BLIND DENOISING OF MIXED GAUSSIAN-IMPULSE NOISE BY SINGLE CNN

Ryo Abiko, Masaaki Ikehara

EEE Dept., Keio Univ., Yokohama, Kanagawa 223-8522, Japan
{abiko, ikehara}@tkhm.elec.keio.ac.jp

ABSTRACT

The removal of mixed noise is a stiff problem since the distribution of the noise cannot be predicted accurately. The most common mixed noise is the combination of Additive White Gaussian Noise (AWGN) and Impulse Noise (IN). Many methods first attempt to remove IN but it might collapse the texture of the image. In this paper, we propose a new learning-based method using convolutional neural network (CNN) for removing mixed gaussian-impulse noise. Since our denoising network can remove various level of mixed noise, neither the preprocessing for removing IN nor noise-level estimation is necessary.

***Index Terms*—** Image denoising, Mixed noise, convolutional neural network, deep learning

1. INTRODUCTION

Digital images are corrupted by noise in the process of data acquisition and transmission. Since the quality of image is affected by these noises, to recover a clean image is a significant work. Image denoising is still a major problem in image processing and it has been studied in wide range of fields [1].

In image capturing system, two types of noises are commonly encountered during the process of data acquisition and transmission. The Additive White Gaussian Noise (AWGN) is usually introduced into images while acquisition. The thermal motion in camera sensors corrupt the image and it affects the all pixels in the image. On the other hand, the Impulse Noise (IN) is often occurred by broken image sensors and the transmission error. The pixels are replaced into random value without regarding of the original value. In practice, the AWGN and the IN often occur simultaneously and it makes the denoising problem complex.

To remove the noise from image without collapsing the texture and edge information, it is important to deal with the appropriate distribution of noise. For example, NLM [2] and BM3D [3] are the major method of removing AWGN. In Addition, DnCNN [4] is a CNN-based method which shows good quality in removing AWGN. As for the removal of IN, AMF [5] is a classical technique still used in many methods [6, 7, 8]. The improved method of AMF is also proposed [9, 10]. Though these methods show good results for single noise removal, they cannot remove mixed noise effectively. This is due to the different distributions of the two noises. Thus, to remove mixed noise is more difficult than to remove single noise.

In this paper, we propose an effective mixed gaussian-impulse noise removal method based on convolutional neural network (CNN). Since many conventional methods use rank order filter (ROF) such as AMF [5] for preprocessing, the quality of denoised image will be affected by the selected ROF. The appropriate filter for ROF varies by the type of the included noises [8] and it cause the lack of robustness in denoising. Thus, our method does not use ROF for preprocessing. In addition, some methods have to estimate the noise level in advance so as to have the best performance but our method can remove mixed noise by a single CNN without any prior information like the noise level. These features are accomplished by training the network with multi-noise level train data. The structure of our CNN and the way of training are remarked in section 3. The experimental results of our method and that of conventional methods are shown in section 4.

2. SUPPORTING METHODS

2.1. Mixed noise model

In this paper, we denote $x(i,j)$ as a noise-free pixel at the location (i,j). The noisy pixel is denoted as $y_{\text{noise}}(i,j)$. When an image is corrupted by AWGN, the noisy pixel $y_{\text{AWGN}}(i,j)$ is modeled as

$$y_{\text{AWGN}}(i,j) = x(i,j) + w(i,j), \tag{1}$$

where $w(i,j)$ is a noise which follows zero mean Gaussian distribution. The noise level can be controlled by the variance σ^2. On the other hand, when the pixel values are bounded by $[r_{\text{min}}, r_{\text{max}}]$, the noisy pixel corrupted by IN is modeled as

$$y_{\text{IN}}(i,j) = \begin{cases} n(i,j) & \text{with probability } p \\ x(i,j) & \text{otherwise} \end{cases}, \tag{2}$$

where $n(i,j) \in [r_{\text{min}}, r_{\text{max}}]$ is a Random Valued Impulse Noise (RVIN), whose density is p. When $n(i,j)$ only takes the values of r_{min} or r_{max}, the noise are especially called Salt-and-Pepper Impulse Noise (SPIN). When the density of SPIN let be s, the model of SPIN can be described as

$$y_{\text{SPIN}}(i,j) = \begin{cases} r_{\text{min}} & \text{with probability } s/2 \\ r_{\text{max}} & \text{with probability } s/2 \\ x(i,j) & \text{otherwise} \end{cases}. \tag{3}$$

Considering the mixed noise composed of AWGN, RVIN and SPIN, the noisy pixel can be modeled as

$$y_{\text{mix}}(i,j) = \begin{cases} n(i,j) & \text{with probability } p \\ r_{\text{min}} & \text{with probability } s/2 \\ r_{\text{max}} & \text{with probability } s/2 \\ x(i,j) + w(i,j) & \text{otherwise} \end{cases}. \tag{4}$$

2.2. Convolutional neural network

Learning-based methods, especially which use Convolutional Neural Network (CNN) have been shown good results in image processing.

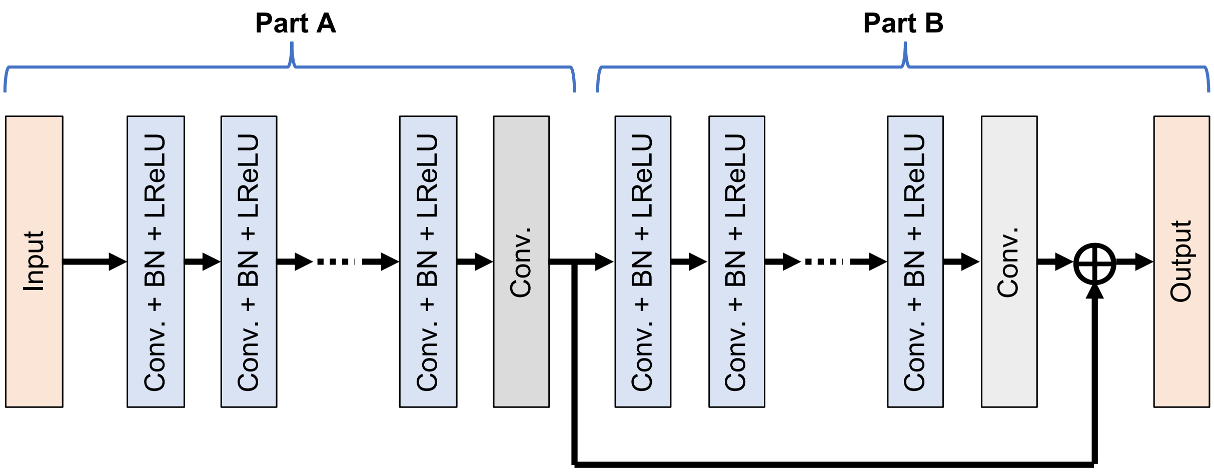

Fig. 1: The architecture of our proposed CNN.

For example, SRCNN [11] is a representative method for super resolution and DnCNN [4] is a method for noise removal.

CNN is usually composed of input layer, hidden layers and output layer. Owing to a trainable feature map in the hidden layer, methods using CNN has an advantage over classical methods in the case of solving non-linear problem. The major layer adopted in hidden layers are convolutional layer and pooling layer [12]. The number of layers affects the result but when too many layers are adopted, the vanishing gradient problem occurs [13]. To avoid this problem, activation function layer [14] and batch normalization layer [15] are often adopted to the network [4, 11, 13, 16].

When the network architecture is expressed as $f(\cdot)$, the aim is to train $f(\cdot)$ which minimize

$$\frac{1}{2|N|}\sum_{i\in N}||f(\mathbf{y_i})-\mathbf{x_i}||^2, \tag{5}$$

where $\mathbf{y_i}$ and $\mathbf{x_i}$ are noisy image and clean image, respectively. i and N are image index and total number of images, respectively [17]. Generally, the network can be expressed as

$$\begin{aligned} &h_0 = \mathbf{y}, \\ &h_l = BN(\phi(\mathbf{W}_l * h_{l-1} + b_{l-1})) \quad (l = 2, ..., L-1), \\ &\mathbf{x} = \mathbf{W}_L * h_{L-1}, \end{aligned} \tag{6}$$

where L is the total number of convolutional layers. $BN(\cdot)$ represents the batch normalization process, which is adopted to speed up training and to get better results [15]. Activation function is expressed in (6) as $\phi(\cdot)$. Sigmoid function [18], Rectified Linear Unit (ReLU) [16] and Leaky ReLU [19] are often used for activation function.

3. PROPOSED METHOD

Our method removes mixed noise by trained-based algorithm using convolutional neural network (CNN). Since training is performed in two steps, each feature of the noises in the mixed noise is effectively trained. By training many kind of noisy images with various noise levels, our method can remove mixed noise without considering the noise level.

3.1. Network Model

We illustrate the proposed CNN architecture in Fig. 1. Our network can be divided into two parts. Considering the first half of our network (part A in Fig. 1), it behaves as an IN removal network and the last half (part B in Fig. 1) behaves as an AWGN removal network.

When we refer the set of convolutional layer, batch normalization layer and leaky ReLU layer [19] as ”ConvUnit”, 25 ConvUnits are adopted in both part A and part B. The filter size of convolutional layer in the ConvUnit is set to 3×3×64 and the coefficient of leaky ReLU layer is set to 0.01. We use 3×3×1 filter for the last convolutional layer of part A and B. Hence, it can be considered that one grayscale image is generated at the each end of part A and B. In addition, skip connection is applied to part B. This is because residual learning shows good result for AWGN removal [4].

3.2. Training

In order to train the characteristic of AWGN and IN explicitly, we first train only part A in Fig. 1. Mean squared error function is attached after the last convolutional layer in part A and Adam solver [20] is used to minimize (5). The noisy image corrupted with AWGN, RVIN and SPIN is inserted into input layer and the noisy image corrupted with AWGN is inserted into loss function. This implies that part A acts like an IN removal network. After training part A, part B is attached after part A instead of loss function. Before training the whole network, the learning rate of part A was set to 0. Now, the noisy image corrupted with AWGN ,RVIN and SPIN is inserted into input layer and the clean image is inserted into loss function which is located in the last layer of part B. Finally, the training is performed on the end to end of the structure shown in Fig. 1 using Adam solver.

To create the training dataset, 100 images were randomly chosen from Microsoft COCO dataset [21] and they were converted into grayscale image. The images are corrupted by mixed noise using (4). The noise levels of each noises are as follows: AWGN (σ = {0, 10, 20, 30, 40, 50}), RVIN (p = {0, 5, 10, 15, 20, 25, 30, 35, 40, 45} / 100) and SPIN (s is randomly set in the range of [0 , 0.40]). Thus, our training dataset includes $100 \times 6 \times 10 = 6000$ pairs of noisy and clean images. The patch size is 33×33 and the batch size for the training is set to 256. The initial learning rate is set to 0.00001 and the epoch number is 10. It takes about 30 hours to train network

(a) Lena (b) Barbara (c) Bridge (d) Boat (e) Airplane (f) Pepper (g) Hill

Fig. 2: Seven test grayscale images. The size of the images are 512×512.

Table 1: Comparison on restoration result in PSNR. The images are corrupted by RVIN + AWGN ($\sigma = 15$). [7] and proposed method are blind denoising methods and else are non-blind denoising methods.

Image 512×512	p	Method			
		[22]+[3]	[7]	[8]	Proposed
Lena	15%	32.41	32.06	32.28	**32.56**
	30%	30.25	30.27	29.10	**31.71**
	45%	26.65	28.09	24.87	**30.36**
Barbara	15%	26.70	27.60	25.67	**29.43**
	30%	24.79	25.62	24.17	**28.32**
	45%	22.59	22.79	21.67	**26.25**
Bridge	15%	**29.60**	29.51	28.86	29.16
	30%	27.16	27.67	26.54	**28.09**
	45%	24.02	22.44	22.86	**26.53**
Boat	15%	29.65	29.16	29.12	**30.30**
	30%	27.55	27.72	27.02	**29.19**
	45%	24.78	25.17	23.62	**27.60**
Airplane	15%	33.42	**33.66**	32.52	32.87
	30%	30.36	31.79	28.81	**31.88**
	45%	25.51	26.73	23.32	**30.44**
Pepper	15%	34.94	**35.02**	34.07	33.49
	30%	31.60	32.26	29.90	**33.36**
	45%	26.71	28.73	24.54	**32.09**
Hill	15%	**32.51**	32.30	31.61	31.73
	30%	30.36	30.40	29.04	**30.98**
	45%	26.56	27.65	24.69	**29.86**
BSDS300	15%	27.64	27.03	27.70	**28.51**
	30%	25.76	25.78	25.29	**27.30**
	45%	22.88	23.59	22.13	**25.87**

Table 2: Comparison on restoration result in PSNR. The images are corrupted by RVIN + AWGN ($\sigma = 25$). [7] and proposed method are blind denoising methods and else are non-blind denoising methods.

Image 512×512	p	Method			
		[22]+[3]	[7]	[8]	Proposed
Lena	15%	29.87	29.81	29.87	**30.46**
	30%	28.10	28.41	27.93	**29.79**
	45%	25.33	26.32	24.88	**28.54**
Barbara	15%	24.91	24.87	24.52	**27.28**
	30%	23.57	23.58	23.36	**26.35**
	45%	21.91	22.04	21.65	**24.75**
Bridge	15%	**26.76**	26.37	26.61	26.72
	30%	25.32	25.31	25.10	**25.97**
	45%	22.98	21.40	22.62	**24.81**
Boat	15%	27.47	27.11	27.61	**28.31**
	30%	26.01	26.09	26.08	**27.39**
	45%	23.70	24.30	23.45	**26.10**
Airplane	15%	30.44	30.62	30.34	**30.73**
	30%	28.06	28.55	28.00	**29.90**
	45%	24.31	24.51	23.97	**28.40**
Pepper	15%	31.65	**31.79**	31.65	31.76
	30%	28.95	29.84	28.70	**31.30**
	45%	25.36	26.56	24.87	**30.04**
Hill	15%	**29.70**	29.30	29.48	29.55
	30%	28.09	28.41	27.80	**28.99**
	45%	25.16	26.10	24.59	**28.01**
BSDS300	15%	25.74	25.42	25.97	**26.83**
	30%	24.35	24.34	24.42	**25.95**
	45%	22.10	22.78	21.89	**24.80**

with our MATLAB implementation on single GeForce GTX 1080 Ti. Full size image is inserted to the network when the prediction is performed.

4. EXPERIMENTAL RESULTS

In this section, we compared our CNN-based method with the other notable methods, including ACWMF [22] + BM3D [3], WSR [7] (preprocessed by AMF or ACWMF) and CNN based method [8] (preprocessed by AMF, ACWMF or Cai's method [6]). Seven 512×512 images including $\{Lena, Barbara, Bridge, Boat, Airplane, Pepper, Hill\}$ and 100 test images from BSDS300 [23] are used for the comparison. Since BSDS300 contains only color image, we converted them into grayscale image before using for the comparison. We used PSNR [24] to assess the performance. AMF and ACWMF are implemented by ourselves, while BM3D, WSR and method in [8] are implemented by the original authors.

4.1. Results

Table 1 shows the comparison on restoration result in PSNR with $\sigma = 15, p = \{15, 30, 45\}/100$ and Table 2 shows the comparison on restoration result in PSNR with $\sigma = 25, p = \{15, 30, 45\}/100$. From Table 1 and 2, we can see that our proposed method achieves much higher PSNR than conventional methods in almost all the images. In particular, our method shows good results when the images are corrupted by strong mixed noises. This is because our method does not use IN detection. When the noise level is strong, IN detection will be a difficult problem and the accuracy becomes bad. Since the other methods strongly depend on the result of the IN detection, the restoration results when the noise level is high tend to be incomplete. The restoration results are shown in Fig. 3. We can see that our proposed method restores the texture and edge information efficiently while the conventional method cannot restore them.

(a) (b) (c)

(d) (e) (f)

Fig. 3: Denoising results of $Barbara$ corrupted with AWGN + SPIN ($\sigma = 15, p = 30$). (a) Clean image. (b) Noisy image. (PSNR: 12.82dB) (c) ACWMF [22] + BM3D [3]. (PSNR: 24.81dB) (d) WSR [7]. (PSNR: 25.62dB) (e) Method in [8]. (PSNR: 24.15dB) (f) Proposed method. (PSNR: 28.32dB)

4.2. Implementation time

Since our method is based on deep CNN, we suppose that the implementation is run on GPU. Our execution environment is Intel Xeon CPU E5-1650 v4 @3.60GHz, 64G RAM and GeForce GTX 1080 Ti. It takes 0.28 seconds to process 512 $\times$ 512 grayscale image on single GPU and 0.76 seconds for 1M pixels image. When we use CPU for the execution, the running time for 512 $\times$ 512 grayscale image was 8.14 seconds and 20.79 seconds for 1M pixels image.

4.3. The effect of training in two stages

Our proposed network is trained in two stages as remarked in Section 3.2. To show the validity of our training method, we trained the same network without dividing into two stages. The comparison of the restoration result is shown in Table 3. We can see that the proposed training method achieves higher PSNR than the training method without divison in all cases. It implies that considering the types of noise is efficacious even if the method is based on CNN.

Table 3: Comparison of the denoising result in PSNR ($\sigma = 25$). Different training method was applied to each method. Test images used in Table. 1 and BSDS300 [23] are used for the validation. The average PSNR is shown in the table.

Image	p	Training method	
		Proposed	Without division
Test image	15%	**29.24**	27.72
	30%	**28.54**	26.82
	45%	**27.22**	25.46
BSDS300	15%	**26.83**	25.84
	30%	**25.95**	24.93
	45%	**24.80**	23.74

5. CONCLUSION

In this paper, we propose a new cnn-based method to remove the mixed Gaussian-impulse noise from images. Since our trained network can remove mixed noise which has various noise level, many problems have been overcome. First, the noise level estimation is not necessary in our method. This feature is remarkable because classical methods (e.g., BM3D) cannot remove noises effectively without estimating the noise level. Second, our method can remove mixed noise in a single trained model and it is important to improve the robustness against the corrupted images. Many conventional methods use impulse noise detector (e.g., AMF, ACWMF) but the denoised images are strongly affected by the result of IN detection. Thus, our method, which does not use IN detector, can remove strong mixed noise more effectively than the conventional methods which use IN detector. Experimental results show that our method can remove mixed noise more effectively than the conventional methods.

6. REFERENCES

[1] 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, June 1996.

[2] Antoni Buades, Bartomeu Coll, and J-M Morel, “A non-local algorithm for image denoising,” in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*. IEEE, 2005, vol. 2, pp. 60–65.

[3] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian, “Image denoising by sparse 3-d transform-domain collaborative filtering,” *IEEE Transactions on image processing*, vol. 16, no. 8, pp. 2080–2095, 2007.

[4] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang, “Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising,” *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3142–3155, 2017.

[5] Humor Hwang and Richard A Haddad, “Adaptive median filters: new algorithms and results,” *IEEE Transactions on image processing*, vol. 4, no. 4, pp. 499–502, 1995.

[6] Jian-Feng Cai, Raymond H Chan, and Mila Nikolova, “Fast two-phase image deblurring under impulse noise,” *Journal of Mathematical Imaging and Vision*, vol. 36, no. 1, pp. 46, 2010.

[7] Tao Huang, Weisheng Dong, Xuemei Xie, Guangming Shi, and Xiang Bai, “Mixed noise removal via laplacian scale mixture modeling and nonlocal low-rank approximation,” *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3171–3186, 2017.

[8] Mohammad Tariqul Islam, SM Mahbubur Rahman, M Omair Ahmad, and MNS Swamy, “Mixed gaussian-impulse noise reduction from images using convolutional neural network,” *Signal Processing: Image Communication*, vol. 68, pp. 26–41, 2018.

[9] Yiqiu Dong and Shufang Xu, “A new directional weighted median filter for removal of random-valued impulse noise,” *IEEE signal processing letters*, vol. 14, no. 3, pp. 193–196, 2007.

[10] Amarjit Roy, Joyeeta Singha, Lalit Manam, and Rabul Hussain Laskar, “Combination of adaptive vector median filter and weighted mean filter for removal of high-density impulse noise from colour images,” *IET Image Processing*, vol. 11, no. 6, pp. 352–361, 2017.

[11] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang, “Image super-resolution using deep convolutional networks,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 2, pp. 295–307, 2016.

[12] Dominik Scherer, Andreas Müller, and Sven Behnke, “Evaluation of pooling operations in convolutional architectures for object recognition,” in *Artificial Neural Networks–ICANN 2010*, pp. 92–101. Springer, 2010.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[14] Donald F Specht, “Probabilistic neural networks,” *Neural networks*, vol. 3, no. 1, pp. 109–118, 1990.

[15] Sergey Ioffe and Christian Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” *arXiv preprint arXiv:1502.03167*, 2015.

[16] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[17] Li Xu, Jimmy SJ Ren, Ce Liu, and Jiaya Jia, “Deep convolutional neural network for image deconvolution,” in *Advances in Neural Information Processing Systems*, 2014, pp. 1790–1798.

[18] George Cybenko, “Approximation by superpositions of a sigmoidal function,” *Mathematics of control, signals and systems*, vol. 2, no. 4, pp. 303–314, 1989.

[19] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller, “Striving for simplicity: The all convolutional net,” *arXiv preprint arXiv:1412.6806*, 2014.

[20] Diederik P Kingma and Jimmy Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.

[21] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick, “Microsoft coco: Common objects in context,” in *European conference on computer vision*. Springer, 2014, pp. 740–755.

[22] Tao Chen and Hong Ren Wu, “Adaptive impulse detection using center-weighted median filters,” *IEEE Signal Processing Letters*, vol. 8, no. 1, pp. 1–3, 2001.

[23] D. Martin, C. Fowlkes, D. Tal, and J. Malik, “A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics,” in *Proc. 8th Int’l Conf. Computer Vision*, July 2001, vol. 2, pp. 416–423.

[24] Alain Hore and Djemel Ziou, “Image quality metrics: Psnr vs. ssim,” in *Pattern recognition (icpr), 2010 20th international conference on*. IEEE, 2010, pp. 2366–2369.