Integration of Swarm Intelligence and Artificial Neural Network for Medical Image Recognition

Dr. Hanan A. R. Akkar
Electrical Engineering Department,University of Technology /Baghdad
Email: Dr_hananuot@yahoo.com
Samera Shams Hussain
Iraqi Commission for Computers and Informatics/Baghdad

ABSTRACT

Neural network technology plays an important role in the development of new medical diagnostic assistance or what is known as "computer aided" that based on image recognition.Thispaper study the method used integration of back propagation neural network and Particle Swarm Optimizing (PSO) in parts of recognition the X-Ray of lungs for two disease cases (cancer and TB) along with the normal case. The experiments show that the improvement of algorithms for recognition side has achieved a good result reached to 88.398% for input image size 1024 pixel and 500 population size. The efficiency and recognition testes for training method was performed and reported in this paper.

Keywords: X-Ray Lung diagnosis, Computer aided; Medical images, Segmentation,
　　　　　Image processing, Recognition, Neural network,
　　　　Particle swarm optimization.

تكامل الشبكات العصبية الاصطناعية وذكاء السرب للتمييز الصور الطبية

الخلاصــــــة

الهجين تعد تكنولوجيا الشبكات العصبية الاصطناعية من التقنيات المهمة في تطور حديث مساعدة التشخيص الطبي او ما يعرف بـ" التشخيص بمساعدة الحاسوب" والمعتمد على تمييز الصوره. هذا العمل يدرس امكانية استخدام طريقة تكامل شبكة الانتشار الخلفي مع استمثال عناصر السرب في اجزاء من تمييز الأشعة السينية للرئه لحالتين مرضيتين هما السرطان والتدرن الى جانب الحالة الطبيعة.

تبين التجارب بأن تحسين الخوارزميات لغرض عملية التمييز حققت نتائج جيدة وصلت الى كفاءة بمقدار 88.398% لصورة ذات حجم (1024)نقطة وحجم مجتمع (500) . تم إجراء اختبارات الكفاءة والتمييز لطرق التدريب وتم ذكرها في هذا العمل.

INTRODUCTION

Artificial neural networks (ANNs) are computational networks which attempt to simulate of the nerve cell (neurons) of the biological central nervous system. In the past years, ANNs have seen an increasingly interests in medical image processing. ANNs have been applied to clinical diagnosis or to analysis and interpretation of images and signals with encouraging results. ANNs

have a very important role in image analysis, too, being used together with processing of digital image in recognition and classification. ANNs provide a powerful tool to help doctors to analyse, model and make sense of complex clinical data across a broad range of medical applications. Most applications of ANNs to medicine are classification problems [1].Applications of ANNs in medical image processing generally falls into the following categories: Pre-processing, Image segmentation, and object detection and recognition. Image pre-processing with ANNs generally falls into one of the following two categories: image reconstruction and image restoration. For using ANNs for medical image detection and recognition, the Back Propagation (BP) NN poses most places, other ANNs, i.e. Hopfield ANN, Adaptive Resonance Theory (ART) ANN, radial basis function ANN, Probabilistic ANN, convolution ANN, and fuzzy ANN, have also found their position in medical image detection and recognition [2].Swarm Intelligence (SI) is one of the scientific fields that are closely related to natural swarms existing in nature, such as ant colonies, bee colonies, and rivers. Among the problem solving techniques inspired from nature are evolutionary computation, ANNs, time adaptive self-organizing maps, ant colony optimization, bee colony optimization, intelligent water drop, and particle swarm optimization.

One of the proposed algorithms in the field of the (SI) is the (PSO) algorithm .The PSO algorithm is an adaptive algorithm based on a social-psychological metaphor; a population of individuals (referred to as particles) adapts by returning stochastically toward previously successful regions. Particle Swarm has two primary operators: Velocity update and Position update. During each generation each particle is accelerated toward the particles previous best position and the global best position. At each iteration, a new velocity value for each particle is calculated based on its current velocity, the distance from its previous best position, and the distance from the global best position. The new velocity value is then used to calculate the next position of the particle in the search space [3].

This paper presents automated image enhancement and method for segmenting anatomy structures in chest x-ray images: lungs and its disease. Themedical context of the enhancement algorithms presented is, Area Openings, and for segmentation is, connected component algorithm. This paper is organized as follows. Section2 prepared data which is true X-Ray images for three lung cases (Cancer, TB and normal). Section3presents image processingfor X-Ray lung images, which is consist of two stages, first stage the image passes through image enhancement processing to removed noise and not useful texture (ribs, windpipe, dusts, etc.) and the second stage is to segment each part of lungs and reshaped it in one image.Section4 for building data matrix, which has been done by bringing images that processed in first stage then resized it to desired sizes, then combined it in one data matrix which represents the input layer units and one labeling matrix which represent output labeling. Training neural networkis introduced in Section5, where integration of back propagation with PSO optimization will be presented for the training neural network and producing weights.Simulations and experimental results are provided in Section6to demonstrate the efficiency, costvaries epoch, recognition program tested for detection trained and non-trained X-Ray images are provided in section6. Finally, several conclusions are included in Section7.

NN TRAINING DATA
The images used to train NN are shots of true X-Ray lung images that put on medical X-Ray viewer which was found in hospitals. The cases classified by

specialist doctors and then arranged in three folders in PC (Cancer, TB, and Normal) to be used for building program data matrix. Table (1) has shown types of X-Ray lung images and its numbers.

Table (1) Types of X-ray images and its numbers.

Type of Images	Number of Images
Cancer X-ray	147
TB X-ray	96
Normal X-ray	119
Total Number of Images	362

X-RAY IMAGE PROCESSING

This part has been used to processed X-Ray images to removed not useful texture (ribs, windpipe, dusts, etc.); this done used the morphological filtering algorithm for image enhancement type "Area Openings" and Connected-component algorithm for segmentation.

X-RAY LUNG IMAGE ENHANCEMENT METHOD

For this method the image processing passed through multiple operations. First converted the x-ray image to gray-scale then converted the gray scaled image to binary image then applying *Area Openings* filter to enhanced x-ray lung Images as shown in Figure (1).
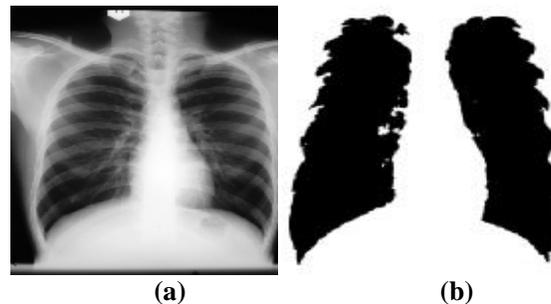
(a)   **(b)**

Figure (1) X-Ray Lung Image: (a) Applying gray-scaled. (b) Converting to binary image with applying Area opening filter.

This operation used MATLAB's (bwareaopen) function:

    Threshold = gray thresh (M);
    M =~im2bw (M, threshold);
    M = bwareaopen (M, 90000);

SEGMENTATION METHOD

After enhancement process, (in which the images convert to binary) Connected Components labelling has been applied which separated lungs to two parts left lung Image and right lung images, as shown in Figure (2). This operation used MATLAB's (regionprops) function for this.[L Ne]=bw label (M); Propied=regionprops (L,'Bounding Box');After segment operation it then recombination in one image and be saved in the lung cases folders.
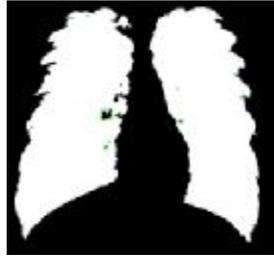
Figure (2) X-Ray Lung Image after passing all image processing.

IMAGE PROCESSINGPROGRAM

The program run and then request three folders of lung cases successively, after that the program automatically enhanced and segmented the X-Ray images and then saved it in 3 data folder (cancer, normal and TB) to be used in generation of data matrix.Figures(4, 5 and 6) have shown the image process result with three lung cases.
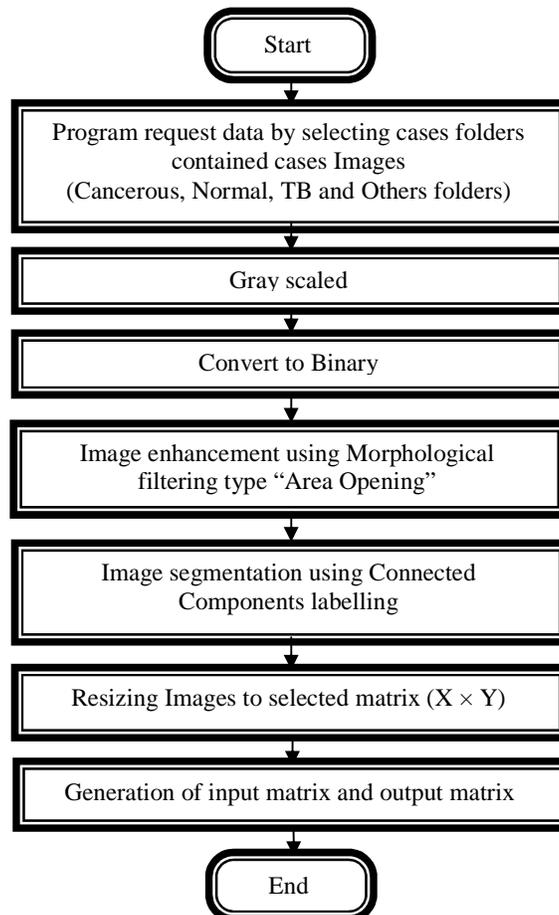
Start

Program request data by selecting cases folders
contained cases Images
(Cancerous, Normal, TB and Others folders)

Gray scaled

Convert to Binary

Image enhancement using Morphological
filtering type "Area Opening"

Image segmentation using Connected
Components labelling

Resizing Images to selected matrix (X × Y)

Generation of input matrix and output matrix

End

**Figure (3) Input/output images matrix
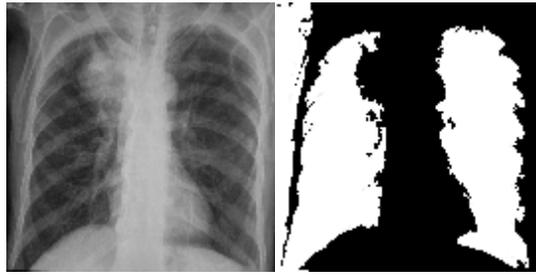Generation scheme.**

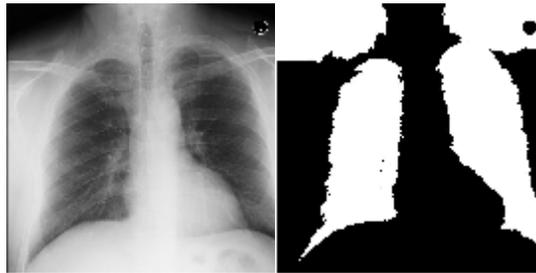**Figure (4) Image processing result for sample of
cancer X-Ray lung image.**

**Figure (5) Image processing result for sample of
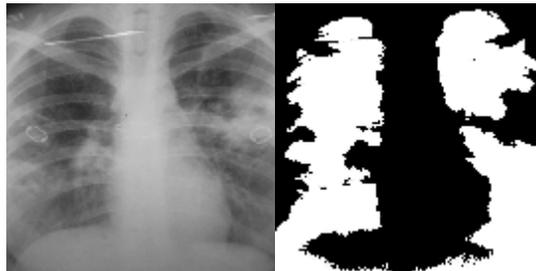normal X-Ray lung image .**

**Figure (6) Image processing result for sample of
TB X-Ray lung image.**

IMAGE MATRIX GENERATION

This part used to get data matrix of images after applying enhancement, segmentation. First stage is to run enhance -segment program to process on X-Ray images, the generated images saved in three folders depend on the input case (cancer, Normal and TB), the total time required takes about (1293 second); then run data matrix generation program that resized these images to multiple sizes. The results, computational times and calculation of neurons that needed for training (362 total x-ray images) can be classified in Table (2).

MATLAB code used for resized and generated image matrix has been:

```
Files = dir ('*.jpg');
str = strcat (files (i).name);
  M=imread(str);
  M=imresize(M,[imagex,imagey]);
size(M);
```

Table (2) Image matrix generation: size, generation time and hidden unit required.

	Resized images	**Output Matrix**			**Hidden layer units needed**
		Input layer units	Output labels	Time/sec	
1	8 × 8	64	3	94	~ 50
2	16 × 16	256	3	96	~ 170
3	32 × 32	1024	3	96	~ 700
4	64 × 64	4096	3	97	~ 2700

TRAINING NEURAL NETWORK BY INTEGRATION BACK PROPAGATION AND PSO

The PSO–BP is an optimization algorithm combining the PSO with the BP. The PSO is employed to enhance the NN training. The model consists of an input layer, hidden layer and an output layer. Inputs are taken as Images for three cases of lung disease. Weights on connections are chosen as random images initially. Then a transfer function (Sigmoid function) is applied on the weighted sum of inputs, which transfers the output to next layer which now is hidden layer. Then at all neurons of hidden layer, input value is compared to threshold value and result is compared to original one which were found, if the results do not match then back propagation algorithm is applied by which weights of previous connections are adjusted. Particle swarm optimization applied to Integrate back propagation neural network for optimizing.

VISUALIZING THE DATA

This part used to load the input data and display it on a 2 dimensional plot Figure (7) by calling the MATLAB's (display Data) function. There are 362 training X-ray Images used in Input matrix, where each training Images is an X pixel by Y pixel gray scale image of the digit. Each pixel is represented by a floating point number indicating the gray scale intensity at that location
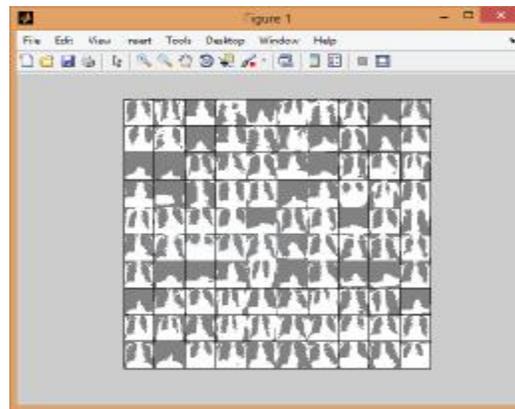

Figure (7) 2D displaying Data.

The 8×8, 16×16, 32×32 and 64×64 grid of pixels is "unrolled" respectively into a 64, 256, 1024 and 4096 dimensional vectors. Each of these training images becomes a single row in data matrix. This gives an X by Y matrix where every row is a training image for an X-Ray lung digit image.

$$X = \begin{bmatrix} (x^{(1)})^T \\ (x^{(2)})^T \\ \vdots \\ (x^{(m)})^T \end{bmatrix} \quad\quad \text{... (1)}$$

The second part of the training set is a 362-dimensional vector y that contains labels for the training set. For using three cases of lung the labels therefore are 3 labels (1, 2, and 3).

BACK PROPAGATION ALGORITHM

This part implements the back propagation algorithm. The procedure for this BP algorithm can be summarized as follows: [10].
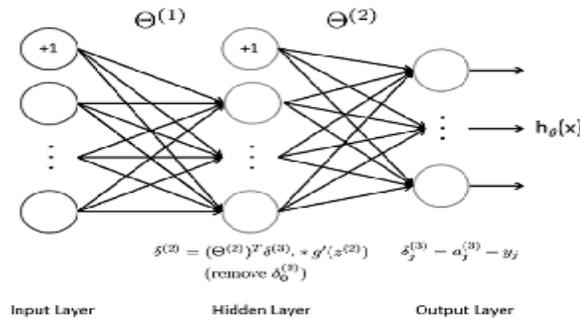

Figure (8) Back propagation Updates [10].

1. Setting the input layer's values a^2 to the *t*-th training example $x^{(t)}$. Perform a feed forward pass (Figure8), computing the activations $(z^{(2)}, a^{(2)}, z^{(3)}, a^{(3)})$ for layers 2 and 3. It needing to add $a+1$ term to ensure that the vectors of activations for layers $a^{(1)}$ and $a^{(2)}$ also include the bias unit.
    Where:

    $a_k^{(1)}$ = Unit k in layer 1 (the input layer)
    $z_k^{(2)} = a_k^{(1)} \times \Theta^{(1)}$
    $a_k^{(2)}$ = Sigmoid ($z_k^{(2)}$)
    $z_k^{(3)} = a_k^{(2)} \times \Theta^{(2)}$
    $a_k^{(3)}$ = Sigmoid ($z_k^{(3)}$)
    $h_\theta(x^{(i)}) = a_k^{(3)}$

    For each output unit k in layer 3 (the output layer), set

$$\delta_k^{(3)} = a_k^{(3)} - y_k \quad\quad \text{... (7)}$$

Where: $y_k \in \{0, 1\}$ indicates whether the current training Image matrix $x^{(t)}$ belongs to class k ($y_k = 1$), or if it belongs to a different class ($y_k = 0$).
2. For the hidden layer $l = 2$,

$$\delta^{(2)} = (\Theta^{(2)})^T \delta^{(3)} .* g'(z^{(2)}) \qquad \ldots (8)$$

3.   Accumulating gradient from using the following formula:

$$\Delta^{(l)} = \delta^{(l+1)}(a^{(l)})^T \qquad \ldots (9)$$

Training Neural Network by Integrate Back Propagation and PSO Swarm Optimization

   After training NN with back propagation algorithm, it then optimized by applying PSO swarm optimization on it. When the PSO algorithm is used in evolving weights of feed forward neural network, every particle represent a set of weights, there are three encoding strategy for every particle, the equation used for PSO optimization is shown in follows: [11?].

$$v_{id}(t+1) = w * v_{id}(t) + c_1 * rand()*[p_{id}(t) - v_{id}(t)] + c_2 * rand()*[p_{gd}(t) - x_{id}(t)] \ldots (10)$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1) \quad 1 \leq i \leq n \quad 1 \leq d \leq D$$

   Where: c_1, c_2 are the acceleration constants with positive values; rand () is a random number between 0 and 1; w is the inertia weight. In addition to the parameters c_1, and c_2 parameters, the implementation of the original algorithm also requires placing a limit on the velocity (v_{max}). After adjusting the parameters w and v_{max}, the PSO can achieve the best search ability.

   In PSO a number of simple entities the particles are placed in the search space of some problem or function, and each evaluates the objective function at its current location. Each particle then determines its movement through the search space by combining some aspect of the history of its own current and best (best-fitness) locations with those of one or more members of the swarm, with some random perturbations. The next iteration takes place after all particles have been moved. Eventually the swarm as a whole, like a flock of birds collectively foraging for food, is likely to move close to an optimum of the fitness function.

   Each individual in the particle swarm is composed of three D-dimensional vectors, where D is the dimensionality of the search space. These are the current position $\vec{x}_i$, the previous best position $\vec{p}_i$, and the velocity $\vec{v}_i$. Current position is evaluated as a problem solution. If that position is better than any that has been found so far, then the coordinates are stored in the second vector, $\vec{\mathbf{p}}_i$. The value of the best function result so far is stored in a variable that can be called pbesti (for "previous best"), for comparison on later iterations. The objective, of course, is to keep finding better positions and updating $\vec{p}_i$ and pbesti. New points are chosen by adding $\vec{\mathbf{v}}_i$ coordinates to $\vec{\mathbf{x}}_i$, and the algorithm operates by adjusting $\vec{\mathbf{v}}_i$, which can effectively be seen as a step size.

The procedure for this PSO-BP algorithm can be summarized as follows:

Step 1: Initialize a population array of particles with random positions and velocities on D-dimensions in the search space.

Step 2: loop.

Step 3: For each particle, evaluate the desired optimization fitness function in D-

variables.

Step 4: Compare particle's fitness evaluation with its *pbest$_i$*. If current value is better than pbest$_i$, then set *pbest$_i$* equal to the current value, and $\vec{p}_i$ equal to the current location $\vec{x}_i$ in D-dimensional space.

Step 5: Identify the particle in the neighborhood with the best success so far, and assign its index to the variable g.

Step 6: Change the velocity and position of the particle according to the equation (10)

Step 7: If a criterion is met (usually a sufficiently good fitness or a maximum number of iterations), exit loop.

 Step 8: end loop.

The parameter w, in the above PSO-BP algorithm also reduces gradually as the iterative generation increases, just like the PSO algorithm. The flow chart of PSO program is shown in Figure (9).

BP-PSO PROGRAM

The program run and then loaded Image matrix data (input layer and output labels), after that the program initialized weight and then applying back propagation to get weight then applying PSO to optimized it. The iteration value used for trained ANN has been set to value of (1000) to get the best performance, the value of variable between -6 and 6, R are random number , the values of c1 equal c2 = 2 and the maximum velocity v$_{max}$ to be (0.1).

The MATLAB's update particle velocity is:

```
 vel == C*(w.*vel + c1 *r1.*(localpar-par) +c2*r2.*(ones(popsize,1)*globalpar-
 par));
```

w=(maxit-iter)/maxit; r1 = rand(popsize,npar); r2 = rand(popsize,npar);

The MATLAB's update particle positions are:

```
par = par + vel;   overlimit=par<=varhi   ;underlimit=par>=varlo;
   %par=par.*overlimit+not(overlimit).*varhi;
   %par=par.*underlimit.*varlo;
pmax=max(max(par;((pmin=min(min(par));
```
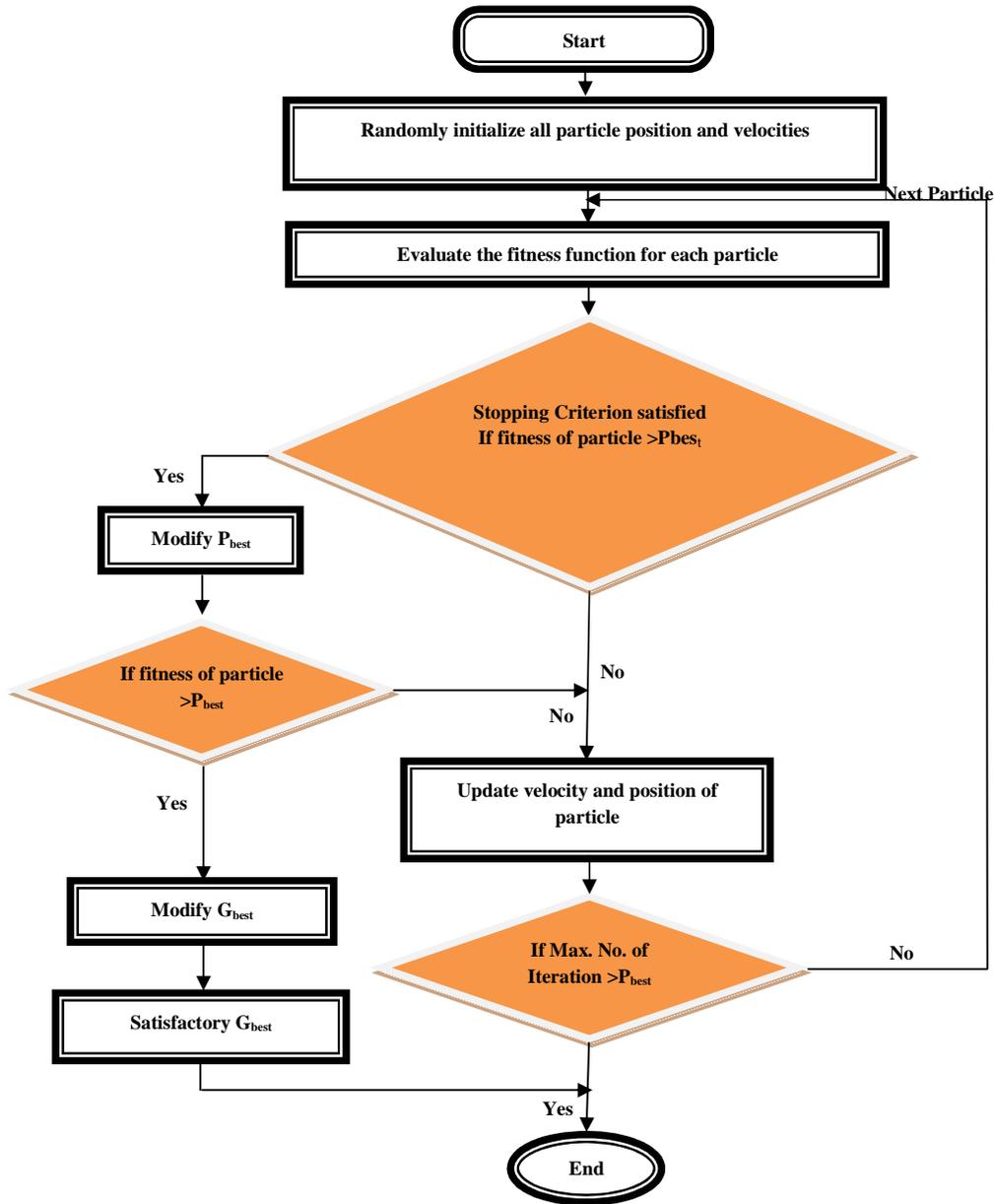
Figure (9) PSO-BP program scheme.

The tests was tested the efficiency of weights generated for different values of population size taken (100, 250 and 500). The weight efficiency, memory and computational time that have been achieved for each input have been classified in Table (3).

Table (3) Consumption of system resources and efficiency of weight generated for 1000 iteration, R1= (0.8), R2= (0.2), c1=c2=2 and v_{max} = 0.1.

| Data Matrix			Consumption of system resources			Efficiency of the weight generated
Input layers	Population size		Hidden layer units	CPU	Net Requirement of memory	Computational time / Sec	
PSO	64	100	50	44%	90MB	64	85.359 %
PSO	64	250	50	46%	100 MB	171	86.188 %
PSO	64	500	50	60%	150 MB	343	86.188 %
PSO	256	100	170	51%	130 MB	445	84.53 %
PSO	256	250	170	42%	350 MB	1126	85.912 %
PSO	256	500	170	41%	2400 MB	2240	83.149 %
PSO	1024	100	700	51%	3600 MB	6987	86.74 %
PSO	1024	250	700	50%	8400 MB	17905	87.293 %
PSO	1024	500	700	56%	17250MB	63475	88.398 %
PSO	4096	100	2700	45%	23950MB	38856	84.254 %
PSO	4096	250	2700	Error "Over System Memory"			
PSO	4096	500	2700	Error "Over System Memory"			

(a)   (b)

(c)   (d)
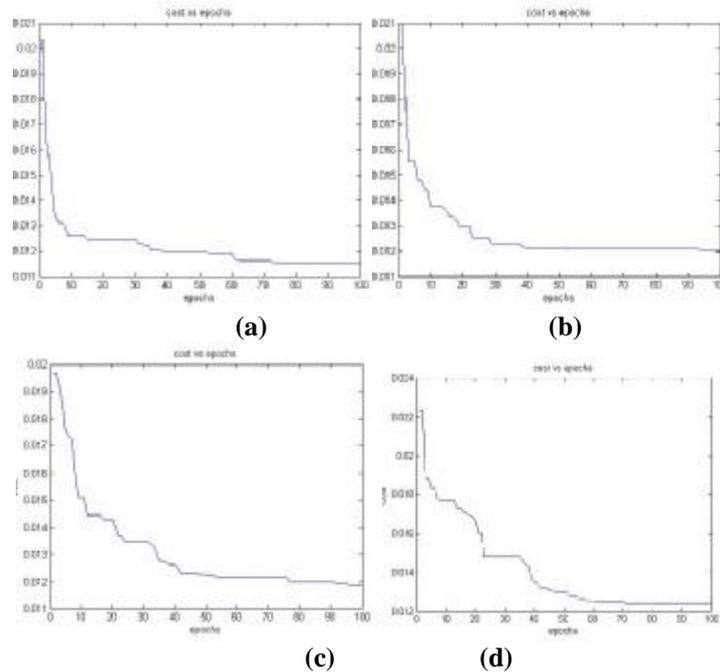
Figure (10) Cost gained with epochs of PSO with 100 population size for:a) 8x8 images matrix, b) 16x16images matrix,c) 32x32 images matrix, d) 64x64 images matrix.

X-RAY IMAGES RECOGNITION TEST (PRACTICAL TESTS)

Recognition has been tested using recognition program to identify lung case for each input image. First program has been tested on images that train used to train NN and then it been tested with images that are not trained. The program tested by used different weights that produced from two training method (back propagation and the integrated way) for same iteration value of (1000), the back propagation recognition

results and BP-PSO recognition results have been classified in Table (4) and Table (5).

Table (4) Recognition Results for trained X-Ray images.

	Type of Recognition	Image matrix size	X-Ray lung image Cases					
			Cancer (147)		TB (96)		Normal (119)	
			Detect	Failed	Detect	Failed	Detect	Failed
1	BP-PSO	8x8	122	25	84	12	108	11
2	BP-PSO	16x16	122	25	88	8	100	19
3	BP-PSO	32x32	134	13	86	10	110	9
4	BP-PSO	64x64	133	14	82	14	90	29

Table (5) Recognition Results for non-trained X-Ray images.

		Type of Recognition	Popul-ation size	X-Ray lung image Cases					
				Cancer (24)		TB (15)		Normal (22)	
				Detect	Failed	Detect	Failed	Detect	Failed
8x8	1	BP-PSO	100	16	8	2	13	22	0
	2	BP-PSO	250	13	11	11	4	22	0
	3	BP-PSO	500	14	10	14	1	21	1
16 x 16	4	BP-PSO	100	14	10	5	10	21	1
	5	BP-PSO	250	18	6	6	9	16	6
	6	BP-PSO	500	14	10	5	10	19	5
32 x 32	7	BP-PSO	100	15	9	4	11	18	4
	8	BP-PSO	250	15	9	9	6	21	1
	9	BP-PSO	500	18	6	14	1	19	5
64 x 64	10	BP-PSO	100	19	5	9	6	20	2

CONCLUSIONS

From the conducted experiments, we can get conclusions that for the following points:

- The emphasis of this paper is to develop a neural network training method used for building a program for X-Ray lung diagnosis that may help doctors in their diagnosis.
- The PSO-BP training results shown that the efficiency of weight updated increased depending on increasing of population size and number of iteration.
- The computational requirement especially memory have been increased rapidly as the input layer size increased and when the population size increased.
- The applying of image processing on X-Ray images before trained it with neural network given advantage that reduce the error and increasing the efficiency that due to the removing un useful features that may be dispersal the NN and it made be possible to reduce the input layer (resized image to small size about 8x8) without fearing from data loses.
- PSO-BP recognition results of trained and non-trained images shown relatively reducing in recognition errors as the input size and population increased, for cancer images detection (that total number 147 images), the tested have been appeared an increased in correction detection from (122 images) for 64 input size to be raised to (133 images) for 4096 input size which improved about (7.47%), and four non-trained cancer images detection (that total number 24 images), the tested have been appeared an increased in correction detection from (16 images) for 64 input size to be raised to (19 images) for 4096 input size for same population size which improved about (12.5%). For the same input size of 64 with different population sizes, the tested have been appeared an increased in correction detection of TB images from (2 images) for 100 population size to be rise to (11 images) for 250 population size and to (14 images) for 500 population sized which improved about (80 %).

REFERENCES

[1].Zhenghao Shi, and Lifeng He, "Application of Neural Networks in Medical Image Processing", Proceedings of the Second International Symposium on Networking and Network Security (ISNNS), Jinggangshan, P. R. China, 2-4, April, 2010.

[2]. Wang,Y. P. Heng, F.M. Wahl, "Image Reconstructions from Two orthogonal Projections, "International Journal of Imaging Systems and Technology.vol.13, no.2, pp.141-145, Aug. 2003.

[3].JunaSomare, SubekShakya, SudipShrestha and SujanThapa, "Breast Cancer Diagnosis System", Project Proposal, Tribhuvan University, 2012.

[4].Jeff Heaton, "Introduction to Neural Networkswith Java", Heaton Research, Inc., Second Edition, 2008.

[5].M Ilunga and D Stephenson "Infilling stream flow data using feed-forward back-propagation (BP) artificial neural networks: Application of standard BP and pseudo Mac Laurin power series BP techniques", Water SA Vol. 31 No. 2, ISSN 0378-4738, April 2005

[6]. Lawrence K.Saul, TommiJaakkola and Michael I.Jordan "Mean Field Theory for Sigmoid Belief Networks", Journal of Artificial Intelligence Research 61-76, 1996.

[7]. Gallant, S.I., "Neural Network Learning and Expert Systems", MIT Press, Cambridge, 1993.

[8].Howard Demuth, Mark Beale and Martin Hagan, "Neural Network Toolbox™ 6", The Math Works, Inc. guide, 2008.

[9].Meena M. Makary and Inas A. Yassine, "Study of Effect of Regularized Neural Network on theAccuracy of Handwriting Recognition", Systems and Biomedical Engineering Department, Faculty of Engineering, Cairo University, 2002.

[10].Ra´ul Rojas, "Neural Networks A Systematic Introduction",Springer-Verlag, Berlin, 1996.

[11]. Jing-Ru Zhang, Jun Zhang, Tat-Ming Lok and Michael R. Lyu, "A Hybrid Particle Swarm Optimization–Back-Propagation Algorithm for Feed forward Neural Network Training", Elsevier Inc., Applied Mathematics and Computation 185, 2007.