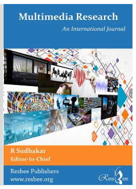
Vol.5 No.4 Oct.2024

Multimedia Research

Received 3 June, Revised 17 September, Accepted 12 October



Brain Tumor classification using the Probabilistic-Deformable Fuzzy system and RideNN

G. Srishailam
Dept of CSE
Maturi Venkata Subba Rao Engineering College, Hyderabad, Telangana, India.

Abstract: In the medical field, brain tumor classification is a very significant treatment process. The manual segmentation and classification of brain tumors is a tedious and time-consuming process. Hence, automatic classification and segmentation methods are very important. This paper presents brain tumor classification using a Probabilistic-deformable fuzzy system (PDFS) with RideNN. The overall procedure of the proposed approach involves four steps, such as pre-processing, segmentation, feature extraction, and classification. Initially, the input image is subjected to pre-processing. Then, the brain tumor segmentation is done using a new model called the Probabilistic-Deformable Fuzzy system. It is developed by combining a deformable model and a fuzzy system. Then the result is combined using probability-based fusion. After the segmentation, feature extraction is carried out by extracting texture-based features such as Local Gradient Pattern (LGP) and tetrolet transform. Finally, the classification is performed based on the features extracted using RideNN which employs Rider optimization Algorithm (ROA) for training the Neural network (NN). The performance of the brain tumor classification is based on PDFS-RideNN and evaluated based on sensitivity, specificity, and accuracy. The proposed PDFS-RideNN method achieves a maximal sensitivity of 93.18%, a maximal specificity of 98.67%, and a maximal accuracy of 98.25%.

Keywords: Brain tumor classification, Magnetic Resonance Imaging, Probabilistic-deformable fuzzy system, RideNN, Local Gradient Pattern

1. Introduction

In the past two years, Magnetic Resonance Imaging plays a very important role in brain anomaly detection by determining the location and size of the affected tissues [22]. MRI is a type of medical image processing. It is mainly utilized for visualization of the inner composition of the human body by the radiologist. It provides details regarding human soft tissue anatomy [5]. MRI is a component in diagnosis and treatment planning, which significantly increased the knowledge of diseased and normal anatomy for medical research [23]. Because of the high spatial resolution and high contrast of soft tissues, MRI does not produce any harmful radiation. MRI is efficient when compared with all other techniques in the application of brain tumor identification and detection [6] [24] [15]. Images captured using MRI are utilized to investigate and analyze the behavior of the brain. MRI of the brain is utilized for monitoring the response of tumors for the treatment process [21]. It is the most advisable one since it does not use ionizing radiation and it also provides greater contrast between various soft tissues of the human body [3].

An unusual growth of tissue in the brain is known as a brain tumor [5]. Brain tumors are abnormal growths in the brain that can be either non-cancerous (benign) or cancerous (malignant). The effects of benign and malignant brain tumors are the same and can cause issues based on the type of tumor and the location of the brain. In the United States, more than 200,000 people are diagnosed with metastatic and primary brain tumors [1] [8]. A brain tumor is one of the most serious and life-threatening tumors, created by either uncontrolled or abnormal cell division within the brain. Tumors may affect the healthy cells indirectly or directly, causing swelling in the brain and increasing the pressure within the cell. The tumor is categorized based on the position of its origin and its malignancy [3] [2]. Tumors are divided into benign and malignant tumors [4]. Acute Lymphoblastic Leukemia (ALL) is the fifth leading cause of cancer death in female adults aged between 20-39, and the second leading cause of cancer death in male adults aged between 20 and 29. Primary brain tumors can metastasize but did not spread outside the central spinal canal or the intracranial cavity [8].

The most common type of brain tumor is the Metastatic brain tumor which spread the cancer tissue from other parts of the body to the brain. They occur in 10-15% of people. In manual diagnosis, the

https://doi.org/10.46253/j.mr.v5i4.a1

amount of MRI images is to be analyzed to make the readings based on visual interpretation intensive, expensive, and inaccurate. Also, manual diagnosis is time-consuming [10] and is based on the subjective decisions of the radiologists which is hard to quantify, which may lead to misclassification [11] [9]. Classification is the technique to classify objects into corresponding classes. For classification, various features of the image are extracted. These features are utilized for categorizing the brain MRI image as abnormal and normal [7]. Several classification techniques like Fuzzy C-Means (FCM), Artificial Neural Network (ANN), Decision Tree, K-Nearest Neighbour (KNN), Support Vector Machine (SVM), and Bayesian classification are available for brain tumor classification [12][13].

This paper presents brain tumor classification using MRI brain images. At first, the MRI image is given for pre-processing. After pre-processing, the brain tumor segmentation is done based on the proposed PDFS for effective segmentation. The proposed PDFS involves the generation of segments, probability estimation, and the convolution of input images using a filter. Here, the pre-processed image is fed to the segmentation. Then, the probability is determined using active contour and sparse FCM. After that, the convolution of the input image is performed using filters for binary mapping. At last, the total probability is calculated separately for the segmented outputs and the original image. The maximum probability is selected for obtaining the final segmentation output. After segmentation, feature extraction is carried out by extracting the features. At last, the classification is performed based on RideNN for training the Deep CNN for the brain tumor classification.

The contributions of this research work toward brain tumor classification are enlisted below:

- Development of PDFS for brain tumor segmentation, which is designed by combining a deformable model and fuzzy system.
- The RideNN is employed for classification that uses the Rider Optimization algorithm to train the NN classifier.

The rest of the paper is organized as follows: Section 1 depicts a brief introduction to the paper. Section 2 explains the literature review. Section 3 describes the proposed Probabilistic-Deformable Fuzzy system with RideNN developed for brain tumor classification. Section 4 explains the results and discussions obtained using the proposed method and Section 5 provides the conclusion.

2. Literature Survey

This section deals with the existing techniques of brain tumor classification as explained as follows. V. Anithaand S. Murugavalli [15] introduced a two-tier classification system for the classification of brain tumors. Initially, the input image was given to pre-processing for removing the noise. Then, the brain tumor segmentation was performed based on adaptive pillar K-means clustering. After the segmentation, the feature extraction was carried out using discrete wavelet transform (DWT). At last, the classification was done based on the classifier to improve the system's performance. The method failed to consider the classification of images with different pathological conditions, types, and disease statuses. Zaka Ur Rehman *et al.* [16] developed an approach for brain tissue classification for abnormal and normal tissues from Fluid Attenuated Inversion Recovery modality of Magnetic Resonance (MR) images. The developed regional classification approach was utilized for segmenting and detecting tumors. The real-time implementation of the classification pipeline for deployment in clinical setups and its applicability to other MRI modalities was not considered. Mohammadreza Soltaninejad*et al.* [17] developed a 3D super voxel-based learning approach from MRI brain images for tumor segmentation. Gabor texton-based features along with the statistical feature were computed for every super voxel. These features were subjected to Random Forests (RF) classifier for classifying each super voxel into tumor edema, healthy brain tissue, and tumor core. The method failed to focus on DTI modalities of tumor tissue subtypes, such as necrosis and enhancing tumors. Adriano Pinto *et al.* [18] developed a hierarchical brain tumor segmentation pipeline automatically based on 20 extremely randomized trees with context and appearance-based features. These features were calculated over non-linear transformations from MRI images. The method contained many pure nodes that may incur over-fitting. Mohammadreza Soltaninejad*et al.* [19] developed an approach for the segmentation and detection of abnormal tissue. At first, the input image was fed to the pre-processing to get the exact region. After pre-processing, the FLAIR image was split into irregular patches with the same size and intensity values. Then, for each superpixel patch, a variety of features, like texton, statistical, and shape features were computed. After that, an Extremely Randomized Trees (ERT) classifier was utilized for classifying each superpixel into healthy brain tissue or tumor. Chao Ma *et al.* [20] developed an approach that integrated an active contour model and random forests for gliomas segmentation from MRI images. Feature representations learning strategy was employed in both contextual and local information for tissue segmentation. Labeled training data were not considered for better system performance. A. Shenbagarajan*et al.* [21]developed an approach for tumor diagnosis from MRI images. The overall procedure of the developed approach involves four steps, such as pre-processing, segmentation, feature extraction, and classification. Initially, the input image was subjected to pre-processing for removing the noise in the image. Then, brain tumor segmentation was performed using an active contour model for segmentation. After the segmentation, feature extraction was done by extracting texture features using Local Directional Pattern (LDP). At last, the classification was carried out using Artificial Neural Network (ANN)-based Levenberg-Marquardt (LM) algorithm for classifying the MRI images as numerous or normal. The neural network was not considered for medical classification.D. Jude Hemanth, J. Anitha [28] employed modified Genetic algorithm (GA) based approaches for classification. Three various GA approaches were introduced for feature selection. Here, Back Propagation Neural Network was utilized as a classifier for all the GA-based methods. The method failed to consider other optimization algorithms to improve the system's performance.

2.1. Challenges

The challenges faced by the existing techniques are listed as follows:

- Automated segmentation and detection of brain tumors are very challenging due to their high variation in shape, appearance, and size (e.g. image uniformity and texture) [25].
- Brain tumor segmentation from MRI images is of great importance for growth rate prediction, improved diagnosis, and treatment planning. However, automating this process is a very challenging task due to the imaging conditions, the presence of severe partial volume effect, and considerable variability in tumor structures, especially for gliomas [20].
- Brain tumor classification [14] plays a very important role in the medical field. The images obtained from various modalities, like MR and CT, should be verified by the physician for further treatment, but the manual classification of the MR images is a time-consuming, and challenging task [2].
- Computer-aided diagnosis (CAD) helps to enhance diagnostic accuracy. However, it is still a challenging issue to automatically classify and segment brain tumors from MRI images due to the variety of shapes, appearance properties, and tumor types10].
- In [21], Levenberg-Marquardt (LM) algorithm is developed for brain tumor classification. Here, the classification accuracy is found better, but LM is extensively never been examined on a neural network for the use of medical condition classification.

3. Brain tumor classification using the proposed Probabilistic-Deformable Fuzzy system and RideNN

This section presents the brain tumor classification using the proposed PDFS-RideNN. Figure 1 deliberates the schematic diagram of the proposed probabilistic-deformable fuzzy system-based classification method. The overall procedure of the proposed approach involves pre-processing, segmentation, feature extraction, and classification. Initially, the multimodal MRI image is subjected to pre-processing. Then, the brain tumor segmentation is performed based on the proposed probabilistic-deformable fuzzy system, which is developed by integrating a deformable model and fuzzy system, and the obtained result is combined using probability-based fusion. After the segmentation, feature extraction is carried out by extracting texture-based features, like Local Gradient Pattern (LGP) and tetrolet transform, and the statistical features, such as mean, area, kurtosis, area, and variance. Finally, the classification is performed based on the extracted features using RideNN [27] which employs Rider optimization Algorithm (ROA) for training the Neural network (NN).

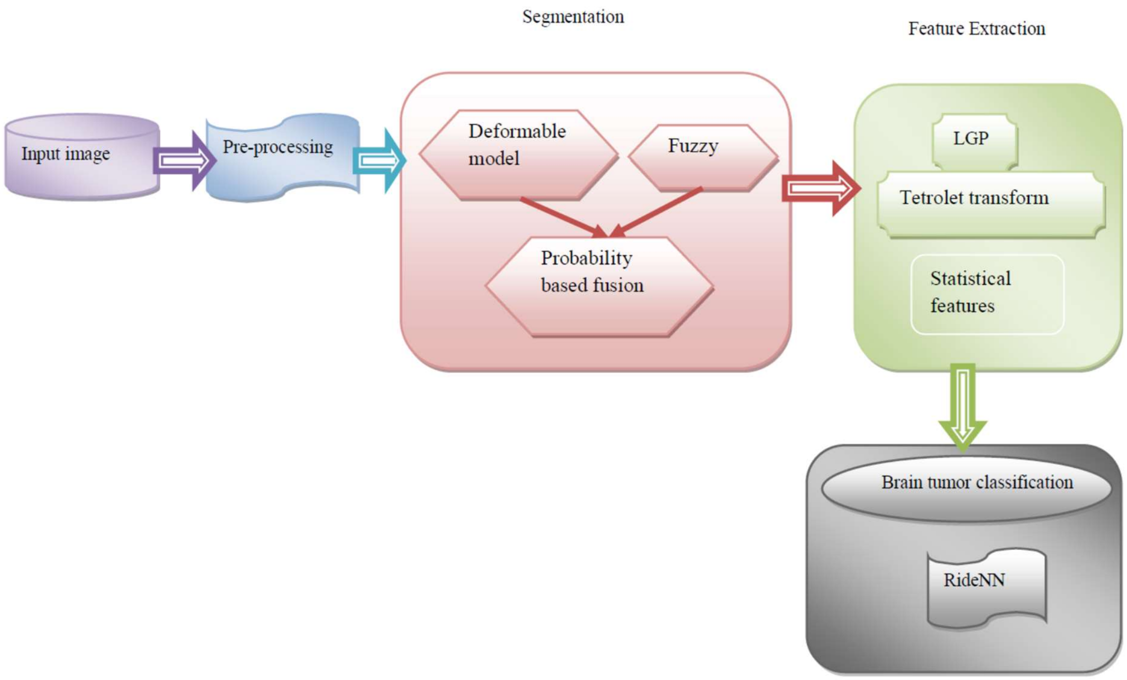


Figure 1 Schematic diagram of the Brain tumor classification method using the proposedPDFS and RideNN

3.1 Pre-processing

Let us assume the input image $I(x,y)$, which is given to the pre-processing phase to improve the contrast of the image. The first step involved in brain tumor classification is pre-processing. Here, the individual slices are pre-processed from the input images. The pre-processed image is denoted as, $I^{*}(x,y)$
.

3.2 Segmentation using the proposed probabilistic-deformable fuzzy system

After pre-processing, the pre-processed image $I^{*}(x,y)$ is subjected to the proposed probabilistic-deformable fuzzy system for segmentation, which is designed by determining the optimal segments using the outputs from the active contour, sparse FCM, and the input image. The proposed probabilistic-deformable fuzzy system includes the generation of segments, probability estimation, and the convolution of input images using a filter. Initially, the pre-processed image is given to the active contour and sparse FCM for segmentation. After that, the probability is determined for the segments obtained using the active contour and sparse FCM. As the third step, the convolution of an input image is performed using a filter by considering the neighborhood of the center pixel, resulting in the binary map. At last, the total probability with '1's and '0's is calculated separately for the segmentation outputs and the original image, and the maximum probability is selected for obtaining the final segmentation output. The steps involved in the proposed algorithm are illustrated below:

3.2.1Generation of segments using Active contour: Once the pre-processing is done, the pre-processed image is given to the active contour. Active contour [20]is utilized for separating the curvature or boundary for the target object regions for segmentation, obtaining deformable structures or models with constraints and forces in an image for segmentation, defining smooth shapes in the image, and forming closed contour for the region. Figure 2 depicts the output of active contour.

1	1	0
1	**0**	**0**
1	**1**	**1**

Figure 2. Active contour output

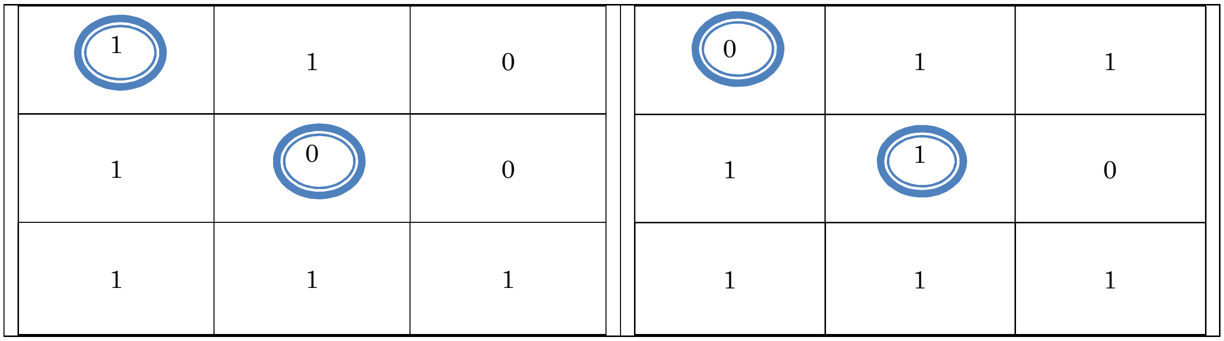
From the above figure, it is shown that the brain regions are marked as"1", whereas the unfilled region is marked as "0". The output of the segmented image using active contour is denoted as, $S_1^*(x,y)$.

3.2.2 Generation of segments using Sparse FCM: Sparse FCM model imposes the sparse regularization to assign exact zero weights to the noisy features for clustering high-dimensional data, which maximizes the weight between-cluster sum of squares (BCSS) with l-norm regularization. Figure 3 depicts the output of the sparse FCM. The output of the sparse FCM image is denoted as $S_2^*(x,y)$.

0	1	1
1	**1**	**0**
1	**1**	**1**

Figure 3. Sparse FCM output

3.2.3 Probability estimation using active contour and the sparse FCM: Once the output is obtained from both the active contour and sparse FCM, the matching is performed between the two pixels. If the variation exists between the corresponding pixels of two segments, then the probability is found out for ones and zeroes. In Figure 4, the blue circle indicates the variation between the two images. Figure 4 shows the computation of probability.



1	1	0
1	0	0
1	1	1

0	1	1
1	1	0
1	1	1

Figure 4. Probability computation

The probability of ones and zeroes from both the output is computed using the below formula,

$$\rho_R^{one} = \frac{\text{Number of pixels with value one}}{\text{Total number of pixels}} \tag{1}$$

$$\rho_R^{zero} = \frac{\text{Number of pixels with value zero}}{\text{Total number of pixels}} \tag{2}$$

where, $R \in \{1,2\}$ represents the active contour output, and the sparse FCM output. For example, the probability of ones in active contour output shown in figure (2) is computed using equation (2) as,

$$\rho_R^{one} = \frac{6}{9}$$

Similarly, the probability of zeroes in active contour output is computed using equation (2),

$$\rho_R^{zeroes} = \frac{3}{9}$$

From the same procedure, the probability of ones and zero for sparse FCM is calculated.

3.2.4 Convolution of the input image using a filter

After the computation of probability, convolution is applied to the input image using the filter. For convolution, assume the filtered image, and then the input image is convoluted with the filter. Here, in the input image, the center pixel has to be replaced by considering the neighborhood of the center pixel. Figure 5 depicts the convoluted image.

Input image

5	5	10
11	**13**	7
4	12	13

$$\otimes$$

Filter image

1	0	1
1	0	0
1	0	1

Figure 5. Convoluted images

From the figure, the center pixel (blue square) has to be replaced. The resultant output obtained after convolution is expressed as,

$$S^b = I_l^* \otimes F \tag{3}$$

$$S^b = 5\times1+15\times0+10\times1+11\times1+13\times0+7\times0+4\times1+12\times0+3\times1 = \frac{33}{9}$$

$$S^b = 3.667$$

where, S^b is the resultant image. Figure 6 depicts the replacement of the center pixel using input and filter images.

5	15	10
11	3.667	7
4	12	3

Figure 6. Replacement of center pixel

Once the center pixel is replaced, the binary map is generated. To generate the binary map, the below conditions are used.

$$S_3(x,y) = \begin{cases} 1 & ;S(x,y) > N(x,y) \\ 0 & :\text{otherwise} \end{cases} \tag{4}$$

where, $S_3(x,y)$ is the binary map, $S(x,y)$ signifies the center pixel of the input image, and the neighboring pixel is denoted as, $N(x,y)$. After these conditions, the neighboring pixel is marked as "0", as shown in Figure 6.

1	1	1

1	0	1
1	1	0

Figure 7. Conversion of neighboring pixel

After the above procedure, the total probability is found by summing the three images, like active contour, sparse FCM, and the original image using ones and zeroes. The total probability of ones in three outputs is expressed as,

$$\rho^{\text{one}} = \sum_{\text{R}=1}^{3} \rho_{\text{R}}^{\text{one}} \tag{5}$$

$$\rho^{\text{one}} = \rho_1^{\text{one}} + \rho_2^{\text{one}} + \rho_3^{\text{one}} \tag{6}$$

$$\rho^{\text{one}} = \frac{6}{9} + \frac{7}{9} + \frac{7}{9} = \frac{20}{9} = 2.22 \tag{7}$$

where, ρ^{one} denotes the probability output, ρ_1^{one} refer to the active contour image with value one, and the sparse FCM image with value one is represented as ρ_2^{one}, and ρ_3^{one} signifies the original image with value one.

The total probability of zeroes in three outputs is expressed as,

$$\rho^{\text{zero}} = \sum_{\text{R}=1}^{3} \rho_{\text{R}}^{\text{zero}} \tag{8}$$

$$\rho^{\text{zero}} = \rho_1^{\text{zero}} + \rho_2^{\text{zero}} + \rho_3^{\text{zero}} \tag{9}$$

$$\rho^{\text{zero}} = \frac{3}{9} + \frac{2}{9} + \frac{2}{9} = \frac{7}{9} = 0.778 \tag{10}$$

From the above two outputs, the maximum probability value is chosen as the output. As the probability of one is high, the corresponding pixel is marked as “1”. The final segmented output is expressed as,

$$\text{s} = \{\text{s}_1, \text{s}_2, ..., \text{s}_j, ..., \text{s}_m\} \tag{11}$$

where, s is the final segmented image.

3.3 Extraction of features

After the segmentation of the tumor image, the feature extraction is carried out by extracting the texture-based features, such as LGP [31] and Tetrolet transform [32] and the statistical features, like mean, variance, kurtosis, area, and skewness.

3.3.1Tetrolet Transform: The tetrolet descriptor is an adaptive Haar wavelet transform, to support tetrominoes, which is formed by joining four squares of similar size. The steps involved in the tetrolet transform are explained below,

a) Formation of image blocks: The segmented input image is partitioned as blocks with the size 4×4.

b) Represent the image blocks as the sparse tetrolet: Each image block is subjected to the sparsest tetrolet representation and for each block 117 tetromino coverings are admitted and each is subjected to the Haar wavelet transform to generate 12 coefficients along with the four low pass coefficients. The sparse image is obtained by performing the tetrolet decomposition for each block.

c) Representation of the Low pass and high pass coefficients: The steps involved in the Tetrolet decomposition algorithm persist with the matrix arrangement using the reshape function.

d) Tetrolet Coefficients: After representing the sparse matrix for the individual blocks, the high pass and the low pass matrices are kept secure for future use.

e) Termination: The procedure is repetitive for the low-pass image and the result obtained is the binary image. The output generated from the Tetrolet transform is represented as T.

3.3.2LGP: LGP [17] is a human and face representation method to generate constant patterns, which is irrespective of the local intensity variations with the edges. The operator in LGP uses the gradient values of eight neighbor pixels, which are determined as the intensity absolute value. The average value of the gradient among the eight neighboring pixels is considered the threshold value. If the gradient value of the neighboring pixel is higher than the threshold, then the value assigned to the pixel is '1', otherwise the value is '0'. Thus, $f_l(x_l, y_l)$ is represented as,

$$f_l(x_l, y_l) = \sum_{l=0}^{u-1} s(v_l - \bar{v})2^l \tag{12}$$

where, $s(r) = \begin{cases} 0, & \text{if } r < 0, \\ 1, & \text{otherwise.} \end{cases}$, v_l denotes the neighboring intensity, and $\bar{v}$ refer to the center pixel intensity. The dimension of $f_l(x_l, y_l)$ is $[1 \times 32.]$.

3.3.3Statistical features

a) Kurtosis and skewness: Kurtosis measures the respective sharpness of the curve, in the frequency distribution, and the skewness indicates the lack of symmetry.

b) Mean: Mean is calculated in the discrete segments by averaging the pixels that are expressed as,

$$\mu = \frac{1}{|p(k_i)|} \times \sum_{i=1}^{|p(k_i)|} p(k_i) \tag{13}$$

where, i signifies the segments, such as a core tumor, edema tumor, and the raw input image, and $|p(k_i)|$ represents the total number of pixels in the segmented image.

c) Variance: Variance is estimated based on the mean, as expressed by,

$$\sigma = \frac{\sum_{i=1}^{|p(k_i)|} |k_i - \mu|}{p(k_i)} \tag{14}$$

where the term μ denotes the average value.

f) Area: This feature denotes the boundary enclosed by the tumor and is denoted as, π. Thus, the dimension of the feature vector is expressed as, $[1 \times 24]$.

$$M = \{\kappa, \delta, \mu, \pi, \ell, \varepsilon, \sigma, \hbar\} \tag{15}$$

where κ denotes the kurtosis, μ denotes the mean. The skewness and the information gain are represented as, δ and ℓ. ε refer to the energy, $\hbar$ denotes the hole entropy, and π denotes the area.

3.4 Brain tumor classification using the proposed RideNN

This section elaborates the Brain tumor classification using RideNN, which is designed by integrating ROA as a training algorithm in Neural Network (NN) classifier. Here, the weights are computed optimally using ROA.ROA [27] is inspired by the group of riders racing toward the target location. The algorithm needs limited time for processing and thus, the computational time is subsequently reduced. NN [29] is one of the artificial intelligence models for performing the classification using two processes, such as training and testing. In the training phase, the features are used to learn the neurons by finding the optimal weights and then, the learned weights are employed to identify the class label of the tracked objects about the direction of movement and activity of the object. The architecture of NN and the algorithmic steps of the proposed ROA are illustrated below,

a) Architecture of NN

The NN consists of an input layer, output layer, and hidden layer. Here, the feature vector M is given as the input of NN. Therefore, the normalization enhances the precision of the whole network. Hidden layers are comprised of neurons that are connected to the output layer along with the weights. The output layer is the composition of the output neurons and all these layers are interrelated to form a network. Figure 7 depicts the architecture of NN for brain tumor classification.

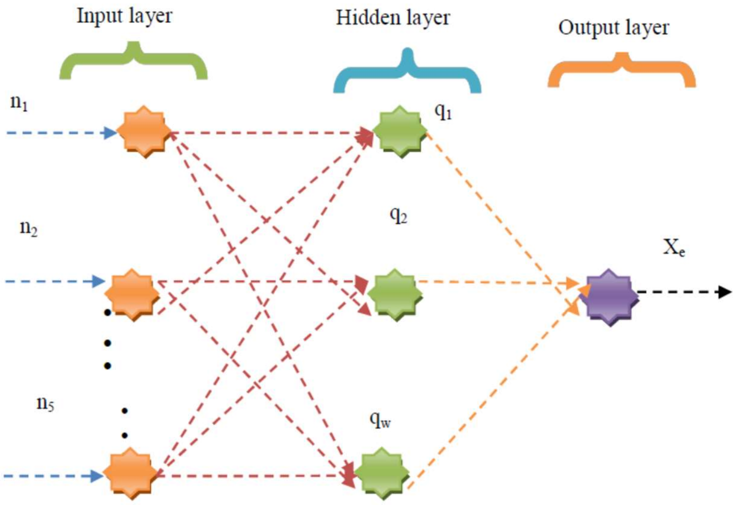


Figure 7. Architecture of NN

Assume n be the input layer in the network represented as

$$n = \{n_1, n_2, ..., n_5\} \tag{16}$$

where, n_5 is the total number of input neurons. The expression for the hidden layers in the network is given as,

$$Q = \{q_1, q_2, ..., q_t, ...p_w\} \tag{17}$$

where, w denotes the number of hidden neurons in the neural network and q_t is the r^{th} hidden neuron, which has the output computed as,

$$q_t = \frac{1}{u}\sum_{z=1}^{u} K_z\, n_z \tag{18}$$

where, K_z is the z^{th} weight between the hidden and the input neurons, and u is the total number of weights. The output of the neural network is computed as,

$$U_d = \sum_{d=1}^{w} q_d \times n_d \tag{19}$$

Rewriting the output layer, the solution becomes $Y_j = R(n_d, n)$, which means the output layer is the function of the input layer and the weights. The representation of the weights in the neural network is expressed as,

$$W = \{\omega_1, \omega_2, ..., \omega_c\} \tag{20}$$

where the number of weights in the NN is denoted as c.

b) Training of NN usingRider Optimization Algorithm

ROA is based on the group of riders, targeting the destination. Unlike any other nature-inspired and artificial computing, ROA is fully based on the new computing approach termed fictional computing, which is duly based on imaginary thoughts and ideas for deciding the winner based on the success rate of the riders. The key terms of the ROA algorithm are defined below,

a) Bypass rider: Bypass rider, reaches the target by bypassing the leading path. This signifies that the bypass rider failed to follow the leading rider, who is in the leading position.
b) Follower: The follower depends on or follows the leading rider in most of the axis.
c) Overtaker: This rider follows his position for reaching the target based on the nearby location of the leading rider.
d) Attacker: The attacker is the aggressive player, who takes the position of the rider to reach the target point with maximum speed. In addition to these four definitions, there includes a term, called the winner, who is the final winner of the race. The steps of the ROA are given below;
Initialization: The riders and their positions are initialized as given,

$$Y_j = \{Y_j(g,h)\};\ (1 \le g \le M);\ (1 \le h \le N) \tag{21}$$

where, M and N refer to the total riders and coordinates. $\mathrm{Y_j(g,h)}$ signifies the location of $\mathrm{g^{th}}$ rider at the time instant j. There are a total of B, F, G, and A number of riders, such as bypass, followers, overtakers, and attackers that are expressed as,

$$\mathrm{M = B + F + G + A} \tag{22}$$

The relation between the above-said parameters is given as,

$$\mathrm{B = F = G = A = \frac{M}{4}} \tag{23}$$

The steering angle, position angle, and coordinate angle of the $\mathrm{g^{th}}$ rider's vehicle are denoted as, $\mathrm{E^{j}_{g,h}}$, ϕ_g, and δ. Let us assume that the gear, accelerator, and brake of $\mathrm{g^{th}}$ rider are denoted as, $\mathrm{q_g}$, $\mathrm{a_g}$, and $\mathrm{f_g}$, respectively. The value of $\mathrm{q_g}$ takes any value between 0 and 4, whereas the values of $\mathrm{a_g}$, and $\mathrm{f_g}$ takes between 0 and 1.

Evaluation of success rate: After that, the success rate is calculated based on the distance, which is measured between the position of the rider and the target, as given below,

$$\mathrm{e_g = \frac{1}{\left\| Y_g - C_L \right\|}} \tag{24}$$

where, $\mathrm{Y_g}$ be the position of the $\mathrm{j^{th}}$ rider, and $\mathrm{C_L}$ refer to the location of the target.

Update the position of bypass rider: The common path is bypassed by the bypass rider, not following the leader, and the position of the bypass rider is modeled as

$$\mathrm{Y^{B}_{j+1}(g,h) = \gamma\left[Y_j(\upsilon,h) * \chi(h) + Y_j(\mu,h) * (1-\chi(h))\right]} \tag{25}$$

where, γ signifies the random number ranging from 0 and 1. υ refer to the random number and it varies between 1 and m, μ signifies the random number between 1 and m. On the other hand, χ refers to the random number between 0 and 1, which is of size $(1 \times \mathrm{N})$. Therefore, the position of the bypass rider is updated to announce the winner.

Update the position of follower: The follower updates the location of the leader to reach the target, which is modeled as

$$\mathrm{Y^{F}_{j+1}(g,h) = Y^{C}(C,h) + \left[\cos\left(E^{j}_{g,h}\right) * Y^{C}(C,h) * \partial^{j}_{g}\right]} \tag{26}$$

where, h signifies the coordinate selector, the position of the leader is denoted as, $\mathrm{Y^C(C,h)}$, C denotes the leading rider, $\mathrm{E^{j}_{g,h}}$ indicates the steering angle of $\mathrm{g^{th}}$ rider in $\mathrm{h^{th}}$ coordinate, and ∂^{j}_{g} specifies the distance travelled by $\mathrm{g^{th}}$ rider. The distance is determined based on the product of the velocity and off-time.

Update the position of overtaker: The overtaker updates the position based on the direction indicator, coordinate selector and success rate. The overtaker updates the position as,

$$\mathrm{Y^{G}_{j+1}(g,h) = Y_j(g,h) + \left[D_j(g) * Y^{C}(C,h)\right]} \tag{27}$$

where, $\mathrm{Y_j(g,h)}$ refers to the position of $\mathrm{g^{th}}$ rider in $\mathrm{h^{th}}$ coordinate and $\mathrm{D_j(g)}$ refers to the direction indicator of $\mathrm{g^{th}}$ rider in time j. The value of $\mathrm{D_j(g)}$ is computed based on the success rate as,

$$\mathrm{D_j(g) = \left[\frac{2}{1 - \log\left(s_j{}^{M}(g)\right)}\right] - 1} \tag{28}$$

where, $\mathrm{s^{M}_{j}(g)}$ specifies the success rate of $\mathrm{g^{th}}$ rider at j, which is the ratio of the success rate of the rider to the maximum success rate obtained among n riders, and the value of lies between 0 and 1. The coordinate selector is computed as the difference between the positions of $\mathrm{g^{th}}$ rider and leading rider.

Update the position of attacker: The attacker attempts to reach the location of the leader, and posses an update process similar to that of the follower. The position of the attacker is given as,

$$Y_{j+1}^{A}(g,h) = Y^{C}(C,H) + \left[\cos\left(E_{g,h}^{j}\right) * Y^{C}(C,h) + \partial_{h}^{j}\right] \tag{29}$$

It is essential to update the parameter of the riders in determining the optimal solution. Thus, gear, steering angle, accelerator, ride off-time, and brake along with a new parameter activity counter are updated at the end of the iteration. The activity counter either takes the value '0' or '1', depending on the success rate.

4. Results and discussion

The results and discussion of the proposed PDFS-RideNNare demonstrated in this section with an effective comparative analysis to prove the effectiveness of the proposed method.

4.1 Experimental Setup

The work is implemented in the MATLAB tool, with the PC configuration as 4 GB RAM, Windows 10 OS, and Intel I3 processor.

4.1.1 Database description

The dataset is taken from BRATS dataset [30] taken from BRATS 2014 for brain tumor classification that has two severity levels of brain tumor images. Here, the image of every patient is collected in four modalities, T1, T1C, T2, and FLAIR. In this dataset, all the datasets are manually segmented, by one to four rates, which follow a similar annotation protocol, approved by experienced doctors.

4.1.2 Evaluation metrics

The evaluation of the proposed technique is performed using three metrics, such as accuracy, sensitivity, and specificity, and the metrics are explained as follows:

a) Sensitivity: Sensitivity measures the true positives correctly detected using the proposed method. The expression for the sensitivity is,

$$\text{Sensitivity} = \frac{p^{+}}{p^{+} + n^{-}} \tag{30}$$

b) Specificity: The rate of probability of true negatives is termed as Sensitivity and is given as,

$$\text{Specificity} = \frac{n^{+}}{n^{+} + p^{-}} \tag{31}$$

c) Accuracy: The accuracy denotes the measure of the closeness of the proposed method for brain tumor classification, and is given as,

$$\text{Accuracy} = \frac{p^{+} + n^{+}}{p^{+} + p^{-} + n^{+} + n^{-}} \tag{32}$$

where, p^{+} is the true positive, n^{+} is the true negative, p^{-} is the false positive, and n^{-} is the false negative.

4.1.3 Competing methods

The methods used for comparison include the Extremely Randomized trees [18], RMD-SMOTE [16], Deep Convolutional Neural Network (DeepCNN), FCM-RideNN, and ANN-LM, and the results of the proposed PDFS-RideNNis compared with the results of the existing methods.

4.1.4 Experimental Results

The experimental results achieved by the proposed PDFS-RideNNare given in this section. Figure 8 depicts the original slice of the four modalities related to sample image 1. Figure 8a) shows the flair modality of the original image 1, figure 8b) shows the T1 modality, figure 8c) depicts the T1C modality, and figure 8d) shows the T2 modality belonging to the original image. Figure 8 e) depicts the segmented output of the FLAIR modality of the original image 1, figure 8 f) shows the segmented image of the T1 modality, figure 8g), and figure 8h) demonstrates the segmented output of T1C modality, and T2 modality.

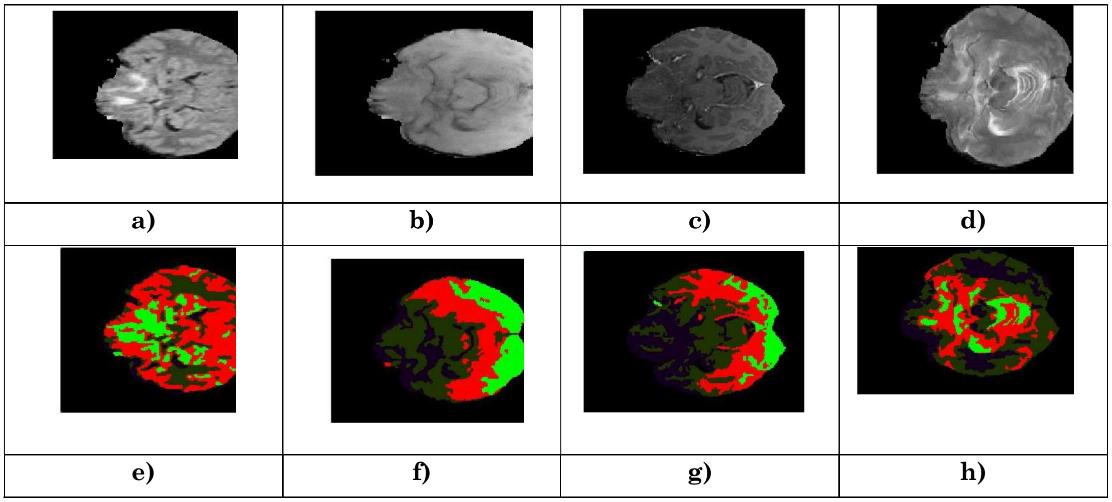


Figure 8. Original Image 1 a) Slice-80 Flair modality, b) Slice-80 T1 modality, c) Slice-80 TIC modality, d) Slice-80 T2 modality, e) segmented output of Flair modality of slice80, f) segmented output of Slice-80 T1 modality, g) segmented output of slice 80 T1C modality, h)segmented output of slice 80 T2 modality

Figure 9 depicts the original slice of the four modalities corresponding to image 2. Figure 9 a), figure 9b), figure 9c), and figure 9d), show the flair, T1, T1C, and T2 modality belonging to the original image. The segmented output from the flair, T1, T1C, and T2 modality are depicted in Figure 9e), figure 9f), figure 9g), and Figure 9h) respectively.

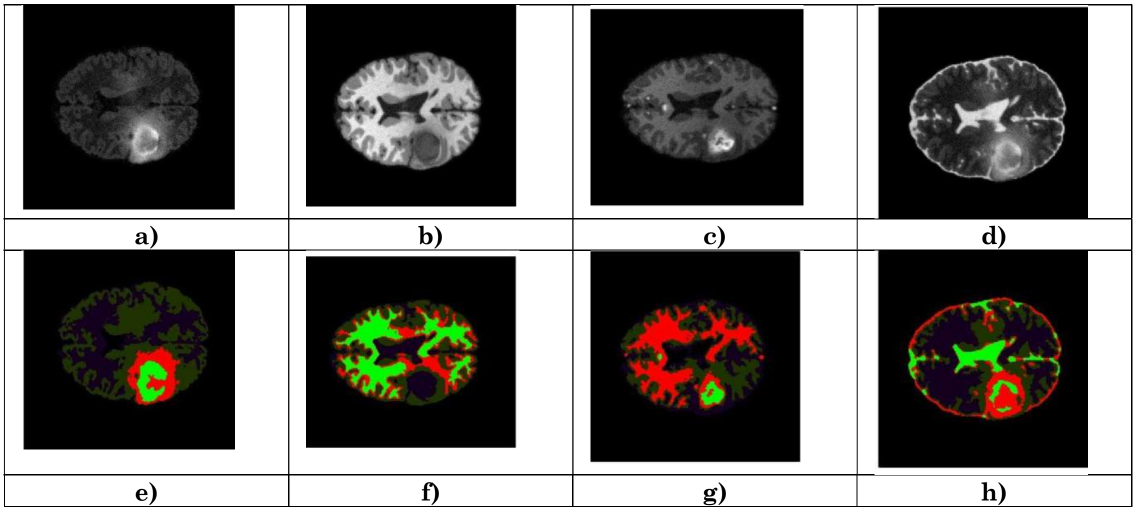


Figure 9. Original Image 1 a) Slice-90 Flair modality, b) Slice-90 T1 modality, c) Slice-90 TIC modality, d) Slice-90 T2 modality, e) segmented output of Flair modality of slice90, f) segmented output of Slice-90 T1 modality, g) segmented output of slice 90 T1C modality, h)segmented output of slice 90 T2 modality

4.1.5 Comparative analysis

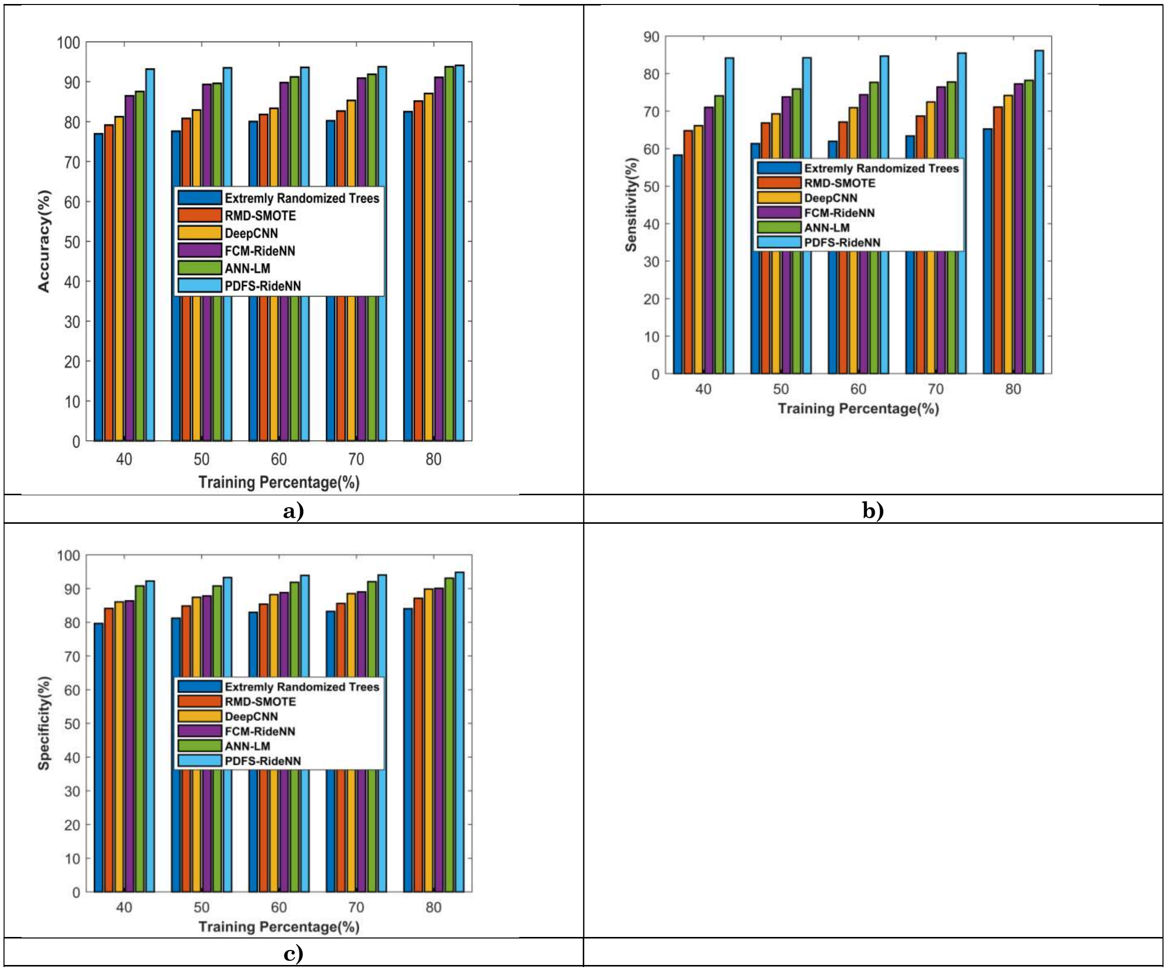
The comparative analysis of the proposed PDFS-RideNN by evaluating the performance of other comparative techniques is elaborated in this section. The comparative analysis is performed by varying the percentage of training data, and the results are evaluated based on metrics, like accuracy, specificity, and sensitivity.

4.1.5.1 Analysis using Image-1

a) Based on training percentage: The comparative analysis based on accuracy, sensitivity, and specificity using image-1 is depicted in Figure 10. Figure 10a) shows the analysis based on accuracy by varying the training percentage. For the training, data=40%, the existing techniques, like Extremely Randomized trees, RMD-SMOTE, DeepCNN, FCM-RideNN, and ANN-LM, have accuracy values of 76.9410%, 79.133%, 81.2180%, 86.4697%, and 87.527%, respectively, which is comparatively lower than the proposed method. For the same training data, the developed PDFS-RideNN has an accuracy value of 93.1773%. Similarly, when the training data percentage increases to 80%, the methods, Extremely Randomized trees, RMD-SMOTE, Deep Convolutional Neural Network, FCM-RideNN, and ANN-LM have accuracy values of 82.482%, 85.143%, 87.040%, 91.0683 %, and 93.740%, and the proposed PDFS-RideNN provides the accuracy value of 94.104%. From the above data, it is seen that the proposed PDFS-RideNN has achieved an improved accuracy value of 94.1048% at 80% training data.

The comparative method based on sensitivity is depicted in Figure 10b). While considering 70% of training data, the existing methods, like Extremely Randomized trees, RMD-SMOTE, DeepCNN, FCM-RideNN, and ANN-LM have sensitivity values of 63.39%, 68.672%, 72.423%, 76.416%, and 77.7929%, respectively. Meanwhile, the developed PDFS-RideNN has a value of 85.483% for the sensitivity metric for 70% of the training data.

The analysis in terms of the specificity metric is depicted in Figure 10c). Here, for the 80% training data, the methods, like Extremely Randomized trees, RMD-SMOTE, DeepCNN, FCM-RideNN, and ANN-LM acquired the specificity values of 84.030%, 87.124%, 89.843%, 90.077%, and 93.083%, respectively, but the proposed PDFS-RideNN has the specificity value of 94.824%.



a)

b)

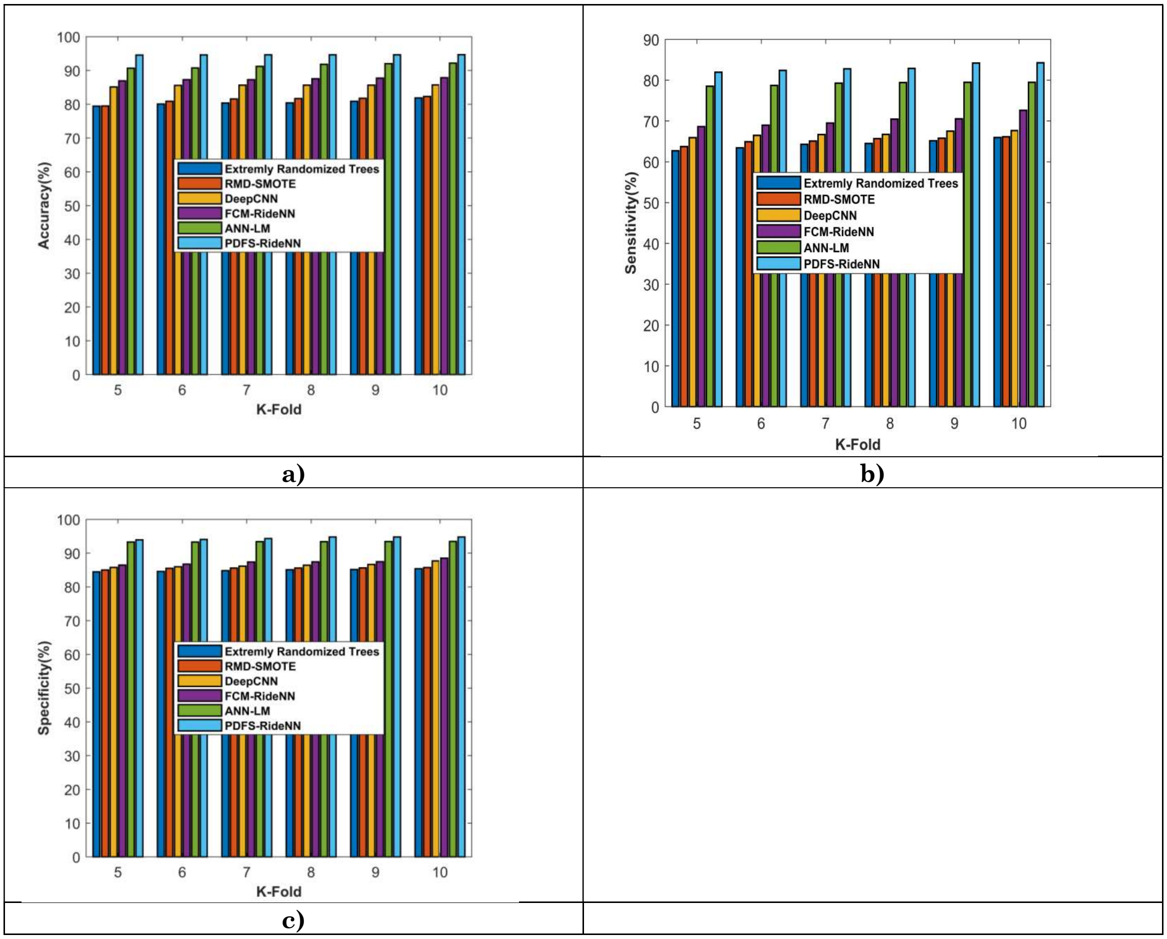
c)

Figure 9 Comparative analysis using Image-1 based on a) Accuracy, b) Sensitivity c) Specificity

b) Based on K-Fold: The comparative analysis of the developed method is analyzed based on accuracy, sensitivity, and specificity using Image-1 by varying the K-Fold as depicted in Figure 11. Figure 11a) illustrates the analysis based on accuracy by varying the K-Fold. When the K-Fold =5, the existing techniques, like Extremely Randomized trees, RMD-SMOTE, DeepCNN, FCM-RideNN, and ANN-LM have accuracy values of 79.439%, 79.495%, 85.118%, 89.906%, and 90.670%, respectively, which is comparatively lower than the proposed method. For the same training data, the developed PDFS-RideNN has an accuracy value of 94.54%. Similarly, when the K-Fold increased to 9, the methods, Extremely Randomized trees, RMD-SMOTE, Deep-CNN, FCM-RideNN, and ANN-LM acquired accuracy values of 80.871%, 81.744%, 85.646%, 87.737%, and 92%, while the proposed PDFS-RideNN attained the accuracy of 94.629%. From the above data, it is seen that the PDFS-RideNN has achieved an improved accuracy value of 94.65% when K-Fold=10.

The comparative method based on the sensitivity metric is depicted in Figure 11b). When K-Fold=6, the existing methods, like Extremely Randomized trees, RMD-SMOTE, Deep -CNN, FCM-RideNN, and ANN-LM acquired sensitivity values of 63.404%, 64.904%, 66.475%, 68.977%, and 78.712%, respectively. Meanwhile, the developed PDFS-RideNN attained a sensitivity value of 82.393%. When K-Fold=9, the comparative methods, like Extremely Randomized trees, RMD-SMOTE, Deep CNN, FCM-RideNN, and ANN-LM attained sensitivity values of 65.13%, 65.7874%, 67.4980%, 70.544%, and 79.4757%, respectively, whereas the sensitivity of the proposed 84.188%.

The analysis is based on specificity metrics depicted in Figure 11c). When the K-Fold is 7, the existing techniques, like Extremely Randomized trees, RMD-SMOTE, Deep CNN, FCM-RideNN, and ANN-LM have specificity values of 84.789%, 85.583%, 86.162%, 87.427%, and 93.412%, respectively, but the proposed PDFS-RideNN has the specificity value of 94.331%. When K-Fold=10 is considered, the existing techniques, such as Extremely Randomized trees, RMD-SMOTE, Deep CNN, FCM-RideNN, and ANN-LM achieved the values of specificity metrics 85.389%, 85.715%, 87.670%, 88.5%, and 93.457%, respectively. Meanwhile, the proposed PDFS-RideNN has a specificity value of 94.7973%. From the above data, the proposed PDFS-RideNN has achieved a maximum specificity value of 94.797 % at K-Fold=10.



a)

b)

c)

Figure 11 Comparative analysis using Image-1 based on a) Accuracy, b) Sensitivity c) Specificity

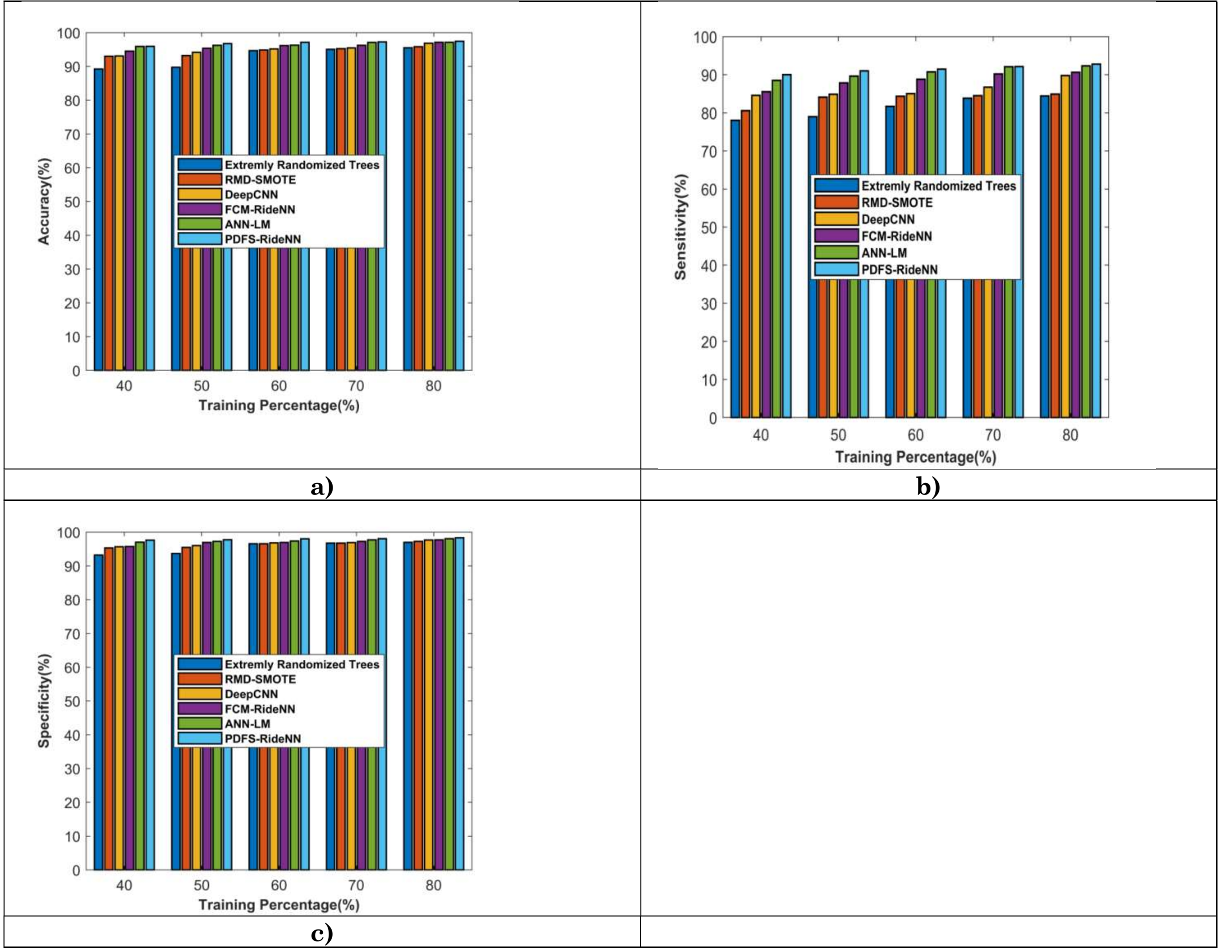
4.1.5.2 Analysis using Image-2

a) Based on the training percentage

The comparative analysis based on accuracy, sensitivity, and specificity using Image-2 is depicted in Figure 12. Figure 12a) shows the analysis based on accuracy by varying the training data percentage. For the training, data=40%, the existing techniques, like Extremely Randomized trees, RMD-SMOTE, DeepCNN, FCM-RideNN, and ANN-LM, have accuracy values of 89.219%, 93.0014%, 93.073%, 94.516%, and 95.88%, respectively, which is comparatively lower than the proposed method. For the same training data, the developed PDFS-RideNN has an accuracy value of 95.945%. Similarly, when the training data percentage is increased to 80%, Extremely Randomized trees, RMD-SMOTE, Deep CNN, FCM-RideNN, and ANN-LM have accuracy values of 95.49%, 95.81%, 96.846%, 97.117%, and 97.117%, and the proposed PDFS-RideNN has the accuracy value of 97.43%. From the above data, it is seen that the proposed PDFS-RideNN has achieved an improved accuracy value of 97.43% at 80% training data.

The comparative analysis based on the sensitivity metric is depicted in Figure 12b). While considering 70% of training data, the existing methods, like Extremely Randomized trees, RMD-SMOTE, Deep CNN, FCM-RideNN, and ANN-LM have sensitivity values of 83.82%, 84.483%, 86.718%, 90.210%, and 92.091%, respectively. Meanwhile, the developed PDFS-RideNN has a value of 92.128% for the sensitivity metric.

The analysis based on the specificity metric is depicted in Figure 12c). Here, for the 80% training data, the existing methods, like Extremely Randomized trees, RMD-SMOTE, Deep CNN, FCM-RideNN, and ANN-LM have specificity values of 96.966%, 97.231%, 97.669%, 97.692%, and 98.071%, but the proposed PDFS-RideNN has the specificity value of 98.327%, respectively.



a)

b)

c)

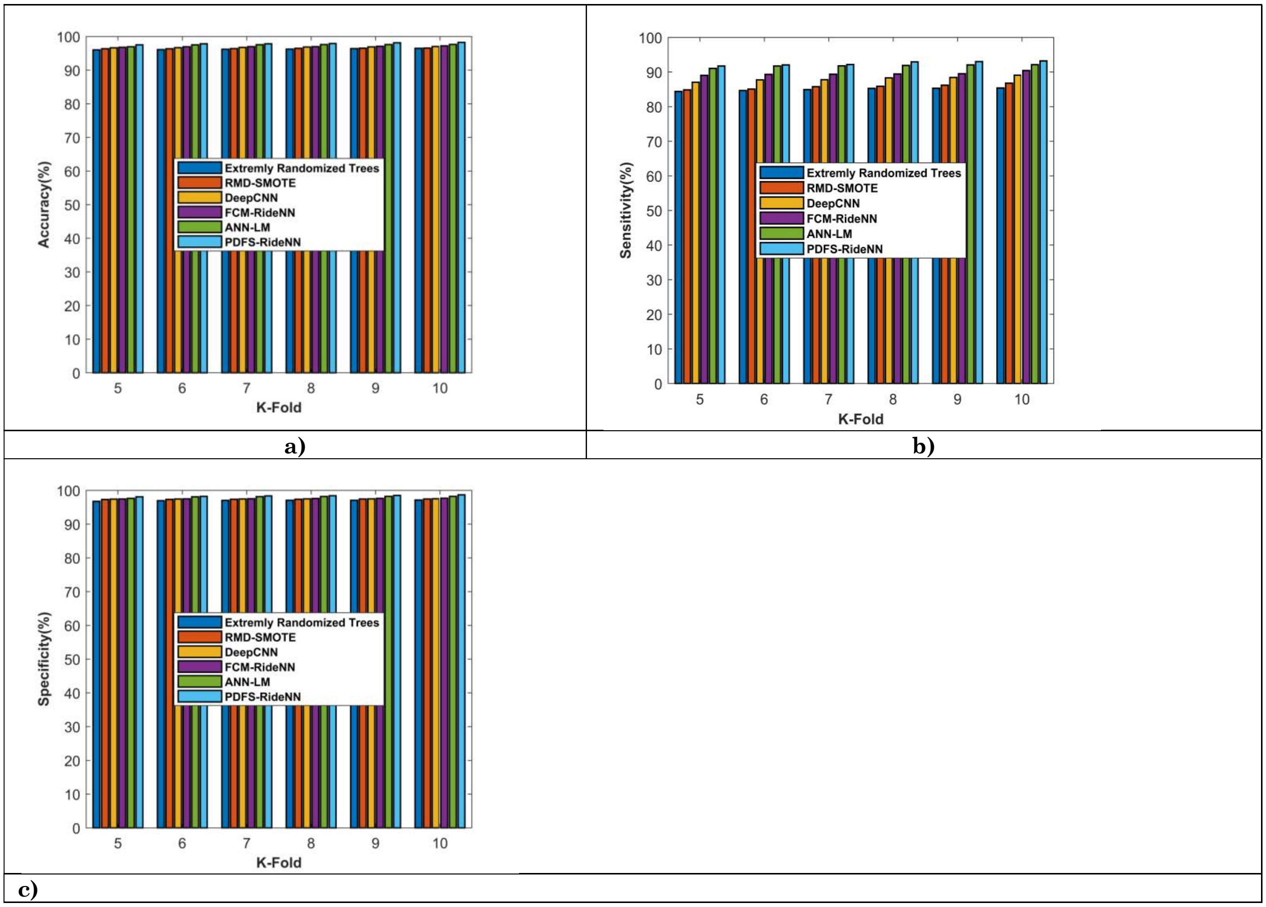
Figure 12 Comparative analysis using Image-2 based on a) Accuracy, b) Sensitivity c) Specificity

b) Based on K-Fold

The comparative analysis of the developed method is analyzed based on accuracy, sensitivity, and specificity using Image-2 by varying the K-Fold as depicted in Figure 13. Figure 13a) illustrates the analysis based on accuracy by varying the K-Fold. When the K-Fold =5, the existing techniques, like Extremely Randomized trees, RMD-SMOTE, Deep CNN, FCM-RideNN, and ANN-LM, have accuracy values of 96.015%, 96.360%, 96.634%, 96.768%, and 96.9516%, respectively, which is comparatively lower than the proposed method. For the same training data, the developed PDFS-RideNN has an accuracy value of 97.51%. Similarly, when K-Fold is increased to 9, the existing methods, such as Extremely Randomized trees, RMD-SMOTE, Deep CNN, FCM-RideNN, and ANN-LM, have accuracy values of 96.41%, 96.520%, 96.892%, 97.090%, and 97.594%, and the proposed PDFS-RideNN has the accuracy value of 98.145%. From the above data, it is seen that the PDFS-RideNN has achieved an improved accuracy value of 98.25% at K-Fold=10.

The comparative method based on the sensitivity metric is depicted in Figure 13b). While K-Fold =6 is considered, the existing methods, like Extremely Randomized trees, RMD-SMOTE, Deep CNN, FCM-RideNN, and ANN-LM have sensitivity values of 84.6493%, 85.056%, 87.726%, 89.317%, and 91.726%, respectively. Meanwhile, the developed PDFS-RideNN has a value of 92.03% for the sensitivity metric. For K-Fold=9, the comparative methods, like Extremely Randomized trees, RMD-SMOTE, Deep CNN, FCM-RideNN, and ANN-LM have sensitivity values of 85.29%, 86.185%, 88.434%, 89.506%, and 92.035%. For a similar case, the proposed PDFS-RideNN has a sensitivity value of 93.025%.

The analysis based on the specificity metric is depicted in Figure 13c). Here, for K-Fold=7, the existing techniques, like Extremely Randomized trees, RMD-SMOTE, Deep CNN, FCM-RideNN, and ANN-LM have specificity values of 97.017%, 97.356%, 97.453%, 97.528%, and 98.126%, respectively, but the proposed PDFS-RideNN has the specificity value of 98.376%. WhenK-Fold=10, the existing techniques, such as Extremely Randomized trees, RMD-SMOTE, Deep CNN, FCM-RideNN, and ANN-LM achieved the values of specificity metrics are 97.150%, 97.437%, 97.498%, 97.673%, and 98.228%, respectively. Meanwhile, the proposed PDFS-RideNN has a specificity value of 98.674%. From the above data, the proposed PDFS-RideNN has achieved a maximum specificity value of 98.674 % at K-Fold=10.



a)

b)

c)

Figure 13 Comparative analysis using Image-2 based on a) Accuracy, b) Sensitivity c) Specificity

4.2 Comparative discussion

Table 1 describes the comparative discussion to reveal the best performance attained by the brain tumor classification methods, in terms of accuracy, sensitivity, and specificity using image 1. Extremely randomized Trees acquired the accuracy, sensitivity, and specificity values of 82.482%, 65.960%, and 96.966%, respectively, while RMD-SMOTE attains the accuracy, sensitivity, and specificity metric values of 85.143%, 71.070%, and 97.231%. The accuracy, sensitivity, and specificity values of DeepCNN are 87.040%, 74.160%, and 97.669%, respectively. FCM-RideNNachieved the accuracy, sensitivity, and specificity values of 91.068%, 77.240%, and 97.692%, respectively, while ANN-LM attains the accuracy, sensitivity, and specificity values of 93.740%, 79.478%, and 98.071%. As depicted in the below table, the existing ANN-LM has comparatively better performance when compared with the existing techniques. Among all the comparative methods, the proposed PDFS-RideNN has improved performance with values of 94.650%, 86.135%, and 98.327% for accuracy, sensitivity, and specificity, respectively.

Table 1 Comparative analysis based on training percentage

Methods	Accuracy (%)	Sensitivity (%)	Specificity (%)
Extremely randomized Trees	95.495	84.395	96.966
RMD-SMOTE	95.810	84.917	97.231
DeepCNN	96.846	89.789	97.669
FCM-RideNN	97.115	90.604	97.692
ANN-LM	97.117	92.298	98.071
Proposed PDFS-RideNN	**97.432**	**92.774**	**98.327**

Table 2 describes the comparative discussion regarding the best performance attained by the brain tumor classification methods, in terms of accuracy, sensitivity, and specificity using image 2. Extremely randomized Trees achieved the accuracy, sensitivity, and specificity values of 96.492%, 85.384%, and 97.150%, respectively, while RMD-SMOTE attains accuracy, sensitivity, and specificity metric values of 96.572%, 86.766%, and 97.437%. The accuracy, sensitivity, and specificity values of DeepCNN are 97.006%, 89.789%, and 97.498%, respectively. FCM-RideNN has the accuracy, sensitivity, and specificity values of 97.217%, 90.604%, and 97.673%, and the ANN-LM provides the accuracy, sensitivity, and specificity values of 97.639%, 92.298%, and 98.228%. As depicted in the below table, the existing ANN-LM has comparatively better performance when compared with the existing techniques. Among all the comparative methods, the proposed PDFS-RideNN has improved performance with values of 98.252%, 93.185%, and 98.674% for accuracy, sensitivity, and specificity, respectively.

Table 2 Comparative analysis based on K-Fold

Methods	Accuracy (%)	Sensitivity (%)	Specificity (%)
Extremely randomized Trees	96.492	85.384	97.150
RMD-SMOTE	96.572	86.766	97.437
DeepCNN	97.006	89.789	97.498
FCM-RideNN	97.217	90.604	97.673
ANN-LM	97.639	92.298	98.228
Proposed PDFS-RideNN	**98.252**	**93.185**	**98.674**

5. Conclusion

This paper presents brain tumor classification through the segmentation, extraction, and training of data. Finally, the performance of the PDFS-RideNN is evaluated based on specificity, accuracy, and accuracy. The proposed method produces a maximal accuracy of 98.25%, maximal sensitivity of 93.18%, and maximal specificity of 98.67, which indicates the superiority of the proposed method. The future dimension of the research will be concentrated on extending the analysis using other standard databases with highly advanced features.

References

[1]R.Lavanyadevi, M.Machakowsalya, J.Nivethitha, and A. Niranjil Kumar, "Brain Tumor Classification and Segmentation in MRI Images using PNN", In proceedings of International Conference on Electrical, Instrumentation and Communication Engineering, 2017.

[2]Reema Mathew A, and Babu Anto P, "Tumor detection and classification of MRI brain image using wavelet transform and SVM", In proceedings of International Conference on Signal Processing and Communication , July 2017.

[3]Charutha S, and M. J. Jayashree, "An Efficient Brain Tumor Detection By Integrating Modified Texture Based Region Growing And Cellular Automata Edge Detection", In proceedings of International Conference on Control, Instrumentation, Communication and Computational Technologies, 2014.

[4]K. Sudharani, T.C. Sarma, and K.SatyaRasad ,"Intelligent Brain Tumor Lesion Classification and Identification from MRI Images using kNN technique", In proceedings of International Conference on Control,lnstrumentation, Communication and Computational Technologies, 2015.

[5]M. H. O. Rashid, M. A. Mamun, M. A. Hossain, and M. P. Uddin, "Brain Tumor Detection Using Anisotropic Filtering, SVM Classifier and Morphological Operation from MR Images", In proceedings of International Conference on Computer, Communication, Chemical, Material and Electronic Engineering (IC4ME2) , pp. 1-4, 2018.

[6]Ehab F. Badran, Esraa Galal Mahmoud, and Nadder Hamdy, "An Algorithm for Detecting Brain Tumors in MRI Images", In proceedings of International Conference on Computer Engineering and Systems (ICCES), pp. 368-373, 2010.

[7]R Sharmila , K Suresh Joseph, "Brain Tumour Detection of MR Image Using Naïve Beyer classifier and Support Vector Machine", International Journal of Scientific Research in Computer Science, Engineering and Information Technology, vol. 3, no.3, 2018.

[8]M. Karnan, T.Logheshwari, "Improved Implementation of Brain MRI image Segmentation using Ant Colony System", In proceedings of Computational Intelligence and Computing Research (ICCIC), pp. 1-4, 2010.

[9]Mustafa R. Ismael, AND Ikhlas Abdel-Qader,"BrainTumor Classification via Statistical Features and Back-Propagation Neural Network", In proceedings of International Conference on Electro/Information Technology (EIT), pp. 0252-0257, 2018.

[10]Abd-Ellah, M.K., Awad, A.I., Khalaf, A.A. and Hamed, H.F., “Design and implementation of a computer-aided diagnosis system for brain tumor classification”, In Proceedings of 28th International Conference on Microelectronics (ICM), pp. 73-76, December 2016.

[11]Nitish Zulpe, and Vrushsen Pawar, "GLCM Textural Features for Brain Tumor Classification", IJCSI International Journal of Computer Science Issues, vol. 9, no. 3, May 2012.

[12]Parveen, and Amritpal singh, "Detection of Brain Tumor in MRI Images, using Combination of Fuzzy C-Means and SVM",In proceedings of 2nd International Conference on Signal Processing and Integrated Networks (SPIN), 2015.

[13]Kharat, K.D., Kulkarni, P.P. and Nagori, M.B., “Brain tumor classification using neural network based methods”, International Journal of Computer Science and Informatics, vol.1, no.4, pp.2231-5292, 2012.

[14]Kaur, M. and Prajapat, B., “Automated Classification of Cancerous Brain Tumours Using Haarlet Transform and Probabilistic Neural Network”, In proceedings of International Conference on Advanced Computing Networking and Informatics , pp. 19-25, 2019.

[15]Anitha V, and Murugavalli, S.,”Brain tumour classification using two-tier classifier with adaptive segmentation technique”, IET computer vision, vol. 10, no.1, pp.9-17, 2016.

[16]Rehman, Z.U., Naqvi, S.S., Khan, T.M., Khan, M.A, and Bashir, T., “Fully automated multi-parametric brain tumour segmentation using superpixel based classification”, Expert Systems with Applications, pp.598-613, 2019.

[17]Soltaninejad, M., Yang, G., Lambrou, T., Allinson, N., Jones, T.L., Barrick, T.R., Howe, F.A. and Ye, X., “Supervised learning based multimodal MRI brain tumour segmentation using texture features from supervoxels” Computer methods and programs in biomedicine, vol.157, pp.69-84, 2018.

[18]Pinto, A., Pereira, S., Rasteiro, D. and Silva, C.A., “Hierarchical Brain Tumour Segmentation using Extremely Randomized Trees”, Pattern Recognition”, 2018.

[19]Soltaninejad, M., Yang, G., Lambrou, T., Allinson, N., Jones, T.L., Barrick, T.R., Howe, F.A. and Ye, X.,”Automated brain tumour detection and segmentation using superpixel-based extremely randomized trees in FLAIR MRI”, International journal of computer assisted radiology and surgery, vol.12, no.2, pp.183-203, 2017.

[20]Ma, C., Luo, G. and Wang, K.,”Concatenated and Connected Random Forests with Multiscale Patch Driven Active Contour Model for Automated Brain Tumor Segmentation of MR Images”, IEEE Transactions on Medical Imaging, 2018.

[21]Shenbagarajan, A., Ramalingam, V., Balasubramanian, C. and Palanivel, S., ”Tumor diagnosis in MRI brain image using ACM segmentation and ANN-LM classification techniques”, Indian Journal of Science and Technology, vol.9, no.1, 2016.

[22]Anbeek, P., Vincken, K.L. and Viergever, M.A., “Automated MS-lesion segmentation by k-nearest neighbor classification”, MIDAS Journal,2008.

[23]John, P.,”Braintumor classification using wavelet and texture based neural network” International Journal of Scientific & Engineering Research, vol.3, no.10, pp.1-7, 2012.

[24]Naik, J. and Patel, S., “Tumor detection and classification using decision tree in brain MRI”, International Journal of Computer Science and Network Security (IJCSNS), vol.14, no.6, pp.87. 2014.

[25]Patel, M.R. and Tse, V.,”Diagnosis and staging of brain tumors”, In Seminars in Roentgenology ,vol. 39, no. 3, pp. 347-360, July.2004.

[26]BraTS datasets taken from”https://www.smir.ch/BRATS/Start2015”, accessed on January 2019.

[27]Binu, D. and Kariyappa, B.S., ”RideNN: A New Rider Optimization Algorithm-Based Neural Network for Fault Diagnosis in Analog Circuits”, IEEE Transactions on Instrumentation and Measurement,2018.

[28]D. Jude Hemanth, and J. Anitha, "Modified Genetic Algorithm approaches for classification of abnormal Magnetic Resonance Brain tumor images", Applied Soft Computing Journal,2018.

[29] Tahmasebi; Hezarkhani, "A hybrid neural networks-fuzzy logic-genetic algorithm for grade estimation", Computers & Geosciences, vol. 42, pp. 18–27, 2012.

[30]BraTS datasets taken from”https://www.smir.ch/BRATS/Start2015”, accessed on January 2019.

[31] B. Jun, I. Choi and D. Kim, "Local Transform Features and Hybridization for Accurate Face and Human Detection," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 35, no. 6, pp. 1423-1436, June 2013.

[32]Jens Krommweh, "Tetrolet Transform: A New Adaptive Haar Wavelet Algorithm for Sparse Image Representation", Journal of Visual Communication and Image Representation, vol.21, no.4, pp.364-374, 2010.