BRAIN TISSUE SEGMENTATION IN MEDICAL IMAGE ANALYSIS USING SELF-ORGANIZING MAPS

Dr. Smitha Sunil Kumaran Nair
Associate Professor, Dept. of Computer Science and Engineering
Manipal University, Karnataka- 576104

Abstract: This paper deals with a machine learning algorithm used for the segmentation of brain into three tissues namely White Matter (WM), Gray Matter (GM) and Cerebrospinal Fluid (CSF) in Magnetic Resonance Images (MRI). The proposed work describes and discusses on how to preprocess the MR image and how to create, initialize, train and interpret a Self-Organizing Map to separate the brain tissues. Finally, the proposals based on Evolutionary Algorithms given are seen to have high potential for improvement and promise to lead to high accurate segmentation in future work.

Keywords: Image denoising, skull stripping, MRI segmentation, Self-Organizing Maps

Introduction
The accurate classification of brain tissues was a popular field of research in the past years which is still of relevance because of its importance in medical diagnosis. Therefore several different approaches have been developed for medical image segmentation [1]: fuzzy models, non-statistical uncertainty, statistical recognition, atlas-based methods, supervised/unsupervised classifier methods like Bayes classifiers with maximum likelihood, k-nearest neighbor and so on.

Furthermore there are different kinds of Artificial Neural Network (ANN) models like Feed forward, Back propagation, Hopfield [2], Kohonen (SOM), Adaptive Resonance Theory (ART), Learning Vector Quantization (LVQ) [3], Probabilistic Neural Network (PNN) [4] and other approaches with Wavelets [5], Support Vector Machine (SVM) [6] and knowledge-based expert systems [7] exist.

In this paper, we propose tissue segmentation based on Self Organizing Map (SOM).

Image pre-processing

Image denoising

MRI is an imaging technique based on the measurement of magnetic field vectors generated after an appropriate excitation with strong magnetic fields and radio-frequency pulses in the nuclei of hydrogen atoms present in the water molecules of the patient’s tissues. Given that the content of water differs for each tissue (e.g. bone and muscle), it is possible to quantify the differences of radiated magnetic energy, and have elements to identify each tissue.

Like every other real world entities, non-synthetic image MRI images are also corrupted by some noise. The sensors and amplifiers of the image acquiring devices are the source for a part of this undesired additive signals. As with all denoising filters, it is a challenge not to cause too much inevitably blurring of finer features.
carried in high frequency modes. One approach to reach this goal is to detect edges first to have knowledge about smooth regions and reconstruct afterwards [8]. Also Wiener Filter [9] and Kalman Filter are used to reduce noise, but the advantages of wavelet transform and the results of using these for denoising purpose changed the direction of research focus onto wavelets, so that a lot of applications, including MRI denoising and related papers are available [10]. A first solid mathematical footing has been established by Mallat [11]. Papers to many different wavelet approaches are available (e.g. [12] (Wavelet Shrinkage), [13]), and nowadays with respect to the Rician-distribution of the noise in MRI specialized Rician-Wavelet denoise filters are developed [14].

The wavelet packet method is a generalization of wavelet decomposition that offers a richer range of possibilities for signal analysis. Wavelet analysis consists of decomposing a signal or an image into a hierarchical set of approximations (A) and details (D). The approximation is then itself split into a second-level approximation and detail, and the process is repeated.

To denoise the MR image, several experiments have been conducted with different wavelets and parameters. Applying Daubechies-5 wavelet denoising to a level of 3 using Matlab Wavelets toolbox (wpdencmp) leads to accurate results as in fig. 1.

As a last step of image enhancement, a Matlab deblurring filter (deconvlucy) is applied to the MRI, using Lucy- Richardson method. This deconvolution filter uses an iterative approach to recover an image degraded by a known point spread function (PSF). It is assumed that the blurring causing PSF can be approximated by a Gaussian distribution. Therefore, a Lucy-Richardson deconvolution filter with a size of 3 pixel and a deviation = 3 is applied to the MR image.

Figure 1 Comparison: Original MRI, denoised MRI (wavelet Daubechies-5, level 3) and absolute difference

**Skull stripping**

For automatic classification of the three different brain matters it is essential, to initially separate the complete brain matter from the surrounding non brain portions like skull and dura. Otherwise misclassifications would occur. For a human viewer it is very easy to use the dark cerebrospinal fluid border to distinguish between brain matter and skull. But in computer vision this is more complicated, especially in regions, where this border gets very thin or even clears away. Human can easily handle this scenes, e.g. because of the humans’ a priori knowledge about the shape of a brain.

A basic approach to do skull stripping is demonstrated below. An overview is shown in fig 2. The main idea is, to identify after binarization, the biggest connected area, which has to be the brain matter [29].

![Skull stripping](image)

**Figure 2. Overview of necessary operations for skull strip**

Therefore the MR image is binarized with a threshold, processed with morphological operations, then biggest area of connected foreground pixel is
found and finally more morphological operations are applied to this area to retrieve a binary mask of the brain matter. This is used to convert all undesired tissues to background as in fig 3.

There can be scenes in MR images, where the border between brain and skull tissue is unclear, blurry or even nonexistent due to the connection to other tissue (e.g. by eye tissue). Consequence of this is that too much tissue (CSF) or undesired tissue areas are not properly cut away. This can be a problem, because it disturbs the automatism of the whole process. Therefore the Watershed algorithm (first introduced by Beucher and Lantuejoul [15]) is proposed to solve this problem. It is a non-parametric contour detection method. To explain how it works, imagine it interprets the gray scale value distribution of the input image as a relief surface, which consists of water catchment basins and watersheds, dependent on the gray scale pixel value. Step by step, that means gray scale value by gray scale value respectively water height levels, these basins are filled up with imaginary water. At positions where the water would flow over the watershed, the contour border is built in the shortest way, as if the watershed is enlarged by sandbags about the amount, which is necessary to prevent water from flowing over. This method promises to be very strong exactly in those cases, where the above described morphological method lacks robustness, because Watershed could be able to build a border between the two "basins" brain and eye in an MR image. The Watershed algorithm [16] was already successful in medical imaging [17], [18], [19] and was improved through several other approaches (e.g. Wavelets [20], [21]).

Figure 3. Comparison: Denoised MRI, morphologically generated binary mask of brain matter, skull stripped brain matter

Tissue classification using som

The main algorithm is divided into five subtasks. An overview is given in fig. 4. First, appropriate features have to be extracted and processed. Then a SOM is created, initialized and trained with the set of extracted features. The quality of the SOM can be measured and therefore Evolutionary Algorithm can be applied for optimizing parameters of feature extraction and SOM creation. After clustering the SOM, images of the segmented tissues can be built.

Feature extraction

To let an ANN classify the tissue types, first a feature space has to be developed. This comprises calculating the values of features for each pixel (or block of pixels) in the image. The quality of the segmentation is highly dependent on the quality of the features used to describe the image. Therefore, it is very important that these features suitably characterize the aspects of the image on which the segmentation is to be based. Particularly when pixel gray level alone is used as a feature, this can result in a large number of misclassifications. To avoid such misclassifications, several features of subareas around the pixel of interest are calculated [3, 22, 23]. This prevents misclassifications due to noise in the image and inevitable overlaps in the gray level distributions between different regions and improves accuracy. These features make up a feature vector for each pixel so that each is represented by a point.
in a multidimensional feature space.

The following features are proposed as promising in literature [22, 23]:
- gray scale value
- smoothed pixel value in subarea output of Laplacian filter at each pixel in subarea
- averaged pixel value in subarea variance of pixel value in subarea
- skewness of the pixel value histogram in subarea
- energy of the pixel value distribution in the subarea minimum and maximum pixel value in subarea
- difference from the minimum and maximum range between minimum and maximum variants in subarea intensity gradient in subarea for 0°, 45°, 90° and 135°.

Figure 4. Overview of steps of classification process

In this work, a feature vector consisting of six features is created: original pixel, averaged pixel, Laplace filter output, entropy, standard deviation and range between minimum and maximum pixel. To present it easily to a Neural Network, it is built with the following shape:

\[
\begin{bmatrix}
    f_1 s_1 & f_2 s_1 & \cdots & f_n s_1 \\
    f_1 s_2 & f_2 s_2 & \cdots & \vdots \\
    \vdots & \vdots & \ddots & \vdots \\
    f_1 s_m & \cdots & \cdots & f_n s_m 
\end{bmatrix}
\]

(1)

Where \( f_n \) is the n-th feature and \( s_m \) the m-th sample.

**Self-Organizing Maps**

MR images are large data sets with a significant number of independent variables and complex relationships, which usually show a nonlinear character which makes classical statistical methods particularly inappropriate for their analysis. This suggests that ANNs are good candidates to analyze such MR data and classify different tissues in terms of texture, intensity or contrast [24, 25].

Another promising candidate for medical image segmentation is Self-Organizing Maps also called Kohonen-maps. SOMs were introduced in 1989 by Teuvo Kohonen [26]. They consist of a two-dimensional or in general low dimensional grid of neurons, where each neuron is represented by a vector of weights (also called codebook vector or prototype vector) with the size of the input vectors. Also the neurons are related to their adjacent neighbors in a way determined by the topology of the map.

The fundamental idea is to find the one particular neuron, whose weights are closest to the input vector. This means:

\[
\|x - m_{BMU}\| = \min_i \{\|x - m_i\|\}
\]

(2)

This neuron is called Best Matching Unit (BMU). The weights of this special neuron are updated in such a way, that it gets closer to the input vector. Because there is a topological relationship between the neurons inside the grid, it is also
possible to select neurons from the adjacent neighborhood in a particular manner in order to update them as well. As a result similar weight vectors group together, in other words nearby data vectors in the input space are mapped onto neighboring locations in the output space. Furthermore there are two different modes of training: sequential and batch training. In the first case, in every training step a randomly chosen sample vector \( x \) from the input data set is taken to calculate the distance to all weight vectors of the SOM. This determines the (BMU) and their neighborhood, which is updated according to the following update rule:

\[
m_i(t + 1) = m_i(t) + \alpha(t) h_{BMU_i}(t) [x(t) - m_i(t)]
\]

where \( \alpha \) is a learning rate and \( h_{BMU_i} \) the neighborhood kernel around the BMU.

In the second case of batch training the whole input dataset instead of a single vector is presented to the network. Then the new weight vector is a weighted average of the data samples in accordance with neighborhood function.

Even if MATLAB has a built-in toolbox for Neural Networks including SOMs, the external SOM-Toolbox of the Helsinki University of Technology, the place where Kohonen first developed the concept of SOMs, is used [27]; mainly because of more flexibility and usability.

This toolbox offers functions to load data into a toolbox specific data structure and to preprocess (normalize) the data set. This is essential with regard to the absolute minimum distance in finding the BMU, because a feature within the range of e.g. \([10, 1000]\) has a bigger influence than a one in the range of e.g. \([0, 1]\).

Therefore the function can be used to normalize to one in the range of \([0, 1]\), apply softmax transform, histogram equalization and others. These operations are logged in the toolbox data structure and can be undone.

![Figure 5 Visualization of trained SOM by som_show; upper left: Unified distance matrix, blue means similar neighbors, red means dissimilar neighbors, clusters can be seen; others: visualized component planes of original pixels, average filter, entropy, standard deviation and range of pixel values.](image)

Then a SOM can be created, initialized and trained by and visualized, see fig. 5. A quality measure of SOM can be obtained, which computes the final quantization error \( q_e \) and the final topographic error \( t_e \).

![Figure 6. Overview of optimization process with evolutionary algorithm](image)

**Evolutionary Algorithms**

Evolutionary algorithms belong to the class of optimization search functions. With a statistical approach they mimic the natural biological evolution based on the principle of survival of the fittest to create individuals representing a better approximation of a solution in the problem domain. Because they work on populations of individuals instead of one single
solution, the optimization is performed in a parallel manner. At each generation, a population of individuals is created by a process of selecting individuals according to their fitness and breeding them together to a new population of better individuals using operators borrowed from natural genetics (e.g. mutation and recombination), see fig. 6, [28]. This makes the optimization approach very suitable for the optimization problem of present work. There are several sets of parameters which have influence on the quality of the SOM, but due to the high dimension of these parameters, an optimization by hand is not practicable. Genetic Algorithms have already been used for feature selection in MR image processing in the past [23]. Therefore it is suggested to use an evolutionary algorithm to optimize the following set of parameters which form the chromosome of each individual:
- Choice of features used
- Parameters of features
- Parameters of SOM

![Fig. 7 Schematic chromosome of an individual, containing all parameters needed to be optimized; b means binary and f floating point representation](image)

First a chromosome vector has to be built. The choice of features can be directly binary coded one gene for each available feature. Then all parameters of the feature extraction functions, for example the average filter, are inserted into chromosome. The average filter has one parameter, which defines the radius of the filter window. This can be an integer in the range of [3; 5; :: ; 11], which also can be binary coded. The Laplace filter needs for example a float in the range of [0::1]. Finally the parameters of the SOM (initialization (binary), training algorithm (binary), perhaps topology (binary) and map size (integer)) inserted in the end of the chromosome. Fig. 7 shows a schematic chromosome, where b means binary coded and f means floating point representation. Then a first population is generated. The above mentioned quality measures of the SOM are now the fitness functions applied to the population. If the optimization criteria are not met, best individuals are selected according to their fitness as parents to produce offspring with rules of mutation and recombination to a new population. The cycle begins again. To implement this evolutionary search algorithm in MATLAB the "Genetic and Evolutionary Algorithm Toolbox for use with Matlab" by [28], http://www.geatbx.com/ is used.

**Clustering**

In fig. 5, it can be easily seen that the neurons of the SOM are clustered into groups. Now it is a challenging task to identify the borders of the clusters, so that the neurons can be afterwards labeled as a specific cluster. Also for this class of tasks the toolbox offers several built-in functions. In this case k-means clustering algorithm was chosen: kmeans_clusters(sD, n_max). It allows specifying how many clusters should be searched for. Outputs are c_i (contains cluster centroids for k = i), p_i (contains cluster indices for k = i), err (squared sum of errors for each value of k) and ind (Davies-Bouldin index cluster separation measure for each clustering, a cluster validation algorithm, which calculates the
average similarity of a cluster with its most similar cluster).

With this index it can now be calculated (for different k) to which cluster the according BMU for every input pixel belongs, in other words: the input image pixel are labeled with the according cluster index. This can also be easily visualized:

To make this result comparable to other works, a simple measurement tool is to calculate the area of the respective matter. This can be done with MATLAB built-in function bwarea, which counts the foreground pixel in a binary image. To calculate the real world area in mm² this has to be multiplied with the ratio stored in DICOM file format Meta information about “Pixel Spacing” of the MRI. In this case it is for both dimensions 0.4297.

Fig. 8 (slice no: 75), CSF, WM, GM

Conclusion and discussion

Self-Organizing Maps (SOM) as a machine learning algorithm can be used in general to separate the three different brain matters CSF, WM and GM in an MR image of a human head using MATLAB and toolboxes as platform. Also it is discussed which image preprocessing steps are needed to prepare MR images (noise removal, skull stripping process and feature extraction) for presentation to a Kohonen-network.

It is shown that the described and implemented skull stripping algorithm (non brain tissue removal) works, but also disadvantages and weaknesses are discussed and the watershed algorithm introduced and proposed to be a promising algorithm extension delivering more robust results in skull stripping.

In the way it is currently implemented in this work SOM neural network and classification algorithm can be used for required purpose of brain tissue separation.

Figure 8. Davies-Bouldin index, which validates cluster separation for every k; in this case SOM with four features (original, average, entropy, standard deviation) and k = 2::15; if the SOM is clustered into 5 clusters the DB-index is minimal.

With regard to the experiences during this research work we would propose three different approaches how to improve the results of the applied algorithm significantly: As mentioned above the used skull stripping algorithm not only lacks robustness but also cuts away some CSF between skull and brain, which not only makes it more difficult to compare the results with other works but also affects brain matter transitions, which are learned by SOM. For this reason using watershed algorithm in skull stripping process would have double benefit. A little pool of basic features has been extracted and used to present the SOM. Experiments with Laplace-filter extracted features show, that the edge-detecting respectively texture sensitive filters deliver valuable information, as the output of the SOM changed obvious. This reveals the need for the development of some more accurate feature extractors, which are sensitive to
edges and texture attributes. Highest potential for the improvement of the algorithm as a whole promise the evolutionary algorithm-based feature selection, feature generating parameter and SOM parameter adjustment. The SOM training function som_make (respectively som_quality) returns two errors (final quantization and final topographic error), which indicates the quality of the resulting network. This quality measure function is used as a fitness function for evolutionary algorithm parameter optimization, where each individual has a chromosome, which consists of three sets of parameters: feature is used or not parameters of all features some parameters of the SOM. The evolutionary algorithm will evolve the chromosome in such a way, that the used fitness functions, respectively the quality function of the SOM, will be minimized.

Reference:
