Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2014) 22: 637 – 649
© TÜBİTAK
doi:10.3906/elk-1208-77

Removing random-valued impulse noise in images using a neural network detector

İlke TÜRKMEN*
Department of Aircraft Electronics, Civil Aviation College, Erciyes University, Kayseri, Turkey

Received: 24.08.2012 • **Accepted:** 17.12.2012 • **Published Online:** 21.03.2014 • **Printed:** 18.04.2014

Abstract: This paper proposes a new method using an artificial neural network to remove random-valued impulse noise (RVIN) in images. The inputs of the neural model used to detect the RVIN are formed using basic and related gradient values. The detection of the noisy pixels is realized in 3 phases using the proposed neural detector. In order to obtain a more robust detector, 2 different networks, which are trained with an artificial training image corrupted with high and low clutter densities, are used. The extensive simulation results show that the proposed method is significantly better than the compared filters in terms of its image restoration and noise detection performance.

Key words: Image denoising, noise detector, random-valued impulse noise, neural networks

1. Introduction

Digital images are degraded by impulse noise (IN) during acquisition, storage, and transmission because of transmission errors, camera sensors with distorted pixel elements, and incorrect memory positions [1]. Thus, suppression of such IN from an image is necessary for further processing. If O_{ij} is the luminance value of a true image O at pixel coordinates (i,j) and $[n_{min}, n_{max}]$ represents the dynamic range of O, the IN model with noise ratio r can be given as:

$$A_{ij} = \begin{cases} O_{ij}, & \text{with probability} \quad 1-r \\ \eta_{ij}, & \text{with probability} \quad r \end{cases}, \tag{1}$$

where $\eta_{ij} \in [n_{min}, n_{max}]$ are random values. For IN, noise distributions can be grouped as fixed-valued impulse noise (FVIN), which can also be described as salt-and-pepper noise, and random-valued impulse noise (RVIN). The noise value η_{ij} only takes values of 0 or 255 with $r/2$ equal probabilities for 8-bit FVIN-corrupted gray-scale images. However, for RVIN, the noise value η_{ij} can take any uniformly distributed integer value between 0 and 255. Therefore, removal of the RVIN is quite a bit more difficult than the removal of FVIN.

Many methods have been proposed in the literature to remove IN [2–32]. For example, the standard median filter is widely utilized for removing IN. It has the advantages of good noise reduction power [2] and effective computation [3]. IN has a gray-level value different from uncorrupted neighborhood pixels. The standard median filter uses this fact to remove IN. If the impulse noise rate is higher than 50%, the filter blurs the base image [4]. In order to overcome this drawback, center-weighted median (CWM) filters [5], tri-state median (TSM) filters [6], and lower-upper median (LUM) filters [7] have been proposed. At a low noise density,

[*] Correspondence: titi@erciyes.edu.tr

these methods have good performances, but at a high noise density, their performances decrease. These methods modify both the corrupted and noncorrupted pixels of a noisy image.

Recently, to prevent the damage of good pixels, many noise removal methods that employ different noise detectors have been proposed [8–20]. The recursive adaptive-CWM (ACWM) filter [10] utilizes the differences between the existing pixel and the CWM filter's output [5] with changed center weights to form estimates. The switching structure of the ACWM filter consists of impulse detection mechanisms. It uses the differences between the outputs of the CWM filters and the existing pixel to make the impulse detection. The final output is switched among the median and the existing pixel itself. As such, although the edge and detail protection performance of the ACWM filter is better than that of the median filter, its noise removal performance decreases.

The modified progressive switching median (MPSM) filter [12] is based on the known progressive switching median (PSM) filter [11] and ACWM filter [10]. The switching strategy of the MPSM filter consists of 2 stages: the ACWM-based impulse noise detector is utilized to find noisy pixels in the first stage and the PSM filtering procedure is performed for these noisy pixels in the second stage.

In the switching median (SM) filter [13], a switching strategy for median filtering is proposed for removing IN in images. The local impulse measurements are used in the switching procedure.

The multistate median (MSM) filter [14] presents a generalized median-based switching strategy. The output of the MSM filter is switched using simple thresholding logic. The output is switched among a set of different CWM filters.

In [15], a 2-state recursive signal-dependent rank-order mean filter (SDROM) method for removing IN was proposed. In the SDROM method, the filtering step is related to a state variable that is defined as the output of a classifier. It provides a good transition between noise reduction and detail protection over the simple median filter with little increase in the computational complexity.

Recently, artificial advanced soft computing methods such as neural networks (NNs) [31,32] and fuzzy systems have been used for IN detection and reduction [21–32]. In the if-then-else fuzzy reasoning (FIRE) filter [21], a fuzzy logic approach is used to enhance the images degraded by IN. The fuzzy operator is dependent on 2-stage fuzzy judgment. The fuzzy IN detection and reduction method (FIDRM) [22] is proposed to reduce different types of impulse noise. The FIDRM can also be used for images that have a mixture of IN and other noise types. It consists of 2 different stages: an IN detection stage and a reduction stage. In the detection stage, a fuzzy set IN is constructed using the fuzzy gradient values concept. The membership functions are used to represent this fuzzy set and it will be utilized in the filtering stage. In the filtering stage, a fuzzy averaging of neighboring pixels is performed. The fuzzy random impulse noise reduction method (FRINRM) [23] is also a 2-step fuzzy filter that uses a fuzzy logic method to restore the images degraded with IN. The FRINRM method contains a fuzzy detection procedure and a fuzzy filtering technique for removing RVIN from corrupted images.

The fuzzy system-based methods given in [21–24] are suitable to model the uncertainty of a noisy image. However, creating the rule-base structure is quite difficult for highly corrupted images. For this reason, many methods [25–29] related to the adaptive neuro-fuzzy inference system (ANFIS) are presented to prevent this difficulty. When ANFIS-based methods are trained properly, they can maintain the details of the images during detection and suppression of the noise.

In [30], a NN was proposed for impulse detection. The inputs of the neural model used in [30] are the median deviations described as the difference between the median value and actual value of the existing pixel. It is decided whether the pixel under test is corrupted or not using the neural model. The main disadvantages of the neural model presented in [30] are that it requires much input data and the difficulty of the training images selection. In [31], a preliminary and NN detector named 2 subdetectors was presented to detect IN. In order to avoid false classification, the impulses assigned in the preliminary stage are applied to the NN. The local statistics are used as input by the NN. A feedforward NN-based detector to detect salt-and-pepper noise was also presented in [32]. The NN has 3 inputs, median, current pixel, and rank-ordered absolute difference values.

The NN based detectors given in [30–32] were used to detect FVIN. However, the presented neural detector can detect RVIN. The inputs of the neural model are the mean value of the 8 basic gradient values and maximum values of the 2 related gradients defined in [22]. The neural detector is trained with an artificial training image formed on a computer [27]. The most important advantage of the presented neural detector is that it does not require discovering the rule-base structure as in [22]. Once the neural detector is trained, it can be used to detect the RVIN of any test image. The success of the presented method is evaluated with the results of the CWM, TSM, ACWM, MPSM, SM, MSM, SDROM, FIRE, and FIDRM methods.

The rest of the paper is outlined as follows: the details of the presented neural detector are given in Section 2. The simulation results are presented in Section 3 and, finally, conclusions are given in Section 4.

2. Proposed method

The main structure of the presented neural detector is given in Figure 1, where it is shown that the presented detector consists of a multilayer perceptron (MLP) NN [33] and a decision maker. For the neural model, input variables are formed using the gradient values described in [22]. The first input of the neural model is the mean value of the 8 basic gradient absolute values, defined by $\bar{\nabla}A(i,j)$. The second and the third inputs of the neural model are the maximum values of the 2 related gradients, which are defined by max $\nabla' A(i,j)$ and max $\nabla'' A(i,j)$. In Figure 1, the output of the detector is represented by $Y(i,j)$. The neural detector output is fed into the inputs of the decision maker. The decision maker output, represented by $y(i,j)$, is calculated as follows:

$$y(i,j) = \begin{cases} L_{\min} & \text{if} \quad Y(i,j) < \frac{L_{\min}+L_{\max}}{2} \\ L_{\max} & \text{if} \quad Y(i,j) \geq \frac{L_{\min}+L_{\max}}{2} \end{cases}, \tag{2}$$

where L_{min} and L_{max} show the minimum and the maximum allowable gray-level values. For 8-bit images, L_{min} and L_{max} are equal to 0 and 255, respectively.

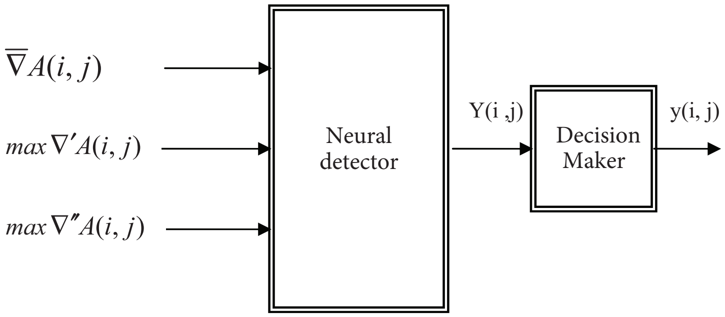


Figure 1. The general structure of the proposed neural RVIN detection operator.

The good generalization of the proposed neural noise detector is closely related to the selected training image. In this paper, the neural detector is trained with the artificially generated training image given in [27].

The artificial training image is preferred because it leads to increased generalization capability. The 512 × 512 artificial training image used as the original training image is illustrated in Figure 2a. In this image, 16 pixels within each 4 × 4 square box have the same randomly generated integer gray-level value, uniformly distributed in [0, 255].

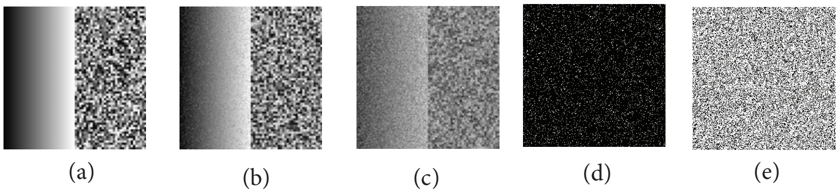


Figure 2. Training images: a) base training image, b) first input training image, c) second input training image, d) first target training image, e) second target training image.

The first and second input training images illustrated in Figures 2b and 2c are formed by corrupting the original training image with RVIN noise densities of 15% and 60%. It is seen from the simulation results that after the training is implemented, the neural detector gives the best results if the noise density of the input training image is equal to that of the noisy test image. It is also seen that if the difference between the noise densities of the input training image and test image increases, the performance of the proposed neural detector decreases. For this reason, 2 different neural models that have 2 different training images with 15% and 60% RVIN, respectively, are used to obtain a more stable filter for all of the noise densities.

Figures 2d and 2e show the target training images used to train the neural detector. These images are difference images obtained by subtracting the input training image pixel values from the corresponding original training image pixel values. Next, the nonzero and zero values are converted to white and black pixels, respectively. In this situation, the white pixels in the obtained target training image show corrupted pixels in the input training image.

In this paper, the MLP NN was trained using the Levenberg–Marquardt (LM) algorithm [34]. The simulation studies show that the change in the learning method does not change the performance of the proposed detector. For this reason, the LM algorithm is preferred because of its fast learning ability. In the MLP, linear transfer functions are used for the input and output layers and tangent sigmoid functions are used for the hidden layers. After several trials, the most proper network structure is chosen as 1 hidden layer with 3 neurons. This network configuration is trained with the 2 different training data sets described above. As such, 2 different trained networks are obtained to detect the noisy pixels of the different test images corrupted with RVIN. The number of epoch was 20 for the training. The value of μ used in the LM algorithm for weight updating is chosen as 0.001. The mean squared error (MSE) values obtained in the training are 0.194 and 0.714 for the first and second networks, respectively.

After the IN detection step is implemented using the proposed neural detector for any test image, the filtering step is realized [17]. First, the detected corrupted pixels are eliminated from 3 × 3 pixel window W. Next, median gray-level values of the other pixels within W are calculated. Finally, the calculated median value is used as the new gray-level value of the corrupted pixel. When the noise density is high, there may not be any uncorrupted pixels within W. In this situation, window size W is increased until at least one pixel within W is ensured.

The proposed neural detector gradually detects the corrupted pixels in 3 phases. First, the testing data set for the test image is formed and applied to the trained network. As such, many of the corrupted pixels are eliminated and the filtering step is realized. After the first detection and filtering phase is completed, the restored image is used as a noisy image in the second phase and similar steps as in the first phase are repeated. Hence, some of the corrupted pixels missed in the first phase are detected and suppressed. The restored image is again used as a noisy image in the third phase. Finally, the third and last detection phase is completed, which is realized to help the determination of the noisy pixels missed in the first and the second phases.

According to all of the information given above, the algorithmic steps of the presented method can be given as follows:

Training stages:

1. Form training inputs $\bar{\nabla}A(i,j)$, max $\nabla' A(i,j)$, and max $\nabla'' A(i,j)$ using an artificially generated training image corrupted with RVIN at 15% noise density for the first network. Form target training image by subtracting the gray-level values of the input training image from the gray-level values of the corresponding pixels of the original training image. Next, convert the nonzero and zero values to white and black pixels, respectively.

2. Train the neural model for 20 epochs with the LM learning algorithm using the data set generated in step 1.

Operational stages:

For a given image, A = [A(i, j)], i = 1, 2, . . . M, and j = 1, 2, . . . , N, apply the following:

1. **First phase:**

   (a) Form test inputs $\bar{\nabla}A(i,j)$, *max* $\nabla' A(i,j)$, and *max* $\nabla'' A(i,j)$ for any noisy test image for all i = 1, 2, . . . M, and j = 1, 2, . . . , N.

   (b) Compute decision maker output $Y(i,j)$ for the test inputs utilizing the NN that is trained in step 2.

   (c) Feed the network output $Y(i,j)$ for each pixel into the inputs of the decision maker, as shown in Figure 1. Find the decision maker output $y(i,j)$ for each pixel using Eq. (2). If the decision maker output is 255, then this pixel is determined as a RVIN-corrupted pixel.

   (d) Perform the filtering step for each corrupted pixel to complete the first phase.

2. **Second phase:** Use the restored image obtained from the first phase as the corrupted image for the second phase. Repeat the procedures realized in the first phase.

3. **Third phase:** Use the restored image obtained from the second phase as the corrupted image for the third phase. Repeat the procedures realized in the first phase.

4. Repeat above third phases for the second network trained with the artificially generated training image corrupted with RVIN at 60% noise density.

5. If the PSNR value of the filtered image obtained utilizing the second network is smaller than that of the first network, accept the results obtained from the second network as the best restoration results. Otherwise, accept the results obtained from the first network as the best restoration results.

A block diagram of the proposed expert system for removing RVIN is given in Figure 3, where neural detector-I and neural detector-II represent the first and second neural detectors trained with the artificially generated training image with RVIN at 15% and 60% noise densities, respectively.

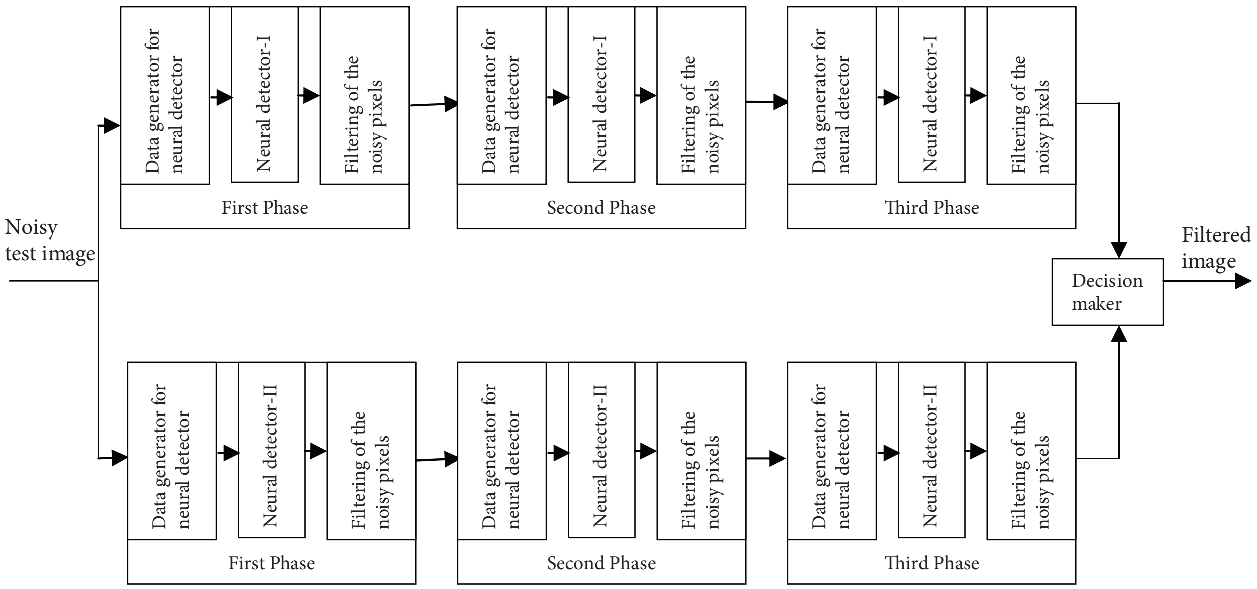


Figure 3. The proposed neural system for removing RVIN in the images.

3. Simulation results

In this section, the success of the presented method is evaluated with a number of experiments under different noise densities. The 512 × 512-sized 4 popular 8-bit gray-level Zelda, Peppers, Bridge, and Goldhill images from the literature are used for the simulations. The 4 images used to test the proposed detector are shown in Figure 4. The noisy test images used in the experiments are generated at different noise densities, changing between 15% and 75%, using a uniformly distributed RVIN model that is within the dynamic range of [0–255].

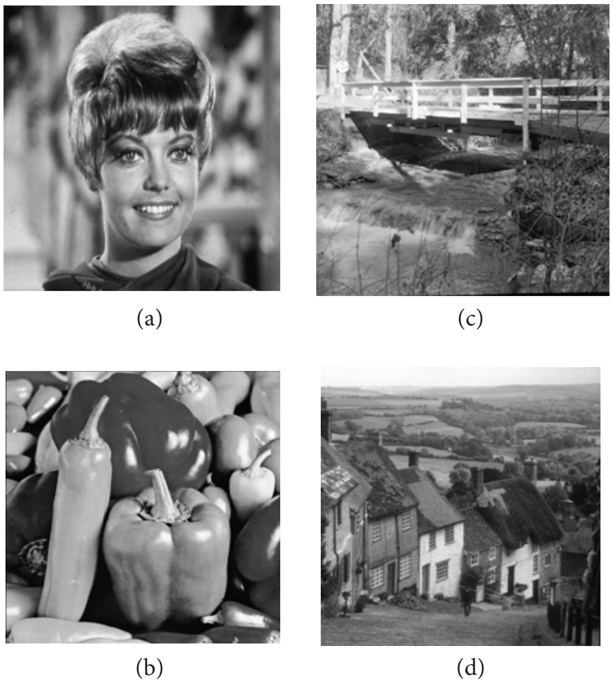
The CWM, TSM, ACWM, SM, MSM, SDROM, and FIRE comparison filters use a number of threshold parameters. For the simulation studies performed in this paper, these parameters are as follows: CWM ($T_0 = 55$, $T_1 = 40$, $T_2 = 25$, $T_3 = 15$), TSM (window size = 3 × 3 pixels, T = 20), ACWM ($\delta_1 = 40$, $\delta_2 = 25$, $\delta_3 = 10$, $\delta_4 = 5$, s = 0.1), SM ($w_s = 3$, T = 40, R = 0), MSM (window size = 3 × 3 pixels, $w_{max} = 5$, T = 20, R = 0), SDROM ($\delta_1 = 8$, $\delta_2 = 20$, $\delta_3 = 40$, $\delta_4 = 50$, s = 0.1, and window size = 3 × 3 pixels), and FIRE (L = 256, a = 40, b = 32).

3.1. Comparison of image restoration

In this study, in order to illustrate the restoration performance of the proposed method, the peak signal-to-noise ratio (PSNR) and normalized cross-correlation coefficient (NCC) [35] image quality assessment metrics are used. The PSNR is given by:

$$PSNR = 10\log_{10}\left(\frac{255^2}{MSE}\right) dB, \tag{3}$$

where the MSE is described as:



(a) (c)

(b) (d)

Figure 4. Test images: a) Zelda, b) Peppers, c) Bridge, d) Goldhill.

$$MSE = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}\left[org(i,j) - img(i,j)\right]^2, \tag{4}$$

where *org* and *img* are the original and filtered images of sizes M and N, respectively.

The NCC is given as:

$$NCC = \frac{\sum\limits_{i=1}^{M}\sum\limits_{j=1}^{N}\left(org(i,j) - \mu_{org(i,j)}\right)\left(img(i,j) - \mu_{img(i,j)}\right)}{\sqrt{\sum\limits_{i=1}^{M}\sum\limits_{j=1}^{N}\left(org(i,j) - \mu_{org(i,j)}\right)^2 \sum\limits_{i=1}^{M}\sum\limits_{j=1}^{N}\left(img(i,j) - \mu_{img(i,j)}\right)^2}}, \tag{5}$$

where $\mu_{org(i,j)}$ and $\mu_{img(i,j)}$ are the mean values of the original and filtered image pixels and are defined as:

$$\mu_{org(i,j)} = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N} org(i,j) \tag{6}$$

and

$$\mu_{img(i,j)} = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N} img(i,j). \tag{7}$$

The PSNR and NCC values of the presented method and the other methods are listed in Tables 1–4 for making a quantitative evaluation of the presented method against the above-mentioned benchmark methods, based on the test images. Tables 1–4 clearly show that the PSNR and NCC values for the proposed method are better than those of the other filters, except for the MPSM, ACWM, and FIDRM at a 15% noise density for only 1 test image. It is also clear from Tables 1–4 that the proposed method is better than the other mentioned filters in terms of noise suppression and detail preservation, although the impulse noise rate is very high.

Table 1. PSNR and NCC values of the proposed filter and comparison filters for the Zelda test image at different noise densities.

Method	PSNR					NCC				
	15%	30%	45%	60%	75%	15%	30%	45%	60%	75%
Noisy	17.16	14.19	12.39	11.15	10.19	0.704	0.513	0.365	0.247	0.149
CWM	34.66	29.81	25.57	21.00	16.58	0.993	0.979	0.945	0.847	0.596
TSM	36.29	27.80	21.81	17.60	14.69	0.995	0.966	0.872	0.685	0.434
ACWM	36.56	31.04	25.88	20.72	16.14	0.996	0.984	0.949	0.837	0.572
MPSM	38.92	31.14	24.52	19.33	15.49	0.997	0.984	0.930	0.780	0.510
SM	32.86	27.11	21.80	17.67	14.76	0.989	0.961	0.872	0.689	0.439
MSM	35.66	26.74	21.02	17.11	14.37	0.994	0.957	0.848	0.654	0.411
SDROM	36.59	27.95	21.85	17.50	14.54	0.995	0.967	0.874	0.680	0.427
FIRE	30.73	27.35	24.52	21.17	16.98	0.983	0.963	0.931	0.853	0.631
FIDRM	39.05	32.54	23.90	17.42	13.41	0.997	0.988	0.918	0.675	0.356
Proposed	39.51	33.86	30.72	26.86	20.84	0.992	0.981	0.967	0.939	0.832

Table 2. PSNR and NCC values of the proposed filter and comparison filters for the Peppers test image at different noise densities.

Method	PSNR					NCC				
	15%	30%	45%	60%	75%	15%	30%	45%	60%	75%
Noisy	16.69	13.70	11.91	10.65	9.69	0.803	0.631	0.475	0.331	0.199
CWM	32.98	28.47	24.16	19.50	15.17	0.994	0.985	0.961	0.887	0.678
TSM	33.71	26.23	20.62	16.48	13.64	0.996	0.976	0.913	0.768	0.530
ACWM	34.05	29.21	24.29	19.29	14.81	0.996	0.988	0.963	0.881	0.652
MPSM	34.96	29.08	23.10	18.00	14.34	0.996	0.987	0.951	0.839	0.606
SM	31.55	25.90	20.65	16.54	13.71	0.993	0.974	0.913	0.771	0.537
MSM	33.25	25.36	19.94	16.06	13.39	0.995	0.971	0.898	0.743	0.505
SDROM	33.72	26.45	20.66	16.41	13.53	0.996	0.977	0.913	0.765	0.524
FIRE	30.27	26.76	23.63	19.81	15.53	0.990	0.979	0.956	0.895	0.706
FIDRM	32.70	29.57	22.12	16.03	12.51	0.994	0.989	0.938	0.747	0.450
Proposed	34.92	31.40	27.34	22.38	18.50	0.994	0.988	0.975	0.952	0.867

The restoration results for the Bridge test image corrupted with 60% IN are shown in Figure 5, where it is clear that the visual quality of the proposed method is better than those of the other filters, even if the impulse noise rate is high.

3.2. Detection performance

In order to obtain a satisfactory restoration performance from the noise reduction filter, the performance of the noise detector is very important. In this study, the performance of the presented method is evaluated with the methods that have noise detectors. Only the CWM method does not use a noise detector. Table 5 illustrates the number of uncorrupted pixels (false detections) and the number of missed corrupted pixels (missed detections) for the Zelda image corrupted with 15% and 75% noise densities. Because the luminance values of the noisy pixels may be similar to those neighbors for RVIN, it is not easy for a noise detector to find most of the corrupted pixels while ensuring the rate of false detection be as small as possible. A good noise detector can find most of the corrupted pixels and its false detection rate must be as low as possible.

Table 3. PSNR and NCC values of the proposed filter and comparison filters for the Bridge test image at different noise densities.

Method	PSNR					NCC				
	15%	30%	45%	60%	75%	15%	30%	45%	60%	75%
Noisy	17.06	14.05	12.24	10.99	10.03	0.799	0.624	0.463	0.321	0.192
CWM	28.07	24.76	21.75	18.63	15.39	0.982	0.962	0.924	0.837	0.627
TSM	27.64	24.01	20.04	16.66	14.10	0.980	0.955	0.885	0.737	0.499
ACWM	28.39	24.91	21.73	18.32	14.98	0.984	0.963	0.923	0.827	0.602
MPSM	27.49	25.00	21.54	17.85	14.74	0.980	0.964	0.920	0.804	0.572
SM	27.60	23.86	20.07	16.73	14.17	0.980	0.954	0.886	0.743	0.507
MSM	27.78	23.66	19.57	16.28	13.84	0.981	0.951	0.872	0.714	0.476
SDROM	27.93	24.02	20.00	16.56	13.96	0.982	0.955	0.885	0.734	0.493
FIRE	27.95	24.61	21.86	18.94	15.73	0.982	0.961	0.926	0.850	0.661
FIDRM	27.30	21.72	20.37	16.07	12.83	0.979	0.924	0.894	0.708	0.416
Proposed	27.12	25.46	22.94	20.58	17.10	0.976	0.963	0.935	0.891	0.797

Table 4. PSNR and NCC values of the proposed filter and comparison filters for the Goldhill test image at different noise densities.

Method	PSNR					NCC				
	15%	30%	45%	60%	75%	15%	30%	45%	60%	75%
Noisy	17.37	14.35	12.57	11.33	10.37	0.771	0.586	0.430	0.294	0.174
CWM	32.04	27.88	24.52	20.67	16.76	0.991	0.977	0.950	0.873	0.665
TSM	32.44	26.80	21.77	17.84	14.99	0.992	0.970	0.903	0.754	0.509
ACWM	32.85	28.44	24.57	20.32	16.24	0.992	0.980	0.950	0.865	0.638
MPSM	32.95	28.81	23.88	19.38	15.77	0.993	0.981	0.941	0.830	0.590
SM	30.94	26.24	21.72	17.89	15.06	0.988	0.966	0.903	0.758	0.516
MSM	32.45	26.06	21.06	17.34	14.66	0.992	0.964	0.886	0.725	0.481
SDROM	32.72	26.79	21.71	17.72	14.83	0.993	0.970	0.902	0.750	0.501
FIRE	29.92	26.56	23.88	20.78	17.19	0.985	0.969	0.942	0.877	0.702
FIDRM	32.15	28.89	23.39	17.59	13.66	0.991	0.981	0.934	0.744	0.428
Proposed	32.84	29.31	26.95	24.70	20.56	0.988	0.974	0.958	0.935	0.855

Table 5 clearly illustrates that at 15% clutter density, the proposed detector produces the lowest false detection rate. Although the missed detection number of the proposed detector is higher than those of the other methods, it is shown from Table 5 that the proposed method produces the lowest sum, except for the MPSM when the numbers of false and missed detected pixels are added. For the 75% low clutter density, the sum of the missed and false detected pixels is also the lowest for the proposed method among all of the methods. If the number of false detections increases in the filtered image, a large number of residual noises will seriously damage the image. On the contrary, if the number of missed detections increases, the image is blurred. Therefore, the proposed neural detector gives a better tradeoff between false detection and missed detection.

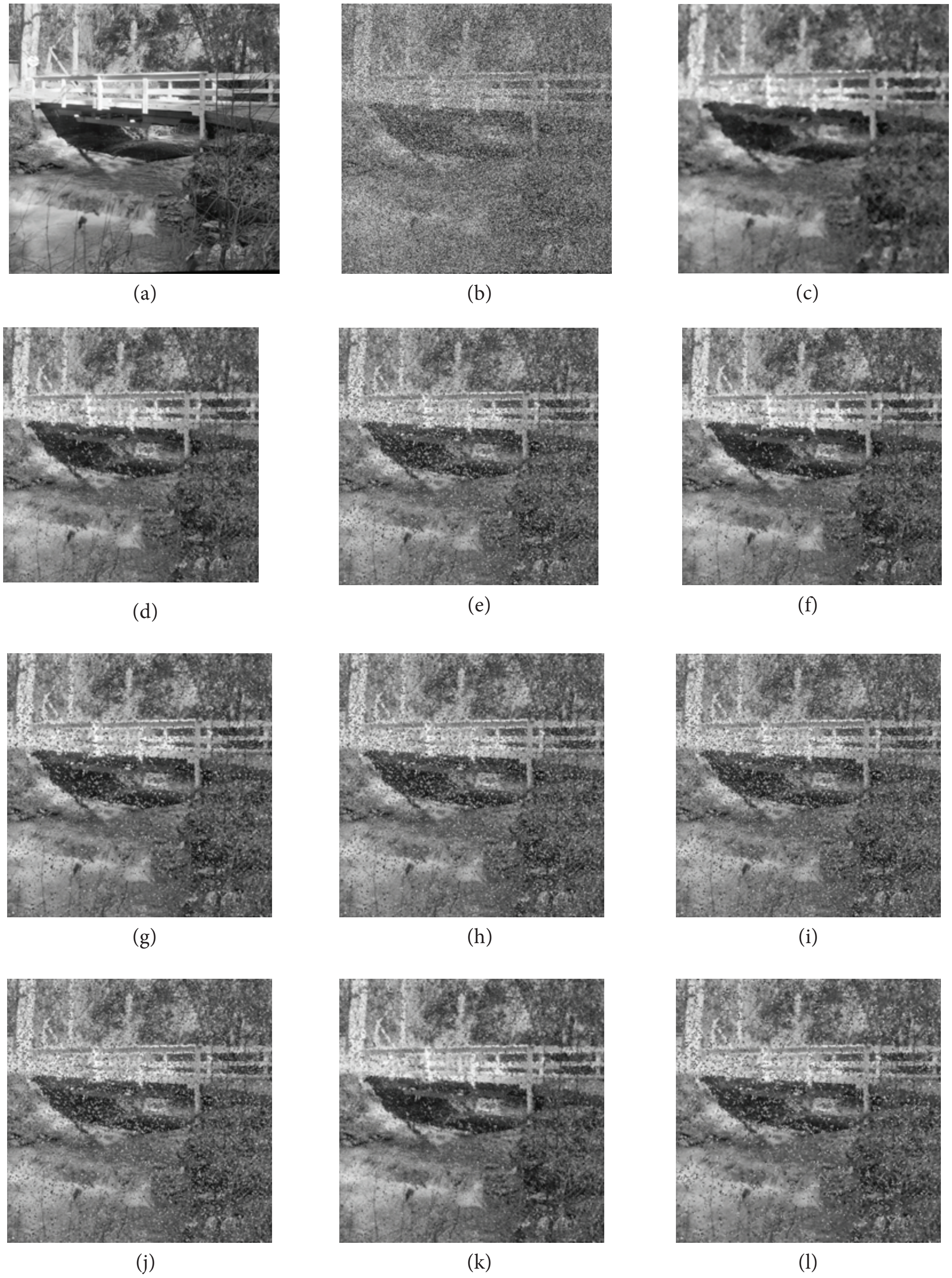


Figure 5. Output images of the operators for the Bridge image corrupted with RVIN at 60% noise density: a) original Bridge image, b) noisy Bridge image, c) new, d) CWM, e) TSM, f) ACWM, g) MPSM, h) SM, i) MSM, j) SDROM, k) FIRE, l) FIDRM.

Table 5. Comparison of the noise detection results for the Zelda test images corrupted with random-valued impulse noise.

Method	15%		75%	
	False	Miss	False	Miss
TSM	602	6509	15,586	63,033
ACWM	17	7794	3351	86,047
MPSM	460	4666	11,518	69,970
SM	87	12,018	12,093	75,774
MSM	488	6359	11,167	73,930
SDROM	105	6522	9792	80,402
FIRE	52	11,683	8118	66,171
FIDRM	10,574	2329	7666	99,411
Proposed	361	5192	37,453	12,293

Another crucial demand expected from modern noise reduction filters is robustness. It is clearly shown from Tables 1–5 and Figure 5 that the presented neural detector is fairly robust for different noise ratios.

3.3. Running time

The running time for the presented method and the other methods mentioned are given in Table 6, where it is clear that the running time of the proposed method for the Peppers test image is better than those of the CWM, ACWM, FIRE, and FIDRM methods and worse than the others at a high noise density. The runtime analysis of the mentioned filters is performed using a Pentium dual-core, 2.52-GHz PC.

Table 6. Runtimes of the mentioned methods for the Peppers image for different IN levels.

Method	Runtime (s)	
	15%	75%
CWM	33.45	34.45
TSM	8.65	8.68
ACWM	39.73	40.61
MPSM	3.53	5.91
SM	0.31	0.31
MSM	0.97	1.01
SDROM	0.66	0.67
FIRE	20.41	20.43
FIDRM	4.38	62.79
Proposed	18.92	18.70

4. Conclusions

In this paper, a new high-performance noise removal method based on a NN is presented. The realization of the proposed neural detector is very easy. The main structure of the detector is a simple 3-input, 1-output NN system. Training of the detector can easily be realized by utilizing artificial training images. The performance of the proposed method for different test images can be easily shown from Tables 1–4, where it is clearly seen that the MSE and NCC values of the presented method are better than the MSE and NCC values of the CWM, TSM, ACWM, MPSM, SM, MSM, SDROM, FIRE, and FIDRM methods. It can also be seen from Figure 5 that the proposed method produces better restoration results according to the other mentioned filters.

The advantages of the proposed method may be summarized as follows. First, the realization of the method is very easy. It is based on a simple 3-input, 1-output NN system. The parameters of the NN system are obtained by training. Training of the system can easily be performed using artificial images generated on a computer. Next, the proposed method effectively removes RVIN from images and the restored image obtained with the proposed method has a higher resolution according to the comparison filters mentioned in this paper, especially for highly corrupted images. Finally, the proposed method gives a better tradeoff between false detection and missed detection and it is robust at low and high noise densities.

The disadvantage of the proposed method is that its computation time is higher than those of the TSM, MPSM, SM, MSM, and SDROM methods at high noise densities.

Acknowledgment

The author is grateful to Dr Stefan Schulte for sending source codes of some of the comparison filters mentioned in this paper.

References

[1] R.C. Gonzalez, R.E. Woods, Digital Image Processing, Englewood Cliffs, NJ, USA, Prentice Hall, 2002.

[2] A. Bovik, Handbook of Image and Video Processing, New York, Academic Press, 2000.

[3] T.S. Huang, G.J. Yang, G.Y. Tang, "Fast two-dimensional median filtering algorithm", IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol. 1, pp. 13–18, 1979.

[4] T.A. Nodes, N.C. Gallagher Jr, "The output distribution of median type filters", IEEE Transactions on Communications, Vol. 32, pp. 532–541, 1984.

[5] S.J. Ko, Y.H. Lee, "Center weighted median filters and their applications to image enhancement", IEEE Transactions on Circuits and Systems, Vol. 38, pp. 984–993, 1991.

[6] T. Chen, K.K. Ma, L.H. Chen, "Tri-state median filter for image denoising", IEEE Transactions on Image Processing, Vol. 8 , pp. 1834–1938, 1999.

[7] R.C. Hardie, C.G. Boncelet, "LUM filters: A class of rank-order-based filters for smoothing and sharpening", IEEE Transactions on Signal Processing, Vol. 41, pp. 1834–1838, 1993.

[8] H.L. Eng, K.K. Ma, "Noise adaptive soft-switching median filter", IEEE Transactions on Image Processing, Vol. 10, pp. 242–251, 2001.

[9] M.E. Paul, K. Bruce, C Language Algorithms for Digital Signal Processing, Englewood Cliffs, NJ, USA, Prentice Hall, 1991.

[10] T. Chen, H.R. Wu, "Adaptive impulse detection using center-weighted median filters", IEEE Signal Processing Letters, Vol. 8, pp. 1–3, 2001.

[11] Z. Wang, D. Zhang, "Progressive switching median filter for the removal of impulse noise from highly corrupted images", IEEE Transactions on Circuits and Systems, Vol. 46, pp. 78–80, 1999.

[12] D. Kuykin, V. Khryashchev, I. Apalkov, "Modified progressive switched median filter for image enhancement", Proceedings of the International Conference on Computer Graphics and Vision, pp. 303–304, 2009.

[13] T. Sun, Y. Neuvo, "Detail-preserving median based filters in image processing", Pattern Recognition Letters, Vol. 15, pp. 341–347, 1994.

[14] T. Chen, H.R. Wu, "Space variant median filters for the restoration of impulse noise corrupted images", IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Process, Vol. 48, pp. 784–789, 2001.

[15] E. Abreu, M. Lightstone, S.K. Mitra, K. Arakawa, "A new efficient approach for the removal of impulse noise from highly corrupted images", IEEE Transactions on Image Processing, Vol. 5, pp. 1012–1025, 1996.

[16] H. Lin, A.N. Willson, "Median filter with adaptive length", IEEE Transactions on Circuits and Systems, Vol. 35, pp. 675–690, 1988.

[17] E. Beşdok, M.E. Yüksel, "Impulsive noise suppression from images with Jarque-Bera test based median filter", AEÜ International Journal of Electronics and Communications, Vol. 59, pp. 105–110, 2005.

[18] J. Wu, C. Tang, "PDE-based random-valued impulse noise removal based on new class of controlling functions", IEEE Transactions on Image Processing, Vol. 20, pp. 2428–2438, 2011.

[19] J. Wu, C. Tang, "A new filter for the removal of random-valued impulse noise from highly corrupted images", AEÜ International Journal of Electronics and Communications, Vol. 66, pp. 847–854, 2012.

[20] Y.Y. Zhou, Z.F. Ye, J.J. Huang, "Improved decision-based detail-preserving variational method for removal of random-valued impulse noise", IET Image Processing, Vol. 6, pp. 976–985, 2012.

[21] F. Russo, G. Ramponi, "A fuzzy filter for images corrupted by impulse noise", IEEE Signal Processing Letters, Vol. 3, pp. 168–170, 1996.

[22] S. Schulte, M. Nachtegael, V.D. Witte, D.V. Der Weken, E.E. Kerre, "A fuzzy impulse noise detection and reduction method", IEEE Transactions on Image Processing, Vol. 15, pp. 1153–1162, 2006.

[23] S. Schulte, V. De Witte, M. Nachtegael, D. Van Der Weken, E.E. Kerre, "Fuzzy random impulse noise reduction method", Fuzzy Sets and Systems, Vol. 158, pp. 270–283, 2007.

[24] T. Mélange, M. Nachtegael , S. Schulte, E.E. Kere, "A fuzzy filter for the removal of random impulse noise in image sequences", Image and Vision Computing, Vol. 29, pp. 407–419, 2011.

[25] C.S. Lee, Y.H. Kuo, "Adaptive fuzzy filter and its application to image enhancement", Fuzzy Techniques in Image Processing, Vol. 52, pp. 172–193, 2000.

[26] M.E. Yüksel, A. Baştürk, "Efficient removal of impulse noise from highly corrupted digital images by a simple neuro-fuzzy operator", AEÜ International Journal of Electronics and Communications, Vol. 57, pp. 214–219, 2003.

[27] M.E. Yüksel, "A median/ANFIS filter for efficient restoration of digital images corrupted by impulse noise", AEÜ International Journal of Electronics and Communications, Vol. 60, pp. 628–637, 2006.

[28] M.T. Yildirim, A. Basturk, M.E. Yuksel, "Impulse noise removal from digital images by a detail-preserving filter based on type-2 fuzzy logic", IEEE Transactions on Fuzzy Systems, Vol. 16, pp. 920–928, 2008.

[29] I. Turkmen, "Efficient impulse noise detection method with ANFIS for accurate image restoration", AEÜ International Journal of Electronics and Communications, Vol. 65, pp. 132–139, 2011.

[30] H. Kong, L. Guan, "A noise-exclusive adaptive filtering framework for removing impulse noise in digital images", IEEE Transactions on Circuits and Systems Part I: Analog Digital Signal Process, Vol. 45, pp. 422–428, 1998.

[31] I.V. Apalkov, P.S. Zvonarev, V.V. Khryashchev, "Neural network adaptive switching median filter for image denoising", Proceedings of EUROCON, pp. 22–24, 2005.

[32] G. Kaliraj, 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, pp. 458–466, 2010.

[33] S. Haykin, Neural Networks: A Comprehensive Foundation, New York, Macmillan Publishing, 1994.

[34] M.T. Hagan, M. Menjah, "Training feedforward networks with the Marquardt algorithm", IEEE Transactions on Neural Networks, Vol. 5, pp. 989–993, 1994.

[35] Z. Wang, A.C. Bovik, H.R. Sheikh, E.P. Simoncelli, "Image quality assessment: from error visibility to structural similarity", IEEE Transactions on Image Processing, Vol. 13, pp. 1–14, 2004.