VILNIUS GEDIMINAS
TECHNICAL UNIVERSITY
FACULTY OF ELECTRONICS
DEPARTMENT OF ELECTRONIC SYSTEMS

Arūnas BUTKUS

SEGMENTATION OF STROKE REGION IN MRI IMAGES

INSULTO SRIČIŲ SEGMENTAVIMAS MAGNETINIO REZONANSO VAIZDUOSE

Master Thesis

Electronics Study Programme, State Code 621H61003
Computerized Electronic Systems Specialization
Electronics Engineering Study Area

Vilnius, 2018

Vilniaus Gedimino technikos universitetas
Elektronikos fakultetas
Elektroninių sistemų katedra

ISBN ISSN
Egz. sk. .....
Data .....-.....-.....

Antrosios pakopos studijų **Elektronikos inžinerijos** programos magistro baigiamasis darbas

Pavadinimas	**Insulto sričių segmentavimas magnetinio rezonanso vaizduose**
Autorius	**Arūnas Butkus**
Vadovas	**Andrius Ušinskas**

Kalba: anglų

Anotacija

Magistrinio darbo tikslas ištirti esamus galvos smegenų analizės metodus, detaliai išanalizuoti du metodus, vieną iš jų patobulinti, kad veiktų su prieinamais duomenimis. Sukurti alternatyvų metodą esamiems. Ištirti efektyvumą esamų metodų ir palyginti su siūlomu metodu.

Magistro baigiamąjį darbą sudaro 59 p., 37 iliustr. ir 4 lent. be priedų; 56 bibl., 3 priedai.

Prasminiai žodžiai: MRI, smegenys, insultas, segmentavimas, regiono auginimas

Vilnius Gediminas Technical University
Faculty of Electronics
Department of Electronic Systems

ISBN ISSN
Copies No. .....
Date .....-.....-.....

Master Degree Studies **Electronics Engineering** study programme Master Graduation Thesis

Title	**Segmentation of Stroke Region in MRI Images**
Author	**Arūnas Butkus**
Academic supervisor	**Andrius Ušinskas**

Thesis language: English

Annotation

Goal of master thesis is to analyse existing methods for brain analysis, in detail analyse two methods, improve one method, so that it would work with available data. Create an alternative method to the existing ones. Test the effectiveness of existing improved method compared to proposed method.

Master thesis paper consists of 59 pages, 37 figures and 4 tables not counting appendices; 56 references, 3 appendices.

Keywords: MRI, brain, lesion, segmentation, region growing

CONTENTS

VARIABLES AND ABBREVIATIONS

Variables

μ_i	–	original signal in bias correction calculation
n_i	–	Gaussian noise in bias correction calculation
ρ_i	–	bias in bias correction calculation
y_i	–	resulting signal in bias correction calculation
$GM_{Affected}$	–	grey matter TMP of input MRI scan affected hemisphere
$WM_{Affected}$	–	white matter TMP of input MRI scan affected hemisphere
$CSF_{Affected}$	–	CSF TMP of input MRI scan affected hemisphere
$GM_{Unaffected}$	–	grey matter TMP of input MRI scan mirrored unaffected hemisphere
$WM_{Unaffected}$	–	white matter TMP of input MRI scan mirrored unaffected hemisphere
$CSF_{Unaffected}$	–	CSF TMP of input MRI scan mirrored unaffected hemisphere
GM_{Prior}	–	grey matter TMP from template
WM_{Prior}	–	white matter TMP from template
CSF_{Prior}	–	CSF TMP from template

Abbreviations

CSF	–	Cerebrospinal Fluid
DSC	–	Dice-Sorensen Coefficient
FWHM	–	Full Width at Half Maximum
MRI	–	Magnetic Resonance Imaging
SPM	–	Statistical Parametric Mapping
TPM	–	Tissue Probability Map

INTRODUCTION

This work analyses, implements and tries to improve upon brain analysis algorithms. Processing of neural images is important in neuroscience as damages to the brain causes cognitive problems for people. Knowing the affected region can help doctor to decide on the course of actions to take.

There is a variety of ways brain can be affected – from tumours to strokes. Most algorithms found during analysis deals with tumours, however here goal is to find a region of brain stroke caused ischemic lesion.

During brain stroke, blood spills onto the brain. This damages grey and white matter. In MRI scans this shows up as dark regions, where there should be bright regions. Issue in delineating them is that there is also CSF in brain that has similar visual characteristics as a lesion, which makes it difficult to discern in from CSF.

Topicality and Aim of the Work

Aim of the work is to analyse and improve upon an existing method for lesion delineation, eventually making an original method with as little interaction required from user as possible. It is topical, because to date there is no precise method for delineating a brain stroke caused ischemic lesion region in an MRI scan that is precise, or fast, or reliable, or does not rely on training for a specific case.

Tasks of the Work

This work constitutes of three main tasks:

- Make literature review and analyse in detail two algorithms – test their effectiveness for available data. Test algorithms that automatically delineate brain stroke induced ischemic lesions.
- Improve one method, so that it would be capable of operating on available test data and attempt to get best results possible. Find what parameters are influencing the result and find best values for them.
- Create own algorithm that would be able to provide comparable results to analysed methods. Test its effectiveness.

Applied Methods of Investigation and Analysis

For testing MRI scans are used, each having a hand delineated lesion region available. To test the efficacy of algorithm DSC formula is applied, which shows how much two data sets coincide. In this work analysis of delineation methods is performed using MATLAB environment. Changing, where possible, processing parameters and applying DSC to determine how

effective their variations are provides a way of fine tuning the algorithms in an attempt to find a best result.

Novelty and Practical Value of the Work

While the idea of automatically processing MRI scans is not new, automation ideas likely was around since the invention of an MRI machine in 1977, it has now experienced resurgence in popularity, similar to neural networks, as cheap and large computational power became easily available. And there is a demand for such methods, since hand delineation is an expensive time consuming process. If an algorithm capable of consistently and accurately delineating a lesion were to be developed many medical institutions would want to incorporate it.

The Scope of the Work

This work overviews a variety of methods for processing MRI images; analyzes in detail two of them; proposes an effective way of finding where an ischemic lesion region is located in brain MRI scan; employs region growing algorithm in an attempt to delineate this found region precisely – as similar to hand delineation as possible; tests the effectiveness of existing method and compares it to the proposed method; attempts to improve existing method so that it would work for available test cases.

This work contains 59 pages, 37 figures and 4 tables not counting appendices; 3 appendices, 56 references.

1 ANALYTICAL REVIEW OF THE BRAIN STROKE DETECTION IN MRI IMAGES

Studies in cognitive neuroscience and neurology provide knowledge of how specific regions of the brain affect cognition and behaviour. Early on these studies where done on cadavers, but recent advances in neuroimaging, like MRI, allow studies of live people, thus increasing sample size and giving easy to access digital images, which are then used in voxel-lesion symptom mapping [1–6]. Looking for a relation between brain structure and its function is achieved by utilizing tissue probability maps and specificaly leison maps when looking for lesion inpact on persons cognition. Lesion voxels within these scans need to be precise and reliable [7]. For this reason, typically, experts in the field hand-draw the lesion regions. This takes a large amount of time [8] and can also have errors due to subjective bias [9], but so far it is still used as a go to, reliable method for lesion areas identification in MRI scans.

Time consuming nature of manual lesion delineation and subsequent costs associated with it, spurred the search for semi-automated and fully-automated methods. Initial automatic methods used voxel-based morphometry – statistical technique that would find regions of abnormal tissue density/volume between populations to identify abnormal voxels in the T1-weighted scans. However, even though voxel-based morphometry can usually highlight rough area of where the lesion is located, the accuracy of the result is unsatisfactory for any further use [10].

Recent developements propose automated [11] and semi-automated [8] lesion delineation methods for T1-weighted scans that combine probabilistic tissue segmentation with outlier-based fuzzy clustering principles. Automated method uses a set of control subjects with healthy brains to make a tissue segmentation map giving probabilities of voxel being grey matter or white matter. These estimated probabilities are used to assign each voxel to a fuzzy set of tissue classes; resulting tissue classes are than combined and applying statistical thresholds provides voxels signifying lesion area. Semi-automated method uses probabilistic tissue segmentation to create 4 feature maps that contain information about tissue composition, shape, homogeneity and laterality for each voxel. These feature maps are used to construct a z-score map at which point user manually thresholds the map to get a final lesion delineation.

Of the two methods semi-automated method provides somewhat more reliable results, but it costs, on average, about 25 min per subject to complete the manual thresholding step [8]. Problem with the automated method is that it requires a control group of healthy control subjects for calibration, which is not always available. Another issue is when giving a large set of healthy subjects computation time increases significantly when trying to estimate probability maps of healthy brains. Also, both of these methods require custom preprocessing and/or segmentation procedures, which are likely to complicate their implementation and introduce another area for user error. Finally, both methods use some arbitrary thresholds that are affected by the control group size and contents for the automated method.

Another type of recent methods utilize supervised learning with classifiers, such as random forests [12] and extra-tree forests [13]. Advantage of these methods is that they do not require

control data of healthy brain scans, user interaction, or arbitrary thresholds to get lesion delineations for new scans. However, these methods usually require results from multiple MRI scan types to achieve optimal results; mostly relying on T2-weighted scans, fluid attenuated inversion recovery and/or diffusion weighted imaging sequences – all of which are rarely performed due to monetary and/or time costs. Most of the times it is T1-weighted scans that are performed.

1.1 Analysed and used established methods and technologies

This section looks through methods analysed and used in this work – what they are and what bases their operation. These include computation intensive spatial probability mapping toolbox [14] for MATLAB and Joseph C. Griffis automated lesion delineation algorithm. SPM toolbox is used to process T1-weighted MRI scans for further calculations. Griffis algorithms gave the an idea of how to attain a lesion probability map.

1.1.1 Spatial probability mapping toolbox

When attempting to segment brain images into certain classes two approaches can usually be taken: tissue classification or registration with a template. Tissue classification method assigns voxels to a tissue class according to their intensities. This is computed by taking intensities of each tissue class and characterizing them, which is usually done by choosing specific voxels to represent each class [15–21]. In an automatic approach this is done by first mapping the brain to some standard space and automatically selecting voxels that have a high probability of belonging to each of the classes. Alternative approach is to model the intensity distribution by a mixture of Gaussians, while using tissue probability maps to weigh the classification applying Bayes rule [22–24]. Registration with a template method involves some specific type of registration where template brain is warped to match the brain T1-weighted scan to be segmented [25–27]. Instead of volume matching methods, alternatively methods that are based on matching surfaces [28–30] would also work in this category. These methods work by overlaying regions that are predefined on the templates. This allows different structures to be identified automatically. Unified segmentation utilizes both – the tissue classification and registration with template methods – for more accurate segmentation of an MRI scan.

To start tissue classification the images are registered with tissue probability maps [31]. After registration these maps represent the prior probability of different tissue classes being found at each location in an image. Bayes rule is then used to combine these priors with tissue type probabilities derived from voxel intensities to provide posterior probability [32]. This procedure is circular – registration requires initial tissue classification, while tissue classification requires initial registration. To resolve this, a single generative model is used. This model also includes parameters that account for the image intensity non uniform variations in both segmentation and registration. Parameters in question are attained by an algorithm that alternates between classification, bias correction and registration steps which provides better results than serial

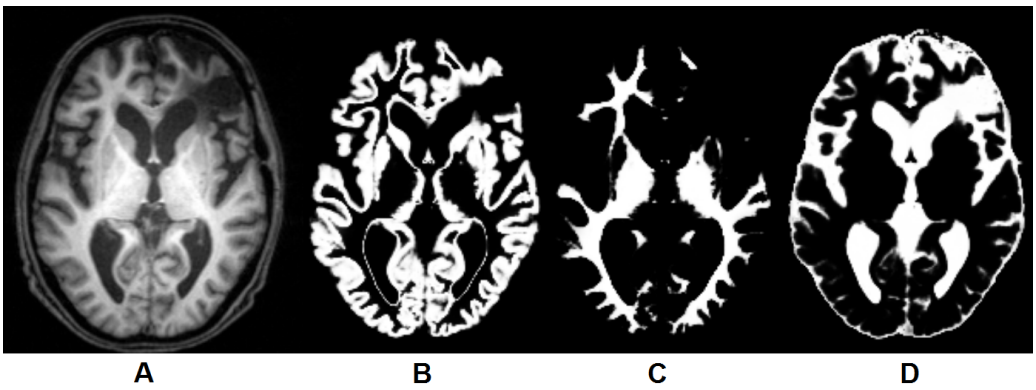


Fig. 1.1. Research results obtained using unified segmentation: patient MRI scan slice (A), grey matter TPM (B), white matter TPM (C) and CSF TPM (D)

application of each component.

To account for image nonuniformity, parametric bias correction is used. Many bias correction models are based on modelling the intensities of different tissues as a mixture of Gaussians. There are three commonly used models of how the bias interacts with noise. First is when the resulting signal (y_i) is an original signal (μ_i), scaled by some bias (ρ_i) with added Gaussian noise (n_i) that does not depend on bias [33]. This assumes that the noise if from MRI scanner itself (1.1).

$$y_i = \mu_i/\rho_i + n_i \tag{1.1}$$

Second model, which is used by the unified segmentation, is similar to first one except the noise is added before the signal is scaled, which implies that the noise is due to variation in tissue properties (1.2). There is an option of accounting for both tissue and scanner noise, which is likely to be a better option especially for the images that have large amounts of bias. However, unified segmentation used in the SPM toolbox uses single source model.

$$y_i = (\mu_i + n_i)/\rho_i \tag{1.2}$$

Third method starts by applying logarithmic transformation to the data first, which then allows multiplicative bias to be modelled as an additive effect in logarithmic space. The cost function for these approaches is related to the entropy of the distribution of log-transformed bias corrected data. As with the non-parametric model based on log-transformed data, low intensity voxels have to be excluded to avoid numerical problems. The generative model is similar to one given in equation (1.3), which then gives a exponential multiplication in resulting signal function (1.4)

$$\log y_i = \log mu_i - - \log \rho_i + n_i \tag{1.3}$$

$$y_i = \mu_i e^{n_i}/\rho_i \tag{1.4}$$

Another parameter used in segmentation is priors probabilities determination or simply template tissue probability map. Rather than using stationary values based on mixing proportions, additional information is used utilizing information from template TPM, preferably generated from similar nationality and build healthy subjects' brain images. Typically, priors are generated by registering a large set of brain images and averaging resulting tissue classes. This gives a set of tissue probability maps – white matter, grey matter and CSF – representing probabilities any of the matter types of being in specific areas of a brain.

1.1.2 Automatic lesion detection utilizing Griffis algorithm

There is a variety of automatic and semi-automatic algorithms for some form of lesion or some other specific brain region delineation [34–44], but use in this work started of by analysing Joseph C. Griffis algorithm, `lesion_gnb.m`, for lesion area delineation [34]. It is run using MATLAB 2018a and, as provided [45], works with MRI T1-weighted scans that have lesions in single hemisphere, stored as .nii files. Though in this particular form algorithm still requires operator interaction during runtime to supply extra configuration variables.

This is a supervised learning method for automatic delineation of stroke lesion in single patient T1-weighted scans. It employs naïve Bayes classification to identify lesion voxels based on information from tissue probability maps: white, grey and CSF TPMs attained by SPM toolbox and missing and abnormal tissue maps calculated from the other TPMs. This method as an input does not require any specially pre-processed images; only required pre-processing it performs itself with SPM toolbox. However it does require training of classifier employed in this algorithm to produce best results. Author believes training with 29 MRI scans makes classifier capable of delineating a lesion region (Fig. 1.2).

1.2 Lesion region detection

Lesions, which this works algorithm tries to detect, have low intensity values in MRI images and, effectively, look like cerebrospinal fluid (Fig. 1.3). Here queues are taken from Griffis algorithm and utilize missing tissue probability map. This map is obtained utilizing segmented MRI scan, mapping template TPM and symmetry of the warped segmented images.

Segmentation allows individual manipulation of white matter, grey matter and CSF in establishing the missing tissue map. As mentioned before, lesion area is similar to CSF which means after segmentation with SPM toolbox the lesion region is assigned to the CSF TPM (Fig. 1.1 D). And the simplest approach could be to subtract white and grey matter tissue probability maps of the template from the CSF TPM, but that would be inaccurate.

SPM toolbox only roughly attempts to map the MRI scan to the template (Fig. 1.4), it does not perfectly map to it and in the end there are a lot of differences. More so, the template TPMs are rough estimates only as they are created after averaging multiple healthy brains. This work does not do this and just use one provided with the toolbox, however for further optimization and specialization a specific type of template TPM could be created.

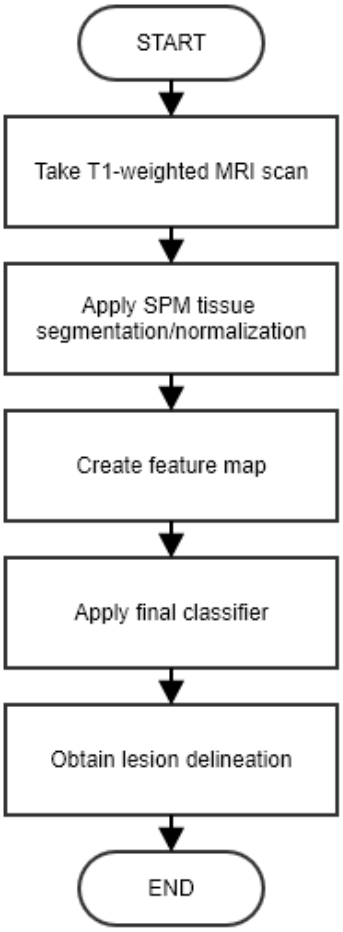


Fig. 1.2. Schematic illustration outlining the application of Griffis algorithm for predicting lesion delineation for new cases [34].

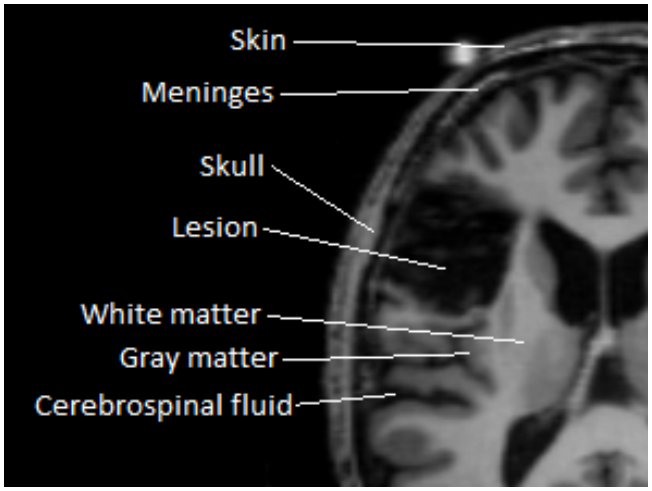


Fig. 1.3. Parts of a brain in a T1-weighted MRI scan. Lesions are low intensity areas similar to CSF.

This template that the algorithm uses is mostly symmetric, at the very least the general shape is, even though the ridges and grooves are not necessarily so. As the SPM tools map the MRI scan to that template, resulting TPMs also become roughly symmetric and most can be flipped left to right hemisphere giving another set of data. This data set represents symmetry,

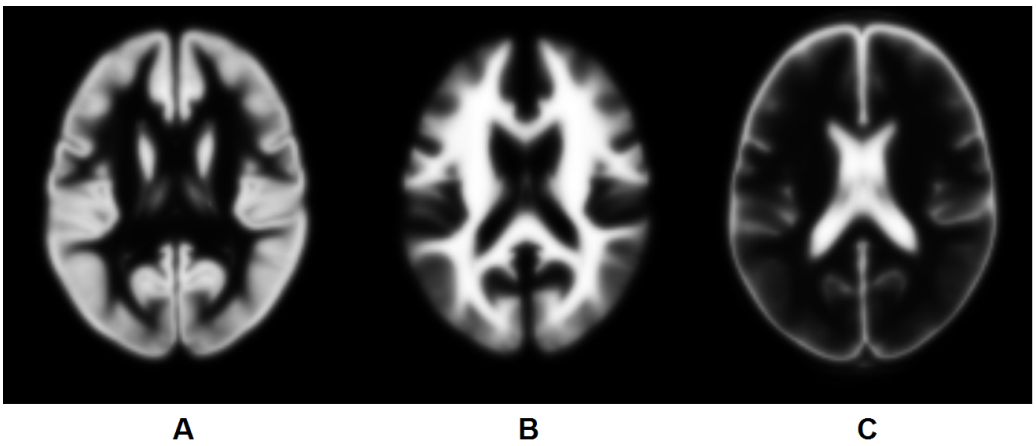


Fig. 1.4. Grey matter (A), white matter (B) and CSF TPMs in a template.

giving another thing to compare to searching for discrepancies.

Having all this data allows calculation of missing tissue probability map. However, at this point all these TPMs still have sharp ridges and that can introduce a lot of high amplitude noise in the missing TPM. So before that all formula input data is smoothed with FWHM algorithm giving blurrier images, without sharp ridges. As a result this makes missing TPM to have smoother transitions too. Even though this helps with further processing, this particular step prevents spotting small lesions as they just get smoothed out, though it is likely that during segmentation this information is lost anyway.

To begin with, lesion region intensity similarity to CSF is what causes a number of issues in spotting the lesions; in cases of small lesions even trained professional might have an issue. In testing it was noticed that a few cases where a small region on the edge of the brain, near skull, was considered by a professional to be a lesion. It was not possible to see most of them by untrained eye even knowing where they should be. Due to the nature of how missing tissue probability map is obtained in this works or Griffis methods, it is impossible to detect such a small lesion automatically with a reasonable degree of certainty. In fact, none of the analysed methods could. Distinguishing whether voxel belongs to CSF or lesion appears in larger lesions too, as CSF is not to be assigned to the lesion region.

1.3 Region growing

Having found a point belonging to a lesion region, next step is finding how big said region is. This work employs the obvious approach – grow the region from that point until healthy tissues are found. This, at least in theory, should work. This approach, considering that this work looks for a single region, is called single seed region growing.

Basic idea of region growing in a 3D image is to connect similar voxels to constitute a region. Single seed refers to the method when a single voxel is selected to grow from. In the most basic of variations, that single voxels' features, like intensity, are what is used to determine growth. How far to grow, is determined by the growth criteria. Simplest being intensity threshold – stop growing when intensity is a certain amount different from the original seed point intensity.

So if the neighbouring voxel satisfies growth criteria, it is assigned to the region [46].

Variation in how region growing behaves comes from how growth criteria are set up and what is checked. In many applications regions are grown in 2D space taking pixels to the right, left, above and below. With simple criteria and when region pixels absolutely must be contiguous this is enough. But a more robust check would be including the diagonals. This doubles the required calculations, but also helps determine true region when the region in question is textured unevenly. This works method circumvented this in growth criteria and do not utilize diagonals.

Adopting this simple version of region growing to 3D space is not difficult – it just adds 2 more voxels to the calculation in a basic neighbours approach, or 18 voxels if including the diagonals. In 3D robustness increase advantage of diagonals can be offset by the increase in computation intensity since moving to 3D space means that now up to 26 times (if history or some such is not used) voxels would need to be evaluated. If the growth criteria formula is complex enough this substantially increases computation times.

As for the growth criteria it is just a series of true/false checks. In an approach utilized by this work, two 3D cubes centred on tested voxel. First cube checks if any of its voxels meet threshold requirements returning true if any do. This achieves diagonals testing effect and also grows the region out based on the cube radius. Second cube checks how much of it meets threshold requirements. This test is intended to prevent region from growing into narrow CSF paths between other tissues.

1.4 Review conclusions

Analysis of literature indicates that ischemic brain stroke lesion delineation is imprecise even in best case scenarios. Often times it requires either pre-training the system before use, which takes a long time and oftentimes requires a control set of data that is not necessarily going to be available. Alternatively there are methods that can work on single MRI scan basis; however accuracy is an issue for these. This work focuses on an approach without any system pre-training with a goal to work for any number of MRI T1-weighted scans out-of-the-box.

First part of this works system uses SPM toolbox and equations from Griffis algorithms to obtain missing TPM. This then followed by a method for finding a single voxel within lesion region. And finishing with region growing, which is taken to avoid requiring to train the algorithm before use, like in a variety of other algorithms. Region growing does not require any previous knowledge about the system and operates solely from the given parameters.

2 ANALYSIS OF THE STROKE REGION DETECTION METHOD

This masters thesis work tries to automatically delineate lesion region by using region growing in the end, but the initial processing is similar to Griffis method. Like in his methods, in this work missing tissue feature map is extracted, but then instead of using predictors utilizing pre-trained neural networks, thresholds, clustering; finding the most likely point to contain a lesion and grow a region from that.

2.1 Determination of the point coordinates inside lesion region

Before region growing can be applied a seed voxel within lesion region needs to be found. For this Griffis' idea of using missing tissue feature map obtainable from TPMs generated with SPM is utilized.

2.1.1 Griffis approach

As Griffis algorithm is main inspiration of this work, it is believed that analysing it in its entirety is worth it. Code in itself is functional, but far from optimized. In its default state, it requires continuous supervision by the user.

When running the script at first it throws an input box asking, should the segmentation be performed on the MRI scan. Input is either "Y" or "N". In general, when running for the first time, user would have to perform the segmentation, thus selecting "Y". Next dialogue asks to pick a directory where all the files will be put. The one after that asks to provide the MRI scan file. Thus processing of the MRI scan begins with its segmentation into grey matter, white matter and cerebrospinal fluid. For this purpose unified segmentation algorithm, which is provided by the SPM toolbox, is used.

After unified segmentation is applied to the brain, multiple images are saved to the provided directory. To be exact it contains 20 files, including the original MRI scan. Of these initially 3 files are used – warped grey matter TPM, warped white matter TPM and warped CSF TPM. These TPMs are warped to match a template TPM taken from SPM toolbox. Warping is used because subsequent processing requires a comparison with a healthy brain half and/or template and for that the image dimensions and content positioning must align.

Next dialogue asks to provide directory containing MRI scan segments. This dialogue is thrown because this is the next piece of code to be executed after sectioning or skipping sectioning in the initial dialogue. This is the first of many optimization issues present in the script. So, in any case, selecting the directory with the segments throws a dialogue with the main settings selection for the algorithm.

This new dialogue has 6 fields to be edited. First one is for naming folder in which results will be placed, whose function is self-explanatory. Next is a selection whether to search for lesion in the left hemisphere, indicating by providing character "L", or in the right hemisphere,

indicating by character "R". This points out one more limitation of this algorithm – neither can it automatically determine which hemisphere has the lesion, nor find lesions that span both hemispheres. Third field is for FWHM smoothing kernel selection. Default suggested value is 8, which is used for performance/results reference later (Section. 3.1). Next field asks if whether the unaffected hemisphere should be used for lesion area detection. Fifth field is for selecting prior lesion probabilities. Last field is implicit mask for smoothing value which is 0 by default.

After gathering these values and doing some shuffling into a single class variable, subscript for feature extraction is run. Processing in this subscript starts with smoothing of all the elements to be used (segment TPMs and SPM template TPM) applying the smoothing kernel FWHM value provided in earlier dialogue. Following more variable shuffling, brain mask from SPM toolbox is loaded. This mask is used to filter out noise that can appear outside of brain before and after some processing steps; for example: to remove data points appearing outside of brain after smoothing TPMs. Next up, all required layers are made for the affected hemisphere. As an example: assuming that the left hemisphere is affected, three sets of three probability maps are created – grey matter, white matter and CSF. First set is the left hemisphere only of MRI scan TMPs with right hemisphere voxels filtered out. Second set is right hemisphere only of MRI scan TMPs flipped so that the right hemisphere overlays the left hemisphere. This one is used only if selected in the previous settings dialogue, however is created either way. Third set is left hemisphere of the template TPM.

Once all the TPMs are ready feature maps for missing tissue (F_1) and abnormal tissue (F_2) are created. The missing tissue map provides information about areas where brain tissues are missing. To find this area a feature of SPM segmentation is used – segmentation algorithm tends to classify chronic stroke lesions as CSF due to missing tissue voxels being assigned low grey or white matter probability values [8, 11]. So in the end the missing tissue area is obtained from the average of two image volumes using Eq. 2.1 and Eq. 2.2. "Affected" refers to input MRI scan affected hemisphere. "Unaffected" refers to input MRI scan unaffected hemisphere flipped onto affected hemisphere. "Prior" refers to template TPM provided by SPM toolbox.

$$(CSF_{Affected} - CSF_{Unaffected}) * ((GM_{Unaffected} + WM_{Unaffected}) - (GM_{Affected} + WM_{Affected})) \tag{2.1}$$

$$(CSF_{Affected} - CSF_{Prior}) * ((GM_{Prior} + WM_{Prior}) - (GM_{Affected} + WM_{Affected})) \tag{2.2}$$

Both of these equations are used if user indicates to use unaffected hemisphere in previous dialogue. Otherwise only Eq. 2.2 is used. When using both equations, before saving result as missing tissue map both the equations results are averaged. Averaging is rationalized by providing two points: using them as separate predictors would be sub-optimal since both volumes contain highly redundant information as both volumes are expected to have lesion

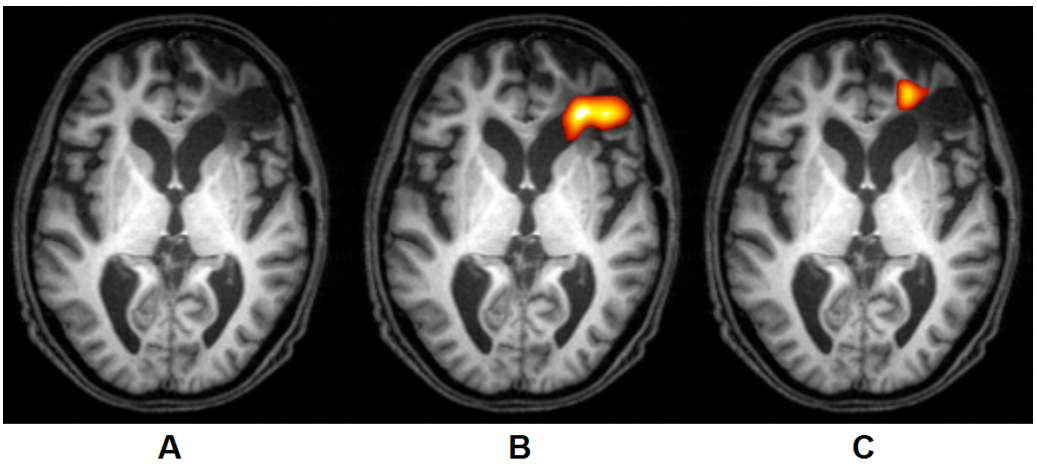


Fig. 2.1. Patient MRI scan slice (A), with missing tissue map overlaid (B) and abnormal tissue map overlaid(C)

values in same areas; averaging them retains the values of concordant voxels, while reducing the values of discordant voxels (e.g. false positives due to inter-hemispheric or inter-individual variability)[34].

Once the missing tissue map is created, next is abnormal tissue map (F_2). It provides information about abnormal tissue and is motivated by the observation that SPM segmentation tends to classify these tissues as grey matter due to the T1-weighted signal intensities being similar to those observed in healthy grey matter [10]. Overall system is same as in missing tissue extraction, just uses different equations (2.3, 2.4). An example of resulting missing and abnormal tissue maps is in figure 2.1.

$$(GM_{Affected} - GM_{Unaffected}) * (WM_{Unaffected} - WM_{Affected}) \tag{2.3}$$

$$(GM_{Affected} - GM_{Prior}) * (WM_{Prior} - WM_{Affected}) \tag{2.4}$$

At the end of feature extraction subscript, both feature maps are saved to files along with MATLAB features magnitudes variable and their corresponding indexes variable. This is followed by classification subscript. By default this uses previously trained classifier to predict full area affected by the lesion [47]. Training was performed by Griffis with 29 cases and tested on a single case left out of the training [34]. Prediction is executed utilizing MATLAB function "predict", supplying it with the trained system and features magnitudes variables saved in feature extraction subscript. This function computes the k-step ahead prediction returning, in this case, labels and posterior. However, these resulting feature maps are noisy and imprecise (Fig. 2.2B). To refine the results some post-processing algorithms are employed next.

Next dialogue to be thrown informs that delineation is complete and suggests applying post-processing. Standard "Y/N" choice is provided and if not performing any post-processing script exits. Data of interest at that point can be found in 4 files: "f1" – missing tissue feature map, "f2" – abnormal tissue feature map, "lesion_labels" – binary lesion probability map, "lesion_posterior" – lesion map with a gradient transition in normalized range. If selecting to

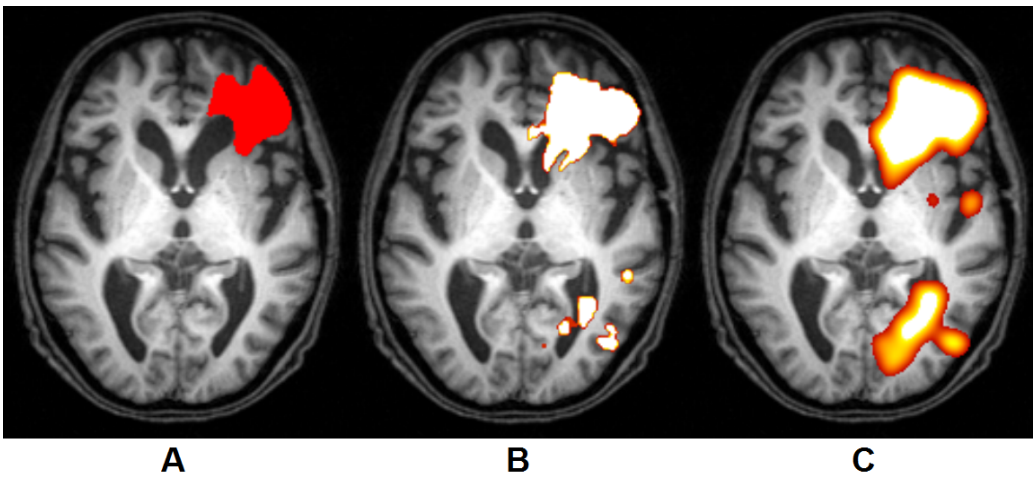


Fig. 2.2. Griffis lesion delineation algorithm results: depiction of hand drawn lesion area (A), lesion area pre post-processing (B), lesion area after post-processing (C)

apply post-processing, next dialogue asks whether FWHM smoothing should be performed and what size smoothing kernel should be applied; default is 8. Then a dialogue shown once more for picking if the implicit masking should be used. Third dialogue is for clustering algorithm – should minimum lesion cluster size algorithm be applied and what is the minimum size per cluster. This comes with a recommendation of 100 voxels per cluster, which is another reason why this algorithm, using default values, could not find small lesions.

These dialogues are thrown intermittently throughout the code that is being executed in the post-processing step. First to be executed is smoothing. Lesion probability map is processed according to the supplied by the user FWHM smoothing kernel and then a threshold is applied to retain values in voxels with magnitudes above 0.25. This is intended to close gaps, smooth rough edges and degrade small isolated lesion clusters given by predictor [34]. This is followed by clustering algorithm, which is used to further remove small clusters of lesions. By default minimum cluster size is 100, which means if a total number of voxels in a cluster is less than 100, those voxels are removed from the lesion map.

This concludes the run of Griffis lesion delineation algorithm with default values using his trained predictor system. For the first test case, upon visual inspection, it finds excessive amounts of areas affected by the lesion (Fig. 2.2C).

2.1.2 Finding the voxel most likely to belong to lesion region

This voxel in this research paper is refered to as lesion centre. In general, centre refers to the arithmetic mean centre of something computed by taking a sum of all points coordinates and dividing from total number of points (2.5); it is also possible to employ weights of data points to steer the centre towards higher weighted points. Testing in this research shows that for lesion region centre determination in an MRI scan a simpler solution gives the best results – that is due to the nature of the data used and the end goal.

$$[x, y, z] = \frac{\sum_{n=1}^{m} [x_n, y_n, z_n]}{m} \tag{2.5}$$

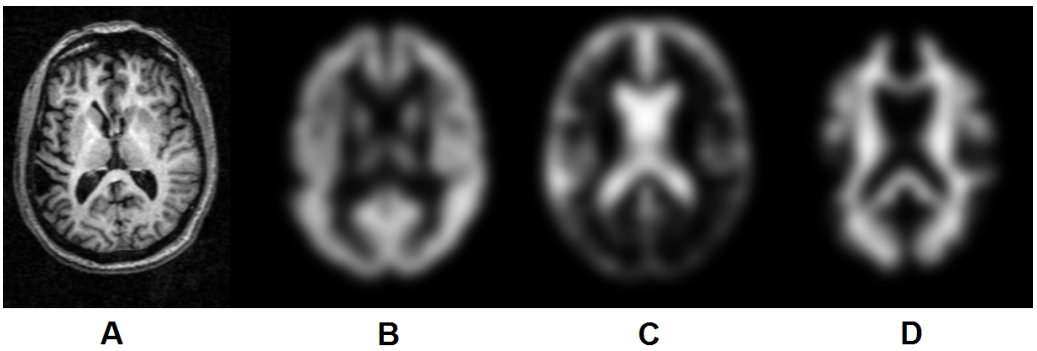


Fig. 2.3. Example MRI scan (A) and its grey matter (B), white matter (C) and CSF (D) TPMs with FWHM smoothing applied.

MRI scans need to be processed to emphasize the region of brain that is likely to be damaged. For this, aforementioned methods are used, methods used by Joseph C. Griffis in his attempt at automatic lesion delineation (Section 2.1.1) and SPM toolbox for MATLAB (Section 1.1.1).

First step is obtaining tissue probability maps. This is achieved using SPM toolbox to segment MRI image into white matter, grey matter and CSF. In this step, parameters obtained from analysing and looking for ways to improve Griffis algorithm (Section 3.1) are used; which means increased FWHM from default 8 to 11, giving even blurrier/smoothed tissue probability maps (Fig. 2.3).

Additional change to Griffis method is that this method now works for brains damaged in either hemisphere without manually specifying beforehand which hemisphere is damaged and even for brains that have damage in both hemispheres, further reducing manual pre-processing of input data. This is achieved by removing code that clears unused hemisphere data from actual, flipped and template tissue probability maps. Only clean-up that remains is clean-up of data outside the brain according to tissue probability map template. Having this data it is now possible to attain missing tissue feature map. Here same functions as in Griffis algorithm (2.1, 2.2) are used, with the only difference being that input data contains both hemispheres at the same time, but that does not interfere with formulas. And this obtained feature map is then used in determining most likely voxel to belong to the lesion region.

Though, another point worth mentioning is application of thresholds to the feature map to determine the likelihood of finding a lesion region. When the magnitudes of values in the feature maps are small it shows that there is little variance between the tissue probability maps, which, in turn, means that algorithm is unlikely to find point in lesion area. As case in point, some statistics for the method for centre determination, which this project went with in the end, without any pre-filtering of MRI scans: without any thresholds – 121 MRI scans are processed with 96 hits giving 79% accuracy; at 0.9 threshold – 3 are processed and all are hits; at 0.1 threshold (which were used in this project in the end) – 108 are processed and this particular method finds points in hand delineated regions 92 times giving 85.19% accuracy.

Lastly, the issue of selecting or pre-filtering data set. Failures to detect a correct point stem from utilizing symmetry and two cases where algorithm fails could be identified. First being significant brain CSF distribution asymmetry. There are multiple cases where CSF area in one

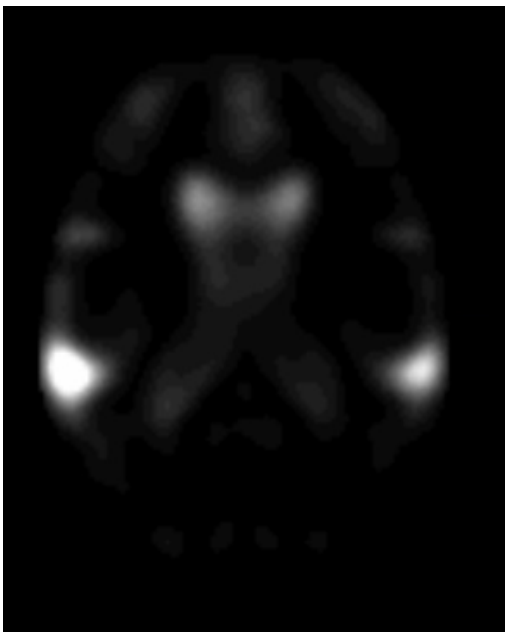


Fig. 2.4. Missing tissue probability map for both hemispheres.

Table 2.1. Results of maximum magnitude method tests on non pre-filtered MRI scans.

Threshold	Hits	Total	Accuracy
0.9	3	3	100
0.8	8	9	89
0.7	14	16	88
0.6	21	23	91
0.5	33	37	89
0.4	45	51	88
0.3	60	67	90
0.2	77	87	89
0.1	92	108	**85**
0	96	121	79

half of the brain is significantly larger than the other, having a clump of CSF on one hemisphere but not the other; but that is not a lesion. Since CSF has same visual features in MRI scan as a lesion, this algorithm falsely identifies this CSF region as a lesion. Second issue is symmetrical lesion. Within a test set there is an MRI scan that has symmetrical lesion on both hemispheres (with only small bits undetectable by the algorithm that are non-symmetrical) which means that when looking for asymmetry there is nothing that stands out and algorithm fails to find the lesion (Fig. 2.5).

2.2 Usage of region growing method for the segmentation of the lesion region

Nest step in proposed algorithm is region growing. Formula for simple region growing itself is not difficult in theory – take point from set, test neighbours, assign to result region if growth

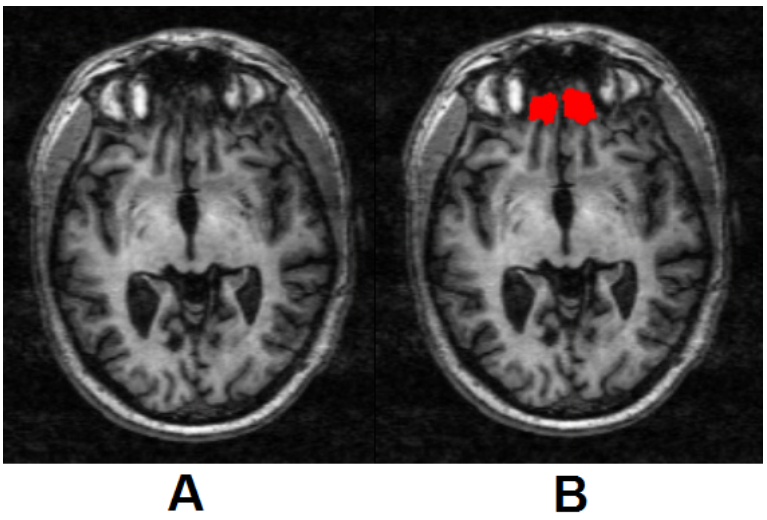


Fig. 2.5. Example of symmetrical lesion (A) with hand delineated regions shown (B), which is invisible to the automated algorithms.

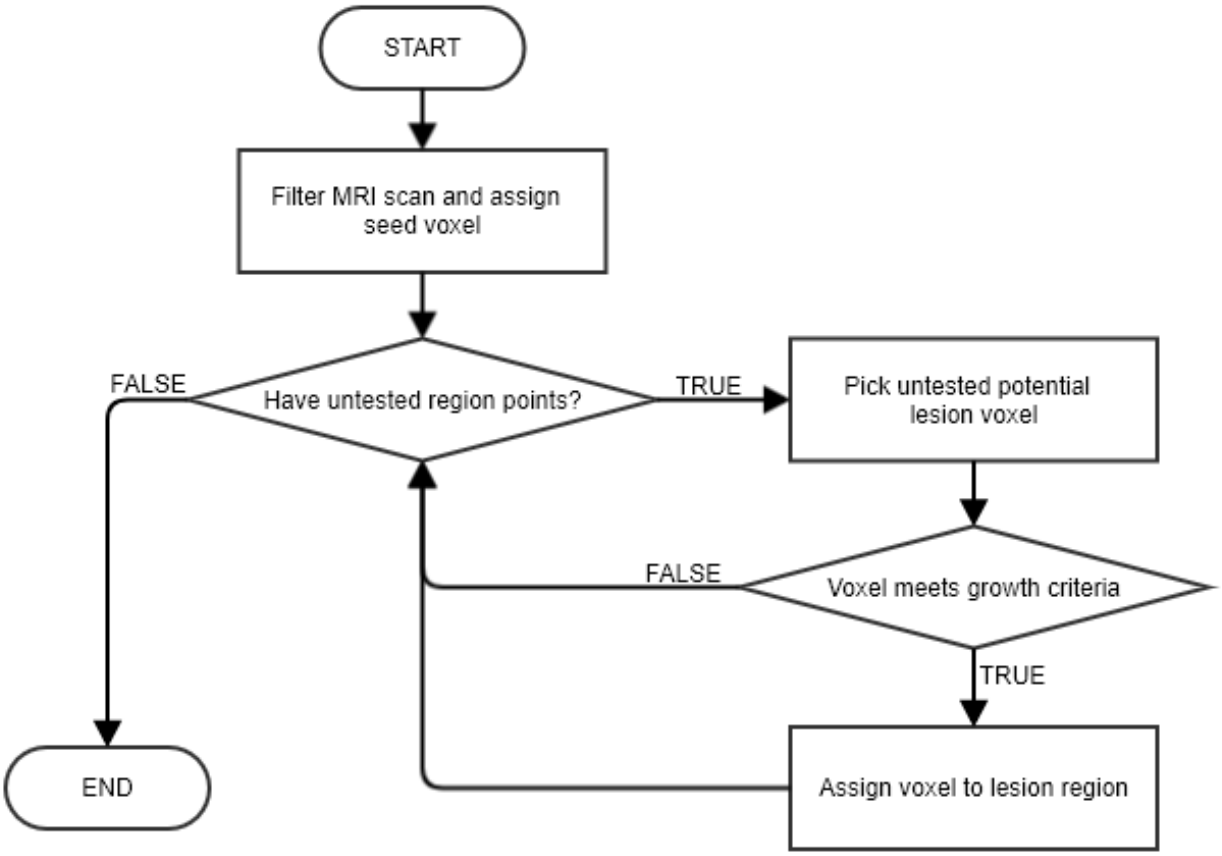


Fig. 2.6. Algorithm schematic for region growing.

criteria are met, repeat. Applying it in code is different, when speed and accuracy is an issue, so some extra means are employed trying to improve the algorithm. There likely is research done about things and they have proper scientific names, but these are thought up specifically for this case from partly heard knowledge, little knowing about them so here they are called as inclusion cube and rejection cube.

In this works attempt as a base a simple region growing algorithm [48] is used. It runs in a cycle until no new points are assigned to the lesion region (Fig. 2.6). It works on an original MRI scan, not segments. Though, before region growing is applied, anisotropic diffusion filtering is also applied. This removes some of the noise, smoothing similar intensity areas, while maintaining the sharp edges [46], as opposed to the Gaussian blur. Initially taking the before calculated lesion point, the 6 three dimensionally adjacent points are tested against growth criteria. Growth criteria is where most of modification lie.

I removed range test and added inclusion and rejection cubes; also removed a redundant check as the author stated himself. So in essence only check if point already belongs to the region and the boundary test is maintained, which is needed to avoid errors in cases where region grows to the edge of the image space. In theory region should never approach edges of MRI scan image, but as far as this work tested, it could be a possibility as lesion region intensity is close enough to the backgrounds', meaning if for some reason a single air voxel would get assigned to the lesion region then said region would grow all the way to the edges.

Range test was removed in favour of a simpler hard cut-off – cycles limit. Range test was implemented to contain the grown region within a specified distance around a seed point. In the current iteration its just allowed to grow till arbitrarily cut-off of 100 000 cycles is reached. This gives at least 100 000 voxels to delineate. If the region did not stop growing by this point then it means it started including CSF or in worse case – started assigning air around the head as lesion region – meaning that algorithm failed. Though in final version it should be removed as 100 000 voxels would not be enough to cover large lesions.

First thing added, second chronologically, is inclusion cube. This substitutes the threshold test by making it test not only the voxel itself, but also the voxels surrounding it. This cube has its radius and threshold predefined. Radius is the number of voxels along either coordinate axis to be included in threshold test; so a radius of 1 would make a 9 voxel cube centred on the tested voxel. Threshold is the maximum intensity difference allowed from the original seed voxel intensity. For the most part what adding inclusion cube achieves is growing the region further by the size of that radius (Fig. 2.7. It is kept since it helps fill in empty spaces that sometimes appear inside the region. Though for the most part it is a remnant of an idea before anisotropic filtering idea and seem to clash with effectiveness of rejection cube.

Passing those tests brings the rejection cube test. This is an attempt to stop region from growing into CSF regions. Idea is simple – test larger surrounding area around test voxel and see how much of it could potentially belong to lesion region; if most of this cube does not belong to lesion area, then prevent assigning this tested voxel to the lesion region. Ideally this check would prevent algorithm from going through narrow CSF regions between grey and white matter regions which is the most prevalent issue with implementing region growing in MRI scans when attempting to detect lesions caused by ischemic insult.

As to why cubes are used instead of spheres or some other geometric shape – simplicity. This avoids extra calculations and is made really simple in MATLAB, which allows quick selection of cube-like matrices within bigger matrices. And speed is key, as this region growing algorithm

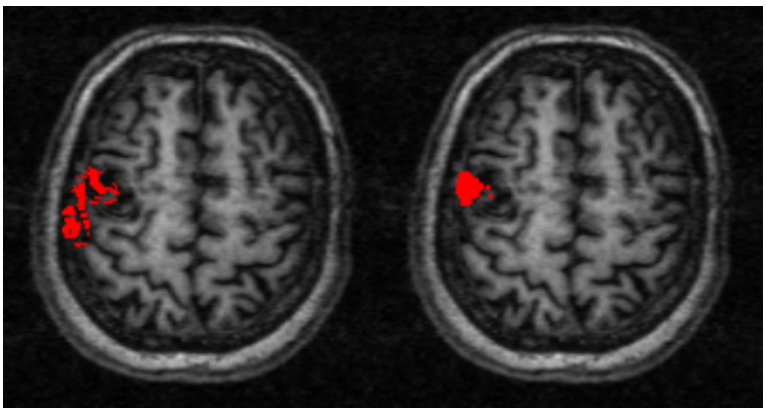


Fig. 2.7. Example of inclusion cube effects: left – radius is zero (just the voxel), right – radius is 2 (cube of 125 voxels).

tests 6 times per newly assigned to the region voxel.

If the tested voxel fulfils growth criteria it is added to the region and queue. Queue contains all the points added in this cycle and queue is used to determine when to exit the growing cycle – when no more points are assigned there is no point in continuing the algorithm.

2.3 Analysis conclusions

Griffis algorithm is fine in multiple situations, but it currently has too many limitations and inclusion of naïve-Bayes classification, which needs training, means that to adapt the algorithm for use in new environment would take time and data. Multiple issues, mostly considering optimization, can be easily circumvented though. But to determine the location of brain stroke, ideas from Griffis algorithm is perfectly functional. With a little tweaking single most likely voxel can be discerned.

To avoid using pre-training of any sort, in this work instead of predictors, region growing is taken. In its simplest form it is not good enough since brain stroke induced lesion area in a T1-weighted MRI scan is too similar to CSF; thus requiring multiple improvements to region growing growth criteria. This work experiments if introduction of inclusion and rejection cubes would allow satisfactory stroke induced lesion region delineation.

3 VALIDATION OF DEVELOPED METHOD

Most of research work went into testing out the existing methods, mostly Griffis algorithm; finding ways to improve them and look for ideas to use in this works own implementation. Large part was fine tuning Griffis method, which gave me parameters to use while implementing SPM toolbox in this works own method. This section, however, does prove that method proposed in this work could be used to at least point out whether there is an issue to look at in the brain and roughly where it is without actually needing to look at it.

3.1 Validation and improvement of Griffis algorithms

Griffis lesion delineation algorithm is neither accurate, nor is it optimized. Accuracy depends on the type of data fed to the algorithm and practice in picking settings like smoothing kernel FWHM. The script is considered not optimized, because it often performs calculations that are irrelevant to subsequent calculations and even saves some of this extraneous data onto the hard drive. This significantly increases processing time and wastes storage space – in above mentioned case all data generated by this algorithm for one patient takes 334 MB (including 8MB of initial MRI scan).

$$DSC = \frac{2|X \cap Y|}{|X| + |Y|} \tag{3.1}$$

The metric used for results evaluation in this work is Dice-Sorensen Coefficient, otherwise known as Dice Similarity Coefficient, or DSC for short [49], which shows the similarity between two sets of data. X and Y in Eq. 3.1 refer to the two sets of data. DSC is obtained by multiplying the number of overlapping points by two and dividing that by the sum of all data points in both sets. In analysis of lesion delineation algorithm results, first data set corresponds to manually delineated lesion area three dimensional matrix which has values of either 0 or 1, with 1 indicating that the lesion is present in that voxel; second data set is algorithm delineated lesion area three dimensional matrix, which is has a threshold applied with, an arbitrarily chosen, 20% cut-off, meaning that values smaller than 20% of maximum value in all set are assigned 0 value and others are assigned 1. Since both sets are binary, with maximum magnitude of 1, resulting DSC values ideally should be equal to 1. That would mean that all automatic delineation data points coincide with all points of hand delineation. For analysis purposes, DSC is calculated for each MRI scan slice going vertically instead of one DSC value for all. This allows to identify slices where algorithm found lesion area accurately, and where it produced just false data.

For testing 124 MRI scans were prepared, however due to limitations of the algorithm majority were rejected. This algorithm due to smoothing involved and due to brains being asymmetrical cannot find small lesions. It can find only lesion areas in MRI scans where a significant portion of the brain is damaged. To pick out which brains have small lesions, their manually delineated lesion regions were used and MRI scans with total lesion area of less than

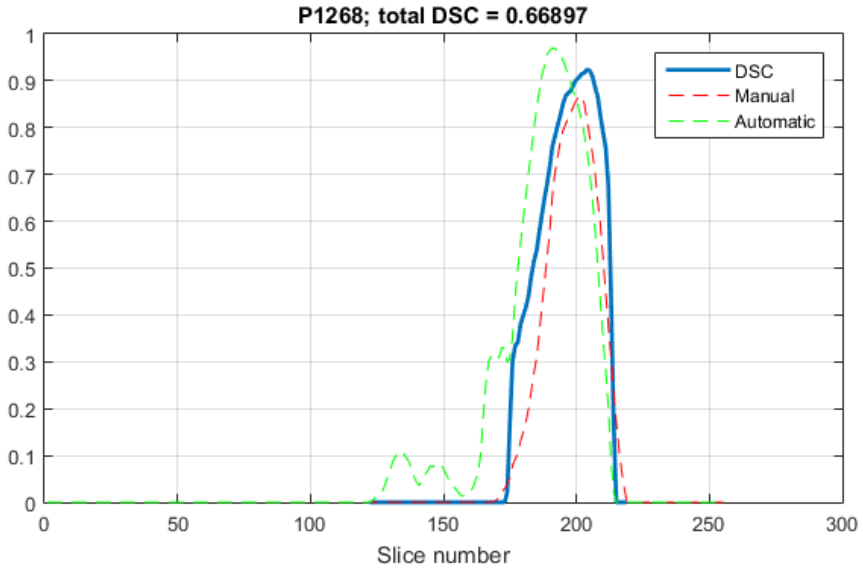


Fig. 3.1. Best attained per slice DSC results after running algorithm for 49 patients.

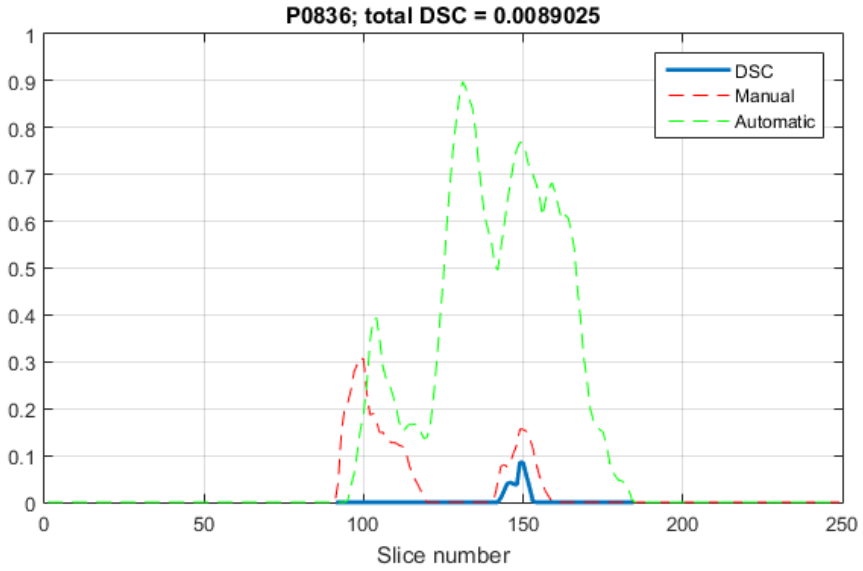


Fig. 3.2. Worst attained per slice DSC results after running algorithm for 49 patients.

5000 voxels were rejected. Another feature for rejection is lesion location. This algorithm cannot delineate the lesion if it spans both hemispheres. That is why only lesions that affect single hemisphere were selected. After this filtering, 49 MRI scans were left and algorithm was run for all of them.

DSC results are plotted as a graph for each slice. Slices where there is no data in either set has no line as the DSC equation (Eq. 3.1) produces a division by zero in all plots there are slices with DSC of 0. This is due to the automatic algorithm finding larger or just different areas than in manually delineated case (Fig. 3.2). These plots also have red dashed line representing relative size of manually delimited lesion area in that slice and green dashed line showing relative size of automatically delimited lesion area in that slice. These lines are used to indicate MRI scans which in the end got seemingly good DSC values in some slices, but that is only due to automatic algorithm delimiting huge areas of brain as affected by lesion.

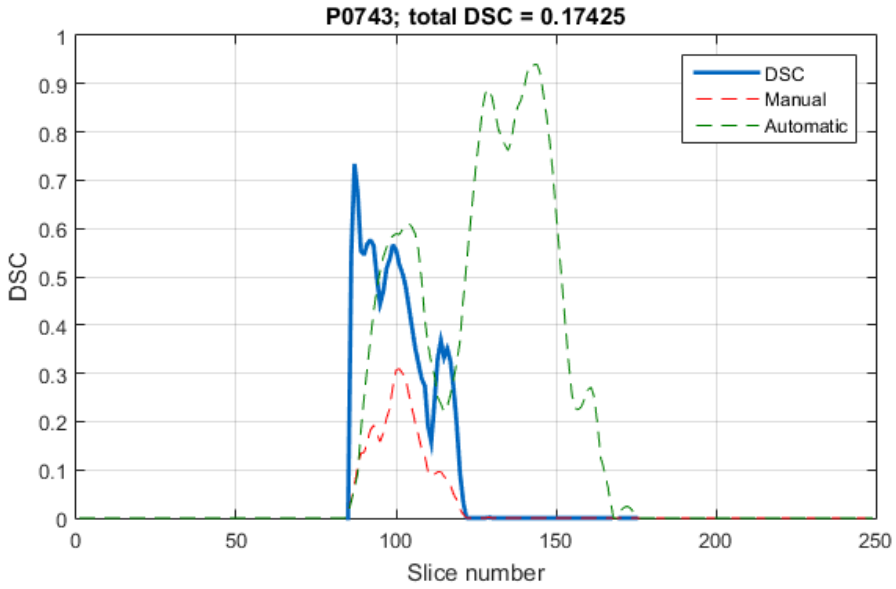


Fig. 3.3. DSC plots showing results of lesion delineation for patient P0743.

None of the MRI scans in any of the slices attained DSC value of 1, which would be the ideal case, however in one case DSC of 0.9 in a slice was attained (Fig. 3.1). 13 more got DSC values around 0.8 for a slice, which shows that this algorithm can delineate some lesions with a degree of precision. However, these 14 patients constitute only 29% of the test cases. DSCs for 3 patients do not reach magnitudes of even 0.4 per slice. That is when algorithm was unable to determine where precisely is the lesion and just marked large swaths of brain as affected by lesion. Patient MRI scans in between these two extremes get some decent DSC values in some slices because of same reason – algorithm delimiting large areas as affected by lesions and marked voxels in sets at particular slice just happen to coincide. In this case total DSC value shows that algorithm failed. As an example, for patient P0743 (Fig. 3.3) one of the slices has DSC over 0.7, but automatic algorithm delimited large areas of the brain as lesion even though there was none, which brings total DSC to 0.17. This same issue is present in few of the results, which looking at the per slice DSC would imply "good" result. In case of patient P1712 (Fig. 3.4), maximum per slice DSC reaches 0.8, but the total DSC is only 0.28 implying a bad result. And really in that case automatic algorithm finds large areas of lesioned brain where there is none (Fig. 3.5). In the end there are cases where automatic algorithm fails completely (Fig. 3.2). However, this case is illustration of algorithm being incapable of detecting small lesions (Fig. 3.6). Analysing MRI scan and manual lesion delineation, one can see that only small areas scattered around the right hemisphere are affected by the lesion. That is the worst case for the algorithm – lesions that are small and on edges of the brain. After smoothing and removal of areas outside of presumed brain area information becomes indistinguishable.

So in the end, according to the algorithm author, lesion delineation with total DSC of 0.60 or higher can be considered "good" [34]. By this estimate out of tested 49 cases only 7 delineations to be "good" – mere 14% of cases. This is likely due to predictor system that was trained by the original author just does not work with the MRI scans this work have tested with. By training your own predictor system it should be possible to increase the accuracy and

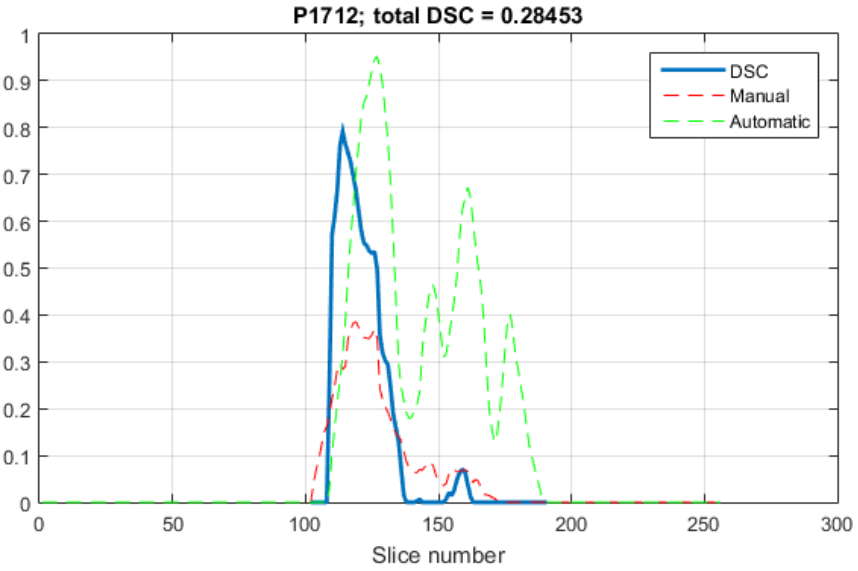


Fig. 3.4. DSC plots showing results of lesion delineation for patient P1712.

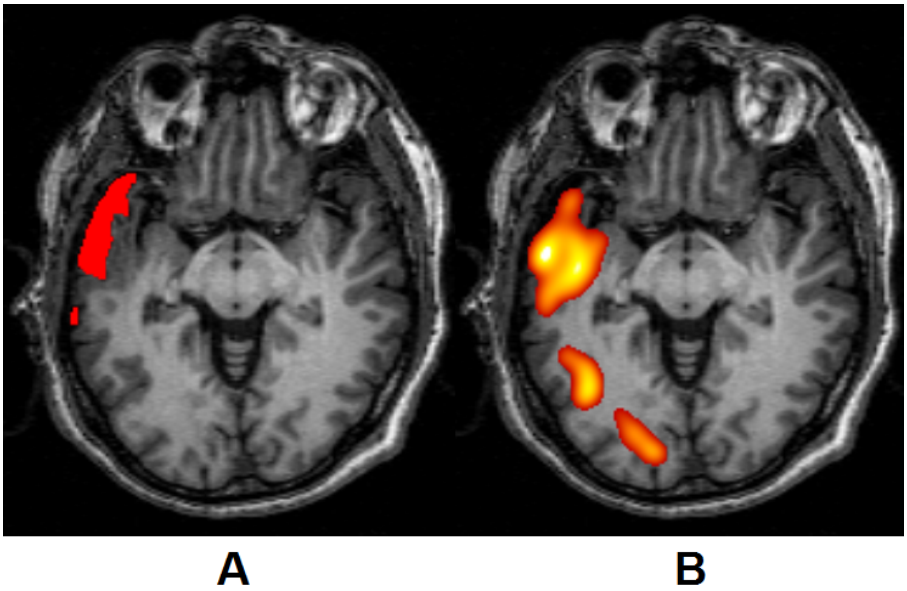


Fig. 3.5. Manual delineation (A) comparison with result of automatic delineation (B) for patient P1712.

improve DSC scores, but that is not likely to work later, when running for other cases. Author had 30 MRI scans to work with, used 29 for training and 1 to test the obtained prediction system. However, later on when testing the system he, presumably, applied it to the same 30 MRI scans. This is why his results are better – 66% of the lesion delineations are "good" [34]. This is to be expected as the system was trained on those MRI scans. And that is why predictor trained there does not work in many of 49 cases tested during this research work, since these vary greatly by placement, size, clustering and intensity.

However the results and overall algorithm operation can be tuned and improved editing the script and default values. So at this point an edited version of authors original `lesion_gnb.m` scrip is created, naming it `et_lesion.m`. Main focus of this script is to increase number of things done automatically; make the script run for multiple patients without user interaction. This

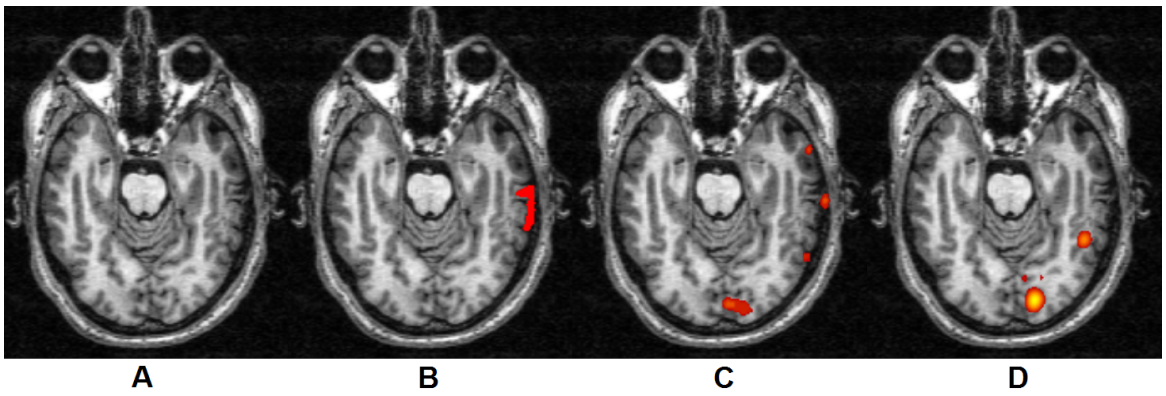


Fig. 3.6. Patient P0836 MRI scan (A) manual delineation (B) comparison with result of automatic delineation (D). Also showing found missing tissue TPM (C).

was especially appealing, as even without any user interaction, using default parameters and scripts, `lesion_gnb.m` for the 49 test cases runs around 3.76 hours (Table 3.1). `lesion_gnb.m` requires a variety of settings on each MRI scan, however only one depends on the MRI scan itself. As such they can be set at the beginning and reused for all cases. The one case where this does not apply is indication which hemisphere of the brain is lesioned. Improved method code determines this by MRI scan file name – last symbol in file name is either "L", signifying that the left hemisphere is lesioned, or "R", signifying that the right hemisphere is lesioned. Patient MRI scans are taken from a single directory, which is expected to contain only patient MRI scans. Results are placed in one directory, set at the beginning, which in the end contains sub folders named after patients MRI scan file names into which all resulting segmentations, feature maps, delineations are placed.

In terms of saving storage space, 8.4 GB of space were saved (Table 3.1), roughly 40%. However, storage space saving was not the real focus of optimization, more of a side product of improving algorithm execution speed. Most time consuming part is MRI scan segmentation in to TPMs using SPM toolbox. Complex segmentation algorithms strive for high accuracy [32], thus all the lengthy repeated calculations that are used. Also big contributor to the long processing times is that these algorithms cannot fully utilize multiple-core/multiple-thread CPUs. Usual load during processing on Intel i5-4460 4 core, 4 thread processor by MATLAB process most of the time is 40%, and 80% in some parts. So segmentation is a time consuming process, which means the amount of processing done during segmentation overall should be minimized.

Originally Griffis lesion delineation algorithm during segmentation saves to storage all native space TPMs – grey matter, white matter, CSF, bone, soft tissue and air/background. In addition it also saves warped modulated and unmodulated TPMs of grey matter, white matter, CSF and bone. Each TPM requires, not equivalent, but significant amount of processing. In the end, lesion delineation script utilizes only warped unmodulated TPMs of grey matter, white matter and CSF, thus first improvement of mine is generating, and saving, only the required warped TPMs – grey matter, white matter, CSF. When saving only these, segmentation function throws a warning in the cleanup process stating that it cannot be performed. Cleanup can be performed when at least one of each TPM is generated. Thus, improved code also saves

Table 3.1. `et_lesion.m` vs `lesion_gnb.m` execution times and results size comparison.

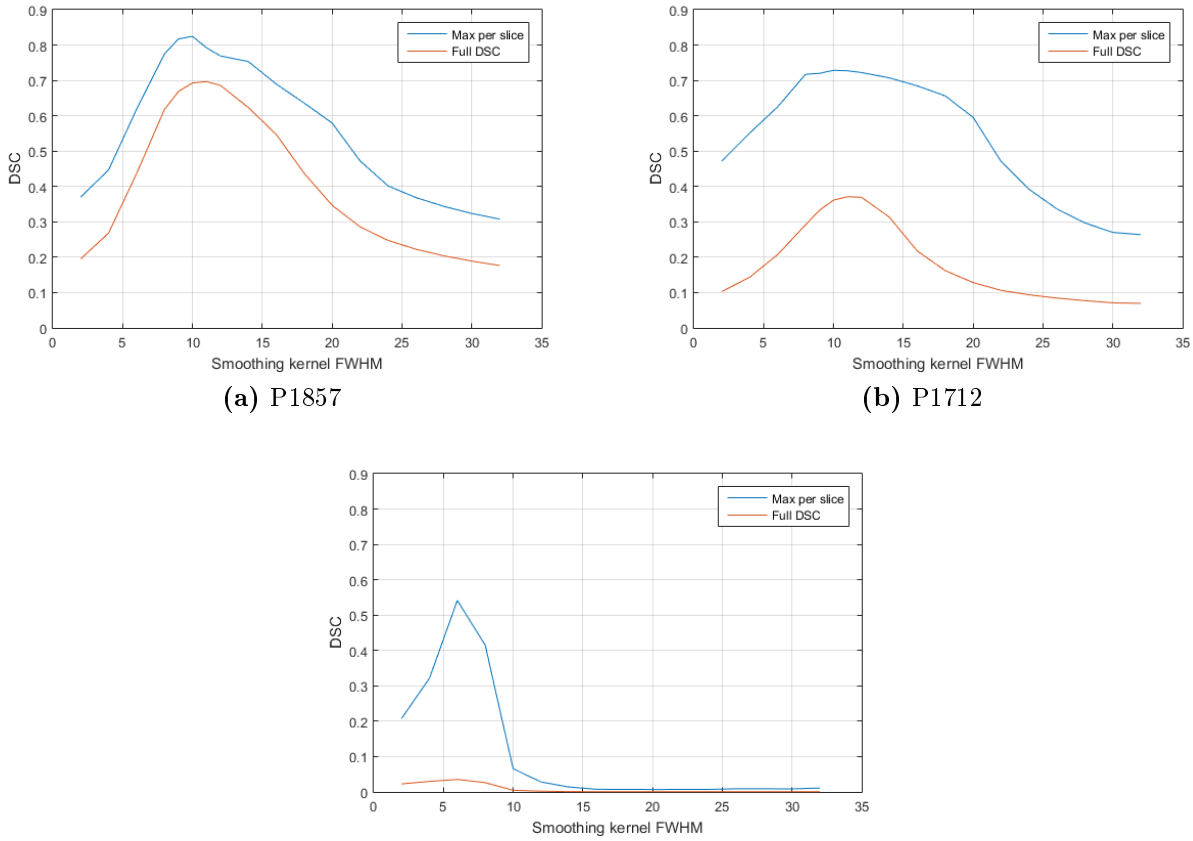
	average segmentation time, s	average other processing time, s	average full processing time, s	full run time, h	results size, GB
`et_lesion.m`	245	4	249	3.58	13.3
`lesion_gnb.m`	264	12	276	3.76	21.7

native space TPMs for bone, soft tissue and air/background. Initial test showed that cleanup is not necessary – tested by simple differences comparison: subtracting TPM generated by Griffis configured segmentation algorithm from TPM generated by segmentation configured by me. Maximum absolute amplitude difference equals 0, meaning there is no difference to the segmentation results from cleanup algorithm. Omission of cleanup and calculation of those extra native space TPMs would save 2 minutes per patient. However, upon manual inspection of few other cases it became apparent, that sometimes TPMs returned from segmentation have false positive values in the air/background, bone, soft tissue regions. To avoid these issues, cleanup process is kept. In the end, segmentation configured by me, running code for all 49 patients, saves on average only 19 seconds (Table 3.1) of processing time per patient.

More time can be saved by implementing a feature that is hinted in the comment in original code, but not actually used – testing if smoothed prior/template already exists, before attempting to smooth it. In feature map calculation equations (Eq. 2.2 and Eq. 2.4)(Section 2.1.1) template TPM, which is referred to as prior in parts of the code, is smoothed with FWHM smoothing kernel defined by the user. As the original non-smoothed prior file used for calculations does not change, smoothing could be performed once per FWHM value and, upon code reruns, previously smoothed TPM could be reused. Check if previously smoothed TPM exists and subsequent omission of smoothing saves 8 seconds per patient in optimized algorithm – execution time changes from 12 seconds down to 4 seconds (Table 3.1). This 4 seconds average is attained, when all runs already had smoothed prior TPM and were skipping smoothing process. Average time of the `lesion_gnb.m` indicates how long processing would take if prior TPM would not be smoothed with a particular FWHM, effectively, showing how long it would take to when running script for the first time.

Further script optimization has comparatively small impact on processing speed or storage space taken and mostly deals in increasing code readability and minimizing redundancy in variables. However, other parameters can be optimized. Focus was on 3 of them – FWHM smoothing kernel, prior coefficient and threshold of post-processed lesion delineation. Both, full DSC value and maximum per slice accuracy, values are used for result comparison purposes. Higher priority is given for full DSC as it shows full lesion delineation accuracy rather than maximum partial, per slice, accuracy.

First, FWHM smoothing kernel. This affects the three dimensional Gaussian blur of each voxel. Test were run for four patients – P1857, which had "good" lesion delineation using default values; P1712, which had "good" maximum per slice DSC, but full DSC of just 0.28; P0836, whose lesion delineation had full DSC of 0.0089 – worst of all test cases. FWHM values



(a) P1857 **(b)** P1712

(c) P0836

Fig. 3.7. FWHM impact on DSC testing results for three patients; (a) – "good" lesion delineation, (b) – erroneous lesion delineation, (c) – completely failed lesion delineation.

were taken in range from 2 to 32 with a step of 2, except for P1857 and P1712 for whom FWHM of 9 and 11 are included to better pinpoint curve peak. This gives a sizeable range which also includes the default FWHM value of 8. Resulting DSC dependence on FWHM plots show the possibility to find an optimal FWHM value (Fig. 3.7).

Analysing first case, P1857 MRI scan, already shows that FWHM of 8 does not give the best possible results (Fig. 3.7A). Total DSC peak is at FWHM smoothing kernel value of 11. Maximum per slice DSC is at its peak at FWHM of 10 and quickly dips below 0.8 at FWHM 11. Thus, this gives two values to pick from. Full lesion area DSC more indicative of overall lesion delineation accuracy, thus FWHM of 11 can be stated as being best in the current testing setup. Using FWHM of 11 instead of 8 increases full DSC by 11% – from 0.6181 to 0.6966. Theory of FWHM of 11 being better than 8 is consistent with curves obtained for P1712 MRI scans (Fig. 3.7B). Maximum per slice DSC curve peaks at FWHM of 10 and full DSC curve peaks at FWHM 11. Few extra checks were carried out best DSC values vary between FWHM 10 and 11, more often than not, 11 being better one. Thus `et_lesion.m` will be using smoothing kernel FWHM of 11.

Further tests show that no FWHM value can help find "good" results in cases where the algorithm fails to find decent results using default parameters. In case of P0836 MRI scan, at FWHM smoothing kernel of 6 maximum per slice DSC does reach values greater than 0.5,

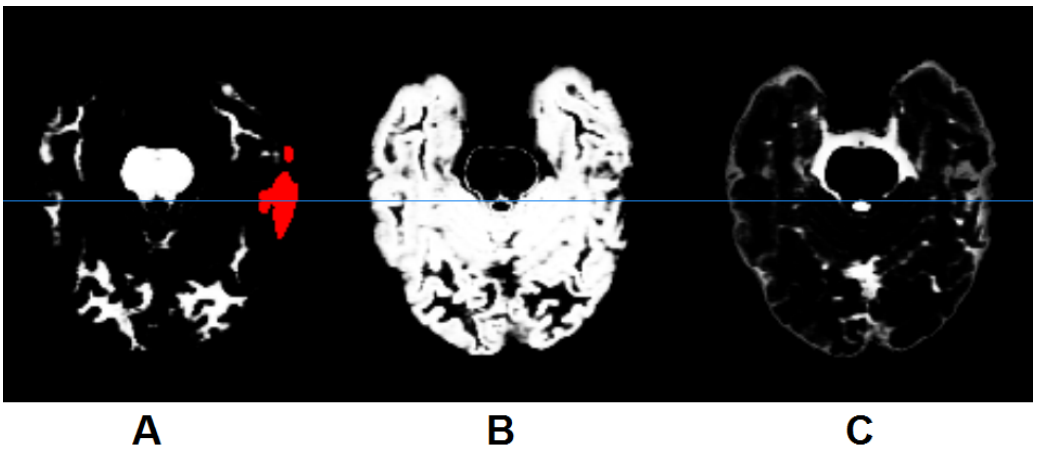


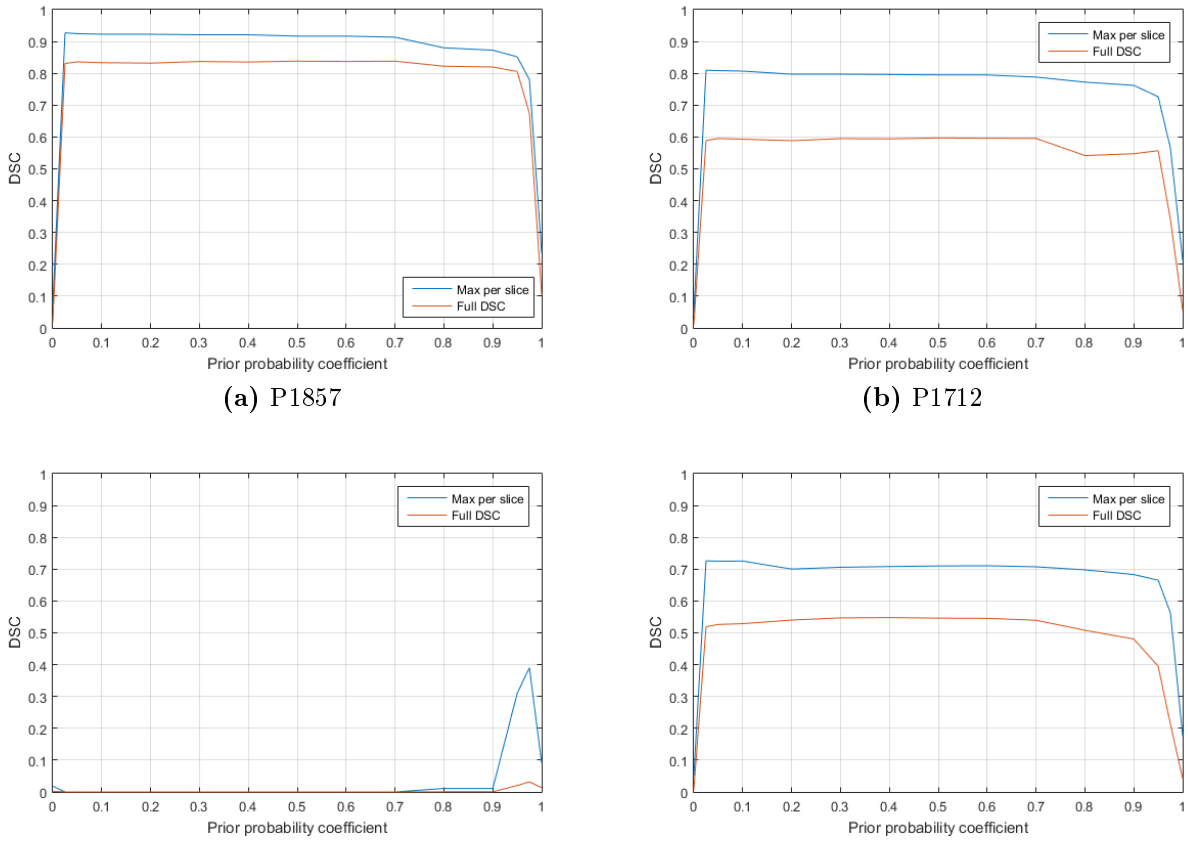
Fig. 3.8. Most obvious slice of segmentations for patient P0836 MRI scan with emphasis on lesion area: grey matter TPM with manually delimited lesion area overlaid (A), white matter (B), CSF (C)

however overall DSC is still less than 0.05. As mentioned before, delineation fails here, because this patients MRI scan has mostly small lesions and they are at the edges of the brain. For untrained eye they were mostly invisible. Even looking at segmentations, though somewhat visible knowing where the lesions should be, they are hard to distinguish (Fig. 3.8), thus it is no surprise that blurring of the image even further obfuscates the useful information.

Next tested parameter set by user that could improve the result is prior probability coefficient. This value is used to make a two element vector, which is then used as one of the parameters in prediction system. First value in the vector is 1 minus the given coefficient, second value is coefficient itself. Analysing the source code for the script it was found that default value for prior probability coefficient of 0.1064 is used by Griffis. However, he suggests 0.5, which is the value used in the control run testing how his script performs for given 49 patient MRI scans. Testing of prior coefficient is performed on same three patients MRI scans from FWHM smoothing kernel impact tests, plus P1419 who had somewhat different curves. FWHM value for these tests is 11 in hopes to find a way how to further improve the result, given best found FWHM. Range of prior probability coefficients is from 0 to 1 with step size of 0.1, including extra points at both ends (0.025, 0.05, 0.95, 0.975) to further determine the point of DSC value drop off. Plots themselves show maximum per slice DSC and full lesion delineation DSC dependence on prior probability coefficient. Resulting curves are in Figure 3.9.

Firstly, prior probability coefficient is not good at 0 or 1. For example for patient P1857 using 0 as prior probability coefficient, gives empty lesion delineation – no results; using 1 as prior probability coefficient, gives all hemisphere as lesion delineation area – too many results. Thus, values in between must be taken.

Curves themselves are quite stable, having small results variation in a wide range, from 0.025 to 0.7 prior probability coefficient values. For patient P1857 with "good" results shown here, full lesion DSC varies from 0.8311 to 0.8383 – maximum difference of 0.0072, which does not give a significant difference. In case of patient P1712 maximum difference is 0.0087 – still not a



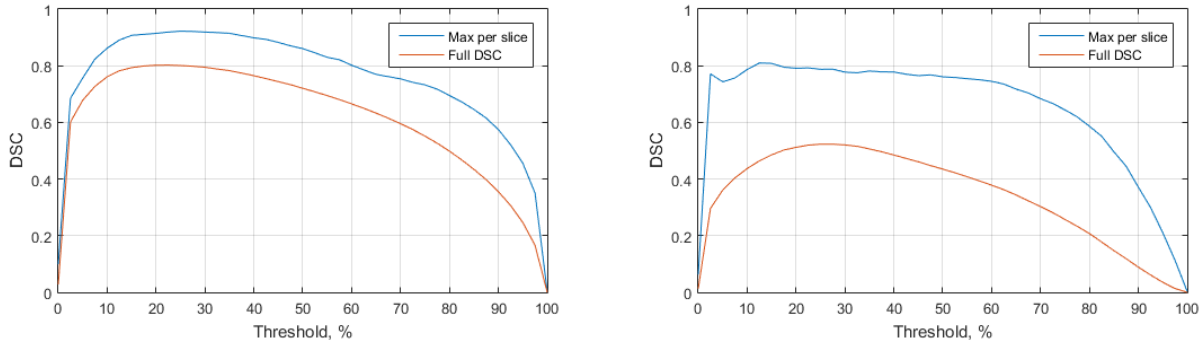
(a) P1857 **(b)** P1712

(c) P0836 **(d)** P1419

Fig. 3.9. Prior probability coefficient impact on DSC testing results for three patients.

significant difference over a wide range. This insignificant difference persists with many other patients who got "good" or at least decent lesion delineations running with default parameters. However, some patients have more pronounced "better" regions – where either maximum per slice DSC or full lesion delineation DSC values are higher than in other regions (Fig. 3.9D). Full DSC and per slice DSC regions do not overlap in this case, however, based on same logic as before during FWHM testing, full lesion delineation DSC values are more important as such values should be picked from that range. As such in `et_lesion.m` prior probability coefficient value is maintained 0.5 – the same as in default case, since it does fall in the "good" range.

Finally, prior probability coefficient does not help with the case of patient P0836, where lesion delineation is complete failure. In the presumed "good" range, 0.025 to 0.7, all values, both per slice and full DSC, are flat zeroes – no result. Some data did happen to overlap at prior probability coefficient value of 0.975, but even then the DSC values are bad and as such this data is not considered to be influential on the decisions made in picking prior probability coefficient to be used in `et_lesion.m`. And as just a test if specially tailored FWHM and prior probability coefficient values could produce a decent result another quick test was run. Maximum per slice DSC of 0.7 is attained at 0.05 prior probability coefficient value, which coincides with other results, implying better maximum per slice DSC in lower ranges of prior probability coefficient. But, in the end maximum full lesion delineation DSC was still only 0.0398.



(a) P1857 **(b)** P1712

Fig. 3.10. Thresholding point impact on DSC testing results for two patients from before.

Third, and final, parameter whose impact on the resulting lesion delineation was tested is the threshold used to convert results to binary values from floating point ones. Original `lesion_gnb.m` script makes binary valued lesion delineation, but after post-processing this delineation once again contains gradients based on given smoothing kernel FWHM. Before DSC calculations this smoothed lesion delineation is once again converted back in to binary values. In the algorithm, threshold value is dynamic and varies per patient. Up to this test the value was 20% of maximum value in all lesion delineation. For example, if maximum value is 1 then all values bellow 0.2 assigned 0 and all value above and including 0.2 would be assigned 1. In practice, however, maximum value is never 1 and in some cases (patient P0836 – failed delineation) as low as 0.18 magnitude.

Test was performed by changing threshold percentage in full range, from 0% to 100%, with a step of 2.5%. All other test parameters are kept the same as in default algorithm, except for FWHM which is set to be 11. Test objects are post-processed lesion delineation of all 49 patients. For result accuracy evaluation once again maximum per slice DSC and full lesion delineation DSC was used.

Firstly, the results for the same three patients as used in previous tests as they represent three distinct results of lesion delineation are checked (Fig. 3.10). Patient P0836 is not included in the figure, because, as seen in previous test, using FWHM of 11 and prior probability coefficient of 0.5 resulting DSCs are 0. Thus threshold application does not change this result as there is no overlapping data to begin with. And only point where DSC rises above zero is at 0%. This is, because with a threshold of 0% all data set becomes ones, due to all values at or above threshold value being set to 1, and this means that inevitably some data does overlap. This feature of the algorithm is noticeable for all patients – everyone have same but small DSC values at threshold of 0%. Inversely, due to the algorithm both DSC values are near zero with threshold at 100%, because, in theory, only one voxel, with the highest magnitude, would be in the comparison set.

Other two patients do have peaks in their curves implying that there is an optimal value for thresholding. For patient P1857 this peak is at 22.5%. Comparing this to the preciously used threshold of 20% this gives an increase in overall DSC of $2 * 10^{-4}$, which is not a significant

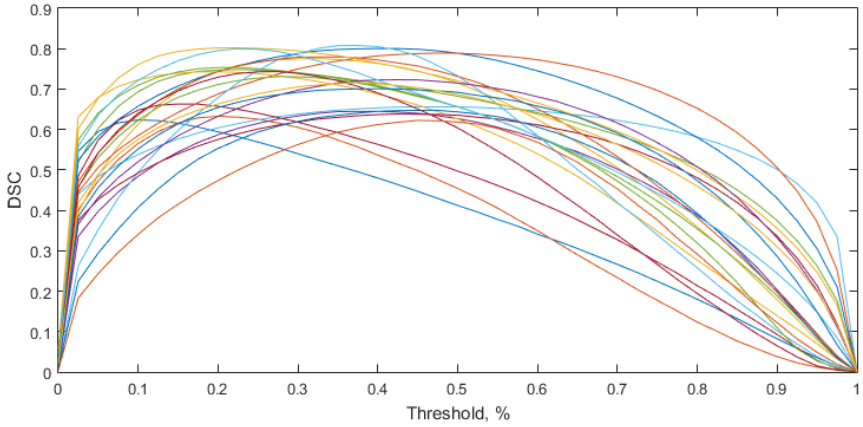


Fig. 3.11. Curves of 24 patients full lesion delineation DSC dependence on thresholding point.

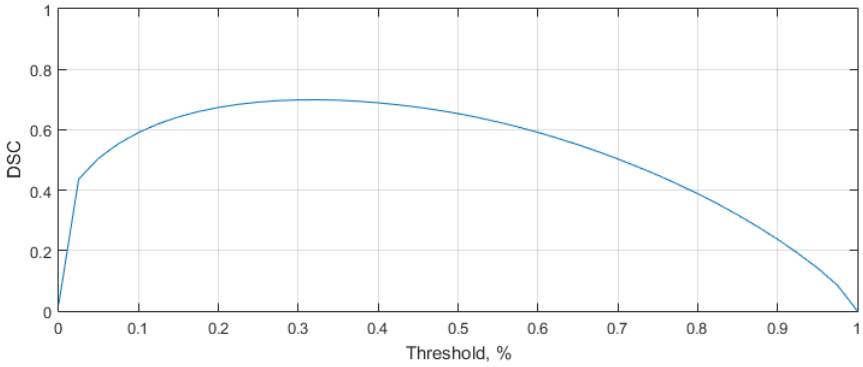
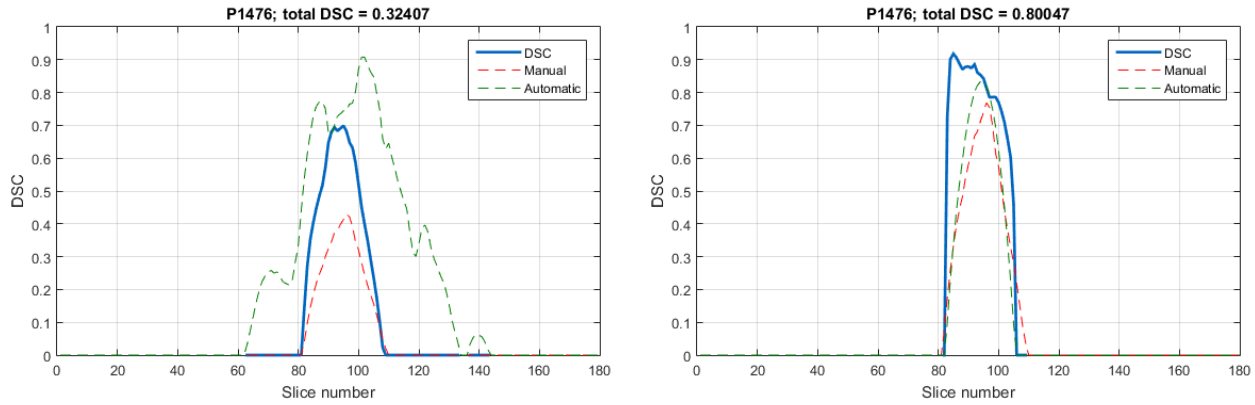


Fig. 3.12. Curve of averages of curves in figure 3.11.

difference. Slightly higher difference is in the case of patient P1712: optimal threshold is at 27.5% and magnitude difference is 0.01 (better result by 2%). These two optimal thresholds do not coincide, as such, different approach is necessary to get best case threshold.

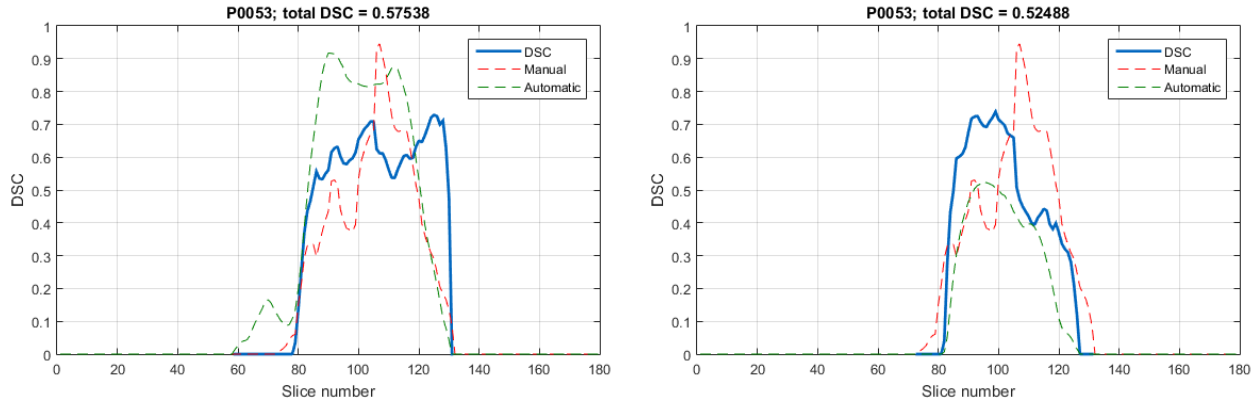
DSC dependence on threshold calculations are performed for all patients, however not all results should be considered further. Results that do not, at any point, reach "good" values [34] are omitted. This leaves 24 patients – half of all patients. However, the resulting curves still have highly varied peaks (Fig. 3.11). To gleam a possible result, results are averaged for these 24 patients for each threshold and looked for a peak in the resulting curve (Fig. 3.12). Maximum magnitude is reached at 32.5% threshold, which, comparing to magnitude attained using 20% threshold in previous tests, gives an increase in DSC of 0.025 (3.6%).

Running modified `et_lesion.m` script and changing two parameters in calculations did improve end results. For results comparison all patients full lesion delineation DSCs were used. DSC improved by 6.28%, meaning an average increase of 0.128 DSC per patient. Largest increase in lesion delineation DSC was for patient P1476, with the DSC increase of 0.476 and also maximum per slice DSC increase of more than 0.2 (Fig. 3.13). This significant increase appears, because the algorithm is able to effectively find the actual area of lesion during feature extraction and after prediction, lesion delineation is similar to the manual delineation. For 44 patients DSC did increase, by some amount. For five patients lesion delineation DSC fell in comparison to DSCs obtained with default parameters. Largest DSC loss was for patient P0053



(a) (b)

Fig. 3.13. DSC graphs for patient P1476 obtained running `lesion_gnb.m` (a) and `et_lesion.m` (b).



(a) (b)

Fig. 3.14. DSC graphs for patient P0053 obtained running `lesion_gnb.m` (a) and `et_lesion.m` (b).

with magnitude of 0.05 (Fig. 3.14). This loss is, due to script now giving significantly smaller lesion volumes, but in this case that did not mean more precise delineation than with default parameters. For patient P0056 lesion delineation total labelled lesion voxels by `lesion_gnb.m` is 65 141, number of voxels labeled by `et_lesion.m` is 27 344 voxels, as also seen in graphs comparison. These smaller lesion volumes are also what give better results in case of other patients lesion delineations.

Total number of voxels marked as lesion area in `et_lesion.m` decreased by at least 25.9% for every patient; on average, decreased by 61.6%, with a maximum decrease of 93.2%. This decrease is mostly caused by threshold increase, but is also affected by prediction algorithm working differently when given slightly different feature maps that are influenced by new FWHM smoothing kernel. This decrease in lesion delineation volume gives higher precision, better DSC, in cases where extracted missing and abnormal tissue probability maps manage to find actual lesion regions. Otherwise prediction system further muddles the information marking large swaths as lesion area and in that case removal of voxels decreases DSC.

So in the end, `et_lesion.m` manages to get "good" [34] lesion delineations of 21 patients,

compared to 7 obtained using script with default parameters. It is still only 43% of patients in the specially selected test group giving good results. As such it is still not usable for general application on any MRI scan and it would be nice to implement some self-evaluation algorithms that would provide the user with an indication of what accuracy in the result should he expect.

3.2 Swiss lesion delineation method analysis

This second method for lesion delineation if from Swiss scientists Raphael Burgener and Stanislaw Adaszewski. For convenience, in this paper it will just be called as "Swiss method". This method uses a lot of complex formulas, works with outdated SPM toolbox and strives to speed up operation by incorporating pieces of code running in C language for some calculation intensive operations. However as of this point this method is still very slow – 12-18 minutes per patient; depending on requested outputs – if user asks for a nifti file with, for example, grey matter segmented out, script runs time consuming SPM segmentation algorithm for a second time.

This method does utilize unified segmentation methods discussed before (Section 1.1.1) for segmenting brain into white matter, grey matter, CSF, bones, soft tissues and rest. However, unlike Griffis method, instead of evaluating sum difference and subsequently growing using predictors, Swiss method uses multiple high complexity and time consuming steps.

First step that is optional is preprocessing of the nifti file that includes registration, bias-correction and skull stripping. This applies unified segmentation algorithms to section the file; specifically from SPM8. And even here the code is outdated and not optimized in the sense that first the basic segmentation is run and then followed by another run that applies deformation (warping) to the received files. In latest SPM libraries this can be done in a single run. This then is followed by masking – area outside of brain itself is removed. And in the end files are shifted around and unnecessary ones are deleted. This step, on average, takes 4.6 minutes, segmentation being still the most time consuming step. Also worth noting that while this step is marked as optional, further calculations require warped grey matter, white matter and CSF segmentations, so it is actually mandatory at least for the first run

This entire step of the Swiss method is highly inefficient in the sense that it actually segments each part out of the brain even if some parts are unnecessary in the end. Swiss method uses those extra segmentations to create masked brain image, but Griffis method circumvents the need of extra segmentations by loading predefined mask. Predefined mask works since the patients brain is warped to a standard template which means shape of brain as a whole in the end is the same for all patients, thus some preset mask works for them all. And all in all, the masking algorithm in Swiss method also introduces what could be considered as noise to the edges of the brain compared to simple masking in Griffis method – high intensity voxels around the edges near blank space, near skull. Though their effect is mitigated in later calculations due to application of Gaussian filter that blurs it out.

Once the preprocessing is done the mandatory part of the code starts and it mimics the brain segmentation based on tissue classification models [15–21], since it calculates intensity,

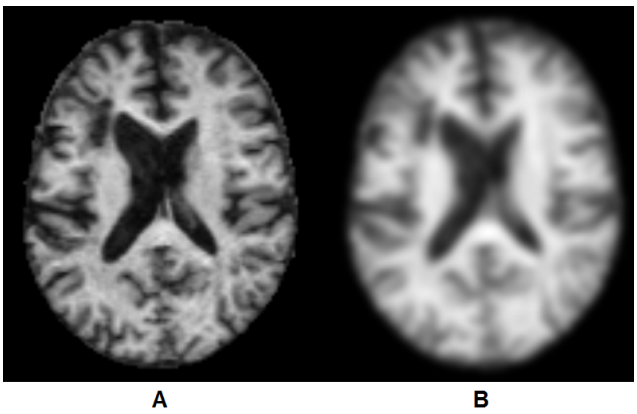


Fig. 3.15. Input (A) and output (B) of intensity feature extraction.

thickness of white matter, grey matter and CSF, and geometry.

It starts with intensity calculations; specifically, intensity normalization. There first is, so called, landmark extraction. Landmarks that are extracted include minimum and maximum intensities at any point, median intensity, percentiles at 25% steps and 10% steps, and mean value of peaks. These landmarks are then used in checking if their distribution adheres to two unspecified theorems and if so then warped brain image is modified using polynomials according to those landmarks. Then, before returning the result to parent function, resulting image is blurred with a Gaussian filter (Fig. 3.15). Intensity calculation takes 1 to 2 seconds, so compared to other parts this one is fast and even though it is called later two more times it does not extend runtime that much by comparison.

Next the geometric feature extraction or "folding" of the nifti file is performed. Here once again intensity normalization is performed as before. Difference here is that afterwards two steerable filters are applied, first and second order, and then summed. Steerable filter is designed to detect edges and ridges using first and second order derivatives respectively. Finishing actual "folding" subfunction, again Gaussian blurring filter is applied to the result (Fig. 3.16 A). However folding is performed a second time with the difference being that the steerable filter Gaussian sigma value is 2 instead of 1 used previously (Fig. 3.16 B). Which on visual inspection does not change anything... And running both these "folds" takes 52 seconds – both noticeable in the grand scheme of things time wise.

Then thickness feature map is extracted. This part utilizes Gaussian mixture model with expectation maximization. This function segments brain images into grey matter, white matter and CSF only based on the intensity of the voxels (Fig. 3.17), not accounting for the special locations like SPM toolbox does. This method basically again performs segmentation, just a reduced one, and so it takes a lot of calculations. And it is inaccurate – even author notes in comments that it is inaccurate since k-means algorithm often does not converge as it should, which then distorts the rest of the results. This thickness feature extraction takes 2.2 minutes, which is faster that full SPM segmentation, but still time/computation costly.

That ends feature extraction and leaves outlier detection. This part of the script compares each slice of the brain, one a a time, with the so called "Atlas" for that slice, which is effec-

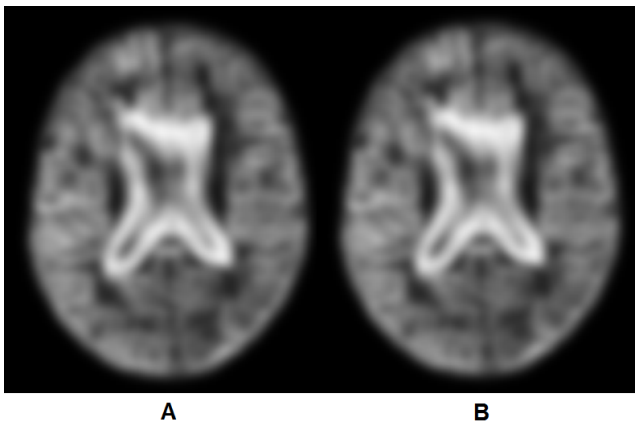


Fig. 3.16. Folding results on same source nifti file as for intensity with steerable filter Gaussian value of 1 (A) and 2 (B).

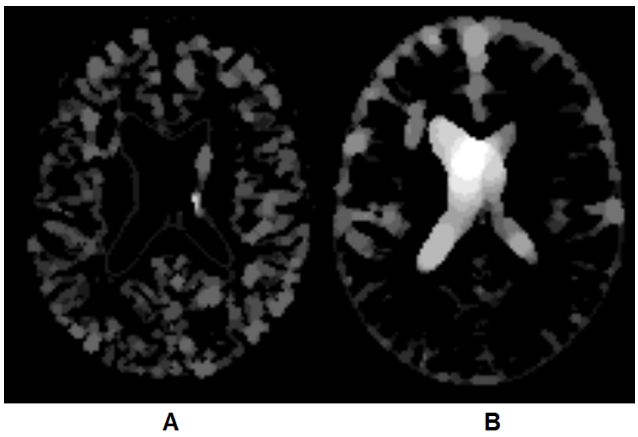


Fig. 3.17. Grey matter (A) and CSF (B) tissue probability maps calculated in thickness feature extraction part.

tively a template of sorts. Previously extracted features (intensity, grey matter thickness, CSF thickness, two geometric feature extracts with different steerable filters) are used to represent the brain in the equation and the "Atlas" slice contains a set of class variables that are not explained what they are. Function generates outlier map (Fig. 3.18 A) and it shows differences from "normal" brain contained in "Atlas", but definitely not lesion area. Results seem similar to CSF and grey mater sum – which was part of the input, so it is somewhat expected. And as far as timing, in takes 85 seconds to calculate for all slices.

So, once the features and outliers are done with, next optional step is region reweighing. This is one of the parts of the Swiss method that has injected C code in so most of the magic is hidden by compiled C code. Goal of this function is similar to Griffis clustering function – reduce amount of small data clusters with the intention of reducing the error. Though the results still seemingly show just a bunch of noise (Fig. 3.18 B) and not the lesion area. At least it is a fast step that takes around half a second.

Then step four is region growing. It goes through each significant voxel in source nifti file applying more complex formulas utilizing previously generated and optionally reweighed outlier maps. Result is still just a grown volume of noise attained during outlier detection (Fig. 3.18

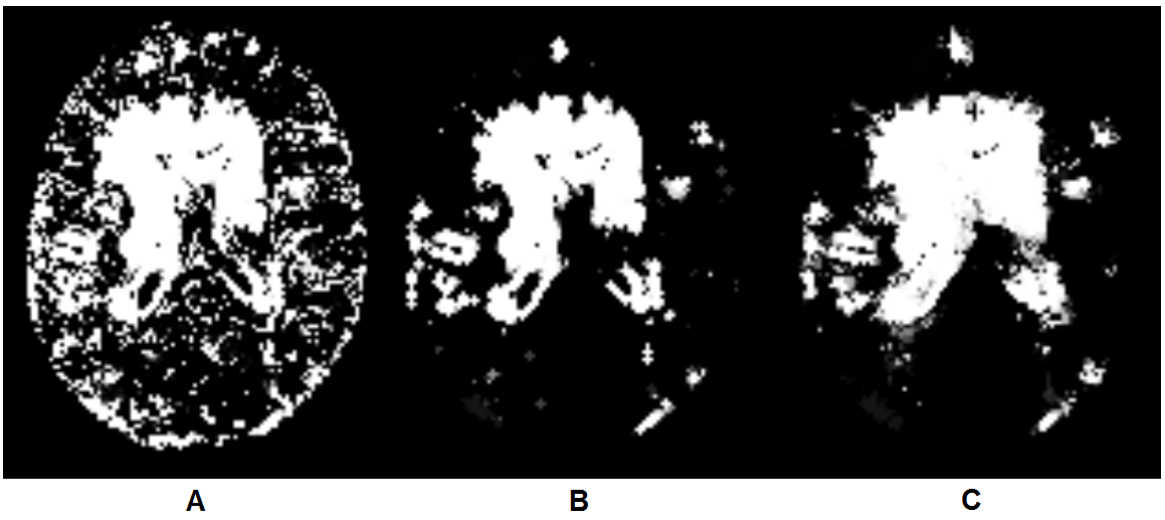


Fig. 3.18. Outliers generated by the Swiss method: initial (A), after region reweighing (B), after region growing (C).

C). Seeing as the purpose of this function is to grow the area of the outlier map, means at least this step is working somewhat. But it does take some time: 38 seconds.

Final fifth step is segmentation of the nifti file with SPM toolbox. Again. This provides nifti files for the script user if he wants them; these are not used by the script afterwards and these are the same as the ones obtained in first optional skull stripping step. So unknowing user extends script runtime by 6.35 minutes just to get some TPMs. Note: user should be able to define which ones he wants, but the script gives them all in any case.

It seems that Swiss method is written by math theoreticians as it uses a lot and complex equations with derivatives. And then it was all put together without any real thought about optimization, end user, or correct execution. To date (2017-06-18) a good result from it was not attained and 12 minutes minimum runtime per patient does not encourage experimentation and bug fixing, especially since the most complex looking function using some preset/template values is giving the noise instead of result.

3.3 Determining the centre of lesion

Initially in this research more scientific and potentially robust approaches using formulas for finding a centre point of a set of points was tried, but in the end resorted to simpler approach which in the end proved to be more reliable and accurate. Here are the results of the methods tried out and implemented.

3.3.1 Arithmetic mean centre and arithmetic mean weighted centre

Missing tissue feature map attained in this works method by same formulas as in Griffis method, provides a feature map that has normalized values across all brain. In this works method, search for the single most likely to belong to lesion region point. Initially, looking for robust mathematically based method, arithmetic mean centre was attempted to use, quickly

Table 3.2. Centre determination methods comparison for a set of 49 MRI scans.

Method	Threshold	Hits	Accuracy, %
Maximum magnitude	0	43	**88**
Mean centre	0.5	19	39
Mean weighted centre	0.5	19	39
Mean centre	0.3	33	67
Mean weighted centre	0.3	34	69
Mean centre	0.1	38	78
Mean weighted centre	0.1	38	78
Mean weighted centre	0	12	24
Mean weighted centre	75%	41	84
Mean weighted centre	50%	39	80
Mean weighted centre	25%	36	74

introducing thresholds and mean weighted centre. Results of testing are visible in Table 3.2. About this test: if method provides no results (no values above threshold) that is considered as a miss and MRI scans used in this test are best case scenarios – with sizeable, homogeneous lesion regions in them.

Issue with using arithmetic mean centre is that there is a large portion of missing tissue feature map that has values above zero, however small the value may be. Throughout the test set on average feature maps have 390 000 voxels that have zero values and 150 000 voxels that are above zero. Now then the issue would be determining the correct threshold. Similar to how it was mentioned before that applying thresholds can improve chances of accurately determining MRI scan that has a lesion, applying thresholds to arithmetic mean centre determination also improves accuracy, in cases where feature map actually has values above the threshold.

Another thing that has a positive influence on the accuracy of arithmetic mean centre determination algorithm is using weights. Feature map contains normalized range of values that signifies how different the brain is at that point compared to the other hemisphere of the brain and tissue probability map template. However without scaling up said normalized values, there was no significant improvement in accuracy. While here this theory was not tested, it is believed that making higher values have significantly higher weights would help the accuracy more.

Thus final attempt at implementing arithmetic mean centre included utilizing weights and automatic maximum magnitude based thresholds. The automatic threshold refers to the idea that the threshold is picked based on highest magnitude in the feature map, e.g.: if the magnitude is 0.8 and 50% automatic threshold is asked for, then threshold of 0.4 would be used. Combining these techniques improved accuracy significantly, however the accuracy was still not satisfactory.

Only at this point of methods analysis obvious issue was noticed – lesion does not need to be a single homogeneous region with its arithmetic centre also belonging to said region. E.g. if the lesion is shaped like a letter "U" – arithmetic centre would be in the whitespace inside the letter. Or if there are two similarly sized regions apart from each other – arithmetic centre

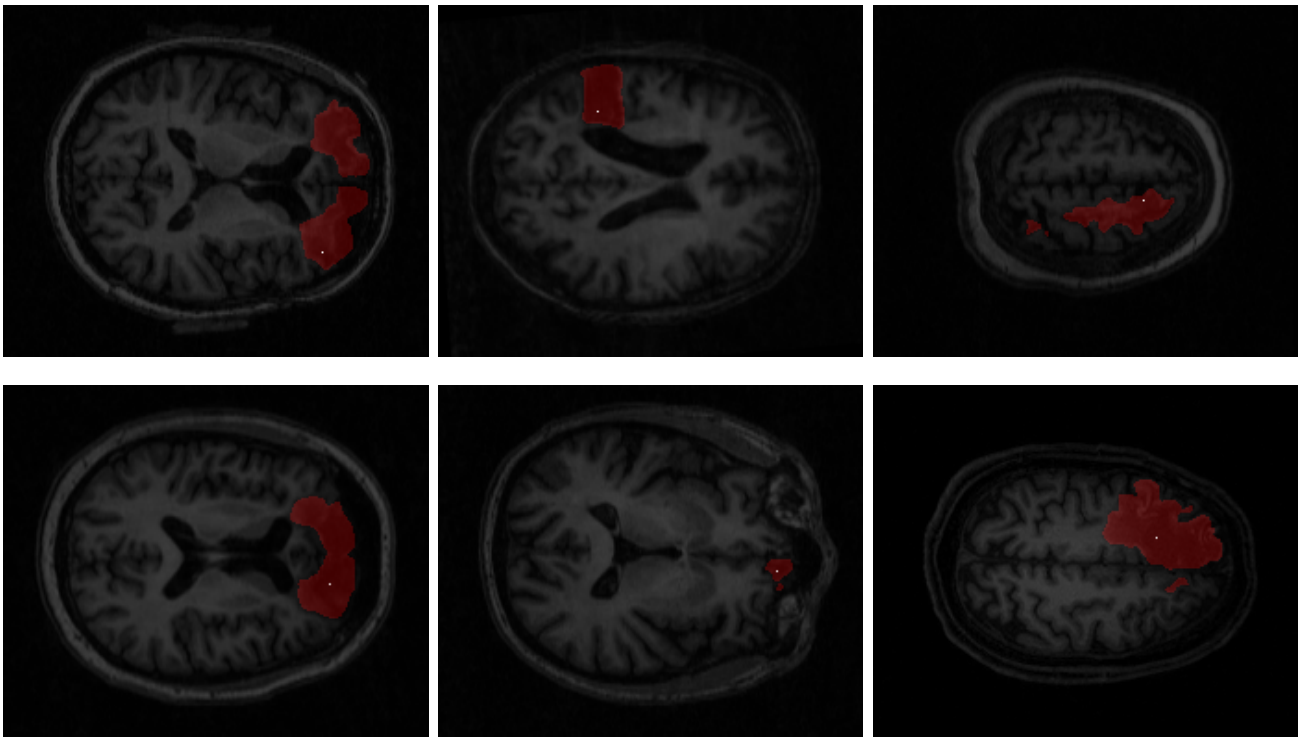


Fig. 3.19. Maximum magnitude method successful point determination examples. Red area is hand delineated region; white pixel in it is the found voxel.

would be in between them. Thus the idea of using arithmetic mean centre as the main idea was abandoned and only kept as a failsafe when maximum magnitude method returns more than one point.

3.3.2 Highest likelihood point

Goal of this project is automatic lesion region delineation, not "regions". This is why method that finds single voxel in MRI scan that is most likely to belong to lesion region was settled on (Fig. 3.19). Method for finding it is determining maximum magnitude of missing tissue feature map. This circumvents the issue arithmetic mean centre algorithms face when there are multiple lesions or when lesion arithmetic mean centre point does not actually belong to lesion region, with some exceptions.

As mentioned before, arithmetic mean centre algorithm with automatic threshold selection does have improved results; maximum magnitude approach could be considered an extreme case of that algorithm. Compared to non-automatically determined thresholds methods of arithmetic mean centre determination methods maximum magnitude approach has at least a 10% increase in accuracy in the initial testing (Table 3.2).

In the end, applying pre-filtering of data, missing feature map thresholds and the maximum magnitude method allowed analysing 100 MRI scans and correctly finding points belonging to lesion in 91 of them (91% accuracy)(Table 3.3). At this point only two things confuse the algorithm. First is significant brain asymmetry (Fig. 3.20 first two) – when there is a sizeable region of CSF on one hemisphere but not the other. CSF has similar visual properties to lesions,

Table 3.3. Results of maximum magnitude method tests on pre-filtered MRI scans.

Threshold	Hits	Total	Accuracy
0.9	3	3	100
0.8	8	9	89
0.7	14	16	88
0.6	21	23	91
0.5	33	36	92
0.4	44	49	90
0.3	59	65	91
0.2	76	84	90
0.1	91	100	**91**
0	95	113	84

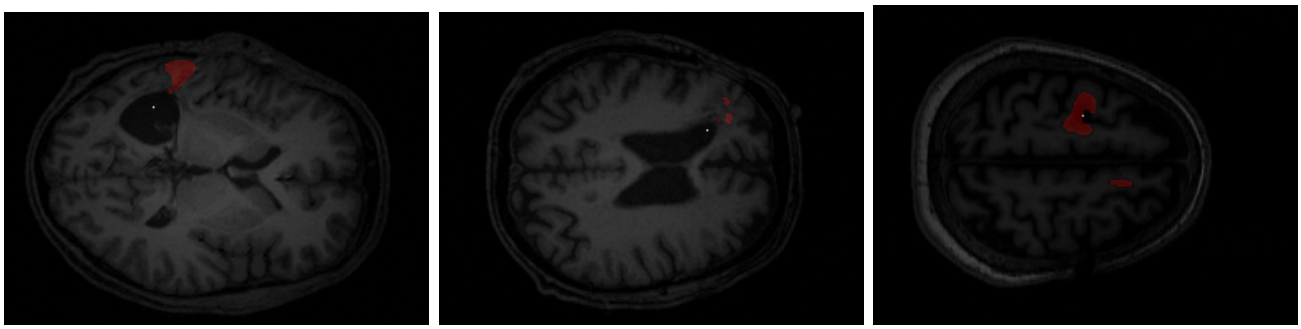


Fig. 3.20. Maximum magnitude method unsuccessful point determination examples. Red area is hand delineated region; white pixel close by is the found voxel.

thus the false identification. Second, could be argued against. As seen in third brain slice in figure 3.20, CSF is surrounded by lesion region. Algorithm is unable to differentiate between lesion and CSF, and in the missing tissue probability map it picks voxel right next to region delineated by hand.

3.4 Results of region growing

Region growing as a solution for ischemic insult lesion delineation from the start seemed like a bad idea and it does not reach good enough results. Main issue being that the intensity of lesion voxels is similar to CSF, which causes region to overgrow, or not grow enough under more constraining parameters.

I took a single MRI scan at random and tried a variety of parameters to attain similar lesion delineation to hand delineated version (Fig. 3.21). Best DSC that was achieved out of 4032 parameter variations was 0.46 – less than half of the lesion delineation coincides. Parameters are: inclusion cube radius of 3, inclusion cube threshold of 3, rejection cube radius of 5, rejection cube threshold of 4 and at least 20% of rejection cube must be lesion-like. This way 14 327 voxels were included. Hand delineated version has 28 989 voxels. Voxel size is 1 mm^3 actual volume, so automatic delineation finds 14 cm^3, while it should find around 29 cm^3. Next step in an attempt to improve the results is looking at the influence of all the parameters.

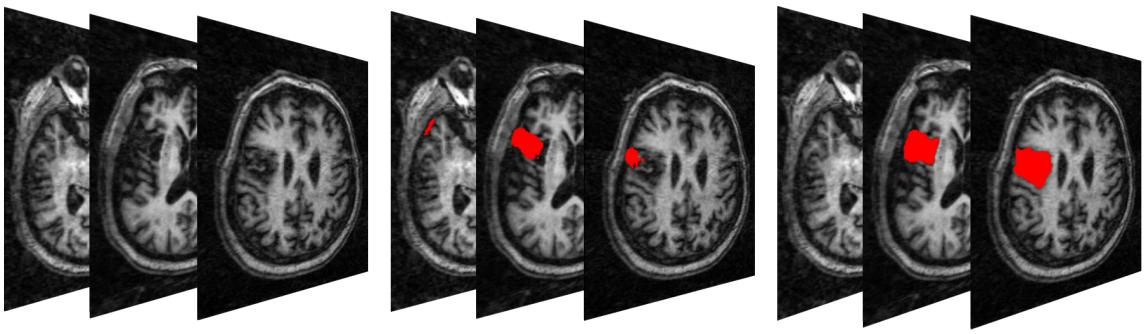


Fig. 3.21. MRI scan used in parameter testing: left – original, middle – best DSC region grown, right – hand delineated region.

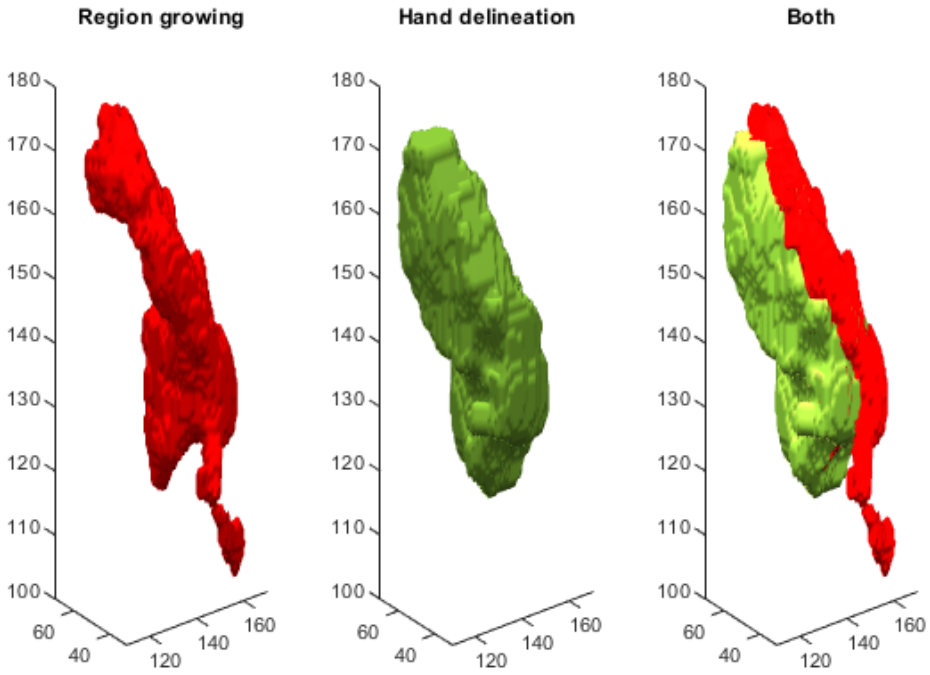


Fig. 3.22. 3D models for region grown delineation and hand delineation.

One of the parameters is rejection cube threshold. It is closely tied to rejection cube radius. Both of these parameters influence on DSC is visible in Figure 3.23. While using aforementioned parameters provide highest DSC, it seems that there is a tendency of higher DSC values at higher thresholds, no matter what radius is used. This then implies that instead a threshold of 20 or more should be used; however that is not the case. Here data is distorted by the 100 000 cycles limiting – region with higher thresholds is thicker, which makes it cover more hand delineated region, but it overflows and continues through the CSF and without limiting region would grow through entire head diminishing DSC in the process.

Similar tendencies can be seen in rejection cube required lesion-like voxels percent dependency impact to DSC (Fig. 3.24). At threshold of 4, the best case found earlier, DSC is highest, but drops off quickly if higher percentage is required to be lesion-like in rejection cube to pass growth criteria evaluation. And at higher threshold change seems to even out. This is also due to 100 000 cycle limit.

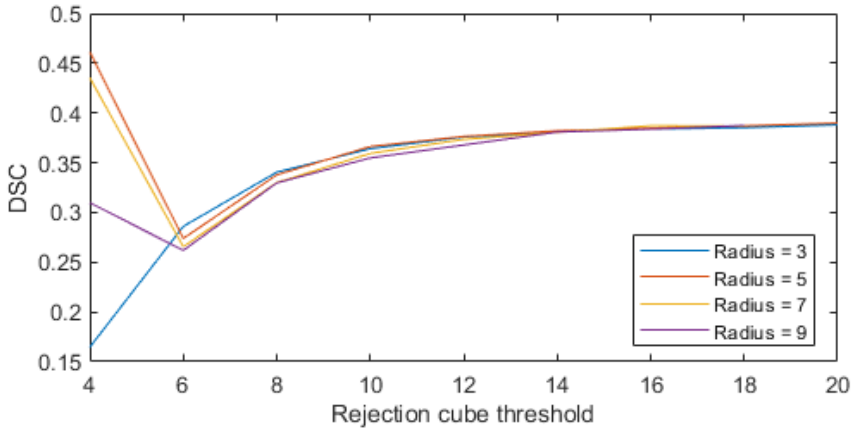


Fig. 3.23. DSC dependence on rejection cube threshold value at different radii.

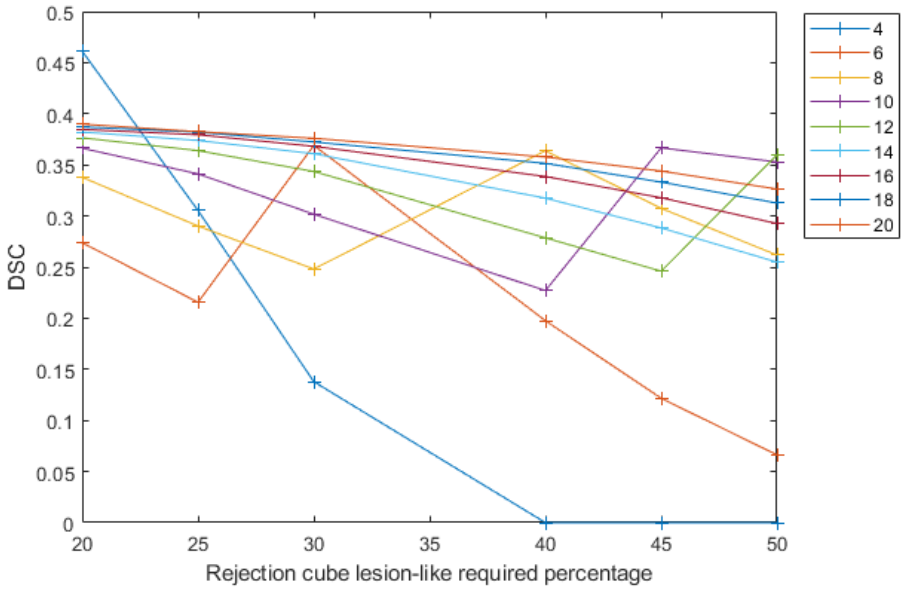


Fig. 3.24. DSC dependence on rejection cube lesion-like required percentage at different thresholds.

To see if it is indeed the limit that causes confusing data, an example run was performed on same MRI scan with same best parameters just increasing rejection cube threshold to 20 and increasing cycle limit to 500 000. Previously at such settings DSC was close to 0.4; now with limit increased it is just 0.0049. Though, still these graphs show that smaller rejection cube values might improve results, as two of the parameters and at the low end.

Further test runs indicate that threshold of 2 and radius of 6 at 10% lesion-like region requirement would improve result. But the improvement is minuscule, less than 0.01, and when full algorithm is run with these parameters overall results are significantly worse. So the initial aforementioned parameters were maintained. More testing allowed to achieve 0.79 DSC (from 0.66 DSC at initial best parameters), but the average drops – less average quality results. So instead of cherry picking an making it work for a few MRI scans this work focuses on having a higher average.

Adding anisotropic diffusion filter, overall, improves DSC. Though changes are not large:

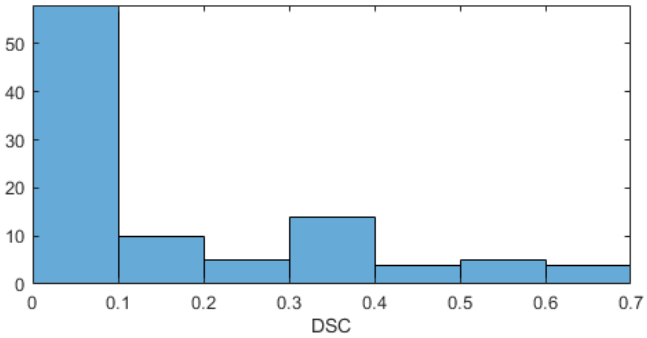


Fig. 3.25. My method DSC histogram after running on all 121 MRI scans without filtering the original MRI scan.

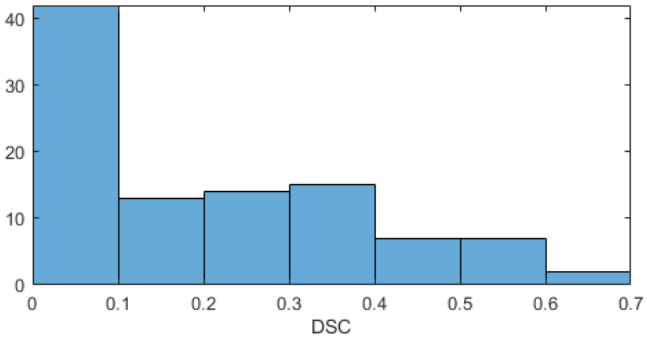


Fig. 3.26. My method DSC histogram after running on all 121 MRI scans.

best DSC dropped by 0.01 and the average increased, by 0.05. While many still fail (42 have less than 0.1 DSC) 16 MRI scans move up into higher than 0.1 range (58 less than 0.1 DSC with non-filtered input).

All in all, running now fully assembled algorithm on full set of 100, DSCs are still not good enough (Fig. 3.26). Considering cases where lesion region failed to be determined as having DSC of 0, highest DSC is 0.66 and average is 0.20. It was not expected to have perfect DSC due to the highest prevailing issue of CSF being included in the region, while in hand delineated regions it is not included. Both maximum and average DSCs need to be higher for this approach to be viable, but now only 2 out of a 100 (2%) have DSC higher than 0.6, "good" by Griffis standards [34], and none above 0.9, which would be ideal.

3.5 Validation conclusions

All in all, validation shows that automatic delineation algorithms so far are incapable of delineating lesion regions comparable to regions delineated by hand by a professional. Griffis algorithm without being trained for a new set is not capable of giving consistent satisfactory results on a new set of input MRI scans. While it does reach respectable similarity at some MRI scan slices to hand delineated lesions, at one point reaching DSC of 0.9 and an overall DSC of over 0.8, Griffis algorithm is far from consistent and only 14% of delineations of MRI scans fed to the algorithm in this work can be considered "good" [34].

On the other hand Griffis algorithm provides a robust framework for finding around where the brain stroke lesion is in an MRI scan. Utilizing SPM toolbox and missing TPM computation formula gives good data to make a decision where a lesion is located. Taking the largest value from the missing TPM, with threshold based filtering applied, gives 91 % chance of picking a voxel belonging to a hand delineated lesion region.

And to try and find the volume and location of lesion region in an MRI scan region growing is applied. The features added to region growing algorithm in this work do improve DSC values, meaning better delineation, but they are far from sufficient. Even in best case, lesion volume is near two times larger than in hand delineated version; DSC reaches only 0.67 and average, across 100 MRI scans tested, is just 0.20. Only 2% of all the scans reach DSC over 0.6 – "good" as defined by Griffis. So this approach did not measure up to expectations. And more likely a different approach than region growing should be taken.

CONCLUSIONS

This work explores methods and algorithms applied in trying to delineate various types of regions within brains. The focus is mostly on finding brain stroke induced ischemic lesion regions in T1-weighted scans. Due to rise of popularity of neural networks, classification, fuzzy logic and such, many approaches utilize one or a few of these. This introduces need for training for the algorithms. Here one of such algorithms is explored – automatic lesion delineation utilizing naïve Bayes classification proposed by Joseph C. Griffis, Jane B. Allendorfer and Jerzy P. Szaflarski.

Their work employs SPM toolbox for MATLAB which is a powerful tool capable of segmenting brains in MRI scans to grey matter, white matter and CSF. This toolbox also is employed in the algorithm proposed in this work. It provides means of finding missing TPM. Griffis algorithm utilizes it and abnormal TPM in his prediction system to further refine it into lesion delineation. Since this system incorporates training, giving MRI scans it is not trained for it was able to delineate 14% of scans to a satisfactory degree.

Other method overviewed in this work is from Swiss researchers. It does not require training and in theory should work; while in this work that method was not capable of producing usable results it does strive for the same – lesion delineation without requiring training.

This works method starts off same as Griffis method – SPM based segmentation followed by missing TPM calculation. At that point algorithm diverges and next up is finding a point for region growing. Under conditions that there is a sizeable lesion (at least roughly 1 cm^3 volume) it is capable of finding a lesion in over 90% of the cases. Its main issue being the asymmetry of the brain – in cases where there is a large pooling of CSF on one hemisphere without symmetrical one on the other hemisphere. Taking these false positives out would increase past 95% percent in the test set. And would improve results of the final step.

Last step in method proposed in this work is region growing. With the improvements applied to the basic region growing algorithm, in this work it is still incapable of giving satisfactory results – only 2% of scans reach minimum desired accuracy threshold.

In the end, this works method, while not capable of precisely measuring the volume of brain stroke induced ischemic lesion or delineate its region, is capable of pointing out MRI scans with significant issues and pointing out the area where the is most likely to be located; with an accuracy of over 90%. And, honestly, large pooling of CSF on one hemisphere is not likely to be healthy either.

REFERENCES

[1] Elizabeth Bates, Stephen M Wilson, Ayse Pinar Saygin, Frederic Dick, Martin I Sereno, Robert T Knight, and Nina F Dronkers. Voxel-based lesion–symptom mapping. *Nature neuroscience*, 6(5):448–450, 2003.

[2] Chris Rorden, Julius Fridriksson, and Hans-Otto Karnath. An evaluation of traditional and novel tools for lesion behavior mapping. *Neuroimage*, 44(4):1355–1362, 2009.

[3] Chris Rorden, Hans-Otto Karnath, and Leonardo Bonilha. Improving lesion-symptom mapping. *Journal of cognitive neuroscience*, 19(7):1081–1088, 2007.

[4] Rong Chen and Edward H Herskovits. Voxel-based Bayesian lesion–symptom mapping. *Neuroimage*, 49(1):597–602, 2010.

[5] Sharon Geva, Jean-Claude Baron, P Simon Jones, Cathy J Price, and Elizabeth A Warburton. A comparison of VLSM and VBM in a cohort of patients with post-stroke aphasia. *NeuroImage: Clinical*, 1(1):37–47, 2012.

[6] Edward A Ashton, Chihiro Takahashi, Michel J Berg, Andrew Goodman, Saara Totterman, and Sven Ekholm. Accuracy and reproducibility of manual and semiautomated quantification of MS lesions by MRI. *Journal of Magnetic Resonance Imaging*, 17(3):300–308, 2003.

[7] Jenny Crinion, Audrey L Holland, David A Copland, Cynthia K Thompson, and Argye E Hillis. Neuroimaging in aphasia treatment research: quantifying brain lesions after stroke. *Neuroimage*, 73:208–214, 2013.

[8] Marko Wilke, Bianca de Haan, Hendrik Juenger, and Hans-Otto Karnath. Manual, semi-automated, and automated delineation of chronic brain lesions: a comparison of methods. *Neuroimage*, 56(4):2038–2046, 2011.

[9] Julie A Fiez, Hanna Damasio, and Thomas J Grabowski. Lesion segmentation and manual warping to a reference brain: Intra-and interobserver reliability. *Human brain mapping*, 9(4):192–211, 2000.

[10] Sonya Mehta, Thomas J Grabowski, Yogi Trivedi, and Hanna Damasio. Evaluation of voxel-based morphometry for focal lesion detection in individuals. *Neuroimage*, 20(3):1438–1454, 2003.

[11] Mohamed L Seghier, Anil Ramlackhansingh, Jenny Crinion, Alexander P Leff, and Cathy J Price. Lesion identification using unified segmentation-normalisation models and fuzzy clustering. *Neuroimage*, 41(4):1253–1266, 2008.

[12] Jhimli Mitra, Pierrick Bourgeat, Jurgen Fripp, Soumya Ghose, Stephen Rose, Olivier Salvado, Alan Connelly, Bruce Campbell, Susan Palmer, Gagan Sharma, et al. Lesion segmentation from multimodal MRI using random forest following ischemic stroke. *NeuroImage*, 98:324–335, 2014.

[13] Oskar Maier, Matthias Wilms, Janina von der Gablentz, Ulrike M Krämer, Thomas F Münte, and Heinz Handels. Extra tree forests for sub-acute ischemic stroke lesion segmentation in MR sequences. *Journal of neuroscience methods*, 240:89–100, 2015.

[14] By members & collaborators of the Wellcome Trust Centre for Neuroimaging. Statistical Parametric Mapping - SPM. Online. Accessed November 2016 at: www.fil.ion.ucl.ac.uk/spm/software/.

[15] Alex P Zijdenbos and Benoit M Dawant. Brain segmentation and white matter lesion detection in MR images. *Critical reviews in biomedical engineering*, 22(5-6):401–465, 1993.

[16] Bruno Alfano, Arturo Brunetti, Michele Larobina, Mario Quarantelli, Enrico Tedeschi, Andrea Ciarmiello, Eugenio M Covelli, and Marco Salvatore. Automated segmentation and measurement of global white matter lesion volume in patients with multiple sclerosis. *Journal of Magnetic Resonance Imaging*, 12(6):799–807, 2000.

[17] Miguel Angel Gonzalez Ballester, Andrew Zisserman, and Michael Brady. Segmentation and measurement of brain structures in MRI including confidence bounds. *Medical Image Analysis*, 4(3):189–200, 2000.

[18] Koen Van Leemput, Frederik Maes, Dirk Vandermeulen, Alan Colchester, and Paul Suetens. Automated segmentation of multiple sclerosis lesions by model outlier detection. *IEEE transactions on medical imaging*, 20(8):677–688, 2001.

[19] Remi K-S Kwan, Alan C Evans, and G Bruce Pike. An extensible MRI simulator for post-processing evaluation. In *Visualization in biomedical computing*, pages 135–140. Springer, 1996.

[20] RK-S Kwan, Alan C Evans, and G Bruce Pike. MRI simulation-based evaluation of image-processing and classification methods. *IEEE transactions on medical imaging*, 18(11):1085–1097, 1999.

[21] David C Taylor and William A Barrett. Image segmentation using globally optimal growth in three dimensions with an adaptive feature set. In *Visualization in Biomedical Computing 1994*, pages 98–107. International Society for Optics and Photonics, 1994.

[22] Pedro Domingos and Michael Pazzani. On the optimality of the simple Bayesian classifier under zero-one loss. *Machine learning*, 29(2-3):103–130, 1997.

[23] Rajeev DS Raizada and Yune-Sang Lee. Smoothness without smoothing: why Gaussian naive Bayes is not naive for multi-subject searchlight studies. *PloS one*, 8(7):e69566, 2013.

[24] Irina Rish, Joseph Hellerstein, and Jayram Thathachar. An analysis of data characteristics that affect naive bayes performance. *IBM TJ Watson Research Center*, 30, 2001.

[25] DL Collins, AC Evans, C Holmes, and TM Peters. Automatic 3D segmentation of neuroanatomical structures from MRI. In *Information processing in medical imaging*, volume 3, pages 139–152. Kluwer Dordrecht, 1995.

[26] Jenny Crinion, John Ashburner, Alex Leff, Matthew Brett, Cathy Price, and Karl Friston. Spatial normalization of lesioned brains: performance evaluation and impact on fMRI analyses. *Neuroimage*, 37(3):866–875, 2007.

[27] P Ripolles, Josep Marco-Pallarés, Ruth de Diego-Balaguer, J Miro, M Falip, M Juncadella, F Rubio, and Antoni Rodriguez-Fornells. Analysis of automated methods for spatial normalization of lesioned brains. *Neuroimage*, 60(2):1296–1306, 2012.

[28] David MacDonald, Noor Kabani, David Avis, and Alan C Evans. Automated 3-D extraction of inner and outer surfaces of cerebral cortex from MRI. *NeuroImage*, 12(3):340–356, 2000.

[29] Alain Pitiot, Hervé Delingette, Paul M Thompson, and Nicholas Ayache. Expert knowledge-guided segmentation system for brain MRI. *NeuroImage*, 23:S85–S96, 2004.

[30] Edward Ashton, Saara Totterman, Chihiro Takahashi, Jose Tamez-Pena, and Kevin J Parker. Automated measurement of structures on CT and MR imagery: A validation study. In *Proceedings of the Fourteenth IEEE Symposium on Computer-Based Medical Systems*, page 300. IEEE Computer Society, 2001.

[31] John Ashburner, Karl J Friston, et al. Nonlinear spatial normalization using basis functions. *Human brain mapping*, 7(4):254–266, 1999.

[32] John Ashburner and Karl J Friston. Unified segmentation. *Neuroimage*, 26(3):839–851, 2005.

[33] David W Shattuck, Stephanie R Sandor-Leahy, Kirt A Schaper, David A Rottenberg, and Richard M Leahy. Magnetic resonance image tissue classification using a partial volume model. *NeuroImage*, 13(5):856–876, 2001.

[34] Joseph C Griffis, Jane B Allendorfer, and Jerzy P Szaflarski. Voxel-based Gaussian naïve Bayes classification of ischemic stroke lesions in individual T1-weighted MRI scans. *Journal of neuroscience methods*, 257:97–108, 2016.

[35] Edward A Ashton, Kevin J Parker, Michel J Berg, and Chang Wen Chen. A novel volumetric feature extraction technique with applications to MR images. *IEEE transactions on medical imaging*, 16(4):365–371, 1997.

[36] Bianca de Haan, Philipp Clas, Hendrik Juenger, Marko Wilke, and Hans-Otto Karnath. Fast semi-automated lesion demarcation in stroke. *NeuroImage: Clinical*, 9:69–74, 2015.

[37] Xin-Wei Li, Qiong-Ling Li, Shu-Yu Li, and De-Yu Li. Local manifold learning for multi-atlas segmentation: application to hippocampal segmentation in healthy population and alzheimer's disease. *CNS Neuroscience & Therapeutics*, 21(10):826–836, 2015.

[38] Rola Harmouche, Nagesh K Subbanna, D Louis Collins, Douglas L Arnold, and Tal Arbel. Probabilistic multiple sclerosis lesion classification based on modeling regional intensity variability and local neighborhood information. *IEEE Transactions on Biomedical Engineering*, 62(5):1281–1292, 2015.

[39] Matthew A Petoe, Winston D Byblow, Esther JM de Vries, Venkatesh Krishnamurthy, Cathy S Zhong, P Alan Barber, and Cathy M Stinear. A template-based procedure for determining white matter integrity in the internal capsule early after stroke. *NeuroImage: Clinical*, 4:695–700, 2014.

[40] Celine R Gillebert, Glyn W Humphreys, and Dante Mantini. Automated delineation of stroke lesions using brain CT images. *NeuroImage: Clinical*, 4:540–548, 2014.

[41] Colm Elliott, Douglas L Arnold, D Louis Collins, and Tal Arbel. Temporally consistent probabilistic detection of new multiple sclerosis lesions in brain mri. *IEEE Transactions On medical imaging*, 32(8):1490–1503, 2013.

[42] Xavier Lladó, Onur Ganiler, Arnau Oliver, Robert Martí, Jordi Freixenet, Laia Valls, Joan C Vilanova, Lluís Ramió-Torrentà, and Àlex Rovira. Automated detection of multiple sclerosis lesions in serial brain MRI. *Neuroradiology*, 54(8):787–807, 2012.

[43] Diane M Renz, Horst K Hahn, Peter Schmidt, Jan Rexilius, Markus Lentschig, Alexander Pfeil, Dieter Sauner, Clemens Fitzek, Hans-Joachim Mentzel, Werner A Kaiser, et al. Accuracy and reproducibility of a novel semi-automatic segmentation technique for MR volumetry of the pituitary gland. *Neuroradiology*, 53(4):233–244, 2011.

[44] Rong Chen, Argye E Hillis, Mikolaj Pawlak, and Edward H Herskovits. Voxelwise Bayesian lesion-deficit analysis. *NeuroImage*, 40(4):1633–1642, 2008.

[45] Joseph C. Griffis. lesion gnb toolbox for SPM12. Online, March 2016. Accessed November 2016.

[46] W. Deng, W. Xiao, H. Deng, and J. Liu. Mri brain tumor segmentation with region growing method based on the gradients and variances along and inside of the boundary curve. In *2010 3rd International Conference on Biomedical Engineering and Informatics*, volume 1, pages 393–396, Oct 2010.

[47] Francisco Pereira, Tom Mitchell, and Matthew Botvinick. Machine learning classifiers and fMRI: a tutorial overview. *Neuroimage*, 45(1):S199–S209, 2009.

[48] Daniel. Region Growing (2D/3D grayscale). Online, 2018. Accessed April 2018.

[49] Lee R Dice. Measures of the amount of ecologic association between species. *Ecology*, 26(3):297–302, 1945.

[50] A. Fabijańska T. Węgliński. Brain tumor segmentation from mri data sets using region growing approach. In *Perspective Technologies and Methods in MEMS Design*, pages 185–188, May 2011.

[51] R. Noureddine, K. Tarhini, and S. Saleh. Segmentation and extraction of brain injury lesions from mri images: Matlab implementation. In *2015 International Conference on Advances in Biomedical Engineering (ICABME)*, pages 45–48, Sept 2015.

[52] R. B. Dubey, M. Hanmandlu, S. K. Gupta, and S. K. Gupta. Region growing for mri brain tumor volume analysis. *Indian Journal of Science and Technology*, 2(9), 2009.

[53] N Mohd Saad, SAR Abu-Bakar, Sobri Muda, M Mokji, and AR Abdullah. Automated region growing for segmentation of brain lesion in diffusion-weighted mri. In *Proceedings of the international MultiConference of Engineers and computer scientists*, volume 1, pages 14–17, 2012.

[54] Ishmam Zabir, Sudip Paul, Md Abu Rayhan, Tanmoy Sarker, Shaikh Anowarul Fattah, and Celia Shahnaz. Automatic brain tumor detection and segmentation from multi-modal mri images based on region growing and level set evolution. In *Electrical and Computer Engineering (WIECON-ECE), 2015 IEEE International WIE Conference on*, pages 503–506. IEEE, 2015.

[55] Baijiang Fan, Yunbo Rao, Wei Liu, Qifei Wang, and Huaiyu Wen. Region-based growing algorithm for 3d reconstruction from mri images. In *Image, Vision and Computing (ICIVC), 2017 2nd International Conference on*, pages 521–525. IEEE, 2017.

[56] Dorian Pustina, H Coslett, Peter E Turkeltaub, Nicholas Tustison, Myrna F Schwartz, and Brian Avants. Automated segmentation of chronic stroke lesions using LINDA: lesion identification with neighborhood data analysis. *Human brain mapping*, 2016.

APPENDICES

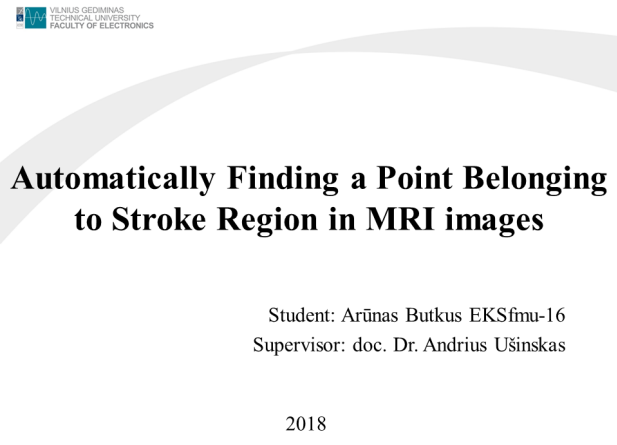
Appendix A Results of automatic stroke segmentation

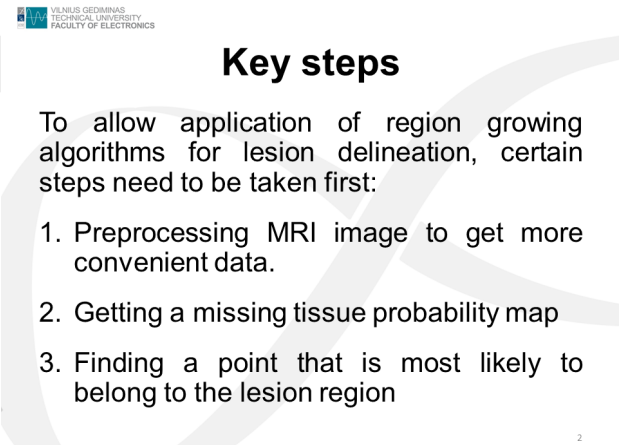
Patient MRI scan ID	Found center coordinates	Method overflowed into CSF	DSC
P0030_990122_4_3D_ax	[135, 183, 130]	No	0.0073294
P0053_970919_T1	[58, 87, 98]	No	0.14432
P0089_000721_8_3D_ax	[106, 158, 171]	No	0.055306
P0150_030807_8_3D_ax	[109, 187, 131]	No	0.56837
P0152_030910_7_3D_ax	[61, 149, 111]	No	0.1062
P0203_980403_3_3D_ax	[85, 174, 112]	No	0.000069711
P0204_980327_4_3D_ax	[49, 114, 138]	No	0.00047916
P0237_980813_5_3D_ax	[70, 129, 125]	No	0.000091253
P0268_981106_3_3D_ax	[85, 182, 134]	No	0.32698
P0270_981016_3_3D_ax	[141, 130, 91]	No	0.2647
P0273_981106_3_3D_ax	[123, 158, 135]	Yes	0
P0280_981113_3_3D_ax	[89, 186, 119]	No	0.071893
P0291_981211_4_3D_ax	[62, 141, 101]	No	0.36505
P0300_030827_8_3D_ax	[65, 167, 133]	No	0.65283
P0307_030911_7_3D_ax	[42, 95, 119]	No	0.16509
P0342_030930_8_3D_ax	[120, 181, 144]	No	0.35335
P0370_990730_4_3D_ax	[139, 149, 110]	No	0.18767
P0392_991001_4_3D_ax	[138, 151, 82]	No	0.35977
P0403_991119_4_3D_ax	[49, 138, 131]	No	0.00014799
P0447_000324_4_3D_ax	[140, 141, 95]	No	0.29393
P0466_000526_5_3D_ax	[48, 175, 132]	No	0.36028
P0495_000728_5_3D_ax	[135, 149, 142]	No	0.31195
P0520_001020_5_3D_ax	[125, 178, 128]	No	0.4011
P0550_010223_5_3D_ax	[50, 144, 138]	No	0.46117
P0591_010629_5_3D_ax	[58, 185, 142]	No	0.084429
P0612_010810_5_3D_ax	[110, 86, 175]	No	0.18641
P0622_010911_2_3D_ax	[58, 118, 148]	No	0.34298
P0732_040803_2_3D_ax	[59, 179, 150]	No	0.52841
P0743_020702_2_3D_ax	[50, 152, 100]	No	0.00022043
P0751_020723_2_3D_ax	[53, 137, 123]	No	0
P0755_020806_2_3D_ax	[54, 189, 131]	No	9.5039e-05
P0767_021008_3_3D_ax	[49, 174, 130]	No	0.2755
P0792_021105_2_3D_ax	[55, 153, 155]	No	0.31975
P0826_030218_2_3D_ax	[53, 131, 169]	No	0.00031862
P0853_030401_2_3D_ax	[44, 116, 144]	No	0.15926
P0921_031014_2_3D_ax	[49, 166, 159]	No	0.0007776
P0957_040210_2_3D_ax	[54, 85, 140]	No	0.003457
P0960_031209_2_3D_ax	[68, 175, 151]	No	0.26985
P1044_040511_2_3D_ax	[44, 119, 157]	No	0.44922
P1058_040525_2_3D_ax	[101, 177, 148]	No	0.00067204
P1066_040622_2_3D_ax	[54, 160, 147]	No	0.12045

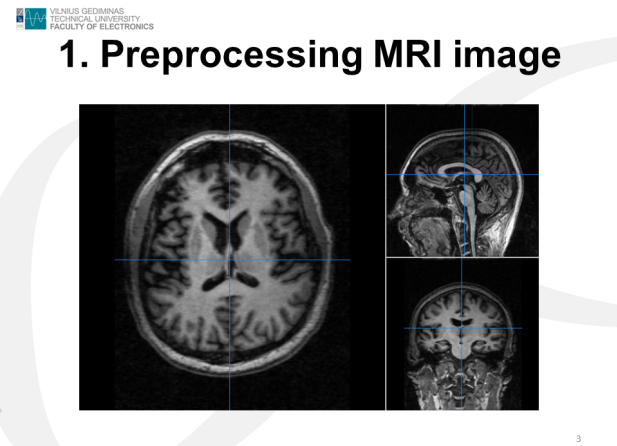
Patient MRI scan ID	Found center coordinates	Method overflowed into CSF	DSC
P1077_040629_2_3D_ax	[77, 159, 178]	No	0.46663
P1108_040831_2_3D_ax	[138, 143, 134]	No	0.15331
P1117_040921_2_3D_ax	[67, 152, 173]	No	0
P1149_041123_2_3D_ax	[99, 175, 128]	No	0.3645
P1167_041221_2_3D_ax	[115, 165, 171]	No	0.051726
P1170_050104_2_3D_ax	[120, 166, 163]	No	0.41163
P1187_050208_T1	[129, 163, 77]	No	0.28992
P1233_050419_1_3D_ax	[80, 165, 180]	No	0.54429
P1264_050906_1_3D_ax	[66, 167, 146]	No	0.0047847
P1268_050726_1_3D_ax	[66, 107, 195]	No	0.39219
P1270_050705_1_3D_ax	[59, 143, 119]	No	0.042591
P1281_050906_2_3D_ax	[44, 147, 130]	No	0.2556
P1287_050809_T1	[55, 89, 93]	No	0.089576
P1299_050913_2_3D_ax	[130, 145, 141]	No	0.21165
P1419_060404_1_3D_ax	[54, 61, 176]	No	0.00010464
P1470_060613_7_3D_ax	[101, 171, 144]	No	0.000086926
P1476_060613_T1	[120, 156, 93]	No	0.65604
P1550_061128_2_3D_ax	[32, 122, 179]	No	0
P1553_061205_T1	[104, 176, 64]	No	0.34268
P1574_070109_2_3D_T1_mdeft	[76, 76, 153]	No	0
P1575_070116_2_3D_T1_mdeft	[80, 180, 141]	Yes	0
P1600_070417_2_3D_T1_mdeft	[116, 193, 155]	No	0.57397
P1654_070612_2_3D_ax	[76, 92, 204]	No	0.1562
P1663_070626_2_3D_T1_mdeft	[47, 77, 114]	Yes	0
P1664_070619_2_3D_T1_mdeft	[112, 200, 138]	No	0.32829
P1706_070904_2_3D_ax	[42, 156, 152]	No	0.023411
P1712_070925_2_3D_T1_mdeft	[52, 138, 121]	No	0.5265
P1723_071009_2_3D_T1_mdeft	[55, 156, 180]	No	0.39793
P1730_071016_2_3D_T1_mdeft	[84, 181, 180]	Yes	0
P1830_080429_11_3D_ax	[73, 120, 174]	No	0.0015548
P1837_080506_2_3D_T1_mdeft	[116, 149, 172]	No	0
P1854_080520_2_3D_T1_mdeft	[87, 163, 123]	No	0.28915
P1857_080603_T1	[59, 145, 92]	No	0.47882
P1876_080715_2_3D_T1_mdeft	[74, 180, 139]	No	0.2528
P1878_080715_2_3D_T1_mdeft	[61, 130, 155]	Yes	0
P1886_080722_2_3D_T1_mdeft	[58, 139, 155]	No	0.59107
P1897_080812_2_3D_T1_mdeft	[67, 127, 186]	No	0.030231
P1909_080902_2_3D_T1_mdeft	[112, 201, 137]	No	0.000075672
P1927_081021_2_3D_T1_mdeft	[84, 162, 179]	No	0.4542
P1947_081118_T1	[130, 172, 69]	No	0.12664
P2003_090331_2_3D_T1_mdeft	[80, 165, 189]	No	0.54306
P2023_090428_2_3D_T1_mdeft	[141, 123, 127]	Yes	0
P2031_120611_t1_mpr_sag	[75, 185, 150]	No	0.000077187
P2062_120612_t1_mpr_sag	[101, 186, 145]	No	0.000040951
P2085_090804_2_3D_T1_mdeft	[107, 97, 203]	No	0.258

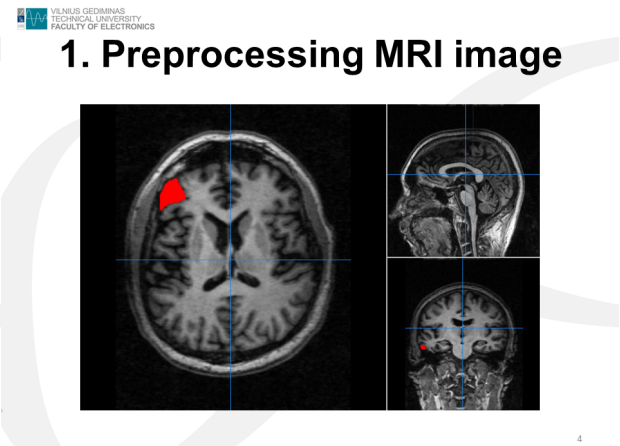
Patient MRI scan ID	Found center coordinates	Method overflowed into CSF	DSC
P2161_100105_2_3D_T1_mdeft	[62, 172, 159]	No	0.25629
P2169_100119_2_3D_T1_mdeft	[85, 201, 128]	No	0.24598
P2174_100126_2_3D_T1_mdeft	[133, 126, 144]	No	0.2619
P2186_100216_2_3D_T1_mdeft	[49, 103, 168]	No	0.32146
P2198_100309_2_3D_T1_mdeft	[57, 175, 147]	No	0.12051
P2222_100427_2_3D_T1_mdeft	[77, 177, 166]	No	0.31951
P2376_120613_t1_mpr_sag	[100, 178, 180]	No	0.000057864
P2503_110927_00176-1	[46, 150, 189]	No	0.10175
P2517_111004_T1	[46, 142, 122]	No	0.000096103
P2697_120710-00176-1	[43, 147, 129]	No	0.00048603
P2734_120828-00176-1	[47, 160, 166]	No	0.00010324
P2783_121127-00176-1	[103, 185, 158]	No	0.2835
P2802_130115-00176-1	[54, 86, 190]	No	0.14192
P2809_130122-00176-1	[80, 163, 169]	No	0.00024117

Appendix B Materials of presentation given at 21st Conference for Lithuanian Junior Researchers









VILNIUS GEDIMINAS TECHNICAL UNIVERSITY FACULTY OF ELECTRONICS

1. Preprocessing MRI image

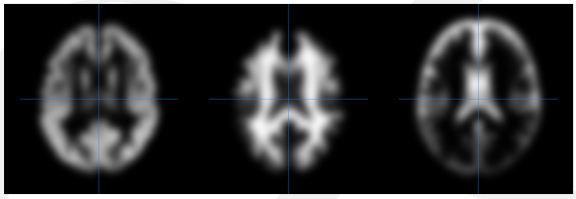
Gray matter TPM | White matter TPM | Cerebro-spinal fluid TPM

SPM12 segmentation results

5

VILNIUS GEDIMINAS TECHNICAL UNIVERSITY FACULTY OF ELECTRONICS

1. Preprocessing MRI image



Gray matter TPM | White matter TPM | Cerebro-spinal fluid TPM

SPM12 segmentation results with 11 voxel FWHM smoothing

6

VILNIUS GEDIMINAS TECHNICAL UNIVERSITY FACULTY OF ELECTRONICS

2. Getting a missing TPM

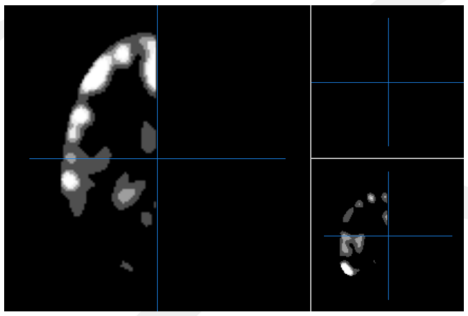
$$(CSF_{Affected} - CSF_{Unaffected}) * ((GM_{Unaffected} + WM_{Unaffected}) - (GM_{Affected} + WM_{Affected}))$$

$$(CSF_{Affected} - CSF_{Prior}) * ((GM_{Prior} + WM_{Prior}) - (GM_{Affected} + WM_{Affected}))$$

7

VILNIUS GEDIMINAS TECHNICAL UNIVERSITY FACULTY OF ELECTRONICS

2. Getting a missing TPM



8

VILNIUS GEDIMINAS TECHNICAL UNIVERSITY FACULTY OF ELECTRONICS

3. Finding lesion center

Geometric center using:

1. Manual threshold (`GTC`)
2. Weighted (`GWTC`)
3. Automatic threshold (`GWATC`)

3-dimensional maximum (`mmmx`)

9

VILNIUS GEDIMINAS TECHNICAL UNIVERSITY FACULTY OF ELECTRONICS

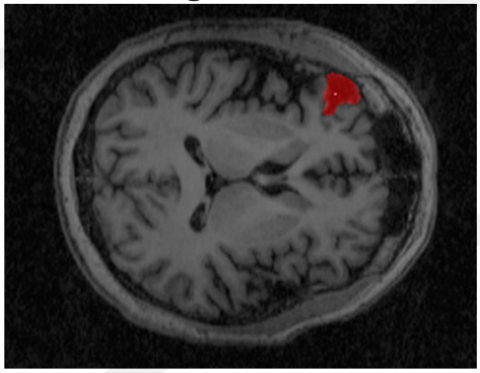
3. Finding lesion center

```
Overall scores:
      mmmx hit 89/113 [78.76%]
 GTC  >.5 hit 33/113 [29.20%]
 GWTC >.5 hit 33/113 [29.20%]
 GTC  >.1 hit 76/113 [67.26%]
 GWTC >.1 hit 77/113 [68.14%]
 GWTC > 0 hit 17/113 [15.04%]
GWATC 75% hit 84/113 [74.34%]
GWATC 50% hit 77/113 [68.14%]
GWATC 25% hit 66/113 [58.41%]
```

10

VILNIUS GEDIMINAS TECHNICAL UNIVERSITY FACULTY OF ELECTRONICS

3. Finding lesion center



11

VILNIUS GEDIMINAS TECHNICAL UNIVERSITY FACULTY OF ELECTRONICS

Results and discussions

Without predefined criteria, `mmmx` method has worked best in experiments. Defining few prerequisites could bring results close to 100%.

Additionally, other methods could be considered for lesion region center determination. Or even a different approach to getting lesion region itself, like post-processing missing tissue feature map.

12