Progress In Electromagnetics Research, Vol. 109, 325–343, 2010

A NOVEL METHOD FOR MAGNETIC RESONANCE BRAIN IMAGE CLASSIFICATION BASED ON ADAPTIVE CHAOTIC PSO

Y. Zhang, S. Wang, and L. Wu

School of Information Science and Engineering
Southeast University, Nanjing, China

Abstract—Automated and accurate classification of magnetic resonance (MR) brain images is an integral component of the analysis and interpretation of neuroimaging. Many different and innovative methods have been proposed to improve upon this technology. In this study, we presented a forward neural network (FNN) based method to classify a given MR brain image as normal or abnormal. This method first employs a wavelet transform to extract features from images, and then applies the technique of principle component analysis (PCA) to reduce the dimensions of features. The reduced features are sent to an FNN, and these parameters are optimized via adaptive chaotic particle swarm optimization (ACPSO). K-fold stratified cross validation was used to enhance generalization. We applied the proposed method on 160 images (20 normal, 140 abnormal), and found that the classification accuracy is as high as 98.75% while the computation time per image is only 0.0452 s.

1. INTRODUCTION

Magnetic resonance imaging (MRI) is an imaging technique that produces high quality images of the anatomical structures of the human body, especially in the brain, and provides critical information for clinical diagnosis and biomedical research [1–6]. The diagnostic values of MRI are greatly magnified by the automated and accurate classification of the MR images.

Wavelet transform is an effective tool for feature extraction because it allows for the analysis of images at various levels of resolution [7]. This technique requires large storage and is

Received 1 September 2010, Accepted 18 October 2010, Scheduled 25 October 2010

Corresponding author: Yu-Dong Zhang (zhangyudongnuaa@gmail.com).

computationally more expensive [8]. In order to reduce the feature vector dimensions and increase the discriminative power, the principal component analysis (PCA) method has been used. PCA is appealing since it effectively reduces the dimensionality of the data and therefore reduces the computational cost of analyzing new data [9]. Subsequently, we need to construct a classifier, which presents a challenge to current researchers.

In recent years, researchers have proposed two categories of approaches to obtain this goal. The first category is supervised classification, such as support vector machine (SVM) [10] and k-nearest neighbors (k-NN) [11]. The other category is unsupervised classification, such as self-organization feature map (SOFM) and fuzzy c-means [12]. While both of these methods achieved satisfactory results, supervised classification performs better than unsupervised classification in terms of classification accuracy (successful classification rate) [13].

In this study, the forward neural network (FNN) was chosen as the classifier because it is a powerful tool among supervised classifiers and it can classify nonlinear separable patterns and approximate an arbitrary continuous functions [14]. Recently, there have been many algorithms available to train the FNN, such as back-propagation (BP) algorithm [15], genetic algorithm (GA) [16], simulating annealing (SA) algorithm [17], and particle swarm optimization (PSO) [18]. Unfortunately, the BP, GA, and SA algorithms all demand expensive computational costs. However, PSO is well-known for its lower computational costs and its most attractive feature is that it requires less computational bookkeeping and only a few lines of implementation codes. In order to improve the performance of PSO, we proposed an adaptive chaotic PSO (ACPSO) method to find the optimal parameters of FNN.

In total, our method consisted of three stages: feature extraction, feature reduction, and NN-based classification as shown in Fig. 1. This is a canonical and standard classification method which has already proven to be the best classification method [19].

The structure of the rest of this paper was organized as follows: Section 2 presented wavelet transform based methods for feature extraction. PCA technique for feature reduction was introduced in

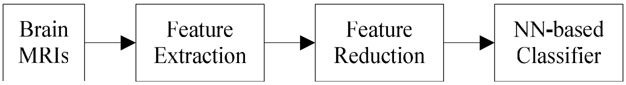


Figure 1. Methodology of our proposed algorithm.

Section 3. Section 4 proposed a novel adaptive chaotic PSO (ACPSO) method. Section 5 presented the structure and training approach of the FNN, introduced the vector encoding strategy, and discussed the K-fold stratified cross validation method to enhance generalization. Experiments in Section 6 demonstrated the effectiveness and rapidness of our proposed algorithm. We concluded this paper in Section 7.

2. FEATURE EXTRACTION

The wavelet theory has already been used in the field of brain image classification by Chaplot [20] and El-Dahshan [21]. Their methods will be compared in the experimental section. Here, we only simply introduce the basic principles of DWT to avoid replication.

2.1. Discrete Wavelet Transform

The DWT is a powerful implementation of the WT using the dyadic scales and positions. The basic fundamental principle of DWT is introduced as follows: Suppose $x(t)$ is a square-integrable function, then the continuous WT of $x(t)$ relative to a given wavelet $\psi(t)$ is defined as

$$W_\psi(a,b)=\int_{-\infty}^{\infty}x(t)\psi_{a,b}(t)dt \tag{1}$$

where

$$\psi_{a,b}(t)=\frac{1}{\sqrt{a}}\psi\left(\frac{t-a}{b}\right) \tag{2}$$

Here, the wavelet $\psi_{a,b}(t)$ is calculated from the mother wavelet $\psi(t)$ by translation and dilation: a is the dilation factor and b is the translation parameter (both real positive numbers). There are several different wavelets that have gained popularity throughout the development of wavelet analysis. The most important of these is the Harr wavelet, which is the simplest and preferred wavelet in many applications [22].

Equation (1) can be discretized by restraining a and b to a discrete lattice ($a=2^b$ & $a>0$) to give the DWT, which can be expressed as follows.

$$\begin{aligned} ca_{j,k}(n)&=DS\left[\sum\nolimits_n x(n)g_j^*\left(n-2^jk\right)\right]\\ cd_{j,k}(n)&=DS\left[\sum\nolimits_n x(n)h_j^*\left(n-2^jk\right)\right]\end{aligned} \tag{3}$$

Here, the coefficients $ca_{j,k}$ and $cd_{j,k}$ refer to the approximation components and the detail components, respectively. The functions $g(n)$ and $h(n)$ denote the low-pass filter and high-pass filter,

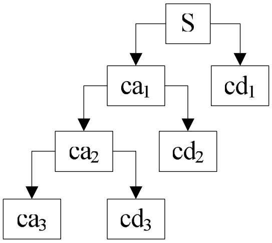


Figure 2. A 3-level wavelet decomposition tree.

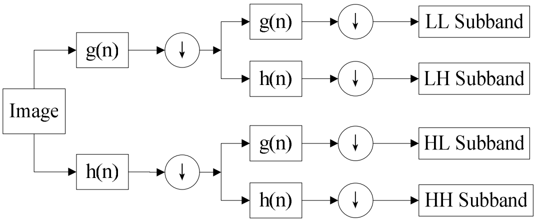


Figure 3. Schematic diagram of 2D DWT.

respectively. The subscripts j and k represent the wavelet scale and translation factors, respectively. The DS operator is used for down sampling.

The above decomposition process can be iterated with successive approximations being decomposed so that one signal is broken down into various levels of resolution [23]. The whole process is called a wavelet decomposition tree as shown in Fig. 2.

2.2. 2D DWT

In applying this technique to MR images, the DWT is applied separately to each dimension. Fig. 3 illustrates the schematic diagram of 2D DWT. As a result, there are 4 sub-band (LL, LH, HH, HL) images at each scale. The sub-band LL is used for the next 2D DWT.

The LL subband can be regarded as the approximation component of the image, while the LH, HL, and HH subbands can be regarded as the detailed components of the image. As the level of the decomposition increased, we obtained more compact yet coarser approximation components. Thus, wavelets provide a simple hierarchical framework for interpreting the image information. In our algorithm, level-3 decomposition via Harr wavelet was utilized to extract features.

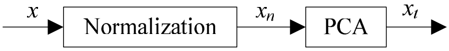


Figure 4. Using Normalization before PCA.

3. FEATURE REDUCTION

Excessive features increase computation time and storage memory which sometimes causes the classification process to become more complicated. This consequence is called the curse of dimensionality. A strategy is necessary to reduce the number of features used in classification.

PCA is an efficient tool to reduce the dimension of a data set consisting of a large number of interrelated variables while retaining the most significant variations. It is achieved by transforming the data set to a new set of ordered variables according to their degree of variance or importance. This technique has three effects: (i) it orthogonalizes the components of the input vectors so that they are uncorrelated with each other, (ii) it orders the resulting orthogonal components so that those with the largest variation come first, and (iii) it eliminates the components in the data set that contributing the least variation.

It should be noted that the input vectors should be normalized to have zero mean and unity variance before performing PCA, which is shown in Fig. 4. The normalization is a standard procedure. Details about PCA can be seen in Ref. [24].

4. ADAPTIVE CHAOTIC PSO

4.1. Introduction of PSO

PSO is a population based stochastic optimization technique, which simulates the social behavior of a swarm of bird, flocking bees, and fish schooling [25]. By randomly initializing the algorithm with candidate solutions, the PSO successfully leads to a global optimum. This is achieved by an iterative procedure based on the processes of movement and intelligence in an evolutionary system. Fig. 5 shows the flow chart of a PSO algorithm.

In PSO, each potential solution is represented as a particle. Two properties (position x and velocity v) are associated with each particle. Suppose x and v of the ith particle are given as

$$x = (x_{i1}, x_{i2}, \cdots, x_{iN}) \tag{4}$$

$$v = (v_{i1}, v_{i2}, \cdots, v_{iN}) \tag{5}$$

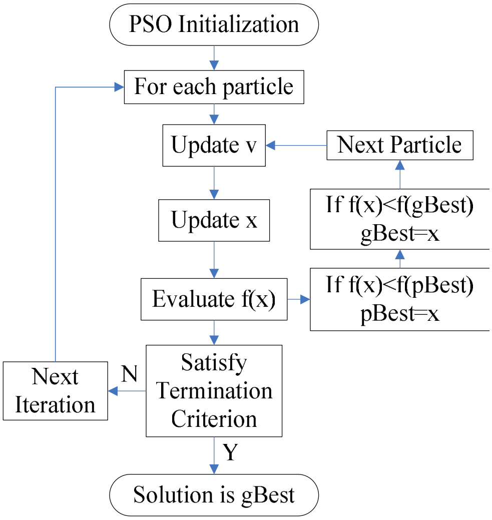


Figure 5. Flow chart of the PSO algorithm.

where N stands for the dimensions of the problem. In each iteration, a fitness function is evaluated for all the particles in the swarm. The velocity of each particle is updated by keeping track of the two best positions. One is the best position a particle has traversed so far and called "*pBest*". The other is the best position that any neighbor of a particle has traversed so far. It is a neighborhood best called "*nBest*". When a particle takes the whole population as its neighborhood, the neighborhood best becomes the global best and is accordingly called "*gBest*". Hence, a particle's velocity and position are updated as follows

$$v = \omega \cdot v + c_1 r_1 (pBest - x) + c_2 r_2 (nBest - x) \tag{6}$$

$$x = x + v\Delta t \tag{7}$$

where ω is called the "*inertia weight*" that controls the impact of the previous velocity of the particle on its current one. The parameters c_1 and c_2 are positive constants, called "*acceleration coefficients*". The parameters r_1 and r_2 are random numbers that are uniformly distributed in the interval [0, 1]. These random numbers are updated every time when they occur. The parameter Δt stands for the given time-step.

The population of particles is then moved according to (6) and (7), and tends to cluster together from different directions. However, a maximum velocity $v_{\max}$, should not be exceeded by any particle to keep the search within a meaningful solution space. The PSO algorithm runs through these processes iteratively until the termination criterion is satisfied.

4.2. Our Proposed Algorithm-ACPSO

The PSO program has proven to be very effective for solving global optimization. It is a recently invented high-performance optimizer that is very easy to understand and implement, and it also requires little computational bookkeeping and only a few lines of code.

The parameters (r_1, r_2) were generated by pseudo-random number generators (RNG) in classical PSO. The RNG cannot ensure the optimization's ergodicity in solution space because they are absolutely random; therefore, a chaotic operator was employed to generate parameters (r_1, r_2) by the following formula:

$$r_i(t+1) = 4.0 * r_i(t) * [1 - r_i(t)] \quad i = 1, 2 \tag{8}$$

Another improvement lies in changing the parameters (ω, c_1, c_2) adaptively. In the search process of PSO, the search space will gradually reduce as the generation increases. Therefore, we hope to search an expansive area with low precision at the prophase stage while search a restricted area with high precision at the anaphase stage as listed in Table 1. The detailed formulas of those adaptive parameters are as follows:

$$\omega = \omega_i - \frac{\omega_i - \omega_f}{MaxGeneration} * Generation \quad (\omega_i > \omega_f) \tag{9}$$

$$c_1 = c_{1i} - \frac{c_{1i} - c_{1f}}{MaxGeneration} * Generation \quad (c_{1i} > c_{1f}) \tag{10}$$

$$c_2 = c_{2i} - \frac{c_{2i} - c_{2f}}{MaxGeneration} * Generation \quad (c_{2i} < c_{2f}) \tag{11}$$

Here, the indexes i and f denotes "initial" and "final", respectively.

Table 1. Parameters variation.

	Ω	c_1	c_2	Performance
Prophase	Larger	Larger	Smaller	PSO Searches for global optimal in an expansive area with low precision
Anaphase	Smaller	Smaller	Larger	PSO Searches for local optimal in a limited area with high precision

5. FNN WITH ACPSO

5.1. Structure

Neural networks are widely used in pattern classification since they do not need any information about the probability distribution and the *a priori* probabilities of different classes [26]. A single-hidden-layer backpropagation neural network is adopted with sigmoid neurons in the hidden layer and a linear neuron in the output layer.

The training vectors were presented to the NN, which is trained in batch mode. The network configuration is *NI* × *NH* × *NO.* This consists of a two-layer network with *NI* input neurons, *NH* neurons in the hidden layer, and *NO* output indicating the brain is normal or abnormal. The structure is depicted in Fig. 6. In this study, since there are only 2 classes (normal brain or abnormal brain), only one output neuron is sufficient. The number 1 denotes for normal and 0 for abnormal. The setting of the other two parameters, *NI* and *NH*, will be discussed in the experiment section.

5.2. Training with ACPSO

From Fig. 6, it can be seen that the output of the jth hidden node is

$$\begin{aligned} f(s_j) &= 1/[1+\exp(-s_j)] \\ s_j &= \sum_{i=1}^{NI} \omega_{ij} x_i + b_j \end{aligned} \tag{12}$$

Here, s_j denotes the weight input sum in the hidden layer, ω_{ij} denotes the connection weight from the ith input neuron to the jth hidden neuron, b_j denotes the bias of the jth hidden neuron, and f denotes the activation function (Sigmoid function). The output of the kth

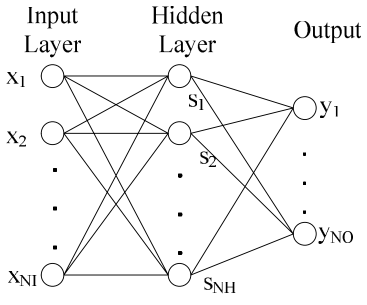


Figure 6. A two-layer FNN used for brain classification.

neuron of output layer is

$$y_k = \sum_{j=1}^{NH} \nu_{jk} f(s_j) + \beta_k \tag{13}$$

where ν_{jk} denotes the connection weight from the jth hidden neuron to the kth output neuron, and β_k denotes for the bias of the kth output neuron.

The learning median square error (MSE) [27] can be calculated as

$$MSE = \frac{1}{NS}\sum_{i=1}^{NS} E_i^2 = \frac{1}{NS}\sum_{i=1}^{NS}\left(y_k^i - d_k^i\right)^2 \tag{14}$$

where NS is the number of training samples, and the term $y_k^i - d_k^i$ is the error of the actual output and desired output of the kth output neuron of ith sample. We defined the fitness function as MSE, and then we used ACPSO to find the optimal weights so that the fitness reaches the minimum.

5.3. Vector Encoding Strategy

In order to expedite the global optimization process, vector encoding strategy is utilized where each particle encodes for a row vector. For example, the FNN with structure of $2 \times 2 \times 1$ has the total number of weights $(NI + 1) * NH + (NH + 1) * NO = (2 + 1) * 2 + (2 + 1) * 1 = 9$. The corresponding encoding style can be presented as

$$\text{particle } (i) = [\omega_{11}\omega_{21}b_1\omega_{12}\omega_{22}b_2\nu_{11}\nu_{21}\beta_1] \tag{15}$$

And the particle matrix can be represented as

$$\text{Particle Matrix} = \begin{bmatrix} \text{Particle}(1) \\ \text{Particle}(2) \\ \dots \\ \text{Particle}(NP) \end{bmatrix} \tag{16}$$

where NP denotes the number of particles of ACPSO.

5.4. *K*-fold Stratified Cross Validation

Cross validation methods consist of three types: Random subsampling, K-fold cross validation, and leave-one-out validation. The K-fold cross validation is applied due to its properties as simple, easy, and using all data for training and validation. The mechanism is to create a K-fold partition of the whole dataset, repeat K times to use $K - 1$ folds for training and a left fold for validation, and finally average the error rates

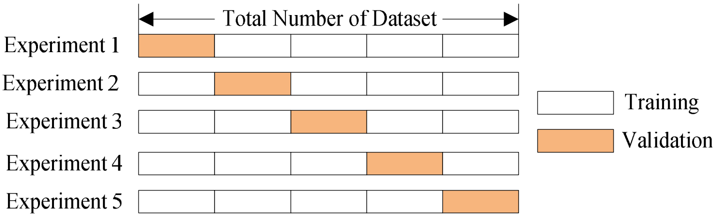


Figure 7. A 5-fold cross validation.

of K experiments. The schematic diagram of 5-fold cross validation is shown in Fig. 7.

The K folds can be purely random partitioned, however, some folds may have a quite different distributions from other folds. Therefore, stratified K-fold cross validation was employed, with which every fold has nearly the same class distributions [28]. Another challenge is to determine the number of folds. If K is set too large, the bias of the true error rate estimator will be small, but the variance of the estimator will be large and the computation will be time-consuming. Alternatively, if K is set too small, the computation time will decrease, the variance of the estimator will be small, but the bias of the estimator will be large [29]. In this study, we empirically determined K as 5 through the trial-and-error method.

6. EXPERIMENTS AND DISCUSSIONS

The experiments were carried out on the P4 IBM platform with 3 GHz main frequency and 2 GB memory running under the Windows XP operating system. The algorithm was developed via the wavelet toolbox, the neural network toolbox, and the statistical toolbox of Matlab 2009b (The Mathworks ©). The programs can be run or tested on any computer platforms where Matlab is available.

6.1. Database

The datasets consists of T2-weighted MR brain images in axial plane and 256 × 256 in-plane resolution, which were downloaded from the Harvard Medical School website (http://med.harvard.edu/AANLIB/). The abnormal brain MR images of the dataset consist of the following diseases: glioma, meningioma, Alzheimer's disease, Alzheimer's disease plus visual agnosia, Pick's disease, sarcoma, and Huntington's disease. A sample of each is shown in Fig. 8.

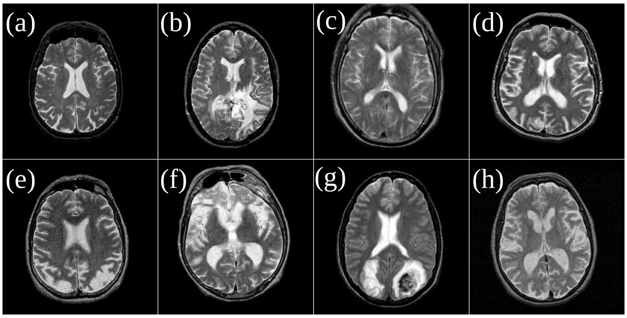


Figure 8. Sample of brain MRIs: (a) normal brain; (b) glioma; (c) meningioma; (d) Alzheimer's disease; (e) Alzheimer's disease with visual agnosia; (f) Pick's disease; (g) sarcoma; (h) Huntington's disease.

Table 2. Setting of training and test images.

Total No. of images	No. of images in training area (128)		No. of images in validation area (32)	
	Normal	Abnormal	Normal	Abnormal
160	16	112	4	28

We randomly selected 20 images for each type of brain. Since there are one type of normal brain and seven types of abnormal brain in the dataset, 160 images was selected consisting of 20 normal and 140 (= 7 types of diseases × 20 images/diseases) abnormal brain images. The setting of the training images and validation images is shown in Table 2 since 5-fold cross validation was used.

6.2. Feature Extraction

The three levels of wavelet decomposition greatly reduce the input image size as shown in Fig. 9. The top left corner of the wavelet coefficients image denotes the approximation coefficients of level-3 whose size is only $32 \times 32 = 1024$. The border distortion should be noticed. In our algorithm the symmetric padding method [30] was utilized to calculate the boundary value.

6.3. Feature Reduction

As stated above, the dimensions of extracted features were reduced from 65536 to 1024. However, this is still results in a high computation

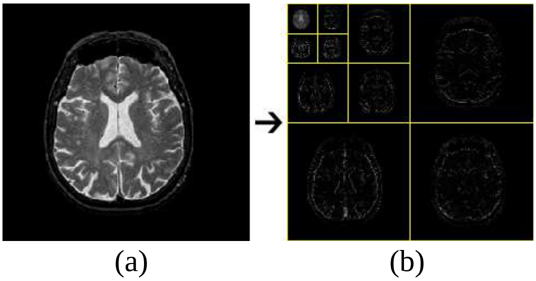


Figure 9. An example of 3-level 2D DWT: (a) normal brain MRI; (b) level-3 wavelet coefficients.

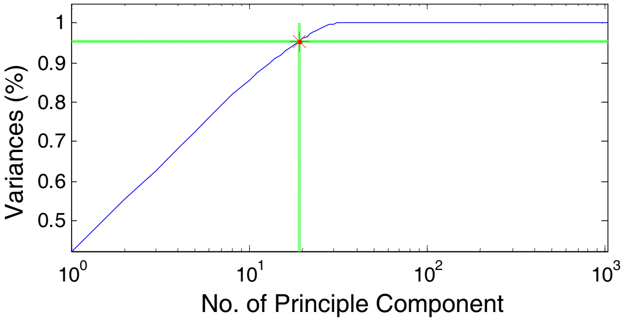


Figure 10. Variances against No. of principle components (x axis is log scale).

Table 3. Detailed data of PCA.

No. of Prin. Comp.	1	2	3	4	5	6	7	8	9	10
Variance (%)	42.3	55.6	62.4	68.1	72.3	76.2	79.3	82.1	84.0	85.6
No. of Prin. Comp.	11	12	13	14	15	16	17	18	**19**	20
Variance (%)	87.3	88.6	89.8	91.0	92.0	93.0	93.9	94.6	**95.4**	96.1

cost. Thus, PCA is used to further reduce the dimensions of features on another level. The curve of cumulative sum of variance with number of principle components is shown in Fig. 10.

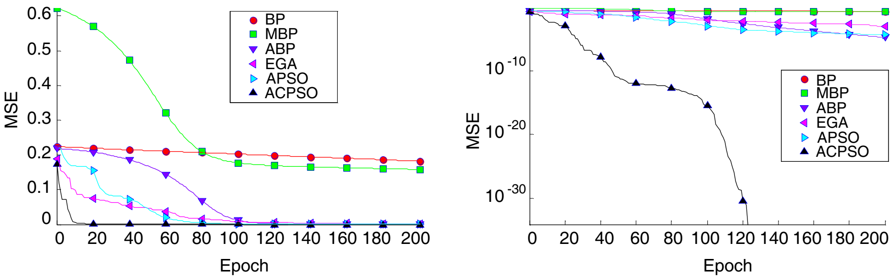
The variance with number of principle components from 1 to 20 are listed in Table 3. It shows that 19 principle components (bold font in table), which are only 1.86% of the original features, preserve 95.4% of the total variance.

6.4. FNN Training

The 19 principle components were then directly sent to the FNN. Thus, the NI is 19 and the NH is determined as 10 according the information entropy method [31]. Consequently, the structure of the neural network is 19-10-1.

Here, we choose the elite GA algorithm with migration (EGA) [32] and weight-adaptive PSO (APSO) method [33] as the comparative methods. The parameters of EGA, APSO, and ACPSO are acquired using trial-and-error method and listed in Table 4. We also compared our method with traditional BP, momentum BP (MBP), adaptive BP (ABP). The results are shown in Fig. 11.

It is clear that our proposed method can converge to less than 10^{-30} at 124th epoch, which is the fastest among all algorithms. The detailed data of MSE versus epoch are shown in Table 5, from which



(a) Linear scale of y axis

(b) Logarithmic scale of y axis

Figure 11. Training performance.

Table 4. Parameters of GA, APSO, and ACPSO.

GA		APSO		ACPSO	
Parameters	Values	Parameters	Values	Parameters	Values
NP	30	NP	30	NP	30
$MaxGeneration$	200	$MaxGeneration$	200	$MaxGeneration$	200
$P_{Crossover}$	0.8	$v_{\max}$	1	$v_{\max}$	1
$P_{mutation}$	0.1	ω_i	0.9	ω_i	0.9
P_{elite}	0.1	ω_f	0.4	ω_f	0.4
$Migration\ Interval$	20	c_1	1	c_{1i}	2.5
$P_{migration}$	0.2	c_2	1	c_{1f}	0.5
				c_{2i}	0.5
				c_{2f}	2.5

Table 5. MSE versus epoch (every 25 epoch).

Epoch	0	25	50	75	100	125	150	175	200
BP	0.2241	0.2184	0.2130	0.2078	0.2027	0.1976	0.1926	0.1876	0.1826
MBP	0.6237	0.5549	0.4078	0.2329	0.1778	0.1684	0.1645	0.1613	0.1582
ABP	0.2190	0.2051	0.1705	9.476e-2	1.326e-2	1.567e-3	2.912e-4	6.588e-5	1.626e-5
EGA	0.1878	7.068e-2	4.640e-2	1.829e-2	7.479e-3	4.836e-3	2.832e-3	1.762e-3	8.386e-4
APSO	0.2202	9.904e-2	4.509e-2	6.969e-3	8.663e-4	1.884e-4	1.159e-4	5.739e-5	3.653e-5
ACPSO	0.1743	3.396e-5	8.433e-12	4.535e-13	3.977e-16	9.3378e-35	Algorithm Terminate		

Table 6. Confusion matrix of our method (O denotes for output, T denotes for Target).

	Normal (O)	Abnormal (O)
Normal (T)	19	1
Abnormal (T)	1	139

the rank of algorithms can be regarded as ACPSO > ABP > APSO > EGA > MBP > BP. The order of the algorithm performance will guide us in future research.

6.5. Classification Accuracy

The confusion matrix of our method is listed in Table 6. The element of the ith row and the jth column represents the classification accuracy belonging to class i and is assigned to class j after the supervised classification.

The results show that our method only misclassified 2 images of the total 160 MRI brain images. The whole classification accuracy is $(19+139)/160 = 98.75\%$. Moreover, we compared the results obtained from our method with other results (DWT+SOM [20], DWT+SVM with linear kernel [20], DWT+SVM with radial basis function based kernel [20], DWT+PCA+ANN [21], and DWT+PCA+kNN [21]) described in the recent literature that used the same MRI datasets, as shown in Table 7. It is clear that our proposed method earns the highest classification accuracy.

Table 7. Classification Accuracy comparison for the same MRI dataset.

Approach	Classification Accuracy (%)
DWT+SOM [20]	94
DWT+SVM with linear kernel [20]	96
DWT+SVM with radial basis function based kernel [20]	98
DWT+PCA+ANN [21]	97
DWT+PCA+kNN [21]	98
DWT+PCA+ACPSO-FNN (Our method)	**98.75**

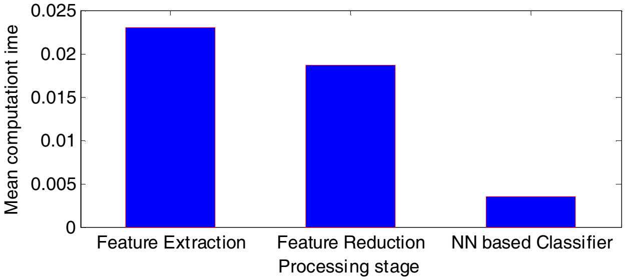


Figure 12. Computation times at different stages.

6.6. Time Analysis

Computation time is another important factor used to evaluate the classifier. The time for network training is not considered, since the weights/biases of the NN should remain unchanged unless the property of images changes a lot. We sent all 160 images into the classifier, recorded the corresponding computation time, computed the mean value, and depicted the consumed time of different stages shown in Fig. 12.

For each image, the computation time for feature extraction, feature reduction, and NN based classifier is 0.023 s, 0.0187 s, and 0.0035 s, respectively. The feature extraction stage is the most time-consuming, which should be improved in future research. The total computation time for each image is about 0.0452 s, which is fast enough for a real time diagnosis.

7. CONCLUSION

In this study, we have developed a novel hybrid classifier to distinguish between normal and abnormal MRIs of the brain. The method obtained 98.75% classification accuracy on both training and test images of the selected datasets. Future work should focus on the following aspects: First, the proposed method could be employed for MR images with other contrast mechanisms such as T1-weighted, proton-density-weighted, and diffusion-weighted images. Second, the computation time could be accelerated by using advanced wavelet transforms such as the lift-up wavelet. Third, Multi-classification, which is focuses on specific brain MRIs disorders, can also be explored in the future.

The discrete wavelet transform can efficiently extract the information from original MR images with litter loss. The advantage of DWT over Fourier Transforms is the spatial resolution, namely, DWT captures both frequency and location information. In current study we choose the Harr wavelet, although there are other outstanding wavelets such as Daubechies series. We will compare the performance of different families of wavelet in future work. Another research direction lies in the stationary wavelet transform and the wavelet packet transform.

The importance of PCA was demonstrated in the discussion section. If we omitted the PCA procedures, we meet a huge search space which will cause heavy computation burden and worsened classification accuracy. There are some other excellent feature transformation methods such as ICA, manifold learning. In the future, we will focus on investigating the performance of these algorithms.

The proposed ACPSO shows superiority to BP, MBP, ABP, EGA and APSO. This is because the ACPSO integrate the adaptive mechanism, chaotic mechanism into canonical PSO algorithm. It is more robust and swifter compared with other algorithms. Therefore, we will apply the ACPSO to other industrial fields.

The most important contribution of this paper is that the integration of PCA, FNN, and ACPSO method presented as a powerful tool for identifying normal MR images from abnormal MR images. This technique of brain MRI classification based on FNN is a potentially valuable tool to be used in computer assisted clinical diagnosis. It would be useful in finding a correlation between such measurements from MR images and behavioral or physiological parameters of research populations.

REFERENCES

1. Mohsin, S. A., N. M. Sheikh, and W. Abbas, "MRI induced heating of artificial bone implants," *Journal of Electromagnetic Waves and Applications*, Vol. 23, No. 5, 799–808, 2009.
2. Cobos Sanchez, C., S. G. Garcia, L. D. Angulo, C. M. De Jong Van Coevorden, and A. Rubio Bretones, "A divergence-free BEM method to model quasi-static currents: Application to MRI coil design," *Progress In Electromagnetics Research B*, Vol. 20, 187–203, 2010.
3. Mohsin, S. A., N. M. Sheikh, and U. Saeed, "MRI induced heating of deep brain stimulation leads: Effect of the air-tissue interface," *Progress In Electromagnetics Research*, Vol. 83, 81–91, 2008.
4. Ravaud, R. and G. Lemarquand, "Magnetic field in MRI yokeless devices: Analytical approach," *Progress In Electromagnetics Research*, Vol. 94, 327–341, 2009.
5. Mishra, M., N. Gupta, A. Dubey, and S. Shekhar, "Application of quasi monte carlo integration technique in efficient capacitance computation," *Progress In Electromagnetics Research*, Vol. 90, 309–322, 2009.
6. Hynynen, K., "MRI-guided focused ultrasound treatments," *Ultrasonics*, Vol. 50, No. 2, 221–229, Ultrasonics, 2010.
7. Danesfahani, R., et al., "Applying shannon wavelet basis functions to the method of moments for evaluating the radar cross section of the conducting and resistive surfaces," *Progress In Electromagnetics Research B*, Vol. 8, 257–292, 2008.
8. Valsan, S. P. and K. S. Swarup, "Wavelet transform based digital protection for transmission lines," *International Journal of Electrical Power & Energy Systems*, Vol. 31, No. 7–8, 379–388, 2009.
9. Camacho, J., J. Picó, and A. Ferrer, "Corrigendum to 'The best approaches in the on-line monitoring of batch processes based on PCA: Does the modelling structure matter?' [*Analytica Chimica Acta* Vol. 642, 59–68, 2009.]," *Analytica Chimica Acta*, Vol. 658, No. 1, 106–106, 2010.
10. Bigler, D. C., et al., "STAMPS: Software tool for automated MRI post-processing on a supercomputer," *Computer Methods and Programs in Biomedicine*, Vol. 658, No. 1, 146–157, 2009.
11. Cocosco, C. A., A. P. Zijdenbos, and A. C. Evans, "A fully automatic and robust brain MRI tissue classification method," *Medical Image Analysis*, Vol. 7, No. 4, 513–527, 2003.
12. Yeh, J.-Y. and J. C. Fu, "A hierarchical genetic algorithm

for segmentation of multi-spectral human-brain MRI," *Expert Systems with Applications*, Vol. 34, No. 2, 1285–1295, 2008.

13. Phaiboon, S. and P. Phokharatkul, "Path loss prediction for low-rise buildings with image classification on 2-D aerial photographs," *Progress In Electromagnetics Research*, Vol. 95, 135–152, 2009.
14. Coulibaly, P. and N. D. Evora, "Comparison of neural network methods for infilling missing daily weather records," *Journal of Hydrology*, Vol. 341, No. 1–2, 27–41, 2007.
15. Parappagoudar, M. B., D. K. Pratihar, and G. L. Datta, "Forward and reverse mappings in green sand mould system using neural networks," *Applied Soft Computing*, Vol. 1, No. 1, 239–260, 2008.
16. Robotham, H., et al., "Acoustic identification of small pelagic fish species in Chile using support vector machines and neural networks," *Fisheries Research*, Vol. 102, No. 1–2, 115–122, 2010.
17. Llobet, E., et al., "Efficient feature selection for mass spectrometry based electronic nose applications," *Chemometrics and Intelligent Laboratory Systems*, Vol. 85, No. 2, 253–261, 2007.
18. Kiranyaz, S., et al., "Evolutionary artificial neural networks by multi-dimensional particle swarm optimization," *Neural Networks*, Vol. 22, No. 10, 1448–1462, 2009.
19. Lan, J., et al., "An investigation of neural network classifiers with unequal misclassification costs and group sizes," *Decision Support Systems*, Vol. 48, No. 4, 582–591, 2010.
20. Chaplot, S., L. M. Patnaik, and N. R. Jagannathan, "Classification of magnetic resonance brain images using wavelets as input to support vector machine and neural network," *Biomedical Signal Processing and Control*, Vol. 1, No. 1, 86–92, 2006.
21. El-Dahshan, E.-S. A., T. Hosny, and A.-B. M. Salem, "Hybrid intelligent techniques for MRI brain images classification," *Digital Signal Processing*, Vol. 20, No. 2, 433–441, 2010.
22. Heric, D. and D. Zazula, "Combined edge detection using wavelet transform and signal registration," *Image and Vision Computing*, Vol. 25, No. 5, 652–662, 2007.
23. Zhou, R., et al., "Mechanical equipment fault diagnosis based on redundant second generation wavelet packet transform," *Digital Signal Processing*, Vol. 20, No. 1, 276–288, 2010.
24. Friedrichs, D. A., et al., "Methodologically induced differences in oak site classifications in a homogeneous tree-ring network," *Dendrochronologia*, Vol. 27, No. 1, 21–30, 2009.
25. Pathak, N., G. K. Mahanti, S. K. Singh, J. K. Mishra, and A. Chakraborty, "Synthesis of thinned planar circular array

antennas using modified particle swarm optimization," *Progress In Electromagnetics Research Letters*, Vol. 12, 87–97, 2009.

26. Zhang, Y. and L. Wu, "Weights optimization of neural network via improved BCO approach," *Progress in Electromagnetics Research*, Vol. 83, 185–198, 2008.
27. Zhang, Y., L. Wu, and G. Wei, "A new classifier for polarimetric SAR images," *Progress in Electromagnetics Research*, Vol. 94, 83–104, 2009.
28. May, R. J., H. R. Maier, and G. C. Dandy, "Data splitting for artificial neural networks using SOM-based stratified sampling," *Neural Networks*, Vol. 23, No. 2, 283–294, 2010.
29. Armand, S., et al., "Linking clinical measurements and kinematic gait patterns of toe-walking using fuzzy decision trees," *Gait & Posture*, Vol. 25, No. 3, 475–484, 2007.
30. Gdeisat, M. A., et al., "Spatial and temporal carrier fringe pattern demodulation using the one-dimensional continuous wavelet transform: Recent progress, challenges, and suggested developments," *Optics and Lasers in Engineering,* Vol. 47, No. 12, 1348–1361, 2009.
31. Ludwig, Jr, O., et al., "Applications of information theory, genetic algorithms, and neural models to predict oil flow," *Communications in Nonlinear Science and Numerical Simulation*, Vol. 14, No. 7, 2870–2885, 2009.
32. Kellegöz, T., B. Toklu, and J. Wilson, "Elite guided steady-state genetic algorithm for minimizing total tardiness in flowshops," *Computers & Industrial Engineering*, Vol. 58, No. 2, 300–306, 2010.
33. Acharjee, P. and S. K. Goswami, "Expert algorithm based on adaptive particle swarm optimization for power flow analysis," *Expert Systems with Applications*, Vol. 36, No. 3, Part 1, 5151–5156, 2009.