Two fully-unsupervised methods for MR brain image segmentation using SOM-based strategies


A. Department of Communication Engineering, University of Malaga, Malaga, Spain
B. Department of Signal Theory, Networking and Communications, University of Granada, Granada, Spain
C. Virgen de las Nieves Hospital, Granada, Spain

Abstract

Image segmentation consists in partitioning an image into different regions. MRI image segmentation is especially interesting, since an accurate segmentation of the different brain tissues provides a way to identify many brain disorders such as dementia, schizophrenia or even the Alzheimer’s disease. A large variety of image segmentation approaches have been implemented before. Nevertheless, most of them use a priori knowledge about the voxel classification, which prevents figuring out other tissue classes different from the classes the system was trained for. This paper presents two unsupervised approaches for brain image segmentation. The first one is based on the use of relevant information extracted from the whole volume histogram which is processed by using self-organizing maps (SOM). This approach is faster and computationally more efficient than previously reported methods. The second method proposed consists of four stages including MRI brain image acquisition, first and second order feature extraction using overlapping windows, evolutionary computing-based feature selection and finally, map units are grouped by means of a novel SOM clustering algorithm. While the first method is a fast procedure for the segmentation of the whole volume and provides a way to model tissue classes, the second approach is a more robust scheme under noisy or bad intensity normalization conditions that provides better results using high resolution images, outperforming the results provided by other algorithms in the state-of-the-art, in terms of the average overlap metric. The proposed algorithms have been successfully evaluated using the IBSR and IBSR 2.0 databases, as well as high-resolution MR images from the Nuclear Medicine Department of the “Virgen de las Nieves” Hospital, Granada, Spain (VNH), providing in any case good segmentation results.

Article history:
Received 1 December 2010
Received in revised form 17 August 2012
Accepted 24 November 2012
Available online 6 December 2012

Keywords:
Image segmentation
MRI
Neural networks
Self-organizing maps
Feature extraction
Genetic algorithms

1. Introduction

Many current problems in image-guided surgery, therapy evaluation and diagnostic tools strongly benefit from the improvement on the medical imaging systems at reduced cost [1]. In this way, magnetic resonance imaging (MRI) has been widely used due to its excellent spatial resolution, tissue contrast and non-invasive character. Moreover, modern medical imaging systems [2] usually provide a vast amount of images to be analyzed. The study and evaluation of these images are usually developed through visual ratings performed by experts and other subjective procedures which are time-consuming and prone to error.

Generally, MRI images are qualitatively analyzed by experts based on their own experience and skills, but it is always limited by the human vision system which it is not able to distinguish among more than several tens of gray levels. However, as current MRI systems can provide images up to 65,535 gray levels, there is much more information contained in a MRI than the human vision is able to extract. This way, computer aided tools (CAD) play an important role for analyzing high resolution and high bit-depth MRI images, as they provide an important source of information for radiologists when diagnosing a disease or looking for a specific anomaly.

Segmentation of MR images consists in identifying the neuro-anatomical structures within medical images or “splitting an image into its constituent parts” [31]. Brain segmentation techniques, as a part of CAD systems, can be used to characterize neurological diseases, such as dementia, multiple sclerosis, schizophrenia and even the Alzheimer’s disease (AD) [3]. In the case of AD, there is no a well-known cause and it is very difficult to diagnose. With the improvements of MR imaging systems, the image processing techniques as well as the discovery of new biomarkers, neurological disorders such as AD are expected to be diagnosed even before the manifestation of any cognitive symptoms [57,58]. Thus, segmentation of brain MRI enables finding common patterns in AD...
patients such as hippocampal volume or cortical gray matter density reduction. Furthermore, these techniques could help to find other causes of brain disorders or anomalies. In fact, the segmentation algorithms presented in this paper are part of a larger study performed by the authors on the use of tissue distribution for the early diagnosis of AD [57,58].

The development of effective tools for grouping and recognizing different anatomical tissues structures and fluids, is a field of growing interest with the improvement of the medical imaging systems. These tools are usually trained to recognize the three basic tissue classes found on a brain MR image: white matter (WM), gray matter (GM) and cerebrospinal fluid (CSF). All of the non-recognized tissues or fluids are classified as suspect to be pathological. In the same way the human expert has to learn to recognize different regions on the MR image, the segmentation algorithms have to be trained.

There are a wide range of brain segmentation techniques. They can be classified into manual, semiautomatic and automatic techniques. Manual techniques are the most common and have been used for years. They require a human expert to select the voxels belonging to a specific object individually. In semiautomatic segmentation, the human expert is usually aided by image processing techniques.

The most common image processing techniques used for semiautomatic segmentation are histogram-based techniques [8,9,16,22], statistical classifiers, fuzzy classifiers, support vector machine (SVM) classifiers, and neural network-based classifiers. Histogram-based techniques are based on thresholding, which consists in determining the thresholds of the voxel intensity values in order to separate the voxels belonging to each class. This requires a previous training process of the system using expert segmentation images. Segmentation techniques based on histogram thresholding use the fact that the peaks on histogram can belong to a specific tissue [35]. Thus, the problem is reduced to classifying and modeling the peaks and valleys on the histogram. There are other histogram-based segmentation techniques which also take into account the relative position of the peaks or other statistics calculated from the histogram [36,37]. Nevertheless, the histogram thresholding segmentation approaches usually do not take into account the spatial information contained on a MR image. On the other hand, spatial information is essential since anatomical regions of the brain [18] are more likely to accommodate a given tissue. As a result, different MR images could have similar histograms and then, similar thresholds.

Other segmentation approaches are based on contour detection techniques [10,11], using the boundaries among different tissues for segmentation. Edge detection algorithms such as Sobel, Prewitt, Laplacian or Canny filters [31] select the border voxels among different objects. These filters generally perform the preprocessing for active contour algorithms [12].

In region-based techniques [13] once a voxel is marked, the algorithm starts to add more voxels surrounding it, preserving some properties such as homogeneity or intensity level. Other algorithms search for voxels belonging to the initial class following a specific geometrical model [38].

Statistical classifiers use some previous learned rules to perform the grouping. These are called clustering techniques which classify voxels in an unsupervised manner, since they group similar voxels into the same class. Thus, a similarity criterion has to be established or learned in order to determine whether a voxel belongs to a given class. Then the classifier [4] will generate different classes which contain group of voxels with the same properties. Some of the statistical classifiers are based on the expectation-maximization algorithms (EM) [1,14,15], maximum likelihood (ML) estimation [16] or Markov random fields [13,17]. K-means and its variants such as Fuzzy k-means are widely used as they avoid abrupt transitions in the classification process [19].

Support vector classifiers [40] are a new type of classifiers based on statistical learning theory which have been successfully applied to image segmentation [20] due to its generalization ability. Other segmentation techniques are based on artificial neural network classifiers [21–27], such as self-organizing maps (SOM) [23–26,28].

As mentioned before, segmentation of MR images can be seen as a pattern classification and recognition problem. Thus, a pre-processing stage is necessary in order to make the segmentation more effective as well as a post-processing stage for ensuring the voxel grouping algorithm is performed correctly. On the other hand, all the above segmentation methods use some a priori knowledge from reference images [5].

Several fully-automated segmentation methods have been proposed, but most of them also use reference images for training [1,5–7]. In [5], an automatic segmentation framework which works in three steps is presented. It uses a combination of different techniques: (i) skull-stripping, (ii) intensity inhomogeneity correction and, finally, (iii) classification of the brain tissue by means of a Fuzzy Kohonen’s Competitive Learning algorithm (F-KCL). In [6] a two-step algorithm is presented. In the first step, the noise is removed and in the second step, an unsupervised image segmentation method based on fuzzy C-means clustering algorithm is applied to the MR image in order to partition it into distinct regions.

1.1. Summary and organization

In this work, we present two different MR image segmentation methods. The first one uses information from the volume image histogram to compose feature vectors to be classified by a SOM, referred as HFS-SOM method in the following. Then, SOM prototypes are clustered by a k-means algorithm. In order to compute the optimum k value for the best clustering performance, the Davies–Boulding index (DBI) [47] is computed for several trials. Thus, the SOM prototypes are grouped into k clusters providing the lower DBI, and each of these clusters corresponds to a different tissue in the image.

The second method splits the acquired images into overlapping windows, and computes first and second order statistical features from each window. A feature selection process is performed by means of multi-objective optimization using a genetic algorithm [32,55] in order to select the most discriminative set of features. The selected features compose the feature vectors used as inputs to train the SOM. During the training stage, the SOM projects the input vectors into a two dimensional space and computes a number of prototypes. These prototypes are a generalization of the input space in a lower number of vectors, meaning the quantization the input space. SOM is a clustering algorithm itself, and it considers each SOM unit as a cluster [33]. Nevertheless, similar units have to be grouped as voxels belonging to the same tissue can be represented by similar prototypes [33]. This way, the SOM prototypes have to be clustered and the borders between clusters have to be defined. The computation of these borders can be addressed by a generic clustering algorithm such as k-means, or by a specific SOM clustering algorithm [46]. In this work, a specific algorithm (i.e. EG-SOM) is devised to improve the segmentation performance which uses the accumulated entropy to cluster the SOM units.

The methods described in this paper do not use any a priori knowledge about the voxel classification, and result in fully-unsupervised methods for MRI image segmentation. In addition, it is not necessary to indicate the number of tissue classes to be found, as the algorithms compute this number maximizing the goodness of the overall clustering process.

The paper is organized as follows: Section 2 presents the materials and methods used in this work. It is divided into four subsections; Section 2.1 describes the image databases used in this work; Section 2.2 shows the pre-processing stage which is common
to the two segmentation approaches; Section 2.3 presents a faster implementation of the method which uses information extracted from the image histogram for segmentation of the whole volume and; Section 2.4 shows a high resolution approach for image slices. Section 3 depicts the experimental results obtained from the evaluation of the proposed methods using the IBR dataset and discusses the main questions derived from them. Finally, conclusions are drawn in Section 4.

2. Materials and methods

This section consists of three subsections which summarize the segmentation methods and the image databases used in this work to evaluate the proposed methods, which include manual segmentation references considered as the ground truth.

2.1. Databases

In order to check the performance of our image segmentation approach in comparison with other existing methods, manual segmentation labeling of the processed databases is required. Internet Brain Segmentation Repository (IBSR) from the Massachusetts General Hospital [25] is suitable for this purpose, as it provides manually guided expert segmentation results along with a set of magnetic resonance images. Thus, IBSR 1.0 provides 20 T1-weighted volumetric images and IBSR 2.0 set provides 18 T1-weighted volumetric images that have been spatially normalized and processed by the Center for Morphometric Analysis (CMA) at the Massachusetts General Hospital with the biasfield correction routines already applied. On the other hand, the overlap comparison metric is provided for each volume for comparing different segmentation methods. Consequently, images from the IBSR 1.0 database were used to compute the average overlap metric in order to compare to other previously proposed algorithms. In addition, images from IBSR 2.0 are also used to assess the performance of our algorithm. Furthermore, an image set consisting of high resolution MR images from the Nuclear Medicine Service of the ‘Virgen de las Nieves’ Hospital, Granada, Spain (VNH) are also used in order to evaluate the proposed segmentation algorithms.

2.2. Image preprocessing

Once the MR image has been acquired, a pre-processing is performed in order to remove noise and clean-up the image background. Brain tissue extraction from undesired structures removal (i.e. skull and scalp) can be done at this stage. Several algorithms have been developed for this purpose such as Brain Surface Extractor (BSE), Brain Extraction tool (BET) [56], Minneapolis Consensus Strip (McStrip) or Hybrid Watershed Algorithm (HWA) [5]. These structures are already removed from the IBSR 1.0 database. Nevertheless images provided by IBSR 2.0 are distributed without the scalp/skull already removed. In the latter database, the brain has been extracted in the pre-processing stage using BET.

In order to remove background noise, we use a binary mask, built by detecting the greatest contiguous object in the image. After multiplying the binary mask (which contains 0 at the background voxels and 1 otherwise) by the original image, we get the background in black. Moreover, the image is centered in order to avoid losing voxels when using the sliding window technique as further described below in this Section (see Fig. 1).

2.3. Fast volume image segmentation by HFS-SOM

Fig. 1 summarizes the two segmentation methods proposed in this paper: (i) the fast volume segmentation algorithm (HFS-SOM) and (ii) the EGS-SOM algorithm (EG-SOM). Table 1 summarizes the main characteristics of the proposed algorithms, regarding the type of image features, the learning paradigm (i.e. supervised or unsupervised), and the feature selection method. In this section, the HFS-SOM segmentation method which uses statistical features extracted from the volumetric image histogram is presented.

2.3.1. A histogram computation

Firstly, the volume image histogram is computed, which describes the probability of occurrence of voxel intensities and provides information regarding different tissues. A common approach to avoid processing the large number of voxels in a MR consists in modeling the intensity values as a finite number of prototypes, improving the computational effectiveness. This is addressed in [43–45], where the voxel intensities are modeled by a mixture of Gaussian distributions [43,44] or α-stable distributions [45]. In this work, we use a SOM to model intensity values by a finite number of prototypes corresponding to the number of map units, as described in Section 2.3.2. Fig. 2a shows the rendered brain surface from the IBSR_12 volume using SPMM [41] and Fig. 2b shows the computed histogram for the whole volume. The probability of each bin is computed as the frequency of occurrence of that intensity in the volume image divided by the total number of different intensities present on the image. Finally, the 1st bin is removed from the histogram since it contains all the background voxels. Thus, only information corresponding to the brain is stored.

Histogram data including the intensity occurrence probabilities (ρ), the relative position regarding the intensity value (bin number, bi), the mean of the probability values over a 3-bins window centered on the bin i (mi) and the variance of that window (σ2) are used to compose the feature vectors F = (ρ, mi, σ2, bi), p, m, σ2 ∈ R, bi ∈ Z, which are the inputs of the SOM.

2.3.2. SOM modeling

In this work, voxel intensities are modeled by the SOM prototypes using the information contained in the histogram instead of using mixtures of probability density functions. This aims to model the peaks and valleys of the image histogram as they retain discriminative information for voxel classification.

Once the volume image histogram has been computed and the feature space has been composed as shown in the preceding section, these vectors are used as inputs for training a SOM [33] with a 2D hexagonal grid since it fitted better the feature space as shown in Section 3.

The SOM algorithm can be summarized as follows. Let X ∈ R,d the data manifold. The winning unit is computed in each iteration according to:

\[ U_{\omega}(t) = \text{argmin}_{\omega} ||x(t) - o_{\omega}(t)|| \]  

where \( x(t), x \in X \), is the input vector at time t and \( o_{\omega}(t) \) is the prototype vector associated to the unit \( \omega \). The unit closer to the input vector \( U_{\omega}(t) \) is referred as winning unit and the associated prototype is updated. To complete the adaptive learning process on the SOM, the prototypes of the units in the neighborhood of the winning unit are also updated according to:

\[ o_{\omega}(t + 1) = o_{\omega}(t) + \alpha(t) h_{\Omega}(t)(x(t) - o_{\omega}(t)) \]  

where \( \alpha(t) \) is the exponential decay learning factor and \( h_{\Omega}(t) \) is the neighborhood function associated to the unit \( \omega \). Both, the learning factor and the neighborhood function diminish with time and the prototypes adaptation process becomes slower as the neighborhood of the unit \( \omega \) contains less number of units.

\[ h_{\Omega}(t) = e^{\left(-\frac{(m_{\Omega}(t) - m_{\Omega}(t-1))^2}{2\sigma_{\Omega}^2}\right)} \]
Eq. (3) shows the neighborhood function, where $r_i$ represents the position on the output space and $\| r_i - r_j \|$ is the distance between the winning unit and the unit $i$ on the output space. The neighborhood is defined by a Gaussian function which shrinks in each iteration as shown in Eq. (4). In this competitive process, the winning unit is named the Best Matching Unit (BMU). On the other hand, $\sigma(t)$ controls the reduction of the Gaussian neighborhood in each iteration according to a time constant $\tau_1$.

$$
\sigma(t) = \sigma_0 e^{-t/\tau_1}
$$

In the SOM, each prototype $o_j$ is the centroid of its Voronoi polyhedron. In other words, SOM projects the prototypes into a two or three dimensional space (depending on the dimension of the output layer) in such a way that the most similar the prototypes are located closer in the output space (in terms of the Euclidean distance). This way, the prototypes and their location on the output space are a valuable source of information which can be used to cluster the SOM [45].

Initialization of the SOM prototypes is performed linearly, taking into account the eigenvalues and eigenvectors of the training data. The prototypes are arranged into hexagonal lattices (corresponding to a width that is proportional to the standard deviation of the first principal component) [33,34]. This initialization method implies that the first dimension of the prototypes is arranged proportionally to the first principal component and the second dimension is arranged proportionally to the second principal component. Thus, once the optimal number of map units has been estimated, the network is trained linearly. All the above calculations are performed during the map initialization process.

After the map is initialized, it is trained using the feature vectors of the reduced feature space normalized for zero mean and unity variance. This normalization procedure is also known as data whitening. The normalization of the vectors used for training the map avoids one dimension to have more influence than others on the training process due to the different nature of the extracted features. As a result of the training process, a set of prototypes grouped on the SOM layer models the features of the volume image histogram. Indeed, a classification process is accomplished by computing the closest prototype for each voxel.

The quality of the map determines the representation of the data by means of the prototypes computed during training. Then, it is important to measure the goodness of the trained map in order to determine (i) the distance between the prototypes and the data manifold (i.e. the prototypes generalizes the input data) and (ii) the distance among similar prototypes in the output space (i.e. topology preservation). The quality of the trained map can be evaluated by means of two measures. These two measures are the quantization error $(e_q)$, which determines the average distance between each data vector and its Best Matching Unit (BMU) and the topological error $(q_e)$, which measures the proportion of data vectors

![Fig. 1. Block diagram of the segmentation method process.](image1)

![Fig. 2. Rendered brain surface extracted from IBSR_12 volume (a) and computed histogram (b).](image2)
for which first and second BMUs are not adjacent units. Both, the quantization error and the topological error are defined by Eqs. (5) and (6) respectively.

\[ L_e = \frac{1}{N} \sum_{i=1}^{N} u(\hat{x}_i) \]  
\[ q_e = \frac{1}{N} \sum_{i=1}^{N} ||\mathbf{x}_i - \hat{b}_{\mathbf{x}_i}|| \]  

In Eq. (5), \( N \) is the total number of data vectors, \( u(\hat{x}_i) \) is 1 if the first and the second BMU for \( \hat{x}_i \) are non-adjacent and 0 otherwise. In Eq. (6) the quantization error is defined where \( \mathbf{x}_i \) is the \( i \)-th data vector on the input space and \( \hat{b}_{\mathbf{x}_i} \) is the weight (prototype) associated to the best matching unit for the data vector \( \mathbf{x}_i \). Therefore, lower values of \( L_e \) and \( q_e \) implies better data representation and topology preservation, which is equivalent to better clustering result. That is to say, the lower values on the quantization error \( (q_e) \) and the topological error \( (L_e) \) the better the goodness of the SOM [33,42].

2.3.3. SOM clustering and voxel classification

The output layer of a trained SOM is composed by a reduced number of prototypes (the number of units on the output layer) which model the input data manifold. Although SOM is a clustering algorithm itself, it considers each unit as a single cluster. However, similar units represent similar data and must be clustered as belonging to a group. Thus, it is necessary to cluster the SOM prototypes in order to define the borders between clusters. This way, \( k \)-means algorithm is used to cluster the SOM, grouping the prototypes into \( k \) different classes.

The DBI [47], which gives lower values for better clustering results, is computed for different \( k \) values to provide a measurement of the clustering validity. Fig. 3a shows the similarity graph for the SOM units. In this figure, the most similar units have similar colors and the number of activations of each unit is represented with different sizes (the higher the number of activations, the greater the size of the unit). This way, cluster borders can be visually identified as the smaller units. In addition, Fig. 3b shows the clustering result after computing the clusters with the \( k \)-means algorithm for 3 clusters (\( k = 3 \)).

DBI is used to determine the \( k \) value for the \( k \)-means algorithm. An image containing only the three basic tissues (WM, GM and CSF) should provide the lowest DBI for \( k = 3 \). More than 3 classes are expected to be found in the image when a lower value of the validity index is given for \( k > 3 \). Nevertheless, the \( k \)-value can be manually selected as the number of expected tissues.

The clusters on the SOM group the units so that they belong to a specific class. As each of these units will be the BMU of a specific set of voxels, the clusters define different voxel classes. This way, each voxel is labeled as belonging to a class (i.e. a tissue or fluid present on the MRI). Fig. 4 shows several slices of the segmented IBSR_12 volume following the described method.

2.4. ECS-SOM segmentation

The aim of this method is to achieve higher resolution images by processing individual slices from a volume. The segmentation process is shown in Fig. 1, where the initial pre-processing stage is the same as the one described in Section 2.2. The method described in this section is also based on SOM for voxel classification, but histogram information from the image volume is replaced by computing a set of discriminative features.

2.4.1. Feature extraction

At this stage, some significant features from the MR image are extracted to be subjected to selection and classification. The main purpose of the feature extraction is to reduce the original data set by calculating some properties that can be used to classify and to recognize patterns included in the input images. As a result, a feature vector which dimension is equal to the number of extracted features is obtained. These features should retain the essential information and should be different enough among classes for a good classification performance. Moreover, the features used to classify the voxels on each image may depend on the specific image and, consequently, the feature extraction process plays a decisive role in the classification performance and thus, in the overall segmentation process.

The statistical features used in this work are classified into first order and second order features. First order features are derived from the gray level of a specific voxel and its neighborhood and the second order features are derived from the spatial relationship among different voxels.

In order to extract features from the image, a sliding and overlapped window of size \( a_x \times a_y \) is moved along the image. As this window is moved voxel-wise, we obtain as many windows as voxels contained in the image. Then, a feature set is computed from each window and referred to the central voxel. The feature set extracted from each window describes each voxel in the image by a different feature vector. Thus, the feature vectors belong to a \( D \)-dimensional feature space, where \( D \) is the number of extracted features. This procedure is slower than the one that uses the mean of the features for each window, however image resolution is preserved. This way, the original image is converted into a set of feature vectors \( f = [f_1, f_2, ..., f_D] \) belonging to the feature space \( F \in \mathbb{R}^D \).

Let us assume we have an image of dimension \( M \times N \) voxels, where the intensity is a function \( i(x,y) \) being \( x,y \) space variables. On the other hand, the function \( i(x,y) \) can only take discrete values \( [1,G] \), where \( G = \max(i(x,y)) \) is the maximum intensity level in the image \( i \) after quantization. Let also assume we split the image into overlapped windows of size \( a_x \times a_y \). Then we define the following first order features:

\[ I = i(x, y) \]  
\[ \mu = \frac{1}{a_x a_y} \sum_{x=1}^{a_x} \sum_{y=1}^{a_y} i(x, y) \]  
\[ \sigma^2 = \frac{1}{(a_x a_y - 1)} \sum_{x=1}^{a_x} \sum_{y=1}^{a_y} (i(x, y) - \mu)^2 \]  

However, these features do not take into account the spatial dependence among the voxels in a window. Therefore, they do not provide textural information. Spatial relationship among voxels can be taken in to account using:
a) **Textural features.** They can be computed using different methods such as Gray-Level Co-occurrence Matrix (GLCM) or through sum and difference histograms of each window. Haralick et al. [29] proposed the use of 14 features for image classification, computed using the GLCM structure. The most significant features depend on the specific image being analyzed. Thus, we computed all the textural features proposed in [29,30] and, afterwards, the most significant and most discriminative features will be selected. The GLCM structure represents the distribution of co-occurring gray values at a given offset. For an image im, the GLCM can be mathematically defined as:

\[
p_{\Delta x, \Delta y}(i,j) = \sum_{p_{1}=0}^{n} \sum_{p_{2}=0}^{n} \begin{cases} 
1 & \text{if } \text{Im}(p_{1}, p_{2}) = i \text{ and } \text{Im}(p_{1} + \Delta x, p_{2} + \Delta y) = j \\
0 & \text{otherwise}
\end{cases}
\]

At the same time, \(p_{\Delta}(i)\) is the \(i\)-th entry in the marginal probability matrix. This matrix is computed by means of \(\sum_{j=1}^{256} p(i,j)\) [31] (see Appendix A for further details).

The set of second order features we have used are energy, entropy, contrast, angular second moment (ASM), mean, sum average, autocorrelation, correlation, inverse difference moment, maximum probability, cluster prominence, cluster shade, dissimilarity and variance. The mathematical formulation for these features is detailed in the Appendix at the end of this paper.

b) **Moment invariants.** In [29,30], a set of parameters invariant under scaling and rotation are introduced for image classification and recognition. In our case, the moment invariants computed from each window centered on each voxel are used to classify different tissues on the MR image. Thus, we have included seven moment invariants defined in [29]. In the same way that first and second orders explained above have to be selected, the most discriminative moment invariants will be selected.

Once the above features have been computed, we construct the feature vector. This feature set is contained in \(\mathbb{R}^9\) and consists of vectors of dimension \(D\), where \(D\) is the number of extracted features. In our case, \(D=24\), and therefore, the feature space is contained in \(\mathbb{R}^{24}\).

### 2.4.2. Feature selection

The original set of features calculated on the previous step, has to be reduced in order to select the most discriminative ones for the set of images considered. In this work, the dimension of the feature space is reduced using evolutionary computation by means of a genetic algorithm (GA).

The GA used for feature selection tries to minimize the quantization and topological errors of the SOM described in Section 2.3.2, as lower values of quantization and topological errors tend to optimize the goodness of the SOM (i.e. a better quality of the clustering process). On the other hand, we have noticed that less number of features usually provides lower error values but keeping first order, second order and moment invariant features at the same time provides better segmentation results. Therefore, we used a penalty function to prioritize those solutions which have a minimum of 5 features and at least one per type (first order, second order or moment invariants), imposing a constraint in the optimization process.

The feature selection process requires training a SOM in each iteration as the evaluation of the fitness function uses the quantization error and topological error. This training process is accomplished by a batch training algorithm which the data set is presented to the SOM as a whole, and vector weight updating is done simply by replacing the prototype vectors with a weighted average over the samples. Thus, new weight vectors are weighted averages of the data vectors [33,34]. This method, known as batch training algorithm is considerably faster than the sequential training algorithm [33].

In the implementation of the feature selection algorithm, the population is encoded using a binary vector which acts as a feature mask in order to select at least five features (at least, 5 bits from the 24 available in the vector should be set to 1).

As indicated in [32] GAs should perform better for moderate population sizes, although it depends on the specific application. The selection algorithm used consists of a modified version of the stochastic uniform selection [31] which provides a uniform distribution of the population. The crossover is implemented through scattered crossover technique [32], and the mutation operator uses a Gaussian function for \(p\)-distribution which shrinks in each iteration.

The fitness function shown in Eq. (10) uses equally weighted measurements of the quantization error \(q_e\) and the topological error \(t_e\). This method, consisting in assigning weights to the functions which can be minimized at the same time is usual in multiobjective optimization problems [32] with compatible objectives. Then, the function to be minimized is:

\[
F_{QT} = 0.5 \times q_e + 0.5 \times t_e \tag{10}
\]

and the penalty function modifies the fitness value according to:

\[
F_p = -F_{QT} + \frac{F_{QT}}{p \times n} \tag{11}
\]

where \(p\) is the penalty factor and \(n\) is the number of bits on the mask string.

The feature selection process is summarized in Fig. 5. As indicated in this figure, the SOM is trained using the initial population and then, the map quality is computed. This initial population is evolved and the SOM is initialized and trained again (i.e. a new GA iteration) until the stop conditions are reached: the maximum number of allowed generations is reached, or the algorithm has converged (i.e. the map quality is no longer improved, and GA converges to a stable value for the fitness function). The algorithm converges in less than 200 generations using the proposed fitness function. Indeed, after 200 generations, the solution proposed by the GA is the same during a number of trials. Moreover, the error in the fitness function for 50 runs is always below 3%, and
the optimization mechanism is robust enough to guarantee the repeatability of the optimized feature sets.

Fig. 6 presents the evolution of the fitness function averaged over 50 runs, showing that GA is able to escape from local minima.

2.4.3. EGS-SOM clustering

Once the dimension of the feature space has been reduced, we use the vectors of this space for training a SOM. A hexagonal lattice is used on the SOM layer and the map size is determined as explained in Section 2.3.

As in the HFS-SOM algorithm the feature vectors are normalized to zero mean and unit variance [33,34]. The vectors are selected with a random permutation distribution in order to weight up the selection of background vectors.

The training stage performs a classification of the feature vectors, providing a map in which different units have been associated to the nearest feature vector. Thus, different regions on the map group vectors from the feature space with similar features. At this point, the voxels whose vectors are associated to each map region could be shown together in an image as we have as many tissues as different regions found during the classification stage. Nevertheless, it would mix different tissues into one image due to the similarity among voxels belonging to different partitions. In other words, the clusters on the map have to be redefined in order to figure out a clear border among different segments. This way, we present results based on two approaches. On the one hand, a 2-neighbor clustering approach is used. This approach consists of three steps for SOM clustering calculation. In the first step, a hit histogram is calculated in order to remove the map units with more hits. As background voxels are included in the classification process, these units correspond to the background. Then, the 2-neighbors to the other map units are calculated and the voxels associated are assigned to an image segment as shown in Fig. 7.

Other procedure devised in this work in order to improve the resolution of the clustering method consists in using the feature vectors associated to each prototype to compute a similarity measurement among the vectors belonging to the rest of prototypes. Next, the prototypes are sorted in ascending order of the contrast, and finally, the feature vectors associated to each prototype are included in a cluster. Each time a new group of voxels belonging to each prototype is added to a cluster, the entropy is calculated. This process is repeated until a threshold on the entropy is reached, dealing with the lowest DBI, as it maximizes the quality of the clustering process. Therefore, all voxels belonging to the feature vectors included on a cluster form an image segment. This procedure calculates the direction of the EGS-SOM from each map unit and the clustering is delimited using the opposite direction.

\[
H_{in}(x, y) = - \sum_{i=1}^{W_x} \sum_{j=1}^{W_y} p_{i,j}(x,y) \log(p_{i,j}(x,y))
\]  

(12)

In Eq. (12), \( \theta \) and \( d \) are the direction and the distance respectively for calculating the GLCM [29], and \((x,y)\) are the coordinates in a image plane. The mean entropy computed for all the directions is called \( H_m \). Thus,

\[
H_m(x, y) = \sum_{k=1}^{4} H_{\theta_k,d}(x, y)
\]

(13)

where \( \theta_1 = 0^\circ, \theta_2 = 45^\circ, \theta_3 = 90^\circ, \theta_4 = 135^\circ \), and \( d = 1 \).
For each map unit, we compute the accumulated entropy,

$$H_m = \sum_{n=1}^{N_p} H_m$$  \hspace{1cm} (14)

being \( i \) the map unit index and \( N_p \) the number of voxels belonging to the map unit in the classification process. This means that the unit \( i \) has \( N_p \) voxels.

Since the map is a two-dimensional space, we calculate the entropy gradient from each map unit. Thus, once the entropy for a map unit is calculated, we move to the opposite direction of the EGS-SOM vector.

$$\nabla H = \frac{dH_m}{dx} \hat{x} + \frac{dH_m}{dy} \hat{y}$$  \hspace{1cm} (15)

In Fig. 8, the block diagram of the EGS-SOM clustering algorithm is shown.

The sub-images referred in Fig. 8 are composed by the voxels belonging to each SOM class as shown in Fig. 9.

In Fig. 10, the entropy gradient as well as the trajectory of minimum increase of entropy is shown. The process stops when the threshold is reached (red marked map unit).

Finally, the units are labeled with the class of the cluster calculated on the previous step.

3. Results and discussion

In this section we show the segmentation results obtained using real MR brain images from two different databases. One of these databases is the IBSR database [35] in two versions, IBSR and IBSR 2.0. Expert manual segmentations are available in both cases, but there are differences between these two sets of images. Details on the images used in our experiments and segmentation methods setup are shown in the following subsection.

3.1. Experimental setup

T1-weighted images with different resolutions have been used to test the proposed algorithms. Actually, we have used images with a resolution of \( 512 \times 512 \times 512, 256 \times 63 \times 256 \) and \( 256 \times 128 \times 256 \) voxels. For instance, in the case of \( 256 \times 63 \times 256 \) voxels images, the feature space of the axial plane is composed by \( 256 \times 63 \times 16 = 16,128 \) feature vectors on the axial plane. This way, images from IBSR 1.0 have a resolution of \( 256 \times 63 \times 256 \) voxels and scalp and skull are already extracted. However, they are not normalized and contain intensity inhomogeneities at different levels (depending on the image). On the other hand, images from IBSR 2.0 have a resolution of \( 256 \times 128 \times 256 \) voxels and are spatially and...
intensity normalized, but scalp and skull are not already removed from the image. This way, it is necessary to use the BET tool to extract the brain and to remove undesirable structures. We also used images with a higher resolution (512 × 512 × 512 voxels) provided by VNH. These images contain the scalp and skull, and we used them as they come to test our algorithms.

As in any SOM-based system, the classification performance depends on the map size. This way, the number of units on the map is usually set a priori in order to avoid map under-sizing. In fact, it is not recommended to use a map with a number of units much less than the training samples [33,34]. An initial estimation of the number of map units is given by the formula

\[ \text{map units} = 5 \times d^{1/2} \]

where \( d \) is the number of training samples. The ratio between the two dimensions of the map size is calculated as the ratio between the two largest eigenvalues of the auto-correlation matrix of the training data [34]. Another important issue which determines the performance of the SOM is the initialization of the map. In this work linear initialization of the SOM weights is used [34]. This is addressed by computing the eigenvectors and eigenvalues of the training data, since

the orientation of the eigenvectors corresponding to the two largest eigenvalues provides the directions in which the training data exhibits the most variance. Nevertheless, the map size has to be experimentally fine-tuned in order to improve the performance of the classification algorithm. The optimal map size for the analyzed images was 8 × 8 units for both, EGS-SOM and HFS-SOM segmentation methods and the SOM was trained using 5000 iterations.

In the case of the EGS-SOM method, a sliding window of 7 × 7 voxels is used as it has been determined to be larger enough to capture textural features without losing resolution (i.e. while larger window sizes tend to loose resolution, smaller ones lose textural information).

Regarding the feature selection process involved in the EGS-SOM method, the GA set-up parameters are shown in Table 2.

<table>
<thead>
<tr>
<th>GA parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Encoding type</td>
<td>Binary vector</td>
</tr>
<tr>
<td>Population size</td>
<td>30</td>
</tr>
<tr>
<td>Crossover probability</td>
<td>0.8</td>
</tr>
<tr>
<td>Mutation probability</td>
<td>0.03</td>
</tr>
<tr>
<td>Selection algorithm</td>
<td>Modified stochastic uniform selection [32,55]</td>
</tr>
<tr>
<td>Crossover algorithm</td>
<td>Scattered crossover [32,55]</td>
</tr>
<tr>
<td>Mutation algorithm</td>
<td>Gaussian with p-distribution [32,55]</td>
</tr>
</tbody>
</table>

Fig. 11 depicts the image formation from the subimages by adding the voxels associated to each prototype, when the EGS-SOM algorithm is applied to a high resolution MR image from VNH. In addition, the resulting subimages when voxels corresponding to each unit on the cluster are added to the tissue, are shown from top to bottom and left to right. Thus, top most left image corresponds to the less entropy when only a subimage has been added and bottom most right corresponds to the accumulated entropy threshold when subimages 1, 34, 39, 49, 19, 3, 16 and 35 have been added to the tissue as indicated in Fig. 10. This way, Fig. 12 shows the binarized version of Fig. 11 when subimage 35 is added. This corresponds to WM.

3.2. Experimental results and discussion

Fig. 13a and b shows the segmentation results for the IBSR volume 100_23 using the HFS-SOM algorithm and the EGS-SOM
algorithm, respectively. In these images, WM, GM and CSF are shown for slices 120, 130, 140, 150, 160 and 170 on the axial plane. Expert segmentation from IBSR database is shown in Fig. 13c.

Visual comparison between automatic segmentation and the ground truth points up that the EGS-SOM method outperforms the fast volume segmentation method. This fact is also stated in Fig. 15 and Table 3 where the Tanimoto’s index is shown for different segmentation algorithms. However, while HFS-SOM performs segmentation of the whole volume, EGS-SOM works slice-by-slice. This way, although EGS-SOM provides a higher resolution, HFS-SOM is a faster segmentation method which outperforms parametric or supervised methods such as MAP based methods.

The performance of the presented segmentation techniques have been evaluated by computing the average overlap rate through the Tanimoto’s index, as it has been widely used by other authors to compare the segmentation performance of their proposals [35,48–50,17,51–54]. Tanimoto’s index can be defined as:

\[ T(S_1, S_2) = \frac{|S_1 \cap S_2|}{|S_1 \cup S_2|} \]  

where \( S_1 \) is the segmentation set and \( S_2 \) is the ground truth.

Fig. 14a shows the segmentation results for the IBSR 2.0 volume 12 using the HFS-SOM algorithm, where each row corresponds to a tissue and each image column corresponds to a different slice. In the same way, Fig. 14b shows the same slices of Fig. 13b but the segmentation is performed using the EGS-SOM method. Fig. 14c shows the segmentation performed by expert radiologists provided by the IBSR database (ground truth).

It is important to highlight that expert segmentations of IBSR database only include internal CSF spaces. Nevertheless, our methods also figure out sulcal CSF. This way, Tanimoto’s index for CSF are lower than for WM or GM, as can be seen in Table 3, since the CSF segments computed by our algorithms may be considerably larger than the ones found on expert segmentations since the IBSR ground

---

**Table 3**

Mean and standard deviation of the Tanimoto’s performance index for the segmentation methods on Fig. 14.

<table>
<thead>
<tr>
<th>Segmentation algorithm</th>
<th>Ref.</th>
<th>WM index</th>
<th>GM index</th>
<th>CSF index</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manual (4 brains averaged over 2 experts)</td>
<td>[35]</td>
<td>0.832 ± *</td>
<td>0.876 ± *</td>
<td>-</td>
</tr>
<tr>
<td>EGS-SOM</td>
<td></td>
<td>0.70 ± 0.04</td>
<td>0.70 ± 0.04</td>
<td>0.22 ± 0.08</td>
</tr>
<tr>
<td>Constrained GMM (CGMM)</td>
<td>[50]</td>
<td>0.68 ± 0.04</td>
<td>0.66 ± 0.06</td>
<td>0.20 ± 0.06</td>
</tr>
<tr>
<td>MPM-MAP</td>
<td>[17]</td>
<td>0.66 ± 0.10</td>
<td>0.66 ± 0.10</td>
<td>-</td>
</tr>
<tr>
<td>HFS-SOM</td>
<td></td>
<td>0.60 ± 0.1</td>
<td>0.60 ± 0.15</td>
<td>0.1 ± 0.05</td>
</tr>
<tr>
<td>Adaptive MAP (amap)</td>
<td>[35,51]</td>
<td>0.57 ± 0.13</td>
<td>0.58 ± 0.17</td>
<td>0.07 ± 0.03</td>
</tr>
<tr>
<td>Biased MAP (bmap)</td>
<td>[35,51]</td>
<td>0.56 ± 0.17</td>
<td>0.58 ± 0.21</td>
<td>0.07 ± 0.03</td>
</tr>
<tr>
<td>Maximum a posteriori probability (map)</td>
<td>[35,52]</td>
<td>0.47 ± 0.11</td>
<td>0.57 ± 0.20</td>
<td>0.07 ± 0.03</td>
</tr>
<tr>
<td>Tree-structure k-means (tskmeans)</td>
<td>[35,53]</td>
<td>0.48 ± 0.12</td>
<td>0.58 ± 0.19</td>
<td>0.05 ± 0.02</td>
</tr>
<tr>
<td>Maximum likelihood (mlc)</td>
<td>[35,54]</td>
<td>0.54 ± 0.16</td>
<td>0.57 ± 0.20</td>
<td>0.06 ± 0.03</td>
</tr>
</tbody>
</table>

* Data not available in the source.
Fig. 14. Segmentation of the IBSR 2.0 volume 12, using the HFS-SOM algorithm (a) and the EGS-SOM algorithm (b). Ground Truth is shown in (c). Slices 100, 110, 120, 130, 140 and 150 on the axial plane are shown on each column. First column corresponds to WM, second column to GM and third column to CSF.

Fig. 15. Tanimoto performance index calculated through images from the IBSR database. Tanimoto’s index for WM (a) and GM (b) are shown for different segmentation algorithms.

In Table 3, a quantitative comparison among different segmentation methods through the Tanimoto’s index of the IBSR 1.0 database. In this table, while values of 1.0 means that the results are very similar, values are near 0.0 when they share no similarly. Standard deviation for each method is included in order to provide the statistical deviation of the index, as it has been calculated over all the images on the IBSR 1.0 database.

Fig. 15a shows the mean Tanimoto’s index calculated over the IBSR 1.0 images for different segmentation algorithms, and Fig. 15b presents the Tanimoto’s index over the IBSR 2.0 images for the two segmentation algorithms presented in this paper.

Table 4 shows the Tanimoto’s index averaged over the images on the IBSR 2.0 database. Segmentation methods such as Fuzzy

Table 4
Mean and standard deviation of the Tanimoto’s performance index for the segmentation methods on Fig. 15.

<table>
<thead>
<tr>
<th>Segmentation algorithm</th>
<th>Ref.</th>
<th>WM index</th>
<th>GM index</th>
</tr>
</thead>
<tbody>
<tr>
<td>EGS-SOM</td>
<td>[48]</td>
<td>0.76 ± 0.04</td>
<td>0.73 ± 0.05</td>
</tr>
<tr>
<td>R-FCM</td>
<td></td>
<td>0.75 ± 0.05</td>
<td>0.65 ± 0.05</td>
</tr>
<tr>
<td>NL-FCM</td>
<td>[49]</td>
<td>0.74 ± 0.05</td>
<td>0.72 ± 0.05</td>
</tr>
<tr>
<td>FCM</td>
<td></td>
<td>0.72 ± 0.05</td>
<td>0.74 ± 0.05</td>
</tr>
<tr>
<td>HFS-SOM</td>
<td></td>
<td>0.60 ± 0.08</td>
<td>0.60 ± 0.09</td>
</tr>
</tbody>
</table>

Table 5
Sensitivity and specificity values achieved by EGS-SOM and HFS-SOM algorithms.

<table>
<thead>
<tr>
<th>Tissue</th>
<th>EGS-SOM Sensitivity</th>
<th>EGS-SOM Specificity</th>
<th>HFS-SOM Sensitivity</th>
<th>HFS-SOM Specificity</th>
</tr>
</thead>
<tbody>
<tr>
<td>White matter</td>
<td>81.7%</td>
<td>95.7%</td>
<td>77.5%</td>
<td>81.5%</td>
</tr>
<tr>
<td>Gray matter</td>
<td>76.8%</td>
<td>96.4%</td>
<td>70.7%</td>
<td>80.3%</td>
</tr>
<tr>
<td>Cerebrospinal fluid</td>
<td>80.8%</td>
<td>99.8%</td>
<td>66.1%</td>
<td>85.4%</td>
</tr>
</tbody>
</table>
Fig. 16. Tanimoto performance index calculated through images from the IBSR2 database. Tanimoto’s index for WM (a) and GM (b) are shown for different segmentation algorithms.

Fig. 17. Segmented tissues of the 128:30:166 slice from the IBSR 100,23 volume, axial plane. (a) 2-Nearest neighbor clustering approach, (b) k-means, (c) EGS-SOM algorithm, (d) ground truth.
C-Means (FCM) [48], Robust FCM (R-FCM) [48] or Non-Local FCM [49] have been applied to the IBSR 2.0 images providing good results as shown in Table 4 and Fig. 16.

HFS-SOM is an unsupervised and automatic method which does not need any parameter to be set up. Moreover, as shown in Fig. 15, it performs better than other parametric or supervised methods. On the other hand, as improved supervised or parametric methods have been used to segment the IBSR 2.0 database they provides higher overlap ratio values than the HFS-SOM method. Although EGS-SOM is also a fully-unsupervised method, it requires some parameters regarding the feature selection and clustering stages.

As shown in Fig. 16, segmentation using the EGS-SOM method yields similar results than other existing methods for normalized images. Nevertheless, the presented methods outperform other methods with images containing intensity inhomogeneities as the IBSR 1.0 images. In the case of the EGS-SOM method, it provides better results with high resolution images due to the features (specially, the second order ones) are best captured using the overlapping window.

Although Tanimoto’s index is used to compare with other approaches, it does not provide an overall performance metric (i.e. in terms of error). This way, we provide sensitivity and specificity values for each tissue in Table 5.

In order to compare the results obtained with the different algorithms, Fig. 17 shows the segmentation results for the slice 166 (axial plane) of the 100.23 IBSR volume.

As shown in Fig. 17, all the methods tend to better delineate WM, while GM and CSF delineation depends on the method. This way, clustering the SOM using k-means (Fig. 17a) fails with GM in terms of its similarity with the ground truth. The 2-nearest neighbor approach (Fig. 17b) figures out GM better than k-means but the performance with WM is lower. On the other hand, the EGS-SOM (Fig. 17c) provides a good trade-off among the three tissues. As commented before in this paper, IBSR expert segmentations do not include internal CSF spaces. This way, assessment of segmentation methods which delineates sulcal CSF cannot be completely addressed using the IBSR references (ground truth) for CSF.

We used the expert manual segmentation provided by the IBSR database as a reference for calculating the overlap metric shown in Table 4.

Images from IBSR have been used to evaluate the segmentation algorithms. These images come with a resolution of 256 × 63 × 256 (IBSR 1.0) and 256 × 128 × 256 (IBSR 2.0). In the following, the EGS-SOM algorithm is tested on a high resolution VHN database. Although expert segmentation references are not available for these high resolution images, the results obtained (Fig. 19) which corresponds to the segmentation of the image in Fig. 18, clearly show the benefits of the proposed method, nevertheless it was not possible to calculate hits and false positive ratios in this case.

As stated in our experiments, the results obtained depend on the image resolution. The method used for calculating the clusters has been proved to be more effective when the window size is small beside the image size. On the other hand, using small window sizes is not effective for texture calculation. In images with a lower resolution, the window size should be decreased in order to have higher image size–window size ratio. Nevertheless, smaller window sizes do not capture the textural features of the image, and it is necessary to achieve a trade off between window size and resolution. Moreover, the HFS-SOM method provides similar results to other previously segmentation algorithms when the image does not contain severe inhomogeneities.

4. Conclusions

Two unsupervised MR image segmentation methods based on self-organizing maps were presented in this paper. The first method uses information computed from the whole volume histogram in order to classify the voxels using SOM (HFS-SOM). Moreover, the SOM prototypes, which generalize the input vectors, are clustered by the k-means algorithm. SOM prototypes generalize and quantize the intensities present on the MRI, taking into account the probability of each voxel intensity. This process does not need any parameter. On the other hand, HFS-SOM is computationally efficient and allows the segmentation of the whole volume at once in a fast way. The evaluation experiments carried out showed that the proposed HFS-SOM method provides good results with images which do not contain severe intensity inhomogeneities. Although the average overlap ratio is lower for the HFS-SOM method, it has been compared with supervised or parametric methods, while HFS-SOM does not require any parameter to be selected, exploiting the generalization properties of the SOM. EGS-SOM does not use the image histogram to compose the feature vectors, but the first and second order computed from the image using an overlapping window. In addition, evolutionary computing is used for selecting the most discriminative set of features leveraging the performance of the classifier. As a result, the number of units on the map is also optimized as well as the classification process. On the other hand, a map clustering method has been devised based on the EGS-SOM which has proved to be robust under noisy or bad intensity normalization conditions and provides good results with high resolution images. Thus, HFS-SOM or EGS-SOM methods can be used depending on the tradeoff between computational cost and precision.
The experiments performed using the IBISR database as well as high resolution images from VNH provide good results. Indeed, the results shown in Section 3 has been compared with the segmentations provided by the IBISR database obtaining about 78% for gray matter and about 81% for white matter of hits from the manual segmentations. Moreover, the clustering method for the SOM devised in this work provides better results for this application than other algorithms such as k-means or Fuzzy k-means. As a result, the number of segments or different tissues found in a MR image is figured out automatically making possible to find out tissues which could be identified with pathology.

Experiments performed using high resolution real brain scans from VNH yield good results, especially for CSF delineation. However, as expert segmentation is not provided for these images, quantitative assessment through the Tanimoto’s index is not possible.

It is worth noting that all the experiments have been performed over real brain scans. Thus, all the images used on this work to test our algorithm contain noise due to the acquisition process.

Segmentation techniques could help to find causes of brain disorders or anomalies such as Alzheimer’s disease. In fact, the segmentation algorithms presented in this paper are part of a larger study performed by the authors on the use of tissue distribution for the early diagnosis of AD. This way, generalization provided by the HFS-SOM method as well as precise tissue distribution generated by EGS-SOM segmentation may be applied over the most relevant slices to build AD brain models allowing further NORMAL/AD classification.

Acknowledgments

This work was partly supported by the MICINN under TEC2008-02113 and TEC2012-34306 projects and the Consejería de Innovación, Ciencia y Empresa (Junta de Andalucía, Spain) under the Excellence Projects P07-TIC-02566, P09-TIC-4530 and P11-TIC-7103.

Appendix A. Textural features

The mathematical formulation for the textural features mentioned in Section 2.3.2 is detailed as follows:

Energy \( E = \sum_{i=1}^{G-1} \sum_{j=0}^{G-1} p^2(i,j) \)

Entropy \( H = -\sum_{i=0}^{G-1} p(i,j) \log(p(i,j)) \)

Contrast \( C = \sum_{i=1}^{G-1} \sum_{j=0}^{G-1} (i-j)^2 p(i,j) \)

Homogeneity \( Hom = \sum_{i=1}^{G-1} \sum_{j=0}^{G-1} \frac{p(i,j)}{1 + (i-j)^2} \)

Sum average \( Sum_{av} = \sum_{i=0}^{2G-2} ip_{ij}(i) \)

Autocorrelation \( Ac = \sum_{i=1}^{G-1} \sum_{j=0}^{G-1} (i-j)p(i,j) \)

Maximal correlation coefficient \( Mor = \sqrt{2^{nd \text{ largest eigenvalue of } Q}} \)

Correlation \( Cor = \frac{\sum_{i=1}^{G-1} \sum_{j=0}^{G-1} ip(i,j) - \mu_x \mu_y}{\sigma_x \sigma_y} \)

Angular second moment \( ASM = \sum_{i=1}^{G-1} \sum_{j=0}^{G-1} p(i,j)^2 \)

Maximum probability \( MP = \max_{i,j} p(i,j) \)

Cluster prominence \( CP = \sum_{i=1}^{G-1} \sum_{j=0}^{G-1} (i-j - \mu_x - \mu_y)^2 p(i,j) \)

Cluster shade \( CS = \sum_{i=1}^{G-1} \sum_{j=0}^{G-1} (i-j - \mu_x - \mu_y)^2 p(i,j) \)

Dissimilarity \( Dis = \sum_{i=1}^{G-1} \sum_{j=0}^{G-1} |i-j| \ p(i,j) \)

Variance \( Var = \sum_{i=1}^{G-1} \sum_{j=0}^{G-1} (1 - \mu)^2 p(i,j) \)

Notation:

- \( p(i,j) \) corresponds to the \((i,j)\)-th entry in the normalized gray level spatial dependence matrix.
- \( p_x(i) = \sum_{j=0}^{G-1} p(i,j) \), \( p_y(j) = \sum_{i=0}^{G-1} p(i,j) \)
- \( p_{x+y}(k) = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} p(i,j), \quad i+j = k, \quad k = 1, \ldots, 2 \times (G-1) \)
- \( \mu_x, \mu_y, \sigma_x, \sigma_y \), are respectively the means and standard deviations of the partial probability density functions \( p_x \) and \( p_y \). In the case of variance calculation, \( \mu \) represents the mean of the values within the Gray Level Co-occurrence Matrix.
- \( \sigma_x^2 = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} (i-\mu_x)^2 \); \( \sigma_y^2 = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} (j-\mu_y)^2 \)
- \( \mu_i = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} ip(i,j) \)
- \( Q(i,j) = \sum_{k=0}^{G-1} \frac{p(k \times \mu_{xy}(k))}{p_x(i)p_y(j)} \)
- \( G \) is the total number of intensity levels.
- \( 2 \times 2 \times 2 \) corresponds to the number of voxels on the overlapped window.

References


