



SAHLGRENSKA ACADEMY

Anatomical segmentation of the human brain: comparative assessment of two automatic methods

Carl von Dorrien

Essay/Thesis:	30 hp
Program and/or course:	Medical Physics
Level:	Second Cycle
Term/year:	Spring 2023
Supervisor:	Rolf A. Heckemann
Examiner:	Magnus Båth

Abstract

Essay/Thesis:	30 hp
Program and/or course:	Medical Physics
Level:	Second Cycle
Term/year:	Spring 2023
Supervisor:	Rolf A. Heckemann
Examiner:	Magnus Båth
Keywords:	MAPER, FreeSurfer, FastSurfer, Automatic brain segmentation, MRI, Deep Neural Network, Brain

Magnetic Resonance Imaging (MRI) is a robust and versatile imaging modality and an integral component of a lot of studies, especially when performing quantitative analysis. MRI is the preferred method of imaging the brain because of its excellent soft tissue contrast. Accurate segmentation of the brain into its anatomical regions enables accurate quantitative analysis of the brain. Three software programs that perform automatic anatomical segmentation of the human brain are FreeSurfer, FastSurfer and MAPER. The purpose of this study was to use FreeSurfer as a baseline, and to investigate how well FastSurfer and MAPER segmentations conform to FreeSurfer's outputs on the same dataset. 185 T1-weighted 3D MR images from the IXI Dataset were segmented using FreeSurfer, FastSurfer, and MAPER. Default training checkpoints were used for FastSurfer and FreeSurfer outputs of the IXI Dataset, along with corresponding brain MR images, were used as a source atlas for MAPER. The FastSurfer and MAPER segmentations were then compared with the FreeSurfer segmentations using the Jaccard Similarity Coefficient. MAPER performed better than FastSurfer at replicating FreeSurfer-conform outputs for subcortical regions. MAPER and FastSurfer performed similarly for the cortical regions.

Contents

List of Figures	iii
List of tables	iii
1 Introduction	1
1.1 FreeSurfer	1
1.2 FastSurfer	2
1.3 MAPER.....	3
1.4 Purpose	3
1.5 Structure of this thesis	3
2 Method	4
2.1 IXI Dataset	4
2.2 Segmentation of the IXI dataset	4
2.2.1 FreeSurfer segmentation	4
2.2.2 FastSurfer segmentation.....	4
2.2.3 MAPER segmentation	4
2.3 Image postprocessing and result generation.....	5
2.3.1 Jaccard Similarity Coefficient	5
2.3.2 Postprocessing	5
2.3.3 Generating results using R and label exclusion.....	5
2.3.4 Statistical analysis	7
2.4 Hardware and programming languages	7
3 Results.....	8
3.1 Segmentation results	8
3.1.1 Overall Jaccard scores	8
3.1.2 Cortical Jaccard scores	8
3.1.3 Subcortical Jaccard scores.....	9
3.2 Regional segmentation differences	10
3.3 MAPER source subjects.....	12
4 Narrative account	13
4.1 FastSurferCNN training	13
4.1.1 Training dataset	13
4.1.2 Training	13
4.2 Training results.....	14
4.3 Training discussion.....	15
5 Discussion.....	16
6 Conclusion.....	17
Acknowledgements.....	18
7 References	19
8 Appendix 1	1
8.1 Folder structure and change of file format.....	1
8.2 FreeSurfer segmentation code	2
8.3 FastSurfer segmentation code	2
8.4 MAPER segmentation code.....	2

8.5	Training FastSurferCNN.....	3
8.5.1	Hdf5 generation	3
8.5.2	Training	3
9	Appendix 2	4
9.1	Scan parameters for the IXI dataset	4
9.2	Scan parameters for the Hammer’s atlas dataset	4

List of Figures

Figure 1.	Figure showing 3D anatomical labels after a successful segmentation using FreeSurfer (right) together with an axial slice of the original 3D MR-image (left). Visualization is done using Freeview, a program included in the FreeSurfer toolkit.....	1
Figure 2.	Schematic over FastSurferCNN. To the right is an overview of how the F-CNNs (Fully Convolutional Neural Networks) operate on their own and are summarised using view aggregation. The structure of each F-CNN is on the left. Image source: deep-mi.org/research/fastsurfer/	2
Figure 3.	Violin plot showing the distribution of the mean JC for each subject by comparison, FreeSurfer-FastSurfer and FreeSurfer-MAPER. The plot includes all labels. The line in the boxplot is the median, the box edges are the 25 th and 75 th percentile, and the black dots are outliers.	8
Figure 4.	Violin plot showing the distribution of the mean JC for each subject by comparison, FreeSurfer-FastSurfer and FreeSurfer-MAPER. The plot only includes cortical labels. The line in the boxplot is the median, the box edges are the 25 th and 75 th percentile, and the black dots are outliers.	9
Figure 5.	Violin plot showing the distribution of the mean JC for each subject by comparison, FreeSurfer-FastSurfer and FreeSurfer-MAPER. The plot only includes subcortical labels. The line in the boxplot is the median, the box edges are the 25 th and 75 th percentile, and the black dots are outliers.	9
Figure 6.	Figure showing FastSurfer segmentation using different checkpoints of a test image. Segmentation using default checkpoints to the left (control), checkpoints from training using Hammers atlas data in the middle-left, and checkpoints from training using FreeSurfer outputs of the Hammers atlas to the middle-right. Segmentation of one of the images used for validation (a17), using checkpoints from training using FreeSurfer outputs of the Hammers atlas, can be seen on the right.....	14

List of tables

Table 1.	Table containing all labels segmented by either FreeSurfer, FastSurfer or MAPER. Labels excluded from the results are marked with an asterisk. Labels numbered 2 to 255 are subcortical regions while labels numbered above 1000 are cortical regions. Label numbers and names taken from FreeSurfer’s look-up table (FreeSurferColorLUT).	6
Table 2.	Table containing, for all labels, the mean JC, along with the standard deviation, and the diff-value, sorted by diff-values (descending).	10
Table 3.	Mean JC for subjects used as source data for MAPER compared to the rest of the dataset. Comparison between FreeSurfer and FastSurfer, and FreeSurfer and MAPER.....	12

1 Introduction

Magnetic Resonance Imaging (MRI) is a robust and versatile imaging modality and an integral component of many studies, especially when performing quantitative analysis. MR imaging is the preferred method of imaging the brain because of its excellent soft tissue contrast.

Accurate segmentation of the brain into its anatomical regions enables accurate quantitative analysis of the brain. Analysis of the features of the brain can be used to understand the impact of various diseases on brain structure. Automatic segmentation of the brain could also help radiologists by making their workflow less time-consuming [1, 2].

The traditional neuroimaging pipelines that automatically segment and evaluate incoming MR images suffer from high runtimes and resource demand, where it can take several hours to evaluate a single patient. An upside is that, given high-performance computing resources, multiple patients can be processed at the same time which makes it possible to analyse large datasets relatively quickly. Supervised deep-learning alternatives can be excellent options because of their substantially lower runtimes and ability to be trained to detect specific diseases, for example atrophy of the thalamus and multiple sclerosis [3, 4].

1.1 FreeSurfer

FreeSurfer is an open-source software suite for processing, visualising, and analysing MR images of the human brain. It includes a number of tools for performing tasks with the focus of volumetric segmentation of the brain into its various anatomical regions, as well as cortical reconstruction and analysis. FreeSurfer is developed by the Laboratories for Computational Neuroimaging at the Athinoula A. Martinos Center for Biomedical Imaging and is widely used in research, with over 60 000 licences given out in total and thousands of studies that reference FreeSurfer every year [5]. FreeSurfer is available for download at surfer.nmr.mgh.harvard.edu/.

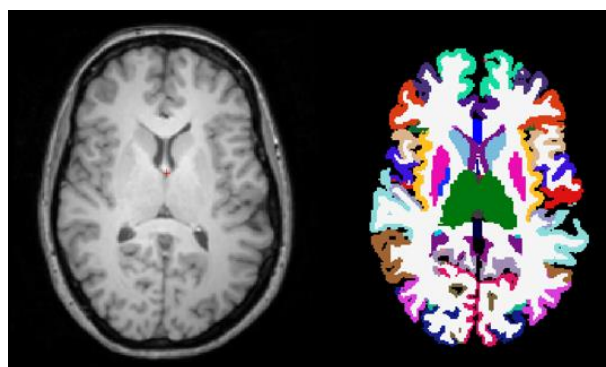


Figure 1. Figure showing 3D anatomical labels after a successful segmentation using FreeSurfer (right) together with an axial slice of the original 3D MR-image (left). Visualization is done using Freeview, a program included in the FreeSurfer toolkit.

FreeSurfer works by using software algorithms to automatically process neuroimaging data and extract information about brain anatomy and function. FreeSurfer's workflow includes the following steps: motion correction and averaging, removal of non-brain tissue, automatic Talairach transformation, segmentation of subcortical regions, intensity normalization, tessellation of the boundary between white and grey matter, automated topology correction, and delineation between grey and white matter as well as between grey matter and cerebrospinal fluid. Further cortical

processing and analysis can then be done, which includes surface inflation, spherical atlas registration, segmentation of the cerebral cortex, and an array of surface-based calculations such as sulcal depth, surface area and cortical thickness [6]. For the present work, the relevant capability is the generation of 3D labels for cortical and subcortical brain regions. After segmentation, the final labels are presented in a 3D image file where each voxel contains a categorical assignment instead of an intensity value as in a conventional MR image. This presentation is the same for the other 3D segmentation software packages used, FastSurfer and MAPER.

In addition to these tools for structural MRIs described above, FreeSurfer is also available for use with functional MRI (fMRI), diffusion MRI (dMRI) and Positron Emission Tomography (PET) [7].

By default, FreeSurfer label outputs are based on two atlases: The DKT (Desikan-Killiany-Tourville) atlas for cortical labels, and the ASEG (Automated Segmentation with Enhanced Generalization) atlas for subcortical labels. Together, they enable FreeSurfer to encompass and segment the whole brain [8, 9, 10]. During the period of the present project, a new FreeSurfer version implementing an alternative subcortical labelling approach (Sequence Adaptive Multimodal Segmentation; SAMSEG), but this was not used for the present work.

1.2 FastSurfer

FastSurfer is an open-source neuroimaging toolkit, based on FreeSurfer, for automated processing of MR images of the brain. It uses deep-learning principles to speed up the reconstruction process which enables it to generate FreeSurfer-compliant output in significantly less time. FastSurfer is made up of two main parts: FastSurferCNN and recon-surf. FastSurferCNN is the deep-learning component and is capable of segmenting a 3D brain MRI into 95 labels in under 1 minute. Recon-surf performs cortical surface reconstruction, thickness analysis and mapping of cortical labels. Recon-surf was not needed for this thesis, instead, the focus will be on FastSurferCNN.

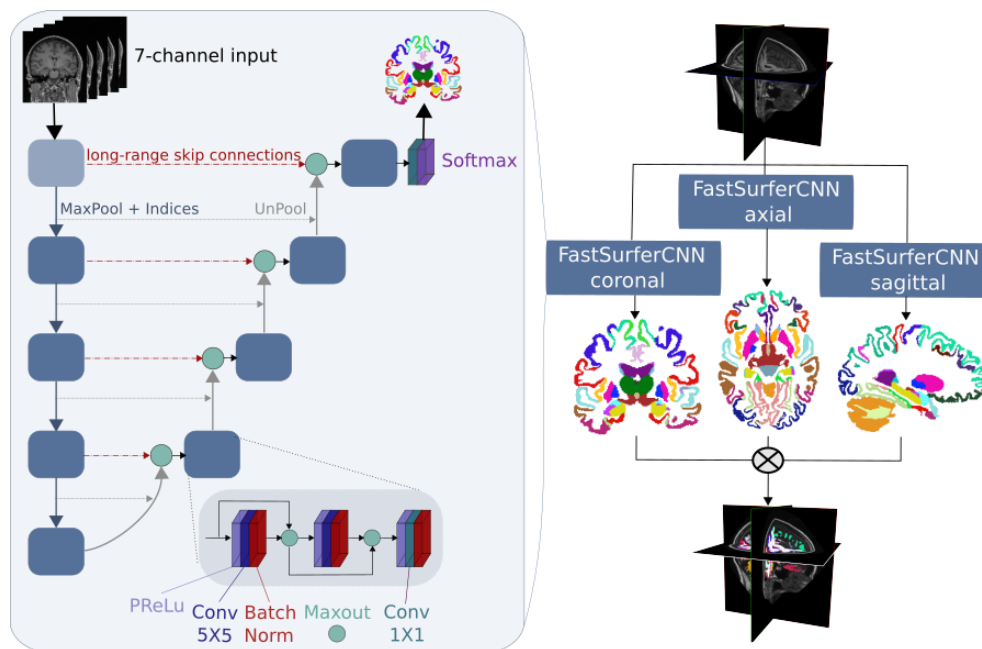


Figure 2. Schematic over FastSurferCNN. To the right is an overview of how the F-CNNs (Fully Convolutional Neural Networks) are summarised using view aggregation. The structure of each F-CNN can be seen on the left. Image source with permission from: deep-mi.org/research/fast-surfer/

FastSurferCNN consists of three fully convolutional neural networks (F-CNN), one for each 2D slice plane (sagittal, coronal, and axial). The three F-CNNs are then combined in a multi-slice view aggregation stage. Each F-CNN uses competitive dense blocks and competitive skip connections for both local and global competition. All this results in a network that is specifically made for accurate segmentation of both subcortical and cortical regions [3].

FastSurfer is developed by the Deep Medical Imaging Lab which is a cooperation of the German Center for Neurodegenerative Diseases and the Martinos Center for Biomedical Imaging. FastSurfer is available for download at github.com/Deep-MI/FastSurfer.

1.3 MAPER

MAPER (Multi-Atlas Propagation with Enhanced Registration) is a software tool for automated segmentation of the human brain. It is designed to produce accurate and consistent segmentations of the brain by using pre-segmented atlases. MAPER then uses advanced image registration techniques to propagate the labels from the atlases to the target image. For this, MAPER uses tools from the NiftySeg suite as well as tools from the Medical Image Registration Toolkit (MIRTK). The default atlas used by MAPER is the Hammers atlas, but MAPER can utilize any set of T1-weighted 3D MR images together with corresponding segmented image files, that adhere to MAPER's specifications, to make up a custom atlas.

MAPER is an open-source software developed by Heckemann et al. [11] and is available at github.com/soundray/maper. MAPER has been utilized in multiple studies with research topics such as Alzheimer's disease [12, 13, 14], the Thalamus [15], and Parkinson's disease [16].

1.4 Purpose

The purpose of this thesis was to use FreeSurfer as a baseline, and to investigate how well FastSurfer and MAPER segmentations conform to FreeSurfer's segmentations on the same dataset.

1.5 Structure of this thesis

This section is meant as a meta explanation of how the thesis is structured and why.

The purpose of this thesis was based on previous work done by Yaakub et al. [17], where FreeSurfer and MAPER were compared against each other, and found that they yielded similar results. The original purpose was to train FastSurfer using our own data and compare the accuracy of FastSurfer with MAPER. The training results ended up being unsatisfactory and therefore the purpose of the thesis was altered.

To only keep the results from the new purpose would not accurately describe what was tried and learned during the making of this thesis. An extra chapter, 4. Narrative account, was added to include a summary of the work that was put into training FastSurfer, the methods used, why the training failed, as well as the conclusions drawn.

2 Method

2.1 IXI Dataset

The IXI (Information eXtraction from Images) dataset is a collection of T1, T2, Proton Density, MRA, and diffusion-weighted images. The entire collection is made up of almost 600 participants, collected using three different MRI scanners: a Philips 3T system, a Philips 1.5T system and a GE 1.5T system [18].

Only T1-weighted images were selected because the methods used (FreeSurfer, FastSurfer and MAPER) are primarily designed for using T1 images as input. To control variations between the MR scanners being a confounder, only subjects from the 3T scanner were used. 185 subjects were utilized in total. The scan parameters used for the acquisition of the images in the IXI Dataset can be seen in Appendix 2 section 9.1.

2.2 Segmentation of the IXI dataset

Before segmentation was done, the dataset had to be pre-processed. The IXI dataset and MAPER share the same file format, compressed NIFTI (.nii.gz). FreeSurfer and FastSurfer do not use this file format and instead use the MGH/MGZ file format (.mgz). To change the file format from NIFTI to MGZ, the function `mri_convert` from the FreeSurfer suite was used. Examples of the code used, as well as how the folders were structured, can be seen in Appendix 1, section 8.1.

2.2.1 FreeSurfer segmentation

Processing of the IXI dataset with FreeSurfer was done using the `recon-all` function, the argument used was `-all`. The runtime per subject was approximately 10 hours and the code called can be seen in Appendix 1, section 8.2.

2.2.2 FastSurfer segmentation

Evaluation of the IXI dataset with FastSurfer was done using the `eval.py` python program. The default, pre-trained, checkpoints were used. The runtime per subject was approximately 1 minute and the code called can be seen in Appendix 1, section 8.3.

2.2.3 MAPER segmentation

Since the default atlas used by MAPER is incompatible with the atlases used by FreeSurfer and FastSurfer, the atlas for MAPER had to be changed. Out of the 185 IXI subjects, 40 were randomly selected to be used as a custom atlas. The subjects in this custom atlas will from here on be referred to as source subjects. The source subjects make up the custom atlas used by MAPER and includes the original 3D T1 images together with corresponding 3D label images generated by FreeSurfer. The 145 subjects not used in the custom atlas were segmented using the 40 source subjects, and each one out of the 40 source subjects were segmented using the other 39 source subjects.

To segment images using MAPER, the input images must be skull-stripped images of the brain. Skull-stripping is done by FreeSurfer as a part of its operation, these skull-stripped images could then be used as input images for MAPER. Two pre-processing functions had to be performed on the dataset images before running MAPER, `onepad` and `posnorm`. Where `onepad` uses a mask to pad the background of the MR image in the vicinity of the brain with the number 1, and `posnorm` performs positional normalization. `onepad` is called by MAPER on the fly, but positional normalization had to

be done using a program called `posnorm`, found at github.com/soundray/posnorm. The runtime per subject was approximately 2 hours and the code called for MAPER can be seen in Appendix 1, section 8.4.

2.3 Image postprocessing and result generation

2.3.1 Jaccard Similarity Coefficient

The Jaccard similarity Coefficient (JC), also known as the Jaccard index or the Jaccard score, is a measure of the similarity between two sets. It is defined as the size (cardinality) of the intersection of the sets divided by the size of the union of the sets, see Eq. 1.

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|} \quad (\text{Eq. 1})$$

The JC is a value between 0 and 1, where a value of 0 indicates that there is no similarity between the sets and a value of 1 indicates that the sets are identical [19]. The JC is widely used in many fields, such as, text analysis, natural language processing, image analysis, and bioinformatics.

2.3.2 Postprocessing

The program `mri_seg_overlap` from the FreeSurfer suite was used to measure JCs for each subject and label. JCs were measured for comparison between FreeSurfer and MAPER, and for comparison between FreeSurfer and FastSurfer. The output from `mri_seg_overlap` resulted in a .json-file for each subject that contained JCs for each label. A bash script was used to summarise all of the .json-files into one text file where each row in the text file contained a JC, a number corresponding to a specific label, a subject id, and which comparison it was from (FreeSurfer-FastSurfer or FreeSurfer-MAPER).

2.3.3 Generating results using R and label exclusion

After the results were summarised, a program was written in R to analyse the results. The R-program imported the text file with the results and started with sorting out unwanted labels. The unwanted labels were labels that were either not being segmented consistently and/or not segmented by all three programs for all subjects, see Table 1.

Table 1. Table containing all labels segmented by either FreeSurfer, FastSurfer or MAPER. Labels excluded from the results are marked with an asterisk. Labels numbered 2 to 255 are subcortical regions while labels numbered above 1000 are cortical regions. Label numbers and names taken from FreeSurfer's look-up table (FreeSurferColorLUT).

Label #	Label Name	Label #	Label Name	Label #	Label Name
2	Left-Cerebral-White-Matter	85	Optic-Chiasm*	2002	ctx-rh-caudalanteriorcingulate
4	Left-Lateral-Ventricle	251	CC_Posterior*	2003	ctx-rh-caudalmiddlefrontal
5	Left-Inf-Lat-Vent	252	CC_Mid_Posterior*	2005	ctx-rh-cuneus
7	Left-Cerebellum-White-Matter	253	CC_Central*	2006	ctx-rh-entorhinal
8	Left-Cerebellum-Cortex	254	CC_Mid_Anterior*	2007	ctx-rh-fusiform
10	Left-Thalamus	255	CC_Anterior*	2008	ctx-rh-inferiorparietal
11	Left-Caudate	1002	ctx-lh-caudalanteriorcingulate	2009	ctx-rh-inferiortemporal
12	Left-Putamen	1003	ctx-lh-caudalmiddlefrontal	2010	ctx-rh-isthmuscingulate
13	Left-Pallidum	1005	ctx-lh-cuneus	2011	ctx-rh-lateraloccipital
14	3rd-Ventricle	1006	ctx-lh-entorhinal	2012	ctx-rh-lateralorbitofrontal
15	4th-Ventricle	1007	ctx-lh-fusiform	2013	ctx-rh-lingual
16	Brain-Stem	1008	ctx-lh-inferiorparietal	2014	ctx-rh-medialorbitofrontal
17	Left-Hippocampus	1009	ctx-lh-inferiortemporal	2015	ctx-rh-middletemporal
18	Left-Amygdala	1010	ctx-lh-isthmuscingulate	2016	ctx-rh-parahippocampal
24	CSF	1011	ctx-lh-lateraloccipital	2017	ctx-rh-paracentral
26	Left-Accumbens-area	1012	ctx-lh-lateralorbitofrontal	2018	ctx-rh-parsopercularis
28	Left-VentralDC	1013	ctx-lh-lingual	2019	ctx-rh-parsorbitalis
30	Left-vessel*	1014	ctx-lh-medialorbitofrontal	2020	ctx-rh-parstriangularis
31	Left-choroid-plexus	1015	ctx-lh-middletemporal	2021	ctx-rh-pericalcarine
41	Right-Cerebral-White-Matter	1016	ctx-lh-parahippocampal	2022	ctx-rh-postcentral
43	Right-Lateral-Ventricle	1017	ctx-lh-paracentral	2023	ctx-rh-posteriorcingulate
44	Right-Inf-Lat-Vent	1018	ctx-lh-parsopercularis	2024	ctx-rh-precentral
46	Right-Cerebellum-White-Matter	1019	ctx-lh-parsorbitalis	2025	ctx-rh-precuneus
47	Right-Cerebellum-Cortex	1020	ctx-lh-parstriangularis	2026	ctx-rh-rostralanteriorcingulate
49	Right-Thalamus	1021	ctx-lh-pericalcarine	2027	ctx-rh-rostralmiddlefrontal
50	Right-Caudate	1022	ctx-lh-postcentral	2028	ctx-rh-superiorfrontal
51	Right-Putamen	1023	ctx-lh-posteriorcingulate	2029	ctx-rh-superiorparietal
52	Right-Pallidum	1024	ctx-lh-precentral	2030	ctx-rh-superiortemporal
53	Right-Hippocampus	1025	ctx-lh-precuneus	2031	ctx-rh-supramarginal
54	Right-Amygdala	1026	ctx-lh-rostralanteriorcingulate	2033	ctx-rh-temporalpole*
58	Right-Accumbens-area	1027	ctx-lh-rostralmiddlefrontal	2034	ctx-rh-transversetemporal
60	Right-VentralDC	1028	ctx-lh-superiorfrontal	2035	ctx-rh-insula
62	Right-vessel*	1029	ctx-lh-superiorparietal		
63	Right-choroid-plexus	1030	ctx-lh-superiortemporal		
72	5th-Ventricle*	1031	ctx-lh-supramarginal		
77	WM-hypointensities	1034	ctx-lh-transversetemporal		
80	non-WM-hypointensities*	1035	ctx-lh-insula		

An overall mean JC was calculated for each comparison based on all of the labels' JC. The data was then divided into two groups based on if the labels were either cortical or subcortical. A mean JC was then calculated for the cortical and subcortical labels. Violin plots were then generated using `ggplot2`, which is a part of the `Tidyverse` collection. The subjects used as source data were compared to the rest of the dataset by comparing the mean JC for the two sets.

To measure the volume discrepancy as a percentage between FastSurfer and MAPER, a variable named *diff* was defined and calculated for each label, see Eq. 2:

$$diff = 200 * \frac{JC_M - JC_F}{JC_M + JC_F} \quad (Eq. 2)$$

Where JC_M was the JC for a specific label in the FreeSurfer-MAPER comparison, and JC_F was the JC for same label for the FreeSurfer-FastSurfer comparison. A positive *diff*-value means that MAPER aligned more accurately with FreeSurfer than FastSurfer for that specific label. Conversely, a negative value means that FastSurfer aligned more with FreeSurfer.

2.3.4 Statistical analysis

A table containing the mean JC for each subject and comparison was generated in R. Welch's two sample t-test was then performed using the table, with the function `t.test`, comparing the results of the FreeSurfer-FastSurfer comparison with the FreeSurfer-MAPER comparison. The result is a p-value describing whether the difference in JCs between the two comparisons are statistically significant. This was done three times, for the cortical labels, subcortical labels, and all labels.

2.4 Hardware and programming languages

The data handling and computational work was enabled by resources provided by the National Academic Infrastructure for Supercomputing in Sweden (NAISS) and the Swedish National Infrastructure for Computing (SNIC) at Chalmers Centre for Computational Science (C3SE) partially funded by the Swedish Research Council through grant agreements no. 2022-06725 and no. 2018-05973.

The computational work was done on a supercomputer called Alvis which is hosted by C3SE. The use of Alvis is built around utilizing NVIDIA Graphical Processing Units (GPUs) to perform calculations significantly faster than what is possible on a conventional computer.

For calculations and processing, a NVIDIA T4 16GB RAM GPU was used for a majority of the work. The other specifications varied depending on which Alvis node was called but the CPU was either 2×8 core Intel(R) Xeon(R) Gold 6244 @ 3.60GHz or 2×16 core Intel(R) Xeon(R) Gold 6226R @ 2.90GHz, both with 768GB DDR4 RAM. Alvis is running Rocky Linux 8, which is an open-source version of Red Hat Enterprise Linux. The programming languages used was bash version 4.4.20 and R version 4.2.1.

The versions of the software used were FreeSurfer version 7.2.0, FastSurfer version 1.1.2, and MAPER version c6e6c21.

3 Results

3.1 Segmentation results

3.1.1 Overall Jaccard scores

Across all of the labels, FreeSurfer-FastSurfer had a mean JC with a standard deviation of 0.674 ± 0.121 and FreeSurfer-MAPER had a mean JC with a standard deviation of 0.689 ± 0.124 , see Figure 3. The difference in JC between the two comparisons is statistically significant ($p < 10^{-5}$).

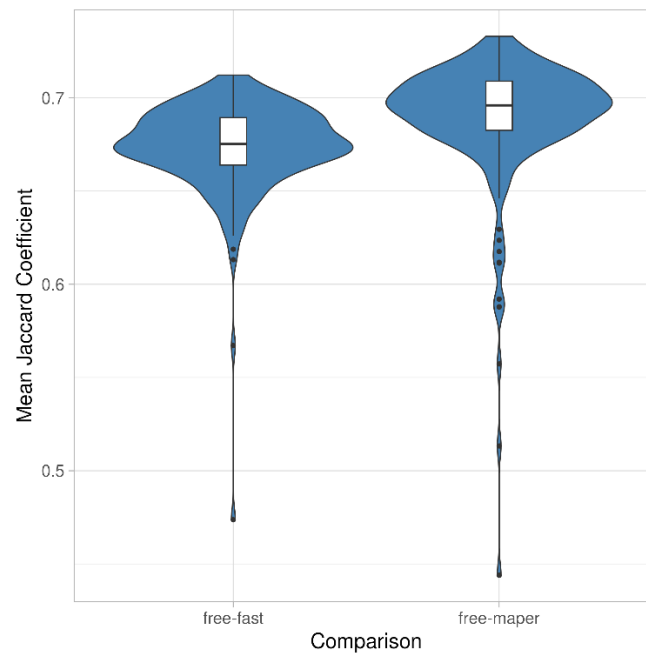


Figure 3. Violin plot showing the distribution of the mean JC for each subject by comparison, FreeSurfer-FastSurfer and FreeSurfer-MAPER. The plot includes all labels. The line in the boxplot is the median, the box edges are the 25th and 75th percentile, and the black dots are outliers.

3.1.2 Cortical Jaccard scores

For the cortical labels, FreeSurfer-FastSurfer had a mean JC with a standard deviation of 0.665 ± 0.083 and FreeSurfer-MAPER had a mean JC with a standard deviation of 0.665 ± 0.083 , see Figure 4. The difference in JC between the two comparisons is not statistically significant ($p = 0.982$).

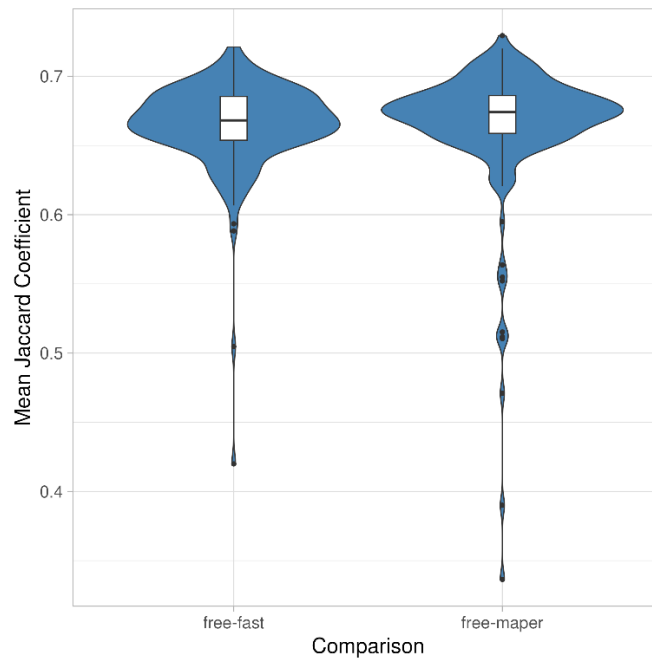


Figure 4. Violin plot showing the distribution of the mean JC for each subject by comparison, FreeSurfer-FastSurfer and FreeSurfer-MAPER. The plot only includes cortical labels. The line in the boxplot is the median, the box edges are the 25th and 75th percentile, and the black dots are outliers.

3.1.3 Subcortical Jaccard scores

For the subcortical labels, FreeSurfer-FastSurfer had a mean JC with a standard deviation of 0.689 ± 0.169 and FreeSurfer-MAPER had a mean JC with a standard deviation of 0.734 ± 0.165 , see Figure 5. The difference in JC between the two comparisons is statistically significant ($p < 10^{-53}$).

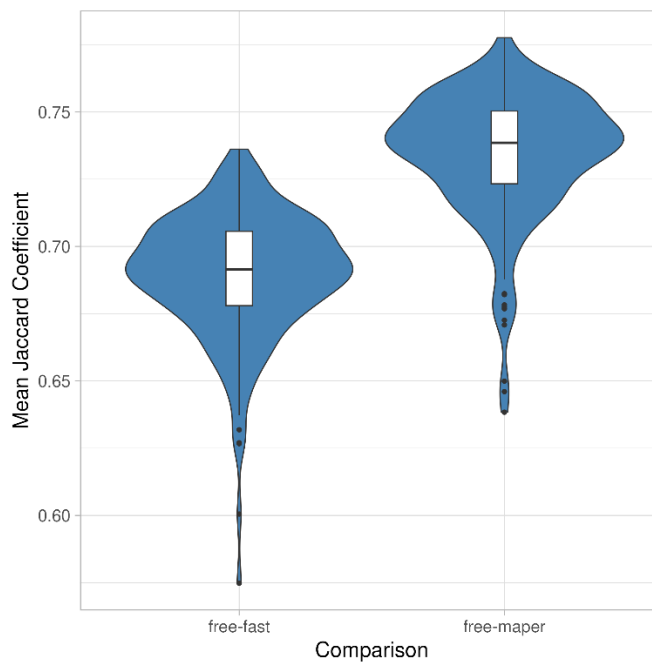


Figure 5. Violin plot showing the distribution of the mean JC for each subject by comparison, FreeSurfer-FastSurfer and FreeSurfer-MAPER. The plot only includes subcortical labels. The line in the boxplot is the median, the box edges are the 25th and 75th percentile, and the black dots are outliers.

3.2 Regional segmentation differences

Labels where MAPER aligned better with FreeSurfer compared to FastSurfer were primarily subcortical labels, with the Left and Right Accumbens area being the best performing, see Table 2. Other notable labels were the Left and Right Caudate as well as Cerebrospinal fluid. The only labels where FastSurfer had comparable *diff*-values to the labels mentioned for MAPER above were the Left and Right Inferior Lateral Ventricles.

Table 2. Table containing, for all labels, the mean JC, along with the standard deviation, and the *diff*-value, sorted by *diff*-values (descending).

Label #	Label Name	FreeSurfer-MAPER		FreeSurfer-FastSurfer		diff (%)
		Mean JC	SD	Mean JC	SD	
26	Left-Accumbens-area	0.583	0.087	0.353	0.164	49.1
58	Right-Accumbens-area	0.644	0.073	0.506	0.128	24.1
11	Left-Caudate	0.802	0.058	0.657	0.097	20.0
50	Right-Caudate	0.793	0.061	0.659	0.103	18.5
24	CSF	0.673	0.050	0.560	0.063	18.4
1010	ctx-lh-isthmuscingulate	0.682	0.083	0.574	0.059	17.2
2010	ctx-rh-isthmuscingulate	0.674	0.079	0.578	0.059	15.3
15	4th-Ventricle	0.816	0.036	0.716	0.052	13.1
14	3rd-Ventricle	0.801	0.043	0.710	0.103	12.0
43	Right-Lateral-Ventricle	0.866	0.038	0.770	0.053	11.7
4	Left-Lateral-Ventricle	0.865	0.040	0.783	0.065	10.0
63	Right-choroid-plexus	0.465	0.068	0.429	0.071	8.1
31	Left-choroid-plexus	0.461	0.064	0.427	0.080	7.6
2006	ctx-rh-entorhinal	0.551	0.107	0.511	0.088	7.6
28	Left-VentralDC	0.811	0.030	0.754	0.027	7.2
10	Left-Thalamus	0.863	0.022	0.804	0.029	7.0
1017	ctx-lh-paracentral	0.653	0.094	0.609	0.058	6.9
53	Right-Hippocampus	0.819	0.032	0.769	0.034	6.3
17	Left-Hippocampus	0.809	0.033	0.759	0.030	6.3
1006	ctx-lh-entorhinal	0.565	0.096	0.533	0.075	5.8
49	Right-Thalamus	0.865	0.024	0.816	0.029	5.8
60	Right-VentralDC	0.806	0.033	0.761	0.031	5.7
2013	ctx-rh-lingual	0.603	0.075	0.572	0.056	5.4
2035	ctx-rh-insula	0.798	0.064	0.757	0.047	5.3
18	Left-Amygdala	0.758	0.074	0.720	0.031	5.1
2017	ctx-rh-paracentral	0.657	0.095	0.625	0.075	5.0
46	Right-Cerebellum-White-Matter	0.717	0.062	0.682	0.062	5.0
7	Left-Cerebellum-White-Matter	0.720	0.060	0.687	0.058	4.6
2016	ctx-rh-parahippocampal	0.697	0.064	0.667	0.052	4.3
1035	ctx-lh-insula	0.802	0.065	0.771	0.051	3.9
1013	ctx-lh-lingual	0.601	0.069	0.579	0.052	3.7
54	Right-Amygdala	0.776	0.081	0.748	0.036	3.6
2026	ctx-rh-rostralanteriorcingulate	0.659	0.096	0.638	0.088	3.3
1012	ctx-lh-lateralorbitofrontal	0.734	0.052	0.713	0.041	3.0
2012	ctx-rh-lateralorbitofrontal	0.732	0.053	0.712	0.039	2.8
2022	ctx-rh-postcentral	0.670	0.097	0.654	0.079	2.4
2014	ctx-rh-medialorbitofrontal	0.666	0.057	0.651	0.047	2.3
16	Brain-Stem	0.895	0.017	0.875	0.018	2.2
1016	ctx-lh-parahippocampal	0.696	0.065	0.681	0.053	2.1
2024	ctx-rh-precentral	0.707	0.095	0.693	0.084	2.1
1014	ctx-lh-medialorbitofrontal	0.668	0.061	0.656	0.051	1.8
13	Left-Pallidum	0.708	0.086	0.696	0.085	1.7
1018	ctx-lh-parsopercularis	0.703	0.087	0.693	0.083	1.3
1026	ctx-lh-rostralanteriorcingulate	0.684	0.072	0.676	0.067	1.2
1022	ctx-lh-postcentral	0.678	0.083	0.671	0.048	1.1
1024	ctx-lh-precentral	0.714	0.080	0.707	0.053	0.9

1023	ctx-lh-posteriorcingulate	0.680	0.072	0.675	0.058	0.8
52	Right-Pallidum	0.721	0.092	0.716	0.087	0.7
47	Right-Cerebellum-Cortex	0.838	0.042	0.832	0.039	0.7
8	Left-Cerebellum-Cortex	0.834	0.044	0.830	0.040	0.4
2030	ctx-rh-superiortemporal	0.754	0.064	0.753	0.052	0.0
2021	ctx-rh-pericalcarine	0.520	0.100	0.520	0.059	-0.1
41	Right-Cerebral-White-Matter	0.886	0.026	0.886	0.021	-0.1
2023	ctx-rh-posteriorcingulate	0.679	0.074	0.680	0.068	-0.1
51	Right-Putamen	0.771	0.060	0.773	0.045	-0.2
2	Left-Cerebral-White-Matter	0.884	0.029	0.887	0.021	-0.3
1028	ctx-lh-superiorfrontal	0.753	0.055	0.755	0.045	-0.4
2028	ctx-rh-superiorfrontal	0.764	0.051	0.767	0.046	-0.4
1005	ctx-lh-cuneus	0.564	0.082	0.567	0.059	-0.5
2003	ctx-rh-caudalmiddlefrontal	0.675	0.085	0.679	0.078	-0.5
1030	ctx-lh-superiortemporal	0.738	0.063	0.742	0.056	-0.7
1002	ctx-lh-caudalanteriorcingulate	0.644	0.073	0.649	0.072	-0.7
12	Left-Putamen	0.770	0.054	0.776	0.048	-0.8
2018	ctx-rh-parsopercularis	0.673	0.080	0.680	0.076	-1.0
1034	ctx-lh-transversetemporal	0.629	0.077	0.636	0.066	-1.1
2002	ctx-rh-caudalanteriorcingulate	0.645	0.110	0.653	0.111	-1.3
2025	ctx-rh-precuneus	0.701	0.062	0.712	0.051	-1.5
2005	ctx-rh-cuneus	0.567	0.080	0.576	0.060	-1.6
1003	ctx-lh-caudalmiddlefrontal	0.681	0.078	0.694	0.070	-1.9
1020	ctx-lh-parstriangularis	0.647	0.079	0.659	0.075	-1.9
2007	ctx-rh-fusiform	0.656	0.068	0.669	0.060	-2.0
77	WM-hypointensities	0.247	0.068	0.252	0.060	-2.1
1019	ctx-lh-parsorbitalis	0.624	0.090	0.638	0.085	-2.2
1011	ctx-lh-lateraloccipital	0.639	0.062	0.654	0.056	-2.3
2031	ctx-rh-supramarginal	0.684	0.077	0.700	0.068	-2.4
1025	ctx-lh-precuneus	0.697	0.067	0.714	0.053	-2.5
1015	ctx-lh-middletemporal	0.692	0.064	0.711	0.075	-2.7
1031	ctx-lh-supramarginal	0.691	0.071	0.710	0.060	-2.8
2034	ctx-rh-transversetemporal	0.600	0.079	0.620	0.060	-3.2
1027	ctx-lh-rostralmiddlefrontal	0.683	0.060	0.707	0.058	-3.4
2019	ctx-rh-parsorbitalis	0.634	0.092	0.657	0.086	-3.6
2015	ctx-rh-middletemporal	0.726	0.054	0.753	0.053	-3.7
1009	ctx-lh-inferiortemporal	0.648	0.056	0.674	0.053	-3.9
1007	ctx-lh-fusiform	0.644	0.070	0.669	0.059	-3.9
2009	ctx-rh-inferiortemporal	0.652	0.058	0.678	0.054	-4.0
2020	ctx-rh-parstriangularis	0.637	0.087	0.666	0.067	-4.5
2027	ctx-rh-rostralmiddlefrontal	0.680	0.056	0.714	0.051	-4.8
1021	ctx-lh-pericalcarine	0.480	0.104	0.504	0.066	-4.9
2011	ctx-rh-lateraloccipital	0.643	0.057	0.678	0.051	-5.3
1008	ctx-lh-inferiorparietal	0.663	0.075	0.702	0.065	-5.7
2029	ctx-rh-superiorparietal	0.643	0.074	0.682	0.058	-5.8
1029	ctx-lh-superiorparietal	0.642	0.074	0.688	0.053	-7.0
2008	ctx-rh-inferiorparietal	0.680	0.062	0.731	0.055	-7.2
44	Right-Inf-Lat-Vent	0.519	0.170	0.615	0.159	-17.0
5	Left-Inf-Lat-Vent	0.444	0.125	0.542	0.125	-19.8

3.3 MAPER source subjects

The subjects used as source data for MAPER showed minimal differences compared to the rest of the dataset, see Table 3.

Table 3. Mean JC for subjects used as source data for MAPER compared to the rest of the dataset. Comparison between FreeSurfer and FastSurfer, and FreeSurfer and MAPER.

	<i>FreeSurfer-FastSurfer</i> <i>JC</i>	<i>FreeSurfer-MAPER</i> <i>JC</i>
<i>Source subjects</i>	0.6734	0.6894
<i>Other subjects</i>	0.6736	0.6892

4 Narrative account

This section contains a narrative account of work that was done as part of the original purpose of the thesis, as explained in section 1.5.

4.1 FastSurferCNN training

4.1.1 Training dataset

To train FastSurfer, 30 participants from the Hammers atlas database were used. The Hammers atlas database contains 3D T1-weighted MRI brain scans of 30 adult subjects, each with a manually segmented label file [20]. The scan parameters used for the acquisition of the images used in the Hammer atlas database can be seen in Appendix 2, section 9.2. The subjects were equally split into two groups: one for training and one for validation. The file structure was set up like this:

Training data:

```
$hm30/trn/a1/  
$hm30/trn/a2/  
$hm30/trn/a3/  
..  
$hm30/trn/a15/
```

Validation data:

```
$hm30/val/a16/  
$hm30/val/a17/  
$hm30/val/a18/  
..  
$hm30/val/a30/
```

Where `$hm30` is the folder containing the Hammers atlas dataset, and `a1` to `a30` are the subjects.

4.1.2 Training

Training was done following FastSurfer's instructions on their github [21].

The dataset had to be converted to hdf5 using FastSurfer's `generate_hdf5` function. Two csv-files were used to specify the subjects involved, one file for training subjects and one for validation subjects. The label files from the Hammers atlas were used as ground truth for the training. A program called Freeview, which is included in the FreeSurfer suite, was used to remove the Corpus Callosum from the label images for the `gt_noCC` argument for the function `generate_hdf5`. Hdf5-data needed to be generated for each slice direction (coronal, sagittal and axial) as well as for training and validation data. An example of the code used can be seen in Appendix 1, section 8.5.1.

After the hdf5-dataset was generated, it was used to train FastSurferCNN using the `train.py` function, once for every slice direction (coronal, sagittal and axial). The default arguments were used, example code can be seen in Appendix 1, section 8.5.1.

The `train.py` function generates a checkpoint file containing the result of the training. The checkpoint files can then be called when segmenting brain images with FastSurfer, as done earlier in section 2.3.2. Training was then repeated using FreeSurfer outputs from the Hammers data as ground truth, instead of the manually segmented label images.

A T1-weighted 3D MR brain image from FreeSurfer's wiki-page (surfer.nmr.mgh.harvard.edu/fswiki) was used as a test image and was segmented three times with FastSurfer using different checkpoints: the default checkpoints, checkpoints trained from the Hammers data as explained above, and checkpoints trained using FreeSurfer outputs of the Hammers data as the label files.

Due to unsatisfactory training results, see section 4.2, contact was made with the FastSurfer creators to ask for feedback. The culmination of the feedback was to repeat the training using a different branch of FastSurferCNN called FastSurferVINN. Due to a lack of up-to-date modules on Alvis at the time, FastSurferVINN could not be used with the correct package versions and could not be executed properly. Because of these issues, the purpose of the thesis was shifted to the new purpose as stated in the introduction.

4.2 Training results

Training results were unsatisfactory, see Figure 6. Segmentation using checkpoints generated from FreeSurfer outputs of the Hammers atlas performed better than using the Hammers atlas with manual segmentations.

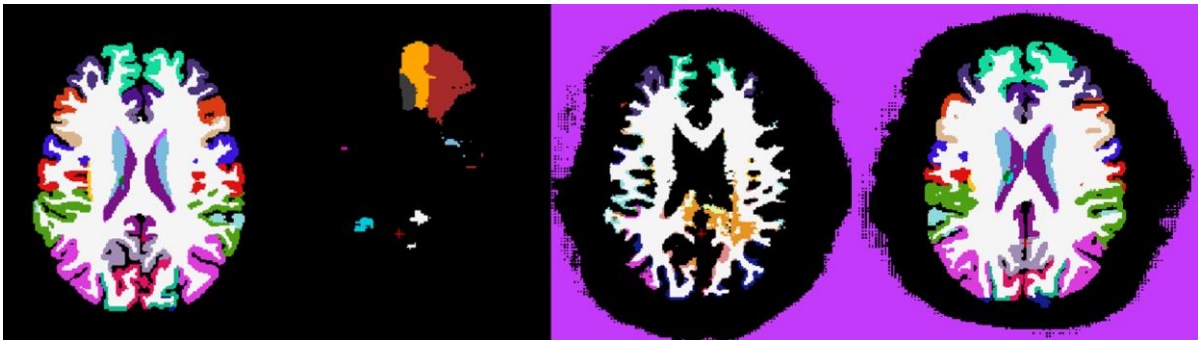


Figure 6. Figure showing FastSurfer segmentation using different checkpoints of a test image. Segmentation using default checkpoints to the left (control), checkpoints from training using Hammers atlas data in the middle-left, and checkpoints from training using FreeSurfer outputs of the Hammers atlas to the middle-right. Segmentation of one of the images used for validation (a17), using checkpoints from training using FreeSurfer outputs of the Hammers atlas, can be seen on the right.

4.3 Training discussion

The reason for the failure of the FastSurfer training is multifaceted but most likely stems from at least one of two things; the training data differed too much from FastSurfer's specifications, and/or there were not enough subjects for training and validation.

The default commands were used when training FastSurfer which stated using 15 subjects for training and 15 subjects for validation. FastSurfer itself was trained using 140 images for training and 20 for validation, which was a lot more than was expected [3]. Our findings highlight that 30 subjects are not enough for training and validation, for the dataset used.

Training FastSurferCNN using the manually labelled images from the Hammers atlas as "ground truth" failed because the label images did not follow the same standard as FreeSurfer/FastSurfer. The ground truth data was then changed to using FreeSurfer outputs from the Hammers atlas as ground truth for the training. FreeSurfer generated all of the files needed for training and while the training went better, there were still issues.

In the previous paragraph the term "ground truth" is used because it was what was used by the FastSurfer creators. It is important to note that the "ground truth" used for training was not an objective ground truth as the term implies, but simply data used for training. Whether we are talking about the manual Hammers atlas segmentations or the FreeSurfer label outputs, there is always some uncertainty or error associated with each procedure. The term "ground truth" ignores this uncertainty and gives the user a false confidence in the training data used, and by extension, the model's output.

Another minor issue could have been that the test image used was too dissimilar compared to the training data. FastSurfer was therefore tested using an image that was not representative of the training dataset. To investigate this, segmentation was repeated using one of the validation images. The results were better compared to earlier results but still not adequate (Figure 6, right), which demonstrated that the test image was not the problem by itself.

Contact was made with the FastSurfer creators through their Github. After troubleshooting, theorising, and concluding in that the reasons listed above were most likely why the training failed. They suggested to try an experimental branch of FastSurfer called FastSurferVINN. Unfortunately, FastSurferVINN required updated packages that were not yet available on Alvis, and we were unable to proceed.

FastSurfer has since been updated and FastSurferVINN is now the default neural network, instead of FastSurferCNN. More work is needed to investigate if using FastSurferVINN would have made a difference to the training, if it was the Hammers atlas MR images that were too different from what FastSurfer expected, a larger training corpus was needed, or if there was any other reason as to why the training failed.

An error was also made when selecting the subjects for either training or validation, there was no randomness in the selection. The first 15 subjects were used for training and the remaining 15 were used for validation. If done again, subjects should be chosen randomly for either training or validation to minimize subject order being a confounder. If the subjects were ordered in a certain way in the atlas, the selection could have led to subjects that share a certain trait being put in the same group, for example training, and with no subjects sharing that trait in the validation group. This would lead to training on a dataset that is not as diverse as it could have been, and therefore leading to subpar training results. This likely was not the cause of the training failure but was bad practice in general.

5 Discussion

MAPER outperformed FastSurfer at generating FreeSurfer-outputs slightly overall. This was due to MAPER's superior performance for the subcortical labels while the results were almost identical for the cortical labels. The results were statistically significant across all labels as well as for the subcortical labels ($p < 10^{-5}$ and $p < 10^{-53}$, respectively). The difference in JC was not statistically significant when comparing the cortical labels, where the two programs scored similarly ($p < 0.975$).

Since MAPER outperformed FastSurfer for the subcortical labels, it is not a surprise that 4 out of 5 labels with the highest *diff*-values were subcortical. Further investigation is needed to examine the differences in segmentation of MAPER and FastSurfer on a label-by-label basis and why they differ for labels with higher (or lower) *diff*-values.

It is important to note that MAPER and FastSurfer were compared to FreeSurfer and not an objective ground truth. We only examined how the two programs compares at replicating FreeSurfer outputs, not an actual ground truth.

The subjects used as source data showed minimal differences in JC compared to the rest of the dataset. If MAPER would have favoured the source data, a higher JC would be seen for those subjects. Although an increase was seen, it was minimal and not significant.

A major advantage of FastSurfer was its segmentation speed compared to MAPER. Each subject took around a minute to segment using FastSurfer and around 2 hours using MAPER. In a clinical situation this could weigh strongly in favour of FastSurfer, as the faster segmentation speed means that it would be easier to integrate FastSurfer into the clinical workflow without slowing down the overall speed. It also means that FastSurfer is substantially quicker at handling larger datasets compared to MAPER.

One disadvantage FastSurfer had was that it was not trained on the same type of data as MAPER, which could have brought down FastSurfers JCs. FastSurfer was used with its default checkpoints whereas MAPER could use the FreeSurfer outputs directly as source data. The next step would be to investigate how much the results would differ if FastSurfer was trained on an external atlas like the IXI atlas.

Similarly, a disadvantage for MAPER was that a non-native atlas was used. It would be interesting to see if the results would differ depending on the atlas used. Comparisons have shown that FreeSurfer and MAPER perform better using their native atlases, the question is to what extent it affects FastSurfer [17].

More studies are needed on FastSurfer in general as there are currently very few published studies out there. For example, there were no studies on PubMed when this thesis was started (January 2022), excluding studies from contributors involved with FastSurfer. FastSurfer is still in its infancy and is constantly being updated, it will be interesting to see what the future holds for the project.

It is important to note that results from software algorithms can change with the version of the program. For example, results from FreeSurfer have been shown to vary slightly between versions [22]. New research is continuously needed to make sure software performance does not decline, and that results are as good as possible.

6 Conclusion

This thesis shows that MAPER is more accurate at replicating FreeSurfer-conform segmentations, compared to FastSurfer, for a majority of the subcortical labels of the brain, most notably the left and right accumbens area, left and right caudate, and CSF. The results of the comparison also show that MAPER and FastSurfer perform equally well, on average, for the cortical labels of the brain.

Acknowledgements

I wish to thank Rolf A. Heckemann for giving me the opportunity to work on this project, as well as for guiding and helping me during this prolonged period. I also wish to thank my fiancé, Ting, without your support I might not have finished this thesis at all.

7 References

- [1] M. Aboian, K. Bousabarah, E. Kazarian, T. Zeevi, W. Holler, S. Merkaj, G. Cassinelli Petersen, R. Bahar, H. Subramanian, P. Sunku, E. Schrickel, J. Bhawnani, M. Zawalich, A. Mahajan, A. Malhotra, S. Payabvash, I. Tocino, L. MingDe och M. Westerhoff, "Clinical implementation of artificial intelligence in neuroradiology with development of a novel workflow-efficient picture archiving and communication system-based automated brain tumor segmentation and radiomic feature extraction," *Frontiers in Neuroscience*, vol. 16, p. 860208, 2022.
- [2] M. Casati, S. Piffer, S. Calusi, L. Marrazzo, G. Simontacchi, V. Di Cataldo, D. Greto, I. Desideri, M. Vernaleone, G. Francolini, L. Livi och S. Pallotta, "Clinical validation of an automatic atlas-based segmentation tool for male pelvis CT images," *Journal of Applied Clinical Medical Physics*, vol. 23, p. e13507, 2022.
- [3] L. Henschel, S. Conjeti, S. Estrada, K. Diers, B. Fischl och M. Reuter, "FastSurfer - A fast and accurate deep learning based neuroimaging pipeline," *NeuroImage*, vol. 219, p. 117012, 2020.
- [4] J. Burggraaff, Y. Liu, J. C. Prieto, J. Simoes, A. de Sitter, S. Ruggieri, I. Brouwer, B. I. Lissenberg-Witte, M. A. Rocca, P. Valsasina, S. Ropele, C. Gasperini, A. Gallo, D. Pareto, J. Sastre-Garriga, C. Enzinger, M. Filippi, N. De Stefano, O. Ciccarelli, H. E. Hulst och C. R. Guttmann, "Manual and automated tissue segmentation confirm the impact of thalamus atrophy on cognition in multiple sclerosis: A multicenter study," *Neuroimage: Clinical*, vol. 29, p. 102549, 2021.
- [5] "FreeSurferWiki - FreeSurfer stats," [Online]. Available: <https://surfer.nmr.mgh.harvard.edu/fswiki/FsStats>. [Accessed 15 02 2023].
- [6] B. Fischl, "FreeSurfer," *Neuroimage*, vol. 62, pp. 774-781, 2012.
- [7] "FreeSurferWiki," [Online]. Available: <https://surfer.nmr.mgh.harvard.edu/fswiki/FreeSurferWiki>. [Accessed 15 02 2023].
- [8] B. Fischl, A. van der Kouwe, C. Destrieux, E. Halgren, F. Ségonne, D. H. Salat, E. Busa, L. J. Seidman, J. Goldstein, D. Kennedy, V. Caviness, N. Makris, B. Rosen och A. M. Dale, "Automatically parcellating the human cerebral cortex," *Cerebral Cortex*, vol. 14, pp. 11-22, 2004.
- [9] A. Klein och J. Tourville, "101 labeled brain images and a consistent human cortical labeling protocol," *Frontiers in Neuroscience*, vol. 6, p. 171, 2012.
- [10] B. Fischl, D. H. Salat, E. Busa, M. Albert, M. Dieterich, C. Haselgrove, A. van der Kouwe, R. Killiany, D. Kennedy, S. Klaveness, A. Montillo, N. Makris, B. Rosen och A. M. Dale, "Whole brain segmentation: automated labeling of neuroanatomical structures in the human brain," *Neuron*, vol. 33, pp. 341-355, 2002.
- [11] R. A. Heckemann, S. Keihaninejad, P. Aljabar, D. Rueckert, J. V. Hajnal och A. Hammers, "Improving intersubject image registration using tissue-class information benefits robustness and accuracy of multi-atlas based anatomical segmentation," *NeuroImage*, vol. 51, pp. 221-227, 2010.

- [12] L.-A. Sapey-Triomphe, R. A. Heckemann, N. Boublay, J.-M. Dorey, M.-A. Hénaff, I. Rouch, C. Padovan, A. Hammers och P. Krolak-Salmon, "Neuroanatomical correlates of recognizing face expressions in mild stages of Alzheimer's disease," *Alzheimer's Disease Neuroimaging Initiative*, vol. 10, p. e0143586, 2015.
- [13] R. A. Heckemann, S. Keihaninejad, P. Aljabar, K. R. Gray, C. Nielsen, D. Rueckert, J. V. Hajnal och A. Hammers, "Automatic morphometry in Alzheimer's disease and mild cognitive impairment," *The Alzheimer's Disease Neuroimaging Initiative*, vol. 56, nr 4, pp. 2024-2037, 2011.
- [14] Y. Klein-Koerkamp, R. A. Heckemann, K. T. Ramdeen, O. Moreaud, S. Keignart, A. Krainik, A. Hammers, M. Baciuc och P. Hot, "Amygdalar atrophy in early Alzheimer's disease," *Alzheimer's Disease Neuroimaging Initiative*, vol. 11, pp. 239-252, 2014.
- [15] C. Traynor, R. A. Heckemann, A. Hammers, J. O'Muircheartaigh, W. R. Crum, G. J. Barker och M. P. Richardson, "Reproducibility of thalamic segmentation based on probabilistic tractography," *NeuroImage*, vol. 52, pp. 69-85, 2010.
- [16] S. Prange, E. Metereau, A. Maillet, E. Lhommée, H. Klinger, P. Pelissier, D. Ibarrola, R. A. Heckemann, A. Castrioto, L. Tremblay, V. Sgambato, E. Broussolle, P. Krack och S. Thobois, "Early limbic microstructural alterations in apathy and depression in de novo Parkinson's disease," *Movement Disorders*, vol. 34, pp. 1644-1654, 2019.
- [17] S. Nurbaya Yaakub, R. A. Heckemann, S. S. Keller, C. J. McGinnity, B. Weber och A. Hammers, "On brain atlas choice and automatic segmentation methods: a comparison of MAPER & FreeSurfer using three atlas databases," *Scientific Reports*, vol. 10, p. 2837, 2020.
- [18] "IXI Dataset - Brain Development," [Online]. Available: <http://brain-development.org/ixi-dataset/>. [Accessed 17 11 2022].
- [19] P. Jaccard, "The distribution of the flora in the Alpine zone," *New Phytologist*, vol. 11, pp. 37-50, 1912.
- [20] "Adult Brain Atlases - Hammers adult atlases," [Online]. Available: <http://brain-development.org/brain-atlases/adult-brain-atlases/>. [Accessed 02 03 2022].
- [21] "FastSurfer/FastSurferCNN," [Online]. Available: <https://github.com/DeepMI/FastSurfer/tree/stable/FastSurferCNN>. [Accessed 22 09 2022].
- [22] P. Filip, P. Bednarik, L. E. Eberly, A. Moheet, A. Svatkova, H. Grohn, A. F. Kumar, E. R. Seaquist och S. Mangia, "Different FreeSurfer versions might generate different statistical outcomes in case-control comparison studies," *Neuroradiology*, vol. 64, pp. 765-773, 2022.
- [23] "Scanner: Philips Medical Systems Intera 3T - Brain Development," [Online]. Available: <http://brain-development.org/scanner-philips-medical-systems-intera-3t/>. [Accessed 6 5 2023].
- [24] A. Hammers, R. Allom, M. J. Koeppe, S. L. Free, R. Myers, L. Lemieux, T. N. Mitchell, D. J. Brooks och J. S. Duncan, "Three-dimensional maximum probability atlas of the human brain, with particular reference to the temporal lobe," *Human Brain Mapping*, vol. 19, pp. 224-247, 2003.

8 Appendix 1

8.1 Folder structure and change of file format

The dataset was organized according to the following file-structure:

```
$ixi/c012/mri/  
$ixi/c013/mri/  
$ixi/c014/mri/  
$ixi/c015/mri/  
..  
$ixi/c661/mri/
```

Where `c012` to `c661` are the different subjects in the dataset and `$ixi` is the path to the folder containing all the subjects. The IXI dataset images were inserted into the `mri/` folder. The MR images were renamed to `orig.nii.gz`.

The function `mri_convert` was used to change the filetype from `.mgz` to `.nii.gz`. `mri_convert` is included in FreeSurfer and to initiate FreeSurfer the following code was used:

```
cd $freesurfer  
export FREESURFER_HOME=$PWD  
source SetUpFreeSurfer.sh
```

Where `$freesurfer` is the path to the folder containing FreeSurfer. `mri_convert` was then called with the following code:

```
cd $ixi  
realpath c* | xargs -I {} mri_convert {}/mri/orig.mgz  
{} /mri/orig.nii.gz
```

This retrieves the file path for each subject, uses the file path as an input for `xargs`, which then calls `mri_convert` for each subject.

8.2 FreeSurfer segmentation code

The code called for FreeSurfer segmentation for each subject:

```
recon-all -all -I $id/mri/orig.mgz -s $id-free/
```

Where `$id` was the name of each subject. The output file used for the comparison for each subject was `aparc.DKTatlas+aseg.mgz`.

8.3 FastSurfer segmentation code

The code called for FastSurfer segmentation for each subject:

```
python FastSurferCNN/eval.py --i_dir $ixi --o_dir $ixi-fast --  
network_sagittal_path  
checkpoints/Sagittal_Weights_FastSurferCNN/ckpts/Epoch_30_training_s  
tate.pkl --network_coronal_path  
checkpoints/Coronal_Weights_FastSurferCNN/ckpts/Epoch_30_training_st  
ate.pkl --network_axial_path  
checkpoints/Axial_Weights_FastSurferCNN/ckpts/Epoch_30_training_stat  
e.pkl
```

Where `$ixi-fast` is a new folder containing the results of the FastSurfer segmentation. The files named `Epoch_30_training_state.pkl` are the checkpoint files used, one for each slice direction (sagittal, axial, and coronal). In this example the default checkpoints were used. The output file used for the comparison for each subject was `aparc.DKTatlas+aseg.deep.mgz`.

8.4 MAPER segmentation code

Example code called to determine the rigid transformation that positionally normalizes IXI image of subject c012:

```
$SOFTWARE/posnorm/posnorm.sh -img  
$IXI/fs/subjects/c012/mri/brainmask.nii.gz -mask  
$IXI/fs/subjects/c012/mri/brainmask.nii.gz -dofout  
$IXI/fs/posnorm/c012.dof.gz
```

MAPER was then called using the following code: Example code called for the pairing of IXI image of subject c015 as source and IXI image of subject c012 as target:

```
$SOFTWARE/maper/maper -tgtid c012 -tgtmri  
$IXI/fs/subjects/c012/mri/brainmask.nii.gz -tpn  
$IXI/fs/posnorm/c012.dof.gz -tgtlabels  
aseg:$IXI/fs/subjects/c012/mri/aparc.DKTatlas+aseg.nii.gz -atlasn 40  
-output-dir $IXI/fsmaper -threads 32 -srcid c015 -srcmri  
$IXI/fs/subjects/c015/mri/brainmask.nii.gz -spn  
$IXI/fs/posnorm/c015.dof.gz -srclabels  
aseg:$IXI/fs/subjects/c015/mri/aparc.DKTatlas+aseg.nii.gz -srccache  
$IXI/fsmaper/cache -tgtcache $IXI/fsmaper/cache
```

8.5 Training FastSurferCNN

8.5.1 Hdf5 generation

To generate a hdf5 dataset of the Hammers atlas, an example of the code used can be seen below for the axial slice direction and training subjects:

```
python3 FastSurferCNN/generate_hdf5.py --hdf5_name
$training_data/training_set_cispa_axial.hdf5 --csv_file csv_trn.csv
--plane axial --image_name mri/orig.mgz --gt_name mri/aseg.mgz --
gt_nocc mri/aseg_nocc_c.mgz
```

Where `$training_data` is the folder containing the training data, `orig.mgz` is the MR image of the brain, `aseg.mgz` is the label file containing the manually segmented brain, and `aseg_nocc.mgz` is a copy of the manually segmented label file but with the Corpus Callosum removed.

8.5.2 Training

To train FastSurferCNN using the previously generated hdf5-dataset the `train.py` function was called three times, once for each slice direction (sagittal, axial, and coronal). The following code was used for the sagittal plane direction:

```
python3 FastSurferCNN/train.py --hdf5_name_train
$training_data/training_set_cispa_sagittal.hdf5 --hdf5_name_val
$training_data/validation_set_cispa_sagittal.hdf5 --plane sagittal -
-log_dir checkpoints/Sagittal_Competitive_APARC_ASEG/ --epochs 30 --
num_channels 7 --num_classes 51 --batch_size 15 --
validation_batch_size 15
```

Where a checkpoint directory is created, with a name chosen with the `-log_dir` argument.

9 Appendix 2

9.1 Scan parameters for the IXI dataset

- Scanner: Philips Medical Systems Intera 3T
- Repetition Time = 9.6
- Echo Time = 4.6
- Number of Phase Encoding Steps = 208
- Echo Train Length = 208
- Reconstruction Diameter = 240.0
- Acquisition Matrix = 208 x 208
- Flip Angle = 8.0 [23]

9.2 Scan parameters for the Hammer's atlas dataset

MRI scans were conducted at the National Society for Epilepsy using a 1.5 Tesla GE Signa Echospeed scanner. A coronal T1 weighted 3D volume was acquired using a GE inversion recovery prepared fast spoiled gradient recall sequence. The TE/TR/NEX parameters were set to 4.2 msec (fat and water in phase)/15.5 msec/1, respectively. A time of inversion (TI) of 450 msec and a flip angle of 20° were applied to obtain 124 slices with a thickness of 1.5 mm. The field of view was 18×24 cm with a 192×256 matrix, providing complete coverage of the entire brain with voxel sizes of 0.9375 × 0.9375 × 1.5 mm. In order to maintain the native resolution, the images were resliced using windowed sinc interpolation, resulting in isotropic voxels measuring 0.9375 × 0.9375 × 0.9375 mm³ [24].