Journal of Signal Processing, Vol.28, No.1, pp.1-17, January 2024

PAPER

Joint Training of Noisy Image Patch and Impulse Response of Low-Pass Filter in CNN for Image Denoising

May Thet Tun, Yosuke Sugiura and Tetsuya Shimamura

Graduate School of Science and Engineering, Saitama University, 255 Shimo-Okubo, Sakura-ku, Saitama 338-8570, Japan
E-mail: may.t.t.218@ms.saitama-u.ac.jp, ysugiura@mail.saitama-u.ac.jp, shima@mail.saitama-u.ac.jp

Abstract **In this paper, we propose the sequential input of a noisy image patch and the impulse response of a low-pass filter (LPF) in the training of the conventional fast and flexible solution for CNN-based image denoising (FFDNet) architecture, which enhances denoising performance and edge preservation and achieves high perceptual quality. The proposed method consists of two steps. In the first step, the power spectrum sparsity is utilized to determine the impulse response of LPF and the resulting impulse response is added to the noisy image patch in a sequential form to estimate the low- and high-frequency components of the input image. In this step, the use of three different types of LPF is also considered. In the second step, the FFDNet architecture, a deep-learning-based image denoiser, is employed. The proposed method achieves satisfactory denoising performance for grayscale and color datasets on synthetic additive white Gaussian noise (AWGN) in terms of the peak signal-to-noise ratio (PSNR), structural similarity index (SSIM), feature similarity index (FSIM), and learned perceptual image patch similarity (LPIPS) compared with the original FFDNet. The performances on realistic noise and for chest X-ray images are also investigated.**

Keywords: image denoising, convolutional neural network, FFDNet, impulse response of low-pass filter, power spectrum sparsity

1. Introduction

Because of low light conditions, slow shutter speed effects, sensor noise of digital cameras, signal fluctuations, and other factors, several types of noise, including salt and pepper noise, Poisson noise, and additive white Gaussian noise (AWGN), corrupt images. The image degradation is mathematically modeled as $d(n_1, n_2) = o(n_1, n_2) + \eta(n_1, n_2)$, where $d(n_1, n_2)$, $o(n_1, n_2)$, and $\eta(n_1, n_2)$ denote the degraded image, the original clean image, and AWGN, respectively. The diagnosis of diseases is made more difficult by the noise included in low-quality medical images such as those of positron emission tomography (PET), magnetic resonance imaging (MRI), and computed tomography (CT). These medical images are mainly affected by the error in thermal energy fluctuations leading to AWGN. Therefore, most of the papers on denoising focus on removing AWGN because, in practice, it corrupts images more than other types of noise.

For decades, noise reduction has been the most extensive research issue in the field of image processing. Image pixels corrupted by noise can degrade the quality of the image owing to a large number of missing pixels. Recent advances in image denoising have focused on maximizing denoising quality while ignoring perceptual quality and edge preservation. The outcomes of these denoising efforts are plagued by image artifacts, blurry and smooth output texture, image edge and fine detail distortion, and poor perceptual quality of images, particularly at high noise levels. Therefore, in current research, attempts have been made to obtain effective criteria for higher perceptual quality and edge preservation through denoising-based solutions [1]. The advancement of image denoising is fueling the ability to provide diverse, high-quality results for inverse problems such as image inpainting and deblurring. Therefore, the two inverse problems of image inpainting and denoising are combined to solve the current image-denoising problem [2]. The most exciting issue for image denoisers nowadays is to overcome the perception-distortion trade-off while pursuing high perceptual quality [3]. The trade-off between perception and distortion has a significant impact on the low-level vision of image processing.

High-quality images are more important than noise reduction in medical imaging, such as tumor identification. Low-perception images, such as medical images, often appear blurry and are difficult to understand and interpret manually. Thus, unlike typical images, medical images have low contrast and are unclear and distorted. Thus, the recent work in image denoising research considers the high perceptual quality of the images by solving the problem of recovering the highly distorted image counterparts [4]. The denoised image encourages perceptually motivated methods to produce a visually pleasing image to overcome the challenges of disease diagnosis. Moreover, the mean square error (MSE) loss, which is pixel-by-pixel reconstruction loss and is widely used in image restoration, has a limitation in reconstructing high-quality images. It evaluates the pixel-wise distance between the denoised and ground truth images and penalizes larger values of weights while ignoring image luminance and color. Because of this, the denoised images become blurry and the high-frequency details are smoothed out. Thus, image-denoising research should concentrate on a perceptually motivated method to improve perceptual quality.

Image denoising methods can be categorized into spatial domain filtering, transform domain filtering, and deep learning methods. Spatial domain filtering methods such as mean filter, median filter, Wiener filter, and others are applied in image noise removal. These linear and nonlinear filterings are effective in removing certain types of noise. Their limitation is that they can only denoise a specific type of noise. The nonlocal means filter [5], one of the legendary filters in image denoising, denoises the noisy image using a weighted average of a similar group of pixels. However, it loses image pixel information and over-smooths the image's edges and textures. Block matching and 3D filtering (BM3D), one of the state-of-the-art denoising methods [6], denoises images using block-matching and collaborative filters with similar types of block. The BM3D is a sparse representation in the transform domain, and a collaborative hard threshold is applied. The BM3D is time-consuming and cannot preserve edges. The fundamental issue of previous image-denoising methods is that they are insufficient to preserve the fine details of the image, especially in high-noise environments.

The frequency domain is associated with pixel adjustments, and its filtering techniques handle enhanced edges and image suppression. The image is converted into the frequency domain by the Fourier transform, which modifies and rejects the unwanted frequency components and has a better ability to retain the edge and fine details. Suhaila and Shimamura [7] proposed a logarithmic power spectrum estimation method based on a block-based frequency domain Wiener filter (FDWF). The noise power spectra are estimated in low- and high-frequency regions based on a threshold, block by block. The FDWF not only removes AWGN based on low- and high-frequency components but also preserves the image's edges and fine details. In 2018, a method for image denoising based on a parametric spectral subtraction Wiener filter was proposed [8]. In this paper, the weighting and power parameters of the parametric Wiener filter (PWF) were determined by dividing the image into three groups, resulting in a faster computational time than the traditional Wiener filters.

Prior-based methods, such as BM3D, are time-consuming, have a complicated architecture, cannot remove all types of realistic noise, and are required for parameter selection. Therefore, deep learning methods, particularly convolutional neural networks (CNN), have recently attracted attention in image processing and computer vision. The residual approach in deep learning has proven to be effective in image restoration and denoising, and learning from the residual images yields faster, more accurate results. Zhang et al. proposed the denoising convolutional neural network (DnCNN), a residual image denoising approach, in 2017 [9]. The DnCNN trains the model by comparing noisy and clean image pairs, and it handles AWGN and JPEG image deblocking problems. However, it fails to recover high-frequency textures and edges and needs to train a specific model for a fixed noise level. Hence, they next proposed a fast and flexible solution for the CNN-based image denoising (FFDNet) model [10]. One of the strategies of the FFDNet is that it handles multiple noise levels with a single trained model, resulting in low computational complexity and high noise reduction performance. Recently, the FFDNet image denoiser has been widely used in remote sensing synthetic aperture radar (SAR) image change detection as well as channel estimation in multi-input and multi-output (MIMO) systems [11]. Nowadays, most of the papers on image denoising extend to two or more neural network layers to enhance the denoising results. However, because of the black-box structure of neural networks, it is difficult to obtain improved results when network layers are extended.

Image frequency domain filtering is widely used in image enhancement, restoration, and other applications. The high-pass filter (HPF) sharpens the fine details of image components, and the smoothing filter, also known as the low-pass filter (LPF), aids in noise removal. The denoising effect is achieved in the frequency domain with an

LPF by defining the cut-off frequency to filter out high-frequency noise, whereby edge information is retained. It is one of the popular techniques in image noise reduction and has a significant impact on the detection of image contours and contrast enhancement. The downsampling process, widely used in neural network image processing, improves computational performance and reduces the number of parameters. However, it causes aliasing and distorts the high-frequency components [12]. Blurring is a type of solution, and it helps with aliasing while preserving useful edge information. To tackle this problem, an LPF layer is added before CNN's down-sampling step in video recognition [12]. Dey et al. [13] proposed the addition of median filter layers to all feature channels in the CNN to eliminate the AWGN. In this study, the blurring operation of the median filter enhances the image noise removal performance.

The simultaneous training in a CNN with two input images improves the generalization ability [14][15] because the network can extract significant features from both input images and use them in unknown images. This approach enables the CNN to accurately capture the image's structure and fill in the missing pixel regions with the additional input image. Furthermore, it can recover corrupted pixel information of a degraded image via an additional input image to produce a high-quality image. In this study, the proposed method applies the impulse response of LPF as the additional input image to obtain prior knowledge of the image's low- and high-frequency components. The pixel value of the impulse response of LPF shares the image structure and enhances the spatial resolution of the degraded image. The proposed method sequentially inputs the noisy image patch and the impulse response of LPF in the training of conventional FFDNet architecture. It aids in the preservation of the image's fine details and makes the edges more prominent. By simultaneously providing the noisy image patch and the impulse response of LPF as inputs, the FFDNet can utilize more feature information in constructing its learning-based trained model.

The rest of the paper is structured as follows. In Sect. 2, we describe the proposed image denoising method. In Sect. 3, the experiments are described and datasets, training information, and comparison results are shown and discussed. Finally, the paper is concluded in Sect. 4.

2. Proposed Method

In this section, the architecture and implementation of the proposed method are explained in detail. Four types of impulse response of LPFs utilized in the proposed method are introduced for investigation and performance comparison.

Table 1 Specifications of the FFDNet and proposed method

	Grayscale	Color
Patch Size	70	50
Number of layers	15	12
Feature Maps	64	96

Table 2 Grayscale image denoising network architecture

Layer	Layer No.	Kernel	Stride,Padding	Activation
Conv	1	3×3	1,1	ReLU
Conv	2 to 14	3×3	1,1	BN+ReLU
Conv	15	3×3	1,1	–

2.1 Architecture and learning

Figure 1 illustrates the architectures of the conventional FFDNet and the proposed method.

For the FFDNet, images are first as elsewhere scaled to 1, 0.9, 0.8, and 0.7 with a stride of 20, and then divided into patch sizes of 70×70 for grayscale images and 50×50 for color images, as shown in Table 1. Next, these image patches are randomly augmented by horizontal and vertical flips up and down and counterclockwise rotations of 90, 180, and 270 degrees. The images are divided into patches, and the argumentation yields more training samples to speed up the learning process. After this process, the image patches are increased to 34400 grayscale and 167000 color samples.

Images are downsampled from W $\times$ $H \times C$ to W$/2 \times H/2 \times 4C$, where W and H are the column and row lengths of the image, respectively and C denotes the number of channels, where one is for a grayscale image and three for corresponding color images. After downsampling, these images are patched into the first layer of the neural network and then upsampled to the original size after training. This procedure is intended to accelerate the neural network training process. The spatially variant synthetic AWGN with a random variance between 0 and 75 is corrupted to obtain clean and noisy image patch pairs. There are three types of layers. Convolution and rectified linear units (ReLU) are adopted for the first layer. As shown in Fig. 1, batch normalization (BN) is added in the middle of the convolution, and ReLU layers for the middle layers and convolution for the final layer.

The combination of residual learning and batch normalization methods results in a faster training speed and improves noise removal. The middle layer of network depth is specified by 15 for grayscale images and 12 for color images, and convolutional

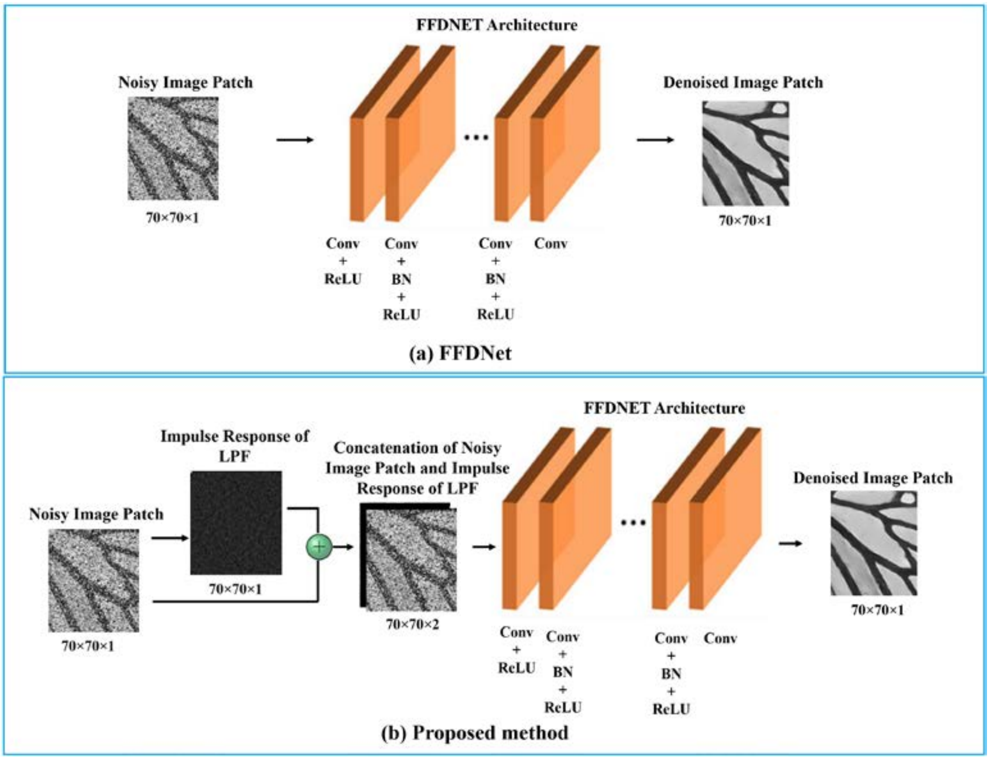


Fig. 1 Architectures of (a) conventional FFDNet and (b) proposed method

operations are constructed by kernels 3 × 3 in size throughout the network. After convolution, a stride of 1 is used to move one pixel, as shown in Table 2. Downsampling, or changing the image size, causes artifacts in the image; therefore, zero padding is applied in the network. The color image requires more channels than the grayscale image; thus, the feature map of each convolutional layer for grayscale images is implemented at 64 and at 96 for color images. The MSE loss function, L, is optimized as

$$L = \frac{1}{2K} \sum_{i=1}^{K} \left\| R\left(v_i, M; \emptyset\right) - u_i \right) \|^2 \qquad (1)$$

where K denotes the number of training image patches and the network learns a residual mapping function $R\left(v_i, M; \emptyset\right)$ from the noise level map M, where $\emptyset$ represents the model parameter that can vary the input noise level. v_i represents the noisy image patches, and u_i denotes the clean image patches. The feature of M in the FFDNet can effectively handle various levels of noise within a single network architecture. If M and the ground truth of the input noise level match, the best result will be obtained. If the predicted noise level is higher than the ground truth noise level of the image, the resulting denoised image will be oversmoothed; otherwise, it will be contaminated by noise artifacts.

Furthermore, the usage of the downsampling, MSE loss function, and residual learning approach in FFDNet can cause oversmoothed images. Image downsampling can result in the loss of critical information, and the MSE loss can lead to blurred images. In addition, the residual learning can produce undesired artifacts at high noise levels. To address this issue of FFDNet, we propose to utilize both the noisy image patch and the impulse response of LPF simultaneously in the FFDNet, which will result in the enhancement of denoising performance and preservation of the image details. The proposed idea has not been investigated as far as we know.

The proposed method differs from the FFDNet in that the additional impulse response of LPF is utilized in the FFDNet, as shown in Fig. 1. The FFDNet takes five features as input, including the noise level map M and downsampled sub-image. Therefore, there exist differences in input feature dimensions between FFDNet and the proposed method, as shown in Table 3. The training process of the FFDNet and the proposed method adjust to minimize the MSE loss between the clean and denoised image patches. This ensures that the size of the denoised image patch matches the clean image patch. Thus, the output features of FFDNet and the proposed approach are identical, as shown in Table 3, and the proposed method generates a denoised image patch, as shown in Fig. 1. For the proposed method, v_i in (1) represents the concatenation of the noisy image patch and the impulse response of LPF.

Table 3 Feature differences between FFDNet and proposed method

	FFDNet		Proposed	
	Grayscale	Color	Grayscale	Color
Input features	5	15	10	30
Output features	4	12	4	12

The proposed method is motivated as follows: the FFDNet [10] removes spatially variant AWGN more effectively than other denoisers. In addition to yielding better-denoised output results, the FFDNet smoothes edges and fine details, especially at high noise levels. The reason is that the FFDNet emphasizes and learns low-frequency components and does not preserve the high-frequency components. Utilizing the impulse response of LPF as an additional input image patch can mitigate the oversmoothing result in FFDNet, and the obvious differences in the frequency domain between clean and noisy image patches become visible in high-frequency regions. Thus, we propose the use of the impulse response of LPF. For the LPF, an LPF designed on the basis of the power spectrum sparsity [8] is derived. Also, three typical types of LPF are additionally considered.

2.2 Design of power spectrum sparsity-based LPF

In this subsection, an LPF designed on the basis of the power spectrum sparsity [8] is derived, which is used in the LPF part in Fig. 1(b). We refer to this LPF as the power spectrum sparsity-based LPF (SLPF) in this paper. First, each of the image patches is transformed into the frequency domain as $FFT[d_p(n_1,n_2)]$, where $d_p(n_1,n_2)$ means an image patch of $d(n_1,n_2)$ (hereafter, the subscript p is used to indicate that it is a patch) and $FFT[.]$ denotes the fast Fourier transform. The size of $FFT[.]$ is set to that of the image patch. The Fourier transform step involves the frequency shift operation as $D_p(\omega_1,\omega_2)=FFTSHIFT[d_p(n_1,n_2)]$, where ω_1 and ω_2 respectively correspond to the horizontal and vertical angular frequencies, which relocate the low-frequency components to the center and the high-frequency components to the corners. Then, we obtain the power spectrum of the image patch as $E_p(\omega_1,\omega_2)=|D_p(\omega_1,\omega_2)|^2$ [7].

Power spectrum sparsity, S_p, [8] indicates image frequency components for each image patch. A significant proportion of the frequency components is largely focused on the horizontal and vertical regions, as illustrated in Fig. 2. A low S_p value indicates that the image frequency components are concentrated in the horizontal and vertical regions [8]. The power spectrum sparsity S_p is formulated by

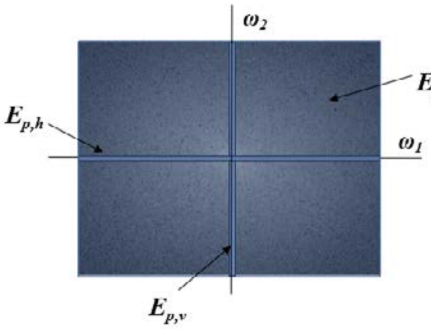


Fig. 2 Power spectrum sparsity of image patch

$$S_p = \frac{E_{p,e}}{E_{p,h}+E_{p,v}} \tag{2}$$

where $E_{p,e}$ denotes the sum of the entire power spectrum of the image patch, and $E_{p,h}$ and $E_{p,v}$ denote the total sums of the center's horizontal and vertical power spectrum values of the image patch, respectively. The SLPF employs the S_p value in the selection of the threshold used to estimate the frequency components to be passed through. The sparsity threshold value, θ, is calculated as

$$\theta \;=\; \alpha \; S_p \tag{3}$$

where α is a scaling parameter. The S_p value can vary, being either less than one or greater than one. A result less than one indicates that the image frequency components are more concentrated in the center's horizontal and vertical regions, whereas a result greater than one suggests that the image frequency components in the center's horizontal and vertical regions are less dominant or less concentrated compared with the overall power spectrum. To optimize the θ value for the threshold selection, we multiply the S_p value, being small, by the scaling parameter α as in (3).

α is the row (or column) length of the image patch and is slightly adjusted from the original size. The length is adjusted in accordance with the percentage of the image patch size. For example, when 80% is considered, the original length is changed to the original length multiplied by 0.8. (When the image patch size is 70 by 70, for the 80% case, α results in 70 × 0.8 = 56). In this paper, percentages from 80% to 130% were considered. The best setting for the percentage obtained from preliminary experiments was utilized in the practical implementation of (3). The example of the percentage and PSNR scores of denoised images obtained by the proposed method are shown in Table 4, where β denotes the percentage in decimal units.

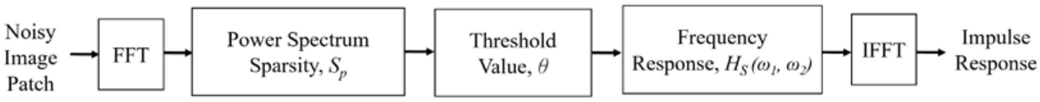

Noisy Image Patch
FFT
Power Spectrum Sparsity, S_p
Threshold Value, θ
Frequency Response, $H_S(\omega_1, \omega_2)$
IFFT
Impulse Response

Fig. 3 Block diagram of calculating the impulse response of SLPF

Table 4 Percentage and PSNR scores for the BSD68 and CBSD68 datasets (σ=25 case)

Dataset	β					
	0.8	0.9	1	1.1	1.2	1.3
BSD68	28.23	28.46	**26.48**	28.43	28.39	28.37
CBSD68	30.44	**30.47**	30.40	30.37	30.33	30.27

When we denote the standard deviation of AWGN as σ, $\sigma = 25$, is generally considered to correspond to a moderate noise level and is known to be effective for a wide range of noise levels, resulting in a widely recognized benchmark for image denoising evaluation [9]. Therefore, the setting has been utilized as the benchmark noise level for evaluating the PSNR scores in Table 4. In Table 4, the β value of 1 yields the greatest PSNR for the BSD68 dataset, while the β value of 0.9 yields the highest PSNR for the CBSD68 dataset. These two datasets BSD68 and CBSD68 have an extensive collection of testing images compared with others. From this overview, the proposed method consistently employed a value of 1 for β in the case of grayscale images and a value of 0.9 for color images.

Then, the center of the image, which corresponds to the low-frequency region containing the majority of the information of the image patch, is assigned to one (which is passed through) on the basis of the θ value. Otherwise, zero is assigned (which is cut out). Thus, the frequency response of the SLPF, $H_S(\omega_1, \omega_2)$, is specified by

$$H_S(\omega_1, \omega_2) = \begin{cases} 1, & \text{if } E_p(\omega_1, \omega_2) \leq \theta \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

After that, the frequency axes are shifted back from the center to the four corners. Finally, the impulse response of the LPF is calculated as $IFFT[H_S(\omega_1, \omega_2)]$, where $IFFT[.]$ denotes the inverse fast Fourier transform. Figure 3 summarizes the above process in a block diagram.

2.3 Implementation of comparative LPFs

To demonstrate the effectiveness of the proposed method, the impulse response of the SLPF described in Sect. 2.2 is compared with those of the direct LPF (DLPF), Gaussian LPF (GLPF), and Butterworth LPF (BLPF). By comparing different impulse responses of LPFs, we can obtain the following information. First, we can choose the best suitable impulse response of LPF for specific types of noise pattern. Second, we can evaluate the trade-off between noise reduction and image detail preservation and select the most effective process in accordance with the image characteristics. Third, we can highlight the strength and limitations of the impulse response of the SLPF, and further research can be conducted using the comparative results. The following steps are used to create the impulse response of each LPF

1. Specify the size of the LPF in accordance with the noisy image patch.
2. Choose the frequency response of LPF.

   For the DLPF,

   $$H_D(\omega_1, \omega_2) = \begin{cases} 1, & \sqrt{(\omega_1^2 + \omega_2^2)} \leqslant w_c \\ 0, & \sqrt{(\omega_1^2 + \omega_2^2)} > \omega_c \end{cases} \tag{5}$$

   For the GLPF,

   $$H_G(\omega_1, \omega_2) = e^{-\left(\omega_1^2 + \omega_2^2\right)/2\omega_c^2} \tag{6}$$

   For the BLPF,

   $$H_B(\omega_1, \omega_2) = \frac{1}{1 + \left[\sqrt{(\omega_1^2 + \omega_2^2)}/\omega_c\right]^{2n}} \tag{7}$$

   where ω_c is the cut-off frequency of the LPF and n denotes the order of the filter. For findings on the impulse response of BLPF, the highest PSNR result was obtained with the first order (n=1).
3. Determine ω_c as $\pi/2$.
4. Calculate the impulse response by $IFFT[.]$ for each case.
5. Finally, both the noisy image patch and the impulse response of LPF are sequentially input into the FFDNet.

3. Experimental Results

The denoising performance of the proposed method is investigated on synthetic AWGN and realistic noise as well as for chest X-ray images. For the proposed method, the four LPF versions described in Sects. 2.2 and 2.3 are considered. First, datasets and the details of parameter settings for the neural network are introduced. Then, the qualitative comparisons between the proposed method and the FFDNet are presented.

The 180×180 image size with 400 images from [16], the grayscale training dataset for DnCNN, and FFDNet as the grayscale training method, and the 256×256 image size with 500 images from the BSD500 dataset [19] were used for color image training. Set12 and Berkeley Segmentation Dataset (BSD68) [19] are two commonly used testing datasets in grayscale image denoising, whereas the RNI6 dataset [19] is used to remove realistic noise. The Set12 dataset consists of the most widely used images from the Standard Image Data-BASE (SIDBA) [20], such as Lenna, Cameraman, House, Peppers, and Barbara. The BSD68 dataset has 68 images with an image size of 321×481. The Kodak24 dataset [19], 24 centered cropped natural images, and the CBSD68 dataset [19], the color image version of the BSD68 dataset, are also adopted for color image denoising. The RNI15 dataset [19] consists of 15 realistic noisy images, such as low-light images from smartphones, old photographs, and aerial images, and was used to evaluate the ability of color-realistic image denoising.

The proposed model was implemented with Pytorch, and experiments were carried out on an Intel ® Core ™ i9-10900K CPU 3.70 GHz, 32 GB of RAM, and NVIDIA Quadro P2200 GPU with CUDA version 11.7. The ADAM optimizer was utilized for learning and the MSE loss function was applied. The weights were initialized by Kaiming_normal to the convolutional filters, and the learning rate was 0.001. The batch size was 128 and the epoch was set to 80. The proposed method and the baseline network, FFDNet, had the same architecture and parameter settings.

3.1 Image quality evaluation metric

Image quality assessment (IQA) yields a measure of the image quality performance. The most widely used IQA metrics in image denoising research may be MSE, PSNR, and structural similarity index (SSIM).

PSNR and MSE indicate the absolute errors between two images, and SSIM is the perception of image quality in terms of brightness, contrast, and structure. SSIM indicates the structural similarity between the original and reconstructed images on the basis of the hypothesis of the Human Visual System (HVS). A good IQA metric is the similarity of the low-level features between the original and distorted images. The feature similarity index (FSIM) is a measure of the saliency of low-level features. The two types of low-level feature based on the IQA of FSIM are phase congruency and gradient magnitude. Fourier waves at different frequency components in an image generate phase congruency, which reveals useful frequency features. The second feature, the gradient magnitude, is the contrast computed using the Sobel, Prewitt, and Scharr gradient operators in both horizontal and vertical directions [17]. In the learned perceptual image patch similarity (LPIPS) evaluation[18], a deep-neural-network-based IQA, the perceptual quality of the denoised image patch is measured. LPIPS is determined through a weighted summation of the distance across channels and a calculation of the perception distance between two image patches using pretrained weights of three well-known neural networks such as SqueezeNet, AlexNet, and VGG. The findings in [18] demonstrate that VGG outperforms SqueezeNet and AlexNet because it has the largest number of training parameters. Hence, the proposed method was evaluated by VGG for LPIPS. The lower the LPIPS value, the more perceptually similar the denoised image is to the original image. Zhang et.al [18] claimed that PSNR and SSIM fail to capture the human perception of similarity. Therefore, in this paper, PSNR, SSIM, FSIM [17], and LPIPS [18] are evaluated to visualize the effectiveness of image quality for denoising.

3.2 Experiments for AWGN removal

Tables 5-8 show the denoising performances of the proposed idea and the impulse responses of the SLPF, DLPF, GLPF, and BLPF+FFDNet, and the conventional FFDNet where each image is corrupted by synthetic AWGN in terms of PSNR, SSIM, FSIM, and LPIPS. The best results are denoted in bold, and the second-best results are underlined. According to the PSNR, SSIM, FSIM, and LPIPS overall performances, the impulse responses of the SLPF, GLPF, and BLPF+FFDNet indicate better performance than the FFDNet not only in grayscale but also in color image AWGN removal.

The average PSNR results of the proposed method, the sequential input of the noisy image patch, and the impulse response of the SLPF in the FFDNet are approximately 0.1 dB in grayscale and 0.2 dB in color images higher than those of FFDNet. The average SSIM and FSIM are approximately 0.01 and 0.005 higher than those of FFDNet, respectively. Additionally, the average LPIPS score exhibits about a 0.03 decrease compared with that of FFDNet. The improvement in SSIM and LPIPS indicates that the proposed method better preserves low-quality image information such as luminance, color, and contrast. The increment in FSIM demonstrates the ability to recognize high-frequency details and resolves the issue of preserving information in high-noise environments, excluding the noise level of 75 in the BSD68 dataset. As a result, the SLPF +FFDNet provides a satisfactory perceptual texture result while better retaining the high-frequency edges and fine details than the FFDNet does.

The GLPF+FFDNet can effectively remove AWGN and enhance the perceptual quality of images. As shown in Tables 5 and 6, the GLPF+FFDNet achieves higher PSNR and SSIM at the noise levels of 15 and 25 in the BSD68 dataset; however, its FSIM is lower than that of the impulse response of the SLPF+FFDNet, which is implemented by the frequency-domain-based power spectrum approach. The SSIM results of the impulse response of the SLPF+FFDNet are better than those of the DLPF, GLPF, and BLPF+FFDNet for the high noise levels of 35, 50, and 75 in the BSD68 dataset, as shown in Table 5. According to the FSIM of the Set12 dataset, the impulse response of the SLPF+FFDNet preserves image edges better than the other methods do. Thus, the impulse response of the SLPF+FFDNet retains the high-frequency details in the grayscale dataset of Set12 and BSD68 datasets.

As shown in Tables 7 and 8, the impulse response of the SLPF, DLPF, GLPF, and BLPF+FFDNet outperforms the FFDNet in terms of PSNR, SSIM, FSIM, and LPIPS for color AWGN removal. The PSNR and SSIM of the impulse response of the SLPF+FFDNet are slightly better than those of the GLPF+FFDNet in the Kodak24 dataset, as shown in Table 7. The impulse response of the SLPF+FFDNet achieves a comparable PSNR result to that of the GLPF+FFDNet in the CBSD68 dataset. However, the SSIM, FSIM, and LPIPS results are slightly lower than those in the GLPF+FFDNet case, as shown in Table 8, because the GLPF+FFDNet performs better in low- and high-frequency splitting in the color images. Overall, the proposed idea enhances denoising results and achieves superior perceptual quality and edge preservation than does the FFDNet.

The visual comparisons of the FFDNet and four types of the proposed method with a high noise level (σ=50), as shown in Figs. 4-7, revealed the following. The FFDNet smooths out the edges of the "$Starfish$" image and the scarf and trousers in the "$Barbara$" image, but the four types of the proposed method retain some of the high-frequency edges in grayscale images, as visually depicted in Figs. 4 and 5. The FFDNet in the "$Kodim$17" image has blurry results at the nasolabial fold lines in the face texture. In contrast, the four types of the proposed method result in better nasolabial fold line edges than does the FFDNet, as shown in Fig. 6. In "$Kodim$19" in Fig. 7, the signboard information on the fence disappears entirely when denoised by the FFDNet. However, the SLPF+FFDNet preserves the signboard information. Furthermore, the DLPF, GLPF, and BLPF+FFDNet display marginal preservation of the signboard. Therefore, the four types of the proposed method outperform the FFDNet in terms of controlling the trade-off between noise removal and edge preservation.

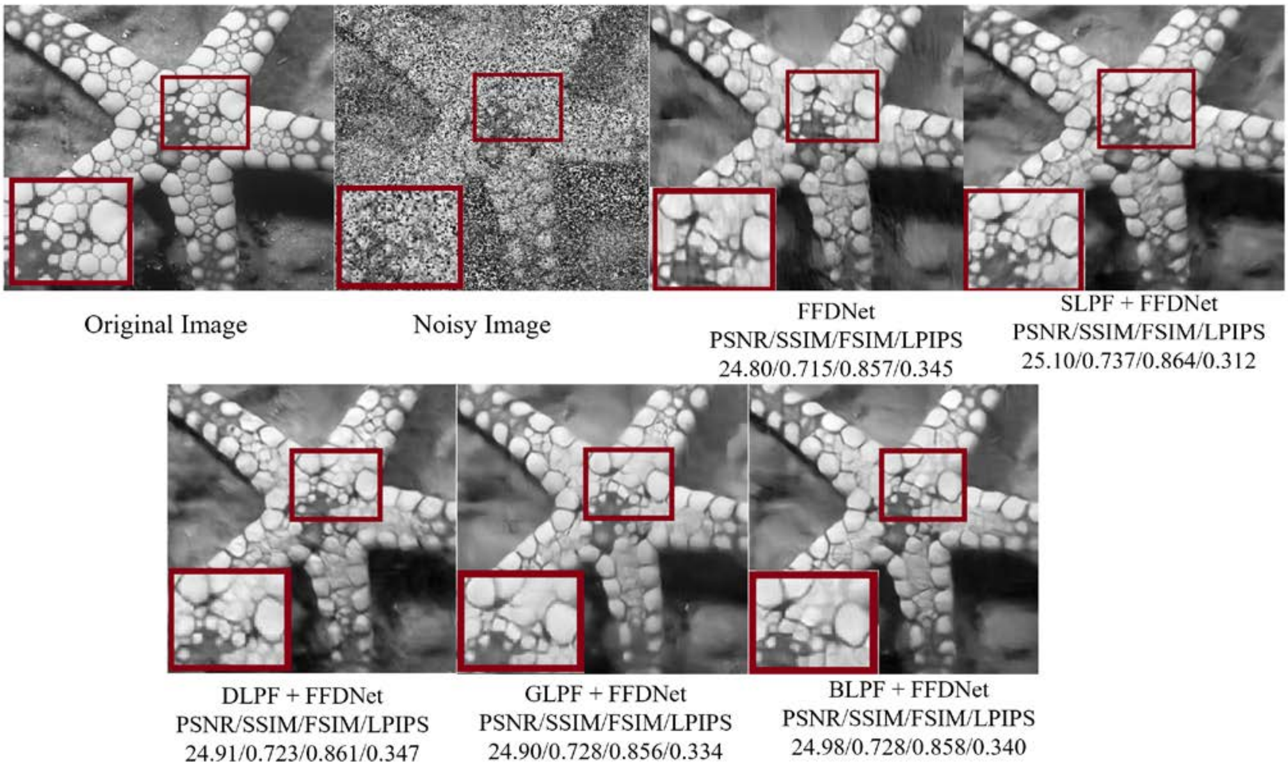
Table 5 Average quantitative results (PSNR/SSIM/FSIM/LPIPS) for the BSD68 dataset [Bold: Best, Underline: Second Best]

σ	Method	PSNR↑	SSIM↑	FSIM↑	LPIPS↓
15	FFDNet	30.59	0.8502	0.9188	0.2675
	SLPF+FFDNet	30.68	0.8529	0.9241	0.2415
	DLPF+FFDNet	30.60	0.8528	**0.9272**	0.2492
	GLPF+FFDNet	**30.83**	**0.8625**	0.9171	**0.2132**
	BLPF+FFDNet	30.79	0.8582	0.9120	0.2451
25	FFDNet	28.34	0.7833	0.8816	0.3329
	SLPF+FFDNet	28.48	0.7912	0.8876	0.3082
	DLPF+FFDNet	28.37	0.7835	**0.8877**	0.3266
	GLPF+FFDNet	28.52	**0.7950**	0.8756	**0.2944**
	BLPF+FFDNet	**28.53**	0.7937	0.8716	0.3125
35	FFDNet	26.96	0.7303	0.8535	0.3783
	SLPF+FFDNet	27.10	**0.7416**	**0.8591**	0.3577
	DLPF+FFDNet	26.96	0.7265	0.8561	0.3822
	GLPF+FFDNet	27.10	0.7404	0.8437	**0.3543**
	BLPF+FFDNet	**27.13**	0.7403	0.8419	0.3648
50	FFDNet	25.53	0.6637	0.8225	0.4269
	SLPF+FFDNet	25.65	**0.6770**	**0.8244**	**0.4187**
	DLPF+FFDNet	25.53	0.6586	0.8206	0.4372
	GLPF+FFDNet	25.64	0.6705	0.8060	0.4239
	BLPF+FFDNet	**25.71**	0.6741	0.8099	0.4244
75	FFDNet	23.93	0.5711	**0.7851**	**0.4783**
	SLPF+FFDNet	**24.03**	**0.5870**	0.7822	0.4895
	DLPF+FFDNet	23.85	0.5643	0.7817	0.4950
	GLPF+FFDNet	23.95	0.5688	0.7618	0.4982
	BLPF+FFDNet	24.01	0.5653	0.7736	0.4951

3.3 Experiments for realistic noise removal

Realistic noisy images are more challenging to denoise because they are affected by illumination, sensor noise, camera shaking, lossy image compression, and other factors. The proposed method performs well not only in AWGN removal but also in realistic noise removal. For the realistic noisy images in Figs. 8, 9, and 10, the mapping noise sigma for denoising is manually set as σ=25 on the RNI6 and RNI15 datasets. The performance evaluation cannot be performed because there are no ground truth images for the RNI6 and RNI15 realistic noisy images.

The "$Building$" image in Fig. 8, the first image



Original Image

Noisy Image

FFDNet
PSNR/SSIM/FSIM/LPIPS
24.80/0.715/0.857/0.345

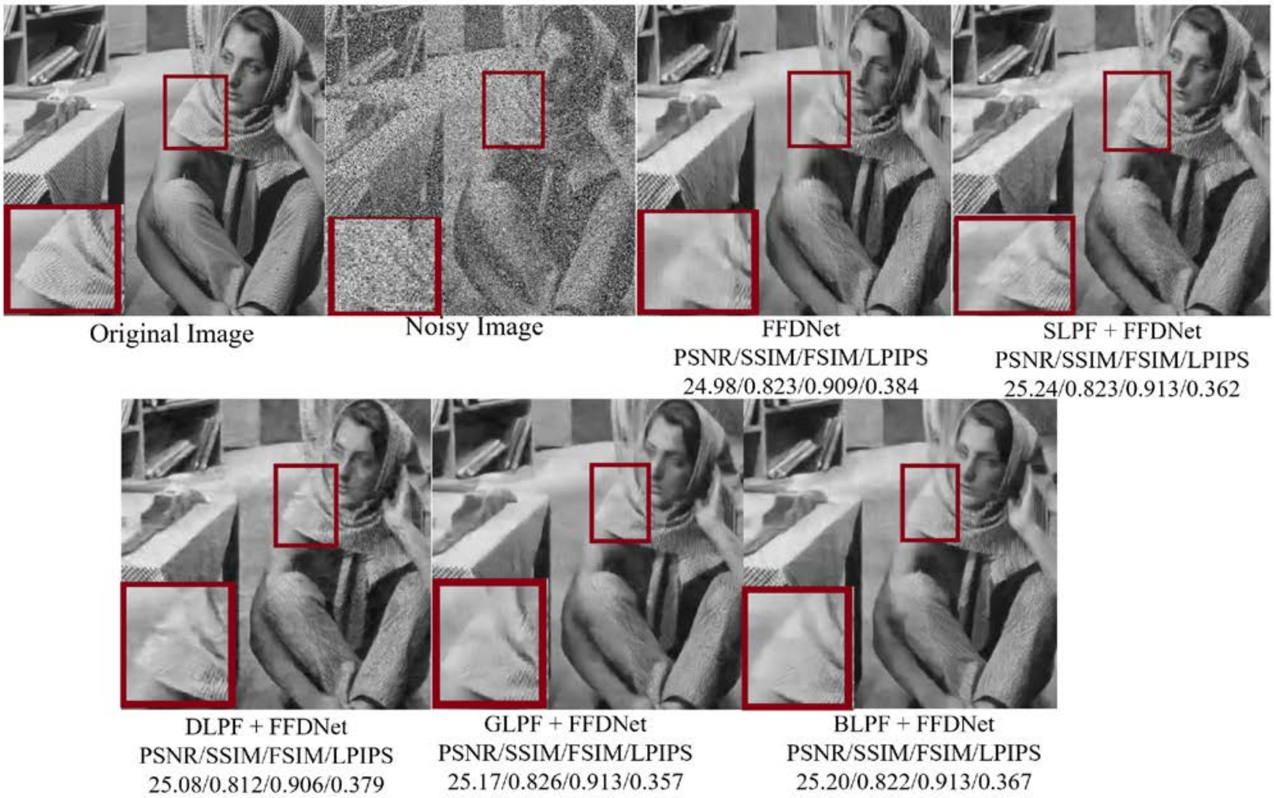
SLPF + FFDNet
PSNR/SSIM/FSIM/LPIPS
25.10/0.737/0.864/0.312

DLPF + FFDNet
PSNR/SSIM/FSIM/LPIPS
24.91/0.723/0.861/0.347

GLPF + FFDNet
PSNR/SSIM/FSIM/LPIPS
24.90/0.728/0.856/0.334

BLPF + FFDNet
PSNR/SSIM/FSIM/LPIPS
24.98/0.728/0.858/0.340

Fig. 4 “*Starfish*” images on $\sigma = 50$ for visual comparisons



Original Image

Noisy Image

FFDNet
PSNR/SSIM/FSIM/LPIPS
24.98/0.823/0.909/0.384

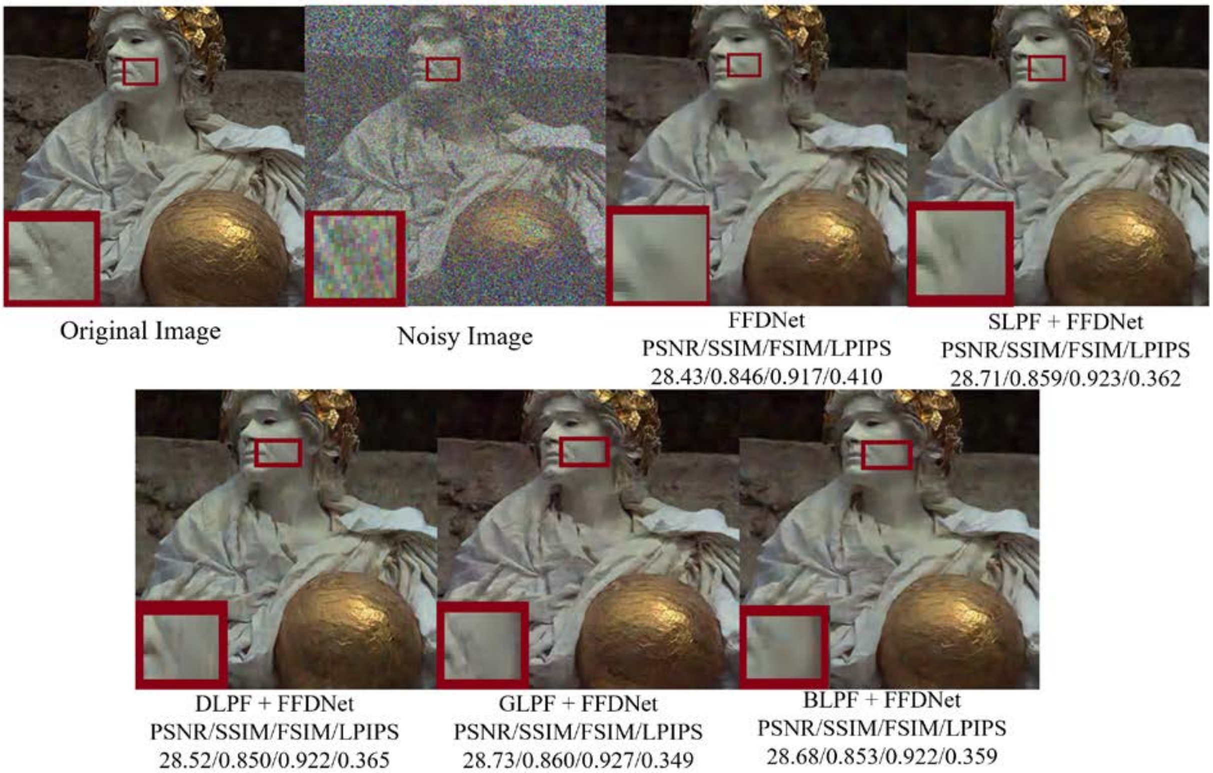
SLPF + FFDNet
PSNR/SSIM/FSIM/LPIPS
25.24/0.823/0.913/0.362

DLPF + FFDNet
PSNR/SSIM/FSIM/LPIPS
25.08/0.812/0.906/0.379

GLPF + FFDNet
PSNR/SSIM/FSIM/LPIPS
25.17/0.826/0.913/0.357

BLPF + FFDNet
PSNR/SSIM/FSIM/LPIPS
25.20/0.822/0.913/0.367

Fig. 5 “*Barbara*” images on $\sigma = 50$ for visual comparisons



Original Image

Noisy Image

FFDNet
PSNR/SSIM/FSIM/LPIPS
28.43/0.846/0.917/0.410

SLPF + FFDNet
PSNR/SSIM/FSIM/LPIPS
28.71/0.859/0.923/0.362

DLPF + FFDNet
PSNR/SSIM/FSIM/LPIPS
28.52/0.850/0.922/0.365

GLPF + FFDNet
PSNR/SSIM/FSIM/LPIPS
28.73/0.860/0.927/0.349

BLPF + FFDNet
PSNR/SSIM/FSIM/LPIPS
28.68/0.853/0.922/0.359

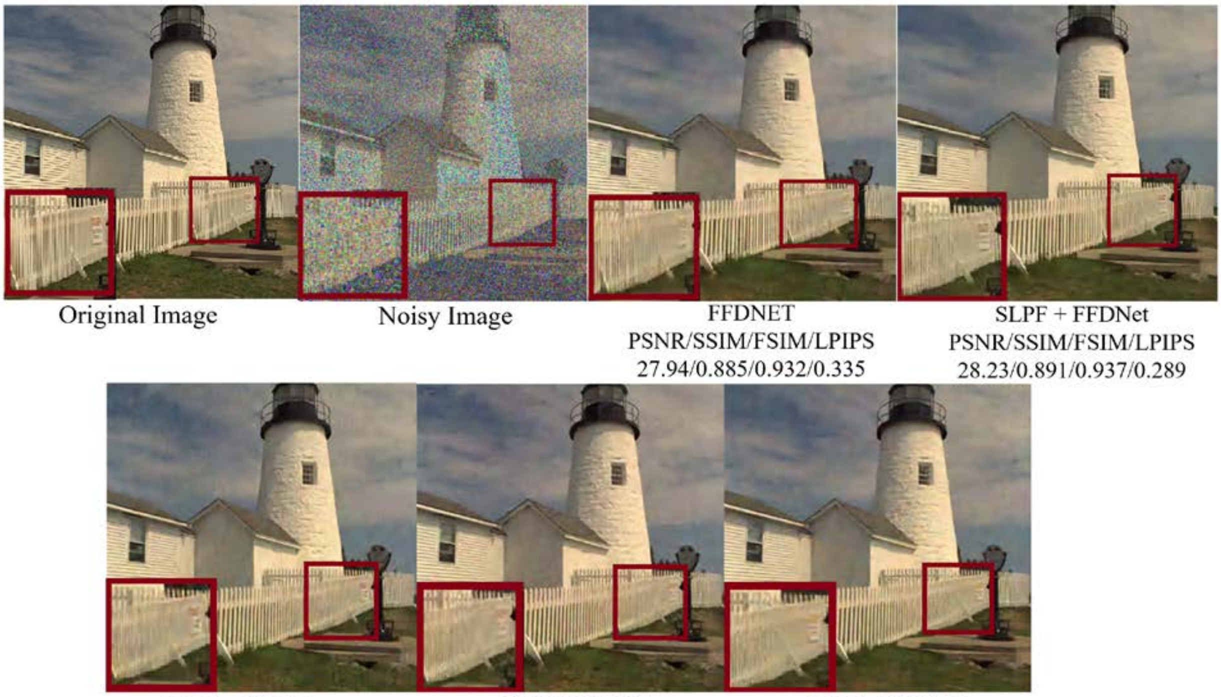
Fig. 6 "*Kodim*17" images on σ= 50 for visual comparisons

Table 6 Average quantitative results (PSNR/SSIM/FSIM/LPIPS) for the Set12 dataset [Bold: Best, Underline: Second Best]

σ	Method	PSNR↑	SSIM↑	FSIM↑	LPIPS↓
15	FFDNet	31.56	0.9034	0.9471	0.2229
	SLPF+FFDNet	31.62	0.9047	**0.9516**	0.2120
	DLPF+FFDNet	31.51	0.9006	0.9512	0.2157
	GLPF+FFDNet	**31.78**	**0.9114**	0.9478	**0.1996**
	BLPF+FFDNet	31.68	0.9075	0.9443	0.2179
25	FFDNet	29.38	0.8620	0.9228	0.2711
	SLPF+FFDNet	29.47	0.8664	**0.9278**	0.2631
	DLPF+FFDNet	29.31	0.8575	0.9262	0.2717
	GLPF+FFDNet	**29.52**	**0.8714**	0.9232	**0.2453**
	BLPF+FFDNet	29.50	0.8697	0.9201	0.2560
35	FFDNet	27.92	0.8264	0.9031	0.3074
	SLPF+FFDNet	27.97	0.8306	**0.9068**	0.3033
	DLPF+FFDNet	27.85	0.8201	0.9057	0.3146
	GLPF+FFDNet	28.02	**0.8355**	0.9031	**0.2874**
	BLPF+FFDNet	**28.03**	0.8341	0.9008	0.2922
50	FFDNet	26.29	0.7754	0.8772	0.3559
	SLPF+FFDNet	26.39	0.7822	**0.8812**	0.3554
	DLPF+FFDNet	26.19	0.7650	0.8779	0.3689
	GLPF+FFDNet	26.35	0.7827	0.8754	**0.3447**
	BLPF+FFDNet	**26.43**	**0.7832**	0.8761	0.3467
75	FFDNet	24.34	0.6857	0.8389	0.4266
	SLPF+FFDNet	**24.39**	**0.6994**	**0.8414**	0.4270
	DLPF+FFDNet	24.21	0.6750	0.8370	0.4387
	GLPF+FFDNet	24.32	0.6939	0.8355	**0.4220**
	BLPF+FFDNet	24.35	0.6869	0.8389	0.4281

Table 7 Average quantitative results (PSNR/SSIM/FSIM/LPIPS) for the Kodak24 dataset [Bold: Best, Underline: Second Best]

σ	Method	PSNR↑	SSIM↑	FSIM↑	LPIPS↓
15	FFDNet	33.19	0.9553	0.9798	0.2075
	SLPF+FFDNet	**33.45**	**0.9582**	**0.9810**	**0.1505**
	DLPF+FFDNet	33.30	0.9554	0.9804	0.1593
	GLPF+FFDNet	33.44	0.9569	0.9806	0.1554
	BLPF+FFDNet	**33.45**	0.9559	0.9804	0.1668
25	FFDNet	30.96	0.9243	0.9624	0.2715
	SLPF+FFDNet	**31.20**	**0.9299**	**0.9650**	**0.2174**
	DLPF+FFDNet	31.05	0.9247	0.9642	0.2283
	GLPF+FFDNet	31.18	0.9277	**0.9650**	0.2188
	BLPF+FFDNet	31.15	0.9256	0.9641	0.2321
35	FFDNet	29.51	0.8942	0.9462	0.3228
	SLPF+FFDNet	**29.74**	**0.9026**	0.9493	0.2739
	DLPF+FFDNet	29.61	0.8955	0.9489	0.2820
	GLPF+FFDNet	29.70	0.8995	**0.9498**	**0.2711**
	BLPF+FFDNet	29.68	0.8959	0.9483	0.2854
50	FFDNet	27.98	0.8522	0.9238	0.3810
	SLPF+FFDNet	**28.23**	**0.8632**	0.9279	0.3392
	DLPF+FFDNet	28.05	0.8527	0.9276	0.3441
	GLPF+FFDNet	28.14	0.8596	**0.9287**	**0.3378**
	BLPF+FFDNet	28.14	0.8549	0.9266	0.3463
75	FFDNet	26.15	0.7843	0.8911	0.4509
	SLPF+FFDNet	**26.43**	**0.8014**	**0.8965**	0.4214
	DLPF+FFDNet	26.33	0.7892	0.8963	0.4200
	GLPF+FFDNet	26.35	0.7967	0.8961	**0.4162**
	BLPF+FFDNet	26.35	0.7919	0.8938	0.4253



Original Image

Noisy Image

FFDNET
PSNR/SSIM/FSIM/LPIPS
27.94/0.885/0.932/0.335

SLPF + FFDNet
PSNR/SSIM/FSIM/LPIPS
28.23/0.891/0.937/0.289

DLPF + FFDNet
PSNR/SSIM/FSIM/LPIPS
27.96/0.882/0.936/0.304

GLPF + FFDNet
PSNR/SSIM/FSIM/LPIPS
28.06/0.889/0.937/0.284

BLPF + FFDNet
PSNR/SSIM/FSIM/LPIPS
28.07/0.884/0.936/0.300

Fig. 7 "*Kodim*19" images on σ= 50 for visual comparisons

Table 8 Average quantitative results (PSNR/SSIM/FSIM/LPIPS) for the CBSD68 dataset [Bold: Best, Underline: Second Best]

σ	Method	PSNR↑	SSIM↑	FSIM↑	LPIPS↓
15	FFDNet	32.73	0.8998	0.9531	0.1910
	SLPF+FFDNet	32.93	0.9068	**0.9573**	0.1408
	DLPF+FFDNet	32.79	0.9060	0.9563	0.1401
	GLPF+FFDNet	32.90	0.9089	0.9560	0.1350
	BLPF+FFDNet	**32.99**	**0.9107**	0.9556	**0.1341**
25	FFDNet	30.29	0.8464	0.9219	0.2566
	SLPF+FFDNet	**30.48**	0.8568	**0.9295**	0.2069
	DLPF+FFDNet	30.35	0.8553	0.9281	0.2100
	GLPF+FFDNet	30.45	0.8589	0.9282	**0.1992**
	BLPF+FFDNet	30.47	**0.8592**	0.9254	0.2034
35	FFDNet	28.76	0.7986	0.8943	0.3077
	SLPF+FFDNet	**28.94**	0.8116	0.9043	0.2627
	DLPF+FFDNet	28.84	0.8105	0.9038	0.2636
	GLPF+FFDNet	28.91	**0.8155**	**0.9047**	**0.2515**
	BLPF+FFDNet	28.91	0.8143	0.8992	0.2589
50	FFDNet	27.17	0.7350	0.8601	0.3692
	SLPF+FFDNet	**27.38**	0.7533	0.8728	0.3272
	DLPF+FFDNet	27.24	0.7490	0.8733	0.3285
	GLPF+FFDNet	27.34	**0.7593**	**0.8747**	**0.3177**
	BLPF+FFDNet	27.34	0.7564	0.8662	0.3255
75	FFDNet	25.36	0.6395	0.8178	0.4399
	SLPF+FFDNet	**25.61**	0.6648	0.8313	0.4100
	DLPF+FFDNet	25.55	0.6706	0.8334	0.4040
	GLPF+FFDNet	25.58	**0.6768**	**0.8354**	**0.3984**
	BLPF+FFDNet	25.56	0.6729	0.8218	0.4097

in the RNI6 dataset, is distorted by structured noise. In the RNI15 dataset, "*Audrey Hepburn*", the first image in Fig. 9, and the "*Movie*" image in Fig. 10, exhibit distortions due to JPEG lossy compression and video noise, respectively. The denoised results of grayscale and color-realistic images are shown in Figs. 8, 9, and 10 and illustrate that the BM3D is not capable of accurately removing all types of realistic noise and cannot retain high-frequency details. This is obvious from "*Building*", "*David Hilbert*", "*Old Tom Morris*", "*Vinegar*", "*Dog*", and "*Window*" realistic images. However, the visual results shown in Figs. 8, 9, and 10 show that the four types of the proposed method remove various types of realistic noise without artifacts and preserve the edges and textures more than the BM3D does. The proposed method has a better balance of noise removal, edge preservation, and the perceptual quality of the image than does the BM3D. This is more noticeable in the fur in the "*Bears*" image and the noise from the wooden window frame in the "*Window*" image. "*Frog*" and "*Dog*" images show that the proposed method is more effective in balancing the elimination of noise with the preservation of illumination, contrast, and edges than is the BM3D, as shown in Fig. 9.

In the visual comparisons of the images in Figs. 8 and 9, it is observed that the four types of the proposed method produce similar visual results. In the comparison between the SLPF+FFDNet and the FFDNet, it is evident that the SLPF+FFDNet achieves superior denoising performance in the images of "*David Hilbert*", "*Old Tom Morris*", "*Vinegar*" and "*Window*". Hence, the FFDNet has limited denoising capabilities, leading to less precise results. It is obvious that the FFDNet smoothes the line edges in Fig. 10; however, the SLPF+FFDNet performs better than others by preserving the color line edges. The SLPF+FFDNet can remove strong realistic noise in the curtain while preserving the contrast and edges of the lamp's shadow. However, it is challenging to precisely evaluate the performances because of the absence of corresponding clean reference images in the RNI6 and RNI15 datasets.

3.4 Experiments for chest X-ray images

Medical image noise removal is a difficult procedure because the subtle structures of medical images are distorted by noise and they are difficult to precisely identify and remove. In this paper, the experiments for medical image noise removal are also conducted on the JSRT (Japanese Society of Radiological Technology) [21] database, which is utilized for a variety of research purposes including image denoising, classification, segmentation, regression, and super-resolution. The image size is commonly 256×256. The JSRT consists of 197 training and 50 testing images. The images are divided into patch sizes of 70×70 and corrupted by synthetic AWGN with random noise variance in the range from 0 to 75.

Table 9 demonstrates that the impulse responses of the SLPF and BLPF+FFDNet make them the most effective image denoisers for JSRT chest X-ray images. The SLPF+FFDNet not only denoises in the high-noise environment of noise levels of 35, 50, and 75 for chest X-ray images but also effectively preserves the subtle structure information as the FFDNet does. According to Table 9, the BLPF+FFDNet provides the best PSNR, especially at high noise levels of 35, 50, and 75, and the SLPF+FFDNet achieves higher SSIM and LPIPS than others, indicating that the denoised images have perceptually satisfying results. Consequently, the SLPF+FFDNet sufficiently overcomes the problem of degradation in image quality caused by noise.

A visual comparison of the denoised images by the FFDNet and the SLPF+FFDNet shown in Fig. 11 is conducted, where the images were corrupted by AWGN with noise levels of 15, 25, 35, 50, and 75. The visual comparison proved that the FFDNet is inappropriate for denoising chest X-ray images because the FFDNet discards the critical stubble medical image structures in chest X-ray images. However, in the chest X-ray images denoised by the SLPF+FFDNet, the subtle image information is preserved because the impulse response of LPF shares the image details and edges.

Table 9 Average quantitative results (PSNR/SSIM/FSIM/LPIPS) for the JSRT database [Bold: Best, Underline: Second Best]

σ	Method	PSNR↑	SSIM↑	FSIM↑	LPIPS↓
15	FFDNet	36.97	0.9365	0.9621	0.1899
	SLPF+FFDNet	**37.10**	**0.9381**	0.9609	**0.1871**
	DLPF+FFDNet	35.88	0.8993	**0.9614**	0.2468
	GLPF+FFDNet	36.38	0.9239	**0.9614**	0.2558
	BLPF+FFDNet	36.67	0.9306	0.9604	0.2199
25	FFDNet	34.79	0.9118	0.9473	0.2626
	SLPF+FFDNet	**34.98**	**0.9174**	0.9481	**0.2383**
	DLPF+FFDNet	34.35	0.8837	0.9483	0.2569
	GLPF+FFDNet	34.77	0.9129	**0.9494**	0.2788
	BLPF+FFDNet	34.91	0.9125	0.9480	0.2725
35	FFDNet	33.21	0.8881	0.9353	0.3187
	SLPF+FFDNet	33.51	**0.8991**	0.9376	**0.2789**
	DLPF+FFDNet	33.10	0.8679	0.9384	0.2866
	GLPF+FFDNet	33.39	0.8983	**0.9389**	0.3096
	BLPF+FFDNet	**33.57**	0.8942	0.9385	0.3153
50	FFDNet	31.39	0.8553	0.9196	0.3891
	SLPF+FFDNet	31.76	**0.8703**	0.9237	**0.3425**
	DLPF+FFDNet	31.60	0.8399	0.9257	0.3608
	GLPF+FFDNet	31.63	0.8700	0.9239	0.3596
	BLPF+FFDNet	**31.97**	0.8654	**0.9262**	0.3690
75	FFDNet	29.17	0.7882	0.8947	0.4736
	SLPF+FFDNet	29.67	**0.8202**	0.9021	0.4349
	DLPF+FFDNet	29.45	0.7720	0.9041	0.4600
	GLPF+FFDNet	28.69	0.7508	0.8896	0.4795
	BLPF+FFDNet	**29.94**	0.8126	**0.9075**	**0.4285**

3.5 Comprehensive quantitative comparison

In Tables 5, 6, 7, and 8, results obtained with the four types of the proposed method and the FFDNet are listed for different datasets: Set12, BSD68, Kodak24, and CBSD68. In Table 9, results of the four types of the proposed method and the FFDNet are given for the JSRT dataset. With these results, we set out to make a comprehensive quantitative comparison. Each score in each table is averaged and the result is shown in Table 10. In Table 10, we see that the use of SLPF in the proposed method achieves a performance superior to the others. This suggests that the use of SLPF in the proposed method is the best option for removing AWGN.

3.6 Execution time

We measured the execution times for "*Barbara*" (512×512), "*Man*" (1024×1024), "*Lenna*" (512×512), and "*Stockton*" (1024×1024) with the SIDBA [20]. Table 11 shows the execution times in seconds for various denoising methods: the BM3D, FFDNet, and four types of the proposed method.

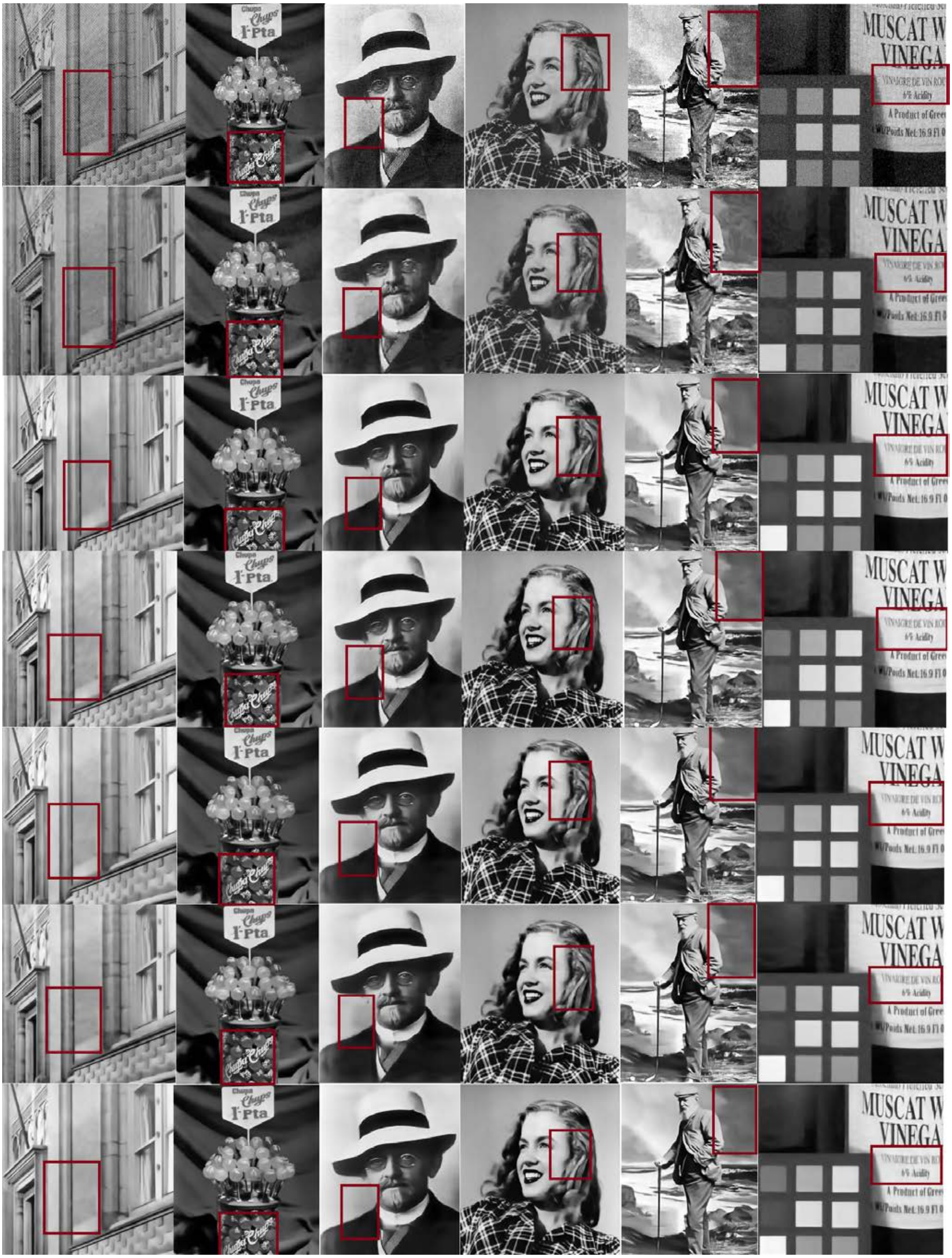


Fig. 8 Results of grayscale realistic noise removal for "*Building*", "*Chupa Chups*", "*David Hilbert*", "*Marilyn*", "*Old Tom Morris*", and "*Vinegar*" in the RNI6 dataset [From top to bottom: Realistic noisy image, Noise removal by BM3D, Noise removal by FFDNet, Noise removal by SLPF+FFDNet, Noise removal by DLPF+FFDNet, Noise removal by GLPF+FFDNet, and Noise removal by BLPF+FFDNet]

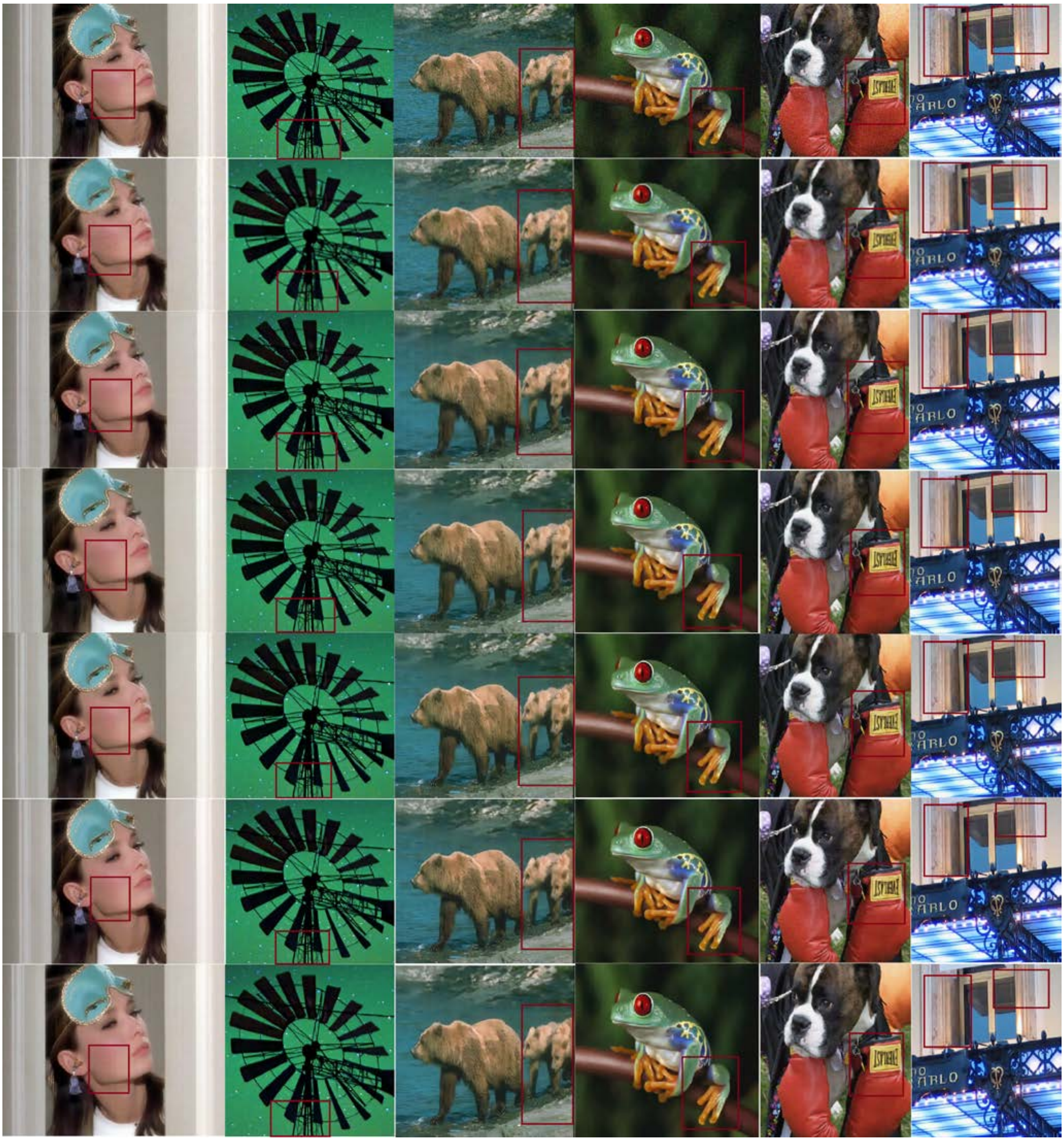


Fig. 9 Results of color realistic noise removal for “*Audrey Hepburn*”, “*Stars*”, “*Bears*”, “*Frog*”, “*Dog*”, and “*Window*” in the RNI15 dataset [From top to bottom: Realistic noisy image, Noise removal by BM3D, Noise removal by FFDNet, Noise removal by SLPF+FFDNet, Noise removal by DLPF+FFDNet, Noise removal by GLPF+FFDNet, and Noise removal by BLPF+FFDNet]

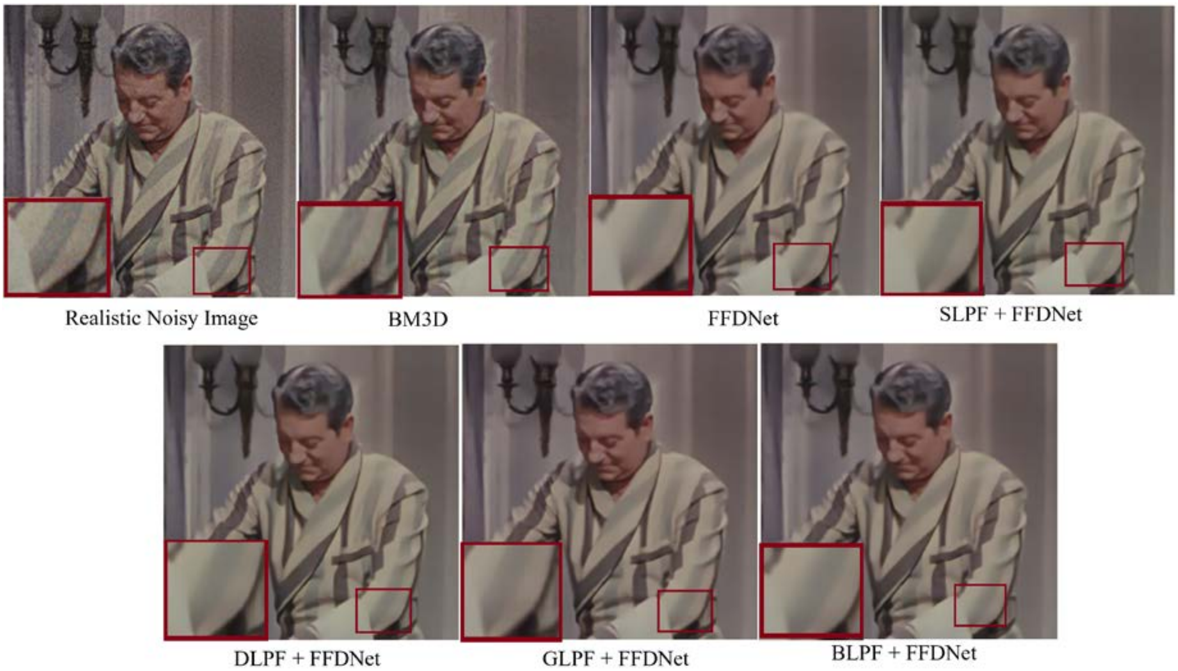


Fig. 10 “*Movie*” images from RNI15 dataset for visual comparisons

Table 10 Average quantitative results (PSNR/SSIM/FSIM/LPIPS) for the BSD68, Set12, Kodak24, CBSD68, and JSRT datasets [Bold: Best, Underline: Second Best]

Method	PSNR↑	SSIM↑	FSIM↑	LPIPS↓
FFDNet	29.30	0.8144	0.9024	0.3320
SLPF+FFDNet	**29.49**	**0.8251**	**0.9069**	**0.3043**
DLPF+FFDNet	29.27	0.8110	0.9064	0.3156
GLPF+FFDNet	29.40	0.8219	0.9029	0.3067
BLPF+FFDNet	**29.49**	0.8226	0.9021	0.3110

Single-threaded (ST) and multiple-threaded (MT) CPUs were considered. The results in Table 11 show the FFDNet and four types of the proposed method to be competitive, and they are also competitive to the BM3D with the image size of 512×512. However, the FFDNet and the four types of the proposed method clearly outperform the BM3D with the image size of 1024×1024.

4. Conclusion and Further Extension

In this paper, we proposed a novel approach for image denoising utilizing the sequential input of the noisy image patch and the impulse response of LPF into the FFDNet, the baseline method. This approach is unique and has not been explored in previous research on image denoising and FFDNet variants. The prior knowledge of the low and high-frequency components, which is reflected in the impulse response of the LPF to be designated, enhances the learning-based approach of FFDNet. Subsequently, it produces denoised images with improved edge preservation and perceptual quality. The experimental results demonstrated that the proposed method achieves an improvement over the FFDNet in terms of PSNR, SSIM, FSIM, and LPIPS.

The effectiveness of the impulse response of the SLPF was empirically compared and evaluated with three impulse responses of LPFs: DLPF, GLPF, and BLPF. To summarize, the impulse response of the SLPF is suitable for reducing grayscale AWGN while preserving the fine details of the images in accordance with the results of FSIM. The GLPF also indicates effective denoising results and provides excellent removal of color AWGN noise. In the color AWGN case, the SLPF suppresses the high-frequency noise components that cause color distortion. The smoothing effect of the GLPF offers a balance between noise removal and preservation of edges. The DLPF ranks as the second-best method for preserving the structural information in both grayscale and color AWGN removal.

The BLPF achieved the highest PSNR values in medical image noise removal while preserving the subtle structures of the image. In addition, the impulse response of the SLPF was also suitable for medical image noise removal, as it yielded the best SSIM and LPIPS results among the filters. It was concluded on the basis of the impulse response, that the SLPF is the best option for removing the AWGN.

The proposed method performed well in various

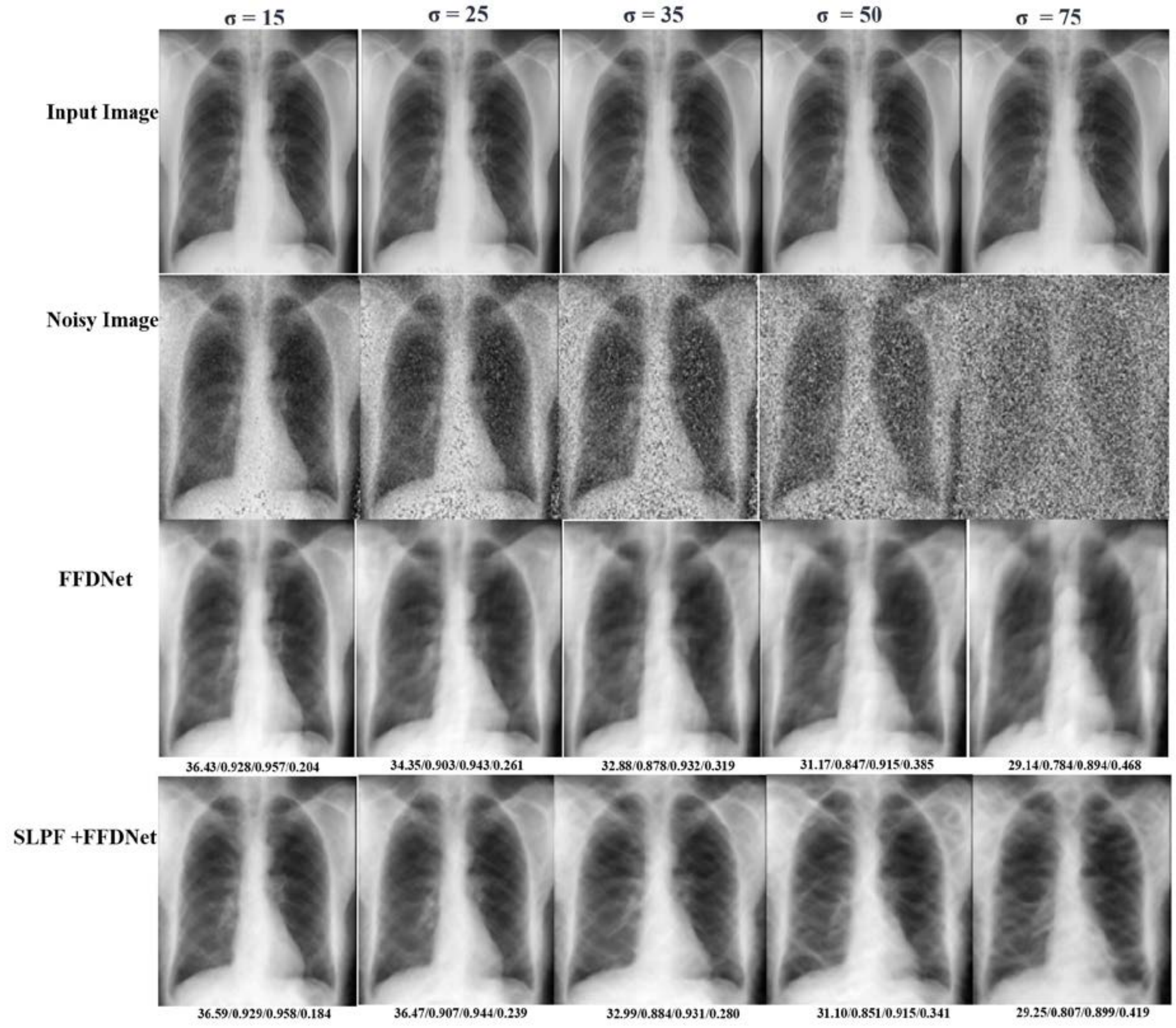


Fig. 11 “$JPCNN054$” images from JSRT dataset for visual comparisons

types of realistic noise, such as JPEG lossy compression noise, movie noise, and structured noise. The proposed method had a faster execution time than the BM3D for a large-size image and could denoise various types of realistic noise. The visual results confirmed that the proposed method deals with the trade-off between denoising performance and edge and texture preservation while maintaining the high perceptual quality of images.

In the future, we will extend this work to remove mixed noise as well as deal with low light conditions and blurriness in realistic noisy images.

Table 11 Execution time in seconds

Method	Device	512×512		1024×1024	
		Gray	Color	Gray	Color
BM3D	CPU(ST)	2.3	3.6	10.3	18.6
FFDNet	CPU(ST)	2.8	3.2	3.8	4.4
	CPU(MT)	0.7	0.7	0.8	0.9
SLPF+FFDNet	CPU(ST)	2.8	3.2	3.8	4.4
	CPU(MT)	0.7	0.7	0.8	0.9
DLPF,GLPF,BLPF + FFDNet	CPU(ST)	2.8	3.2	3.8	4.4
	CPU(MT)	0.7	0.7	0.8	0.9

References

[1] M. Cai, H. Zhang, H. Huang, Q. Geng, Y. Li and G. Huang: Frequency domain image translation: More photo-realistic, better identity-preserving, Proc. IEEE/CVF Int. Conf. on Computer Vision, pp. 13930-13940, 2021.

[2] D. Ulyanov, A. Vedaldi and V. Lempitsky: Deep image prior, Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition, pp. 9446-9454, 2018.

[3] Y. Blau and T. Michaeli: The perception-distortion tradeoff, Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition, pp. 6228-6237, 2018.

[4] G. Ohayon, T. Adrai, G. Vaksman, M. Elad and P. Milanfar: High perceptual quality image denoising with a posterior sampling CGAN, Proc. IEEE/CVF Int. Conf. on Computer Vision, pp. 1805-1813, 2021.

[5] A. Buades, B. Coll and J.M. Morel: A non-local algorithm for image denoising, IEEE Conf. on Computer Vision and Pattern Recognition, Vol. 2, pp. 60-65, 2005.

[6] K. Dabov, A. Foi, V. Katkovnik and K. Egiazarian: Image denoising by sparse 3-D transform-domain collaborative filtering, IEEE Transactions on Image Processing, Vol. 16, No. 8, pp. 2080-2095, 2007.

[7] S. Suhaila and T. Shimamura: Power spectrum estimation method for image denoising by frequency domain Wiener filter, Int. Conf. on Computer and Automation Engineering (ICCAE), Vol. 3, pp. 608-612, 2010.

[8] N. J. Nyunt, Y. Sugiura and T. Shimamura: Parametric Wiener filter based on image power spectrum sparsity, Journal of Signal Processing, Vol. 22, No. 6, pp. 287-297, 2018.

[9] K. Zhang, W. Zuo, Y. Chen, D. Meng and L. Zhang: Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising, IEEE Transactions on Image Processing, Vol. 26, pp. 3142-3155, 2017.

[10] K. Zhang, W. Zuo and L. Zhang: FFDNet: Toward a fast and flexible solution for CNN based image denoising, IEEE Transactions on Image Processing, Vol. 27, pp. 4608-4622, 2018.

[11] Z. Gao, Y. Wang, X. Liu, F. Zhou and K.K Wong: FFDNet-based channel estimation for massive MIMO visible light communication systems, IEEE Wireless Communications Letters, Vol. 9, No. 3, pp. 340-343, 2019.

[12] X. Zou, F. Xiao, Z. Yu, Y. Li and Y. J. Lee: Delving deeper into anti-aliasing in ConvNets, International Journal of Computer Vision, pp. 67-81, 2022.

[13] S. Dey, R. Bhattacharya and R. Sarkar: Median filter aided CNN model for removal of Gaussian noise from images, IEEE Recent Advances in Intelligent Computational Systems, pp. 178-183, 2020.

[14] Z. Wan, J. Zhang, D. Chen and J. Liao: High-fidelity pluralistic image completion with transformers, Proc. IEEE/CVF Int. Conf. on Computer Vision, pp. 4692-4701, 2021.

[15] T. Wang, H. Ouyang and Q. Chen: Image inpainting with external-internal learning and monochromic bottleneck, Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition, pp. 5120-5129, 2021.

[16] Y. Chen and T. Pock: Trainable nonlinear reaction-diffusion: A flexible framework for fast and effective image restoration, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 39, No. 6, pp. 1256-1272, 2016.

[17] L. Zhang, L. Zhang, X. Mou and D. Zhang: FSIM: A feature similarity index for image quality assessment, IEEE Transactions on Image Processing, Vol. 20, pp. 2378-2386, 2011.

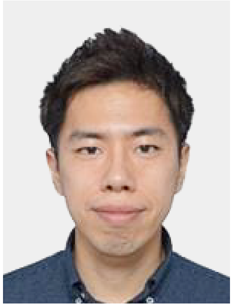
[18] R. Zhang, P. Isola, A. A. Efros, E. Shechtman and O. Wang: The unreasonable effectiveness of deep features as a perceptual metric, Proc. IEEE/CVF Int. Conf. on Computer Vision, pp. 586-595, 2018.

[19] https://www.kaggle.com/datasets

[20] https://imagingsolution.net/tag/sidba/

[21] http://imgcom.jsrt.or.jp/minijsrtdb/

May Thet Tun received her B.C.Sc. (Hons.) and M.C.Sc. degrees from the University of Computer Studies, Yangon, Myanmar, in 2015 and 2018, respectively. She is currently pursuing her Ph.D. degree in the field of digital image processing at the Graduate School of Science and Engineering, Saitama University, Saitama, Japan.



Yosuke Sugiura received his B.E., M.E., and Ph.D. degrees from Osaka University, Osaka, Japan in 2009, 2011 and 2013 respectively. In 2013, he joined Tokyo University of Science, Tokyo, Japan. In 2015, he joined Saitama University, Saitama, Japan, where he is currently an Assistant Professor. His research interests include digital signal processing, adaptive filter theory, and speech information processing.

Tetsuya Shimamura received his B.E., M.E., and Ph.D. degrees in electrical engineering from Keio University, Yokohama, Japan, in 1986, 1988, and 1991, respectively. In 1991, he joined Saitama University, Saitama, Japan, where he is currently a Professor. He was a visiting researcher at Loughborough University, U.K., in 1995 and at Queen's University of Belfast, U.K., in 1996, respectively. Professor Shimamura is an author and coauthor of six books. He serves as an editorial member of several international journals and is a member of the organizing and program committees of various international conferences. His research interests are in digital signal processing and its application to speech, image, and communication systems.

(Received May 21, 2023; revised August 5, 2023)