IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 3, No 3, May 2012
ISSN (Online): 1694-0814
www.IJCSI.org

354

GLCM Textural Features for Brain Tumor Classification

Nitish Zulpe[1] and Vrushsen Pawar[2]

**[1]College of Computer Science and
Information Technology, Latur-413512
Maharashtra, India**

**[2]Department of computational science,
SRTM University Nanded,
Maharashtra**

Abstract

Automatic recognition system for medical images is challenging task in the field of medical image processing. Medical images acquired from different modalities such as Computed Tomography (CT), Magnetic Resonance Imaging (MRI), etc which are used for the diagnosis purpose. In the medical field, brain tumor classification is very important phase for the further treatment. Human interpretation of large number of MRI slices (Normal or Abnormal) may leads to misclassification hence there is need of such a automated recognition system, which can classify the type of the brain tumor. In this research work, we used four different classes of brain tumors and extracted the GLCM based textural features of each class, and applied to two-layered Feed forward Neural Network, which gives 97.5% classification rate.

Keywords: MRI, CT, GLCM, Neural Network

1. Introduction

Abnormal growth of cell in the brain causes the brain tumor and may affect any person almost of any age. Brain tumors can have a variety of shapes and sizes; it can appear at any location and in different image intensities [1]. Brain tumor classification is very significant phase in the medical field. The images acquired from different modalities such as CT, MR that should be verified by the physician for the further treatment, but the manual classification of the MR images is the challenging and time consuming task [2]. Human observations may lead to misclassification and hence there is need of automatic or semiautomatic classification techniques to make the difference between different tumor types.

We found many classification techniques have been given for the determining the tumor type from the given MR images such as, Matthew C. Clarke et al. [3] developed a method for abnormal MRI volume identification with slice segmentation using Fuzzy C-means (FCM) algorithm. Chang et al. [4,5] reported the SVM is an best tool in sonography for the diagnosis of

breast cancer. W. Chu *et. al.* [6] proposed that LS-SVM generally are able to deliver higher classification accuracy than the other existing data classification algorithms.

In medical image analysis, the determination of tissue type (normal or abnormal) and classification of tissue pathology are performed by using texture. MR image texture proved to be useful to determine the tumor type [7]. Haralick *et. al.* [8] suggested a set of 14 textural features which can be extracted from the co-occurrence matrix, and which contain information about image textural characteristics such as homogeneity, linearity, and contrast. In this research paper, we used the GLCM textural features for tumor classification using the feed forward neural network.

2. Methods and Materials

2.1 Magnetic Resonance Imaging

A magnetic resonance imaging instrument (MRI Scanner) uses powerful magnets to polarize and excite hydrogen nuclei i.e. proton in water molecules in human tissue, producing a detectable signal which is spatially encoded, resulting in images of the body. MRI uses three electromagnetic fields

1) A very strong static magnetic field to polarize the hydrogen nuclei, called the static field.

2) A weaker time varying field(s) for spatial encoding called the gradient field.

3) A weak radio frequency field for manipulation of hydrogen nuclei to produce measurable signals Collected through RF antenna.

Class I (Astrocytoma)

The patient was a 35-year-old man; MR demonstrates an area of mixed signal intensity on proton density (PD) and T2-weighted (T2) images in a left occipital region. Contrast enhancement shows the lesion to contain cystic elements. Thallium images show an anterior border of

high uptake, consistent with a small region of tumor recurrence.

Class II (Meningioma)

The patient was a 75-year-old man who had an 8 - 10 month history of progressive difficulty walking. He had noted some left lower extremity weakness and some difficulty with memory and concentration. He was alert and oriented, but had slow and hesitating speech. He could recall only 1 of 3 objects at five minutes.

Class III (Metastatic bronchogenic carcinoma)

This 42 year old woman with a long history of tobacco use began having headaches one month before these images were obtained. Brain images show a large mass with surrounding edema, and compression of adjacent midbrain structures. The MR demonstrates the tumor as an area of high signal intensity on proton density (PD) and T2-weighted (T2) images in a large left temporal region.

Class IV (Sarcoma)

The patient was a 22 year old man who was admitted for resection of Ewing's sarcoma (peripheral/primitive neuroepithelial tumor- PNET). Vaguely described visual difficulty was noted retrospectively to have begun approximately one month prior to admission.
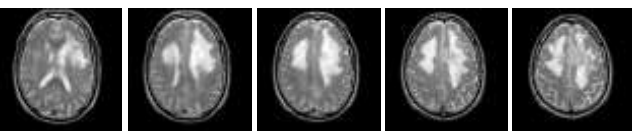
2.2 Dataset

Four different classes of Brain tumor MR images are used for this experimental work in which each class contains 20 samples; total 80 samples are collected from the Whole Brain Atlas (WBA). Every image is having the exact size of 256x256 in axial view.
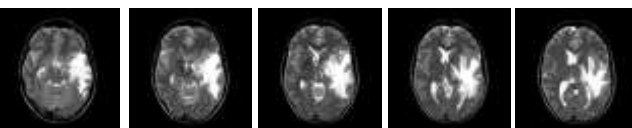
Class I (Astrocytoma)

Class II (Meningioma)

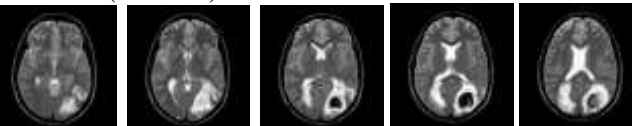Class III (Metastatic bronchogenic carcinoma)

Class IV (Sarcoma)

Fig. 1 Five sample MR images of four classes

2.3 Preprocessing

Medical image analysis requires the preprocessing, because of the noise may be added to the MR images due to imaging devices. We used the Gaussian filter to improve the quality of the Image by Noise suppression, contract enhancement, intensity equalization, outlier elimination. The Gaussian distribution in 1-D has the form:

$$G(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}} \tag{1}$$

Where σ is the standard deviation of the distribution. We have also assumed that the distribution has a mean of zero. In 2-D, an isotropic (*i.e.* circularly symmetric) Gaussian has the form:

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

2.4 Features extraction

Gray-level co-occurrence matrix (GLCM) is the statistical method of examining the textures that considers the spatial relationship of the pixels. The GLCM functions characterize the texture of an image by calculating how often pairs of pixel with specific values and in a specified spatial relationship occur in an image, creating a GLCM, and then extracting statistical measures from this matrix. The graycomatrix function in MATLAB creates a gray-level co-occurrence matrix (GLCM) by calculating how often a pixel with the intensity (gray-level) value i occurs in a specific spatial relationship to a pixel with the value j. By default, the spatial relationship is defined as the pixel of interest and the pixel to its immediate right (horizontally adjacent), but you can specify other spatial relationships between the two pixels. Each element (i, j) in the resultant GLCM is simply the sum of the number of times that the pixel with value i occurred in the specified spatial relationship to a pixel with value j in the input image.

A GLCM is a matrix where the number of rows and columns is equal to the number of gray levels, G, in the image. The matrix element $P(i, j \,|\, \Delta x, \Delta y)$ is the relative frequency separated by a pixel distance $(\Delta x, \Delta y)$. Matrix element also represented as $P(i, j \,|\, d, \theta)$ which contains the second order probability values for changes between gray level I and j at distance d a particular angle θ. Various features are extracted from GLCM, G is the number of gray levels used and μ is the mean value of P μ_x, μ_y, σ_x and σ_y are the means and standard deviations of P_x and P_y. $P_x(i)$ is the i_{th} entry obtained by summing the rows of $P(i,j)$:

$$p_x(i) = \sum_{j=0}^{G-1} P(i, j) \text{ and } P_y(j) = \sum_{i=0}^{G-1} P(i, j) \tag{3}$$

$$\mu_x = \sum_{i=0}^{G-1} i P_x(i) \text{ and } \mu_y = \sum_{j=0}^{G-1} j P_y(j) \qquad (4)$$

$$\left.\begin{array}{l} \sigma_x^2 = \sum_{i=0}^{G-1}(P_x(i) - \mu_x(i))^2 \\[6pt] \sigma_y^2 = \sum_{j=0}^{G-1}(P_y(j) - \mu_y(j))^2 \end{array}\right\} \qquad (5)$$

By using following equations, we can calculate the different textural features that can be used to train the classifier.

Homogeneity (Angular Second Moment)

$$ASM = \sum_{i=0}^{G-1}\sum_{j=0}^{G-1}\{P(i,j)\}^2 \qquad (6)$$

Contrast

$$Contrast = \sum_{n=0}^{G-1} n^2 \{\sum_{i=1}^{G}\sum_{j=1}^{G} P(i,j)\}, |i-j| = n \qquad (7)$$

Inverse Difference Moment

$$IDM = \sum_{i=0}^{G-1}\sum_{j=0}^{G-1}\frac{1}{1+(i-j)^2}P(i,j) \qquad (8)$$

Entropy

$$Entropy = -\sum_{i=0}^{G-1}\sum_{j=0}^{G-1} P(i,j) \times \log(P(i,j)) \qquad (9)$$

Correlation

$$Correlation = \sum_{i=0}^{G-1}\sum_{j=0}^{G-1}\frac{\{i \times j\} \times P(i,j) - \{\mu_x \times \mu_y\}}{\sigma_x \times \sigma_y} \qquad (10)$$

Variance

$$Variance = \sum_{i=0}^{G-1}\sum_{j=0}^{G-1}(i-\mu)^2 P(i,j) \qquad (11)$$

Sum Average

$$Aver = \sum_{i=0}^{2G-2} i P_{x+y}(i) \qquad (12)$$

Sum Entropy

$$Sent = -\sum_{i=0}^{2G-2} P_{x+y}(i) \log(P_{x+y}(i)) \qquad (13)$$

Difference Entropy

$$Dent = -\sum_{i=0}^{G-1} P_{x+y}(i) \log(P_{x+y}(i)) \qquad (14)$$

Inertia

$$Intertia = \sum_{i=0}^{G-1}\sum_{j=0}^{G-1}(i-j)^2 \times P(i,j) \qquad (15)$$

Cluster Shade

$$Shade = \sum_{i=0}^{G-1}\sum_{j=0}^{G-1}\{i+j-\mu_x-\mu_y\}^3 \times P(i,j) \qquad (16)$$

Cluster Prominence

$$prom = \sum_{i=0}^{G-1}\sum_{j=0}^{G-1}\{i+j-\mu_x-\mu_y\}^4 \times P(i,j) \qquad (17)$$

Following table gives an example of some of the features that are extracted from all the classes.

Table 1: Features extracted from Slice No.10 of each class

Class/Features	Class I	Class II	Class III	Class IV
Autocorrelation	0.56958	2.089459	2.189791	2.00143
Contrast	1.732217	0.241603	0.272438	0.176089
Correlation	0.624228	0.5699	0.526011	0.658647
Cluster Prom.	0.375112	1.915923	1.65596	1.742295
Cluster Shade	1.256381	0.65392	0.489271	0.675075
Dissimilarity	0.892752	0.229177	0.256782	0.167784
Energy	0.892037	0.360822	0.326417	0.416605
Entropy	0.55052	1.301338	1.362772	1.1614
Homogeneity	2.098217	0.887483	0.874218	0.917492
Max. Prob.	3.574402	0.53491	0.482899	0.588983
Sum of S.V.	1.083287	2.167098	2.283165	2.047427
Sum average	0.228808	2.778051	2.855607	2.706693
Sum variance	0.542568	3.637988	3.77016	3.677384
Sum entropy	0.996494	1.117896	1.154711	1.026883
Diff. variance	0.54334	0.241603	0.272438	0.176089
Diff. entropy	0.975889	0.559004	0.595968	0.464998

3. Classification

For the classification, purpose we used the two layers feed forward neural network in which learning assumes the availability of a labeled (i.e. ground-truthed) set of training data made up of N input and output

$$T = \{(X_i, d_i)\}_{i=1}^{N} \qquad (18)$$

Where
X_i is input vector for the i_{th} example
d_i is the desired output for the i_{th} example
N is the sample size.

A two layer feed forward network with sigmoid activation function is designed with 44 input neurons, 10 hidden neurons and 4 output neurons for the classification.

IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 3, No 3, May 2012
ISSN (Online): 1694-0814
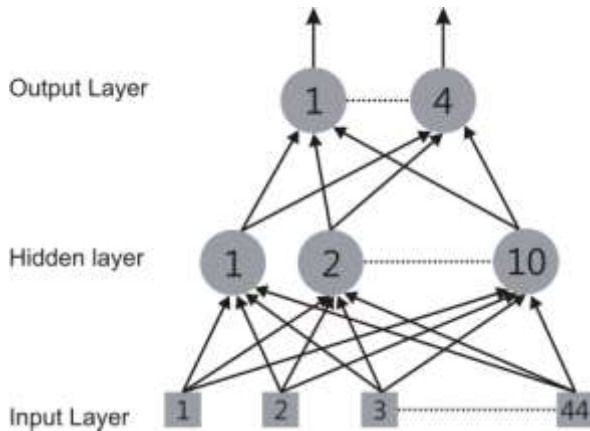www.IJCSI.org

357

Network architecture

Fig. 2 Network Architecture of Two-Layer FF Network

The Levenberg-Marquardt algorithm [9] is used for the training the neural network which is a very simple, but robust, method for approximating a function. Basically, it consists in solving the equation:

$$(J^t J + \lambda I)\delta = J^t E \qquad (19)$$

Where J is the Jacobian matrix for the system, λ is the Levenberg's damping factor, δ is the weight update vector that we want to find and E is the error vector containing the output errors for each input vector used on training the network. The δ tell us by how much we should change our network weights to achieve a (possibly) better solution.

The $J^t J$ matrix can also be known as the approximated Hessian. The λ damping factor is adjusted at each iteration, and guides the optimization process. If reduction of E is rapid, a smaller value can be used, bringing the algorithm closer to the Gauss–Newton algorithm, whereas if iteration gives insufficient reduction in the residual, λ can be increased, giving a step closer to the gradient descent direction. For the training purpose we used the 56 samples, 16 samples for validation and 8 samples for the testing. The training stops when a classifier gives a higher accuracy value with minimum training and testing errors.

4. Experimental Result

Near about 44 GLCM textural features 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 training algorithm outperformed in this experiment by classifying the input data in 15 epochs with the average training time of 20 seconds. The performance measured and outcome of the network are as follows:

Table 2: Performance Measure of LM training algorithm

Classes	No of classified Images	No of Miss-classified Images	Accuracy (%)
Class-I	20	Nil	100%
Class-II	20	Nil	100%
Class-III	19	01	95%
Class-IV	19	01	95%

Table 3: Performance Measure of LM training algorithm

Number of epochs	15
Performance	0.0400
Training Performance	0.0306
Validation Performance	0.0716
Testing Performance	0.0424
classification Rate	97.5%

The error measures like Mean Squared Error (MSE) and Percent Error (PE) are recorded. MSE is the mean of the squared error between the desired output and the actual output of the neural network.
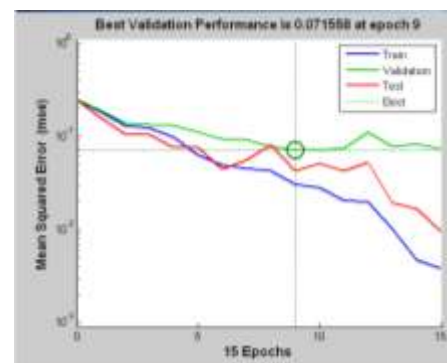

Fig. 3 Performance Graph of the classifier

MSE is calculated using following equation

$$MSE = \frac{\sum_{j=0}^{P}\sum_{i=0}^{N}(d_{ij} - y_{ij})^2}{NP} \qquad (20)$$

Where, P = number of output processing elements

N = number of exemplars in the training data set

y_{ij} = estimated network emissions output for pattern i at processing element j

d_{ij} = actual output for emissions exemplar i at processing element j.

Percent Error indicates the fraction of samples, which are misclassified. Value o means no misclassifications.

IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 3, No 3, May 2012
ISSN (Online): 1694-0814
www.IJCSI.org

358

$$\%\,error = \frac{100}{NP}\sum_{j=0}^{P}\sum_{i=0}^{N}\frac{|\,dy_{ij} - dd_{ij}\,|}{dd_{ij}} \tag{21}$$

Where

P = number of output processing elements

N = number of patterns in the training data set

dy_{ij} = demoralized network emissions output for pattern i at processing element j

dd_{ij} = demoralized desired network emissions output for exemplar i at processing element j.

4.1 Confusion Matrix

The confusion matrix gives the accuracy of the classification problem. 80 MR images of 4 different classes are classified with 97.5% accuracy with Levenberg-Marquardt algorithm. The diagonal elements of the confusion matrix shows classified groups. The following confusion matrix illustrates that, Class I and Class II are exactly recognized in which all images fall in the same class. But for the Class III and Class IV only one image fall outside the class therefore average error rate is 2.5%.

Fig. 4 Confusion Matrix showing the recognition rate

4.2 Contour Plot

A contour plot is a graphical technique for representing a 3-dimensional surface by plotting constant z slices, called contours, on a 2-dimensional format. That is, given a value for z, lines are drawn for connecting the (x, y) coordinates where that z value occurs. Target contour plot represents desired output vector and output contour plot represents actual output. Following contour plot shows the target and output contours.
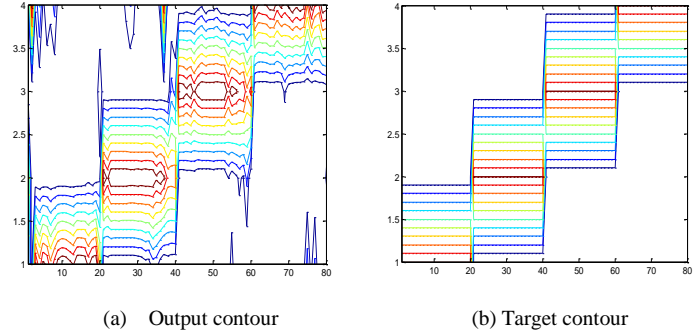

(a) Output contour   (b) Target contour

Fig. 5 Contour plot of Target and actual Output

5. Conclusion

In this research paper, we tried to classify the four different classes of tumor types 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 recognition rate of 97.5%. May this work will assist the physician to make the final decision for the further treatment.

References

[1] Ricci P.E., Dungan D. H., "Imaging of low- and intermediate-grade gliomas", Seminars in Radiation Oncology, 2001, 11(2), p. 103-112.

[2] Kaus *et.al.,* " Automated segmentation of MR images of brain tumors", Journal of radiology, vol.218, no.2, pp: 586-591, 2001.

[3] M. C. Clark, L. O. Hall, D. B. Goldgof, L.P. Clarke, R. P. Velthuizen, and M. S.Silbiger, "MRI Segmentation using Fuzzy Clustering Techniques", IEEE Engineering in Medicine and Biology, pp. 730-742, 1994.

[4] Chang R.F., Wu W.J., Moon W.K., Chou Y.H., Chen D.R., "Support vector machines for diagnosis of breast tumors on US images", Academic Radiology, 2003, 10(2), p. 189-197.

[5] Chang R.F., Wu W.J., Moon W.K., Chou Y.H., Chen D.R., "Improvement in breast tumor discrimination by support vector machines and speckle-emphasis texture analysis", Ultrasound in Medicine and Biology, 2003, 29(5), p. 679-686.

[6] W. Chu, C. J. Ong, and S. S. Keerthi "An improved conjugate gradient scheme to the solution of least squares SVM", IEEE Transactions on Neural Networks 16(2): pp. 498-501, 2005.

IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 3, No 3, May 2012
ISSN (Online): 1694-0814
www.IJCSI.org

359

[7] Schad L.R., Bluml S., "MR tissue characterization of intracranial tumors by means of texture analysis", Magnetic Resonance Imaging, 1993, 11(6), p. 889-896.

[8] Haralick R. M., Shanmugam K., Dinstein I., "Textural Features for Image Classification", IEEE Trans. on Systems Man and Cybernetics, 1973, 3(6), p. 610-621.

[9] Martin T. Hagan and Mohammad B. Menhaj, "Training Feed forward Networks with the Marquardt Algorithm", IEEE Trans. on Neural Networks, Vol. *5,* No. *6,* November 1994.

Nitish S Zulpe received the MCS degree from SRTM University, Nanded in the year 2004. He received the M.Phil. Degree in Computer Science from Y.C.M.O. University, Nashik in the year 2009. He is currently working as lecturer in the College of Computer Science and Information Technology, Latur, Maharastra. He is leading to PhD degree in University of Pune.

Vrushsen Pawar received MS, Ph.D.(Computer) Degree from Dept .CS & IT, Dr.B.A.M. University & PDF from ES, University of Cambridge, UK. Also Received MCA (SMU), MBA (VMU) degrees respectively. He has received prestigious fellowship from DST, UGRF (UGC), Sakaal foundation, ES London, ABC (USA) etc. He has published 90 and more research papers in reputed national international Journals & conferences. He has recognize Ph.D Guide from University of Pune, SRTM University & Sighaniya University (India). He is senior IEEE member and other reputed society member. Currently working as a Asso. Professor in CS Dept. SRTMU, Nanded.