Neural-network-based Impulse Noise Removal Using Group-based Weighted Couple Sparse Representation

Yongwoo Lee[1], Toan Duc Bui[1], Jitae Shin[1*] and Byung Tae Oh[2]
[1] School of Electronic and Electrical Engineering, Sungkyunkwan University
Suwon, South Korea
[e-mail: tencio2001@skku.edu; toanhoi@skku.edu jtshin@skku.edu]
[2] School of Electronic, Korea Aerospace University
Gyonggi-Do, South Korea
[e-mail: byungoh@kau.ac.kr]
*Corresponding author: Jitae Shin

Abstract

In this paper, we propose a novel method to recover images corrupted by impulse noise. The proposed method uses two stages: noise detection and filtering. In the first stage, we use pixel values, rank-ordered logarithmic difference values, and median values to train a neural-network-based impulse noise detector. After training, we apply the network to detect noisy pixels in images. In the next stage, we use group-based weighted couple sparse representation to filter the noisy pixels. During this second stage, conventional methods generally use only clean pixels to recover corrupted pixels, which can yield unsuccessful dictionary learning if the noise density is high and the number of useful clean pixels is inadequate. Therefore, we use reconstructed pixels to balance the deficiency. Experimental results show that the proposed noise detector has better performance than the conventional noise detectors. Also, with the information of noisy pixel location, the proposed impulse-noise removal method performs better than the conventional methods, through the recovered images resulting in better quality.

1. Introduction

In digital image processing, image denoising is an essential and challenging research topic used in numerous circumstances. Digital images often suffer from various types of noise. For example, impulse noise (IN) corrupts images during image acquisition, recording, and transmission [1]. IN is categorized as fixed-valued impulse noise (FVIN), also known as salt-and-pepper noise (SPN), and random-valued impulse noise (RVIN). When a pixel in an 8-bit image is corrupted by FVIN, its pixel value is either 0 or 255 with equal probability. An RVIN-corrupted pixel, on the other hand, has a value that is randomly changed between 0 and 255. Therefore, RVIN is more difficult to detect in a noisy image than FVIN.

Several techniques have been proposed to denoise IN corrupted images. The median filter is one popular method. The median filter first computes a median value in a mask. Next, the pixel in the center of the mask is replaced by the median value. However, because it adopts only the median value, the original pixel value cannot be found. Therefore, its performance is poor and it cannot preserve fine edges. To address those issues, some median filter–based modifications have been designed. However, those techniques, such as the weighted median filter and center weighted median (CWM) filter [2], still fail to remove IN thoroughly because of the limitations of the median filter. Recently, image in-painting techniques work very well for the noise removal [3-5]. In [4], salt-and-pepper noise removal method is proposed using image in-painting method. It takes noisy pixels as missing data and recovers the image by carefully choosing convolution mask according to the local regions. However, it is limited to salt-and-pepper noise.

Because IN corrupts only a few pixels, leaving the rest untouched, many IN removal techniques use two stages [6-14]. Those methods use noise detectors to detect IN-corrupted pixels in the first stage and then recover those pixels in the second stage. The robust outlyingness ratio with nonlocal means [6] and weighted mean filter with a two-phase detector [7] are examples that adopt the two-stage method.

Many researchers have proposed IN detectors. Chen et al. tried to solve the poor performance of median-based IN detection strategies for RVIN [10] by proposing the adaptive center-weighted median (ACWM) method that combines CWM with various center weights. Abreu et al. proposed the signal-dependent rank ordered mean (SDROM) technique, which replaces noise pixels with the rank-ordered mean of the surrounding pixels [11]. Schulte et al. proposed the fuzzy impulse noise detection and reduction method (FIDRM), which uses fuzzy gradient values to determine whether a particular pixel is IN-corrupted [12]. It calculates eight different gradient values according to direction, and considers two related gradient values in each direction to verify whether an edge in an image is causing a large gradient value. Its membership function calculates the gradient values using a fuzzy rule to identify noisy pixels. Schulte et al. also proposed the fuzzy random impulse noise reduction method (FRINRM) [13], a two-step fuzzy filter that uses fuzzy logic to enhance images corrupted with IN.

In the past few years, some neural network (NN)–based methods have been published. Sa et al. proposed the improved adaptive impulse noise suppression (IAINS) technique [14]. It uses the pixel-wise median of the absolute deviations from the median and rank-ordered absolute difference (ROAD) to train an NN. The trained network decides whether an identified pixel is clean or corrupted. Turkmen also proposed an artificial NN-based detector (ANN) that trains a network using ROAD and rank-ordered logarithmic difference (ROLD) [15]. Unlike [14], it

trains the NN using an artificially designed image with a gradient and square box patterns that are IN-corrupted with 70% noise density.

For the filtering stage, sparse representation (SR) has gained an excellent reputation over many years for degraded image restoration [16-18]. The basic idea of SR is that a signal can be estimated using a spare coefficient vector and a set of atoms called a dictionary. SR has been applied in many image processing areas, such as image denoising, deblurring, compressive sensing, super-resolution, image fusion and object recognition. Group-based sparse representation (GSR) [19] is one technique that provides a complete solution for image restoration. For image denoising, GSR performs well in removing Gaussian noise (GN). Unfortunately, GN and IN have different characteristics, so GSR methods for removing GN often fail when used for IN removal.

In this paper, we propose the NN-based Detector-Weighted couple group-based SR method (NND-WSR), which uses two stages to recover IN corrupted images. Our main contributions are as follows:

- To propose an IN detector based on an NN.
- To propose the WSR method to remove IN.
- To overcome the limitations of GSR and compensate for the insufficient information in images with high noise density by applying a weighted coupling method.
- To analyze NN-based and deep-learning-based noise detectors.

The rest of the paper is organized as follows. In Section 2, we provide background about IN model and several techniques to facilitate understanding of our proposed method. In Section 3, we describe our proposed method. In Section 4, we show experimental results and discuss deep-learning-based IN detectors. We conclude in Section 5.

2. Background

In IN removal, many proposed techniques use a two-stage process [6-15]. In the first stage, a noise detector identifies IN-corrupted pixels. In the second stage, the noisy pixels are recovered in some way. In this section, we discuss the noise model and reference methods that support our proposed technique.

2.1 Noise Model

Let x_{ij} be the gray level of an original image located at (i,j) in the range of $[v_{min}, v_{max}]$, and let y_{ij} be the gray level of a corrupted image located at (i,j). Impulse noise is expressed as

$$y_{ij} = \begin{cases} x_{ij} \ with \ 1-p \\ \eta_{ij} \ with \ p, \end{cases} \tag{1}$$

where η_{ij} is the corrupted IN value with a noise density p located at (i,j). When η_{ij} is a random value selected in the range of $[v_{min}, v_{max}]$, the image is said to be corrupted by RVIN, and if η_{ij} is either v_{min} or v_{max}, it is known as FVIN or SPN. An 8-bit image has v_{min} and v_{max} of 0 and 255, respectively. In this work, we consider only RVIN noise.

2.2 ROLD Statistic

The ROLD statistic was proposed in [9] as a modification of ROAD [8]. ROAD is a noise

detector that measures the probability that a pixel is corrupted by IN. Dong et al. improved ROAD into ROLD as follows. Let Ω_N denote the set of coordinates in a $(2N + 1)$-by-$(2N + 1)$ window centered at $(0,0)$ defined as

$$\Omega_N = \{(s,t)| - N \leq s, t \leq N\} \tag{2}$$

and let $\Omega_N^0 = \Omega_N(0,0)$. The logarithmic absolute difference is:

$$\widetilde{D}_{st}(y_{i,j}) = \log_a|y_{i+s,j+t} - y_{i,j}|, \ \forall(s,t) \in \Omega_N^0. \tag{3}$$

For any $a > 1$, the number $\widetilde{D}_{st}$ is always in $(-\infty, 0]$. Keeping it in the dynamic range $[0, 1]$ requires truncation and a linear transformation,

$$D_{st}(y_{i,j}) \equiv 1 + max\{\log_a|y_{i+s,j+t} - y_{i,j}|, -b\}/b \ \forall(s,t) \in \Omega_N^0, \tag{4}$$

where a, b are positive constants. All D_{st} values are formed in increasing order, and the kth smallest D_{st} for all $(s,t) \in \Omega_N^0$ is defined as R_k. The ROLD statistic is thus given by:

$$\text{ROLD}_m(y_{i,j}) = \sum_{k=1}^m R_k(y_{i,j}), \tag{5}$$

where $2 \leq m \leq (2N + 1)^2 - 2$.

2.3 Conventional GSR

GSR was proposed in [19] and shows image restoration in image in-painting, deblurring, and compressive sensing recovery. The main concept is to group similar patches and consider the relationships among them. The split Bregman iteration (SBI) is used to solve the optimization problem and reduce the computational complexity during the dictionary learning step. However, because it assumes an additive white Gaussian noise model, GSR cannot be directly applied to IN removal.

To apply GSR for our purpose, we considered the characteristics of IN. Because IN corrupts a certain number of pixels and leaves the rest untouched, clean pixels should be unchanged after reconstruction. Therefore, we apply a weighting operator to the conventional GSR method to only affect the corrupted pixels. A detailed explanation of GSR is found in [19]. Briefly, the conventional GSR is formulated as:

$$\widehat{\boldsymbol{\alpha}}_G = \text{argmin}_{\boldsymbol{\alpha}_G} \frac{1}{2}\|\boldsymbol{HD}_G \circ \boldsymbol{\alpha}_G - \boldsymbol{y}\|_2^2 + \lambda\|\boldsymbol{\alpha}_G\|_0, \tag{6}$$

where $\boldsymbol{H}$ is a non-invertible linear degradation operator, $\boldsymbol{D}_G$ is the GSR dictionary, $\boldsymbol{y}$ is the observed image, and λ is the regularization parameter. With $\widehat{\boldsymbol{\alpha}}_G$, the reconstructed image can be represented as $\widehat{\boldsymbol{x}} = \boldsymbol{D}_G \circ \widehat{\boldsymbol{\alpha}}_G$. Because we use clean pixels to recover the noisy pixels, we use the weighting operator $\boldsymbol{W}$ to distinguish clean pixels from noisy ones. The weighting operator has the same size of $\boldsymbol{y}$ and $w_{i,j}$ as its elements. The GSR equation for IN removal is thus as follows:

$$\widehat{\boldsymbol{\alpha}}_{G,IN} = \mathrm{argmin}_{\boldsymbol{\alpha}_{G,IN}} \frac{1}{2} \left\| \boldsymbol{W} \otimes \left(\boldsymbol{HD}_{G,IN} \circ \boldsymbol{\alpha}_{G,IN} - \boldsymbol{y} \right) \right\|_2^2 + \lambda \left\| \boldsymbol{\alpha}_{G,IN} \right\|_0, \qquad (7)$$

where $\otimes$ denotes the Hadamard (element-wise) product. When a pixel is IN corrupted, the corresponding element $w_{i,j}$ in $\boldsymbol{W}$ is zero because only clean pixels are used to estimate the original pixel value.

3. Proposed method

In NND-WSR, we use a two-stage IN removal process. For the first stage, we designed an NN-based IN detector. We use pixel values and compute the median and ROLD values to accurately detect IN. For the second stage, we designed the WSR method to restore the noisy pixels. In **Fig. 1**, we show an overall flowchart that accounts for the proposed method.
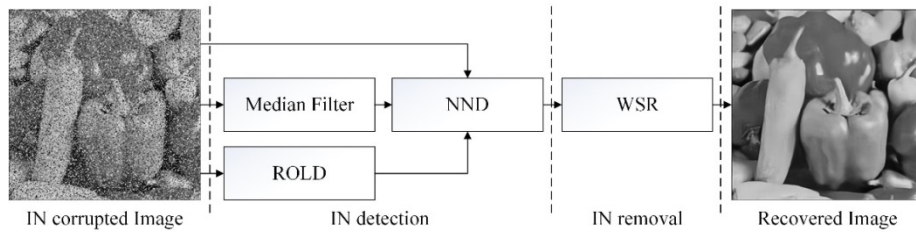

Fig. 2. A flowchart for the proposed IN removal method

3.1 Neural-network-based Detector (NND)

In [14] and [15], IN detectors are proposed based on NNs. In [14], pixel wise median of absolute deviation and ROAD values are used to train the NN with a 128×128 Lena image 15% corrupted with RVIN. In [15], ROAD and ROLD values are fed to the NN. A 128×128 test image is artificially generated and corrupted with RVIN at a 70% noise density. In the test image, one-tenth of the pixels are randomly selected for NN training. For our IN detector, we use ROLD, median and pixel values to feed the NN and assign more information. In addition, we generate artificial training images, as in [15], to prevent biased results and promote generalization. Although 30% noise density applied to our test images is much less than noisy density in [15], we choose this value because the noisy image with 30% noise density for learning will sufficiently cover the case with 0 to 60% noise density. A neural-network based method highly depends on input and label data during the training step. That is why we choose 30% noise density as median value of the range of 0 to 60%. In addition, we use every pixel in a 256×256 training image to have enough number of input data to train the network.

Before getting into the deeper explanation, we investigate how pixel, median, and ROLD values differentiate noisy and noise-free pixels from an image. First, median value is used with pixel value to estimate a pixel if their absolute difference is bigger than a certain threshold. Second, ROLD finds impulse noise by probablistic attempt in a range between 0 and 1. To support the proposed ideas, we show the distribution of noisy and noise-free pixels in the normalized pixel, median, and ROLD domains in **Fig. 2**, where approximately 500 pixels for each noisy and noise-free pixels are plotted. We can see that most of noise-free pixels are located on the line, while noisy pixels are scattered away from the line. Therefore, we can conclude that the noisy pixels can be differentiated using pixel, median, and ROLD value. To

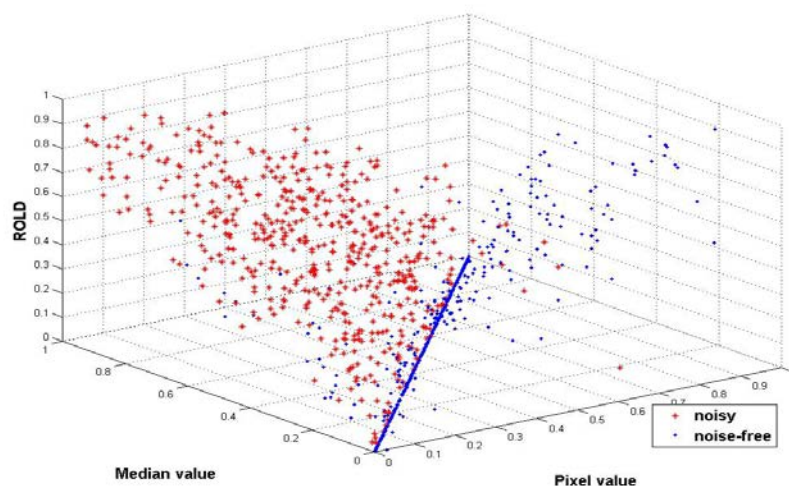solve this task, we apply the neural network-based approach.

Fig. 2. Distribution of noisy and noise-free pixels

The detailed procedure is as follows. First, we generate an artificial training image for the NN model, as depicted in **Fig. 3**. We generate a 256×256 image for the original training image. In the image, pixels are arranged in 8×8 patches, each of which has a single pixel value randomly chosen from within the normal distribution of [0 255]. Then, the image is corrupted with RVIN at a 30% noise density. A target training image is generated by subtracting the original image from the corrupted image. Non-zero values indicate noisy pixel locations, and those pixels are converted to white. In the training process, we calculate the ROLD and median values for each pixel and feed them to the NN model, with the noisy pixel location map used as the output. In total, we use 65,536 pixels to train the NN model via the feed forward neural network (FFNN) method. We design two hidden layers, with four nodes in each layer, because the number of layers and nodes does not affect the performance overmuch. The structure of our NN is depicted in **Fig. 4**.

Once the training is finished, our proposed IN detector uses the ROLD, median, and pixel values calculated from the test image. Those values are used as the input for our NN model, and the output values determine whether a pixel is IN corrupted or clean. The output $n(x_{i,j})$ is then used to calculate $w_{i,j}$, as follows:

$$w_{i,j} = \begin{cases} 0 \; if \; n(x_{i,j}) \geq 0.5 \\ 1 \; if \; n(x_{i,j}) \; < 0.5, \end{cases} \tag{8}$$

where $w_{i,j}$ is a weight value in the (i,j)-th pixel in the weighting operator $\boldsymbol{W}$ to be used in the second stage.

In this paper, we train our NN model with the resilient backpropagation (RP) algorithm because it has a fast learning capability and the choice of algorithm has little effect on performance, as we show in the discussion section.
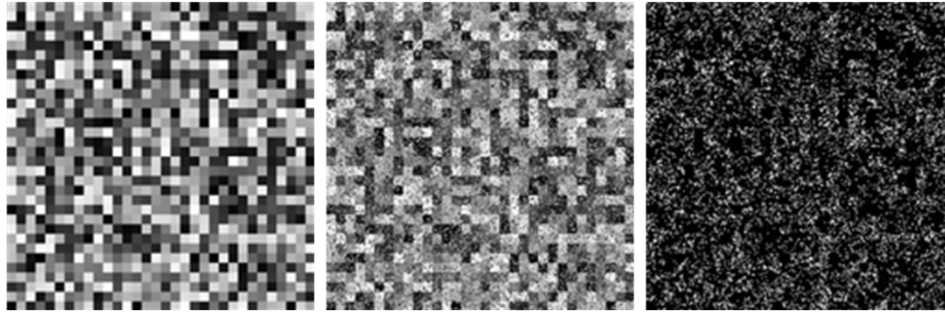
Fig. 3. Training images starting from left:
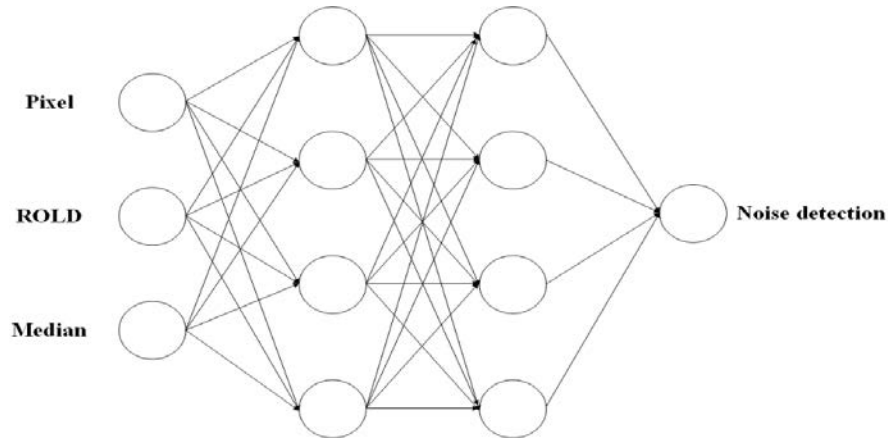(a) original training image, (b) corrupted training image, (c) target training image

Fig. 4. Proposed NN structure for training and testing

3.2 Weighted Couple Group-based Sparse Representation (WSR)

Equation (7) works well when many useful pixel data are available from uncorrupted pixels. However, when the noise density is high, only a few pixel data are available to estimate values for the corrupted pixels. The main assumption of the SR-based noise removal method is that the noisy image and original image share a dictionary, so the original image can be reconstructed using a linear combination of the coefficients. The reconstructed image can be estimated after the coefficients are calculated from the noisy image by using the Hadamard (element-wise) product value of the corresponding dictionary and the coefficients. The final reconstructed image can be obtained using the reconstructed image patches as follows:

$$\widehat{X} = \left(\sum_{i,j} \boldsymbol{R}_{i,j}^T \boldsymbol{R}_{i,j} \right)^{-1} \cdot \sum_{i,j} \boldsymbol{R}_{i,j}^T \boldsymbol{D_G} \boldsymbol{\alpha}_{G,IN,i,j}, \tag{9}$$

where $\widehat{X}$ is the reconstructed image and $\boldsymbol{R}_{i,j}$ denotes an $n \times N$ matrix used to extract the (i,j)-th $\sqrt{n} \times \sqrt{n}$ patch from the image. In most noise removal, that method shows good performance. However, it has several shortcomings. First, because it uses only clean pixels to recover noisy images, performance suffers when noise density is high and the number of clean pixels is small. Second, noise detector accuracy is imperfect, so the detector might provide the wrong location for clean pixels, especially in an RVIN-corrupted image. Some noisy pixels might thus be regarded as clean pixels and be used for sparse coding. Therefore, coefficients

that are inaccurately estimated in the noise removal stage can lead to defects in image reconstruction.

We propose the following method to address those issues. In [20], the authors proposed the coupling method to overcome those issues. Inspired by the fact that the reconstructed image contains fewer noisy pixels than the original noisy image, we simultaneously code the noisy image and the reconstructed image. Our proposed method is formulated as follows:

$$\{\hat{\alpha}_{G,IN,i,j}\} = \arg\min\left\{\sum_{i,j}\frac{1}{2}\left\|R_{i,j}W\otimes\left(R_{i,j}X - D_G\alpha_{G,IN,i,j}\right)\right\|_2^2\right.$$
$$\left. + \sum_{i,j}\frac{1}{2}\left\|R_{i,j}(I-W)\otimes\left(R_{i,j}Y - D_G\alpha_{G,IN,i,j}\right)\right\|_2^2\right\}$$
$$s.t.\left\|\alpha_{G,IN,i,j}\right\|_0 \leq L, \tag{10}$$

where Y denotes the estimation of X and I is a matrix of the same size as W whose entries are all ones. W is a weighting operator, as in (7). Its elements are ones and zeros defined by (8). Equation (10) contains the two parts. The first part selects clean pixels through the element of W in (8) to estimate the coefficients using only clean pixels. If noise density becomes bigger, however, the number of clean pixels is small, so that it is difficult to correctly estimate the coefficients. To solve the difficulty, we add the second part that uses $I - W$ to select noisy pixels, and uses reconstructed image Y instead of X because reconstructed image contains less noise. Therefore, the reconstructed image compensates for the information missing from the noisy image, providing adequate information even if the number of clean pixels is limited by high noise density.

We adopt similar approach in [20]. However, there are two main difference. First, the proposed method use group-based sparse representation for denoising, which is different from patch-based methods in [20]. The group-based method can take advantage of local sparsity and nonlocal similarity in the image. Second, the dictionary learning step is designed with low complexity because similar patches share the same dictionary.

4. Experimental results and discussion

4.1 IN Detection Results

To test our NN-based IN detector, we simulated our method alongside six conventional methods: ACWM [10], SDROM [11], FIDRM [12], FRINRM [13], IAINS [14], and ANN [15]. ACWM and SDROM are median-based IN detectors. FIDRM and FRINRM are fuzzy-based techniques. IAINS and ANN are NN-based methods that use different inputs compared to the proposed method. For the simulation, we used the parameters for each method given in their respective papers. For our method, we used $m = 4$ for ROLD and a 3x3 window for the median value and ran the tests in MATLAB.

Table 1 shows the average comparison between our proposed NND and the conventional methods, simulated using 256×256 *Barbara*, *Hill*, *Peppers*, and *Lena* corrupted by RVIN. With noise densities ranging from 20% to 60%, the results are compared using the false positive rate (α) and false negative rate (β) as:

$$\alpha = \frac{Number\ of\ false\ positives}{Number\ of\ clean\ pixels}, \quad \beta = \frac{Number\ of\ false\ negatives}{Number\ of\ noisy\ pixels} \tag{11}$$

where false positives represent clean pixels classified as noisy and false negatives represent noisy pixels classified as clean. If a method has the smallest value for both α and β, it is considered to be the best method. **Table 1** shows that the β values of our proposed method are lower than those of the conventional methods. Although the α values of our method are similar or slightly larger than those of some methods, the balance between α and β values for our method is better than for any other methods.

Additionally, we show the results for the false alarming ratio (FAR) of each method. FAR is defined as $\mathrm{FAR}(\%) = \frac{N_I}{N_T} \times 100$, where N_I and N_T are the number of incorrectly classified pixels and total pixels, respectively. All methods in the experiment have decreasing performance as the noise density increases. Median filter based methods (ACWM, SDROM) have worst performance since their performance depend on a certain threshold value. Fuzzy based methods (FIDRM, FRINRM) have better performance than the median filter based methods but still the accuracy is lower compared to neural network based methods. Neural network based methods (IAINS, ANN, proposed) have good performance in general, and the proposed method shows the best performance.

Table 1. Average false positive rate (α), false negative rate (β) and FAR results (FAR unit: %)

Method	Noise density								
	20%			40%			60%		
	α	β	FAR	α	β	FAR	α	β	FAR
ACWM	0.790	0.481	22.89	0.241	0.430	32.28	0.492	0.380	42.59
SDROM	0.098	0.511	22.56	0.204	0.447	30.84	0.412	0.377	39.17
FIDRM	0.061	0.182	9.75	0.115	0.209	16.27	0.257	0.319	30.08
FRINRM	0.089	0.158	8.47	0.150	0.194	15.28	0.214	0.224	24.48
IAINS	0.025	0.148	6.16	0.063	0.154	9.99	0.173	0.143	15.51
ANN	0.032	0.116	5.11	0.056	0.123	8.46	0.122	0.132	12.83
Proposed	0.028	0.086	4.53	0.049	0.103	7.31	0.111	0.116	11.42

4.2 IN Removal Results

After IN detection, our WSR filters the corrupted pixels. In our simulation, we used six images to test the IN removal performance. For the noise removal comparison, we used ACWM [10], ANN [15], and conventional GSR [19]. Because GSR was originally designed for white Gaussian noise, we had to use the weighted GSR given in Eq. 7. Also, conventional GSR is designed for deblurring, and does not include noise detection. Therefore, every pixel value in the image will change, which means that clean pixels are degraded by this method. The weighted GSR performs dictionary learning using only clean pixels and recovers corrupted pixels by estimation with the dedicated dictionary and coefficients.

In **Table 2** and **Table 3**, we show the PSNR and SSIM results for the 6 images in **Fig. 5** (*Hill, Lena, Boat, Peppers, Cameraman,* and *Barbara*) with different densities of RVIN, with the best values marked in bold for easy comparison. Clearly, the proposed method achieves PSNR scores that are similar to or better than those of the other three methods. Also, we can see that the proposed method achieves better SSIM scores mostly except 60% noise density case in *Boat* and *Pepper* image. PSNR and SSIM value do not always show the same performance tendency. Nevertheless, we can conclude that the proposed method can achieve

the best performance overall.

Figs. 6 and 7 show the results from the different methods for *Peppers* and *Boat* under 40% noise density. In both figures, ACWM did not suppress enough IN. The ANN and GSR methods removed IN overall; however, they also removed the fine edges. Our proposed method removed the IN and preserved the fine details, which is especially clear in the writing on the boat in Fig. 7.

Table 2. Comparison of reconstruction results in PSNR (Unit: dB)

Image		Noisy	ACWM [10]	ANN [15]	GSR [19]	WSR
Hill	20%	16.07	30.17	30.65	30.87	**31.41**
	40%	13.02	23.11	27.81	28.22	**29.44**
	60%	11.26	17.68	26.24	26.76	**27.19**
Lena	20%	16.21	31.47	32.95	33.38	**34.10**
	40%	13.24	24.86	29.92	30.32	**32.31**
	60%	11.47	17.14	28.01	28.52	**29.14**
Boat	20%	16.31	31.57	30.78	31.14	**32.71**
	40%	13.29	24.98	27.89	28.14	**30.68**
	60%	11.53	17.59	25.63	26.40	**27.41**
Pepper	20%	15.88	30.89	31.18	31.90	**32.17**
	40%	12.95	23.78	27.68	28.41	**29.11**
	60%	11.19	15.45	25.55	**26.32**	26.06
Cameraman	20%	15.35	29.64	30.47	31.16	**31.31**
	40%	12.38	22.15	27.12	27.97	**28.46**
	60%	10.64	16.83	25.30	**26.11**	25.93
Barbara	20%	15.83	30.69	31.78	31.09	**32.53**
	40%	12.82	23.46	28.19	27.35	**29.28**
	60%	11.04	15.32	25.89	24.92	**26.75**

Table 3. Comparison of reconstruction results in SSIM

Image		Noisy	ACWM [10]	ANN [15]	GSR [19]	WSR
Hill	20%	0.205	0.789	0.851	0.865	**0.889**
	40%	0.095	0.478	0.746	0.808	**0.865**
	60%	0.050	0.196	0.589	0.761	**0.763**
Lena	20%	0.231	0.707	0.832	0.909	**0.934**
	40%	0.121	0.318	0.734	0.835	**0.912**
	60%	0.065	0.135	0.556	0.796	**0.851**
Boat	20%	0.260	0.812	0.864	0.873	**0.899**
	40%	0.133	0.449	0.747	0.851	**0.853**
	60%	0.074	0.157	0.549	**0.808**	0.773
Pepper	20%	0.246	0.791	0.820	0.863	**0.879**
	40%	0.130	0.413	0.749	0.812	**0.849**
	60%	0.073	0.163	0.541	**0.727**	0.709
Cameraman	20%	0.224	0.744	0.848	0.869	**0.873**

	40%	0.117	0.387	0.715	0.805	**0.835**
	60%	0.071	0.151	0.572	0.722	**0.764**
Barbara	20%	0.288	0.810	0.881	0.827	**0.891**
	40%	0.148	0.496	0.592	0.797	**0.846**
	60%	0.082	0.215	0.609	0.706	**0.815**

Fig. 5. Test images from left to right: *Hill*, *Lena*, *Boat*, *Peppers*, *Cameraman*, and *Barbara*

Fig. 6. *Peppers* from left to right: ACWM, ANN, GSR, and WSR is applied

Fig. 7. *Boat* from left to right: ACWM, ANN, GSR, and WSR is applied

4.3 Discussion: Convolutional Neural Network

In this research, we developed an NN-based IN detector. In Section 3, we claimed that changing the neural network algorithm would not provide a significant performance increase. In this section, we substantiate that claim by changing our conventional NN into a deep-learning-based NN.

Compared to conventional NNs, convolutional neural networks (CNNs) are a type of deep-learning-based NN with a very deep architecture; they've been found to be effective in many digital image processes [21]. As modern GPUs have become more powerful, CNNs have become popular in many image processing areas, such as segmentation [22], object detection [23], object recognition [24], and noise correction [25]. In this section, we compare a simple NN with a deep-learning-based CNN to show that adopting a deep network is unnecessary for IN detection.

As far as we know, this is the first attempt to use a CNN for IN detection. Therefore, we had to modify some deep-learning-based image processing techniques for our purpose. We investigated many methods and concluded that deep-learning-based denoising methods are a

good choice for the first step. Thus, we modified a deep-learning-based image denoising method for IN detection.

We had many deep-learning-based image denoising methods from which to choose. Most of them deal with white Gaussian noise via a residual learning–based algorithm that uses a residual image instead of a noise-free image. Because the residual image looks similar to our noise location map, we selected the DnCNN residual-learning-based method, which was published in 2017 [25]. Its main purpose is to remove white Gaussian noise from images. It trains a CNN using a noisy image as input and a residual image as output. This method showed state-of-the-art performance compared to other techniques. To adapt the DnCNN for IN detection, we first used a noisy image as input. Then we trained the CNN using a noise location map image, labeling real noise-pixel locations as the output. When we applied an IN-corrupted test image to this network, the algorithm produced images with the noise-pixel locations labeled as output. The image has pixel values between 0 and 1, and values higher than or equal to the threshold value of 0.5 are deemed to be noisy. Finally, we verified whether those pixels accurately matched the locations of the noisy pixels.

For the simulation, we used randomly chosen VOC 2012 images [26] for training and testing. For the training, we used 100 images, and for the testing we used the 12 images shown in **Fig. 8**. We added IN to the images at specific noise density values, and those were converted by the ROLD method for input to the network. **Table 4** shows the DnCNN noise detection result by noise density. The FAR result increases with the noise density. However, if we compare the DnCNN results with the results in **Table 1**, we can see that the results are similar to those from the conventional NN-based methods. If we apply multiple source input that combines pixel values, median values, and ROLD as we did in our method, the DnCNN would yield better results. However, considering the effort involved in redesigning the deep learning network and cost required to train the network, we conclude that a simple NN-based IN detector is adequately efficient and effective.

Fig. 8. Test images randomly chosen from the VOC 12 image set: Simply labeled as Image A, B … L from the top left corner to the bottom right corner.

Table 4. FAR results for DnCNN IN detection with different noise densities (Unit: %)

	Noise density				
	10%	20%	30%	40%	50%
IMAGE A	3.18	6.18	9.05	11.96	15.17
IMAGE B	2.13	4.02	5.72	7.52	9.70
IMAGE C	2.32	4.47	6.71	9.03	11.65
IMAGE D	1.94	3.53	5.15	6.57	8.56
IMAGE E	2.79	5.45	7.89	10.53	13.61
IMAGE F	3.25	5.99	8.59	11.36	14.34
IMAGE G	2.65	4.99	7.52	9.71	12.50
IMAGE H	2.04	3.93	5.85	7.94	10.51

IMAGE I	2.38	4.55	6.77	8.77	11.25
IMAGE J	2.37	4.55	6.83	9.00	11.64
IMAGE K	3.59	6.88	10.14	13.07	16.32
IMAGE L	2.61	4.84	7.03	9.42	12.08
AVERAGE	**2.60**	**4.95**	**7.27**	**9.57**	**12.28**

4.4 Running time

For our experiments, we use Intel Core i7-6700 at 3.4GHz, 16GB memory and MATLAB 2016. The proposed IN detector involves the computation of median and ROLD. However, the time required to compute median and ROLD is negligible compared to the rest of the system. Also, time complexity to apply the neural-network for testing is negligible since it has very small number of nodes and layers. However, when it comes to the noise removal step, we admit that the proposed WSR is very complex and takes about 8 mins. It is because the proposed method is implemented on MATLAB and the code is not fully optimized yet. It is main disadvantage of the proposed method, compared to the other methods.

6. Conclusion

In this paper, we proposed NND-WSR for IN removal. To remove RVIN from images, we used a two-stage method, with IN detection as the first stage and filtering as the second stage. In the first stage, we proposed an IN detector based on an NN using ROLD, median, and pixel values. In the second stage, we proposed the WSR method to remove IN. The WSR overcomes the limitations of GSR and compensate for the insufficient information in images with high noise density by applying a weighted coupling method. Experiment results show the proposed noise detector has the best detection performance in general. In addition, the proposed WSR shows the best results regarding PSNR and SSIM in most cases. We also analyzed NN-based and deep-learning-based noise detectors in the discussion section. FAR results show that deep-learning-based noise detectors give similar or worse performance than NN-based noise detector. In future work, we will investigate an IN removal method robust against noise density.

Acknowledgements

References

[1] Rafael C. Conzalez, Richard E. Woods, "Image Restoration," *Digital Image Processing, Prentice Hall*, pp. 220-276. 2002.

[2] S.-J. Ko and Y. H. Lee, "Center weighted median filters and their applications to image enhancement," *IEEE Trans. Circuits Syst.*, vol. 38, no. 9, pp. 984–993, Sep. 1991. Article (CrossRef Link)

[3] Qin, Chuan, et al. "An inpainting-assisted reversible steganographic scheme using a histogram shifting mechanism," *IEEE Transactions on Circuits and Systems for Video Technology* 23.7, 1109-1118, 2013. Article (CrossRef Link)

[4] Zhang, Xianquan, et al. "Salt and pepper noise removal with image inpainting," *AEU-International Journal of Electronics and Communications* 69.1, 307-313, 2015. Article (CrossRef Link)

[5] Qin, Chuan, et al. "Visible watermark removal scheme based on reversible data hiding and image inpainting," *Signal Processing: Image Communication,* 60, 160-172, 2018. Article (CrossRef Link)

[6] B. Xiong and Z. Yin, "A universal denoising framework with a new impulse detector and nonlocal means," *IEEE Trans. Image Process.*, vol. 21, no. 4, pp. 1663–1675, Apr. 2012. Article (CrossRef Link)

[7] L. Liu, C. L. P. Chen, Y. Zhou, and X. You, "A new weighted mean filter with a two-phase detector for removing impulse noise," *Inf. Sci.*, vol. 315, pp. 1–16, Sep. 2015. Article (CrossRef Link)

[8] R. Garnett, T. Huegerich, C. Chui, and W. He, "A universal noise removal algorithm with an impulse detector," *IEEE Trans. Image Process.*, vol. 14, no. 11, pp. 1747–1754, Nov. 2005. Article (CrossRef Link)

[9] Y. Dong, R. H. Chan, and S. Xu, "A detection statistic for random valued impulse noise," *IEEE Trans. Image Process.*, vol. 16, no. 4, pp. 1112–1120, Apr. 2007. Article (CrossRef Link)

[10] Chen, Tao, and Hong Ren Wu. "Adaptive impulse detection using center-weighted median filters," *IEEE Signal Processing Letters*, vol. 8, no. 1, pp. 1-3, Jan. 2001. Article (CrossRef Link)

[11] Abreu, Eduardo, et al. "A new efficient approach for the removal of impulse noise from highly corrupted images," *IEEE transactions on image processing,* 5.6, pp. 1012-1025, 1996. Article (CrossRef Link)

[12] S. Schulte, M. Nachtegael, V.D. Witte, D. Van der Weken, E.E. Kerre, "A fuzzy impulse noise detection and reduction method," *IEEE Trans. Image Process*, 15, pp. 1153–1162, 2006. Article (CrossRef Link)

[13] S. Schulte, V. De Witte, M. Nachtegael, D. Van der Weken, E.E. Kerre, "Fuzzy random impulse noise reduction method," *Fuzzy Sets Syst*, 158, pp. 270– 283, 2007. Article (CrossRef Link)

[14] P.K. Sa, B. Majhi, "An improved adaptive impulsive noise suppression scheme for digital images," *Int. J. Electron. Commun. (AEÜ)*, 64, 322–328, 2010. Article (CrossRef Link)

[15] Ilke Turkmen, "The ANN based detector to remove random-valued impulse noise in images," *J. Vis. Commun. Image R.*, vol. 34, pp. 28-36, Jan. 2016. Article (CrossRef Link)

[16] A. M. Bruckstein, D. L. Donoho, and M. Elad, "From sparse solutions of systems of equations to sparse modeling of signals and images," *SIAM Rev.*, vol. 51, no. 1, pp. 34–81, 2009. Article (CrossRef Link)

[17] Shao, Ling, et al. "From heuristic optimization to dictionary learning: A review and comprehensive comparison of image denoising algorithms," *IEEE Transactions on Cybernetics* 44.7, pp. 1001-1013, 2014. Article (CrossRef Link)

[18] Dogra, Ayush, Bhawna Goyal, and Sunil Agrawal. "From multi-scale decomposition to non-multi-scale decomposition methods: A comprehensive survey of image fusion techniques and its applications," *IEEE Access* 5, 16040-16067, 2017. Article (CrossRef Link)

[19] Jian Zhang, Debin Zhao, Wen Gao, "Group-Based Sparse Representation for Image Restoration," *IEEE Trans. On Image Process.,* vol. 23, no. 8, pp. 3336-3351, Aug. 2014. Article (CrossRef Link)

[20] Chen, Chun Lung Philip, et al. "Weighted couple sparse representation with classified regularization for impulse noise removal," *IEEE Transactions on Image Processing,* 24.11, pp. 4014-4026, 2015. Article (CrossRef Link)

[21] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. of Computer Vision and Pattern Recongnition.,* 2015, pp. 1–14. Article (CrossRef Link)

[22] Chen, Liang-Chieh, et al. "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," in *Proc. Computer Vision and Pattern Recognition,* 2017. Article (CrossRef Link)

[23] Shin, Hoo-Chang, et al. "Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning," *IEEE transactions on medical imaging* 35.5, pp. 1285-1298, 2016. Article (CrossRef Link)

[24] He, Kaiming, et al. "Deep residual learning for image recognition," in *Proc. of Proceedings of the IEEE conference on computer vision and pattern recognition.* 2016. Article (CrossRef Link)

[25] Zhang, Kai, et al. "Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising," *IEEE Transactions on Image Processing*, 2017. Article (CrossRef Link)

[26] M. Everingham, L. V. Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL visual object classes (VOC) challenge," *International journal of computer vision,* 88.2 303-338, 2010. Article (CrossRef Link)

Yongwoo Lee received his B.S. degree from Sungkyunkwan University in 2013 and is currently a Ph.D. candidate in the Department of Electronic, Electrical and Computer Engineering, College of Information and Communication Engineering, Sungkyunkwan University, Republic of Korea. His research interests include image processing and machine learning, focused on image denoising and object detection.

Toan Duc Bui received a B.S. degree from Hanoi University of Science and Technology, Vietnam, in 2012. He received M.S. and Ph.D. degrees in electrical and computer engineering from Sungkyunkwan University, the Republic of Korea, in 2014 and 2017, respectively. He is working as a post-doc in the Department of Electronic, Electrical and Computer Engineering, College of Information and Communication Engineering, Sungkyunkwan University. His research interests include image processing and machine learning, with a special focus on medical image segmentation, semantic segmentation, and deep learning.
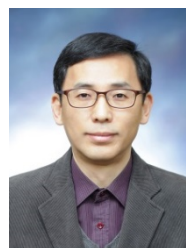
Jitae Shin received his B.S. degree from Seoul National University in 1986 and his M.S. degree from the Korea Advanced Institute of Science and Technology (KAIST) in 1988. After working at Korea Electric Power Corp. (KEPCO) and the Korea Atomic Energy Research Institute (KAERI), he returned to study and received M.S. and Ph.D. degrees in electrical engineering from the University of Southern California, Los Angeles, in 1998 and 2001, respectively. Currently, he is a professor in the School of Electronic and Electrical Engineering of Sungkyunkwan University, Suwon, Republic of Korea. His current research interests include image/video signal processing, video transmission over wireless/mobile communication systems, and deep learning applications.

Byung Tae Oh received the B.S. degree in Electrical Engineering at Yonsei University, Seoul, Korea, in 2003, and the M.S. and Ph.D. degrees in Electrical Engineering at the University of Southern California (USC), LA, CA in 2007 and 2009, respectively. From 2009 to 2013, he was a Research Staff with the Samsung Advanced Institute of Technology (SAIT), Samsung Electronics, Korea. Since 2013, he has been with the School of Electronics and Information Engineering, Korea Aerospace University (KAU), where he is an Associate Professor. His research interests include image/video restoration and compression, and image/video forensics.