

Research Article

Cat Swarm Optimization Based Functional Link Artificial Neural Network Filter for Gaussian Noise Removal from Computed Tomography Images

M. Kumar,¹ S. K. Mishra,¹ and S. S. Sahu²

¹Department of EEE, BIT Mesra, Ranchi, India

²Department of ECE, BIT Mesra, Ranchi, India

Correspondence should be addressed to M. Kumar; manish.guptasssss007@gmail.com

Received 25 November 2015; Revised 1 April 2016; Accepted 5 April 2016

Academic Editor: Thunshun W. Liao

Copyright © 2016 M. Kumar et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Gaussian noise is one of the dominant noises, which degrades the quality of acquired Computed Tomography (CT) image data. It creates difficulties in pathological identification or diagnosis of any disease. Gaussian noise elimination is desirable to improve the clarity of a CT image for clinical, diagnostic, and postprocessing applications. This paper proposes an evolutionary nonlinear adaptive filter approach, using Cat Swarm Functional Link Artificial Neural Network (CS-FLANN) to remove the unwanted noise. The structure of the proposed filter is based on the Functional Link Artificial Neural Network (FLANN) and the Cat Swarm Optimization (CSO) is utilized for the selection of optimum weight of the neural network filter. The applied filter has been compared with the existing linear filters, like the mean filter and the adaptive Wiener filter. The performance indices, such as peak signal to noise ratio (PSNR), have been computed for the quantitative analysis of the proposed filter. The experimental evaluation established the superiority of the proposed filtering technique over existing methods.

1. Introduction

CT imaging is one of the modern noninvasive medical imaging modalities, where the X-ray beam and digital geometry processing are used to generate slices of a two-dimensional image. The X-ray beam is scanned around the body to form many views or projection. The detector senses the acquired data and store them in the computer memory for processing. After that the algorithm like “Back projection” is used for image reconstruction. The volumetric data set can be further used to construct 3D images of human organs. CT imaging can provide quality visual information regarding human tissues at a particular time. Presently, CT imaging is becoming an indispensable part of the medical treatment process [1, 2]. CT images can be distorted at any stages, such as during acquisition, processing, compression, storage, transmission, and reproduction. It may result in a

degradation of the medical image quality, which restricts the further applications of the output images [3].

CT images can get noisy due to Gaussian noise. It creates unnecessary low contrast images and makes difficulties in the minute observation of any abnormalities or disease. The Gaussian noise consists of a high-frequency component in noisy medical images [4]. Hence, the low pass filters like the mean or moving average filter could be a solution to this problem. Unfortunately, these filters blur the crucial edge information of the medical image. The medical image denoising is image specific and depends on the characteristics of noise. The selection of a filter is a critical task for any automated system, and to make it feasible according to hardware implementation, its algorithm should be computationally cheap. So, to avoid such problems, adaptive image filters like the “Wiener” filter, Artificial Neural Networks (ANNs) filter, and so forth were introduced [5, 6]. The “Wiener” filter is

an optimal linear adaptive approach, and under the ideal circumstances, “Wiener” filters work very efficiently. In the last two decades, the ANNs have been playing a vital role in nonlinear image processing. ANNs can be used for a range of applications, such as image classification, recognition, preprocessing, and feature extraction.

In fact, in the case of a real image, noises are highly nonlinear in nature and sometimes not easy to express mathematically. Hence, in such cases, a nonlinear adaptive filter is required to solve the problem. The literature shows that ANNs can be applied in the designing of a nonlinear adaptive filtering. Initially, the supervised feed-forward ANNs filters were implemented by many researchers to develop a nonlinear image filter [7–9]. It consists of input, hidden, and output layers. The selection of the hidden layer and computational time is always an issue for the researcher. FLANN was initially developed by Pao [10] and it has expanded inputs based on some of the functional expansions like trigonometric expansion, Chebyshev expansion, power series expansion, and so forth. It can be assigned to a nonlinear task in a very efficient manner. The merit of FLANN accounts for the fact that it is simple in algorithm design and implementation. It takes less computational time than the ordinary feed-forward network like MLP and so forth [11]. So, FLANN was introduced and it can solve any nonlinear problem more efficiently than any other competitive networks. The structure of FLANN is a single layer neural network unlike the multilayer network. It is easy to design conducting less computational time [12]. Earlier, ANN and its variants training or updating of weights was based on the gradient search algorithms like Least Mean Squares (LMS), Back Propagation (BP), Recursive Least Square (RLS), and so forth. The limitation of these types of algorithms is that they can be fixed at a local optimum point and the derivative function should be continuous. Hence, to avoid the gradient insufficiency, heuristic or evolutionary computing techniques are to be employed in the training of ANNs [13–17]. The authors in [18] highlight the accurate application of CSO to solve different optimization problems in less computational time. The authors in [19] introduce an algorithm, that is, Enhanced Parallel Cat Swarm Optimization (EPCSO), to provide the routing scheme for wireless sensor network (WSN), so that the power consumption of sensors node can be minimized. Similarly, the authors in [20] applied CSO in solving multiobjective optimization problems and their benefits over other nature inspired techniques.

The principal objective of this paper is to restore a CT image from Gaussian noise by using a CS-FLANN filter. The Cat Swarm Optimization (CSO) has been introduced for the selection of optimum weight of the neural network, so that cost function, that is, $\text{error}(\cdot)$, can be minimized. The layout of the paper is as follows: Section 2 describes the way in which LMS-FLANN and CSO-FLANN filter are applied for CT image denoising. Section 3 is dealing with simulation studies and the results of the different proposed methodology. Section 4 highlights the conclusion and some future scope of the work.

2. CSO Based FLANN Filter

2.1. Cat Swarm Optimization. The CSO was first introduced by Chu and Tsai in the year 2007 to achieve better performance than the PSO [21]. The behavior of a cat in searching for its prey is modeled to solve the optimization problem by employing a new set of learning mechanisms. Interestingly, the CSO cat corresponds to a particle in PSO, and little bit difference is in its algorithm [22]. Cat activities are divided into two phases or modes, that is, the seeking phase and the tracing phase. In the seeking phase the cat observes the surrounding region and attempts to move the next position. Similarly, in the tracing phase, the cats chase some desired targets. If they find a nearest potential prey, they immediately shift themselves into the tracing phase [23, 24]. The seeking phase corresponds to a global search procedure in the optimization problem, whereas the tracing phase corresponds to a local search procedure. The parameters such as seeking memory pool (SMP), count dimension change (CDC), and mixed ratio (MR) are to be fixed. Next, the fitness values of all the candidate points are to be calculated and then assigned a similar probability to identical fitness function of the cats. Otherwise, calculate the selecting probability of each candidate point using (1). In this mode only best, maximum, and minimum fitness values are to be calculated and it can be further used in tracing mode or to solve optimization problems. In tracing mode the velocity and the position are to be assigned to each candidate. In every iteration, the fitness function values of cats are evaluated, and according to the best fitness values results will be preserved in the computer memory arrays. The algorithm is executed until the termination condition criterion is satisfied; otherwise, all the processes are recurring all over again according to the CSO algorithm; the behavior of a cat is customized as follows.

Seeking Mode. Probability is as follows:

$$P_k = \frac{FS_k - FS_b}{FS_{\max} - FS_{\min}}. \quad (1)$$

Tracing Mode. Velocity is as follows:

$$V_{k,d} = \beta \times V_{k,d} + c \times r_1 \times (x_{\text{best},d} - x_{k,d}), \quad (2)$$

where dimension is

$$d = 1, 2, \dots, M \quad (3)$$

and position is

$$x_{k,d} = x_{k,d} + V_{k,d}, \quad (4)$$

where FS , P_i are the fitness value and probability of each candidates. Similarly, β is the inertia weight, c is the acceleration constant, and r_1 is any random values between 0 and 1. $x_{\text{best},d}$, $x_{k,d}$ are the global and present positions, respectively. The algorithm of a CSO is given in Algorithm 1.

Algorithm 1. Algorithm for Cat Swarm Optimization is as follows.

Step 1. Initialization of random position of cats in M dimensional space that is, $x_{k,d}$.

Step 2. Initialization of random cat velocity as $V_{k,d}$.

Step 3. Cats are randomly picked from the population according to the mixture ratio; assign cats for seeking mode and tracing mode.

Step 4. The fitness of each cat should be calculated and accordingly the local position $x_{i,d}$ and global position $x_{best,d}$ of cat will be assigned.

Step 5. The previous global best and recent global best position are compared according to fitness function and according to that save the best one.

Step 6. For a new population the cat's position and velocity will be updated using (2) and (4).

Step 7. Check the termination condition; if satisfied, terminate the program; otherwise, repeat Steps 4–6.

2.2. CS-FLANN Filter. In the proposed work, selected nine noisy input pixels from a sliding window of size 3×3 are exponentially expanded after a randomly generated weight “ W ” is assigned to expand the input. $\text{logsig}()$ is used as an activation function to obtain the estimated output, that is, “ y .” The desired value “ d ” which is a middle pixel of the reference image is used to calculate the estimated error “ e .” Similarly, the whole process will be applied all over the same noisy image. The architecture and equations used for FLANN are as follows:

Noisy few pixels exponentially expanded as

$$T1 \begin{cases} e^{0.x1} \\ e^{1.x1} \\ e^{2.x1} \\ e^{3.x1} \\ e^{4.x1} \end{cases} \quad (5)$$

Exponentially expanded noisy pixel array

$$T = [T1, T2, T3, \dots, T8, T9]. \quad (6)$$

Estimated output

$$y(n) = \varphi * [T(n) * W^T(n)]. \quad (7)$$

Error

$$e(n) = d(n) - y(n). \quad (8)$$

LMS for weight updating of the LMS-FLANN filter

$$w_i(n+1) = w_i(n) + T \times 2 \times e(n) \times u, \quad (9)$$

where $w(n+1)$ and $w(n)$ represent the new weight and the previous weight, respectively. Similarly, obtain error is “ $e(n)$ ” and “ u ” is the learning rate of FLANN-LMS filter. In the CS-FLANN filter, to avoid the programming complexity and to reduce the training computational time, we introduced certain modification in seeking mode of proposed filter. Initially, we initialize the arbitrary sets of weights and the arrangement of weights will be as follows.

Sets of weights

$$\begin{aligned} & \text{“S”} \times \text{FLANN expanded inputs} \\ & \times \text{No. random weights.} \end{aligned} \quad (10)$$

For each set of weights the error(\cdot) will be evaluated and according to the minimum error, the single set of best weight positions will be stored. The tracing parts (12) and (13) are used for generation and updating of new weight and velocities of candidates. By comparing the fitness of previous cost function and current cost function, the global best weight is stored.

For the next iteration, the velocity and the weight position are initialized by using (12) and (13). The above process will continue for “ n ” number of times.

Equations (7) and (8) imply the following.

Fitness function:

$$e(n) = d(n) - \varphi \times [T(n) \times W^T(n)]. \quad (11)$$

New velocity:

$$v^{n+1} = \beta \times v^n + c \times r \times (G_{best}^{(n)} - w^n). \quad (12)$$

New weight position:

$$w^{n+1} = w^n + v^{n+1}, \quad (13)$$

where N and M are the numbers of rows and columns of an image. $G_{best}^{(n)}$, w^n are the global best weights and the positions of weight of the n th iteration. Similarly, β , c , and r are the inertia, acceleration constant, and the randomly generated values ranging between 0 and 1, respectively. Figure 1 shows the basic structure of the FLANN filter. The adaptive algorithms LMS and CSO are applied for the training of proposed filter. The parameter-like error is utilized in the updating weights of FLANN.

3. Simulation Studies

MATLAB software is used for simulation, verification, and testing of the proposed filter. The CT image of an ankle bone has been taken for the experimental purpose, and the format of the image is in DICOM. The size of the image in row M and column N is 501×501 . The mean and variance of Gaussian noise are 0 and 0.04, respectively. The learning rate of FLANN-LMS is 0.02 and the number of iterations is fixed

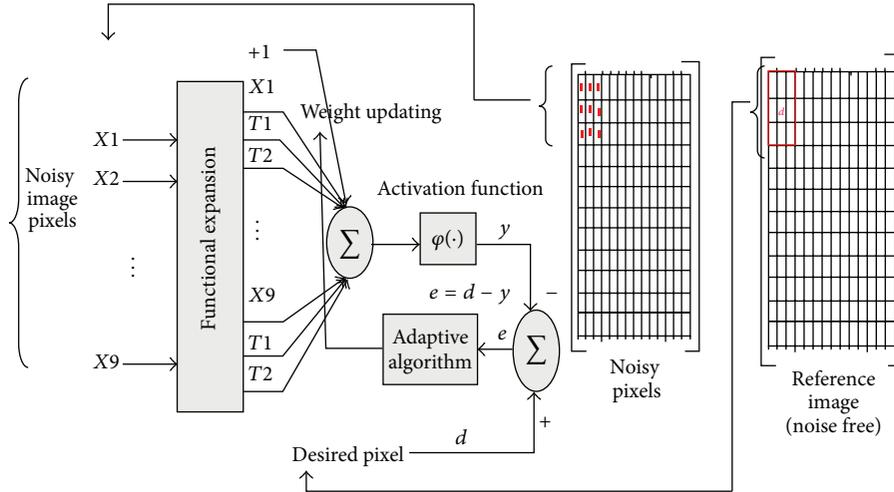


FIGURE 1: Structure of FLANN-LMS or CS-FLANN filter.

TABLE 1: Result for filters in terms of PSNR value in dB.

Images	Noisy	Mean filter	Wiener filter	FLANN-LMS	CSO-FLANN
Image 1	41.77	44.37	44.98	45.40	45.91
Image 2	41.73	44.49	45.15	45.60	46.07
Image 3	41.75	44.35	44.20	44.99	46.28
Average computational time (sec.)		1.78 s	1.86 s (testing)	321.48 s (training + testing)	7573.621 s (training + testing)

as 1000 times for both models. The number of randomly generated sets of weights for LMS-FLANN and CS-FLANN is to be 45 $\{9 \times 5\}$, 450 $\{9 \times 5 \times 10\}$. Finally, best 45 weights are selected for training the proposed ANN filter. The size of the output denoised noise is 499×499 . The mathematical formula of the range of normalization for regenerating the processed noisy image is given in (14). Similarly, the mathematical form of SNR and PSNR calculation is given in (15) and (16). The experimental result, as well as the PSNR value of different CT images, is shown in Figure 2 and Table 1, respectively. Images 2 and 3 are the CT image of the head and the format of the image is in JPG. The size of both images is taken 501×501 for denoising, and the images are present in the Appendix of this paper. The average time required for computation of Wiener filter is 1.86 sec. as only testing is considered:

$$y = \frac{x - \min \text{ pixel}}{\max \text{ pixel} - \min \text{ pixel}} \times 255. \quad (14)$$

Mean square error is

$$\text{MSE} = \frac{\sum_{m=1}^M \sum_{n=1}^N (X(m, n) - Y(m, n))^2}{M \times N}. \quad (15)$$

Peak signal to noise ratio is

$$\text{PSNR} = 10 \cdot \log_{10} \left(\frac{1}{\text{MSE}} \right). \quad (16)$$

4. Conclusion

In this paper, we have proposed an evolutionary image filter, that is, the CSO based FLANN adaptive filter, for Gaussian noise removal from CT images. The visual output of the CS-FLANN filter and PSNR values shows its superiority over the existing filters. Due to the inherent structure of CS-FLANN, the computational time for training network is higher. However, the testing time is comparable with other methodologies. Once the network is trained, then different noisy images can be tested with the same filter in less time. The proposed CS-FLANN network can be used in real-time medical image denoising and also for the elimination of a wide range of noises such as Salt and Pepper, Rician, and Speckle noise.

Appendix

See Figure 3.

Competing Interests

The authors declare that there is no conflict of interests.

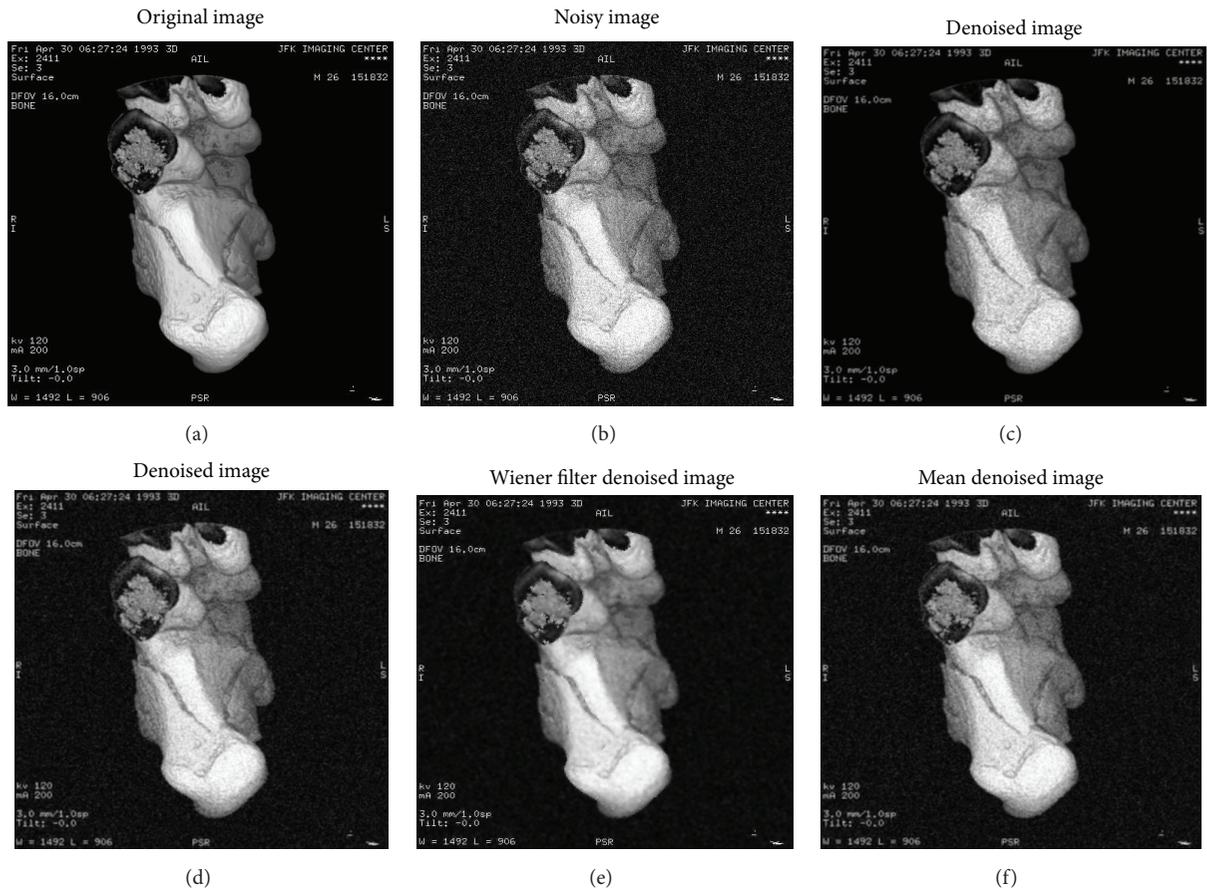


FIGURE 2: (a) CT image of the ankle bone, (b) CT image with Gaussian noise, (c) denoised image using CSO-FLANN adaptive filter, (d) denoised image using FLANN-LMS adaptive filter, (e) denoised image using Wiener filter, and (f) filtered image using a mean filter (Image Courtesy: CT-MONO2-16-ankle.dcm, MATLAB R2015a).

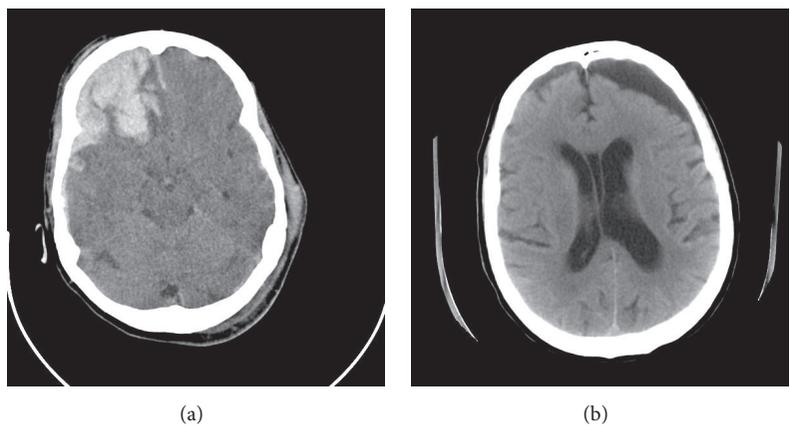


FIGURE 3: (a) Image 2: CT head cerebral contusion, and (b) Image 3: CT Head Chronic SDH (Image Courtesy: <http://lifeinthefastlane.com/resources/radiology-database/>).

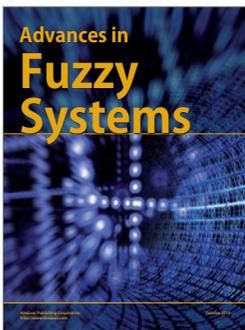
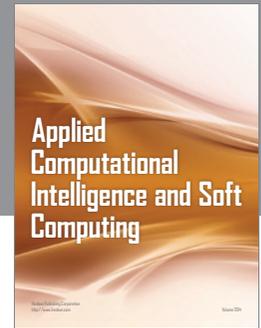
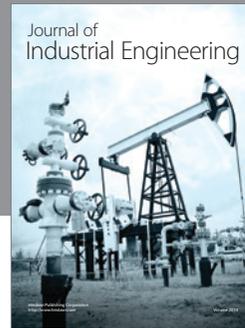
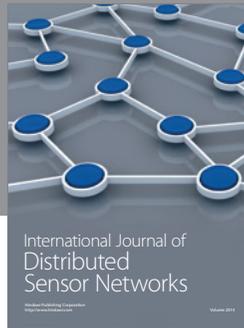
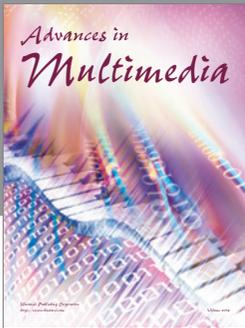
References

[1] D. Zhang and H. N. Toshi, “Medical image noise reduction using Radon transform and Walsh list in Laplacian Pyramid Domain,” in *Proceedings of the IEEE 13th International Symposium on Consumer Electronics (ISCE '09)*, pp. 756–760, Kyoto, Japan, May 2009.

[2] R. M. Rangayyan, *Biomedical Image Analysis*, CRC Press, Boca Raton, Fla, USA, 2005.

[3] Y. Zhu, Z. Mengliu, Z. Yunsong, L. Hongwei, and Z. Peng, “Noise reduction with low dose CT data based on a modified ROF model,” *Optics Express*, vol. 20, no. 16, pp. 17987–18004, 2012.

- [4] C. G. Rafael and E. W. Richard, *Digital Imaging Processing*, vol. 3, Publishing House of Electronics Industry, Beijing, China, 2nd edition, 2006.
- [5] K. Krajesk and R. Mester, "The edge preserving wiener filter for scalar and tensor valued image," in *Pattern Recognition: 28th DAGM Symposium, Berlin, Germany, September 12–14, 2006. Proceedings*, vol. 4174 of *Lecture Notes in Computer Science*, pp. 91–100, Springer, Berlin, Germany, 2006.
- [6] S. Bhattacharyya, P. Pal, and S. Bhowmick, "Binary image denoising using a quantum multilayer self organizing neural network," *Applied Soft Computing Journal*, vol. 24, pp. 717–729, 2014.
- [7] D. De Ridder, R. P. W. Duin, M. Egmont-Petersen, L. J. Van Vliet, and P. W. Verbeek, "Nonlinear image processing using artificial neural networks," *Advances in Imaging and Electron Physics*, vol. 126, pp. 351–450, 2003.
- [8] S.-S. Xiong and Z.-Y. Zhou, "Neural filtering of colored noise based on Kalman filter structure," *IEEE Transactions on Instrumentation and Measurement*, vol. 52, no. 3, pp. 742–747, 2003.
- [9] X.-P. Zhang, "Thresholding neural network for adaptive noise reduction," *IEEE Transactions on Neural Networks*, vol. 12, no. 3, pp. 567–584, 2001.
- [10] Y. H. Pao, *Adaptive Pattern Recognition and Neural Network*, Addison-Wesley, Reading, Mass, USA, 1989.
- [11] S. K. Mishra, G. Panda, S. Meher, and A. K. Sahoo, "Exponential functional link artificial neural networks for denoising of image corrupted by gaussian noise," in *Proceedings of the International Conference on Advanced Computer Control (ICACC '09)*, pp. 355–359, IEEE, Singapore, January 2009.
- [12] B. Majhi and P. K. Sa, "FLANN-based adaptive threshold selection for detection of impulsive noise in images," *International Journal Electronic Communication*, vol. 61, no. 7, pp. 478–484, 2007.
- [13] P. P. Dash and D. Patra, "Evolutionary neural network for noise cancellation in image data," *International Journal of Computational Vision and Robotics*, vol. 2, no. 3, pp. 206–217, 2011.
- [14] M. Kumar and S. K. Mishra, "Particle swarm optimization-based functional link artificial neural network for medical image denoising," in *Computational Vision and Robotics*, pp. 105–111, Springer, New Delhi, India, 2015.
- [15] M. Yaghini, M. M. Khoshraftar, and M. Fallahi, "A hybrid algorithm for artificial neural network training," *Engineering Applications of Artificial Intelligence*, vol. 26, no. 1, pp. 293–301, 2013.
- [16] S. Mirjalili, S. Z. M. Hashim, and H. M. Sardroudi, "Training feedforward neural networks using hybrid particle swarm optimization and gravitational search algorithm," *Applied Mathematics and Computation*, vol. 218, no. 22, pp. 11125–11137, 2012.
- [17] J.-R. Zhang, J. Zhang, T.-M. Lok, and M. R. Lyu, "A hybrid particle swarm optimization-back-propagation algorithm for feedforward neural network training," *Applied Mathematics and Computation*, vol. 185, no. 2, pp. 1026–1037, 2007.
- [18] P.-W. Tsai, J.-S. Pan, S.-M. Chen, and B.-Y. Liao, "Enhanced parallel cat swarm optimization based on the Taguchi method," *Expert Systems with Applications*, vol. 39, no. 7, pp. 6309–6319, 2012.
- [19] L. Kong, J.-S. Pan, P.-W. Tsai, S. Vaclav, and J.-H. Ho, "A balanced power consumption algorithm based on enhanced parallel cat swarm optimization for wireless sensor network," *International Journal of Distributed Sensor Networks*, vol. 2015, Article ID 729680, 10 pages, 2015.
- [20] P. M. Pradhan and G. Panda, "Solving multiobjective problems using cat swarm optimization," *Expert Systems with Applications*, vol. 39, no. 3, pp. 2956–2964, 2012.
- [21] S.-C. Chu and P.-W. Tsai, "Computational intelligence based on the behavior of cats," *International Journal of Innovative Computing, Information and Control*, vol. 3, no. 1, pp. 163–173, 2007.
- [22] J.-H. So and W. K. Jenkins, "Comparison of cat swarm optimization with particle swarm optimization for IIR system identification," in *Proceedings of the Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, Calif, USA, November 2013.
- [23] J. P. T. Yusiong, "Optimizing artificial neural networks using cat swarm optimization algorithm," *International Journal of Intelligent Systems and Applications*, vol. 5, no. 1, pp. 69–80, 2012.
- [24] S. Temel, N. Unaldi, and O. Kaynak, "On deployment of wireless sensors on 3-D terrains to maximize sensing coverage by utilizing cat swarm optimization with wavelet transform," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 44, no. 1, pp. 111–120, 2014.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

