A 3D U-net for Segmentation of Subcortical Structures in MR Images of 12 and 24 month-old infants

Nicholas Tapp-Hughes
Department of Computer Science
University of North Carolina at Chapel Hill

October 2020

Abstract

Recent research has determined links between the development of some neurodevelopmental disorders, including Autism Spectrum Disorder (ASD), and the morphometry of subcortical brain structures in children aged 12 and 24 months old, which is usually obtained via segmentation of a Magnetic Resonance (MR) image. Deep-learning based methods, such as the u-net, have emerged as the fastest way to compute segmentations with admissible accuracy. I have constructed a segmentation pipeline with a processing step and neural network step using a 3D u-net model trained on processed data from 830 automatically segmented images and 27 manually segmented images. The model is trained with a loss function that has structure volume and surface accuracy considerations. The segmentation pipeline is available as an open source software on GitHub. The model is validated on a set of 5 manually segmented images and obtains moderately good results compared to the state-of-the-art.

Approved By:

Thesis Advisor: Martin A. Styner: _____

Secondary Reader: Stephen M. Pizer: _____

Contents

List of Figures

List of Tables

1 Background

1.1 Subcortical segmentation in clinical and research settings

Semantic segmentation, the task of labeling each element (pixel or voxel) of an image as one of several categories, is essential in many domains of computer science research and practical application. Biomedical image segmentation, specifically, is a common tool for clinical diagnosis as well as research of various diseases and disorders, ranging from melanoma [1] to COVID-19 [2]. Segmentation of subcortical brain structures from Magnetic Resonance (MR) images is important for the study or diagnosis of common Neurodevelopmental Disorders (NDDs), such as Autism Spectrum Disorder (ASD) [3], multiple-sclerosis [4], schizophrenia [5], Parkinson's disease [6], Alzheimer's [7], and major depression [8]. It has been found that the amygdala and hippocampus are enlarged in children with ASD [3], so determining the volume of these structures from subcortical segmentations of MR images may be useful as a predictive factor for diagnosis or for longitudinal studies.

In many cases, diagnoses of NDDs are determined after two years of age, usually via behavioral or cognitive patterns. Thus, segmentation of MR images taken at or before 24 months of age are paramount to obtaining earlier diagnosis of NDDs so that interventions may be taken earlier, which greatly improves the prognoses of affected individuals [10]. According to the Centers for Disease Control (CDC), ASD is not diagnosed reliably until at least 2 years of age, and diagnoses may come much later than that. ASD diagnosis is based on behavioral patterns including (1) avoiding eye contact, (2) having little interest in other children or caretakers, (3) limited display of lan-

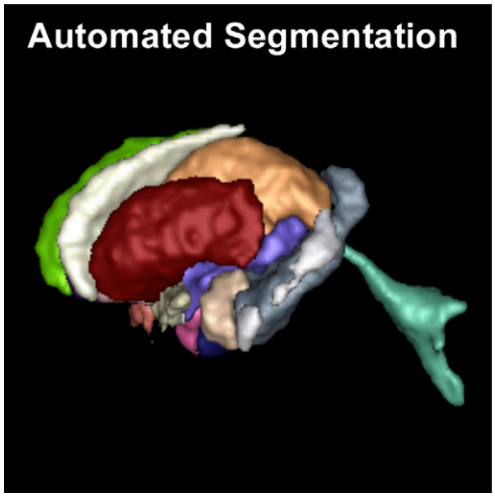

Figure 1: A subcortical segmentation obtained by the AutoSeg pipeline. Image source: [9]

guage, or (4) getting upset by minor changes in routine [11]. There is already some success in diagnosing ASD at 12 months of age by segmentation-based methods. Hazlett et al. used deep learning-based segmentation methods to predict ASD diagnosis at 24 months with 80% positive predictive value, from MRI taken at 6 and 12 months [12]. While this is a good result, there is still much room for improvement and investigation of techniques.

1.2 CNNs for segmentation of infant MR images

With the importance of segmentations established, I turn to the question of how segmentations of biomedical images are obtained. There is of course the method of manual segmentation, where a human expert delineates the boundaries of an anatomical structure, but for 3D images and in clinical settings this method impractically time-consuming. Furthermore, it has been shown that there is significant inconsistency between the manual segmentations of different physicians, as well as between different segmentations from the same physician, which can be a hindrance to potential applications [13]. While manual segmentation remains a popular method in clinical settings as well as a standard benchmark for automatic segmentation accuracy, computed segmentations are often generated much faster and have been shown to achieve higher accuracy in some cases.

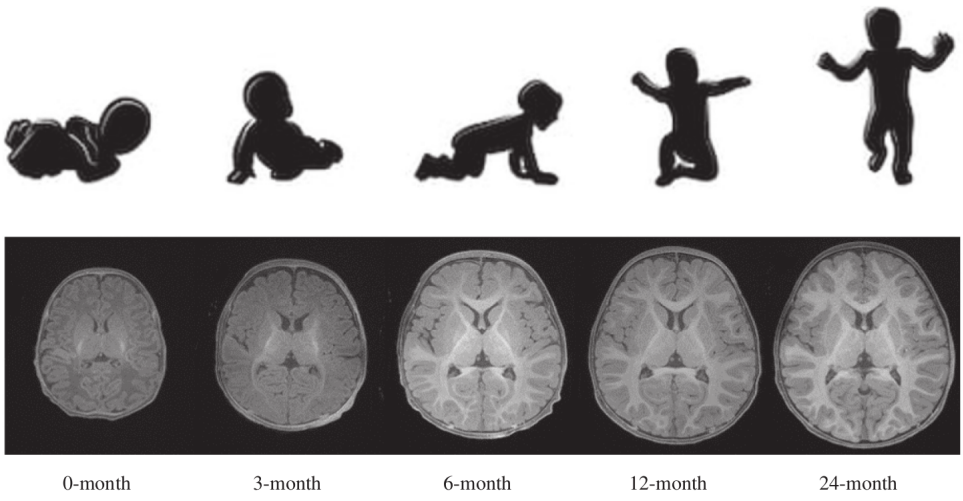

Figure 2: T1 MR images of a neurotypical infant. Image source: [14]

Over the last few decades, several non-deep learning computational methods have been proposed to segment infant MR brain images. These methods can be roughly categorized into parametric [15], classification [16], multi-atlas fusion [17], and deformable models [18]. All of these methods achieve accurate results but struggle in some use cases, such as segmentation of myelinated vs. unmyelinated white matter or tissue segmentation at the isointense stage (6 months), where there is low contrast between white and gray matter and low signal-to-noise ratio [14]. Of these categories, multi-atlas-based based methods such as AutoSeg [9] have emerged as the method achieving best scores on evaluation metrics. However, multi-atlas fusion techniques have a high computational burden—a single multi-atlas segmentation computed on a standard CPU can take several days to complete [19], rendering this method inconvenient and in some settings impractical. Recently, there has been a surge of interest in deep learning-based methods, which involve multi-layer neural networks, for segmentation of MR images.

Neural networks were studied as early as 1943 [20] and gained popularity as a research subject in the 1980's. One of the most popular types of neural network is the Convolutional Neural Network (CNN), which uses the image-processing convolution operation to compute outputs. The idea of a CNN was first introduced in 1980 [21], but research on the subject did not take flight until recently, after the first CNN was trained using a GPU in 2006 [22]. Since 2015, CNNs have proven to be a highly effective tool in image analysis problems and have outperformed the state-of-the-art in many applications, including segmentation of infant MR images and other biomedical images [23, 24, 25]. In particular, CNNs can be exceptionally accurate where other methods struggle, such as in the task of brain tissue segmentation of isointense 6 month MR images [14]. Another benefit is the broad potential for research and application of CNNs due to the highly variable and versatile structures of CNN models. Furthermore, trained CNNs can produce segmentations in a matter of seconds, several orders of magnitude more quickly than multi-atlas fusion and other methods, making them ideal for use with general software programs or embedded systems. There is much potential for CNN-based technology in medicine, as well as in many other domains.

1.3 An overview of neural networks

A neural network is a nonlinear, highly complex, parameterized mapping from some input domain to some output space. Neural networks consist

of multiple composed layers, which are themselves parameterized mappings. Computationally intensive layers usually involve some kind of linear component that depends on parameters, such as a linear transformation and translation (fully connected layer) or a convolution operation (convolutional layer), followed by a nonlinear component (e.g. ReLU, hyperbolic tangent, softmax, or the sigmoid function) called the activation function. Non-computationally intensive layers, such as pooling layers or dropout layers, are added to simplify the structure of the network or enhance training. The outputs of layers can be thought of as modeling activations of neurons, which then influence the activation of other neurons in a network. The depth of a neural network is the number of layers it has (hence "deep learning"). The first neural network models were composed of layers arranged in a serial feed-forward fashion, with the output of each layer taken as the input of the next layer; however this is not required. Modern neural networks take the output of some layers as input to multiple subsequent layers, allowing for aggregation of information. Networks which use the output of each layer as input to all subsequent layers are called dense networks. Networks which contain convolutional layers are called Convolutional Neural Networks (CNNs). Factors such as the depth of the network, the size of the layers, the choice of layer type, and the structure of the network are called hyperparameters. In addition, there are many other types of neural networks and layers not mentioned here.

The complexity and large number of parameters of neural networks allows for close approximation of desired mappings, if the parameters of the network are well-adjusted [26]. For example, the mapping of MR brain images to their segmentation images by expert physicians can be well approximated by a CNN. If the MR image is a 3D tensor with size $(96 \times 112 \times 96)$ and there are c categories (classes) in the segmentation, then the mapping g that takes MR images to their segmentations may take the representation:

$$g : \mathbb{R}^{96\times112\times96} \to \mathbb{R}^{96\times112\times96\times c}$$

where the output segmentation is represented as a one-hot encoding (each voxel is a vector of length c, where there is a 1 at the index of the label number, and zeros elsewhere). It should also be noted that we usually only care about approximating the mapping on a subset of the domain, which is characterized by a distribution over the input space. In the last example, we only want to segment MR images, not all of $\mathbb{R}^{96\times112\times96}$.

But how do we optimize the parameters of a neural network such that

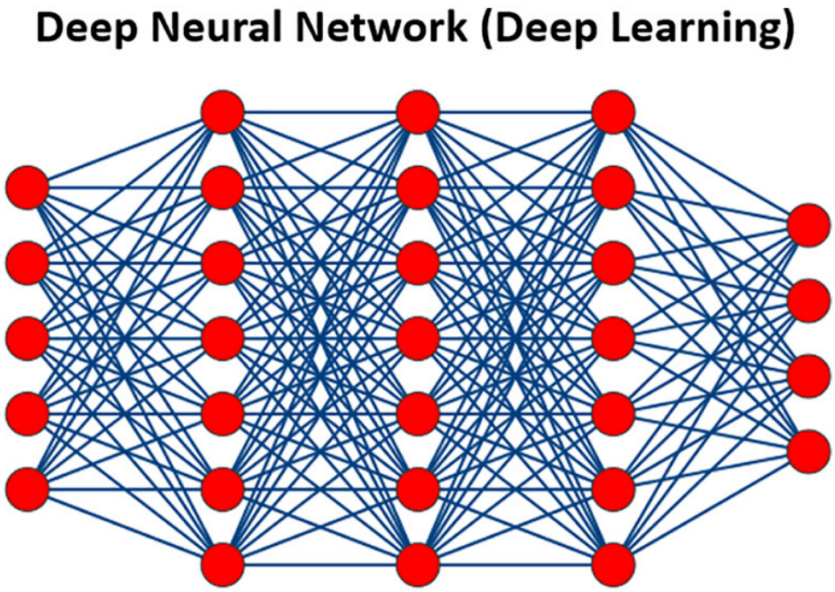

Figure 3: The architecture of a feed-forward neural network with fully connected layers. Image source: [14]

close approximation of a target mapping is achieved? This is accomplished through training of the neural network, which requires several things:

1. A large set of training data and their ground truth outputs by the target mapping

2. A loss function, which measures the disagreement between an output generated by the neural network and the ground truth and is differentiable with respect to the network parameters

3. The use of a backpropagation algorithm [27], to minimize the loss over the network parameters

The training dataset gives the neural network a set of points by which to approximate the target mapping. It is important that the training dataset is representative of the distribution of use-case inputs and comprehensive in that it contains as many of the features of the distribution of use-case inputs as possible, so that the network is robust to all plausible variations in input. A lack of training data can cause overfitting, which occurs when a neural

network approximates the target mapping well on the training set but not on other data.

The loss function, sometimes referred to as the objective function, is also an important aspect of training. It is a non-negative, scalar-valued function of the set neural network outputs on the training dataset and the corresponding ground truth labels, which measures the disagreement between the two. There are many choices for the loss, such as the L2 norm (Euclidean distance), cross-categorical entropy, or Huber loss [28]. Each choice of loss function has advantages and disadvantages. Often, loss functions are constructed to meet the exact needs of a specific application. The loss function must be differentiable with respect to the network parameters, so that it can be minimized over the network parameters. This implies that the neural network itself must be differentiable with respect to its parameters as well.

The general ideas of optimization are given here. Suppose that a network has parameters β, and the loss is $L(f(x;\beta), g(x))$, where f is the neural network, g is the target mapping, and x is the training dataset. Then, to find a local minimum of L, we can take an initial guess β^0 and proceed by iterative gradient descent in steps:

$$\beta^{i+1} = \beta^i - \gamma \nabla_\beta \mathcal{L}(f(x;\beta^i), g(x))$$

where γ is the step size. In practice, the gradient $\nabla_\beta \mathcal{L}$ is impractical to compute when x is the entire training dataset. Instead, we iterate on batches of data points $x_i, \ldots, x_{i+m-1}$, where m is the batch size. The method of choosing these batches randomly is called stochastic gradient descent [29] and is commonly used in neural network training. In the interest of uniformity, we often have that the batch size divides the total number of data points in the training set. When we have iterated over the entire dataset, we say that we have trained for an epoch. In each iteration, the gradient $\nabla_\beta \mathcal{L}$ is computed by a backpropagation algorithm, in which gradients are computed for each layer recursively, starting from the last layers of a network and working backwards. There are several implementations of optimization algorithms used commonly in deep learning, including the popular Adam [30]. We must also have a method of choosing β^0, our initial guess for the parameters. This is consequential, since we are finding a local minimum of $\mathcal{L}$ near β_0. Usually, parameters are initialized according to some distribution, such as all 0s, all 1s, the uniform distribution on $[-1, 1]$, the normal distribution with $\mu = 0, \sigma^2 = 1$, the He normal distribution [31], or the Glorot

normal distribution [32].

Once you have trained your network, there is a question of how to evaluate it. Evaluation metrics measure the disagreement between network-generated output and ground truth but are not required to be differentiable, like the loss function. For 3D segmentation tasks, some common evaluation metrics are categorical accuracy, Intersection over Union (IoU), dsc, Mean Absolute surface Distance (MAD), and volumetric overlap error [33]. Each evaluation metric measures a different aspect of the segmentation, and there are peculiarities to each of them. Usually, segmentations need to be evaluated by visual inspection as well.

CNNs are most often used on image data, since convolution operations are well suited for image processing. In a convolutional layer, several kernels (also called filters) are convolved over the input image to create the output. In addition, there is a bias associated with each kernel, which is added to each element of the output of that convolution operation, and a nonlinear activation function such as ReLU is applied. Kernels typically have size 1^2, 3^2, or 5^2 for 2D convolutions, or 1^3, 3^3, or 5^3 for 3D convolutions. The trainable parameters of the convolutional layer are the elements of each kernel and the biases. Convolutions may also be dilated [34] or strided, for increased receptive field of output elements or network simplicity. Convolutions at the border of the image may use mirrored or zero padding or not be computed at all. The output associated with each kernel of a convolutional layer is called an activation map. Convolutional layers are often composed with pooling layers, which reduce the size of activation maps by partitioning the image into small patches and replacing each patch with its average or maximum value. Convolutional layers are capable of extracting semantic spacial information from an image. An initial convolution on an image may produce activation maps corresponding to the presence of edges in the image. Another convolution on the resulting activation map may allow detection of more complex features of the original image, such as small shapes or patterns. Thus, multiple convolutional layers and pooling layers, when composed together, are capable of extracting high-level semantic information from an image. This allows CNNs to be highly effective at image classification, image segmentation, and image-to-image tasks.

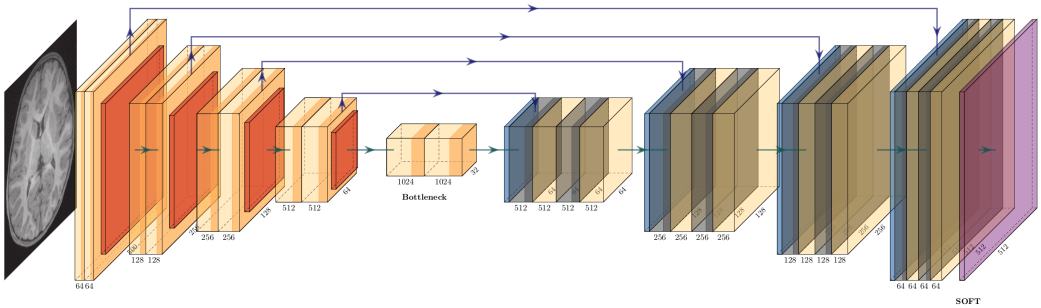

Figure 4: A 2D u-net architecture with 5 levels.

1.4 An overview of the u-net

One famous and popular CNN architecture is the u-net [24], which is often used for segmentation of biomedical images. The u-net is composed of convolutional layers, pooling layers, and transposed convolutional layers, which up-sample images by an inverse convolution operation. The u-net has two major parts, which form halves of the network: the encoder and the decoder. The encoder is composed of convolutional and pooling layers, down-sampling the image resolution several times in "levels," eventually encoding the image into a tensor of high-level semantic features. The decoder up-samples those features into an output segmentation image, where the layers at each level of the decoder take the output of the preceding level of the decoder, as well as the output of the corresponding level of the encoder, as input. These output-input connections between the encoder and decoder layers on the same level are known as skip connections. They allow layers in the decoder to have access to the abstract, "coarse" features passed up from the preceding level, as well as the simple, "fine" features passed through the skip connection. Finally, the last convolutional layer uses c kernels, where c is the number of segmentation categories, and applies a softmax activation function across the c channels, outputting a probability map where each element corresponds to the probability of the pixel or voxel being of a category. The u-net aggregates coarse and fine features of the input image during computation of the output, allowing for accurate segmentations.

In this project, I utilize a 3D u-net to segment MR images of 12 and 24 month subjects into seven categories. The categories correspond to background and 6 subcortical brain structures: the amygdala, the caudate nucleus, the hippocampus, the globus pallidus, the putamen, and the thalamus.

2 Methods

2.1 Training data

The MR image data used in this project comes from the Infant Brain Imaging Study (IBIS) [35]. Participants enrolled in the study were infants who may have had high or low familial risk for ASD, depending on whether they are the younger sibling of someone with ASD. MR scans were taken at 4 sites: the University of North Carolina, University of Washington, Washington University in St. Louis, and Children's Hospital of Philadelphia. A description of the MR scan acquisition and quality control considerations is given in [36]. A series of MR scans were taken of the participants, as they aged from 6 to 24 months old.

Each multi-modality MR scan produced a T1 and a T2 image. 862 scans, from 12 and 24 month subjects, were used for this project. Scans underwent processing, including geometric distortion correction, transformation to steroactic space, and skull-stripping, as described in the supplemental section of [37]. 830 of the scans were segmented automatically via the AutoSeg pipeline [9], and 32 scans were segmented manually by researchers in University of North Carolina's Autism Research Center [38]. The 830 automatically segmented images formed a pre-training dataset for the network, 27 of the 32 manually segmented images formed a training dataset for the network, and the remaining 5 manually segmented images were used as a validation set. Optimization of the network was carried out on the pretraining set and then on the training set, to take advantage of transfer learning [39].

2.2 Data pre-processing

Before the images were ready to be input to my network, they required further processing, which is summarized in Figure 5. The histograms of the T1 and T2 images were matched to average histograms

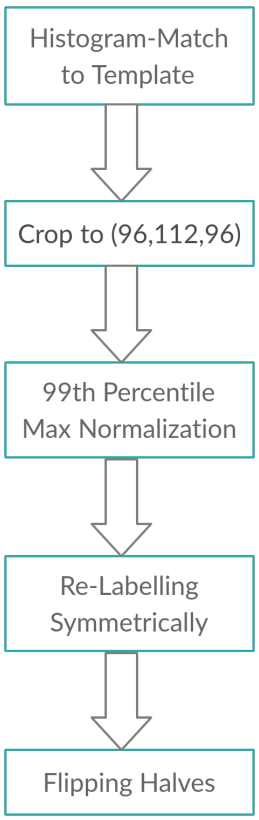

Figure 5: The image processing pipeline

of skull-stripped 12-month MR images from IBIS [40]. Then the images were cropped to their central tensor of size ($96 \times 112 \times 96$), which eliminated most of the background and contained all the subcortical structures. Next, the voxel intensity values of each image were divided by the 99^{th} percentile (when sorted in ascending order) value of that image. The effect of this is to scale the intensity values roughly to the interval $[0, 1]$, while maintaining robustness to a few artificially high values. Then, since right and left symmetrical brain structures were labeled differently in the ground truth segmentations, structures were re-labeled so that right and left structures had the same label. Ground truth segmentations were also converted to one-hot format. Finally, the pre-training and training datasets were doubled by splitting the images in half along the median (sagittal) plane and then reflecting (flipping) each half across the same plane, so that the new images were perfectly symmetric. These processing steps helped to homogenize the dataset without loss of features and to synthetically increase the number of data points.

Name	**Label**	**VWF**
Background	0	1.04
Amygdala	1	453
Caudate Nucleus	2	151
Hippocampus	3	196
Globus Pallidus	4	466
Putamen	5	119
Thalamus	6	76.3

Table 1: Segmented subcortical structures and their relative VWFs. Higher VWF implies smaller structure volume.

In addition, calculating the average volumes of each structure was of interest for training. There are 7 categories I am interested in segmenting. Their names, numeric labels, and Volumetric Weighting Factor (VWF) are given in Table 1. The VWF of a structure is the inverse of the average relative volume of that structure in the pretraining and training dataset, given by

$$\text{VWF}_i = \frac{|D| \cdot \text{vol}(I)}{\Sigma_{I \in D} \, \Sigma_{j \in I} \, \delta_{ij}}$$

where i is the label number, I is the ground truth segmentation tensor, D is the set of labels in the pretraining and the training dataset, $\mathrm{vol}(I)$ is the number of voxels in each of the true segmentation tensors, and δ_{ij} is the Kronecker delta. Structures with smaller volume have higher VWF. The importance of the VWFs during training is discussed in the next section.

2.3 Network architecture and loss

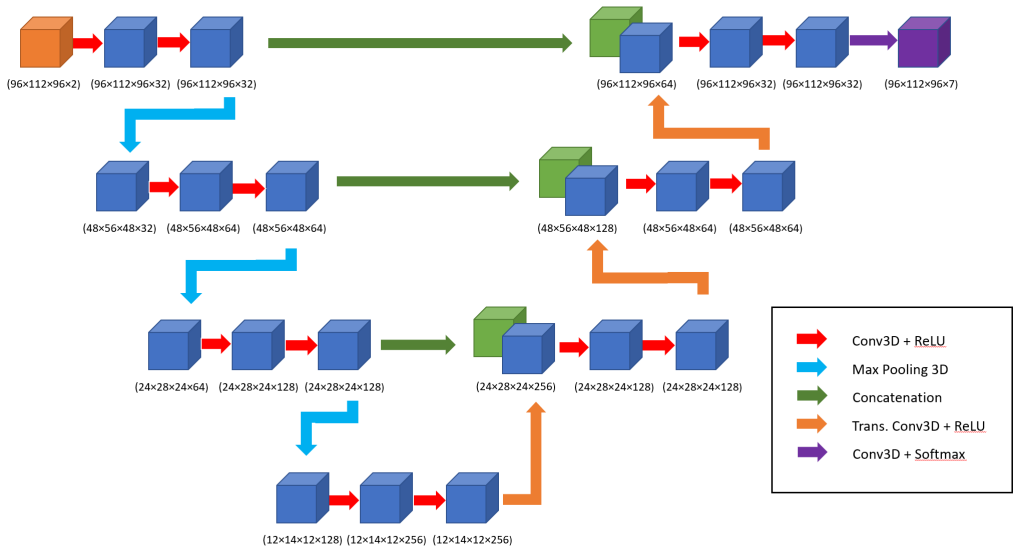

Figure 6: The 3D u-net model used for obtaining subcortical segmentations. The sizes below each block are written in channels-last format. T1 and T2 images are processed and concatenated along the channel axis to create the input (orange). There are 7 channels in the output (purple) corresponding to the 7 semantic categories. Tensors in lower levels have lower resolution but more channels. In total, the network has 4,789,447 parameters.

For the task of segmenting subcortical brain structures, a 3D u-net neural network architecture was chosen. The network structure is shown in Figure 6. There are 4 levels in the u-net architecture, the same as Çiçek et al.'s original 3D u-net [41]. In the first level, convolutional layers had 32 kernels, while the second level had 64 kernels, the third had 128 kernels, and the bottom level had 256 kernels, with the exception of the last convolutional layer,

which had 7 kernels, corresponding to the 7 segmentation categories. All the kernels in convolutional layers had shape $(3 \times 3 \times 3)$. Max pooling layers took maximums over $(2 \times 2 \times 2)$ patches, and transposed convolutional layers used kernels with size $(2 \times 2 \times 2)$ and stride 2 so that up-sampling returned tensors to the same shape they were before pooling. Since input tensors are down-sampled by these pooling layers 3 times as they go through the network, it is most convenient if each dimension of the tensor is divisible by 8, which was a motivating factor in choosing the cropping size $(96 \times 112 \times 96)$. All kernel parameters were initialized according to a Glorot normal distribution [32], and the biases were initialized to 0. ReLU activation was used for each convolutional and transposed convolutional layer, except for the last convolutional layer, which used softmax activation. A dropout layer (not shown in Figure 6) was included in the lowest level of the network with dropout rate = 0.5, to enhance training.

Categorical cross-entropy was used for the loss function, with special weighting for voxels according to the VWF of their true category and whether or not they are on the boundary between two categories in the ground truth segmentation. For each 3D vector-valued (one-hot) ground truth segmentation tensor T, we construct a 3D tensor of VWFs W, where each element of W is the computed VWF of the corresponding label in T. We also construct a 3D boundary tensor B, where each element of B is a 1 if the corresponding element of T is located on a boundary (i.e. one of its neighbors is a different label) or a 0 otherwise. Then for each 3D vector-valued network output Y, the overall loss is calculated:

$$\mathcal{L}(Y,T) = \sum_k \left[\eta \mathcal{C}(Y,T) \odot W + (1-\eta)\mathcal{C}(Y,T) + \omega \mathcal{C}(Y,T) \odot B\right]_k$$

where η is a scalar $\in [0,1]$ which determines the influence of the VWFs, ω is a scalar > 0 which multiplies the loss for boundary voxels, $\odot$ denotes element-wise tensor multiplication, k is a voxel index, and $\mathcal{C}(Y,T)$ is the 3D tensor-valued categorical cross-entropy, each element of which is computed as

$$\mathcal{C}(Y,T)_k = -\sum_{i=1}^{c} T_{k,i} \log(Y_{k,i})$$

where c is number of segmentation categories, and k is the voxel index (such that T_k and Y_k are vectors of length c). The values $\eta = 0.01, 0.005, 0.0025$

and $\omega = 5$ were used for training. Since we are segmenting 7 categories, we have $c = 7$.

This loss was developed with the goal that segmentation volumes and surfaces would be as accurate as possible. In particular, the amygdala and hippocampus volumes are particularly relevant to diagnosis of ASD, but these structures are small and do not impact the loss very much, so the volumetric weighting scheme was devised to ensure accurate segmentation of small structures. Additionally, preliminary testing showed that structures were often over-segmented or under-segmented by 1 or 2 voxels around the boundary surface of the structure. So, boundary tensors B were computed and used during training to weight the loss of boundary voxels and encourage accurate surface shape in the output segmentation. The VWF tensors W and the boundary tensors B were computed prior to training and used during computation of loss in the optimization loop.

During training and inference, T1 and T2 input tensors were concatenated along the channel axis and taken as input to the network, according to an early-fusion strategy [14]. Optimization was carried out in iterations on batches of size 2 using stochastic gradient descent, and TensorFlow's `tf.GradientTape` class was used to calculate gradients by backpropagation. On the pretraining set, the model was optimized for 5 epochs with a learning rate (step size) of 10^{-4}, another 8 epochs with a learning rate of 10^{-6}, and another 7 epochs with a learning rate of 10^{-7}. On the training set, the model was optimized for 10 epochs at 10^{-6} and 10 epochs at 10^{-7} The loss during optimization on the pre-training set is shown in Figure 7.

2.4 Post-processing steps

The output of the neural network is a probability map with seven channels, where the values of each channel in a given voxel correspond to categorical probabilities and sum to one. Naively, we may take argmax over the channel axis to obtain our segmentation. However, there are many voxels in the probability map where there is not an outstanding maximum over the channel axis. Instead, we take argmax on voxels where the max is greater than 0.5. Otherwise, we label that voxel as background. In addition, we add a post-processing step that iterates over all voxels in the segmentation, re-labeling a voxel if 22 of its 26 neighbors share a common label different from the original voxel's label. This smooths structure surfaces and removes isolated voxels.

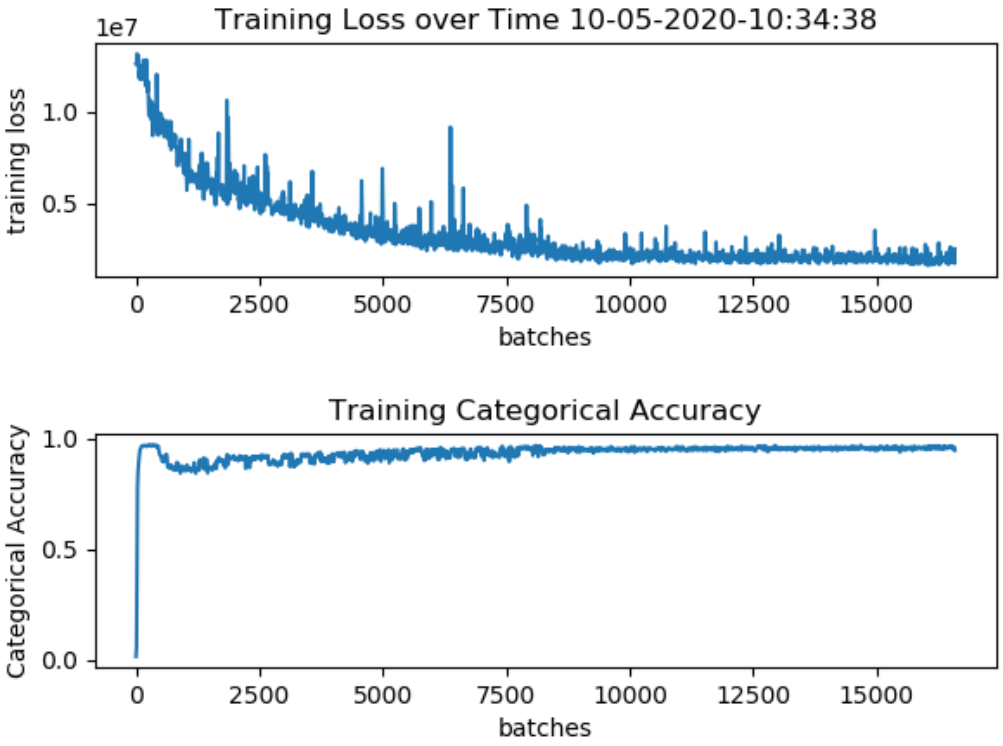

Figure 7: The loss and categorical accuracy during optimization on the pre-training set. Note that while loss decreases over time, the categorical accuracy is not always increasing. This shows why categorical accuracy is not a great metric for 3D segmentations.

2.5 Development and Implementation details

Network development and training were entirely carried out on the University of North Carolina's Longleaf computing server [42]. Development of the network model and training loop was done using TensorFlow [43] version 2.2.1. Some pre-processing steps rely on the SimpleITK [44] package for handling MR data. ITK-SNAP [45] was used for rendering 3D images of segmentations. Training was done using NVIDIA GeForce GTX 1080 and Tesla V100-SXM2 GPUs interchangeably. Preliminary testing determined that a batch size of 2 was the maximum that would allow training on 8 gigabytes

of GPU memory without causing an out-of-memory error.

A note about the implementation of the optimization loop is given here. The TensorFlow class `tf.GradientTape` is used in the optimization loop, which loops over training batches. A `GradientTape` instance is set to "watch" the trainable parameters of the network during inference and computation of the loss and then calculate the gradient of the loss with respect to the network parameters. A method from `tf.keras.optimizers.Adam` is used to apply the gradient, updating the parameters of the network. This optimization implementation paradigm is relatively new. The open-source code associated with this project is available in a GitHub repository. The scripts `pretrain.py` and `train.py` are used for optimization on the pre-training and training datasets, respectively.

The associated open-source software, NetSeg, was developed using QT [46]. The software has command line and graphical interface functionality. It allows users to input T1 and T2 MR images that have been pre-processed according to [37] in any common file format used for MR images and outputs segmentation images and probability maps (the raw output of the network, with softmax applied). At the time of writing, NetSeg is still under development and will be completed by October 19, 2020.

3 Results

Networks optimized on the training set with $\eta = 0.01, 0.005, 0.0025$ were evaluated on a set of 5 manually-segmented validation images. Each network began optimization on the training set with the same set of parameters obtained from optimization on the pre-training set. Metrics used for evaluation are the Dice Similarity Coefficient (DSC), Relative Volume Difference (RVD), and Mean Absolute surface Distance (MAD). DSC is given by

$$\text{DSC} = 2 \times \frac{|V_{\text{seg}} \cap V_{\text{label}}|}{|V_{\text{seg}}| + |V_{\text{label}}|} \times 100\%$$

RVD is given by

$$\text{RVD} = \frac{|V_{\text{seg}}| - |V_{\text{label}}|}{|V_{\text{label}}|} \times 100\%$$

and MAD is the average absolute surface distance, in millimeters, between the segmentation and the ground truth (label) images.

η	**Structure**	**DSC** (%)	**RVD** (%)
0.01	Amygdala	58.9 ± 4.06	89.7 ± 37.12
0.01	Caudate	60.8 ± 6.13	108 ± 32.9
0.01	Hippocampus	49.5 ± 7.13	122 ± 11.5
0.01	Glob. Pall.	67.7 ± 3.06	53.8 ± 21.9
0.01	Putamen	70.1 ± 4.296	56.0 ± 10.5
0.01	Thalamus	73.7 ± 3.53	52.4 ± 24.9
0.005	Amygdala	66.4 ± 4.10	49.8 ± 33.1
0.005	Caudate	67.1 ± 4.07	63.7 ± 22.3
0.005	Hippocampus	55.8 ± 7.26	70.7 ± 11.19
0.005	Glob. Pall.	69.3 ± 2.65	28.2 ± 19.0
0.005	Putamen	73.2 ± 3.95	38.4 ± 8.98
0.005	Thalamus	76.4 ± 2.63	40.3 ± 20.9
0.0025	Amygdala	69.6 ± 3.54	19.6 ± 29.2
0.0025	Caudate	70.2 ± 2.50	35.6 ± 16.6
0.0025	Hippocampus	58.0 ± 8.00	26.5 ± 9.9
0.0025	Glob. Pall.	63.0 ± 4.33	-11.9 ± 16.0
0.0025	Putamen	74.2 ± 4.02	19.8 ± 6.63
0.0025	Thalamus	76.2 ± 2.28	36.4 ± 21.8

Table 2: Evaluation of segmentations of models trained with different values of η, which multiplies the volumetrically weighted loss. DSC and RVD are the metrics used. Averages and standard deviations are taken across the 5 validation images. For DSC, higher is better, for RVD, closer to zero is better.

Table 2 gives DSC and RVD scores for each segmented structure for models trained with $\eta = 0.01, 0.005, 0.0025$. In this table, we see that the lower η is, the better the results are, in general. For reference, the AutoSeg pipeline [9] achieves average DSC scores in the 70s to high 80s on varying structures and much better RVD scores compared to the results here. We see that the RVD is almost always positive in Table 2, indicating that the structures were over-segmented by significant margins in almost every case.

Table 3 gives the MAD for the different values of η. We see that a lower value of η begets better results in terms of MAD. For reference, the AutoSeg pipeline achieves an average MAD of approximately 0.5 mm over 27 segmentation categories.

η	**MAD** (mm)
0.01	1.76 ± 0.203
0.005	1.36 ± 0.278
0.0025	1.20 ± 0.160

Table 3: Average MAD of segmentations of models trained with different values of η. Averages are taken across structures and validation images. Standard deviations are taken across validation images. For MAD, lower is better.

The following figures show segmentation images generated by networks trained with different values of η. Figure 8 shows 3D segmentations of the same image by models trained with different values of η. Structures are visibly smaller in segmentations with lower η. There are some common flaws in all the segmentations: the two thalami (pink) and the two caudate nuclei (green) are connected. There exist some disconnected artifacts in the segmentations: two (red) structures towards the bottom of the image with $\eta = 0.01$ and some disconnected (pink and dark blue) structures towards the rear of the brain, at the forefront of the images in the right column. We also observe that the surfaces of segmented structures are quite rough, or "bumpy," compared to the smooth surfaces of the ground truth. The shapes of the segmented structures are similar for different values of η, probably because all three models stem from the same pre-training parameters.

Figure 9 shows 2D slices of segmentations by the different models. Again we see that lower η corresponds to decreased volume of segmented structures. In the 2D image, artifacts around the boundary of the structure are more noticeable: isolated voxels exist outside and nearby structures, contributing to the bumpiness of the surfaces. There are also isolated background voxels on the interior of structures. At $\eta = 0.0025$, we see 3 isolated background voxels on the interior of the caudate nucleus (dark blue) in the left column that are not present in $\eta = 0.005, 0.01$. Conversely, we see background voxels in the interior of the thalamus (yellow) in the left column in $\eta = 0.005, 0.01$ but not $\eta = 0.0025$. These artifacts persist despite the post-processing steps taken after network segmentation.

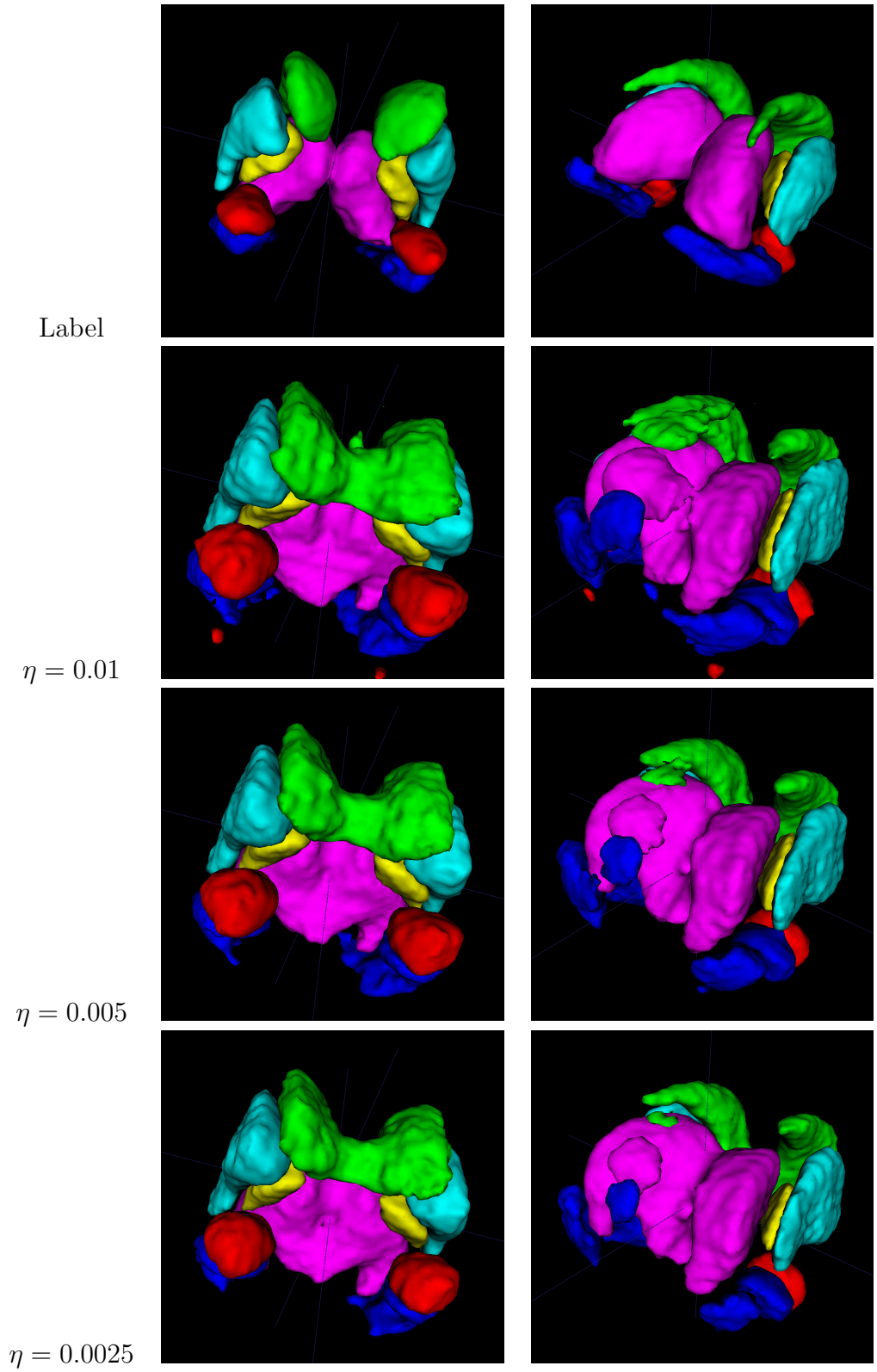

Figure 8: Comparison of a ground truth segmentation (top) with segmentations from models trained with different values of η. Both columns are the same validation image from different points of view.

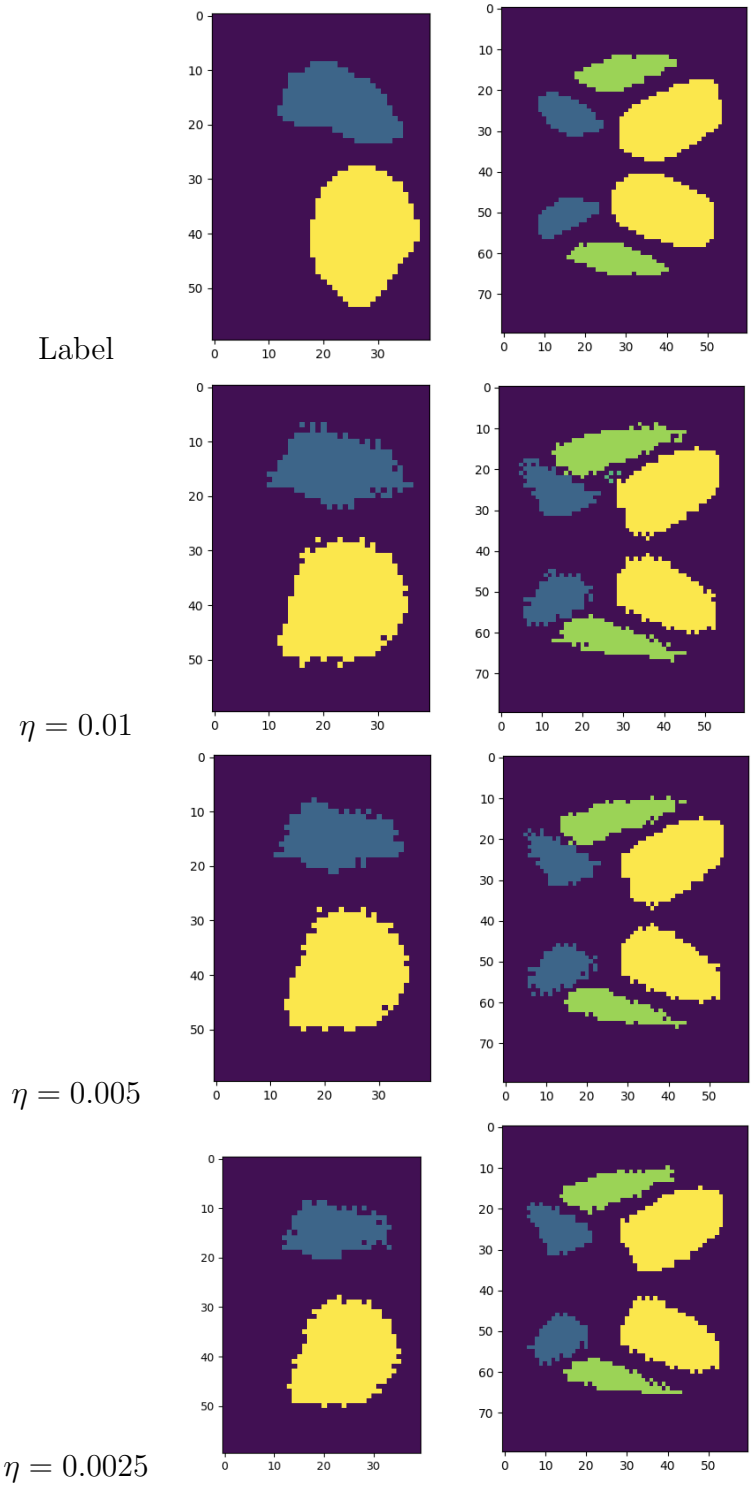

Figure 9: Comparison of a ground truth segmentation (top) with 2D slices of segmentations from models trained with different values of η. The structures shown are the caudate nucleus (dark blue), putamen (green), and thalamus (yellow).

4 Discussion

I trained a 3D u-net for the task of segmenting subcortical structures in brain MR images of 12 and 24 month infants. The network was trained on a pre-training set of 1660 pre-processed automatically segmented images and a training set of 54 pre-processed manually segmented images, with a loss dependent on the average volume of segmented structures and presence of boundary voxels. It is found that the network does not perform as well on evaluation metrics as the AutoSeg pipeline [9] but does obtain moderately good results. There are several reasons why the results of this network were limited and future actions that could be taken to improve results, which are discussed in the next sections.

4.1 Limitations

The current network is unoptimized in a number of ways. One such matter is the volumetric weighting scheme, where the loss corresponding to a particular structure is weighted according to its relative volume. This scheme is needed to ensure that the small structures are segmented accurately. Without it the network does not have sufficient incentive to accurately segment these structures since in some cases they only account for $1/500^{\text{th}}$ of the (cropped) 3D image. The loss weighting given to the background voxels, however, is much smaller than that of the subcortical structures. Therefore, the network is inclined to over-segment the structures at their boundary, especially the small structures, since there is little penalty for misclassifying the background voxels. This problem is multiplied when we examine the background voxels that lie between structures, especially when the structures are close together. These tiny volumes of background are especially important for accurate surface segmentation. While the boundary-weighted loss should mitigate the issue in theory, I found in testing that it was not effective. I speculate that the ineffectiveness of the boundary-weighted loss to solve over-segmentation is that the background around the boundary of a structure "looks" similar to the structure itself, as the receptive field of these voxels does overlap with the nearby structure, so classifying these background voxels correctly comes at the cost of misclassifying some voxels of the structure, which is heavily penalized due to the volumetric weighting scheme.

There is another problem with the weighted loss in relation to overfitting. I argue that the network does not suffer greatly from a problem of overfitting

in general, as there is much more training data than there are model parameters. The dropout layer also mitigates overfitting. However, if we limit our view to the network's ability to correctly segment boundary voxels and small structures, there is a case to be made for overfitting, since the boundary and small structure voxels make up approximately $1/100^{\text{th}}$ of the training data, reducing the data-to-parameters ratio. Next steps to address these problems are discussed below.

4.2 Further Work

An immediate step that can be taken to improve the results of this model is hyperparameter optimization. Further investigation is needed to find optimal hyperparameters η and ω, which control the influence of the volumetrically weighted and boundary weighed losses, respectively. Much of the poor results found in preliminary testing were due to suboptimal setting of these hyperparameters. It would also be interesting to adjust η and ω over the course of training the network, with the idea that the network will start by over-segmenting the structures. Then as the weighting of the background is increased, the surfaces of the output segmentation approach the ground truth surfaces. Additionally, the number of convolutional layers and number of filters per convolutional layer can be adjusted, to increase the number of parameters in the hopes of improving performance. In the post-processing steps, we can optimize the 0.5 probability threshold for obtaining the segmentation from the probability map, as well as the number of voxel neighbors with common label needed for voxel re-labeling. The artificial structures in the output segmentations shown in Figure 8 that do not correspond to any structures in the ground truth are detrimental to the DSC and RVD scores and must be removed with better post-processing.

Another interesting route of investigation would be to make use of a distance tensor, rather than a boundary tensor, in the loss function. In a distance tensor, each voxel takes the value of the minimum distance between it and a boundary voxel. With the distance tensor, the voxel-wise loss can be weighted so as the voxels closer to the boundary are weighted higher than those farther away. This would allow for high-weighting of background voxels that do not lie on the boundary but are nearby. With the current model, the low weighting of important background voxels is a major problem.

There are also a number of more significant changes to the network architecture that can be made. Zeng and Zheng [47] use a late-fusion strategy,

where information from T1 and T2 is not combined until the bottom level of the network is reached. Furthermore, their model is composed of two separate u-nets. The first is used to generate a segmentation which is converted into a distance tensor. The second u-net takes the distance tensor from the first, as well as the original images, to compute a final segmentation. To better combine the coarse and fine features, Nie et al. [48] utilize a version of the u-net with transformation and fusion modules in place of the u-net's skip connections (concatenations).

Lastly, this network should be applied to other segmentation tasks, so that results can be accumulated across a variety of tasks. For instance, segmentation of white matter, gray matter, and cerebrospinal fluid in MR images of 6-month subjects is a common application in this field. I also expect that results would improve if the network was tasked with binary segmentation, where each voxel is classified as either background or as part of the structure of interest. Then, a distinct network could be trained for each structure, and their combined outputs could be taken as the multi-categorical segmentation.

4.3 Conclusion

There is still much work to be done. I am hopeful that in the near future, deep learning-based methods will perform as well as state-of-the-art methods such as the multi atlas-based AutoSeg pipeline on the task of segmenting subcortical structures from MR images from 12 and 24 month subjects, and possibly on other tasks as well. The state of deep learning is evolving rapidly, and new techniques may prove to be more fruitful.

5 Acknowledgements

I cannot express enough gratitude towards Dr. Martin Styner, who was a wise, caring, and knowledgeable advisor to me during this project. I am also very grateful for the assistance afforded to me by the NIRAL team, especially Dr. Juan Prieto and Dr. Mahmoud Mostapha. I thank Dr. Stephen Pizer for reading this thesis and the SURF committee for funding my research during the Summer of 2020.

References

[1] M. Silveira, J. C. Nascimento, J. S. Marques, A. R. S. Marcal, T. Mendonca, S. Yamauchi, J. Maeda, and J. Rozeira, "Comparison of segmentation methods for melanoma diagnosis in dermoscopy images," *IEEE journal of selected topics in signal processing*, vol. 3, pp. 35–45, feb 2009.

[2] F. Shi, J. Wang, J. Shi, Z. Wu, Q. Wang, Z. Tang, K. He, Y. Shi, and D. Shen, "Review of artificial intelligence techniques in imaging data acquisition, segmentation and diagnosis for COVID-19.," *IEEE reviews in biomedical engineering*, vol. PP, apr 2020.

[3] C. M. Schumann, J. Hamstra, B. L. Goodlin-Jones, L. J. Lotspeich, H. Kwon, M. H. Buonocore, C. R. Lammers, A. L. Reiss, and D. G. Amaral, "The amygdala is enlarged in children but not adolescents with autism; the hippocampus is enlarged at all ages.," *The Journal of Neuroscience*, vol. 24, pp. 6392–6401, jul 2004.

[4] X. Lladó, A. Oliver, M. Cabezas, J. Freixenet, J. C. Vilanova, A. Quiles, L. Valls, L. Ramió-Torrentà, and Rovira, "Segmentation of multiple sclerosis lesions in brain MRI: A review of automated approaches," *Information sciences*, vol. 186, pp. 164–185, mar 2012.

[5] T. G. M. van Erp, D. P. Hibar, J. M. Rasmussen, D. C. Glahn, G. D. Pearlson, O. A. Andreassen, I. Agartz, L. T. Westlye, U. K. Haukvik, A. M. Dale, I. Melle, C. B. Hartberg, O. Gruber, B. Kraemer, D. Zilles, G. Donohoe, S. Kelly, C. McDonald, D. W. Morris, D. M. Cannon, A. Corvin, M. W. J. Machielsen, L. Koenders, L. de Haan, D. J. Veltman, T. D. Satterthwaite, D. H. Wolf, R. C. Gur, R. E. Gur, S. G. Potkin, D. H. Mathalon, B. A. Mueller, A. Preda, F. Macciardi, S. Ehrlich, E. Walton, J. Hass, V. D. Calhoun, H. J. Bockholt, S. R. Sponheim, J. M. Shoemaker, N. E. M. van Haren, H. E. Hulshoff Pol, R. A. Ophoff, R. S. Kahn, R. Roiz-Santiañez, B. Crespo-Facorro, L. Wang, K. I. Alpert, E. G. Jönsson, R. Dimitrova, C. Bois, H. C. Whalley, A. M. McIntosh, S. M. Lawrie, R. Hashimoto, P. M. Thompson, and J. A. Turner, "Subcortical brain volume abnormalities in 2028 individuals with schizophrenia and 2540 healthy controls via the ENIGMA consortium.," *Molecular Psychiatry*, vol. 21, pp. 547–553, apr 2016.

[6] R. Geevarghese, D. E. Lumsden, N. Hulse, M. Samuel, and K. Ashkan, "Subcortical structure volumes and correlation to clinical variables in parkinson's disease.," *Journal of Neuroimaging*, vol. 25, pp. 275–280, apr 2015.

[7] M. P. Laakso, K. Partanen, P. Riekkinen, M. Lehtovirta, E. L. Helkala, M. Hallikainen, T. Hanninen, P. Vainio, and H. Soininen, "Hippocampal volumes in alzheimer's disease, parkinson's disease with and without dementia, and in vascular dementia: An MRI study.," *Neurology*, vol. 46, pp. 678–681, mar 1996.

[8] C. Lange and E. Irle, "Enlarged amygdala volume and reduced hippocampal volume in young women with major depression.," *Psychological Medicine*, vol. 34, pp. 1059–1064, aug 2004.

[9] J. Wang, C. Vachet, A. Rumple, S. Gouttard, C. Ouziel, E. Perrot, G. Du, X. Huang, G. Gerig, and M. Styner, "Multi-atlas segmentation of subcortical brain structures via the AutoSeg software pipeline.," *Frontiers in neuroinformatics*, vol. 8, p. 7, 2014.

[10] M. Helt, E. Kelley, M. Kinsbourne, J. Pandey, H. Boorstein, M. Herbert, and D. Fein, "Can children with autism recover? if so, how?," *Neuropsychology Review*, vol. 18, pp. 339–366, dec 2008.

[11] "Screening and diagnosis of autism spectrum disorder |CDC."

[12] H. C. Hazlett, H. Gu, B. C. Munsell, S. H. Kim, M. Styner, J. J. Wolff, J. T. Elison, M. R. Swanson, H. Zhu, K. N. Botteron, D. L. Collins, J. N. Constantino, S. R. Dager, A. M. Estes, A. C. Evans, V. S. Fonov, G. Gerig, P. Kostopoulos, R. C. McKinstry, J. Pandey, S. Paterson, J. R. Pruett, R. T. Schultz, D. W. Shaw, L. Zwaigenbaum, J. Piven, I. Network, C. Sites, D. C. Center, I. P. Core, and S. Analysis, "Early brain development in infants at high risk for autism spectrum disorder.," *Nature*, vol. 542, pp. 348–351, feb 2017.

[13] M. A. Deeley, A. Chen, R. Datteri, J. H. Noble, A. J. Cmelak, E. F. Donnelly, A. W. Malcolm, L. Moretti, J. Jaboin, K. Niermann, E. S. Yang, D. S. Yu, F. Yei, T. Koyama, G. X. Ding, and B. M. Dawant, "Comparison of manual and automatic segmentation methods for brain structures in the presence of space-occupying lesions: a multi-expert

study.," *Physics in Medicine and Biology*, vol. 56, pp. 4557–4577, jul 2011.

[14] M. Mostapha and M. Styner, "Role of deep learning in infant brain MRI analysis.," *Magnetic resonance imaging*, vol. 64, pp. 171–189, Dec. 2019.

[15] A. Makropoulos, I. S. Gousias, C. Ledig, P. Aljabar, A. Serag, J. V. Hajnal, A. D. Edwards, S. J. Counsell, and D. Rueckert, "Automatic whole brain MRI segmentation of the developing neonatal brain.," *IEEE transactions on medical imaging*, vol. 33, pp. 1818–1831, sep 2014.

[16] P. Moeskops, M. J. N. L. Benders, S. M. Chiţ, K. J. Kersbergen, F. Groenendaal, L. S. de Vries, M. A. Viergever, and I. Išgum, "Automatic segmentation of MR brain images of preterm infants using supervised classification.," *Neuroimage*, vol. 118, pp. 628–641, sep 2015.

[17] N. I. Weisenfeld and S. K. Warfield, "Automatic segmentation of newborn brain MRI.," *Neuroimage*, vol. 47, pp. 564–572, aug 2009.

[18] L. Wang, F. Shi, P.-T. Yap, J. H. Gilmore, W. Lin, and D. Shen, "4D multi-modality tissue segmentation of serial infant images.," *Plos One*, vol. 7, p. e44596, sep 2012.

[19] J. E. Iglesias and M. R. Sabuncu, "Multi-atlas segmentation of biomedical images: A survey.," *Medical Image Analysis*, vol. 24, pp. 205–219, aug 2015.

[20] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *The Bulletin of mathematical biophysics*, vol. 5, pp. 115–133, dec 1943.

[21] K. Fukushima, "Neocognitron: a self organizing neural network model for a mechanism of pattern recognition unaffected by shift in position.," *Biological Cybernetics*, vol. 36, no. 4, pp. 193–202, 1980.

[22] K. Chellapilla, S. Puri, and P. Simard, "High performance convolutional neural networks for document processing," oct 2006.

[23] W. Zhang, R. Li, H. Deng, L. Wang, W. Lin, S. Ji, and D. Shen, "Deep convolutional neural networks for multi-modality isointense infant brain image segmentation.," *Neuroimage*, vol. 108, pp. 214–224, mar 2015.

[24] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention (MICCAI)* (N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, eds.), vol. 9351 of *Lecture notes in computer science*, pp. 234–241, Cham: Springer International Publishing, 2015.

[25] P. Moeskops, M. A. Viergever, A. M. Mendrik, L. S. de Vries, M. J. N. L. Benders, and I. Isgum, "Automatic segmentation of MR brain images with a convolutional neural network.," *IEEE transactions on medical imaging*, vol. 35, pp. 1252–1261, mar 2016.

[26] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Mathematics of Control, Signals, and Systems*, vol. 2, pp. 303–314, dec 1989.

[27] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[28] P. J. Huber, "Robust estimation of a location parameter," *The Annals of Mathematical Statistics*, vol. 35, pp. 73–101, mar 1964.

[29] H. Robbins, "[PDF] a stochastic approximation method |semantic scholar," *undefined*, 2007.

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

[31] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," in *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 1026–1034, IEEE, dec 2015.

[32] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," *Journal of Machine Learning Research*, jan 2010.

[33] M. Styner, J. Lee, B. Chin, M. S. Chin, and S. Warfield, "3D segmentation in the clinic: A grand challenge II: MS lesion segmentation," nov 2007.

[34] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," *arXiv*, nov 2015.

[35] "Recruitment."

[36] H. C. Hazlett, H. Gu, R. C. McKinstry, D. W. W. Shaw, K. N. Botteron, S. R. Dager, M. Styner, C. Vachet, G. Gerig, S. J. Paterson, R. T. Schultz, A. M. Estes, A. C. Evans, J. Piven, and I. Network, "Brain volume findings in 6-month-old infants at high familial risk for autism.," *The American Journal of Psychiatry*, vol. 169, pp. 601–608, jun 2012.

[37] M. R. Swanson, M. D. Shen, J. J. Wolff, J. T. Elison, R. W. Emerson, M. Styner, H. C. Hazlett, K. Truong, L. R. Watson, S. Paterson, N. Marrus, K. N. Botteron, J. Pandey, R. T. Schultz, S. R. Dager, L. Zwaigenbaum, A. M. Estes, J. Piven, and IBIS Network, "Subcortical Brain and Behavior Phenotypes Differentiate Infants With Autism Versus Language Delay.," *Biological psychiatry. Cognitive neuroscience and neuroimaging*, vol. 2, pp. 664–672, Nov. 2017.

[38] "Home page |UNC autism research center."

[39] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?," *arXiv*, nov 2014.

[40] L. G. Nyúl, J. K. Udupa, and X. Zhang, "New variants of a method of MRI scale standardization.," *IEEE transactions on medical imaging*, vol. 19, pp. 143–150, feb 2000.

[41] Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger, "3D u-net: Learning dense volumetric segmentation from sparse annotation," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2016* (S. Ourselin, L. Joskowicz, M. R. Sabuncu, G. Unal, and W. Wells, eds.), vol. 9901 of *Lecture notes in computer science*, pp. 424–432, Cham: Springer International Publishing, 2016.

[42] "Longleaf cluster - information technology services."

[43] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah,

M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015. Software available from tensorflow.org.

[44] "SimpleITK - home."

[45] "ITK-SNAP home."

[46] "Qt |cross-platform software development for embedded & desktop."

[47] G. Zeng and G. Zheng, "Multi-stream 3D FCN with multi-scale deep supervision for multi-modality isointense infant brain MR image segmentation," in *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*, pp. 136–140, IEEE, apr 2018.

[48] D. Nie, L. Wang, E. Adeli, C. Lao, W. Lin, and D. Shen, "3-d fully convolutional networks for multimodal isointense infant brain image segmentation.," *IEEE transactions on cybernetics*, vol. 49, pp. 1123–1136, mar 2019.

Acronyms

ASD Autism Spectrum Disorder. 1, 4, 5, 12, 16

CDC Centers for Disease Control. 4

CNN Convolutional Neural Network. 2, 5–7, 10, 11

CPU Computer Processing Unit. 6

DSC Dice Similarity Coefficient. 3, 18, 19, 24

GPU Graphics Processing Unit. 6, 17, 18

IBIS Infant Brain Imaging Study. 12, 13

IoU Intersection over Union. 10

MAD Mean Absolute surface Distance. 3, 10, 18–20

MR Magnetic Resonance. 1–7, 11–13, 17, 18, 23, 25

NDD Neurodevelopmental Disorder. 4

NIRAL Neuro Image Research and Analysis Laboratories. 25

ReLU Rectified Linear Unit. 7, 10, 15

RVD Relative Volume Difference. 3, 18, 19, 24

T1 T1-Weighted. 5, 12, 16, 18, 25

T2 T2-Weighted. 12, 16, 18, 25

VWF Volumetric Weighting Factor. 3, 13–16