Image Denoising Using Deep CNN with Batch Renormalization

Chunwei Tian[a,b], Yong Xu[a,b,*], Wangmeng Zuo[c]

[a]*Bio-Computing Research Center, Harbin Institute of Technology, Shenzhen, Shenzhen, 518055, Guangdong, China*
[b]*Shenzhen Medical Biometrics Perception and Analysis Engineering Laboratory, Harbin Institute of Technology, Shenzhen, Shenzhen, 518055, Guangdong, China*
[c]*School of Computer Science and Technology, Harbin Institute of Technology, Harbin, 150001, Heilongjiang, China*

Abstract

Deep convolutional neural networks (CNNs) have attracted great attention in the field of image denoising. However, there are two drawbacks: (1) it is very difficult to train a deeper CNN for denoising tasks, and (2) most of deeper CNNs suffer from performance saturation. In this paper, we report the design of a novel network called a batch-renormalization denoising network (BRDNet). Specifically, we combine two networks to increase the width of the network, and thus obtain more features. Because batch renormalization is fused into BRDNet, we can address the internal covariate shift and small mini-batch problems. Residual learning is also adopted in a holistic way to facilitate the network training. Dilated convolutions are exploited to extract more information for denoising tasks. Extensive experimental results show that BRDNet outperforms state-of-the-art image-denoising methods. The code of BRDNet is accessible at http://www.yongxu.org/lunwen.html.

Keywords: Image denoising, CNN, Residual learning, Batch renormalization, Dilated convolution

1. Introduction

Image denoising aims to recover a clean image from a noisy image, which is a classical-inverse problem in computer vision. Since image-denoising techniques can recover original images well, and restore the details, they are widely applied in many fields, such as remote-sensing image [15] and medical image [37]. For a noisy image y, the problem of image denoising can be represented by $y = x + \upsilon$, where x is the original image (also referred to as the clean image) and υ represents additive Gaussian noise (AWGN) with standard deviation σ. In the view of Bayesian rule, image-prior-based methods are good choices for image denoising. For example, block-matching and three-dimensional (3D) filtering (BM3D) exploited collaborative alteration to enhance the sparsity for image denoising [11]. The simultaneous use of sparse representation based on dictionary learning and non-local means based on self-similarities can remove the noise from noisy images

*Corresponding author
Email address: yongxu@ymail.com (Yong Xu)

[42]. Non-locally centralized sparse representation (NCSR) centralized the sparse coding to suppress the sparse-coding noise [14]. Weighted nuclear norm minimization (WNNM) [19], markov random field (MRF) [43, 1, 2], gradient methods [3, 69] and total-variation (TV) methods [49, 5] are also very popular image-denoising methods.

Although the above methods have shown excellent performance in image denoising, most of these methods are faced with two major challenges [61]: (1) manual tuning of the penalty parameters, and (2) complex optimized algorithms. Owing to the adaptive strong learning ability, deep-learning techniques, especially convolutional neural networks (CNNs) [38], have become the most favored methods of addressing these issues [12, 53, 28]. Specifically, the recently proposed deep CNNs have been widely applied in image restoration [33]. For example, low-resolution images were directly mapped into high-resolution images by a CNN [12]. Residual and iterative ideas were embedded into the CNN to improve the performance for image super-resolution [53]. The combination of an optimization method and a CNN was a good tool for image super-resolution [55, 47]. Increasing the diversity of the network was also very effective for image restoration [67]. CNNs with prior knowledge can better deal with real-noisy images [62]. Optimizing the network architecture was popular for image restoration [26]. Although the above deep-network methods can improve the denoising performance, most of these methods suffer from the problems of performance saturation (i.e., vanishing or exploding gradients), [23] and the difficulty of training deep networks [25, 45]. Additionally, their partial networks with batch normalization (BN) [25] can generate errors in small mini-batches. Some of the above methods have a very high computational cost for image restoration.

In this paper, we propose a novel network for image denoising named a batch-renormalization denoising network (BRDNet). First, BRDNet combines two networks to increase the width of BRDNet and obtain more features for image denoising. Next, BRDNet uses batch renormalization (BRN) [24] to address the small mini-batch problem, and applies residual learning (RL) with skip connection [23] to obtain clean images. Finally, to reduce the computational cost, dilation convolutions (Dilated Conv) [60] are used to capture more features. Also, extensive experiments demonstrate that the proposed BRDNet outperforms state-of-the-art methods, e.g., a denoising convolutional neural network (DnCNN) [61], a fast and flexible denoising network (FFDNet)[63], and an image-restoration CNN (IRCNN) [62]. Additionally, the extension of BRDNet also has good effects on both synthetic and real noisy images.

The proposed method comprising BRDNet has the following contributions:

(1) A novel deep CNN is proposed for image denoising in this paper, which can directly obtain a clean image from a noisy image. Unlike the existing CNN denoising methods, the proposed network increases the width rather than depth to enhance the learning ability of the denoising networks.

(2)Batch renormalization is used for image denoising, which can address small mini-batch problems. Moreover, BRN can also accelerate the convergence of training the network, and does not have any requirement for a specific hardware platform. As a consequence, the combination of BRN and CNN for image denoising is a good choice for low-configuration hardware devices, such as GTX 960 and GTX 970.

(3) BRDNet uses dilated convolutions to enlarge the receptive field, which enables the network to extract more context information and reduce the computational cost. Moreover, it can also

prevent vanishing or exploding gradients. In addition, residual learning can further promote the image-denoising performance.

(4) Experimental results prove that BRDNet is robust to both synthetic and real noisy images.

The rest of this paper is organized as follows. In Section 2, we provide related techniques of the proposed method. In Section 3, we offer the proposed method. In Section 4, we present experimental results of the proposed method. In Section 5, we present conclusions.

2. Related Work

2.1. Deep CNNs for Image Denoising

Deep CNNs are very popular for image denoising. For example, a 17-layer DnCNN [61] has been proposed as a CNN-based method of predicting noise. This baseline improves the denoising performance by stacking multiple convolutional layers, a skip connection, and batch normalization [25]. The combination of a CNN and a prior was effective for blind denoising [63]. A discriminative learning method with an optimization method was used to deal with real noisy images [62]. Improving a spatial activation function was exploited to reduce the computational cost in image denoising [30]. The recently proposed generative adversarial network (GAN) is very popular for image denoising [6]. A CNN with a prior was also very effective for noise of certain type [10, 66, 13]. The CNN is very robust for other applications, such as medical images [34]. The aforementioned deep-network-based-CNN methods [51] have achieved better performance than BM3D. Motivated by that, we use a deep CNN for image denoisining.

2.2. Batch Renormalization

To the best of our knowledge, the manner of end-to-end connection is an important factor for the success of a CNN. A CNN generally consists of an activation function [32], pooling operation [21], initial parameter setting [22], and gradient-based optimization methods [29] and convolutional kernels. Although these plug-in components can improve the performance in image applications compared with traditional methods, such as multi-layer perception (MLP), the distribution of training data is clearly changed by a convolutional operation. The amount of training data is larger, the predicted results will be more inaccurate. To solve this problem, BN was developed [25]. The BN method used a normalization operation, and scale and shift operations to resolve the internal covariate shift problem. The proposed BN method can not only prevent exploding or vanishing gradients, and accelerate the convergence of training the network, but can also improve performance. However, BN was invalid for small mini-batches, which seriously limited the applications of BN, such as for image detection and video tracking [24]. Inspired by the fact that BRN can effectively address the dilemma of BN, because BRN used individual samples instead of the entire mini-batch to approximate the distribution of training data. BRN can effectively solve the small mini-batch size and non-independent identically distributed mini-batch (non-i.i.d. mini-batch) problems. The non-i.i.d. mini-batch problem is that samples in same mini batch are non-independent identically and distributed, which can make machine learning or deep learning methods have poor performance. For more information about BRN, please refer to [24]. As a consequence, a CNN with BRN is a good choice for image denoising.

2.3. Residual Learning and Dilated Convolution

Residual learning was proposed by He et al. [23] to achieve the tradeoff between increasing depth and addressing network performance degradation. It fused extracted features and the input of several stack layers as the input of the current layer, which can solve the vanishing or exploding gradients problem. To this end, many variants of residual neural network (ResNet) were proposed for low-level-vision tasks [27, 31]. For example, very deep super-resolution (VDSR)[27] exploited global residual learning (GRL) for image restoration. GRL and gradient clipping operations can make VDSR accelerate the convergence. A deeply recursive convolutional network (DRCN) [28] combined a recursive mechanism and GRL to address the overfitting problem for image restoration. A deep recursive residual network (DRRN) integrated GRL, local RL, and recursive learning to improve image-restoration performance [53].

Extracting more suitable features is important for image recognition [57, 17, 56, 20, 40], image segmentation [39], object detection [9], image denoising [16], and image super-resolution [61, 63]. Traditional CNNs used pooling operations to reduce the dimensionality from original images. However, the results suffered from information loss. Enlarging the receptive field was one of the effective approaches used to address this issue. Increasing depth and filter size are very popular ways to enlarge the receptive field. However, increasing depth may lead to network performance degradation. Enlarging filter size will increase the number of parameters and computational cost. The recently proposed dilated convolution used 3×3 filters and large depth to address the above receptive field problems [60]. The receptive field size of a dilated convolution can use the dilation factor f and number of dilated convolution layers, n, as $(4n + 1)\times(4n + 1)$. For example, the designed network has five dilated convolution layers with 3×3 filters when $f = 2$ and $n = 10$, and the receptive field size of the network is 41×41, which is equivalent to a common CNN with 20 layers. Thus, the dilated convolution technique has great prospects. Existing studies show that dilated convolutions also have good effects on image denoising [54]. However, some scholars ignore the combination of BRN, RL, and dilated convolutions in low-level-vision tasks.

3. Proposed CNN-based Denoising Method

In this section, we propose BRDNet based on a CNN for image denoising. Generally, training of a deep CNN model includes two stages for specific task: network design and model learning on training samples. For the design of network architecture, we concatenated two CNNs to design a novel network. For the model learning, we integrated BRN, RL, and dilated convolutions into the designed network for training a denoising model.

3.1. Network Architecture

In general, different network architectures can extract different features [67]. These extracted features are complementary in image denoising. Increasing the width of the network to enhance performance is a good choice for image denoising. Thus, we propose a novel network based on two networks as shown in Figure. 1. This network is called BRDNet and consists of two different networks: an upper and lower networks. The upper network only consists of RL and BRN. The lower network includes BRN, RL, and dilated convolutions. It is known that the receptive field is larger, the designed network will have higher computational cost. Therefore, we choose one

network (the lower network) to use dilated convolutions. Take into account the balance of performance and efficiency, the 2-8 and 10-15 layers of the lower network use dilated convolutions to capture more context information. The first and sixteenth layers use BRN to normalize data, which makes the outputs of the two sub-networks keep the same distribution. Also, BRN is very useful for small-batch-size tasks, which is very beneficial for low-configuration hardware platforms, such as GTX960 and GTX970. Next, an RL technique is fused into two-channel networks to improve performance. The effectiveness of each technique was tested (see Section 4.3), and the performance of this network is shown in Section 4. The detailed information of BRDNet is introduced in the latter section.

The depth of the upper network (also called the first network) is 17. It consists of two different types of layers: Conv+BRN+ReLU and Conv. Conv, BRN, and ReLU are referred to as convolution, batch renormalization, and rectified linear units, respectively. Conv+BRN+ReLU means that convolution, batch renormalization, and rectified linear units are implemented in sequence. Layers $1-16$ are Conv+BRN+ReLU and the 17th layer is Conv. Except for the first and last layers, the size of each layer is $64\times3\times3\times64$. The sizes of the first and last layers are $c\times3\times3\times64$ and $64\times3\times3\times c$, respectively, where c denotes the number of channels. Here $c=1$ and $c=3$ represent the numbers of the pixel channels of gray and color images, respectively.

The receptive field can capture more information from the context with the help of the dilation factor. For example, the receptive filed size is $(2l+1)\times(2l+1)$ when the dilation factor in the first network is 1, where l is the number of layers. In addition, $\oplus$ denotes the implementation of the RL idea in this paper, that is, the subtraction operation in BRDNet in practice. The 'Concat' operation is used to concatenate two sub-networks in BRDNet by their channels. For example, if the output sizes of the above two sub-networks are $64\times3\times3\times c$, the output size of their combinations by concatenation operation is $64\times3\times3\times2c$.

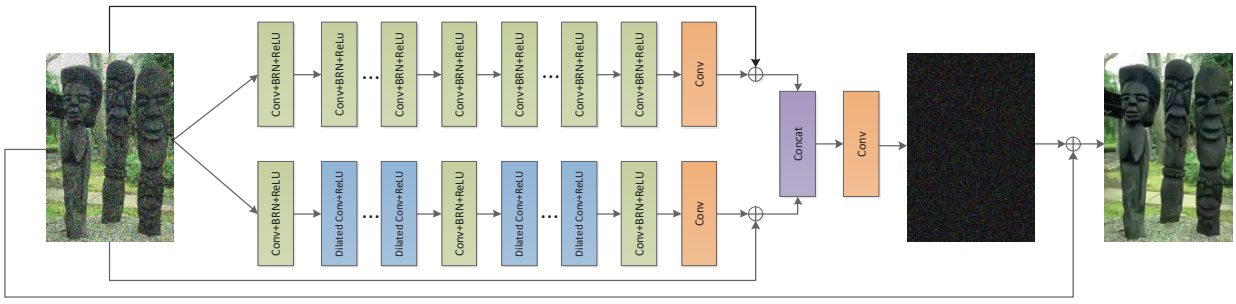

Figure 1: Architecture of proposed BRDNet network.

We refer to the lower network with a depth of 17 as the second network. The first, ninth, and sixteenth layers in the second network are Conv+BRN+ReLU. Layers 2-8 and 10-15 are dilated convolutions. The final layer is Conv. The filter size of each layer is the same as that for the first network. However, layers 2-8 and 10-15 receive more information from a broader field, because the dilation factor is 2. Specifically, for layers 2-8 and 10-15, their receptive field size can be computed by $(4n+1))(4n+1)$, respectively. As a result, the receptive fields of all 17 layers are 3, (7, 11, 15, 19, 23, 27, 31), 33, (35, 39, 43, 47, 51, 55, 59), and 61, which achieves a performance comparable to that with 30 layers under the same filter-size settings. In other words, dilated

convolutions can reduce computational cost for image denoising. Also, the dilated convolutions and two sub-networks can reduce depth. That is, BRDNet has only 18 layers, which is relatively shallow, and does not result in vanishing or exploding gradients.

The merits of BRDNet as described above are four-fold: (1) it increases width by using two sub-networks rather than depth to improve the denoising performance, (2) it uses BRN to address small-batch and internal covariate-shift problems, (3) it applies RL to prevent vanishing or exploding gradients, and (4) it utilizes dilated convolutions to save computational cost. Further, experimental results show that BRDNet is more effective than other state-of-the-art denoising methods, such as DNCNN, FFDNet, and IRCNN, which proves the effectiveness of the proposed network. In Section 4.3, we present results of testing the contributions of RL, BRN, and dilated convolutions for BRDNet, respectively.

3.2. Loss Function

In our work, motivated by GooLeNet and DnCNN, we choose the mean-square error (MSE) to obtain the optimal parameters of the network. Let x be a clean image and y be a noisy image. When training dataset $\{x_j, y_j\}_{j=1}^{N}$ is given, BRDNet uses RL to obtain a model and predict a residual image $f(y)$, where $f(y)$ denotes the noise mapping. Then, we transform a noisy image into a clean image via $x = y - f(y)$. In other words, we have $f(y) = y - x$, and the training samples approximately obtain this equality. Specifically, the optimal parameters can be obtained by minimizing the following loss function with Adam [29]:

$$l(\theta) = \frac{1}{2N} \sum_{j=1}^{N} \left\| f(y_i, \theta) - (y_i - x_i) \right\|^2, \tag{1}$$

where N is the number of noisy-image patches and θ denotes the parameters of the proposed model. In general, different regions of an image have different structural information. To this end, the noisy-image patches are easier to use to learn position-specific features than the entire noisy image [68]. Moreover, using noisy-image patches can significantly save memory and reduce computational cost compared with the entire noisy image [68]. As seen in Eq. (1), it can be known that the objective function is relevant to the noisy image y , clean image x, and residual image $f(y)$.

3.3. Integration of BRN, Dilated Convolution, and RL

One advantage of BRDNet is the combination of two different and complementary networks for image denoising. As shown in Figure. 1, the first network mainly includes BRN and residual learning. The second network integrates BRN, dilated convolution, and RL. From Figure. 1, we can see that BRDNet can obtain a latent clean image by predicting additive white Gaussian noise with standard deviation σ (σ = 75). That is, at first, BRDNet can be used to predict the noise v. Then, it uses the obtained noise v to produce clean image x. The designed network adheres to the following rules.

First, it is known that deeper networks can result in vanishing or exploding gradients. Thus, we design a novel denoising network called BRDNet, that uses two different sub-networks to reduce the depth of the network and obtain more features. The depth is reduced and it does not generate vanishing or exploding gradients.

Second, the distribution of training data is changed through a convolutional kernel. BN is considered to be a good choice for addressing the problem. However, it is not very effective when the batch size is small, which limits its applications. In real applications, many hardware devices limited memory, and can run programs with high computational complex. Thus, we use BRN instead of BN to normalize the data, and improve the convergence speed of the denoising network. The principle of BRN is as follows [24].

Algorithm 1 The implementations of Batch Renormalization

Input: Values of x over a training mini-batch $B = \{x_{1...m}\}$; parametersγ, β; current moving mean μ and standard deviation σ; moving average update rate α; maximum allowed corrections $\tau_{\max}, d_{\max}$.
Output: $y_i = BatchRenorm(x_i)$;updated μ ,σ.
$\mu_B \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i$.
$\sigma_B \leftarrow \sqrt{\varepsilon + \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_B)^2}$.
$r \leftarrow stop\_grdient(clip_{[1/r_{\max}, r_{\max}]}\left(\frac{\sigma_B}{\sigma}\right))$.
$d \leftarrow stop\_grdient(clip_{[-d_{\max}, d_{\max}]}\left(\frac{\mu_B - \mu}{\sigma}\right))$.
$\hat{x}_i \leftarrow \frac{x_i - \mu_B}{\sigma_B} \times \gamma + d$.
$y_i \leftarrow \gamma\hat{x}_i + \beta$.
$\mu := \mu + \alpha(\mu_B - \mu)$ //Update moving averages.
$\sigma := \sigma + \alpha(\sigma_B - \sigma)$.
Inference: $y \leftarrow \gamma \times \frac{x-\mu}{\sigma} + \beta$.

Third, it is known that the deep network can extract more accurate features. However, a deep network will lose some context information. As a consequence, we use dilated convolutions in BRDNet to enlarge the receptive field and capture more context information. Specifically, dilated convolutions can use fewer layers to play the same role as more layers. It is known from previous research that increasing the width can extract more features [52], and BRDNet has the network architecture, that increases the width of the network rather than its depth to extract more robust features (referred to as a two-channel network). Therefore, the combination of two-channel networks and dilated convolution is very effective in improving image-denoising performance. Moreover, reduction of the network depth can also prevent vanishing or exploding gradients. In this way, BRDNet can reduce the computational cost. However, the lower network has only dilated convolutions, which can make the two sub-networks generate complementary features and improve the network's generalization ability. It seems that, although the two sub-networks do not have deep depths, integrating them can perform very well in comparison with very deep single networks, e.g., FFDNet and IRCNN. In our opinion, dilated convolutions have similar functions to deep networks in increasing the size of the receptive field. Finally, we use a RL method in BRDNet to improve performance again.

4. Experimental Results

In this section, we mainly introduce the experimental results from the following aspects: datasets,experimental setting, component analysis, BRDNet for gray synthetic noisy images, and for color synthetic noisy images, real-noisy-image denoising, and running time. First, we present the parameters of BRDNet. Next, we prove the effectiveness of main techniques detailed in this paper. Then, the performance of BRDNet on the BSD68 and Set12 [48] public datasets for gray synthetic noisy image denoising is reported. Several state-of-the-art methods for gray synthetic noisy image denoising, such as BM3D [11], WNNM [19], MLP [4], trainable nonlinear reaction diffusion (TNRD)[7], expected patch log likelihood (EPLL) [68], cascade of shrinkage fields (CS-F) [50], DnCNN [61], IRCNN [62], and FFDNet [63] are chosen for comparison with the proposed method.

To test the performance of the proposed method, we use the recognized peak signal-to-noise ration (PSNR) [64] and visual effect to verify the denoising effect. If the PSNR value of the denoising method on the test dataset is larger, the denoising method has exhibited better performance. Furthermore, to clarify the visual effect on the denoised images, we zoom in on one area from one obtained potential clean image to demonstrate it. If the magnified area is clearer, we deem that the tested method is more effective. In particular, $PSNR = 10 * \log_{10} (MAX)^2/MSE$, where MAX represents the maximum pixel value of each image. It is noted that MAX is 1.0 in Figures. 3 to 8. The MSE is the error between a real clean image and a predicted clean image, and computed by $MSE = \frac{1}{n} \sum_{j=1}^{n} \sum_{i=1}^{n} (x_j^i - y_j^i)^2$, where x_j^i and y_j^i denote the pixels of point (i, j) from a given clean and recovered clean images, respectively. To give an example, we assume that a given real clean image from cc dataset is xx in Section 4.5, the corresponding recovered clean image is yy from denoising model of BRDNet. MAX is the maximum pixel between xx and yy. Thus, their PSNR values can be computed by $PSNR = 10 * \log_{10} (MAX)^2/MSE$. Next, we show the performance of BRDNet on color synthetic noisy images denoising. We also illustrate the results of our method and several popular methods on real noisy images. Specifically, we refer to Refs. [61, 63, 62, 59], and obtain convincing denoising results of other comparative methods for synthetic and real noisy images. Finally, we measure the computational cost, the results of which show that our method is also very competitive.

4.1. Datasets

4.1.1. Training Datasets

For Gaussian image denoising, we used 3,859 images from the Waterloo Exploration Database to train the model. The training images were cropped into $20 \times \ 58,319$ patches in *.bmp format. The patch size was set as 50×50. The patch size was set as 50×50 for the following reasons: our designed network consists of two sub-networks, the depth of each being 17, the depth of BRDNet is 18, and the receptive field of one network is $2 \times 17 + 1 = 35$ and that of the other is 61. If patch size is obviously greater than receptive field size, the designed network will consume more computational cost. Therefore, we calculated the average value of the receptive field size of the two sub-networks as the receptive field of BRDNet, and its size is $[(35+61)+2]/2 = 49$. Here the patch size is set as 50×50 (50>49) in this work. It is noted that patch size of 50 is less receptive

field size of the second network, which can not fully map the second sub-network. However, the patch size is greater receptive field size of the first network, which can provide complementary information for the second network. Thus, to make a tradeoff between efficiency and performance, patch size of 50 is proper.

For real noisy image denoising, we used 100 images from Ref. [58] to train the model. These real noisy images were captured by different cameras, i.e., Canon 5D Mark II, Canon 80D, Canon 600D, Nikon D800, and Sony A7 II, the sensor sizes of the cameras and scenes are all different. The training images were cropped into 423,200 patches in *.bmp format. Each patch size is the same as the gray and color image patch sizes.

4.1.2. Test Datasets

For gray-noisy image denoising, we use the Gaussian noise to train the denoising model. According to the DnCNN and FFDNet methods, we choose Berkeley segmentation dataset 68 (BSD68) and Set12 [48] as test datasets. The BSD68 dataset includes 68 natural images with a size of 481×321 or 321×481. The Set12 is composed of 12 gray images.

For color-noisy image denoising, we use CBSD68, Kodak24 [18], and McMaster to test the BRDNet for image denoising. The CBSD68 includes 68 color images and is the same background as BSD68. The Kodak24 is composed of 24 natural images and their sizes are 500×500. The McMaster consists of 18 color images and their sizes are 500×500.

It is known that real noisy images are usually captured by cameras of different types with different ISO values [46]. Motivated by the fact, we choose cc [44] as the test dataset for real noisy image, as shown in Figure. 2. The cc dataset has 15 noisy images, that are captured by three different cameras, i.e., Nikon D800, Nikon D600, and Canon 5D Mark III with different ISO values (i.e., 1,600, 3,200, and 6,400).

4.2. Experimental Setting

We set the BRDNet depth as 18 for gray synthetic, color synthetic and real noisy images denoising. The objective function is used to predict the residual image as shown in Eq. (1). We utilize learning rate of 1×10^{-3}, beta_1 of 0.9, beta_2 of 0.999, epsilon of 1×10^{-8}, and the method of Ref. [22] to initialize the weights. The mini-batch size is 20. The number of epochs is 50 for training the BRDNet models. The learning rates of the 50 epochs vary from 1×10^{-3} to 1×10^{-4}.

We apply the Keras packet [8] to train the proposed BRDNet denoising model. All experiments are implemented in the Ubuntu 14.04 and Python 2.7 environments, and run on a PC with an Intel Core i7-6700 CPU, 16GB RAM, and a Nvidia GeForce GTX 1080Ti graphical processing unit (GPU). The Nvidia CUDA of 8.0 and cuDNN of 7.5 are chosen to accelerate the computational ability of the GPU. It takes approximately 144h to train the proposed model for color synthetic noisy images.

4.3. Component Analysis

To test the effectiveness of each technique for BRDNet on image denoising, we design six visual images. The effectiveness of the RL for image denoising is shown in Figures. 3 and 4. It is obvious from the figures that, the average PSNR of BRDNet without a RL operation is lower than that of BRDNet with a RL operation. The RL techniques in two sub-networks are used to

Figure 2: Twelve images from the cc dataset.

obtain clean images, respectively. Then, obtained clean images rather than noise mapping are fused to obtain cleaner image, which can prevent enhancement of noise. It is noted that although fused clean image can improve the quality of image, it may result in part pixels are excessively prominent in comparison with the given clean image. Thus, the convolution in the last layer is used to eliminate the naïve effect above and coverts the obtained features into corresponding noise

Table 1: Gaussian denoising results of eight specific models for color synthetic noisy images. All four models were trained with $\sigma = 50$ and evaluated on the CBSD68 dataset.

Methods	Epochs=50
Upper network	27.33
Lower network	28.06
Lower network without dilated convolutions	27.74
Two sub-networks with dilated convolutions	28.16
Two sub-networks without RL	28.11
DnCNN	28.01
Concatenation of two DnCNNs	28.01
BRDNet	28.16

mapping. Finally, the RL in the end of BRDNet is used to obtain the final clean image via noisy image and predicted noise mapping. The performance of using three RLs and only one RL in BRDNet for image denoising can be shown via 'Two sub-networks without RL' and 'BRDNet' in Table 1. In addition, some other plug-in units, such as BRN and dilated convolutions, are also very beneficial to improve the denoising performance.

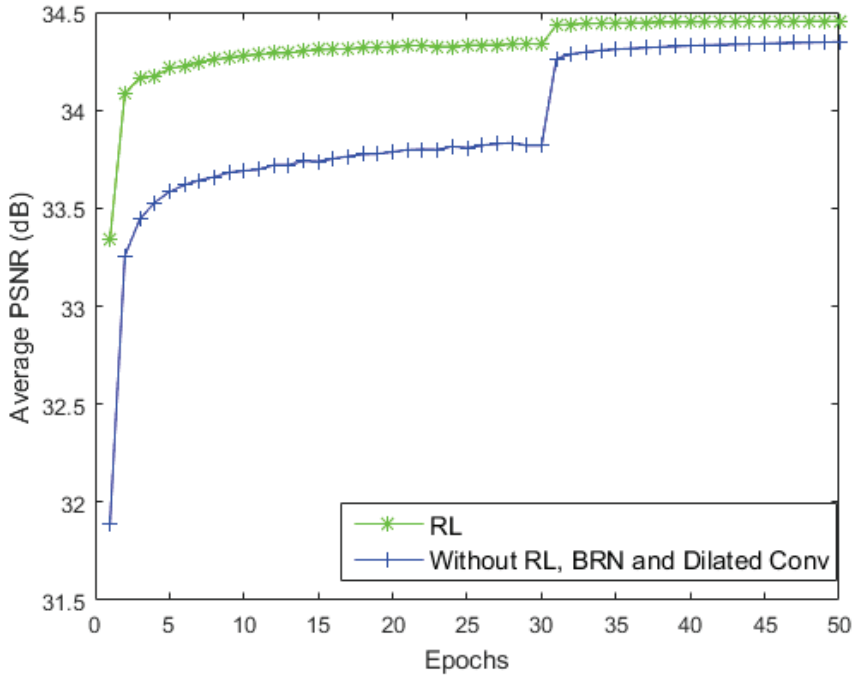

Figure 3: Gaussian denoising results for two specific models, one consisting of RL, and the other of only convolutional layers. Both were trained with $\sigma = 15$. Results for 68 images from the CBSD68 dataset were evaluated.

We use the BRN technique to improve the image-denoising performance. BRN uses individual

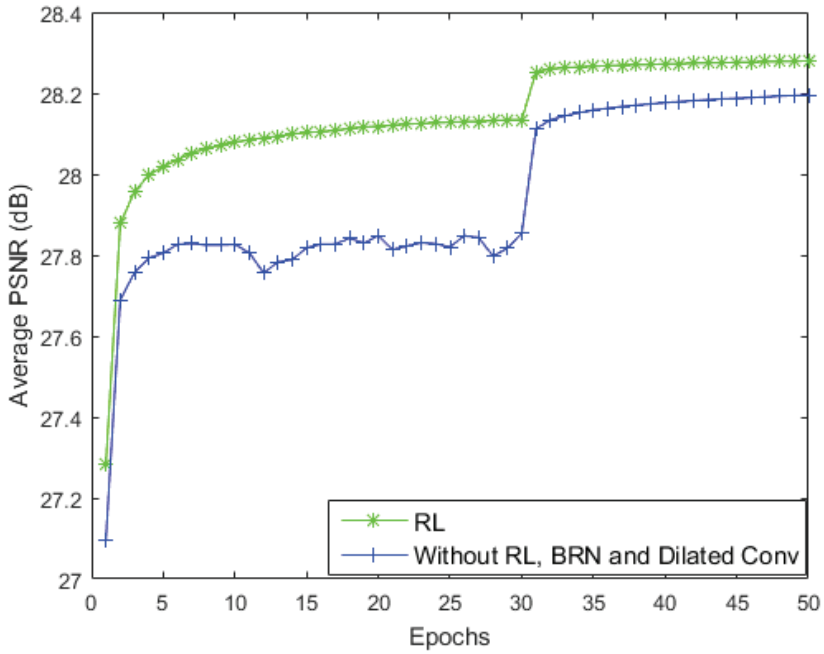

Figure 4: Gaussian denoising results for two specific models, one consisting of RL and the other of only convolutional layers. Both were trained with σ = 50. Results for 68 images from the CBSD68 dataset were evaluated.

samples instead of the entire mini-batch to approximate the distribution of training data. This approach is not only effective in tackling small-batch-size and non-i.i.d. mini-batch problems, but also inherits the aforementioned merits of BN. BRN was described in detail in Section 2.3. To verify that BRN is more effective in image denoising than BN with a small batch, we design several experiments with σ = 15. Batch sizes are set to 20, 32, and 64, respectively. The training dataset is the Waterloo Exploration Database [41]. The test set is McMaster [65]. From Table 2, when the batch size is smaller, BRN's performance is better. For example, BRDNet exhibits excellent performance for $Batchsize$ = 20. Thus, BRN is more suitable for image denoising under small mini-batch conditions. Small mini-batches are very suitable to low-configuration hardware, thus, BRDNet with BRN is very appropriate for real applications.

The integration of RL and BRN for image denoising is also interesting. Accordingly, we perform comparative experiments, the results of which are in Figures. 5 and 6, where the average PSNR of BRN with RL is higher than that of single RL in image denoising .

Dilated convolution allows the network to capture more information from the context by enlarging the receptive field. Thus, dilated convolutions are used in BRDNet for image denoising. The effectiveness of dilated convolutions is shown in Figures. 7 and 8. It can be seen from the figures that the performance of the combination of RL, BRN, and dilated convolution is better than that of the combination of RL and BRN for image denoising. Moreover, the PSNR of the lower network is higher than that of the lower network without dilated convolutions from Table 1, which

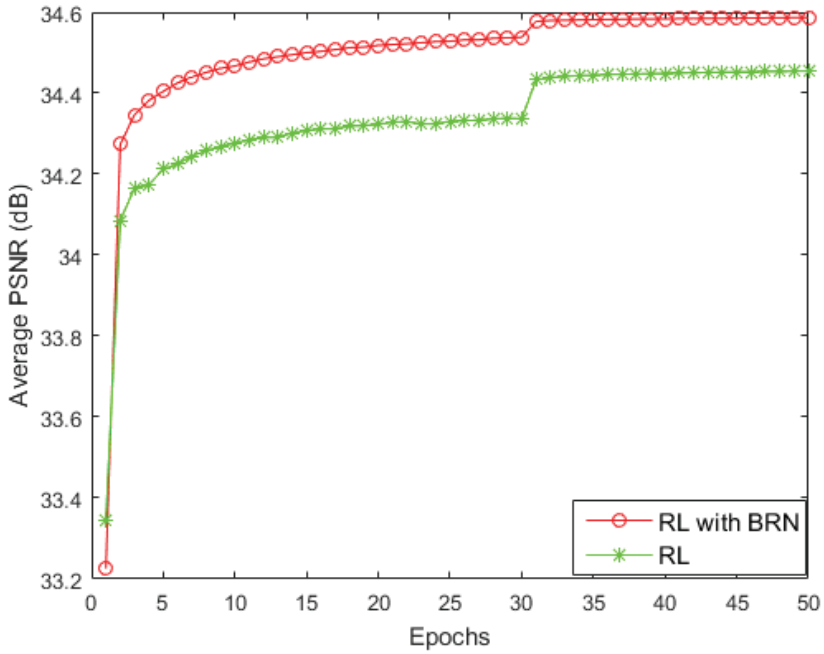

Figure 5: Gaussian denoising results for two specific models, one consisting of BRN and RL and the other of only RL. Both were trained with σ = 15. Results for 68 images from the CBSD68 dataset were are evaluated.

Table 2: Comparison of image-denoising results for BRN and BN in proposed network.

Datasets	Methods	Batchsize=20	Batchsize=32	Batchsize=64
McMaster	BRN	35.08	35.09	35.08
	BN	34.94	35.07	35.07

also proves the effectiveness of dilated convolutions in BRDNet for image denoising.

It is known that different network architectures can generate different features [67] that are complementary in image denoising. Thus, we design two different network architectures for BRNet (i.e., upper and lower networks) to improve denoising performance. The fusion of two networks is more useful than a single network in image denoising. For example, BRDNet has a better PSNR than these of the upper and lower networks when $\sigma = 50$, as shown Table 1. In addition, it is noteworthy that we use only dilated convolutions in the lower network for image denoising, which can not only produce a greater difference between two networks, but also have a higher efficiency than those of two networks with dilated convolutions, proved in Table 3. When two sub-networks (i.e., two lower networks) utilize dilated convolutions for 2-8 and 10-15 layers, their running time is higher than that of BRNet. However, their performance is the same as shown in Table 1. Thus, network architecture of BRDNet is proper. Further, BRDNet's performance is illustrated in Sections 4.4-4.6.

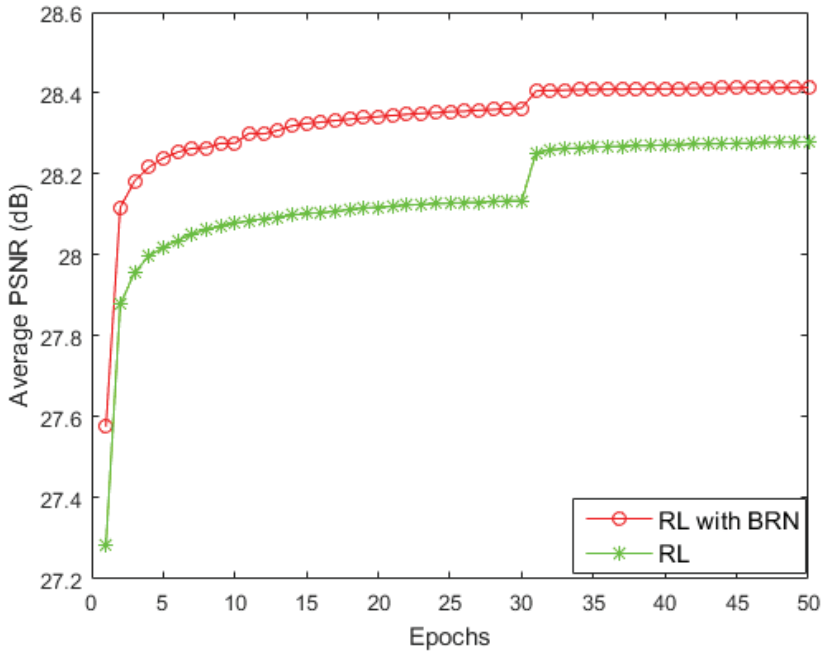

Figure 6: Gaussian denoising results for two specific models, one consisting of BRN and RL and the other of RL only. Both were trained with $\sigma = 50$. Results for 68 images from the CBSD68 dataset were evaluated.

Table 3: Running time of two different methods for denoising images of sizes 256×256, 512×512 and 1024×1024.

Methods	Device	256×256	512×512	1024×1024
Two sub-networks with dilated convolutions	GPU	0.081	0.238	0.935
BRDNet	GPU	0.062	0.207	0.788

4.4. BRDNet for Gray- and Color- Image denoising

For gray-noisy image denoising, BRDNet and several state-of-the-art methods (i.e., BM3D, WNNM, EPLL, MLP, CFS, TNRD, DnCNN, IRCNN and FFDNet) are used to conduct experiments on the BSD68. As shown in Table 4, the proposed BRDNet can obtain the highest PSNR, which is better than those of the benchmarks, BM3D and DnCNN, for gray-image denoising. The best and second-best PSNR results for different σ values are highlighted in red and blue, respectively, in Table 4. The average PSNR of BRDNet is 0.72 dB higher than that of BM3D for $\sigma = 25$, which indicates that BRDNet has better performance.

To easily observe the performance of BRDNet and other methods, we zoom in on one area from one potential clean image obtained using different methods, as illustrated in Figure. 9. To observe the performance of a single class image, we use Set12 to conduct the experiments. Table 5 shows the PSNR values of a single image from Set12 obtained using the proposed method and the above-

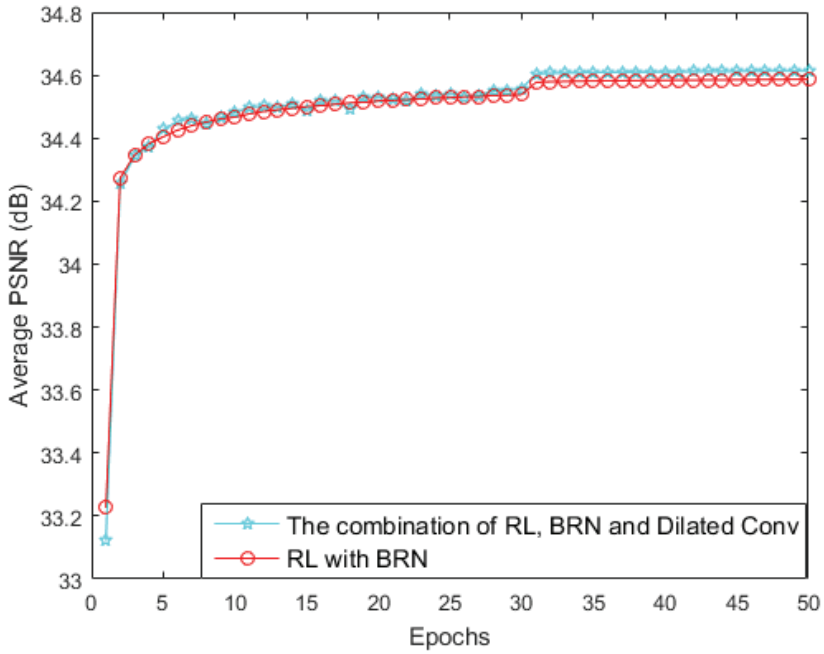

Figure 7: Gaussian denoising results of two specific models, one consisting of BRN and RL and the other of RL, BRN, and Dilated Conv. Both were trained with $\sigma = 15$. Results for 68 images from the CBSD68 dataset were are evaluated.

Table 4: Average PSNR (dB) results of different methods on BSD68 dataset with noise levels of 15, 25 and 50.

Methods	BM3D	WNNM	EPLL	MLP	CSF	TNRD	DnCNN	IRCNN	FFDNet	BRDNet
$\sigma = 15$	31.07	31.37	31.21	-	31.24	31.42	31.72	31.63	31.62	31.79
$\sigma = 25$	28.57	28.83	28.68	28.96	28.74	28.92	29.23	29.15	29.19	29.29
$\sigma = 50$	25.62	25.87	25.67	26.03	-	25.97	26.23	26.19	26.30	26.36

mentioned methods. The best and second-best PSNR results for each method are highlighted in red and blue, respectively, in Table 5. Figure. 11 illustrates the visual results obtained from the aforementioned methods. It can be seen from the figure that the proposed method can recover and obtain clearer images compared with the other methods.

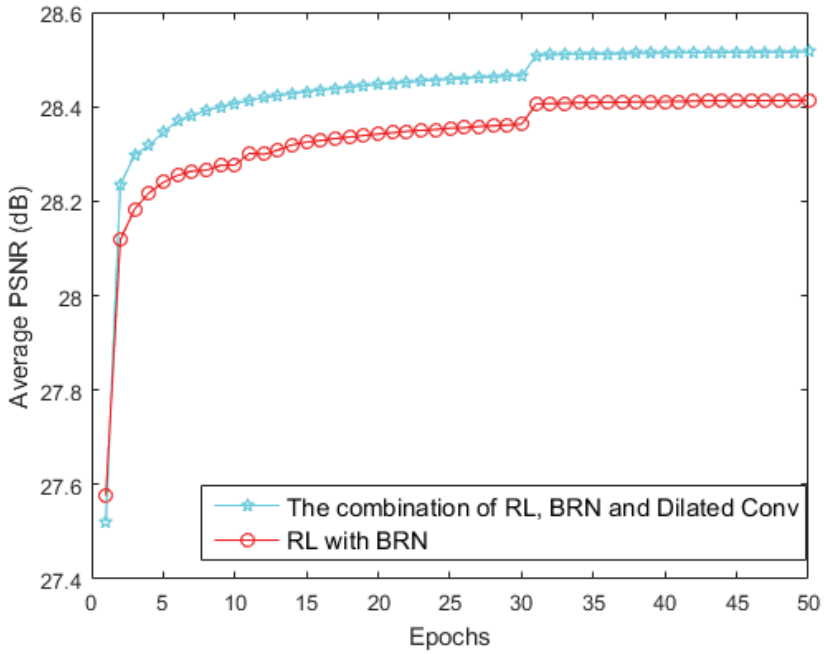

Figure 8: Gaussian denoising results of two specific models, one consisting of BRN and RL and the other consists of RL, BRN and Dilated Conv. Both were are trained with $\sigma = 50$. Results for 68 images from the CBSD68 were evaluated.

Figure 10: Twelve widely-used test images in experiments.

For color-noisy image denoising, we exploit six noise levels (σ = 15, 25, 35, 50, 60, 75) to train the models. We compar BRDNet with the state-of-the-art methods on the CBSD68, Kodak24 [18], and McMaster datasets for color-image denoising. The best and second-best PSNR results for different σ values are highlighted in red and blue, respectively, in Table 6. It can be seen from the table that the proposed BRDNet is more effective than other methods in color-image denoising, which indicates that the proposed method is more robust to low- and high- level noises. For example, for Kodak24 and $\sigma = 75$, the average PSNR of BRDNet is 0.24 dB higher than that of FFDNet. For McMaster and $\sigma = 15$, the average PSNR of BRDNet is 0.50 dB higher than that

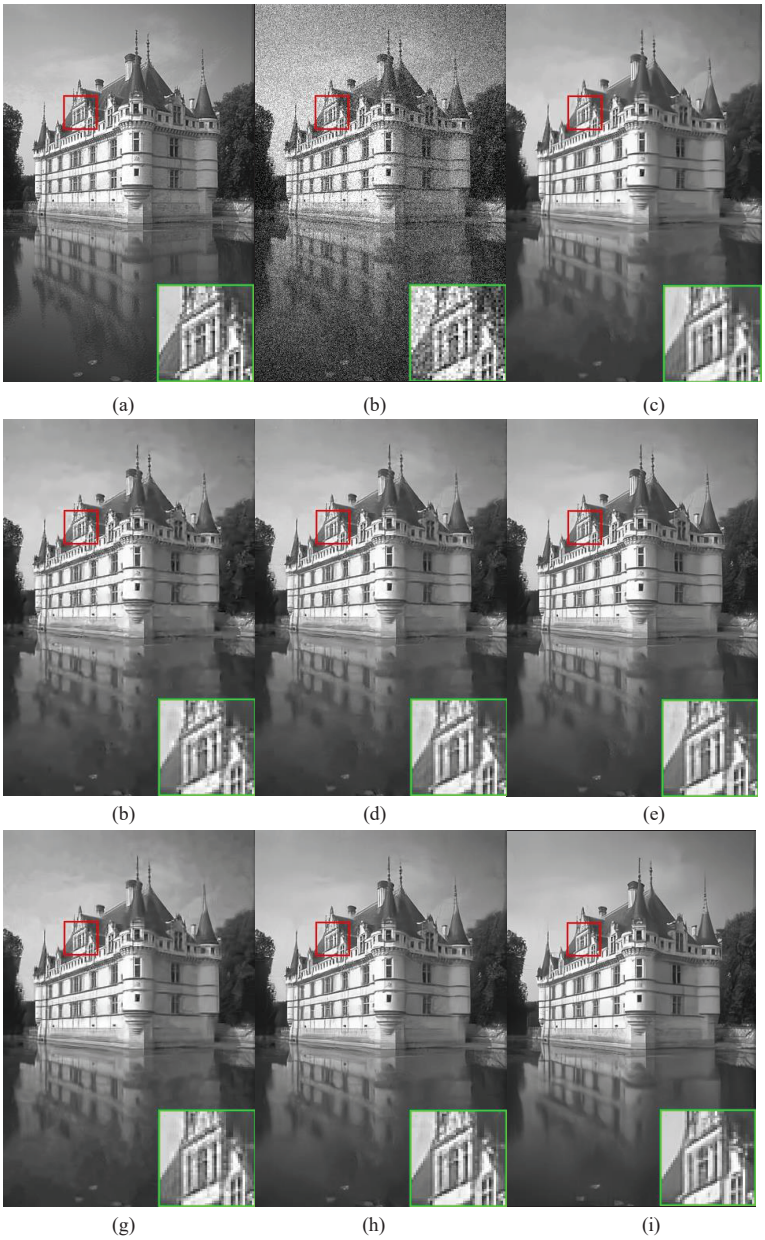

Figure 9: Denoising results of one image from the BSD68 dataset with noise level 25 using for different methods: (a) original image, (b) noisy image /20.30 dB, (c) WNNM/29.75 dB, (d) E-PLL/29.59 dB, (e) TNRD/29.76 dB, (f) DnCNN/30.16 dB, (g) BM3D/29.53 dB, (h) IRCNN/30.07 dB, and(i) BRDNet/30.27 dB.

of IRCNN.

Figures. 12 and 13 vividly describe the results of different methods with $\sigma = 35$ and $\sigma = 60$, respectively, for color-image denoising. One can see from the figures that the clean images

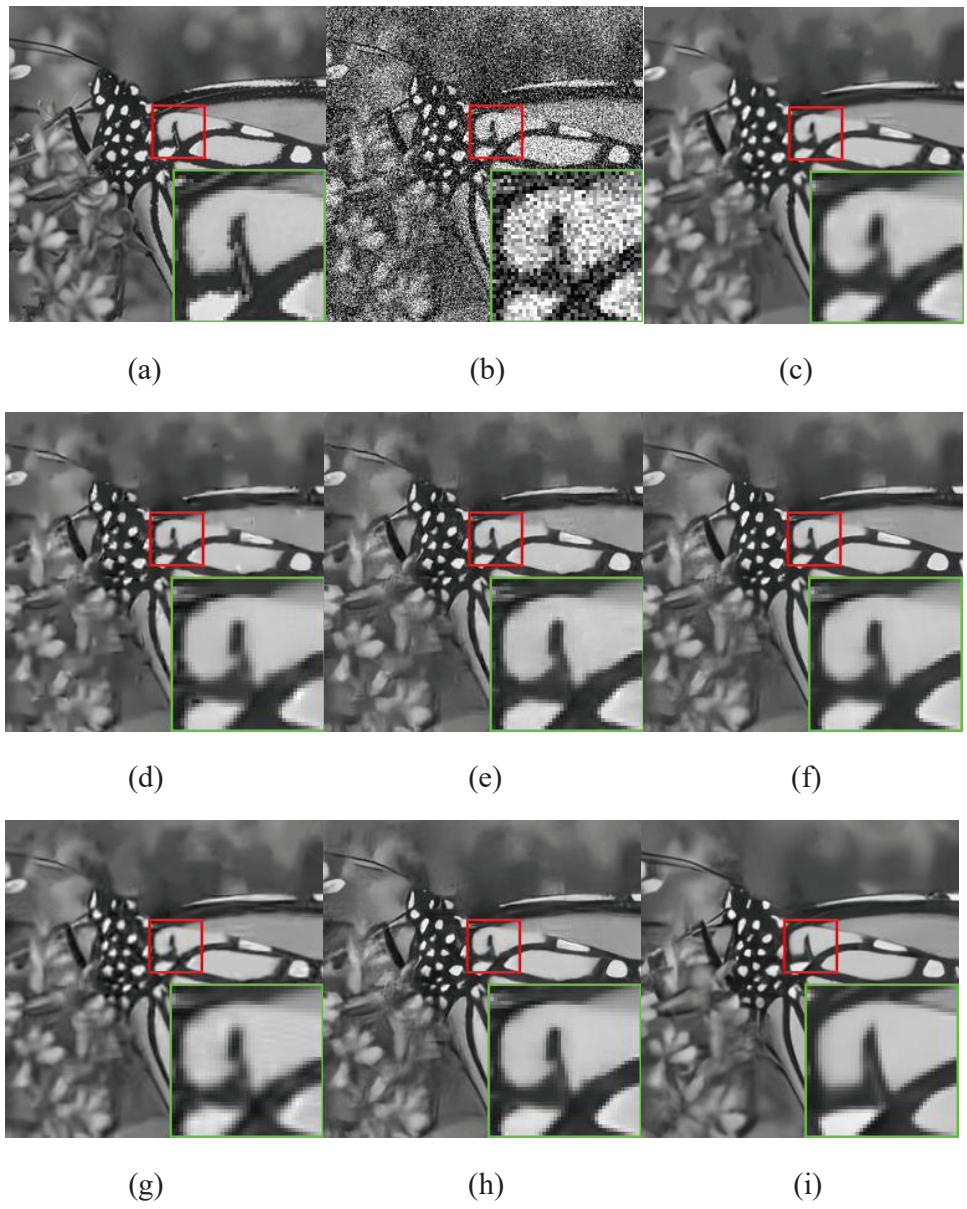

Figure 11: Denoising results of image "monar" from Set12 with noise level 50 using different methods: (a) original image, (b) noisy image/14.71 dB, (c) WNNM/26.32 dB, (d) EPLL/25.94 dB, (e) TNRD/26.31 dB, (f) DnCNN/26.78 dB, (g) BM3D/25.82 dB, (h) IRCNN/26.61 dB, and(i) BRDNet/26.97 dB.

obtained using the proposed method are clearer than those obtained using other methods, that is, the proposed method is more suitable for color-image denoising. From the experimental results for the gray and color images, it can be seen that the proposed method is more robust and effective than other state-of-the-art methods in image denoising. From these results, it is known that the proposed BRDNet is superior to both traditional denoising methods, such as BM3D and state-of-

Table 5: PSNR (dB) results for different methods on 12 widely used images with noise levels of 15, 25 and 50.

Images	C.man	House	Peppers	Starfish	Monarch	Airplane	Parrot	Lena	Barbara	Boat	Man	Couple	Average
Noise level	$\sigma = 15$												
BM3D[11]	31.91	34.93	32.69	31.14	31.85	31.07	31.37	34.26	33.10	32.13	31.92	32.10	32.37
WNNM[19]	32.17	35.13	32.99	31.82	32.71	31.39	31.62	34.27	33.60	32.27	32.11	32.17	32.70
EPLL[68]	31.85	34.17	32.64	31.13	32.10	31.19	31.42	33.92	31.38	31.93	32.00	31.93	32.14
CSF[50]	31.95	34.39	32.85	31.55	32.33	31.33	31.37	34.06	31.92	32.01	32.08	31.98	32.32
TNRD[7]	32.19	34.53	33.04	31.75	32.56	31.46	31.63	34.24	32.13	32.14	32.23	32.11	32.50
DnCNN[61]	32.61	34.97	33.30	32.20	33.09	31.70	31.83	34.62	32.64	32.42	32.46	32.47	32.86
IRCNN[62]	32.55	34.89	33.31	32.02	32.82	31.70	31.84	34.53	32.43	32.34	32.40	32.40	32.77
FFDNet[63]	32.43	35.07	33.25	31.99	32.66	31.57	31.81	34.62	32.54	32.38	32.41	32.46	32.77
BRDNet	32.80	35.27	33.47	32.24	33.35	31.85	32.00	34.75	32.93	32.55	32.50	32.62	33.03
Noise level	$\sigma = 25$												
BM3D[11]	29.45	32.85	30.16	28.56	29.25	28.42	28.93	32.07	30.71	29.90	29.61	29.71	29.97
WNNM[19]	29.64	33.22	30.42	29.03	29.84	28.69	29.15	32.24	31.24	30.03	29.76	29.82	30.26
EPLL[68]	29.26	32.17	30.17	28.51	29.39	28.61	28.95	31.73	28.61	29.74	29.66	29.53	29.69
MLP[4]	29.61	32.56	30.30	28.82	29.61	28.82	29.25	32.25	29.54	29.97	29.88	29.73	30.03
CSF[50]	29.48	32.39	30.32	28.80	29.62	28.72	28.90	31.79	29.03	29.76	29.71	29.53	29.84
TNRD[7]	29.72	32.53	30.57	29.02	29.85	28.88	29.18	32.00	29.41	29.91	29.87	29.71	30.06
DnCNN[61]	30.18	33.06	30.87	29.41	30.28	29.13	29.43	32.44	30.00	30.21	30.10	30.12	30.43
IRCNN[62]	30.08	33.06	30.88	29.27	30.09	29.12	29.47	32.43	29.92	30.17	30.04	30.08	30.38
FFDNet[63]	30.10	33.28	30.93	29.32	30.08	29.04	29.44	32.57	30.01	30.25	30.11	30.20	30.44
BRDNet	31.39	33.41	31.04	29.46	30.50	29.20	29.55	32.65	30.34	30.33	30.14	30.28	30.61
Noise level	$\sigma = 50$												
BM3D[11]	26.13	29.69	26.68	25.04	25.82	25.10	25.90	29.05	27.22	26.78	26.81	26.46	26.72
WNNM[19]	26.45	30.33	26.95	25.44	26.32	25.42	26.14	29.25	27.79	26.97	26.94	26.64	27.05
EPLL[68]	26.10	29.12	26.80	25.12	25.94	25.31	25.95	28.68	24.83	26.74	26.79	26.30	26.47
MLP[4]	26.37	29.64	26.68	25.43	26.26	25.56	26.12	29.32	25.24	27.03	27.06	26.67	26.78
TNRD[7]	26.62	29.48	27.10	25.42	26.31	25.59	26.16	28.93	25.70	26.94	26.98	26.50	26.81
DnCNN[61]	27.03	30.00	27.32	25.70	26.78	25.87	26.48	29.39	26.22	27.20	27.24	26.90	27.18
IRCNN[62]	26.88	29.96	27.33	25.57	26.61	25.89	26.55	29.40	26.24	27.17	27.17	26.88	27.14
FFDNet[63]	27.05	30.37	27.54	25.75	26.81	25.89	26.57	29.66	26.45	27.33	27.29	27.08	27.32
BRDNet	27.44	30.53	27.67	25.77	26.97	25.93	26.66	29.73	26.85	27.38	27.27	27.17	27.45

the-art denoising methods, such as DnCNN in gray and color noisy images.

Figure 12: Denoising results for one color image from the McMaster dataset with noise level 35: (a) original image/ $\sigma = 35$, (b) noisy image/18.62 dB, (c) CBM3D/31.04 dB, (d) FFDNet/31.94 dB, and (e) BRDNet/32.25 dB.

Table 6: Average PSNR (dB) results of different methods on the CBSD68, Kodak24, and McMaster datasets with noise levels of 15, 25, 35, 50, and 75.

Datasets	Methods	$\sigma = 15$	$\sigma = 25$	$\sigma = 35$	$\sigma = 50$	$\sigma = 75$
CBSD68	CBM3D[11]	33.52	30.71	28.89	27.38	25.74
	FFDNet[63]	33.80	31.18	29.57	27.96	26.24
	DnCNN[61]	33.98	31.31	29.65	28.01	-
	IRCNN[62]	33.86	31.16	29.50	27.86	-
	BRDNet	34.10	31.43	29.77	28.16	26.43
Kodak24	CBM3D[11]	34.28	31.68	29.90	28.46	26.82
	FFDNet[63]	34.55	32.11	30.56	28.99	27.25
	DnCNN[61]	34.73	32.23	30.64	29.02	-
	IRCNN[62]	34.56	32.03	30.43	28.81	-
	BRDNet	34.88	32.41	30.80	29.22	27.49
McMaster	CBM3D[11]	34.06	31.66	29.92	28.51	26.79
	FFDNet[63]	34.47	32.25	30.76	29.14	27.29
	DnCNN[61]	34.80	32.47	30.91	29.21	-
	IRCNN[62]	34.58	32.18	30.59	28.91	-
	BRDNet	35.08	32.75	31.15	29.52	27.72

Figure 13: Denoising results for one color image from the Kodak24 dataset with noise level 60: (a) original image/ $\sigma = 60$, (b) noisy image/13.45 dB, (c) CBM3D/31.00 dB, (d) FFDNet/31.49 dB, and (e) BRDNet/31.85 dB.

4.5. BRDNet for Real Noisy Image Denoising

To test the performance of the proposed method for real noisy images, we choose the popular methods, e.g., color block-matching and 3-D filtering(CBM3D), MLP , TNRD, DnCNN, CSF [50], noise clinic (NC) [36, 35] and WNNM [19] to design contrast experiments. It can be seen from Table 7 that using the proposed method results in 0.12 dB, 0.96 dB and 2.87 dB improvements over TNRD, WNNM, and DnCNN, respectively. In the table, red and blue entries represent the best and second-best results, respectively, for condition in Table 7. Thus, the proposed method is more suitable to deal with more complex noisy images, such as real noisy images.

4.6. Running Time

Testing speed is a more important index than training speed in evaluating performance for low-level vision [11, 19, 50]. We therefore compare BM3D, WNNM, EPLL, MLP, TNRD, CSF and DnCNN with the proposed method in running time experiments, using gray noisy images of sizes

Table 7: PSNR (dB) results for different methods on real noisy images.

Camera settings	CBM3D [11]	MLP [4]	TNRD [7]	DnCNN [61]	CSF [50]	NC [36]	WNNM [19]	BRDNet
Canon 5D ISO=3200	39.76	39.00	39.51	37.26	35.68	38.76	37.51	37.63
	36.40	36.34	36.47	34.13	34.03	35.69	33.86	37.28
	36.37	36.33	36.45	34.09	32.63	35.54	31.43	37.75
Nikon D600 ISO=3200	34.18	34.70	34.79	33.62	31.78	35.57	33.46	34.55
	35.07	36.20	36.37	34.48	35.16	36.70	36.09	35.99
	37.13	39.33	39.49	35.41	39.98	39.28	39.86	38.62
Nikon D800 ISO=1600	36.81	37.95	38.11	35.79	34.84	38.01	36.35	39.22
	37.76	40.23	40.52	36.08	38.42	39.05	39.99	39.67
	37.51	37.94	38.17	35.48	35.79	38.20	37.15	39.04
Nikon D800 ISO=3200	35.05	37.55	37.69	34.08	38.36	38.07	38.60	38.28
	34.07	35.91	35.90	33.70	35.53	35.72	36.04	37.18
	34.42	38.15	38.21	33.31	40.05	36.76	39.73	38.85
Nikon D800 ISO=6400	31.13	32.69	32.81	29.83	34.08	33.49	33.29	32.75
	31.22	32.33	32.33	30.55	32.13	32.79	31.16	33.24
	30.97	32.29	32.29	30.09	31.52	32.86	31.98	32.89
Average	35.19	36.46	36.61	33.86	35.33	36.43	35.77	36.73

Table 8: Run time for different methods in denoising images of sizes 256×256, 512×512, and 1024×1024.

Methods	Device	256×256	512×512	1024×1024
BM3D[11]	CPU	0.59	2.52	10.77
WNNM[19]	CPU	203.1	773.2	2536.4
EPLL[68]	CPU	25.4	45.5	422.1
MLP[4]	CPU	1.42	5.51	19.4
TNRD[7]	CPU	0.45	1.33	4.61
CSF[50]	GPU	-	0.92	1.72
DnCNN[61]	GPU	0.036	0.111	0.410
BRDNet	GPU	0.062	0.207	0.788

of 256×256, 512×512, and 1024×1024 with $\sigma = 25$. Also, we find that BRDNet is also very competitive with some methods on a GPU, such as DnCNN, as shown in Table 8.

A good denoiser should realize a tradeoff between PSNR and running time [62]. The effectiveness of the proposed BRDNet with BRN has been proved via previously described experiments. For example, BRDNet with BRN has superiority to low-configuration hardware, such as GTX960 and GTX970. Further, due to the complementarity of two sub-networks, BRDNet is more robust than state-of-the-art denoising methods, such as DnCNN in color synthetic and real noisy images. In addition, shallow architecture of BRDNet with dilated convolutions is also very competitive with two DnCNNs in performance and complexity for image denoising as shown in Tables 1 and 9, respectively. That is, the proposed BRDNet has small computational cost, which is very suitable to smart phone and camera. In summary, those experiments all prove that the proposed BRDNet is a stronger denoiser.

5. Conclusions

In this paper, we propose a novel model-based-CNN denoiser named BRDNet, that combines two different networks to enhance image-denoising performance. Moreover, BRDNet uses BRN,

Table 9: Complexity analysis of BRDNet, DnCNN and two DnCNNs.

Methods	Parameters	GFlops
DnCNN[61]	0.56M	1.40
Concatenation of two DnCNNs	1.11M	2.78
BRDNet	1.11M	2.78

RL, and dilated convolutions to improve the denoising performance, and make the model more easily trained. BRN is used not only to accelerate the convergence of BRDNet, but also to address the small-batch problem. The RL is applied to separate the noise from noisy images, and to obtain latent clean images in BRDNet. Dilated convolutions can enlarge the receptive field to obtain more context information. Experimental results show that BRDNet is very competitive with other state-of-the-art methods for image denoising. In the future, we plan to use CNN with prior knowledge to deal with more complex real noisy image denoising, such as low-light and blurred images.

Acknowledgments

This paper is supported in part by the National Nature Science Foundation of China (Grant no. 61876051), in part by the Shenzhen Municipal Science and Technology Innovation Council (Gant no. JCYJ20170811155725434).

References

[1] Barbu, A., 2009. Learning real-time mrf inference for image denoising.

[2] Barbu, A., 2009. Training an active random field for real-time image denoising. IEEE Transactions on Image Processing 18 (11), 2451–2462.

[3] Beck, A., Teboulle, M., 2009. Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems. IEEE Transactions on Image Processing 18 (11), 2419–2434.

[4] Burger, H. C., Schuler, C. J., Harmeling, S., 2012. Image denoising: Can plain neural networks compete with bm3d? In: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on. IEEE, pp. 2392–2399.

[5] Chambolle, A., 2004. An algorithm for total variation minimization and applications. Journal of Mathematical imaging and vision 20 (1-2), 89–97.

[6] Chen, J., Chen, J., Chao, H., Yang, M., 2018. Image blind denoising with generative adversarial network based noise modeling. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3155–3164.

[7] Chen, Y., Pock, T., 2017. Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration. IEEE transactions on pattern analysis and machine intelligence 39 (6), 1256–1272.

[8] Chollet, F., et al., 2015. Keras: Deep learning library for theano and tensorflow. URL: https://keras. io/k 7 (8).

[9] Chu, M., Wu, S., Gu, Y., Xu, Y., 2017. Rich features and precise localization with region proposal network for object detection. In: Chinese Conference on Biometric Recognition. Springer, pp. 605–614.

[10] Cruz, C., Foi, A., Katkovnik, V., Egiazarian, K., 2018. Nonlocality-reinforced convolutional neural networks for image denoising. arXiv preprint arXiv:1803.02112.

[11] Dabov, K., Foi, A., Katkovnik, V., Egiazarian, K., 2007. Image denoising by sparse 3-d transform-domain collaborative filtering. IEEE Transactions on image processing 16 (8), 2080–2095.

[12] Dong, C., Loy, C. C., He, K., Tang, X., 2016. Image super-resolution using deep convolutional networks. IEEE transactions on pattern analysis and machine intelligence 38 (2), 295–307.

[13] Dong, W., Wang, P., Yin, W., Shi, G., Wu, F., Lu, X., 2018. Denoising prior driven deep neural network for image restoration. arXiv preprint arXiv:1801.06756.

[14] Dong, W., Zhang, L., Shi, G., Li, X., 2013. Nonlocally centralized sparse representation for image restoration. IEEE Transactions on Image Processing 22 (4), 1620–1630.
[15] Du, B., Wei, Q., Liu, R., 2019. An improved quantum-behaved particle swarm optimization for endmember extraction. IEEE Transactions on Geoscience and Remote Sensing.
[16] Du, B., Xiong, W., Wu, J., Zhang, L., Zhang, L., Tao, D., 2017. Stacked convolutional denoising auto-encoders for feature representation. IEEE transactions on cybernetics 47 (4), 1017–1027.
[17] Fei, L., Lu, G., Jia, W., Teng, S., Zhang, D., 2018. Feature extraction methods for palmprint recognition: A survey and evaluation. IEEE Transactions on Systems, Man, and Cybernetics: Systems.
[18] Franzen, R., 1999. Kodak lossless true color image suite. source: http://r0k. us/graphics/kodak 4.
[19] Gu, S., Zhang, L., Zuo, W., Feng, X., 2014. Weighted nuclear norm minimization with application to image denoising. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2862–2869.
[20] Guo, K., Wu, S., Xu, Y., 2017. Face recognition using both visible light image and near-infrared image and a deep network. CAAI Transactions on Intelligence Technology 2 (1), 39–47.
[21] He, K., Zhang, X., Ren, S., Sun, J., 2014. Spatial pyramid pooling in deep convolutional networks for visual recognition. In: European conference on computer vision. Springer, pp. 346–361.
[22] He, K., Zhang, X., Ren, S., Sun, J., 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: Proceedings of the IEEE international conference on computer vision. pp. 1026–1034.
[23] He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778.
[24] Ioffe, S., 2017. Batch renormalization: Towards reducing minibatch dependence in batch-normalized models. In: Advances in Neural Information Processing Systems. pp. 1945–1953.
[25] Ioffe, S., Szegedy, C., 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167.
[26] Jiao, J., Tu, W.-C., He, S., Lau, R. W., 2017. Formresnet: formatted residual learning for image restoration. In: Computer Vision and Pattern Recognition Workshops (CVPRW), 2017 IEEE Conference on. IEEE, pp. 1034–1042.
[27] Kim, J., Kwon Lee, J., Mu Lee, K., 2016. Accurate image super-resolution using very deep convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1646–1654.
[28] Kim, J., Kwon Lee, J., Mu Lee, K., 2016. Deeply-recursive convolutional network for image super-resolution. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1637–1645.
[29] Kingma, D., Ba, J., 2014. Adam: A method for stochastic optimization. Computer Science.
[30] Kligvasser, I., Shaham, T. R., Michaeli, T., 2017. xunit: Learning a spatial activation function for efficient image restoration. arXiv preprint arXiv:1711.06445.
[31] Kokkinos, F., Lefkimmiatis, S., 2018. Iterative residual network for deep joint image demosaicking and denoising. arXiv preprint arXiv:1807.06403.
[32] Krizhevsky, A., Sutskever, I., Hinton, G. E., 2012. Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. pp. 1097–1105.
[33] Lai, W.-S., Huang, J.-B., Ahuja, N., Yang, M.-H., 2017. Deep laplacian pyramid networks for fast and accurate superresolution. In: IEEE Conference on Computer Vision and Pattern Recognition. Vol. 2. p. 5.
[34] Latif, G., Iskandar, D. A., Alghazo, J., Butt, M., Khan, A. H., 2018. Deep cnn based mr image denoising for tumor segmentation using watershed transform. International Journal of Engineering & Technology 7 (2.3), 37–42.
[35] Lebrun, M., Colom, M., Morel, J.-M., 2015. Multiscale image blind denoising. IEEE Transactions on Image Processing 24 (10), 3149–3161.
[36] Lebrun, M., Colom, M., Morel, J.-M., 2015. The noise clinic: a blind image denoising algorithm. Image Processing On Line 5, 1–54.
[37] Li, S., Yin, H., Fang, L., 2012. Group-sparse representation with dictionary learning for medical image denoising and fusion. IEEE Transactions on biomedical engineering 59 (12), 3450–3459.
[38] Lin, G., Milan, A., Shen, C., Reid, I. D., 2017. Refinenet: Multi-path refinement networks for high-resolution

semantic segmentation. In: Cvpr. Vol. 1. p. 5.
[39] Long, J., Shelhamer, E., Darrell, T., 2015. Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3431–3440.
[40] Lu, Y., Yuan, C., Zhu, W., Li, X., 2018. Structurally incoherent low-rank nonnegative matrix factorization for image classification. IEEE Transactions on Image Processing 27 (11), 5248–5260.
[41] Ma, K., Duanmu, Z., Wu, Q., Wang, Z., Yong, H., Li, H., Zhang, L., 2017. Waterloo exploration database: New challenges for image quality assessment models. IEEE Transactions on Image Processing 26 (2), 1004–1016.
[42] Mairal, J., Bach, F., Ponce, J., Sapiro, G., Zisserman, A., 2009. Non-local sparse models for image restoration. In: Computer Vision, 2009 IEEE 12th International Conference on. IEEE, pp. 2272–2279.
[43] Malfait, M., Roose, D., 1997. Wavelet-based image denoising using a markov random field a priori model. IEEE Transactions on image processing 6 (4), 549–565.
[44] Nam, S., Hwang, Y., Matsushita, Y., Joo Kim, S., 2016. A holistic approach to cross-channel image noise modeling and its application to image denoising. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1683–1691.
[45] Pascanu, R., Mikolov, T., Bengio, Y., 2013. On the difficulty of training recurrent neural networks. In: International conference on machine learning. pp. 1310–1318.
[46] Plotz, T., Roth, S., 2017. Benchmarking denoising algorithms with real photographs. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1586–1595.
[47] Ren, C., He, X., Pu, Y., 2018. Nonlocal similarity modeling and deep cnn gradient prior for super resolution. IEEE Signal Processing Letters 25 (7), 916–920.
[48] Roth, S., Black, M. J., 2009. Fields of experts. International Journal of Computer Vision 82 (2), 205.
[49] Rudin, L. I., Osher, S., Fatemi, E., 1992. Nonlinear total variation based noise removal algorithms. Physica D: nonlinear phenomena 60 (1-4), 259–268.
[50] Schmidt, U., Roth, S., 2014. Shrinkage fields for effective image restoration. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2774–2781.
[51] Si-Yao, L., Ren, D., Hu, Z., Li, J., Yin, Q., Guo, P., 2018. A concatenated residual network for image deblurring. arXiv preprint arXiv:1804.06042.
[52] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A., 2015. Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1–9.
[53] Tai, Y., Yang, J., Liu, X., 2017. Image super-resolution via deep recursive residual network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Vol. 1. p. 5.
[54] Wang, T., Sun, M., Hu, K., 2017. Dilated residual network for image denoising. arXiv preprint arXiv:1708.05473.
[55] Wang, Z., Liu, D., Yang, J., Han, W., Huang, T., 2015. Deep networks for image super-resolution with sparse prior. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 370–378.
[56] Wen, J., Fang, X., Xu, Y., Tian, C., Fei, L., 2018. Low-rank representation with adaptive graph regularization. Neural Networks.
[57] Wen, J., Xu, Y., Li, Z., Ma, Z., Xu, Y., 2018. Inter-class sparsity based discriminative least square regression. Neural Networks 102, 36–47.
[58] Xu, J., Li, H., Liang, Z., Zhang, D., Zhang, L., 2018. Real-world noisy image denoising: A new benchmark. arXiv preprint arXiv:1804.02603.
[59] Xu, J., Zhang, L., Zhang, D., 2018. External prior guided internal prior learning for real-world noisy image denoising. IEEE Transactions on Image Processing 27 (6), 2996–3010.
[60] Yu, F., Koltun, V., 2015. Multi-scale context aggregation by dilated convolutions. arXiv preprint arXiv:1511.07122.
[61] Zhang, K., Zuo, W., Chen, Y., Meng, D., Zhang, L., 2017. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. IEEE Transactions on Image Processing 26 (7), 3142–3155.
[62] Zhang, K., Zuo, W., Gu, S., Zhang, L., 2017. Learning deep cnn denoiser prior for image restoration. In: IEEE Conference on Computer Vision and Pattern Recognition. Vol. 2.
[63] Zhang, K., Zuo, W., Zhang, L., 2018. Ffdnet: Toward a fast and flexible solution for cnn based image denoising.

IEEE Transactions on Image Processing.
[64] Zhang, K., Zuo, W., Zhang, L., 2018. Learning a single convolutional super-resolution network for multiple degradations. In: IEEE Conference on Computer Vision and Pattern Recognition. Vol. 6.
[65] Zhang, L., Wu, X., Buades, A., Li, X., 2011. Color demosaicking by local directional interpolation and nonlocal adaptive thresholding. Journal of Electronic imaging 20 (2), 023016.
[66] Zhang, L., Zuo, W., 2017. Image restoration: From sparse and low-rank priors to deep priors [lecture notes]. IEEE Signal Processing Magazine 34 (5), 172–179.
[67] Zhang, Y., Tian, Y., Kong, Y., Zhong, B., Fu, Y., 2018. Residual dense network for image super-resolution. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
[68] Zoran, D., Weiss, Y., 2011. From learning models of natural image patches to whole image restoration. In: Computer Vision (ICCV), 2011 IEEE International Conference on. IEEE, pp. 479–486.
[69] Zuo, W., Zhang, L., Song, C., Zhang, D., Gao, H., 2014. Gradient histogram estimation and preservation for texture enhanced image denoising. IEEE Trans. Image Processing 23 (6), 2459–2472.