Engineering Deep Learning Systems for Robust and Accurate Focal Pathology Segmentation and Detection

Brennan Nichyporuk

Department of Electrical and Computer Engineering

McGill University, Montreal

August, 2021

A thesis submitted to McGill University in partial fulfillment of the requirements of the degree of

Master of Science

©Brennan Nichyporuk, 2021

Abstract

This thesis presents a deep learning framework, along with key insights and design decisions, for the problem of Multiple Sclerosis (MS) lesion segmentation and detection. Our approach, inspired by nnU-Net ('No-New-Net'), confirms that a baseline UNet, properly configured and optimized, can achieve state-of-the-art performance on medical image segmentation tasks. Our approach shows significant performance gains over previously published results on the same dataset. Next, we examine the segmentation-detection tradeoff that exists when segmenting MS lesions of different sizes. Specifically, in cases where there is a mix of small and large lesions, standard binary cross entropy loss will result in better segmentation of large lesions at the expense of missing small ones. We propose Lesion Size Reweighing (LSR) to reweigh the loss function such that good detection performance does not come at the cost of segmentation quality. Experiments show significant improvements in small lesion detection performance while maintaining segmentation accuracy. Finally, we examine the role of dataset bias in the context of aggregated datasets, proposing a generalized affine conditioning framework to learn and account for complex cohort biases across multi-source datasets.

Abrégé

Cette thèse présente un cadre d'apprentissage profond, ainsi que des perspectives clés et des décisions de conception, pour les problèmes de segmentation et de détection des lésions de sclérose en plaques (SEP). Notre approche, inspirée de nnU-Net («No-New-Net»), confirme qu'un UNet de base, correctement configuré et optimisé, peut atteindre des performances de pointe sur la tâche de segmentation d'images médicales. Notre méthode montre des gains de performance significatifs par rapport aux résultats précédemment publiés sur le même jeu de données. Ensuite, nous examinons le compromis segmentation-détection inhérent à la segmentation des lésions de SEP de tailles différentes. Plus précisément, dans les cas où il existe un mélange de petites et de grandes lésions, l'utilisation de la perte d'entropie croisée binaire standard se traduit par une meilleure segmentation des grandes lésions au détriment des petites lésions qui ne sont pas détectées. Nous proposons la repondération de la taille des lésions (LSR) pour pondérer la fonction de perte de telle sorte que les bonnes performances de détection ne se fassent pas aux dépens de la qualité de la segmentation. Les expériences montrent une amélioration significative de la performance de détection des petites lésions tout en maintenant la précision de la segmentation. Enfin, nous examinons le rôle du biais de données dans le contexte des jeux de données agrégés, en proposant un cadre de conditionnement affine généralisé pour apprendre et tenir compte des biais de cohorte complexes dans des jeux de données multi-sources.

Acknowledgements

First and foremost, I would like to thank my parents, Derrick and Phyllis, for their undying love and support over the years. I would like to thank my supervisor, Prof. Tal Arbel, for her support and guidance over the last few years. Her positive attitude made this journey just that much more rewarding. I would like to thank Dr. Douglas Arnold, Dr. Sridar Narayanan, and Zografos (Aki) Caramanos, for providing their clinical expertise. I would like to thank Dr. Louis Collins and Dr. Mahsa Dadar for handling data preprocessing, a critical part of making effective use of the data we had available. Finally, I would like to thank Justin Szeto and Jillian Cardinell for their assistance running numerous experiments and writing multiple papers, without which I would have not made it this far.

Thank you to NeuroRx Ltd. for providing several large labeled clinical trial datasets, without which the research discussed in the thesis would not be possible. Finally, I would like to thank the International Progressive MS Alliance, who provided the award (PA-1603-08175) that funded this research.

Contribution of Authors

Peer-Reviewed Conference Publications

- NICHYPORUK, B., CARDINELL, J., SZETO, J., MEHTA, R., TSAFTARIS, S., ARNOLD, D. L., AND ARBEL, T. Chort Bias Adaptation in Aggregated Datasets for Lesion Segmentation. In *Domain Adaptation and Representation Transfer, and Affordable Healthcare and AI for Resource Diverse Global Health*. Springer, 2021, pp. 101–111

Peer-Reviewed Conference Papers

- NICHYPORUK, B., SZETO, J., ARNOLD, D., AND ARBEL, T. Optimizing Operating Points for High Performance Lesion Detection and Segmentation Using Lesion Size Reweighting. In *Medical Imaging with Deep Learning* (2021). Eprint arXiv:2107.12978 (Short Paper)

Competition Papers

- NICHYPORUK, B., VASILEVSKI, K., HU, A., MYERS-COLET, C., CARDINELL, J., SZETO, J., FALET, J.-P., ZIMMERMANN, E., SCHROETER, J., ARNOLD, D. L., ET AL. Consensus learning with multi-rater labels for segmenting and detecting new lesions. *MSSEG-2 challenge proceedings: Multiple sclerosis new lesions segmentation challenge using a data management and processing infrastructure* (2021), 85. (Short Paper)

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Deep learning has become an important method across a number of fields, bringing about breakthroughs in processing text, speech, audio, images, and video [56]. In the context of computer vision, deep learning gained widespread attention with the publication of AlexNet [53] in 2012, with deep learning methods becoming state-of-the-art across a number of computer vision tasks shortly thereafter [104]. Comparatively speaking, the adoption of deep learning in the medical imaging field has been relatively slow, in part due to the increased computation required to process large 3D medical data. However, with the publication of a landmark biomedical image segmentation approach, the UNet in 2015, the field saw explosive growth, with the UNet outperforming competing methods on a number of public challenges [1, 65]. However, while UNet-based methods have performed well based on overlap metrics biased towards larger structures and pathologies such as DICE, some neurological diseases such as Multiple Sclerosis (MS), typically entail multiple lesions that span a wide range of sizes, from as little as three voxels to thousands of voxels. In such contexts, initial results appeared to indicate that the UNet performed poorly or otherwise under-performed other approaches [42, 93]. As a result, numerous publications have proposed new mechanisms or enhancements to improve or otherwise build upon the UNet architecture.

More recent work, nnU-Net ('No New-Net'), has shown that a UNet, without major modifications, can achieve State Of the Art (SOTA) performance on not just larger focal pathologies, but small ones too, provided that the model is well-tuned, and trained using a well engineered pipeline [38]. Although these findings appear to contradict a significant portion of the published literature, they are persuasive as they were established on 49 segmentation tasks across 19 *public* medical imaging datasets with test sets that are not accessible to participants. Overall, nnU-Net sets a new SOTA on 29 of the 49 tasks, while otherwise achieving results that are on par with, or near SOTA on the remaining tasks. More recently, this same approach was entered into two different medical image segmentation challenges, achieving 1st place out of 78 teams in the 2020 BRATS brain tumor segmentation challenge [40], and 2nd place out of 98 teams in the 2020 COVID19 lung lesion segmentation challenge (ranked 1st place among the 97 teams that did not have access to additional COVID19 lung lesion data) [79].

Overall, the No New-Net publication claims that *in practice* the large performance differences observed between methods typically have more to do with differences between experimental pipelines (i.e. implementation, hyperparameter optimization, pipeline configuration, etc.), rather than differences between the model architecture itself. This was also pointed out by Litjens et al. who made the case that "the exact architecture is not the most important determinant in getting a good solution" [60]. However, while the No New-Net publication does claim that many previously published architecture modifications are likely not superior to baseline, they do not make this claim in general, pointing out on the CREMI dataset [26] that: "while nnU-Net's performance is highly competitive (rank 6/39), manual adaptation of the loss function as well as electron microscopy-specific preprocessing may be necessary to surpass state-of-the-art performance" [38]. This indicates that while a well engineered pipeline is a necessary condition to obtain SOTA performance, it is not a sufficient one.

While the No New-Net publication is an important contribution to the literature, demonstrating the need to properly engineer a robust pipeline and to establish a strong

baseline *before* attempting to build upon it, it is focused on obtaining SOTA performance in the context of challenge datasets. However, in practice, achieving SOTA performance is not the only consideration. Uncertainty estimates, for instance, are particularly important in medical imaging contexts, and only require that dropout layers be added to the model [71]. Another emerging area of interest is generalization across datasets in the context of medical imaging segmentation problems. Medical images contain many sources of variability including differences in scanners, imaging acquisition parameters, and resolutions, with significant research being done in this area [8, 48, 100]. Variability in labeling, uncommon in the context of natural images, is a significant source of variability in the medical image segmentation context. Medical experts (i.e. raters) commonly disagree with respect to where the boundary of a lesion actually is [31], leading to high inter-rater variability. Recent work has examined modeling labeling style differences between individual raters in the context of a single dataset [15, 45, 89, 103]. However, as all of this work is done within the context of a single dataset, it does not investigate how labeling style differences affect generalization across datasets. Furthermore, labeling style differences between datasets prevents combining datasets, which is a significant disadvantage given the small size of most medical datasets.

In this thesis, we explore the process of engineering a deep learning systems, and making several distinct contributions along the way. First, we discuss the process of engineering a robust deep learning baseline for focal pathology segmentation and detection, which with the success of methods like No New-Net, we would argue is currently the most significant challenge the research community currently faces. We cover several important considerations with respect to metrics, pointing out easily made errors with respect to how metrics are calculated, monitored, and interpreted. Furthermore, we cover techniques to improve computational efficiency, which with common deep learning models taking days or even weeks to train, are an important consideration. We cover the process of establishing a robust baseline model on which to build on, examining common optimization difficulties, the process of tuning and configuring each baseline to the task

at hand, and several other important considerations that must be made when comparing methods. Second, through close metric analysis, we show that a significant gap exists between the optimal operating points (i.e. thresholds) for segmentation vs. detection, and present a method to close this gap such that optimal segmentation and detection performance can be obtained simultaneously. Third, we present the first approach that, by modeling rater biases across datasets, enables a single model to be trained on an aggregated dataset, containing multiple different cohort biases, without a significant drop in performance.

1.1 Multiple Sclerosis

Multiple Sclerosis (MS) is a neurological disease in which the insulating cover neurons in the brain and spinal cord become damaged over time. According to the Multiple Sclerosis International Federation, an estimated 2.3 Million people are afflicted with the disease, with Canada having the highest rates of MS in the world, with a prevalence of 291 per 100,000 people [12]. MS results in a wide range of signs and symptoms, with progressively worse sensory, motor, and cognitive deficits developing over time. There is no cure for the disease, with life expectancy 5 to 10 years lower than the general population, and with those toward the end of life requiring significant supportive care to carry out daily activities [18].

In Multiple Sclerosis, a common hallmark of the disease is the appearance of focal lesions in the brain and spinal cord as a result of the breakdown in the insulating cover of neurons [18]. Through the use of brain Magnetic Resonance Imaging (MRI), clinicians can monitor the development of focal lesions to monitor and stage the disease, or to determine the efficacy of treatment [69]. Labelling focal lesions remains a challenge, requiring trained experts to delineate lesions across 3D medical volumes. This process is subject to significant inter-rater and intra-rater variability, in which different raters will annotate the same sample differently, as will the same rater across different reads of the same sam-

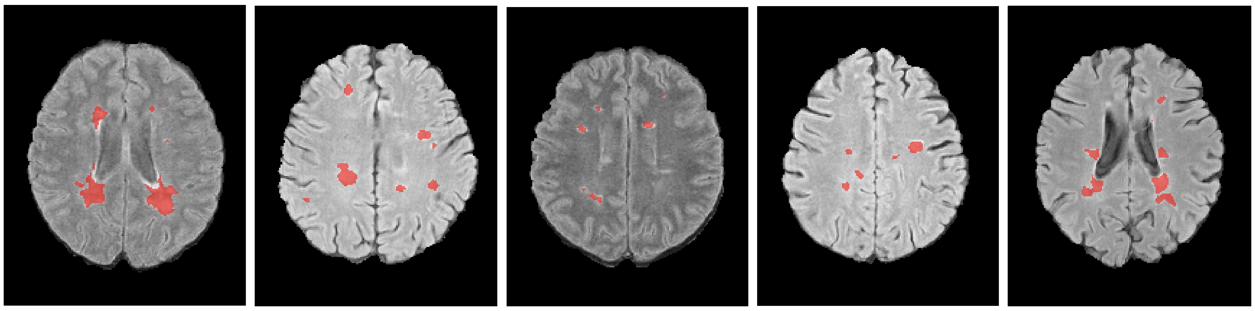


Figure 1.1: A cross-sectional slice from five random data samples. Example shows the degree of variability in Multiple Sclerosis imaging data, both in terms of lesion size (from very small to very large), and in terms of imaging differences across samples. All samples are of the fluid-attenuated inversion recovery (FLAIR) sequence type.

ple [27, 86]. The lesions themselves also present a great deal of variability, with lesions presenting in all sorts of shapes and sizes, and can range anywhere from three voxels to thousands of voxels in size. In Figure 1.1, we provide an example of the extent of variability that can be expected across different data samples. We discuss automated methods for segmentation in the context of multiple sclerosis in more detail in Section 2.3.

1.2 Contributions of Thesis

In this thesis, we will be examining the process of engineering deep learning systems in the context of focal pathology segmentation, presenting unique solutions to some of the problems we encountered along the way.

1. **Engineering Deep Learning Systems**: We discuss the importance of a well-engineered pipeline, backing this up with several sets of experiments to show the need for careful consideration of several inter-related factors, particularly with respect to computing and interpreting metrics, and computational efficiency. Next, we establish a robust model for segmentation and detection of focal pathologies, demonstrating the need to properly configure and tune models to the problem at hand.

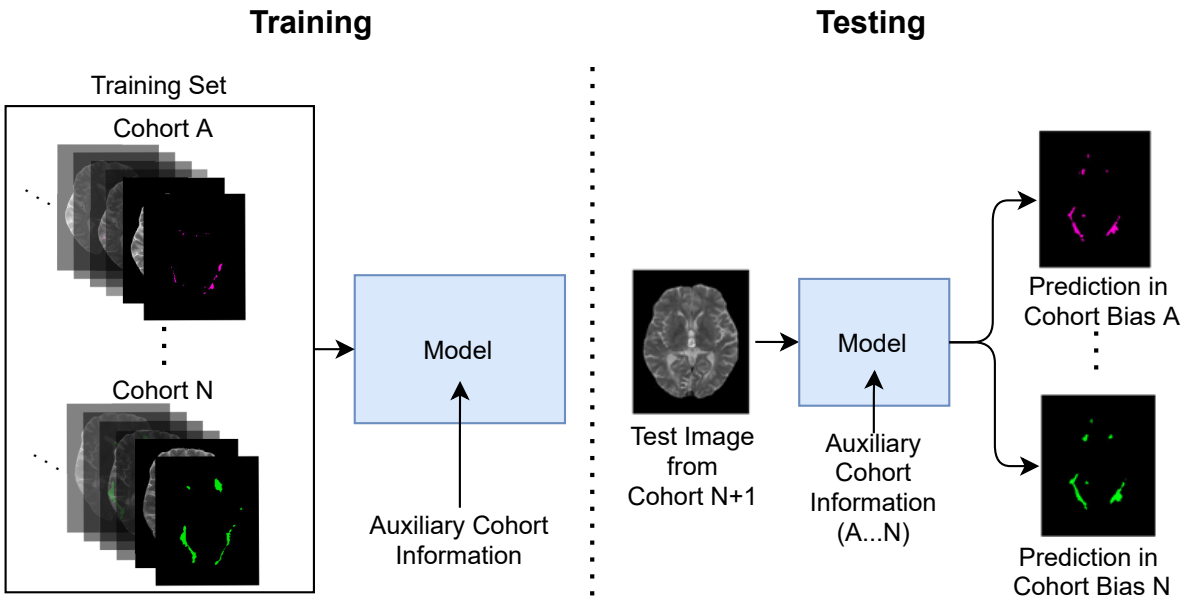


Figure 1.2: System overview showing training on the left and testing on the right. The left shows how we train with multiple cohorts and use auxiliary cohort information to learn the associated bias. On the right is how we use cohort information during testing to generate multiple labels for an image in a desired style.

2. **Lesion Size Reweighting**: Multiple Sclerosis lesions can span a range of sizes, from as little as 3 voxels, to well over 1000 voxels (see Figure 1.1). Given the wide range of lesion sizes, segmentation models trained with standard loss functions typically better segment large lesions at the expense of missing small ones. We present a novel loss reweighing strategy that substantially improves small lesion detection performance while maintaining segmentation performance.

3. **Cohort Bias Adaptation in Aggregated Datasets**: Medical datasets are typically small, which can be an obstacle to training a robust model that can generalize well. Combining datasets isn't straightforward given the high degree of intra-rater and inter-rater variability, along with differences in scanner, site, and acquisition parameters. We present a novel mechanism to model biases between patient cohorts as manifested in the labeling. We demonstrate how this mechanism can be used

to aggregate datasets with different labeling distributions. An overview of the approach can be found in Figure 1.2.

1.3 Outline of Thesis

This thesis discusses the process of engineering deep learning systems for focal pathology segmentation and detection.

Chapter 2 introduces relevant literature. We first discuss image segmentation in the context of natural images, discussing both machine learning and deep learning, highlighting important differences between the two approaches. Next, we cover image segmentation in the context of medical imaging in more detail, discussing recent developments which establish that a well-engineered U-Net (2015) remains state-of-the-art in 2021. We review work related to aggregating datasets, particularly with respect to the unique dataset-specific biases that must be taken into account.

Chapter 3 presents background on deep learning, the process of training deep learning models, and an in-depth look at the metrics used in the context of focal pathology segmentation.

Chapter 4 describes the engineering process involved in building deep learning systems, demonstrating the importance of several key implementation decisions, and the implications of these decisions on the model development and analysis process. Next, we examine several important configuration and hyperparameter decisions, demonstrating the importance of domain knowledge, in both deep learning and the problem under consideration, to effectively configure and tune deep learning models for the task of focal pathology segmentation.

Chapter 5 presents Lesion Size Reweighting (LSR) [75], an approach that reweights each lesion as a function of the number of voxels that it contains. LSR addresses a problem inherent with voxel-wise loss functions, which while effective at training models to produce accurate segmentations as measured by voxel-wise metrics such as DICE, suffer

from an inherent bias towards large lesions that contain more voxels at the expense of missing small ones. LSR closes the segmentation/detection performance gap, showing that with the right lesion reweighing strategy, high overall simultaneous detection and segmentation accuracies are achievable.

Chapter 6 proposes Source-Conditioned Instance Normalization (SCIN) [74], an approach that models source-specific (or cohort-specific) biases in aggregated datasets. By doing so, SCIN makes it possible to train a high performance model on an aggregate dataset, avoiding the performance penalty observed when naively pooling datasets with different biases together.

Chapter 7 concludes the thesis, summarizing the key contributions detailed in Chapter 4 (Engineering Deep Learning Systems), Chapter 5 (Lesion Size Reweighting), and Chapter 6 (Cohort Bias Adaptation in Aggregated Datasets).

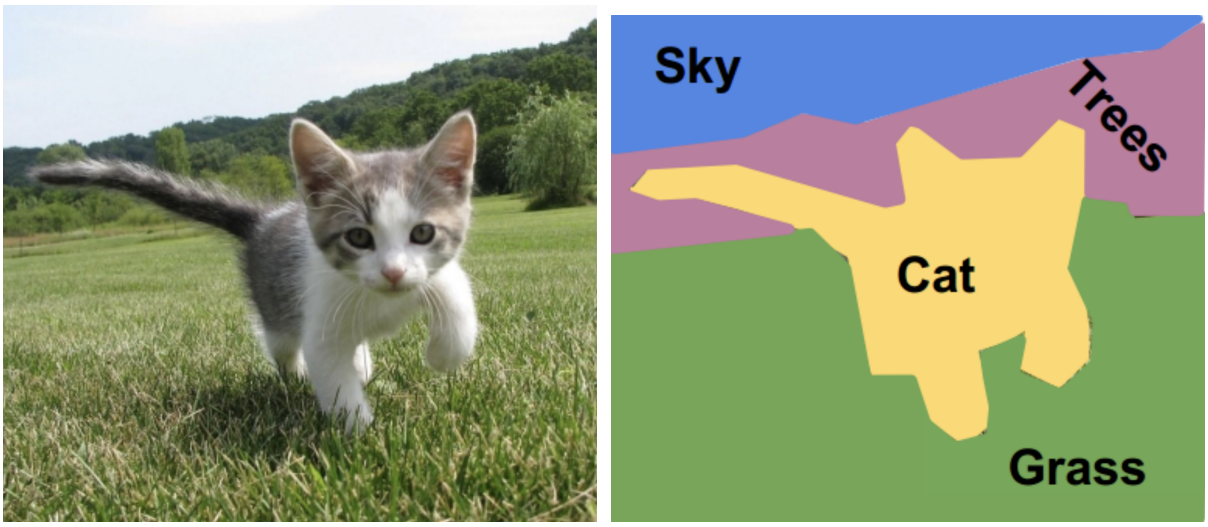
Chapter 2

Related Work

This chapter reviews pertinent literature related to image segmentation in the medical imaging context. First, we review previous work on image segmentation in the context of natural images. Second, we review prior work in the medical image segmentation field, pointing out recent developments that highlight the need for an increased focus on reproducibility. Third, we take a closer look at previous work related to medical image segmentation in the context of multiple sclerosis. Fourth, we review prior literature on learning with aggregated medical imaging datasets.

2.1 Image Segmentation

Image segmentation is the task of labeling every pixel of an image with a specific class, partitioning the image into specific segments. An example of image segmentation in the context of natural images can be found in Figure 2.1, where an image of a cat against an outdoor background is segmented into four classes: Cat, Grass, Sky, and Trees. Classification methods originally developed for unstructured data, such as Support Vector Machines [107], or Random Forests [83], when applied to the raw pixel intensities in an image, make the implicit assumption that each pixel is completely independent. However, this assumption clearly doesn't hold in the context of imaging data, as the intensities



(a) Cat **(b) Segmentation: Grass, Cat, Trees, Sky**

Figure 2.1: An image segmentation example. Images courtesy of [58]

of nearby pixels in images are generally correlated. For example, in Figure 2.1a, we see that small neighborhoods of pixels belonging to the 'Grass' class end up forming a common pattern. To take advantage of correlations between neighboring pixel intensities, we can utilize feature extraction techniques that implicitly model spatial relationships, such as the convolution operation. Features extracted with such techniques are a function of the pixel intensity values of more than a single pixel, allowing spatial relationships between pixels to be modelled. For example, the Sobel Operator [47], a type of convolution, emphasizes edges in imaging data.

One limitation of pixel-wise classifiers is that while spatial relationships between pixels were modeled implicitly through feature extraction techniques (e.g. sobel operator, textures, etc.), pixel-wise classifiers do not explicitly model spatial relationships between neighboring *predictions*. In other words, pixel-wise classifiers do not explicitly take into account the consistency of a particular pixel's *class prediction* with the *class prediction* of neighboring pixels. To address these limitations, Markov Random Fields (MRFs) [59] and Conditional Random Fields (CRFs) [35] were introduced, allowing spatial relationships between neighboring labels to be considered.

Although MRFs and CRFs improved upon pixel-wise classifiers that did not *explicitly* consider spatial relationships, it is important to note that the benefit produced by these methods is directly related to the degree to which the extracted features are able to model spatial relationships. If extracted features incorporate sufficient context, explicitly modeling spatial relationships with MRFs and CRFs is not necessary. Indeed, recently developed methods that make use of *deep learning* [78] are able to learn features that incorporate much more context that hand-crafted methods, alleviating the need to explicitly model spatial relationships between pixels. We discuss deep learning applied to the task of image segmentation in more detail in the following section.

2.1.1 Deep Learning for Image Segmentation

Although feature extraction techniques, like the Sobel Operator [47], allow the modeling of spatial relationships between image pixels, they are hand-crafted, requiring manual design and thus limiting the complexity of any individual feature extractor to that which can be designed by a human expert. Much more recent work has focused on learning features via *deep learning*, which allow arbitrarily complex features to be learned through backpropagation [56].

Deep learning based techniques form the basis of most state-of-the-art techniques in use today. Deep learning models have been more successful than traditional methods because they can learn complex features (also known as *representations*) optimized for the task at hand rather than using much less complex hand-crafted features that are created by domain experts. Deep learning models learn these features through the use of multi-layer models, that sequentially build up representations of data at multiple levels of abstraction [56]. In the context of natural images, common pixel-wise segmentation models include FCN [61], an type of *encoder-only* architecture that combines predictions from multiple scales. Skip connections of this kind resemble deep supervision [106], but also help provide more fine-grained predictions. This technique was improved upon by SegNet [6], a type of *encoder-decoder* architecture, which uses a fully convolutional de-

coder in combination with Max Unpooling to make use of fine-grained location, but not content, information from the encoder. This was further improved upon by methods such as FC-DenseNet [41], a type of *encoder-decoder* architecture, which uses a fully convolutional decoder in combination with feature maps from the encoder to make use of fine-grained location and content information.

2.2 Image Segmentation in Medical Imaging

Medical image segmentation is the process of subdividing medical images into two or more salient regions. Some examples include identifying specific brain structures such as the hippocampus, corpus callosum, or thalamus. Or identifying pathologies such as brain lesions. Automatic segmentation is useful for diagnosis, treatment, surgery planning, and clinical trials. Various imaging modalities, including Ultrasound, PET, X-Ray, CT, and MRI can be used to understand the internal anatomy of the patient. Examples of a few of these modalities can be found in Figure 2.2.

Medical image segmentation poses a set of unique challenges that differ from natural image segmentation. One of the most overt differences is the 3D nature of many types of medical imaging, which can require significantly more computational resources to process compared to natural images. Other important differences include the varied nature of many types of pathologies, which can range from just a few voxels, to thousands of voxels in size. Depending on the modality (X-Ray, MRI, etc.), there is also considerable uncertainty with respect to where the boundary of a pathology actually is. In practice, this results in rates of *intra*-rater and *inter*-rater reliability that are much higher then can generally be expected when labeling natural images. Furthermore rating protocols designed to increase *inter*-rater reliability *within* a dataset, are not uniform across the field, and can thus introduce systematic *biases* across datasets. In Section 2.4, we discuss common forms of variance and bias in medical datasets in more detail.

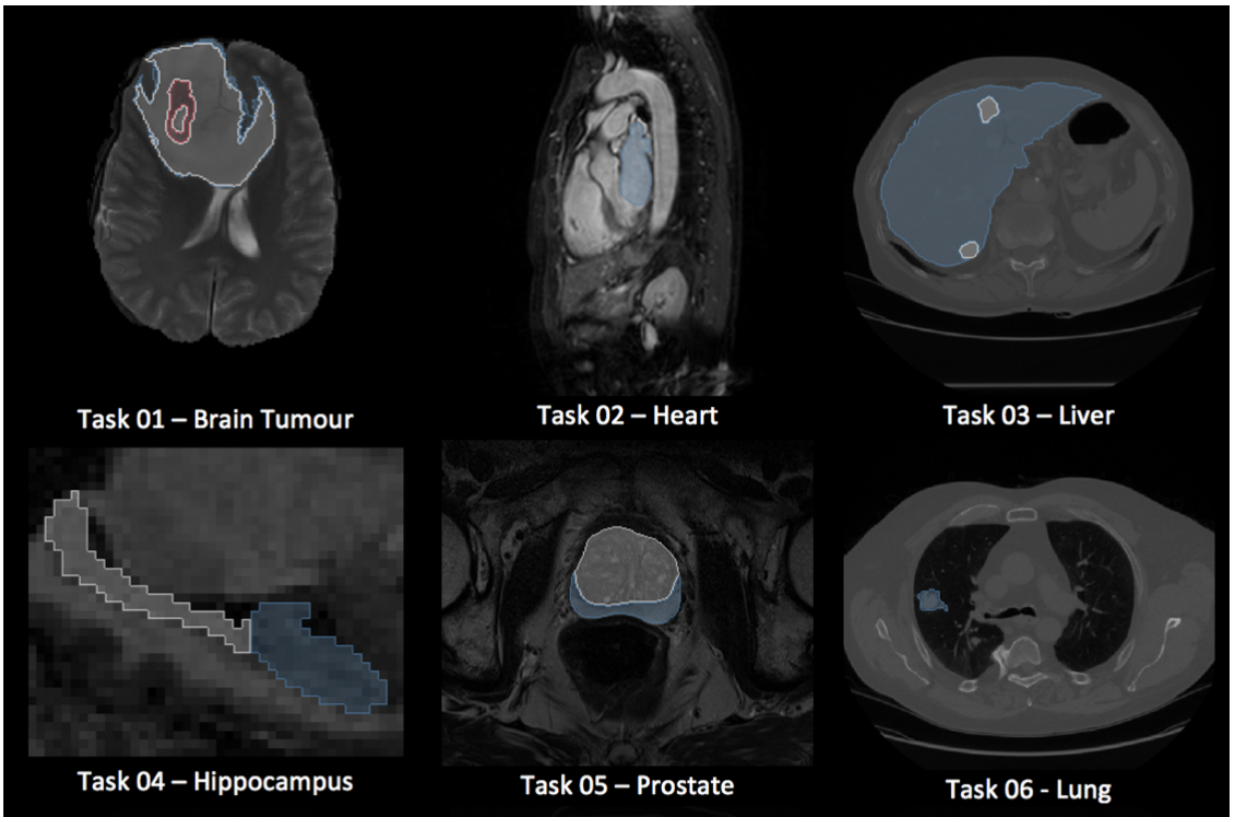


Figure 2.2: Examples of different imaging modalities of several different parts of anatomy. Task 01: Brain Tumour MRI, Task 02: Heart MRI, Task 03: Liver CT, Task 04: Hippocampus MRI, Task 05: Prostate MRI, Task 06: Lung CT. Images courtesy of: [90]

The medical image segmentation field has evolved substantially over time. Early approaches typically used simple thresholding techniques [57], although these methods were limited as they did not make use of contextual information. More recent work includes atlas based methods, that make use of prior information regarding the location of key brain structures to produce an approximate segmentation [66, 109]. However, atlas-based methods are only applicable to the segmentation of well-defined structures (hippocampus, thalamus, cortex, etc.) in a healthy brain, rather than focal pathologies which can potentially appear anywhere. Methods that utilize machine learning classifiers (e.g. Logistic Regression [21], SVM [20], etc.) and hand-crafted features (e.g. SPIN [43], SIFT [62], etc.) are another common approach, and have been applied to both healthy structure segmentation and pathology segmentation [30,93]. However, standard machine learning classifiers do not explicitly model spatial relationships (e.g. consistency) between

class predictions. Probabilistic graphical models, such as Markov Random Fields (MRFs) or Conditional Random Fields (CRFs), improved on traditional machine learning classifiers by explicitly modeling spatial relationships between voxels [35, 59]. Hierarchical CRFs, allowed the modeling of even higher-level relationships [55]. However, machine learning and probabilistic graphical models, like in the case of natural images, are limited by the quality of the features extracted, with performance highly dependent on the feature extraction process [23]. Like natural images, deep learning has allowed models to learn arbitrarily complex high-level features, but adoption has lagged behind that of natural images due to a number of practical considerations, the most important of which we discuss in Section 2.2.1.

2.2.1 Deep Learning for Medical Image Segmentation

Deep learning has also revolutionized the Medical Image Segmentation Field, addressing the same key issue that also existed in natural image segmentation, namely the limited representational capacity of handcrafted features. However, in the context of medical images, progress has been hindered by a number of practical considerations. Medical images are typically much larger (e.g. 256x256x256) compared to natural images, which even at high resolution (e.g. 1200x1200) take up much less memory. As a result, methods and training procedures designed for natural images do not typically translate well to the medical imaging field. Indeed, given GPU memory constraints, batch sizes are typically limited to just one or two samples. Smaller batch sizes entail significantly more gradient noise, and don't permit the use of methods that typically speed up the optimization process, such as batch normalization, making models much more difficult to optimize. To make matters worse, wall-clock training times of **several days** on a single GPU are common, making the number of models a researcher can train in a given time period extremely limited. Indeed, given the computational resources required to train a model, researchers have *no choice* but to tune hyperparameters *by hand*, as hyperparameter optimization algorithms (e.g. AutoML) are too costly, with a single hyperparameter opti-

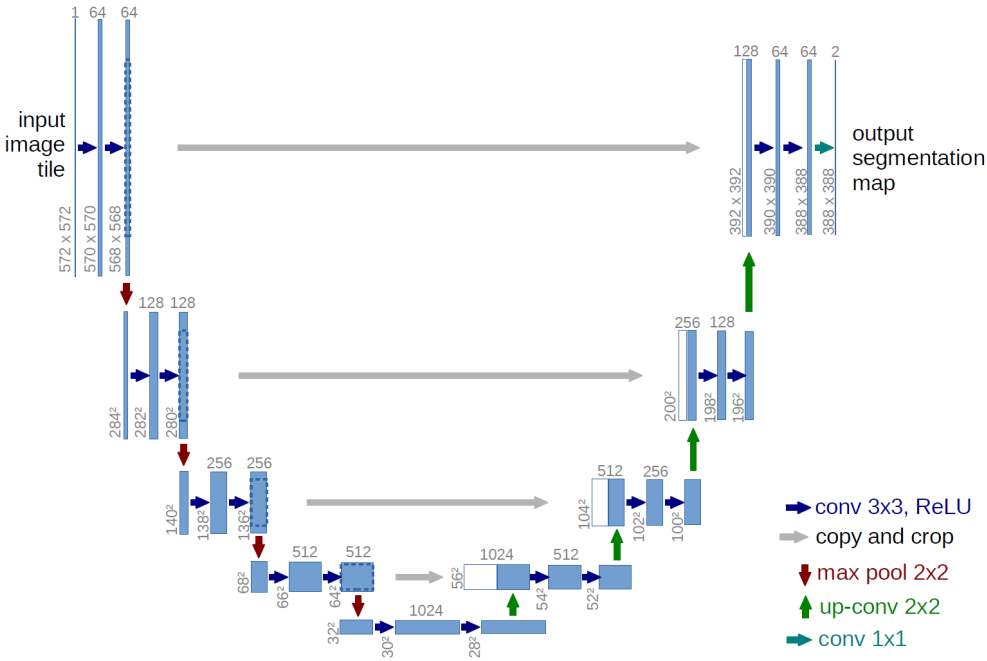


Figure 2.3: The UNet architecture originally proposed by Ronneberger et al. Each level of the contracting path (left), consists of two 3x3 Convolution-ReLU blocks and, with the exception of the last level, a max pooling operation to downsample the multi-channel feature maps. Each level of the expanding path (right), consists of an 'up-conv' (upsampling+convolution) which halves the number of feature maps, a concatenation with the corresponding cropped feature maps from the contracting path, and two 3x3 Convolution-ReLU blocks. The output classifier consists of a single 1x1 Convolution. Figure courtesy of [78].

mization process with 50 experimental runs potentially taking up to a year to complete on a single GPU.

The development of better optimization algorithms, like Adam [51], or SGD with increased Momentum (e.g. 0.99) [61], and methods such as instance normalization [96], have made training high performance models with small batch sizes much more tenable. Indeed, one such method, the UNet [78], was a key milestone for deep learning based medical image segmentation, with the approach taking first place in the 2015 ISBI cell tracking challenge [65] by a large margin. The UNet architecture consists of a contracting path that builds up representations at coarser and coarser scales, and a expanding path that integrates higher-level representations with finer grained representations via skip connections from the expanding path. The skip connections also play a role in gradient flow, improving the rate at which the network converges. The UNet architecture can be

found in Figure 2.3. Other important deep learning architectures proposed for medical image segmentation include DeepMedic for Brain Tumor and Stroke Lesion Segmentation [46] and VNet for prostate segmentation [70].

Reproducibility and the Current State of the Art

Despite the strides made towards training deep learning models in the context of medical image segmentation, deep learning models remain difficult to optimize due to the sheer number of codependent design choices involved. This is further complicated by the amount of computational resources required to train each model. Indeed, with models taking days, and even weeks to train, automated hyperparameter optimization methods, such as random search [7], become intractable, with a 50 experiment hyperparameter optimization process taking up to a year to complete on a single GPU. Within these constraints, most researchers must tune hyperparameters 'by hand' (see Figure 2.4), a process that is highly dependent on the skill and experience of the researcher, inevitably leading to suboptimal models and pipelines [38].

Since its publication in 2015, the UNet architecture has been the default baseline for medical image segmentation tasks. Indeed, the UNet publication has garnered over $29,000$ citations as of mid-2021, with many publications proposing new extensions to further improve the architecture. However, despite the sheer number of publications that propose improvements to the UNet architecture, the top performing method across a

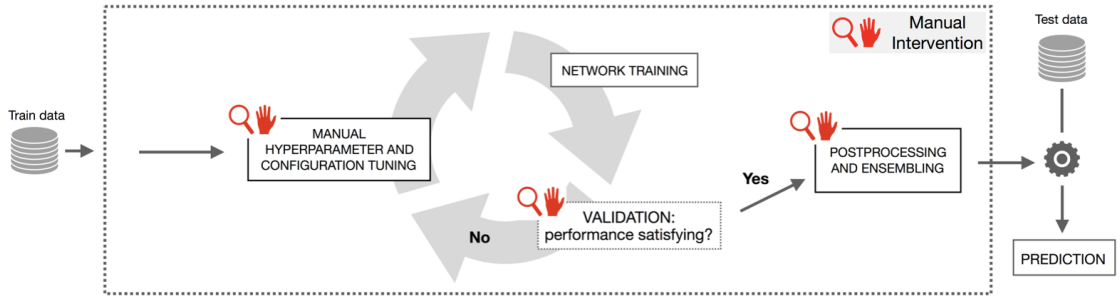


Figure 2.4: Current practice for developing biomedical image segmentation models is iterative. Model configuration is manually designed, and hyperparameters are manually tuned. Figure courtesy of [38].

wide variety of public medical imaging segmentation challenges is nn-UNet (No-New Net) [38]. As the name suggest, nn-UNet is not a new deep learning architecture or method, instead the paper advocates that a *properly implemented*, *well-trained* UNet can achieve competitive state-of-the-art performance without major modifications. Although contrary to much of the literature, the findings of nn-UNet are persuasive as they are demonstrated in the context of public challenges with test set labels that are not accessible to participants. nn-UNet has further proven itself as the current state-of-the-art in a number of recent challenges, including a first place finish in BRATS 2020 [40], and a second place finish in the COVID19 lung lesion segmentation challenge (ranked first among methods that did not use additional data) [79].

It should be noted that the findings of nn-UNet, although suggestive, do not prove that all purposed extensions to the UNet architecture are of no benefit. Rather, the findings demonstrate that basic, often overlooked design, methodological, and optimization choices have the *most significant* impact on model performance.

While it is impossible to conclusively demonstrate why the vast majority of published modifications to the UNet architecture have not seen widespread adoption among the community, the shear difficulty involved in properly configuring deep learning systems (including competitive baselines), has been suggested as a key factor [39]. Numerous expert decisions are involved in configuring a deep learning system for medical image segmentation, with factors such as the optimal learning rate schedule, loss function, normalization strategy, and data augmentation strategy varying widely between datasets. Given the amount of time required to train a deep learning system for medical image segmentation, extensive expert knowledge and experience is required to efficiently configure and optimize the overall system, necessitating insight into the relationship between not only each sub-component (e.g. loss function, learning rate schedule, etc.), but the properties of the dataset itself (e.g. distribution of voxel spacing, class ratio, imaging characteristics, etc.). Indeed, given that a typical experiment can take days to run, the the number of experiments that can be run in a given period of time is severely limited, with automated

hyperparameter optimization methods, tuning just a small subset of the available configuration and hyperparameter choices, taking up to a year to complete given the large number of experimental runs required [7].

Another possible reason why many published methods have not seen widespread adoption are implementation errors in the model or pipeline itself. Although there exists little research on the prevalence of implementation errors that materially impact research findings in the context of deep learning for medical image segmentation, recent work by Narang et al. [73] provides some insight into the reproducibility of modifications to the transformer architecture [101] (a model designed for natural language processing). In this publication, the authors implement and evaluate twelve published transformer architecture variants, finding that the majority produced no performance improvements at all, a discovery that directly contradicted the results reported in the original publications. Additionally, they argue that hyperparameter optimization, while a contributing factor, was not the only aspect driving the failure to reproduce the studied methods. Indeed, an extensive hyperparameter optimization process of one of the published methods (Universal Transformers, ICLR 2019 [22]) revealed that the method under-performed the baseline transformer architecture no matter which hyperparameters were selected. This despite the authors using the same dataset and validating the (re)implementation of each method with the original authors.

Although these findings may seem shocking at first glance, there has been growing awareness of a significant replication crisis across multiple scientific disciplines. For example, a 2015 meta-analysis of five independent replication projects (each replicating multiple publications) by five independent groups of researchers placed the replication rate at 22% to 49% [28]. One landmark study, "Evaluating the replicability of social science experiments in *Nature* and *Science* between 2010 and 2015", involving 21 publications, with research plans approved by the original authors and with samples sizes on average five times larger than the original studies, found that only 62% of publications could

be reproduced, with effect sizes that were approximately half of the effect size originally reported.

2.3 Image Segmentation in Multiple Sclerosis

Early work applying deep learning to Multiple Sclerosis includes a 3D patch-based approach entered into the 2015 Longitudinal Multiple Sclerosis Lesion Segmentation Challenge [13, 97], achieving results comparable to inter-rater variability. Later work used a modified UNet to segment T2 lesions, however this method under-performed approaches that did not utilize deep learning, particularly with respect to small lesions [42, 93]. Later work cascaded two 3D patched-based approaches, where the second network is intended to reduce the false positive rate of the first network, but the false positive rate of this approach remains higher than traditional approaches. While early deep learning methods appeared to struggle with small lesions, more recent work on a large private dataset showed that a UNet can indeed achieve respectable performance, even on small lesions [29]. Indeed, recent work by nnU-Net (No New-Net), demonstrated this in the context of the *public* 2008 Multiple Sclerosis Segmentation Challenge Dataset [86], showing that a baseline UNet architecture, properly trained and optimized, can outperform traditional approaches [39].

To provide some background regarding the performance range of various deep learning techniques, particularly in the context of the multiple sclerosis dataset used in this thesis, we direct the readers attention to Table 2.1. The table lists a number of different deep learning methods. The 'style' of the CNN that is used is either 'patch-wise', defined as a model that is trained and tested on patches of data, or 'semantic-wise' defined as a model trained on a full volume (or slice) of data. The 'Dim', or dimensionality of the method refers to whether the method takes as input a 3D volume, a 2D slice, or a 2.5D slab (a stack of 3 or more slices). Methods are stratified by 'database', with results reported on ISBI 2015 [13], MICCAI 2008 [86], MICCAI 2016 [17], or a 'Proprietary' dataset (refer to

Methods	Database	Dim	CNN Style	DSC	PPV
Roy et al. (2018) [80]	ISBI 2015	3D	Semantic-wise	0.524	0.866
Birenbaum and Greenspan (2016) [9]	ISBI 2015	3D	Patch-wise	0.627	0.789
Valverde et al. (2019) [99]	ISBI 2015	3D	Patch-wise	0.63	0.840
Aslani et al. (2019) [3]	ISBI 2015	2D	Semantic-wise	0.61	0.899
Aslani et al. (2018) [2]	ISBI 2015	2D	Semantic-wise	0.698	0.74
Zhang et al. (2019a) [112]	ISBI 2015	2.5D	Semantic-wise	0.693	0.908
Valverde et al. (2016) [98]	MICCAI2016	3D	Patch-wise	0.54	N/A
McKinley et al. (2016) [67]	MICCAI2016	3D	Semantic-wise	0.59	N/A
Kazancli et al. (2018) [50]	Proprietary	3D	Patch-wise	0.58	N/A
La Rosa et al. (2020) [54]	Proprietary	3D	Semantic-wise	0.60	0.64
Brosch et al. (2015) [11]	Proprietary	3D	Semantic-wise	0.355	0.414
Gabr et al. (2020) [29]	Proprietary	3D	Semantic-wise	0.82	N/A
Coronado et al. (2020) [19]	Proprietary	3D	Semantic-wise	0.77	N/A
Zhang et al. (2018) [111]	Proprietary	2D	Semantic-wise	0.672	0.724
Aslani et al. (2020) [4]	Proprietary	3D	Semantic-wise	0.50	0.519
Gessert et al. (2020a) [33]	Proprietary	4D	Semantic-wise	0.64	N/A
Gessert et al. (2020b) [34]	Proprietary	3D	Semantic-wise	0.656	N/A
Zhang et al. (2019b) [113]	Proprietary	2D	Semantic-wise	0.660	N/A

Table 2.1: Proposed deep learning methods to segment T2 lesions in multiple sclerosis. Table adapted from [110].

the publication of each method for a description of the private dataset used). The metrics reported include 'DSC', which refers to voxel-level DICE performance, and 'PPV', which refers to voxel-level positive predictive value. Lesion-Level detection results, despite being an important metric for this problem, were not consistently reported across methods and therefore do not appear in this table. A description of both metrics, including the applicability of each metric to the problem at hand, can be found in Section 3.4.

2.4 Learning with Aggregated Medical Imaging Datasets

Given that modern image segmentation techniques benefit from large amounts of data, the relatively small size of medical imaging datasets can be an obstacle to training high performance focal pathology segmentation systems. One strategy to increase the size of the training set is to aggregate multiple medical imaging datasets, labeled with the same type of focal pathology, together. However, this may actually *decrease* overall performance due to cohort biases that affect the label distribution of each constituent dataset.

Current literature has focused on accounting for distributional differences between samples with known differences in image acquisition parameters or image sequences [8, 48, 100], or on accounting for distributional differences in the labeling of two or more raters of the *same* sample [36, 44, 45, 52, 108]. However, there has been little research that accounts for *unknown* cohort biases that exist between *different* datasets. In Chapter 6, we discuss this problem in more detail, and propose a novel solution.

2.5 Summary

In this chapter, we reviewed relevant literature for the task of image segmentation, focusing on recent developments in deep learning applied to medical image segmentation in particular. We began by discussing image segmentation in the context of natural images, we then discussed image segmentation in the context of medical imaging before narrowing our focus to image segmentation in the context of Multiple Sclerosis. We presented a brief overview of published techniques in all three cases, focusing on medical image segmentation in general, and Multiple Sclerosis lesion segmentation in particular. Techniques discussed include traditional pixel-wise approaches based on hand-crafted features and pixel-wise machine learning classifiers (e.g. SIFT + Support Vector Machines), methods that explicitly model spatial relationships (e.g. MRFs, CRFs), and more recent state-of-the-art techniques that make use of deep learning (e.g. UNet). Beyond the method's themselves, we discussed recent work showing that a well-optimized baseline UNet remains SOTA on most tasks, highlighting the need for an increased focus on reproducibility. In the next chapter, we detail important background to set the foundation for the rest of the thesis.

Chapter 3

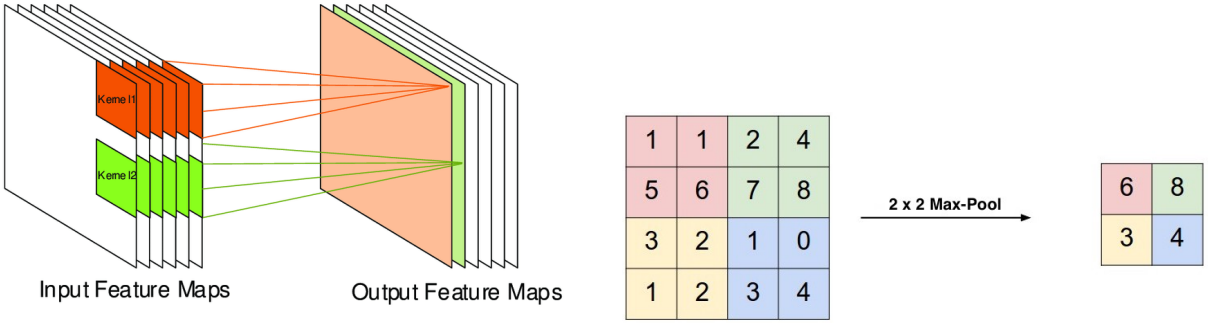
Background

This chapter reviews background relevant to the task of engineering deep learning systems for focal pathology segmentation and detection. We review key terms and concepts within the deep learning field and describe the metrics used to measure model performance. Overall, this chapter provides the fundamentals required to understand and evaluate the techniques introduced in the next three chapters.

3.1 Deep Learning

Deep learning is a subfield of machine learning that has revolutionized the fields of computer vision and natural language processing. Deep learning, in general, is the use of multi-layer computational models to buildup representations of data at multiple levels of abstraction [56, 84]. These models are trained through the backpropagation algorithm, which updates the parameters of each layer such that the output of the model more closely matches the target [81]. Deep learning can be differentiated from more traditional machine learning by recognizing that deep learning methods can learn representations ('features') directly from data, whereas traditional machine learning algorithms require representations ('features') to be crafted by domain experts. By learning features end-to-

end, deep learning algorithms are able to learn more complex and robust features than can be extracted by more rigid human-written algorithms.



(a) Multi-Channel Convolution **(b)** 2x2 Max-Pool

Figure 3.1: Visualization of two of the most common operators in Convolutional Neural Networks. Images courtesy of [102] and [49], respectively.

3.2 Convolutional Neural Networks

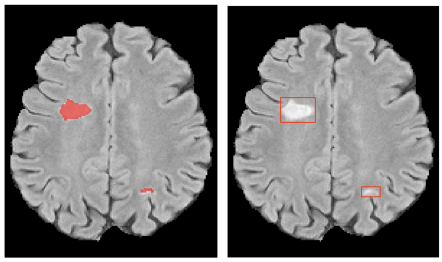
Convolutional Neural Networks (CCNs) are a form of deep learning the makes use of convolutional kernals. Convolutional kernals make the implicit assumption that pixels that are spatially near each other are related. This assumption allows convolutional kernals to act as feature detectors within a constrained receptive field (typically a 3x3 pixel/voxel space). Such approaches are very paramamter efficent as the convolution can be applied across the output of the *entire* previous layer (known as weight-sharing). The output of a convolutional layer is the dot product of the convolutional kernal with the input to that layer. As can be seen in figure 3.1a, spatial relationships are preserved in the output of a convolutional layer.

CNNs employ pooling-layers to reduce the spatial dimensions of the input and allow subsequent layers to work at a higher level of abstraction than a single-layer convolutional neural network. Figure 3.1b demonstrates the application of 2x2 (stride 2) max-pooling to a 4x4 input. We can see that the maximum activation within each non-overlapping window is taken as the output.

Between subsequent convolutional layers, a non-linear activation function is typically used. The use of a non-linear activation function allows CNNs (and ANNs more generally) to learn non-linear representations of the input. In general, CNNs use the ReLU activation function (or similar variants such as the LeakyReLU activation function [63]).

3.3 Training Neural Networks

Training neural networks consists of repeated application of the backpropagation algorithm [56,84] followed by a weight update step. At a very basic level, this consists of passing a batch of data through the network, computing some loss between the the neural network output and the target output, using the backpropagation algorithm to calculate gradients for all weights in the network with respect to this loss, and then applying a weight update step that is a function of these gradients. In practice, there are a number of different loss functions that can be used, but any such loss function must be *differentiable*. Choosing the most appropriate loss function is an important consideration when designing a deep learning system, as the network weights will be optimized to minimize this loss.



(a) Segmentation **(b)** Detection

Figure 3.2: Example illustrating the difference between segmentation (left) and detection (right).

The training process continues until the metric of interest, computed on the hold-out validation set, plautus. Although the loss on the validation set *usually* serves as an adequate measure of model performance, it is important to remember that the loss function is simply a differential *surrogate* for the metric that we are actually interested in (Detection F1, Segmentation DICE, etc.). In some cases, the metric of interest cannot be computed in real-time without parallelization, a programming paradigm that can complicate the pipeline development process.

Deep learning models are also susceptible to overfitting, a process where the model learns features from the training set that do not generalize to unseen data. Overfitting can be addressed by using regularization. The most common forms of regularization include L2 weight decay, data augmentation, and dropout [92].

3.4 Metrics

What constitutes the best model is typically based on some metric, in particular for segmentation tasks: F1 (DICE) score, or average precision (AP), are most often used. Furthermore, since models can take hundreds of GPU hours to train, it is important to make sure that the chosen metric improves over the course of training so that poor models can be terminated early, and to insure that any issues can be quickly identified.

Metrics give an indication of the performance of the model. For class imbalanced problems, TPR (True Positive Rate, or Recall) and PPV (Positive Predictive Value, or Precision) are more informative than metrics than make use of TN (True Negative) as this term can dominate the metric when the negative label is the majority class [82].

$$TPR = \frac{TP}{TP + FN} = 1 - FNR \tag{3.1}$$

$$PPV = \frac{TP}{TP + FP} = 1 - FDR \tag{3.2}$$

$$F1 = \frac{2 \cdot TP}{2 \cdot TP + FP + FN} = 2 \cdot \frac{TPR \cdot PPV}{TPR + PPV} \tag{3.3}$$

In the context focal pathologies, both segmentation and detection are important metrics to gauge model performance. Segmentation metrics, however are inherently biased towards lesions that contain more voxels. But given that the detection of lesions of all sizes is typically important (to diagnose disease, monitor treatment efficacy, etc.), seg-

mentation metrics alone are insufficient in most contexts. In practice, both metrics are considered.

3.4.1 Segmentation Metrics

Voxel-Level metrics provide an indication of the quality of the segmentation output produced by the network. They are a function of TP, FP, and FN at the *voxel-level*. Voxel-Level TP, FP, and FN are computed by binarizing the output of the model at a given threshold.

3.4.2 Detection Metrics

Lesion-Level metrics provide an indication of the lesion detection capability of the model. These metrics are a function of TP, FP, and FN at the *lesion-level*. Lesion-level TP, FP, and FN are calculated as follows: First the output of the model is binarized. Next, a connected component analysis (using an 18-connected component neighborhood as illustrated in Figure 3.3) is performed on both the binarized output and the 'ground truth' labeling, with each connected component representing a distinct lesion. A lesion in the binarized model output is considered a TP if it overlaps with at least three, or more than 50%, of the ground truth lesion voxels. Otherwise, it is a FP. Ground truth lesions with insufficient overlap are considered a FN.

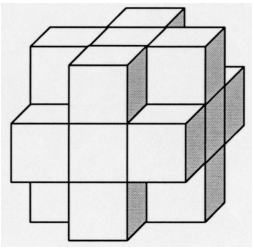


Figure 3.3: View of the 18-connected component neighborhood of a single voxel. Image courtesy of [95].

Metrics are stratified by size. With each TP, FP, or FN lesion classified as either 'Tiny' (1-2 voxels), 'Small' (3-10 voxels), 'Medium' (11-50 voxels), or 'Large' (51+ voxels). Metrics are computed over a range of thresholds to produce a precision-recall (PR) curve.

Ground Truth

It important to point out that perfect ground truth labels, in the case of medical imaging, do not exist [16, 64, 105]. This is a direct result of the considerable uncertainty involved in evaluating medical imaging data, from which it can be difficult to distinguish between various types of tissues or anatomical structures. Furthermore, given the relatively low resolution of medical images, multiple tissues may contribute to a single voxel, making it impossible to classify a single voxel into one class or the other. Indeed, there is considerable debate on what constitutes the boundary of a focal pathology in the first place. Practically, this means that the labels produced by any two experts will not necessarily be exactly the same [16, 64, 105]. In fact, even labels produced by the same expert are likely to be somewhat different across ratings [10, 32, 64].

Metric Conventions

Per-Scan metrics are calculated on a per-scan basis. In particular, TP, FP, and FN counts are tallied for each scan and then used to compute a scan-specific TPR, PPV, and F1-score which are then averaged across all scans. *Per-Scan* metrics can be problematic in cases where there exists scans without any lesions identified in ground truth (TPR, PPV, and F1 are undefined). In such cases, a arbitrary convention can be defined.

It should be noted that there is also ambiguity with respect to how per-scan PR curves should be merged. One approach, suggested in a popular machine learning library tutorial *scikit-learn*[*], while appropriate in the context suggested, computes the sample mean approximating $\overline{\text{TPR}} = E[\text{TPR}|\text{PPV}]$ without explicitly setting a fixed threshold. Since we must select a fixed threshold to produce a binary segmentation map at test time, the sample mean approximating $(\overline{\text{PPV}}, \overline{\text{TPR}}) = E[(\text{PPV}, \text{TPR})|\text{THRESHOLD}]$ is more correct, as this will produce a sample mean $(\overline{\text{PPV}}, \overline{\text{TPR}})$ that is representative of the expected value of these variables given a fixed threshold. The former implementation is overly opti-

[*] https://scikit-learn.org/stable/auto_examples/model_selection/plot_roc_crossval.html

mistic, computing an average TPR at a fixed PPV (rather than at a fixed THRESHOLD), averaging together TPR values obtained at potentially different thresholds. Small *implementation* details like this can have a significant impact on the results reported, making a fair comparison across methods difficult or impossible. Furthermore, detection metrics, can be even more arbitrary. Consider just a single implementation consideration: How will lesions that are less than 3 voxels in size be removed from consideration? One option is to simply remove 1-2 voxel lesions from the ground truth and the prediction. But in that case, a 3 voxel lesion predicted where a 2 voxel lesion exists in ground truth would be considered a FP. Does this make sense considering the uncertainty in the underlying segmentations? In practice, we've found that it is not uncommon for different implementations of this metric to produce very different detection curves.

Dataset-Level metrics are calculated at the trial level. That is, TP, FP, and FN counts are aggregated over all patients in a trial at a particular threshold. Aggregated counts are then used to calculate a single TPR, PPV, and F1-score.

3.5 Summary

In this chapter, we reviewed the fundamentals required to understand and evaluate the upcoming three chapters. Specifically, we discussed deep learning, focusing on convolutional neural networks and the procedure through which they are trained, namely backpropagation. We then discussed the segmentation and detection metrics we use to evaluate our models, emphasizing the importance of detection metrics in problems that contain multiple focal pathologies that span a range of sizes. In the following chapter, we introduces our approach, discussing the importance of the engineering process in building deep learning systems for focal pathology segmentation.

Chapter 4

Engineering Deep Learning Systems

Deep learning became more popular in the context of medical imaging as GPUs with increased computation/memory became available and with the development of more easily trained architectures and better optimization algorithms. As solutions to the unique challenges posed by applying deep learning in the context of medical imaging became available, deep learning began to significantly advance the state-of-the-art across the medical image segmentation field [38]. However, given the shear number of codependent design decisions involved when designing a model, and given the high computational cost of training a model (days or even weeks), model configuration and hyperparameter optimization necessarily becomes an iterative process, highly dependent on the skill, experience, and computation available to individual researchers. This process inevitably leads to significant variability in results as reported for any method, including baselines. This is further complicated by subtle *implementation* errors and/or the failure to develop and employ subtle engineering 'tricks', which in practice can impact results just as much as the choice of method itself. After discussing the experimental methodology, we discuss the process of engineering a deep learning pipeline, demonstrating the importance of several key implementation decisions, and the implications of these decisions on the model development and analysis process. Next, we examine several important configuration and hyperparameter decisions, demonstrating the importance of domain knowledge, in

both deep learning and the problem under consideration, to effectively configure and tune deep learning models for focal pathology segmentation.

4.1 Methodology

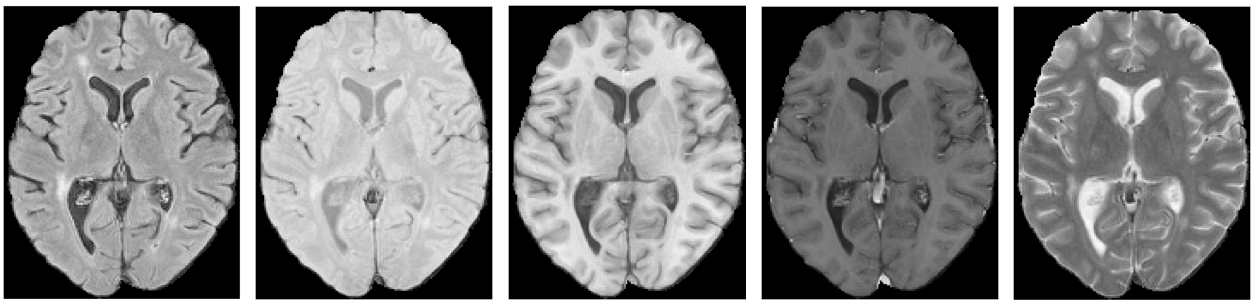
This section describes the model and dataset. Since each experiment can take up to one week to complete, the base approach for each set of experiments will differ modestly from the exact configuration we describe here. In each experiment, important configuration differences, if they exist, are identified.

4.1.1 Dataset

We describe the datasets that we conduct experiments on in this chapter. The vast majority of experiments were conducted on the proprietary Multiple Sclerosis clinical trial dataset we describe in the next section for the task of T2 lesion segmentation. Otherwise, a single experiment is conducted on the MSSEG-2 dataset [14] for the task of segmenting *New* T2 lesions that develop between two sequential brain scans.

Proprietary MS Clinical Trial Dataset (w/ T2-Lesion Labels)

We make use of a large proprietary, multi-center, multi-scanner clinical trial dataset including 1000 patients. Each patient sample consists of 5 MRI sequences (T1-Weighted, T1-Weighted with Gadolinium Contrast Agent, T2-Weighted, Fluid Attenuated Inverse Recovery, and Proton Density), with an example provided in Figure 2.2. All MRI sequences were acquired at 1mm x 1mm x 3mm resolution. T2 lesion labels were generated at the end of the clinical trial, and were produced through an external process where trained expert annotators manually corrected a proprietary automated segmentation method. We use a 60/20/20 split for training/validation/testing respectively. An example of the data can be found in Figure 4.1.

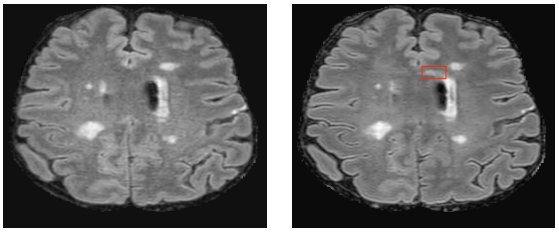


(a) FLAIR **(b)** PDW **(c)** T1P **(d)** T1C **(e)** T2W

Figure 4.1: Sample Image from a Multiple Sclerosis dataset acquired over the course of a clinical trial. We visualize five different MRI *sequences*: Fluid-Attenuated Inversion Recovery (FLAIR), Proton-Density Weighted (PDW), T1-Weighted (T1P), T1-Weighted w/ Gadolinium Contrast (T1C), T2-Weighted (T2W). Images courtesy of NeuroRX.

Public MSSEG-2 Dataset (w/ New T2 Lesion Labels)

New T2 lesion detection is the process of identifying T2 lesions that develop in the interval between two scans taken at two different points in time (months or years apart). An example of new T2 lesion detection can be found in Figure 4.2. We make use of the dataset made available through the public 2021 MSSEG-2 New T2 Lesion Segmentation and Detection Challenge [14]. The dataset contains a total of 40 samples, with each sample consisting of two Fluid Attenuated Inverse Recovery (FLAIR) scans of the same patient taken at two different points in time, and a label identifying New T2 lesion voxels.



T0 T1

Figure 4.2: New T2 segmentation/detection task. FLAIR scan of the same subject at two different points in time. Samples from the MSSEG-2 Dataset [14]. Apparent differences in brain shape and size as compared to Figure 4.1 are the result of differences in *scanner resolution*.

4.1.2 Model

Inspired by nnU-Net, we develop a robust and accurate approach for focal pathology segmentation. The full architecture can be found in Figure 4.3, and important hyperparameters in Table 4.1. On the proprietary clinical trial dataset, the inputs are the five

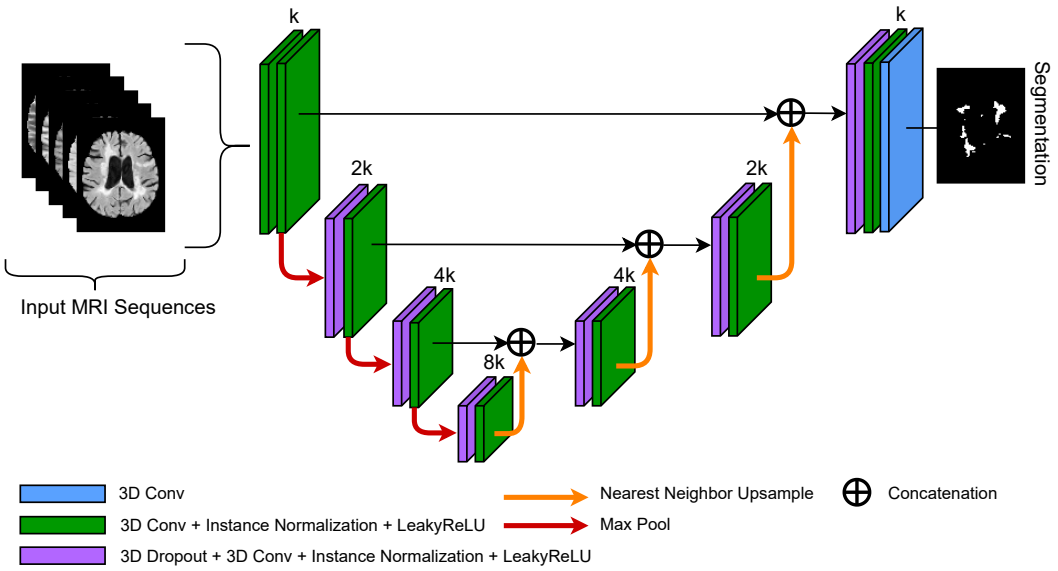


Figure 4.3: Overview of modified nnU-Net [39] architecture used to segment Multiple Sclerosis T2 lesions.

Hyperparameter	**Value**
Capacity (K)	32
Batch Size	1
Optimizer	Adam
Learning Rate	1E-4
Exponential Moving Average Metric	Dataset-Level Average Precision
Exponential Moving Average Metric Beta	0.90
Learning Rate Scheduler Factor	0.20
Learning Rate Scheduler Patience	20 Epochs
Early Stopping Patience	40 Epochs
Dropout Probability	0.20
Random Flip (Symmetric Axis) Probability	0.50
Random Affine Augmentation Probability	0.80
Random Affine Augmentation Rotate	U(-0.14, 0.14)
Random Affine Augmentation Shear	U(-0.08, 0.08)
Random Affine Augmentation Scale	U(0.92, 1.08)
Random Contrast Augmentation Probability	0.80
Random Contrast Augmentation Gamma	U(0.67, 1.50)

Table 4.1: Hyperparameters for the proposed method. 'U(a, b)' represents a uniform distribution defined between a and b. The criteria for the learning rate schedule and early stopping are based on the exponential moving average of the specified metric.

available MRI sequences, and the output is the predicted T2 lesion segmentation map. On the MSSEG-2 dataset, the inputs are the two FLAIR sequences (one from each time point), and the output is the predicted New T2 lesion segmentation map. Both datasets are described in Section 4.1.1.

It is important to note that, in many respects, our approach is focused more on the engineering process involved in implementing, configuring, and optimizing deep learning systems rather than on a particular configuration or a specific set of hyperparameters. We emphasize this point because the optimal system configuration or hyperparameters will vary significantly depending on the characteristics of the dataset at hand [40]. While many of the insights we describe here will transfer to other medical image segmentation or detection tasks to an extent, the optimal system configuration or hyperparameters will always be dependent on the dataset.

4.2 Engineering a Deep Learning Pipeline

In this section, we discuss several key implementation decisions, and the implications of these decisions on the model development and analysis process.

4.2.1 Metrics

We explore several implementation-level concepts on the calculation and interpretation of metrics. Since metrics are the measure by which model performance is evaluated, an understanding of how these metrics are implemented, particularly with respect to any assumptions that are made, is crucial to fully evaluate and fairly compare methods.

Thresholding

Binary segmentation tasks involve classifying each voxel as one class or the other. To do so, a particular threshold must be selected to produce a binary output (0 or 1) from the real-valued output of the segmentation model. Selecting a threshold should not be arbitrary, as there exists inherent trade-offs between, for instance, TPR and PPV, with higher thresholds (more confident predictions) resulting in a lower TPR but a higher PPV. Therefore, what constitutes an 'optimal threshold' is highly dependent on the purpose of the output segmentation. If the output segmentation is to be corrected by a human rater,

we may use a lower threshold in order to identify as many TP as possible, and then direct the human rater to 'clean up' any remaining FP. On the other hand, if the output of the model is to be utilized as-is, we may use a threshold corresponding to the maximum F1 (a.k.a. maximum DICE).

Whatever the purpose of the output segmentation, it is important to note that an optimal threshold can not be selected in advance, rather it must be computed based on the segmentation output of the model. As a result, it is necessary to calculate TPR and PPV at a wide variety of thresholds to produce what is known as a precision-recall curve. From this curve, an optimal threshold can be selected based on the purpose of the produced segmentation output. Figure 4.4 shows F1 plotted against the corresponding thresholds for a segmentation model trained with weighted cross entropy. From this figure, we see why the practice of selecting an arbitrary threshold such as 0.00 (or 0.50 after the sigmoid) is ill-advised. Indeed, using a threshold of 0.00 will result in a DICE of 0.56, which is clearly sub-optimal when considering that a threshold of 4.7 will result in a DICE of 0.79. Interestingly, this was also discussed in a 2015 publication by Sculley et al. [85] where "Fixed Thresholds in Dynamic Systems" was described as a common source of error when incorporating machine learning into commercial products.

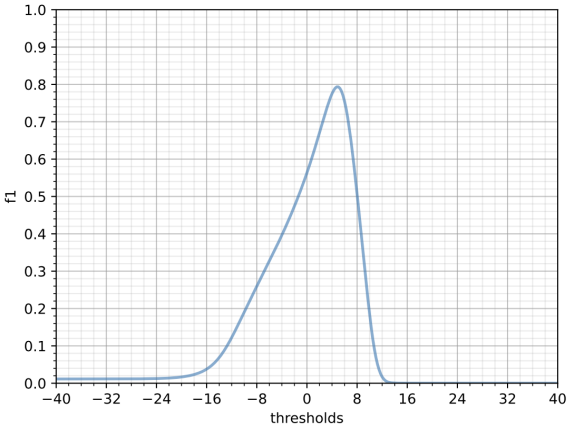


Figure 4.4: Voxel-Level F1 by threshold for a segmentation model with a weighted loss function. Thresholds correspond to σ^{-1} (e.g. thresholds -3.00, 0.00, and 3.00 before the sigmoid correspond to 0.05, 0.50, and 0.95 after the sigmoid). In this example, the common practice of selecting a pre-sigmoid threshold of 0.00 (0.50 after the sigmoid) would clearly be suboptimal.

Notwithstanding the need to compute metrics at multiple thresholds, it is generally not feasible to do so during model training without the use of multi-processing, a programming paradigm that not all researchers have the ability to effectively employ. If researchers are unable to effectively employ multiprocessing in their deep learning pipelines,

(a) Validation Loss

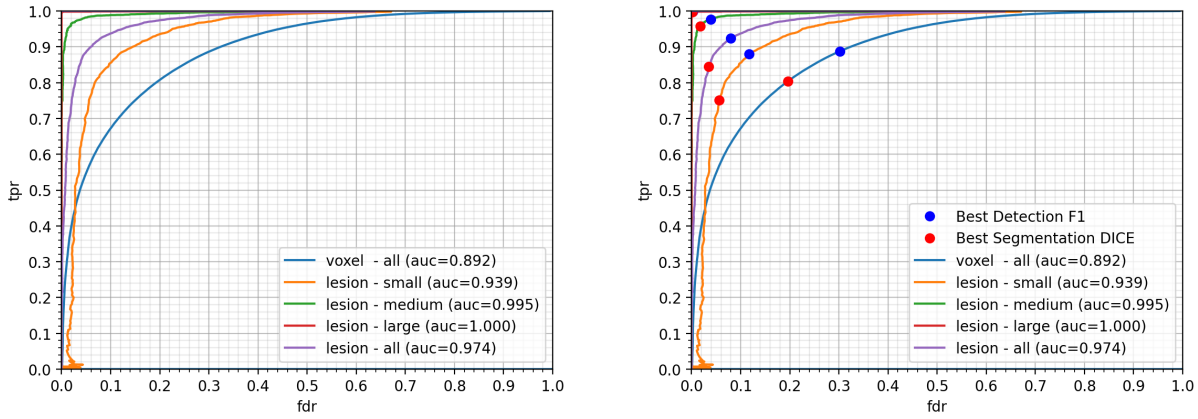
(b) Validation F1-score (DICE)

Figure 4.5: Left: Validation loss during training; Right: Validation F1-score (DICE) during training.

researchers may instead monitor metrics at an arbitrary threshold, or use a surrogate metric such as the loss. Previously, we showed why monitoring a metric at an arbitrary threshold is bad practice (see Figure 4.4). In Figure 4.5, we show why monitoring the metric of interest, the F1-score, is more important than monitoring the loss, whose only purpose is to act as a differential surrogate for the metric that we actually care about (F1-score). Notice that F1-score continues to increase long after the loss has been minimized, showing why monitoring a non-linear metric, such as the loss function, can be suboptimal.

Operating Point Correspondence

When presenting more than one precision-recall (or ROC curve), it is important to identify corresponding operating points (thresholds) on each curve. We illustrate this by way of the following example. If we examine Figure 4.6a, we see an example plotting a voxel-level segmentation curve, as well as a set of lesion-level detection curves. First, consider that within the medical community, it is a convention to select operating points with an fdr of 0.20 or less [25]. Next, consider that for the purposes of this example, we decide that a model with an all lesion detection fdr/tpr of $0.08/0.92$ would be optimal. We might then be tempted to report that overall, our model achieves lesion detection fdr/tpr of $0.08/0.92$ and a voxel segmentation fdr/tpr of $0.08/0.63$. However, looking at Figure 4.6, we can



(a) Operating Points Not Identified (Common Practice)

(b) Operating Points Identified

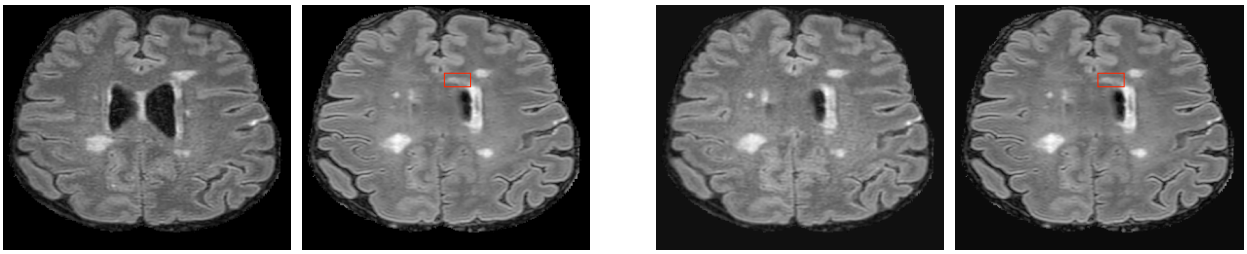
Figure 4.6: Example showing the need to identify corresponding operating points when plotting more than a single curve. TPR vs. FDR curves are plotted for voxel-level segmentation, and size-stratified lesion-level detection. The **blue dot** represents the operating point (a.k.a. threshold) that results in the maximum detection F1-score on the 'lesion - all' curve. The **red dot** represents the operating point (a.k.a. threshold) that results in the maximum segmentation DICE on the 'voxel - all' curve.

see that this is clearly not correct, as at the all lesion detection fdr/tpr of $0.08/0.92$, the corresponding operating point on the voxel-level curve is an fdr/tpr of $0.30/0.89$, which is much different than the voxel-level fdr/tpr that we might have otherwised assumed.

4.2.2 Data and Model Analysis

Metrics can serve an important role in the method development process that goes far beyond hyperparameter tuning. The right metrics, paired with adequate visualizations, can help identify cases where the method does poorly. In some cases, this information can be used to further develop the model or method. For instance, consider Figure 4.7b, an example of a New T2 lesion segmentation task, where the goal is to segment T2 lesions that develop in the interval between two scans taken at two different points in time (months or years apart). In order to identify where the new lesions have developed, the two scans must be properly aligned [*]. However, the automated methods that are used to align brain

[*] *Registered* in the literature. *Registration*, is the process by which images are registered.



T0 T1 T0 T1

(a) Not Aligned **(b)** Aligned

Figure 4.7: New T2 segmentation/detection task. FLAIR scan at two different points in time. Samples from the MSSEG-2 Dataset [14].

scans are not foolproof, and alignment failures do happen on occasion. In Figure 4.7a, we provide an example of one such case we encountered in practice on the MSSEG-2 dataset [14]; identified after a close analysis of per-sample metrics. After analyzing the sample in more detail, it become clear that a non-linear alignment method would be better suited for this dataset. In Figure 4.7b, we show the result of re-aligning the two samples with symmetric diffeomorphic image registration [5]. This shows how close model and data analysis can influence the engineering of the overall system.

4.2.3 Computational Efficiency

Computational efficiency is particularly important when applying deep learning in a medical image context. Medical images are generally three dimensional, requiring significantly more GPU memory (and computation) to process compared to natural images. Training can take days, or even weeks, on some datasets. Therefore, it is important to make all operations as efficient as possible. Below, we describe three key techniques to maximize computational efficiency.

Data Preprocessing

Despite being quite simple, tightly cropping around the brain region is an effective way to save on memory and computation. In our main Multiple Sclerosis dataset (described

in Section 4.1.1), images can be cropped to reduce overall image size by 46%. Given that processing time is directly proportional to the size of the image, training time will be reduced by roughly the same amount. Furthermore, roughly an equivalent drop in GPU memory utilization can enable improvements to the model architecture (e.g. capacity) or optimization strategy (e.g. batch size) that may have not been possible otherwise.

Model

Some underlying knowledge of the way NumPy/PyTorch allocates array memory, and understanding the mathematical equivalence of linear operations can provide additional GPU memory savings. The UNet architecture requires a concatenation operation that fuses activations from the contracting and expanding paths before the convolution. Given that PyTorch requires that tensors be contiguous, the concatenation operation must duplicate activations from both paths that must then be stored as a newly fused tensor (thus utilizing additional memory). However, if we note that a single convolution on a fused tensor can be expressed (equivalently) as the sum of two independent convolutions on the two un-fused tensors, then there is no need to explicitly perform the concatenation operation. This simple insight provides additional memory savings (roughly 11% in our experiments). It is important to note that this is not equivalent to simply adding the activations of the contracting and expanding paths together. This computational 'trick' may also apply to architectures that make even heavier use of the concatenation operation (e.g. DenseNets [41]).

Metrics

Computing the precision-recall curve during training and validation is important for monitoring during training and for model selection. However, in practice, computing metrics, such as the precision-recall curve, can be a real bottleneck, with a poorly engineered metric implementation slowing down training by an order of magnitude or more. In Table 4.2, we provide wall clock times for several different metric implementations.

#	Case	Epoch Time (Seconds)
1	Metrics Within Cropped Volume	1620
2	Metrics Within Brain Region	460
3	Metrics Within Brain Region + Multiprocessing	170
4	Metrics Within Brain Region + Multiprocessing + Shared Memory	155
5	Metrics Not Computed	145

Table 4.2: Per-epoch wall clock time for several different metric implementations.

We show that by computing metrics within the region of interest, by making use of multiprocessing, and by using shared memory for inter-process communication, the precision-recall curve can be computed without significantly slowing down training. By excluding irrelevant voxels from consideration when computing the precision-recall curve, a significant speedup can be observed (2 vs. 1). By using multiprocessing to compute metrics in a separate process (running on its own CPU core), metrics can be computed in parallel while training continues in the main process (3 vs. 2). Finally, by placing each batch of voxels into memory shared between processes, the high CPU cost incurred when piping large arrays between processes can be avoided (4 vs. 3). For comparison, the epoch time for a run where metrics are not computed can be found in (5). Overall, we show that a well-engineered metric implementation can compute real-time metrics without seriously impacting overall training time.

4.3 Establishing a Well-Optimized UNet

In this section, we discuss the surprisingly challenging task of establishing a strong segmentation model in the context of focal pathology segmentation. We perform an ablation study of the method we propose in Section 4.1 in which we examine several important configuration and hyperparameter decisions, through which we demonstrate the substantial benefit that domain knowledge provides when optimizing a UNet for focal pathology segmentation. We stress that the optimization *process* we discuss here, which enables us to achieve optimal (or near optimal) performance, is only made possible by the careful application of the engineering process we discuss in Section 4.2. We justify our fo-

cus on the engineering process by baselining against previously published results on the same dataset, outperforming competing methods by a large margin in both voxel-wise segmentation and lesion-level detection performance.

4.3.1 Experiments and Results

In this section, we run a series of experiments to demonstrate the importance of several configuration and hyperparameter choices. Given that even computationally efficient UNet models can take up to seven days to train, using automatic methods to configure a method (i.e. choosing an optimizer, loss function, model architecture, data preprocessing strategy, etc.) or to tune hyperparameters (e.g. learning rate, dropout, capacity, etc.) is not computationally feasible. As a result, proposed methods must be configured and tuned by hand, requiring an intuitive understanding of how various configuration and hyperparameters choices interact to properly optimize a particular method, to the extent possible, within a limited compute budget. Through a number of experiments, we identify several of these important configuration and hyperparameter choices, illustrating the need for careful consideration of such choices to obtain optimal (or near optimal) performance levels. Results for the proposed method, along with a comparison against results published by others on the same dataset, can be found at the end of the chapter in Section 4.4.

Overcoming Optimization Difficulties

This set of experiments aims to illustrate a common optimization difficulty faced when optimizing deep learning models in the context of medical image segmentation. Although we demonstrate this issue in the context of MS lesion segmentation, *gradient noise* is a well known phenomena inherent in any training process that involves *stochastic* gradient decent [22]. However, in contrast to natural images, practical considerations, such as GPU memory, become a limitation in the context of 3D medical image segmentation,

forcing the use of small batch sizes, resulting in much more noisy gradients, ultimately making medical image segmentation models more difficult to train.

Figure 4.8 depicts validation results for several variants of the UNet architecture. The four experiments include the following: (i) Baseline UNet w/ SGD+Momentum (Blue), (ii) Baseline UNet w/ Dropout and SGD+Momentum (Pink), (iii) Baseline UNet w/ Adam (Purple), (iv) Baseline UNet w/ Dropout and Adam. Results show several key interactions between Adam (which by taking into account the variance of gradients between steps, mitigates the effect of gradient noise [51]) and dropout (which exacerbates gradient noise [92]). Comparing the blue and pink curves, we can see that adding dropout to a baseline UNet model introduces enough gradient noise to completely prevent the model from converging. Without understanding / knowing about the concept of gradient noise, one might assume that dropout *decreases* performance in UNet architectures. However, if we compare the purple and orange curves, we see that when using the Adam optimizer, dropout marginally *increases* performance. Overall, this set of experiments shows why researchers must have an understanding of how various configuration and hyperparameter choices can impact training dynamics. Indeed, with so many codependent hyperparameters, automatic optimization methods are just not computationally feasible, and without sufficient knowledge or experience, researchers may find it difficult to properly hand-tune a model, potentially resulting in a suboptimal method or baseline. Since an objective comparison of two competing methods requires that both methods be well-optimized, a poorly optimized model potentially raises concerns over the reproducibility of published results, and may partially explain why nnU-Net has achieved SOTA performance on dozens of medical image segmentation tasks, despite nnU-Net being a baseline method [39].

Test results, which can be found in Table 4.3, are comparable to the validation results presented in Figure 4.8, supporting the conclusion that our validation/test sets are of sufficient size to serve as a proxy for generalization performance on this dataset.

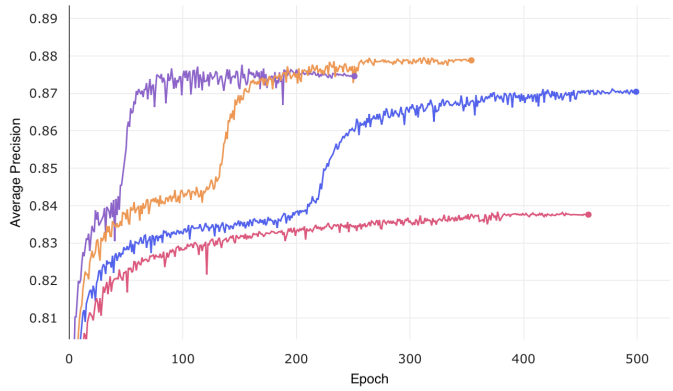


Figure 4.8: Validation voxel-level average precision curves during training. Blue: SGD+Momentum, Red: SGD+Momentum w/ Dropout, Purple: Adam, Orange: Adam w/ Dropout.

Method	AP	DICE
SGDM	0.878	0.790
SGDM + DO	0.847	0.755
Adam	0.882	0.794
Adam + DO	0.885	0.796

Table 4.3: Test voxel-level average precision and voxel-level DICE. 'DO' is short for Dropout [92]

Adapting to the Problem

This set of experiments aims to show the importance of adapting any approach or technique to the problem at hand. In particular, we consider data normalization, an important and well documented part of any deep learning approach [91]. Normalizing data such that it has zero mean and unit variance (a.k.a. z-score normalization) ensures that there is sufficient contrast between different tissue types and is important to insure optimal gradient propagation [37, 96]. However, despite how simple the concept of normalization may appear to be, we show that it must adapted to the problem at hand to ensure that the *region of interest*, in particular, is properly normalized.

The original UNet publication applied the model to a 2D (512x512) cell histology segmentation problem [65, 78], with the cells to be segmented (i.e. the region of interest) spanning the entire image. However, if we take a look at the data sample in Figure 2.2, we can see that the region of interest, the brain region, is spans only part of the image, with there being zero probability of a lesion outside this region. As a result, it only makes sense to normalize within the brain region, as this region is where the model must learn an effective representation to distinguish a lesion from health tissue. On the other hand, if we instead normalize over the full volume, the zeros in the background are included in the mean and standard deviation calculations, ultimately resulting in a intensity dis-

tribution within the brain region that is not centered and has non-unit variance. Indeed, even when the data volume is cropped to a minimum common size, the brain region, on average, makes up only $\sim 25\%$ of the entire 3D volume. This consideration extends to the loss weighting function as well, since a commonly selected default positive weight is $\frac{\text{\# negative voxels}}{\text{\# positive voxels}}$ (e.g. PyTorch documentation [77]), an increase in negative voxels (from outside the brain mask), will increase the overall loss weighting (in our experiments, the loss weighting increased from 170 to 673). If we constrain the calculation of this loss weight to within the brain region (and compute the loss only within the brain region), a less extreme loss weight is used. Indeed, based on the results obtained, we might also consider removing the loss weight altogether, to verify if the default loss weighing is actually optimal in this context. Although these steps may seem elementary, narrowing the region that we consider when designing a normalization or loss weighting strategy, and more broadly, moving away from default configurations and hyperparameters can have a significant performance impact. Despite this reality, in our experience these factors are often overlooked.

To demonstrate the importance of adapting a particular method to the problem at hand, we conduct three experiments: (i) Full-Volume Data Normalization and Full-Volume Loss Weighting (Gray), (ii) Brain Region Data Normalization and Brain Region Loss Weighting (Blue), (iii) Brain Region Data Normalization and No Loss Weighting (Green). 'Full-Volume Data Normalization' refers to Z-Score normalization applied over the full data volume, and 'Full-Volume Loss Weighting' refers to computing the loss weighing based on the class distribution of voxels across the entire image. 'Brain Region Data Normalization' refers to only normalizing data within the brain region (setting voxels outside the brain region to zero), and 'Brain Region Loss Weighting' refers to computing the loss weighing based on the class distribution of voxels *within the brain region*. 'No Loss Weighting' means that we ignore the suggested default weighing, using a non-weighted loss function instead.

Figure 4.9 depicts validation results for several variants of the nn-UNet architecture. We can see that by adjusting our *region of interest*, we can enjoy a significant performance improvement (compare the gray and blue curves). Additionally, since decreasing the loss weight improved performance, we may then consider ignoring the default loss weighting scheme altogether, and try an unweighted loss function (green), which we can see actually outperforms the default. On closer inspection, it is also clear that these decisions can also effect the number of epochs required to train the model, with the best model converging much more quickly than the worst one. Given that models for medical image segmentation can take up to a week to train, a significant decrease in training time can be an important advantage. Indeed, one of the most common learning rate schedules involves cutting the learning rate when the loss, or metric of interest, plateaus. Thus, a method that converges too slowly is more likely to have its learning rate cut prematurely, leading to suboptimal results.

Test results, which can be found in Table 4.4, are comparable to the results we report on validation in Figure 4.9. Overall, our results show the importance of carefully adapting an approach to the problem at hand, with remarkably simple configuration and hyperparameter choices having significant performance impacts. Furthermore, we saw that by carefully analyzing our results, we were able to infer that an experiment that further decreased the positive class loss weight was warranted.

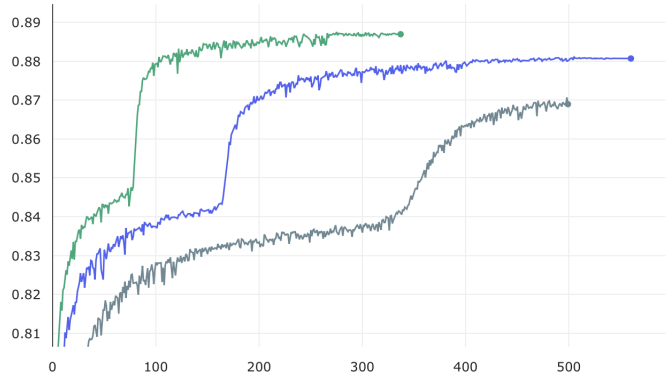


Figure 4.9: Validation voxel-level average precision curves. Grey: Weighted Loss & Normalization (Full Volume), Blue: Weighted Loss & Normalization (Brain Region), Green: Unweighted Loss & Normalization (Brain Region)

Method	AP	DICE
Full Volume (N&LW)	0.878	0.791
Brain Region (N&LW)	0.886	0.798
Brain Region (N)	0.892	0.804

Table 4.4: Test voxel-level average precision and voxel-level DICE. 'N' is short for 'Normalization'. 'LW' is short for 'Loss Weighting'.

Considering the Relationship Between Effect Size and Dataset Size

This set of experiments aims to show the relationship between the size of the dataset, and the effect size of a particular configuration or hyperparameter choice. We demonstrate this in the context of an important configuration decision that we touched on earlier. Specifically, we compare several data normalization schemes, contrasting results obtained on the full training set against those obtained on a small subset of full training set. We show that while important configuration and hyperparameter decisions may only have a modest impact on model performance when training on a large dataset, these same decisions can have a large impact on performance when training on a small dataset.

Results on the full training set for z-score normalization (within the brain region), z-score normalization (over the full volume), and 0-1 normalization (i.e. subtract the minimum and divide by the maximum) can be found in Figure 4.10 and Table 4.5. Based on these results, we can see that while the normalization scheme used had a significant impact on overall training time, the impact on performance was *modest*, with z-score normalization (within the brain region) performing marginally better than the other two normalization techniques.

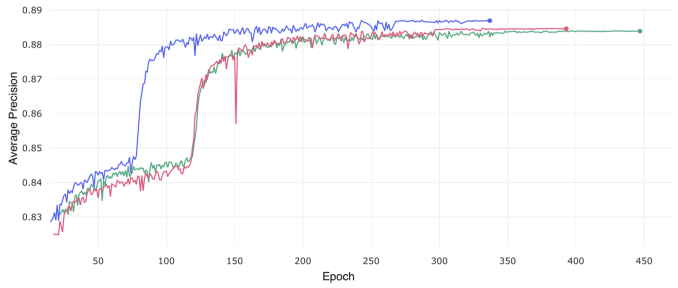


Figure 4.10: Validation average precision for a number of different normalization strategies. Blue: Z-Score Standardization within Brain Region, Pink: Z-Score Standardization over Entire Volume, Green: Zero-One Normalization.

Normalization	AP	DICE
Brain Region	0.892	0.804
Full Volume	0.891	0.803
Zero-One	0.889	0.801

Table 4.5: Test voxel-level average precision for a number of different normalization methods.

Noting a relatively small difference in performance between normalization schemes on the full dataset, we repeat the same experiment on a small subset (20 samples) of the full dataset (600 samples). The validation and testing sets remain identical. Results can be found in Figure 4.11 and Table 4.6. In the low data regime, the performance gap between methods in much more pronounced, demonstrating that data normalization, while usually considered a relatively simple implementation detail, can have a significant performance impact when data is scarce. This demonstrates the importance of considering dataset size, along with other characteristics of the dataset, when attempting to extrapolate a conclusion drawn on one dataset to another (particularly with respect to the magnitude of the effect).

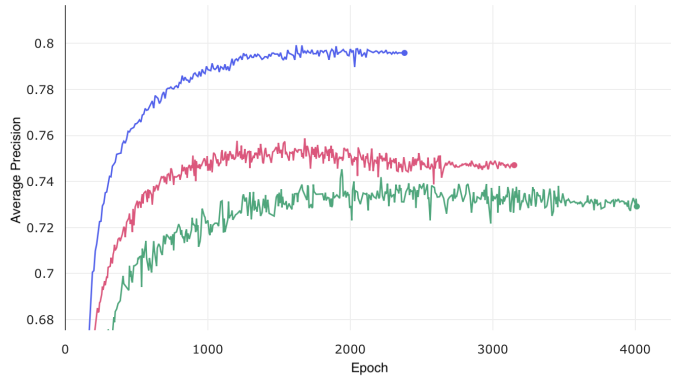


Figure 4.11: Validation average precision for a number of different normalization strategies. Blue: Z-Score Standardization within Brain Region, Green: Zero-One Normalization.

Normalization	AP	DICE
Brain Region	0.809	0.732
Full Volume	0.767	0.700
Zero-One	0.742	0.681

Table 4.6: Test voxel-level average precision for a number of different normalization methods.

Understanding the Role of Model Capacity

This set of experiments aims to demonstrate the role of model capacity on several performance measures. We also use this set of experiments to illustrate the importance of considering operating point correspondence (see Section 4.2.1) when reporting metrics, showing that reporting multiple precision-recall curves without identifying corresponding operating points can make comparing multiple models difficult, potentially resulting is conclusions that are erroneous.

Although the relationship between model capacity and performance is known [94], in medical imaging, practical considerations, particularly GPU memory, typically force the use of reduced capacity models. Deep learning practitioners must find the right balance between model capacity, normalization layers (instance normalization, batch normalization, etc.), patch size, and batch size, which are the primary drivers of GPU memory consumption. However, given the compute required to train deep learning models, deep learning researchers may fix the capacity, patch size, and batch size of their base model, only then introducing additional mechanisms to further improve performance. This practice is a necessary consequence of the massive compute requirements imposed by processing 3D medical data by even the most well-established baselines (e.g. UNet [78]). However, having a sufficiently broad knowledge base, such as an understanding of the

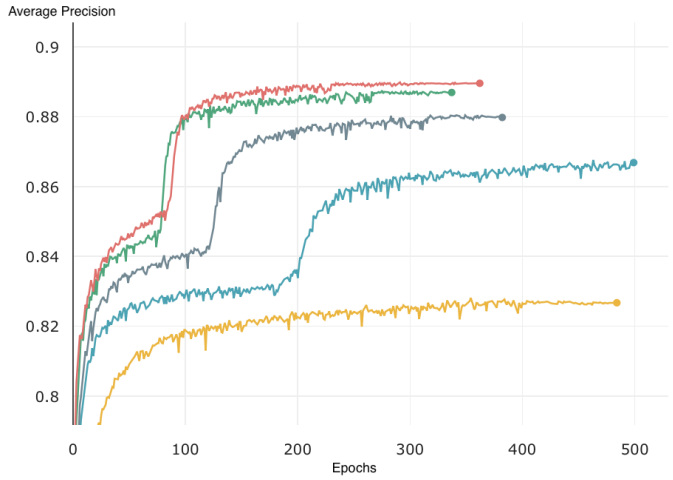


Figure 4.12: Validation average precision curves for several models of different capacity (K). Yellow: K=4, Cyan: K=8, Grey: K=16, Green: K=32, Red: K=64.

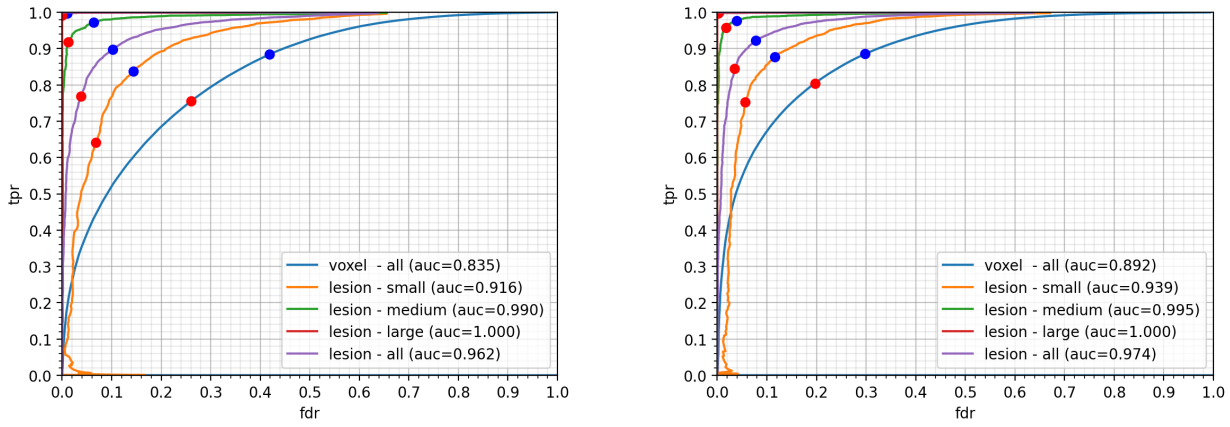
Capacity	AP	DICE
K=4	0.835	0.747
K=8	0.857	0.765
K=16	0.886	0.798
K=32	0.892	0.804
K=64	0.895	0.807

Table 4.7: Test average precision (AP) and DICE (computed at the optimal voxel-level operating point).

relationship between model capacity and Double Decent [72], can help guide the decision as to which experiments are actually run. Indeed, the experiments we present in this section illustrate the large magnitude of the performance differences that small changes in model capacity can bring about, particularly near the 'interpolation threshold' that underlies the Double Descent phenomena.

Results for several models of different capacity can be found in Figure 4.12 and Table 4.7. We can see that increasing model capacity can make a significant difference, particularly near the 'interpolation threshold' between the K=4 and K=8 models. After this point, performance improves logarithmic as capacity increases from K=8 to K=64. Given that the epoch at which the model undergoes double decent is similar between the K=32 and K=64 models, we believe the K=32 model best optimizes the trade off between performance and training time, and begins to fully converge by epoch 100.

Beyond the voxel-level performance improvement we just discussed, we compare the performance characteristics of a UNet with low capacity (K=4, Figure 4.13a), and a UNet with high capacity (K=32, Figure 4.13b), pointing out several different operating points on the lesion-level and voxel-level precision-recall curves. As we also point out in Section 4.2.1, we note that what might be considered a favorable operating point on the



(a) U-Net (K=4)

(b) U-Net (K=32)

Figure 4.13: Lesion-Level Detection and Voxel-Level Segmentation Results. Left: Low Capacity (K=4). Right: High Capacity (=32). Blue Dot: Operating point with optimal detection F1, Red Dot: Operating point with optimal segmentation DICE.

lesion-level curve will not necessarily translate to a acceptable operating point on the voxel-level curve (and vice-versa). Indeed, for the low capacity model, we see a clear rightward shift (over segmentation) of all operating points on the voxel-level curve compared to the high capacity model, this despite the much smaller performance differences visible on the lesion-level detection curves. We stress the importance of actually selecting and plotting operating points (or reporting results through a table at a fixed operating point) when comparing multiple performance curves.

Qualitative results on the test set between the K=4 and K=32 models, at each model's optimal lesion-level detection operating point can be found in Figure 4.14. We can see that despite similar looking lesion-level detection curves, the operating points themselves can shift significantly between methods. Therefore, results must be presented such that the value of all metrics can be discerned at a fixed operating point. Incidentally, we note that the majority of the published methods we compare our well-trained UNet to *do not* report or otherwise make it possible to identify the value of all metrics at any specific operating point. This shows that while identifying corresponding operating points on multiple curves is a straight forward idea, it is not universally applied in practice.

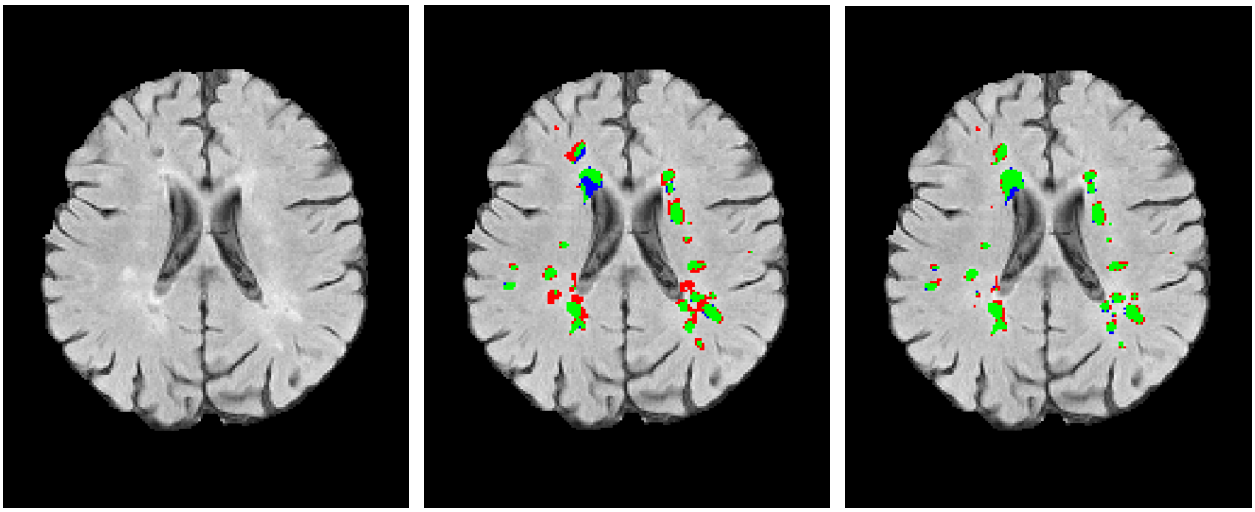


Figure 4.14: Voxel-Level Segmentation Results at the Optimal *Per-Dataset Lesion-Level Detection F1* Left: Low Capacity (K=4). Right: High Capacity (K=32). TP: Green, FP: Red, FN: Blue.

4.3.2 Final Model

In Table 4.8, we present the results of the method we proposed in Section 4.1, engineered through the use of the engineering principles we discussed in Section 4.2 & 4.3, against several published methods that report results on the same dataset. The results we report for third-party methods are taken from their respective publications, obtained from a *different random split* of the dataset. Indeed, we can see that our model outperforms all others by a large margin, achieving higher recall (TPR) at a precision (PPV) of 0.80 on both lesion-level detection, and the voxel-level segmentation measures of performance. [†]

Although the methods we compare against do not explicitly focus on optimizing segmentation or detection performance, they significantly under-perform the approach we've outlined here. And while it may seem surprising that similar methods can differ performance-wise to this extent, our results are actually in-line with those of both nnU-Net, which showed on a number of public datasets that a carefully configured and optimized U-Net can achieve state-of-the-art performance [39], and with Litjens et al., who made the case that "the exact architecture is not the most important determinant in getting a good solution" [60].

[†]Given that competing approaches do not report both segmentation and detection results at a fixed operating point, we instead report the TPR corresponding to a PPV of 0.80 for each metric separately.

#	Method	Type	PPV	TPR
1	Ours	U-Net	0.80	**0.69**
2	Mehta [68] †	U-Net	-	-
3	Sepahvand [87] †	U-Net	0.80	0.55
4	Nair [71] †	U-Net	0.80	0.48
5	Subbanna [93]	MRF	-	-

(a) Per-Scan Voxel-Level Segmentation

#	Method	Type	PPV	TPR
1	Ours	U-Net	0.80	**0.96**
2	Mehta [68] †	U-Net	0.80	0.84
3	Sepahvand [87] †	U-Net	0.80	0.84
4	Nair [71] †	U-Net	0.80	0.77
5	Subbanna [93]	MRF	-	-

(b) Per-Scan Lesion-Level Detection

#	Method	Type	PPV	TPR
1	Ours	U-Net	0.80	**0.80**
2	Mehta [68] †	U-Net	-	-
3	Sepahvand [87] †	U-Net	-	-
4	Nair [71] †	U-Net	-	-
5	Subbanna [93]	MRF	-	-

(c) Per-Dataset Voxel-Level Segmentation

#	Method	Type	PPV	TPR
1	Ours	U-Net	0.82	**0.97**
2	Mehta [68] †	U-Net	-	-
3	Sepahvand [87] †	U-Net	-	-
4	Nair [71] †	U-Net	-	-
5	Subbanna [93]	MRF	0.82	0.79

(d) Per-Dataset Lesion-Level Detection

Table 4.8: Test results of our method alongside those published by several others on the same dataset. We report average *per-scan* metrics, as well as aggregate *per-dataset* metrics, for both voxel-level segmentation and lesion-level detection (see Section 3.4.2 for definitions). Results are rounded to two significant digits. To compare results across publications, we report results at an operating point corresponding to an PPV of roughly 0.80.
†Methods not explicitly focused on optimizing segmentation/detection performance.

4.4 Summary

We discussed the process of engineering a deep learning pipeline, pointing out important details with respect to how metrics are implemented, interpreted, and utilized. We demonstrate the need for real-time metrics, discussing parallelization as one way to achieve this. We consider several ways to improve GPU compute and memory efficiency, important given the computational requirements involved in training a deep learning model on 3D medical data. Next, we considered the process of establishing a well-trained UNet, which given compute requirements, must be configured and tuned by hand. To do this, we discussed overcoming common optimization difficulties, using background domain knowledge to guide configuration decisions and hyperparameter selection. We discussed the need to adapt our model to the problem at hand, noting the need to consider the *region of interest*. We discussed the need to consider the relationship between effect size and dataset size when evaluating the usefulness of a new configuration, or approach. We then

discussed the role of model capacity, demonstrating the important role this has on model performance.

Chapter 5

Lesion Size Reweighting

In the previous chapter, we saw that a well-engineered deep learning pipeline, a principled optimization strategy, and a robust implementation, are several key factors required to obtain impressive segmentation and detection performance in the context of Multiple Sclerosis. Although we've established impressive results, surpassing all previously reported methods, we also noted a key weakness, namely a significant operating point gap between optimal segmentation, and optimal detection performance operating points. On closer analysis, we identified that this gap was driven predominately by small lesions, which showed a large discrepancy in small lesion detection performance between optimal segmentation and optimal detection operating points.

To better understand why a gap between optimal operating points exists, we first turn our attention to the characteristics of voxel-wise loss functions. Specifically, since voxel-wise loss functions operate at the voxel-level, there is an inherent bias towards larger lesions that contain more voxels. As a result, smaller lesions are typically missed at operating points that are optimal for segmentation metrics. And while it is possible to reduce the operating point (i.e. threshold) such that small lesions are detected (see Section 4.3.1), this comes at the cost of considerable over-segmentation. Inspired by the finding that weighting positive voxels will typically shift the optimal operating point upward (see Section 4.4), we can consider applying such a strategy here in a more targeted

way. Specifically, we can apply increased weight to smaller lesions such that we close the gap between the optimal detection and segmentation operating points.

Indeed, recent research has suggested that assigning a weight to each voxel such that the cumulative weight of each structure (i.e. the sum of all voxel weights of each structure) is the same, can be an effective way to improve small lesion performance [88]. However, this approach can be problematic in the contexts we focus on here, where lesions span a very wide range of sizes, which can result in weights that range over several orders of magnitude, resulting in a significant segmentation/detection imbalance and significant training instability.

The content in this chapter is based on published work [75], of which I am first author.

5.1 Methodology

We propose a lesion weighing function, where the objective is to have the optimal detection and segmentation operating points converge by assigning more weight to small lesions than would otherwise be assigned by binary cross entropy. Although small lesions can be weighed more, they should still be assigned less weight than larger lesions, which are typically much more certain. Our conjecture is that assigning too much weight to small lesions can produce suboptimal results.

Formally, each lesion L_j is assigned a weight W_j that is a function of the number of voxels $|L_j|$ that comprise that lesion. In practice, weights must be assigned to individual voxels rather than individual lesions, so we also define the voxel weight w_j, related to W_j via $w_j = \frac{W_j}{|L_j|}$.

$$W_j = |L_j| + \alpha e^{-\frac{1}{\beta}(|L_j|-1)} \qquad w_j = 1 + \frac{\alpha}{|L_j|} e^{-\frac{1}{\beta}(|L_j|-1)} \tag{5.1}$$

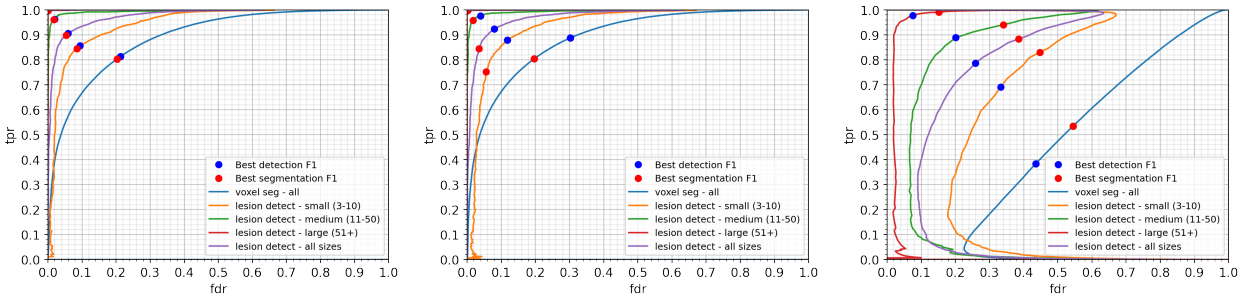
where α and β are hyperparameters such that $\alpha \leq \beta$ to ensure monotonicity in the weight with respect to lesion size. Background (i.e. non lesions) voxels retain a weight of 1.

5.2 Implementation Details

We use the experimental setup detailed in Section 4.1, and use the baseline established as our base model. For all subsequent experiments, we freeze almost all hyperparameters, modifying only the loss function and learning rate. The hyperparameters of the proposed BCE+LSR loss function were tuned on a log_2 scale, with $\alpha = 4$ and $\beta = 4$ performing best in our experiments.

5.3 Experiments and Results

Figure 5.1 shows the TPR vs FDR curves and compares overall segmentation performance with detection performance for small (3-10 voxels), medium (11-50 voxels), and large (51+ voxels) lesions for the proposed BCE+LSR, as compared to BCE and BCE+IW. In the case of BCE+LSR, the optimal operating points for segmentation and detection (red and blue dots) overlap and the method performs well on both tasks. This is in contrast to BCE, for which the optimal operating points are comparatively far apart, and which shows a degree of over-segmentation at the optimal detection operating points (and under-detection at the optimal segmentation operating point, particularly for small lesions). WBCE and FL exhibited performance characteristics similar to BCE. For BCE+IW, the distance between the optimal detection and segmentation operating points is even larger, and the method significantly underperforms all others. Given the significant decrease in performance for BCE+IW relative to both BCE and BCE+LSR, further analysis revealed that BCE+IW applied substantial weight to extremely small lesions. Since the lesion weights computed by BCE+IW ranged over several orders of magnitude, training was extremely unstable. On the other hand, using the proposed BCE+LSR, the weights remain in a reasonable range, upper bounded by $1 + \frac{\alpha}{|L_j|}$. Since smaller lesions are considerably more uncertain, using a weighting scheme with a reasonable upper bound prevented training instability.



(a) BCE+LSR **(b)** BCE **(c)** BCE+IW

Figure 5.1: TPR vs FDR curves: voxel-level segmentation and lesion-level detection. The best detection F1 operating point (blue dot) is based on the *lesion - all* curve. The best segmentation F1 operating point (red dot) is based on the *voxel - all* curve. The closer the operating points the better. The operating points overlap for the proposed BCE+LSR (i.e. BCE+LSR achieves the highest simultaneous detection and segmentation F1).

5.4 Summary

In this chapter, we discussed lesion size reweighing (LSR), a reweighting strategy based on lesion size that closes the gap between the optimal segmentation and the optimal detection operating points. By using this strategy, we enable a single model to achieve optimal performance on both segmentation and detection simultaneously.

Chapter 6

Cohort Bias Adaptation in Aggregated Datasets

In the previous two chapters, we presented work that utilized deep learning for T2 lesion segmentation in the context of Multiple Sclerosis. We evaluated our model with a relatively large *test set*, which would ideally represent the true generalization performance of our model. However, in the context of medical images, the concept of generalization is much more subtle. As detailed in Section 3.4.2, there is no 'ground truth' in medical imaging problems, particularly in contexts where there is considerable uncertainty. When a dataset, typically acquired from single (patient) cohort, is labeled, the exact labeling will be highly dependent on the labeling protocol, as well as the opinion of the associated medical expert. Additionally, even in cases where the labeling protocol and medical expert are held constant, the labeling will be influenced by the medical expert's knowledge about the sample and the cohort as a whole (termed *observer bias*). In this context, it becomes clear why naively pooling multiple datasets from different cohorts together may not increase, and may even decrease, model performance due to cohort-specific biases that ultimately manifest themselves in 'ground truth' labels.

In this chapter, we propose a generalized conditioning framework that learns and accounts for cohort-specific biases across multi-cohort pooled datasets. To do this, we

make use of Conditional Instance Normalization (CIN) [24] to condition the network on the cohort identity of each data sample. This approach, which we refer to as Source-Conditioned Instance Normalization (SCIN)*, enables the training of a single model on a multi-cohort dataset without significant performance loss. This is in contrast to baseline which, unable to condition on auxiliary information, is unable to output a cohort-specific segmentation.

The content of this chapter is based on published work [74], of which I am first author.

6.1 Methodology

We propose a framework that explicitly takes into account cohort-specific biases, enabling a single model to be trained on an aggregate dataset. To do this, we make use of Conditional Instance Normalization (CIN) [24], learning source-specific instance normal-

*for the purposes of this chapter, 'source' and 'cohort' are used interchangeably

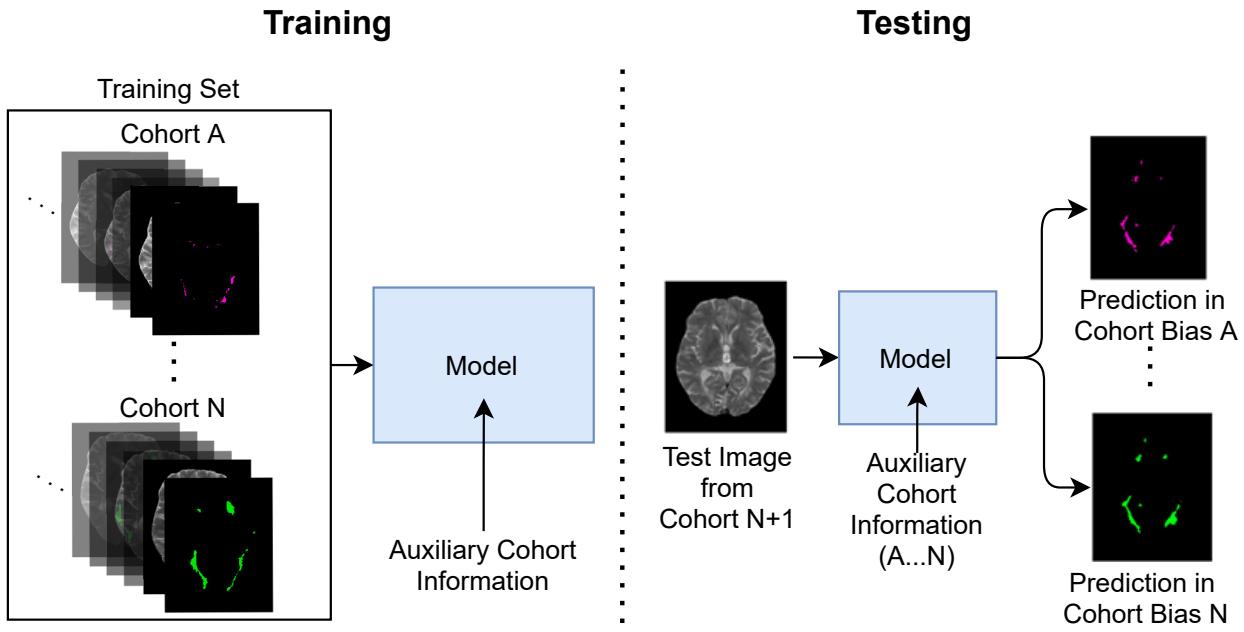


Figure 6.1: System overview showing training on the left and testing on the right. The left shows how we train with multiple cohorts and use auxiliary cohort information to learn the associated bias. On the right is how we use cohort information during testing to generate multiple labels for an image in a desired style.

ization parameters that, by scaling and shifting normalized activations in the network, model source-specific biases. In practice, as each sample is passed through the network, that sample's cohort identity is used to select the corresponding source-specific affine scale/shift parameters which are then used to modulate the activations of the network. During the backward pass, only the affine parameters that were used to scale/shift the activations of the corresponding sample will have a non-zero gradient. As a result, each set of source-specific affine parameters are learned only from the samples that make up that source. Mathimatically, the approach can be described as follows:

$$\text{CIN}(z) = \gamma_s \left(\frac{z - \mu(z)}{\sigma(z)} \right) + \beta_s \tag{6.1}$$

where γ_s and β_s are *source-specific* scale and shift parameters, and where $\mu(z)$ and $\sigma(z)$ are the per-sample per-channel mean and standard deviation. All other learnable parameters, consisting of all convolutional layers in the network, are common. This approach allows the network to model source-specific cohort biases with a relatively small number of parameters, but to otherwise leverage the entire aggregated dataset for most of the common parameters in the network. A full system overview can be found in Figure 6.1.

6.2 Implementation Details

6.2.1 Network Architecture and Training Parameters

We use a network architecture that is nearly identical to the network architecture we defined in Figure 4.1. The only difference is that we swap out the original instance normalization layers that contained just a single set of affine parameters to be used for all samples, with a conditional instance normalization layer, where each source has its own set of affine parameters. The full architecture can be found in Figure 6.2.

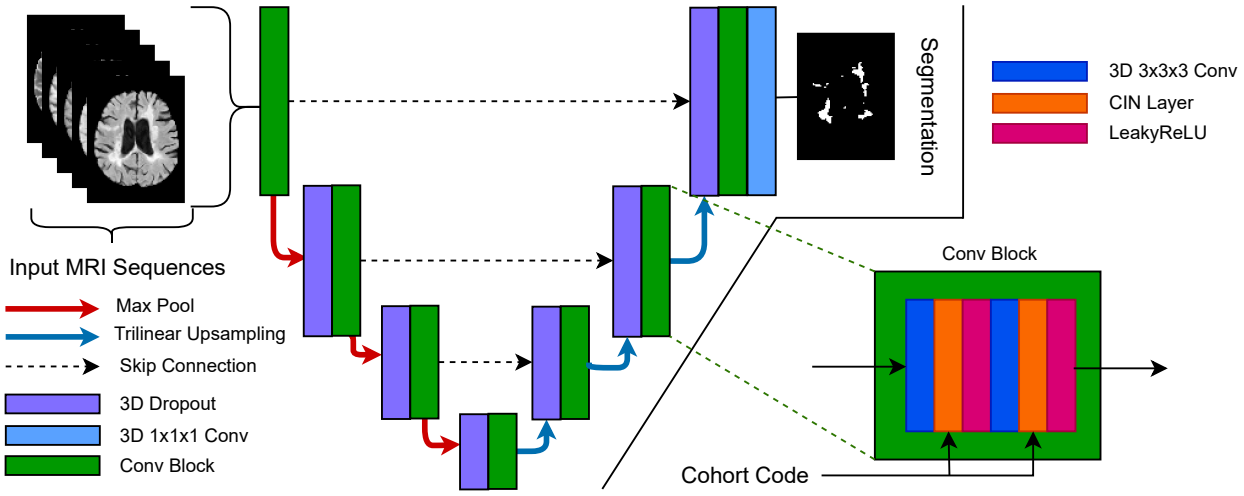


Figure 6.2: The model architecture in nearly identical to the architecture we develop in Chapter 4. Left: Overview of modified nnUNet [39] architecture used to segment MS T2 lesions. Right: Detail of a conv block. It consists of a series of 3D 3x3x3 Convolution Layer, CIN layer, and a LeakyReLU activation layer.

6.2.2 Data Set

We make use of three different datasets, each of which consists of patient samples collected over the course of a different clinical trial. In each clinical trial, each patient sample consists of five MRI sequences (T1-weighted, T1-weighted with gadolinium contrast agent, T2-weighted, Fluid Attenuated Inverse Recovery, and Proton Density) at a 1mm x 1mm x 3mm resolution. Patient samples were labeled at the end of each trial, where expert neuroradiologists manually corrected a proprietary automated segmentation method. We use a 60/20/20 split for training, validation, and testing respectively. More details about each dataset can be found in Table 6.1.

Identity	# Patients	Acquisition Period	Disease Subtype	Disease Stage
Trial-A	1000	2011-2015	Secondary Progressive MS (SPMS)	Late Stage
Trial-B	1000	2008-2011	Relapsing Remitting MS (RRMS)	-
Trial-C	500	2004-2009	Secondary Progressive MS (SPMS)	Early Stage

Table 6.1: Details of the clinical trial datasets used in this chapter.

6.3 Experiments and Results

We perform three different sets of experiments to demonstrate the usefulness of the proposed SCIN approach. In the first experiment, we show that SCIN is able to strategically pool diverse datasets with differing cohort biases. The second experiment demonstrates the clinical utility of SCIN to adapt to new cohort biases with limited available labeled data. Finally, we show that SCIN is able to model complex cohort biases by simulating a type of cohort bias where small lesions (10 voxels or less) were not labeled.

6.3.1 Trial Conditioning

Experiments in this section aim to show how the proposed SCIN approach allows for pooling of data from multiple cohorts while taking into account cohort specific biases. We use two different clinical trial datasets (Trial-A and Trial-B) for these experiments. These two trials were collected several years apart with patients of different disease subtypes. Given that each trial is processed independently and at different points in time, minor differences in site/scanner distribution and annotation style will exist. Together, the patient population, site/scanner distribution, and annotation style create a distinct cohort bias, which we aim to account for with the proposed method.

We train four different models on these datasets: (i) a model trained on only Trial-A, with Instance Norm (IN) [96], (ii) a model trained on only Trial-B (with IN), (iii) a model trained on a naively pooled dataset consisting of Trial-A and Trial-B (with IN), and (iv) a model trained on both Trial-A and Trial-B using the SCIN approach. All four models were tested on the same held-out test set from both trials.

Table 6.2 depicts the performance of the aforementioned models on the hold-out test sets. Results indicate that models trained on only one trial (Row-1 and Row-2) generalize poorly when tested on the other trial. A model trained on the naively pooled dataset, consisting of data from both trials (Row-3), shows better generalization across trials, but still falls short of the performance achieved by each trial-specific model, especially in

	Model	Train Set		Conditioned On		Test Performance	
		Trial-A	Trial-B	Trial-A	Trial-B	Trial-A	Trial-B
1	**Single-Trial**	✓		-		0.793	0.689
2	**Single-Trial**		✓	-		0.715	0.803
3	**Naive-Pooling**	✓	✓	-		0.789	0.748
4	**SCIN-Pooling**	✓	✓	✓		0.794	0.700
5					✓	0.725	0.797

Table 6.2: Dice scores shown on Trial-A and Trial-B test sets for models trained with different combinations of Trial-A and Trial-B training sets. Trial-A and Trial-B training sets each contain 600 patients.

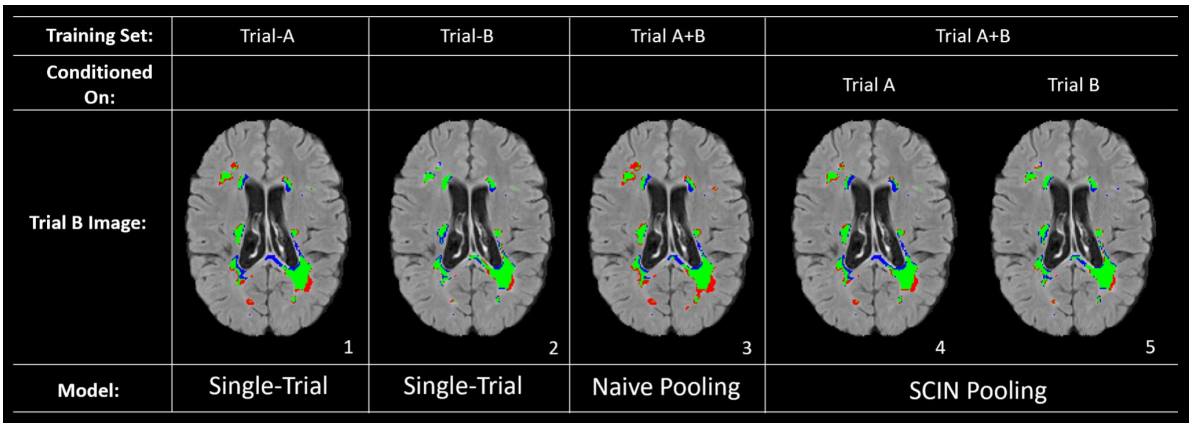


Figure 6.3: Qualitative lesion segmentation labels (red is false positives, blue is false negatives, green is true positives) superimposed onto a FLAIR test image from Trial B. The results are based on the models from Rows 1-5 (left to right) of Table 1. Figure originally part of a manuscript accepted for publication [74].

the case of the Trial-B dataset. At first glance, this might appear surprising given the expectation that a model trained on a larger, pooled dataset should generally perform better relative to a model that has access to less data. However, given the knowledge that biases can exist between cohorts, it is no mystery why the naively pooled model would be unable to generate a labeling consistent with the cohort of the sample, given that the cohort / labeling bias cannot be identified from the image alone. On the other hand, a single SCIN-pooling model conditioned on each trial (Row-4 and Row-5), is able to learn trial-specific parameters to model the bias specific to the trial in question, improving performance relative to the naively-pooled model (Row-3). Note that using the incorrect set of trial-specific parameters with the SCIN-Pooling model (Row-4 and Row-5) results in a performance decline similar to that observed when testing the Single-Trial models

	Model	Fine-Tuned on Trial-C	Conditioned on			Test Performance
			Trial-A	Trial-B	Trial-C	
1	**Naive-Pooling**			-		0.774
2		✓		-		0.819
3	**SCIN-Pooling**		✓			0.763
4				✓		0.806
5		✓			✓	0.834

Table 6.3: Dice scores shown on the Trial-C test set from the Naive-pooling and SCIN-pooling models trained on Trial A and B. Dice scores are also shown for fine-tuned versions of those models, where the IN parameters were tuned using 10 Trial-C samples. Figure originally part of a manuscript accepted for publication [74].

(Row-1 and Row-2) on the corresponding unseen trial. This simply serves as a sanity check, and confirms that the proposed method effectively models the cohort bias of each dataset.

Qualitative results for labels produced by different models on a single Trial-B test case are shown in Figure 6.3. From this, we can see that generating labels on a Trial-B test case using a Single-Trial model trained on the Trial-A dataset (Image-1) leads to an increased number of false positive and false negative voxels. This is also true when testing a naively pooled model (Image-3). On the other hand, the proposed SCIN-pooled model conditioned on Trial-B (Image 5) does not suffer from a significant degradation in segmentation quality, showing that the SCIN approach enables leveraging multiple datasets with different cohort biases without significant performance decrements. Visually, note that the labeling style of the the SCIN-pooled model is similar to that of the corresponding Single-Trial model, showing that SCIN-pooled method is able to learn a cohort-specific bias for each trial.

6.3.2 Fine-tuning to New Cohort Bias

The second set of experiments aims to mimic a clinical situation where large datasets (Trial-A and Trial-B) have a known cohort bias. A new small dataset (Trial-C) is provided with an unidentified cohort bias. We take two pre-trained models (Naively-Pooled and SCIN-pooled) from the previous set of experiments (see Sec 6.3.1), and fine-tune the affine

parameters of the IN/CIN layers with only 10 labeled samples from the Trial-C dataset. Segmentations are performed on a hold-out test set from Trial-C. Similar to Experiment 1, time of collection and disease subtype were the primary differences between the three trials (along with minor differences in scanner/site distribution and labeling protocol).

Table 6.3 depicts the results for this set of experiments. We can see that the performance of the naively-pooled model improves when the IN parameters of the model are fine-tuned (Row-2) compared to no fine-tuning (Row-1). Furthermore, a SCIN-pooled model shows good Trial-C performance when conditioned on Trial-B (Row-4), indicating that Trial-C has similar cohort biases. By fine-tuning the trial-specific CIN layer parameters of this model on 10 samples of Trial-C, we are able to then condition the model on Trial-C during test time. This leads to the highest performance improvement (Row-5) over all models, including fine-tuning the naively pooled model (Compare Row-2 and Row-5). This shows that with SCIN, we can more effectively learn features common to both Trail-A and Trial-B, resulting in better performance after fine-tuning on a hold-out trial.

6.3.3 Accounting for Complex Cohort Biases - Missing Small Lesions

The final set of experiments examine whether the SCIN approach is able to learn complex non-linear cohort biases. Accordingly, we isolate biases that arise solely from different labelling protocols while keep all other factors, such as disease stage and time of collection, constant. To that end, we utilize a held-out clinical trial dataset (Trial-C) and artificially modify half of the dataset by removing small lesions (10 voxels or less) from the provided labels. This can be thought of as being equivalent to a labeling protocol that misses or ignores small lesions (Trial-MSL). The labels of the remaining half of the dataset are not modified in any way (Trial-Orig).

Table 6.4 shows the results for this set of experiments on a non-modified Trial-C test set (Trial-Orig). We report detailed results specific to the detection of small lesions in order to examine whether the proposed strategy learns to account for a labeling style

	Model	Train Set		Conditioned On		Test Performance	
		Trial-Orig	Trial-MSL	Trial-Orig	Trial-MSL	Sm Lesion F1	Voxel Dice
1	**Single-Trial**	✓		-		0.795	0.844
2	**Single-Trial**		✓	-		0.419	0.837
3	**Naive-Pooling**	✓	✓	-		0.790	0.797
4	**SCIN-Pooling**	✓	✓	✓		0.784	0.854
5					✓	0.496	0.850

Table 6.4: Voxel based Dice scores and small lesion detection F1 scores shown on Trial-C (Trial-Orig) held-out test set using models trained on different combinations of the original dataset (Trial-Orig, 150 training patients) and the dataset with missing small lesions (Trial-MSL, 150 training patients). Figure originally part of a manuscript accepted for publication [74].

that ignores small lesions. The results in Row-1 show that when both train set and test set have the same labeling protocol (Trial-Orig), the Single-Trial model performs well according to both lesion-level detection and voxel-level segmentation metrics. On the other hand, when there is a significant shift in the labeling protocol between the train and test set, the Single-Trial model trained on the Trial-MSL dataset (Row-2) exhibits poor small lesion detection performance. The degradation in small lesion detection performance is expected as Trial-MSL has small lesions labeled as background, while the test set (Trial-Orig) has small lesions marked as lesions. The Naively-Pooled model (Row-3), which is trained on both Trial-Orig and Trial-MSL, learns to completely ignore the bias of the Trial-MSL dataset as measured by lesion-level detection performance. However, voxel-level segmentation performance suffers significantly. On the other hand, a model trained using the SCIN approach is able to adapt to the difference in labeling styles and exhibit good lesion-level detection and voxel-level segmentation performance when conditioned on the appropriate cohort (Row-4). Looking at SCIN-Pooling conditioned on Trial-MSL (Row-5), we see that SCIN is able to learn the Trial-MSL label bias quite effectively, and is able to ignore small lesions while maintaining voxel-level segmentation performance. This shows that SCIN is able to model complex non-linear labeling biases – its not just a matter of over or under segmentation.

6.4 Summary

In this chapter, we proposed SCIN, an approach that learns source-specific instance normalization parameters, effectively modeling the bias of each cohort present in an aggregated dataset. We show that the instance normalization parameters of a pre-trained SCIN model can be fine-tuned, allowing the model to learn the bias of an independent cohort with very little data. We demonstrate that the biases learned can be non-linear, resulting in complex differences in the segmentation outputs corresponding to each cohort. Most importantly, we show that proposed method makes it possible to train a high performance model on an aggregate dataset, avoiding the performance penalty observed with naive pooling. Overall, SCIN is simple to implement, and can potentially benefit any application that wishes to leverage a large aggregated dataset in the context of segmentation.

Chapter 7

Conclusion

This thesis presented an in-depth look at the process of engineering a deep learning system for the task of focal pathology segmentation and detection. We first illustrated the importance of building a well-engineered pipeline, pointing out a number of design decisions that can impact model evaluation and development. These decisions reinforce the need for a meticulous engineering process, involving an in-depth understanding of the problem at hand, including an understanding of the metrics by which success will be measured. Next, we turn our attention to the model development/configuration process, discussing a number of factors that, properly considered, enable the development of high performance models within common computational constraints. Through this process, we established a robust, well-trained UNet, outperforming other approaches on the same dataset. Having established a well-trained UNet, we propose Lesion Size Reweighting, a method to close the segmentation-detection performance gap, enabling a single model to achieve optimal performance on both tasks simultaneously. Finally, we turn our attention to the problem of aggregating datasets, proposing a new approach to model cohort-specific biases, enabling a single model to leverage a multi-cohort aggregated dataset.

Bibliography

[1] ARGANDA-CARRERAS, I., TURAGA, S. C., BERGER, D. R., CIRESAN, D. C., GIUSTI, A., GAMBARDELLA, L. M., SCHMIDHUBER, J., LAPTEV, D., DWIVEDI, S., BUHMANN, J. M., LIU, T., SEYEDHOSSEINI, M., TASDIZEN, T., KAMENTSKY, L., BURGET, R., UHER, V., TAN, X., SUN, C., PHAM, T. D., BAS, E., UZUNBAS, M. G., CARDONA, A., SCHINDELIN, J. E., AND SEUNG, H. S. Crowdsourcing the creation of image segmentation algorithms for connectomics. *Frontiers in Neuroanatomy 9* (2015).

[2] ASLANI, S., DAYAN, M., MURINO, V., AND SONA, D. Deep 2d encoder-decoder convolutional neural network for multiple sclerosis lesion segmentation in brain mri. In *International MICCAI Brainlesion Workshop* (2018), Springer, pp. 132–141.

[3] ASLANI, S., DAYAN, M., STORELLI, L., FILIPPI, M., MURINO, V., ROCCA, M. A., AND SONA, D. Multi-branch convolutional neural network for multiple sclerosis lesion segmentation. *NeuroImage 196* (2019), 1–15.

[4] ASLANI, S., MURINO, V., DAYAN, M., TAM, R., SONA, D., AND HAMARNEH, G. Scanner invariant multiple sclerosis lesion segmentation from mri. In *2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI)* (2020), IEEE, pp. 781–785.

[5] AVANTS, B. B., EPSTEIN, C. L., GROSSMAN, M., AND GEE, J. C. Symmetric diffeomorphic image registration with cross-correlation: Evaluating automated labeling of elderly and neurodegenerative brain. *Medical image analysis 12 1* (2008), 26–41.

[6] BADRINARAYANAN, V., KENDALL, A., AND CIPOLLA, R. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence 39* (2017), 2481–2495.

[7] BERGSTRA, J., AND BENGIO, Y. Random Search for Hyper-Parameter Optimization. *J. Mach. Learn. Res. 13* (2012), 281–305.

[8] BIBERACHER, V., SCHMIDT, P., KESHAVAN, A., BOUCARD, C. C., RIGHART, R., SÄMANN, P. G., PREIBISCH, C., FRÖBEL, D., ALY, L., HEMMER, B., ZIMMER, C., HENRY, R. G., AND MÜHLAU, M. Intra- and interscanner variability of magnetic resonance imaging based volumetry in multiple sclerosis. *NeuroImage 142* (2016), 188–197.

[9] BIRENBAUM, A., AND GREENSPAN, H. Longitudinal Multiple Sclerosis Lesion Segmentation Using Multi-view Convolutional Neural Networks. In *LABELS/DLMIA@MICCAI* (2016).

[10] BØ, H. K., SOLHEIM, O., JAKOLA, A. S., KVISTAD, K. A., REINERTSEN, I., AND BERNTSEN, E. M. Intra-rater variability in low-grade glioma segmentation. *Journal of Neuro-Oncology 131* (2016), 393–402.

[11] BROSCH, T., YOO, Y., TANG, L. Y., LI, D. K., TRABOULSEE, A., AND TAM, R. Deep convolutional encoder networks for multiple sclerosis lesion segmentation. In *International conference on medical image computing and computer-assisted intervention* (2015), Springer, pp. 3–11.

[12] BROWNE, P., CHANDRARATNA, D., ANGOOD, C., TREMLETT, H., BAKER, C., TAYLOR, B. V., AND THOMPSON, A. J. Atlas of Multiple Sclerosis 2013: A growing global problem with widespread inequity. *Neurology 83* (2014), 1022 – 1024.

[13] CARASS, A., ROY, S., JOG, A., CUZZOCREO, J. L., SWEENEY, E. M., GHERMAN, A., BUTTON, J., NGUYEN, J., PRADOS, F., SUDRE, C. H., CARDOSO, M. J.,

CAWLEY, N., CICCARELLI, O., WHEELER-KINGSHOTT, C. A. M., OURSELIN, S., CATANESE, L., DESHPANDE, H., MAUREL, P., COMMOWICK, O., BARILLOT, C., TOMAS-FERNANDEZ, X., AND IHEME, L. O. Longitudinal multiple sclerosis lesion segmentation: Resource and challenge. *NeuroImage 148* (2017), 77–102.

[14] CERVENANSKY, F., COMMOWICK, O., COTTON, F., DOJAT, M., AND EDAN, G. Multiple sclerosis new lesions segmentation challenge. *Zenodo* (2021).

[15] CHOTZOGLOU, E., AND KAINZ, B. Exploring the Relationship Between Segmentation Uncertainty, Segmentation Performance and Inter-observer Variability with Probabilistic Networks. In *LABELS/HAL-MICCAI/CuRIOUS@MICCAI* (2019).

[16] COMMOWICK, O., CERVENANSKY, F., AND AMÉLI, R. MSSEG Challenge Proceedings: Multiple Sclerosis Lesions Segmentation Challenge Using a Data Management and Processing Infrastructure. In *MICCAI 2016* (2016).

[17] COMMOWICK, O., ISTACE, A., KAIN, M., LAURENT, B., LERAY, F., SIMON, M., POP, S. C., GIRARD, P., AMELI, R., FERRÉ, J.-C., ET AL. Objective evaluation of multiple sclerosis lesion segmentation using a data management and processing infrastructure. *Scientific Reports* (2018).

[18] COMPSTON, A., AND COLES, A. J. Multiple sclerosis. *The Lancet 372* (2008), 1502–1517.

[19] CORONADO, I., GABR, R. E., AND NARAYANA, P. A. Deep learning segmentation of gadolinium-enhancing lesions in multiple sclerosis. *Multiple Sclerosis Journal 27*, 4 (2021), 519–527.

[20] CORTES, C., AND VAPNIK, V. N. Support-vector networks. *Machine Learning 20* (2004), 273–297.

[21] COX, D. R. The Regression Analysis of Binary Sequences. *Journal of the royal statistical society series b-methodological 20* (1958), 215–232.

[22] DEHGHANI, M., GOUWS, S., VINYALS, O., USZKOREIT, J., AND KAISER, L. Universal transformers. In *International Conference on Learning Representations* (2018).

[23] DU BUF, J. H., KARDAN, M., AND SPANN, M. Texture feature performance for image segmentation. *Pattern Recognition 23*, 3-4 (1990), 291–309.

[24] DUMOULIN, V., SHLENS, J., AND KUDLUR, M. A learned representation for artistic style. *International Conference on Learning Representations* (2017).

[25] EFRON, B. Size, power and false discovery rates. *Annals of Statistics 35* (2007), 1351–1377.

[26] FETTER, R., ZHENG, Z., ROBINSON, C. G., MILKIE, D., PERLMAN, E., PRICE, J., BOCK, D., KAZHDAN, M., KHAIRY, K., KARSH, B., TRAUTMAN, E., LI, P., FUNKE, J., ORDISH, C., SAALFELD, S., BUHMANN, J., AND ZHANG, C. CREMI dataset. `https://cremi.org`.

[27] FRANCIS, S. J. *Automatic Lesion Identification in MRI of Multiple Sclerosis Patients*. PhD thesis, McGill, 2004.

[28] FREEDMAN, L. P., COCKBURN, I. M., AND SIMCOE, T. S. The economics of reproducibility in preclinical research. *PLoS biology 13*, 6 (2015), e1002165.

[29] GABR, R. E., CORONADO, I., ROBINSON, M., SUJIT, S. J., DATTA, S., SUN, X., ALLEN, W. J., LUBLIN, F. D., WOLINSKY, J. S., AND NARAYANA, P. A. Brain and lesion segmentation in multiple sclerosis using fully convolutional neural networks: a large-scale study. *Multiple Sclerosis Journal 26*, 10 (2020), 1217–1226.

[30] GARCIA-LORENZO, D., PRIMA, S., ARNOLD, D. L., COLLINS, D. L., AND BARILLOT, C. Trimmed-likelihood estimation for focal lesions and tissue segmentation in multisequence mri for multiple sclerosis. *IEEE transactions on medical imaging 30*, 8 (2011), 1455–1467.

[31] GARCÍA-LORENZO, D., FRANCIS, S., NARAYANAN, S., ARNOLD, D., AND COLLINS, L. Review of automatic segmentation methods of multiple sclerosis white matter lesions on conventional magnetic resonance imaging. *Medical Image Analysis* (2013).

[32] GERIG, G., JOMIER, M., AND CHAKOS, M. Valmet: A new validation tool for assessing and improving 3d object segmentation. In *International conference on medical image computing and computer-assisted intervention* (2001), Springer, pp. 516–523.

[33] GESSERT, N., BENGS, M., KRÜGER, J., OPFER, R., OSTWALDT, A.-C., MANOGARAN, P., SCHIPPLING, S., AND SCHLAEFER, A. 4D Deep Learning for Multiple-Sclerosis Lesion Activity Segmentation. In *Medical Imaging with Deep Learning* (2020).

[34] GESSERT, N., KRÜGER, J., OPFER, R., OSTWALDT, A.-C., MANOGARAN, P., KITZLER, H. H., SCHIPPLING, S., AND SCHLAEFER, A. Multiple sclerosis lesion activity segmentation with attention-guided two-path cnns. *Computerized Medical Imaging and Graphics 84* (2020), 101772.

[35] HE, X., ZEMEL, R. S., AND CARREIRA-PERPINÁN, M. A. Multiscale conditional random fields for image labeling. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.* (2004), vol. 2, IEEE, pp. II–II.

[36] HELLER, N., DEAN, J., AND PAPANIKOLOPOULOS, N. Imperfect segmentation labels: How much do they matter? In *Intravascular Imaging and Computer Assisted Stenting and Large-Scale Annotation of Biomedical Data and Expert Label Synthesis*. Springer, 2018, pp. 112–120.

[37] IOFFE, S., AND SZEGEDY, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning* (2015), PMLR, pp. 448–456.

[38] ISENSEE, F., JAEGER, P. F., KOHL, S. A., PETERSEN, J., AND MAIER-HEIN, K. H. nnu-net: a self-configuring method for deep learning-based biomedical image segmentation. *Nature methods 18*, 2 (2021), 203–211.

[39] ISENSEE, F., KICKINGEREDER, P., WICK, W., BENDSZUS, M., AND MAIER-HEIN, K. H. No new-net. In *International MICCAI Brainlesion Workshop* (2018), Springer, pp. 234–244.

[40] ISENSEE, F., AND MAIER-HEIN, K. H. nnu-net for brain tumor segmentation. In *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries: 6th International Workshop, BrainLes 2020, Held in Conjunction with MICCAI 2020, Lima, Peru, October 4, 2020, Revised Selected Papers, Part II* (2021), vol. 12658, Springer Nature, p. 118.

[41] JÉGOU, S., DROZDZAL, M., VAZQUEZ, D., ROMERO, A., AND BENGIO, Y. The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops* (2017), pp. 11–19.

[42] JESSON, A., AND ARBEL, T. Hierarchical mrf and random forest segmentation of ms lesions and healthy tissues in brain mri. *proceedings of the 2015 longitudinal multiple sclerosis lesion segmentation challenge* (2015), 1–2.

[43] JOHNSON, A. E., AND HEBERT, M. Using spin images for efficient object recognition in cluttered 3d scenes. *IEEE Transactions on pattern analysis and machine intelligence 21*, 5 (1999), 433–449.

[44] JOSKOWICZ, L., COHEN, D., CAPLAN, N., AND SOSNA, J. Inter-observer variability of manual contour delineation of structures in ct. *European radiology 29*, 3 (2019), 1391–1399.

[45] JUNGO, A., MEIER, R., ERMIS, E., BLATTI-MORENO, M., HERRMANN, E., WIEST, R., AND REYES, M. On the effect of inter-observer variability for a reliable es-

timation of uncertainty of medical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention* (2018), Springer, pp. 682–690.

[46] KAMNITSAS, K., LEDIG, C., NEWCOMBE, V. F., SIMPSON, J. P., KANE, A. D., MENON, D. K., RUECKERT, D., AND GLOCKER, B. Efficient multi-scale 3d cnn with fully connected crf for accurate brain lesion segmentation. *Medical image analysis 36* (2017), 61–78.

[47] KANOPOULOS, N., VASANTHAVADA, N., AND BAKER, R. L. Design of an image edge detection filter using the sobel operator. *IEEE Journal of solid-state circuits 23*, 2 (1988), 358–367.

[48] KARANI, N., CHAITANYA, K., BAUMGARTNER, C., AND KONUKOGLU, E. A lifelong learning approach to brain mr segmentation across scanners and protocols. In *International Conference on Medical Image Computing and Computer-Assisted Intervention* (2018), Springer, pp. 476–484.

[49] KARPATHY, A. Maxpooling Example. `https://cs231n.github.io/convolutional-networks/`.

[50] KAZANCLI, E., PRCHKOVSKA, V., RODRIGUES, P., VILLOSLADA, P., AND IGUAL, L. Multiple sclerosis lesion segmentation using improved convolutional neural networks. In *VISIGRAPP (4: VISAPP)* (2018), pp. 260–269.

[51] KINGMA, D. P., AND BA, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[52] KOHL, S. A., ROMERA-PAREDES, B., MAIER-HEIN, K. H., REZENDE, D. J., ESLAMI, S., KOHLI, P., ZISSERMAN, A., AND RONNEBERGER, O. A hierarchical probabilistic u-net for modeling multi-scale ambiguities. *arXiv preprint arXiv:1905.13077* (2019).

[53] KRIZHEVSKY, A., SUTSKEVER, I., AND HINTON, G. E. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems 25* (2012), 1097–1105.

[54] LA ROSA, F., ABDULKADIR, A., FARTARIA, M. J., RAHMANZADEH, R., LU, P.-J., GALBUSERA, R., BARAKOVIC, M., THIRAN, J.-P., GRANZIERA, C., AND CUADRA, M. B. Multiple sclerosis cortical and wm lesion segmentation at 3t mri: a deep learning method based on flair and mp2rage. *NeuroImage: Clinical 27* (2020), 102335.

[55] LADICKỲ, L., RUSSELL, C., KOHLI, P., AND TORR, P. H. Associative hierarchical crfs for object class image segmentation. In *2009 IEEE 12th International Conference on Computer Vision* (2009), IEEE, pp. 739–746.

[56] LECUN, Y., BENGIO, Y., AND HINTON, G. Deep learning. *Nature 521*, 7553 (2015), 436–444.

[57] LEE, C., HUH, S., KETTER, T. A., AND UNSER, M. Unsupervised connectivity-based thresholding segmentation of midsagittal brain mr images. *Computers in biology and medicine 28*, 3 (1998), 309–338.

[58] LI, F.-F., KRISHNA, R., AND XU, D. Lecture 15: Detection and segmentation.

[59] LI, S. Z. *Markov Random Field Modeling in Image Analysis*. Springer Science & Business Media, 2009.

[60] LITJENS, G. J. S., KOOI, T., BEJNORDI, B. E., SETIO, A. A. A., CIOMPI, F., GHAFOORIAN, M., VAN DER LAAK, J., VAN GINNEKEN, B., AND SÁNCHEZ, C. I. A survey on deep learning in medical image analysis. *Medical image analysis 42* (2017), 60–88.

[61] LONG, J., SHELHAMER, E., AND DARRELL, T. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2015), pp. 3431–3440.

[62] LOWE, D. G. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision* (2004).

[63] MAAS, A. L., HANNUN, A. Y., NG, A. Y., ET AL. Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml* (2013), vol. 30, Citeseer, p. 3.

[64] MAIDAN, I., FREEDMAN, T., TZEMAH, R., GILADI, N., MIRELMAN, A., AND HAUSDORFF, J. Introducing a new definition of a near fall: intra-rater and inter-rater reliability. *Gait & posture 39*, 1 (2014), 645–647.

[65] MAŠKA, M., ULMAN, V., SVOBODA, D., MATULA, P., MATULA, P., EDERRA, C., URBIOLA, A., ESPAÑA, T., VENKATESAN, S., BALAK, D. M., ET AL. A benchmark for comparison of cell tracking algorithms. *Bioinformatics 30*, 11 (2014), 1609–1617.

[66] MAZZIOTTA, J. C., TOGA, A. W., EVANS, A., FOX, P., LANCASTER, J., ET AL. A probabilistic atlas of the human brain: theory and rationale for its development. *Neuroimage 2*, 2 (1995), 89–101.

[67] MCKINLEY, R., WEPFER, R., GUNDERSEN, T., WAGNER, F., CHAN, A., WIEST, R., AND REYES, M. Nabla-net: A deep dag-like convolutional architecture for biomedical image segmentation. In *International Workshop on Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries* (2016), Springer, pp. 119–128.

[68] MEHTA, R., CHRISTINCK, T., NAIR, T., BUSSY, A., PREMASIRI, S., COSTANTINO, M., CHAKRAVARTY, M., ARNOLD, D. L., GAL, Y., AND ARBEL, T. Propagating uncertainty across cascaded medical imaging tasks for improved deep learning inference. *IEEE Transactions on Medical Imaging* (2021).

[69] MILLER, D., GROSSMAN, R., REINGOLD, S., AND MCFARLAND, H. The role of magnetic resonance techniques in understanding and managing multiple sclerosis. *Brain: a journal of neurology 121*, 1 (1998), 3–24.

[70] MILLETARI, F., NAVAB, N., AND AHMADI, S.-A. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *2016 fourth international conference on 3D vision (3DV)* (2016), IEEE, pp. 565–571.

[71] NAIR, T., PRECUP, D., ARNOLD, D. L., AND ARBEL, T. Exploring uncertainty measures in deep networks for multiple sclerosis lesion detection and segmentation. *Medical image analysis 59* (2020), 101557.

[72] NAKKIRAN, P., KAPLUN, G., BANSAL, Y., YANG, T., BARAK, B., AND SUTSKEVER, I. Deep double descent: Where bigger models and more data hurt. In *International Conference on Learning Representations* (2019).

[73] NARANG, S., CHUNG, H. W., TAY, Y., FEDUS, W., FEVRY, T., MATENA, M., MALKAN, K., FIEDEL, N., SHAZEER, N., LAN, Z., ZHOU, Y., LI, W., DING, N., MARCUS, J., ROBERTS, A., AND RAFFEL, C. Do transformer modifications transfer across implementations and applications?, 2021.

[74] NICHYPORUK, B., CARDINELL, J., SZETO, J., MEHTA, R., TSAFTARIS, S., ARNOLD, D. L., AND ARBEL, T. Chort Bias Adaptation in Aggregated Datasets for Lesion Segmentation. In *Domain Adaptation and Representation Transfer, and Affordable Healthcare and AI for Resource Diverse Global Health*. Springer, 2021, pp. 101–111.

[75] NICHYPORUK, B., SZETO, J., ARNOLD, D., AND ARBEL, T. Optimizing Operating Points for High Performance Lesion Detection and Segmentation Using Lesion Size Reweighting. In *Medical Imaging with Deep Learning* (2021). Eprint arXiv:2107.12978 (Short Paper).

[76] NICHYPORUK, B., VASILEVSKI, K., HU, A., MYERS-COLET, C., CARDINELL, J., SZETO, J., FALET, J.-P., ZIMMERMANN, E., SCHROETER, J., ARNOLD, D. L., ET AL. Consensus learning with multi-rater labels for segmenting and detecting new lesions. *MSSEG-2 challenge proceedings: Multiple sclerosis new lesions segmentation challenge using a data management and processing infrastructure* (2021), 85. (Short Paper).

[77] PASZKE, A., GROSS, S., MASSA, F., LERER, A., BRADBURY, J., CHANAN, G., KILLEEN, T., LIN, Z., GIMELSHEIN, N., ANTIGA, L., DESMAISON, A., KOPF, A., YANG, E., DEVITO, Z., RAISON, M., TEJANI, A., CHILAMKURTHY, S., STEINER, B., FANG, L., BAI, J., AND CHINTALA, S. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019, pp. 8024–8035.

[78] RONNEBERGER, O., FISCHER, P., AND BROX, T. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention* (2015), Springer, pp. 234–241.

[79] ROTH, H. R., XU, Z., DÍEZ, C. T., JACOB, R. S., ZEMBER, J., MOLTÓ, J., LI, W., XU, S., TURKBEY, B., TURKBEY, E. B., YANG, D., HAROUNI, A. E., RIEKE, N., HU, S., ISENSEE, F., TANG, C., YU, Q., SÖLTER, J., ZHENG, T., LIAUCHUK, V., ZHOU, Z., MOLTZ, J. H., OLIVEIRA, B. A. S., XIA, Y., MAIER-HEIN, K. H., LI, Q., HUSCH, A. D., ZHANG, L., KOVALEV, V. A., KANG, L., HERING, A., VILAÇA, J. L., FLORES, M. G., XU, D., WOOD, B. J., AND LINGURARU, M. G. Rapid artificial intelligence solutions in a pandemic - the covid-19-20 lung ct lesion segmentation challenge. *Research Square* (2021).

[80] ROY, S., BUTMAN, J. A., REICH, D. S., CALABRESI, P. A., AND PHAM, D. L. Multiple sclerosis lesion segmentation from brain mri via fully convolutional neural networks. *ArXiv abs/1803.09172* (2018).

[81] RUMELHART, D. E., HINTON, G. E., AND WILLIAMS, R. J. Learning representations by back-propagating errors. *Nature 323*, 6088 (1986), 533–536.

[82] SAITO, T., AND REHMSMEIER, M. The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets. *PloS one 10*, 3 (2015), e0118432.

[83] SANTNER, J., UNGER, M., POCK, T., LEISTNER, C., SAFFARI, A., AND BISCHOF, H. Interactive texture segmentation using random forests and total variation. In *BMVC* (2009), Citeseer, pp. 1–12.

[84] SCHMIDHUBER, J. Deep learning in neural networks: An overview. *Neural networks 61* (2015), 85–117.

[85] SCULLEY, D., HOLT, G., GOLOVIN, D., DAVYDOV, E., PHILLIPS, T., EBNER, D., CHAUDHARY, V., YOUNG, M., CRESPO, J.-F., AND DENNISON, D. Hidden technical debt in machine learning systems. *Advances in Neural Information Processing Systems 28* (2015), 2503–2511.

[86] SCULLY, M., MAGNOTTA, V., GASPAROVIC, C., PELLIGRIMO, P., FEIS, D.-L., AND BOCKHOLT, H. J. 3d segmentation in the clinic: A grand challenge ii at miccai 2008 - ms lesion segmentation. *Grand Challenge Workshop: Multiple Sclerosis Lesion Segmentation Challenge* (2008).

[87] SEPAHVAND, N. M., HASSNER, T., ARNOLD, D. L., AND ARBEL, T. CNN Prediction of Future Disease Activity for Multiple Sclerosis Patients from Baseline MRI and Lesion Labels. In *International MICCAI Brainlesion Workshop* (2018), Springer, pp. 57–69.

[88] SHIROKIKH, B., SHEVTSOV, A., KURMUKOV, A., DALECHINA, A., KRIVOV, E., KOSTJUCHENKO, V., GOLANOV, A., AND BELYAEV, M. Universal loss reweighting to balance lesion size inequality in 3d medical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention* (2020), Springer, pp. 523–532.

[89] SHWARTZMAN, O., GAZIT, H., SHELEF, I., AND RIKLIN-RAVIV, T. The worrisome impact of an inter-rater bias on neural network training, 2020.

[90] SIMPSON, A. L., ANTONELLI, M., BAKAS, S., BILELLO, M., FARAHANI, K., VAN GINNEKEN, B., KOPP-SCHNEIDER, A., LANDMAN, B. A., LITJENS, G., MENZE, B.,

RONNEBERGER, O., SUMMERS, R. M., BILIC, P., CHRIST, P. F., DO, R. K. G., GOLLUB, M., GOLIA-PERNICKA, J., HECKERS, S. H., JARNAGIN, W. R., MCHUGO, M. K., NAPEL, S., VORONTSOV, E., MAIER-HEIN, L., AND CARDOSO, M. J. A large annotated medical image dataset for the development and evaluation of segmentation algorithms, 2019.

[91] SINGH, D., AND SINGH, B. Investigating the impact of data normalization on classification performance. *Applied Soft Computing 97* (2020), 105524.

[92] SRIVASTAVA, N., HINTON, G., KRIZHEVSKY, A., SUTSKEVER, I., AND SALAKHUTDINOV, R. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research 15*, 1 (2014), 1929–1958.

[93] SUBBANNA, N., PRECUP, D., ARNOLD, D., AND ARBEL, T. IMaGe: iterative multilevel probabilistic graphical model for detection and segmentation of multiple sclerosis lesions in brain mri. In *International Conference on Information Processing in Medical Imaging* (2015), Springer, pp. 514–526.

[94] SUN, C., SHRIVASTAVA, A., SINGH, S., AND GUPTA, A. Revisiting unreasonable effectiveness of data in deep learning era. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)* (2017), pp. 843–852.

[95] TORIWAKI, J., AND YONEKURA, T. Euler number and connectivity indexes of a three dimensional digital picture. *Forma* (2002).

[96] ULYANOV, D., VEDALDI, A., AND LEMPITSKY, V. Instance normalization: The missing ingredient for fast stylization, 2017.

[97] VAIDYA, S., CHUNDURU, A., MUTHUGANAPATHY, R., AND KRISHNAMURTHI, G. Longitudinal multiple sclerosis lesion segmentation using 3D convolutional neural networks. *Proceedings of the 2015 longitudinal multiple sclerosis lesion segmentation challenge* (2015), 1–2.

[98] VALVERDE, S., CABEZAS, M., ROURA, E., GONZÁLEZ-VILLA, S., SALVI, J., OLIVER, A., AND LLADÓ, X. Multiple sclerosis lesion detection and segmentation using a convolutional neural network of 3d patches. *MSSEG Challenge Proceedings: Multiple Sclerosis Lesions Segmentation Challenge Using a Data Management and Processing Infrastructure 75* (2016).

[99] VALVERDE, S., SALEM, M., CABEZAS, M., PARETO, D., VILANOVA, J. C., RAMIÓ-TORRENTÀ, L., ROVIRA, À., SALVI, J., OLIVER, A., AND LLADÓ, X. One-shot domain adaptation in multiple sclerosis lesion segmentation using convolutional neural networks. *NeuroImage: Clinical 21* (2019), 101638.

[100] VAN OPBROEK, A., IKRAM, M., VERNOOIJ, M., AND DE BRUIJNE, M. Transfer learning improves supervised image segmentation across imaging protocols. *IEEE transactions on medical imaging 34*, 5 (2014), 1018–1030.

[101] VASWANI, A., SHAZEER, N., PARMAR, N., USZKOREIT, J., JONES, L., GOMEZ, A. N., KAISER, Ł., AND POLOSUKHIN, I. Attention is all you need. In *Advances in neural information processing systems* (2017), pp. 5998–6008.

[102] VÉSTIAS, M. P. A survey of convolutional neural networks on edge with reconfigurable computing. *Algorithms 12*, 8 (2019), 154.

[103] VINCENT, O., GROS, C., AND COHEN-ADAD, J. Impact of individual rater style on deep learning uncertainty in medical imaging segmentation, 2021.

[104] VOULODIMOS, A., DOULAMIS, N., DOULAMIS, A., AND PROTOPAPADAKIS, E. Deep learning for computer vision: A brief review. *Computational intelligence and neuroscience 2018* (2018).

[105] WACK, D. S., DWYER, M. G., BERGSLAND, N., DI PERRI, C., RANZA, L., HUSSEIN, S., RAMASAMY, D., POLONI, G., AND ZIVADINOV, R. Improved assessment of multiple sclerosis lesion segmentation agreement via detection and outline error estimates. *BMC medical imaging 12*, 1 (2012), 1–10.

[106] WANG, L., LEE, C.-Y., TU, Z., AND LAZEBNIK, S. Training deeper convolutional networks with deep supervision, 2015.

[107] WANG, X.-Y., WANG, T., AND BU, J. Color image segmentation using pixel wise support vector machine classification. *Pattern Recognition 44*, 4 (2011), 777–787.

[108] WARFIELD, S. K., ZOU, K. H., AND WELLS, W. M. Simultaneous truth and performance level estimation (staple): an algorithm for the validation of image segmentation. *IEEE transactions on medical imaging 23*, 7 (2004), 903–921.

[109] WU, M., ROSANO, C., LOPEZ-GARCIA, P., CARTER, C. S., AND AIZENSTEIN, H. J. Optimum template selection for atlas-based segmentation. *NeuroImage 34*, 4 (2007), 1612–1618.

[110] ZENG, C., GU, L., LIU, Z., AND ZHAO, S. Review of Deep Learning Approaches for the Segmentation of Multiple Sclerosis Lesions on Brain MRI. *Frontiers in Neuroinformatics 14* (2020), 55.

[111] ZHANG, C., SONG, Y., LIU, S., LILL, S., WANG, C., TANG, Z., YOU, Y., GAO, Y., KLISTORNER, A., BARNETT, M., ET AL. Ms-gan: Gan-based semantic segmentation of multiple sclerosis lesions in brain magnetic resonance imaging. In *2018 Digital Image Computing: Techniques and Applications (DICTA)* (2018), IEEE, pp. 1–8.

[112] ZHANG, H., VALCARCEL, A. M., BAKSHI, R., CHU, R., BAGNATO, F., SHINOHARA, R. T., HETT, K., AND OGUZ, I. Multiple sclerosis lesion segmentation with tiramisu and 2.5 d stacked slices. In *International Conference on Medical Image Computing and Computer-Assisted Intervention* (2019), Springer, pp. 338–346.

[113] ZHANG, H., ZHANG, J., ZHANG, Q., KIM, J., ZHANG, S., GAUTHIER, S. A., SPINCEMAILLE, P., NGUYEN, T. D., SABUNCU, M., AND WANG, Y. Rsanet: Recurrent slice-wise attention network for multiple sclerosis lesion segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention* (2019), Springer, pp. 411–419.