Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

Dual-constraint burst image denoising method*

Dan ZHANG, Lei ZHAO‡, Duanqing XU, Dongming LU

Network and Media Laboratory, College of Computer Science and Technology, Zhejiang University, Hangzhou 310027, China

E-mail: cszhd@zju.edu.cn; cszhl@zju.edu.cn; xdq@zju.edu.cn; ldm@zju.edu.cn

Received July 17, 2020; Revision accepted Mar. 28, 2021; Crosschecked Oct. 12, 2021; Published online Jan. 24, 2022

Abstract: Deep learning has proven to be an effective mechanism for computer vision tasks, especially for image denoising and burst image denoising. In this paper, we focus on solving the burst image denoising problem and aim to generate a single clean image from a burst of noisy images. We propose to combine the power of block matching and 3D filtering (BM3D) and a convolutional neural network (CNN) for burst image denoising. In particular, we design a CNN with a divide-and-conquer strategy. First, we employ BM3D to preprocess the noisy burst images. Then, the preprocessed images and noisy images are fed separately into two parallel CNN branches. The two branches produce somewhat different results. Finally, we use a light CNN block to combine the two outputs. In addition, we improve the performance by optimizing the two branches using two different constraints: a signal constraint and a noise constraint. One maps a clean signal, and the other maps the noise distribution. In addition, we adopt block matching in the network to avoid frame misalignment. Experimental results on synthetic and real noisy images show that our algorithm is competitive with other algorithms.

Key words: Image denoising; Burst image denoising; Deep learning
https://doi.org/10.1631/FITEE.2000353 **CLC number:** TP391

1 Introduction

Images are crucial information carriers that are developed for recording, sharing, and analyzing messages. On one hand, with the development of science and technology, people require a high-quality life. Image quality directly determines our visual enjoyment. On the other hand, image quality influences the efficiency and reliability of image processing and image analysis. Clarity has a further influence on massive vision tasks like object recognition and tracking. Above all, image denoising is an essential task in computer vision.

The purpose of burst image denoising and single image denoising is to recover latent information from corrupted images. The imaging device and environment have significant effects on image quality. When light is limited, current major cameras embedded in mobile devices still take pictures with noise. Setting a longer exposure time and a lower ISO when capturing images can reduce noise, but for handheld devices, unwanted motion blur is induced when the integration time is much longer. Therefore, many images are still captured in a short time with a higher ISO. Burst mode continuously takes multiple pictures of the same scene. The resulting burst images are noisy but not blurred. Burst mode is ubiquitous in current cameras. There has been research on

‡ Corresponding author

* Project supported by the National Major Program of Key Technology Research of Database Construction and Intelligent Retrieval of Culture Relics Prohibited from Trading, China (No. 2020YFC1523202), the National Major Social Science Fund of Collation and Comprehensive Study of Bronze Ware Data in HAN Dynasty, China (No. 19ZDA197), the Zhejiang Fund of Ancient Painting Image Restoration Based on Prior Background Knowledge Constraints, China (No. LY21F020005), and the Key Scientific Research Base for Digital Conservation of Cave Temples (Zhejiang University), the State Administration for Cultural Heritage, China

ORCID: Dan ZHANG, https://orcid.org/0000-0002-5033-8128; Lei ZHAO, https://orcid.org/0000-0003-4791-454X

recovering clean images from burst images.

Regarding the unwanted degradation as additive white Gaussian noise (AWGN), the problem is described as $X_i = Z_i + N_i$, where X_i is the corrupted image, Z_i stands for the clean signal, N_i is the random noise, and i stands for different capturing moments. The objective of the burst image denoising algorithm, Alg, is to estimate a clean image $\hat{Z}$ from the observations X_i, that is, $\hat{Z} = \mathrm{Alg}(X_i)$. In this paper, we use four burst images for denoising, i.e., $i = 1,\ 2,\ 3,\ 4$.

Generally, denoising algorithms can be classified into three categories: spatial domain algorithms, transform domain algorithms, and learning-based algorithms. Leading spatial domain methods (Tomasi and Manduchi, 1998; Chambolle, 2004; Buades et al., 2005) examine the similarity between local image pixels and non-local image patches. Transform domain approaches explore internal relations in the transform domain. Block matching and 3D filtering (BM3D) (Dabov et al., 2007), representative of the transform domain approaches, fuses the idea of patch similarity (Buades et al., 2005) and the wavelet transform domain method. Learning-based methods design an algorithm to learn a model from an image dataset. K-SVD (Aharon et al., 2006) and recently popular deep learning methods are efficient for image denoising tasks. When considering burst images, one choice is to denoise burst images individually by a single image denoising method and then calculate their mean value as the final estimate. Other researchers use multi-frame images as an entire input and explore their inner relationships (Mildenhall et al., 2018; Tassano et al., 2019).

The existing multi-frame denoising methods have demonstrated the principle that multi-frame images contain more information than single-frame images. Unfortunately, the performance is limited and is sensitive to the alignment error. In this work, we adopt block matching in the convolutional neural network (CNN) framework for alignment. Yang and Sun (2018) proposed to fuse the BM3D pipeline into their CNN model. The result is attractive but not outstanding, compared to deep neural network models. We suggest using BM3D as a preprocessor for the deep CNN model. We increase the width of the network by combining two branches. The two branches accept the noisy images and the preprocessed images, and can therefore capture more features. In addition, we assign different losses on the two branches, because Mosseri et al. (2013) indicated that pixels behave with different sensitivities to different denoising methods. A denoising method may output a favorable result for some patches, while the denoised version of other patches is over-smoothed or still noisy. We use the two losses to constrain both the signal and noise distributions. The contributions of this paper can be summarized as follows:

1. We propose a neural network that directly maps multi-frame images to clean images. We combine two branches to increase the width of the network. Specifically, the two branches accept different inputs and thus obtain more features.

2. We prove that constraining the two branches based on different loss functions can improve the performance. The target-noise removal and texture preservation are combined in a light CNN block.

3. The framework is robust to the frame alignment error. We can restore misaligned frames by finding similar patches between multi-frame images.

2 Related work

In this section, we review several classical and popular image denoising approaches. Spatial domain methods are reviewed first, followed by transform domain methods and learning-based methods.

1. Spatial domain methods

Spatial domain methods work by exploring the spatial correlation between pixels. In the beginning, researchers studied only limited local pixel neighborhoods; then they expanded their scope to in-process images and even the whole image space based on patch similarity rather than spatial domain. The mean filter, Gaussian filter, and anisotropic filter are presented based on the assumption that a clean image is usually smooth. The mean filter calculates the values of denoised pixels using a straightforward average of their neighborhood pixels. Later research focused on choosing the size, direction, parameter, and weight of the neighborhood. The Gaussian filter is also based on neighborhood averaging, but it uses a weighted average strategy. The weight depends on the location relationship of the pixels. The Gaussian filter proposes the assignment of a higher weight to nearby pixels, and states that the smaller the physical distance between pixels, the larger the effect. Perona and Malik (1990) suggested accepting

an image as a heat field. They stated that pixels always flow to the one that has a similar intensity according to the anisotropic diffusion theory. Specifically, there is a great difference between the values of the edge and its non-edge neighborhoods. The edge can reserve its information because it does not diffuse in non-edge directions. The non-edge pixel is similar to its neighborhoods, it quickly diffuses to its neighborhoods, and the noise is removed by smoothing. Tomasi and Manduchi (1998) calculated the weight according to the pixel value and its spatial location. The algorithm reveals that similar intensity and spatial relations are significant factors for filtering. They assigned a large weight to pixels that are close to one another in position and similar in intensity. Buades et al. (2005) proposed to use redundancy in images. The non-local mean (NLM) method is superior when the image is strongly self-similar. At first, they cropped the noisy image into small patches and considered the similarity between cropped patches for collaborative processing. The non-local idea is illuminating and efficient, but has high computation complexity.

2. Transform domain methods

Transform domain methods design a certain transform of the image. First, the image from the spatial domain is transformed to the frequency domain. Then, the noise and signal are separated in the frequency domain. After that, the image is transformed back into the spatial domain by the inverse transform. The best recommended transforms are Fourier transform, discrete cosine transform, and wavelet transform. The Haar wavelet transform is a typical and efficient wavelet that is used for denoising, in which the signal is represented based on a set of Haar basis functions. A series of multi-level Haar wavelet transforms decomposes the image into different frequencies. Because noise usually centralizes separately in high frequency, we remove it by applying a suitable threshold function on the coefficient. BM3D combines the idea of non-local similarity and the 3D transform domain, and state-of-the-art performance is achieved. It is a two-stage algorithm and there are three steps in each stage: grouping, collaborative filtering, and aggregation. Grouping stacks similar patches into a 3D block. In the first stage, similar patches are selected based on the noisy image, but in the second stage, patch similarities are calculated depending on the filtered result of the first stage. Collaborative filtering transforms the 3D block using hard thresholding (stage two uses Winer filtering), shrinks the spectrum coefficients to remove noise, and produces the estimated block using an inverse 3D transform. Aggregation generates the final estimate by averaging the front blocks. To denoise color images, CBM3D is born. VBM3D and VBM4D are extended for multi-frame image denoising. The BM3D family is still competitive in image denoising algorithms.

3. Learning-based methods

Learning-based methods are sometimes accomplished through learning a dictionary. K-SVD (Aharon et al., 2006) is a dictionary-learning algorithm. An overcomplete dictionary is built by sparsity, and denoising is done by selecting proper dictionary items and coefficients to approximate the image. Another class of learning-based methods learns a neural network that trains on a large dataset. The multi-layer perception (MLP) model was proposed to solve the image denoising problem (Burger et al., 2012). This work achieved excellent results on AWGN using an MLP model with four hidden layers. The model can map a noisy patch to a noise-free patch. Another widely used network is the CNN (LeCun et al., 1998). CNNs are competitive in solving various computer vision problems. The auto-encoder, successfully designed for unsupervised feature learning, was introduced in image denoising by Vincent et al. (2010). This work also integrated sparse coding in auto-encoders. Mao et al. (2016) introduced a deep encoding-decoding network for image denoising. The framework consists mainly of symmetric convolutional layers and deconvolutional layers. Special skip connections were introduced to overcome the gradient vanishing problem and recover detailed textures. To overcome the difficulty of creating a large dataset of noisy and clean image pairs, Lehtinen et al. (2018), Lempitsky et al. (2018), and Krull et al. (2019) proposed to learn from the corrupted images only. Deep image prior model was proposed to restore the clean image from the noisy image itself. Based on this principle, one needs only to train on the noisy images. The input is a random noise map, and the learning target is the corrupted image. Different from the deep image prior model, the Noise2Noise model minimizes the loss between two corrupted observations. After training on noisy image pairs, the model can output the clean image.

When considering burst images, one strategy is to recover images using the above algorithms individually and then calculate their mean value for the final estimate. Other researchers have developed new methods of collaborative denoising by considering all the burst images simultaneously (Liu et al., 2014; Godard et al., 2018; Mildenhall et al., 2018). Mildenhall et al. (2018) suggested training a kernel prediction network (KPN) for burst image denoising. Different from the single-image denoising network, KPN takes all burst images as the input. When testing, they recovered each of the burst images using the predicted kernels; the final result is the mean value of the deconvolved images. Liu et al. (2014) proposed several tactics for faster and more efficient burst image denoising using the homography flow to accelerate the alignment operation. They explored how to use consistent pixels at every pixel location to solve the scene motion problem. The final estimate was obtained by temporal and multi-scale pixel fusion. Godard et al. (2018) implemented fast and efficient burst denoising and video denoising using well-designed spatial and temporal blocks.

3 Proposed method

In this study, we propose a deep CNN to preserve texture details while removing noise from burst images. We increase the width of the network by combining two branches to improve the expression ability of the network. Specifically, we use two parallel CNN blocks to capture features separately from preprocessed images and noisy images. Then we employ a light CNN block to combine the results of the two branches. In addition, we constrain one branch on noise loss (Zhang et al., 2017, 2018) and the other on signal loss (Divakar and Babu, 2017; Godard et al., 2018). We also use a set of long skip connections to transfer the information and avoid the vanishing gradient problem. The frames are taken at different times, so they may not be well aligned. We pick the first frame as the reference and employ the block matching technique to resolve the frame alignment error.

An overview of the proposed method is described in Algorithm 1. First, the burst images are preprocessed using BM3D. Second, image patches are cropped from the noisy and preprocessed frames. Specifically, we solve the problem of frame misalignment by performing patch matching in the related frames. Extracting a patch p^{X_1} from the reference frame X_1 at location (w, h), the patches p^{X_2}, p^{X_3}, and p^{X_4} are selected from a local region surrounding (w, h) from related frames X_2, X_3, and X_4, respectively, according to the patch distance. Third, the patch groups are fed to the neural network and are recovered. Finally, we reconstruct the result using the estimated patches.

Algorithm 1 Dual-constraint algorithm for burst image denoising

1: **Input:** X_i, i=1, 2, 3, 4
2: **Output:** $\hat{Z}$
3: Preprocess X_i with BM3D, and mark the results Y_i
4: Crop burst images X_i to small 8×8 patches p^{X_i}, and perform decomposition on Y_i to obtain p^{Y_i}
5: Feed pairs p^{X_i} and p^{Y_i} to the proposed network, and output $\hat{p}$
6: Compose $\hat{p}$ to $\hat{Z}$

3.1 Network architecture

The problem is to restore a latent clean image from a set of noisy observations. Given the noisy frames X_i ($i = 1, 2, 3, 4$), the initial estimates Y_i ($i = 1, 2, 3, 4$) are generated using BM3D. We feed the noisy patches and the estimated patches to the dual-constraint network. The network can output a cleaner patch p:

$$p = M\left(p^{X_1}, ..., p^{X_N}, p^{Y_1}, ..., p^{Y_N}\right), \tag{1}$$

where p^{X_i} stands for the patches cropped from X_i. Patches cropped from Y_i are marked as p^{Y_i}. M is the trainable CNN model, as shown in Fig. 1. M consists mainly of convolutional layers, batch normalization layers, rectified linear unit (ReLU), and long skip connections.

Existing research has proved the superiority of the CNN for image denoising. We build the proposed model on the convolutional operation. The idea of using patch groups as the input was proposed by Xu et al. (2015) and Ahn and Cho (2017). Xu et al. (2015) directly concatenated the noisy and pilot patches and fed them to a single branch. Our proposed model deals with them using two parallel CNN branches. The two branches are trained to separately estimate clean patches from the preprocessed patches and noisy patches. The outputs show different levels of success for noise removal and texture

preservation; we use a light CNN block to combine the outputs of the two branches.

In our model, as shown in Fig. 1, the network contains convolutional layers, batch normalization layers, and ReLU. There are two parallel convolutional branches and a light CNN block. One of the branches takes noisy patches as the input, and the other accepts the preprocessed patches. Ahn and Cho (2017) and Zhang et al. (2017) have shown that deep networks built with small filters can achieve favorable performance. In our model, the filters are of size 3×3, stride 1, and padding 1. We adopt a deep structure that significantly improves the image processing tasks (Simonyan and Zisserman, 2014). Based on DnCNN and BMCNN, we set the depth of the two branches to 17. There is a simple way to combine the results of different branches by calculating their mean value. Instead, we employ a three-layer CNN to improve the performance. The convolutional layers, except for layer 17 and layer 20, contain batch normalization and ReLU operations. Layer 17 and the last layer use a single convolutional layer to construct the output. In general, the noisy patch group p^{X_i} and the preprocessed patch group p^{Y_i} are separately convolved by two parallel 17-layer blocks. The outputs $\hat{p}^X$ and $\hat{p}^Y$ are concatenated and fed into the three-layer convolutional block, which reconstructs the final estimate $\hat{p}$.

The other component, long skip connection, has been introduced to eliminate noise by Mao et al. (2016). We employ long skip connections to improve the performance. Because the burst images are captured at different moments, there may be some distinctions. We pick the first frame as the reference, and the output is a clean version of p^{X_1}. We use long skip connections to transfer information from the reference frame. That is, p^{X_1} and p^{Y_1} are skipped to layer 5 and layer 11 in different branches, respectively. p^{X_1} is also connected to the input of the combination block.

3.2 Loss function

The proposed model is optimized based on the minimization of the mean squared error (MSE). In line with other research, we assume that the corruption is caused by AWGN. The problem can be written as $X_i = Z_i + N_i$, where X_i is the observed value corroded by the environment, Z_i is the clean signal, N_i marks the depressing noise, and i represents

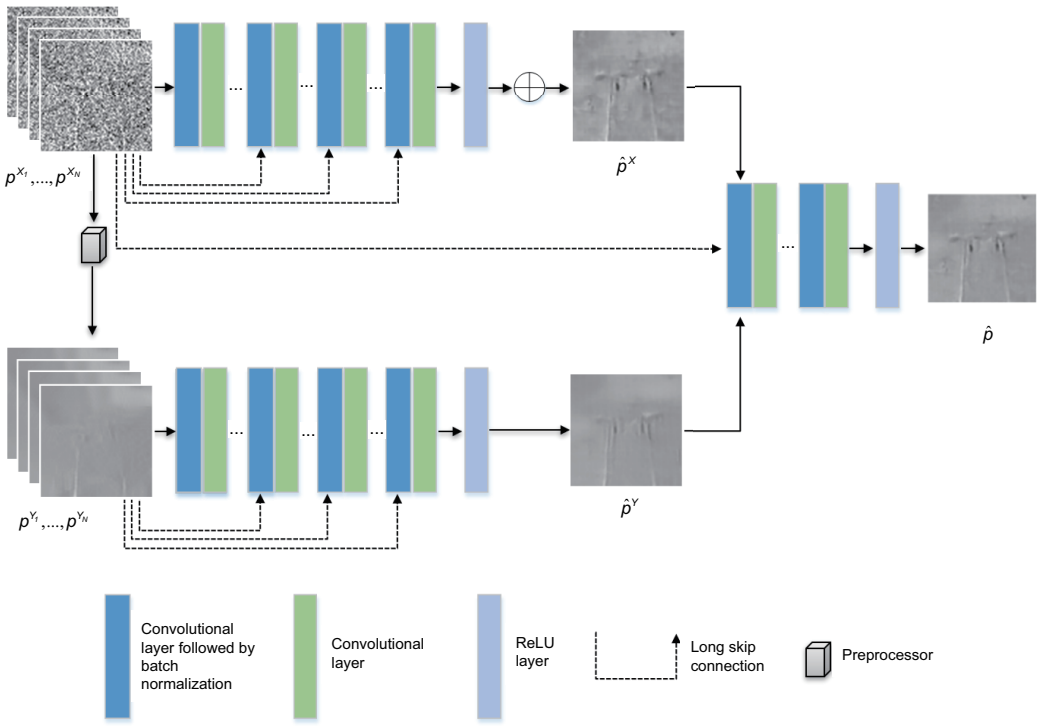

Fig. 1 Network architecture of the proposed method (References to color refer to the online version of this figure)

different capturing moments. In this study, we use four burst images and choose the first frame Z_1 as the reference; the target can roughly be written as

$$\text{argmin}((Z_1 - \hat{Z})^2 + (N_1 - \hat{N})^2). \tag{2}$$

Different from existing algorithms, we propose to minimize the MSE based on the combined signal and noise constraints. The former is the immediate loss for image processing problems, and Divakar and Babu (2017) and Godard et al. (2018) have proved its efficiency. The latter makes the use of residual learning. Simonyan and Zisserman (2014) promoted the performance of image classification and object detection by residual learning. Zhang et al. (2017) adopted residual learning and batch normalization in the network. Experimental results showed that these techniques contribute to the blind denoising and maintain the image texture.

Specifically, the loss function consists of three items:

$$L = \alpha L^{\text{noise}} + \beta L^{\text{signal}} + \gamma L^{\text{out}}. \tag{3}$$

L^{noise} is the loss function of the upper branch (branch X). It accepts the noisy image patches p^{X_i} and outputs the denoised patch $\hat{p}^X$. $\hat{p}^X$ is calculated as $\hat{p}^X = p^{X_1} - f_1(p^{X_i})$, and f_1 is the model that learns to map the noise distribution. We are now minimizing the loss as

$$L^{\text{noise}} = (p^{N_1} - f_1(p^{X_i}))^2. \tag{4}$$

L^{signal} is the loss function of the lower branch (branch Y). It receives the preprocessed image patches p^{Y_i} and generates the denoised patch $\hat{p}^Y$. Marking the model as f_2, we can write the loss as

$$L^{\text{signal}} = (p^{Z_1} - f_2(p^{Y_i}))^2. \tag{5}$$

L^{out} is the loss function of the final result. This model combines the results of f_1 and f_2. p^{X_1} is also added to transfer the information. Denoting the model as f_3, the loss is

$$L^{\text{out}} = \left[p^{Z_1} - f_3(f_1(p^{X_i}), f_2(p^{Y_i}), p^{X_1})\right]^2. \tag{6}$$

The loss contains three items. The constraints of the first item are based on the noise, which preserves more information. The constraints of the second item are based on the signal, which produces a smoother result. Because the two branches output different and complementary results, we combine them to generate a preferred result. Fig. 2 illustrates the visual comparison of different outputs. The first column shows the noisy images, the second column shows the output of the upper branch (branch X), the third column displays the result of the lower branch (branch Y), and the last column reveals the appearance of the clean images. At first glance, branch Y is a visually appealing picture, but details, like the eyeball in the first image, the line of the second image, and the spot on the third image, are smoothed out. The result of branch X is noisier but contains more details.

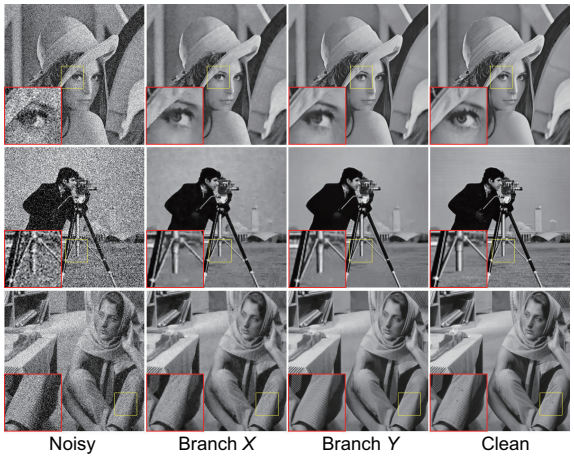

Fig. 2 Visual comparison of different branches. Branch X is constrained by the noise loss and branch Y is constrained by the signal loss

The parameters α, β, and γ balance the three parts. Based on the assumption that noise removal and detail preservation have equal importance in the final result, we suggest setting $\alpha = 0.3$, $\beta = 0.3$, and $\gamma = 0.4$. Although Mosseri et al. (2013) revealed that different patches have different preferences, in this work, we use a three-layer CNN block to learn the inner relationship. In future work, we will consider setting different weights based on the patch signal-to-noise ratio (PSNR).

3.3 Implementation

The proposed model is a fully CNN. We set the kernel size of the convolutional layers to 3×3, stride 1, and padding 1. The size of the output feature maps is the same as that of the input. The input has four channels that are composed of burst images

captured at different moments. The proposed model is patch-based, and the patch size is 8×8.

The loss is minimized by the momentum stochastic gradient descent solver. The learning rate is set to 10^{-2} at the beginning and momentum = 0.9. The net is well trained on NVIDIA GeForce GTX 1080 Ti, 16 GB memory, with a mini-batch size of 64.

4 Experiments and results

4.1 Dataset

We used 4961 images from Pascal VOC2007 for training. We experimented on the AWGN with noise level $\sigma = 50$. We randomly generated four noise maps and added them to the clean image individually. Then the noisy images were preprocessed by BM3D. After that, we extracted 1 350 000 patch pairs from the burst images. Each of the pairs contains nine patches: one clean patch, four noisy patches, and four preprocessed patches.

4.2 Ablation study

In this work, we introduced several components to improve the performance of burst image denoising. In this subsection, we trained seven models to demonstrate the effectiveness of those components. We compared different models for restoring images corrupted by AWGN of noise level $\sigma = 50$ in terms of PSNR. The result is shown in Table 1. The best results are in bold. The first row model contains a single branch and takes the noisy frames as the input. The second row model is similar to that in the first row, but the input of the second row model also contains the images preprocessed by BM3D. The PSNR values are increased by 0.70 dB on Set12 and 0.39 dB on BSD68. We conclude that the pilot images also contribute to the denoising performance. In addition, we proposed to increase the width of the network for burst image denoising. In the third and fourth rows, the network has two branches. The original noisy patches and the preprocessed patches are fed into two parallel branches and are convolved separately. Then a light CNN combines the outputs. The difference between the two models is the loss function. The third row model performs noise mapping on the two branches, and the fourth row model constrains the signal recovery. The results show that the loss function is the basis of denoising performance. We can improve the performance by widening the width of the network with a proper loss function. As illustrated in Fig. 2, the two different loss functions force the network to capture somewhat different information. We employed them in the parallel branches to obtain more features. The proposed architecture, shown in the fifth row of Table 1, achieves the highest PSNR. To verify the effectiveness of the long skip connections, we removed them from the proposed model. The result is shown in the sixth row in Table 1. The PSNR values are lower. Finally, we experimented on the combination block. In the proposed model, a three-layer CNN was used. The seventh row model in Table 1 replaces the three-layer CNN with an average operation. It simply averages these outputs of the two branches. Because the CNN block combines the outputs more flexibly, it performs better than the average operation. Based on these components, we proposed the dual-constraint framework for burst image denoising.

Table 1 Ablation study on the synthetic dataset with the noise level $\sigma = 50$

Model	PSNR (dB)	
	Set12	BSD68
Single branch, noisy input	28.55	27.88
Single branch, noisy and pilot inputs	29.25	28.27
Two branches, noise constraint	29.00	28.16
Two branches, signal constraint	29.65	28.55
The proposed	**29.71**	**28.61**
Structure without long skip connection	29.19	28.18
Combination without three-layer CNN	29.09	27.10

Best results are in bold

4.3 Test on synthetic images

In the experiments, we compared the proposed method with several state-of-the-art methods, whose source codes are publicly available. We demonstrated the results in terms of quantitative analysis and subjective visual effect. We used two metrics for quantitative performance evaluation, PSNR and structural similarity (SSIM), which are recommended for image quality evaluation and are widely used in image denoising.

We conducted a set of experiments on three testing datasets to evaluate the performance of the proposed model. There were 68 gray images in the BSD68 dataset. Set12 contained 12 gray images.

Kodak24 had 24 color images. These images were commonly employed in image denoising tasks because they are abundant in texture. We compared the proposed approach to VBM3D, BM3D (Dabov et al., 2007), DnCNN (Zhang et al., 2017), and DVDNet (Tassano et al., 2019). We also made comparisons to VBM4D and FBID (Liu et al., 2014) for color image denoising. DnCNN was an efficient CNN-based image denoising algorithm. DnCNN-C was used for color image denoising. VBM3D, VBM4D, FBID, and DVDNet were designed for multi-frame image denoising.

Fig. 3 shows 12 images from the Set12 dataset. They have been widely used as the benchmark for gray image denoising. There are seven images of the size of 256×256 and five images of the size of 512×512. Table 2 shows the PSNR (dB) comparison on Set12, in which images are corrupted by AWGN with noise level $\sigma = 50$. This table shows the results recovered by the compared methods, BM3D, VBM3D, DnCNN, DnCNN-AVG, DVDNet, and the proposed approach. We point out that DnCNN-AVG was calculated using the mean value of recovered burst images that were processed by DnCNN individually. From the comparison, we can see that the proposed algorithm usually outperforms the compared methods in terms of PSNR. There is a significant improvement compared to the preprocessor BM3D. DVDNet also achieves a competitive result. Table 3 shows the SSIM comparison, and we can see the superiority of the proposed approach. Fig. 4 presents a visual comparison of two examples from the Set12 dataset. The first column shows the clean reference frame and a noisy observation corroded by AWGN with noise level $\sigma = 50$. The second to sixth columns show the images recovered by BM3D, VBM3D, DnCNN-AVG, DVDNet, and the proposed approach, respectively. Note that the proposed approach can better capture a visually clean result and preserve more details. DnCNN-AVG outputs the smoothest result but smooths out the details. DVDNet preserves more texture but with significant noise.

Fig. 3 Images of Set12 dataset

BSD68 consisted of 68 images of size 321×481 or 481×321. Table 4 reveals the PSNR result tested on two noise levels, $\sigma = 50$ and $\sigma = 25$. We calculated the maximum, minimum, and mean values of the 68 images for comparison. We can see that the proposed algorithm obtains the best results in most cases. The mean value is more reflective of performance, and on average, our approach has improved 2.9879 dB for noise level $\sigma = 50$ and 2.5895 dB for noise level $\sigma = 25$ compared to the preprocessor BM3D. We emphasize that our model is trained on noise level $\sigma = 50$ but still performs effectively on noise level $\sigma = 25$.

Set12 and BSD68 contained gray images only, and we used Kodak24 for color image testing. There were 24 color images of size 768×512 or 512×768.

Table 2 PSNR (dB) results of different methods on Set12 dataset with the noise level $\sigma = 50$

Image	PSNR (dB)						
	Original	BM3D	VBM3D	DnCNN	DnCNN-AVG	DVDNet	Ours
Cameraman	14.8565	26.1130	26.3541	27.0047	26.3295	26.6340	**29.2958**
House	14.5914	29.6380	30.4681	30.0136	31.2604	30.0157	**32.2391**
Pepper	14.7104	26.6244	27.8517	27.2926	27.6585	27.7167	**29.6813**
Fishstar	14.9202	25.0165	26.0405	25.7012	26.3724	26.5935	**28.7913**
Monarch	14.7085	25.7192	27.1462	26.7648	28.0151	27.7944	**29.6881**
Airplane	15.0593	25.1457	26.4944	25.8656	26.4403	26.6210	**28.5899**
Parrot	15.0288	25.9217	26.2420	26.4830	25.8877	26.3779	**28.7566**
Lena	14.6148	29.0460	30.2706	29.3604	30.2800	29.6825	**31.7309**
Barbara	14.7560	27.2219	27.2955	26.2301	26.9460	27.1064	**29.4681**
Ship	14.5878	26.7175	27.8773	27.1896	27.7527	27.6964	**29.5703**
Man	14.6386	26.8117	27.9689	27.2409	27.9361	27.7163	**29.5781**
Couple	14.5540	26.1038	27.5152	26.8928	27.5083	27.6010	**29.0776**

Best results are in bold

Table 3 SSIM results of compared methods on Set12 dataset with the noise level $\sigma = 50$

Image	SSIM						
	Original	BM3D	VBM3D	DnCNN	DnCNN-AVG	DVDNet	Ours
Cameraman	0.1844	0.7689	0.7691	0.7981	0.7758	0.7343	**0.8319**
House	0.1274	0.8153	0.8306	0.8227	0.8416	0.7593	**0.8454**
Pepper	0.1940	0.7928	0.8300	0.8113	0.8323	0.7689	**0.8610**
Fishstar	0.2431	0.7373	0.7824	0.7623	0.7919	0.7886	**0.8476**
Monarch	0.2545	0.8159	0.8585	0.8446	0.8783	0.8211	**0.8946**
Airplane	0.2214	0.7787	0.8245	0.8041	0.8121	0.7986	**0.8533**
Parrot	0.2123	0.7856	0.7931	0.8010	0.7800	0.7376	**0.8458**
Lena	0.1175	0.7995	0.8243	0.8121	0.8283	0.7554	**0.8475**
Barbara	0.2045	0.7935	0.7935	0.7692	0.7800	0.7688	**0.8621**
Ship	0.1605	0.7009	0.7348	0.7172	0.7317	0.7132	**0.7871**
Man	0.1474	0.7034	0.7447	0.7218	0.7390	0.7130	**0.7970**
Couple	0.1694	0.6929	0.7365	0.7234	0.7254	0.7254	**0.7923**

Best results are in bold

Table 4 Maximum, minimum, and mean PSNR (dB) results of different algorithms on the BSD68 dataset with the noise levels $\sigma = 50$ and $\sigma = 25$

Method	Maximum PSNR (dB)		Minimum PSNR (dB)		Mean PSNR (dB)	
	$\sigma = 50$	$\sigma = 25$	$\sigma = 50$	$\sigma = 25$	$\sigma = 50$	$\sigma = 25$
BM3D	33.3945	36.9632	20.1365	23.3751	25.6188	28.5686
VBM3D	33.9171	38.9064	21.4437	26.2466	26.0362	30.3723
DnCNN	34.6586	38.2792	21.0121	24.0781	26.2329	28.8919
DnCNN-AVG	**37.2091**	**39.9345**	22.5082	26.3654	27.8091	30.5834
DVDNet	32.4783	39.6269	22.3471	26.7741	26.2493	30.8836
Ours	36.1108	39.4885	**23.5319**	**26.8706**	**28.6067**	**31.1581**

Best results are in bold

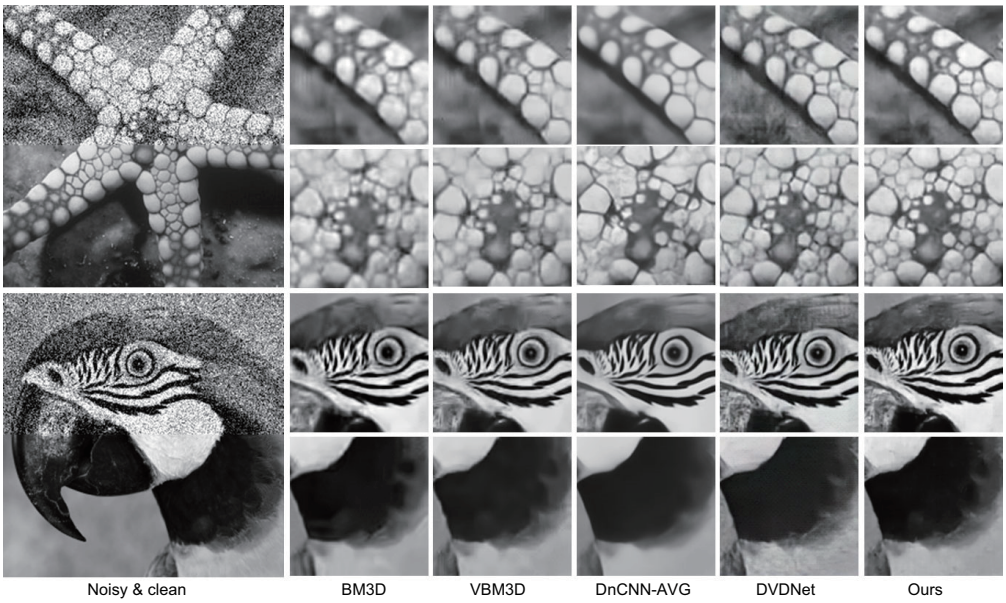

Fig. 4 Visual comparison of examples from Set12 dataset with the noise level $\sigma = 50$. The first column is stitched using the noisy and clean versions; the other columns show the results of BM3D, VBM3D, DnCNN-AVG, DVDNet, and the proposed method

Figs. 5 and 6 show the quantitative comparison; the images were corrupted by AWGN with noise level $\sigma = 50$. We experimented on CBM3D, VBM4D, DnCNN-C, DnCNN-AVG, FBID, and DVDNet. For clarity, we do not show the results of CBM3D and DnCNN-C, which are usually worse than those of VBM4D and DnCNN-AVG. DnCNN-C is derived from DnCNN and is applicable for color image denoising. Fig. 5 shows a comparison based on PSNR, Fig. 6 shows a comparison based on SSIM, and we can see that the proposed approach obtains better results in most examples. VBM4D generates a frustrating result for color image denoising in terms of PSNR. FBID and VBM4D always have the lowest SSIM values. Fig. 7 shows a visual comparison of an example from Kodak24. The clean image is corrupted by AWGN with noise level $\sigma = 50$. We enlarged two blocks for careful observation. From the noisy image, we cannot determine the shape of the clouds or recognize the letters, and the PSNR is 14.8142 dB. The image quality is increased dramatically using VBM4D, DVDNet, FBID, and the proposed approach. The result of FBID is much noisier. VBM4D result preserves the cloud shape well, but the noise is obvious. DVDNet and the proposed approach perform better for noise removal and texture preservation. We can observe that the proposed approach gives a clearer cloud shape. The PSNR of the proposed method is 29.5960 dB, which is higher than those of the others.

In addition, we compared frames with a larger frame alignment error. The frames were corrupted by AWGN with the noise level $\sigma = 50$. Different from existing experiments, the related frames were randomly shifted by a maximum of 10 pixels. As shown in Fig. 8, our approach performs better than the comparison approaches. The proposed approach provides clearer texture.

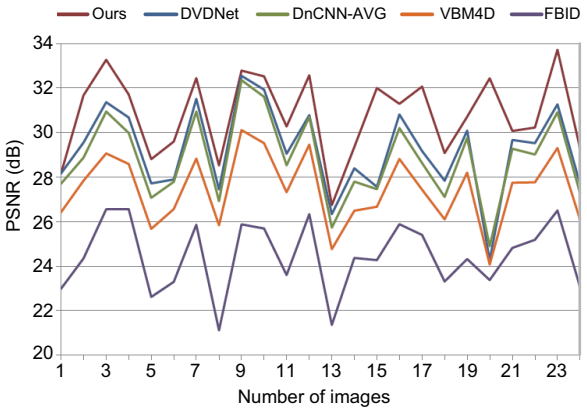

Fig. 5 PSNR (dB) results of different methods on the Kodak24 dataset with the noise level $\sigma = 50$ (References to color refer to the online version of this figure)

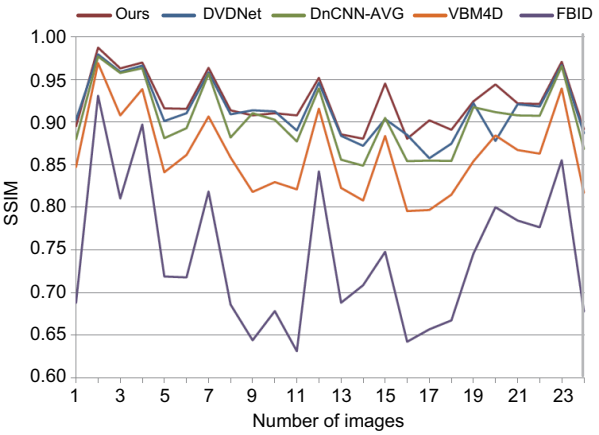

Fig. 6 SSIM results of compared algorithms on the Kodak24 dataset with the noise level $\sigma = 50$ (References to color refer to the online version of this figure)

4.4 Test on realistic images

The real noise distribution is similar but more complex than Gaussian. In this subsection, we presented the results of our experiment on burst images captured with automatic camera settings in a low-light environment. Fig. 9 shows the visual comparison of images captured by Honor V30 with ISO of 6400 and exposure time of 1/17 s. Fig. 9 displays the noisy image and the images denoised by FBID, DnCNN, VBM4D, DVDNet, and the proposed approach. Based on these experiments, all the listed methods demonstrated a certain ability in realistic burst image denoising. We enlarged three blocks for a detailed comparison; each block included a smooth area and texture information. DnCNN is not very successful in realistic image denoising, and there is still a lot of noise that should be removed. FBID and VBM4D have better results than DnCNN, but FBID and VBM4D still introduce artifact. The window frames, building edges, and bicycle axials are crooked. DVDNet and the proposed approach overcome those variants, preserve more texture information, and remove more noise. DVDNet creates a much smoother result and generates speckles in the flat area.

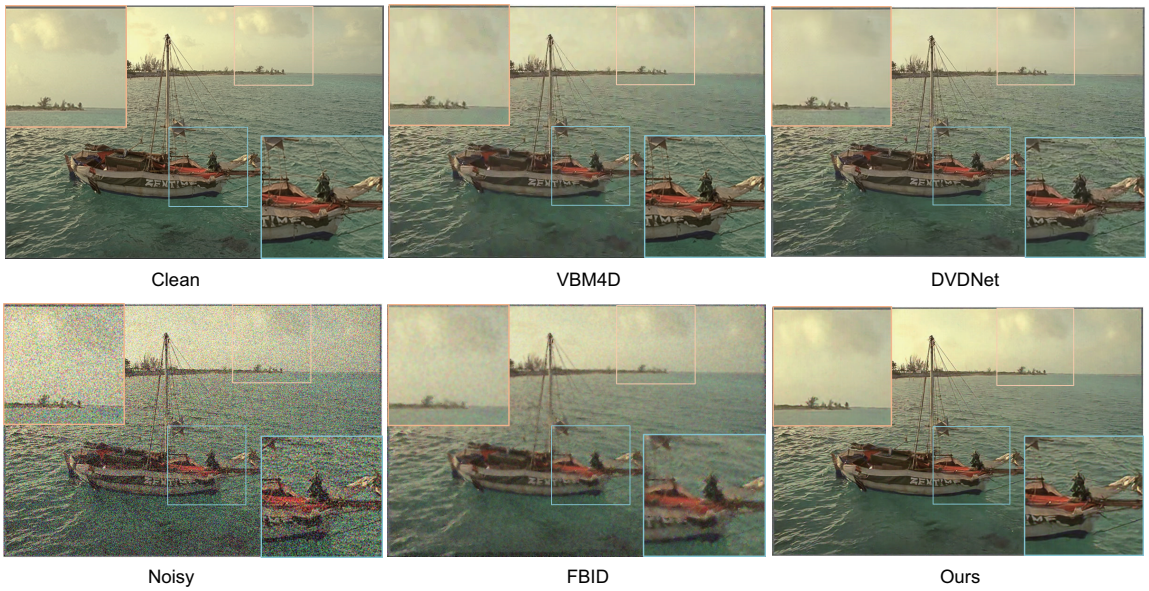

Fig. 7 Visual comparison of color image denoising (one example in the Kodak24 dataset)

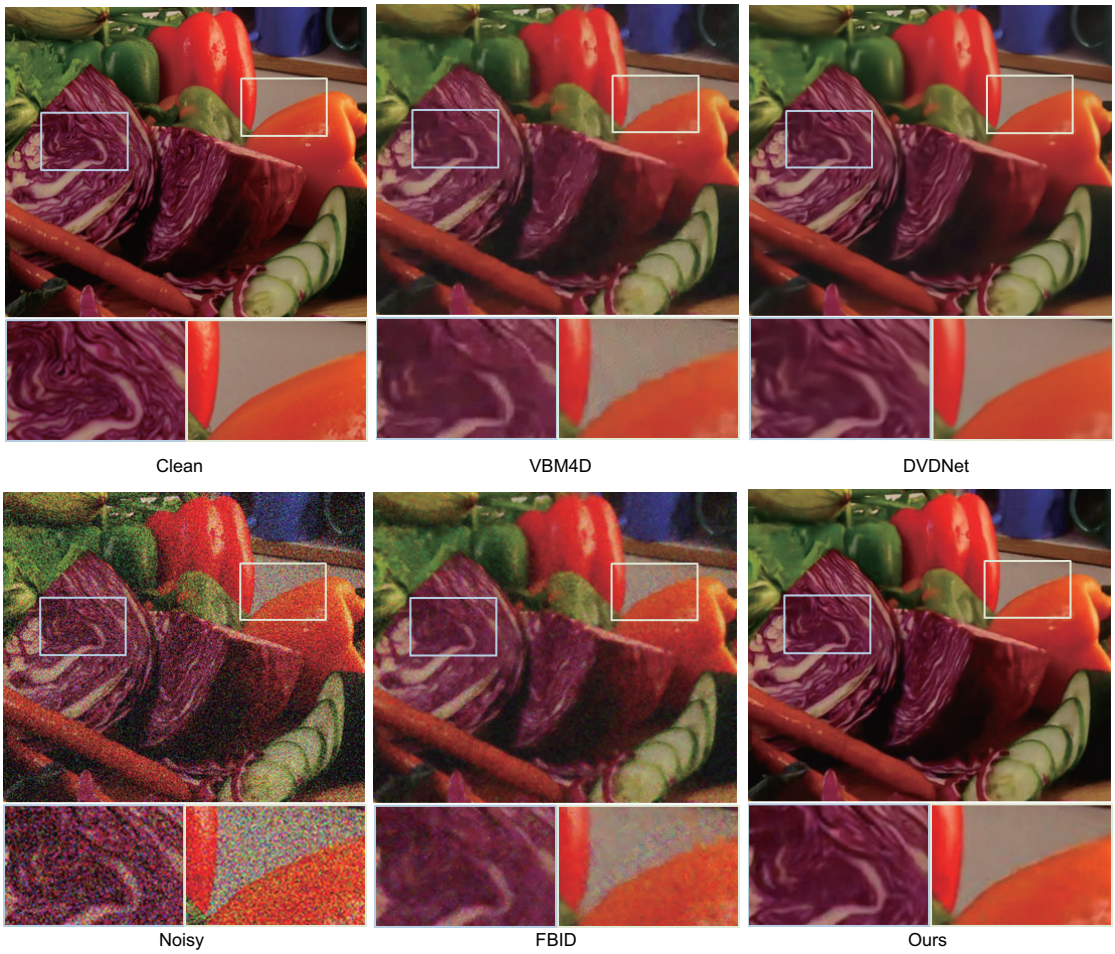

Fig. 8 Comparison of images with frame alignment errors

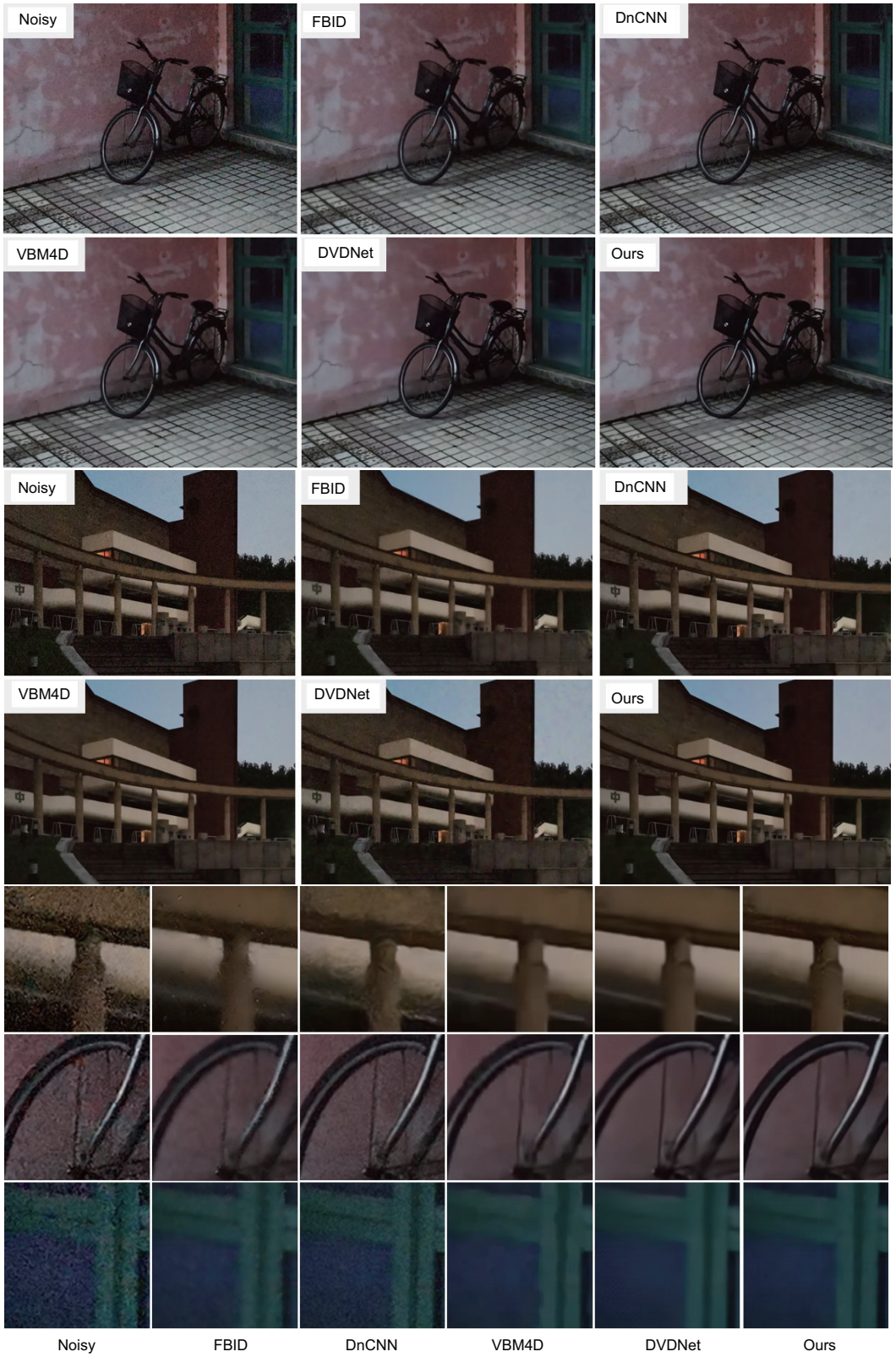

Fig. 9 Visual comparison of realistic images

5 Conclusions

In this study, we proposed a CNN-based approach for burst image denoising. We used a dual-path CNN to combine the power of BM3D and CNN for burst image denoising. We improved the denoising ability by a noise constraint and a signal constraint simultaneously. In addition, we adopted block matching in the network to resolve misalignment errors. In experiments on synthetic datasets and realistic low-light images, both quantitative analysis and qualitative indicators demonstrated the effectiveness of our method. The proposed result showed a better balance between noise removal and texture preservation.

Contributors

Dan ZHANG and Lei ZHAO designed the research. Dan ZHANG conducted the experiments and drafted the paper. Lei ZHAO helped organize the paper. Duanqing XU and Dongming LU revised and finalized the paper.

Compliance with ethics guidelines

Dan ZHANG, Lei ZHAO, Duanqing XU, and Dongming LU declare that they have no conflict of interest.

References

Aharon M, Elad M, Bruckstein A, 2006. K-SVD: an algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Trans Signal Process*, 54(11):4311-4322. https://doi.org/10.1109/TSP.2006.881199

Ahn B, Cho NI, 2017. Block-matching convolutional neural network for image denoising. https://arxiv.org/abs/1704.00524

Buades A, Coll B, Morel JM, 2005. A non-local algorithm for image denoising. IEEE Computer Society Conf on Computer Vision and Pattern Recognition, p.60-65. https://doi.org/10.1109/CVPR.2005.38

Burger HC, Schuler CJ, Harmeling S, 2012. Image denoising: can plain neural networks compete with BM3D? IEEE Conf on Computer Vision and Pattern Recognition, p.2392-2399. https://doi.org/10.1109/CVPR.2012.6247952

Chambolle A, 2004. An algorithm for total variation minimization and applications. *J Math Imag Vis*, 20(1-2):89-97. https://doi.org/10.1023/B:JMIV.0000011325.36760.1e

Dabov K, Foi A, Katkovnik V, et al., 2007. Image denoising by sparse 3-D transform-domain collaborative filtering. *IEEE Trans Image Process*, 16(8):2080-2095. https://doi.org/10.1109/TIP.2007.901238

Divakar N, Babu RV, 2017. Image denoising via CNNs: an adversarial approach. Proc IEEE Conf on Computer Vision and Pattern Recognition Workshops, p.1076-1083. https://doi.org/10.1109/CVPRW.2017.145

Godard C, Matzen K, Uyttendaele M, 2018. Deep burst denoising. Proc European Conf on Computer Vision, p.560-577. https://doi.org/10.1007/978-3-030-01267-0_33

Krull A, Buchholz TO, Jug F, 2019. Noise2Void—learning denoising from single noisy images. Proc IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.2124-2132. https://doi.org/10.1109/CVPR.2019.00223

LeCun Y, Bottou L, Bengio Y, et al., 1998. Gradient-based learning applied to document recognition. *Proc IEEE*, 86(11):2278-2324. https://doi.org/10.1109/5.726791

Lehtinen J, Munkberg J, Hasselgren J, et al., 2018. Noise2Noise: learning image restoration without clean data. https://arxiv.org/abs/1803.04189

Lempitsky V, Vedaldi A, Ulyanov D, 2018. Deep image prior. Proc IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.9446-9454. https://doi.org/10.1109/CVPR.2018.00984

Liu ZW, Yuan L, Tang XO, et al., 2014. Fast burst images denoising. *ACM Trans Graph*, 33(6):Article 232. https://doi.org/10.1145/2661229.2661277

Mao XJ, Shen CH, Yang YB, 2016. Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections. https://arxiv.org/abs/1603.09056v2

Mildenhall B, Barron JT, Chen JW, et al., 2018. Burst denoising with kernel prediction networks. Proc IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.2502-2510. https://doi.org/10.1109/CVPR.2018.00265

Mosseri I, Zontak M, Irani M, 2013. Combining the power of internal and external denoising. IEEE Int Conf on Computational Photography, p.1-9. https://doi.org/10.1109/ICCPhot.2013.6528298

Perona P, Malik J, 1990. Scale-space and edge detection using anisotropic diffusion. *IEEE Trans Patt Anal Mach Intell*, 12(7):629-639. https://doi.org/10.1109/34.56205

Simonyan K, Zisserman A, 2014. Very deep convolutional networks for large-scale image recognition. https://arxiv.org/abs/1409.1556v4

Tassano M, Delon J, Veit T, 2019. DVDNET: a fast network for deep video denoising. IEEE Int Conf on Image Processing, p.1805-1809. https://doi.org/10.1109/ICIP.2019.8803136

Tomasi C, Manduchi R, 1998. Bilateral filtering for gray and color images. Sixth Int Conf on Computer Vision, p.839-846. https://doi.org/10.1109/ICCV.1998.710815

Vincent P, Larochelle H, Lajoie I, et al., 2010. Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion. *J Mach Learn Res*, 11:3371-3408.

Xu J, Zhang L, Zuo WM, et al., 2015. Patch group based nonlocal self-similarity prior learning for image denoising. Proc IEEE Int Conf on Computer Vision, p.244-252. https://doi.org/10.1109/ICCV.2015.36

Yang D, Sun J, 2018. BM3D-Net: a convolutional neural network for transform-domain collaborative filtering. *IEEE Signal Process Lett*, 25(1):55-59. https://doi.org/10.1109/LSP.2017.2768660

Zhang K, Zuo WM, Chen YJ, et al., 2017. Beyond a Gaussian denoiser: residual learning of deep CNN for image denoising. *IEEE Trans Image Process*, 26(7):3142-3155. https://doi.org/10.1109/TIP.2017.2662206

Zhang K, Zuo WM, Zhang L, 2018. FFDNet: toward a fast and flexible solution for CNN-based image denoising. *IEEE Trans Image Process*, 27(9):4608-4622. https://doi.org/10.1109/TIP.2018.2839891