



# Image denoising via deep residual convolutional neural networks

Rushi Lan<sup>1</sup> · Haizhang Zou<sup>1</sup> · Cheng Pang<sup>1</sup> · Yanru Zhong<sup>1</sup> · Zhenbing Liu<sup>2</sup> · Xiaonan Luo<sup>3</sup>

Received: 4 March 2019 / Revised: 1 June 2019 / Accepted: 10 July 2019 / Published online: 6 August 2019  
© Springer-Verlag London Ltd., part of Springer Nature 2019

## Abstract

Recently, convolutional neural network (CNN)-based methods have achieved impressive performance on image denoising. Notably, CNN with deeper and thinner structures is more flexible to extract the image details. However, direct stacking some existing networks is difficult to achieve satisfactory denoising performance. In this paper, we propose a novel *deep residual convolutional neural network (DRCNN)* for image denoising. The main structure of DRCNN is the residual block that consists of two convolutional layers, and there are skip connections between these two convolutional layers without the batch normalization operation. The skip connection not only directly transfers the input image information to the hidden layer but also reduces the path length of gradient transfer, making the gradient transfer in a short path and alleviating the vanishing-gradient problem. DRCNN is compared with several state-of-the-art algorithms, and the experimental results demonstrated its denoising effectiveness.

**Keywords** Image denoising · Convolutional neural network · Residual learning

## 1 Introduction

Image denoising aims to recover a clean image from a noisy observation. It is a fundamental research topic in the fields of image processing and computer vision because it benefits many high-level applications, such as bioinformatics [14, 19, 25], image encryption [13, 15, 18], texture classification [16, 26], and many others [30–34]. According to the generation mechanism, there are different kinds of image noise, such as additive white Gaussian noise (AWGN), impulse noise, salt and pepper noise, and Poisson noise. In this work, our attention is focused on removing the AWGN because it is the most common noise that corrupts images in practice. Let  $u(x)$  be a clean (noise-free) image. The noised image is generated as follows:

$$v(x) = u(x) + n(x), \quad (1)$$

where  $v(x)$  is the noised version of  $u(x)$  and  $n(x)$  is the noise added to  $v(x)$ . In this situation,  $n(x)$  follows the Gaussian distribution, namely  $n(x) \sim N(\mu, \sigma^2)$ .  $\mu$  and  $\sigma^2$  represent the mean and variance of the noise.

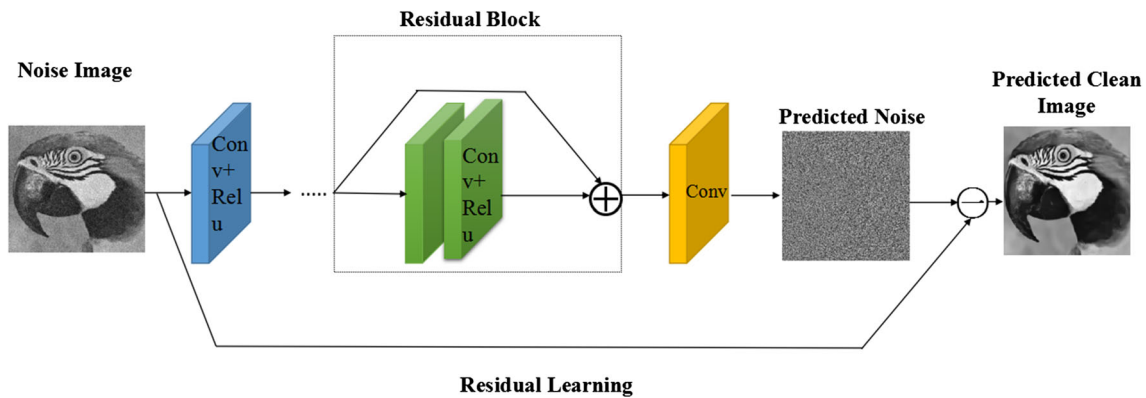
In the last two decades, numerous and diverse image denoising algorithms have been developed from various perspectives, such as image filtering, shrinkage of coefficients, sparse representation of a learned dictionary, and non-local self-similarity statistics. Representation methods include bilateral filtering [29], non-local means (NLM) [2], block matching and 3D filtering (BM3D) [4], K-SVD [6], higher-order singular value decomposition (HOSVD) [22], and weighted nuclear norm minimization (WNNM) [7]. To further improve the performance of the aforementioned approaches, the authors in [23] proposed a scheme called cascade of shrinkage fields (CSF) for image denoising, which was a kind of unified random field model. Recently, Chen and Pock [3] developed a trainable nonlinear reaction diffusion (TNRD) model. CSF and TNRD employ a large amount of prior knowledge of images and then use the forward propagation to optimize the network. Although CSF and TNRD are able to reduce the computational efficiency and improve the quality of denoising, they need to train a specific model to determine the characteristic noise, which is not universally suitable for image denoising.

✉ Cheng Pang  
pangcheng3@guet.edu.cn

<sup>1</sup> Guangxi Key Laboratory of Image and Graphic Intelligent Processing, Guilin University of Electronic Technology, Guilin 541004, China

<sup>2</sup> Guangxi Key Laboratory of Trusted Software, Guilin University of Electronic Technology, Guilin 541004, China

<sup>3</sup> National and Local Joint Engineering Research Center of Satellite Navigation and Location Service, Guilin University of Electronic Technology, Guilin 541004, China

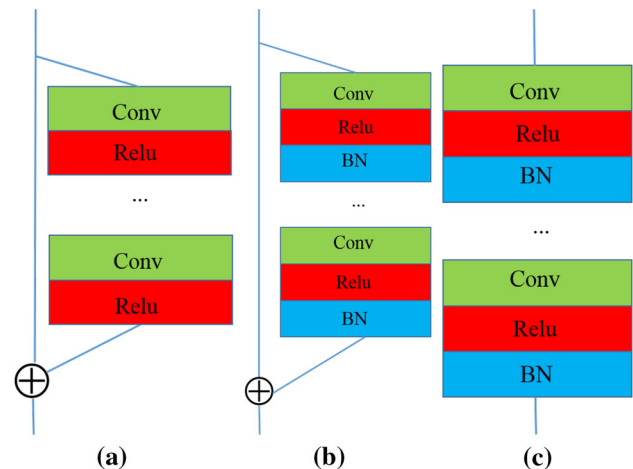


**Fig. 1** The main architecture of the proposed DRCNN network. Learning potential clean images by residual learning

Inspired by the excellent success of deep learning models in diverse vision applications, especially image super-resolution, an increasing number of researchers have attempted to employ the deep learning techniques for image denoising. The multilayer perceptron (MLP) [8] and the stack denoising auto-encoder (SDA) convolutional neural network [27] are the first two denoising algorithms to use deep learning techniques, and they have achieved performance comparable to that of the representative BM3D. However, the layer number of MLP and SDA networks is shallow because of the gradient vanishing over the depth network, which limits their performance. Mao et al. [20] proposed a very deep convolutional encoder–decoder network for image restoration. By skip connecting the encoding layer and the decoding layer, the number of network layers reached to 30, and the proposed network achieved satisfactory performance in both image denoising and super-resolution. Recently, a novel method named DnCNN was proposed for image denoising; this method contains 17 convolutional layers and takes into account the residual learning technique [28]. DnCNN not only converges quickly but also significantly improves the performance of previous algorithms.

Although DnCNN has achieved impressive denoising performance, the layer number of DnCNN is still not deep enough. A deeper network is possible to achieve better performance. In order to improve the denoising capacity, we must deepen the network. However, the gradient will vanish as the neural network deepens. It is necessary to introduce other techniques to avoid the conflict between deep layers and gradient vanishing.

In order to alleviate the gradient vanishing caused by deepening the network depth, we propose a novel image denoising method named *deep residual convolutional neural network (DRCNN)*, which is based on the DnCNN and the ResNet [9] architecture. We first optimize it by analyzing and removing unnecessary modules to simplify the network architecture and then use skip connections and residual learning strategies to alleviate the gradient vanishing and



**Fig. 2** Comparison of different core structures. **a** Ours, **b** ResNet, and **c** DnCNN

accelerate the convergence of the network. The proposed method has been evaluated on publicly available benchmark datasets [21] and outperforms the current state-of-the-art approaches.

The main contributions of this work can be summarized as follows. First, we design a deep residual convolutional neural network that uses skip connections between the convolution layers to form a residual block, which can then alleviate the problem of the gradient vanishing and network performance degradation due to the excessive depth of the network layer. Second, we introduce the residual learning and simplify the network structure by removing the BN [10] layer. The network can converge very quickly when the number of network layers is very deep. In addition, this enhances the denoising performance in PSNR value and also has a good visual effect.

The rest of this paper is organized as follows. Section 2 provides some related works. Section 3 introduces the proposed method in detail. Several experimental results are presented in Sect. 4. Section 5 finally gives the conclusion.



Fig. 3 Twelve widely used testing images

## 2 Related work

In this section, we briefly introduce two related techniques that are used in the proposed method.

### 2.1 Skip connection

The receptive field is the size of the unit extracted from the original input image. The deeper the network and the larger the receptive field, the better the effect of image feature extraction is. As the number of network layers increases, it becomes easier for the gradient to vanish while training the convolution neural network, resulting in network degradation.

To alleviate the gradient vanishing, Srivastava et al. [24] put forward the skip connection method. The  $i$  layer is connected directly to the  $i + n$  ( $n > 1$ ) layer and is applied to the high-speed network (highway networks). By skip connection, the number of highway network layers is more than 100 layers, and there is no network degradation. He et al. [9] proposed a residual network (ResNet), in which the fitted mapping,  $H(x)$ , is expressed as  $H(x) = F(x) + x$ , where  $F(x)$  is called the residual mapping and  $x$  is the input signal. By skip connection, the learning of  $H(x)$  is transformed into  $F(x)$  learning. The authors proved that  $F(x)$  is more easy to learn than  $H(x)$ .

### 2.2 Residual learning

Direct fitting of clean pictures sets up lowly when setting the learning rate, which then leads to an excessive convergence time or difficulties in converging. VDSR [11] proposes a residual learning strategy, which defines a residual image as  $r = y - x$ , in which it is quicker to fit the  $r$  than to fit the  $x$ , and the learning rate is 1000 times that of SRCNN [5], which greatly accelerates the convergence and performs well on the super-resolution. DnCNN also uses a residual learning strategy to directly fit noise pictures, and achieves very good denoising effects.



Fig. 4 Twelve example images from the BSD68 data set

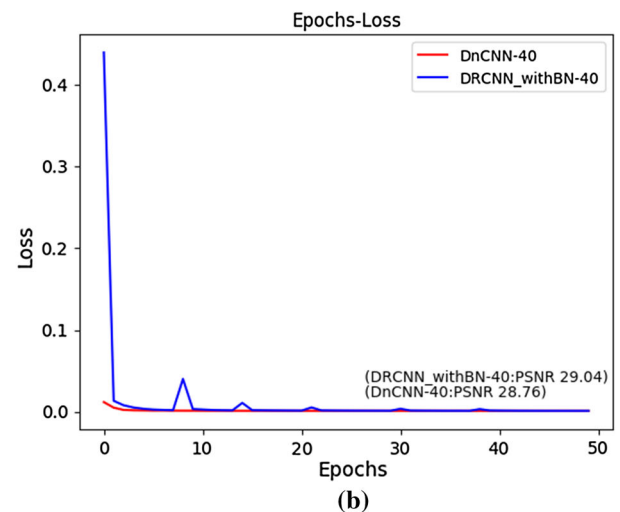
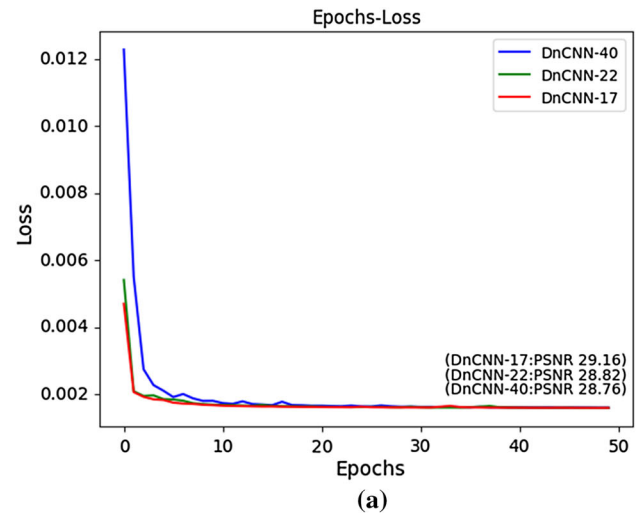


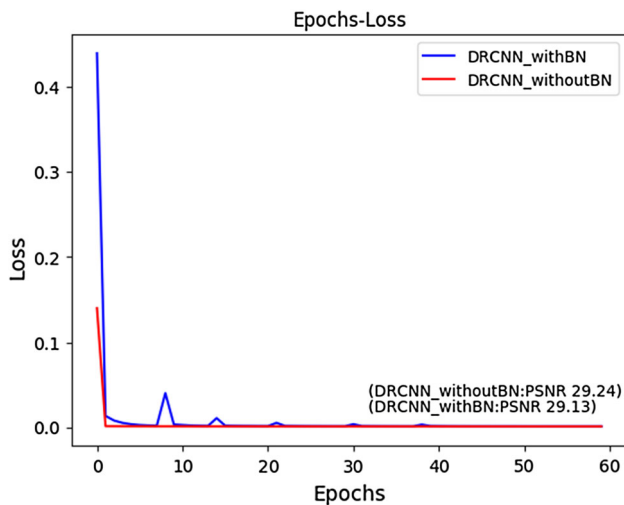
Fig. 5 **a** is the loss curve of DnCNN-17, DnCNN-22, and DnCNN-40. **b** is the loss curve of DRCNN-withBN-40 and DnCNN-40. The results are evaluated on the BSD68 dataset, when the noise level is 25

## 3 Proposed method

This section presents the proposed DRCNN in detail, including the network structure and the training procedure of DRCNN.

### 3.1 Network structure

For image denoising, we use a very deep residual convolutional neural network inspired by ResNet. The configuration

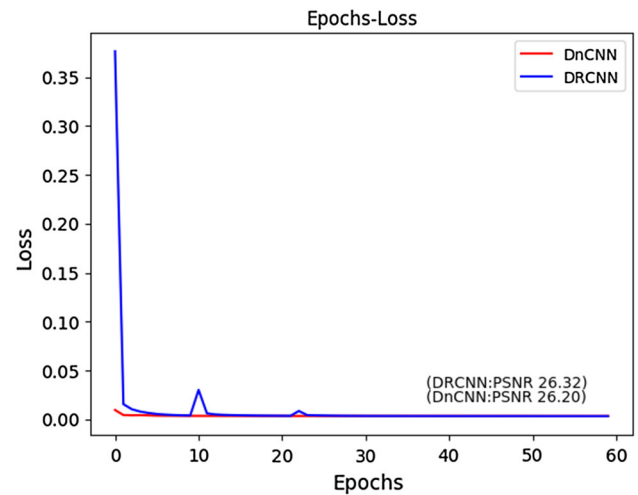


**Fig. 6** The loss curve of DRCNN–withBN and DRCNN–withoutBN (ours). The results are evaluated on the BSD68 dataset, When the noise level is 25

is outlined in Fig. 1. DRCNN is mainly composed of three parts and has 40 convolutional layers. The first part is mainly to learn the features of the noise image. It consists of a convolutional layer and a rectified linear unit (ReLU) activation layer, in which the convolution layer has 64 filters with a filter size of  $3 \times 3$ . The second part is made up of 19 residual blocks, each of which consists of two convolutional layers, and each convolutional layer has a ReLU layer for nonlinear mapping. Each convolutional layer is composed of  $64 \times 3 \times 3$  filters. The third part is made up of a convolutional layer, which is a clean image for processing output, consisting of a  $3 \times 3$  filter. If the output image is a color image, the number of filters is 3.

The input information will transfer many convolutional layers. As the information transmission path becomes longer, it is easy to cause the gradient vanishing/explosion. In the image denoising, the input image is very similar to the output image, so the difference between the input and output images is very small or 0 [11]. Fitting these values is easier to converge than the direct fit of the clean image. By subtracting the noisy image from the predicted noise image, we can obtain the predicted clean image.

The main structure of our network is the residual block. Residual networks exhibit excellent performance in computer vision problems, and our network structure is similar to ResNet. What is different is that we have removed the BN layer of each layer and simplified the neural network. The main component of the residual block is formed by two convolutional layers that are skip connected. Each convolutional layer directly carries out nonlinear mapping with a ReLU layer. We compare the difference between our residual blocks and DnCNN and ResNet core structures, as shown in Fig. 2.



**Fig. 7** The loss curves of DnCNN and DRCNN. The results are evaluated on the BSD68 dataset when the noise level is 50

### 3.2 Training

After completing the construction of the network, we need to train it to optimize the parameters in the network. In this paper, the training optimization method is Adam [12], and the number of training epochs is 60, where the first 50 epochs use a 0.001 learning rate and the second 10 epochs adopt a learning rate of 0.0001. We denoise at the noise levels of 25, 50, 75, and 100. The input of our DRCNN is a noisy observation,  $y = x + b$ . For DRCNN, the network uses the true noise image instead of the clean image as the label. In other words, we train a network to map  $R(y) = b$  instead of  $F(y) = x$ . We use the average mean square error (mse) as the cost function of the network, which can be represented as follows:

$$\text{loss}(\Theta) = \frac{1}{N} \sum_{i=1}^N \|R(y_i; \Theta) - (y_i - x_i)\|_F^2, \quad (2)$$

where  $\Theta$  represents the trainable parameters to be learned in DRCNN and  $(y_i, x_i)$  represents the  $i$ th noisy-clean training image pairs.  $R$  denotes the residual mapping to predict the residual image, and  $N$  is the number of total training images.

## 4 Experiments

In this section, we provide several experimental results to evaluate the proposed DRCNN. The setting of our experiments is first introduced. The studies of network degradation and batch normalization are then given. The comparison results with the state-of-the-art denoising algorithms are presented last.

**Table 1** Denoising performance of different algorithms on the BSD68 dataset in terms of PSNR

Methods	BM3D	WNNM	EPLL	MLP	CSF	TRND	DnCNN-S	DRCNN
$\sigma = 25$	28.57	28.83	28.68	28.95	28.74	28.92	29.19	<b>29.24</b>
$\sigma = 50$	25.62	25.87	25.67	26.03	–	25.97	26.20	<b>26.32</b>
$\sigma = 75$	24.21	24.39	24.09	24.57	–	–	24.65	<b>24.74</b>
$\sigma = 100$	23.24	23.38	23.06	–	–	–	23.64	<b>23.77</b>

The AGWN intensity  $\sigma$  here is set to 25, 50, 75, and 100, respectively. The best result is highlighted in bold

**Table 2** Denoising performance of different algorithms on 12 widely used testing images in terms of PSNR

Methods	C.man	House	Peppers	Starfish	Monar	Airpl	Parrot	Lena	Barbara	Boat	Man	Couple	Average
Noise	$\sigma = 25$												
BM3D	29.45	32.85	30.16	28.56	29.25	28.42	28.93	32.07	30.71	29.90	29.61	29.71	29.969
WNNM	29.64	33.22	30.42	29.03	29.84	28.69	29.15	32.24	<b>31.24</b>	30.03	29.76	29.82	30.257
EPLL	29.26	32.17	30.17	28.51	29.39	28.61	28.95	31.73	28.61	29.74	29.66	29.53	29.692
CSF	29.48	32.39	30.32	28.80	29.62	28.72	28.90	31.79	29.03	29.76	29.71	29.53	29.837
TNRD	29.72	32.53	30.57	29.02	29.85	28.88	29.18	32.00	29.41	29.91	29.87	29.71	30.055
DnCNN-S	<i>30.11</i>	<i>33.05</i>	<b>30.87</b>	<i>29.36</i>	<i>30.20</i>	<b>29.22</b>	<i>29.45</i>	<i>32.39</i>	<i>30.00</i>	<b>30.16</b>	<b>30.07</b>	<i>30.02</i>	<i>30.408</i>
DRCNN	<b>30.19</b>	<b>33.18</b>	<i>30.81</i>	<b>29.42</b>	<b>30.38</b>	<i>29.07</i>	<b>29.49</b>	<b>32.50</b>	29.96	<i>30.15</i>	<i>30.05</i>	<b>30.08</b>	<b>30.440</b>
Noise	$\sigma = 50$												
BM3D	26.13	29.69	26.68	25.04	25.82	25.10	25.90	29.05	27.22	26.78	26.81	26.46	26.722
WNNM	26.45	30.33	26.95	25.44	26.32	25.42	26.14	29.25	<b>27.79</b>	26.97	26.94	26.64	27.052
EPLL	26.10	29.12	26.80	25.12	25.94	25.31	25.95	28.68	24.83	26.74	26.79	26.30	26.471
MLP	26.37	29.64	26.68	25.43	26.36	25.56	26.12	29.32	25.24	27.03	27.06	26.67	26.783
TNRD	26.62	29.48	27.10	25.42	26.31	25.59	26.16	28.93	25.70	26.94	26.98	26.50	26.812
DnCNN-S	<i>26.98</i>	<i>30.09</i>	<i>27.33</i>	<i>25.66</i>	<i>26.82</i>	<i>25.85</i>	<i>26.44</i>	<i>29.42</i>	<i>26.22</i>	<i>27.11</i>	<i>27.19</i>	<i>26.83</i>	<i>27.160</i>
DRCNN	<b>27.16</b>	<b>30.66</b>	<b>27.59</b>	<b>25.88</b>	<b>27.13</b>	<b>25.92</b>	<b>26.57</b>	<b>29.62</b>	26.54	<b>27.30</b>	<b>27.31</b>	<b>27.09</b>	<b>27.368</b>
Noise	$\sigma = 75$												
BM3D	24.33	27.51	24.73	23.27	23.91	23.47	24.19	27.26	25.12	25.12	25.32	24.7	24.911
WNNM	24.55	28.24	24.92	23.47	24.31	23.74	24.37	27.54	<b>25.81</b>	25.30	25.42	24.86	25.211
EPLL	24.14	26.91	24.52	23.05	23.62	23.46	23.89	26.52	22.97	24.78	25.08	24.44	24.448
MLP	24.58	27.83	24.94	23.42	24.29	24.01	24.52	27.64	23.45	25.37	25.58	25.06	25.058
DnCNN-S	<i>25.28</i>	<i>27.76</i>	<i>25.33</i>	<b>23.53</b>	<i>24.64</i>	<i>24.08</i>	<b>24.70</b>	<i>27.67</i>	<i>24.15</i>	<i>25.49</i>	<i>25.62</i>	<i>25.01</i>	<i>25.272</i>
DRCNN	<b>25.34</b>	<b>28.48</b>	<b>25.53</b>	<i>23.50</i>	<b>24.92</b>	<b>24.13</b>	<i>24.67</i>	<b>27.74</b>	24.06	<b>25.50</b>	<b>25.72</b>	<b>25.20</b>	<b>25.399</b>
Noise	$\sigma = 100$												
BM3D	23.08	25.87	23.39	22.10	22.52	22.11	22.96	25.95	23.62	23.97	24.22	23.51	23.608
WNNM	23.36	26.66	<i>23.45</i>	<b>22.23</b>	22.95	22.55	23.19	26.21	24.37	24.11	24.36	23.56	23.917
EPLL	22.86	25.20	22.86	21.81	22.14	22.18	22.48	25.32	22.17	23.76	23.98	23.34	23.175
DnCNN-S	<i>24.02</i>	<i>26.53</i>	<b>23.95</b>	<i>22.17</i>	<b>23.39</b>	<b>22.99</b>	<i>23.35</i>	<i>26.31</i>	<b>22.93</b>	<i>24.34</i>	<i>24.63</i>	<i>23.88</i>	<i>24.041</i>
DRCNN	<b>24.05</b>	<b>27.15</b>	<b>23.95</b>	<i>22.21</i>	<i>23.30</i>	<i>22.92</i>	<b>23.45</b>	<b>26.52</b>	<i>22.71</i>	<b>24.38</b>	<b>24.74</b>	<b>24.08</b>	<b>24.122</b>

The AGWN intensity  $\sigma$  here is set to 25, 50, 75, and 100, respectively. The best two results are highlighted in bold and italics, respectively

## 4.1 Experimental setting

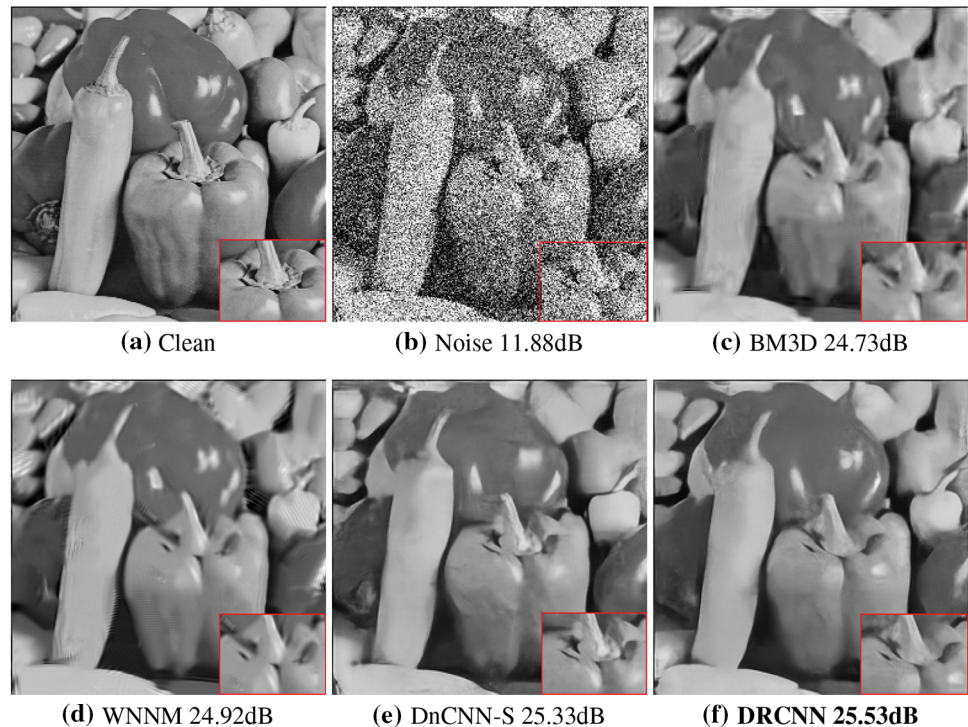
As a variant of the CNNs, DRCNN involves a large number of matrix calculations, resulting in a very high computation cost. To address this problem, the computation of DRCNN is performed on the Tesla P100 GPUs. A commonly used deep learning framework, Tensorflow [1], is utilized here. Similar to some representative previous works, the BSD400

dataset is selected as the training images. In this work, we follow the operations used in DnCNN to generate the  $40 \times 40$  image patches to train DRCNN. It takes approximately 14 h to complete the training procedure of DRCNN.

In our experiments, two sets of images are applied as test images to study the denoising performance of different competing methods. The first set is formed by 12 widely used images, such as Lena, Cameraman, House, Peppers, and



**Fig. 8** Denoising results of the image from Set12 with a noise level of 75



Barbara. All these images are shown in Fig. 3. The second dataset is the BSD68 that contains 68 different images. Figure 4 shows example images that are randomly selected from BSD68. It can be seen that all these images in the two sets include different types of image characteristics, such as textured and smooth regions; hence, they can be employed to comprehensively study all competing denoising methods.

In this work, the proposed DRCNN is compared with the following methods: BM3D, WNNM, EPLL, MLP, CSF, TNRD, and DnCNN, respectively. The noised images are generated via Eq. (1), and the noise intensity  $\sigma$  of the AWGN is set to 25, 50, 75, and 100, as in the previous literature. It is quite challenging to recover the original noise-free images as the noise intensity increases to a large value. Similarly, to quantitatively describe the denoising performance, the commonly used peak signal-to-noise ratio (PSNR) is applied here.

## 4.2 Study of network degradation

For image denoising, the deeper the network is, the greater the degradation of the network due to the vanishing of the gradient, and the worse the performance of the denoising. In this work, we use a network of different layers of the same structure. We deepened the number of layers of DnCNN to 22 and 40 and compared them with the DnCNN (17 convolutional layers). We selected the BSD400 dataset as the training set, trained at the noise level of 50, and used Adam as the optimization method. Referencing the training details

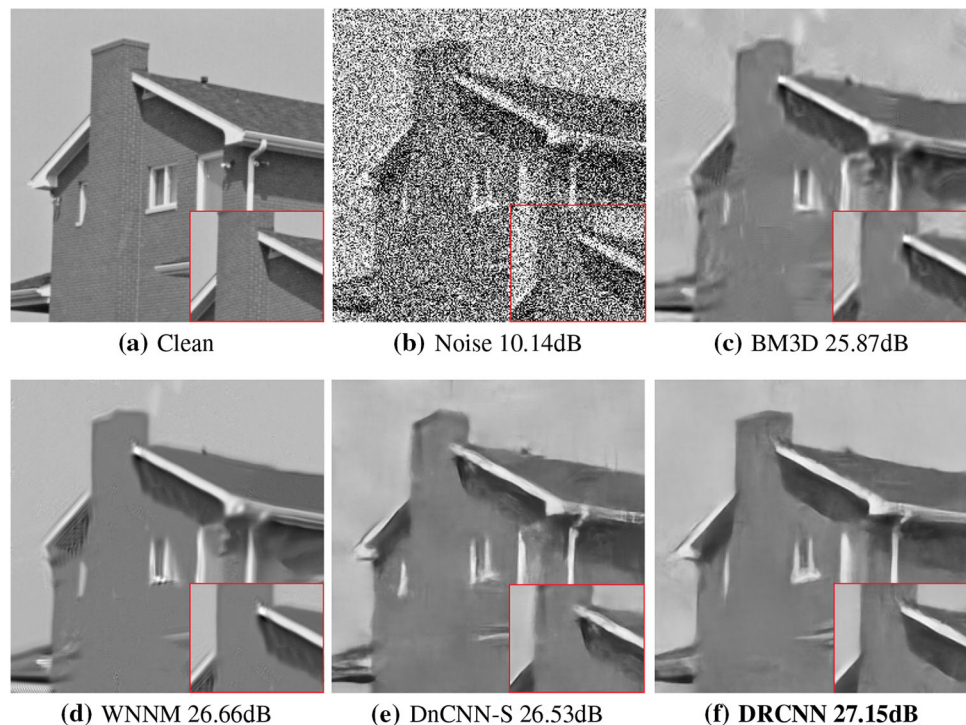
mentioned above, we trained 50 epochs and used BSD400 at a noise level of 25 to train and used BSD68 to evaluate the denoising performance. The results of the loss curve during training are shown in Fig. 5a. From Fig. 5a, we can see that the average PSNR of DnCNN-17 is the highest. We can now show that the deeper the network is, the lower the noise reduction performance.

The skip connection alleviates the network degradation problem. In the DRCNN network, we propose to add a BN layer named DRCNN-withBN. The difference between DRCNN-withBN and DnCNN-40 is that DRCNN-withBN has a skip connection on the network structure. In this work, we use DRCNN-withBN and DnCNN-40 to train 50 epochs, use BSD400 with a noise level of 25 to train, and then use BSD68 to evaluate the denoising performance. The results of the loss curve during training are shown in Fig. 5b. From Fig. 5b, we can see that the average PSNR of DRCNN-withBN is the higher than that of DnCNN-40. We can now prove that the skip connection can alleviate the vanishing of the gradient.

## 4.3 Study of batch normalization operation

The BN layer can accelerate convergence but also removes range flexibility from networks by normalizing the features [17]. In image denoising, the feature in each layer is unnecessary to be normalized. On the contrary, the BN layer will destroy the original image features. Removing the BN layer can improve the denoising performance. To

**Fig. 9** Denoising results of the image from Set12 with a noise level of 100



demonstrate this conclusion, we use DRCNN–withoutBN (ours) and DRCNN–withBN for testing. The only difference between the two is whether they have BN structure. As described in Sect. 3.2, we train for 60 epochs. We still selected the BSD400 to train at the noise level of 25 and then use the BSD68 dataset to test the PSNR values obtained by different methods, respectively. The loss curve is shown in Fig. 6. As seen in Fig. 6, the average PSNR of DRCNN–withoutBN is higher than that of DRCNN–withBN. Therefore, BN is not important in denoising, and sometimes removing the BN layer can improve the denoising performance.

#### 4.4 Comparisons with state-of-the-art methods

The loss curves of DnCNN and DRCNN are shown in Fig. 7. From Fig. 7, we can see that our DRCNN model converges faster than the DnCNN model, and the final PSNR is higher. The average PSNR results of different methods on the BSD68 dataset are shown in Table 1. As one can see, our DRCNN model can achieve the best PSNR results over the competing methods at almost every noise level. Compared to the benchmark BM3D, the PSNR value of our model is higher by approximately about 0.7 dB over that of the BM3D model when the noise level is 50.

Table 2 lists the PSNR results of different methods on the 12 test images shown in Fig. 3. The best PSNR result for each image with each noise level is highlighted

in bold. It can be seen that the denoising results of our model are better than that of the comparison method in almost every image. At low noise levels, our denoising effect is similar to that of DnCNN-S. However, at high noise levels, the denoising performance of our model is notably better than those of other methods. Figures 8 and 9 illustrate the visual results of different methods. It can be seen that for BM3D and DnCNN-S lost more texture details. When magnified, some details also became blurred.

## 5 Conclusion

In this work, we presented DRCNN as a novel image denoising method using very deep networks. It is difficult to train a very deep network because of the slow convergence rate. Gradient vanishing and explosion are the two largest difficulties in the process of neural network deepening. To address this limitation, we use the residual learning and skip connection operations to optimize a very deep network for DRCNN. By applying the methods introduced in this paper, the neural network is deepened and the network denoising ability is not inhibited by network degradation. Based on the experimental results, we compared the existing denoising algorithms and demonstrated that the method we proposed is not only improved on PSNR but also exhibited a very good in visual performance.

**Acknowledgements** This work was partially supported by the National Natural Science Foundation of China (Nos. 61702129, 61772149, 61762028, and U1701267), China Postdoctoral Science Foundation (No. 2018M633047), and Guangxi Science and Technology Project (Nos. AD18216004, AD18281079, and 2018GNSFAA138132).

## References

- Abadi, M.: Tensorflow: learning functions at scale. *ACM SIGPLAN Not.* **51**(9), 1–1 (2016)
- Buades, A., Coll, B., Morel, J.M.: A non-local algorithm for image denoising. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 60–65 (2005)
- Chen, Y., Pock, T.: Trainable nonlinear reaction diffusion: a flexible framework for fast and effective image restoration. *IEEE Trans. Pattern Anal. Mach. Intell.* **39**(6), 1256–1272 (2017)
- Dabov, K., Foi, A., Katkovnik, V., Egiazarian, K.: Image denoising by sparse 3-D transform-domain collaborative filtering. *IEEE Trans. Image Process.* **16**(8), 2080–2095 (2007)
- Dong, C., Chen, C.L., He, K., Tang, X.: *Learning a Deep Convolutional Network for Image Super-Resolution*. Springer, Berlin (2014)
- Elad, M., Aharon, M.: Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Trans. Image Process.* **15**(12), 3736–3745 (2006)
- Gu, S., Zhang, L., Zuo, W., Feng, X.: Weighted nuclear norm minimization with application to image denoising. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2862–2869 (2014)
- Harmeling, S.: Image denoising: Can plain neural networks compete with BM3D? In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2392–2399 (2012)
- He, K., Zhang, X., Ren, S., Sun, J.: *Identity Mappings in Deep Residual Networks*. Springer, Berlin (2016)
- Ioffe, S., Szegedy, C.: Batch normalization: accelerating deep network training by reducing internal covariate shift. In: *International Conference on Machine Learning*, pp. 448–456 (2015)
- Kim, J., Lee, J.K., Lee, K.M.: Accurate image super-resolution using very deep convolutional networks. In: *CVPR*, pp. 1646–1654 (2015)
- Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. In: *The 3rd International Conference on Learning Representations (ICLR 2015)*
- Lan, R., He, J., Wang, S., Gu, T., Luo, X.: Integrated chaotic systems for image encryption. *Signal Process.* **147**, 133–145 (2018)
- Lan, R., Zhou, Y., Liu, Z., Luo, X.: Prior knowledge-based probabilistic collaborative representation for visual recognition. In: *IEEE Transactions on Cybernetics*, pp. 1–11 (2018). <https://doi.org/10.1109/TCYB.2018.2880290>
- Lan, R., He, J., Wang, S., Liu, Y., Luo, X.: A parameter-selection-based chaotic system. *IEEE Trans. Circuits Syst. II: Express Briefs* **66**(3), 492–496 (2019)
- Lan, R., Lu, H., Zhou, Y., Liu, Z., Luo, X.: An LBP encoding scheme jointly using quaternionic representation and angular information. In: *Neural Computing and Applications*, pp. 1–7 (2019). <https://doi.org/10.1007/s00521-018-03968-y>
- Lim, B., Son, S., Kim, H., Nah, S., Lee, K.M.: Enhanced deep residual networks for single image super-resolution. In: *Computer Vision and Pattern Recognition Workshops*, pp. 1132–1140 (2017)
- Liu, Y.N., Wang, Y.P., Wang, X.F., Xia, Z., Xu, J.F.: Privacy-preserving raw data collection without a trusted authority for IoT. *Comput. Netw.* **148**, 340–348 (2019)
- Malshika Welhenge, A., Taparugssanagorn, A.: Human activity classification using long short-term memory network. *Signal Image Video Process.* **13**(4), 651–656 (2019)
- Mao, X., Shen, C., Yang, Y.B.: Image restoration using very deep convolutional encoder–decoder networks with symmetric skip connections. In: *Advances in Neural Information Processing Systems*, pp. 2802–2810 (2016)
- Martin, D., Fowlkes, C., Tal, D., Malik, J.: A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: *Proceedings Eighth IEEE International Conference on Computer Vision, 2001. ICCV 2001*, vol. 2, pp. 416–423 (2002)
- Rajwade, A., Rangarajan, A., Banerjee, A.: Image denoising using the higher order singular value decomposition. *IEEE Trans. Pattern Anal. Mach. Intell.* **35**(4), 849–862 (2013)
- Schmidt, U., Roth, S.: Shrinkage fields for effective image restoration. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2774–2781 (2014)
- Srivastava, R.K., Greff, K., Schmidhuber, J.: Highway networks. In: *International Conference on Machine Learning Deep Learning workshop* (2015)
- Tabatabaei, S.M., Chalechale, A.: Local binary patterns for noise-tolerant sEMG classification. *Signal Image Video Process.* **13**(3), 491–498 (2019)
- Wang, J., Fan, Y., Li, Z., Lei, T.: Texture classification using multi-resolution global and local Gabor features in pyramid space. *Signal Image Video Process.* **13**(1), 163–170 (2019)
- Xie, J., Xu, L., Chen, E.: Image denoising and inpainting with deep neural networks. In: *International Conference on Neural Information Processing Systems*, pp. 341–349 (2012)
- Zhang, K., Zuo, W., Chen, Y., Meng, D., Zhang, L.: Beyond a Gaussian denoiser: residual learning of deep CNN for image denoising. *IEEE Trans. Image Process.* **26**(7), 3142–3155 (2017)
- Zhang, M., Gunturk, B.K.: Multiresolution bilateral filtering for image denoising. *IEEE Trans. Image Process.* **17**(12), 2324–2333 (2008)
- Zhao, S., Yao, H., Gao, Y., Ji, R., Ding, G.: Continuous probability distribution prediction of image emotions via multitask shared sparse regression. *IEEE Trans. Multimed.* **19**(3), 632–645 (2016)
- Zhao, S., Ding, G., Gao, Y., Han, J.: Approximating discrete probability distribution of image emotions by multi-modal features fusion. In: *IJCAI'17*, vol. 1000(1), pp. 4669–4675 (2017)
- Zhao, S., Ding, G., Gao, Y., Zhao, X., Tang, Y., Han, J., Yao, H., Huang, Q.: Discrete probability distribution prediction of image emotions with shared sparse learning. In: *IEEE Transactions on Affective Computing*, pp. 1–1 (2018). <https://doi.org/10.1109/TAFFC.2018.2818685>
- Zhao, S., Gao, Y., Ding, G., Chua, T.: Real-time multimedia social event detection in microblog. *IEEE Trans. Cybern.* **48**(11), 3218–3231 (2018)
- Zhao, S., Yao, H., Gao, Y., Ding, G., Chua, T.: Predicting personalized image emotion perceptions in social networks. *IEEE Trans. Affect. Comput.* **9**(4), 526–540 (2018)

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.