Chapter 4

GLCM TEXTURAL FEATURES FOR BRAIN TUMOR CLASSIFICATION
4.1 INTRODUCTION

There are many techniques for the discriminating the diverse kinds of tumors from the given MR scan such as, Research author Matthew C. Clarke et.al. (56) Designed a technique for disease affected MR volume determination with segmenting MR scan using FCM procedure.

Another researcher’s chang et.al. (57, 58) described the SVM is a finest procedure in sonography for the finding of breast tumor. The researcher W. Chu et.al. (59) Presented that LS-SVM usually able to distribute greater discrimination accurateness than the other remaining data discrimination procedures. In medicinal practice, image processing, the identification of tissue form (normal or disease affected) and discrimination of tissue pathology are accomplished by using texture. MR scans texture showed to be beneficial to identify the tumor type (60).

For the textural characteristics, the researchers Haralick et.al. (61) Recommended a collection of 14 texture features which can be retrieved from the co-occurrence matrix, and which hold facts about image texture characteristics such as linearity, contrast and homogeneity. In our research work, we utilized the GLCM textural characteristics for tumor discrimination using the FFNN.

Proposed Method

![Diagram](image)

Figure 4.1: Proposed Method of Brain Tumor Classification using GLCM
4.2 METHODS AND MATERIALS

In our research paper entitled “GLCM Textural features for the Brain Tumor Classification” we used four different classes of brain tumors for the experimentation.

Class I (Astrocytoma)

The MR scans were gathered from the tolerant having the 35 age. Magnetic resonance validates a region of diversified signal intensity on PD and T2-weighted MR scans, in a leftward occipital area. Contrast improvement displays the affected part to comprise cystic components. Thallium imaginings display a frontal edge of tall uptake, steady with a lesser area of cancer reappearance.

Class II (Meningioma)

The 75 year old patient who had ten months history of tolerant struggle for walking. He observed certain left lower extreme feebleness and certain trouble with reminiscence and attentiveness. He was watchful and focused, but had deliberate and undetermined dialogue. He might recollect only 1 of 3 objects at 5 minutes.

Class III (Metastatic bronchogenic carcinoma)

The 42 year old lady with a lengthy past of tobacco usage originated having annoyances one month previously these imageries were got. Brain imageries display a big mass with neighboring edema, and density of near by midbrain constructions. The magnetic resonance proves the cancer as a region of high signal intensity on PD and T2-weighted scans in a big leftward temporal area.

Class IV (Sarcoma)

The person with the age of 22 year who was in the hospital for resection of Ewing's sarcoma. Unclearly defined pictorial trouble was noted with hindsight to have instigated roughly one month previous to admitted to the hospital.

Dataset

Four groups of cancer affected MR scans are utilized for the tentative work in which every class comprises 20 scans; entire 80 scans are retrieved from WBA. Each MR scan has the precise dimension 256x256.
Class I (Astrocytoma)

Class II (Meningioma)

Class III (Metastatic bronchogenic carcinoma)

Class IV (Sarcoma)

Figure 4.2: Dataset of four different tumor classes.

Preprocessing

Medicinal image analysis needs the initial processing since the disturbances may be included to the MR scan due to imaging modalities. We applied a Gaussian filter to enhance the excellence of the Image by Noise destruction, contract improvement, intensity equalization, outlier removal.

The 1D form of the Gaussian is:

\[ G(x) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{x^2}{2\sigma^2}} \] (4.1)
In the above expression $\sigma$ represents standard deviance of the distribution. Also we consider, this takes a mean of 0. In two dimensional, rounded symmetric Gaussian can be expressed as:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (4.2)$$

### 4.3 GLCM TEXTURAL FEATURE EXTRACTION

GLCM is the statistic technique of inspecting the textural features that reflects the dimensional association of the pixels. The GLCM function illustrate the textural feature of an picture by computing how frequently couple of pixels with exact values and in a definite spatial relationship happen in an image, producing a GLCM, and then retrieving statistics characteristics from this matrix. The function in MATLAB namely gray-comatrix, generates a GLCM by computing how often a pixel with the intensity value $i$ arises in a precise spatial relationship with pixel having the $j$ value. By default, the spatial relationship is well-defined as the pixel of attention and the pixel to its instant right (straight next to), but you can stipulate other spatial relationship amongst 2 pixels. Every element $(i,j)$ in the resulting GLCM is just the summation of the number of times that the pixel with $i$ value arisen in the definite spatial relationship to a pixel having value $j$ in the provided image.

In the matrix GLCM, total row and column are equivalent to the complete gray level, $G$, in the given MR scan. The component of the matrix $P(i, j | \Delta x, \Delta y)$ is the comparative frequency disconnected by a distance $(\Delta x, \Delta y)$. Component of the matrix also characterized as $P(i, j | d, \theta)$ which covers the 2nd order probability measures for deviations amongst gray level $i$ and $j$ at space $d$ and specific angle $\theta$. Different statistical measures are retrieved using GLCM. $G$ is the total gray levels applied and $\mu$ is the mean of $P$. $\mu_x, \mu_y, \sigma_x$ and $\sigma_y$ can be treated as means and standard deviations of $P_x$ and $P_y$. $P_x(i)$ can be considered as the $i_{th}$ entry got by summation of the rows of $P(i,j)$:

$$p_x(i) = \sum_{j=0}^{G-1} P(i, j) \quad \text{and} \quad p_y(j) = \sum_{i=0}^{G-1} P(i, j) \quad (4.3)$$
\( \mu_x = \sum_{i=0}^{G-1} i P_x(i) \) and \( \mu_y = \sum_{j=0}^{G-1} j P_y(j) \)  

(4.4)

\[ \sigma_x^2 = \sum_{i=0}^{G-1} (P_x(i) - \mu_x(i))^2 \] and \[ \sigma_y^2 = \sum_{j=0}^{G-1} (P_y(j) - \mu_y(j))^2 \]  

(4.5)

Homogeneity (ASM)

\[ ASM = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} \{ P(i, j) \}^2 \]  

(4.6)

Contrast

\[ \text{Contrast} = \sum_{n=0}^{G-1} n^2 \{ \sum_{i=1}^{G} \sum_{j=1}^{G} P(i, j) \}, |i - j| = n \]  

(4.7)

IDM

\[ IDM = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} \frac{1}{1 + (i - j)^2} P(i, j) \]  

(4.8)

Entropy

\[ \text{Entropy} = -\sum_{i=0}^{G-1} \sum_{j=0}^{G-1} P(i, j) \times \log(P(i, j)) \]  

(4.9)

Correlation

\[ \text{Correlation} = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} \frac{i \times j \times P(i, j) - \{ \mu_x \times \mu_y \}}{\sigma_x \times \sigma_y} \]  

(4.10)
Variance

\[
\text{Variance} = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} (i - \mu)^2 P(i, j)
\]  
(4.11)

Aver

\[
\text{Aver} = \sum_{i=0}^{2G-2} i P_{x+y}(i)
\]  
(4.12)

Sum Entropy

\[
\text{Sent} = -\sum_{i=0}^{2G-2} P_{x+y}(i) \log(P_{x+y}(i))
\]  
(4.13)

Difference Entropy

\[
\text{Dent} = -\sum_{i=0}^{G-1} P_{x+y}(i) \log(P_{x+y}(i))
\]  
(4.14)

Inertia

\[
\text{Intertia} = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} (i - j)^2 \times P(i, j)
\]  
(4.15)

Shade

\[
\text{Shade} = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} \{i + j - \mu_x - \mu_y\}^3 \times P(i, j)
\]  
(4.16)

Prominence

\[
\text{prom} = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} \{i + j - \mu_x - \mu_y\}^4 \times P(i, j)
\]  
(4.17)
TABLE 04.1: An Sample of GLCM feature set of a slice 10.

<table>
<thead>
<tr>
<th>Class/Features</th>
<th>Group I (Scan 10)</th>
<th>Group II (Scan 10)</th>
<th>Group III (Scan 10)</th>
<th>Group IV (Scan 10)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Autocorrelation</td>
<td>0.569</td>
<td>2.089</td>
<td>2.189</td>
<td>2.001</td>
</tr>
<tr>
<td>Contrast</td>
<td>1.732</td>
<td>0.241</td>
<td>0.272</td>
<td>0.176</td>
</tr>
<tr>
<td>Correlation</td>
<td>0.62</td>
<td>0.569</td>
<td>0.526</td>
<td>0.658</td>
</tr>
<tr>
<td>Cluster Prominence</td>
<td>0.375</td>
<td>1.915</td>
<td>1.655</td>
<td>1.742</td>
</tr>
<tr>
<td>Cluster Shade</td>
<td>1.256</td>
<td>0.65</td>
<td>0.489</td>
<td>0.675</td>
</tr>
<tr>
<td>Dissimilarity</td>
<td>0.892</td>
<td>0.229</td>
<td>0.256</td>
<td>0.167</td>
</tr>
<tr>
<td>Energy</td>
<td>0.892</td>
<td>0.360</td>
<td>0.326</td>
<td>0.416</td>
</tr>
<tr>
<td>Entropy</td>
<td>0.555</td>
<td>1.30</td>
<td>1.36</td>
<td>1.161</td>
</tr>
<tr>
<td>Homogeneity</td>
<td>2.098</td>
<td>0.887</td>
<td>0.874</td>
<td>0.917</td>
</tr>
<tr>
<td>Maximum probability</td>
<td>3.574</td>
<td>0.534</td>
<td>0.482</td>
<td>0.588</td>
</tr>
<tr>
<td>Sum square: Variance</td>
<td>1.083</td>
<td>2.167</td>
<td>2.283</td>
<td>2.047</td>
</tr>
<tr>
<td>Sum average</td>
<td>0.228</td>
<td>2.778</td>
<td>2.855</td>
<td>2.706</td>
</tr>
<tr>
<td>Sum variance</td>
<td>0.542</td>
<td>3.637</td>
<td>3.770</td>
<td>3.677</td>
</tr>
<tr>
<td>Sum entropy</td>
<td>0.996</td>
<td>1.117</td>
<td>1.154</td>
<td>1.026</td>
</tr>
<tr>
<td>Difference variance</td>
<td>0.54</td>
<td>0.24</td>
<td>0.272</td>
<td>0.176</td>
</tr>
<tr>
<td>Difference entropy</td>
<td>0.975</td>
<td>0.559</td>
<td>0.59</td>
<td>0.464</td>
</tr>
</tbody>
</table>

### 4.4 CLASSIFICATION USING ANN

For the classification, purpose we used the two layers FFNN in which learning assumes the availability of a labeled set of trained data that is computed of N input and actual output

\[ T = \{(X_i, d_i)\}_{i=1}^N \]  \hspace{1cm} (4.18)

In the expression given,

- \( X_i \) is input vector for the \( i_{th} \) sample
- \( d_i \) is the desired output for the \( i_{th} \) sample
- \( N \) is the sample size.

A dual-layer FFNN with activation function sigmoid is developed with following specification.
Input neurons 44.

Hidden neurons 10.

Output neurons 4.

**Network architecture**

![Diagram of a two layer feed forward network](image)

**Figure 4.3**: A two layer feed forward Network.

The LM training algorithm (62) is utilized for the training the NN which is a very simple, but strong, technique for approaching a function. Fundamentally, it contains in resolving the expression:

\[(J^T J + \lambda I)\delta = J^T E\]  \(4.19\)

In the above expression, \(J\) is the Jacobian matrix, \(\lambda\) is the damping factor, \(\delta\) is the weight upgrade vector that we wish to discover and \(E\) is the error vector consisting the resultant errors for each input vector utilized to train the network. The \(\delta\) say us by how much we can upgrade the network weights to get better resolution.

The matrix \(J^T J\) also known as Hessian. The \(\lambda\) damping factor is updated at every repetition, and suggest the optimization procedure. If decrease of \(E\) is fast, a lesser measure can be utilized, fetching the procedure nearer to the Gauss–Newton procedure, while if repetition gives inadequate decrease in the residual, \(\lambda\) can be enlarged, giving a step nearer to the gradient descent way. For the upgrading NN, we utilized the 56 scans, 16 scans for validation and 8 scans for the testing. The training stops when a discriminator provides a high accurateness measure with less training and testing errors.
4.5 EXPERIMENTAL RESULTS

Near about 44 GLCM statistical measures of each slice is calculated. 80 samples of four classes (Class I, Class II, Class III, and Class IV) which form the input vector of size 80x44. Target output is designed with the size of 80x4. The LM algorithm outperformed in this practical approach by discriminating the inputs in 15 periods with the typical time for the training is 20 seconds. The performance measured and outcome of the network is given below:

<table>
<thead>
<tr>
<th>Class</th>
<th>No of Classified Images</th>
<th>No of Miss-classified Images</th>
<th>Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class-I</td>
<td>20</td>
<td>Nil</td>
<td>100%</td>
</tr>
<tr>
<td>Class-II</td>
<td>20</td>
<td>Nil</td>
<td>100%</td>
</tr>
<tr>
<td>Class-III</td>
<td>19</td>
<td>01</td>
<td>95%</td>
</tr>
<tr>
<td>Class-IV</td>
<td>19</td>
<td>01</td>
<td>95%</td>
</tr>
</tbody>
</table>

TABLE 4.2: Accuracy of LM training algorithm

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of periods</td>
<td>15</td>
</tr>
<tr>
<td>Performance</td>
<td>0.0400</td>
</tr>
<tr>
<td>Training Performance</td>
<td>0.0306</td>
</tr>
<tr>
<td>Validation Performance</td>
<td>0.0716</td>
</tr>
<tr>
<td>Testing Performance</td>
<td>0.0424</td>
</tr>
<tr>
<td>Classification Rate</td>
<td>97.5%</td>
</tr>
</tbody>
</table>

TABLE 4.3: Performance Measure of LM training algorithm

The error measurement like MSE and PE are documented. MSE is the mean of the square off error amongst the anticipated result and the real outcome of the FFNN.
MSE is totaled using expression such as

\[
MSE = \frac{\sum_{j=0}^{P} \sum_{i=0}^{N} (d_{ij} - y_{ij})^2}{NP}
\]  

(4.20)

Where

- \( P \) = Total output processing elements
- \( N \) = Total patterns in the given data set

\( y_{ij} \) = projected network emissions output for pattern \( i \) at processing element \( j \)

\( d_{ij} \) = real output for emissions exemplar \( i \) at processing element \( j \).

Percent Error specifies the portion of examples, which are not classified. Value 0 means no misclassifications.

\[
\%error = \frac{100 \sum_{j=0}^{P} \sum_{i=0}^{N} |dy_{ij} - dd_{ij}|}{NP \sum_{j=0}^{P} \sum_{i=0}^{N} dd_{ij}}
\]  

(4.21)

Where

- \( P \) = Total output processing elements
- \( N \) = Total patterns in the training data set
\[ dy_{ij} = \text{demoralized network emissions output for pattern } i \text{ at working element } j \]

\[ dd_{ij} = \text{demoralized desired network releases result for example } i \text{ at working element } j. \]

**Confusion Matrix (CM)**

The CM provides the accurateness of the classification problem. Total 80 images of 4 various classes are discriminated with 97.5% accuracy. The diagonal numbers of the CM displays discriminated groups. The following CM demonstrates that, Class I and Class II are exactly documented in which all imageries drop in the similar class. But then again for the Class III and Class IV only single scan drop outside the class therefore average error rate is 2.5%.

![Confusion Matrix](image)

**Figure 4.5: Confusion Matrix showing the recognition rate.**

A contour plot is a graphical method for representing a 3D superficial by drawing steady \( z \) slices, described as contours, on a 2D format. That is, provide a measure for \( z \), outlines are put for linking the \((x,y)\) coordinates where that \( z \) measure occurs. Given contour plot indicates the target and output contours.
4.6 CONCLUSION

In this research work, we expected to discriminate the 4 different classes of tumor categories such as Astrocytoma, Meningioma, Metastatic bronchogenic carcinoma, and Sarcoma. All the MRI slices collected from the WBA and after preprocessing the GLCM textural features used to train the feed forward neural network with Levenberg Marquart (LM) nonlinear optimization algorithm which gives the better classification rate of 97.5%. May this work will assist the physician to mark the final decision for the further treatment.