MULTIDIMENSIONAL ATTENTION BASED NEURAL NETWORK FOR 3D IMAGE SEGMENTATION

by

RUTU GANDHI

(Under the Direction of Yi Hong)

ABSTRACT

Segmenting an entire 3D image often has high computational complexity and requires large memory consumption; by contrast, performing volumetric segmentation in a slice-by-slice manner is efficient but does not fully leverage the 3D data. To address this challenge, we propose a multi-dimensional attention network (MDA-Net) to efficiently integrate slice-wise, spatial, and channel-wise attention into a U-Net based network, which results in high segmentation accuracy with a low computational cost. We evaluate our model on the MICCAI iSeg and IBSR datasets, and the experimental results demonstrate consistent improvements over existing methods. For the IBSR dataset, we report an average dice score of 95.12%, 97.27%, 95.04% on the Sagittal view, 94.89%, 95.31%, 92.01% on Axial and 94.52%, 91.45%, 92.22% Coronal for the cerebrospinal fluid, white matter and gray matter respectively.

INDEX WORDS: Attention network, 3D image segmentation, Squeeze and excitation block

MULTIDIMENSIONAL ATTENTION BASED NEURAL NETWORK FOR 3D IMAGE SEGMENTATION

by

RUTU GANDHI

B.Eng. Pune Institute of Computer Technology, Pune, India, 2018

A Thesis Submitted to the Graduate Faculty of The University of Georgia in Partial Fulfillment of the Requirements for the Degree

MASTER OF SCIENCE

ATHENS, GEORGIA

2021

MULTIDIMENSIONAL ATTENTION BASED NEURAL NETWORK FOR 3D IMAGE SEGMENTATION

by

RUTU GANDHI

Major Professor: Yi Hong

Committee: Frederick Maier, Tianming Liu

Electronic Version Approved:

Ron Walcott

Dean of the Graduate School

The University of Georgia

May 2021

ACKNOWLEDGEMENTS

I would like to thank all my committee members for their time and the numerous suggestions they provided throughout my research. In particular, I would like to thank Dr. Hong for her guidance, constant encouragement and motivation to keep me going. I owe a great debt to Raunak Dey and Ankita Joshi who facilitated my introduction into the subject, and whose support laid the foundation for my work. I am thankful to my family and friends who kept me sane during these unprecedented times, where the whole world has come to a standstill due to COVID-19. I would like to thank everyone at the *Institute for Artificial Intelligence* who presented me with an opportunity to be a part of this project. I would also like to thank Ms. Tino, whose assistance with the administrative tasks at the university made my life incredibly easy.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Image segmentation is a fundamental task in image understanding, which distinguishes regions of interest from image background for further analysis. Recently, deep segmentation networks, e.g., fully convolutional networks (FCN) introduced by Long et al. [17], U-Net by Ronnenberger et al. [24], tackle the 2D image segmentation problem and outperform conventional approaches. However, segmenting 3D image volume like brain MRI scans is still challenging, especially in medical image analysis.

Models extended from FCNs and U-Nets have been proposed to handle the 3D image segmentation, e.g., V-Net by Milletari et al. [19]. Due to the high-dimensional nature of image data, most existing models have a high demand for computational resources, especially the GPU memory, and often have a large number of parameters to estimate. Upon trying some 3D segmentation models, the limited GPU memory that we had available could not fit the entire 3D dataset at once. Another solution that similar papers talk about as we'll see in the literature review is to chunk the 3D volumes, perform image segmentation on the chunks independently and then combine the results back into the whole volumes. This method fails to fully leverage the 3rd dimension.

The 2D counterparts of the above image segmentation models have lower memory consumption since they slice the 3D data and look at each slice individually but this again results in not leveraging the 3rd-dimensional information.

In this thesis, we propose an economical solution for segmenting 3D image volume, which roots on a 2D network and handles 2.5D data by augmenting a 2D image slice with an additional image condensed from the third dimension with attention. To integrate the information across slices of an image volume, we propose a compression technique based on the squeeze and excitation (SE) technique proposed by Hu et al. [8] to concisely abstract multiple neighboring slices of a volume into a single slice. Apart from the augmented input for the 2D backbone network, we also augment the network with spatial and channel-wise attention by upgrading the concurrent SE block proposed by

Roy et al. [25]. The resulting network benefits from both the low computation cost of the 2D network and enriched information from image volumes and learned features with attention. Figure 8 depicts the overall architecture of our multi-dimensional attention network (MDA-Net) for segmenting 3D images. The MDA-Net is an end-to-end solution and can automatically learn how to compress volumetric image information and extract useful features in an attention scheme.

In particular, our MDA-Net aims to mimic the process of the manual segmentation for a 3D image. When handling an image volume, we often select one main view to sequentially segment 2D slices and check the third dimension across slices occasionally to obtain additional information. To integrate the information among image slices, we condense the ordered slice difference computed with respect to the current main slice. This compression step allows us to collect extra information, i.e., image residuals, to assist the segmentation. Another benefit of using our model is increased data samples. If we have p slices in one view, we can convert one 3D volume into at most p samples, and each sample contains a 2D slice and another slice compressed from the original volume. This compression is achieved by using slice-wise attention. We also have the spatial and channel-wise attention used in the image segmentation network, resulting in our multi-dimensional attention network. The attention is automatically estimated via modified squeeze and excitation (MSE) blocks, which improve segmentation performance over the original SE block and concurrent scSE block.

1.1 MAIN CONTRIBUTIONS

The main contributions of this paper are as follows:

- Modified SE (MSE) Block: We upgrade the way channel attention mechanism in the concurrent scSE and replace the sigmoid function with softmax to ensure the weights for measuring the attention to be normalized.
- Slice-wise condensing module: We propose a compression module that uses slice-wise attention to extract residual information in the third dimension.
- Multi-Dimension Attention Network (MDA-Net): We propose an efficient 3D image segmentation network, which fully leverages the 3D data with a balanced computational cost.

1.2 PERFORMANCE METRICS

We evaluate our MDA-Net on the MICCAI iSeg dataset published by Wang et al. [30] and the IBSR dataset published by Rohl et al. [23] through segmenting 3D brain scans. The segmentation results on both datasets show the improvement over previous methods with five-fold cross-validation. For the iSeg dataset, we report an average dice score of 95.12%, 97.27%, 95.04% respectively for cerebrospinal fluid, gray matter, white matter on the Sagittal view of our model, 94.89%, 95.31%, 92.01% respectively on the Axial View and 94.52%, 91.45%, 92.22% respectively on the Coronal View. Along with the region-wise dice scores, we also compare the number of trainable parameters that each model has. We see that replacing the global average pooling and dense layers of the cscSE architecture results in a drop in the number of parameters while still maintaining comparable performance. We further see that even though the MDA-Net is a 2.5D model the number of trainable parameters it has is equivalent to 2D architectures like the plain UNet, cscSE, and 2D attention-augmented net.

We further compare the inference times in seconds to make sure that our model even while using data from all three dimensions is not significantly increasing it. Upon comparing we see that the 2D baseline models and our model have comparable inference times.

CHAPTER 2

LITERATURE REVIEW

The 3D variants of the U-Net by Milletari e al. [19] and Cicek et al. [19, 3] were proposed to handle volumetric images. Compared to a 2D U-Net working slice by slice, its 3D version fully uses the data in all dimensions. However, 3D models face two main challenges in segmenting the entire high-dimensional image volume. We often have limited computational resources, especially limited GPU memory, to handle the whole image volume. Compared to a greatly increased number of model parameters when switching from a 2D network to 3D, we have a reduced number of data samples since a 2D slice sequence becomes one sample in a 3D U-Net. Existing approaches to address these challenges include downsampling the 3D images to fit in memory [3], assembling multiple 2D networks for accepting different image views proposed by Chlebus et al. [4], working on 3D image patches proposed by Liu et al. [16], modifying the existing segmentation architectures proposed by Lucas et al. [18] and Brugger et al. [18, 2] or combining 2D slices with 3D patches proposed by Dey and Hong [6].

Unlike previous approaches, we segment an image volume in the slice-to-slice fashion while integrating the residuals across slices. Work related to ours is the volumetric attention Mask-RCNN proposed by Wang et al. [31], which considers three adjacent slices when calculating attention. Our attention model is built based on the SE block, which is not limited to three slices and provides the first-order statistic across slices.

The following subsections talk about the different aspects of our model and relevant previous work. We conclude each subsection by stating how the multidimensional attention net (MDA-Net) compares with these previous architectures and their pros over them. We start off with 3D segmentation architectures followed by fully connected CNN models, models that use spatial squeeze and excite block, other attention and gating mechanisms used similarly concluding with architectures that were introduced to perform model acceleration.

2.1 3D SEGMENTATION

As introduced above, 3D variants of the UNet [19, 3] have been popularly used to perform 3D image segmentation. Even though they fully leverage information from all three dimensions, this property of theirs hurts their resource friendliness and these models often run into some critical issues as discussed below.

LIMITED GPU RESOURCES. We often have limited computational resources, especially limited GPU memory, to handle the whole image volume. Working with high-resolution image volumes particularly increases the resources required. To get around this, one might expect to consider images of lower resolution but this is problematic since it may give rise to problems of low-precision and miss detection especially when it comes to tissue segmentation. This would lead to a need to trade-off spatial resolution with the number of 2D slices used. This would further mean that there would a trade-off between the precision with which segmentation is performed and the amount of contextual information in the z dimension.

MODEL PARAMETERS VS NUMBER OF SAMPLES. Compared to a greatly increased number of model parameters when switching from a 2D network to 3D, we have a reduced number of data samples since a 2D slice sequence becomes one sample in a 3D U-Net. Existing approaches to address these challenges include downsampling the 3D images to fit in memory [3], assembling multiple 2D networks for accepting different image views [4], working on 3D image patches proposed by Howard et al. [7], or combining 2D slices with 3D patches [6].

Different works introduced ways to combat these shortcomings of the 3D models. Cicek et al. [3] as shown in Figure 1 introduced a model that only needs data with sparse annotations to train on and it outputs the corresponding dense annotations. Coupe et al. [5] as shown in Figure 2 proposed a patch based segmentation model to reduce the amount of computational resources required. Brugger et al. [2] as shown in Figure 3 proposed ways to modify the architecture of models. Liu et al. [16] as shown in Figure 4 proposed hierarchical architectures to perform coarse-grained and then fine-grained segmentation.

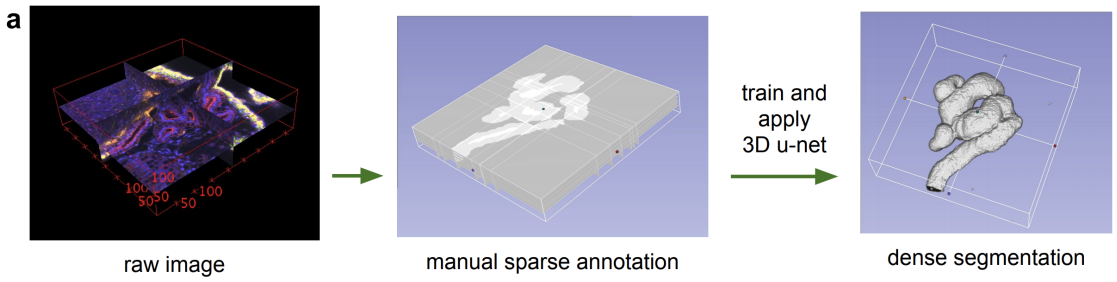

Figure 1: Sparse anotation as introduced by Cicek et al. [3] to combat the resource constraint problem faced in 3D segmentation.

Unlike previous approaches, we segment an image volume in the slice-to-slice fashion while integrating the residuals across slices. Work related to ours is the volumetric attention Mask-RCNN [31], which considers three adjacent slices when calculating attention. Our attention model is built based on the SE block, which is not limited to three slices and provides the first-order statistic across slices.

2.2 FULLY CONNECTED CNN ARCHITECTURES

FCNNs have been popularly used in biomedical images for precise segmentation. The UNet proposed by Ronnenberger et al. [24]. Figure 5 taken from the research is shown below. The DenseNet proposed by Iandola et al. [9] is another such state-of-the-art model that implements efficient convnet descriptor pyramids. For 3D volumes, though, they make 2D slices and segment each slice independently

2.3 USING SPATIAL AND CHANNEL SQUEEZE AND EXCITE

CNNs have been popularly used to tackle various tasks in the field of computer vision. The central building block of CNNs is the convolution operator that enables networks to construct informative features by fusing both spatial and channel-wise information within local receptive fields at each layer. CNNs interleave a series of convolution layers, activation layers and downsampling layers to produce image representation and global theoretical receptive fields. One of the goals of

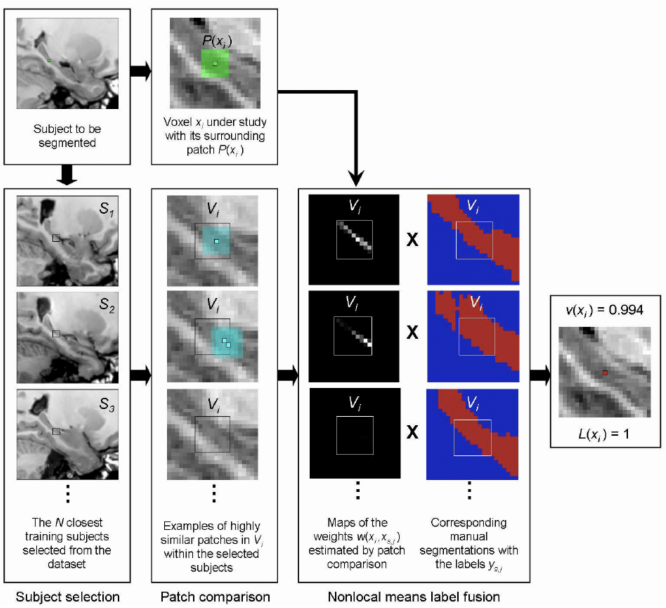

Figure 2: Patch based segmentation introduced in Coupe et al. [5]

computer vision has been to make these representations more powerful and in turn increase the performance and efficiency of the network. Recent research has focused on introducing learning mechanisms for the spatial correlation in images. One such research is the Inception net proposed by Szegedy et al. [28] which introduces multi-scale processes in the model to increase performance. Further works have focused on spatial attention by Bell et al. [1], Newell et al. [21].

The Squeeze and excite net by Hu et al. [8] changed the direction of network design by focusing on channel attention instead. They introduced the Squeeze and excite block that learns the channel inter-dependencies. This block explicitly models these dependencies between the channels of the convolution layer. They introduce the mechanism of feature recalibration that learns to use global information to selectively emphasize important features and suppress less useful ones. For every convolution block, they introduce a corresponding squeeze and excite block. The feature maps $\mathbf{U}$ outputted by a convolution block are first passed through a squeeze operation which aggregates the spatial information of a channel and produces a corresponding channel descriptor. The excite operation acts like a gating mechanism that takes in the channel descriptor array and outputs a collection of per-channel output weights. These weights are applied to the feature maps $\mathbf{U}$.

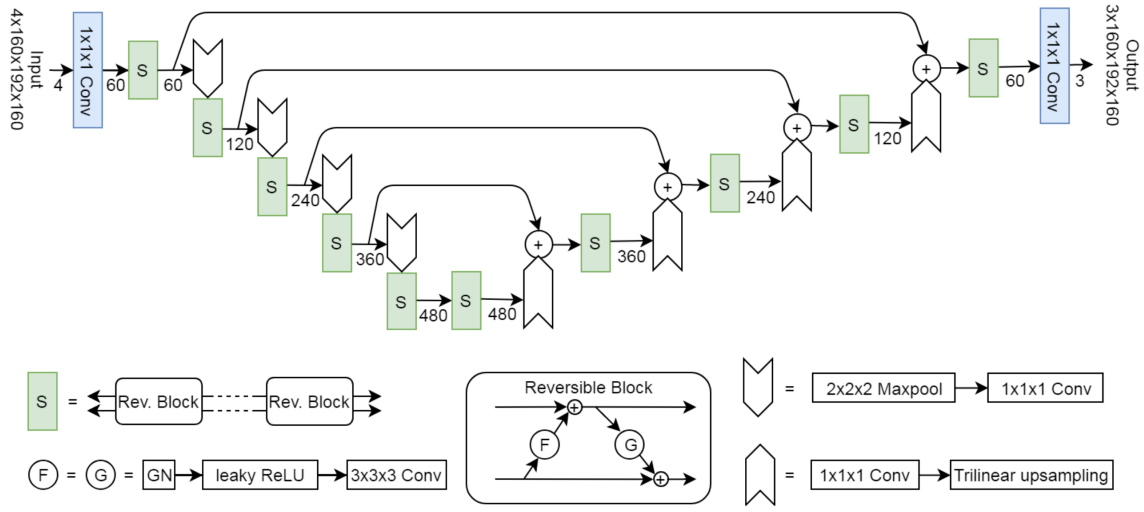

Figure 3: Modified structures introduced in Brugger et al. [2]. Top row: features extracted from coarse-grained voxel-based branch. Bottom row: features extracted from fine-grained pixel-based branch.

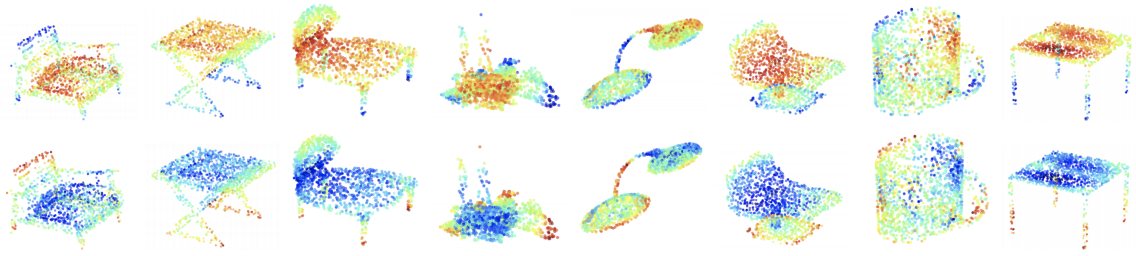

Figure 4: Liu et al. assemble different networks to deal with different views [16]

A convolution network with SE block won first place in the ILSVRC 2017 classification competition in the ImageNet dataset indicating its effectiveness.

The SE blocks were further modified to be used in image segmentation while focusing on the fully connected CNNs by Roy et al. [25]. In this work, the previous SE blocks are referred to as channel SE (cSE) blocks since they only excite in the channel dimension. They introduce two new versions of the SE block namely:

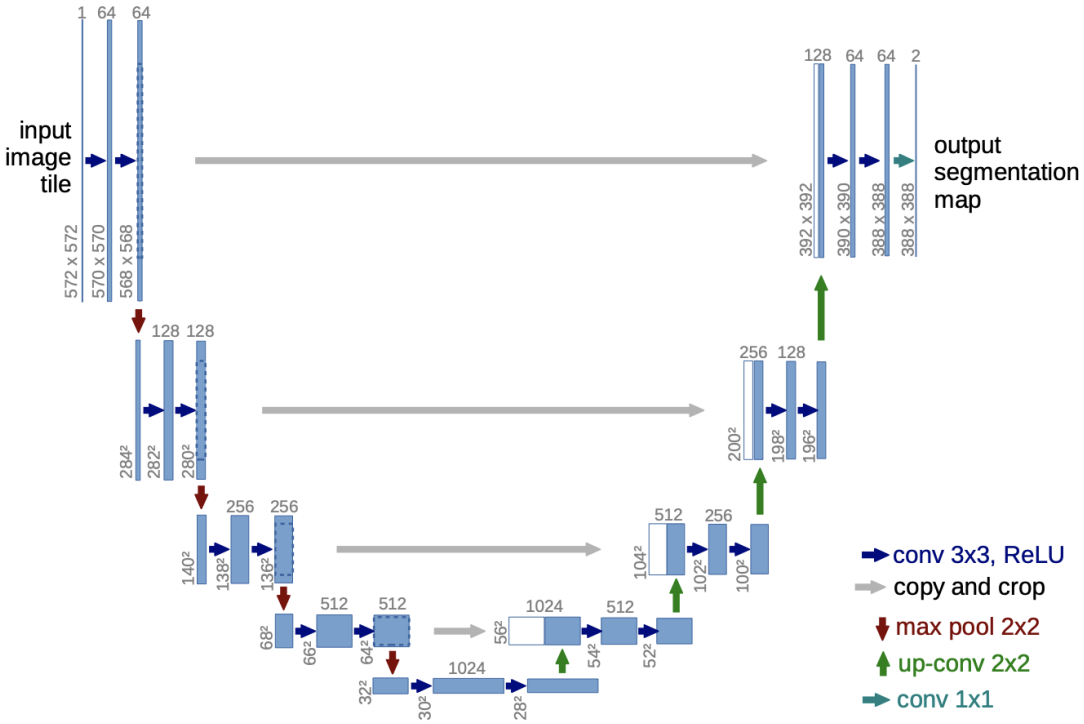

Figure 5: U-Net as introduced in [24]

1. spatial SE which squeezes along the channels and excites spatially;

2. concurrent spatial and channel SE which recalibrates the features along the spatial and channel dimension separately and then combines the output.

Roy et al. [25] explore an alternate direction of recalibrating the feature maps adaptively, to boost meaningful features, while suppressing weak ones. They concurrently perform channel squeeze, spatial excitation and channel excitation and spatial squeeze to achieve this. The model uses global pooling to perform spatial squeeze i.e. the 3D volume $U_{H \times W \times C}$ to a vector of dimension C. In our model, We break this transition into two steps. First a depthwise layer with kernel size $H \times 1$ is applied to $U_{H \times W \times C}$ giving $U_W \times C$. This is followed by another depthwise layer with kernel size $1 \times W$ to further reduce this to a vector of size C. The advantage of doing this is that the height dimension is considered separately followed by then considering the width dimension to get features of importance.

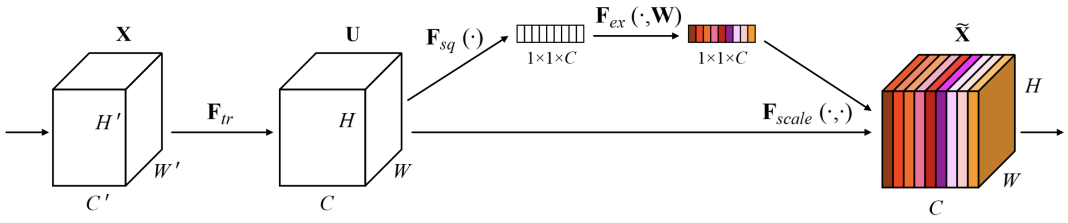

Figure 6: Squeeze and excite network as introduced in [8]

Keeping in mind that this modified way would introduce more parameters, in this paper, we introduce a further resource-constrained architecture as explained in the attention-augmented net. A similar argument can be made while comparing [31] to our work. They have an architecture inspired by [25] and [8] i.e. global pooling is used to summarize spatial information.

2.4 ATTENTION AND GATING MECHANISMS

Attention mechanisms are means of biasing the use of available computational resources towards features that are more important. Olshausen et al. [22] discuss a position and scale-invariant neurobiological model for attention. The representations of the object in the real world in this mechanism do not depend on its position and relative scale. This model relies on control neurons that strengthen the intracortical connections so that information from a particular viewing window is forwarded to higher cortical areas. Local spatial relationships are maintained as this transfer occurs. This allows the representations to be object-centered thus making them position and scale-invariant.

Itti et al. [12] present a model that combines multiscale image features into a single saliency map. A dynamic neural network then selects in order of decreasing saliency the attended locations. This system breaks down the problem of scene understanding by selecting conspicuous regions that need to be attended with greater detail thus increasing the computational efficiency.

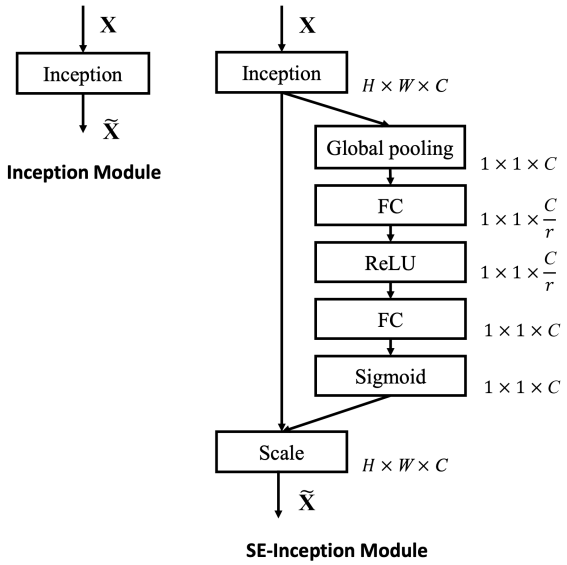

Figure 7: Plain inception architecture (left) inception architecture with squeeze and excite block (right) as introduced in [8]

Itti et al. [13] outline the trends that have emerged from computational literature. First, the saliency of stimuli depends on the context of the object in its surrounding. If the surrounding changes or the way that the object appears changes with respect to other objects, the saliency is bound to change. Computationally, this means that the contextual influence must be included in models. The second is a bottom-up control strategy which generates a saliency map that topographically encodes over the visual scene. Third, inhibition-of-return (IOR), the process in which the currently attended location is inhibited transiently. IOR implements a memory of the most recently visited locations. Without this attention would endlessly be attracted towards the most salient areas. Fourth, there is a strong interconnection between attention and eye movements. For models concerned with visual search, it is particularly important to study the interaction between overt and covert attention. Last, scene understanding and object recognition tend to keep attended locations constrained.

Larochelle et al. [15] introduce a third-order restricted Boltzman machine (RBM) that has the ability to gather information about shape over several fixations. The model decides a sequence of fixations that cover only a small area of the image since the available resolution is relatively low. It then combines the glimpse with the location of the fixation and integrates this information with other glimpses of the same object.

Recurrent models for visual attention introduced by Mnih et al. [20] extend the convolutional neural networks. The objective of the model is to tackle the problem of the linear increase in computational complexity of convolutional neural networks with the increase in the number of pixels that is the image size. Their model is capable of extracting information from an image or video. This it does by taking into a sequence of regions but only processing some selected regions at a high resolution. The model, like CNNs, has a translation-invariance built-in. Unlike the CNNs, though the computational complexity is not linearly dependent on the size of the image. The model is non-differentiable and can be trained using reinforcement learning to learn task-specific policies.

Vaswani et al. [29] present a novel model called the Transformer that completely gets rid of recurrence and convolutional layers. These models are proven to be more parallelizable and require significantly lesser time to train. To draw global dependencies between input and output, this model completely relies on attention mechanisms.

2.5 MODEL ACCELERATION

SqueezeNet proposed by Iandola et al. [10] reduces computation and parameters significantly while maintaining accuracy. SENet [8] boosts performance by introducing an architectural unit with slight computational cost. Mobilenet [7] utilizes the depthwise separable convolutions to reduce the computation and make the model lightweight while maintaining state-of-the-art performance. Recurrent models look at the symbol positions in the input and output and factor computation along them. A sequence of hidden states h_t that is a function of the previous hidden state h_{t-1} and the input for position t is generated by aligning the position to steps in computation time. This sequential nature prevents parallelization within training samples. At long sequence lengths, this becomes troublesome since memory constraints may start creeping in. Recent research has been

able to achieve parallelization using conditional computation proposed by Shazeer et al. [26] and factorization tricks proposed by Kuchaiev et al. [14]. Two factorization tricks are presented here: first, they talk about a method where the matrix is factorized by design where the LSTM matrix is represented as the product of two smaller matrices. The second factorization method partitions the matrix into the input and its states. Here we use the depthwise convolution in a way that flattens the 3D volume it is operating on in order to apply the same filter on all slices of the volume. This significantly reduces the number of trainable parameters.

CHAPTER 3

PROPOSED MODEL: MDA-NET

To provide an economical solution for 3D image segmentation, we adhere to treat the 3D data as a sequence of 2D image slices, which allows us to work in a lower-dimensional space with relatively low demand in computational resources. For a specific 2D slice, we augment it with the first-order information in the third dimension, which is achieved by condensing ordered image differences in its neighborhood into one slice. As shown in Fig. 8, our model has an image dimension reduction component using an ordering-based image difference compression. The resulting slice is concatenated with the associated 2D slice as inputs for an attention-augmented 2D U-Net. We stack the 2D segmentation masks back to form the segmentation mask for an image volume.

3.1 MDA-NET ARCHITECTURE

Figure 8 shows how all modules introduced above work together. The gray module shows the difference and order operation and how the right branch of the MSE block is used to compress its output, the blue module shows the attention-augmented UNet and the pink one shows the Modified SE Block. The following sub-chapters discuss these different components of the model followed by some important characteristics.

3.2 SLICE-WISE COMPRESSION

Assume we segment the i-th image slice, e.g., $I^{(i)}_{m\times n}$, of an image volume $I_{m\times n\times p}$, $i = 1, \cdots, p$. Besides the 2D slice, i.e., $I^{(i)}_{m\times n}$, we prepare an additional slice $\bar{I}^{(i)}_{m\times n}$ that associates with $I^{(i)}_{m\times n}$ and contains information across the slices. The concatenation, $I^{(i)}_{m\times n} \cup \bar{I}^{(i)}_{m\times n}$, is the input of our attention-augmented U-Net. This new presentation allows us to take full use of the data in all three dimensions when segmenting one 2D slice of the 3D image volume.

A typical way to compress the data from 3D to 2D is by performing a weighted average over the third dimension. Instead of directly compressing the original image volume, we choose to compress the difference images $I^{(j+i)}_{m\times n} - I^{(i)}_{m\times n},\ j \in [-r, \cdots, -1, 1, \cdots, r]$. Here, $2r$ images in the 2D slice neighborhood are selected in the compression phase. A weighted average of these difference images is the condensed slice $\bar{I}^{(i)}_{m\times n}$, and the weights are estimated using the right branch of our MSE block, as described in the next paragraph. As the center image $I^{(j)}_{m\times n}$ changes, the difference images will change accordingly, and their contributions in the estimated condensed image, which are measured by their weights, vary as well. However, once the network is trained, the weights for the difference images are fixed. To make sure the weights are consistently associated with the difference images, we order the difference images based on the sum of their absolute values. That is, the compressed slice $\bar{I}^{(j)}_{m\times n}$ is the weighted average of the ordered difference images, as shown in Figure 8.

To estimate the weights for the ordered difference images, we follow the spatial squeeze and channel excitation idea Hu et al. [8] but with some modifications on channel-wise SE block. Firstly, the original SE block squeezes the spatial domain using the global average pooling. That is, the SE block summarizes each channel using its average over pixels with equal weights. The foreground's pixels are often sparse in the difference images, and nearly-zero pixels dominate in the background. Therefore, we need the flexibility in spatial squeeze and adopt convolution filters to average over pixels with learned weights. A simple way to get a weight for each difference image (or a channel used later) is to use a filter of size $m \times n$. To reduce the number of parameters, we decompose this 2D filter into two 1D filters, i.e., one with size $m \times 1$ and the other with size $n \times 1$, and apply depth-wise convolutions. In this way, we can reduce the number of parameters from mn to $m + n$. Since we expect to summarize the information in each channel independently, we need to reshape the input of each depth-wise convolution accordingly, as shown in Figure 8. As a result, we obtain a vector $z \in R_p$ as the unnormalized weights for difference images. Also, instead of using Sigmoid in [8], which normalizes each weight independently into the range [0, 1], we choose the Softmax function, which counts the correlation among weights and enforces their sum to be 1. The normalized weights are used to rescale the ordered difference images, and then a weighted average results in the compressed slice $\bar{I}^{(i)}_{m\times n}$.

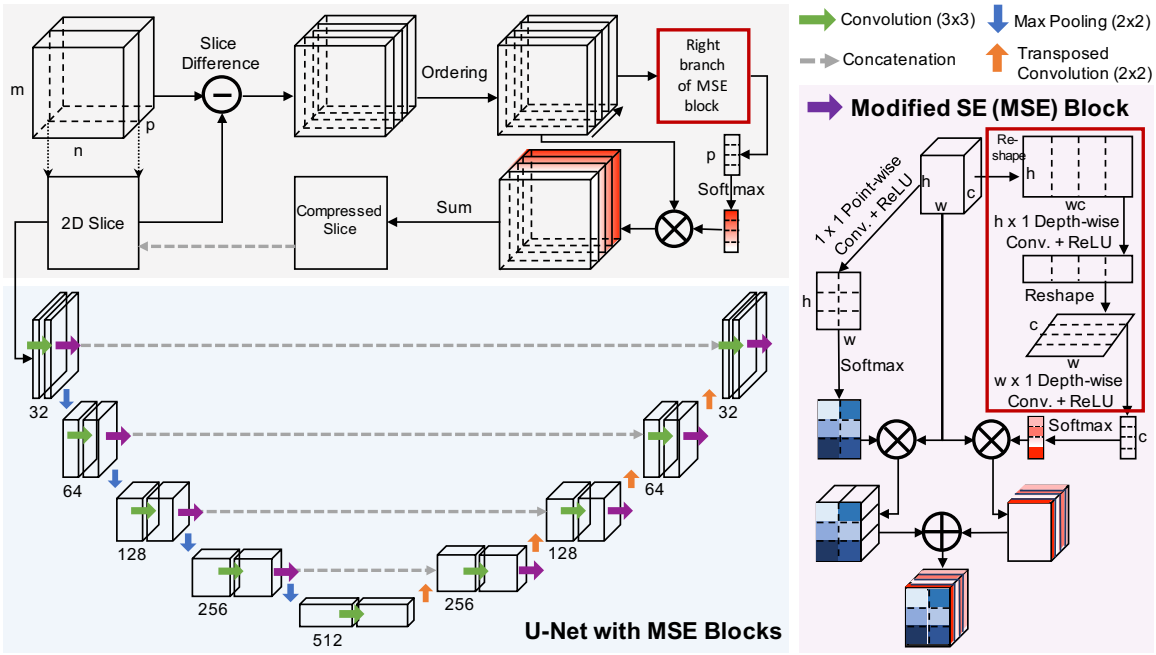

Figure 8: Overview of our proposed multi-dimensional attention network (MDA-Net) for 3D image segmentation. It consists of a third-dimensional compression and an attention-augmented U-Net with modified squeeze and excitation (MSE) blocks.

3.3 MODIFIED SE BLOCK

The Squeeze and Excite block (SE block) [8] performs a squeeze operation that performs global information embedding. Global average pooling is used to aggregate feature maps in their spatial dimension ($H \times W$). This aggregated information is called the channel descriptor. If z is the channel descriptor also referred to as the channel-wise statistic, the c^{th} element of z is the following:

$$z_c = F_{sq}(u_c) = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} u_c(i,j) \tag{1}$$

where F_{sq} represents the squeeze operation, u_c is the c^{th} feature map of $U \in R^{H \times W \times C}$. The global average pooling layer was originally designed to replace the dense layer at the end of a CNN architecture. It outputs one average value for the $H \times W$ dimension of the feature map. [8] mention that this is the simplest averaging technique and can be replaced by something more sophisticated.

We propose a modified way to aggregate the spatial information. Our idea is to break it down into steps where each step captures the aggregate information in one dimension. Instead of the feature map reducing from $H \times W$ to one channel descriptor (which is a single value), it would first reduce to $H \times 1$, getting the average in the W dimension and then into a single value, getting the average in the H dimension.

We further modify the MSE block by making use of the depthwise convolution operation for the Spatial Squeeze and Channel Excitation. This block is explained in detail in Chapter 3.

3.4 DEPTHWISE SEPARABLE CONVOLUTIONS

Depthwise separable convolutions were first introduced by Sifre et al. [27] and were subsequently used in the Inception family of networks proposed by Ioffe et al. [11] and the MobileNet architecture proposed by Howard et al. [7]. This operation factorizes the standard convolution into two steps: the depthwise convolution and the pointwise convolution. The depthwise convolution applies a different filter on each channel of the input and thus outputs a map with the same number of channels. The pointwise operation then combines the channels of this output into one by using a 1×1 convolution. A standard convolution performs the filtering and combining operations in one shot. There is a drastic reduction in computation and model size due to this. The input to the standard convolution is a $D_F \times D_F \times M$ feature map $\mathbf{F}$ as explained in [7] and the output is a $D_G \times D_G \times N$ feature map $\mathbf{G}$. Here, D_F is the height and width of each channel, M is the number of input channels, N is the number of output channels. The convolution kernel K would then be $D_K \times D_K \times M \times N$. The standard convolution thus has a computational cost of:

$$D_F \times D_F \times M \times N \times D_G \times D_G \tag{2}$$

And it can be formalized as follows:

$$G_{k,l,n} = \sum_{i,j,m} K_{i,j,m,n} \cdot F_{k+i-1,l+j-1,m} \tag{3}$$

Thus, the computational cost multiplicatively depends on the number of input channels, number of output channels, the size of the kernel and the size of the feature map. This operation can be factorized into two steps, filtering and combining. The depthwise separable convolution has two steps that respectively perform these two operations.

The depthwise convolutions use one filter per input channel. This can be formalized as:

$$G_{k,l,m} = \sum_{i,j} K_{i,j,m} \cdot F_{k+i-1,l+j-1,m} \tag{4}$$

The overall computation cost of the depthwise separable layer is:

$$D_F \times D_F \times M \times D_G \times D_G + M \times N \times D_F \times D_F \tag{5}$$

3.5 2D ATTENTION-AUGMENTED NET

After compressing the residual information in the third dimension, we represent a 3D volume as a pair of 2D slices, i.e., the current 2D slice $I^{(j)}_{m\times n}$ and the slice-wise compression $\bar{I}^{(j)}_{m\times n}$. A 2D U-Net takes our 2D slice pairs as a two-channel input for segmenting the current 2D image. Using a similar approach in the slice-wise compression, which offers us the third dimension's attention, we add MSE blocks to a plain 2D UNet, which provides both spatial and channel-wise attention on its input and extracted feature maps.

Like the concurrent SE block in [25], we add both channel squeeze and spatial excitation branch (sSE) and spatial squeeze and channel excitation branch (cSE), as shown in Fig. 9. But, different from [25], we use the Softmax operator instead of Sigmoid to normalize the spatial or channel-wise weights in both branches. We consider the attention depends on the space and the channels, and the separated weights allow us to treat spatial pixels or channels differently. In particular, the sSE branch uses a pixel-wise convolution to summarize the input feature maps across channels, which is then normalized over the spatial domain using Softmax to ensure the weights' sum is 1. The cSE branch is the same as introduced in [25]. In particular, given an input feature map $U_{H\times W\times C}$

($H \times W$ is the spatial size of the feature map and C is the number of channels), we apply depth-wise convolution with an $H \times 1$ filter followed by a $W \times 1$ filter. The resulting $C \times 1$ vector is normalized using Softmax to rescale the channels before taking their average. Reshaping the feature maps from 3D to 2D or from 1D to 2D is required before applying the depth-wise convolutions. The addition of these two branches gives us the output of the MSE block. Each MSE block is used after the convolution pair at each resolution of the U-Net.

3.6 SOFTMAX VS SIGMOID

As discussed above, we replace the Sigmoid operator of the original SE block with a Softmax operator. Apart from having the advantage of ensuring the sum of the weights is 1, another advantage this provides is stability during training. The Softmax activation distributes the probability throughout each output node. This makes it easier for the model to converge.

3.7 DROP IN THE NUMBER OF PARAMETERS

The concurrent SE block, along with global average pooling uses fully connected layers to squeeze in the spatial dimension. Fully connected layers perform a linear operation in which every input is connected to every output by a weight. This results in

$$\text{total parameters} = \text{number of inputs} \times \text{number of output weights} \tag{6}$$

This causes significant increases in the total number of trainable parameters in the model. Our MSE block completely abandons the need to use Fully Connected layers since depthwise convolution layers are being used to extract information from the spatial dimension.

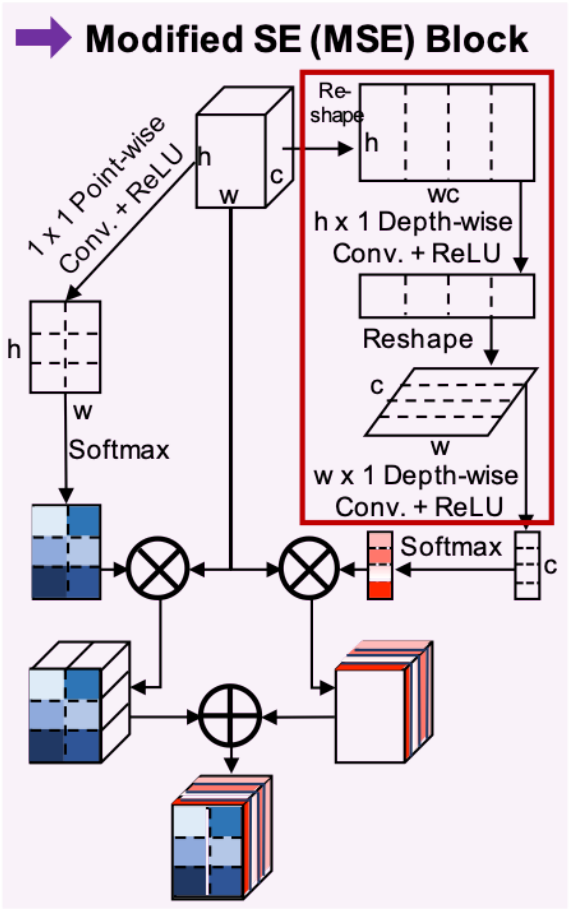

Figure 9: Modified SE Block where the channel squeeze and spatial excite branch is the same as the SE Block but the spatial squeeze and channel excite branch is made more efficient with the help of depthwise convolution layers

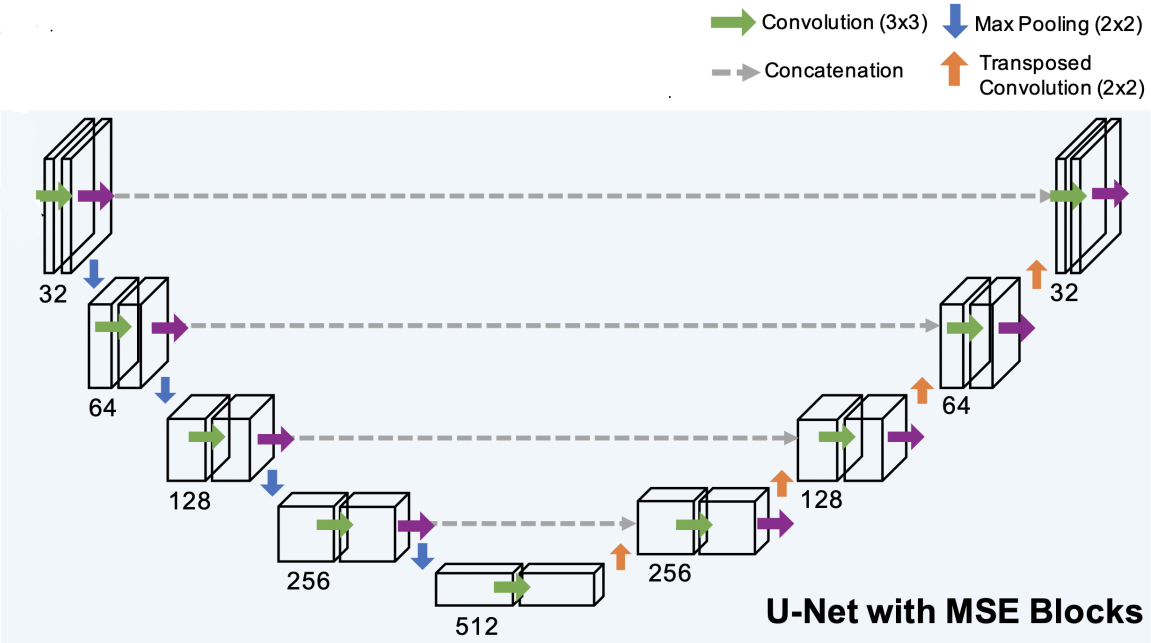

Figure 10: The above shown attention-augmented 2D UNet uses the modified SE block in place of the purple arrows shown above.

CHAPTER 4

EXPERIMENTAL SETUP AND RESULTS

4.1 DATASETS AND EXPERIMENTAL SETTINGS

iSeg Dataset: We evaluate our model on the dataset provided by the MICCAI iSeg challenge published by Wang et al. [30], which aims to segment the brain MRI scans into the cerebrospinal fluid (CSF), white matter (WM), gray matter (GM), and the background. The dataset consists of T1-weighted brain MRI scans collected from 10 6-month infants. Each scan has an image dimension of $144 \times 256 \times 192$ with a spatial resolution of $1 \times 1 \times 1mm^3$.

IBSR Dataset: We use the IBSR dataset published by Rohlfing et al. [23] to further demonstrate the performance improvement of our model over other approaches. The MR images provided in this dataset is a T1-weighted three-dimensional coronal brain scan preprocessed with a positional normalization and a skull-stripping step. It contains 18 Subjects with an image dimension of $512 \times 128 \times 256$ and a spatial resolution of 1.5mm in each dimension.

Ablation Study: The backbone network of our method is the 2D U-Net [24], which is augmented by our MSE blocks. In the ablation study, we compare our MSE-UNet with both the plain UNet and the concurrent scSE-UNet [25](cscSE-UNet) as discussed in Chapter 3. These three models take only a single 2D slice of a 3D image scan to perform the 3D segmentation task in a slice-by-slice manner. Our whole model (MDA-UNet) is also reported to demonstrate the performance gain by including the slice-wise compression component into the MSE-UNet.

We examine three views for both datasets, i.e., Sagittal, Axial, and Coronal, respectively. Take the iSeg dataset for instance, we have 1152 samples (including a 2D slice and a sequence of difference images) in the Sagittal view, 2048 samples in the axial view, and 1536 samples in the coronal view.

The plain U-Net, concurrent scSE-UNet, our modified scSE-UNet and our MDA-Net are compared by performing 5-fold cross validation on each of the three views: Sagittal, Axial and Coronal for each of the 4 above-mentioned models. Tables 4, 5, 6 reports the average over the 5 folds. For the iSeg dataset, the 10 subjects are randomly shuffled and split into training and test sets to perform five-fold cross validation. For the IBSR dataset, the cross-validation is performed in a similar way on the 18 subjects. Table 7, 8, 9 reports the results for IBSR.

OPTIMIZATION. We use Adam optimizer which is a combination of the AdaGrad and RMSProp optimizers. The AdaGrad optimizer performs well on convex optimizations, however, the accumulation of squared gradients that it performs from the start of training can result in premature decrease in the effective learning rate. The RMSProp, as we know, modifies AdaGrad to work well on non-convex optimization by modifying the gradient accumulation into an exponential moving average. The drawback of the RMSProp is that its second order moment estimate may have high bias early in training. The Adam optimizer overcomes this disadvantage and additionally adds momentum to RMSProp. It is particularly efficient on problems with sparse gradients and noisy datasets. We use it with a learning rate of $5e^{-5}$.

REGULARIZATION STRATEGIES. Early stopping is one of the regularization strategies we used here. We know that models with sufficient representational capacity to overfit the task, we often see that at one point, the training error decreases steadily but the validation error begins to rise again. Early stopping helps us combat this by remembering the point where the lowest validation error was recorder and only updates it if there is a further decrease. The maximum number of epochs required using this strategy approximately is 300. Early stopping is one of the most simple and effective regularization methods. One of the common strategies used in machine learning for regularization is bagging and ensemble methods. Dropout is a kind of bagging that is popularly used for neural networks. Dropout generates an ensemble of sub-neuralnets that can be generated by removing non-output units from the base network. Dropout and bagging have one major difference - in bagging, the networks are all independent of each other whereas in dropout the models share parameters and each model inherits a different subset of parameters.

We use a dropout of 0.3 and the L2 regularizer. We perform 5 fold cross-validation. The dice score is used during training and evaluation. All models were trained on one NVIDIA GeForce 1080 8GB GPU. We use the five-fold cross-validation on the subjects for both datasets. We implement our model using Keras and the Tensorflow backend.

4.2 PREPROCESSING

The first step is to get a general idea of the quality of the datasets. This was done by making sure the data didn't have any obvious artifacts (completely black images). We also make sure all the image files are readable so that the training is not interrupted midway. The datasets were in the nifti datatype so we used the nibabel package to convert them into numpy arrays. Next we inspect the image sizes and aspect ratios and get a distribution of APR to classify which of the following 3 categories it belongs to- uniform distribution that is most images are of the same dimension, bimodal distribution that is most images have APR between 0.7 and 1.5 and dataset with a lot of extreme values. We found that both the iSeg and IBSR datasets belong to the first category i.e. uniformly distributed. Hence, no major changes in the dimensions of the images was required. Thirdly, we studied if there was an imbalance in the regions.

The image volume data has pixel intensity values ranging from 0 to 1000. As we know that normalization of the data is important for neural netwroks. This ensures that each input parameter (here image pixels) has a similar data distribution. The training of the network becomes easier and it converges faster once this operation is performed. We use min-max normalization shown by the equation below:

$$\frac{value - min}{max - min} \tag{7}$$

4.3 EXPERIMENTAL SETUP

Since this module works on 2D data, we considered 2D slices of the 3D volumes. We started of with a plain 2D UNet as proposed by Ronnenberger et al.[24] as a baseline. Secondly, we considered a concurrent spatial and channel squeeze and excitation network as introduced by Roy et al.[25] that

	Sagittal (%)		
	CSF	GM	WM
Plain UNet [24]	91.41±1.12	81.54±1.37	75.13±1.19
cscSE-UNet [25]	92.71±1.74	85.56±1.14	79.14±1.11
MSE-UNet (Ours)	**93.11±1.34**	**87.21±1.54**	**81.13±2.34**

Table 1: Segmentation comparison (measured in dice score) among the 2D baselines and the proposed 2D model applied on the Sagittal view of the MICCAI iSeg dataset.

	Axial (%)		
	CSF	GM	WM
Plain UNet [24]	95.22±1.01	92.33±1.18	89.51±2.16
cscSE-UNet [25]	96.47±1.01	92.38±1.18	89.83±2.13
MSE-UNet (Ours)	**96.27±1.01**	**92.62±1.18**	**89.89±2.13**

Table 2: Segmentation comparison (measured in dice score) among the 2D baselines and the proposed 2D model applied on the Axial view of the MICCAI iSeg dataset.

	Coronal (%)		
	CSF	GM	WM
Plain UNet [24]	92.48±2.34	84.36±1.16	80.24±2.35
cscSE-UNet [25]	92.11±1.45	87.12±1.09	82.15±1.42
MSE-UNet (Ours)	**93.91±2.34**	**88.25±2.16**	**84.23±2.65**

Table 3: Segmentation comparison (measured in dice score) among the 2D baselines and the proposed 2D model applied on the Coronal view of MICCAI iSeg dataset.

is the extension of the original squeeze and excitation net introduced by Hu et al.[8]. As introduced above, we compared the results of the above two models our 2D modified squeeze and excite net. We perform experiments on two datasets - the MICCAI Iseg dataset and the IBSR dataset. The details for each of these are provided in the ablation study section of Chapter 4. Along with the dice scores we also compared these models based on the number of parameters and the inference time.

4.4 EXPERIMENTAL RESULTS

Tables 1, 2, 3 compare the dice scores for the CSF, WM and GM for the Sagittal, Axial and Coronal views respectively for the Iseg dataset. It compares all the 2D baseline models - plain UNet, cscSE Net and the 2D attention-augmented net (also called the MSE-Net). The results show that the MSE-Net performs better than the other 2D models in every region. Thus, by performing this comparison on this model that is a submodule of the our MDA-Net, we see that each individual submodule has significance. Similar comparisons were made for the second dataset the results for which can be seen in Tables 7, 8, 9.

	Sagittal (%)		
	CSF	GM	WM
Plain UNet [24]	91.41±1.12	81.54±1.37	75.13±1.19
cscSE-UNet [25]	92.71±1.74	85.56±1.14	79.14±1.11
MSE-UNet (Ours)	93.11±1.34	87.21 ±1.54	81.13 ±2.34
MDA-Net (Ours)	**93.46±1.05**	**87.23±2.64**	**82.79±2.55**

Table 4: Segmentation comparison (measured in dice score) among different approaches applied on the Sagittal view of the MICCAI iSeg dataset.

	Axial (%)		
	CSF	GM	WM
Plain UNet [24]	95.22±1.01	92.33±1.18	89.51±2.16
cscSE-UNet [25]	96.47±1.01	92.38±1.18	89.83±2.13
MSE-UNet (Ours)	96.27 ±1.01	92.62±1.18	89.89 ±2.13
MDA-Net (Ours)	**96.81±1.15**	**94.32±2.13**	**90.01±1.24**

Table 5: Segmentation comparison (measured in dice score) among different approaches applied on the Axial view of the MICCAI iSeg dataset.

	Coronal (%)		
	CSF	GM	WM
Plain UNet [24]	92.48±2.34	84.36±1.16	80.24±2.35
cscSE-UNet [25]	92.11±1.45	87.12±1.09	82.15±1.42
MSE-UNet (Ours)	93.91±2.34	88.25±2.16	84.23 ±2.65
MDA-Net (Ours)	**94.28±2.01**	**88.44±1.45**	**85.23±1.67**

Table 6: Segmentation comparison (measured in dice score) among different approaches applied on the Coronal view of MICCAI iSeg dataset.

Now we add our 2.5D model the MDA-Net to the experiments. Tables 4, 5, 6 report the segmentation results for the MICCAI iSeg dataset. Tables 7, 8, 9 report the segmentation results for the IBSR dataset. The brain segmentation performance for both datasets is steadily improved from plain U-Net, to cscSE-UNet, to MSE-UNet, and then to MDA-Net for each image view. The improved performance of MSE-UNet over the cscSE-UNet demonstrates the effectiveness of using MSE-Blocks in the U-Net, and the improved performance of MDA-Net over the MSE-UNet demonstrates the effectiveness of using the slice-wise compression. We also observed that replacing Sigmoid with Softmax makes the network training more stable and easier to converge. Figure 11 shows qualitative results sampled from the iSeg dataset. Figure 12 shows the graphical representation of the dice score comparisons for iSeg and Figure 13 for IBSR.

Table 10 reports the computational cost of the above four models. Compared to cscSE-UNet, our models have a reduced number of parameters and a reduced amount of training time but slightly increased inference time. The graphical representation of this comparisons are shown in

tables 14, 15, 16. Our proposed model's maximum training time was around six hours, and the inference time is within seconds for a subject. We tried experiments with a 3D U-Net and a convolutional LSTM; however, we met memory difficulties on our machine, which drives MDA-Net's development.

	Sagittal (%)		
	CSF	GM	WM
Plain UNet [24]	90.81±1.85	95.91±2.24	92.92±1.55
cscSE-UNet [25]	92.41±1.15	96.06±2.14	93.61±2.55
MSE-UNet (Ours)	94.01±1.95	96.96±2.14	94.79±3.55
MDA-Net (Ours)	**95.12±1.00**	**97.27±2.14**	**95.04±1.15**

Table 7: Segmentation comparison (measured in dice score) among different approaches applied on the Sagittal view of the IBSR dataset.

	Axial (%)		
	CSF	GM	WM
Plain UNet [24]	91.24±1.15	90.33±2.16	88.38±1.84
cscSE-UNet [25]	91.42±2.15	91.17±2.10	88.81±1.04
MSE-UNet (Ours)	92.23±1.25	92.65±2.14	89.82±1.64
MDA-Net (Ours)	**94.89±1.05**	**95.31±2.73**	**92.01±2.24**

Table 8: Segmentation comparison (measured in dice score) among different approaches applied on the Axial view of the IBSR dataset.

	Coronal (%)		
	CSF	GM	WM
Plain UNet [24]	91.45±2.08	87.30±1.46	89.22±2.67
cscSE-UNet [25]	92.15±1.01	88.12±1.95	89.82±1.27
MSE-UNet (Ours)	93.93±2.01	89.21±1.15	90.58±1.60
MDA-Net (Ours)	**94.52±2.31**	**91.45±1.35**	**92.22±1.67**

Table 9: Segmentation comparison (measured in dice score) among different approaches applied on the Coronal view of the IBSR dataset.

	#Param.	Training time (per epoch)	Inference time (per patient)
Plain UNet	7.775M	30s	1.565s
cscSE-UNet	7.958M	59s	1.575s
MSE-UNet	7,823M	57s	2.285s
MDA-Net	7.825M	58s	2.795s

Table 10: Model comparison on the number of model parameters and the training and testing time for the iSeg dataset.

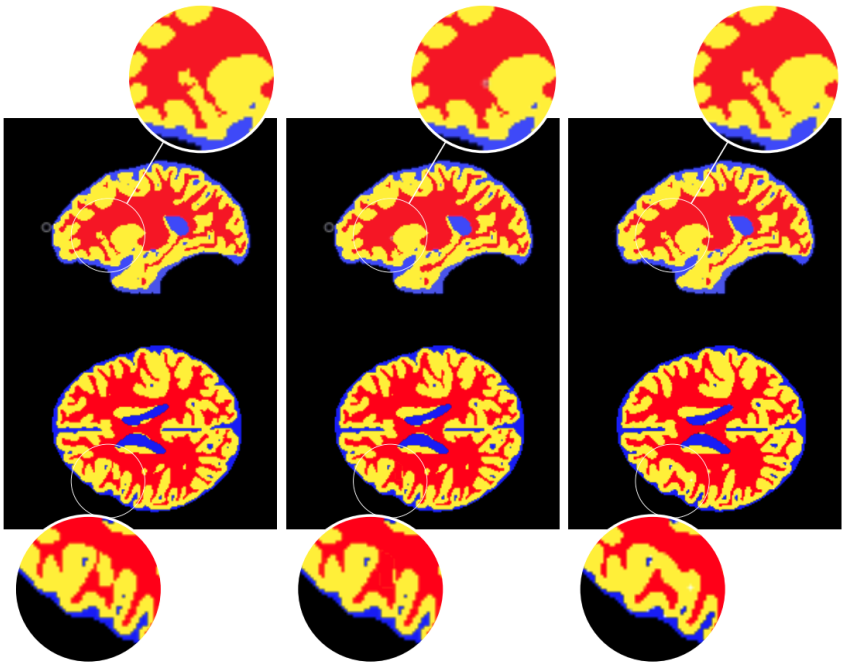

Figure 11: Qualitative iSeg sample results (from left to right: Ground-Truth, cscSE-UNet, and our MDA-Net. **Blue**: cerebrospinal fluid, **yellow**: gray matter, **red**: white matter.

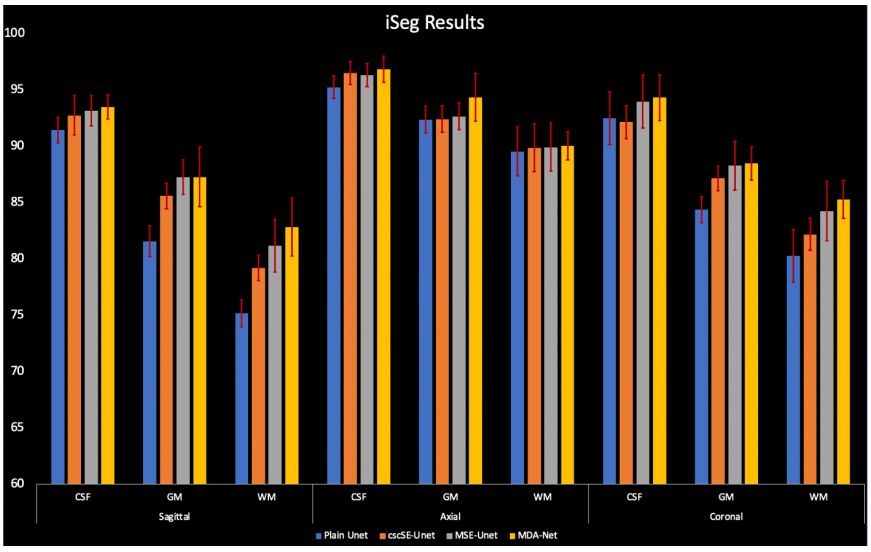

Figure 12: This graph depicts that our models outperform the others in all 3 views of the iSeg dataset. The y-axis shows the dice scores in percentage and the x-axis shows the scores for the 3 regions CSF, GM and WM of the 3 views Sagittal, Axial and Coronal along with the standard deviation.

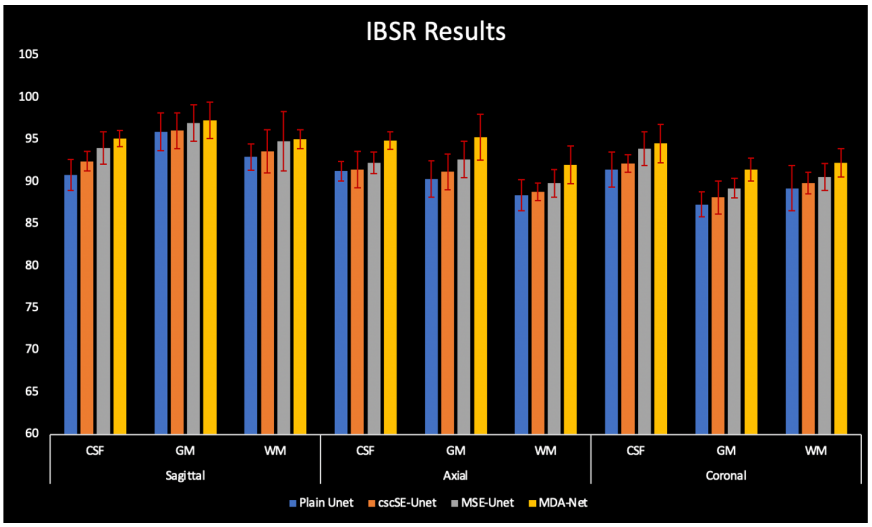

Figure 13: This graph depicts that our models outperform the others in all 3 views of the IBSR dataset. The y-axis shows the dice scores in percentage and the x-axis shows the scores for the 3 regions CSF, GM and WM of the 3 views Sagittal, Axial and Coronal alongwith the standard deviation.

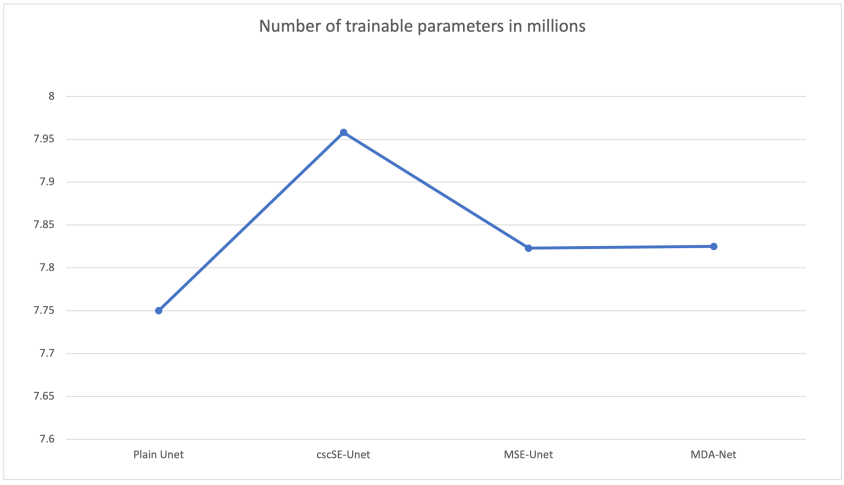

Figure 14: This graph depicts that our 2.5D model has number of trainable parameters comparable to the Plain Unet which is a pure 2D model and does not use data from the 3rd dimension. The y-axis shows number of parameters in millions.

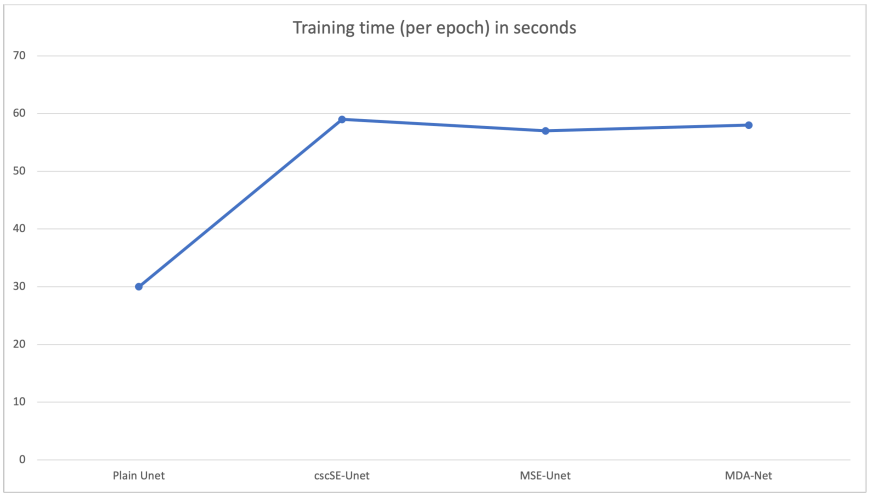

Figure 15: This graph depicts that our models have comparable training times to the baseline 2D models

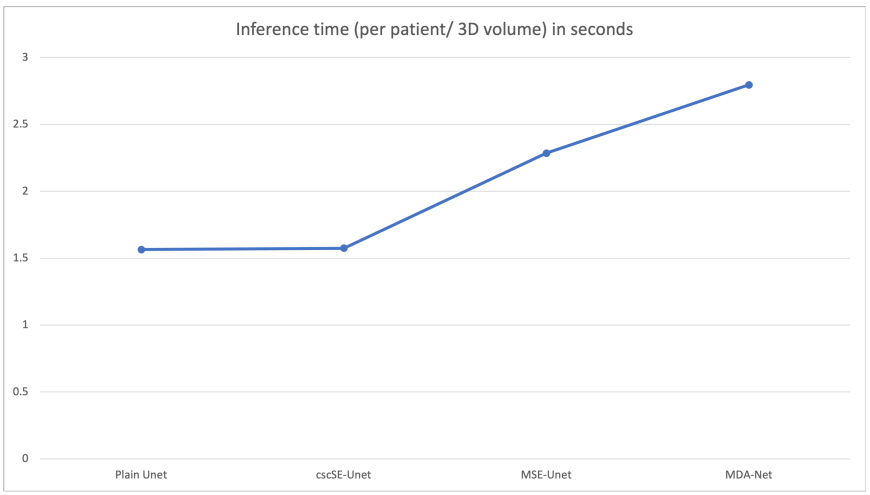

Figure 16: This graph depicts that our models have comparable inference times to the baseline 2D models

CHAPTER 5

CONCLUSION & FUTURE DIRECTIONS

This paper investigated the 3D image segmentation problem and proposed an efficient solution using a multi-dimensional attention network. The objective of the thesis was to reduce the computational resources required to perform 3D semantic segmentation while at the same time using information available from all three dimensions.

To this end, we firstly introduce a difference and order operation that, for every slice of the 3D volume, summarizes information from r of its neighbors to efficiently take advantage of the 3rd dimension as well. The r here was chosen as 5 experimentally. This operation provides an independent difference and ordered volume for each slice respectively. These volumes are compressed as described in the following paragraphs.

Secondly, we designed a modified version of the concurrent spatial and channel squeeze and excite block (MSE block) which replaces global average pooling and fully connected Layers with depthwise convolution layers to abstract information from the spatial dimension with a reduction in the number of parameters. The global average pooling layer summarizes the entire spatial information using an average over pixels with equal weights. Since the output volumes obtained have sparse non-zero values, the convolution layers provide flexibility while summarizing in the spatial dimension and an average over pixels is taken with learned weights. This modified spatial squeeze and channel excite branch of the block is used to compress the volumes obtained from the difference and order operation and create a 2D slice that is associated with the 2D slice that was used to obtain the volume.

Thirdly, we design an attention-augmented 2D UNet that takes 2D slices of the 3D volume as input. The UNet is called attention-augmented because each convolution block is followed by the entire MSE block. At this point, MSE simultaneously performs spatial and channel squeeze and excite and learns weights for spatial dimension and channels of the outputs obtained from the convolution layers. Performing this kind of weight reassignment allows the model to focus on

regions of importance thus leading to a drop in the computational resources required for training. We perform comparisons of this 2D submodule of our module MSE-Net with the plain UNet and the csSE-Net as baselines. The region-wise dice scores, number of trainable parameters and inference time are used to make these comparisons. The conclusion here is that the MSE blocks cause a reduction in the number of parameters and an increase in the dice scores of each of the individual regions.

Finally, we put all our modules together to form the ultimate model - the MDA-Net. Here, the input to the MSE-Net discussed above is now a set of two 2D slices, one the original slice and the other its differenced, ordered, and compressed counterpart that summarizes the information in its neighborhood. We continue our comparison now including this MDA-Net to it. Our observation is that even after adding the extra contextual information, this model still has lesser trainable parameters compared to its 2D counterparts mentioned above. Thus we achieve our objective of using the 3rd dimension efficiently.

We perform our experiments on two 3D image volume datasets namely the MICCAI iSeg and IBSR. Each of them has three regions and our models outperform others in the individual dice scores of all three regions of both datasets.

A limitation worth noting is that there is an added hyperparameter (r) as discussed above which stands for the number of neighbors before and after the current slice that should be taken into consideration while forming the compressed slice with information from the 3rd dimension. We found out experimentally that r = 5 works best in our case. Taking lesser r values reduced the dice score and increasing it did not add significant value. However, for different datasets, this value could change and needs to be decided experimentally keeping in mind the trade-off between the computational complexity and evaluation metrics.

We tested the network on image volumes; however, it could be extended to handle spatiotemporal data like videos or longitudinal images. We will apply our model to other segmentation tasks with different image types in the future. For multi-modality image scans, we could explore image-wise attention, that is, measuring the contributions of each modality to the segmentation task.

REFERENCES

[1] Sean Bell et al. "Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 2874–2883.

[2] Robin Brügger, Christian F Baumgartner, and Ender Konukoglu. "A partially reversible u-net for memory-efficient volumetric image segmentation". In: *International conference on medical image computing and computer-assisted intervention*. Springer. 2019, pp. 429–437.

[3] Özgün Çiçek et al. "3D U-Net: learning dense volumetric segmentation from sparse annotation". In: *International conference on medical image computing and computer-assisted intervention*. Springer. 2016, pp. 424–432.

[4] Grzegorz Chlebus et al. "Automatic liver tumor segmentation in CT with fully convolutional neural networks and object-based postprocessing". In: *Scientific reports* 8.1 (2018), pp. 1–7.

[5] Pierrick Coupé et al. "Patch-based segmentation using expert priors: Application to hippocampus and ventricle segmentation". In: *NeuroImage* 54.2 (2011), pp. 940–954.

[6] Raunak Dey and Yi Hong. "Hybrid Cascaded Neural Network for Liver Lesion Segmentation". In: *arXiv preprint arXiv:1909.04797* (2019).

[7] Andrew G Howard et al. "Mobilenets: Efficient convolutional neural networks for mobile vision applications". In: *arXiv preprint arXiv:1704.04861* (2017).

[8] Jie Hu, Li Shen, and Gang Sun. "Squeeze-and-excitation networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 7132–7141.

[9] Forrest Iandola et al. "Densenet: Implementing efficient convnet descriptor pyramids". In: *arXiv preprint arXiv:1404.1869* (2014).

[10] Forrest N Iandola et al. "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and¡ 0.5 MB model size". In: *arXiv preprint arXiv:1602.07360* (2016).

[11] Sergey Ioffe and Christian Szegedy. “Batch normalization: Accelerating deep network training by reducing internal covariate shift”. In: *arXiv preprint arXiv:1502.03167* (2015).

[12] Laurent Itti, Christof Koch, and Ernst Niebur. “A model of saliency-based visual attention for rapid scene analysis”. In: *IEEE Transactions on pattern analysis and machine intelligence* 20.11 (1998), pp. 1254–1259.

[13] Laurent Itti and Christof Koch. “Computational modelling of visual attention”. In: *Nature reviews neuroscience* 2.3 (2001), pp. 194–203.

[14] Oleksii Kuchaiev and Boris Ginsburg. “Factorization tricks for LSTM networks”. In: *arXiv preprint arXiv:1703.10722* (2017).

[15] Hugo Larochelle and Geoffrey E Hinton. “Learning to combine foveal glimpses with a third-order boltzmann machine”. In: *Advances in neural information processing systems*. 2010, pp. 1243–1251.

[16] Zhijian Liu et al. “Point-voxel cnn for efficient 3d deep learning”. In: *arXiv preprint arXiv:1907.03739* (2019).

[17] Jonathan Long, Evan Shelhamer, and Trevor Darrell. “Fully convolutional networks for semantic segmentation”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 3431–3440.

[18] Christian Lucas et al. “Multi-scale neural network for automatic segmentation of ischemic strokes on acute perfusion images”. In: *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*. IEEE. 2018, pp. 1118–1121.

[19] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. “V-net: Fully convolutional neural networks for volumetric medical image segmentation”. In: *2016 Fourth International Conference on 3D Vision (3DV)*. IEEE. 2016, pp. 565–571.

[20] Volodymyr Mnih, Nicolas Heess, Alex Graves, et al. “Recurrent models of visual attention”. In: *Advances in neural information processing systems* 27 (2014), pp. 2204–2212.

[21] Alejandro Newell, Kaiyu Yang, and Jia Deng. “Stacked hourglass networks for human pose estimation”. In: *European conference on computer vision*. Springer. 2016, pp. 483–499.

[22] Bruno A Olshausen, Charles H Anderson, and David C Van Essen. "A neurobiological model of visual attention and invariant pattern recognition based on dynamic routing of information". In: *Journal of Neuroscience* 13.11 (1993), pp. 4700–4719.

[23] Torsten Rohlfing. "Image similarity and tissue overlaps as surrogates for image registration accuracy: widely used but unreliable". In: *IEEE transactions on medical imaging* 31.2 (2011), pp. 153–163.

[24] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation". In: *International Conference on Medical image computing and computer-assisted intervention*. Springer. 2015, pp. 234–241.

[25] Abhijit Guha Roy, Nassir Navab, and Christian Wachinger. "Concurrent spatial and channel 'squeeze & excitation'in fully convolutional networks". In: *International conference on medical image computing and computer-assisted intervention*. Springer. 2018, pp. 421–429.

[26] Noam Shazeer et al. "Outrageously large neural networks: The sparsely-gated mixture-of-experts layer". In: *arXiv preprint arXiv:1701.06538* (2017).

[27] Laurent Sifre and Stéphane Mallat. "Rigid-motion scattering for image classification". In: *Ph. D. thesis* (2014).

[28] Christian Szegedy et al. "Going Deeper With Convolutions". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2015.

[29] Ashish Vaswani et al. "Attention is all you need". In: *Advances in neural information processing systems*. 2017, pp. 5998–6008.

[30] Li Wang et al. "Benchmark on automatic six-month-old infant brain segmentation algorithms: the iSeg-2017 challenge". In: *IEEE transactions on medical imaging* 38.9 (2019), pp. 2219–2230.

[31] Xudong Wang et al. "Volumetric attention for 3D medical image segmentation and detection". In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer. 2019, pp. 175–184.

APPENDIX A

MODEL HYPERPARAMETERS

A.1 Plain UNet2D

- Learning rate

  $5e^{-5}$ with Adam optimizer

- Epochs

  300 with early stopping

- Hidden Layers

  4 downsampling (DS) UNnet blocks, 4 upsampling and 1 bottom layer. Number of units in convolution layers: 32, 64, 128, 256, 512, 256, 128, 32

- Activation function

  ReLU

- Dropout

  Dropout rate: 0.3

- Network initialization

  Weight initializer for convolution layers: *'glorot uniform'* Bias initializer for convolution layers: *'zeros'*

- Momentum

  No momentum added explicitly. The Adam optimizer adds momentum by default.

- Kernel regularizer

  Regularizer for convolution layers: l2 with $\lambda = 0.0002$

A.2 Concurrent Spatial and Channel Squeeze and Excite Net

- Learning rate

  $5e^{-5}$ with Adam optimizer

- Epochs

  300 with early stopping

- Hidden Layers

  4 downsampling (DS) UNnet blocks, 4 upsampling and 1 bottom layer. Number of units in convolution layers: 32, 64, 128, 256, 512, 256, 128, 32

- Activation function

  ReLU for first convolution layer of each block and Sigmoid for second

- Dropout

  Dropout rate: 0.3

- Network initialization

  Weight initializer for convolution layers: *'glorot uniform'* Bias initializer for convolution layers: *'zeros'*

- Momentum

  No momentum added explicitly. The Adam optimizer adds momentum by default.

- Kernel regularizer

  Regularizer for convolution layers: l2 with $\lambda = 0.0002$

- SE block

  Reduction ratio $r = 2$

A.3 Attention-Augmented 2D UNet

- Learning rate

  $5e^{-5}$ with Adam optimizer

- Epochs

  300 with early stopping

- Hidden Layers

  4 downsampling (DS) UNnet blocks, 4 upsampling and 1 bottom layer. Number of units in convolution layers: 32, 64, 128, 256, 512, 256, 128, 32

- Activation function

  ReLU for first convolution layer of each block and Softmax for second

- Dropout

  Dropout rate: 0.3

- Network initialization

  Weight initializer for convolution layers: *'glorot uniform'* Bias initializer for convolution layers: *'zeros'*

- Momentum

  No momentum added explicitly. The Adam optimizer adds momentum by default.

- Kernel regularizer

  Regularizer for convolution layers: l2 with $\lambda = 0.0002$

A.4 MDA-Net

- Learning rate

  $5e^{-5}$ with Adam optimizer

- Epochs

  300 with early stopping

- Hidden Layers

  4 downsampling (DS) UNnet blocks, 4 upsampling and 1 bottom layer. Number of units in convolution layers: 32, 64, 128, 256, 512, 256, 128, 32

- Activation function

  ReLU for first convolution layer of each block and Softmax for second

- Dropout

  Dropout rate: 0.3

- Network initialization

  Weight initializer for convolution layers: *'glorot uniform'* Bias initializer for convolution layers: *'zeros'*

- Momentum

  No momentum added explicitly. The Adam optimizer adds momentum by default.

- Kernel regularizer

  Regularizer for convolution layers: l2 with $\lambda = 0.0002$

- Difference and order block

  Number of neighbors r = ± 5