CHAPTER 3

PERFORMANCE ANALYSIS OF CLUSTERING ALGORITHMS IN BRAIN TUMOR DETECTION FROM MR IMAGES

3.1 BRAIN TUMORS

Brain is the anterior most part of the central nervous system. Along with the spinal cord, it forms the Central Nervous System (CNS). The Cranium, a bony box in the skull protects it. Virtually everything is controlled by the brain. Succinctly put, the brain is our survival kit (Badran et al 2010)

Brain tumors are the tumors that grow in the brain. Tumor is an abnormal growth caused by cells reproducing themselves in an uncontrolled manner. A benign brain tumor consists of benign (harmless) cells and has distinct boundaries. Surgery alone may cure this type of tumor. A malignant brain tumor (Akram et al 2011) is life-threatening. It may be malignant because it consists of cancer cells, or it may be called malignant because of its location. A malignant brain tumor made up of cancerous cells may spread or seed to other locations in the brain or spinal cord. It can invade and destroy healthy tissue malfunction. The structure and function of the brain can be studied noninvasively by doctors and researchers using Magnetic Resonance Imaging (MRI) (Baune et al 2009) Figure 3.1 is a Gray scale MR image of thin horizontal slice of the brain. Magnetic Resonance Imaging (MRI), strongly depend on computer technology to generate or display digital images. Segmentation is an important process in most medical image analysis.
Clustering to Magnetic Resonance (MR) brain tumors maintains efficiency. Clustering is suitable for biomedical image segmentation (Bahreini et al 2010) as the number of clusters is usually known for images of particular regions of the human anatomy.

The various problems in the existing processes are listed (Iftekharuddin et al 2006 and Alan Wee-Chung Liew and Hong Yan 2006):

- It is very difficult to conduct surgery without using Image Processing techniques
- Structures like tumor, brain tissue and skull cannot be identified without Image Segmentation
- In MRI images, the amount of data is too much for manual interpretation and analysis
- It takes a long time for diagnosis without Image Processing Techniques
- There is a chance of wrong diagnosis without Image Processing Techniques
3.2 PROPOSED SYSTEM

This system uses color-based segmentation method. This system analyses various clustering techniques to track tumor objects in Magnetic Resonance (MR) brain images. The input to this system is the MRI image of the axial view of the human brain. The Clustering algorithms used are K-means, SOM, Hierarchical Clustering and fuzzy C-Means Clustering. A given gray-level MR image is converted into a color space image and clustering algorithms are applied. The position of tumor objects is separated from other items of an MR image by using clustering algorithms and histogram-clustering. This method distinguishes exactly the lesion. Fuzzy C-Means clustering operates on gray scale MRI image. Then tumor detection is performed, based on improved fuzzy classification. The result constitutes the initialization of a segmentation method based on a deformable model, leading to a precise segmentation of the tumors. Imprecision and variability are taken into account at all levels, using appropriate fuzzy models (Murugavalli and Rajamani 2007). Figure 3.2 depicts the System Model of the proposed system. The three stages of the system are,

- Pseudo Color Translation (RGB Color Space) (Ming-Ni Wu et al 2007)
- Color Space Translation (L* a* b* Color Space) (Ming-Ni Wu et al 2007)
- Fuzzy C Means Clustering Algorithm (Gopal and Karnan 2010).

In this system, efficient Clustering Algorithms are used for the Brain tumor (Murugesan and Sukanesh 2009) detection Process. After the clustering process, the cluster containing the tumor is selected as the primary segment. To eliminate the pixels which are not related to the tumor pixels, Histogram clustering is applied. The performance analysis is conducted by taking MRI Brain tumor image as the input and applying all the four
clustering algorithms on the MRI Brain Image. The performance of the above four clustering algorithms are found based on Sensitivity, Specificity and Accuracy. The efficiency of all the four algorithms in this system is found by applying all the four algorithms to a database of 100 MR Images collected from Medical experts and from the web medical database.

![System Model Diagram]

**Figure 3.2 System Model**

### 3.3 IMPLEMENTATION DETAILS

The implementation of the proposed system include following phases.

1. Pseudo Color Translation
2. Color Space Translation
3. Implementation of Clustering Algorithms
   - Implementation of K-means Algorithm
   - Implementation of Self Organizing Maps
   - Implementation of Hierarchical Clustering and
   - Implementation of Fuzzy C-Means Algorithm

4. Cluster Selection

5. Histogram Clustering and

6. Region Elimination

3.3.1 Pseudo Color Translation

Original MR Brain image is a gray-level image, which is insufficient to support fine features. To obtain more useful features and enhance the visual density, the proposed method applies pseudo-color transformation, a mapping function that maps a gray-level pixel to a color-level pixel by a lookup table in a predefined color map. An RGB color map (Asad et al 2011) contains R, G, and B values for each item. Each gray value maps to an RGB item. The proposed method has adopted the standard RGB color map, which gradually maps gray-level values 0 to 255 into blue-to-green-to-red color.

3.3.2 Color Space Translation

To retrieve important features to benefit the clustering process, the RGB color space is further converted to a CIELab color model ($L^{*}a^{*}b^{*}$) (Badran et al 2010). The $L^{*}a^{*}b^{*}$ space consists of a

where, $L^{*}$ - luminosity layer,

$a^{*}$ - chromaticity layer, which indicates where color falls along the red-green axis,

$b^{*}$ - chromaticity-layer, which indicates where the color falls along the blue-yellow axis.
Steps for converting RGB to L*a*b*

Step1:

Convert RGB to WYZ

\[ W = 0.4303R + 0.3416G + 0.1784B \]  \hspace{1cm} (3.1)

\[ Y = 0.2219R + 0.7068G + 0.0713B \]  \hspace{1cm} (3.2)

\[ Z = 0.0202R + 0.1296G + 0.9393B. \]  \hspace{1cm} (3.3)

Step2:

The CIELab color model is calculated as

\[ L^* = 116(h(Y/Y_s)) - 16 \]  \hspace{1cm} (3.4)

\[ a^* = 500(h(W/W_s)) - h(Y/Y_s) \]  \hspace{1cm} (3.5)

\[ b^* = 200(h(Y/Y_s)) - h(Z/Z_s) \]  \hspace{1cm} (3.6)

\[ h(q) = \begin{cases} 3\sqrt{q} & q > 0.008856 \\ 7.787q + \frac{16}{116} & q \leq 0.008856 \end{cases} \]  \hspace{1cm} (3.7)

Equations (3.1) to (3.3) show the equations for converting RGB to WYZ. \(Y_s, W_s,\) and \(Z_s\) in Equations (3.4) to (3.6) are the standard stimulus coefficients. Both the \(a^*\) and \(b^*\) layers contain all required color information.

3.3.3 Clustering Algorithms

Clustering algorithms used for the implementation are,

- K-means Algorithm
- Self Organizing Maps
- Hierarchical Clustering and
- Fuzzy C-Means Algorithm
3.3.3.1 Implementation of K-means clustering segmentation

Algorithm:

- The initial partitions are chosen by getting the R, G, B values of the pixels.
- Every pixel in the input image is compared against the initial partitions using the Euclidian Distance and the nearest partition is chosen and recorded (Lue Vincent and Pierre Soille 1991, Ming-Ni Wu et al 2007).
- Then, the mean in terms of RGB color of all pixels within a given partition is determined. This mean is then used as the new value for the given partition.
- Once the new partition values have been determined, the algorithm returns to assigning each pixel to the nearest partition.
- The algorithm continues until the pixels no longer change their partition to which they are associated with or until none of the partition value changes.

3.3.3.2 Implementation of self organizing maps

Self-organizing maps (SOMs) (Murugavalli and Rajamani 2007) are data visualization techniques invented by Professor Teuvo Kohonen which reduce the dimensions of data through the use of self-organizing neural networks. The problem that data visualization attempts to solve is that humans simply cannot visualize high dimensional data. So techniques are created to help us understand this high dimensional data. The way SOMs go about reducing dimensions is by producing a map of usually 1 or 2 dimensions which plot the similarities of the data by grouping similar data items together. So SOMs accomplish two tasks: they reduce dimensions and
display similarities. Colors are grouped together such as the greens are all in the top left corner and the purples are all grouped around the bottom right corner. Figure 3.3 depicts the example of SOM clustering.

![Figure 3.3 SOM Clustering](image)

**Components of SOM**

**A. Sample Data:** The first part of a SOM (Murugavalli and Rajamani 2007) is the data. Above shown is the example of 3 dimensional data which are commonly used when experimenting with SOMs.

Here the colors are represented in three dimensions (Red, Blue, and Green). Figure 3.4 depicts the SOM Input.

![Figure 3.4 SOM Input](image)

**B. Weights:** The second component to SOMs is the weight vectors (Yuan Jiang et al 2003). Each weight vector has two components to them. The first part of a weight vector is its data. This is of the same dimensions as the sample vectors and the second part of a weight vector is its natural
location. The good thing about colors is that the data can be shown by displaying the color, so in this case the color is the data, and the location is the x, y position of the pixel on the screen.

Figure 3.5 is a skewed view of a grid where the n-dimensional array for each weight and each weight has its own unique location in the grid. Weight vectors do not necessarily have to be arranged in two dimensions. The data part of the weight must be of the same dimensions as the sample vectors. Weights are sometimes referred as Neurons since SOMs are actually Neural Networks.

![SOM Weights](image)

**Figure 3.5 SOM Weights**

**SOM Clustering:** The way that SOMs go about organizing themselves is by competing for representation of the samples. Neurons are also allowed to change themselves by learning to become more like samples in hopes of winning the next competition. It is this selection and learning process that makes the weights organize themselves into a map representing similarities. So with these two components (the sample and weight vectors), one can order the weight vectors in such a way that they will represent the similarities of the sample vectors. This is accomplished by using the very simple algorithm shown below
**Algorithm**

1. Map is initialized
2. For ‘t’ from 0 to 1 randomly a sample is selected
3. Best matching unit is selected
4. Neighbors are scaled
5. t is increased by a small amount
6. End for

This whole process is then repeated a large number of times, usually more than 1000 times.

**Initialize Map:** The first step in constructing a SOM is to initialize the weight vectors.

**Selecting a Sample:** Select a sample vector randomly and search the map of weight vectors to find which weight best represents that sample.

**Getting the Best Matching Unit (BMU):** Since each weight vector has a location, it also has neighboring weights that are close to it. The weight that is chosen is rewarded by being able to become more like that randomly selected sample vector.

**Scaling the neighbors:** In addition to this reward, the neighbors of that weight are also rewarded by being able to become more like the chosen sample vector. From this step increase ‘t’ (Yuan Jiang et al 2003).

In the case of colors, the program would first select a color from the array of samples such as green, and then search the weights for the location containing the greenest color. From there, the colors surrounding that weight are then made greener. Then another color is chosen, such as red, and the process continues.
SOM Algorithm: Proposed System

1. The weight vectors (w) of map nodes are randomized
2. An input vector (v(t)) is grabbed
3. Each node in the map is traversed
4. Euclidean distance formula is used to find the similarity between the input vector and the map’s node’s weight vector.
5. The node that produces the smallest distance (this node is the best matching unit, BMU) has been tracked
6. The nodes in the neighbourhood of BMU are updated by pulling them closer to the input vector by using equation (3.8)

\[ W_{v(t+1)} = W_{v(t)} + \alpha (D(t) - W_{v(t)}) \]  \hspace{1cm} (3.8)

where, Alpha is monotonically decreasing learning coefficient. It is 1 for neurons close to BMU and zero for others.
D(t) - input vector neighbourhood function shrinks with time. At the beginning, when the neighbourhood is broad, the self organizing takes place on a global scale. When the neighbourhood has shrunk to just a couple of neurons, the weights are converging to local estimates.
7. ‘t’ incremented and the steps are repeated from 2.

3.3.3.3 Implementation of Hierarchical Clustering

**Divisive Hierarchical Clustering:** A hierarchical clustering method works by grouping data objects into a tree of clusters. This follows top down strategy. This algorithm starts with all objects in one cluster (Sabyasachi et al 2008).
It subdivides the cluster into smaller and smaller pieces until each object forms a cluster on its own or until it satisfies certain termination conditions such as desired number of clusters is obtained. Figure 3.6 depicts the Hierarchical Clustering.

![Figure 3.6 Hierarchical Clustering](image)

**Algorithm**

1. The whole image is in one cluster.
2. The most dissimilar point in the image is found and the images divided into two clusters.
3. Step 2 is repeated for each cluster.
4. A tree like structure is formed. Step 2 is repeated until level 4 is reached. Level 4 has 8 clusters.

**3.3.3.4 Implementation of Fuzzy C-Means Clustering**

- The Fuzzy C-Means algorithm (often abbreviated to FCM) is an iterative algorithm that find clusters in data and which uses the
concept of fuzzy membership: instead of assigning a pixel to a single cluster, each pixel will have different membership values on each cluster (Boudraa et al 2000)

- The Fuzzy C-Means attempts to find clusters in the data by minimizing an objective function shown in the equation (3.9) below:

\[
J = \sum_{i=1}^{N} \sum_{j=1}^{C} \mu_{ij}^m |x_i - c_j|^2
\]

(3.9)

where, \( J \) is the objective function. The value of \( J \) is smaller after iteration of the algorithm. It means the algorithm is converging or getting closer to a good separation of pixels into clusters.

\( N \) is the number of pixels in the image.

\( C \) is the number of clusters used in the algorithm, and which must be decided before execution.

\( \mu \) is the membership table. A table of \( N \times C \) entries which contains the membership values of each data point and each cluster.

\( m \) is a fuzziness factor (a value larger than 1).

\( x_i \) is the \( i^{th} \) pixel in \( N \).

\( c_j \) is \( j^{th} \) cluster in \( C \).

| \( x_i - c_j \) | is the Euclidean distance between \( x_i \) and \( c_j \).

**Algorithm**

The input to the algorithm is the \( N \) pixels on the image and \( m \), the fuzziness value. The fuzziness value of two is used. The steps for the algorithm are listed below.
Steps:

1. The value of $\mu$ is initialized with random values between zero and one; but with the sum of all fuzzy membership table elements for a particular pixel should be equal to 1 or the sum of the memberships of a pixel for all clusters must be one.

2. An initial value is calculated for $J$ using equation (3.10)

$$J = \sum_{i=1}^{N} \sum_{j=1}^{C} \mu_{ij}^{m} | x_i - c_j |^{2}$$  \hspace{1cm} (3.10)

3. The centroids of the clusters ($c_j$) are calculated using equation (3.11)

$$C_j = \frac{\sum_{i=1}^{N} \mu_{ij}^{m} x_i}{\sum_{i=1}^{N} \mu_{ij}^{m}}$$  \hspace{1cm} (3.11)

4. The fuzzy membership table is calculated using equation (3.12)

$$\mu_{ik} = \frac{1}{\sum_{k=1}^{C} \left( \frac{x_i - c_j}{|x_j - c_k|} \right)^{m-1}}$$  \hspace{1cm} (3.12)

5. $J$ is recalculated

6. Step 3 is repeated until a stopping condition was reached.

Some possible stopping conditions of the algorithm are as follows:

1. A number of iterations were executed, and it can be considered that the algorithm achieved a "good enough" clustering of the data.
2. The difference between the values of $J$ in consecutive iterations is small (smaller than a user-specified parameter $\varepsilon$), therefore the algorithm has converged.

**Defuzzification**

- At the end of the execution, the membership values for each pixel is obtained in each cluster.
- Traditionally the algorithm can then *defuzzify* its results by choosing a "winning" cluster, i.e. the one which is closer to the pixel in the feature spaced, is the one for which the membership value is highest

3.3.4 **Cluster Selection**

After the clustering process, the cluster containing a Region of interest (tumor) (Badran et al 2010) is selected as the primary segment. Most image-related tasks can process a subset of the pixels in an input image. Depending on the task, the selected pixels may either be a fairly arbitrary region, or only a regular sub image of the input image.

3.3.5 **Histogram Based Clustering**

To eliminate the pixels which are not related to the interest in the selected cluster, histogram clustering is applied by luminosity feature $L*$ and color information $a*$ and $b*$ to derive the final segmented result. The K-means algorithm uses $L*$ for histogram clustering where as SOM uses $a*$ and $b*$. The histogram clustering in hierarchical segmentation uses $l*$ to achieve the final segmentation result. In RGB color space, histogram clustering uses red value to derive the final segmentation result.
3.3.6 Region Elimination

The output of histogram clustering consists of tumor region as well as the other regions which has the same luminance and color values as the tumor. The regions which are smaller than the tumor are eliminated using region growing algorithm.

3.4 EXPERIMENTAL RESULTS

In this section, the results of the proposed approach are explained. In this thesis, MATLAB (R2010a) is used as the computation environment. MATLAB is chosen as it enables to perform computationally intensive tasks faster.

Dataset: The proposed system is tested on a database of 100 MRI images. The data set is collected from the web database http://brainweb.bic.mni.mcgill.ca/brainweb/selection_ms.html.

3.4.1 K-means Clustering Algorithm

This technique is to track tumor objects in Magnetic Resonance (MR) (Ming-Ni Wu et al. 2007) brain images. The key concept in the color-based segmentation algorithm with K-means is to convert a given gray-level MR image into a color space image and then separate the position of tumor objects from other items of an MR image by using K-means clustering.

Figure 3.7 shows the original MRI brain tumor image. Figure 3.8 shows the image after Pseudo color translation. Figure 3.9 shows the image after K-means Clustering in L*a*b* color space. Figure 3.10 shows the image after cluster selection. Figure 3.11 shows the image after region elimination. Figure 3.12 shows the segmented tumor image.
Figure 3.7 Input Image

Figure 3.8 Image after Pseudo color translation

Figure 3.9 Image from K-means Clustering

Figure 3.10 Cluster Selection
3.4.2 Self Organizing Map

The input to Self Organizing Map (SOM) is MRI image with Brain tumor is shown in Figure 3.7. The resultant image after Pseudo Color Translation is shown in Figure 3.8. Figure 3.13 shows the image after Clustering using Self Organising Map neural network (Murugavalli and Rajamani 2007). The clustering is in L*a*b* color space. The number of iterations is chosen manually as 250 by Brute Force Technique. Figure 3.14 shows the segmented tumor image after cluster selection and after region elimination.

3.4.3 Hierarchical Clustering

The input is a MRI brain tumor image after Pseudo Color Translation as shown in Figure 3.8. Figure 3.15 shows the level1 image after Hierarchical Clustering which has only one cluster. Figure 3.16 shows the level 2 image which has two clusters. Figure 3.17 shows the level3 image
which has four clusters. Figure 3.18 shows the level 4 image which has eight clusters and the clustering is in RGB color space. Figure 3.19 shows the image after cluster selection. Figure 3.20 shows the segmented tumor image after region elimination.
3.4.4 Fuzzy C-means Clustering

The input is a MRI brain tumor image after Pseudo Color Translation as shown in the Figure 3.8. Figure 3.21 shows the tumor detected image after clustering using Fuzzy C-means Clustering. The Clustering is in RGB color space. (Kobashi et al. 2006, Hezam Izakian and Ajith Abraham 2011).
3.5 PERFORMANCE ANALYSIS

The Sensitivity (Se), Specificity (Sp) and Accuracy (acc) for the clustering algorithms are evaluated as shown in the equations (3.13) – (3.15)

\[
Se = \frac{TP}{TP + FP} 
\]  

(3.13)
\[ Sp = \frac{TP}{TN + FP} \]  
\[ Acc = \frac{TP + TN}{TP + FN + TN + FP} \]

where, TP-True Positive, FN-False Negative, TN-True Negative, FP-False Positive. Table 3.1 and Table 3.2 show the performance analysis of the four clustering techniques implemented. The following graphs on Figures 3.22 and 3.23 shows the pictorial representation of performance of the four clustering techniques based on the Sensitivity, Specificity and Prediction Accuracy as per the formulae given.

Table 3.1 Clustering Techniques Performance in RGB Color Space

<table>
<thead>
<tr>
<th>Type of Clustering</th>
<th>Sensitivity (%)</th>
<th>Specificity (%)</th>
<th>Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>K-means Clustering</td>
<td>85.5</td>
<td>91.3</td>
<td>93.2</td>
</tr>
<tr>
<td>Self Organizing Maps</td>
<td>72.7</td>
<td>74.0</td>
<td>82.8</td>
</tr>
<tr>
<td>Divisive Hierarchical Clustering</td>
<td>71.9</td>
<td>81.2</td>
<td>92.4</td>
</tr>
<tr>
<td>Fuzzy C-Means Clustering</td>
<td>75.9</td>
<td>79.9</td>
<td>80.7</td>
</tr>
</tbody>
</table>

Table 3.2 Clustering Techniques Performance in L*a*b* Color Space

<table>
<thead>
<tr>
<th>Type of Clustering</th>
<th>Sensitivity (%)</th>
<th>Specificity (%)</th>
<th>Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>K-means Clustering</td>
<td>85.2</td>
<td>91</td>
<td>91.9</td>
</tr>
<tr>
<td>Self Organizing Maps</td>
<td>77.8</td>
<td>79.1</td>
<td>80.2</td>
</tr>
<tr>
<td>Divisive Hierarchical Clustering</td>
<td>82.3</td>
<td>89.1</td>
<td>90.7</td>
</tr>
</tbody>
</table>

The four clustering algorithms were tested with a set of MRI brain images. K-means clustering achieved 93.2% in RGB Color Space and 91.9% in L*a*b* Color Space.
Figure 3.22  Clustering Techniques Performance Comparison in RGB Color Space

Figure 3.23  Clustering Techniques Performance Comparison in L*a*b* Color Space
3.6 DISCUSSION

In this system, the popular Clustering Algorithms have been selected to predict tumors from the given MR images. After the Clustering process, the cluster containing the tumor is selected as the primary segment. To eliminate the pixels which are not related to the interest in the selected cluster, histogram clustering is applied by luminosity feature L* and color information a* and b* to derive the final segmented result. The K-means algorithm uses L* for histogram clustering whereas SOM uses a* and b*. The histogram clustering in hierarchical segmentation uses L* to achieve the final segmentation result. In RGB color space, histogram clustering uses red value to derive the final segmentation result. The performance of the above four clustering algorithms are measured based on Sensitivity, Specificity and Accuracy. Regarding the Prediction accuracy, K-means clustering and Hierarchical clustering gave better results than other methods. The four clustering algorithms were tested with a set of MRI brain images. K-means clustering and hierarchical clustering achieved 93.2% and 92.4% whereas SOM and FCM achieved a result of about 80% Prediction accuracy.