Texture Feature Selection Using GA for Classification of Human Brain MRI Scans

M. Nouman Tajik[1(✉)], Atiq ur Rehman[1], Waleed Khan[1], and Baber Khan[2]

[1] Department of Electrical Engineering, Comsats Institute of Information Technology (CIIT), Abbottabad, Pakistan
tajiknomi@gmail.com, khanwaleed247@gmail.com, atiqjadoon@ciit.net.pk

[2] Department of Electronics Engineering, International Islamic University (IIUI), Islamabad, Pakistan
baber.khan@iiu.edu.pk

Abstract. Intelligent Medical Image Analysis plays a vital role in identification of various pathological conditions. Magnetic Resonance Imaging (MRI) is a useful imaging technique that is widely used by physicians to investigate different pathologies. Increase in computing power has introduced Computer Aided Diagnosis (CAD) which can effectively work in an automated environment. Diagnosis or classification accuracy of such a CAD system is associated with the selection of features. This paper proposes an enhanced brain MRI classifier targeting two main objectives, the first is to achieve maximum classification accuracy and secondly to minimize the number of features for classification. Two different machine learning algorithms are enhanced with a feature selection pre-processing step. Feature selection is performed using Genetic Algorithm (GA) while classifiers used are Support Vector Machine (SVM) and K-Nearest Neighbor (KNN).

Keywords: Feature selection · Brain MRI · Genetic algorithm · Support vector machine · Classifier · Machine learning · Supervised learning

1 Introduction

Since MRI does not use any ionizing radiation, it is generally safe compared to CT (Computed Tomography) scan. Manual diagnosis of medical images is subjective, time consuming and costly. Looking at an image (visual perception) and interpreting what is seen is often prone to errors due to technician oversights [1]. At the same time due to various imaging constraints and tissue characteristics, automated classification of brain MRI into normal and abnormal studies is also quite difficult.

Classification is an automated process that intends to order every information/data/instance in specific class, in light of the data portrayed by its features. However, without previous knowledge, useful features cannot be determined for classification. So initially it requires an introduction of large number of features for classification of a particular dataset. Introducing a large number of features may include irrelevant and

Y. Tan et al. (Eds.): ICSI 2016, Part II, LNCS 9713, pp. 233–244, 2016.
DOI: 10.1007/978-3-319-41009-8_25

redundant features which are not helpful for classification and this can even lessen the performance of a classifier due to large search space known as "the curse of dimensionality" [2]. This problem can be subsided by selecting just relevant features for grouping. By omitting irrelevant and unnecessary features, feature selection reduces the training time and minimizing the feature set, thus improving the performance of classifier [3, 4].

During the analysis of tissue in MRI by radiologists, image texture plays a pre-dominant role. In fact texture (in) homogeneity is one of the most common individual MRI features used for tumor diagnosis [5]. Studies have shown that texture information can improve accuracy of classification and produce comparable/preferable results to radiologists when used for machine classification of MRI tissue [6]. Different families of texture calculation methods are being used for MRI analysis and it has been shown that combination of texture features from different families can lead to better classification performance [7].

Feature selection has two principal goals; the first one is to boost the classification performance by minimizing the error rate and the second is to reduce feature set. These goals are paradoxical, and the ideal choice should be made in the vicinity of a tradeoff between them. Treating feature selection as a multi-objective problem can obtain a set of non-dominated feature subsets to meet different requirements in real-world applications. Although GA, multi-objective optimization, and feature selection have been individually investigated frequently, there are very few studies on multi-objective feature selection.

The general objective of this paper is to build up a multi objective approach which can achieve high classification accuracy and at the same reducing the feature set to minimum possible features. The goal is achieved by employing GA for feature selection along with two machine learning classifiers for testing the performance. Classifiers used are SVM and KNN, both of which are tested with different classifier parameters and results are compared based on sensitivity, specificity and accuracy. Feature set is composed of two different texture feature families' i.e Grey-Level Co-Occurrence Matrix (GLCM) and Discrete Wavelet Transform (DWT). Feature set contains total 63 features and among those 51 features are extracted using GLCM and 12 are DWT features.

The paper is presented as follows: Sect. 2: Preprocessing. Section 3: Feature extraction process. Section 4: Feature selection using GA. Section 5: Classification via supervised learning. Section 6: Experimental results Sect. 7: Conclusion.

2 Preprocessing

Brain MRI scans contain some redundant tissues and skull portion with an absence of hard intensity boundaries [8] that needs to be removed in order to achieve better performance. Therefore, to remove the unwanted portion of the scan, three threshold boundaries i.e. Lower(LB), Medium(MB) and Upper boundary (UB) are selected from image histogram as shown in Fig. 1.

The objective is to remove all pixels within LB and UB. Simple binarization technique cannot be used for this purpose because some intensity pixels of unwanted

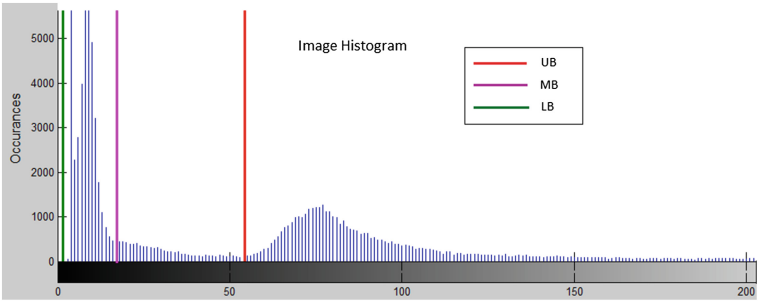


Fig. 1. MRI scan histogram (Color figure online)

skull matter will be left around the brain portion. In order to avoid loss of information in Region of interest (ROI) and to accurately remove the outer strip, the following sequence of steps are followed.

Step 1: (Dividing Image): MRI brain image is divided in two equal halves, shown in Fig. 2(b).

Step 2: (Stripping Outer portion): Using one part at a time, traversing each row of original image $F(x, y)$ from top-left and removing all pixels between MB and UB, A flag $f \in Z_2$ is used to make sure that pixels are removed only from the outer strip $(f = 1)$ and not from the ROI $(f = 0)$. Once all rows are traversed, the 2nd part is flipped and the same procedure to applied to it. This elimination of pixels makes sure that all intensity values between MB and UB become zero (from outer strip only). Once the outer strip is removed, two portions are merged together. Stripped mask $g(x, y)$ is formed from the original image $F(x, y)$ using (1.1).

$$g(x,y) = \begin{cases} \{(\mathrm{MB} < F(x,y) < UB) \,\&\, AND\,(f = 1)\}0 \\ \text{else} \qquad\qquad F(x,y) \end{cases} \tag{1.1}$$

Step 3: (Binary Image): Using binary mask $h(x, y)$ from (1.2) a binary image is obtained to eliminate the background pixels, resulting image from step 2 and step 3 is shown in Fig. 2(c).

$$h(x,y) = \begin{cases} \mathrm{LB} < g(x,y) < UB & 0 \\ F(x,y) \geq \mathrm{UB} & 1 \end{cases} \tag{1.2}$$

Step 4: (Filling Holes): Step 3 result in the loss of pixels from brain area. In order to fill those holes, a matrix of ones $M(x, y)$ of dimension n*n is used where $\{n|n = 2r + 1, r \in \mathbb{N}^+\}$. This mask is applied on all pixels (except the boundary) of image in order to determine its surrounding region. If the area around the pixel is of the majority white, it fills the hole by putting value '1' in that position and value '0' vice versa. Pixel-by-pixel multiplication of $M(x, y)$ with binary image $h(x, y)$ extracts n*n values around single pixel p is represented in (1.3).

$$I(x,y)_{n*n} = M(x,y) * h(x,y) \tag{1.3}$$

Summing up values of $I(x,y)_{n*n}$ provides number of white pixels.

$$\text{Region identifier} = \sum_{k=1}^{n} \sum_{l=1}^{n} I(k,l)_{n*n} \tag{1.4}$$

Holes are filled using (1.5)

$$J(x,y) = \begin{cases} \text{Region identifier} \geq \frac{n*n}{2} & 1 \\ \text{else} & 0 \end{cases} \tag{1.5}$$

Thus, for every pixel p (except boundary) in binary image $h(x,y)$ (1.3), (1.4) and (1.5) are evaluated, resulting image is shown in Fig. 2(d).

Step 5: (Brain Matter Extraction): Once the holes are filled, final gray scale image $S(x,y)$ is obtained by taking product of Binary image $J(x,y)$ and original image $F(x,y)$ using (1.6), result is shown in Fig. 2(e, f).

$$S(x,y) = J(x,y) * F(x,y) \tag{1.6}$$

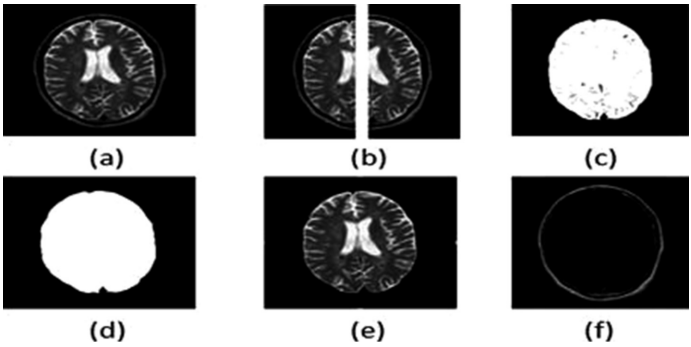


Fig. 2. Preprocessing steps (a) Original scan (b) Dividing image (c) Stripping and binarization (d) Holes Filling (e) Brain Matter (f) Stripped portion

3 Feature Extraction

3.1 GLCM (Gray-Level Co-occurrence Matrix)

GLCM is used to represent the statistical texture features over a spatial domain. It has been proven to be a very powerful tool for image segmentation [9]. Features produced using this method is known as Haralick features, after Haralick et al. [10]. Co-occurrence

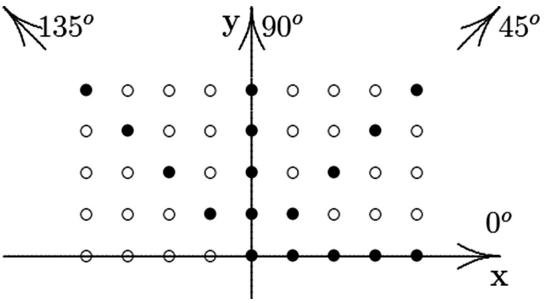


Fig. 3. GLCM angles

matrix is often formed by using two offsets i-e distance (d = 1, 2, 3…) and angle (θ = 0, 45, 90, and 135). For a given distance d, four angular GLCM are calculated as shown in Fig. 3.

The number of gray levels is reduced to avoid the computation cost because GLCM are sensitive to size of texture samples. Sixteen gray levels for GLCM are computed which are sufficient to differentiate brain MRI textures [11]. To avoid direction dependency, Haralik also suggested using the angular mean $M_T(d)$, variance $V_T^2(d)$ and range $R_T(d)$ of GLCM given in (2.1), (2.2) and (2.3) respectively.

$$M_T(d) = \frac{1}{N_\emptyset} \sum\nolimits_\emptyset T(d, \emptyset) \tag{2.1}$$

$$R_T(d) = Max[T(d, \emptyset)] - Min[T(d, \emptyset)] \tag{2.2}$$

$$V_T^2(d) = \frac{1}{N_\emptyset} \sum\nolimits_\emptyset [T(d, \emptyset) - M_T(d)]^2 \tag{2.3}$$

Where $N_\emptyset$ represents the number of angular measurements and $T(d, \emptyset)$ are scalar texture measures.

A total of 17 GLCM features proposed by Haralik are initially computed for d = 1 at four different θ values i.e. 0, 45, 90, and 135. $M_T(d)$, $V_T^2(d)$ and $R_T(d)$ are computed from the extracted features, so a feature vector of 51(17 × 3) GLCM features is obtained.

3.2 Wavelet Transforms (WT)

WT provides a multi-scale analysis of an image such as Information of Horizontal (HL), Vertical (LH), Diagonal (HH) and Approximation (LL). The advantage of DWT compared to Fourier transform is that it provides both frequency and temporal details [12].

Haar, Daubechies and Symlets are used to assess the relating transforms. Total of twelve features are extracted (Four features for each wavelet). Two out of four features

are the mean and variance of energy distribution of level one 2D transform and other two features are mean and variance of energy distribution of level two 2D transform.

For a sub band $b(l, h)$ with the limits $1 < l < L$ and $1 < h < H$ the energy is calculated as:

$$Energy = \frac{1}{LH} \sum_{l=1}^{L} \sum_{h=1}^{H} |b(l,h)|^2 \tag{2.4}$$

4 Feature Selection

The main assumption when using feature selection is that there are a lot of redundant or irrelevant features which sometimes reduces the classification accuracy [13]. Features are evaluated against a fitness function, thus selecting the best rated features among the feature set.

A feature selection is an operator f_s which maps from m dimensional (input) space to n dimensional (output) space given in mapping.

$$f_s : R^{r \times m} \mapsto R^{r \times n} \tag{3.1}$$

Where $m \geq n$ and $m,n \in Z^+$; $R^{r \times m}$ is matrix containing original feature set having r instances; $R^{r \times n}$ is a reduced feature set containing r instances in subset selection.

4.1 Discrete Binary Genetic Algorithm

GA is a heuristic process of natural selection which is inspired from the procedures of evolution in nature. This algorithm uses the Darwin's theory "Survival of fittest" motivated by inheritance, mutation, selection, and crossover. In comparative terminology to human genetics, gene represent feature, chromosome are bit strings and allele is the feature value [14]. From algorithm perspective, population of individuals represented by chromosomes are the arrangement of binary strings in which each bit (gene) represents a specific feature within a Chromosome (bit strings). Chromosomes are evaluated using Objective function (fitness function) which ranks individual chrome by its numerical value (fitness) within a population.

Process of GA

Step 1 (Generation begins): A random population $(a_{11}a_{12}\ldots a_{nm})_2$ matrix p of size n x m is generated using population size n and number of features m shown in (3.2).

$$p = \begin{bmatrix} a_{11} & \cdots & a_{1m} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nm} \end{bmatrix} \tag{3.2}$$

Where $m, n \in Z_2$

Step 2 (Tournament): This phase selects the best-fit individuals for reproduction. Two chromosomes (parent chromes) with the highest fitness will take part in cross over.
Step 3 (Cross Over): Analogous to biological crossover, it is the exchange of bits within the selected parents to produce offspring. Number of bits b selected from parent p_n is computed using (3.3), where parameter: $0 < k < 1$ is a crossover probability.

$$b = k\text{x } p_n \tag{3.3}$$

Step 4 (Mutation): Mutation refers to the change (growth) in the genome of chromosome, flipping of bit strings (genes) of chromosome.
Step 5 (Fitness Evaluation): Analogous to "survival of fittest", chromosomes with a certain level of fitness will survive for next generation while the others whose fitness is less than the threshold value will be discarded.

5 Classification via Supervised Learning

Classification is the method of identification, discrimination of objects or patterns on the basis of their attributes. It is done using supervised learning. In this type of machine learning, machine classifies objects on the basis of previous knowledge. The system is trained using some attributes (features) along with their label, these attributes are used by classifier to guess the unknown objects.

Two classifiers are used here viz. KNN and SVM.

5.1 K Nearest Neighbor Classifier—KNN

It is a simple algorithm that stores all the available patterns/cases and classifies new patterns/cases based on distance function. In binary classification, there are only two classes $\{C_i, C_j\}$; a new unknown case U_n is classified as C_i if the majority of observation specified by parameter k is C_i and vice versa. The parameter k is user-adjustable parameter such that $\{k = 2r + 1, r \in Z^+\}$. The Euclidean distance d between two points a and b in the plane with coordinates (x, y) is given by (4.1)

$$d^2 = (x - a)^2 + (y - b)^2 \tag{4.1}$$

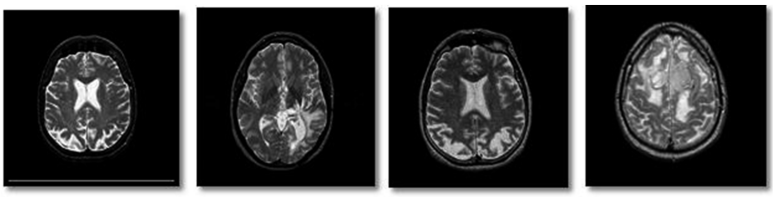
KNN faces genuine difficulties when pattern of distinctive classes overlap in vector space, and sometimes it shows ignorance when it arranges patterns on the premise of checking more number of neighbors which are far separated than the least number of neighbors which are more close together [15]. Three different values of k (1, 3 and 5) are tested for KNN classifier.

5.2 Support Vector Machine—SVM

A supervised classifier, previously, found best for soft tissues classification by Juntu et al. [16]. It is also found to be the second most efficient classification method in all 17 families available today [17]. A hyperplane is made by this classifier in high dimension space which is used as a boundary to classify patterns. A good SVM model is the one which has its hyperplane at quite a large distance from the input data. In this paper, Linear and Polynomial (degree 3) kernel is used for testing SVM accuracy.

6 Experimental Results

Project is carried out on Intel(R) (Core(TM) i5-4530 s CPU @ 2.30 GHz, with 4.00 GB of RAM).MATLAB 8.1.0 (R2013a) is used for simulation. A set of 60 images of size (256 × 256) with a format of Portable Network Graphics (png) are taken from Harvard Medical School website http://www.med.harvard.edu/AANLIB/ among which 20 are normal and remaining 40 are abnormal scans. The abnormal scans consist of three diseases viz. glioma, visual agnosia and meningioma. Samples from dataset are shown in Fig. 4.



(a) Normal Brain *(b) Glioma* *(c) Visual agnosia* *(d) Meningioma*

Fig. 4. Sample images from dataset

Total number of features extracted from both families is:

Number of features:
GLCM = 51
Wavelet = 12
Total = 63

Configuration used for GA is shown in Table 1. Validation is done using 10-fold cross Validation; the data set is divided into ten equal parts (60/10) where each part is known as fold. Nine folds (i.e. 54 images) are used for training and the remaining one fold (i.e. 6 images) is used for testing. The process is repeated 10 times taking one fold from dataset in each iteration to acquire mean accuracy.

Algorithm pseudocode

```
Input : [Population size 'n' , Number of features 'm' , Generation_Limit , Offspring_Limit, Training data, Testing data]
Ouput : [Best Chromosome, Accuracy]
Generation = 1; k = 1;
While Generation < Generation_Limit, do
    Step 1: P(x,y) <--- Generate_Pop(n,m)                    // Generate Binary Population Matrix P(x,y)
    While offspring < Offspring_Limit, do
        Step 2A: [s1,s2,s3,s4] <--- Rand_4_Chrome(P(x,y))    // Pick Random 4 solutions from P(x,y)
        Step 2B: [P1, P2] <--- Tornament(s1,s2,s3,s4)        // Pick Best 2 solutions (i.e. Parent chromes)
        Step 3: [P1', P2'] <--- Cross_Over(P1,P2, C_p)       // Cross the Parents Chromes with Cross probability C_p
        Step 4: [P1",P2"] <--- Mutation(P1', P2', M_p))      // Change the Offspring Chrome with specified probability
    M_p
        [P(k,m),P(k+1,m)] <--- [P1",P2"]                     // Put the offspring chromes in Population Matrix P(x,y)
        k = k+2                                              // Incriment P(x,y) index for future offsprings
        Offspring++                                          // Incriment the offspring counter
Step 5: Best_Chrome <--- Fitness_Evaluation(P(x,y))          // The best solution in Population Matrix for Generation N
Generation ++                                                // Incriment the Generation counter
k = 1                                                        // Initialize k for every new Generation
Step 6: Mean_Accuracy <--- Ten_fold(Best Chrome)             // Evaluating Mean accuracy using ten-fold validation
```

Classification accuracies of selected features using GA-SVM and GA-KNN with different classification parameters are shown in Table 2.

Table 1. Genetic algorithm configuration

Parameters	Value
Population size	500
Genome length	63
Population type	Binary string
Fitness function	KNN/SVM
Number of generations	40
Offspring	500
Cross over	Uniform Crossover
Mutation	Bit inversion
Cross over probability	0.5
Mutation probability	0.2
Selection scheme	Tournament of size 2

Table 2. Classification accuracy of GA-SVM and GA-KNN

Sr	No. of features	SVM kernel	Accuracy	Sensitivity	Specificity
1	3	Linear	98.34 %	95.23 %	100 %
2	12	Linear	96.67 %	100 %	95 %
3	2	Polynomial degree 3	96.67 %	100 %	95 %
4	15	Polynomial degree 3	95 %	90 %	97.5 %
Sr	No. of features	KNN value of 'K'	Accuracy	Sensitivity	Specificity
1	3	K = 1	93.34 %	90 %	95 %
2	2	K = 3	86.6 %	75 %	92.5 %
3	5	K = 5	96.67 %	100 %	95 %

A comparison between selected and isolated texture families in terms of classification accuracy is shown in Table 3. Optimum selected features for both classifiers are shown in Tables 4A and 4B.

For visualation purpose, we had used scatter plot to show how the data are correlated to their respective classes. The plot is made using three isolated features which are deduced from the GA-SVM. Samples are concentrated in the vicinity of its class-type which shows how effectively the data is classified by the SVM (linear), shown in Fig. 5.

Table 3. Isolated feature families vs. selected features

Texture family	KNN k = 1	KNN k = 3	KNN k = 5	SVM polynomial degree 3	SVM Linear
All features (GLCM + Wavelet)	71.67 %	70 %	70 %	96.67 %	96.67 %
GLCM features	86.67 %	76.67 %	75 %	92 %	96.67 %
Wavelet features	71.66 %	66.67 %	66.67 %	81.67 %	80 %
Reduced features	93.34 %	86.6 %	96.67 %	96.67 %	98.34 %

Table 4A. Optimum selected features for KNN

GA-KNN (k = 1)	GA-KNN (k = 3)	GA-KNN (k = 5)
(i) Difference entropy (mean) (ii) Mean Y(mean) (iii) Standard deviation X (range)	(i) Variance (range) (ii) Entropy (range)	(i) Cluster prominence (mean) (ii) Entropy (mean) (iii) Mean of energy distribution of 2D Haar transform (iv) Mean of energy distribution of 2D Daubechies 5 transform (v) Mean of energy distribution of 2D Symantic transform
Three/93.34 %	Two/86.6 %	Five/96.67 %

Table 4B. Optimum selected features for SVM

GA-SVM polynomial degree 3	GA-SVM linear
(i)Homogeneity(mean) (ii)Sum average (mean)	(i)Difference entropy(mean) (ii)Mean Y(mean) (iii)Deviation X(range)
Two/96.67 %	Three/98.34 %

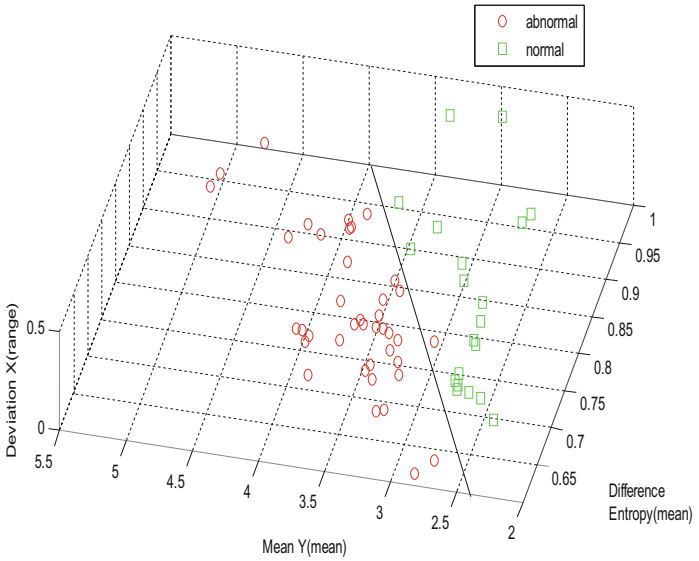


Fig. 5. Scatter plot of normal and abnormal scans using only three features (Color figure online)

7 Conclusion and Future Work

Combining two different texture families for classification has enhanced the accuracy of the classifier. Results reveal the superiority of combination of texture families over the isolated texture family for classification. Similarly feature selection process has also enhanced the classification results. It is clearly seen, that without feature selection the classifier performance is weak, when compared with classification after feature selection. GA-SVM has improved classification accuracy using least number of features. The proposed work can be extended to classify different abnormalities in brain, for example, Alzheimer's disease, visual agnosia, Glioma with tumor, Herpes encephalitis with a tour, bronchogenic carcinoma and Multiple sceloris with a tour. We encourage our readers to analyze the method and results in the proposed paper to help us set a better approach, thereby, reducing computational cost and further increase mean-accuracy for the classification of T2-Weighted Human Brain MRI scans.

References

1. Castellino, R.A.: Computer aided detection (CAD): an overview. Cancer Imaging **5**, 17–19 (2005)
2. Gheyas, I.A., Smith, L.S.: Feature subset selection in large dimensionality domains. Pattern Recognit. **43**(1), 5–13 (2010)
3. Dash, M., Liu, H.: Feature selection for classification. Intell. Data Anal. **1**(1–4), 131–156 (1997)

4. Unler, A., Murat, A.: A discrete particle swarm optimization method for feature selection in binary classification problems. Eur. J. Oper. Res. **206**(3), 528–539 (2010)
5. De Schepper, A., Vanhoenacker, F., Parizel, P., Gielen, J. (eds.): Imaging of Soft Tissue Tumors, 3rd edn. Springer, Berlin (2005)
6. Atiq ur, R. et al.: Hybrid feature selection and tumor identification in brain MRI using swarm intelligence. In: 11th International Conference on Frontiers of Information Technology (2013)
7. García, M.A., Puig, D.: Improving Texture Pattern Recognition by Integration of Multiple Texture Feature Extraction Methods, pp. 7–10 (2002)
8. Sidra et al.: Improved tissue segmentation algorithm using modified Gustafson-kessel Clustering for brain MRI (2014)
9. du Buf, J.M.H., Kardan, M., Spann, M.: Texture feature performance for image segmentation. Pattern Recogn. **23**(3/4), 291–309 (1990)
10. Haralick, R.M., Shanmugam, K., Dinstein, I.H.: Textural Features for Image Classification. IEEE Trans. Syst. Man Cybern. B Cybern. **SMC-3**(6), 610–621 (1973)
11. Fritz et al.: Statistical Texture Measures Computed from Gray Level Concurrence Matrices, 5 November 2008
12. Prasad, B.: Speech, Audio, Image and Biomedical Signal Processing Using Neural Networks. Springer, Berlin (2008). 356 p
13. Bruzzone, L., Persello, C.: A novel approach to the selection of robust and invariant features for classification of hyperspectral images. Department of Information Engineering and Computer Science, University of Trento (2010)
14. Sivanandam, S.N., Deepa, S.N.: Introduction to Genetic Algorithms. Springer, Heidelberg (2008)
15. Yao, M.: Research on learning evidence improvement for KNN based classification algorithm. Int. J. Database Theory Appl. **7**(1), 103–110 (2014)
16. Juntu, J., De Schepper, A.M., Van Dyck, P., VanDyck, D., Gielen, J., Parizel, P.M., Sijbers, J.: Classification of Soft Tissue Tumors By Machine Learning Algorithms (2011)
17. Ferńandez-Delgado, M., et al.: Do we need hundreds of classifiers to solve real world classification problems? J. Mach. Learn. Res. **15**, 3133–3181 (2014)