

Research Article

An Adaptive Filtering Algorithm Based on Genetic Algorithm-Backpropagation Network

Kai Hu,^{1,2,3} Aiguo Song,¹ Min Xia,² XiaoLing Ye,² and YanYan Dou²

¹ Remote Measurement and Control Key Lab of Jiangsu Province, School of Instrument Science & Engineering, Southeast University, Nanjing 210096, China

² School of Information and Control, Nanjing University of Information Science & Technology, Nanjing 210044, China

³ Jiangsu Key Laboratory of Meteorological Observation and Information Processing, Nanjing University of Information Science & Technology, Nanjing 210044, China

Correspondence should be addressed to Aiguo Song; a.g.song@seu.edu.cn

Received 24 December 2012; Accepted 8 March 2013

Academic Editor: Yang Tang

Copyright © 2013 Kai Hu 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.

A new image filtering algorithm is proposed. GA-BPN algorithm uses genetic algorithm (GA) to decide weights in a back propagation neural network (BPN). It has better global optimal characteristics than traditional optimal algorithm. In this paper, we used GA-BPN to do image noise filter researching work. Firstly, this paper uses training samples to train GA-BPN as the noise detector. Then, we utilize the well-trained GA-BPN to recognize noise pixels in target image. And at last, an adaptive weighted average algorithm is used to recover noise pixels recognized by GA-BPN. Experiment data shows that this algorithm has better performance than other filters.

1. Introduction

Digital images are frequently corrupted by impulse noise during image gaining or transmission. Preservation of image details and attenuation of noise are the two important requirements of image processing, but they are contradictory in nature. So this research emphasis is on the removal of impulse noise while lessening the loss of details as low as possible.

Compared with traditional algorithms, nonlinear techniques have superiority and provide more satisfactory results while preserving image details. The most basic nonlinear filter is the standard median (SM) filter [1]. It replaces each pixel in the image by the median value of the corresponding neighborhood window centered at this pixel. The SM filter works effectively for low noise densities but at the cost of blurring the image. One solution to this problem is the weighted median (WM) filter [2] which gives more weight to some values within the window than others. It emphasizes or deemphasizes specific samples, because in most applications,

not all samples are equally important. The special case of the WM filter is the centre weighted median (CWM) [3] filter which gives more weight only to the centre value of the window. However, these filters do not perform well at higher noise densities. Besides, the filtered image is blurred with poor preservation of the image details. Each pixel of the image is considered to be noisy which may not be in practice. These filters do not detect whether a pixel in the image is actually corrupted by impulse or not and simply replace each pixel by the median value. A better tactic to avoid this drawback is to incorporate some decision making or switching action in the filtering. Firstly, it finds out whether each pixel is contaminated or not. Then, recovery method is applied on the pixel only if it is corrupted by noise. Corrupted pixels are replaced by the median values, while the noise-free pixels are left unaltered. Since not every pixel is filtered, undue distortion can be avoided as far as possible.

In recent years, there been having many researchers try to combine GA with Artificial Neural Networks (ANNs), to find an effective solution to complex problem and have

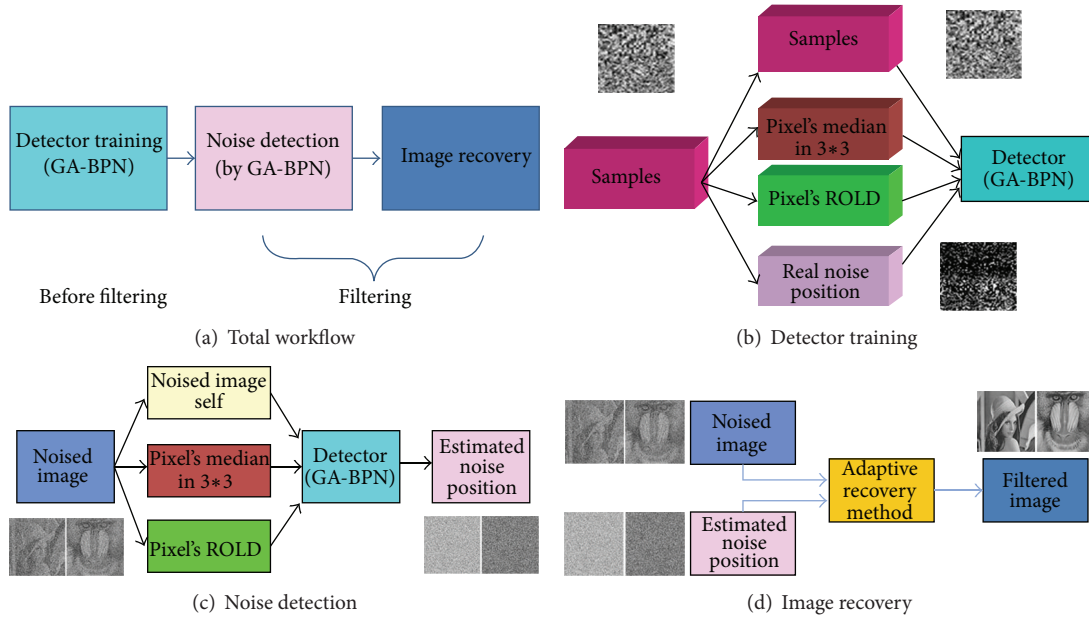


FIGURE 1: Our workflow.

a better understanding to the relations between learning and evolution, which become an active topic in the field of artificial life [3–14].

This paper proposes making use of gene algorithm to improve backpropagation neural networks to find the most suitable network connection weights and network, then form a GA-BP model and apply it to image filtering. The experiment points out that it gains good performance in image recovery.

The rest of the paper is organized as follows. Section 2 impresses our work, including workflow and theories adopted. And Section 3 presents the experiment results of the proposed algorithm. Finally, Section 4 concludes this paper.

2. Our Work

2.1. Workflow. According to the function, the proposed algorithm can be divided into 3 stages: detector training, noise detection, and image recovery. It is shown in Figure 1(a).

Figure 1(b) shows the working content of detector training. Those known noises we added have several characteristic: pixel's value, pixel's median in 3×3 window size, pixel's ROLD, pixel's position. We used value, median and ROLD as input, and use position as output, intending to train a GA-BPN as a detector. After this step, we get a well-trained detector which is capable of noise pixels position detection.

In Figure 1(c), we can see from the process how noise pixels are detected. We enter the noised image itself (target image), pixel's median in 3×3 window size, and pixel's ROLD into the detector mentioned above; then, we can estimate noised pixels position in target image.

At last, we used an adaptive weighted average algorithm which varies for windows size and distance to recover noise pixels which are recognized by GA-BPN. It is shown in Figure 1(d).

2.2. Related Theory

2.2.1. Noise Model. The classical salt and pepper noise is added to the value interval $[0, \delta] \cup [255 - \delta, 255]$, where δ is 0 or a small positive integer. The model for images with noise is described as follows [15]:

$$I_{i,j} = \begin{cases} N_{i,j} & p \\ S_{i,j} & 1 - p, \end{cases} \quad (1)$$

where $I_{i,j}$ is the image with noise, $S_{i,j}$ is the original part of $I_{i,j}$, $N_{i,j}$ is the noise part of $I_{i,j}$ (with usual value of 0 or 255), and p is the noise density.

2.2.2. Rank-Ordered Logarithmic Difference. In the literatures [15], Dong et al. proposed a new local image statistic ROLD which is based on ROAD feature [16]. It can identify more noisy pixels with less false hits and can be well applied to deal with random-valued impulse noise. Simulation results show that it outperforms a number of existing methods both visually and quantitatively. Based on this report, we adopt it in our algorithm. Its definition is shown as follows.

(1) Map the gray value of image $I_{i,j}$ to a value of $[0, 1]$ by linear transformation, that is, $u(m) \in [0, 1]$. This is the gray value of the pixel in image u at position m . The image has a window with its center on pixel $u(m)$, and its size is $(2d + 1) \times (2d + 1)$; pixel $u(n)$ is one pixel in the window. The distance between the two pixels n and m is defined as

$$D_{m,n} = 1 + \frac{\max\{\log_a |u(m) - u(n)|, -b\}}{b}, \quad n \in N_m^0(d), \quad (2)$$

where $a, b > 0$. The best distinction is achieved when $a = 2$ and $b = 5$.

(2) Sort all $D_{m,n}$ in a window in a descending order. If noise density $p > 25\%$, then ROLD is the sum of the biggest 12 values sorted under a $5 * 5$ window size. Otherwise, it is the sum of the biggest 4 values that are sorted under a $3 * 3$ window size.

For a variation between the noise pixel and its adjacent pixel in the window, the ROLD value will be a high integer. Based on this value, the ROLD of a normal pixel should be smaller for the consistency between itself and its adjacent pixel. Therefore, ROLD serves as an important definition of pixels and distinguishes between noise pixels and normal pixels.

2.2.3. GA-BPN. Neural networks are parallel-processing-structures consisting of nonlinear processing elements interconnected by fixed or variable weights [17]. These structures can be designed to generate arbitrarily complex decision regions for specific mappings, so ANNs are well suited for use as detectors and classifiers. Classic pattern recognition algorithms require assumptions concerning the underlying statistics of the environment that generated the available data. Neural networks are nonparametric and effectively address a broad class of problems. Furthermore, neural networks have an intrinsic fault tolerance. Some “neurons” may fail, but the overall network can still perform well because the information relating the mapping from input to output is distributed across all of the elements of the network.

The process of training neural networks by the implementation of a gradient-based optimization algorithm (e.g., backpropagation) will lead to locally optimal solutions which may be far removed from the global optimum. However, evolutionary optimization methods offer a procedure to stochastically search for suitable weights and bias terms given a specific network topology [18–35]. In consequence, so as to solve our problem, we can adopt GA-BPN as a noise detector.

Ye et al. designed a GA-BPN model in currency recognition and got good research result [32]. We used its steps in our algorithm. To improve BP neural networks global optimum by gene algorithm, we use all search characteristics of gene algorithm to find the most suitable network connection weights and network. Different network has different network weight numbers and thresholds. Thus, according to different networks, their best individuals can be generated by GA algorithm. Then, the fitness value of those best individual will be compared to select the best individual whose training times are fewer and fitness value is the largest to be used to build BP neural networks. By comparison, we adapted a three-layer neural network, while the number of input layer and output layer was determined by the original image samples. So our main work is to optimize hidden-layer nodes number [18–35]. These steps can be expressed as follows.

Step 1. Organize all weights and thresholds together by sequence order, and generate N chromosomes randomly.

Step 2. Design a fitness function which represents the reciprocal of the sum-squared error between network output, actual value, and desired value. Use it to calculate the fitness degree of each chromosome, and then determine whether it

accords with the optimization criteria or cycles times. If it accords with it, then go to Step 4.

Step 3. According to the fitness degree, select individuals and produce new individuals according to certain crossover probability and mutation probability. Return to Step 2.

Step 4. Save the best individual. If cycle times are less than the number of networks, change hidden layer nodes number, and then return to Step 1; otherwise, comparison will be executed among the best individuals to select the best individual.

Step 5. Split the best individual to gain the initial weights and thresholds of BPN.

Step 6. Carry forward propagation for BP neural networks, and calculate overall error, and then determine whether it accords with the request. If it is, then stop training.

Step 7. If less than cycle times, then carry backpropagation for BP neural network; adjust the weights; return to Step 5; otherwise, stop network’s training.

2.3. Our Work

2.3.1. Detector Training. Ye et al. proposed a method applying to train the networks to improve their generalization of noise detection [32]. The method created a set of primitive chessboard images in [Pt1], which has a total size of $128 * 128$ pixels and comprises many $4 * 4$ pixels grids, whereby the values of 16 inner pixels are the same and random. Add noise with a different density to [Pt1], so we get a set of noised images [Pt2]. Compare [Pt1] and [Pt2], we get [Pt3], which represent the information of noise pixels. For example, Figure 1 is one group which we used to train the network which is adapted to upper 25% noise density. Figure 2(a) (belong to [Pt1]) is an original chessboard image. Add noise with different noise density level, we get Figure 2(b) (belonging to [Pt2]); then, by comparing Figures 2(a) and 2(b), we get Figure 2(c) (belonging to [Pt3]); which represent the noise information we added.

We use [Pt2] as input and [Pt3] as output to train GA-BPN. The so-called network training is an attempt to make its output closes to [Pt3] by adjusting the inner parameters under several iterative calculations.

2.3.2. Recovery Method. Noise pixels should be replaced by estimating the original value, and better noise filtering is generally achieved when the estimated value is closer to the original value. In this paper, we used an adaptive weighted average algorithm which varies for windows size and distance. This algorithm works as follows.

- (1) Set a filter window W_f where the size is $W_s \times W_s$ (W_s is an odd value with a value of ≥ 3); its original size is $3 * 3$, and its center is on the pixel of the image. If the central point is not a noise point, shift the window W_f to make its central point be the next pixel of the image. Go to Step (2) when the central point is a noise point.

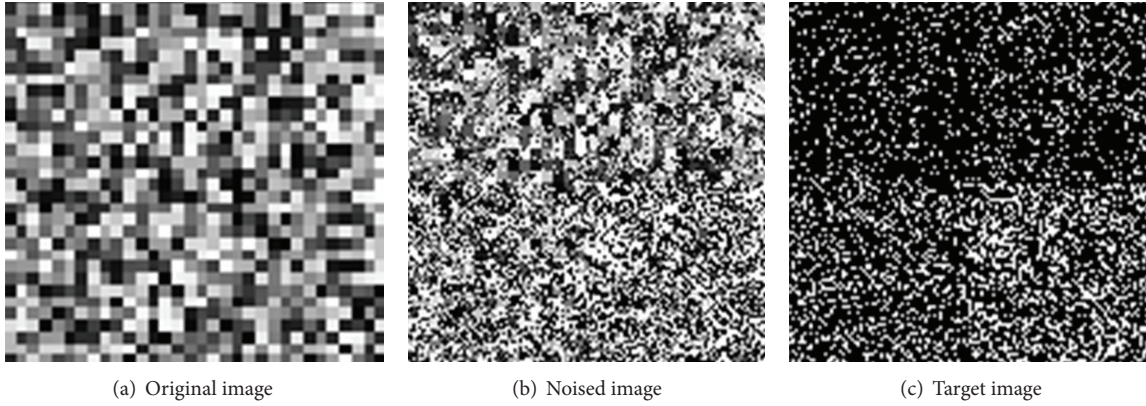


FIGURE 2: Training sample.

TABLE I: Noise detection performance test.

rd%	Lena		Baboon		Pepper		Cameraman		Barbara	
	Wn	FAR%	Wn	FAR%	Wn	FAR%	Wn	FAR%	Wn	FAR%
10	0	0.0503	0	0.0507	0	0.0502	0	0.0498	0	0.0497
20	0	0.1002	0	0.0996	0	0.1009	0	0.1018	0	0.0993
30	0	0.1492	0	0.1493	0	0.1506	0	0.1504	0	0.1498
40	0	0.1999	0	0.2004	0	0.1993	0	0.2020	0	0.2003
50	0	0.2501	0	0.2510	0	0.2502	0	0.2494	0	0.2504
60	0	0.3009	0	0.3002	0	0.3002	0	0.2957	0	0.2993
70	0	0.3506	0	0.3500	0	0.3489	0	0.3523	0	0.3508
80	0	0.3990	0	0.4004	0	0.4012	0	0.4042	0	0.3988
90	0	0.4485	0	0.4487	0	0.4521	0	0.4470	0	0.4497

Wrong detected number (Wn), false alarm ratio (FAR).

- (2) Calculate the sum M of unpolluted pixels in W_f . If $M > 9$ or $M > (W_s \times W_s + 1)/2$, go to Step (4). If not, go to Step (3).
- (3) Resize W_f , $W_s = W_s + 2$, and go back to Step (2).
- (4) Using M unpolluted pixels, calculate the output $L_{(i+s,j+t)}$ using formula (3), and replace the noise pixel. Consider the following:

$$L_{(i+s,j+t)} = \frac{\sum_{n=1}^M W_p(s,t) I_{(i,j)}^0(i+s,j+t)}{\sum_{n=1}^M W_p(s,t)}, \quad (3)$$

where $I_{(i,j)}^0(i+s,j+t)$ represents an unpolluted pixel value, with position $(i+s,j+t)$ and window center on (i,j) . $W_p(s,t)$ is the weight; a pixel is more important if it is closer to the noise point. Therefore,

$$W_p(s,t) = \frac{1}{|s| + |t|}. \quad (4)$$

3. Experiment Data

3.1. Detection Performance. The accuracy of noise detection is an important condition for recent filters. The main indicators are wrong detected number (Wn) and false alarm ratio (FAR)

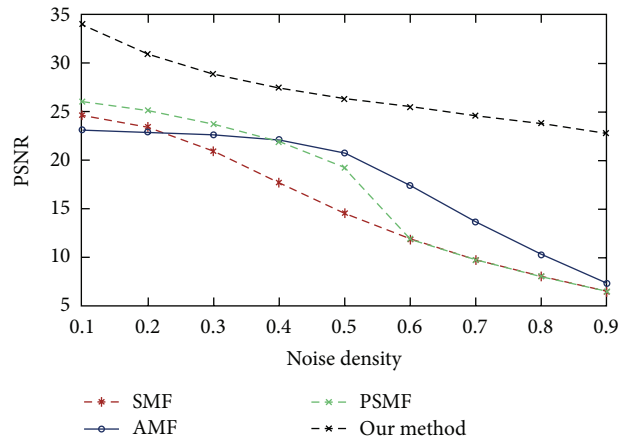


FIGURE 3: PSNR of several algorithms.

[22–26]. The performance of the proposed detector could be seen in Table I, which shows data on the images taken of Lena, baboon, peppers, cameraman, and Barbara as test images, with 10% to 90% of noise density added.

A statistical analysis revealed that the Wn and FAR of Lena, Cameraman, and Barbara are almost zero, but those of baboon and peppers are relatively large. These images, which

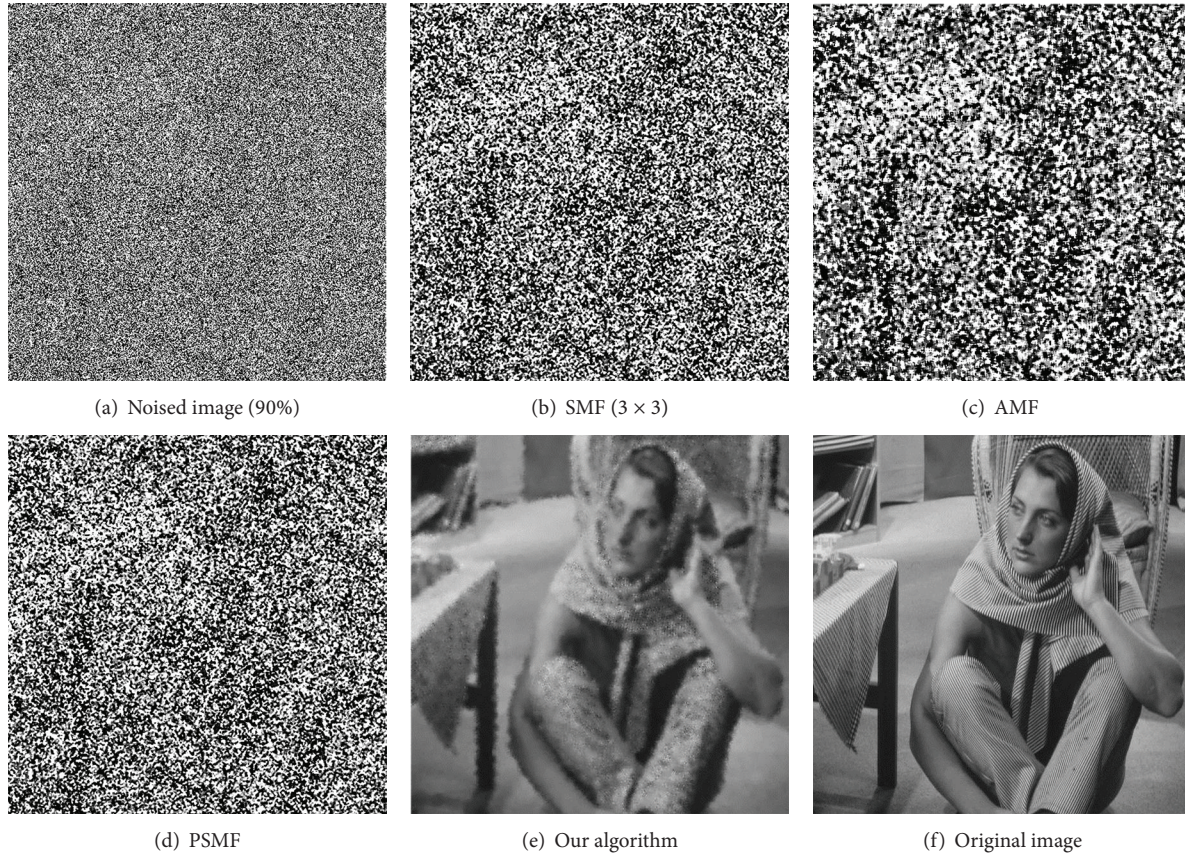


FIGURE 4: Barbara with 90% noise density.

have rich texture and detail, could still be improved; this issue will be the basis of our future study. The above indicators showed that the proposed method performs robustly in noise detection; therefore, it can be used for the next filtering experiment.

3.2. Recovery Performance. The qualitative assessment of the recovered image is done by forming a different (between the original and the recovered) image. For quantitative assessment of the restoration quality, the commonly used peak signal-to-noise ratio (PSNR) was used. Therefore,

$$\text{PSNR} = 20 \log_{10} \left(\frac{255}{\sqrt{\text{MSE}}} \right), \quad (5)$$

$$\text{MSE} = \frac{\sum_{i=1}^N \sum_{k=1}^m (x_{ik} - o_{ik})^2}{Nm},$$

where m is the total number of color components, N is the total number of image pixels, and x_{ik} and o_{ik} are the k th component of the noisy image pixel channel and its original value at pixel position “ i ”, respectively.

Experiment 1. Add noise of 10%, 50%, and 90% densities to classical images. Use standard median filter (SMF), progressive switching median filter (PSMF), and adaptive median filter (AMF), including the proposed algorithm, to filter these

noised images. Then, calculate their PSNR values, and data are shown in Table 2.

It is easy to see that, for every image, the PSNR value of the proposed algorithm is higher than those of others; indicating the universality of the proposed algorithm’s excellent performance.

In removing noise and preserving details, our algorithm showed a robust performance, which became more obvious under a higher noise density.

Experiment 2. Add 10% to 90% of noise densities to image Lena. Use several algorithms, including the proposed algorithm, to filter these noised images. Then, calculated their PSNR values, data are shown in Figure 3.

All PSNR data of the proposed method are higher than those of others, the last column titled “Improved degree” turn out the excellent improvement. It means that our algorithm’s filter performance is comprehensively better than other algorithms. The biggest improvement is observed in extremely high noise densities, at 90% noise density, it improved to 14.68 which made us surprised. The smallest improvement is obtained at 40% noise density; it raised 7.47 which also made us excities. In other different noise density, it also has an excellent improvement.

So proposed algorithm increased filter performance in all level noise densities considerably.

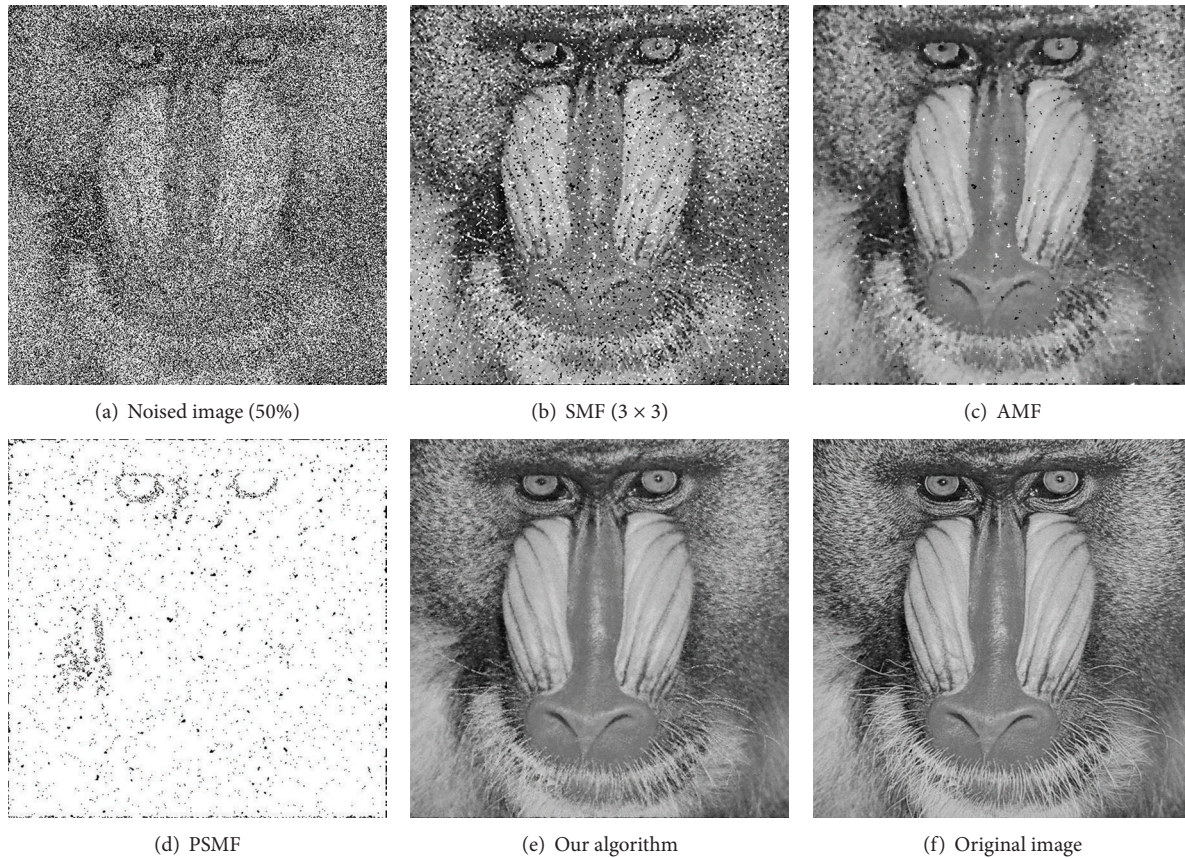


FIGURE 5: Baboon with 50% noise density.

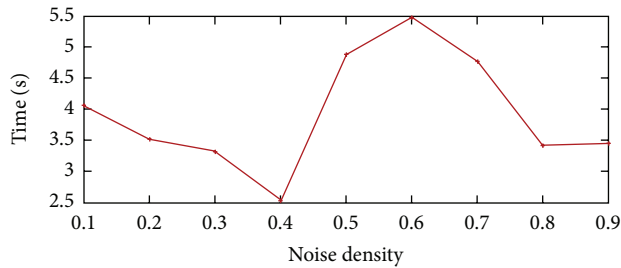


FIGURE 6: Detection time.

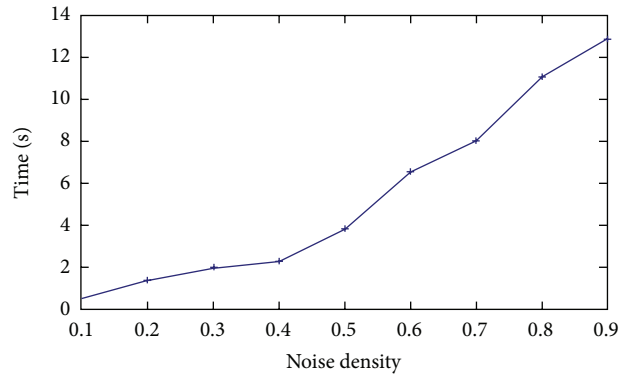


FIGURE 7: Recovery time.

Experiment 3. Add 90% noise density to “Barbara.” Use several algorithms, including the proposed algorithm, to filter these noised images, and show them in Figure 4. Compare the results subjectively; Figures 4(b), 4(c), and 4(d) are blurred seriously. In contrast, our method performs better and can suppress the noise successfully while preserving more details. It is obvious that Figure 4(e) has amazing performance in recovery compared to other traditional algorithms. This points out that our algorithm has better performance than other traditional algorithms when the normal image is highly corrupted.

Experiment 4. To further compare the capability of preserving image details [15], in Figure 5, we give the restored results

for “baboon” with rich details but corrupted by 50% impulse noise. Restore it by SMF, AMF, PSMF, and the proposed algorithm, and show results in Figure 5. It is obvious that our algorithm is better than others. The experimental results show that the approach gives good results in highly complex and detail-rich images. We see that for the other methods, there are still some noticeable noises unremoved, and there exists some loss and discontinuity of the details, such as the hair around the mouth of the baboon and the edges of the bridge. In contrast, the visual qualities of our restored images

TABLE 2: PSNR of different algorithm when the image is corrupted by different noise density.

Noised image	SMF	PSMF	AMF	Proposed	Improved
Lena					
10%	32.94	37.11	30.95	42.07	4.96
50%	15.35	20.71	24.22	32.59	8.37
90%	6.65	6.64	7.54	25.95	18.41
Baboon					
10%	22.95	24.41	20.93	32.06	7.65
50%	14.41	18.27	19.08	24.07	4.99
90%	6.72	6.69	7.56	20.20	12.64
Boat					
10%	29.52	31.89	26.96	38.22	6.33
50%	15.13	20.14	22.63	29.21	6.58
90%	6.65	6.64	7.58	23.47	15.89
Barbara					
10%	24.58	26.01	23.12	34.19	8.18
50%	14.69	19.30	20.76	26.31	5.55
90%	6.64	6.57	7.45	22.81	15.36
Peppers					
10%	33.02	36.61	29.38	40.03	3.42
50%	14.92	21.08	23.37	31.13	7.76
90%	6.32	6.3	7.23	25.16	17.93

are quite good, even with the abundance of image details and the high noise level present in the images.

3.3. Computation Time. Computation time is an important indicator when weighing an algorithm; for it reflects whether the algorithm can be practically applied or not [15]. The proposed algorithm includes two: noise detection and noise filtering. Two independent experiments were done on these. Ten experiments were done on image Lena under 10% to 90% noise densities, and average results are shown in Figures 6 and 7. It shows that time of noise filtering is much greater than time of noise detection, and it grows when noise density grows. Totally, its computation time is acceptable. The running used was Pentium IV 2.0 G and 1 G RAM, and MATLAB 7.6 was also utilized.

4. Conclusions

This paper proposes making use of GA-BPN as noise detector and ROLD feature as input, to do image filtering. From experiment data, compared with some other traditional filters, the proposed algorithm performed robustly in aspects of noise removal and detail preservation, and its merit will be more obvious under higher noise densities.

Conflict of Interests

The authors declare that they have no conflict of interests.

Authors' Contribution

All authors drafted the paper, read, and approved the final paper.

Acknowledgments

This paper was funded under a grant from the National Natural Science Foundation of P.R. China (no. 61105115, no. 61075068, and no. 61172029), a major project to cultivate science and technology innovation from the Ministry of Education of China (no. 708045), Nanjing University of Information Science and Technology Research (no. 20070063), Open Project (KDX1102) of Jiangsu Key Laboratory of Meteorological Observation and Information Processing, National Department Public Benefit Research Foundation (GYHY200806017), and Industry-academic Joint Technological Innovations Fund Project of Jiangsu Province (2012t026).

References

- [1] R. C. Gonzalez, R. E. Woods, and S. L. Eddins, *Digital Image Processing Using MATLAB*, Pearson Education India, 2004.
- [2] T. Sun and Y. Neuvo, "Detail-preserving median based filters in image processing," *Pattern Recognition Letters*, vol. 15, no. 4, pp. 341–347, 1994.
- [3] E. Abreu, M. Lightstone, S. K. Mitra, and K. Arakawa, "A new efficient approach for the removal of impulse noise from highly corrupted images," *IEEE Transactions on Image Processing*, vol. 5, no. 6, pp. 1012–1025, 1996.
- [4] C. Xing, S. Wang, and H. Deng, "A new filtering algorithm based on extremum and median value," *Journal of Image and Graphics*, vol. 6, no. 6, pp. 533–536, 2001.
- [5] Z. Wang and D. Zhang, "Progressive switching median filter for the removal of impulse noise from highly corrupted images," *IEEE Transactions on Circuits and Systems II*, vol. 46, no. 1, pp. 78–80, 1999.
- [6] R. H. Chan, C. W. Ho, and M. Nikolova, "Salt-and-pepper noise removal by median-type noise detectors and detail-preserving regularization," *IEEE Transactions on Image Processing*, vol. 14, no. 10, pp. 1479–1485, 2005.
- [7] I. V. Apalkov, P. S. Zvonarev, and V. V. Khryashchev, "Neural network adaptive switching median filter for image denoising," in *Proceedings of the International Conference on Computer as a Tool (EUROCON '05)*, pp. 959–962, November 2005.
- [8] J. Zhang, Z. Lu, L. Shi et al., "Filtering images contaminated with salt and pepper noise with pulse-coupled neural networks," *Science in China E*, vol. 34, no. 8, pp. 882–894, 2004 (Chinese).
- [9] B. Majhi, P. K. Sa, and G. K. Panda, "ANN based adaptive thresholding for impulse detection," in *Proceedings of the 3rd IASTED International Conference on Signal Processing, Pattern Recognition, and Applications (SPPRA '06)*, pp. 294–297, February 2006.
- [10] X. Min, Z. Wang, and J. Fang, "Temporal association based on dynamic depression synapses and chaotic neurons," *Neurocomputing*, vol. 74, pp. 3242–3247, 2011.
- [11] Q. Liu, H. Jin, X. Tang, H. Lu, and S. Ma, "A new extension of kernel feature and its application for visual recognition," *Neurocomputing*, vol. 71, no. 10–12, pp. 1850–1856, 2008.

- [12] T. Li and X. Ye, "Improved stability criteria of neural networks with time-varying delays: an augmented LKF approach," *Neurocomputing*, vol. 73, no. 4–6, pp. 1038–1047, 2010.
- [13] L. Weng, W. Cai, M. J. Zhang, X. H. Liao, and D. Y. Song, "Neural-memory based control of micro air vehicles (MAVs) with flapping wings," in *Proceedings of the 4th International Symposium on Neural Networks (ISNN '07)*, vol. 4491 of *Lecture Notes in Computer Science*, pp. 70–80, Nanjing, China, 2007.
- [14] J. Xu, Y. Y. Cao, D. Pi, and Y. Sun, "An estimation of the domain of attraction for recurrent neural networks with time-varying delays," *Neurocomputing*, vol. 71, no. 7–9, pp. 1566–1577, 2008.
- [15] Y. Dong, R. H. Chan, and S. Xu, "A detection statistic for random-valued impulse noise," *IEEE Transactions on Image Processing*, vol. 16, no. 4, pp. 1112–1120, 2007.
- [16] N. Z. Janah and B. Baharudin, "Mixed impulse fuzzy filter based on MAD, ROAD, and genetic algorithms," in *Proceedings of the International Conference on Soft Computing and Pattern Recognition (SoCPaR '09)*, pp. 82–87, December 2009.
- [17] D. B. Fogel, L. J. Fogel, and V. W. Porto, "Evolutionary methods for training neural networks," in *Proceedings of the IEEE Conference on Neural Networks for Ocean Engineering*, pp. 317–327, August 1991.
- [18] M. Xia, J. Fang, F. Pan, and E. Bai, "Robust sequence memory in sparsely-connected networks with controllable steady-state period," *Neurocomputing*, vol. 72, no. 13–15, pp. 3123–3130, 2009.
- [19] K. Hu, A. G. Song, W. L. Wang, Y. Zhang, and Z. Fan, "Fault detection and estimation for non-Gaussian stochastic systems with time varying delay," *Advances in Difference Equations*, vol. 2013, article 22, 2013.
- [20] Y. C. Zhang and C. Chen, "An improved collaborative filtering algorithm based on bipartite network," in *Proceedings of the 7th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD '10)*, pp. 2446–2449, August 2010.
- [21] Q. Ran, "Edge information extraction algorithm for CT cerebrovascular medical image based on neural network," *Journal of Electronic Measurement and Instrument*, vol. 4, article 010, 2010.
- [22] M. Xia, Z. Wang, and J. Fang, "Sequence memory with dynamic synapses and chaotic neurons," in *Proceedings of the International Conference on Cognitive Neurodynamics (ICCN '08)*, pp. 219–223, 2008.
- [23] C. Sun, Y. Feng, and Z. Ding, "New locally adaptive image interpolation algorithm based on edge preserving," *Chinese Journal of Scientific Instrument*, vol. 31, no. 10, pp. 2279–2284, 2010.
- [24] G. Kaliraj and S. Baskar, "An efficient approach for the removal of impulse noise from the corrupted image using neural network based impulse detector," *Image and Vision Computing*, vol. 28, no. 3, pp. 458–466, 2010.
- [25] Z. Fang and J. J. W. Peizhen, "Learning-based resolution enhancement technique for single-frame coke micrograph," *Journal of Electronic Measurement and Instrument*, vol. 7, article 005, 2011.
- [26] Y. Hanada, M. Muneyasu, and A. Asano, "An improvement of unsupervised design method for weighted median filters using GA," in *Proceedings of the International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS '09)*, pp. 204–207, December 2009.
- [27] B. Smolka, "On the adaptive impulsive noise attenuation in color images," in *Proceedings of the 3rd International Conference on Image Analysis and Recognition (ICIAR '06)*, vol. 4141 of *Lecture Notes in Computer Science*, pp. 307–317, Povoá de Varzim, Portugal, 2006.
- [28] W. Zhang, Y. Tang, J.-A. Fang, and X. Wu, "Stability of delayed neural networks with time-varying impulses," *Neural Networks*, vol. 36, pp. 59–63, 2012.
- [29] M. Xia, Y. Zhang, L. Weng, and X. Ye, "Fashion retailing forecasting based on extreme learning machine with adaptive metrics of inputs," *Knowledge-Based Systems*, vol. 36, pp. 253–259, 2012.
- [30] K. Hu et al., "Edge detection of optimal wavelet scale space image," *Journal of Nanjing University of Information Science & Technology*, vol. 3, no. 3, pp. 259–264, 2011.
- [31] B. Q. Cao and J. X. Liu, "Currency recognition modeling research based on bp neural network improved by gene algorithm," in *Proceedings of the International Conference on Computer Modeling and Simulation (ICCMS '10)*, pp. 246–250, January 2010.
- [32] X. L. Ye, L. Qian, and K. Hu, "An adaptive denoising method for salt and pepper noise detected by neural network," *Opto-Electronic Engineering*, vol. 38, no. 3, pp. 119–124, 2011 (Chinese).
- [33] F. Su, G. Fang, and N. M. Kwok, "Adaptive color feature identification in image for object tracking," *Mathematical Problems in Engineering*, vol. 2012, Article ID 509597, 18 pages, 2012.
- [34] M. Yin, W. Liu, J. Shui, and J. Wu, "Quaternion wavelet analysis and application in image denoising," *Mathematical Problems in Engineering*, vol. 2012, Article ID 493976, 21 pages, 2012.
- [35] M. Yin, W. Liu, J. Shui, and J. Wu, "Quaternion wavelet analysis and application in image denoising," *Mathematical Problems in Engineering*, vol. 2012, Article ID 493976, 10 pages, 2012.



Hindawi

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

