Automatic Segmentation of Low-Field MRI Brain Scans by Integrating Analytic and Deep Learning Techniques

M. C. Zijta

TU Delft

Automatic Segmentation of Low-Field MRI Brain Scans by Integrating Analytic and Deep Learning Techniques

by

Marcella Cynthia Zijta

to obtain the degrees of
Master of Science
in Applied Mathematics
and
Master of Science
in Biomedical Engineering
at the Delft University of Technology,
to be defended publicly on Wednesday December 8th, 2021 at 2:00 PM

Student number:	4383257	
Project duration:	December 14, 2020 - December 8, 2021	
Thesis committee AM:	Prof. Dr. M. van Gijzen,	TU Delft, chair of thesis committee
	Dr. F. Vos,	TU Delft, supervisor
	Dr. Y. van Gennip	TU Delft
	Ir. M. de Leeuw den Bouter,	TU Delft, supervisor
Thesis committee BME:	Dr. F. Vos,	TU Delft, chair of thesis committee
	Prof. Dr. M. van Gijzen,	TU Delft, supervisor
	Dr. Y. van Gennip	TU Delft
	Prof. Dr. B. Rieger,	TU Delft

An electronic version of this thesis is available at `http://repository.tudelft.nl/`.

Preface

This thesis project is a graduation project for two Masters: Applied Mathematics and Biomedical Engineering. The topic of automatic segmentation of MRI scans is relevant for both Masters. However, this thesis project has clear distinguishable parts that each corresponds to one of the Masters. In the first part of this thesis report mathematical experiments are set out. In the second part of this thesis the objective of the Biomedical Engineering project is discussed. This is why, this report has a different structure as other theses. The first part consists of chapters about research into related work, methods and results. This structure is echoed in the second part. It was chosen to not separate these two parts into two reports, since conclusions drawn from the first part instigate research of the second part. Due to the project parts being interwoven, introduction, discussion and conclusions are considered for both parts jointly.

Abstract

Hydrocephalus is a disease where an excess of cerebrospinal fluid (CSF) is built up in the brain. It affects approximately 180.000 infants per year in sub-Saharan Africa. Magnetic resonance imaging (MRI) is an advantageous imaging method to diagnose hydrocephalus and examine the amount of fluid in the brain for treatment. Unfortunately, in sub-Saharan Africa there is limited access to MRI scanners. That is why an inexpensive, portable, low-field MRI scanner is being built for the treatment of hydrocephalus in Uganda. One main restriction of this scanner is simplicity in use. Therefore, the goal is to make software for the MRI scanner that automatically makes and processes the scans. Part of the processing is the automatic segmentation of the scan into CSF and brain tissue regions. Automatic segmentation is complex due to noise and artifacts present in low-field scans. Also, automation of segmentation processes is complicated. Therefore, in this thesis project we aimed to realize the foundations for a fast, practical and automatic 3D segmentation method for brain scans obtained by the low-field MRI scanner. First, analytic segmentation methods were investigated. Multiple segmentation methods were applied to different high-field and low-field scans. Data analysis showed that Li's method, where the intensity non-homogeneity artifact was corrected for during segmentation, improved the segmentation results evidently when the scan was affected by a distinct bias field. The major disadvantage of Li's method was the number of parameters and initialization values that had to be chosen. Therefore, it would be complex to satisfy the automation requirement by using an analytic segmentation method. However, this led to the idea of integrating Li's method with a neural network. A neural network would solve the problem of automation, while the incorporation of Li's method, would lead to the fitting energy of the segmentation being minimized, which could improve the segmentation results. A neural network for the segmentation of CSF, white matter and gray matter and the prediction of the bias field was built. Unfortunately, a low-field dataset was not available to train on. Therefore, for the training of the network, artificial high-field data was used, together with its ground truth segmentations. Then, the network was trained by not only comparing the predictions with the ground truth segmentations and bias field, but also by minimizing the fitting energy of the predicted segmentation. To evaluate the segmentation results Dice scores were computed between the ground truth and the predicted segmentations. The Dice scores of the train and test set showed that the segmentation results of the neural network improved when the analytic segmentation loss was added to the network. To further investigate the promising effect of the analytic segmentation loss, the trained network was transferred to a new dataset containing infant brain scans, for which the ground truth segmentations were ignored. The network was therefore trained on this new dataset by only using the analytic segmentation loss function. Unfortunately, the results of segmentation became worse. This occurred due to non-brain tissues wrongly being segmented in the brain tissue clusters, since the signal intensities were close to each other. After the non-brain tissues were removed by brain masking, the results of segmentation for all tissues had improved. When more low-field brain scans are available the neural network should be transferred to these scans, since automatic segmentation of low-field scans is the final objective. It is recommended to first implement an automatic brain masking technique on the scans for optimal results. The promising results of Li's method applied to the low-field scans and of transferring the neural network to the infant dataset show excellent future perspective for the fast, practical and automatic 3D segmentation of low-field scans.

Nomenclature

2D	Two-Dimensional
3D	Three-Dimensional
CNN	Convolutional Neural Network
CSF	Cerebrospinal Fluid
CT	Computed Tomography
FCN	Fully Convolutional Network
GM	Gray Matter
ICV	Intracranial Volume
MRF	Magnetic Resonance Fingerprinting
MRI	Magnetic Resonance Imaging
MSE	Mean Squared Error
PD	Proton Density
PVA	Partial Volume Averaging
RF	Radio Frequency
SAR	Specific Absorption Rate
SNR	Signal-to-Noise Ratio
SSE	Sum of the Squared Error
T	Tesla
T1	Longitudinal Relaxation Time
T2	Transverse Relaxation Time
TPM	Tissue Probability Map
US	Ultrasonography
WM	White Matter

Contents

1 Introduction

Hydrocephalus is a disease where an excess of cerebrospinal fluid (CSF) is built up in the brain. The disease is also called 'water on the brain' due to the overflow in the ventricles [Marieb and Hoehn, 2010]. CT scans of a normal and hydrocephalic brain are shown in Figure 1.1. Normally, CSF flows through the ventricles and surrounds the brain and the spinal cord, acting as a mechanical buffer. In people with hydrocephalus the flow of CSF or its absorption is blocked which causes pressure to build up in the ventricles of the brain. This excessive pressure can cause malfunction and cause brain damage. If left untreated, hydrocephalus can be fatal. The most common cause for hydrocephalus in infants in Uganda is infections such as ventriculitis or meningitis [Warf, 2005]. In sub-Saharan Africa hydrocephalus affects approximately 180.000 infants per year [Warf, 2013]. There is no cure for hydrocephalus, but it can be treated by inserting a shunt to provide a passage to excess fluid or by making a tiny hole in the third ventricle to relieve pressure from the ventricles.

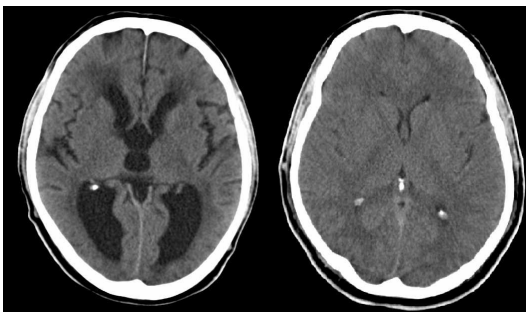


Figure 1.1: A CT scan of a hydrocephalic brain on the left-hand side and a normal brain on the right-hand side [Lecturio, 2020].

Hydrocephalus is diagnosed and examined with a clinical neurological exam and by brain imaging. In the CURE Children's Hospital of Uganda children are diagnosed with hydrocephalus by using ultrasonography (US) and computed tomography (CT) [Warf, 2005]. However, it is difficult to obtain brain images with US imaging since ultrasound waves are largely blocked by the skull [Eber and Villaseñor, 1991]. It can therefore only be used in prenatal cases to image the brain. In CT imaging high doses of radiation are involved, which can cause cancer and should be avoided if possible [Davies et al., 2011]. Magnetic Resonance Imaging (MRI) would be beneficial for examination of hydrocephalus in infants, since no ionizing radiation is involved and MRI shows good contrast between brain tissues. Unfortunately, in Uganda there is only limited access to MR scanners, due to economic limitations and the complexity of operating an MR system [Marques et al., 2019]. In 2013 only 0.08 MRI scanners were available per million population in Uganda, while there are 12.11 scanners per million population in The Netherlands in 2013 [GHO, 2016]. Therefore, Delft University of Technology (TU Delft), Leiden University Medical Center (LUMC), Pennsylvania State University (PSU) and Mbarara University of Science and Technology (MUST) set up a project to create an affordable, portable and simplified version of the MRI scanner for the CURE Childrens Hospital in Uganda to diagnose and examine the coverage of fluid in the brain in children with hydrocephalus.

One main restriction of this scanner is that it is required to be easy to use. Therefore, the goal is to make software for the MRI scanner that automatically makes and processes the scans. Part of the processing concerns the automatic segmentation of the scan into CSF and brain tissue regions. However, such automatic segmentation is a complex task.

Firstly, this complexity is due to problems specific to MR imaging. MR images are often affected by different artifacts, one of which is the intensity non-uniformity artifact (bias) [Liew and Yan, 2006]. This bias arises from radiofrequency (RF) excitation field inhomogeneity, non-uniform reception coil sensitivity, eddy currents driven by field gradients and electrodynamic interaction with the object in the scanner [Sled and Pike, 1998]. This bias field hinders automatic segmentation techniques, especially the methods that assume homogeneous signal intensity within each tissue class [Sled and Pike, 1998]. To make the segmentation of MR images more precise, an accurate estimation of the bias field is necessary, which is removed from the original image to obtain the true signal. Another major MR artifact is the partial volume averaging (PVA) artifact [Liew and Yan, 2006]. This artifact occurs when multiple tissues are combined in one voxel due to the limited resolution of the imaging device. The intensity of the voxel is then a weighted average of the tissues in the voxel [Santago and Gage, 1993]. Additionally, the signal is affected by random noise. The noise is caused by stray currents in the detector coil due to the magnetic field or thermal fluctuations in the detector coil itself [Liew and Yan, 2006]. Due to the cheap, portable, low-field build of the scanner, the images contain artifacts and have a low SNR, which increases the complexity of automatic segmentation. Added to these issues, the automation of segmentation in itself is complicated. This is because, tissues differ in intensities over different scans. Furthermore, the automatic segmentation should work on scans with hydrocephalus, which contain brains with a great variability in size, shape and distribution of CSF and other brain tissue.

1.1. Contributions

In this thesis project we aim to realize the foundations for a fast, automatic and practical 3D segmentation method for low-field MRI brain scans through several contributions:
Aim 1. The segmentation of low-field MRI scans is new. Therefore, this study sets out to find the most promising techniques for the segmentation of low-field scans. Our aim is to analyse different segmentation methods. In particular this study will research segmentation methods that are robust to noise and the bias field artifact, which are common in the low-field scans. We aim to facilitate propitious results with a bias-correcting segmentation technique evaluated on artificial images and low-field scans.
Aim 2. One of the most promising segmentation techniques nowadays is the use of neural networks. One of its advantages is that, after training, segmentation is fully automatic, which is required for our method. Neural networks for segmenting an MR image are employed regularly. However, we aim to develop a neural network that predicts both the segmentation and bias field from an input MR image. With the extraction of the bias field, we aim to improve the segmentation result found by the neural network, through incorporating an analytic segmentation method into the neural network. Our idea is that, by integrating an analytic segmentation method into a segmentation network, the fitting energy of the segmentation is minimized and therefore the segmentation results could be improved. We target to accomplish this improvement of segmentation by adding to the supervised loss (error measure between ground truth and prediction), an unsupervised loss function that is based on a bias-correcting analytic segmentation method.
Aim 3. Furthermore, we aim to find a positive effect of using an analytic unsupervised loss function on data, for which no ground truth segmentation is available. If no ground truth segmentations are available and only a supervised loss is applied, no training would be possible. Training can however be achieved when using an unsupervised loss. A positive effect of using an analytic unsupervised loss function would be advantageous for the segmentation of datasets that do not have ground truth segmentations available, like the low-field dataset. This effect will be investigated by transferring a trained neural network to a new dataset of which the ground truth segmentations will be ignored. We target to improve segmentation of this new dataset through training with the unsupervised loss.

1.2. Report Outline

Since this thesis is a graduation project for two Masters, this report consists of two parts, which are interwoven. The first part goes into the analytic segmentation of low-field scans. Firstly, a literature overview of available techniques is set out in Chapter 2. The most promising methods found in Chapter 2 were used for a preliminary data analysis on low-field scans. This analysis is presented in Chapter 3. Then in Chapter 4

exploration into segmentation improved by adjusting for the bias field is described. The results on low-field scans are shown in Chapter 5. Thereafter, the second section of this report begins, where neural networks are studied for the automatic segmentation of MRI scans. This starts with theory on deep learning for segmentation purposes in Chapter 6. In the next chapter the methodology for segmentation with a neural network is presented. Then, in Chapter 8 the results of the neural network are presented, analyzed and evaluated. Subsequently, in Chapter 9 the results of the analytic and deep learning method are discussed and recommendations for future research are given. Finally, conclusions on the automatic segmentation methods proposed in this thesis are reached in Chapter 10.

2

Segmentation Techniques for MRI

In this chapter the concept of image segmentation and different types of image segmentation methods will be described. Furthermore, research on different analytic techniques used for brain segmentation are discussed in this chapter. In the end, it will be concluded which techniques seem promising for the segmentation of low-field scans and will be analysed further.

2.1. Image Segmentation

Image processing can be defined as the computerized processing of images of different types to obtain the desired output [Hemalatha et al., 2018]. Segmentation is a form of image processing where certain information of the same category is extracted from the image. In segmentation the image is partitioned in non-intersecting regions such that each region is homogeneous and the union of no two adjacent regions is homogeneous [Pal and Pal, 1993]. This segmentation process leaves us with region(s) of interest which can be investigated further. There are many possible procedures available for image segmentation but there is no algorithm available that is considered good for all images [Pal and Pal, 1993]. Fu and Mui characterized segmentation techniques into three classes: (1) characteristic feature thresholding or clustering, (2) edge detection, and (3) region extraction [Fu and Mui, 1981]. Liew and Yan classify MR brain image segmentation approaches in three similar categories: classification-based, region-based, and contour-based approaches [Liew and Yan, 2006]. In the next paragraphs some classic segmentation approaches will be given. These methods will also come back in Section 2.2, where many examples are given of how researchers have used the segmentation techniques for segmenting CSF from brain scans.

In classification-based segmentation, voxels are classified in tissue classes according to a certain criterion. The simplest technique is thresholding, where the classification is based on the intensity range [Gonzalez and Woods, 2008]. Statistical classification methods are methods where the probability density functions of the intensity for different tissue classes are modeled as a mixture of Gaussians [Liew and Yan, 2006]. These methods are more robust than straightforward thresholding. Another frequently used classification technique is clustering [Stockman and Shapiro, 2001]. A clustering algorithm attempts to classify a voxel to a tissue class by using the similarity in each class. There exist hard clustering techniques, but also fuzzy clustering which allows a voxel to contain multiple tissue classes.

The region-based techniques are based on the assumption that a region belonging to an object generally has homogeneous characteristics. Unlike clustering methods, region-based methods consider spatial interactions between neighboring voxels [Gonzalez and Woods, 2008]. The simplest region growing techniques start by locating some seeds from which regions are grown. Neighboring voxels are added to the same region beginning at the seeds if they share homogeneous characteristics. The algorithm stops when the grown regions cover the entire image. Other examples of region-based techniques are the split-and-merge technique and watershed-based segmentation [Gonzalez and Woods, 2008].

The contour-based segmentation approaches segment objects by detecting their boundaries. To do so, these approaches rely on the gradient features near an object boundary. Methods that rely only on detecting edges

in an image are inherently more prone to noise and artifacts, since noise or artifacts in the gradient can have a great influence on the algorithm [Liew and Yan, 2006]. The simplest contour-based technique is mere edge detection. This usually leads to over- or under-segmentation since edges usually do not result in closed contours. For this algorithm to work well, an efficient edge linking algorithm is required. Another contour-based segmentation technique is based on deformable templates or active contours. These methods are more robust to noise, since they only impose contour smoothness and contour continuity [Hemalatha et al., 2018]. A disadvantage is that the initial contour needs to be close to the actual boundary of the object for the algorithms to converge successfully [Hemalatha et al., 2018]. Another issue with active contours is that the algorithms do not handle deeply convoluted boundaries well due to the contour smoothness requirement, which is an issue for convoluted tissues such as CSF, WM and GM [Liew and Yan, 2006]. A different type of active contour model is the level set method. In this method the contours are represented as the zero level set of a higher dimensional function, the level set function. An advantage of the level set method is that it is not necessary to explicitly parameterize the contour since the segmentation problem is formulated as a function over a fixed Cartesian grid. Moreover, the level set method is able to represent contours with complex topology and adapt their topology. Thus, the segmentation can change from one object to multiple objects or the other way around [Li et al., 2011].

Furthermore, there are segmentation techniques that are used for MR brain image segmentation, but that do not fall within the three categories. Atlas-based segmentation is a technique where a known segmented template is used as a reference frame for a new image by registration of the template to the image [Liew and Yan, 2006]. Atlas-guided segmentations are better suited for segmentation of structures that are stable over the population of study. Machine learning can also be used to segment tissues in the brain. An efficient learning algorithm is necessary to do so together with a representative learning set for satisfactory results [Liew and Yan, 2006]. Machine learning used for segmentation purposes is further delved into in Chapter 6.

2.2. Related Work: Automatic Brain Segmentation

In this section the focus is on analyzing research where segmentation of the brain was performed automatically, with special attention to research of automatic segmentation of hydrocephalic MRI brain scans.

2.2.1. Available Software

A number of techniques have been proposed for the automated segmentation of gray matter (GM), white matter (WM) and CSF of MRI scans. These techniques are statistical-based, geometrical-based, atlas-based and learning-based segmentation methods [Kazemi and Noorizadeh, 2014]. There are three software packages that are widely used in neuroimaging analysis for brain tissue segmentation [Kazemi and Noorizadeh, 2014]. These packages are SPM [Ashburner and Friston, 2005], FSL [Smith et al., 2004] and BrainSuite [Shattuck and Leahy, 2002]. They contain automated segmentation routines. To facilitate automated segmentation, skull stripping and intensity non-uniformity (bias) correction is applied to the input image. Skull stripping is the process of segmenting only the brain from the complete image by removing the skull, eyes, skin, and other soft tissue not involved in the brain. By performing skull stripping a brain image remains that only contains GM, WM and CSF. Registration is more robust if these non-brain parts are automatically removed from the image [Smith et al., 2004]. Bias correction is correction applied to the image for the modulation of the intensity of the image (bias) [Ashburner and Friston, 2005]. For visual inspection, this bias is usually not a problem. However, with automated techniques based on intensity values of the voxels the bias field can impede processing of the image. Therefore, the bias is corrected for. The three packages all have their own routines for skull stripping, bias correction and segmentation. Another frequently used segmentation program is called FreeSurfer [Dale et al., 1999]. This program can be integrated within FSL and SPM.

SPM SPM is based on a unified segmentation model that combines skull stripping, bias correction and segmentation [Ashburner and Friston, 2005]. Tissue probability maps are used as prior information of the tissue classes. Bayes' rule is applied to produce the posterior probability of each tissue class [Ashburner and Friston, 2005]. Thus, probability values are assigned to each voxel and voxels corresponding to a greater tissue probability in the maps are counted as members of that class [Kazemi and Noorizadeh, 2014]. In the paper by Ishii et al. [Ishii et al., 2008] SPM was used to segment intracranial volume (ICV), GM, WM and CSF to investigate morphological changes in the brain of idiopathic normal pressure hydrocephalic patients. They

used the standard voxel based morphology (VBM) in the SPM tool [Wright et al., 1995] to segment the different tissues. Manually, extra-cranial segments were removed from the segmentations, taking 15-45 minutes per subject. To further improve the results of the segmentation customized templates and probability maps were made for all patients [Ishii et al., 2008], since brains with hydrocephalus have a great variability in size, shape and distribution of CSF. The manual pre- and postprocessing changes the automatic segmentation procedure of SPM into a segmentation procedure that is time consuming for the user and user-dependent. Still, SPM is widely used as a segmentation method within this field of work [Gunter et al., 2019], [Ishii et al., 2013], [Serulle et al., 2014], [Yamashita et al., 2009].

FSL In FSL the image is first stripped of all non-brain tissues with the brain extraction tool (BET) [Smith, 2002]. Then, in the next step, the segmentation is performed whilst also correcting for the bias. This procedure is based on the fact that intensity correction requires knowledge of the tissue class of a voxel (segmentation) and segmentation requires a correct intensity for every voxel [Zhang et al., 2001]. In this method, the histogram of the input image is modeled as a mixture of Gaussians with mean and variance for each class. The segmentation allows a reconstruction of the image; subtracting this from the real image gives an estimate of the non-uniformity. This whole process is then iterated between segmentation and intensity non-uniformity correction until convergence [Smith et al., 2004], [Kazemi and Noorizadeh, 2014]. In the paper by Grimm et al. [Grimm et al., 2020a] FSL is used to segment 2 classes of tissues: brain matter and CSF. Their goal is to automatically extract volumetry of CSF and brain volume in severe pediatric hydrocephalus, in the clinical course after intervention. The quality of the segmentation was visually checked and in the study group of 38 patients the visual inspection revealed a proper segmentation without need for manual correction [Grimm et al., 2020a].

BrainSuite In BrainSuite the three routines follow after each other: first skull stripping, then bias correction and at last brain tissue segmentation [Shattuck and Leahy, 2002]. The edge of the skull is found by using mathematical morphological operators. The bias correction is performed by estimating the correction field based on a series of local estimates of the tissue gain variation. Finally, the segmentation is performed via a partial volume classifier [Kazemi and Noorizadeh, 2014].

FreeSurfer FreeSurfer started as an automatic program for obtaining an accurate and explicit representation of the cortical surface in individual subjects [Dale et al., 1999]. In later work an automatic whole brain segmentation method is developed. The method can assign one of 37 labels to each voxel. This method is atlas-based. They take a Bayesian approach with the atlas and anatomical knowledge as prior. The segmentation is computed iteratively by the maximum a posteriori (MAP) estimate [Fischl et al., 2002]. In Eide et al. [Eide et al., 2020] FreeSurfer is used to segment the choroid plexus within the lateral ventricles as well as the CSF within the lateral ventricles. The images of each patient were used to create a population specific median atlas. The segmentations were checked and no visible errors were found.

2.2.2. Magnetic Resonance Fingerprinting

Standard MR imaging is mainly qualitative imaging. This is because the same material can have different intensities in different datasets. These values are different, since they are dependent on the type and the setup of the scanner, the detector used and other factors that are involved in the image acquisition [Ma et al., 2013]. Therein lies the complexity of automatically segmenting a certain tissue from an MRI scan. It will have a different intensity over different scans and the intensities in the scan of the same tissue will also vary due to for example the bias. To deal with these issues Ma et al. introduced magnetic resonance fingerprinting (MRF) as a way of quantitative MR imaging [Ma et al., 2013]. MRF permits the simultaneous non-invasive quantification of multiple important properties of a material or tissue. This is done by acquiring the signal through pseudo-randomized acquisition that causes the signals from different materials or tissues to have a unique signal evolution or “fingerprint” that is simultaneously a function of the multiple material properties under investigation [Ma et al., 2013]. In the research by Virhammer et al. quantitative MRI is used to segment CSF volume in patients with hydrocephalus [Virhammar et al., 2015]. The MR signal was acquired with the MR imaging sequence QRAPMASTER [Warntjes et al., 2008]. This provides a rapid simultaneous quantification of longitudinal relaxation time (T1), transverse relaxation time (T2) and proton density (PD). The acquired signal is then put into the postprocessing software SyntheticMR, which uses the combination of T1, T2 and PD to segment ICV, WM, GM, and CSF. The automatic CSF volume corresponded well with manual segmentations.

2.2.3. Atlas-Based Segmentation

Another segmentation method is the atlas-based method. In this method the atlases are made (partially) manually and are specific to the group of subjects. An example was already mentioned when SPM was used for segmentation, with patient specific tissue probability maps for the whole patient group [Ishii et al., 2008]. Another group of researchers developed a segmentation technique called RUDOLPH to first segment the CSF and then segment the CSF into 7 different classes [Ellingsen et al., 2016]. RUDOLPH integrates two segmentation approaches: a patch-based tissue classification method and a multi-atlas label propagation with expectation maximization. The first step of this approach is to register 10 manually labeled atlases into the subject's space. The output is a probabilistic segmentation. This segmentation is improved by correcting inaccurate labels with expectation maximization based on Euclidean distance. In subsequent research the RUDOLPH method was improved by incorporating an adaptive atlas and by correction based on a weighted distance based on image intensities [Shao et al., 2018].

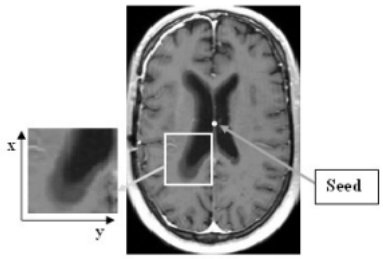


Figure 2.1: Automatically selected seed point in ventricles for the start of the active contour model [Linguraru and Butman, 2009].

2.2.4. Active Contours

A segmentation method based on active contour models is proposed in Butman and Linguraru [Butman and Linguraru, 2008]. Their approach uses a combination of fast marching and geodesic active contour level sets. As data they use 3D T1-weighted scans obtained following injection of an intravenous contrast agent. The fast marching technique uses a seed point from which the surface expands. This seed point is chosen by the user on the septum pellucidum (line between the bodies of the lateral ventricles). The segmentation provided by the fast marching technique is used as input level image into the geodesic active contour. The contour is optimized by minimizing the energy function of the contour via a gradient flow equation. In this equation the speed, curvature and the attraction to edges of the contour are controlled by selectable weights [Caselles et al., 1997]. In a later article [Linguraru and Butman, 2009] the researchers used the same segmentation technique as before, but they began the segmentation by automatically setting a seed on the septum pellucidum (Figure 2.1).

2.2.5. Region Growing

Region growing is another popular segmentation technique. Region growing appends neighboring voxels to the same region if they share homogeneous characteristics. Most region growing algorithms start with choosing seeding points, from which the regions are grown. The user decides the homogeneity threshold, that determines if two neighboring voxels are part of the same region. Automated 3D region growing algorithms do exist. This automation can be done via an assessment function applied to the image's histogram. This assessment function determines the optimal homogeneity criterion [Muller et al., 2002]. The seeds are automatically selected by Otsu's automated thresholding [Otsu, 1979] followed by a 3D binary erosion, so that the seeds are inside the object.

2.2.6. Clustering

Fuzzy clustering was used to segment CSF, WM and GM from an axial MRI brain image [Brandt et al., 1994]. Fuzzy logic classification is based on the idea that a voxel can simultaneously belong to more classes and does so to a varying degree. Essentially, fuzzy clustering takes the PVA artifact into account. The algorithm proposed by Brandt et al. consists of three steps: 1) elimination of low-level intensity noise via thresholding, 2) creation of a brain mask from the image, 3) fuzzy clustering for the segmentation. To begin the clustering procedure, an initial estimate of the gray scale values representing the centroid for each cluster must be pro-

vided. Centroids are data points representing the intensity centers of the clusters. Three clusters are chosen which represent the three tissue classes: CSF, WM and GM. The final solution is independent of the initial estimate of the centroid values, but if the initial estimate is far from the solution, it will take more iterations Brandt et al. [1994]. A limitation of this method is that a small cluster can get absorbed into a larger adjacent one.

2.2.7. Bias Field Correction

As mentioned in the Introduction one major artifact that can cause problems for automatic segmentation of an MRI scan is the non-uniformity intensity (bias) artifact. This bias artifact disturbs the intensity values in the image. Since many automatic segmentation techniques described above use in some way the intensity value as a metric to segment the image, a spatial disturbance in the intensity can cause an incorrect segmentation. Therefore, in most automatic software like SPM and FSL this bias artifact is corrected for by determining the bias field and extracting this field from the measured image. In BrainSuite the bias field is determined first, after which the segmentation is performed. SPM and FSL combine the segmentation with the bias field correction, since the segmentation depends on the correct intensities for the tissues and the bias correction requires knowledge of the tissue in a voxel. Then the bias field and the segmentation are iteratively updated until convergence. Li et al. proposed an algorithm to estimate in a local region the bias field and the segmentation [Li et al., 2011]. It is an extension of the Chan-Vese level set algorithm, which is an active contour algorithm. The algorithm starts with an initial bias field and contour and is then iteratively updated by minimizing the energy of the segmentation. Results of Li's model applied to different images are given in Figure 2.2.

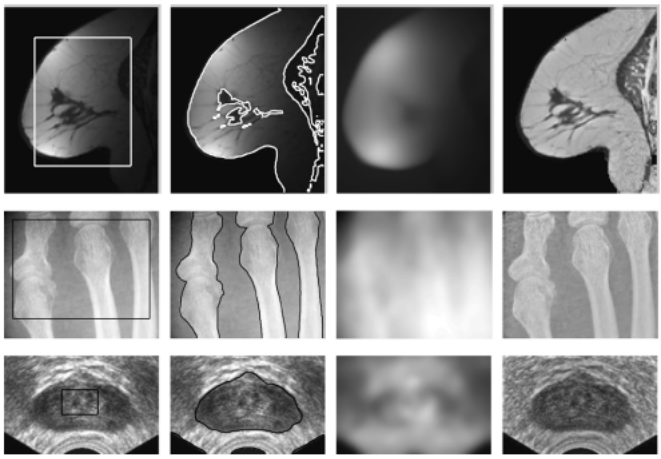


Figure 2.2: From top to bottom: an MR image of a breast, an X-ray of bones, and an ultrasound image of a prostate. Column 1 shows the original image with the starting contour of the level set. Column 2 shows the final contours. The approximated bias field is shown in column 3. Column 4 displays the bias-corrected image. [Li et al., 2011]

2.3. Promising Techniques for Low-Field MRI

In this chapter many segmentation methods were discussed. Not all algorithms seem promising for their use on low-field scans. This is because the low-field scans have a low SNR, are riddled with artifacts, and only few scans are available. Also, for the future purpose the segmentation method should be chosen that takes the variability of hydrocephalic brains into account. In the following paragraphs the possible issues of the mentioned algorithms will be discussed and it will be concluded which methods seem the most promising.

First, we will take a look at the available software. Based on the quantitative comparison of SPM, FSL and BrainSuite performed by Kazemi and Noorizadeh [Kazemi and Noorizadeh, 2014], Brainsuite performs the worst at segmenting CSF from the Brainweb simulated MR images [Kwan et al., 1999]. Brainsuite also underperforms in noisy images, which will be the case in our project. SPM and FSL yield similar results [Kazemi and Noorizadeh, 2014]. Due to the necessity for probability maps in the SPM tool and the knowledge that there is a great variability in brains of hydrocephalic patients, SPM will be likely to be less accurate as an automatic tool. Therefore, FSL seems the most promising to apply to the low-field data, but due to the low SNR and low

resolution its performance could still be suboptimal.

Then, segmentation by performing MRF was discussed. Performing MRF on the low-field MRI scanner is still being researched. Because it is still in development, we chose not to move further with segmentation based on MRF. Atlas-based techniques are based on good reference data for the population of study. The limited availability of data and the unavailability of hydrocephalic data makes using atlas-based techniques impossible.

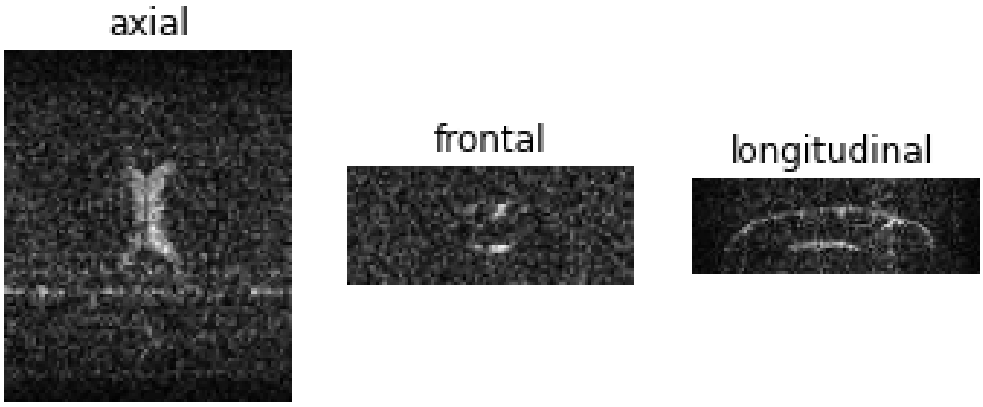
A few techniques seemed to have potential: region growing algorithms, clustering algorithms and active contour algorithms. The clustering and active contour algorithms are however, not fully automatic since parameters need to be chosen for them to work. The clustering algorithm needs the initial centroid values and the number of clusters. However, taking four clusters is a very logical choice in the case of segmenting the brain and the wrong choice in centroid values will only slow down the algorithm but the algorithm will still converge. For active contour models the initialization of the contour and the parameters of the loss function need to be chosen. Data analysis will establish the influence of these parameters and therefore the effect on the automatic process of segmentation. Nevertheless, these techniques are dependent on the intensity values of the image. Therefore, adjusting for the bias field in the segmentation procedure should be favorable and thus investigated further.

3

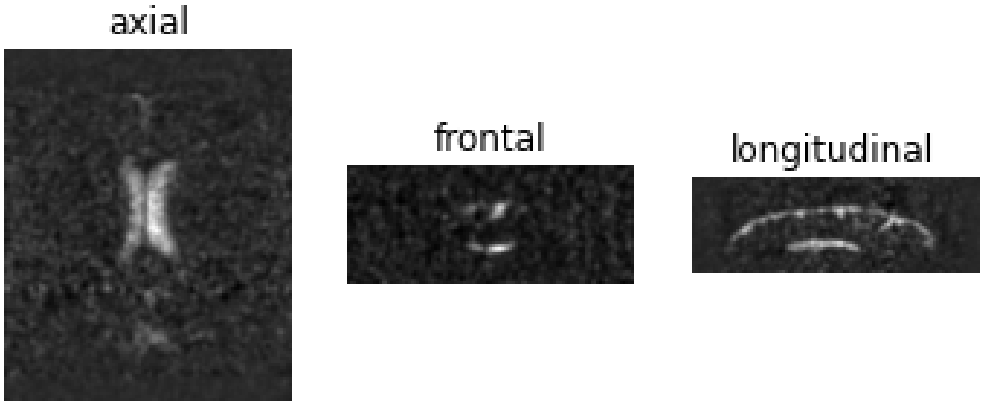
Preliminary Segmentation Approaches

In this chapter a data analysis will be performed. This chapter should be conceived by the reader as the starting ground of segmenting the CSF from low-field MRI data in this project. The available data comprises one T2-weighted scan of an adult brain, acquired using the low-field MRI scanner. The T2-weighted sequence that was chosen yields a signal solely originating from CSF. The segmentation techniques applied in this chapter are the techniques that were encountered during the literature research described in Chapter 2. In the literature research they seemed promising for application in this project. Furthermore, they are (semi-)automatic and easy to implement, so that preliminary results could quickly be generated. In this chapter some 3D segmentation results are shown. Segmentation techniques of which the results look promising are investigated in further detail in the remainder of this research project.

The original T2-weighted image has a size of 30x88x72 voxels. It has a low SNR, a low resolution and the image contains artifacts. In Figure 3.1a three slices of the original T2-weighted scan are shown, from which the high noise level is very clear. With some simple algorithms the image was preprocessed, such that the contrast was improved and the noise was suppressed. The preprocessing started with zero padding at the boundaries of the K-space of the signal. At all boundaries half the dimension's lengths were zero padded. The information that the K-space already contained remained unchanged. When the Fourier transform was applied to this zero padded K-space, the same information was obtained but now in an image twice as large in all directions. Zero padding therefore improved the apparent spatial resolution, since zero padding acted as a method to interpolate the signals from neighboring voxels, which made the image smoother [Bernstein et al., 2001]. Following the zero padding, the stripe artifact had to be removed. The stripe artifact was clear in all the axial and longitudinal slices. It had roughly the same intensity as the CSF which made it difficult for the process of automatic segmentation of the CSF. The removal of the stripe was accomplished by first taking the mean of the first 10 and last 10 axial slices of the complete image. These 20 slices did not contain any CSF data. The mean of these slices therefore did not contain important information. However, the 20 slices did contain the stripe artifact, since this artifact was present in every axial slice. This mean slice was then substracted from every axial slice in the 3D scan. This process suppressed the stripe artifact. Then the image was rescaled such that intensities are ranging from 0 to 1. Furthermore, a Gaussian filter [Virtanen et al., 2020] was applied to the resulting image to suppress the noise. The Gaussian filter was implemented with a kernel with a standard deviation of 1. In Figure 3.1b the preprocessed image is shown. From the slices it is clear that the stripe artifact is suppressed and the noise is filtered, which has increased the contrast between CSF and background.



(a) The original image



(b) The preprocessed image

Figure 3.1: Above axial, frontal and longitudinal slices of the original T2-weighted scan are shown. Below axial, frontal and longitudinal slices of the preprocessed image are shown.

3.1. Region Growing

The region growing algorithm starts from an initial seed point. From this seed point the region is grown until no neighboring points can be added to the region compliant with the condition. The condition is as follows: the absolute difference between a voxel on the side of the region and the neighboring voxels must be below a predetermined threshold and the absolute difference between the seed point and the voxel on the side of the region is below a predetermined tolerance. The first part of the condition ensures that the region stops at an edge. The second part of the condition ensures that the region growing algorithm will stop when the new points are too different from the seed point. Otherwise the algorithm can add every voxel of the image to the region when the image gradient is smooth due to low contrast. This problem can also occur when the image is corrupted with a lot of noise, which distorts edges and can cause under- or oversegmentation. The region growing algorithm is used to segment the lateral ventricles from the brain scan. The region growing algorithm is described below, in Algorithm 1. For this algorithm a threshold, a tolerance and a seed point must be chosen. Therefore, the algorithm is semi-automatic. The seed point can be chosen automatically, however, by finding the maximum value intensity point inside the brain scan. Figure 3.2 shows the automatically selected seed point. The seed point is in the lateral ventricles of the brain. Figure 3.3 shows the region found by implementing the region growing algorithm employing the seed point found by automatic seed point selection. This region shows a resemblance to the lateral ventricles of the brain. Unfortunately, with this method only one region of CSF is found, unless multiple seed points are taken. The automatic choice of seed points will then be less trivial. Additionally, the whole algorithm will become more complex, when not only CSF but also GM and WM will be segmented, especially with the high noise level. Thus, application to more low-field data was not investigated further.

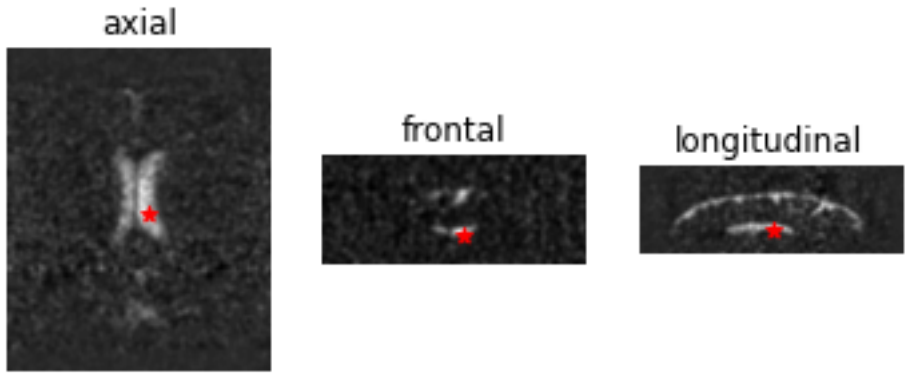


Figure 3.2: Automatic seed point selection

Algorithm 1 Region Growing Algorithm

```
Specify the threshold T and the tolerance tol
Automatically set the seed point from where the region R is grown
for point P in R do
    Determine the neighboring points of P that are not in R
    if |I(P)-I(neighboring point)| < T AND |I(seed point)-I(neighboring point)|<tol then
        Add neighboring point to R
    end if
    STOP when all neighboring points are checked
end for
return R
```

Figure 3.3: The first three boxes show the axial, frontal and longitudinal slice of segmentation found by region growing algorithm. The threshold is taken to be 0.15 and the tolerance 0.45. The region is grown in our preprocessed image from the seed point specified by the red star in Figure 3.2.

3.2. K-Means Clustering

K-means clustering is a technique where the voxels are clustered on the basis of similarity. K stands for the number of clusters the data will be divided into. In comparison to the fuzzy c-clustering method, k-means clustering classifies every voxel in one cluster only. As input of the algorithm the user needs to choose k centroids. The algorithm works through expectation-maximization. The first step of the process, the expectation step, assigns each data point to its nearest centroid. Then in the maximization step the new means of the clusters are computed and the centroids are updated. When the centroids converge the sum of the squared error (SSE) is computed. The SSE gives the squared Euclidean distance between each point in the cluster and the cluster's centroid as a measure of error. The objective of the k-means clustering algorithm is to minimize

this error. The overview of the algorithm is given below (see Algorithm 2 [Mohd et al., 2012]). The in-built function *Kmeans* of the module Scikit-learn [Pedregosa et al., 2011] is used to compute the clusters in the 3D image. As input 2 clusters are given with centroids [0,1] as the image is scaled from 0 to 1. K is chosen as 2, because there are 2 main types of elements in the brain scan: CSF and background. The results of the algorithm are given in Figure 3.4. From these results it is immediately clear that the CSF clusters contains a lot more voxels than just CSF. This is due to the large amount of noise in the image. To remove this noise from the CSF cluster, the algorithm starts with more clusters. Then, noise that has a higher intensity than the background can be classified into an intermediate cluster. Starting with 3 clusters with centroids [0,0.5,1] already gives much better results for the CSF cluster (see Figure 3.5). This method is much faster than the region growing algorithm. Also, one cluster can contain multiple objects. This will be the case for segmentation of CSF in the brain, since CSF in not only in the ventricles but also in the region around the brain. Therefore, this method will be applied to other scans.

Algorithm 2 K-Means Clustering Algorithm

Specify the number k of clusters to assign
Give k random centroids
repeat
  expectations: Assign each point to its closest centroid
  maximization: Compute the new centroid of each cluster
until The centroid positions do not change

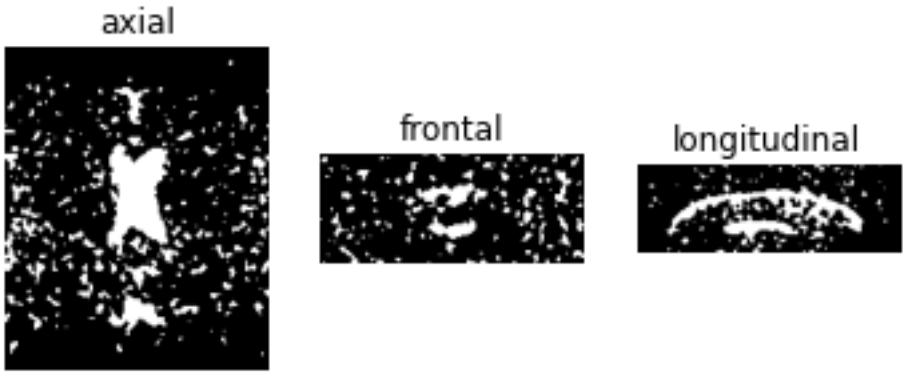


Figure 3.4: Results of the K-means algorithm with $k = 2$. White is the first cluster and black the second. The final centroids are [0.22,0.32].

Figure 3.5: Results of the K-means algorithm with $k = 3$. White is the CSF cluster and black the other two clusters. The final centroids are [0.21,0.26,0.49].

3.3. Chan-Vese Level Set Method

Other promising segmentation methods mentioned in Chapter 2 are active contour models. Since there is no indication of a bias field, the first focus is on segmentation with active contour models without bias correction. Active contour models can be solved explicitly with snake models. In those methods the contour of the snake is attracted to the edges of the object. However, CSF is everywhere in the brain and cannot be defined as one object. Therefore, a better option might be to apply a level set method. In a level set method a differential equation is solved iteratively. The solution is a function of the image where the positions of its zero level set are the positions of the contour that separates an object from the rest of the image:

$$C = \{\mathbf{x} \in \Omega : \phi(\mathbf{x}) = 0\}, \tag{3.1}$$

where C is the contour, $\mathbf{x}$ is a voxel in the image, Ω is the domain of the image and ϕ is the level set function. Thus, the contour is not extracted explicitly as in a snake method but implicitly with the zero-crossing of the level set function ϕ. Therefore, the segmentation boundary can handle topological changes more easily. This means that the contour given by the level set can efficiently handle convex objects compared to the snake model and it can split into multiple contours surrounding multiple objects. A particular type of level set technique is the Chan-Vese level set [Chan and Vese, 2001]. The method is not based on an edge-function like snake models. This is advantageous for noisy images and images where the objects have very smooth boundaries and thus gradients are low. Essentially, the Chan-Vese method minimizes the fitting energy of the segmentation derived from the contour together with regularization terms for the contour:

$$\arg\min_{c_1,c_2,C} \Big[\mu \,\text{Area}(C) + \nu \,\text{Volume}(\text{inside}(C)) + \lambda_1 \int_{\text{inside}(C)} |f(\mathbf{x}) - c_1|^2 d\mathbf{x} + \lambda_2 \int_{\text{outside}(C)} |f(\mathbf{x}) - c_2|^2 d\mathbf{x}\Big], \tag{3.2}$$

where c_1 and c_2 are unknown constants, f is the image, and $\mu > 0$, ν, $\lambda_1, \lambda_2 \geq 0$ are fixed parameters chosen by the user. In this energy minimization, the first term controls the smoothness of the contour by penalizing an increase in the area of the contour. The second term controls the size of the volume enclosed by C. The last two terms are image depending terms that describe the fitting energy. By penalizing these terms a segmentation is obtained as the best two-phase piecewise constant approximation of the image f [Getreuer, 2012]. The parameters determine the impact of the term on the energy minimization. In the Chan-Vese level set formulation the level set function ϕ is set to be Lipschitz continuous and is chosen to be positive inside the contour C and negative outside. Therefore we can rewrite the area of C, the volume inside of C and the image integrals as integrals over the whole domain [Zhao et al., 1996]:

$$\text{Area}(C) = \int_\Omega |\nabla H(\phi(\mathbf{x}))| d\mathbf{x} = \int_\Omega \delta(\phi(\mathbf{x})) |\nabla \phi(\mathbf{x})| d\mathbf{x}, \tag{3.3}$$

$$\text{Volume}(\text{inside}(C)) = \int_\Omega H(\phi(\mathbf{x})) d\mathbf{x}, \tag{3.4}$$

$$\int_{\text{inside}(C)} |f(\mathbf{x}) - c_1|^2 d\mathbf{x} = \int_\Omega |f(\mathbf{x}) - c_1|^2 H(\phi(\mathbf{x})) d\mathbf{x}, \tag{3.5}$$

$$\int_{\text{outside}(C)} |f(\mathbf{x}) - c_2|^2 d\mathbf{x} = \int_\Omega |f(\mathbf{x}) - c_2|^2 (1 - H(\phi(\mathbf{x}))) d\mathbf{x}, \tag{3.6}$$

$$\tag{3.7}$$

with H the Heaviside function and δ its derivative, the Dirac delta function:

$$H(t) = \begin{cases} 1, & \text{if } t \geq 0 \\ 0, & \text{if } t < 0 \end{cases}, \qquad \delta(t) = \frac{d}{dt} H(t).$$

Thus, the complete energy minimization can now be computed as an integral over the whole domain Ω:

$$\begin{aligned} \arg\min_{c_1,c_2,\phi} \Big[\mu \int_\Omega \delta(\phi(\mathbf{x})) |\nabla \phi(\mathbf{x})| d\mathbf{x} + \nu \int_\Omega H(\phi(\mathbf{x})) d\mathbf{x} + \\ \lambda_1 \int_\Omega |f(\mathbf{x}) - c_1|^2 H(\phi(\mathbf{x})) d\mathbf{x} + \lambda_2 \int_\Omega |f(\mathbf{x}) - c_2|^2 (1 - H(\phi(\mathbf{x}))) d\mathbf{x}\Big]. \end{aligned} \tag{3.8}$$

The minimization is solved by alternatingly updating c_1, c_2 and ϕ. For fixed ϕ, the optimal values for c_1 and c_2 are the region averages (see Appendix A.2 for the derivation):

$$c_1 = \frac{\int_\Omega f(\mathbf{x}) H(\phi(\mathbf{x})) d\mathbf{x}}{\int_\Omega H(\phi(\mathbf{x})) d\mathbf{x}} \tag{3.9}$$

$$c_2 = \frac{\int_\Omega f(\mathbf{x}) (1 - H(\phi(\mathbf{x}))) d\mathbf{x}}{\int_\Omega (1 - H(\phi(\mathbf{x}))) d\mathbf{x}} \tag{3.10}$$

Keeping c_1 and c_2 fixed and minimizing the energy with respect to ϕ (Equation 3.8), we obtain the Euler-Lagrange equation for ϕ, which can be numerically solved to find the level set function. It is important that the algorithm does not get stuck in a local minimum, but finds the level set function ϕ that gives us the global minimum. To solve this problem the Heaviside function and the Dirac delta function are regularized as follows:

$$H_\epsilon(t) = \frac{1}{2}\Big(1 + \frac{2}{\pi}\arctan(\frac{t}{\epsilon})\Big) \text{ and } \delta_\epsilon = \frac{d}{dt} H_\epsilon = \frac{1}{\pi}\frac{\epsilon}{\epsilon^2 + t^2}, \tag{3.11}$$

in which ϵ specifies the width of the functionals. Now, when applying the Euler-Lagrange equation to the energy minimization (Equation 3.8) the equation will act on all the level curves, but stronger on the level curves around $\phi = 0$. Therefore, a global minimum can be obtained independent of the position of the initial contour. The following Euler-Lagrange equation is obtained (see Appendix A.1 for the derivation):

$$\begin{cases} \frac{\partial \phi}{\partial t} & = \delta_\epsilon \Big[\mu \text{div}\Big(\frac{\nabla \phi}{|\nabla \phi|}\Big) - \nu - \lambda_1 (f - c_1)^2 + \lambda_2 (f - c_2)^2\Big] \quad \text{in } \Omega, \\ \frac{\delta_\epsilon(\phi)}{|\nabla \phi|}\frac{\partial \phi}{\partial \vec{\mathbf{n}}} & = 0 \quad \text{on } \partial\Omega, \end{cases} \tag{3.12}$$

where $\partial\Omega$ is the boundary of the domain. The time derivation of the level set is specified in the first line. The second line states the boundary condition. The level set function can now be found by solving the Euler-Lagrange equation numerically. Many different numerical methods can be used to solve this problem. He and Osher proposed a method based on the topological derivative [He and Osher, 2007], Getreuer proposed a semi-implicit gradient descent method [Getreuer, 2012], Badshah and Chen developed a multigrid method [Badshah and Chen, 2008] and algorithms based on graph cuts were brought forward [Bae and Tai, 2009], [El-Zehiry et al., 2007]. My implementation is based on the 2D implementation in Python [Van der Walt et al., 2014] and the mathematics described in the paper by Getreuer [Getreuer, 2012] and the paper by Chan and Vese [Chan and Vese, 2001]. For the numerical implementation, the 3D image space Ω is divided into a cartesian grid with unit cubes as elements. Each voxel of the original image is a unit cube. ϕ on the right-hand side of the Euler-Lagrange equation (Equation A.8) is discretized in space with finite differences. The Neumann boundary condition is enforced by duplicating pixels at the border of the image space Ω. The time variable is discretized with explicit/forward Euler (Equation 3.13) and implicit/backward Euler (Equation 3.14), where n is the current time step and $n+1$ is the next time step. Forward Euler can be easily implemented as the whole right-hand side is known. Backward Euler is more complex to implement as partial derivative in time is based on ϕ^{n+1}, which is unknown. This issue is solved by taking the nonlinear terms in time-step n and the other terms in time-step $n+1$. The nonlinear terms are the divergence term and $\delta_\epsilon(\phi_{i,j,k})$.

$$\phi^{n+1}_{i,j,k} = \phi^n_{i,j,k} + \Delta t \frac{\partial \phi^n_{i,j,k}}{\partial t} \tag{3.13}$$

$$\phi^{n+1}_{i,j,k} = \phi^n_{i,j,k} + \Delta t \frac{\partial \phi^{n+1}_{i,j,k}}{\partial t} \tag{3.14}$$

The complete algorithm will stop when the mean squared difference between ϕ^n and ϕ^{n+1} is below a certain tolerance, tol, or when a certain amount of iterations is reached. This means that ϕ has converged and thus an energy minimum has been found. The initial level set function can be anything, but it has been observed that starting with a checkerboard pattern leads to fast convergence [Getreuer, 2012]. A coarse overview of the algorithm is given below in Algorithm 3. The solutions of the Chan-Vese algorithm with the explicit time step are given in Figures 3.6 and 3.7. The implicit method was also implemented. However, during testing no convergence problems occurred with the explicit method. Therefore, using the implicit method was not necessary and results of this method are not shown.

Algorithm 3 Chan-Vese Level Set Algorithm

Initialize ϕ^0
while $|\phi^{n+1}$-$\phi^n| >$ tol AND n<ITER **do**
  $\phi^{n+1} \rightarrow \phi^n$
  Compute c_1 and c_2 as the region average with the current ϕ^n
  Evolve ϕ^{n+1} with implicit or explicit time step
end while

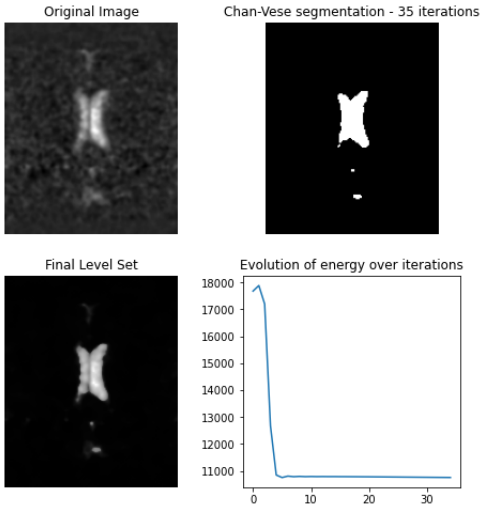


Figure 3.6: The solution of the Chan-Vese level set algorithm computed with the explicit time step. In the left-top corner a slice of the original image is shown. The right-top corner shows the segmentation based on the level-set (left-bottom corner) of the final iteration. The right-bottom corner shows the evolution of the energy over the iterations. The parameters used for the computation are μ=0.05, ν=0.2, λ_1=1, λ_2=5, dt=10, tol= 0.005, and with a max number of iterations of 1000.

Figure 3.7: Axial, frontal and longitudinal slices of the segmentation solution of the Chan-Vese level set algorithm with the explicit time step from Figure 3.6.

4

Bias Field Correction

The bias field artifact can have a huge influence on the resulting segmentation as most segmentation techniques described in Chapter 2 and 3 are based on the intensity values of the image. This bias field artifact changes the intensities of the image non-uniformly. In Section 2.2.7 Li's model was proposed as a way to compute the bias field and determine the segmentation simultaneously. In this chapter a 2D implementation of Li's level set model is applied on a 7T T2-weighted brain scan. For this image, the goal is to remove the bias field such that a better CSF segmentation can be obtained. Then, a multi-phase implementation of the model is applied on a 2D T1-weighted Brainweb image [Aubert-Broche et al., 2006], which is affected by different artificial bias fields. GM, WM and CSF will be segmented from this image while removing the bias field. Since the scan is an artificial image, ground truth segmentations are available. The results will therefore be evaluated by comparing to these ground truth segmentations.

4.1. T2-Weighted Scan

The 7T T2-weighted scan used in this experiment is shown in Figure 4.1. The size of the image is 500x512 and the intensity of the image ranges from 0 to 255. The image shows the CSF in the ventricles and in the subarachnoid space. Signal from fat (skin around the skull) is also visible in an oval around the CSF regions. From Figure 4.1 it is clear that there is a non-uniform intensity artifact present, since the upper and lower regions of the CSF have a higher intensity than the regions in between, while the signal comes from the same 'tissue'.

In previous sections the k-means algorithm and the Chan-Vese level set algorithm were implemented to obtain a segmentation. These algorithms are dependent on the intensity of the regions. In Figure 4.2 the result of the k-means algorithm is shown. The algorithm starts with two clusters with centroid values [0,100]. As can be seen from the image, the lower and upper region of the CSF is part of the segmentation but, from the CSF regions in between, many pixels are missing. This is because the signal intensity of the middle regions of CSF are much lower compared to the upper and bottom CSF regions. In Figure 4.3 the results of the Chan-Vese level set algorithm are shown for different parameters. The left-hand side image shows roughly the same result as the k-means algorithm. The segmentation produced with λ_2=3 shows a complete different result (see Figure 4.3c). Firstly, part of the fat is segmented as well. Secondly, the segmentation looks oversegmented as some pixels are added to the segmentation which should not be included: the overall segmentation looks 'thickened'. Some regions where this is clearly visible are accentuated with red circles shown in Figure 4.3c. From adjusting some parameters it can be concluded that this oversegmentation is due to the choice of parameters and the non-uniform intensity artifact. Another Chan-Vese level set segmentation is shown in Figure 4.3b. In this segmentation less fat is segmented and the CSF regions in the middle have increased, but still some regions of CSF are not part of the segmentation and oversegmentation is also present.

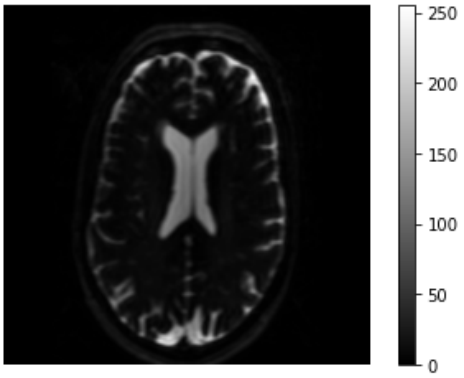


Figure 4.1: A 2D axial slice of a 7T T2-weighted MRI brain scan. The intensity value range is given in a colorbar on the right-hand side of the image

Figure 4.2: Results of the K-means algorithm for 2 clusters with begin centroid values of [0,100]. Black is the first cluster and white the second. The final centroids are [6,78,105,81].

(a) The segmentation result with λ_2=1.1 (b) The segmentation result with λ_2=2 (c) The segmentation result with λ_2=3

Figure 4.3: The solution of the Chan-Vese level set algorithm computed with the explicit time step. The parameters used for both computations are μ=0.025, ν=0., λ_1=1, dt=50, tol= 0.05, and with a max number of iterations of 5000. On the left-hand side λ_2=1.1, on the right-hand side λ_2=3.The green oval shows a region where fat is segmented into the CSF segmentation. The red ovals shows regions where the CSF is oversegmented.

Thus, the k-means clustering and the Chan-Vese level set algorithm cannot achieve good enough segmentation results since this image is corrupted with a non-uniform intensity artifact. Therefore, Li's model is investigated. The goal of Li's algorithm is to estimate the bias field and the true signal from the measured signal. In this algorithm it is assumed that the measured image, I, is described as follows [Li et al., 2011]:

$$I = bJ + n, \tag{4.1}$$

where b is the bias field, J is the true signal and n is additive noise. It is assumed that the bias field and the true signal have the following properties:

1. The bias field b is slowly varying in the entire domain.
2. The true image intensities J are approximately constant within each class of tissue. This means that $J(\mathbf{x}) \approx c_i$ for $\mathbf{x} \in \Omega_i$ where $\{\Omega\}_{i=1}^{N}$ is a partition of image domain Ω.

Based on assumption 1, the following holds for points $\mathbf{y}$ in a small neighborhood O_x around point $\mathbf{x}$:

$$b(\mathbf{y}) \approx b(\mathbf{x}). \tag{4.2}$$

In that small neighborhood there are voxels present from the different classes of tissues. Thus, if $\mathbf{y} \in \Omega_i$, then $b(\mathbf{y})J(\mathbf{y}) \approx b(\mathbf{x})c_i$ according to assumption 1 and 2. Then the k-means algorithm can be used to determine the correct clustering of the voxels in a neighborhood around a point $\mathbf{x}$. This is done via minimizing the local energy of the k-means algorithm with respect to regions Ω_i, bias field b and tissue class values c_i:

$$E_x = \sum_{i=1}^{N} \int_{\Omega_i} K(\mathbf{x}-\mathbf{y}) |I(\mathbf{y}) - b(\mathbf{x})c_i|^2 d\mathbf{y}, \tag{4.3}$$

where K is a kernel function that is zero outside the neighborhood O_x. The choice of the kernel function is flexible but should satisfy the condition $\int K(\mathbf{u})d\mathbf{u} = 1$. The size of the neighborhood is determined by a radius ρ. This ρ depends on the degree of the intensity inhomogeneity. If the bias field shows large local fluctuations, a smaller neighborhood (a smaller ρ) is required to satisfy Equation 4.2. Otherwise a bigger ρ is necessary. The truncated uniform kernel is chosen as kernel K, which is given by:

$$K(r) = \begin{cases} \frac{1}{a}, & \text{for } |r| \le \rho \\ 0, & \text{for } |r| > \rho \end{cases} \tag{4.4}$$

where a is the normalization constant such that the condition $\int K(\mathbf{u})d\mathbf{u} = 1$ is satisfied.

To obtain the segmentation and the approximate bias field of the complete image, the energy given in Equation 4.3 should be jointly minimized for all $\mathbf{x}$ in Ω. Therefore, the energy that needs to be minimized is given as follows [Li et al., 2011]:

$$E = \int_{\Omega} \left(\sum_{i=1}^{N} \int_{\Omega_i} K(\mathbf{x}-\mathbf{y}) |I(\mathbf{y}) - b(\mathbf{x})c_i|^2 d\mathbf{y} \right) d\mathbf{x}. \tag{4.5}$$

This model can be easily extended to a level set algorithm. If the image consists of two disjoint regions, these regions can be defined by the level set function ϕ as $\Omega_1 = \{\mathbf{x} : \phi(\mathbf{x}) > 0\}$ and $\Omega_2 = \{\mathbf{x} : \phi(\mathbf{x}) < 0\}$. The energy (Equation 4.5) is rewritten by introducing a heavy-side function instead of an integral over region Ω_i:

$$E = \int_{\Omega} \left(\sum_{i=1}^{N} \int_{\Omega} K(\mathbf{x}-\mathbf{y}) |I(\mathbf{y}) - b(\mathbf{x})c_i|^2 d\mathbf{y} \right) u_i(\phi(\mathbf{x})) d\mathbf{x}, \tag{4.6}$$

where $u_i(\phi)$ is the membership function and i is the index of the different regions. Since the goal here is to segment the image in two regions, N is taken as 2. The membership functions are defined as $u_1(\phi) = H(\phi)$ and $u_2(\phi) = 1 - H(\phi)$ which represent respectively regions Ω_1 and Ω_2. As with the Chan-Vese level set method regularization terms are added to the energy that is minimized. One regularization term is a length term of the contour, which serves to smooth the contour by penalizing its arc length [Li et al., 2011]. This 2D term is comparable to the area term of the 3D Chan-Vese level set algorithm (Equation 3.3). The second regularization term is defined by a potential energy density function. In the article by Li et al. the potential function chosen is $p(s) = 1/2(s-1)^2$ [Li et al., 2011]. This potential energy term is added to the method to ensure that the level set function is not too steep or too flat (in a vicinity of its zero level set) during the level set evolution, which ensures that the level set evolution is stable and the numerical computations are accurate. The minimization of this potential energy density term maintains the signed distance property, which is $|\nabla\phi| = 1$, in a vicinity of the zero level set. The complete energy minimized in this model is given by the following equation (N=2):

$$E = \int_{\Omega} \left(\sum_{i=1}^{N} \int_{\Omega} K(\mathbf{x}-\mathbf{y}) |I(\mathbf{y}) - b(\mathbf{x})c_i|^2 d\mathbf{y} \right) u_i(\phi(\mathbf{x})) d\mathbf{x} + \nu \int_{\Omega} |\nabla H(\phi)| d\mathbf{x} + \mu \int_{\Omega} p(|\nabla\phi|) d\mathbf{x}, \tag{4.7}$$

where ν and μ are positive constants. Via the standard gradient descent method the corresponding gradient flow equation is written as follows (see Appendix B for the derivation)

$$\begin{cases} \frac{\partial \phi}{\partial t} & = -\delta(\phi)(e_1 - e_2) + \nu\delta(\phi)\text{div}\left(\frac{\nabla\phi}{|\nabla\phi|}\right) + \mu\text{div}\left(\frac{\nabla\phi p'(|\nabla\phi)|}{|\nabla\phi|}\right) \quad \text{in } \Omega, \\ \nu\frac{\delta(\phi)}{|\nabla\phi|}\frac{\partial\phi}{\partial\vec{\mathbf{n}}} + \mu\frac{p'(|\nabla\phi|)}{|\nabla\phi|}\frac{\partial\phi}{\partial\vec{\mathbf{n}}} & = 0 \quad \text{on } \partial\Omega, \end{cases} \tag{4.8}$$

with

$$e_i = I^2\mathbf{1}_K - 2c_i I(b * K) + c_i^2(b^2 * K). \tag{4.9}$$

In this equation $\mathbf{1}_K$ is defined as $\mathbf{1}_K = \int K(\mathbf{y}-\mathbf{x})d\mathbf{y}$, which is 1 everywhere except near the boundary of the image domain Ω. The boundary is given by $\partial\Omega$. For regularization the same regularized Dirac delta and Heaviside functions are used as in Equation 3.11. The numerical implementation is implemented using finite difference schemes, similar to the schemes used for the Chan-Vese level set method. The boundary condition is again a Neumann boundary. The explicit Euler method (Equation 3.13) is used for time discretization. A next iteration begins if ϕ has not converged or if the number of iterations has not reached the specified maximum number of iterations. In every iteration the energy is also minimized with respect to c_i and b. This means that the optimal values of these parameters are required every iteration. These optimal values are obtained by computing the values that ensure that the temporal derivative is zero. Using Euler-Lagrange the temporal derivative is computed through the derivative of the energy E to respectively value c_i or b. This is very similar to the derivation of the optimal values c_i for the Chan-Vese method, which can be found in Appendix A.2. The optimal values are given by the following equations:

$$c_i = \frac{\int (b * K) I u_i d\mathbf{y}}{\int (b^2 * K) I u_i d\mathbf{y}} \tag{4.10}$$

$$b = \frac{(I(\sum_{i=1}^{N} c_i u_i)) * K}{(\sum_{i=1}^{N} c_i^2 u_i) * K} \tag{4.11}$$

The complete algorithm is given in Algorithm 4.

Algorithm 4 Li's Level Set Algorithm

Initialize ϕ^0 and b^0
Set tol, ITER, and i=0
while $|\phi^{n+1} - \phi^n|$>tol and i<ITER **do**
  $\phi^{n+1} \rightarrow \phi^n$
  Compute c_1 and c_2 as the region average with the current ϕ^n and b
  Evolve ϕ^{n+1} with explicit time step
  Update the bias field b using the current c_1, c_2 and ϕ^{n+1}
  i=i+1
end while

Now, the segmentation with simultaneous bias correction can be applied on Figure 4.1. The initial bias field is set to 1 in every pixel. The initial level set function is -1 everywhere except for a small block of 20x40 pixels in the middle of the image where the level set function is 1. The result of the segmentation is given in Figure 4.4. The result of the segmentation shows a much better segmentation of the CSF regions. The middle regions are just as well segmented as the lower and upper regions. The contours of the segmentation look sharp. Contrasting with the Chan-Vese results little oversegmentation is visible and the 'thickening' effect is gone (compare to the red ovals in Figure 4.3c). Unfortunately, in this segmentation result a lot more fat (skin) is included, since the intensity is close to the intensity of CSF. However, such small errors are easy to correct, by, for example, taking a brain mask.

Figure 4.4: In this figure the segmentation result of Li's level set method is shown. The parameters are taken as follows: $\nu = 65.025$, $\mu = 1$, $\epsilon = 1$, $\sigma = 3$, $\rho = 10$ $dt = 0.1$ and the maximum number of iterations is 100.

4.2. T1-Weighted Scan

In the previous sections images were segmented that contained only CSF and background. However, the goal is to make an algorithm that segments CSF, GM, WM and background. Therefore, for this experiment a T1-weighted BrainWeb [Aubert-Broche et al., 2006] scan is used, which is shown in Figure 4.5. It is a 2D axial slice of an artificial MRI scan of the brain. The size of the image is 218x183 and the intensity of the image ranges from 0 to 255. The scan shows good contrast between WM, GM and CSF. Signal from fat (skin around the skull) is also visible in an oval around the brain tissue.

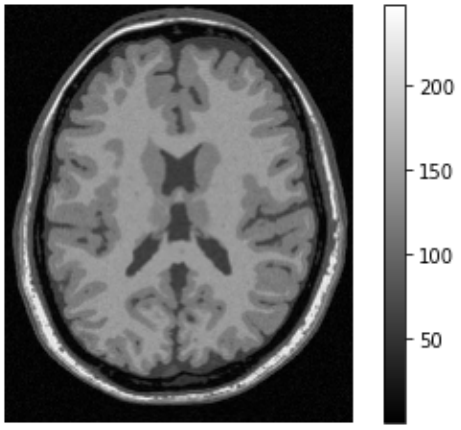


Figure 4.5: A 2D axial slice of a T1-weighted MRI BrainWeb image [Aubert-Broche et al., 2006]. The intensity value range is given in a colorbar on the right-hand side of the image

The segmentation method introduced in the last section could only separate two regions as it contained one level set function, which determined the two segmentation regions. Now four regions need to be segmented. Therefore, a multi-phase algorithm needs to be established. In a multi-phase algorithm multiple level set functions are introduced and solved every iteration. An example of a three-phase framework is shown in Figure 4.6. Here, a four-phase algorithm will be set up. The energy minimized in this algorithm is given by Equation 4.7 with N=4. This four-phase algorithm contains two level sets: ϕ_1 and ϕ_2. The membership functions are defined as $u_1(\phi_1,\phi_2) = H(\phi_1)H(\phi_2)$, $u_2(\phi_1,\phi_2) = H(\phi_1)(1-H(\phi_2))$, $u_3(\phi_1,\phi_2) = (1-H(\phi_1))H(\phi_2)$ and $u_4(\phi_1,\phi_2) = (1-H(\phi_1))(1-H(\phi_2))$ which divides the image in four regions. In this case, the gradient flow functions of the levels sets can be written as follows:

$$\begin{cases} \frac{\partial\phi_1}{\partial t} = -\delta(\phi_1)((e_1-e_3)H(\phi_2)+(e_2-e_4)(1-H(\phi_2))) + \nu\delta(\phi_1)\mathrm{div}\left(\frac{\nabla\phi_1}{|\nabla\phi_1|}\right) + \mu\mathrm{div}\left(\frac{\nabla\phi_1 p'(|\nabla\phi_1|)}{|\nabla\phi_1|}\right) \text{ in } \Omega, \\ \nu\frac{\delta(\phi_1)}{|\nabla\phi_1|}\frac{\partial\phi_1}{\partial\vec{\mathbf{n}}} + \mu\frac{p'(|\nabla\phi_1|)}{|\nabla\phi_1|}\frac{\partial\phi_1}{\partial\vec{\mathbf{n}}} = 0 \quad \text{on } \partial\Omega, \end{cases} \tag{4.12}$$

$$\begin{cases} \frac{\partial\phi_2}{\partial t} = -\delta(\phi_2)((e_1-e_2)H(\phi_1)+(e3-e4)(1-H(\phi_1))) + \nu\delta(\phi_2)\mathrm{div}\left(\frac{\nabla\phi_2}{|\nabla\phi_2|}\right) + \mu\mathrm{div}\left(\frac{\nabla\phi_2 p'(|\nabla\phi_2|)}{|\nabla\phi_2|}\right) \text{ in } \Omega, \\ \nu\frac{\delta(\phi_2)}{|\nabla\phi_2|}\frac{\partial\phi_2}{\partial\vec{\mathbf{n}}} + \mu\frac{p'(|\nabla\phi_2|)}{|\nabla\phi_2|}\frac{\partial\phi_2}{\partial\vec{\mathbf{n}}} = 0 \quad \text{on } \partial\Omega. \end{cases} \tag{4.13}$$

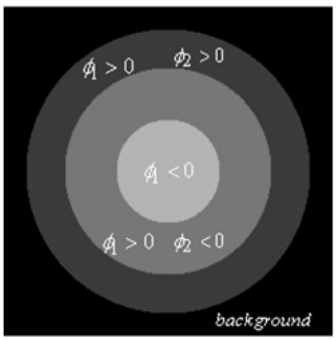


Figure 4.6: Example of the three-phase framework for the multi-phase level set formulation [Zhan et al., 2012]

The equations to compute element e_i, the bias field b and the tissue class value c_i remain the same as with the two-phase algorithm (see Equations 4.10 and 4.11). The complete algorithm is given in Algorithm 5.

Algorithm 5 Four-Phase Level Set Algorithm

Initialize ϕ_1^0, ϕ_2^0 and b^0
Set tol, ITER and i=0
while $|\phi_1^{n+1} - \phi_1^n| + |\phi_2^{n+1} - \phi_2^n| >$tol and i<ITER **do**
  $\phi_1^{n+1} \rightarrow \phi_1^n$ and $\phi_2^{n+1} \rightarrow \phi_2^n$
  Compute c_1, c_2, c_3 and c_4 as the region average with the current ϕ_1^n, ϕ_2^n and b
  Evolve ϕ_1^{n+1} and ϕ_2^{n+1} with an explicit time step
  Update the bias field b using the current c_i for $i = 1,2,3,4$ and ϕ_1^{n+1} and ϕ_2^{n+1}
  i=i+1
end while

There is no non-uniformity bias artifact present in this artificial scan. For both the k-means algorithm and Li's algorithm fat and skin around the brain are mistakenly put into one of the segmented regions. They can be removed through a brain mask, since we are only interested in a segmentation of the brain. The brain mask is shown in Figure 4.7d. The non-brain tissues are removed before the segmentation so that it does not influence the tissue class values. The results of the k-means algorithm and the four-phase algorithm applied to the T1-weighted image (Figure 4.5) are given in Figure 4.8. For the k-means algorithm four clusters were chosen with starting centroids of [0,1,2,100]. The final clusters have centroids of [6.4,70.1,122.0,159.0] (see Figure 4.8a). The four-phase algorithm starts with a bias field that is 1 everywhere. ϕ_1 is initialized with a small block of 20x40 pixels in the middle of the image that is 1 and the rest is -1. ϕ_2 is randomly distributed 1 and -1 over the whole image domain. The parameters were set as follows: $\nu = 1$, $\mu = 1$, $\epsilon = 0.5$, $\rho = 25$, $dt = 0.01$ and the maximum number of iterations is 100. The result is shown in Figure 4.8b. The segmentation of the brain looks fine, but at the corners of the image domain weird spots occur. This is due to the random initialization of ϕ_2 and the brain masking which sets the background values to 0. Therefore the bias field computed at those corners will be zero, which ensures that the optimal values c_i will not be influenced by pixels at the corners. These spots are removed through taking the the brain mask again. The segmentation results can be compared to the ground truth segmentations. The ground truth of the CSF, GM and WM are available from the BrainWeb website [Aubert-Broche et al., 2006] and are depicted in Figure 4.7. The ground truth segmentations take into account that a pixel can a combination of tissues due to partial volume effects. In that case, the intensities reflect the ratio of the tissue in the pixel. The k-means clustering algorithms gives binary results; the pixel is either in a region or outside of it. The segmentation results of Li's four-phase algorithm is due to the regularization of the Heaviside function also a continuous map. Therefore, the binary maps of the ground truths and Li's segmentation results are calculated, where at least 50% of the pixel contains tissue n to be considered inside the binary segmentation map of tissue n. The similarity between the segmentation and the ground truth are computed with the Dice score, which determines the overlap between two binary images. The Dice score is between 0 and 1 and when the Dice score is 1 there is complete overlap between the two images. Thus, when the Dice score is close to 1, the algorithm works extremely well. The

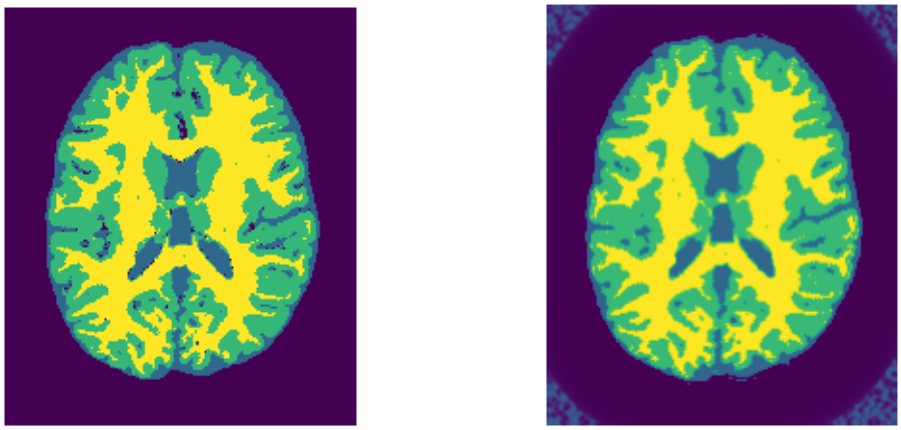
equation to determine the Dice score between binary image X and Y is as follows:

$$\text{Dice} = \frac{2|Y \cap X|}{|Y| + |X|}. \tag{4.14}$$

The Dice scores of the GM, WM and CSF between the k-means clustering algorithm and the ground truth and the four-phase algorithm and the ground truth can be determined. The overlapping Dice scores for the different tissues are given in Table 4.1. As is clear from the table, both methods work equally well, when there is no bias artifact.

(a) CSF ground truth (b) WM ground truth (c) GM ground truth (d) Brain mask

Figure 4.7: The ground truth segmentations are shown for the CSF, WM and GM in the left three pictures. They combine to a brain mask which is shown in the right picture.



(a) K-means clustering (b) Four-phase segmentation

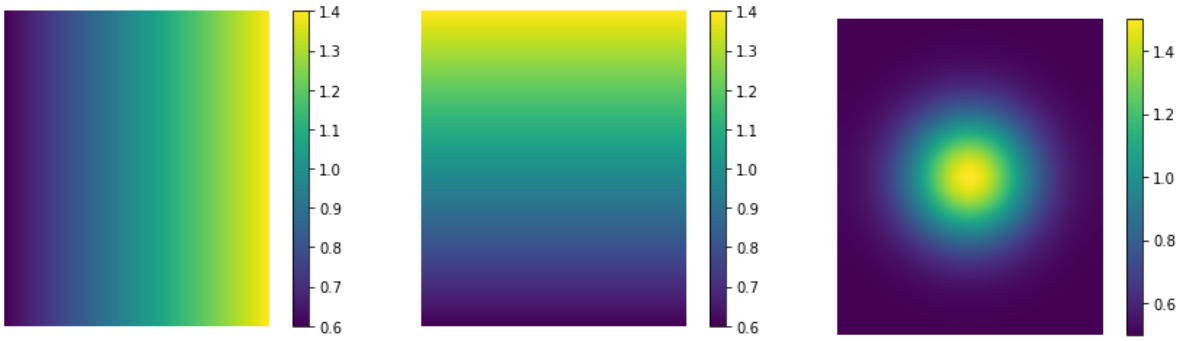
Figure 4.8: On the left-hand side the result of the k-means clustering algorithm is shown applied to a T1-weighted image. On the right-hand side the last iteration of the four-phase model is shown.

Table 4.1: Dice scores between the ground truth segmentations of the three tissue classes and the results of the k-means and four-phase algorithm after brain masking.

	k-means	Li
CSF	0.88	0.91
WM	0.96	0.95
GM	0.94	0.93

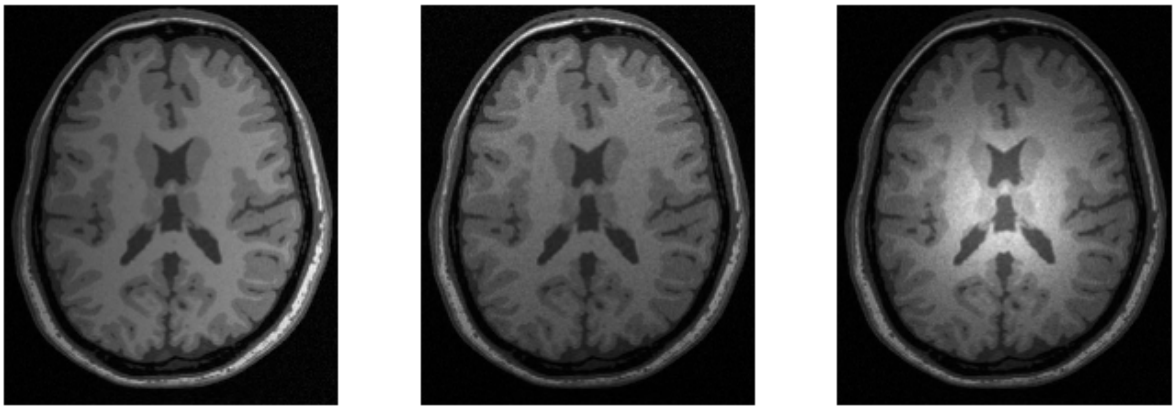
Now, an artificial bias field and noise are added to the image to understand the influence of the bias field and the noise on the segmentation result. First, random Gaussian noise is added to the image. Then, the image is corrupted by a bias field by multiplying an artificial bias field with the original image. Three different bias

fields were created so the influence of the bias field on the segmentation results could be tested in 3 experiments. The different artificial bias fields and the images affected by these bias fields are given in Figures 4.9 and 4.10. Three segmentation methods are tested on these experiments: the k-means clustering algorithm, the k-means clustering algorithm applied to a bias-corrected image (obtained by applying first SimpleITK's bias field correction algorithm [Beare et al., 2018]), and Li's method in which bias field correction and segmentation are integrated. This integration should improve the segmentation results. The Dice scores for the three tissue types between the segmentation algorithms and the ground truths are given in Table 4.2. The results of three algorithms for the three experiments are given in Figure 4.11. From Table 4.2 the Dice scores for all three experiments and for all three regions are the highest for the results of the four-phase segmentation model. Therefore, it can be concluded that the four-phase algorithm works better than k-means algorithm applied on the bias distorted image and on the image for which the bias is corrected for. Thus, Li's four-phase algorithm seems promising and will be investigated further by using it on real low-field scans.



(a) Bias field 1 (b) Bias field 2 (c) Bias field 3

Figure 4.9: The three artificial bias fields used in the three experiments are shown.

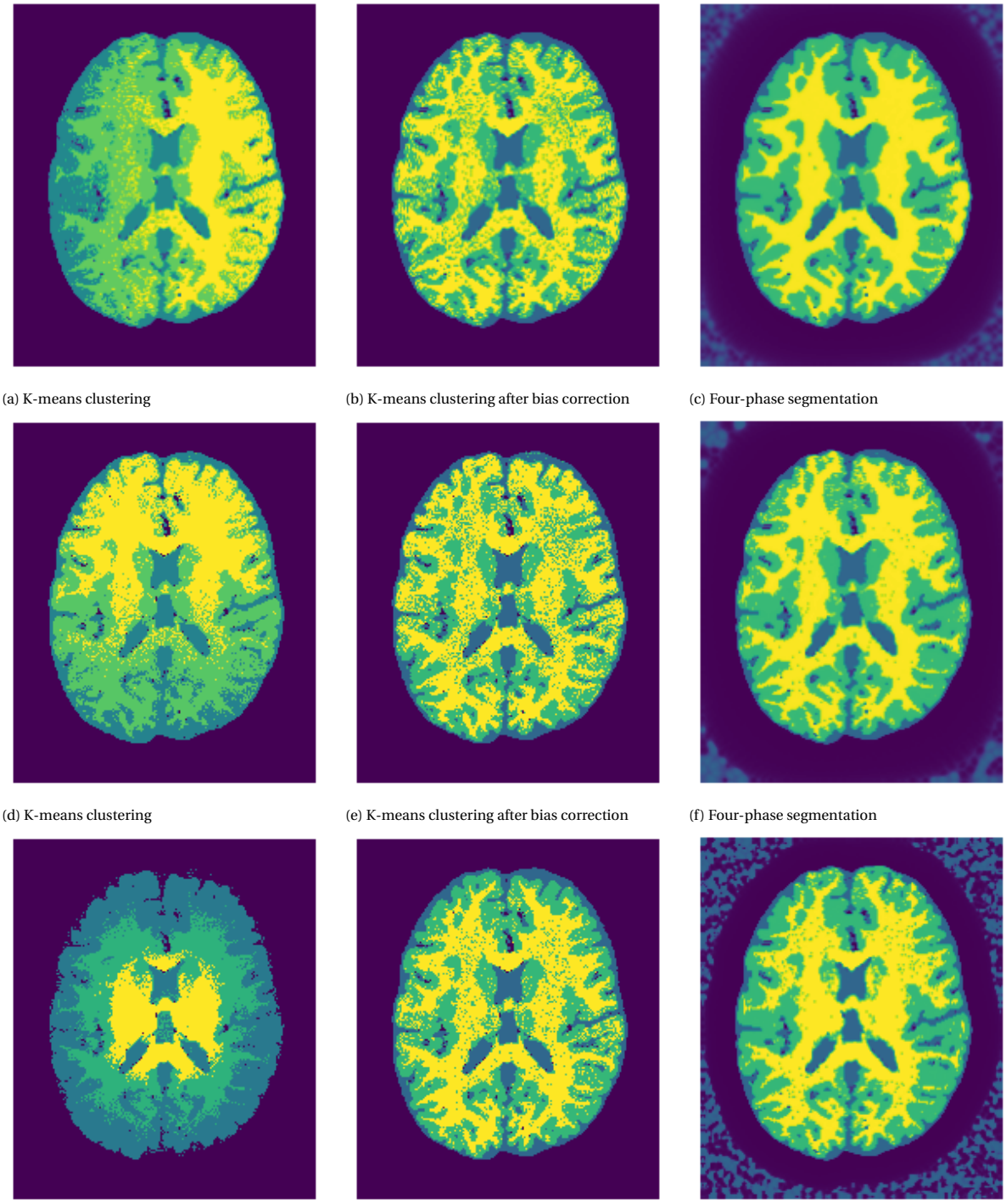


(a) Bias distorted image 1 (b) Bias distorted image 2 (c) Bias distorted image 3

Figure 4.10: The bias distorted images are shown corresponding to the three bias fields from Figure 4.9.

Table 4.2: Dice score between the ground truth segmentations of the three tissue classes and the results of the algorithms for the three experiments.

		K-means	K-means corr	Li
Experiment 1	CSF	0.72	0.93	0.92
	WM	0.63	0.80	0.93
	GM	0.58	0.78	0.91
Experiment 2	CSF	0.75	0.93	0.92
	WM	0.65	0.78	0.91
	GM	0.62	0.77	0.89
Experiment 3	CSF	0.31	0.93	0.91
	WM	0.59	0.82	0.89
	GM	0.11	0.80	0.87



(a) K-means clustering (b) K-means clustering after bias correction (c) Four-phase segmentation

(d) K-means clustering (e) K-means clustering after bias correction (f) Four-phase segmentation

(g) K-means clustering (h) K-means clustering after bias correction (i) Four-phase segmentation

Figure 4.11: Every row shows a different experiment. The images of the original scan distorted with the three different bias fields on which the experiments are based are given in Figure 4.9. The first column shows the results of the k-means clustering algorithm. The second column shows the results of k-means algorithm applied to the image for which the bias was corrected for. The third column shows the segmentation found by the four-phase algorithm. The parameters for Li's model for the first experiment are $\nu = 10$, $\mu = 5$, $\epsilon = 1$, $\rho = 20$, $dt = 0.01$ and the maximum number of iterations is 200. The second experiment has the following parameters: $\nu = 10$, $\mu = 5$, $\epsilon = 0.5$, $\rho = 20$, $dt = 0.01$ and the maximum number of iterations is 200. The parameters of the last experiment are $\nu = 1$, $\mu = 1$, $\epsilon = 0.1$, $\rho = 10$, $dt = 0.01$ with a maximum number of iterations of 200.

5

Segmentation of Low-Field Scans

In this chapter the segmentation methods discussed in Chapters 3 and 4 will be applied on scans from the low-field MRI scanner. To investigate if the results of the analytic segmentation technique for the low-field MRI scans are satisfactory, a phantom was scanned in the low-field MRI scanner. The results of the segmentation techniques applied on the phantom scans are shown and compared to the ground truth segmentation. Thereafter, for illustration, two low-field scans of real brains are segmented.

5.1. Phantom

The phantom is a 3D printed model of a brain, made by L. Bossoni and W. Teeuwisse. In Figure 5.1 the phantom is depicted with the different regions (CSF, WM, GM, and background) colored in. These regions can be compared to the results of the analytic segmentation methods applied to the scans.

Figure 5.1: Depiction of the phantom. WM is yellow, GM is green, CSF is light blue, and the background is dark blue.

To scan the phantom, the different regions of the phantom were filled with mixtures containing water, agarose and copper sulfate ($CuSO_4$). These mixtures were chosen such that they have signal intensities comparable to CSF, GM and WM. The different mixtures were 0.5% agarose for CSF, 1% agarose and 2.5mM $CuSO_4$ for GM and 1% agarose and 3mM $CuSO_4$ for WM. Then, the phantom was put upright into a brain coil which was put into the low-field MRI scanner. During scans of approximately 20 minutes 3D images were made of the phantom. In Figure 5.2 2D slices of three of the 3D scans are shown. Scans 1, 2 and 3 were made with a turbo spin echo sequence with a repetition time of respectively 1000 ms, 1000 ms and 1500 ms. During a turbo spin echo sequence multiple lines of k-space (i.e., phase-encoding steps) are acquired within a given repetition time, which decreases the imaging time [Bushberg, 2011]. The echo time was respectively 18 ms, 15 ms and 15 ms. As preprocessing, all scans were zero-padded and a Gaussian filter with standard deviation of 0.5 was applied. After preprocessing, the scans were of the size 128x128 pixels. Scan 2 and 3 were made in the same scanning session, scan 1 on a different day. In scan 1 and 2 a zipper artifact is visible. When resetting the system, this zipper artifact disappeared, after which scan 3 was made.

(a) scan 1 (b) scan 2 (c) scan 3

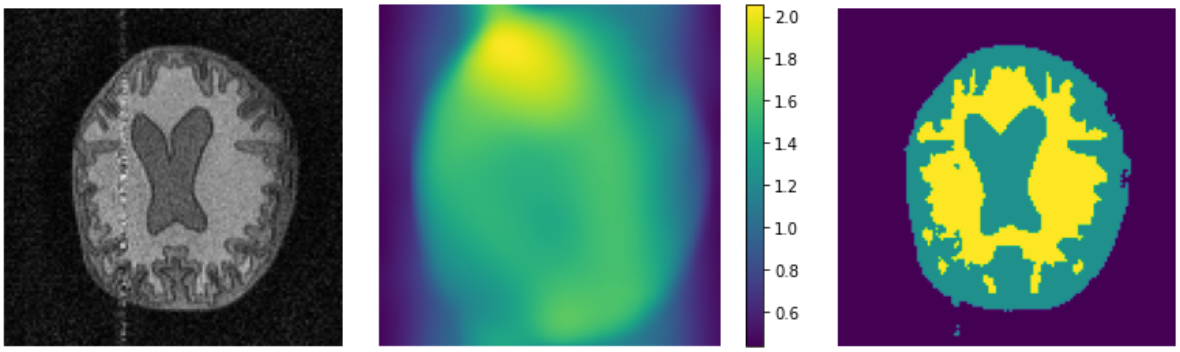
Figure 5.2: Three 2D slices of 3D scans of the phantom made with a turbo spin echo sequence in the low-field MRI scanner.

The goal is to segment these three scans into CSF, GM, WM and background. However, the difference in intensity between the GM region and the CSF region is very small. Also, the region of GM in itself is small compared to the regions of WM and CSF. Consequently, the GM region blends in with the CSF when applying the four-phase segmentation or k-means clustering with four clusters. Therefore, the scans are segmented into three regions: WM, GM+CSF and background. Thus, the multi-phase algorithm is rewritten such that the energy minimized in this algorithm is given by Equation 4.7 with N=3. The three-phase algorithm still contains two level sets: ϕ_1 and ϕ_2. However, the membership functions are different. They are defined as $u_1(\phi_1,\phi_2) = H(\phi_1)H(\phi_2)$, $u_2(\phi_1,\phi_2) = H(\phi_1)(1-H(\phi_2))$ and $u_3(\phi_1,\phi_2) = (1-H(\phi_1))$ which divides the image in three regions. In this case, the gradient flow functions of the levels sets can be written as follows:

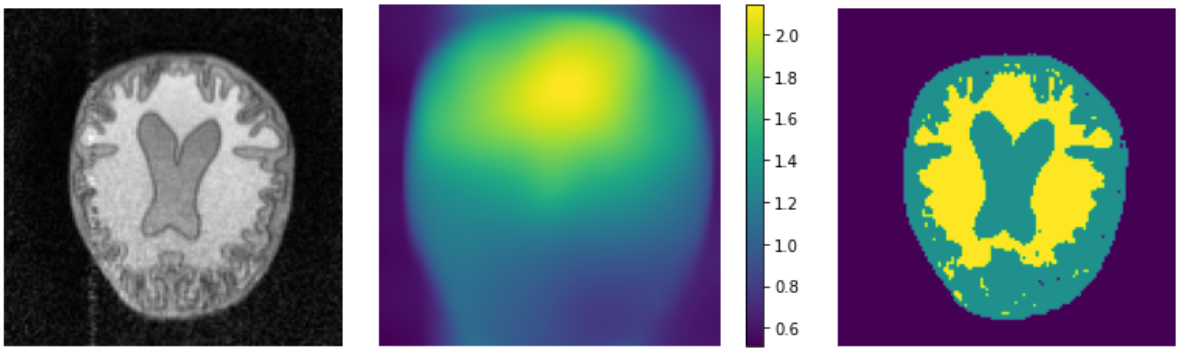
$$\begin{cases} \frac{\partial\phi_1}{\partial t} = -\delta(\phi_1)(e_1 H(\phi_2) + e_2(1-H(\phi_2)) - e_3) + \nu\delta(\phi_1)\mathrm{div}\Big(\frac{\nabla\phi_1}{|\nabla\phi_1|}\Big) + \mu\mathrm{div}\Big(\frac{\nabla\phi_1 p'(|\nabla\phi_1|)}{|\nabla\phi_1|}\Big) \text{ in } \Omega, \\ \nu\frac{\delta(\phi_1)}{|\nabla\phi_1|}\frac{\partial\phi_1}{\partial\vec{\mathbf{n}}} + \mu\frac{p'(|\nabla\phi_1|)}{|\nabla\phi_1|}\frac{\partial\phi_1}{\partial\vec{\mathbf{n}}} = 0 \quad \text{on } \partial\Omega, \end{cases} \tag{5.1}$$

$$\begin{cases} \frac{\partial\phi_2}{\partial t} = -\delta(\phi_2)((e_1-e_2)H(\phi_1)) + \nu\delta(\phi_2)\mathrm{div}\Big(\frac{\nabla\phi_2}{|\nabla\phi_2|}\Big) + \mu\mathrm{div}\Big(\frac{\nabla\phi_2 p'(|\nabla\phi_2|)}{|\nabla\phi_2|}\Big) \text{ in } \Omega, \\ \nu\frac{\delta(\phi_2)}{|\nabla\phi_2|}\frac{\partial\phi_2}{\partial\vec{\mathbf{n}}} + \mu\frac{p'(|\nabla\phi_2|)}{|\nabla\phi_2|}\frac{\partial\phi_2}{\partial\vec{\mathbf{n}}} = 0 \quad \text{on } \partial\Omega. \end{cases} \tag{5.2}$$

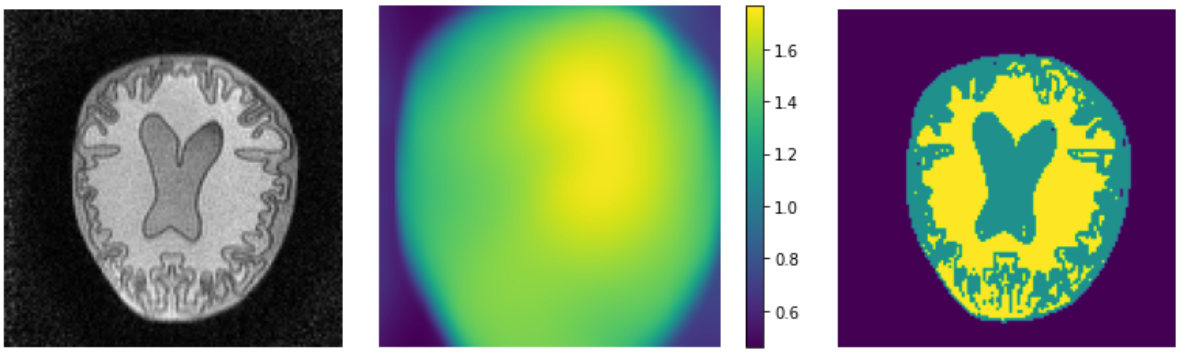
The rest of the algorithm remains the same as the four-phase algorithm. The results of implementing the segmentation algorithm on the three scans are shown in Figure 5.3. For all scans the bias field, the bias-corrected image, and the segmented image are shown. In the next section the quality of these results are checked.



(a) Bias-corrected image of scan 1 (b) Estimated bias field of scan 1 (c) Three-phase segmentation result of scan 1



(d) Bias-corrected image of scan 2 (e) Estimated bias field of scan 2 (f) Three-phase segmentation result of scan 2



(g) Bias-corrected image of scan 3 (h) Estimated bias field of scan 3 (i) Three-phase segmentation result of scan 3

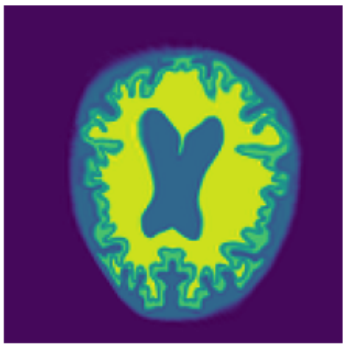
Figure 5.3: Every row shows the results of the three-phase segmentation model for one of the three different scans shown in Figure 5.2. The first column shows the bias-corrected image. The second column shows the estimated bias field. The third column shows the segmentation found by the three-phase algorithm. The parameters for the three-phase model for the first scan are $\nu = 1$, $\mu = 2$, $\epsilon = 0.1$, $\rho = 10$, $dt = 0.05$ and the maximum number of iterations is 200. The second scan has the following parameters: $\nu = 10$, $\mu = 10$, $\epsilon = 0.2$, $\rho = 10$, $dt = 0.005$ and the maximum number of iterations is 200. The parameters of the last scan are $\nu = 3$, $\mu = 3$, $\epsilon = 0.25$, $\rho = 20$, $dt = 0.005$ with a maximum number of iterations of 200.

5.1.1. Registration Phantom

To compare the improvement of the three-phase segmentation method for low-field scans against other methods, the results of the segmentation methods are checked against the ground truth (GT) image shown in Figure 5.1. The results of the segmentation algorithm and this GT image cannot be compared one to one, since the size of the phantom in the picture is different to the size in the scans. Also, deformities are introduced in the scans. To compensate for such effects, the GT image is registered to the scans. This is done via a program called SimpleElastix, which is a toolbox for rigid and nonrigid registration of images [Klein* et al., 2010], [Shamonin et al., 2014]. This program automatically registers a moving image to a specified fixed image. For this to work, the images need to have the same dimensions and they should be similar in some aspects as the moving image needs corresponding features to eventually match the target image. Using the original scan as the target image did not work. This was because the difference between features was too big due to the bias field. Therefore, the bias-corrected images were chosen as target images (see Figure 5.3a

and 5.3g) and the phantom depiction as the moving image (see Figure 5.1). First, rigid registration was performed using translation, so that the images would be closer to each other. Then, the moving image was registered using a B-spline transform, which is a nonrigid registration method. The registrations obtained from this method are shown in Figure 5.4. The Dice scores between the results and the registered GTs are shown in Table 5.1. The Dice scores are computed for k-means clustering applied to the original image, k-means clustering applied to the image, for which the bias was automatically corrected by using SimpleITK's bias field correction algorithm [Beare et al., 2018] and the results from the three-phase level set algorithm. The results of the first two algorithms are shown in Appendix C. For both the WM and the GM/CSF regions the three-phase algorithm performed the best in the first two scans. These are the scans where the bias field is the most prominent. For the third scan the best results were obtained by using k-means clustering on the original scan, closely followed by the results of the three-phase method. Therefore, it can be concluded that the three-phase method is a very promising segmentation technique for the low-field scanner when there is a clear bias field. Otherwise a simple k-means clustering technique would give comparable results.

(a) Registered GT to scan 1



(b) Registered GT to scan 3

Figure 5.4: The registered images of the phantom ground truth to scan 1 on the left side and scan 3 on the right side. The registration is performed via SimpleElastix [Klein* et al., 2010],[Shamonin et al., 2014]

		k-means	k-means cor	Three-phase
Scan1	WM	0.90	0.86	0.96
	GM+CSF	0.87	0.84	0.94
Scan 2	WM	0.67	0.82	0.88
	GM+CSF	0.75	0.83	0.88
Scan 3	WM	0.87	0.83	0.85
	GM+CSF	0.87	0.82	0.85

Table 5.1: Dice scores between the ground truths obtained from the registered images from Figure 5.4 and the segmentation results from k-means clustering, k-means clustering after correcting for the bias field and the three-phase algorithm for the three different scans.

5.2. Brain Scans

In this section two brain scans made with the low-field MRI scan are segmented. One scan contains more detail but does not have contrast within the brain tissue. The other scan contains more contrast within the brain tissue, but the scan has a low SNR. Therefore, the four-phase algorithm is applied to the low SNR scan and the three-phase algorithm is employed on the low contrast scan. The results in this section are just to illustrate how the segmentation algorithms can be applied to different scans, since there is no ground truth available.

5.2.1. Low Tissue Contrast Scan

The scan from Figure 5.5 shows low contrast between brain tissue, but the SNR is good. Because of the low contrast, a 3D implementation of the three-phase algorithm and not the four-phase algorithm is applied to the scan, which segments the scan in three regions: CSF, brain tissue, and background. The segmentation result is shown in Figure 5.7 and the bias-corrected image is displayed in Figure 5.6. It seems that the bias

field is estimated well and the simultaneous segmentation looks adequate. Skull, fat and skin are put into the brain tissue region, because of the similar intensity. Brain masking should be implemented to remove these tissues.

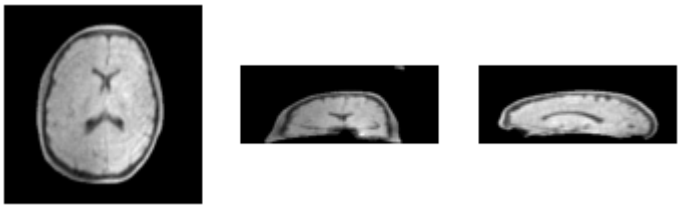


Figure 5.5: Axial, frontal and longitudinal slices of the original T1-weighted scan.

Figure 5.6: Axial, frontal and longitudinal slices of the T1-weighted scan corrected for the estimated bias field.

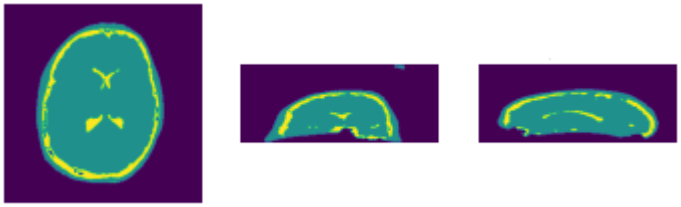


Figure 5.7: Axial, frontal and longitudinal slices of the segmented T1-weighted scan. The parameters are taken as follows: $\nu = 1$, $\mu = 1$, $\rho = 10$, $\epsilon = 0.01$, $\rho = 10$, $dt = 0.1$ and the maximum number of iterations is 100.

5.2.2. Low SNR Scan

This is a scan made during an inversion recovery sequence. From the scan, Figure 5.8, it is clear that some contrast between WM, GM and CSF is present. However, the scan has a very low SNR, which hinders the segmentation. The scan is segmented with a 3D implementation of the four-phase segmentation model. The segmentation result is shown in Figure 5.9. There is a clear distinction between WM, GM, CSF and the rest, but, due to the very low SNR level, background noise is segmented wrongly into the CSF region. This result could definitely be improved by making a brain mask, so that the background noise, the skull, fat and other tissues do not influence the result.

Figure 5.8: Axial, frontal and longitudinal slices of the original T1-weighted scan.

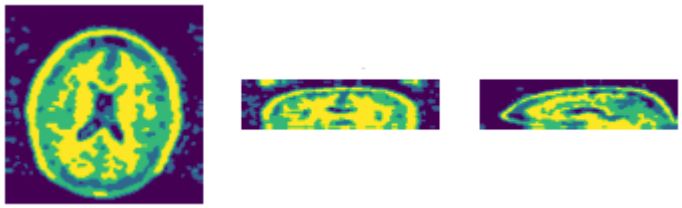


Figure 5.9: Axial, frontal and longitudinal slices of the segmented T1-weighted scan. The parameters are taken as follows: $\nu = 10$, $\mu = 1$, $\epsilon = 0.1$, $\rho = 10$, $dt = 0.1$ and the maximum number of iterations is 100.

6

Machine Learning for Segmentation

In the last chapters some well known segmentation methods for brain tissues were discussed and some results of applying these techniques to low-field MR images were shown. One technique that was omitted was segmentation by machine learning techniques. This method is explored in the following chapters, since segmentation by machine learning easily satisfies the requirement of automation, in contrast to analytic segmentation techniques. In this chapter some basic concepts of machine learning and neural networks will be introduced. Then, several examples of algorithms used for the segmentation of brain images are discussed. Furthermore, different elements of a neural network used for segmentation purposes are explored and possible ways to improve segmentation by neural networks are considered.

6.1. Machine Learning

Machine learning is an upcoming field, which is applied more and more in image processing. Machine learning is seen as part of artificial intelligence. Essentially, machine learning algorithms are algorithms that improve automatically through experience [Mitchell, 1997]. Machine learning algorithms build a model based on certain training data. After training, the model makes predictions or decisions by itself based on input of new data. The performance of machine learning algorithms depends heavily on the representation of the input data. The user therefore decides the features that the algorithm is going to learn [Goodfellow et al., 2016]. The algorithms used in machine learning are networks. Some of these networks model a brain, in which information is transmitted via synapses to other neurons (layers in neural network). For this reason, such machine learning algorithms are called neural networks. Deep learning is a class of machine learning algorithms that employs multiple layers to progressively extract higher-level features from the raw input [Deng and Yu, 2014]. The name deep learning is derived from the use of the (multiple) hidden layers. A sketch of a deep learning neural network is shown in Figure 6.1. Deep learning allows computers to learn difficult concepts based on simpler concepts by utilizing the hidden layers. For example, a deep learning model can represent an image by combining simpler concepts, such as corners and contours, which are in turn defined by configurations of edges. In such a way, the computer decides the features (simpler concepts) which it learns [Goodfellow et al., 2016]. Deep learning has become more popular in image segmentation problems since they showed exceptional performance on many image processing tasks [Sultana et al., 2020].

6.2. Neural Networks

As mentioned in the last paragraph the name neural network comes from the manner in which the artificial algorithm mimics the process of how information is passed through in the brain. A neural network consists of multiple connections, called neurons, that are linked with each other. The input is passed into the neuron, whereafter the neuron performs a linear transformation. The output of the neuron is the activation function applied to the linear transformation. Thus, the output is given by the following equation:

$$Y = \text{Activation}\Big(\sum(wI + b)\Big), \tag{6.1}$$

where Y is the output of the neuron, I the input, w the weights of the neuron and b the bias. The weights and bias are not features chosen by the user of the neural network but they are automatically adapted during training of the model. Data is put in the model and at the end output is generated by the model. This is

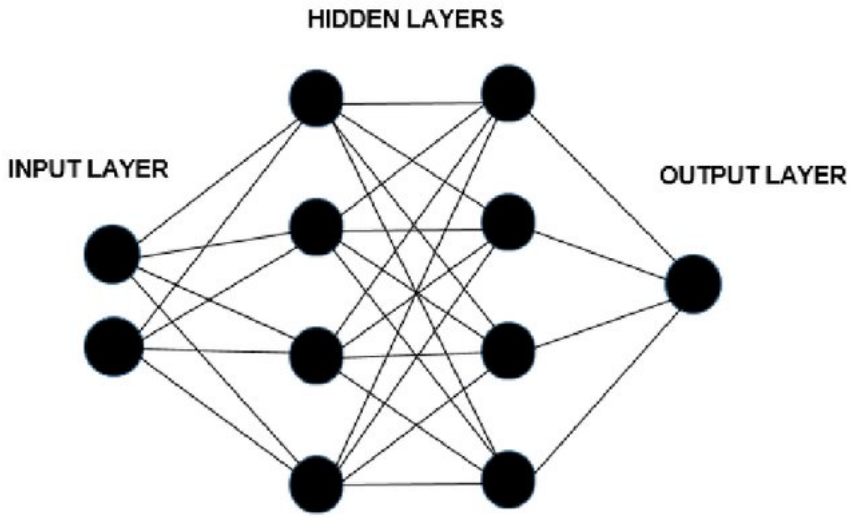


Figure 6.1: Sketch of a deep neural network [Bartzatt, 2014].

called forward propagation. During the training stage of the neural network the output of the neural network is compared to the ground truth. How the output and the ground truth are compared is defined by the loss function or error function. The user decides which requirements the output should satisfy and hereby decides on the loss function. Thus, if the model did not make the right choices after the image went through the network, a non-zero loss is found. This non-zero loss is backpropagated through the neural network, such that the weights and biases of the neurons are updated. This backpropagation is achieved via gradient descent, which is a numerical procedure to adjust the weights and biases such that the loss is minimized. The gradient is the derivative of the loss function with respect to all network parameters. This gives us the slope of the loss function for the current model parameters. If the model is updated in the decreasing direction, the loss function should decrease. Therefore, the name of this process is called gradient descent. This means that per layer the partial derivatives are computed. The process starts with the final layer, which has parameters that do not influence any other layers. The partial derivatives with respect to the parameters of this layer are computed and put into the gradient vector. Then the partial derivatives of the second last layer are computed by utilizing the chain rule. This process advances until the partial derivatives of all layers are determined. Then one step of gradient descent is taken by subtracting the gradient vector times a learning rate from the current parameters. This learning rate determines the step size of the gradient descent method. Thereafter, the input goes into the model again. A new output is generated, from which a new loss is computed. This loss is backpropagated and so the training proceeds [Nielsen, 2015].

The activation function decides, based on the input of the neuron, if information is passed through the neuron and if so in what way. If the activation function was not there a neural network would consist only of linear transforms. Thus, only linear patterns could be learned by the model. The activation function introduces non-linearity in the network and therefore the ability to learn complex patterns, which is very useful [Nielsen, 2015]. There are different types of activation functions. The simplest one is just a step function, where the output is 1 if the input is above a certain threshold and 0 otherwise. This is a good activation function for a binary classifier. However, the gradient is zero everywhere except for at the threshold. A zero gradient causes issues with backpropagation, since the weights and biases are not updated for those neurons. This creates 'dead' neurons, which are neurons that are not active and cannot be activated anymore [Nielsen, 2015]. Other activation functions take care of this problem, but they also come with their own issues. Depending on the type of problem a choice of activation function must be made. Rectified linear unit (ReLU) is the most generally used activation function in neural network nowadays (see Figure 6.2) [Dubey and Jain, 2019]. The reason for its popularity is that it offers better performance and overall faster computation than other activation functions. Also, there is no saturation problem, which means that the backpropagation is efficient because there are no exploding gradients. However, dead neurons can arise in the network, since the gradient is zero for negative input. This can be prevented by adjusting the ReLU function [Dubey and Jain, 2019]. For example, LeakyReLU is an adaptation of ReLU which can prevent dead neurons as the gradient is

not zero anymore for negative input (Figure 6.2).

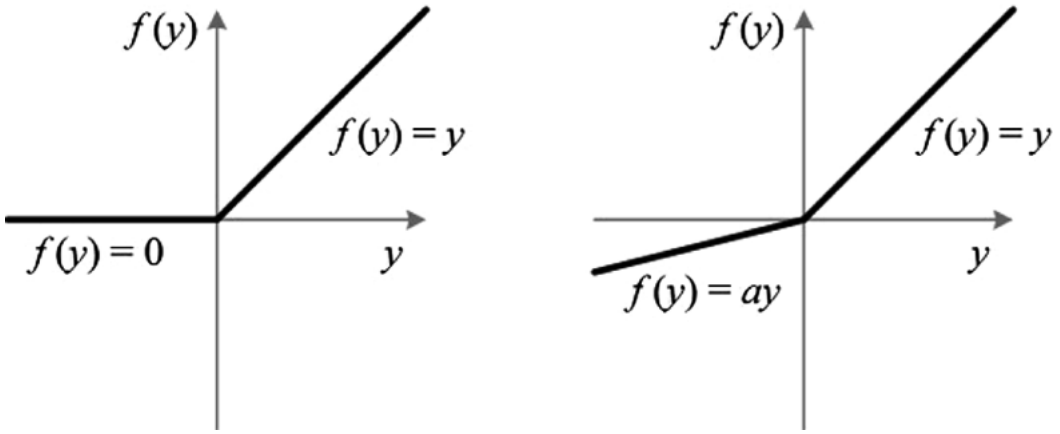


Figure 6.2: Sketch of the ReLU activation function on the left-side and of the LeakyReLU activation function on the right-side [Dubey and Jain, 2019].

6.3. Current Machine Learning Methods for Brain Segmentation

Many researchers have segmented brain scans via machine learning algorithms. Some examples are stated in this section. In the article by Klimont et al. a deep learning method was used to segment the ventricular system in pediatric patients [Klimont et al., 2019]. These researchers proposed the application of U-Net convolutional neural network (CNN) for the segmentation. U-Net is a widely-used neural network that has proven successful for different interdisciplinary tasks [Klimont et al., 2019]. U-Net was optimized for their case, such that with a limited amount of data (80 CT scans), the experiment showed near human-level performance. The mean Dice score, computed by the overlap between the automatic segmentation mask and the manually segmented mask was for the training set and the test set over 0.91 [Klimont et al., 2019]. The data used in this study was a dataset of CT scans of patients with hydrocephalus in the age group of 0 to 18.

Another deep learning algorithm for segmentation of CSF in patients with pediatric hydrocephalus was proposed in the article by Grimm et al. [Grimm et al., 2020b]. Here they used FSL to generate four classes of segmentation: brain, CSF, background, and tissue. These were used as the reference image segmentation input in a pre-initialized CNN (VGG16 [Simonyan and Zisserman, 2014]). Their paper examined if the CNN could be trained with the generated segmented data and could deliver reliable segmentation results. They also investigated if the CNN was capable of producing suitable segmentation results with test data with a known ground truth. The training was performed with 95 3D MR images, which were put into the FSL tool for brain extraction and segmentation. The slices of these 95 MR images were divided into the training (60%), test (20%), and validation set (20%). Additional test set data was selected from the BrainWeb dataset [Aubert-Broche et al., 2006] of which the ground truth is known. The segmentation from the CNN of the clinical dataset and the BrainWeb dataset showed comparable results (Dice coefficient of 0.86 vs 0.84). This is also comparable to segmentation of the BrainWeb dataset with FSL (Dice coefficient of 0.87). Thus, the transfer from using the FSL tool to a deep learning network seemed very promising.

In the article by Quon et al. a fully automated deep learning model was created for pediatric cerebral ventricle segmentation and volume calculation [Quon et al., 2020]. They adapted a 2D U-Net neural network to automatically segment the cerebral ventricles. The dataset consisted of 400 T2-weighted scans containing 200 obstructive hydrocephalic cases and 200 cases in the control group. The study cohort was divided into 266 scans for training, 67 for validation, and 67 for testing. Apart from these scans, 9 scans from another institute were taken as a proxy for testing model performance with data from other hospitals. Before the training the data was manually segmented to generate a ground truth reference. The model had a overall Dice score of 0.901. The model was more accurate and faster than FreeSurfer [Quon et al., 2020].

6.4. Neural Network for Segmentation

Convolutional neural networks (CNNs) have shown great success in the field of segmentation [Sultana et al., 2020]. CNNs consist of multiple layers. The two most important blocks that are usually in a CNN are convolutional layers and pooling layers. In convolutional layers a kernel with chosen size is convoluted with the input. The output is called a feature map. These feature maps contain lower-level information, like edges. Some of these convolutional layers downsample the data. Therefore, spatial information is obtained at different levels. These different levels are important because deeper layers can extract features that are combinations of lower-level features extracted from the raw image. This means, for example, that multiple lines extracted from the lower-level features can be combined to express shapes [Goodfellow et al., 2016]. The number of feature maps obtained per convolutional layers is determined by the user. Besides the kernel size and the number of feature maps created, the user can choose multiple variables like the stride size (stride of the convolution along each dimension), the padding (determines output dimensions) and the activation. Such aspects are referred to as the hyperparameters that determine the architecture of the network.

After some convolutional layers pooling layers are introduced. These pooling layers, as the name suggests, pool a patch of feature maps into one single neuron. There are various ways to pool, but the most commonly used ones are maxpooling and average pooling. With maxpooling the maximum value of the neuron cluster is taken and with average pooling the average value. Due to pooling the representation becomes invariant to small translations in the input. This improves the statistical efficiency of the network [Goodfellow et al., 2016]. Like in the convolutional layer, the user chooses the stride size and the padding.

The fully convolutional networks (FCN) like AlexNet [Krizhevsky et al., 2012], VGGNet [Simonyan and Zisserman, 2014], and GoogleNet [Szegedy et al., 2014] are deep networks that downsample the data till in the end the data is classified [Sultana et al., 2020]. FCNs only employ locally connected layers like convolutional and pooling layer and not fully connected layers. This ensures fewer trainable network parameters, making the network train faster [Shelhamer et al., 2017]. In the case of segmentation the input has the same dimensions as the output, since the output is the segmentation of the input image. By using only a FCN the input is downsampled and thus the output will not have the correct dimensions. A smart approach to solve this problem is to construct a neural network out of an encoder and a decoder. The encoder is a FCN and the decoder is the opposite of the encoder. In the decoder the data is upsampled. The encoder path is used to extract the lower-level features and connect them such that context can be interpreted. The decoder path allows for the localization of the context such that a segmentation is found in the end [Shelhamer et al., 2017]. U-net is the most popular example of an encoder-decoder network [Klimont et al., 2019], [Quon et al., 2020].

Most of these encoder-decoder neural networks have skip connections. Skip connections connect two layers in the neural network of the same dimension that are not directly connected in the neural network. The connections therefore skips one or more layers in between them. The introduction of skip connections in deep neural networks have greatly improved their training [He et al., 2016]. One of the things skip connections improve on is the problem with vanishing gradients in deep neural networks. The vanishing gradient problem exists due to the backpropagation of the error through the network. It was already mentioned that the activation functions play an important role in ensuring that neurons keep being trained and do not become dead neurons. But even if the activation of neurons works well it can still happen that the neuron becomes inactive. This is due to the amount of layers in a deep neural network. The error at the end of the network is backpropagated through all the layers. But as one can imagine the weights of the layers at the end of the neural network get updated by larger factors than the layers close to the input layer. This causes the gradient to become smaller and smaller when going back through the network and the neurons near the input of the network can therefore still die [Drozdzal et al., 2016]. This can be solved by skip connections, since they connect a layer at the beginning and at the end of the neural network, which provides an alternative path to the gradient [He et al., 2016]. Another reason to use skip connections is the forward propagation. The initial layers in the neural network capture information from the input image that is of value for the final layers. By connecting the layers with skip connections the lower semantic information is available in the final layers [He et al., 2016]. Spatial details that have been lost during pooling can be restored via these skip layers [Hussain et al., 2020]. For an encoder-decoder neural network it appeared that more fine-grained details could be recovered in the prediction by using skip connections [Drozdzal et al., 2016]. Even though there is no theoretical justification, experimentally it has been shown that long symmetric skip connections work very well in deep learning algorithms for so called dense prediction tasks, like image segmentation [Drozdzal et al., 2016].

6.5. Overfitting

With limited training data overfitting can occur in deep neural networks. Overfitting means that the neural network becomes more and more tuned to the training data during the training stage and will therefore give unsatisfactory results on new data [Shorten and Khoshgoftaar, 2019]. There are many methods to reduce overfitting. One is increasing the amount of training data. Another method to reduce overfitting is adapting the training strategy. This can be done by adding dropout layers to the network during training. The term "dropout" refers to dropping out units (hidden and visible) in a neural network [Srivastava et al., 2014]. By dropping out a unit, a node with its complete input and output is removed. The choice of removing the node is random in each training step. When part of the neural network drops out other neurons will have to step in and handle the representation required to make predictions for the missing neurons. Then after many training steps, the neural network is a combination of all the thinned neural networks that have originated from the dropout layers. This has the effect that the network becomes less sensitive to the specific weights of neurons and is therefore less likely to overfit on the training data. Dropout can be interpreted as a way of regularizing a neural network by adding noise to its hidden units [Srivastava et al., 2014].

Another approach to decrease overfitting, is by increasing the training data. Unfortunately, more data is not always available. In those cases data augmentation is performed. Data augmentation encompasses a suite of techniques that increase the size and quality of training datasets [Shorten and Khoshgoftaar, 2019]. Data augmentation artificially inflates the training dataset size by either data warping or oversampling. Data warping consists of augmentations that perform transformations on the data space to generate additional samples [Wong et al., 2016]. Oversampling creates synthetic instances and adds them to the training dataset. Examples of data warping are: rotation, flipping, cropping, translation, noise injection, and filtering [Shorten and Khoshgoftaar, 2019].

6.6. Loss functions

The previously mentioned machine learning algorithms that segment the brain were all trained by computing the MSE or the Dice score (see Equation 4.14) as the error function. The MSE measures the average of the squares of the errors between each voxel in the output image (Y_{pred}) and the ground truth image (Y_{true}) and is given by:

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}(Y_{\text{truth},i} - Y_{\text{pred},i})^2, \tag{6.2}$$

where N is total amount of voxels in the image and i is the index that goes over all the voxels of the image.

For most cases the ground truth is obtained by manually segmenting the region of interest. This is a tedious, time-consuming task and the segmentations are affected by error due to subjective bias. Alternatively, some papers have used new loss functions that incorporate classic segmentation techniques, some of which are mentioned in chapter 2. This type of segmentation is called unsupervised segmentation, as the segmentation is extracted without use of ground-truth segmentation masks [Kim and Ye, 2020]. In the paper by Chen et al. a convolutional neural network was proposed to segment the ventricles and the myocardium of MRI heart images [Chen et al., 2019]. The neural network was trained with three different loss functions: Dice score, cross entropy and an AC (active contour) loss function. The cross entropy loss function is a measure of the difference between two probability distributions of the voxels of the output segmentation and the ground truth segmentation. The AC loss was a loss function created by the authors. This loss function was based on active contours. It does not need a ground truth segmentation but only used the information of the original image. The AC loss incorporated area and contour size information into the neural network, like the Chan-Vese level set segmentation method. The paper showed that the AC loss function outperformed the other loss functions [Chen et al., 2019]. Hatamizadeh et al. proposed another CNN trained with a loss function that depended on the level set energy [Hatamizadeh et al., 2019].

As was discovered in Chapter 4 the bias field has a substantial influence on the analytic segmentation methods. This is of course also the case if such an AC loss is introduced in the loss function of a neural network. Therefore, segmentation can be improved by also adjusting for the bias field. This bias field can be extracted from the input image by the neural network. Kim and Ye proposed such a method for a completely unlabeled dataset [Kim and Ye, 2020]. When semantic labels were available the loss function was a combination of a Mumford-Shah functional (active contour segmentation) and a cross entropy loss between the labels and the

output. This cross entropy was only computed if the label for that image was present. If no labels were available Li's segmentation loss (Equation 4.3) was used as the loss of the neural network. Accordingly, the neural network simultaneously estimated the bias field and segmentation of the input image by minimizing this loss function. Eventually, the bias and the bias-corrected segmentation mask could be obtained from the same neural network [Kim and Ye, 2020].

A CNN can also be improved by integrating in prior anatomical knowledge. The CNN then encourages the segmentation to follow anatomical knowledge of the underlying anatomy [Oktay et al., 2017]. In the paper by Jurdia et. al. high level priors were reviewed [Jurdia et al., 2020]. This is prior information incorporated at the loss function level of the neural network. The priors were categorized as follows: object shape, size, topology and the inter-region constraints.

6.7. Transfer Learning

Transfer learning is the improvement of learning in a new task through the transfer of knowledge from a related task that has already been learned [Torrey and Shavlik, 2010]. Transfer learning might improve learning in three ways: the initial performance of the target task is better using transferred knowledge, the neural network learns faster from transferred knowledge than from scratch and the final performance in the target task is better [Torrey and Shavlik, 2010]. The need for transfer learning occurs when there is a limited supply of target training data [Weiss et al., 2016]. This is especially the case for data from the low-field scanner.

7

Neural Network for Segmentation of MRI Scans

In this section we go into the neural network, built for the segmentation of MRI scans. This neural network is based on a U-Net, which is conform the theory from Chapter 6. A branch is added to the U-Net that extracts the bias field from the input image. The reason for the bias field branch is the incorporation of an unsupervised bias-correcting segmentation loss in the loss function of the neural network. In this way analytic segmentation is integrated into a neural network. This is done because Li's method showed promising results for the low-field scans, but automatic segmentation by Li's method is complex due to the many parameters. This issue of automation is solved by using a neural network, since, after training, the neural network segments images automatically. Furthermore, integrating Li's method ensures that the energy of the segmentation is minimized during training, which could improve the segmentation results. In this chapter the network architecture and the loss function will be delved into. Also, the data used to train this network is discussed.

7.1. Network

7.1.1. Framework

Assume that x is the 3D input image, y_n is the ground truth 3D segmentation of tissue n and b is the ground truth 3D bias field. Suppose that $\bar{y}_n$ is the 3D segmentation of tissue n predicted by the neural network and $\bar{b}$ is the predicted 3D bias field. x is an MR image affected by an artificial bias field b. x is obtained by multiplying b with MR image z: $x = bz$. z and its corresponding segmentation y_n are obtained from the artificial BrainWeb dataset [Aubert-Broche et al., 2006]. The segmentation images are normalized. The values of the segmentation images are continuous between 0 and 1. This continuity is to take the partial volume averaging artifact into account: a voxel $\mathbf{r}$ contains only tissue n if $y_n(\mathbf{r}) = 1$, it does not contain tissue n if $y_n(\mathbf{r}) = 0$ and if $y_n(\mathbf{r})$ is between 0 and 1 the voxel $\mathbf{r}$ partially contains tissue n. Prediction $\bar{y}_n$ has the same property. Artificial bias field b is randomly constructed by making a random Gaussian filtered field between 0 and 2 with a mean of 1. After construction, b is multiplied with z to obtain input image x. Now, the aim is to find encoder-decoder pairs e_0, d_0 and e_n, d_n for all segmented tissues n such that the error E is minimised:

$$E = \epsilon(x, y_n, b, \bar{y_n}, \bar{b}), \text{ where }, \bar{b} = d_1(e_1(x)) \text{ and } \bar{y}_n = d_n(e_n(x)) \text{ for all } n \in N, \tag{7.1}$$

where N is the number of tissues that we would like to segment. Following the training of the neural network, the Dice score is used to compute the similarity between y_n and $\bar{y}_n$. The Dice score is a metric of the prediction value of the neural network. The images of the predicted segmentations and the ground truths have continuous values, while the Dice score computes the overlap of binary images. Therefore, for both the ground truth and the predicted segmentations a voxel $\mathbf{r}$ was determined to be part of respectively the ground truth or predicted binary segmentation of tissue n if the intensity of $y_n(\mathbf{r})$ or $\bar{y}_n(\mathbf{r})$ was over 0.5. The Dice scores are computed using these binary segmentations. The metric of the bias field is not computed, since predicting the bias field is just an aid to improve the segmentation.

7.1.2. Network Architecture

In this project an encoder-decoder structure, like U-Net is used to encode and decode 3D MRI scans. In contrast to a U-Net, the neural network in this project has multiple encoder-decoder pairs instead of one. The network architecture will be explained schematically for the neural network that segments CSF. For the segmentation of CSF, WM and GM two more encoder-decoder pairs are added to the network. A schematic overview of the neural network is shown in Figure 7.1. For simplification the 3D images are shown as a 2D slice. After the input layer the neural network splits into two encoder-decoder paths. The upper encoder-decoder structure extracts the bias field ($\bar{b}$) from the input image. The lower encoder-decoder structure predicts the CSF segmentation ($\bar{y}_1$), where CSF is denoted tissue 1. Then the loss of the neural network can be computed and this error is backpropagated through the network by performing gradient descent with every iteration.

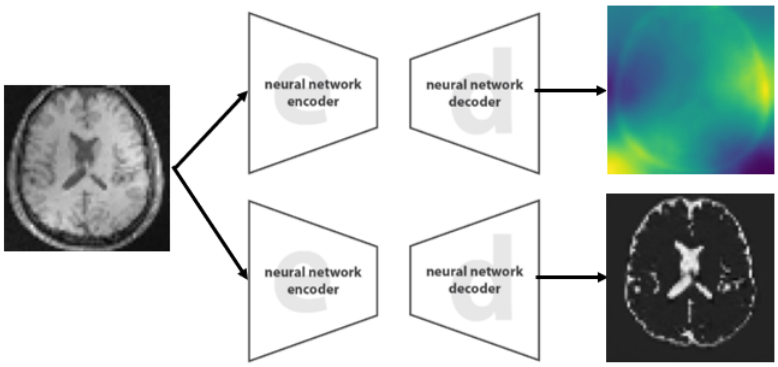


Figure 7.1: A schematic example of the neural network predicting the bias field and the segmentation.

The more detailed architecture of the neural network for CSF segmentation is depicted in Figure 7.2. The two encoder-decoder blocks of the network consist of identical structures. For the segmentation of more tissues, more identical encoder-decoder blocks are added. The neural network consists of convolutional layers, maxpooling layers, drop out layers and skip connections. The functions of these layers are explained in the previous chapter. The encoder-decoder structure used in this project is based on the implementation of a U-net made by Tomar [Tomar, 2021]. The encoder consists of multiple convolutional blocks, drop out and maxpooling layers. In the encoder structure the feature maps are downsampled. The architecture of the decoder follows the same pattern as the encoder, but upsamples the feature maps until an image with the same dimensions as the input image is obtained. The decoder consists of multiple transpose convolutional (deconvolutional) blocks and drop out layers. Skip connections are added between the convolutional blocks and the deconvolutional blocks.

7.1.3. Loss Function

The loss function of the neural network is composed of two parts: the reconstruction loss and the analytic segmentation loss. In Equation 7.2 the loss function is given, where the first two terms represent the reconstruction loss and the last terms represent the analytic segmentation loss.

$$E = \alpha_0 \text{MSE}(b, \bar{b}) + \sum_{n=1}^{N} \alpha_n \text{MSE}(y_n, \bar{y}_n) + \beta(\zeta \sum_{n=1}^{N+1} \int_\Omega |x(\mathbf{r}) - \bar{b}(\mathbf{r}) c_n|^2 \bar{y}_n^2(\mathbf{r}) d\mathbf{r} + \gamma \int_\Omega |\nabla \bar{b}(\mathbf{r})|^2 d\mathbf{r}), \tag{7.2}$$

where c_n are the tissue cluster values and α_0, α_n, β, ζ and γ are positive constants. As the name suggests the reconstruction term is the term that ensures that a good reconstruction is made. This is done by minimising the difference between the known ground truth bias field and segmentation and the predicted output. The analytic segmentation loss is based on the error minimized in Li's segmentation model (see Section 2.2.7 and Chapter 4) and a regularisation term that regularises the smoothness of the bias field. The user can choose the influence of the reconstruction terms and the analytic segmentation terms on the training loss by the constants α_0, α_n, β, ζ and γ in Equation 7.2. The influence of these parameters on the results are checked in a parameter study. If the ground truth segmentation(s) and/or bias field are unknown, the factors α_n and/or α_0 are zero.

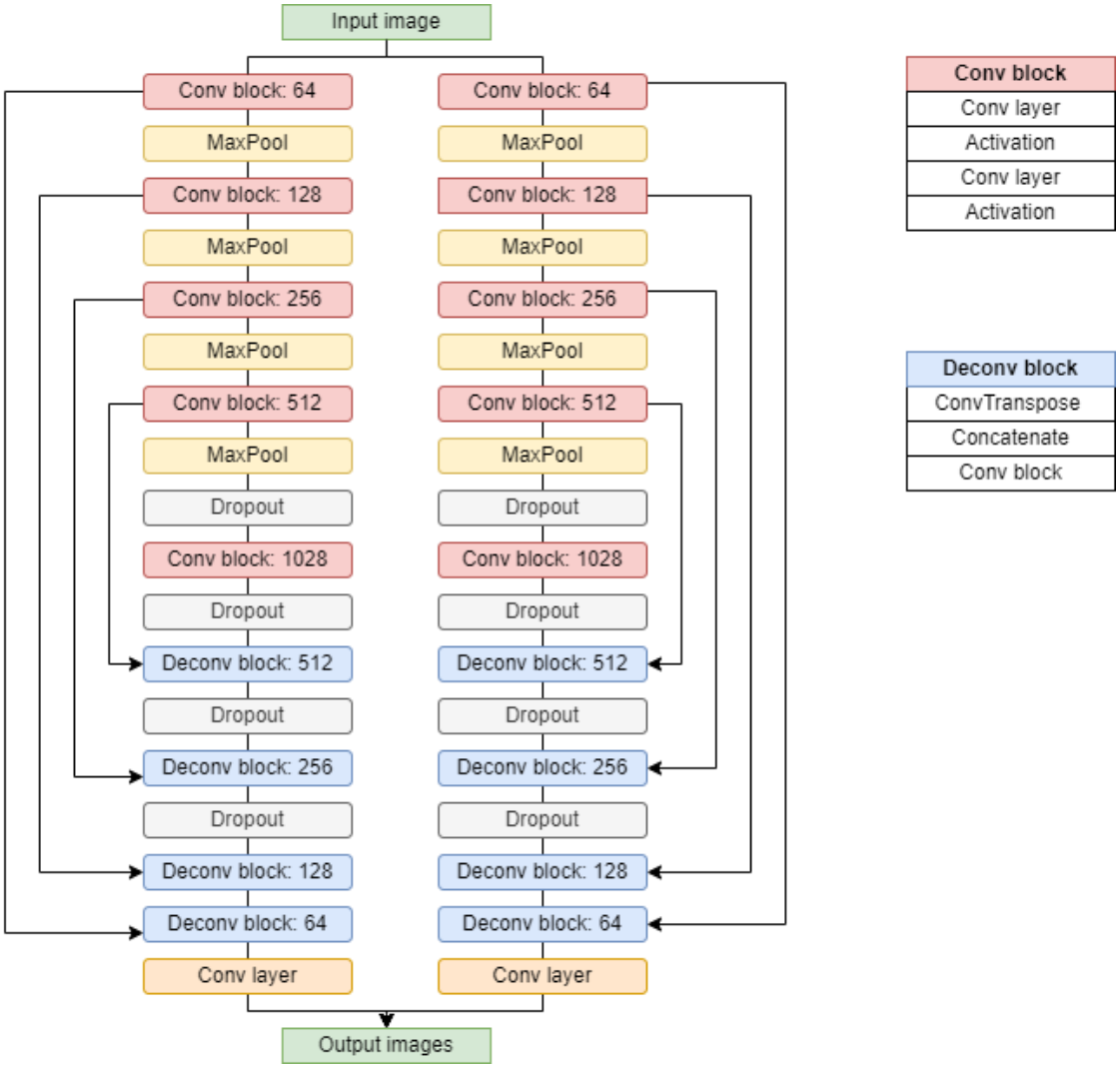


Figure 7.2: Neural network used for the segmentation of CSF and the bias field predicted from the BrainWeb images [Aubert-Broche et al., 2006] affected by a bias field. The skip connections are depicted with the black arrows between the conv and deconv blocks. The number in these blocks signify the number of filters used in the convolutional layers. The pool size of all maxpooling layers is 2 and these layers downsample the feature maps. The ConvTranspose layers upsample the feature maps. The kernel size used in all convolutional layers is 3 and the convolutional layers are activated with ReLU activation. The dropout in all dropout layers is 0.5. The optimizer used to train the model is ADAM and the batch size is 1.

The reconstruction loss is minimized by reducing the mean squared error (MSE) between b and $\bar{b}$ and the MSE between y_n and $\bar{y}_n$ for the tissues n that we would like to segment. The formula of the MSE loss is given in Equation 6.2.

The analytic segmentation loss consists of two terms inspired by Li's model (see Equation 4.5). The derivation from this equation to the analytic loss function of the neural network is given in Equation 7.3. The derivation steps are explained in the following paragraphs.

For simplicity the dirac delta function is chosen as the kernel function: $K(\mathbf{x}-\mathbf{y}) = \delta(\mathbf{x}-\mathbf{y})$. This makes implementing the term in the loss function of the neural network less complex. However, one of the conditions of Li's model is that the bias field is smooth and slowly varying, which is not complied with when the dirac delta function is taken as the kernel function. Therefore, a term is added that requires the bias field to be smooth when this term is minimized.

Then, instead of integrating over each Ω_n region separately, the integration can be performed over Ω by including the outcome of the predicted segmentation $\bar{y}_n$. In the level set implementation of Li's method this

is accomplished with the Heaviside functions of the level set(s). Unfortunately, the Heaviside function is discontinuous and thus the gradients determined from the loss function after implementation of the Heaviside function cannot be backpropagated into the neural network. A solution for the gradient problem could be a regularized function. For example $\bar{y}_n$ could go through a Sigmoid activation layer, which acts as a regularized Heaviside function and puts all output between 0 and 1, which is exactly as desired. Unfortunately, during experimentation this activation layer caused the output neuron to die, which stopped the learning of the neural network. Thus, to ensure that the predicted segmentation is between 0 and 1, a background cluster is introduced, which is defined by $\bar{y}_{N+1} = 1 - \sum_{n=1}^{N} \bar{y}_n$. By minimizing the energy of this cluster as well as the other tissue clusters, it is enforced that for every voxel $\mathbf{r}$ $\sum_{n=1}^{N+1} \bar{y}_n(\mathbf{r}) = 1$. However, a problem remains: negative results for output $\bar{y}_n(\mathbf{r})$ would be advantageous since this can add a negative term to the total loss function. Therefore, the choice was made to use $\bar{y}_n^2(\mathbf{r})$ in the loss function. This ensures that the clustering error is positive for every cluster and together with the definition of the background cluster, this leads to $\bar{y}_n$ being between 0 and 1. Consequently, the analytic segmentation loss becomes:

$$\begin{aligned} E_{AS} &= \int_\Omega \Big(\sum_{n=1}^{N+1} \int_{\Omega_n} K(\mathbf{r}-\mathbf{r}_0) |x(\mathbf{r}_0) - \bar{b}(\mathbf{r}) c_n|^2 d\mathbf{r}_0 \Big) d\mathbf{r} \\ \Longleftrightarrow \zeta &\int_\Omega \Big(\sum_{n=1}^{N+1} \int_{\Omega_n} \delta(\mathbf{r}-\mathbf{r}_0) |x(\mathbf{r}_0) - \bar{b}(\mathbf{r}) c_n|^2 d\mathbf{r}_0 \Big) d\mathbf{r} + \gamma \int_\Omega |\nabla \bar{b}(\mathbf{r})|^2 d\mathbf{r} \\ \Longleftrightarrow \zeta &\int_\Omega \Big(\sum_{n=1}^{N+1} \int_{\Omega} \delta(\mathbf{r}-\mathbf{r}_0) |x(\mathbf{r}_0) - \bar{b}(\mathbf{r}) c_n|^2 d\mathbf{r}_0 \Big) \bar{y}_n^2(\mathbf{r}) d\mathbf{r} + \gamma \int_\Omega |\nabla \bar{b}(\mathbf{r})|^2 d\mathbf{r} \\ \Longleftrightarrow \zeta &\sum_{n=1}^{N+1} \int_\Omega |x(\mathbf{r}) - \bar{b}(\mathbf{r}) c_n|^2 \bar{y}_n^2(\mathbf{r}) d\mathbf{r} + \gamma \int_\Omega |\nabla \bar{b}(\mathbf{r})|^2 d\mathbf{r} \end{aligned} \tag{7.3}$$

and the optimal values of the cluster means are (obtained using Euler-Lagrange):

$$c_n = \frac{\int_\Omega \bar{b}(\mathbf{r}) x(\mathbf{r}) \bar{y}_n^2(\mathbf{r}) d\mathbf{r}}{\int_\Omega \bar{b}(\mathbf{r})^2 \bar{y}_n^2(\mathbf{r}) d\mathbf{r}}. \tag{7.4}$$

7.1.4. Implementation

The framework of the neural network is implemented using Tensorflow [Abadi et al., 2016]. For training of the neural network the ADAM optimizer is used. A parameter study is performed to choose an adequate learning rate. Both the model for CSF segmentation and the CSF, WM and GM segmentation model are trained for 60 epochs with a batch size of 1. A higher batch size could not be chosen due to memory limits.

7.2. Data

The data that is used as input of the neural network are 3D BrainWeb images, of which there is a known CSF, GM and WM segmentation [Aubert-Broche et al., 2006]. The 3D data files were 181x217x183 in size and were scaled down to 64x64x64 such that the neural network is smaller and it takes less memory to train. All images are normalized. The dataset consists of 20 3D BrainWeb images together with their segmentation. 14 images are used as training data, 3 as validation data, and 3 as test data. Since the training dataset consist of only 14 samples, overfitting occurs when training the neural network. Since there is no access to more segmented BrainWeb data, data augmentation is applied to the available scans. Data warping is used as data augmentation technique. The types of data augmentation applied are noise, kernel filtering, rotation, shifts and zoom. This resulted in a training dataset of 300 images, a validation set of 80 images and a test set also of 80 images.

8

Results of Neural Network

In this chapter the results of the neural network used for the segmentation of MRI scans are discussed. First, an analysis into the different parameters of the neural network is included to study the effect of the parameters and especially of the analytic segmentation loss on the segmentation results. Then the output of the CSF and the CSF, WM and GM segmentation neural network are shown for the experiment with the best parameters. The development of the different elements of the loss function are also examined. Finally, the trained neural network is transferred to an infant dataset. The neural network is trained further on this dataset by only using the analytic segmentation loss, to evaluate the development of the segmentation results through training by only using the unsupervised loss.

8.1. Parameter Study

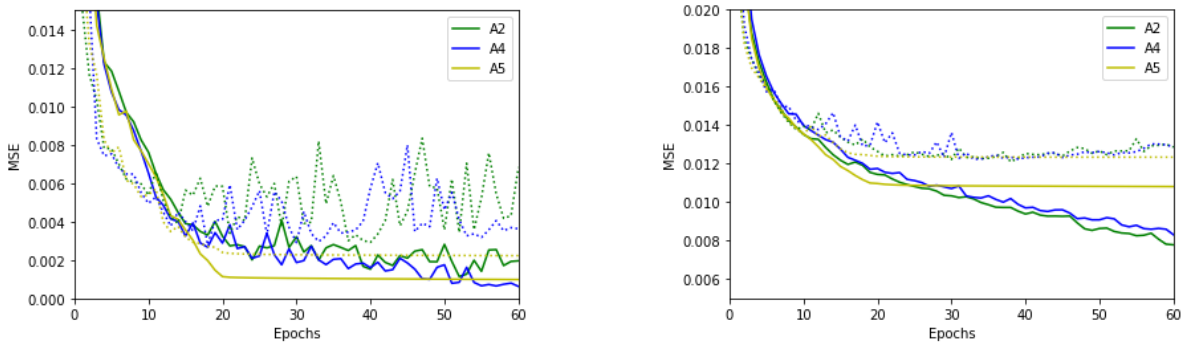
In this section the effect of the different parameters of the neural network and training strategy are studied. The parameters that will be investigated in this section are the learning rate and the constants of the loss function (see Equation 7.2). Multiple experiments are performed by changing one parameter and keeping the rest of the parameters fixed. The performance of the different experiments are checked by computing the Dice scores of the train and the test set between the segmentation ground truth and predicted segmentation.

8.1.1. Learning Rate

First, the learning rate is investigated. The learning rate determines the step size of the gradient descent method for propagating the error through the neural network. Table 8.1 shows the learning rate experiments. In the first three experiments, the learning rate is kept constant. From the Dice scores given in Table 8.5 it can be concluded that 10^{-4} is the best constant learning rate for this neural network. However, the learning rate can be adapted throughout training. Therefore, in Experiment A4 and A5 the learning rate was lowered from 10^{-4} over the epochs. This was implemented, because it is logical to first take big steps in the gradient descent method to get faster to the eventual solution. Then after some epochs the improvement of the loss gets smaller, because a large learning rate makes the model converge fast but causes it to possibly jump over minima. When the improvement of loss is small it is time to decrease the learning rate, which ensures that we do not jump over the minima [Buduma and Locascio, 2017]. The development of the MSE over the training steps (epochs) between the predicted bias field and the ground truth and the predicted CSF segmentation and the ground truth are given in Figure 8.1 for Experiment A2, A4 and A5. It can be concluded from these figures that for Experiment A2, and A4 the MSE of the training set decreases over the epochs, but the MSE of the validation set goes up. Thus, overfitting on the training set occurs. However, due to the data augmentation and drop out layers, overfitting is limited. In Experiment A5 the learning rate is decreased up to a level where the training first goes faster than Experiment A2 and A4 and then stops. This results in the best MSE values of the bias field and the segmentation for the validation set. From Table 8.5 it can be concluded that the learning rate from experiment A5 is the best, since the Dice score of the test set is the highest.

Exp	Learning rate
A1	10^{-3}
A2	10^{-4}
A3	10^{-5}
A4	10^{-4} for epoch ≤ 10 min(10^{-4} - 10^{-6} (epoch-10), 10^{-7}) for epoch>10
A5	10^{-4} for epoch ≤ 10 min(10^{-4} - 10^{-5} (epoch-10), 10^{-7}) for epoch>10

Table 8.1: Experiments with different learning rates. The rest of the parameters are $\alpha_0 = 1$, $\alpha_1 = 1$ and $\beta = 0$.



(a) Bias field MSE

(b) CSF segmentation MSE

Figure 8.1: On the left the MSE between the predicted bias field and the ground truth bias field over 60 epochs for the experiments A2, A4 and A5 of Table 8.1 is given. On the right the MSE between the predicted CSF segmentation and the ground truth CSF segmentation is shown. The filled line depicts the MSE of the train set, the striped line the MSE of the validation set.

8.1.2. Loss Function Parameters

In this section the parameters of the loss function are investigated. The loss function is specified in Equation 7.2. The loss function contains 5 parameters. α_0 and α_1 are the parameters in front of respectively the bias field and the CSF segmentation MSE, which are the supervised segmentation loss parts. The other parameters are part of the unsupervised analytic segmentation loss part. β determines if the analytic segmentation loss is taken into account and how much influence it has on the total loss. ζ and γ are respectively the parameters in front of the clustering loss and the bias smoothness loss. In a couple of experiments the influence on the segmentation results of these parameters and in general the unsupervised loss function are checked. The experiments are given in Tables 8.2, 8.3 and 8.4. The learning rate in the experiments is set to 10^{-4}.

The first experiments check the effect of α_0 and α_1. Since the bias field estimation and segmentation are predicted in separate encoder-decoder structures, the influence of α_0 and α_1 can only be checked by including a term in the loss that connects them. This is the clustering term and thus β and ζ should be set nonzero in these experiments. In the second round of experiments the influence of ζ and γ are checked. The Experiments D1-D5 are executed to check the effect of β. The respective Dice scores for the train and test set are given in Table 8.5. From Experiment B2-B5 it can be concluded that, when the parameters are tuned correctly, adding the analytic segmentation loss to the loss function improves the Dice score of the segmentation. Thus, the predictions of the neural network are better by using a combination of supervised and unsupervised learning. The best Dice scores of Experiment B1-B5 are for parameters $\alpha_0 = 1$ and $\alpha_1 = 100$. Therefore, in Experiments C1-C5 α_0 is set to 1 and α_1 is set to 100. In these experiments different values for ζ and γ are taken. From Experiments C1-C5 it can be concluded that the best result are obtained when both ζ and γ are set to 1. Experiments D1-D6 reveal that when parameter β is above 10^{-4} it gives worse results than while β is zero. Thus, in this case the segmentation loss is too overpowering and worsens the result, because it determines the wrong clusters. This is possibly due to the algorithm getting stuck in the wrong local minima. Another reason might be the fact that the images contain not only CSF and background but also other tissues that have similar intensities to, in this case, CSF. The other tissues are then grouped incorrectly into the CSF cluster instead of the background cluster. The analytic segmentation loss improves the segmentation result

Exp	α_0	α_1
B1	1	1
B2	10	1
B3	100	1
B4	1	10
B5	1	100

Table 8.2: Experiments with α_0 vs. α_1. The rest of the parameters are $\beta = 10^{-5}$, $\zeta = 1$ and $\gamma = 1$.

Exp	ζ	γ
C1	1	1
C2	5	1
C3	10	1
C4	1	5
C5	1	10

Table 8.3: Experiments with ζ vs. γ. The rest of the parameters are $\beta = 10^{-5}$, $\alpha_0 = 1$ and $\alpha_1 = 100$.

Exp	β
D1	10^{-2}
D2	10^{-3}
D3	10^{-4}
D4	10^{-5}
D5	10^{-6}
D6	10^{-7}

Table 8.4: Experiments with different values for β. The rest of the parameters are $\alpha_0 = 1$, $\alpha_1 = 100$, $\zeta = 1$ and $\gamma = 1$.

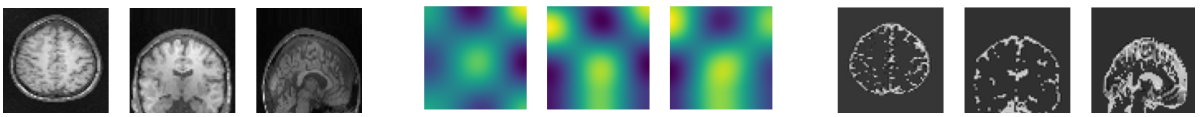
Exp	DS train	DS test	Exp	DS train	DS test	Exp	DS train	DS test	Exp	DS train	DS test
A1	0.7328	0.7272	B1	0.7634	0.7448	C1	0.7776	0.7547	D1	0.7448	0.7274
A2	0.7671	0.7465	B2	0.7688	0.7522	C2	0.7724	0.7490	D2	0.7630	0.7482
A3	0.7115	0.6933	B3	0.7701	0.7524	C3	0.7764	0.7538	D3	0.7644	0.7501
A4	0.7613	0.7430	B4	0.7667	0.7518	C4	0.7721	0.7524	D4	0.7776	0.7547
A5	0.7580	0.7487	B5	0.7776	0.7547	C5	0.7676	0.7498	D5	0.7761	0.7513
									D6	0.7712	0.7481

Table 8.5: Dice scores for train and test set for the experiments of Tables 8.1, 8.2, 8.3 and 8.4

when β is below 10^{-4}. The best Dice scores are for β is 10^{-5}. In conclusion, the segmentation results improve when the unsupervised loss is added to the supervised loss function of the neural network, if the parameters are tuned correctly. The optimal parameters, determined by this parameter study, for training this neural network are α_0=1, α_1=100, β= 10^{-5} , ζ=1, and γ=1.

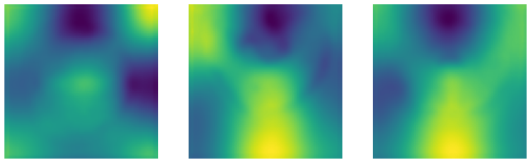
8.2. CSF Segmentation

In this section the results of the neural network that predicts the bias field and the CSF segmentation are shown. The parameters for this run are α_0=1, α_1=100, $\beta = 10^{-5}$, ζ=1, and γ=1. These parameters were determined by the parameter study. The learning rate was first chosen as in Experiment A5, but the Dice score for the test set was worse than the Dice score of Experiment D4, which was run with the same parameters but with a different learning rate. This is because, while Experiments A1-A5 showed that the learning rate of Experiment A5 was the best choice, the learning rates were investigated for the training without taking the analytic segmentation loss into account. On further investigation the following learning rate was chosen: 10^{-4} for the first 10 epochs, whereafter it decreased by 5 10^{-6} with every epoch. Thus, the training still improved when the learning rate was slowly lowered, but the factor in which it was lowered had to be adjusted. The Dice score for the train set is 0.7675 and the Dice score of the test is 0.7550. An example of an input image, that is part of the test set, with its ground truth bias field and segmentation is shown in Figure 8.2. The predictions of the neural network for this input image are shown in Figure 8.3. The course of the different parts of the loss function over the epochs for the train and validation data are shown in Figure 8.4. This shows that all different parts of the loss go down over the epochs.

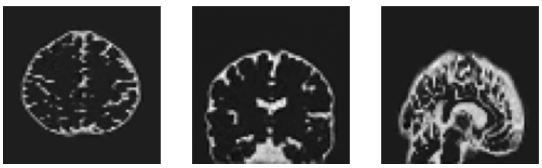


(a) Input image (b) Ground truth bias field (c) Ground truth CSF segmentation

Figure 8.2: In this figure an example of the input image and the corresponding ground truth segmentation and bias field are shown. The input and ground truths are 3D images. Here these 3D images are represented as three 2D slices through the middle of the image. This scan is part of the test set.

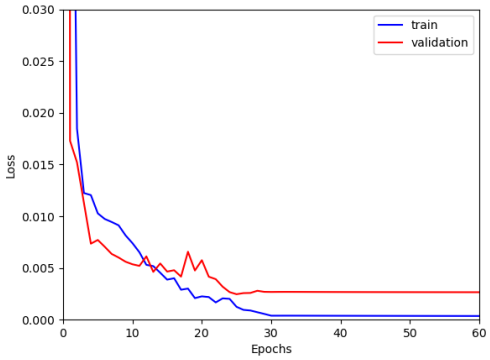


(a) Predicted bias field

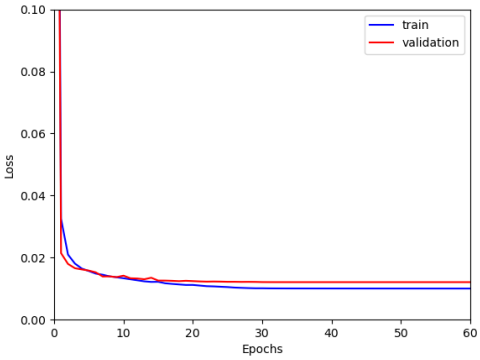


(b) Predicted CSF segmentation

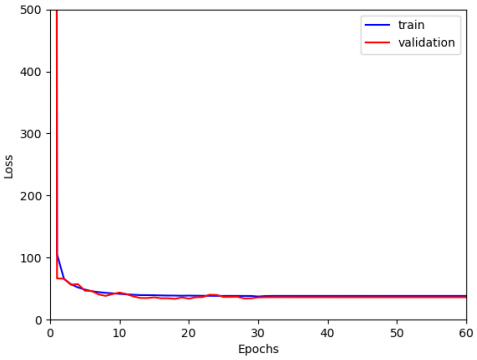
Figure 8.3: In this figure the output of the neural network is given based on the input image of Figure 8.2a. The neural network is trained for 60 epochs with the following parameters: α_0=1, α_1=100, $\beta = 10^{-5}$, ζ=1, and γ=1. The output are two 3D images. Here these 3D images are represented as three 2D slices through the middle of the image.



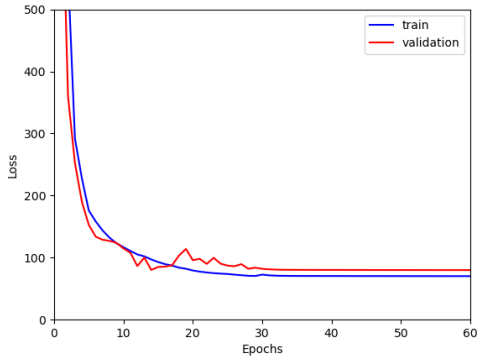
(a) Bias field MSE



(b) Segmentation MSE



(c) Clustering loss



(d) Bias smoothness loss

Figure 8.4: In these figures the course of the different parts of the loss function during the training stage are shown.

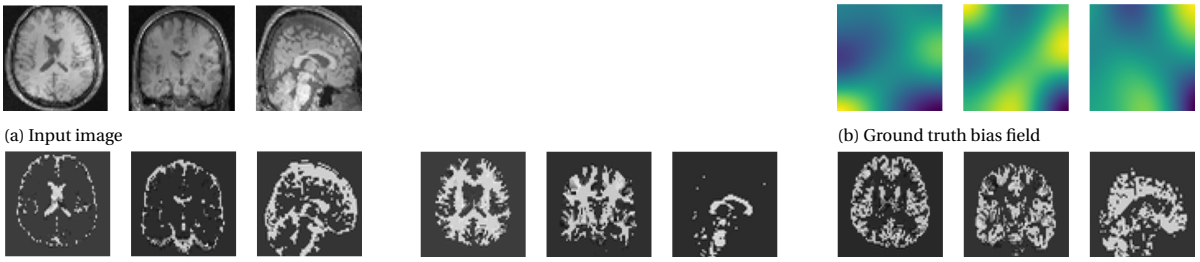
8.3. CSF, WM and GM Segmentation

The neural network for the segmentation of CSF was extended with two more encoder-decoder pairs that would extract the WM and GM regions from the input image. The N in the loss function, Equation 7.2, is thus 3. The indexes 1, 2, and 3 for tissues n are respectively CSF, WM and GM. Index 4 is the background, and is extracted from the neural network through taking $y_4 = 1 - y_1 - y_2 - y_3$. This neural network is trained with the same dataset as the CSF segmentation network. The parameters of the neural network are taken as follows: α_0=1, α_1=100, α_2=100, α_3=100, $\beta = 10^{-5}$, ζ=1, and γ=1. The learning rate is again 10^{-4} for the first 10 epochs, whereafter it decreased by 5 10^{-6} with every epoch. The Dice scores for the train and test set are given in Table 8.6. It is interesting to notice that the Dice score for the CSF segmentation has improved for both the train and the test set compared to the Dice scores of the CSF segmentation network. This is probably because the background cluster is now much smaller. It contains mostly actual background voxels and non-brain tissue, while in the CSF segmentation model WM and GM voxels were also part of the background cluster. If the background cluster contains voxels with many different intensities, the cluster value c_{N+1} is not a good representation, since the second condition of Li's algorithm is not satisfied. This condition stated that the

true image intensities are approximately constant within each class of tissue. Therefore, the energy of the background clustering term is large if this condition is not satisfied. Consequently, the energy term of the background clustering is smaller during the training of CSF, WM and GM network. Therefore, the clustering of the other tissues becomes better and thus the CSF segmentation is better estimated in this segmentation network, where three tissues are segmented instead of one. An example of an input image with its ground truth bias field and segmentations are shown in Figure 8.5. The predictions of the neural network for this input image are shown in Figure 8.6. The course of the different parts of the loss function over the epochs for the train and validation data are shown in Figure 8.7. As before, all different parts of the loss go down over the epochs.

Dice scores	CSF	WM	GM
Train set	0.7814	0.8812	0.8787
Test set	0.7631	0.8761	0.8733

Table 8.6: Dice scores of the train and test set for CSF, WM and GM segmentation.



(a) Input image

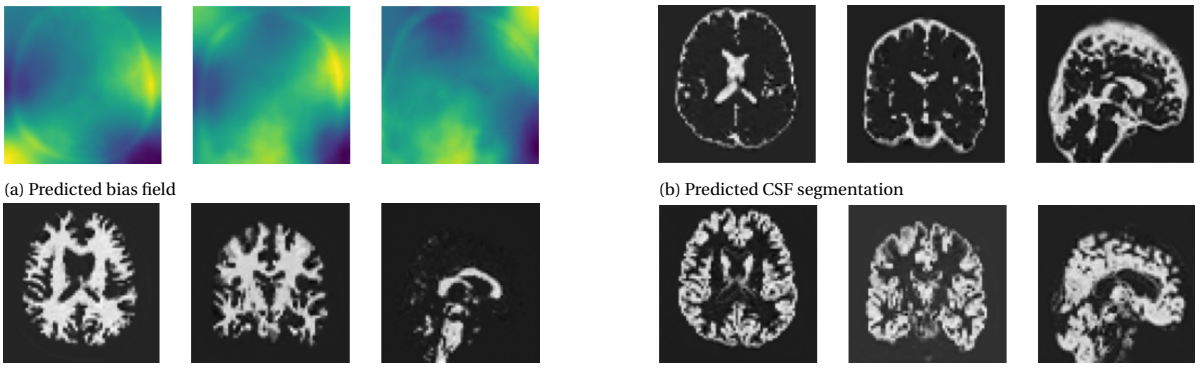
(b) Ground truth bias field

(c) Ground truth CSF segmentation

(d) Ground truth WM segmentation

(e) Ground truth GM segmentation

Figure 8.5: In this figure an example of the input image and the corresponding ground truth segmentations and bias field are shown. The input and ground truths are 3D images. Here these 3D images are represented as three 2D slices through the middle of the image. This scan is part of the test set.



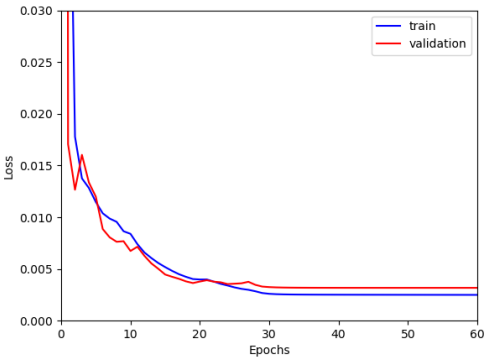
(a) Predicted bias field

(b) Predicted CSF segmentation

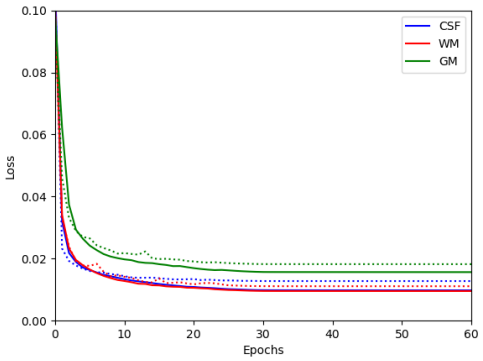
(c) Predicted WM segmentation

(d) Predicted GM segmentation

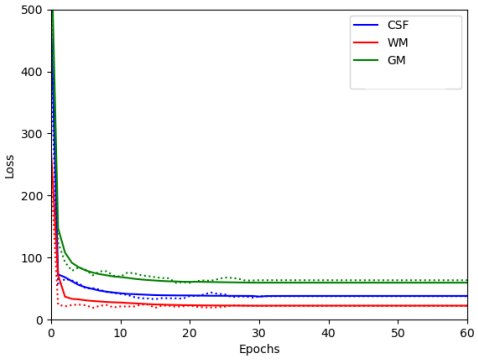
Figure 8.6: In this figure the output of the neural network is given. The neural network is trained for 60 epochs with the following parameters: α_0=1, α_1=100, α_2=100, α_3=100, $\beta = 10^{-5}$, ζ=1, and γ=1. The output are 3D images.



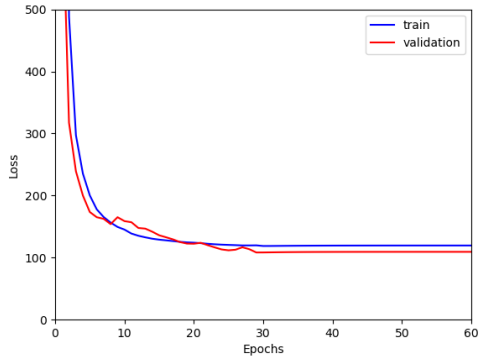
(a) Bias field MSE



(b) Segmentation MSE. Train set (filled), validation set (dotted).



(c) Clustering loss. Train set (filled), validation set (dotted).



(d) Bias smoothness loss

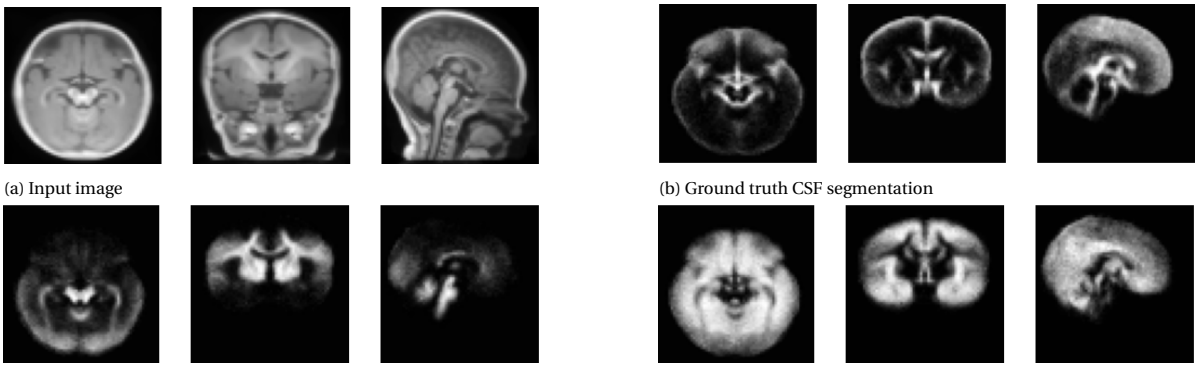
Figure 8.7: In these figures the course of the different parts of the loss function during the training stage are shown.

8.4. Training Without Ground Truth Segmentation

Now, a neural network for brain segmentation has been trained. However, the training was performed on artificial BrainWeb data that depicted T1-weighted images from a high-field scanner. The idea is to use this network on low-field scans. Therefore, the network needs to be trained on low-field scans such that it gives adequate output for these scans. This is unfortunately impossible, since only few scans are available from the low-field scanner and these scans are too diverse to use as a train set. Thus, to still show the advantage of the analytic segmentation in the loss function, for future application to low-field scans, the trained neural network was transferred to a dataset containing infant brain scans [Richards, 2019]. These scans are made with a high-field scanner, but are of children of age 0 to 5, which is the target group of the low-field scanner project. The scans were available with ground truth segmentations for CSF, WM and GM. The dataset contained 7 scans. As with the BrainWeb data all images were scaled from 0 to 1 and scaled down to the dimensions of 64x64x64. An example of a scan from the dataset with its ground truth segmentations is displayed in Figure 8.8. The ground truth segmentations of this dataset are like the BrainWeb segmentations continuous from 0 to 1. Therefore, the segmentations take the partial volume effect into account by setting values between 0 and 1. As can be seen from Figure 8.8 the level of fuzzyness (values between 0 and 1) of the ground truths of this dataset is much higher than in the BrainWeb dataset. There is no reason for this, other than that the reseachers determined the ground truths differently. The 7 scans were put into the trained neural network from Section 8.3. The trained neural network is thus transferred to the new data. The initial weights of the neural network for training on the infant dataset are equal to the final weights determined by training on the BrainWeb dataset. Then, using this initial guess, the network is trained on the infant dataset. All layers remained trainable for the new training.

Since using this infant dataset is the first step to using this network for segmentation of low-field scans, it is assumed that the ground truth segmentations are not available. The ground truth segmentations are ignored, because we want to show that segmentation results can be improved by only using the unsupervised

loss. We want to show this improvement, because the analytic segmentation results were satisfactory for the low-field scans and because there is not always a ground truth segmentation available, which is the case for the few available low-field scans and may be the case in the future, when more scans are available. This means that in the loss function α_i for $i = 0,..,3$ is zero (see Equation 7.2). Thus, the neural network is only trained with the analytic segmentation loss. The parameters for training are $\beta = 10^{-5}$, ζ=1, γ=1, the learning rate is 10^{-5} and the network is trained for 10 epochs. The predicted segmentations of the input scan, depicted in Figure 8.8a, before and after training can be found in respectively Figure 8.9 and 8.10. The Dice scores before and after training for the different regions are given in Table 8.7. Unfortunately, the Dice scores are much worse after training. An explanation for this effect is something that was discovered during research on analytic segmentation techniques. Voxels of tissues, that have signal intensities close to the target tissue, can be wrongfully segmented into the cluster of this target tissue. When the ground truth segmentations were taken into account, this was less of a problem, since the analytic segmentation loss was steered in the right direction through the MSE loss. But now, no ground truth segmentations are available and thus these other tissues, like skin, fat, eyes, etc., are segmented into the wrong regions. This is also clear from the predicted segmentations shown in Figure 8.10.



(a) Input image

(b) Ground truth CSF segmentation

(c) Ground truth WM segmentation

(d) Ground truth GM segmentation

Figure 8.8: In this figure an example of an input image and its corresponding ground truth segmentations are shown. The input and ground truths are 3D images. The 3D images are represented as three 2D slices through the middle of the image.

(a) Predicted CSF segmentation

(b) Predicted WM segmentation

(c) Predicted GM segmentation

Figure 8.9: Example of the predicted segmentations based on a scan from the infant dataset (Figure 8.8a) before training.

(a) Predicted CSF segmentation

(b) Predicted WM segmentation

(c) Predicted GM segmentation

Figure 8.10: Example of the predicted segmentations based on a scan from the infant dataset (Figure 8.8a) after training.

Dice scores	CSF	WM	GM
Before training	0.4488	0.6335	0.7483
After training	0.2558	0.2483	0.6347

Table 8.7: Dice scores before and after training of the neural network for the infant brain dataset [Richards, 2019].

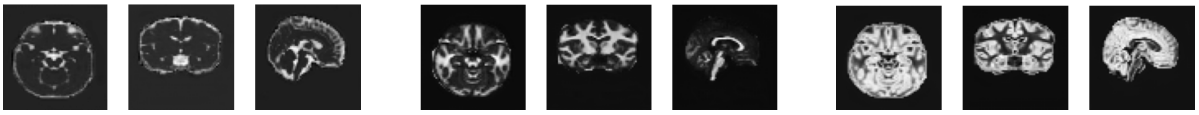
As before, this problem can be solved by brain masking. The masks of the brain scans are available from the dataset [Richards, 2019]. The masked brain of the scan shown in Figure 8.8a is depicted in Figure 8.11. The

masked brains were put into the neural network and trained with the same parameters as before. The predicted segmentations before and after training of the masked brain from Figure 8.11 are respectively shown in Figure 8.12 and 8.13. The results of the Dice scores are given in Table 8.8. After training the Dice scores for all tissue clusters have improved. Thus, there is an advantage in using the analytic segmentation method solely if no ground truth segmentations are available, but for a good segmentation the scans have to be brain-masked before putting them into the neural network.

Figure 8.11: Masked brain of the scan depicted in Figure 8.8a

(a) Predicted CSF segmentation (b) Predicted WM segmentation (c) Predicted GM segmentation

Figure 8.12: Example of the predicted segmentations based on a scan from the masked infant dataset (Figure 8.11) before training.



(a) Predicted CSF segmentation (b) Predicted WM segmentation (c) Predicted GM segmentation

Figure 8.13: Example of the predicted segmentations based on a scan from the masked infant dataset (Figure 8.11) after training.

Dice scores	CSF	WM	GM
Before training	0.4980	0.5988	0.6645
After training	0.5125	0.6050	0.7635

Table 8.8: Dice scores before and after training of the neural network for the masked infant brain dataset [Richards, 2019].

9

Discussion and Recommendations

9.1. Discussion of Results

In the first part of this thesis, analytic segmentation techniques were studied. The most promising methods from the literature search were applied to segment high- and low-field data. Via the Dice scores the results of the analytic segmentation methods were checked against the ground truth segmentation. Not for every image the ground truth segmentation was available, and in those cases, no Dice scores could be computed. For the high-field artificial image, computing the Dice score was based on registered segmentations available for the image. Futhermore, for the phantom scan, the model of the phantom was available. However, scanning of the phantom yielded images that were not one-to-one comparable to the phantom model. Therefore, rigid and non-rigid transformations were needed to register the phantom model to the scans. The model was registered to the bias-corrected image obtained by the three-phase model. This causes some uncertainties in the results. Firstly, using a registered model as a ground truth brings in uncertainties, since the registration is not perfect. Secondly, the model was registered to the bias-corrected image obtained from the three-phase algorithm. This can cause a certain result bias for this algorithm. Unfortunately, SimpleElastix [Klein* et al., 2010], [Shamonin et al., 2014] was not able to register the phantom model to the original image or the bias-corrected image obtained by using SimpleITK's bias field correction algorithm [Beare et al., 2018]. Therefore, the choice was made to use the bias-corrected image obtained from the three-phase algorithm for the registration of the ground truth model to the scans. Another method to evaluate the results of segmentation is to manually make a ground truth segmentation of the scans. Due to the low SNR and low resolution, manually creating the ground truth segmentation is quite complex. Additionally, it introduces a user bias in the results. Manually constructing ground truth segmentations is also very time-consuming. Therefore, the registered phantom model seemed the best method for evaluation of the analytic segmentation results.

The Dice scores showed that when the scan contained a distinct bias field, the simultaneous segmentation and bias field correction algorithm of Li's method gave distinctively the best results. When there was no bias corruption or only of minimal size, a simple clustering algorithm performed comparably. This was the case for both the artificial high-field images and the low-field phantom scans. An advantage of the k-means clustering method compared to Li's method was the clear time advantage. A 2D scan takes less than a second for the clustering algorithm, while it takes ~30 seconds for Li's algorithm to converge. For 3D segmentation, which is preferred, the long computation time can become an issue. However, computation time does not have to be a problem, since the numerical computations can be improved so that the speed of the algorithm increases. Another issue with both methods, but especially with Li's algorithm, is the number of parameters that had to be chosen. The right choice in parameters ensures convergence of the algorithm to the correct minimum. For another choice in parameters the segmentation may not converge, or can get stuck in a wrong local minimum. Therefore, these parameters are checked and possibly adapted for every new input image. The number of sections in which the image is segmented must also be selected a priori. These choices make it seem impossible to incorporate this algorithm into an automatic segmentation process integrated in the software of the low-field scanner. Therefore, this method should not stand on its own. However, it could be combined with another method that ensures the automation, but does not yet use the analytic information of the segmentation. This led to the idea that Li's method could be included in a neural network used for seg-

mentation. The parameters are only tweaked in the training stage, whereafter a fully trained neural network is obtained. This trained network then gives the results fast and automatically; no parameters are involved after training. Thus, the neural network ensures the automation, but Li's method ensures that the energy of the segmentation is minimized, which could improve the segmentation results. Therefore, in the second part of this project a neural network integrated with Li's method was built.

The neural network was trained with artificial high-field scans and not with low-field scans. This was because there were only few of such scans available. These brain scans were also made with very different set-ups. Thus, the low-field dataset was too small and too diverse to be useful for training. Another reason why the artificial high-field dataset was chosen, was because ground truth segmentations were available.

The results showed that adding the analytic segmentation loss to the total loss function improved the segmentation result, when the parameters were tuned correctly. Conclusions on the parameters were based on the Dice scores computed for the training and test set. This has some drawbacks. Firstly, the Dice scores are based on one run. Running the neural network with the same parameters will give similar but slightly different results. This is because training happens non-deterministically. Another reason why results differ slightly between runs is the small batch size. Since the batch size is 1, the training loss does not go down smoothly but varies over the epochs. Increasing the batch size would solve this problem, since this increases the stability of the training process. Unfortunately, due to memory limits increasing the batch size was not possible. Another issue with the parameter study is that when investigating one parameter the others need to be fixed. In which position they are fixed has influence on the end results. For example, a new analysis was required to find a good learning rate for the optimal parameters, since the combination of the optimal learning rate and the optimal parameters did not achieve the best segmentation results. Therefore, one can imagine that better parameter and learning rate combinations can be found for this neural network. Especially, since discrete steps were taken for every parameter, there are much more combinations available.

It was shown that all separate terms in the loss function went down during training of the neural network. This matches with the idea that over the epochs the segmentation becomes better, since the MSE decreases and the analytic segmentation loss goes down. The neural network for CSF segmentation was expanded to also segment WM and GM. This network was transferred to the infant dataset. It was shown that the analytic segmentation loss only improved the segmentation, when it was applied to brain scans that were masked. This was because the analytic segmentation does not work well when the brain scan contains tissues with similar intensities as the tissues that the method targets to segment. Unfortunately, masked brains were not part of the original dataset, leading to slightly worse results when applying the neural network on the masked brain infant images instead of the original infant scans. The results for the masked brain scans could be improved by including masked brains in the original dataset.

9.2. Recommendations

For future research it is recommended to look further into using a neural network combined with analytic segmentation techniques for the segmentation of MR brain scans. The analysis of multiple analytic segmentation techniques showed problems with automation due to the influence of parameters integrated in the algorithm. Neural networks solve these problems. In this project it was shown that the created neural network can be transferred to another dataset. A vast dataset should be created of scans from the low-field scanner. Then, the neural network should be transferred to this low-field dataset by using the analytic segmentation loss. After training, this network should be integrated in the software for the low-field scanner such that segmentation is almost instantly and automatically available. Since the eventual goal is to use this technique on hydrocephalic patients, the dataset should eventually contain brains of hydrocephalic infants.

As discussed, for the analytic segmentation to work, a brain mask of the brain scans has to be made. For the high-field data used in this project the masks of the brains were available, but this will not be the case when a low-field image is made. Therefore, it is recommended to construct an algorithm that finds the brain masks of the low-field scans. It should be investigated what the ideal brain masking method is for the low-field scans. This could be done by applying a neural network. Subsequently, the output of this brain masking network would be the input of the network built in this project. Another idea is using a combination of image processing tools, for example a region growing or clustering method with morphological operations.

The neural network should also be trained on MR images made with different sequences. Now, only T1-weighted images were used, but eventually the neural network should be able to segment T2- and PD-weighted images as well. Furthermore, if different types of images are available of the same subject, the segmentation could be improved by using information of multiple scans.

10 Conclusion

In the first part of this project we set out to find propitious segmentation techniques for low-field scans. Li's method showed the most promising results on scans that were affected by a bias field, due to the simultaneous bias field prediction and segmentation. When no or a minor bias field was present the results were comparable to k-means clustering. The major disadvantage of these methods and especially Li's method were the initialization values and the number of parameters that had to be chosen. Using these analytic methods for automatic segmentation therefore seemed quite complex. However, the promising results of Li's method led to the idea to incorporate this method into a neural network, since it would solve the issue with automation, but analytic information about the segmentation would still be used. Consequently, a neural network was built for segmentation. The neural network extracted the bias field and predicted the segmentation. Then, in the loss function of this neural network Li's method was introduced. The introduction of this method improved the segmentation results of the neural network for the CSF segmentation and the segmentation of CSF, WM and GM. After training on the original dataset, the neural network that segments CSF, WM and GM was applied on a new dataset containing scans of infant brains. When retraining the network to this new dataset, using only the analytic segmentation loss, the results of the segmentation became worse. This was because the segmentation algorithm clustered non-brain tissues, which voxels had similar signal intensities as the voxels of the target clusters, into the target regions. This problem was solved by using brain-masked images as input, since this removed the non-brain tissues. For these masked brain scans, it was shown that segmentation results improved by applying the analytic segmentation loss. It is therefore recommended, when more low-field brain data is available, to transfer the neural network to these new scans. However, for segmentation to work, the low-field data should first be brain-masked. It is recommended to investigate brain masking techniques for implementation in the complete segmentation algorithm.

In this thesis project the aim was to realize the foundations for a fast, automatic and practical 3D segmentation method for low-field MRI brain scans. By building a neural network for segmentation a fast, automatic and practical 3D segmentation method was created. The promising results of Li's method applied to the low-field scans and of transferring the neural network to the infant dataset show excellent future perspective for the segmentation of low-field scans.

11

Acknowledgements

I would like to express my sincerest appreciation to all those who gave me support to complete my master thesis project. I would first like to thank my supervisors, Frans Vos and Martin van Gijzen. Feedback given in our meetings helped me when I was stuck and brought my work to a higher level. I appreciate that every week you made time for me, to not only check in with my work but also on how I was doing personally, which was really appreciated in these COVID times. Additionally, the humour that you both brought to the meetings, made for a really nice break in the week. Thanks to Merel for your meetings, in which you not only helped me with problems in the project but you also let me vent about thesis struggles.

Tom, thank you so much for the data of the low-field scanner and especially with helping me do the experiments on the scanner. I know that you were extremely busy, so thank you for still making the time. Thanks to Andrew for setting up this project with Martin and having some nice discussions about the research questions.

Major thanks to my boyfriend Niek for proofreading part of this thesis, making studying much more fun with our study sessions, and for all the support this year. Last but not least, I would like to additionally thank my family, roommates, teammates and other friends for their consistent support during the graduation period.

Bibliography

M. Abadi, P. Barham A. Agarwal, E. Brevdo, C. Citro Z. Chen, G.S. Corrado, J. Dean A. Davis, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, and Y. Jia. TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems, 2016.

J. Ashburner and K.J. Friston. Unified Segmentation. *NeuroImage*, 26(3):839 – 851, 2005. ISSN 1053-8119. doi: https://doi.org/10.1016/j.neuroimage.2005.02.018. URL `http://www.sciencedirect.com/science/article/pii/S1053811905001102`.

B. Aubert-Broche, M. Griffin, B. Pike, A. Evans, and L. Collins. Twenty New Digital Brain Phantoms for Creation of Validation Image Data Bases. *IEEE transactions on medical imaging*, 25:1410–6, 12 2006. doi: 10.1109/TMI.2006.883453.

N. Badshah and K. Chen. Multigrid Method for the Chan-Vese Model in Variational Segmentation. *Commun. Comput. Phys.*, 4, 08 2008.

E. Bae and X. Tai. Efficient Global Minimization for the Multiphase Chan-Vese Model of Image Segmentation. In *Energy Minimization Methods in Computer Vision and Pattern Recognition*, pages 28–41, 08 2009. ISBN 978-3-642-03640-8. doi: 10.1007/978-3-642-03641-5_3.

R. Bartzatt. Determination of Dermal Permeability Coefficient (Kp) by Utilizing Multiple Descriptors in Artificial Neural Network Analysis and Multiple Regression Analysis. *Journal of Scientific Research and Reports*, 3:2884–2899, 01 2014. doi: 10.9734/JSRR/2014/13125.

R. Beare, B. Lowekamp, and Z. Yaniv. Image Segmentation, Registration and Characterization in R with SimpleITK. *Journal of Statistical Software*, 86(8):1–35, 2018. doi: 10.18637/jss.v086.i08. URL `https://www.jstatsoft.org/index.php/jss/article/view/v086i08`.

M.A. Bernstein, S.B. Fain, and S.J. Riederer. Effect of Windowing and Zero-Filled Reconstruction of MRI Data on Spatial Resolution and Acquisition Strategy. *Journal of Magnetic Resonance Imaging*, 14(3):270–280, 2001. doi: https://doi.org/10.1002/jmri.1183. URL `https://onlinelibrary.wiley.com/doi/abs/10.1002/jmri.1183`.

M.E. Brandt, T.P. Bohant, L.A. Kramer, and J.M. Fletcher. Estimation of CSF, White and Gray matter volumes in Hydrocephalic Children using Fuzzy Clustering of MR Images. *Computerized Medical Imaging and Graphics*, 18(1):25 – 34, 1994. ISSN 0895-6111. doi: https://doi.org/10.1016/0895-6111(94)90058-2. URL `http://www.sciencedirect.com/science/article/pii/0895611194900582`.

Nikhil Buduma and Nicholas Locascio. *Fundamentals of Deep Learning Designing Next-Generation Machine Intelligence Algorithms*. O'Reilly, 2017.

Jerrold T. Bushberg. The Essential Physics of Medical Imaging, Dec 2011.

J. Butman and M.G. Linguraru. Assessment of Ventricle Volume from Serial MRI scans in Communicating Hydrocephalus. pages 49 – 52, 06 2008. ISBN 978-1-4244-2002-5. doi: 10.1109/ISBI.2008.4540929.

V. Caselles, R. Kimmel, and G. Sapiro. Geodesic Active Contours. *International Journal of Computer Vision*, 22(1):61–79, 1997. ISSN 1573-1405. doi: 10.1023/A:1007979827043. URL `https://doi.org/10.1023/A:1007979827043`.

T.F. Chan and L.A. Vese. Active Contours Without Edges. *IEEE Transactions on Image Processing*, 10(2):266–277, 2001. doi: 10.1109/83.902291.

X. Chen, B. Williams, S. Vallabhaneni, G. Czanner, R. Williams, and Y. Zheng. Learning Active Contour Models for Medical Image Segmentation. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11624–11632, 06 2019. doi: 10.1109/CVPR.2019.01190.

A.M. Dale, B. Fischl, and M.I. Sereno. Cortical Surface-Based Analysis: I. Segmentation and Surface Reconstruction. *NeuroImage*, 9(2):179 – 194, 1999. ISSN 1053-8119. doi: https://doi.org/10.1006/nimg.1998.0395.

H.E. Davies, C.G. Wathen, and F.V. Gleeson. The Risks of Radiation Exposure Related to Diagnostic Imaging and How to Minimise Them. *Bmj*, 342:d947, 2011. ISSN 0959-8138. doi: 10.1136/bmj.d947.

L. Deng and D. Yu. Deep Learning: Methods and Applications. Technical report, Microsoft, May 2014.

M. Drozdzal, E. Vorontsov, G. Chartrand, S. Kadoury, and C. Pal. The Importance of Skip Connections in Biomedical Image Segmentation. In *Conference: International Workshop on Large-Scale Annotation of Biomedical Data and Expert Label Synthesis International Workshop on Deep Learning in Medical Image Analysis*, 08 2016. ISBN 978-3-319-46975-1. doi: 10.1007/978-3-319-46976-8_19.

A.K. Dubey and V. Jain. Comparative Study of Convolution Neural Network's Relu and Leaky-Relu Activation Functions. In Sukumar Mishra, Yog Raj Sood, and Anuradha Tomar, editors, *Applications of Computing, Automation and Wireless Systems in Electrical Engineering*, pages 873–880, Singapore, 2019. Springer Singapore. ISBN 978-981-13-6772-4.

J. Eber and C. Villaseñor. Ultrasound: Advantages, Disadvantages, and Controversies. *Nurse Pract Forum*, 2 (4):239–42, 1991. ISSN 1045-5485 (Print) 1045-5485.

P.K. Eide, L.M. Valnes, A.H. Pripp, K.A. Mardal, and G. Ringstad. Delayed Clearance of Cerebrospinal Fluid Tracer from Choroid Plexus in Idiopathic Normal Pressure Hydrocephalus. *J Cereb Blood Flow Metab*, 40 (9):1849–1858, 2020. ISSN 0271-678X (Print) 0271-678x. doi: 10.1177/0271678x19874790.

N. El-Zehiry, S. Xu, P. Sahoo, and A. Elmaghraby. Graph Cut Optimization for the Mumford-Shah Model. pages 182–187, 01 2007.

L. Ellingsen, S. Roy, A. Carass, A. Blitz, D. Pham, and J. Prince. Segmentation and Labeling of the Ventricular System in Normal Pressure Hydrocephalus using Patch-Based Tissue Classification and Multi-Atlas Labeling. In *Proceedings of SPIE–the International Society for Optical Engineering*, volume 9784, 03 2016. doi: 10.1117/12.2216511.

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, and A. M. Dale. Whole Brain Segmentation: Automated Labeling of Neuroanatomical Structures in the Human Brain. *Neuron*, 33(3):341–55, 2002. ISSN 0896-6273 (Print) 0896-6273. doi: 10.1016/s0896-6273(02)00569-x.

K.S. Fu and J.K. Mui. A Survey on Image Segmentation. *Pattern Recognition*, 13(1):3 – 16, 1981. ISSN 0031-3203. doi: https://doi.org/10.1016/0031-3203(81)90028-5. URL `http://www.sciencedirect.com/science/article/pii/0031320381900285`.

P. Getreuer. Chan-Vese Segmentation. *Image Processing On Line*, 2:214–224, 2012. `https://doi.org/10.5201/ipol.2012.g-cv`.

GHO. GHO - By Category - Medical Equipment - Data by Country, 2016. URL `https://apps.who.int/gho/data/node.main.510`.

R.C. Gonzalez and R.E. Woods. *Digital Image Processing*. Parson, 2nd edition, 2008.

I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

F. Grimm, F. Edl, I. Gugel, S.R. Kerscher, B. Bender, and M.U. Schuhmann. Automatic Volumetry of Cerebrospinal Fluid and Brain Volume in Severe Paediatric Hydrocephalus, Implementation and Clinical Course after Intervention. *Acta Neurochir (Wien)*, 162(1):23–30, 2020a. ISSN 0001-6268. doi: 10.1007/s00701-019-04143-5.

F. Grimm, F. Edl, S. Kerscher, K. Nieselt, I. Gugel, and M. Schuhmann. Semantic Segmentation of Cerebrospinal Fluid and Brain Volume with a Convolutional Neural Network in Pediatric Hydrocephalus—Transfer Learning from Existing Algorithms. *Acta Neurochirurgica*, 162:1–12, 10 2020b. doi: 10.1007/s00701-020-04447-x.

N.B. Gunter, C.G. Schwarz, J. Graff-Radford, J.L. Gunter, D.T. Jones, N.R. Graff-Radford, R.C. Petersen, D.S. Knopman, and C.R. Jack. Automated Detection of Imaging Features of Disproportionately Enlarged Subarachnoid Space Hydrocephalus using Machine Learning Methods. *NeuroImage: Clinical*, 21, 2019. ISSN 2213-1582. doi: 10.1016/j.nicl.2018.11.015.

A. Hatamizadeh, D. Sengupta, and D. Terzopoulos. End-to-End Deep Convolutional Active Contours for Image Segmentation. *CoRR*, abs/1909.13359, 2019. URL `http://arxiv.org/abs/1909.13359`.

K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition, 2016.

L. He and S. Osher. Solving the Chan-Vese Model by a Multiphase Level Set Algorithm Based on the Topological Derivative. volume 4485, pages 777–788, 05 2007. doi: 10.1007/978-3-540-72823-8_67.

R.J. Hemalatha, T.R. Thamizhvani, A. Dhivya, J. Joseph, B. Babu, and R. Chandrasekaran. *Active Contour Based Segmentation Techniques for Medical Image Analysis*. 07 2018. ISBN 978-1-78923-330-8. doi: 10.5772/intechopen.74576.

S. Hussain, X. Xi, I. Ullah, Y. Wu, C. Ren, Z. Lianzheng, C. Tian, and Y. Yin. Contextual Level-Set Method for Breast Tumor Segmentation. *IEEE Access*, 8:189343–189353, 2020. doi: 10.1109/ACCESS.2020.3029684.

K. Ishii, T. Kawaguchi, K. Shimada, S. Ohkawa, N. Miyamoto, T. Kanda, T. Uemura, T. Yoshikawa, and E. Mori. Voxel-Based Analysis of Gray Matter and CSF Space in Idiopathic Normal Pressure Hydrocephalus. *Dement Geriatr Cogn Disord*, 25(4):329–35, 2008. ISSN 1420-8008. doi: 10.1159/000119521.

K. Ishii, T. Soma, K. Shimada, H. Oda, A. Terashima, and R. Kawasaki. Automatic Volumetry of the Cerebrospinal Fluid Space in Idiopathic Normal Pressure Hydrocephalus. *Dementia and geriatric cognitive disorders extra*, 3:489–96, 12 2013. doi: 10.1159/000357329.

R.E. Jurdia, C. Petitjean, P. Honeine, V. Cheplygina, and F. Abdallah. High-level Prior-Based Loss Functions for Medical Image Segmentation: A Survey. *ArXiv*, abs/2011.08018, 2020.

K. Kazemi and N. Noorizadeh. Quantitative Comparison of SPM, FSL, and Brainsuite for Brain MR Image Segmentation. *J Biomed Phys Eng*, 4(1):13–26, 2014. ISSN 2251-7200 (Print) 2251-7200.

B. Kim and J.C. Ye. Mumford–Shah Loss Functional for Image Segmentation With Deep Learning. *IEEE Transactions on Image Processing*, 29:1856–1866, 2020. ISSN 1941-0042. doi: 10.1109/tip.2019.2941265. URL `http://dx.doi.org/10.1109/TIP.2019.2941265`.

S. Klein*, M. Staring*, K. Murphy, M. A. Viergever, and J. P.W. Pluim. Elastix: a Toolbox for Intensity-Based Medical Image Registration. *IEEE Transactions on Medical Imaging*, 29(1):196 – 205, January 2010.

M. Klimont, M. Flieger, J. Rzeszutek, J. Stachera, A. Zakrzewska, and K. Jończyk-Potoczna. Automated Ventricular System Segmentation in Paediatric Patients Treated for Hydrocephalus Using Deep Learning Methods. *BioMed Research International*, Volume 2019:9, 07 2019. doi: 10.1155/2019/3059170.

A. Krizhevsky, I. Sutskever, and G.E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012. URL `https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf`.

R.K. Kwan, A.C. Evans, and G.B. Pike. MRI Simulation-Based Evaluation of Image-Processing and Classification Methods. *IEEE Transactions on Medical Imaging*, 18(11):1085–1097, 1999. doi: 10.1109/42.816072.

Lecturio. Hydrocephalus (Water on the Brain) - Symptoms and Surgery (Shunt), Jul 2020. URL `https://www.lecturio.com/magazine/hydrocephalus/`.

C. Li, R. Huang, Z. Ding, C. Gatenby, D. Metaxas, and J. Gore. A Level Set Method for Image Segmentation in the Presence of Intensity Inhomogeneities With Application to MRI. *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society*, 20:2007–16, 07 2011. doi: 10.1109/TIP.2011.2146190.

A.W. Liew and H. Yan. Current Methods in the Automatic Tissue Segmentation of 3D Magnetic Resonance Brain Images. *Current Medical Imaging Reviews*, 2, 02 2006. doi: 10.2174/157340506775541604.

M.G. Linguraru and J. Butman. Segmentation Propagation for the Automated Quantification of Ventricle Volume from Serial MRI. *Progress in Biomedical Optics and Imaging - Proceedings of SPIE*, 7260, 02 2009. doi: 10.1117/12.810940.

D. Ma, V. Gulani, N. Seiberlich, K. Liu, J. Sunshine, J. Duerk, and M. Griswold. Magnetic Resonance Fingerprinting. *Nature*, 495:187–92, 03 2013. doi: 10.1038/nature11971.

E.N. Marieb and K. Hoehn. *Human Anatomy & Physiology*, page 483–483. Pearson, 10 edition, 2010.

J.P. Marques, F.F.J. Simonis, and A.G. Webb. Low-field MRI: An MR Physics Perspective. *Journal of Magnetic Resonance Imaging*, 49(6):1528–1542, 2019. doi: https://doi.org/10.1002/jmri.26637. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/jmri.26637.

T.M. Mitchell. *Machine Learning*. McGraw-Hill, 1997.

W. Mohd, A. Beg, T. Herawan, and K. Rabbi. MaxD K-Means: A Clustering Algorithm for Auto-generation of Centroids and Distance of Data Points in Clusters. *Communications in Computer and Information Science*, 316:192–199, 01 2012. doi: 10.1007/978-3-642-34289-9_22.

C. Muller, F. Peyrin, Y. Carillon, and C. Odet. Automated 3D Region Growing Algorithm Based on an Assessment Function. *Pattern Recognition Letters*, 23:137–150, 01 2002. doi: 10.1016/S0167-8655(01)00116-7.

M.A. Nielsen. *Neural Networks and Deep Learning*. Determination Press, 2015.

O. Oktay, E. Ferrante, K. Kamnitsas, M.P. Heinrich, W. Bai, J. Caballero, R. Guerrero, S.A. Cook, A. de Marvao, T. Dawes, D.P. O'Regan, B. Kainz, B. Glocker, and D. Rueckert. Anatomically Constrained Neural Networks (ACNN): Application to Cardiac Image Enhancement and Segmentation. *CoRR*, abs/1705.08302, 2017. URL http://arxiv.org/abs/1705.08302.

N. Otsu. A Threshold Selection Method from Gray-Level Histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66, 1979. doi: 10.1109/TSMC.1979.4310076.

N.R. Pal and S.K. Pal. A Review on Image Segmentation Techniques. *Pattern Recognition*, 26(9):1277 – 1294, 1993. ISSN 0031-3203. doi: https://doi.org/10.1016/0031-3203(93)90135-J. URL http://www.sciencedirect.com/science/article/pii/003132039390135J.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

J.L. Quon, M. Han, L.H. Kim, M.E. Koran, L.C. Chen, E.H. Lee, J.W., V. Ramaswamy, R.M. Lober, M.D. Taylor, G.A. Grant, S.H. Cheshier, J.R.W. Kestle, M.S.B. Edwards, and K.W. Yeom. Artificial Intelligence for Automatic Cerebral Ventricle Segmentation and Volume Calculation: a Clinical Tool for the Evaluation of Pediatric Hydrocephalus. *Journal of Neurosurgery: Pediatrics PED*, pages 1 – 8, 2020. doi: 10.3171/2020.6.PEDS20251. URL https://thejns.org/pediatrics/view/journals/j-neurosurg-pediatr/aop/article-10.3171-2020.6.PEDS20251/article-10.3171-2020.6.PEDS20251.xml.

J. Richards. Neurodevelopmental MRI Database: Infants, Jun 2019. URL https://www.nitrc.org/doi/landing_page.php?table=frs_file&id=11439&doi=10.25790/bml0cm.51.

P. Santago and H.D. Gage. Quantification of MR Brain Images by Mixture Density and Partial Volume Modeling. *IEEE Trans Med Imaging*, 12(3):566–74, 1993. ISSN 0278-0062 (Print) 0278-0062. doi: 10.1109/42.241885.

Y. Serulle, H. Rusinek, I. Kirov, H. Milch, E. Fieremans, A. Baxter, J. Mcmenamy, R. Jain, J. Wisoff, J. Golomb, O. Gonen, and A. George. Differentiating Shunt-Responsive Normal Pressure Hydrocephalus from Alzheimer Disease and Normal Aging: Pilot Study using Automated MRI Brain Tissue Segmentation. *Journal of neurology*, 261, 08 2014. doi: 10.1007/s00415-014-7454-0.

D.P. Shamonin, E.E. Bron, B.P.F. Lelieveldt, M. Smits, S. Klein, and M. Staring. Fast Parallel Image Registration on CPU and GPU for Diagnostic Classification of Alzheimer's Disease. *Frontiers in Neuroinformatics*, 7(50):1–15, January 2014.

M. Shao, X. Li, B. Dewey, A. Blitz, J. Prince, A. Carass, and L. Ellingsen. Multi-Atlas Segmentation of the Hydrocephalus Brain using an Adaptive Ventricle Atlas. In *Medical Imaging 2018: Biomedical Applications in Molecular, Structural, and Functional Imaging*, page 14, 03 2018. doi: 10.1117/12.2295613.

D.W. Shattuck and R.M. Leahy. BrainSuite: An Automated Cortical Surface Identification Tool. *Medical Image Analysis*, 6(2):129 – 142, 2002. ISSN 1361-8415. doi: https://doi.org/10.1016/S1361-8415(02)00054-3. URL http://www.sciencedirect.com/science/article/pii/S1361841502000543.

E. Shelhamer, J. Long, and T. Darrell. Fully Convolutional Networks for Semantic Segmentation. *IEEE Trans Pattern Anal Mach Intell*, 39(4):640–651, 2017. ISSN 0098-5589. doi: 10.1109/tpami.2016.2572683.

C. Shorten and T.M. Khoshgoftaar. A Survey on Image Data Augmentation for Deep Learning. *Journal of Big Data*, 6(1):60, 2019. ISSN 2196-1115. doi: 10.1186/s40537-019-0197-0. URL https://doi.org/10.1186/s40537-019-0197-0.

K. Simonyan and A. Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv 1409.1556*, 09 2014.

J. Sled and B. Pike. *Understanding Intensity Non-Uniformity in MRI*, volume 1496, pages 614–622. 08 1998. ISBN 978-3-540-65136-9. doi: 10.1007/BFb0056247.

S.M. Smith. Fast Robust Automated Brain Extraction. *Human Brain Mapping*, 17(3):143–155, 2002. doi: 10.1002/hbm.10062.

S.M. Smith, M. Jenkinson, M.W. Woolrich, C.F. Beckmann, T.E.J. Behrens, H. Johansen-Berg, P.R. Bannister, M. De Luca, I. Drobnjak, D.E. Flitney, R.K. Niazy, J. Saunders, J. Vickers, Y. Zhang, N. De Stefano, J.M. Brady, and P.M. Matthews. Advances in Functional and Structural MR Image Analysis and Implementation as FSL. *NeuroImage*, 23:S208 – S219, 2004. ISSN 1053-8119. doi: https://doi.org/10.1016/j.neuroimage.2004.07.051. URL http://www.sciencedirect.com/science/article/pii/S1053811904003933. Mathematics in Brain Imaging.

N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *J. Mach. Learn. Res.*, 15(1):1929–1958, January 2014. ISSN 1532-4435.

G. Stockman and L.G. Shapiro. *Computer Vision*. Prentice Hall PTR, USA, 1st edition, 2001. ISBN 0130307963.

F. Sultana, A. Sufian, and P. Dutta. Evolution of Image Segmentation using Deep Convolutional Neural Network: A Survey. *Knowledge-Based Systems*, 201-202, 05 2020. doi: 10.1016/j.knosys.2020.106062.

C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going Deeper with Convolutions, 2014.

N. Tomar. UNET Implementation in TensorFlow using Keras API, Feb 2021. URL https://idiotdeveloper.com/unet-implementation-in-tensorflow-using-keras-api/.

L. Torrey and J. Shavlik. *Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Technique*. IGI Global, 2010. doi: http://doi:10.4018/978-1-60566-766-9.ch011.

S. Van der Walt, J.L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J.D. Warner, N. Yager, E. Gouillart, and T. Yu. Scikit-image: Image Processing in Python. *PeerJ*, 2:e453, 2014.

J. Virhammar, M. Warntjes, K. Laurell, and E.M. Larsson. Quantitative MRI for Rapid and User-Independent Monitoring of Intracranial CSF Volume in Hydrocephalus. *American Journal of Neuroradiology*, 37, 12 2015. doi: 10.3174/ajnr.A4627.

P. Virtanen, R. Gommers, T.E. Oliphant, M. Haberland, and et al. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020. doi: 10.1038/s41592-019-0686-2.

B. Warf. Hydrocephalus in Uganda: The Predominance of Infectious Origin and Primary Management with Endoscopic Third Ventriculostomy. *Journal of neurosurgery*, 102:1–15, 02 2005. doi: 10.3171/ped.2005.102.1.0001.

B.C. Warf. Educate One to Save a Few. Educate a Few to Save Many. *World Neurosurg*, 79(2 Suppl):S15.e15–8, 2013. ISSN 1878-8750. doi: 10.1016/j.wneu.2010.09.021.

J.B.M. Warntjes, O. Dahlqvist Leinhard, J. West, and P. Lundberg. Rapid Magnetic Resonance Quantification on the Brain: Optimization for Clinical Usage. *Magnetic resonance in medicine : official journal of the Society of Magnetic Resonance in Medicine / Society of Magnetic Resonance in Medicine*, 60:320–9, 08 2008. doi: 10.1002/mrm.21635.

K. Weiss, T.M. Khoshgoftaar, and D. Wang. A Survey of Transfer Learning. *Journal of Big Data*, 3(1), 2016. doi: 10.1186/s40537-016-0043-6.

S.C. Wong, A. Gatt, V. Stamatescu, and M.D. McDonnell. Understanding Data Augmentation for Classification: When to Warp? In *2016 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, 2016.

I.C. Wright, P.K. McGuire, J.B. Poline, J.M. Travere, R.M. Murray, C.D. Frith, R.S.J. Frackowiak, and K.J. Friston. A Voxel-Based Method for the Statistical Analysis of Gray and White Matter Density Applied to Schizophrenia. *NeuroImage*, 2(4):244 – 252, 1995. ISSN 1053-8119. doi: https://doi.org/10.1006/nimg.1995.1032. URL `http://www.sciencedirect.com/science/article/pii/S1053811985710324`.

F. Yamashita, M. Sasaki, S. Takahashi, H. Matsuda, K. Kudo, S. Narumi, Y. Terayama, and T. Asada. Detection of Changes in Cerebrospinal Fluid Space in Idiopathic Normal Pressure Hydrocephalus using Voxel-Based Morphometry. *Neuroradiology*, 52:381–6, 10 2009. doi: 10.1007/s00234-009-0610-z.

T. Zhan, J. Zhang, L. Xiao, Y. Chen, and Z. Wei. An Improved Variational Level Set Method for MR Image Segmentation and Bias Field Correction. *Magnetic resonance imaging*, 31, 12 2012. doi: 10.1016/j.mri.2012.08.002.

Y. Zhang, M. Brady, and S. Smith. Segmentation of Brain MR Images through a Hidden Markov Random Field Model and the Expectation-Maximization Algorithm. *IEEE Trans Med Imaging*, 20(1):45–57, 2001. ISSN 0278-0062 (Print) 0278-0062. doi: 10.1109/42.906424.

H.K. Zhao, T. Chan, B. Merriman, and S. Osher. A Variational Level Set Approach to Multiphase Motion. *Journal of Computational Physics*, 127(1):179–195, 1996. ISSN 0021-9991. doi: https://doi.org/10.1006/jcph.1996.0167. URL `https://www.sciencedirect.com/science/article/pii/S0021999196901679`.

A

Euler-Lagrange Derivation for Chan-Vese Method

A.1. Euler-Lagrange Derivation for the Level Set

The energy minimized in this method is given as follows:

$$\begin{aligned} E = \Big[\mu \int_{\Omega} \delta(\phi(\mathbf{x}))|\nabla\phi(\mathbf{x})|d\mathbf{x} + \nu \int_{\Omega} H(\phi(\mathbf{x}))d\mathbf{x} + \\ \lambda_1 \int_{\Omega} |f(\mathbf{x}) - c_1|^2 H(\phi(\mathbf{x}))d\mathbf{x} + \lambda_2 \int_{\Omega} |f(\mathbf{x}) - c_2|^2 (1 - H(\phi(\mathbf{x})))d\mathbf{x}\Big]. \end{aligned} \tag{A.1}$$

The energy is minimized with respect to c_1, c_2 and ϕ. When c_1 and c_2 are fixed, the energy is minimized with respect to ϕ via obtaining the Euler-Lagrange equation. The Euler-Lagrange equation is obtained by computing:

$$\frac{\partial\phi}{\partial t} = -\frac{\partial E(\phi)}{\partial\phi}. \tag{A.2}$$

The right side is computed by taking the Gateaux derivative. In this derivative ϕ is changed with $\phi + \sigma\eta$ where η is a test function and σ is small factor that should be set to zero.

$$\frac{\partial E(\phi)}{\partial\phi} = \frac{\partial E(\phi + \sigma\eta)}{\partial\sigma}\Big|_{\sigma=0}. \tag{A.3}$$

The derivation of the Gateaux derivative of the energy function is as follows:

$$\begin{aligned} \frac{\partial E(\phi+\sigma\eta)}{\partial\sigma}\Big|_{\sigma=0} &= \frac{\partial}{\partial\sigma}\Big[\mu \int_{\Omega} \delta(\phi+\sigma\eta)|\nabla(\phi+\sigma\eta)|d\mathbf{x} + \nu \int_{\Omega} H(\phi+\sigma\eta)d\mathbf{x} + \\ &\quad \lambda_1 \int_{\Omega} |f - c_1|^2 H(\phi+\sigma\eta)d\mathbf{x} + \lambda_2 \int_{\Omega} |f - c_2|^2 (1 - H(\phi+\sigma\eta))d\mathbf{x}\Big]\Big|_{\sigma=0} \\ &= \mu \int_{\Omega} \frac{\partial}{\partial\sigma}\delta(\phi+\sigma\eta)|\nabla(\phi+\sigma\eta)|d\mathbf{x} + \nu \int_{\Omega} \eta\delta(\phi+\sigma\eta)d\mathbf{x} + \\ &\quad \lambda_1 \int_{\Omega} \eta|f - c_1|^2\delta(\phi+\sigma\eta)d\mathbf{x} - \lambda_2 \int_{\Omega} \eta|f - c_2|^2\delta(\phi+\sigma\eta)d\mathbf{x}\Big]\Big|_{\sigma=0} \end{aligned} \tag{A.4}$$

Since taking the derivative of the first term is difficult, we will will take a closer look at the derivation separately. For simplification we write $\phi+\sigma\eta$ as ϕ_σ

$$\begin{aligned}
\mu\int_\Omega \frac{\partial}{\partial\sigma}\delta(\phi_\sigma)|\nabla(\phi_\sigma)|d\mathbf{x} &= \mu\int_\Omega \frac{\partial}{\partial\sigma}(\delta(\phi_\sigma))|\nabla(\phi_\sigma)| + \delta(\phi_\sigma)\frac{\partial}{\partial\sigma}(|\nabla(\phi_\sigma)|)d\mathbf{x}\\
&= \mu\int_\Omega \delta'(\phi_\sigma)\eta|\nabla(\phi_\sigma)| + \delta(\phi_\sigma)\frac{\partial}{\partial\sigma}\sqrt{\langle\nabla\phi_\sigma,\nabla\phi_\sigma\rangle}d\mathbf{x}\\
&= \mu\int_\Omega \delta'(\phi_\sigma)\eta|\nabla(\phi_\sigma)| + \delta(\phi_\sigma)\frac{\frac{\partial}{\partial\sigma}\langle\nabla\phi_\sigma,\nabla\phi_\sigma\rangle}{2|\nabla\phi_\sigma|}d\mathbf{x}\\
&= \mu\int_\Omega \delta'(\phi_\sigma)\eta|\nabla(\phi_\sigma)| + \delta(\phi_\sigma)\frac{2\langle\nabla\phi,\nabla\eta\rangle + 2\sigma\langle\nabla\eta,\nabla\eta\rangle}{2|\nabla\phi_\sigma|}d\mathbf{x}\\
&\overset{\sigma=0}{=} \mu\int_\Omega \delta'(\phi)\eta|\nabla(\phi)| + \delta(\phi)\frac{\langle\nabla\phi,\nabla\eta\rangle}{|\nabla\phi|}d\mathbf{x}\\
&\overset{\text{prod. rule of div.}}{=} \mu\int_\Omega \delta'(\phi)\eta|\nabla(\phi)| + \frac{\langle\delta(\phi)\nabla\phi,\nabla\eta\rangle}{|\nabla\phi|}d\mathbf{x}\\
&\overset{\text{See Equation A.6}}{=} \mu\int_\Omega \delta'(\phi)\eta|\nabla(\phi)| - \delta(\phi)\mathrm{div}\Big(\frac{\nabla\phi}{|\nabla\phi|}\Big)\eta - \delta'(\phi)\frac{|\nabla\phi|^2}{|\nabla\phi|}\eta d\mathbf{x}\\
&= \mu\int_\Omega -\delta(\phi)\mathrm{div}\Big(\frac{\nabla\phi}{|\nabla\phi|}\Big)\eta d\mathbf{x}
\end{aligned} \tag{A.5}$$

with

$$\begin{aligned}
&\int_\Omega \frac{\langle\delta(\phi)\nabla\phi,\nabla\eta\rangle}{|\nabla\phi|}d\mathbf{x}\\
&\overset{\text{Green's first theorem}}{=} \int_\Omega \nabla*(\delta(\phi)\eta\frac{\nabla\phi}{|\nabla\phi|})d\mathbf{x} - \int_\Omega \eta\nabla*(\delta(\phi)\frac{\nabla\phi}{|\nabla\phi|})d\mathbf{x}\\
&\overset{\text{Divergence theorem}}{=} \int_{\partial\Omega}\delta(\phi)\eta\frac{\nabla\phi}{|\nabla\phi|}*\mathbf{n}d\Omega - \int_\Omega \eta\nabla\delta(\phi)*\frac{\nabla\phi}{|\nabla\phi|}d\mathbf{x} - \int_\Omega \eta\delta(\phi)\nabla*(\frac{\nabla\phi}{|\nabla\phi|})d\mathbf{x}\\
&= \int_{\partial\Omega}\delta(\phi)\eta\frac{\nabla\phi}{|\nabla\phi|}*\mathbf{n}d\Omega - \int_\Omega \eta\delta'(\phi)\nabla\phi*\frac{\nabla\phi}{|\nabla\phi|}d\mathbf{x} - \int_\Omega \eta\delta(\phi)\mathrm{div}(\frac{\nabla\phi}{|\nabla\phi|})d\mathbf{x}\\
&= \int_{\partial\Omega}\delta(\phi)\eta\frac{\nabla\phi}{|\nabla\phi|}*\mathbf{n}d\Omega - \int_\Omega \eta\delta'(\phi)\frac{|\nabla\phi|^2}{|\nabla\phi|}d\mathbf{x} - \int_\Omega \eta\delta(\phi)\mathrm{div}(\frac{\nabla\phi}{|\nabla\phi|})d\mathbf{x}
\end{aligned} \tag{A.6}$$

The first term in this last equation leads to the boundary condition when setting to zero. Now the last term of Equation A.5 is put into Equation A.4.

$$\begin{aligned}
\frac{\partial E(\phi+\sigma\eta)}{\partial\sigma}\Big|_{\sigma=0} &= \mu\int_\Omega \frac{\partial}{\partial\sigma}\delta(\phi+\sigma\eta)|\nabla(\phi+\sigma\eta)|d\mathbf{x} + \nu\int_\Omega \eta\delta(\phi+\sigma\eta)d\mathbf{x} +\\
&\quad \lambda_1\int_\Omega \eta|f-c_1|^2\delta(\phi+\sigma\eta)d\mathbf{x} - \lambda_2\int_\Omega \eta|f-c_2|^2\delta(\phi+\sigma\eta)d\mathbf{x}\Big|_{\sigma=0}\\
&= \mu\int_\Omega -\delta(\phi)\mathrm{div}\Big(\frac{\nabla\phi}{|\nabla\phi|}\Big)\eta d\mathbf{x} + \nu\int_\Omega \eta\delta(\phi)d\mathbf{x} +\\
&\quad \lambda_1\int_\Omega \eta|f-c_1|^2\delta(\phi)d\mathbf{x} - \lambda_2\int_\Omega \eta|f-c_2|^2\delta(\phi)d\mathbf{x} + \int_{\partial\Omega}\delta(\phi)\eta\frac{\nabla\phi}{|\nabla\phi|}*\mathbf{n}d\Omega\\
&= \int_\Omega \eta\Big(-\mu\delta(\phi)\mathrm{div}\Big(\frac{\nabla\phi}{|\nabla\phi|}\Big)d\mathbf{x} + \nu\delta(\phi) +\\
&\quad \lambda_1|f-c_1|^2\delta(\phi) - \lambda_2|f-c_2|^2\delta(\phi)\Big)d\mathbf{x} + \int_{\partial\Omega}\delta(\phi)\eta\frac{\nabla\phi}{|\nabla\phi|}*\mathbf{n}d\Omega
\end{aligned} \tag{A.7}$$

Since this is true for every test function η integrated over the boundary Ω we can remove the integral and the η function. Together with Equation A.9 and the boundary condition obtained from Equation A.6 the Euler-Lagrange method has now been obtained.

$$\begin{cases}
\frac{\partial\phi}{\partial t} &= \delta\Big[\mu\mathrm{div}\Big(\frac{\nabla\phi}{|\nabla\phi|}\Big) - \nu - \lambda_1(f-c_1)^2 + \lambda_2(f-c_2)^2\Big] \quad \text{in } \Omega,\\
\frac{\delta(\phi)}{|\nabla\phi|}\frac{\partial\phi}{\partial\vec{\mathbf{n}}} &= 0 \quad \text{on } \partial\Omega.
\end{cases} \tag{A.8}$$

A.2. Optimal Cluster Values

Every iteration the optimal values for c_1 and c_2 need to be found. The optimal values are when:

$$\frac{\partial c_i}{\partial t} = -\frac{\partial E}{\partial c_i} = 0. \tag{A.9}$$

Then for value c_1

$$\frac{\partial E}{\partial c_1} = \lambda_1 \int_\Omega 2(f - c_1) H(\phi) d\mathbf{x} = 0 \tag{A.10}$$

Thus

$$c_1 \int_\Omega 2H(\phi) d\mathbf{x} = \int_\Omega 2fH(\phi) d\mathbf{x} \tag{A.11}$$

which leads us to

$$c_1 = \frac{\int_\Omega fH(\phi) d\mathbf{x}}{\int_\Omega fH(\phi) d\mathbf{x}} \tag{A.12}$$

A similar derivation is found for the optimal value of c_2

B

Euler-Lagrange Derivation for the Level Set of Li's Method

The gradient flow equation for the level set of Li's method is derived similarly as for the Chan-Vese model. The energy minimized in this method is given as follows:

$$E = \int \left(\sum_{i=1}^{N} \int_{\Omega} K(\mathbf{x}-\mathbf{y}) |I(\mathbf{y}) - b(\mathbf{x})c_i|^2 d\mathbf{y} \right) u_i(\phi(\mathbf{x})) d\mathbf{x} + \nu \int |\nabla H(\phi)| d\mathbf{x} + \mu \int p(|\nabla \phi|) d\mathbf{x}. \tag{B.1}$$

The energy is minimized with respect to c_i, b and ϕ. The Euler-Lagrange equation for ϕ is obtained by computing the Gateaux derivative:

$$\frac{\partial \phi}{\partial t} = -\frac{\partial E(\phi)}{\partial \phi} = -\frac{\partial E(\phi + \sigma\eta)}{\partial \sigma}\Big|_{\sigma=0}. \tag{B.2}$$

The derivation of the Gateaux derivative of the energy function is as follows:

$$\begin{aligned}
\frac{\partial E(\phi+\sigma\eta)}{\partial \sigma}\Big|_{\sigma=0} &= \frac{\partial}{\partial \sigma}\Big[\int \left(\sum_{i=1}^{N} \int_{\Omega} K(\mathbf{x}-\mathbf{y}) |I(\mathbf{y}) - b(\mathbf{x})c_i|^2 d\mathbf{y} \right) u_i(\phi+\sigma\eta) d\mathbf{x} + \\
&\quad \nu \int |\nabla H(\phi+\sigma\eta)| d\mathbf{x} + \mu \int p(|\nabla(\phi+\sigma\eta)|) d\mathbf{x} \Big] \Big|_{\sigma=0} \\
&= \Big[\int \left(\sum_{i=1}^{N} \int_{\Omega} K(\mathbf{x}-\mathbf{y}) |I(\mathbf{y}) - b(\mathbf{x})c_i|^2 d\mathbf{y} \right) u_i'(\phi+\sigma\eta)\eta d\mathbf{x} + \\
&\quad \nu \int \delta(\phi+\sigma\eta) \mathrm{div}\Big(\frac{\nabla(\phi+\sigma\eta)}{|\nabla(\phi+\sigma\eta)|} \Big) d\mathbf{x} + \mu \frac{\partial}{\partial \sigma} \int p(|\nabla(\phi+\sigma\eta)|) d\mathbf{x} \Big] + \int_{\partial\Omega} \delta(\phi+\sigma\eta)\eta \frac{\nabla(\phi+\sigma\eta)}{|\nabla(\phi+\sigma\eta)|} * \mathbf{n} d\Omega \Big|_{\sigma=0}
\end{aligned} \tag{B.3}$$

Since the membership functions are defined as $u_1(\phi) = H(\phi)$ and $u_2(\phi) = 1 - H(\phi)$, $u_1' = \delta(\phi)$ and $u_2' = -\delta(\phi)$. The second term was already derived in Appendix A.1. Thus, only the last term needs to be investigated.

$$\begin{aligned}
\frac{\partial}{\partial\sigma}\int p(|\nabla(\phi+\sigma\eta)|)d\mathbf{x}\Big|_{\sigma=0} &= \int \frac{\partial}{\partial\sigma} p(|\nabla(\phi+\sigma\eta)|)d\mathbf{x}\Big|_{\sigma=0} \\
&= \int p'(|\nabla(\phi+\sigma\eta)|)\frac{\partial}{\partial\sigma}(|\nabla(\phi+\sigma\eta)|)d\mathbf{x}\Big|_{\sigma=0} \\
&= \int p'(|\nabla(\phi+\sigma\eta)|)\frac{\partial}{\partial\sigma}\sqrt{(\nabla(\phi+\sigma\eta))^2}d\mathbf{x}\Big|_{\sigma=0} \\
&= \int p'(|\nabla(\phi+\sigma\eta)|)\frac{2\nabla(\phi+\sigma\eta)\frac{\partial}{\partial\sigma}(\nabla(\phi+\sigma\eta))}{2\sqrt{(\nabla(\phi+\sigma\eta))^2}}d\mathbf{x}\Big|_{\sigma=0} \\
&= \int \frac{p'(|\nabla(\phi+\sigma\eta)|)}{|\nabla(\phi+\sigma\eta)|} * \nabla\eta d\mathbf{x}\Big|_{\sigma=0} \\
&\overset{\text{Green's first theorem}}{=} \int \mathrm{div}(\frac{\nabla(\phi+\sigma\eta)p'(|\nabla(\phi+\sigma\eta)|)}{|\nabla(\phi+\sigma\eta)|}\eta) - \eta\mathrm{div}(\frac{\nabla(\phi+\sigma\eta)p'(|\nabla(\phi+\sigma\eta)|)}{|\nabla(\phi+\sigma\eta)|})d\mathbf{x}\Big|_{\sigma=0} \\
&\overset{\text{Divergence theorem}}{=} \int_{\partial\Omega}\frac{\nabla(\phi+\sigma\eta)p'(|\nabla(\phi+\sigma\eta)|)}{|\nabla(\phi+\sigma\eta)|}\eta * \mathbf{n}d\Omega - \int \eta\mathrm{div}(\frac{\nabla(\phi+\sigma\eta)p'(|\nabla(\phi+\sigma\eta)|)}{|\nabla(\phi+\sigma\eta)|})d\mathbf{x}\Big|_{\sigma=0} \\
&= \int_{\partial\Omega}\frac{\nabla\phi p'(|\nabla(\phi)|)}{|\nabla(\phi)|}\eta * \mathbf{n}d\Omega - \int \eta\mathrm{div}(\nabla\phi\frac{p'(|\nabla(\phi)|)}{|\nabla(\phi)|})d\mathbf{x}
\end{aligned} \tag{B.4}$$

The first term of this last line is part of the boundary condition. The second term gives us the term in the Euler-Lagrange equation. Thus, the complete Gateaux derivative of the energy function is

$$\begin{aligned}
\frac{\partial E(\phi+\sigma\eta)}{\partial\sigma}\Big|_{\sigma=0} = \int\Big(\sum_{i=1}^{N}\int_\Omega K(\mathbf{x}-\mathbf{y})|I(\mathbf{y}) - b(\mathbf{x})c_i|^2 d\mathbf{y}\Big)u_i'(\phi)\eta d\mathbf{x} + \\
\nu\int\delta(\phi)\mathrm{div}\Big(\frac{\nabla\phi}{|\nabla\phi|}\Big)d\mathbf{x} + \nu\int_{\partial\Omega}\delta(\phi)\eta\frac{\nabla\phi}{|\nabla\phi|} * \mathbf{n}d\Omega \\
+\mu\int_{\partial\Omega}\frac{\nabla\phi p'(|\nabla(\phi)|)}{|\nabla(\phi)|}\eta * \mathbf{n}d\Omega - \mu\int\eta\mathrm{div}(\frac{\nabla\phi p'(|\nabla(\phi)|)}{|\nabla(\phi)|})d\mathbf{x}
\end{aligned} \tag{B.5}$$

Similarly to the derivation of the gradient flow equation of the Chan-Vese method, the test function η and the integral can be removed to obtain the gradient flow equation given in Equation 4.8.

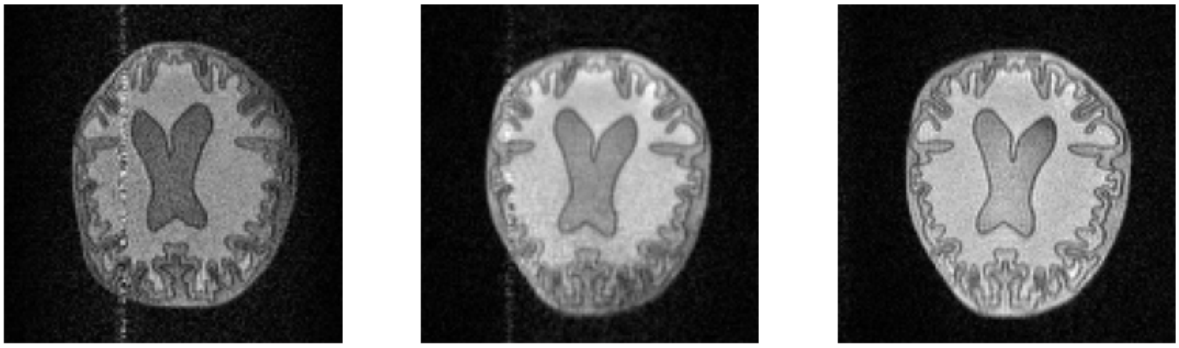
C

Segmentation Results of Phantom Scans

In this chapter the results of the k-means algorithm on the original and the bias corrected scans of the phantom are shown. In both cases k is three, since the images are segmented into three regions. In both cases the initial cluster values are [0,1,2]. In Figure C.1 the results of the k-means algorithm on the original scans are shown. The scans are corrected for their bias by using SimpleITK's bias field correction algorithm [Beare et al., 2018]. The resulting images are shown in Figure C.2 and the k-means segmentation of these images are depicted in Figure C.3.

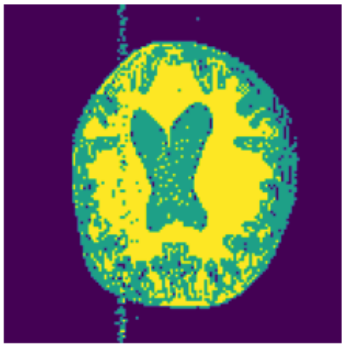
(a) K-means segmentation of scan 1 (b) K-means segmentation of scan 2 (c) K-means segmentation of scan 3

Figure C.1: The results of k-means algorithm applied to the phantom scans of Figure 5.2.

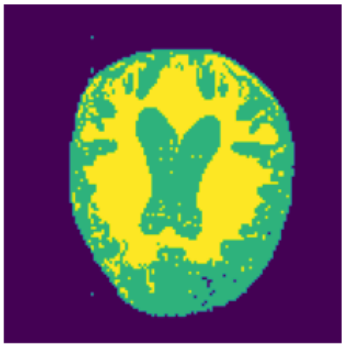


(a) Bias corrected image of scan 1 (b) Bias corrected image of scan 2 (c) Bias corrected image of scan 3

Figure C.2: The results of the bias corrected image of the automatic bias correction algorithm of sitk applied to the phantom scans of Figure 5.2.



(a) K-means segmentation of Figure C.2a



(b) K-means segmentation of Figure C.2b

(c) K-means segmentation of Figure C.2c

Figure C.3: The results of the k-means algorithm applied to the bias corrected images of Figure C.2.