Received August 11, 2019, accepted August 20, 2019, date of publication September 2, 2019, date of current version September 13, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2938799

A Blind CNN Denoising Model for Random-Valued Impulse Noise

JUNQING CHEN, GUIZHEN ZHANG, SHAOPING XU, AND HAIWEN YU
School of Information Engineering, Nanchang University, Nanchang 330031, China

Corresponding author: Shaoping Xu (xushaoping@ncu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61662044, Grant 51765042, and Grant 61163023, in part by the Jiangxi Provincial Natural Science Foundation under Grant 20171BAB202017, and in part by the Jiangxi Provincial Graduate Innovation Special Fund under Grant YC2018-S066.

ABSTRACT Denoising convolutional neural networks (DnCNNs), initially developed for Gaussian noise removal, are powerful nonlinear mapping models in image processing. After changes in training data, they can be used for suppression of random-valued impulse noise (RVIN) with excellent results. To achieve favorable denoising performance, however, it is necessary to have an accurate perception of the noise ratio so that the most suitable DnCNN can be chosen for denoising. Thus, this model is severely limited in flexibility. To address this problem, we propose a blind CNN model for RVIN denoising with a flexible noise ratio predictor (NRP) as an indicator. Some patches are randomly selected from the RVIN-corrupted test image, and feature vectors that indicate whether the center pixel is contaminated or not are extracted by the predictor. These feature vectors are composed of multiple statistics, namely, the multiple rank-ordered absolute differences (ROADs), the clean pixel median deviation (CPMD), and the edge pixel difference (EPD). They are rapidly mapped to noise/clean (1 for noise, 0 for clean) labels by the pre-trained noise detector (the key component of our NRP). According to the ratio of the obtained noisy labels to the total number of selected patches, the predictor provides the noise ratio of the whole image. From the output of the NRP, i.e., the predicted noise ratio, the most appropriate DnCNN specifically trained for this noise ratio is exploited for denoising. Under the guidance of the NRP, the proposed method has the ability to handle unknown noise ratios. Simulation results indicate that our blind denoising CNN model achieves state-of-the-art performance in terms of both execution efficiency and restoration results.

INDEX TERMS Image denoising, random-valued impulse noise, noise ratio prediction, convolutional neural networks.

I. INTRODUCTION

Noisy camera sensors or imperfect transmission frequently lead to the introduction of impulse noise (IN) into digital images. Impulse noise can be divided into two categories: fixed-valued impulse noise (FVIN) and random-valued impulse noise (RVIN). FVIN takes the minimum (0 or black dots) or the maximum (255 or white dots) grayscale value with equal probabilities in the corrupted gray-level image. For an RVIN-contaminated image, noise pixels are randomly valued in the range [0–255]. Therefore, removal of RVIN is much more difficult than that of FVIN. In this paper, we focus mainly on RVIN denoising.

To suppress RVIN, a number of techniques have been proposed over the last few decades. These include nonlinear filters such as the median (MED) filter [1], the center-weighted median (CWM) filter [2], and the lower upper median (LUM) filter [3]. These methods are simple and of low computational cost, however, they process all pixels without identifying whether each pixel is corrupted or not, which leads to blurring of the image details and edges. To solve this problem, and thus avoid changing clean pixels, several filters that are integrated with noise detectors have been proposed [4]–[9], such as the adaptive switching median (ASWM) filter [4] and the ROR nonlocal mean (ROR-NLM) [8]. Garnett *et al.* [10] found that the information pixels have a strong relationship with their surrounding pixels in natural images. The rank-ordered absolute difference (ROAD) was then proposed to describe noise pixels. Also based on this strategy, the rank-ordered

The associate editor coordinating the review of this article and approving it for publication was Alexandros Iosifidis.

logarithmic difference (ROLD) [11] can amplify the difference between clean and noise pixels by using a logarithmic function. As a noise-detecting statistic, ROLD is always combined with edge-preserving regularization (EPR) [12] for RVIN removal. Executing iteratively, ROLD-EPR performs well both in noise detection and image restoration, but at the expense of a long execution time.

In the last few years, neural networks and fuzzy systems, with their powerful data processing capabilities, have gradually been introduced into image processing. In [13], an impulse noise filter using fuzzy logic to process the RVIN-contaminated images was presented, which outperforms conventional methods in its powerful noise suppression ability as well as in its ability to preserve image details. In [14], a neural detector trained by an artificial neural network was proposed for RVIN detection. This method has two different input training parameters: a pixel-wise median of the absolute deviations from the median (MAD) [15] and the ROAD statistics [10]. However, as a component of improved adaptive impulsive noise suppression (IAINS), the detector is only suitable for low-level noise detection. To solve this problem, an improved neural detector was introduced in [16]. Taking ROAD and ROLD statistics as training inputs, the noise detector proposed in [16] can detect noise at higher ratios, but it cannot distinguish between edge pixels well, and its detection efficiency is not good enough. Based on similar standard median filters, an extension was introduced in [17]. By utilizing a modified hybrid median algorithm, the improved filter replaces the current pixel in the filtering process. Although it has better performance than standard median filters, it is only effective at low noise ratios. In [18], a multilayer neural-network-based noise detector with five training parameters was proposed for RVIN suppression.

In contrast to the traditional iterative-based methods, these training-based methods possess high execution efficiency, since they avoid multiple iterative computations. However, they still suffer from several drawbacks. For instance, for complex images, even if the model is well trained, the detection accuracy will suffer somewhat of a loss. In other words, these noise detectors are not completely free from false detections and missed detections. False detections lead to blurring of the image, while missed detections lead to residual noise pixels, both of which will degrade image quality. Furthermore, filtering by traditional RVIN denoisers is affected by imperfect detection results and the processing only of noise-detected pixels, leading to ineffective restoration.

In recent years, convolutional neural networks (CNNs) have attracted much attention owing to their powerful processing ability. They show much potential for use in image feature extraction, which can benefit from their deep architecture. The development of CNNs, in particular with regard to improvements in their denoising performance, has greatly benefited from the emergence of new techniques such as the rectified linear unit (ReLU) [19], residual learning [20]–[22], batch normalization [23], parameter initialization [24], tradeoff between depth and width [25], and gradient-based optimization algorithms [26]–[28]. In addition, the advent of powerful graphics processing units (GPUs) has greatly reduced the execution time of CNNs. Simonyan *et al.* [29] proposed an ImageNet classification based on a CNN with very deep architecture. Zhang *et al.* [30] introduced a CNN denoiser with seven dilated convolution layers [31], which gives a good tradeoff between denoising ability and execution efficiency. On the basis of residual learning [20], a residual network can easily learn an accurate residual mapping for a few stacked layers. Extremely deep CNNs can be easily trained, and favorable accuracy can been achieved for image denoising when the image likelihood is known. As long as a DnCNN model is well trained, it can achieve state-of-the-art performance for noise ratios that are close to its training noise ratio; however, this is also a limitation. In [32], a deep CNN for natural image denoising was proposed, which gives a favorable restoration result in terms of both quantitative measures and visual effect. In contrast to most of the traditional denoising methods that remove noise pixel-by-pixel, which leads to inefficient restoration, the CNN-based technique completes denoising rapidly and effectively owing to its powerful denoising ability, which benefits from the deep architecture. In [33], a discriminative DnCNN model for Gaussian denoising was proposed, which has outstanding performance in terms of both noise suppression and computational efficiency. Considering the favorable performance of DnCNN models, we can use this technique in RVIN denoising. However, this model is severely limited in flexibility, i.e., to produce a satisfactory performance, a DnCNN model needs to know the severity of corruption of the noisy image. For RVIN denoising, the noise ratio is a good metric to represent this severity.

To overcome this drawback, in this paper, we propose an adaptive blind CNN-based model for RVIN denoising with a noise ratio predictor that can measure the severity of corruption (i.e., the noise ratio) of the image rapidly and efficiently. Under the guidance of the noise ratio predictor (NRP), the most suitable DnCNN model is exploited for image restoration. By randomly selecting some patches from the test image and extracting noise feature vectors for the corresponding center pixels with these patches, our noise detector (the key component of our NRP) is able to identify whether the center pixel is noise or not, based on the knowledge acquired in the training phase. To strengthen the detection accuracy, our proposed neural noise detector adopts a feature vector consisting of multiple robust statistics, the multiple rank-ordered absolute differences (ROADs), the clean pixel median deviation (CPMD), which describes how likely the investigated pixel is to be a noise pixel, and the edge pixel difference (EPD), which can distinguish the edge pixels and the noise pixels. After calculating the ratio of the identified noise pixels to the total number of selected patches, the predictor rapidly gives the predictive noise ratio of the test image, and the most appropriate DnCNN, which is trained by provided training image patches obtained from

RVIN-contaminated images, is then exploited for denoising. Compared with traditional switching denoising algorithms, the proposed model avoids image degradation resulting from the dependence on imperfect detection, and restores images by utilizing prior knowledge obtained from the training phase. The use of an NRP allows the proposed model to overcome the limitations on flexibility from which traditional CNN denoising methods suffer.

The remainder of this paper is organized as follows: related work is described in Section II, the proposed method is introduced in Section III, simulation results are compared in Section IV, and conclusions are presented in Section V.

II. RELATED WORK

A. REVIEW OF RVIN DENOISING ALGORITHMS

To address RVIN denoising problems, many techniques have been proposed over the last few decades. PSMF [34], ROLD-EPR [11], ASWM [4], and ROR-NLM [8] are commonly used switching-based algorithms, whose denoising process relies heavily on the detection results. To achieve higher detection accuracy and better denoising performance, these algorithms are implemented in an iterative manner, leading to long execution times. The training-based denoising algorithms, such as ANN [16], possess high execution efficiency and favorable performance. However, their detection accuracy is still unsatisfactory, which may lead to image degradation in the subsequent denoising process. Recently, RVIN denoising algorithms based on matrix completion have been proposed [35]–[39]. Ongie *et al.* [35] proposed an image inpainting method called the generic iterative reweighed annihilation filter (GIRAF) algorithm, which works in the original unlifted domain by exploiting the convolutional structure of the lifted matrix. However, GIRAF removes only the detected noise pixel-by-pixel, which means that the quality of restored image is still limited by the detection accuracy. Jin *et al.* [39] noted that an image corrupted by RVIN has intact pixels and proposed to use low-rank Hankel structured matrix to model the underlying image. Although GIRAF [35] and ALOHA [39] can achieve promising denoising performance, they require long execution times.

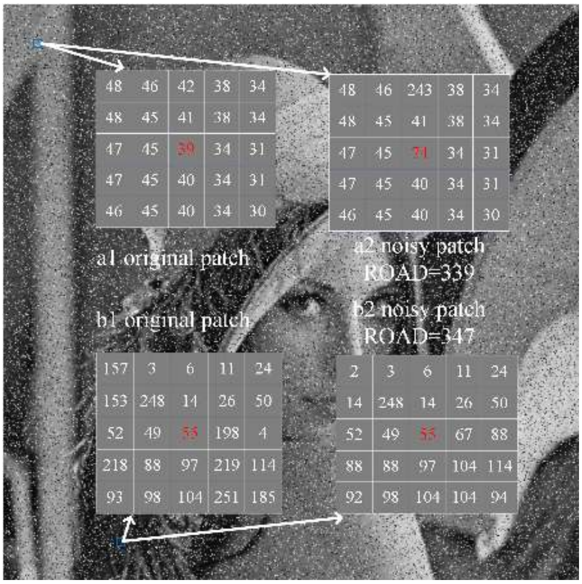
B. REVIEW OF ROAD

The ROAD statistic proposed in [10] is effective for detecting pixels corrupted with RVIN. It can be briefly summarized as follows. Let $x_{i,j}$ be the intensity value of the pixel located at (i, j); then a $(2N + 1) \times (2N + 1)$ window containing the set of coordinates centered at (i, j) can be defined as

$$\Omega_{i,j} = \{x_{i+s,j+t}\} \tag{1}$$

where $-N \leq s, t \leq N$. The absolute difference of each pixel in this window is calculated as

$$d_{i,j}(s, t) = |x_{i+s,j+t} - x_{i,j}|. \tag{2}$$



48	46	42	38	34
48	45	41	38	34
47	45	39	34	31
47	45	40	34	31
46	45	40	34	30

a1 original patch

48	46	243	38	34
48	45	41	38	34
47	45	71	34	31
47	45	40	34	31
46	45	40	34	30

a2 noisy patch
ROAD=339

b1 original patch

157	3	6	11	24
153	248	14	26	50
52	49	53	198	4
218	88	97	219	114
93	98	104	251	185

b2 noisy patch
ROAD=347

2	3	6	11	24
14	248	14	26	50
52	49	55	67	88
88	88	97	104	114
92	98	104	104	94

FIGURE 1. Comparison of ROAD statistics of two different center pixels.

Then, all the $d_{i,j}$ values are sorted in ascending order, and the nth smallest $d_{i,j}$ are selected to calculate ROAD as

$$\mathrm{ROAD}_n(x_{i,j}) = \sum_{k=1}^{n} r_k(x_{i,j}) \tag{3}$$

where $r_k(x_{i,j})$ is the kth smallest $d_{i,j}$.

C. WEAKNESS OF ROAD

According to [10], larger ROAD values indicate noise pixels, while smaller values indicate clean pixels. The ROAD strategy identifies noise pixels by only comparing the ROAD value of the investigated center pixel with a predefined threshold, which has limited detection capability. In Fig. 1, two 5×5 patches are extracted from the *Lena* image corrupted by 20% RVIN. Patches a2 and b2 are extracted from a smooth area and a detailed region, respectively. As shown in Fig. 1, the ROAD statistics of patches a2 and b2 are 339 and 347, respectively, which are very close. Hence, if we use only one threshold, these two center pixels will both be identified as noise pixels and consequently will be changed in the subsequent filtering procedure. However, the center pixels of patches a2 and b2 are actually noise and clean pixels, respectively. This reveals the weak detection capability of the conventional ROAD strategy. As shown in Table 1, the difference between patches a2 and b2, ranging from ROAD_1 to ROAD_n, is obvious. As the center pixel of patch a2 is corrupted by RVIN, its intensity is quite different from those of its surrounding pixels. Thus, the ROAD_n statistics of patch a2 increase rapidly when n is small, while this would not be the case for clean one. Hence, we adopt multiple different ROAD_n statistics to better identify noise and clean pixels. Furthermore, to enhance the description capability of our NRP, we also adopt another two features (namely, CPMD and EPD), which will be described in detail in Section III.

TABLE 1. Comparison of ROAD$_n$ values of the center pixels in Fig. 1.

Image patch	ROAD$_n$											
	$n=1$	$n=2$	$n=3$	$n=4$	$n=5$	$n=6$	$n=7$	$n=8$	$n=9$	$n=10$	$n=11$	$n=12$
a2	26	52	79	106	134	162	191	220	249	278	308	339
b2	3	9	20	44	70	103	136	173	214	255	298	347

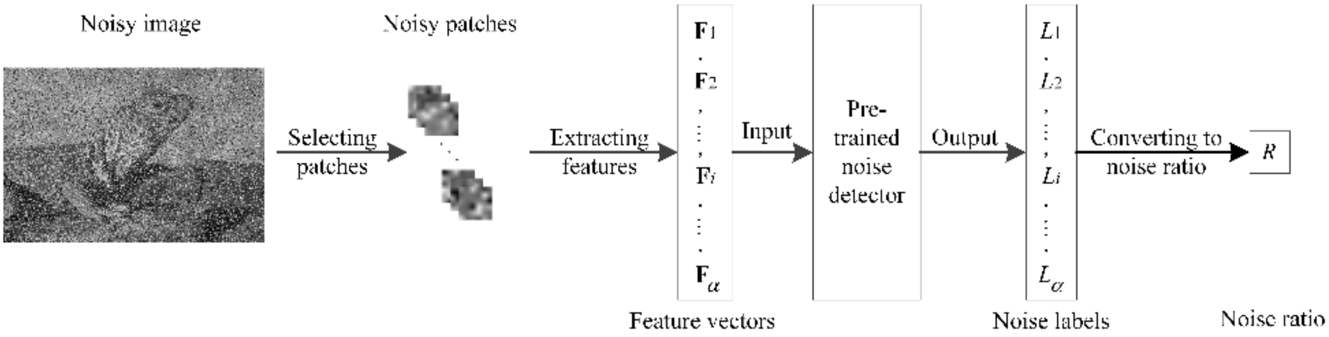


FIGURE 2. The overview of the proposed noise ratio predictor.

III. PROPOSED METHOD

To handle denoising problems with unknown noise ratio, a good blind denoising algorithm is expected to have the following desirable properties: (1) it should be able to handle unknown noise ratios in a flexible manner; (2) it should possess state-of-the-art execution efficiency and denoising performance. To achieve this goal, we propose an NRP and combine it with CNN models for RVIN denoising. To improve the performance of the training-based RVIN predictor in terms of both execution efficiency and detection accuracy, we train our noise detector (the key component of our NRP) with multiple statistics, including ROADs, CPMD, and EPD, which are able to provide good descriptions of the characteristics of noise pixels and edge pixels. For RVIN denoising, we adopt a CNN model, since it can achieve fast and effective denoising performance, benefiting from its deep architecture. Our blind denoising model provides fast yet accurate prediction at a wide range of noise ratios, independent of any parameters (without using predefined thresholds) and, based on the knowledge acquired during the training phase, it then adaptively adopts the most appropriate CNN model for denoising.

A. NOISE RATIO PREDICTOR

Fig. 2 shows the framework of the proposed NRP. A test noisy image was first decomposed into overlapping patches by raster scanning in a pixel-by-pixel sliding window manner. Then, some patches were randomly selected for extracting feature vectors that were fed forward to the pre-trained noise detector. Based on a large number of training samples, the pre-trained noise detector quickly assigned a noise/clean label to the center pixel of each selected patch. Finally, the noise ratio of the identified noise pixels to the total number of selected patches was calculated as the predictive noise ratio for the test noisy image.

For noise detection, the input feature vector $\mathbf{F}_i$ of the pre-trained noise detector is composed of multiple ROAD statistics, the CPMD statistic, and the EPD statistic. The ROAD statistic has already been reviewed in Section II, and we will describe the other two features in the following two subsections.

1) CPMD

Although the ROAD statistic can efficiently identify noise pixels, it does not on its own provide good enough performance. The detection performance of a noisy window is always far worse than that of a clean window. Therefore, a CPMD strategy based on the ROAD statistic is adopted to boost the noise detection capability. This strategy can be described as follows:

1) Calculate the ROAD statistics of the sliding window $\Omega_{i,j}$ using (3), then sort ROAD$_n(x_{i,j})$ in ascending order and denote this by

$$R(x_{i,j}) = \left\{r_1, r_2, \ldots, r_k, \ldots, r_{(2N+1)^2}\right\} \tag{4}$$

where r_k is the ascending sorted value of $r_k(x_{i,j})$.

2) Calculate the absolute difference between the two adjacent r_k for all the elements in $R(x_{i,j})$, defined as

$$U_{i,j}=\{u_k|u_k=|r_{k+1}-r_k|, k=1,2,\ldots,(2N+1)^2-1\}. \tag{5}$$

3) Find a u_k in $U_{i,j}$, let $u_g \leq T$ for $g < k$, and $u_k > T$, where T is the average of the set U. Then we get the $k+1$ minimum ROAD statistics, denoted by

$$\tilde{R}(x_{i,j}) = \{r_1, r_2, \ldots, r_{k+1}\}. \tag{6}$$

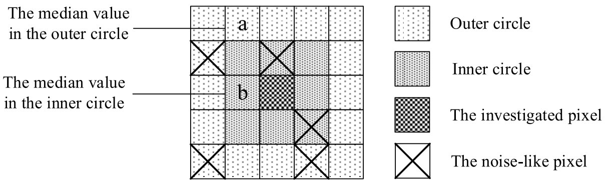

The median value in the outer circle
The median value in the inner circle
a
b
Outer circle
Inner circle
The investigated pixel
The noise-like pixel

FIGURE 3. Example of a CPMD calculation.

4) Those pixels with ROAD statistics that correspond to $\tilde{R}(x_{i,j})$ are picked as clean-like pixels, defined as

$$\mathrm{CP}_{i,j} = \{\tilde{x}_k | R(\tilde{x}_k) = r_k, r_k \in \tilde{R}(x_{i,j})\}. \tag{7}$$

5) The median of $\mathrm{CP}_{i,j}$ of the inner or outer circle is computed by

$$\mathrm{CPM}_{i,j} = \mathrm{median}\{\mathrm{CP}_{i,j}\}. \tag{8}$$

Here, taking a 5 × 5 window as an example, the inner (outer) median is calculated in Fig. 3.

6) Calculate the deviation between $x_{i,j}$ and its corresponding inner and outer medians, and select the smaller one as the CPMD:

$$\mathrm{CPMD}_{i,j} = \min\{|\mathrm{CPM}_{i,j}^{\mathrm{inner}} - x_{i,j}|, |\mathrm{CPM}_{i,j}^{\mathrm{outer}} - x_{i,j}|\} \tag{9}$$

where $\mathrm{CPM}_{i,j}^{\mathrm{inner}}$ and $\mathrm{CPM}_{i,j}^{\mathrm{outer}}$ represent the corresponding inner and outer medians, respectively.

This feature reflects the difference between the investigated pixel and "clean" pixels; in other words, if $x_{i,j}$ is a noise pixel, its corresponding CPMD should be large.

2) EPD

Edge detection has always been a tricky problem in noise detection. Some information pixels in the edges can easily be falsely identified as noise pixels. Therefore, we propose a third feature, namely, the edge pixel difference (EPD), to better distinguish between noise pixels and edge pixels. If the current pixel is on the edge, its derivative in the edge direction should be small, i.e., the intensity values between edge pixels should be very close, whereas this is not the case for noise pixels. The concrete steps in determining the EPD are as follows:

1) Calculate the following datasets for four directions in the sliding window $\Omega_{i,j}$, i.e., horizontal, vertical, left diagonal, and right diagonal, respectively:

$$e_{i,j}^p = \{d_n^p | d_n^p = |x_{h,l}^p - x_{i,j}|\} \tag{10}$$

where p represents the different directions, and $x_{h,l}$ represents the edge pixels on the four directions passing through the center pixel, i.e., $x_{h,l} = x_{i,j-N+n}$ for horizontal, $x_{h,l} = x_{i-N+n,j}$ for vertical, $x_{h,l} = x_{i-N+n,j-N+n}$ for left diagonal, and $x_{h,l} = x_{i-N+n,j+N-n}$ for right diagonal, where $0 \leq n \leq 2N, n \neq N$.

2) A logarithmic function is employed to amplify the difference between pixels while keeping the original small difference from growing too much. It is defined as

$$\tilde{e}_{i,j}^p = \left\{\tilde{d}_n^p \middle| \tilde{d}_n^p = 1 + \frac{\max\{\log_a(d_n^p), -b\}}{b}\right\} \tag{11}$$

where a and b are suggested to take the values 2 and 5, respectively, after trials.

3) To further distinguish noise and other pixels, each set is summed, and the minimum is selected and defined as the EPD:

$$\mathrm{EPD}_{i,j} = \min\left\{D_n^p \middle| D_n^p = \sum_{n=1}^{2N} \tilde{d}_n^p\right\}. \tag{12}$$

After trials, we suggest that our noise detector should be trained with the following feature vector:

$$\mathbf{F}_i = \{\mathrm{ROAD}_1, \ldots, \mathrm{ROAD}_n, \mathrm{CPMD}, \mathrm{EPD}\} \tag{13}$$

where $n = \frac{(2N+1)^2-1}{2}$ and ROAD, CPMD, and EPD are calculated using (3), (9), and (12), respectively. It is worth noting that if the center pixel of a patch is noisy, then all the features to which it corresponds will be large. For a clean center pixel, the ROAD_n ($n = 1, \ldots, \frac{(2N+1)^2-1}{2}$) values of this pixel are not all large, whereas this is not the case for noise pixels, which indicates that our multiple ROAD statistics scheme possesses a stronger description capability than the single ROAD scheme. Moreover, this ROAD_n ($n = 1, \ldots, \frac{(2N+1)^2-1}{2}$) scheme takes full advantage of the cumulative summation procedure, which does not require any extra execution time. In this paper, we have adopted a 5 × 5 sliding window, i.e., $N = 2$, ROAD_n ($n = 1, \ldots, 12$).

B. NOISE DETECTOR TRAINING

To improve our noise detector in terms of both flexibility and accuracy, we corrupted 200 images in the Berkeley segmentation dataset (BSD300) [40] from 10% to 70% RVIN in increments of 2%. Then, we randomly selected 50 patches for each image at each noise ratio, from which we extracted the proposed feature vector, i.e., (13), and the corresponding noise label indicating whether the center pixel is noise or not. About 300 000 training data pairs were eventually obtained and used for training.

To achieve fast yet accurate noise detection, we adopt a back-propagation (BP) neural network to train our detector, owing to its fast learning ability. To establish the most appropriate network architecture, we vary the number of neurons from three to twelve, and the number of hidden layers from one to four, with the results shown in Figs. 4 and 5. Fig. 4 reveals that the accuracy rate of the noise detectors with five neurons is the best; however, from Fig.5, the accuracy rates of the noise detectors with two, three, and four hidden layers are very close. To make a tradeoff between detection accuracy and execution efficiency, we eventually modeled a noise detector on the basis of a BP neural network with

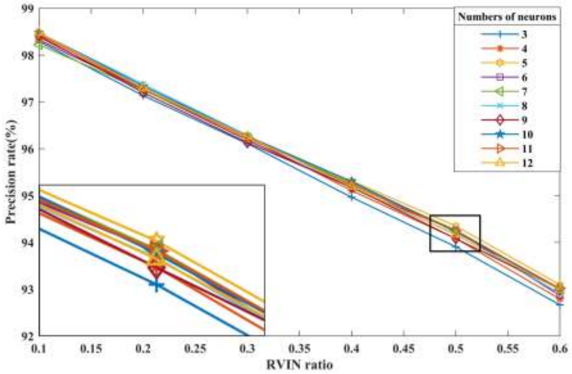


FIGURE 4. Average precision rate of the proposed noise detector with different numbers of neurons.

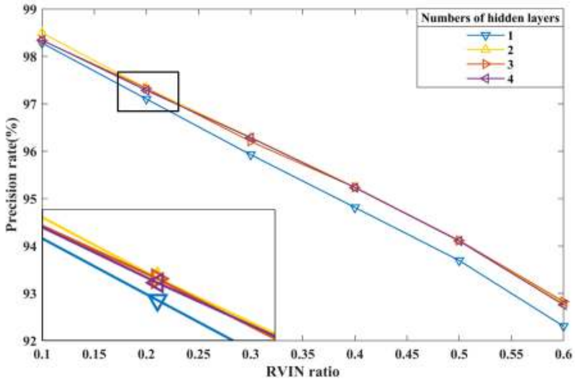


FIGURE 5. Average precision rate of the proposed noise detector with different numbers of hidden layers (five neurons each).

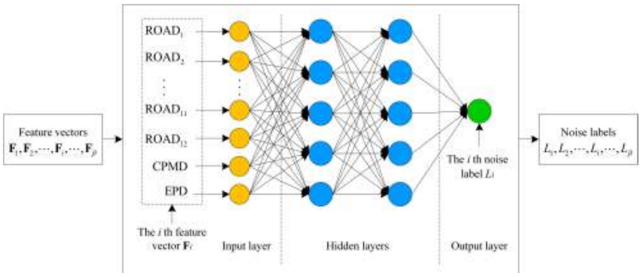

Feature vectors
$\mathbf{F}_1, \mathbf{F}_2, \cdots, \mathbf{F}_i, \cdots, \mathbf{F}_\beta$
ROAD$_1$
ROAD$_2$
ROAD$_{11}$
ROAD$_{12}$
CPMD
EPD
The i th feature vector $\mathbf{F}_i$
Input layer
Hidden layers
Output layer
The i th noise label L_i
Noise labels
$L_1, L_2, \cdots, L_i, \cdots, L_\beta$

FIGURE 6. Architecture of the proposed noise detector.

two hidden layers, each with five neurons. The architecture adopted for the proposed noise detector is shown in Fig. 6.

C. DnCNN Model

After completing the training of the noise detector, we next establish the DnCNN model. From a residual learning strategy, deep CNN models can be easily implemented and can achieve favorable denoising performance. We adopt the same network architecture as proposed by Zhang *et al.* [33], with "Conv + ReLU" being adopted for the first layer, "Conv + BN + ReLU" for the middle layers, and "Conv + ReLU" for the last layer. At the same time, we extend the DnCNN model by providing an RVIN-corrupted image patch pair to enable it to handle RVIN denoising. The noisy observation $\mathbf{y} = \mathbf{x} + \mathbf{v}$ is employed as the input of DnCNN, where $\mathbf{x}$ and $\mathbf{v}$ denote the clean image and random-valued impulse noise, respectively. Simultaneously, a residual learning formulation is used to obtain a residual mapping $\mathcal{R}(\mathbf{y}) \approx \mathbf{v}$. During the training process, the desired parameters Θ are learned in the deep architecture of the CNN and represent the averaged mean square error between the residual images and the investigated ones from the noisy input, which can be expressed as follows:

$$\Gamma(\Theta) = \left\{ \frac{1}{2M} \sum_{m=1}^{M} \|\mathcal{R}(\mathbf{y}_m; \mathbf{\Theta}) - (\mathbf{y}_m - \mathbf{x}_m)\|_F^2 \right\} \quad (14)$$

where $\mathbf{y}_m$ and $\mathbf{x}_m$ denote the mth corresponding RVIN-corrupted and clean image training patch pair, respectively.

Zhang *et al.* [33] noted that the CNN models are capable of handling RVIN denoising after changing the training pairs. They adopted a residual learning formulation [20] and batch normalization [23] in their CNN models, since these are mutually beneficial, and, even in a deep network structure, they can still provide fast convergence and good restoration results. Hence, we adopt these two techniques in our DnCNN model to speed up the training phase as well as to boost the denoising performance. In this paper, we adopt the same parameter settings as in [33], which are as follows: a receptive field of 40 × 40 with a corresponding depth of 17, a batch size of 128, a weight decay of 0.0005, and a momentum of 0.9. We find that our DnCNN models can achieve state-of-the-art RVIN denoising performance with these parameter settings.

To train a CNN model for RVIN denoising, 256 000 patches were extracted from the Berkeley segmentation dataset (BSD) [40]. We trained six CNN denoising models, setting RVIN ratios at 10%, 20%, 30%, 40%, 50%, and 60%, respectively, and each DnCNN model was effective over a small range of corresponding noise ratios.

IV. RESULTS AND DISCUSSION

To illustrate the excellent noise suppression capability of the proposed method, we compared it with some state-of-the-art methods, such as PSMF [34], ROLD-EPR [11], ASWM [4], ROR-NLM [8], ANN [16], GIRAF [35], and ALOHA [39]. For RVIN ratios from 10% to 60% with interval 10%, we trained six specific CNN denoising models, whose parameter settings are the same as in [33]. As shown in Fig. 7, some commonly used gray-level test images, such as *Lena*, *Boat*, *House*, *Bridge*, and *Man*, were contaminated with RVIN of various noise ratios for the experiments. We used the root mean square error (RMSE) to evaluate the performance of our predictor. For the restoration results, the peak signal-to-noise ratio (PSNR), structural similarity (SSIM) [41], and feature similarity (FSIM) [42] were employed for image quality assessment. All the experiments were carried out in a MATLAB (R2018a) environment on a PC with Intel Core i7-6700K CPU at 4.00 GHz, with 16 GB of RAM and an Nvidia Quadro M4000 GPU.

A. EDGE DETECTION PERFORMANCE

To verify that the EPD strategy can identify edge pixels in a noisy image, the detection results with and without EPD

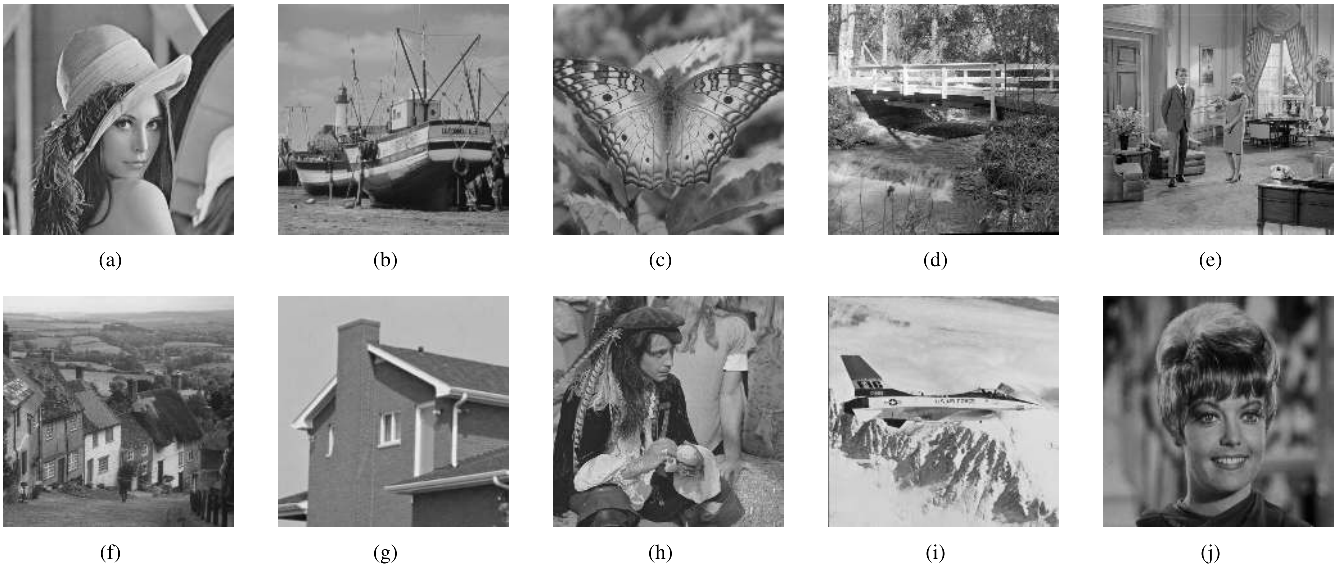


FIGURE 7. Examples of test images. (a) *Lena*. (b) *Boat*. (c) *Butterfly*. (d) *Bridge*. (e) *Couple*. (f) *Goldhill*. (g) *House*. (h) *Man*. (i) *Plane*. (j) *Zelda*.

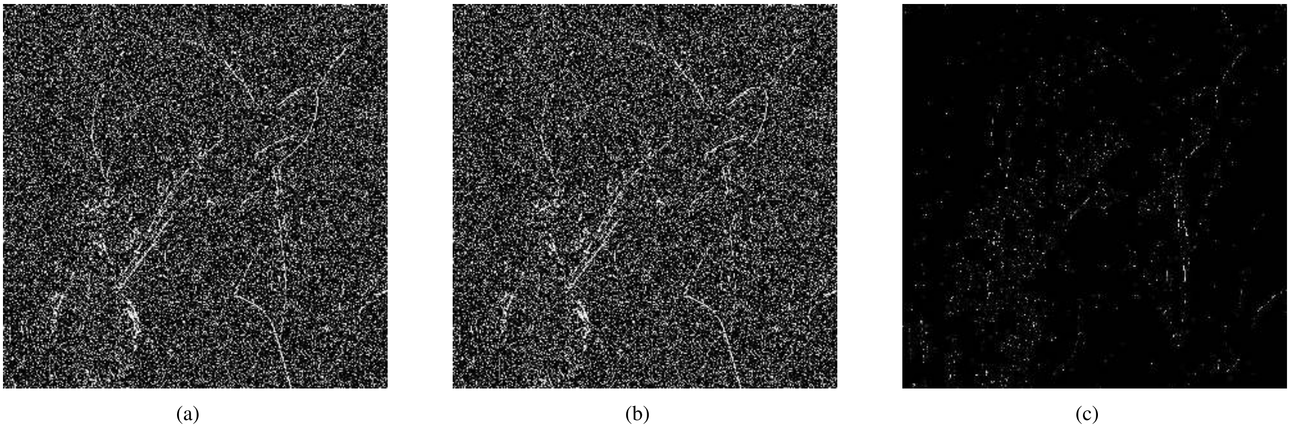


FIGURE 8. Comparison of detection results with and without EPD strategy for the *Lena* image corrupted by 20% noise ratio. (a) Without EPD. (b) With EPD. (c) Difference.

strategy based on a 20% RVIN-corrupted *Lena* image are compared in Fig. 8. Figs. 8(a), 8(b), and 8(c) show the detection result without EPD strategy, the result with EPD strategy, and the difference between these two results, respectively. In Figs. 8(a) and 8(b), the black dots represent pixels identified as noise, and the white dots represent clean pixels. In Fig. 8(c), the white dots represent the edge pixels detected by the EPD strategy. Clearly, Fig. 8(c) contains most of the edge pixels and a small number of noise pixels, which indicates that our method can detect edge pixels accurately.

B. NOISE PREDICTION PERFORMANCE OF DIFFERENT FEATURE COMBINATIONS

In this subsection, to prove the superiority of the proposed feature combination, we compare the noise detection performances within a 5 × 5 window of four different feature combinations, which we name “ROAD”, “Proposed-0”, “Proposed-1”, and “Proposed-2” (Table 2). For ease of comparison, the top two values are shown in bold. From Table 2, we can easily see that the detection precisions produced by the single ROAD statistic degrade severely as the the noise ratio increases. The results produced by Proposed-0 (i.e., the ROAD_n $(n = 1, \ldots, 12)$ statistics) were significantly improved compared with those from single ROAD, especially at high noise ratios. Moreover, the Proposed-1 results had higher precision than those of Proposed-0, which reveals that the noise detection performance is indeed improved by combining the CPMD statistic with the ROAD_n $(n = 1, \ldots, 12)$ statistics. Since the EPD statistic has been proved to be efficient in identifying edge pixels, it is natural to expect that after integrating the EPD statistic into Proposed-1, Proposed-2 (i.e., the proposed method) can achieve better noise detection performance than the the other feature combinations (Table 2).

TABLE 2. Comparison of noise detection precisions of different feature combinations.

Noise ratio	Method	*Butterfly*	*Goldhill*	*Lena*	*Plane*	*Boat*
10%	ROAD	**95.85%**	96.08%	96.57%	95.01%	95.69%
	Proposed-0	95.39%	96.06%	96.55%	**95.04%**	95.62%
	Proposed-1	95.52%	**96.10%**	**96.62%**	95.02%	**95.79%**
	Proposed-2	**95.90%**	**96.61%**	**96.95%**	**95.44%**	**96.10%**
20%	ROAD	93.73%	94.38%	94.50%	93.16%	93.66%
	Proposed-0	93.95%	94.51%	95.34%	94.04%	94.10%
	Proposed-1	**94.10%**	**94.60%**	**95.42%**	**94.15%**	**94.21%**
	Proposed-2	**94.36%**	**94.87%**	**95.65%**	**94.40%**	**94.68%**
30%	ROAD	91.66%	92.29%	92.53%	91.34%	91.69%
	Proposed-0	92.69%	92.87%	94.11%	93.21%	92.94%
	Proposed-1	**92.73%**	**93.02%**	**94.24%**	**93.29%**	**93.11%**
	Proposed-2	**92.84%**	**93.19%**	**94.28%**	**93.36%**	**93.37%**
40%	ROAD	89.57%	90.13%	90.48%	89.51%	89.60%
	Proposed-0	91.23%	91.37%	92.96%	92.35%	91.92%
	Proposed-1	**91.43%**	**91.45%**	**93.00%**	**92.44%**	**92.01%**
	Proposed-2	**91.32%**	**91.51%**	**92.97%**	**92.37%**	**92.07%**
50%	ROAD	87.54%	88.11%	88.57%	87.77%	87.64%
	Proposed-0	89.82%	89.68%	91.57%	91.30%	90.70%
	Proposed-1	**89.88%**	**89.81%**	**91.73%**	**91.47%**	**90.75%**
	Proposed-2	**89.88%**	**89.83%**	**91.61%**	**91.44%**	**90.73%**
60%	ROAD	85.42%	85.55%	86.50%	85.94%	85.52%
	Proposed-0	88.26%	87.98%	90.16%	90.16%	89.34%
	Proposed-1	**88.29%**	**88.02%**	**90.36%**	**90.33%**	**89.44%**
	Proposed-2	**88.37%**	**88.10%**	**90.18%**	**90.43%**	**89.47%**

"ROAD" represents the detection results produced by the single ROAD statistic, "Proposed-0" those produced by our method using just the ROAD$_n$ $(n = 1, \ldots, 12)$ statistics, "Proposed-1" those produced by the ROAD$_n$ $(n = 1, \ldots, 12)$ + CPMD statistics, and "Proposed-2" those produced by the ROAD$_n$ $(n = 1, \ldots, 12)$ + CPMD + EPD statistics (i.e., the proposed method).

C. COMPARISON OF NOISE PREDICTION

In the elimination of RVIN, accurate noise prediction is of great importance for the subsequent denoising procedure. To achieve superior restoration results, our proposed predictor provides a noise ratio prediction for the test image, and, guided by the noise ratio, a specifically trained DnCNN model is employed for denoising. In this subsection, our predictor is evaluated by comparison with some well-known noise detection methods, including PSMF [34], ROLD-EPR [11], ASWM [4], ROR-NLM [8], and ANN [16]. Considering that all these methods were originally developed for noise pixel detection, we convert their detection results into noise ratios for a fair comparison. Since GIRAF [35] and ALOHA [39] remove RVIN directly without an explicit noise detection process, we did not conduct noise detection experiments with these two methods. The performances of the above mentioned methods are assessed by the absolute deviation of prediction (ADP):

$$\mathrm{ADP}_i = |y_i - z_i| \tag{15}$$

where y_i and z_i are the predicted and true noise ratios, respectively.

Table 3 lists the absolute deviation of prediction results for the corrupted images *Lena*, *Plane*, and *Couple*. A good noise ratio predictor should be able to give a prediction that is close to the true noise ratio, i.e., smaller absolute deviation represents better performance. As shown in Table 3, although the deviations obtained by our method are slightly greater than the best ones in a few cases, in general our predictor comprehensively outperforms the other methods. Furthermore, the predictions obtained by our method deviate only slightly from the true values, which ensures that we can match the most appropriate DnCNN model for image restoration. Conventional noise detection methods, such as ROLD-EPR [11], always employ a single feature to identify noise pixels, which limits their ability to describe the characteristics of noise pixels. These methods suffer from three drawbacks: (1) blurring of edges; (2) fragile noise detection performance (especially at high noise ratios); (3) low execution efficiency. It is evident from Table 3 that the absolute

TABLE 3. Comparison of absolute deviation of predictions of different methods.

Method	PSMF	ROLD-EPR	ASWM	ROR-NLM	ANN	Proposed
Lena						
10%	2.72%	1.72%	1.72%	0.79%	2.05%	**0.71%**
20%	5.73%	2.74%	1.13%	0.83%	3.72%	**0.16%**
30%	8.82%	3.23%	**0.39%**	1.95%	4.07%	0.90%
40%	12.15%	4.40%	**0.73%**	3.81%	3.53%	1.56%
50%	15.85%	5.22%	2.35%	5.66%	4.44%	**2.27%**
60%	19.70%	5.05%	3.63%	7.80%	4.45%	**2.87%**
Plane						
10%	2.37%	2.08%	2.11%	2.08%	**1.87%**	2.86%
20%	5.46%	3.39%	1.69%	**0.77%**	2.52%	1.61%
30%	8.55%	4.77%	0.71%	0.68%	3.31%	**0.60%**
40%	11.90%	5.69%	0.46%	2.28%	1.85%	**0.22%**
50%	15.17%	5.53%	1.76%	3.92%	3.31%	**1.12%**
60%	18.57%	5.79%	3.94%	4.63%	3.51%	**2.11%**
Couple						
10%	**0.08%**	2.45%	6.03%	4.85%	2.41%	2.85%
20%	3.03%	3.98%	4.76%	2.76%	3.17%	**1.87%**
30%	6.18%	4.94%	3.68%	0.55%	4.25%	**0.51%**
40%	9.98%	5.84%	2.01%	1.70%	1.06%	**0.52%**
50%	14.19%	5.92%	**0.24%**	4.35%	3.02%	1.41%
60%	18.15%	6.96%	2.12%	6.40%	2.16%	**1.76%**

FIGURE 9. Examples of images for comparison of noise ratio prediction.

deviation of the prediction results produced by ROLD-EPR increases rapidly as the noise ratio increases, which means that ROLD-EPR cannot provide satisfactory prediction at high noise ratios. Our NRP outperforms the training-based method, i.e., ANN [14], for the majority of noise ratios. The reason for this is that our NRP possesses more robust and reliable features, the ROAD$_n$ $(n = 1, \ldots, 12)$ and CPMD statistics achieve favorable detection performance, and the EPD statistic prevents degradation of the edges.

To verify the general applicability of the proposed predictor, we took 50 images for testing and calculated the RMSE value for each noise ratio. The RMSE can be expressed as follows:

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{i=1}^{K}(\mathrm{ADP}_i)^2}{K}} \tag{16}$$

where ADP$_i$ is calculated using (15) and K is the total number of test images. All the test images were taken from the Berkeley segmentation dataset (BSD300) [40] and then corrupted with noise ratios at 10%, 20%, 30%, 40%, 50%, and 60%, respectively. Fig. 9 shows some examples of the selected images. From the results shown in Table 4, it can be seen

TABLE 4. Comparison of RMSE between predicted values and true values of different methods.

Method	PSMF	ROLD-EPR	ASWM	ROR-NLM	ANN	Proposed
10%	**0.0231**	0.0299	0.0626	0.0695	0.0438	0.0411
20%	0.0372	0.0457	0.0505	0.0505	0.0656	**0.0278**
30%	0.0676	0.0571	0.0389	0.0338	0.0897	**0.0158**
40%	0.1004	0.0663	0.0236	0.0284	0.1093	**0.0080**
50%	0.1373	0.0721	0.0117	0.0366	0.1302	**0.0116**
60%	0.1740	0.0755	0.0229	0.0499	0.1533	**0.0176**

TABLE 5. Comparison of restoration results in terms of PSNR for test images.

Images	Noise Ratio	PSMF	ROLD-EPR	ASWM	ROR-NLM	ANN	GIRAF	ALOHA	Proposed
Lena	20%	30.53	33.86	34.71	32.87	32.60	33.92	35.24	**39.15**
	40%	24.94	31.09	31.04	29.57	30.32	31.16	28.72	**33.17**
	60%	19.52	28.12	24.81	23.76	27.22	27.28	22.10	**28.29**
Plane	20%	29.07	29.41	29.47	28.15	28.96	30.47	33.98	**37.37**
	40%	22.92	27.29	27.72	26.38	27.51	28.08	26.96	**31.30**
	60%	17.16	25.65	22.45	20.87	25.03	24.76	18.26	**26.05**
Boat	20%	30.25	32.10	32.50	30.54	31.27	32.88	34.67	**37.99**
	40%	24.57	29.41	29.41	27.98	28.76	29.80	27.44	**32.05**
	60%	19.06	26.70	23.69	22.58	25.89	26.09	21.21	**27.32**
Man	20%	29.69	31.57	31.41	29.97	30.92	31.23	31.90	**35.78**
	40%	24.38	28.81	28.84	27.60	28.18	28.76	26.50	**30.64**
	60%	19.11	26.49	23.85	22.86	25.97	25.54	21.23	**26.53**
Couple	20%	28.60	29.80	29.63	27.60	29.83	29.76	31.56	**35.72**
	40%	23.73	27.24	27.09	25.53	26.85	27.31	26.07	**30.06**
	60%	19.18	24.93	22.89	22.11	24.23	24.24	20.88	**25.63**

that the proposed predictor achieves the best RMSE values among all the compared methods for almost all the noise ratios, which means that our method has a comprehensively more accurate noise prediction capability.

D. COMPARISON OF IMAGE RESTORATION

In this subsection, we evaluate the restoration performance of our proposed method by comparing it with some state-of-the-art RVIN denoising algorithms, including PSMF [34], ROLD-EPR [11], ASWM [4], ROR-NLM [8], ANN [16], GIRAF [35], and ALOHA [39]. Note that as GIRAF [35] is an inpainting algorithm that only works well under the guidance of noise detection results, we provide the noise mask obtained by our noise detector for GIRAF. For the proposed algorithm, we utilize the noise ratio estimated by our NRP to select the most appropriate CNN model for denoising. We employ the peak signal-to-noise ratio (PSNR), structural similarity (SSIM) [41], and feature similarity (FSIM) [42] as the metrics of the quality of our restored images. For these three metrics, larger values indicate better image quality.

Tables 5–7 list the PSNR, SSIM, and FSIM values for the test images *Lena*, *Plane*, *Boat*, *Man*, and *Couple*, with 20%, 40%, and 60% RVIN corruption, respectively. For ease of comparison, the best values in each table are shown in bold. Table 5 reveals that the best PSNR values for all the test images are obtained by our method. Tables 6 and 7 show that, for SSIM and FSIM, our method achieves the majority of the best values.

For a fair comparison, the pre-trained DnCNN models are utilized for RVIN denoising, according to the noise ratios estimated by different methods (Table 8). It is clear from Table 8 that even with the support of DnCNN models, the denoising results obtained by the conventional methods still cannot compete with the results obtained by ours. The reason for this is that our NRP gives predictions that are superior to those of the compared conventional methods, benefiting from its robust and reliable noise detector, which ensures that we can match the most appropriate DnCNN model for denoising.

For a more intuitive comparison, the denoised images of *Plane*, *Man*, and *Boat* obtained by these methods are shown in Figs. 10–12. We also provide a magnified view of each image in order to give a better comparison. Fig. 10 shows the restored image of *Plane* corrupted by 20% RVIN. It can be seen that the proposed method is effective in preserving fine details, such as the signs on the fuselage of the plane. The 40% RVIN restored results of *Man* in Fig. 11 illustrate that

TABLE 6. Comparison of restoration results in terms of SSIM for test images.

Images	Noise Ratio	PSMF	ROLD-EPR	ASWM	ROR-NLM	ANN	GIRAF	ALOHA	Proposed
Lena	20%	0.7979	0.9498	0.9617	0.9533	0.9204	0.9674	0.9531	**0.9688**
	40%	0.5694	0.9104	0.9159	0.8773	0.8747	**0.9298**	0.8192	0.8916
	60%	0.3114	0.8275	0.7391	0.6328	0.7987	**0.8334**	0.5920	0.7931
Plane	20%	0.7757	0.9367	0.9667	0.9487	0.9214	0.9678	0.9546	**0.9695**
	40%	0.4999	0.9017	0.9182	0.8594	0.8760	**0.9313**	0.7876	0.8820
	60%	0.2481	0.8204	0.6538	0.4946	0.7957	**0.8306**	0.3426	0.7571
Boat	20%	0.8367	0.9316	0.9403	0.9247	0.9124	0.9524	0.9560	**0.9588**
	40%	0.5905	0.8804	0.8830	0.8477	0.8504	**0.9178**	0.8016	0.8782
	60%	0.3270	0.7821	0.6984	0.6058	0.7563	**0.8093**	0.5691	0.7599
Man	20%	0.8344	0.9270	0.9222	0.9102	0.9085	0.9380	0.9170	**0.9619**
	40%	0.6056	0.8652	0.8585	0.8241	0.8441	0.8607	0.7298	**0.8697**
	60%	0.3383	**0.7634**	0.6683	0.5802	0.7462	0.7610	0.4764	0.7509
Couple	20%	0.8467	0.9093	0.9018	0.8724	0.9034	0.9326	0.9275	**0.9672**
	40%	0.6214	0.8402	0.8291	0.7787	0.8210	0.8631	0.7772	**0.8822**
	60%	0.3611	0.7301	0.6550	0.5785	0.6987	0.7282	0.5330	**0.7415**

TABLE 7. Comparison of restoration results in terms of FSIM for test images.

Images	Noise Ratio	PSMF	ROLD-EPR	ASWM	ROR-NLM	ANN	GIRAF	ALOHA	Proposed
Lena	20%	0.9749	0.9907	0.9925	0.9881	0.9883	0.9916	0.9920	**0.9958**
	40%	0.9061	0.9774	0.9785	0.9676	0.9750	0.9803	0.9614	**0.9815**
	60%	0.7465	**0.9443**	0.8973	0.8575	0.9416	0.9433	0.8737	0.9430
Plane	20%	0.9706	0.9790	0.9801	0.9709	0.9783	0.9833	0.9868	**0.9945**
	40%	0.8771	0.9586	0.9625	0.9424	0.9615	0.9665	0.9354	**0.9746**
	60%	0.6797	**0.9210**	0.8341	0.7595	0.9140	0.9136	0.7704	0.9180
Boat	20%	0.9744	0.9845	0.9844	0.9746	0.9829	0.9864	0.9883	**0.9941**
	40%	0.8949	0.9611	0.9637	0.9443	0.9623	0.9690	0.9342	**0.9766**
	60%	0.7289	0.9148	0.8514	0.8075	0.9101	0.9175	0.8156	**0.9246**
Man	20%	0.9739	0.9852	0.9838	0.9775	0.9843	0.9842	0.9826	**0.9931**
	40%	0.9028	0.9643	0.9650	0.9510	0.9611	0.9683	0.9334	**0.9741**
	60%	0.7536	0.9255	0.8764	0.8311	0.9246	0.9230	0.8276	**0.9286**
Couple	20%	0.9716	0.9791	0.9743	0.9588	0.9811	0.9771	0.9823	**0.9933**
	40%	0.9101	0.9532	0.9496	0.9261	0.9528	0.9540	0.9332	**0.9720**
	60%	0.7865	0.9062	0.8725	0.8386	0.8975	0.9030	0.8236	**0.9178**

our DnCNN model preserves the edges well while restoring the picture. To verify that our method also has excellent denoising capability at high noise ratios, the restored images of *Boat* corrupted by 60% RVIN are compared in Fig. 12. It is evident that the other denoising methods cannot suppress noise completely. In the images restored by PSMF, ASWM, and ROR-NLM, not only have a large number of noise pixels not been removed, but there has been extensive loss of detail in the images. For ANN and ROLD-EPR, there still exist some visible noise pixels that may result from the use of an imperfect noise detector. ALOHA produces over-smooth edges and textures, which leads to image blurring. As can be seen, there are still some noise pixels remaining in the result from GIRAF. The reason for this is that the denoising process of GIRAF relies on a noise mask, which means that only the identified noise pixels are processed, while the unidentified

TABLE 8. Comparison of restoration results of PSNR produced by using DnCNN models according to the noise ratios estimated by different methods.

Method	PSMF	ROLD-EPR	ASWM	ROR-NLM	ANN	Proposed
Lena						
10%	42.75	42.77	42.83	42.66	42.83	**43.04**
20%	35.24	39.06	39.10	39.05	38.70	**39.15**
30%	33.83	36.68	**36.72**	36.63	36.62	36.68
40%	32.07	33.12	33.15	33.16	32.98	**33.17**
50%	25.84	29.43	30.83	29.42	31.06	**31.28**
60%	24.01	27.23	27.94	27.07	27.10	**28.29**
Plane						
10%	41.98	42.00	42.04	42.24	42.39	**42.43**
20%	33.64	36.33	36.45	36.57	37.07	**37.37**
30%	32.15	34.28	34.42	34.39	34.42	**34.80**
40%	30.38	30.39	31.23	31.23	31.19	**31.30**
50%	25.14	27.91	28.63	28.85	28.72	**28.96**
60%	22.80	25.54	25.63	25.76	25.62	**26.05**
Couple						
10%	39.80	39.87	**39.94**	39.54	39.44	39.89
20%	35.66	35.70	35.59	35.65	35.58	**35.72**
30%	31.00	32.97	32.94	33.07	33.12	**33.24**
40%	29.21	29.56	**30.17**	30.06	30.14	30.06
50%	26.94	26.98	27.92	28.12	28.07	**28.14**
60%	22.79	25.01	25.57	24.95	25.50	**25.63**

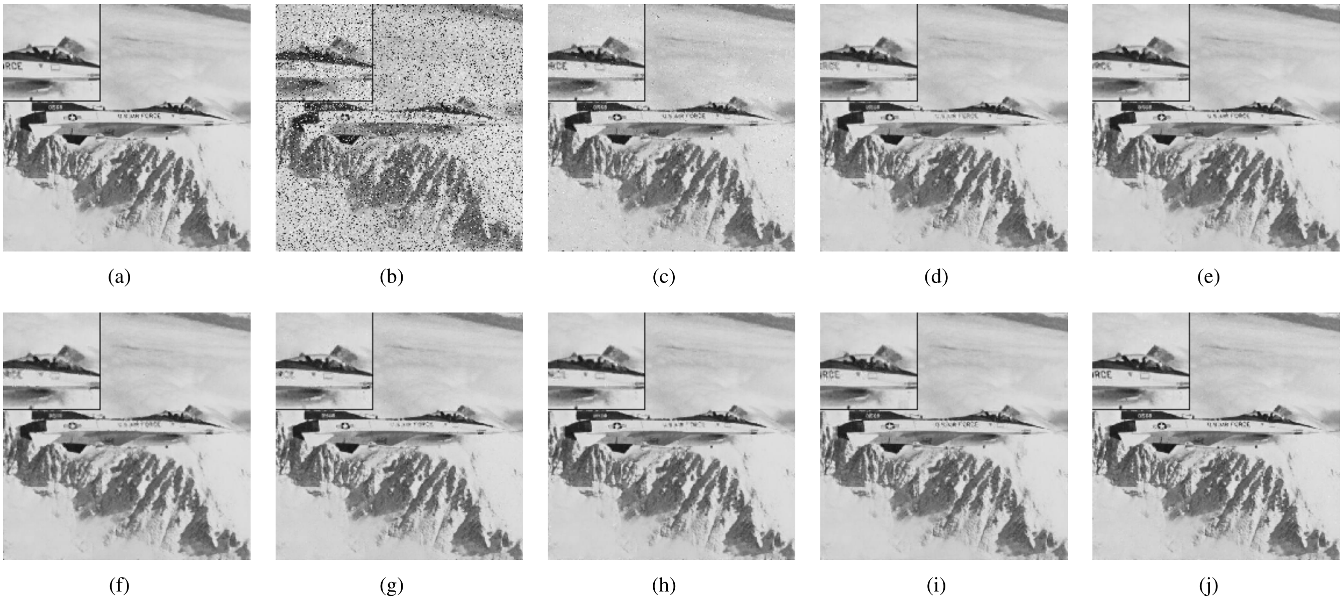


FIGURE 10. Comparison of restoration results of different methods for image *Plane* corrupted by 20% RVIN. (a) Original image. (b) 20% noise. (c) PSMF, PSNR = 29.07 dB. (d) ROLD-EPR, PSNR = 29.41 dB. (e) ASWM, PSNR = 29.47 dB. (f) ROR-NLM, PSNR = 28.15 dB. (g) ANN, PSNR = 28.96 dB. (h) GIRAF, PSNR = 30.47 dB. (i) ALOHA, PSNR = 33.98 dB. (j) Proposed, PSNR = 37.37 dB.

ones remain unchanged. By comparison, the image generated by the proposed method not only removes almost all the noise, but also retains image details to the greatest extent, even at high noise ratios. One reason for this is that our NRP converts the noise mask into a noise ratio, and, according to this ratio, we select the most appropriate CNN model for denoising, rather than restoring image by removing the detected RVIN noise pixel-by-pixel.

E. COMPARISON OF RUNNING TIME

In the last part of this section, we compare the running time for noise prediction and image restoration with 512 × 512 *Lena* for all of the above algorithms. All the experiments were conducted on the same computing platform. From Table 9, it is apparent that our proposed method takes the shortest time for noise prediction. The reason for this is that the proposed predictor takes image patches from

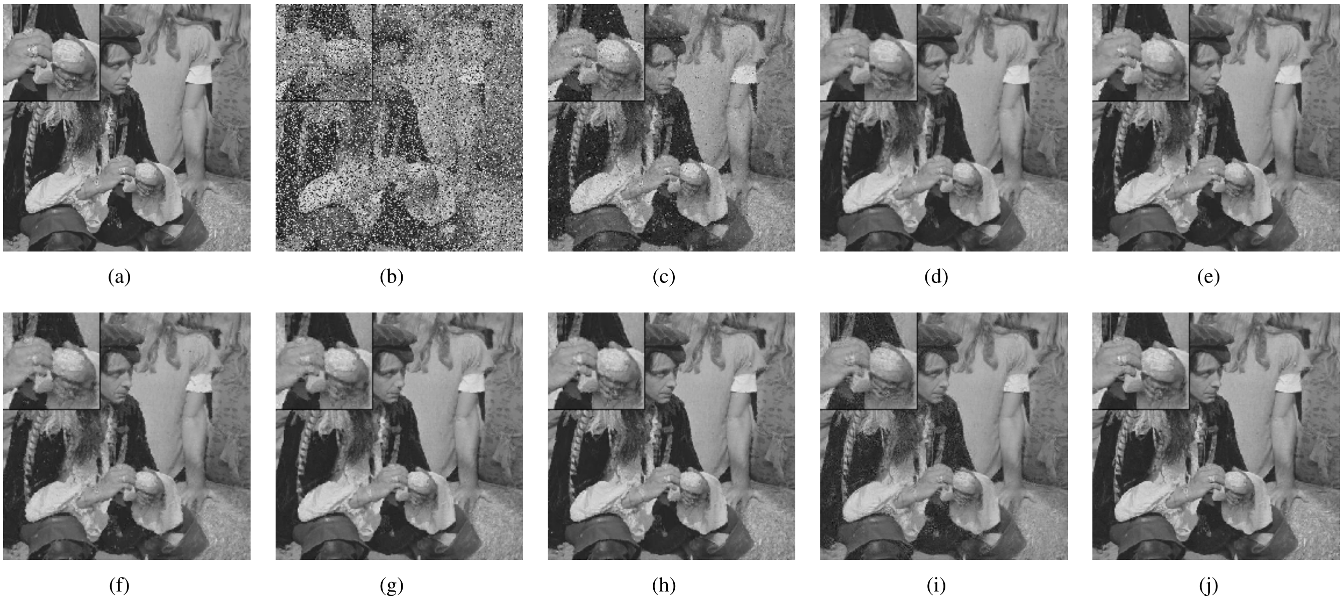


FIGURE 11. Comparison of restoration results of different methods for image *Man* corrupted by 40% RVIN. (a) Original image. (b) 40% noise. (c) PSMF, PSNR = 24.38 dB. (d) ROLD-EPR, PSNR = 28.81 dB. (e) ASWM, PSNR = 28.84 dB. (f) ROR-NLM, PSNR = 27.60 dB. (g) ANN, PSNR = 28.18 dB. (h) GIRAF, PSNR = 28.76 dB. (i) ALOHA, PSNR = 26.50 dB. (j) Proposed, PSNR = 30.64 dB.

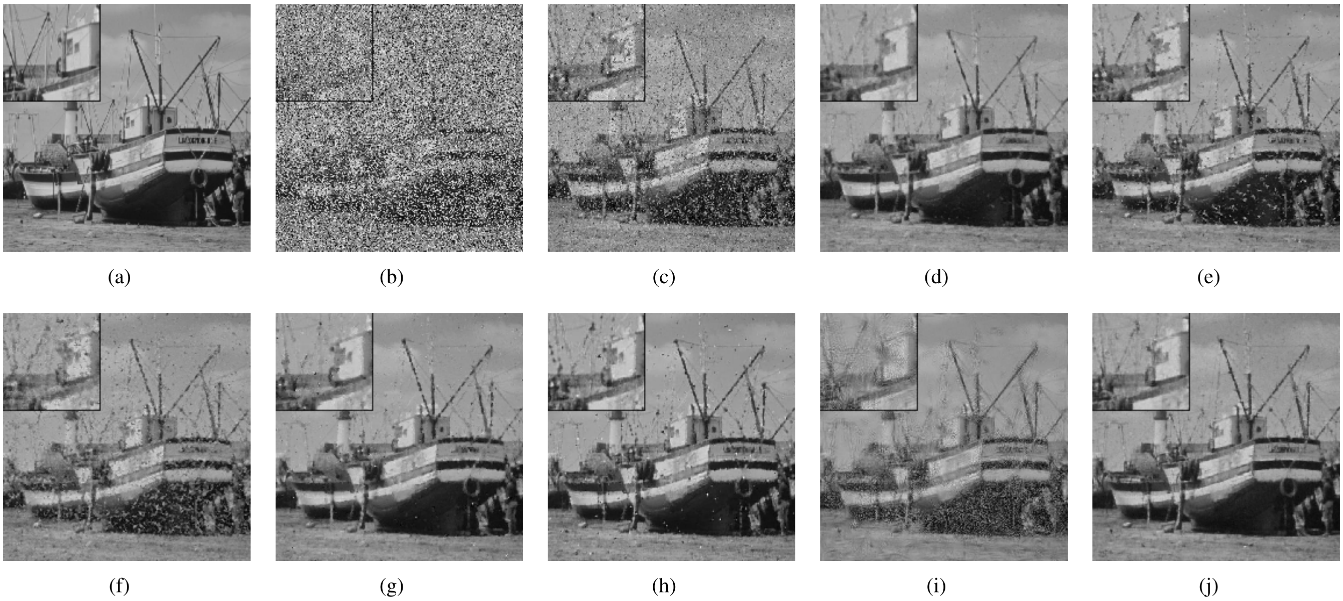


FIGURE 12. Comparison of restoration results of different methods for image *Boat* corrupted by 60% RVIN. (a) Original image. (b) 60% noise. (c) PSMF, PSNR = 19.06 dB. (d) ROLD-EPR, PSNR = 26.70 dB. (e) ASWM, PSNR = 23.69 dB. (f) ROR-NLM, PSNR = 22.58 dB. (g) ANN, PSNR = 25.89 dB. (h) GIRAF, PSNR = 26.09 dB. (i) ALOHA, PSNR = 21.21 dB. (j) Proposed, PSNR = 27.32 dB.

the test image at random to obtain an overall perception, thereby avoiding the traditional time-consuming iterative detection.

As shown in Table 10, our DnCNN model takes less CPU time for denoising than the majority of methods, except for PSMF and ANN. Although PSMF takes the shortest time for restoration, it does not give satisfactory results, which makes its apparent high efficiency worthless. For iterative filters, the time consumption grows with noise ratio. For instance, although ANN has a competitive time consumption for restoration at low noise ratios, its time consumption increases rapidly as the noise ratio increases and ends up higher than ours at 60% noise ratio (Table 10). Benefiting from parallel computation on the GPU, our DnCNN model takes much less time for denoising.

In summary, in comparison with other methods, the proposed method maintains a short and stable time consumption on CPU and a much shorter time consumption on GPU, as well as providing state-of-the-art denoising performance at both low and high noise ratios.

TABLE 9. Comparison of detection time consumption on CPU with *Lena* (in seconds).

	Noise Ratio					
Method	10%	20%	30%	40%	50%	60%
PSMF	0.18	0.15	0.17	0.16	0.17	0.15
ROLD-EPR	7.04	6.63	6.73	6.08	6.79	10.68
ASWM	54.75	58.04	60.71	63.87	66.84	68.76
ROR-NLM	3.84	3.88	3.93	3.98	3.89	3.90
ANN	0.96	1.04	1.02	1.04	1.03	1.02
Proposed	0.02	0.03	0.02	0.02	0.02	0.03

TABLE 10. Comparison of restoration time consumption with *Lena* (in seconds).

		Noise Ratio					
Method	Execution environment	10%	20%	30%	40%	50%	60%
PSMF	CPU	0.05	0.05	0.07	0.07	0.07	0.07
ROLD-EPR	CPU	7.04	7.63	7.73	7.08	7.79	11.68
ASWM	CPU	54.75	58.04	60.71	63.87	66.84	68.76
ROR-NLM	CPU	7.89	12.85	18.98	23.7	28.45	33.42
ANN	CPU	0.57	0.83	0.98	3.6	5.91	15.26
GIRAF	CPU	212.25	211	211.06	210.1	209.26	205.95
ALOHA	GPU	1438.57	1570.8	1631.74	1366.32	1187.81	1225.81
Proposed	CPU	6.36	6.37	6.86	6.34	6.38	6.75
Proposed	GPU	0.03	0.03	0.02	0.02	0.03	0.03

V. CONCLUSION

In this paper, we propose a blind CNN denoising model with a noise ratio predictor. Taking the NRP output as an indicator of accurate prediction to adopt the most appropriate CNN denoising model, we find that our method can achieve perceptually appealing denoising results. Unlike traditional CNN denoising models that are tailored for a specific noise ratio, our adaptive DnCNN model is able to remove RVIN with unknown noise ratios, owing to its incorporation of the NRP. By adopting the ROAD$_n$ $(n = 1, \ldots, 12)$ statistics and a new measure, CPMD, to describe the difference between noise pixels and clean pixels, as well as an edge-preserving scheme (EPD), our NRP can provide an accurate noise ratio for the test image instantly by selecting patches randomly from this image. It is then possible to utilize the most appropriate DnCNN model for RVIN denoising. Simulation results indicate that our adaptive denoising method not only has strong flexibility in dealing with unknown noise ratios, but also provides favorable denoising performance effectively and efficiently.

REFERENCES

[1] T. Nodes and N. Gallagher, "Median filters: Some modifications and their properties," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-30, no. 5, pp. 739–746, Oct. 1982.

[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.

[3] R. C. Hardie and C. Boncelet, "LUM filters: A class of rank-order-based filters for smoothing and sharpening," *IEEE Trans. Signal Process.*, vol. 41, no. 3, pp. 1061–1076, Mar. 1993.

[4] S. Akkoul, R. Lédée, R. Leconge, and R. Harba, "A new adaptive switching median filter," *IEEE Signal Process. Lett.*, vol. 17, no. 6, pp. 587–590, Jun. 2010.

[5] Y. Dong and S. Xu, "A new directional weighted median filter for removal of random-valued impulse noise," *IEEE Signal Process. Lett.*, vol. 14, no. 3, pp. 193–196, Mar. 2007.

[6] D. Kuykin, V. Khryashchev, and I. Apalkov, "Modified progressive switched median filter for image enhancement," in *Proc. Int. Conf. Comput. Graph. Vis.*, Aug. 2009, pp. 303–304.

[7] T. Chen and H. R. Wu, "Adaptive impulse detection using center-weighted median filters," *IEEE Signal Process. Lett.*, vol. 8, no. 1, pp. 1–3, Jan. 2001.

[8] 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.

[9] W. Luo, "A new efficient impulse detection algorithm for the removal of impulse noise," *IEICE Trans. Fundam. Electron.*, vol. E88-A, no. 10, pp. 2579–2586, Oct. 2005.

[10] 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.

[11] 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.

[12] M. Nikolova, "A variational approach to remove outliers and impulse noise," *J. Math. Imag. Vis.*, vol. 20, nos. 1–2, pp. 99–120, Jan. 2004.

[13] F. Russo and G. Ramponi, "A fuzzy filter for images corrupted by impulse noise," *IEEE Signal Process. Lett.*, vol. 3, no. 6, pp. 168–170, Jun. 1996.

[14] P. Sa and B. Majhi, "An improved adaptive impulsive noise suppression scheme for digital images," *Int. J. Electron. Commun.*, vol. 64, no. 4, pp. 322–328, Apr. 2010.

[15] V. Crnojevic, V. Senk, and Z. Trpovski, "Advanced impulse detection based on pixel-wise MAD," *IEEE Signal Process. Lett.*, vol. 11, no. 7, pp. 589–592, Jul. 2004.

[16] I. Turkmen, "The ANN based detector to remove random-valued impulse noise in images," *J. Vis. Commun. Image Represent.*, vol. 34, pp. 28–36, Jan. 2016.

[17] M. S. Darus, S. N. Sulaiman, I. S. Isa, Z. Hussain, N. M. Tahir, and N. A. M. Isa, "Modified hybrid median filter for removal of low density random-valued impulse noise in images," in *Proc. 6th IEEE Int. Conf. Control Syst., Comput. Eng. (ICCSCE)*, Nov. 2016, pp. 528–533.

[18] S. Soleimany and M. Hamghalam, "A novel random-valued impulse noise detector based on MLP neural network classifier," in *Proc. Artif. Intell. Robot. (IRANOPEN)*, Apr. 2017, pp. 165–169.

[19] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017.

[20] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[21] D. Xie, J. Xiong, and S. Pu, "All you need is beyond a good init: Exploring better solution for training extremely deep convolutional neural networks with orthonormality and modulation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6176–6185.

[22] S. Zagoruyko and N. Komodakis, "DiracNets: Training very deep neural networks without skip-connections," Jun. 2017, *arXiv:1706.00388.* [Online]. Available: https://arxiv.org/abs/1706.00388

[23] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn. (ICML)*, Jul. 2015, pp. 448–456.

[24] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," in *Proc. Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1026–1034.

[25] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, Jun. 2015, pp. 1–9.

[26] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *J. Mach. Learn. Res.*, vol. 12, pp. 2121–2159, Feb. 2011.

[27] M. D. Zeiler, "ADADELTA: An adaptive learning rate method," Dec. 2012, *arXiv:1212.5701.* [Online]. Available: https://arxiv.org/abs/1212.5701

[28] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," Jan. 2017, *arXiv:1412.6980.* [Online]. Available: https://arxiv.org/abs/1412.6980

[29] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, Apr. 2015, pp. 1–14.

[30] K. Zhang, W. Zuo, S. Gu, and L. Zhang, "Learning deep CNN denoiser prior for image restoration," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3929–3938.

[31] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," in *Proc. 4th Int. Conf. Learn. Represent. (ICLR)*, May 2016, pp. 1–13.

[32] X. Wang, Q. Tao, L. Wang, D. Li, and M. Zhang, "Deep convolutional architecture for natural image denoising," in *Proc. Int. Conf. Wireless Commun. Signal Process. (WCSP)*, Oct. 2015, pp. 1–4.

[33] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a Gaussian Denoiser: Residual learning of deep CNN for image denoising," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3142–3155, Jul. 2017.

[34] Z. Wang and D. Zhang, "Progressive switching median filter for the removal of impulse noise from highly corrupted images," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 46, no. 1, pp. 78–80, Jan. 1999.

[35] G. Ongie and M. Jacob, "A fast algorithm for convolutional structured low-rank matrix recovery," *IEEE Trans. Comput. Imag.*, vol. 3, no. 4, pp. 535–550, Dec. 2017.

[36] K. H. Jin and J. C. Ye, "Annihilating filter-based low-rank Hankel matrix approach for image inpainting," *IEEE Trans. Image Process.*, vol. 24, no. 11, pp. 3498–3511, Nov. 2015.

[37] J. C. Ye, J. M. Kim, K. H. Jin, and K. Lee, "Compressive sampling using annihilating filter-based low-rank interpolation," *IEEE Trans. Inf. Theory*, vol. 63, no. 2, pp. 777–801, Feb. 2017.

[38] K. H. Jin, D. Lee, and J. C. Ye, "A general framework for compressed sensing and parallel MRI using annihilating filter based low-rank Hankel matrix," *IEEE Trans. Comput. Imag.*, vol. 2, no. 4, pp. 480–495, Dec. 2016.

[39] K. H. Jin and J. C. Ye, "Sparse and low-rank decomposition of a Hankel structured matrix for impulse noise removal," *IEEE Trans. Image Process.*, vol. 27, no. 3, pp. 1448–1461, Mar. 2018.

[40] P. Arbeláez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 898–916, May 2011.

[41] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[42] L. Zhang, L. Zhang, X. Mou, and D. Zhang, "FSIM: A feature similarity index for image quality assessment," *IEEE Trans. Image Process.*, vol. 20, no. 8, pp. 2378–2386, Aug. 2011.

JUNQING CHEN is currently pursuing the bachelor's degree with the School of Information Engineering, Nanchang University. His current research interests include image processing and neural networks.

GUIZHEN ZHANG received the B.Tech. degree from the Jiangxi University of Traditional Chinese Medicine, China, in July 2016. She is currently pursuing the M.S. degree with the School of Information Engineering, Nanchang University, China, under the supervision of Prof. S. Xu. Her research interests include digital image processing and machine learning.

SHAOPING XU received the M.S. degree in computer application from the China University of Geosciences, Wuhan, China, in 2004, and the Ph.D. degree in mechatronics engineering from the University of Nanchang, Nanchang, China, in 2010. He is currently a Professor with the Department of Computer Science, School of Information Engineering, Nanchang University. He has published more than 30 research articles. His current research interests include digital image processing and analysis, computer graphics, virtual reality, and surgery simulation. He serves as a Reviewer for several journals, including the IEEE TRANSACTIONS ON INSTRUMENTATION AND MEASUREMENT.

HAIWEN YU received the M.Eng. degree in computer application from Nanchang University, Jiangxi, China, in 2005, where she is currently a Lecturer with the Department of Computer Science, School of Information Engineering. She has published several research articles. Her current research interests include digital image processing and analysis, computer graphics, and neural networks. She has researched and directed research on a number of subjects and projects.