Brain Tumor Classification Using Machine Learning Algorithms

B.Balakumar¹, P.Raviraj², E.Divya Devi³

¹Assistant Professor, Centre for Information Technology and Engineering, Manonmaniam Sundaranar University, Tirunelveli, India, hellobala2006@yahoo.co.in
²Professor, Department of CSE, Kalaignar Karunanidhi Institute of Technology, Coimbatore, Tamilnadu, India, drpraviraj@gmail.com
³PG Scholar, Centre for Information Technology and Engineering, Manonmaniam Sundaranar University, Tirunelveli, India, devidivya589@gmail.com

Abstract: A brain tumor is a collection of tissue that is grouped by a gradual addition of anomalous cells and it is important to classify brain tumors from the magnetic resonance imaging (MRI) for treatment. Human investigation is the routine technique for brain MRI tumor detection and tumors classification. Interpretation of images is based on organized and explicit classification of brain MRI and also various techniques have been proposed. Information identified with anatomical structures and potential abnormal tissues which are noteworthy to treat are given by brain tumor segmentation on MRI, the proposed system uses the adaptive pillar K-means algorithm for successful segmentation and the classification methodology is done by the two-tier classification approach. In the proposed system, at first the self-organizing map neural network trains the features extracted from the discrete wavelet transform blend wavelets and the resultant filter factors are consequently trained by the K-nearest neighbour and the testing process is also accomplished in two stages. The proposed two-tier classification system classifies the brain tumors in double training process which gives preferable performance over the traditional classification method. The classifiers can accurately classifying the status of the brain image into normal / abnormal.

Keywords: Image segmentation, MRI (Magnetic Resonance Imaging), Adaptive pillar k-means algorithm

1. INTRODUCTION

Brain tumor is one of the major causes of death among other types of the cancers. Proper and timely diagnosis can prevent the life of a person to some extent. Therefore we have proposed an automated reliable system for the diagnosis of the brain tumor. Proposed system is a two-tier system for brain tumor diagnosis and tumor region extraction. First, noise removal is performed as the pre-processing step on the brain MR images. Texture features are extracted from these noise free brain MR images. Next phase of the proposed system is Self-organizing Mapping based feature training and followed by the SVM based classification that is based on these extracted features. More than 89.5% accuracy is achieved by the classification phase. Results of the proposed technique show that tumor images are recognized quite accurately. This technique has been tested against the datasets of different patients received from Brainix dataset where the proposed work is tested on 30 normal images and 30 abnormal images.

2. LITERATURE SURVEY

The brain tumour identification from a MRI is a complex process hence artificial intelligence is applied to detect the exact tumour position in a brain MRI. In this paper we are developed a novel technique for the tumour segmentation and classification in brain MRI. The proposed approach for the brain MRI classification is implemented in the working platform of Matlab and the detailed explanation on the implementation performance measure is as follows. Brain tumour is most treatable and curable if captured at the initial stages of the infection. This locates exorbitant pressure on the brain, causing an incremented intracranial pressure and can cause permanent brain damage and eventually death. Untreated or progressive brain cancer can only spread inward because the skull will not let the brain tumour expands outward. Only invasive techniques such as biopsy and spinal tap methods can determine whether the brain tumour is cancerous or non-cancerous. However the system designed in this work avails segmenting the MR image with Adaptive Pillar K means clustering algorithm and classifying cancerous and non-cancerous brain tumors automatically by the two-tier classifier approach, which uses the statistical texture features extracted by DWT. To demonstrate and evaluate the
performance of the proposed system, mainly applied to brain MRI.
The skull captured in the brain MRI is removed by the skull stripping technique. The segmentation process is done by using Adaptive Pillar k means technique to optimize the computation time and hence improved the precision and enhance the quality of image segmentation. All classification result could have an error rate and on occasion will either fail to identify an abnormality, or identify an abnormality which is not present.

3. METHODOLOGY

3.1 Existing system
- Optical flow analysis
  Drawback: sensitive to noisy environment and it is not suitable for real time implementations
- Temporal difference analysis
  Drawback: The extracted shapes of moving objects are generally incomplete, especially when the moving objects in a scene are stationary or exhibit slow motion.

- Background subtraction
  Drawback: increased time complexity
- Bayesian approach
  Drawback: computational effort is high and not suitable for long video files.

3.2 Proposed system
- The multi-background generation module effectively generates a flexible probabilistic model through an unsupervised learning process to fulfill the property of either dynamic background or static background.
- The moving object detection module achieves complete and accurate detection of moving objects by only processing blocks that are highly likely to contain moving objects.

3.3 Advantages
- Very efficient
- Dynamic scenes are very easy to detect
- Low memory capacity needed
- Scalable
- Robust for any kind of video

4. PRE-PROCESSING TECHNIQUES

Brain MRIs are degraded during the process of imaging due to image transmission and image digitization by noise and existence of extra-cranial tissues in MRI such as Skull, bone, skin, air, muscles, and fat.

Pre-processing is a procedure to eliminate these noises and extra-cranial tissues from the Brain MRI and alters the heterogeneous image into homogeneous image. Though there are lots of filters which have been used for filtering the images, some of them corrupt the miniature details of the image and some conventional filters will process the image incessantly (smoothing) and consequently harden the edges of the image. Hence, the proposed pre-processing steps namely De-noising and skull stripping provide better Image clarity.

4.1 Sobel Operator

The operator consists of a pair of 3×3 convolution kernels as shown in Figure 1. One kernel is simply the other rotated by 90°.

\[
\begin{array}{ccc}
-1 & 0 & +1 \\
-2 & 0 & +2 \\
-1 & 0 & +1 \\
\end{array}
\]

\[
\begin{array}{ccc}
+1 & +2 & +1 \\
0 & 0 & 0 \\
-1 & -2 & -1 \\
\end{array}
\]

G_x  G_y

These kernels are designed to respond maximally to edges running vertically and horizontally relative to the pixel grid, one kernel for each of the two perpendicular orientations. The kernels can be applied separately to the input image, to produce separate measurements of the gradient component in each orientation (call these G_x and G_y). These can then be combined together to find the absolute magnitude of the gradient at each point and the orientation of that gradient. The gradient magnitude is given by:

\[
|G| = \sqrt{G_x^2 + G_y^2}
\]

Typically, an approximate magnitude is computed using:

\[
|G| = |G_x| + |G_y|
\]
which is much faster to compute.

The angle of orientation of the edge (relative to the pixel grid) giving rise to the spatial gradient is given by:

$$\theta = \arctan(G_y/G_x)$$

4.2 Robert's cross operator:

The Roberts Cross operator performs a simple, quick to compute, 2-D spatial gradient measurement on an image. Pixel values at each point in the output represent the estimated absolute magnitude of the spatial gradient of the input image at that point.

The operator consists of a pair of 2×2 convolution kernels as shown in Figure. One kernel is simply the other rotated by 90°. This is very similar to the Sobel operator.

$$\begin{bmatrix}
+1 & 0 \\
0 & -1
\end{bmatrix} \quad \begin{bmatrix}
0 & +1 \\
-1 & 0
\end{bmatrix}
$$

These kernels are designed to respond maximally to edges running at 45° to the pixel grid, one kernel for each of the two perpendicular orientations. The kernels can be applied separately to the input image, to produce separate measurements of the gradient component in each orientation (call these $G_x$ and $G_y$). These can then be combined together to find the absolute magnitude of the gradient at each point and the orientation of that gradient. The gradient magnitude is given by:

$$|G| = \sqrt{G_x^2 + G_y^2}$$

although typically, an approximate magnitude is computed using:

$$|G| = |G_x| + |G_y|$$

which is much faster to compute.

The angle of orientation of the edge giving rise to the spatial gradient (relative to the pixel grid orientation) is given by:

$$\theta = \arctan(G_y/G_x) - 3\pi/4$$

4.3 Prewitt's operator:

Prewitt operator is similar to the Sobel operator and is used for detecting vertical and horizontal edges in images.

$$h_1 = \begin{bmatrix}
1 & 1 & 1 \\
0 & 0 & 0 \\
-1 & -1 & -1
\end{bmatrix} \quad h_3 = \begin{bmatrix}
-1 & 0 & 1 \\
-1 & 0 & 1 \\
-1 & 0 & 1
\end{bmatrix}$$

4.4 Laplacian of Gaussian:

The Laplacian is a 2-D isotropic measure of the 2nd spatial derivative of an image. The Laplacian of an image highlights regions of rapid intensity change and is therefore often used for edge detection. The Laplacian is often applied to an image that has first been smoothed with something approximating a Gaussian Smoothing filter in order to reduce its sensitivity to noise. The operator normally takes a single gray level image as input and produces another gray level image as output.

The Laplacian $L(x, y)$ of an image with pixel intensity values $I(x, y)$ is given by:

$$L(x, y) = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}$$

Since the input image is represented as a set of discrete pixels, we have to find a discrete convolution kernel that can approximate the second derivatives in the definition of the Laplacian. Three commonly used small kernels are shown in Figure 1.
Figure 1 Three commonly used discrete approximations to the Laplacian filter.

Because these kernels are approximating a second derivative measurement on the image, they are very sensitive to noise. To counter this, the image is often Gaussian Smoothed before applying the Laplacian filter. This pre-processing step reduces the high frequency noise components prior to the differentiation step.

In fact, since the convolution operation is associative, we can convolve the Gaussian smoothing filter with the Laplacian filter first of all, and then convolve this hybrid filter with the image to achieve the required result. Doing things this way has two advantages:

- Since both the Gaussian and the Laplacian kernels are usually much smaller than the image, this method usually requires far fewer arithmetic operations.
- The LOG (`Laplacian of Gaussian') kernel can be pre-calculated in advance so only one convolution needs to be performed at run-time on the image.

Denoising

In the pre-processing of brain MRI, the noise will be removed by utilizing the non-local mean filter which does not update a pixel’s value with an average of the pixels around it, instead updates it using a weighted average of the pixels judged to be most kindred. The weight of each pixel depends on the distance between its intensity grey level vector and that of the target pixel. De-noised image of each pixel $i$ of the non-local means is computed with the following equation

$$N(i, j) = \sum_{j \in D} w(i, j)D(i, j)$$

Where, $j$ is the noisy image and $N$ is the de-noised image, and weights $w(i, j)$ meet the following conditions $0 \leq w(i, j) \leq 1$.

Adaptive pillar K means algorithm with the efficient distance metric

The system utilizes the authentic size of the image to perform high quality image segmentation which causes high-resolution image data points to be clustered. Therefore utilize the K-means algorithm for clustering image data by considering that it has ability to cluster immensely colossal data and additionally outliers’ payments are utilized expeditiously and efficiently. Because of starting points engendered arbitrarily, one of the local minima leads to erroneous clustering results hence K-means algorithm is arduous to reach global optimum. To evade this phenomenon, the proposed system utilizes adaptive pillar algorithm, which is very robust and superior for initial clusters optimization for K-means by deploying all centroids far discretely among them in the data distribution. This algorithm is inspired by the cerebration process of determining a set of pillars’ locations to make a stable house or building. In the proposed adaptive pillar K-means algorithm we modified the pillar K-means algorithm in [26]. In the proposed adaptive pillar K-means algorithm we find out the average mean of the data point instead of grand mean in the previous algorithm. The average mean based initial centroid point selection can improve the performance of the clustering than the grand mean based existing method. Locate two, three, and four pillars, to withstand the pressure distributions of several different roof structures composed of discrete points. It is inspiring that by distributing the pillars as far as possible from each other within a roof, as number of centroids among the gravity weight of data distribution in the vector space the pillars can withstand the roof’s pressure and stabilize a house or building. Therefore, this algorithm designates positions of initial centroids in the farthest accumulated distance between them in the data distribution.
GLCM Features

Gray co matrix calculates the GLCM from a scaled version of the image. By default, if I is a binary image, gray co matrix scales the image to two gray-levels. If I is an intensity image, gray co matrix scales the image to eight gray-levels. You can specify the number of gray-levels gray co matrix uses to scale the image by using the 'Num Levels' parameter, and the way that gray co matrix scales the values using the 'Gray Limits' parameter.

The following figure shows how gray co matrix calculates several values in the GLCM of the 4-by-5 image I. Element (1,1) in the GLCM contains the value 1 because there is only one instance in the image where two, horizontally adjacent pixels have the values 1 and 1. Element (1,2) in the GLCM contains the value 2 because there are two instances in the image where two, horizontally adjacent pixels have the values 1 and 2. Gray co matrix continues this processing to fill in all the values in the GLCM.

For quantitatively describing the first order statistical features of the image, useful features of the image can be obtained from the histogram. Mean is the average value of the intensity of the image. Variance tells the intensity variation around the mean. Skewness is the measure which tells the symmetries of the histogram around the mean. Kurtosis is the flatness of the histogram. Uniformity of the histogram is represented by the entropy. Following is the list of features obtained using histogram of the image.

\[
\text{Mean} : \mu = \frac{1}{G-1} \sum_{i=0}^{G-1} I(i)
\]

\[
\text{Variance} : \sigma^2 = \frac{1}{G-1} \sum_{i=0}^{G-1} (I(i) - \mu)^2
\]

\[
\text{Skewness} : \mu_3 = \frac{1}{G-1} \sum_{i=0}^{G-1} (I(i) - \mu)^3
\]

\[
\text{Kurtosis} : \mu_4 = \frac{1}{G-1} \sum_{i=0}^{G-1} (I(i) - \mu)^4 - 3
\]

\[
\text{Energy} : E = \sum_{i=0}^{G-1} [p(i)]^2
\]

\[
\text{Entropy} : H = -\sum_{i=0}^{G-1} p(i) \log_2[p(i)]
\]

Histogram based features are local in nature. These features do not consider spatial information into consideration. So for this purpose gray-level spatial co-occurrence matrix \( h_{d(i,j)} \) based features are defined which are known as second order histogram based features. These features are based on the joint probability distribution of pairs of pixels. Distance and angle \( \theta \) within a given neighborhood are used for calculation of joint probability distribution between pixels [21]. Normally \( d = 1,2 \) and \( \theta = 0^\circ, 45^\circ, 90^\circ \) and \( 135^\circ \) are used for calculation. Texture features can be described using this co-occurrence matrix. Following equations define these features.
Angular second moment (energy): $\sum_{i=0}^{G-1} \sum_{j=0}^{G-1} [p(i,j)]^2$

Correlation: $\sum_{i=0}^{G-1} \sum_{j=0}^{G-1} \frac{i \cdot p(i,j) - \mu_x \mu_y}{\sigma_x \sigma_y}$

Inertia: $\sum_{i=0}^{G-1} (i-j)^2 p(i,j)$

Absolute value: $\sum_{i=0}^{G-1} \sum_{j=0}^{G-1} |i-j| p(i,j)$

Inverse difference: $\sum_{i=0}^{G-1} \sum_{j=0}^{G-1} \frac{p(i,j)}{1+(i-j)^2}$

Entropy: $H = -\sum_{i=0}^{G-1} \sum_{j=0}^{G-1} p(i,j) \log_2[p(i,j)]$

Maximum probability: $\max_{i,j} p(i,j)$

**Self-organizing map (SOM) based initial training**

A self-organizing map (SOM) or self-organizing feature map (SOFM) is a type of artificial neural network (ANN) that is trained using unsupervised learning to produce a low-dimensional (typically two-dimensional), discretized representation of the input space of the training samples, called a map. Self-organizing maps differ from other artificial neural networks as they apply competitive learning as opposed to error-correction learning (such as back propagation with gradient descent), and in the sense that they use a neighborhood function to preserve the topological properties of the input space.

A self-organizing map showing U.S. Congress voting patterns visualized in Synapse. The first two boxes show clustering and distances while the remaining ones show the component planes. Red means a yes vote while blue means a no vote in the component planes (except the party component where red is Republican and blue is Democratic).

This makes SOMs useful for visualizing low-dimensional views of high-dimensional data, akin to multidimensional scaling. The artificial neural network introduced by the Finnish professor Teuvo Kohonen in the 1980s is sometimes called a Kohonen map or network. The Kohonen net is a computationally convenient abstraction building on work on biologically neural models from the 1970s and morphogenesis models dating back to Alan Turing in the 1950s.

Like most artificial neural networks, SOMs operate in two modes: training and mapping. "Training" builds the map using input examples (a competitive process, also called vector quantization), while "mapping" automatically classifies a new input vector.

A self-organizing map consists of components called nodes or neurons. Associated with each node is a weight vector of the same dimension as the input data vectors, and a position in the map space. The usual arrangement of nodes is a two-dimensional regular spacing in a hexagonal or rectangular grid. The self-organizing map describes a mapping from a higher-dimensional input space to a lower-dimensional map space. The procedure for placing a vector from data space onto the map is to find the node with the closest (smallest distance metric) weight vector to the data space vector.

While it is typical to consider this type of network structure as related to feed forward networks where the nodes are visualized as being attached, this type of architecture is fundamentally different in arrangement and motivation. Useful extensions include using toroidal grids where opposite edges are connected and using large numbers of nodes.

It has been shown that while self-organizing maps with a small number of nodes behave in a way that is similar
to K-means, larger self-organizing maps rearrange data in a way that is fundamentally topological in character.

It is also common to use the U-Matrix.\cite{5} The U-Matrix value of a particular node is the average distance between the node’s weight vector and that of its closest neighbors. In a square grid, for instance, we might consider the closest 4 or 8 nodes (the Von Neumann and Moore neighborhoods, respectively), or six nodes in a hexagonal grid.

Training is an iterative process which requires a lot of computational effort and thus is time-consuming. The extracted features are trained by drawing sample vectors from the input dataset and ‘teaching’ them to the classifier. The teaching is the process of choosing a winner unit by means of a similarity measure and updating the values of codebook vectors in the neighborhood of the winner unit and it is repeated a number of times. The SOM is a neural network model belongs to the category of competitive learning networks and it is predicated on unsupervised learning, that denotes no human intervention is needed during the learning but little needs to be known about the characteristics of the input data. The SOM can be used for clustering data without knowing the class memberships of the input data and acclimated to detect features innate to the quandary and thus has additionally been called self-organizing feature map.

\[
\|d(z)\| = \sqrt{\sum_{i=1}^{n} (d_i(z))^2}
\]

Then, the Euclidean distance can define in terms of the Euclidean norm:

\[
d_E(d_1(z), d_2(z)) = \|d_1(z) - d_2(z)\|
\]

The BMU is the codebook vector usually noted as \(m_c\) that is a bestmatch of a given input vector \(d(z)\).

\[
\|d(z) - m_c\| = \min_i \{\|d(z) - m_i\|\}
\]

After finding the BMU, units in the SOM are updated and the BMU updated to be a little closer to the sample vector in the input space. Similarly the topological neighbors of the BMU are also updated. This update procedure stretches the BMU and its topological neighbors towards the sample vector. The update procedure is illustrated in Fig. 3.4 in that the codebook vectors are situated in the crossings of the solid lines. The lines drawn in figure represent the topological relationships of the SOM. The input fed to the network is marked by \(d(z)\) in the input space and the BMU, or the winner neuron is the codebook vector closest to the sample in the middle above \(d(z)\). The winner neuron and its topological neighbors are updated by moving them a little towards the input sample, which is represented in figure by dashed lines. The neighborhood in this case consists of eight neighboring units mentioned in the figure.

\[
\begin{align*}
\text{Figure: SOM architecture} \\
\text{In the initial training, the extracted feature vector is drawn} \\
\text{randomly from the input data set, then it is fed to all units in} \\
\text{the network and a similarity measure is calculated between} \\
\text{the input data sample and all the codebook vectors. The} \\
\text{codebook vector is chosen as the best-matching unit (BMU) with} \\
\text{greatest similarity with input sample. The similarity is} \\
\text{usually defined by means of a distance measure, for example} \\
\text{in the case of Euclidean distance the BMU is the closest} \\
\text{neuron to the sample in the input space. Consider } AZ = \{d(z),} \\
\text{\(n = 1, 2, \ldots N\) is the extracted MRI feature vectors and the} \\
\text{Euclidean norm of the vector } d(z) \text{ is defined as}
\end{align*}
\]

\[
\begin{align*}
\text{Figure: updating the best matching unit and its neighbors} \\
\text{The BMU among all the neurons and updating the codebook} \\
\text{vectors in the neighborhood of the winner unit are the major} \\
\text{computational effort of SOM. If the neighborhood is large in}
\end{align*}
\]
the beginning of the training process, then lot of codebook
vectors to be updated and if the network is large, then
relatively larger portion of the time is spent for looking a
winner neuron. Time to be spent on each of these phases
depends on particular software and hardware used.

**Classification:**

In machine learning and statistics, classification is the
problem of identifying to which of a set of categories (sub-
populations) a new observation belongs, on the basis of
a training set of data containing observations (or instances)
whose category membership is known.

An example would be assigning a given email into "spam" or
"non-spam" classes or assigning a diagnosis to a given patient
as described by observed characteristics of the patient
gender, blood pressure, presence or absence of certain
symptoms, etc.). Classification is an example of pattern
recognition.

In the terminology of machine learning, classification is
considered an instance of supervised learning, i.e. learning
where a training set of correctly identified observations is
available. The corresponding unsupervised procedure
is known as clustering, and involves grouping data into
categories based on some measure of inherent similarity
or distance.

![Figure: Concept of machine learning based classification](image)

**Support Vector Machine Classifier**

Support vector machines (SVMs) are a set of supervised
learning methods used for classification, regression and
outliers detection. More formally, a support vector machine
constructs a hyperplane or set of hyper planes in a high-
or infinite-dimensional space, which can be used for
classification, regression, or other tasks. Intuitively, a good
separation is achieved by the hyperplane that has the largest
distance to the nearest training-data point of any class (so-
called functional margin), since in general the larger the
margin the lower the generalization error of the classifier.

**SVM Algorithm**

Classifying data is a common task in machine learning.
Suppose some given data points each belong to one of two
classes, and the goal is to decide which class a new data point
will be in. In the case of support vector machines, a data point
is viewed as a p-dimensional vector , and we want to know
whether we can separate such points with a (p-1)-dimensional
hyperplane. This is called a linear classifier. There are many
hyperplanes that might classify the data. One reasonable
choice as the best hyperplane is the one that represents the
largest separation, or margin, between the two classes. So we

to previous observations by means of a similarity or distance
function.

An algorithm that implements classification, especially in a
concrete implementation, is known as a classifier. The term
"classifier" sometimes also refers to the mathematical function, implemented by a classification
algorithm, that maps input data to a category.

Terminology across fields is quite varied. In statistics, where
classification is often done with logistic regression or a
similar procedure, the properties of observations are
termed explanatory variables (or independent variables,
regressors, etc.), and the categories to be predicted are known
as outcomes, which are considered to be possible values of
the dependent variable. In machine learning, the observations
are often known as instances, the explanatory variables are
termed features (grouped into a feature vector), and the
possible categories to be predicted are classes. Other fields
may use different terminology: e.g. in community ecology,
the term "classification" normally refers to cluster analysis.
choose the hyper plane so that the distance from it to the nearest data point on each side is maximized. If such a hyperplane exists, it is known as the maximum-margin hyperplane and the linear classifier it defines is known as a maximum margin classifier; or equivalently, the perception of optimal stability.

![Figure: Linearly Separable patterns](image)

**Applications**

SVMs are helpful in text and hypertext categorization as their application can significantly reduce the need for labeled training instances in both the standard inductive and transductive settings.

Classification of images can also be performed using SVMs. Experimental results show that SVMs achieve significantly higher search accuracy than traditional query refinement schemes after just three to four rounds of relevance feedback. This is also true of image segmentation systems, including those using a modified version SVM that uses the privileged approach as suggested by Vapnik. The SVM algorithm has been widely applied in the biological and other sciences. They have been used to classify proteins with up to 90% of the compounds classified correctly. Permutation tests based on SVM weights have been suggested as a mechanism for interpretation of SVM models. Support vector machine weights have also been used to interpret SVM models in the past. Posthoc interpretation of support vector machine models in order to identify features used by the model to make predictions is a relatively new area of research with special significance in the biological sciences.

**Advantages of support vector machines are:**

- Effective in high dimensional spaces.
- Still effective in cases where number of dimensions is greater than the number of samples.
- Uses a subset of training points in the decision function (called support vectors), so it is also memory efficient.
- Versatile: different Kernel functions can be specified for the decision function. Common kernels are provided, but it is also possible to specify custom kernels.

**Disadvantages of support vector machines:**

If the number of features is much greater than the number of samples, the method is likely to give poor performances.

SVMs do not directly provide probability estimates, these are calculated using an expensive five-fold cross-validation.

5. **SCREENSHOTS**

The proposed system is implemented using a MATLAB 2014a. The algorithm is tested on a BRAINX database of 397 images where it is evaluated for brain tumor detection. These images are the most widely used standard test images used for image retargeting algorithms. The images contain a nice mixture of detail, flat regions, shading and texture that do a good job of testing various image processing algorithms. These images are used for many image processing researches.

![Figure: Test images (Top row: normal images & Bottom row: Abnormal images)](image)
This section elaborates the results obtained for the proposed work and the proposed work is implemented on MATLAB GUI shown in below figures.

![GUI for Pre-processing](image1)

![GUI with Layer segmentation](image2)

The following table 4.1 presenting the features distribution of normal and tumorous images in this proposed work.

**Table : Feature matrix of 30 sample**

![GUI for Two-Tier classification](image3)

The accuracy of the proposed work is show in the following plots.

6. **CONCLUSION**

This proposed work genesis an efficient recognition system for the brain tumor classification and the by not focusing on the traditional way, this work travels on two tier classification method. This work presented for totally 60 images of both normal and abnormal images and MATLAB image processing toolbox is useful for developing the proposed work. Firstly the brain MRI images are pre-processed because it is well-known to every one about the noises present in the MRI images. The pre-processed images are undergone for the segmentation where the brain ROI extracted precisely. The feature extraction is done by processing the GLC matrix and the shape and texture features are collected for every image. The collected features vectors are trained with the help of SOM neural network where the variance among the feature set is increased tightly and making the classifier learning rate as high. The SVM classifier is effectively classifying the images very accurately and the also it assures maximum accuracy as 89.5%.

7. **FUTURE WORK**

This proposed work is focusing the traditional parametric feature extraction technique for this recognition task. The future work may be extended this feature extraction process in terms of deep learning neural network. Convolution neural networks are the most delicate network for the field of image processing and also the algorithm need to be extended for segmenting the tumorous area.
REFERENCES


