Jurnal
Teknologi

Full Paper

Room Material Identification System From Photo Images Using GLCM, Modified Zernike Moments, and PSO-BP Application

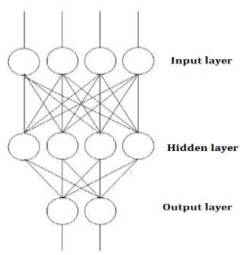
Article history
Received
16 February 2016
Received in revised form
7 June 2016
Accepted
15 September 2016

Fathin Liyana Zainudin[a], Abd Kadir Mahamad[a*], Sharifah Saon[a], Musli Nizam Yahya[b]

[a]Embedded Computing System (EmbCoS) Research Focus Group, Faculty of Electrical & Electronics, Universiti Tun Hussein Onn Malaysia, 86400 Parit Raja, Johor, Malaysia
[b]Faculty of Mechanical & Manufacturing Engineering, Universiti Tun Hussein Onn Malaysia, 86400 Parit Raja, Johor, Malaysia

*Corresponding author
kadir@uthm.edu.my

Graphical abstract



Abstract

In acoustic engineering, the types of material used in a room are basically one of the fundamental features that are essential in some of room acoustic parameters computation. This paper proposed an improved system to identify room material type from its surface photographic image. Data images of several room surfaces were collected for the system input. This improved system implements Gray Level Co-occurrence Matrix (GLCM) and modified Zernike moments for image extraction and hybrid Particle Swarm Optimization and back-propagation (PSO-BP) algorithm for classification. For comparison purpose, experiments using variations combination of GLCM and modified Zernike moments extraction as well as Levenberg-Marquardt, back-propagation neural network (BPNN), and PSO-BP algorithm were executed. By applying the proposed methods, the system accuracy increased around 30% compared to previous research. Moreover, the convergence attained during training was three times faster compared to BP algorithm. Thus using the new methods in identifying material surface images had positively improved the system in becoming more efficient and reliable.

Keywords: Image processing, GLCM, Zernike moments, neural network, PSO-BP

Abstrak

Dalam kejuruteraan akustik, jenis bahan yang digunakan di dalam sebuah bilik merupakan satu ciri yang asas yang amat penting dalam pengiraan beberapa parameter kepada akustik bilik. Artikel ini mencadangkan penambahbaikan pada sistem untuk mengenalpasti jenis bahan yang digunakan melalui imej fotografi pada permukaan bilik. Data imej daripada beberapa buah bilik telah dikumpul dan dijadikan input. Sistem yang ditambahbaik ini mengimplimentasikan *Gray Level Co-occurrence Matrix* (GLCM) dan *modified Zernike moments* untuk pemprosesan imej dan algoritma gabungan antara *Particle Swarm Optimization* dan *back-propagation* (PSO-BP) untuk tujuan klasifikasi. Sebagai pembandingan, eksperimen-eksperimen yang menggunakan gabungan variasi GLCM dan *modified Zernike moments* bersama dengan algoritma Levenberg-Marquardt, *back-propagation neural network* (BPNN), dan PSO-BP telah dijalankan. Dengan mengaplikasikan gabungan kaedah yang dicadangkan, ketepatan pada sistem meningkat 30% berbanding kajian terdahulu. Tambahan lagi, waktu penumpuan dapat dicapai tiga kali lebih pantas sewaktu latihan pada rangkaian neural berbanding dengan apabila menggunakan algoritma BP. Secara keseluruhannya, sistem yang diperbaiki ini berjaya meningkatkan kecekapan dan kebolehharapan sistem.

Kata kunci: Pemprosesan imej, GLCM, Zernike, moments, rangkaian neural, PSO-BP

1.0 INTRODUCTION

Currently, neural network implementation in predicting acoustic parameters is not foreign. Nannariello and Fricke were the first to propose implementation of neural network in predicting acoustic parameters. In 1999, they had done a research to predict reverberation time by using existing measurements of several buildings as the dataset [1]. They found out that neural network analysis may have a potential application in predicting the room reverberation time. Nannariello and Fricke also did another research in 2001 to predict other acoustic parameters; the strength factor (G values), clarity factor, and the lateral energy fraction in concert halls by using neural network analysis [2], [3]. Other researchers continued to explore the capability of using neural network in predicting acoustic properties in rooms [4], [5].

Type of room material has been proven to be one of the most important features in room acoustic parameters computation especially in determining the room reverberation time [6]. Different quantities of sound energy are absorbed by different types of materials at different frequencies. Materials that were chosen in the room can make a big difference in the room's acoustic properties.

In 2011, a research on the room material identification system from the surface image was done using GLCM for image extraction with 4 Haralicks features as its input and image classification by using Levenberg-Marquardt algorithm [7]. Although the reseach produced an adequate results, the research employed too many restriction and limitation in selecting the images making it impractical when working in real world scenario. Another research was executed using image extraction from GLCM using 13 Haralicks features and modified Zernike moments as the input in 2014 [8].

In order to further improve and aiming to achieve a better accuracy and a more reliable system, this paper proposed another method in identifying material surface from photographic input by using GLCM and modified Zernike moments for processing the images and PSO-BP algorithm for classification. Here, a standard back-propagation neural network is furthermore being enhanced by fusing it with an optimization algorithm that was proven to have a better global search for large data [9].

The system is targeted at engineers and designers to enable them to identify precisely the room material type without the need of having any physical contact or measurement, thus the room material can be identified anywhere as long as the photographic image of the material surface are provided.

2.0 METHODOLOGY

2.1 Image Processing

Gray Level Co-occurrence Matrix (GLCM) is a matrix that represents the angular spatial relationship and distance of neighbouring pixels of the grayscale intensity of an image. First, the raw image has to be converted into grayscale image as GLCM work built on the different level of intensities of the pixels (shades of gray) in an image. The matrix will compute centered on the relation of gray level between neighbouring pixels. Four neighbouring angles (θ) can be computed; 0° [0 0], 45° [0 *d*], 90° [-*d* 0], and 135° [-*d d*] with *d* is the distance between pixels.

Haralick is one of the earlier pioneers in which he presented the statistical approach based on second order joint probability distribution and managed to derive different features from gray level co-occurrence matrix (GLCM) [10]. Figure 1 shows the example for construction of GLCM [0 1] and GLCM [-1 0].

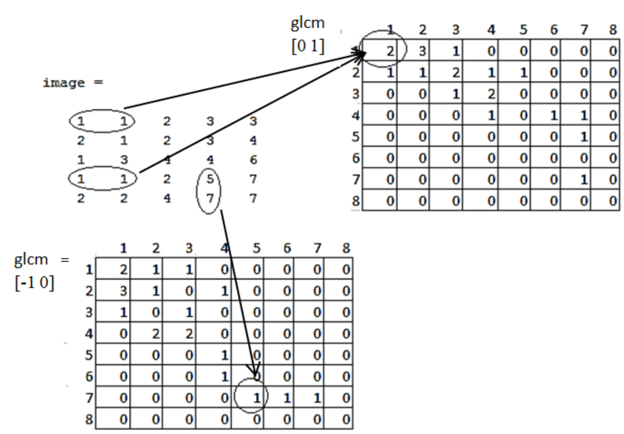


Figure 1 GLCM for θ = 0° and θ = 90° with d=1

GLCM features alone were not enough as shown by Musli *et al*. in 2011 as the features dependency extensively on image brightness. Due to having different lighting for different rooms, the captured images produce variations of brightness between images. Considering the above reason, therefore, in this research, features extracted from modified Zernike moments were added to the GLCM features to overcome this problem.

Zernike moments is a representation of an image to a series of polynomials that are orthogonal to each other or more known as Zernike polynomials [11]. Hence it is able to represent the image properties without the moments getting overlap with each other. Zernike moments is also preferable for shape identification since it can distinguish the characteristics of an object as the magnitudes are independent of the object rotation angle.

In this modified Zernike moments, for texture identification specifically where the image consists of patterns are distributed over the image, thus making it impossible to distinguish a specific shape out of the texture image, the photo image is first transformed using Discrete Fourier Transform (DFT). The DFT magnitude image then will give a distinct shape for each different surface that will be used by the Zernike moments for the image features extraction. Figure 2 shows the block diagram of the modified Zernike moments.

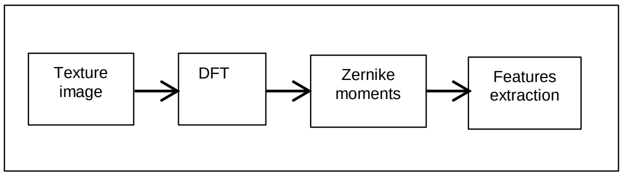


Figure 2 Modified Zernike moments

Equation (1) shows the DFT transformation for $F(k_1,k_2)$ from $f(n_1,n_2)$ signal with $N_1 x N_2$ image size.

$$F(k_1,k_2)=\sum_{n_1=0}^{N_1-1}\sum_{n_2=0}^{N_2-1} f(n_1,n_2)e^{-j\frac{2\pi}{N_1}k_1n_1}e^{-j\frac{2\pi}{N_2}k_2n_2} \quad (1)$$

where $f(n_1,n_2)$ is the image with size of $N_1 x N_2$ and $0 \leq k_1 < N_1,\ 0 \leq k_2 < N_2$.

(a) Actual image (b) DFT magnitude of the actual image (c) DFT phase of the actual image

Figure 3 DFT image transformation

The DFT will produce the magnitude and phase as in Figure 3. The DFT magnitude of the image is then normalized before it is extracted using Zernike moments as in equation (2).

$$DMA_{nm}=\frac{n+1}{\pi}\sum_{k_1=0}\sum_{k_2=0}\log|F(k_1,k_2)|^2\frac{V^*{}_{nm}(\sqrt{\rho},\theta)}{2\rho}\rho d\rho \quad (2)$$

where DMA_{nm} is the discrete modified Zernike moments for the (n,m) basis function that are computed from the normalized power spectrum of an input signal $|FN(\rho,\theta)|^2$, and * is the complex conjugate. The magnitude of the discrete modified Zernike moment, ZM_i is shown in equation (3).

$$ZM_i = |DMA_{nm}| \quad (3)$$

where $n \geq 0$, $n \geq m$, and $n-|m|$ is even, and i represents the number of magnitude.

2.2. Image Classification

2.2.1 *Back-propagation Neural Network (BPNN)*

In 1986, Rumelhart *et al*. developed BPNN algorithm as an answer to the multi-layer perceptron training problem. The most significant improvement regarding BPNN algorithm was the presence of the differential transfer function at each node as well as applying the back-propagation error to modify the weights of the internal network. This process is repeated after each training epoch. Figure 4 shows the architecture of a BPNN algorithm model. BPNN algorithm computes the squared error of the neural network as the back-propagation error, E as in equation (4) with t=actual output and y=output from neural network.

$$E=\sum|t-y|^2 \quad (4)$$

The actual value of the previous expression depends on the weights of the network. BPNN algorithm updates the weights by shifting them alongside the gradient descending direction [12] as in equation (5).

$$\Delta w = -\alpha\nabla E \quad (5)$$

where α is the learning rate in which influences the learning pace.

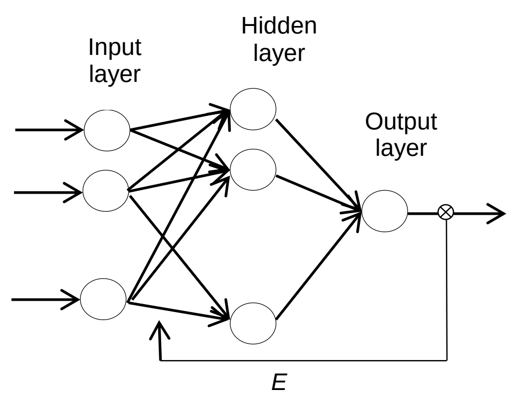


Figure 4 Back-Propagation Neural Network model architecture

2.2.2 *Particle Swarm Optimization (PSO)*

Particle Swarm Optimization (PSO) algorithm was first suggested by Kennedy and Eberhart in 1995 where they modeled the algorithm after the behavior of birds in a flock [13]. It has several solutions to the optimization problem that referred as swarms and particles. In the search space, each particle will go through and continuously adjust its position based on the distance between each particle best position and the swarm best particle [14].

The position of the i_{th} particle, in a d-dimensional search space, $x_i = (x_{i1}, x_{i2}, x_{i3}, \ldots x_{id})$ is determined by updating the velocity, $v_i = (v_{i1}, v_{i2}, v_{i3}, \ldots v_{id})$ as in (6) and updating the position as in (7).

$$v_{id(t+1)} = v_{id(t)} + c_1 r_1 (p_{id} - x_{id(t)}) + c_2 r_2 (p_{gd} - x_{id(t)}) \quad (6)$$

$$x_{id(t+1)} = x_{id(t)} + v_{id(t+1)} \quad (7)$$

where p_{id} is the p_{best} position, p_{gd} is the g_{best} position, c_1 and c_2 are the acceleration constants or the learning factors, r_1 and r_2 are vectors with their elements sampled from a uniform distribution.

2.2.3 PSO-BP algorithm

PSO-BP algorithm is a hybrid algorithm of PSO and BPNN. PSO algorithm has good global search ability and converges fast without too many parameters as the particles would remember their previous position [15]. BPNN algorithm has a great ability of local searching [16] but has a slow convergent speed and easily stuck to local minima. The main idea of this hybrid is to make use of the sturdy PSO global search ability as well as the BP local searching ability.

In this paper, the BPNN algorithm is infused into the PSO algorithm where the BPNN algorithm will search for local minima and the PSO algorithm to search for global minima.

2.3 Experimental Procedure

For the database, data for 5 different types of material surfaces; concrete wall, wooden wall, floor, wooden door, and ceiling as in Figure 5 were collected using a digital single-lens reflex (DSLR) camera with MICRO Nikkor 105mm 1:2.4 lens.

The images were captured from 6 different classrooms in Universiti Tun Hussein Onn Malaysia (UTHM) with 3 feet distance between the camera and the surface using autofocus mode. For this research specifically, the respective lens setting for shutter and ISO speed were 1/50 and 400. All images were set to a standard size of 4608x3072 pixels with a resolution of 300dpi.

After selecting non-blurred and undamaged images were picked making a total of 369 images selected and the data distribution is shown in Table 1. In the previous research done by Musli *et al*. in 2011, a limit was set; hence making a standard range for the input features and thus resulting in excluding more than 50% of the images, differ than this research where no limit was set.

In this paper, a few experiments were conducted; 4-input GLCM, 13-input GLCM, as well as 13-input GLCM and Zernike moments combined with LM, BPNN, and PSO-BP algorithm for data training.

Table 1 Quantity of collected images

Surface Type	Quantity
Concrete wall	90
Wooden wall	93
Floor	60
Door	65
Ceiling	61
Total	369

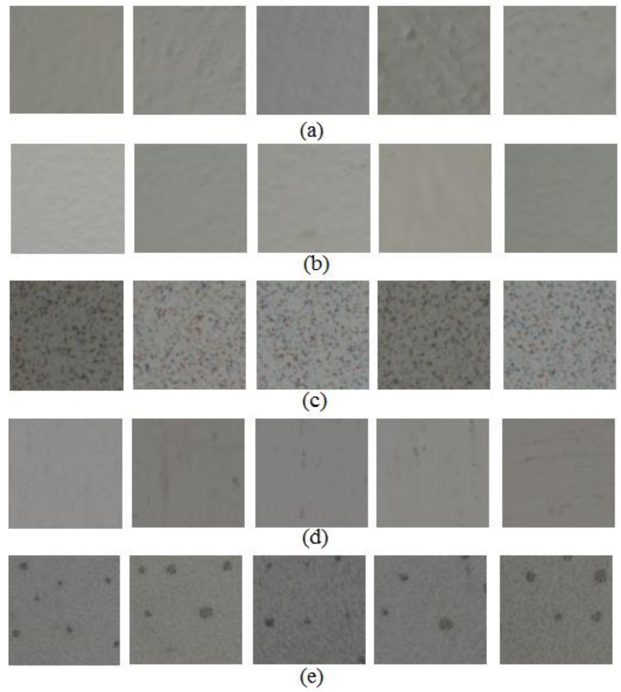


Figure 5 Sample images of each material surfaces (a) concrete wall, (b) wooden wall, (c) floor, (d) door, and (e) ceiling

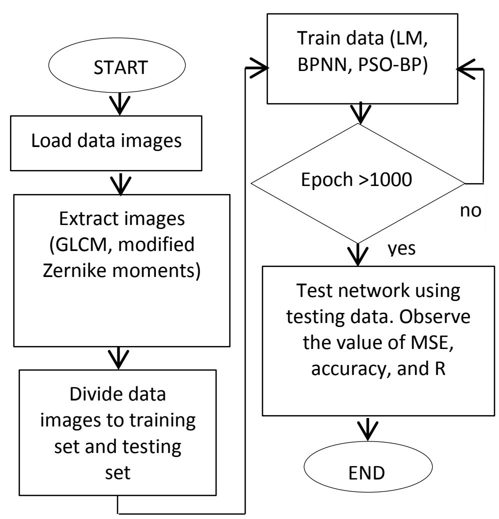


Figure 6 Experimental flow chart

Each experiment was conducted as in Figure 6. First, all images were loaded into the system before its features were extracted using GLCM and modified Zernike moments. Next, the extracted data were divided into 2 different portions. The first portion that took up 70% from the whole data reserved for training and the rest 30% are for testing. Three different neural network and optimization algorithms were conducted; LM, BPNN, and PSO-BP.

To check the reliability of the network, it will then be tested using a new set of data. Mean Squared Error (*MSE*), the regression, *R*, and the percentage of accuracy of the testing data were observed. *MSE* shows the measurement of error between the predicted and actual output while *R* measure the relationship strength between the two outputs. *R* value is important to ensure the reliability of the system.

Equation (8), (9), and (10) are the formula for *MSE*, *R*, and percentage of accuracy with e_i is the error, t_i is the desired value, y_i is the predicted value, t_{mean} and y_{mean} are the mean values, and *N* is number of data.

$$MSE = \sum_{i=1}^{N}(e_i)^2 = \sum_{i=1}^{N}(t_i - y_i)^2 \tag{8}$$

$$R = \frac{\sum_{i=1}^{N}(t_i - t_{mean})(y_i - y_{mean})}{\sqrt{\sum_{i=1}^{N}(t_i - t_{mean})^2}\sqrt{\sum_{i=1}^{N}(y_i - y_{mean})^2}} \tag{9}$$

$$acc\ (\%) = \frac{\text{Image correctly classified}}{\text{Total image}} \times 100\% \tag{10}$$

In this experiment, GLCM were calculated using distance, *d*=1 and the average value from the 4 different angles were measured. Using Haralick *et al*. features extractions, with a total of 13 textural features were extracted. Before continuing with the neural network classification step, dataset will be normalized in order to standardize the range of independent variables of data. GLCM is normalized by computing the sum of all the values in each GLCM in the array and divide each element by its sum as in (11) for a GLCM with *N* x *N* size with *i*=row, *j*=column, and *V*(*i,j*) is the value for each specified spatial relationship that occurs.

$$p(i,j) = \frac{V(i,j)}{\sum_{i,j=0}^{N-1} V(i,j)} \tag{11}$$

The features that were computed using GLCM were contrast, correlation, cluster prominence, cluster shade, dissimilarity, energy, entropy, homogeneity, autocorrelation, maximum probability, sum average, sum variance, and sum entropy.

For modified Zernike extraction, the images were transformed using equation (1) and then normalized before it can be extracted by using equation (2) and (3). The DFT magnitude of the texture image provides each texture surface a distinct shape as seen in Figure 7.

The magnitude for *n*=1,2,3,4,5,6,7,8,9, and 10 were computed and thus bring to a total of 32 different combinations of order and repetition. For a better retrieval result for geometrically transformed textures, the mean, P_{mean} using equation (12) and AC power, P_{AC} features using equation (13) were also included [17].

$$P_{mean} = \frac{\sum_{n_1}\sum_{n_2} f(n_1, n_2)}{N_1 N_2} \tag{12}$$

$$P_{AC} = \frac{\sum_{n_1}\sum_{n_2}\left(f(n_1, n_2) - P_0\right)^2}{N_1 N_2} \tag{13}$$

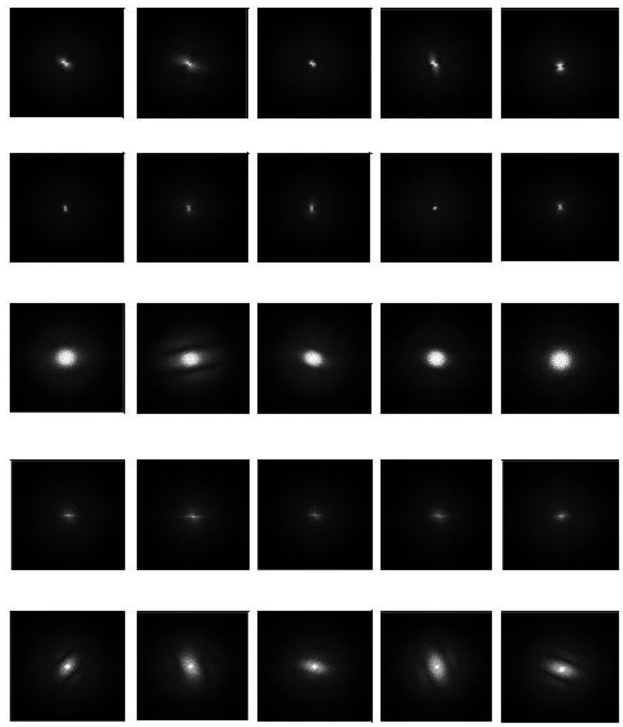


Figure 7 Sample DFT magnitude images of each material surfaces (a) concrete wall, (b) wooden wall, (c) floor, (d) door, and (e) ceiling

2.3.1 Levenberg-Marquardt Implementation

A simple Levenberg–Marquardt algorithm was tested to perform a comparison with the previous researches. This algorithm is one of the fastest FFNN algorithm available in the MATLAB toolbox although a larger memory is required than other algorithms.

For this experiment, the Levenberg–Marquardt algorithm was executed using function available in MATLAB toolbox.

A trial and error scheme was conducted from 2 to 15 hidden nodes and the best network performance is selected.

2.3.2 BPNN Algorithm Implementation

A simple back-propagation was executed using the basic concept. First the weights need to be initialized using pseudorandom numbers before the training start. The weights will be updated each time until it reached the maximum number of epoch.

Pseudocode as in Algorithm 1 for BPNN algorithm function is saved as *fun_bp.m* and will be used in PSO-BP algorithm.

Algorithm 1. Back-Propagation Neural Network

```
update weights:
for i = 1:N ;N=number of data
  for j = 1:nh ;nh=number of hidden nodes
    wij=wij - a(∂E/∂wij) ;a=learning rate and E= bp
    error
  end
end
```

2.3.3 PSO-BP aAgorithm Implementation

The idea of combining PSO algorithm and BP algorithm to create a hybrid PSO-BP algorithm is to make use of PSO algorithm ability in the global search and BPNN algorithm ability in the local search in which will produce a better result in this hybrid algorithm. PSO algorithm used mainly to search the optimal position by making the MSE from the BPNN algorithm as the fitness function in PSO algorithm. As the swarm move towards its objective (MSE), each particle will adjust its position according to its own personal best, p_{best}. The best particle in the swarm is determined, and all particles move towards this global best, g_{best} particle. The flow chart of PSO-BP method that was applied can be seen in Figure 8.

For the combination of BPNN algorithm to the PSO algorithm, the number of BPNN weights must be equal to the number of particles in PSO as the weights will represent the particles.

Algorithm 2 shows the pseudocode for PSO-BP algorithm that was applied in the system.

Algorithm 2. PSO-BP

%data training and optimization
Step 1: Match the number of BP weights with the number of particles in PSO
particles=((ni+1) * nh) + ((nh+1) * no) ;where ni=number of input, nh=number of hidden nodes, no=number of output

Step 2: Initialize position and velocity
position = rand(swarm,particles)
velocity = rand(swarm,particles)

Step 3: set maximum epoch
epoch = 0; epoch_max = 1000
while(epoch < epoch_max)

Step 4: update particles velocity and position with w is the inertia weight constant
velocity = w.*velocity + c1*r1.*(pbest - position) +c2*r2.*(gbest - position)
position =position + velocity

Step 5: update each particle point by using BP algorithm function fun_bp.m (Algorithm 1)
[PosFitness,position]= Fun_bp(position)

Step 6: update pbest
if PosFitness<PbestFitness
pbest = position;

Step 7: updating gbest
If mse pbest<mse gbest, then gbest=pbest

Step 8: Repeat step 4-7 until reached maximum epoch

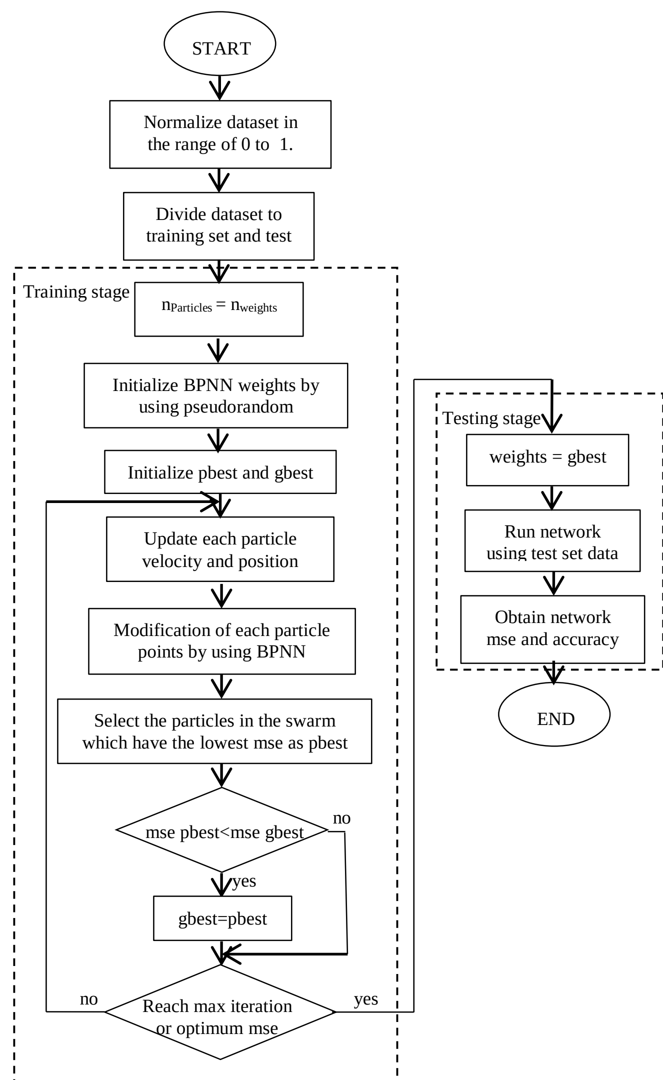


Figure 8 Flow chart of PSO-BP algorithm

3.0 RESULTS AND DISCUSSION

The result of all experiments conducted was shown in Table 2. This result was taken using test data on the neural network.

Between all 3 input types tested, the system with 13-input GLCM and modified Zernike moments extracted features produced the best accuracy, *MSE*, and *R*. The 13-input GLCM accuracy has increased around 30% from the initial 4-input GLCM. When Zernike moments features added to the system with 13-input GLCM, the system accuracy shows another significant increase around 30%. The MSE was also reduced to around 0.01. The addition of modified Zernike moments features to the input has proven its capability in increasing the system reliability and accuracy

From the graph, PSO-BP algorithm shows the highest accuracy amongst other tested algorithms. Around 4% increase in accuracy can be observed when PSO-BP algorithm was applied. The system with 13-input GLCM and Zernike moments that applied PSO-BP algorithm managed to achieve 84% of accuracy compared to 81.3% recorded for BP algorithm and 76.0% for LM algorithm as shown in Figure 9.

Figure 10 shows the comparison between different experimental inputs during training using PSO-BP algorithm. The input dataset of 13-input GLCM and modified Zernike moments showed a significant improvement of its *MSE* value as well as the system convergence speed.

Table 2 Experimental results

Input	Training Method	*MSE*	Accuracy (%)	*R*
4-input GLCM	LM	0.0694	25.2	0.3948
	BP	0.0776	22.6	0.3257
	PSO-BP	0.0706	29.4	0.4195
13-input GLCM	LM	0.0407	54.6	0.7292
	BP	0.0390	56.3	0.7228
	PSO-BP	0.0357	58.1	0.7529
13-input GLCM and modified Zernike moments	LM	0.0121	76.0	0.9200
	BP	0.0134	81.3	0.9107
	PSO-BP	0.0111	84.0	0.9279

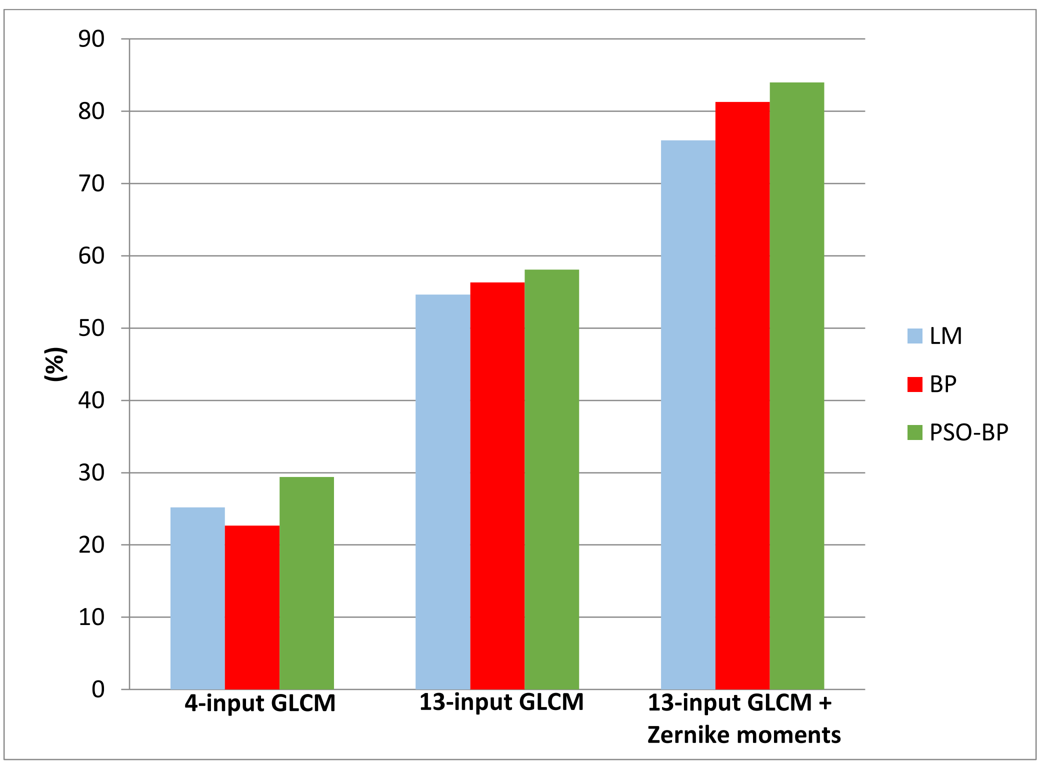


Figure 9 Percentage of accuracy for each case

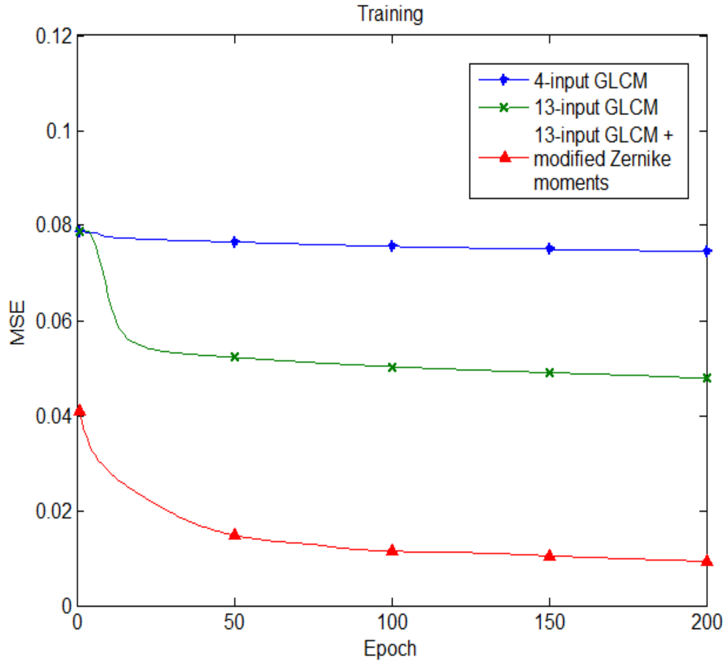


Figure 10 PSO-BP training for different experiment input dataset

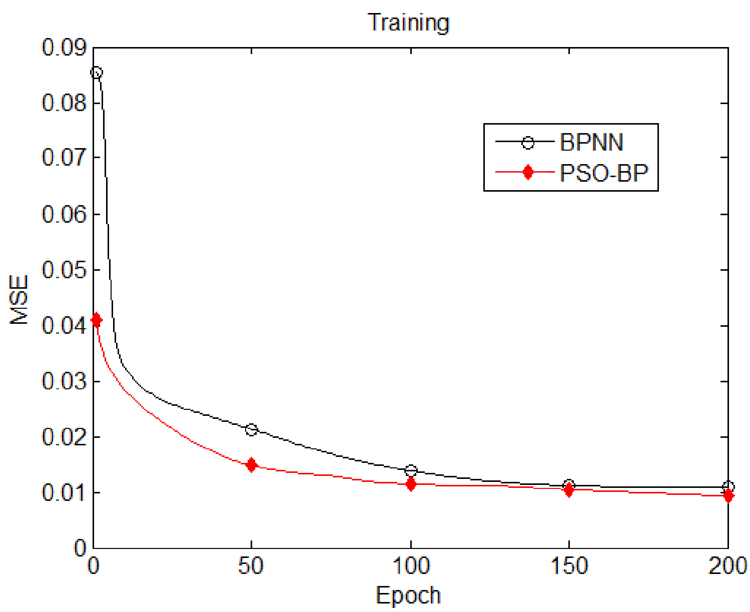


Figure 11 BPNN and PSO-BP training for 13-input GLCM and modified Zernike moments dataset

Figure 11 shows the comparison between PSO-BP and BPNN algorithm for 13-input GLCM and modified Zernike moments dataset during training. Although PSO-BP algorithm only showed a slight improvement compared to BP algorithm, the epoch taken to reach the optimum value during training has reduced significantly from 150 epoch for BP algorithm compared to only 50 epoch for PSO-BP algorithm. Therefore, a faster convergence time is obtained for PSO-BP algorithm system training.

4.0 CONCLUSIONS

This paper proposed an improved method in material identification from surface photographic input using combination of extracted image features from GLCM and modified Zernike moments as well as application of hybrid PSO-BP algorithm to classify the images. The end result shows a huge improvement from the initial 4-input GLCM and FFNN conducted by Yahya *et al*. in 2011. This research has succeeded in overcoming the problem of the previous research where the system depends too much to the image brightness making more than half of the collected images being discarded. The final system was able to produced MSE value as low as 0.0111 and the system accuracy was recorded as high as 84%. The PSO-BP algorithm also converges three times faster than BP algorithm resulting in less time taken for training. This research can be extended in predicting the material absorption coefficient and the room acoustic reverberation time directly from the photographic surface material images that were identified from the system.

Acknowledgement

The financial support received from the Research Acculturation Grant Scheme (RAGS): Vot R014, Office for Research, Innovation, Commercialization and Consultancy Management (ORICC), University Tun Hussein Onn Malaysia is gratefully acknowledged.

References

[1] Nannariello, J., and Fricke, F. 1999. The Prediction of Reverberation Time Using Neural Network Analysis. *Applied Acoustics*. 58: 305-325.

[2] Nannariello, J., and Fricke, F. 2001. The Use Of Neural Network Analysis To Predict The Acoustic Performance Of Large Rooms Part I. Predictions Of The Parameter G Utilizing Numerical Simulations. *Applied Acoustics*. 62: 917-950.

[3] Nannariello, J., and Fricke, F. 2001. The Use Of Neural Network Analysis To Predict The Acoustic Performance Of Large Rooms Part II . Predictions Of The Acoustical Attributes Of Concert Halls Utilizing Measured Data. *Applied Acoustics*. 62: 951-977.

[4] Husssain, A. J., Jameel A. J., Al-jumeily, D., and Ghazali, R. 2009. Speech Prediction Using Higher Order Neural Networks. *International Conference on Innovations in Information Technology*. United Arab Emirates. 15-17 December 2009. 294-298.

[5] Aliabadi, M., Golmohammadi, R., Ohadi, A., Mansoorizadeh, Z., Khotanlou, H., and Sarrafzadeh, M.S. 2014. Development of an Empirical Acoustic Model for Predicting Reverberation Time in Typical Industrial Workrooms Using Artificial Neural Network. *Acta Acustica united with Acustica*. 100: 1090-1097.

[6] Neubauer, R., and Kostek, B. 2001. Prediction Of The Reverberation Time In Rectangular Rooms With Non-Uniformly Distributed Sound Absorption. *Archives of Acoustics*. 26: 183-201.

[7] Yahya, M. N., Otsuru, T., Tomiku, R., and Okozono, T. 2010. Investigation the Capability of Neural Network in Predicting Reverberation Time on Classroom. *International Journal of Sustainable Construction Engineering and Technology*. 1: 19-31.

[8] Zainudin, F. L., Mahamad, A. K., Saon S., and Yahya, M. N. 2014. Comparison between GLCM and Modified Zernike Moments for Material Surfaces Identification from Photo Images. *International Conference on Computational Science and Technology (ICCST)*. Sabah, Malaysia. 27-28 August 2014. 1-4.

[9] Zhang, J.-R., Zhang, J., Lok, T.-M., and Lyu, M. R. 2007. A Hybrid Particle Swarm Optimization–Back-Propagation Algorithm For Feedforward Neural Network Training. *Applied Mathematics and Computation*. 185: 1026-1037.

[10] Haralick, R. M., Shanmugam, K., and Dinstein, I. 1973. Textural Features For Image Classification. *IEEE Transaction System*. 3: 610-621.

[11] Tahmasbi, A., Saki, F., and Shokouhi, S. B. 2011. Classification Of Benign And Malignant Masses Based On Zernike Moments. *Computers in Biology and Medicine*. 41: 726-735.

[12] Khan, K., and Sahai A. 2012. A Comparison of BA, GA, PSO, BP and LM for Training Feed forward Neural Networks in e-Learning Context. *International Journal of Intelligent Systems and Applications*. 4: 23-29.

[13] Kennedy, J., and Eberhart, R. 1995. Particle Swarm Optimization. *Neural Networks*. 1942-1948.

[14] Omran, A. S. M., and Engelbrecht, A. P. 2005. Particle Swarm Optimization Method for Image Clustering. *International Journal of Pattern Recognition and Artificial Intelligence*. 19: 297-321.

[15] Li, Z., Liu, X., and Duan, X. 2010. Comparative Research on Particle Swarm Optimization and Genetic Algorithm. *Computer and Information Science*. 3: 1942-1948.

[16] Han, F., Gu, T., and Ju, S. 2011. An Improved Hybrid Algorithm Based on PSO and BP for Feedforward Neural Networks. *International Journal of Digital Content Technology and its Applications*. 5: 106-115.

[17] Sim, D. G., Kim, H. K., and Park, R. H. 2004. Invariant Texture Retrieval using Modified Zernike Moments. *Image and Vision Computing*. 22: 331-342.