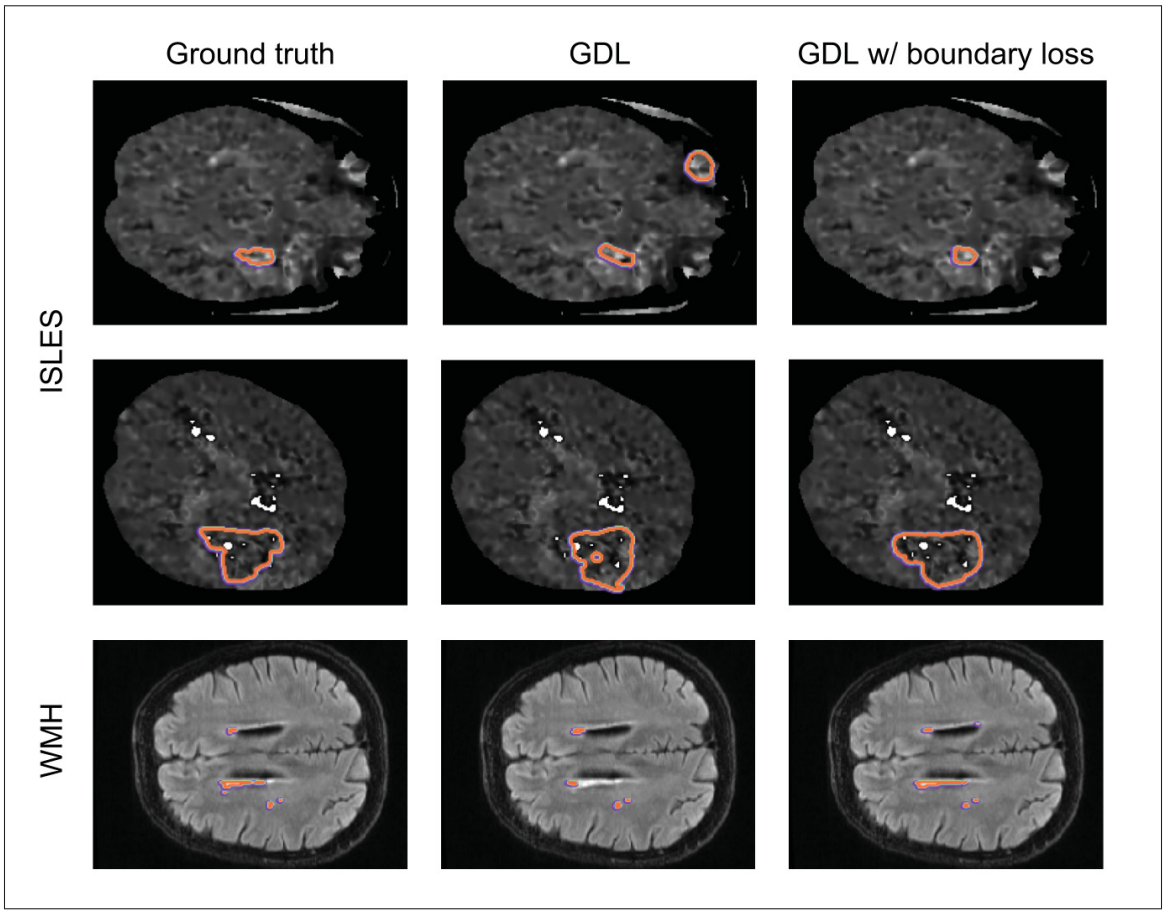


Figure 4.4 Visual comparison on two different datasets from the validation set.

Table 4.2 Training time required by different losses.
We report the average and standard deviation batch time
in seconds for each method.

Loss	Time (s) per batch	
	ISLES (Batch size = 4)	WMH (Batch size = 8)
GDL	0.187 (0.129)	0.345 (0.132)
w/ $\mathcal{L}_B$	0.190 (0.128)	0.345 (0.129)
w/ $\mathcal{L}_{HD}$	0.210 (0.108)	0.392 (0.092)

α can already provide improvement over the regional loss alone. Observe that increasing the weight of constant α yields better performances, up to a certain value, with the performances decreasing starting from $\alpha = 1.5$. With $\alpha = 2$,the performance is similar to a network trained

with the boundary loss alone. In contrast, using any of the two proposed scheduling strategies (increasing α or re-balancing) yields better results than any constant α, without having to explore many configurations.

From the learning curves (Figure 4.7), we can notice that the GDL alone and the GDL with a small constant $\alpha = 0.001$ have a similar training DSC over time, but that their validation DSC are significantly different. A similar behaviour can be observed by examining the results with constant $\alpha = 1$ and the rebalanced α: while the rebalancing training DSC is slightly higher during the whole training, the validation DSC becomes significantly better around half the training time, where the high constant α performances starts decreasing.

The rebalancing strategy was used in all other experiments, and as showed in Table 4.1, proved to be a good default strategy to integrate the boundary loss with another regional loss.

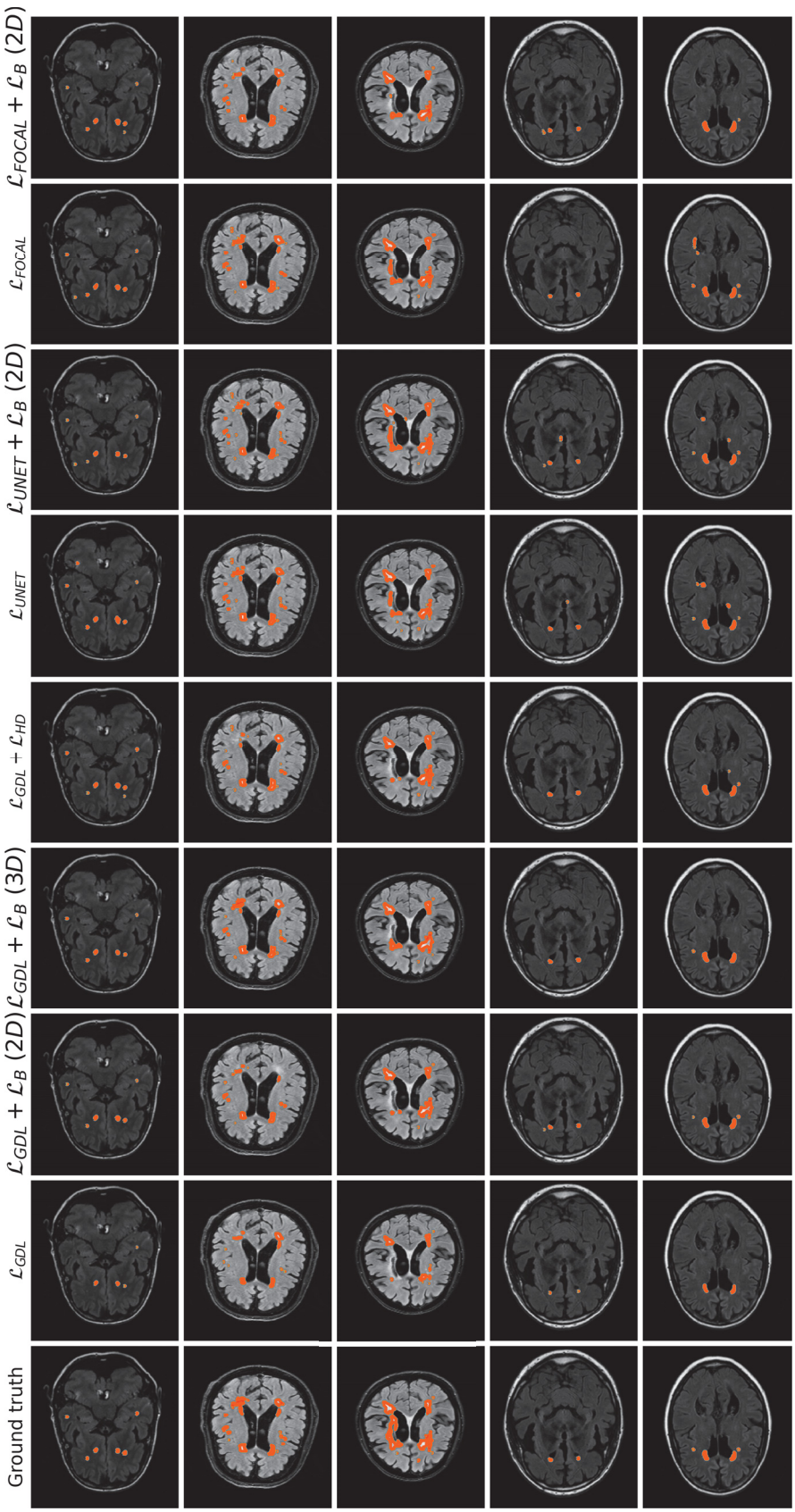


Figure 4.5 Visual comparison on the WMH dataset for different training losses. The last column depicts a failure case, where the proposed loss does not enhance the regional loss performance. Best viewed in colors.

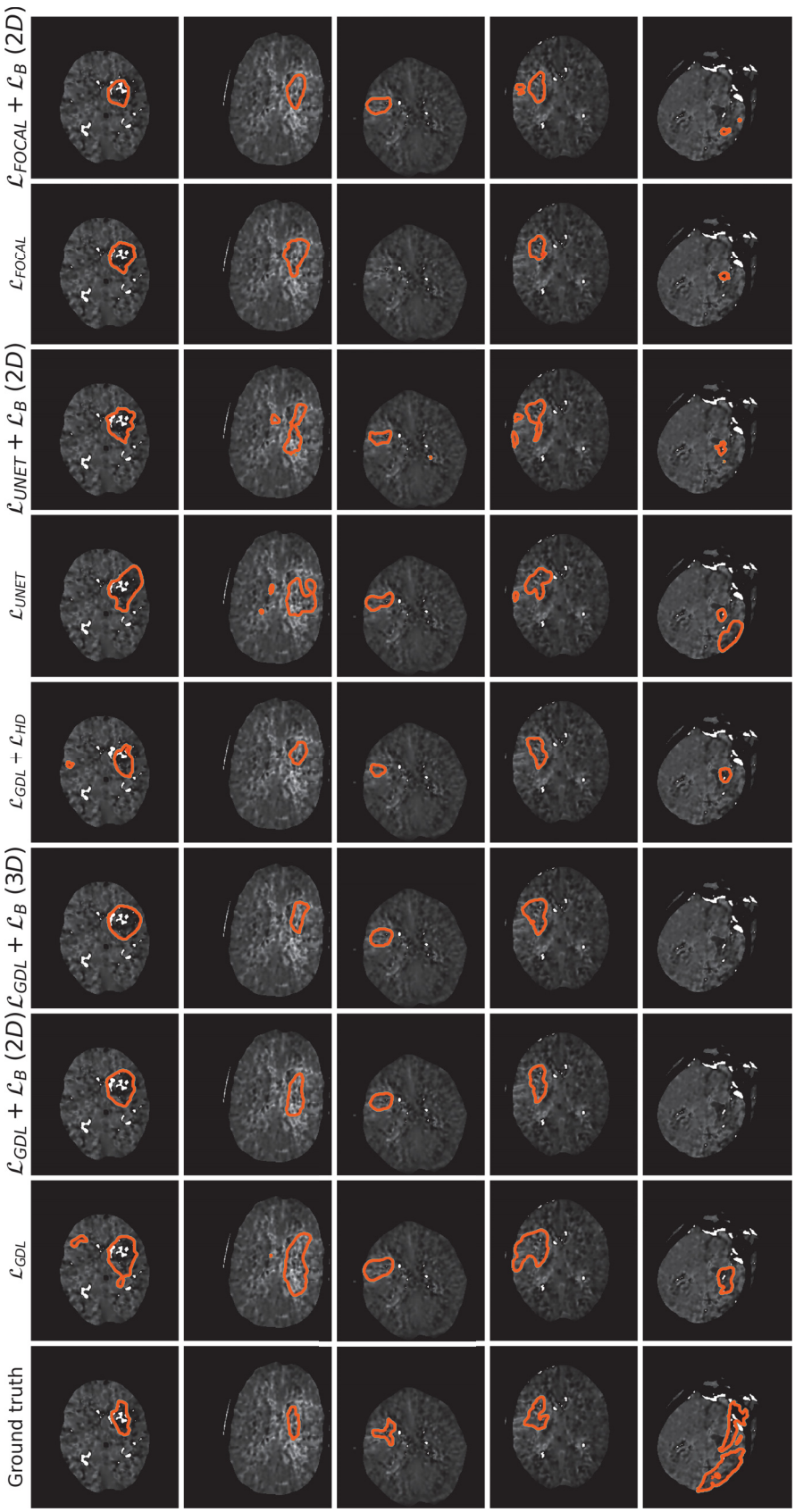


Figure 4.6 Visual comparison on the ISLES dataset for different training losses. The last column depicts a failure case, where the proposed loss does not enhance the regional loss performance. Best viewed in colors.

Table 4.3 Results on ISLES validation set for different α.

Strategy		ISLES	
		DSC	HD95
GDL only		0.511 (0.016)	5.320 (1.742)
Constant α	$\alpha = 0.001$	0.545 (0.020)	4.778 (1.546)
	$\alpha = 0.01$	0.566 (0.019)	5.052 (1.395)
	$\alpha = 0.05$	0.606 (0.015)	5.326 (1.712)
	$\alpha = 0.1$	0.605 (0.010)	5.762 (1.782)
	$\alpha = 0.5$	0.604 (0.006)	9.234 (10.463)
	$\alpha = 1$	*0.628* (0.023)	*2.462* (0.706)
	$\alpha = 1.5$	0.565 (0.074)	3.335 (1.164)
	$\alpha = 2$	0.549 (0.084)	20.275 (16.603)
Increase α		*0.622* (0.004)	*4.952* (1.773)
Rebalance α		***0.644*** (0.026)	***4.795*** **(3.712)**

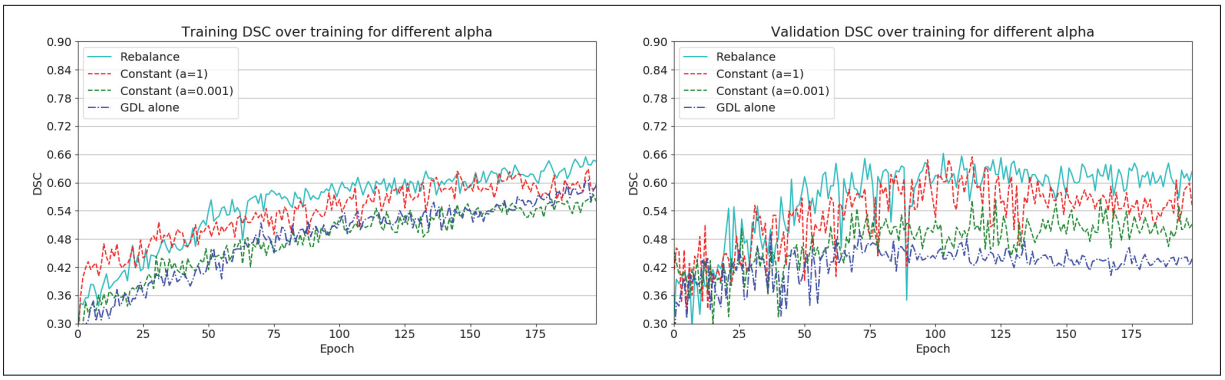


Figure 4.7 Comparison of the training and validation DSC curves for different α selection strategies. For readability, not all settings from Table 4.3 have been included. Best viewed in colors.

4.4 Conclusion and future works

We proposed a boundary loss term that can be easily combined with any standard regional loss, to tackle segmentation tasks in highly unbalanced scenarios. Furthermore, the proposed term can be implemented with any existing deep network architecture and for any N-D segmentation problem. Our experiments on two challenging and highly unbalanced datasets demonstrated

the benefits of including our boundary loss during training. It consistently improved the performances, and by a large margin on one data set, with enhanced training stability.

In this work, we evaluated the proposed boundary loss in the context of class imbalance. However, there are other interesting avenues for extending and evaluating our approach. For instance, our boundary loss has a spatial regularization effect because it is based on distance-to-boundary information. In particular, we observed experimentally that it yield contours, which are, typically, smoother than those obtained with regional losses. Focused on the important problem of unbalanced segmentation, our experiments did not fully investigate the benefits of such a spatial regularization. An interesting future research avenue will be to explore such a regularization effect in applications with challenging imaging noise, which may prevent regional losses from generating smooth contours, e.g., ultrasound imaging. Another limitation of our formulation and experiments is that they were limited to binary (two-region) segmentation problems. It will be interesting to investigate extensions of boundary loss to the multi-region scenario, with competing distance maps from multiple structures and various/complex topological constraints (e.g., one structure fully included within another).

CHAPTER 5

CURRICULUM SEMI-SUPERVISED SEGMENTATION

Hoel Kervadec[a], Jose Dolz[b], Eric Granger[a], Ismail Ben Ayed[a]

[a]Département de génie des systèmes, ÉTS Montréal, QC, Canada,
[b]Département de génie logiciel et des TI, ÉTS Montréal, QC, Canada,

Presented as a poster at MICCAI 2019, Shenzhen, China.

Abstract

This study investigates a curriculum-style strategy for semi-supervised CNN segmentation, which devises a regression network to learn image-level information such as the size of the target region. These regressions are used to effectively regularize the segmentation network, constraining the softmax predictions of the unlabeled images to match the inferred label distributions. Our framework is based on inequality constraints, which tolerate uncertainties in the inferred knowledge, e.g., regressed region size. It can be used for a large variety of region attributes. We evaluated our approach for left ventricle segmentation in magnetic resonance images (MRI), and compared it to standard proposal-based semi-supervision strategies. Our method achieves competitive results, leveraging unlabeled data in a more efficient manner and approaching full-supervision performance.

5.1 Introduction

In the recent years , deep learning architectures, and particularly convolutional neural networks (CNNs), have achieved state-of-the-art performances in a breadth of visual recognition tasks. These architectures currently dominate the literature in medical image segmentation Litjens *et al.* (2017). The generalization capabilities of these networks typically rely on large and annotated datasets, which, in the case of segmentation, consist of precise pixel-level annotations. Obtaining expert annotations in medical images is a costly process that also requires clinical expertise. The lack of large annotated datasets has driven research in deep segmentation models

that rely on reduced supervision for training, such as weakly Kervadec *et al.* (2019b); Khoreva *et al.* (2017); Lin *et al.* (2016); Pathak *et al.* (2015a) or semi-supervised Bai *et al.* (2017); Sedai, Mahapatra, Hewavitharanage, Maetschke & Garnavi (2017) learning. These strategies assume that annotations are limited or coarse, such as image-level tags Papandreou *et al.* (2015); Pathak *et al.* (2015a), scribbles Tang *et al.* (2018b) or bounding-boxes Rajchl *et al.* (2017).

In this paper, we focus on semi-supervised learning, a common scenario in medical imaging, where a small set of images are assumed to be fully annotated, but an abundance of unlabeled images is available. Recent progress of these techniques in medical image segmentation has been bolstered by deep learning Bai *et al.* (2017); Baur, Albarqouni & Navab (2017); Ganaye, Sdika & Benoit-Cattin (2018); Nie, Gao, Wang & Shen (2018); Sedai *et al.* (2017); Zhou, Wang, Tang, Bai, Shen, Fishman & Yuille (2019b). Self-training is a common semi-supervised learning strategy, which consists of employing reliable predictions generated by a deep learning architecture to re-train it, thereby augmenting the training set with these predictions as pseudo-labels Bai *et al.* (2017); Pathak *et al.* (2015a); Rajchl *et al.* (2017). Although this approach can leverage unlabeled images, one of its main drawbacks is that early mistakes are propagated back to the network, being re-amplified during training Chapelle, Scholkopf & Zien (2009); Zhu & Goldberg (2009). Several techniques were proposed to overcome this issue, such as co-training Zhou *et al.* (2019b) and adversarial learning Dong, Kampffmeyer, Liang, Wang, Dai & Xing (2018); Mondal, Dolz & Desrosiers (2018); Zhang, Yang, Chen, Fredericksen, Hughes & Chen (2017b). Nevertheless, with these approaches, training typically involves several networks, or multiple objective functions, which might hamper the convergence of such models.

Alternatively, some weakly supervised segmentation approaches have been proposed to constrain the network predictions with global label statistics, for example, in the form of target-region size Jia *et al.* (2017); Kervadec *et al.* (2019b); Pathak *et al.* (2015a). For instance, Jia et *.al* Jia *et al.* (2017) employed an $\mathcal{L}_2$ penalty to impose equality constraints on the size of the target regions in the context of histopathology image segmentation. However, their formulation requires the exact knowledge of region size, which limits its applicability. More recently,

Kervadec et *al.* Kervadec *et al.* (2019b) proposed using inequality constraints, which provide more flexibility, and significantly improves performance compared to cases where learning relies on partial image labels in the form of scribbles. Nevertheless, the values used to bound network predictions in Kervadec *et al.* (2019b) are derived from manual annotations, which is a limiting assumption. Another closely related work is the curriculum learning strategy proposed in the context of unsupervised domain adaptation for urban images in Zhang *et al.* (2017a). In this case, the authors proposed to match global label distributions over source (*labelled*) and target (*unlabelled*) images by minimizing the KL-divergence between distributions. Finally, it is worth noting that the semi-supervised learning technique in Ganaye *et al.* (2018) embeds semantic constraints on the adjacency graph of a given region.

Inspired by this research, we propose a curriculum-style strategy for deep semi-supervised segmentation, which employs a regression network to predict image-level information such as the size of the target region. These regressions are used to effectively regularize the segmentation network, enforcing the predictions for the unlabeled images to match the inferred label distributions. Contrary to Zhang *et al.* (2017a), our framework uses inequality constraints, which provides greater flexibility, allowing uncertainty in the inferred knowledge, e.g., regressed region size. Another important difference is that the proposed framework can be used for a large variety of region attributes (e.g., shape moments). We evaluated our approach in the task of left ventricle segmentation in magnetic resonance images (MRI), and compared it to standard proposal-based semi-supervision strategies. Our method achieves very competitive results, leveraging unlabeled data in a more efficient manner and approaching full-supervision performance. We made our code publicly available[1].

5.2 Self-training for semi-supervised segmentation

Let $X : \Omega \subset \mathbb{R}^{2,3} \rightarrow \mathbb{R}$ denotes a training image, with Ω its spatial domain. Consider a semi-supervised scenario with two subsets: $\mathcal{S} = \{(X_i, Y_i)\}_{i=1,\dots,n}$ which contains a set of images X_i and their corresponding pixel-wise ground-truth labels Y_i, and $\mathcal{U} = \{X_j\}_{j=1,\dots,m}$ a set of

[1]https://github.com/LIVIAETS/semi_curriculum

unlabeled images, with $m \gg n$. In the fully supervised setting, training is formulated as minimizing the following loss with respect to network parameters $\boldsymbol{\theta}$:

$$\mathcal{L}_Y(\boldsymbol{\theta}) = -\sum_{i \in \mathcal{S}} \sum_{p \in \Omega} Y_{i,p} \log S(X_i|\boldsymbol{\theta})_p \tag{5.1}$$

where $S(X_i|\boldsymbol{\theta})_p$ represents a vector of softmax probabilities generated by the CNN at each pixel p and image i. To simplify the presentation, we consider the two-region segmentation scenario (i.e., two classes), with ground-truth binary labels $Y_{i,p}$ taking values in $\{0, 1\}$, 1 indicating the target region (foreground) and 0 indicating the background. However, our formulation can be easily extended to the multi-region case. Common approaches for semi-supervised segmentation Bai *et al.* (2017); Papandreou *et al.* (2015) generate fake full masks (segmentation proposals) $\tilde{Y}$ for the unlabeled images, which are then used iteratively for network training by adding a standard cross-entropy loss of the form in Eq. (5.1): $\min_{\boldsymbol{\theta}} \mathcal{L}_Y(\boldsymbol{\theta}) + \mathcal{L}_{\tilde{Y}}(\boldsymbol{\theta})$. The process consists of alternating segmentation-proposal generation and updating network parameters using both labeled data and the new generated masks. Typically such proposals are refined with additional priors suh as dense CRF Tang *et al.* (2018b). However, errors in such proposals may mislead training as the cross-entropy loss is minimized over mislabled points and, reinforcing early mistakes during training, as is well-known in the semi-supervised learning literature Chapelle *et al.* (2009); Zhu & Goldberg (2009).

5.3 Curriculum semi-supervised learning

The general principle of curriculum learning consists of solving easy tasks first in order to infer some necessary properties about the unlabeled images. In particular, the first task is to learn image-level properties, e.g. the size of the target region, which is easier than learning pixelwise segmentations within an exponentially large label space. Then, we use such image-level properties to facilitate segmentation via constrained CNNs. Fig. 5.1 depicts an illustration of our curriculum semi-supervised segmentation. We first use an auxiliary network that predicts the target-region size for a given image. Particularly, we train a regression network R (with

parameters $\tilde{\boldsymbol{\theta}}$) by solving the following minimization problem:

$$\min_{\tilde{\boldsymbol{\theta}}} \sum_{i \in \mathcal{S}} \left(R(X_i|\tilde{\boldsymbol{\theta}}) - \sum_{p \in \Omega} Y_{i,p} \right)^2 . \tag{5.2}$$

This amounts to minimizing the squared difference between the predicted size and the actual region size.

Now we can define our constrained-CNN segmentation problem using auxiliary size predictions $R(X_i|\tilde{\boldsymbol{\theta}})$:

$$\begin{aligned} \min_{\boldsymbol{\theta}} \quad & \mathcal{L}_Y(\boldsymbol{\theta}) \\ \text{s.t.} \quad & (1-\gamma) R(X_i|\tilde{\boldsymbol{\theta}}) \leq \sum_{p \in \Omega} S(X_i|\boldsymbol{\theta})_p \leq (1+\gamma) R(X_i|\tilde{\boldsymbol{\theta}}) \qquad \forall i \in \mathcal{U}, \end{aligned} \tag{5.3}$$

where the inequality constraints impose the learned image-level information (i.e., region size) on the outputs of the segmentation network for unlabeled images, and γ is a hyper-parameter controlling constraints tightness. We use a penalty-based approach Kervadec *et al.* (2019b) for handling the inequality constraints, which accommodates standard stochastic gradient descent. This amounts to replacing the constraints in (5.3) with the following penalty over unlabeled samples:

$$\mathcal{L}_{\mathcal{U}}(\boldsymbol{\theta}) = \sum_{i \in \mathcal{U}} C \left(\sum_{p \in \Omega} S(X_i|\boldsymbol{\theta})_p \right) \tag{5.4}$$

$$C(t) = \begin{cases} (t - (1-\gamma) R(X_i|\tilde{\boldsymbol{\theta}}))^2 & \text{if } t \leq (1-\gamma) R(X_i|\tilde{\boldsymbol{\theta}}) \\ (t - (1+\gamma) R(X_i|\tilde{\boldsymbol{\theta}}))^2 & \text{if } t \geq (1+\gamma) R(X_i|\tilde{\boldsymbol{\theta}}) \\ 0 & \text{otherwise} \end{cases} \tag{5.5}$$

This gives our final unconstrained optimization problem: $\min_{\boldsymbol{\theta}} \mathcal{L}_Y(\boldsymbol{\theta}) + \lambda \mathcal{L}_{\mathcal{U}}(\boldsymbol{\theta})$, with λ a hyper-parameter controlling the relative contribution of each term.

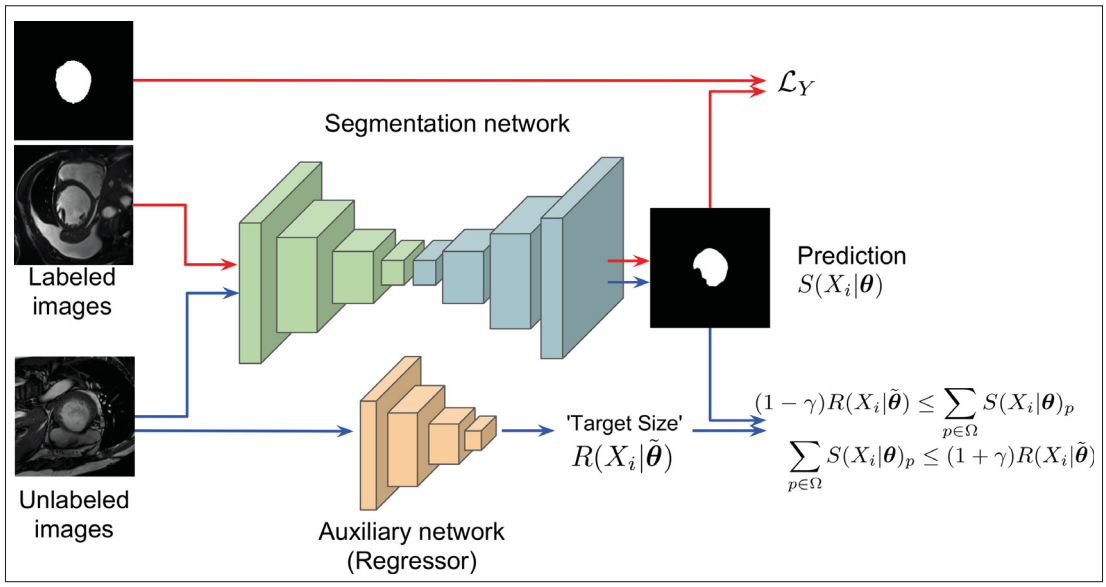


Figure 5.1 Illustration of our curriculum semi-supervised segmentation strategy.

5.4 Experiments

5.4.1 Setup

5.4.1.0.1 Data

Our experiments focused on left ventricular endocardium segmentation. We used the training set from the publicly available data of the 2017 ACDC Challenge Bernard, Lalande, Zotti, Cervenansky, Yang, Heng, Cetin, Lekadir, Camara, Ballester et al. (2018). This set consists of 100 cine magnetic resonance (MR) exams covering well defined pathologies: dilated cardiomyopathy, hypertrophic cardiomyopathy, myocardial infarction with altered left ventricular ejection fraction and abnormal right ventricle. It also included normal subjects. Each exam only contains acquisitions at the diastolic and systolic phases. We sliced and resized the exams into 256×256 images. No additional pre-processing was performed.

5.4.1.0.2 Training

For the experiments, we employed 75 exams for training and the remaining 25 for validation. From the training set, we consider that n images are fully annotated and the pixel-wise annotations of the remaining 75-n images are unknown. The n images, and their corresponding ground truth, are employed to train both the auxiliary size predictor and the main segmentation network, in a separate way. To validate both networks, we split the validation set into two smaller subsets of 5 and 20 exams, respectively. The training set undergoes data augmentation only to train the size regressor, by flipping, mirroring and rotating (up to 45°) the original images, obtaining a training set that is 10 times larger.

5.4.1.0.3 Implementation details

We employed ResNeXt 101 Xie, Girshick, Dollár, Tu & He (2017) as the backbone architecture for our regressor model, with the squared $\mathcal{L}_2$ norm as the objective function. We trained via standard stochastic gradient descent, with a learning rate of 5×10^{-6}, a momentum of 0.9 and a weight decay of 10^{-4}, for 200 epochs. The learning rate was halved at epochs 100 and 150. We used a batch size of 10. We used ENet Paszke *et al.* (2016) as the segmentation network, trained with Adam Kingma & Ba (2015), a learning rate of 5×10^{-4}, $\beta_1 = 0.9$ and $\beta_2 = 0.99$ for 100 epochs. The learning rate was halved if validation DSC did not improve for 20 epochs. We used a batch size of 1, and γ from Eq. (5.4) is set at $\gamma = 0.1$. We did not use any form of post-processing on the network output.

5.4.1.0.4 Comparative Methods

We compare the performance of the proposed semi-supervised curriculum segmentation approach to several models. First, we train a network using only n exams and their corresponding pixel-wise annotations, which is referred to as *FS*. Then, once this model is trained, and following standard proposal-based strategies for semi-supervision, e.g., Bai *et al.* (2017), we perform the inference on the remaining 75-n exams, and include the CNN predictions in the training set,

which serve as pseudo-labels for the non-annotated images (referred to as *Proposals*). In this particular case, the training reduces to minimizing the cross-entropy over all the pixels in the manually annotated images and over the pixels predicted as left-ventricle in the pseudo-labels. Since we investigate how to leverage unlabeled data only by learning from the subset of labeled data, we do not integrate any additional cues during training, such as Conditional Random Fields (CRF)[2]. Finally, we train a model with the exact size derived from the ground truth for each image, as in Kervadec *et al.* (2019b), which will serve as an upper bound, referred to as *Oracle*.

5.4.1.0.5 Evaluation

We resort to the common dice (DSC) overlap metric between the ground truth and the CNN segmentation to evaluate the performances of the segmentation models. More specifically, we report the mean and standard deviation of the validation DSC over the last 50 epochs of training.

5.4.2 Results

We report in Table 5.1 and Fig. 5.2 the quantitative evaluation of the different segmentation models. First, we can observe that integrating the size predicted on unlabeled images by the auxiliary network improves the performance compared to solely training from labeled images. The gap is particularly significant when few annotated images are available, ranging from nearly 15 to 25% of difference in terms of DSC. As more labeled images are available, the proposed strategy still improves the performance of the fully supervised counterpart, but by a smaller margin, which goes from 1 to 3%. Compared to the *Oracle*, our method achieves comparable results as the number of training samples increases. This suggests that, when few annotated patients are available, having a better estimation of the size helps to better regularize the network. It is noteworthy to mention that in the *Oracle*, the exact size is known for each image, which results in extra supervision compared to the proposed method. The *proposals* method achieves the same or worse results than its *FS* counterpart, for all the n values evaluated. These results

[2]Note that the proposal-based methods in Bai *et al.* (2017) use CRF to boost performance.

indicate that n patients are not sufficient to train an auxiliary network that generates usable pseudo-labels, due to the difficulty of the segmentation task. This confirms that training a network on an easier task, e.g., learning the size of the target region, can guide the training in a semi-supervised setting.

Table 5.1 Quantitative results for the different models. Values represent the mean Dice (and standard deviation) over the last 50 epochs.

# of labelled patients	Method			
	FS	Proposals	Proposed	Oracle Kervadec *et al.* (2019b)
5	24.8 (4.9)	8.1 (0.8)	53.1 (3.0)	74.3 (2.5)
10	44.4 (8.3)	43.9 (2.9)	58.5 (3.6)	75.7 (3.9)
20	71.7 (3.2)	49.1 (5.0)	72.7 (1.6)	79.0 (2.5)
30	73.1 (1.7)	62.6 (4.4)	75.4 (1.6)	77.0 (1.9)
40	75.8 (2.4)	68.8 (5.6)	76.3 (2.1)	80.4 (2.1)
75	81.6 (1.9)	NA	NA	NA

Evolution of DSC on the validation set over training for some models is depicted in Fig. 5.3. From these plots, we can observe that the auxiliary network facilitates the training of a harder task, consistently achieving higher performance and better stability than its *FS* counterpart, especially when few labeled images are available. Regarding the instability of the *FS* method, it may be caused by the small number of samples employed for training, with no other source of information that regularizes the network.

Qualitative results are depicted in Fig. 5.4. Particularly, we show the prediction on the same slice with the different methods and for increasing n. We first observe that predictions of the *FS* model are very unstable, not clearly improving as more labeled images are included in the training, which aligns with the results found in Fig. 5.3. Then, the *Proposals* approach fails to generate visually acceptable segmentations, even with 30 pixel-wise labeled patients. Although its performance improves with the number of labeled patients used in training, its results are not visually satisfying for any value of n. Our curriculum semi-supervised segmentation approach achieves decent results from n=5. It only requires 20 patients to yield comparable segmentations to those of the *Oracle* and the manual ground truth.

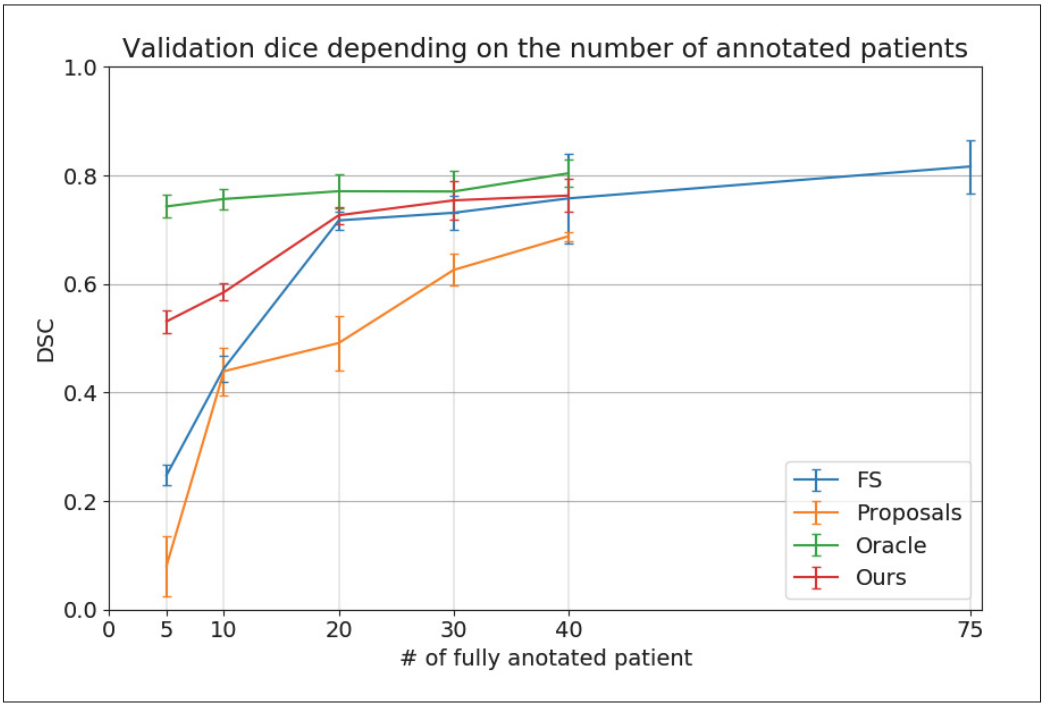


Figure 5.2 Mean DSC per method and for several n annotated patients.

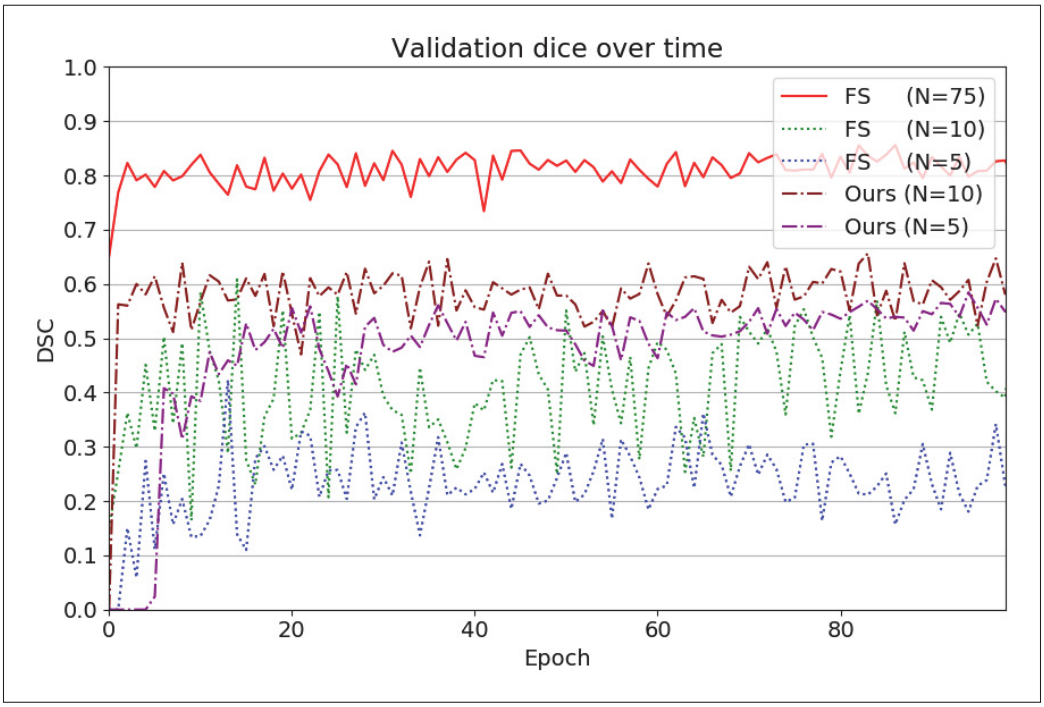


Figure 5.3 Validation DSC over time, with a subset of the evaluated models.

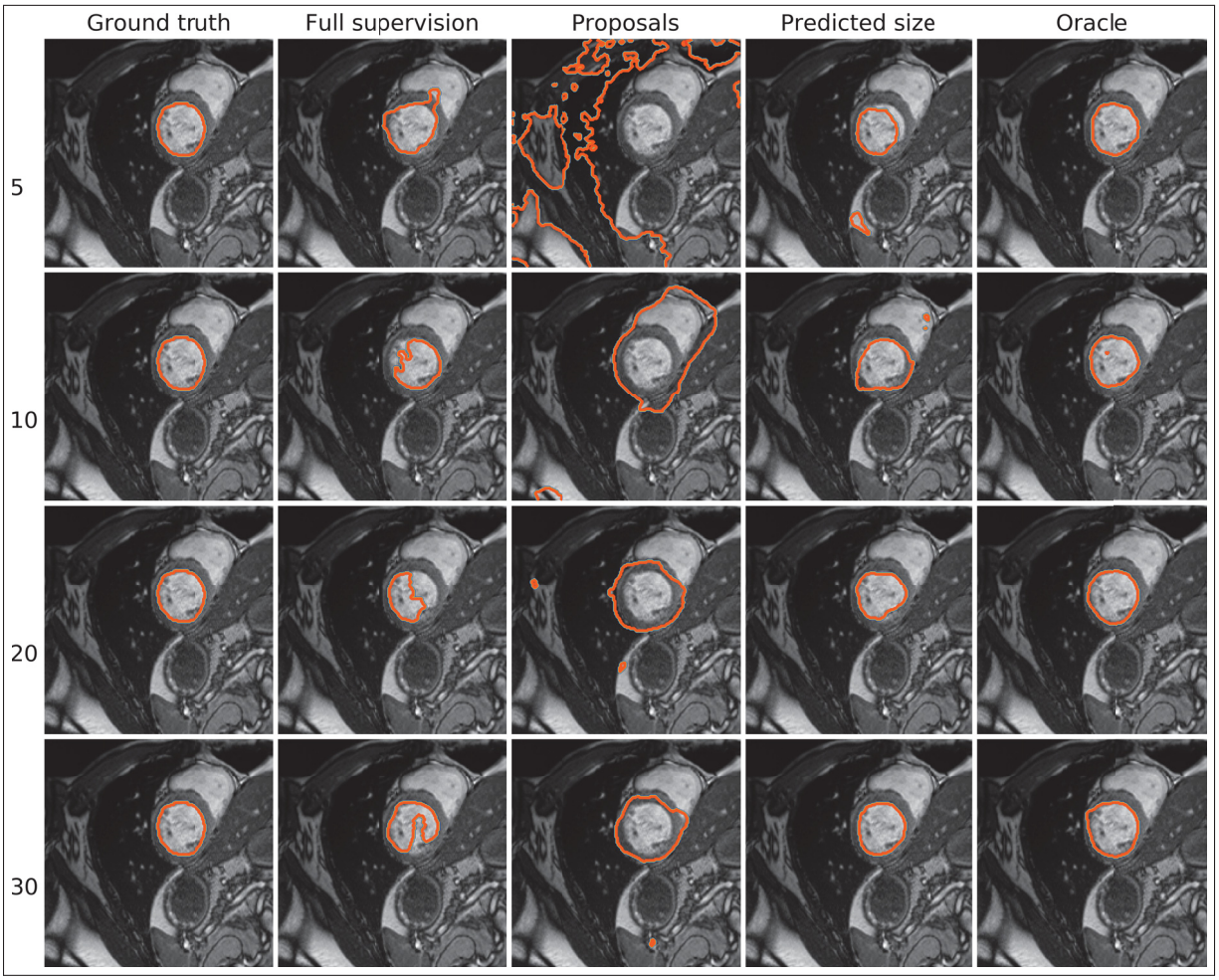


Figure 5.4 Visual comparison for the different methods, with varying number of fully annotated patients used for training. Best viewed in colors

CHAPTER 6

BOUNDING BOXES FOR WEAKLY SUPERVISED SEGMENTATION: GLOBAL CONSTRAINTS GET CLOSE TO FULL SUPERVISION

Hoel Kervadec[a], Jose Dolz[b], Shanshan Wang[c], Eric Granger[a], Ismail Ben Ayed[a]

[a] Département de génie des systèmes, ÉTS Montréal, QC, Canada,
[b] Département de génie logiciel et des TI, ÉTS Montréal, QC, Canada,
[c] Shenzhen Institutes of Advanced Technology, China.

Oral presentation at Medical Imaging with Deep Learning (MIDL) 2020, Montréal.

Abstract

We propose a novel weakly supervised learning segmentation based on several global constraints derived from box annotations. Particularly, we leverage a classical tightness prior to a deep learning setting via imposing a set of constraints on the network outputs. Such a powerful topological prior prevents solutions from excessive shrinking by enforcing any horizontal or vertical line within the bounding box to contain, at least, one pixel of the foreground region. Furthermore, we integrate our deep tightness prior with a global background emptiness constraint, guiding training with information outside the bounding box. We demonstrate experimentally that such a global constraint is much more powerful than standard cross-entropy for the background class. Our optimization problem is challenging as it takes the form of a large set of inequality constraints on the outputs of deep networks. We solve it with sequence of unconstrained losses based on a recent powerful extension of the log-barrier method, which is well-known in the context of interior-point methods. This accommodates standard stochastic gradient descent (SGD) for training deep networks, while avoiding computationally expensive and unstable Lagrangian dual steps and projections. Extensive experiments over two different public data sets and applications (prostate and brain lesions) demonstrate that the synergy between our global tightness and emptiness priors yield very competitive performances, approaching full supervision and outperforming significantly DeepCut. Furthermore, our approach removes the need for computationally expensive proposal generation. Our code is publicly available.

6.1 Introduction

Semantic segmentation is of paramount importance in the understanding and interpretation of medical images, as it plays a crucial role in the diagnostic, treatment and follow-up of many diseases. Even though the problem has been widely studied during the last decades, we have witnessed a tremendous progress in the recent years with the advent of deep convolutional neural networks (CNNs) Dolz *et al.* (2018); Litjens *et al.* (2017); Rajchl *et al.* (2017); Ronneberger *et al.* (2015). Nevertheless, a main limitation of these models is the need of large annotated datasets, which hampers the performance and limits the scalability of deep CNNs in the medical domain, where pixel-wise annotations are prohibitively time-consuming. Weakly supervised learning has gained popularity to alleviate the need of large amounts of pixel-labeled images. Weak labels can come in the form of image tags Pathak *et al.* (2015a), scribbles Lin *et al.* (2016), points Bearman *et al.* (2016), bounding boxes Dai *et al.* (2015); Hsu *et al.* (2019); Khoreva *et al.* (2017) or global constraints Jia *et al.* (2017); Kervadec *et al.* (2019b). A common paradigm in the weakly supervised learning setting is to employ weak annotations to generate *pseudo-masks* or *proposals*. These proposals are ‘’fake" labels, which are generated iteratively to refine the parameters of deep CNNs, thereby mimicking full supervision. Unfortunately, as discussed in several recent works Kervadec *et al.* (2019b); Tang *et al.* (2018b), proposals contain errors, which might be propagated during training, affecting severely segmentation performances. Furthermore, iterative proposal generation increases significantly the computation load for training. More recently, several studies investigated global loss functions, e.g., in the form of constraints on the target-region size Bateson, Kervadec, Dolz, Lombaert & Ayed (2019); Jia *et al.* (2017); Kervadec *et al.* (2019b); Pathak *et al.* (2015a). This can be done by constraining the softmax outputs of deep networks, leveraging unlabeled data with a single loss function and removing the need for iterative proposal generation. Nevertheless, despite the good performances achieved by these works in certain practical scenarios, their applicability might be limited by the assumptions underlying such global constraints, e.g., precise knowledge of the target region size.

Among different weak supervision approaches, bounding box annotations are an appealing alternative due to their simplicity and low-annotation cost. In practice, bounding boxes can be defined with two corner coordinates, allowing fast placement and light storage. Furthermore, they provide localization-awareness, which spatially constrains the problem. This form of supervision has indeed become popular in computer vision to initialize shallow segmentation models, whose outputs are later used to train deep networks, as in full supervision Dai *et al.* (2015); Khoreva *et al.* (2017); Papandreou *et al.* (2015); Pu, Huang, Guan & Zou (2018). A naive use of bounding boxes amounts to generating pseudo-labels by simply considering each pixel within the bounding box as a positive sample for the respective class Papandreou *et al.* (2015); Rajchl *et al.* (2017). However, in a realistic scenario, a bounding box also contains background pixels. To account for this, some advanced foreground extraction methods are employed. Particularly, the very popular GrabCut Rother *et al.* (2004) is a standard choice to generate segmentation masks from bounding boxes, even though alternative approaches such as Multiscale Combinatorial Grouping (MCG) Pont-Tuset, Arbelaez, Barron, Marques & Malik (2017) were recently used for the same purpose Dai *et al.* (2015).

6.1.1 Contributions

We propose a novel weakly supervised learning paradigm based on several global constraints derived from box annotations. First, we leverage the classical tightness prior in Lempitsky *et al.* (2009) to a deep learning setting, and re-formulate the problem by imposing a set of constraints on the network outputs. Such a powerful topological prior prevents solutions from excessive shrinking by enforcing any horizontal or vertical line within the bounding box to contain, at least, one pixel of the foreground region. Furthermore, we integrate our deep tightness prior with a global background emptiness constraint, guiding training with information outside the bounding box. As we will see in our experiments, such a global constraint is much more powerful than standard cross-entropy for the background class. Our optimization problem is challenging as it takes the form of a large set of inequality constraints, which are difficult to handle in the context of deep networks. We solve it with sequence of unconstrained losses based

on a recent powerful extension of the log-barrier method Kervadec, Dolz, Yuan, Desrosiers, Granger & Ben Ayed (2020), which is well-known in the context of interior-point methods. This accommodates standard stochastic gradient descent (SGD) for training deep networks, while avoiding computationally expensive and unstable Lagrangian dual steps and projections. Extensive experiments over two different public data sets and applications (prostate and brain lesions) demonstrate that the synergy between our global tightness and emptiness priors yield very competitive performances, approaching full supervision and outperforming significantly DeepCut Rajchl *et al.* (2017). Furthermore, our approach removes the need for computationally expensive proposal generation.

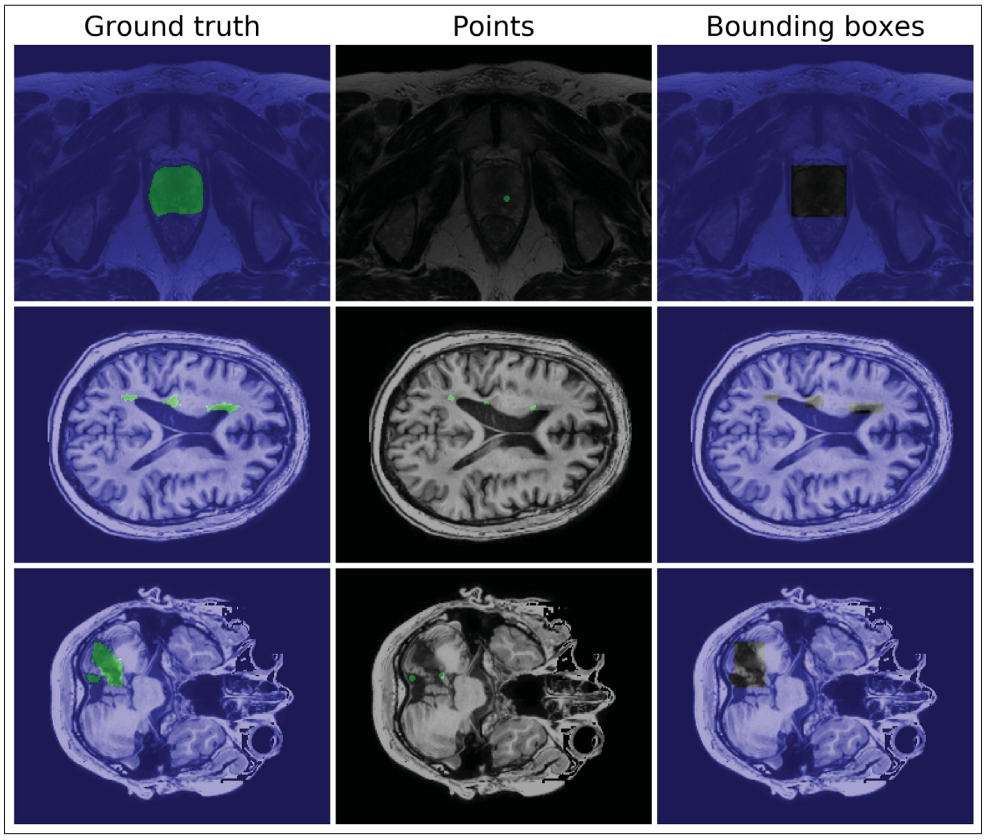


Figure 6.1 Example of weak labels on two different tasks: prostate segmentation and stroke lesion segmentation.

6.2 Related works

6.2.1 Weakly supervised medical image segmentation

Despite the increasing interest in weakly supervised segmentation models in the computer vision community, the literature on these models in medical imaging remains scarce. The authors of Qu *et al.* (2019) leverage point annotations in the context of histopathology images. From labeled points, they derived additional information in the form of a voronoi diagram, so as to generate coarse labels for nuclei segmentation. Their objective function integrated the cross-entropy with coarse labels and the conditional random field (CRF) loss in Tang *et al.* (2018b). Similarly to previous works in computer vision, Nguyen, Pica, Rosa, Hrbacek, Weber, Schalenbourg, Sznitman & Cuadra (2019) used classification activation maps (CAMs) derived from the networks as a pseudo-masks to train a CNN in a fully supervised manner. To constrain the location of the target, they employed an Active Shape Model (ASM) as a prior information. Nevertheless, this method presents two limitations. First, as in similar works, inaccuracies of the pseudo-masks may lead to sub-optimal performances. Second, the ASM is tailored to this specific application, as its generation for novel classes is dependent on the segmentation masks. More recently, Wu, Du, Luo, Wen, Shen & Feng (2019) proposed to refine the generated CAM with attention, with the goal of generating more reliable pseudo-masks. Alternatively, other recent methods investigated how to constrain network predictions with global statistics, for instance, the size of the target region Bateson *et al.* (2019); Jia *et al.* (2017); Kervadec, Dolz, Granger & Ben Ayed (2019a); Kervadec *et al.* (2019b). This type of prior information can be imposed as equality Jia *et al.* (2017) or inequality Bateson *et al.* (2019); Kervadec *et al.* (2019b) constraint. Although such constrained-CNN predictions achieved outstanding performances in a few weakly-supervised learning scenarios, their applicability remains limited to certain assumptions.

6.2.2 Bounding box supervision

Most CNN-based methods under the umbrella of bounding-box supervision fall under the category of proposal-based methods. In these approaches, the bounding box annotations are exploited to obtain initial pseudo-masks, or proposals, typically with a shallow segmentation method, e.g., the very popular GrabCut method Rother *et al.* (2004). Then, training typically follows an iterative scheme, which involves two steps, one updating the network parameters and the other adjusting the pseudo-labels Dai *et al.* (2015); Khoreva *et al.* (2017); Papandreou *et al.* (2015). To further refine the pseudo-labels generated at each iteration, several works Rajchl *et al.* (2017); Song, Huang, Ouyang & Wang (2019) used the popular DenseCRF Krähenbühl & Koltun (2011b) or other heuristics. While this might be very effective on some datasets, DenseCRF typically assumes that all the training images have consistent and strong contrast between the foreground and background regions. Finding the optimal DenseCRF parameters[1] is difficult when the contrast of the object edge varies significantly within the same dataset. Moreover, the ensuing training is not end-to-end, as it still relies on a DenseCRF post-processing, even at inference time. Another drawback of those bounding-box based learning approaches—which is also shared by other proposal-based methods in general—is that early mistakes will re-enforce themselves during training. For example, in DeepCut Rajchl *et al.* (2017), while the pseudo-labels cannot grow beyond the bounding box, the inner foreground may gradually disappear. More recently, Hsu et al Hsu *et al.* (2019) employed a Multiple Instance Learning (MIL) framework to impose a tightness prior in the context of instance segmentation of natural images. Focusing on instance segmentation, the method used bounding boxes generated by R-CNN. In such MIL framework, positive bags are composed of box lines while negative bags correspond to lines outside the box. The MIL loss function is defined so as to push the maximum predicted probability within each positive bag to 1, and the maximum predicted probability within each negative bag to 0. This MIL loss is integrated with a GridCRF loss Marin, Tang, I. & Y. (2019) to ensure consistency between neighboring pixels. As many other works, the final predictions are refined with DenseCRF Krähenbühl & Koltun (2011b).

[1]Several hyper-parameters controls the edge sensitivity of popular DenseCRF Krähenbühl & Koltun (2011b), mostly θ_β and θ_γ, but also ω_1, ω_2 and θ_α to some extent.

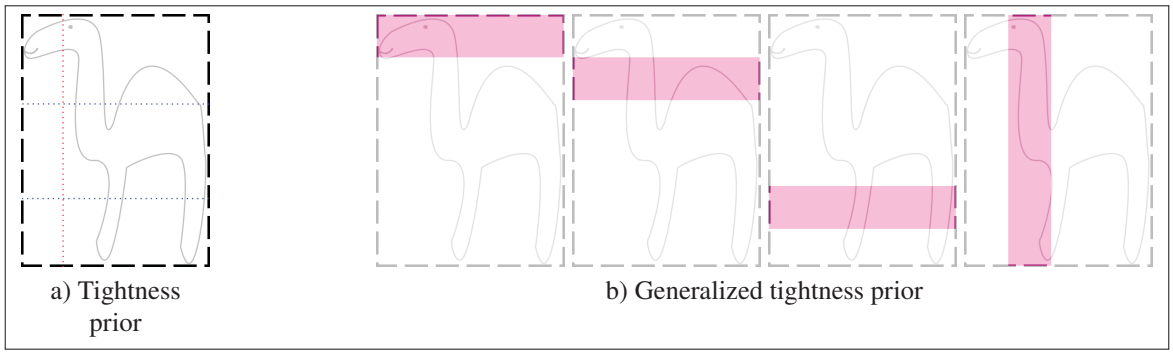


Figure 6.2 (a) Illustration of the tightness prior: any vertical (red) or horizontal (blue) line will cross at least one (1) pixel of the camel. (b) This can be generalized, where segments of width w cross at least w pixels of camel.

6.3 Method

6.3.1 Preliminary notations

Let $X : \Omega \subset \mathbb{R}^{2,3} \rightarrow \mathbb{R}$ denotes a training image, and Ω its corresponding spatial domain. In a standard fully supervised setting, we can denote the training set as $\mathcal{D} = \{(X, Y)\}^D$, where $X \in \mathbb{R}^{\Omega}$ are input images and $Y \in \{0, 1\}^{\Omega}$ their corresponding pixel-wise labels. In the context of this work, however, labels Y take the form of bounding boxes (as shown in Figure 6.1, third column). Thus, we use Ω_O and Ω_I to define the area outside and inside the bounding box, respectively, with $\Omega_O \cup \Omega_I = \Omega$. Let $s_\theta \in [0, 1]^{\Omega}$ denote the probabilities predicted by the CNNs, where 0 and 1 represent background and foreground, respectively. In fully supervised setting, one would typically optimize the standard cross-entropy loss:

$$\min_{\theta} \quad \mathcal{L}_{\text{CE}}(\theta) := -\sum_{p \in \Omega} \left[y_p \log(s_\theta(p)) + (1 - y_p) \log(1 - s_\theta(p)) \right] .$$

6.3.2 Dealing with box annotations

Certainty outside the box

As shown in Figure 6.1, we certainly know that all pixels p outside a given bounding box (Ω_O) belong to the background. A straightforward solution would be to employ the cross-entropy, but only partially for each of those pixels outside the bounding box:

$$\mathcal{L}_{\text{MCE}} := -\sum_{p \in \Omega_O} \log(1 - s_{\boldsymbol{\theta}}(p)).$$

Alternatively, notice that the size of the predicted foreground[2], when computed over the background pixels (Ω_O), should be equal to zero. This gives the following global constraint for our optimization problem, which enforces that the background region is empty:

$$\sum_{p \in \Omega_O} s_{\boldsymbol{\theta}}(p) \leq 0. \tag{6.1}$$

We will refer to this constraint as the *emptiness constraint*, $\mathcal{L}_{EMP}$. $\mathcal{L}_O$ will denote either $\mathcal{L}_{\text{MCE}}$ or $\mathcal{L}_{\text{EMP}}$.

Uncertainty inside the box

While bounding box annotations provide cues about the spatial location of the target regions, pixel-wise information still remain uncertain. However, the bounding box can be further exploited to impose a powerful topological prior, referred to as *tightness prior* Lempitsky *et al.* (2009). This global prior assumes that the target region should be sufficiently close to each of the sides of the bounding box. Therefore, we can expect that each horizontal or vertical line will cross at least one pixel of the target region (as illustrated in Figure 6.2), and for any region shape. Furthermore, we can regroup the lines into segments of width w, each containing w

[2]Here we refer the size as the sum of the softmax probabilities, as it is easy to compute and differentiable. Therefore, it accommodates standard Stochastic Gradient Descent.

lines. In this case, we can assume that at least w pixels of the object will be crossed by the segment. Formally, we can write this as a set of inequality constraints:

$$\sum_{p \in s_l} y_p \geq w \qquad \forall s_l \in \mathcal{S}_L \tag{6.2}$$

where $\mathcal{S}_L := \{s_l\}$ is the set of segments parallel to the sides of the bounding boxes. This can be easily translated into inequality constraints on the outputs of the CNN, where the sum of the softmax probabilities for each segment should be greater or equal to its width. The set of segments $\mathcal{S}_L$ can be efficiently pre-computed; only the masked softmax sum is required during training.

6.3.3 Additional regularization: constraining the global size

The first two parts of the loss are biased toward opposed, trivial solutions: $\mathcal{L}_O$ trivial solution is to predict the whole image as background, while the easiest way to satisfy the tightness constraints (6.2) is to predict everything as foreground. But there is more information that we can exploit from the boxes: their total size gives an upper bound on the object size. We can also assume that a small fraction ϵ of the box belongs to the target region, which yield another lower bound. This takes the form of region-size constraint similar to Kervadec *et al.* (2019b):

$$\begin{aligned} \min_{\boldsymbol{\theta}} \quad & \mathcal{L}_1(\boldsymbol{\theta}) + \ldots + \mathcal{L}_n(\boldsymbol{\theta}) \\ \text{s.t.} \quad & \epsilon|\Omega_I| \leq \sum_{p \in \Omega} s_{\boldsymbol{\theta}}(p) \leq |\Omega_I|. \end{aligned} \tag{6.3}$$

6.3.4 Lagrangian optimization with log-barrier extensions

Optimizing $\mathcal{L}_O$ with the constraints from sections 6.3.2 and 6.3.3 gives the following constrained optimization problem:

$$\begin{aligned} \min_{\boldsymbol{\theta}} \quad & \mathcal{L}_O(\boldsymbol{\theta}) && (6.4) \\ \text{s.t.} \quad & \sum_{p \in s_l} s_{\boldsymbol{\theta}}(p) \geq w && \forall s_l \in \mathcal{S}_L \\ & \epsilon |\Omega_I| \leq \sum_{p \in \Omega} s_{\boldsymbol{\theta}}(p) \leq |\Omega_I|. \end{aligned}$$

This formulation involves a large number of competing constraints. Recent optimization works on constrained CNNs Kervadec *et al.* (2020) suggest that, in the case of multiple competing constraints, log-barrier extensions provide approximations of Lagrangian optimization in the form of sequences of unconstrained losses, which removes completely expensive and unstable primal-dual steps in the context of deep networks, handling the multiple constraints fully within SGD. Therefore, log-barriers can accommodate the interplay between multiple competing constraints, unlike naive penalty-based methods. These desirable properties are consistent with well-established interior-point and log-barrier methods in convex optimization Boyd & Vandenberghe (2004).

For an inequality constraint in the form of $z \leq 0$, the log-barrier extension can be defined as follows:

$$\widetilde{\psi}_t(z) = \begin{cases} -\frac{1}{t} \log(-z) & \text{if } z \leq -\frac{1}{t^2} \\ tz - \frac{1}{t} \log(\frac{1}{t^2}) + \frac{1}{t} & \text{otherwise,} \end{cases} \tag{6.5}$$

where t is a parameter that *raise* the barrier over time (i.e., during training). The main difference with a penalty (such as $\max(0, z)^2$, used by Kervadec *et al.* (2019b)) is that (6.5) acts as a *barrier* even when the constraint is satisfied ($z \leq 0$), with a gradient getting more aggressive when approaching constraint-violation boundary. This makes the training more stable, and prevents already satisfied constraints from being violated during the next training epochs. Using a

penalty could oscillate, alternating between zero and a high-penalty values Kervadec *et al.* (2020).

6.3.5 Final model

Using the log-barrier extension, we obtain the final unconstrained optimization problem, which can be optimized with standard SGD:

$$\min_{\boldsymbol{\theta}} \mathcal{L}_O(\boldsymbol{\theta}) + \lambda \left[\sum_{s_l \in \mathcal{S}_L} \widetilde{\psi}_t \left(w - \sum_{p \in s_l} s_{\boldsymbol{\theta}}(p) \right) \right] + \widetilde{\psi}_t \left(\epsilon |\Omega_I| - \sum_{p \in \Omega} s_{\boldsymbol{\theta}}(p) \right) + \widetilde{\psi}_t \left(\sum_{p \in \Omega} s_{\boldsymbol{\theta}}(p) - |\Omega_I| \right). \quad (6.6)$$

λ is a real number balancing the tightness prior with respect to the other parts of the loss. Notice that all log-barrier extensions $\widetilde{\psi}_t$ use the same t, with a common scheduling strategy for t. This limits the number of hyper-parameters and simplifies the model.

6.4 Experiments

6.4.1 Datasets and evaluation

We evaluate our method on two different tasks: prostate segmentation in MR-T2 and brain lesion segmentation in MR-T1. Among these tasks, lesion segmentation is particularly challenging, due to the heterogeneity of the lesions and high imbalance in the number of foreground and background pixels.

Prostate segmentation on MR-T2

The first dataset that we use was made available at the MICCAI 2012 prostate MR segmentation challenge[3] Litjens *et al.* (2014). It contains the transversal T2-weighted MR images of

[3]https://promise12.grand-challenge.org

50 patients acquired at different centers, with multiple MRI vendors and different scanning protocols. The images include patients with benign diseases, as well as with prostate cancer. Images resolution ranges from $15 \times 256 \times 256$ to $54 \times 512 \times 512$ voxels, with a spacing ranging from $2 \times 0.27 \times 0.27$ to $4 \times 0.75 \times 0.75\text{mm}^3$. We employed 40 patients for training and 10 for validation.

Brain lesion segmentation on MR-T1

We also evaluated the proposed method on the Anatomical Tracings of Lesions After Stroke (ATLAS) Liew, Anglin, Banks, Sondag, Ito, Kim, Chan, Ito, Jung, Khoshab et al. (2018), an open-source dataset of stroke lesions. It contains 229 T1-weighted MR images, coming from different cohorts and different scanners. All the images have a resolution of $197 \times 233 \times 189$ pixels, with a spacing of $1 \times 1 \times 1$ mm. The annotations were done by a team of 11 experts, who received a standardized training. We retained 26 images for validation, while the rest were used for training.

Evaluation

To compare quantitatively the performances of the different methods, we employed the Dice similarity coefficient, a standard performance metric in medical image segmentation. In addition to the baseline models, we also perform comprehensive comparisons with DeepCut Rajchl *et al.* (2017), whose learning setting is also based on bounding box annotations.

6.4.2 Implementation details

To evaluate our method under different settings, we experimented with a differnt network architecture for each task. We employ a residual version of the well-known UNet Ronneberger *et al.* (2015) to segment the prostate, whereas ENet Paszke *et al.* (2016) was a backbone architecture in the stroke lesion segmentation experiments. The models were trained with ADAM Kingma & Ba (2015), an initial learning rate of 5×10^{-4} and a batch size of 4 for the

prostate and 32 for stroke lesions. While we employed offline data augmentation (i.e., mirroring, flipping, rotation) to augment the PROMISE12 dataset, no augmentation was performed on the ATLAS dataset. The reason for this is the low number of images on the PROMISE12 dataset compared to ATLAS.

The log-barrier parameters were set following Kervadec *et al.* (2020), and were shared across all the log-barrier instances. We set λ (from Eq. (6.6)) as 0.0001 for both datasets. The DenseCRF hyper-parameters are the same as in Rajchl *et al.* (2017), and the proposals are updated every 10 epochs for PROMISE12, and every 5 epochs for ATLAS. We empirically found that changes on the width w of the segments for the tightness constraints did not have a significant impact on the results. Therefore, w was set to 5 in all the experiments.

All methods are implemented in PyTorch, with the exception of the DenseCRF Krähenbühl & Koltun (2011b) which uses the Python wrapper PyDenseCRF [4]. To speed the proposal generation of DeepCut, the CRF inference is parallelized using the standard Python multiprocessing module, with a careful use of SharedArrays to avoid un-necessary and costly copies of arrays between the processes. The code is available online[5].

6.4.3 Sensitivity study on box-annotation precision

While the main experiments are performed on tight boxes (i.e., the gap between the target regions and the bounding-box sides is not significant), we perform additional experiments where a margin m of 10 pixels was added on each side. This enables us to evaluate the robustness of each model to imprecise bounding-box placement. Robustness to placement is of significant importance, since perfect annotation of all bounding boxes might be unrealistic. Furthermore, robustness to imprecision also alleviates the problem of annotator subjectivity.

[4] https://github.com/lucasb-eyer/pydensecrf
[5] https://github.com/LIVIAETS/boxes_tightness_prior

6.5 Results

6.5.1 Main experiment

The results of the segmentation experiments are reported in Table 6.1. We can observe that the proposed method consistently outperforms DeepCut Rajchl *et al.* (2017) across the two datasets. The differences in performance range from 1% in the PROMISE12 dataset to 10% in the case of ATLAS. Furthermore, the results obtained from the two loss functions designed to deal with the background constraints indicate that the proposed global emptiness constraint is more effective in our setting. We hypothesize this is due to several factors. First, employing the emptiness constraint on background pixels results in all the constraint losses being on the same scale, which has very nice properties from an optimization perspective. Second, the imbalance nature of the segmentation task in the ATLAS dataset makes the use of the cross-entropy over all the background pixels a suboptimal alternative, forcing solutions that encourage empty segmentations. Finally, we can observe that the proposed method achieves performances comparable to full supervision, particularly in the task of stroke lesion segmentation. Using only a subset of the losses does not give optimal results, showing their synergy.

Table 6.1 Results on the validation set for the proposed method, and the different baselines in both PROMISE12 and ATLAS datasets. The best results in the weakly supervised setting are highlighted in bold. NA means that the network didn't learn to segment anything meaningful.

Method	PROMISE12	ATLAS
	DSC	DSC
Deep cut Rajchl *et al.* (2017)	0.827 (0.085)	0.375 (0.246)
Tightness prior		
w/ emptiness constraint	NA	0.161 (0.145)
Tightness prior + box size	0.620 (0.100)	0.146 (0.134)
w/ masked cross-entropy ($\mathcal{L}_{\text{MCE}}$)	0.774 (0.045)	0.159 (0.203)
w/ emptiness constraint ($\mathcal{L}_{\text{EMP}}$)	**0.835 (0.032)**	**0.474 (0.245)**
Full supervision (Cross-entropy)	0.901 (0.025)	0.489 (0.294)

Figure 6.3 depicts the validation results over training of the different models. Even though DeepCut achieves similar results as the proposed approach in the PROMISE12 dataset, we can see that it is very unstable during training, as is the case generally for proposal-based methods. Additionally, its performance degrades over time. This effect is even more noticeable on the ATLAS dataset, where it collapses to empty segmentations after 25 epochs. This behaviour is a clear example of the instability of proposal-based methods, since we observed similar findings on the training images. More details about this issue are provided in Appendix 1.

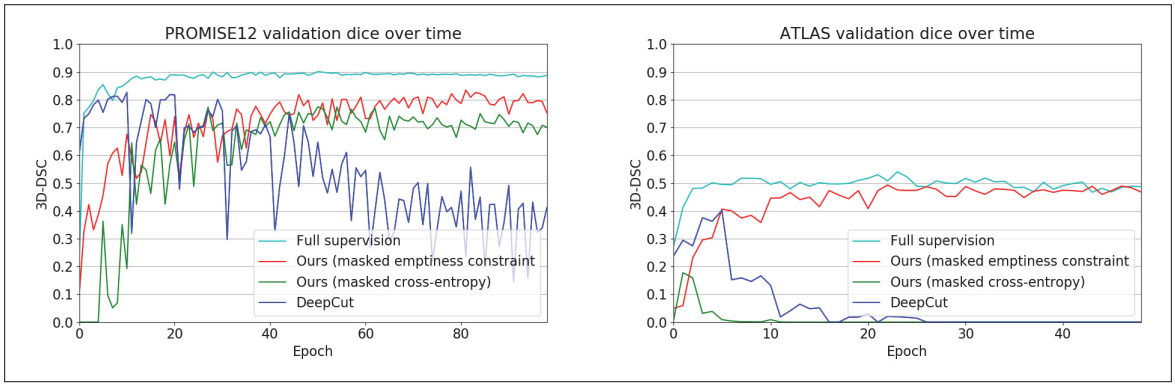


Figure 6.3 Evolution the validation DSC values over time for both PROMISE12 and ATLAS, and for different methods.

Qualitative segmentation results are depicted in Fig 6.4. We can observe how the proposed method with masked CE achieves satisfactory visual results on the prostate (first two rows), but fails to properly segment stroke lesions (last two rows). In contrast, when background segmentations are optimized with the proposed emptiness constraint, we observe how the segmentation results approach full supervision performance in both datasets. This is in line with the results reported in Table 6.1. On the other hand, DeepCut succeeds to segment the prostate but it is not able to obtain satisfactory segmentations for brain lesions. Looking closer at these segmentations, we can observe that they do not reliably follow the target boundaries. This can be explained by the fact that denseCRF assumes strong contrasts between foreground and background regions, which is not the case in many of these images. Furthermore, the results provided by denseCRF are sensitive to its hyper-parameters θ_β, θ_γ, ω_1 and ω_2, which control the edge sensitivity. Since the set of hyper-parameters were fixed across all the images

in the whole dataset, it might happen that an optimal set of hyper-parameters for a given image performs sub-optimally for another image.

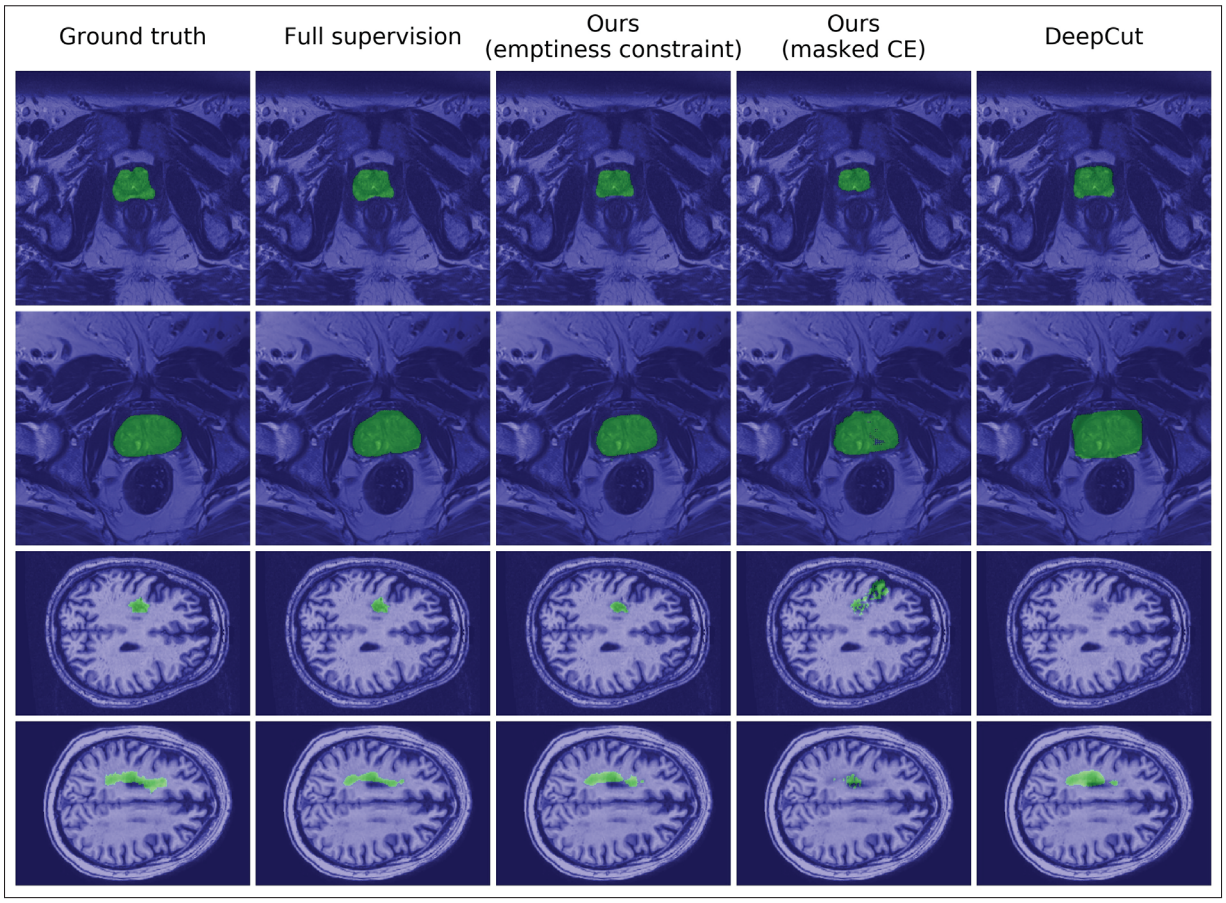


Figure 6.4 Predicted segmentation on the validation set for the two tasks.

Table 6.2 Sensitivity study wrt. the box margins on the PROMISE12 dataset. Best results highlighted in bold.

Method	Margin=0	Margin=10
DeepCut	0.827 (0.085)	0.684 (0.069)
Ours (emptiness constraint)	**0.835 (0.032)**	**0.778 (0.047)**

6.5.2 Resilience to box imprecision

Results of the sensitivity study on the box precision are reported in Table 6.2. While all methods were able to reach similar performances when the bounding box annotation is nearly perfect (despite stability issues for some methods), their performance degrades as the margin between the region of interest and the borders of the bounding box increases. Specifically, if a margin m of 10 pixels is added on each side, the performance of the proposed method only drops by 5%, in terms of DSC, whereas DeepCut performance decreases by 14%.

Finally, the computational cost of the different methods is discussed in more details in Appendix 2.

6.6 Conclusion

In this paper we proposed a novel weakly-supervised learning paradigm based on several global constraints, which are derived from bounding box annotations. First, the classical tightness prior is integrated into a a deep learning framework by reformulating the problem as a set of constraints on the outputs of the network. Second, a global background emptiness constraint is employed to enforce empty segmentations outside the bounding box, which is demonstrated to be more powerful than standard cross-entropy for handling the background class. Integration of such a large set of inequality constraints on deep networks represents a challenging optimization problem.

We solve it with sequence of unconstrained losses, which are based on a recent extension of the log-barrier method. Since this formulation accommodates standard stochastic gradient descent, it can be easily trained on deep networks. We performed comprehensive experiments on two public benchmarks for the challenging tasks of prostate and brain stroke lesion segmentation, and demonstrated that the proposed approach outperforms state-of-the-art approaches with bounding-box supervision. Furthermore, quantitative and qualitative results indicate that the proposed approach has the potential to close the gap between bounding-box annotations and full supervision in semantic-segmentation tasks.

The sensibility study showed that the proposed method is resilient to imprecision in the box tightness. Future works will investigate the use of 3D bounding boxes as annotations, which will make the corresponding 2D boxes looser. Such a workflow could further speed up the annotation process. The proposed framework could also be extended to 3D-CNN, by generating segments for the tightness prior along the three axes. Furthermore, our approach is also compatible with multi-class segmentation problems, even when bounding boxes of different classes overlap.

CONCLUSION

In this dissertation , we addressed the problem of enforcing global inequality constraints during the training of deep convolutional neural networks. By using either simple quadratic penalties or more principled log-barrier extensions, we were able to bypass expensive and difficult primal-dual steps in standard Lagrangian optimization. Through our papers, we demonstrated that the log-barrier extension is the best method for complex settings, and performs significantly better than other methods in the literature. These algorithmic developments enabled us to test various priors and constraints, improving performances in semi- and weakly-supervised segmentation.

Our work has limitations and open problems, especially when dealing with global constraints over 3D (or larger) image domains. For constraints that involve non-linear functions of summations over the input domains, our framework may become intractable for mini-batch training of deep networks; it requires storing of and performing summations over all the pointwise gradients within the input-image domain. This is often not feasible with current hardware for very big inputs. Methods for such images often sub-patch the image, processing only a part of it at a time. This cannot accommodate all the global constraints that we used, such as size or centroid (Chapter 3). Managing to enforce global constraints on very large inputs for the processing hardware is still an open problem for future research.

Some of the most interesting constraints remain to investigate, like spatial relationships Deng, Todorovic & Jan Latecki (2015), very relevant to multi-organ segmentation, where prior information about the organs position is text-book knowledge. We did not investigate multi-class settings, which could bring interesting constraints (For example, the myocardium encompass the left-ventricle cavity). Our work could also trigger future investigations beyond image segmentation. Indeed, the powerful and general log-barrier extensions from Chapter 3 can be used in other domains Nandwani *et al.* (2019), constraining either the network output, or regularizing the inner layers of a network.

Chapter 4 has already proved to be a useful work for the community, with many positive feedbacks and reports of improved performances on various tasks. Yet, this boundary-loss

work focused only on the binary setting and Euclidean distance. The multi-class setting would be very interesting to investigate, as it naturally removes the trivial solutions that exist in the binary case (i.e., empty foreground predictions): A trivial solution for one class would be a non-suitable solution for another class (and vice-versa), mitigating each other errors naturally. This is illustrated in Figure 6.5, where preliminary results show the boundary loss alone can learn to segment a 4-class setting. Other distance functions could be a way to use image content and edge information, potentially enabling its use for weakly annotated images.

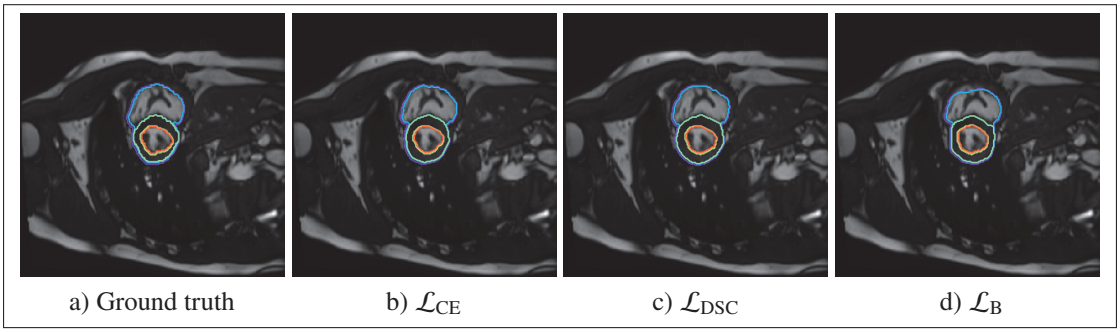


Figure 6.5 Results on the ACDC, a 4-classes dataset, when training with different losses. Unlike in the binary case, here the boundary loss is able to learn to segment the object properly.

APPENDIX I

ADDITIONAL MATERIALS FOR CHAPTER 3

1. Proof of Proposition 2

In this section, we provide a detailed proof for the duality-gap bound in Prop. 2. Recall our unconstrained approximation for inequality-constrained CNNs:

$$\min_{\boldsymbol{\theta}} \mathcal{E}(\boldsymbol{\theta}) + \sum_{i=1}^{P} \sum_{n=1}^{N} \tilde{\psi}_t \left(f_i(s^n_{\boldsymbol{\theta}}) \right) \tag{A I-1}$$

where $\tilde{\psi}_t$ is our log-barrier extension, with t strictly positive. Let $\boldsymbol{\theta}^*$ be the solution of problem (A I-1) and $\boldsymbol{\lambda}^* \in \mathbb{R}^{P \times N}$ the corresponding vector of implicit dual variables given by:

$$\lambda^*_{i,n} = \begin{cases} -\frac{1}{t f_i(s^n_{\boldsymbol{\theta}^*})} & \text{if } f_i(s^n_{\boldsymbol{\theta}^*}) \leq -\frac{1}{t^2} \\ t & \text{otherwise} \end{cases} \tag{A I-2}$$

We assume that $\boldsymbol{\theta}^*$ verifies approximately[1] the optimality condition for a minimum of (A I-1):

$$\nabla \mathcal{E}(\boldsymbol{\theta}^*) + \sum_{i=1}^{P} \sum_{n=1}^{N} \tilde{\psi}'_t \left(f_i(s^n_{\boldsymbol{\theta}^*}) \right) \nabla f_i(s^n_{\boldsymbol{\theta}^*}) \approx 0 \tag{A I-3}$$

It is easy to verify that each dual variable $\lambda^*_{i,n}$ corresponds to the derivative of the log-barrier extension at $f_i(S_{\boldsymbol{\theta}^*})$:

$$\lambda^*_{i,n} = \tilde{\psi}'_t \left(f_i(s^n_{\boldsymbol{\theta}^*}) \right)$$

Therefore, Eq. (A I-3) means that $\boldsymbol{\theta}^*$ verifies approximately the optimality condition for the Lagrangian corresponding to the original inequality-constrained problem in Eq. (3.1) when $\boldsymbol{\lambda} = \boldsymbol{\lambda}^*$:

$$\nabla \mathcal{E}(\boldsymbol{\theta}^*) + \sum_{i=1}^{P} \sum_{n=1}^{N} \lambda^*_{i,n} \nabla f_i(s^n_{\boldsymbol{\theta}^*}) \approx 0 \tag{A I-4}$$

[1] When optimizing unconstrained loss via stochastic gradient descent (SGD), there is no guarantee that the obtained solution verifies exactly the optimality conditions.

It is also easy to check that the implicit dual variables defined in (A I-2) corresponds to a feasible dual, i.e., $\boldsymbol{\lambda}^* > 0$ element-wise. Therefore, the dual function evaluated at $\boldsymbol{\lambda}^* > 0$ is:

$$g(\boldsymbol{\lambda}^*) = \mathcal{E}(\boldsymbol{\theta}^*) + \sum_{i=1}^{P} \sum_{n=1}^{N} \lambda_{i,n}^* f_i(s_{\boldsymbol{\theta}^*}^n),$$

which yields the duality gap associated with primal-dual pair $(\boldsymbol{\theta}^*, \boldsymbol{\lambda}^*)$:

$$\mathcal{E}(\boldsymbol{\theta}^*) - g(\boldsymbol{\lambda}^*) = -\sum_{i=1}^{P} \sum_{n=1}^{N} \lambda_i^* f_i(s_{\boldsymbol{\theta}^*}^n) \tag{A I-5}$$

Now, to prove that this duality gap is upper-bounded by PN/t, we consider three cases for each term in the sum in (A I-5) and verify that, for all the cases, we have $\lambda_{i,n}^* f_i(s_{\boldsymbol{\theta}^*}^n) \geq -\frac{1}{t}$.

- $f_i(s_{\boldsymbol{\theta}^*}^n) \leq -\frac{1}{t^2}$: In this case, we can verify that $\lambda_{i,n}^* f_i(s_{\boldsymbol{\theta}^*}^n) = -\frac{1}{t}$ using the first line of (A I-2).
- $-\frac{1}{t^2} \leq f_i(s_{\boldsymbol{\theta}^*}^n) \leq 0$: In this case, we have $\lambda_{i,n}^* f_i(s_{\boldsymbol{\theta}^*}^n) = t f_i(s_{\boldsymbol{\theta}^*}^n)$ from the second line of (A I-2). As t is strictly positive and $f_i(s_{\boldsymbol{\theta}^*}^n) \geq -\frac{1}{t^2}$, we have $t f_i(s_{\boldsymbol{\theta}^*}^n) \geq -\frac{1}{t}$, which means $\lambda_{i,n}^* f_i(s_{\boldsymbol{\theta}^*}^n) \geq -\frac{1}{t}$.
- $f_i(s_{\boldsymbol{\theta}^*}^n) \geq 0$: In this case, $\lambda_{i,n}^* f_i(s_{\boldsymbol{\theta}^*}^n) = t f_i(s_{\boldsymbol{\theta}^*}^n) \geq 0 > -\frac{1}{t}$ because t is strictly positive.

In all the three cases, we have $\lambda_{i,n}^* f_i(s_{\boldsymbol{\theta}^*}^n) \geq -\frac{1}{t}$. Summing this inequality over i gives:

$$-\sum_{i=1}^{P} \sum_{n=1}^{N} \lambda_{i,n}^* f_i(s_{\boldsymbol{\theta}^*}^n) \leq \frac{PN}{t}.$$

Using this inequality in (A I-5) yields the following upper bound on the duality gap associated with primal $\boldsymbol{\theta}^*$ and implicit dual feasible $\boldsymbol{\lambda}^*$ for the original inequality-constrained problem:

$$\mathcal{E}(\boldsymbol{\theta}^*) - g(\boldsymbol{\lambda}^*) \leq PN/t$$

□

This bound yields sub-optimality certificates for feasible solutions of our approximation in (A I-1). If the solution $\boldsymbol{\theta}^*$ that we obtain from our unconstrained problem (A I-1) is feasible, i.e., it satisfies constraints $f_i(s^n_{\boldsymbol{\theta}^*}) \leq 0$, $\forall i, \forall n$, then $\boldsymbol{\theta}^*$ is PN/t-suboptimal for the original inequality constrained problem: $\mathcal{E}(\boldsymbol{\theta}^*) - \mathcal{E}^* \leq PN/t$. Our upper-bound result can be viewed as a generalization of the duality-gap equality for the standard log-barrier function Boyd & Vandenberghe (2004). Our result applies to the general context of convex optimization. In deep CNNs, of course, a feasible solution for our approximation may not be unique and is not guaranteed to be a global optimum as $\mathcal{E}$ and the constraints are not convex.

2. Qualitative results on PROMISE12

Examples of the labels used are shown in Figure I-1, and qualitative comparisons between methods are available in Figures I-1.

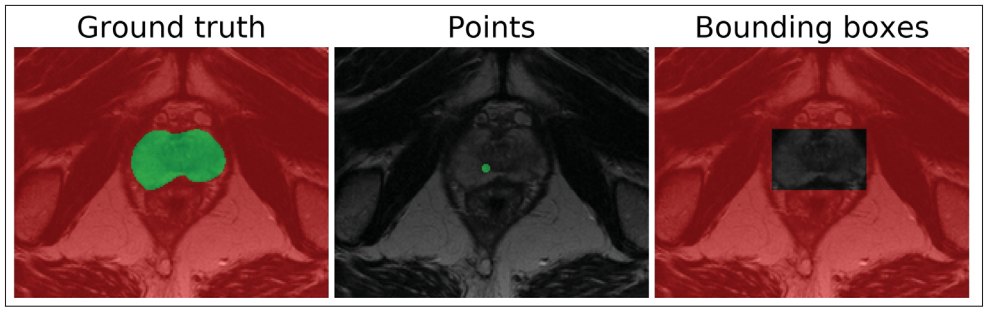


Figure-A I-1 Full mask of the prostate (*left*) and the generated point and box annotations (*middle* and *right*) on PROMISE12. The background is depicted in red and the foreground in green. No color means that no information is provided about the pixel class. The figures are best viewed in colors.

3. Analysis of the dual step for the ReLU Lagrangian

As pointed out by Nandwani *et al.* (2019), when imposing P different constraints on each data point, we end-up with a dual variable $\boldsymbol{\lambda} \in \mathbb{R}^{P \times N}$. The authors of Nandwani *et al.* (2019) mentioned that this could be an issue for scalability. Here, we argue that, from a computational perspective, this is not a very significant issue.

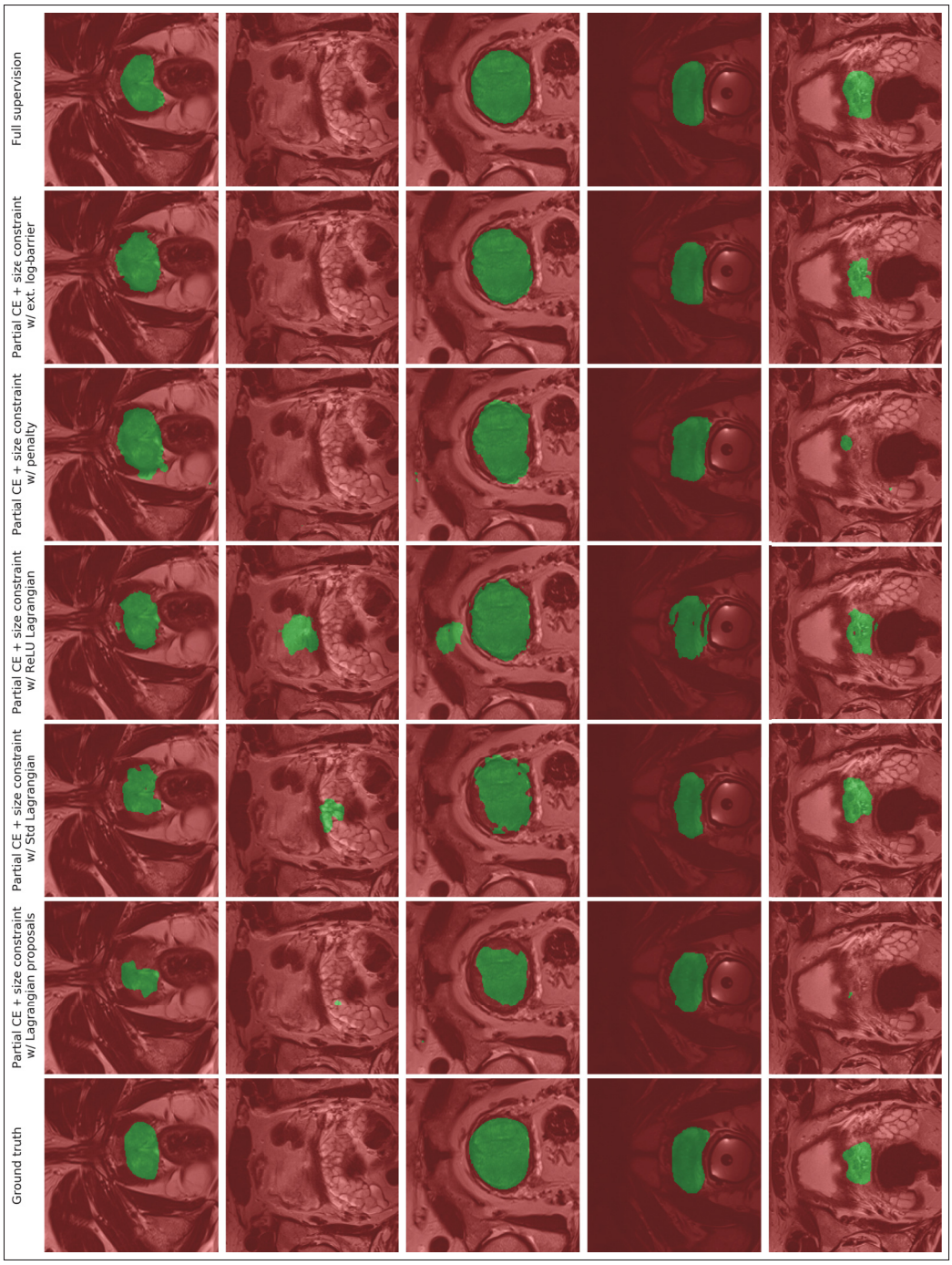


Figure-A I-2 Results on the PROMISE12 dataset. Images are cropped for visualization purposes. The background is depicted in red, and foreground in green. The figures are best viewed in colors.

Assuming 2 different constrained functions per datapoint, each with a lower and upper bound, we have 4 float value to store per data point. If each value is represented as a `float32` (which

is very reasonable, as the extra precision of a `float64` is rarely used in Deep Learning), this yields a total of 128 bits per datapoint, or 16 bytes. This gives `16MB` to store per million datapoint, which is within the reach of modern computers. While it is true that fetching the current $\lambda_{i,n}$ for each n adds some complexity in the code, it only adds a constant and a negligible cost with respect to N.

However, regrouping $\lambda_{i,n}\ \forall n$ into a single λ_i (so that $\boldsymbol{\lambda} \in \mathbb{R}^P$) introduces the following, potentially undesirable property during the $\boldsymbol{\lambda}$ udpates:

$$\nabla\lambda_i = \sum_n \max(0, f_i(s^n_{\boldsymbol{\theta}})) \geq \max_n \max(0, f_i(s^n_{\boldsymbol{\theta}}))$$

$$\Longrightarrow$$

$$\exists m \in \mathcal{D} : f_i(s^m_{\boldsymbol{\theta}}) > 0 \Rightarrow \lambda_i^{t+1} > \lambda_i^t$$

In other words, if a single data point has an unsatisfied constraint, λ_i will keep increasing for the whole dataset. This may make the balancing of competing constraints very difficult, as shown by our experiments (especially on PROMISE12 with Setting I). $\boldsymbol{\lambda}$ kept increasing until reaching very high values, making constraint balancing difficult to reach.

4. Lagrangian with proposals: process and equations

The method of Pathak *et al.* (2015a) optimize Equation (3.5). For a sample n[2]:

$$\mathcal{L}_n(\tilde{y}, \boldsymbol{\lambda}, \boldsymbol{\nu}) = KL(\tilde{y}^n || s^n_{\boldsymbol{\theta}}) + \sum_{i=1}^{P} \lambda_i^n (\tilde{y}^{n\top} a_i - b_i) + \sum_{p \in \Omega} \nu_p^n (\mathbf{1}^\top \tilde{y}^n_p - 1), \tag{A I-6}$$

4.1 Updating $\tilde{y}$

[2]Each $\mathcal{L}_n$ is independent, and this makes the notation easier to read.

Optimizing (A I-6) w.r.t. $\tilde{y}$ is convex ; strong duality holds if a feasible points exist. In this case, the solution of the primal problem (A I-6) is the same as the dual problem:

$$\min_{\tilde{y}} \max_{\boldsymbol{\lambda}\geq 0,\boldsymbol{\nu}} \mathcal{L}_n(\tilde{y}^n, \boldsymbol{\lambda}, \boldsymbol{\nu}) = \max_{\boldsymbol{\lambda}\geq 0,\boldsymbol{\nu}} \min_{\tilde{y}^n} \mathcal{L}_n(\tilde{y}, \boldsymbol{\lambda}, \boldsymbol{\nu}) \tag{A I-7}$$

Therefore, the global optimum can be obtained by setting the derivative equal to 0. Notice that $\mathcal{L}_n$ is separable over variable $\tilde{y}_p^n$. In fact, we can write $\mathcal{L}_n$ (up to a constant) in the form of sum of independent functions, each corresponding to one pixel $p \in \Omega$:

$$-\tilde{y}_p^{n\top} \log \boldsymbol{s}_{p,\boldsymbol{\theta}}^n + \tilde{y}_p^{n\top} \log \tilde{y}_p^n + \sum_i \lambda_{i,n}(\tilde{y}_p^{n\top} a_i) + \nu_{p,n}\mathbf{1}^\top \tilde{y}_p. \tag{A I-8}$$

Setting the derivative w.r.t. $\tilde{y}_p^n$ equal to zero gives:

$$-\log \boldsymbol{s}_{p,\boldsymbol{\theta}}^n + \log \tilde{y}_p^n + \sum_i \lambda_{i,n} a_i + (\nu_{p,n} + 1)\mathbf{1} = 0 \tag{A I-9}$$

This yields the following closed-form solution:

$$\begin{aligned} \tilde{y}_{k,p}^{n*} &= e^{-\sum_i \lambda_{i,n} a_{i,k} + \log \boldsymbol{s}_{k,p,\boldsymbol{\theta}}^n} e^{-\nu_{p,n}-1} \\ \tilde{y}_{k,p}^{n*} &= \boldsymbol{s}_{k,p,\boldsymbol{\theta}}^n e^{-\sum_i \lambda_{i,n} a_{i,k}} e^{-\nu_{p,n}-1}, \end{aligned} \tag{A I-10}$$

where k represent the class number.

4.2 Computing the dual function

We want to maximize the dual function, which is given by

$$g_n(\boldsymbol{\lambda}, \boldsymbol{\nu}) = \min_{\tilde{y}^n} \mathcal{L}_n(\tilde{y}^n, \boldsymbol{\lambda}, \boldsymbol{\nu}) = \mathcal{L}_n(\tilde{y}^{n*}, \boldsymbol{\lambda}, \boldsymbol{\nu}).$$

The dual function is concave w.r.t. dual variables $\boldsymbol{\lambda}, \boldsymbol{\nu}$ (minimum of linear functions). Maximizing g_n w.r.t. $\boldsymbol{\nu}$ can be done in closed-form by setting the derivative of $g_n(\boldsymbol{\lambda}, \boldsymbol{\nu})$ w.r.t. each

Algorithm I-1 Overview of Pathak *et al.* (2015a) method.

Output: *Network parameters* $\boldsymbol{\theta}$ Init $\boldsymbol{\lambda}$ to 0
Init $\tilde{y}^n$ with the image level label
while $\tilde{y}^n$ *not converged* **do**
    $\boldsymbol{\theta}^t = \arg\min_{\boldsymbol{\theta}} \mathcal{L}_{\text{Cross-entropy}}(\tilde{y}^n, s^n_{\boldsymbol{\theta}}))$
    while *Not converged* **do**
        Solve $\tilde{y}^{n*}$ analytically with Equation (A I-12)
        Update $\boldsymbol{\lambda}$ with projected gradient descent in Equation (A I-13)
    end
    $\tilde{y}^n = \tilde{y}^{n*}$
end

$\boldsymbol{\nu}_{p,n}$ equal to zero, which yields simplex constraints $\mathbf{1}^\top \tilde{y}^n_p - 1 = 0 \quad \forall p$. Plugging (A I-10) into the simplex constraint yields the following closed-form optimality condition over each ν_p:

$$\begin{aligned} \mathbf{1}^\top \tilde{y}^{n*}_p - 1 &= 0 \\ \Leftrightarrow \sum_k \tilde{y}^{n*}_{k,p} &= 1 \\ \Leftrightarrow e^{-\nu_{i,n}-1} &= \frac{1}{\sum_k \boldsymbol{s}^n_{k,p,\boldsymbol{\theta}} e^{-\sum_i \lambda_{i,n} a_{i,k}}} \end{aligned} \tag{A I-11}$$

Plugging back into (A I-10) yields the following solution:

$$\tilde{y}^{n*}_{k,p} = \frac{\boldsymbol{s}^n_{k,p,\boldsymbol{\theta}} e^{-\sum_i \lambda_{i,n} a_{i,k}}}{\sum_{k'} \boldsymbol{s}^n_{k',p,\boldsymbol{\theta}} e^{-\sum_i \lambda_{i,n} a_{i,k'}}} \tag{A I-12}$$

Now the dual function depends only on $\boldsymbol{\lambda}$: $g_n(\boldsymbol{\lambda}) = \mathcal{L}_n(\tilde{y}^{n*}, \boldsymbol{\lambda})$.

Now, again, the dual function is concave w.r.t. $\boldsymbol{\lambda}$ and, therefore, can be optimized globally with projected gradient ascent. The gradient of the dual function w.r.t. to $\boldsymbol{\lambda}$ is

$$\nabla L_{n\lambda} = \frac{\partial \mathcal{L}_n(\tilde{y}^{n*}, \boldsymbol{\lambda})}{\partial \boldsymbol{\lambda}} = \tilde{y}^{n*\top} a_i - b_i$$

4.3 Projected gradient ascent

To solve $max_{\lambda \geq 0} \mathcal{L}_n(\tilde{y}^{n*}, \boldsymbol{\lambda})$, we use a projected gradient ascent ; we simply choose the point nearest to $\boldsymbol{\lambda}^t + \nabla \mathcal{L}_{n\boldsymbol{\lambda}}(\tilde{y}^{n*}, \boldsymbol{\lambda})$ in the set $\{\boldsymbol{\lambda} \geq 0\}$. This gives the following updates:

$$\boldsymbol{\lambda}^{t+1} = \begin{cases} \boldsymbol{\lambda}^t + \nabla \mathcal{L}_{n\boldsymbol{\lambda}}(\tilde{y}^{n*}, \boldsymbol{\lambda}^t) & \text{if} \quad \nabla \mathcal{L}_{n\boldsymbol{\lambda}}(\tilde{y}^{n*}, \boldsymbol{\lambda}) \geq 0 \\ 0 & \text{otherwise} \end{cases} \tag{A I-13}$$

The overall algorithm is summarized in Algorithm I-1.

APPENDIX II

ADDITIONAL MATERIALS FOR CHAPTER 6

1. DeepCut training instability

We investigated the generated pseudo-labels (as showed in Figure II-1) by DeepCut, and the main culprit is when the proposal under-segment the object inside the box. This forces, at the next training step, the network to segment the object as background. This kind of conflicting feedback to the network (some other proposal label similar looking patches as foreground) makes the training unstable, and slowly skew the network toward empty predictions. This will cause the next batch of proposals to be even smaller, until the network outputs empty foreground for all the images.

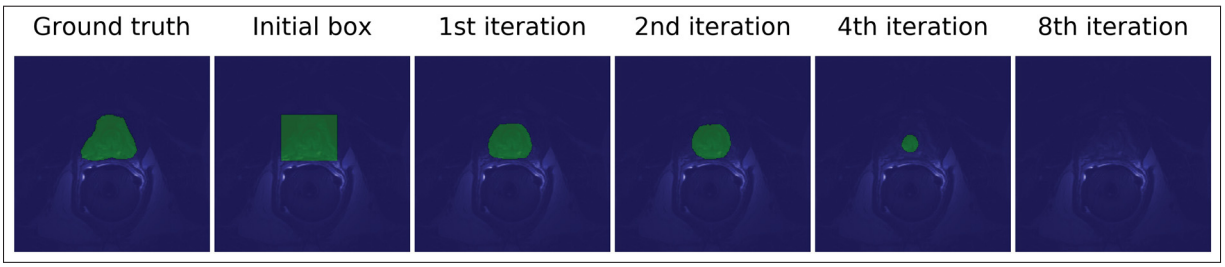


Figure-A II-1 Progression of the pseudo-labels from DeepCut: only a few of those cases can make the training very unstable.

2. Implementation and performances

Performances were measured on a machine equipped with an AMD Ryzen 1700X, 32GB of RAM (frequency did not affect speed) and an NVIDIA Titan RTX. They are reported in Table II-1. The settings and hyper-parameters are the same as described in Section 6.4.2.

Most of the extra time introduced by our model comes from the naive log-barrier implementation that we used. Instead of leveraging `if/else` switch and code vectorization we used a standard Python `for` loop over all constraints. This could be improved using the recent PyTorch development of its JIT compiler. The width parameter of the segments will affect the overhead

of our method: wider segments means less of them, which, in turns, results in less constraints to handle.

Notice that implementing the DenseCRF post-processing in a parallel and efficient fashion introduces a lot of software engineering uncommon in modern learning frameworks. While the DenseCRF implementation itself is highly efficient, it remains a single process that can handle only one image at a time. Performing it in parallel should be easy in theory, but is actually not very efficient with Python standard multiprocessing tools. In practice, all the arrays (containing either the image or probabilities) are pickled and copied across processes. Those back-and-forth copies can add up quickly and slow-down the processing substantially, on top of filling the computer memory more quickly. The solution is to carefully use SharedArray[1], which will contain all the batch in a single object. The sub-processed will read and write only a subset of those SharedArrays, corresponding to their assigned batch item.

Table-A II-1 Comparison in training speed between the different methods on the two datasets, PROMISE12 (Pr) and ATLAS (At).

	Time per epoch (s)		Proposals update (s)		**Total (h)**	
Method	Pr	At	Pr	At	Pr	At
Full supervision	150	235	-	-	4.2	3.3
Ours	170	325	-	-	4.7	4.5
DeepCut	150	235	440	3120	6.6	11.9

[1]Carefully, because they are not concurrency safe.

BIBLIOGRAPHY

Abraham, N. & Khan, N. M. (2019). A novel focal tversky loss function with improved attention u-net for lesion segmentation. 683–687.

Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., Süsstrunk, S. et al. (2012). SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11), 2274–2282.

Arnab, A., Zheng, S., Jayasumana, S., Romera-Paredes, B., Larsson, M., Kirillov, A., Savchynskyy, B., Rother, C., Kahl, F. & Torr, P. H. (2018). Conditional random fields meet deep neural networks for semantic segmentation: Combining probabilistic graphical models with deep learning for structured prediction. *IEEE Signal Processing Magazine*, 35(1), 37–52.

Bai, W., Oktay, O., Sinclair, M., Suzuki, H., Rajchl, M., Tarroni, G., Glocker, B., King, A., Matthews, P. M. & Rueckert, D. (2017). Semi-supervised learning for network-based cardiac MR image segmentation. *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pp. 253–260.

Bateson, M., Kervadec, H., Dolz, J., Lombaert, H. & Ayed, I. B. (2019). Constrained domain adaptation for segmentation. *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 326–334.

Baur, C., Albarqouni, S. & Navab, N. (2017). Semi-supervised deep learning for fully convolutional networks. *MICCAI*, pp. 311–319.

Bearman, A., Russakovsky, O., Ferrari, V. & Li, F. (2016). What's the Point: Semantic Segmentation with Point Supervision. *European Conference on Computer Vision (ECCV)*, pp. 549–565. doi: 10.1007/978-3-319-46478-7_34.

Bernard, O., Lalande, A., Zotti, C., Cervenansky, F., Yang, X., Heng, P.-A., Cetin, I., Lekadir, K., Camara, O., Ballester, M. A. G. et al. (2018). Deep learning techniques for automatic MRI cardiac multi-structures segmentation and diagnosis: Is the problem solved? *IEEE TMI*, 37(11), 2514–2525.

Bertsekas, D. P. (1995). *Nonlinear Programming*. Athena Scientific, Belmont, MA.

Blake, A., Kohli, P. & Rother, C. (2011). *Markov random fields for vision and image processing*. Mit Press.

Boyd, S. & Vandenberghe, L. (2004). *Convex Optimization*. Cambridge University Press.

Boykov, Y. & Funka-Lea, G. (2006). Graph Cuts and Efficient N-D Image Segmentation. *International Journal of Computer Vision*, 70, 109–131.

Boykov, Y., Kolmogorov, V., Cremers, D. & Delong, A. (2006). An integral solution to surface evolution PDEs via geo-cuts. *European Conference on Computer Vision*, pp. 409–422.

Boykov, Y., Isack, H. N., Olsson, C. & Ayed, I. B. (2015). Volumetric Bias in Segmentation and Reconstruction: Secrets and Solutions. *IEEE International Conference on Computer Vision (ICCV)*, pp. 1769–1777. doi: 10.1109/ICCV.2015.206.

Brosch, T., Yoo, Y., Tang, L. Y., Li, D. K., Traboulsee, A. & Tam, R. (2015). Deep convolutional encoder networks for multiple sclerosis lesion segmentation. *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 3–11.

Buhmann, J. M., Ferrari, V. & Vezhnevets, A. (2012). Weakly supervised structured output learning for semantic segmentation. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 845–852.

Caselles, V., Kimmel, R. & Sapiro, G. (1997). Geodesic Active Contours. *International Journal of Computer Vision*, 22, 61–79.

Chapelle, O., Schölkopf, B. & Zien, A. (2006). *Semi-Supervised Learning (Adaptive Computation and Machine Learning Series)*. The MIT Press.

Chapelle, O., Scholkopf, B. & Zien, A. (2009). Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]. *IEEE Transactions on Neural Networks*, 20(3), 542–542.

Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K. & Yuille, A. L. (2015). Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs. *ICLR*.

Cootes, T. F., Taylor, C. J., Cooper, D. H. & Graham, J. (1995). Active shape models-their training and application. *Computer vision and image understanding*, 61(1), 38–59.

Dai, J., He, K. & Sun, J. (2015). Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation. *IEEE International Conference on Computer Vision (ICCV)*, pp. 1635–1643.

Deng, Z., Todorovic, S. & Jan Latecki, L. (2015). Semantic segmentation of rgbd images with mutex constraints. *Proceedings of the IEEE international conference on computer vision*, pp. 1733–1741.

Dolz, J., Desrosiers, C. & Ayed, I. B. (2017). 3D fully convolutional networks for subcortical segmentation in MRI: A large-scale study. *NeuroImage*, -.

Dolz, J., Desrosiers, C. & Ben Ayed, I. (2018). 3D fully convolutional networks for subcortical segmentation in MRI: A large-scale study. *NeuroImage*, 170, 456–470.

Dong, N., Kampffmeyer, M., Liang, X., Wang, Z., Dai, W. & Xing, E. (2018). Unsupervised domain adaptation for automatic estimation of cardiothoracic ratio. *MICCAI*, pp. 544–552.

Everingham, M., Van Gool, L., Williams, C. K., Winn, J. & Zisserman, A. (2010). The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2), 303–338.

Fiacco, A. V. & McCormick, G. P. (1990). *Nonlinear programming: sequential unconstrained minimization techniques*. SIAM.

Fletcher, R. (1987). *Practical Methods of Optimization*. John Wiley & Sons.

Ganaye, P.-A., Sdika, M. & Benoit-Cattin, H. (2018). Semi-supervised learning for segmentation under semantic constraint. *MICCAI*, pp. 595–602.

Gill, P., Murray, W. & Wright, M. (1981). *Practical Optimization*. Academic Press.

Goodfellow, I., Bengio, Y. & Courville, A. (2016). *Deep learning*. MIT press.

Gorelick, L., Schmidt, F. R. & Boykov, Y. (2013). Fast Trust Region for Segmentation. *Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1714–1721. doi: 10.1109/CVPR.2013.224.

Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V. & Courville, A. C. (2017). Improved Training of Wasserstein GANs. *Neural Information Processing Systems (NIPS)*, pp. 5767–5777.

Hardt, M., Recht, B. & Singer, Y. (2016). Train faster, generalize better: Stability of stochastic gradient descent. *International Conference on Machine Learning (ICML)*, pp. 1225-1234.

Havaei, M., Davy, A., Warde-Farley, D., Biard, A., Courville, A., Bengio, Y., Pal, C., Jodoin, P.-M. & Larochelle, H. (2017). Brain tumor segmentation with deep neural networks. *Medical image analysis*, 35, 18–31.

He, F. S., Liu, Y., Schwing, A. G. & Peng, J. (2017). Learning to Play in a Day: Faster Deep Reinforcement Learning by Optimality Tightening. *International Conference on Learning Representations (ICLR)*, pp. 1-13.

Hsu, C.-C., Hsu, K.-J., Tsai, C.-C., Lin, Y.-Y. & Chuang, Y.-Y. (2019). Weakly Supervised Instance Segmentation using the Bounding Box Tightness Prior. *Advances in Neural Information Processing Systems*, pp. 6582–6593.

Hu, Z., Yang, Z., Salakhutdinov, R., Qin, L., Liang, X., Dong, H. & Xing, E. P. (2018). Deep Generative Models with Learnable Knowledge Constraints. *Neural Information Processing Systems (NeurIPS)*, pp. 10522–10533.

Jia, Z., Huang, X., Eric, I., Chang, C. & Xu, Y. (2017). Constrained deep weak supervision for histopathology image segmentation. *IEEE Transactions on Medical Imaging*, 36(11), 2376–2388.

Kamnitsas, K., Ledig, C., Newcombe, V. F., Simpson, J. P., Kane, A. D., Menon, D. K., Rueckert, D. & Glocker, B. (2017). Efficient multi-scale 3D CNN with fully connected CRF for accurate brain lesion segmentation. *Medical image analysis*, 36, 61–78.

Karimi, D. & Salcudean, S. E. (2019). Reducing the Hausdorff distance in medical image segmentation with convolutional neural networks. *IEEE transactions on medical imaging*.

Ker, J., Wang, L., Rao, J. & Lim, T. (2018). Deep learning applications in medical image analysis. *IEEE Access*, 6, 9375–9389.

Kervadec, H., Dolz, J., Granger, E. & Ben Ayed, I. (2019a). Curriculum semi-supervised segmentation. *International Conference on Medical Image Computing and Computer-Assisted Intervention*.

Kervadec, H., Dolz, J., Tang, M., Granger, E., Boykov, Y. & Ayed, I. B. (2019b). Constrained-CNN losses for weakly supervised segmentation. *Medical image analysis*, 54, 88–99.

Kervadec, H., Dolz, J., Yuan, J., Desrosiers, C., Granger, E. & Ben Ayed, I. (2020). Constrained Deep Networks: Lagrangian Optimization via Log-Barrier Extensions. *arXiv preprint arXiv:1904.04205*.

Khoreva, A., Benenson, R., Hosang, J. H., Hein, M. & Schiele, B. (2017). Simple Does It: Weakly Supervised Instance and Semantic Segmentation. *CVPR*, 1(2), 3.

Kingma, D. P. & Ba, J. (2015). Adam: A Method for Stochastic Optimization. *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. Consulted at http://arxiv.org/abs/1412.6980.

Klodt, M. & Cremers, D. (2011). A convex framework for image segmentation with moment constraints. *IEEE International Conference on Computer Vision (ICCV)*, pp. 2236–2243.

Koch, L. M., Rajchl, M., Bai, W., Baumgartner, C. F., Tong, T., Passerat-Palmbach, J., Aljabar, P. & Rueckert, D. (2018). Multi-atlas segmentation using partially annotated data: Methods and annotation strategies. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(7), 1683–1696.

Kolesnikov, A. & Lampert, C. H. (2016). Seed, expand and constrain: Three principles for weakly-supervised image segmentation. *European Conference on Computer Vision (ECCV)*, pp. 695–711.

Krähenbühl, P. & Koltun, V. (2011a). Efficient inference in fully connected crfs with gaussian edge potentials. *Advances in neural information processing systems*, pp. 109–117.

Krähenbühl, P. & Koltun, V. (2011b). Efficient inference in fully connected crfs with gaussian edge potentials. *Advances in neural information processing systems*, pp. 109–117.

LeCun, Y., Bottou, L., Bengio, Y. & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278–2324.

Lempitsky, V., Kohli, P., Rother, C. & Sharp, T. (2009). Image segmentation with a bounding box prior. *2009 IEEE 12th international conference on computer vision*, pp. 277–284.

Liew, S.-L., Anglin, J. M., Banks, N. W., Sondag, M., Ito, K. L., Kim, H., Chan, J., Ito, J., Jung, C., Khoshab, N. et al. (2018). A large, open source dataset of stroke anatomical brain images and manual lesion segmentations. *Scientific data*, 5, 180011.

Lim, Y., Jung, K. & Kohli, P. (2014). Efficient Energy Minimization for Enforcing Label Statistics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(9), 1893–1899.

Lin, D., Dai, J., Jia, J., He, K. & Sun, J. (2016). Scribblesup: Scribble-supervised convolutional networks for semantic segmentation. *CVPR*, pp. 3159–3167.

Lin, T.-Y., Goyal, P., Girshick, R., He, K. & Dollár, P. (2018). Focal loss for dense object detection. *IEEE transactions on pattern analysis and machine intelligence*.

Litjens, G., Toth, R., van de Ven, W., Hoeks, C., Kerkstra, S., van Ginneken, B., Vincent, G., Guillard, G., Birbeck, N., Zhang, J. et al. (2014). Evaluation of prostate segmentation algorithms for MRI: the PROMISE12 challenge. *Medical image analysis*, 18(2), 359–373.

Litjens, G. J. S., Kooi, T., Bejnordi, B. E., Setio, A. A. A., Ciompi, F., Ghafoorian, M., van der Laak, J. A. W. M., van Ginneken, B. & Sánchez, C. I. (2017). A survey on deep learning in medical image analysis. *Medical Image Analysis*, 42, 60–88.

doi: 10.1016/j.media.2017.07.005.

Long, J., Shelhamer, E. & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3431–3440.

Ma, J., Wei, Z., Zhang, Y., Wang, Y., Lv, R., Zhu, C., Chen, G., Liu, J., Peng, C., Wang, L. et al. (2020). How Distance Transform Maps Boost Segmentation CNNs: An Empirical Study. *Medical Imaging with Deep Learning*.

Marin, D., Tang, M., I., B. A. & Y., B. (2019). Beyond Gradient Descent for Regularized Segmentation Losses. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10187–10196.

Márquez-Neila, P., Salzmann, M. & Fua, P. (2017). Imposing Hard Constraints on Deep Networks: Promises and Limitations. *CVPR Workshop on Negative Results in Computer Vision*, pp. 1–9.

Milletari, F., Navab, N. & Ahmadi, S.-A. (2016). V-Net: Fully convolutional neural networks for volumetric medical image segmentation. *3D Vision (3DV), 2016 Fourth International Conference on*, pp. 565–571.

Milletari, F., Ahmadi, S.-A., Kroll, C., Plate, A., Rozanski, V., Maiostre, J., Levin, J., Dietrich, O., Ertl-Wagner, B., Bötzel, K. et al. (2017). Hough-CNN: deep learning for segmentation of deep brain regions in MRI and ultrasound. *Computer Vision and Image Understanding*, 164, 92–102.

Mitiche, A. & Ben Ayed, I. (2011). *Variational and level set methods in image segmentation*. Springer.

Mondal, A. K., Dolz, J. & Desrosiers, C. (2018). Few-shot 3D Multi-modal Medical Image Segmentation using Generative Adversarial Learning. *arXiv:1810.12241*.

Nandwani, Y., Pathak, A., Singla, P. et al. (2019). A Primal Dual Formulation For Deep Learning With Constraints. *Advances in Neural Information Processing Systems*, pp. 12157–12168.

Nguyen, H.-G., Pica, A., Rosa, F. L., Hrbacek, J., Weber, D. C., Schalenbourg, A., Sznitman, R. & Cuadra, M. B. (2019). A novel segmentation framework for uveal melanoma based on magnetic resonance imaging and class activation maps. *International Conference on Medical Imaging with Deep Learning*.

Nie, D., Gao, Y., Wang, L. & Shen, D. (2018). ASDNet: Attention Based Semi-supervised Deep Networks for Medical Image Segmentation. *MICCAI*, pp. 370–378.

Niethammer, M. & Zach, C. (2013). Segmentation with area constraints. *Medical Image Analysis*, 17(1), 101–112. doi: 10.1016/j.media.2012.09.002.

Papandreou, G., Chen, L.-C., Murphy, K. P. & Yuille, A. L. (2015). Weakly-and semi-supervised learning of a deep convolutional network for semantic image segmentation. *Proceedings of the IEEE international conference on computer vision (ICCV)*, pp. 1742–1750.

Papert, S. A. (1966). The summer vision project.

Paszke, A., Chaurasia, A., Kim, S. & Culurciello, E. (2016). Enet: A deep neural network architecture for real-time semantic segmentation. *arXiv preprint arXiv:1606.02147*.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L. & Lerer, A. (2017). Automatic differentiation in pytorch.

Pathak, D., Krahenbuhl, P. & Darrell, T. (2015a). Constrained convolutional neural networks for weakly supervised segmentation. *International Conference on Computer Vision (ICCV)*, pp. 1796–1804.

Pathak, D., Shelhamer, E., Long, J. & Darrell, T. (2015b). Fully convolutional multi-class multiple instance learning. *ICLR Workshop*.

Pinheiro, P. O. & Collobert, R. (2015). From image-level to pixel-level labeling with convolutional networks. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1713–1721.

Platt, J. C. & Barr, A. H. (1988). *Constrained differential optimization.*

Pont-Tuset, J., Arbelaez, P., Barron, J. T., Marques, F. & Malik, J. (2017). Multiscale combinatorial grouping for image segmentation and object proposal generation. *IEEE transactions on pattern analysis and machine intelligence*, 39(1), 128–140.

Pu, M., Huang, Y., Guan, Q. & Zou, Q. (2018). GraphNet: Learning Image Pseudo Annotations for Weakly-Supervised Semantic Segmentation. *2018 ACM Multimedia Conference on Multimedia Conference*, pp. 483–491.

Qu, H., Wu, P., Huang, Q., Yi, J., Riedlinger, G. M., De, S. & Metaxas, D. N. (2019). Weakly supervised deep nuclei segmentation using points annotation in histopathology images. *International Conference on Medical Imaging with Deep Learning*, pp. 390–400.

Quan, T. M., Hildebrand, D. G. & Jeong, W.-K. (2016). Fusionnet: A deep fully residual convolutional neural network for image segmentation in connectomics. *arXiv preprint arXiv:1612.05360*.

Rajchl, M., Lee, M. C., Schrans, F., Davidson, A., Passerat-Palmbach, J., Tarroni, G., Alansary, A., Oktay, O., Kainz, B. & Rueckert, D. (2016). Learning under distributed weak supervision. *arXiv preprint arXiv:1606.01100*.

Rajchl, M., Lee, M. C., Oktay, O., Kamnitsas, K., Passerat-Palmbach, J., Bai, W., Damodaram, M., Rutherford, M. A., Hajnal, J. V., Kainz, B. et al. (2017). DeepCut: Object segmentation from bounding box annotations using convolutional neural networks. *IEEE Transactions on Medical Imaging*, 36(2), 674–683.

Ravi, S. N., Dinh, T., Lokhande, V. S. & Singh, V. (2019). Explicitly imposing constraints in deep networks via conditional gradients gives improved generalization and faster convergence. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33, 4772–4779.

Ronneberger, O., Fischer, P. & Brox, T. (2015). U-Net: Convolutional networks for biomedical image segmentation. *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241.

Rony, J., Hafemann, L. G., Oliveira, L. S., Ben Ayed, I., Sabourin, R. & Granger, E. (2019). Decoupling Direction and Norm for Efficient Gradient-Based L2 Adversarial Attacks and Defenses. *Computer Vision and Pattern Recognition (CVPR)*, pp. 1-10.

Rother, C., Kolmogorov, V. & Blake, A. (2004). Grabcut: Interactive foreground extraction using iterated graph cuts. *ACM transactions on graphics (TOG)*, 23(3), 309–314.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M. et al. (2015). Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3), 211–252.

Salehi, S. S. M., Erdogmus, D. & Gholipour, A. (2017). Tversky loss function for image segmentation using 3D fully convolutional deep networks. *International Workshop on Machine Learning in Medical Imaging*, pp. 379–387.

Sedai, S., Mahapatra, D., Hewavitharanage, S., Maetschke, S. & Garnavi, R. (2017). Semi-supervised segmentation of optic cup in retinal fundus images using variational autoencoder. *MICCAI*, pp. 75–82.

Shen, D., Wu, G. & Suk, H.-I. (2017). Deep learning in medical image analysis. *Annual review of biomedical engineering*, 19, 221–248.

Song, C., Huang, Y., Ouyang, W. & Wang, L. (2019). Box-driven class-wise region masking and filling rate guided loss for weakly supervised semantic segmentation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3136–3145.

Sudre, C. H., Li, W., Vercauteren, T., Ourselin, S. & Cardoso, M. J. (2017). Generalised Dice overlap as a deep learning loss function for highly unbalanced segmentations. In *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support* (pp. 240–248). Springer.

Tang, M., Djelouah, A., Perazzi, F., Boykov, Y. & Schroers, C. (2018a). Normalized Cut Loss for Weakly-supervised CNN Segmentation. *Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1818-1827.

Tang, M., Perazzi, F., Djelouah, A., Ben Ayed, I., Schroers, C. & Boykov, Y. (2018b). On Regularized Losses for Weakly-supervised CNN Segmentation. *European Conference on Computer Vision (ECCV), Part XVI*, pp. 524-540.

Trafalis, T. B., Tutunji, T. A. & Couellan, N. P. (1997). Interior Point Methods for Supervised Training of Artificial Neural Networks with Bounded Weights. *Network Optimization*, pp. 441-470.

Turing, A. M. (1950). Computing machinery and intelligence.

Valverde, S., Cabezas, M., Roura, E., González-Villà, S., Pareto, D., Vilanova, J. C., Ramio-Torrenta, L., Rovira, À., Oliver, A. & Lladó, X. (2017). Improving automated multiple sclerosis lesion segmentation with a cascaded 3D convolutional neural network approach. *NeuroImage*, 155, 159–168.

Verbeek, J. & Triggs, B. (2007). Region classification with markov field aspect models. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–8.

Vernaza, P. & Chandraker, M. (2017). Learning random-walk label propagation for weakly-supervised semantic segmentation. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 3, 3.

Vezhnevets, A. & Buhmann, J. M. (2010). Towards weakly supervised semantic segmentation by means of multiple instance and multitask learning. *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pp. 3249–3256.

Vezhnevets, A., Ferrari, V. & Buhmann, J. M. (2011). Weakly supervised semantic segmentation with a multi-image model. *Computer Vision (ICCV), 2011 IEEE International Conference on*, pp. 643–650.

Wei, Y., Liang, X., Chen, Y., Shen, X., Cheng, M.-M., Feng, J., Zhao, Y. & Yan, S. (2017). Stc: A simple to complex framework for weakly-supervised semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(11), 2314–2320.

Weston, J., Ratle, F., Mobahi, H. & Collobert, R. (2012). Deep learning via semi-supervised embedding. In *Neural Networks: Tricks of the Trade* (pp. 639–655). Springer.

Wong, K. C., Moradi, M., Tang, H. & Syeda-Mahmood, T. (2018). 3D segmentation with exponential logarithmic loss for highly unbalanced object sizes. *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 612–619.

Wu, K., Du, B., Luo, M., Wen, H., Shen, Y. & Feng, J. (2019). Weakly Supervised Brain Lesion Segmentation via Attentional Representation Learning. *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 211–219.

Xie, S., Girshick, R., Dollár, P., Tu, Z. & He, K. (2017). Aggregated residual transformations for deep neural networks. *CVPR*, pp. 1492–1500.

Xu, J., Schwing, A. G. & Urtasun, R. (2014). Tell me what you see and i will show you where it is. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3190–3197.

Xu, J., Schwing, A. G. & Urtasun, R. (2015). Learning to segment under various forms of weak supervision. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3781–3790.

Yu, L., Yang, X., Chen, H., Qin, J. & Heng, P.-A. (2017). Volumetric ConvNets with Mixed Residual Connections for Automated Prostate Segmentation from 3D MR Images. *AAAI*, pp. 66–72.

Zhang, S. & Constantinides, A. (1992). Lagrange programming neural networks. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, 39(7), 441-452.

Zhang, Y., David, P. & Gong, B. (2017a). Curriculum Domain Adaptation for Semantic Segmentation of Urban Scenes. *International Conference on Computer Vision (ICCV)*, pp. 2039–2049. doi: 10.1109/ICCV.2017.223.

Zhang, Y., David, P., Foroosh, H. & Gong, B. (2019). A Curriculum Domain Adaptation Approach to the Semantic Segmentation of Urban Scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Zhang, Y., Yang, L., Chen, J., Fredericksen, M., Hughes, D. P. & Chen, D. Z. (2017b). Deep adversarial networks for biomedical image segmentation utilizing unannotated images. *MICCAI*, pp. 408–416.

Zheng, S., Jayasumana, S., Romera-Paredes, B., Vineet, V., Su, Z., Du, D., Huang, C. & Torr, P. H. (2015). Conditional random fields as recurrent neural networks. *Proceedings of the IEEE international conference on computer vision*, pp. 1529–1537.

Zhou, Y., Li, Z., Bai, S., Wang, C., Chen, X., Han, M., Fishman, E. & Yuille, A. (2019a). Prior-aware Neural Network for Partially-Supervised Multi-Organ Segmentation. *International Conference on Computer vision (ICCV)*, pp. 1-10.

Zhou, Y., Wang, Y., Tang, P., Bai, S., Shen, W., Fishman, E. & Yuille, A. (2019b). Semi-Supervised 3D Abdominal Multi-Organ Segmentation Via Deep Multi-Planar Co-Training. *IEEE WACV*, pp. 121–140.

Zhou, Y., Wang, Y., Tang, P., Bai, S., Shen, W., Fishman, E. K. & Yuille, A. L. (2019c). Semi-Supervised 3D Abdominal Multi-Organ Segmentation Via Deep Multi-Planar Co-Training. 121–140. doi: 10.1109/WACV.2019.00020.

Zhu, W., Huang, Y., Zeng, L., Chen, X., Liu, Y., Qian, Z., Du, N., Fan, W. & Xie, X. (2019). AnatomyNet: Deep learning for fast and fully automated whole-volume segmentation of head and neck anatomy. *Medical physics*, 46(2), 576–589.

Zhu, X. & Goldberg, A. B. (2009). Introduction to semi-supervised learning. *Synthesis lectures on artificial intelligence and machine learning*, 3(1), 1–130.