CEREBELLUM PARCELLATION FROM MAGNETIC RESONANCE IMAGING USING DEEP LEARNING

by
Shuo Han

A dissertation submitted to Johns Hopkins University in conformity with the requirements for the degree of Doctor of Philosophy

Baltimore, Maryland
March, 2022

Abstract

The human cerebellum plays an important role in both motor and cognitive functions, and these functions have a topological mapping within the cerebellum. It is possible to use structural magnetic resonance imaging (MRI) to study the cerebellum since it is a non-invasive modality and provides good soft-tissue contrast. Deep learning (DL) techniques have been recently used to process medical images with great success. In this dissertation, we focus on developing DL algorithms to automatically parcellate the cerebellum—i.e., to divide the cerebellum into its sub-regions—from MRI images with both accuracy and efficiency in mind. With these algorithms, we can then study the morphological properties of cerebellar sub-regions to better understand the cerebellum.

First, we developed ACAPULCO, a cerebellum parcellation algorithm based on convolutional neural networks (CNNs). It is the first DL algorithm that outperforms conventional methods, and it is being used around the world. We also experimented with incorporating anatomical knowledge into the network design as a potential improvement to ACAPULCO.

Second, we parcellated over 2,000 T1-weighted MRI images using ACAPULCO to study the changes of the cerebellum during normal aging. We performed linear mixed-effect regressions of these sub-regional volumes to estimate their longitudinal trajectories. Our study is one step forward to better understand the cerebellum.

Finally, we studied DL-based super-resolution (SR) to improve the quality of MRI images for better cerebellum parcellation. We proposed ESPRESO, an algorithm using a modified generative adversarial network to estimate the slice profiles of 2D multi-slice MRI images to measure their resolutions. We then improved an internally supervised SR algorithm and equipped it with ESPRESO for better SR performance. We further showed that ACAPULCO could be improved by taking super-resolved T2-weighted MRI images as input.

Dissertation Committee:
Dr. Jerry L. Prince (first reader and advisor)
Dr. Michael Schär (secondary reader)
Dr. René Vidal

Acknowledgements

I would like to thank my advisor, Dr. Jerry L. Prince, for his guidance and support. I learned how to be a researcher from him. His knowledge, insight, and passion are always inspiring. I would like to thank Dr. René Vidal and Dr. Michael Schär for being on my dissertation committee and their help in this journey.

I would like to thank my colleagues and friends from the Image Analysis and Communications Laboratory at the Johns Hopkins University. I would like to thank Yufan He and Muhan Shao for their consistent support. They are one of the main reasons that I have made it this far. I would like to thank Aaron Carass for his help with both my life and study. I have learned a lot from Blake E. Dewey. The curiosity and passion from Yihao Liu and Lianrui Zuo always motivate me. I would like to thank Zhao Can, Xue Yuan, Samuel W. Remedios, Ahmed Alshareef, and other colleagues for their help. I would like to thank Dr. Daniel A. Herzka and all other professors in the Johns Hopkins University. I would also like to thank Dr. Susan M. Resnick, Yang An, Murat Bilgel, and other colleagues from the National Institute on Aging.

I would like to thank my friends, my family, and my country. Without them, I would not be who I am today.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

The human cerebellum sits at the lower back of the head (see Fig. 1-1(A)). Although it only takes up 10% of the brain volume, the cerebellum has about half of the total neurons in the brain [1]. Previous studies have shown its associations with both motor and cognitive functions [2–4]. Instead of directly controlling, the cerebellum seems to be more involved in modulating and coordinating these functions. For example, the cerebellum contributes to calibration of eye movement and reducing eye instability [2]. Patients with cerebellar lesions exhibit undershooting or overshooting when performing voluntary limb movements [2] and affective dyscontrol [4].

The cerebellum can be grossly divided into three lobes [5]—the anterior lobe, the posterior lobe, and the flocculonodular lobe—according its primary fissure and postero-lateral fissure. From medial to lateral, the cerebellum can also be divided into the vermis, the intermediate zone (paravermis), and the lateral hemispheres. These regions can be further divided into smaller parts called lobules (see Section 1.2 for more details). Previous studies have shown that different functions can correspond to different parts of the cerebellum [4, 6–8]. For example, the vermis and intermediate zone are involved in eye movement and gait, while the posterior lobe and lateral hemispheres are more

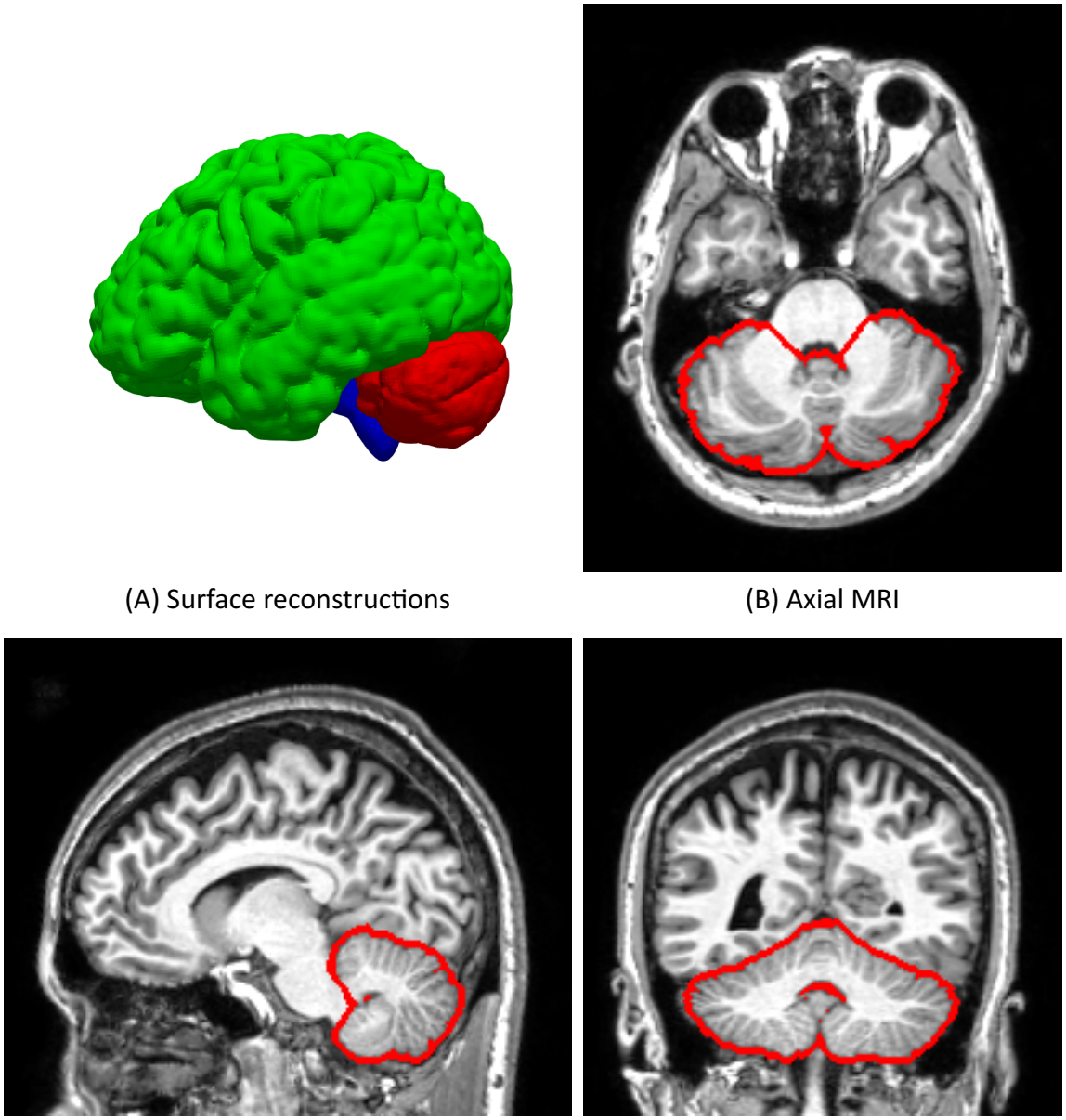
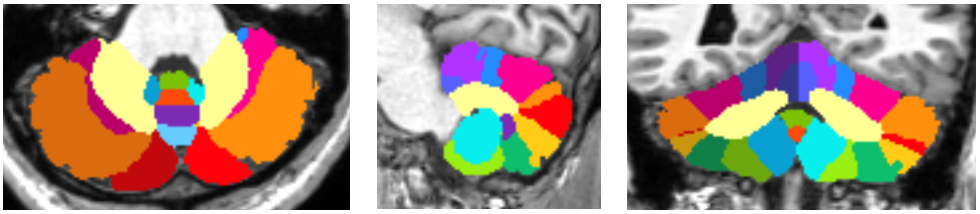


Figure 1-1. Visualization of the cerebellum. (A) shows surface reconstructions of the cerebrum, cerebellum, and brain stem in green, red, and blue, respectively. (B), (C), and (D) are axial, sagittal, and coronal slices from an MRI image, respectively. The cerebellum is outlined in red.

involved in higher order cognitive functions [2–4]. Therefore, it is important to be able to study the cerebellum with respect to its sub-regions.



(A) Axial (B) Sagittal (C) Coronal

Figure 1-2. Example parcellation of the cerebellum from an MRI image. Cerebellar sub-regions are labeled with different colors. (A), (B), and (C) are an axial slice, a sagittal slice, and a coronal slice, respectively.

Magnetic resonance imaging (MRI) is a non-invasive modality which can provide *in vivo* images of the cerebellum (see Fig. 1-1(B)–(D)[1]). Functional MRI has been used to show the different activity patterns of the cerebellum during different tasks [3, 10]. Structural MRI has been used to correlate cerebellar sub-regional changes with aging, biological sex, and function assessments in both normal subjects [11–13] and subjects with diseases [14–16]. In this dissertation, we are interested in analyzing structural MRI images of the cerebellum. Particularly, we are interested in a type of image processing technique called image parcellation. It is analogous to semantic segmentation in natural image processing, and when used in medical images, especially brain images, it means to divide a region of interest into its sub-regions. Cerebellum parcellation using MRI images dates back to the work by Schmahmann *et al.* [5] which constructed a single atlas of the cerebellum to relate image features to its division. Instead of a single image, Diedrichsen [17] averaged 20 images to build a spatially unbiased atlas template of the cerebellum. Following their work, several datasets of multiple manual delineations are now available [18, 19]. However, manual delineations require expertise and are very time-consuming to generate. To permit large-scale studies, automatic algorithms to parcellate the cerebellum are desirable, and this is the main of focus of this dissertation.

[1]The MRI images are from https://www.nitrc.org/projects/multimodal [9].

See Fig. 1-2 for an example parcellation. A review of previous automatic cerebellum parcellation algorithms can be found in Section 1.3.

1.1 MRI Images and Processing

The contrast of an MRI image mainly depends on the inherent properties of the tissues being imaged and the MRI sequence. The tissue properties that mostly affect the contrast are the longitudinal relaxation time (T1), the transverse relaxation time (T2), and the proton density (PD), and they are different among different tissue types. Different MRI sequences can emphasize different aspects of these properties. For example, the commonly used magnetization-prepared rapid acquisition with gradient echo (MPRAGE) sequence [20] can be regarded as a T1-weighted (T1w) sequence, meaning that its contrast is primarily determined by the T1 values. The turbo spin echo (TSE) sequence [21] is usually PD-weighted (PDw) or T2-weighted (T2w), meaning that its contrast is primarily determined by the PD or T2 values, respectively. In addition to the sequence types, specific parameters of these sequences, such as the repetition time (TR), echo time (TE), and flip angle, can also affect the contrast. For example, the TE should be short when acquiring PDw images, while it should be roughly equal to the T2 values of the tissues being imaged when acquiring T2w images. Due to these flexibilities, it is a common practice in clinics and research to acquire multiple images with different contrasts to reveal different aspects of the region of interest (ROI). Table 1-I summarizes typical intensity levels of brain tissues in T1w and T2w images, and Fig. 1-3[1] shows an example of these two contrasts of the same subject at the cerebellum. More information on MRI contrasts can be found in Liang & Lauterbur [23], Bernstein *et al.* [24], and Prince & Links [25].

[1] The MRI images are from the OASIS-3 dataset https://www.oasis-brains.org/ [22].

Table 1-I. Intensity levels of brain tissues in T1w and T2w images.

	T1w	**T2w**
Gray matter	Medium	Low
White matter	High	Low
Cerebrospinal fluid	Low	High

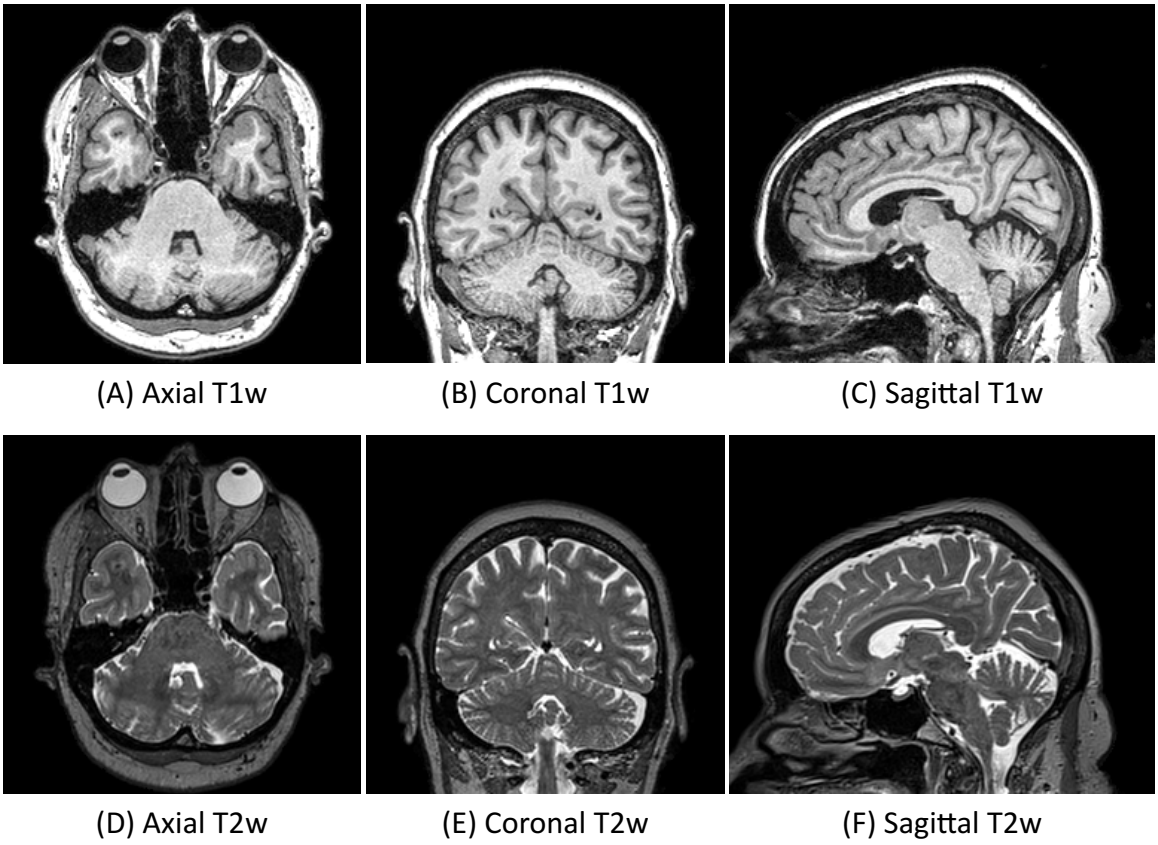


Figure 1-3. Example T1w and T2w images of the same subject at the cerebellum. (A), (B), and (C) are axial, coronal, and sagittal slices of a T1w image, respectively; (D), (E), and (F) are axial, coronal, and sagittal slices of a T2w image, respectively.

Some processing routines are regularly applied to MRI images. Usually, the first step is to correct the intensity inhomogeneity in an MRI image (see Fig. 1-4). Intensity inhomogeneity is also called the bias field or gain field. It is mainly caused by non-uniformity of the radio-frequency (RF) coils (the devices that excite and receive the signals in

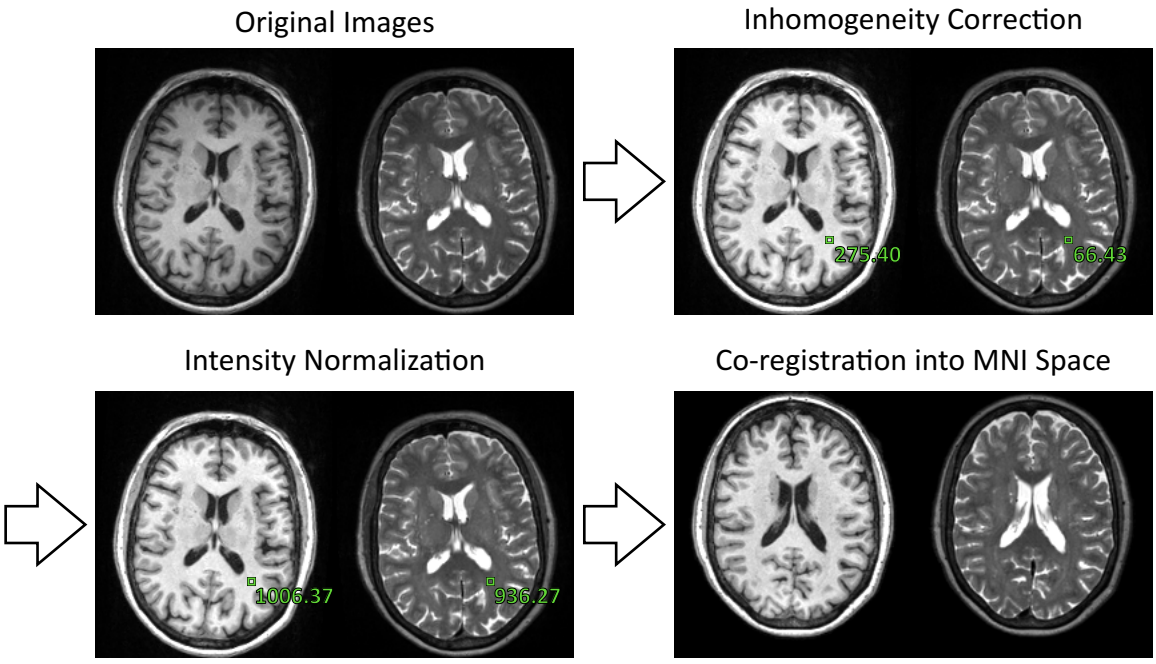


Figure 1-4. Illustration of common pre-processing steps. Note that the image intensities are scaled for display purposes. The intensity values within the green squares are shown in green font before and after intensity normalization to show its effect.

MRI) and other factors such as gradient eddy-currents [26]. Intensity inhomogeneity appears as a spatially slowly-varying multiplicative field in almost every MRI image, causing the same tissue to have different intensities (see Fig. 1-4). We use the N4 algorithm [27] to correct intensity inhomogeneity throughout the whole dissertation for its accuracy and fast computation. Unlike a computed tomography (CT) image, an MRI image typically does not have a fixed scale for its intensities, and it varies even within the same scanner, imaging protocol, and patient [28]. In brain images, it is a common practice to apply a linear transform to the image intensities, so the white matter (WM) can have a relatively fixed intensity value among different images [28–30]. An example of this intensity normalization is included in Fig. 1-4, where the mean intensity of WM is normalized to 1,000. The same patient's MRI images that are acquired in different imaging sessions or even within the same imaging session usually do not align up due to

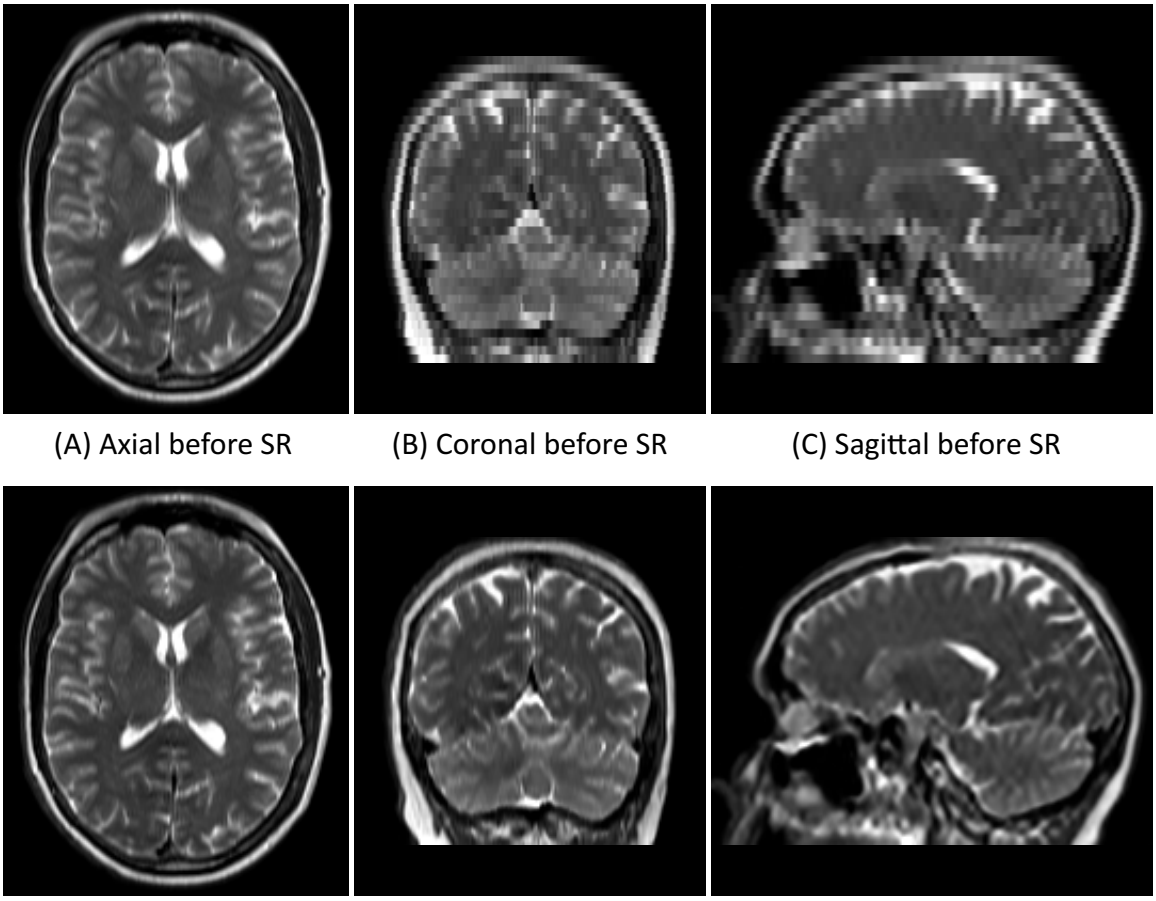


Figure 1-5. Example super-resolution (SR). (A), (B), and (C) show axial, coronal, and sagittal slices of an image before SR, respectively; (D), (E), and (F) show axial, coronal, and sagittal slices of an image after SR, respectively. The low-resolution images are shown with nearest neighbor interpolation for display purposes.

calibration of the scanner, settings of the imaging protocol, and the patient's movement, etc. Therefore, it is a common practice to rigidly co-register them together; we also rigidly register them onto the same template image in a specific coordinate space, so different images can have a standardized orientation. Throughout the whole dissertation, we use the MNI space [31, 32] as this coordinate space, which is defined by a brain atlas averaged from multiple images[1]. See Fig. 1-4 for an example of this image registration.

[1]See https://www.mcgill.ca/bic/icbm152-152-nonlinear-atlases-version-2009.

In addition to these aforementioned processing steps, there are some optional processings that can be done before image parcellation. For example, if an MRI image has anisotropic resolutions along different axes, which is typically the case for a 2D multi-slice acquisition (see Section 5.1.1 for more details of this acquisition), we can use a technique called super-resolution (SR) to enhance its low-resolution (LR) axis. See Fig. 1-5 for an example. Other processing can include brain extraction [33] (or called skull stripping in some literature) and lesion filling [34].

1.2 Cerebellar Sub-Regions

The definition and naming of cerebellar sub-regions in this dissertation are based on the atlas proposed by Schmahmann *et al.* [5] and two public cerebellum parcellation datasets, which are named the T dataset and the M dataset, from Carass *et al.* [19]. The delineations of these two datasets generally follow the regions in Schmahmann *et al.* [5] but merge some of them differently. In both datasets, the corpus medullare is delineated as the main body of the cerebellar WM. The rest of the cerebellum—i.e., the cerebellar GM and the cerebellar WM branches that are outside the corpus medullare—is further divided into multiple regions. For simplicity, we use "cerebellar GM" to refer to both the GM and these WM branches in this dissertation. From medial to lateral, the cerebellar GM is divided into the vermis and hemispheres (see Fig. 1-6) (see a discussion of the paravermis in Schmahmann *et al.* [5]). According to the primary fissure and the posterolateral fissure of the cerebellum, the GM can be divided into the anterior lobe, the posterior lobe, and the flocculonodular lobe (see Fig. 1-7). The vermal and the hemispheric lobes are further divided into lobules that are named according to the Schmahmann nomenclature [5].

In the anterior lobe, the vermis is divided into vermal lobules I–II, III, IV, and V. As

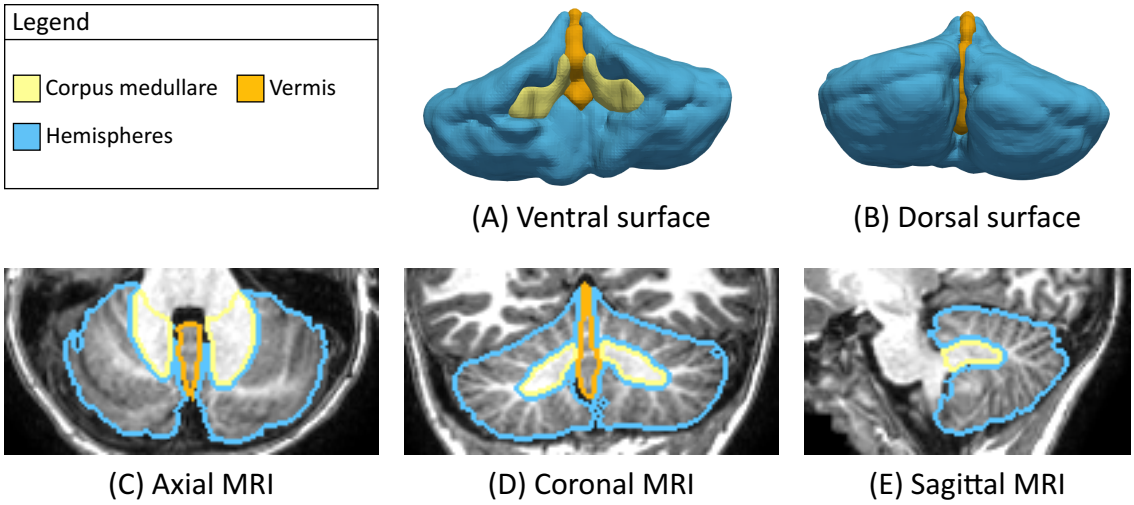


Figure 1-6. The cerebellum can be divided into the corpus medullare, the vermis, and the hemispheres. (A) and (B) show example cerebellar surfaces from the ventral and dorsal views, respectively. (C), (D), and (E) show axial, coronal, sagittal slices of an MRI image, respectively. These regions are shown or outlined in their corresponding colors that are shown in the legend. The image is from the M dataset.

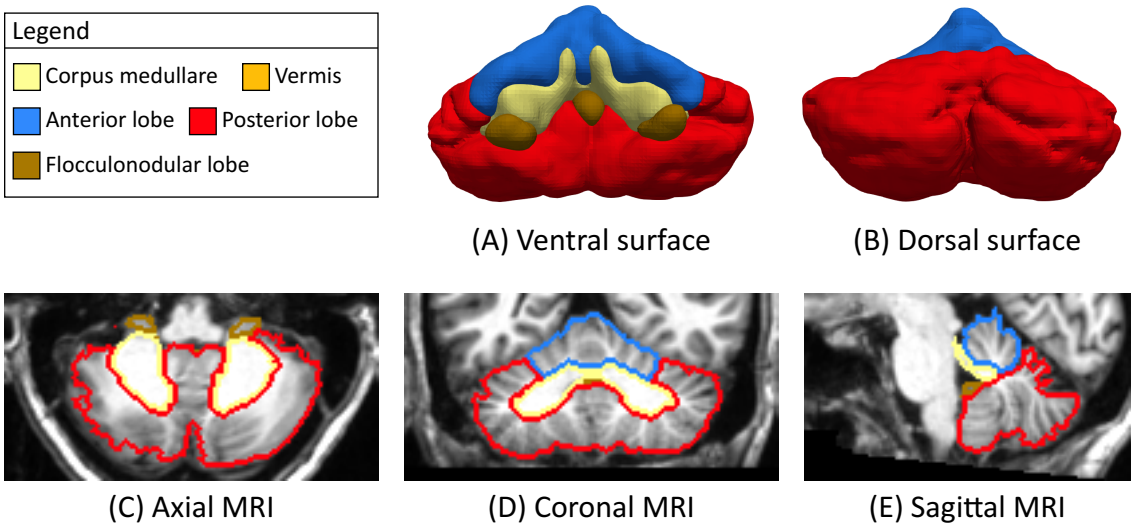


Figure 1-7. The cerebellum can be divided into the corpus medullare, the anterior lobe, and the posterior lobe. (A) and (B) show example cerebellar surfaces from the ventral and dorsal views, respectively. (C), (D), and (E) show axial, coronal, sagittal slices of an MRI image, respectively. These regions are shown or outlined in their corresponding colors that are shown in the legend. The image is from the T dataset.

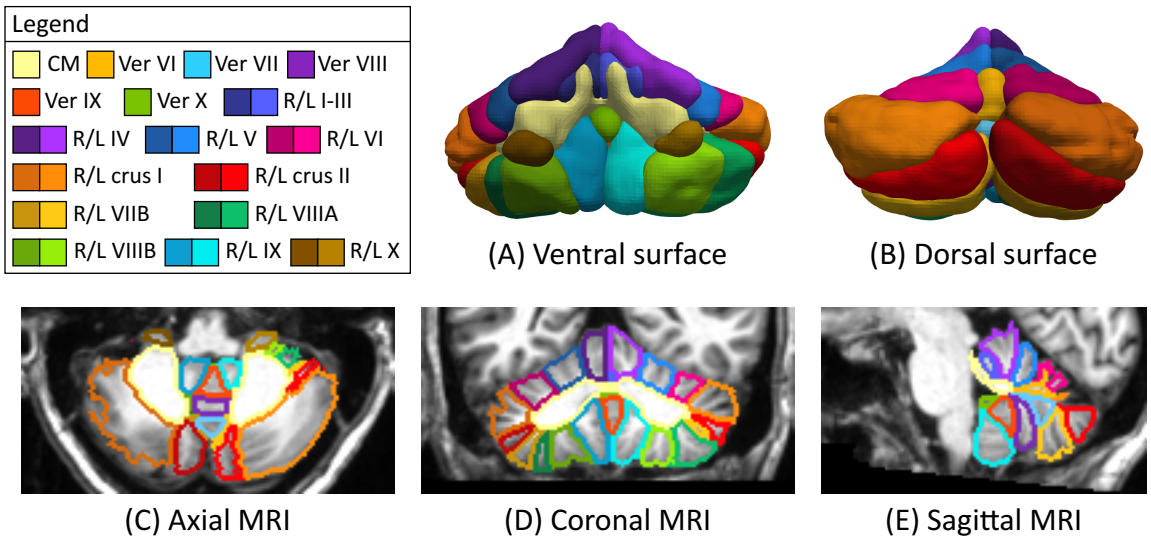


Figure 1-8. Cerebellar lobules of the T dataset. CM: corpus medullare. Ver: vermal lobule. R: right hemispheric lobule. L: left hemispheric lobule. (A) and (B) show example cerebellar surfaces from the ventral and dorsal views, respectively. (C), (D), and (E) show axial, coronal, sagittal slices of an MRI image, respectively. These regions are shown or outlined in their corresponding colors that are shown in the legend.

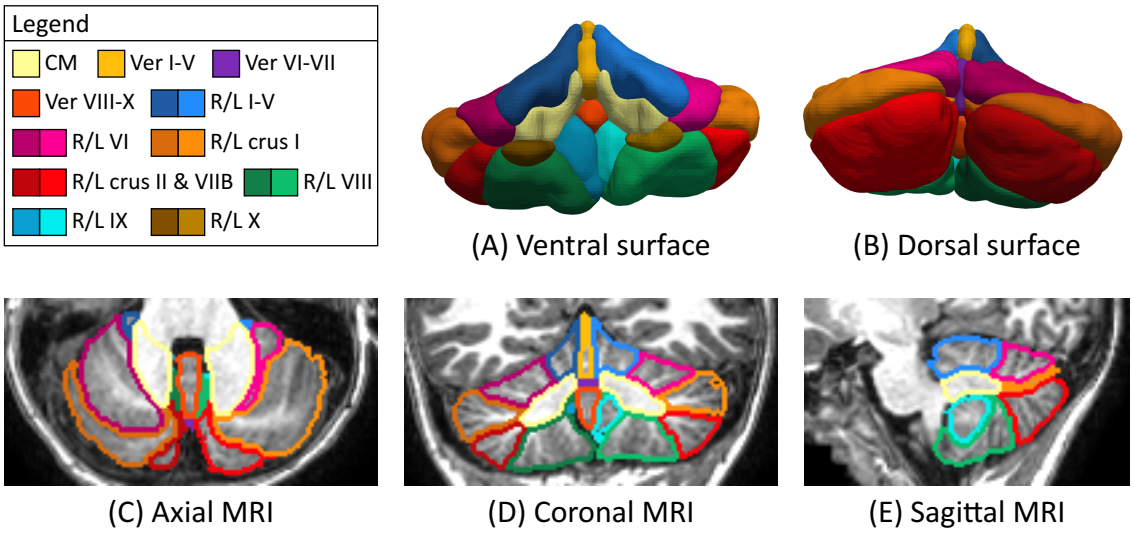


Figure 1-9. Cerebellar lobules of the M dataset. CM: corpus medullare. Ver: vermal lobule. R: right hemispheric lobule. L: left hemispheric lobule. (A) and (B) show example cerebellar surfaces from the ventral and dorsal views, respectively. (C), (D), and (E) show axial, coronal, sagittal slices of an MRI image, respectively. These regions are shown or outlined in their corresponding colors that are shown in the legend.

the lateral extension of these vermal lobules, the hemispheric lobules are also given the same numerals. In the T dataset, the vermis in the anterior lobe is not delineated; these vermal lobules are split in the medial and are merged into left and right hemispheric lobules. Lobules I–II and III are also merged together (see Fig. 1-8). In the M dataset, the anterior lobe is only delineated into the vermis and hemispheres (Fig. 1-9).

In the posterior lobe, the vermis is divided into vermal lobules VI, VIIAf VIIAt, VIIB, VIIIA, VIIIB, and IX. The hemispheric lobules corresponding to vermal lobules VIIAf and VIIAt are named crus I and crus II, respectively. The rest of hemispheric lobules is given the same numerals as their corresponding vermal lobules. In the T dataset, vermal lobules VIIAf, VIIAt, and VIIB are merged as vermal lobule VII, and vermal lobules VIIIA and VIIIB are merged as vermal lobule VIII (see Fig. 1-8). In the M dataset, vermal lobules VI and VII are merged as vermal lobules VI–VII, and vermal lobules VIII, IX, and X are merged as vermal lobules VIII–X. Crus II and hemispheric lobule VIIB are also merged together (see Fig. 1-9). We note that Carass *et al.* [19] separates the posterior lobe along the prepyramidal/prebiventer fissure into two parts, i.e., the superior posterior lobe—which contains vermal and hemispheric lobules VI and VII—and the inferior posterior lobe—which contains vermal and hemispheric lobules VIII and IX. We adopt this separation in this dissertation, but we note that it is possible to separate these two along the horizontal fissure between vermal lobules VIIAf and VIIAt and between crus I and crus II [11].

The flocculonodular lobe only contains vermal and hemispheric lobules X. In the M dataset, vermal lobule X is merged into the posterior lobe, forming vermis VIII–X (see Fig. 1-9).

This hierarchical definitions of the cerebellar regions can be represented as a tree. We show such a tree containing the definitions of both datasets in Fig. 1-10; each region is shown as a tree node whose sub-regions are shown as its child nodes.

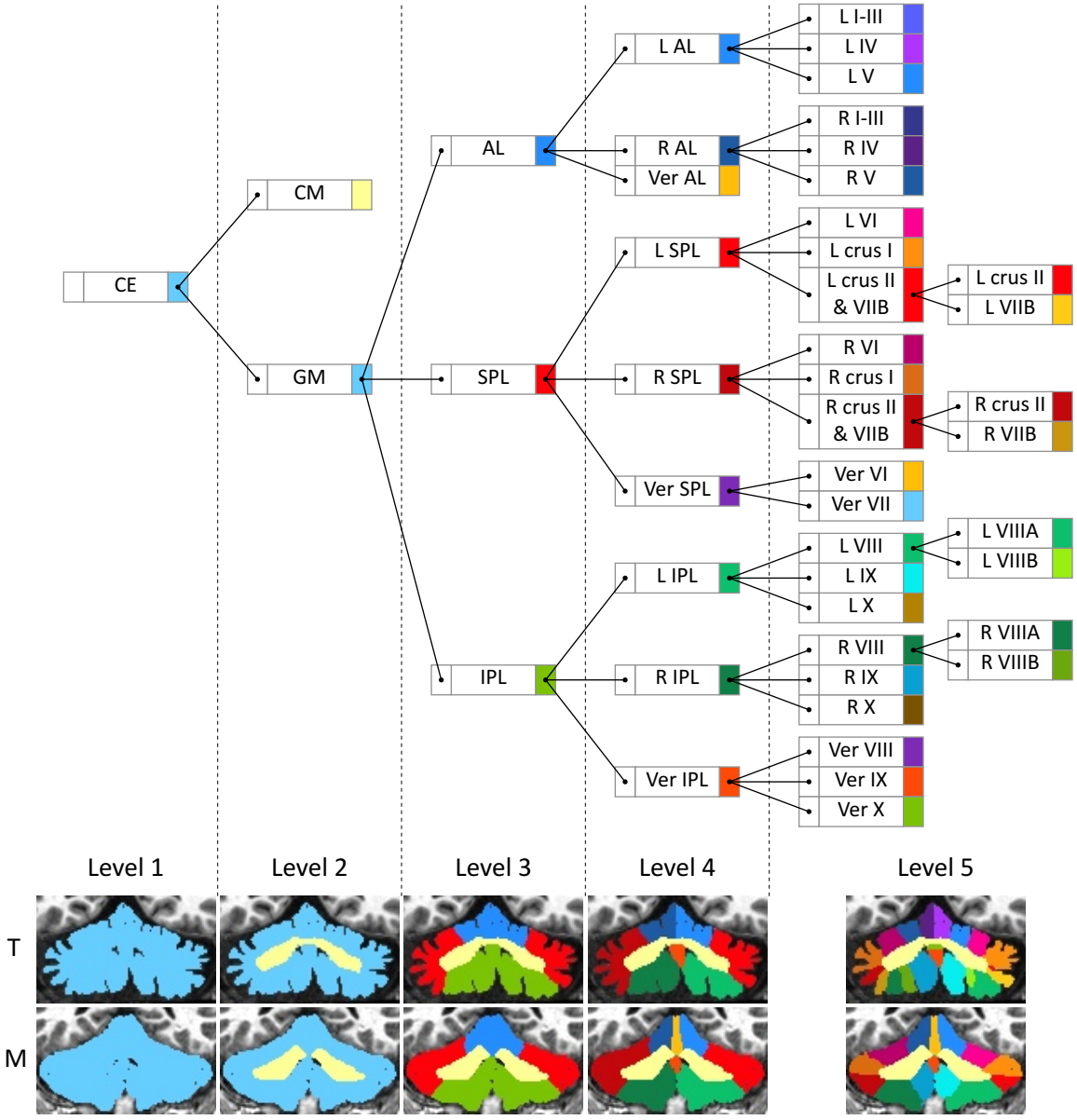


Figure 1-10. Hierarchical definitions of cerebellar sub-regions as a tree. This tree contains definitions of both the T and M datasets. Example parcellations at each level of these two datasets are shown at the bottom. We group vermal and hemispheric lobules X into the inferior posterior lobe for simplicity. CE: cerebellum. CM: corpus medullare. GM: gray matter. AL: anterior lobe. SPL: superior posterior lobe. IPL: inferior posterior lobe. L: left hemisphere. R: right hemisphere. Ver: vermis.

1.3 Previous Cerebellum Parcellation Algorithms

Several automated cerebellum parcellation methods have been proposed previously. SUIT [17] was the first published method to fully automatically parcellate the cerebellum. It constructs a spatially unbiased atlas and nonlinearly registers it to the target image. This approach was subsequently updated to use a probabilistic atlas [35]. Powell *et al.* [36] were the first to use machine learning methods for this task; in particular, a three-layer fully connected neural network and a support vector machine algorithm were applied to voxels with hand-crafted features. ACCLAIM [37] is a multi-object geometric deformable model [38, 39] driven by a boundary classification derived from a random forest. MAGeT [40] nonlinearly registers multiple atlases to the target image and fuses them using majority voting. CATK [41] uses Bayesian active appearance modeling to incorporate priors on shape, intensity, and inter-shape relationships. RASCAL [42] is a patch-matching approach with a subject-specific patch library constructed using nonlinear registration, and patches that are most similar to the query are selected for label fusion with majority voting. CERES [43] also uses the patch-matching framework to drive the label fusion. Yang *et al.* [44] uses multi-atlas registration and random-forest classification which are refined together using a graph-cut. Several other methods have also been reported in the literature [45, 46].

In a recent work by Carass *et al.* [19], eight cerebellum parcellation algorithms—including SUIT, two variants of SUIT, RASCAL, CERES2 (an improved version of CERES), and three DL-based algorithms—were compared using the T and M datasets. Despite the success of DL in other tasks of medical images, CERES2, which is based on conventional multi-atlas segmentation, performed the best in that comparison.

1.4 Deep Learning

Deep learning (DL) has been recently applied to medical image processing and has achieved great success. We focus on using DL techniques to develop cerebellum parcellation algorithms in this dissertation. In this section, we give a brief overview of DL techniques.

DL is a type of machine learning algorithm that uses deep artificial neural networks. A typical deep network is composed of a large series of simple linear and non-linear operations. Each step in the series is called a "layer". Most commonly used linear operations include the fully connected layer (a linear combination between its weights and the input) and the convolutional layer. Non-linear operations, also known as activations, can include, for example, the rectified linear activation function (ReLU, a piecewise linear function that keeps positive values but zeros out the negative) [47] and the sigmoid function. Other commonly used layers include the pooling (downsampling), dropout [48], batch normalization [49], and spatial-wise or channel-wise [50] attention layers. A deep network can essentially be regarded as a parameterized function that can approximate any function provided a sufficient number of layers and channels [51].

Like conventional machine learning, DL can be supervised learning, unsupervised learning, or even reinforcement learning. In typical supervised learning, the training data have "labels", such as the class of the whole image in a classification task or the class of each individual pixel/voxel in a segmentation task. A network or a set of networks then learns to map the input to its corresponding label. The cerebellum parcellation that we investigate in this dissertation falls into this category. In comparison, unsupervised learning does not have these labels and is thought to explore the structure or patterns within the training data. Generative models such as the generative adversarial network (GAN) [52] and the conditional GAN [53] are generally thought to fall

into this category since they do not seek to predict a label but rather to learn the whole data distribution by either mapping from a simple random variable (such as Gaussian) or conditioning on an image. Some techniques are categorized as semi-supervised learning, which is a mixture between supervised and unsupervised learning since their training data are only partially labeled. Reinforcement learning differs from these by learning to reach a goal through interacting with an environment (real or artificial).

A deep network is typically optimized by minimizing a loss function. In supervised learning, for example, the loss function is an error measurement between the network output and the target. It can be the mean squared error (MSE) if the output is an intensity image, or it can be the cross-entropy in a classification or segmentation task. This optimization is typically done by stochastic gradient descent (SGD), and the network parameters are updated via "back-propagation" [54], which is essentially the chain rule to calculate function derivatives. SGD can be regarded as a Monte Carlo version of the full-batch gradient descent, where the parameter gradients are calculated with respect to a single input sample or a small subset (mini-batch) of input samples during each iteration. While many state-of-the-art (SOTA) algorithms are optimized using vanilla SGD or SGD with momentum, adaptive SGD, such as Adam [55], are also widely used for its fast convergence and easy hyper-parameter tuning. It is also possible to incorporate regularization, such as weight decay, during the optimization. Data augmentations, such as reflection, rotation, and deformation [56], are often used to increase the amount of training data.

Convolutional neural networks (CNNs) refer to the networks that are mainly composed of convolutional instead of fully connected layers. CNNs are usually used for images since they are in theory shift-invariant and thus suitable to handle repetitively occurred patterns in an image. A convolutional layer typically has a very small kernel which is 3 by 3 for 2D images or 3 by 3 by 3 for 3D images. The stack of convolutions in

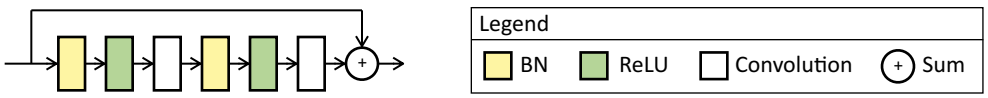


Figure 1-11. Architecture of a residual unit. BN: batch normalization.

a CNN then has receptive fields from small to large and learns to recognize low-level to high-level information. Image classification is one of the first applications that was greatly improved by deep CNNs. Various architectures (i.e., how a network is composed) of CNNs have been proposed over the years, including AlexNet [57], the VGG network [58], and ResNet [59, 60]. ResNet is composed of a series of residual units (see Fig. 1-11), and their residual connections make the optimization easier when the network is very deep. Although many architectures have been proposed nowadays, ResNet still serves as a baseline to compare with, and some work also claims that ResNet can still reach the SOTA performance with carefully tuned hyper-parameters [61].

For semantic segmentation in natural images or parcellation in medical images, people usually use a type of CNNs called fully convolutional networks (FCNs) [62]. Unlike CNNs that are used for image classification which gives a single label to the whole image, FCNs assign labels for each pixel (for 2D images) or voxel (for 3D images) of the image. An FCN can usually be divided into two parts: the encoder network and the decoder network, which contain downsampling and upsampling, respectively. The use of downsampling in the encoder reduces the GPU memory usage and increases the receptive field of the network to capture higher-level information; the upsampling in the decoder then restores the resolution of the image. The most famous FCN used in medical images is probably the U-Net [63]. U-Net-like networks have concatenations between each level of the encoder and decoder, and it works well for small sets of training data, which is typically the case for medical images. An example U-Net architecture is shown in Fig. 1-12.

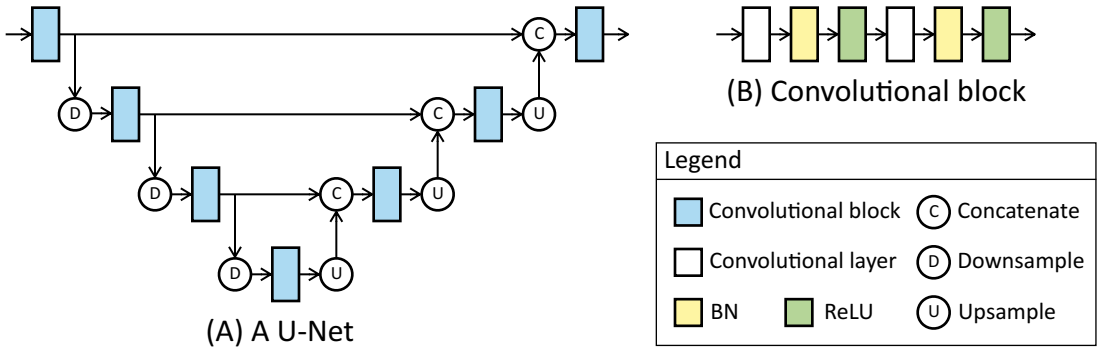


Figure 1-12. Architecture of a U-Net. (A) shows the architecture of a U-Net. (B) shows the architecture of the convolutional block (the blue box in (A)). BN: batch normalization.

1.5 More Details of the Manual Delineation Datasets

The T and M datasets of cerebellum manual delineations [19] are used throughout this dissertation. The T dataset contains 20 adult subjects. Their MPRAGE images were acquired from a 3.0 T scanner as axial slices with a thickness of 1.1 mm and an isotropic in-plane resolution of 0.8 mm . Their cerebella were expertly labeled into 28 regions (see Fig. 1-8), and these images were resampled to have a 1.0 mm isotropic resolution prior to the manual delineation, resulting in a spatial size of $182 \times 218 \times 182$ (in the order of left-right, anterior-posterior, and superior-inferior directions). Fifteen images of them are provided as training data of which six are healthy subjects and nine have cerebellar atrophy. All of the five testing images have cerebellar atrophy.

The M dataset contains 30 pediatric subjects. Their MPRAGE images were acquired from a 3.0 T scanner but as coronal slices with a thickness of 1.2 mm and an isotropic in-plane resolution of 1.0 mm, and they have a spatial size of $256 \times 155 \times 256$ (in the order of left-right, anterior-posterior, and superior-inferior directions). Their cerebella were expertly labeled into 18 regions (see Fig. 1-8), but they were not resampled prior to the manual delineation. Twenty images of them are provided as training data of which ten are healthy subjects and ten have diseases. The ten testing images contain five

Table 1-II. Summary of training data. The digital resolutions and spatial sizes are in the order of left-right, anterior-posterior, and superior-inferior directions.

	T dataset	M dataset
# training data (health/disease)	15 (6/9)	20 (10/10)
# testing data (health/disease)	5 (0/5)	10 (5/5)
# cerebellar regions	28	18
Digital resolution (mm)	$1.0 \times 1.0 \times 1.0$	$1.0 \times 1.2 \times 1.0$
Spatial size (# voxels)	$182 \times 218 \times 182$	$256 \times 155 \times 256$

healthy subjects and five subjects with diseases.

These two datasets are summarized in Table 1-II. Additional details can be found in Carass *et al.* [19].

1.6 Dissertation Overview

1.6.1 Contributions

There are four contributions in this dissertation.

1.6.1.1 Parcellating the Cerebellum Into Its Sub-Regions

In this contribution, we designed a CNN-based algorithm called ACAPULCO [64, 65] to parcellate the cerebellum to achieve better accuracy in shorter computational time compared to conventional methods. ACAPULCO uses two CNNs. The first CNN estimates a bounding box around the cerebellum, and the second CNN parcellates the region within this bounding box. ACAPULCO has been compared to previous algorithms using public benchmarks [19] and achieves the SOTA results. It is publicly available as

both Singularity[1] and Docker[2] containers at https://gitlab.com/shuohan/acapulco and is widely used around the world[3].

1.6.1.2 Incorporating Anatomical Knowledge into Network Architectures

We are also interested in incorporating anatomical knowledge of the cerebellum into the design of CNN architectures. First, we note that the brain is approximately left-right symmetric. Intuitively, suppose a convolution kernel in a CNN can recognize some image features on the left side of the brain, then a reflected version of this kernel should also be included in this CNN to recognize the counterpart features on the right side. Inspired by this rationale, we modified the group convolution [66] to implement a left-right-reflection-equivariant CNN [67]. This network architecture outperforms a conventional CNN trained with reflection augmentation in various brain segmentation tasks (although this improvement is not statistically significant in cerebellum parcellation). The second anatomical knowledge that we want to incorporate into the network architecture is the hierarchical definition of the cerebellum (see Fig. 1-10). We designed a network that was constructed in a tree structure with each node representing a cerebellar region and having child nodes that further subdivide the region into finer substructures. These two modifications of network architectures—i.e., incorporating the left-right symmetry and the hierarchical definition—do not improve upon the CNN of ACAPULCO with statistical significance; therefore, we did not explore them further and leave them as potential research directions in the future.

[1] See https://sylabs.io/singularity/.
[2] See https://www.docker.com/.
[3] See http://enigma.ini.usc.edu/ongoing/enigma-ataxia/.

1.6.1.3 Conducting Statistical Analysis of Cerebellar Sub-Regional Volumes

Developing a parcellation algorithm is only the first step towards analyzing the cerebellum. In this contribution, we use ACAPULCO to study the cerebellar volumes during aging. Previous studies of cerebellar volumes are limited to either cross-sectional analyses or small numbers of subjects and cerebellar sub-regions [11–13, 68–70]. In this contribution, we applied ACAPULCO to 2,023 MRI images of 822 cognitively normal subjects from the Baltimore Longitudinal Study of Aging (BLSA) [71] to study both cross-sectional and longitudinal changes of sub-regional volumes of the cerebellum during normal aging [72]. These cerebella were divided into 28 regions, and we applied linear mixed effect models [73] to each region with its volume as the dependent variable and age, biological sex, and their interactions as covariates. We provide the longitudinal trajectories of these volumes with respect to age and sex and provide maps of whether a covariate contributes to each region with statistical significance. Our analysis is a step forward to better understand the cerebellum.

1.6.1.4 Super-Resolving MRI for Better Parcellation

ACAPULCO and previous methods [35, 43, 44] only uses a T1w MRI image to parcellate the cerebellum. However, some T1w images have low contrast between the cerebellar gray matter and the transverse and the sigmoid sinuses (see Fig. 1-3(B)) which can sometimes cause oversegmentation of the cerebellum. In contrast, a T2w image has lower intensity in the sinuses and higher intensity in the cerebellum, which makes it easier to distinguish between these two. Therefore, we would like to use a co-registered T2w image of the same subject as a complement to the T1w image to parcellate the cerebellum.

However, unlike the T1w image which is typically acquired using a 3D MRI se-

quence (such as MPRAGE), a T2w image is commonly acquired with a 2D multi-slice sequence which has a lower through-plane resolution than its in-plane resolution. Therefore, we first focus on super-resolution (SR) to improve the quality of such images in this contribution. We first developed an algorithm called ESPRESO [74, 75] to estimate the through-plane point spread function (PSF)—i.e., the slice profile—of a 2D multi-slice acquisition. In a 2D multi-slice acquisition, high-resolution (HR) and LR patches can be extracted along its in-plane and through-plane directions, respectively. We propose a variant of the GAN [52] to match the distributions of these two kinds of patches, and the slice profile is learned as a part of the mapping between them. ESPRESO is the first algorithm to estimate the slice profile without access to the MRI scanner or details of the MRI sequence. We next proposed an improved implementation of an internally supervised SR algorithm, SMORE [76]. Our new implementation, termed S-SMORE, uses the RCAN architecture [77] with PixelShuffle [78] to improve the computational speed and the accuracy. By incorporating ESPRESO into S-SMORE to create more faithful training data, we are able to improve S-SMORE performance even further. Finally, we conducted experiments to demonstrate the benefits of using a super-resolved T2w image alongside the T1w image to parcellate the cerebellum in this contribution.

1.6.2 Organization

Chapter 2 presents our cerebellum parcellation algorithm, ACAPULCO. Chapter 3 explores the incorporation of anatomical knowledge into the CNN architecture design. Chapter 4 presents our analyses of the cerebellar sub-regional volumes with respect to age and sex during normal aging. Chapter 5 presents our work on super-resolving 2D multi-slice T2w images and incorporating them into ACAPULCO. Chapter 6 includes a discussion of the results and presents future research directions.

Chapter 2

Parcellating the Cerebellum Into Its Sub-Regions

2.1 Introduction

In this chapter, we describe ACAPULCO (automatic cerebellum anatomical parcellation using U-Net with locally constrained optimization), a new approach to cerebellum parcellation that uses a cascade of two CNNs, as shown in Fig. 2-1. The locating network detects the cerebellum thus reducing the spatial size of the input to the parcellating network which divides the cerebellum into anatomically meaningful regions. The strategy of cascaded networks has been used in previous work in medical image segmentation [79–82]. Our first stage is not a coarse anatomical segmentation but instead a regression on the coordinates of a 3D bounding box containing the cerebellum. In addition, we exclusively use 3D CNNs since all the dimensions of the image context are thought to be necessary to accurately parcellate the cerebellum. Our locating network was modified from the pre-activation ResNet in He *et al.* [60], and our parcellating

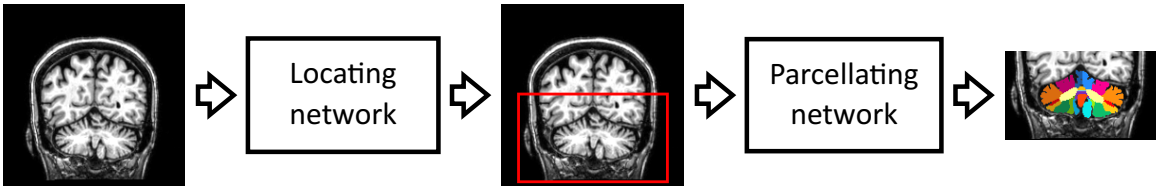


Figure 2-1. Flowchart of ACAPULCO. The cerebellum is parcellated by two CNNs: the locating network finds a bounding box around the cerebellum, and the parcellating network labels the regions within this bounding box.

network was modified from a 3D U-Net [83] with residual connections as in the ResNet. We compared ACAPULCO to CERES2 (the best method in Carass *et al.* [19]) and to an improved version of CGCUTS [44] using the T and M datasets from Carass *et al.* [19]. ACAPULCO was also evaluated on the Kirby dataset [9] to assess its intra-subject stability. Finally, it was also applied to a pediatric dataset [84], our own cerebellum ataxia dataset [14], and the OASIS-3 Alzheimer's disease dataset [22][1] to show its broad applicability. It is publicly available as both Singularity and Docker containers at`https://gitlab.com/shuohan/acapulco`. It has been adopted by the ENIGMA-Ataxia working group[2] and is widely used around the world.

2.2 Methods

2.2.1 Pre-processing

Since we have observed that skull-stripping [33, 85] can sometimes remove part of the cerebellum, our approach is designed to work on MRI images of the whole head. As the first step of the pre-processing, the intensity inhomogeneity of these images

[1]OASIS-3: Principal Investigators: T. Benzinger, D. Marcus, J. Morris; NIH P50 AG00561, P30 NS09857781, P01 AG026276, P01 AG003991, R01 AG043434, UL1 TR000448, R01 EB009352. AV-45 doses were provided by Avid Radiopharmaceuticals, a wholly owned subsidiary of Eli Lilly.

[2]See `http://enigma.ini.usc.edu/ongoing/enigma-ataxia/`.

is corrected using N4 [27]. When using N4, a confidence image can be specified to weight the input voxels during the B-spline fitting. For better performance, we created a confidence image for N4 using a brain mask extracted from ROBEX [33] and then blurred it using a 3D Gaussian kernel with a standard deviation (SD) of 3 mm.

As the second step of pre-processing, the inhomogeneity-corrected image is rigidly registered to the 1 mm isotropic ICBM 2009c template [31] in MNI space. We do not require intensity normalization [28, 30] as we use instance normalization [86] in our networks, as discussed in Section 2.2.9. The T dataset did not require registration to MNI space as it was roughly aligned to that space before delineation.

2.2.2 The Locating Network

Our 3D locating network, shown in Fig. 2-2(A), is used to find a bounding box around the cerebellum. This network is composed of a series of contracting blocks, shown in Fig. 2-2(B), producing increasing numbers of feature maps at decreasing spatial sizes. The spatial global average pooling and the following fully connected layer convert the feature maps into six numbers specifying the starting and stopping coordinates of the bounding box along the x, y, and z axes. Our contracting blocks are adopted from Kayalibay *et al.* [83] and are analogous to the dimension-halving residual unit of the pre-activation ResNet [60]. In a conventional residual unit, two pairs of nonlinear operations and convolutions are used to calculate a residual which is added back to the input. This process can enable more direct information propagation from the input, making it easier to optimize the whole network during training [60]. In contrast, the dimension-halving residual block applies an additional convolution to the input to change its spatial size, resulting in three convolutions in total for a block [59, 60]. Our architecture uses a simpler approach wherein the first convolution, originally calculating only the

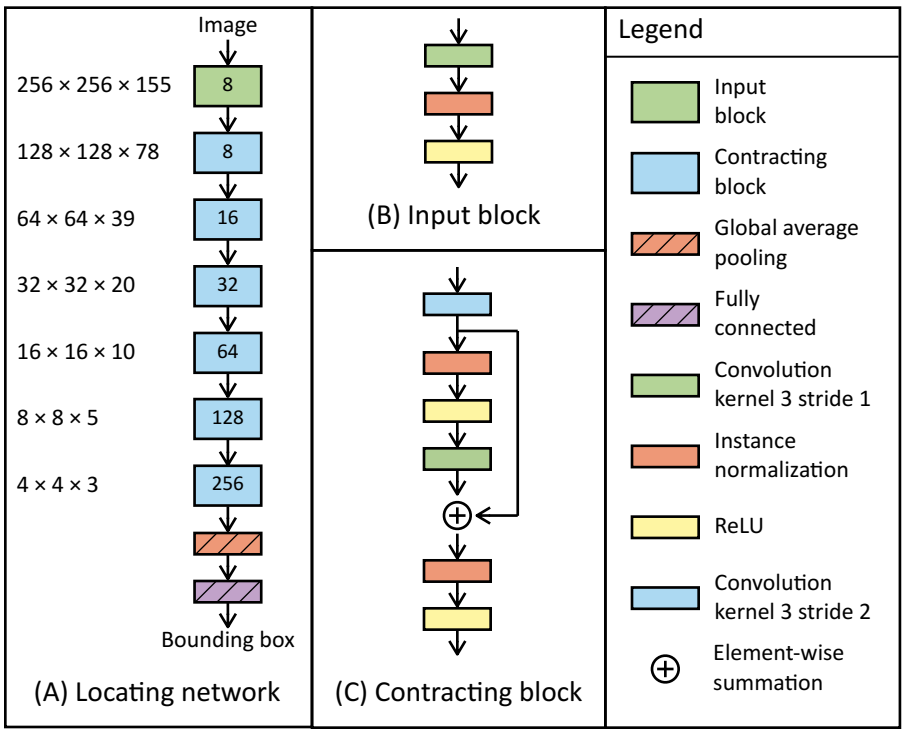


Figure 2-2. Architectures of (A) the locating network, (B) the input block, and (C) the contracting block. In (A), the number of output feature maps is marked within each block, and the output spatial size is marked on the side.

residual, is used to halve the dimension of the input as well, which in turn results in only two convolutions in this block. This yields comparable results to the dimension-halving residual unit and has fewer weights to train. We note that zero-padding with a size of one was used before applying $3 \times 3 \times 3$ convolutions and the "He normal" weight initialization [87] was used. We also used instance normalization [86] instead of batch normalization [49].

2.2.3 Training the Locating Network

The ground truth bounding box was obtained from the union of all the manually delineated labels, i.e., the whole cerebellum. The minimum and maximum coordinates among

all the cerebellum voxels were used as the starting and stopping coordinates of the bounding box, respectively, for each of the x, y, and z axes. The smooth $l1$ norm [88], L_1, was used as the loss function during the training:

$$L_1 = \frac{1}{6}\sum_{i=1}^{6} s(x_i - y_i), \text{ where } s(u) = \begin{cases} 0.5u^2, & \text{if } |u| < 1, \\ |u| - 0.5, & \text{otherwise,} \end{cases} \tag{2.1}$$

where x_i is a predicted bounding box coordinate with its corresponding ground truth y_i. The Adam optimization algorithm [55] was used with parameters $\alpha = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-7}$. This network was trained for 600 epochs with a mini-batch size of 1. Left-right flipping and random translation, scaling, and rotation were used for data augmentations described in detail in Section 2.2.8.

2.2.4 Post-processing of the Bounding Box

The locating network, which is trained to find a relatively tight bounding box around the cerebellum, yields a bounding box that does not have a fixed size and sometimes cuts off part of the cerebellum. Therefore, instead of resampling the image region within the bounding box [89], we expand the bounding box symmetrically in all six cardinal directions so that it has a fixed size of 160 $\times$ 96 $\times$ 96 voxels in the T dataset and of 128 $\times$ 96 $\times$ 96 voxels in the M dataset. These bounding box sizes, which are of adequate sizes according to the literature [90], have included the cerebella in all of the data that we have tested to date.

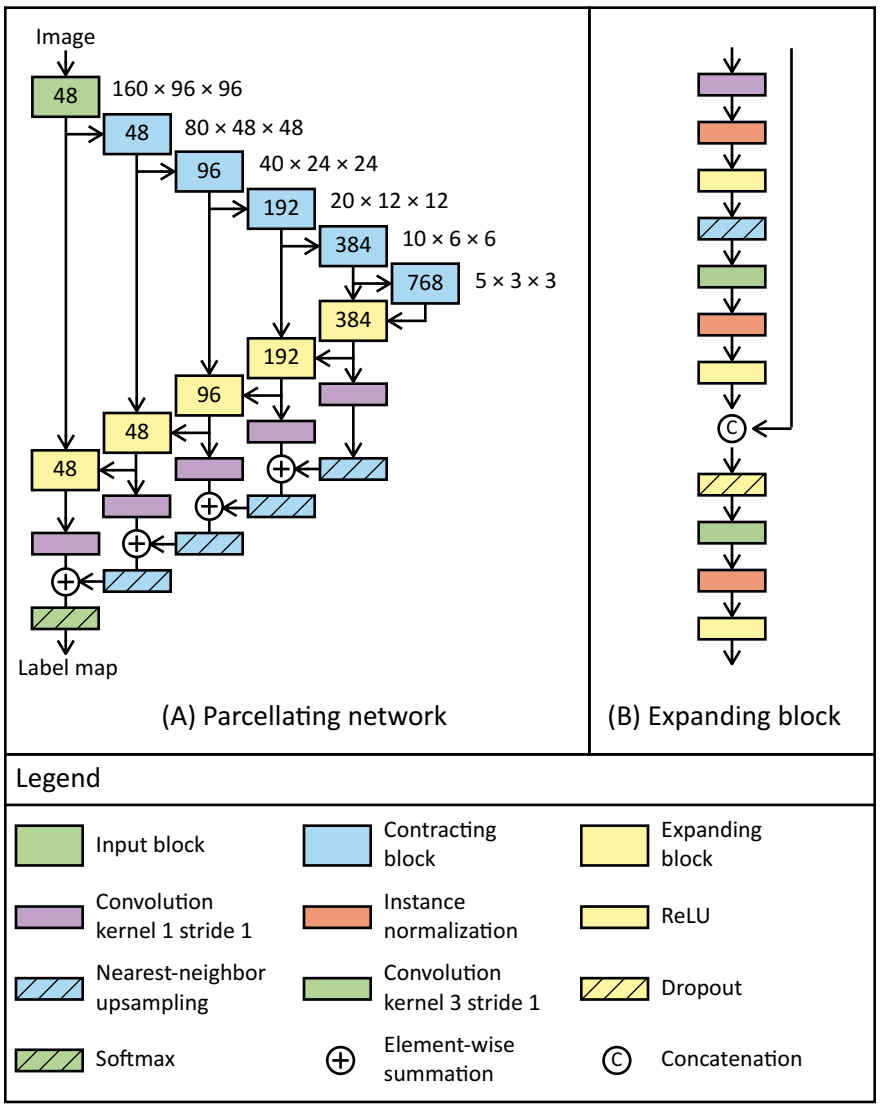


Figure 2-3. Architectures of (A) the parcellating network and (B) the expanding block. The number of output feature maps is marked within each block, and the output spatial size is marked on the side of contracting blocks. The output spatial sizes of the expanding blocks are the same as their corresponding contracting blocks.

2.2.5 The Parcellating Network

The volume defined by the estimated bounding box from our locating network is used as input to our parcellating network shown in Fig. 2-3(A). This network, modified from a 3D U-Net in Kayalibay *et al.* [83], has a series of contracting blocks with the same

structure as our locating network (see Fig. 2-2(C)). These contracting blocks gradually expand the receptive field through successive strided and non-strided convolutions and therefore capture both local and global context. To restore the resolution, an expanding block operates on the feature maps from its previous block as well as the corresponding contracting block, as shown in Fig. 2-3(B), acting as a type of learnable interpolation.

Our expanding blocks have a dropout rate of 0.5. Their outputs are converted into new feature maps whose dimensions match the number of labels in the training data using convolutions with a kernel size of 1 (solid purple boxes in Fig. 2-3(A)). These are further upsampled using nearest-neighbor interpolation (striped blue boxes in Fig. 2-3(A)) and added to the maps at the next higher resolution. This strategy, described in Kayalibay *et al.* [83] (and different from the original U-Net [63, 91]), can encourage faster training convergence. In the final step, the softmax generates a probability map for each label at all voxels. For the T dataset, we have 29 labels (28 cerebellar regions plus the background), while the M dataset has 19 labels (18 cerebellar regions plus the background).

2.2.6 Training the Parcellating Network

We converted the label maps into C binary channels, where each channel represents a label, and only one channel can be activated at each voxel. The loss function, L_D, was computed as one minus the average Dice coefficient of the C channels [92],

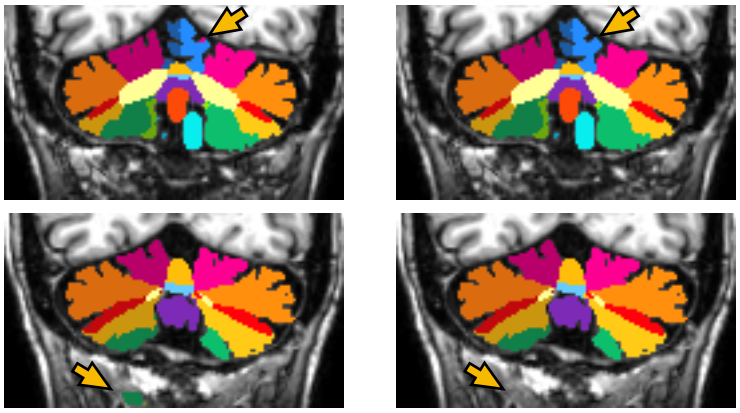
$$L_D = 1 - \frac{1}{C}\sum_{c=1}^{C}\frac{\epsilon + 2\sum_{i=1}^{N} x_{ci}y_{ci}}{\epsilon + \sum_{i=1}^{N}(x_{ci} + y_{ci})}, \tag{2.2}$$

where N is the number of voxels in the spatial domain, x_{ci} is the i^{th} voxel in the c^{th} channel of the prediction $X \in \mathbb{R}^{C\times N}$, y_{ci} is the corresponding voxel from the ground

truth $Y \in \mathbb{R}^{C \times N}$, and $\epsilon = 0.001$ which prevents division by zero. The Adam optimization algorithm [55] was used with the same parameters as in the training of the locating network. We found that the training of our parcellating network was in a plateau after 400 and 300 epochs for the T and M datasets, respectively, so we used these numbers of epochs in all our experiments. The mini-batch size was 1. Left-right flipping and random scaling, rotation, and deformation were used for data augmentations, as described in detail in Section 2.2.8.

2.2.7 Post-processing of the Parcellation

Since the parcellating network performs per-voxel classification, isolated mislabeling can happen even though each cerebellar region should be a connected component (see Fig. 2-4). To correct this, we use a post-processing step based on connected components of each label. First, connected components are calculated for each cerebellar label separately. Second, for each cerebellar label, we find the largest connected component and define a threshold T as 0.9 times its volume. We categorize the other connected components into two groups: a *larger* connected component if its volume is greater than T and a *smaller* connected component, otherwise. Third, a larger connected component keeps its original label while a smaller connected component has its label changed according to its adjacency to other larger connected components. If it is next to at least one larger connected component, the label is changed to the larger connected component with which it shares most of its boundary; otherwise, the label becomes background. Finally, any holes within a label are filled using binary morphological operations.



(A) Before post-processing (B) After post-processing

Figure 2-4. Comparison between (A) before and (B) after the post-processing. Note that the post-processing can correct isolated mislabeling as indicated by the yellow arrows.

2.2.8 Data Augmentations

To improve the generalization of the inference, several data augmentation methods were applied during training (see Fig. 2-5). Specifically, flipping, translation, scaling, and rotation were used for the locating network, while flipping, scaling, rotation, and deformation were used for the parcellating network:

- **Flipping** (Fig. 2-5(B)). An image and its manual delineations were flipped left-to-right, and hemispheric regions (such as left and right hemispheric lobules X) in the flipped image were relabeled to match their sides.
- **Translation** (Fig. 2-5(C)). Random integer offsets along the x, y, and z axes were uniformly sampled from -30 to 30 voxels.
- **Scaling** (Fig. 2-5(D)). Enlarging/shrinking was decided with equal probability, and three random numbers were uniformly sampled from 1 to 1.6. If enlarging, these three numbers were used as the scaling factors for the x, y, and z axes, respectively; if shrinking, the inverses of them were used as the scaling factors.

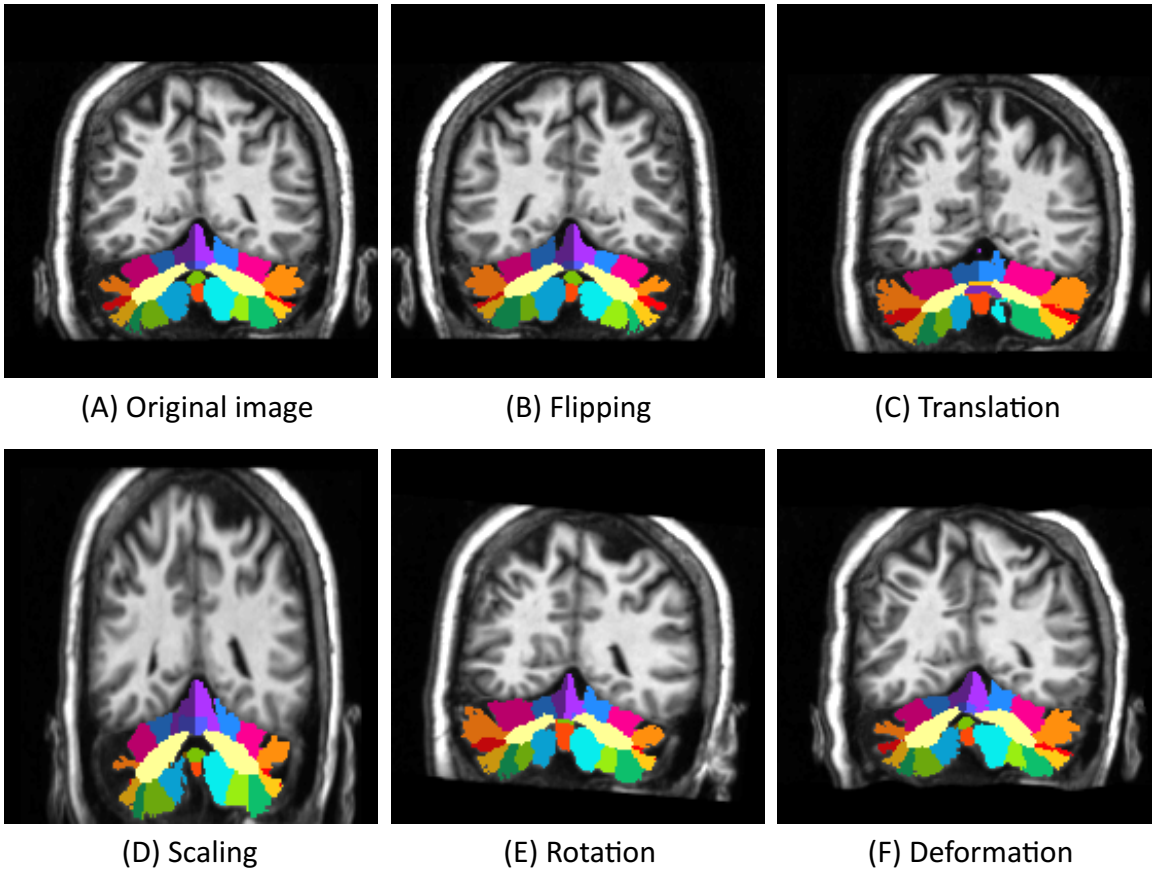


Figure 2-5. Examples of data augmentations: (A) the original, (B) the flipped, (C) the translated, (D) the scaled, (E) the rotated, and (F) the deformed images. The transformed label maps are plotted on top of the images.

- **Rotation** (Fig. 2-5(E)). The random rotation angles around the x, y, z axes were uniformly sampled from -15 to 15 degrees, and the image was rotated with respect to its image center.

- **Deformation** (Fig. 2-5(F)). Random translation per voxel was uniformly sampled, and a Gaussian smoothing with sigma equal to 5 voxels was applied separately to the x, y, and z components. The x, y, and z components were then independently scaled across all the voxels so that the largest value was 8 voxels.

Translation, scaling, rotation, and deformation were applied to the flipped images

as well as the original images, but these four were not composed. Note that our data augmentations were performed "on the fly"—i.e., not computed and stored in advance—and was applied before cropping around the cerebellum using the bounding box.

2.2.9 Instance Normalization and MRI Intensity Normalization

We noted in Section 2.2.1 that no *intensity* normalization is required in our pre-processing steps. We show here that, unlike batch normalization [49], use of instance normalization [86] makes intensity normalization processes, such as in Nyúl & Udupa [28] and Reinhold *et al.* [30], unnecessary. Given input $\boldsymbol{X} \in \mathbb{R}^{B \times C \times N}$, where B is the number of samples within a mini-batch, C is the number of channels (or feature maps), and N is the number of voxels in the spatial domain, instance normalization standardizes the $(b, c, n)^{\text{th}}$ element, x_{bcn}, of $\boldsymbol{X}$, to generate the output y_{bcn}. Specifically,

$$y_{bcn} = \frac{x_{bcn} - \mu_{bc}}{\sqrt{\sigma_{bc}^2 + \epsilon}}, \text{ with } \mu_{bc} = \frac{1}{N}\sum_{n=1}^{N} x_{bcn} \text{ and } \sigma_{bc}^2 = \frac{1}{N}\sum_{n=1}^{N}(x_{bcn} - \mu_{bc})^2, \tag{2.3}$$

where ϵ is a small number to prevent division by zero (e.g., its default value is 1×10^{-5} in PyTorch version 1.3.1 and 1×10^{-3} in Keras-contrib version 2.0.8).

Since MRI scanners acquire images with arbitrary units, it is usually assumed that observed MRI image intensities are affine-transformed values with respect to a true value in a normalized space in which the value of a specific tissue is comparable across subjects, visits, and sites [29]. That is, given a tensor $\boldsymbol{X} \in \mathbb{R}^{B \times C \times N}$ drawn from this normalized space, an observed tensor $\boldsymbol{X}' \in \mathbb{R}^{B \times C \times N}$ is assumed to be related by $\boldsymbol{X}' = u_{bc}\boldsymbol{X} + v_{bc}$, where $\forall u_{bc}, v_{bc} \in \mathbb{R}$ and $u_{bc} > 0$ are sample-specific constants.

Instance normalization then computes,

$$\mu'_{bc} = \frac{1}{N}\sum_{n=1}^{N} x'_{bcn} = \frac{1}{N}\sum_{n=1}^{N}(u_{bc}x_{bcn} + v_{bc}) = u_{bc}\mu_{bc} + v_{bc}, \tag{2.4}$$

and

$$\begin{aligned} {\sigma'_{bc}}^2 &= \frac{1}{N}\sum_{n=1}^{N}(x'_{bcn} - \mu'_{bc})^2 = \frac{1}{N}\sum_{n=1}^{N}(u_{bc}x_{bcn} + v_{bc} - u_{bc}\mu_{bc} - v_{bc})^2 \\ \Rightarrow {\sigma'_{bc}}^2 &= u_{bc}^2\sigma_{bc}^2. \end{aligned} \tag{2.5}$$

Therefore,

$$y'_{bcn} = \frac{x'_{bcn} - \mu'_{bc}}{\sqrt{{\sigma'_{bc}}^2 + \epsilon}} = \frac{u_{bc}x_{bcn} + v_{bc} - u_{bc}\mu_{bc} - v_{bc}}{\sqrt{u_{bc}^2\sigma_{bc}^2 + \epsilon}} \approx y_{bcn}, \tag{2.6}$$

when ϵ is small enough, which indicates that instance normalization can account for underlying intensity transformations of MRI images. We also note that although an instance normalization layer is always *after* a convolutional layer in our networks (see Figs. 2-2 and 2-3), Eq. (2.6) still holds for a single-channel image (as in our case) since convolution is a linear operation (suppose the effect of zero-padding in a convolutional layer is negligible).

In contrast, batch normalization calculates the mean, μ_c, and variance, σ_c^2, across multiple samples within the mini-batch as follows,

$$y_{bcn} = \frac{x_{bcn} - \mu_c}{\sqrt{\sigma_c^2 + \epsilon}} \text{ with } \mu_c = \frac{1}{BN}\sum_{b=1}^{B}\sum_{n=1}^{N} x_{bcn} \text{ and } \sigma_c^2 = \frac{1}{BN}\sum_{b=1}^{B}\sum_{n=1}^{N}(x_{bcn} - \mu_c)^2. \tag{2.7}$$

In this scenario where the u_{bc}'s and v_{bc}'s vary from sample to sample, the output will be different from the underlying true value in general. Another problem with batch normalization is that it usually uses the accumulated mean and variance from the training to perform the inference. Instance normalization, on the other hand, uses the

sample-specific mean, μ_{bc}, and variance, σ^2_{bc} (as in training), which can be different from those of the training images. For these two reasons, we do not perform intensity normalization in pre-processing and instead assume that our networks built with instance normalization can handle MRI intensity non-standardization.

2.3 Experiments and Results

2.3.1 Execution Time and Memory Consumption

To record the execution time of ACAPULCO, we monitored the processing of an image for ten repeats. On a 16-core CPU (Xeon E5-2620 v4, Intel corporation), loading the models took (mean $\pm$ SD) 24.9 $\pm$ 0.17 seconds, applying the locating network took 3.35 $\pm$ 0.02 seconds, applying the parcellating network took 16.48 $\pm$ 0.02 seconds, and the overall parcellation took 48.96 $\pm$ 0.34 seconds. On a GPU (Tesla M40, NVIDIA Corporation), loading the models took 26.0 $\pm$ 0.11 seconds, applying the locating network took 2.44 $\pm$ 0.03 seconds, applying the parcellating network took 3.61 $\pm$ 0.02, and the overall parcellation took 36.64 $\pm$ 0.49 seconds. Since loading the models costs most of the computational time, the parcellation can be accelerated by loading the models once and processing multiple images in series. Post-processing a parcellation took 18.71 $\pm$ 0.21 seconds on a single core of the same CPU. The peak memory consumption is approximately 6 GB when applying our networks on the CPU.

2.3.2 Comparison to Other Methods

Carass *et al.* [19] compared eight cerebellum parcellation algorithms using the T and M datasets. In this section, we used the same datasets to compare ACAPULCO

to CERES2 (the top method in Carass *et al.* [19]) and an improved implementation of CGCUTS [44] (a recent cerebellum parcellation algorithm that was not included in Carass *et al.* [19]). Note that to train and test ACAPULCO for this comparison, the images of these two datasets were not transformed into MNI space; otherwise, we would need to transform the output parcellations back into the space of the manual delineations for evaluation, which may cause extra interpolation error. ACAPULCO was trained from scratch for each dataset separately. The Dice coefficients [93] between the parcellation of each testing image and its corresponding manual delineation were calculated for each cerebellar region and averaged across all regions. The Dice coefficient, DSC, of two binary segmentations, X and Y, is defined as

$$DSC = 2\frac{|X \cap Y|}{|X| + |Y|}, \tag{2.8}$$

where $|\cdot|$ indicates the number of positive voxels. We show these Dice coefficients in Figs. 2-6 and 2-7 for the T and M datasets, respectively, and report the mean values of them across all testing images in Tables 2-I and 2-II. Two-sided paired Wilcoxon tests between the Dice coefficients of CERES2 and ACAPULCO for each region were computed and significant differences ($*$ for $p < 0.05$ and $**$ for $p < 0.01$) are denoted in Figs. 2-6 and 2-7 and Tables 2-I and 2-II. For the T dataset, no region is significantly different, while for the M dataset, five regions—left and right crus I, left hemispheric lobule IX, and left and right hemispheric lobules X—are significantly different, and ACAPULCO has better Dice coefficients. In terms of the mean Dice coefficients across all testing images, i.e., the bars on top of the dots in Figs. 2-6 and 2-7 and the values in Tables 2-I and 2-II, ACAPULCO scores the best in 18 out of 28 regions for the T dataset and 16 out of 18 regions for the M dataset. Example parcellations from ACAPULCO are shown in Figs. 2-8 and 2-9 for the T and M datasets, respectively.

Table 2-I. Dice coefficients of CERES2, CGCUTS, and ACAPULCO of the T dataset. The means and standard deviations (SDs) of each region are calculated across all testing images. The bottom row shows the average mean values and the average SDs from all regions. The best means among the three algorithms are highlighted in blue. CM: corpus medullare. Ver: vermal lobule. L: left hemispheric. R: right hemispheric.

	CERES2	CGCUTS	ACAPULCO
	Mean $\pm$ **SD**	**Mean** $\pm$ **SD**	**Mean** $\pm$ **SD**
CM	0.8921 $\pm$ 0.0278	0.8732 $\pm$ 0.0463	0.8992 $\pm$ 0.0249
Ver VI	0.7886 $\pm$ 0.0523	0.7911 $\pm$ 0.0463	0.8178 $\pm$ 0.0565
Ver VII	0.7634 $\pm$ 0.0671	0.7776 $\pm$ 0.0600	0.7976 $\pm$ 0.0554
Ver VIII	0.8933 $\pm$ 0.0179	0.8944 $\pm$ 0.0149	0.8912 $\pm$ 0.0244
Ver IX	0.8589 $\pm$ 0.0224	0.8616 $\pm$ 0.0319	0.8589 $\pm$ 0.0544
Ver X	0.8471 $\pm$ 0.0469	0.8480 $\pm$ 0.0258	0.8460 $\pm$ 0.0404
L I–III	0.7688 $\pm$ 0.0492	0.7920 $\pm$ 0.0233	0.7994 $\pm$ 0.0238
R I–III	0.6322 $\pm$ 0.1866	0.6770 $\pm$ 0.1595	0.6779 $\pm$ 0.1835
L IV	0.7742 $\pm$ 0.1167	0.7738 $\pm$ 0.1150	0.7779 $\pm$ 0.1478
R IV	0.7606 $\pm$ 0.1177	0.7369 $\pm$ 0.0999	0.7773 $\pm$ 0.0962
L V	0.6444 $\pm$ 0.2882	0.6972 $\pm$ 0.1963	0.6273 $\pm$ 0.3532
R V	0.6583 $\pm$ 0.1845	0.5801 $\pm$ 0.1104	0.6589 $\pm$ 0.2139
L VI	0.8435 $\pm$ 0.1166	0.8643 $\pm$ 0.0830	0.8389 $\pm$ 0.1298
R VI	0.8567 $\pm$ 0.0292	0.8434 $\pm$ 0.0126	0.8711 $\pm$ 0.0400
L crus I	0.9337 $\pm$ 0.0171	0.9262 $\pm$ 0.0264	0.9384 $\pm$ 0.0131
R crus I	0.9094 $\pm$ 0.0274	0.9111 $\pm$ 0.0150	0.9139 $\pm$ 0.0168
L crus II	0.7943 $\pm$ 0.0953	0.7622 $\pm$ 0.0855	0.8079 $\pm$ 0.0732
R crus II	0.8398 $\pm$ 0.0640	0.8544 $\pm$ 0.0608	0.8464 $\pm$ 0.0826
L VIIB	0.5624 $\pm$ 0.3186	0.5716 $\pm$ 0.3098	0.5779 $\pm$ 0.3154
R VIIB	0.6467 $\pm$ 0.3184	0.7332 $\pm$ 0.2659	0.6613 $\pm$ 0.3526
L VIIIA	0.6510 $\pm$ 0.2408	0.7435 $\pm$ 0.1798	0.7576 $\pm$ 0.1784
R VIIIA	0.6572 $\pm$ 0.2607	0.6950 $\pm$ 0.2222	0.6745 $\pm$ 0.1402
L VIIIB	0.8242 $\pm$ 0.1493	0.8894 $\pm$ 0.0300	0.9006 $\pm$ 0.0276
R VIIIB	0.8237 $\pm$ 0.0703	0.7962 $\pm$ 0.0500	0.8018 $\pm$ 0.0595
L IX	0.9039 $\pm$ 0.0372	0.9078 $\pm$ 0.0370	0.9192 $\pm$ 0.0304
R IX	0.8992 $\pm$ 0.0263	0.9006 $\pm$ 0.0266	0.8980 $\pm$ 0.0373
L X	0.7165 $\pm$ 0.0470	0.7264 $\pm$ 0.0289	0.7510 $\pm$ 0.0157
R X	0.7450 $\pm$ 0.0510	0.7414 $\pm$ 0.0839	0.8099 $\pm$ 0.0632
Average	0.7818 $\pm$ 0.1088	0.7928 $\pm$ 0.0874	0.7999 $\pm$ 0.1018

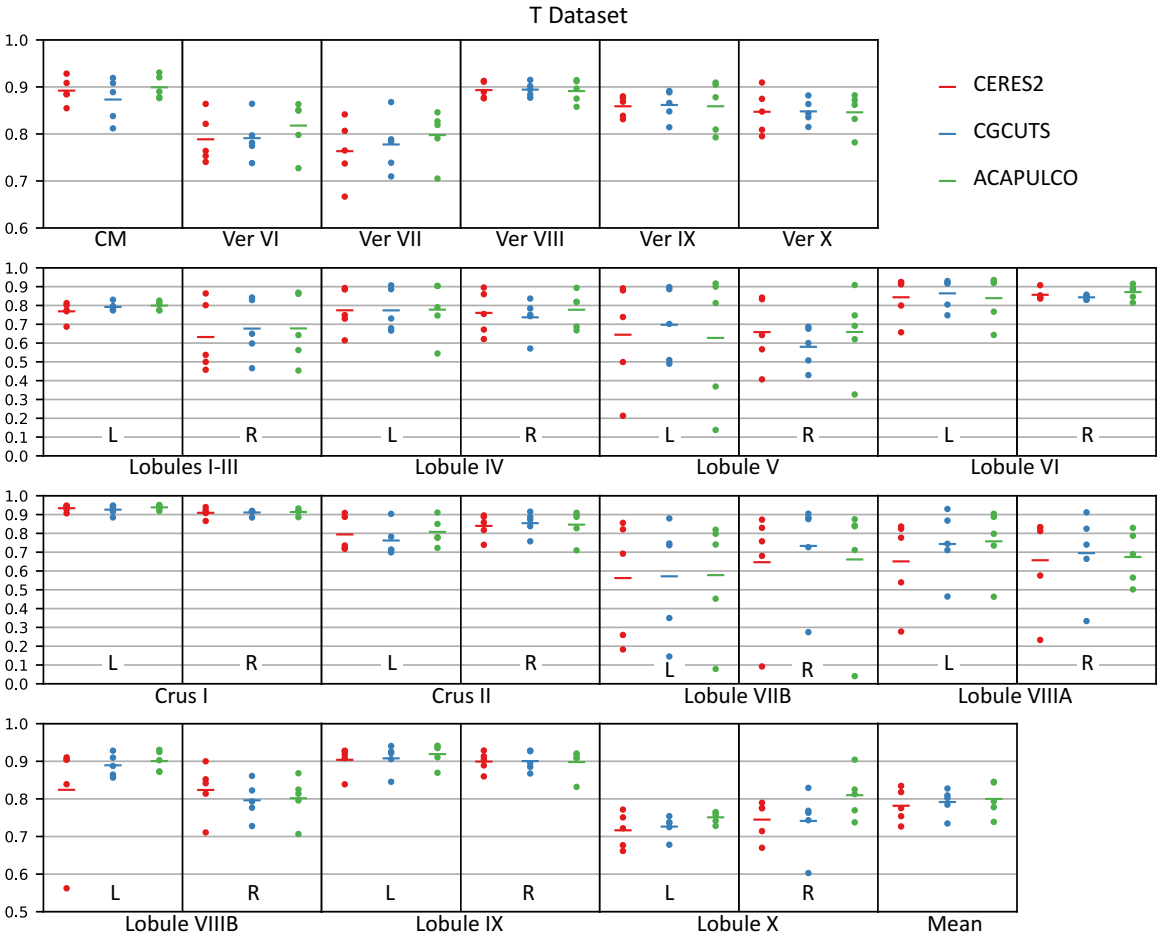


Figure 2-6. Dice coefficients of CERES2, CGCUTS, and ACAPULCO for the T dataset. Vertical axes are Dice coefficients. Dots represent testing images, and bars represent their means. The mean Dice coefficients across all regions for each testing image are shown in the last subfigure. The difference between ACAPULCO and CERES2 is not statistically significant, but ACAPULCO scores the best in terms of the mean Dice coefficients (the bars in subfigures) in 18 out of 28 regions. CM: corpus medullare. Ver: vermal lobule. L: left hemispheric. R: right hemispheric.

Table 2-II. Dice coefficients of CERES2, CGCUTS, and ACAPULCO of the M dataset. The mean values and standard deviations (SDs) of each region are calculated across all testing images. The bottom row shows the average means and the average SDs from all regions. The best means among the three algorithms are highlighted in blue. Five significantly different regions and the average mean across all regions between CERES2 and ACAPULCO are marked by asterisks ($*$: $p < 0.05$, $**$: $p < 0.01$). CM: corpus medullare. Ver: vermal lobule. L: left hemispheric. R: right hemispheric.

	CERES2	**CGCUTS**	**ACAPULCO**
	Mean $\pm$ **SD**	**Mean** $\pm$ **SD**	**Mean** $\pm$ **SD**
CM	0.9380 $\pm$ 0.0162	0.9371 $\pm$ 0.0088	0.9425 $\pm$ 0.0074
Ver I–V	0.8869 $\pm$ 0.0232	0.8648 $\pm$ 0.0238	0.8993 $\pm$ 0.0239
Ver VI–VII	0.8363 $\pm$ 0.0464	0.8221 $\pm$ 0.0352	0.8412 $\pm$ 0.0486
Ver VIII–X	0.9056 $\pm$ 0.0292	0.8905 $\pm$ 0.0419	0.9154 $\pm$ 0.0234
L I–V	0.8884 $\pm$ 0.0564	0.8755 $\pm$ 0.0500	0.8916 $\pm$ 0.0518
R I–V	0.8908 $\pm$ 0.0378	0.8683 $\pm$ 0.0442	0.8922 $\pm$ 0.0504
L VI	0.9065 $\pm$ 0.0329	0.8978 $\pm$ 0.0317	0.9106 $\pm$ 0.0289
R VI	0.9046 $\pm$ 0.0319	0.8954 $\pm$ 0.0375	0.9054 $\pm$ 0.0388
L crus I $**$	0.9285 $\pm$ 0.0205	0.9177 $\pm$ 0.0148	0.9410 $\pm$ 0.0100
R crus I $*$	0.9335 $\pm$ 0.0196	0.9233 $\pm$ 0.0110	0.9469 $\pm$ 0.0054
L crus II & VIIB	0.9134 $\pm$ 0.0278	0.8983 $\pm$ 0.0226	0.9004 $\pm$ 0.0411
R crus II & VIIB	0.9231 $\pm$ 0.0172	0.9117 $\pm$ 0.0181	0.9313 $\pm$ 0.0167
L VIII	0.9078 $\pm$ 0.0301	0.8880 $\pm$ 0.0293	0.8910 $\pm$ 0.0441
R VIII	0.9153 $\pm$ 0.0231	0.9017 $\pm$ 0.0308	0.9211 $\pm$ 0.0250
L IX $*$	0.9245 $\pm$ 0.0196	0.9024 $\pm$ 0.0420	0.9360 $\pm$ 0.0148
R IX	0.9218 $\pm$ 0.0282	0.9097 $\pm$ 0.0380	0.9307 $\pm$ 0.0248
L X $**$	0.8488 $\pm$ 0.0443	0.8317 $\pm$ 0.0349	0.8926 $\pm$ 0.0248
R X $*$	0.8530 $\pm$ 0.0427	0.8287 $\pm$ 0.0399	0.8850 $\pm$ 0.0437
Average $**$	0.9015 $\pm$ 0.0304	0.8889 $\pm$ 0.0308	0.9097 $\pm$ 0.0291

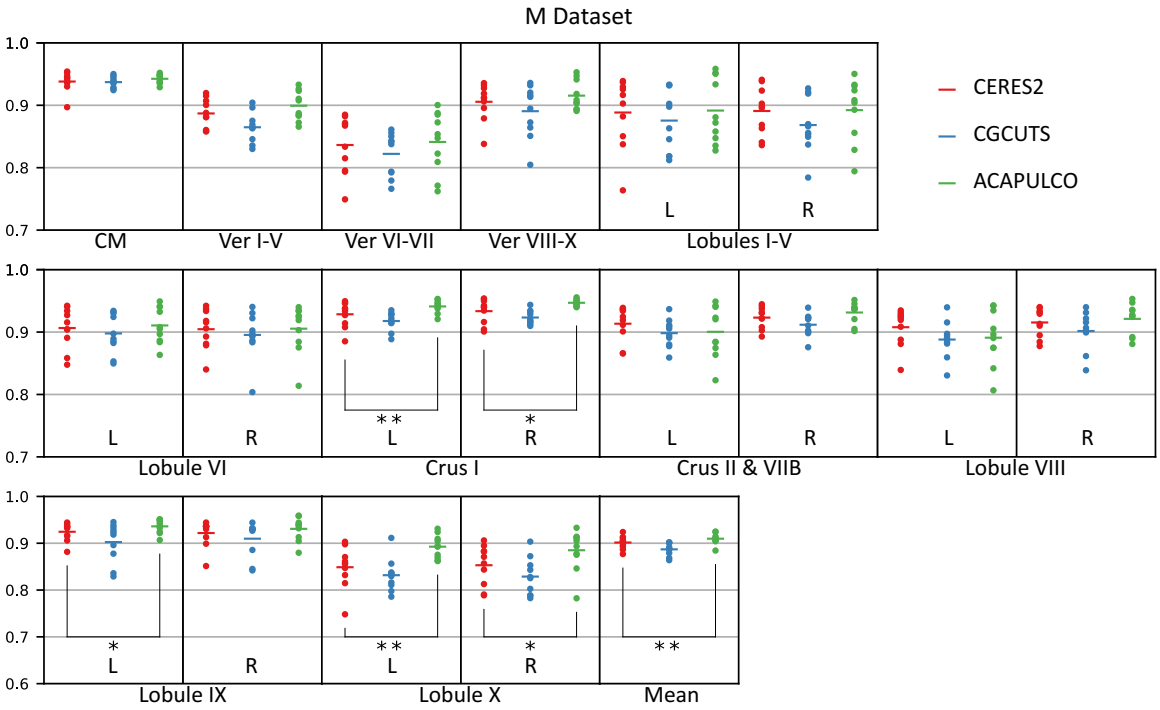


Figure 2-7. Dice coefficients of CERES2, CGCUTS, and ACAPULCO for the M dataset. Vertical axes are Dice coefficients. Dots represent testing images, and bars represent their means. The mean Dice coefficients across all regions for each testing image are shown in the last subfigure. Five significantly different regions and the mean across all regions between CERES2 and ACAPULCO are marked by asterisks ($*$: $p < 0.05$, $**$: $p < 0.01$). ACAPULCO scores the best in terms of the mean Dice coefficients (the bars in subfigures) in 16 out of 18 regions. CM: corpus medullare. Ver: vermal lobule. L: left hemispheric. R: right hemispheric.

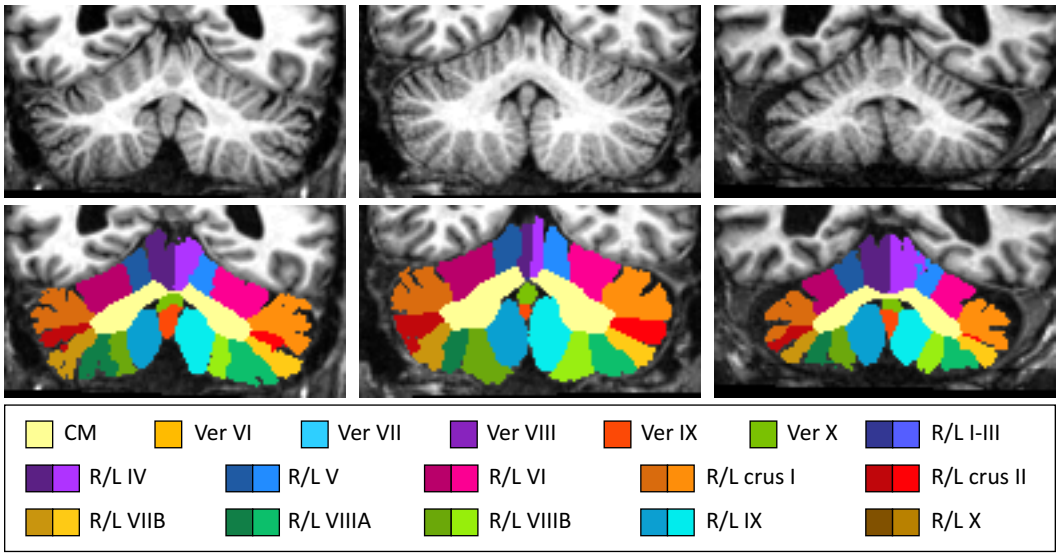


Figure 2-8. Coronal slices of three testing images of the T dataset and their corresponding parcellations from ACAPULCO. CM: corpus medullare. Ver: vermal lobule. R: right hemispheric lobule. L: left hemispheric lobule.

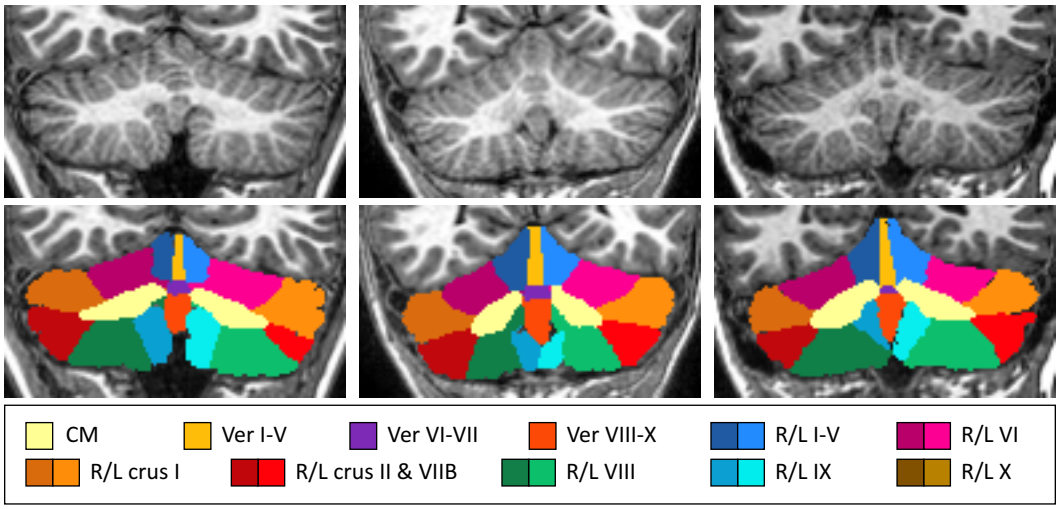


Figure 2-9. Coronal slices of three testing images of the M dataset and their corresponding parcellations from ACAPULCO. CM: corpus medullare. Ver: vermal lobule. R: right hemispheric lobule. L: left hemispheric lobule.

2.3.3 Reproducibility Analysis

The Kirby dataset [9] was used to assess the reproducibility of ACAPULCO. This dataset contains 21 subjects in which each subject had two identical MRI scanning sessions with a short break (out of the scanner) between the scans. The MPRAGE image for each session (42 in total) was parcellated using ACAPULCO trained from the T dataset, and the volumes of the resulting 28 cerebellar regions in each scan were calculated. Based on the one-way random model [94], the intraclass correlation coefficient (ICC) of each volume between the two imaging sessions was calculated (R version 3.5.1 with package `irr` version 0.84) and is shown in Table 2-III with the corresponding 95% confidence interval. ICCs between 0.75 and 0.9 are considered good, and ICCs between 0.90 and 1.00 are considered excellent [95]. We note that our ICCs are all above 0.9, and the lower bounds of the 95% confidence interval are above 0.9 except for the left hemispheric lobule V and right hemispheric lobule X whose smallest value is 0.8894. Example parcellations from the Kirby dataset are shown in Fig. 2-10.

2.3.4 Other Datasets

To show the broad applicability of ACAPULCO, we processed several other datasets[1].

- **Kwyjibo dataset**. We first applied ACAPULCO trained from the T dataset to the Kwyjibo dataset [14]. This dataset contains subjects with various types of ataxia such as spinocerebellar ataxia type 2 (SCA2), spinocerebellar ataxia 3 (SCA3), and spinocerebellar ataxia type 6 (SCA6). A total of 246 images were processed, and we did not find major failures for most of the results. Coronal slices of a healthy subject, an SCA2 subject, an SCA3 subject, and an SCA6 subject are

[1] GNU parallel [96] was used to facilitate the processing.

Table 2-III. Intraclass correlation coefficients (ICCs) of each cerebellar region calculated from the Kirby dataset. All ICCs are above 0.9, which are considered excellent [95]. CM: corpus medullare. Ver: vermal lobule. L: left hemispheric lobule. R: right hemispheric lobule.

	ICC	95% confidence interval	
		Lower bound	Upper bound
CM	0.9962	0.9906	0.9984
Ver VI	0.9863	0.9670	0.9944
Ver VII	0.9827	0.9587	0.9929
Ver VIII	0.9962	0.9906	0.9984
Ver IX	0.9916	0.9794	0.9966
Ver X	0.9708	0.9305	0.9880
L I-III	0.9899	0.9757	0.9959
R I-III	0.9741	0.9365	0.9894
L IV	0.9761	0.9428	0.9902
R IV	0.9861	0.9662	0.9943
L V	0.9535	0.8894	0.9809
R V	0.9685	0.9252	0.9870
L VI	0.9939	0.9852	0.9975
R VI	0.9949	0.9877	0.9979
L crus I	0.9920	0.9807	0.9967
R crus I	0.9913	0.9788	0.9965
L crus II	0.9906	0.9773	0.9961
R crus II	0.9944	0.9862	0.9977
L VIIB	0.9832	0.9596	0.9931
R VIIB	0.9792	0.9498	0.9915
L VIIIA	0.9922	0.9812	0.9968
R VIIIA	0.9895	0.9743	0.9957
L VIIIB	0.9859	0.9660	0.9942
R VIIIB	0.9762	0.9346	0.9908
L IX	0.9927	0.9800	0.9971
R IX	0.9905	0.9770	0.9961
L X	0.9821	0.9569	0.9927
R X	0.9570	0.8980	0.9823

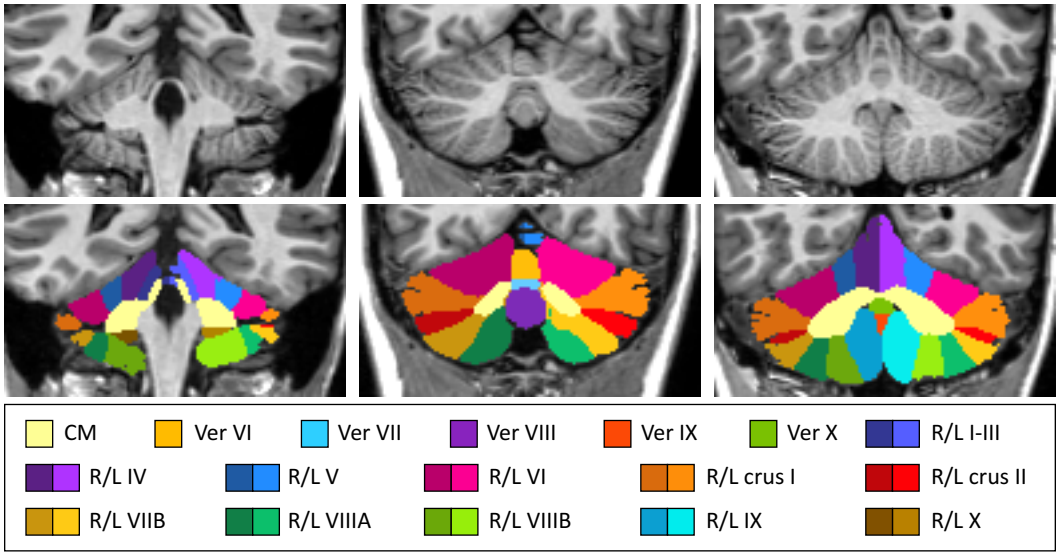


Figure 2-10. Three coronal slices and their corresponding parcellations of a Kirby subject. CM: corpus medullare. Ver: vermal lobule. R: right hemispheric lobule. L: left hemispheric lobule.

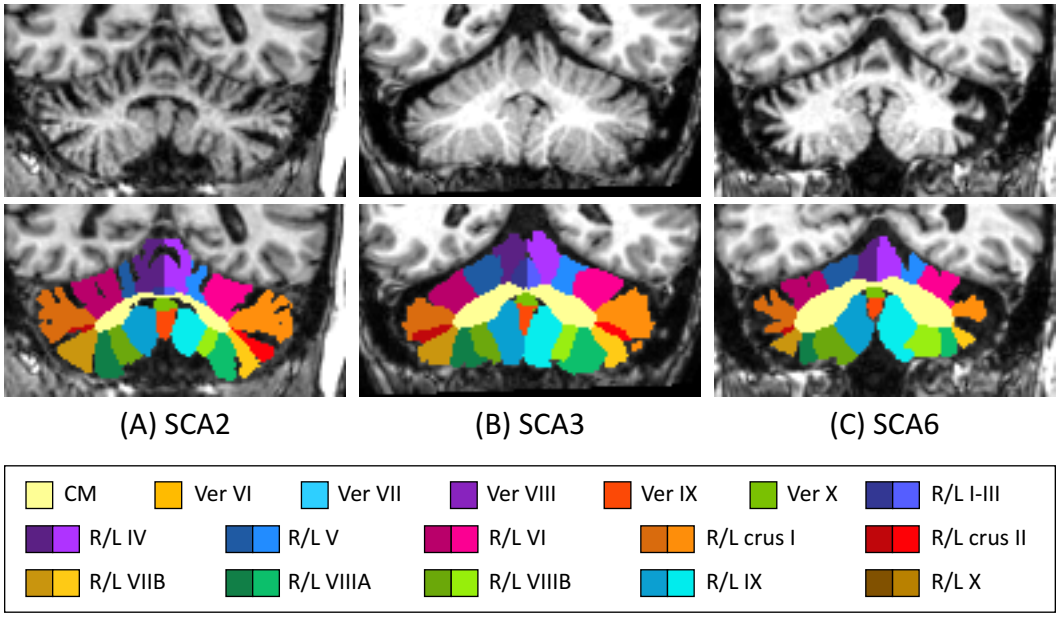
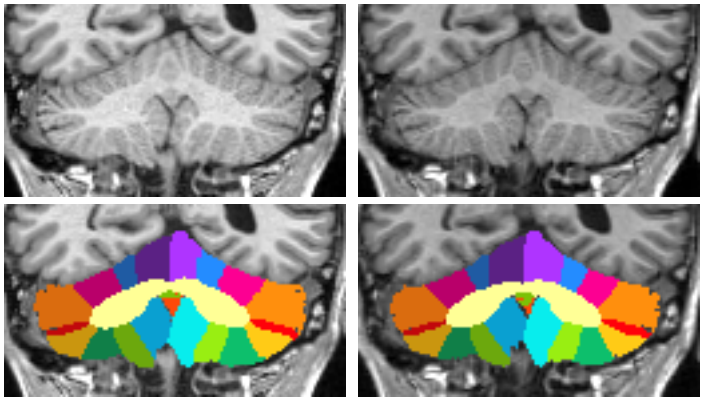
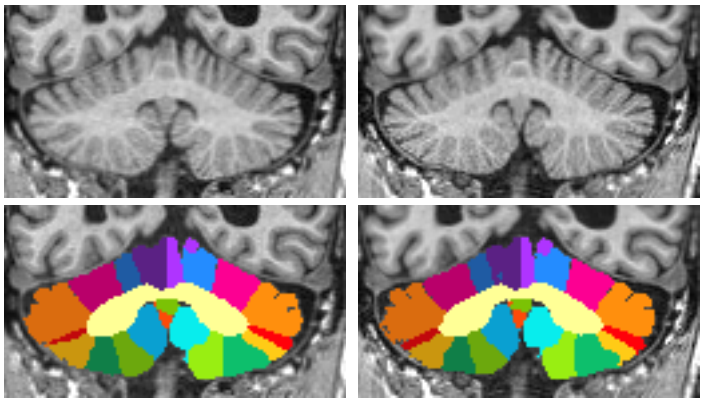


Figure 2-11. Example parcellations of the Kwyjibo dataset. ACAPULCO can parcellate cerebella with atrophy. (A): a spinocerebellar ataxia type 2 (SCA2) subject. (B): a spinocerebellar ataxia 3 (SCA3) subject. (C): a spinocerebellar ataxia type 6 (SCA6) subject. CM: corpus medullare. Ver: vermal lobule. R: right hemispheric lobule. L: left hemispheric lobule.



(A) Healthy subject 495 days apart



(B) AD subject 707 days apart

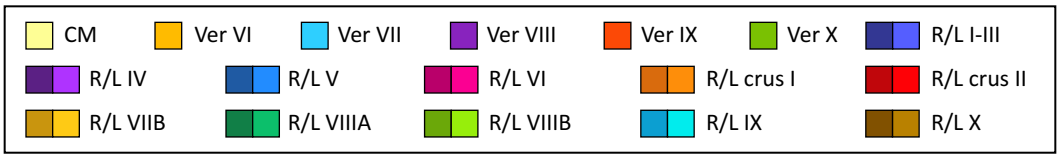


Figure 2-12. Example parcellations of the OASIS-3 dataset. (A): a healthy subject's scans taken 495 days apart. (B): an Alzheimer's disease (AD) subject's scans taken 707 days apart. CM: corpus medullare. Ver: vermal lobule. R: right hemispheric lobule. L: left hemispheric lobule.

shown in Fig. 2-11. Visually, ACAPULCO works well even with severe atrophy as shown in Fig. 2-11(B)–(D).

- **OASIS-3 dataset**. The OASIS-3 dataset [22] contains longitudinal scans of subjects with Alzheimer's disease and healthy subjects. A total of 1,931 MPRAGE

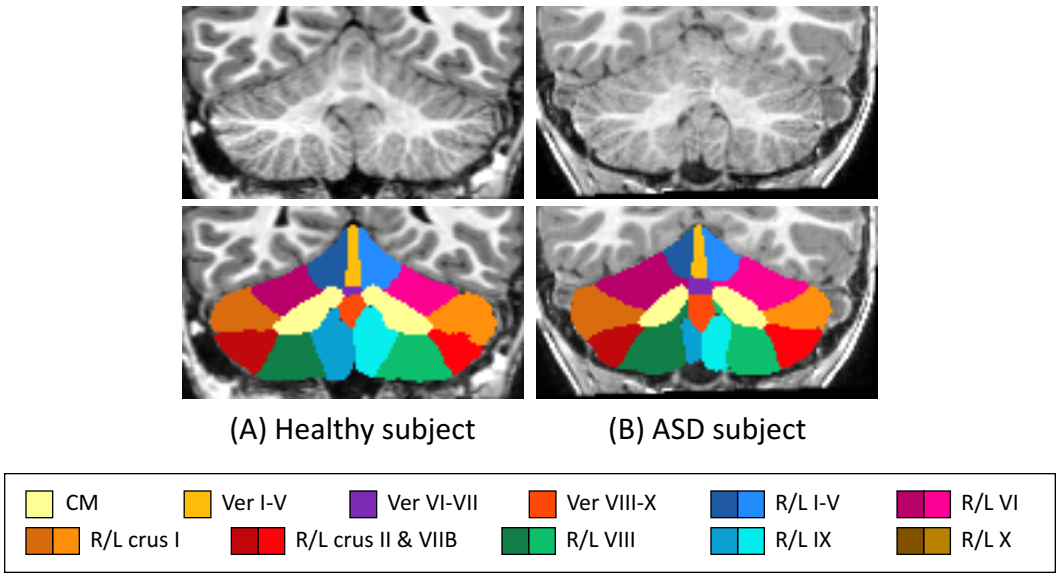


Figure 2-13. Example parcellations of the ABIDEII dataset. (A): a healthy subject. (B): a subject with autism spectrum disorder (ASD). CM: corpus medullare. Ver: vermal lobule. R: right hemispheric lobule. L: left hemispheric lobule.

images were processed by ACAPULCO trained from the T dataset, and we did not find major failures for most of the results. Longitudinal scans of a healthy subject and a subject with Alzheimer's disease are shown in Fig. 2-12.

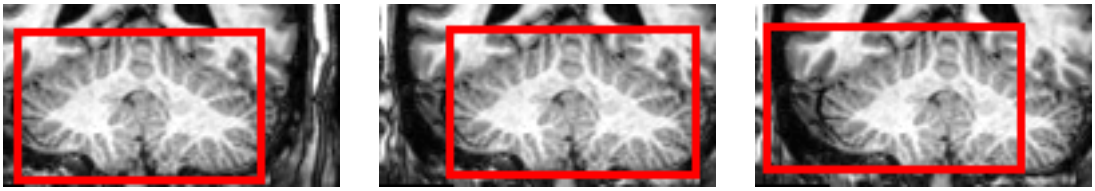
- **ABIDEII dataset**. The ABIDEII dataset [84] contains subjects with autism spectrum disorder (ASD) and healthy subjects. A total of 795 MPRAGE images of subjects younger than 16 years old were processed by ACAPULCO trained from the M dataset in MNI space. For the images without severe noise and artifact, we did not find major failures for most of them. Coronal slices of a healthy subject and three ASD subjects are shown in Fig. 2-13.

2.4 Discussion

Although CNNs have been applied to cerebellum parcellation previously, CERES2, which is based on multi-atlas segmentation, outperformed all other algorithms as reported in Carass *et al.* [19]. To our knowledge, ACAPULCO is the first cerebellum parcellation method based on CNNs to achieve the SOTA results. Although the results of ACAPULCO trained on the T dataset were not different from CERES2 with statistical significance, the mean Dice coefficients across all five testing images are better than the results of CERES2 for a majority of cerebellar regions. We speculate that the lack of a statistical difference may be due to the relatively small number of testing images. Meanwhile, ACAPULCO trained on the M dataset is significantly better than CERES2 for five regions while other regions are comparable. In terms of computational time without pre-processing, ACAPULCO takes 67.67 seconds in average on a CPU and 55.35 seconds in average on a GPU, while CERES takes 212 seconds [43].

Instead of using manually delineated images as atlases, as in previous methods such as SUIT [35], MAGeT [40], RASCAL [42], CERES [43], and CGCUTS [44], we used these images as training data for CNNs to generate learnable features for parcellation, i.e., voxelwise classification. There are several factors contributing to the better performance of ACAPULCO. First, to incorporate the 3D information as much as possible, we constructed 3D networks and used the entire region of the cerebellum as input to the parcellating network, and the receptive field of the deepest convolution can cover the whole image. The large number of convolution channels in our networks can also help to learn complex features.

Second, although batch normalization is a common practice in CNNs, we used instance normalization instead. Wu & He [97] showed that the performance of batch normalization is worse with a smaller mini-batch size. In our work, since the whole

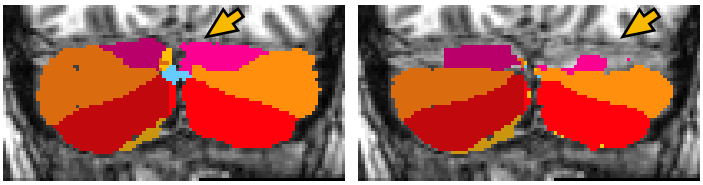


(A) Before translation (B) After translation *with* augmentation (C) After translation *without* augmentation

Figure 2-14. Comparison between bounding boxes predicted from the locating network that is trained with and without the translation augmentation. (A) shows the prediction before translating the image. (B) and (C) show predictions from the locating network that is trained with and without translation augmentation, respectively, after translating the image. Note that the network in (C) fails to move the bounding box accordingly.

3D image and the whole region around the cerebellum were used as inputs to our networks, only one or two samples could be included in a mini-batch while maintaining the capability of these networks due to the GPU memory constraint. In this case, the small size of our mini-batches could be a reason that instance normalization was beneficial in our case. Since Wu & He [97] show that group normalization outperforms batch normalization for the task of object detection and segmentation, our future work will include evaluating the effect of group normalization on cerebellum parcellation.

Third, although the parcellating network performs voxelwise classification and each voxel can be regarded as one training sample, these voxels are highly correlated. Therefore, data augmentation plays a crucial role in training. The locating network is not inherently translation-invariant, and we show in Fig. 2-14 that the network trained without translation augmentation cannot shift the bounding box when the image is translated. In this case, we suspect that this network simply remembered the average location of the bounding boxes across the training images. In Fig. 2-15, we show that the parcellating network trained without scaling augmentation fails to label part of a large cerebellum correctly. According to our preliminary experiments, all data augmentation methods that we used increase the inference accuracy.



(A) *With* scaling augmentation

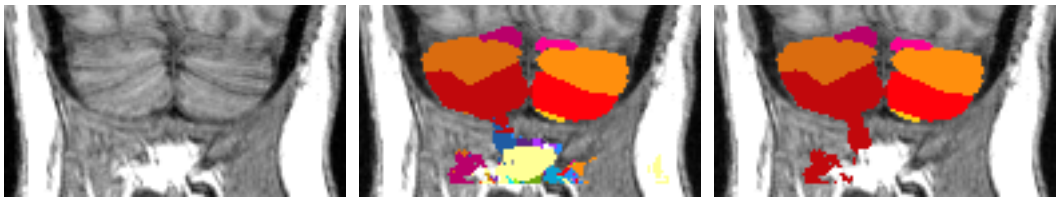
(B) *Without* scaling augmentation

Figure 2-15. Comparison between the results from the parcellating network that is trained (A) with and (B) without the scaling augmentation. Note that the network in (B) fails to label part of the cerebellum as indicated by the yellow arrows.

Fourth, the post-processing imposes some topological knowledge to the parcellations to correct isolated mislabeling as shown in Fig. 2-4. Although we did not observe such mistakes in parcellations of the testing images from the T and M datasets, the isolated mislabeling does occur in some images from other datasets mostly around the neck area. Although these datasets do not have manual delineations to numerically evaluate the post-processing, qualitatively it can produce more robust parcellations.

To incorporate the whole cerebellum in the input to our parcellating network while maintaining adequate network capability, we needed to crop the image around the cerebellum. Without using the locating network, we would have to find an alternative such as using a cerebellum mask in MNI space. However, we note that since each cerebellum can have a different shape and size, it would not necessarily be at the center of such a mask. Additionally, MNI registration can still have failure cases where the MNI cerebellum mask would not be correct.

A potential criticism of ACAPULCO is that its bounding boxes are expanded symmetrically in all six cardinal directions to a fixed size for the T and M datasets. We could alternatively use the output bounding boxes and resample the data appropriately. This, however, might necessitate expanded ground truth bounding boxes during the training of the locating network to improve robustness. Using such an approach can



(A) Testing image (B) CNN output (C) After post-processing

Figure 2-16. Mislabeling the neck as part of the cerebellum. (A): a testing image from the OASIS-3 dataset. (B): output of the CNNs trained with the T dataset. (C): the post-processing result. The field of views of the training images in the T dataset do not cover the neck. Note that the image in (A) contains the neck, and the CNNs fails to classify it as non-cerebellar in (B). Since the mislabeling is connected to the cerebellum, our post-processing cannot remove it in (C).

still run into three problems: (1) the generated bounding boxes may be too tight (even with expanded ground truth bounding boxes) and thus risk removing portions of the cerebellum; (2) resampling the data may lead to issues when restoring the parcellated labels to the original (un-resampled) coordinate space which is required at the end of the process; and (3) resampled voxels would have different resolutions, which may degrade network performance. Our approach circumvents these issues without any obvious drawbacks.

There are several limitations in ACAPULCO. Because of the manual delineation used in the T and M datasets, the corpus medullare in our work covers the main body of the WM and does not represent all of the WM. We believe that using training images with higher resolution and finer delineation of the WM can lead to better prediction of WM boundary. We have also observed that optimization of the networks can be heavily affected by the training data, i.e., easily overfit. For example, the field of view of the images in the T dataset (the image is cut off below the cerebellum as shown in Fig. 1-8(D) and (E)) does not cover the neck. As a result, it is possible that the neck in some images are classified as part of the cerebellum (see Fig. 2-16). This is an error that our post-processing cannot currently resolve, as the mislabeling sometimes

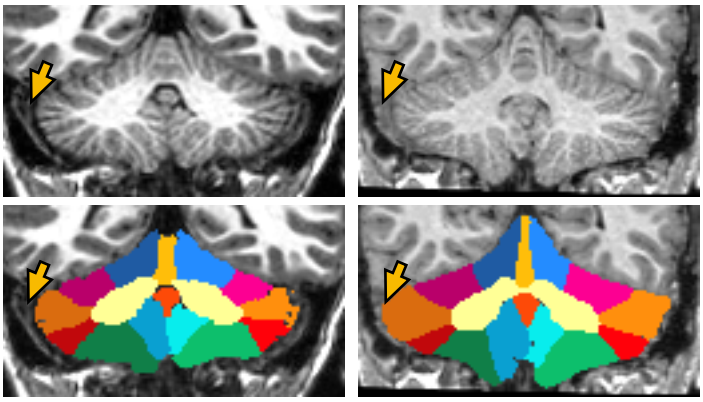


Figure 2-17. Oversegmentation when the sinus is bright. (A): a manually delineated image. (B): the parcellation of another image in the ABIDEII dataset. The yellow arrows point to the sinus.

is connected to the cerebellum as shown in Fig. 2-16. Additionally, the networks do not generalize well to images with different contrasts. For example, the sinuses on the left and right sides of the cerebellum in Fig. 2-17(B) have a brighter signal than the training image in Fig. 2-17(A). This results in a portion of the sinus in Fig. 2-17(B) being classified as part of the cerebellum. We address this problem using T2w images in Chapter 5.

2.5 Summary

In this chapter, we presented ACAPULCO, a DL algorithm to parcellate the cerebellum. ACAPULCO was compared to previous methods using two datasets, and it achieved the SOTA performance. We published both Singularity and Docker containers of ACAPULCO, and they have been used worldwide at this point.

Chapter 3

Incorporating Anatomical Knowledge into Network Architectures

In this chapter, we explores incorporating anatomical knowledge into the CNN architectures. Two properties are covered: left-right symmetry of the human brain and hierarchical definition of the cerebellum.

3.1 Incorporating Left-Right Symmetry into the Network Architecture

3.1.1 Introduction

Since human brains are approximately left-right symmetric, reflection can be used for augmenting training data to improve the inference accuracy [56, 98]. However, this symmetry information has never before been directly incorporated into a CNN architecture for a better segmentation.

A function is equivariant when this function commutes with the transformation applied to its input. In other words, if the input is transformed, the output will be transformed in the same way. Unlike translation, equivariance with respect to left-right reflection is not guaranteed by convolutions although reflection is commonly used in data augmentation. Cohen & Welling [66] proposed group convolutions to introduce (into CNNs) equivariance with respect to rotation with discrete angles and reflection around each axis. Group convolutions have been applied to 2D lymph node tumor segmentation [99], 3D pulmonary nodule segmentation [100], and 2D brain tumor segmentation [101] to achieve improved performance.

Brain MRI images are usually brought into MNI space so that they have a relatively fixed orientation. Therefore, in contrast to previous studies, we investigated equivariance with respect to only left-right reflection for 3D CNNs to segment brain MRI images. We also extended group convolutions to tasks of segmentation where *paired regions*, such as the left and right hippocampus, are delineated. We show that the proposed reflection-equivariant (RE) CNNs have better performance in several tasks—i.e., skull stripping, brain tissue segmentation, subcortical structure segmentation, and cerebellum parcellation—compared with conventional CNNs trained with left-right reflection augmentation.

3.1.2 Methods

For a multi-channel function $\boldsymbol{f}$ of spatial location $\boldsymbol{x} \in \mathbb{R}^3$, e.g., a multi-channel 3D image or a set of feature maps, suppose R is left-right reflection, and $R\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{f}(R^{-1}\boldsymbol{x}) = \boldsymbol{f}(R\boldsymbol{x})$ where R^{-1} is the inverse of R and $R^{-1} = R$ (the inverse of a left-right reflection is still a left-right reflection). In the following discussion, we use $\boldsymbol{f}^{(0)}$ to indicate the multi-channel input image, and use $\boldsymbol{f}^{(1)}$, $\boldsymbol{f}^{(2)}$, ..., $\boldsymbol{f}^{(l)}$, ..., $\boldsymbol{f}^{(L)}$ to indicate the feature

maps produced by a series of L convolutional layers. We use a subscript to denote an individual channel of such a multi-channel function; for example, the k^{th} channel of $\boldsymbol{f}^{(l)}$ is denoted as $f_k^{(l)}$. We use $\boldsymbol{\phi}^{(1)}$, $\boldsymbol{\phi}^{(2)}$, ..., $\boldsymbol{\phi}^{(l)}$, ..., $\boldsymbol{\phi}^{(L)}$ to indicate kernels of these L convolutional layers. We use subscripts to denote the corresponding input and output channels; for example, $\phi_{r,s}^{(l)}$ corresponds to the r^{th} input channel and the s^{th} output channel for $\boldsymbol{\phi}^{(l)}$. The bias vectors of these convolutional layers are represented as $\boldsymbol{b}^{(1)}$, $\boldsymbol{b}^{(2)}$, ..., $\boldsymbol{b}^{(l)}$, ..., $\boldsymbol{b}^{(L)}$ where the k^{th} element of a bias vector $\boldsymbol{b}^{(l)}$ is represented as $b_k^{(l)}$.

There are three types of RE convolutions [99] to consider in a CNN: 1) an image to feature maps, 2) feature maps to feature maps, and 3) feature maps to a segmentation. To convert an image $\boldsymbol{f}^{(0)}$ into feature maps $\boldsymbol{f}^{(1)}$, we use

$$\begin{aligned} f_{2k_1}^{(1)} &= \sum_{k_0=0}^{K_0-1} f_{k_0}^{(0)} \star \phi_{k_0,k_1}^{(1)} + b_{k_1}^{(1)}, \\ f_{2k_1+1}^{(1)} &= \sum_{k_0=0}^{K_0-1} f_{k_0}^{(0)} \star R\phi_{k_0,k_1}^{(1)} + b_{k_1}^{(1)}, \end{aligned} \tag{3.1}$$

where K_0 and $2K_1$ are the numbers of channels of $\boldsymbol{f}^{(0)}$ and $\boldsymbol{f}^{(1)}$, respectively, and $\star$ indicates cross-correlation (note that cross-correlation instead of convolution is performed in a convolutional layer in many DL libraries such as PyTorch). To convert feature maps $\boldsymbol{f}^{(l-1)}$ into feature maps $\boldsymbol{f}^{(l)}$, we use

$$\begin{aligned} f_{2k_l}^{(l)} &= \sum_{k_{l-1}=0}^{K_{l-1}-1} \left(f_{2k_{l-1}}^{(l-1)} \star \phi_{2k_{l-1},k_l}^{(l)} + f_{2k_{l-1}+1}^{(l-1)} \star \phi_{2k_{l-1}+1,k_l}^{(l)} \right) + b_{k_l}^{(l)}, \\ f_{2k_l+1}^{(l)} &= \sum_{k_{l-1}=0}^{K_{l-1}-1} \left(f_{2k_{l-1}}^{(l-1)} \star R\phi_{2k_{l-1}+1,k_l}^{(l)} + f_{2k_{l-1}+1}^{(l-1)} \star R\phi_{2k_{l-1},k_l}^{(l)} \right) + b_{k_l}^{(l)}, \end{aligned} \tag{3.2}$$

where $2K_{l-1}$ and $2K_l$ are the numbers of channels of $\boldsymbol{f}^{(l-1)}$ and $\boldsymbol{f}^{(l)}$, respectively.

Next we prove that if the reflected image $R\boldsymbol{f}^{(0)}$ is used as input, the L^{th} convolution

outputs $Rf^{(L)}_{2k_L+1}$ and $Rf^{(L)}_{2k_L}$, which are both reflected—i.e., the reflection R is applied—and swapped—i.e., the output is in the order of $2k_l+1$ and $2k_l$. First, we use the reflected image $R\boldsymbol{f}^{(0)}(\boldsymbol{x}) = \boldsymbol{f}^{(0)}(R\boldsymbol{x})$ as input in Eq. (3.1). Suppose $\boldsymbol{x}' = R\boldsymbol{x}$. Recall that $R = R^{-1}$, so $\boldsymbol{x} = R\boldsymbol{x}'$. According to Eq. (3.1), we have

$$\begin{aligned}
\sum_{k_0=0}^{K_0-1} f^{(0)}_{k_0}(R\boldsymbol{x}) \star \phi^{(1)}_{k_0,k_1}(\boldsymbol{x}) + b^{(1)}_{k_1} &= \sum_{k_0=0}^{K_0-1} f^{(0)}_{k_0}(\boldsymbol{x}') \star \phi^{(1)}_{k_0,k_1}(R\boldsymbol{x}') + b^{(1)}_{k_1} \\
&= f^{(1)}_{2k_1+1}(\boldsymbol{x}') = f^{(1)}_{2k_1+1}(R\boldsymbol{x}), \\
\sum_{k_0=0}^{K_0-1} f^{(0)}_{k_0}(R\boldsymbol{x}) \star \phi^{(1)}_{k_0,k_1}(R\boldsymbol{x}) + b^{(1)}_{k_1} &= \sum_{k_0=0}^{K_0-1} f^{(0)}_{k_0}(\boldsymbol{x}') \star \phi^{(1)}_{k_0,k_1}(\boldsymbol{x}') + b^{(1)}_{k_1} \\
&= f^{(1)}_{2k_1}(\boldsymbol{x}') = f^{(1)}_{2k_1}(R\boldsymbol{x}),
\end{aligned} \tag{3.3}$$

where the output feature maps are both reflected and swapped. In Eq. (3.2), suppose that we use the reflected and swapped feature maps $f^{(l-1)}_{2k_{l-1}+1}(R\boldsymbol{x})$ and $f^{(l-1)}_{2k_{l-1}}(R\boldsymbol{x})$ as input; then we have

$$\begin{aligned}
&\sum_{k_{l-1}=0}^{K_{l-1}-1} \left(f^{(l-1)}_{2k_{l-1}+1}(R\boldsymbol{x}) \star \phi^{(l)}_{2k_{l-1},k_l}(\boldsymbol{x}) + f^{(l-1)}_{2k_{l-1}}(R\boldsymbol{x}) \star \phi^{(l)}_{2k_{l-1}+1,k_l}(\boldsymbol{x}) \right) + b^{(l)}_{k_l} \\
=&\sum_{k_{l-1}=0}^{K_{l-1}-1} \left(f^{(l-1)}_{2k_{l-1}+1}(\boldsymbol{x}') \star \phi^{(l)}_{2k_{l-1},k_l}(R\boldsymbol{x}') + f^{(l-1)}_{2k_{l-1}}(\boldsymbol{x}') \star \phi^{(l)}_{2k_{l-1}+1,k_l}(R\boldsymbol{x}') \right) + b^{(l)}_{k_l} \\
=&f^{(l)}_{2k_l+1}(\boldsymbol{x}') = f^{(l)}_{2k_l+1}(R\boldsymbol{x}), \\
&\sum_{k_{l-1}=0}^{K_{l-1}-1} \left(f^{(l-1)}_{2k_{l-1}+1}(R\boldsymbol{x}) \star \phi^{(l)}_{2k_{l-1}+1,k_l}(R\boldsymbol{x}) + f^{(l-1)}_{2k_{l-1}}(R\boldsymbol{x}) \star \phi^{(l)}_{2k_{l-1}}(R\boldsymbol{x}) \right) + b^{(l)}_{k_l} \\
=&\sum_{k_{l-1}=0}^{K_{l-1}-1} \left(f^{(l-1)}_{2k_{l-1}+1}(\boldsymbol{x}') \star \phi^{(l)}_{2k_{l-1}+1,k_l}(\boldsymbol{x}') + f^{(l-1)}_{2k_{l-1}}(\boldsymbol{x}') \star \phi^{(l)}_{2k_{l-1},k_l}(\boldsymbol{x}') \right) + b^{(l)}_{k_l} \\
=&f^{(l)}_{2k_l}(\boldsymbol{x}') = f^{(l)}_{2k_l}(R\boldsymbol{x}),
\end{aligned} \tag{3.4}$$

where the output is also reflected and swapped. Therefore, by mathematical induction, if

the input image $\boldsymbol{f}^{(0)}$ is reflected, the channels of the output feature maps $\boldsymbol{f}^{(L)}$ are both reflected and swapped.

To convert feature maps into a segmentation without left-right paired labels (such as in skull stripping and brain tissue segmentation), the group convolution needs to sum up the corresponding $f^{(L)}_{2k_L}$ and $f^{(L)}_{2k_L+1}$ for each k_L after the final convolutional layer. To account for left-right paired labels in a segmentation, we modified the group convolution to only sum up the channels for each of the labels without pairs and to use $f^{(L)}_{2k_L}$ and $f^{(L)}_{2k_L+1}$ for the paired left and right labels, respectively. By doing so, when the input image is reflected, segmentations without pairs will simply be reflected, and paired segmentations will be both reflected and swapped (e.g., a right region is reflected to the left side and is assigned with its corresponding left label); thus, equivariance is guaranteed.

Since the ReLU is a pointwise operation, it does not affect the RE property. However, batch [49] and instance normalization [86] do affect the RE property and should be modified. Specifically, the paired channels $f^{(l)}_{2k_l}$ and $f^{(l)}_{2k_l+1}$ should use the same weight and bias. In batch normalization, the paired channels should also use the same running mean and variance. In contrast, since the mean and variance in instance normalization are calculated from each individual feature map, they need not be modified. Spatial dropout [48] zeroes out random channels during training, but it uses all channels (which are scaled with the dropout probability) during inference; thus, it maintains the RE property. Feature concatenation, pooling, and upsampling, as in the U-Net [91], also maintain the RE property.

3.1.3 Experiments and Results

We modified the 3D U-Net [91] and incorporated RE convolutions. We used instance instead of batch normalization and spatial dropout with probability 0.2 after each ReLU. We used Adam [55] as the optimizer with learning rate $= 1 \times 10^{-3}$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 1 \times 10^{-8}$. We used Eq. (2.2) as the loss function. We used random rotation, scaling, and deformation to augment the training data. To evaluate the RE U-Net, we compared it with the conventional U-Net trained with left-right reflection data augmentation. If training with reflection, the original as well as the left-right reflected images were used as input. We then applied the trained networks to the original and the reflected testing images to evaluate their performance. Two-sided paired Wilcoxon tests were performed between the mean Dice coefficients of all labels (including background) of these two networks.

3.1.3.1 Skull Stripping

We used NFBS [102], a manually corrected human skull-stripping dataset of 125 T1w MRI images, in this experiment. The images were inhomogeneity-corrected using N4 [27], registered to the 1 mm isotropic ICBM 2009c template [31] in MNI space, and then cropped or zero-padded to 192 $\times$ 256 $\times$ 192 voxels. We randomly selected 62 and 63 images as training and testing data, respectively.

The RE and the conventional U-Nets both have 6 pooling layers. The RE U-Net has 4 pairs of channels before the first pooling layer and has 8,936,343 trainable parameters. The conventional U-Net has 6 channels before the first pooling layer and has 10,054,636 trainable parameters. The RE U-Net was trained for 200 epochs while the conventional U-Net was trained for 100 epochs since it took both the original and reflected images as training data during each epoch. The batch size was 1. The average

Table 3-I. Dice coefficients (mean $\pm$ standard deviation) and p-values from paired Wilcoxon tests. In all experiments, both conventional and RE U-Net were tested with the original and the reflected testing images. The better mean Dice coefficients between the two networks are highlighted in blue. The RE U-Net is significantly better ($p < 0.01$) than the conventional U-Net trained with reflection augmentation in the first three experiments. Although not significantly better, the RE U-Net has better mean Dice coefficients in the last experiment.

Experiment	Image	U-Net	RE U-Net	p-value
Skull stripping	**Original**	0.9840 $\pm$ 0.0028	0.9858 $\pm$ 0.0050	8×10^{-8}
	Reflected	0.9840 $\pm$ 0.0027	0.9858 $\pm$ 0.0050	2×10^{-7}
	Both	0.9840 $\pm$ 0.0027	0.9858 $\pm$ 0.0050	7×10^{-14}
Tissue segmentation	**Original**	0.9357 $\pm$ 0.0093	0.9405 $\pm$ 0.0100	6×10^{-6}
	Reflected	0.9356 $\pm$ 0.0093	0.9406 $\pm$ 0.0100	6×10^{-6}
	Both	0.9357 $\pm$ 0.0092	0.9406 $\pm$ 0.0099	3×10^{-11}
Subcortical segmentation	**Original**	0.8821 $\pm$ 0.0241	0.8851 $\pm$ 0.0240	9×10^{-4}
	Reflected	0.8818 $\pm$ 0.0246	0.8852 $\pm$ 0.0241	2×10^{-3}
	Both	0.8819 $\pm$ 0.0240	0.8851 $\pm$ 0.0237	3×10^{-6}
Cerebellum parcellation	**Original**	0.8324 $\pm$ 0.0115	0.8311 $\pm$ 0.0126	0.8125
	Reflected	0.8283 $\pm$ 0.0137	0.8321 $\pm$ 0.0122	0.4375
	Both	0.8304 $\pm$ 0.0121	0.8316 $\pm$ 0.0117	0.6250

Dice coefficients (see Eq. 2.8) of all labels (including background) against the ground truth for both networks and the p-values between them are shown in Table 3-I. We see that the RE U-Net is significantly better ($p < 0.01$). Example results are shown in Fig. 3-1.

3.1.3.2 Brain Tissue Segmentation

The dataset from Landman & Warfield [103] was used in this experiment. This dataset contains 15 and 20 T1w brain MRI images with 1-mm isotropic resolution for training and testing, respectively. These images were manually delineated into more than 130 regions. For tissue segmentation, we combined these regions into three classes

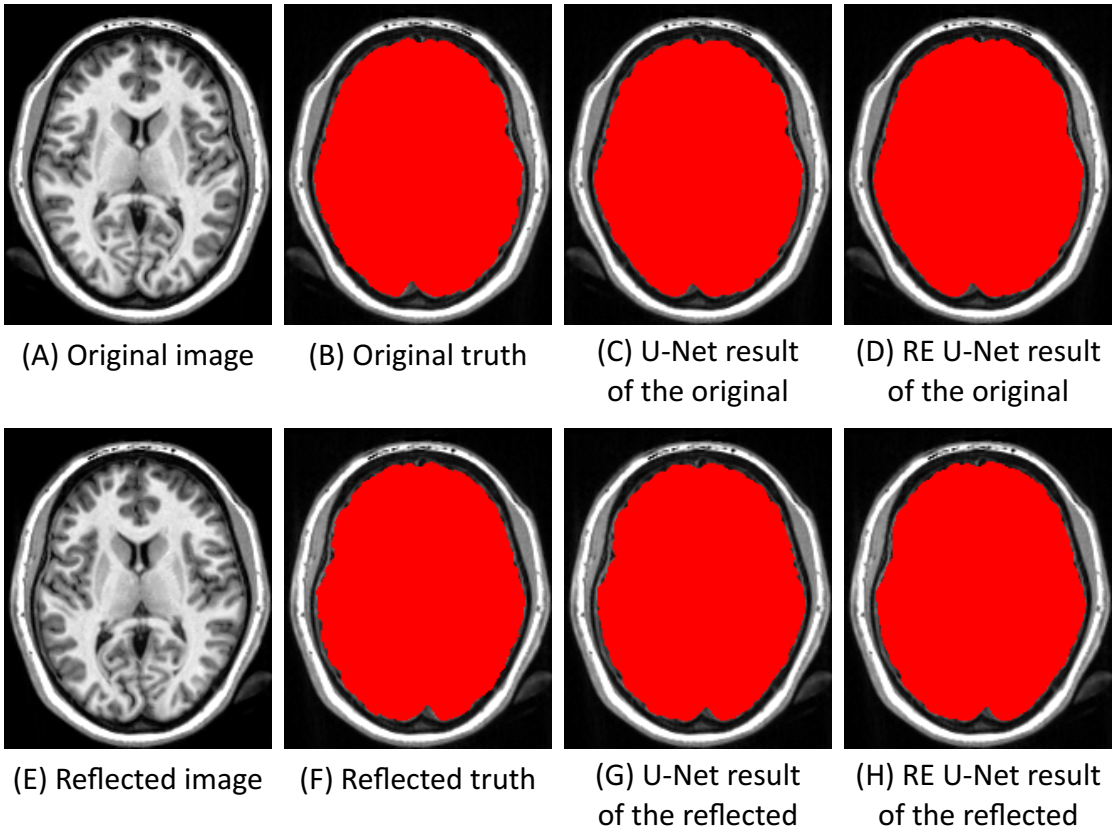


Figure 3-1. Example skull stripping results. (A), (E): original and reflected testing images. (B), (F): true segmentations (manual delineations). (C), (G): results of the conventional U-Net trained with reflection augmentation. (D), (H): results of the RE U-Net.

of GM, WM, and CSF. No inhomogeneity or MNI registration were performed (these images have already had roughly the same orientations). The images were cropped into 160 $\times$ 160 $\times$ 192 voxels around brain masks that were predicted by ROBEX [33].

The RE and the conventional U-Nets both have 5 pooling layers. The RE U-Net has 4 pairs of channels before the first pooling layer and has 2,233,650 trainable parameters. The conventional U-Net has 6 channels before the first pooling layer and has 2,513,473 trainable parameters. The RE U-Net was trained for 2,000 epochs while the conventional U-Net was trained for 1,000 epochs since it took both the original and reflected images as training data during each epoch. The batch size was 3. The average Dice coefficients

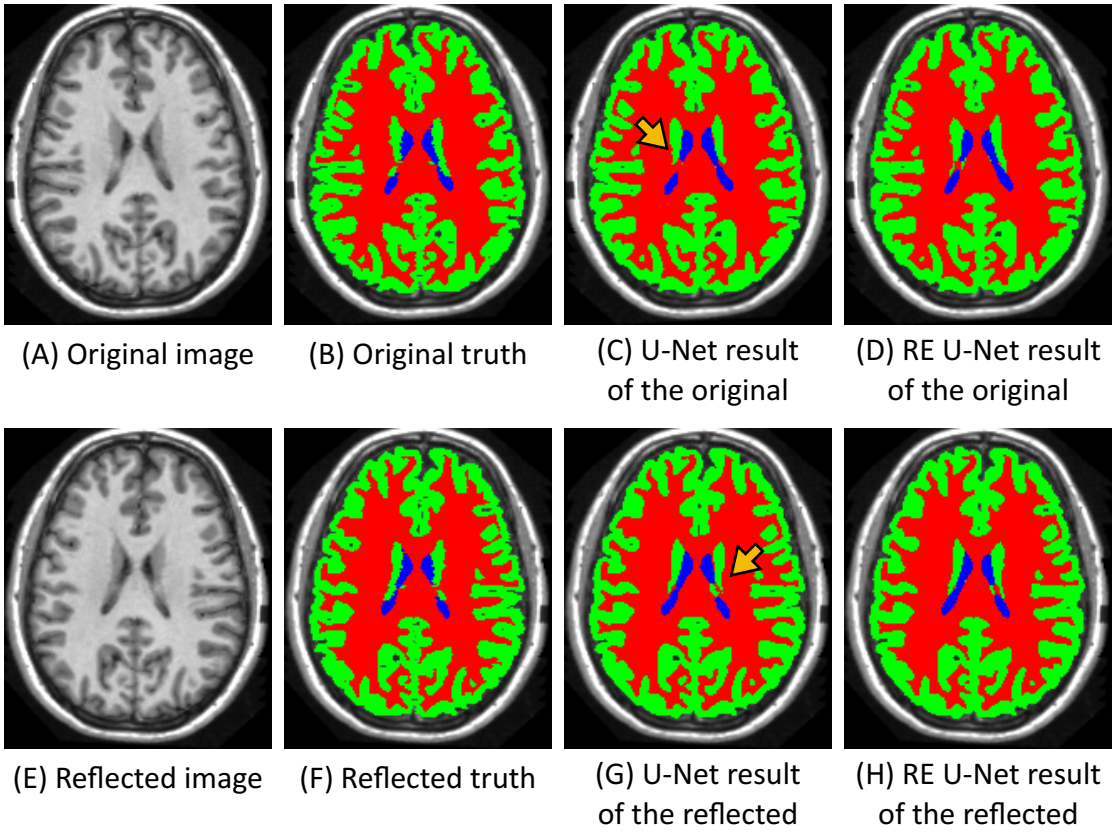


Figure 3-2. Example brain tissue segmentations. (A), (E): original and reflected testing images. (B), (F): true segmentations (manual delineations). (C), (G): results of the conventional U-Net trained with reflection augmentation. (D), (H): results of the RE U-Net. Yellow arrows point to some inconsistency of the results of the conventional U-Net.

of all labels (including background) against the ground truth for both networks and the p-values between them are shown in Table 3-I. We see that the RE U-Net is significantly better ($p < 0.01$). Example segmentations are shown in Fig. 3-2.

3.1.3.3 Subcortical Structure Segmentation

The same dataset as in Section 3.1.3.2 was used. Seven pairs of subcortical structures were extracted from the delineations: left and right thalamus, caudate, putamen, pallidum, hippocampus, amygdala, and accumbens. The images were cropped into 96 $\times$

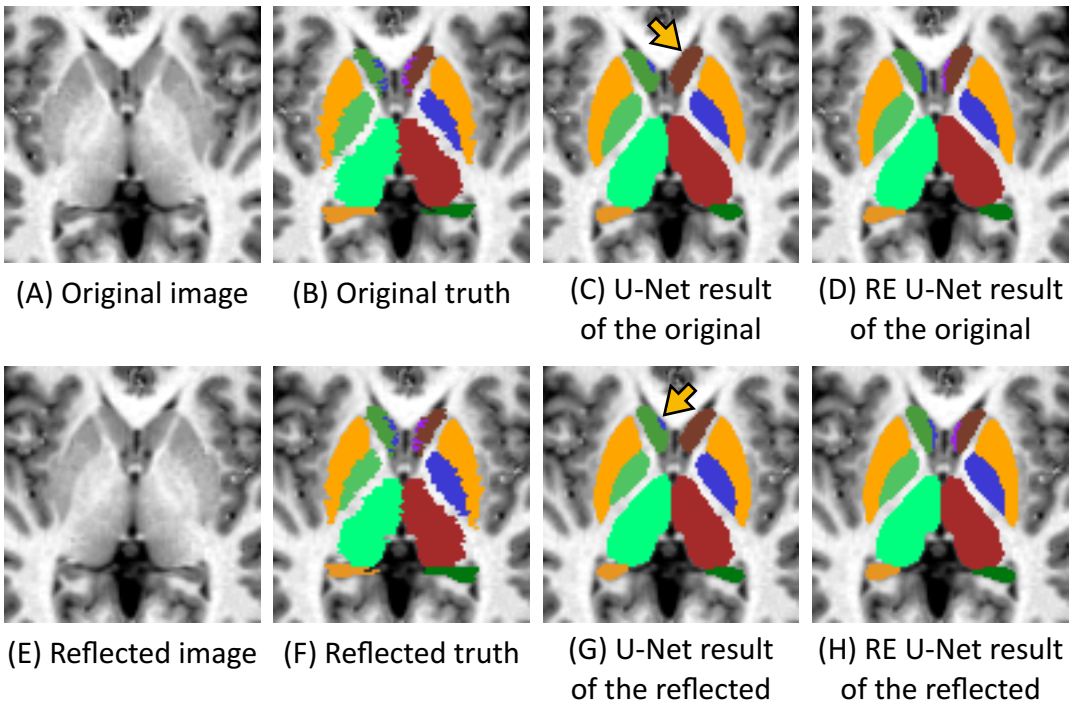


Figure 3-3. Example subcortical structure segmentations. (A), (E): original and reflected testing images. (B), (F): true segmentations (manual delineations). (C), (G): results of the conventional U-Net trained with reflection augmentation. (D), (H): results of the RE U-Net. Yellow arrows point to some inconsistency of the results of the conventional U-Net.

96×96 voxels around these structures.

The RE and the conventional U-Nets both have 5 pooling layers. The RE U-Net has 24 pairs of channels before the first pooling layer and has 80,360,156 training parameters. The conventional U-Net has 35 channels before the first pooling layer and has 81,171,399 trainable parameters. The RE U-Net was trained for 1,000 epochs while the conventional U-Net was trained for 500 epochs since it took both the original and reflected images as training data during each epoch. The batch size was 3. The average Dice coefficients of all labels (including background) against the ground truth for both networks and the p-values between them are shown in Table 3-I. We see that the RE U-Net is significantly better ($p < 0.01$). Example segmentations are shown in Fig. 3-3.

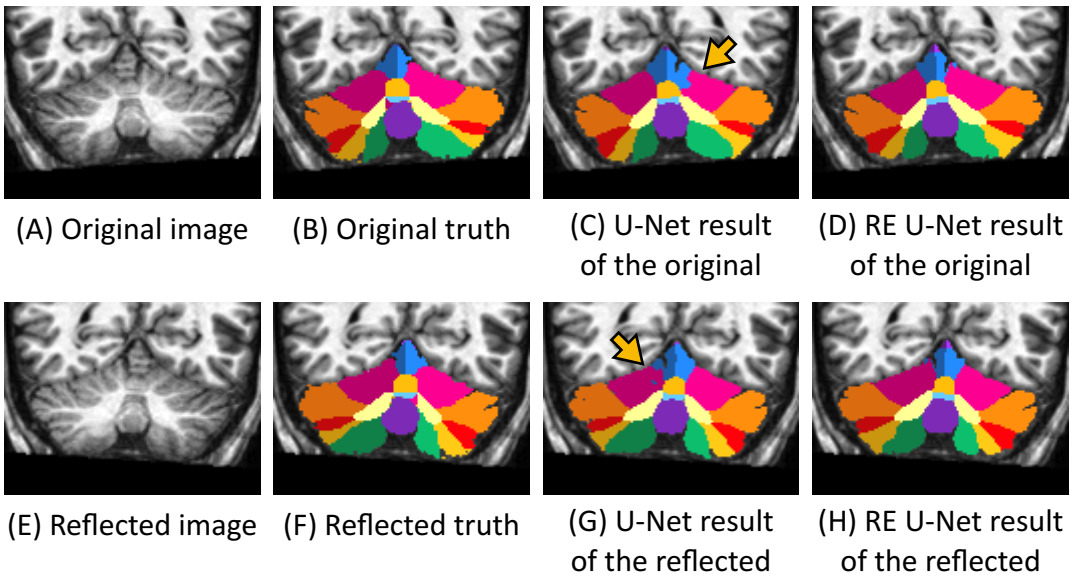


Figure 3-4. Example cerebellum parcellations. (A), (E): original and reflected testing images. (B), (F): true parcellations (manual delineations). (C), (G): results of the conventional U-Net trained with reflection augmentation. (D), (H): results of the RE U-Net. Yellow arrows point to some inconsistency of the results of the conventional U-Net.

3.1.3.4 Cerebellum Parcellation

Fifteen images from the T dataset from Carass *et al.* [19] (the testing dataset of Section 1.5 was not available for this work, so we split the training dataset of Section 1.5 into training and testing data) were used in this experiment. This dataset contains delineations of both paired regions (such as the left and right hemispheric lobules X) as well as labels without pairs (such as the corpus medullare). The images were inhomogeneity-corrected, but were not brought into MNI space (these images have already had roughly the same orientations). All images were cropped into 128 $\times$ 96 $\times$ 96 voxels around the cerebellum. We selected five images including three SCA6 subjects and two healthy controls as the testing images and the ten other images as the training images.

The RE and the conventional U-Nets both have 5 pooling layers. The RE U-Net

has 24 pairs of channels before the first pooling layer and has 80,360,646 trainable parameters. The conventional U-Net has 35 channels before the first pooling layer and has 81,171,903 trainable parameters. The RE U-Net was trained for 2,000 epochs. The conventional U-Net with reflection augmentation was trained for 1,000 epochs since it took both the original and reflected images as training data. The batch size was 2. The average Dice coefficients of all labels (including background) against the ground truth for both networks the p-values between them are shown in Table 3-I. The RE U-Net has a better mean Dice coefficient but the p-value is not significant. Example parcellations are shown in Fig. 3-4.

3.1.4 Discussion

In this work, we extended the group convolutions with respect to left-right reflection to 3D segmentation with paired labels and compared the RE and conventional U-Nets to segment anatomical structures of the human brain in T1w MR images. For skull stripping, brain tissue segmentation, and subcortical structure segmentation, the RE U-Nets are statistically significantly better than the conventional U-Nets trained with reflection data augmentation. For cerebellum parcellation, although the RE U-Net performs better in terms of the mean Dice coefficient, more testing images might be needed in order to show statistical significance. Note that these experiments were designed so that the number of parameters of the RE U-Net is less than that of the conventional U-Net in each of the four experiments. These results suggest that the RE U-Net might have more efficient parameter allocation when considering the symmetry information in the brain. For some of the experiments, the RE U-Net shows different Dice coefficients between the original and reflected images, although in theory it should be reflection equivariant. After some testings, we speculate that this is due to the floating-point errors

that accumulate throughout all network layers. One of the drawbacks of the RE U-Net is that it consumes more memory, approximately 1.25 times that of the conventional U-Net, and needs more computation. This is because the RE U-Net also performs the reflected convolution of a given kernel as shown in Eqs. (3.1) and (3.2). In addition to addressing the memory problem, future work might include fine-tuning the network parameters and training schemes and combining with other techniques to further improve brain segmentation.

3.2 Incorporating Region Hierarchy into the Network Architecture

3.2.1 Introduction

As introduced in Section 1.2, the cerebellar sub-regions are hierarchically defined. However, although this hierarchical organization of the cerebellum has been used in manual delineation protocols [104], none of previous automatic methods have explicitly utilized this knowledge. Recently, Liang *et al.* [105] incorporated semantic hierarchy concepts to construct a tree-structured CNN to achieve improved performance for segmentation. Based on their work, we explicitly built a cerebellar hierarchical organization into our 3D CNN. Our 3D network is comprised of a feature extractor and a predictor. At each voxel, our network generates features that are used to perform the corresponding hierarchical classification. The predictor is implemented using a tree structure. Each node of the tree detects a cerebellar region with child nodes subdividing it into finer sub-regions. For example, the first node in the hierarchy tree differentiates the cerebellum from the background; the cerebellum is then broken down into the corpus medullare and the

GM. See Fig. 1-10 for the complete hierarchy. We note that the two datasets, i.e., the T and M datasets (see Section 1.5), label the cerebellum using different hierarchies. As in Liang *et al.* [105], these different datasets can be used simultaneously to train the network by selecting different subsets of the tree nodes. Despite the differences in the hierarchies of these two kinds of training data, they can contribute to the training of the nodes that both hierarchies have in common. The performance of the proposed network was compared to a network that is modified from ACAPULCO, and it shows promising results.

3.2.2 Methods

3.2.2.1 Network Architectures

Our 3D network is comprised of a feature extractor and a predictor. The feature extractor is modified from the parcellating network of ACAPULCO (see Section 2.2.5) to generate 32 features for each voxel, and its output layers are replaced by our predictor network. As in Liang *et al.* [105], the predictor is constructed using a tree structure corresponding to the cerebellar hierarchical organization shown in Fig. 1-10. Each tree node uses a projection convolution to convert the input features into a single-channel image for binary classification of the corresponding region to distinguish it from the remaining labels. Note that the classification in each node is performed separately and does not compete with its sibling nodes. The whole prediction is then done recursively. Two variations of the predictor are reported for comparison purposes. The first one, the *identity predictor*, simply takes the same 32 features from the feature extractor to perform classification at all nodes in the tree. The second one, the *dense predictor*, has a similar architecture to Liang *et al.* [105]. Each node has an additional encoding block to generate two feature maps from its input, and only the two feature maps are used by the

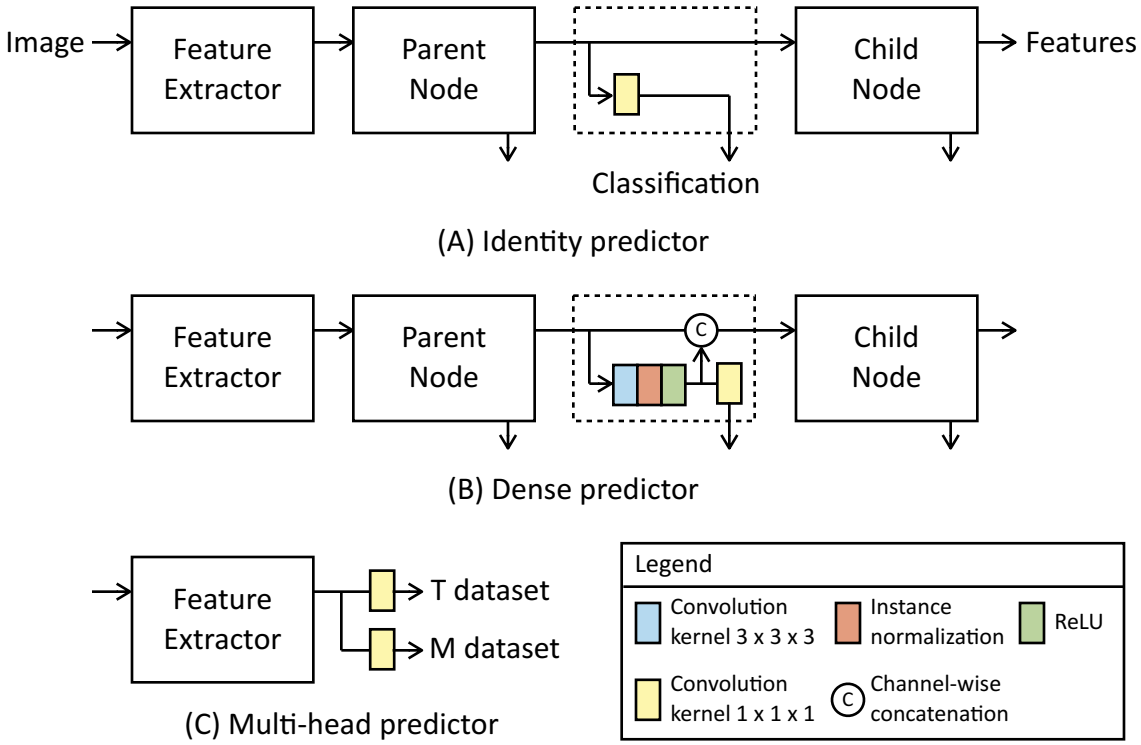


Figure 3-5. Predictor architectures: (A) the identity predictor, (B) the dense predictor, and (C) the multi-head predictor. The tree structures of (A) and (B) are only partially shown.

following projection convolution for the binary classification. Its child nodes then take the concatenation of all ancestors' newly generated two-features with the features from the feature extractor as the input. Their architectures are illustrated in Fig. 3-5(A) and (B). A more detailed illustration of the dense predictor is shown in Fig. 3-6; the tree nodes of the identity predictor are organized in the same way. To compare their performance with the parcellating network of ACAPULCO, a third predictor, the *multi-head predictor*, is used. For use with multiple datasets, this network simply uses different projection convolutions for each dataset to directly classify all the presented labels (see Fig. 3-5(C)).

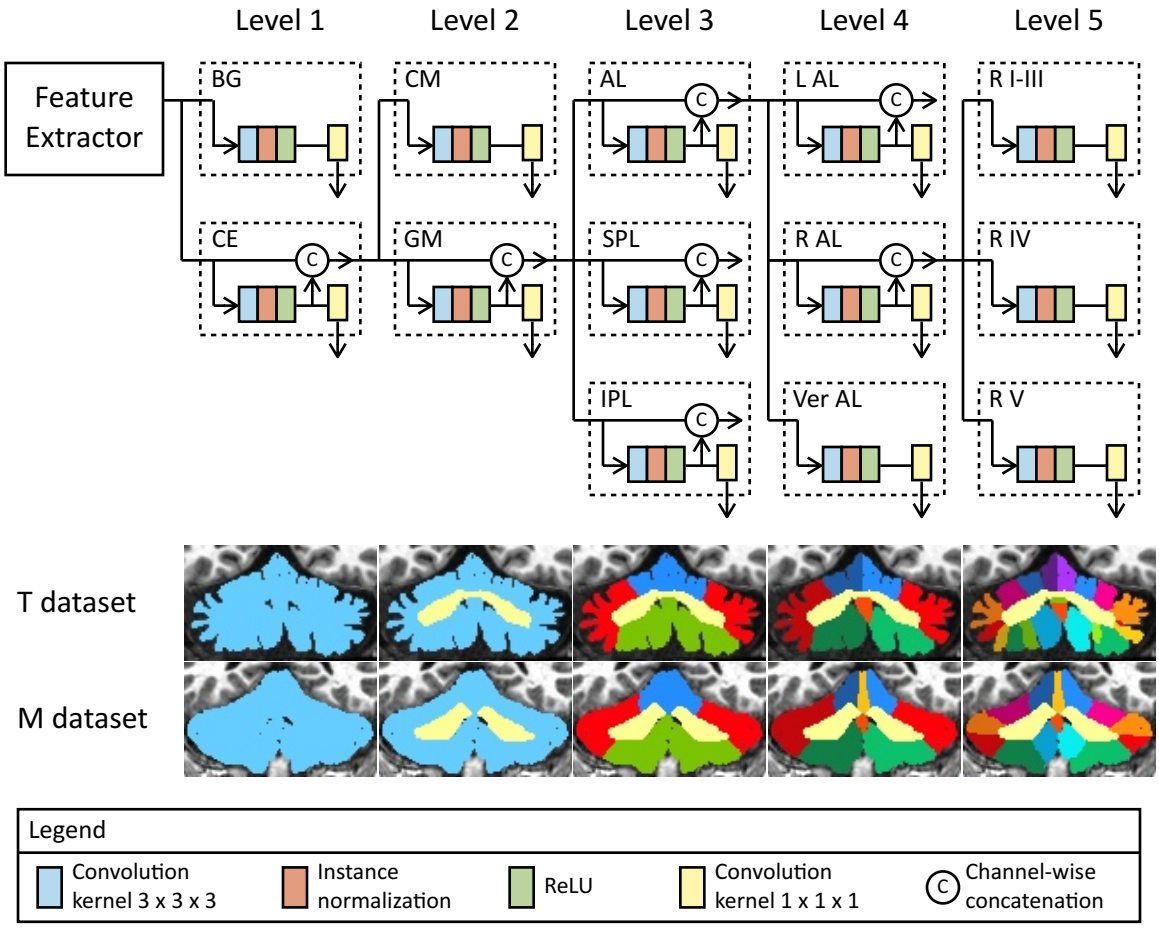


Figure 3-6. A more detailed illustration of the architecture of the dense predictor and corresponding example parcellations at each level. Only part of the tree is shown. The tree nodes of the identity predictor is organized in the same way. BG: background. CE: cerebellum. CM: corpus medullare. GM: gray matter. AL: anterior lobe. SPL: superior posterior lobe. IPL: inferior posterior lobe. L: left hemispheric lobule. R: right hemispheric lobule. Ver: vermal lobule.

3.2.2.2 Dynamic Selection of Predictor Nodes

The predictor tree is the union of all the possible manual delineation hierarchies used by the two datasets. Therefore, only a subset of the nodes in the predictor are selected for a particular sample during training. In other words, the training loss is only calculated over the available hierarchy concepts of this sample, and only the parameters of the corresponding nodes are updated in back-propagation. For example, if the dataset only

labels the cerebellar lobes rather than the lobules, although the predictor has classifiers for detecting the lobules, their parameters would not be updated. Similarly, during inference, subsets of the predictor nodes can be chosen to produce parcellations of different hierarchy concepts.

3.2.2.3 Training and Inference

To generate the true binary image for each node of the predictor during training, the foreground voxels of a node is recursively unioned from the foreground voxels of its child nodes. Since different datasets define different hierarchies, different predictor nodes must be selected. As a result, back-propagation of the training would be inefficient if images from different datasets are combined into a single mini-batch. Therefore, for each iteration, we select—at random—our training images from the same dataset. Although these datasets have different hierarchies, they always share the shallower portions of the hierarchy and differ only in the deeper levels. Consequently, the parameters of the deeper nodes of the predictor tree are updated less frequently compared to the shallower nodes. Therefore, we use different learning rates for different nodes according to the occurrences of the corresponding regions in the training data. For example, suppose the learning rate of the feature extractor is 0.002; then for a node that only presents in half of the training images, its learning rate is set to 0.004. We trained our network (the feature extractor and the predictor) from scratch using the following loss function which is based on Dice coefficients (see Eq. (2.8) for the definition of a Dice coefficient),

$$L = 1 - \frac{1}{N}\sum_{i}^{N} \frac{\epsilon + 2\sum_{j}^{M} \mathrm{sigmoid}(x_{ij})y_{ij}}{\epsilon + \sum_{j}^{M} \mathrm{sigmoid}(x_{ij}) + \sum_{j}^{M} y_{ij}}\,, \tag{3.5}$$

where N is the number of selected nodes from the predictor tree, M is the number of voxels, x_{ij} is the network output for voxel j at node i, y_{ij} is the truth for voxel j at node

i, and $\epsilon = 1 \times 10^{-8}$ prevents division by zero. For inference, only the outputs of the leaf nodes are used. These outputs are concatenated channel-wise, and a softmax is applied to convert them into a label probability map. The label of the channel with the largest probability is assigned to the voxel in the final parcellation.

3.2.3 Experiments and Results

3.2.3.1 Data

MPRAGE images from the T and M datasets were used to train and test the proposed method. N4 [27] was applied to correct the inhomogeneity, and the M dataset was rigidly registered to the 1 mm isotropic ICBM 2009c template [31] in MNI space. The images were then zero-padded to 192 $\times$ 256 $\times$ 192. Three SCA6 subjects and two healthy controls from the T dataset and ten random subjects from the M dataset were selected as the testing data. The remaining twenty images (ten from each dataset) were used as the training data (the testing datasets of Section 1.5 were not available for this work, so we split the training datasets of Section 1.5 into training and testing data).

3.2.3.2 Training and Testing

To train our networks, the training images were cropped to a size of 128 $\times$ 96 $\times$ 96 around the manual delineation of the cerebellum. The whole cropped-out region was used as the input to the proposed networks. For the testing images, we used a locating network to detect the positions of the cerebella. This network takes the whole 3D image as input and outputs a binary prediction of the cerebellum. The testing images were then cropped to 128 $\times$ 96 $\times$ 96 around the largest connected component of this network output. The Adam optimizer was used with the learning rate of the feature extractor

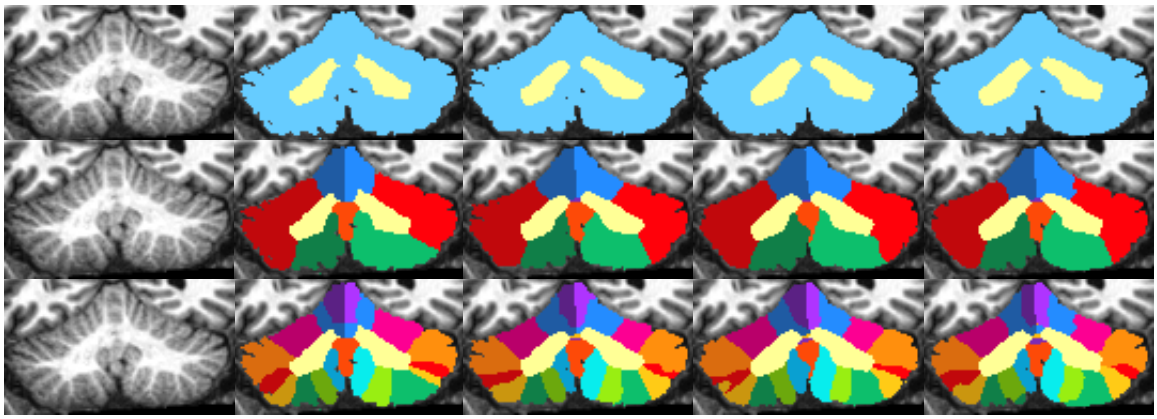
Table 3-II. Dice coefficients of the single-dataset and double-dataset training. The Dice coefficients are averaged across all labels and all subjects. The largest Dice coefficients among the three methods are highlighted in blue. NA: not applicable.

			Level 1	Level 2	Level 3	Level 4	Level 5
Single	**T dataset**	**Multi-head**	0.8374	0.8435	0.7952	NA	NA
		Identity	0.8613	0.8587	0.8138	NA	NA
		Dense	0.8585	0.8550	0.8024	NA	NA
	M dataset	**Multi-head**	0.9617	0.9314	0.9033	0.8669	0.8465
		Identity	0.9649	0.9229	0.9013	0.8668	0.8471
		Dense	0.9632	0.9246	0.8996	0.8653	0.8454
Double	**T dataset**	**Multi-head**	0.9617	0.9246	0.8980	0.8826	0.7951
		Identity	0.9540	0.9231	0.8897	0.8679	0.7790
		Dense	0.9603	0.9266	0.8988	0.8802	0.7847
	M dataset	**Multi-head**	0.9621	0.9217	0.8984	0.8648	0.8456
		Identity	0.9637	0.9265	0.9013	0.8667	0.8491
		Dense	0.9631	0.9246	0.8988	0.8651	0.8470

equal to 0.002, and other parameters were defaults in PyTorch. The batch size was 2.

We first trained the networks only on the M dataset but tested on both datasets (*single-dataset* training). All three networks were trained for 600 epochs. The Dice coefficients between the network outputs and the manual delineations were evaluated (see Table 3-II). For the M dataset, all five hierarchy levels were evaluated. For the T dataset, since its delineation protocol is different from the M dataset's, and thus there is no truth available for the last two levels, only the first three levels were evaluated.

We then trained the networks on both datasets (*double-dataset* training). The multi-head predictor and the identity predictor were trained for 300 epochs. The dense predictor was trained for 500 epochs. The Dice coefficients between the network outputs and the manual delineations are shown in Table 3-II. Two-sided paired Wilcoxon tests were performed between the multi-head predictor and the dense predictor for each label and for each dataset. The T dataset did not show any statistical differences. For the



(A) Image (B) Truth (C) Multi-head (D) Identity (E) Dense

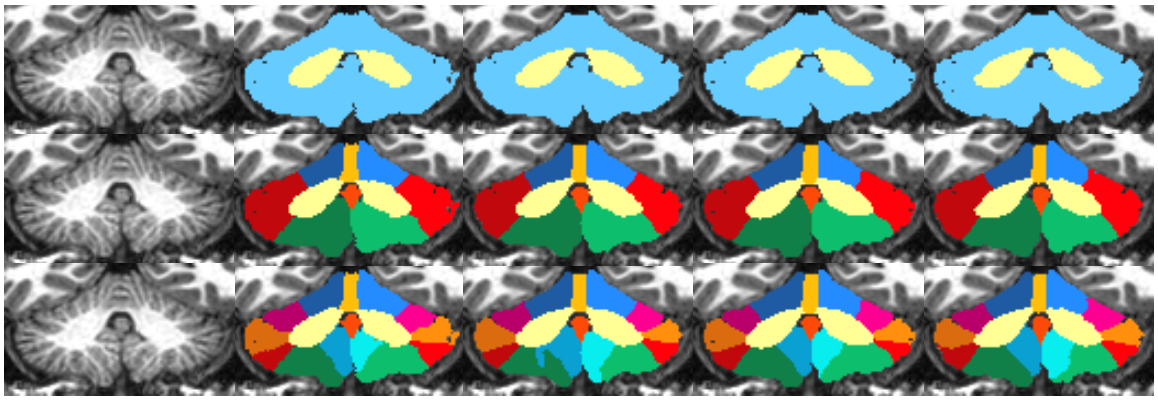
Figure 3-7. Example parcellations of an image from the T dataset in double-dataset training. From top to bottom: Level 1, Level 4, and Level 5 hierarchies. (A) the image, (B) the true parcellations, (C) the results of the multi-head predictor, (D) the results of the identity predictor, and (E) the results of the dense predictor.

M dataset, two regions, the corpus medullare and the right lobule Crus II & VIIB, were statistically better ($p < 0.05$) for the dense predictor, and one region, vermis inferior posterior, was statistically better ($p < 0.05$) for the multi-head predictor for the M dataset. Other regions were comparable. A visual comparison between these three predictors is shown in Figs. 3-7 and 3-8.

For the M dataset, the Dice coefficients of the double-dataset training are not always better than those of the single-dataset training, despite the fact that it had more training data; for the T dataset, the Dice coefficients of the double-dataset training are better than those of the single-dataset training in each available level.

3.2.4 Discussion

In this work, a tree-structured network was used to explicitly incorporate the cerebellar hierarchical organization into parcellation and also take different datasets simultaneously as training data. To improve the performance, an additional loss function could be used



(A) Image (B) Truth (C) Multi-head (D) Identity (E) Dense

Figure 3-8. Example parcellations of an image from the M dataset in double-dataset training. From top to bottom: Level 1, Level 4, and Level 5 hierarchies. (A) the image, (B) the true parcellations, (C) the results of the multi-head predictor, (D) the results of the identity predictor and (E) the results of the dense predictor.

during the training to further encourage the agreement between the output of a node and the outputs of its child nodes. Instead of using binary classification separately, the sibling nodes could be trained to learn to compete with each other via multi-label classification. During inference, the prediction of a node could be more explicitly involved in the prediction of its child nodes, for example, with conditional probability. Although the proposed method was only comparable to the baseline multi-head predictor, it shows promising results as the first method to explicitly take anatomical hierarchy into the design of a cerebellum parcellation algorithm.

3.3 Summary

In this chapter, we explored incorporating anatomical knowledge into the design of network architectures. Although they did not improve on ACAPULCO with statistical significance, our explorations show promising results and can be further investigated in the future.

Chapter 4

Conduct Longitudinal Analysis of Cerebellar Sub-Regional Volumes

4.1 Introduction

Previous studies have shown that spatial locations within the cerebellum relate to specific motor and cognitive functions [4, 6–8]. For example, in the functional MRI (fMRI) image analysis by Guell *et al.* [8], activation during performance of motor tasks was found in lobules IV, V, VI, and VIII while activation during performance of cognition tasks were found in lobules VI, crus I, crus II, VIIB, IX, and X. Since different functions exhibit different trajectories of change during aging [106, 107] and between men and women [108], it is of interest to characterize regional changes of the cerebellum in cognitively normal individuals during aging.

By parcellating the cerebellum into its sub-regions using structural MRI images, previous studies have shown age and sex differences in cerebellar sub-regional volumes. Luft *et al.* [68] used a semi-automated method to parcellate the cerebellum into 11

regions for 50 subjects. Age effects on volumes were found in the vermis, and vermal lobules VI–VII and the medial superior posterior lobe were found to be larger for women than men when adjusted for the intracranial volume (ICV). Bernard & Seidler [69] used an automatic algorithm, SUIT [35], to parcellate the cerebellum into 27 regions, and group differences were analyzed for two populations with distinct age distributions. Eleven regions were found significantly different for those two groups when adjusted for ICV. In another study [12], the authors analyzed a set of 123 subjects from 12 to 65 years old. Parcellation with SUIT was performed, but statistical results were provided only for 7 combined regions. They showed that the volumes of different regions were best fitted with respect to age using different functions such as logarithmic, linear, and quadratic functions, but the effects of sex were not studied quantitatively. Koppelmans *et al.* [13] also used SUIT to parcellate the cerebellum and then combined the sub-regions into 11 regions for 213 subjects. Age effects were found for 8 regions when adjusted for sex and ICV. In summary, Luft *et al.* [68] and Koppelmans *et al.* [13] both found a reduced volume in the vermis with older age. Bernard & Seidler [69] and Koppelmans *et al.* [13] found reduced volumes of bilateral crus I with older age. Results for other sub-regions vary across studies and are hard to compare due to use of different regional definitions. In addition to cross-sectional studies, Raz *et al.* [109–112] conducted longitudinal analyses on the cerebellum and found shrinkage over time, but they focused on the cerebellar hemispheres instead of lobules. In the following, we use “vermis” to indicate “vermal lobule” and use “lobule” to indicate “hemispheric lobule” for simplicity.

It is evident that previous studies on cerebellar sub-regional volumes are limited in either the sample size of cognitively normal older subjects or the number of parcellated regions. Furthermore, since the majority of studies are cross-sectional, the analyses describe between-subject variation, or age differences, rather than intra-individual changes from longitudinal analysis where the same subject underwent multiple visits. In

this chapter, we describe our longitudinal analyses of cerebellar sub-regional volumes during normal aging for 822 non-demented participants with 2,023 MPRAGE images from the Baltimore Longitudinal Study of Aging (BLSA) [71][1]. ACAPULCO trained with the T dataset was used to parcellate the cerebellum into 28 regions, which were further grouped into three additional levels according to the anatomical hierarchies of the cerebellum [5, 19]. The results then underwent visual inspection before inclusion in the current study. We use linear mixed-effect models [73, 113] to analyze the relationship between cerebellar sub-regional volumes and age and sex. Our analyses answer the following questions of cerebellar sub-regional volumes in older adults during normal aging: What are the cross-sectional and longitudinal effects of age and sex? How do age and sex modify the longitudinal changes?

4.2 Linear Mixed-Effects Model

Compared with linear regression models, linear mixed-effects models [113] account for the correlations of multiple measurements from each individual by incorporating individual-specific random effects. These models contain both *fixed* effects—which are the population average regression coefficients—and *random* effects—which are each individual's deviations from the fixed effects and are assumed to be drawn from a Gaussian distribution. For a subject i, suppose the dependent variable $\boldsymbol{y}_i$ is a n_i-by-1 vector where n_i is the number of observations for this subject. A linear mixed-effects model can be written as

$$\boldsymbol{y}_i = X_i\boldsymbol{\beta} + Z_i\boldsymbol{b}_i + \boldsymbol{\epsilon}_i, \tag{4.1}$$

[1]This study was supported in part by the Intramural Research Program, National Institute on Aging, NIH.

where X_i is the n_i-by-p fixed-effects design matrix (which contains covariates such as, for example, the age of this subject at different visits and their biological sex), $\boldsymbol{\beta}$ is the p-by-1 vector of the fixed-effect coefficients, Z_i is the n_i-by-q random-effects design matrix (which can contain different covariates compared to X_i), $\boldsymbol{b}_i$ is the q-by-1 random-effects coefficients, and $\boldsymbol{\epsilon}_i$ is the n_i-by-1 residual errors. $\boldsymbol{b}_i$ is modeled as random variables drawn from a Gaussian distribution $\mathcal{N}(0, D)$ where D is the covariance matrix. Each element of $\boldsymbol{\epsilon}_i$ is usually assumed to be independent and identically distributed; i.e., $\boldsymbol{\epsilon}_i \in \mathcal{N}(0, \sigma^2 I)$ where σ^2 is its variance, and I is an identity matrix. We are generally interested in making statistical inference on each element of $\boldsymbol{\beta}$—i.e., whether it is significantly associated with the dependent variable $\boldsymbol{y}_i$ on the population level.

4.3 Methods

4.3.1 Participants

The BLSA is an observational study that began in 1958 and is currently conducted by the National Institute on Aging Intramural Research Program [71]. Recruitment is ongoing, and participants and visits included in these analyses represent a snapshot in time. The current visit schedule depends on age: participants younger than 60 are assessed every 4 years, participants between 60 and 80 are assessed every 2 years, and participants older than 80 are assessed every year. The varied numbers of visits per participant primarily reflect the timing of enrollment and their age. 2,381 available MPRAGE images for 1,017 participants were processed. 70 images with artifacts or low parcellation quality were excluded, as summarized in Table 4-I. We excluded 98 images for visits after the year of onset of dementia or mild cognitive impairment [114]. Due to very limited longitudinal data in younger participants in the BLSA, we focused this study

Table 4-I. Exclusion criteria of images in our analyses.

Reason	No. subjects	No. images
Failed MNI alignment	14	30
Failed inhomogeneity correction	1	1
Failed parcellation	18	27
Image artifacts	12	12

on subjects older than 50 years, resulting in 2,033 images. The ICV was calculated using a brain extraction algorithm, MASS [115], from a separate study. We used the ICV value from the earliest visit for each subject, and 10 subjects without ICV were excluded. The final dataset included 2,023 images from 822 subjects. The mean age at baseline is 70.7 years with standard deviation (SD) 10.2 years. The mean follow-up interval for subjects with multiple visits is 3.7 years with SD 1.9 years. These subjects are highly educated (17.0 years of education on average) and mostly Caucasian (67.5%). The demographic characteristics of the participants are further summarized in Table 4-II.

4.3.2 MRI Acquisition and Image Analysis

The images were acquired on 3.0 T MRI scanners (Achieva, Phillips Medical Systems, Netherlands). Image matrix = 256 $\times$ 240, number of sagittal slices = 170, pixel size = 1 mm $\times$ 1 mm, slice thickness = 1.2 mm, flip angle = 8°, TE = 3.1 ms, 47 images were acquired with TR = 6.8 ms, and 1,976 images were acquired with TR = 6.6 ms. ACAPULCO was applied to all images to parcellate the cerebellum. Since some previous methods report statistical analysis of coarser levels of cerebellar divisions [13, 68, 112], we also provide results of anatomically meaningful grouped regions in addition to sub-components, so as to facilitate more direct comparisons among the literature. Following Schmahmann *et al.* [5], these regions were grouped into the bilateral anterior,

Table 4-II. Sample characteristics in our analyses. Follow-up intervals are calculated for subjects with two or more visits.

	Overall	Female	Male
Number of subjects	822	454	368
Number of visits	2,023	1,136	887
Age (years)			
Mean (SD)	70.7 (10.2)	69.8 (10.0)	72.0 (10.3)
Range	50.1–95.1	50.1–95.1	50.8–94.7
Follow-up (years)			
Mean (SD)	3.7 (1.9)	3.8 (1.9)	3.5 (1.9)
Range	0.8–9.3	0.8–9.3	0.8–9.0
ICV (cm^3)			
Mean (SD)	1,388.12 (140.20)	1,309.58 (105.96)	1,485.02 (114.32)
Range	999.75–1,886.29	999.75–1,672.97	1,216.47–1,886.29
Number of subjects by the total number of visits			
1	285	159	126
2	183	92	91
3	167	93	74
4	110	65	45
5	53	30	23
6	10	7	3
7	8	4	4
8	4	2	2
9	2	2	0

posterior, and flocculonodular lobes, vermes VI–VII, VIII–IX, and X, and further into the bilateral hemispheres, the whole vermis, and the whole cerebellum. Finally, the volumes of each region at each grouping level were calculated.

We calculated intraclass correlation coefficients (ICCs) for each cerebellar sub-region to further validate ACAPULCO. Here we used linear mixed-effects models to calculate ICCs. The fixed effects included the intercept and age, the random effect included only

Table 4-III. ICCs of cerebellar sub-regions.

Region	ICC	Region	ICC
Corpus Medullare	0.96	**Vermis VI**	0.96
Vermis VII	0.96	**Vermis VIII**	0.98
Vermis IX	0.95	**Vermis X**	0.94
Left Lobules I–III	0.94	**Right Lobules I–III**	0.95
Left Lobule IV	0.90	**Right Lobule IV**	0.87
Left Lobule V	0.89	**Right Lobule V**	0.89
Left Lobule VI	0.97	**Right Lobule VI**	0.97
Left Crus I	0.96	**Right Crus I**	0.97
Left Crus II	0.92	**Right Crus II**	0.93
Left Lobule VIIB	0.93	**Right Lobule VIIB**	0.91
Left Lobule VIIIA	0.94	**Right Lobule VIIIA**	0.94
Left Lobule VIIIB	0.93	**Right Lobule VIIIB**	0.95
Left Lobule IX	0.99	**Right Lobule IX**	0.99
Left Lobule X	0.95	**Right Lobule X**	0.94

the intercept, and the data were grouped by subjects. The ICC is defined as

$$\text{ICC} = \frac{\sigma_1^2}{\sigma_1^2 + \sigma_0^2}, \tag{4.2}$$

where σ_1^2 is the variance of intercept in random effects, and σ_0^2 is the variance of residuals. ICCs of the 28 cerebellar sub-regions ranging from 0.87 to 0.99 are shown in Table 4-III. ICCs in the range [0.75–0.90) are considered good and values [0.90–1.00] are considered excellent [95]. In summary, this equates to 25 of our cerebellum regions as excellent and the remaining three as good.

4.3.3 Statistical Analysis

Linear mixed-effects models were used to study the relationship of age and sex with baseline and the longitudinal change in each cerebellar volume individually. There are

28 lobular regions from ACAPULCO plus 9 grouped regions, resulting in 37 regions in total (see Tables 4-IV and 4-V for results). The volume in mm^3 of each of these regions was used as a separate outcome in each of the 37 regressions. The fixed effects are the intercept, ICV in cm^3 centered around 1,400, baseline age in years centered around 70, sex with 0.5 indicating male and -0.5 indicating female, follow-up interval in years, the interaction between baseline age and follow-up interval (baseline age $\times$ follow-up interval), and the interaction between sex and follow-up interval (sex $\times$ follow-up interval). The random effects are the intercept and follow-up interval. To account for multiple comparisons, p-values of each fixed effect were adjusted across the 37 regions using Bonferroni correction. Type I error level $p \leq 0.05$ was applied to the adjusted p-values to test whether the fixed effects are significantly different from 0. All linear mixed-effects models were fit in R version 3.5.1 using the `lme` function from the `nlme` library version 3.1.137 [73].

4.4 Results

The estimated coefficients, standard errors, and *raw* p-values of baseline age, sex, follow-up interval, and the interactions for each cerebellar region are shown in Tables 4-IV and 4-V. Significant effects for *Bonferroni adjusted* $p \leq 0.05$ of each region are highlighted in blue.

4.4.1 Total Cerebellum, Corpus Medullare, and Hemispheres

Both baseline age and follow-up interval are significant for the total cerebellum, corpus medullare, and bilateral hemispheres, indicating that these volumes are smaller with higher age and decline longitudinally. The interaction between baseline age and follow-

Table 4-IV. Fixed effect coefficients (β), standard errors (SE), and *raw* p-values (p) for baseline age (Age), sex, follow-up interval (Time). Significant (*Bonferroni adjusted* $p \leq 0.05$) effects are highlighted in blue. AL: anterior lobe. CM: corpus medullare. H: hemisphere. PL: posterior lobe. L: left. R: right. Ver: vermis.

	Age			Sex			Time		
	β	SE	p	β	SE	p	β	SE	p
Total	−414.30	31.60	9×10^{-36}	464.25	824.05	6×10^{-1}	−419.73	20.78	3×10^{-78}
CM	−61.54	4.82	3×10^{-34}	287.43	125.88	2×10^{-2}	−95.71	4.93	4×10^{-73}
L H	−178.93	13.95	2×10^{-34}	15.13	363.53	1×10^{0}	−188.24	10.49	6×10^{-64}
R H	−164.86	13.98	1×10^{-29}	204.96	364.43	6×10^{-1}	−129.00	10.55	2×10^{-32}
Ver	−8.49	1.79	2×10^{-6}	−42.61	46.74	4×10^{-1}	−7.67	1.14	2×10^{-11}
Ver VI–IX	−8.02	1.73	4×10^{-6}	−23.95	45.27	6×10^{-1}	−7.20	1.11	1×10^{-10}
Ver VI	−4.85	0.69	5×10^{-12}	−19.28	18.03	3×10^{-1}	−1.72	0.46	2×10^{-4}
Ver VII	−0.16	0.57	8×10^{-1}	−15.41	14.94	3×10^{-1}	−1.18	0.43	6×10^{-3}
Ver VIII	−0.22	0.98	8×10^{-1}	61.19	25.54	2×10^{-2}	−1.28	0.58	3×10^{-2}
Ver IX	−2.79	0.47	4×10^{-9}	−49.43	12.21	6×10^{-5}	−2.94	0.43	1×10^{-11}
Ver X	−0.46	0.17	7×10^{-3}	−18.53	4.45	3×10^{-5}	−0.48	0.16	2×10^{-3}
L AL	−4.68	2.84	1×10^{-1}	−94.32	73.79	2×10^{-1}	2.35	2.82	4×10^{-1}
R AL	−10.72	2.78	1×10^{-4}	75.46	72.20	3×10^{-1}	−9.58	2.74	5×10^{-4}
L I–III	−1.45	0.63	2×10^{-2}	−37.04	16.37	2×10^{-2}	−3.09	0.61	6×10^{-7}
R I–III	−2.10	0.64	1×10^{-3}	−56.46	16.72	8×10^{-4}	−3.55	0.55	2×10^{-10}
L IV	2.87	1.81	1×10^{-1}	15.85	46.78	7×10^{-1}	17.66	2.23	5×10^{-15}
R IV	4.36	1.78	1×10^{-2}	82.54	46.10	7×10^{-2}	13.93	2.58	8×10^{-8}
L V	−6.10	1.41	2×10^{-5}	−71.96	36.62	5×10^{-2}	−12.38	2.02	1×10^{-9}
R V	−13.01	1.47	5×10^{-18}	48.12	37.56	2×10^{-1}	−19.44	1.88	4×10^{-24}
L PL	−173.36	12.69	2×10^{-38}	104.39	330.19	8×10^{-1}	−188.69	10.17	1×10^{-67}
R PL	−153.41	12.62	2×10^{-31}	128.18	328.77	7×10^{-1}	−118.54	10.13	5×10^{-30}
L VI	−40.33	4.03	2×10^{-22}	−125.73	104.76	2×10^{-1}	−55.64	2.78	3×10^{-77}
R VI	−30.80	3.91	1×10^{-14}	−246.55	101.80	2×10^{-2}	−37.31	2.70	3×10^{-40}
L crus I	−59.69	5.44	3×10^{-26}	−189.04	141.08	2×10^{-1}	−47.36	4.33	1×10^{-26}
R crus I	−57.88	5.51	3×10^{-24}	−10.81	143.77	9×10^{-1}	−22.14	4.53	1×10^{-6}
L crus II	−22.04	3.74	6×10^{-9}	181.57	97.28	6×10^{-2}	−25.85	3.72	6×10^{-12}
R crus II	−19.16	3.99	2×10^{-6}	228.66	103.56	3×10^{-2}	−17.27	3.92	1×10^{-5}
L VIIB	−27.93	3.10	1×10^{-18}	−234.96	80.07	3×10^{-3}	−27.45	3.04	6×10^{-19}
R VIIB	−27.73	3.11	3×10^{-18}	−203.93	80.09	1×10^{-2}	−30.98	3.33	7×10^{-20}
L VIIIA	−8.93	2.88	2×10^{-3}	419.85	75.10	3×10^{-8}	−11.19	3.20	5×10^{-4}
R VIIIA	−0.22	2.45	9×10^{-1}	348.53	63.42	5×10^{-8}	3.71	2.85	2×10^{-1}
L VIIIB	−6.65	1.96	7×10^{-4}	110.22	50.48	3×10^{-2}	−8.65	1.89	5×10^{-6}
R VIIIB	−10.84	1.92	2×10^{-8}	74.30	49.95	1×10^{-1}	−7.12	1.70	3×10^{-5}
L IX	−7.89	2.06	1×10^{-4}	−42.45	53.72	4×10^{-1}	−9.69	0.83	6×10^{-30}
R IX	−6.74	2.03	9×10^{-4}	−41.60	52.82	4×10^{-1}	−8.05	0.92	5×10^{-18}
L X	−1.02	0.24	2×10^{-5}	10.22	6.13	1×10^{-1}	−0.99	0.23	2×10^{-5}
R X	−0.84	0.23	3×10^{-4}	2.70	6.04	7×10^{-1}	−0.48	0.22	3×10^{-2}

Table 4-V. Fixed effect coefficients (β), standard errors (SE), and *raw* p-values (p) for *interactions* between baseline age (Age), sex, follow-up interval (Time). Significant (*Bonferroni adjusted* $p \leq 0.05$) effects are highlighted in blue. AL: anterior lobe. CM: corpus medullare. H: hemisphere. PL: posterior lobe. L: left. R: right. Ver: vermis.

	Age × Time			**Sex × Time**		
	β	**SE**	p	β	**SE**	p
Total	−7.75	2.22	5×10^{-4}	−82.24	41.16	5×10^{-2}
CM	−0.21	0.53	7×10^{-1}	−43.38	9.80	1×10^{-5}
L H	−4.69	1.12	3×10^{-5}	−26.01	20.78	2×10^{-1}
R H	−3.01	1.13	8×10^{-3}	−18.72	20.96	4×10^{-1}
Ver	−0.42	0.12	5×10^{-4}	−2.03	2.26	4×10^{-1}
Ver VI–IX	−0.39	0.12	1×10^{-3}	−2.32	2.20	3×10^{-1}
Ver VI	−0.23	0.05	2×10^{-6}	0.34	0.91	7×10^{-1}
Ver VII	−0.09	0.05	5×10^{-2}	−1.56	0.85	7×10^{-2}
Ver VIII	−0.10	0.06	1×10^{-1}	−0.84	1.15	5×10^{-1}
Ver IX	0.03	0.05	5×10^{-1}	0.11	0.85	9×10^{-1}
Ver X	−0.03	0.02	6×10^{-2}	0.32	0.31	3×10^{-1}
L AL	−1.09	0.30	3×10^{-4}	19.47	5.61	5×10^{-4}
R AL	−1.45	0.29	9×10^{-7}	4.33	5.45	4×10^{-1}
L I–III	−0.14	0.07	3×10^{-2}	−0.03	1.22	1×10^{0}
R I–III	−0.10	0.06	1×10^{-1}	−0.75	1.10	5×10^{-1}
L IV	−0.62	0.24	1×10^{-2}	13.47	4.44	2×10^{-3}
R IV	−0.89	0.28	1×10^{-3}	11.09	5.14	3×10^{-2}
L V	−0.26	0.22	2×10^{-1}	5.44	4.01	2×10^{-1}
R V	−0.37	0.20	7×10^{-2}	−5.39	3.74	1×10^{-1}
L PL	−3.42	1.09	2×10^{-3}	−44.29	20.16	3×10^{-2}
R PL	−1.34	1.09	2×10^{-1}	−20.05	20.12	3×10^{-1}
L VI	−0.80	0.30	7×10^{-3}	−18.07	5.51	1×10^{-3}
R VI	−0.15	0.29	6×10^{-1}	−1.27	5.37	8×10^{-1}
L crus I	−1.43	0.46	2×10^{-3}	−4.74	8.59	6×10^{-1}
R crus I	−0.33	0.49	5×10^{-1}	−3.20	9.01	7×10^{-1}
L crus II	−0.43	0.40	3×10^{-1}	−20.66	7.38	5×10^{-3}
R crus II	−0.13	0.42	8×10^{-1}	−27.04	7.78	5×10^{-4}
L VIIB	0.14	0.32	7×10^{-1}	10.69	6.02	8×10^{-2}
R VIIB	−0.42	0.36	2×10^{-1}	16.76	6.62	1×10^{-2}
L VIIIA	−0.48	0.34	2×10^{-1}	−2.67	6.36	7×10^{-1}
R VIIIA	−0.05	0.31	9×10^{-1}	−11.15	5.67	5×10^{-2}
L VIIIB	−0.35	0.20	9×10^{-2}	−0.78	3.76	8×10^{-1}
R VIIIB	−0.24	0.18	2×10^{-1}	1.47	3.38	7×10^{-1}
L IX	−0.07	0.09	4×10^{-1}	−3.98	1.65	2×10^{-2}
R IX	−0.15	0.10	1×10^{-1}	−0.76	1.82	7×10^{-1}
L X	−0.07	0.02	6×10^{-3}	−0.08	0.46	9×10^{-1}
R X	−0.09	0.02	3×10^{-4}	−1.10	0.44	1×10^{-2}

up interval is significant for the total cerebellum and the left hemisphere. Negative coefficients suggest that these volumes decline faster at more advanced baseline age. Sex is not significant for any of these regions. The interaction between sex and follow-up interval is significant for the corpus medullare, suggesting that this volume declines faster for men.

4.4.2 Vermis and Vermal Lobules

Baseline age, follow-up interval, and the interaction between these two factors are significant for the whole vermis, suggesting that the vermis volume is smaller at higher baseline age and declines longitudinally over time, and this decline is faster at higher baseline age.

For the sub-regions of the vermis, baseline age and follow-up interval are significant for vermes VI–IX (the part corresponding to the posterior lobe) and its sub-regions vermes VI and IX, suggesting smaller volumes at higher age and longitudinal declines over time. The interaction between baseline age and follow-up interval is significant for vermes VI–IX and its sub-region vermis VI, suggesting faster declines at more advanced baseline age. Sex is significant for vermis IX and vermis X (the part corresponding to the flocculonodular lobe) with negative coefficients, suggesting smaller volumes for men than women.

4.4.3 Hemispheric Anterior Lobe and Lobules

Both baseline age and follow-up interval are significant for the right anterior lobe, indicating that its volume is smaller with higher age and declines longitudinally over time. The interaction between baseline age and follow-up time is significant for both sides, indicating faster declines at higher baseline age. There is no significant sex

difference, but there is a significant interaction between sex and follow-up interval for the left anterior lobe. This interaction shows less steep decline in men compared with women at advanced age.

For lobules of the anterior lobe, baseline age is significant for right lobules I–III and bilateral lobules V, indicating smaller volumes with higher age. Sex is significant for right lobules I–III, suggesting that the volume is greater for women than men at baseline. Longitudinal change is significant for all lobules of the anterior lobe. Note that the coefficient of follow-up interval for bilateral lobules IV is positive, indicating increasing volumes over time. The interaction between baseline age and follow-up interval is significant for right lobule IV, suggesting a steeper decline at more advanced baseline age.

4.4.4 Hemispheric Posterior Lobe and Lobules

Baseline age and follow-up interval are significant for the bilateral posterior lobes, suggesting that the volumes are smaller with higher baseline age and decline longitudinally. Sex is not significant for the posterior lobe.

For lobules of the posterior lobe, baseline age is significant for all lobules except for bilateral lobules VIIIA, suggesting smaller volumes with higher baseline age. Sex is significant for bilateral lobules VIIIA with positive coefficients, indicating larger volumes for men at baseline. Follow-up interval is significant except for right lobule VIIIA, suggesting longitudinal declines over time. The interaction between sex and follow-up interval is significant for right crus II with a negative coefficient, suggesting a faster decline for men than women.

4.4.5 Hemispheric Flocculonodular Lobes

The flocculonodular lobe is only composed of lobule X. Baseline age is significant for bilateral lobules X, indicating smaller volumes at higher baseline age. Follow-up interval is significant for the left, indicating a longitudinal decline over time. The interaction between baseline age and follow-up interval is significant for the right, indicating a steeper decline at higher baseline age.

4.4.6 Visualization

To visualize the results, we show the fitted population-average trajectories of all regions (see Figs. 4-1–4-23). To plot these trajectories, we used baseline age from 55 to 90 in 5-year age bands, follow-up intervals from 0 to 4 years for each baseline age, and ICV 1,400 cm^3, by sex. Note that if the coefficients of the interactions—i.e., baseline age $\times$ follow-up interval and sex $\times$ follow-up interval—are not significant, we do not incorporate them into these figures. The color-coded *raw* p-values of baseline age, sex, and follow-up interval on top of a cerebellum illustration are further shown in Fig. 4-24.

4.5 Discussion

In this work, we analyzed age differences and longitudinal changes of cerebellar subregional volumes in a large sample of non-demented individuals with baseline age 50 years and older. Our results indicate that the cerebellum volume has spatially varying trajectories with respect to baseline age and longitudinal change over time, and only a few sub-regions show sex differences after adjustment for ICV. Twenty of the 28 parcellated regions show statistically significant cross-sectional age effects. Twenty-three of the 28 regions have statistically significant longitudinal changes. For three of

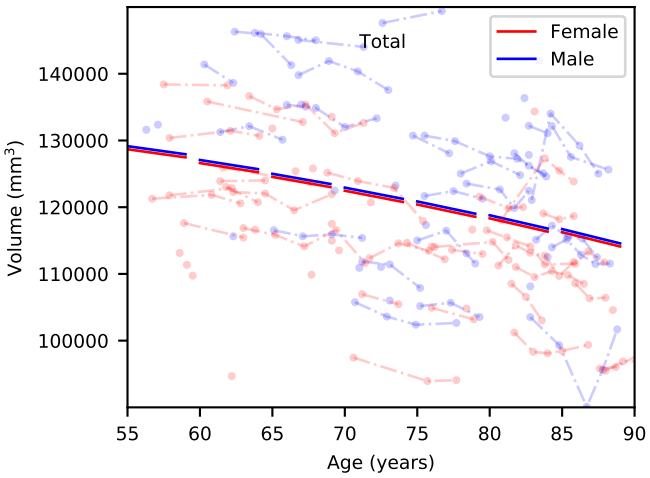


Figure 4-1. Fitted population average trajectories of the *total cerebellum*. Thick lines indicate the trajectories. Dots indicate volumes of randomly selected subjects and are connected by thin lines for the same subjects. Each thick line segment is plotted with the baseline age at its starting point, 0–4 follow-up years, and 1,400 cm^3 ICV. Female and male data are plotted in red and blue, respectively.

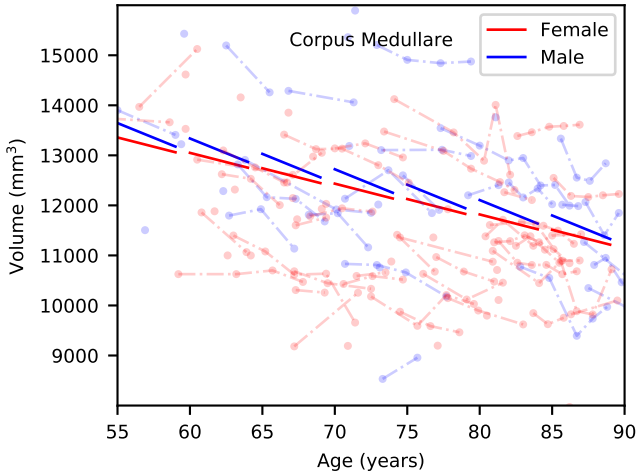


Figure 4-2. Fitted population average trajectories of the *corpus medullare*. Thick lines indicate the trajectories. Dots indicate volumes of randomly selected subjects and are connected by thin lines for the same subjects. Each thick line segment is plotted with the baseline age at its starting point, 0–4 follow-up years, and 1,400 cm^3 ICV. Female and male data are plotted in red and blue, respectively.

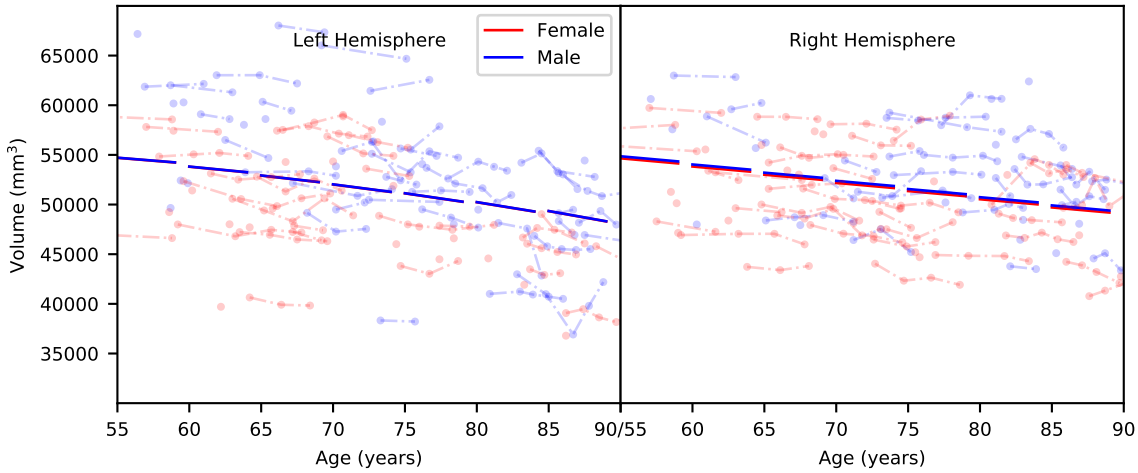


Figure 4-3. Fitted population average trajectories of the bilateral *hemispheres*. Thick lines indicate the trajectories. Dots indicate volumes of randomly selected subjects and are connected by thin lines for the same subjects. Each thick line segment is plotted with the baseline age at its starting point, 0–4 follow-up years, and 1,400 cm^3 ICV. Female and male data are plotted in red and blue, respectively.

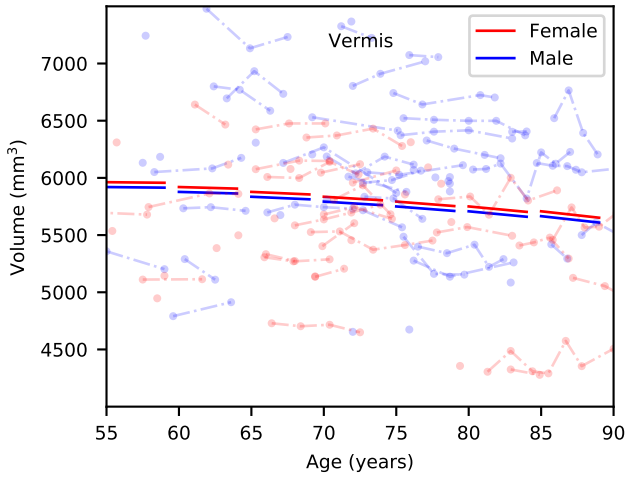


Figure 4-4. Fitted population average trajectories of the whole *vermis*. Thick lines indicate the trajectories. Dots indicate volumes of randomly selected subjects and are connected by thin lines for the same subjects. Each thick line segment is plotted with the baseline age at its starting point, 0–4 follow-up years, and 1,400 cm^3 ICV. Female and male data are plotted in red and blue, respectively.

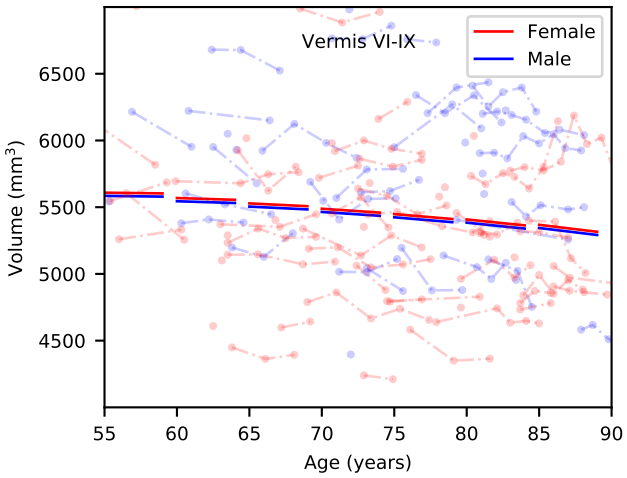


Figure 4-5. Fitted population average trajectories of *vermes VI–IX*. Thick lines indicate the trajectories. Dots indicate volumes of randomly selected subjects and are connected by thin lines for the same subjects. Each thick line segment is plotted with the baseline age at its starting point, 0–4 follow-up years, and 1,400 cm^3 ICV. Female and male data are plotted in red and blue, respectively.

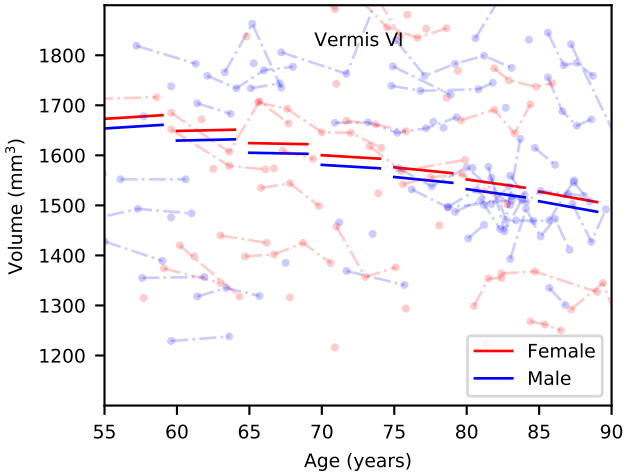


Figure 4-6. Fitted population average trajectories of *vermis VI*. Thick lines indicate the trajectories. Dots indicate volumes of randomly selected subjects and are connected by thin lines for the same subjects. Each thick line segment is plotted with the baseline age at its starting point, 0–4 follow-up years, and 1,400 cm^3 ICV. Female and male data are plotted in red and blue, respectively.

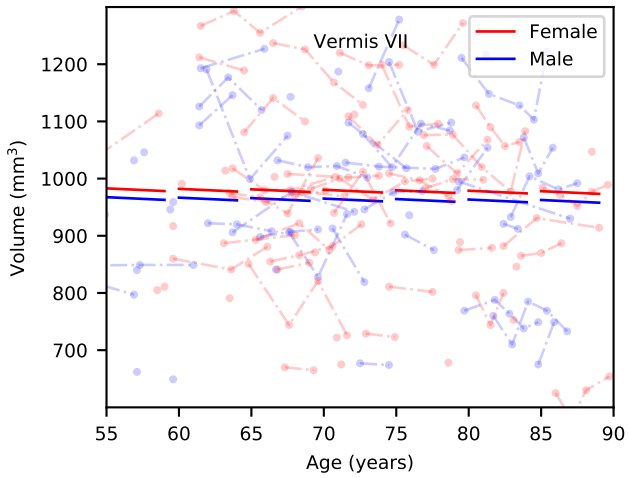


Figure 4-7. Fitted population average trajectories of *vermis VII*. Thick lines indicate the trajectories. Dots indicate volumes of randomly selected subjects and are connected by thin lines for the same subjects. Each thick line segment is plotted with the baseline age at its starting point, 0–4 follow-up years, and 1,400 cm^3 ICV. Female and male data are plotted in red and blue, respectively.

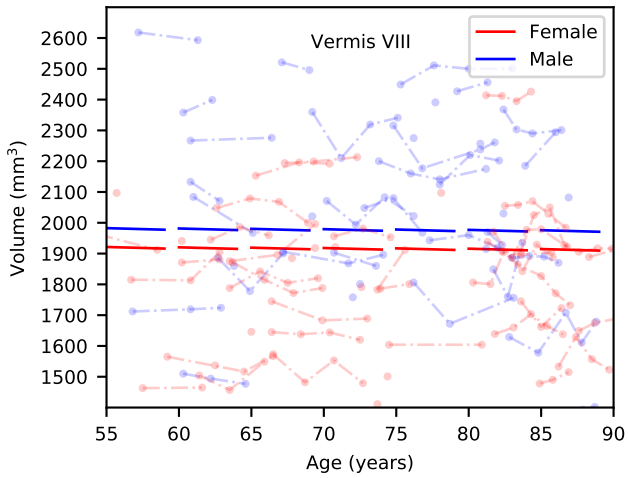


Figure 4-8. Fitted population average trajectories of *vermis VIII*. Thick lines indicate the trajectories. Dots indicate volumes of randomly selected subjects and are connected by thin lines for the same subjects. Each thick line segment is plotted with the baseline age at its starting point, 0–4 follow-up years, and 1,400 cm^3 ICV. Female and male data are plotted in red and blue, respectively.

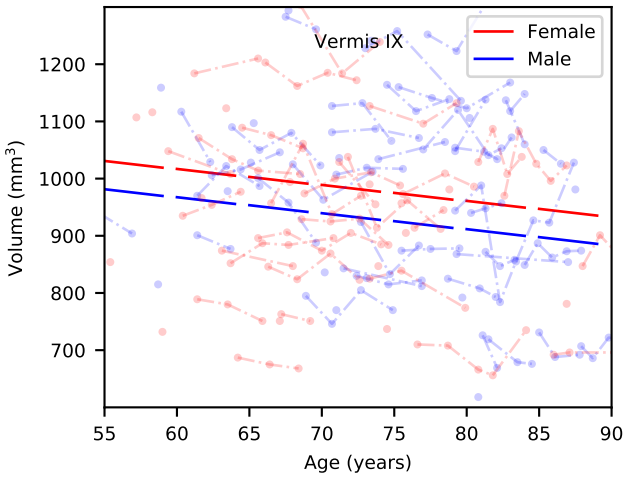


Figure 4-9. Fitted population average trajectories of *vermis IX*. Thick lines indicate the trajectories. Dots indicate volumes of randomly selected subjects and are connected by thin lines for the same subjects. Each thick line segment is plotted with the baseline age at its starting point, 0–4 follow-up years, and 1,400 cm^3 ICV. Female and male data are plotted in red and blue, respectively.

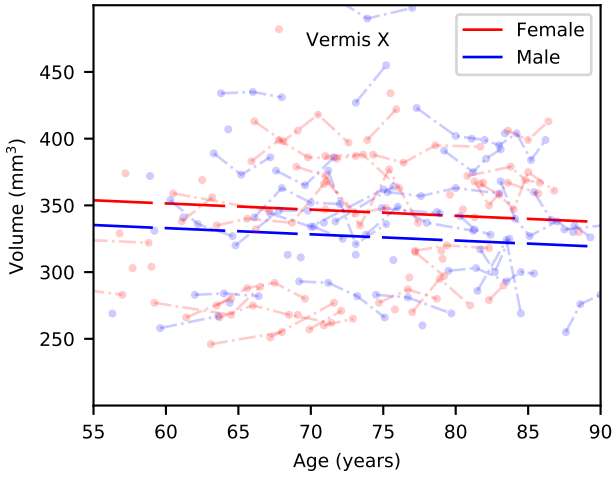


Figure 4-10. Fitted population average trajectories of *vermis X*. Thick lines indicate the trajectories. Dots indicate volumes of randomly selected subjects and are connected by thin lines for the same subjects. Each thick line segment is plotted with the baseline age at its starting point, 0–4 follow-up years, and 1,400 cm^3 ICV. Female and male data are plotted in red and blue, respectively.

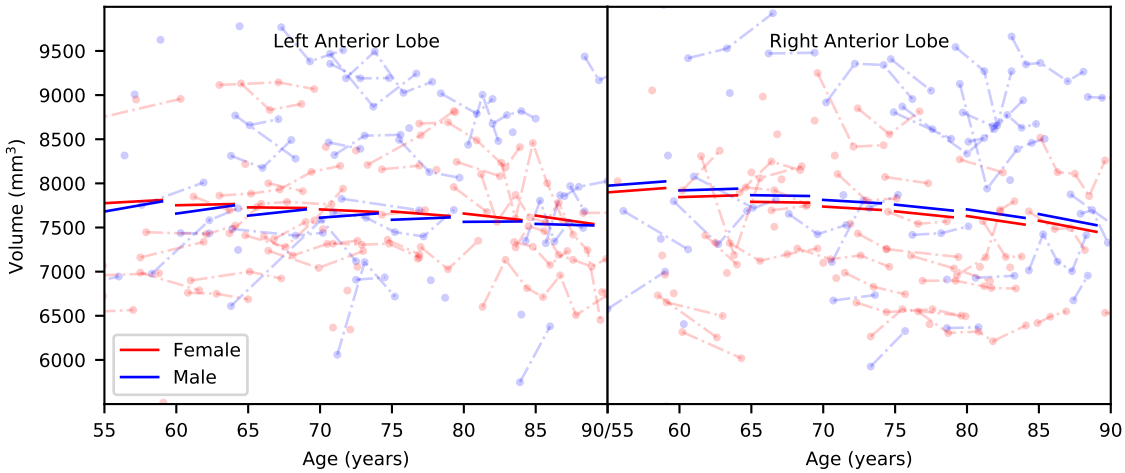


Figure 4-11. Fitted population average trajectories of bilateral *anterior lobes*. Thick lines indicate the trajectories. Dots indicate volumes of randomly selected subjects and are connected by thin lines for the same subjects. Each thick line segment is plotted with the baseline age at its starting point, 0–4 follow-up years, and 1,400 cm^3 ICV. Female and male data are plotted in red and blue, respectively.

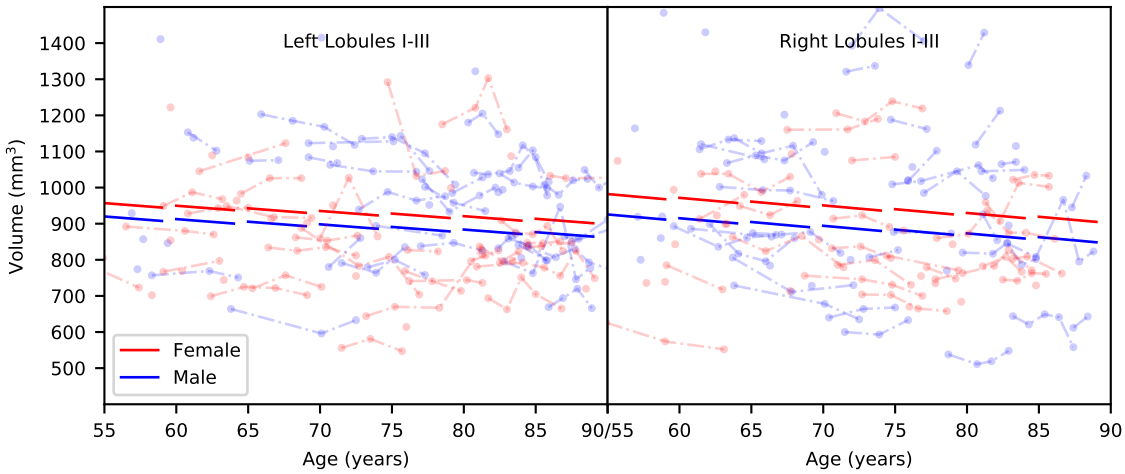


Figure 4-12. Fitted population average trajectories of bilateral *lobules I–III*. Thick lines indicate the trajectories. Dots indicate volumes of randomly selected subjects and are connected by thin lines for the same subjects. Each thick line segment is plotted with the baseline age at its starting point, 0–4 follow-up years, and 1,400 cm^3 ICV. Female and male data are plotted in red and blue, respectively.

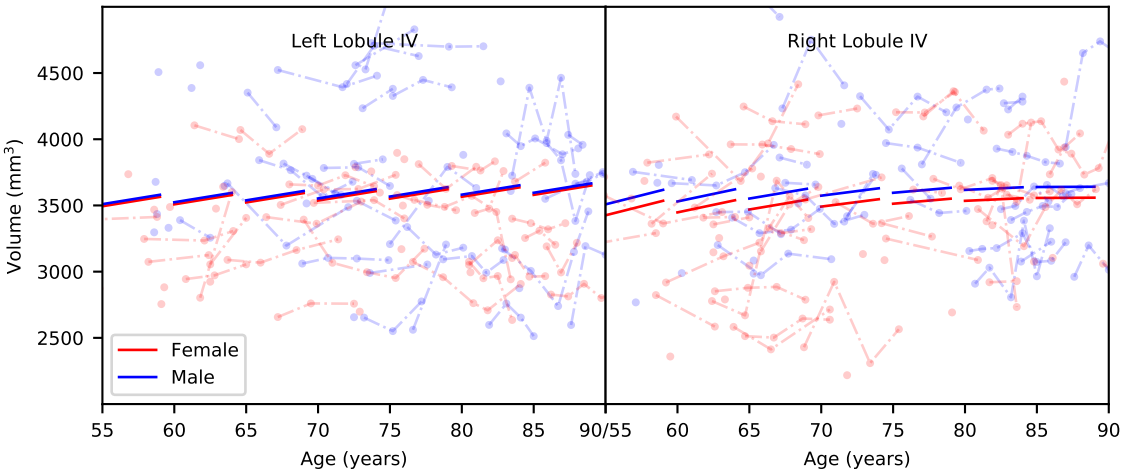


Figure 4-13. Fitted population average trajectories of bilateral *lobules IV*. Thick lines indicate the trajectories. Dots indicate volumes of randomly selected subjects and are connected by thin lines for the same subjects. Each thick line segment is plotted with the baseline age at its starting point, 0–4 follow-up years, and 1,400 cm^3 ICV. Female and male data are plotted in red and blue, respectively.

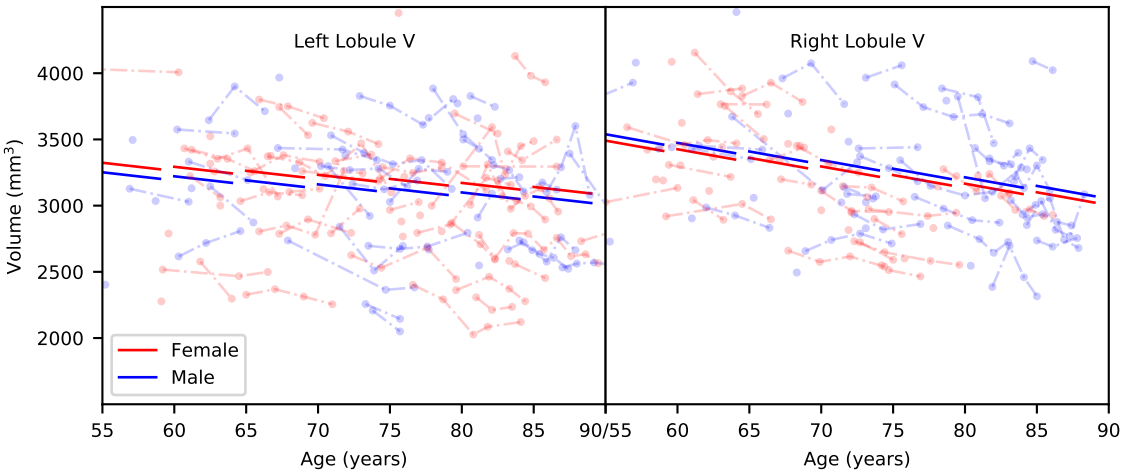


Figure 4-14. Fitted population average trajectories of bilateral *lobules V*. Thick lines indicate the trajectories. Dots indicate volumes of randomly selected subjects and are connected by thin lines for the same subjects. Each thick line segment is plotted with the baseline age at its starting point, 0–4 follow-up years, and 1,400 cm^3 ICV. Female and male data are plotted in red and blue, respectively.

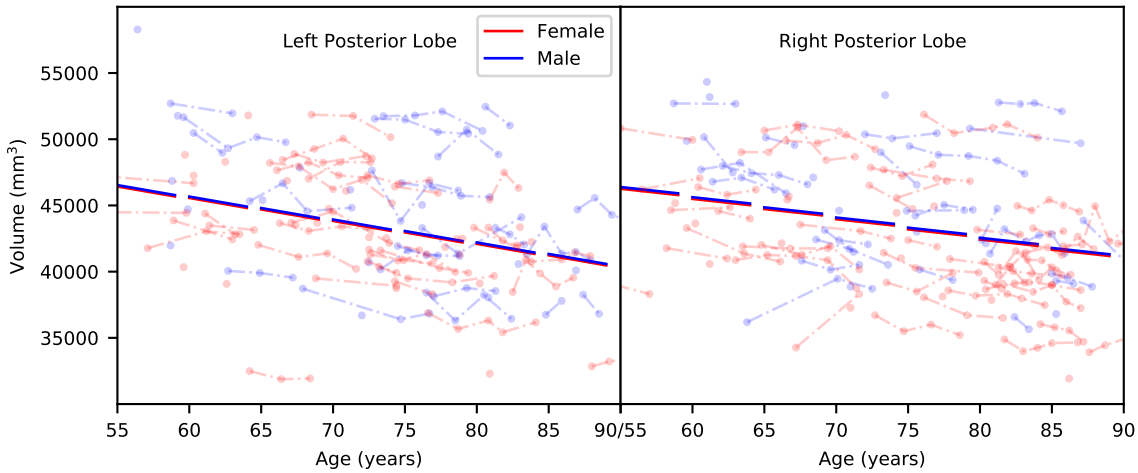


Figure 4-15. Fitted population average trajectories of bilateral *posterior lobes*. Thick lines indicate the trajectories. Dots indicate volumes of randomly selected subjects and are connected by thin lines for the same subjects. Each thick line segment is plotted with the baseline age at its starting point, 0–4 follow-up years, and 1,400 cm^3 ICV. Female and male data are plotted in red and blue, respectively.

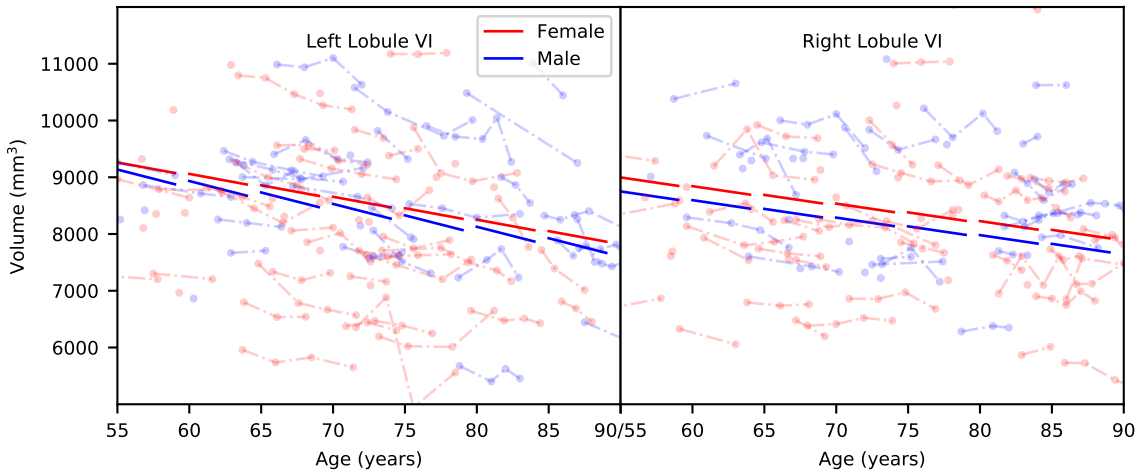


Figure 4-16. Fitted population average trajectories of bilateral *lobules VI*. Thick lines indicate the trajectories. Dots indicate volumes of randomly selected subjects and are connected by thin lines for the same subjects. Each thick line segment is plotted with the baseline age at its starting point, 0–4 follow-up years, and 1,400 cm^3 ICV. Female and male data are plotted in red and blue, respectively.

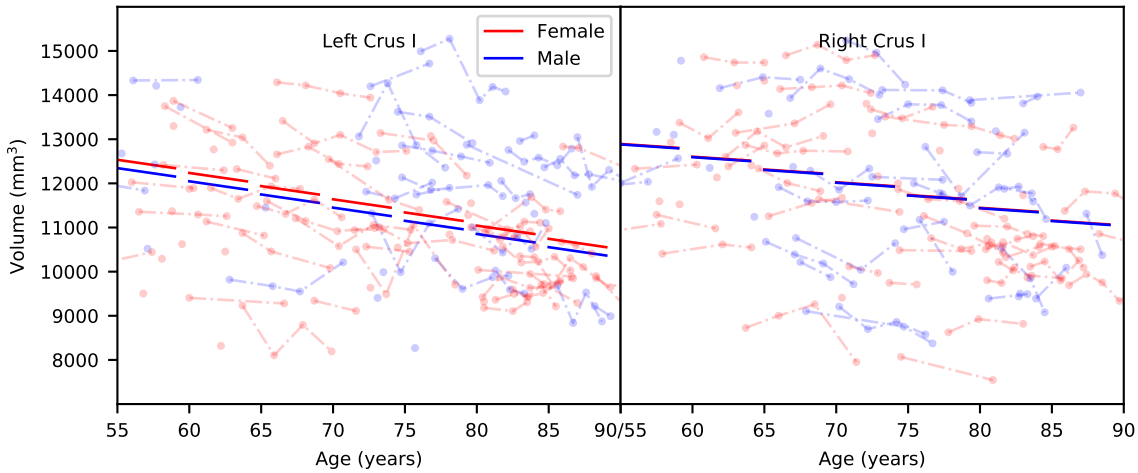


Figure 4-17. Fitted population average trajectories of bilateral *crus I*. Thick lines indicate the trajectories. Dots indicate volumes of randomly selected subjects and are connected by thin lines for the same subjects. Each thick line segment is plotted with the baseline age at its starting point, 0–4 follow-up years, and 1,400 cm^3 ICV. Female and male data are plotted in red and blue, respectively.

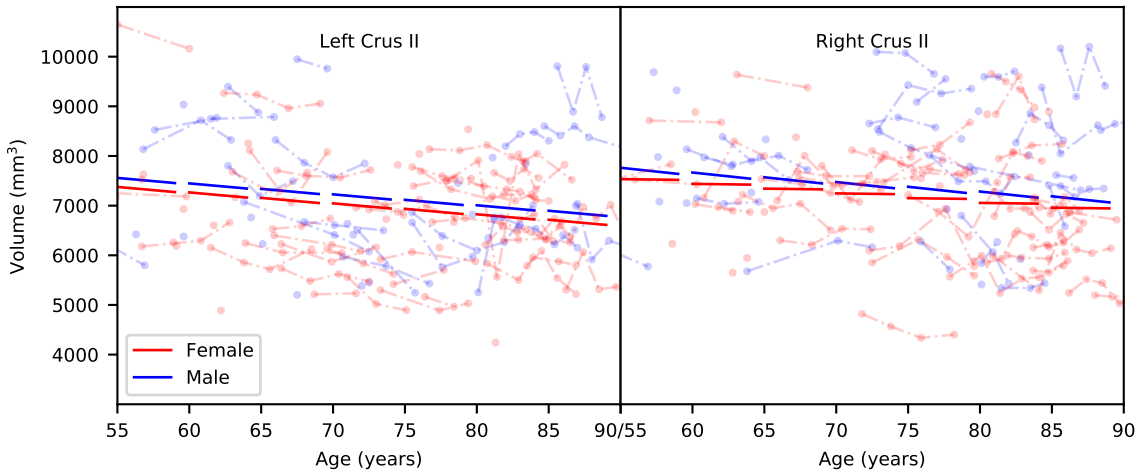


Figure 4-18. Fitted population average trajectories of bilateral *crus II*. Thick lines indicate the trajectories. Dots indicate volumes of randomly selected subjects and are connected by thin lines for the same subjects. Each thick line segment is plotted with the baseline age at its starting point, 0–4 follow-up years, and 1,400 cm^3 ICV. Female and male data are plotted in red and blue, respectively.

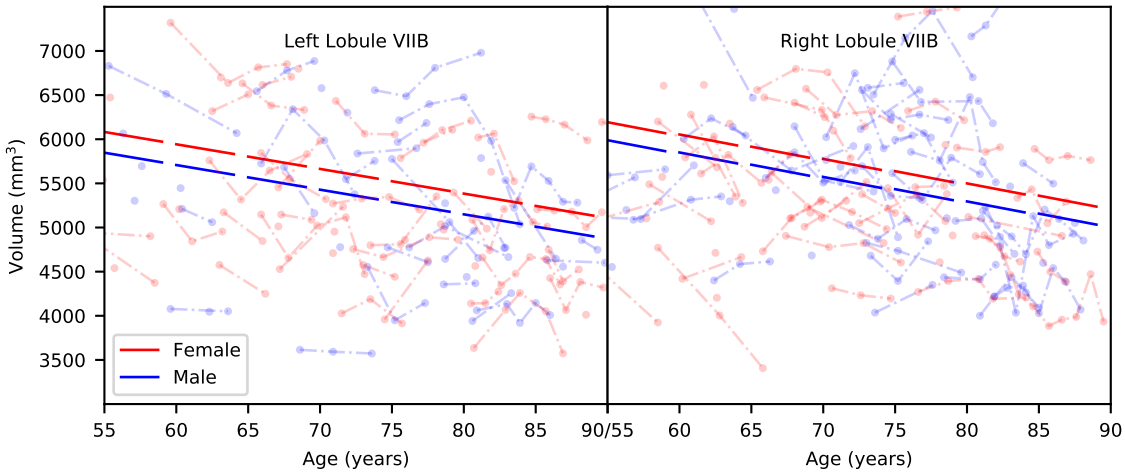


Figure 4-19. Fitted population average trajectories of bilateral *lobules VIIB*. Thick lines indicate the trajectories. Dots indicate volumes of randomly selected subjects and are connected by thin lines for the same subjects. Each thick line segment is plotted with the baseline age at its starting point, 0–4 follow-up years, and 1,400 cm^3 ICV. Female and male data are plotted in red and blue, respectively.

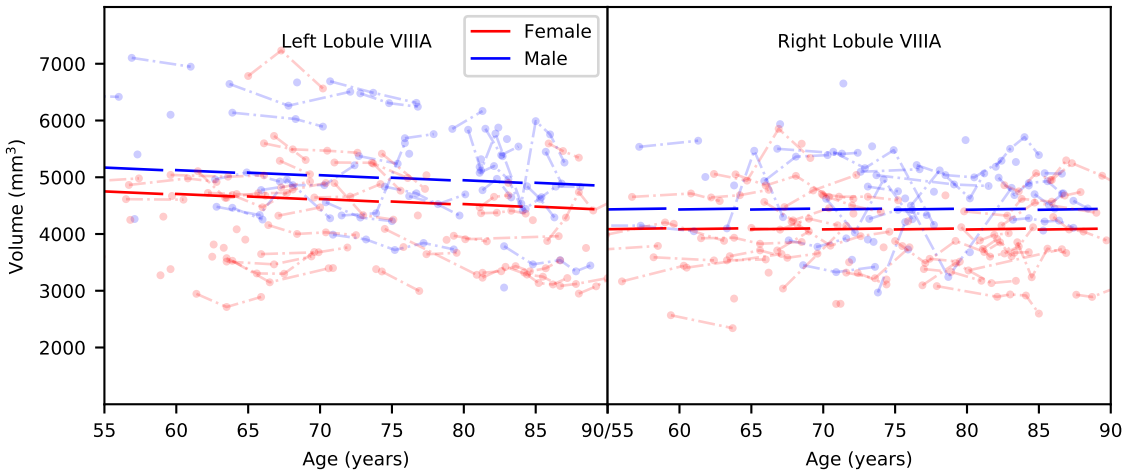


Figure 4-20. Fitted population average trajectories of bilateral *lobules VIIIA*. Thick lines indicate the trajectories. Dots indicate volumes of randomly selected subjects and are connected by thin lines for the same subjects. Each thick line segment is plotted with the baseline age at its starting point, 0–4 follow-up years, and 1,400 cm^3 ICV. Female and male data are plotted in red and blue, respectively.

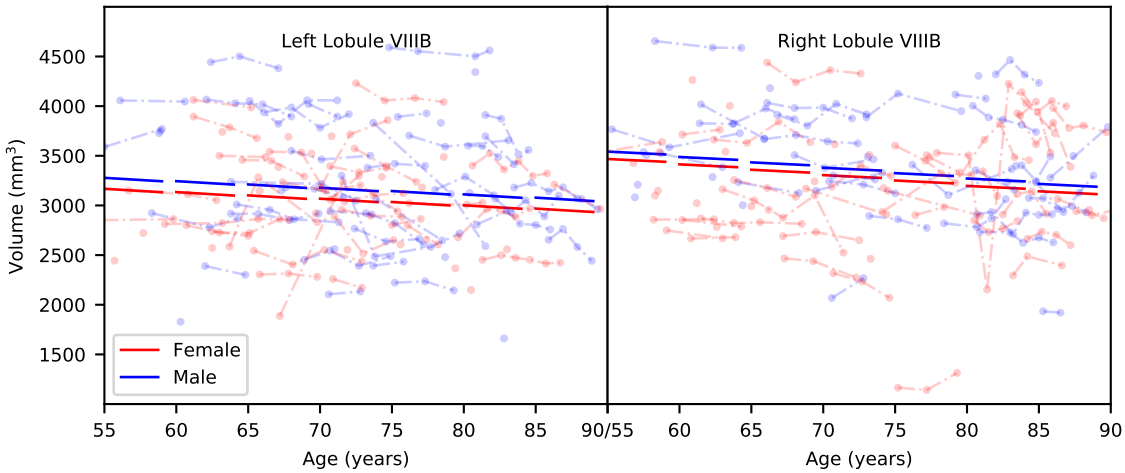


Figure 4-21. Fitted population average trajectories of bilateral *lobules VIIIB*. Thick lines indicate the trajectories. Dots indicate volumes of randomly selected subjects and are connected by thin lines for the same subjects. Each thick line segment is plotted with the baseline age at its starting point, 0–4 follow-up years, and 1,400 cm^3 ICV. Female and male data are plotted in red and blue, respectively.

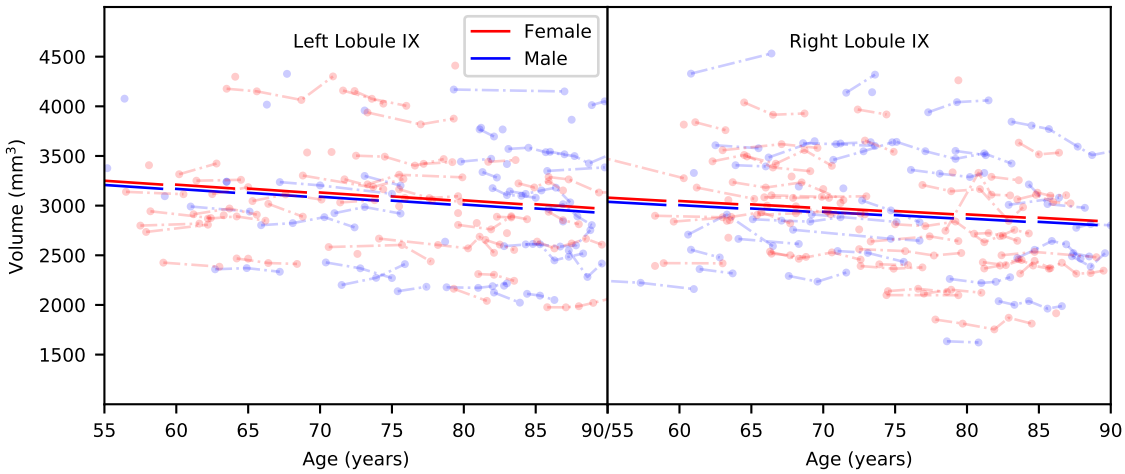


Figure 4-22. Fitted population average trajectories of bilateral *lobules IX*. Thick lines indicate the trajectories. Dots indicate volumes of randomly selected subjects and are connected by thin lines for the same subjects. Each thick line segment is plotted with the baseline age at its starting point, 0–4 follow-up years, and 1,400 cm^3 ICV. Female and male data are plotted in red and blue, respectively.

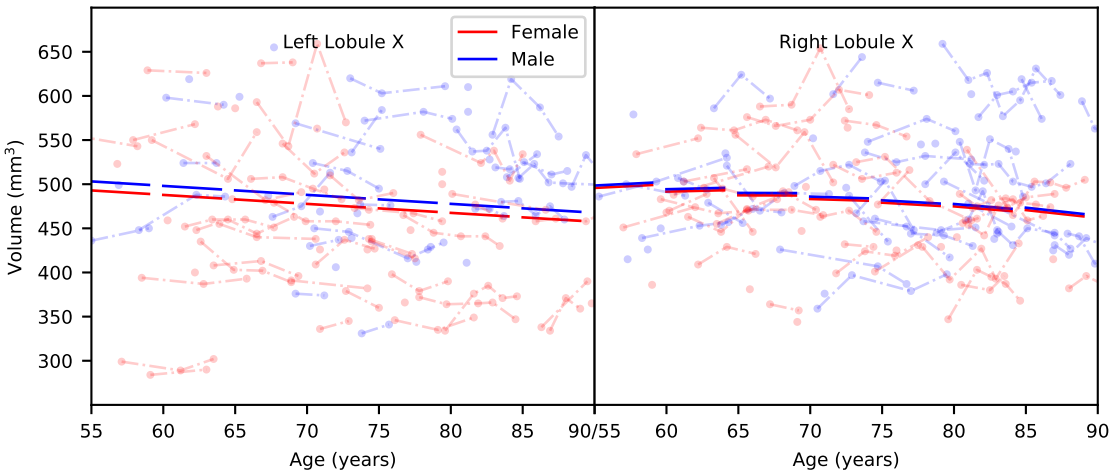


Figure 4-23. Fitted population average trajectories of bilateral *lobules X*. Thick lines indicate the trajectories. Dots indicate volumes of randomly selected subjects and are connected by thin lines for the same subjects. Each thick line segment is plotted with the baseline age at its starting point, 0–4 follow-up years, and 1,400 cm^3 ICV. Female and male data are plotted in red and blue, respectively.

the 28 regions, longitudinal volume loss is greater with advancing baseline age. Five of the 28 regions show statistically significant sex effects at baseline, and men compared with women showed greater longitudinal volume loss in three regions.

We used ACAPULCO to parcellate the cerebellum in this work. Compared with previous methods, such as SUIT [35]—which was used by Bernard & Seidler [69], Bernard *et al.* [12] and Koppelmans *et al.* [13]—and MAGet Brain [18]—which was used by Steele & Chakravarty [70]—our parcellation algorithm has two advantages. First, our algorithm is fully-automatic, and takes approximately a minute to parcellate a cerebellum (without considering image pre-processing such as intensity inhomogeneity correction [27] and MNI space alignment [31]). In comparison, SUIT requires manual intervention and takes about 10 minutes to parcellate a cerebellum image; although MAGeT Brain is fully-automatic, it takes approximately 6 hours for parcellation [18]. These features enable us to process thousands of images in a reasonable amount of time. Second, ACAPULCO has better parcellation accuracy compared with previous

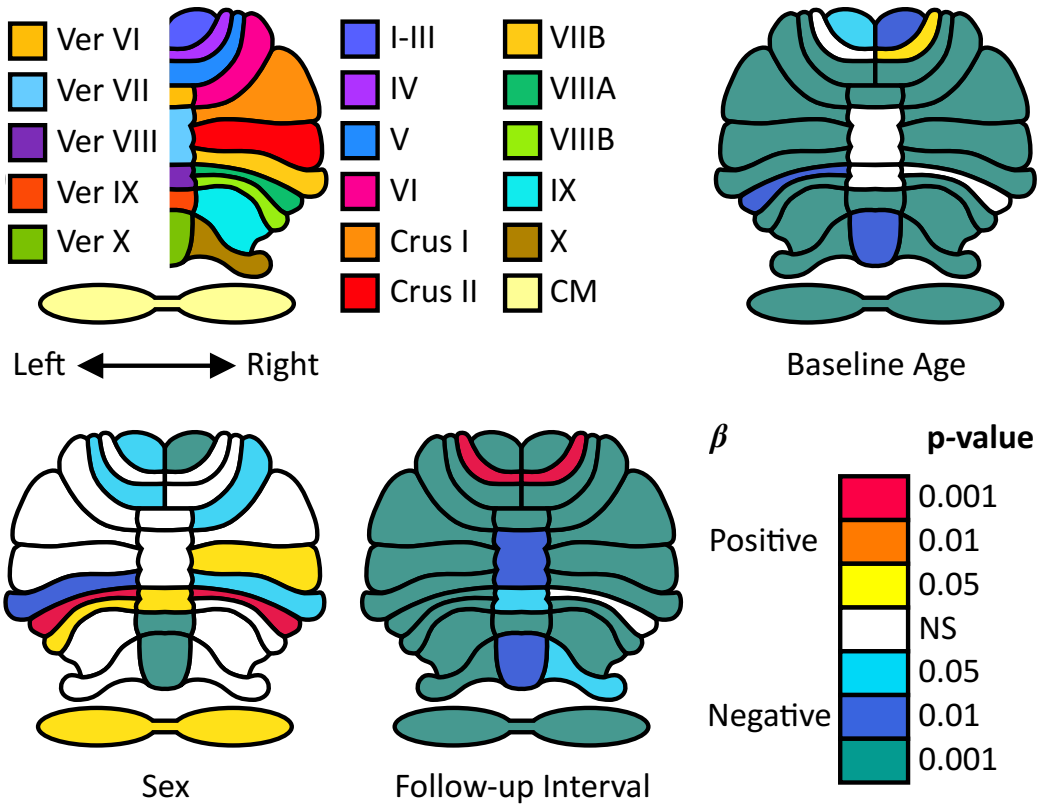


Figure 4-24. *Raw* p-values of baseline age, sex, and follow-up interval for each region. Ver: vermis. CM: corpus medullare. β: fixed coefficients.

methods [19, 65]. A disadvantage of ACAPULCO is that it is sensitive to image contrast. Therefore, we only analyzed MPRAGE images despite also having images acquired with the spoiled gradient echo sequence for earlier visits. Similar to SUIT and MAGeT Brain, the definition of cerebellar regions in our algorithm is based on [5]. However, SUIT and MAGet Brain provide more detailed divisions of the vermis compared to our algorithm, while our algorithm provides more detailed divisions of the anterior lobe than SUIT.

The main benefit of longitudinal analyses is the ability to investigate intra-individual changes over time—i.e., longitudinal changes of the cerebellar sub-regions—and the effects of inter-individual differences on the intra-individual changes—i.e., whether baseline age and sex modify the longitudinal changes in total and sub-regional cerebellar

volumes. Our analyses extend prior findings that investigate inter-individual differences based on cross-sectional analyses or findings of longitudinal changes that are restricted to total or hemispheric cerebellar volumes. Linear mixed-effects models use all available data to estimate longitudinal trajectories. Different subjects can have different numbers of visits—i.e., the data is unbalanced. One of the strengths of the linear mixed-effects model is the ability to appropriately handle such unbalanced data. In addition to subjects with multiple visits, we also included subjects with only one visit. These data would mainly contribute to the cross-sectional effects in our linear mixed-effects models, so excluding them from the analysis will yield similar results for the longitudinal effects. However, the advantage of including participants with a single visit is to improve power for estimation of cross-sectional effects.

We used Bonferroni correction to account for multiple comparisons. However, because our regions are correlated, the hypotheses are correlated as well. Since commonly used approaches to correct for multiple comparisons, including Bonferroni and false discovery rate corrections, do not fully account for these correlations [116], the adjusted p-values are too conservative. Unfortunately, alternatives such as permutation and bootstrapping remain challenging to apply in linear mixed-effects models [117]. This limitation could be investigated in the future.

In contrast to previous work, we conducted longitudinal analyses of a hierarchy of cerebellar sub-regions—cerebellar lobules, lobes, and hemispheres—of non-demented subjects in this age range. Nearly all previous work on cerebellar sub-regions was based on cross-sectional analysis [12, 13, 68–70]. To our knowledge, the single publication of sub-regional longitudinal analysis of the cerebellum [11] focused on cerebellum development in children and adolescence. The work by Raz *et al.* [109–112] also conducted longitudinal analysis but focused on cerebellum hemispheres instead of lobular regions.

In terms of the cross-sectional differences—baseline age and sex—our results are not entirely in agreement with previous work. Bernard & Seidler [69] found significant age differences for the volumes of bilateral lobules I–IV, V, and VI, bilateral crus I, left crus II, vermis VI, and vermis VIIB. However, we do not find significant age differences in left lobules I–III, bilateral lobules IV, or vermis VII, and we find significant age differences in many other regions, perhaps due to greater sample size. Bernard *et al.* [12] modeled the volumes of the vermis and bilateral anterior lobes using logarithmic fits, bilateral crus I using linear fits, and the posterior lobes using quadratic fits. They further showed that females and males could be modeled using different fits for these regions. However, we do not find the coefficient of baseline age is significantly different from zero for the left anterior lobe in our study, and their results with respect to sex are not directly comparable to ours since they did not statistically analyze the differences between the two sexes. Koppelmans *et al.* [13] analyzed the volumes of bilateral lobules I–VI, crus I, crus II–lobules VIIB, lobules VIIIB–IX, lobules X, and the vermis, and found significant age differences except for right crus II–lobules VIIB and bilateral lobules X. In contrast, we find significant age differences in right crus II, right lobule VIIB, and bilateral lobules X, and we do not find significant age difference in left lobules I–III or bilateral lobules IV. Steele & Chakravarty [70] found that bilateral crus II and vermis VI are larger for women, while right lobule V, bilateral lobules VIIIA and VIIIB, and vermis VIIIA and VIIIB are larger for men after dividing each region by the volume of total cerebellar gray matter. In contrast, we incorporated ICV as a covariate in our regressions, and we find no significant sex differences for bilateral crus II, vermis VI, right lobule V, bilateral lobules VIIIB, and vermis VIII. Consistent with findings of Steele & Chakravarty [70], we find that bilateral lobules VIIIA are significantly larger for men. Additionally, we find significant sex differences in right lobules I–III, vermis IX, and vermis X. Note that we found significant increasing volumes of bilateral lobules IV over time. Whether it

is physiologically meaningful or the result of noise from the parcellation algorithm is unclear at present; further investigation is warranted.

Given that this study is the largest study of its nature to date, it is tempting to see this work as the definitive *"Atlas of the Aging Cerebellum"*. We, however, caution that our findings are not consistent with some of the earlier work in the literature. We are still in the discovery phase when it comes to understanding the (aging) cerebellum. We also note that our study differs from prior work with respect to region definition, statistical analysis, and study sample. In addition, our results may lack generalizability because the subjects are highly educated and mostly Caucasian. The imaging visits included in these analyses were restricted to those where participants remained free of cognitive impairment. Future work can investigate whether acceleration of regional cerebellum loss occurs in individuals who ultimately develop cognitive impairment. Future work can also include analysis of the relationships between regional cerebellum volumes and motor and cognitive function tests.

4.6 Summary

In this chapter, we analyzed spatially varying cerebellar patterns with respect to baseline age, follow-up interval, and sex in non-demented subjects older than 50 years. The results show that both age differences and longitudinal declines in cerebellar sub-regional volumes. The effects of age and aging vary across sub-regions. Additionally, longitudinal changes can depend on baseline age and sex. Our findings can help to further understand the trajectories of cerebellum changes during normal aging and provide a normative standard against which effects of disease can be measured.

Chapter 5

Super-Resolving MRI for Better Parcellation

5.1 Introduction

ACAPULCO uses only a T1w MPRAGE image to parcellate the cerebellum. As mentioned in Section 2.4 and shown in Fig. 2-17, ACAPULCO can oversegment the cerebellum into the transverse/sigmoid sinus when the contrast between the GM and sinus is low. However, as shown in Fig. 1-3, the sinuses appear very dark compared to the GM in a T2w image. Therefore, we seek to use both the T1w and T2w images of the same subject to parcellate the cerebellum to prevent such oversegmentation. The MPRAGE sequence that acquires a T1w image uses a 3D acquisition protocol and typically has an isotropic resolution. A T2w image, however, is often acquired with a 2D multi-slice protocol which usually has a lower though-plane resolution than its in-plane resolution. To better use a T2w image, we study super-resolving the through-plane direction of a 2D multi-slice acquisition to have the same resolution as its in-plane direction.

In this chapter, we first present ESPRESO, an algorithm to estimate a point spread function (PSF) that characterizes the through-plane resolution of a 2D multi-slice MRI image (relative to its in-plane direction). We then present S-SMORE, our improved implementation of a super-resolution (SR) algorithm called SMORE [76, 118], to super-resolve the through-plane direction of a 2D multi-slice image. We also incorporated ESPRESO into S-SMORE to improve the SR performance even further. Finally, we show that we can use a super-resolved T2w image in ACAPULCO to better parcellate the cerebellum. ESPRESO is publicly available at https://github.com/shuohan/espreso2. S-SMORE is available at https://github.com/shuohan/ssmore.

5.1.1 2D Multi-Slice Acquisition, Its Resolution, and Slice Profile

A 3D MRI protocol encodes all three spatial axes in k-space (i.e., the frequency domain) to acquire an image. A 2D multi-slice protocol, in contrast, excites multiple slices to cover the whole volume and encodes only two spatial axes of an excited slice, one at a time. To reduce acquisition time while maintaining adequate signal-to-noise ratio (SNR), 2D multi-slice protocols usually acquire "thick" slices, which can sometimes have gaps between them, resulting in lower through-plane than in-plane resolution.

Although the *digital resolution*—i.e., the separation between adjacent voxels—is readily determined from the image header, the *physical resolution*, specified by the PSF of the imaging process, is less straightforward to determine. In 2D multi-slice MRI acquisition protocols, the in-plane physical resolutions are well-approximated by the standard Fourier resolutions that are determined by the Fourier acquisition window extents, but the through-plane physical resolution is determined by the slice profile, which is usually unknown. It is common, in fact, to simply assume that the slice profile is Gaussian with a full width at half maximum (FWHM) that is equal to either the slice

separation [76, 119] or the slice thickness [120–122].

The slice profile quantifies the transverse magnetization of the spin system in the through-plane direction, as produced by the slice selection process of a 2D multi-slice MRI acquisition protocol. With a bell-shaped slice profile, for example, the spins that are closer to the central position of this slice (which is really a "slab" of tissue, but is conventionally called a "slice") exhibit larger transverse magnetization, thus larger signal within the slice [25, 123]. Since the scanner integrates the signals from all the excited spins across the slice, the slice selection process can essentially be modeled as a continuous convolution between the slice profile and the underlying object being imaged, followed by sampling at the slice separation interval. The slice profile therefore acts as a PSF, yielding the slice thickness (as quantified by the FWHM of the PSF), which may be different from the slice separation. We note that 2D multi-slice MRI images are often acquired with slices gaps, where the slice thickness is smaller than the slice separation, and both the slice thickness and slice separation can be quite different between imaging protocols.

5.1.2 ESPRESO to Estimate a Relative Slice Profile for Better SR

Super-resolution (SR) algorithms using CNNs have been successfully applied to MRI images [76, 118, 119, 121, 122, 124–126], benefiting both image visualization and down-stream image processing [119, 122]. Since paired high-resolution (HR) and low-resolution (LR) images are hard to acquire, these algorithms usually simulate LR from acquired HR images and train SR CNNs with supervised learning. For 2D multi-slice MRI images, simulating such LR training images requires their through-plane resolution to be determined. Previous SR methods generally assume the slice profile to be Gaussian with an FWHM equal to either the slice separation [76, 119] or the

slice thickness [120–122]. However, there are several pitfalls to be aware of when doing so. First, the true slice profile, which depends on the specific imaging protocol being used, might not be well approximated by a Gaussian function. Second, using the slice separation as the FWHM is inaccurate when slice gaps or overlaps are present. Third, although it is common and more accurate to use the slice thickness as the slice profile FWHM, the true slice thickness may differ substantially from the recorded value [127] in the MRI scanner or in a medical image file such as the digital imaging and communications in medicine (DICOM); many other commonly used medical image file formats, such as neuroimaging informatics technology initiative (NIfTI), do not even contain the slice thickness in their headers. In addition to these considerations, many SR algorithms [121, 122] are trained for specific slice profiles—thus assuming specific acquisition protocols—so they will not perform as well when applied to images from different protocols. Conventional methods to determine the slice profile for a given acquisition protocol use either a physical phantom [128] or numerical simulations [123, 129]. These methods require either access to the MRI scanner or specific knowledge of the MRI pulse sequence, neither of which may be possible or practical in many cases. Therefore, we seek to estimate the slice profile directly from the digital MRI image itself, without knowing details of the MRI pulse sequence (except that we know it is from a 2D multi-slice acquisition).

ESPRESO (estimating the slice profile for resolution enhancement from a single image only) assumes that the statistics of intensities of an isotropic image are independent of orientation [76, 130]. In other words, 2D image patches acquired from any orientation should look the same from a statistical point of view if their resolutions are the same. Intuitively, if we degrade the in-plane patches in one direction using the true slice profile as the PSF (to be more precise, it is the *relative* PSF between the true slice profile and the in-plane PSF, and we discuss this in Sections 5.2 and 5.5.4), these patches should

be statistically identical to the patches that are degraded in the through-plane direction by the scanner itself. Therefore, to find the slice profile, we simply need to search for the PSF that, when applied to in-plane directions, yields a patch probability distribution that matches that of the through-plane direction. To match these two distributions, we use a modified generative adversarial network (GAN) [52] where the (relative) slice profile is learned as part of the GAN generator. A GAN was used in previous work to estimate the resolution degradation [131] in medical images, but in contrast to that approach, here we explicitly design the GAN generator to yield the PSF itself instead of a degraded image. Our method has similarities with the method of Bell-Kligler *et al.* [132], which estimates the PSF using self-similarity across downsampling scales in natural images. In our work, however, we exploit the nature of 2D multi-slice MRI images, which have both HR and LR directions within the same image. Accordingly, after estimating this PSF, we can create training data with the PSF to super-resolve the image volume in the through-plane direction to yield a volume in which the in-plane and through-plane resolutions are the same.

5.1.3 S-SMORE to Improve SR of a 2D Multi-Slice Acquisition

In this chapter, we also present S-SMORE, which is an improved implementation of SMORE[1]. SMORE [76, 118] can be considered an internally supervised SR algorithm, meaning that it is trained from scratch or fine-tuned for a given 2D multi-slice image without any external training data. The training data for SMORE are created from the HR in-plane slices of the given image using a Gaussian PSF with an FWHM equal to the slice separation. SMORE uses networks with the EDSR architecture [133]. For a small SR factor (≤ 3), an SR network is trained and then applied to the through-plane

[1]To be more precise, S-SMORE is an improved implementation of iterative SMORE [118], or iSMORE, that only uses 2D networks. See Section 5.3.4 for more details.

direction to super-resolve the image to be isotropic; for a large SR factor (> 3), an anti-aliasing (AA) network is also used. In comparison, S-SMORE uses a single network that uses a modified RCAN [77] architecture with PixelShuffle [78] to replace the two networks in SMORE. It has better accuracy in a shorter training time compared to the original implementation. We also incorporate ESPRESO into S-SMORE to create more faithful training data to further improve S-SMORE.

5.2 Theory of ESPRESO

Let the object being imaged in a scanner be represented by a continuous function $f(x, y, z)$ in the spatial domain, where $(x, y, z) \in \Omega$ and $\Omega \subset \mathbb{R}^3$. A patch from f can be regarded as a "fragment" of f which is hypothetically (since we cannot directly observe f) sampled on a local Cartesian grid at an arbitrary location within Ω. Suppose a patch with an orientation $\boldsymbol{d}$ is $f_{\boldsymbol{d}}(x, y, z)$, where $(x, y, z) \in \Omega_{\boldsymbol{d}}$, and $\Omega_{\boldsymbol{d}} \in \Omega$ contains the coordinates of the grid of $f_{\boldsymbol{d}}$. For specific grid spacing, spatial size, and orientation, we uniformly sample $\Omega_{\boldsymbol{d}}$ within Ω, and the resulting patches can form a probability distribution (note that such a patch takes on values at all possible locations of the given object f instead of values across different objects). The fundamental assumption of ESPRESO is that f is "isotropic" in the sense that such patches with different orientations have the same probability distribution. Now consider a 2D patch f_{xz} which is sampled in the x-z planes of f (the first and second dimensions are along the x and z axes, respectively), and a 2D patch f_{zx} which is sampled in the z-x planes of f (the first and second dimensions are along the z and x axes, respectively). Suppose these two kinds of patches have the same grid spacing and spatial size, our assumption indicates that f_{xz} and f_{zx} are random vectors that have the same probability distribution (the randomness only comes from the random sampling locations within the given object f).

We note that this assumption of "isotropy" empirically holds for brain MRI images but requires the spatial size of these patches to be small enough.

We do not directly observe the continuous f. Instead, we acquire a digital image I of f from the scanner using a 2D multi-slice acquisition. Without loss of generality, we assume x and y are HR in-plane axes, and z is the LR through-plane axis. In the following, we use the subscript l to denote LR and h to denote HR. The slice selection process in a 2D multi-slice acquisition can be modeled as first convolving f with the 1D slice profile $p_l(z)$ and then sampling the slices with the step size s_l. To simplify the problem, we further assume that the x and y axes have the same resolution (we comment on the case of different x and y resolutions in Section 5.5.5). Therefore, both the frequency and phase encoding can be modeled as convolutions with the same PSF p_h, followed by a sampling step of interval s_h (note that, for example, Cartesian k-space encoding is equivalent to convolution with a sinc PSF in the image domain followed by sampling). The acquired digital image I can then be expressed as

$$I = \{f *_z p_l *_x p_h *_y p_h\} \downarrow_{(s_l, s_h, s_h)}, \tag{5.1}$$

where $*_x$, $*_y$, and $*_z$ denote 1D convolutions along the x, y, and z axes, respectively, and $\downarrow$ denotes sampling.

In practice, we can only observe digital patches from I instead of f_{xz} and f_{zx} from f. These two corresponding digital patches can be expressed as

$$\begin{aligned} I_{hl} &= \{f_{xz} *_1 p_h *_2 p_l\} \downarrow_{(s_h, s_l)} \text{ and} \\ I_{lh} &= \{f_{zx} *_1 p_l *_2 p_h\} \downarrow_{(s_l, s_h)}, \end{aligned} \tag{5.2}$$

where $*_1$ and $*_2$ denote 1D convolutions along the first and second dimensions of the patches, respectively. For clarity, we use the subscript hl to indicate HR and LR in the

first and second dimensions, respectively, and the subscript lh to indicate the opposite. As we can see from Eq. (5.2), the digital patches I_{hl} and I_{lh} have different resolutions in both their first and second dimensions. Therefore, unlike f_{xz} and f_{zx}, I_{hl} and I_{lh} cannot be assumed to have the same probability distribution.

Since the through-plane direction has lower resolution than the in-plane direction, we assume that there exists a PSF p as the "difference" between p_h and p_l. The PSF p represents an additional blur—a relative slice profile—to make in-plane directions have the same resolution as that of the through-plane direction. In addition, since the underlying continuous patches f_{xz} and f_{zx} share the same distribution (by assumption), we know that the patches

$$\begin{aligned} \tilde{I}_{hl} &= \{I_{hl} *_1 p\} \downarrow_{(s,1)} \text{ and} \\ \tilde{I}_{lh} &= \{I_{lh} *_2 p\} \downarrow_{(1,s)}, \end{aligned} \tag{5.3}$$

where $s = s_l/s_h$, must come from the same distribution (note that, with a slight abuse of notation, the convolutions $*_1$ and $*_2$ and the sampling $\downarrow$ here are digital operators as opposed to the continuous convolutions and sampling used in Eq. (5.2)). Accordingly, we can find p by matching the distributions of $\tilde{I}_{hl}$ and $\tilde{I}_{lh}$. Finding p is sufficient to super-resolve the through-plane direction to have the same resolution as the in-plane direction, since the training pairs used to super-resolve the through-plane direction in SMORE are created from the in-plane slices. We comment on estimating p_l from p in Section 5.5.4 of the Discussion.

5.3 Methods

5.3.1 ESPRESO Flowchart

We use a modified GAN [52], as shown in Fig. 5-1, to match the patch distributions of $\tilde{I}_{hl}$ and $\tilde{I}_{lh}$. A conventional GAN learns a generator network to output images that match the distribution of true images. In contrast, ESPRESO matches the distributions of two sets of generated images that are blurred and downsampled along different axes. Our generator network learns the blur—i.e., the relative slice profile p—to match these two probability distributions of patches. Suppose that we randomly select a 2D patch from the image volume where the horizontal direction of this patch is the HR x or y axis, and its vertical direction is the LR z axis. In ESPRESO, the generator network (G in the flowchart) blurs and downsamples the horizontal direction of this patch. If we transpose the resulting patch from G, as indicated by T in top row of the flowchart, its horizontal direction becomes the real LR. On the other hand, if we do not transpose the patch, as shown in the bottom row of the flowchart, its horizontal direction remains the generated LR. Our discriminator network (D in the flowchart) then judges whether the horizontal direction of an incoming patch is fake or real and outputs a pixelwise probability map. We train G and D adversarially until they reach equilibrium, which happens when the patch distributions match each other.

5.3.2 ESPRESO Network Architectures

As shown in Fig. 5-2, instead of directly applying a CNN to the input patch, our generator network first calculates a 1D function as the estimated relative slice profile, p, convolves the input patch with it, then downsamples the blurred patch to the desired digital resolution. By doing so, we can guarantee that this process respects our model in

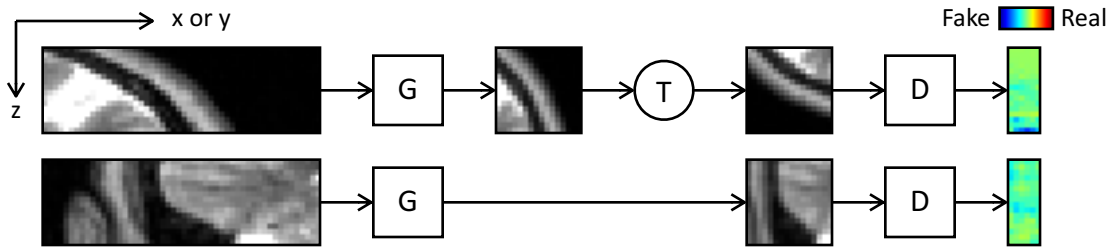


Figure 5-1. Flowchart of ESPRESO. 2D patches are randomly sampled from the image volume. The generator network, G, blurs and then downsamples a patch along the horizontal direction (the x or y axis). A patch can be transposed, as by T in the top row, after which its horizontal direction becomes the real low-resolution (LR). The discriminator network, D, checks whether the horizontal direction is real LR (i.e., from the z axis) or fake (generated from G).

Eq. (5.3) and can also impose regularization to p. In our generator network, we apply two 1D convolutional layers with a ReLU layer in between on top of a trainable "embedding vector" to calculate p. There are two reasons for designing the generator this way. First, according to Bell-Kligler *et al.* [132], it is easier to optimize multiple layers of convolutions instead of a single one. Second, we can use an $l2$ weight decay to encourage smoothness of p. This is inspired by the deep image prior [134] because, as shown in Cheng *et al.* [135], weight decay can encourage local correlation of the output of a deep image prior network. We also apply a softmax operator to the output to guarantee that p has positive values and sums to 1. The number of channels of the embedded vector is 256, and the output numbers of channels for the two convolutional layers are 256 and 1. The kernel size of both convolution layers is 3. The length of the embedded vector is equal to the length of p plus 4 since we do not use padding in the following convolutional layers. We set the length of p equal to 21 throughout all our experiments, and it is changeable in our algorithm if a different size is desired. Since there is no padding to the input patch before the convolution with p, its horizontal spatial size is reduced by 20. After the convolution between the input patch and p, it is then downsampled by s as in Eq. (5.3) using cubic interpolation. We note that in our current

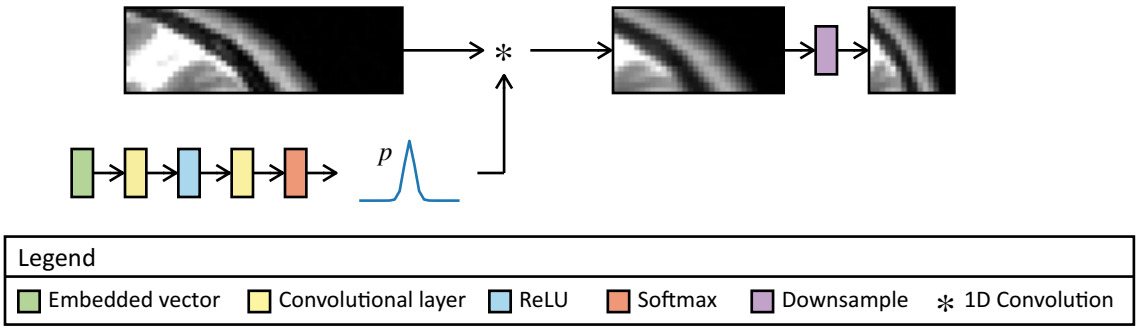


Figure 5-2. Architecture of the generator network G. A series of 1D convolutional and ReLU layers are applied to a trainable embedded vector. A softmax is also applied, so the estimated relative slice profile has positive values and sums to 1. The input patch is convolved with the estimated relative slice profile and then downsampled with cubic interpolation along its horizontal direction.

setting, p has the same digital resolution as the horizontal direction of input patches (i.e., in-plane direction of the whole image). This is sufficient for creating training data for SR, but may reduce the precision of calculating its FWHM and the accuracy in recovering the real p_l from p. We comment on this in Section 5.5.4.

The architecture of ESPRESO's discriminator network is shown in Fig. 5-3. It is composed of five 1D convolutional layers interleaved with leaky ReLUs [136], and we use spectral normalization [137] to stabilize the training. The negative slope of each leaky ReLU is set to 0.1 as in Miyato *et al.* [137]. The number of output channels is 64 for all convolutional layers except for the last one. The kernel size of all convolutional layers is 3, resulting in a 11-pixel receptive field. Since we do not use padding, the horizontal spatial size of the resulting probability map is reduced by 10. The discriminator uses a sigmoid to convert the network output to a probability map. Since we use binary cross entropy with logits to train our network (see Section 5.3.3), the sigmoid operator is incorporated in the loss function.

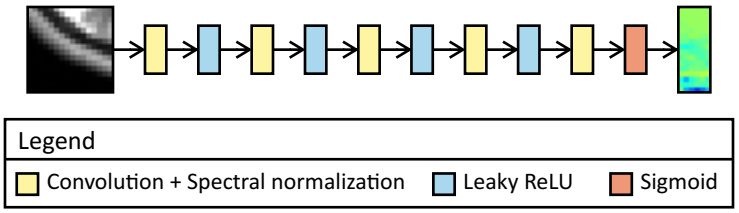


Figure 5-3. Architecture of the discriminator network D. A series of 1D convolutional layers with spectral normalization and leaky ReLU layers are applied to the horizontal direction of the input patch to generate a probability map of whether the horizontal direction is real or fake LR.

5.3.3 ESPRESO Loss Functions and Training

To train our discriminator network to match the distributions of the two sets of generated patches, we modify the loss function of a conventional GAN [52] as follows,

$$L_D = -\frac{1}{2M}\sum_{m=1}^{M} \log D(G(I_m)^T) + \log(1 - D(G(I'_m))), \tag{5.4}$$

where I_m and I'_m represent 2D patches independently sampled from the image volume I, and M is the mini-batch size. Accordingly, the adversarial loss function for our generator network is

$$L_{adv} = -L_D. \tag{5.5}$$

Since there are potentially many PSFs p that will cause the patch distributions to match, it is essential to provide regularization on p to encourage the type of solution that we expect and also to stabilize training. As noted in Section 5.3.2, we use a softmax operator to guarantee p to be positive and add to unity, and we use weight decay to encourage p to be smooth. To encourage p to be close to zero at its edges, we use the loss

$$L_b = p_1 + p_2 + p_N + p_{N-1}, \tag{5.6}$$

where N is the length of p, and p_i is the i^{th} element of p. To encourage p to have a single peak, we use the loss

$$\begin{aligned} L_p = & \sum_{i=2}^{(N+1)/2} \max\{p_{i-1} - p_i, 0\} \\ & + \sum_{(N+3)/2}^{N} \max\{p_i - p_{i-1}, 0\}, \end{aligned} \tag{5.7}$$

which penalizes negative derivatives on the left half of p and positive derivatives on the right half of p (note that N is an odd integer in all our experiments).

The complete loss function for our generator is a weighted summation of these terms:

$$L_G = L_{adv} + \lambda_b L_b + \lambda_p L_p + \lambda_{wd} L_{wd}, \tag{5.8}$$

where L_{wd} is an $l2$ weight decay applied only to the generator network, $\lambda_b = 10$, $\lambda_p = 1$, and $\lambda_{wd} = 0.005$. To further constrain p, we average p with its flipped version to ensure symmetry.

We use the Adam optimizer [55] with $\beta_1 = 0.5$ and $\beta_2 = 0.999$ to train ESPRESO, and we use the one-cycle learning rate policy [138] to speed up the training. The maximum learning rate is 0.001, and the number of iterations is 2,000. Other parameters of this learning rate policy are the defaults in PyTorch 1.8.1. The patch size along the LR z axis is 16, and the size along the HR x or y axes are set such that it is reduced to 16 after the generator G (resulting in a square patch that is input to the discriminator). The size of a mini-batch is 128. To speed up and stabilize the training, we also use a warm-up training phase before the adversarial training. The warm-up phase takes 80 iterations and is trained in a supervised fashion to match an impulse slice profile. In other words, the generator network tries learning to generate an output that is the same as its input image patch; we stop the warm-up before it fully converges, after which the

slice profile is somewhere between a “flat” function (since the weights of the generator network are randomly initialized) and an impulse function. The learning rate during the warm-up is 0.0001 without a learning rate policy. We also use flipping to augment the image patches in both the warm-up and adversarial training. The hyper-parameters of ESPRESO were tuned using the data described in Section 5.4.1.

5.3.4 S-SMORE

S-SMORE is an improved implementation of iSMORE [118]. iSMORE is an extension of SMORE [76], which uses multiple “phases” (as explained below) to gradually improve SR; it is the same as SMORE when the number of phases is 1. A simplified pseudocode for iSMORE (for a SR factor $>$ 3) is shown in Algorithm 5.1. Suppose I_0 is the input volume acquired from a 2D multi-slice protocol. Without loss of generality, we assume that x and y are the HR in-plane axes of I_0, and z is the LR through-plane axis of I_0. We further assume that the x and y axes have the same resolution as in Section 5.2. iSMORE super-resolves the z axis of I_0 to have the same resolution as its x-y plane. The SR scale factor s is thus chosen to be the ratio—which can be a non-integer value—between the voxel sizes of the z axis and the x or y axes. iSMORE first interpolates the z axis of I_0 with the scale factor s to have an isotropic *digital* resolution. It then creates training data as

$$\begin{aligned} I_{lh} &= I_{hh} * p, \\ I'_{lh} &= I_{lh} \downarrow_s \uparrow_s, \end{aligned} \tag{5.9}$$

where I_{hh} is a 2D HR patch that is randomly extracted from the x-y slices of the interpolated I_0 (the subscript hh indicates that its both directions are HR), I_{lh} and I'_{lh} are the resulting patches whose resolutions along their first dimensions are degraded (the

Algorithm 5.1: A simplified pseudocode for iSMORE.

Data: An image volume I_0, its through-plane PSF p, and the scale factor s.
Result: A super-resolved image volume I_n.
$I_0 \leftarrow$ Interpolate the through-plane direction of I_0 with s;
for i *in 1, 2, ..., n phases* **do**
    Simulate training data pairs from in-plane slices of I_{i-1} using Eq. (5.9);
    Train the anti-aliasing network for k_i iterations;
    Train the super-resolution network for k_i' iterations;
    $I_i \leftarrow$ Apply the networks to the through-plane direction of I_0;
end

subscript lh indicates that the first and second dimensions are LR and HR, respectively), $*$ is 1D convolution, p is the z-axis PSF, $\downarrow$ indicates downsampling, and $\uparrow$ indicates upsampling. The PSF p models the *physical* resolution degradation of the z axis compared to the x and y axes of I_0 (see Section 5.2 for more details). In Zhao *et al.* [76], p is assumed to be a Gaussian function with an FWHM equal to the slice separation (i.e., the voxel size along the z axis). iSMORE uses the pair of I'_{lh} and I_{lh} to train the AA network to remove the aliasing caused by sampling with s and uses the pair of I'_{lh} and I_{hh} to train the SR network. These two networks are sequentially applied to the z-x or z-y slices of the interpolated I_0 to super-resolve this volume.

Although the x-y slices of I_0 are considered HR, they are typically "thick" slices since their thicknesses are determined by the physical resolution along the z axis. In comparison, the z-x and z-y slices are "thin" since their thicknesses are determined by the physical resolutions along the y and x axes, respectively. This creates a discrepancy between the the training data—i.e., the x-y slices—and the testing data—i.e., the z-x and z-y slices. iSMORE performs multiple training phases (i.e., the for-loop in Algorithm 5.1; we use the word "phase" to refer to an iteration of this for-loop) to address this discrepancy. Suppose the number of phases is n. In each phase i, the super-resolved volume I_{i-1} from the previous phase $i-1$ is used to simulate training

Algorithm 5.2: A simplified pseudocode for S-SMORE. Key differences from Algorithm 5.1 are highlighted in blue.

Data: An image volume I_0, its through-plane PSF p, and the scale factor s.
Result: A super-resolved image volume I_n.
$I \leftarrow$ Interpolate the through-plane direction of I_0 with $s/\lfloor s \rfloor$;
for i *in 1, 2, ..., n phases* **do**
    Simulate training data pairs from in-plane slices of I_{i-1} using Eq. (5.10);
    Train the network with PixelShuffle scaling factor $\lfloor s \rfloor$ for k_i iterations;
    $I_i \leftarrow$ Apply the network to the through-plane direction of I_0;
end

data (except for the first phase where the interpolated I_0 is used). After each phase, the x-y slices get thinner, thus serving as better training data. Note that the trained networks are always applied to the interpolated I_0 instead of the super-resolved ones from previous phases to avoid accumulated errors. In practice, I_0 is rotated around the z axis with multiple rotation angles, and Fourier burst accumulation (FBA) [139] is used to combine the super-resolved volumes from each of these angles to improve accuracy.

We note that iSMORE uses the pairs of (I'_{lh}, I_{lh}) and (I'_{lh}, I_{hh}) to train the AA and the SR networks, respectively, but applies them sequentially to the interpolated I_0 during inference. We argue that this can increase the discrepancy between the training and testing data. Additionally, since iSMORE trains from scratch or fine-tunes the networks for each individual image volume, using two networks is computationally intensive. Therefore, the main motivation of S-SMORE is to unify these two networks into a single one. According to our preliminary experiments, we found that applying a single network to patches of an *interpolated* image has an effectively small receptive field (since interpolation does not add new information) when the scale factor s is large. We therefore would like to use PixelShuffle [78] within the network instead of interpolating the image beforehand to increase the receptive field. Since PixelShuffle rearranges the channels into spatial dimensions to increase the spatial size of incoming feature maps,

it can only handle an integer upsampling factor. Therefore, we use the floor value $\lfloor s \rfloor$ in our PixelShuffle layer and use the residual $s/\lfloor s \rfloor$ to interpolate I_0 before applying the network. Accordingly,

$$I_{lh} = (I_{hh} * p) \downarrow_s \uparrow_{s/\lfloor s \rfloor} \tag{5.10}$$

is used to create training data for S-SMORE. We note that it is important, and yet easily overlooked, to maintain correct sampling steps when outputting integer shapes from interpolations $\downarrow_s$ and $\uparrow_{s/\lfloor s \rfloor}$. See Section 5.3.5 for a more detailed discussion.

A simplified pseudocode for S-SMORE is shown in Algorithm 5.2. We use a modified RCAN network in S-SMORE (see Fig. 5-4). Our network has 2 residual groups, each of which has 8 residual channel attention blocks. The patch size is 32 by 32 or equal to the number of slices if the input image has fewer slices. We train S-SMORE for 4 phases. The first phase has 20,000 iterations, and the following phases have 2,000 iterations. The size of a mini-batch in each iteration is 32. Adam [55] is used with a learning rate of 0.0002, and other parameters of Adam are the defaults in PyTorch 1.8.1. We pick the best SR image I_i in each phase according to a set of validation data. The validation data include 128 patches that are extracted from the 45°-rotated x-y slices of I_{i-1} (we note that these 45°-rotated slices do not have the same physical resolution as non-rotated x-y slices, which is discussed in Section 5.5.5). The training data are extracted from x-y slices that are either not rotated or rotated with 90°, and flipping data augmentation is also used. The trained network is applied to both the z-x and z-y planes, and these two super-resolved volumes are averaged together which is equivalent to FBA with equal weights. We note that iSMORE has an option to use 3D networks after the first training phase (its first phase still uses 2D networks as in SMORE), but the S-SMORE network is 2D for all phases since it is faster to train. Accordingly, we compare SMORE/iSMORE with S-SMORE in Section 5.4.2 only after the first training phase.

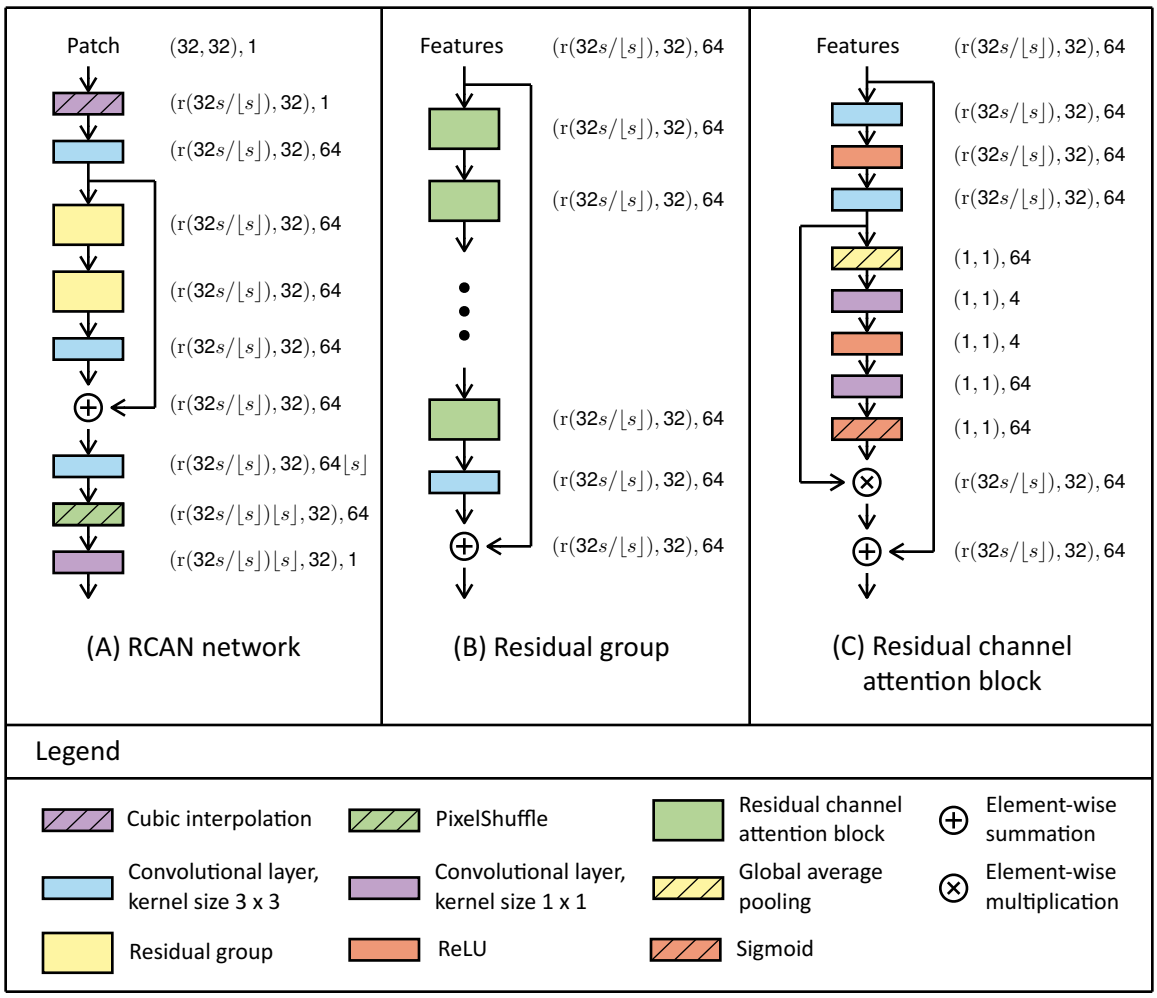


Figure 5-4. The network architecture of S-SMORE. (A) shows the architecture of our modified RCAN network. (B) shows the architecture of a residual group (the yellow boxes in (A)). (C) shows the architecture of a residual channel attention block (the green boxes in (B)). The spatial size of each input or the output of each block/layer is shown in parentheses, and the number of channels is shown next to the spatial size. $\lfloor \cdot \rfloor$ is the floor operator. $\mathrm{r}(\cdot)$ is the rounding operator.

5.3.5 Sampling Interval and Field of View in Interpolation

A digital image can be viewed as a set of values on an evenly spaced grid. Accordingly, interpolation of this image can be viewed as sampling values from a new grid with a different spacing. We refer to the extent and position of an image grid as its *field of*

view (FOV) and refer to the spacing of this grid as its *sampling interval*. It is important to use a correct sampling interval when interpolating a medical image to avoid undesirable scaling; otherwise, this image will not be able to align with another image from the same subject, and measurements from it will also be affected. Meanwhile, we usually want the FOVs before and after an interpolation to have (approximately) the same size and also want them to center around the same position to avoid shifting the contents.

The following discussion focuses on 1D, and it can be easily extended to higher dimensions. The sampling intervals before and after an interpolation are denoted by Δs and $\Delta s'$, respectively, and the numbers of pixels before and after the interpolation are denoted by N and N', respectively. We further use r to denote the ratio $\Delta s'/\Delta s$. Before diving into more details, we first introduce a coordinate system to define the position of a sampling grid. For the image before interpolation, we assume that the coordinates of the centers of its left-most and the right-most pixels are 0 and $(N-1)\Delta s$, respectively; therefore, the whole image spans from $x_s = -0.5\Delta s$ to $x_e = (N-0.5)\Delta s$ when accounting for pixel borders. We further assume that an image represents values at the coordinates of pixel centers. In other words, the i^{th} element of the image represents the value at coordinate $x_i = i\Delta s$ for $i = 0, 1, \ldots, N-1$. See Fig. 5-5(A) for an example.

Since the interpolated image must have an integer number of pixels, we use

$$N' = \text{round}\left(\frac{N\Delta s}{\Delta s'}\right) = \text{round}\left(\frac{N}{r}\right), \tag{5.11}$$

where $r = \Delta s'/\Delta s$. However, we find that some software and libraries, such as `scipy.ndimage.zoom` version 1.7.3 (the original implementations of SMROE and iSMORE use an early version of this function), modifies the ratio r when rounding the new number of pixels. Specifically, `scipy.ndimage.zoom` anchors the positions of the first and last pixels and changes r to $(N-1)/(N'-1)$. See Fig. 5-5(B) for an illustration.

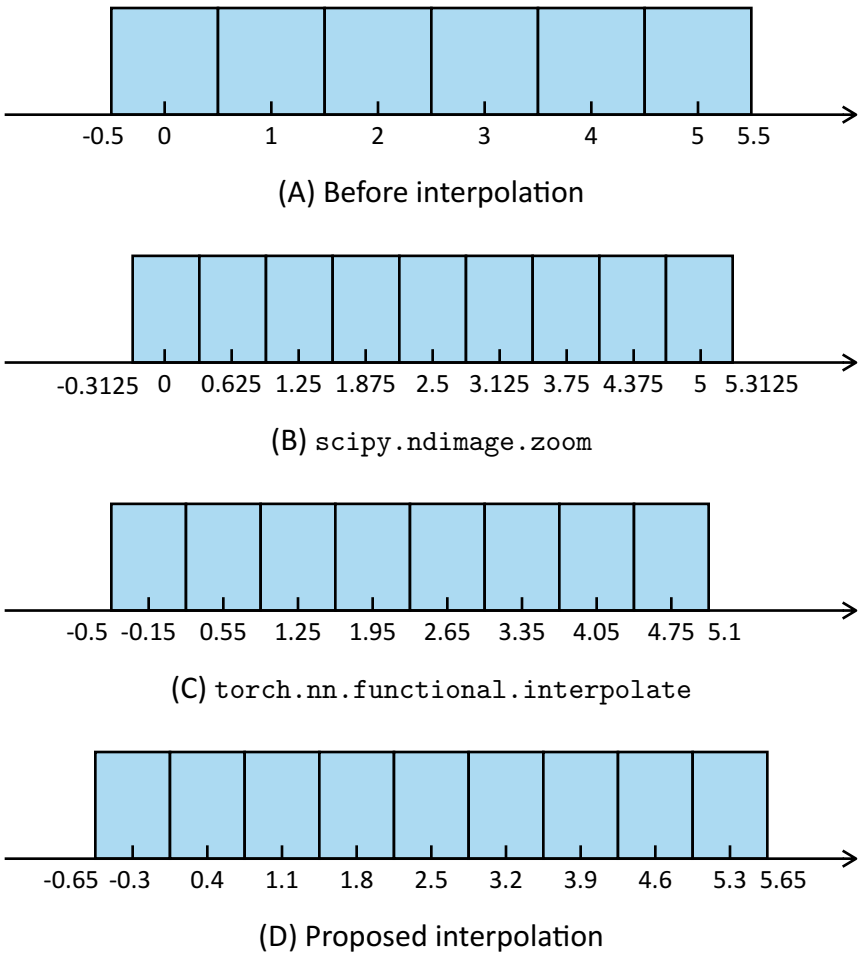


Figure 5-5. Illustrations of three interpolation methods. (A) shows the 1D image before interpolation, (B) shows `scipy.ndimage.zoom`, (C) shows `torch.nn.functional.interpolate`, and (D) shows the proposed interpolation. The blue boxes represent pixels. Their coordinates are marked on the horizontal axes. The number of pixels and the sampling interval before the interpolations are $N = 6$ and $\Delta s = 1$, respectively. We use $r = \Delta s'/\Delta s = 0.7$ to interpolate this image. Note that (B) does not preserve the sampling interval, and the FOV in (C) do not center around the same position. See https://github.com/shuohan/resize/blob/master/tests/compare_interp.py for a code snippet of using these interpolation methods.

In the following, we describe a way to preserve the sampling ratio r when performing an interpolation, as illustrated in Fig. 5-5(D). Since the FOVs before and after an interpolation should center around the same position, we calculate the start and the end of the FOV after the interpolation as

$$\begin{aligned} x'_s &= \left(\frac{N - rN'}{2} - 0.5\right) \Delta s \quad \text{and} \\ x'_e &= \left(\frac{N + rN'}{2} - 0.5\right) \Delta s \,, \end{aligned} \tag{5.12}$$

respectively. The coordinates x'_i for $i = 0, 1, \ldots, N' - 1$ of the interpolated image are

$$x'_i = x'_s + (0.5 + i) r \Delta s \,. \tag{5.13}$$

As a sanity check,

$$\begin{aligned} x'_{N'-1} &= \left(\frac{N - rN'}{2} - 0.5\right) \Delta s + (0.5 + N' - 1) r \Delta s \\ &= \left(\frac{N + rN'}{2} - 0.5\right) \Delta s - 0.5 r \Delta s = x'_e - 0.5 r \Delta s \,, \end{aligned} \tag{5.14}$$

which is $0.5r\Delta s = 0.5\Delta s'$ (half an interpolated pixel) away from x'_e. We also have $x'_i - x'_{i-1} = r\Delta s = \Delta s'$, which preserves the sampling interval. In comparison, `torch.nn.funcitonal.interpolate` version 1.8.1 starts its FOV from -0.5 instead of the x'_s in Eq. (5.12); therefore, this function introduces a slight shift of the contents. See Fig. 5-5(C) for an illustration.

The original implementations of SMORE [76] and iSMORE [118] use the function `scipy.ndimage.zoom`, so they introduce an undesirable scaling of its SR result in some cases. A newer implementation of SMORE (which is compared to S-SMORE in Section 5.4.2) and our S-SMORE both use the interpolation principles described here.

5.3.6 SR for Better Cerebellum Parcellation

In this chapter, we also propose to use paired T1w and T2w images in ACAPULCO (see Chapter 2 for more details of ACAPULCO) to improve cerebellum parcellation. Since many T2w images are acquired with 2D multi-slice protocols, we use S-SMORE with ESPRESO to super-resolve them. Two variants of the parcellating network of ACAPULCO are proposed to incorporate T2w images. Both variants use a pair of T1w and T2w images as a dual-channel input. The first one (Method 1) directly outputs a parcellation. In contrast, the second method (Method 2) outputs a cerebellum mask (i.e., the union all sub-regions) which is used to intersect the parcellation from the original ACAPULCO (which only takes as input the T1w image); we note that only the labels near the transverse/sigmoid sinuses are intersected with this mask to avoid unnecessary false negatives in other regions. In the T dataset, these regions are left and right lobules I–III through VIIIB. See Fig. 5-6 for the flowcharts of these two methods.

The T dataset (see Section 1.5 for more details) was used to train and numerically evaluate these two methods (the M dataset does not have T2w images available). The T2w images of the subjects of the T dataset were acquired using 2D multi-slice protocols with 2.2 mm slice separations along the superior-inferior direction. The acquired in-plane sizes of all images are no larger than 192, but 19 out of 20 images (15 training images and 5 testing images) were reconstructed into 512 $\times$ 512 matrices by the scanner, resulting in in-plane digital resolutions (i.e., the voxel sizes) of 0.4 $\times$ 0.4 mm^2. Since ESPRESO only works well for scale factors that are smaller than 6 (see a discussion in Section 5.5.5), we downsampled the in-plane directions of these 19 images by half, resulting in digital resolutions of 0.8 $\times$ 0.8 $\times$ 2.2 mm^3. Note that since the digital resolutions of the T1w images of the T dataset are 1-mm isotropic, and we would register these T2w images to their corresponding T1w images as described below, this

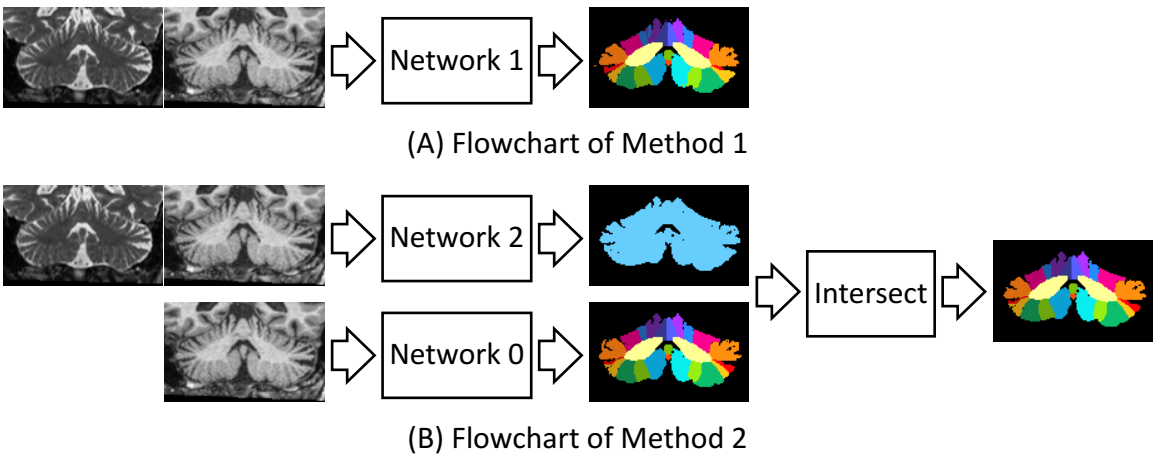


Figure 5-6. Flowcharts of the two methods to incorporate a T2w image into the parcellating network (Network) of ACAPULCO. (A) and (B) show the flowcharts of Methods 1 and 2, respectively. Networks 1 and 2 take paired T1w and T2w images as input. Network 1 in Method 1 directly outputs a parcellation. Network 2 in Method 2 outputs a mask to intersect the output of the original network (Network 0).

loss of digital resolution should not affect the parcellation results. The other image has a digital resolution of $0.828 \times 0.828 \times 2.2$ mm^3.

In addition to the pre-processing steps described in Section 2.2.1, we also normalized the mean intensities of the WM to 1,000 for both the T1w and the T2w images. We note that although some of these images do not have the same number of frequency and phase encoding steps, they do not differ much, so we used ESPRESO as if these two directions have the same resolution (see a discussion of different in-plane resolutions in Section 5.5.5).

5.4 Experiments and Results

5.4.1 Accuracy of Slice Profile Estimation

We used MRI images with simulated low through-plane resolution to test the accuracy of ESPRESO. We randomly selected 30 T1w and 60 T2w 1-mm isotropic brain images from the OASIS-3 dataset [22]. For each image, we used N4 [27] to correct their intensity inhomogeneity and normalized the mean intensities of their white matter to 1,000 [30]. To simulate LR, we first blurred each image along its superior-inferior direction with a PSF as the relative slice profile and then downsampled the image with a pre-defined scale factor. We used both Gaussian and rect PSFs in these simulations. We chose downsampling scales (i.e., the SR scale factors) of 2.0, 3.5, and 4.9 mm to cover the range of common slice separations in MRI images. To simulate slice gaps when using the Gaussian PSFs, we set their FWHMs to be 50%, 75%, and 100% of the slice separations (which are equal to the downsampling factors), representing large slice gaps, mild slice gaps, and no slice gaps, respectively. Although not as common in practice, we also simulated a slice overlap case for each scale factor by using an FWHM that is 125% of the slice separation. The same cases were used for the FWHMs of the rect PSFs except that each FWHM—i.e., the number of non-zeros values in the rect function—was rounded to its nearest odd integer which is easier to evaluate when the length of p is an odd integer (i.e, 21).

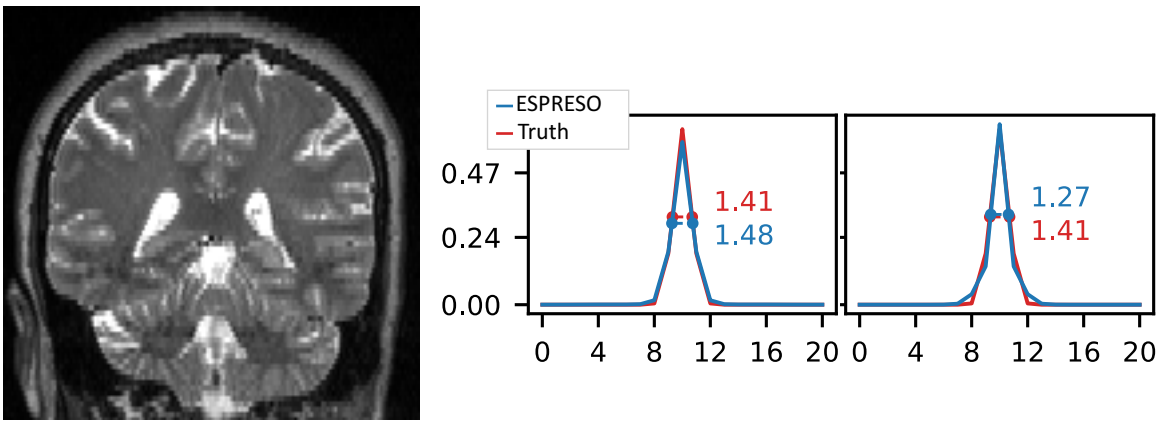
The above settings resulted in 19 simulations for each image. Simulations of 30 T2w images were used to tune the parameters of ESPRESO. Simulations of the other 30 T1w and 30 T2w images, yielding 1,140 simulations in total, were used to evaluate ESPRESO. Two evaluation metrics were used. The FWHM absolute error (FAE) calculates the absolute error between the FWHMs of the true and ESPRESO-estimated slice profiles.

Note that to calculate the FWHM of a slice profile, we first find the adjacent points around the two half maxima from the array of the slice profile values, use linear interpolation to calculate both coordinates, then use their distance as the FWHM of this slice profile. We note that this calculated FWHM from a Gaussian PSF (e.g., the red PSFs in Figs. 5-7(B) and (C), Figs. 5-8(B) and (C), and Figs. 5-9(B) and (C)) can be different from the function parameter to generate this PSF. The sum of absolute errors (SAE) calculates the sum of absolute errors between each element of the true and ESPRESO-estimated slice profiles. We compare the current version (v0.3.0) and the previously published version (v0.1.0)[1] [75] in Table 5-I. Example relative slice profiles estimated with v0.3.0 and v0.1.0 are shown in Figs. 5-7–5-12. As shown in Table 5-I, these two versions have similar performances for the Gaussian PSFs, but v0.3.0 has better performance in all cases of rect PSFs. We also note that v0.3.0 takes about 1 minute for each image to train from scratch on a GeForce RTX 2080 Ti GPU (NVIDIA Corporation, USA) while v0.1.0 takes about 20 minutes on the same hardware.

[1]We note that the version that we describe in this dissertation is v0.3.0. The main differences from v0.1.0 are: 1) we use a one-cycle learning rate policy to accelerate the training; 2) our training scheme has been improved; 3) our loss function has been simplified; 4) we tuned the hyper-parameters via extensive evaluations.

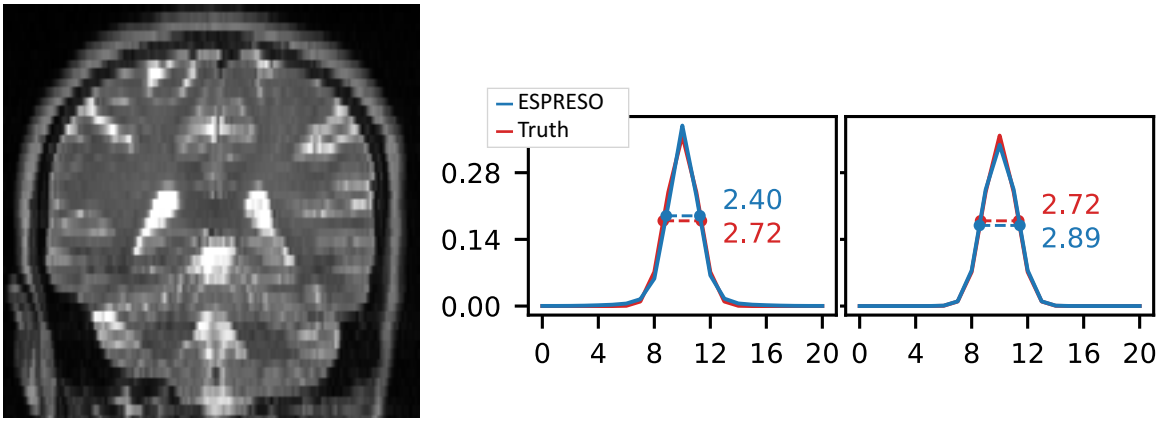
Table 5-I. Accuracy of the estimated relative slice profiles. The unit for the scale factors and FWHMs is mm. The numbers shown are means ± standard deviations. Better means between versions v0.1.0 and v0.3.0 are highlighted in blue. FAE: the absolute error of FWHMs between an estimated and the true relative slice profiles. SAE: the sum of absolute errors between an estimated and the true relative slice profiles.

Gaussian relative slice profiles					
Scale factor	**FWHM**	**FAE**		**SAE**	
		v0.1.0	**v0.3.0**	**v0.1.0**	**v0.3.0**
2.0	1.000	0.3013 ± 0.0438	0.0777 ± 0.0187	0.5095 ± 0.0473	0.2725 ± 0.0381
	1.500	0.1325 ± 0.0538	0.0809 ± 0.0492	0.1363 ± 0.0402	0.1510 ± 0.0217
	2.000	0.1347 ± 0.0675	0.1587 ± 0.1021	0.0909 ± 0.0172	0.0954 ± 0.0251
	2.500	0.2374 ± 0.0923	0.0685 ± 0.0560	0.1111 ± 0.0137	0.0438 ± 0.0103
3.5	1.750	0.2188 ± 0.1030	0.6214 ± 0.1056	0.1502 ± 0.0405	0.2399 ± 0.0291
	2.625	0.3778 ± 0.1101	0.0912 ± 0.0666	0.1208 ± 0.0165	0.0297 ± 0.0142
	3.500	0.4794 ± 0.0950	0.2371 ± 0.1170	0.1169 ± 0.0168	0.1029 ± 0.0083
	4.375	0.3830 ± 0.1530	0.5835 ± 0.0681	0.0832 ± 0.0180	0.1587 ± 0.0124
4.9	2.450	0.2332 ± 0.1286	0.3042 ± 0.1585	0.1092 ± 0.0357	0.0827 ± 0.0415
	3.675	0.6800 ± 0.2333	0.3041 ± 0.2158	0.1609 ± 0.0465	0.1656 ± 0.0544
	4.900	0.7651 ± 0.2887	0.3069 ± 0.1982	0.1183 ± 0.0359	0.2053 ± 0.0260
	6.125	0.3631 ± 0.2816	0.8402 ± 0.1963	0.0860 ± 0.0244	0.2265 ± 0.0042
Rect relative slice profiles					
Scale factor	**FWHM**	**FAE**		**SAE**	
		v0.1.0	**v0.3.0**	**v0.1.0**	**v0.3.0**
2.0	1.000	0.3335 ± 0.0409	0.1151 ± 0.0150	0.6946 ± 0.0472	0.4292 ± 0.0355
	3.000	1.0044 ± 0.1017	0.7955 ± 0.1050	0.4486 ± 0.0071	0.4054 ± 0.0158
3.5	3.000	1.0501 ± 0.0981	0.6025 ± 0.1005	0.4534 ± 0.0074	0.3807 ± 0.0088
	5.000	1.7088 ± 0.1181	0.8192 ± 0.1412	0.4074 ± 0.0146	0.2851 ± 0.0192
4.9	3.000	0.8285 ± 0.1538	0.2604 ± 0.1410	0.4389 ± 0.0044	0.3626 ± 0.0101
	5.000	2.2073 ± 0.2447	1.6549 ± 0.3063	0.4877 ± 0.0401	0.4099 ± 0.0521
	7.000	1.7609 ± 0.3676	1.1687 ± 0.3662	0.3720 ± 0.0283	0.2743 ± 0.0505



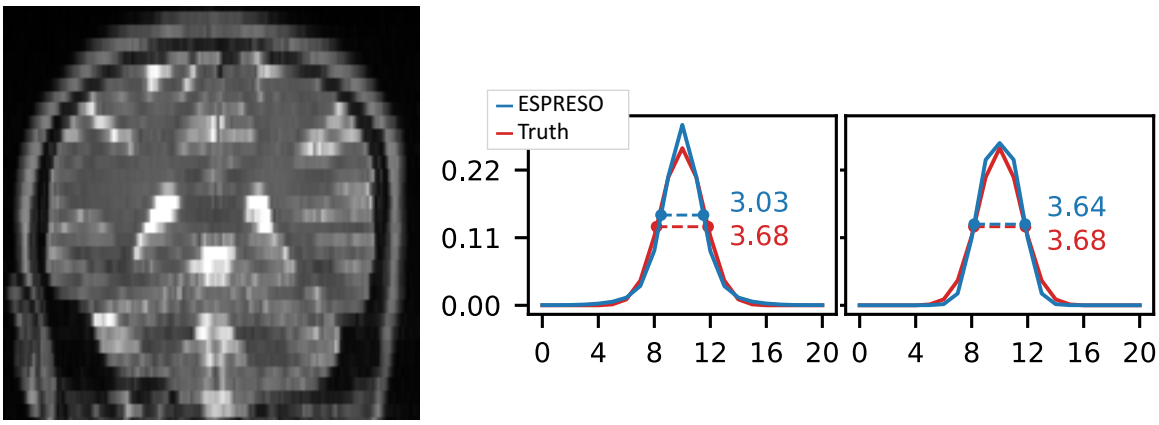
(A) Input LR image (B) v0.1.0 (C) v0.3.0

Figure 5-7. Example estimated relative slice profiles from ESPRESO v0.1.0 and v0.3.0 of a low-resolution (LR) image that is simulated using a *Gaussian PSF with a scale factor of 2.0 and an FWHM of 1.500*. (A) shows a coronal slice of the LR image (it is shown with nearest-neighbor interpolation for display purposes). (B) and (C) show the estimated relative slice profiles from v0.1.0 and v0.3.0 in blue, respectively, and the true relative slice profile is shown in red. Their FWHMs are shown in the text in their corresponding colors.



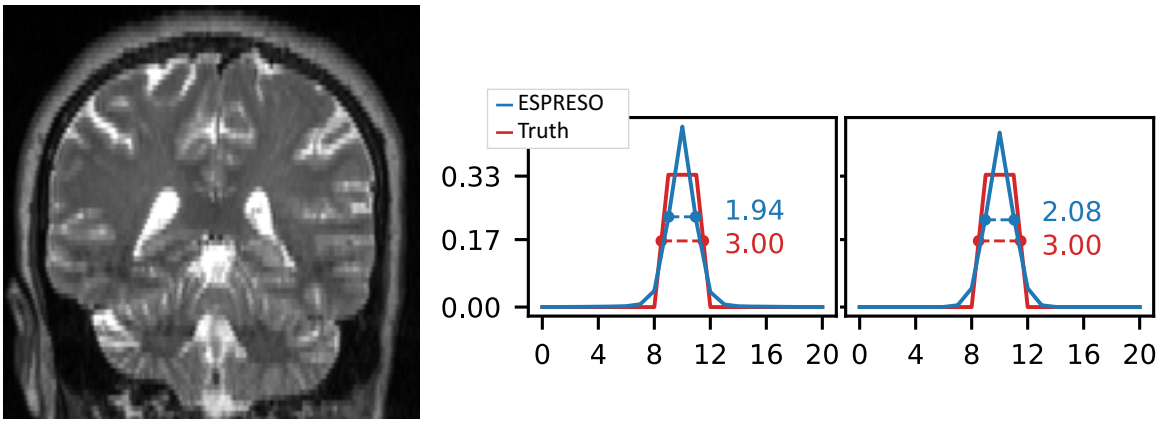
(A) Input LR image (B) v0.1.0 (C) v0.3.0

Figure 5-8. Example estimated relative slice profiles from ESPRESO v0.1.0 and v0.3.0 of a low-resolution (LR) image that is simulated using a *Gaussian PSF with a scale factor of 3.5 and an FWHM of 2.625*. (A) shows a coronal slice of the LR image (it is shown with nearest-neighbor interpolation for display purposes). (B) and (C) show the estimated relative slice profiles from v0.1.0 and v0.3.0 in blue, respectively, and the true relative slice profile is shown in red. Their FWHMs are shown in the text in their corresponding colors.



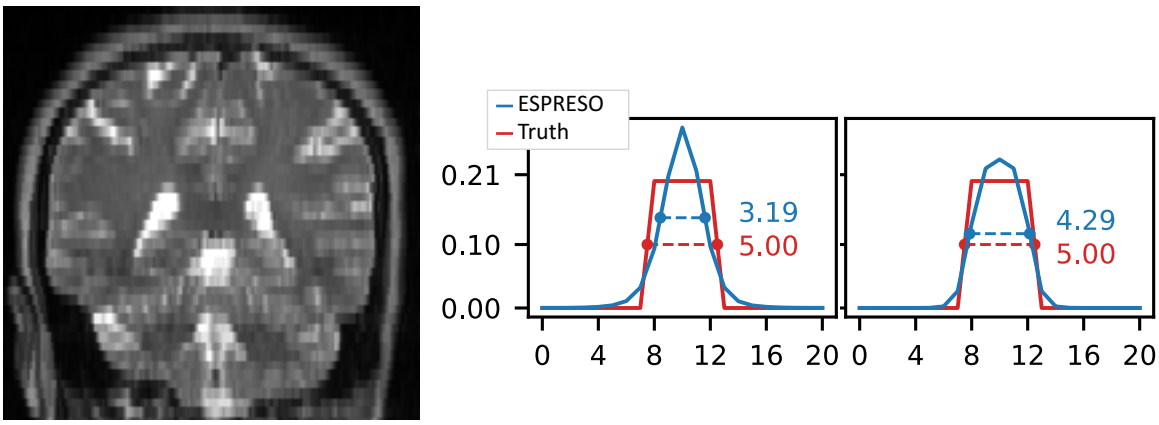
(A) Input LR image (B) v0.1.0 (C) v0.3.0

Figure 5-9. Example estimated relative slice profiles from ESPRESO v0.1.0 and v0.3.0 of a low-resolution (LR) image that is simulated using a *Gaussian PSF with a scale factor of 4.9 and an FWHM of 3.675*. (A) shows a coronal slice of the LR image (it is shown with nearest-neighbor interpolation for display purposes). (B) and (C) show the estimated relative slice profiles from v0.1.0 and v0.3.0 in blue, respectively, and the true relative slice profile is shown in red. Their FWHMs are shown in the text in their corresponding colors.



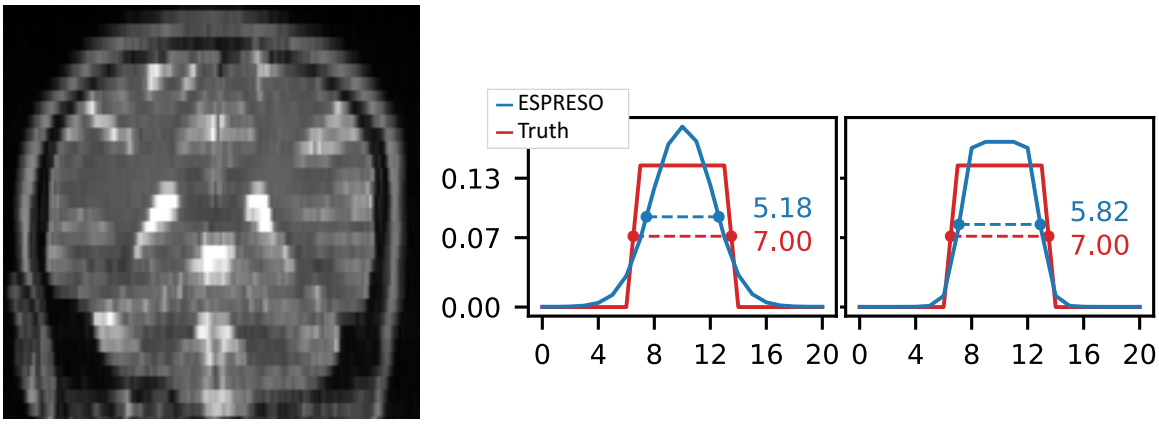
(A) Input LR image (B) v0.1.0 (C) v0.3.0

Figure 5-10. Example estimated relative slice profiles from ESPRESO v0.1.0 and v0.3.0 of a low-resolution (LR) image that is simulated using a *rect PSF with a scale factor of 2.0 and an FWHM of 3.000*. (A) shows a coronal slice of the LR image (it is shown with nearest-neighbor interpolation for display purposes). (B) and (C) show the estimated relative slice profiles from v0.1.0 and v0.3.0 in blue, respectively, and the true relative slice profile is shown in red. Their FWHMs are shown in the text in their corresponding colors.



(A) Input LR image (B) v0.1.0 (C) v0.3.0

Figure 5-11. Example estimated relative slice profiles from ESPRESO v0.1.0 and v0.3.0 of a low-resolution (LR) image that is simulated using a *rect PSF with a scale factor of 3.5 and an FWHM of 5.000*. (A) shows a coronal slice of the LR image (it is shown with nearest-neighbor interpolation for display purposes). (B) and (C) show the estimated relative slice profiles from v0.1.0 and v0.3.0 in blue, respectively, and the true relative slice profile is shown in red. Their FWHMs are shown in the text in their corresponding colors.



(A) Input LR image (B) v0.1.0 (C) v0.3.0

Figure 5-12. Example estimated relative slice profiles from ESPRESO v0.1.0 and v0.3.0 of a low-resolution (LR) image that is simulated using a *rect PSF with a scale factor of 4.9 and an FWHM of 7.000*. (A) shows a coronal slice of the LR image (it is shown with nearest-neighbor interpolation for display purposes). (B) and (C) show the estimated relative slice profiles from v0.1.0 and v0.3.0 in blue, respectively, and the true relative slice profile is shown in red. Their FWHMs are shown in the text in their corresponding colors.

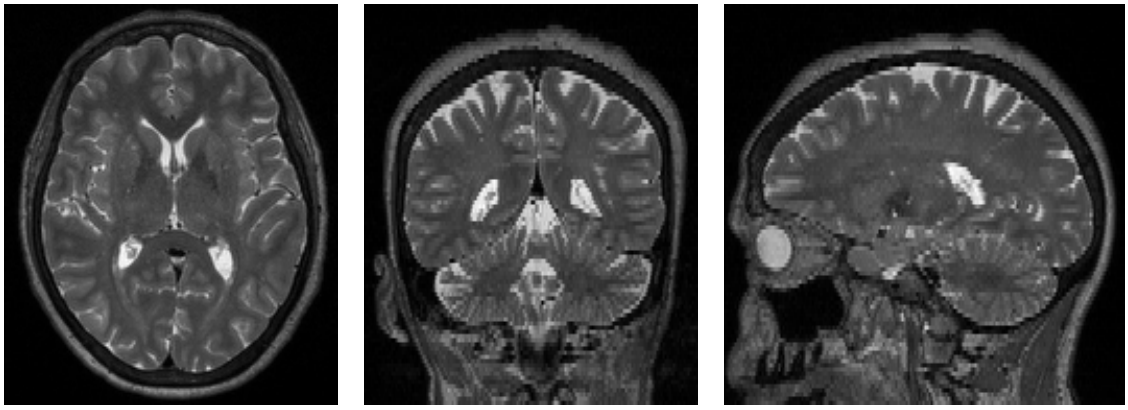
5.4.2 Compare SMORE and S-SMORE

In this section, we compare SMORE (iSMORE is the same as SMORE if there is only one training phase; see Section 5.3.4 for more details of iSMORE and SMORE) with S-SMORE after the first training phase (S-SMORE 1st). Here we use an improved implementation of SMORE over that in Zhao *et al.* [76] with the interpolation method described in Section 5.3.5. To evaluate both algorithms, we used five T2w and five T1w images with simulated low through-plane resolution. These simulations were randomly picked from those that were used to evaluate ESPRESO in Section 5.4.1 (see Fig. 5-13 for some examples). These simulations used downsampling factors (i.e., the SR scale factors) of 2.0, 3.5, and 4.9 mm and used Gaussian relative slice profiles with FWHMs of 50%, 75%, 100%, and 125% of these downsampling factors. These true relative slices profiles in the simulations were used to create training data for SMORE and S-SMORE. Both methods were trained from scratch for each image. We calculated the peak signal-to-noise ratio (PSNR) and structural similarity index (SSIM) of the SR results against the true HR images to compare these two methods (see Table 5-II). S-SMORE 1st is better than SMORE for all types of simulations in terms of PSNR and for eight out of twelve types of simulations in term of SSIM. Example SR results are shown in Figs. 5-14–5-16.

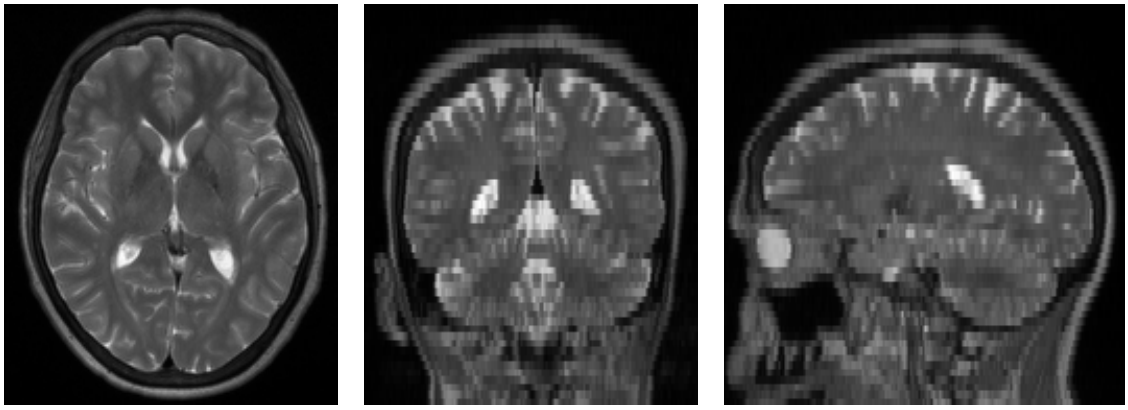
We note that we used S-SMORE with four training phases (S-SMORE 4th) in all following experiments. To show the improvement of S-SMORE 4th over S-SMORE 1st, we additionally include the results of S-SMORE 4th in Table 5-II. S-SMORE 4th is better than S-SMORE 1st in all types of simulations.

Table 5-II. PSNR (dB) and SSIM of SMORE (which is the same as iSMORE after the 1st training phase), S-SMORE 1st (S-SMORE after the 1st training phase), and S-SMORE 4th (S-SMORE after the 4th training phase). The numbers shown are means ± standard deviations. The unit of the scale factors (SF) and FWHMs is mm. Better numbers between SMORE and S-SMORE 1st are highlighted in blue. S-SMORE 4th is better than the other two in all measurement means.

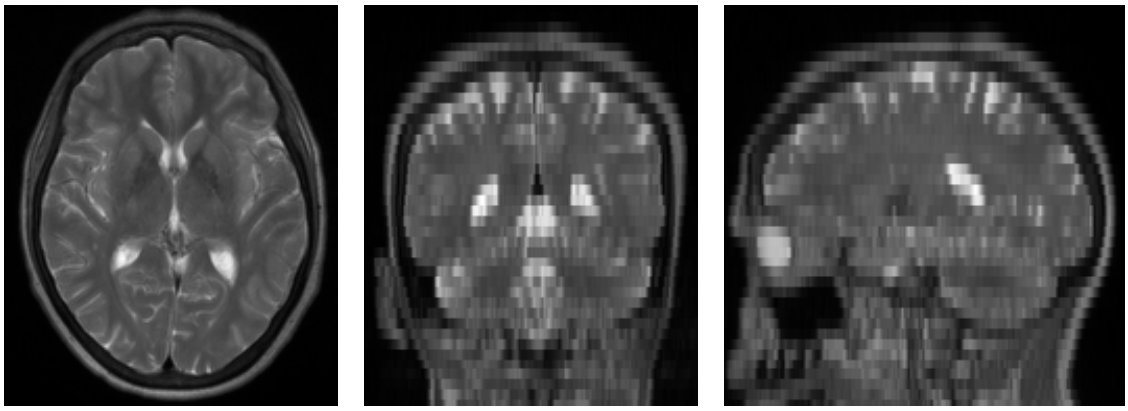
SF	FWHM	PSNR			SSIM		
		SMORE	S-SMORE 1st	S-SMORE 4th	SMORE	S-SMORE 1st	S-SMORE 4th
2.0	1.000	35.21±1.02	35.94±0.99	35.96±0.98	0.9650±0.0044	0.9722±0.0041	0.9723±0.0041
	1.500	36.07±0.92	36.26±0.94	36.31±0.94	0.9732±0.0036	0.9737±0.0037	0.9739±0.0037
	2.000	36.29±0.96	36.39±0.93	36.46±0.94	0.9743±0.0036	0.9742±0.0035	0.9745±0.0036
	2.500	36.33±0.93	36.44±0.92	36.53±0.94	0.9745±0.0035	0.9744±0.0035	0.9748±0.0035
3.5	1.750	31.90±0.99	32.57±0.99	32.59±1.00	0.9363±0.0073	0.9442±0.0067	0.9444±0.0068
	2.625	32.45±0.99	32.77±0.98	32.87±0.97	0.9437±0.0068	0.9462±0.0063	0.9469±0.0063
	3.500	32.58±1.01	32.82±0.97	32.96±0.98	0.9455±0.0059	0.9465±0.0060	0.9477±0.0062
	4.375	32.67±1.01	32.81±0.96	32.98±0.96	0.9467±0.0058	0.9463±0.0059	0.9477±0.0061
4.9	2.450	30.03±1.01	30.89±0.99	30.96±0.99	0.9045±0.0094	0.9208±0.0080	0.9215±0.0079
	3.675	30.56±0.92	31.02±1.00	31.12±1.00	0.9159±0.0081	0.9222±0.0078	0.9232±0.0077
	4.900	30.83±0.96	30.98±0.93	31.15±0.96	0.9207±0.0080	0.9213±0.0074	0.9234±0.0073
	6.125	30.98±0.99	31.02±0.95	31.18±0.96	0.9231±0.0077	0.9214±0.0072	0.9233±0.0073



(A) Downsampling factor 2.0, FWHM 1.000

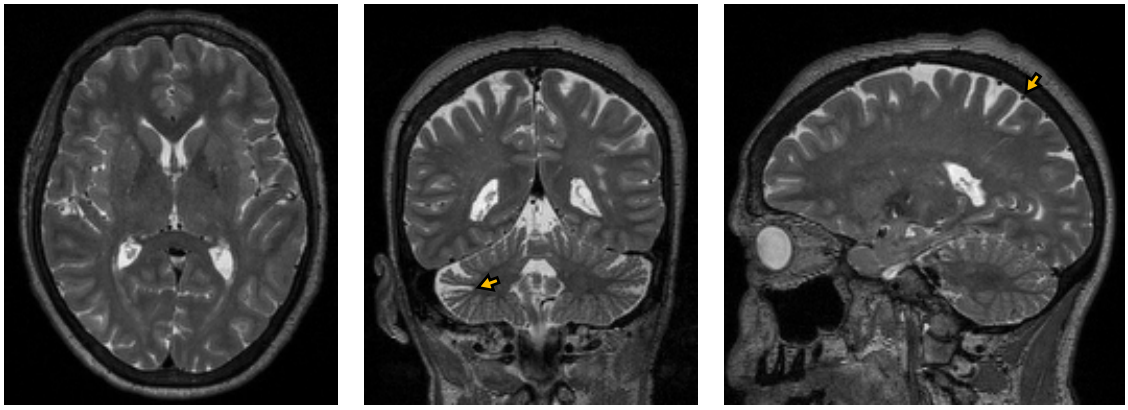


(B) Downsampling factor 3.5, FWHM 3.500

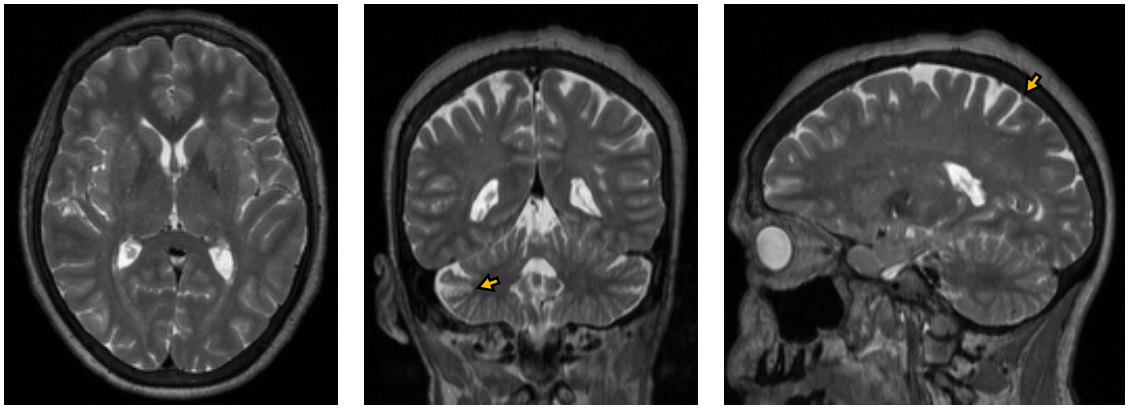


(C) Downsampling factor 4.9, FWHM 6.125

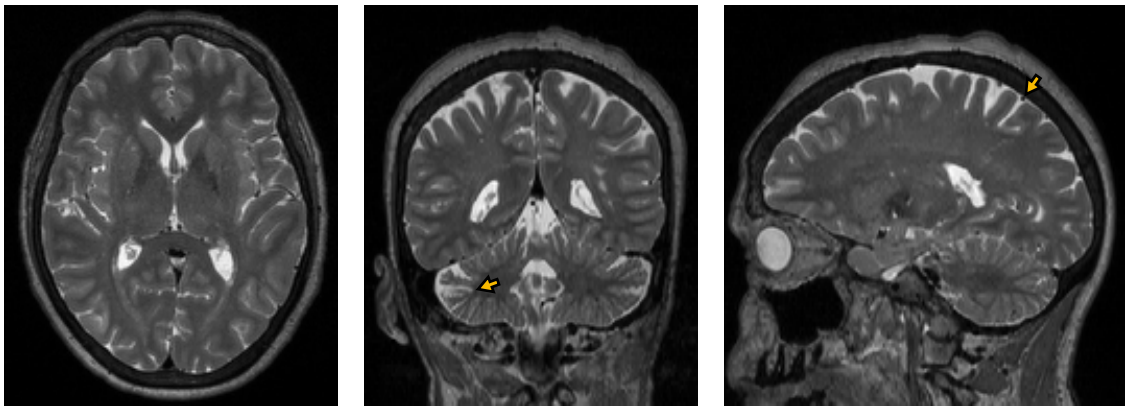
Figure 5-13. Example simulations to compare SMORE and S-SMORE. An axial, a coronal, and a sagittal slices of each simulation are shown in this figure. The low resolutions are simulated along the superior-inferior direction (the vertical direction in the coronal and sagittal slices). These simulations use Gaussian slice profiles with (A) a downsampling factor of 2.0 and an FWHM of 1.000, (B) a downsampling factor of 3.5 and an FWHM of 3.500, and (C) a downsampling factor of 4.9 and an FWHM of 6.125.



(A) True HR image

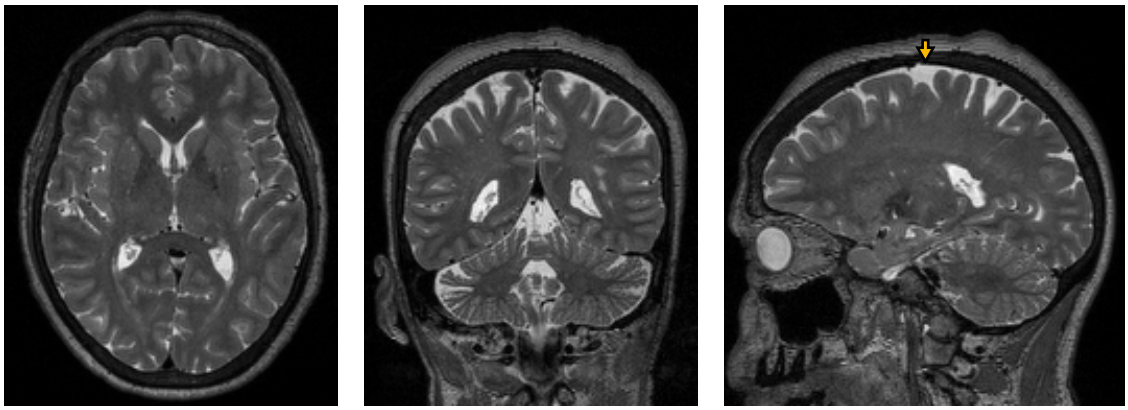


(B) SMORE result

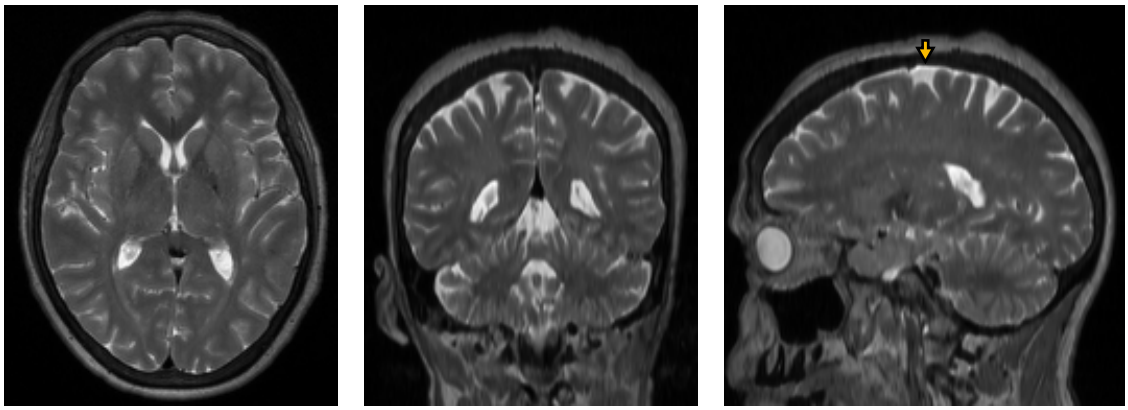


(C) S-SMORE result

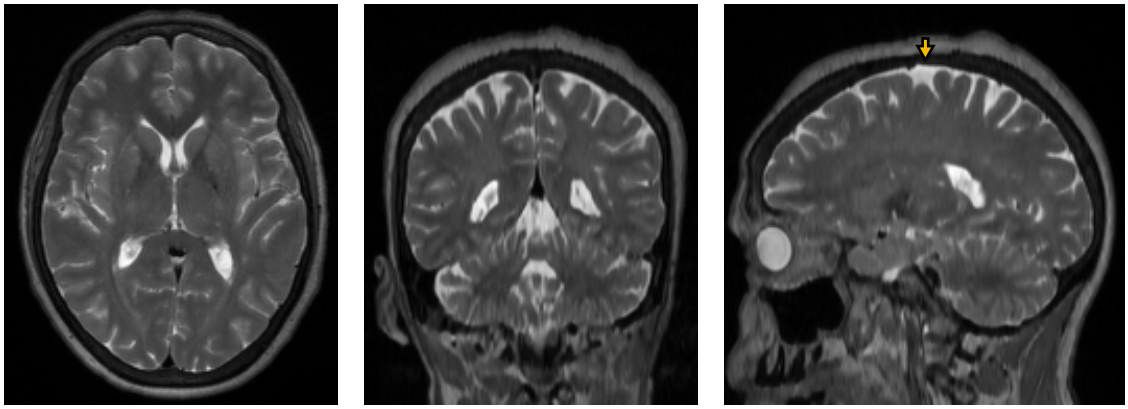
Figure 5-14. SMORE and S-SMORE results of the simulation with *a Gaussian slice profile with a downsampling factor of 2.0 and an FWHM of 1.000*. An axial, a coronal, and a sagittal slices of each simulation are shown. (A), (B), and (C) show the true high resolution image, the SMORE result, and the S-SMORE result, respectively. Note that the low resolution is simulated along the superior-inferior direction. Yellow arrows point to some differences. See Fig. 5-13(A) for the input image.



(A) True HR image

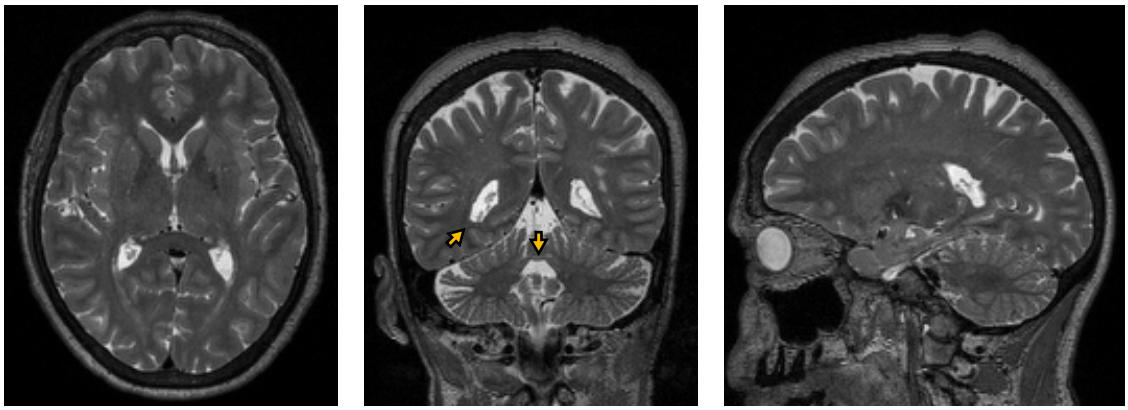


(B) SMORE result

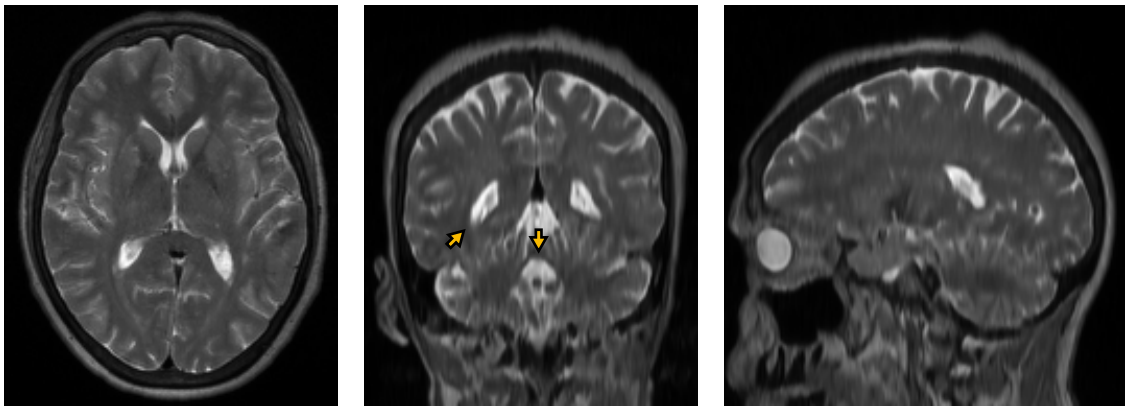


(C) S-SMORE result

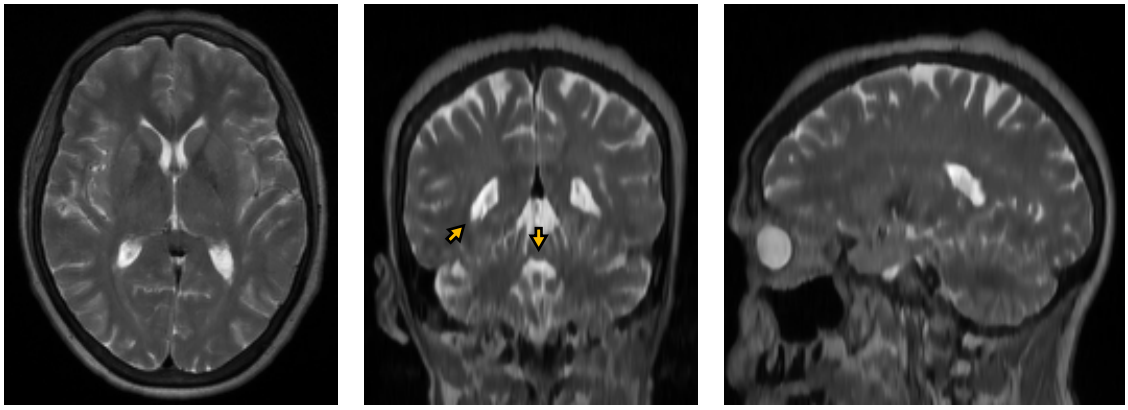
Figure 5-15. SMORE and S-SMORE results of the simulation with *a Gaussian slice profile with a downsampling factor of 3.5 and an FWHM of 3.500*. An axial, a coronal, and a sagittal slices of each simulation are shown. (A), (B), and (C) show the true high resolution image, the SMORE result, and the S-SMORE result, respectively. Note that the low resolution is simulated along the superior-inferior direction. Yellow arrows point to a difference. See Fig. 5-13(B) for the input image.



(A) True HR image



(B) SMORE result



(C) S-SMORE result

Figure 5-16. SMORE and S-SMORE results of the simulation with *a Gaussian slice profile with a downsampling factor of 4.9 and an FWHM of 6.125*. An axial, a coronal, and a sagittal slices of each simulation are shown. (A), (B), and (C) show the true high resolution image, the SMORE result, and the S-SMORE result, respectively. Note that the low resolution is simulated along the superior-inferior direction Yellow arrows point to a difference. See Fig. 5-13(C) for the input image.

5.4.3 Compare SR with and without ESPRESO

In this experiment, we show that incorporating ESPRESO into S-SMORE (see Section 5.3.4 for the details of S-SMORE) can improve its performance. As in SMORE [76], we first used a Gaussian PSF whose FWHM is equal to the slice separation as the relative slice profile to create training data for S-SMORE. We call this slice profile the "conventional slice profile" or "conventionally assumed slice profile" in the following. We then used the ESPRESO-estimated slice profile in S-SMORE for comparison.

The simulated images from Section 5.4.1 were used to evaluate S-SMORE. PSNR and SSIM were calculated between the SR results and their true HR images. As a reference, we also performed S-SMORE with the true relative slice profiles (which are generally unknown in practice). Numerical comparisons are shown in Tables 5-III and 5-IV, and example images are shown in Figs. 5-17–5-19. As shown in Tables 5-III and 5-IV, we note that in the three sets of simulations that were created with Gaussian slice profiles whose FWHMs are equal to the scale factors, the conventionally assumed slice profiles are exactly the same as the truth and are thus expected to have better SMORE results than ESPRESO. Otherwise, SMORE with ESPRESO is better for almost all other cases. We also note that using ESPRESO is worse than the conventional slice profiles in the case of the rect slice profile with a scale factor of 4.9 and an FWHM of 7.000. We speculate that in this case, neither slice profile is similar to the truth, and there might be other factors dominating the SR performance (such as the downsampling factor). As shown in Figs. 5-17 and 5-18, S-SMORE with conventional slice profiles can have "oversharpening" artifacts when there are slice gaps. The results of these two slice profiles are visually similar in Fig. 5-19, which is a slice overlap case.

Table 5-III. PSNR (dB) of S-SMORE results against the true high resolution images with different relative slice profiles. The unit of the scale factors and FWHMs is mm. The numbers shown are means ± standard deviations. Better numbers between using the conventional and ESPRESO slice profiles are highlighted in blue. The numbers for the true slice profiles are only for reference, as they are generally unknown in practice. We note that in simulations that were created with Gaussian slice profiles whose FWHMs are equal to the scale factors, a conventionally assumed slice profile is exactly the same with the truth and are thus expected to be better than ESPRESO.

Gaussian slice profiles				
Scale factor	**FWHM**	**Conventional**	**ESPRESO**	**True**
2.0	1.000	31.80 ± 1.16	35.40 ± 1.08	35.78 ± 1.07
	1.500	35.00 ± 1.06	36.06 ± 1.09	36.13 ± 1.06
	2.000	36.29 ± 1.06	36.27 ± 1.08	36.29 ± 1.06
	2.500	35.51 ± 1.05	36.36 ± 1.07	36.37 ± 1.08
3.5	1.750	29.40 ± 1.20	32.10 ± 1.06	32.38 ± 1.06
	2.625	31.75 ± 1.09	32.63 ± 1.07	32.65 ± 1.06
	3.500	32.74 ± 1.07	32.63 ± 1.07	32.74 ± 1.07
	4.375	32.08 ± 1.08	32.61 ± 1.08	32.75 ± 1.07
4.9	2.450	27.87 ± 1.23	30.55 ± 1.07	30.67 ± 1.07
	3.675	30.05 ± 1.12	30.70 ± 1.09	30.84 ± 1.08
	4.900	30.90 ± 1.08	30.65 ± 1.09	30.90 ± 1.08
	6.125	30.46 ± 1.09	30.73 ± 1.11	30.93 ± 1.08
Rect slice profile				
Scale factor	**FWHM**	**Conventional**	**ESPRESO**	**True**
2.0	1.000	30.70 ± 1.17	34.96 ± 1.09	35.64 ± 1.07
	3.000	36.14 ± 1.06	36.24 ± 1.08	36.43 ± 1.08
3.5	3.000	30.15 ± 1.15	32.43 ± 1.05	32.53 ± 1.06
	5.000	32.49 ± 1.08	32.56 ± 1.08	32.79 ± 1.07
4.9	3.000	27.07 ± 1.28	30.31 ± 1.07	30.58 ± 1.07
	5.000	29.73 ± 1.11	30.60 ± 1.10	30.84 ± 1.06
	7.000	30.65 ± 1.09	30.57 ± 1.11	30.96 ± 1.07

Table 5-IV. SSIM of S-SMORE results against the true high resolution images with different slice profiles. The unit of the scale factors and FWHMs is mm. The numbers shown are means ± standard deviations. Better numbers between using the conventional and ESPRESO slice profiles are highlighted in blue. The numbers for the true slice profiles are only for reference, as they are generally unknown in practice. We note that in simulations that were created with Gaussian slice profiles whose FWHMs are equal to the scale factors, a conventionally assumed slice profile is exactly the same as the truth and are thus expected to be better than ESPRESO.

Gaussian slice profile				
Scale factor	**FWHM**	**Conventional**	**ESPRESO**	**True**
2.0	1.000	0.9523 ± 0.0065	0.9709 ± 0.0052	0.9715 ± 0.0051
	1.500	0.9699 ± 0.0051	0.9735 ± 0.0049	0.9731 ± 0.0049
	2.000	0.9738 ± 0.0049	0.9741 ± 0.0049	0.9738 ± 0.0049
	2.500	0.9694 ± 0.0052	0.9743 ± 0.0049	0.9741 ± 0.0049
3.5	1.750	0.9244 ± 0.0098	0.9426 ± 0.0082	0.9431 ± 0.0083
	2.625	0.9428 ± 0.0082	0.9456 ± 0.0081	0.9457 ± 0.0081
	3.500	0.9464 ± 0.0082	0.9447 ± 0.0082	0.9464 ± 0.0082
	4.375	0.9378 ± 0.0088	0.9442 ± 0.0084	0.9464 ± 0.0083
4.9	2.450	0.8968 ± 0.0119	0.9185 ± 0.0106	0.9194 ± 0.0105
	3.675	0.9182 ± 0.0104	0.9177 ± 0.0109	0.9213 ± 0.0104
	4.900	0.9217 ± 0.0107	0.9160 ± 0.0113	0.9217 ± 0.0107
	6.125	0.9124 ± 0.0116	0.9174 ± 0.0116	0.9216 ± 0.0108
Rect slice profile				
Scale factor	**FWHM**	**Conventional**	**ESPRESO**	**True**
2.0	1.000	0.9431 ± 0.0075	0.9689 ± 0.0054	0.9708 ± 0.0052
	3.000	0.9729 ± 0.0050	0.9741 ± 0.0049	0.9743 ± 0.0050
3.5	3.000	0.9315 ± 0.0088	0.9448 ± 0.0081	0.9444 ± 0.0083
	5.000	0.9437 ± 0.0085	0.9442 ± 0.0085	0.9463 ± 0.0083
4.9	3.000	0.8864 ± 0.0131	0.9166 ± 0.0106	0.9180 ± 0.0105
	5.000	0.9150 ± 0.0103	0.9165 ± 0.0111	0.9208 ± 0.0106
	7.000	0.9173 ± 0.0112	0.9151 ± 0.0117	0.9215 ± 0.0108

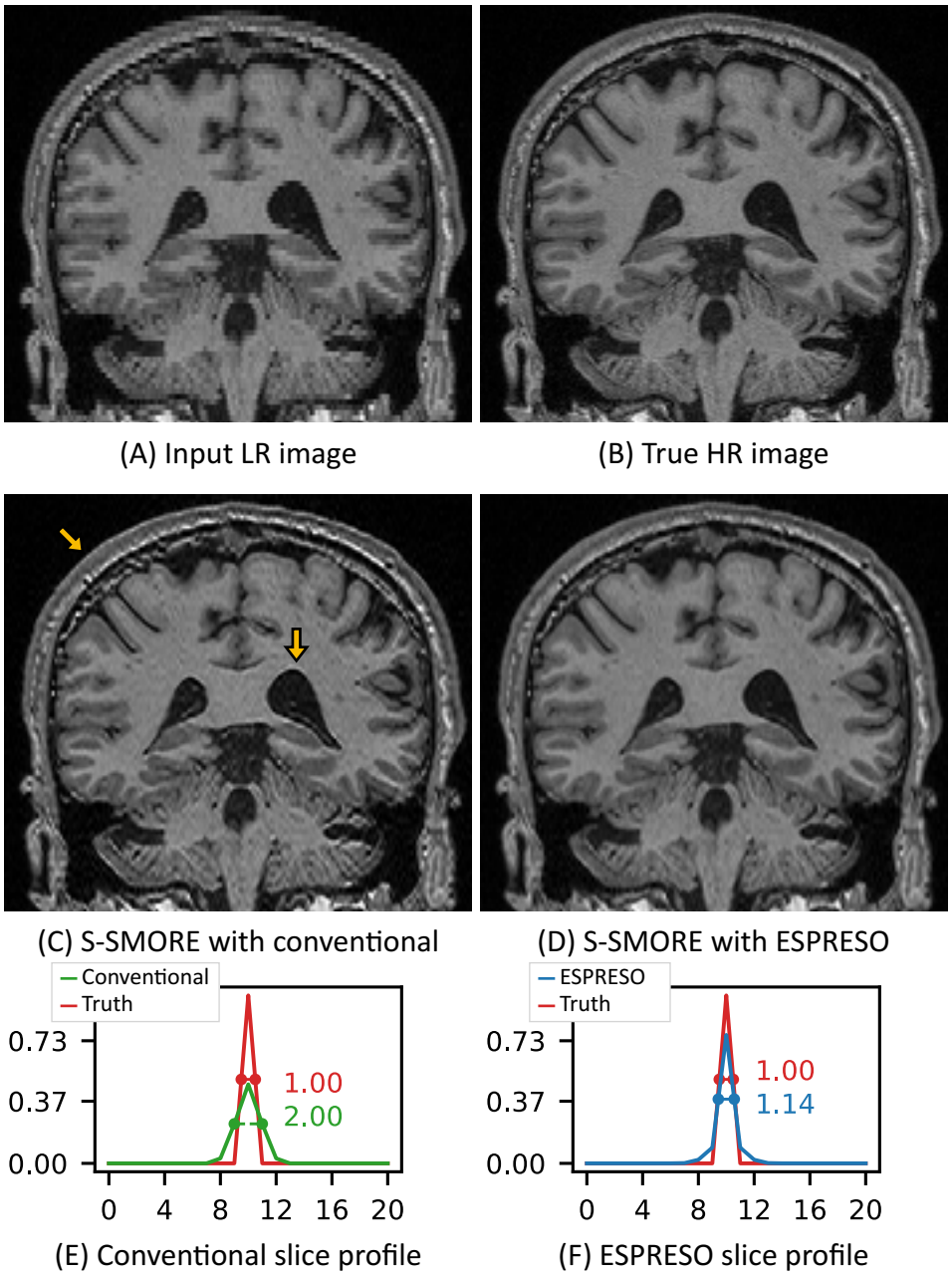


Figure 5-17. Example S-SMORE results using conventional and ESPRESO slice profiles. The low-resolution (LR) of the input image is simulated using a *Gaussian PSF with a scale factor of 2.0 and an FWHM of 1.000*. (A) shows a coronal slice of the input image (with nearest-neighbor interpolation for display purposes). (B) shows the true high-resolution (HR) image. (C) and (D) show the S-SMORE results with the conventional and ESPRESO slice profiles, respectively. The conventional (green) and ESPRESO (blue) slice profiles are plotted with the truth (red) in (E) and (F), respectively. Their FWHMs are shown in the text in their corresponding colors. Yellow arrows point to some artifacts of using the conventional slice profile.

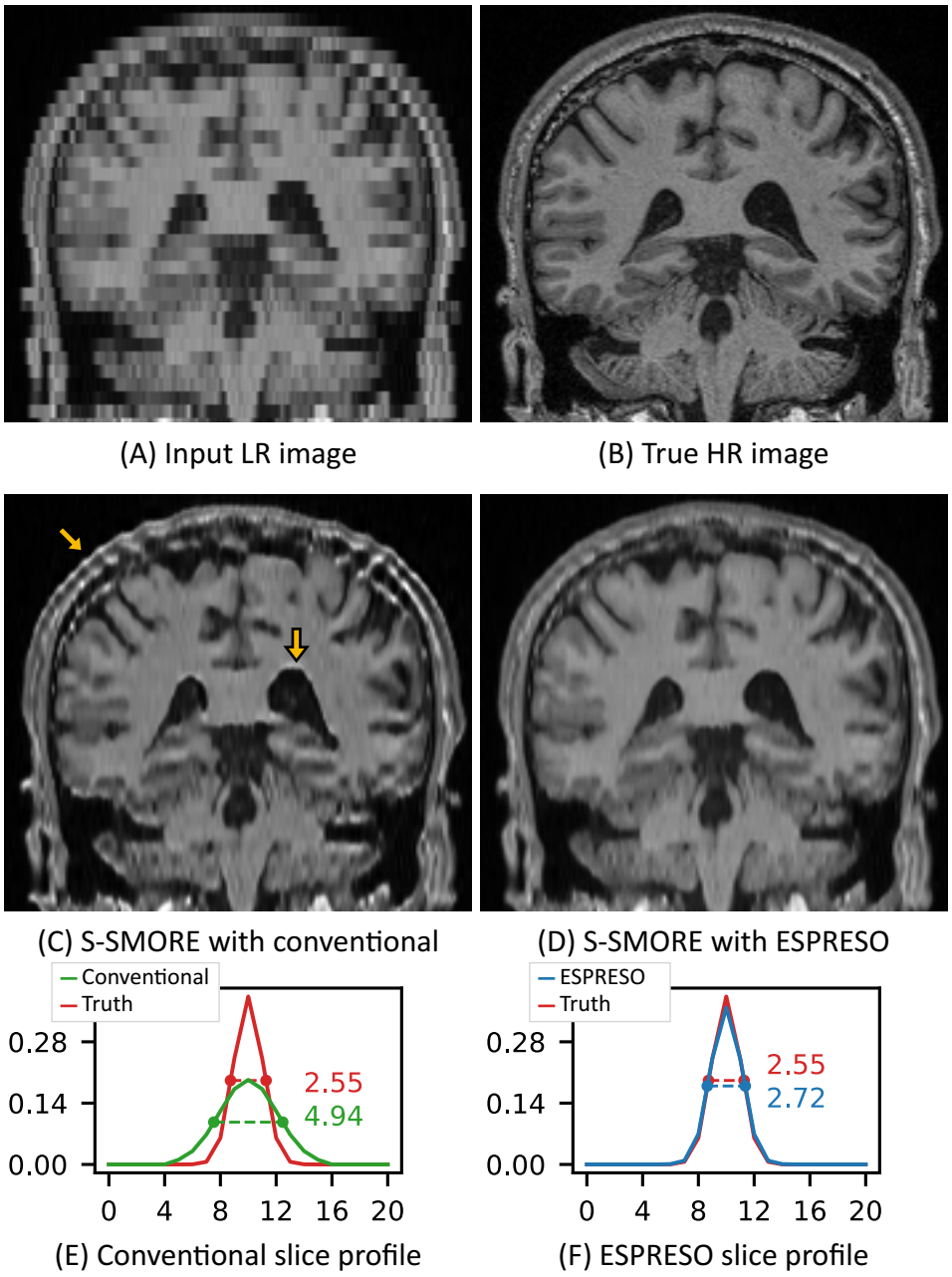


Figure 5-18. Example S-SMORE results using conventional and ESPRESO slice profiles. The low-resolution (LR) of the input image is simulated using a *Gaussian PSF with a scale factor of 4.9 and an FWHM of 2.450*. (A) shows a coronal slice of the input image (with nearest-neighbor interpolation for display purposes). (B) shows the true high-resolution (HR) image. (C) and (D) show the S-SMORE results with the conventional and ESPRESO slice profiles, respectively. The conventional (green) and ESPRESO (blue) slice profiles are plotted with the truth (red) in (E) and (F), respectively. Their FWHMs are shown in the text in their corresponding colors. Yellow arrows point to some artifacts of using the conventional slice profile.

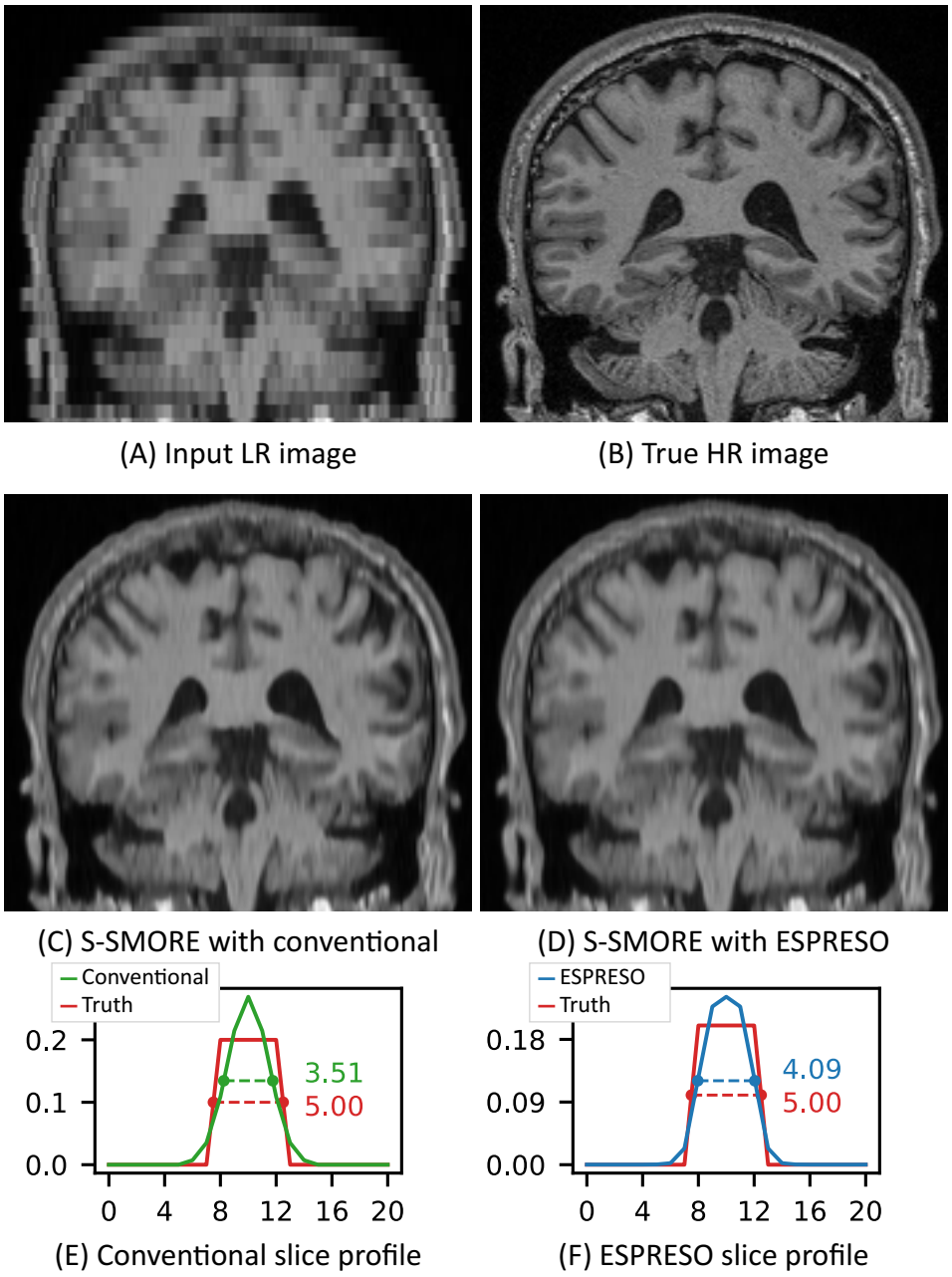


Figure 5-19. Example S-SMORE results using conventional and ESPRESO slice profiles. The low-resolution of the input image is simulated using a *rect PSF with a scale factor of 4.9 and an FWHM of 7.000*. (A) shows a coronal slice of the input image (with nearest-neighbor interpolation for display purposes). (B) shows the true HR image. (C) and (D) show the S-SMORE results with the conventional and ESPRESO slice profiles, respectively. The conventional (green) and ESPRESO (blue) slice profiles are plotted with the truth (red) in (E) and (F), respectively. Their FWHMs are shown in the text in their corresponding colors.

5.4.4 Apply ESPRESO to Real Images

In this experiment, we applied ESPRESO to 926 T2w images each with digital resolution $1 \times 1 \times 4$ mm^3 from the OASIS-3 dataset. Although they have the same digital resolution, the mean values of the estimated slice profile FWHMs, as shown in Table 5-V, are in fact clustered into two subsets, which indicates two different physical resolutions. Here we denote the images with mean estimated FWHMs of 4.6240 mm and 2.3380 mm as Subset 1 and Subset 2, respectively. We note that for 775 out of the 778 images of Subset 1, their slice thicknesses are recorded as 4 mm according to OASIS-3, but the remaining 151 images (3 from Subset 1 and 148 from Subset 2) have no value for this entry.

We super-resolved some of these images from both subsets using S-SMORE with the conventionally assumed and ESPRESO-estimated slice profiles. Example S-SMORE results are shown in Figs. 5-20 and 5-21 for a visual comparison. For Subset 1, as shown in Fig. 5-20, the S-SMORE results of both slice profiles are very similar to each other since ESPRESO estimated a similar FWHM (4.7606 mm) to the slice separation. For Subset 2, as shown in Fig. 5-21, the FWHM estimated by ESPRESO (2.3222 mm) is about half of the slice separation, and its SR result is visually better than using the conventional slice profile. In this case, S-SMORE with the conventional slice profile also produces oversharpening because the through-plane FWHM is assumed to be much greater than the truth.

Table 5-V. ESPRESO results of real images. The unit of FWHMs is mm. The estimated FWHM values from ESPRESO are shown as their mean ± standard deviation for each subset. Note that the digital resolution of both subsets is $1 \times 1 \times 4$ mm^3.

	Subset 1	**Subset 2**
Number of images	778	148
Estimated FWHM	4.6240 ± 0.2258	2.3380 ± 0.1147

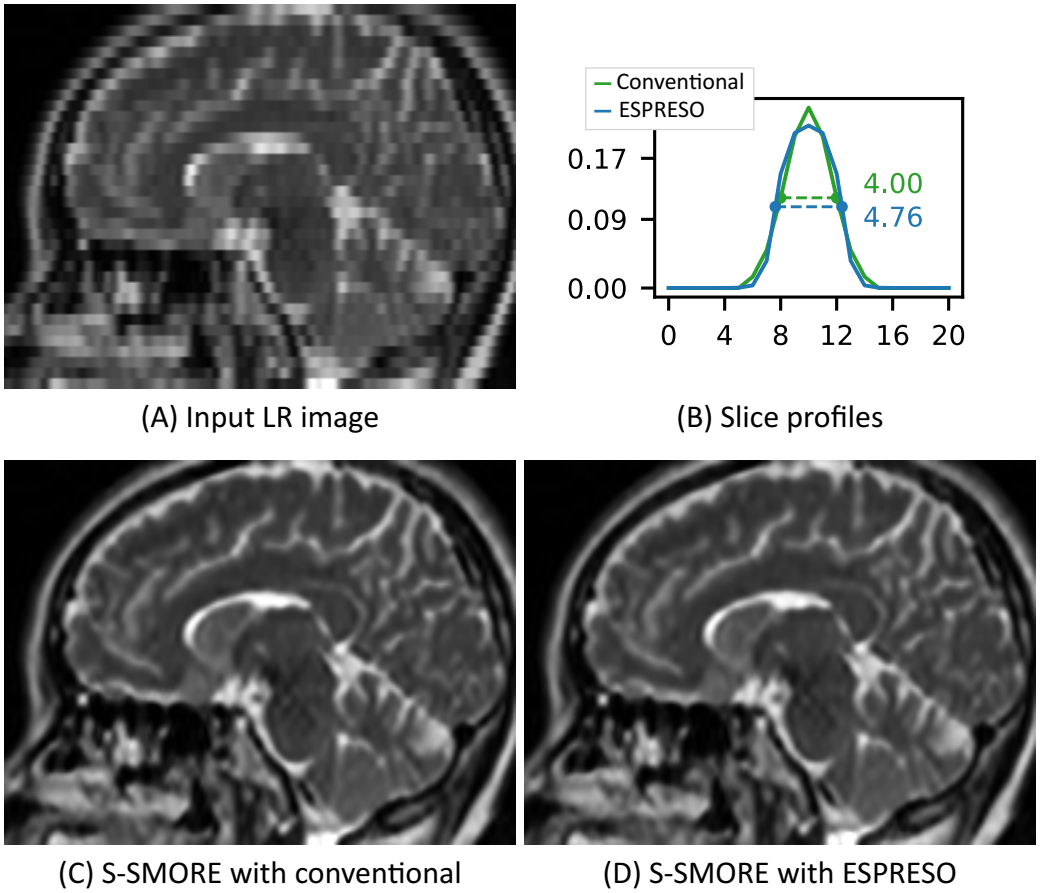
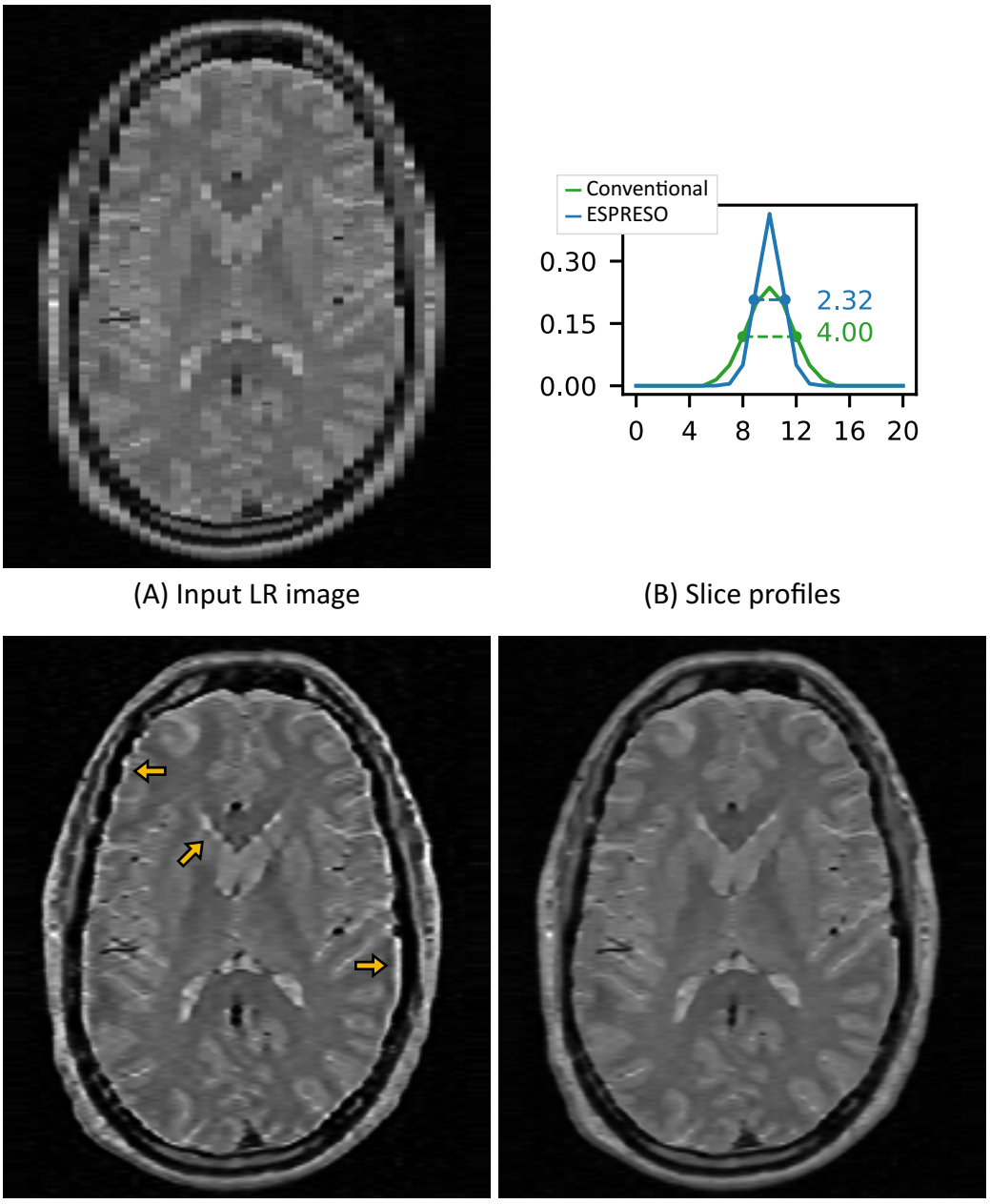


Figure 5-20. Example S-SMORE results of a real T2w image from *Subset 1*. The through-plane direction is from superior to inferior. (A) shows a sagittal slice of this image (it is shown with nearest-neighbor interpolation for display purposes). (B) shows the conventional (green) and ESPRESO (blue) slice profiles. Their FHWMs are shown in the text in their corresponding colors. (C) and (D) show the S-SMORE results with the conventional and ESPRESO slice profiles, respectively.



(A) Input LR image

(B) Slice profiles

(C) S-SMORE with conventional

(D) S-SMORE with ESPRESO

Figure 5-21. Example S-SMORE results of a real T2w image from *Subset 2*. The through-plane direction is from left to right. (A) shows an axial slice of this image (it is shown with nearest-neighbor interpolation for display purposes). (B) shows the conventional (green) and ESPRESO (blue) slice profiles. Their FHWMs are shown in the text in their corresponding colors. (C) and (D) show the S-SMORE results with the conventional and ESPRESO slice profiles, respectively. Yellow arrows point to some artifacts of using the conventional slice profiles.

5.4.5 ACAPULCO with Paired T1w and T2w Images

In this experiment, we show that using a T2w image as an additional input to ACAPULCO can improve the cerebellum parcellation. To evaluate the effect of super-resolving a T2w image in this scenario, we also trained networks that take as input LR T2w images for Methods 1 and 2; i.e., these LR T2w images were directly registered to their corresponding T1w images using cubic interpolation without super-resolving their through-plane directions. We trained all networks as in Section 2.2.6 using the 15 training images from the T dataset. We then calculated the Dice coefficients of the 5 testing images from the T dataset to compare them (see Table 5-VI). Method 2 (which outputs a cerebellum mask to intersect the parcellation from the T1w-only network) with SR T2w images has the best average mean Dice coefficient. We note that Method 1 is better when using LR T2w images instead of SR T2w image. See Section 5.5.6 for a discussion. Fig. 5-22 shows a visual comparison of these methods using a testing image. The oversegmentation into the sinuses from the original ACAPULCO can be avoided (to some extent) by using T2w images. We further applied these methods to some images of Subset 1 of the OASIS-3 dataset (see Section 5.4.4 for more details of these image). The T2w images from Subset 1 have a lower through-plane resolution (4 mm) compared to the training data from the T dataset (2.2 mm). Although Table 5-VI shows that using LR T2w images in Method 1 is better than using SR T2w images, super-resolving the image from Subset 1 produces a visually better cerebellum parcellation as shown in Fig. 5-23. This is possibly because the SR T2w image has a closer resolution to the training data compared to the LR T2w image.

Table 5-VI. Dice coefficients of cerebellum parcellations from only using T1w images and from Methods 1 and 2 using low-resolution (LR) or super-resolved (SR) T2w images in addition to T1w images. The numbers shown are means ± standard deviations (SDs). The bottom row shows the average mean values and the average SDs from all regions. The best means among these methods are shown in blue. CM: corpus medullare. Ver: vermal lobule. L: left hemispheric. R: right hemispheric.

	T1w only	Method 1		Method 2	
		LR T2w	SR T2w	LR T2w	SR T2w
CM	0.8952 ± 0.0246	0.8928 ± 0.0276	0.8876 ± 0.0341	0.8952 ± 0.0246	0.8952 ± 0.0246
Ver VI	0.8236 ± 0.0515	0.8026 ± 0.0492	0.7972 ± 0.0516	0.8236 ± 0.0515	0.8236 ± 0.0515
Ver VII	0.8070 ± 0.0474	0.7952 ± 0.0381	0.8017 ± 0.0618	0.8070 ± 0.0474	0.8070 ± 0.0474
Ver VIII	0.8996 ± 0.0202	0.8826 ± 0.0205	0.8809 ± 0.0281	0.8996 ± 0.0202	0.8996 ± 0.0202
Ver IX	0.8668 ± 0.0447	0.8551 ± 0.0418	0.8467 ± 0.0482	0.8668 ± 0.0447	0.8668 ± 0.0447
Ver X	0.8435 ± 0.0361	0.8330 ± 0.0390	0.8300 ± 0.0442	0.8435 ± 0.0361	0.8435 ± 0.0361
L I–III	0.7695 ± 0.0125	0.7826 ± 0.0214	0.7833 ± 0.0247	0.7781 ± 0.0122	0.7808 ± 0.0143
R I–III	0.6239 ± 0.1556	0.6836 ± 0.1396	0.6903 ± 0.1469	0.6289 ± 0.1555	0.6281 ± 0.1547
L IV	0.7757 ± 0.1351	0.7766 ± 0.1242	0.7879 ± 0.1427	0.7771 ± 0.1370	0.7780 ± 0.1379
R IV	0.7552 ± 0.0826	0.7607 ± 0.0911	0.7673 ± 0.0859	0.7560 ± 0.0832	0.7563 ± 0.0827
L V	0.6499 ± 0.3113	0.6408 ± 0.3005	0.6484 ± 0.2983	0.6515 ± 0.3117	0.6511 ± 0.3118
R V	0.6526 ± 0.2267	0.6497 ± 0.1884	0.6649 ± 0.2064	0.6552 ± 0.2286	0.6546 ± 0.2287
L VI	0.8374 ± 0.1090	0.8232 ± 0.1028	0.8274 ± 0.1020	0.8426 ± 0.1091	0.8438 ± 0.1093
R VI	0.8668 ± 0.0521	0.8586 ± 0.0322	0.8689 ± 0.0451	0.8734 ± 0.0564	0.8706 ± 0.0548
L Crus I	0.9383 ± 0.0114	0.9217 ± 0.0396	0.9308 ± 0.0190	0.9416 ± 0.0091	0.9424 ± 0.0084
R Crus I	0.9090 ± 0.0191	0.9172 ± 0.0109	0.9118 ± 0.0126	0.9160 ± 0.0171	0.9135 ± 0.0179
L Crus II	0.8042 ± 0.0554	0.7803 ± 0.0798	0.7888 ± 0.0647	0.8047 ± 0.0546	0.8050 ± 0.0542
R Crus II	0.8401 ± 0.0684	0.8544 ± 0.0527	0.8419 ± 0.0647	0.8394 ± 0.0701	0.8398 ± 0.0710
L VIIB	0.5619 ± 0.2897	0.6399 ± 0.2141	0.6145 ± 0.2423	0.5632 ± 0.2897	0.5642 ± 0.2907
R VIIB	0.6635 ± 0.3164	0.6810 ± 0.3082	0.6789 ± 0.3148	0.6656 ± 0.3174	0.6658 ± 0.3176
L VIIIA	0.7448 ± 0.1637	0.7808 ± 0.1332	0.7712 ± 0.1373	0.7557 ± 0.1638	0.7600 ± 0.1644
R VIIIA	0.6903 ± 0.1327	0.6905 ± 0.1435	0.6800 ± 0.1734	0.6926 ± 0.1312	0.6953 ± 0.1301
L VIIIB	0.8786 ± 0.0287	0.8835 ± 0.0356	0.8801 ± 0.0380	0.8921 ± 0.0269	0.8990 ± 0.0305
R VIIIB	0.8249 ± 0.0323	0.8080 ± 0.0491	0.7832 ± 0.0516	0.8304 ± 0.0357	0.8302 ± 0.0361
L IX	0.9053 ± 0.0312	0.9030 ± 0.0330	0.9003 ± 0.0380	0.9053 ± 0.0312	0.9053 ± 0.0312
R IX	0.9104 ± 0.0296	0.9020 ± 0.0252	0.8858 ± 0.0380	0.9104 ± 0.0296	0.9104 ± 0.0296
L X	0.7714 ± 0.0154	0.7648 ± 0.0253	0.7716 ± 0.0399	0.7714 ± 0.0154	0.7714 ± 0.0154
R X	0.8044 ± 0.0528	0.8109 ± 0.0710	0.7928 ± 0.0995	0.8042 ± 0.0528	0.8044 ± 0.0528
Average	0.7969 ± 0.0913	0.7991 ± 0.0871	0.7969 ± 0.0948	0.7997 ± 0.0915	0.8002 ± 0.0917

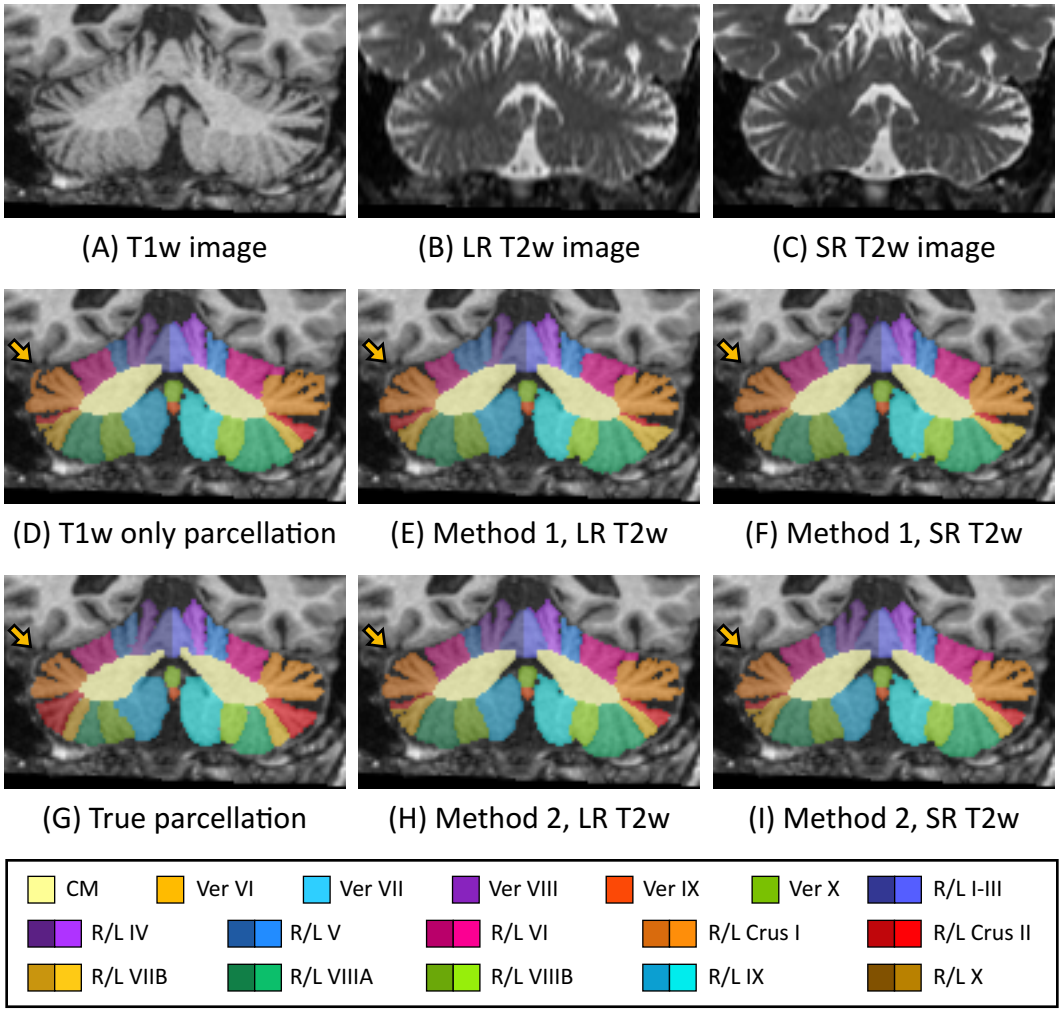


Figure 5-22. A visual comparison between cerebellum parcellations from only using a T1w image and using paired T1w and T2w images of a testing subject from the *T dataset*. (A) shows the T1w image. (B) and (C) show the cubic-interpolated low-resolution (LR) and super-resolved (SR) T2 images, respectively. (D) shows the parcellation of only using the T1w image as input. (E) and (F) show parcellations of Method 1 with LR and SR T2w images, respectively. (G) shows the true manual delineation. (H) and (I) show parcellations of Method 2 with LR and SR T2w images, respectively. Yellow arrows point to an oversegmentation that is avoided by using a T2w image. The parcellations between using LR and SR T2w images are visually similar for this subject. CM: corpus medullare. Ver: vermal lobule. L: left hemispheric. R: right hemispheric.

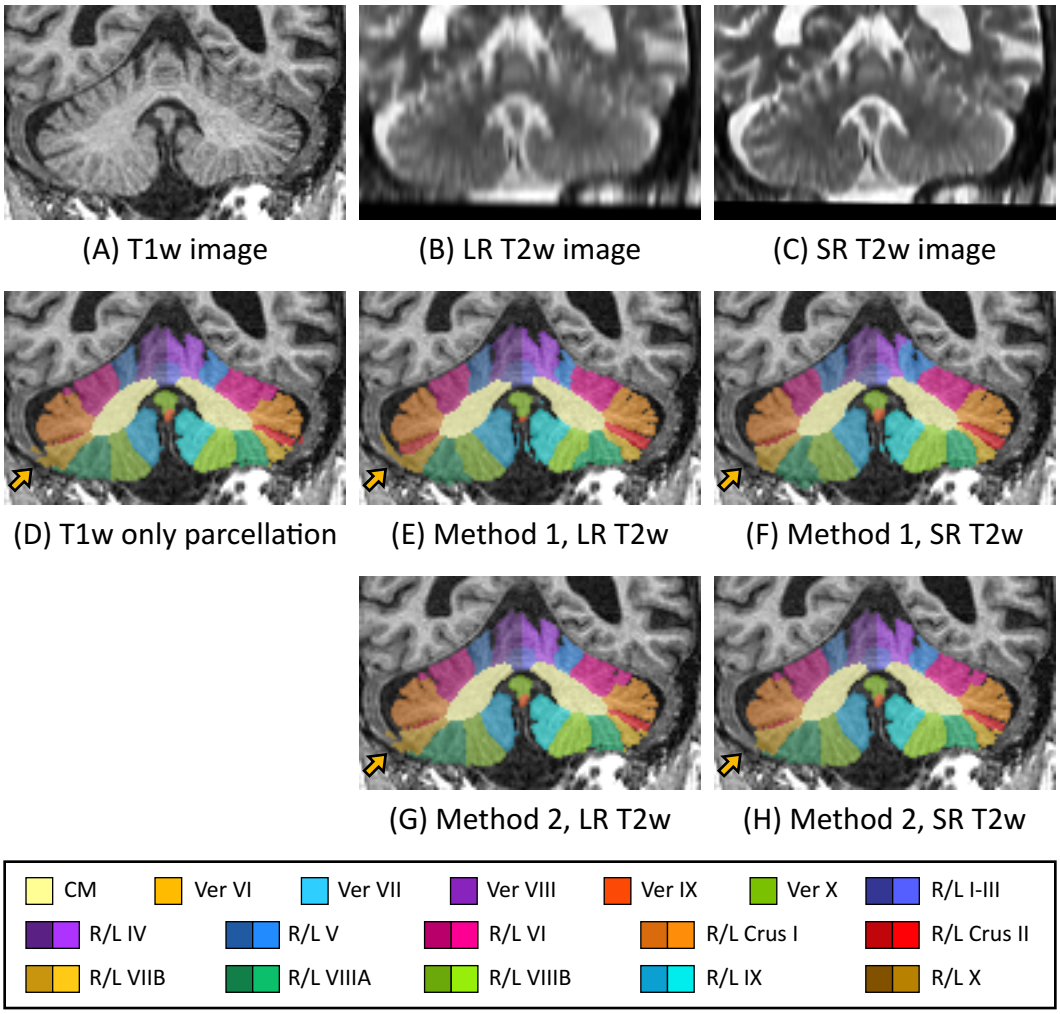


Figure 5-23. A visual comparison between cerebellum parcellations from only using a T1w image and using paired T1w and T2w images of a testing subject from the *OASIS-3 dataset*. (A) shows the T1w image. (B) and (C) show the cubic-interpolated low-resolution (LR) and super-resolved (SR) T2 images, respectively. (D) shows the parcellation of only using the T1w image as input. (E) and (F) show parcellations of Method 1 with LR and SR T2w images, respectively. (G) and (H) show parcellations of Method 2 with LR and SR T2w images, respectively. Yellow arrows point to an oversegmentation that is avoided by using the SR T2w image. Note that this oversegmentation persists when using the LR T2w image. CM: corpus medullare. Ver: vermal lobule. L: left hemispheric. R: right hemispheric.

5.5 Discussion

5.5.1 Influence of Downsampling Factor on ESPRESO

The downsampling factor of a 2D multi-slice image is determined by the ratio between the through-plane and in-plane digital resolutions (s in Eq. (5.3)). In preliminary experiments, we also tried ESPRESO in simulations that only blur the image without any downsampling, i.e., as in deblurring as opposed to SR. With a different set of hyper-parameters (the maximum learning rate is 0.002, weight decay λ_{wd} is 0.002, the number of adversarial iterations is 5,000, and the number of warm-up iterations is 160), we found that ESPRESO can recover the true relative slice profiles fairly well, as shown in Fig. 5-24. This indicates that the errors introduced by ESPRESO may result from the downsampling factor.

5.5.2 ESPRESO Evaluation of SR Performance

ESPRESO can potentially be used to measure the resulting through-plane resolution of super-resolved images to evaluate the performance of an SR algorithm. Using the hyper-parameters in Section 5.5.1, we applied ESPRESO (with the downsampling factor $s = 1$) to the results of S-SMORE trained with the true relative slice profiles from our Gaussian simulations in Section 5.4.3. We use the FWHM of the resulting "slice profile" as a indicator of the "remaining" through-plane resolution that is undone by S-SMORE; i.e, a smaller FWHM indicates a better SR. As shown in Table 5-VII, images with larger slice separations generally have worse super-resolved resolutions, which corresponds to intuition and visual inspection of the super-resolved images (see Figs. 5-17–5-19 for some examples). This approach can potentially be an SR evaluation as complementary to PSNR and SSIM, and it can also be used when there is no true HR image available.

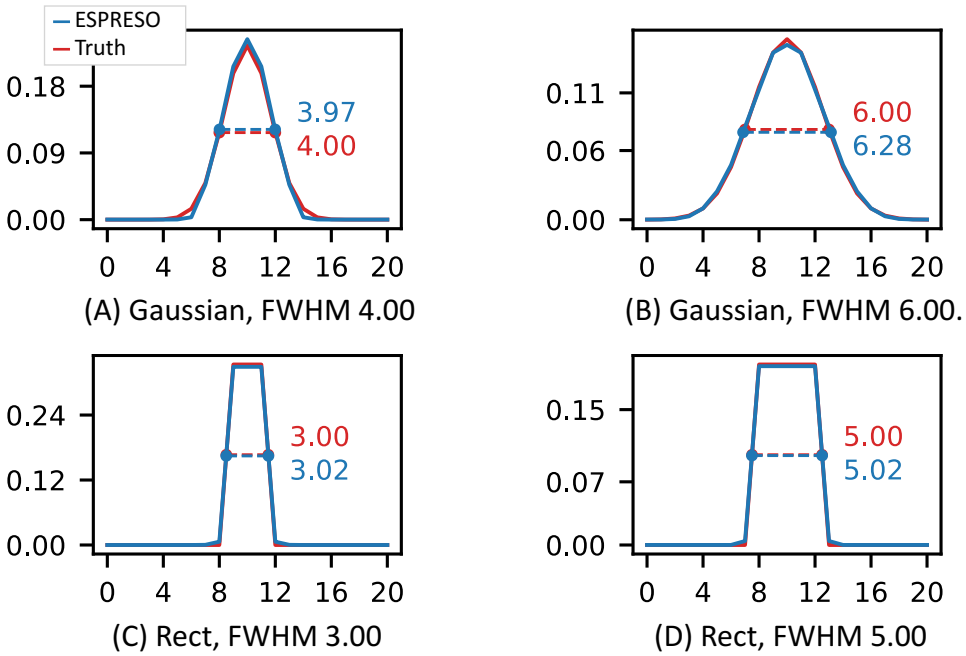


Figure 5-24. ESPRESO can recover the true relative slice profiles fairly well in the simulations without downsampling. The ESPRESO-estimated and the true relative slice profiles are plotted in blue and red, respectively. The settings of the true relative slice profiles in these example simulations are as follows. (A): Gaussian, FWHM 4.00. (B): Gaussian, FWHM 6.00. (C): rect, FWHM 3.00. (D): rect, FWHM 5.00. The FWHMs of these slice profiles are shown in the text in their corresponding colors.

However, we note that the standard deviations in Table 5-VII are very large. Given that the ESPRESO results of simulated blurs (with $s = 1$) from Section 5.5.1 are much more stable, we suspect that a super-resolved image cannot be simply modeled as a convolution between its corresponding HR image and a blur. This can be further investigated in the future.

5.5.3 Regularizations in ESPRESO

We use many regularizations in the training. The $l2$ weight decay of the generator encourages the smoothness of the estimated slice profile. Our generator network is actually a small-scale deep image prior network [134]. As proved in Cheng *et al.* [135],

Table 5-VII. Use ESPRESO to evaluate super-resolution (SR). ESPRESO is applied to super-resolved images from our simulations with Gaussian PSFs. The FWHMs of the resulting "slice profiles" from ESPRESO are shown as mean $\pm$ standard deviation in number of pixels.

Simulation settings		**Estimated FWHM after SR**
Scale factor	**FWHM**	
2.0	1.000	1.6093 $\pm$ 0.0932
	1.500	1.7750 $\pm$ 0.0927
	2.000	1.8667 $\pm$ 0.1498
	2.500	1.9136 $\pm$ 0.2236
3.5	1.750	2.3132 $\pm$ 0.2311
	2.625	2.6379 $\pm$ 0.4589
	3.500	2.8042 $\pm$ 0.6279
	4.375	2.9378 $\pm$ 0.7202
4.9	2.450	2.9504 $\pm$ 0.5067
	3.675	3.4168 $\pm$ 0.7733
	4.900	3.7407 $\pm$ 0.8721
	6.125	3.6733 $\pm$ 0.8813

by using an $l2$ weight decay in such a network, the deep image prior approximates a stationary Gaussian process prior, meaning that the values of the network output, i.e., the values of the slice profile vector in our case, are jointly Gaussian distributed; a key property is that the correlation of two values only depends on the distance between their coordinates, and the smaller the distance, the larger their correlation. Intuitively speaking, an $l2$ weight decay encourages the weights of a convolutional layer to be small but less sparsely distributed, which makes adjacent values of the output more similar. Such an "implicit" regularization works better than imposing "explicit" regularizations, such as an $l2$ norm, directly to the values of the slice profiles. We also use regularization in Eq. (5.7) to encourage the slice profile to have a single peak. Without this regularization, the estimated slice profile tends to have multiple peaks, as shown in Fig. 5-25.

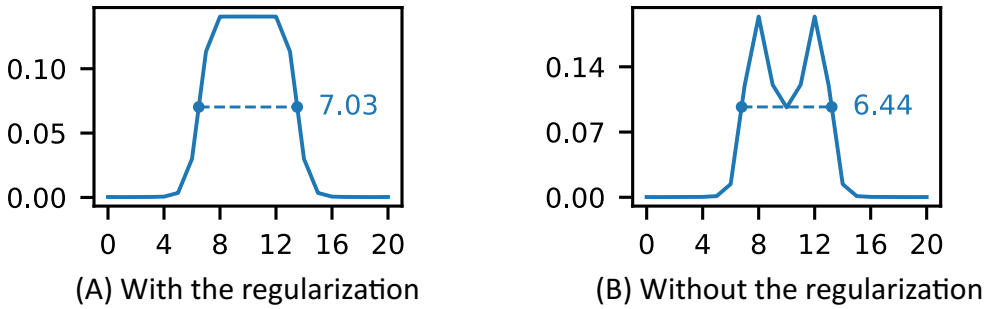


Figure 5-25. ESPRESO (A) with and (B) without using regularization in Eq. (5.7). The estimated slice profile has two peaks without this regularization in (B). The FWHMs of these slice profiles are shown in blue text.

5.5.4 Calculating the Real Slice Profile

ESPRESO estimates p instead of the real slice profile p_l. To see this clearly, we substitute Eq. (5.2) into Eq. (5.3), yielding

$$\begin{aligned}\tilde{I}_{hl} &= \{\{f_{xz} *_1 p_h *_2 p_l\} \downarrow_{(s_h,s_l)} *_1 p\} \downarrow_{(s,1)}, \\ \tilde{I}_{lh} &= \{\{f_{zx} *_1 p_l *_2 p_h\} \downarrow_{(s_l,s_h)} *_2 p\} \downarrow_{(1,s)} .\end{aligned} \tag{5.15}$$

Suppose there exist $f_{xz} = f_{zx}$ and $\tilde{I}_{hl} = \tilde{I}_{lh}$ (note that we assume that f_{xz} and f_{zx} have the same distribution, and $\tilde{I}_{hl}$ and $\tilde{I}_{lh}$ also have the same distribution). According to Eq. (5.15), it appears that p is a relative "difference" between p_h and p_l; i.e., convolving p_l is equivalent to convolving p_h then p. If we know p_h, we can then calculate an estimate of the real slice profile from p. Indeed, for certain MRI acquisitions, it is possible to know the in-plane PSF p_h (e.g., a sinc function for a Cartesian k-space sampling). According to Eq. (5.15), we can approximate p_l as the convolution between p and the digital version of p_h. However, as noted in Section 5.3.2, p has the same digital resolution with the in-plane direction of the image I. To more accurately calculate p_l from p and p_h, we can either upsample the input patch to the generator network or simply output a denser p but downsample it before convolving an image patch with it in the generator. Doing so

can also potentially improve the precision when calculating the FWHM of p.

5.5.5 Limitations of ESPRESO and S-SMORE

We assume that the in-plane, i.e., the readout and phase-encoding, directions have the same PSFs. If they are different, it is possible to estimate two relative slice profiles with respect to both directions, and we use both slice profiles when creating training data in S-SMORE. We used 45°-rotated x-y slices as validation data in S-SMORE. However, for Cartesian sampling, the diagonals of in-plane slices have larger k-space extends than the x and y axes, resulting in different physical resolution of these validation data compared to the training data in S-SMORE. A better way to build training and validation data of S-SMORE could be investigated in the future. In ESPRESO, we used a softmax to enforce the estimated slice profile to have positive values. However, we note that MRI signals are acquired as complex numbers; a slice profile can possibly have sidelobes with negative values. We could therefore remove the softmax and possibly use complex number operations in ESPRESO in the future. Although we use weight decay to regularize the slice profile as discussed in Section 5.5.3, we did not find a single set of hyper-parameters that works equally well for all shapes of slice profiles. ESPRESO also fails when the downsampling factor s is too large (approximately $s \geq 6.0$), producing results that are far from true slice profiles in our preliminary simulations. These limitations can be further investigated in the future.

5.5.6 Using T2w images in ACAPULCO

Using an SR T2w image in ACAPULCO can correct (to some extent) oversegmentation into the transverse/sigmoid sinuses to produce a better cerebellum parcellation. We note that our retrained T1w-only parcellating network in Section 5.4.5 does not have the

same performance as our previously published version [65] (see Tables 2-I and 5-VI). This is possibly due to the randomness during network training. We also note that the T dataset only includes five testing images. More images might be needed to more robustly evaluate our methods.

Although Method 2 with SR T2w images achieves the best performance, Method 1 using LR T2w images has a better average Dice coefficient than using SR T2w images, as shown in Table 5-VI. As mentioned above, this could be due to the randomness of training the networks and the small number of testing images for evaluation. In addition, we note that these images have a slice separation of 2.2 mm, which is not a very bad resolution, and we can see that the T2w image before and after SR are visually similar (see Fig. 5-22(B) and (C)); it is possible that SR does not bring much benefit in this case. However, we do observe that there is an improvement for Subset 1 of the OASIS-3 dataset by using an SR T2w image (see Fig. 5-23). These images have a slice separation of 4 mm; SR can improve their resolution to be closer to our training data, which may contribute to the better cerebellum parcellation.

We note that directly using a pair of T1w and T2w images as a dual-channel input to produce a parcellation (i.e., Method 1) is not necessarily better than using only the T1w image. Despite that using T2w can reduce the oversegmentation into sinuses, it provides worse parcellations of other regions such as the vermal lobules (see Table 5-VI). This could be due to the misalignment between T1w and T2w images which is caused by deformable motion of brain tissues; in addition, since we used LR/SR T2w images, they do not provide as fine details as the T1w images. Therefore, we designed Method 2 which only used paired T1w and T2w images to produce a cerebellum mask. We can then choose to only intersect this mask with the regions near the sinuses of a parcellation from a T1w-only network. Future work can include designing a network architecture to better incorporate different modalities.

5.6 Summary

In this chapter, we presented ESPRESO, an algorithm to estimate the (relative) slice profile of a 2D multi-slice MRI acquisition without external training data. We also implemented S-SMORE as a redesign of SMORE/iSMORE to super-resolve a 2D multi-slice MRI acquisition. We then incorporated ESPRESO into S-SMORE to further improve the SR performance. Finally, we applied these algorithms to T2w images to better parcellate the cerebellum.

Chapter 6

Conclusions and Future Work

6.1 Summary

In this dissertation, we developed algorithms to parcellate the cerebellum, conducted statistical analyses of cerebellum sub-regions, and developed algorithms to improve image resolution. In Chapter 2, we proposed ACAPULCO based on deep learning (DL) to parcellate the cerebellum. In Chapter 3, we explored incorporating anatomical knowledge into CNN architectures. In Chapter 4, we conducted longitudinal analyses of volume changes of cerebellar sub-regions during normal aging. In Chapter 5, we developed ESPRESO and S-SMORE to improve resolution of 2D multi-slice MRI images and further used super-resolved T2w images to improve cerebellum parcellation. In the following, we summarize each topic and discuss potential future work.

6.2 Parcellating the Cerebellum Into Its Sub-Regions

6.2.1 Key Points and Results

1. We developed ACAPULCO to parcellate the cerebellum from a T1w MRI image. ACAPULCO uses a locating network to detect the region that contains the cerebellum and uses a parcellating network to further parcellate the cerebellum within this detected region.

2. ACAPULCO was evaluated on two public datasets with manual delineations and achieved the state-of-the-art results. It achieved an average Dice coefficient of 0.7999 for the T dataset and an average Dice coefficient of 0.9097 for the M dataset.

3. We applied ACAPULCO to healthy subjects and subjects with spinocerebellar ataxia, Alzheimer's disease, or autism spectrum disorder to show its broad applicability.

4. We provide Singularity and Docker containers of ACAPULCO that are available to the public.

6.2.2 Future Work

DL-based algorithms typically require a large amount of training data to generalize better. However, it is very time-consuming and requires expertise to acquire manual delineations of brain structures. It is therefore of interest to utilize the vast amount of unlabeled MRI data by using semi-supervised methods [140] to improve the cerebellum parcellation. MRI images can also have flexible contrasts depending on the parameters of their acquisition protocols; networks that are trained on a specific MRI

dataset often do not perform as well on other datasets that are acquired with different protocols. Therefore, in addition to semi-supervised methods as mentioned above, MRI harmonization techniques [141] can also be used as a pre-processing step of cerebellum parcellation. Datasets that are delineated for other brain structures (such as the whole brain parcellation datasets in Huo *et al.* [142]) can also be used to pre-train the cerebellum parcellation networks.

We generally assume that each cerebellar sub-region has a spherical topology, i.e., a single connected-component without any holes or handles. Therefore, we can incorporate a topology constraint or encourage such a topology [143, 144] for each region in our method to better parcellate the cerebellum. Since the relative positions between these sub-regions are roughly fixed, it is also of interest to incorporate the information of their relative positions into cerebellum parcellation. We can also try other more advanced network architectures in our method.

6.3 Incorporating Anatomical Knowledge into Network Architectures

6.3.1 Key Points and Results

1. We incorporated the left-right symmetry of the brain into network architecture design. The proposed method implemented a 3D left-right-reflection (RE) equivariant network to segment brain structures that have paired or unpaired regions.

2. We incorporated the hierarchical organization of the cerebellum into network architecture design. The proposed network contains a tree-structured classifier with each node representing a cerebellar region and having child nodes that

further subdivide the region into finer substructures.

3. Although they did not improve on ACAPULCO with statistical significance, our explorations show promising results and can be further investigated in the future.

6.3.2 Future Work

In medical imaging processing, we always have a limited amount of labeled data. Therefore, instead of being purely data-driven, using prior knowledge of a problem can potentially improve our algorithms. Motivated by this, we incorporated anatomical knowledge into network architectures for better segmentation. Meanwhile, a better understanding of medical images is required not just in terms of anatomical knowledge but also in many other aspects such as the imaging device and procedure. Although our results do not improve on ACAPULCO with statistical significance, they do provide a promising research direction in the future.

Our RE network was only evaluated on the T dataset for the task of cerebellum parcellation; since the T dataset only contains five testing images, we can also evaluate our network on the M dataset in the future. According to our experiments, the RE network roughly spent 1.25 times of the GPU memory compared to a conventional network when their numbers of parameters are comparable. Data and model parallelism [145] can be investigated in the future to handle this problem. For our hierarchical network, a better way to organize the tree nodes of each region can be investigated in the future, potentially involving the use of conditional probability. More thorough experiments can also be done to compare it with ACAPULCO.

6.4 Conducting Statistical Analysis of Cerebellar Sub-Regional Volumes

6.4.1 Key Points and Results

1. We applied ACAPULCO to 822 cognitively normal subjects with 2,023 MRI images from the Baltimore Longitudinal Study of Aging (BLSA) and calculated the volumes of 28 cerebellar sub-regions.

2. We conducted longitudinal analyses of these volumes with respect to age and sex using linear mixed-effect models. Our findings suggest spatially varying atrophy patterns across the cerebellum with respect to age and sex both cross-sectionally and longitudinally.

6.4.2 Future Work

As mentioned in Section 6.2.2, ACAPULCO can be improved in various aspects to better parcellate the cerebellum. The BLSA images were acquired using different MRI protocols from the training data of ACAPULCO; MRI harmonization and semi-supervised learning can be used for a better generalizability. Additionally, since the BLSA subjects are mostly elderly subjects (50.1–95.1 years in our study), their images potentially have a different anatomical appearance from our training data (e.g., elderly subjects tend to have thicker cortical CSF). More thorough experiments might be needed in the future to investigate if our algorithm introduces bias because of this.

We only studied whether there were any correlations between cerebellar sub-regional volumes and age and sex. Future work can include studying if these volumes are correlated with cognitive and motor function tests. We only studied each individual volume

separately. A unified statistical analysis that includes all sub-regions together is also of interest in the future to take the correlations between sub-regions into consideration.

6.5 Super-Resolving MRI for Better Parcellation

6.5.1 Key Points and Results

1. We developed ESPRESO using a modified framework of the generative adversarial network to estimate the through-plane resolution of a given 2D multi-slice MRI image without external training data. ESPRESO was used in our super-resolution (SR) algorithm, S-SMORE, to create more faithful training data.

2. We implemented S-SMORE as an improved version of an internally supervised SR algorithm SMORE [76, 118]. We showed that S-SMORE performs better than SMORE.

3. We used S-SMORE with ESPRESO to super-resolve T2w images to be used in ACAPULCO. We found that using paired T1w and SR T2w images can prevent oversegmentation into transverse/sigmoid sinuses and improve cerebellum parcellation.

6.5.2 Future Work

We only tested ESPRESO in brain images. Further experiments in other regions of the body could be done in the future. We note that ESPRESO is based on the assumption of isotropy of the image—i.e., patches extracted from different orientations should have the same probability distribution after accounting for the resolutions. Future experiments can examine if this assumption is valid in different regions of interest. Other regularizations or

training schemes for ESPRESO can be investigated in the future to improve its accuracy. ESPRESO did not work well for scale factors that are greater than 6; this could be addressed in the future.

We used super-resolved T2w images to improve cerebellum parcellation. Our SR algorithm S-SMORE was trained with data created from the in-plane slices of the image. Since the in-plane slices are typically "thick" while the through-plane slices are "thin", this creates a discrepancy between the training data (in-plane slices) and testing data (through-plane slices). Although iterative SMORE [118] can address this problem to some extent, better use of external data might also be helpful. Self-supervised learning [146] is gaining more attention recently. In this technique, a pretext task is used to learn general visual features before the target task. A typical pretext task involves degenerating or altering an image (such as masking out some regions of this image) for the network to learn to recover the original image. An SR algorithm usually learns from simulated training data; i.e., a low-resolution (LR) image is generated from a high-resolution (HR) image to form a training pair. This procedure of simulating training data is similar to creating a pretext task. This might indicate that techniques from self-supervised learning can be possibly used in SR. Other forms of pretext tasks can also help to train a better SR network. Conventional supervised learning can be regarded as a maximum likelihood optimization. It should be also helpful to convert the training of SR into a maximum *a posteriori* optimization. While some regularizations such as the total-variation norm can be used as the prior, we can also use a generative adversarial network (GAN) to learn a probability distribution of the HR images as the the prior. This suggests that an effective combination of supervised learning and GAN training could be investigated in the future for better SR. An SR network is typically trained individually for a different scale factor and PSF. It is possible to learn a hyper-network [147] conditioned on the scale factor and PSF to output the weights for the SR network. In medical images,

we do not want SR to introduce spurious details. SR networks producing outputs that are guaranteed to be degraded to the input LR image, such as in Sønderby *et al.* [148], can be investigated in the future. We used paired T1w and T2w images to parcellate the cerebellum. Better network architectures to more effectively use multiple modalities can be investigated in the future.

6.6 Conclusions

In this dissertation, we developed DL algorithms to parcellate the cerebellum and analyzed longitudinal changes of cerebellar sub-regional volumes with respect to age and sex. Our work contributes to medical image processing techniques and advances our understanding of the cerebellum.

DL is a rapidly growing field in recent years. Researchers have being borrowing many techniques from DL into medical image processing, and we have been achieving great improvement. Meanwhile, we should always have a good understanding of the characteristics and uniqueness of medical images and adapt these techniques accordingly. In addition, I believe that a good algorithm should always try to address practical problems which can include both clinical usage (for diagnosis) and large-scale statistical studies (e.g., a study of neurological development of a population). Algorithm development also involves collaboration with radiologists and medical doctors to discover real valuable problems. In other words, our work should come from the practice and also take effect in practice. This was my primary motivation during my PhD study. I hope that the work presented in this dissertation can help move the field forward and have real value in healthcare and medical science.

References

1. D'Angelo, E. The Cerebellum Gets Social. *Science* **363,** 229–229 (2019).

2. Manto, M. *et al.* Consensus Paper: Roles of the Cerebellum in Motor Control—the Diversity of Ideas on Cerebellar Involvement in Movement. *Cerebellum* **11,** 457–487 (2012).

3. Buckner, R. L. The Cerebellum and Cognitive Function: 25 Years of Insight from Anatomy and Neuroimaging. *Neuron* **80,** 807–815 (2013).

4. Schmahmann, J. D. The Cerebellum and Cognition. *Neuroscience Letters* **688,** 62–75 (2019).

5. Schmahmann, J. D. *et al.* Three-Dimensional MRI Atlas of the Human Cerebellum in Proportional Stereotaxic Space. *NeuroImage* **10,** 233–260 (1999).

6. Mottolese, C. *et al.* Mapping Motor Representations in the Human Cerebellum. *Brain* **136,** 330–342 (2013).

7. Stoodley, C. J. & Schmahmann, J. D. Functional Topography of the Human Cerebellum. *Handbook of Clinical Neurology* **154,** 59–70 (2018).

8. Guell, X., Gabrieli, J. D. E. & Schmahmann, J. D. Triple Representation of Language, Working Memory, Social and Emotion Processing in the Cerebellum: Convergent Evidence from Task and Seed-Based Resting-State fMRI Analyses in a Single Large Cohort. *NeuroImage* **172,** 437–449 (2018).

9. Landman, B. A. *et al.* Multi-Parametric Neuroimaging Reproducibility: A 3-T Resource Study. *NeuroImage* **54,** 2854–2866 (2011).

10. Marvel, C. L., Faulkner, M. L., Strain, E. C., Mintzer, M. Z. & Desmond, J. E. An fMRI Investigation of Cerebellar Function during Verbal Working Memory in Methadone Maintenance Patients. *Cerebellum* **11,** 300–310 (2012).

11. Tiemeier, H. *et al.* Cerebellum Development during Childhood and Adolescence: A Longitudinal Morphometric MRI Study. *NeuroImage* **49,** 63–70 (2010).

12. Bernard, J. A., Leopold, D. R., Calhoun, V. D. & Mittal, V. A. Regional Cerebellar Volume and Cognitive Function from Adolescence to Late Middle Age. *Human Brain Mapping* **36,** 1102–1120 (2015).

13. Koppelmans, V. *et al.* Regional Cerebellar Volumetric Correlates of Manual Motor and Cognitive Function. *Brain Structure & Function* **222,** 1929–1944 (2017).

14. Kansal, K. *et al.* Structural Cerebellar Correlates of Cognitive and Motor Dysfunctions in Cerebellar Degeneration. *Brain* **140,** 707–720 (2017).

15. Laidi, C. *et al.* Cerebellar Anatomical Alterations and Attention to Eyes in Autism. *Scientific Reports* **7,** 12008 (2017).

16. Cocozza, S. *et al.* Cerebellar Lobule Atrophy and Disability in Progressive MS. *Journal of Neurology, Neurosurgery, and Psychiatry* **88,** 1065–1072 (2017).

17. Diedrichsen, J. A Spatially Unbiased Atlas Template of the Human Cerebellum. *NeuroImage* **33,** 127–138 (2006).

18. Park, M. T. M. *et al.* Derivation of High-Resolution MRI Atlases of the Human Cerebellum at 3T and Segmentation Using Multiple Automatically Generated Templates. *NeuroImage* **95,** 217–231 (2014).

19. Carass, A. *et al.* Comparing Fully Automated State-of-the-Art Cerebellum Parcellation from Magnetic Resonance Images. *NeuroImage* **183,** 150–172 (2018).

20. Brant-Zawadzki, M., Gillan, G. D. & Nitz, W. R. MP RAGE: A Three-Dimensional, T1-Weighted, Gradient-Echo Sequence—Initial Experience in the Brain. *Radiology* **182,** 769–775 (1992).

21. Hennig, J., Nauerth, A. & Friedburg, H. RARE Imaging: A Fast Imaging Method for Clinical MR. *Magnetic Resonance in Medicine* **3,** 823–833 (1986).

22. LaMontagne, P. J. *et al.* OASIS-3: Longitudinal Neuroimaging, Clinical, and Cognitive Dataset for Normal Aging and Alzheimer Disease. *medRxiv* (2019).

23. Liang, Z.-P. & Lauterbur, P. C. *Principles of Magnetic Resonance Imaging: A Signal Processing Perspective* 1st (SPIE Optical Engineering Press, 2000).

24. Bernstein, M. A., King, K. F. & Zhou, X. J. *Handbook of MRI Pulse Sequences* 1st (Academic Press, 2004).

25. Prince, J. L. & Links, J. M. *Medical Imaging Signals and Systems* 2nd (Pearson, 2014).

26. Sled, J., Zijdenbos, A. & Evans, A. A Nonparametric Method for Automatic Correction of Intensity Nonuniformity in MRI Data. *IEEE Transactions on Medical Imaging* **17,** 87–97 (1998).

27. Tustison, N. J. *et al.* N4ITK: Improved N3 Bias Correction. *IEEE Transactions on Medical Imaging* **29,** 1310–1320 (2010).

28. Nyúl, L. G. & Udupa, J. K. On Standardizing the MR Image Intensity Scale. *Magnetic Resonance in Medicine* **42,** 1072–1081 (1999).

29. Shinohara, R. T. *et al.* Statistical Normalization Techniques for Magnetic Resonance Imaging. *NeuroImage : Clinical* **6,** 9 (2014).

30. Reinhold, J. C., Dewey, B. E., Carass, A. & Prince, J. L. Evaluating the Impact of Intensity Normalization on MR Image Synthesis. *Proceedings of SPIE–the International Society for Optical Engineering* **10949,** 109493H (2019).

31. Fonov, V., Evans, A., McKinstry, R., Almli, C. & Collins, D. Unbiased Nonlinear Average Age-Appropriate Brain Templates from Birth to Adulthood. *NeuroImage* **47,** S102 (2009).

32. Evans, A. C., Janke, A. L., Collins, D. L. & Baillet, S. Brain Templates and Atlases. *NeuroImage* **62,** 911–922 (2012).

33. Iglesias, J. E., Liu, C.-Y., Thompson, P. M. & Tu, Z. Robust Brain Extraction across Datasets and Comparison with Publicly Available Methods. *IEEE Transactions on Medical Imaging* **30,** 1617–1634 (2011).

34. Valverde, S., Oliver, A. & Lladó, X. A White Matter Lesion-Filling Approach to Improve Brain Tissue Volume Measurements. *NeuroImage : Clinical* **6,** 86–92 (2014).

35. Diedrichsen, J., Balsters, J. H., Flavell, J., Cussans, E. & Ramnani, N. A Probabilistic MR Atlas of the Human Cerebellum. *NeuroImage* **46,** 39–46 (2009).

36. Powell, S. *et al.* Registration and Machine Learning-Based Automated Segmentation of Subcortical and Cerebellar Brain Structures. *NeuroImage* **39,** 238–247 (2008).

37. Bogovic, J. A., Bazin, P.-L., Ying, S. H. & Prince, J. L. Automated Segmentation of the Cerebellar Lobules Using Boundary Specific Classification and Evolution. *International Conference on Information Processing in Medical Imaging* **23,** 62–73 (2013).

38. Bogovic, J. A., Prince, J. L. & Bazin, P.-L. A Multiple Object Geometric Deformable Model for Image Segmentation. *Computer Vision and Image Understanding: CVIU* **117,** 145–157 (2013).

39. Carass, A. & Prince, J. L. An Overview of the Multi-Object Geometric Deformable Model Approach in Biomedical Imaging. *Medical Image Recognition, Segmentation, and Parsing,* 259–279 (2016).

40. Chakravarty, M. M. *et al.* Performing Label-Fusion-Based Segmentation Using Multiple Automatically Generated Templates. *Human Brain Mapping* **34,** 2635–2654 (2013).

41. Price, M., Cardenas, V. A. & Fein, G. Automated MRI Cerebellar Size Measurements Using Active Appearance Modeling. *NeuroImage* **103,** 511–521 (2014).

42. Weier, K., Fonov, V., Lavoie, K., Doyon, J. & Collins, D. L. Rapid Automatic Segmentation of the Human Cerebellum and Its Lobules (RASCAL)—Implementation and Application of the Patch-Based Label-Fusion Technique with a Template Library to Segment the Human Cerebellum. *Human Brain Mapping* **35,** 5026–5039 (2014).

43. Romero, J. E. *et al.* CERES: A New Cerebellum Lobule Segmentation Method. *NeuroImage* **147,** 916–924 (2017).

44. Yang, Z. *et al.* Automated Cerebellar Lobule Segmentation with Application to Cerebellar Structural Analysis in Cerebellar Disease. *NeuroImage* **127,** 435–444 (2016).

45. van der Lijn, F. *et al. Cerebellum Segmentation in MRI Using Atlas Registration and Local Multi-Scale Image Descriptors* in *2009 IEEE International Symposium on Biomedical Imaging: From Nano to Macro* (2009), 221–224.

46. Plassard, A. J. *et al. Improving Cerebellar Segmentation with Statistical Fusion* in *Medical Imaging 2016: Image Processing* **9784** (2016), 753–759.

47. Nair, V. & Hinton, G. E. Rectified Linear Units Improve Restricted Boltzmann Machines. *Proceedings of the 27th International Conference on Machine Learning,* 807–814 (2010).

48. Tompson, J., Goroshin, R., Jain, A., LeCun, Y. & Bregler, C. *Efficient Object Localization Using Convolutional Networks* in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2015), 648–656.

49. Ioffe, S. & Szegedy, C. *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift* in *Proceedings of the 32nd International Conference on Machine Learning* (2015), 448–456.

50. Woo, S., Park, J., Lee, J.-Y. & Kweon, I. S. *CBAM: Convolutional Block Attention Module* in *Computer Vision — ECCV 2018* **11211** (2018), 3–19.

51. Hornik, K., Stinchcombe, M. & White, H. Multilayer Feedforward Networks Are Universal Approximators. *Neural Networks* **2,** 359–366 (1989).

52. Goodfellow, I. *et al. Generative Adversarial Nets* in *Advances in Neural Information Processing Systems* **27** (2014).

53. Mirza, M. & Osindero, S. Conditional Generative Adversarial Nets. *arXiv:1411.1784 [cs, stat].* arXiv: 1411.1784 [cs, stat] (2014).

54. Rumelhart, D. E., Hinton, G. E. & Williams, R. J. Learning Representations by Back-Propagating Errors. *Nature* **323,** 533–536 (1986).

55. Kingma, D. P. & Ba, J. Adam: A Method for Stochastic Optimization. *arXiv:1412.6980 [cs].* arXiv: 1412.6980 [cs] (2017).

56. Shorten, C. & Khoshgoftaar, T. M. A Survey on Image Data Augmentation for Deep Learning. *Journal of Big Data* **6,** 60 (2019).

57. Krizhevsky, A., Sutskever, I. & Hinton, G. E. *ImageNet Classification with Deep Convolutional Neural Networks* in *Advances in Neural Information Processing Systems* **25** (2012).

58. Simonyan, K. & Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv:1409.1556 [cs].* arXiv: 1409.1556 [cs] (2015).

59. He, K., Zhang, X., Ren, S. & Sun, J. *Deep Residual Learning for Image Recognition* in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2016), 770–778.

60. He, K., Zhang, X., Ren, S. & Sun, J. *Identity Mappings in Deep Residual Networks* in *Computer Vision — ECCV 2016* (2016), 630–645.

61. Bello, I. *et al.* Revisiting ResNets: Improved Training and Scaling Strategies. *arXiv:2103.07579 [cs].* arXiv: 2103.07579 [cs] (2021).

62. Long, J., Shelhamer, E. & Darrell, T. *Fully Convolutional Networks for Semantic Segmentation* in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2015), 3431–3440.

63. Ronneberger, O., Fischer, P. & Brox, T. *U-Net: Convolutional Networks for Biomedical Image Segmentation* in *Medical Image Computing and Computer-Assisted Intervention — MICCAI 2015* (2015), 234–241.

64. Han, S., He, Y., Carass, A., Ying, S. H. & Prince, J. L. *Cerebellum Parcellation with Convolutional Neural Networks* in *Medical Imaging 2019: Image Processing* **10949** (2019), 143–148.

65. Han, S., Carass, A., He, Y. & Prince, J. L. Automatic Cerebellum Anatomical Parcellation Using U-Net with Locally Constrained Optimization. *NeuroImage* **218,** 116819 (2020).

66. Cohen, T. & Welling, M. *Group Equivariant Convolutional Networks* in *Proceedings of The 33rd International Conference on Machine Learning* (2016), 2990–2999.

67. Han, S., Prince, J. L. & Carass, A. *Reflection-Equivariant Convolutional Neural Networks Improve Segmentation over Reflection Augmentation* in *Medical Imaging 2020: Image Processing* **11313** (2020), 806–813.

68. Luft, A. R. *et al.* Patterns of Age-Related Dhrinkage in Verebellum and Brainstem Observed In Vivo Using Three-Dimensional MRI Volumetry. *Cerebral Cortex* **9,** 712–721 (1999).

69. Bernard, J. A. & Seidler, R. D. Relationships between Regional Cerebellar Volume and Sensorimotor and Cognitive Function in Young and Older Adults. *The Cerebellum* **12,** 721–737 (2013).

70. Steele, C. J. & Chakravarty, M. M. Gray-Matter Structural Variability in the Human Cerebellum: Lobule-Specific Differences across Sex and Hemisphere. *NeuroImage* **170,** 164–173 (2018).

71. Shock, N. W. *et al.* Normal Human Aging: The Baltimore Longitudinal Study on Aging. *Washington, D.C: National Institutes of Health* (1984).

72. Han, S., An, Y., Carass, A., Prince, J. L. & Resnick, S. M. Longitudinal Analysis of Regional Cerebellum Volumes during Normal Aging. *NeuroImage* **220,** 117062 (2020).

73. Pinheiro, J., Bates, D., DebRoy, S., Sarkar, D. & R Core Team. *nlme: Linear and Nonlinear Mixed Effects Models* R package version 3.1-140 (2019).

74. Han, S., Carass, A., Schär, M., Calabresi, P. A. & Prince, J. L. *Slice Profile Estimation from 2D MRI Acquisition Using Generative Adversarial Networks* in *2021 IEEE 18th International Symposium on Biomedical Imaging (ISBI)* (2021), 145–149.

75. Han, S., Remedios, S., Carass, A., Schär, M. & Prince, J. L. *MR Slice Profile Estimation by Learning to Match Internal Patch Distributions* in *Information Processing in Medical Imaging* **12729** (2021), 108–119.

76. Zhao, C. *et al.* SMORE: A Self-Supervised Anti-Aliasing and Super-Resolution Algorithm for MRI Using Deep Learning. *IEEE Transactions on Medical Imaging* **40,** 805–817.

77. Zhang, Y. *et al. Image Super-Resolution Using Very Deep Residual Channel Attention Networks* in *Computer Vision — ECCV 2018* **11211** (2018), 294–310.

78. Shi, W. *et al.* Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network. *arXiv:1609.05158 [cs, stat].* arXiv: 1609.05158 [cs, stat] (2016).

79. Christ, P. F. *et al. Automatic Liver and Lesion Segmentation in CT Using Cascaded Fully Convolutional Neural Networks and 3D Conditional Random Fields* in *Medical Image Computing and Computer-Assisted Intervention — MICCAI 2016* (2016), 415–423.

80. Haghighi, M., Warfield, S. K. & Kurugol, S. *Automatic Renal Segmentation in DCE-MRI Using Convolutional Neural Networks* in *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)* (2018), 1534–1537.

81. Liu, J. *et al. Cascaded Coarse-to-Fine Convolutional Neural Networks for Pericardial Effusion Localization and Segmentation on CT Scans* in *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)* (2018), 1092–1095.

82. Wachinger, C., Reuter, M. & Klein, T. DeepNAT: Deep Convolutional Neural Network for Segmenting Neuroanatomy. *NeuroImage* **170,** 434–445 (2018).

83. Kayalibay, B., Jensen, G. & van der Smagt, P. CNN-Based Segmentation of Medical Imaging Data. *arXiv:1701.03056 [cs]*. arXiv: 1701.03056 [cs] (2017).

84. Di Martino, A. *et al.* Enhancing Studies of the Connectome in Autism Using the Autism Brain Imaging Data Exchange II. *Scientific Data* **4,** 170010 (2017).

85. Carass, A. *et al.* Simple Paradigm for Extra-Cerebral Tissue Removal: Algorithm and Analysis. *NeuroImage* **56,** 1982–1992 (2011).

86. Ulyanov, D., Vedaldi, A. & Lempitsky, V. *Improved Texture Networks: Maximizing Quality and Diversity in Feed-Forward Stylization and Texture Synthesis* in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2017), 6924–6932.

87. He, K., Zhang, X., Ren, S. & Sun, J. *Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification* in *Proceedings of the IEEE International Conference on Computer Vision* (2015), 1026–1034.

88. Girshick, R. *Fast R-CNN* in *Proceedings of the IEEE International Conference on Computer Vision* (2015), 1440–1448.

89. He, K., Gkioxari, G., Dollar, P. & Girshick, R. *Mask R-CNN* in *Proceedings of the IEEE International Conference on Computer Vision* (2017), 2961–2969.

90. Solov'ev, S. V. The Weight and Linear Dimensions of the Human Cerebellum. *Neuroscience and Behavioral Physiology* **36,** 479–481 (2006).

91. Çiçek, Ö., Abdulkadir, A., Lienkamp, S. S., Brox, T. & Ronneberger, O. *3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation* in *Medical Image Computing and Computer-Assisted Intervention — MICCAI 2016* (2016), 424–432.

92. Sudre, C. H., Li, W., Vercauteren, T., Ourselin, S. & Jorge Cardoso, M. Generalised Dice Overlap as a Deep Learning Loss Function for Highly Unbalanced Segmentations. *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support* **2017,** 240–248 (2017).

93. Dice, L. R. Measures of the Amount of Ecologic Association between Species. *Ecology* **26,** 297–302 (1945).

94. Bartko, J. J. The Intraclass Correlation Coefficient as a Measure of Reliability. *Psychological Reports* **19,** 3–11 (1966).

95. Koo, T. K. & Li, M. Y. A Guideline of Selecting and Reporting Intraclass Correlation Coefficients for Reliability Research. *Journal of Chiropractic Medicine* **15,** 155–163 (2016).

96. Tange, O. *GNU Parallel 2018* (Ole Tange, Mar. 2018).

97. Wu, Y. & He, K. *Group Normalization* in *Proceedings of the European Conference on Computer Vision (ECCV)* (2018), 3–19.

98. Nalepa, J., Marcinkiewicz, M. & Kawulok, M. Data Augmentation for Brain-Tumor Segmentation: A Review. *Frontiers in Computational Neuroscience* **13,** 83 (2019).

99. Linmans, J., Winkens, J., Veeling, B. S., Cohen, T. S. & Welling, M. Sample Efficient Semantic Segmentation Using Rotation Equivariant Convolutional Network. *arXiv:1807.00583 [cs].* arXiv: 1807.00583 [cs] (2018).

100. Winkels, M. & Cohen, T. S. Pulmonary Nodule Detection in CT Scans with Equivariant CNNs. *Medical Image Analysis* **55,** 15–26 (2019).

101. Razzak, M. I., Imran, M. & Xu, G. Efficient Brain Tumor Segmentation with Multiscale Two-Pathway-Group Conventional Neural Networks. *IEEE Journal of Biomedical and Health Informatics* **23,** 1911–1919 (2019).

102. Puccio, B., Pooley, J. P., Pellman, J. S., Taverna, E. C. & Craddock, R. C. The Preprocessed Connectomes Project Repository of Manually Corrected Skull-Stripped T1-weighted Anatomical MRI Data. *GigaScience* **5,** 45 (2016).

103. Landman, B. A. & Warfield, S. K. *MICCAI 2012 Workshop on Multi-Atlas Labeling* in *MICCAI Grand Challenge and Workshop on Multi-Atlas Labeling* (2012).

104. Bogovic, J. A. *et al.* Approaching Expert Results Using a Hierarchical Cerebellum Parcellation Protocol for Multiple Inexpert Human Raters. *NeuroImage* **64,** 616–629 (2013).

105. Liang, X., Xing, E. & Zhou, H. *Dynamic-Structured Semantic Propagation Network* in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2018), 752–761.

106. Tian, Q. *et al.* The Brain Map of Gait Variability in Aging, Cognitive Impairment and Dementia—A Systematic Review. *Neuroscience & Biobehavioral Reviews* **74,** 149–162 (2017).

107. Goh, J. O., An, Y. & Resnick, S. M. Differential Trajectories of Age-Related Changes in Components of Executive and Memory Processes. *Psychology and Aging* **27,** 707–719 (2012).

108. McCarrey, A. C., An, Y., Kitner-Triolo, M. H., Ferrucci, L. & Resnick, S. M. Sex Differences in Cognitive Trajectories in Clinically Normal Older Adults. *Psychology and Aging* **31,** 166–175 (2016).

109. Raz, N. *et al.* Differential Age-Related Changes in the Regional Metencephalic Volumes in Humans: A 5-Year Follow-Up. *Neuroscience Letters* **349,** 163–166 (2003).

110. Raz, N. *et al.* Regional Brain Changes in Aging Healthy Adults: General Trends, Individual Differences and Modifiers. *Cerebral Cortex* **15,** 1676–1689 (2005).

111. Raz, N., Ghisletta, P., Rodrigue, K. M., Kennedy, K. M. & Lindenberger, U. Trajectories of Brain Aging in Middle-Aged and Older Adults: Regional and Individual Differences. *NeuroImage* **51,** 501–511 (2010).

112. Raz, N. *et al.* Differential Brain Shrinkage Over 6 Months Shows Limited Association with Cognitive Practice. *Brain and Cognition* **82,** 171–180 (2013).

113. Laird, N. M. & Ware, J. H. Random-Effects Models for Longitudinal Data. *Biometrics* **38,** 963–974 (1982).

114. Armstrong, N. M. *et al.* Predictors of Neurodegeneration Differ between Cognitively Normal and Subsequently Impaired Older Adults. *Neurobiology of Aging* **75,** 178–186 (2019).

115. Doshi, J., Erus, G., Ou, Y., Gaonkar, B. & Davatzikos, C. Multi-Atlas Skull-Stripping. *Academic Radiology* **20,** 1566–1576 (2013).

116. Bretz, F., Hothorn, T. & Westfall, P. *Multiple Comparisons Using R* (2010).

117. Joo, J. W. J., Hormozdiari, F., Han, B. & Eskin, E. Multiple Testing Correction in Linear Mixed Models. *Genome Biology* **17,** 62 (2016).

118. Zhao, C., Son, S., Kim, Y. & Prince, J. L. *iSMORE: An Iterative Self Super-Resolution Algorithm* in *Simulation and Synthesis in Medical Imaging* (2019), 130–139.

119. Zhao, C. *et al.* Applications of a Deep Learning Method for Anti-Aliasing and Super-Resolution in MRI. *Magnetic Resonance Imaging* **64,** 132–141 (2019).

120. Greenspan, H. Super-Resolution in Medical Imaging. *The Computer Journal* **52,** 43–63 (2009).

121. Oktay, O. *et al. Multi-Input Cardiac Image Super-Resolution Using Convolutional Neural Networks* in *Medical Image Computing and Computer-Assisted Intervention — MICCAI 2016* (2016), 246–254.

122. Pham, C.-H. *et al.* Multiscale Brain MRI Super-Resolution Using Deep 3D Convolutional Networks. *Computerized Medical Imaging and Graphics* **77,** 101647 (2019).

123. Pauly, J., Le Roux, P., Nishimura, D. & Macovski, A. Parameter Relations for the Shinnar-Le Roux Selective Excitation Pulse Design Algorithm (NMR Imaging). *IEEE Transactions on Medical Imaging* **10,** 53–65 (1991).

124. Chen, Y. *et al. Efficient and Accurate MRI Super-Resolution Using a Generative Adversarial Network and 3D Multi-Level Densely Connected Network* in *Medical Image Computing and Computer Assisted Intervention — MICCAI 2018* (2018), 91–99.

125. Lyu, Q. *et al.* Multi-Contrast Super-Resolution MRI through a Progressive Network. *IEEE Transactions on Medical Imaging* **39,** 2738–2749 (2020).

126. Du, J. *et al.* Super-Resolution Reconstruction of Single Anisotropic 3D MR Images Using Residual Convolutional Neural Network. *Neurocomputing* **392,** 209–220 (2020).

127. Schneider, E. *et al.* The Osteoarthritis Initiative (OAI) Magnetic Resonance Imaging Quality Assurance Methods and Results. *Osteoarthritis and Cartilage* **16,** 994–1004 (2008).

128. Lerski, R. A. An Evaluation Using Computer Simulation of Two Methods of Slice Profile Determination in MRI. *Physics in Medicine and Biology* **34,** 1931–1937 (1989).

129. Liu, H., Michel, E., Casey, S. O. & Truwit, C. L. *Actual Imaging Slice Profile of 2D MRI* in *Medical Imaging 2002: Physics of Medical Imaging* **4682** (2002), 767–773.

130. Jog, A., Carass, A. & Prince, J. L. *Self Super-Resolution for Magnetic Resonance Images* in *Medical Image Computing and Computer-Assisted Intervention — MICCAI 2016* (2016), 553–560.

131. Deng, S. *et al. Isotropic Reconstruction of 3D EM Images with Unsupervised Degradation Learning* in *Medical Image Computing and Computer Assisted Intervention — MICCAI 2020* (2020), 163–173.

132. Bell-Kligler, S., Shocher, A. & Irani, M. *Blind Super-Resolution Kernel Estimation Using an Internal-GAN* in *Advances in Neural Information Processing Systems* **32** (2019).

133. Lim, B., Son, S., Kim, H., Nah, S. & Mu Lee, K. *Enhanced Deep Residual Networks for Single Image Super-Resolution* in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops* (2017), 136–144.

134. Ulyanov, D., Vedaldi, A. & Lempitsky, V. Deep Image Prior. *International Journal of Computer Vision* **128,** 1867–1888 (2020).

135. Cheng, Z., Gadelha, M., Maji, S. & Sheldon, D. *A Bayesian Perspective on the Deep Image Prior* in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2019), 5438–5446.

136. Maas, A. L., Hannun, A. Y. & Ng, A. Y. *Rectifier Nonlinearities Improve Neural Network Acoustic Models* in *In ICML Workshop on Deep Learning for Audio, Speech and Language Processing* (2013).

137. Miyato, T., Kataoka, T., Koyama, M. & Yoshida, Y. *Spectral Normalization for Generative Adversarial Networks* in *International Conference on Learning Representations* (2018).

138. Smith, L. N. & Topin, N. *Super-Convergence: Very Fast Training of Neural Networks Using Large Learning Rates* in *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications* **11006** (2019), 369–386.

139. Delbracio, M. & Sapiro, G. Hand-Held Video Deblurring via Efficient Fourier Aggregation. *IEEE Transactions on Computational Imaging* **1,** 270–283 (2015).

140. Yang, X., Song, Z., King, I. & Xu, Z. A Survey on Deep Semi-Supervised Learning. *arXiv:2103.00550 [cs].* arXiv: 2103.00550 [cs] (2021).

141. Zuo, L. *et al.* Unsupervised MR Harmonization by Learning Disentangled Representations Using Information Bottleneck Theory. *NeuroImage* **243,** 118569 (2021).

142. Huo, Y. *et al.* 3D Whole Brain Segmentation Using Spatially Localized Atlas Network Tiles. *NeuroImage* **194,** 105–119 (2019).

143. Han, X., Xu, C. & Prince, J. A Topology Preserving Level Set Method for Geometric Deformable Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **25,** 755–768 (2003).

144. Zeng, Q. *et al. Liver Segmentation in Magnetic Resonance Imaging via Mean Shape Fitting with Fully Convolutional Neural Networks* in *Medical Image Computing and Computer Assisted Intervention — MICCAI 2019* (2019), 246–254.

145. Ben-Nun, T. & Hoefler, T. Demystifying Parallel and Distributed Deep Learning: An in-Depth Concurrency Analysis. *ACM Computing Surveys* **52,** 65:1–65:43 (2019).

146. Ohri, K. & Kumar, M. Review on Self-Supervised Image Recognition Using Deep Neural Networks. *Knowledge-Based Systems* **224,** 107090 (2021).

147. Ha, D., Dai, A. M. & Le, Q. V. *HyperNetworks* in *International Conference for Learning Representations* (2017).

148. Sønderby, C. K., Caballero, J., Theis, L., Shi, W. & Huszár, F. *Amortised MAP Inference for Image Super-Resolution* in *The International Conference on Learning Representations* (2017).

Vita

Shuo Han received his Bachelor degree in Biomedical Engineering with a minor in Computer Technology and Application from Tsinghua University, Beijing, China in 2014. He received his degree of Master of Science in Engineering in Biomedical Engineering from the Johns Hopkins University, Maryland, USA in 2016. He was subsequently enrolled in the Biomedical Engineering PhD program in the Johns Hopkins University and joined the Image Analysis and Communications Laboratory under the supervision of Dr. Jerry L. Prince. His research focuses on neuroimaging and medical image processing and analysis.