

BRAIN TUMOR TISSUE CATEGORIZATION IN 3D MAGNETIC RESONANCE IMAGES USING IMPROVED PSO FOR EXTREME LEARNING MACHINE

Baladhandapani Arunadevi* and Subramaniam N. Deepa

Dept. of EEE, Anna University, Regional Centre, Coimbatore, Jothipuram, Coimbatore 641047, India

Abstract—Magnetic Resonance Imaging (MRI) technique is one of the most useful diagnostic tools for human soft tissue analysis. Moreover, the brain anatomy features and internal tissue architecture of brain tumor are a complex task in case of 3-D anatomy. The additional spatial relationship in transverse, longitudinal planes and the coronal plane information has been proved to be helpful for clinical applications. This study extends the computation of gray level co-occurrence matrix (GLCM) and Run length matrix (RLM) to a three-dimensional form for feature extraction. The sub-selection of rich optimal bank of features to model a classifier is achieved with custom Genetic Algorithm design. An improved Extreme Learning Machine (ELM) classifier algorithm is explored, for training single hidden layer artificial neural network, integrating an enhanced swarm-based method in optimization of the best parameters (input-weights, bias, norm and hidden neurons), enhancing generalization and conditioning of the algorithm. The method is modeled for automatic brain tissue and pathological tumor classification and segmentation of 3D MRI tumor images. The method proposed demonstrates good generalization capability from the best individuals obtained in the learning phase to handle sparse image data on publically available benchmark dataset and real time data sets.

Received 2 January 2013, Accepted 12 February 2013, Scheduled 14 February 2013

* Corresponding author: Baladhandapani Arunadevi (arunaamurthy@gmail.com).

1. INTRODUCTION

Brain tumors are the second most common malignancy among children. About 7% of the reported brain and Central Nervous System (CNS) tumors occurred in children ages 0–19 years. Brain tumors account for 85% to 90% of all primary CNS tumors. Advances in Neuro-imaging techniques, supplement to clinical and radiological findings, for pre-operative planning. Studies revealed that many neuro-imaging approach models require precise recognition of the brain in MRI head [29]. The need for defined anatomic three-dimensional (3D) models substantially improves spatial distribution concerning the relationships of critical structures (e.g., functionally significant cortical areas, vascular structures) and disease [8, 9, 18].

Texture analysis on images are native and complex visual patterns that reproduce the data of gray level statistics, anatomical intensity variations, texture, spatial relationships, shape, structure and so on. Image texture analysis aims to interpret and understand these real-world visual patterns, which involves the study of methods broadly used in image filtering, normalization, classification, segmentation, labeling, synthesis and shape from texture. Texture classification involves extracting features from different texture images to build a classifier. It determines to which of a finite number of physically defined classes (such as normal and abnormal tissue) a homogeneous texture region belongs [1]. The classifier is then used to classify new instances of texture images. The textural properties of spatial patterns on digital images have been successfully applied to many practical vision systems, such as the classification of images to analyze diagnosis tissues for dementia, tumors, hyper spectral satellite images for remote sensing, content based retrieval, detection of defects in industrial surface inspection, and so on.

This research study on absolute exploration of three-dimensional (3D) texture features in the volumetric data sets requires extension of conventional 2D GLCM and run length texture computation into a 3D form for better texture feature analysis. Genetic Algorithm (GA) selects relevant elements in feature selection method. Classification is performed with ELM, with Improved Particle Swarm Optimization (IPSO) technique to select the best parameters (Input-weights, Bias, Norm, Hidden neurons) for better generalization and conditioning of the classifier for brain tissue and tumor pathology tissue characterization as White Matter (WM), Gray Matter (GM), Cerebrospinal Fluid (CSF) and Tumor. Figure 1 depicts framework of the proposed model.

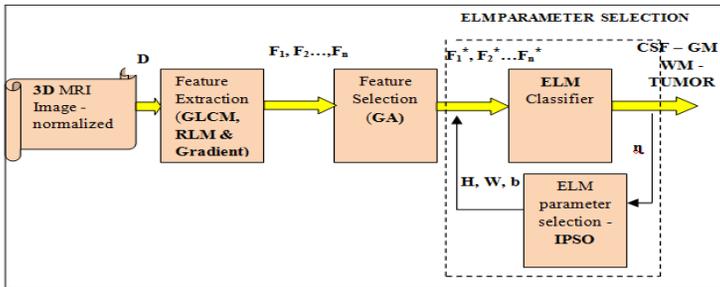


Figure 1. System framework of the model proposed.

2. VOLUMETRIC TEXTURE ANALYSIS ON 3D MRI IMAGES

2.1. Literature Review

Several techniques have been developed for texture feature extraction of biomedical images [7]. Identified texture analysis is categorized as structural, model-based, statistical and spectral transform, according to the means employed to evaluate the interrelationships of the pixels. Statistical models involve computation of spatial grey level distribution of grey values. Model based methods govern an underlying process of the arrangement of pixel used to extract the perceived qualities of the texture. Spectral transform methods involve application of filters to compute features. Structural methods represent texture by use of well-defined primitives that make up a texture, followed by computation of statistical properties of the primitives using geometric or syntactic rules.

Cascade of feature extraction involves data, pixel, edge, texture and region levels. The low-level features are used to represent medical images. Texture-based features mainly capture the granularity and repetitive patterns in the pixel distribution. Identification of the local description (neighborhood dependent modeling) to a global and vice versa directs to co-occurrence matrix classifying brain MRI tissues significantly. No single best feature has been detected for any given problem. The criterion for evaluating the performance of texture feature extraction methods is the classification accuracy. The development and analysis of low-level feature characteristics have been extensively studied earlier. Amongst the vastly employed approaches are gray level co-occurrence matrix (GLCM) [23], Run-length Matrix (RLM) [16], histogram of oriented gradients (HOG) [11], scale-invariant feature transform(SIFT) [30], local binary

patterns(LBP) [32]. Furthermore, several works comparing different feature descriptors and filters are reviewed in literature [1, 25, 34].

2.2. Proposed Feature Extraction of 3D Gray Level Co-occurrence Matrix-run Length Matrix and Gradient Element

Optimal feature vector set is characterized to aggregate in requisites of geometric, spectral, image intensity and texture. The statistical approach adopted here to extract texture parameters from the MR images was based on the GLCMs and Gray run Length [17] and gradient magnitude. Volumetric equivalents of these features were computed in the present study. In a 3D volume, adjacency and consecutiveness can occur in each of 13 directions as in Figure 2, generating 13 gray-level co-occurrence and run-length matrices [4].

The grey-level run length method (GLRLM) allow to capture the coarseness characteristic is based on the analysis of higher-order statistical information [16]. In a 3D image, for a preferred slice, 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. For 3D run length encoding the size of the matrix P is n by k , where n is the maximum gray level n in the MRI image and k is equal to the possible maximum run length in corresponding image. In a 3D discrete space, the directions are selected by linking a voxel to each of its nearest 26 ($= 3 * 9 - 1$) neighbours respectively, leading to 13 different displacements from a total of 26 possible displacements. All slices are

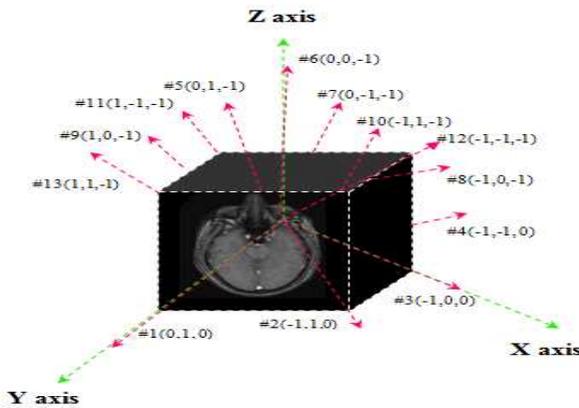


Figure 2. GLCM and Run length direction for volumetric 3D Image.

Table 1. The list of 17 3D GLCM and GRLM features extracted.

Group Features
Co-occurrence
Angular Second Moment, Entropy, Contrast, Homogeneity
Run length Matrix
Short Run Emphasis (SRE), Long Run Emphasis (LRE), High Gray-level Run Emphasis (HGRE), Low Gray-level Run Emphasis (LGRE), Short Run Low Gray-level Emphasis (SRLGE), Short Run High Gray-level Emphasis (SRHGE), Long Run Low Gray-level Emphasis (LRLGE), Long Run High Gray-level Emphasis (LRHGE), Run-length Non Uniformity (RLNU), Grey-level Non-Uniformity (GLNU), Run Percentage (RPC).
Gradient Vector and orientation

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. 11 texture features are computed to characterize the texture for each sub-region [12].

Spatial Gray Level Co-occurrence Matrices (GLCM) captures the spatial dependence of gray-level values across multiple slices. The matrix is calculated for 0, 45, 90 and 135 degrees for θ and a distance scale of 1. This paper describes a 45 spatial-degree resolution of directions [26]. In this approach, four spatial distances of displacement 1, 2, 4, and 8 voxels and thirteen directions are selected, resulting in 52 displacement vectors and co-occurrence matrices. Hence, four Haralick statistical measures quantifying the spatial dependence of gray-level values are energy, entropy, contrast and homogeneity, are computed from each matrix, giving a feature vector of 3D texture 4 (measures) * 52 (matrices) = 208 components. Other feature include 3D gray level gradient based feature. These features amplify the significant differences between class areas. The optimal features with 17 texture descriptors were calculated. These features as in Table 1 are computed as part of the segmentation process.

2.3. Feature Sub-selection

Contribution of sub-selection in image analysis rely on choosing the optimal feature subset vector, from an existing feature extracted set to describe the target conceptions of machine learning in classification. The objective is to trace for best minimum subset in the original

Table 2. Parameters of genetic algorithm for feature selection.

Parameters	Description	Value
P_m	Mutation Probability	0.02
P_c	Cross-over probability	0.1
M_g	Maximum No. of generations	50
N	Population size	50

element set, relatively than transforming the data to an entirely new set of dimensions. All extracted features with pooled texture measures was analyzed as possibly highly correlated features. This process aids in removing any bias towards certain features which might afterwards affect the classification procedure. In spite of all texture measure likely to characterize the examined texture from a different perspective, some features extracted occur to behave similar. In addition, alleviation of the curse of dimensionality of texture features decreases the computational time and memory required. Identification of features that is correlated with or predictive of the class label is the criterion.

Studies on feature selection introduces three approaches: the filter, wrapper and embedded approach. Selection of the most pertinent elements by evaluating random subsets of the original features is called the wrapper method in machine learning. The wrapper approach is a classifier dependent feature selection algorithm and uses a specific learning model, like decision trees, and ANN's, to evaluate the feature subset via the performance of training. Relevant features are chosen with the use of the learning algorithm itself and hence generally result in higher learning performance, e.g., accuracy. Objective of this search in sub-set includes maximization of this criterion. Feature sub-selection model typically incorporates a search strategy for exploring the space of feature subsets [24]. Genetic algorithm(GA), a stochastic adaptive problem solving search technique, which tends to find approximate solutions to optimization and search problems [38] is employed here. The genetic algorithm reduces the dimensionality into 6 most decisive very few features out of 27 features extracted. The parameters of custom genetic algorithm design are tabulated in Table 2.

2.4. Pattern Recognition Technique Using ELM-IPSO for Brain Tumor and Tissue Classification

The sub-selected features describe the target conceptions of machine learning in classification. Literature studies indicate that ANN are a

particularly good choice for pattern recognition and classification of MR images because their generalization properties require the labeling of only few training points, and they produce results faster than other traditional classifiers [13]. The advantage is that it does not rely on any assumption about the underlying probability density functions, thus possibly improving the results when the data significantly depart from normality. The main limitation of Support Vector Machines (SVM) is of taking longer time to select [2, 9, 41]. The current limitations are mostly in the preprocessing speed and the interface for brain tumor analysis.

To obtain reasonable outcome, a complex prior model or a large amount of training data is required, thus restricting the range of application by the domain of training algorithms. Traditional training of networks based on gradient-descent algorithms tend to generally slower and get stuck in local minima. These problems have been prevailed by Extreme Learning Machine algorithm recently proposed by [20], suitable for training single layer feed-forward neural networks.

2.4.1. Extreme Learning Machine (ELM)

Extreme Learning Machine (ELM) [20] is a single-hidden layer feed forward neural network (SLFNN) which randomly selects input weights and hidden neuron biases without training. The output weights are analytically determined using the norm least-square solution and Moore-Penrose inverse of a general linear system, thus allowing a significant training time reduction. Studies on ELM [19–22] proved enhanced performance in comparison to other classifiers for larger training samples. The sigmoid activation function was used and the effect of the number of neurons in the hidden layer by using different ratios of the number of features in the training and test data was explored. This simple learning algorithm is comparable to traditional gradient-decent based algorithms in terms of Root Mean Square Error (RMSE) and classification rate for brain tumor classification problem.

The SLFN can have P hidden nodes, activation function $\phi(x)$ and it can be approximated through the given N pairs of input/output values, namely, using universal approximation. Let the given training set $N = \{(\mathbf{X}_i, \mathbf{T}_i)\}$, $i = 1, 2, \dots, N$, where the training sample $X_i = [x_{i1}, x_{i2}, \dots, x_{in}]^T \in \mathbf{R}^n$ is a n -dimensional attribute of the set sample i and corresponding target value $T_i = [t_{i1}, t_{i2}, \dots, t_{im}]^T \in \mathbf{R}^m$ where m is the coded label considered. The sample x_i is assigned to the coded class label. As a multi-class function, the samples are to be considered as m distinct classes for identification. The SLFNs determine the functional association between random samples and its

respective class label. The classifier function $T = F(X)$, gives the necessary data on the probability of predicting the class label with the desired accuracy.

Huang et al. [20] evaluated that SLFNs with P hidden neurons and activation function $\phi(x)$ approximate any function to desired level of accuracy using universal approximation theory. It evolved that there existed optimal weight for approximating the function for bounded inputs in the layers of network. The multi-classification ELM is modelled as follows. Let W be $P \times n$ input weights, b be $P \times 1$ threshold values for each hidden neurons and β be $m \times P$ output weights. With m distinct classes, the output \hat{T} of the ELM network is as:

$$t_k = \sum_{i=1}^P \beta_{ik} \cdot \phi_i \cdot (w_i, b, X_i), \quad k = 1, \dots, m \quad (1)$$

where $\phi_i(\cdot)$ is the activation function of the i -th hidden neuron. The sigmoid function is defined as:

$$\phi_i = (W, b, X_i) = \tanh \left(b_i + \sum_{k=1}^N v_{ik} \cdot x_{jk} \right), \quad i = 1, \dots, P \quad (2)$$

Equation (1) can be rewritten in matrix form as

$$\mathbf{H}\boldsymbol{\beta} = T \quad (3)$$

where \mathbf{H} is an $P \times N$ dimension hidden layer output matrix defined as

$$\mathbf{H}(W, b, X_i) = \begin{bmatrix} \phi_1(W, b, X_1) & \phi(W, b, X_2) & \dots & \phi(W, b, X_N) \\ \vdots & \vdots & \vdots & \vdots \\ \phi_P(W, b, X_1) & \phi_P(W, b, X_2) & \dots & \phi_P(W, b, X_N) \end{bmatrix} \quad (4)$$

Results in [19] suggested an alternate way to train a SLFN by finding a least square solution $\hat{\boldsymbol{\beta}}$ of the linear system represented by Equation (3). The input weights (W), threshold (b) are arbitrarily chosen for a given number of hidden neurons for the ELM algorithm. Assuming the predicted output \hat{T} is equal to the coded labels T , the output weights are estimated using the unique minimum norm least-square (LS) solution, modelled as

$$\hat{\boldsymbol{\beta}} = \mathbf{TH}^\dagger \quad (5)$$

where \mathbf{H}^\dagger is the Moore-Penrose generalized pseudo-inverse of hidden layer output matrix. As analyzed in [20], ELM using such MP inverse method tends to obtain good generalization performance with dramatically increased learning speed. The summarization of the ELM algorithm can be as:

Given a training set $N = \{(\mathbf{x}_i, \mathbf{t}_i) | \mathbf{x}_i \in \mathbf{R}^n, \mathbf{t}_i \in \mathbf{R}^m, i = 1, \dots, N\}$, kernel function $\phi(x)$, and hidden neuron P .

- Step 1: Select suitable activation function and number of hidden neurons \tilde{N} for the given problem.
- Step 2: Assign arbitrary input weight W and threshold b .
- Step 3: Calculate the output matrix β at the hidden layer using Equation (5).
- Step 4: Use the calculated weights (W, b, β) for estimating class label.

2.4.2. Drawbacks of Conventional Extreme Learning Machine (ELM)

Several studies [20, 21, 35–37] have been emphasized on the accuracy of solutions obtained by ELM, whereas in general numerical stability is ignored. Hence random selection of initial parameters (W, b, H) influence the performance of the classifier. Training of ELM with large hidden neurons usually constitutes an ill-posed problem. Hence, the results achieved by ELM may be receptive to data perturbation and become a poor evaluation to the truth. Appropriate selection of input weights, threshold values and hidden neurons significantly influence the performance [37]. Hence the need in improvement to conventional ELM model is necessary. The objective is to find the optimal number of hidden neurons H and the corresponding W and b values, to analytically calculate the V such that the generalization ability of the ELM network can be improved. The best weights and threshold values for the ELM is computed and tuned using Improved Particle Swarm Optimization (IPSO) technique.

2.4.3. Particle Swarm Optimization (PSO)

Particle Swarm Optimization developed by [33], is an evolutionary computation technique that was developed through simulation of simplified social behaviour on swarms such as fish schooling and bird flocking. The major advantage of it is fewer computation times and less memory. Being an optimization method, the objective is to find the global optimum of a real-valued function (fitness function) defined in a given search space. Each individual is termed a “particle”, and is subject to a movement across the multidimensional search space. Each particle’s movement is the composition of an initial random velocity and two randomly weighted influences: individuality, the tendency to return to the particle’s best previous position, and sociality, the tendency to move towards the neighborhood’s best previous position for the entire population. Each particle updates its own velocity and position based on the best experience of its own and the entire

population. The model optimizes certain objective function for its performance.

Assuming N particles in the swarm in the D dimensional search space, the i -th particle is represented as:

$$X_i = (x_{i1}, x_{i2}, \dots, x_{iD}) \quad (6)$$

The best precious position giving the best fitness value is given by

$$P_i = (p_{i1}, p_{i2}, \dots, p_{iD}) \quad (7)$$

The rate of change (velocity) of the particle is given by

$$V_i = (v_{i1}, v_{i2}, \dots, v_{iD}) \quad (8)$$

The PSO algorithm consists of following steps :

Step 1: Initially, the positions and velocities of the initial swarm of particles are randomly generated using upper and lower bounds on the design variables values.

Step 2: Velocity update — Update the velocities of all particles at time k (current iteration) using the particles objective or fitness values which are functions of the particles current positions in the design space at time k . At each iteration, the velocities of all particles are updated according to,

$$v_i(k+1) = wv_i(k) + c_1\gamma_1(p_{i,best} - p_i) + c_2\gamma_2(g_{i,best} - p_i) \quad (9)$$

where p_i and v_i are the position and velocity of particle i , respectively; $p_{i,best}$ and $g_{i,best}$ is the position with the 'best' objective value found so far by particle i and the entire population respectively; w is the inertia weight; γ_1 and γ_2 are random variables in the distributed range $[0, 1]$; c_1 and c_2 are acceleration factors controlling the related weighting of corresponding terms with constant positive values.

Step 3: Position update — The Position of each particle is updated using its velocity vector as follows:

$$p_i(k+1) = x_i + v_i(k+1) \quad (10)$$

Step 4: Memory update — Update $p_{i,best}$ and $g_{i,best}$ when condition is met,

$$\begin{aligned} p_{i,best} &= p_i & \text{if } f(p_i) > f(p_{i,best}) \\ g_{i,best} &= g_i & \text{if } f(g_i) > f(g_{i,best}) \end{aligned} \quad (11)$$

Step 5: Stopping Criteria — The algorithm repeats steps 2 to 4 until certain stopping conditions are met, i.e., fitness function and maximum number of iterations. Once stopped, the algorithm reports the values of g_{best} and $f(g_{best})$ as its solution.

The learning mean square error (*MSE*) can be calculated as

$$MSE = \frac{1}{N} \sum_{i=1}^N E_i^2 = \frac{1}{N} \sum_{i=1}^N (y_k^i - d_k^i)^2 \quad (12)$$

where N is the number of training samples, and the term y_k and d_k is the error of the actual output and target output of the k th output neuron of i th sample. The fitness function $f(x)$ is defined by the MSE. PSO was used to find the best optimal weights W and bias b values so that the fitness reaches the minimum to achieves better generalization performance [36]. In addition to the c_1 and c_2 parameters, the implementation of the original algorithm also requires to place a limit on the velocity (v_{\max}). Improvements on PSO had been in research, which is problem dependent. An evolutionary ELM (E-ELM) approach was proposed by [39], which used the differential evolutionary algorithm to select the input weights and Moore-Penrose (MP) generalized inverse to analytically determine the output weights, achieved good generalization performance. Other models of evolutionary ELM found in the survey studies include, particle swarm optimization (PSO) [14, 40] to optimize the input weights and hidden biases of the SLFN to improve ELM, solving some prediction problems.

Successful implementation of ELM-PSO by [36] was applied for gene classification on imbalance data. Improved PSO based on adaptive PSO was proposed by [14] to select the input weights and hidden biases of the SLFN, and the MP generalized inverse is used to analytically calculate the output weights. The algorithm optimized the input weights and hidden biases according to the root mean squared error on validation set and the norm of the output weights, obtaining good performance with more compact and well-conditioned networks than other ELMs. Another promising modified PSO for image registration in multi-dimensional space, which employs benefits from the Gaussian and the uniform distribution, when updating the velocity equation in the PSO algorithm was proposed by [27].

The inertial weight constant and velocity particle movement is varied in similar studies, depending on application. The introduction of β as a constriction coefficient introduced by [5] controls the velocity of particles and ensures convergence. The constriction co-efficient is modeled as:

$$\beta = \frac{2\kappa}{\left| 2 - \psi - \sqrt{(\psi(\psi - 4))} \right|} \quad (13)$$

where $\kappa \in [0, 1]$ and $\psi = c_1 + c_2$. A smaller value of κ results in faster convergence with local exploitation, while larger values results in slow

convergence. κ is often set to 1 which is successful here as in [5].

2.4.4. Improved Extreme Learning Machine (IPSO-ELM)

This research incorporates an approach named IPSO-ELM combining an improved PSO with ELM. This modified ELM with the improved PSO [14, 27] to select the input weights to enhance the generalization performance and the conditioning of the SLFN. The details of the proposed method are as follows:

Step 1: Initialize a population array of swarm particles with of a set of input weights and hidden biases: $P_i = [W_{11}, W_{12}, \dots, W_{1n}, \dots, W_{21}, W_{22}, \dots, W_{2n}, \dots, W_{H1}, W_{H2}, \dots, W_{Hn}, b_1, b_2, \dots, b_H]$ with random initialized within the range of $[-1, 1]$ on D dimensions in the search space.

Step 2: For each swarm particle, the corresponding output weights are computed according to ELM as in Equation (5).

Step 3: Then the fitness of each particle $f(x)$ is evaluated as in Equation (16). In order to avoid overfitting of the SLFN, the fitness of each particle is adopted as the root mean squared error (RMSE) on the validation set only instead of the whole training set as in [40].

$$p_{i,best} = \begin{cases} p_i & (f(p_{i,best}) - f(p_i) > \eta f(p_{i,best})) \text{ or } (f(p_{i,best}) \\ & - f(p_i) < \eta f(p_{i,best}) \text{ and } \|wo_{p_i}\| < \|wo_{p_{i,best}}\|) \\ P_{i,best} & \text{else} \end{cases} \quad (14)$$

$$g_{i,best} = \begin{cases} p_i & (f(g_{i,best}) - f(p_i) > \eta f(g_{i,best})) \text{ or } (f(g_{i,best}) \\ & - f(p_i) < \eta f(g_{i,best}) \text{ and } \|wo_{p_i}\| < \|wo_{g_{i,best}}\|) \\ g_{i,best} & \text{else} \end{cases} \quad (15)$$

where $f(P_i)$, $f(P_{i,best})$ and $f(g_{i,best})$ are the corresponding fitness for the i -th particle, the best position of the i -th particle and global best position of all particles, respectively. wo_{P_i} , $wo_{P_{i,best}}$ and $wo_{g_{i,best}}$ are the corresponding output weights obtained by MP generalized inverse when the input weights are set as the i -th particle, the best position of the i -th particle and global best position of all particles, respectively. The parameter $\eta > 0$ is tolerance rate.

Step 4: Velocity update — Update the velocities of all particles at time k (current iteration) using the particles objective or fitness values which are functions of the particles current positions in the design space at time k . At each iteration, the velocities of all particles are updated as:

$$v_i(k+1) = \beta [v_i(k) + c_1 \gamma_1 (p_{i,best} - p_i) + c_2 \gamma_2 (g_{i,best} - p_i)] \quad (16)$$

The two parameters are often set to $c_1 = c_2 = 2.05$ [27], which scales the cognitive and social components equally. Difficulty to

Table 3. Parameters of IPSO for selection of input weights and bias in ELM.

Parameters	Description	Value
N	No. of particles	50
c_1, c_2	Acceleration learning co-efficient	2.5, 2.5
γ_1, γ_2	Random numbers that help the ability of stochastic searching	0.5
β	Constriction co-efficient	0.52
κ	Influences the speed of convergence	1
Topology	Default Ring Topology	-

know priori the individual weights of these two components, it seems logical to weigh them equally. β is the constriction coefficient as in Equation (13).

Step 5: Position update — The position of each particle is updated using velocity vector as follows:

$$p_i(k+1) = x_i + v_i(k+1) \quad (17)$$

Step 6: Memory update — Update $p_{i,best}$ and $g_{i,best}$ when condition is met and new population is generated.

Step 7: Stopping Criteria — The algorithm repeats steps 3 to 6 until certain criteria are met, along with hard threshold value as maximum number of iterations. Once stopped, the algorithm reports values of g_{best} and $f(g_{best})$ as its solution.

Thus improved PSO (IPSO) with ELM, finds the best optimal weights W and bias b so that the fitness reaches the minimum to achieves better generalization performance, with minimum number of hidden neurons. It makes advantage of both ELM and PSO. In the process of selecting the input weights, the modified PSO consider not only the RMSE on validation set but also the norm of the output weights [14]. The parameters for the best optimal weights W , bias b and hidden neurons are as is Table 3.

3. EXPERIMENT RESULTS

In order to evaluate the resilience of ELM-IPSOs to both Surgical Planning Analysis (SPL) bench mark dataset (10) and real time datasets (20) of brain tumor images, 17 features were extracted based on texture analysis, and compared with Back Propagation Network (BPN), SVM and ELM classifiers. The experiments were carried using

the neural network toolbox of Matlab 2012a. The SPL benchmark datasets included T1-weighted both contrast and non-contrast 3D MR brain images in axial plane with histology tumours as astrocytoma, low grade glioma and meningioma. The real time datasets of 20 volumes were obtained from PSG IMSR & Hospitals, diagnosed of glioma, meningioma and metastasis, of T1 contrast images, processed in 1.5 Tesla Siemens Unit. The classification of each voxel in the image to be segmented and computation of the consequent classification probability is performed using ELM classifier. Validation of the classifiers is done by creating classifiers using only part of the expert defined training samples, and then applying the classifiers to those excluded samples to determine how well the classification agrees with the expert's interpretation [15].

The approach employed a leave-one-case-out validation approach with different feature vector sets to train and test the classifier. In the leave-one-case-out validation, a case is first left out and reserved for testing, and the remaining cases are used for training the classifier. The trained classifier is finally tested on the reserved case. The ELM-IPSO is repeated multiple times so that each case is left out once for testing. Using the best optimal features vector and ELM parameters (H, W, b), the classifier computes the brain tissue and tissue categorization. The computation of the segmented brain tumor image includes optimizing a corresponding object function for segmentation iteratively based on the initial label information on thresholding and classifying the feature vector of each voxel using ELM-IPSO classification method.

Quantitative results are calculated using classification rates, sensitivity and specificity values, computed on some of the expert defined samples excluded from training, to provide a measure of degree of certainty in identifying the tumor and the healthy tissue. Classification rate is the percentage of correctly classified voxels with respect to the total number of training voxels available for that class. The sensitivity and specificity show the percentage of correctly classified positive and negative samples respectively. Sensitivity = $TP * 100 / (TP + FN)$ and Specificity = $TN * 100 / (FP + TN)$. Tissue segmentation is obtained by assigning the voxel to the class having the highest discriminant value, among the four classes.

Figure 5 illustrates the original MRI, features extracted, algorithm and expert segmented image and segmented tumor portion for case 3 in SPL data set. The validation results are tabulated in Table 5. Hence a discriminatively-trained supervised model based on GA-ELM-IPSO gives better results than SVM and BPN. Based on the SPL dataset, the tumor segmentation results as in Table 7, the mean is approximately same as in the original SPL dataset. The standard deviation has

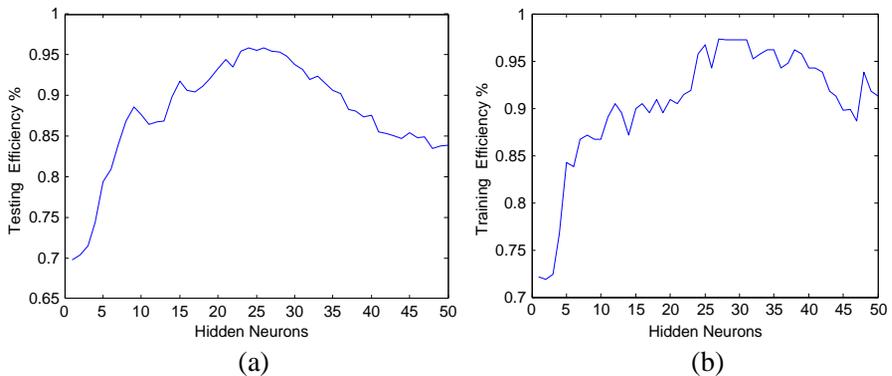


Figure 3. Effect of initial parameter selection using incremental learning selection of weights and bias. (a) Testing efficiency of standard ELM. (b) Training efficiency of standard ELM.

improved results than neighbourhood connection segmentation.

The SLFN can learn N distinct samples (x_i, t_i) with negligibly small error. For N arbitrary distinct samples (x_i, t_i) , standard SLFN's with N hidden neurons and activation function $\phi(x)$ are mathematically modelled as in Equation (1). To formulate the number of hidden neurons for brain tumor classification with incremental learning, when N hidden neurons are considered, the generalization error is prone to 70% training efficiency and 69% testing efficiency as in Figure 3 (random selection of input weights and bias). Hence for small sets, the PSO-ELM approach shows considerable results as proved by [36]. Number of hidden neurons, proper selection of input weights and bias in ELM has significance impact on sparse data classification problems which have few number of training samples [35]. The real time data set achieved 94.65% classification accuracy. The accuracy is fair when compared to benchmark dataset, but reasons out due to issue on anisotropy of the live patient data acquired. The bench mark data set (10) achieved 98.05% accuracy. Tables denote values for SPL dataset.

4. DISCUSSION

The research experiments a multi-parametric framework of profile for brain tumor tissue recognition. The GLCM co-occurrence, run length matrix and gradients features in three-dimensional extension, exemplify the tumor categorization augmenting spatial dimensionality relationship, though prohibitively expensive. The dimensions of

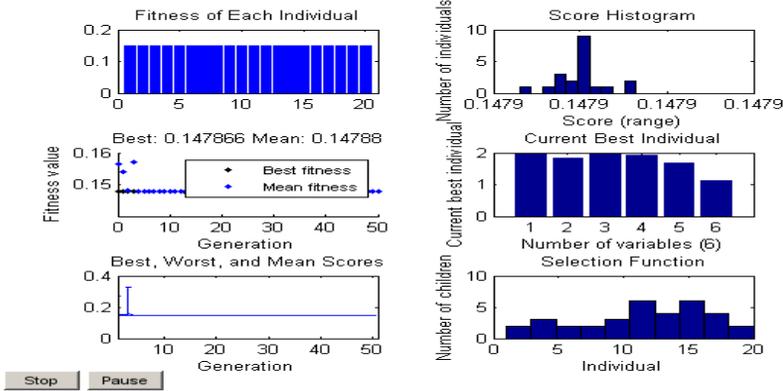


Figure 4. Feature sub-selection using genetic algorithm.

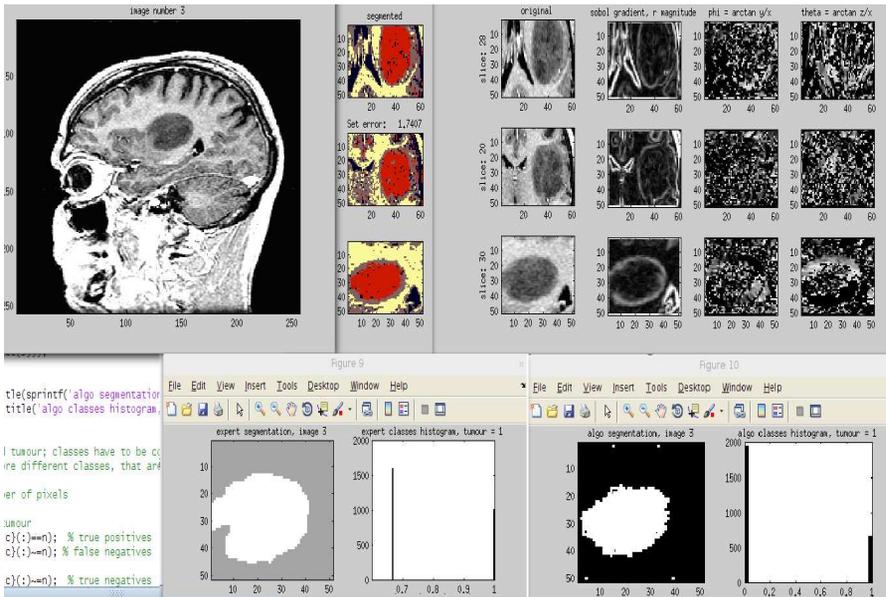


Figure 5. SPL data case 3 image analysis.

extracted features were for a distance of $d = 1-55$ (maximum distance), of 4250 sample texture measures. 17 texture features of rotation invariant, were evaluated to carry out the most informative data. The findings of this preliminary research revealed further insight towards the 3D texture analysis of tumor classification. However, this still

Table 4. Comparative analysis of the classifiers (SPL data).

Classifiers	Sensitivity %	Specificity %	Accuracy w/o Feature sub-selection %	Accuracy with Feature sub-selection %
BPN	82.30	88.23	76.19	82.25
GA-SVM	90.21	93.45	92	91.2
GA-ELM	94.77	97.98	92.8	93.25
GA-ELM-IPSO	95.90	98.88	94.88	98.05

Table 5. Validation classification result (SPL data) on tumor detection for GA-ELM-PSO approach.

Parameter	CSF	WM	GM	Tumor
Sensitivity	89.06%	78.31%	72.04%	94.17%
Specificity	96.67%	84.09%	82.32%	96.4%

Table 6. Simulation results for ELM-IPSO.

Model	Norm	Condition	MSE Error	Training Efficiency % Mean STD		Testing Efficiency % Mean STD		hidden neurons	Accuracy %	Time ^a min
BPN	/	/	0.866	83.9	8.02	81.2	11.12	34	82.25	15
GA-SVM	/	/	0.909	93.9	8.02	88.2	7.12	8 ^b	91.2	10–12
GA-ELM	432.56	24.825e+5	0.966	92.92	6.05	91.27	6.89	27	93.2	3
GA-ELM-IPSO	185.34	4.665e+5	0.991	97.42	5.92	96.85	5.45	5	98.15	3–5

^aComputational time (includes hidden neurons, feature extraction, selection & classification with PSO)

^bNumber of support vectors

results in high computation cost. Thus, genetic Algorithm is used to further select the features on another level. Only 6 decisive features preserve 94.5% of variance as in Figure 4. The feature vector was evaluated on every dataset, with ELM classifier, which was certain for simplicity and speed. Thus, the inputs (I) is 6 and the hidden neurons (H) is determined as 5 with the help of best parameters been selected with improved PSO. Consequently, the structure of the single SLFN ELM neural network is 6-5-4. Here, traditional ELM classifier, is used

as the comparative methods along with well known SVM and BPN networks. Tables 4 and 6 compares the ELM-IPSO simulation results with existing tested classifiers.

Intensity discrepancy of brain overlaps between dissimilar types of tissues. This overlies occurs for any intensity based discrete voxel labeling approach. It reasons out due to the finite spatial distribution of the image acquisition: voxels at the boundary between tissue types have more than one tissue contributing to the measured signal (partial volume effect). The GA-ELM-IPSO texture based model, is relatively informative on the intensity of pure tissue voxels. Since the boundary voxels are located in high gradient areas of the MR image, which tends to be limited by the approach proposed by [6], the proposed approach with the incorporation of local and global gradient information based segmentation proves advantageous for ELM-IPSO based tumor tissue classification.

Reduced computation time (an evaluating analysis factor) is proven, for the ELM classifier which is best acknowledged for its speed and simplicity [20]. All 10 benchmark dataset images were given to the classifier for record of corresponding computation time and mean value. For each image, the computation time including feature extraction, feature reduction, and ELM classifier is 0.2622 s, with minor variation in case 3 image. The feature extraction stage is more intensive, which a thrust area of study in future research. The total computation time for benchmark 10 dataset is about 3–5 min which is fast enough

Table 7. Summary of tumor segmentation results as compared to the SPL [31] and ITK segmentation [10].

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

when compared to existing literature studies. When compared, the automated method proposed by [31] allowed rapid identification of brain and tumor tissue with an accuracy and reproducibility in 5–10 mins. Parameter optimization with IPSO, however clearly indicates in more CPU time, than traditional ELM. Despite, the time, network accurately classifies the training data which fits the data to find good solution. The segmentation results is depicted in Table 7, where the ELM-IPSO outperforms the results of [31] and ITK segmentation [10] on the benchmark dataset. The case 5 is highly contrasted, where the results are likely to be fair as compared to benchmark dataset.

Random selection of input parameters in the design of extreme learning machine results to ill-condition problem, reducing the generalization performance, which emerges a challenging task in classifier design. Optimization of user-defined parameters substantiates this problem. The testing and the training efficiency of conventional ELM for various random initial parameters are calculated with incremental learning on number of hidden neurons. Figure 3(a) illustrates the increase in training efficiency with increase in hidden neurons. The efficiency is reached maximum with 30 hidden neurons. Similarly testing efficiency reaches a maximum when the hidden nodes are between 25 to 30 as in Figure 3(b). The performance of the ELM network with 27 hidden neurons, increases considerably with respect to initial parameters and training efficiency also reaches maximum at some random runs during this interval. Hence a need for improved ELM is advantageous. Neural networks tend to have better generalization performance with the weights of smaller norm [3]. Here an improved PSO [14], optimizes input weights and hidden biases according to root mean squared error on validation set and norm of the output weights. Introduction of constriction co-efficient ensures convergence similar to inertia parameter.

The ELM-IPSO approach were carried out by using the training and test data set with 5 hidden nodes. The number of hidden neurons, is less than the number of features. In conventional ELM [21], claimed, N number of hidden nodes are sufficient to learn N samples with negligibly small error. With a improved IPSO approach, the ELM handles, less No. of hidden neurons than the number of samples. Results suggest that Extreme learning machine with IPSO improves its classification accuracy to roughly 4% of 98.2%. The input data is normalized between values 0 and 1. The unipolar, sigmoidal activation function $f(x) = 1/(1 + e^{-x})$, with a uniform distribution range of $[-1, 1]$, produced better results. The Mean Square Error (MSE) is 0.991. The mean and standard deviation of training efficiency is 97.42% and 5.92. Similarly mean and standard deviation of testing efficiency

is 96.85 and 5.45. The norm (185.34) and condition ($4.665e+5$) output values of ELM-IPSO has a good variance when compared to traditional ELM as in Table 6.

5. CONCLUSION

In this study, an improved classifier with ELM-IPSO for brain tumor tissue characterization was explored. The classifier obtained 98.25% accuracy on SPL Harvard benchmark dataset, for both contrast and non-contrast images. The significance of feature sub-selection was revealed. Nevertheless, forfeit of this stage, leading to huge dimensionality in feature space and ill-conditioning of classifier performance. Exploration of brain mapping techniques will help in neuro-oncology management, featuring multi faceted purview. Interactive delineation of the preferred segmentation by expertise often suffers from intra-expert and inter-expert variability, though an acceptable gold standard approach. Automation of a model for computing an estimate of the “ground truth” segmentation from a group of expert segmentations, and a simultaneous measure of the quality of each expert is required to readily assess the automated image classification and segmentation algorithm performance.

The brain and tumor tissue identification provides a better perceptive of the spatial relationship; thereby lend assistance to the adage of pre-operative treatment planning. It potentially augments to the ability to diagnose nociception (body’s response to and perception of pain) which instigates with tissue injury. It also reflects the likelihood that a given voxel (spatial location) is healthy tissue, tumor, edema, neoplastic infiltration or a combination thereof. However, more real time extensive training studies are required to further substantiate these effects to further validate the performance of this computer analysis methodology. The thrust of studies towards stereotaxic space’ of MRI images can be extended in future, for multi-modality tissue characterization and protocol streamlining. Further analysis on magnetic source imaging (MSI) with magneto encephalography (MEG), with brain mappings will correlate well with intra-operative cortical simulation. The analysis is to further be extended on T2, Flair and PD images to construct a fully automated MRI analyzer.

The proposed GA-ELM-IPSO shows superiority to other literature studies artificial neural nets. The integrative mechanism of selecting the input weights and bias, along with consideration of RMSE on validation set with norm of the output weights, bring forth better generalization performance. The significant finding of this research employing volumetric texture analysis, Genetic algorithm,

ELM and improved PSO, substantiate the correlation with evaluation of MR image measures and both anatomical and histo-pathological parameters of research.

ACKNOWLEDGMENT

This work was supported by the Council of Scientific and Industrial Research (CSIR), India. The authors would like to thank Drs. W. Simon, K. Michael, N. Arya, K. Ron, B. Peter and J. Ferenc for sharing the tumor database. The authors thank PSG IMSR & Hospitals, for providing image data.

REFERENCES

1. Materka, A. and M. Strzelecki, "Texture analysis methods — A review," Technical University of Lodz, Institute of Electronics, COST B11 Report, Brussels, 1998.
2. Byun, H. and S. Lee, "Applications of support vector machines for pattern recognition: A survey," *Proc. Int. Work. Pattern Recognition with Support Vector Machines*, 213–236, Canada, 2003.
3. Barlett, P. L., "The sample complexity of pattern classification with neural networks: The size of the weights is more important than the size of the network," *IEEE Trans. Inform. Theory*, Vol. 44, 525–536, 1998.
4. Chu, A., 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, 415–420, 1990.
5. Clerc, M. and J. Kennedy, "The particle swarm — Explosion stability, and convergence in a multidimensional complex space," *IEEE Trans. on Evolutionary Computation*, Vol. 6., No. 1, 58–73, 2002.
6. Cocosco, C. A., A. P. Zijdenbos, and A. C. Evans, "A fully automatic and robust brain MRI tissue classification method," *Medical Image Analysis*, Vol. 7, No. 4, 513–527, 2003.
7. Castellano, G., L. Bonilha, L. M. Li, and F. Cendes, "Texture analysis of medical images," *Clinical Radiology*, Vol. 59, No. 12, 1061–1069, 2004.
8. Cobzas, D., N. Birkbeck, M. Schmidt, and M. Jagersan, "3D variational brain tumor segmentation using a high dimensional feature set," *Proc. Int. Conf. Computer Vision*, 1–8, 2007.

9. Deepa, S. N. and D. B. Aruna, "Second order sequential minimal optimization for brain tumour classification," *European Journal of Scientific Research*, Vol. 64, No. 3, 377–386, 2011.
10. Padfield, D. and J. Ross, "Validation tools for image segmentation," *Proc. of SPIE, Medical Imaging*, 72594W, 2009.
11. Dalai, N., B. Triggs, I. Rhone-Alps, and F. Montbonnot, "Histograms of oriented gradients for human detection," *Proc. Conference on Computer Vision and Pattern Recognition*, 886–893, 2005.
12. Xu, D.-H., A. S. Kurani, J. D. Furst, and D. S. Raicu, "Run length encoding for volumetric texture," *Proc. of Visualization, Imaging, and Image Processing*, 2004.
13. Zacharaki, E. I., S. Wang, S. Chawla, D. S. Yoo, R. Wolf, E. R. Melhem, and C. Davatzikos, "Classification of brain tumor type and grade using MRI texture and shape in a machine learning scheme," *Magn. Reson. Med.*, Vol. 62, No. 6, 1609–1618, 2009.
14. Han, F., H.-F. Yao, and Q.-H. Ling, "An improved extreme learning machine based on particle swarm optimization," *Proc. of International Conference on Intelligent Computing*, 699–704, 2012.
15. Good, C. D., R. I. Scahill, and N. T. Fox, "Automatic differentiation of anatomical patterns in the human brain: Validation with studies of degenerative dementias," *Neuroimage*, Vol. 17, No. 1, 29–46, 2002.
16. Galloway, M. M., "Texture analysis using grey level run lengths," *Comp. Graph. and Image Proc.*, Vol. 4, 172–179, 1975.
17. Georgiadis, P., D. Cavouras, I. Kalatzis, D. Glotsos, K. Sifaki, M. Malamas, G. Nikiforidis, and E. Solomou, "Computer aided discrimination between primary and secondary brain tumors on MRI: From 2D to 3D texture analysis," *E-Journal of Science & Technology (E-JST)*, 9–18, 2008.
18. Hu, X., K. K. Tan, and D. N. Levin, "Three-dimensional magnetic resonance images of the brain: Application to neurosurgical planning," *J. Neurosurg.*, Vol. 72, 433–440, 1990.
19. Huang, G. B., Q. Y. Zhu, and C. K. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, Vol. 70, No. 1, 489–501, 2006.
20. Huang, G. B., Q. Y. Zhu, and C. K. Siew, "Extreme learning machine: A new learning scheme of feed forward neural networks," *Proc. Int. Conf. Neural Networks*, 985–990, Budapest, Hungary, 2004.

21. Huang, G. B., "Learning capability and storage capacity of two-hidden-layer feed forward networks," *IEEE Transactions on Neural Networks*, Vol. 14, No. 2, 274–281, 2004.
22. Huang, G. B. and H. Babri "General approximation theorem on feed forward networks," *Proc. International Conference on Information, Communications and Signal Processing*, 698–702, Singapore, 1997.
23. Haralick, R. M., K. Shanmugam, and I. Dinstein, "Textural features for image classification," *IEEE Trans. Syst. Man Cybern.*, Vol. 3, 610–621, 1973.
24. Guyon, I. and A. Elisseeff, "An introduction to variable and feature selection," *Journal of Machine Learning Research*, Vol. 3, 1157–1182, 2003.
25. Zhang, J., C. Yu, G. Jiang, W. Liu, and L. Tong, "3D texture analysis on MRI images of Alzheimer's disease," *Brain Imaging and Behavior*, Vol. 6, 61–69, 2012.
26. Kurani, A. S., D. H. Xu, J. D. Furst, and D. S. Raicu, "Co-occurrence matrices for volumetric data," *Proc. 7th IASTED International Conf. on Computer Graphics and Imaging*, 2004.
27. Khalid, K. M. and I. Nystrom, "A modified particle swarm optimization applied in image registration," *Proc. Int. Conf. Pattern Recognition*, 2302–2305, 2010.
28. Krohling, R. A. and E. Mendel, "Bare bones particle swarm optimization with Gaussian or Cauchy jumps," *Proc. of Congress on Evolutionary Computation*, 3285–3291, 2009.
29. Leemput, K. V., F. Maes, D. Vandermeulen, and P. Suetens, "Automated model-based tissue classification of MR images of the brain," *IEEE Trans. Med. Imag.*, Vol. 18, No. 10, 897–908, 1999.
30. Lowe, D. G., "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, Vol. 60, 91–110, 2004.
31. Kaus, M. R., S. K. Warfield, A. Nabavi, M. Peter, A. Ferenc, and R. Kikinis, "Automated segmentation of MR images of brain tumors," *Radiology*, Vol. 218, 586–591, 2001.
32. Ojala, T., M. Pietikinen, and D. Harwood, "A comparative study of texture measures with classification based on featured distributions," *Pattern Recognition*, Vol. 9, No. 1, 51–59, 1996
33. Poli, R., J. Kennedy and T. Blackwell, "Particle swarm optimization: An overview," *Swarm Intelligence*, Vol. 1, 33–57, 2007.
34. Randen, T. and J. H. Husoy, "Filtering for texture classification:

- A comparative study,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 21, No. 4, 291–310, 1999.
35. Suresh, S., S. Saraswathi, and N. Sundararajan, “Performance enhancement of extreme learning machine for multi-category sparse cancer classification problems,” *Engineering Applications of Artificial Intelligence*, Vol. 23, 1149–1157, 2010.
 36. Saraswathi, S., S. Suresh, N. Sundararajan, Z. Michael, and N. Marit, “Performance enhancement of extreme learning machine for multi-category sparse cancer classification problems,” *ACM Transactions on Computational Biology and Bioinformatics*, Vol. 8, No. 2, 452–463, 2011.
 37. Suresh, S., V. Babu, and H. J. Kim, “No-reference image quality assessment using modified extreme learning machine classifier,” *Applied Soft Computing*, Vol. 9, 541–552, 2009.
 38. Siedlecki, W. and J. Sklanky “A note on genetic algorithms for large-scale feature selection,” *Pattern Recognition Letters*, Vol. 10, No. 5, 335–347, 1989.
 39. Xu, Y. and Y. Shu, “Evolutionary extreme learning machine — Based on particle swarm optimization,” *Proc. Inter. Symp. Neural Networks*, Vol. 3791, 644–652, 2006.
 40. Zhu, Q. Y., A. K. Qin, P. N. Suganthan, and G. B. Huang, “Evolutionary extreme learning machine,” *Pattern Recognition*, Vol. 38, No. 10, 1759–1763, 2005.
 41. Zhang, Y. and L. Wu, “An MR brain images classifier via principal component analysis and kernel support vector machine,” *Progress In Electromagnetics Research*, Vol. 130, 369–388, 2012.