Removing Impulsive Noise from Color Images via a Residual Deep Neural Network Enhanced by Post-Processing

Sahar Sadrizadeh, Hatef Otroshi-Shahreza, and Farokh Marvasti

***Abstract*—Impulsive noise is a common type of noise that affects gray-scale/color images and videos. In this paper, we present a residual fully Convolutional Neural Network (CNN) to remove impulsive noise from color images in an end-to-end fashion. We train our residual CNN model on a customized dataset which contains noisy images with different impulsive noise density. The proposed dataset omits the need for multiple models for different noise densities. Moreover, we employ a multi-term loss function to train our model. One of the terms of the proposed loss function imposes the sparsity of the impulsive noise in the observation domain. To the best of our knowledge, this term has not been employed as a loss function for training a denoising CNN. Finally, we employ an iterative post-processing stage to further improve the performance of our method. Simulation results demonstrate that the proposed approach outperforms other notable algorithms in the literature. Furthermore, our method is quite fast, especially when implemented on a GPU-equipped system.**

***Index Terms*—Color image, convolutional neural network, impulsive noise, loss function, sparsity.**

I. Introduction

By the rapid growth of digital images and videos, image denoising still remains a challenging task. Digital images and videos are often degraded by impulsive noise which is caused by faulty sensors, transmission errors, and memory loss [1]. Before any further processing such as object tracking, and object detection, it is necessary to restore the images by removing this type of noise. Impulsive noise, in contrast to the Gaussian noise, affects only some random pixels of the images such that the values of the noisy pixels are independent of the original values. There are two common types of impulsive noise based on the values of the noisy pixels, Salt-and-Pepper Noise (SPN) and Random-Valued Impulsive Noise (RVIN). The noisy pixels can take the maximum or the minimum value of the dynamic range of an image in the case of SPN. However, RVIN changes the value of noisy pixels to a random value in the whole intensity range of the image. When color images are corrupted by impulsive noise, two scenarios are feasible [2]. On the one hand, impulsive noise may affect the color channels independently, i.e., each channel is degraded like a gray-scale image. On the other hand, the locations of the noisy pixels may be identical for all three color channels.

Authors are with Electrical Engineering Dept., Sharif University of Technology, Tehran, Iran. Emails: {sahar.sadrizadeh, hatef.otroshi}@ee.sharif.edu, marvasti@sharif.edu

(a) 11.44dB (b) 34.23 dB (c) 34.31 dB

Fig. 1: Color *Parrots* image corrupted by 50% RVIN. (a) Noisy image, (b) Output of the proposed CNN, and (c) Output of the post-processing.

Various methods have been proposed to remove impulsive noise from color images. Some of them are component-wise which process each color channel separately. One can simply extend the methods proposed for gray-scale image denoising to color images. However, these methods may produce some artefacts to the restored color image since the correlation of the color channels is neglected [3]. In contrast, vector-processing approaches outperform the component-wise ones by treating color images as a vector field [4].

Vector median filter is the first classical vector filter for suppressing impulsive noise in color images [3]. In spite of its low complexity, the texture and the edges of the reconstructed image are blurred. By detecting the noisy pixels, switching vector filters tries to reduce the blurriness [5], [6], [7], [8]. A great number of algorithms in the literature employ an impulse detector as their first stage. These impulse detectors may be based on the fuzzy logic [9], machine/deep learning [10], [11], [12], or thresholding according to a defined metric[13]. However, the accuracy of the impulse detectors highly affects the final performance of the denoising algorithm. Therefore, other algorithms remove the impulsive noise without a detection stage. These methods utilize different approaches such as sparse representation [14], [15], [16], [17], sparse and low-rank decomposition [18], [19], [2], and maximum likelihood [20]. In recent years, deep-learning-based methods, especially Convolutional Neural Networks (CNNs), have achieved high performances in different applications of computer vision [21], [22]. Several papers apply Deep Neural Networks (DNNs) to detect the pixels contaminated with impulsive noise [11], [12], while some of them remove the impulsive noise in an end-to-

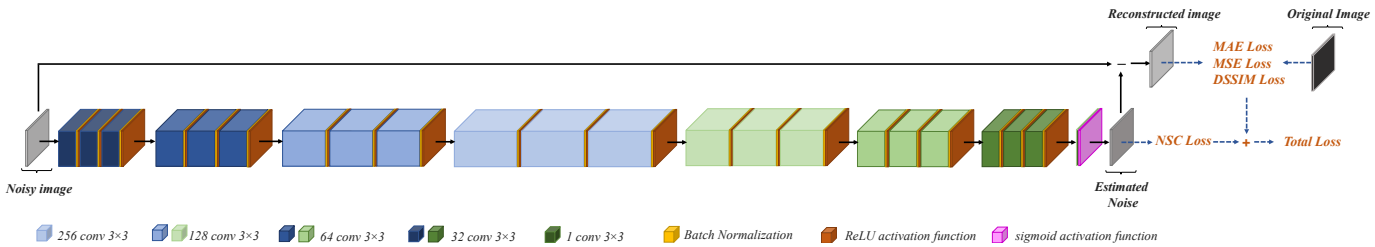

Fig. 2: Network structure

end fashion [23].

In this paper, we propose a fully convolutional DNN model to reconstruct the color images degraded by RVIN. Our method removes the noise in an end-to-end fashion. Therefore, no pre-processing for impulse detection or noise percentage estimation is required. Moreover, our dataset consists of noisy images contaminated with multi-level impulsive noise percentage such the noisier images are more frequent. This dataset omits the need for multiple neural networks and improves the reconstruction quality. Furthermore, the proposed multi-term loss function helps the training procedure and hence enhances the objective and subjective reconstruction quality. Moreover, we utilize the output of the network in a post-processing stage to further enhance the reconstruction quality. Figure 1 depicts a sample of noisy image and the reconstructed results of our proposed CNN and the post-processing stage.

The rest of this paper is organized as follows. In Section II, we explain the proposed method, including the architecture of the network, the multi-term loss function, the training procedure, and the proposed post-processing stage. The experimental results are discussed in Section III. Finally, the paper is concluded in Section IV.

II. The Proposed Method

Our proposed residual deep CNN model, as depicted in figure 2, gets a noisy image (I_n) as an input and recovers the original image (I_{org}) by estimating the impulsive noise in the output ($N = I_n - I_{org}$). To estimate the noise in the output, we train our CNN model, $F_{\mathbf{W}}(.)$, with weights $\mathbf{W}$ by optimizing the following loss function:

$$\mathbf{W}^* = \arg\min_{\mathbf{W}} \frac{1}{m} \sum_{j=1}^{m} \mathcal{L}(F_{\mathbf{W}}(I_{\mathrm{n}}^j), N^j), \tag{1}$$

where $\mathcal{L}$ is a multi-term loss function which is calculated between the output of the neural network and the impulsive noise contaminated the original image. We describe our multi-term loss function in section II-B. We train our network with a generated customized dataset which is described in section II-A.

A. Customized Dataset

To generate the dataset for training our CNN model, we randomly chose 12,000 256×256 color images from Places2 Database [24] as the original images. To generate the noisy images of the dataset, each color image is then corrupted by a random pattern of RVIN:

$$\forall c \in \{1, 2, 3\} :$$
$$(I_{\mathrm{n}})_{i,j,c} = \begin{cases} n_{i,j,c} & \text{with probability } p \\ (I_{\mathrm{org}})_{i,j,c} & \text{otherwise} \end{cases}, \tag{2}$$

where $n_{i,j,c}$ is a random variable in the intensity range of the image (zero to 255 for 8-bit images) with uniform distribution. Moreover, p represents the density of the impulsive noise, and c represents each color channel, which means each channel is corrupted by impulsive noise like a separate gray-scale image.

Instead of training our CNN models on the whole 256×256 image, we extract 64×64 patches from the original and noisy images. By doing so, we can reduce the required memory for training the DNN models. Therefore, a total number of $12,000 \times 16 = 192,000$ pairs of original and noisy patches are produced, 80% of which are utilized for training and the rest are used for validation.

We aim to train a single blind CNN model which can remove impulsive noise with different densities. Our dataset contains noisy images with various impulsive noise density (p) in the $[0.1, 0.7]$. In other methods, the distribution of the impulsive noise density over the dataset is uniform. However, in our generated dataset, noisy images corrupted by higher densities of impulsive noise are more frequent so that the reconstruction quality of noisier images improves. The distribution of impulsive noise density (p) over our dataset is $\mathcal{N}(0.6, 0.3)$ truncated to the $[0.1, 0.7]$.

B. Multi-term Loss Function

As opposed to other CNN-based impulsive denoisers which use Mean Square Error (MSE) as their loss function, we introduce a multi-term loss function for training our CNN models. One of the terms of our loss function imposes the sparsity of the impulsive noise, Noise Sparsity Constraint (NSC). Our multi-term loss function is a weighted sum of MSE, Mean Absolute Error (MAE), Dissimilarity Structural Index Metric (DSSIM), and NSC as follows:

$$\mathcal{L}_{\text{Total}} = \mathcal{L}_{\text{MSE}} + 0.1\ \mathcal{L}_{\text{MAE}} + 0.04\ \mathcal{L}_{\text{DSSIM}} + 0.005\ \mathcal{L}_{\text{NSC}}. \tag{3}$$

It is worth mentioning that coefficients of each term of the loss function are selected based on preliminary experiments such that the importance of each term (considering its weight) is the same in the total loss function.

1) **MSE**: MSE is a common loss function since its minimization results in maximizing Peak Signal to Noise Ratio (PSNR), which is a popular metric for assessing the reconstruction quality of images:

$$\mathcal{L}_{\text{MSE}} = \sum_j (F_{\mathbf{W}}(I_{\text{n}}^k) - N^j)^2. \tag{4}$$

2) **MAE**: Since the effect of small errors in the MSE loss function is low, we add MAE to our loss function to highlight the effect of small errors:

$$\mathcal{L}_{\text{MAE}} = \sum_j |F_{\mathbf{W}}(I_{\text{n}}^j) - N^j|. \tag{5}$$

3) **DSSIM**: MSE and MAE are both objective metrics. To consider a subjective metric we add DSSIM to our loss function.

$$\mathcal{L}_{\text{DSSIM}} = \sum_j \frac{1 - \text{SSIM}(I_{\text{n}}^j - F_{\mathbf{W}}(I_{\text{n}}^j), I_{\text{org}}^j)}{2}. \tag{6}$$

4) **NSC**: Impulsive noise is sparse in the observation domain since a few of the pixels of the image corrupted by impulsive noise. Since the impulsive noise affects the color channels separately, we add the l_1 norm, the relaxation of the l_0 norm, of the estimated noise to apply the sparsity of the noise to the loss function.

$$\mathcal{L}_{\text{NSC}} = \sum_j |F_{\mathbf{W}}(I_{\text{n}}^j)|. \tag{7}$$

C. Network Structure

Our proposed blind residual DNN is fully convolutional which reconstructs the impulsive noise, and hence, the original image in an end-to-end fashion. Figure 2 represents the structure of our CNN model. In our network, the number of channels increases from 3 to 256 and then decreases symmetrically. By this choice, more features are extracted by deeper layers of the network, which results in the improvement of the reconstruction quality. In the proposed neural network, each convolutional layer is proceeded by batch normalization [25] and *ReLU* activation function. However, the activation function of the last layer is *tanh* since our network is residual, and its output had positive and negative values.

D. Training Procedure

To optimize the parameters of the network, we apply Adam optimizer to batches of size 32 of the images. The learning rate of the optimizer is divided by 10 if the validation performance does not improve for 10 epochs, and its initial value is set to 10^{-3}. Moreover, we stop the training procedure when the validation loss does not decrease for 25 epochs (early stopping technique). The training procedure ends after 82 epochs which takes about 27 hours with Keras implementation on a system equipped with 64-GB RAM, an Intel CORE i7 − 7700 CPU 3.60 GHz, and an NVIDIA GeForce GTX 960 GPU.

Fig. 3: 32 natural, *Lena*, and *Parrots* images used for assessing the performance of our method

E. Post-processing

In order to further improve the reconstruction quality, we employ a novel iterative method for the post-processing step. The reconstructed image by the proposed network is utilized for the initial estimation of Algorithm 1, as well as for estimating the positions of noisy pixels. To derive the input binary mask M with zeros in the locations of noisy pixels of I_n, we can use:

$$M_{i,j} = \begin{cases} 1, & if \ \ \exp(-|F_{\mathbf{W}}(I_{\text{n}})_{i,j}|) > 10^{-7} \\ 0, & o.w. \end{cases}. \tag{8}$$

Then, as presented in Algorithm 1, we utilize M in a simple non-uniform sampling iterative method [26] with an exponential filter [27]. Exponential filtering is using the smoothing kernel $-1.2\exp(-|x|^{1.2})$ with kernel of size 5 to highlight the low-frequency components of the image.

Algorithm 1 Post-processing

Input:
  Noisy Image: I_{n}, Initial estimation: I^0, Binary impulse mask: M, Max. number of iterations: $k_{max} = 20$, Threshold: $\delta = 0.02$
Output:
  Recovered estimate of the image: $\hat{I}$
procedure
  for $k = 1 : k_{max}$ **do**
    $I^k \leftarrow I^{k-1} +$ exponential-filter$((I_{\text{n}} - I^{k-1}) \odot M)$
    if $\|I^k - I^{k-1}\|_F < \delta$ **then**
      break
    end if
  end for
  return $\hat{I} \leftarrow I^k$
end procedure

III. Experimental Results

In this section, we will evaluate the performance of our algorithm in comparison with other notable methods, including classical VMF [3], AVMF [5], FPGF [6], VF-AQCD [28], ALOHA [2], and the method of [12]. *Lena* and *Parrot* color images of size 256×256, and 32 natural images, which are depicted in Figure 3, are utilized for the comparison. The results for other methods are all extracted from [12].

A. Performance Comparison

The results in terms of PSNR, Normalized Color Difference (NCD), and FSIM for various impulsive noise densities are

TABLE I: PSNR and FSIM for different RVIN densities (In each case, the best two results are embolded)

		PSNR					NCD					FSIM				
	p	**10%**	**20%**	**30%**	**40%**	**50%**	**10%**	**20%**	**30%**	**40%**	**50%**	**10%**	**20%**	**30%**	**40%**	**50%**
Lena	**VMF**	29.38	24.20	19.75	19.15	16.41	0.0250	0.0516	0.1077	0.1398	0.2258	0.9441	0.8732	0.7643	0.7075	0.628
	AVMF	29.77	24.05	22.80	19.40	16.55	0.0195	0.0534	0.0734	0.1337	0.2192	0.9613	0.8805	0.8309	0.7306	0.6392
	FPGF	30.32	24.34	23.06	19.48	16.59	0.0162	0.0472	0.0668	0.1270	0.2139	0.9653	0.8846	0.8482	0.7417	0.6405
	VF-AQCD	33.83	28.92	25.05	21.34	17.32	0.0095	0.0229	0.0439	0.0842	0.1725	0.9803	0.9409	0.8718	0.7599	0.6084
	ALOHA	32.57	28.36	24.96	19.13	16.04	0.0202	0.0464	0.0761	0.1699	0.2533	0.9749	0.9372	0.8995	0.7744	0.6569
	[12]	**43.07**	**39.02**	35.46	32.68	29.40	0.0042	0.0084	0.0132	0.0191	0.0291	**0.9973**	0.9932	0.9863	0.9760	0.9528
	Our CNN	41.61	38.04	**36.20**	**34.62**	**33.27**	**0.0021**	**0.0024**	**0.0025**	**0.0026**	**0.0028**	0.9964	**0.9940**	**0.9908**	**0.9861**	**0.9793**
	Our CNN + Post-processing	**43.12**	**38.93**	**36.53**	**34.70**	**33.30**	**0.0019**	**0.0022**	**0.0023**	**0.0025**	**0.0028**	**0.9989**	**0.9960**	**0.9917**	**0.9868**	**0.9796**
Parrots	**VMF**	27.71	23.75	19.30	18.37	15.85	0.0236	0.0468	0.1071	0.142	0.2262	0.9443	0.8791	0.7432	0.7078	0.6297
	AVMF	27.86	23.55	22.15	18.53	15.91	0.0208	0.0509	0.0708	0.1389	0.2237	0.9479	0.8666	0.8216	0.7107	0.6288
	FPGF	28.25	23.83	22.45	18.73	16.05	0.0170	0.0438	0.0647	0.1301	0.2145	0.9581	0.8819	0.8412	0.7238	0.6321
	VF-AQCD	32.02	27.69	24.25	20.38	16.69	0.0091	0.0209	0.0413	0.0899	0.1769	0.9796	0.9441	0.8788	0.7644	0.6443
	ALOHA	31.15	27.29	23.42	18.64	15.90	0.0213	0.0494	0.0787	0.1626	0.2483	0.9639	0.9211	0.8808	0.7604	0.6425
	[12]	39.07	35.23	32.35	29.65	27.09	0.0051	0.0092	0.0133	0.0182	0.0244	0.9950	0.9894	0.9818	0.9687	0.9510
	Our CNN	**44.65**	**41.18**	**39.78**	**36.01**	**34.23**	**0.0018**	**0.0020**	**0.0025**	**0.0025**	**0.0028**	**0.9993**	**0.9979**	**0.9962**	**0.9921**	**0.9863**
	Our CNN + Post-processing	**45.67**	**41.80**	**39.97**	**36.09**	**34.31**	**0.0016**	**0.0018**	**0.0024**	**0.0024**	**0.0028**	**0.9995**	**0.9980**	**0.9963**	**0.9921**	**0.9863**
32 images	**VMF**	29.33	22.63	17.60	16.41	13.89	0.0446	0.0966	0.2131	0.3043	0.4228	0.9760	0.8938	0.7603	0.6977	0.6268
	AVMF	29.53	22.66	20.99	16.65	13.96	0.0289	0.0868	0.1395	0.2821	0.4131	0.9809	0.9013	0.8407	0.7090	0.6287
	FPGF	30.09	22.76	21.29	16.99	14.19	0.0262	0.0844	0.1280	0.2583	0.3885	0.9818	0.8979	0.8557	0.7294	0.6374
	VF-AQCD	33.29	28.86	24.34	19.31	15.16	0.0175	0.0385	0.0792	0.1844	0.3402	0.9906	0.9701	0.9069	0.7627	0.6265
	ALOHA	29.69	23.65	20.84	16.50	14.07	0.0499	0.1296	0.2008	0.3294	0.4284	0.9772	0.9190	0.8771	0.7888	0.7122
	[12]	39.26	35.89	33.27	30.84	28.30	0.0122	0.0222	0.0324	0.0433	0.0569	0.9971	0.9935	0.9877	0.9781	0.9606
	Our CNN	**42.48**	**39.88**	**37.75**	**35.76**	**33.64**	**0.0022**	**0.0026**	**0.0029**	**0.0032**	**0.0038**	**0.9987**	**0.9978**	**0.9964**	**0.9941**	**0.9901**
	Our CNN + Post-processing	**43.56**	**40.49**	**38.02**	**35.91**	**33.67**	**0.0020**	**0.0024**	**0.0028**	**0.0032**	**0.0038**	**0.9993**	**0.9986**	**0.9970**	**0.9947**	**0.9903**

reported in Table I. It is worth noting that the color difference is calculated in the CIELAB color space, which is perceptually uniform. The results demonstrate that our method outperforms other algorithms in all cases, especially at higher noise densities. The second best method is [12], which utilizes two CNNs for detecting and reconstructing the images. However, our end-to-end model has better performance with less complexity since the accuracy of the detection stage of [12] highly affects its final performance. The proposed multi-term loss function and the generated dataset may also be responsible for the superiority of our method. The post-processing stage also improves the performance of our CNN model, specifically when the impulsive noise density is low.

Moreover, Figure 1 depicts the output of our algorithm for the *Parrots* image corrupted by various noise densities. As can be seen, our CNN is capable of removing impulsive noise while retaining the texture and edges of the original image even for high densities of impulsive noise.

B. Run-Time

We compare the complexity of different algorithm based on their run-time for denoising 256×256 *Lena* image. Table II represents the run-time of different algorithms on CPU or GPU. The operated hardware is determined in the parentheses. The run-times of VMF, AVMF, and FPGF algorithms are not reported due to their simplicity and poor performance. According to this table, our algorithm is faster than other methods, even on CPU, and its run-time is very low on GPU. It is worth mentioning that the proposed post-processing stage takes only 0.02 seconds on CPU while improving the performance of our CNN model.

TABLE II: Run-time (seconds) for denoising *Lena* image (The best results are embolded)

p	**10%**	**20%**	**30%**	**40%**	**50%**
VF-AQCD (CPU)	26.98	27.03	26.96	27.03	26.89
ALOHA (GPU)	865.12	699.44	532.50	577.65	605.04
[12] (CPU)	3.96	3.91	3.97	3.89	3.91
[12] (GPU)	0.37	0.36	0.37	0.37	0.36
Our CNN (CPU)	**1.98**	**1.98**	**1.95**	**1.84**	**1.84**
Our CNN (GPU)	**0.19**	**0.19**	**0.18**	**0.17**	**0.17**

IV. Conclusion

In this paper, we proposed a fully-convolutional residual neural network to remove impulsive noise from color images in an end-to-end fashion. To enhance the performance of the proposed neural network for removing impulsive noise with higher density, we generated a customized dataset which contains noisier images with higher frequencies. Moreover, to help the training procedure, we presented a multi-term loss function with a term imposing the sparsity of the impulsive noise. Finally, we proposed a post-processing stage to further improve the reconstruction quality of our method. Experimental results demonstrate that the proposed method with the customized dataset and multi-term loss function is successful in removing the impulsive noise from color images, and it is fast, especially when running on a GPU-equipped system. Furthermore, the proposed post-processing step can further improve the performance in about 20 milliseconds.

References

[1] Namyong Kim, Hyung-Gi Byun, Young-Hwan You, and Kihyeon Kwon, "Blind signal processing for impulsive noise channels," *Journal of Communications and Networks*, vol. 14, no. 1, pp. 27–33, 2012.

[2] Kyong Hwan Jin and Jong Chul Ye, "Sparse and low-rank decomposition of a hankel structured matrix for impulse noise removal," *IEEE Transactions on Image Processing*, vol. 27, no. 3, pp. 1448–1461, 2017.

[3] Konstantinos N Plataniotis and Anastasios N Venetsanopoulos, *Color image processing and applications*, Springer Science & Business Media, 2013.

[4] Rastislav Lukac, Bogdan Smolka, Karl Martin, Konstantinos N Plataniotis, and Anastasios N Venetsanopoulos, "Vector filtering for color imaging," *IEEE Signal Processing Magazine*, vol. 22, no. 1, pp. 74–86, 2005.

[5] Rastislav Lukac, "Adaptive vector median filtering," *Pattern Recognition Letters*, vol. 24, no. 12, pp. 1889–1899, 2003.

[6] Bogdan Smolka and Andrzej Chydzinski, "Fast detection and impulsive noise removal in color images," *Real-Time Imaging*, vol. 11, no. 5-6, pp. 389–402, 2005.

[7] Lukasz Malinski and Bogdan Smolka, "Fast averaging peer group filter for the impulsive noise removal in color images," *Journal of Real-Time Image Processing*, vol. 11, no. 3, pp. 427–444, 2016.

[8] Jiangtao Xu, Lei Wang, and Zaifeng Shi, "A switching weighted vector median filter based on edge detection," *Signal Processing*, vol. 98, pp. 359–369, 2014.

[9] Jian Wu and Chen Tang, "Random-valued impulse noise removal using fuzzy weighted non-local means," *Signal, Image and Video Processing*, vol. 8, no. 2, pp. 349–355, 2014.

[10] Amarjit Roy, Joyeeta Singha, Salam Shuleenda Devi, and Rabul Hussain Laskar, "Impulse noise removal using svm classification based fuzzy filter from gray scale images," *Signal Processing*, vol. 128, pp. 262–273, 2016.

[11] Hui Ying Khaw, Foo Chong Soon, Joon Huang Chuah, and Chee-Onn Chow, "High-density impulse noise detection and removal using deep convolutional neural network with particle swarm optimisation," *IET Image Processing*, vol. 13, no. 2, pp. 365–374, 2018.

[12] Wenhua Zhang, Lianghai Jin, Enmin Song, and Xiangyang Xu, "Removal of impulse noise in color images based on convolutional neural network," *Applied Soft Computing*, vol. 82, pp. 105558, 2019.

[13] Yiqiu Dong, Raymond H Chan, and Shufang Xu, "A detection statistic for random-valued impulse noise," *IEEE Transactions on Image Processing*, vol. 16, no. 4, pp. 1112–1120, 2007.

[14] Sahar Sadrizadeh and Farokh Marvasti, "Impulsive noise removal from gray-scale video sequences via adaptive thresholding," in *2020 10th International Symposium on Telecommunications (IST)*, 2020, pp. 142–145.

[15] Jielin Jiang, Lei Zhang, and Jian Yang, "Mixed noise removal by weighted encoding with sparse nonlocal regularization," *IEEE Transactions on Image Processing*, vol. 23, no. 6, pp. 2651–2662, 2014.

[16] Licheng Liu, Long Chen, CL Philip Chen, Yuan Yan Tang, et al., "Weighted joint sparse representation for removing mixed noise in image," *IEEE Transactions on Cybernetics*, vol. 47, no. 3, pp. 600–611, 2016.

[17] Sahar Sadrizadeh, Nematollah Zarmehi, Ehsan Asadi, Hamidreza Abin, and Farokh Marvasti, "A fast iterative method for removing impulsive noise from sparse signals," *IEEE Transactions on Circuits and Systems for Video Technology*, 2020.

[18] Ruixuang Wang, Markus Pakleppa, and Emanuele Trucco, "Low-rank prior in single patches for nonpointwise impulse noise removal," *IEEE Transactions on Image Processing*, vol. 24, no. 5, pp. 1485–1496, 2015.

[19] 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.

[20] Yang Chen, Yudong Zhang, Huazhong Shu, Jian Yang, Limin Luo, Jean-Louis Coatrieux, and Qianjin Feng, "Structure-adaptive fuzzy estimation for random-valued impulse noise suppression," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 2, pp. 414–427, 2016.

[21] Andrey Ignatov, Nikolay Kobyshev, Radu Timofte, Kenneth Vanhoey, and Luc Van Gool, "Dslr-quality photos on mobile devices with deep convolutional networks," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 3277–3285.

[22] Kai Zhang, Wangmeng Zuo, Shuhang Gu, and Lei Zhang, "Learning deep cnn denoiser prior for image restoration," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 3929–3938.

[23] Ryo Abiko and Masaaki Ikehara, "Blind denoising of mixed gaussian-impulse noise by single cnn," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 1717–1721.

[24] Bolei Zhou, Agata Lapedriza, Antonio Torralba, and Aude Oliva, "Places: An image database for deep scene understanding," *Journal of Vision*, vol. 17, no. 10, pp. 296–296, 2017.

[25] Sergey Ioffe and Christian Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2015, pp. 448–456.

[26] Farokh Marvasti, *Nonuniform sampling: theory and practice*, Springer Science & Business Media, 2012.

[27] Mitra Basu and Min Su, "Image smoothing with exponential functions," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 15, no. 04, pp. 735–752, 2001.

[28] Lianghai Jin, Zhiliang Zhu, Enmin Song, and Xiangyang Xu, "An effective vector filter for impulse noise reduction based on adaptive quaternion color distance mechanism," *Signal Processing*, vol. 155, pp. 334–345, 2019.