# Advanced Deep Learning for Medical Image Segmentation

## Towards global and data-efficient learning

Shuai Chen

# Advanced Deep Learning for Medical Image Segmentation

## Towards global and data-efficient learning

**Shuai Chen**

**Acknowledgements**:

The work presented in this thesis was conducted at the Department of Radiology & Nuclear Medicine of the Erasmus MC, Rotterdam, the Netherlands.

The following organizations are gratefully acknowledged: the Chinese Scholarship Council (File No.201706170040) for funding the research and the Department of Radiology & Nuclear Medicine (Erasmus MC) for supporting the publication of this thesis.

**ISBN**: 978-94-6469-106-1
**Cover**: Shuai Chen
**Layout**: Shuai Chen
**Printing**: ProefschriftMaken | www.proefschriftmaken.nl

# Advanced Deep Learning for Medical Image Segmentation

## Towards global and data-efficient learning

Geavanceerd machinaal leren voor medische beeldsegmentatie

Op weg naar globaal en data-efficiënt leren

**THESIS**

to obtain the degree of Doctor from the
Erasmus University Rotterdam
by command of the
rector magnificus

Prof. dr. A.L. Bredenoord

and in accordance with the decision of the Doctorate Board.

The public defence shall be held on
Wednesday 21 December 2022 at 10.30 hrs

by

**Shuai Chen**
born in Beijing, China

**Erasmus University Rotterdam**

## Doctoral Committee

| | |
|---|---|
| **Promotor** | prof. dr. M. de Bruijne |
| **Other members** | prof. dr. B. van Ginneken<br>prof. dr. M. Vernooij<br>prof. dr. K. Zhou |
| **Copromotor** | dr. ir. G. van Tulder |

**To my parents**

# Contents

# Chapter 1

## Introduction

Image segmentation is one of the most important tasks in medical image analysis. It aims to outline structures such as organs or lesions given images from one or multiple imaging modalities [1], such as Computed Tomography (CT), MRI, Ultrasound, etc. The segmentation results may provide valuable clinical biomarkers that can help clinicians to make accurate diagnoses and treatment decisions [2, 3, 4]. For example, segmentations of the aorta and pulmonary artery provide measurements such as the diameter ratio of these two vessels, which is an important risk factor for exacerbations of COPD [2]. Similarly, airway and vessel segmentation from chest CT scans provides measurements such as lumen diameter, airway-artery ratio, and wall thickness, which are relevant to quantitatively assessing lung diseases [3].

In recent years, deep learning-based methods gained tremendous attention in the computer vision and medical imaging fields. In medical image segmentation tasks, U-Net-like architectures provide state-of-the-art results in many applications [5, 6] due to their good performance and ease of implementation. However, deep learning networks such as U-Net still suffer from many problems. For example, the fully-convolutional nature of U-Net makes it inefficient to model global information within images and labels, such as image-level constraints based on the anatomy of organs or the topology of objects in ground-truth segmentations. Also, training a good U-Net model usually requires a relatively large amount of training data, but acquiring manual segmentations is time-consuming and, as a result, labeled data is scarce.

This thesis proposes solutions to these problems, by developing global and data-efficient deep learning methods. This introduction chapter provides background on deep learning segmentation in medical imaging (Section 1.1), followed by a discussion of the two main challenges addressed in this thesis: the ability to learn global information (Section 1.2) and the scarcity of label data (Section 1.3). Then, we show a summary of the contributions and the outline of this thesis (Section 1.4).

## 1.1 Deep learning in medical image segmentation

Deep learning is a branch of machine learning algorithms that usually employs a convolutional neural network (CNN). The key to the popularity of deep learning is that it learns useful features automatically from data through backpropagation, which makes a deep learning network a very powerful feature extractor that can theoretically fit any kind of function given enough model capacity.

However, training a deep neural network is challenging due to its high computational burden and difficult optimization. Over the last decade, many techniques have been proposed to overcome these difficulties [7, 8, 9, 10], making deep learning feasible in practice. Moreover, the open-source codes greatly accelerate the technical transfer of deep learning from computer vision to medical image analysis.

The publication of U-Net was an important milestone of deep learning in medical image segmentation [11]. U-Net efficiently concatenates the low-level features in the encoder with high-level semantic features in the decoder. Later, many variations based on U-Net have been proposed to further enhance its modeling ability and robustness in different applications [5]. Recently, it was shown that a simple, deeply supervised 2D or 3D U-Net (or an ensemble of them) without any additional variations is able to achieve top performance in many segmentation tasks [6].

However, although the CNN methods are accurate in many segmentation tasks, there are still problems in challenging segmentation tasks. For example, segmenting complex structures like airways [3] or brain vessels [4], or lesions like brain tumors [12]. A typical segmentation network such as U-Net makes predictions on a local and voxel-wise level without image-level constraints such as spatial consistency. This may hinder the model performance when the segmentation requires global information such as shape priors.

Another problem of the CNN methods in medical image segmentation is that the available labeled data for model training can be very scarce. The annotations often require knowledge from experienced radiologists. This is different from the situation in natural images in the computer vision field where the labels can be collected through crowdsourcing from the Internet. In medical imaging, often a large amount of data without annotations (i.e, unlabeled data) is more easily available, which may contain useful information to train a better segmentation network. Thus, methods that can learn from these unlabeled data such as semi-supervised and self-supervised learning may improve performance in many segmentation tasks.

## 1.2 Global learning methods

Enhancing the ability of deep learning models to learn global (as opposed to local) image information can be beneficial in segmentation tasks. In previous works, several methods were investigated to solve this problem. For example, graph-based methods such as Conditional Random Fields (CRF) refine the segmentation results by encouraging label smoothness on a global scale (or, to reduce computational costs, between neighboring voxels) and can be used as a post-processing method [13], or trained as an additional layer on top of the neural network in an end-to-end manner [14]. This makes it possible to apply useful regularizations such as within-class intensity similarity or spatial smoothness to predictions and improve segmentation performance.

Another popular graph-based method is graph neural networks (GNNs), which enhance the global modeling ability of the network by replacing the conventional convolution operation with graph-based convolution [15]. However, GNN has not been widely used in medical image segmentation yet due to the high computation cost of graph-based convolution, where usually only a few connections can be implemented within each graph filter. This limits the potential of GNN in many segmentation tasks.

Recently, the self-attention mechanism (Transformer) was introduced in computer vision, enabling neural networks to learn global information through several stacks of transformer layers [16]. These networks show high efficiency in computing and transferring global information between layers. The main limitation is that Transformers require a large amount of data to train [17], while labeled data is usually very scarce in medical imaging applications.

Most of the CNN-based methods mentioned above aim to learn global information from the input images. However, these networks may not sufficiently capture the information within segmentation labels. In some of the experiments in this thesis, we noticed that without explicitly learning global label information, our networks would sometimes make segmentation errors that are obvious from looking at the output labels alone. For example, in tree-structure segmentation tasks, e.g. airway segmentation and

vessel segmentation, the resulting segmentations may contain discontinuous branches. One solution is to force the continuity of tree-like branches in the predictions, or use an adaptive loss function such as centerline-in-volume-dice-coefficient loss to preserve connectivity [18, 19]. However, these methods focus on tree-like structures and may be difficult to apply to other applications.

In this thesis, we propose our solutions to these problems. Specifically, to better learn global information from the input images and the intermediate feature maps, we present an end-to-end training method called Posterior-CRF based on the combination of CNN (such as U-Net) and a learning-based CRF layer. Previous CNN-CRF methods mostly use intensity as the CRF feature [14] and encourage the model to assign voxels with similar intensities to the same class. However, the intensity-based information provides a limited feature space for the CRF inference and may not be very useful for segmentation tasks that require higher-level semantic features. For example, in aorta and pulmonary artery segmentation, the aorta and pulmonary artery are two different vessels in anatomy but they share very similar intensity values in non-contrast CT images. This may confuse the intensity-based CRF and lead to wrong predictions. Differently, our Posterior-CRF allows the CRF to use the high-level semantic features learned by a CNN, instead of the fixed intensity features from the input images. This may improve the segmentation performance of the CRF.

Although Posterior-CRF is able to perform global, image-level inference based on the inputs and CNN features, the global information in labels is not encoded in the model explicitly and thus the model may make structural errors in the predictions. To fix these errors, a novel label refinement method is presented in this thesis. The main idea of this method is to generate synthetic errors that are similar to the ones that are present in the initial segmentation results and to subsequently train a label refinement network to correct them. We evaluate this approach in tree-shaped structures: lung airways and brain vessels. By learning to correct synthetic errors in segmentation continuity, the network is expected to learn how to refine the segmentation that contains real errors.

## 1.3 Data-efficient learning methods

The second problem in medical imaging is that manual annotations are very expensive and time-consuming to make. In practice, the available labeled data in many segmentation tasks may not be sufficient to train a deep learning model with high accuracy. Therefore, the methods that can learn useful features from unlabeled data have attracted much attention in recent years. One popular research direction is semi-supervised learning, where unlabeled data is combined with labeled data to train models. For example, some methods train autoencoders to extract features from reconstructing the unlabeled input data, which extracts features that may contain useful information for a segmentation task [20]; or to generate pseudo labels for unlabeled data as additional training data, e.g., self-training [21] or mean-teacher [22].

In the deep learning era, a new research direction called self-supervised learning has been proposed and is still under fast iteration nowadays [23]. The main idea of self-supervised learning is using a deep learning model to learn useful features from a manually designed task with unlabeled data. The task is usually designed by

removing a specific kind of information from the original images, e.g., by masking or shuffling imaging parts, and forcing the network to predict the missing information. The learned features can be used as a better starting point to optimize the downstream segmentation task, compared to random initialization. Most self-supervised learning models can be trained without annotations, as the learned features are general and can usually be reused across different segmentation tasks.

Considering the scarcity of labeled data in medical image segmentation tasks, this thesis explores three ways to learn from unlabeled data. First, a new semi-supervised method is presented based on multi-task and attention-based learning. This method combines reconstruction and segmentation in an autoencoder network. Unlike the traditional reconstruction task that aims to reconstruct the whole image, the proposed task reconstructs the foreground and background texture separately. This encourages the autoencoder to focus more on reconstructing the boundaries between foreground and background, which are highly related to the main segmentation task. The learned features are shared with a U-Net with the same encoder in a multi-task learning manner and may improve the segmentation performance.

We also propose two novel self-supervised learning methods. The first one is a region-of-interest-guided supervoxel inpainting method. In this method, the regions to recover in images are masked using supervoxel-based irregular tiles in the area to be segmented, instead of voxel-based square tiles in a random area. Compared to the original inpainting task [24], the proposed task is able to extract more useful features by recovering the masked foreground region, which may help improve the downstream segmentation task. The second method is a new self-supervised task called Source Identification (SI), inspired by the classic blind source separation (BSS) problem [25]. In the SI task, the network is trained to identify and separate the source image from an image that mixes the source image with images from other sources. To successfully distinguish the source image from the others, the network needs to learn not only the local features but also global features such as semantics and anatomical information. To the best of our knowledge, this is the first time that a BSS-like task is applied as a self-supervised task in deep learning. The proposed task may increase the diversity of features that can be learned by traditional self-supervised learning methods.

Other than self-supervised learning methods, the label refinement method presented in Chapter 6 can also work in a semi-supervised setting using unlabeled data. The way to use unlabeled data is to add the synthetic errors to the intermediate predictions. Then, the new pseudo labels with synthetic errors and unlabeled data can be used as additional training data to train a stronger label refinement network. With the help of unlabeled data, the label refinement network would be able to see more synthetic errors and learn to fix them. Compared to traditional semi-supervised methods that use pseudo labels, this method is able to increase the diversity of the training sets with the awareness of the label structural information.

## 1.4 Outline of this thesis

This thesis develops and validates advanced deep learning segmentation methods. The contributions of this thesis can be divided into global learning methods and data-efficient learning methods:

**Global learning**

- *Chapter 2* develops a new end-to-end trainable algorithm called Posterior-CRF for medical image segmentation. It uses CNN-learned features in a CRF and simultaneously optimizes the CNN and CRF parameters.
- *Chapter 6* develops a new label refinement method that can correct label structural errors in initial segmentation results.

**Data-efficient learning**

- *Chapter 3* develops a new semi-supervised learning method that combines reconstruction and segmentation in an encoder-decoder network. The combination of these two tasks forces the autoencoder to reconstruct the foreground and background separately, which may help improve the segmentation performance.
- *Chapter 4* develops a new self-supervised inpainting task with a region-of-interested guided supervoxel technique. Instead of using random masking with regular square tiles in images in the original inpainting task, this method masks the segmentation foreground with supervoxel tiles to guide the inpainting task for self-supervision. This may provide more segmentation-relevant features compared to the original inpainting task and improve the downstream segmentation performance.
- *Chapter 5* develops a new self-supervised task inspired by the classic blind-source separation (BSS) problem. The task is to identify and separate the source images from sets of synthetic images that mix the source image with multiple images from other sources. Useful information such as anatomy between patients can be extracted and can be used as pretrained features to improve downstream segmentation performance.
- *Chapter 6* evaluates the label refinement method in a semi-supervised setting. Synthetic errors can be added to the new pseudo labels to train a stronger label refinement network.

# Chapter 2

## An End-to-end Approach to Segmentation in Medical Images with CNN and Posterior-CRF

2


## Abstract

*Conditional Random Fields (CRFs) are often used to improve the output of an initial segmentation model, such as a convolutional neural network (CNN). Conventional CRF approaches in medical imaging use manually defined features, such as intensity to improve appearance similarity or location to improve spatial coherence. These features work well for some tasks, but can fail for others. For example, in medical image segmentation applications where different anatomical structures can have similar intensity values, an intensity-based CRF may produce incorrect results. As an alternative, we propose Posterior-CRF, an end-to-end segmentation method that uses CNN-learned features in a CRF and optimizes the CRF and CNN parameters concurrently. We validate our method on three medical image segmentation tasks: aorta and pulmonary artery segmentation in non-contrast CT, white matter hyperintensities segmentation in multi-modal MRI, and ischemic stroke lesion segmentation in multi-modal MRI. We compare this with the state-of-the-art CNN-CRF methods. In all applications, our proposed method outperforms the existing methods in terms of Dice coefficient, average volume difference, and lesion-wise F1 score.*


Based on: **S. Chen**, Z. Sedghi Gamechi, F. Dubost, G. van Tulder, and M. de Bruijne, "An end-to-end approach to segmentation in medical images with CNN and Posterior-CRF," *Medical Image Analysis*, vol. 76, p. 102 311, 2022. DOI: 10.1016/j.media.2021.102311

## 2.1 Introduction

After the breakthrough of deep learning in computer vision [26, 27, 28], deep convolutional neural networks (CNNs) and their variants [29, 30, 31] quickly started to dominate medical image segmentation, outperforming traditional machine learning methods in many applications [32, 33, 34, 35]. To refine the prediction from the CNN, it is common to combine CNN with a conditional random field (CRF) [36]. By modeling pairwise relationships and interactions between voxel-wise variables over the whole image, the CRF can improve the coherence of the segmentation. In previous work, CRFs based on predefined features such as intensity similarity and spatial coherence have been used as an efficient post-processing technique or trained in an end-to-end manner in a recurrent neural network to refine the CNN outputs [31, 37, 38, 39].

Most often, a CRF uses a combination of voxel intensity and voxel location as pairwise potentials. Although this works well in several computer vision applications [39, 40], there can be challenges in other applications. The approach assumes that voxels that have similar intensity and are close to each other in the image are likely to belong to the same class. There are many applications among others in medical image analysis in which this assumption does not hold. For example, the intensity-based features of the CRF are not sufficient for problems where the intensity is not informative enough to identify object boundaries, such as the artery segmentation problem in Figure 2.1a. The spatial component of the CRF, on the other hand, requires extra careful tuning when the CRF is applied to data with isolated small objects, such as the white matter hyperintensities in Figure 2.1b, which may be erroneously removed by excessive smoothing. In stroke lesion segmentation, a large appearance difference between lesion objects of the same class also goes against the CRF assumption that the same class objects should have similar intensity (see Figure 2.1c).

In this chapter, we propose *Posterior-CRF*, a new learning-based CRF approach for image segmentation that allows the CRF to use features learned by a CNN, optimizing the CRF and CNN parameters concurrently. The learning-based CRF makes the CNN features update to work best with CRF in an end-to-end manner. During training, the CRF inference works in the CNN feature space, which is more likely to contain useful high-level features for segmentation compared to the original intensity values.

We demonstrate our method in three medical image analysis applications. Our first application is the segmentation of the aorta and pulmonary artery in non-contrast, non-ECG-gated chest CT scans. In these images, the aorta and the pulmonary artery share similar intensity values, which goes against the CRF assumption that similar classes should share similar intensity [41, 42]. The boundaries between the objects are not recognizable by intensity alone, making a standard CRF less effective (Figure 2.1a). Our second application is the segmentation of white-matter hyperintensities in brain MRI. These small objects are sparsely distributed in the brain (see Figure 2.1b) and may be removed by the CRF, which optimizes for the spatial coherence of segmentation. Our third application is the segmentation of ischemic stroke lesions in brain MRI, which have very heterogeneous intensities and shapes within the same lesion class (Figure 2.1c).

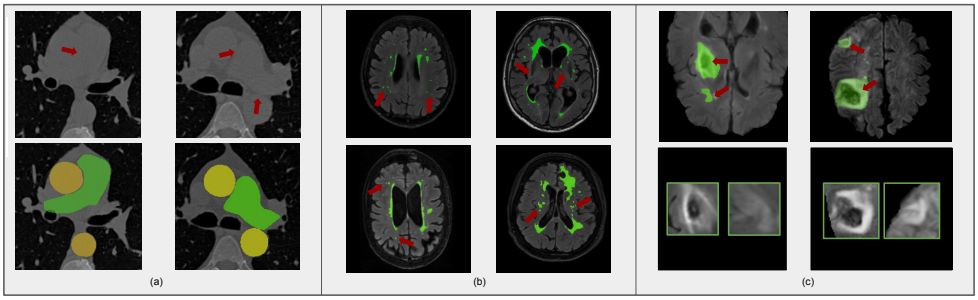



***Figure 2.1:*** ***Difficult cases for conventional CRF inference in medical image segmentation.*** *(a) Segmentation of arteries in CT: first row shows two axial slices of the CT scan with red arrows indicating indistinguishable boundaries; second row shows the corresponding ground truth of the aorta (yellow) and pulmonary artery (green); (b) White matter hyperintensities segmentation in MRI: four examples are shown with the ground truth of the lesions (green), red arrows indicate small isolated lesions that can be easily removed by CRF; (c) Ischemic stroke lesions segmentation in MRI: first row shows the ground truth of the lesions (green) where large appearance difference between lesions can be observed (red arrows); second row shows a close-up view of the lesions. Best viewed in color with zoom.*

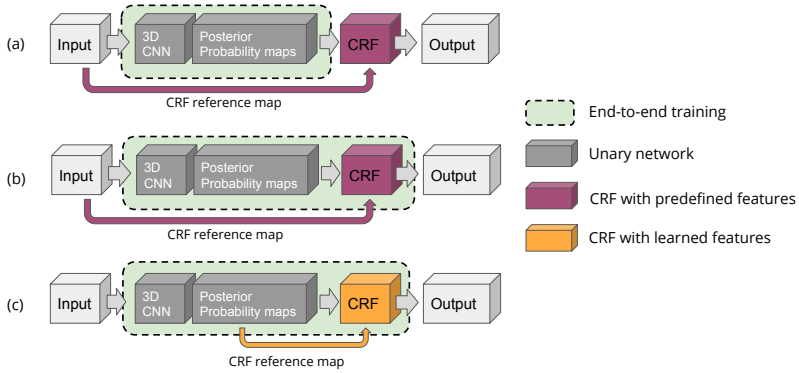



***Figure 2.2:*** ***Different CRF-based approaches*** *For each graph: (a) Post-processing CRF [31, 37]; (b) End-to-end training CRF with predefined features [39]; (c) Proposed Posterior-CRF, which uses CNN feature maps as CRF reference maps. Best viewed in color with zoom.*

**Contributions**

1. We present a new end-to-end trainable algorithm for image segmentation called *Posterior-CRF* using learnable features in CRF pairwise potentials. We explore how the proposed method affects CNN learning during training.

2. We compare the performance of a fully-connected CRF in several settings: post-processing, end-to-end training with predefined features, and end-to-end training with learned features. Ablation experiments are conducted to investigate the influence of CRF parameters and which level of the CNN feature maps are more likely to benefit the CRF inference. We found that the features in the last CNN feature maps provide a more consistent improvement than features in early CNN layers and predefined intensity features.

3. We evaluate our methods in three applications: aorta and pulmonary artery segmentation in non-contrast CT, which can be used to compute important biomarkers such as the pulmonary artery to aorta diameter ratio [41]; white matter hyperintensities segmentation in multi-sequence MRI, which is of key importance in many neurological research studies [34]; and ischemic stroke lesion segmentation in multi-sequence MRI, which can provide biomarkers for stroke diagnosis [35]. In the experiments, the proposed Posterior-CRF outperforms CNN without CRF, post-processing CRF, end-to-end intensity-based CRF, and end-to-end spatial-based CRF.



## 2.2 Related Work

### 2.2.1 End-to-end Training of CRF and CNN

CRF is widely used as an efficient post-processing method to refine the output of CNN segmentation models (for example, [31, 37, 38]). However, applying a CRF as post-processing means that the CNN is not able to adapt its output to the CRF. Zheng et al. [39] proposed to optimize CNN and CRF jointly by reformulating the CRF inference as a recurrent neural network (RNN) operation, such that the CRF weights can be learned together with the CNN. This approach makes the unary potentials and the kernel weights in pairwise potentials trainable, which saves the computational cost of grid search for other approaches to tune these weights, although the CRF still works in the predefined fixed feature space. In this paper, we focus on a new CRF approach where the CRF inference works in a learning-based CNN feature space.

### 2.2.2 Locally-connected CRFs with Learned Potentials

While conventional CRFs use predefined Gaussian edge potentials, the potentials can also be learned through a neural network. Vemulapalli et al. [43] learn the pairwise potentials of a Gaussian CRF in a bipartite graph structure. This approach uses a simpler continuous CRF model which provides better convergence of mean-field inference than the conventional discrete CRF models. In this paper, we focus on the most widely used discrete CRF model which is a natural fit for the dense

segmentation problem. Lin et al. [44], Li et al. [45] and Wang et al. [46] learn pairwise CRF potentials to model patch-wise (or local) relationships using free form functions learned by neural network rather than a combination of predefined Gaussians to calculate the pairwise potentials. The patch-wise potentials provide a better ability to model the semantic compatibility between image regions and have different effects compared to our approach, where we do not consider patch-wise relationships. Our method uses traditional Gaussian edge potentials [36] similar to Zheng et al. [39] which are easier to compute in a fully-connected manner. Unlike Zheng et al., we derive the potentials from the feature space learned by a CNN. This allows us to model global interactions between voxel-wise variables using learning-based features.

### 2.2.3 Other Methods Related to CRF

Next to CRF, there are several other approaches that aim to model interactive relationships or add global information to neural networks. Graph neural networks (GNN) [47, 48] model interactions between variables by applying graph convolution filters, which allow them to learn global relationships between voxels. We further address GNN in the Discussion. The recently proposed non-local CNN [49] uses layer-wise self-attention [16, 50, 51] to make each layer in the network focus on the areas that encoded the most non-local information in the preceding layer. While this allows non-local CNNs to model long-range dependencies, they are unable to model the interactions that can be learned by a CRF or GNN. In this paper, we focus on the fully-connected CRF model which is an efficient approach of modeling both interactive relationships and global information.

## 2.3 Methodology

Our method consists of two parts that are optimized jointly: 3D CNN and 3D CRF. In Section 2.3.1, we describe the CNN model, which provides unary potentials for the CRF inference as well as features for the pairwise potentials for the proposed Posterior-CRF. Then we introduce the CRF in Section 2.3.2. We show two previously proposed ways to perform CRF inference using predefined features: post-processing (Section 2.3.3) and end-to-end training with predefined features (Section 2.3.3). Our proposed end-to-end training with learned features is presented in Section 2.3.4, followed by Section 2.3.4 about the back-propagation of the proposed learning-based CRF. The mean-field inference algorithm used in the proposed method is explained in Appendix Section 2.9.1.

### 2.3.1 CNN Model

Our CNN model is based on UNet [29], the most widely used network architecture for medical image segmentation. It has a multi-scale design with skip-connections that connect the encoding and decoding parts of the network, which allow the decoding path to use the early, high resolution feature maps without losing information through pooling. We use 3D UNet as the basic CNN architecture to provide the unary potentials for CRF inference as well as features for the pairwise potentials for the proposed

Posterior-CRF. Details of the network layout used in our experiments are given in Figure 2.3.

### 2.3.2 Conditional Random Fields



In this section, we describe the CRF as proposed in [36]. In image segmentation, a CRF models voxel-wise variable $x_i$ taking values in $\{1, ..., C\}$ as a set of random variables $\mathcal{X} = \{x_1, ..., x_N\}$, where $C$ is the number of classes and $N$ is the number of voxels in the image. During training, $x_i$ is converted into a soft classification vector of length $C$, indicating for each class the probability that the $i$th voxel belongs to that class, with the $L_1$ norm $|x| = 1$. $x_i$ obey a Markov property conditioned on a global observation, the image $\mathbf{I}$ consisting of variables $\mathcal{I} = \{I_1, ..., I_N\}$. In this paper, $\mathbf{I}$ is the observed 3D CT/MRI scans, with its length given by the number of imaging modality channels $M$ times the number of voxels per channel $N$.

Consider a fully-connected pairwise CRF model ($\mathbf{X}$,$\mathbf{I}$) characterized by a prior Gibbs distribution:

$$P(\mathbf{X}|\mathbf{I}) = \frac{1}{Z(\mathbf{I})}\exp(-\sum_{c\in\mathcal{C}_\zeta}\phi_c(\mathbf{X}_c|\mathbf{I})) \tag{2.1}$$

where $\zeta = (\mathcal{V}, \mathcal{E})$ is an undirected graph describing the random field $\mathbf{X}$. Each clique $c$ in a complete set of unary and pairwise cliques $\mathcal{C}_\zeta$ in $\zeta$, and $\phi$ is the potential for each clique. We seek a maximum a posteriori probability (MAP) estimation $\mathbf{x}$ that minimizes the corresponding Gibbs energy $E(\mathbf{X} = \mathbf{x}|\mathbf{I})$:

$$E(\mathbf{X} = \mathbf{x}|\mathbf{I}) = \sum_i \varphi_u(x_i|\mathbf{I}) + \sum_{i<j}\varphi_p(x_i, x_j|\mathbf{I}) \tag{2.2}$$

$$MAP(P(\mathbf{X}|\mathbf{I})) : \mathbf{x}^* = \underset{\mathbf{x}}{\operatorname{argmin}} E(\mathbf{X} = \mathbf{x}|\mathbf{I}) \tag{2.3}$$

where $i$ and $j$ range from 1 to $N$. The first term $\varphi_u(x_i)$ in Equation 2.2 is the unary potential, which in our case is the current $C$ length vector of voxel $i$ representing the class probabilities in the CNN posterior probability maps. The second term $\varphi_p(x_i, x_j)$ is the pairwise potential:

$$\varphi_p(x_i, x_j) = \mu(x_i, x_j)\sum_{m=1}^{K}\omega_m k_m \tag{2.4}$$

where $\mu(x_i, x_j)$ is the label compatibility function that describes the interactive influences between different pairs of classes, $\omega_m$ is the linear combination weight of different pre-defined kernels $k_m$ and $K$ is the total number of kernels. Each $k_m$ is a modified Gaussian kernel with specific feature vector $\mathbf{f}$:

$$k(\mathbf{f}_i, \mathbf{f}_j) = \prod_{s=1}^{S}\mathbf{exp}(-\frac{1}{2}(f_i^s - f_j^s)^{\mathrm{T}}\mathbf{\Lambda}^s(f_i^s - f_j^s)) \tag{2.5}$$

The feature vector $\mathbf{f}$ is defined from $S$ arbitrary feature spaces. $\mathbf{\Lambda}$ is a symmetric positive-definite precision matrix that defines the shape of each kernel. In semantic

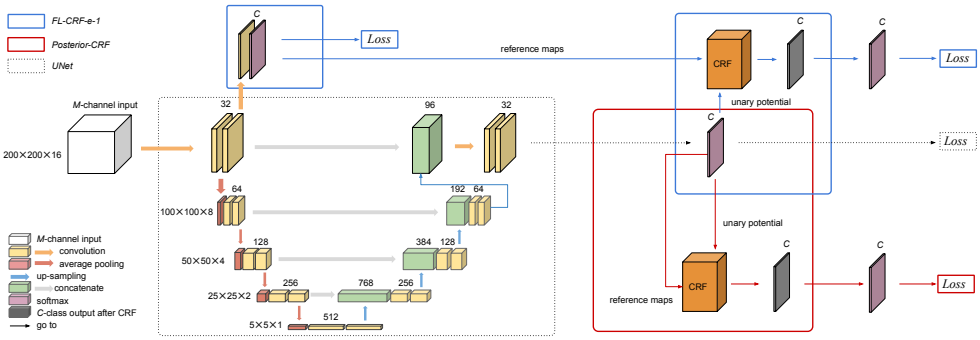



***Figure 2.3: Proposed feature-learning-based CRF using early/later CNN feature maps.*** *The backbone architecture is based on 3D UNet. The skip-connections concatenate the feature maps from the encoder path with the upsampled ones from the decoder path. The CRF module is placed on top of the CNN and infers the most likely posterior class probability conditioned on the CRF features. M is the number of input imaging modalities. C is the number of output classes. Two proposed CRF variants are shown in this figure: 1. Posterior-CRF (red rectangle and arrows), which uses the last CNN layer as CRF reference maps; 2. FL-CRF-e-1 (blue rectangles and arrows), which uses the first level CNN layer as CRF reference maps. Best viewed in color with zoom.*

segmentation, typically a combination of intensity ($I$) and position features ($p$) has been used [31, 36, 39]:

$$\begin{aligned}\varphi_p(x_i, x_j) =& \mu(x_i, x_j)[\omega_1 \mathbf{exp}(-\frac{|p_i - p_j|^2}{2\theta_\alpha^2} - \frac{|I_i - I_j|^2}{2\theta_\beta^2}) \\ &+ \omega_2 \mathbf{exp}(-\frac{|p_i - p_j|^2}{2\theta_\gamma^2})]\end{aligned} \tag{2.6}$$

where the first kernel controlled by $\omega_1$ is called *appearance kernel* and the second kernel controlled by $\omega_2$ is called *smoothness kernel*. The parameters $\theta_\alpha$, $\theta_\beta$ and $\theta_\gamma$ control the influence of the corresponding feature spaces. The appearance kernel is inspired by the observation that nearby voxels with similar intensity are likely to be in the same class, while voxels that are either further away or have larger intensity difference are less likely to be in the same class. The smoothness kernel can remove isolated regions and produce smooth segmentation results [31, 36]. Note that the position feature appears in both appearance kernel and smoothness kernel, where spatial information has different contributions to each of the two kernels, depending on the spatial standard deviations $\theta_\alpha$ and $\theta_\gamma$.

### 2.3.3 CRF with Predefined Features

Conventional CRFs use predefined features, such as the image intensity and spatial position shown in Equation 2.6. These features are commonly used in CRFs to encourage intensity and spatial coherence, based on the assumption that voxels that have a similar intensity or are close together are likely to belong to the same class.

We evaluate two state-of-the-art approaches to combine CRFs with predefined features with a CNN: 1. Apply the CRF as post-processing to refine the CNN outputs; 2. Implement the CRF as a neural network layer that can be trained together with the CNN in an end-to-end manner.

#### CRF as Post-processing

After we train a CNN model and get its predictions, we can apply CRF as a post-processing method to refine the results [37]. We refer to this method as *Postproc-CRF* (Figure 2.2a).

#### End-to-end Training CRF

The CNN and CRF can be combined more elegantly by optimizing them together in an end-to-end manner [39] (Figure 2.2b), which allows the CRF to influence the CNN optimization. The end-to-end CRF uses the same pairwise potentials as that in the post-processing CRF (Equation 2.6). We refer to this variant as *Intensity-CRF*.

To investigate the spatial term in the end-to-end CRF, we can also use only the position features as the CRF feature space, which means that the CRF layer will only encourage nearby voxels to have the same class. We implement this CRF by setting the weight of the appearance kernel $\omega_1$ to zero and make it not trainable. We refer to this method as *Spatial-CRF*.

### 2.3.4 Proposed CRF with Learning-based Features

Our proposed CRF uses a learning-based feature space. We replace the intensity feature vector $I$ in the CRF kernel (Equation 2.6) with the new feature vector $F(\mathbf{I})$ from the CNN feature maps. The information in these CNN feature maps differs per level: in the first level of UNet the feature maps contain information close to the intensity, while in the last level of the UNet they contain more context for each voxel and potentially more class-discriminative information.

We refer to the CRF that uses features learned by CNN as *feature-learning-based CRF* (see Figure 2.2c) and refer to the specific form of CRF using the features in the last CNN softmax layer as *Posterior-CRF* (see Figure 2.3).

Unlike the CRFs with predefined features, our CRF takes CNN feature maps as the reference maps and updates the random field $\mathbf{X}$ based on $F(\mathbf{I})$ instead of on $\mathbf{I}$ directly. Compared to the original CRF pairwise potential in Equation 2.6, the feature $I$ is replaced with $F(\mathbf{I})$ and the new pairwise potential becomes:

$$\varphi_p(x_i, x_j) = \mu(x_i, x_j)[\omega_1 \mathbf{exp}(-\frac{|p_i - p_j|^2}{2\theta_\alpha^2} - \frac{|F_i(\mathbf{I}) - F_j(\mathbf{I})|^2}{2\theta_\beta^2}) + \omega_2 \mathbf{exp}(-\frac{|p_i - p_j|^2}{2\theta_\gamma^2})] \quad 2.7$$

### Back-propagation of the Learning-based CRF

The back-propagation of the proposed end-to-end feature-learning-based CRF is shown in Figure 2.4. There are five steps within one optimization iteration. Steps 1∼3 are the forward process that generates the output of the CNN. In the 4th step, CRF weights will adapt to the outputs calculated by the reference maps and unary maps, both given by CNN feature maps before back-propagation. In the 5th step, CNN weights are updated to provide new unary maps and reference maps for CRF for the next iteration. When the optimization converges, both CNN and CRF weights become stable close to their optimal values. Note that the mean-field inference in CRF happens in the forward process (after step 2 and before step 3) and thus contributes to the gradient updates of both CNN and CRF weights. The derivation of the mean-field inference gradient is omitted due to the length of the paper and can be found in Section 4.2 of the paper by Zheng et al. [39].

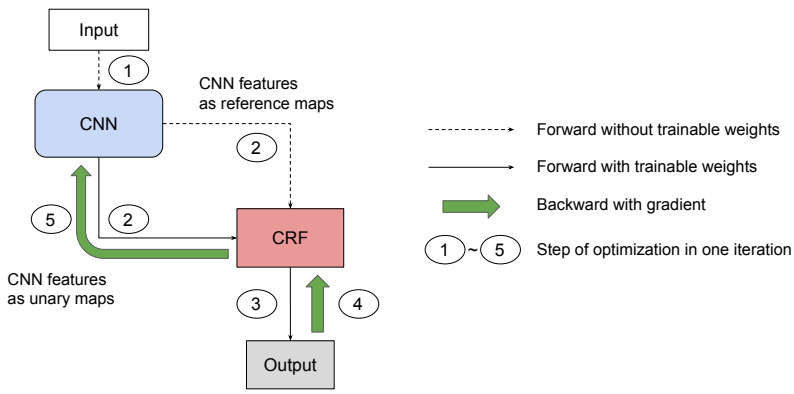



***Figure 2.4: One end-to-end optimization iteration of the proposed CRF method.*** *Best viewed in color with zoom.*

## 2.4 Experiments

In this section, we present experiments to evaluate the proposed method and compare it to the baseline methods: 3D UNet, Post-processing CRF, Intensity-CRF, and Spatial-CRF. Implementation details are discussed in Section 2.4.1, followed by the experimental settings (Section 2.4.2), the description of the datasets and pre-processing (Section 2.4.3), data augmentation and training details (Section 2.4.4) and evaluation metrics (Section 2.4.5).

### 2.4.1 Implementation

#### CNN Implementation

We implement all the algorithms in the TensorFlow framework. The detailed CNN architecture for the experiments is shown in Figure 2.3. All convolution layers use ReLU as the activation function except for the last output layer, which uses softmax to produce the final probability maps. For a fair comparison, the 3D UNet architecture that is tuned for the CNN baseline method is applied to all the CRF methods in Table 2.3. The 5-layer depth of UNet (tuned from 3 to 6) and 32 base feature maps (tuned from 8 to 64) are tuned based on all three datasets.

All segmentation models are optimized by minimizing the Dice loss [52]:

$$\mathcal{L}_{dc} = -\frac{2}{|C|}\sum_{c\in C}\frac{\sum_{i\in I} u_i^c v_i^c}{\sum_{i\in I} u_i^c + \sum_{i\in I} v_i^c} \tag{2.8}$$

where $v_i^c$ is the predicted probability that voxel $i$ belongs to the $c$th class. $u_i^j$ is the true label. The loss is minimized using the Adam optimizer [53].

#### CRF Implementation

In CRF, mean-field approximation can be used to calculate the maximum a posteriori probability (MAP) of the inference. We use an efficient approximation algorithm for mean-field inference [36, 54] built on a fast high-dimensional filtering using the permutohedral lattice [55] that allows voxel-wise fully-connected CRF to be iteratively computed in linear time. For a fair comparison, all the CRF methods in this paper are implemented in 3D fully-connected manner. The codes are publicly available: https://github.com/ShuaiChenBIGR/Posterior-CRF.

### 2.4.2 CRF Settings

#### Post-processing CRF

For *Postproc-CRF*, we fix the label compatibility $\mu$ in Equation 2.6 to the identity matrix, which means that the CRF does not model label-specific interaction. In the case of multi-modal input, each imaging modality has a specific $\theta_\beta$ to control the strength of the intensity term.

#### End-to-end CRF with Predefined Features

We consider two forms of end-to-end CRFs with predefined features: *Intensity-CRF* uses intensity of the input image $\mathbf{I}$ and position information as its feature space. *Spatial-CRF* uses only the position information (the smoothness term in Equation 2.6). The label compatibility is a $C \times C$ parameter matrix which is optimized during training to allow the CRF to learn the label compatibility automatically. The weights $\omega_1$ of the appearance kernel for *Intensity-CRF* and $\omega_2$ of the spatial kernel for *Spatial-CRF* are $C \times C$ matrices, which we restrict to diagonal matrices because the relationship between classes is already covered by the label compatibility matrix. Inner product is

calculated by multiplying the matrices. For simplicity, only one $\theta_\beta$ is applied for all modalities.



#### End-to-end CRF with Learned Features

The proposed *Posterior-CRF* uses the last softmax layer of the CNN as its reference map. The hyperparameters are the same as end-to-end CRF with predefined features. Note that Posterior-CRF is a special case of the feature-learning-based CRF. We can also use early CNN feature maps as CRF reference maps. An ablation study investigating other CRF variants can be seen in Section 2.5.4.

#### CRF Parameters

Parameters in the post-processing CRF for each dataset were obtained by grid search on the validation set and are shown in Table 2.1. We computed results with 500 different configurations of Postproc-CRF on each dataset for grid-search. Parameters in the end-to-end CRFs (*Intensity-CRF*, *Spatial-CRF*, *Posterior-CRF*) are initialized with the same values as were used in post-processing CRF. Although the end-to-end CRF approaches have the ability to learn CRF weights automatically during training, we initialize all CRF approaches in the same way to facilitate visualization of the evolution of CRF parameters during training (see Figure 2.5). We study the sensitivity to different CRF parameter initializations in Section 2.5.3.

The initial label compatibility matrix is set to an identity matrix and can be optimized during training. In the multi-modality case, the initial value of $\theta_\beta$ is averaged over all modalities. The initial values for each dataset are shown in Table 2.2.

#### Computation Costs of CRF

The training and testing time of the proposed CRF method is the same as Intensity-CRF but a bit slower than Spatial-CRF, since there is no bilateral term in Spatial-CRF. Although the proposed CRF uses CNN's features to compute the pairwise potential, the gradients only flow through the unary map path but not the reference map path which is the same as that in traditional Intensity-CRF. Therefore, there is no additional time and memory cost of the proposed method compared to traditional end-to-end CRF approaches with fixed feature space. Post-processing CRF is after the CNN training and takes more time for inference compared to the end-to-end CRFs, since the inference is done by CPU but not GPU.

### 2.4.3 Datasets and Preprocessing

We evaluate the proposed method on three segmentation problems: CT arteries, MRI white matter hyperintensities, and MRI ischemic stroke lesions. We chose these problems to study the generalizability of the method as these applications differ a lot in object shapes and appearances, imaging modalities, and suffer from different problems (see Fig. 2.1).

#### CT Arteries Dataset

We use 25 non-contrast lung CT scans from 25 different subjects enrolled in the Danish Lung Cancer Screening Trial (DLCST) [56]. The selection of the 25 subjects was completely random and it was done before the development of this algorithm for an unrelated study. The aorta and pulmonary artery were manually segmented by a trained observer (ZS). Images have an anisotropic voxel resolution of 0.78mm $\times$ 0.78mm $\times$ 1.00mm and are of size 512x512 with on average 336 slices (range 271-394). The 25 scans are split into three parts of 10, 5, and 10 scans for training, validation, and testing respectively. Due to the limitation of GPU memory, we first crop the original CT images and only keep the axial central part of $256 \times 256$ voxels for all slices. Then, 3D patches of the size $256 \times 256 \times 16$ are extracted from the cropped images. All training patches have 80% overlap in z-axis between neighboring patches to mitigate border effects. In total, there are 840 3D patches for training. We use the original CT intensities without normalization.

#### MRI White Matter Hyperintensities (WMH) Dataset

The White Matter Hyperintensities (WMH) Segmentation Challenge [34] provided images from 60 subjects (T1 and FLAIR) acquired from three hospitals and manually segmented for background and white matter hyperintensities. We randomly split these in 36 subjects for training, 12 for validation, and 12 for testing. For each subject, we cropped/padded MRI images into a constant size $200 \times 200 \times Z$, where $Z$ is the number of slices in the image. We use Gaussian normalization to normalize the intensities inside the brain mask in each image to zero mean and unit standard deviation. We extract training patches of size $200 \times 200 \times 16$ with 80% overlap in z-axis between patches. In total, there are 528 3D patches for training.

#### MRI Ischemic Stroke Lesions (ISLES) Dataset

The ISLES 2015 Challenge [57] is a public dataset of diverse ischemic stroke cases. There are 4 MRI sequences available for each patient (T1, T2, FLAIR, and DWI). We use the sub-acute ischemic stroke lesion segmentation (SISS) dataset (28 subjects) with the lesion labels for experiments and randomly split them as 14 for training, 7 for validation and 7 for testing. The images are cropped/padded to the size $200 \times 200 \times Z$. Gaussian normalization is applied for normalizing the intensities in each image. Training patches of the size $200 \times 200 \times 16$ with 80% overlap in z-axis are extracted. In total, there are 560 3D patches for training.

### 2.4.4 Data Augmentation and Training Details

The network is trained on all mini-batches (each mini-batch contains one 3D patch). For each 3D patch in the current mini-batch we apply 3D random rotation sampled from ([-5,5],[-5,5],[-10,10]) degrees, shifting ([-24,24],[-24,24],[-7,7]) voxels, as well as random horizontal (left and right) flipping. We stopped training when the validation loss is not decreasing anymore and chose the model that achieved the best validation performance. The experiments are run on an Nvidia GeForce GTX1080 GPU. The

average training time is 5∼10 hours for one CNN baseline model and 1∼2 hours more when the CRF layer is added.

### 2.4.5 Evaluation Metrics

We use four voxel-wise metrics of segmentation quality: Dice similarity coefficient (DSC), indicating the relative overlap with the ground truth (larger is better); 95th percentile Hausdorff distance (H95), showing the extremes in contour distance from ground truth to the prediction (smaller is better); Average volume difference (AVD) as a percentage of the difference between ground truth volume and segmentation volume over ground truth volume (smaller is better), and Recall score (larger is better). For the lesion segmentations (WMH and ISLES), we additionally assess accuracy of lesion detection by computing the lesion-wise Recall and lesion-wise F1 score (larger is better). The lesion-wise metrics use the 3D connected components, while the voxel-wise metrics do not use 3D connected components. The correct detection of a lesion is determined by the overlap (at least one voxel) of the 3D components. F1 score is equivalent to lesion-wise Dice score and is calculated by 2*(precision*recall)/(precision+recall), where precision is calculated by true positives/(true positives+false positives).

## 2.5 Results

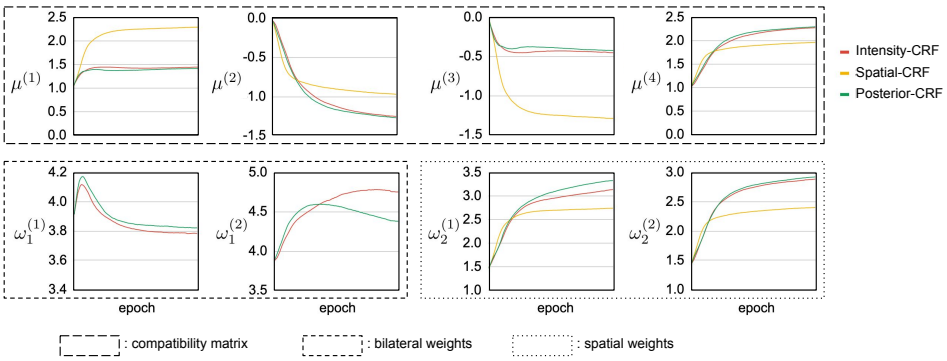



***Figure 2.5: CRF parameters during training in WMH dataset.*** *The initial values of the CRF parameters can be found in Table 2.2. Best viewed in color with zoom.*

### 2.5.1 Segmentation Results

Table 2.3 shows the segmentation results for all three datasets. In most metrics, Posterior-CRF had the best performance in all datasets. For all datasets, CNN without CRF provides good baseline results, which indicates that 3D UNet is an efficient architecture to extract useful features for segmentation in these applications. Intensity-CRF performed worse on DSC than Posterior-CRF (statistically significant in

***Table 2.1: Post-processing CRF parameters for each dataset.*** *Search range indicates the range of parameter values explored during grid search.*

| Datasets | *CT Arteries* | *WMH* | *ISLES* | Search range |
|---|---|---|---|---|
| $\omega_1$ | 6.39 | 3.85 | 9.75 | (0.1, 10) |
| $\theta_\alpha$ | 4.09 | 4.46 | 8.74 | (0.1, 10) |
| $\theta_\beta$ for CT | 1.10 | - | - | (0.1, 10) |
| $\theta_\beta$ for T1 | - | 7.01 | 9.26 | (0.1, 10) |
| $\theta_\beta$ for T2 | - | - | 9.73 | (0.1, 10) |
| $\theta_\beta$ for FLAIR | - | 2.64 | 2.36 | (0.1, 10) |
| $\theta_\beta$ for DWI | - | - | 6.85 | (0.1, 10) |
| $\omega_2$ | 3.40 | 1.41 | 2.34 | (0.1, 10) |
| $\theta_\gamma$ | 4.83 | 0.11 | 1.35 | (0.1, 10) |
| Iterations | 3 | 1 | 2 | (1, 5) |

***Table 2.2: Initial end-to-end CRF parameters for each dataset.***

| Methods | $\omega_1$ | $\theta_\alpha$ | $\theta_\beta$ | $\omega_2$ | $\theta_\gamma$ | Iterations |
|---|---|---|---|---|---|---|
| | | | *CT Arteries* | | | |
| Spatial-CRF | - | - | - | 3.40 | 4.83 | 3 |
| Others | 6.39 | 4.09 | 1.10 | 3.40 | 4.83 | 3 |
| | | | *WMH* | | | |
| Spatial-CRF | - | - | - | 1.41 | 0.11 | 1 |
| Others | 3.85 | 4.46 | 4.83 | 1.41 | 0.11 | 1 |
| | | | *ISLES* | | | |
| Spatial-CRF | - | - | - | 2.34 | 1.35 | 2 |
| Others | 9.75 | 8.74 | 7.05 | 2.34 | 1.35 | 2 |

aorta segmentation and WMH segmentation), which reveals the limitation of intensity features. Among all end-to-end CRF methods, Spatial-CRF performs worst for all datasets except ISLES. From these results, we conclude that spatial coherence alone is not sufficient and often detrimental to segmentation accuracy, and that the CNN features in the last layer are more informative for CRF than the intensity features in the original images.

CRFs that depend strongly on intensity-based features have difficulties detecting objects that are similar in intensity. Examples of this problem can be observed in the segmentations for the CT arteries and ISLES datasets (Figure 2.6). In CT arteries segmentation, the aorta and pulmonary artery have very similar intensities, which causes most of the methods in our experiments to sometimes misclassify part of the aorta as pulmonary artery. This is especially true for Post-processing CRF but also for Intensity-CRF.

Posterior-CRF achieves a DSC segmentation overlap of 95.4% and an H95 lower

2

***Table 2.3: Results.*** *Mean (standard deviation). The best results are marked in bold. Each experiment is repeated 5 times with different random data split. The last two colomns are lesion-wise metrics. *: significantly better than CNN baseline (p<0.05). ◇: significantly worse than Posterior-CRF (p<0.05). P-values are calculated by two-sided paired t-test. All CRF methods are implemented in 3D fully-connected manner and share the same CNN architecture and hyperparameters.*

| Methods | DSC | H95(mm) | AVD(%) | Recall | Recall(lesion) | F1(lesion) |
|---|---|---|---|---|---|---|
| *CT Arteries: Aorta* | | | | | | |
| CNN baseline | 0.9291(0.02)◇ | 5.5560(1.96)◇ | 6.8780(4.17)◇ | 0.8993(0.03)◇ | N/A | N/A |
| Postproc-CRF | 0.9264(0.02)◇ | 5.1591(1.59)◇ | 8.5326(4.81)◇ | 0.8878(0.04)◇ | N/A | N/A |
| Intensity-CRF | 0.9457(0.01)*◇ | 3.2802(0.77)*◇ | 3.1967(2.58) | 0.9548(0.02)* | N/A | N/A |
| Spatial-CRF | 0.9188(0.02)◇ | 7.6562(3.98)◇ | 6.1013(5.13)◇ | 0.8939(0.05)◇ | N/A | N/A |
| Posterior-CRF | **0.9538**(0.01)* | **2.8699**(0.86)* | **2.3688**(2.29)* | **0.9555**(0.02)* | N/A | N/A |
| *CT Arteries: Pulmonary Artery* | | | | | | |
| CNN baseline | 0.8510(0.05)◇ | 10.3000(4.87)◇ | 16.7687(12.60)◇ | 0.8867(0.09) | N/A | N/A |
| Postproc-CRF | 0.8561(0.05) | 10.0052(5.22)◇ | 13.7071(10.26)◇ | 0.8698(0.09)◇ | N/A | N/A |
| Intensity-CRF | 0.8773(0.04)* | 8.9208(3.09)* | 11.8671(8.66)* | **0.9079**(0.06) | N/A | N/A |
| Spatial-CRF | 0.8558(0.06)◇ | 10.5672(5.19)◇ | 13.7399(13.47) | 0.8603(0.09)◇ | N/A | N/A |
| Posterior-CRF | **0.8935**(0.04)* | **7.6635**(3.92)* | **8.9245**(7.07)* | 0.8979(0.07) | N/A | N/A |
| *WMH* | | | | | | |
| CNN baseline | 0.7557(0.13)◇ | 6.5015(9.87)◇ | 28.3351(45.64)◇ | **0.7977**(0.14) | 0.6476(0.14) | 0.6648(0.11)◇ |
| Postproc-CRF | 0.6970(0.17)◇ | 8.8659(7.79)◇ | 35.0786(22.69)◇ | 0.5947(0.20)◇ | 0.3476(0.16)◇ | 0.4831(0.16)◇ |
| Intensity-CRF | 0.7706(0.10)◇ | 4.9403(4.58) | 15.6263(16.44)* | 0.7751(0.12) | 0.6803(0.15)* | 0.6705(0.10)◇ |
| Spatial-CRF | 0.7602(0.11)◇ | 5.8469(5.82)◇ | 23.5154(25.76)◇ | 0.7831(0.13) | **0.6876**(0.14)* | 0.6569(0.11)◇ |
| Posterior-CRF | **0.7887**(0.09)* | **4.2972**(3.87)* | **14.8427**(12.66)* | 0.7707(0.12) | 0.6670(0.14) | **0.6952**(0.10)* |
| *ISLES* | | | | | | |
| CNN baseline | 0.5795(0.28) | 27.6725(25.58) | 72.3048(121.12) | 0.6590(0.31) | 0.7586(0.33) | 0.4941(0.35) |
| Postproc-CRF | 0.5621(0.31) | **19.5302**(20.72) | 59.1030(85.99) | 0.6132(0.34) | 0.6518(0.39) | 0.5545(0.36) |
| Intensity-CRF | 0.5758(0.26) | 46.6002(32.17)◇ | 65.9278(68.98) | 0.6397(0.30) | 0.7350(0.33) | 0.4094(0.31)◇ |
| Spatial-CRF | 0.5898(0.26) | 31.1519(29.50) | 93.1006(171.83) | **0.6794**(0.28) | **0.7848**(0.31) | 0.4945(0.34) |
| Posterior-CRF | **0.6075**(0.24) | 25.1834(23.27) | **47.5171**(38.34) | 0.6501(0.29) | 0.7443(0.31) | **0.5625**(0.32) |

than 2.87mm in aorta segmentation, which is significantly better than all other methods on this dataset. We argue that this is because the features from the last CNN feature maps are more informative than the intensity-based features, which allows the CRF inference to focus on refining the object boundary without expanding into neighboring class voxels with similar intensities. The Posterior-CRF also gives a performance improvement in the segmentation of the pulmonary artery, but this is not always statistically significant. One reason is that the blurred boundary between the aorta and pulmonary artery often results in the oversegmentation of pulmonary artery, the errors in pulmonary artery are emphasized because the overall pulmonary artery volume is lower. Another reason could be the curved shape of the pulmonary artery, which makes the results vary a lot between patients.



We see similar behavior on the ISLES dataset. The intensity boundaries of the large ischemic stroke lesions are ambiguous and their appearance varies a lot between lesions. Most of the methods fail to segment the boundaries accurately (see Figure 2.6 ISLES). Post-processing CRF hardly solves the problem and performs slightly worse than CNN. Posterior-CRF achieves better (while less significant due to the large prediction variance between samples) segmentation performance on DSC, AVD, lesion-wise F1.

A properly tuned spatial component of the post-processing CRF can benefits CT arteries and ischemic stroke lesion segmentation (Appendix Section 2.9.2, Figure 2 (a) and (c)). However, it can cause problems to white matter hyperintensities no matter how we try to tune it (Appendix Section 2.9.2, Figure 2 (b)), where we can see a positive $\omega_2$ always leads to a decreased performance since the spatial smoothing contributes to remove both isolated true positives and false positives if they are small enough. The complete SHAP analysis will be discussed in Appendix Section 2.9.2.

The negative effect of the spatial smoothing results in the low average lesion-wise recall score in WMH segmentation for Postproc-CRF (34.8%) and can be observed in the WMH segmentation results (see Figure 2.6). In this case, Postproc-CRF is always worse than vanilla CNN (within our grid-search range). This is because the scenario where post-processing CRF has no influence (with both $\omega_1$ and $\omega_2$ set to zero) was not included in the grid search range (0.1,10). Intensity-CRF has a higher lesion-wise average recall than CNN baseline (68% to 64.8%) but a lower (not significantly) voxel-wise recall (77.5% to 79.8%): although it detects more correct lesions than CNN due to the intensity features, its use of spatial features causes it to undersegment individual lesions (see Figure 2.6). Spatial-CRF also suffers from this problem, with a high lesion-wise recall of 68.8% but low lesion-wise F1 of 65.7%.

For CT arteries, the proposed method performs better than the state-of-the-art [41] in aorta segmentation (0.95 vs. 0.94) and worse in pulmonary segmentation (0.89 vs. 0.92). Note that five-fold cross-validation is applied in [41] and in this paper we apply five random data splits, which may lead to different test data. Unlike in [41], we do not cut the pulmonary artery prediction from the bottom level. In some cases, our method produces segments that extend beyond the manual annotations, which leads to a lower Dice performance. For WMH, the proposed method performs slightly worse than the best performance in the leaderboard using 5 2D UNet ensembles (0.78 vs. 0.81) using the same test data. The top 3 methods in the leaderboard are all 2D UNet ensembles (0.81 vs. 0.80 vs. 0.80), which shows a well-tuned UNet can provide strong

baseline performance for WMH segmentation. The best non-ensemble approach is brain atlas guided attention UNet which is more comparable to the proposed method (0.79 vs. 0.78). For ISLES, note that the test sets used in this paper are different from the ones that are used to calculate the leaderboard performance. The performance of the proposed method using 14 training images is quite comparable to the best performance in the leaderboard (0.61 vs. 0.59), which is the only CNN-based method [31] among the top-3 methods in Dice metrics (0.59 vs. 0.55 vs. 0.47).

### 2.5.2 Optimization of the End-to-end CRF

We show the evolution of the trainable CRF parameters in one data split of WMH dataset in Figure 2.5. For the four parameters in the $2 \times 2$ compatibility matrix $\mu$ and the two diagonal spatial kernel weights $\omega_2$, Spatial-CRF falls into different local optimal values compared to other CRF methods, probably because different parameter scaling due to the lack of the appearance kernel. In contrast, Intensity-CRF and Posterior-CRF converged to similar optimal values for $\mu$ and $\omega_2$. For the two diagonal bilateral kernel weights in $\omega_1$ that control the appearance kernel, Intensity-CRF and Posterior-CRF converged to two different optimal values. This suggests that different CRF feature spaces contribute mostly through the appearance kernel and less through the compatibility matrix or the spatial kernel. Interestingly, for the second diagonal bilateral weight $\omega_1^{(2)}$, there is a different trend of Posterior-CRF compared to Intensity-CRF, which may indicate that at the early training stage Posterior-CRF uses similar feature space like that in Intensity-CRF, but at the later stage it finds and learns another set of features that may help categorize the lesion class better, which are more reliable than the original intensity features.

### 2.5.3 Influence of CRF Hyperparameters

We conduct experiments to investigate the influence of CRF hyperparameters on both end-to-end CRF with predefined features and the proposed CRF with learned features.
**Trainable CRF parameters.** The CRF weights $\mu$, $\omega_1$, and $\omega_2$ in the end-to-end CRF learning can be automatically updated together with CNN weights. We run Intensity-CRF and Posterior-CRF using WMH datasets with five different initializations of CRF weights randomly sampled from the search scale with all other parameters the same as in Table 2.2. The CNN initializations are the same for all experiments. The results in Table 2.4 show that Intensity-CRF and Posterior-CRF converge to similar optimal points across different initializations. Spatial-CRF shows higher variances across experiments and is less stable to the change of initializations. Posterior-CRF is more robust to changes in initialization, achieving higher average performance and smaller standard deviations compared to Intensity-CRF and Spatial-CRF.
**Empirically tuned parameters.**The CRF standard deviation parameters $\theta_\alpha$ and $\theta_\gamma$, controlling the spatial terms, and $\theta_\beta$ controlling the appearance term, were tuned empirically to give the best results for post-processing CRF. We here test, for WMH segmentation, five different values of $\theta_\alpha$, $\theta_\beta$, and $\theta_\gamma$ for Intensity-CRF and Posterior-CRF and five different values of $\theta_\gamma$ for Spatial-CRF within the search scale. All other parameters are the same as in Table 2.2. The results are shown in Figure 2.7. We

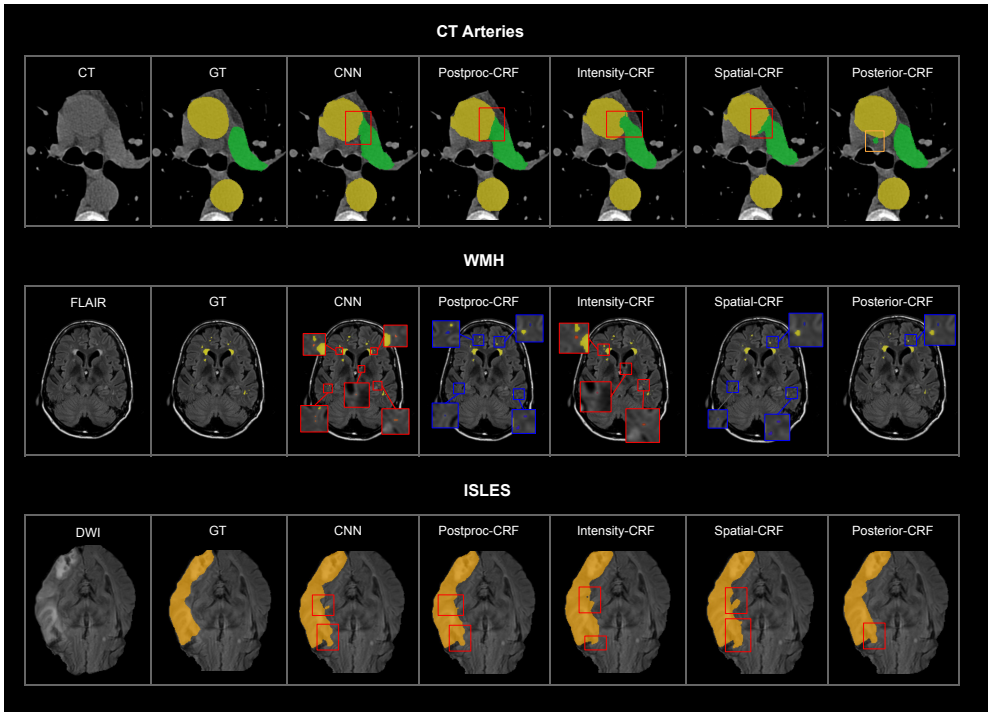



***Figure 2.6:*** ***Example segmentation results.*** *From left for each row: (1) Original image (2) Manual annotation (3) CNN baseline (4) Postproc-CRF (5) Intensity-CRF (6) Spatial-CRF (7) Posterior-CRF. Aorta is colored with yellow and the pulmonary artery is green, white matter hyperintensities and ischemic stroke lesions in yellow. Red/blue rectangles indicate areas with over/under segmented voxels and the orange rectangle indicates another branch of pulmonary artery whose annotation starts in the next few slices and merged with the main branch gradually. In the WMH example (second row), only detections that do not overlap with any ground truth voxel (false positive lesions) or ground truth lesions for which no voxel is detected (false negative lesions) are highlighted, and in the zoomed patches red and blue voxels indicate false positive and false negative lesions respectively. Better viewed in color with zoom.*

can see that Posterior-CRF is more robust to $\theta_\alpha$ and $\theta_\beta$ and has consistently better performance than Intensity-CRF within the search scale, suggesting that Posterior-CRF parameters are more easy to tune. All CRF methods degenerate performance when $\theta_\gamma$ becomes larger and show the best performance when using a similar value as that in the grid search for post-processing CRF. Spatial-CRF is more robust to $\theta_\gamma$ compared to other CRF methods and has similar performance as CNN baseline with larger $\theta_\gamma$. This indicates that large $\theta_\gamma$ reduces the CRF effect and the spatial term

***Table 2.4: Performance (Dice score) across 5 different initializations of CRF weights on WMH dataset.***

| Methods | Intensity-CRF | Spatial-CRF | Posterior-CRF |
|---|---|---|---|
| Mean (std) | 0.7570 (0.008) | 0.7507 (0.02) | **0.7833** (**0.003**) |

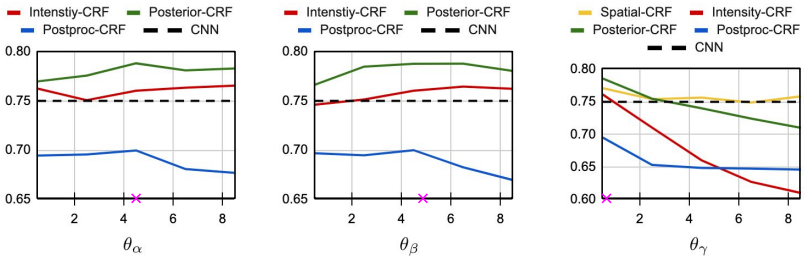



***Figure 2.7: Dice performance of varying $\theta$ for CRF methods on WMH dataset.*** *CNN result is shown as the black dash line. Purple crosses indicate the values used in Table 2.4. Best viewed in color with zoom.*

may introduce more incorrect segmentation when there is also an appearance term in the end-to-end CRF like Intenity-CRF and Posterior-CRF.

### 2.5.4 Influence of Hierarchical CNN Features as CRF Reference Maps

We conduct experiments to investigate which level of features – earlier or deeper in the network – are more useful for the feature-learning-based CRF. We implement nine variants of feature-learning-based CRF with different levels of CNN feature maps as reference maps in the same 3D UNet architecture. For example, the method *FL-CRF-e-1* indicates the feature-learning-based CRF using the level 1 feature maps in the UNet encoder path as CRF reference maps. The implementation detail of FL-CRF-e-1 is shown in Figure 2.3. To reduce the computational cost and keep the same layer capacity as Posterior-CRF, the 32-channel (or more in deeper layers) feature maps are encoded into $C$-channel feature maps and go through a softmax layer as the CRF reference maps. Since there is no gradient flowing back through the reference map path, we optimize the softmax layer with the segmentation loss directly in order to preserve as much semantic information as possible. Note that for CRF methods that use deeper CNN layers as reference maps, such as FL-CRF-e-2 to FL-CRF-d-2, we upsample the reference maps to the original image scale using nearest neighbor interpolation and optimize them with the segmentation loss, similar to FL-CRF-e-1.

The results are shown in Figure 2.8. Note that if we use the CNN input as CRF reference maps, it turns into Intensity-CRF; if we use the last CNN layer as CRF

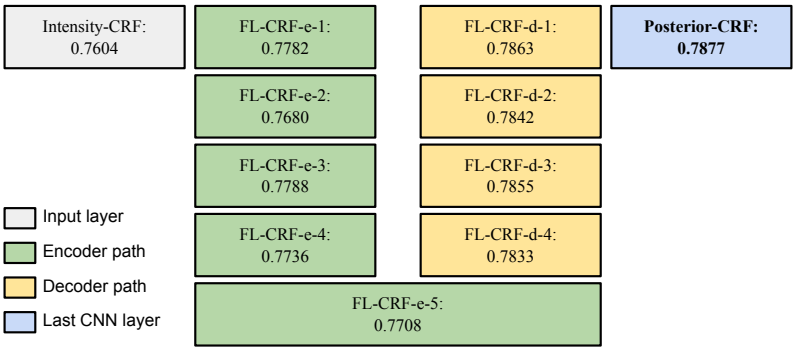



***Figure 2.8:*** ***Dice performance of end-to-end CRFs using different CNN feature maps in an independent run on WMH dataset.*** *Different blocks indicate different level of CNN feature maps used as CRF reference maps. Best viewed in color with zoom.*

reference maps, it turns into Posterior-CRF. In the figure, we can see that all feature-learning-based CRF approaches (including Posterior-CRF) outperform Intensity-CRF and the overall Dice performance in the decoder path is better than that in the encoder path, indicating that CNN learned features are more useful to the CRF inference than intensity is and later CNN features are more useful than early features. The performance degenerates towards the middle part of the UNet (from FL-CRF-e-1 to FL-CRF-e-5 and FL-CRF-d-1 to FL-CRF-d-4) but fluctuates at the 2nd/3rd level. We argue that this may be due to the pooling effect which enables CNN to extract higher-level features but loses the spatial information at the same time. Posterior-CRF achieves the best performance among all variants and we argue that this is because the last CNN layer are more likely to contain more useful information for CRF inference and it still keeps the same spatial scale as the original image.

### 2.5.5 Evolution of CNN and CRF Outputs

The concurrent optimization of CNN and CRF in our end-to-end models allows the CNN and CRF to interact during training. We observed that this has a strong effect on what the CNN learns in the early training epochs. Figure 2.9 shows the evolution of CNN and CRF outputs for three typical examples. The baseline CNN without CRF converges quickly and focuses on the large lesions, already producing a fairly sparse output after the first epoch. The end-to-end models converge more slowly, and in this case the output of the CNN is influenced by the choice of CRF mostly in the early stage of training. For example, the CNN in the Intensity-CRF model initially tends to highlight voxels with similar intensity as the foreground (1 to 20 epoch), while the CNN in the Spatial-CRF model preserves the spatial coherence between voxels and outputs many small groups of voxels (5 epoch). The CNN in the Posterior-CRF model first focuses on the coarse area that might contain the target lesions (1 to 5

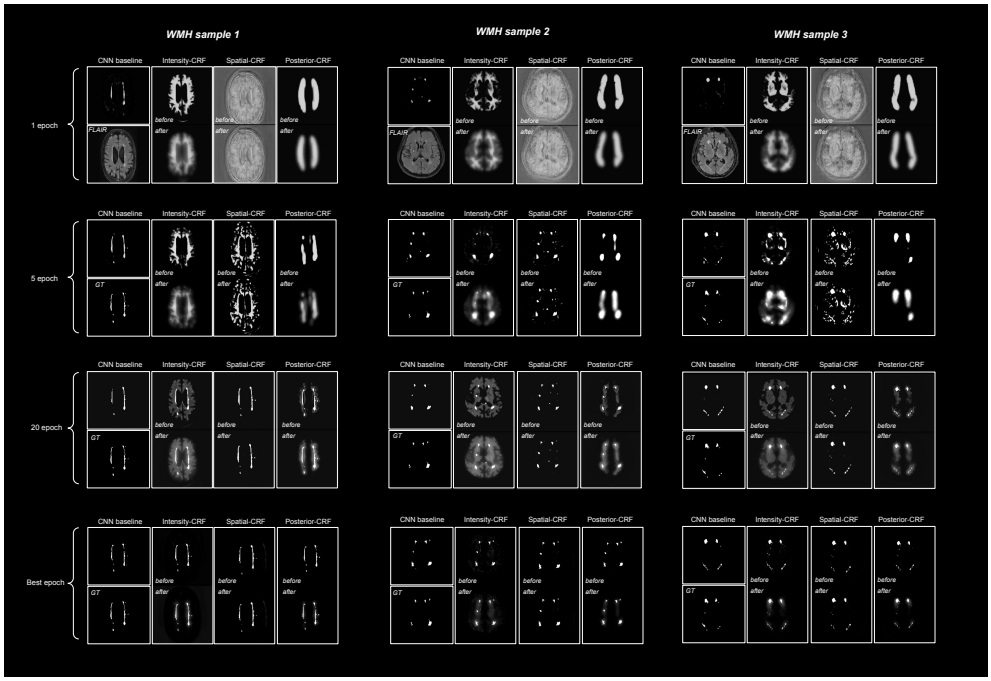



***Figure 2.9: Evolution of CNN and CRF outputs during training.*** *The CNN output maps and CRF results for WMH segmentation in 3 different MRI images (columns) are shown at, from top row to bottom row, epoch 1, 5, 20, and the best epoch. The best epoch is chosen when the model shows the best validation performance till the end of training (usually at 50∼80 epoch). FLAIR: the input FLAIR image of the current training sample. GT: ground truth. CNN baseline: the last layer (softmax output) of CNN. Intensity-CRF, Spatial-CRF, Posterior-CRF: the probability maps before/after the CRF layer at different epochs during training. Best viewed with zoom.*

epoch) and then refine the prediction gradually to the ground truth (5 to 20 epoch). Eventually, all models converge to a result close to the ground truth.

## 2.6 Discussion

In this paper, we explored efficient methods to combine the global inference capabilities of a CRF with the feature extraction from a CNN. Our end-to-end approach optimizes the CRF and CNN at the same time, and allows the two components of the approach to cooperate in learning effective feature representations. This gives our method an advantage over traditional CRFs that only use the original image intensities and position information. Intensity-based features can be suboptimal for problems where

the intensity does not provide sufficient information to find the object boundaries, for example because the contrast between objects is too small.

Unlike other CRF methods, our Posterior-CRF uses adaptive learning-based features that are learned by the CNN and can combine spatial and appearance information in a way that suits the CRF. The results show our method can achieve stable, good performance across a range of segmentation applications and imaging modalities. FL-CRF variants that use early CNN features in Section 2.5.4 achieve in-between performance between Intensity-CRF and Posterior-CRF, using learning-based features that range from more similar to intensity to more similar to posterior probability maps. Finally, we found that integrating learned features into the CRF model reduces the need to fine-tune CRF parameters, making the method easier to apply than CRF methods with predefined features.



### 2.6.1 Interaction between CRF and CNN

Figure 2.9 leads to the counter-intuitive observation that, at least initially, the CNNs in end-to-end models seem to imitate the CRF instead of complementing it. For example, the CNN output in Intensity-CRF highlights the ground truth, but also finds areas with similar intensities, producing something that looks very similar to the original image (20 epoch). The CNN output in Spatial-CRF selects the ground truth but also includes clusters of voxels in other areas (5 epoch).

This effect can be explained by the way the CNN and CRF interact during training. In Intensity-CRF and Spatial-CRF, the only interaction between CRF and CNN takes place through the unary map (Figure 2.4, step 5, green arrow). For example, consider how this works in the Intensity-CRF. In WMH segmentation, the ground truth is usually high-intensity area. However, for the voxels with high intensities but not the target lesions, it is difficult to get both low pairwise CRF potentials and low segmentation loss, since labeling them as non-lesion goes against the CRF assumption that voxels with similar high-intensities are more likely to be the lesion class. For convenience, we call these voxels as *hard voxels*, indicating the voxels that do not fit the CRF assumption. In order to keep the correctly segmented lesions and reduce the CRF effect on the hard voxels at the same time, the CNN tends to provide unary maps that 1) highlight the ground truth area for lower segmentation loss, and 2) look similar to the CRF reference maps on the hard voxels for lower pairwise CRF potentials. In the later stage of training, CNN is encouraged to push the confidence of its outputs even further to minimize unary potentials and thus prevent CRF from undoing segmentation improvement on the hard voxels. From Figure 2.9, we can see that there are many hard voxels in Intensity-CRF (1 to 20 epoch, areas that look like the original image) and Spatial-CRF (5 epoch, clusters of voxels that do not belong to the ground truth) which may harm the segmentation. This indicates that the predefined features may not be the optimal feature space for the end-to-end CRF.

In the Posterior-CRF model, the CRF inference happens within the CNN feature space, which can improve the interaction between CNN and CRF. First, the features learned by CNN during training may contain information that is more useful for segmentation than that in the predefined features, which makes CRF benefit most from the CNN features. Second, using the learning-based features as CRF reference

maps avoids the CRF assumption of the predefined features which may introduce many hard voxels, e.g., Intensity-CRF and Spatial-CRF, as discussed in the previous paragraph. With fewer hard voxels, the CNN in Posterior-CRF may provide better unary maps for the CRF inference.



### 2.6.2 Posterior-CRF vs. Mean-field Network

The mean-field approximation (MFA) in Posterior-CRF is somewhat similar to that in Mean-field networks (MFN) [58], since both methods use it to get the posterior probabilities of the variables. Therefore, MFN could be a promising alternative to the MFA process in our method. MFN has the advantage that it utilizes each layer of the network as an iteration of MFA, which has the advantage of allowing more relaxation on parameters and provides some efficiency improvements. This makes the idea of formulating Posterior-CRF as a feed-forward network like MFN very attractive. There are, however, a few limitations that would need to be solved.

The first limitation is in training. MFN is designed to provide a faster and more flexible way to obtain the prediction of MFA, by fitting a powerful function that predicts the real MFA result. To train an MFN, we first need to acquire the ground truth calculated by conventional mean-field iterations, which takes time during training but saves time during inference. On the other hand, Posterior-CRF provides a flexible and adaptive feature space for the conventional MFA, speeding up the procedure by applying Gaussian convolution in the message passing updates. As a result, the thing Posterior-CRF does is difficult to replicate with a MFN because the feature space of a Posterior-CRF changes during training, while MFN requires a predefined feature space to get the ground truth.

The second limitation is the tradeoff between dense inference and computation cost in the MFN. In its feed-forward network implementation, the computation cost increases exponentially when more neighbor nodes and number of layers are included, which limits its ability to model dense prediction problems such as segmentation tasks.

### 2.6.3 Posterior-CRF vs. Graph Neural Networks

The proposed Posterior-CRF shares some similarities with graph neural networks (GNN) [47, 48]: both approaches aim to model interactions between variables within a graph model. The difference is that Posterior-CRF pre-defines the global relationship between variables through the mean-field assumptions and solves the maximum a posteriori problem, whereas GNN learns the global variable relationship by applying graph convolution filters and mapping the input graph to the output graph [48].

It could be interesting to combine the global view of the Posterior-CRF and the more local view of the GNN. The Posterior-CRF might benefit from using a GNN to replace its CNN component for feature extraction. The graph-based network may extract better features for Posterior-CRF than a CNN, which is not designed to extract unary and pairwise features for a graphical model. Similarly, the GNN may benefit from the efficient message passing of the Posterior-CRF, which would allow it to use the local graph-based features as CRF features for global interactive modeling in a computationally efficient way.

### 2.6.4 Limitations

In this paper, we show that the proposed Posterior-CRF method has benefits in the three medical imaging applications. Considering the medical imaging datasets are usually small largely because the manual annotations are very expensive to make, difference between Posterior-CRF and UNet may be smaller in larger training sets. But we know from literature that Intensity-CRF helps in some computer vision applications with large training sets (e.g., 10k 2D images or even more), it would be promising to test our method on these datasets. This is considered as our future work.



In Section 2.5.3, we show that Posterior-CRF is robust to different CRF initializations and hyperparameters. However, the standard deviation parameters still require careful tuning, especially for $\theta_\gamma$ in the spatial term. $\theta_\gamma$ is sensitive to the image scale of different datasets and the size of the target object in different applications. Nevertheless, we recommend the researchers to use the default (or optimal if it is available) setting of post-processing CRF as a reference for tuning Posterior-CRF rather than random initialization. Posterior-CRF is more robust to $\theta_\alpha$ and $\theta_\beta$ compared to Intensity-CRF, which facilitates exhaustive tuning of these parameters.

The computational expense of the CRF also restricts the choice of applications. Compared to UNet ($\sim$5 mins for 1 epoch in WMH experiment), there is around 20% training time increased on average when applying a CRF layer on top of the network ($\sim$6 mins for 1 epoch). All end-to-end CRFs share similar computational costs. Given that Posterior-CRF uses posterior probability maps as its reference maps, it can become computationally expensive in multi-class segmentation problems. For a similar reason, Intensity-CRF and Postproc-CRF can become expensive when there are too many imaging modalities in the input channels $M$.

In the experiments, we use a plain 3D UNet as the backbone network for all methods. The training pipeline and hyperparameters are determined empirically and kept the same for all datasets, which could be suboptimal compared to elaborate automatic configuration strategies like nnU-Net [52]. On the WMH dataset we therefore checked the performance of nnU-Net (3D version without ensembling). Average Dice score of nnU-net (0.77) was slightly higher than our CNN baseline (0.76, difference not statistically significant) but lower than the proposed posterior CRF using the CNN baseline as a backbone (0.79), which performed significantly better than the CNN baseline (see Table 2.3). Though our experiments have been limited to a standard 3D U-net architecture, We expect that posterior CRF can improve results of other segmentation architectures and other hyperparameter settings (such as nnU-net) as well.

## 2.7 Conclusions

In conclusion, we present a novel end-to-end segmentation method called Posterior-CRF that uses learning-based, class-informative CNN features for CRF inference. The proposed method is evaluated in three medical image segmentation tasks, including different MRI/CT imaging modalities and covering a range of object sizes, appearances and anatomical classes. In the quantitative evaluation, our method outperforms end-to-end CRF with early CNN features, end-to-end CRF approaches with predefined

features, post-processing CRF, as well as a baseline CNN with similar architecture. In two of the three applications, our method significantly improves the segmentation performance. The qualitative comparison demonstrates that our method has good performance on segmenting blurred boundaries and very small objects.



## 2.8 Acknowledgments

The authors would like to thank Raghavendra Selvan, Gerda Bortsova for their constructive suggestions for the paper, Dr. Zaigham Saghir from DLCST for providing us with the chest CT scans, and organizers of WMH 2017 and ISLES 2015 Challenges for providing the public datasets. This work was partially funded by Chinese Scholarship Council (File No.201706170040), Iranian Ministry of Science, Research and Technology, and The Netherlands Organisation for Scientific Research (NWO).

## 2.9 Appendix

### 2.9.1 Mean-field Inference

Mean-field inference is an efficient approximation to computing distribution $Q(\mathbf{X})$ instead of the real CRF distribution $P(\mathbf{X})$, which could be done in an iterative algorithm 1 (see also Figure 2.10). X is the random field w.r.t the current 3D image patch $\mathbf{I}$.

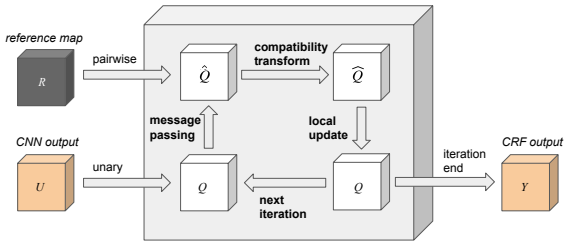



***Figure 2.10: Mean-field approximation in the end-to-end CRF layer****. There are two inputs of the CRF layer, where $U$ is the CNN probability maps as the unary maps and the pairwise distribution are calculated by the initialized distribution $Q$ and the reference map* $\mathbf{I}$*. The updated distribution $Y$ is the output of the layer at the end of the iteration. Best viewed in color with zoom.*

There are three main steps inside the inference iteration. First is message passing, which is the most calculation-intense step that could be expressed as a convolution operation on all the pairwise kernels $k$ and the initialized $Q(\mathbf{X})$. An efficient way to perform high-dimensional convolution is using permutohedral lattice algorithm [55]. In compatibility transform as the second step, all the convolution results $\hat{Q}_i^{(m)}(x_i)$ are weighted by $\omega^{(m)}$ in different sort of kernels and shared between labels to a varied

**Algorithm 1** Mean-field inference in fully-connected CRF

$Q_i(x_i) \leftarrow U_i(x_i), i = 1, 2, ..., N$ ▷ **Initialize** $Q(\mathbf{X})$
**while** not reach max iteration number **do**
  $\hat{Q}_i^{(m)}(x_i) \leftarrow \sum_{j \neq i} k^{(m)}(\mathbf{f}_i, \mathbf{f}_j) Q_j(x_i)$ for all $m$ ▷ **Message Passing**
  $\widehat{Q}_i^{(m)}(x_i) \leftarrow \sum_{l \in \mathcal{L}} \mu^{(m)}(x_i, l) \sum_m \omega^{(m)} \hat{Q}_i^{(m)}(l)$ ▷ **Compatibility Transform**
  $Q_i(x_i) \leftarrow \exp\{-\varphi_u(x_i) - \widehat{Q}_i(x_i)\}$ ▷ **Local Update**
  normalize $Q_i(x_i)$
**end while**

extent, depending on the compatibility $\mu$ between these labels. At last, $Q(\mathbf{X})$ will be updated by the calculated pairwise potential and used as the input for the next iteration.

### 2.9.2 SHAP Analysis of Post-processing CRF

We conduct SHAP (SHapley Additive exPlanations) [59] analysis on the post-processing CRF grid search results to investigate the contribution of each individual CRF parameter to the segmentation performance. With this analysis, we show that it is difficult to tune traditional CRF parameters to achieve a consistent performance improvement on different applications, and our proposed method does not require tuning parameters. Moreover, the analysis shows the importance of each modality to each dataset, which can be automatically adapted in the proposed method but not in traditional methods. The model is trained using *XGBoost* [60] for 100 iterations using a learning rate of 0.5, 0.01, and 0.01 for CT Arteries, WMH, and ISLES respectively. Note that the SHAP analysis results can only be explained under the assumption of the current parameter search scales and XGBoost models.

The results are shown in Figure 2.11. The summary plot in the left sub-graph shows an overview of all parameter sets with the most important parameters on top of the list. For each dataset, the best and worst parameter settings are shown in the right sub-graph. For all datasets, the post-processing quality is affected most by the spatial parameters $\omega_2$ and $\theta_\gamma$, and less by the intensity parameters per modality $\theta_\beta$.

The results on the CT arteries data (Figure 2.11a *left*) are more stable (with smaller SHAP values) than the results for WMH and ISLES, indicating that the post-processing CRF can hardly change the CNN output of the artery segmentation (see Figure 6 in the paper as an example).

In the WMH dataset, looking at independent parameter contributions, low values for spatial parameters $\omega_2$, $\theta_\gamma$ (less smoothing), and a smaller number of iterations lead to an improved performance. This is not unexpected, because white matter lesions are sparsely distributed and spatial smoothing tends to remove small lesions. Too strong spatial correlations (either large weight $\omega_2$ or small $\theta_\gamma$) will remove true positives as well (see Figure 6 in the paper). The summary plot (Figure 2.11b *left*) shows, as expected, that the FLAIR image has a larger impact on the model than the T1

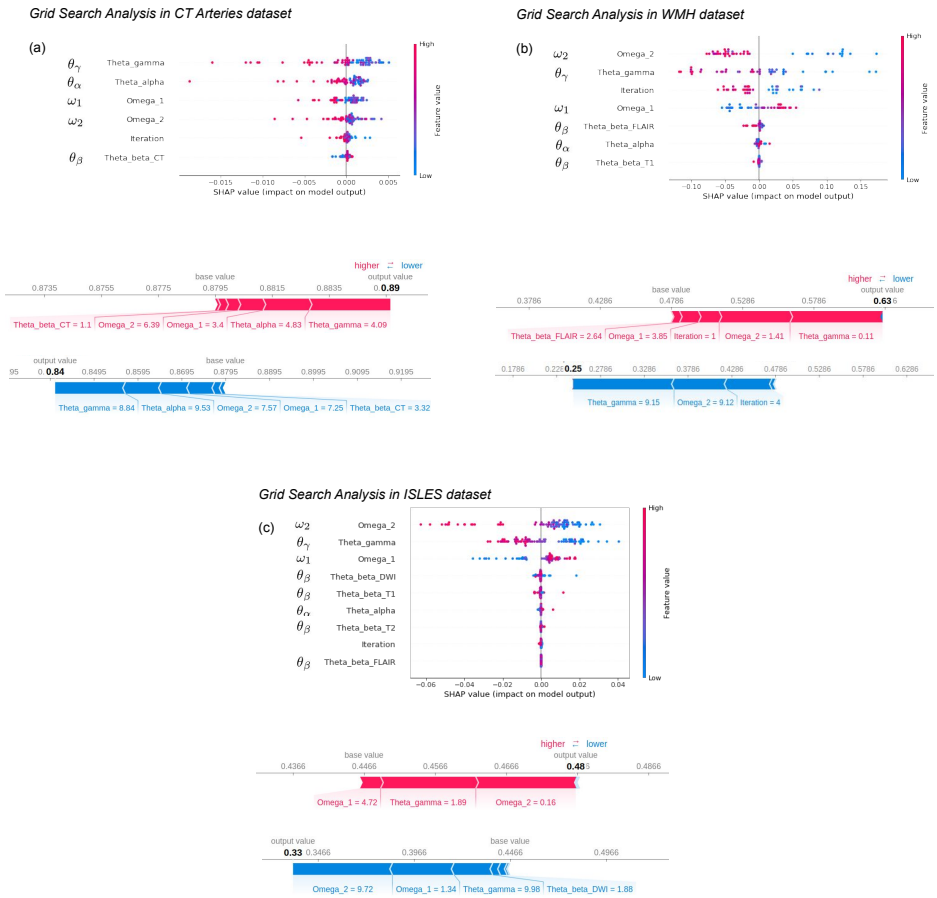



***Figure 2.11:*** ***SHAP analysis of the grid search results.*** *See Section 2.9.2 for an explanation. Upper sub-graphs: summary plots of all parameter sets evaluated during grid search. Positive SHAP values indicates a positive contribution to the performance and vice versa. The legend (feature value bar) shows the search range for each parameter. This reveals for example that lower values of* $\omega_2$ *lead to better segmentation performance for all datasets. Lower sub-graphs: the best (1st row) and worst (2nd row) parameter sets for each dataset. Red bar represents positive contribution to the performance and blue bar is negative contribution. Base value is the average DSC of all grid search results and output value is the DSC in the parameter set depicted. Best viewed in color with zoom.*

image. Table 1 also shows a smaller $\theta_\beta$ selected (corresponding to higher influence) for FLAIR.

Similar trends can be found for the ISLES dataset (Figure 2.11c). Spatial parameters $\omega_2$ and $\theta_\gamma$ are important to tune and high values can strongly harm the performance. The summary plot shows that the DWI image has a larger impact on the model than T1, T2, and FLAIR. In Table 1, $\theta_\beta$ for FLAIR and DWI are smaller than $\theta_\beta$ for T1 and T2, which means that FLAIR and DWI images are more informative for the segmentation of ischemic stroke lesions.

# Chapter 3

## Multi-task Attention-based Semi-supervised Learning for Medical Image Segmentation

3

**Abstract**

*We propose a novel semi-supervised learning (SSL) image segmentation method that simultaneously optimizes a segmentation and an auxiliary reconstruction objective. The auxiliary reconstruction task is guided by an attention mechanism that encourages the encoder to learn more discriminative features from unlabeled data. The proposed approach is evaluated on two applications: brain tumor segmentation and white matter hyperintensities segmentation. Our method, trained on unlabeled images and a small number of labeled images, outperforms a supervised CNN trained with only labeled images and a CNN with unsupervised pretraining on the unlabeled data. The proposed method even outperforms a supervised CNN trained using labels for all images. In ablation experiments, the proposed attention mechanism strongly improves segmentation performance over a similar network without the attention mechanism. We explore two multi-task training strategies: joint training and alternated training. Alternated training requires fewer hyperparameters and achieves a better, more stable performance than joint training. Finally, we analyze the features learned by the different methods and find that the attention mechanism helps to learn more discriminative features in deeper layers of the encoders.*

Based on: **S. Chen**, G. Bortsova, A. Garcia-Uceda Juarez, G. van Tulder, and M. de Bruijne, "Multi-task attention-based semi-supervised learning for medical image segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, 2019, pp. 457–465. DOI: 10.1007/978-3-030-32248-9_51

## 3.1 Introduction

Semi-supervised learning (SSL) uses unlabeled data to improve the generalization performance of a supervised model. This can be useful in medical image segmentation, where manual annotations can be expensive and tedious to produce and are often limited to only a small subset of the available training data.

One approach to semi-supervised learning is multi-task learning, which trains a network to solve an auxiliary task, learned from unlabeled data, in addition to the supervised task learned from labeled data. This approach has been used for image classification (e.g., [61, 62]) and image segmentation (e.g., [63]) with architectures that combine supervised classification with unsupervised reconstruction, for example by including an additional autoencoder objective.



Multi-task learning can be non-trivial to combine with popular image segmentation architectures like UNet[29] and its variants[30, 64], which use skip-connections to preserve high-resolution information in early layers of the network. However, these skip-connections make it difficult to add an autoencoder as the auxiliary task in the same segmentation network, because they would allow the network to skip the dimensionality reduction required by the autoencoder.

We propose a novel semi-supervised method called Multi-task Attention-based Semi-Supervised Learning (MASSL), in which we successfully combine an autoencoder with a UNet-like network. Instead of training it to reconstruct the original input, we train the autoencoder to reconstruct synthetic segmentation labels created by the introduced attention mechanism. This allows our model to learn the discriminative features for segmentation from unlabeled images. To our best knowledge, semi-supervised learning and attention have not been combined before.

Attention-based methods are often used to focus a network on relevant areas for supervised learning (e.g., [65]). Differently, our attention mechanism is designed to tackle unsupervised learning problems. Our method has some similarities with self-training and co-training, which also create new labels for the unlabeled training data on-the-fly. However, whereas self-training[66] and co-training[67, 68] create labels for the segmentation task, our method creates labels for the reconstruction task. This guides the unsupervised, auxiliary task to learn a better latent representation from unlabeled data that would be limited to the earlier layers in a traditional segmentation network.

Our contributions can be summarised as follows: firstly, we propose a new multi-task semi-supervised learning method and study its performance in combination with two training strategies. Secondly, we evaluate our method on two segmentation problems (brain tumors and white matter hyperintensities), on which it outperforms a fully supervised CNN baseline, two pre-training approaches, as well as multi-task learning without the proposed attention mechanism. Thirdly, we discuss how the attention mechanism affects the features learned by the encoder and show that it helps the deeper layers to learn more discriminative features.

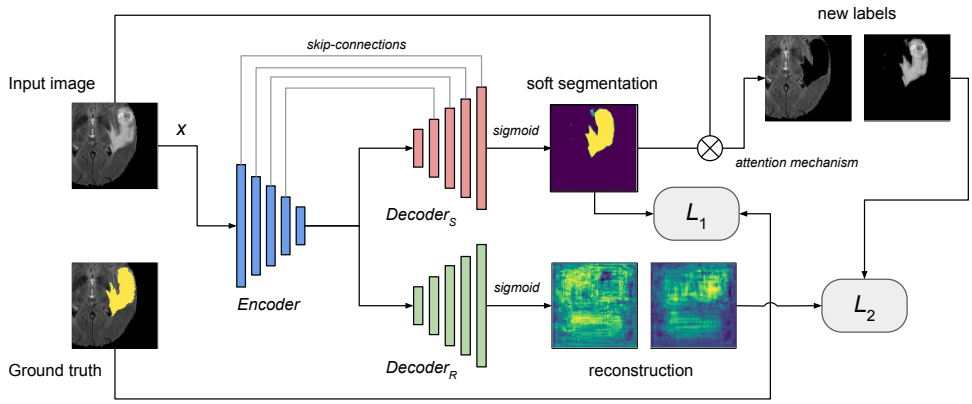



***Figure 3.1: Proposed MASSL framework.*** *Segmentation loss $L_1$ is calculated by the soft segmentation prediction and the ground truth. Reconstruction loss $L_2$ is calculated by the reconstructed foreground and background prediction and the new labels created through the attention mechanism.*

## 3.2 Methods

Our semi-supervised learning method (Fig. 3.1) is composed of a multi-task learning (MTL) framework and an attention mechanism for connecting the two tasks. The MTL framework has segmentation as its main task (Section 3.2.1) and reconstruction as its auxiliary task (Section 3.2.2). The attention mechanism is introduced in Section 3.2.3.

### 3.2.1 Segmentation Network

The forward path of our CNN-based segmentation network can be formulated as follows:

$$\tilde{y} = D_S[Encoder(x)], \quad L_1 = Dice(\tilde{y}, y) \tag{3.1}$$

where $x$ is the input image, *Encoder* represents the encoder of the network. $D_S$ represents the segmentation decoder with skip-connections that makes the segmentation prediction. The variables $\tilde{y}$ and $y$ are the predicted labels and segmentation ground truth respectively. The Dice similarity coefficient is used as segmentation loss function.

### 3.2.2 Reconstruction Network

As the auxiliary task, we use a reconstruction network with an autoencoder objective that is trained together with the segmentation network. The two networks share the same encoder parameters in order to learn useful features for segmentation and reconstruction simultaneously:

$$\hat{y} = D_R[Encoder(x)], \quad L_2 = MSE(\hat{y}, x) \tag{3.2}$$

where $D_R$ is the reconstruction decoder, without skip-connections, and $\hat{y}$ is the reconstructed image as predicted by the autoencoder. Mean squared error (MSE) is used as the reconstruction loss. We can reconstruct both labeled samples and unlabeled samples to make full use of all the available images. In the remainder of the paper, we refer to this approach combining segmentation and reconstruction networks as our Multi-task SSL (MSSL) method.

### 3.2.3 Attention Mechanism



We want to further connect the two tasks not only by the shared encoder in MSSL method. We introduce an attention mechanism to fuse both the segmentation task and the reconstruction task into the autoencoder. We use the soft segmentation predictions as attention maps to create new labels for the reconstruction task, in which foreground and background image regions are reconstructed separately. This encourages the encoder to learn more discriminative features through the reconstruction path. The new reconstruction loss is weighted by the size of the background and foreground to avoid paying more attention to small structures:

$$\tilde{y} = D_S[Encoder(x)], \quad \hat{y}_b, \hat{y}_f = D_R[Encoder(x)] \tag{3.3}$$

$$w_1 = \frac{\sum(\mathbf{1} - \tilde{y})}{\sum(\mathbf{1})}, \quad w_2 = \frac{\sum(\tilde{y})}{\sum(\mathbf{1})} \tag{3.4}$$

$$L_2 = w_1 MSE[\hat{y}_b, x(\mathbf{1} - \tilde{y})] + w_2 MSE[\hat{y}_f, x\tilde{y}] \tag{3.5}$$

where $\tilde{y}$ is the prediction by the foreground segmentation network. $\mathbf{1}$ is the tensor of ones that has the same size as $\tilde{y}$. $\hat{y}_b$ and $\hat{y}_f$ are the background and foreground reconstruction predictions. We refer to this method that combines the segmentation network, the reconstruction network and the attention mechanism as our Multi-task Attention-based SSL (MASSL) method.

### 3.2.4 Training Strategy

The two tasks of the MSSL and MASSL networks can be optimized jointly or alternatingly:

**Joint training:** Given a minibatch containing an equal number of labeled samples $x_L$ and unlabeled samples $x_U$, the unlabeled samples $x_U$ are first segmented using the most recent segmentation network parameters, to create the foreground and background images for the reconstruction task. Then, the weights of the entire network are updated by optimizing the objective function of both segmentation and reconstruction tasks. The loss is a linear combination of segmentation and reconstruction losses controlled by the hyperparameter $\gamma \in [0, 1]$:

$$L(x_L, x_U) = \gamma L_1(x_L) + (1 - \gamma) L_2(x_U) \tag{3.6}$$

**Alternated training:** For each epoch, labeled and unlabeled images are randomly sampled by the same amount (the smaller amount of either labeled and unlabeled images) from their corresponding training sets. A minibatch contains either labeled samples $x_L$ or the same amount of unlabeled samples $x_U$. The two types of batch are alternated during training. The weights of the segmentation path and reconstruction path are updated individually according to the given batch type and the corresponding loss. Then, no $\gamma$ is needed:



$$L(x) = \begin{cases} L_1(x), & \text{if } x = x_L \\ L_2(x), & \text{if } x = x_U \end{cases} \tag{3.7}$$

## 3.3 Experiments

### Data

We use the public data from the BraTS 2018 Challenge[33, 69] and the White Matter Hyperintensities 2017 Challenge[1]:

*BraTS18:* 220 MRI scans from patients with high grade glioma are randomly split into 120, 50, 50 scans for training, validation and testing respectively, with 5-fold cross-validation. To simplify comparison between the different segmentation tasks we perform binary classification and segment only the whole tumor, including all four tumor structures, and use only the FLAIR sequence.

*WMH17:* There are 60 FLAIR MRI scans provided with corresponding manual segmentations of white matter hyperintensities (WMH). The scans are acquired at three sites, 20 at each site. In our experiments, we use 30 scans for training, 10 for validation and 20 for testing, ensuring approximately equal numbers for each site in each of the three sets. We use 5-fold cross-validation.

### Network and hyperparameters

The network layout is shown in Figure 3.1. Our network is inspired by the UNet[29] architecture but has several differences. The input size of the network is $128 \times 128 \times 32$. There are 5 resolution levels in the encoder and in each of the decoders. Each level consists of two $3 \times 3 \times 3$ convolution layers using zero-padding, instance normalization[70] and *LeakyReLU* activation functions, except for the last layer of both decoders which use *sigmoid* to make the final prediction. There is an average pooling/upsampling layer between each level. The number of feature channels is 16 in the first level, which is doubled/halved after each pooling/upsampling to a maximum of 256 features at the deepest level. The feature maps in the segmentation upsampling path are concatenated with earlier ones through skip-connections. The reconstruction network has the same architecture as the segmentation network but with no skip-connections. For joint training, we use one *Adam* optimizer to optimise the loss in Eq. 3.6. For alternated training, we use two individual *Adam* optimizers to optimise the two types of loss in Eq. 3.7 separately. Based on the performance on the validation sets, we set the

[1] https://wmh.isi.uu.nl/

initial learning rate to 0.01 and 0.001 for the segmentation and reconstruction tasks respectively. Random rotation, scaling, and horizontal flipping are applied as data augmentation.

**Feature analysis**

We use linear regression analysis to evaluate how well the features can discriminate between foreground and background regions in the last layer of every encoder level. We consider each voxel as an individual sample, using its values in each feature map as the regression variables. The label for each voxel is obtained by the taking binary segmentation ground truth and then down-sampling this with average pooling to the required resolution.



## 3.4 Results

The segmentation results are shown in Table 3.1 and Table 3.2. For the semi-supervised setting (first two colomns), there is no overlap between labeled and unlabeled data. For the fully-supervised setting (last colomn), all the images are used as labeled and unlabeled data. For *Pretrain(Dec)* we pretrain the reconstruction network with unlabeled data first and then train the decoder path of segmentation network with labeled data, while keeping the encoder part fixed to ensure that the segmentation task can only use the features learned from unlabeled images. For *Pretrain(CNN)* we pretrain the reconstruction network with unlabeled data first and then train the whole segmentation network using labeled data, which allows the network to fine-tune the encoder parameters if necessary. *MASSL* and *MSSL* are the proposed multi-task SSL methods with and without the attention mechanism, where $\gamma$ and *alter* indicate joint training and alternated training respectively. For joint training, we tried $\gamma = 0.5, 0.7, 0.9$ and the network did not converge when $\gamma = 0.5$. The results show that MASSL(alter) achieves the best segmentation performance of all methods. The joint training strategy achieved a slightly lower performance than alternated training, which also varied a lot between different labeled/unlabeled data splits, reflecting the instability of the joint training strategy and the difficulty to tune $\gamma$.

The results of feature analysis are shown in Table 3.3. We can see that the features of the MASSL method are more discriminative in the deeper levels than those of CNN and MSSL. This supports our hypothesis that the attention mechanism could make the deeper layers of the encoder learn more discriminative features and still keep the property of the reconstruction autoencoder.

## 3.5 Discussion and Conclusion

In this paper, we propose a new semi-supervised learning method called MASSL that combines a segmentation task and a reconstruction task through an attention mechanism in a multi-task learning network. The proposed method is evaluated on two applications. For both applications, MASSL using part of the labeled images outperforms the fully supervised CNN baseline using the same number of labeled images, pretraining+finetuning methods, and the proposed approach without attention (MSSL).

***Table 3.1: BraTS18 results.*** *5-fold cross-validation. Dice similarity coefficient is reported. The last column uses all labeled images also as unlabeled images, except for CNN baseline which could only use labeled images. *: significantly better than CNN baseline (p<0.05). ◇: significantly worse than MASSL(alter) (p<0.05). P-values are calculated by two-sided t-test in each column.*

| #Labeled(unlabeled) | 20 (100) | 50 (70) | 120 (120) |
|---|---|---|---|
| CNN baseline | 0.6939(±0.03) | 0.7054(±0.03) | 0.7342(±0.02) |
| Pretrain(Dec) | 0.6948(±0.03)◇ | 0.6886(±0.03)◇ | 0.7162(±0.02)◇ |
| Pretrain(CNN) | 0.7125(±0.03)◇ | 0.7167(±0.03)◇ | 0.7530(±0.02) |
| MSSL($\gamma$=0.7) | 0.6140(±0.04)◇ | 0.7433(±0.02)*◇ | 0.7310(±0.02)◇ |
| MSSL($\gamma$=0.9) | 0.6297(±0.03)◇ | 0.7466(±0.02)*◇ | 0.7568(±0.02) |
| MSSL(alter) | 0.7261(±0.03)*◇ | 0.7462(±0.03)◇ | 0.7461(±0.02) |
| MASSL($\gamma$=0.7) | 0.6096(±0.03)◇ | 0.7412(±0.02)*◇ | 0.7589(±0.02) |
| MASSL($\gamma$=0.9) | 0.6168(±0.04)◇ | 0.7159(±0.03)◇ | 0.7660(±0.02)* |
| **MASSL(alter)** | **0.7553**(±0.03)* | **0.7710**(±0.02)* | **0.7702**(±0.02)* |

***Table 3.2: WMH17 results.*** *5-fold cross-validation. Dice similarity coefficient is reported. The last column uses all labeled images also as unlabeled images, except for CNN baseline which could only use labeled images. *: significantly better than CNN baseline (p<0.05). ◇: significantly worse than MASSL(alter) (p<0.05). P-values are calculated by two-sided t-test in each column.*

| #Labeled(unlabeled) | 10 (20) | 20 (10) | 30 (30) |
|---|---|---|---|
| CNN baseline | 0.6030(±0.05) | 0.6762(±0.02) | 0.6915(±0.02) |
| Pretrain(Dec) | 0.6088(±0.02)◇ | 0.6252(±0.03)◇ | 0.6439(±0.05)◇ |
| Pretrain(CNN) | 0.6615(±0.03)* | 0.6779(±0.02) | 0.6890(±0.02) |
| MSSL($\gamma$=0.7) | 0.5930(±0.04)◇ | 0.6326(±0.03)◇ | 0.6860(±0.02) |
| MSSL($\gamma$=0.9) | 0.6189(±0.03) | 0.6163(±0.03)◇ | 0.6906(±0.02) |
| MSSL(alter) | 0.6509(±0.03) | 0.6646(±0.03)◇ | 0.6880(±0.02) |
| MASSL($\gamma$=0.7) | 0.6074(±0.03)◇ | 0.6869(±0.03) | 0.6900(±0.02) |
| MASSL($\gamma$=0.9) | 0.6654(±0.03)* | 0.6925(±0.02) | 0.6806(±0.03) |
| **MASSL(alter)** | **0.6670**(±0.03)* | **0.7111**(±0.02) | **0.7204**(±0.02) |

When using the segmentation and reconstruction loss for all images, MASSL also improves over baseline CNN, although this difference was only statistically significant for the BRATS data. This is mainly due to the sparse distribution of foreground in WMH data, which makes our attention maps less effective.

The improvement of our method mainly comes from the new attention mechanism, which introduces the segmentation task into the reconstruction task and links them better than before. The mechanism can be easily integrated into any CNN architecture and generalized to multi-class segmentation. Compared to joint training, alternated training is a practical strategy that allows task-dependent variations in the learning

***Table 3.3: Discriminative power of the encoded features.*** *Using the trained models of all 5 folds on BRATS data, with 50 labeled/70 unlabeled data splits. 5 training/testing data are randomly chosen from the testing sets and used for all models because of the size limitation of the earlier feature maps. The experiment is repeated 5 times with different random data and the mean $R^2$ score (variance) between 5 experiments averaged over all 5-fold models is reported. Note, that results can only be compared within columns because the ground truth and dimensionality change between levels.*

| #Level | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| CNN baseline | 0.301(.01) | **0.527**(.01) | 0.496(.01) | 0.422(.01) | 0.486(.04) |
| MSSL(alter) | **0.344**(.02) | 0.515(.01) | **0.524**(.01) | 0.476(.02) | 0.471(.03) |
| MASSL(alter) | 0.340(.02) | 0.508(.01) | 0.501(.01) | **0.478**(.01) | **0.535**(.03) |

rate and does not require fine-tuning $\gamma$, although one still needs to choose proper initial learning rates. Alternated training is not guaranteed be stable because the encoder parameters change discontinuously between the two tasks. During experiments, we found the training was sufficiently stable when choosing a smaller initial learning rate for reconstruction than segmentation, and in most cases, the performance of the alternated optimization was – without exhaustive tuning of $\gamma$ – much better than that of joint optimization.

Since the aim of this paper was to compare several multi-task learning strategies, we made some simplifications. For the pretraining method, unlike Sedai et al.[63], we use a regular autoencoder rather than a variational autoencoder (VAE) in this paper. We think our SSL method could also work well with VAE and perhaps fuse the two tasks even better. In the regression analysis we use a simple regression model that could only show the linear discriminative power of the features. It would be interesting to use a more complicated non-linear model to show the non-linear discriminative power, too. Since we use only one MRI sequence and a subset of scans, our performance on BraTS18 and WMH17 are lower than the state of the art. The best Dice performances of BraTS18 (whole tumor) and WMH17 on testing sets are 0.8839 [71] and 0.80 [72] respectively, and first work also uses variational autoencoder to provide more regularization effect similar to the Ladder network [62] and our MSSL method.

In conclusion, MASSL is a promising segmentation framework for simple and efficient multi-task learning that can achieve strong improvements in semi-supervised as well as in fully supervised settings.

## 3.6 Acknowledgements

This research is supported by the China Scholarship Council (File No.201706170040). We gratefully acknowledge the support of the computational resources provided by SURFsara services and Cartesius.

# Chapter 4

## Region-of-interest guided Supervoxel Inpainting for Self-supervision

4

**Abstract**

*Self-supervised learning has proven to be invaluable in making best use of all of the available data in biomedical image segmentation. One particularly simple and effective mechanism to achieve self-supervision is inpainting, the task of predicting arbitrary missing areas based on the rest of an image. In this work, we focus on image inpainting as the self-supervised proxy task, and propose two novel structural changes to further enhance the performance. Our method can be regarded as an efficient addition to self-supervision, where we guide the process of generating images to inpaint by using supervoxel-based masking instead of random masking, and also by focusing on the area to be segmented in the primary task, which we term as the region-of-interest. We postulate that these additions force the network to learn semantics that are more attuned to the primary task, and test our hypotheses on two applications: brain tumour and white matter hyperintensities segmentation. We empirically show that our proposed approach consistently outperforms both supervised CNNs, without any self-supervision, and conventional inpainting-based self-supervision methods on both large and small training set sizes.*

Based on: S. Kayal, **S. Chen**, and M. de Bruijne, "Region-of-interest guided supervoxel inpainting for self-supervision," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, 2020, pp. 500–509. DOI: 10.1007/978-3-030-59710-8_49

## 4.1 Introduction and Motivation

*Self-supervised learning* points to methods in which neural networks are explicitly trained on large volumes of data, whose labels can be determined automatically and inexpensively, to reduce the need for manually labeled data. Many ways of performing self-supervision exist, amongst which a popular way is the *pre-train and fine-tune* paradigm where: (1) a convolutional neural network is pre-trained on a proxy task for which labels can be generated easily, and (2) it is then fine-tuned on the main task using labeled data. Utilizing a suitable and complex proxy task, self-supervision teaches the network robust and transferable visual features, which alleviates overfitting problems and aides its performance when fine-tuned on the main task [73].

In the medical imaging domain a variety of proxy tasks have been proposed, such as sorting 2D slices derived from 3D volumetric scans [74], predicting 3D distance between patches sampled from an organ [75], masking patches or volumes within the image and learning to predict them [76], and shuffling 3D blocks within an image and letting a network predict their original positions [77]. Recently, state-of-the-art results were achieved on several biomedical benchmark datasets by networks which were self-supervised using a sequence of individual proxy tasks [78].



Prior works in self-supervision literature have designed the proxy task largely uninfluenced by the downstream task in focus. However, since the features that the network learns are dependent on where it is focusing on during the self-supervision task, it might be beneficial to bias or *guide* the proxy task towards areas that are of interest to the main task. Specifically for image segmentation, these would be the foreground areas to be segmented in the main task, which we term as the *region-of-interests or ROIs*.

We experiment with the proxy task of inpainting [79], where the network must learn to fill-in artificially created gaps in images. In the context of biomedical imaging, a network that learns to inpaint healthy tissue will learn a different set of semantics than one which inpaints various kinds of tumours. Thus, if the main task is that of segmenting tumours, it can be hypothesized that having a network inpaint tumourous areas as a proxy task will likely teach it semantics attuned to segmenting tumours, and thereby be more beneficial for the main task than learning general semantics. In other words, by increasing the frequency of inpainting tumours, we can teach the network features which are more related to the tumour segmentation task.

Furthermore, in prior work the selection of regions to mask has largely been uninformed and random. We try to improve upon this situation by selecting regions which are homogeneous. Masking such regions could force the network to learn more about the anatomical meaning and relation to other structures of the masked tissue. For example, masking small regions in a lung CT scan would only require the model to correctly interpolate the structures (airways, vessels) around the masked region. In contrast, when a full airway or vessel branch is masked, inpainting requires understanding of the relation between branches in vessel or airway trees and/or the relation between airways and arteries, a piece of information that has been found to improve airway segmentation [80].

The contributions of this work are twofold. Firstly, this paper demonstrates that guiding the inpainting process with the main class(es) of interest (,i.e., the segmentation

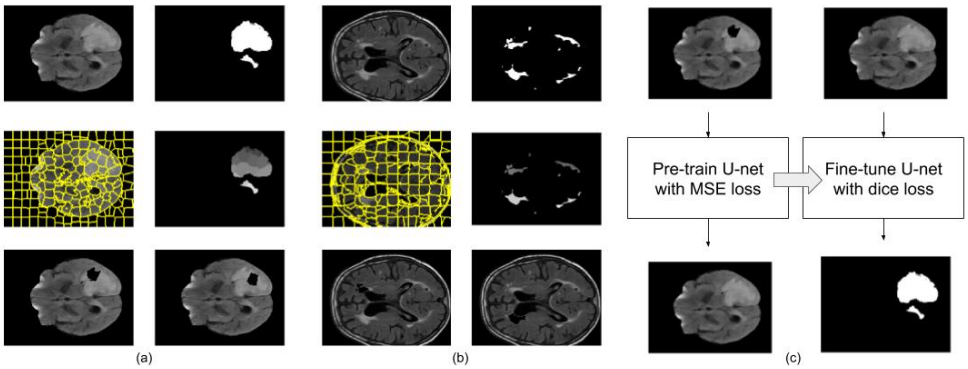



***Figure 4.1: Proposed ROI-guided inpainting.*** *(a) Examples from BraTS 2018 dataset (left to right from top to bottom): original FLAIR image-slice, ground-truth segmentation map, FLAIR image-slice with superpixels overlaid, region-of-interest (ROI) influenced superpixels, examples of synthesized images to be inpainted. (b) Examples from White Matter Hyperintensities 2017 dataset. Notice that the ground-truth segmentations are much smaller in size. (c) first a U-net is pre-trained on the inpainting task with MSE loss, next it is fine-tuned on the main segmentation task with Dice loss.*

foreground, interchangeably used with the term *ROI* in this paper) during the self-supervised pre-training of a network improves its performance over using random regions. Therefore, the proposed method can be thought of as an efficient addition to self-supervision when manual annotations are available. Secondly, we show that instead of inpainting regions of regular shapes in an uninformed way, further performance gain is possible if the masked regions are chosen to be homogeneous. This is done by constructing supervoxels and using these as candidate regions to be inpainted. In order to show the efficiency of these proposed changes, we conduct empirical analyses on two popularly used public datasets for biomedical image segmentation.

## 4.2 Methods

The proposed method (Figure 4.1) utilizes supervoxelization to create candidate regions, followed by selecting only those supervoxels which have an overlap with (any of) the foreground class(es). The selected supervoxels are utilized in the inpainting process, where we use them as masks to suppress areas in an image to train a network to predict (or *inpaint*) them based on their surroundings. Since we control the parameters of this process, it can be used to create an arbitrarily large amount of synthetic training images for pre-training.

### 4.2.1 Region-of-interest guided inpainting

Inpainting is an effective proxy task for self-supervision, which proceeds by training a network to reconstruct an image from a masked version of it. In this section, we explain our proposed masking approach, followed by the description of the network in Section 4.2.2.

**Supervoxelization:** While previous works primarily use random grids and cubes as candidate regions to inpaint, the first step in our proposed approach is to select regions based on some notion of homogeneity. One way of achieving this is to construct supervoxels, which may be defined as homogeneous groups of voxels that share some common characteristics. A particularly efficient algorithm to construct such supervoxels is *SLIC* or *simple linear iterative clustering* [81].

For 3D medical images, SLIC can cluster voxels based on their intensity values, corresponding to the various modalities, and spatial coordinates of the voxel within the image. SLIC has two main hyperparameters: one, *compactness*, controls the balance between emphasis on intensity values and spatial coordinates (larger values make square/cubic grids), and the other defines the maximum number of supervoxels. Examples in the second row of Figure 4.1, subfigure (a) and (b).

In this work, we use SLIC with intensity values corresponding to the two modalities we used in our experiments, FLAIR and T1 (or contrast enhanced T1), in order to construct supervoxels. The exact parameter settings for supervoxelization are described later in Section 4.3.3.

**ROI-guided masking for inpainting image synthesis:** Once the supervoxel labels have been created, the next step is to retain only those supervoxels which have an overlap with the region-of-interest. To achieve this, we first convert the segmentation map to a binary one by considering all foreground areas to be a class with a label value as 1 and the background as 0, since there may be multiple regions-of-interest in a multi-class segmentation setting. Then an elementwise *and* operation is performed between the resulting binary segmentation map and the generated supervoxel. For all the supervoxels that remain, training images for the inpainting task can be synthesized by masking an area corresponding to such a *ROI-guided supervoxel*, with the original unmasked image being the target for the network. Some examples of this are in the second row of Figure 4.1, subfigure (a) and (b).

By constructing a training set for the inpainting task in this fashion, we are essentially increasing the frequency of inpainting regions which are important to the main task more than random chance. This is what, we posit, will bring about improvements in the performance of the network on the main task.

Formally, let $D_{train} = \{(I_i, S_i)\}_{i=1..n}$ be the training dataset containing $n$ images with $I_i$ being a 3D multi-modal training image and $S_i$ being the segmentation ground-truth label, containing zero values representing background. If $f$ is a supervoxelization algorithm (in our case, SLIC), then a ROI-guided supervoxelized image is given by $R_i = f(I_i) \odot S_i$, where $\odot$ signifies elementwise multiplication.

$R_i$ contains supervoxel regions having non-zero labels corresponding to foregound supervoxels. Then, the synthetic dataset for inpainting, $D_{inp}$ is constructed as:

$$D_{inp} = \left\{ \left\{ \left( I_i \odot r^0_{ij}, I_i \right) \right\}_{r_{ij} \in R_i, j=1..m_i} \right\}_{i=1..n} \tag{4.1}$$

where $r_{ij}$ is a single supervoxel region in the set $R_i$, which contains a total of $m_i$ supervoxels, and $r^0_{ij}$ is the corresponding inverted region-mask, containing 0 for voxels belonging to the region and 1 everywhere else. $I_i \odot r_{ij}$ is then the masked image input to the network and $I_i$ is the expected output to reconstruct, the target for the inpainting task. Thus, the maximum cardinality of $D_{inp}$ can be $n \times m_i$.

Examples $D_{inp}$ are in the last row of Figure 4.1, subfigure (a) and (b).

### 4.2.2 Training Strategy

**Network:** For all the experiments, a shallow 3D U-net [82] containing 3 resolution levels has been used, with a batch-normalization layer after every convolution layer. In our experiments we find that 3 layers provide sufficient capacity for both the inpainting and the segmentation task. Since we use two modalities for our experiments, the U-net has two input channels.



If we were to use an image reconstruction proxy task, a U-net would learn to copy over the original image because of its skip connections, and would not be useful in learning features. In our task of inpainting the network never sees the masked regions and, therefore, cannot memorize it, making the use of a U-net reasonable.
**Pre-training:** In order to pre-train the network, it is fitted to the $D_{inp}$ dataset by minimizing the mean squared error (MSE) between the masked and the original images using the *Adam* [83] optimizer. We call this model *inpainter U-net*.
**Fine-tuning:** The inpainter U-net is then fine-tuned on the (main) segmentation task using the original labeled training dataset, $D_{train}$, by optimizing the Dice segmentation overlap objective on the labeled images. If the data is multi-modal, the inpainter U-net will be trained to produce multi-channel outputs, in which case we would need to replace the last 3D convolutional layer to have a single-channel output for segmentation.

More details about the network parameters are provided in section 4.3.3.

## 4.3 Experimental Settings

### 4.3.1 Data

For our experiments, we use two public datasets containing 3D MRI scans and corresponding manual segmentations.
**BraTS 2018** [84]: 210 MRI scans from patients with high-grade glioma are randomly split three times into 150, 30 and 30 scans for training, validation and testing, respectively, using a 3-fold Monte-carlo cross-validation scheme. To be able to easily compare our method against baselines, we focus on segmenting the whole tumour and use two of the four modalities, *FLAIR* and *T1-gd*, which have been found to be the most effective at this task [85].
**White Matter Hyperintensities (WMH) 2017** [86]: The total size of the dataset is 60 FLAIR and T1 scans, coming from 3 different sites, with corresponding manual segmentations of white matter hyperintensities. We employ a 3-fold Monte-carlo cross-validation scheme again, splitting the dataset into 40, 10 and 10 for training,

validation and testing, respectively, and use both of the available modalities for our experiments.

To study the effect of training set sizes on the proposed approach, experiments were performed on the full training dataset as well as smaller fractions of it. For BraTS, we perform experiments on 25%, 50% and 100% of the training data, while for WMH, which is much smaller in size, we only perform an extra set of experiments with 50% of the data. To keep the comparisons fair, we use the same subset of the training data in the pre-training procedure as well. Note that, even though self-supervision by inpainting (with or without supervoxels) could be applied to unlabeled data as well, in our experiments we only use fully labeled training samples to facilitate comparison.

### 4.3.2 Baseline Methods



We term the technique proposed in this paper as *roi-supervoxel* to denote the use of the segmentation map and supervoxelization to guide the inpainting process used for pre-training. In order to validate its effectiveness, it is tested against the following baselines: *vanilla-unet*: a U-net without any pre-training; *restart-unet*: a U-net pre-trained on the main (segmentation) task and fine-tuned on the same task for an additional set of epochs; *noroi-grid*: the more traditional inpainting mechanism where random regular sized cuboids are masked; *roi-grid*: a similar process as *roi-supervoxel*, except for the use of regular cuboids overlapping with the segmentation map, instead of supervoxel regions, for masking; *noroi-supervoxel*: where random supervoxels are masked.

### 4.3.3 Settings

**Inpainting Parameters**: The inpainting process starts by creating the supervoxel regions using SLIC[1]. We fix these the compactness value at 0.15 and choose the maximum number of supervoxels to be 400, by visual inspection of the nature of the supervoxels that contain the tumour and the white matter hyperintensities for the two datasets. For example, between a setting where one supervoxel is part tumour and part background, versus another where one supervoxel fully represents tumour, we choose the latter case.

We then use either the supervoxels or simple cuboids (for the baseline methods) as areas to be masked, and the question arises of how many and how large areas to choose as masks to construct synthetic images for $D_{inp}$. Too small a volume, and it might be trivial for a network to inpaint it; too large, and it might not be a feasible task. For our experiments, we choose masks whose volume is at least 1500 voxels. For constructing cuboids, we randomly generate cuboids which are at least 12 units in each dimension (as $12^3$ is more than 1500, but $11^3$ is not). Finally, we ensure that the size of $D_{inp}$ is roughly 10 times that of the $D_{train}$, by choosing masks which fit the volume criteria as they are generated, and producing at most 10 synthetic images on-the-fly for a single real input image.

[1]We use the implementation in https://scikit-image.org/docs/dev/api/skimage.segmentation.html?highlight=slic#skimage.segmentation.slic

**Network Parameters**: The input size to the 3D U-net is 160 × 216 × 32, such that each input image is centre-cropped to 160 × 216 ($X - Y$ axes) to tightly fit the brain region in the scan, while we use the overlapping tile strategy in the Z-axis as inspired by the original U-net. Each of the 3 resolution levels consists of two 3 × 3 × 3 convolution layers using zero-padding and *ReLU* activation, except for the last layer which is linear in the inpainter U-net and *sigmoid* in the fine-tuned U-net. The number of feature channels are 16, 32 and 64 at the varying resolution levels. The feature maps in the upsampling path are concatenated with earlier ones through skip-connections.
**Optimization Parameters**: The inpainter U-net is optimized on MSE while fine-tuning is performed using a Dice objective, both using *Adam*. The learning rate is 0.0001 and 0.001 for BraTS and WMH datasets, respectively. We used a batch-size of 4, as permitted by our GPU memory. For pre-training, we use 100 epochs while for fine-tuning we employ another 150, both without the possibility of early stopping, saving the best performing model based on the validation loss at every epoch.

To foster open-science, all of the code will be released[2].

## 4.4 Results and Discussion

The segmentation results are shown in Table 4.1. It can be observed that the proposed method (*roi-supervoxel*) outperforms the basic U-net (*vanilla-unet*) by a large margin, and traditional inpainting based pre-training (*noroi-grid*) by a small, but significant, margin.

The deductions from the empirical results can be summarized as follows:

**Restarts improve U-net performance:** It can be observed that for both datasets, the performance of the *restart-unet* is better than that of the *vanilla-unet*. This is in line with observations in literature [87], where warm restarts have aided networks to find a more stable local minimum. Based on this observation, we argue that for any proposed method involving pre-training models, the results should always be compared to such a *restarted* model.

**Adding ROI information to the inpainting proxy task is helpful:** For both the datasets, the performance of the *roi-supervoxel* method exceeds that of all other baselines. Importantly, it exceeds the performance of the *restart-unet* and the *noroi-grid*, which is the traditional inpainting procedure, by 3.2% and 5% (relative) respectively for BraTS, and 4.9% and 2.9% for WMH, when all of the data is used. Also important to note is that the performance of methods which use the region-of-interest information to generate the masked areas is always better than those which do not.

**Inpainting is more beneficial when the size of the training set is smaller:** The difference in performance between the inpainting-assisted methods and *vanilla-unet* is larger when the size of the training dataset is smaller. For example, for BraTS, the difference between *vanilla-unet* and *roi-supervoxel* (our proposed approach) is as large as 41.2% (relative) when the size of the training dataset is 25% of the total. This trend is also observed between the methods with and without ROI information.

[2]https://github.com/DeepK/inpainting

***Table 4.1: Dice-scores on BraTS 2018 and WMH 2017.*** *The results represent the mean and standard deviation (in brackets) of the Dice coefficient averaged over the three folds. The top results are depicted in* ***bold****.* $^*$*: indicates that a result is significantly worse than* roi-supervoxel *($p < 0.05$) in the same column, p-values calculated by a two-sided t-test.*

| Fraction Training Data | | | | | |
|---|---|---|---|---|---|
| | **BraTS** | | | **WMH** | |
| **Method** | **.25** | **.50** | **1.0** | **.50** | **1.0** |
| **vanilla-unet** | 0.257 (.05)* | 0.585 (.03)* | 0.784 (.02) | 0.576 (.05)* | 0.745 (0.02)* |
| **restart-unet** | 0.302 (.05)* | 0.607 (.03)* | 0.793 (.02) | 0.610 (.05)* | 0.776 (.03)* |
| **noroi-grid** | 0.311 (.06)* | 0.611 (.04)* | 0.780 (.03)* | 0.632 (.04)* | 0.791 (.03) |
| **roi-grid** | 0.354 (.06) | 0.620 (.04) | 0.795 (.03) | 0.653 (.04) | 0.812 (.03) |
| **noroi-supervoxel** | 0.340 (.06) | 0.621 (.04) | 0.791 (.02) | 0.650 (.04) | 0.797 (.03) |
| **roi-supervoxel** | **0.363 (.06)** | **0.646 (.04)** | **0.814 (.03)** | **0.671 (.04)** | **0.814 (.03)** |

4

**Supervoxels help more when areas to be segmented are larger rather than finer:** ROI-guided inpainting can be postulated to have a better chance of affecting the downstream performance when the ROI itself is larger. Taking into account that tumours in BraTS are, on-average, larger than the hyperintensities to be segmented in the WMH dataset, it can be observed that the performance difference between the inpainting methods using supervoxels (*roi-supervoxel* and *noroi-supervoxel*) versus the ones which do not (*roi-grid* and *noroi-grid*) is smaller in the case of WMH than for BraTS. For example, when using all of the training data, the difference in performance between *roi-supervoxel* and *roi-grid* is 3% (relative) for BraTS but only 0.25% for WMH. This could likely be alleviated by problem specific selection of parameters for SLIC, which we did not explore. This would ensure that the supervoxels are not too large as compared to the ROI, in which case the effect of ROI would not be significant.

These results show that our approach is promising. An important point to note is that a similar approach may be valuable in other forms of local self-supervision techniques like jigsaw puzzle solving [77], where the shuffling could be guided by the ROI and the tiles could be picked by ensuring homogeneity constraints.

Although efficient, this method does have some limitations: firstly, its efficiency depends on the parameters of the supervoxelization process and a poor choice of parameters could lead to limited performance gain; secondly, although sizeable synthetic datasets can be created in this process, the reliance on ROI means that we still need segmentation annotations. Perhaps one way of solving the second problem would be using co-training [88] to label all of the data and then employ our method using the entire corpus.

## 4.5 Conclusion

In summary, this work explores the use of supervoxels and foreground segmentation labels, termed the *region-of-interest (ROI)*, to guide the proxy task of inpainting for self-supervision. Together, these two simple changes have been found to add a significant

boost in the performance of a convolutional neural network for segmentation (as much as a relative gain of 5% on the BraTS 2018 dataset), in comparison to traditional methods of inpainting-based self-supervision.

## 4.6 Acknowledgements

This research was partly funded by the Netherlands Organisation for Scientific Research (NWO), as well as by the China Scholarship Council (File No.201706170040).

# Chapter 5

## Source Identification: A Self-Supervision Task for Dense Prediction

5


## Abstract

*Self-supervised learning is a research direction that focuses on representation learning from raw data without the need for labor-consuming annotations, which is the main bottleneck of current data-driven methods. Self-supervision tasks are often used to pretrain a neural network with a large amount of unlabeled data and extract generic features of the dataset. The learned model is likely to contain useful information which can be transferred to the downstream main task and improve performance compared to random parameter initialization. In this paper, we propose a new self-supervision task called source identification (SI), which is inspired by the classic blind source separation problem. Synthetic images are generated by fusing multiple source images and the network's task is to reconstruct the original images, given the fused images. A proper understanding of the image content is required to successfully solve the task. We validate our method on two medical image segmentation tasks: brain tumor segmentation and white matter hyperintensities segmentation. The results show that the proposed SI task outperforms traditional self-supervision tasks for dense predictions including inpainting, pixel shuffling, intensity shift, and super-resolution. Among variations of the SI task fusing images of different types, fusing images from different patients performs best.*


Based on: **S. Chen***, S. Kayal*, and M. de Bruijne, "Source identification: A self-supervision task for dense prediction," *Submitted*

## 5.1 Introduction

The success of deep learning, and in particular convolutional neural networks (CNNs), may be partially attributed to the exponential increase in the amount of available annotated data. However, in highly specialized domains such as medical image segmentation, it is much harder to acquire precise and dense annotations. Self-supervision is one research direction that enables the network to learn from images themselves without requiring labor-consuming annotations, where the learned features might be useful for the downstream tasks, such as classification and segmentation.

In general, self-supervised learning refers to a collection of approaches that deliberately withhold information in the original data and task a neural network to predict the missing information from the existing incomplete information. In doing so, the network is encouraged to learn general-purpose features which have been found to transfer well to downstream tasks [89]. The self-supervision pipeline often employs a pre-train and fine-tune strategy. The first step is to pre-train a CNN on a large volume of unannotated samples using a manually designed proxy task, in which the CNN explores and learns generic features of the data itself. The learned features may contain meaningful information of the image data, e.g., intensity distribution, spatial coherence, and anatomical knowledge in medical imaging, etc., depending on how the proxy task is designed. The second step is to fine-tune this pre-trained network on the target (main) downstream task that we are more interested in, which usually has a small set of annotated data in practice. We expect that by exploiting unannotated data and restarting the training from a set of rich pre-trained features, a more robust model on the main task can be trained.



In this paper, we propose a novel self-supervision task called *Source Identification* (SI), which is inspired by the classic blind source separation (BSS) problem. The proposed task is able to train a dense prediction network in a self-supervised manner using unlabeled data.

**Contributions:**

1. We propose a novel self-supervision task, SI, wherein a neural network is (pre-)trained to identify one image (source) from mixtures of images. This way, both encoder and decoder are trained and the network is encouraged to learn not only local features but also global semantic features to identify and separate the target source signal. To the best of our knowledge, this is the first BSS-like self-supervised method for deep neural networks.

2. We investigate the task ambiguity in the source identification problem and show in which settings it can be solved by a neural network. The proposed SI method provides a straightforward way to avoid ambiguity.

3. We conduct extensive experiments on public datasets for two medical image segmentation applications: brain tumor segmentation and white matter hyperintensities segmentation, both from brain MRI. We compare with various existing self-supervision tasks. The results show that the proposed SI method outperforms self-supervision baselines including inpainting, pixel shuffling, intensity shift, and super-resolution in segmentation accuracy in both applications.

## 5.2 Related Work

### 5.2.1 Self-Supervision Tasks

Self-supervision is an active research direction in machine learning, permeating from computer vision to natural language processing [90, 91, 92]. In imaging, early self-supervision tasks could be grouped into two main categories: reconstruction based and context prediction based. For example, inpainting is a popular reconstruction based self-supervision task [79] where areas in an image are hidden and then reconstructed using a CNN. In a similar fashion, recolorization can be done by removing color of the image and training a CNN to recover it [93], and super-resolution by recovering the original resolution of an image from a downsampled image [94]. On the other hand, context prediction based tasks make the network learn relationships between parts of an image, such as choosing arbitary tiles in an image and predicting their relative spatial locations [95]. An improved version of this method can be seen in [96], where tiles were chosen, shuffled and the network was taught to identify the shuffle pattern, thereby forcing it to learn how the tiles make up the original image. Self-supervision has also been applied in medical imaging [97], including inpainting [98, 99] and puzzle solving by treating a 3D image as a shuffled Rubik's cube [77].



All the above self-supervision tasks are designed to learn useful features from a single input image by recovering information withhold from the image itself. However, the rich information that discriminates one image from another is not explicitly considered. The proposed source identification task in this paper aims to learn not only features that can identify each image but also features that can distinguish one image from different images within the dataset.

The proposed source identification task shares some similarities with the contemporary contrastive learning method [100], which is also gaining popularity in medical imaging [101, 102, 103, 104, 105]. In contrastive learning, the neural network is tasked with recognizing the similarity or dissimilarity of a pair of images input to it, which can be categorized as a context prediction-based rather than reconstruction-based method. As an example, the state-of-art method known as *SimCLR* [106] works by drawing random samples from the original dataset, applying two augmentations (both sampled from the same family of augmentations) on the samples to create two sets of views. Then these views are passed through a CNN and a fully connected neural network layer to generate latent representations. Finally, these representations are used to train the network, such that the augmented views from the same class are pushed together and the augmented views from different classes are repelled using a contrastive loss. This may encourage the latent features to be more compact and separated, which may provide additional regularization for optimizing the network. However, most contrastive learning approaches are aimed at the downstream task of classification, pretraining only the encoder portion of the network. Thus, in this paper, we focus on the comparison between reconstruction-based methods that are more relevant to the proposed source identification task, as they pre-train the entire network and focus on dense prediction downstream tasks.

### 5.2.2 Blind Source Separation

Blind source separation (BSS), also known as signal separation, is the classic problem of identifying a set of source signals from an observed mixed signal. One example of BSS is the cocktail party problem, where a number of people are talking simultaneously in a noisy environment (a cocktail party) and a listener is trying to identify and separate a certain individual source of voice from the discussion. The human brain can handle this sort of auditory source separation problem very well, but it is a non-trivial problem in digital signal processing. Traditional methods such as independent component analysis (ICA) variants are proposed to tackle the BSS problem [25, 107, 108, 109, 110]. In the deep learning era, convolutional neural networks have been used to solve BSS problems in signal processing applications such as speech recognition [111, 112] and target instrument separation [113]. These works typically employ an encoder network to learn the embeddings of the observed signals and then use traditional techniques like k-means or spectral clustering to cluster the embeddings according to the number of sources. The clustering can also be done by a deep neural network [114]. This paper introduces a BSS-like self-supervised task on image data, in which a neural network is trained that aims to identify and restore the source image content in mixtures with multiple images.



### 5.2.3 Relation to Denoising

A related task to the proposed source identification is denoising [115] which is used to identify and remove undesired imaging artifacts. In denoising, the image and the noise are regarded as two different sources and a model is trained to separate them. The statistical properties of the signal and the noise are very different, unlike in our case, where a mixed image is constructed from images belonging to the same dataset. A denoising network is likely to learn more local features to distinguish noise from clean images rather than high-level semantic features of the image content. Different from the denoising task, the proposed source identification approach tries to separate one image from a fused image with other images rather than with noise. This is a more difficult task that is more likely to capture useful semantic features from the dataset.

### 5.2.4 Relation to Mixup

Mixup was first proposed as a data augmentation strategy while training CNNs in a general setting [116], and has been validated to work well in medical image segmentation as well [117]. Mixup, in a segmentation setting, works by randomly selecting an image pair from the training data and generating a weighted combination of the input images as well as the target segmentation maps. These generated images are then fed to a CNN during training, in addition to any other data augmentation strategies that may be suitable.

The similarity of our work with Mixup is in the way our mixed images are made, which, in our case, the network learns to identify sources from. However, our approach is a self-supervision strategy, with the aim of teaching the network useful features during pre-training, whereas Mixup is a data augmentation method. Nevertheless,

in order to compare the two, we also include a set of experiments with Mixup as an additional data augmentation strategy.

## 5.3 Methods

In Section 5.3.1, we provide a general definition of source identification. In Section 5.3.2, we discuss whether and when the source identification task is solvable for a neural network. In Section 5.3.3, we describe how source identification can be used as a proxy task for a self-supervised network. Lastly, we describe four popular competing baseline self-supervision tasks that we compare to in this paper in Section 5.3.4.

### 5.3.1 Definition of The Source Identification Problem

Consider domain $D$, in which each source signal can be distinguished from others, e.g., each signal is an image from a different patient in a medical imaging dataset. Multiple source signals $\mathbf{s}(t) = (s_1(t)...s_n(t))$ sampled from $D$ are linearly 'mixed' to produce $m$ mixtures $\mathbf{x}(t) = (x_1(t)...x_m(t))$ using an $m \times n$ matrix $W$:



$$\mathbf{x}(t) = W\mathbf{s}(t) \tag{5.1}$$

The blind source separation (BSS) problem is to reconstruct individual signals that constitute the mixtures without knowing the transformation $W$ and the original signals $\mathbf{s}$. For simplicity, we refer to a collection of source signals corresponding to an SI problem as $\mathbf{s}$ and mixtures as $\mathbf{x}$, and denote individual source signals as $s_i$ and individual mixtures as $x_j$ in the rest of the paper.

As an example, given two different randomly sampled signals $s_1$ and $s_2$, a signal mixture $x$ can be created by a linear combination and used as an input sample to the model $f(\cdot)$:

$$x = ws_1 + (1-w)s_2, \quad w \in (0,1) \tag{5.2}$$

where the weight $w$ and the original source signals $s_1$ and $s_2$ are unknown to the model $f(\cdot)$. We can train the model to solve SI problems by minimizing the loss $\mathcal{L}(\theta)$:

$$\begin{aligned}\mathcal{L}(\theta) &= \frac{1}{MN}\sum_{j\in 1...M}\sum_{i\in 1...N}\ell(s_i^{(j)}, f_i(\mathbf{x}^{(j)};\theta)) \\ &= \frac{1}{MN}\sum_{j=1}^{M}\sum_{i=1}^{N}\left(|s_i^{(j)} - f_i(\mathbf{x}^{(j)};\theta)| + (s_i^{(j)} - f_i(\mathbf{x}^{(j)};\theta))^2\right)\end{aligned} \tag{5.3}$$

where $M$ is the total number of generated mixtures $\mathbf{x}^{(j)}$ and the corresponding ground truth set $(s_i^{(j)}, i \in 1...N)$. $M$ can be infinite when $W$ is sampled from a continuous distribution; $N$ is the number of source signals we wish the network to reconstruct within the original source set $\mathbf{s}^{(j)}$; and $f_i(\cdot)$ is the corresponding output channel of model $f(\cdot)$ for reconstructing each source signal $s_i$. The function $\ell(\cdot,\cdot)$ is composed of the $L_1$ and $L_2$ norm of the difference between the original source signal $s_i$ and

the corresponding model output $f_i(\cdot)$. In practice, we may not be interested in reconstructing all sources but some of them, which allows $N$ to be smaller than $n$.

Note that the learning problem is ill-posed, since any possible permutation of $s_i$ would be a correct solution of the problem. In the next section, we show how the ambiguous order of $\mathbf{s}$ would influence the solvability of the task.

### 5.3.2 Is Source Identification Solvable?

The source identification problem as defined in the previous section is ill-posed due to the order ambiguity of sources $s_i$. That means, if sources are sampled from the same distribution, we cannot separate them with the learning objective in Equation 5.3.

This is illustrated in the following small experiment. We are given a medical imaging dataset $D_I$ and two randomly sampled source signals (images) $s_1$ and $s_2$ for every training iteration. Here we set $s_1$ as the target source signals and $s_1$ is expected to be reconstructed in the output channel $f_1(\cdot)$ of a neural network $f(\cdot)$. Note that $s_1$ and $s_2$ are exchangeable as ground truth for $f_1(\cdot)$ which introduces ambiguity and would make the reconstruction unsolvable. One way to make the reconstruction task less ambiguous would be to sample signals $s_2$ from a different domain than $s_1$, for instance by adding noise to $s_2$ as follows:

$$s_2' = (1-\lambda)s_2 + \lambda s_N, \quad \lambda \in [0,1] \tag{5.4}$$

where $s_{Nk} \sim \mathcal{N}(0,1)$ for the $k$th voxel in $s_N$. When $\lambda = 0$, $s_2' = s_2$, and the generated mixture $x$ becomes the same as that in Equation 5.2 in the previous section; when $\lambda = 1$, $s_2'$ is purely a Gaussian noise which belongs to an obviously different domain compared to the imaging domain in dataset $D_I$. Therefore, the new mixture $\tilde{x}$ can be created and the loss is shown as follows (with, $m = 1$ and $n = 2$ for generating each mixture $\tilde{x}$, and reconstructing the first source $s_1$ ($N = 1$) in Equation 5.3):

$$\begin{aligned} \tilde{x} &= ws_1 + (1-w)s_2', \quad w \in (0,1) \\ \mathcal{L} &= \frac{1}{M} \sum_{j \in 1 \ldots M} \ell(s_1^{(j)}, f_1(\tilde{x}^{(j)})) \end{aligned} \tag{5.5}$$

where $f_1(\cdot)$ is the output channel of $f(\cdot)$.

The results of a neural network (we use 2D U-net here) optimized to minimize the reconstruction loss of $s_1, s_2'$ with various value of $\lambda$ are visualized in Figure 5.1. It can be observed that when $\lambda$ is small (0.1), the output is an average of the two images $s_1$ and $s_2$ and the model failed to separate $s_1$ from their mix $\tilde{x}$. When $\lambda$ gradually increases (to 0.9), $s_1$ becomes clearer and better separated.

As this experiment illustrates, the network can not separate sources when they are sampled from the same distribution and mixtures are made arbitrarily, without any hint on the order of sources. To make the order of $s$ unambiguous, one simple way is to sample sources $s$ from different domains, for instance an MRI scan and Gaussian noise. However, the case $\lambda = 1$ is similar to a self-supervised denoising task where the model may focus on learning the differences between image domain and noise domain. These learned features may contain trivial local patterns and may be less likely to provide useful semantic features for downstream tasks like segmentation. We compare

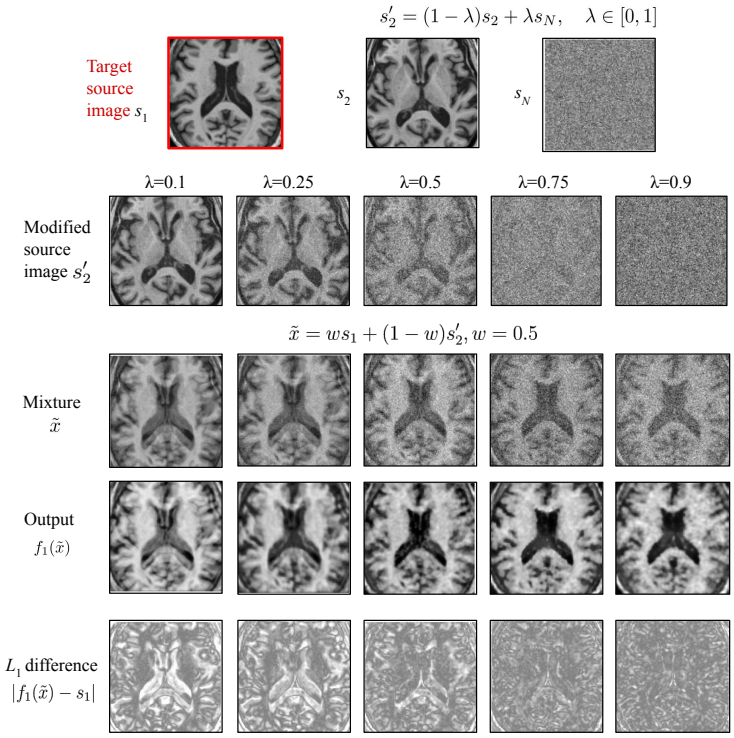



***Figure 5.1: Qualitative results of recovering $s_1$ with various $\lambda$.*** *We can see that the model is able to separate and reconstruct $s_1$ from $\tilde{x}$ gradually when $\lambda$ increases from 0.1 to 0.9. The dataset in this experiment contains 30 brain MRI scans from 30 different patients. Best viewed with zoom.*

this denoising SI variant (DSI) in the experiments and the results are shown in Section 5.5.

### 5.3.3 Proposed Source Identification Task

In this paper, we propose a simple variation of the source identification task that solves the issue of ambiguity of the source order. In this task, we sample $W$ so that one of the sources $s_i$ is present in every input mixture and make $s_i$ the only target output. This assumes the number of input mixtures $m$ is set to two or larger. In the case of $m = 2$, $n = 3$, and $N = 1$, the proposed task would be to identify and separate

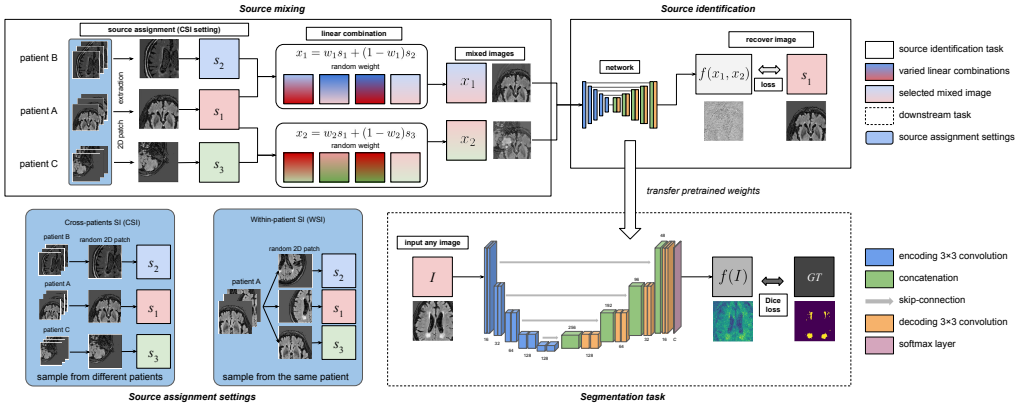



***Figure 5.2: The proposed source identification task.*** *Three source images $s_1$, $s_2$, and $s_3$ are used for this illustration. Cross-patients SI (CSI) and within-patient SI (WSI) are two different strategies to extract source signals, which focus on learning features between different patients and within one individual patient respectively. $2 \times 2$ downsampling and upsampling are applied between different resolutions in the UNet. Best viewed in color with zoom.*

the target signal, e.g., $s_1$, from two mixtures $x_1$ and $x_2$:

$$\begin{aligned} x_1 &= w_1 s_1 + (1 - w_1) s_2, \quad w_1 \in (0, 1) \\ x_2 &= w_2 s_1 + (1 - w_2) s_3, \quad w_2 \in (0, 1) \\ \mathcal{L} &= \frac{1}{M} \sum_{j \in 1 \ldots M} l(s_1^{(j)}, f_1(\mathbf{x}^{(j)})) \\ &= \frac{1}{M} \sum_{j \in 1 \ldots M} l(s_1^{(j)}, f_1([x_1, x_2]^{(j)})) \end{aligned} \tag{5.6}$$

note that the two different orders $[x_1, x_2]$ and $[x_2, x_1]$ in $\mathbf{x}$ are equivalent since $x_1$ and $x_2$ are statistically exchangeable due to the random sampling and both orders share the same ground truth $s_1$. Each mixture $x_1$ and $x_2$ in a training sample uses different values of $w_1$ and $w_2$ that are sampled independently from a uniform distribution between 0 and 1. It should be noted that even though all source signals $s_1$, $s_2$, and $s_3$ are sampled from the same domain $D_I$, this task is solvable for a neural network $f(\cdot)$ since the target source signal $s_1$ is specific and invariant to the order of either mixtures $\mathbf{x}$ or sources $\mathbf{s}$ in Equation 5.6. The workflow of the proposed task is shown in Figure 6.2.

It is worth mentioning that although it is trivial to solve the linear equations in Equation 5.6 and obtain $s_1$, $s_2$, and $s_3$ in math, it is non-trivial for the network to solve when formulated as a learning problem. This makes the proposed SI variant an efficient way to learn useful features from a dataset, without labor-consuming annotations, and

avoid the ambiguity problem at the same time. Compared to introducing a different domain to solve the ambiguity problem and learning differences between different domains in Section 5.3.2, the proposed method focuses on the same domain which is more likely to learn useful features for the downstream tasks.

### 5.3.4 Baseline Self-supervision Tasks

We compare the proposed method to four widely used self-supervision tasks for dense prediction [97, 118]. The first three tasks focus on the reconstruction and context-based prediction in an image, while the last task focuses on the intensity correction.

#### Inpainting

Image inpainting is the process of reconstructing the missing or damaged contents of an image, historically employed for restoring paintings and photographs [119]. Inpainting, as a self-supervision task, proceeds by intentionally masking selected areas within an image and a network must learn to recover the missing content.



In this paper, we implement inpainting self-supervision by overlaying an image $I$ with a regular grid $G$ of a fixed size and randomly masking selected grid cells. Formally, a selected grid cell of pixels, indicated as $g(I)$, where $g \in G$, is transformed as:

$$g'(I) = \begin{cases} g(I) & \text{if } \mathbb{B}(\gamma) = 1, \\ 0 & \text{otherwise.} \end{cases} \tag{5.7}$$

where $\mathbb{B}(\gamma)$ follows a Bernoulli distribution with $\gamma$ probability of being 1. $\gamma$ is a hyperparameter ranged from 0 to 1. That means in any minibatch, a network only sees approximately $\gamma$ random contents of the input images and tries to predict the rest of them. By masking grids in such a non-deterministic manner, we avoid cases where the network may focus on easy reconstructions and learning trivial features.

The resultant synthetic image $I'$ is made up of all the selected grids, $g'(I)$, thereby retaining $1 - \gamma$ fraction of the original image.

#### Local Pixel Shuffling

Local pixel shuffling has been known to aid a network in learning about the local information within an image, without compromising the global structures [97]. This task is similar to inpainting but with additional information on the distribution of intensities to inpaint. In this task, synthetic images are generated by randomly shuffling pixels within the selected grid cell, as shown in the following equation:

$$g'(I) = \begin{cases} Pg(I)Q & \text{if } \mathbb{B}(\gamma) = 1, \\ g(I) & \text{otherwise.} \end{cases} \tag{5.8}$$

where $\gamma$ is a hyperparameter ranged from 0 to 1 similar to that in inpainting; $P$ and $Q$ are permutation matrices. A permutation matrix is a binary square matrix which can permute the rows of an arbitrary matrix when being pre-multiplied to it, and permute the columns when being post-multiplied. Thus, in the first case of Equation

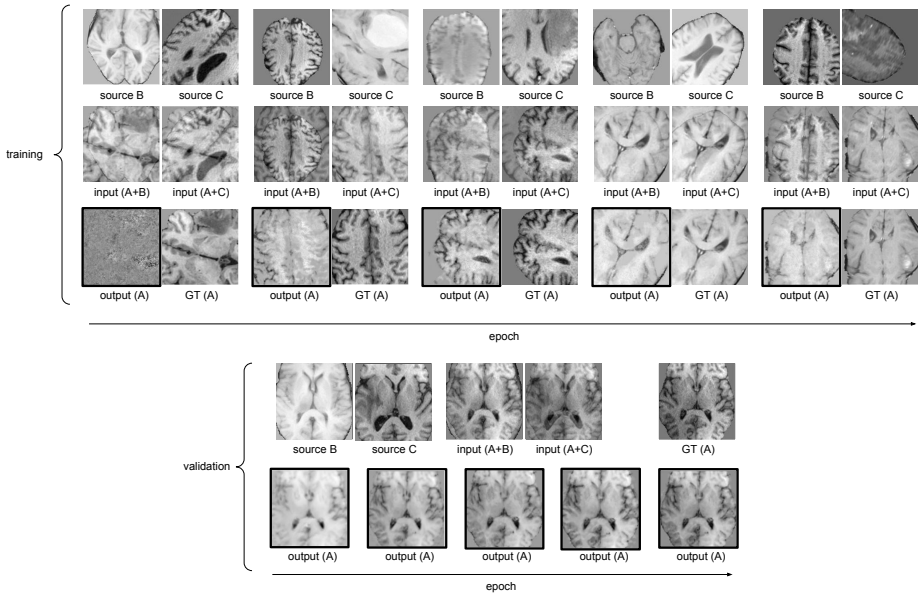



***Figure 5.3:*** ***Visualization of the network solving the SI task while being trained on the BraTS dataset.*** *T1 modality is used for visualization. The subfigure highlighted by thick boundary is the network output at each epoch. Epochs increase from left to right. Different training samples and the same validation sample from five intermediate epochs are visualized, to remain faithful to the fact that data augmentation is applied during training but not during validation. Cross-patients SI setting with the involved source number three is applied for visualization. The details of the source setting can be seen in Section 5.4.2. We can see that the network is able to identify and reconstruct the target source signal $A$ from the input mixtures $A+B$ and $A+C$ gradually during training.*

5.8, a new grid cell of pixels is generated by shuffling both the rows and columns of the original grid.

### Super-resolution

Super-resolution can be implemented as a self-supervision task [120], wherein a network is trained to deblur the low-resolution image. To create the low-resolution images from high-resolution ones for training, we blur the high-resolution images by transforming every grid cell by replacing all its values with that in the center of the grid:

$$g'(I) = g(I)_{(w/2,h/2)} \tag{5.9}$$

where $w$ and $h$ are the width and height of the grid cell $g(I)$. In the training process, given a transformed image as input, the network learns to predict the high resolution version which is the original image before transformation.

**Non-linear Intensity Shift**

The intensity shift mechanism is proposed by Zhou et al. [97], where each pixel value in the image is translated monotonically using a Bezier curve (denoted as function $B$) [121]. In medical imaging, since the intensity values in a image usually correspond to the underlying anatomical details, this task can be used to encourage a network to learn useful anatomical features.

Given a voxel value $v$ which is normalized between $[0, 1]$, end-points $p_0$, $p_3$, and two control-points $p_1$, $p_2$, the transformed value of the pixel is given by:

$$\begin{aligned} v' = B(v) = (1 - v^3)p_0 + 3x(1 - v^2)p_1 \\ +3v^2(1 - v)p_2 + v^3 p_3 \end{aligned} \tag{5.10}$$

where points from $p_0$ to $p_3$ are sampled independently at every epoch from a continuous uniform distribution between 0 to 1.



## 5.4 Experiments

### 5.4.1 Datasets

We apply our method on two medical imaging segmentation problems: brain tumor segmentation and white matter hyperintensities segmentation. Both datasets are brain MR images. All the trained self-supervised models in Section 5.3.3 and 5.3.4 are used as pre-trained models for the main segmentation task.

**BraTS Dataset**

Multimodal Brain Tumor Segmentation Challenge 2018 [12, 122, 123] focuses on evaluating methods for the segmentation of brain tumors in multimodal magnetic resonance imaging (MRI) scans. There are in total 210 MR images acquired from different patients. Each MR image contains four modalities: pre-contrast T1-weighted, post-contrast T1-weighted, T2-weighted, and FLAIR. Three brain tumor classes are provided as manual annotations: 1) the necrotic and the non-enhancing tumor core (NCR&NET); 2) the peritumoral edema (ED); and 3) the enhancing tumor (ET). Since the evaluation classes of the challenge are the combined classes: whole tumor (NCR&NET+ED+ET), tumor core (NCR&NET+ET), and enhancing tumor (ET), we use these combined classes for the actual training. We randomly split the dataset in 1) 100 subjects for training the self-supervision tasks and the main segmentation task; 2) 10 subjects for validation; and 3) 100 subjects for testing. For each subject, we cropped/padded MR images into a constant size of $200 \times 200 \times Z$ ($Z$ is the number of axial slices of the image) where the main brain tissues are preserved. Following the preprocessing of nnUNet [124], Gaussian normalization (subtracting mean and dividing by standard deviation) is applied on the brain foreground for each modality for each image individually.

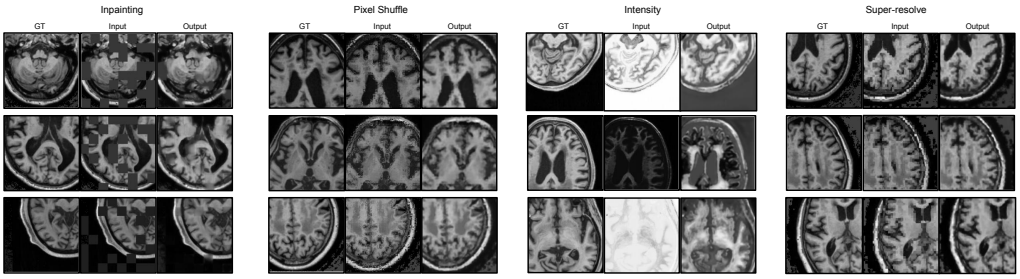



***Figure 5.4: Visual examples of baseline self-supervised tasks.*** *The results with best validation performance are used for visualization. Best viewed with zoom.*

### WMH Dataset

The White Matter Hyperintensities (WMH) Segmentation Challenge [34] evaluates methods for the automatic segmentation of WMH in brain MR images. The provided MR images contain T1-weighted and FLAIR MR sequences and are acquired from 60 patients, where each group of 20 patients are from a different hospital. The manual segmentation of WMH lesions are also provided for each image. We randomly split the dataset in 1) 30 subjects for training the self-supervision tasks and the main segmentation task; 2) 10 subjects for validation; and 3) 20 subjects for testing. For each subject, we centre-cropped/padded MR images into a constant size of $200 \times 200 \times Z$, where Z is the number of axial slices in the 3D image. The cropping/padding of was necessary as images from the different hospitals have slightly different sizes and it was convenient to have images of a constant size to have all of them processed in the same way by the network. Additionally, the size of $200 \times 200$ covers the main brain tissue, which is what the network needs to consume for learning. We use Gaussian normalization to normalize the intensities inside the brain foreground similar to the BraTS dataset.



## 5.4.2 Settings for The Proposed SI Task

There are two hyperparameters to tune in the proposed task. First is the transformation matrix $W$ in every mixture. In Section 5.3.3, we considered the example where $W$ defines linear combinations of three signals $s_1$, $s_2$, and $s_3$. More complicated $W$ can be constructed under the condition when the number of mixtures $m \geq 2$, for example, when $m = 2, n = 5$:

$$\begin{aligned} x_1 &= w_1 s_1 + w_2 s_2 + w_3 s_3, \quad w_1 + w_2 + w_3 = 1 \\ x_2 &= w_4 s_1 + w_5 s_4 + w_6 s_5, \quad w_4 + w_5 + w_6 = 1 \end{aligned} \tag{5.11}$$

where all weights are randomly sampled between 0 and 1 under the conditions.

The second hyperparameter is the source assignment strategy. In this paper, we consider three types of source assignment strategies:

**Cross-patients SI**

To make the network learn to identify the target source image and discriminate it from the other source images, $n$ random patients are used to extract signals (2D patch per patient) respectively in every training sample. We refer to this SI variant as *Cross-patients SI* (CSI).

**Within-patient SI**

To make the network focus on each particular source in the dataset, we use the same patient image to extract all $n$ signals (all 2D patches from the same patient). Since information from only 1 patient is used in each mixture, the network is unlikely to learn the cross-sources information among different patients. We refer to this SI variant as *Within-patient SI* (WSI).

**Denoising SI**



To investigate the difference between the proposed SI task and the traditional denoising task, we replace sources $s_2$ and $s_3$ in CSI with random Gaussian noise with zero mean and unit variance. This task is similar to a traditional denoising task and would encourage the network to learn representative features that distinguish differently distributed sources like image and noise explicitly. We refer to this SI variant as *Denoising SI* (DSI).

All experiments in Section 5.4 apply the linear combination of constituting three signals in two mixtures as showed in Equation 5.11, where in total $n = 5$ signals are used to generate a training sample. This setting is tuned on the validation set for both datasets. To avoid the network from learning trivial features, all combinations without any overlapping between the brain regions are excluded as training samples.

### 5.4.3 Settings for The Baseline Tasks

For inpainting, the grid size is tuned from ranging $[2, 2]$ to $[64, 64]$ and the masking percentage ranging from 0% to 100%; for local pixel shuffling and super-resolution, the grid size has the same tuning range as that in inpainting. There is no hyperparameter to tune for non-linear intensity shift. All hyperparameters are tuned on the validation set on the main task performance.

### 5.4.4 Network Architecture

We use the same network backbone for both the self-supervision proxy tasks and the segmentation main task. It is based on 2D UNet [11] and details of the network are shown in Figure 6.2. The network has two input-output layer settings: 1) for training the proposed SI task, the input layer has $T \times 2$ channels where $T$ is the number of imaging modalities for the two input mixtures. The output layer has $T$ channels for reconstructing all modalities of $s_1$; 2) for segmentation, the input layer is replaced with a new layer with $T$ channels for the input image $x$ and the output layer is replaced with a layer with $C$ channels for the segmentation predictions where $C$ is

***Table 5.1: Results of fully-supervised setting.*** *Each experiment is repeated 3 times with different random data split. For BraTS, the same 100 images are used for training the main segmentation task (with labels) and the self-supervision proxy tasks (without labels); for WMH, the same 30 images used for both labeled and unlabeled data. Mean Dice score (standard deviation) over all experiment testing data is reported for each class individually, where WT=whole tumor, TC=tumor core, ET=enhancing tumor, All=WT+TC+ET (only for BraTS), WMH=white matter hyperintensities. CSI: cross-patients source identification with different image sources. WSI: within-patient source identification with the same image source. DSI: denoising source identification. *: significantly better than the CNN baseline ($p < 0.05$). ⋄: significantly worse than the CNN baseline ($p < 0.05$). P-values are calculated by two-sided paired t-test in each class. Boldface: best and not significantly different from the best results.*

| | BraTS | | | | |
|---|---|---|---|---|---|
| Methods/Class | WT | TC | ET | *All* | WMH |
| CNN | 0.866(0.11) | **0.835**(0.17) | 0.785(0.16) | 0.846(0.11) | 0.775(0.11) |
| CNN-restart | 0.868(0.11) | 0.825(0.19)⋄ | 0.786(0.16) | 0.848(0.11) | 0.781(0.11) |
| Inpainting | 0.867(0.11) | **0.838**(0.17) | 0.788(0.16) | 0.850(0.11) | 0.782(0.13) |
| Pixel Shuffle | 0.859(0.13) | 0.829(0.20)⋄ | 0.777(0.18)⋄ | 0.844(0.13) | 0.777(0.12) |
| Intensity | 0.865(0.12) | **0.838**(0.18) | 0.787(0.16) | 0.846(0.12) | 0.775(0.13) |
| Super-resolve | 0.852(0.13)⋄ | **0.838**(0.18) | 0.786(0.17) | 0.842(0.13) | 0.776(0.12) |
| DSI | 0.868(0.09) | 0.821(0.17)⋄ | 0.783(0.16) | 0.850(0.10) | 0.771(0.12) |
| WSI | 0.869(0.09) | 0.817(0.19)⋄ | 0.781(0.16) | 0.851(0.10) | 0.769(0.12) |
| CSI(ours) | **0.878**(0.09)* | **0.837**(0.17) | **0.796**(0.15)* | **0.861**(0.09)* | **0.793**(0.11)* |

the number of classes. All the intermediate layers are shared between the pretrained proxy task and the main task. When no pretrained network is used, the weights of all convolutional layers are initialized by Kaiming initialization [125].

The choice of the network parameters are influenced by the state-of-the-art nnUNet [124] model, described in a forthcoming Section 5.4.5.

### 5.4.5 Training Strategy and Data Augmentation

We conduct main experiments in fully-supervised setting and semi-supervised setting for both datasets.

#### Fully-supervised Setting

There are two steps to train the network in a self-supervised manner. First, we need to pre-train the network with the corresponding proxy task, as described in Sections 5.3.3 and 5.3.4. The proxy task uses the same dataset as the main task; e.g., for the BraTS dataset, we pre-train and fine-tune the network on same 100 (labeled) images from the training set. A batchsize of 1 is used for the proxy task for all experiments

in this paper, realized via tuning from 1 to 4, based on the validation set. As for the main task, we use a batchsize of 8 and 4 for training the main task in BraTS and WMH, respectively, obtained by tuning between 1 to 16 based on the validation set.

#### Semi-supervised Setting

In the fully supervised setting, we utilize the entire training dataset to pre-train and fine-tune the network. Since the strength of self-supervision comes from a network needing a much smaller volume of data to be fine-tuned, we also conduct experiments to test this hypothesis, which we call the semi-supervised setting. In this setting, the network is pre-trained on the entire training dataset but fine-tuned on only a fraction of the training data. 25 of 100 labeled images were used from the training set to fine-tune the pre-trained model for BraTS; for WMH we used only 5 images. The same batchsize is used for both proxy task and main task as was used in fully-supervised setting.



#### Data Augmentation and Optimization Parameters

Random rotation, scaling, flipping, and elastic deformation are applied to the original 2D images as data augmentation for all experiments. Following the nnUNet paper, we use SGD optimizer and 'poly' learning rate policy $(1 - (\text{epoch}/\text{epoch}_{max})^{0.9})$, where $\text{epoch}_{max} = 1000$ and for the BraTS dataset and 10000 for WMH, with the initial learning rate $1 \times 10^{-2}$, momentum 0.99, and weight decay $3 \times 10^{-5}$ for both the proxy task and the main task. Early stopping is applied when there is no improvement for 50 epochs to avoid overfitting to the validation set. We also tried restarting the optimization for the main task with initial parameters, which we call *CNN-restart* for both datasets for fair comparison.

## 5.5 Results

### 5.5.1 Segmentation Results

Table 5.1 shows the segmentation results for the two datasets in the fully-supervised setting. The proposed Cross-patients SI method achieves the best average performance (except TC: the tumor core in BraTS) in both datasets and shows significant improvement over the other baselines and SI variants in four out of five classes (WT: whole tumor, ET: enhancing tumor, *All*: WT+TC+ET, and WMH). The *All* class calculates the Dice coefficient of WT+TC+ET together (by concatenating the three classes but not summing up them into one class) and is the most important one in BraTS.

Among the three different settings of source identification task (CSI, WSI, and DSI), CSI achieves the best results with a Dice score of 0.861 (All) and 0.793 in BraTS and WMH datasets separately, which is significantly better than WSI and DSI. WSI and DSI have similar performance in both datasets and are not significantly different from each other. This suggests the importance of the cross-source setting. One reason could be that compared to WSI and DSI, CSI is using the data more efficiently where the network sees more source images per epoch. It should also be noted that the

pixel shuffle task shows worse performance than the CNN baseline in four out of five classes (significant in TC and ET classes). In the tumor core (TC) segmentation, four methods (inpainting, intensity shift, super-resolve, and CSI) show comparable improvements to the CNN baseline (not significant to each other), which indicates the efficiency of different self-supervised methods may vary through different classes and the tumor core segmentation is more difficult to improve compared to other classes. Nevertheless, overall, the proposed CSI can provide a better starting point for the segmentation task than most of the self-supervision baseline tasks.

### 5.5.2 Semi-supervised Results

We conduct experiments on both datasets in semi-supervised settings in order to investigate how much the proposed self-supervision task would help when only a smaller amount of labeled data is available to train the proxy task. The results are shown in Table 5.2. Similar trends can be observed from these semi-supervised results compared to those in fully-supervised results. Similar to Table 5.1, the proposed CSI method gets the largest improvements in BraTS (except the tumor core) and WMH. The improvements are significant compared to all other methods in whole tumor and *All* in BraTS. In WMH, both the proposed CSI method and inpainting are significantly better than the other methods. It should also be noted that when only few labeled images are available, more self-supervision methods show significant improvements compared to CNN baseline (12∗ results in Table 5.2 compared to 4∗ results in Table 5.1). This shows the general advantages of feature learning in self-supervision methods compared to CNN baseline.

The SI variants WSI and DSI still show close performance to each other in most classes and perform significantly worse than CSI. Similar to the fully-supervised setting, the pixel shuffle task does not show improvements compared to the CNN baseline in most classes. It should be noted that the CSI performance in semi-supervised setting (0.837 in BraTS and 0.783 in WMH) is very comparable to the fully-supervised CNN baseline result (0.846 in BraTS and 0.775 in WMH), which required 4 times more training images. Inpainting and super-resolve show better performance than CNN baseline, but still worse than CSI (significant in BraTS). The proposed method shows larger performance improvements in WMH dataset where far fewer labeled data are used compared to BraTS dataset (5 labeled vs. 25 labeled and with 4.4% vs. 3.1% Dice improvements to the CNN baseline). This shows in a practical situation in medical imaging where segmentation labels are scarce, a well-designed self-supervision task can still preserve considerable performance given enough unlabeled data.

### 5.5.3 Influence of The Number of Sources

We conduct experiments to investigate the influence of the number of images used in the proposed SI task. $n = 3, 5, 7$ sources (e.g. in Equation 5.11, $n = 5$) are tested to generate $m = 2$ fused images as input to the network. The experiments are independent runs on BraTS and WMH dataset in fully-supervised setting. Note that the hyperparameter $n$ is tuned on the validation set for all experiments. The results are shown in Figure 5.5. We can see that the $n = 5$ sources setting achieves the best

***Table 5.2: Results of semi-supervised setting.*** *The best results are marked in bold. Each experiment is repeated 3 times with different random data split. For BraTS, all 100 training images are used to train the unlabeled self-supervised task; fine-tuning is performed on 25 of the training images using the segmentation labels and the 25 labeled images are contained in the 100 unlabeled images. For WMH, 5 images are used for labeled data and 30 images are used for unlabeled data and the 5 labeled images are contained in the 30 unlabeled images. Mean Dice score (standard deviation) over all experiment testing data is reported for each class individually, where WT=whole tumor, TC=tumor core, ET=enhancing tumor, All=WT+TC+ET (only for BraTS), WMH=white matter hyperintensities. *: significantly better than the CNN baseline ($p < 0.05$). ⋄: significantly worse than the CNN baseline ($p < 0.05$). P-values are calculated by two-sided paired t-test in each class. Boldface: best and not significantly different from the best results.*

5

| | BraTS | | | | |
|---|---|---|---|---|---|
| Methods/Class | WT | TC | ET | *All* | WMH |
| CNN | 0.823(0.11) | 0.780(0.21) | 0.743(0.19) | 0.816(0.12) | 0.739(0.16) |
| CNN-restart | 0.821(0.13) | 0.775(0.22) | 0.739(0.19) | 0.812(0.13) | 0.731(0.16)⋄ |
| Inpainting | 0.842(0.15) | **0.817**(0.20)* | 0.754(0.18)* | 0.827(0.15) | **0.761**(0.12)* |
| Pixel Shuffle | 0.823(0.17) | 0.782(0.23) | 0.723(0.21)⋄ | 0.806(0.17) | 0.744(0.15) |
| Intensity | 0.832(0.16) | 0.804(0.21)* | 0.746(0.19) | 0.817(0.16) | 0.740(0.15) |
| Super-resolve | 0.848(0.15) | **0.819**(0.20)* | **0.760**(0.19)* | 0.829(0.14) | 0.756(0.13)* |
| DSI | 0.823(0.16)⋄ | 0.776(0.22) | 0.747(0.20) | 0.804(0.16)⋄ | 0.755(0.13) |
| WSI | 0.836(0.13) | 0.779(0.20) | 0.749(0.18) | 0.814(0.13) | 0.754(0.12) |
| CSI(ours) | **0.855**(0.12)* | **0.811**(0.18)* | **0.764**(0.17)* | **0.837**(0.12)* | **0.783**(0.11)* |

performance in the main segmentation task for CSI and WSI while for DSI the effect is much smaller. Too few sources may make it too easy to reconstruct the target signal which may result in trivial features, while too many sources may make it too difficult to recognize the target, resulting in arbitrary features.

### 5.5.4 Comparison to Mixup

Table 5.3 shows the results to compare our proposed approach to Mixup. Here, the Cross-patients Source Identification (CSI) based self-supervised pre-training is compared to the baseline CNN without any pre-training, with and without Mixup as an additional data augmentation strategy. Results show that Mixup improves both the baseline and our proposed approach, with a higher relative improvement for detecting tumours in the BraTS dataset.

***Table 5.3: Comparison of CSI to Mixup.*** *The best results are marked in bold. Each experiment is repeated 3 times with random data splits, and the mean dice scores are reported.*

| Methods/Class | BraTS WT | WMH |
|---|---|---|
| CNN | 0.866 | 0.774 |
| CNN+mixup | 0.875 | 0.794 |
| CSI | 0.878 | 0.801 |
| CSI+mixup | **0.886** | **0.803** |

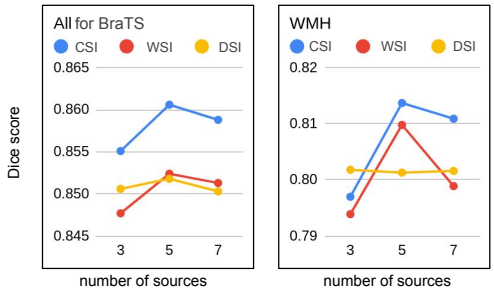



***Figure 5.5: Influence of the number of fused sources.*** *The results are obtained by an independent run on BraTS and WMH dataset using the same data in a random data split with fully-supervised setting, similar to Table 5.1. The number of sources 5 is used for experiments in Table 5.1 and 5.2. Best viewed in color.*

## 5.6 Discussion

In this paper, we propose a new self-supervision task named source identification (SI) which is inspired by the blind source separation problem, and we investigate the task ambiguity in the SI problem for neural networks. Unlike most previous reconstruction-based self-supervision tasks that focus on restoring image contents from only one source image, the proposed task enables the network to see multiple images from mixtures and learn to separate the source image from the others and reconstruct it. The experiments show that the proposed method outperforms baseline methods in both datasets including the CNN baseline, restart-CNN with the initial learning rate, and commonly used self-supervised methods inpainting, pixel shuffle, intensity shift, super-resolution, and denoising. The proposed method shows the largest improvements in the semi-supervised setting when very few labeled data and many unlabeled data are available, which is a common scenario in medical imaging applications.

### 5.6.1 Comparison to Other Self-supervision Methods

One main difference between the proposed SI task and existing reconstruction-based self-supervision tasks is that SI learns features from not only the remaining part

of the same distorted image but also from other images of the same domain. By distinguishing each image from others, potentially useful discriminative features can be learned while reconstructing the target image. These features may better capture general domain knowledge, e.g. anatomy and pathology knowledge, by seeing and comparing different patients' images at the same time. A proper understanding of anatomy and pathology across different individuals is required to successfully solve a single image identification and reconstruction. Features learned by SI may therefore provide a better starting point for optimization of the downstream task than the features learned by previous self-supervision tasks such as inpainting, pixel shuffling, intensity shift, super-resolution, and denoising.

In this paper, we focus on the comparison between reconstruction-based self-supervised methods, which all use the synthetic distorted image as input and the original target image as ground truth. We consider the context prediction-based methods such as tiles location prediction [95], puzzle solving [77], contrastive learning [100] as another category of self-supervised tasks. These methods optimize a predefined classification/regression task based on the information within a single image [77, 95] or across different images [100], and thus they usually do not train a relevant (dense) decoder. On the contrary, the reconstruction-based methods inherently require a dense decoder for learning concrete and high-resolution features and outputting dense pixelwise predictions, which may result in a model that fits better to dense prediction tasks like segmentation.



### 5.6.2 Apply SI using Unlabeled Data with Less Overfitting

Self-supervised learning allows using unlabeled data without additional annotations from experts and pretraining with both labeled and unlabeled data before fully-supervised learning. The quality of the learned features from self-supervised tasks is usually evaluated on downstream tasks like segmentation. In our experiments, larger improvements are observed in the semi-supervised setting compared to fully-supervised setting, especially for the WMH dataset. Our results show that given the same amount of unlabeled data, the proposed SI can learn more useful features from unlabeled data compared to other self-supervised tasks. One reason could be that the proposed SI task may suffer less from the overfitting problem compared to traditional methods like inpainting and super-resolution. For example, given the unlabeled data, the model may try to solve the inpainting or super-resolution task by memorizing the input images and restoring the missing content when there is enough model capacity, which may result in learning trivial features. In contrast, the SI task takes inputs from many more different combinations of images given the same amount of unlabeled data (when $m = 2$ and $n = 5$ in 100 images, the number of possible image combinations would be the binomial coefficient $C(100, 5) \times 5 \approx 3.8 \times 10^8$), which makes the model more difficult to memorize and overfit to a particular image but has to find a more general way to solve the SI task, e.g. learning anatomy knowledge, which can be non-trivial and useful for downstream tasks like segmentation.

### 5.6.3 Application to Other Dense Prediction Tasks

In this paper, we apply the proposed SI method to segmentation, a dense prediction task. The pretrained SI features can also be transferred to other medical imaging dense prediction tasks such as for instance depth estimation [126], image registration [127], and detection based on distance maps [128]. Moreover, these tasks may also benefit from the cross-sources features learned in the SI method. For example, a good image registration model may require not only the alignments between local patterns across different modalities (within one patient) but also the general anatomy knowledge across different patients to constrain possible transformations. With a proper design of the proxy dataset and the SI setting, the potential scenarios to apply the proposed method can be greatly extended.

### 5.6.4 Limitations

It has been studied in literature that the performance of self-supervised approaches differ significantly based on the difficulty of the pretraining task and it's relatedness to the main task [99, 129, 130]. For example, the performance of inpainting as a self-supervision task would suffer when the size of the masked area is too large or too small. Too large the masked area, and the pretraining task would be too difficult to solve; too small and it would be very easy. This would affect the quality of the learned features, and hence the efficiency of the network on the main task. Similarly, for our approach, the performance of the network is determined by how separable the mixed images are and how much information the network needs to learn to separate them.

We indirectly test the former hypothesis in Section 5.3.2, where it is shown that very similar images would be extremely hard to separate. To explore whether this is a practical problem in our case, we devise a simple statistical experiment. First, pairs of 2D slices are randomly sampled from different images in one dataset (BraTS

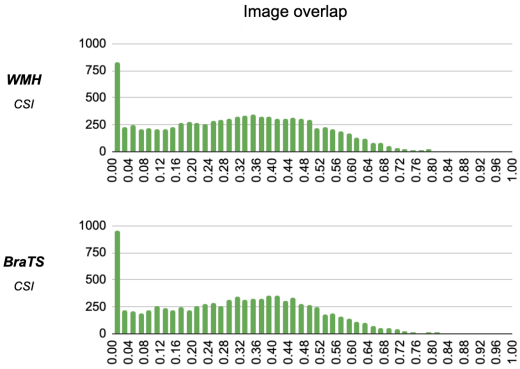



***Figure 5.6: Distribution of the degree of mixing randomly sampled images.*** *We observe that the degree of mixing randomly sampled 2D slices from distinct images is almost uniform in the two datasets used.*

or WMH), and data augmentation is applied to them as described in Section 5.4.5. Next, the brain mask is extracted from the resulting images, via simple intensity based thresholding, and the overlap of the corresponding brain masks are measured using Jaccard Similarity. Finally, the distribution of the similarities measured is plotted and shown in Figure 5.6. As we can observe, the similarities almost uniformly range from very low (nearly 0) to moderately high (0.75), indicating that for our datasets, the network would receive a wide range of mixed images for training. As mentioned in Section 5.4.2, we exclude all mixed images with 0 similarities (no overlap at all) to avoid the network from learning trivial features. Thus, for our experiments, we do not need extra control of the degree of mixing images.

The second hypothesis revolves around how much information the network needs to learn to identify the source from the mixed images. In Section 5.5.3 we empirically demonstrate the effect of the number of fused sources on the final performance. It is noticed that too few or too many fused sources are detrimental to the efficiency of the network.

Our proposed approach is sensitive to these two degrees of freedom and, although we have enough empirical evidence for the datasets in question, further testing is required to make a general comment about the sensitivity of our method to these two factors.



## 5.7 Conclusion

In conclusion, we propose a novel self-supervision task called source identification which is inspired by the classic blind source separation problem. The proposed task is to identify and separate a target source image from mixtures with other images in the dataset, which requires features that are also relevant for the downstream task of segmentation. On two brain MRI segmentation tasks, the proposed method provides a significantly better pretrained model for segmentation compared to other self-supervision baselines including inpainting, local pixel shuffling, non-linear intensity shift, and super-resolution in both fully-supervised and semi-supervised settings. The proposed method can be generalized to other dense prediction applications.

## 5.8 Acknowledgment

The authors would like to thank Gerda Bortsova for her constructive suggestions for the paper, and organizers of BraTS 2018 and WMH 2017 Challenges for providing the public datasets. This work was partially funded by Chinese Scholarship Council (File No.201706170040).

# Chapter 6

## Label Refinement Network from Synthetic Error Augmentation for Medical Image Segmentation

## Abstract

*Deep convolutional neural networks for image segmentation do not learn the label structure explicitly and may produce segmentations with an incorrect structure, e.g., with disconnected cylindrical structures in the segmentation of tree-like structures such as airways or blood vessels. In this paper, we propose a novel label refinement method to correct such errors from an initial segmentation, implicitly incorporating information about label structure. This method features two novel parts: 1) a model that generates synthetic structural errors, and 2) a label appearance simulation network that produces synthetic segmentations (with errors) that are similar in appearance to the real initial segmentations. Using these synthetic segmentations and the original images, the label refinement network is trained to correct errors and improve the initial segmentations. The proposed method is validated on two segmentation tasks: airway segmentation from chest computed tomography (CT) scans and brain vessel segmentation from 3D CT angiography (CTA) images of the brain. In both applications, our method significantly outperformed a standard 3D U-Net and other previous refinement approaches. Improvements are even larger when additional unlabeled data is used for model training. In an ablation study, we demonstrate the value of the different components of the proposed method.*




Based on: **S. Chen***, A. Garcia-Uceda*, J. Su*, G. van Tulder, L. Wolff, T. van Walsum, and M. de Bruijne, "Label refinement network from synthetic error augmentation for medical image segmentation," *Submitted*

## 6.1 Introduction

Convolutional neural networks (CNNs) are the state-of-the-art for many biomedical imaging segmentation tasks. Many CNN segmentation architectures have been proposed, such as fully connected networks [131], Dense-Net [132] and the U-Net [11]. The U-Net has become the most popular network for biomedical image segmentation, due to its efficient structural design featuring skip-connections, showing superior accuracy and robustness in various segmentation tasks [6, 133]. Most CNN-based segmentation methods including the U-Net do not fully exploit and encode the structural information of the objects to be segmented. Consequently, these methods may produce segmentations with errors that become obvious when looking at the full segmented structure. Examples of such errors are discontinuities in the segmentations of elongated tubular structures, such as airways in the lungs, as shown in Figure 6.1. Using label structural knowledge such as continuity in the branches of the airway tree can help prevent these errors. However, it is not trivial to explicitly encode this global information in CNNs.

In this paper, we propose a framework to implicitly encode the label structural information into CNNs by formulating this as a label refinement step. Specifically, we generate structural errors in labels (such as the ground truth or initial segmentations) and train a label refinement network to correct these errors. The trained network is expected to generalize to the real errors in the initial segmentations produced by a baseline segmentation network and correct them. To enhance the generalizability of the label refinement network on the initial segmentations, a label appearance simulation network is applied to reduce the appearance difference between the synthetic labels and the initial segmentations. With these synthetic labels (and the initial segmentations) together with the original image as inputs and the ground truth segmentations as reference, the label refinement network can learn to correct those errors and incorporate this in its segmentation decisions.

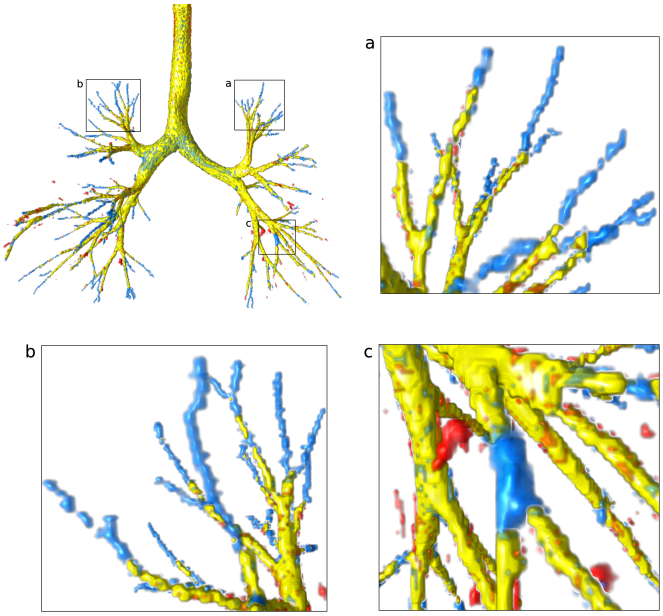
We validated the proposed label refinement method on two segmentation tasks: airway segmentation from chest computed tomography (CT) scans [3] and brain vessel segmentation from 3D CT angiography (CTA) images of the brain [4]. We compared our method with a U-Net baseline and other refinement networks, including DoubleU-Net [134] and SCAN [135], an adversarial refinement network. Moreover, we conducted an ablation study to show the contribution of each individual component of the label refinement method. Finally, we performed experiments in a semi-supervised setting to train our method using additional unlabeled data.



## 6.2 Related Work

### 6.2.1 Label Refinement

In this work, we apply a refinement network on the initial segmentation from a baseline segmentation network together with the original image, with the aim of correcting errors in the initial segmentation. A similar approach has been used in other previous papers. Jha et al. [134] attached a second U-Net network to a baseline U-Net, using as inputs the original image multiplied with the output of the first U-Net. Yang et al. [136] refined low-quality manual annotations made by non-experts




6

***Figure 6.1: Common structural errors in the segmentations obtained by a U-Net, trained to segment airways in the lungs [3].*** *True positives are displayed in yellow, false negatives in blue and false positives in red. Detailed views a-b show errors as missing terminal branches, and view c shows a discontinuity error in the branch. Better to view in zoom in color.*

by training their method with added noise in order to reduce the inter-observer inconsistency of the annotations. Unlike our method, Yang et al. do not focus on refining an initial automatic segmentation and therefore the label appearance simulation network is not needed. Dai et al. [135] refined the segmentations from a fully convolutional network by using adversarial training to reduce the domain gap between the target predictions and the ground truth segmentations on training data. Araújo et al. [137] attached a variational auto-encoder after a U-Net network to encode the label topology of the ground truth segmentations for a better label reconstruction. Different from Dai et al. and Araújo et al., our work does not focus on encoding [137] or discriminating [135] the overall label topology, but instead on learning to correct the most common errors in the segmentations.

### 6.2.2 Airway Segmentation

The airway tree in the lungs forms a complex 3D tree-like branching network, with many branches of different sizes and orientations. The peripheral branches of smaller size are challenging to segment from chest CT scans, as they have obscured borders due to partial volume effects. Many classical methods for airway tree extraction

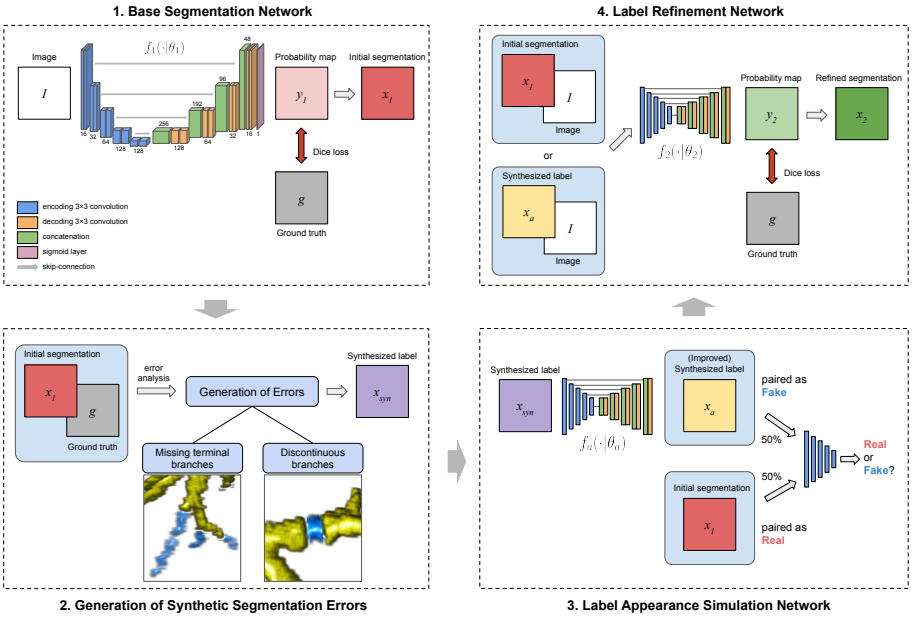



***Figure 6.2: Schematics of the proposed label refinement method.*** *First, a base segmentation network $f_1$ is trained to obtain the initial predictions $x_1$. Second, we synthesize a new dataset $x_{\text{syn}}$ that contains similar errors to $x_1$. Third, a label appearance improvement network $f_{\text{a}}$ (together with a discriminator $D$) is trained to obtain a more realistic dataset $x_{\text{a}}$. Finally, the label refinement network $f_2$ is trained with $x_1$ and $x_{\text{a}}$ together with the image $I$ as inputs.*

are based on a region growing algorithm [138, 139, 140]. However, their accuracy is limited, and they typically miss a large number of the smaller peripheral airways [141]. Many state-of-the-art airway segmentation methods are based on CNNs, and especially the U-Net [3, 142, 143, 144]. CNN-based methods can obtain more accurate and complete segmentations than previous intensity-based methods. However, even the latest U-Net-based methods usually miss several terminal branches, and make errors in continuity around the smaller segmented branches.

### 6.2.3 Brain Vessel Segmentation

The brain vessels form a complex 3D branching network that consists of veins and arteries. In 3D CTA images of the brain, many seemingly isolated vessel structures can be present due to the image acquisition and vascular diseases, such as ischemic large vessel occlusions. State-of-the-art vessel segmentation methods have been applied to 3D time-of-flight (TOF) magnetic resonance angiography (MRA) images [145, 146, 147], and to 3D and 4D CTA images [148] using U-Nets. Su et al. [4] used a

U-Net-based method to extract a dilated vessel centerline approximation. Compared to previous vessel segmentation methods [145, 146, 147, 148], centerline extraction recovers the topology of the vessel structure more accurately (e.g., "kissing vessels" appear connected in the full segmentations but are disconnected in centerline extraction). However, the U-Net still makes other topological errors such as local connectivity gaps in vessel branches.

## 6.3 Method

### 6.3.1 Overview

The proposed method consists of four steps, schematically shown in Figure 6.2. Firstly, a baseline segmentation network generates the initial segmentations (Section 6.3.2). Secondly, synthetic errors are generated and added to every ground truth segmentation, in order to generate synthetic labels to train the label refinement network (Section 6.3.3). Thirdly, a label appearance simulation network (LASN) based on adversarial learning is used to reduce the appearance difference between the synthetic labels and the initial segmentations (Section 6.3.4). Finally, a label refinement network is trained to predict the final segmentation, using the synthetic labels (or the initial segmentations) and the original images as inputs, and the ground truth segmentations as reference (Section 6.3.5).



### 6.3.2 Base Segmentation Network

We use a base segmentation network $f_1$ to predict an initial segmentation. Given a medical imaging dataset that contains an image $I$ and the ground truth segmentation $g$ for each subject, the model $f_1(I|\theta_1)$, with $\theta_1$ the trainable parameters, is trained by minimizing the Dice loss $\mathcal{L}_1 = \mathcal{L}_{\mathrm{dc}}(f_1(I), g)$:

$$\mathcal{L}_{\mathrm{dc}}(y, g) = -\frac{2\sum_{i\in I} y_i g_i}{\sum_{i\in I} y_i + \sum_{i\in I} g_i} \tag{6.1}$$

where $y_i$ and $g_i$ are the $i$th voxel values of the probability maps output by the model (in this case $y = f_1(I)$), and the ground truth segmentation, respectively.

The initial predicted segmentation is $x_1$, obtained by thresholding the output probability maps $y_1$ of the network with value 0.5. $x_1$ may contain label structural errors, such as discontinuous branches in a tree-like structure. Next, we show how to design synthetic errors similar to those in $x_1$ that can be used to train the label refinement network.

### 6.3.3 Generation of Synthetic Segmentation Errors

We use synthetic labels $\mathbf{x}_{\mathrm{syn}}$ with added synthetic errors to train the label refinement network. Depending on the experimental set-up, the errors can be added to the ground truth or to the initial segmentations. The synthetic errors are generated to resemble those in the initial segmentations $x_1$, based on our initial analysis of common errors. In this paper, we focus on two structures: airways in the lungs and vessels in the

brain. Airways and vessels share several characteristics: they both form 3D branching networks, with branches of cylindrical shape and various sizes and orientations. We use this prior shape knowledge to generate synthetic errors, as described below for each structure.

### Synthetic Errors For Airways

Most of the errors in airway segmentations can be grouped into two types: 1) missing terminal branches, partially or totally, and 2) discontinuity in the segmented branches, which occurs more frequently in smaller branches. Examples of errors in airway segmentations obtained by the baseline segmentation network in Section 6.3.2 are shown in Figure 6.1. To generate similar, synthetic errors, we select a random subset of branches in the airway tree and partially remove the segmentation of the selected branches by masking it at a random position and with a random length. Branches are identified using the airway centerline tree, extracted from the airway segmentation [141]. Single branches are defined as the segments between two bifurcation points or between the last bifurcation and the end of terminal branches. The applied masking is defined differently for each type of error:

*Missing terminal branches*: The subset of branches in which to synthesize errors is randomly sampled from all the terminal branches in the airway tree, defined as branches with no further bifurcations downstream. A mask of cylindrical shape is applied to (partially) remove the selected branch. The mask is defined by 1) a starting point, that is a random position along the branch centerline between the branch start and middle points; 2) a length, that is the distance between the mask start point and branch end; and 3) a width, that is three times the branch diameter.

*Discontinuity in branches*: The subset of branches with errors is randomly sampled from all the branches in the airway tree, excluding the trachea, the two main bronchi and the 2$^{\text{nd}}$ generation airways, and including the terminal branches. We assign a higher sampling probability to branches of higher airway generation, where the generation is defined as the number of branch bifurcations counted in the path linking the given branch and the root of the airway tree, i.e., the trachea. The sampling probability $p_i$ for each candidate branch is defined as $p_i = g_i / \sum_{k=1}^{N_c} g_k, \forall i = 1 \dots N_c$, where $g_i$ is the airway generation and $N_c$ the number of candidate branches. A mask of cylindrical shape is applied to create a gap in the selected branch. The mask is defined by 1) a center, that is a random position along the branch centerline; 2) a length, that is a random distance between a minimum of 10 voxels and the total branch length; and 3) a width, that is three times the branch diameter.

*Parameters*: The extent of each type of errors in the airway synthetic labels is determined by a separate parameter, denoted as $p_1^{\text{a}}$ and $p_2^{\text{a}}$. $p_1^{\text{a}}$ is the proportion of selected branches with errors of type "missing terminal branches", with respect to all the terminal branches. $p_2^{\text{a}}$ is the proportion of selected branches with errors of type "discontinuity in branches", with respect to all the branches in the airway tree.

### Synthetic Errors for Brain Vessels

Most of the errors in brain vessel segmentation are in the form of incomplete or missing vessel branches. To generate similar, synthetic errors, we create random discontinuous

gaps in the segmentation of each vessel by masking it at a random position and with a random length. Since the errors occur more frequently for long vessels than for short ones, we group all the vessels into three equal-sized groups: long, medium size and short, based on the relative centerline segment lengths in each subject. The distribution of vessel lengths (in voxels), using the median and interquartile range (IQR), is: for long segments 70 (49–106), for medium-size segments 29 (22–36) and for short segments 13 (9–17). For long segments, the maximum number of injected gaps is 6 (randomly sampled from a uniform distribution between 0 and 6 positions) with gap length between 10–35 voxels. For medium-size segments, the maximum number of gaps is 4 with gap length between 10–20 voxels. For the short segments, the maximum number of gaps is 2 with gap length between 6–15 voxels. Those error injections are applied on the 1 voxel-wide ground truth centerlines, by dilating it with a 3×3×3 cubic structure element to generate the final vessel synthetic label.

*Parameters*: The extent of errors in the vessel synthetic labels is determined by only one parameter, denoted as $p^{\mathrm{v}}$. $p^{\mathrm{v}}$ is the proportion of selected branches with errors with respect to all the branches in the vessel network.

### 6.3.4 Label Appearance Simulation Network

Although the synthetic labels $x_{\mathrm{syn}}$ are designed to have similar structural errors to the initial segmentation $x_1$, there may be an appearance difference between $x_1$ and $x_{\mathrm{syn}}$ (see an example in Figure 6.3a and b). The label refinement network trained on $x_{\mathrm{syn}}$ may therefore generalize poorly to $x_1$. To prevent this, we use a label appearance simulation network $f_{\mathrm{a}}(\cdot|\theta_{\mathrm{a}})$ to change the appearance of $x_{\mathrm{syn}}$ to be more similar to that of $x_1$, while preserving the synthetic errors that we added to $x_{\mathrm{syn}}$.

The label appearance simulation network $f_{\mathrm{a}}(\cdot|\theta_{\mathrm{a}})$, with $\theta_{\mathrm{a}}$ the trainable parameters, is optimized by adversarial learning via a discriminator $D$:

$$f_{\mathrm{a}}^{*} = \arg\min_{f_{\mathrm{a}}}((\max_{D} \mathcal{L}_{\mathrm{adv}}(f_{\mathrm{a}}, D)) + \lambda \mathcal{L}_{\mathrm{dc}}(x_{\mathrm{a}}, x_{\mathrm{syn}})) \tag{6.2}$$

with the adversarial loss $\mathcal{L}_{\mathrm{adv}}$ defined as:

$$\mathcal{L}_{\mathrm{adv}}(f_{\mathrm{a}}, D) = \mathbb{E}_{x_1}[\log D(x_1)] + \mathbb{E}_{x_{\mathrm{a}}}[\log(1 - D(x_{\mathrm{a}}))] \tag{6.3}$$

where $D$ is a classifier, discriminating the given label $x$ and the initial segmentation $x_1$. It outputs a probability between 0.0 and 1.0. $x_{\mathrm{a}} = f_{\mathrm{a}}(x_{\mathrm{syn}})$ is the appearance-enhanced label of $x_{\mathrm{syn}}$. We added a Dice-based identity loss $\mathcal{L}_{\mathrm{dc}}(x_{\mathrm{a}}, x_{\mathrm{syn}})$ to train $f_{\mathrm{a}}(\cdot)$, in order to preserve the synthetic errors that we added in $x_{\mathrm{syn}}$. The hyperparameter $\lambda$ controls the balance between the identity loss and the dissimilarity adversarial loss.

### 6.3.5 Label Refinement Network

Finally, we optimize a label refinement network $f_2$ to predict the ground truth segmentations, based on the synthetic labels with errors $x_{\mathrm{a}}$ together with the original image as inputs. This way, $f_2$ learns to correct segmentation errors and can be used to improve the initial segmentations $x_1$. The model $f_2(I|\theta_2)$, with $\theta_2$ the trainable parameters, is trained by minimizing the Dice loss $\mathcal{L}_2 = \mathcal{L}_{\mathrm{dc}}(f_2(I, \tilde{x}), g)$, given by

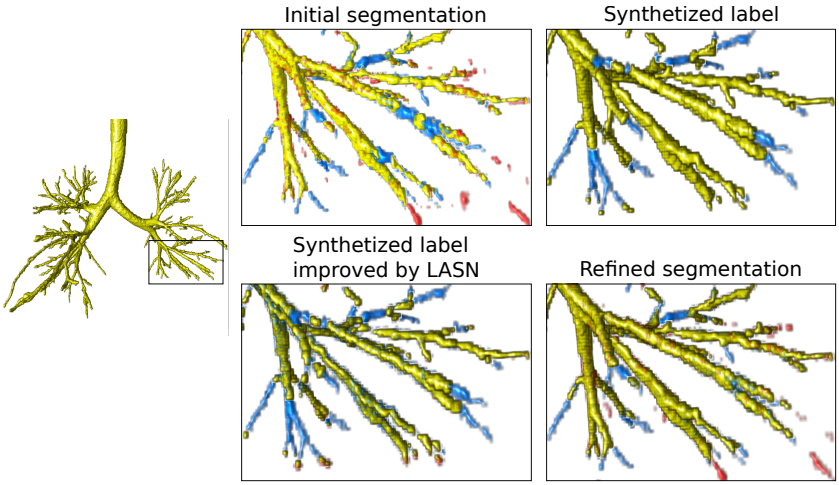



***Figure 6.3:*** ***Example of segmentation of airways in the lungs obtained by the different components of the proposed method.*** *In the detailed views, true positives are displayed in yellow, false negatives in blue and false positives in red. Better to view in zoom in color.*



equation 6.1. The final segmentation result is $x_2$, obtained by thresholding the output probability maps $y_2$ of the refinement network with value 0.5.

## 6.4 Experiments

### 6.4.1 Datasets

We validated the proposed method on two biomedical imaging segmentation tasks: segmenting airways from chest CT scans and brain vessels from CTA images of the brain.

#### Chest CT data

The dataset of chest CT scans is from a retrospective study of pediatric patients (6 to 17 years old) with cystic fibrosis lung disease, acquired routinely at the hospital Erasmus MC-Sophia Rotterdam [149]. The CT scans show noticeable structural airway abnormalities resulting from the disease. In our study, we used 178 low-dose CT scans acquired at full inspiration breath-hold. All CT scans have slice dimensions 512×512, with a variable number of slices between 200–1000. Each CT scan has an in-plane voxel size in the range 0.35–0.65 mm, with slice thickness between 0.75–1.0 mm, and slice spacing between 0.3–0.8 mm. A random subset of 65 CT scans from the total 178 scans have annotations of the airway lumen. To obtain these annotations, Thirona's lung quantification software LungQ (Thirona, Nijmegen, the Netherlands) was used to

automatically extract the airway lumen from the CT scan. Then, these segmentations were visually checked by trained data analysts for accuracy, and corrected as needed.

For our experiments, we used as testing data 41 random CT scans from the subset of 65 CT scans with ground truth segmentations. From the remaining 24 CT scans with annotations, we used three different random data splits with 20 CT scans for training the networks and 4 CT scans for validation. The remaining 113 CT scans without ground truth segmentations were used as unlabeled training data for the experiments with semi-supervised learning.

#### CTA data of the Brain

The dataset of CTA images of the brain is from the MR CLEAN Registry [150], an ongoing registry for patients who underwent endovascular treatment for acute ischemic stroke in one of 19 hospitals in the Netherlands since March 2014. The data was collected during clinical practice, and we applied the following data inclusion criteria: 1) slice thickness $\leq$1.5 mm, 2) slice spacing $\leq$1.5 mm, 3) the contrast acquisition phase has to be peak arterial phase, equilibrium or early venous phase [151], and 4) the image should cover at least half of the brain. In our study, we used 69 CTA images from 69 different subjects used in [4]. All CTA images were skull-stripped with an atlas-based registration method [152]. 20 CTA images had no vessel annotations, 9 CTA images had a complete brain vessel centerline annotation, and the remaining 40 CTA images (randomly sampled from the whole dataset) had vessel centerline annotations in a randomly sampled sub-volume of 140×140×140 voxels. The centerline annotations were dilated with a 3×3×3 cubic structure element to obtain the ground truth segmentations. Each CTA image has an in-plane voxel size in the range 0.4–0.68 mm, with slice thickness between 0.5–1.5 mm, and slice spacing between 0.3–1.0 mm.

For our experiments, we used as testing data 2 random full-volume CTA scans and 20 random CTA cubes from the set of 9 full-volume, annotated CTA scans and 40 CTA cubes, respectively. From the remaining data with annotations, we used three different random data splits with 7 full-volume CTA scans and 14 CTA cubes for training the networks, and 6 CTA cubes for validation. The remaining 20 full-volume CTA scans without manual annotations were used as unlabeled training data for the experiments with semi-supervised learning.

### 6.4.2 Parameters for Generating Synthetic Errors

The generation of synthetic errors depends on the parameters $p_1^{\mathrm{a}}$ and $p_2^{\mathrm{a}}$ for airways, and $p^{\mathrm{v}}$ for vessels, described in Sections 6.3.3 and 6.3.3 respectively. In the rest of the paper we refer to these parameters as "synthetic error rate", for each type of error. For each training sample, the synthetic error rate is randomly sampled from a uniform distribution between 0.0 and the upper bound, or maximum synthetic error rate. These upper bounds are hyperparameters for the proposed method, denoted as $P_1^{\mathrm{a}}$ and $P_2^{\mathrm{a}}$ for airways, and $P^{\mathrm{v}}$ for vessels.

We conducted experiments varying the hyperparameters for the error generation in the proposed method, i.e., the maximum synthetic error rates ($P_1^{\mathrm{a}}$ and $P_2^{\mathrm{a}}$ for airways, and $P^{\mathrm{v}}$ for vessels), to investigate their influence in the method performance. The results are shown in Section 6.5.3 below.

In our further experiments, the optimal hyperparameters were determined on the validation set for each of the three random data splits that we used, for both applications. Each hyperparameter was searched independently, from 0.0 to 1.0, while fixing the parameters for other error types to 0.0.

### 6.4.3 Network Architecture

The baseline segmentation network $f_1$ is a 3D U-Net [153], shown in Figure 6.2. The label refinement network $f_2$ and the label appearance simulation network $f_\mathrm{a}$ use a similar U-Net layout, with the discriminator $D$ in $f_\mathrm{a}$ using the same layout as the U-Net encoder. The U-Net consists of an encoding path followed by a decoding path, with skip-connections linking the two paths. The network has 5 levels of depth, 16 feature channels in the first layer, and an input image size of 128×128×128. Each level of the encoding / decoding paths consists of two 3×3×3 convolutional layers followed by a 2×2×2 pooling or upsampling layer, respectively. Each convolutional layer consists of 3×3×3 convolution with zero padding followed by instance normalization and leaky ReLU activation. The number of feature channels is doubled or halved after every pooling or upsampling layer, respectively. The last layer of the U-Net is a 1×1×1 convolution, combining the outputs into a single feature map, followed by a sigmoid activation. A training batch contains only one image due to GPU memory limits. The networks are implemented using PyTorch [154]. The source code is publicly available: https://github.com/ShuaiChenBIGR/Label-refinement-network.



### 6.4.4 Details of Training and Inference of Networks

For training, we first apply random rigid transformations as data augmentation, in the form of 1) random 3D rotations up to 30 degrees for all axes, 2) random scaling with factor between 0.7–1.4 and 3) random flipping in the three directions. Then, we generate samples by extracting random image patches of size 128×128×128 on the fly from the input training images and corresponding ground truth segmentations. For the airway segmentation experiments, a lung mask is applied to the output of the network and the ground truth patches before computing the training loss. For this operation, we use a pre-computed lung mask that is easily obtained with a region growing algorithm [139]. During training, we used the Adam optimizer [155] with an initial learning rate of $1\times10^{-2}$. To train the refinement network $f_2$, the label $\tilde{x}$ in each training sample is randomly sampled with equal probability from either the initial segmentation $x_1$ or the synthetic label $x_\mathrm{a}$ after the label appearance simulation network.

During inference on new images, the input patches are extracted in a sliding-window fashion, with an overlap of 50% in the three directions. Then, the patch-wise predicted output by the network is aggregated by stitching the patches together, to reconstruct the full-size segmentation result. For the airway segmentation experiments, we applied a lung mask to the final segmentation to remove any spurious noise prediction outside the lungs. For this operation, we use the same region growing algorithm as during training.

For the adversarial loss in equation 6.2, the weight $\lambda$ is set to 0.01 for all experiments in this paper, based on visual inspection of the generated synthetic labels $x_{\mathrm{a}}$.

### 6.4.5 Comparisons

We compared the results of our proposed method with the baseline 3D U-Net segmentation network described in Section 6.4.3 (U-Net baseline). Additionally, we compared our method with two previous refinement approaches: DoubleU-Net [134] and SCAN [135]. For both baselines, we reimplemented the methods from the original papers. The DoubleU-Net method consists of two consecutive U-Nets, with skip connections from the encoder of the first U-Net to the decoders of both U-Nets. The SCAN method uses a U-Net with a discriminator and adversarial loss, discriminating between the segmentation results and the ground truth. The weight for balancing the segmentation loss and the adversarial loss (low value on the adversarial term) is tuned between 0.001 and 0.1, on the validation sets for each application. For DoubleU-Net, no additional hyperparameters need to be tuned. Our implementations of DoubleU-Net and SCAN use the same 3D U-Net backbone as our proposed method and the first baseline.

We also conducted an ablation study of the proposed method (LR+Syn+LASN) by removing some of the components. We evaluated 1) a simple label refinement method by inputting the original images and the initial segmentations without any synthetic errors (LR), 2) a label refinement method with synthetic errors added to the initial segmentations (LR+Syn(init)), and 3) a label refinement method with synthetic errors added to the ground truth segmentations but without the label appearance simulation network (LR+Syn).



### 6.4.6 Evaluation Metrics

We evaluated the methods with the Dice coefficient to measure the overall segmentation quality, as well as with three metrics designed for tree-like structures: centerline completeness, centerline leakage, and number of gaps. For the airway segmentation experiments, the required centerlines were obtained by applying a skeletonization method [156] to the ground truth segmentation mask. For the vessel segmentation experiments, the ground truth centerlines were manually annotated. The evaluation metrics are defined below:

*Dice coefficient* measures the voxelwise overlap between the predicted mask $Y$ and the ground truth mask $G$:

$$Dice = \frac{2|Y \cap G|}{|Y| + |G|} \tag{6.4}$$

*Centerline completeness* measures the proportion of the length of correctly detected centerlines (i.e., the intersection between the predicted mask $Y$ and the ground truth centerlines $G_{\mathrm{cl}}$) with respect to the length of ground truth centerlines $G_{\mathrm{cl}}$:

$$Completeness = \frac{|Y \cap G_{\mathrm{cl}}|}{|G_{\mathrm{cl}}|} \tag{6.5}$$

*Centerline leakage* measures the proportion of the length of false positive centerlines (i.e., the intersection between the predicted centerlines $Y_{\text{cl}}$ and the ground truth background $1 - G$) with respect to the length of ground truth centerlines $G_{\text{cl}}$:

$$Leakage = \frac{|Y_{\text{cl}} \cap (1 - G)|}{|G_{\text{cl}}|} \tag{6.6}$$

*Gaps* measures the number of continuity gaps in the correctly detected centerlines (i.e., the intersection between the predicted mask $Y$ and the ground truth centerlines $G_{\text{cl}}$). It is calculated with connected component analysis [157] as follows:

$$Gaps = NCC(Y \cap G_{\text{cl}}) - NCC(G_{\text{cl}}) \tag{6.7}$$

with $NCC$ counting the number of 26-neighbor-connected components in the input centerlines.

## 6.5 Results

### 6.5.1 Segmentation Results

***Table 6.1: Results for airway segmentation.*** *Average performance (standard deviation) over the results obtained from three random data splits. LR: simple label refinement network. LR+Syn(init): label refinement method with synthetic errors on initial segmentations. LR+Syn: label refinement method with synthetic errors on ground truth segmentations. LR+Syn+LASN: label refinement method with label appearance simulation network. ↑: significantly better than the U-Net baseline ($p < 0.05$). ↓: significantly worse than the U-Net baseline ($p < 0.05$). ⇑: significantly better than the proposed method ($p < 0.05$). P-values are calculated by the paired two-sided Student's T-test (on the average results from the three data splits). Boldface: best and not significantly different from the best results (semi-supervised results are not considered).*

| Methods | Dice | Completeness | Leakage | Gaps |
|---|---|---|---|---|
| U-Net baseline [3] | 0.76 (0.05) | 0.74 (0.12) | 0.23 (0.19) | 95.73 (47.94) |
| DoubleU-Net [134] | 0.77 (0.05)↑ | 0.73 (0.11) | 0.21 (0.18) | 99.93 (48.11) |
| SCAN [135] | 0.77 (0.05)↑ | 0.75 (0.11)↑ | 0.31 (0.23)↓ | 98.83 (48.81) |
| LR | 0.76 (0.05) | 0.74 (0.11)↑ | 0.23 (0.17) | 94.90 (47.66) |
| LR+Syn(init) | 0.77 (0.06) | 0.73 (0.12) | 0.19 (0.17) | 94.92 (50.14) |
| LR+Syn | **0.79** (0.05)↑ | 0.73 (0.12) | **0.17** (0.17)↑ | **93.54** (50.83)↑ |
| LR+Syn+LASN (proposed) | **0.79** (0.05)↑ | **0.75** (0.11)↑ | 0.20 (0.16)↑ | **91.63** (48.63)↑ |
| LR+Syn+LASN+Unlabeled | 0.81 (0.04)⇑ | 0.77 (0.10)⇑ | 0.19 (0.16)↑ | 90.53 (48.80)↑ |

The results of our experiments for airway and brain vessel segmentation are shown in Tables 6.1 and 6.2, respectively. In both applications, the proposed label refinement method achieves the highest Dice and completeness scores, the lowest number of gaps,

with a moderate leakage compared to the other methods. This indicates that our method succeeds in learning from the errors in the synthetic labels to correct errors in the real data. In both applications, the baselines with the highest completeness (SCAN for airways and DoubleU-Net for vessels) show a much higher leakage than our method. This indicates that these methods may lack the ability to learn relevant label structural information, and over-segment branches to increase the completeness rather than correcting errors in continuity.

In the ablation study, the label refinement method with synthetic errors (LR+Syn) achieves better Dice, leakage, and number of gaps scores than the baseline refinement network (LR), for both applications. For airway segmentation, the (LR+Syn) method has slightly lower completeness, while this is similar for vessel segmentation. Moreover, adding synthetic errors to the initial segmentations (LR+Syn(init)), in contrast to doing so to the ground truth segmentations (LR+Syn), achieves similar results in all metrics when compared to the baseline U-Net, for both applications. This suggests that the initial segmentations are too incomplete to add sufficient useful synthetic errors to train the refinement network. The proposed method, combining the synthetic errors and the label appearance simulation network (LR+Syn+LASN), achieves a much higher completeness, with similar Dice, leakage and number of gaps scores when compared to the method with only synthetic errors (LR+Syn), for both applications.



***Table 6.2: Results for brain vessel segmentation.*** *Average performance (standard deviation) over the results obtained from three random data splits. LR: simple label refinement network. LR+Syn(init): label refinement method with synthetic errors on initial segmentations. LR+Syn: label refinement method with synthetic errors on ground truth segmentations. LR+Syn+LASN: label refinement method with label appearance simulation network. ↑: significantly better than the U-Net baseline ($p < 0.05$). ↓: significantly worse than the U-Net baseline ($p < 0.05$). ⇑: significantly better than the proposed method ($p < 0.05$). P-values are calculated by the paired two-sided Student's T-test (on the average results from the three data splits). Boldface: best and not significantly different from the best results (semi-supervised results are not considered).*

| Methods | Dice | Completeness | Leakage | Gaps |
|---|---|---|---|---|
| U-Net baseline [4] | 0.57 (0.10) | 0.70 (0.18) | 0.19 (0.18) | 106.68 (161.41) |
| DoubleU-Net [134] | 0.59 (0.09)↑ | **0.73** (0.18)↑ | 0.18 (0.16) | 92.41 (151.27)↑ |
| SCAN [135] | 0.57 (0.09) | 0.70 (0.18) | 0.17 (0.15) | 104.05 (160.91) |
| LR | 0.57 (0.10) | 0.70 (0.18) | 0.16 (0.16)↑ | 82.05 (139.45)↑ |
| LR+Syn(init) | 0.58 (0.11) | 0.71 (0.19) | 0.18 (0.15) | 69.91 (126.89)↑ |
| LR+Syn | 0.60 (0.11)↑ | 0.71 (0.19) | **0.12** (0.11)↑ | 64.86 (115.21)↑ |
| LR+Syn+LASN (proposed) | **0.62** (0.10)↑ | **0.74** (0.20)↑ | 0.14 (0.11)↑ | **46.64** (76.57)↑ |
| LR+Syn+LASN+Unlabeled | 0.63 (0.09)⇑ | 0.75 (0.18)↑ | 0.13 (0.11)↑ | 42.45 (71.26)⇑ |

### 6.5.2 Semi-supervised Results

We conducted experiments using semi-supervised learning to train the proposed label refinement method, to investigate the benefit of using additional unlabeled data for training. As labels in which to synthesize errors for the unlabeled data, we used segmentation results on the same data obtained by the proposed method (LR+Syn+LASN) trained on the labeled data. We denote these results as "pseudo labels". The error generation in these pseudo labels follows the same strategy and hyperparameters as in the previous experiments (Sections 6.3.3 and 6.4.2). The pseudo labels are also used as ground truth segmentations for the unlabeled images. These unlabeled data together with the labeled data in the previous experiments are then used to train a new label refinement network.

The results of our semi-supervised experiments for airway and brain vessel segmentation are shown in the last rows in Tables 6.1 and 6.2, respectively. Adding unlabeled data for training significantly improves the Dice score while the leakage remains similar, for both applications. For airway segmentation, the completeness is also improved, while this is similar for vessel segmentation. This suggests that for vessels, the labeled data provides enough information for the method to obtain segmentations with good completeness. For vessels, the number of gaps is also improved.

### 6.5.3 Influence of the Synthetic Error Rate



The results of our experiments varying the maximum synthetic error rates (Section 6.4.2) are shown in Figure 6.4. For airway segmentation, with a smaller amount of "discontinuity" errors (0.1) the completeness is increased. Between 0.1 and 0.5, changing the amount of "discontinuity" errors in the synthetic labels does not affect much the method performance. In contrast, increasing the amount of "missing terminal branches" errors improves both Dice and completeness scores, reaching a peak when the maximum error rate is $\approx 0.75$. This supports our hypothesis that missing terminal branches are relevant errors to be corrected in the initial airway segmentations. For vessel segmentation, a moderate amount (0.6) of "discontinuity" errors has a positive effect in the method performance.

When compared to the LR+Syn and LR+Syn(init) methods, the proposed label refinement method is able to learn from higher amounts of synthetic errors, thereby improving the label refinement performance.

## 6.6 Discussion

In this paper, we propose a novel label refinement method that can correct errors in the initial segmentations from a standard deep segmentation network such as the U-Net. The novelty of our method is that it uses labels augmented with realistic, synthetic errors as training samples, from where the label refinement network can learn to correct the errors. The synthetic errors are automatically generated to simulate common errors observed in the initial segmentations, and are then refined by a label appearance simulation network to resemble the appearance of real errors in the initial segmentations.

We evaluated our method on the segmentation of airways from chest CT scans and brain vessels from CTA images of the brain. In both applications, our method achieved significantly higher Dice overlap and completeness scores, with lower number of gaps and a comparable leakage, when compared to the baseline U-Net and other previous label refinement methods. When segmenting branching structures, a higher completeness means that more and/or longer branches are detected, especially the smaller ones which are challenging to segment.

The ability of our method to segment highly complete tree-like structures with more branches is clinically important, as this could lead to more sensitive biomarkers. For example, in airway analysis, the airway-artery ratio [158] and airway tapering [159] measures can be used to assess cystic fibrosis lung disease, and including more measurements from the smaller peripheral branches can allow earlier detection of the disease [160]. Moreover, the ability of our method to correct errors in continuity and thereby connect the segmentation is beneficial, as most methods to measure branches assume a fully connected segmentation and discard branches after a discontinuity.

The proposed method outperformed the state-of-the-art label refinement methods DoubleU-Net [134] and SCAN [135]. Moreover, using semi-supervised learning techniques to train our method with additional unlabeled data we can further improve the method performance, when compared to the fully supervised setting.



### 6.6.1 Comparison to Other Label Refinement Methods

The main difference between the proposed method and other label refinement methods is the use of a training dataset that includes labels augmented with synthetic errors. Instead of synthetic errors, the DoubleU-Net [134] method uses the original images masked by the initial segmentations to train the second network. Although the increased model capacity of DoubleU-Net may improve the segmentations, its ability to correct the errors may be limited by the fact that no new errors are introduced to the input of the second network. This makes it less efficient to implicitly exploit the label structural information similar to a standard U-Net. The SCAN [135] method refines the segmentation by making it indistinguishable from the ground truth segmentation through an adversarial loss, where the distribution of the learned features may also provide the general structural information of the objects to be segmented. SCAN mainly focuses on simulating the appearance of the ground truth segmentations. However, SCAN is not designed to learn to correct structural errors explicitly, thus it may not capture the local continuity information as efficiently as our method. This is reflected by the significantly worse completeness reported by SCAN in Tables 6.1 and 6.2, for both applications. Our method provides an implicit way to enhance the network awareness of the structural information in the ground truth segmentations. For example, after seeing many errors in continuity, the refinement network is expected to understand the local continuity within elongated structures, and consequently to be able to correct these errors in the initial segmentations.

### 6.6.2 Synthetic Errors for Semi-supervised Learning

With the proposed method, synthetic errors can be added to any pseudo labels obtained on unlabeled data, to be used in semi-supervised learning. In Section 6.5.2 we have shown that our method performance was significantly improved when using additional unlabeled data for training. Our approach to generate synthetic errors could be used together with other common semi-supervised approaches using pseudo labels, e.g., to optimize the prediction consistency of the same image from different models [22], or the prediction consistency of the same image with different transformations [161]. Using synthetic errors in these methods may improve the segmentation quality of pseudo labels from the unlabeled data, which could provide more informative features from these data and thereby improve the method performance.

### 6.6.3 Importance of Realistic Synthetic Errors

The proposed label refinement network may underperform if the synthetic labels with errors used for training are too different from the initial segmentations. In our method, the synthetic errors are added to the ground truth segmentations, which have a fine and smooth appearance. In contrast, the initial segmentations are more irregular. Our proposed solution is to use a label appearance simulation network trained with an adversarial loss in order to make the appearance of the synthetic labels resemble that of the real initial segmentations. The results in Tables 6.1 and 6.2 clearly show the benefit of using the LASN network in our method. In both applications, without the LASN network could our method (LR+Syn) only slightly improve the segmentation performance with a reduced leakage, when compared to the baseline (LR). This may be due to the positive regularization effect of increasing the variety in the training data by including the synthetic labels with errors. Only after introducing the LASN network was our method able to improve the completeness while retaining an adequate leakage.



### 6.6.4 Applications to Other Segmentation Tasks

The proposed label refinement method via error synthesis can be applied to other segmentation tasks. The core step is to identify common types of errors in the initial segmentations. For example, a common error we observed in prior work using the U-Net for the segmentation of the aorta and pulmonary arteries from chest CT scans [162] was that the segmentation of one of the structures often leaked into the other one, while being both independent anatomical structures. This is mostly due to the obscured boundaries of both arteries on the CT scan. This type of error can be simulated by locally removing the boundaries between the aorta and pulmonary artery classes. Applying our method to correct such errors may improve the overall segmentation performance for this application.

### 6.6.5 Limitations of The Proposed Method

The main limitation of the proposed method lies in the two-step design and implementation: 1) analyze the errors in the initial segmentations to identify the relevant types

of errors, and 2) design and generate the synthetic errors based on these results. The first step requires observation and interpretation by the developers. The synthetic errors we used in this paper are suitable for the segmentation of tree-like structures. However, the relevant types of errors generally differ across different applications and datasets, and therefore the synthetic errors we used are not directly applicable to other segmentation tasks. The second step is typically a complex image processing task. Nevertheless, once the synthetic errors are successfully designed for a given application, the training of our label refinement method can be done fully automatically.

A limitation of our validation of the proposed method is that we considered only two types of false negative errors (i.e., missing terminal branches and errors in continuity). We did not consider false positive errors because these were much less frequent in the initial segmentations and often appeared as disconnected blobs that can be easily removed without the need for more complex label refinement. Nevertheless, from the results obtained in this paper we expect that our method can successfully correct other types of errors as well.

## 6.7 Conclusion

We presented a novel label refinement method that is able to learn from synthetic errors to refine the initial segmentations from a base segmentation network. A label appearance simulation network was applied to reduce the appearance difference between the synthetic labels and the real initial segmentations, thereby improving the generalizability of our method. On two segmentation tasks for branching structures, the proposed method achieved a significantly higher Dice overlap and centerline completeness, together with an improved continuity, when compared to previous label refinement methods. The segmentation performance of our method was further improved by using additional unlabeled data for training with semi-supervised learning techniques.



## 6.8 Acknowledgments

This work was partially funded by Chinese Scholarship Council (File No.201706170040), Netherlands Organisation for Scientific Research (NWO) project VI.C.182.042, the MARBLE project (EFRO/OP-Oost: PROJ-00887), the Contrast project (Dutch Heart Foundation (CVON2015-01: CONTRAST), the Brain Foundation Netherlands (HA2015.01.06) and additional funding by the Ministry of Economic Affairs by means of the PPP Allowance made available by the Top Sector Life Sciences & Health to stimulate public-private partnerships (LSHM17016)). We want to thank Thirona and H. Tiddens, M. van de Corput and M. Bonte (Erasmus MC - Lung Analysis group) for providing the airway segmentations and anonymous chest CT data used in this study.

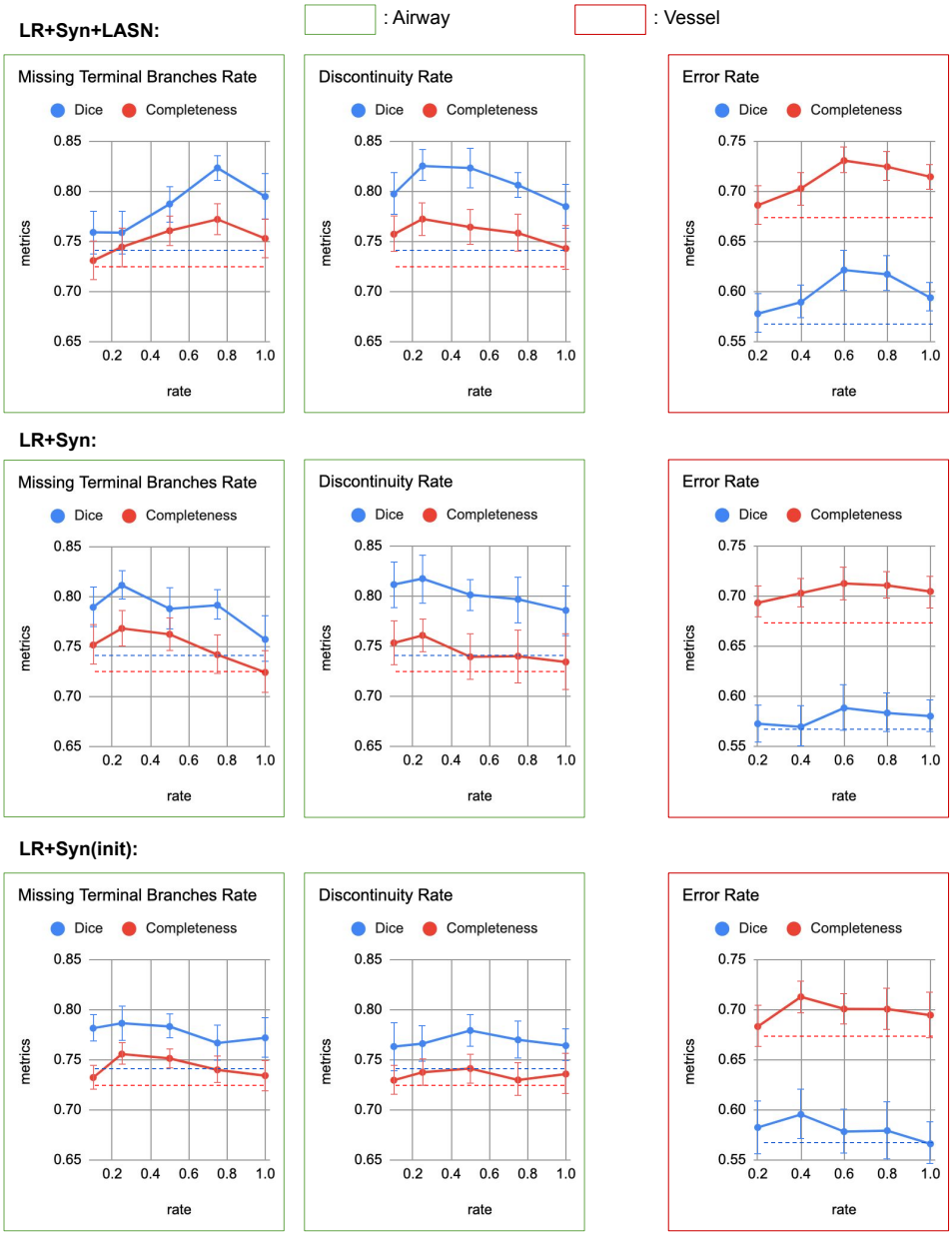



***Figure 6.4:*** ***Influence of the hyperparameters of the proposed method, the maximum synthetic error rates, in the method performance, for airway and brain vessel segmentation.*** *Results are shown as average performance with standard deviation (error bars), for Dice and completeness metrics, over three random data splits. The results for the baseline (LR) are displayed as dashed line. Better viewed in color.*

# Chapter 7

## Discussion

In this thesis, we developed advanced deep learning segmentation methods that incorporate global information to make more accurate segmentation decisions, and use additional unlabeled data to counter the scarcity of labeled data in medical imaging. To incorporate the global information into CNNs, two directions are explored: combining deep learning methods with traditional graph-based methods like conditional random fields (CRFs, Chapter 2), and introducing synthetic errors to enable the network to learn label information such as the topology of the objects to be segmented (Chapter 6). We also explore three semi-supervised learning approaches to use a combination of labeled and unlabeled data: combining segmentation and reconstruction to learn features that are more relevant for segmentation (Chapter 3), proposing new self-supervised learning methods that enable the network to learn without annotations (Chapter 4 and Chapter 5) and introducing synthetic errors into unlabeled data to further enhance the label refinement network (Chapter 6). This chapter summarizes the contributions of this thesis and discusses the limitations of the proposed methods, and possible directions for future research.

## 7.1 Global information in images, features, and labels

### 7.1.1 Is it worthwhile to do global learning in segmentation?

A good segmentation model is expected to provide accurate results based on all useful information, from both local and global scales within images. Local features such as texture are required to generate accurate target boundaries. The popular U-Net segmentation network relies primarily on those local features. However, these networks may sometimes generate errors that are difficult to solve by learning solely local features but would be easy to solve by employing global information on the predictions. For example, the consistent long-distance relationships between target objects may better reveal their locations than the local texture and shape features. Many approaches have been explored to learn and use such global information, such as Transformer networks. These methods make changes to the traditional CNN structure to allow it to learn global features, which increases the computational cost and requires more training data. In this thesis, we introduce the Posterior-CRF which is a less aggressive global learning method that preserves the traditional CNN structure as the main part of the network and only adds one learning-based CRF layer at the end. This not only enables the network to learn global features but also keeps the traditional convolution filters for efficient local feature extraction. In our experience, the local features are still the most important features for image segmentation. It may not be worthwhile to replace all conventional convolution structures with expensive global learning structures, because this would make the whole network much harder to train. In this sense, Posterior-CRF can serve as a functional and practical alternative to the GNNs and Transformer networks.

### 7.1.2 Advantages of using high-level features for global learning

Traditional CRF methods such as post-processing CRF methods usually use low-level features extracted from input images for the CRF inference, e.g. by encouraging

intensity similarity where similar classes of objects should have similar intensities. Encouraging the intensity similarity feature may improve segmentation performance in tasks when the objects to segment have similar intensity in the input images. However, it may fail in many cases, for example when different classes of objects share similar intensity (e.g., in aorta and pulmonary artery segmentation in CT images) or when objects in the same class have a more heterogeneous appearance (e.g., ischemic stroke lesions in MR images).

In Posterior-CRF, the CRF layer uses the learned features in the CNN feature maps for inference. Compared to the low-level features such as intensity, the CNN feature maps usually contain more high-level anatomical and semantic information. These features may better represent the characteristics of the objects and can be useful for a CRF model to perform more accurate inference. When connecting the Posterior-CRF layer and CNN network in an end-to-end manner, the CNN can learn useful high-level features for CRF inference on-the-fly. This combines the advantages of deep learning models and traditional CRFs and makes them better adapted to each other.

### 7.1.3 Exploiting label structure

While Posterior-CRF is able to extract useful global information from the intermediate feature maps, it does not exploit global structure in the ground-truth labels or the actual predictions from the model. This is similar to most other methods capable of modeling more global information, such as graph neural networks, Transformers, etc. In our experiments, we observed that these models sometimes make segmentation errors that are obvious to a human observer, such as discontinuities within airway and vessel tree segmentation. Some of these errors could be avoided if we take into account the global structure within the labels, e.g., by requiring that the airway branches and vessel branches are always continuous.



The label refinement method we propose in Chapter 6 is a solution to this problem. Instead of proposing new global learning architectures, we choose a more implicit way to make traditional networks be able to learn useful global information from labels: by designing and introducing synthetic errors to the ground truth labels as additional training data. The network is optimized using these additional data to learn to fix these synthetic errors and is expected to fix real errors made by traditional CNN models.

There are three main advantages of the proposed label refinement method compared to other global learning methods. First, it focuses on the global structural information within labels and predictions that may contain information that is beneficial to improving the structural coherence of segmentation. Second, we design errors based on intuitive observation, which makes it easier to apply to other applications. Third, the proposed method makes no significant change to the network architecture and keeps the network easy to train. These advantages make the proposed method an efficient way to exploit global information in labels and predictions in existing medical image segmentation solutions.

### 7.1.4 Can we apply the proposed global learning methods to other networks?

The proposed global learning methods in Chapter 2 and Chapter 6 can be easily applied to any popular segmentation network, such as U-Net, nnU-Net, etc. For Posterior-CRF, the CRF layer should follow the last convolution layer that has the same size as the predictions. The feature maps in the last convolution layer are used for the CRF inference to make the final predictions. It is relatively easy to apply the label refinement method to new segmentation networks since there is no need to change the network architectures. The only thing we need to do is to retrain the network using the additional data with synthetic errors. The easy plug-and-play property of the proposed methods increases their potential value to be applied in more advanced network architectures in the future.

## 7.2 Using unlabeled data

Semi-supervised learning and self-supervised learning are two popular research directions to use unlabeled data in medical image segmentation. In this thesis, we propose three different ways to use unlabeled data to help segmentation, including one semi-supervised method and two self-supervised methods. We also explore applying the label refinement method, which is proposed in Chapter 6 and focuses on global learning, to a semi-supervised setting. Significant improvements can be observed when training with additional unlabeled data in applications like brain tumor segmentation, white matter hyperintensities segmentation, airway segmentation, and vessel segmentation, which shows the efficiency of the proposed data-efficient methods.



### 7.2.1 Make reconstruction a better auxiliary task for segmentation

Our semi-supervised learning approach presented in Chapter 3 uses a reconstruction autoencoder to regularize the main segmentation task. The autoencoder can be trained together with a segmentation network using a shared encoder to regularize the segmentation task in a multi-task manner. Most previous methods use autoencoders to reconstruct the entire images, however, the learned features may not be the most relevant for the segmentation task. For example, in white matter hyperintensities segmentation, the lesions to segment are too small to meaningfully contribute to the reconstruction loss. The regular reconstruction task could then put more priority on reconstructing the large structures, such as the ventricles while ignoring the target lesions that are more relevant to the segmentation task. As a result, the learned features in an autoencoder may distract the segmentation network from learning proper features about the target lesions and thus limit the potential value of the autoencoder in regularizing the segmentation task.

To overcome this, we proposed to use an autoencoder to reconstruct the segmentation foreground (the lesions) and background (the rest of the brain) separately. In this way, the autoencoder would not only learn to reconstruct the entire image but also to distinguish the foreground and background, which may result in features that are more relevant to the segmentation task. We also provide insights in Table 3.3 that

the learned features in deeper layers of the proposed method are more relevant to the segmentation masks compared to the features in an autoencoder without the attention mechanism.

### 7.2.2 Two ways to improve inpainting-based self-supervised learning

We propose two self-supervised methods in Chapter 4 and Chapter 5. The first method in Chapter 4 is based on inpainting [24] and learns more segmentation-relevant features by masking and recovering only parts of the foreground area. The contents and boundaries of the foreground are key features to perform accurate segmentation. We also introduce a supervoxel technique for masking to further increase the shape diversity of the imaging areas to be recovered. The two techniques greatly boost the segmentation performance compared to self-supervision by inpainting randomly chosen areas.

Many inpainting-based approaches in the literature, as well as the proposed method in Chapter 4, focus on learning and recovering contents within a single image. Still, they do not consider variation between images or patients. However, this variation may contain potentially useful information, for example, because different patients may have similar anatomy. The method presented in Chapter 5 aims to learn such information by solving a source identification task. This source identification task is designed to learn distinguishable information across different sources (patients) to separate the target image from a synthetic image that mixes the target image with images from other sources. In our opinion, source identification is still an inpainting-based task, but very different from the traditional inpainting tasks since it requires information from both the target image and other images to inpaint the target itself. A model pretrained with source identification may therefore perform better at the downstream segmentation task.



### 7.2.3 Label refinement using unlabeled data

As discussed in Section 7.1.3, the proposed label refinement method in Chapter 6 focuses on learning global structural information from labels and predictions. In Section 6.5.2, we also show that it is possible to do global learning from unlabeled data. For semi-supervised methods that learn from intermediate predictions from unlabeled data such as self-training, synthetic errors can also be added to the intermediate predictions and served as additional training data. Potentially useful global information can be learned from unlabeled data predictions, although these predictions usually have lower accuracy compared to the ground truth labels as that used in a fully-supervised setting. This semi-supervised label refinement strategy can be combined with more advanced semi-supervised methods that use intermediate predictions of unlabeled data, such as mean-teacher [22], to achieve better segmentation performance.

## 7.3 Applications

All applications addressed in this thesis were only evaluated in a clean laboratory setting, which did not cover any real clinical research or application. In some applications such as aorta and pulmonary artery segmentation, the accuracy approaches human-level performance. However, the robustness of these methods to real data and reliability in clinical practice is still an open problem.

### 7.3.1 Suitable applications for the proposed methods

All methods in this thesis can be applied and generalized to other medical segmentation problems than the ones studied in this thesis. However, different methods may work better in different types of applications. For example, Posterior-CRF (Chapter 2) and source identification (Chapter 5) might be more useful for applications when local information is not sufficient to solve the problem and global information can be useful, like the global spatial relationship between targets may contribute to more accurate segmentation locations. MASSL (Chapter 3) and our new inpainting task (Chapter 4) might perform well in applications when the foreground and background are imbalanced, such as brain lesion segmentation, where a traditional reconstruction model may put more priority on reconstructing the background and ignore the foreground. The label refinement method (Chapter 6) might be suitable for applications when the regular CNN models make structural errors, such as continuous structures with erroneous gaps, neighboring objects with erroneously overlapped, or connected with each other, etc. It requires data with a clear structure in which realistic synthetic errors can be designed.



### 7.3.2 Proper task design for self-supervised methods

For self-supervised learning, it is important to choose the right self-supervision task and task parameters for each application, since this may dramatically affect the task difficulty and the learning efficiency of the network. Take the proposed inpainting or source identification tasks as an example: if the task is too easy to solve (small area to inpaint or no content overlap between different source images to separate), the network would learn trivial features that are less useful for the segmentation task. If the task is too difficult and impossible to solve (very large area to inpaint or indistinguishable overlap between source images to separate), the network would fail to converge. Therefore, it is crucial to design the task with proper difficulty and tune the hyperparameters for self-supervision methods every time when given a new dataset.

### 7.3.3 Application-specific error design for the label refinement method

The label refinement method in Chapter 6 depends on carefully designed synthetic errors, which are application-specific. It requires analysis of the initial segmentation results and design of the synthetic errors. The type of synthetic errors may vary depending on what label structural information we want the network to learn to

avoid the corresponding segmentation mistakes. For example, in aorta and pulmonary artery segmentation, the two types of vessels are sometimes misclassified as each other since they share very similar intensities in non-contrast CT images. In this case, the synthetic errors can be designed as misclassifications around the adjacency between the two vessels where the segmentation mistakes are usually made. The more realistic and representative the synthetic errors are, the more effective the label refinement method might be in new applications.

## 7.4 Computational complexity of the proposed methods

Advanced deep learning methods usually have an increasing computation cost, which may limit their usage in practice. The difficulty of balancing the tradeoff between the advanced modeling ability and the increasing computation cost may vary across the proposed methods.

Posterior-CRF could in principle be easy to apply to any other multi-class segmentation tasks. However, increasing the number of segmented classes exponentially increases the CRF inference time, which may make the method impractical when there are too many classes to be segmented. This is a common limitation of CRF-like methods and can be partly overcome by reducing the fully-connected CRF to a locally-connected CRF.

The computation time of the MASSL method presented in Chapter 3 is less affected when increasing the number of segmentation classes and the size of the last autoencoder layer. However, memory consumption can increase linearly. When determining the foreground and background area in multi-class segmentation, we consider each class separately. This results in reconstructing the corresponding foregrounds plus a combined background. Considering the fast development of GPU hardware, if GPU memory allows, a multi-class situation can be affordable.



The self-supervised learning methods presented in Chapter 4 and Chapter 5 are affected by the number of input modalities and the image size as well. The size of the input and output layers in both the region-of-interest guided supervoxel inpainting task and source identification task increases with the increase of the number of input modalities. In this thesis, there is one modality when using CT scans and less than four sequences when using multi-sequence MRI, which fits a common GPU with 8GB memory.

## 7.5 Limitations & Future Directions

This thesis contributed to the methodological development of advanced deep learning methods in medical image segmentation. In this section, I summarize the limitations of this thesis and discuss possible future directions for improvements and translation to clinical practice.

One limitation of the study is that most methods are only trained and evaluated in small datasets (usually less than 100 training and testing images). Although small datasets are quite common in medical imaging, the small test data reduces the confidence of the evaluation. The methods I investigated are designed to use limited

datasets optimally and are less important when training data is abundant. Collecting larger datasets and more manual annotations is still the easiest way to train a stronger deep learning model in most applications.

Another limitation is the transparency of the models. Deep learning-based methods are famous for their powerful feature learning ability but infamous for their black box properties. The features learned by the CNN models are abstract and difficult to explain, especially for the features in deeper layers. This may make it more difficult to trust deep learning in clinical practice. Pushing the modeling ability of deep learning to a higher level may make the models learn more complex features, while the learned features would become harder to explain at the same time. Explainable deep learning can be a direction to alleviate this problem. For example, we can visualize the gradients in each layer to investigate which features are more important in the decision-making of the model. This may reveal whether the learned features of the proposed methods cover the correct anatomical information.

Bias is a potential problem in machine learning approaches, and this may also happen in the proposed methods. The fairness of prediction can be affected by biases, e.g. gender, age, race, etc, although such biases can sometimes help to make more accurate predictions. A good model is expected to make decisions with consistent confidence and uncertainty for each subject. Thus, developing techniques such as feature decoupling for fairness is important for advanced deep learning methods. This can accelerate the translation to clinical practice and better integrate deep learning technology into society.

## 7.6 Conclusion



As a powerful and fast-evolving tool, deep learning shows its importance in medical imaging and achieves human-level performances in more and more applications. However, traditional deep learning tools may perform poorly in many practical circumstances, such as in segmentation tasks that require global information and with limited training data. In this thesis, I developed advanced deep learning segmentation methods for global and data-efficient learning to better generalize these scenarios. The methods were evaluated in several segmentation tasks in MRI, CT, and CTA, and presented better results in most applications compared to the state-of-the-art methods. When adding more unlabeled training data, significant improvements are observed in brain tumor segmentation, white matter hyperintensities segmentation, airway segmentation, and vessel segmentation, which shows the potential value of the proposed methods.

# Summary

Deep learning is a widely used tool in medical imaging, and it is especially effective in segmentation applications. However, the traditional deep learning methods for image segmentation can have problems in applications where they lack the ability to learn global information for accurate segmentation and the full annotations for medical imaging segmentation are scarcely available. This thesis provides solutions to these problems, based on methods for global and data-efficient learning. Developing methods that can learn and use more global information within images and labels can be beneficial if the learned information such as global spatial relationships between objects or the global structural information in labels help to improve the segmentation performance. Making better use of available unlabeled data may alleviate the data scarcity problem, which requires more data-efficient learning strategies.

**Chapter 1** introduces the background of deep learning in medical image segmentation and discusses why it is important to develop advanced global and data-efficient deep learning methods.

**Chapter 2** presents an end-to-end global deep learning algorithm called Posterior-CRF that uses CNN-learned features in conditional random fields (CRF) inference. As a traditional machine learning method, CRF can provide efficient global learning for CNN with limited additional computational cost. This method is validated on three medical segmentation tasks: aorta and pulmonary artery segmentation in non-contrast CT, white matter hyperintensities segmentation and ischemic stroke lesion segmentation in multi-modal MRI. The results show that Posterior-CRF achieves high accuracy and outperforms previous CNN-CRF methods with fixed features in all three segmentation tasks. Significant improvements are observed in aorta and white matter hyperintensities segmentation.

**Chapter3** presents a data-efficient learning method that uses an autoencoder to learn from unlabeled data. Unlike the traditional autoencoder that reconstructs the whole image, the proposed method reconstructs the foreground and background separately. The extracted features from the proposed method may be more relevant for the segmentation task than that from traditional autoencoders. The features learned are shared between segmentation and reconstruction, using the same encoder for both tasks. This method is validated on brain tumor segmentation and white matter hyperintensities segmentation in multi-modal MRI. The results show that the proposed method outperforms previous methods using traditional autoencoders. The learned features show more discriminative power for segmentation compared to the

features encoded by traditional autoencoders.

**Chapter 4** presents a method for self-supervision using region-of-interest guided supervoxel inpainting. Instead of inpainting random rectangular tiles, this method works on complete supervoxels in the segmentation foreground only, thus focusing the self-supervision on learning foreground features and predicting coherent regions. The method is validated in two applications, which are brain tumor segmentation and white matter hyperintensities segmentation in multi-modal MRI. The results show that in comparison to self-supervised learning using traditional inpainting, the two simple changes in the proposed method add a significant boost to the segmentation performance.

**Chapter 5** presents a new self-supervised learning task called Source Identification, which is inspired by the classic blind source separation problem. The task is to identify and separate a source image from a set of synthetic images, which mix the source image with images from other sources. Both local and more high-level, global features are required to separate the source image successfully. The method is validated in brain tumor segmentation and white matter hyperintensities segmentation in multi-modal MRI. In both applications, the proposed task achieves better downstream accuracy than other self-supervised learning approaches, including inpainting, pixel shuffling, intensity shift, and super-resolution.

**Chapter 6** presents a label refinement method that is able to correct errors in the initial segmentation results. Synthetic errors are generated in ground truth segmentations and an appearance simulation network is applied to ensure the appearance of the resulting labels resembles that of the real labels. A label refinement network is trained on both the synthetic and real labels to correct the errors. The method is validated in two tree-shaped structure segmentation tasks: lung airway segmentation in CT scans and brain vessel segmentation in CTA images. The results show that the proposed method significantly improves the continuity and completeness of the initial segmentation for both applications, and outperforms common segmentation and label refinement approaches.

**Chapter 7** summarizes the contributions of this thesis and provides a general discussion about the limitations of the proposed methods and possible directions for future research.

# Samenvatting

Deep learning is een veelgebruikte techniek voor medische beeldanalyse. De methodes zijn bijzonder geschikt voor segmentatietaken. De traditionele deep learning-methodes voor beeldsegmentatie missen echter de mogelijkheid voor het benutten van globale beeldinformatie, en zijn afhankelijk van de beschikbaarheid van voldoende volledig geannoteerde trainingsbeelden. Dit proefschrift biedt oplossingen voor deze twee beperkingen met nieuwe methodes voor globaal en data-efficiënt leren. Het gebruik van globale informatie kan de segmentatiekwaliteit verbeteren, bijvoorbeeld als deze extra inzicht geeft in de spatiële verbanden tussen objecten of in de globale structuur van de beelden. Data-efficiënte methoden zijn op hun beurt vereist om beter gebruik te kunnen maken van kleinere datasets en van niet-geannoteerde beelden.

**Hoofdstuk 1** introduceert de achtergrond van deep learning in medische beeldanalyse en bespreekt het belang van geavanceerde methodes voor globaal en data-efficiënt leren.

**Hoofdstuk 2** presenteert Posterior-CRF, een end-to-end deep learning-algoritme dat de feature learning van een CNN combineert met de globale inferentie van een conditional random field (CRF). Door het gebruik van een CRF kan het gecombineerde model op een computationeel efficiënte manier leren van globale informatie. De methode is getest met drie medische beeldsegmentatietaken: segmentatie van de aorta en longslagader in CT-beelden zonder contrastmiddel, segmentatie van witte stof hyperintensiteiten, en segmentatie van herseninfarcten in multi-modale MRI. Posterior-CRF levert zeer nauwkeurige resultaten en presteert in alle onderzochte taken beter dan voorgaande CNN-CRF-methodes met vaste features. Voor aorta en witte stof hyperintensiteiten zijn deze verschillen statistisch significant.

**Hoofdstuk 3** presenteert een data-efficiënte methode op basis van autoencoders voor het leren van niet-geannoteerde data. Anders dan een traditionele autoencoder, die een heel beeld tegelijk reconstrueert, reconstrueert de gepresenteerde methode de voorgrond en achtergrond afzonderlijk. De features die op deze manier worden verkregen zijn mogelijk meer relevant voor de segmentatietaak dan de features uit traditionele autoencoders. In de voorgestelde netwerkarchitectuur worden de features gedeeld door het segmentatie- en reconstructie-gedeelte van het model, die beide dezelfde encoder gebruiken. De methode is toegepast voor de segmentatie van hersentumoren en witte stof hyperintensiteiten in multi-modale MRI. De voorgestelde methode presteert beter dan de voorgaande methodes op basis van traditionele autoencoders. De features die worden geleerd bevatten daarnaast meer discriminatieve informatie voor segmentatie

dan features die worden geleerd door traditionele autoencoders.

**Hoofdstuk 4** presenteert een methode voor self-supervision op basis van inpainting. In plaats van inpainting van willekeurig gekozen rechthoeken, gebruikt deze methode inpainting uitsluitend voor complete supervoxels in de segmentatievoorgrond. Op deze manier wordt de self-supervision geconcentreerd op het leren van features van de voorgrond en op het voorspellen van coherente gebieden. De methode is vergeleken in twee toepassingen: segmentatie van hersentumoren en witte stof hyperintensiteiten in multi-modale MRI. In vergelijking met self-supervised learning op basis van traditionele inpainting leidt de voorgestelde methode tot significant betere segmentaties.

**Hoofdstuk 5** presenteert Source Identification, een nieuwe taak voor self-supervised learning geïnspireerd op het klassieke probleem van blind source separation. Het doel van deze taak is om een echt beeld te onderscheiden in een verzameling synthetische afbeeldingen, waarbij de synthetische beelden worden samengesteld door verschillende beelden uit verschillende bronnen te combineren. Om deze taak succesvol uit te kunnen voeren moet het model zowel lokale als globale features leren. De methode is geëvalueerd voor de segmentatie van hersentumoren en witte stof hyperintensiteiten in multi-modale MRI. Voor beide toepassingen bereikt de voorgestelde methode een hogere nauwkeurigheid dan andere methodes voor self-supervised learning, zoals inpainting, pixel shuffling, intensity shift en super resolution.

**Hoofdstuk 6** presenteert een methode voor label refinement waarmee fouten in initiële segmentaties kunnen worden verbeterd. De methode voegt synthetische fouten toe aan de ground-truth segmentaties en gebruikt vervolgens een appearance simulation network om te zorgen dat de synthetische labels lijken op echte labels. Het label refinement network wordt getraind op synthetische en echte labels om de fouten te verbeteren. The methode is geëvalueerd met twee segmentatietaken: luchtwegsegmentatie in CT-scans en bloedvaten in de hersenen in CTA-beelden. De voorgestelde methode leidt bij beide taken tot een significante verbetering in de continuïteit en volledigheid van de initiële segmentaties. De performance is ook beter dan die van de gebruikelijke methodes voor segmentatie en label refinement.

**Hoofdstuk 7** geeft een samenvatting van de bijdragen van dit proefschrift, bespreekt de beperkingen van de voorgestelde methodes, en presenteert mogelijke richtingen voor toekomstig onderzoek.

# Acknowledgements

It all feels like yesterday when I first came to the BIGR group in 2017 and had my first coffee with Marleen. My English at that time did not make for an impressive chat for us, but you knew how to teach me to do so: introducing Florian and Gijs to me. Thanks to our communicative Florian and intellectual Gijs, I had a great time in my first group discussion about my first project in my PhD: how to pronounce the name "Gijs" (whiteboard in use).

After that, I learned more pronunciations of more names, not only Dutch but also international ones. I really enjoyed and appreciated the international environment inside BIGR, which unites so many interesting people together for the same goal: pushing forward the medical imaging field. It has been an honor to contribute to this goal.

Before looking back at these interesting people I met during my PhD journey, I would like to first thank the members of the doctoral committee. Thank you for taking the time to read my thesis and attend my defense. You made me more confident in my work.

The first person I want to thank is Marleen. It has been a pleasant experience working with you. As my main supervisor, you feel more like a friend to me who always enjoys a philosophically whimsical chat and knows when to pull us back to the real world. We talked about many things: politics, culture, aesthetics, philosophy, and also our main topic, research. You showed me how to be a good researcher, with beliefs, and never let them solely exist in words.

Dutch is famous for its freedom-loving heart, and you showed me that as well in our weekly discussions. I always had mixed feelings when you start our conversation with the line: "what do you want to do next?" In my experience, this is not a common question I have been asked seriously before. Touching freedom can be joyful and painful at the same time, as people may easily get lost in carelessly coming up with new ideas but never realizing any of them. To my luck, every time I got lost, you were always able to pull me back and make me calm down again. Your love for freedom is like a heavy anchor, and only people with a strong spirit can throw it far away and never get lost. Thank you for taking me on the journey, and I look forward to more of our talks in the future.

Gijs (if I pronounce it correctly), thanks for being my co-supervisor. Your freelance and minimalist work style influenced me a lot. You enjoy researching and know how to throw the lances closer to the target with elegance. No matter when I and the other BIGR members need help, you always show us a kind heart and patience. It usually does not take much time to solve our problems, since you know machine learning and programming so well. To my end, I liked discussing my careless deep learning ideas with you and seeing where they can go. I found it most interesting when both of us got confused, which may be partly due to the annoying black-box nature of deep learning. Although you say that you are not really interested in medical imaging, I think your research taste may lead medical imaging to a higher level. I would always wonder what your next work will be like!

Wiro, thanks for leading me to the beautiful city of Rotterdam and the BIGR family. We did not talk much about research, but you showed me how many possibilities a researcher can have. Your attractive personality united BIGR together, and even the whole field. There is always something new in you to learn from!

Now, let's move to the front line of PhD, a place that is filled with successes and failures. I could have never accomplished my PhD without the model-based teammates, who are all brave warriors that are ready to face the unknown:

Florian, it will never be boring to have you stay with us. You seem to have some kind of magic to cheer up everything, even a chair can be convinced to sit more comfortably. I can not count how many jokes we made since we know each other, from Rotterdam to Shanghai and Beijing. We make a good team playing Dark Souls, where you control the directions and I do the attack and dodge (very difficult game indeed). You always bring new ideas and enjoy realizing them. I have to admit that it was really fun and cheerful to be your witness.

Richard, previously as Gerda, thanks for your help to improve my work. You are knowledgeable, and never hesitate to show your kindness to others. You have a world in mind which is far different from the world we are living in now. From our daily chatting, I can sometimes have a glimpse of the scenery of that world. I have to say I like what I see and would be happy to help you realize it.

Zahra, thanks for teaching me my first lesson on Mevislab and literature reviewing. You are very well-organized in working and optimistic in life, and also the bravest PhD student to organize the cruel paintball match for us!

Antonio, thanks for saving my programming skills. You are always friendly to others and willing to contribute. It has been nice to work with you and write a paper together. Also congratulations for your newborn boy David! Time for him to learn some python :).

Kim, thanks for your help in the PVS project. You did impressive work in organizing the Valdo Challenge at MICCAI 2021. Your efforts to research will definitely have a large impact on the BIGR group and the medical imaging field.

Deep, thanks for your fruitful deep learning ideas and literature sharing with the model-based group. You are a humble and dedicated person that I admire a lot. I look forward to more collaborations with you in the future!

Robin and Hoel, thanks for your programming lessons. You two are a good team to make excellent papers. I wonder how many secret skills you still have to improve a paper!

Annegreet, you are like a big sister to me who always stands in the office to work. The PhD experience you shared with us makes me think more about how to value work and life. Also, the famous octopus incident showed me that the methodologically serious model-based meeting can also be hilariously funny sometimes.

Outside of the model-based group, I would also like to thank my roommates:

Gennady, thanks for helping me when my Linux broke down as well as for the funny daily coffee chats. You showed me how to write big papers: by inputting hundreds of coauthors into the submitting system for the whole night!

Gokhan, thanks for sharing so many interesting youtube videos with us. I really enjoyed your teasing, letting me guess how you did image registration in one line!

Hua, thanks for your support in work and daily life. We can talk for hours about where artificial intelligence can go, although you do not really like the word AI. Your pragmatic opinions always make me reconsider the situation.

And to my Chinese friends, who are all proud of our best Chinese food:

Jiahang and Ruisheng, thanks for your homemade banquets. You are always willing to help others with your talents in creating a better living environment!

Bo, thanks for the BIGR touring together on the first day we came here. You can be a good scientist to whom I admire a lot!

Lau, thanks for taking me around Europe on your self-driving tour. You are a good drinking companion with lots of stories!

Yaoyao and Wenhao, thanks for inviting us to your home on Chinese New Year's day and for your working advice. You made me feel that the Spring Festival can also be nice even far away from China!

Also, to the traditional BIGR lunch group:

Gijs, Florian, Hua, Martijn, Karin, Wietske, Jose, Richard, Antonio, and so many others. Thanks to the BIGR lunch group where lots of shining sparks happen. There were so many hilarious discussions with you during these precious working breaks.

Great thanks to Sebastian, who loves games, and especially board games, for organizing the exciting board game nights. Also thanks for the PS5 from the black market. That really made it much easier for me to work from home during COVID time :).

Especially, to the supreme banana group:

Jose, Johnny, Maria, Marloes, Arno, Dirk, Florian, Alice. Thanks for taking me into the supreme banana group. You are like a family to me, who always support each other, share, and enjoy every interesting thing that happened on the earth. The days and nights we experienced together were definitely my happiest time in the Netherlands.

For the wider BIGR members:

Theo, Stefan, Jifke, Esther, Eirk, Hakim, Marcel, Petra, Desiree, Marise, Annemarijn, Danilo, Luisa, Jing, Yuanyuan, Haidong, Chaoping, Shengnan, Ivan, Henri, Ihor, Mart, Pierre, Riwaj, Douwe, Samantha, Roman, Vikram, Mohamed. Thanks for supporting me and the BIGR group.

Finally, I would like to thank my parents. You always care about my feelings and respect my choices, even the most naive ones. Because of you, I can stay relaxed and go to the next stage of life.

*Shuai Chen*
*Shanghai, October 2022*

# About the author

Shuai Chen was born in July 1991 in Beijing, China, and grew up in the ancient city of Baoding, in Hebei province. There he had a memorable childhood full of toys, video games, and fireworks. In primary school, he received from his friends one of his favorite books "*Brief History of Time*", written by the famous physicist Stephen Hawking. Being not a genius himself, Shuai only enjoyed the beautiful pictures in the book and did not understand a single word. In high school, he showed great interest in physics thanks to his physics teacher, and kept that interest until now.

From 2010 to 2017, Shuai did his bachelor's and master's degree in Geophysics at Jilin University, in the city of Changchun. During that time, he learned to do electromagnetic imaging of the earth and explored the natural processes underground using mathematical algorithms. At the same time, he learned about artificial intelligence (AI) and got struck by the fact that these methods could beat humans at the "Go" game. Therefore, he started self-studying AI and machine learning.

In September 2017, Shuai joined the Biomedical Imaging Group in Rotterdam (BIGR), at Erasmus Medical Center, in the Netherlands, to pursue his PhD degree, under the supervision of Marleen de Bruijne and Gijs van Tulder. During his PhD project, he developed advanced machine learning algorithms for medical image analysis, as presented in this thesis.

From 2021, Shuai continued as a PostDoc at the same BIGR group, working on weakly supervised perivascular space (PVS) segmentation in MR images.

# Publications

## Journal Papers

**S. Chen***, S. Kayal*, and M. de Bruijne, "Source identification: A self-supervision task for dense prediction", *Submitted*.

**S. Chen***, A. Garcia-Uceda*, J. Su*, G. van Tulder, L. Wolff, T. van Walsum, and M. de Bruijne, "Label refinement network from synthetic error augmentation for medical image segmentation", *Submitted*.

C. H. Sudre, K. van Wijnen, F. Dubost, H. Adams, D. Atkinson, F. Barkhof, M. A. Birhanu, E. E. Bron, R. Camarasa, N. Chaturvedi, Y. Chen, Z. Chen, **S. Chen**, Q. Dou, T. Evans, I. Ezhov, H. Gao, M. Girones Sanguesa, J. Domingo Gispert, B. Gomez Anson, A. D. Hughes, M. Arfan Ikram, S. Ingala, H. Rolf Jaeger, F. Kofler, H. J. Kuijf, D. Kutnar, M. Lee, B. Li, L. Lorenzini, B. Menze, J. Luis Molinuevo, Y. Pan, E. Puybareau, R. Rehwald, R. Su, P. Shi, L. Smith, T. Tillin, G. Tochon, H. Urien, B. H.M. van der Velden, I. F. van der Velpen, B. Wiestler, F. J. Wolters, P. Yilmaz, M. de Groot, M. W. Vernooij, M. de Bruijne, and for the ALFA study, "*Where is VALDO?* vascular lesions detection and segmentation challenge at MICCAI 2021", *Submitted*.

**S. Chen**, Z. Sedghi Gamechi, F. Dubost, G. van Tulder, and M. de Bruijne, "An end-to-end approach to segmentation in medical images with CNN and Posterior-CRF", *Medical Image Analysis*, vol. 76, p. 102 311, 2022. DOI: 10.1016/j.media.2021.102311.

## Conference Papers

S. Kayal, **S. Chen**, and M. de Bruijne, "Region-of-interest guided supervoxel inpainting for self-supervision", in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, 2020, pp. 500–509. DOI: 10.1007/978-3-030-59710-8_49.

**S. Chen**, G. Bortsova, A. Garcia-Uceda Juarez, G. van Tulder, and M. de Bruijne, "Multi-task attention-based semi-supervised learning for medical image segmentation", in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, 2019, pp. 457–465. DOI: 10.1007/978-3-030-32248-9_51.

## Conference Abstracts

**S. Chen** and M. de Bruijne, "An end-to-end approach with CNN and Posterior-CRF in white matter hyperintensities segmentation", in *WMH 2017 Challenge*, 2019.

**S. Chen** and M. de Bruijne, "An end-to-end approach to semantic segmentation with 3d CNN and Posterior-CRF in medical images", in *Medical Imaging meets NeurIPS Workshop, NeurIPS 2018*, 2018. DOI: 10.48550/arXiv.1811.03549.

* indicates equal contributions

# PhD portfolio

| **Courses** | **Year** | **ECTS** |
|---|---|---|
| NFBIA summer school<br>*Utrecht, The Netherlands* | 2017 | 2 |
| Front-End Vision & Multi Scale Image Analysis<br>*TU Eindhoven, The Netherlands* | 2017 | 4 |
| Scientific Integrity<br>*Erasmus MC, The Netherlands* | 2018 | 0.3 |
| Advanced Pattern Recognition<br>*Delft University of Technology, The Netherlands* | 2018 | 4 |
| Course Bayesian statistics and JASP<br>*Erasmus MC, The Netherlands* | 2018 | 2 |
| Medical Imaging Summer School<br>*Sicily, Italy* | 2018 | 4 |
| Scientific Visualization<br>*SURFsara, The Netherlands* | 2019 | 1 |
| Machine learning summer school<br>*Tübingen, Germany* | 2020 | 2 |
| Biomedical writing for PhD candidates<br>*Erasmus MC, The Netherlands* | 2021 | 3 |
| **Total** | | 22.3 |

| **International Conference and Workshop Attendance** | **Year** | **ECTS** |
|---|---|---|
| NVPHBV<br>*Eindhoven, The Netherlands* | 2017 | 0.5 |
| EuSoMII<br>*Erasmus MC, The Netherlands* | 2017 | 0.5 |
| MISP<br>*Erasmus MC, The Netherlands* | 2018 | 0.5 |
| NeurIPS<br>*Montreal, Canada* | 2018 | 2 |
| MICCAI<br>*Shenzhen, China* | 2019 | 2 |
| MIDL<br>*Montreal, Canada* | 2020 | 0.5 |
| MICCAI<br>*Lima, Peru* | 2020 | 1.2 |
| **Total** | | 7.2 |

| **Teaching activities** | **Year** | **ECTS** |
|---|---|---|
| Image Processing<br>*Erasmus MC, The Netherlands* | 2020 | 2 |
| Reproducing Deep Learning Results Course<br>*Delft University of Technology, The Netherlands* | 2021 | 2 |
| **Total** | | 4 |

| **Student supervision** | **Year** | **ECTS** |
|---|---|---|
| Supervision Master Thesis - Emmanuel Ahenkan<br>*Self-supervision via Triplet Loss in Medical Image Segmentation* | 2020 – 2021 | 1 |
| **Total** | | 1 |

| **Grants & Awards** | **Year** |
|---|---|
| Valdo Challenge<br>*First place in PVS segmentation, second place in overall tasks* | 2021 |
| Pilot grant SURFsara (500,000 billing units) | 2017 – 2021 |

| **Others** | **Year** | **ECTS** |
|---|---|---|
| Research lunch & BIGR seminar<br>*Presenter* | 2017 – 2021 | 2 |
| Medical imaging meets NeurIPS workshop<br>*Committee member* | 2019 | 0.5 |
| **Total** | | 2.5 |

# Bibliography


[1] 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.

[2] Z. S. Gamechi, A. M. Arias-Lorza, J. H. Pedersen, and M. de Bruijne, "Aorta and pulmonary artery segmentation using optimal surface graph cuts in non-contrast ct", in *Medical Imaging 2018: Image Processing*, SPIE, vol. 10574, 2018, pp. 616–622.

[3] A. Garcia-Uceda, R. Selvan, Z. Saghir, H. Tiddens, and M. de Bruijne, "Automatic airway segmentation from computed tomography using robust and efficient 3-d convolutional neural networks", *Scientific Reports*, vol. 11, no. 1, p. 16 001, 2021.

[4] J. Su, L. Wolff, A. C. M. van Es, W. van Zwam, C. Majoie, D. W. Dippel, A. van der Lugt, W. J. Niessen, and T. Van Walsum, "Automatic collateral scoring from 3d cta images", *IEEE transactions on medical imaging*, vol. 39, no. 6, pp. 2190–2200, 2020.

[5] N. Siddique, S. Paheding, C. P. Elkin, and V. Devabhaktuni, "U-net and its variants for medical image segmentation: A review of theory and applications", *Ieee Access*, vol. 9, pp. 82 031–82 057, 2021.

[6] F. Isensee, P. F. Jaeger, S. A. Kohl, J. Petersen, and K. H. Maier-Hein, "Nnu-net: A self-configuring method for deep learning-based biomedical image segmentation", *Nature methods*, vol. 18, no. 2, pp. 203–211, 2021.

[7] 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.

[8] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift", in *International conference on machine learning*, PMLR, 2015, pp. 448–456.

[9] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting", *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[10] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization", *arXiv preprint arXiv:1412.6980*, 2014.

[11] 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.

[12] 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.

[13] 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.

[14] P. Krähenbühl and V. Koltun, "Efficient inference in fully connected crfs with gaussian edge potentials", *Advances in neural information processing systems*, vol. 24, 2011.

[15] R. Selvan, M. Welling, J. H. Pedersen, J. Petersen, and M. d. Bruijne, "Mean field network based graph refinement with application to airway tree extraction", in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, 2018, pp. 750–758.

[16] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need", *Advances in neural information processing systems*, vol. 30, 2017.

[17] F. Shamshad, S. Khan, S. W. Zamir, M. H. Khan, M. Hayat, F. S. Khan, and H. Fu, "Transformers in medical imaging: A survey", *arXiv preprint arXiv:2201.09873*, 2022.

[18] A. Feragen, P. Lo, M. de Bruijne, M. Nielsen, and F. Lauze, "Toward a theory of statistical tree-shape analysis", *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 8, pp. 2008–2021, 2012.

[19] J. C. Paetzold, S. Shit, I. Ezhov, G. Tetteh, A. Ertürk, H. Z. Munich, and B. Menze, "Cldice—a novel connectivity-preserving loss function for vessel segmentation", in *Medical Imaging Meets NeurIPS 2019 Workshop*, 2019.

[20] S. Sedai, D. Mahapatra, S. Hewavitharanage, S. Maetschke, and R. Garnavi, "Semi-supervised segmentation of optic cup in retinal fundus images using variational autoencoder", in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, 2017, pp. 75–82.

[21] V. Cheplygina, M. de Bruijne, and J. P. Pluim, "Not-so-supervised: A survey of semi-supervised, multi-instance, and transfer learning in medical image analysis", *Medical image analysis*, vol. 54, pp. 280–296, 2019.

[22] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results", *Advances in neural information processing systems*, vol. 30, 2017.

[23] L. Jing and Y. Tian, "Self-supervised visual feature learning with deep neural networks: A survey", *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 11, pp. 4037–4058, 2020.

[24] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, "Context encoders: Feature learning by inpainting", in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2536–2544.

[25] A. J. Bell and T. J. Sejnowski, "An information-maximization approach to blind separation and blind deconvolution", *Neural computation*, vol. 7, no. 6, pp. 1129–1159, 1995.

[26] 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.

[27] 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.

[28] 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.

[29] 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.

[30] 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.

[31] 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.

[32] L. Yu, X. Yang, J. Qin, and P.-A. Heng, "3d fractalnet: Dense volumetric segmentation for cardiovascular mri volumes", in *Reconstruction, segmentation, and analysis of medical images*, Springer, 2016, pp. 103–110.

[33] 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.

[34] H. J. Kuijf, J. M. Biesbroek, J. de Bresser, R. Heinen, S. Andermatt, M. Bento, M. Berseth, M. Belyaev, M. J. Cardoso, A. Casamitjana, *et al.*, "Standardized assessment of automatic segmentation of white matter hyperintensities; results of the wmh segmentation challenge", *IEEE transactions on medical imaging*, 2019.

[35] O. Maier, M. Wilms, J. von der Gablentz, U. M. Krämer, T. F. Münte, and H. Handels, "Extra tree forests for sub-acute ischemic stroke lesion segmentation in mr sequences", *Journal of neuroscience methods*, vol. 240, pp. 89–100, 2015.

[36] 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.

[37] 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.

[38] Q. Dou, L. Yu, H. Chen, Y. Jin, X. Yang, J. Qin, and P.-A. Heng, "3d deeply supervised network for automated segmentation of volumetric medical images", *Medical image analysis*, vol. 41, pp. 40–54, 2017.

[39] 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.

[40] A. G. Schwing and R. Urtasun, "Fully connected deep structured networks", *arXiv preprint arXiv:1503.02351*, 2015.

[41] Z. Sedghi Gamechi, A. M. Arias-Lorza, J. H. Pedersen, and M. De Bruijne, "Aorta and pulmonary artery segmentation using optimal surface graph cuts in non-contrast ct", in *Medical Imaging 2018: Image Processing*, International Society for Optics and Photonics, vol. 10574, 2018, p. 105742D.

[42] Y. Xie, J. Padgett, A. M. Biancardi, and A. P. Reeves, "Automated aorta segmentation in low-dose chest ct images", *International journal of computer assisted radiology and surgery*, vol. 9, no. 2, pp. 211–219, 2014.

[43] R. Vemulapalli, O. Tuzel, M.-Y. Liu, and R. Chellapa, "Gaussian conditional random field network for semantic segmentation", in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 3224–3233.

[44] G. Lin, C. Shen, A. Van Den Hengel, and I. Reid, "Efficient piecewise training of deep structured models for semantic segmentation", in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 3194–3203.

[45] Y. Li and W. Ping, "Cancer metastasis detection with neural conditional random field", *arXiv preprint arXiv:1806.07064*, 2018.

[46] G. Wang, M. A. Zuluaga, W. Li, R. Pratt, P. A. Patel, M. Aertsen, T. Doel, A. L. David, J. Deprest, S. Ourselin, *et al.*, "Deepigeos: A deep interactive geodesic framework for medical image segmentation", *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 7, pp. 1559–1572, 2018.

[47] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model", *IEEE Transactions on Neural Networks*, vol. 20, no. 1, pp. 61–80, 2008.

[48] R. Selvan, M. Welling, J. H. Pedersen, J. Petersen, and M. de Bruijne, "Mean field network based graph refinement with application to airway tree extraction", in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, 2018, pp. 750–758.

[49] Z. Wang, N. Zou, D. Shen, and S. Ji, "Non-local u-nets for biomedical image segmentation.", in *AAAI*, 2020, pp. 6315–6322.

[50] 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.

[51] H. Yuan, N. Zou, S. Zhang, H. Peng, and S. Ji, "Learning hierarchical and shared features for improving 3d neuron reconstruction", in *2019 IEEE International Conference on Data Mining (ICDM)*, IEEE, 2019, pp. 806–815.

[52] F. Isensee, P. F. Jaeger, S. A. Kohl, J. Petersen, and K. H. Maier-Hein, "Nnu-net: A self-configuring method for deep learning-based biomedical image segmentation", *Nature Methods*, pp. 1–9, 2020.

[53] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization", *arXiv preprint arXiv:1412.6980*, 2014.

[54] M. Monteiro, M. A. Figueiredo, and A. L. Oliveira, "Conditional random fields as recurrent neural networks for 3d medical imaging segmentation", *arXiv preprint arXiv:1807.07464*, 2018.

[55] A. Adams, J. Baek, and M. A. Davis, "Fast high-dimensional filtering using the permutohedral lattice", in *Computer Graphics Forum*, Wiley Online Library, vol. 29, 2010, pp. 753–762.

[56] J. H. Pedersen, H. Ashraf, A. Dirksen, K. Bach, H. Hansen, P. Toennesen, H. Thorsen, J. Brodersen, B. G. Skov, M. Døssing, *et al.*, "The danish randomized lung cancer ct screening trial—overall design and results of the prevalence round", *Journal of Thoracic Oncology*, vol. 4, no. 5, pp. 608–614, 2009.

[57] 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.

[58] Y. Li and R. Zemel, "Mean-field networks", *arXiv preprint arXiv:1410.5884*, 2014.

[59] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions", in *Advances in Neural Information Processing Systems*, 2017, pp. 4765–4774.

[60] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system", in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, ACM, 2016, pp. 785–794.

[61] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization", *arXiv preprint arXiv:1412.6980*, 2014.

[62] A. Rasmus, M. Berglund, M. Honkala, H. Valpola, and T. Raiko, "Semi-supervised learning with ladder networks", in *NeurIPS*, 2015, pp. 3546–3554.

[63] S. Sedai, D. Mahapatra, S. Hewavitharanage, S. Maetschke, and R. Garnavi, "Semi-supervised segmentation of optic cup in retinal fundus images using variational autoencoder", in *MICCAI*, Springer, 2017, pp. 75–82.

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

[65] J. Schlemper, O. Oktay, M. Schaap, M. Heinrich, B. Kainz, B. Glocker, and D. Rueckert, "Attention gated networks: Learning to leverage salient regions in medical images", *Medical Image Analysis*, 2019.

[66] X. You, Q. Peng, Y. Yuan, Y.-m. Cheung, and J. Lei, "Segmentation of retinal blood vessels using the radial projection and semi-supervised approach", *Pattern Recognition*, vol. 44, no. 10-11, pp. 2314–2324, 2011.

[67] Y. Xia, F. Liu, D. Yang, J. Cai, L. Yu, Z. Zhu, D. Xu, A. Yuille, and H. Roth, "3d semi-supervised learning with uncertainty-aware multi-view co-training", *arXiv preprint arXiv:1811.12506*, 2018.

[68] L. Zhou, Z. Zhong, A. Shah, and X. Wu, "3-d surface segmentation meets conditional random fields", *arXiv preprint arXiv:1906.04714*, 2019.

[69] 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)", *TMI*, vol. 34, no. 10, pp. 1993–2024, 2015.

[70] X.-Y. Zhou and G.-Z. Yang, "Normalization in training u-net for 2d biomedical semantic segmentation", *IEEE Robotics and Automation Letters*, 2019.

[71] A. Myronenko, "3d mri brain tumor segmentation using autoencoder regularization", in *International MICCAI Brainlesion Workshop*, Springer, 2018, pp. 311–320.

[72] H. Li, G. Jiang, J. Zhang, R. Wang, Z. Wang, W.-S. Zheng, and B. Menze, "Fully convolutional network ensembles for white matter hyperintensities segmentation in mr images", *NeuroImage*, vol. 183, pp. 650–665, 2018.

[73] L. Jing and Y. Tian, "Self-supervised visual feature learning with deep neural networks: A survey", *CoRR*, vol. abs/1902.06162, 2019. arXiv: 1902.06162.

[74] P. Zhang, F. Wang, and Y. Zheng, "Self supervised deep representation learning for fine-grained body part recognition", in *2017 IEEE 14th International Symposium on Biomedical Imaging (ISBI 2017)*, 2017, pp. 578–582.

[75] H. Spitzer, K. Kiwitz, K. Amunts, S. Harmeling, and T. Dickscheid, "Improving cytoarchitectonic segmentation of human brain areas with self-supervised siamese networks", in *Medical Image Computing and Computer Assisted Intervention - MICCAI 2018 - 21st International Conference, Granada, Spain, September 16-20, 2018, Proceedings, Part III*, 2018, pp. 663–671.

[76] L. Chen, P. Bentley, K. Mori, K. Misawa, M. Fujiwara, and D. Rueckert, "Self-supervised learning for medical image analysis using image context restoration", *Medical image analysis*, vol. 58, p. 101 539, 2019.

[77] X. Zhuang, Y. Li, Y. Hu, K. Ma, Y. Yang, and Y. Zheng, "Self-supervised feature learning for 3d medical images by playing a rubik's cube", in *Medical Image Computing and Computer Assisted Intervention - MICCAI 2019 - 22nd International Conference, Shenzhen, China, October 13-17, 2019, Proceedings, Part IV*, 2019, pp. 420–428.

[78] Z. Zhou, V. Sodha, M. M. Rahman Siddiquee, R. Feng, N. Tajbakhsh, M. B. Gotway, and J. Liang, "Models genesis: Generic autodidactic models for 3d medical image analysis", in *International conference on medical image computing and computer-assisted intervention*, Springer, 2019, pp. 384–393.

[79] D. Pathak, P. Krähenbühl, J. Donahue, T. Darrell, and A. A. Efros, "Context encoders: Feature learning by inpainting", in *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, 2016, pp. 2536–2544.

[80] P. Lo, J. Sporring, H. Ashraf, J. J. Pedersen, and M. de Bruijne, "Vessel-guided airway tree segmentation: A voxel classification approach", *Medical image analysis*, vol. 14, no. 4, pp. 527–538, 2010.

[81] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "Slic superpixels compared to state-of-the-art superpixel methods", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, pp. 2274–2282, 2012.

[82] 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.

[83] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization", in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2015.

[84] H. J. Kuijf, J. M. Biesbroek, J. De Bresser, R. Heinen, S. Andermatt, M. Bento, M. Berseth, M. Belyaev, M. J. Cardoso, A. Casamitjana, *et al.*, "Standardized assessment of automatic segmentation of white matter hyperintensities and results of the wmh segmentation challenge", *IEEE Transactions on Medical Imaging*, vol. 38, no. 11, pp. 2556–2568, 2019.

[85] G. v. Tulder and M. d. Bruijne, "Why does synthesized data improve multi-sequence classification?", in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, 2015, pp. 531–538.

[86] 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, 2015.

[87] I. Loshchilov and F. Hutter, "SGDR: stochastic gradient descent with warm restarts", in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017.

[88] A. Blum and T. Mitchell, "Combining labeled and unlabeled data with co-training", in *Proceedings of the Eleventh Annual Conference on Computational Learning Theory*, COLT' 98, Madison, Wisconsin, USA: Association for Computing Machinery, 1998, pp. 92–100, ISBN: 1581130570.

[89] L. Jing and Y. Tian, "Self-supervised visual feature learning with deep neural networks: A survey.", *CoRR*, vol. abs/1902.06162, 2019.

[90] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space", *arXiv preprint arXiv:1301.3781*, 2013.

[91] R. Kiros, Y. Zhu, R. Salakhutdinov, R. S. Zemel, A. Torralba, R. Urtasun, and S. Fidler, "Skip-thought vectors", *arXiv preprint arXiv:1506.06726*, 2015.

[92] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding", *arXiv preprint arXiv:1810.04805*, 2018.

[93] G. Larsson, M. Maire, and G. Shakhnarovich, "Colorization as a proxy task for visual understanding", in *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, IEEE Computer Society, 2017, pp. 840–849. DOI: 10.1109/CVPR.2017.96.

[94] C. Ledig, L. Theis, F. Huszár, *et al.*, "Photo-realistic single image super-resolution using a generative adversarial network", in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 105–114. DOI: 10.1109/CVPR.2017.19.

[95] C. Doersch, A. Gupta, and A. A. Efros, "Unsupervised visual representation learning by context prediction", in *International Conference on Computer Vision (ICCV)*, 2015.

[96] M. Noroozi and P. Favaro, "Unsupervised learning of visual representations by solving jigsaw puzzles", in *ECCV*, 2016.

[97] Z. Zhou, V. Sodha, M. M. Rahman Siddiquee, R. Feng, N. Tajbakhsh, M. B. Gotway, and J. Liang, "Models genesis: Generic autodidactic models for 3d medical image analysis", in *Medical Image Computing and Computer Assisted Intervention – MICCAI 2019*, D. Shen, T. Liu, T. M. Peters, L. H. Staib, C. Essert, S. Zhou, P.-T. Yap, and A. Khan, Eds., Cham: Springer International Publishing, 2019, pp. 384–393, ISBN: 978-3-030-32251-9.

[98] L. Chen, P. Bentley, K. Mori, K. Misawa, M. Fujiwara, and D. Rueckert, "Self-supervised learning for medical image analysis using image context restoration", *Medical Image Analysis*, vol. 58, p. 101 539, 2019.

[99] S. Kayal, S. Chen, and M. de Bruijne, "Region-of-interest guided supervoxel inpainting for self-supervision", in *Medical Image Computing and Computer Assisted Intervention - MICCAI 2020 - 23rd International Conference, Lima, Peru, October 4-8, 2020, Proceedings, Part I*, A. L. Martel, P. Abolmaesumi, D. Stoyanov, D. Mateus, M. A. Zuluaga, S. K. Zhou, D. Racoceanu, and L. Joskowicz, Eds., Lecture Notes in Computer Science, vol. 12261, Springer, 2020, pp. 500–509. DOI: 10.1007/978-3-030-59710-8_49.

[100] S. Becker and G. E. Hinton, "A self-organizing neural network that discovers surfaces in random-dot stereograms", *Nature*, vol. 355, pp. 161–163, 1992.

[101] J. Jiao, Y. Cai, M. Alsharid, L. Drukker, A. T. Papageorghiou, and J. A. Noble, "Self-supervised contrastive video-speech representation learning for ultrasound", in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, 2020, pp. 534–543.

[102] J. Li, G. Zhao, Y. Tao, P. Zhai, H. Chen, H. He, and T. Cai, "Multi-task contrastive learning for automatic ct and x-ray diagnosis of covid-19", *Pattern Recognition*, vol. 114, p. 107 848, 2021.

[103] K. Chaitanya, E. Erdil, N. Karani, and E. Konukoglu, "Contrastive learning of global and local features for medical image segmentation with limited annotations", *arXiv preprint arXiv:2006.10511*, 2020.

[104] C. Feng, C. Vanderbilt, and T. Fuchs, "Nuc2vec: Learning representations of nuclei in histopathology images with contrastive loss", in *Medical Imaging with Deep Learning*, 2021.

[105] H. Li, X. Yang, J. Liang, W. Shi, C. Chen, H. Dou, R. Li, R. Gao, G. Zhou, J. Fang, *et al.*, "Contrastive rendering for ultrasound image segmentation", in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, 2020, pp. 563–572.

[106] T. Chen, S. Kornblith, M. Norouzi, and G. E. Hinton, "A simple framework for contrastive learning of visual representations", 2020.

[107] S.-i. Amari, A. Cichocki, H. H. Yang, *et al.*, "A new learning algorithm for blind signal separation", in *Advances in neural information processing systems*, Morgan Kaufmann Publishers, 1996, pp. 757–763.

[108] A. Hyvarinen, "Blind source separation by nonstationarity of variance: A cumulant-based approach", *IEEE transactions on neural networks*, vol. 12, no. 6, pp. 1471–1474, 2001.

[109] S. Choi, A. Cichocki, H.-M. Park, and S.-Y. Lee, "Blind source separation and independent component analysis: A review", *Neural Information Processing-Letters and Reviews*, vol. 6, no. 1, pp. 1–57, 2005.

[110] T. Isomura and T. Toyoizumi, "A local learning rule for independent component analysis", *Scientific reports*, vol. 6, no. 1, pp. 1–17, 2016.

[111] L. Drude, T. von Neumann, and R. Haeb-Umbach, "Deep attractor networks for speaker re-identification and blind source separation", in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2018, pp. 11–15.

[112] L. Drude and R. Haeb-Umbach, "Integration of neural networks and probabilistic spatial models for acoustic blind source separation", *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 4, pp. 815–826, 2019.

[113] P. Chandna, M. Miron, J. Janer, and E. Gómez, "Monoaural audio source separation using deep convolutional neural networks", in *International conference on latent variable analysis and signal separation*, Springer, 2017, pp. 258–266.

[114] J. R. Hershey, Z. Chen, J. Le Roux, and S. Watanabe, "Deep clustering: Discriminative embeddings for segmentation and separation", in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2016, pp. 31–35.

[115] C. Tian, L. Fei, W. Zheng, Y. Xu, W. Zuo, and C.-W. Lin, "Deep learning on image denoising: An overview", *Neural Networks*, 2020.

[116] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "Mixup: Beyond empirical risk minimization", *International Conference on Learning Representations*, 2018.

[117] Z. Eaton-Rosen, F. J. S. Bragman, S. Ourselin, and M. J. Cardoso, "Improving data augmentation for medical image segmentation", *Medical Imaging with Deep Learning*, 2018.

[118] S. Shurrab and R. Duwairi, *Self-supervised learning methods and applications in medical imaging analysis: A survey*, 2021. arXiv: 2109.08685.

[119] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, "Image inpainting", in *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '00, USA: ACM Press/Addison-Wesley Publishing Co., 2000, pp. 417–424, ISBN: 1581132085.

[120] C. Zhao, B. E. Dewey, D. L. Pham, P. A. Calabresi, D. S. Reich, and J. L. Prince, "Smore: A self-supervised anti-aliasing and super-resolution algorithm for mri using deep learning", *IEEE transactions on medical imaging*, 2020.

[121] M. E. Mortenson, *Mathematics for Computer Graphics Applications: An Introduction to the Mathematics and Geometry of CAD/Cam, Geometric Modeling, Scientific Visualizati*, 2nd. USA: Industrial Press, Inc., 1999, ISBN: 083113111X.

[122] S. Bakas, H. Akbari, A. Sotiras, M. Bilello, M. Rozycki, J. S. Kirby, J. B. Freymann, K. Farahani, and C. Davatzikos, "Advancing the cancer genome atlas glioma mri collections with expert segmentation labels and radiomic features", *Scientific data*, vol. 4, p. 170 117, 2017.

[123] 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.

[124] F. Isensee, J. Petersen, A. Klein, *et al.*, "Nnu-net: Self-adapting framework for u-net-based medical image segmentation", *CoRR*, vol. abs/1809.10486, 2018. arXiv: 1809.10486.

[125] K. He, X. Zhang, S. Ren, and J. Sun, *Deep residual learning for image recognition*, 2015. arXiv: 1512.03385.

[126] X. Liu, A. Sinha, M. Ishii, G. D. Hager, A. Reiter, R. H. Taylor, and M. Unberath, "Dense depth estimation in monocular endoscopy with self-supervised learning methods", *IEEE transactions on medical imaging*, vol. 39, no. 5, pp. 1438–1447, 2019.

[127] G. Balakrishnan, A. Zhao, M. R. Sabuncu, J. Guttag, and A. V. Dalca, "An unsupervised learning model for deformable medical image registration", in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 9252–9260.

[128] K. M. H. van Wijnen, F. Dubost, P. Yilmaz, M. A. Ikram, W. J. Niessen, H. Adams, M. W. Vernooij, and M. de Bruijne, "Automated lesion detection by regressing intensity-based distance with a neural network", in *Medical Image Computing and Computer Assisted Intervention – MICCAI 2019*, D. Shen, T. Liu, T. M. Peters, L. H. Staib, C. Essert, S. Zhou, P.-T. Yap, and A. Khan, Eds., Cham: Springer International Publishing, 2019, pp. 234–242, ISBN: 978-3-030-32251-9.

[129] J.-C. Su, S. Maji, and B. Hariharan, "When does self-supervision improve few-shot learning?", in *European Conference on Computer Vision*, Springer, 2020, pp. 645–666.

[130] Y. You, T. Chen, Z. Wang, and Y. Shen, "When does self-supervision help graph convolutional networks?", in *International Conference on Machine Learning*, PMLR, 2020, pp. 10 871–10 880.

[131] 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.

[132] 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.

[133] N. Siddique, S. Paheding, C. P. Elkin, and V. Devabhaktuni, "U-net and its variants for medical image segmentation: A review of theory and applications", *IEEE Access*, vol. 9, pp. 82 031–82 057, 2021. DOI: 10.1109/ACCESS.2021.3086020.

[134] D. Jha, M. A. Riegler, D. Johansen, P. Halvorsen, and H. D. Johansen, "Doubleu-net: A deep convolutional neural network for medical image segmentation", in *2020 IEEE 33rd International Symposium on Computer-Based Medical Systems (CBMS)*, IEEE, 2020, pp. 558–564.

[135] W. Dai, N. Dong, Z. Wang, X. Liang, H. Zhang, and E. P. Xing, "Scan: Structure correcting adversarial network for organ segmentation in chest x-rays", in *Deep learning in medical image analysis and multimodal learning for clinical decision support*, Springer, 2018, pp. 263–273.

[136] Y. Yang, Z. Wang, J. Liu, K.-T. Cheng, and X. Yang, "Label refinement with an iterative generative adversarial network for boosting retinal vessel segmentation", *arXiv preprint arXiv:1912.02589*, 2019.

[137] R. J. Araújo, J. S. Cardoso, and H. P. Oliveira, "A deep learning design for improving topology coherence in blood vessel segmentation", in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, 2019, pp. 93–101.

[138] M. Graham, J. Gibbs, D. Cornish, and W. Higgins, "Robust 3-D airway tree segmentation for image-guided peripheral bronchoscopy", *IEEE Transactions on Medical Imaging*, vol. 29, no. 4, pp. 982–997, 2010.

[139] P. Lo, J. Sporring, H. Ashraf, J. Pedersen, and M. de Bruijne, "Vessel-guided airway tree segmentation: A voxel classification approach", *Medical image analysis*, vol. 14, no. 4, pp. 527–538, 2010.

[140] P. Lo, J. Sporring, J. Pedersen, and M. de Bruijne, "Airway tree extraction with locally optimal paths", *Medical Image Computing and Computer-Assisted Intervention MICCAI*, pp. 51–58, 2009.

[141] P. Lo, B. van Ginneken, J. Reinhardt, *et al.*, "Extraction of airways from CT (EXACT'09)", *IEEE Transactions on Medical Imaging*, vol. 31, no. 11, pp. 2093–2107, 2012.

[142] Y. Qin, H. Zheng, Y. Gu, X. Huang, J. Yang, L. Wang, F. Yao, Y. Zhu, and G. Yang, "Learning tubule-sensitive CNNs for pulmonary airway and artery-vein segmentation in CT", *IEEE Transactions on Medical Imaging*, vol. 40, no. 6, pp. 1603–1617, 2021.

[143] G. Cheng, X. Wu, W. Xiang, C. Guo, H. Ji, and L. He, "Segmentation of the airway tree from chest CT using tiny atrous convolutional network", *IEEE Access*, vol. 9, pp. 33 583–33 594, 2021. DOI: 10.1109/ACCESS.2021.3059680.

[144] H. Zheng, Y. Qin, Y. Gu, F. Xie, J. Sun, J. Yang, and G. Yang, "Refined local-imbalance-based weight for airway segmentation in CT", in *Medical Image Computing and Computer Assisted Intervention - MICCAI 2021*, 2021, pp. 410–419. DOI: 10.1007/978-3-030-87193-2_39.

[145] P. Sanchesa, C. Meyer, V. Vigon, and B. Naegel, "Cerebrovascular network segmentation of mra images with deep learning", in *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*, IEEE, 2019, pp. 768–771.

[146] M. Livne, J. Rieger, O. U. Aydin, A. A. Taha, E. M. Akay, T. Kossen, J. Sobesky, J. D. Kelleher, K. Hildebrand, D. Frey, *et al.*, "A u-net deep learning framework for high performance vessel segmentation in patients with cerebrovascular disease", *Frontiers in neuroscience*, vol. 13, p. 97, 2019.

[147] A. Hilbert, V. I. Madai, E. M. Akay, O. U. Aydin, J. Behland, J. Sobesky, I. Galinovic, A. A. Khalil, A. A. Taha, J. Wuerfel, *et al.*, "Brave-net: Fully automated arterial brain vessel segmentation in patients with cerebrovascular disease", *Frontiers in Artificial Intelligence*, vol. 3, p. 78, 2020.

[148] M. Meijs, A. Patel, S. C. van de Leemput, M. Prokop, E. J. van Dijk, F.-E. de Leeuw, F. J. Meijer, B. van Ginneken, and R. Manniesing, "Robust segmentation of the full cerebral vasculature in 4d ct of suspected stroke patients", *Scientific reports*, vol. 7, no. 1, pp. 1–12, 2017.

[149] N. Bouma, H. Janssens, E. Andrinopoulou, and H. Tiddens, "Airway disease on chest computed tomography of preschool children with cystic fibrosis is associated with school-age bronchiectasis", *Pediatric Pulmonology*, vol. 55, no. 1, pp. 141–148, 2020. DOI: https://doi.org/10.1002/ppul.24498.

[150] I. G. Jansen, M. J. Mulder, and R.-J. B. Goldhoorn, "Endovascular treatment for acute ischaemic stroke in routine clinical practice: Prospective, observational cohort study (mr clean registry)", *bmj*, vol. 360, 2018.

[151] D. Rodriguez-Luna, D. Dowlatshahi, R. I. Aviv, C. A. Molina, Y. Silva, I. Dzialowski, C. Lum, A. Czlonkowska, J.-M. Boulanger, C. S. Kase, *et al.*, "Venous phase of computed tomography angiography increases spot sign detection, but intracerebral hemorrhage expansion is greater in spot signs detected in arterial phase", *Stroke*, vol. 45, no. 3, pp. 734–739, 2014.

[152] R. Peter, B. J. Emmer, A. C. van Es, and T. van Walsum, "Cortical and vascular probability maps for analysis of human brain in computed tomography images", in *2017 IEEE 14th International Symposium on Biomedical Imaging (ISBI 2017)*, IEEE, 2017, pp. 1141–1145.

[153] 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.

[154] A. Paszke, S. Gross, F. Massa, *et al.*, "PyTorch: An imperative style, high-performance deep learning library", *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[155] D. Kingma and J. Ba, "Adam: A method for stochastic optimization", *ArXiv e-prints*, 2017. arXiv: arXiv:1412.6980.

[156] T. Lee, R. Kashyap, and C. Chu, "Building skeleton models via 3-D medial surface axis thinning algorithms", *CVGIP: Graphical Models and Image Processing*, vol. 56, no. 6, pp. 462–478, 1994.

[157] C. Fiorio and J. Gustedt, "Two linear time union-find strategies for image processing", *Theoretical Computer Science*, vol. 154, no. 2, pp. 165–181, 1996.

[158] W. Kuo, M. de Bruijne, J. Petersen, K. Nasserinejad, H. Ozturk, Y. Chen, A. Perez-Rovira, and H. Tiddens, "Diagnosis of bronchiectasis and airway wall thickening in children with cystic fibrosis: Objective airway-artery quantification", *European Radiology*, vol. 27, no. 11, pp. 4680–4689, 2017.

[159] W. Kuo, A. Perez-Rovira, H. Tiddens, M. de Bruijne, and N. C. C. study group, "Airway tapering: An objective image biomarker for bronchiectasis", *European Radiology*, vol. 30, no. 5, pp. 2703–2711, 2020.

[160] H. Tiddens, S. Donaldson, M. Rosenfeld, and P. Pare, "Cystic fibrosis lung disease starts in the small airways: Can we treat it more effectively?", *Pediatric Pulmonology*, vol. 45, no. 2, pp. 107–117, 2010. DOI: https://doi.org/10.1002/ppul.21154.

[161] G. Bortsova, F. Dubost, L. Hogeweg, I. Katramados, and M. de Bruijne, "Semi-supervised medical image segmentation via learning consistency under transformations", in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, 2019, pp. 810–818.

[162] S. Chen, Z. S. Gamechi, F. Dubost, G. van Tulder, and M. de Bruijne, "An end-to-end approach to segmentation in medical images with cnn and posterior-crf", *Medical Image Analysis*, p. 102 311, 2021.