Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2014, Article ID 291581, 14 pages
http://dx.doi.org/10.1155/2014/291581

Research Article

A Hybrid DE-RGSO-ELM for Brain Tumor Tissue Categorization in 3D Magnetic Resonance Images

K. Kothavari, B. Arunadevi, and S. N. Deepa

Department of EEE, Regional Centre, Anna University, Coimbatore 641 047, India

Correspondence should be addressed to K. Kothavari; kothaskvn@gmail.com

Received 28 February 2014; Accepted 21 May 2014; Published 21 July 2014

Academic Editor: Anand Paul

Medical diagnostics, a technique used for visualizing the internal structures and functions of human body, serves as a scientific tool to assist physicians and involves direct use of digital imaging system analysis. In this scenario, identification of brain tumors is complex in the diagnostic process. Magnetic resonance imaging (MRI) technique is noted to best assist tissue contrast for anatomical details and also carries out mechanisms for investigating the brain by functional imaging in tumor predictions. Considering 3D MRI model, analyzing the anatomy features and tissue characteristics of brain tumor is complex in nature. Henceforth, in this work, feature extraction is carried out by computing 3D gray-level cooccurence matrix (3D GLCM) and run-length matrix (RLM) and feature subselection for dimensionality reduction is performed with basic differential evolution (DE) algorithm. Classification is performed using proposed extreme learning machine (ELM), with refined group search optimizer (RGSO) technique, to select the best parameters for better simplification and training of the classifier for brain tissue and tumor characterization as white matter (WM), gray matter (GM), cerebrospinal fluid (CSF), and tumor. Extreme learning machine outperforms the standard binary linear SVM and BPN for medical image classifier and proves better in classifying healthy and tumor tissues. The comparison between the algorithms proves that the mean and standard deviation produced by volumetric feature extraction analysis are higher than the other approaches. The proposed work is designed for pathological brain tumor classification and for 3D MRI tumor image segmentation. The proposed approaches are applied for real time datasets and benchmark datasets taken from dataset repositories.

1. Introduction

Brain tumor is a common malignancy found in humans. Basic knowledge of tumor activities and interactions in malignancies enables efficient diagnosis and clinical understanding for the physician. As a result, brain tumor optimization methodologies possess higher potential for carrying out clinical diagnostics. In the growing scenario, medical imaging techniques (X-Ray, CT, MRI), functional imaging techniques (single photon emission computed tomography—SPECT, positron emission tomography—PET, and functional magnetic resonance imaging—fMRI), and signal diagnostic techniques such as ECG and EEG are used widely to achieve a prior analysis of the malignancies and for appropriate therapy planning. Locating brain tumor segmentation and classification of brain and tissues within magnetic resonance (MR) images is integral to the treatment of brain cancer. Segmenting brain tumors in MR images involves classifying each voxel as normal brain tissues and tumor tissues, based on the description of that voxel. This task, a prerequisite for treating brain cancer using radiation therapy, is typically done manually by expert medical doctors, who find this process laborious and time-consuming. Replacing this manual effort with a good automated pattern recognition model would save expert time; the resulting labels may also be more accurate and more consistent.

Texture analysis [1] is an efficient measure to estimate the structural orientation, roughness, smoothness, or regularity differences of diverse regions in an image scene, showing promising results as an image analysis method for detecting nonvisible and visible lesions, with a number of applications in magnetic resonance imaging MRI. Most classification techniques offer gray-level pixel based statistical features. Also, many statistical and machine learning perceptions have been identified for medical image classification. The major disadvantages in the existing approaches [2–6] are based on the independency in the considered process from any functional form, even when no prior assumptions are possible and when only data is available. As a result the process of feature selection and classification remains a challenge due to the facts of lack of large diversity in shape and appearance of clear edge between adjacent tissues [7], intensities of same tissues varying in different locations, noisy images, accuracy in classifier design and interpretability of the volumetric feature analysis.

The choice of highly discriminating texture features is the most important factor for a success in texture segmentation, but this has been neglected in earlier approaches. 2D cooccurrence or run-length features may lack the sensitivity to identify larger scale or more coarse changes in spatial frequency. Secondly, there are a great many pixels (such as 256 × 256 × 124) for 3D MR images. Consequently, segmentation will have high computational complexity and require large memory storage. This problem can be solved by applying the 2D methods sequentially, and even experts segment the images in this way. However, this will lose some of the geometric information.

To overcome the above difficulty, in recent years, some of the research work shows various 3D texture features [8–10]. All these approaches applied a 3D GLCM and 3D RLM method to computed tomography (CT) images to separate various organs of human body. Few other approaches [2, 11] computed statistical, gradient, and Gabor filter features at multiple scales and orientations in 3D to capture the entire range of shape, size, and orientation of the tumor on automated segmentation of prostatic adenocarcinoma from high-resolution MR images of brain. Certain approaches analyzed mean and standard deviation of FLAIR signal intensities of texture analysis on the hippocampal body [12]. A volumetric congruent local binary pattern (LBP) algorithm for 3D neurological image retrieval was also carried out. This proposed work performs volumetric three dimensional texture analysis to extract the better features of the considered images and these extracted features are further utilized for classification the images.

Therefore in this paper, basic differential evolution [13] is employed to select appropriate features for dimensionality reduction and thus DE is used as a feature subselection process after the features are extracted employing 3D GLCM and RLM. After the selection of optimal features using DE algorithm, then the refined group search optimizer based extreme learning machine is used to carry out the classification process. In the proposed RGSO based ELM, the weights and bias to ELM are optimized using RGSO for better simplification and classification of pathological brain tissues and tumors. The schematic flow for 3D image classification framework is shown in Figure 1.

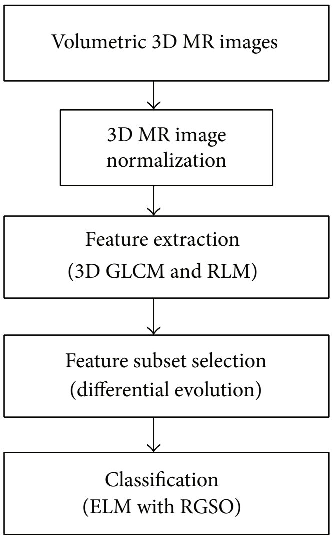
Volumetric 3D MR images
3D MR image normalization
Feature extraction (3D GLCM and RLM)
Feature subset selection (differential evolution)
Classification (ELM with RGSO)

FIGURE 1: Framework of proposed classification architecture.

2. Proposed 3D Volumetric Feature Extraction and Subselection Model

In general, feature extraction is a process that transforms the input data into a set of features. A feature is said to be a rich one when it is characterized by appropriate texture, shape, intensity, and location. All these said features are selected with an expert opinion from the radiologists. 3D texture measures describe the distribution using joint second order statistics of volume element (voxel) values in an image. Since the 2D GLCM algorithm is not able to describe the pixel pair displacements in 3D images, the approach extends to the GLCM and RLM for 3D volumetric data. Basically the image is represented by a function $f(x, y, z)$ containing three space variables x, y, and z, where $x = 0, 1, \ldots, N - 1$, $y = 0, 1, \ldots, M - 1$, and $z = 0, 1, \ldots, L - 1$. The function $f(x, y, z)$ can take any value $i = 0, 1, \ldots, G - 1$, where G is total number of intensity levels in the image. The optimal feature set is categorized as follows.

Image intensity (e.g.: mean, standard deviation)

Texture features (e.g.: gray-level cooccurrence and run-length matrices).

2.1. 3D Gray-Level Cooccurrence Matrices. The gray-level cooccurrence matrix was originally proposed in the year 1973 [14]. The 3D GLCM developed in this work is extended from original 2D GLCM, a three-dimensional cooccurrence matrix, M, which is also an $n \times n$ matrix, where n is the number of gray levels of an image. In order to improve the computational efficiency, the number of gray levels (i.e., databit) is usually reduced. The 3D GLCM is basically defined as

$$M(i, j) = \sum_{z=1}^{Wz-dz} \sum_{x=1}^{Wx-dx} \sum_{y=1}^{Wy-dy} \text{CONDITION}$$

$$\text{CONDITION} = (G(x, y, z) = i \wedge G(x + dx, y + dy, z + dz) = j)?1 : 0 \qquad (1)$$

$$M_d(i,j) = \sum_{z=1}^{Wz-dz} \sum_{x=1}^{Wx-dx} \sum_{y=1}^{Wy-dy} \delta\left[G(x,y,z) = i, G(x+dx, y+dy, z+dz) = j\right]$$
$$\text{if } \delta \ [\text{True}] = 1, \text{ else } \delta \ [\text{False}] = 0, \tag{2}$$

where i represents the ith column and j represents the jth row in 3D GLCM; d represents the relation of a voxel pair (voxel pairs are defined by a distance and direction which can be represented by a displacement vector $d = (dx, dy, dz)$, where dx and dy represent the displacement (in pixels) along x-axis and y-axis in spatial domain and dz is the number of bands in the spectral domain). $G(x, y, z)$ and $G(x+dx, y+dy, z+dz)$ are the gray values in the position (x, y, z) and $(x + dx, y + dy, z + dz)$ of a moving window $(Wx \times Wy \times Wz)$, and x, y, z are denoted as the position of a moving window in the volumetric data. The three-component vector (x, y, z) with each component equal to –1, 0, or 1 is used to define different directions [15]. The vector describes neighbour's direction in Cartesian coordinate system [6].

Four variables that are required to use GLCM include the quantization level of the image, the size of the moving window, the direction and distance of pixel pairs, and the statistics used as texture measurements. According to these four parameters, texture images can be extracted using 3D GLCM and can be used as features for analysis or classification.

2.2. Run-Length Matrices. Run-length statistics capture the coarseness of a texture in specified directions. A run is defined as a string of consecutive pixels which have the same gray-level intensity along a specific linear orientation to describe the frequency of appearance. Fine textures tend to contain more short runs with similar gray-level intensities, while coarse textures have more long runs with significantly different gray-level intensities [1, 16].

A run-length matrix P is defined as follows: each element $P(i, j)$ represents the number of runs with pixels of gray-level intensity equal to i and length of run equal to j along a specific orientation. For a given 3D image, presented as a series of slices in a preferred slice orientation, a run-length matrix P is defined as follows: each element $P(i, j)$ represents the number of runs with pixels of gray-level intensity equal to i and length of run equal to j along the $d(x, y, z)$ direction. An orientation is defined using a displacement vector $d(x, y, z)$, where x, y, and z are the displacements for the x-axis, y-axis, and z-axis, respectively. Unlike the 2D texture characterization, the volumetric texture requires 13 different displacements from a total of 26 possible displacements in a three dimensional space [17].

The length of the run is the total number of pixel points in the run and the gray-level run-length feature is estimated using

$$R(\theta) = \left[g(i,j) \mid \theta\right], \quad 0 \le i \le G, \ 0 \le j \le R_{\max}, \tag{3}$$

where G is the maximum gray level and $R_{\max}$ is the maximum run length which is equal to max $\{L_x, L_y, L_z\}$. The element $g(i, j \mid \theta)$ specifies the estimated number of times that a given picture contains a run length j for a gray level i in the direction of the angle θ and L_x, L_y, and L_z denote the length of the ROI in x, y, and z directions. The textural features are measured from $R(\theta)$.

Once the run-length matrices are calculated along each direction, several texture descriptors are calculated to capture the texture properties and are differentiated among various features. These descriptors can be used either with respect to each direction or by combining them if a global view of the texture information is required. Eleven descriptors are typically extracted from the run-length matrices: short run emphasis (SRE), long run emphasis (LRE), high gray-level run emphasis (HGRE), low gray-level run emphasis (LGRE), pair-wise combinations of the length and gray-level emphasis (SRLGE, SRHGE, LRLGE, and LRHGE), run-length nonuniformity (RLNU), gray-level nonuniformity (GLNU), and run percentage (RPC). Some of these descriptors reflect specific characteristics in the image. For example, SRE measures the distribution of short runs in an image, while run percentage measures both the homogeneity and the distribution of runs of an image in a specific direction.

2.3. 3D GLCM and RLM Based Feature Extraction. The extraction of optimal features is based on the characteristics of texture, shape, intensity of image, and spectral density. The key approach employed to extract volumetric features is based on 3D GLCM, gradient vector composition, and RLM. 3D volume analysis process occurs in 13 directions, which results in 13 3D GLCM and RLM. Figure 2 shows the 3D volumetric directions.

In a 3D volumetric space, the directions are selected by linking a voxel to each of its nearest 26 $(= \ 3 * 9 - 1)$ neighbours, respectively, leading to 13 different directions from a total of 26 possible directions. Each of these slices is processed at once producing only one run-length encoding matrix for all consecutive slices forming the 3D image and, thus, the run-length computation for the volumetric texture is faster. In this case, it should be observed that 11 texture features are calculated for characterizing the texture for each subregion as in Table 1.

The spatial dependence of gray-level values across multiple slices is captured by 3D GLCM. The GLCM matrix is calculated for 0, 45, 90, and 135 degrees for θ and a distance scale of 1. Here 4 spatial distances and 13 directions are chosen, computing 52 (13 × 4) directional vectors and cooccurrence matrices. Henceforth, statistical measures considered are variance, entropy, energy, contrast, and homogeneity, which are calculated for each matrix. Employing these statistical measures, the feature vector of 260 (5 × 52) components is formed. The remaining features include 11 RLM and 2 gradient vector and gradient orientation features. The optimal features set with 18 texture descriptors were calculated.

2.4. Feature Subselection Model. The process of subselection in image processing selects optimal feature subset from

Table 1: 3D feature directions.

	3D texture directions	
	Offset	Degree direction (θ, ϕ)
1	0, 1, 0	0°, 0°
2	−1, 1, 0	45°, 0°
3	−1, 0, 0	90°, 0°
4	−1, −1, 0	135°, 0°
5	0, 1, −1	0°, 45°
6	0, 0, −1	0°, 90°
7	0, −1, −1	0°, 135°
8	−1, 0, −1	90°, 45°
9	1, 0, −1	90°, 135°
10	−1, 1, −1	45°, 45°
11	1, −1, −1	45°, 135°

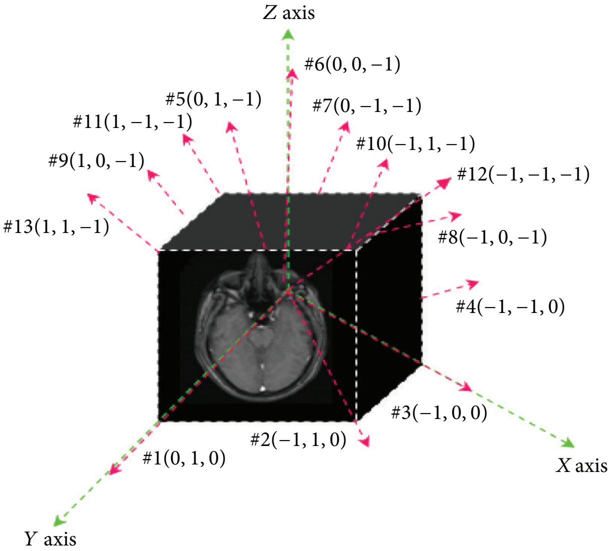

Figure 2: 3D volumetric directions.

the existing feature sets to achieve output conceptions in image classification. The aim of the subselection process is to extract best minimal subset from the original element set, instead of transforming the data to new dimensions. Various measures will be taken to analyze features from different perspectives, but few features extracted tend to behave similar without any specific variations. Also, the improvement in the dimensionality of texture features decreases the memory and the computational time. The criterion here is to identify the features that are correlated or predict the class label. The objective of this search in subset includes maximization of this criterion. Feature subselection model typically incorporates a search strategy for exploring the space of feature subsets [18]. More sophisticated search strategies such as genetic algorithm (GA) [19, 20] or simulated annealing (SA) can be employed to better explore the search space. This work invokes differential evolution (DE) technique for feature subselection.

In recent years, there has been a growing interest in evolutionary algorithms for diverse fields of science and engineering. The differential evolution algorithm (DE) [13] is a relatively novel optimization technique for solving numerical-optimization problems. The algorithm has successfully been applied to several sorts of problems as it has claimed a wider acceptance and popularity, following its simplicity, robustness, and good convergence properties.

2.4.1. Need for DE Algorithm. The DE algorithm is population based including a simple and direct searching algorithm for globally optimizing multimodal functions. Just like the genetic algorithms (GA), it employs crossover and mutation operators as selection mechanisms. As previously mentioned, an important difference among other evolutionary computational techniques, such as genetic algorithms (GA), is that the GA relies on the crossover operator which provides the exchange of information required to build better solutions. DE algorithm fully relies on the mutation operation as its central procedure. The applicability of DE algorithm, in comparison with that of the other approaches like GA, simulated annealing, or Tabu search, includes its exploration and exploitation capability because of its nonuniform crossover and mutation operations. The advantage of DE algorithm in exploring is utilized for its convergence for feature selection process. This enables the search to be focused on the most promising area of the solution space. The mutation operation is based on the differences of randomly sampled pairs of solutions within the population. Besides being simple and capable of globally optimizing multimodal search spaces, the DE algorithm shows other benefits: it is fast, easy to use, and can very easily adapt in the case of integer or discrete optimizations. It is quite effective for nonlinear constraint optimization, including penalty functions.

2.4.2. DE Algorithm Flow Process [13]. Classic DE algorithm begins by initializing a population of N_p and D-dimensional vectors with parameter values which are randomly and uniformly distributed between the prespecified lower initial parameter bound $X_{j,\text{low}}$ and the upper initial parameter bound $X_{j,\text{high}}$:

$$X_{j,i,t} = X_{j,\text{low}} + \text{rand}(0,1) \cdot \left(X_{j,\text{high}} - X_{j,\text{low}}\right); \quad j = 1, 2, \ldots, D; \; i = 1, 2, \ldots, N_p; \; t = 0. \tag{4}$$

The subscript t is the generation index, while j and i are the parameter and population indexes, respectively. Hence, $X_{j,i,t}$ is the jth parameter of the ith population vector in generation t.

To generate a trial solution, DE algorithm first mutates a best solution vector from the current population by adding it to the scaled difference of two other vectors from the current population,

$$V_{i,t} = X_{\text{best},t} + F \cdot \left(X_{r_1,t} - X_{r_2,t}\right); \quad r_1, r_2 \in \left\{1, 2, \ldots, N_p\right\}, \tag{5}$$

with $V_{i,t}$ being the mutant vector. Vector indexes r_1 and r_2 are randomly selected considering that both are distinct and different from the population index i (i.e., $r_1 \neq r_2 \neq i$). The mutation scale factor F is a positive real number typically less than 1.

The next step considers one or more parameter values of the mutant vector $V_{i,t}$ to be exponentially crossed with those belonging to the ith population vector $X_{i,t}$. The result is the trial vector $U_{i,t}$

$$U_{j,i,t} = \begin{cases} v_{j,i,t}, & \text{if rand}\,(0,1) \le c_r \text{ or } j = j_{\text{rand}} \\ x_{j,i,t}, & \text{otherwise} \end{cases} \qquad \text{with } j_{\text{rand}} \in \{1, 2, \ldots, D\}. \tag{6}$$

The crossover constant ($0.0 \le c_r \le 1.0$) controls the section of parameters belonging to the mutant vector which contributes to the trial vector. In addition, the trial vector always inherits the mutant vector parameter with a random index j_{rand} to ensure that the trial vector differs by at least one parameter from the vector to be compared ($x_{i,t}$).

Finally, a selection operation is used to improve the solutions. If the cost function of the trial vector is less than or equal to the target vector, then the trial vector replaces the target vector on the next generation. Otherwise, the target vector remains in the population for at least one new generation:

$$X_{i,t+1} = \begin{cases} U_{i,t}, & \text{if } f(U_{i,t}) \le f(X_{i,t}), \\ X_{i,t}, & \text{otherwise.} \end{cases} \tag{7}$$

Here, f represents the cost function. These steps are repeated until a termination criterion is attained or a predetermined generation number is reached.

The differential evolution process reduces the dimensionality into 7 most decisive features out of the 18 features extracted. The optimal features selected after this process include gradient vector parameter, gray-level nonuniformity, energy, variance, entropy, sum variance, and short run emphasis. Therefore, from the existing 18 features, the optimal 7 feature subsets are extracted and applied for the next step.

3. ELM-RGSO for Brain Tumor and Tissue Classification

Over the past few decades it has been noted that artificial neural networks [21] play a major role in pattern recognition and image classification applications. It is because of their generalization and conditioning capabilities, requirement of minimal training points, and faster convergence time. ANNs are found to perform better and result in faster output in comparison with that of the conventional classifiers. Various neural network architectures [21] like radial basis function network, probabilistic neural network, back propagation neural network, and support vector machines and its variants were used for pattern and image classification applications; the underlying limitations in all these cases include the selection time incurred due to the preprocessing speed delay. For better classification accuracy, more training data is to be utilized in comparison to that of the testing data. Also, the selection of appropriate training algorithms is important, which enables the considered application to not experience the local or global minima problem. The above addressed criteria for improving the training performance, so as to result in better classification accuracy, have been noted in a neural network architecture, extreme learning machine (ELM) classifier proposed in [22, 23], which handles the training for single hidden layer feedforward neural networks.

3.1. Basic Extreme Learning Machine (ELM) Classifier.

Extreme learning machine classifier [23] is a single-hidden layer feedforward neural network, wherein the weights for the input and hidden layer, as well as the respective biases, are randomly assigned without any training process. ELM is considered as a network architecture, which reduces the training time since its network output weights are computed using Moore-Penrose inverse and using the norm least square solution. ELM [24, 25] is best suited for larger training samples and also the effect of number of hidden neurons using different ratios of the number of features of testing and training data was examined. This classifier is compared with that of the conventional neural network classifiers using the classification rate for brain tissue and tumor classification. Figure 3 shows the basic ELM architecture.

The basic ELM classifier algorithm is given as follows.

Give a training set $N = \{(x_i, t_i) \mid x_i \in R^n, t_i \in R^m, i = 1, \ldots, N\}$, kernel function $f(x)$, and hidden neuron $\widetilde{N}$.

Step 1. Select suitable activation function and number of hidden neurons $\widetilde{N}$ for the given problem.

Step 2. Assign arbitrary input weight w_i and bias b_i, $i = 1, \ldots, H$.

Step 3. Calculate the output matrix H at the hidden layer

$$\sum_{i=1}^{\widetilde{N}} \beta_i \cdot f\left(w_i \cdot x_j + b_i\right) = t_j, \ j = 1, \ldots, N; \ H\beta = T$$
$$H = f \cdot (w \oplus x + b). \tag{8}$$

Step 4. Calculate the output weight β as

$$\beta' = H^{\dagger} T, \tag{9}$$

where $H^{\dagger}$ is the Moore-Penrose generalized pseudoinverse of hidden layer output matrix.

3.2. Group Search Optimizer (GSO) Algorithm.

The group search optimizer algorithm [26] used in this paper is based on the biological producer-scrounger (PS) model, which assumes group members search either for “finding” (producer) or for “joining” (scrounger) opportunities. Animal scanning mechanisms (e.g., vision) are incorporated to develop the GSO algorithm. GSO also employs “rangers” which perform random walks to avoid entrapment in local minima. The population of the GSO algorithm is called a group and each individual in the population is called a member. The GSO algorithm is implemented for this work because of its nature of random walk in various directions. The movements of the members to find the solution are

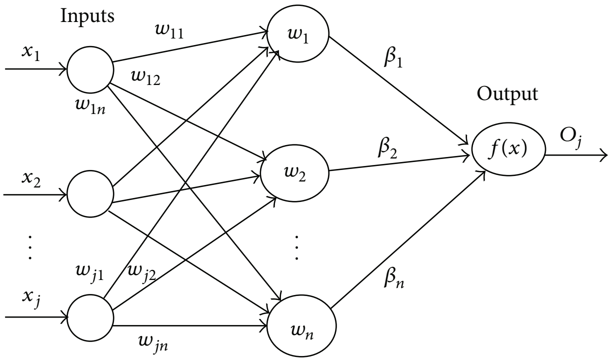

Figure 3: ELM classifier architecture.

carried out in a fast manner by eliminating the less efficient members in the group. This results in the accurate and faster convergence of the algorithmic process and, henceforth, this GSO algorithm has been chosen to be implemented for the proposed work.

In GSO [27], there are three types of members in a group: producers, scroungers, and dispersed members. There is only one producer and remaining members are either scroungers or dispersed members. Dispersed members are less efficient members who perform random walks. At each iteration, a group member, located in the most promising area, conferring the best fitness value, is chosen as the producer. The other group members are selected as scroungers or rangers by random. Then, each scrounger takes a random walk towards the producer, and each ranger takes a random walk in the arbitrary direction.

In n-dimensional search space, the ith member at kth iteration has a current position $X_l^k \in R^n$ and a head angle $\phi_i^k = (\phi_{i1}^k, \dots, \phi_{i(n-1)}^k) \in R^{n-1}$. Search direction of ith member is a unit vector $D_i^k(\phi_i^k) = (d_{i1}^k, \dots, d_{in}^k) \in R^n$ that can be calculated from ϕ_i^k via a Polar to Cartesian coordinate transformation.

At kth iteration the producer X^p behaves as follows.

(1) The producer will scan at zero degree and then scan laterally by randomly sampling three points in the scanning field: one point at zero degree,

$$X_z = X_p^k + r_1 l_{\max} D_p^k\left(\phi^k\right), \tag{10}$$

one point in the right hand side hypercube

$$X_r = X_p^k + r_1 l_{\max} D_p^k\left(\phi^k + \frac{r_2\theta_{\max}}{2}\right), \tag{11}$$

and one point in the left hand side hypercube

$$X_l = X_p^k + r_1 l_{\max} D_p^k\left(\phi_k - \frac{r_2\theta_{\max}}{2}\right). \tag{12}$$

$\theta_{\max} \in R^1$ is maximum pursuit angle and $l_{\max} \in R^1$ is maximum pursuit distance. $r_1 \in R^1$ is a normally distributed random number with mean 0 and standard deviation 1 and $r_2 \in R^{n-1}$ is a uniformly distributed random sequence in the range $(0, 1)$.

(2) The producer will then find the best point with the best resource (fitness value). If the best point has a better resource than its current position, then it will fly to this point or it will stay in its current position and turn its head to a new randomly generated angle. Consider

$$\phi^{k+1} = \phi^k + r_2\alpha_{\max}, \tag{13}$$

where $\alpha_{\max} \in R^1$ is the maximum turning angle.

(3) If the producer cannot find a better area after iterations, it will turn its head back to zero degree,

$$\phi^{k+a} = \phi^k, \tag{14}$$

where $a \in R^1$ is a constant.

At kth iteration, ith scrounger walks randomly towards the producer. Consider

$$X_i^{k+1} = X_i^k + r_3 o\left(X_p^k - X_i^k\right), \tag{15}$$

where $r_3 \in R^n$ is a uniform random sequence in the range $(0, 1)$. Operator o is the Hadamard product or the Schur product, which calculates the entry wise product of the two vectors. If a scrounger finds a better location than the current producer and other scroungers, then it will switch as producer in the next iteration.

The group members, who are less efficient foragers than the dominant, will be dispersed from the group. If the ith group member is dispersed, it will perform ranging. At the kth iteration, it generates a random head angle ϕ_i using (4); and then it chooses a random distance

$$l_i = a r_1 l_{\max} \tag{16}$$

and moves to the new point

$$X_i^{k+1} = X_i^k + l_i D_i^k\left(\phi^{k+1}\right). \tag{17}$$

To maximize their chances of finding resources, the GSO algorithm employs the fly-back mechanism [9] to handle the problem specified constraints. When the optimization process starts, the members of the group search the solution in a feasible manner. If any member moves into the infeasible region, it will be forced to move back to the previous position to guarantee a feasible solution.

3.3. Developed Refined GSO (RGSO). In the original GSO, 75% rest members will perform scrounging and the remaining 25% rest members will perform ranging. In this paper, the ranging operation for the remaining 25% rest members will not be done as it is done in original GSO. Instead, these members for ranging operation will learn from the "worst" member in its group. This refinement of learning from "worst" member leads to discovering better solution regions in complex optimization search spaces.

Compared to the original GSO [26], RGSO algorithm searches more promising regions to find the global optimum.

The difference between GSO and RGSO is that the differential operator is applied to only accept the basic GSO generating new better solution for each krill instead of accepting all the krill updating adopted in krill herd (KH). This is rather greedy. The original GSO is very efficient and powerful but highly prone to premature convergence. Therefore, to evade premature convergence and further improve the exploration ability of the original GSO, a differential guidance is used to tap useful information in all the krill individuals to update the position of a particular krill individual. Equation (18) expresses the differential mechanism. Consider

$$\begin{aligned} & Z_i - Z_j \\ & \quad = \left(z_{i1} \ \ z_{i2} \ \ z_{i3} \ \ \cdots \ \ z_{in}\right) - \left(z_{\rho 1} \ \ z_{\rho 2} \ \ z_{\rho 3} \ \ \cdots \ \ z_{\rho n}\right), \end{aligned} \tag{18}$$

where z_{i1} is the first element in the n dimension vector Z_i. z_{in} is the nth element in the n dimension vector Z_i. $z_{\rho 1}$ is the first element in the n dimension vector Z_p. ρ is the random integer generated separately for each z, between 1 and n, but $\rho \neq i$.

Accordingly, the refined GSO is presented in Algorithm 1.

Since the problem of interest in this research is complex in nature, the above refined GSO will be able to discover better regions to train the ELM for brain tumor classification.

3.4. RGSO Based ELM for Brain Tumor Classification. This proposed methodology combines the concept of RGSO for optimizing the weights in ELM neural network. This refined GSO with ELM enables the selection of input weights to increase the generalization performance and the conditioning of the single layer feedforward neural network. The steps of the proposed approach are as follows.

Step 1. Initialize positions and head angles with a set of input weights and hidden biases: $[W_{11}, W_{12}, \ldots, W_{1n}, \ldots, W_{21}, W_{22}, \ldots, W_{2n}, \ldots, W_{H1}, W_{H2}, \ldots, W_{Hn}, b_1, b_2, \ldots, b_H]$. These will be randomly initialized within the range of $[-1, 1]$ on D dimensions in the search space.

Step 2. For each member in the group, the respective output final weights are computed at ELM as given in (9).

Step 3. Now invoke refined GSO as in Table 1.

Step 4. Then the fitness, which is the mean square error (MSE) of each member, is evaluated as given below:

$$\text{MSE} = \frac{1}{N} \sum_{i=1}^{N} E_i^2 = \frac{1}{N} \sum_{i=1}^{N} \left(y_k^i - d_k^i\right)^2, \tag{19}$$

where N is the number of training samples and the terms y_k and d_k are the error of the actual output and target output of the kth output neuron of ith sample. Thus, fitness function $f(x)$ is defined by the MSE. In order to avoid overfitting of the single layer feedforward neural network, the fitness of each member is adopted as the mean squared error (MSE) on the validation set only instead of the whole training set as in [28].

Step 5. Find the producer of the group based on the fitness.

Step 6. Update the position of each member as given in (13) to (17).

Step 7. Stopping criteria: the algorithm repeats Steps 2–6 until certain criteria are met, along with hard threshold value as maximum number of iterations. On reaching the stopping criteria, the algorithm returns the optimal weights with minimal MSE as its solution.

Thus refined GSO (RGSO) with ELM finds the best optimal weights W and bias b so that the fitness reaches the minimum to achieve better generalization performance, with minimum number of hidden neurons, considering both the advantages of both ELM and RGSO. In the process of selecting the input weights, the refined GSO considers not only the MSE on validation set but also the norm of the output weights [29]. The proposed RGSO based ELM will combine the feature of RGSO into ELM to compute the optimal weights and bias to make the MSE minimal.

4. Experimental Results and Discussion

4.1. Clinical Datasets. The details of tumor benchmark dataset, normal healthy benchmark dataset, and real time clinical data from hospitals used in the simulation are given in Table 2. The clinical specimen utilized in the present study consists of brain MR-images of 70 clinical routine cases with verified and untreated intracranial tumors. Each image sequence with a 3 mm brain slice-interval (i.e., the voxel size was 0.4492 mm × 0.4492 mm × 33 mm) in axial plane was measured. Figure 4 shows the 10 SPL dataset.

The simulation tool employed for implementing the proposed work is MATLAB (Version 7.11) and was executed in computer with Intel core i5 processor with 2.53 GHz speed and 3 GB of RAM.

4.2. Volumetric Feature Analysis of Datasets. In medical applications, the small homogeneous tissue region is selected manually and is referred to as region of interest (ROI). In the 3D texture analysis, the same homogeneous region was maintained over ten additional neighboring slices from the reference slice (five up and five below) to form a cubic volume of interest (VOI) of about 1330 voxels in each class and in each subject. Four volume(s) of interest (VOI) of different brain tissues and tumor parts were identified for each patient by an experienced neuroradiologist: (i) solid (active tumor), (ii) white matter (WM), (iii) gray matter (GM), and (iv) cerebrospinal fluid (CSF) [30].

Homogeneous VOIs (400 to 1600 voxels) were carefully selected avoiding signals from adjacent tissues. A normalization process was carried out based on histogram and ROI/VOI. It delineates training areas for all classes over each VOI reducing the initial image's gray levels number to 128 in order to shorten calculation time and to avoid sparse matrices. 5 parameters of cooccurrence matrix, 11 parameters of run-length matrix, and 2 parameters of Sobel and Laplace gradient along with its orientations were calculated for 3D

```
Set $k$:= 0;
Randomly initialize positions $X_i$ and head angles $\phi$ of all members;
WHILE (the termination conditions are not met)
  For (each members $i$ in the group)
      Calculate fitness:    Calculate the fitness value of current member: $f(X_i)$
      Choose producer:      Find the producer $X_p$ of the group;
      Perform producing:  (1) The producer will scan at zero degree and then scan laterally by randomly sampling three points
                              in the scanning field using (10) to (12).
                          (2) Find the best point with the best resource (fitness value). If the best point has a better resource
                              than its current position, then it will fly to this point. Otherwise it will stay in its current
                              position and turn its head to new angle using (13).
                          (3) If the producer cannot find a better area after $\alpha$ iterations, it will turn its head back to zero
                              degree using (14);
      Perform scrounging: Randomly select 75% from the rest members to perform scrounging using (15);
      Perform ranging:      For the rest 25% members, they will perform ranging:
                            Find the worst point with the worst resource (fitness value). If the worst point has a better
                            resource than its current position, then it will fly to this point. These members will update
                            their position based on this worst resource.
  END FOR
  Set $k$:= $k$ + 1;
END WHILE
```

Algorithm 1: Refined GSO algorithm.

Table 2: Clinical datasets.

Dataset	Number of cases	Tumor types	Slices
Surgical Planning Laboratory (SPL) (Harvard Medical School, Brigham and Women's Hospital, Boston)	10	Astrocytoma (3) Meningiomas (3) Gliomas (4)	124 slices each
McConnell Brain Imaging Centre (Montreal Neurological Institute)	20	Healthy MRI brain	30 slices each
PSG Institute of Medical Science and Research (Coimbatore)	70	Meningiomas (28) Gliomas (35) Normal (7)	100 slices each

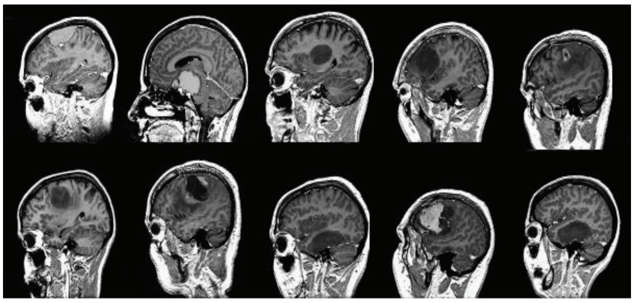

Figure 4: SPL brain tumor dataset (cases 1–10).

methods, with five distance $d = 1, 2, 3, 4$, and 5. The 3D feature extraction method from GLCM, RLM, and gradient model considers the 26 neighbors of a voxel. Tables 3 and 4 show the 3D VOI GLCM texture measures for real time clinical datasets and SPL datasets, respectively.

The relevant features are identified using area under the curve (AUC) value from receiver operating characteristics (ROC) curve and P values from 2-tailed Student's t-test. Statistical analysis is evaluated based on P and AUC values, where the best features are ranked. Table 5 shows the relevant features as indicated by high value of AUC and low value P for 3D VOI features.

4.3. Feature Subselection Using DE. The rich aggregate bank of features of the tissue image was calculated by using the gray-level run-length matrix method, GLCM, gradients, and intensity. In the proposed system, twenty-seven parametric features were extracted, with a total of 1755 features for each VOI. The possible outcome of the employing DE is to provide the optimal set of features which can be used as input to the classifier. These parameters were selected using differential evolution which provides excellent selection capabilities for any dataset. So out of the 18 features, an optimal 7 decisive features are selected and are taken for the next step to classification process. Table 6 shows the optimal subselected features using differential evolution approach.

From Table 6, the maximum and minimum values of subselected feature parameters for healthy and tumorous brains can be noted. Figures 5 and 6 show the MRI brain

Table 3: 3D VOI GLCM texture measures for real time datasets.

Real time sample cases	Average contrast	Average variance	Average entropy	Average homogeneity	Average energy
Sample 1	4.2678	0.9625	0.0945	0.9786	0.9123
Sample 2	16.343	0.8456	0.3395	0.9302	0.8909
Sample 3	1.1556	0.9956	0.0303	0.9967	0.9201
Sample 4	74.567	0.6454	0.7563	0.6839	0.5976
Sample 5	26.455	0.7683	0.4452	0.8976	0.9100

Table 4: 3D VOI GLCM texture measures for SPL datasets.

SPL data	Average contrast	Average variance	Average entropy	Average homogeneity	Average energy
Case 1	12.5422	0.9045	0.26735	0.9453	0.9932
Case 2	7.3437	0.9328	0.15245	0.9617	0.9411
Case 3	4.6964	0.9585	0.10232	0.9863	0.9910
Case 4	8.2945	0.9243	0.16943	0.9615	0.7628
Case 5	94.5528	0.4291	0.8954	0.5945	0.3219
Case 6	23.5943	0.7834	0.4433	0.9078	0.4631
Case 7	24.8594	0.7711	0.4664	0.90685	0.8621
Case 8	11.4932	0.8945	0.2466	0.9402	0.9275
Case 9	8.2312	0.9198	0.1684	0.9645	0.7513
Case 10	3.2587	0.9722	0.0635	0.9876	0.8871

Table 5: Clinical and imaging characteristics of 70 patients using Student's t-test for 3D VOI.

Features	AUC value	P values (Student's t-test)	Ranking
Energy	0.91	$4.28E^{-4}$	3
Entropy	0.87	$3.55E^{-6}$	5
Contrast	0.83	$3.02E^{-2}$	10
Homogeneity	0.56	$8.85E^{-6}$	27
Correlation	0.63	$2.74E^{-2}$	23
Variance	0.90	$1.20E^{-5}$	4
Inverse difference moments	0.61	$8.46E^{-6}$	24
Sum mean	0.84	$5.56E^{-3}$	9
Sum variance	0.86	$3.13E^{-5}$	6
Sum entropy	0.72	$3.66E^{-3}$	16
Difference entropy	0.85	$4.76E^{-8}$	8
Infm. correlation 1	0.64	$5.05E^{-5}$	22
Infm. correlation 2	0.71	$4.67E^{-5}$	17
Maximum correlation coefficient.	0.65	$3.75E^{-2}$	21
Short run emphasis	0.85	$7.92E^{-8}$	7
Long run emphasis	0.69	$1.562E^{-3}$	19
Low gray-level run emphasis	0.68	$4.56E^{-2}$	20
High gray-level run emphasis	0.81	$3.21E^{-2}$	11
Short run low gray-level run emphasis	0.77	$5.12E^{-1}$	14
Short run high gray-level run emphasis	0.79	$6.10E^{-3}$	12
Long run low gray-level run emphasis	0.61	$3.88E^{-8}$	25
Long run high gray-level run emphasis	0.72	$1.459E^{-4}$	15
Gray-level nonuniformity	0.93	$3.15E^{-2}$	2
Run-level nonuniformity	0.59	$1.49E^{-5}$	26
Run percentage	0.70	$1.47E^{-8}$	18
Gradient vector parameter	0.93	$6.2E^{-2}$	1
Gradient vector orientation	0.78	$4.2E^{-3}$	13

Table 6: Optimal texture features subselected.

Selected optimal features	Rank	3D ROI			
		Healthy brain		Tumorous brain	
		Minimum	Maximum	Minimum	Maximum
Gradient vector parameter	1	0.2139	4.5392	20.7156	35.720
Gray-level nonuniformity	2	0.2267	0.4327	0.4962	0.9197
Energy	3	0.1971	6.1215	31.3257	38.5102
Variance	4	5.0025	5.9903	10.7982	13.0159
Entropy	5	0.9731	0.9963	0.7987	0.8721
Sum variance	6	4.3196	5.1957	13.9621	15.5164
Short run emphasis	7	1.0096	1.7203	3.4512	4.3271

Table 7: Comparison of tumor classification results to state-of-the-art techniques.

Case (SPL)	CT	NC	CC	SPL	GA-SVM	GA-ELM	GA-ELM-IPSO [28]	DE-ELM-RGSO
1	0.94	0.97	0.97	0.98	0.97	0.97	0.98	0.98
2	0.94	0.91	0.86	0.91	0.89	0.90	0.92	0.96
3	0.97	0.95	0.97	0.96	0.96	0.94	0.95	0.94
4	0.87	0.91	0.88	0.91	0.88	0.89	0.88	0.88
5	0.68	0.71	0.70	0.85	0.65	0.70	0.74	0.72
6	0.96	0.94	0.87	0.98	0.91	0.90	0.94	0.96
7	0.57	0.69	0.93	0.84	0.90	0.90	0.90	0.94
8	0.98	0.96	0.96	0.88	0.93	0.92	0.94	0.96
9	0.93	0.94	0.94	0.97	0.93	0.94	0.94	0.96
10	0.92	0.94	0.94	0.96	0.93	0.94	0.94	0.96
Mean	0.87	0.86	0.90	0.91	0.895	0.9	0.912	0.9260
STD	0.14	0.14	0.08	0.053	0.09	0.074	0.066	0.0772

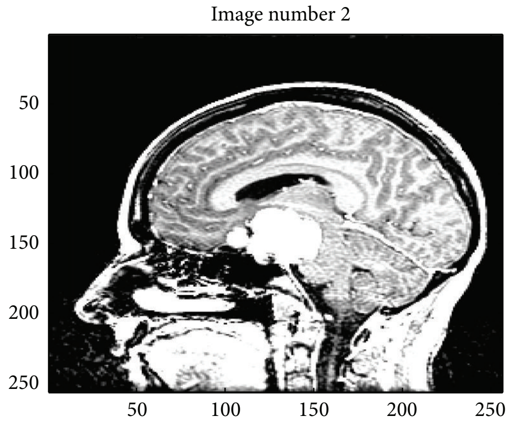

Figure 5: MRI brain case 2 SPL dataset.

case 2 SPL dataset and its volumetric feature extraction, respectively.

4.4. Brain Tissue and Tumor Classification Using ELM with RGSO. Hybrid ELM-RGSO is now invoked after the subselection process for efficient and accurate tumor classification. The details of the simulation and the performance of the classifier for the dataset are presented in this section. The weights and bias to the input layer and hidden layer of the ELM architecture are optimized using the proposed RGSO. The inputs (I) are 7 and the hidden neurons (H) are 5, for which the best optimal weight parameters are selected employing RGSO. The number of outputs specifies each tissue, that is, cerebrospinal fluid (CSF), white matter (WM), gray matter (GM), and tumor. Consequently, the structure of the single SLFN ELM neural network is 7-5-4.

Table 7 compares the ELM-RGSO simulation results with existing tested classifiers. The ELM-RGSO classification performance is appreciably good for 7 dimensional spaces. Thus, extreme learning machine outperforms the standard binary linear SVM and BPN for medical image classifier and proves better classification of healthy and tumor tissues. The comparison proves that the mean and standard deviation produced by volumetric feature extraction analysis is higher than the other approaches: connected threshold (CT), neighborhood connected (NC), and confidence connected (CC). The results are almost similar to the approach in [31] (original SPL data analysis). But it can be viewed that a significant difference exists in the computation time amongst the approaches.

Table 8: Average of true positive fraction (TPF), false positive fraction (FPF), and similarity index (SI) of different brain tissues and tumor tissues of SPL dataset with interobserver.

Classifier	CSF			WM			GM			TUMOR		
	TPF	FPF	SI	TPF	FPF	SI	TPF	FPF	SI	TPF	FPF	SI
Conventional kNN [5]	0.86	0.84	0.81	0.96	0.78	0.94	0.87	0.72	0.90	NA	NA	NA
Manual expert interobserver	NA	NA	0.86	NA	NA	0.96	NA	NA	0.94	NA	NA	0.97
IELM-PSO [28]	0.90	0.92	0.83	0.98	0.95	0.94	0.86	0.85	0.90	0.96	0.98	0.94
Proposed ELM-RGSO	0.93	0.92	0.85	0.98	0.97	0.96	0.92	0.91	0.97	0.97	0.98	0.97

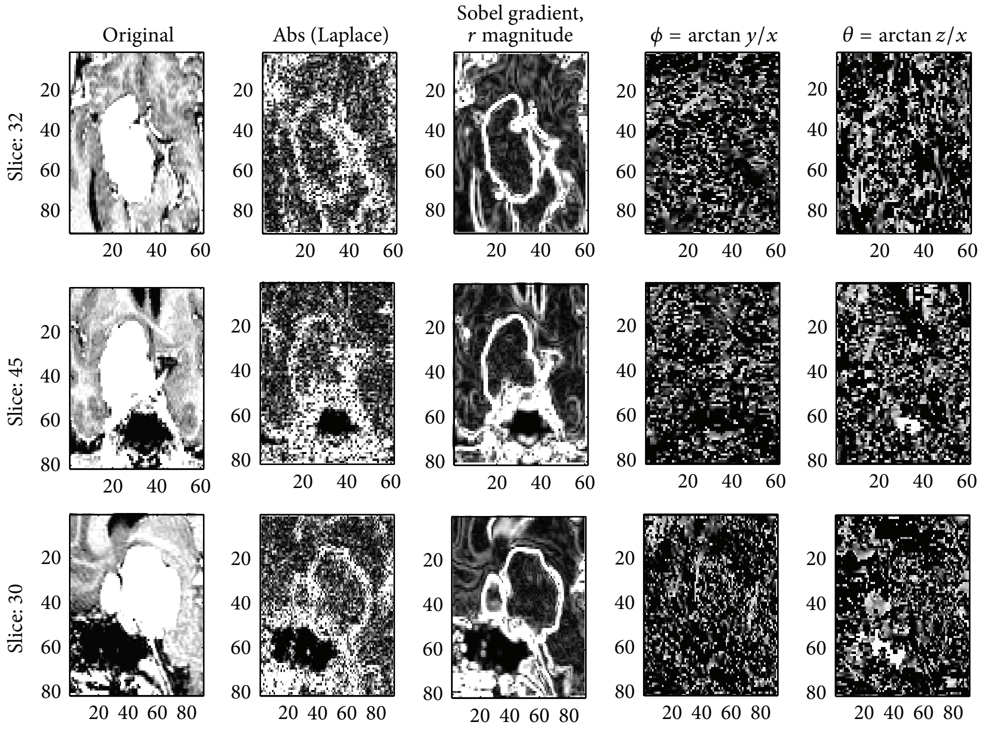

Figure 6: Volumetric feature extraction analysis (case 2).

The validation on the above tumor classification is evaluated quantitatively by calculation of the similarity index (SI) or dice coefficient between the automatic and the manual classifications. Consider

$$\mathrm{SI} = \frac{2\left(S_1 \cap S_2\right)}{\left(S_1 + S_2\right)}, \tag{20}$$

where S_j with $j = 1, 2$ is a segmented volume and $(S_1 \cap S_2)$ is the overlap of S_1 and S_2. The SI is also used as a measure for the interobserver variability. The true positive fraction (TPF), or sensitivity, and the specificity or false positive fractions (FPF) are also used for evaluation. Table 8 depicts the average value of the TPF, FPF, and SI of various classifiers. The proposed method showed high accuracy for all tissue classes and the SIs were close to the interobserver SI of the manual segmentations when compared to the state-of-the-art model using kNN classifier as given in [5]. However, for tissue types with less overlap, the SI measure shows a better distinction between the segmentation methods.

The ELM-RGSO classifier showed the highest overlap with the manual segmentation for all tissues. Table 9 depicts the average value based on the tumor segmentation methods based on feature extraction model and compared to the state-of-the-art techniques [31]. The proposed model using 3D feature extraction proves better and improves values on sensitivity and specificity for the 10 cases of SPL dataset. The algorithm segmented tumor and expert segmented tumor are shown in Figures 7 and 8.

4.5. Classification Analysis. The aim of this work is to classify normal and tumor images correctly. In the analyses of the images, each image is classified into one of two prognostic classes (normal and tumor image). Based on the features that have been extracted, the classifier separates these features. Combining both the publically available datasets (SPL and McConnell Brain Centre images), the dataset of 30 patients with texture measurements are well separated for the two classes; that is, the patients were carefully selected with respect to the average texture value of the measures.

Table 10 depicts the classification accuracy between various classifiers with proposed classification model. It is evident

Table 9: Comparison of various tumor classification methods.

SPL dataset	Traditional approach [32]		kNN approach [5]		(IELM approach) [28]		Proposed EM-RGSO	
	Sensitivity	Specificity	Sensitivity	Specificity	Sensitivity	Specificity	Sensitivity	Specificity
Case 1	100	100	100	100	99.8	100	100	100
Case 2	82.2	99.9	94.7	99.9	97.8	99.8	98.20	99.9
Case 3	93.6	100	79.8	100	94.56	99.8	95.61	100
Case 4	73.3	100	73.3	100	72.4	100	73.6	100
Case 5	79.3	99.9	83.7	100	80.6	99.6	81.2	100
Case 6	90.5	99.9	96	99.9	93.5	99.6	94.9	99.9
Case 7	94.6	99.9	94.7	99.9	94.2	99.8	94.1	99.6
Case 8	73.6	99.9	93.1	99.9	88.6	99.4	88.1	99.9
Case 9	91.8	99.9	99.3	99.8	96.8	99.5	97.92	99.9
Case 10	97.8	100	95.6	99.8	96.8	99.6	97.4	99.8
Average	87.67	99.9	91.02	99.9	91.506	99.71	92.103	99.9

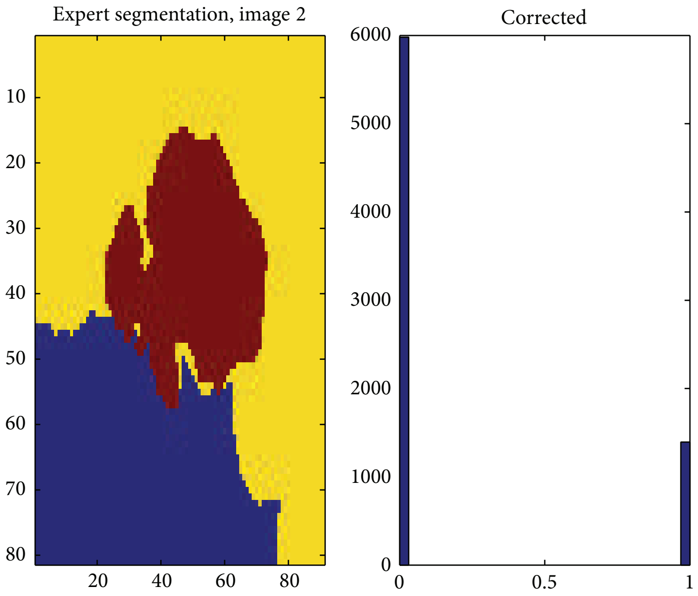

Figure 7: Tumor classified by expert.

Table 10: Accuracy results amongst various classifiers using 3D proposed feature extraction model.

Dataset	ELM	SVM	BPN	IELM-PSO	ELM-RGSO
Real time data (70 cases)	90.0%	86.6%	80.0%	96.6%	97.8%
	86.6%	83.3%	76.6%	93.3%	94.6%
	86.6%	80.0%	76.6%	90.0%	89.9%
	83.3%	80.0%	76.6%	86.6%	88.67%
	76.6%	70.0%	66.6%	83.3%	89.01%
SPL and MBC sets (30 cases)	92.8%	90.0%	85.7%	97.14%	97.63%
	92.8%	88.5%	82.85%	94.28%	96.59%
	91.4%	90.0%	88.5%	94.28%	96.00%
	90.0%	88.5%	85.7%	92.8%	94.61%
	90.0%	82.85%	78.5%	91.4%	93.89%

that images with low texture values are patients with good prognosis (normal brain) and vice versa. The images were then classified by ELM-RGSO classifier with leave-one-out cross-validation; that is, the classifier was trained with 19 patients and then the one patient not used in training was classified. This is rotated in such a way that all patients are used as a test set. In the above, the samples chosen are 70 cases of real time data and 30 cases of SPL and MBC set. The samples are converted into their respective pixel values and are inputted to the ELM model. All the samples were loaded sequentially; hence the delay process was within the permissible extent.

The proposed ELM-RGSO classifier, which classifies the patients to the class which is most probable, will result in one patient data being wrongly classified, leading to a correct

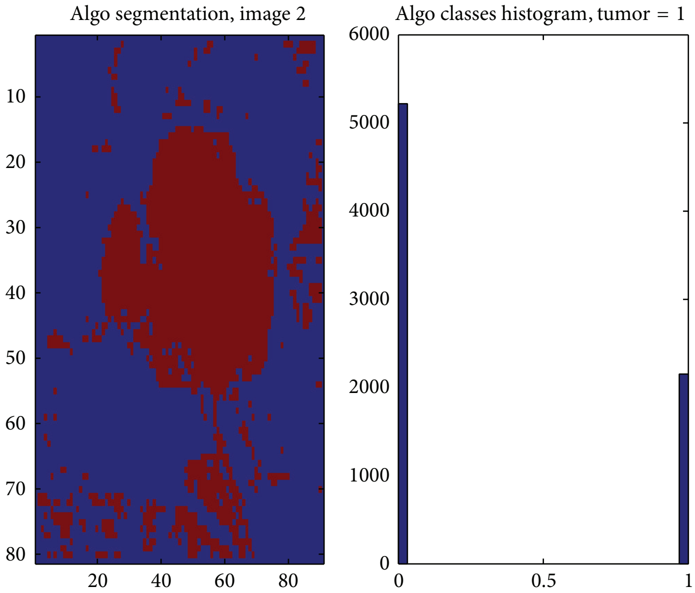

Figure 8: Classification of tumor using ELM-RGSO.

Table 11: Comparison result of ELM and IELM-PSO on SPL analysis.

Model	GA-ELM [33]	GA-IELM-PSO [28]	Proposed DE-ELM-RGSO
Norm	432.56	185.34	120.96
Condition	$24.825e + 5$	$4.665e + 5$	$3.366e + 5$
MSE error	0.966	0.991	0.932
Training efficiency %			
Mean	92.92	97.42	98.26
STD	6.05	5.92	6.25
Testing efficiency %			
Mean	91.27	96.85	97.12
STD	6.89	5.45	5.76
Hidden neurons	27	5	5
Accuracy %	92.8	97.14	98.29
Time			
Min.	3	3.5	3

classification rate for the 30 patients of 98.29%. Table 11 tabulates comparison of ELM, IELM-PSO classifier, and ELM-RGSO classifier.

5. Conclusion

The major issue in the development of pattern recognition towards brain tumor tissue classification is the formation of feature extraction analogy and the classifier. In this work, a refined GSO based extreme learning machine classifier is proposed to develop the brain tumor tissue classification along with the DE as feature subselector. Simulation result shows that the proposed volumetric feature extraction method works well and produces a minimum number of features with higher classification accuracy when compared with conventional 2D and 3D feature extraction methods.

The developed hybrid GSO and ELM [23] along with DE as feature subselector shows highest improvement in comparison with other literature studies in neural networks. On the whole, the significant finding of this work employing differential evolution, ELM, refined GSO, and volumetric analysis validates the correlation of magnetic resonance image measures as well as anatomical and histopathological parameters of research. Thus, a hybrid GSO and ELM classifier is developed in this research work for brain tumor tissue categorization in 3D MR images to reduce the computational cost and time prevailed in all earlier classifiers.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

The authors thank PSG IMSR and Hospitals for providing image data for carrying out this research work.

References

[1] M. M. Galloway, "Texture analysis using grey level run lengths," *Computer Graphics and Image Processing*, vol. 4, pp. 172–179, 1975.

[2] C. Philips, D. Li, D. Raicu, and J. Furst, "Directional invariance of co-occurrence matrices within the liver," in *Proceedings of the IEEE International Conference on Biocomputation, Bioinformatics, and Biomedical Technologies (BIOTECHNO '08)*, pp. 29–34, July 2008.

[3] E. I. Zacharaki, S. Wang, S. Chawla et al., "Classification of brain tumor type and grade using MRI texture and shape in a machine learning scheme," *Magnetic Resonance in Medicine*, vol. 62, no. 6, pp. 1609–1618, 2009.

[4] S. Suresh, S. Saraswathi, and N. Sundararajan, "Performance enhancement of extreme learning machine for multi-category sparse data classification problems," *Engineering Applications of Artificial Intelligence*, vol. 23, no. 7, pp. 1149–1157, 2010.

[5] R. de Boer, H. A. Vrooman, M. A. Ikram et al., "Accuracy and reproducibility study of automatic MRI brain tissue segmentation methods," *NeuroImage*, vol. 51, no. 3, pp. 1047–1056, 2010.

[6] L. Harrison, *Clinical applicability of MRI texture analysis [Ph.D. thesis]*, School of Medicine, University of Tampere, 2011.

[7] C. A. Cocosco, A. P. Zijdenbos, and A. C. Evans, "A fully automatic and robust brain MRI tissue classification method," *Medical Image Analysis*, vol. 7, no. 4, pp. 513–527, 2003.

[8] K. Valkealahti and E. Oja, "Reduced multidimensional co-occurrence histograms in texture classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 1, pp. 90–94, 1998.

[9] A. S. Kurani, D. H. Xu, and J. Frust, "Co-occurrence matrices for volumetric Data," in *Proceedings of the International Conference on Computer Graphics and Imaging*, Kauai, Hawaii, USA, 2004.

[10] D. Cobzas, N. Birkbeck, M. Schmidt, M. Jagersand, and A. Murtha, "3D variational brain tumor segmentation using a high dimensional feature set," in *Proceedings of the IEEE 11th International Conference on Computer Vision (ICCV '07)*, pp. 1–8, October 2007.

[11] A. Madabhushi, M. Feldman, D. Metaxas, D. Chute, and J. Tomaszewski, "A novel stochastic combination of 3D texture features for automated segmentation of prostatic adenocarcinoma from high resolution MRI," in *Medical Image Computing and Computer-Assisted Intervention—MICCAI 2003*, vol. 2878 of *Lecture Notes in Computer Science*, pp. 581–591, 2003.

[12] K. Jafari-Khouzani, H. Soltanian-Zadeh, and K. Elisevich, "Hippocampus volume and texture analysis for temporal lobe epilepsy," in *Proceedings of the IEEE International Conference on Electro Information Technology*, pp. 394–397, May 2006.

[13] R. Storn and K. Price, "Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.

[14] R. M. Haralick, K. Shanmugam, and I. Dinstein, "Textural features for image classification," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 3, no. 6, pp. 610–621, 1973.

[15] F. Tsai, C.-K. Chang, J.-Y. Rau, T.-H. Lin, and G.-R. Liu, "3D computation of gray level Co-occurrence in hyperspectral image cubes," in *Energy Minimization Methods in Computer Vision and Pattern Recognition*, vol. 4679 of *Lecture Notes in Computer Science*, pp. 429–440, 2007.

[16] A. Chu, C. M. Sehgal, and J. F. Greenleaf, "Use of gray value distribution of run lengths for texture analysis," *Pattern Recognition Letters*, vol. 11, no. 6, pp. 415–419, 1990.

[17] D. Xu, A. S. Kurani, J. D. Furst, and D. S. Raicu, "Run-length encoding for volumetric texture," in *Proceedings of the 4th IASTED International Conference on Visualization, Imaging, and Image Processing*, pp. 534–539, September 2004.

[18] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *The Journal of Machine Learning Research*, vol. 3, pp. 1157–1182, 2003.

[19] D. E. Goldberg and J. H. Holland, "Genetic algorithms and machine learning," *Machine Learning*, vol. 3, no. 2, pp. 95–99, 1988.

[20] W. Siedlecki and J. Sklansky, "A note on genetic algorithms for large-scale feature selection," *Pattern Recognition Letters*, vol. 10, no. 5, pp. 335–347, 1989.

[21] S. N. Sivanandam, S. Sumathi, and S. N. Deepa, *Introduction to Neural Networks Using Matlab 6.0*, Tata McGraw Hill, New Delhi, India, 2006.

[22] Q. Zhu, A. K. Qin, P. N. Suganthan, and G. Huang, "Evolutionary extreme learning machine," *Pattern Recognition*, vol. 38, no. 10, pp. 1759–1763, 2005.

[23] G. B. Huang, Q. Y. Zhu, and C. K. Siew, "Extreme learning machine: theory and applications," *Neurocomputing*, vol. 70, no. 1–3, pp. 489–501, 2006.

[24] S. Suresh, R. Venkatesh Babu, and H. J. Kim, "No-reference image quality assessment using modified extreme learning machine classifier," *Applied Soft Computing Journal*, vol. 9, no. 2, pp. 541–552, 2009.

[25] G. Huang, D. H. Wang, and Y. Lan, "Extreme learning machines: a survey," *International Journal of Machine Learning and Cybernetics*, vol. 2, no. 2, pp. 107–122, 2011.

[26] S. He, Q. H. Wu, and J. R. Saunders, "A novel group search optimizer inspired by animal Behavioral ecology," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '06)*, pp. 1272–1278, Sheraton Vancouver Wall Center, Vancouver, Canada, July 2006.

[27] S. He, Q. H. Wu, and J. R. Saunders, "Group search optimizer: an optimization algorithm inspired by animal searching behavior," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 5, pp. 973–990, 2009.

[28] B. Arunadevi and S. N. Deepa, "Brain tumor tissue categorization in 3D magnetic resonance images using improved PSO for extreme learning machine," *Progress in Electromagnetics Research B*, no. 49, pp. 31–54, 2013.

[29] F. Han, H.-F. Yao, and Q.-H. Ling, "An improved Extreme learning machine based on particle swarm optimization," in *Proceedings of the International Conference on Intelligent Computing*, pp. 699–704, 2012.

[30] J. Wu, A. Paul, Y. Xing et al., "Morphological dilation image coding with context weights prediction," *Signal Processing: Image Communication*, vol. 25, no. 10, pp. 717–728, 2010.

[31] M. R. Kaus, S. K. Warfield, A. Nabavi, P. M. Black, F. A. Jolesz, and R. Kikinis, "Automated segmentation of MR images of brain tumors," *Radiology*, vol. 218, no. 2, pp. 586–591, 2001.

[32] Y. Zhang and L. Wu, "An MR brain images classifier via principal component analysis and kernel support vector machine," *Progress in Electromagnetics Research*, vol. 130, pp. 369–388, 2012.

[33] S. Saraswathi, S. Suresh, N. Sundararajan, Z. Michael, N. Marit, and S. Saraswathi, "Performance enhancement of extreme learning machine for multi-category sparse cancer classification problems," *ACM Transactions on Computational Biology and Bioinformatics*, vol. 8, no. 2, pp. 452–463, 2011.

Advances in Operations Research
Advances in Decision Sciences
Journal of Applied Mathematics
Algebra
Journal of Probability and Statistics
The Scientific World Journal
International Journal of Differential Equations
International Journal of Combinatorics
Advances in Mathematical Physics
Journal of Complex Analysis
Journal of Mathematics
Mathematical Problems in Engineering
Abstract and Applied Analysis
Discrete Dynamics in Nature and Society
International Journal of Mathematics and Mathematical Sciences
Journal of Discrete Mathematics
Journal of Function Spaces
International Journal of Stochastic Analysis
Journal of Optimization