POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

On medical image segmentation and on modeling long term dependencies

EUGENE VORONTSOV
Département de génie informatique et génie logiciel

Thèse présentée en vue de l'obtention du diplôme de *Philosophiæ Doctor*
Génie informatique

Mai 2020

© Eugene Vorontsov, 2020.

POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Cette thèse intitulée :

On medical image segmentation and on modeling long term dependencies

présentée par **Eugene VORONTSOV**
en vue de l'obtention du diplôme de *Philosophiæ Doctor*
a été dûment acceptée par le jury d'examen constitué de :

Michel DESMARAIS, président
Samuel KADOURY, membre et directeur de recherche
Christopher J. PAL, membre et codirecteur de recherche
Benjamin DE LEENER, membre
Ismail BEN AYED, membre externe

DEDICATION

To all the curious and passionate.

ACKNOWLEDGEMENTS

There's something both satisfying and anti-climactic about completing a dissertation after years of study. It's amazing to think that so much time has passed since I began, not knowing at first what it is really that I'm doing but guessing what it is that I would like to do or learn. It's easy to forget those who made a difference through that transformative journey; these are acknowledgements for the work in this dissertation but these are also thanks for the people in this part of my life. I would like to thank my advisors for the opportunities that they provided me with. Samuel Kadoury, who took me on as an intern and then as a student in Montreal when I thought it may be difficult to find a graduate position. Chris Pal, who encouraged me before I was even his student, telling me what it means to be a researcher. Thanks also for tossing me some funds when I ran very low, even though I was not yet your student. Thanks also to both for the freedom and independence I enjoyed in my graduate studies. Thanks, of course, to Roland Memisevic who was so ready to discuss all those things that fascinated me or that I didn't understand when I was just beginning. And thanks to Yoshua Bengio for encouraging me so warmly, once he finally learned my name. And to all those others with whom I've had interesting, exciting, or even frustrating discussions about research.

But beyond that, I would like to thank all the friends I've made in the labs at Poly and Mila, at Imagia, at Microsoft Research, and at Nvidia who made all those places feel a bit like home and made the dark and stressful times lighter. Finally, thanks to my family for their patience and understanding.

RÉSUMÉ

La délimitation (segmentation) des tumeurs malignes à partir d'images médicales est importante pour le diagnostic du cancer, la planification des traitements ciblés, ainsi que les suivis de la progression du cancer et de la réponse aux traitements. Cependant, bien que la segmentation manuelle des images médicales soit précise, elle prend du temps, nécessite des opérateurs experts et est souvent peu pratique lorsque de grands ensembles de données sont utilisés. Ceci démontre la nécessité d'une segmentation automatique. Cependant, la segmentation automatisée des tumeurs est particulièrement difficile en raison de la variabilité de l'apparence des tumeurs, de l'équipement d'acquisition d'image et des paramètres d'acquisition, et de la variabilité entre les patients. Les tumeurs varient en type, taille, emplacement et quantité; le reste de l'image varie en raison des différences anatomiques entre les patients, d'une chirurgie antérieure ou d'une thérapie ablative, de différences dans l'amélioration du contraste des tissus et des artefacts d'image. De plus, les protocoles d'acquisition du scanner varient considérablement entre les cliniques et les caractéristiques de l'image varient selon le modèle du scanner. En raison de toutes ces variabilités, un modèle de segmentation doit être suffisamment flexible pour apprendre les caractéristiques générales des données.

L'avènement des réseaux profonds de neurones à convolution (*convolutional neural networks*, CNN) a permis une classification exacte et précise des images hautement variables et, par extension, une segmentation de haute qualité des images. Cependant, ces modèles doivent être formés sur d'énormes quantités de données étiquetées. Cette contrainte est particulièrement difficile dans le contexte de la segmentation des images médicales, car le nombre de segmentations pouvant être produites est limité dans la pratique par la nécessité d'employer des opérateurs experts pour réaliser un tel étiquetage. De plus, les variabilités d'intérêt dans les images médicales semblent suivre une distribution à longue traîne, ce qui signifie qu'un nombre particulièrement important de données utilisées pour l'entraînement peut être nécessaire pour fournir un échantillon suffisant de chaque type de variabilité à un CNN. Cela démontre la nécessité de développer des stratégies pour la formation de ces modèles avec des segmentations de vérité-terrain disponibles limitées.

Cette thèse se concentre sur (1) l'établissement et l'évaluation de l'état de l'art en segmentation de tumeurs hépatiques; (2) le perfectionnement des réseaux de neurones profonds pour la segmentation en évaluant les connexions de saut et l'orthogonalité; et (3) surmonter le manque de vérité-terrain de segmentation en imagerie médicale.

En suivant (1), l'état de l'art en segmentation des tumeurs hépatiques est établi en aidant à

mettre en place le «défi de segmentation des tumeurs hépatiques» (*liver tumour segmentation challenge*, LiTS). Une soumission qui figure parmi les meilleures entrées de ce défi est présentée. Une évaluation de tous les résultats du défi (non inclus) révèle que la segmentation automatisée pour cette tâche reste médiocre. Néanmoins, des travaux supplémentaires évaluant l'utilité clinique des résultats montrent qu'en corrigeant une segmentation automatisée au lieu de segmenter à partir de zéro, les opérateurs experts peuvent réduire considérablement leur temps de segmentation. Dans ce même travail, la variabilité intra et inter-opérateurs des segmentations est évaluée à la fois pour les segmentations à partir de zéro et pour les corrections de segmentation automatisée; de plus, l'accord Bland Altman est évalué dans toutes les segmentations, y compris la segmentation automatisée. D'autres directions à explorer pour l'amélioration des modèles de segmentation sont proposées dans la discussion.

En suivant (2), la question de la disparition des gradients, qui se pose avec les réseaux de neurones profonds, est explorée. Le flux de gradients à travers un tel modèle est visualisé et, par conséquent, divers ajustements au modèle sont suggérés, tels que l'inclusion de connexions à saut court et la normalisation des couches. Plus académiquement, cette veine de recherche se poursuit vers l'étude d'une contrainte d'orthogonalité sur les matrices de poids des réseaux neuronaux. Cette étude est réalisée sur des réseaux de neurones récurrents comme un simple modèle de substitution pour d'autres réseaux de neurones profonds, afin de faciliter l'analyse. Les matrices orthogonales sont des opérateurs préservant les normes et sont donc utiles pour éviter à la fois la disparition et l'explosion des gradients pendant la rétropropagation des erreurs à travers un réseau. Cependant, cet ouvrage montre qu'il est avisé d'assouplir cette contrainte dans le spectre de valeurs singulières car cet assouplissement améliore la vitesse de convergence et les performances du modèle. On suppose dans un premier temps que cet assouplissement devrait augmenter l'expressivité des matrices; il est toutefois intéressant de noter que dans certains cas pathologiques de dépendances à long terme, l'écart optimal est si faible qu'on ne s'attend pas à ce qu'il produise un espace de paramètres beaucoup plus grand. D'autres hypothèses liées à la dynamique des réseaux, y compris la dynamique transitoire non normale, sont proposées dans la discussion pour expliquer les résultats de cette analyse.

Enfin, en suivant (3), un modèle semi-supervisé est proposé; ce modèle est adapté à un scénario courant dans la segmentation d'images médicales, où le contenu de toutes les images est connu mais de nombreuses images manquent la segmentation. Plus précisément, en l'absence d'un échantillon suffisant de données avec des étiquettes de segmentation au niveau des pixels, ce modèle utilise des étiquettes faibles qui sont souvent facilement disponibles. Pour les lésions hépatiques, ces étiquettes par image faibles déterminent si le cas est «sain» ou «malade». Ce modèle apprend à transformer des images d'un domaine à l'autre et *vice versa*. L'hypothèse est que pour effectuer cette transformation, le modèle doit apprendre à démêler

les mêmes facteurs de variation que pour la segmentation: c'est-à-dire que les variations qui provoquent l'apparition de lésions dans l'image («unique») doivent être séparées de ceux qui provoquent l'apparition du foie et de l'abdomen («communs»), de sorte qu'une image saine peut toujours être générée à partir de caractéristiques «communes» et une image malade peut toujours être générée à partir de la combinaison de caractéristiques «communes» et «uniques». Ainsi, cet objectif de translation de domaine sert comme une alternative non-supervisée pour une tache de segmentation. En outre, le décodeur de segmentation proposé a une double utilisation dans la tâche de traduction de domaine, de sorte qu'il obtient toujours des mises à jour de gradient avec chaque image d'entrée, même lorsque la plupart des images n'ont pas les étiquettes de segmentation nécessaires pour un signal d'erreur de segmentation. Ce nouveau modèle est évalué à la fois sur des données synthétiques et sur des données réelles basées sur un ensemble de données de tumeur cérébrale en IRM.

Dans l'ensemble, cette thèse établit une base à partir de laquelle améliorer la segmentation des images médicales, propose quelques nouveaux modèles dans cette direction et discute certaines directions de recherche qui naissent de cet ouvrage.

ABSTRACT

The delineation (segmentation) of malignant tumours in medical images is important for cancer diagnosis, the planning of targeted treatments, and the tracking of cancer progression and treatment response. However, although manual segmentation of medical images is accurate, it is time consuming, requires expert operators, and is often impractical with large datasets. This motivates the need for training automated segmentation. However, automated segmentation of tumours is particularly challenging due to variability in tumour appearance, image acquisition equipment and acquisition parameters, and variability across patients. Tumours vary in type, size, location, and quantity; the rest of the image varies due to anatomical differences between patients, prior surgery or ablative therapy, differences in contrast enhancement of tissues, and image artefacts. Furthermore, scanner acquisition protocols vary considerably between clinical sites and image characteristics vary according to the scanner model. Due to all of these variabilities, a segmentation model must be flexible enough to learn general features from the data.

The advent of deep convolutional neural networks (CNN) allowed for accurate and precise classification of highly variable images and, by extension, of high quality segmentation images. However, these models must be trained on enormous quantities of labeled data. This constraint is particularly challenging in the context of medical image segmentation because the number of segmentations that can be produced is limited in practice by the need to employ expert operators to do such labeling. Furthermore, the variabilities of interest in medical images appear to follow a long tail distribution, meaning a particularly large amount of training data may be required to provide a sufficient sample of each type of variability to a CNN. This motivates the need to develop strategies for training these models with limited ground truth segmentations available.

This dissertation focuses on (1) establishing and evaluating the state of the art of liver tumour segmentation; (2) working towards improving deep neural networks for segmentation by evaluating skip connections and orthogonality; and (3) overcoming the lack of segmentation ground truth in medical imaging.

Pursuing (1), the state of the art of liver tumour segmentation is established by helping to set up the liver tumour segmentation challenge (LiTS). A submission that placed among the top entries in this challenge is presented. An evaluation of all results in the challenge (not included) reveals that automated segmentation for this task remains poor. Nevertheless, further work evaluating the clinical utility of the results shows that by correcting an automated

segmentation instead of segmenting from scratch, expert operators can substantially reduce their segmentation time. In this same work, the intra- and inter-operator variability in segmentations is evaluated both for segmentations from scratch and for corrections of automated segmentation; also, the Bland Altman agreement is evaluated across all segmentations, including automated segmentation. Further directions for improvement of segmentation models are proposed in the discussion.

Pursuing (2), the issue of vanishing gradients that arises with deep neural networks is explored. The flow of gradients through such a model is visualized and consequently, various tweaks to the model are suggested, such as the inclusion of short skip connections and layer normalization. More academically, this vein of research is pursued further toward the study of an orthogonality constraint on neural network weight matrices. This study is performed on recurrent neural networks as a simple surrogate model for other deep neural networks, in order to ease analysis. Orthogonal matrices are norm-preserving operators and thus useful for avoiding both vanishing and exploding gradients during error backpropagation through a network. However, this work shows that it is prudent to relax this constraint in the singular value spectrum as this relaxation yields improved convergence speed and model performance. It is at first hypothesized that this relaxation should increase the expressivity of the matrices; interestingly however, in some pathological cases of long term dependencies, the optimal deviation is so small as to not be expected to yield a significantly larger parameter space. Other hypotheses related to network dynamics, including non-normal transient dynamics, are proposed in the discussion to explain the results of this analysis.

Finally, pursuing (3), a semi-supervised model is proposed that is tailored to a common scenario in medical image segmentation where the contents of all images are known but many images lack segmentations. Specifically, in the absence of a sufficient sample of data with pixel-level segmentation labels, this model uses weak labels that are often readily available. For liver lesions, these weak per-image labels are whether the case is a "healthy" one or a "sick" one. This model learns to transform images from one domain to the other and *vice versa*. It is hypothesized that in order to perform this transformation, the model must learn to disentangle the same factors of variation as it must for segmentation: that is, variations that cause lesions to appear in the image ("unique") must be separate from those that cause the appearance of the liver and abdomen ("common"), so that a healthy image could always be generated from "common" features and a sick image could always be generated from the combination of "common" and "unique" features. Thus, this domain translation objective acts as a good unsupervised surrogate for a segmentation objective. Furthermore, the proposed segmentation decoder has a dual use in the domain translation task, so that it always gets gradient updates with every input image, even when most images lack the

segmentation labels necessary for a segmentation error signal. This novel model is evaluated on both synthetic data and real data based on a dataset of brain tumour in MRI.

Overall, this dissertation defines a basis from which to improve segmentation for medical images, proposes some novel models in that direction, and discusses some directions of research that emerge from this work.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ACRONYMS

BRATS	Brain Tumour Segmentation (challenge)
CI	confidence interval
CLM	colorectal liver metastases
CT	computed tomography
CNN	convolutional neural network
CRF	conditional random field
DSC	Dice similarity coefficient
FastPD	Fast primal dual (optimization)
FCN	fully convolutional network
GAN	generative adversarial network
GRU	gated recurrent unit
ICC	intraclass correlation coefficient
ISBI	International Symposium on Biomedical Imaging
ISLES	Ischemic Stroke Lesion Segmentation
LiTS	Liver Tumour Segmentation (challenge)
LSTM	long short term memory
LUNA16	LUng Nodule Analysis 2016 (challenge)
MICCAI	The Medical Image Computing and Computer Assisted Intervention Society
MDRNN	multi-dimensional recurrent neural network
NRU	non-saturating recurrent unit
MRF	Markov random field
MRI	magnetic resonance imaging
PASCAL VOC	PASCAL Visual Object Classes (challenge)
ResNet	residual neural network
RNN	recurrent neural network
SLIVER07	MICCAI 2007 liver segmentation challenge

LIST OF APPENDICES

CHAPTER 1 INTRODUCTION

The detection and delineation of objects in images is critical for some workflows in medical image analysis and is common in computer vision, enabling diverse tasks such as pedestrian counting, facial recognition, and machine vision for navigation. Such object delineation is referred to as *semantic image segmentation*. This dissertation explores segmentation mainly in the context of cancerous tumour segmentation in medical images. Tumour segmentation is required to assess tumour load, to plan treatments such as radiotherapy or surgery, and to track treatment response and cancer progression. Although manually performed image segmentation is accurate, it is time-consuming and requires radiologists or medical technicians to perform it which greatly limits the number of segmentations that can be performed. As a consequence, tumour volume is often estimated by measuring the two largest mutually perpendicular diameters of a tumour across the transverse plane [5]. On the other hand, automated segmentation can help measure the tumour volume directly, allowing for better prediction of patient survival [6].

Automated segmentation is challenging. For tumour segmentation in computed tomography (CT) volumes (Figure 1.1), a model must capture the variability in tumour appearance, image acquisition equipment and acquisition parameters, and variability across patients. There are many different tumour types and even a single type presents different shapes with different textures. The sizes, locations, and quantity of tumours vary greatly. Furthermore, image statistics and the artefacts therein differ with the use of different scanner acquisition protocols or even different scanners. Even with a consistent protocol, tissue enhancement from contrast agent administration differs across trials due to timing error and differing perfusion characteristics across patients. Further differences across patients include differing organ appearance, whether innate or due to pathologies, prior surgery, or ablative therapy. All of these variabilities need to be captured by an automated segmentation model and this requires a large population sample on which to tune or evaluate the model.

Early methods relied on hand-crafted image features to interpret image contents. These methods suffered from poor generalization to new data and, frequently, a lack of spatial context in the prediction of each voxel's class. In these methods, hand-crafted image features are local statistics, computed on the intensity values of small image patches or small cuboids in a volume. The top automated methods in segmentation challenges in CT and magnetic resonance imaging (MRI) employed pixel/voxel classifiers [7–13] or region growing [14] on such features. Spatial context was added separately and after feature extraction and classification

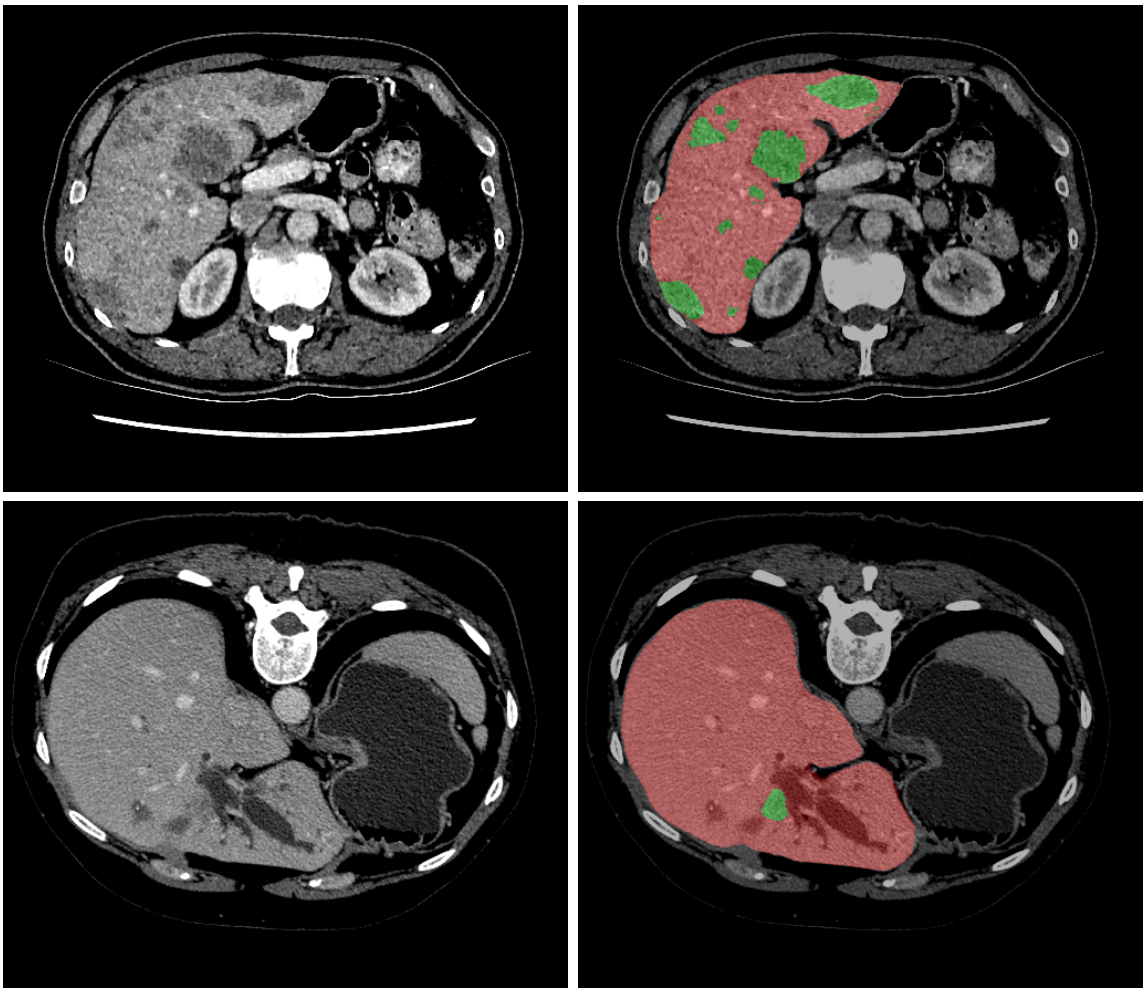


Figure 1.1 An example of manual liver (red) and lesion (green) segmentation in abdominal CT volumes with portal venous phase contrast enhancement. Sample axial slice views are shown through two volumes.

by a conditional random field (CRF) [8], a Markov random field (MRF) [9], or a deformable model [10–13]. Hand-crafted features are arbitrary and impractical to tune on large data samples that are necessary to capture the variations in the data (for example, about 20,000 chest x-ray images were required to train a robust model for pathology classification [15]). It is more practical for a model to learn the feature representations from the data.

Feature representation learning at scale has been enabled by artificial neural networks. A simple feedforward network that jointly learns representations and classification of image patches still requires a second stage model (such as a deformable model) for spatial context [16]. With enough data, much better feature representations may be learned with a convolutional neural network (CNN) which introduce some shift invariance and exploit the property of natural images where spatially local statistics dominate and tend to be scale invariant [17]. A CNN was used to match or beat the methods that placed best in the MICCAI brain tumour segmentation (BRATS) challenge 2013, while performing orders of magnitude faster [18]. This method is still patch-based; fully convolutional neural networks (FCN) [2, 3] were introduced for image segmentation that work on whole images or volumes, integrating spatial context during feature representation and pixel/voxel classification. The U-Net [19] is a successful evolution of this approach and is the basis of all state of the art medical image segmentation methods. Such methods topped the 2017 and 2018 BRATS challenges [20, 21], as well as both 2017 liver tumour segmentation (LiTS) challenges [22–27]. The LiTS challenge is introduced in Chapter 4.2 and a top entry in the challenge is proposed in Chapter 4.1. Challenges of this sort allow some principled evaluation of the state of the art and reveal where models fall short.

The results of the LiTS challenge revealed that the automatic segmentation of colorectal cancer metastases in the liver (contrast enhanced CT volumes) is poor when compared to the manual segmentation ground truth prepared by trained technicians and validated by radiologists. Furthermore, the clinical utility of the results is explored in Chapter 4.3. This analysis shows that the state of the art in automated segmentation can speed up segmentation in practice if technicians correct the automated results instead of creating their own; however, the automated results are not robust. As shown by the LiTS challenge [1], automated methods performed well on some cases and very poorly on others, suggesting that some variations in the data were poorly captured or omitted by the models. A larger training data sample is warranted but obtaining so many ground truth segmentations is impractical. This motivates the need for models that learn from additional data that lacks ground truth segmentations. To this end, a semi-supervised segmentation method that uses weak labels commonly available for medical images is proposed in Chapter 6.

Another technical consideration for deep fully convolutional networks is that of gradient propagation through all of the neural network layers. With deep networks, there is a risk of vanishing or exploding gradients due to the gain on gradient magnitudes accumulated across many layers. One common approach to avoiding exploding gradients is to softly upperbound the gain of network layers by introducing a weight norm penalty that effectively limits the spectral radius of layer weight matrices. Avoiding vanishing gradients is more challenging. The introduction of residual skip connections to feed-forward neural networks helped ensure that a gradient signal reaches all layers [28]. In Chapter 5.1, these skip connections are proposed for the U-Net segmentation model to make sure all layers are sufficiently trained. This follows a visualization and analysis of the gradient flow in these networks. Further ongoing work on handling vanishing gradients involves modifying weight initialization [29, 30], imposing norm constraints [31, 32] or spectral constraints [33–48] on weight matrices and the introduction of an attention mechanism for weight updates [49]. Spectral constraints focus on maintaining orthogonal or unitary weight matrices that preserve information but also prevent noise from vanishing. While it has been shown that orthogonality constraints are helpful not only in recurrent neural networks (RNN) but also in fully connected networks [47] and CNNs [48], these constraints may be too restrictive. In Chapter 5.2, the utility of deviating from this constraint is analyzed.

Altogether, this dissertation explores neural network methods in medical image segmentation, illuminating the clinical utility of state of the art segmentation approaches, proposes some research directions concerning gradient stability, and proposes a semi-supervised model and framework that is best adapted for many medical data.

CHAPTER 2 LITERATURE REVIEW

This section introduces the reader to the concepts and literature related to the work presented in this dissertation. The reader is assumed to be familiar with basic machine learning and feed-forward artificial neural networks trained by gradient descent backpropagation.

2.1 Early segmentation methods

Early segmentation methods relied on hand-crafted image features computed on small image patches to interpret image contents (Figure 2.1). Later methods introduced feature learning by the model from the data but are still limited in considering interactions between pixels or voxels. These methods suffer from poor generalization to new data and, frequently, a lack of spatial context in the prediction of each voxel's class.

Per-voxel patch-based classification lacks spatial context. Information beyond the patch could be integrated in a region growing method, via a deformable surface model or level set, with atlas-based methods, or by optimizing a graphical model such as a Markov random field (MRF) or conditional random field (CRF) over the voxels.

2.1.1 Region growing

Region growingn is a semi-automatic method that grows a segmentation *region* from *seed points* that are typically selected manually by an operator [50]. This is a greedy algorithm that iteratively expands the region from these seed points based on simple rules on which pixels at the border of the region absorb into the region. These rules are typically manually determined. Robustness to variabilities in the data is a challenge with region growing, making it susceptible to leaking the region growth over unwanted parts of an image. Nevertheless, this method continues to prove useful on some data [51].

2.1.2 Deformable models

Considerable focus has gone into the development of deformable models after the seminal work on active contours [52] that iteratively deform to match data from a two dimensional image, subject to geometric constraints. In a three dimensional image, a surface is deformed to wrap the object of interest in the image. In general, starting with an initial model of an object that is targeted for segmentation, a deformable surface model is deformed according

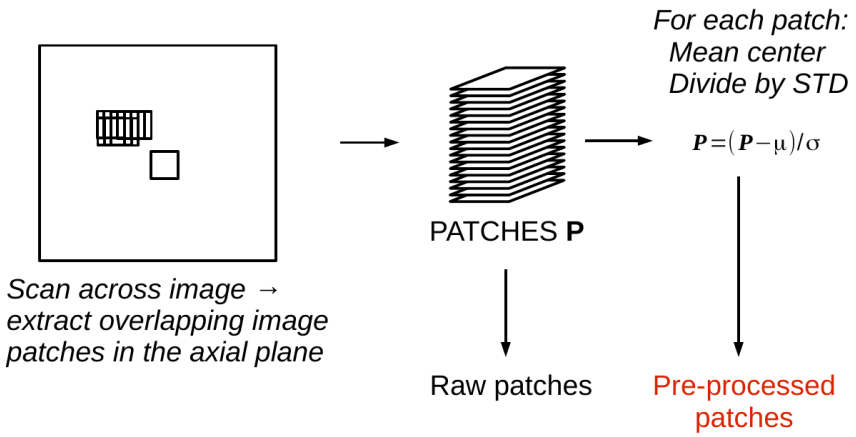


Figure 2.1 An illustration of common patch based classification where patches are first extracted with a sliding window such that there is a patch centered on each pixel that is to be classified. Then, patches are typically processed by canonical normalization which is mean centering and division by the standard deviation.

to model-specific mechanisms to match salient image features in order to produce a final representation of the object that closely matches the true object. Deformable models used for image segmentation are composed of the following three general components: (1) a representation of the object(s) being segmented; (2) rules for detecting salient features in the observed data; and (3) mechanisms for optimizing the model shape and position in response to detected features, subject to prior shape knowledge. These mechanisms involve rules for regularizing the deformation of the model form.

The rules for adapting a deformable model's surface to image data are encoded in the model's objective function. An optimization process drives the deformation of the model shape by seeking the minimum of the objective function. The objective function is composed of two general terms: a data term and a regularization term. The data term takes image data and local surface positions as input to stretch the surface toward image features. The regularization term regulates the displacement of surface positions with respect to other surface positions, stabilizing the surface deformation. Deformable models can incorporate shape statistics [53, 54] and thus serve as useful models for the segmentation of targets with known shapes such as organs [55]; however, this makes them suboptimal for segmenting less predictable targets such as lesions.

2.1.3 Level set

A level set segmentation method is an implicit approach to modeling a boundary curve or surface where the zero set of some function determines the boundary [56]. Level set methods define this function in a region based or edge based manner. Region based methods evaluate the membership of pixels based on the image features found at those positions. As they consider all pixels, they are susceptible to over-segmentation. Edge based methods are similar to deformable models in that the function determines pressure on the edge such that the edge can be moved during optimization. As with deformable models, edge based methods are sensitive to the edge initialization and may undersegment structures if artefacts block the expansion of an edge.

2.1.4 Atlas based segmentation

Atlas based image segmentation elastically deforms an atlas of expected image regions to match a target image [57, 58]. This method requires that the topography of regions that are to be mapped in an image remains consistent across all data. This makes it useful for applications such as coarse brain region segmentation but not for lesion segmentation since lesion location is unknown *a priori*.

2.1.5 Graph methods

By interpreting an image as a graph of pixels, or a deformable surface as a graph of surface nodes with possible target displacement locations [13, 59], image segmentation can be optimized via a *graph cut* algorithm.

Graphs are often considered as Markov random fields (MRF) [60] whereby each node is conditionally independent given its neighbours. This allows implicitly modeling long distance dependencies across nodes in a computationally tractable manner. In image segmentation, MRFs define a spatial prior over the pixel labels. Thus when predicting pixel labels, the prior enables the use of maximum *a posteriori* prediction instead of maximum likelihood, yielding an energy minimization problem with energy potentials for every clique in the graph. In an MRF, *unary potentials* relate to individual nodes and are conditioned on the observed image features. If all other potentials that encode relations between nodes are also conditioned on the data, then the MRF is called a conditional random field (CRF) [61, 62]. Some key limitations of MRFs are that only low order MRFs that model small cliques of node interactions are tractable (typically only pairs of nodes). They are also typically used with hand-crafted potentials.

Graph cut optimization seeks the lowest cost cut through the graph. Common graph cut optimization methods include alpha expansion and belief propagation methods. Alpha expansion [63] is a graph cuts optimization method that produces an approximately globally optimal solution for the optimization of a first order MRF (MRF with cliques of size 2: pairwise potentials). The fast primal-dual (FastPD) method [64] generalizes alpha expansion to a wider range of pairwise potentials, using the duality theory of linear programming. Belief propagation algorithms are iterative optimization methods that use message passing between graph nodes to choose node labels [65]. They place no constraints on clique potentials but may lack optimality guarantees or be computationally costly when using many labels, although priority message passing reduces this cost [66].

2.1.6 Classic methods in the wild

The winner of the 2007 MICCAI liver segmentation challenge (SLIVER07) in MRI [67] used region growing from automatically determined seed points, relying on manual image features to differentiate the liver from the rest of the abdomen [14]. While region growing introduces a greedy form of spatial dependence across voxels, it can easily fail for segmentation in images where some contents contradict the few simple hand-picked rules that the algorithm relies on. Nevertheless, recent top methods on the SLIVER07 data continue to use classic approaches such as region growing refined by a level set [51], active contour models [68], level sets [51, 69], deformable surface models [70], and multi-atlas segmentation refined by graph cuts with shape constraints [71]. Other patch-based segmentation methods in CT used a fuzzy c-means classifier [10–12] or a support vector machine classifier [13] followed by a deformable surface model which provided spatial context after classification. The top performing automated segmentation method in the MICCAI brain tumor segmentation (BRATS) challenge 2012 [72] classified voxels with a random forest and used a hierarchical CRF to add spatial context after classification [8]. Similarly, the top 2013 method used an MRF over the random forest for spatial context, as well as an additional random forest thereafter [9].

2.2 Fully convolutional networks

Convolutional neural networks (CNN) revolutionized image classification [73]. Applied to image patches, they are useful for segmentation, classifying one pixel at a time. Such methods achieved state of the art results on various segmentation challenges. DeepMedic won the Ischemic Stroke Lesion Segmentation (ISLES) 2015 challenge with a CNN applied on 3D patches [74–76]. [18, 77] improved on BRATS 2013 results with a CNN-based voxel classifier. However, these methods need to use a second stage model to integrate broad spatial context

into the prediction of pixel labels. DeepMedic used a CRF on the output predictions of the CNN, whereas [18, 77] used a cascade of CNNs, with one CNN processing the predictions of the last. This limitation of CNNs for segmentation was addressed by fully convolutional neural networks (FCN).

While CNNs are typically realized by a contracting path built from convolutional, pooling and fully connected layers, fully convolutional networks add an expanding path built with deconvolutional or unpooling layers. The expanding path recovers spatial information by merging features skipped from the various resolution levels on the contracting path. The Overfeat model introduced fully convolutional segmentation [78] where fully connected layers at the end are replaced with convolutional layers so that the CNN may be applied at multiple points in the image in a sliding-window fashion (at multiple scales). [3] introduced the first FCN that processes the entire image at once without applying a sliding window (Figure 2.2). They replaced fully connected layers in a CNN with convolution layers (using 1x1 kernels) and skipped feature maps from lower levels to the output in order to recover fine detail lost in the later, low resolution levels of a CNN. To combine feature maps of different resolutions, feature maps were upsampled via learned deconvolution layers (transposed convolution [79]), producing upsampled linear predictions. Predictions sourced from different feature map resolutions were summed together and further upsampled to produce an output segmentation of the desired resolution. In a parallel work, [2] upsampled the feature maps using simple bilinear interpolation (Figure 2.2). The FCN architecture in [3] was extended in [80, 81] by introducing additional convolutions after upsampling features and before computing and summing predictions, thus yielding top segmentation results on the 2012 ISBI EM segmentation challenge [82] for neuronal slice segmentation and the 2015 MICCAI Gland segmentation challenge [83]. DeepLab also built on the FCN concept in [3] by adapting VGGNet [84] to produce coarse segmentation predictions that are then upsampled, achieving state of the art segmentation on the PASCAL Visual Object Classes (PASCAL VOC) 2012 challenge data [85, 86]. Importantly, although the output of DeepLab is coarse (as in [3]), it uses a CRF to improve and sharpen the final segmentation output. As in [78], DeepLab eschews pooling layers with the last few convolutional layers, choosing instead to dilate the convolution kernels so as to make them perceive a wider field of view at coarser detail; however, unlike [78], it uses an efficient *atrous convolution* algorithm, also referred to as *dilated convolution* [87], to do this. The second version of DeepLab adopts ResNet [28] skip connections and spatial pyramid pooling, once again setting the state of the art on PASCAL VOC 2012 data.

The U-Net extends the FCN by modifying the expanding path to propagate wide contextual information from lower resolution feature maps to higher resolutions (Figure 2.3). At each

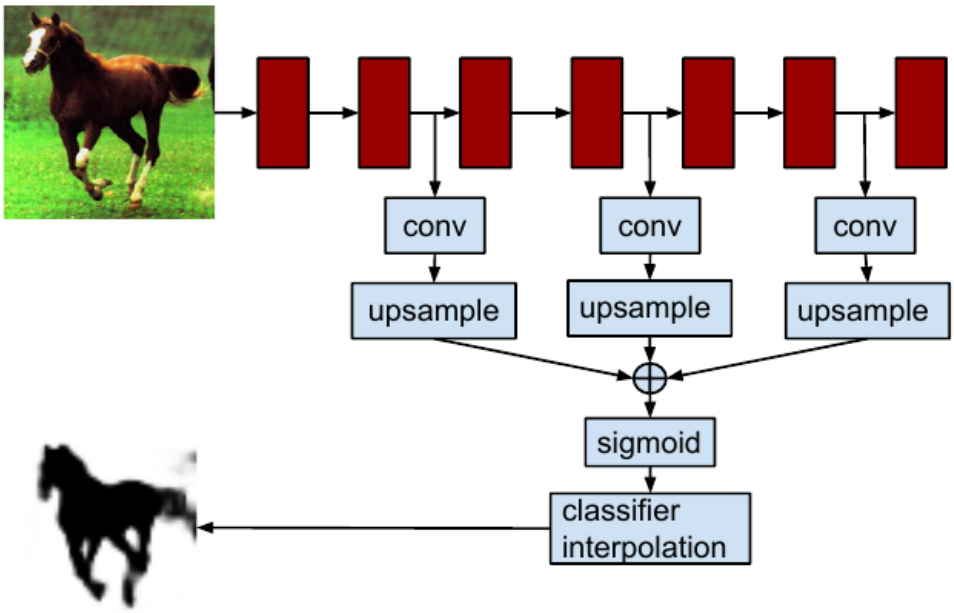


Figure 2.2 The FCN architecture of [2]. The FCN architecture in [3] is similar, differing in that the upsampling layers are transposed convolutions with learnable parameters. (Figure taken from [2].)

resolution level, upsampling is performed not simply on the feature maps from the contracting (CNN) path (as in the original FCN) but on features derived from a combination of lower resolution features and the feature maps skipped from the contracting path. That is, upsampled features are concatenated with features skipped from the contracting path and the combination is processed with convolutions before being further upsampled. In this way, the expanding path is symmetric to the contracting path.

Several similar models were developed concurrently with the U-Net and various models have been derived from the U-Net. The recombinator network introduced the same kind of model but for keypoint detection instead of image segmentation [88]. Similar to the U-Net, SegNet [89] differs in the way information is skipped from the contracting path to the expanding path. Instead of concatenating skipped feature maps like in the U-Net, pooling indices are transferred instead, allowing some spatial information to be recovered during unpooling in the expanding path. The authors claim that this has the advantage of reducing memory wasted on concatenating features. An alternative approach to achieve this memory reduction is to perform an elementwise sum between the feature maps in the expanding path and those skipped from the contracting path [90, 91]. In another model similar to the U-Net, referred to as the DeconvNet [92], skip connections are not used at all; however, this model suffers

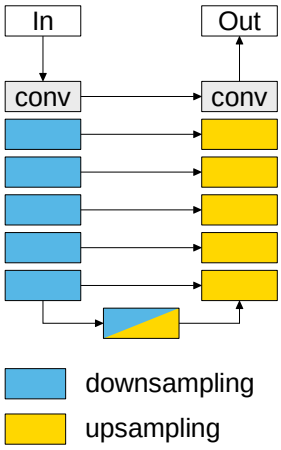


Figure 2.3 A basic sketch of an FCN based on a U-Net. Each unlabeled block contains one or more convolution and nonlinearity pairs, as well as possibly other components such as normalization layers. The first and last (labeled) blocks are convolution ("conv") blocks. Information from the encoder (downsampling, left) is passed to the decoder (upsampling, right)—typically by either summing features together or by concatenating them together.

from training difficulty and poor prediction of segmentation details [89]. By replacing 2D convolutions with 3D convolutions, the U-Net was extended for processing 3D inputs in [93] and [91] for the segmentation of kidney embryos (flourescence microscopy) and prostate (MRI). Finally, [94, 95] extended the U-Net to include learnable parameters on the long skip connections from the encoder to the decoder, yielding some improvement in segmentation results.

One potential issue with deep networks, including the U-Net, is the potential for vanishing gradients due to its depth. This issue was addressed for the original FCN in [80] by adding auxiliary loss functions at the end of each upsampling operation, in an attempt to shorten the path to a loss function from each layer. Experiments in [90] (Chapter 5.1) confirm that lower resolution features in a sufficiently deep FCN receive few to no updates. However, the network can still be trained because the skip connections between the contracting and expanding paths act as shortcuts that allow for gradient propagation from the shallow layers of the expanding path to the shallow layers of the contracting path. Unsurprisingly, the addition of batch normalization [96] at all layers of the network increases the minimum depth at which vanishing gradients become a problem and allows for the use of a much larger learning rate. Importantly, [90] and the parallel work in [91] explored the use of shorter skip connections as in ResNets [28] and confirmed that they stabilize gradient flow for very deep networks, allowing all layers to be updated. [97] won the LUng Nodule Analysis 2016

(LUNA16) challenge [98] with a U-Net augmented with such short skip connections. Similar to ResNets, DenseNets use short skip connections with concatenation instead of summation [99], achieving higher performance on image classification; [100] adapts this to a model based on the U-Net.

2.3 Structured prediction

Image segmentation is in principle structured prediction in that a model, given an image, must predict a certain structure (an output mask). Image segmentation could be achieved by classifying each pixel independently (eg. applying a CNN on small image patches to classify the voxel in the center of each patch); however, this prediction may not be robust as it cannot consider the structure of the output mask. Furthermore, although FCNs consider the entire input image upon computing a prediction, the maximum receptive field at the coarsest layer (eg. middle of the U in Figure 2.3) only covers a part of the image and predictions at the network's output are conditionally independent given the rest of the network. In other words, CNN-based models do not do complete structured prediction because they lack lateral connections between neurons in each layer. Or put more broadly, at each layer, each neuron lacks global context because it only considers a fraction of its input and its output, at each neuron, is conditionally independent of the neighbouring neurons, given the input.

A popular way of dealing with this issue is by training a conditional random field (CRF) on the output of a segmentation model. Typically, tractable CRFs model low-order (usually pairwise) interactions between neighbouring nodes in a graph, assuming that nodes are conditionally independent given their neighbourhoods (Markov property). By limiting explicitly modeled interactions to be local (within-neighbourhood) interactions, long distance dependencies between nodes must be modeled implicitly, as the combined result of all local interactions. This prohibits the modeling of rich long distance dependencies. To alleviate this issue in segmentation, instead of associating each image pixel with a node, pixels are typically clustered into irregularly shaped superpixels (each associated with a node), thus extending the reach of local neighbourhoods across an image. Recently, [101] introduced a CRF approach that approximates full pair-wise connectivity – that is, every node is connected to every other node. They apply a mean field inference approach, modeling pairwise interactions between pixels by a linear combination of Gaussian kernels. This work was used by [85] to clean up and sharpen the low resolution segmentation output of a CNN, producing quality segmentations on the PASCAL VOC 2012 segmentation benchmark. [102] report successful liver segmentation by passing the output of a U-Net through this fully connected CRF. The DeepLab model versions 1 and 2 both set the state of the art on the PACAL VOC

2012 data, using a fully connected CRF to improve segmentation output [85, 103]. Also, the winner of the ISLES 2015 challenge used a fully connected CRF to produce a segmentation from the voxel labels proposed by a CNN applied on 3D patches. A couple shortcomings of the above approaches are that the unary potentials of the CRF are learned by the CNN or FCN without anticipating a CRF in the neural network's objective and that the pairwise potentials are hand tuned. [104] model the fully connected CRF as a recurrent neural network and integrate it with an FCN as a single model, trained end-to-end. They refer to this approach as CRFasRNN. One step of the RNN modeling of mean field inference is essentially comprised of full-image Gaussian convolutions on a softmax output, followed by 1x1 kernel convolutions, followed by a softmax output. Because the FCN and RNN are trained end-to-end and because no assumption is made on pairwise potentials, this model resolves the aforementioned shortcomings.

A couple other ways to incorporate global context in CNN based encoder-decoder models have also been explored. In DeepLab version 3, global context is added to the decoder of an FCN by using dilated convolution at multiple scales, using spatial pyramid pooling [105]. This allows the decoder to consider pixel interactions across many scales during inference. This approach could be extended to every layer in the decoder of a U-Net. Another approach that was found to be useful for image generation is a self-attention mechanism [106] based on non-local neural networks [107]. In this formulation, an attention map over feature maps is constructed from the product of the features (after 1x1 convolution) and their transpose along spatial dimensions (after 1x1 convlution). This attention map thus gains some long distance spatial dependence; however, while this method is computationally cheap, the spatial dependence structure it imposes is arbitrary and relatively simple. It may be fruitful to search for more tricks to introduce global context to FCN decoders.

2.4 Recurrent neural networks for image segmentation

Besides convolutional neural networks, there has been some progress in semantic segmentation using recurrent neural networks (RNNs). [108] introduced the multi-dimensional RNN (MDRNN) and tested it on the segmentation of synthetic texture images. The MDRNN processes an input image from the corners, taking at each step two inputs. For example, for an MDRNN propagating from the top-left corner of an image, a pixel takes as input the RNN output at the adjacent pixel on top and the RNN output at the adjacent pixel on the left (as shown in Figure 2.4, panel (a)). [109] stacked multiple MDRNN layers, separated by fully connected layers, achieving state of the art segmentation results on the Stanford Background dataset and the SIFT Flow dataset in 2015. A major downside of this model

is that it is hardly parallelizable. Recently, [4] devised a processing trick to tackle this issue in their PyraMiD-LSTM model, greatly improving the parallelizability of multidimensional RNNs. They rotated the pixel-to-pixel connections by 45 degrees thus ensuring that all input connections come from a single edge of the image (detailed in Figure 2.4 panels (b) and (c)). This allows pixels in each row and column of an input image to be processed by a convolutional operator; thus, convolutions are applied sequentially one column at a time or one row at a time. A major advantage of the PyraMiD-LSTM over CNNs is that it can model dependencies across any pixels at any distance, lending the model global context upon prediction. However, the PyraMiD-LSTM suffers from extremely high memory usage, limiting processing on a 12GB GPU to tiny 64x64x64 volumes, much smaller than what could be handled by an FCN.

Claiming easier parallelizability, [110] introduced ReNet, a reinvention of the seminal convolutional neural network LeNet [111]. In place of convolution and pooling operations, four RNNs sweep the image from left to right, right to left, top to bottom, and bottom to top. These RNNs sweep the image in sequence, consuming as input the output of the previous RNN. By stacking multiple such layers, and reducing the size of each layer's output with respect to its input, ReNet produces a high level compact representation of the input image, useful for image classification. [112] then extended this approach to segmentation (ReSeg) by appending a ReNet to a pretrained CNN and then upsampling the output of the ReNet via transposed convolution. While this allows global context to be considered within the ReNet segment of the network, this processing is done at a low resolution, further reduces the resolution, and makes use of few transposed convolutions to upsample the output. In practice, the use of ReNet is restricted to a single low resolution segment due to the high computational cost of using ReNet layers; this unfortunately limits the utility of ReNet in a segmentation pipeline.

2.5 Semi-supervised segmentation

Few semi-supervised methods have been proposed to date for image segmentation, particularly in the medical image domain. Some early methods used weak labels such as bounding boxes [113, 114] or image level labels [114, 115] in place of full pixel-level segmentation labels. However, bounding boxes may be difficult to collect for medical image data as these require expert detection and the methods that use image level labels require a dataset to have impractically many types of labels, making them unsuitable for medical image datasets.

[116] developed a semi-supervised segmentation method with the unsupervised objective of matching per-pixel learnt embeddings between similar cases. However, the similarity of

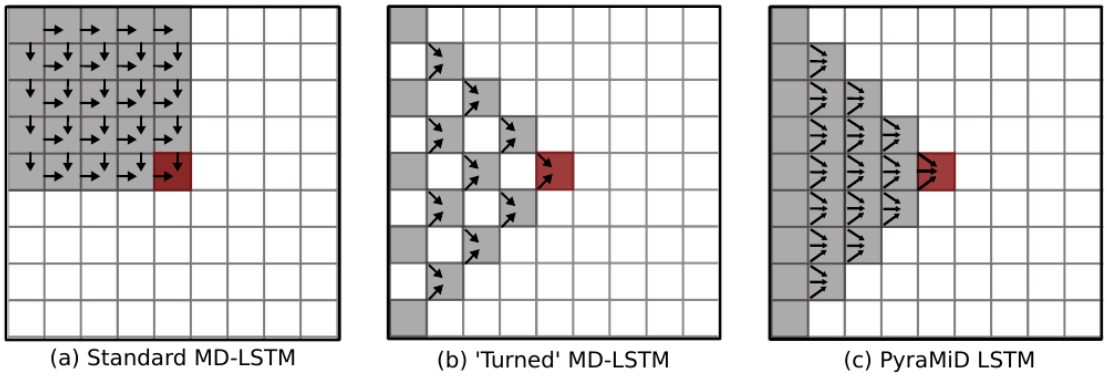


Figure 2.4 MDRNN signal propagation as shown in [4]. (a) In the standard MDRNN (here MD-LSTM), inputs at a pixel position are taken from two orthogonal edges. (b) [4] propose to rotate the connections by 45 degrees. (c) To avoid holes, additional non-diagonal connections are added.

cases is determined by an arbitrary similarity metric involving template matching between cases. Although this method was applied to brain MRI data, there were very few cases and this data is private, limiting interpretability of the results. [117] adapted the mean teacher method to fully convolutional networks for segmentation. In this formulation, the teacher network is an exponential moving average of weights from the student network and the student network seeks to learn to segment (supervised) and to be consistent with the teacher network's predictions (unsupervised).

[118] used an adversarial approach where the pixel classifer for PASCAL VOC segmentation is also a discriminator that learns to discriminate between ground truth and generated segmentations. Furthermore, the segmentation generator is conditioned on image-level labels, allowing for some weak supervision. Similarly, [119] and [120] developed adversarial semi-supervised segmentation methods that rely on discrimination between ground truth and generated segmentations. These adversarial methods may not scale well with the proportion of unannotated data since the quality of the discriminator depends on the available number of ground truth segmentations; the discriminator may not generalize well beyond the annotated dataset on which it is trained.

Other adversarial and generative approaches mainly focused on learning lesion localization, given a set of cases known to be healthy and another set of cases known to contain lesions. To roughly localize lesions in brain MRI, [121] fitted the healthy data distribution with an autoencoder. Given an image presenting a lesion, the lesion is localized via the residual of its reconstructed image which is likely to appear healthy. Similarly, [122] and [123] employed a generative adversarial network (GAN) to locate anomalies in retinal images and brain MR

images, respectively. These models are unsupervised, provide only rough localization, and only use the healthy cases for training. Other unsupervised adversarial methods that use both healthy and sick cases were based on image-to-image translation. By translating from sick to healthy images, [124] trains a network to localize Alzheimer's derived brain morphological changes using the output residual. [125] further proposes a multi-modal variant of CycleGAN [126] to translate in both directions, applied to brain MR images with cancer. Sick images that are translated to healthy images are translated back to the original sick image via a residual inpainting of the lesions. Lesions are localized and segmented by predicting a minimal region to which to apply inpainting. Segmentation is unsupervised, with a prior that minimizes the inpainting region. This method has not been compared to other unsupervised methods, has been tested on a single dataset, and has not been extended to a weakly- or semi-supervised setting.

2.6 Vanishing and exploding gradients

Deep neural networks trained by gradient backpropagation can suffer from vanishing or exploding gradients [127]. After passing through multiple neural network layers that shrink the same dimensions of their inputs, gradients can vanish. Similarly, gradients can become impractically large to compute when they are repeatedly expanded along some dimensions. This is best exemplified in a simple recurrent neural network (RNN) model. For the purpose of the following discussion, an RNN may be considered "deep" when it is applied to a long sequence with many time steps. If the computation is unrolled across time, one gets a deep neural network with the same number of consecutive layers as there are time steps. Thus, other deep feedforward neural networks are analogous to the RNN, except that the weights of all the layers are not shared as they are in the RNN.

An RNN is stable if small perturbations of the initial state do not diverge to large changes in the state evolved over time. From the perspective of ordinary differential equations (as in Eq. 2.2):

Definition 2.6.1. (**Stability**) A solution $\mathbf{h}(t)$ of Eq. 2.2 is stable if for every small $\epsilon > 0$ there exists a $\delta > 0$ such that any other solution $\hat{\mathbf{h}}(t)$ that starts as $|\mathbf{h}(0) - \hat{\mathbf{h}}(0)| < \delta$ remains in $|\mathbf{h}(t) - \hat{\mathbf{h}}(t)| < \epsilon$.

Stability is satisfied when the RNN dynamics are limited to stable fixed point attractors and limit cycles (orbital trajectories); however, this regime can cause catastrophic forgetting and vanishing gradients. Vanishing and exploding gradients in RNNs can be illuminated by looking at the gradient back-propagation chain. A network with T time steps has state

pre-activations

$$\mathbf{a}_t(\mathbf{h}_{t-1}) = \mathbf{W}\ \mathbf{h}_{t-1} + \ \mathbf{b}, \qquad t \in \{2, \cdots, T-1\}, \tag{2.1}$$

where $\mathbf{h}_{t-1}$ is the state state at time $t-1$, $\mathbf{W}$ is the state transition matrix, and $\mathbf{b}$ are the activation biases. For convenience, consider $\mathbf{W}$ and $\mathbf{b}$ combined in an affine matrix $\boldsymbol{\theta}$. For a loss function L at time T, the derivative with respect to parameters $\boldsymbol{\theta}$ at time t is $\frac{\partial L}{\partial \boldsymbol{\theta}}(t) = \frac{\partial \mathbf{a}_t}{\partial \boldsymbol{\theta}} \prod_{i=t}^{T} (\mathbf{D}_i \mathbf{W}) \frac{\partial L}{\partial \mathbf{a}_{n+1}}$ where $\mathbf{D}$ is the Jacobian of the activation function. This shows that the backpropagated gradient can grow or shrink exponentially across each time step depending on the gain of $\mathbf{W}$ and $\mathbf{D}_i$. For *tanh* or *ReLU* activation functions, the Jacobian is diagonal with singular values in the interval $[0, 1]$. So exploding gradients are caused only by large singular values in $\mathbf{W}$ and vanishing gradients are caused by small singular values in $\mathbf{W}$ or by the activation function. Exploding gradients are often avoided by clipping large gradients [32] or introducing an L_2 or L_1 weight norm penalty.

Vanishing and exploding gradients can be avoided by maintaining constant gradient norm over time. [32] propose penalizing differences in successive gradient norm pairs in the backward pass and [31] penalize norm changes in the forward pass. Also, a norm-preserving linear operator across RNN state transitions can prevent exploding gradients and reduce vanishing gradients. Orthogonal matrices are norm-preserving. With all singular values equal to 1, their gain is unity in each dimension for any input. However, singular values less than 1 are necessary in order to store information reliably [32, 128].

2.6.1 Skip connections

Recently, there has been extensive progress in training very deep neural networks. [129] hypothesized that very deep networks could not be trained because it is difficult for layers to learn an identity mapping and layers instead tend to degrade the signal passing through them. To maintain the signal, they introduce shortcut connections from the input of a layer to the layer's output [28, 129] (Figure 2.5), acting as an identity mapping that is capable of bringing a representation from any part of the network directly to the network's output. In effect, these skip connections provide a direct path from any feature representation to the loss. Furthermore, by adding a layer's input to its output, the layer is forced to learn residual transformations which could be interpreted as corrections of the input. Due to these residuals, networks with these skip connections are referred to as residual networks or ResNets. These skip connections stabilize training, allowing the use of very high learning rates with stochastic gradient descent (SGD) and permit the training of very deep networks—[28] trained a 1001 layer ResNet on CIFAR-10 and CIFAR-100 image classification. Importantly, [28] show that

very deep networks with few parameters can perform as well as shallow networks with many parameters. On the other hand, [130] present wide residual networks, where they show that wide relatively shallow networks can achieve the state of the art on CIFAR image classification and even a 16 layer network with sufficiently many parameters can outperform the previously proposed deep, narrow networks on this dataset.

Various other works extending ResNets achieved marginal performance improvements on CIFAR, including residual networks of residual networks [131], which introduce additional longer distance skip connections, and DenseNet [99] which employs a dense skip connectivity and uses feature map concatenation instead of summation. Earlier, [132] analyzed similar variations in their exploration of the recurrent nature of ResNets. They note that a ResNet is nearly equivalent to a simple RNN that takes a single input, differing in that weights are not shared across layers. Thus, they compare the use of unshared weights to the use of shared weights and further explore different connectivity patterns (recurrence structures). They find that a ResNet with shared weights performs almost as well as a ResNet with unshared weights while keeping far fewer parameters. The best performing network in their analysis has a fully connected recurrence structure where features are skipped not only from the input of each layer to its output but also to the output of every subsequent and of every previous layer (ie. the network is fully connected through time).

[133] proposed highway networks shortly before residual networks were published. These also allow training deep networks but introduce gating to both the skip connection and the residual. The authors of ResNets claim [28] that this gating may cause vanishing gradients and does not allow the training of networks as deep as those trained with gateless identity skip connections.

There has been growing evidence that ResNets perform a form of iterative inference. [134] offer the interesting perspective that ResNets behave like "exponential ensembles of relatively shallow networks". They note that while removing a layer in a VGG network results in nearly 90% error, removing a layer in a ResNet barely reduces its performance. Indeed,

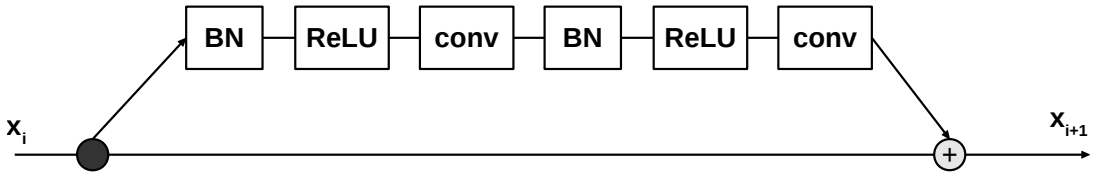


Figure 2.5 An input x_i is added back into output x_{i+1}, the later produced by passing the input through a series of batch normalization layers (BN), rectified linear units (ReLU), and convolutions (conv).

[135] introduce a regularization method for ResNets that randomly drops residuals. Even more surprisingly, [134] show that switching the order of layers at test time only slightly reduces the model's performance. This suggests that residuals in ResNets tend to learn very similar transformations. [136] exploit this ensemble view of ResNets by adding multiple parallel residuals in each layer, increasing the multiplicity of paths for signal propagation. Finally [137] argue that highway networks and ResNets perform unrolled iterative estimation, iteratively refining a representation. They show that these networks tend to learn residuals that minimize their average estimation error. This is intuitive because at each point in a ResNet, the input to a layer is the sum of all previous outputs and thus the input to the network's classifier is the sum of all of the network's residuals.

[137] highlight the idea that ResNets and highway networks do not follow the traditional pattern of learning a hierarchical representation. Because these networks tend to not alter their inputs much, it may be important to consider the use of traditional deep networks without skip connections as preprocessing networks, as suggested by [132]. Currently, only a single convolution layer is typically used for preprocessing. This highlights the need for ways to better optimize deep networks without ResNet skip connections.

2.6.2 Orthogonal weight matrices

Managing long term dependencies and taming vanishing and exploding gradients in recurrent neural networks (RNNs) is a topic of continued interest. One direction in this topic has been the use of an orthogonality constraint on the state transition matrix. Since orthogonal operators are norm-preserving, this stabilizes signals that flow through the network. When using orthogonal state transition matrices in a linear network, vanishing and exploding gradients are eliminated; with non-linear activation functions, vanishing gradients are reduced and exploding gradients are eliminated. Orthogonal initialization and a soft orthogonality constraint were shown to be helpful for toy tasks that require long term memory [29, 30, 33]. Networks with transition matrices constrained to be orthogonal or unitary yielded good performance on these and similar tasks [34–43]. However, this constraint appears to be too rigid.

Many recent methods have constrained transition matrices to be norm-preserving. [30] and [33] have shown that orthogonal initialization (including identity) is beneficial. [33] and [39] have used a soft penalty term to encourage orthogonality throughout training. [34] maintain a strictly unitary complex-valued transition matrix by construction, building it as a composition of multiple simple unitary operators. Such an arbitrary composition of operators covers only a subset of possible unitary matrices, leading [35] and [39] to use a computation-

ally costly gradient retraction approach via the Cayley transform to get unitary or orthogonal matrices, respectively. Additionally, [39] show that deviating from orthogonality allows faster convergence and better model performance even on tasks with long term dependencies. [40] point out that the Cayley approach cannot represent matrices with negative one eigenvalues and proposed a scaling trick to remedy this; however, since scaling terms are either positive or negative one, the number of negative one eigenvalues is fixed at model construction. [41] overcome this limitation by doing the same in the complex space, where scaling terms can be smoothly learned as phasors with modulus one. [36] use an expensive matrix exponential mapping to maintain orthogonality. However, [42] recently noted that Padé approximations to the matrix exponential are actually fast and effectively exact when computed to machine precision. Importantly, any error does not result in the transition matrix not being orthogonal. [38] propose a novel matrix composition inspired by the fast Fourrier transform (FFT) that covers some subset of possible orthogonal matrices and is theoretically computationally cheap but hard to parallelize. They also propose an alternate method that uses a composition of rotation matrices. Similarly, [37] use a composition of Householder reflections. [43] employ the FFT-like method to create a gated recurrent unit (GRU) with an orthogonal transition matrix. While these methods avoid vanishing and exploding gradients when the state transition operator is linear, nonlinear activation functions still contribute to vanishing gradients. Recently, [44] overcame this by applying stability criteria for an ordinary differential equation view of RNNs (as in Definition 2.6), relying on antisymmetric but not orthogonal transition matrices.

2.6.3 Dynamics

The recurrent application of a state transition operator in an RNN to inputs over many time steps yields a dynamical system. The dynamics are then important to consider both for model performance and gradient backpropagation during training. Manifesting as fixed points, limit cycles, or chaos, RNN dynamics affect the trajectories along which a state can evolve.

2.6.4 Fixed points

For an RNN transition operator T acting on state $\mathbf{h}$, a fixed point occurs where $T(\mathbf{h}) = \mathbf{h}$. The neighbourhood about a fixed point may attract states to the point or repel them from it. Neighbourhoods that attract along all dimensions define basins of attraction for fixed point attractors. Those that repel, contain a repellor. Those that attract along some dimensions and repel along others contain a saddle point. Saddle points appear to mediate the evolution

of state trajectories over time toward different fixed point attractors [138].

If all singular values are less than 1, then there is only a single fixed point at zero. If some singular values are less than 1, then a linear transition operator T has an n-dimensional subspace of fixed points where n is the number of singular values equal to 1. Multiple fixed point attractors are possible with the addition of a nonlinear activation function, such as *tanh* or *sigmoid*. These impart a contractive effect that may need to be be overcome by singular values greater than 1. An RNN with multiple fixed points may be driven by inputs to shift its state between fixed point attractors like a state machine.

Fixed point attractors in biologic neural networks have long been considered in computational neuroscience literature as a mechanism for latching memory. In RNNs, this role of fixed point attractors was analyzed and contrasted with them causing vanishing gradients [32, 128, 139, 140]. This is because stable fixed point attractors require singular values less than 1 in the transition matrix, which makes gradients exponentially decay along these dimensions and possibly effectively vanish.

The issue of vanishing gradients is reduced by introducing a constant additive memory path in LSTM and GRU. This path, via a gating mechanism, skips information forward over many time steps and likewise skips the gradient backward. This memory mechanism is conceptually similar to a fixed point but does not require dynamics that cause vanishing gradients, unlike stable fixed point attractors. For this reason, it is thought that such gated networks may not require fixed point attractors [141]. However, [142] improved the performance of a GRU by adding a pretrained attractor network to clean up its state at every state transition. Thus, the GRU was trained to make use of this network, augmenting the state dynamics over time. The attractor network itself is simply a recurrent denoising autoencoder that is trained, for a set of n random inputs, to learn n fixed point attractors by learning to denoise the inputs. The improvement of GRU performance when fixed point attractors are added to the state transition dynamics suggests that gated memory does not obviate the utility of traditional RNN dynamics in GRU and LSTM.

To better understand the role of fixed points in RNNs, it is useful to find these points in trained networks, given real data. [138] propose linear conditions near fixed points for empirically finding fixed and slow points in a trained RNN. Considering an RNN state transition as a system of first order differential equations

$$\dot{\mathbf{h}} = F(\mathbf{h}), \tag{2.2}$$

points of slow (or zero) change are found by minimizing

$$q(\mathbf{h}) = \frac{1}{2} \left| F(\mathbf{h}) \right|^2 , \tag{2.3}$$

where F defines the transition operator for the state $\mathbf{h}$. To identify fixed point attractors, repellors, and saddles, it is then sufficient to evaluate a linearization about $\mathbf{h}$ by considering the spectrum of the Jacobian of F. Beyond fixed points, minimizing q also reveals slow points where $\mathbf{h}$ changes slowly along some dimensions. By plotting state trajectories with different initial conditions, [138] show that both saddles and slow points appear to have important roles in RNNs, mediating convergence to different fixed point attractors. Plotting the trajectories also revealed different interesting dynamics that make use of fixed points for solving different tasks. A moving average task yielded a plane attractor along which the inputs to be averaged together are latched and shifted. A flip flop task yielded stable fixed point states with saddle points mediating input-specific transitions between these states. Interestingly, a sine wave generation task yielded fixed points that were origins for orbital oscillations, each corresponding to a desired wave frequency. This suggests that fixed point dynamics are useful for more than latching memory in RNNs.

2.6.5 Chaos

An RNN state transition exhibits chaotic behaviour when the transition operator is sufficiently expansive so that trajectories move away from each other. The rate at which trajectories diverge is measured by Lyapunov exponents, where positive exponents indicate divergence and negative exponents indicate convergence to stable fixed points and limit cycles. In a linear transition operator, expansion along some dimensions requires singular values greater than 1. This expansion can lead to the unbounded growth of the state norm during inference or the gradient norm during gradient backpropagation, resulting in exploding gradients. Thus, neural network initializations tend to produce singular values up to about 1, resulting in a weight initialization just on the *edge of chaos* [29]. Just beyond, the chaotic regime gives rise to strange attractors that induce fractal trajectories. These trajectories follow an attractor-specific structure but never exactly repeat. Unlike with non-chaotic attractors, the trajectory is input-dependent.

Interestingly, RNNs like the LSTM and GRU that learn non-chaotic dynamics, tend to be chaotic in the absence of inputs. To avoid this, [143] design a chaos-free RNN and show that such an RNN can perform almost as well as an LSTM with simple dynamics, suggesting that chaos may not be necessary for good performance. On the other hand, [144] show that a strongly chaotic RNN significantly outperforms one that is initialized on the edge of chaos for

sequence classification. The sensitivity of chaotic trajectories to the input appears to allow the model to be more discriminative with fewer parameters. Even an untrained RNN that is initialized as chaotic allows the classifier to learn classification with perfect accuracy whereas an untrained RNN initialized at the edge of chaos yields poor classification and, once trained, only approaches perfect accuracy. This motivates the utility of chaos for RNN performance. Although it is simpler to analyze models restricted to fixed point dynamics, chaos may be common in biologic neural networks [145].

2.7 Adversarial learning

Generative adversarial networks (GAN) [146] are neural networks that are trained against each other in an adversarial manner. This formulation provides interesting new modeling opportunities with deep neural networks. Briefly, training involves a *generator* network and a *discriminator* network, where the discriminator continually learns to identify whether its input is generated or is a real data sample. In this way, the generator learns to generate data within the real data distribution. The generator is commonly mapping samples from a simple prior distribution (eg. Gaussian or uniform noise) to the data; thus, the generator learns to produce and interpolate between known data. By training a GAN on image and segmentation pairs, new pairs can be generated for data augmentation during the training of a segmentation network. GANs are used to augment liver lesion examples for classification in [147]. [148] synthesize data to cover uncommon cases such as peripheral nodules touching the lung boundary. [149] and [150] introduce a segmentation mask generator to augment small training datasets.

The input to a generator is, however, arbitrary. Given image inputs, a generator may learn to translate images from one domain to another, such as horses to zebras [126]. Such image-to-image translation was most prominently done with the CycleGAN [126] which does bidirectional translation between two domains. UNIT [151] proposed a similar approach but with a common latent space, shared by both domains, from which latent codes could be sampled. Augmented CycleGAN [152] and Multimodal UNIT [153] respectively extended both methods from one-to-one mappings to many-to-many.

Another powerful application of GANs is in the learning of an objective for training. For example, instead of defining by hand a segmentation objective function that matches a segmentation generator's prediction to the ground truth, a discriminator could be trained, instead. The discriminator learns such a function from the data. Recently, such adversarial training has been used to improve segmentation results on medical images [154–163].

CHAPTER 3 OBJECTIVES AND CONTRIBUTIONS

This dissertation is mainly concerned with the improvement of automated segmentation in the medical image domain, focusing on metastatic lesions in the liver in computed tomography. Toward that end, the groundwork is laid to evaluate and establish the state of the art in this domain, model improvements are explored, and the issue of high data variability and insufficient labels is addressed in a novel way. It was while considering model improvements that the common issue of vanishing gradients was investigated for the deep neural network models that are the state of the art in image segmentation. This investigation then prompted more academic study on managing long term dependencies in data when using gradient descent—a topic concerned with the elimination of vanishing gradients in deep neural network training by error backpropagation. Consequently, the contributions presented in this dissertation were driven by the following **objectives**:

1. Establish the state of the art for liver tumour segmentation.
2. Address the issue of vanishing gradients.
3. Overcome the issue of insufficient segmentation labels.

3.1 Structure of the dissertation

The specific contributions of this dissertation are summarized below, along with associated publications, for each chapter. The contributions and their relations to the objectives are illustrated in Figure 3.1.

Chapter 4

Overall contribution

Establishing and evaluating the state of the art of liver tumour segmentation.

Specific contributions

The liver tumour segmentation (LiTS) challenge is set up, allowing the state of the art to be established for colorectal metastatic tumor segmentation in the liver in CT. Automated segmentation is found to be poor and not robust. A top performing entry is presented based on a fully convolutional network. Further evaluation is performed on whether a state of the art LiTS method can be used to accelerate segmentation or improve inter-rater variability

when corrected by experts. The intra- and inter-rater variability are evaluated. Manual correction of automated segmentation is found to significantly accelerate segmentation in practice, achieving near-manual quality and reduced inter-rater variability.

Publications included

E. Vorontsov, A. Tang, C. Pal, and S. Kadoury, "Liver lesion segmentation informed by joint liver segmentation," in *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*. IEEE, 2018, pp. 1332–1335

E. Vorontsov, M. Cerny, P. Régnier, L. Di Jorio, C. J. Pal, R. Lapointe, F. Vandenbroucke-Menu, S. Turcotte, S. Kadoury, and A. Tang, "Deep learning for automated segmentation of liver lesions at ct in patients with colorectal cancer liver metastases," *Radiology: Artificial Intelligence*, vol. 1, no. 2, p. 180014, 2019

Chapter 5

Overall contribution

Evaluating skip connections and orthogonality.

Specific contributions

The flow of gradients through a deep model based on the U-Net is analyzed. In order to avoid vanishing gradients, the addition short skip connections [28] and batch normalization [96] is suggested. Following along this vein of research, a novel analysis is performed on the use of an orthogonality constraint on neural network weight matrices. For simplicity, this initial analysis is performed in an RNN model. The utility of deviating from this constraint is demonstrated. Deviating slightly from orthogonality in the spectrum of weight matrices allows for faster convergence and better model performance.

Publications included

E. Vorontsov, C. Trabelsi, S. Kadoury, and C. Pal, "On orthogonality and learning recurrent networks with long term dependencies," in *Proceedings of the 34th International Conference on Machine Learning*. JMLR.org, 2017, vol. 70, pp. 3570–3578

Chapter 6

Overall contribution

Overcoming the lack of segmentation ground truth in medical imaging.

Specific contributions

Automated segmentation in medical imaging may be made more robust by training models on a larger sample of data that is more representative of the variations in the population. However, creating so many ground truth segmentations is impractical, so a semi-supervised method is proposed. Since the presence or absence of pathologies is often known in medical images, these are used as weak labels for the model. Given any input, the presented model learns to output both *sick* and *healthy* variants. By learning to translate between these two domains, the model learns to disentangle the same variations required for segmentation, thus making effective use of data that lacks ground truth segmentations.

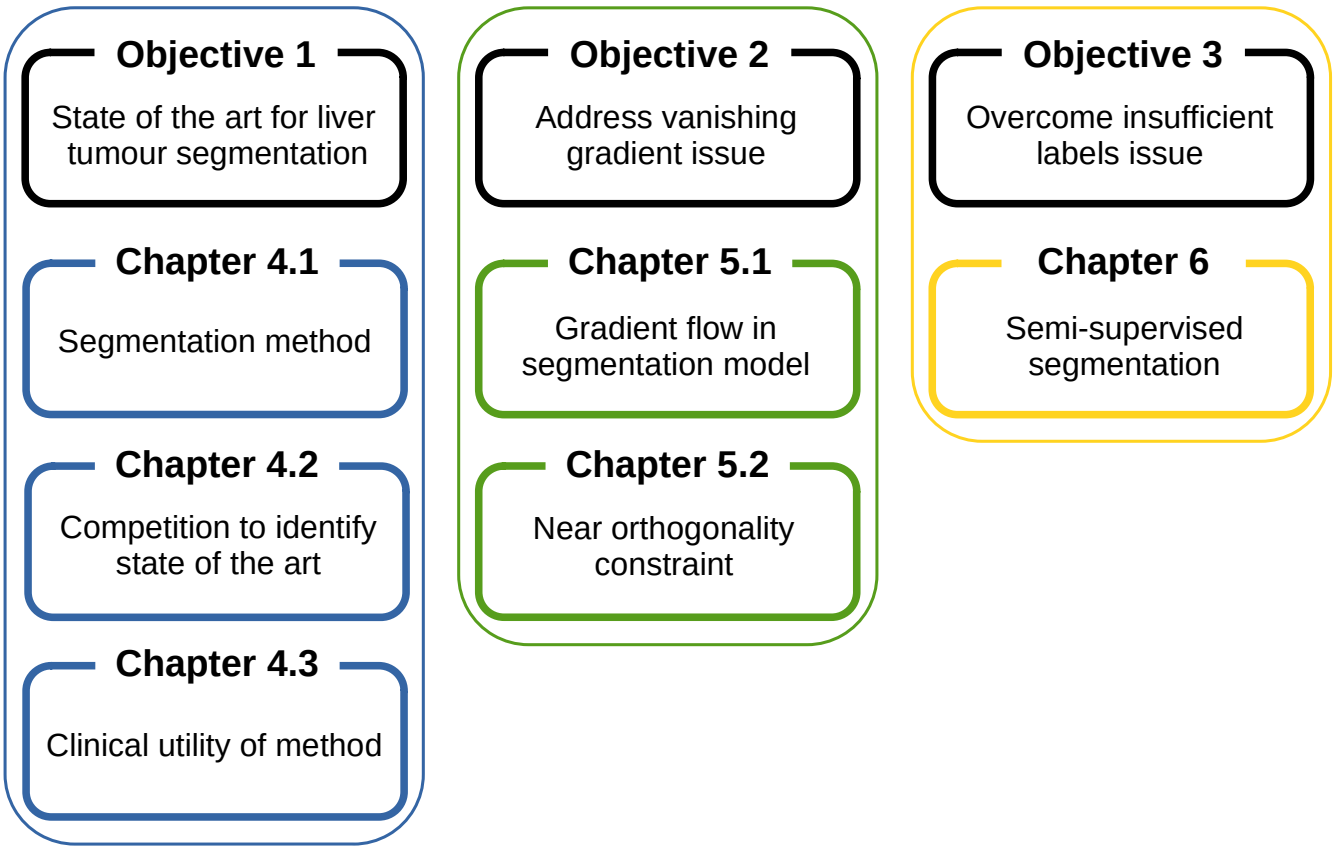


Figure 3.1 The contributions of this dissertation are shown associated with the objectives.

3.2 Other related publications

The following publications were produced as a result of the work presented in this dissertation but are either not reproduced in full or are omitted completely:

E. Vorontsov, N. Abi-Jaoudeh, and S. Kadoury, "Metastatic liver tumor segmentation using texture-based omni-directional deformable surface models," in *International MICCAI Workshop on Computational and Clinical Challenges in Abdominal Imaging*. Springer, 2014, pp. 74–83

M. Drozdzal, E. Vorontsov, G. Chartrand, S. Kadoury, and C. Pal, "The importance of skip connections in biomedical image segmentation," in *Deep Learning and Data Labeling for Medical Applications*. Springer, 2016, pp. 179–187

E. Vorontsov, A. Tang, D. Roy, C. J. Pal, and S. Kadoury, "Metastatic liver tumour segmentation with a neural network-guided 3d deformable model," in *Medical & biological engineering & computing*, vol. 55, no. 1, pp. 127–139, 2017

G. Chartrand, P. M. Cheng, E. Vorontsov, M. Drozdzal, S. Turcotte, C. J. Pal, S. Kadoury, and A. Tang, "Deep learning: a primer for radiologists," in *Radiographics*, vol. 37, no. 7, pp. 2113–2131, 2017

M. Drozdzal, G. Chartrand, E. Vorontsov, M. Shakeri, L. Di Jorio, A. Tang, A. Romero, Y. Bengio, C. Pal, and S. Kadoury, "Learning normalized inputs for iterative estimation in medical image segmentation," in *Medical image analysis*, vol. 44, pp. 1–13, 2018

S. Chandar, C. Sankar, E. Vorontsov, S. E. Kahou, and Y. Bengio, "Towards non-saturating recurrent units for modelling long-term dependencies," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 3280–3287

G. Kerg, K. Goyette, M. P. Touzel, G. Gidel, E. Vorontsov, Y. Bengio, and G. Lajoie, "Non-normal recurrent neural network (nnrnn): learning long time dependencies while improving expressivity with transient dynamics," in *Advances in Neural Information Processing Systems*, 2019, pp. 13 591–13 601

P. Bilic, P. F. Christ, E. Vorontsov, G. Chlebus, H. Chen, Q. Dou, C.-W. Fu, X. Han, P.-A. Heng, J. Hesser et al., "The liver tumor segmentation benchmark (lits)," *arXiv preprint arXiv:1901.04056*, 2019

A. L. Simpson, M. Antonelli, S. Bakas, M. Bilello, K. Farahani, B. Van Ginneken, A. Kopp-Schneider, B. A. Landman, G. Litjens, B. Menze et al., "A large annotated medical image dataset for the development and evaluation of segmentation algorithms," *arXiv preprint arXiv:1902.09063*, 2019

CHAPTER 4 AUTOMATED SEGMENTATION IN COMPUTED TOMOGRAPHY OF COLORECTAL METASTASES IN THE LIVER

This chapter covers work on the segmentation of lesions in the liver in computed tomography images. Specifically, a relevant segmentation challenge is briefly presented along with a top entry into that challenge; then, the clinical utility of the proposed automated segmentation is evaluated. This analysis shows that automated segmentation in this domain is poor but can significantly speed up the manual segmentation workflow if operators correct the automated results instead of segmenting from scratch.

4.1 (Article 1) Liver lesion segmentation informed by joint liver segmentation

This paper describes my entry in the LiTS challenge at MICCAI 2017, where it ranked among the top methods. It is a minor refinement of my previous entry in LiTS at ISBI 2017 [23].

Title

Liver lesion segmentation informed by joint liver segmentation

Authors

Eugene Vorontsov[1,2], An Tang[3], Chris Pal[1,2], Samuel Kadoury[1,3]

Affiliations

[1] École Polytechnique de Montréal

[2] Montreal Institute for Learning Algorithms (MILA)

[3] Centre de Recherche du Centre hospitalier de l'Université de Montréal (CRCHUM)

Publication

This paper has been published in the *IEEE transactions on the 15th International Symposium on Biomedical Imaging (ISBI 2018)* in 2018, pp 1332-1335 [164].

Contribution

My contributions in this work cover all the ideas, data processing, coding, planning, experiments, and writing.

4.1.1 Abstract

We propose a model for the joint segmentation of the liver and liver lesions in computed tomography (CT) volumes. We build the model from two fully convolutional networks, connected in tandem and trained together end-to-end. We evaluate our approach on the 2017 MICCAI Liver Tumour Segmentation Challenge, attaining competitive liver and liver lesion detection and segmentation scores across a wide range of metrics. Unlike other top performing methods, our model output post-processing is trivial, we do not use data external to the challenge, and we propose a simple single-stage model that is trained end-to-end. However, our method nearly matches the top lesion segmentation performance and achieves the second highest precision for lesion detection while maintaining high recall.

4.1.2 Introduction

The segmentation of liver tumours tumours in computed tomography (CT) is required for assessment of tumour load, treatment planning, prognosis, and monitoring of treatment response. Because manual segmentation is time consuming, tumour size is usually estimated in clinical practice from measurements in the axial plane of the largest diameter of the tumour and the diameter perpendicular to it [5]. Nevertheless, tumour volume is a better predictor of patient survival than diameter [6]. Hence, there is a clear need for tools to aid with tumour detection and segmentation.

Recent advances in computer vision have spurred the resurgence and refinement of deep neural networks which can now exceed human performance in object classification from natural images [96]. Exploration of this promising avenue has only recently begun for medical image segmentation. Current models [80, 90, 91, 102, 165] are based on fully convolutional neural networks (FCN) [2, 3], often similar to the UNet [19]. We exploit the architecture that is evaluated in [90] to construct a model configuration for segmenting metastatic lesions in the liver within CT volumes.

We attain competitive liver and liver lesion detection and segmentation scores across a wide range of metrics in the 2017 MICCAI Liver Tumour Segmentation Challenge (LiTS). Unlike other top scoring methods, we do not pre-process the data, we employ only trivial post-processing of model outputs, and we propose a single-stage model that is trained end-to-end.

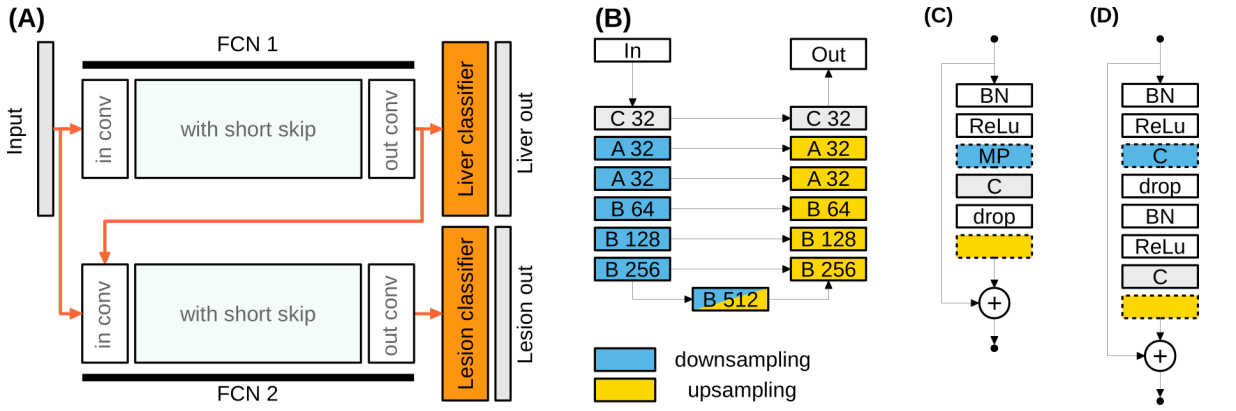


Figure 4.1 (A) Two FCNs, FCN 1 and 2, each take a 2D axial slice as input. FCN 1 produces a segmentation mask for the liver; FCN 2 for lesions. The latent representation produced by FCN 1 is passed as an additional input to FCN 2. (B) FCN structure with the number of convolution filters noted in each block. Blocks coloured blue perform downsampling while those coloured yellow perform upsampling. “C" denotes a 3x3 pixel convolution layer; “A" and “B" denote blocks A and B, shown in (C) and (D), respectively. “BN", “ReLU", and “MP", denote batch normalization, rectified linear units, and max pooling, respectively. Blocks with dashed lines are used in only the upsampling or the downsampling path, as denoted by colour.

4.1.3 Method

4.1.3.1 Model

We construct a model with two fully convolutional networks (FCNs), one on top of the other, trained end-to-end to segment 2D axial slices. Both networks are UNet-like [19] with short and long skip connections as in [90]. The combined network is shown in Figure 4.1 (A). FCN 1 takes an axial slice as input and its output is passed to a linear classifier that outputs (via a sigmoid) a probability for each pixel being within the liver. FCN 2 takes as input both the axial slice and the output of FCN 1. The input thus has a number of channels equal to the number of channels in the representation produced by FCN 1 plus one channel which contains the axial slice. The representation produced by FCN 1 is effectively passed to every layer of FCN 2 due to short skip connections, after first passing through the first convolution layer of FCN 2. The output representation of FCN 2 is passed to a lesion classifier, of the same type as the liver classifier.

The FCN 1 and 2 networks have an identical architecture, as shown in Figure 4.1 (B). In each FCN, an input passes through an initial convolution layer and is then processed by a sequence of convolution blocks at decreasing resolutions and an increasing receptive field size. This contracting path is shown in blue on the left. An expanding path (right, in

yellow) then reverses the downsampling performed by the contracting path. The expanding path mirrors the structure of the contracting path. Each block in the expanding path takes as input the sum of the previous block's output and the output of its corresponding block from the contracting path; this allows the expanding path to recover spatial detail lost with downsampling. Representations are thus skipped from left to right along long skip connections.

We used two types of blocks: block A and block B. Both have short skip connections which sum the block's input into its output, as shown in Figure 4.1 (C), (D). Both blocks contain dropout layers, a downsampling layer when used along the contracting path, and an upsampling layer when used along the expanding path. The downsampling layer in block A is max pooling. In block B it is basic grid subsampling, achieved by applying convolutions with a stride of 2. The upsampling layer performs simple nearest neighbour interpolation. The main difference between blocks A and B is in the number of convolution operations: block A contains one convolution layer and block B contains 2. All convolution layers use 3x3 filters; the number of filters is shown for each block in Figure 4.1 (B).

4.1.3.2 Data set

The proposed segmentation method was applied to metastatic lesions in the liver imaged with CT. The dataset included 200 CT volumes with variable coverage, either limited to the abdomen or including the entire abdomen and thorax. All volumes were enhanced with a contrast agent, imaged in the portal venous phase. All volumes contained a variable number of axial slices with a resolution of 512x512 pixels, with varying slice thicknesses. Of the 200 volumes, 130 volumes were provided publicly with manual segmentations of the liver and liver lesions while 70 were withheld until near the end of the LiTS challenge for evaluation. Manual segmentations were not provided for this evaluation set.

Of the 130 cases with segmentations, we used 115 for training and 15 for validating our segmentation models. We did not apply any pre-processing to the images except for basic image-independent scaling of the intensities to ensure inputs to our neural networks were within a reasonable range: we divided all pixel values by 255 and then clipped the resulting intensities to within [-2, 2].

4.1.3.3 Training the model

We trained the model only on 2D axial slices that contain the liver, using RMSprop [166] and the Dice loss defined in [90, 91]. For data augmentation, we applied random horizontal

and vertical flips, rotations up to 15 degrees, zooming in and out up to 10%, and elastic deformations as described by [19]. In order to improve training time, allowing us to test many models and hyperparameters in a short time, we first downscaled all slices from a 512x512 resolution to 256x256. This initial model was trained with a 0.001 learning rate (0.9 momentum). The model was then fine-tuned on full resolution slices, using a 0.0001 learning rate.

Table 4.1 Segmentation and detection metrics evaluated for the proposed method on the MICCAI LiTS 2017 test set.

Segmentation						
	Dice	VOE (%)	RVD (%)	ASSD (mm)	MSD (mm)	RMSD (mm)
leHealth	-	39.4	5.921	1.189	6.682	1.726
hchen	-	35.6	5.164	1.073	6.055	1.562
hans.meine	-	38.3	0.464	1.143	7.322	1.728
our	0.773	35.7	12.124	1.075	6.317	1.596
Detection	**>50% overlap**		**>0% overlap**		**Mixed measures**	
	Precision	Recall	Precision	Recall	Global Dice	Dice per case
leHealth	0.156	0.437	-	-	0.794	0.702
hchen	0.409	0.408	-	-	0.8290	0.686
hans.meine	0.496	0.397	-	-	0.796	0.676
our	0.446	0.374	0.686	0.574	0.783	0.661

The model was trained for 200 epochs on downscaled slices (batch size 40) and fine-tuned for 30 epochs on full-resolution slices (batch size 10). The final model weights were those which yielded the best loss on the validation set.

The proposed model is limited to processing 2D slices due to memory constraints. To improve segmentation performance and consistency across slices for the LiTS challenge, we introduced some cross-slice context. For every slice, three consecutive slices were considered (one above, one below). The pre-classifier outputs from each of the three slices were combined by a convolution (3x3 kernel); a new classifier for the middle slice was trained on the resultant features.

4.1.3.4 Generating segmentations

At test time, segmentation predictions were averaged across all four input orientations achieved by vertical and horizontal flips. This was done for three similar models and the predictions of the ensemble were averaged. A liver segmentation was extracted by selecting the largest connected component in the model's liver segmentation prediction. A lesion segmentation was extracted by cropping the model's lesion segmentation prediction to a dilated

version of the liver segmentation. For dilation, we chose to extend the liver's boundaries by 20mm. This eliminated false positives outside of the liver without incorrectly cropping out lesions when the liver is slightly under-segmented. Beyond cropping to a single liver, no post-processing was performed on the model outputs.

4.1.4 Results and discussion

The proposed method performed relatively well in the MICCAI LiTS challenge, achieving similar scores to other top methods, as shown in Table 4.1 (example segmentation in Figure 4.2. Segmentation metrics evaluate the segmentation of detected lesions (averaged across lesions). They are comprised of a per-lesion Dice score, a volume overlap error (VOE), a relative volume difference (RVD), the average symmetric surface distance (ASSD), the maximum surface distance (MSD), and the root means square symmetric surface distance (RMSD). Detection metrics were evaluated as precision and recall at >50% and >0% overlap (measured by intersection over union) of each predicted lesion with the corresponding ground truth. Dice metrics that confound both detection and segmentation were the Dice score computed on all combined volumes (global Dice) and the mean Dice score per volume (Dice per case). Entries in the challenge were ranked according to the Dice per case, placing our method fourth in lesion segmentation with a score of 0.661. Liver segmentation performed well, with an average Dice per case of 0.951 (the best entry scored 0.963).

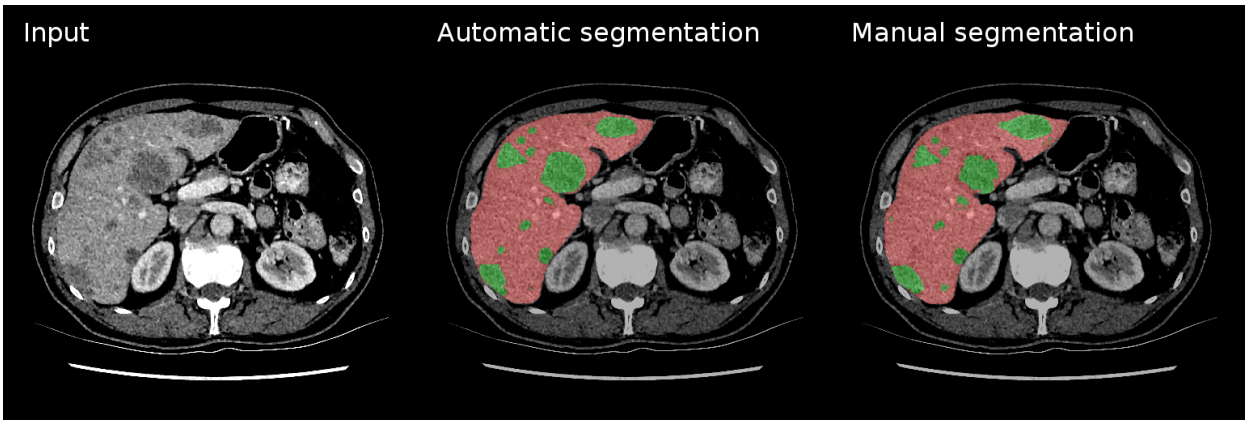


Figure 4.2 Example of segmentation output compared to ground truth ("Manual segmentation"). Lesions in green, liver in red.

Although three methods attained higher mean Dice per case, our method compares favourably in terms of higher detection scores or lower complexity. While *leHealth* attained the top Dice per case score of 0.702, the method suffers from very low precision (0.156 compared to our 0.446, at >50% overlap). It also relies on extensive model ensembling and post-processing.

The second method, labeled *hchen* in Table 4.1, attained a Dice per case of 0.686 but at the cost of a lower precision (0.409 at >50% overlap) [25]. This method relies on a three-stage process where the liver is first roughly segmented, then liver and lesions are segmented with a 2D FCN, and finally, the segmentations are refined with a small 3D FCN that takes the initial segmentation predictions as input. The authors found that using a pre-trained 2D model significantly boosted performance. By contrast, we developed a single-stage pipeline in which we did not use pre-trained models; we will extend our method to 3D in the future. Finally, *hans.meine* (using the approach described in [24]) attained a Dice per case of 0.676 with detection scores at 50% overlap that are slightly higher than for our method; however, that method involved post-processing with a random forest classifier to improve precision and used data other than that provided in the challenge to train a liver segmentation model. In comparison, our post-processing was trivial and we trained our model on only the data provided in the challenge.

All top methods used an FCN for lesion segmentation, conditioned on prior liver segmentation. In this regard, our approach differs only in that it is a single-stage model, trained end-to-end, and the lesion segmentation (FCN 2) is conditioned on the high-dimensional pre-classifier representation in the liver (FCN 1), rather than on the liver classifier outputs. This configuration allows FCN 2 to focus on the liver when performing lesion segmentation and ignore lesions far from the liver. We found that using a single FCN to segment lesions and the liver simultaneously is less effective. This may be because this does not model the dependence of the lesion segmentations on that of the liver. In addition, training FCN 1 and 2 end-to-end allows FCN 1 to learn a representation that is amenable for lesion segmentation, boosting the performance of FCN 2. Indeed, [167] found that an FCN may act as an effective learned pre-processor for another FCN.

4.1.5 Conclusion

The proposed model performs end-to-end joint liver and lesion segmentation in CT quickly without any need for pre-processing of input images or complicated post-processing of the outputs. Segmentation performance could be improved by extending the proposed model to processing the whole CT volume rather than slice inputs. The proposed model's simplicity makes it a good base model for architectural research toward improving liver and liver lesion segmentation.

4.1.6 Acknowledgements

We thank An Tang and Gabriel Chartrand for preparing some of the data used in the LiTS challenge.

4.2 Liver tumour segmentation (LiTS) challenge

The liver tumour segmentation (LiTS) challenge [1] was held in 2017 at ISBI and then again at MICCAI. Its goal was to establish the state of the art for the segmentation of colorectal metastases in the liver in CT on a standardized dataset. The training data remains publicly available. Furthermore, the evaluation system is also publicly hosted online in order to continue evaluating submitted methods on a withheld test set, the segmentation labels for which are not available to the public.

4.2.1 Data

The LiTS dataset contains 201 CT volumes of which 194 contains lesions with portal venous phase enhancement. 131 cases are publicly available for training segmentation models and 70 cases are withheld from the public and reserved for testing. Cases contain a variety of lesion types, including colorectal, breast, and lung metastatic tumours, as well as primary hepatocelular carcinoma. The number of lesions per case ranges from none to 75. This data is very variable in many respects. Sourced from seven academic and clinical sites around the world, this data is acquired with different CT scanners and acquisition protocols. Both the image resolution and the field of view are highly variable. Both hyper-dense and hypo-dense contrast enhanced images are included. Some images contain artefacts. The cases are a mix of both pre- and post-operative scans. Liver and lesions in the liver were manually segmented by a variety of trained radiologists or technicians and the segmentations were verified by three expert radiologists in an expert review.

4.2.2 Evaluation criteria

All submissions to the LiTS challenge were evaluated by an online system on a test set that is withheld from the public. Evaluation metrics considered the detection and segmentation of lesions and the segmentation of the liver. Although many metrics were computed, all submissions were ranked according to a single metric, defined in section 4.2.2.1.

4.2.2.1 Mixed metrics

There are two related metrics that jointly evaluate detection and segmentation performance, based on the Dice score. The Dice score is an F1 score which measures the harmonic mean of precision and recall, in this case for binary pixel classification. This score is essentially a per-pixel/voxel detection score. When applied to a binary segmentation task, it evaluates the

degree of overlap between the predicted segmentation mask and the reference segmentation mask. Given binary masks A and B, the Dice score evaluates as:

$$DICE(A, B) = \frac{2|A \cap B|}{|A| + |B|}. \tag{4.1}$$

in the interval [0,1]; a perfect segmentation yields a Dice score of 1.

Poor segmentation manifests in a poor overlap as measured by the Dice score. For lesion segmentation, the success of lesion detection also impacts the Dice score though not in an ideal manner. False positive and false negative lesion predictions impart a penalty in terms of the erroneous overlap of the prediction with the reference mask. However, this penalty depends on the relative size of the erroneous lesion with respect to the collective sizes of rest of the predicted and reference lesions. Thus, failing to predict a nodular lesion is much more costly in a volume with no other lesions than in a volume with many lesions or with another large lesion. In section 4.2.2.3, we separate the Dice score from lesion detection by evaluating it only for each detected lesion, as a segmentation metric.

As a mixed segmentation and detection metric, we evaluate the Dice score in two ways. First, as a *global Dice* score, applied across all cases as if they combine in a single volume. Second, as a Dice score applied per case (*Dice per case*) and averaged over all cases. The global Dice score is affected more by large lesions than by small lesions. The Dice per case score applies a higher penalty to prediction errors in cases with fewer actual lesions. Despite the drawbacks of these metrics, they are fairly informative; nevertheless, we also compute a host of other metrics to better understand detection and segmentation performance.

4.2.2.2 Detection metrics

A lesion is considered detected if the predicted lesion has a sufficient overlap with its corresponding reference lesion, measured as the intersection over union of their respective segmentation masks. This allows for a count of true positive, false positive, and false negative detections, from which we compute the *precision* and *recall* of lesion detection. Detection performance is evaluated at intersection over union greater than 0 as well as greater than 0.5, with the former being the most sensitive.

Since lesions are not predicted for one reference lesion at a time, a correspondence between reference lesions and predictions must be established. Connected components are identified in both the prediction mask and the reference mask. Components may not necessarily have

a one-to-one correspondence between the two masks. A single reference component can be predicted as multiple components (split error); similarly, multiple reference components can be covered by a single large predicted component (merge error). As a first step, the detection algorithm turns this many-to-many mapping into a many-to-one mapping by merging all reference lesions that are connected by predicted components. Thus, a single corresponding (merged) reference component is found for every predicted component (except those that do not overlap any reference component). Before evaluating intersection over union, all predicted components that correspond to the same reference component are merged. In order to maintain the immutability of the reference, detection of any merged components in the reference masks counts for the number of lesions merged.

4.2.2.3 Segmentation metrics

We evaluated the quality of segmentation of the liver and of detected lesions (at intersection over union greater than 0.5). These include overlap measures and surface distance metrics. Of the four overlap measures, three are reformulations of the same measurement: *Dice* score, *Jaccard* index, and volume overlap error (VOE). The Dice score is measured for each detected lesion as in equation 4.1. The Jaccard index is the intersection over union of the predicted lesion mask with the reference lesion mask. Volume overlap error is the complement of the Jaccard index:

$$VOE(A, B) = 1 - \frac{|A \cap B|}{|A \cup B|}. \tag{4.2}$$

The relative volume difference (**RVD**) is an asymmetric measure defined as:

$$RVD(A, B) = \frac{|B| - |A|}{|A|}. \tag{4.3}$$

Surface distance metrics are a set of correlated measures of the distance between the surfaces of a reference and predicted lesion. Let $S(A)$ denote the set of surface voxels of A. The shortest distance of an arbitrary voxel v to $S(A)$ is defined as:

$$d(v, S(A)) = \min_{s_A \in S(A)} ||v - s_A||, \tag{4.4}$$

where $||.||$ denotes the Euclidean distance. The average symmetric surface distance (**ASD**)

is then given by:

$$ASD(A,B) = \frac{1}{|S(A)| + |S(B)|}\left(\sum_{s_A \in S(A)} d(s_A, S(B)) + \sum_{s_B \in S(B)} d(s_B, S(A))\right). \tag{4.5}$$

The maximum symmetric surface distance (**MSD**), also known as the Symmetric Hausdorff Distance, is similar to ASD except that the maximum distance is taken instead of the average:

$$MSD(A,B) = \max\left\{\max_{s_A \in S(A)} d(s_A, S(B)), \max_{s_B \in S(B)} d(s_B, S(A))\right\}. \tag{4.6}$$

4.2.2.4 Tumor burden

The tumor burden of the liver is a measure of the fraction of the liver afflicted by cancer. As a metric, we measure the root mean square error (RMSE) in tumor burden estimates from lesion predictions.

$$RMSE = \sqrt{\frac{1}{n}\Sigma_{i=1}^{n}\Big(A_i - B_i\Big)^2} \tag{4.7}$$

4.2.3 Contribution

My contributions to setting up the LiTS challenges were:

- Determining all metrics to use for evaluation.
- Writing efficient metric evaluation code, hosted on the submission site.
- Co-writing the paper [1].

Because I submitted my own entry to the challenge, described in Section 4.1, I did not choose which evaluation metrics were to be used for ranking submissions and in which way. Instead, I encouraged the other co-organizers to make those decisions as they saw fit.

4.2.4 Challenge results

The results of the MICCAI 2017 LiTS challenge for all accepted entries are given in Table 4.2 for lesion segmentation and 4.3 for lesion detection. Values for all metrics in Section 4.2.2 are listed for these entries. All entries produced segmentations that significantly disagreed with the ground truth segmentations prepared by human operators. The degree of this disagreement is presented in Section 4.3 for the method in Section 4.1.

This method placed fourth in the segmentation rankings when ranked by Dice per case (Table 4.2). However, this method outperforms the top three in other metrics. The Dice per case metric is noisy because it applies a higher penalty to prediction errors in cases with fewer actual lesions, so the value of the penalty depends on the image within which an error is made. Furthermore, the Dice per case metric computes a minimal score of zero for those cases that contain no lesions if any voxel in the entire volume is classified as a lesion. This kind of error can significantly affect the score.

All entries showed poor automated lesion detection. It is important to consider that the lesion detection scores in Table 4.3 are for intersection over union "overlap" greater than 0.5 (see Section 4.2.2) which is quite strict about what constitutes a detection. Nevertheless, detection is still shown to be poor in Section 4.3 for the method in Section 4.1 with the most relaxed definition of detection (at overlap greater than 0), even though this method has among the best precision and recall scores in the challenge. It is particularly the small lesions that are frequently not detected. Clearly, automated segmentation and detection of liver lesions needs improvement.

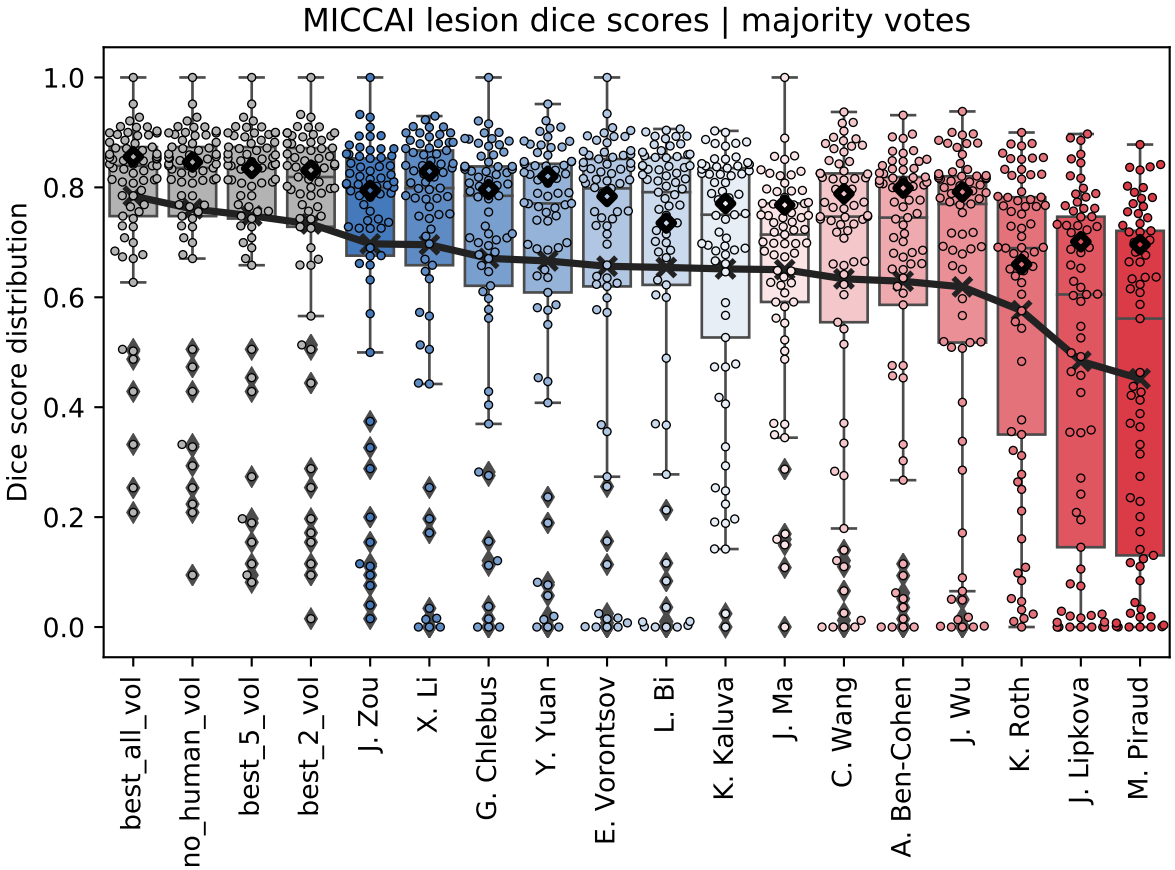


Figure 4.3 Dispersion of test Dice scores from individual algorithms described in short-papers, and various fused algorithmic segmentations (gray). Boxplots show quartile ranges of the scores on the test datasets; whiskers and dots indicate outliers. Black squares indicate the global dice metric whereas the black line indicates the ranking based on the dice per case metric. Also shown are results of four fused algorithmic segmentations. Figure from [1].

Table 4.2 MICCAI lesion submissions ranked by Dice per case score. * indicates missing short paper submission. Table from [1].

Ranking	Name	Instituion	Dice per case	Dice global	VOE	RVD	ASSD	MSD	RMSD
1	Tian et al.	Lenovo	0.7020 (1)	0.7940 (5)	0.394 (11)	5.921 (18)	1.189 (12)	6.682 (5)	1.726 (8)
2	Li et al.	CUHK	0.6860 (2)	0.8290 (1)	0.356 (3)	5.164 (17)	1.073 (5)	6.055 (1)	1.562 (2)
3	Chlebus et al.	Fraunhofer	0.6760 (3)	0.7960 (4)	0.383 (10)	0.464 (12)	1.143 (8)	7.322 (12)	1.728 (9)
4	Vorontsov et al.	MILA	0.6610 (4)	0.7830 (9)	0.357 (4)	12.124 (24)	1.075 (6)	6.317 (3)	1.596 (3)
5	Yuan et al.	MSSM	0.6570 (5)	0.8200 (2)	0.378 (9)	0.288 (11)	1.151 (9)	6.269 (2)	1.678 (5)
6	Ma et al.	NJUST	0.6550 (6)	0.7680 (12)	0.451 (19)	5.949 (19)	1.607 (24)	9.363 (25)	2.313 (24)
7	Bi et al.	Uni Sydney	0.6450 (7)	0.7350 (16)	0.356 (3)	3.431 (13)	1.006 (2)	6.472 (4)	1.520 (1)
8	Kaluva et al.	Predible Health	0.6400 (8)	0.7700 (11)	0.340 (1)	0.190 (9)	1.040 (3)	7.250 (11)	1.680 (6)
9	Han	N.A.	0.6300 (9)	0.7700 (11)	0.350 (2)	0.170 (8)	1.050 (4)	7.210 (9)	1.690 (7)
10	Wang et al.	KTH	0.6250 (10)	0.7880 (7)	0.378 (9)	8.300 (21)	1.260 (15)	6.983 (8)	1.865 (13)
11	Wu et al.	N.A.	0.6240 (11)	0.7920 (6)	0.394 (11)	4.679 (14)	1.232 (14)	7.783 (17)	1.889 (14)
12	Ben-Cohen et al.	Uni Tel Aviv	0.6200 (12)	0.8000 (3)	0.360 (5)	0.200 (10)	1.290 (16)	8.060 (19)	2.000 (16)
12*	LP777	N.A.	0.6200 (12)	0.8000 (3)	0.421 (14)	6.420 (20)	1.388 (20)	6.716 (6)	1.936 (15)
13*	Micro	N.A.	0.6130 (13)	0.7830 (9)	0.430 (16)	5.045 (16)	1.759 (27)	10.087 (26)	2.556 (25)
13*	Njrwin	N.A.	0.6130 (13)	0.7640 (13)	0.361 (6)	4.993 (15)	1.164 (11)	7.649 (15)	1.831 (12)
14*	mbb	ETH Zurich	0.5860 (14)	0.7410 (15)	0.429 (15)	39.763 (27)	1.649 (26)	8.079 (20)	2.252 (23)
15*	szm0219	Uni Illinois	0.5850 (15)	0.7450 (14)	0.364 (7)	0.001 (4)	1.222 (13)	7.408 (13)	1.758 (10)
16*	MICDIIR	N.A.	0.5820 (16)	0.7760 (10)	0.446 (18)	8.775 (22)	1.588 (23)	7.723 (16)	2.182 (22)
17	Roth et al.	Volume Graphics	0.5700 (17)	0.6600 (20)	0.340 (1)	0.020 (5)	0.950 (1)	6.810 (7)	1.600 (4)
18*	jkan	N.A.	0.5670 (18)	0.7840 (8)	0.364 (7)	0.112 (7)	1.159 (10)	7.230 (10)	1.690 (7)
19*	huni1115	N.A.	0.4960 (20)	0.7000 (18)	0.400 (12)	0.060 (6)	1.342 (19)	9.030 (24)	2.041 (17)
20*	mahendrakhened	IITM	0.4920 (21)	0.6250 (23)	0.411 (13)	19.705 (26)	1.441 (21)	7.515 (14)	2.070 (19)
21	Lipkova et al.	TU Munich	0.4800 (22)	0.7000 (18)	0.360 (5)	0.060 (6)	1.330 (17)	8.640 (22)	2.100 (20)
22*	jinqi	N.A.	0.4710 (23)	0.6470 (22)	0.514 (20)	17.832 (25)	2.465 (28)	14.588 (28)	3.643 (27)
23*	MIP_HQU	N.A.	0.4700 (24)	0.6500 (21)	0.340 (1)	-0.130 (1)	1.090 (7)	7.840 (18)	1.800 (11)
24	Piraud et al.	TU Munich	0.4450 (25)	0.6960 (19)	0.445 (17)	10.121 (23)	1.464 (22)	8.391 (21)	2.136 (21)
25*	QiaoTian	N.A.	0.2500 (26)	0.4500 (24)	0.370 (8)	-0.100 (2)	1.620 (25)	11.720 (27)	2.620 (26)

Table 4.3 Table MICCAI precision and recall scores for submissions. Submissions ranked by lesion Dice per case score. * indicates missing short paper submission. Table from [1].

Ranking	Name	Instituion	Precision at 50% overlap	Recall at 50 % overlap
1	Tian et al.	Lenovo	0.156 (14)	0.437 (3)
2	Li et al.	CUHK	0.409 (4)	0.408 (4)
3	Chlebus et al.	Fraunhofer	0.496 (2)	0.397 (5)
4	Vorontsov et al.	MILA	0.446 (3)	0.374 (6)
5	Yuan et al.	MSSM	0.328 (5)	0.397 (5)
6	Ma et al.	NJUST	0.499 (1)	0.289 (17)
7	Bi et al.	Uni Sydney	0.315 (6)	0.343 (11)
8	Kaluva et al.	Predible Health	0.140 (16)	0.330 (12)
9	Han	N.A.	0.160 (13)	0.330 (12)
10	Wang et al.	KTH	0.160 (13)	0.349 (9)
11	Wu et al.	N.A.	0.179 (12)	0.372 (7)
12	Ben-Cohen et al.	Uni Tel Aviv	0.270 (7)	0.290 (16)
12*	LP777	N.A.	0.239 (9)	0.446 (2)
13*	Micro	N.A.	0.095 (19)	0.328 (13)
13*	jrwin	N.A.	0.241 (8)	0.290 (16)
14*	mbb	ETH Zurich	0.054 (23)	0.369 (8)
15*	szm0219	Uni Illinois	0.224 (10)	0.239 (19)
16*	MICDIIR	N.A.	0.143 (15)	0.463 (1)
17	Roth et al.	Volume Graphics	0.070 (20)	0.300 (15)
18*	jkan	N.A.	0.218 (11)	0.250 (18)
19*	huni1115	N.A.	0.041 (25)	0.196 (22)
20*	mahendrakhened	IITM	0.117 (17)	0.348 (10)
21	Lipkova et al.	TU Munich	0.060 (22)	0.190 (23)
22*	jinqi	N.A.	0.044 (24)	0.232 (20)
23*	MIP_HQU	N.A.	0.030 (26)	0.220 (21)
24	Piraud et al.	TU Munich	0.068 (21)	0.325 (14)
25*	QiaoTian	N.A.	0.010 (27)	0.060 (25)

4.3 (Article 2) Deep learning for automated segmentation of liver lesions on computed tomography in patients with colorectal cancer liver metastases

This paper presents an analysis on the clinical utility of the state of the art in liver lesion segmentation, as determined by the LiTS challenges. The inter-operator variability is evaluated for manual segmentation as well as for operator corrections of automated segmentations. The time taken for segmentation is also evaluated. The automated model used here is the one presented in the previous section since it ranked among other top methods in the challenges. Training details are provided in Appendix A.1 and are the same as in Section 4.1.

Title

Deep Learning for Automated Segmentation of Liver Lesions on Computed Tomography in Patients with Colorectal Cancer Liver Metastases

Authors

Eugene Vorontsov[1,2,*], Milena Cerny[3,4,*], Philippe Régnier[3], Lisa Di Jorio[6], Christopher Pal[1,2], Réal Lapointe[5], Franck Vandenbrouke-Menu[5], Simon Turcotte[3,5], Samuel Kadoury[1,3], An Tang[3,4]

[*] Equal contribution

Affiliations

[1] École Polytechnique de Montréal

[2] Montreal Institute for Learning Algorithms (MILA)

[3] Centre de Recherche du Centre hospitalier de l'Université de Montréal (CRCHUM)

[4] Department of Radiology, Centre hospitalier de l'Université de Montréal (CHUM)

[5] Department of Surgery, Hepatopancreatobiliary and Liver Transplantation Division, Centre hospitalier de l'Université de Montréal (CHUM)

[6] Imagia Cybernetics

Publication

This paper has been published in the journal *Radiology: Artificial Intelligence* on March 13, 2019, Volume 1, Issue 2, pp 180014. © Radiological Society of North America

Reproduced in this dissertation with permission from the publisher.

Contribution

My contributions in this work were model design, training, and application; statistical analysis design; all code for evaluation and statistical analysis; performing the evaluation and analysis; and some of the writing.

4.3.1 Summary statement

A deep learning method shows promise for facilitating detection and segmentation of colorectal liver metastases. User correction of automated segmentations can generally resolve deficiencies of fully automated segmentation for small metastases and is faster than manual segmentation.

4.3.2 Implications for patient care

1. Use of deep learning with convolutional neural networks may prove useful for lesion detection and segmentation in patients with colorectal liver metastases on contrast-enhanced CT.

2. A user-corrected method (i.e. automated segmentations manually corrected by an image analyst) achieved similar or higher per-patient detection performance (sensitivity: 0.76 - 0.83 and positive predictive value [PPV]: 0.94 - 0.95) than manual segmentation (sensitivity: 0.82 - 0.83, PPV: 0.86 - 0.91) or fully automated segmentation (sensitivity: 0.59, PPV: 0.80).

3. Fully automated and user-corrected segmentations were more time efficient than manual segmentation with automated run time per volume under 1 sec and a mean interaction time of 4.8 ± 2.1 min vs 7.7 ± 2.4 min ($P < 0.001$).

4.3.3 Abstract

Purpose: To evaluate the performance, agreement, and efficiency of a fully convolutional network (FCN) for lesion detection and segmentation on computed tomography (CT) examinations in patients with colorectal liver metastases (CLM).

Materials and Methods: This retrospective study evaluated an automated method using FCN, that was trained, validated, and tested with 115, 15, and 26 contrast-enhanced CT examinations containing 261, 22, and 105 lesions, respectively. Manual detection and segmentation by a radiologist was the reference standard. Performance of fully automated and user-corrected segmentations were compared to manual segmentations. The inter-user agreement and interaction time of manual and user-corrected segmentations were assessed. Analyses included performance detection, segmentation accuracy, Cohen's kappa, Bland-Altman analyses, and analysis of variance.

Results: For lesion size < 10 mm, 10-20 mm, and > 20 mm, the detection sensitivity of

the automated method was 10%, 71%, and 85%; positive predictive value was 25%, 83%, and 94%; Dice similarity coefficient was 0.14, 0.53, and 0.68; maximum symmetric surface distance was 5.2, 6.0, and 10.4 mm; and average symmetric surface distance was 2.7, 1.7, and 2.8 mm, respectively. For manual and user-corrected segmentation, the kappas were 0.42 and 0.52; inter-reader agreement on volume were -0.10 $\pm$ 0.07 and -0.10 $\pm$ 0.08; and mean interaction time of 7.7 $\pm$ 2.4 min and 4.8 $\pm$ 2.1 min ($P < 0.001$), respectively.

Conclusion: A FCN-based automated detection and segmentation method provided higher performance for larger lesions. Agreement was similar for manual and user-corrected segmentation. However, user-corrected segmentation was more time efficient.

4.3.4 Introduction

Colorectal cancer is the third most commonly diagnosed cancer worldwide [168, 169]. More than half of patients with colorectal cancer develop liver metastases at the time of diagnosis or later during the progression of their disease [170]. The 5-year survival decreases from 64.3% to 11.7% in the presence of colorectal liver metastases (CLM) which constitute the leading cause of death in these patients [171]. Surgical resection of CLMs, when possible, is the standard of care [172], achieving long-term survival [173] and possibility of cure [174]. More efficient chemotherapy, evolution of surgical resectability criteria and strategies to improve CLM resectability [172] have contributed to improvement in patient survival [175].

Preoperative imaging, often obtained by computed tomography (CT) [176] is mandatory to determine the number, size, and location of CLMs with respect to adjacent structures; inform prognosis [177]; and to evaluate resectability [178]. The current approach for assessment of tumor burden relies on uni- or bi-dimensional measurements of the tumor along the largest axes [5], applied to lesions measuring more than 1 cm [5] and for a total of five lesions with a maximum of two per organ according to Response Evaluation Criteria In Solid Tumors (RECIST) 1.1 [5]. However, uni- or bi-dimensional diameter measurement of liver metastases [179] may not accurately reflect tumor size and growth since tumors often have an irregular shape [180]. Tumor segmentation would more accurately capture the tumor volume, growth, and shape. Yet, tumor segmentation is a tedious and time-consuming task susceptible to intra- and inter-operator variability if performed manually by a human. Hence, there is an emerging opportunity for automated tumor segmentation to address all these shortcomings.

Recent studies have demonstrated successful application of deep learning techniques for automated detection, segmentation, and classification tasks [181]. Tumor segmentation tasks may be performed using semi-automated techniques that rely on a combination of interac-

tive or contouring approaches or achieve fully automated segmentation followed by manual corrections, if needed [182, 183]. Among deep learning techniques, fully convolutional neural networks (FCN), consisting of multi-layer neural networks, have become the preferred approach for analysis of medical images [184, 185]. Deep FCNs are capable of learning from examples and building an hierarchical representation of the data [17]. Barriers to the development and clinical adoption of fully automated methods include insufficient training data sets [186–190], lack of labelled data, and undefined performance metrics adapted to clinical needs. We hypothesize that recent advances in deep learning [17, 181] may be applied for fully automated detection and segmentation of CLM. However, there is a need to systematically assess the performance, agreement, and efficiency of deep learning-based detection and segmentation of CLM.

Therefore, the purpose of this study was to evaluate the performance, agreement, and efficiency of two different FCN architectures for lesion detection and segmentation on CT examinations in patients with CLM, using manual segmentation as the reference standard.

4.3.5 Materials and methods

4.3.5.1 Study design and subjects

Our institutional review board at the Centre Hospitalier de l'Université de Montréal approved this retrospective, cross-sectional study. Patient consent was waived for access to the training, validation, and test datasets. This study included two stages: a training and validation stage for various types of liver tumors on a public dataset, as well as a testing stage for specific colorectal liver metastasis (CLM) on an imaging dataset of patients seen at our institution. The study workflow is illustrated in Figure 4.4.

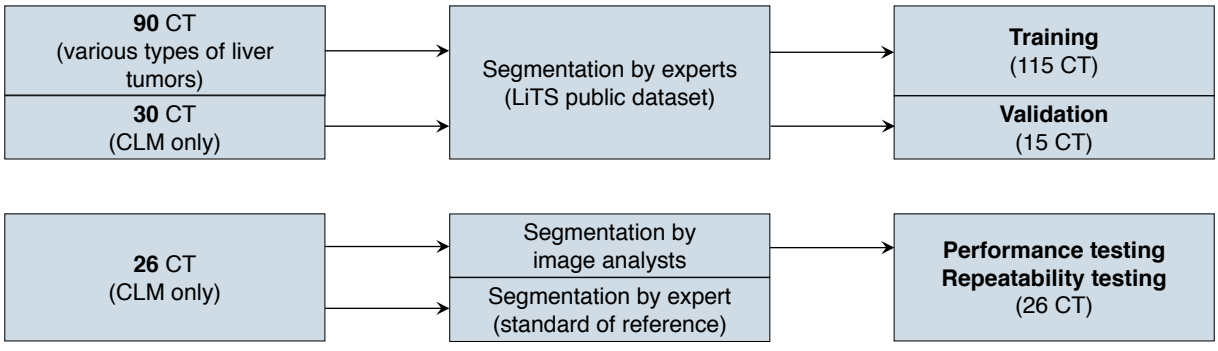


Figure 4.4 Study workflow and datasets used in this study.

4.3.5.2 Training and validation dataset

Contrast-enhanced CT examinations from the public Liver Tumor Segmentation Challenge (LiTS) [188] dataset were used for training and validation. The training and validation sets included 115 CT examinations (261 liver lesions) and 15 CT examinations (22 liver lesions), respectively. Patients had various types of liver tumors, either primary (hepatocellular carcinoma) or metastatic (i.e., colorectal, breast, and lung cancer). Cases from LiTS were provided by seven institutions and therefore performed on different CT scanners with various protocols. Images resolution ranges from 0.56 mm to 1.0 mm in axial plane, and the number of slices from 42 to 1026. The cases included masks of liver metastases segmented at each clinical site by different radiologists, and reviewed by 3 experienced radiologists [188].

4.3.5.3 Testing dataset

The testing dataset included 26 CT examinations (not included in the training dataset) from patients referred to our tertiary center for surgery. The cases were randomly selected from a biobank registered by a Tumor Repository Network [191]. Patients with resected colorectal cancer liver metastases who had undergone baseline (pre-chemotherapy and pre-treatment) CTs between October 2010, and December 2015, at our institution were eligible for registration in the biobank. Patients from the testing set were 16 men and 10 women, mean age 68 years $\pm$ 10 [standard deviation]. The total number of lesions was 105 and the average number of lesions per patient was 4.0 $\pm$ 2.6. The lesions were stratified by size $<$ 10 mm (n = 30), 10-20 mm (n = 35), and $>$ 20 mm diameter (n = 40) with a mean size of 19.4 $\pm$ 15.0 mm [range: 1.10 - 89.9]. Preoperative CT were performed at our institution (n = 10) and at others institutions (n = 16). Typical CT imaging technique parameters used in our center are reported in Table A.1 (Appendix A.2).

4.3.5.4 Model architecture

The model is composed of two fully convolutional networks (FCNs), one on top of the other (Figure 4.5). Both networks have a U-Net [19] type of architecture with short (between layers) and long (across the network) skip connections. FCN1 takes an axial CT slice as input and outputs a probability for each pixel being within the liver. FCN2 takes as input both the axial slice and the output of FCN1, and outputs a probability of each pixel being a lesion. Each FCN is composed of a sequence of convolution blocks that extract features at the expense of spatial details along a contracting path (downsampling). The spatial details are then recovered along an expanding path. At each level along the expanding path, the

corresponding feature representations are integrated to the spatial details through long skip connections. The technical details of the fully automated segmentation method of liver lesions are presented in [164].

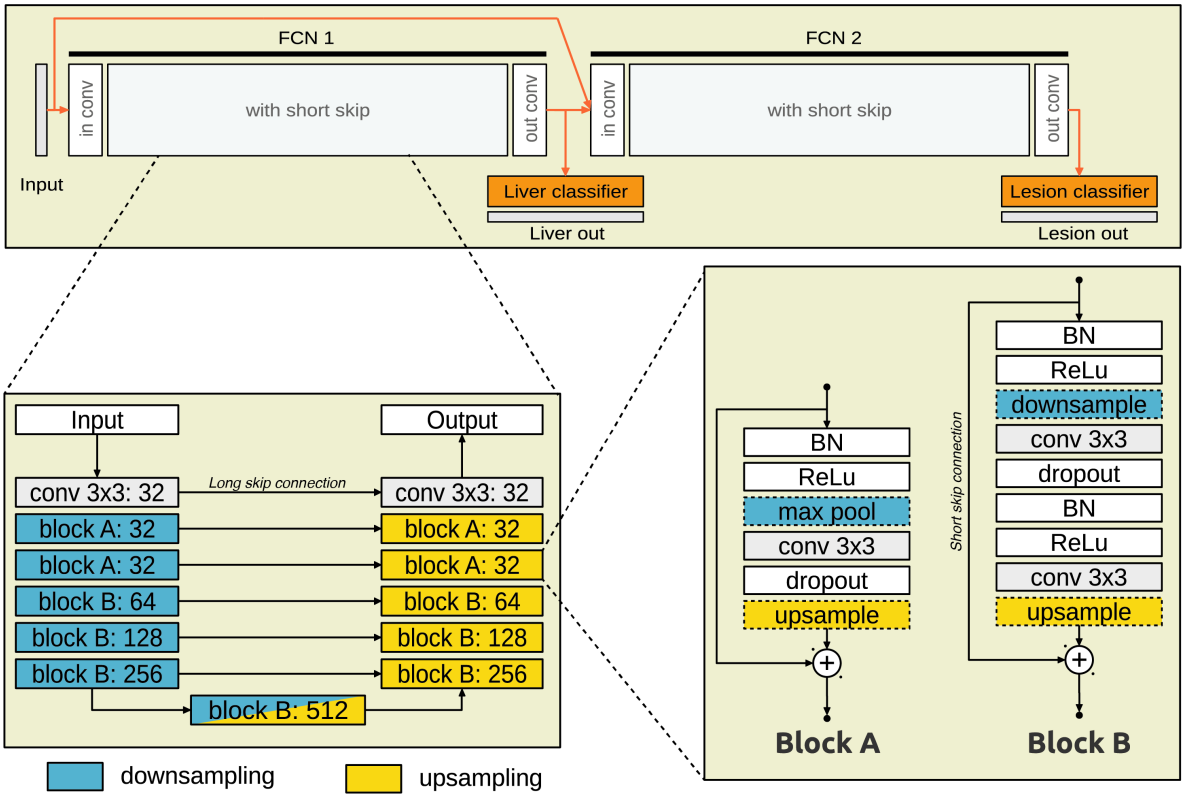


Figure 4.5 Model structure of the CNN used in the study. The CT is provided as input to FCN 1, which outputs probability for each pixel being within the liver. FCN 2 takes as input FCN 1 output and the CT and outputs probability for each liver pixel. *Conv* = convolution kernel; *BN* = batch normalization; *ReLU* = Rectified linear unit.

4.3.5.5 Manual and user-corrected segmentation

For each examination, CT images were imported into a free open-source image post-processing software system (The Medical Imaging Interaction Toolkit [MITK], Heidelberg, Germany) [192] as Digital Imaging and Communications in Medicine (DICOM) files. Binary lesion and non-lesion masks were created by manually segmenting the lesions on the CT images using a cursor to contour the lesions. Lesion volumes were measured by counting the number of constituent voxels and considering the voxel size.

For user-corrected segmentations, CT-examinations with binary lesion and non-lesion masks

created by automated segmentation were imported into MITK and each mask was manually corrected by analysts.

4.3.5.6 Reference standard

Manual segmentation of liver lesions from 26 CT examinations of the testing set by a fellowship-trained abdominal radiologist (M.C., 8 years of experience) was used as the reference standard for lesion detection and segmentation tasks. Lesions were segmented during the portal venous phase as it accentuates the contrast between hypovascular metastatic lesions and normal liver parenchyma [193].

4.3.5.7 Agreement and reliability

To assess the intra- and inter-observer agreement and reliability, the testing set was independently segmented by two image analysts (Walid El Abyad and Assia Belblidia, 3 and 14 years of experience) who performed manual segmentations twice and corrections of automated segmentations twice. Each segmentation session was performed one week apart and segmentations were performed in a randomized order to prevent recall bias.

4.3.5.8 Efficiency

User interaction time was recorded for manual segmentations and for manual corrections of automated segmentations. We evaluated efficiency as interaction time for manual segmentation compared to user-corrected segmentation.

4.3.5.9 Blinding

Image analysts were blinded to their previous segmentation results, those of the other analyst and to the reference standard. The radiologist who performed the manual segmentation of the reference standard was blinded to the segmentation results of image analysts.

4.3.5.10 Statistical analysis

Detection performance—Detection performance was evaluated against the reference standard, and calculated as the intersection over union. Minimal thresholds were reported for overlap: > 0 and > 0.5.

Segmentation accuracy—Accuracy was evaluated against reference standard and assessed for overlap of > 0 and > 0.5 by Dice similarity coefficient (DSC) per detected lesion, maximum

symmetric surface distance (MSSD), and average symmetric surface distance (ASSD) [194]. Measure definitions and formulas are given in Table A.2 (Appendix A.2).

Detection and segmentation agreement—Inter-reader, intra-reader, and inter-method reliability of detection status evaluated against reference standard was estimated using the pooled Cohen's Kappa coefficient (κ) [195].

Repeatability and reproducibility—Per-lesion intra-reader repeatability and inter-reader and inter-method reproducibility on lesion volume estimates were assessed with Bland-Altman analyses. Both replicates of measurements by each image analyst were used in the evaluations. Volume measurements were considered only for lesions in the reference standard that were detected in at least one replicate of either measurement being compared. To approximate homoscedasticity by removing proportional bias, the difference in volume measurements was normalized by the mean of volume measurements, yielding a proportional difference in volume measurements. Accounting for variable replicates, variance and limits of agreement were estimated according to equation 2, and 95% CI were estimated according to equation 4, using the delta method presented by Zou et al [196]. The significance of systematic bias was evaluated with a paired T-test on proportional difference in volume measurements.

Bootstrapping—For all metrics other than those related to Bland-Altman analyses, 95% confidence internals (CI) were estimated by bootstrapping using random sampling methods. The approach would estimate the intervals by repeating the resampling process from the distribution of lesion measurements.

Efficiency—We compared the user interaction time for manual segmentation only with user correction of automated segmentations via two-factor repeated ANOVA.

4.3.6 Results

4.3.6.1 Detection performance

Examples of concordant and discordant detection are shown in Figure 4.6. Per-patient and per-lesion detection sensitivity and PPV for manual, user-corrected, and automated segmentation methods at > 0 and > 0.5 overlap are summarized in Table 4.4 and Table A.3 (Appendix A.2), respectively. For an overlap > 0, manual segmentation achieved per-patient sensitivity of 0.82 - 0.83 and PPV of 0.86 - 0.91, user-corrected automatic segmentation achieved sensitivity of 0.76 - 0.83 and PPV of 0.94 - 0.95, and automated segmentation achieved a sensitivity of 0.59 and a PPV of 0.80.

Per-lesion sensitivity for small lesions < 10 mm was very low with automated segmentation (0.10), but higher for user-corrected segmentation (0.30 - 0.57) and manual segmentation

Table 4.4 Detection performance at minimum overlap > 0. Data in parentheses are 95% confidence intervals. A = automated segmentation, C^i = user-corrected segmentation by reader i, FN = false negative, FP = false positive, M^i = manual segmentations by reader i, TN = true negative, TP = true positive. Number of true-negative findings was not reported, because there is a potentially very high number of nonlesional pixels.

		Overall No. of Liver Lesions					Per Lesion		
Method	Users	TP	FN	FP	TN*	Per Patient	<10 mm	10–20 mm	>20 mm
					Sensitivity				
Manual	M^1	87	19	9	…	0.82 (0.78, 0.88)	0.58 (0.46, 0.70)	0.83 (0.74, 0.92)	1.00 (1.00, 1.00)
	M^2	88	18	14	…	0.83 (0.78, 0.88)	0.70 (0.59, 0.82)	0.81 (0.73, 0.91)	0.95 (0.91, 1.00)
User-corrected	C^1	88	18	6	…	0.83 (0.78, 0.88)	0.57 (0.44, 0.69)	0.89 (0.82, 0.97)	0.99 (0.98, 1.00)
	C^2	80	25	4	…	0.76 (0.71, 0.82)	0.30 (0.18, 0.41)	0.89 (0.82, 0.96)	1.00 (1.00, 1.00)
Automated	…	62	43	16	…	0.59 (0.50, 0.69)	0.10 (0.00, 0.20)	0.71 (0.57, 0.87)	0.85 (0.75, 0.97)
					Positive Predictive Value				
Manual	M^1	87	19	9	…	0.91 (0.88, 0.96)	0.76 (0.64, 0.89)	0.93 (0.87, 0.99)	1.00 (1.00, 1.00)
	M^2	88	18	14	…	0.86 (0.81, 0.91)	0.69 (0.56, 0.80)	0.88 (0.80, 0.96)	0.99 (0.97, 1.00)
User-corrected	C^1	88	18	6	…	0.94 (0.91, 0.98)	0.81 (0.70, 0.94)	0.95 (0.91, 1.00)	1.00 (1.00, 1.00)
	C^2	80	25	4	…	0.95 (0.92, 0.99)	0.82 (0.68, 1.00)	0.94 (0.89, 1.00)	1.00 (1.00, 1.00)
Automated	A	62	43	16	…	0.80 (0.71, 0.89)	0.25 (0.00, 0.50)	0.83 (0.70, 0.98)	0.94 (0.89, 1.00)

Note.—Data in parentheses are 95% confidence intervals. A = automated segmentation, C^i = user-corrected segmentation by reader i, FN = false negative, FP = false positive, M^i = manual segmentations by reader i, TN = true negative, TP = true positive. [21]
* Number of true-negative findings was not reported, because there is a potentially very high number of nonlesional pixels.

Table 4.5 Detection reliability at minimum overlap > 0. Data in parentheses are 95% confidence intervals. A = automated segmentation, C = user-corrected segmentation, C^i = user-corrected segmentation by reader i, $C_j = j^{th}$ user-corrected segmentation by both readers, M = manual segmentation, M^i = manual segmentations by reader i, $M_j = j^{th}$ manual segmentation by both readers.

Parameter	Users	Pooled Cohen κ	Quantity Disagreement	Allocation Disagreement
Intrareader				
Manual	M_1, M_2	0.65 (0.51, 0.81)	0.07 (0.02, 0.10)	0.04 (0.00, 0.07)
User-corrected	C_1, C_2	0.65 (0.51, 0.82)	0.04 (0.00, 0.08)	0.08 (0.03, 0.12)
Interreader				
Manual	M^1, M^2	0.42 (0.24, 0.63)	0.05 (0.01, 0.08)	0.11 (0.06, 0.17)
User-corrected	C^1, C^2	0.52 (0.36, 0.72)	0.07 (0.01, 0.11)	0.09 (0.04, 0.13)
Intermethod				
Automated, manual	M, A	0.31 (0.18, 0.45)	0.24 (0.15, 0.32)	0.08 (0.02, 0.12)
Automated, user-corrected	C, A	0.54 (0.43, 0.67)	0.21 (0.14, 0.27)	0.01 (0.00, 0.03)
Manual, user-corrected	M, C	0.46 (0.32, 0.62)	0.05 (0.01, 0.07)	0.11 (0.08, 0.16)

Note.—Data in parentheses are 95% confidence intervals. A = automated segmentation, C = user-corrected segmentation, C^i = user-corrected segmentation by reader i, C_j = j^{th} user-corrected segmentation by both readers, M = manual segmentation, Mi = manual segmentations by reader i, Mi = jth manual segmentations by both readers.[22]

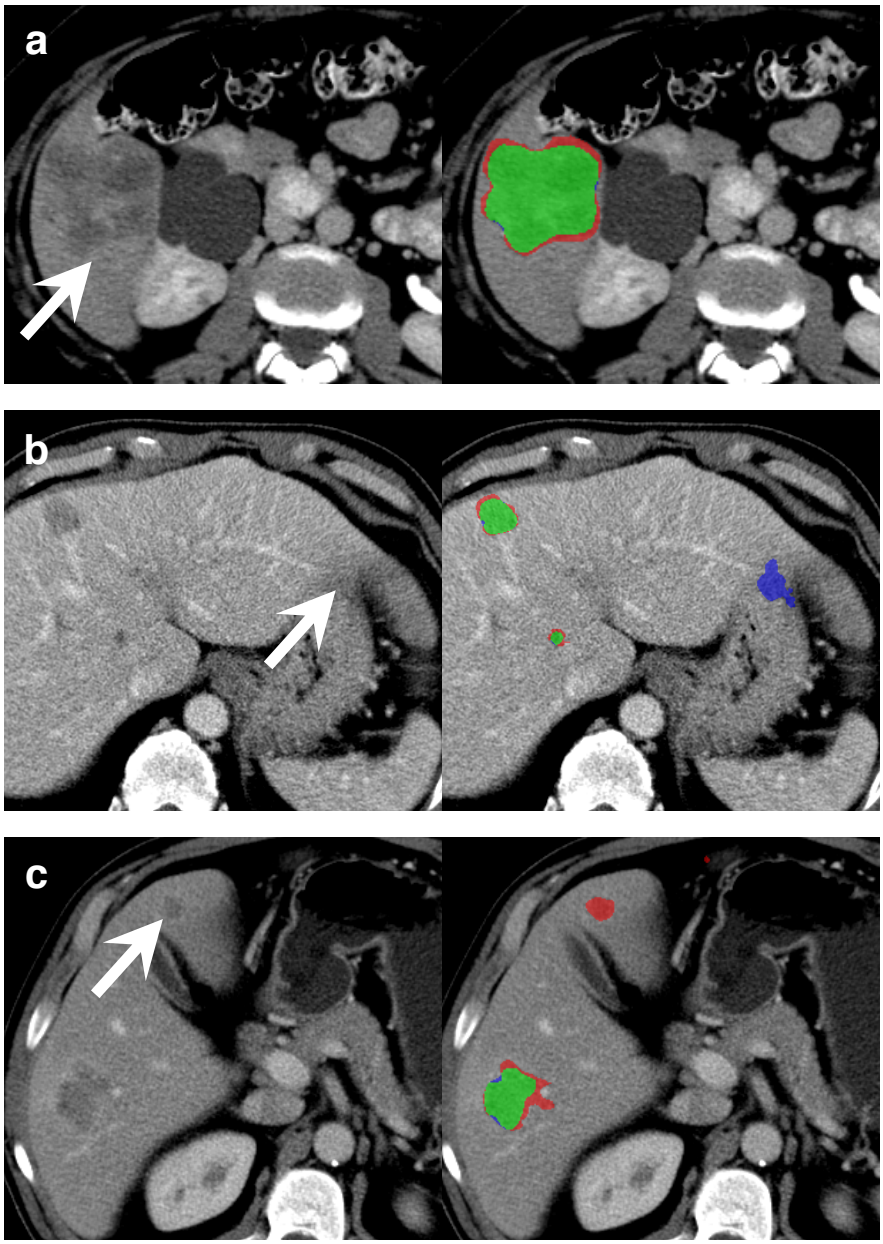


Figure 4.6 Contrast-enhanced axial computed tomography images in three different patients with colorectal liver metastases demonstrating **(a)** good agreement with ground-truth segmentations (green) of a metastasis in segment VI (arrow), **(b)** false-positive pixels (blue) of partial volume in segment II (arrow), and **(c)** false-negative pixels (red) of a metastasis in segment IVb (arrow) for representative cases.

(0.58 - 0.70). Per-lesion sensitivity for lesions 10-20 mm was moderate with automated segmentation (0.71), but higher for user-corrected segmentation (0.89) and manual segmentation (0.81 - 0.83).

4.3.6.2 Segmentation accuracy

Figure 4.7 shows an example lesion surface, produced by the automated method, with a color error map. Metrics of segmentation accuracy for manual, user-corrected, and automated segmentation at > 0 and > 0.5 overlap are summarized in Table 4.6 and Table A.4 (Appendix A.2), respectively. The overall per-lesion Dice similarity coefficients were 0.64 - 0.82 for manual, 0.62 - 0.78 for user-corrected, and 0.14 - 0.68 for automated segmentation methods. The overall MSSD were 3.07 - 6.44 mm for manual, 2.98 - 7.13 mm for user-corrected, and 5.15 - 10.42 mm for the automated segmentation method. The overall ASSD were 0.68 - 0.89 mm for manual, 0.65 - 1.20 mm for user-corrected, and 1.65 - 2.82 mm for automated segmentation methods.

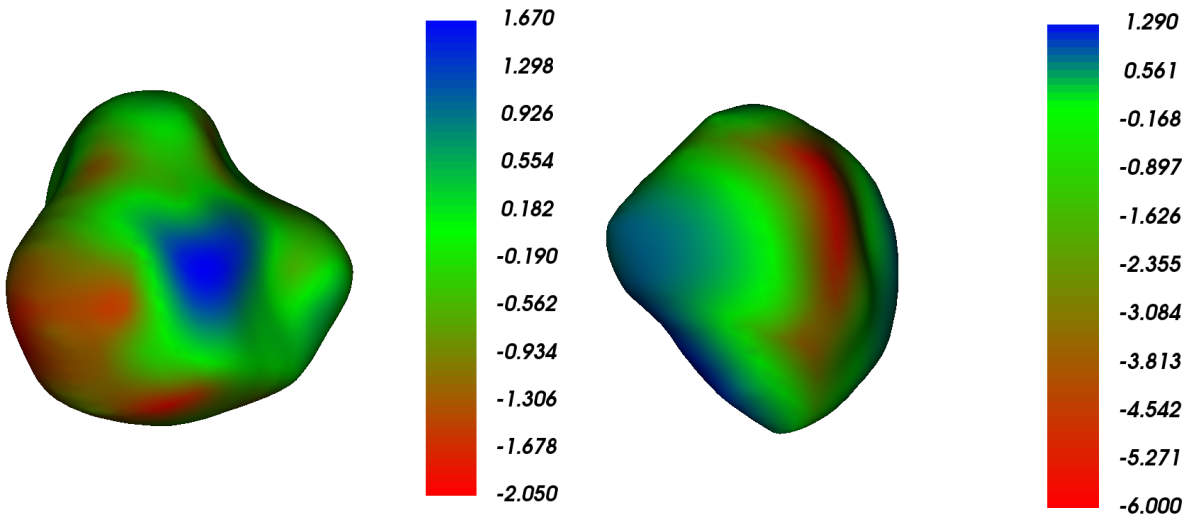


Figure 4.7 An example of a lesion segmented with the automated method, shown from two views. The surface is color mapped according to a signed distance (mm) to the reference surface.

4.3.6.3 Detection reliability

Intra-reader and inter-reader reliability evaluation using the pooled variant of Cohen's Kappa (κ) along with quantity and allocation disagreements, and their confidence intervals, are summarized in Table 4.5 for overlap > 0 and Table A.5 (Appendix A.2) for overlap > 0.5.

At overlap > 0, intra-reader agreement on detection appeared higher than inter-reader agreement ($\kappa = 0.65$ vs $\kappa = 0.42$) for manual segmentation. Agreement between manual and automated segmentations ($\kappa = 0.31$) and between manual and user-corrected segmentations

Table 4.6 Segmentation performance measures at minimum overlap > 0. Data are accuracies with 95% confidence intervals in parentheses. Ideal values for Dice similarity coefficient per detected lesion, maximum symmetric surface distance, and average symmetric surface distance are 1, 0 mm, and 0 mm, respectively. A = automated segmentation, C^i = user-corrected segmentation by reader i, M^i = manual segmentations by reader i.

		Dice Similarity Coefficient per Detected Lesion			Maximum Symmetric Surface Distance (mm)			Average Symmetric Surface Distance (mm)		
Method	Users	<10 mm	10–20 mm	>20 mm	<10 mm	10–20 mm	>20 mm	<10 mm	10–20 mm	>20 mm
Manual	M^1	0.64 (0.60, 0.69)	0.74 (0.72, 0.76)	0.81 (0.79, 0.83)	3.28 (2.92, 3.63)	4.49 (4.10, 4.87)	6.44 (5.70, 7.04)	0.68 (0.55, 0.80)	0.73 (0.65, 0.81)	0.89 (0.75, 1.01)
	M^2	0.65 (0.60, 0.69)	0.74 (0.72, 0.76)	0.82 (0.81, 0.84)	3.07 (2.65, 3.45)	4.77 (4.25, 5.23)	6.09 (5.66, 6.52)	0.66 (0.52, 0.79)	0.76 (0.64, 0.86)	0.80 (0.73, 0.87)
User-corrected	C^1	0.64 (0.58, 0.70)	0.62 (0.58, 0.66)	0.76 (0.73, 0.79)	2.98 (2.52, 3.40)	5.47 (4.98, 5.94)	7.13 (6.51, 7.71)	0.65 (0.45, 0.82)	1.19 (1.04, 1.34)	1.20 (1.04, 1.34)
	C^2	0.67 (0.63, 0.72)	0.65 (0.61, 0.68)	0.78 (0.76, 0.81)	3.62 (2.86, 4.22)	5.34 (4.91, 5.78)	6.96 (6.32, 7.54)	0.72 (0.51, 0.88)	1.08 (0.93, 1.23)	1.02 (0.90, 1.13)
Automated	A	0.14 (0.02, 0.28)	0.53 (0.44, 0.62)	0.68 (0.60, 0.77)	5.15 (4.53, 6.21)	6.00 (5.17, 6.79)	10.42 (6.24, 13.48)	2.65 (1.70, 3.60)	1.65 (1.23, 2.04)	2.82 (0.66, 4.30)

Note.—Data are accuracies, with 95% confidence intervals in parentheses. Ideal values for Dice similarity coefficient per detected lesion, maximum symmetric surface distance, and average symmetric surface distance are 1, 0 mm, and 0 mm, respectively. A = automated segmentation, C^1 = corrections of automated segmentation by reader 1, C^2 = corrections of automated segmentation by reader 2, M^1 = manual segmentations by reader 1, M^2 = manual segmentations by reader 2.

Table 4.7 Lesion volume: intrareader, interreader, and intermethod agreement at minimum overlap > 0 using Bland-Altman analyses. A = automated segmentation, C = user-corrected segmentation, C^i = user-corrected segmentation by reader i, M = manual segmentation, M^i = manual segmentations by reader i, M_j = j^{th} manual segmentation by both readers. * Data are means ± 95% confidence intervals with 95% limits of agreement in parentheses. † Data are coefficients of repeatability ± 95% confidence intervals, with 99.9% confidence intervals in parentheses.

Parameter	Readers	Mean Bias*[23]	*P* Value	Coefficient of Repeatability†
Intrareader				
Manual	M_1, M_2	0.00 ± 0.06 (-0.56, 0.55)	.882	0.56 ± 0.07
User-corrected	C_1, C_2	0.04 ± 0.08 (−0.71, 0.78)	.359	0.74 ± 0.10
Interreader				
Manual	M^1, M^2	−0.10 ± 0.07 (−0.71, 0.51)	.003	0.61 ± 0.09
User-corrected	C^1, C^2	−0.10 ± 0.08 (−0.83, 0.63)	.018	0.73 ± 0.10
Intermethod				
Automated, manual	M, A	0.57 ± 0.09 (−0.31, 1.44)	<.001	0.88 ± 0.18
Automated, user-corrected	C, A	0.28 ± 0.12 (−0.66, 1.22)	<.001	0.94 ± 0.17
Manual, user-corrected	M, C	0.33 ± 0.09 (-0.49, 1.15)	<.001	0.82 ± 0.10

Note.—A = automated segmentation, C = user-corrected segmentation, C^i = corrections of automated segmentation by reader i, C_j = j^{th} corrected automated segmentation by both readers, M = manual segmentation, M_j = j^{th} manual segmentations by both readers.
* Data are means ± 95% confidence intervals, with 95% limits of agreement in parentheses.
† Data are coefficients of repeatability ± 95% confidence intervals, with 99.9% confidence intervals in parentheses.

($\kappa = 0.46$) appeared similar to inter-reader agreement. The lower kappa for the former was due to a significantly higher quantity disagreement (0.24 vs 0.05). User correction resolved the quantity disagreement by correcting missed lesion detections.

4.3.6.4 Repeatability and reproducibility

Intra-reader repeatability and inter-reader and inter-method reproducibility on lesion volume estimation calculated by Bland-Altman analyses are summarized in Table 4.7 and Table A.6 (Appendix A.2). Inter-method variation was higher than that of manual segmentation, as shown by significantly ($P < 0.001$) higher repeatability coefficients. Both intra-reader and inter-reader coefficients of repeatability were significantly ($P < 0.001$) higher for corrections of automated segmentations than for manual segmentations. The Bland-Altman plots in Figure A.1 [a, c] (Appendix A.2) revealed a wider volume dispersion for small lesions, especially for manual segmentation.

Substantial statistically significant biases were observed with inter-method analyses ($P < 0.001$) with the largest bias (0.57 ± 0.09) observed when comparing automated segmentations to manual and a smaller bias observed when comparing automated segmentations to corrected automated segmentations (0.28 ± 0.12) or corrected to manual (0.33 ± 0.09). As can be seen in Figure A.1 [f] (Appendix A.2), corrections were mostly either addition of lesions missed by the automated method or corrections of under-segmented lesions. These corrections were not sufficient to remove systematic bias compared to manual segmentations. Inter-reader bias was small in magnitude, at -0.10 ± 0.07 ($P = 0.003$) for manual segmentation and 0.04 ± 0.08 ($P = 0.018$) for corrections of automated segmentations.

4.3.6.5 Efficiency

Mean interaction time was 7.7 ± 2.4 minutes per case for manual segmentation and 4.8 ± 2.1 minutes per case for user-corrected segmentation, with automated run time at around 1 second. Interaction time was significantly shorter for user-corrected and fully automated segmentation than for manual segmentation ($P < 0.001$).

4.3.7 Discussion

Machine learning has several use cases in the clinical workflow, such as for triage, replacement, or add-on tool to augment the work of a radiologist [197]. In our study, we found that deep learning with a pair of convolutional neural networks could be used for lesion detection and segmentation in patients with colorectal liver metastases on contrast-enhanced

CT; however, automated results are not yet at the level of trained annotators with a training set of 261 lesions annotated from CT. We further explored manual correction of automated segmentation results, thus achieving similar per-patient detection performance as entirely manual segmentation, in less time, and higher than fully automated segmentation. Overall, sensitivity and PPV increased for larger lesion size (i.e. from < 10 mm, 10-20 mm to > 20 mm). Using a fully automated lesion detection method, the per-lesion sensitivity was low for lesions < 10 mm, moderate for lesions 10-20 mm and performed well, approaching manual and user-corrected performance, for lesions > 20 mm. A prior study and meta-analysis reporting the per-lesion sensitivity of manual CLM detection on CT and using histopathology as the reference standard reported overall sensitivities ranging from 74 to 81% but also found lower sensitivities of 8 to 55% for detection of small CLMs ≤ 10 mm [198, 199]. In addition, a meta-analysis on CLM detected by imaging commented that the reference standard in several studies tended to be suboptimal because small metastases were excluded from analysis which led to inflation of detection rates [200]. A CNN-based fully automated method achieved reported sensitivity of 86% of liver metastasis detection, but lesion size was not reported [201].

In our study, we found that a user-corrected segmentation method improved segmentation performance when compared to automated segmentation, especially for small lesions < 10 mm and to a lesser extent for lesions 10-20 mm and > 20 mm. Further, user-corrected segmentations achieved a performance similar to that of manual segmentation. Our results for automated segmentation of lesions > 20 mm were within the Dice coefficient range of 65-94% reported in previous studies using fully automated methods [202–204]. Lower segmentation accuracy for small lesions observed in our study was consistent with prior studies, including one using an FCN-based algorithm for lesion segmentation [203, 204]. Except for studies using the 3D Image Reconstruction for Comparison of Algorithm Database (3Dircadb) [187], lesion size is often not reported in datasets, an approach consistent with the RECIST policy of excluding lesions under 10 mm [204, 205].

Intra-reader reliability of lesion detection was substantial and identical for manual and user-corrected segmentation. Inter-reader reliability for user-corrected was substantial and higher than for manual segmentation. Inter-reader reliability for manual segmentation was moderate and similar to reported agreement of CLM detection on CT between five observers of different experience described previously [206], and the reported disagreement for segmentation between 5 radiologists on MIDAS dataset of 9.8% [205].

Intra- and inter-reader agreement of lesion volume estimation was better for manual segmentation than user-corrected automated segmentation. The inter-method agreement was

lower between manual and either automated or user corrected segmentation than inter-rater agreement for manual segmentation. Similarly, segmentation scores were improved from automated segmentation after user correction but remained below that of manual segmentation. The automated method is repeatable by design; however, automated segmentation introduced a substantial proportional bias in segmentation volume, tending to under-segment lesions. Although user correction successfully resolved deficiencies in lesion detection by the automated method, it reduced but failed to remove this segmentation bias. This may be because user correction is biased by the segmentations being corrected. It is thus more important to reduce segmentation bias in the automated segmentation method than to improve detection, when user correction is available.

The overall user interaction time was significantly reduced (approximately by half) by correction of automated predictions compared to manual segmentation, although user correction of the predicted lesions remained the most time-consuming step in the proposed method. When the method was used as fully automated and left uncorrected, segmentation time was further reduced to under one second. Our per-case runtime for user-corrected segmentation method (4.8 min per case) was shorter than a fully automated method reported by Moghbel et al (16 min per case) [205]. Lesion detection and segmentation being a necessary but often time-consuming step in a clinical setting, a 38% improvement in efficiency would facilitate the clinical workflow. The trade-off for faster segmentation is usually losses in repeatability and accuracy [207].

Our study has limitations. First, all patients in the training and testing datasets had liver lesions. The lack of subjects without hepatic lesions may have biased the performance of the model towards pathological livers. Future studies are needed to evaluate the model on patients with a spectrum of liver disease from healthy to steatotic liver parenchyma. Our detection performance and segmentation accuracy were calculated based on manual segmentation as the reference standard, which may have lead to variability. However, we have assessed inter-reader agreement and repeatability to demonstrate the strength of our method in comparison to manual segmentation. Finally, the training set comprised of only 261 lesions from 115 patient CT scans, which is an order of magnitude less compared to other deep learning applications in radiology, which includes thousands of images.

4.3.7.1 Conclusion

In conclusion, a deep learning method shows promise for facilitating detection and segmentation of colorectal liver metastases. User correction of automated segmentations can generally resolve deficiencies of fully automated segmentation for small metastases and is faster than

manual segmentation.

4.4 Robustness of automated methods

Automated liver lesion segmentation methods appear to frequently fail on small lesions. Furthermore, while the quality of automatic segmentation on large lesions approaches the level of expert operators, there is room for improvement. Similar results were found for automated brain tumour segmentation in the BRATS challenge [208]. Automated methods must perform much better in order to be clinically useful without correction. Until then, interactive methods may remain preferable in practice. It appears that the robustness of automated methods may be improved by training models on more data, in order for the training data to adequately capture all the variabilities to which we want the models to be robust. However, collecting many expert segmentations is time consuming and often impractical. For this reason, a semi-supervised approach that makes use of weakly labeled data in addition to fully labeled data is presented in Chapter 6.

CHAPTER 5 GRADIENT FLOW IN DEEP ARTIFICIAL NEURAL NETWORKS

While evaluating fully convolutional networks (FCN) for segmentation, the issue of vanishing gradients that is endemic to deep networks was investigated. The first section of this chapter presents an analysis on the gradient flow in a network based on the U-Net [19] and proposes basic tweaks to make sure that all layers receive enough of an error signal to be adequately trained. The next section further explores orthogonality constraints on neural network weights. While such a constraint allows linear networks to maintain gradient norm and nonlinear networks to prevent gradient explosion and reduce vanishing gradients, we found that deviating from the constraint can yield faster convergence and better model performance. This analysis on orthogonality was performed on recurrent neural networks (RNN) instead of convolutional neural networks (CNN) or on FCNs because the simplicity of RNNs makes them a great initial test bed and proof of concept.

5.1 Gradient flow in fully convolutional networks for segmentation

This section reproduces in part the publication in [90] with permission from all co-authors. My contributions to this paper were the main ideas, as well as the majority of the experimental design, code, experiments, and writing. In this paper, we study the influence of both long and short skip connections on Fully Convolutional Networks (FCN) for biomedical image segmentation. In standard FCNs, only long skip connections are used to skip features from the contracting path to the expanding path in order to recover spatial information lost during downsampling. We extend FCNs by adding short skip connections, that are similar to the ones introduced in residual networks, in order to build very deep FCNs (of hundreds of layers). A review of the gradient flow confirms that for a very deep FCN it is beneficial to have both long and short skip connections. Finally, we show that a very deep FCN can achieve near-to-state-of-the-art results on the EM dataset without any further post-processing.

5.1.1 Introduction

Semantic segmentation in medical image analysis is dominated by fully convolutional networks (FCN) [3]. For example, for the EM ISBI 2012 dataset [82], BRATS [72] or MS lesions [209], the top entries are built on CNNs [19, 77, 80, 210]. Fully convolutional networks extend convolutional neural networks (CNN) to segmentation. While CNNs are typically realized by a contracting path built from convolutional, pooling and fully connected layers, FCN adds an expanding path built with deconvolutional or unpooling layers. The expanding path recovers spatial information by merging features skipped from the various resolution levels on the contracting path.

Variants of these skip connections are proposed in the literature. In [3], upsampled feature maps are summed with feature maps skipped from the contractive path while [19] concatenate them and add convolutions and non-linearities between each upsampling step. These skip connections have been shown to help recover the full spatial resolution at the network output, making fully convolutional methods suitable for semantic segmentation. We refer to these skip connections as long skip connections.

A concern not addressed by long skip connections is whether all layers deep in the FCN are updated. Significant network depth has been shown to be helpful for image classification [28, 129, 211, 212]. However, network depth is limited by the issue of vanishing gradients when backpropagating the signal across many layers. In [211], this problem is addressed with additional levels of supervision, while in [28, 129] skip connections are added around non-linearities, thus creating shortcuts through which the gradient can flow uninterrupted allowing parameters to be updated deep in the network. Moreover, [213] have shown that these skip connections allow for faster convergence during training. We refer to these skip connections as short skip connections.

In this work, we explore deep, fully convolutional networks for semantic segmentation. We expand FCN by adding short skip connections that allow us to build very deep FCNs. With this setup, we perform an analysis of short and long skip connections on a standard biomedical dataset (EM ISBI 2012 challenge data). We observe that short skip connections speed up the convergence of the learning process; moreover, we show that a very deep architecture with a relatively small number of parameters can reach near-state-of-the-art performance on this dataset.

5.1.2 Residual network for semantic image segmentation

Our approach augments fully convolutional networks by adding short skip connections (Figure 5.1(a)). We perform spatial reduction along the contracting path (left) and expansion along the expanding path (right). The contracting path is inspired by Residual Neural Networks [129]. As in [3] and [19], spatial information lost along the contracting path is recovered in the expanding (upsampling) path by skipping equal resolution features from the former to the latter. Similarly to the short skip connections in Residual Networks, we choose to sum the features on the expanding path with those skipped over the long skip connections.

We consider three types of blocks, each containing at least one convolution and activation function: bottleneck, basic block, simple block (Figure 5.1(b)-5.1(d)). Each block is capable of performing batch normalization on its inputs as well as spatial downsampling at the input (marked blue; used for the contracting path) and spatial upsampling at the output (marked yellow; for the expanding path). The bottleneck and basic block are based on those introduced in [129] which include short skip connections to skip the block input to its output with minimal modification, encouraging the path through the non-linearities to learn a residual representation of the input data. To minimize the modification of the input, we apply no transformations along the short skip connections, except when the number of filters or the spatial resolution needs to be adjusted to match the block output. We use 1×1 convolutions to adjust the number of filters but for spatial adjustment we rely on simple decimation or simple repetition of rows and columns of the input so as not to increase the number of parameters. We add an optional dropout layer to all blocks along the residual path.

We experimented with both binary cross-entropy and dice loss functions. Let $o_i \in [0, 1]$ be the i^{th} output of the last network layer passed through a sigmoid non-linearity and let $y_i \in \{0, 1\}$ be the corresponding label. The binary cross-entropy is then defined as follows:

$$L_{bce} = \sum_i y_i \log o_i + (1 - y_i) \log (1 - o_i) \tag{5.1}$$

The dice loss is:

$$L_{Dice} = -\frac{2 \sum_i o_i y_i}{\sum_i o_i + \sum_i y_i} \tag{5.2}$$

We implemented the model in Keras [214] using the Theano backend [215] and trained it using RMSprop[214] (learning rate 0.001) with weight decay set to 0.001. We also experimented with various levels of dropout.

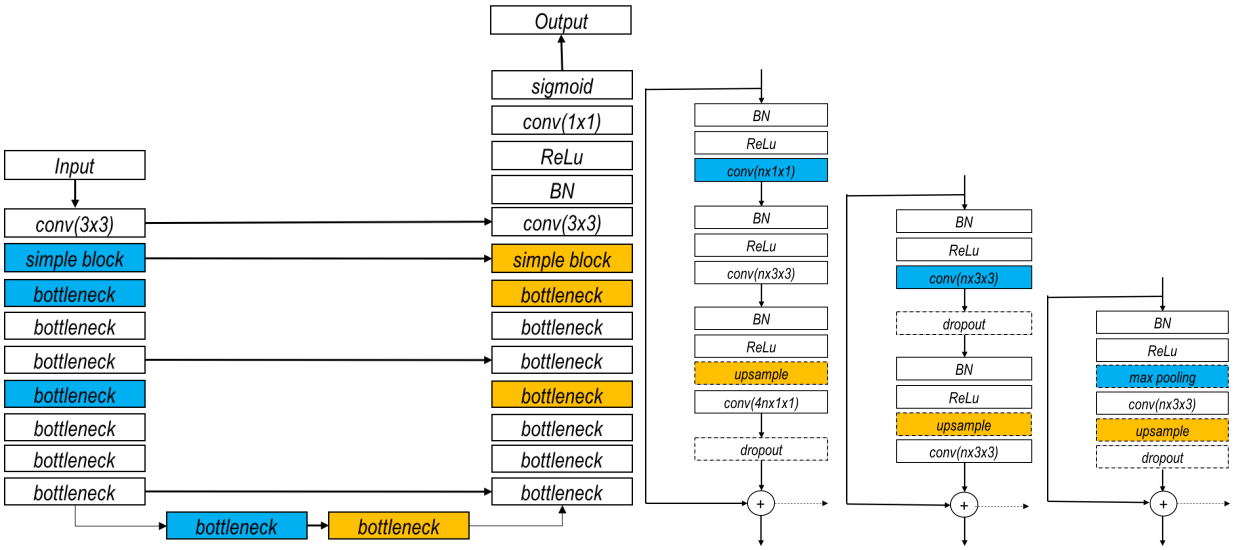


Figure 5.1 An example of residual network for image segmentation. (a) Residual Network with long skip connections built from bottleneck blocks, (b) bottleneck block, (c) basic block and (d) simple block. Blue color indicates the blocks where an downsampling is optionally performed, yellow color depicts the (optional) upsampling blocks, dashed arrow in figures (b), (c) and (d) indicates possible long skip connections. Note that all blocks (b), (c) and (d) can have a dropout layer (depicted with dashed line rectangle).

5.1.3 Experiments

In this section, we test the model on electron microscopy (EM) data [82] (Section 5.1.3.1) and perform an analysis on the importance of the long and short skip connections (Section 5.1.3.2).

5.1.3.1 Segmenting EM data

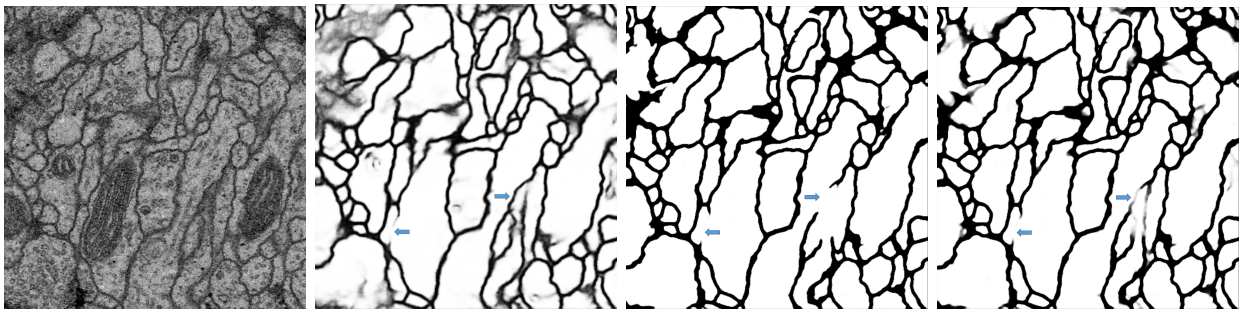
EM training data consist of 30 images (512×512 pixels) assembled from serial section transmission electron microscopy of the Drosophila first instar larva ventral nerve cord. The test set is another set of 30 images for which labels are not provided. Throughout the experiments, we used 25 images for training, leaving 5 images for validation.

During training, we augmented the input data using random flipping, sheering, rotations, and spline warping. We used the same spline warping strategy as [19]. We used full resolution (512×512) images as input without applying random cropping for data augmentation. For each training run, the model version with the best validation loss was stored and evaluated. The detailed description of the highest performing architecture used in the experiments is shown in Table 5.1.

Table 5.1 Detailed model architecture used in the experiments. Repetition number indicates the number of times the block is repeated.

Layer name	block type	output resolution	output width	repetition number
Down 1	conv 3×3	512×512	32	1
Down 2	simple block	256×256	32	1
Down 3	bottleneck	128×128	128	3
Down 4	bottleneck	64×64	256	8
Down 5	bottleneck	32×32	512	10
Across	bottleneck	32×32	1024	3
Up 1	bottleneck	64×64	512	10
Up 2	bottleneck	128×128	256	8
Up 3	bottleneck	256×256	128	3
Up 4	simple block	512×512	32	1
Up 5	conv 3x3	512×512	32	1
Classifier	conv 1×1	512×512	1	1

Interestingly, we found that while the predictions from models trained with cross-entropy loss were of high quality, those produced by models trained with the Dice loss appeared visually cleaner since they were almost binary (similar observations were reported in a parallel work [91].); borders that would appear fuzzy in the former (see Figure 5.2(b)) would be left as gaps in the latter (Figure 5.2(c)). However, we found that the border continuity can be improved for models with the Dice loss by implicit model averaging over output samples drawn at test time, using dropout [216] (Figure 5.2(d)). This yields better performance on the validation and test metrics than the output of models trained with binary cross-entropy (see Table 5.2).



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

Figure 5.2 Qualitative results on the test set. (a) original image, (b) prediction for a model trained with binary cross-entropy, (c) prediction of the model trained with dice loss and (d) model trained with dice loss with 0.2 dropout at the test time.

Two metrics used in this dataset are: Maximal foreground-restricted Rand score after thinning (V_{rand}) and maximal foreground-restricted information theoretic score after thinning (V_{info}). For a detailed description of the metrics, please refer to [82].

Table 5.2 Comparison to published entries for EM dataset. For full ranking of all submitted methods please refer to challenge web page: `http://brainiac2.mit.edu/isbi_challenge/leaders-board-new`. We note the number of parameter, the use of post-processing, and the use of model averaging only for FCNs.

Method	V_{rand}	V_{info}	FCN	post-processing	average over	parameters (M)
CUMedVision [80]	0.977	0.989	YES	YES	6	8
Unet [19]	0.973	0.987	YES	NO	7	33
IDSIA [217]	0.970	0.985	NO	-	-	-
motif [218]	0.972	0.985	NO	-	-	-
SCI [219]	0.971	0.982	NO	-	-	-
optree-idsia[220]	0.970	0.985	NO	-	-	-
PyraMiD-LSTM[4]	0.968	0.983	NO	-	-	-
Ours (L_{Dice})	0.969	0.986	YES	NO	Dropout	11
Ours (L_{bce})	0.957	0.980	YES	NO	1	11

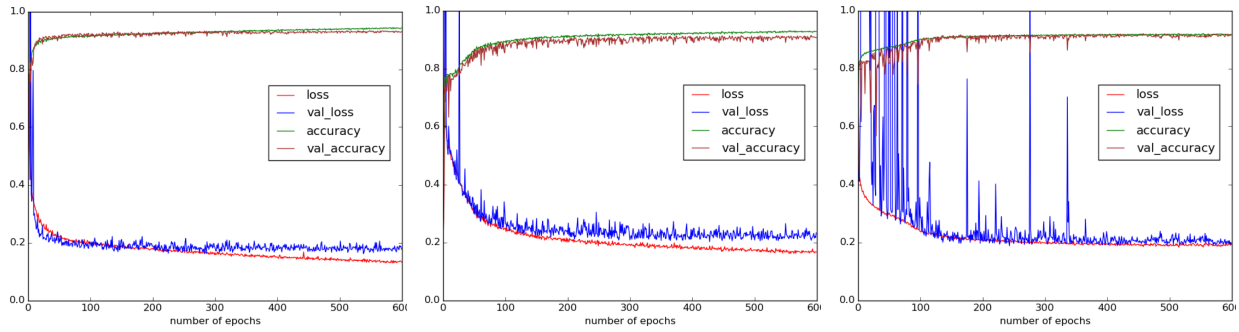
Our results are comparable to other published results that establish the state of the art for the EM dataset (Table 5.2). Note that we did not do any post-processing of the resulting segmentations. We match the performance of UNet, for which predictions are averaged over seven rotations of the input images, while using less parameters and without sophisticated class weighting. Note that among other FCN available on the leader board, CUMedVision is using post-processing in order to boost performance.

5.1.3.2 On the importance of skip connections

The focus in the paper is to evaluate the utility of long and short skip connections for training fully convolutional networks for image segmentation. In this section, we investigate the learning behavior of the model with short and with long skip connections, paying specific attention to parameter updates at each layer of the network. We first explored variants of our best performing deep architecture (from Table 5.1), using binary cross-entropy loss. Maintaining the same hyperparameters, we trained (Model 1) with long and short skip connections, (Model 2) with only short skip connections and (Model 3) with only long skip connections. Training curves are presented in Figure 5.3 and the final loss and accuracy values on the training and the validation data are presented in Table 5.3.

We note that for our deep architecture, the variant with both long and short skip connections is not only the one that performs best but also converges faster than without short skip connections. This increase in convergence speed is consistent with the literature [213]. Not surprisingly, the combination of both long and short skip connections performed better than having only one type of skip connection, both in terms of performance and convergence speed. At this depth, a network could not be trained without any skip connections. Finally, short

skip connections appear to stabilize updates (note the smoothness of the validation loss plots in Figures 5.3(a) and 5.3(b) as compared to Figure 5.3(c)).



(a) (b) (c)

Figure 5.3 Training and validation losses and accuracies for different network setups: (a) Model 1: long and short skip connections enabled, (b) Model 2: only short skip connections enabled and (c) Model 3: only long skip connections enabled.

Table 5.3 Best validation loss and its corresponding training loss for each model.

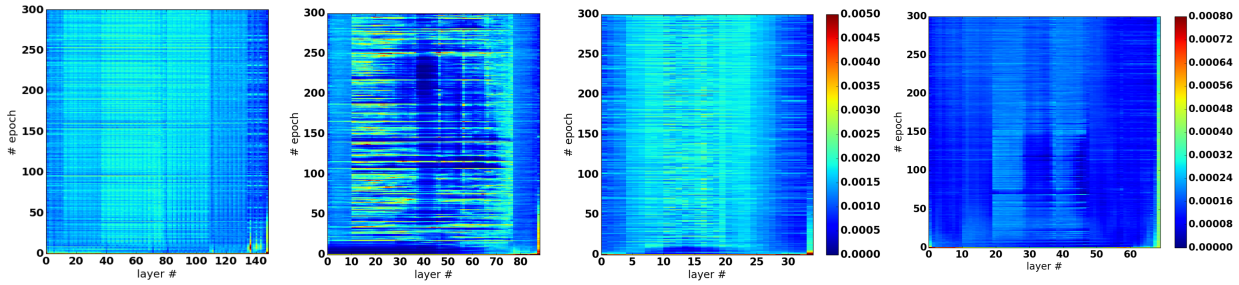
Method	training loss	validation loss
Long and short skip connections	0.163	0.162
Only short skip connections	0.188	0.202
Only long skip connection	0.205	0.188

We expect that layers closer to the center of the model can not be effectively updated due to the vanishing gradient problem which is alleviated by short skip connections. This identity shortcut effectively introduces shorter paths through fewer non-linearities to the deep layers of our models. We validate this empirically on a range of models of varying depth by visualizing the mean model parameter updates at each layer for each epoch (see sample results in Figure 5.4). To simplify the analysis and visualization, we used simple blocks instead of bottleneck blocks.

Parameter updates appear to be well distributed when short skip connections are present (Figure 5.4(a)). When the short skip connections are removed, we find that for deep models, the deep parts of the network (at the center, Figure 5.4(b)) get few updates, as expected. When long skip connections are retained, at least the shallow parts of the model can be updated (see both sides of Figure 5.4(b)) as these connections provide shortcuts for gradient flow. Interestingly, we observed that model performance actually drops when using short skip connections in those models that are shallow enough for all layers to be well updated (eg. Figure 5.4(c)). Moreover, batch normalization was observed to increase the maximal updatable depth of the network. Networks without batch normalization had diminishing

updates toward the center of the network and with long skip connections were less stable, requiring a lower learning rate (eg. Figure 5.4(d)).

It is also interesting to observe that the bulk of updates in all tested model variations (also visible in those shown in Figure 5.4) were always initially near or at the classification layer. This follows the findings of [221], where it is shown that even randomly initialized weights can confer a surprisingly large portion of a model's performance after training only the classifier.



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

Figure 5.4 Weight updates in different network setups: (a) the best performing model with long and short skip connections enabled, (b) only long skip connections enabled with 9 repetitions of simple block, (c) only long skip connections enabled with 3 repetitions of simple block and (d) only long skip connections enabled with 7 repetitions of simple block, without batch normalization. Note that due to a reduction in the learning rate for Figure (d), the scale is different compared to Figures (a), (b) and (c).

5.1.4 Conclusions

In this paper, we studied the influence of skip connections on FCN for biomedical image segmentation. We showed that a very deep network can achieve results near the state of the art on the EM dataset without any further post-processing. We confirm that although long skip connections provide a shortcut for gradient flow in shallow layers, they do not alleviate the vanishing gradient problem in deep networks. Consequently, we apply short skip connections to FCNs and confirm that this increases convergence speed and allows training of very deep networks. On the other hand, the performance of shallow networks may be reduced with short skip connections.

5.2 (Article 3) On orthogonality and learning RNNs with long term dependencies

This paper explores an orthogonality constraint on neural network weights that is intended to diminish the issue of vanishing and exploding gradients. This paper does not propose such a constraint as novel since it had been proposed shortly before this work and instead focuses on the analysis of relaxing the orthogonality constraint. It follows the hypothesis that nearly orthogonal weights allow for better model performance than fully orthogonal weights which are less expressive. This hypothesis is confirmed by the analysis and alternative explanations for these phenomena are proposed in the discussion in Chapter 7.2.

Title

On orthogonality and learning RNNs with long term dependencies

Authors

Eugene Vorontsov[1,2], Chiheb Trabelsi[1,2], Samuel Kadoury[2,3], Chris Pal[1,2]

Affiliations

[1] École Polytechnique de Montréal

[2] Montreal Institute for Learning Algorithms (MILA)

[3] Centre de Recherche du Centre hospitalier de l'Université de Montréal (CRCHUM)

Publication

This paper has been published in the *Proceedings of the 34th International Conference on Machine Learning* on August 6, 2017, Volume 70, pp. 3570-3578.

Contribution

My contributions to this work were the main ideas, as well as most of the code, experiments, and writing.

5.2.1 Abstract

It is well known that it is challenging to train deep neural networks and recurrent neural networks for tasks that exhibit long term dependencies. The vanishing or exploding gradient problem is a well known issue associated with these challenges. One approach to addressing vanishing and exploding gradients is to use either soft or hard constraints on weight matrices so as to encourage or enforce orthogonality. Orthogonal matrices preserve gradient norm during backpropagation and may therefore be a desirable property. This paper explores issues with optimization convergence, speed and gradient stability when encouraging or enforcing orthogonality. To perform this analysis, we propose a weight matrix factorization and parameterization strategy through which we can bound matrix norms and therein control the degree of expansivity induced during backpropagation. We find that hard constraints on orthogonality can negatively affect the speed of convergence and model performance.

5.2.2 Introduction

The depth of deep neural networks confers representational power, but also makes model optimization more challenging. Training deep networks with gradient descent based methods is known to be difficult as a consequence of the vanishing and exploding gradient problem [127]. Typically, exploding gradients are avoided by clipping large gradients [32] or introducing an L_2 or L_1 weight norm penalty. The latter has the effect of bounding the spectral radius of the linear transformations, thus limiting the maximal gain across the transformation. [31] attempt to stabilize the norm of propagating signals directly by penalizing differences in successive norm pairs in the forward pass and [32] propose to penalize successive gradient norm pairs in the backward pass. These regularizers affect the network parameterization with respect to the data instead of penalizing weights directly.

Both expansivity and contractivity of linear transformations can also be limited by more tightly bounding their spectra. By limiting the transformations to be orthogonal, their singular spectra are limited to unitary gain causing the transformations to be norm-preserving. [30] and [222] have respectively shown that identity initialization and orthogonal initialization can be beneficial. [223] have gone beyond initialization, building unitary recurrent neural network (RNN) models with transformations that are unitary by construction which they achieved by composing multiple basic unitary transformations. The resulting transformations, for some n-dimensional input, cover only some subset of possible $n \times n$ unitary matrices but appear to perform well on simple tasks and have the benefit of having low complexity in memory and computation. Similarly, [224] introduce an efficient algorithm to cover a large subset.

The entire set of possible unitary or orthogonal parameterizations forms the Stiefel manifold. At a much higher computational cost, gradient descent optimization directly along this manifold can be done via geodesic steps [225, 226]. Recent work [227] has proposed the optimization of unitary matrices along the Stiefel manifold using geodesic gradient descent. To produce a full-capacity parameterization for unitary matrices they use some insights from [226], combining the use of canonical inner products and Cayley transformations. Their experimental work indicates that full capacity unitary RNN models can solve the copy memory problem whereas both LSTM networks and restricted capacity unitary RNN models having similar complexity appear unable to solve the task for a longer sequence length ($T = 2000$). [36] and [228] introduced more computationally efficient full capacity parameterizations. [47] also find that the use of fully connected "Stiefel layers" improves the performance of some convolutional neural networks.

We seek to gain a new perspective on this line of research by exploring the optimization of real valued matrices within a configurable margin about the Stiefel manifold. We suspect that a strong constraint of orthogonality limits the model's representational power, hindering its performance, and may make optimization more difficult. We explore this hypothesis empirically by employing a factorization technique that allows us to limit the degree of deviation from the Stiefel manifold[1]. While we use geodesic gradient descent, we simultaneously update the singular spectra of our matrices along Euclidean steps, allowing optimization to step away from the manifold while still curving about it.

5.2.2.1 Vanishing and exploding gradients

The issue of vanishing and exploding gradients as it pertains to the parameterization of neural networks can be illuminated by looking at the gradient back-propagation chain through a network.

A neural network with N hidden layers has pre-activations

$$\mathbf{a}_i(\mathbf{h}_{i-1}) = \mathbf{W}_i\, \mathbf{h}_{i-1} + \ \mathbf{b}_i, \quad i \in \{2, \cdots, N-1\} \tag{5.3}$$

For notational convenience, we combine parameters $\mathbf{W}_i$ and $\mathbf{b}_i$ to form an affine matrix $\boldsymbol{\theta}$. We can see that for some loss function L at layer n, the derivative with respect to parameters $\boldsymbol{\theta}_i$ is:

$$\frac{\partial L}{\partial \boldsymbol{\theta}_i} = \frac{\partial \mathbf{a}_{n+1}}{\partial \boldsymbol{\theta}_i} \frac{\partial L}{\partial \mathbf{a}_{n+1}} \tag{5.4}$$

[1]Source code for the model and experiments located at `https://github.com/veugene/spectre_release`

The partial derivatives for the pre-activations can be decomposed as follows:

$$\begin{aligned}\frac{\partial \mathbf{a}_{i+1}}{\partial \boldsymbol{\theta}_i} &= \frac{\partial \mathbf{a}_i}{\partial \boldsymbol{\theta}_i}\frac{\partial \mathbf{h}_i}{\partial \mathbf{a}_i}\frac{\partial \mathbf{a}_{i+1}}{\partial \mathbf{h}_i} \\ &= \frac{\partial \mathbf{a}_i}{\partial \boldsymbol{\theta}_i}\,\mathbf{D}_i\mathbf{W}_{i+1} \;\rightarrow\; \frac{\partial \mathbf{a}_{i+1}}{\partial \mathbf{a}_i} = \mathbf{D}_i\mathbf{W}_{i+1},\end{aligned} \tag{5.5}$$

where $\mathbf{D_i}$ is the Jacobian corresponding to the activation function, containing partial derivatives of the hidden units at layer $i+1$ with respect to the pre-activation inputs. Typically, $\mathbf{D}$ is diagonal. Following the above, the gradient in equation 5.4 can be fully decomposed into a recursive chain of matrix products:

$$\frac{\partial L}{\partial \boldsymbol{\theta}_i} = \frac{\partial \mathbf{a}_i}{\partial \boldsymbol{\theta}_i}\prod_{j=i}^{n}(\mathbf{D}_j\mathbf{W}_{j+1})\frac{\partial L}{\partial \mathbf{a}_{n+1}} \tag{5.6}$$

In [32], it is shown that the 2-norm of $\dfrac{\partial \mathbf{a}_{i+1}}{\partial \mathbf{a}_i}$ is bounded by the product of the norms of the non-linearity's Jacobian and transition matrix at time t (layer i), as follows:

$$\begin{gathered}\left|\left|\frac{\partial \mathbf{a}_{t+1}}{\partial \mathbf{a}_t}\right|\right| \leq ||\mathbf{D}_t||\;||\mathbf{W}_t|| \leq \lambda_{\mathbf{D}_t}\;\lambda_{\mathbf{W}_t} = \eta_t, \\ \lambda_{\mathbf{D}_t}, \lambda_{\mathbf{W}_t} \in \mathbb{R}.\end{gathered} \tag{5.7}$$

where $\lambda_{\mathbf{D}_t}$ and $\lambda_{\mathbf{W}_t}$ are the largest singular values of the non-linearity's Jacobian $\mathbf{D}_t$ and the transition matrix $\mathbf{W}_t$. In RNNs, $\mathbf{W}_t$ is shared across time and can be simply denoted as $\mathbf{W}$.

Equation 5.7 shows that the gradient can grow or shrink at each layer depending on the gain of each layer's linear transformation $\mathbf{W}$ and the gain of the Jacobian $\mathbf{D}$. The gain caused by each layer is magnified across all time steps or layers. It is easy to have extreme amplification in a recurrent neural network where $\mathbf{W}$ is shared across time steps and a non-unitary gain in $\mathbf{W}$ is amplified exponentially. The phenomena of extreme growth or contraction of the gradient across time steps or layers are known as the exploding and the vanishing gradient problems, respectively. It is sufficient for RNNs to have $\eta_t \leq 1$ at each time t to enable the possibility of vanishing gradients, typically for some large number of time steps T. The rate at which a gradient (or forward signal) vanishes depends on both the parameterization of the model and on the input data. The parameterization may be conditioned by placing appropriate constraints on $\mathbf{W}$. It is worth keeping in mind that the Jacobian $\mathbf{D}$ is typically contractive, thus tending to be norm-reducing) and is also data-dependent, whereas $\mathbf{W}$ can vary from being contractive to norm-preserving, to expansive and applies the same gain on the forward signal as on the back-propagated gradient signal.

5.2.3 Our approach

Vanishing and exploding gradients can be controlled to a large extent by controlling the maximum and minimum *gain* of $\mathbf{W}$. The maximum gain of a matrix $\mathbf{W}$ is given by the spectral norm which is given by

$$||\mathbf{W}||_2 = \max\left[\frac{||\mathbf{W}\mathbf{x}||}{||\mathbf{x}||}\right]. \tag{5.8}$$

By keeping our weight matrix $\mathbf{W}$ close to orthogonal, one can ensure that it is close to a norm-preserving transformation (where the spectral norm is equal to one, but the minimum gain is also one). One way to achieve this is via a simple soft constraint or regularization term of the form:

$$\lambda \sum_i ||\mathbf{W}_i^T \mathbf{W}_i - \mathbf{I}||^2. \tag{5.9}$$

However, it is possible to formulate a more direct parameterization or factorization for $\mathbf{W}$ which permits *hard bounds* on the amount of expansion and contraction induced by $\mathbf{W}$. This can be achieved by simply parameterizing $\mathbf{W}$ according to its singular value decomposition, which consists of the composition of orthogonal basis matrices $\mathbf{U}$ and $\mathbf{V}$ with a diagonal spectral matrix $\mathbf{S}$ containing the singular values which are real and positive by definition. We have

$$\mathbf{W} = \mathbf{U}\mathbf{S}\mathbf{V}^T. \tag{5.10}$$

Since the spectral norm or maximum gain of a matrix is equal to its largest singular value, this decomposition allows us to control the maximum gain or expansivity of the weight matrix by controlling the magnitude of the largest singular value. Similarly, the minimum gain or contractivity of a matrix can be obtained from the minimum singular value.

We can keep the bases $\mathbf{U}$ and $\mathbf{V}$ orthogonal via *geodesic gradient descent* along the set of weights that satisfy $\mathbf{U}^T\mathbf{U} = \mathbf{I}$ and $\mathbf{V}^T\mathbf{V} = \mathbf{I}$ respectively. The submanifolds that satisfy these constraints are called Stiefel manifolds. We discuss how this is achieved in more detail below, then discuss our construction for bounding the singular values.

During optimization, in order to maintain the orthogonality of an orthogonally-initialized matrix $\mathbf{M}$, i.e. where $\mathbf{M} = \mathbf{U}$, $\mathbf{M} = \mathbf{V}$ or $\mathbf{M} = \mathbf{W}$ if so desired, we employ a Cayley transformation of the update step onto the Stiefel manifold of (semi-)orthogonal matrices, as in [225] and [226]. Given an orthogonally-initialized parameter matrix $\mathbf{M}$ and its Jacobian,

$\mathbf{G}$ with respect to the objective function, an update is performed as follows:

$$\begin{aligned} \mathbf{A} &= \mathbf{G}\mathbf{M}^T - \mathbf{M}\mathbf{G}^T \\ \mathbf{M}_{new} &= (\mathbf{I} + \frac{\eta}{2}\mathbf{A})^{-1}(\mathbf{I} - \frac{\eta}{2}\mathbf{A})\mathbf{M}, \end{aligned} \tag{5.11}$$

where $\mathbf{A}$ is a skew-symmetric matrix (that depends on the Jacobian and on the parameter matrix) which is mapped to an orthogonal matrix via a Cayley transform and η is the learning rate.

While the update rule in (5.11) allows us to maintain an orthogonal hidden to hidden transition matrix $\mathbf{W}$ if desired, we are interested in exploring the effect of stepping away from the Stiefel manifold. As such, we parameterize the transition matrix $\mathbf{W}$ in factorized form, as a singular value decomposition with orthogonal bases $\mathbf{U}$ and $\mathbf{V}$ updated by geodesic gradient descent using the Cayley transform approach above.

If $\mathbf{W}$ is an orthogonal matrix, the singular values in the diagonal matrix $\mathbf{S}$ are all equal to one. However, in our formulation we allow these singular values to deviate from one and employ a sigmoidal parameterization to apply a hard constraint on the maximum and minimum amount of deviation. Specifically, we define a margin m around 1 within which the singular values must lie. This is achieved with the parameterization

$$s_i = 2m(\sigma(p_i) - 0.5) + 1, \qquad s_i \in \{\text{diag}(\mathbf{S})\},\ m \in [0,\ 1]. \tag{5.12}$$

The singular values are thus restricted to the range $[1 - m,\ 1 + m]$ and the underlying parameters p_i are updated freely via stochastic gradient descent. Note that this parameterization strategy also has implications on the step sizes that gradient descent based optimization will take when updating the singular values – they tend to be smaller compared to models with no margin constraining their values. Specifically, a singular value's progression toward a margin is slowed the closer it is to the margin. The sigmoidal parameterization can also impart another effect on the step size along the spectrum which needs to be accounted for. Considering 5.12, the gradient backpropagation of some loss L toward parameters p_i is found as

$$\frac{dL}{dp_i} = \frac{ds_i}{dp_i}\frac{dL}{ds_i} = 2m\frac{d\sigma(p_i)}{dp_i}\frac{dL}{ds_i}. \tag{5.13}$$

From (5.13), it can be seen that the magnitude of the update step for p_i is scaled by the margin hyperparameter m. This means for example that for margins less than one, the effective learning rate for the spectrum is reduced in proportion to the margin. Consequently, we adjust the learning rate along the spectrum to be independent of the margin by renormalizing

it by $2m$.

This margin formulation both guarantees singular values lie within a well defined range and slows deviation from orthogonality. Alternatively, one could enforce the orthogonality of $\mathbf{U}$ and $\mathbf{V}$ and impose a regularization term corresponding to a mean one Gaussian prior on these singular values. This encourages the weight matrix $\mathbf{W}$ to be norm preserving with a controllable strength equivalent to the variance of the Gaussian. We also explore this approach further below.

5.2.4 Experiments

In this section, we explore hard and soft orthogonality constraints on factorized weight matrices for recurrent neural network hidden to hidden transitions. With hard orthogonality constraints on $\mathbf{U}$ and $\mathbf{V}$, we investigate the effect of widening the spectral margin or bounds on convergence and performance. Loosening these bounds allows increasingly larger margins within which the transition matrix $\mathbf{W}$ can deviate from orthogonality. We confirm that orthogonal initialization is useful as noted in [222], and we show that although strict orthogonality guarantees stable gradient norm, loosening orthogonality constraints can increase the rate of gradient descent convergence. We begin our analyses on tasks that are designed to stress memory: a sequence copying task and a basic addition task [127]. We then move on to tasks on real data that require models to capture long-range dependencies: digit classification based on sequential and permuted MNIST vectors [30, 229]. Finally, we look at a basic language modeling task using the Penn Treebank dataset [230].

The copy and adding tasks, introduced by [127], are synthetic benchmarks with pathologically hard long distance dependencies that require long-term memory in models. The copy task consists of an input sequence that must be remembered by the network, followed by a series of blank inputs terminated by a delimiter that denotes the point at which the network must begin to output a copy of the initial sequence. We use an input sequence of $T + 20$ elements that begins with a sub-sequence of 10 elements to copy, each containing a symbol $a_i \in \{a_1, ..., a_p\}$ out of $p = 8$ possible symbols. This sub-sequence is followed by $T - 1$ elements of the blank category a_0 which is terminated at step T by a delimiter symbol a_{p+1} and 10 more elements of the blank category. The network must learn to remember the initial 10 element sequence for T time steps and output it after receiving the delimiter symbol.

The goal of the adding task is to add two numbers together after a long delay. Each number is randomly picked at a unique position in a sequence of length T. The sequence is composed of T values sampled from a uniform distribution in the range $[0, 1)$, with each value paired with an indicator value that identifies the value as one of the two numbers to remember

(marked 1) or as a value to ignore (marked 0). The two numbers are positioned randomly in the sequence, the first in the range $[0, \frac{T}{2} - 1]$ and the second in the range $[\frac{T}{2}, T - 1]$, where 0 marks the first element. The network must learn to identify and remember the two numbers and output their sum.

In the sequential MNIST task from [30], MNIST digits are flattened into vectors that can be traversed sequentially by a recurrent neural network. The goal is to classify the digit based on the sequential input of pixels. The simple variant of this task is with a simple flattening of the image matrices; the harder variant of this task includes a random permutation of the pixels in the input vector that is determined once for an experiment. The latter formulation introduces longer distance dependencies between pixels that must be interpreted by the classification model.

The English Penn Treebank (PTB) dataset from [230] is an annotated corpus of English sentences, commonly used for benchmarking language models. We employ a sequential character prediction task: given a sentence, a recurrent neural network must predict the next character at each step, from left to right. We use input sequences of variable length, with each sequence containing one sentence. We model 49 characters including lowercase letters (all strings are in lowercase), numbers, common punctuation, and an unknown character placeholder. We use two subsets of the data in our experiments: in the first, we first use 23% of the data with strings with up to 75 characters and in the second we include over 99% of the dataset, picking strings with up to 300 characters.

5.2.4.1 Loosening hard orthogonality constraints

In this section, we experimentally explore the effect of loosening hard orthogonality constraints through loosening the spectral margin defined above for the hidden to hidden transition matrix.

In all experiments, we employed RMSprop [166] when not using geodesic gradient descent. We used minibatches of size 50 and for generated data (the copy and adding tasks), we assumed an epoch length of 100 minibatches. We cautiously introduced gradient clipping at magnitude 100 (unless stated otherwise) in all of our RNN experiments although it may not be required and we consistently applied a small weight decay of 0.0001. Unless otherwise specified, we trained all simple recurrent neural networks with the hidden to hidden matrix factorization as in (5.10) using geodesic gradient descent on the bases (learning rate 10^{-6}) and RMSprop on the other parameters (learning rate 0.0001), using a tanh transition nonlinearity, and clipping gradients of 100 magnitude. The neural network code was built on the Theano framework [215]. When parameterizing a matrix in factorized form, we apply the weight decay on the

composite matrix rather than on the factors in order to be consistent across experiments. For MNIST and PTB, hyperparameter selection and early stopping were performed targeting the best validation set accuracy, with results reported on the test set.

5.2.4.1.1 Convergence on synthetic memory tasks For different sequence lengths T of the copy and adding tasks, we trained a factorized RNN with 128 hidden units and various spectral margins m. For the copy task, we used Elman networks without a transition non-linearity as in [222]. We also investigated the use of nonlinearities, as discussed below.

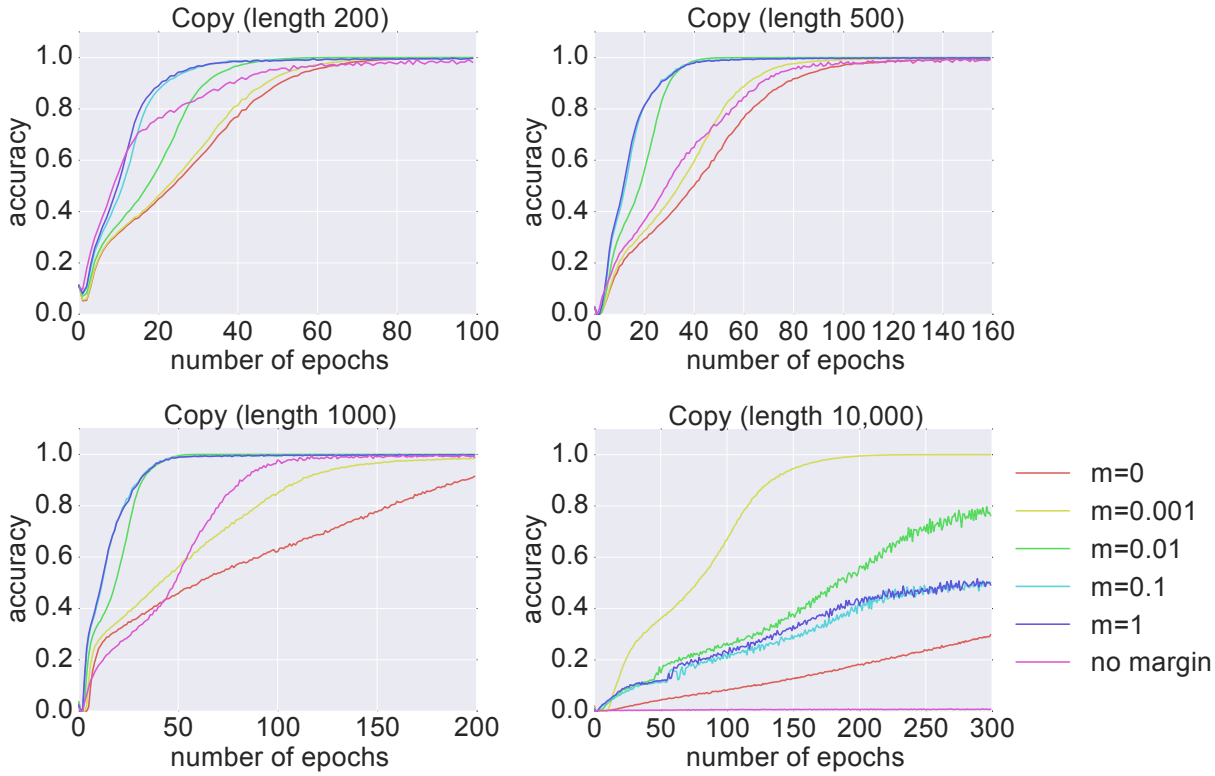


Figure 5.5 Accuracy curves on the copy task for different sequence lengths given various spectral margins. Convergence speed increases with margin size; however, large margin sizes are ineffective at longer sequence lengths (T=10000, right).

As shown in Figure 5.5 we see an increase in the rate of convergence as we increase the spectral margin. This observation generally holds across the tested sequence lengths ($T = 200$, $T = 500$, $T = 1000$, $T = 10000$); however, large spectral margins hinder convergence on extremely long sequence lengths. At sequence length $T = 10000$, parameterizations with spectral margins larger than 0.001 converge slower than when using a margin of 0.001. In addition, the experiment without a margin failed to converge on the longest sequence length. This follows the expected pattern where stepping away from the Stiefel manifold may help

with gradient descent optimization but loosening orthogonality constraints can reduce the stability of signal propagation through the network.

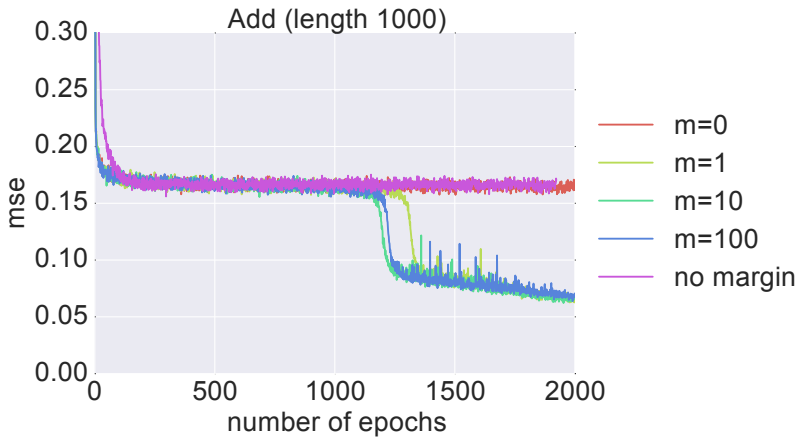


Figure 5.6 Mean squared error (MSE) curves on the adding task for different spectral margins m. A trivial solution of always outputting the same number has an expected baseline MSE of 0.167.

For the adding task, we trained a factorized RNN on $T = 1000$ length sequences, using a ReLU activation function on the hidden to hidden transition matrix. The mean squared error (MSE) is shown for different spectral margins in Figure 5.6. Testing spectral margins $m = 0$, $m = 1$, $m = 10$, $m = 100$, and no margin, we find that the models with the purely orthogonal ($m = 0$) and the unconstrained (no margin) transition matrices failed to begin converging beyond baseline MSE within 2000 epochs.

We found that nonlinearities such as a rectified linear unit (ReLU) [231] or hyperbolic tangent (tanh) made the copy task far more difficult to solve. Using tanh, a short sequence length ($T = 100$) copy task required both a soft constraint that encourages orthogonality and thousands of epochs for training. It is worth noting that in the unitary evolution recurrent neural network of [223], the non-linearity (referred to as the "modReLU") is actually initialized as an identity operation that is free to deviate from identity during training. Furthermore, [222] derive a solution mechanism for the copy task that drops the non-linearity from an RNN. To explore this further, we experimented with a parametric leaky ReLU activation function (PReLU) which introduces a trainable slope α for negative valued inputs x, producing $f(x) = max(x, 0) + \alpha min(x, 0)$ [232]. Setting the slope α to one would make the PReLU equivalent to an identity function. We experimented with clamping α to 0.5, 0.7 or 1 in a factorized RNN with a spectral margin of 0.3 and found that only the model with $\alpha = 1$ solved the $T = 1000$ length copy task. We also experimented with a trainable slope α, initialized to 0.7 and found that it converges to 0.96, further suggesting the optimal

solution for the copy task is without a transition nonlinearity. Since the copy task is purely a memory task, one may imagine that a transition nonlinearity such as a tanh or ReLU may be detrimental to the task as it can lose information. Thus, we also tried a recent activation function that preserves information, called an orthogonal permutation linear unit (OPLU) [233]. The OPLU preserves norm, making a fully norm-preserving RNN possible. Interestingly, this activation function allowed us to recover identical results on the copy task to those without a nonlinearity for different spectral margins.

5.2.4.1.2 Performance on real data Having confirmed that an orthogonality constraint can negatively impact convergence rate, we seek to investigate the effect on model performance for tasks on real data. In Table 5.4, we show the results of experiments on ordered and permuted sequential MNIST classification tasks and on the PTB character prediction task.

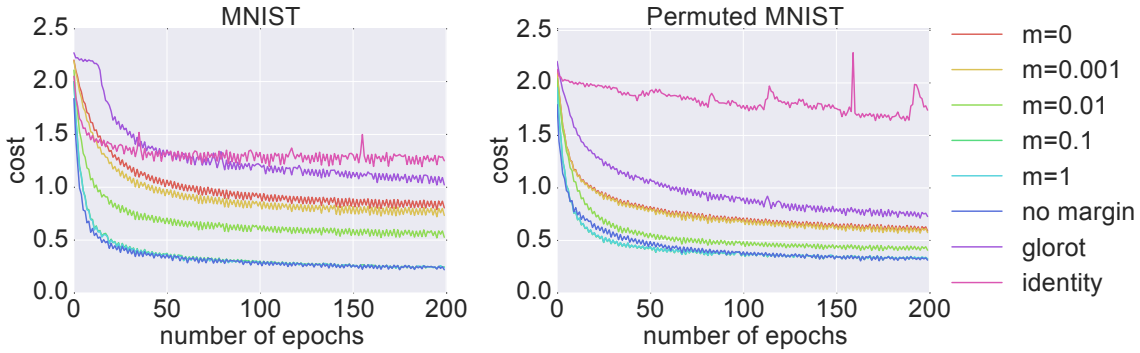


Figure 5.7 Loss curves for different factorized RNN parameterizations on the sequential MNIST task (left) and the permuted sequential MNIST task (right). The spectral margin is denoted by m; models with no margin have singular values that are directly optimized with no constraints; Glorot refers to a factorized RNN with no margin that is initialized with Glorot normal initialization. Identity refers to the same, with identity initialization.

For the sequential MNIST experiments, loss curves are shown in Figure 5.7 and reveal an increased convergence rate for larger spectral margins. We trained the factorized RNN models with 128 hidden units for 120 epochs. We also trained an LSTM with 128 hidden units (tanh activation) on both tasks for 150 epochs, configured with peephole connections, orthogonally initialized (and forget gate bias initialized to one), and trained with RMSprop (learning rate 0.0001, clipping gradients of magnitude 1).

For PTB character prediction, we evaluate results in terms of bits per character (bpc) and prediction accuracy. Prediction results are shown in 5.4 both for a subset of short sequences (up to 75 characters; 23% of data) and for a subset of long sequences (up to 300 characters;

Table 5.4 Performance on MNIST and PTB for different spectral margins and initializations. Evaluated on classification of sequential MNIST (MNIST) and permuted sequential MNIST (pMNIST); character prediction on PTB sentences of up to 75 characters (PTBc-75) and up to 300 characters (PTBc-300).

		MNIST	pMNIST	PTBc-75		PTBc-300	
margin	initialization	accuracy	accuracy	bpc	accuracy	bpc	accuracy
0	orthogonal	77.18	83.56	2.16	55.31	2.20	54.88
0.001	orthogonal	79.26	84.59	-	-	-	-
0.01	orthogonal	85.47	89.63	2.16	55.33	2.20	54.83
0.1	orthogonal	94.10	91.44	2.12	55.37	2.24	54.10
1	orthogonal	93.84	90.83	2.06	57.07	2.36	51.12
100	orthogonal	-	-	2.04	57.51	2.36	51.20
none	orthogonal	93.24	90.51	2.06	57.38	2.34	51.30
none	Glorot normal	66.71	79.33	2.08	57.37	2.34	51.04
none	identity	53.53	42.72	2.25	53.83	2.68	45.35
LSTM		97.30	92.62	1.92	60.84	1.64	65.53

99% of data). We trained factorized RNN models with 512 hidden units for 200 epochs with geodesic gradient descent on the bases (learning rate 10^{-6}) and RMSprop on the other parameters (learning rate 0.001), using a tanh transition nonlinearity, and clipping gradients of 30 magnitude. As a rough point of reference, we also trained an LSTM with 512 hidden units for each of the data subsets (configured as for MNIST). On sequences up to 75 characters, LSTM performance was limited by early stopping of training due to overfitting.

Interestingly, for both the ordered and permuted sequential MNIST tasks, models with a non-zero margin significantly outperform those that are constrained to have purely orthogonal transition matrices (margin of zero). The best results on both the ordered and sequential MNIST tasks were yielded by models with a spectral margin of 0.1, at 94.10% accuracy and 91.44% accuracy, respectively. An LSTM outperformed the RNNs in both tasks; nevertheless, RNNs with hidden to hidden transitions initialized as orthogonal matrices performed admirably without a memory component and without all of the additional parameters associated with gates. Indeed, orthogonally initialized RNNs performed almost on par with the LSTM in the permuted sequential MNIST task which presents longer distance dependencies than the ordered task. Although the optimal margin appears to be 0.1, RNNs with large margins perform almost identically to an RNN without a margin, as long as the transition matrix is initialized as orthogonal. On these tasks, orthogonal initialization appears to significantly outperform Glorot normal initialization [234] or initializing the matrix as identity. It is interesting to note that for the MNIST tasks, orthogonal initialization appears useful while orthogonality constraints appear mainly detrimental. This suggests that while orthog-

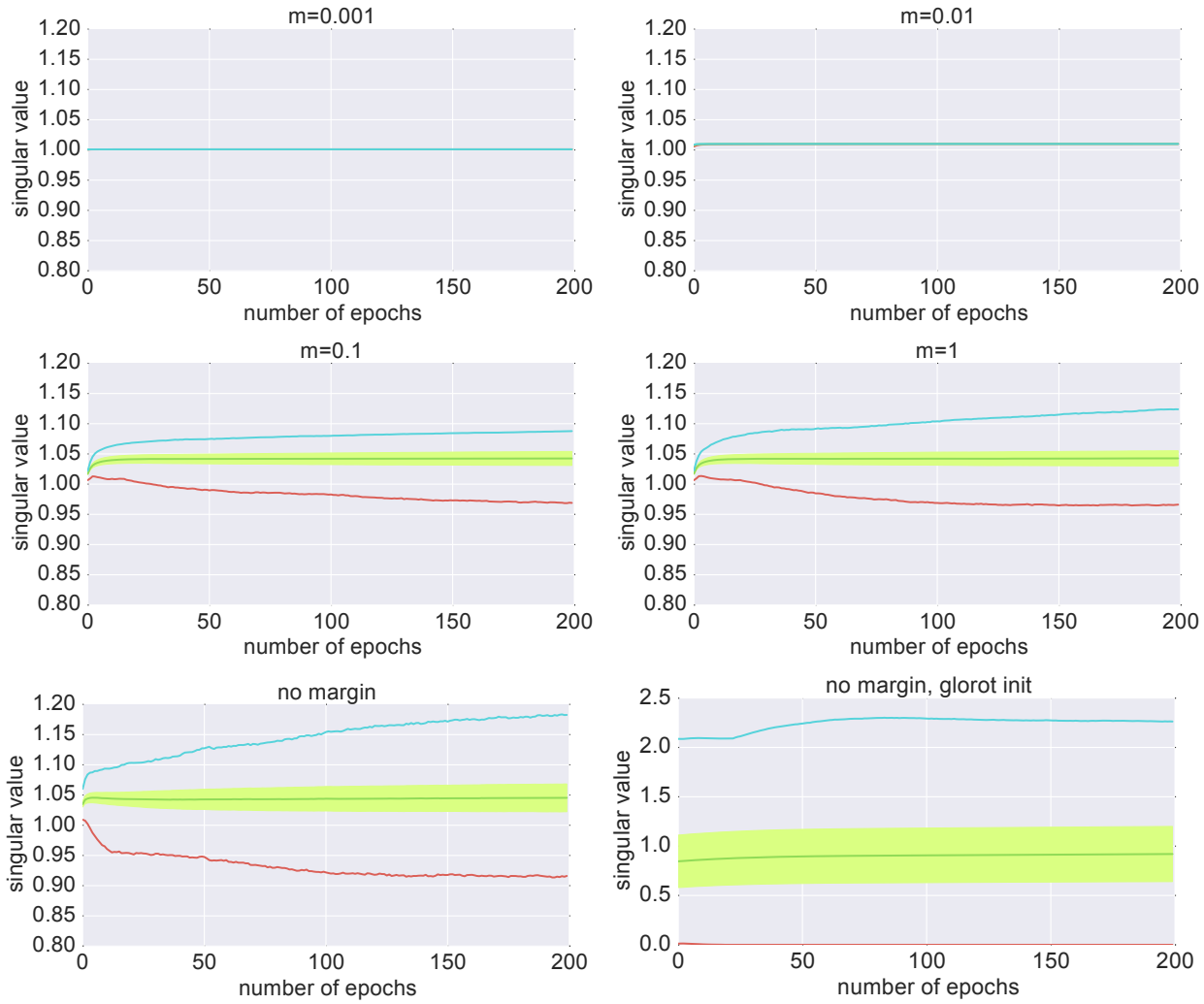


Figure 5.8 Singular value evolution on the permuted sequential MNIST task for factorized RNNs with different spectral margin sizes (m). The singular value distributions are summarized with the mean (green line, center) and standard deviation (green shading about mean), minimum (red, bottom) and maximum (blue, top) values. All models are initialized with orthogonal hidden to hidden transition matrices except for the model that yielded the plot on the bottom right, where Glorot normal initialization is used.

onality helps early training by stabilizing gradient flow across many time steps, orthogonality constraints may need to be loosened on some tasks so as not to over-constrain the model's representational ability.

Curiously, larger margins and even models without sigmoidal constraints on the spectrum (no margin) performed well as long as they were initialized to be orthogonal, suggesting that evolution away from orthogonality is not a serious problem on MNIST. It is not surprising that orthogonality is useful for the MNIST tasks since they depend on long distance signal

propagation with a single output at the end of the input sequence. On the other hand, character prediction with PTB produces an output at every time step. Constraining deviation from orthogonality proved detrimental for short sentences and beneficial when long sentences were included. Furthermore, Glorot normal initialization did not perform worse than orthogonal initialization for PTB. Since an output is generated for every character in a sentence, short distance signal propagation is possible. Thus it is possible that the RNN is first learning very local dependencies between neighbouring characters and that given enough context, constraining deviation from orthogonality can help force the network to learn longer distance dependencies.

5.2.4.1.3 Spectral and gradient evolution It is interesting to note that even long sequence lengths (T=1000) in the copy task can be solved efficiently with rather large margins on the spectrum. In Figure 5.9 we look at the gradient propagation of the loss from the last time step in the network with respect to the hidden activations. We can see that for a purely orthogonal parameterization of the transition matrix (when the margin is zero), the gradient norm is preserved across time steps, as expected. We further observe that with increasing margin size, the number of update steps over which this norm preservation survives decreases, though surprisingly not as quickly as expected.

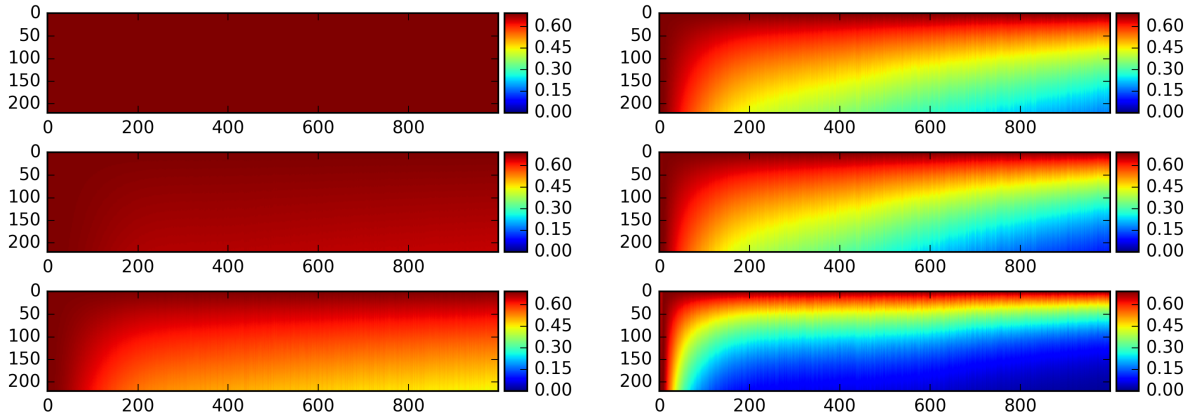


Figure 5.9 The norm of the gradient of the loss from the last time step with respect to the hidden units at a given time step for a length 220 RNN over 1000 update iterations for different margins. Iterations are along the abscissa and time steps are denoted along the ordinate. The first column margins are: 0, 0.001, 0.01. The second column margins are: 0.1, 1, no margin. Gradient norms are normalized across the time dimension.

Although the deviation of singular values from one should be slowed by the sigmoidal parameterizations, even parameterizations without a sigmoid (no margin) can be effectively trained for all but the longest sequence lengths. This suggests that the spectrum is not deviating far

from orthogonality and that inputs to the hidden to hidden transitions are mostly not aligned along the dimensions of greatest expansion or contraction. We evaluated the spread of the spectrum in all of our experiments and found that indeed, singular values tend to stay well within their prescribed bounds and only reach the margin when using a very large learning rate that does not permit convergence. Furthermore, when transition matrices are initialized as orthogonal, singular values remain near one throughout training even without a sigmoidal margin for tasks that require long term memory (copy, adding, sequential MNIST). On the other hand, singular value distributions tend to drift away from one for PTB character prediction which may help explain why enforcing an orthogonality constraint can be helpful for this task, when modeling long sequences. Interestingly, singular values spread out less for longer sequence lengths (nevertheless, the T=10000 copy task could not be solved with no sigmoid on the spectrum).

5.2.4.2 Exploring soft orthogonality constraints

We visualize the spread of singular values for different model parameterizations on the permuted sequential MNIST task in Figure 5.8. Curiously, we find that the distribution of singular values tends to shift upward to a mean of approximately 1.05 on both the ordered and permuted sequential MNIST tasks. We note that in those experiments, a tanh transition nonlinearity was used which is contractive in both the forward signal pass and the gradient backward pass. An upward shift in the distribution of singular values of the transition matrix would help compensate for that contraction. Indeed, [29] describe this as a possibly good regime for learning in deep neural networks. That the model appears to evolve toward this regime suggests that deviating from it may incur a cost. This is interesting because the cost function cannot take into account numerical issues such as vanishing or exploding gradients (or forward signals); we do not know what could make this deviation costly.

Unlike orthgonally initialized models, the RNN on the bottom right of Figure 5.8 with Glorot normal initialized transition matrices begins and ends with a wide singular spectrum. While there is no clear positive shift in the distribution of singular values, the mean value appears to very gradually increase for both the ordered and permuted sequential MNIST tasks. If the model is to be expected to positively shift singular values to compensate for the contractivity of the tanh nonlinearity, it is not doing so well for the Glorot-initialized case; however, this may be due to the inefficiency of training as a result of vanishing gradients, given that initialization.

That the transition matrix may be compensating for the contraction of the tanh is supported by further experiments: applying a 1.05 pre-activation gain appears to allow a model with

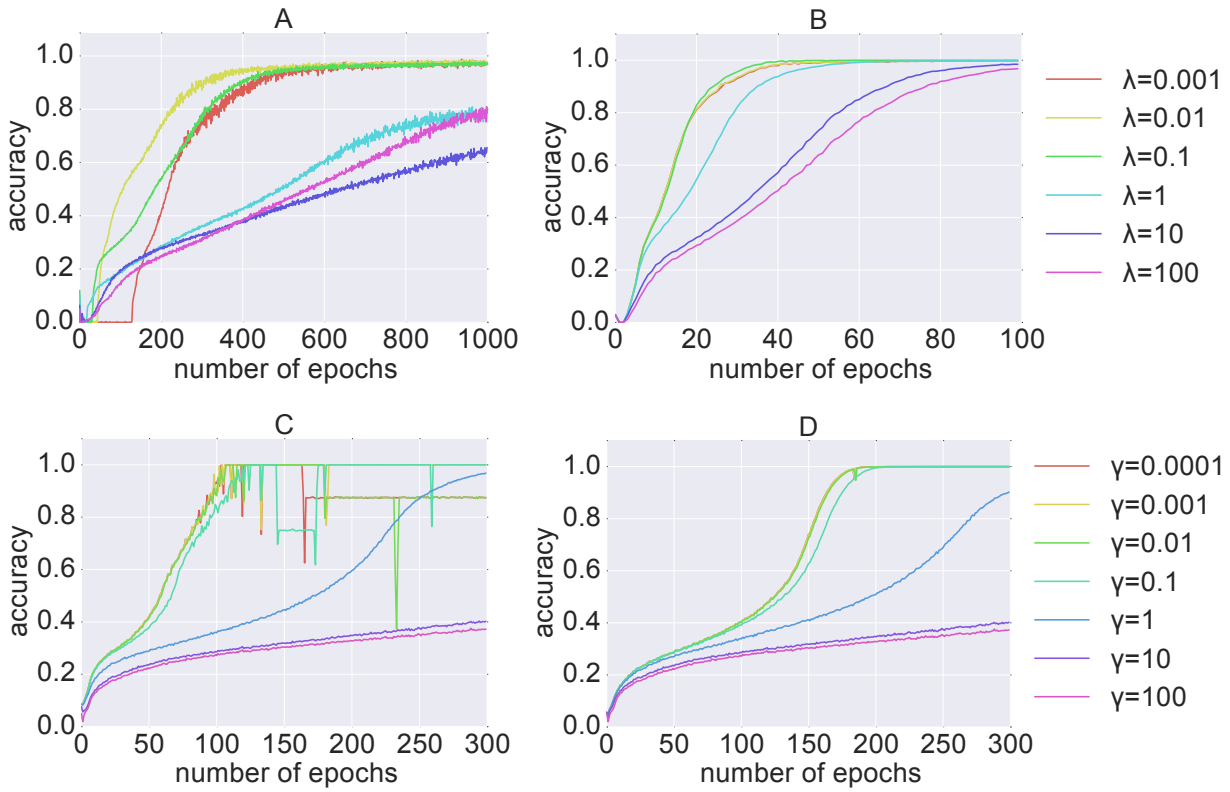


Figure 5.10 Accuracy curves on the copy task for different strengths of soft orthogonality constraints. All sequence lengths are $T = 200$, except in (B) which is run on $T = 500$. A soft orthogonality constraint is applied to the transition matrix $\mathbf{W}$ of a regular RNN in (A) and that of a factorized RNN in (B). A mean one Gaussian prior is applied to the singular values of a factorized RNN in (C) and (D); the spectrum in (D) has a sigmoidal parameterization with a large margin of 1. Loosening orthogonality speeds convergence.

a margin of 0 to nearly match the top performance reached on both of the MNIST tasks. Furthermore, when using the OPLU norm-preserving activation function [233], we found that orthogonally initialized models performed equally well with all margins, achieving over 90% accuracy on the permuted sequential MNIST task.

It is reasonable to assume that the sequential MNIST task benefits from a lack of a transition nonlinearity because it is a pure memory task. However, using no transition nonlinearity results in reduced accuracy. Furthermore, while a nonlinear transition may lose information, thus countering the task of memory preservation, it is critical on more advanced tasks like PTB character prediction. Indeed, tasks that require more than just memory preservation in the hidden state can benefit from forgetting. [235] demonstrated that adding a forgetting mechanism to RNNs constrained to have an orthogonal transition matrix improved performance on such tasks. Some forgetting could be achieved by allowing deviation from

orthogonality.

Having established that it may indeed be useful to step away from orthogonality, here we explore two forms of soft constraints (rather than hard bounds as above) on hidden to hidden transition matrix orthogonality. The first is a simple penalty that directly encourages a transition matrix $\mathbf{W}$ to be orthogonal, of the form $\lambda||\mathbf{W}^T\mathbf{W} - \mathbf{I}||_2^2$. This is similar to the orthogonality penalty introduced by [222]. In subfigures (A) and (B) of Figure 5.10, we explore the effect of weakening this form of regularization. We trained both a regular non-factorized RNN on the $T = 200$ copy task (A) and a factorized RNN with orthogonal bases on the $T = 500$ copy task (B). For the regular RNN, we had to reduce the learning rate to 10^{-5}. Here again we see that weakening the strength of the orthogonality-encouraging penalty can increase convergence speed.

The second approach we explore replaces the sigmoidal margin parameterization with a mean one Gaussian prior on the singular values. In subfigures (C) and (D) of Figure 5.10, we visualize the accuracy on the length 200 copy task, using geoSGD (learning rate 10^{-6}) to keep $\mathbf{U}$ and $\mathbf{V}$ orthogonal and different strengths γ of a Gaussian prior with mean one on the singular values s_i: $\gamma \sum_i ||s_i - 1||^2$. We trained these experiments with regular SGD on the spectrum and other non-orthogonal parameter matrices, using a 10^{-5} learning rate. We see that strong priors lead to slow convergence. Loosening the strength of the prior makes the optimization more efficient. Furthermore, we compare a direct parameterization of the spectrum (no sigmoid) in (C) with a sigmoidal parameterization, using a large margin of 1 in (D). Without the sigmoidal parameterization, optimization quickly becomes unstable; on the other hand, the optimization also becomes unstable if the prior is removed completely in the sigmoidal formulation (margin 1). These results further motivate the idea that parameterizations that deviate from orthogonality may perform better than purely orthogonal ones, as long as they are sufficiently constrained to avoid instability during training.

5.2.5 Conclusions

We have explored a number of methods for controlling the expansivity of gradients during backpropagation based learning in RNNs through manipulating orthogonality constraints and regularization on weight matrices. Our experiments indicate that while orthogonal initialization may be beneficial, maintaining hard constraints on orthogonality can be detrimental. Indeed, moving away from hard constraints on matrix orthogonality can help improve optimization convergence rate and model performance. However, we also observe with synthetic tasks that relaxing regularization which encourages the spectral norms of weight matrices to be close to one too much, or allowing bounds on the spectral norms of weight matrices to be

too wide, can reverse these gains and may lead to unstable optimization.

5.2.6 Acknowledgments

We thank the Natural Sciences and Engineeering Research Council (NSERC) of Canada and Samsung for supporting this research.

CHAPTER 6 TOWARDS SEMI-SUPERVISED SEGMENTATION VIA IMAGE-TO-IMAGE TRANSLATION

6.1 Preamble

This work is currently in the arXiv [236]. For this work, I proposed the main ideas, wrote the code, and did the experiments and the writing.

6.2 Introduction

Semantic object segmentation from natural images is known to perform well with deep neural networks but these require a large quantity of pixel-level annotations. Obtaining a sufficient quantity of annotations is difficult and sometimes impractical; on the other hand, unlabeled or weakly categorized data is easier to obtain. We propose a semi-supervised segmentation model that can use this weakly characterized data.

Many works have explored the use of generative adversarial networks (GANs) to improve semantic segmentation of medical images. However, while these methods make better use of the training data by either improving the training objective [155–163] or performing data augmentation within the training set [149, 150, 237], they do not augment the training set to better cover the variations in the data population. On the other hand, some works have explored unsupervised anomaly localization using autoencoding [121] or GANs [122, 123] to learn a generative model of healthy cases. Another GAN based approach is to train an error model that could be used for updates on unlabeled data [119]. These approaches are approximate and do not make full use of available weak labels (healthy and sick domain labels). Making better use of available data, some recent approaches relied on image-to-image translation between sick and healthy cases [124, 125] but these were unsupervised and either approximate or not validated against baselines or on multiple tasks.

We focus on the common scenario in medical imaging, where a large number of images lack segmentation labels but are known to be either healthy or sick cases. This knowledge can be considered as weak proxy labels that identify whether there is something to be segmented in an image. For example, when segmenting cancerous lesions, images marked ‘healthy’ do not contain cancer while images marked ‘sick’ do. We argue that the objective of translating from sick to healthy images is a good unsupervised surrogate for segmentation. Consequently, we develop a semi-supervised segmentation method with image-to-image translation, trained on unpaired images from sick and healthy domains.

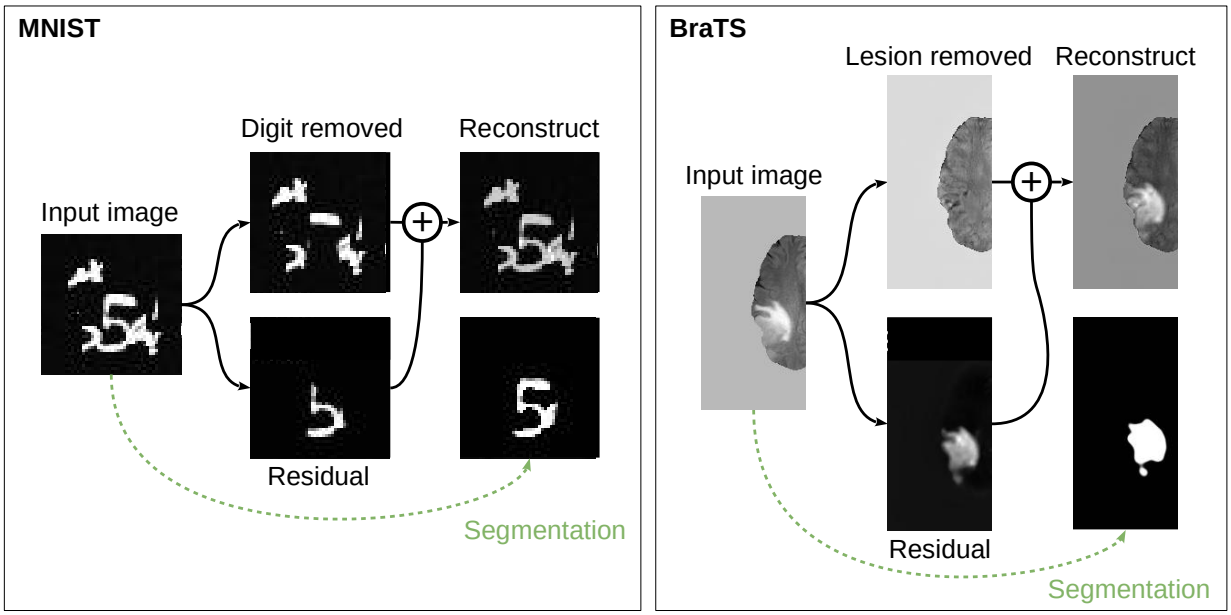


Figure 6.1 *Left*: Images presenting digits transformed to images with only the background clutter, a residual image that isolates the digit, and a segmentation of the digit. *Right*: Images presenting cancer lesions in the brain are transformed to healthy images, a residual image that isolates the lesion, and a segmentation of the lesion.

Considering the sick domain as a superset of the variations in the healthy domain, we encode images into two latent codes: variations that are *common* to both and variations that are *unique* to the sick domain. This allows us to split decoding into two parts: (1) a 'healthy' image decoder that interprets the *common* latent code and (2) a residual decoder that additionally considers the *unique* code in order to compute a residual change to the 'healthy' output image, making it 'sick'.

Because the output of the residual decoder is highly correlated with the segmentation output we can re-use the decoder for segmentation. In doing so, we maximize the proportion of model parameters that receive updates even when there are no pixel-level annotations available to guide image segmentation during training. Examples of these mappings, including both decoders and the segmentation output, are shown in Figure 6.1. Furthermore, whereas image-to-image translation models do not use long skip connections from the encoder to the decoder, we propose a long skip connection variant in our method. Long skip connections are common with supervised encoder-decoder models [90], where they help preserve spatial detail in the decoder even when the encoding is very deep. Overall, we summarize our contributions as follows:

- We propose a semi-supervised segmentation method leveraging image-to-image trans-

lation.

- We propose the use of (new) long skip connections for image-to-image translation, from encoder to decoder.
- We propose a dual-function decoder (translation, segmentation), thus maximizing the number of parameters updated in the absence of pixel-level annotations.
- We validate our method on challenging synthetic data and real brain tumor MR images, significantly improving over well-tuned baselines.

6.3 Related works

Image-to-image translation. Image to image translation was most prominently done with the CycleGAN [126] which does bidirectional translation between two domains. UNIT [151] proposed a similar approach but with a common latent space, shared by both domains, from which latent codes could be sampled. Augmented CycleGAN [152] and Multimodal UNIT [153] respectively extended both methods from one-to-one mappings to many-to-many.

Disentangling domain-specific variations. Both [153] and [238] present methods that learn shared and domain-specific latent codes. These differ from the proposed method in that they do not segment and do not assume (and benefit from) an *absence* domain as a subset of a *presence* domain. In addition, the domain-specific "style" codes are encoded with a shallow network which may bias the model to indeed learn domain-specific styles; whereas, the proposed method uses deep encodings for all codes. Explicit disentangling of variations between these codes has recently been proposed in [239] by way of a gradient reversal layer [240].

Data Augmentation. GANs are used to augment liver lesion examples for classification in [147]. [148] synthesize data to cover uncommon cases such as peripheral nodules touching the lung boundary. [149] and [150] introduce a segmentation mask generator to augment small training datasets.

Anomaly localization. Generative models have been used to fit the distribution of healthy images in order to find anomalies. To localize lesions in brain MR images that are known to be either healthy or sick, [121] fit the healthy data distribution with an autoencoder. Given an image presenting a lesion, the lesion is localized via the residual of its reconstructed image which is likely to appear healthy. Similarly, [122] and [123] employ GAN to locate anomalies in retinal images and brain MR images, respectively. While these models require that weak

'sick' or 'healthy' labels are known, they are trained only on the latter. Furthermore, they allow only rough unsupervised localization.

Image-to-image translation for segmentation. By translating from sick to healthy images, [124] trains a network to localize Alzheimer's derived brain morphological changes using the output residual. [125] further proposes a multi-modal variant of CycleGAN [126] to translate in both directions, applied to brain MR images with cancer. Sick images that are translated to healthy images are translated back to the original sick image via a residual inpainting of the lesions. Lesions are localized and segmented by predicting a minimal region to which to apply inpainting. Segmentation is unsupervised, with a prior that minimizes the inpainting region. This method has not been compared to other unsupervised methods, has been tested on a single dataset, and has not been extended to a weakly- or semi-supervised setting. Our work differs in that we develop a semi-supervised architecture that uses fewer parameters by reusing mappings, we skip information from the encoder to decoders, we propose a decoder that is trained with both translation and segmentation objectives, and we validate the method on multiple tasks.

Adversarial semi-supervised segmentation. A semi-supervised segmentation method for medical images was proposed by [119], where a discriminator learns a segmentation error signal on the annotated dataset which can be applied on unannotated data. This method may be limited in how well it could scale with the proportion of unannotated data since the discriminator's behaviour may not generalize well beyond the annotated dataset on which it is trained. Because this method can be applied to the output of any segmentation model, we consider it complementary to our proposed method.

6.4 Methods

Segmentation labels are typically available for an insufficiently representative sample of data. We propose a semi-supervised method that extends supervised segmentation to weakly labeled data using a domain translation objective. In addition to a segmentation objective, the method attempts to translate between the distribution of images presenting the segmentation target (P) and the distribution of images where this target is absent (A).

6.4.1 Translation, segmentation, and autoencoding

Translating between images where the segmentation target object is present or absent requires a model to localize the target. It follows then that in order to add, remove, or modify the target in an image, the variations caused by it should be disentangled from everything else

(Figure 6.2, left). We conjecture that segmentation relies on the same disentangling and that this is the most difficult part of both objectives. Thus, we identify domain translation as an unsupervised surrogate loss for segmentation. We propose an encoder-decoder model that extends segmentation with image-to-image translation. In addition, we leverage the similarity between these two objectives to employ a decoder that is shared by both.

Although domain translation aligns well with segmentation, the canonical objective for unsupervised feature learning is autoencoding of the model input. A deep autoencoder may disentangle causal features of an image; that is, encoding the image may yield information about the features that produce it (Figure 6.2, right). When labels could be considered to cause the image, one would expect autoencoding to learn features that are useful for classification or segmentation [241]. Indeed, [21] recently won the Brain Tumor Segmentation challenge (2018) by augmenting a fully convolutional segmentation network with an autoencoding objective. This objective is easy to set up and train for. Unlike with domain translation, no knowledge about the images' domain is required. On the other hand, information about presence (P) or absence (A) of the segmentation target in the image may guide a domain translation objective to more specifically isolate the variations that are important for segmentation [239].

6.4.2 Our method

The proposed model builds on an encoder-decoder fully convolutional network (FCN) segmentation setup by introducing translation between a domain of images presenting the segmentation target (P) and a domain where it is absent (A), as in Figure 6.3. The encoder separates variations into those that are *common* to both A and P and those that are *unique* to P; essentially, P is a superset of the variations in A. For example, in the case of medical images of cancer, both A and P contain the same organs but P additionally contains cancerous lesions.

Latent code decomposition. Starting with images $\mathbf{x}$ in domains A or P, the encoder (f) yields *common* ($\mathbf{c}$) and *unique* ($\mathbf{u}$) codes:

$$\begin{aligned} [\mathbf{c_A}, \mathbf{u_A}] &= f(\mathbf{x_A}), \\ [\mathbf{c_P}, \mathbf{u_P}] &= f(\mathbf{x_P}). \end{aligned} \tag{6.1}$$

This decomposition of the latent codes is reminiscent of the *style* and *content* decomposition in [153] or the domain-specific codes in [238].

Presence to absence translation. Translation is achieved by selectively decoding from

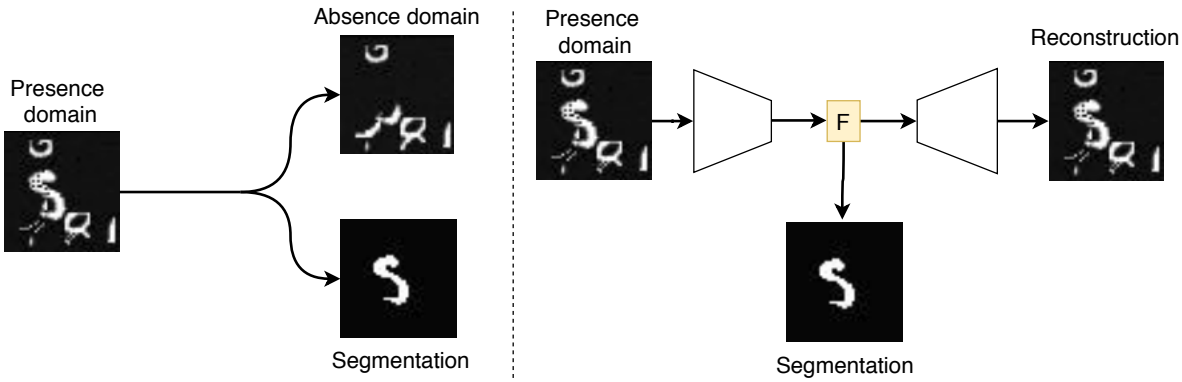


Figure 6.2 *Left:* Translating images from a domain presenting the segmentation target object (Presence) to one in which it is absent (Absent) involves disentangling the object's variations from the rest. The former is useful for segmentation, the latter for producing an image without the object. *Right:* Autoencoding may produce disentangled features (F) that are useful but not optimal for segmentation.

the latent codes $\mathbf{c}$ and $\mathbf{u}$. A *common* decoder (g_{com}) uses only common variations, $\mathbf{c}$, to generate images in A:

$$\begin{aligned} \mathbf{x_{AA}} &= g_{com}(\mathbf{c_A}), \\ \mathbf{x_{PA}} &= g_{com}(\mathbf{c_P}), \end{aligned} \tag{6.2}$$

where $\mathbf{x_{AA}}$ is essentially an autoencoding of $\mathbf{x_A}$, whereas $\mathbf{x_{PA}}$ is a translation of $\mathbf{x_P}$ to the A domain where the segmentation target is removed. With this translation, the target variations can be recovered separately, by computing a residual change $\mathbf{\Delta_{PA}}$ to $\mathbf{x_{PA}}$ that reconstructs $\mathbf{x_P}$ as $\mathbf{x_{PP}}$. This is done with a second *residual* decoder (g_{res}) which uses both common variations and those unique to P (see Figure 6.3):

$$\begin{aligned} \mathbf{x_{PP}} &= \mathbf{x_{PA}} + \mathbf{\Delta_{PA}}, \\ \text{where } \mathbf{\Delta_{PA}} &= g_{res}(\mathbf{c_P}, \mathbf{u_P}). \end{aligned} \tag{6.3}$$

The residual decoder requires all latent codes, $\{\mathbf{c_P}, \mathbf{u_P}\}$, as its input because the manifestation of unique variations in the image space is dependent on the common variations. For example, the way cancer manifests in a brain scan depends on the location and structure of the brain in the scan. Note also that because the common decoder only uses the common latent code, the encoder must learn to disentangle common and unique variations.

Segmentation. The $\mathbf{c_P}$ and $\mathbf{u_P}$ codes or the $\mathbf{\Delta_{PA}}$ residual contain sufficient information for segmentation. Indeed we reuse the residual decoder, used with $\mathbf{x_P}$, for segmentation. We parameterize a segmentation decoder g_{seg} in terms of the residual decoder g_{res}, with

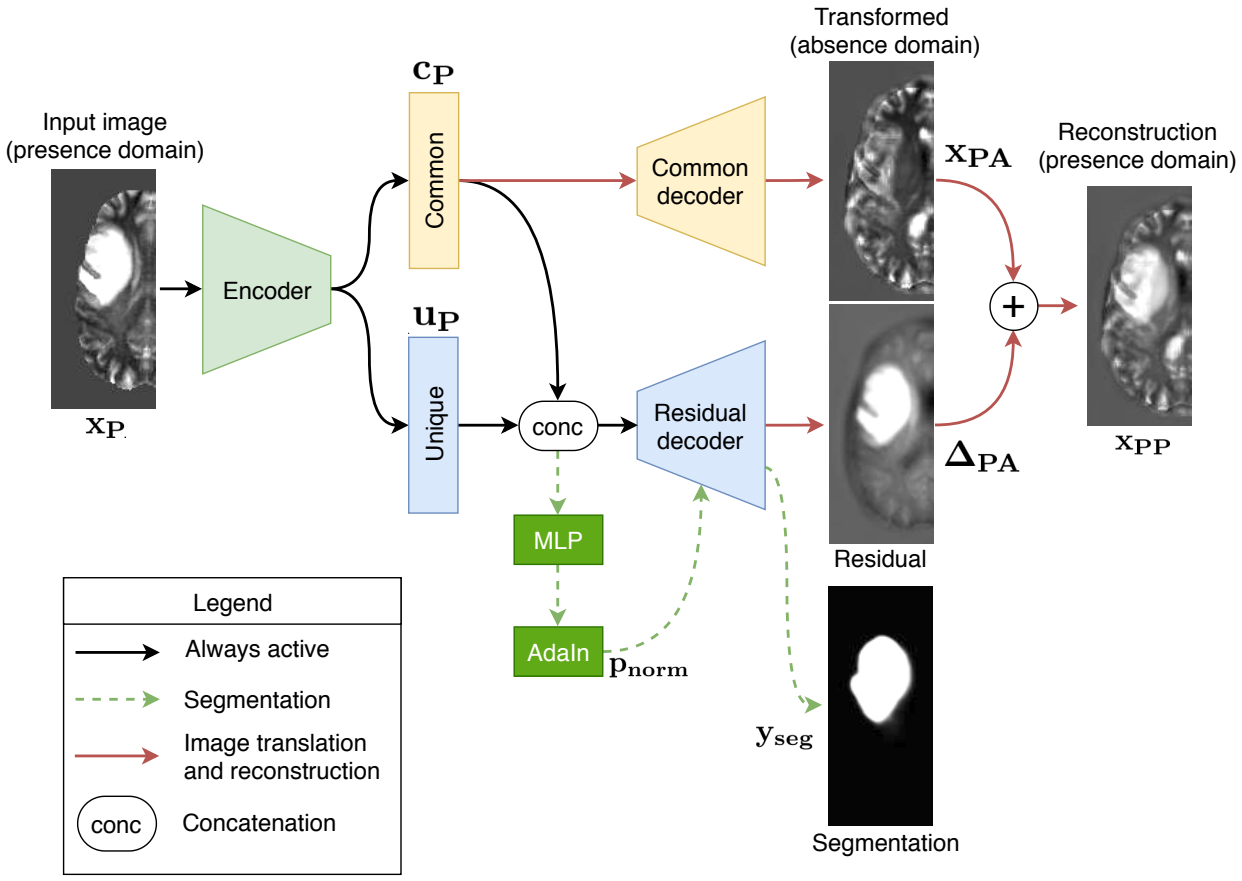


Figure 6.3 Framework overview of simultaneous segmentation, image translation and reconstruction. Images are transformed from the *presence* domain into the *absence* domain. Transformations are evaluated by a discriminator (not shown). The encoder and each decoder share skip connections for higher quality image generation.

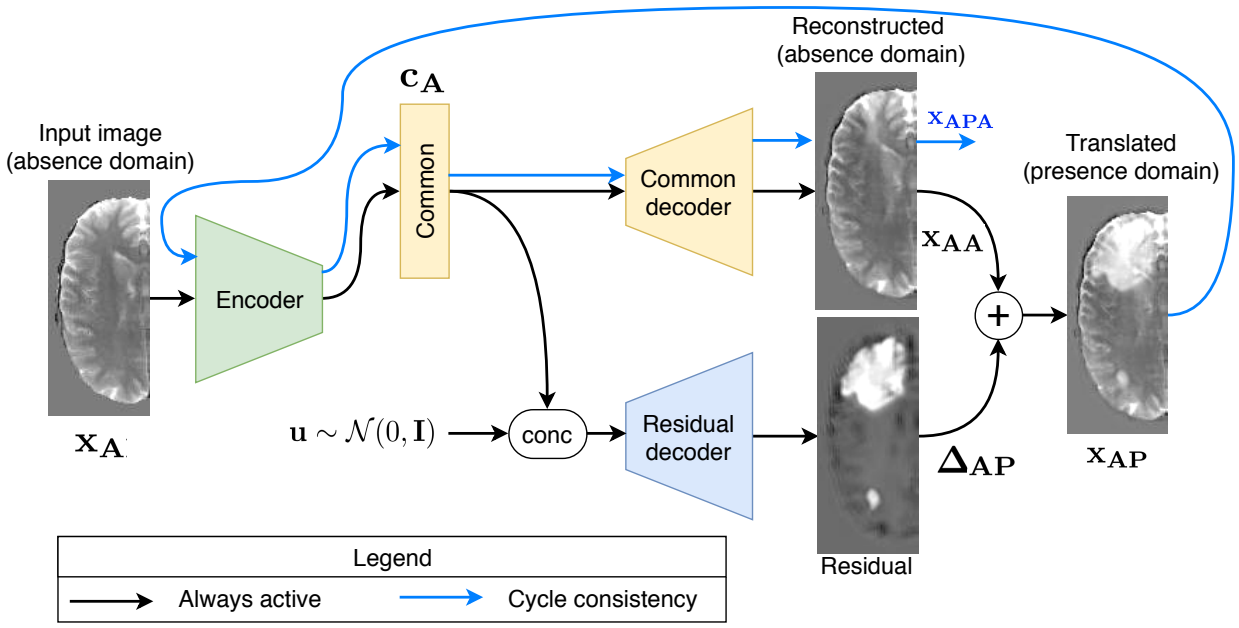


Figure 6.4 Image-to-image translation from the *absence* domain to the *presence* domain. The common code extracted by the encoder is used to reconstruct the input image. The unique code is sampled from a Normal distribution and concatenated to the common code to produce a residual image which, when added to the reconstructed image, yields a new image in the *presence* domain. We cycle the image back through the encoder and the common decoder to ensure that the reconstructed image remains unchanged.

segmentation specific per-layer instance normalization [242] parameters $\mathbf{p_{norm}}$:

$$\begin{aligned} \mathbf{y_{seg}} &= g_{seg}(\mathbf{c_P}, \mathbf{u_P}), \\ &= (\hat{g}_{res} \circ s)(\mathbf{c_P}, \mathbf{u_P}), \end{aligned} \tag{6.4}$$

where s is a pixelwise classification layer and $\hat{g}_{res}$ is a subset of the g_{res} network that contains all but the last layer, using normalization parameters produced from the latent code by a multi-layer perceptron (MLP):

$$\mathbf{p_{norm}} = MLP(\mathbf{c_P}, \mathbf{u_p}). \tag{6.5}$$

Absence to presence translation. Finally, we conclude the set of autoencoding and translation equations with $\mathbf{x_{AP}}$ and $\mathbf{x_{APA}}$, where images in A are translated to images in P (Figure 6.4). We note that although these translations are not useful for segmentation, they are useful during training since they effectively augment the training updates that our encoders and decoders can receive. Since P contains additional variations to those found in A, we must either add these variations from an image in A or sample them from a prior distribution:

$$\begin{aligned} \mathbf{x_{AP}} &= g_{com}(\mathbf{c_A}) + g_{res}(\mathbf{c_A}, \mathbf{u} \sim \mathcal{N}(0, \mathbf{I})), \\ \mathbf{x_{APA}} &= g_{com}(\mathbf{c_{AP}}), \\ [\mathbf{c_{AP}}, \mathbf{u_{AP}}] &= f(\mathbf{x_{AP}}). \end{aligned} \tag{6.6}$$

Here, $\mathbf{x_{AP}}$ requires a sample $\mathbf{u}$ from a zero-mean, unit variance prior over the unique variations, $\mathcal{N}(0, \mathbf{I})$. Note that unlike in a variational autoencoder, the encoder f does not parameterize a conditional distribution over the unique variations but rather encodes a sample directly. We ensure that the distribution of encoded samples matches the prior by making $\mathbf{u_{AP}}$ match $\mathbf{u}$, as detailed further in the description of our training objective. The translation of $\mathbf{x_{AP}}$ to $\mathbf{x_{APA}}$ completes a cycle as in [126]. When $\mathbf{x_{APA}}$ must match $\mathbf{x_A}$, this ensures that the translations retain information about their source images, ensuring that the encoder and decoders do not learn trivial functions. As shall be seen below, this is already achieved by other objectives, making the cycle optional.

Total loss. The training objective consists of a segmentation loss L_{seg} combined with four translation losses, each weighted by some scalar λ:

$$\begin{aligned} L_{total} = L_{seg} &+ \lambda_{rec} L_{rec} + \lambda_{lat} L_{lat} \\ &+ \lambda_{cyc} L_{cyc} + \lambda_{adv} L_{adv}. \end{aligned} \tag{6.7}$$

Segmentation loss. We use Dice loss for segmentation, as in [90, 91], which measures the overlap between the predicted segmentation $\mathbf{y}$ and reference segmentation $\hat{\mathbf{y}}$:

$$L_{seg} = \text{Dice}(\mathbf{y}, \hat{\mathbf{y}}). \tag{6.8}$$

Reconstruction losses. To ensure that the encoder and decoders can cover the distribution of images, we reconstruct input images:

$$L_{rec} = L_{rec}(\mathbf{x_P}, \mathbf{x_{PP}}) + L_{rec}(\mathbf{x_A}, \mathbf{x_{AA}}). \tag{6.9}$$

Similarly, we reconstruct the latent codes so as to ensure that their distributions match across domains A and P, or in the case of unique codes, match the prior:

$$\begin{aligned}
[\mathbf{c_{PA}}, \mathbf{u_{PA}}] &= f(\mathbf{x_{PA}}) \\
[\mathbf{c_{PP}}, \mathbf{u_{PP}}] &= f(\mathbf{x_{PP}}) \\
L_{lat} &= L_{lat}(\mathbf{c_P}, \mathbf{c_{PA}}) + L_{lat}(\mathbf{c_A}, \mathbf{c_{AP}}) \\
&\quad + L_{lat}(\mathbf{c_A}, \mathbf{c_{AA}}) + L_{lat}(\mathbf{c_P}, \mathbf{c_{PP}}) \\
&\quad + L_{lat}(\mathbf{u_P}, \mathbf{u_{PP}}) + L_{lat}(\mathbf{u}, \mathbf{u_{AP}}).
\end{aligned} \tag{6.10}$$

We define a cycle consistency loss for the APA cycle:

$$L_{cyc} = L_{rec}(\mathbf{x_A}, \mathbf{x_{APA}}). \tag{6.11}$$

Note that there is no PAP cycle since in the proposed method this is equivalent to PP reconstruction, as can be seen in Figure 6.3. Because both images and their latent codes are reconstructed, the cycle consistency loss is optional.

We use the L_1 distance for all reconstruction losses.

Adversarial loss. Finally, we use the hinge loss for the adversarial objective, together with spectral norm on the encoder and decoders as in [106]:

$$\begin{aligned}
L_{adv} = \sum_{d \in \{A,P\}} \min_G \max_D \Big[& \\
& - \mathbb{E}_{\mathbf{x_d} \sim p_d} \left[\min(0, D_d(\mathbf{x_d}) - 1) \right] \\
& - \mathbb{E}_{\hat{\mathbf{x}}_\mathbf{d} \sim \hat{p}_d} \left[\min(0, -D_d(G_d(\hat{\mathbf{x}}_\mathbf{d})) - 1) \right] \\
& - \mathbb{E}_{\hat{\mathbf{x}}_\mathbf{d} \sim \hat{p}_d} D_d(G_d(\hat{\mathbf{x}}_\mathbf{d})) \Big],
\end{aligned} \tag{6.12}$$

where, for each domain $d \in \{A, P\}$, G_d is the generator network for some generated image $\hat{\mathbf{x}}_\mathbf{d} \sim \hat{p}_d$ and D_d is a discriminator network which discriminates between real data $\mathbf{x_d} \sim p_d$ and generated data $\hat{\mathbf{x}}_\mathbf{d}$.

6.4.3 Baseline methods

The proposed method is compared against two baseline approaches: a fully supervised, fully convolutional network (FCN) and another one augmented with a reconstruction objective for semi-supervised training. To ease comparison, all models (baselines and proposed) share the same encoder and decoder architectures. The fully supervised method uses an encoder with a single decoder ("Only segmetation" in Table 6.1). This is equivalent to the proposed method with only the segmentation loss, using only the residual decoder. The semi-supervised method ("AE baseline" in Table 6.1), adds an additional decoder that reconstructs the input.

6.4.4 Compressed long skip connections

In all models, including baselines, every decoder accepts long skip connections from the encoder, as in [90]. These connections skip features from each layer in the encoder to the corresponding layer in the decoder, except for the first and the last layers. Because long skip connections make autoencoding trivial, they are not used with the reconstruction decoder in the semi-supervised baseline method, however skip connections are used between the encoder and segmentation decoder.

Typically, feature maps from the encoder are either directly summed with [90] or concatenated to [19] those in the decoder. We proposed a modified variant of long skip connections where any stack of feature maps is first compressed (via 1×1 convolution) to a single map before concatenation (see Figure 6.5). We note that concatenating all feature maps is costly computationally and appears to increase training time for image translation whereas summing feature maps makes the image translation task very difficult to learn. To further stabilize training, all features skipped from the encoder are normalized with instance normalization. We find that these long skip connections help train the model faster and help produce higher quality image outputs even with a deep encoder.

6.4.5 Model implementation and training details

Details on the model architectures, parameter initializations, and optimization hyperparameters are provided in Appendix B.1.

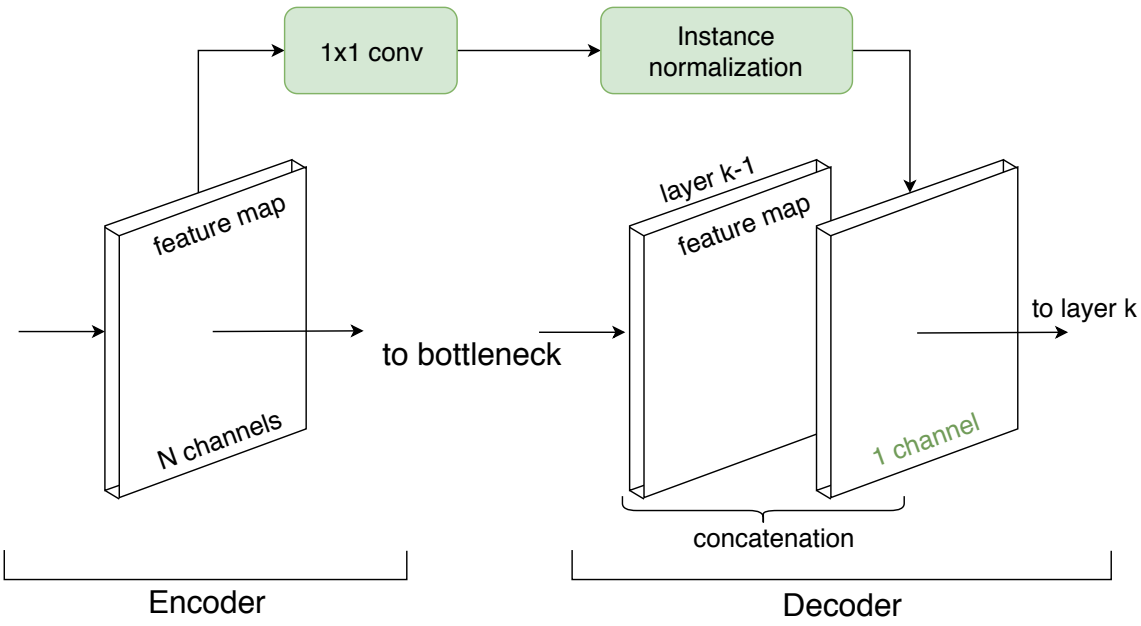


Figure 6.5 Compressed skip connection as a way to limit information bypass while preserving spatial detail.

6.5 Experiments

In this section, we evaluate our proposed semi-supervised segmentation method on both synthetic and real data. We confirm that this method outperforms the fully supervised and semi-supervised segmentation baselines. We present examples of image translation results to illustrate the correlation between segmentation and image translation outputs. We begin with a synthetic MNIST-based task that simulates the common situation, where some images are known to present the segmentation target (P), while in others it is known to be absent (A). With this data, we experiment with increasing the difficulty of the segmentation task. Then, we proceed to evaluate the proposed method on brain tumour segmentation on real MRI data (BraTS).

6.5.1 Cluttered MNIST

We construct a synthetic task for digit segmentation using MNIST digits, similar to the cluttered MNIST dataset in [243]. Each image in P contains a complete randomly positioned digit placed on a background of clutter. The clutter is produced from randomly cropped digits within the same data fold (training, validation, or test set). In all experiments, we used crops of 10x10 pixels. We dither regions where MNIST digits or clutter components

Table 6.1 Segmentation Dice scores of proposed method compared to baselines for synthetic MNIST and real BraTS segmentation tasks: mean (standard deviation).

	MNIST			BraTS
	48×48 simple	48×48 hard	128×128	240×120
Only segmentation	0.61 (0.01)	0.36 (0.01)	0.15 (0.01)	0.69 (0.04)
AE baseline	0.75 (0.01)	0.49 (0.02)	0.57 (0.02)	0.73 (0.02)
Proposed	**0.79 (0.01)**	**0.57 (0.00)**	0.65 (0.01)	**0.79 (0.02)**
Proposed (sep dec)	0.78 (0.02)	0.50 (0.01)	**0.70 (0.00)**	0.77 (0.03)

overlap, so as to prevent models from identifying these boundaries.

We tested the proposed model and the baseline methods on three variants of the cluttered MNIST task, at two resolutions: *48×48 simple*, with 8 pieces of clutter; *48×48 hard*, with 24 pieces of clutter; and *128×128*, with 80 pieces of clutter. Samples from these generated datasets are shown in Figure 6.6; all datasets were generated prior to training. In all experiments, we provided reference segmentations for 1% of the training examples. In addition, to mimic the issue of small training datasets where the training set fails to cover all modes of variation of the data population, we trained on reference segmentations only for the digit 9.

As shown in Table 6.1, the proposed model significantly outperforms both the semi-supervised and the fully supervised baselines. The improvement is greater for the harder variants of the task: *48×48 hard* and *128×128*. We suspect that the greater improvement on the *48×48 hard* task may be due to the greater difficulty in separating digits from clutter as compared to *48×48 simple*, in which case the translation objective that seeks to specifically disentangle digit variations from clutter variations should be particularly helpful.

Examples of image translation and segmentation are shown in Fig. 6.7. We first discuss

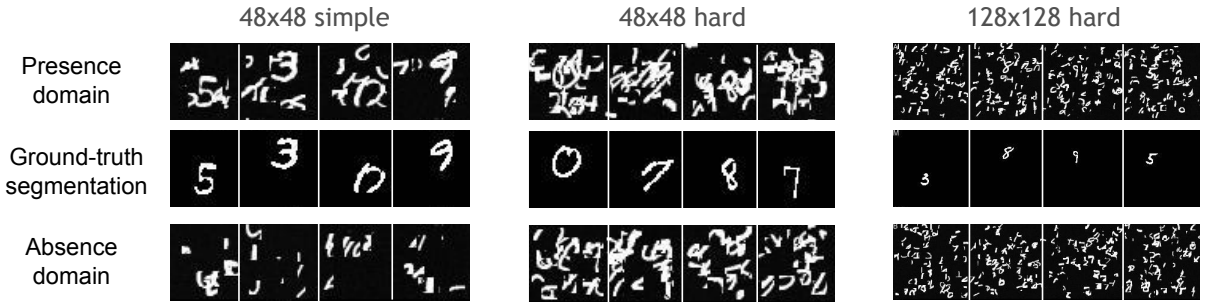


Figure 6.6 Examples of images from the synthetic MNIST datasets. Samples from the *presence* domain and corresponding ground truth segmentations are in the first and second rows; unrelated samples from the *absence* domain are in the third row.

presence to *absence* translation. For the MNIST *48×48 simple* case, almost the entire digit is removed from the image during translation and the residual is similar to the segmentation result which supports our conjecture that translation is a good surrogate task for the segmentation task. The MNIST *48×48 hard* dataset has very challenging images, in which localizing the true digit is very difficult even for people. The model learned to partially remove the digit in order to fulfill the GAN objective. Therefore, the residual does not contain the entire digit, however it attends to the correct location in the image which may guide segmentation. We note that a digit does not need to be completely removed in order for the image to appear to contain only clutter because any remaining digit parts could appear as clutter. During *absence* to *presence* translation, the model learns the distribution of correct digits and is able to insert them into the image, as shown for MNIST *48×48 simple*. With more clutter (MNIST *48×48 hard*) it becomes challenging; generated residuals have less variety and many look like variations of the digit 0.

These experiments demonstrate that semi-supervised segmentation benefits from image-to-image translation. We observed significant improvements over supervised and semi-supervised segmentation baselines.

6.5.2 BraTS

Moving beyond synthetic data, we evaluated the proposed method on brain tumour segmentation challenge 2017 data (BraTS). Because this dataset contains only magnetic resonance imaging (MRI) volumes presenting cancer, we artificially split the data along 2D axial slices into P and A domains as a proof of concept, following the setup in [244]. As in [244], we split the brains into hemispheres and select only those half-slices that contain at least 25% brain pixels, so as to better balance the slice distributions between the two domain. Else, slices from the center of the brain, containing more brain matter, would be over-represented in P as compared to A. In P, we also limit the minimal number of lesion pixels to 1% of brain pixels. We pre-process every volume by mean-centering the brain pixels (ignoring background) and dividing them by their standard deviation. Finally, we use half-slices extracted from the processed volumes as model inputs. Each input has four channels, corresponding to four registered MRI sequences: T1, T2, T1C, and FLAIR.

We trained the proposed model and baselines with reference labels available for 1% of the training data. As shown in Table 6.1, the proposed model achieves a 0.79 Dice score, significantly outperforming both the segmentation baseline, 0.69, and the semi-supervised autoencoding baseline, 0.73. Image translation and segmentation examples are shown in Figure 6.8. As evident in the figure, lesions were well removed by image-to-image translation. Unlike

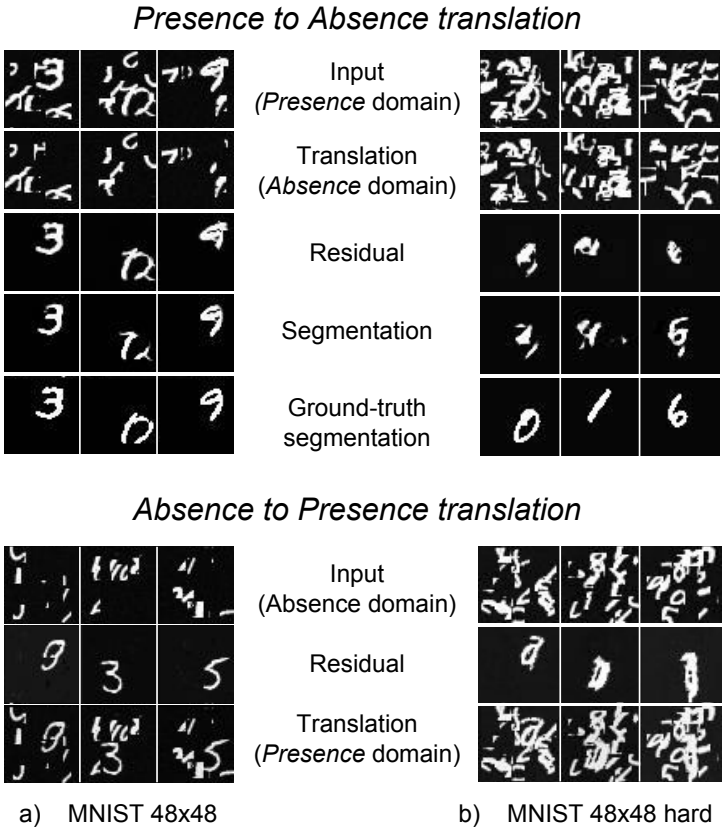
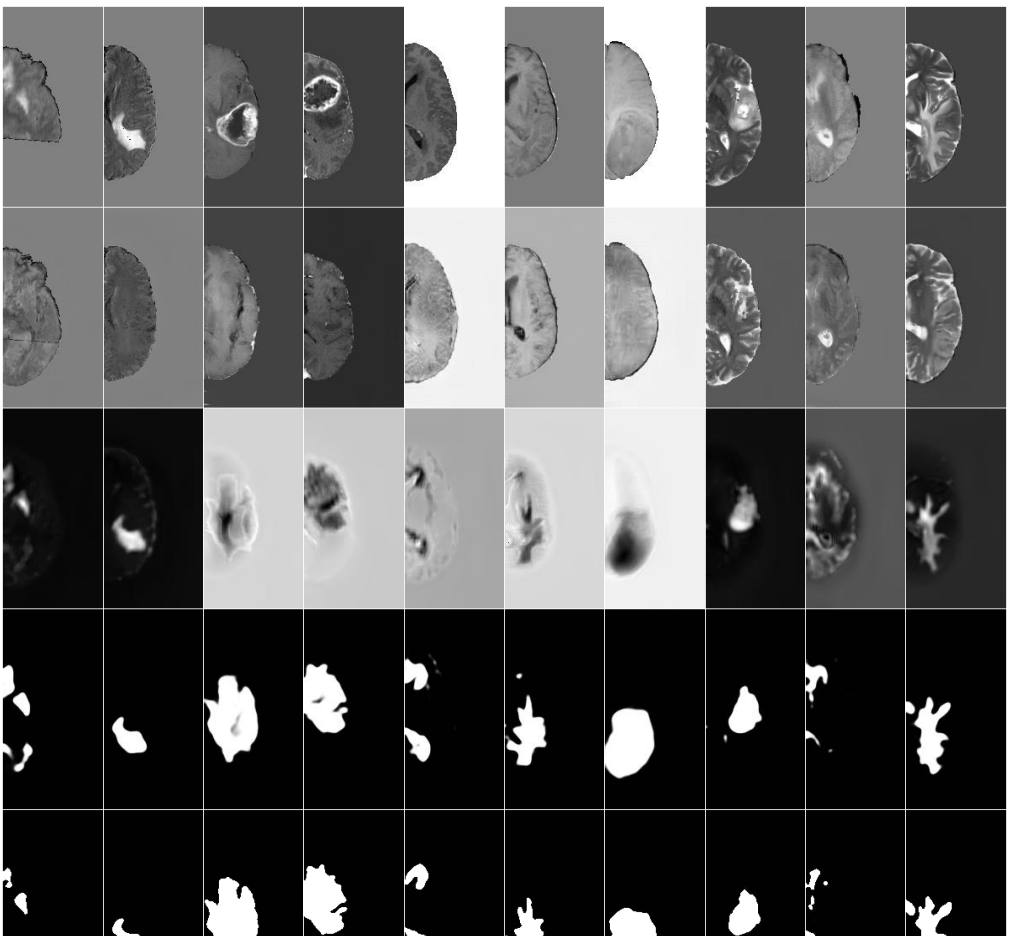


Figure 6.7 Example of image translation and segmentation for cluttered MNIST.


Input (*Presence*)
Translation (*Absence*)
Residual
Segmentation
Ground-truth segmentation

with the cluttered MNIST data, some of the sequences (T1, T1c) result in fairly complicated residuals that are nonetheless correctly reinterpreted as segmentations via the residual decoder.

The first column in Figure 6.8 reveals an artifact of distribution imbalance where a rare truncated input slice is transformed into a common non-truncated slice. Artifacts of this sort are particularly common when there is an imbalance in the distribution of slice sizes between P and A (which we try to avoid). Ideally, entire brain volumes would be used as inputs instead of slices as in this proof of concept.

6.5.3 Ablation study

We tested some of our model design choices on the MNIST and BRATS tasks.

Compressed skip connections. In Table 6.2, we compare (on MNIST 48x48 hard) different types of long skip connections: the proposed compressed skips ("Compress"), concatenation ("Concat"), summation ("Sum"), and no skips ("No skip"). We note that a significant segmentation improvement over baselines is retained for all skip types, with the proposed approach appearing to slightly outperform the others.

Shared decoder. We further compare, in Table 6.1, the use of a shared translation/segmentation decoder ("Proposed") to models with a separate segmentation decoder ("sep dec") for every task. Our results thus far demonstrate that we can share decoders; these ablation results demonstrate that doing so can yield a significant improvement in performance (MNIST 48x48 hard). On the other hand, it may be preferable to use a separate decoder as this yields better performance on MNIST 128x128. We conjecture that a shared decoder is useful when the model tends to overfit the data.

Table 6.2 Ablation studies of proposed method, using Dice scores for synthetic MNIST and real BraTS segmentation tasks: mean (standard deviation).

	MNIST 48×48 hard
Compress	0.57 (0.00)
Concat	0.57 (0.01)
Sum	0.56 (0.00)
No skip	0.42 (0.01)

6.6 Extensions and applications

Although we present work on two domains, P and A, we note that the proposed method can be easily extended to any greater number of domains. For example, if different types of pathology are known to be present in a medical image dataset, a domain-specific code (with a corresponding residual decoder) could be encoded for each pathology in addition to a neutral code with all pathologies absent. Most interestingly, our image-to-image translation approach would allow any number of pathologies to be present in an image at a time, unlike for example the StarGAN multi-domain image-to-image translation architecture [245].

Finally, we note that there are many different data outside of medical imaging that can be split into P and A domains. For example, any material fault analysis, such as rust detection, microchip defects, or the decay of building facades can be expressed in this way. Another interesting application may be the surveying of flood damage by learning the difference between pre-flood and post-flood urban aerial images. Extending the proposed method to more than two domains, one could explore such multi-domain problems as shadow segmentation where different times of day constitute different domains (with noon in A).

6.7 Conclusion

We propose a semi-supervised segmentation method that makes use of image-to-image translation in order to leverage unsegmented training data with cases presenting the object (P) of interest and cases in which it is absent (A). We argue that this objective is a good unsupervised surrogate for segmentation because it should similarly rely on disentangling of object variations from other variations. Indeed, we validate our method on both synthetic cluttered MNIST segmentation tasks and brain tumour segmentation in MR images, where we achieve significant improvement over supervised segmentation and a semi-supervised baseline.

CHAPTER 7 GENERAL DISCUSSION AND FUTURE WORK

This dissertation presented some exploration and development of deep neural network based tools for automated medical image segmentation, as well as some exploratory work on the issue of vanishing gradients in deep neural networks. Briefly, a liver tumour segmentation challenge and a corresponding entry to the challenge were presented in Chapter 4, illuminating the state of the art in liver tumour segmentation in CT. The clinical utility of this state of the art was evaluated, concluding that automated methods perform poorly on this task but can speed up manual segmentation if corrected by operators (when compared to segmenting by hand from scratch). In an effort to better understand and improve the fully convolutional networks (FCN) used for segmentation, gradient flow in an FCN was visualized in Chapter 5, motivating tweaks to the U-Net architecture. Following along this line of investigation, orthogonality constraints were analyzed in a recurrent neural network (RNN) test bed. This work confirmed the utility of orthogonality for avoiding vanishing gradients but motivated a spectral deviation from this constraint, yielding faster convergence and improved model performance. Finally, Chapter 6 addressed the common lack of a sufficient number of ground truth segmentations for medical images and the impracticality of getting more, proposing a novel semi-supervised framework that makes effective use of weak labels that are commonly available with medical images.

Each theme in the aforementioned chapters is accompanied with a discussion in those chapters, so those discussions will not be repeated here. Instead, this chapter provides further reflection on the work presented here and highlights future directions of research motivated by this work.

7.1 Improving fully convolutional networks

Fully convolutional networks (FCN) for image segmentation could be further improved in a number of ways. First, by combining some existing tricks such as the use of an adversarial discriminator, the addition of a shape prior, the addition of global context, the use of spectral constraints such as orthogonality, or a reduction in memory usage.

Adversarial training. Adversarial training of FCNs allows an objective function to be effectively learned by the discriminator instead of hand-crafted, improving segmentation performance [154–163].

Shape prior. A shape prior is useful for segmenting objects with known shape distribu-

tions, such as organs. While shape priors were common among classic segmentation methods (mainly relying on a statistical shape model [53, 54, 246]), incorporating such a prior in a convolutional neural network (CNN) proved challenging. However, [247] propose a method based on learning a generative model of atlas shapes with an autoencoder for imposing a shape prior on an FCN. Further work on shape priors would be useful.

Global context. Since pixels predictions in the decoder of an FCN are progressively fine, they are also progressively more local. Incorporating broader spatial information in the predictions is useful [105], especially for image generation [106]. It is less important for image segmentation because for that task, the decoder can infer the placement of predicted details by aligning them with salient features passed to it from the encoder at every resolution. Further work on global context would be interesting, whether by applying the non-local mechanism in [106] or by creating a new mechanism.

Spectral constraints. As shown in Chapter 5.2, constraining weight matrices to be near orthogonal can be useful for model performance. While most of this research has been done on RNNs, this has also been shown for fully connected feedforward neural network layers [47] and for CNNs [48, 248–250]. Orthogonal regularization has also been found to help with generative adversarial networks (GAN) [251, 252]. Applying such constraints to an FCN may prove useful. Furthermore, adopting other spectral constraints from the GAN literature, such as spectral normalization or renormalization [253, 254] may prove helpful for FCNs.

Memory efficiency. FCNs use a large amount of memory, limiting in practice the size of images that can be processed or the complexity of the model used to do the processing. There has been considerable work on reducing the computational and memory footprints of CNNs that can be adapted to FCNs [255–259]. Further work on memory efficiency would be welcomed by many.

Data efficiency. All deep neural networks require large amounts of data for reliable training. This is particularly a problem with medical image segmentation. Improved data efficiency would be very helpful in such cases and may indeed be necessary. This may require further research into task-specific unsupervised objective functions and new kinds of models with more useful inductive biases.

7.2 Stability and expressiveness

An orthogonality constraint on a recurrent neural network (RNN) transition matrix yields good stability but limits important dynamics, resulting in poor performance or convergence speed. Chapter 5.2 showed that a small deviation from orthogonality (deviation by up to

some margin m from singular values of one) can yield greatly improved convergence rates and better performance. Interestingly, this result can be observed with m as small as 0.001. Such a small deviation makes it unlikely that improved performance is due to a larger number of parameters covered by the transition matrix. Similarly, it is unlikely that during convergence, any shortcut is taken away from the manifold of orthogonal matrices when the transition matrix stays very close to this manifold. Loosening an orthogonality constraint is useful but it is unclear why.

A better angle from which to consider the described behaviour is that of dynamics. The dynamics induced by an orthogonal transition matrix are limited to limit cycles and lack chaotic or fixed point attractors. While these approaches work well for stable gradient propagation without vanishing gradients, they are not optimal for all tasks. In a basic RNN, an orthogonal matrix has been shown to be an optimal solution to the copy task where a short input sequence is outputted after some constant time delay [33]. However, if the time delay is varied, orthogonal RNNs fail [260]. This is likely because the state transition matrix works as a sort of clock mechanism that rotates the state by different amounts depending on the time delay but cannot hold the input fixed for different time delays [33]. Both chaotic [144] and, more commonly, fixed point attractors [32] are thought to be useful, with the latter thought to latch onto memories. On the other hand, [141] argue that they are unnecessary in gated networks like the LSTM and GRU since these networks can store memory in a separate memory cell [127, 261]. Nevertheless, [142] show that using attractor networks to clean up an RNN's state yields improved model performance not only for basic RNNs but also for GRUs. This motivates extending a gated model such as the GRU to benefit from the stability induced by orthogonality while retaining useful dynamics. Furthermore, it would be interesting to explore how to improve model dynamics and to elucidate the utility of chaos—especially since [144] show that chaotic RNNs may be more discriminative with fewer parameters.

Another plausible explanation for the results observed in Chapter 5.2 with deviation from orthogonality is that such a deviation allows for non-normal matrices, whereas all orthogonal matrices are normal matrices. Like orthogonal matrices, non-normal matrices can be constrained to have eigenvalues with unit norm, thus preventing vanishing gradients. However, their singular values are then greater than one, resulting in a polynomial (not exponential) growth in gradient norm across layers or RNN time steps. In addition, non-normal matrices produce transient dynamics that add expressivity that is useful for some tasks [46, 262].

7.3 Other ways to preserve gradients

While weight constraints in RNNs can help reduce vanishing gradients, they do not address the contraction caused by the nonlinearities such as *tanh*. The non-saturating unit (NRU) helps overcome this contraction by using only *ReLU* nonlinearities in a model inspired by neural turing machines [263]. Furthermore, the NRU also solves the issue of vanishing gradients at the gates of gated RNNs by using a linear or piece-wise linear memory addressing mechanism. That is, in the gated recurrent unit (GRU) [261] or long short term memory (LSTM), [127] gates use *sigmoid* nonlinearities that cause gradients to the gates to vanish when the gates are fully on or fully off; however, turning gates fully on or off is necessary to maximize gradient flow through the hidden states across time. The NRU removes this trade-off.

The issue of gradient propagation over long distances can also be sidestepped by using attention for credit assignment. The recent transformer model uses only self-attention and eschews recurrence and convolutions [264]. This model outperforms RNNs on many tasks and it remains to be seen if it will obsolete recurrent models. On the other hand, sparse attentive backprop outperforms the transformer on tasks with pathologically long term dependencies [49]. This model uses an attention mechanism on an RNN over all time steps in order to do credit assignment. As a result, gradients do not have to go through the RNN but instead take a short path through the attention mechanism. However, this attention is quadratic with sequence length in memory in its current implementation, making it difficult to scale it to tasks with long sequences.

7.4 Semi-supervised medical image segmentation

The semi-supervised segmentation approach presented in Chapter 6 is a proof of concept and can be extended. As proposed, this method expects data to be separated into two domains: *healthy* and *sick*; however, the model could be extended to handle more than two domains, such as when the sick cases contain multiple possible pathologies. This is the case with chest x-rays [265, 266], where an x-ray may present any number of pathologies at the same time. In this case, the model would have one latent code for *common* variations (including all healthy cases) and one *unique* latent code for each pathology. Furthermore, since the model is expected to disentangle the variations across all the codes, it may be useful to enforce a disentangling objective such as via continuous space neural mutual information estimation [267] or via a gradient reversal layer [268] as in [239]. Finally, further work will be to apply the proposed model to liver tumour segmentation.

7.5 Open research

It is important that research can be reproduced by others and that others may build on prior work. Therefore, much of the work presented in this dissertation is either released as freely licensed, publicly hosted open source code or will be released as such. These public code repositories are listed in Table 7.1, along with the chapters that they relate to.

Table 7.1 Publicly hosted, freely licensed code for each chapter of contributed work in the dissertation. Every repository URL follows "https://github.com/veugene/<URL suffix>".

Chapter	**Content**	**URL suffix**
Ch. 4.2	LiTS challenge FCN entry	lits
Ch. 4.2	Results evaluation for LiTS	lits-challenge-scoring/tree/miccai
Ch. 4.3	Clinical analysis	lits-challenge-scoring/tree/clinical_eval
Ch. 5.1	Fully convolutional networks	fcn_maker
Ch. 5.2	Orthogonality in RNNs	spectre_release
Ch. 6	Semi-supervised segmentation	(code pending public release)

In addition, in collaboration with the Centre hospitalier de l'Université de Montréal, we worked to contribute 55 cases of portal venous phase contrast enhanced abdominal CT volumes with colorectal metastases with expert segmentations to the LiTS challenge [1] dataset. This dataset includes 201 cases collected from seven sites around the world, as detailed in Chapter 4.2.

CHAPTER 8 CONCLUSION

Significant progress has been made in medical image segmentation with the emergence of deep convolutional neural networks; however, while some medical tasks exceed human performance [265], others such as the liver tumour segmentation in CT task explored in this dissertation fall far short. In this work, a benchmark segmentation challenge was presented and used to establish the state of the art in liver tumour segmentation. As a result, it became evident that current automated methods are inadequate compared to expert operators but can be clinically useful in that they can significantly reduce segmentation time if manually corrected. Further work is required to improve automated segmentation performance, especially on small lesions.

In an effort to improve segmentation robustness by using more data, a semi-supervised method tailored to the medical domain was proposed. This method uses weak labels of "sick" or "healthy" (*ie.* containing lesion or not) that are often readily available with medical data, whereas pixel-level annotations are difficult or impractical to collect. This work is a proof of concept and further work should be done to further improve its performance, extend it to more medical data, including liver lesion segmentation, and extend it to multi-domain data that presents more than one type of pathology.

Finally, further recommendations are proposed for improving current fully convolutional networks for segmentation. Among these is the potential application of the spectral constraint work presented in this dissertation, where it is shown that constraining network weights to be near orthogonal (but not exactly orthogonal) may be useful for faster convergence speed and improved performance. This work was done in a simple proof of concept manner and should be extended to segmentation networks; however, there are already promising works showing the extension of such constraints to convolutional neural networks.

REFERENCES

[1] P. Bilic, P. F. Christ, E. Vorontsov, G. Chlebus, H. Chen, Q. Dou, C.-W. Fu, X. Han, P.-A. Heng, J. Hesser *et al.*, "The liver tumor segmentation benchmark (lits)," *arXiv preprint arXiv:1901.04056*, 2019.

[2] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik, "Hypercolumns for object segmentation and fine-grained localization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 447–456.

[3] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3431–3440.

[4] M. F. Stollenga, W. Byeon, M. Liwicki, and J. Schmidhuber, "Parallel multi-dimensional lstm, with application to fast biomedical volumetric image segmentation," in *Advances in neural information processing systems*, 2015, pp. 2998–3006.

[5] E. Eisenhauer, P. Therasse, and J. B. et al., "New response evaluation criteria in solid tumours: revised recist guideline (version 1.1)," *European journal of cancer*, vol. 45, no. 2, pp. 228–247, 2009.

[6] J. Chapiro, R. Duran, M. Lin, R. E. Schernthaner, Z. Wang, B. Gorodetski, and J.-F. Geschwind, "Identifying staging markers for hepatocellular carcinoma before transarterial chemoembolization: comparison of three-dimensional quantitative versus non–three-dimensional imaging markers," *Radiology*, vol. 275, no. 2, pp. 438–447, 2015.

[7] A. Shimizu, T. Narihira, D. Furukawa, H. Kobatake, S. Nawano, and K. Shinozaki, "Ensemble segmentation using adaboost with application to liver lesion extraction from a ct volume," in *Proc. MICCAI Workshop on 3D Segmentation in the Clinic: A Grand Challenge II., NY, USA*, 2008.

[8] S. Bauer, T. Fejes, J. Slotboom, R. Wiest, L.-P. Nolte, and M. Reyes, "Segmentation of brain tumor images based on integrated hierarchical classification and regularization," in *MICCAI BraTS Workshop. Nice: Miccai Society*, 2012, p. 11.

[9] N. Tustison, M. Wintermark, C. Durst, and B. Avants, "Ants andarboles," *proc of BRATS Challenge - MICCAI 2013*, p. 47, 2013.

[10] Y. Häme, "Liver tumor segmentation using implicit surface evolution," *The Midas Journal*, pp. 1–10, 2008.

[11] H. Khotanlou, O. Colliot, J. Atif, and I. Bloch, "3d brain tumor segmentation in mri using fuzzy classification, symmetry analysis and spatially constrained deformable models," *Fuzzy sets and systems*, vol. 160, no. 10, pp. 1457–1473, 2009.

[12] A. Rajendran and R. Dhanasekaran, "Fuzzy clustering and deformable model for tumor segmentation on mri brain image: a combined approach," *Procedia Engineering*, vol. 30, pp. 327–333, 2012.

[13] E. Vorontsov, N. Abi-Jaoudeh, and S. Kadoury, "Metastatic liver tumor segmentation using texture-based omni-directional deformable surface models," in *International MICCAI Workshop on Computational and Clinical Challenges in Abdominal Imaging*. Springer, 2014, pp. 74–83.

[14] L. Rusko, G. Bekes, G. Nemeth, and M. Fidrich, "Fully automatic liver segmentation for contrast-enhanced ct images," *MICCAI Wshp. 3D Segmentation in the Clinic: A Grand Challenge*, vol. 2, no. 7, 2007.

[15] J. A. Dunnmon, D. Yi, C. P. Langlotz, C. Ré, D. L. Rubin, and M. P. Lungren, "Assessment of convolutional neural networks for automated classification of chest radiographs," *Radiology*, vol. 290, no. 2, pp. 537–544, 2019.

[16] E. Vorontsov, A. Tang, D. Roy, C. J. Pal, and S. Kadoury, "Metastatic liver tumour segmentation with a neural network-guided 3d deformable model," *Medical & biological engineering & computing*, vol. 55, no. 1, pp. 127–139, 2017.

[17] G. Chartrand, P. M. Cheng, E. Vorontsov, M. Drozdzal, S. Turcotte, C. J. Pal, S. Kadoury, and A. Tang, "Deep learning: a primer for radiologists," *Radiographics*, vol. 37, no. 7, pp. 2113–2131, 2017.

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

[19] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2015, pp. 234–241.

[20] K. Kamnitsas, W. Bai, E. Ferrante, S. McDonagh, M. Sinclair, N. Pawlowski, M. Rajchl, M. Lee, B. Kainz, D. Rueckert *et al.*, "Ensembles of multiple models and architectures for robust brain tumour segmentation," in *International MICCAI Brainlesion Workshop*. Springer, 2017, pp. 450–462.

[21] A. Myronenko, "3d mri brain tumor segmentation using autoencoder regularization," *arXiv preprint arXiv:1810.11654*, 2018.

[22] X. Han, "Automatic liver lesion segmentation using a deep convolutional neural network method," *arXiv preprint arXiv:1704.07239*, 2017.

[23] E. Vorontsov, A. Tang, C. Pal, and S. Kadoury, "Liver lesion segmentation informed by joint liver segmentation," *arXiv preprint arXiv:1707.07734v1*, 2017.

[24] G. Chlebus, H. Meine, J. Moltz, and A. Schenk, "Neural network-based automatic liver tumor segmentation with random forest-based candidate filtering," *arXiv preprint arXiv:1706.00842*, 2017.

[25] X. Li, H. Chen, X. Qi, Q. Dou, C. Fu, and P. Heng, "H-denseunet: Hybrid densely connected unet for liver and liver tumor segmentation from ct volumes," *arXiv preprint arXiv:1709.07330*, 2017.

[26] G. Chlebus, A. Schenk, J. H. Moltz, B. van Ginneken, H. K. Hahn, and H. Meine, "Automatic liver tumor segmentation in ct with fully convolutional neural networks and object-based postprocessing," *Scientific reports*, vol. 8, no. 1, pp. 1–7, 2018.

[27] X. Wang, S. Han, Y. Chen, D. Gao, and N. Vasconcelos, "Volumetric attention for 3d medical image segmentation and detection," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2019, pp. 175–184.

[28] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *European conference on computer vision*. Springer, 2016, pp. 630–645.

[29] A. M. Saxe, J. L. McClelland, and S. Ganguli, "Exact solutions to the nonlinear dynamics of learning in deep linear neural networks," *arXiv preprint arXiv:1312.6120*, 2013.

[30] Q. Le, N. Jaitly, and G. Hinton, "A simple way to initialize recurrent networks of rectified linear units," *arXiv:1504.00941*, 2015.

[31] D. Krueger and R. Memisevic, "Regularizing rnns by stabilizing activations," *arXiv preprint arXiv:1511.08400*, 2015.

[32] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks." *ICML (3)*, vol. 28, pp. 1310–1318, 2013.

[33] M. Henaff, A. Szlam, and Y. LeCun, "Recurrent orthogonal networks and long-memory tasks," *arXiv:1602.06662*, 2016.

[34] M. Arjovsky, A. Shah, and Y. Bengio, "Unitary evolution recurrent neural networks," in *ICML*, 2016, pp. 1120–1128.

[35] S. Wisdom, T. Powers, J. Hershey, J. L. Roux, and L. Atlas, "Full-capacity unitary recurrent neural networks," in *NeurIPS*, 2016, pp. 4880–4888.

[36] S. L. Hyland and G. Rätsch, "Learning unitary operators with help from u (n)." in *AAAI*, 2017, pp. 2050–2058.

[37] Z. Mhammedi, A. Hellicar, A. Rahman, and J. Bailey, "Efficient orthogonal parametrisation of recurrent neural networks using householder reflections," in *ICML*. JMLR.org, 2017, pp. 2401–2409.

[38] L. Jing, Y. Shen, T. Dubcek, J. Peurifoy, S. Skirlo, Y. LeCun, M. Tegmark, and M. Soljačić, "Tunable efficient unitary neural networks (eunn) and their application to rnns," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017, pp. 1733–1741.

[39] E. Vorontsov, C. Trabelsi, S. Kadoury, and C. Pal, "On orthogonality and learning recurrent networks with long term dependencies," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017, pp. 3570–3578.

[40] K. Helfrich, D. Willmott, and Q. Ye, "Orthogonal recurrent neural networks with scaled cayley transform," *arXiv:1707.09520*, 2017.

[41] K. Maduranga, K. Helfrich, and Q. Ye, "Complex unitary recurrent neural networks using scaled cayley transform," *arXiv:1811.04142*, 2018.

[42] M. Lezcano-Casado and D. Martínez-Rubio, "Cheap orthogonal constraints in neural networks: A simple parametrization of the orthogonal and unitary group," *arXiv:1901.08428*, 2019.

[43] L. Jing, C. Gulcehre, J. Peurifoy, Y. Shen, M. Tegmark, M. Soljacic, and Y. Bengio, "Gated orthogonal recurrent units: On learning to forget," *Neural computation*, vol. 31, no. 4, pp. 765–783, 2019.

[44] B. Chang, M. Chen, E. Haber, and E. H. Chi, "Antisymmetricrnn: A dynamical system view on recurrent neural networks," *arXiv preprint arXiv:1902.09689*, 2019.

[45] O. Ludwig, "Deep learning with eigenvalue decay regularizer," *arXiv preprint arXiv:1604.06985*, 2016.

[46] G. Kerg, K. Goyette, M. P. Touzel, G. Gidel, E. Vorontsov, Y. Bengio, and G. Lajoie, "Non-normal recurrent neural network (nnrnn): learning long time dependencies while improving expressivity with transient dynamics," in *Advances in Neural Information Processing Systems*, 2019, pp. 13 591–13 601.

[47] M. Harandi and B. Fernando, "Generalized backpropagation, étude de cas: orthogonality," *arXiv preprint arXiv:1611.05927*, 2016.

[48] N. Bansal, X. Chen, and Z. Wang, "Can we gain more from orthogonality regularizations in training deep networks?" in *Advances in Neural Information Processing Systems*, 2018, pp. 4261–4271.

[49] N. R. Ke, A. G. A. P. GOYAL, O. Bilaniuk, J. Binas, M. C. Mozer, C. Pal, and Y. Bengio, "Sparse attentive backtracking: Temporal credit assignment through reminding," in *Advances in neural information processing systems*, 2018, pp. 7640–7651.

[50] R. Adams and L. Bischof, "Seeded region growing," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 16, no. 6, pp. 641–647, 1994.

[51] L. Xu, Y. Zhu, Y. Zhang, and H. Yang, "Liver segmentation based on region growing and level set active contour model with new signed pressure force function," *Optik*, vol. 202, p. 163705, 2020.

[52] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *International journal of computer vision*, vol. 1, no. 4, pp. 321–331, 1988.

[53] T. F. Cootes, A. Hill, C. J. Taylor, and J. Haslam, "The use of active shape models for locating structures in medical images," in *Biennial International Conference on Information Processing in Medical Imaging*. Springer, 1993, pp. 33–47.

[54] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham, "Active shape models-their training and application," *Computer vision and image understanding*, vol. 61, no. 1, pp. 38–59, 1995.

[55] H. Lamecker, T. Lange, and M. Seebass, "Segmentation of the liver using a 3d statistical shape model," 2004.

[56] S. Osher and J. A. Sethian, "Fronts propagating with curvature-dependent speed: algorithms based on hamilton-jacobi formulations," *Journal of computational physics*, vol. 79, no. 1, pp. 12–49, 1988.

[57] C. Broit, "Optimal registration of deformed images," 1981.

[58] J. C. Gee, M. Reivich, and R. Bajcsy, "Elastically deforming a three-dimensional atlas to match anatomical brain images," 1993.

[59] D. Kainmueller, H. Lamecker, M. O. Heller, B. Weber, H.-C. Hege, and S. Zachow, "Omnidirectional displacements for deformable surfaces," *Medical image analysis*, vol. 17, no. 4, pp. 429–441, 2013.

[60] S. Geman and D. Geman, "Stochastic relaxation, gibbs distributions, and the bayesian restoration of images," *IEEE Transactions on pattern analysis and machine intelligence*, no. 6, pp. 721–741, 1984.

[61] J. Lafferty, A. McCallum, and F. C. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," 2001.

[62] X. He, R. S. Zemel, and M. Á. Carreira-Perpiñán, "Multiscale conditional random fields for image labeling," in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, vol. 2. IEEE, 2004, pp. II–II.

[63] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 23, no. 11, pp. 1222–1239, 2001.

[64] N. Komodakis, G. Tziritas, and N. Paragios, "Performance vs computational efficiency for optimizing single and dynamic mrfs: Setting the state of the art with primal-dual strategies," *Computer Vision and Image Understanding*, vol. 112, no. 1, pp. 14–29, 2008.

[65] J. Pearl, *Reverend Bayes on inference engines: A distributed hierarchical approach*. Cognitive Systems Laboratory, School of Engineering and Applied Science . . . , 1982.

[66] N. Komodakis, "Optimization algorithms for discrete markov random fields, with applications to computer vision," Ph.D. dissertation, Ph. D. dissertation, Computer Science Department, University of Crete, 2006.

[67] B. Van Ginneken, T. Heimann, and M. Styner, "3d segmentation in the clinic: A grand challenge," in *MICCAI Workshop on 3D Segmentation in the Clinic: A Grand Challenge*, vol. 1, 2007, pp. 7–15.

[68] A. Zareei and A. Karimi, "Liver segmentation with new supervised method to create initial curve for active contour," *Computers in Biology and Medicine*, vol. 75, pp. 139–150, 2016.

[69] Z. Zheng, X. Zhang, H. Xu, W. Liang, S. Zheng, and Y. Shi, "A unified level set framework combining hybrid algorithms for liver and liver tumor segmentation in ct images," *BioMed research international*, vol. 2018, 2018.

[70] G. Chartrand, T. Cresson, R. Chav, A. Gotra, A. Tang, and J. A. De Guise, "Liver segmentation on ct and mr using laplacian mesh optimization," *IEEE Transactions on Biomedical Engineering*, vol. 64, no. 9, pp. 2110–2121, 2016.

[71] X. Lu, Q. Xie, Y. Zha, and D. Wang, "Fully automatic liver segmentation combining multi-dimensional graph cut with shape information in 3d ct images," *Scientific reports*, vol. 8, no. 1, pp. 1–9, 2018.

[72] B. H. Menze, A. Jakab, S. Bauer, J. Kalpathy-Cramer, K. Farahani, J. Kirby, Y. Burren, N. Porz, J. Slotboom, R. Wiest *et al.*, "The multimodal brain tumor image segmentation benchmark (brats)," *IEEE transactions on medical imaging*, vol. 34, no. 10, pp. 1993–2024, 2014.

[73] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[74] K. Kamnitsas, L. Chen, C. Ledig, D. Rueckert, and B. Glocker, "Multi-scale 3d convolutional neural networks for lesion segmentation in brain mri," *Ischemic stroke lesion segmentation*, vol. 13, p. 46, 2015.

[75] K. Kamnitsas, C. Ledig, V. F. Newcombe, J. P. Simpson, A. D. Kane, D. K. Menon, D. Rueckert, and B. Glocker, "Efficient multi-scale 3d cnn with fully connected crf for accurate brain lesion segmentation," *Medical image analysis*, vol. 36, pp. 61–78, 2017.

[76] O. Maier, B. H. Menze, J. von der Gablentz, L. Häni, M. P. Heinrich, M. Liebrand, S. Winzeck, A. Basit, P. Bentley, L. Chen *et al.*, "Isles 2015-a public evaluation benchmark for ischemic stroke lesion segmentation from multispectral mri," *Medical image analysis*, vol. 35, pp. 250–269, 2017.

[77] M. Havaei, A. Davy, D. Warde-Farley, A. Biard, A. Courville, Y. Bengio, C. Pal, P.-M. Jodoin, and H. Larochelle, "Brain tumor segmentation with deep neural networks," *arXiv preprint arXiv:1505.03540*, 2015.

[78] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "Overfeat: Integrated recognition, localization and detection using convolutional networks," *arXiv preprint arXiv:1312.6229*, 2013.

[79] V. Dumoulin and F. Visin, "A guide to convolution arithmetic for deep learning," *arXiv preprint arXiv:1603.07285*, 2016.

[80] H. Chen, X. Qi, J. Cheng, and P. Heng, "Deep contextual networks for neuronal structure segmentation," in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*. AAAI Press, 2016, pp. 1167–1173.

[81] H. Chen, X. Qi, L. Yu, and P.-A. Heng, "Dcan: deep contour-aware networks for accurate gland segmentation," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2016, pp. 2487–2496.

[82] I. Arganda-Carreras, S. C. Turaga, D. R. Berger, D. Cireşan, A. Giusti, L. M. Gambardella, J. Schmidhuber, D. Laptev, S. Dwivedi, J. M. Buhmann *et al.*, "Crowdsourcing the creation of image segmentation algorithms for connectomics," *Frontiers in neuroanatomy*, vol. 9, p. 142, 2015.

[83] K. Sirinukunwattana, J. P. Pluim, H. Chen, X. Qi, P.-A. Heng, Y. B. Guo, L. Y. Wang, B. J. Matuszewski, E. Bruni, U. Sanchez *et al.*, "Gland segmentation in colon histology images: The glas challenge contest," *Medical image analysis*, vol. 35, pp. 489–502, 2017.

[84] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[85] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected crfs," *arXiv preprint arXiv:1412.7062*, 2014.

[86] M. Everingham, S. A. Eslami, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes challenge: A retrospective," *International journal of computer vision*, vol. 111, no. 1, pp. 98–136, 2015.

[87] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," *arXiv preprint arXiv:1511.07122*, 2015.

[88] S. Honari, J. Yosinski, P. Vincent, and C. Pal, "Recombinator networks: Learning coarse-to-fine feature aggregation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 5743–5752.

[89] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.

[90] M. Drozdzal, E. Vorontsov, G. Chartrand, S. Kadoury, and C. Pal, "The importance of skip connections in biomedical image segmentation," in *Deep Learning and Data Labeling for Medical Applications*. Springer, 2016, pp. 179–187.

[91] F. Milletari, N. Navab, and S. Ahmadi, "V-net: Fully convolutional neural networks for volumetric medical image segmentation," in *3D Vision (3DV), 2016 Fourth International Conference on*. IEEE, 2016, pp. 565–571.

[92] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1520–1528.

[93] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger, "3d u-net: learning dense volumetric segmentation from sparse annotation," in *International conference on medical image computing and computer-assisted intervention*. Springer, 2016, pp. 424–432.

[94] Z. Zhou, M. M. R. Siddiquee, N. Tajbakhsh, and J. Liang, "Unet++: A nested u-net architecture for medical image segmentation," in *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*. Springer, 2018, pp. 3–11.

[95] ——, "Unet++: Redesigning skip connections to exploit multiscale features in image segmentation," *IEEE Transactions on Medical Imaging*, 2019.

[96] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[97] T. Lan, Y. Li, J. K. Murugi, Y. Ding, and Z. Qin, "Run: residual u-net for computer-aided detection of pulmonary nodules without candidate selection," *arXiv preprint arXiv:1805.11856*, 2018.

[98] A. A. A. Setio, A. Traverso, T. De Bel, M. S. Berens, C. van den Bogaard, P. Cerello, H. Chen, Q. Dou, M. E. Fantacci, B. Geurts *et al.*, "Validation, comparison, and combination of algorithms for automatic detection of pulmonary nodules in computed tomography images: the luna16 challenge," *Medical image analysis*, vol. 42, pp. 1–13, 2017.

[99] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

[100] S. Jégou, M. Drozdzal, D. Vazquez, A. Romero, and Y. Bengio, "The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2017, pp. 11–19.

[101] P. Krähenbühl and V. Koltun, "Efficient inference in fully connected crfs with gaussian edge potentials," in *Advances in neural information processing systems*, 2011, pp. 109–117.

[102] P. Christ, M. Elshaer, and F. E. et al., "Automatic liver and lesion segmentation in ct using cascaded fully convolutional neural networks and 3d conditional random fields," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2016, pp. 415–423.

[103] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 4, pp. 834–848, 2017.

[104] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. Torr, "Conditional random fields as recurrent neural networks," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1529–1537.

[105] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," *arXiv preprint arXiv:1706.05587*, 2017.

[106] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, "Self-attention generative adversarial networks," *arXiv preprint arXiv:1805.08318*, 2018.

[107] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7794–7803.

[108] A. Graves, S. Fernández, and J. Schmidhuber, "Multi-dimensional recurrent neural networks," in *International conference on artificial neural networks*. Springer, 2007, pp. 549–558.

[109] W. Byeon, T. M. Breuel, F. Raue, and M. Liwicki, "Scene labeling with lstm recurrent neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3547–3555.

[110] F. Visin, K. Kastner, K. Cho, M. Matteucci, A. Courville, and Y. Bengio, "Renet: A recurrent neural network based alternative to convolutional networks," *arXiv preprint arXiv:1505.00393*, 2015.

[111] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural computation*, vol. 1, no. 4, pp. 541–551, 1989.

[112] F. Visin, K. Kastner, A. Courville, Y. Bengio, M. Matteucci, and K. Cho, "Reseg: A recurrent neural network for object segmentation," *arXiv preprint arXiv:1511.07053*, 2015.

[113] J. Dai, K. He, and J. Sun, "Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1635–1643.

[114] G. Papandreou, L.-C. Chen, K. P. Murphy, and A. L. Yuille, "Weakly-and semi-supervised learning of a deep convolutional network for semantic image segmentation," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1742–1750.

[115] S. Hong, H. Noh, and B. Han, "Decoupled deep neural network for semi-supervised semantic segmentation," in *Advances in neural information processing systems*, 2015, pp. 1495–1503.

[116] C. Baur, S. Albarqouni, and N. Navab, "Semi-supervised deep learning for fully convolutional networks," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2017, pp. 311–319.

[117] C. S. Perone and J. Cohen-Adad, "Deep semi-supervised segmentation with weight-averaged consistency targets," in *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*. Springer, 2018, pp. 12–19.

[118] N. Souly, C. Spampinato, and M. Shah, "Semi supervised semantic segmentation using generative adversarial network," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 5688–5696.

[119] Y. Zhang, L. Yang, J. Chen, M. Fredericksen, D. P. Hughes, and D. Z. Chen, "Deep adversarial networks for biomedical image segmentation utilizing unannotated images," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2017, pp. 408–416.

[120] H. Xiao, Y. Wei, Y. Liu, M. Zhang, and J. Feng, "Transferable semi-supervised semantic segmentation," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[121] C. Baur, B. Wiestler, S. Albarqouni, and N. Navab, "Deep autoencoding models for unsupervised anomaly segmentation in brain mr images," *arXiv preprint arXiv:1804.04488*, 2018.

[122] T. Schlegl, P. Seeböck, S. M. Waldstein, U. Schmidt-Erfurth, and G. Langs, "Unsupervised anomaly detection with generative adversarial networks to guide marker discovery," in *International Conference on Information Processing in Medical Imaging*. Springer, 2017, pp. 146–157.

[123] X. Chen and E. Konukoglu, "Unsupervised detection of lesions in brain mri using constrained adversarial auto-encoders," *arXiv preprint arXiv:1806.04972*, 2018.

[124] C. F. Baumgartner, L. M. Koch, K. C. Tezcan, J. X. Ang, and E. Konukoglu, "Visual feature attribution using wasserstein gans," *arXiv preprint arXiv:1711.08998*, 2017.

[125] S. Andermatt, A. Horváth, S. Pezold, and P. Cattin, "Pathology segmentation using distributional differences to images of healthy origin," *arXiv preprint arXiv:1805.10344*, 2018.

[126] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *IEEE International Conference on Computer Vision*, 2017.

[127] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[128] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE transactions on neural networks*, vol. 5, no. 2, pp. 157–166, 1994.

[129] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[130] S. Zagoruyko and N. Komodakis, "Wide residual networks," *arXiv preprint arXiv:1605.07146*, 2016.

[131] K. Zhang, M. Sun, T. X. Han, X. Yuan, L. Guo, and T. Liu, "Residual networks of residual networks: Multilevel residual networks," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 6, pp. 1303–1314, 2017.

[132] Q. Liao and T. Poggio, "Bridging the gaps between residual learning, recurrent neural networks and visual cortex," *arXiv preprint arXiv:1604.03640*, 2016.

[133] R. K. Srivastava, K. Greff, and J. Schmidhuber, "Highway networks," *arXiv preprint arXiv:1505.00387*, 2015.

[134] A. Veit, M. Wilber, and S. Belongie, "Residual networks are exponential ensembles of relatively shallow networks," *arXiv preprint arXiv:1605.06431*, vol. 1, no. 2, p. 3, 2016.

[135] G. Huang, Y. Sun, Z. Liu, D. Sedra, and K. Q. Weinberger, "Deep networks with stochastic depth," in *European conference on computer vision*. Springer, 2016, pp. 646–661.

[136] M. Abdi and S. Nahavandi, "Multi-residual networks: Improving the speed and accuracy of residual networks," *arXiv preprint arXiv:1609.05672*, 2016.

[137] K. Greff, R. K. Srivastava, and J. Schmidhuber, "Highway and residual networks learn unrolled iterative estimation," *arXiv preprint arXiv:1612.07771*, 2016.

[138] D. Sussillo and O. Barak, "Opening the black box: low-dimensional dynamics in high-dimensional recurrent neural networks," *Neural computation*, vol. 25, no. 3, pp. 626–649, 2013.

[139] Y. Bengio, P. Frasconi, and P. Simard, "The problem of learning long-term dependencies in recurrent networks," in *IEEE international conference on neural networks*. IEEE, 1993, pp. 1183–1188.

[140] S. Hochreiter, Y. Bengio, P. Frasconi, J. Schmidhuber *et al.*, "Gradient flow in recurrent nets: the difficulty of learning long-term dependencies," 2001.

[141] R. Jozefowicz, W. Zaremba, and I. Sutskever, "An empirical exploration of recurrent network architectures," in *ICML*, 2015, pp. 2342–2350.

[142] M. Mozer, D. Kazakov, and R. Lindsey, "State-denoised recurrent neural networks," *arXiv:1805.08394*, 2018.

[143] T. Laurent and J. Brecht, "A recurrent neural network without chaos," *arXiv:1612.06212*, 2016.

[144] M. Farrell, S. Recanatesi, G. Lajoie, and E. Shea-Brown, "Dynamic compression and expansion in a classifying recurrent network," *bioRxiv*, p. 564476, 2019.

[145] H. Sompolinsky, A. Crisanti, and H.-J. Sommers, "Chaos in random neural networks," *Physical review letters*, vol. 61, no. 3, p. 259, 1988.

[146] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[147] M. Frid-Adar, I. Diamant, E. Klang, M. Amitai, J. Goldberger, and H. Greenspan, "Gan-based synthetic medical image augmentation for increased cnn performance in liver lesion classification," *Neurocomputing*, vol. 321, pp. 321–331, 2018.

[148] D. Jin, Z. Xu, Y. Tang, A. P. Harrison, and D. J. Mollura, "Ct-realistic lung nodule simulation from 3d conditional generative adversarial networks for robust lung segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2018, pp. 732–740.

[149] J. T. Guibas, T. S. Virdi, and P. S. Li, "Synthetic medical images from dual generative adversarial networks," *arXiv preprint arXiv:1709.01872*, 2017.

[150] T. C. Mok and A. C. Chung, "Learning data augmentation for brain tumor segmentation with coarse-to-fine generative adversarial networks," in *International MICCAI Brainlesion Workshop*. Springer, 2018, pp. 70–80.

[151] M.-Y. Liu, T. Breuel, and J. Kautz, "Unsupervised image-to-image translation networks," in *Advances in Neural Information Processing Systems*, 2017, pp. 700–708.

[152] A. Almahairi, S. Rajeswar, A. Sordoni, P. Bachman, and A. Courville, "Augmented cyclegan: Learning many-to-many mappings from unpaired data," *arXiv preprint arXiv:1802.10151*, 2018.

[153] X. Huang, M.-Y. Liu, S. Belongie, and J. Kautz, "Multimodal unsupervised image-to-image translation," *arXiv preprint arXiv:1804.04732*, 2018.

[154] P. Luc, C. Couprie, S. Chintala, and J. Verbeek, "Semantic segmentation using adversarial networks," *arXiv preprint arXiv:1611.08408*, 2016.

[155] S. Kohl, D. Bonekamp, H.-P. Schlemmer, K. Yaqubi, M. Hohenfellner, B. Hadaschik, J.-P. Radtke, and K. Maier-Hein, "Adversarial networks for the detection of aggressive prostate cancer," *arXiv preprint arXiv:1702.08014*, 2017.

[156] W. Dai, J. Doyle, X. Liang, H. Zhang, N. Dong, Y. Li, and E. P. Xing, "Scan: Structure correcting adversarial network for chest x-rays organ segmentation," *arXiv preprint arXiv:1703.08770*, 2017.

[157] Y. Xue, T. Xu, H. Zhang, L. R. Long, and X. Huang, "Segan: Adversarial network with multi-scale l 1 loss for medical image segmentation," *Neuroinformatics*, vol. 16, no. 3-4, pp. 383–392, 2018.

[158] P. Moeskops, M. Veta, M. W. Lafarge, K. A. Eppenhof, and J. P. Pluim, "Adversarial training and dilated convolutions for brain mri segmentation," in *Deep learning in medical image analysis and multimodal learning for clinical decision support*. Springer, 2017, pp. 56–64.

[159] J. Son, S. J. Park, and K.-H. Jung, "Retinal vessel segmentation in fundoscopic images with generative adversarial networks," *arXiv preprint arXiv:1706.09318*, 2017.

[160] W. Zhu, X. Xiang, T. D. Tran, and X. Xie, "Adversarial deep structural networks for mammographic mass segmentation," *arXiv preprint arXiv:1612.05970*, 2016.

[161] D. Yang, D. Xu, S. K. Zhou, B. Georgescu, M. Chen, S. Grbic, D. Metaxas, and D. Comaniciu, "Automatic liver segmentation using an adversarial image-to-image network," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2017, pp. 507–515.

[162] M. Rezaei, H. Yang, and C. Meinel, "Whole heart and great vessel segmentation with context-aware of generative adversarial networks," in *Bildverarbeitung für die Medizin 2018*. Springer, 2018, pp. 353–358.

[163] Z. Li, Y. Wang, and J. Yu, "Brain tumor segmentation using an adversarial network," in *International MICCAI Brainlesion Workshop*. Springer, 2017, pp. 123–132.

[164] E. Vorontsov, A. Tang, C. Pal, and S. Kadoury, "Liver lesion segmentation informed by joint liver segmentation," in *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*. IEEE, 2018, pp. 1332–1335.

[165] R. Poudel, P. Lamata, and G. Montana, "Recurrent fully convolutional neural networks for multi-slice mri cardiac segmentation," *arXiv preprint arXiv:1608.03974*, 2016.

[166] T. Tieleman and G. Hinton, "Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude," *COURSERA: Neural networks for machine learning*, vol. 4, no. 2, pp. 26–31, 2012.

[167] M. Drozdzal, G. Chartrand, E. Vorontsov, L. D. Jorio, A. Tang, A. Romero, Y. Bengio, C. Pal, and S. Kadoury, "Learning normalized inputs for iterative estimation in medical image segmentation," *arXiv preprint arXiv:1702.05174*, 2017.

[168] R. L. Siegel, K. D. Miller, S. A. Fedewa, D. J. Ahnen, R. G. Meester, A. Barzi, and A. Jemal, "Colorectal cancer statistics, 2017," *CA: a cancer journal for clinicians*, vol. 67, no. 3, pp. 177–193, 2017.

[169] J. Ferlay, I. Soerjomataram, R. Dikshit, S. Eser, C. Mathers, M. Rebelo, D. M. Parkin, D. Forman, and F. Bray, "Cancer incidence and mortality worldwide: sources, methods and major patterns in globocan 2012," *International journal of cancer*, vol. 136, no. 5, pp. E359–E386, 2015.

[170] M. Donadon, D. Ribero, G. Morris-Stiff, E. K. Abdalla, and J.-N. Vauthey, "New paradigm in the management of liver-only metastases from colorectal cancer," *Gastrointestinal cancer research: GCR*, vol. 1, no. 1, p. 20, 2007.

[171] R. Siegel, C. DeSantis, K. Virgo, K. Stein, A. Mariotto, T. Smith, D. Cooper, T. Gansler, C. Lerro, S. Fedewa *et al.*, "Cancer treatment and survivorship statistics, 2012," *CA: a cancer journal for clinicians*, vol. 62, no. 4, pp. 220–241, 2012.

[172] R. Adam, A. de Gramont, J. Figueras, N. Kokudo, F. Kunstlinger, E. Loyer, G. Poston, P. Rougier, L. Rubbia-Brandt, A. Sobrero *et al.*, "Managing synchronous liver metastases from colorectal cancer: a multidisciplinary international consensus," *Cancer treatment reviews*, vol. 41, no. 9, pp. 729–741, 2015.

[173] M. Hwang, T. T. Jayakrishnan, D. E. Green, B. George, J. P. Thomas, R. T. Groeschl, B. Erickson, S. G. Pappas, T. C. Gamblin, and K. K. Turaga, "Systematic review of outcomes of patients undergoing resection for colorectal liver metastases in the setting of extra hepatic disease," *European journal of cancer*, vol. 50, no. 10, pp. 1747–1757, 2014.

[174] J. S. Tomlinson, W. R. Jarnagin, R. P. DeMatteo, Y. Fong, P. Kornprat, M. Gonen, N. Kemeny, M. F. Brennan, L. H. Blumgart, and M. D'Angelica, "Actual 10-year survival after resection of colorectal liver metastases defines cure," *Journal of Clinical Oncology*, vol. 25, no. 29, pp. 4575–4580, 2007.

[175] D. J. Gallagher and N. Kemeny, "Metastatic colorectal cancer: from improved survival to potential cure," *Oncology*, vol. 78, no. 3-4, pp. 237–248, 2010.

[176] S. H. Tirumani, K. W. Kim, M. Nishino, S. A. Howard, K. M. Krajewski, J. P. Jagannathan, J. M. Cleary, N. H. Ramaiya, and A. B. Shinagare, "Update on the role of imaging in management of metastatic colorectal cancer," *Radiographics*, vol. 34, no. 7, pp. 1908–1928, 2014.

[177] K. Imai, M.-A. Allard, C. C. Benitez, E. Vibert, A. S. Cunha, D. Cherqui, D. Castaing, H. Bismuth, H. Baba, and R. Adam, "Early recurrence after hepatectomy for colorectal liver metastases: what optimal definition and what predictive factors?" *The oncologist*, vol. 21, no. 7, p. 887, 2016.

[178] R. E. Schwarz, E. K. Abdalla, T. A. Aloia, and J.-N. Vauthey, "Ahpba/sso/ssat sponsored consensus conference on the multidisciplinary treatment of colorectal cancer metastases," *HPB*, vol. 15, no. 2, pp. 89–90, 2013.

[179] M. Mantatzis, S. Kakolyris, K. Amarantidis, A. Karayiannakis, and P. Prassopoulos, "Treatment response classification of liver metastatic disease evaluated on imaging. are recist unidimensional measurements accurate?" *European radiology*, vol. 19, no. 7, pp. 1809–1816, 2009.

[180] J. H. Rothe, C. Grieser, L. Lehmkuhl, D. Schnapauff, C. P. Fernandez, M. H. Maurer, A. Mussler, B. Hamm, T. Denecke, and I. G. Steffen, "Size determination and response assessment of liver metastases with computed tomography—comparison of recist and volumetric algorithms," *European Journal of Radiology*, vol. 82, no. 11, pp. 1831–1839, 2013.

[181] G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, J. A. Van Der Laak, B. Van Ginneken, and C. I. Sánchez, "A survey on deep learning in medical image analysis," *Medical image analysis*, vol. 42, pp. 60–88, 2017.

[182] M. Moghbel, S. Mashohor, R. Mahmud, and M. I. B. Saripan, "Review of liver segmentation and computer assisted detection/diagnosis methods in computed tomography," *Artificial Intelligence Review*, vol. 50, no. 4, pp. 497–537, 2018.

[183] A. Gotra, L. Sivakumaran, G. Chartrand, K.-N. Vu, F. Vandenbroucke-Menu, C. Kauffmann, S. Kadoury, B. Gallix, J. A. de Guise, and A. Tang, "Liver segmentation: indications, techniques and future directions," *Insights into imaging*, vol. 8, no. 4, pp. 377–392, 2017.

[184] B. Norman, V. Pedoia, and S. Majumdar, "Use of 2d u-net convolutional neural networks for automated cartilage and meniscus segmentation of knee mr imaging data to determine relaxometry and morphometry," *Radiology*, vol. 288, no. 1, pp. 177–185, 2018.

[185] K. Yasaka, H. Akai, O. Abe, and S. Kiryu, "Deep learning with convolutional neural network for differentiation of liver masses at dynamic contrast-enhanced ct: a preliminary study," *Radiology*, vol. 286, no. 3, pp. 887–896, 2018.

[186] "Midas liver tumor dataset from national library of medicines imaging methods assessment and reporting (imar) project," http://hdl.handle.net/1926/1721, 2016.

[187] "3dircadb dataset from research institute against digestive cancer."

[188] "Lits - liver tumor segmentation challenge," https://competitions.codalab.org/competitions/17094, 2017.

[189] "Cancer imaging archive of the frederick national laboratory for cancer research cir dataset," www.cancerimagingarchive.net.

[190] "Sliver'07 dataset from the medical image computing and computer assisted intervention society miccai liver segmentation challenge," http://www.sliver07.org/, 2007.

[191] "Canadian tissue repository network," http://www.ctrnet.ca.

[192] "The medical imaging interaction toolkit (mitk)," http://mitk.org/wiki/The_Medical_Imaging_Interaction_Toolkit_(MITK).

[193] P. M. Silverman, "Liver metastases: imaging considerations for protocol development with multislice ct (msct)," *Cancer Imaging*, vol. 6, no. 1, p. 175, 2006.

[194] A. Gotra, G. Chartrand, K. Massicotte-Tisluck, F. Morin-Roy, F. Vandenbroucke-Menu, J. A. de Guise, and A. Tang, "Validation of a semiautomated liver segmentation method using ct for accurate volumetry," *Academic radiology*, vol. 22, no. 9, pp. 1088–1098, 2015.

[195] H. De Vries, M. N. Elliott, D. E. Kanouse, and S. S. Teleki, "Using pooled kappa to summarize interrater agreement across many items," *Field Methods*, vol. 20, no. 3, pp. 272–282, 2008.

[196] G. Zou, "Confidence interval estimation for the bland–altman limits of agreement with multiple observations per individual," *Statistical methods in medical research*, vol. 22, no. 6, pp. 630–642, 2013.

[197] A. Tang, R. Tam, A. Cadrin-Chênevert, W. Guest, J. Chong, J. Barfett, L. Chepelev, R. Cairns, J. R. Mitchell, M. D. Cicero *et al.*, "Canadian association of radiologists white paper on artificial intelligence in radiology," *Canadian Association of Radiologists' Journal*, vol. 69, no. 2, pp. 120–135, 2018.

[198] Y. Ko, J. Kim, J. K.-H. Park, H. Kim, J. Y. Cho, S.-B. Kang, S. Ahn, K. J. Lee, and K. H. Lee, "Limited detection of small ($\leq$ 10 mm) colorectal liver metastasis at preoperative ct in patients undergoing liver resection," *PloS one*, vol. 12, no. 12, 2017.

[199] M. C. Niekel, S. Bipat, and J. Stoker, "Diagnostic imaging of colorectal liver metastases with ct, mr imaging, fdg pet, and/or fdg pet/ct: a meta-analysis of prospective studies including patients who have not previously undergone treatment," *Radiology*, vol. 257, no. 3, pp. 674–684, 2010.

[200] A. R. van Erkel, M. E. Pijl, A. A. van den Berg-Huysmans, M. N. Wasser, C. J. van de Velde, and J. L. Bloem, "Hepatic metastases in patients with colorectal cancer: relationship between size of metastases, standard of reference, and detection rates," *Radiology*, vol. 224, no. 2, pp. 404–409, 2002.

[201] A. Ben-Cohen, I. Diamant, E. Klang, M. Amitai, and H. Greenspan, "Fully convolutional network for liver segmentation and lesions detection," in *Deep learning and data labeling for medical applications*. Springer, 2016, pp. 77–85.

[202] M. G. Linguraru, W. J. Richbourg, J. Liu, J. M. Watt, V. Pamulapati, S. Wang, and R. M. Summers, "Tumor burden analysis on computed tomography by automated liver

and tumor segmentation," *IEEE transactions on medical imaging*, vol. 31, no. 10, pp. 1965–1976, 2012.

[203] W. Li *et al.*, "Automatic segmentation of liver tumor in ct images with deep convolutional neural networks," *Journal of Computer and Communications*, vol. 3, no. 11, p. 146, 2015.

[204] P. F. Christ, F. Ettlinger, F. Grün, M. E. A. Elshaera, J. Lipkova, S. Schlecht, F. Ahmaddy, S. Tatavarty, M. Bickel, P. Bilic *et al.*, "Automatic liver and tumor segmentation of ct and mri volumes using cascaded fully convolutional neural networks," *arXiv preprint arXiv:1702.05970*, 2017.

[205] M. Moghbel, S. Mashohor, R. Mahmud, and M. I. B. Saripan, "Automatic liver tumor segmentation on computed tomography for patient treatment planning and monitoring," *EXCLI journal*, vol. 15, p. 406, 2016.

[206] M. H. Albrecht, J. L. Wichmann, C. Müller, T. Schreckenbach, S. Sakthibalan, R. Hammerstingl, W. O. Bechstein, S. Zangos, H. Ackermann, and T. J. Vogl, "Assessment of colorectal liver metastases using mri and ct: impact of observer experience on diagnostic performance and inter-observer reproducibility with histopathological correlation," *European journal of radiology*, vol. 83, no. 10, pp. 1752–1758, 2014.

[207] J. K. Udupa, V. R. LeBlanc, Y. Zhuge, C. Imielinska, H. Schmidt, L. M. Currie, B. E. Hirsch, and J. Woodburn, "A framework for evaluating image segmentation algorithms," *Computerized medical imaging and graphics*, vol. 30, no. 2, pp. 75–87, 2006.

[208] S. Bakas, M. Reyes, A. Jakab, S. Bauer, M. Rempfler, A. Crimi, R. T. Shinohara, C. Berger, S. M. Ha, M. Rozycki *et al.*, "Identifying the best machine learning algorithms for brain tumor segmentation, progression assessment, and overall survival prediction in the brats challenge," *arXiv preprint arXiv:1811.02629*, 2018.

[209] M. Styner, J. Lee, B. Chin, M. Chin, O. Commowick, H. Tran, S. Markovic-Plese, V. Jewells, and S. Warfield, "3d segmentation in the clinic: A grand challenge ii: Ms lesion segmentation," *Midas Journal*, vol. 2008, pp. 1–6, 2008.

[210] T. Brosch, L. Y. Tang, Y. Yoo, D. K. Li, A. Traboulsee, and R. Tam, "Deep 3d convolutional encoder networks with shortcuts for multiscale feature integration applied to multiple sclerosis lesion segmentation," *IEEE transactions on medical imaging*, vol. 35, no. 5, pp. 1229–1239, 2016.

[211] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[212] A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio, "Fitnets: Hints for thin deep nets," *arXiv preprint arXiv:1412.6550*, 2014.

[213] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Thirty-first AAAI conference on artificial intelligence*, 2017.

[214] F. Chollet, "Keras," https://github.com/fchollet/keras, 2015.

[215] R. Al-Rfou, G. Alain, A. Almahairi, C. Angermueller, D. Bahdanau, N. Ballas, F. Bastien, J. Bayer, A. Belikov, A. Belopolsky *et al.*, "Theano: A python framework for fast computation of mathematical expressions," *arXiv preprint arXiv:1605.02688*, 2016.

[216] A. Kendall, V. Badrinarayanan, and R. Cipolla, "Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding," *arXiv preprint arXiv:1511.02680*, 2015.

[217] D. Ciresan, A. Giusti, L. M. Gambardella, and J. Schmidhuber, "Deep neural networks segment neuronal membranes in electron microscopy images," in *Advances in neural information processing systems*, 2012, pp. 2843–2851.

[218] X. Wu, "An iterative convolutional neural network algorithm improves electron microscopy image segmentation," *arXiv preprint arXiv:1506.05849*, 2015.

[219] T. Liu, C. Jones, M. Seyedhosseini, and T. Tasdizen, "A modular hierarchical approach to 3d electron microscopy image segmentation," *Journal of neuroscience methods*, vol. 226, pp. 88–102, 2014.

[220] M. G. Uzunbaş, C. Chen, and D. Metaxsas, "Optree: a learning-based adaptive watershed algorithm for neuron segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2014, pp. 97–105.

[221] A. M. Saxe, P. W. Koh, Z. Chen, M. Bhand, B. Suresh, and A. Y. Ng, "On random weights and unsupervised feature learning." in *ICML*, vol. 2, no. 3, 2011, p. 6.

[222] M. Henaff, A. Szlam, and Y. LeCun, "Orthogonal rnns and long-memory tasks," *arXiv preprint arXiv:1602.06662*, 2016.

[223] M. Arjovsky, A. Shah, and Y. Bengio, "Unitary evolution recurrent neural networks," *arXiv preprint arXiv:1511.06464*, 2015.

[224] L. Jing, Y. Shen, T. Dubček, J. Peurifoy, S. Skirlo, M. Tegmark, and M. Soljačić, "Tunable efficient unitary neural networks (eunn) and their application to rnn," *arXiv preprint arXiv:1612.05231*, 2016.

[225] Y. Nishimori, "A note on riemannian optimization methods on the stiefel and the grassmann manifolds," *dim*, vol. 1, p. 2, 2005.

[226] H. D. Tagare, "Notes on optimization on stiefel manifolds," Tech. Rep., Yale University, Tech. Rep., 2011.

[227] S. Wisdom, T. Powers, J. Hershey, J. Le Roux, and L. Atlas, "Full-capacity unitary recurrent neural networks," in *Advances in Neural Information Processing Systems*, 2016, pp. 4880–4888.

[228] Z. Mhammedi, A. Hellicar, A. Rahman, and J. Bailey, "Efficient orthogonal parametrisation of recurrent neural networks using householder reflections," *arXiv preprint arXiv:1612.00188*, 2016.

[229] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[230] M. P. Marcus, M. A. Marcinkiewicz, and B. Santorini, "Building a large annotated corpus of english: The penn treebank," *Computational linguistics*, vol. 19, no. 2, pp. 313–330, 1993.

[231] V. Nair and G. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 807–814.

[232] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1026–1034.

[233] A. Chernodub and D. Nowicki, "Norm-preserving orthogonal permutation linear unit activation functions (oplu)," *arXiv preprint arXiv:1604.02313*, 2016.

[234] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks." in *Aistats*, vol. 9, 2010, pp. 249–256.

[235] L. Jing, C. Gulcehre, J. Peurifoy, Y. Shen, M. Tegmark, M. Soljačić, and Y. Bengio, "Gated orthogonal recurrent units: On learning to forget," *arXiv preprint arXiv:1706.02761*, 2017.

[236] E. Vorontsov, P. Molchanov, C. Beckham, W. Byeon, S. De Mello, V. Jampani, M.-Y. Liu, S. Kadoury, and J. Kautz, "Towards semi-supervised segmentation via image-to-image translation," *arXiv preprint arXiv:1904.01636*, 2020.

[237] S. Izadi, Z. Mirikharaji, J. Kawahara, and G. Hamarneh, "Generative adversarial networks to segment skin lesions," in *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*. IEEE, 2018, pp. 881–884.

[238] H.-Y. Lee, H.-Y. Tseng, J.-B. Huang, M. Singh, and M.-H. Yang, "Diverse image-to-image translation via disentangled representations," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 35–51.

[239] A. Gonzalez-Garcia, J. van de Weijer, and Y. Bengio, "Image-to-image translation for cross-domain disentanglement," *arXiv preprint arXiv:1805.09730*, 2018.

[240] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *International Conference on Machine Learning*, 2015, pp. 1180–1189.

[241] B. Schölkopf, D. Janzing, J. Peters, E. Sgouritsa, K. Zhang, and J. Mooij, "On causal and anticausal learning," in *Proceedings of the 29th International Coference on International Conference on Machine Learning*. Omnipress, 2012, pp. 459–466.

[242] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Instance normalization: The missing ingredient for fast stylization," *arXiv preprint arXiv:1607.08022*, 2016.

[243] M. Jaderberg, K. Simonyan, A. Zisserman *et al.*, "Spatial transformer networks," in *Advances in neural information processing systems*, 2015, pp. 2017–2025.

[244] J. P. Cohen, M. Luck, and S. Honari, "Distribution matching losses can hallucinate features in medical image translation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2018, pp. 529–536.

[245] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo, "Stargan: Unified generative adversarial networks for multi-domain image-to-image translation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8789–8797.

[246] T. F. Cootes, G. J. Edwards, and C. J. Taylor, "Active appearance models," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 23, no. 6, pp. 681–685, 2001.

[247] O. Oktay, E. Ferrante, K. Kamnitsas, M. Heinrich, W. Bai, J. Caballero, S. A. Cook, A. De Marvao, T. Dawes, D. P. O'Regan *et al.*, "Anatomically constrained neural networks (acnns): application to cardiac image enhancement and segmentation," *IEEE transactions on medical imaging*, vol. 37, no. 2, pp. 384–395, 2017.

[248] K. Jia, S. Li, Y. Wen, T. Liu, and D. Tao, "Orthogonal deep neural networks," *IEEE transactions on pattern analysis and machine intelligence*, 2019.

[249] M. Hasannasab, J. Hertrich, S. Neumayer, G. Plonka, S. Setzer, and G. Steidl, "Parseval proximal neural networks," *arXiv preprint arXiv:1912.10480*, 2019.

[250] Z. Zhang, W. Ma, Y. Wu, and G. Wang, "Self-orthogonality module: A network architecture plug-in for learning orthogonal filters," *arXiv preprint arXiv:2001.01275*, 2020.

[251] A. Brock, J. Donahue, and K. Simonyan, "Large scale gan training for high fidelity natural image synthesis," *arXiv preprint arXiv:1809.11096*, 2018.

[252] J. Müller, R. Klein, and M. Weinmann, "Orthogonal wasserstein gans," *arXiv preprint arXiv:1911.13060*, 2019.

[253] Y. Yoshida and T. Miyato, "Spectral norm regularization for improving the generalizability of deep learning," *arXiv preprint arXiv:1705.10941*, 2017.

[254] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral normalization for generative adversarial networks," *arXiv preprint arXiv:1802.05957*, 2018.

[255] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[256] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510–4520.

[257] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun, "Shufflenet v2: Practical guidelines for efficient cnn architecture design," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 116–131.

[258] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7132–7141.

[259] M. Tan and Q. V. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," *arXiv preprint arXiv:1905.11946*, 2019.

[260] S. Chandar, C. Sankar, E. Vorontsov, S. E. Kahou, and Y. Bengio, "Towards non-saturating recurrent units for modelling long-term dependencies," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 3280–3287.

[261] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

[262] A. E. Orhan and X. Pitkow, "Improved memory in recurrent neural networks with sequential non-normal dynamics," *arXiv preprint arXiv:1905.13715*, 2019.

[263] A. Graves, G. Wayne, and I. Danihelka, "Neural turing machines," *arXiv preprint arXiv:1410.5401*, 2014.

[264] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[265] P. Rajpurkar, J. Irvin, K. Zhu, B. Yang, H. Mehta, T. Duan, D. Ding, A. Bagul, C. Langlotz, K. Shpanskaya *et al.*, "Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning," *arXiv preprint arXiv:1711.05225*, 2017.

[266] J. Irvin, P. Rajpurkar, M. Ko, Y. Yu, S. Ciurea-Ilcus, C. Chute, H. Marklund, B. Haghgoo, R. Ball, K. Shpanskaya *et al.*, "Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 590–597.

[267] M. I. Belghazi, A. Baratin, S. Rajeswar, S. Ozair, Y. Bengio, A. Courville, and R. D. Hjelm, "Mine: mutual information neural estimation," *arXiv preprint arXiv:1801.04062*, 2018.

[268] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," *arXiv preprint arXiv:1409.7495*, 2014.

[269] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," in *NIPS-W*, 2017.

[270] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.

[271] X. Huang and S. Belongie, "Arbitrary style transfer in real-time with adaptive instance normalization," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1501–1510.

[272] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, "High-resolution image synthesis and semantic manipulation with conditional gans," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8798–8807.

[273] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. icml*, vol. 30, no. 1, 2013, p. 3.

[274] S. J. Reddi, S. Kale, and S. Kumar, "On the convergence of adam and beyond," 2018.

[275] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," in *Advances in Neural Information Processing Systems*, 2017, pp. 6626–6637.

APPENDIX A DEEP LEARNING FOR AUTOMATED SEGMENTATION OF LIVER LESIONS ON COMPUTED TOMOGRAPHY IN PATIENTS WITH COLORECTAL CANCER LIVER METASTASES

A.1 Technical Details on Model Architecture

We trained the model only on 2D axial sections that contain the liver, using RMSprop [166] with the Dice loss, with extensive data augmentation using horizontal and vertical flips, rotations, zooming, and elastic deformations. We followed a three-stage training process:

Step 1.
Training for 200 epochs (batch size 40) with 0.001 learning rate (0.9 momentum) on down-scaled 256 $\times$ 256 sections.

Step 2.
Fine-tuning for 30 epochs (batch size 10) on full resolution 512 $\times$ 512 sections, at a 0.0001 learning rate.

Step 3.
Combining logit predictions for every set of three consecutive sections using a new 3 $\times$ 3 convolution kernel and training this kernel and a new classifier for the middle section. Step 3 integrates information from neighboring sections.

At test time, all predictions were averaged across all four input orientations achieved by vertical and horizontal flips, as well as across three similar models, resulting in an ensemble of 12 predictions. Liver segmentation postprocessing was limited to selecting one largest connected component. Lesion segmentation postprocessing was limited to discarding lesion predictions 20 mm beyond the boundaries of the predicted liver.

A.2 Supplementary results

Table A.1 Representative CT imaging technique.

Parameters	Philips Brilliance 64	GE Lightspeed 16
Rotation time	0.75	0.8 s
Detector collimation	64 × 0.625 mm	16 × 1.25 mm
Helical pitch	0.891	1.375
Tube voltage	120 kV	120–140 kV
Tube current	126–499 mA	75–440 mA
Section thickness	2.5 mm	2.5 mm
Section gap	2 mm	2 mm
Matrix	512 × 512 pixels	512 × 512 pixels
Pixel spacing	0.55 and 0.78 mm	0.55 and 0.78 mm
Reconstruction method	Iterative reconstruction (iDose)	Iterative reconstruction (ASIR)
Reconstruction kernel	Standard-B	Standard

Table A.2 Segmentation performance measures.

Measure	Definition	Formula
Dice similarity coefficient (DSC)	Dice similarity coefficient (DSC) (50) is a spatial overlap index between two segmentation results (*A* = automated segmentation, *M* = reference standard) and a reproducibility validation metric that ranges from 0 (indicating no spatial overlap between two segmentation results) to 1 (indicating a complete overlap).	DSC*(A,M)*= 2*(A*∩*M)/(A+M)* where ∩ is the intersection
Jaccard index	The Jaccard index (51) is a spatial overlap index between two segmentation results (*A* = automated segmentation, *M* = reference standard) and a reproducibility validation metric that ranges from 0 (indicating no spatial overlap between two segmentation results) to 1 (indicating a complete overlap).	J *(A,M)* = *(A*∩*M)/(A* ∪ *M)* = DSC *(A,M)*/(2–DSC *(A,M)*)
Average Symmetric Surface Distance (ASSD)	Surface distance based measures quantify the distance in mm of the disparity between the automated segmentation (A) and the reference standard (M). ASSD is the average of all the distance from points of the boundary of automated segmentation to standard of reference boundary.	$\mathrm{ASSD}(A,M)=\frac{1}{\lvert S(A)\rvert+\lvert S(M)\rvert}\left(\sum_{S_A\in S(A)} d(S_A,S(M))+\sum_{S_M\in S(M)} d(S_M,S(A))\right)$
Maximum Symmetric Surface Distance (MSSD)	MSSD is the maximal distance found from a point of automated segmentation to one of standard of reference boundary.	$MSSD(A,M)=\max\left\{max_{s_A\in S(A)}d(s_A,S(M)),max_{s_M\in S(M)}d(s_M,S(A))\right\}$

Table A.3 Detection performance at minimum overlap > 0.25
Data in parentheses are 95% confidence intervals. A = automated segmentation, C^i = user-corrected segmentation by reader i, M^i = manual segmentations by reader i, FN = false negative, FP = false positive, TN = true negative, TP = true positive. * Number of true-negative findings was not reported, because there is a potentially very high number of nonlesional pixels.

Method	Users	No. of Observations				Per patient		Per lesion	
		TP	FN	FP	TN*		< 10 mm	10–20 mm	> 20 mm
Sensitivity									
Manual	M[1]	70	36	26	—	0.82 (0.77, 0.87)	0.57 (0.44, 0.69)	0.83 (0.74, 0.92)	1.00 (1.00, 1.00)
	M[2]	68	36	34	—	0.82 (0.77, 0.87)	0.65 (0.53, 0.77)	0.81 (0.73, 0.91)	0.95 (0.91, 1.00)
User-corrected	C[1]	53	52	42	—	0.78 (0.72, 0.83)	0.48 (0.36, 0.61)	0.80 (0.71, 0.90)	0.98 (0.95, 1.00)
	C[2]	54	50	30	—	0.73 (0.67, 0.79)	0.30 (0.18, 0.41)	0.80 (0.71, 0.90)	0.99 (0.98, 1.00)
Automated	—	31	74	49	—	0.44 (0.34, 0.53)	0.00 (0.00, 0.00)	0.49 (0.32, 0.65)	0.72 (0.60, 0.86)
Positive predictive value									
Manual	M[1]	70	36	26	—	0.91 (0.87, 0.95)	0.74 (0.62, 0.88)	0.92 (0.86, 0.99)	1.00 (1.00, 1.00)
	M[2]	68	36	34	—	0.84 (0.80, 0.89)	0.63 (0.51, 0.75)	0.88 (0.80, 0.96)	0.99 (0.97, 1.00)
User-corrected	C[1]	53	52	42	—	0.88 (0.83, 0.92)	0.60 (0.47, 0.74)	0.93 (0.88, 1.00)	1.00 (1.00, 1.00)
	C[2]	54	50	30	—	0.91 (0.87, 0.96)	0.62 (0.44, 0.81)	0.93 (0.88, 1.00)	1.00 (1.00, 1.00)
Automated	—	31	74	49	—	0.58 (0.48, 0.69)	0.00 (0.00, 0.00)	0.68 (0.50, 0.88)	0.94 (0.87, 1.00)

Table A.4 Detection performance at minimum overlap > 0.5
Data in parentheses are 95% confidence intervals. A = automated segmentation, C^i = user-corrected segmentation by reader i, M^i = manual segmentations by reader i, FN = false negative, FP = false positive, TN = true negative, TP = true positive. * Number of true-negative findings was not reported, because there is a potentially very high number of nonlesional pixels.

Method	Users	No. of Observations				Per patient		Per lesion	
		TP	FN	FP	TN*		< 10 mm	10–20 mm	> 20 mm
Sensitivity									
Manual	M[1]	70	36	26	—	0.66 (0.60, 0.73)	0.30 (0.18, 0.41)	0.67 (0.56, 0.78)	0.92 (0.88, 0.99)
	M[2]	68	36	34	—	0.65 (0.59, 0.72)	0.35 (0.23, 0.47)	0.59 (0.47, 0.70)	0.94 (0.89, 0.99)
User-corrected	C[1]	53	52	42	—	0.50 (0.44, 0.57)	0.32 (0.20, 0.43)	0.39 (0.27, 0.50)	0.75 (0.66, 0.85)
	C[2]	54	50	30	—	0.52 (0.45, 0.59)	0.15 (0.05, 0.23)	0.46 (0.34, 0.57)	0.85 (0.78, 0.93)
Automated	—	31	74	49	—	0.30 (0.21, 0.38)	0.00 (0.00, 0.00)	0.23 (0.09, 0.36)	0.58 (0.42, 0.72)
Positive predictive value									
Manual	M[1]	70	36	26	—	0.74 (0.67, 0.80)	0.41 (0.26, 0.56)	0.70 (0.59, 0.81)	0.95 (0.92, 1.00)
	M[2]	68	36	34	—	0.67 (0.61, 0.74)	0.33 (0.21, 0.44)	0.64 (0.53, 0.76)	0.99 (0.97, 1.00)
User-corrected	C[1]	53	52	42	—	0.56 (0.49, 0.63)	0.27 (0.16, 0.37)	0.49 (0.36, 0.62)	0.94 (0.89, 1.00)
	C[2]	54	50	30	—	0.65 (0.58, 0.72)	0.23 (0.09, 0.35)	0.56 (0.43, 0.69)	0.96 (0.92, 1.00)
Automated	—	31	74	49	—	0.39 (0.28, 0.49)	0.00 (0.00, 0.00)	0.33 (0.13, 0.52)	0.89 (0.77, 1.03)

Table A.5 Segmentation performance measures at minimum overlap > 0.25 Data in parentheses are 95% confidence intervals. A = automated segmentation, C^i = user-corrected segmentation by reader i, M^i = manual segmentations by reader i.

Method	Users	Dice similarity coefficient per detected lesion			Maximum symmetric surface distance (mm)			Average symmetric surface distance (mm)		
Ideal Value		1			0 mm			0 mm		
Lesion size		< 10 mm	10–20 mm	> 20 mm	< 10 mm	10–20 mm	> 20 mm	< 10 mm	10–20 mm	> 20 mm
Manual	M^1	0.65 (0.61, 0.69)	0.74 (0.72, 0.76)	0.81 (0.79, 0.83)	3.24 (2.90, 3.59)	4.49 (4.11, 4.85)	6.44 (5.71, 7.04)	0.64 (0.54, 0.75)	0.73 (0.65, 0.81)	0.89 (0.74, 1.01)
	M^2	0.67 (0.64, 0.71)	0.74 (0.72, 0.76)	0.82 (0.81, 0.84)	3.01 (2.58, 3.40)	4.77 (4.25, 5.23)	6.09 (5.65, 6.52)	0.59 (0.46, 0.70)	0.76 (0.64, 0.86)	0.80 (0.73, 0.87)
User-corrected	C^1	0.69 (0.65, 0.73)	0.66 (0.62, 0.69)	0.76 (0.74, 0.79)	2.83 (2.33, 3.86)	5.18 (4.73, 5.61)	7.09 (6.46, 7.68)	0.50 (0.35, 0.63)	1.02 (0.90, 1.14)	1.17 (1.02, 1.30)
	C^2	0.67 (0.63, 0.72)	0.68 (0.65, 0.71)	0.79 (0.77, 0.81)	3.62 (2.90, 4.23)	5.24 (4.80, 5.67)	6.89 (6.26, 7.45)	0.72 (0.51, 0.88)	0.95 (0.83, 1.07)	0.99 (0.88, 1.09)
Automated	A	—	0.66 (0.60, 0.71)	0.78 (0.74, 0.83)	—	5.18 (4.48, 5.88)	7.19 (5.86, 8.32)	—	1.05 (0.85, 1.25)	1.17 (0.87, 1.44)

Table A.6 Segmentation performance measures at minimum overlap > 0.5 Data in parentheses are 95% confidence intervals. A = automated segmentation, C^i = user-corrected segmentation by reader i, M^i = manual segmentations by reader i.

Parameter	Users	Dice similarity coefficient per detected lesion			Maximum symmetric surface distance (mm)			Average symmetric surface distance (mm)		
Ideal Value		1			0 mm			0 mm		
Lesion size		< 10 mm	10–20 mm	> 20 mm	< 10 mm	10–20 mm	> 20 mm	< 10 mm	10–20 mm	> 20 mm
Manual	M^1	0.76 (0.74, 0.78)	0.77 (0.76, 0.79)	0.83 (0.81, 0.84)	2.51 (2.20, 2.85)	4.22 (3.83, 4.60)	6.09 (5.62, 6.55)	0.39 (0.31, 0.48)	0.64 (0.57, 0.70)	0.79 (0.71, 0.86)
	M^2	0.76 (0.74, 0.78)	0.78 (0.77, 0.80)	0.82 (0.81, 0.84)	2.24 (2.03, 2.42)	4.27 (3.82, 4.70)	6.09 (5.65, 6.51)	0.37 (0.29, 0.44)	0.61 (0.51, 0.69)	0.80 (0.72, 0.86)
User-corrected	C^1	0.76 (0.74, 0.78)	0.76 (0.74, 0.78)	0.82 (0.80, 0.84)	2.34 (1.96, 2.70)	4.41 (3.80, 5.02)	6.72 (6.06, 7.33)	0.32 (0.24, 0.38)	0.65 (0.56, 0.74)	0.97 (0.87, 1.07)
	C^2	0.76 (0.74, 0.78)	0.75 (0.73, 0.77)	0.82 (0.80, 0.84)	3.25 (2.45, 3.92)	4.74 (4.21, 5.24)	6.76 (6.04, 7.37)	0.48 (0.36, 0.58)	0.68 (0.59, 0.76)	0.88 (0.79, 0.97)
Automated	A	—	0.76 (0.73, 0.80)	0.83 (0.81, 0.86)	—	4.87 (3.74, 6.00)	6.38 (5.36, 7.28)	—	0.71 (0.54, 0.86)	0.89 (0.72, 1.04)

Table A.7 Segmentation performance measures at minimum overlap > 0, > 0.25, and > 0.5 using Jaccard index. Data in parentheses are 95% confidence intervals. A = automated segmentation, C^i = user-corrected segmentation by reader i, M^i = manual segmentations by reader i.

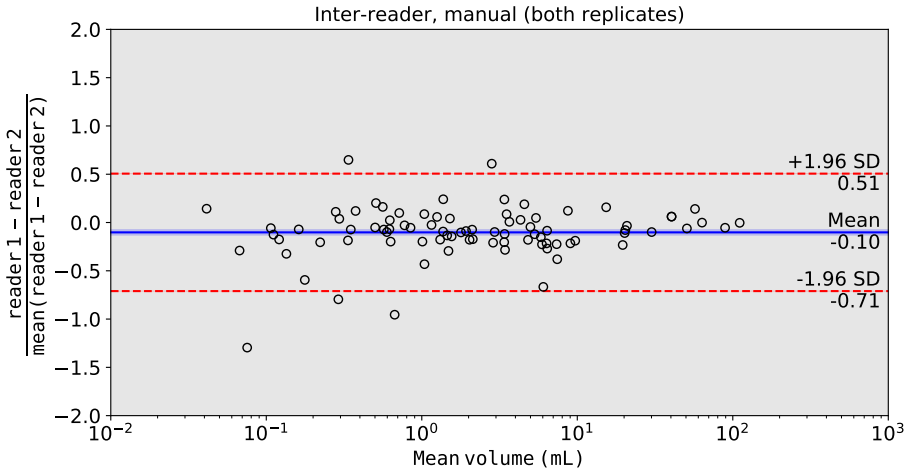
Parameter	Users	Jaccard index per detected lesion								
		Overlap > 0			Overlap > 0.25			Overlap > 0.5		
Ideal Value		1						1		
Lesion size	—	< 10 mm	10–20 mm	> 20 mm	< 10 mm	10–20 mm	> 20 mm	< 10 mm	10–20 mm	> 20 mm
Manual	M^1	0.47 (0.43, 0.53)	0.59 (0.56, 0.61)	0.68 (0.65, 0.71)	0.48 (0.44, 0.53)	0.59 (0.56, 0.62)	0.68 (0.66, 0.71)	0.61 (0.59, 0.64	0.63 (0.61, 0.65)	0.71 (0.68, 0.72)
	M^2	0.48 (0.43, 0.53)	0.59 (0.56, 0.61)	0.69 (0.68, 0.72)	0.51 (0.47, 0.55)	0.59 (0.56, 0.62)	0.70 (0.68, 0.72)	0.61 (0.59, 0.64)	0.64 (0.63, 0.67)	0.69 (0.68, 0.72)
User-corrected	C^1	0.47 (0.41, 0.54)	0.45 (0.41, 0.49)	0.61 (0.57, 0.65)	0.53 (0.48, 0.58)	0.49 (0.45, 0.52)	0.62 (0.59, 0.65)	0.61 (0.59, 0.64)	0.61 (0.59, 0.64	0.69 (0.67, 0.72)
	C^2	0.50 (0.46 0.56)	0.48 (0.44 0.52)	0.64 (0.61 0.68)	0.51 (0.46, 0.56)	0.52 (0.48, 0.55)	0.65 (0.62, 0.68)	0.61 (0.59, 0.64)	0.60 (0.57, 0.63	0.69 (0.67, 0.72)
Automated	A	0.08 (0.01, 0.16)	0.36 (0.28, 0.45)	0.52 (0.43, 0.63)	—	0.49 (0.43, 0.56)	0.64 (0.59, 0.71)	—	0.61 (0.57, 0.67	0.71 (0.68, 0.75)

Table A.8 Detection reliability at minimum overlap > 0.25
Data in parentheses are 95% confidence intervals. A = automated segmentation, C = user-corrected segmentation, C^i = user-corrected segmentation by reader i, $C_j = j^{th}$ user-corrected segmentation by both readers, M = manual segmentation, M^i = manual segmentations by reader i, $M_j = j^{th}$ manual segmentation by both readers.

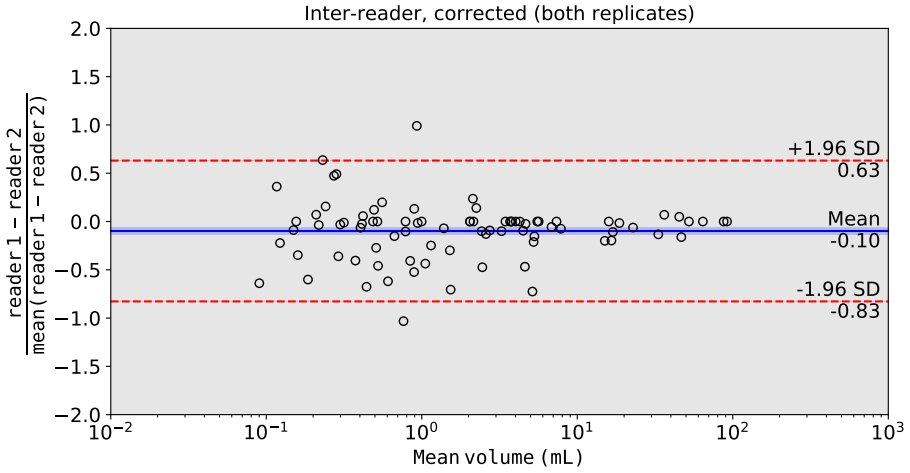
Parameter	Users	Pooled Cohen κ	Quantity Disagreement	Allocation Disagreement
Intrareader				
Manual	M_1, M_2	0.67 (0.53–0.83)	0.07 (0.02–0.11)	0.04 (0.00–0.08)
User-corrected	C_1, C_2	0.62 (0.48–0.78)	0.07 (0.01–0.11)	0.09 (0.02–0.14)
Interreader				
Manual	M^1, M^2	0.42 (0.25–0.64)	0.05 (0.01–0.07)	0.12 (0.07–0.19)
User-corrected	C^1, C^2	0.53 (0.39–0.71)	0.06 (0.00–0.09)	0.12 (0.08–0.18)
Intermethod				
Automated, manual	M, A	0.29 (0.18–0.39)	0.38 (0.30–0.46)	0.02 (0.00–0.05)
Automated, user-corrected	C, A	0.45 (0.34–0.56)	0.31 (0.24–0.39)	0.01 (0.00–0.02)
Manual, user-corrected	M, C	0.42 (0.28–0.59)	0.08 (0.03–0.11)	0.12 (0.08–0.16)

Table A.9 Detection reliability at minimum overlap > 0.5
Data in parentheses are 95% confidence intervals. A = automated segmentation, C = user-corrected segmentation, C^i = user-corrected segmentation by reader i, $C_j = j^{th}$ user-corrected segmentation by both readers, M = manual segmentation, M^i = manual segmentations by reader i, $M_j = j^{th}$ manual segmentation by both readers.

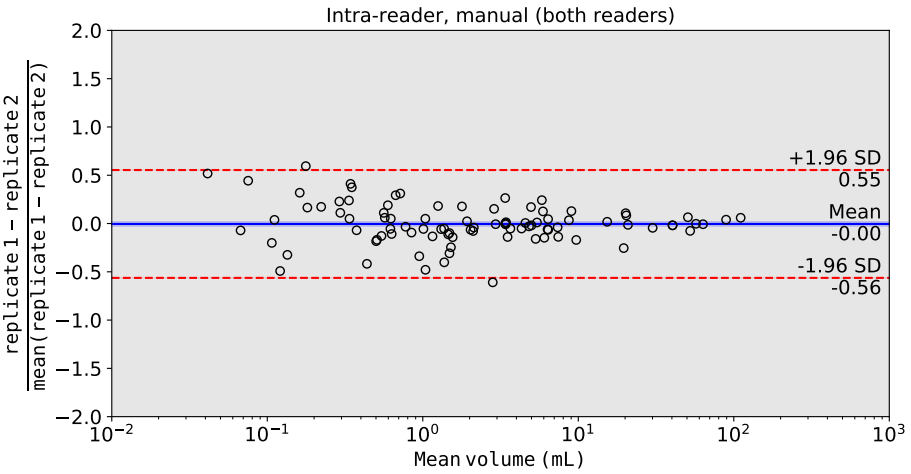
Parameter	Users	Pooled Cohen κ	Quantity Disagreement	Allocation Disagreement
Intrareader				
Manual	M_1, M_2	0.69 (0.58–0.82)	0.08 (0.01–0.13)	0.09 (0.02–0.14)
User-corrected	C_1, C_2	0.79 (0.71–0.90)	0.04 (0.00–0.08)	0.10 (0.05–0.15)
Interreader				
Manual	M^1, M^2	0.59 (0.47–0.74)	0.04 (0.00–0.06)	0.16 (0.11–0.23)
User-corrected	C^1, C^2	0.71 (0.62–0.83)	0.03 (0.00–0.04)	0.16 (0.11–0.23)
Intermethod				
Automated, manual	M, A	0.45 (0.34–0.56)	0.36 (0.28–0.44)	0.01 (0.00–0.03)
Automated, user-corrected	C, A	0.70 (0.61–0.80)	0.22 (0.15–0.28)	0.01 (0.00–0.01)
Manual, user-corrected	M, C	0.50 (0.38–0.65)	0.14 (0.08–0.21)	0.12 (0.07–0.17)


Inter-reader, manual (both replicates)
reader 1 − reader 2 / mean(reader 1 − reader 2)
+1.96 SD
0.51
Mean
-0.10
-1.96 SD
-0.71
Mean volume (mL)

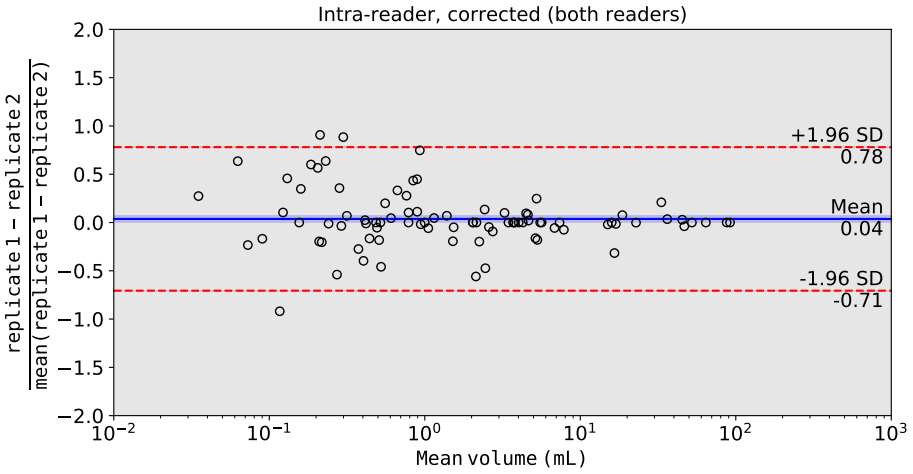
(a)


Inter-reader, corrected (both replicates)
reader 1 − reader 2 / mean(reader 1 − reader 2)
+1.96 SD
0.63
Mean
-0.10
-1.96 SD
-0.83
Mean volume (mL)

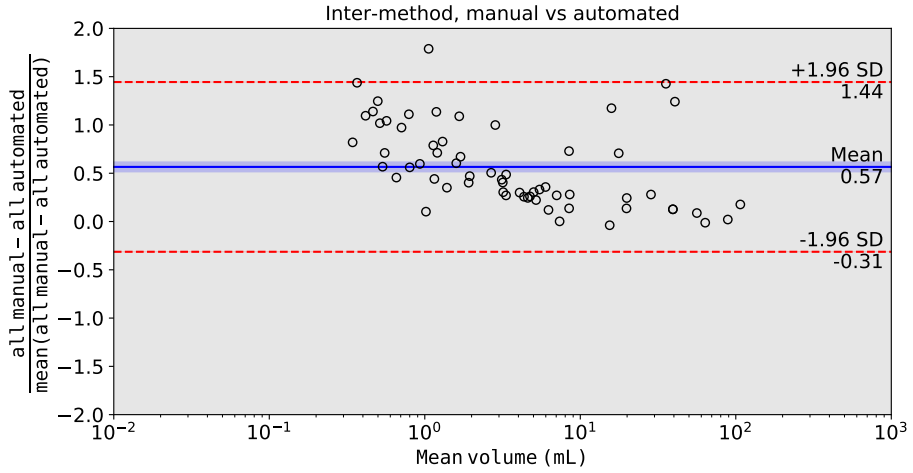
(b)


Intra-reader, manual (both readers)
replicate 1 − replicate 2 / mean(replicate 1 − replicate 2)
+1.96 SD
0.55
Mean
-0.00
-1.96 SD
-0.56
Mean volume (mL)

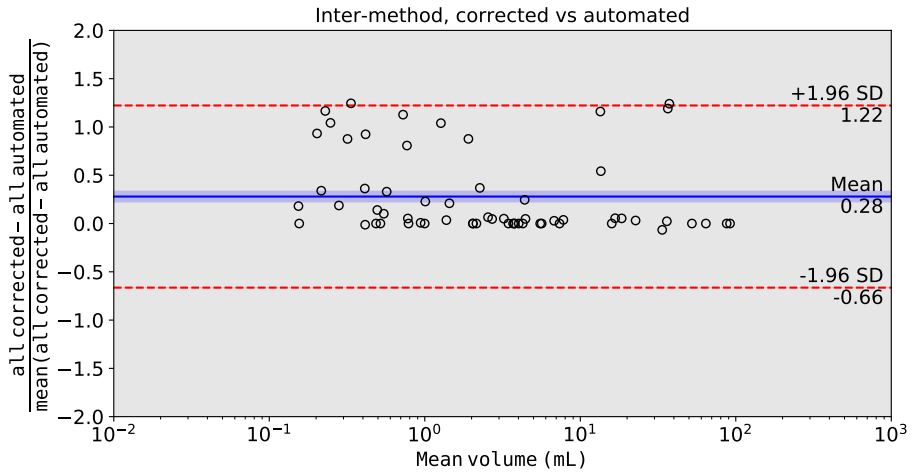
(c)


Intra-reader, corrected (both readers)
replicate 1 − replicate 2 / mean(replicate 1 − replicate 2)
+1.96 SD
0.78
Mean
0.04
-1.96 SD
-0.71
Mean volume (mL)

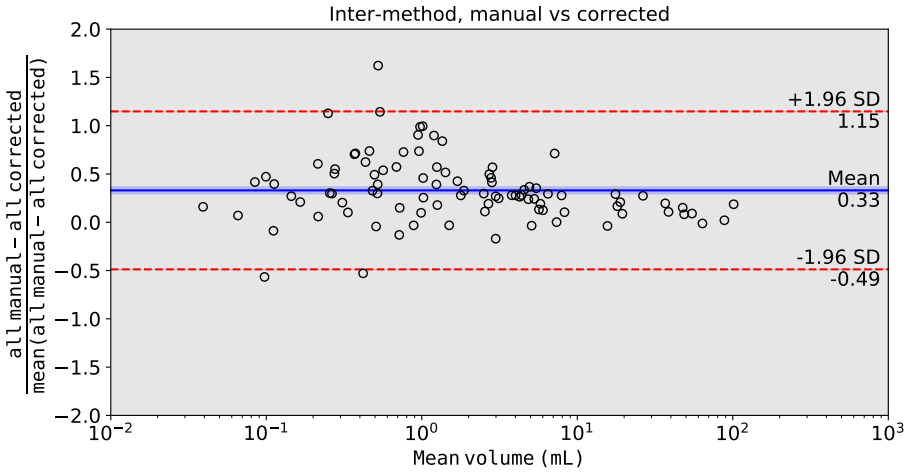
(d)


Inter-method, manual vs automated
all manual − all automated / mean(all manual − all automated)
+1.96 SD
1.44
Mean
0.57
-1.96 SD
-0.31
Mean volume (mL)

(e)


Inter-method, corrected vs automated
all corrected − all automated / mean(all corrected − all automated)
+1.96 SD
1.22
Mean
0.28
-1.96 SD
-0.66
Mean volume (mL)

(f)



(g)

Figure A.1 Bland-Altman plots of the volume difference (overlap > 0) between segmentation for **(a)** intra-reader manual, **(b)** intra-reader user-corrected, **(c)** inter-reader manual, **(d)** inter-reader user-corrected, **(e)** manual vs automated, **(f)** user-corrected vs automated, and **(g)** manual vs user-corrected method. M^i = manual segmentations by i^{th} analyst. C^i = corrections of automated segmentations by i^{th} analyst. $M^i = i^{th}$ manual segmentation by both analysts. $C^i = i^{th}$ correction of automated segmentation by both analysts.

APPENDIX B TOWARDS SEMI-SUPERVISED SEGMENTATION VIA IMAGE-TO-IMAGE TRANSLATION

B.1 Model and training details

This section details model architectures, parameter initializations, and optimization hyper-parameters. Network layers are described in Tables B.1-B.8. Here, *conv block* refers to a residual layer (as in the ResNet [28]) that chains together a normalization operation, an activation function, and a convolution, with a *short skip* connection from the input to the output as shown in in Figure B.1. For all experiments we use PyTorch [269].

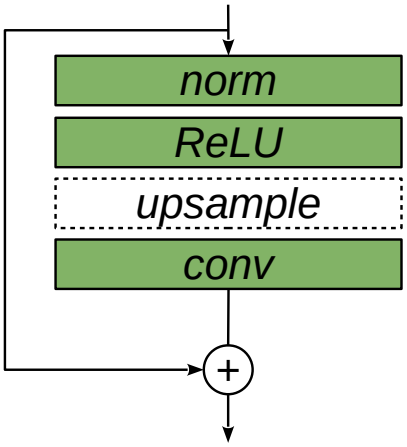


Figure B.1 The *conv block* chains a normalization operation (norm), a rectified linear unit (ReLU), and a convolution (conv). When used in a decoder, 2× upsampling is performed prior to convolution by simple repetition of pixel rows and columns. The input is summed to the output via a *short skip* connection.

General model structure The proposed model has one encoder and two decoders: *common* and *residual*. Additionally, it uses two discriminators, one for each direction of translation. The autoencoding baseline has one encoder and two decoders: segmentation and reconstruction. The segmentation baseline has one encoder and one segmentation decoder.

Reusing encoders and decoders. To compare the effect of different training objectives, we try to reduce the confounding effect of differing architectures between the proposed model and baseline models. For each task, we use the same encoder for all models; likewise, the *common* decoder in the proposed model and all decoders in the baseline models are the same. The *residual* decoder in the proposed model is similar, differing in that it lacks short

skip connections and uses slightly larger convolution kernels. All encoders and decoders are initialized with the Kaiming Normal approach [232]. Convolutions are applied to inputs with reflection padding. All activation functions are rectified linear units (ReLU).

Skip connections. We use *long skip* connections from the encoder to every decoder except the reconstruction decoder of the autoencoding baseline. Long skip connections bridge representations of the same resolution (these have the same number of channels). Specifically, the representation in the encoder is compressed to a single channel with a 1×1 convolution and then concatenated to the corresponding decoder representation. The encoder and all decoders have *short skip* connections (as in the ResNet), except for the *residual* decoders of the proposed model.

Latent code split. All latent bottleneck representations of every model have 512 channels. In the proposed model, 128 of these channels are specified as the *residual* latent code and the rest as the *common* latent code.

Normalization. All encoders use instance normalization [242]. All decoders use layer normalization [270]. The *residual* decoder of the proposed model performs segmentation by adopting a segmentation-specific optimization approach that differs from the layer normalization used with translation.

Segmentation via residual decoder. In the proposed method, the *residual* decoder is used both in translation and in segmentation. For segmentation, all but the last layer are used and a classification layer is appended: 1×1 convolution with N channels, where N is the number of classes. In order to adapt the features learned via translation to the segmentation task, inference is modified by using a different normalization approach during segmentation than during translation. For the MNIST tasks, a four layer multi-layer perceptron with 256 units per layer is used to map the latent code (both *common* and *unique*) to parameters for adaptive instance normalization [271] in the *residual* decoder. For BraTS segmentation, the *residual* decoder uses separate layer normalization parameters for segmentation.

Discriminator. Two discriminators are used with the proposed method, one for each direction of translation. We use multi-scale discriminators as proposed in [151, 272]. The discriminator architectures shown in Table B.7 and Table B.8 describe the network that is applied at each of three scales. At some scales, discriminators output a map of values per image instead of a single value. First, all pixels in this map are averaged and second, the resulting discriminator values are averaged across all scales. All discriminators use leaky ReLU [273] with a slope of 0.2.

Optimization. For all experiments, we used the AMSGrad optimizer [274] with $\beta_1 = 0.5$ and $\beta_2 = 0.999$. We used a learning rate of 0.0001 for all networks except discriminators which were trained with a learning rate of 0.001, following [275]. We used a batch size of 20 images. For MNIST experiments, we ran training for 300 epochs; for BraTS, 500 epochs. In the proposed method, we used the hinge loss for the adversarial objective, with spectral normalization [254] applied to all networks, as in [106, 251].

Weighted objectives. We found that the following objective weights yielded the best overall performance: $\lambda_{avd} = 3$, $\lambda_{rec} = 50$, $\lambda_{lat} = 1$, $\lambda_{cyc} = 50$, $\lambda_{seg} = 0.01$. (AE: $\lambda_{rec} = \lambda_{seg} = 1$).

Data augmentation We applied data augmentation on the fly during training for BraTS but not for MNIST tasks since a large amount of data is generated for the latter. Data augmentation involved random rotations up to 3 degrees, random zooms up to 10% in or out, random intensity shifts up to 10%, random horizontal and/or vertical flips, and spline warping. Spline warping used a 3×3 grid of control points with each point placed according to a Normal distribution with variance $\sigma = 5$. In those cases where data augmentation created new pixels along image edges or corners, these were filled by reflecting the image outward toward the edges and corners.

Table B.1 The encoder used for all models with MNIST 48×48.

Encoder (MNIST 48×48)

Layer	Channels	Kernel	Stride
Convolution	32	3×3	1
Conv block	64	3×3	2
Conv block	128	3×3	2
Conv block	256	3×3	2
Conv block	512	3×3	2
Norm+ReLU			

Table B.2 The decoder used for all models (*common* but not *residual* decoder in the proposed method) with MNIST 48×48.

Decoder (MNIST 48×48)

Layer	Channels	Kernel	Stride
Convolution	256	3×3	1
Conv block	128	3×3	1
Conv block	64	3×3	1
Conv block	32	3×3	1
Norm+ReLU+Conv	1	3×3	1

Table B.3 The *residual* decoder used in the proposed method with MNIST 48×48.

Residual decoder (MNIST 48×48)

Layer	Channels	Kernel	Stride
Convolution	256	5×5	1
Conv block (no short skip)	128	5×5	1
Conv block (no short skip)	64	5×5	1
Conv block (no short skip)	32	5×5	1
Norm+ReLU+Conv	1	5×5	1

Table B.4 The encoder used for all models with MNIST 128×128 and BraTS.

Encoder (MNIST 128×128 and BraTS)

Layer	Channels	Kernel	Stride
Convolution	16	3×3	1
Conv block	32	3×3	2
Conv block	64	3×3	2
Conv block	128	3×3	2
Conv block	256	3×3	2
Conv block	512	3×3	2
Norm+ReLU			

Table B.5 The decoder used for all models (*common* but not *residual* decoder in the proposed method) with MNIST 128×128 and BraTS.

Decoder (MNIST 128×128 and BraTS)

Layer	Channels	Kernel	Stride
Convolution	256	3×3	1
Conv block	128	3×3	1
Conv block	64	3×3	1
Conv block	32	3×3	1
Conv block	16	3×3	1
Norm+ReLU+Conv	1	3×3	1

Table B.6 The *residual* decoder used in the proposed method with MNIST 128×128 and BraTS.

Residual decoder (MNIST 128×128 and BraTS)

Layer	Channels	Kernel	Stride
Convolution	256	5×5	1
Conv block (no short skip)	128	5×5	1
Conv block (no short skip)	64	5×5	1
Conv block (no short skip)	32	5×5	1
Conv block (no short skip)	16	5×5	1
Norm+ReLU+Conv	1	5×5	1

Table B.7 The discriminator used in the proposed method with MNIST 48×48 and 128×128.

Discriminator (MNIST)

Layer	Channels	Kernel	Stride
Convolution	128	4×4	1
Norm+ReLU+Conv	128	4×4	2
Norm+ReLU+Conv	256	4×4	2
Norm+ReLU+Conv	512	4×4	2
Convolution	1	1×1	1

Table B.8 The discriminator used in the proposed method with BraTS.

Discriminator (BraTS)

Layer	Channels	Kernel	Stride
Convolution	64	4×4	1
Norm+ReLU+Conv	64	4×4	2
Norm+ReLU+Conv	128	4×4	2
Norm+ReLU+Conv	256	4×4	2
Norm+ReLU+Conv	512	4×4	2
Convolution	1	1×1	1