Magn Reson Med Sci 2021; 20; 410–424
doi:10.2463/mrms.mp.2020-0073

Published Online: February 11, 2021

MAJOR PAPER

Introducing Swish and Parallelized Blind Removal Improves the Performance of a Convolutional Neural Network in Denoising MR Images

Taro Sugai[1], Kohei Takano[2], Shohei Ouchi[3], and Satoshi Ito[1*]

Purpose: To improve the performance of a denoising convolutional neural network (DnCNN) and to make it applicable to images with inhomogeneous noise, a refinement involving an activation function (AF) and an application of the refined method for inhomogeneous-noise images was examined in combination with parallelized image denoising.

Methods: Improvements in the DnCNN were performed by three approaches. One is refinement of the AF of each neural network that constructs the DnCNN. Swish was used in the DnCNN instead of rectifier linear unit. Second, blind noise removal was introduced to the DnCNN in order to adapt spatially variant noises. Third, blind noise removal was applied to parallelized image denoising, referred to herein as ParBID. The ParBID procedure is as follows: (1) adjacent 2D slice images are linearly combined to obtained higher peak SNR (PSNR) images, (2) combined images with different weight coefficients are denoised using the blind DnCNN, and (3) denoised combined images are separated into original position images by algebraic calculation.

Results: Experimental studies showed that the PSNR and the structural similarity index (SSIM) were improved by using Swish for all noise levels, from 2.5% to 7.5%, as compared to the conventional DnCNN. It was also shown that a well-trained CNN could remove spatially variant noises superimposed on images. Experimental studies with ParBID showed that the greatest PSNR and SSIM improvements were obtained at the middle slice when three slice images were used for linear image combination. More fine structures of images and image contrast remained when the proposed ParBID procedure was used.

Conclusion: Swish can improve the denoising performance of the DnCNN, and the denoising performance and effectiveness were further improved by ParBID.

Keywords: *convolutional neural network, noise, rectifier linear unit, Swish*

Introduction

MRI is widely used in the medical field because of its high soft tissue contrast and noninvasiveness. With the increase of main magnetic field strength and other technological advances, MR images have achieved higher spatial resolutions and SNRs. On the other hand, low-SNR images can be obtained by imaging methods such as functional MRI or parallel imaging techniques. There is still a great demand for the removal of noise from MR images. To date, numerous methods have been proposed for denoising MR images, but balancing the conflicting requirements of noise removal while preserving the detailed structure is difficult, and there is still plenty of room for improvement.

Denoising methods that have been developed for natural images are often used to address the issue of denoising in medical images. In recent years, natural image denoising methods incorporating new signal processing methods have been proposed. Such methods include the total variation filter,[1] the total generalized variation filter,[2] the anisotropic diffusion filter, which solves differential equations in the image space,[3] the non-local mean filter,[4] the block-matching and 3D (BM3D)

[1]Department of Innovation Systems Engineering, Graduate School of Engineering, Utsunomiya University, Utsunomiya, Tochigi, Japan

[2]Information and Control Systems Science, Graduate School of Engineering, Utsunomiya University, Utsunomiya, Tochigi, Japan. Now works at ND Software Co., Ltd., Nanyou, Yamagata, Japan

[3]Intelligence and Information Science Course, Graduate School of Engineering Doctoral Degree Program, Utsunomiya University, Utsunomiya, Tochigi, Japan

*Corresponding author: Graduate Program in Information, Electrical and Electronic Systems Engineering, Utsunomiya University, 7-1-2, Yoto, Utsunomiya, Tochigi 321-8585, Japan. Phone/Fax: +81-28-689-6276, E-mail: itohst@is.utsunomiya-u.ac.jp

Received: May 12, 2020 | Accepted: November 27, 2020

filter,[5] which samples regions within the image with similar patterns and performs smoothing, the weighted nuclear norm minimization (WNNM) filter, which adaptively assigns weights to different singular values in the low-rank matrix approximation problem,[6] and the dual-domain image denoising filter, which performs denoising in two spaces: wavelet space and image space.[7] These filters are known to have high denoising abilities while preserving the sharpness of contours of natural images. In medical imaging applications, fine structures and small orifices in the images may provide information about the lesion, and the loss of these structures may reduce the accuracy of diagnostic imaging. Therefore, maintaining spatial resolution and image contrast in medical imaging are more important than obtaining natural images.

Recently, the application of deep learning to the image denoising problem has attracted significant attention, not only for natural images, but also for medical images. He et al. proposed residual learning in a deep learning network[8] to speed up the training process and improve the performance of image recognition. Zhang et al. constructed a denoising convolutional neural network (DnCNN) for natural images using residual learning.[9] A rectifier linear unit (ReLU) and batch normalization (BN) are used to boost the denoising performance. Zhang et al. demonstrated that the DnCNN exhibited high effectiveness in several image denoising tasks.[9] Inspired by the DnCNN of Zhang et al., several attempts at applying this method to MR images have been reported. Manjón et al. applied the DnCNN to 3D MR images by constructing a 3D CNN.[10] Jiang et al. applied the DnCNN to 2D MR images as a preprocess for image segmentation.[11] Latif et al. proposed a denoising method that combines the DnCNN and an anisotropic diffusion filter[12] to obtain better results for the segmentation of the tumorous portion of 3D MR images.[13] All of these studies used ReLU as an activation function (AF). Isogawa examined the soft thresholding function for AF in an image denoising CNN[14] and demonstrated adaptivity to natural images with various noise levels in one CNN. The main function of soft thresholding is to make the CNN adjustable to unknown noise levels. The AF of the CNN is still being improved, and there is still room for improvement.

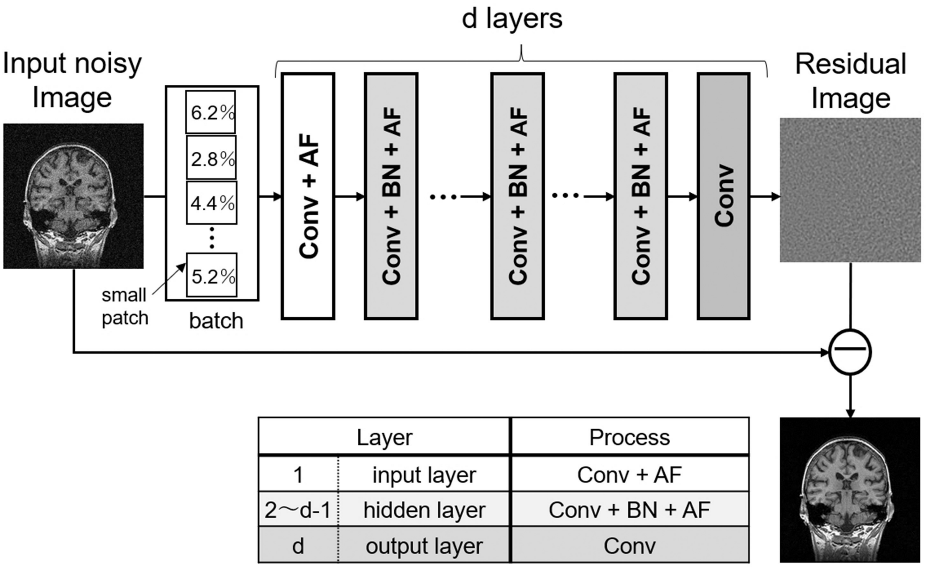
In the present study, the DnCNN was improved by three approaches. One is refinement of the AF of each neural network that constructs the DnCNN. The ReLU function has become a widely used AF in many CNN applications. However, the discussion of the AF is insufficient as related to the DnCNN because gradients are able to propagate only when the input to ReLU is positive and show zero activations and derivatives in the negative region. Recently, Ramachandran et al. proposed Swish as a new AF, with which they showed that Swish matches or outperforms ReLU on many applications using deep learning networks, such as image classification or machine translation.[15]

In the present paper, we adopted Swish as an AF of the DnCNN in order to improve the denoising performance of the original DnCNN using ReLU. Second, blind noise removal was introduced to the MR image, in which the noise level varies spatially. The DnCNN can be extended to handle general image denoising tasks, where the noise level of the image is unknown.[9] Zhang et al. revealed the effectiveness of blind denoising for natural images (referred to herein as denoising convolutional neural network-blind [DnCNN-B]). Kidoh et al.[16] and Isogawa et al.[14] also demonstrated image denoising methods with various noise levels in one CNN. However, these studies did not state the effectiveness of their methods for images in which the noise level varies spatially on the image space. In general, the noise levels of MR images are unknown and sometimes vary spatially in such cases for the sensitivity encoding method (SENSE[17]). However, accurately estimating the noise level on the image space is difficult. Therefore, a denoising method to deal with images in which the noise level varies spatially and the associated learning method were investigated in the present paper. Third, we applied this blind image denoising to parallelized image denoising. If adjacent slice images have a similar anatomical structure and the noise distribution of each slice image is independent, then averaging these slice images will result in improvement of the SNR. Denoising these averaged images is expected to result in the improvement of image sharpness and contrast preservation. Blind noise removal using the DnCNN is useful and effective for denoising these combined images, in which noise levels are unknown and vary spatially. Since combined images suffer from blurring, slice images are separated by solving linear equations. The refined DnCNN with parallelized denoising was compared with the conventional DnCNN and state-of-the-art nonlinear denoising filters.

Materials and Methods

Network architecture and optimization

The basic neural network used in the present study is the DnCNN proposed by Zhang et al.[9] The architecture of the DnCNN is illustrated in Fig. 1. The network depth $d = 17$ and the receptive field size of 35 were used. Convolution (Conv) and an AF were applied in the first layer. BN was added between Conv and AF, and Conv, BN, and AF were applied from the second layer up to the second-to-last layer. Finally, Conv alone was applied in the final layer. Here, Conv and AF serve to extract features of the input data, whereas BN serves to boost learning efficiency. A total of 64 filters of size 3 × 3 were used for network convolution, which exhibited good outputs.[18] In denoising tasks of MR images, the size of output image should be the same as that of the input image. Therefore, simple zero data padding at the boundary of noisy input images was carried out so that the feature map of the middle layers has the same size as the input image.[9]



Layer		Process
1	input layer	Conv + AF
2～d-1	hidden layer	Conv + BN + AF
d	output layer	Conv

Fig. 1 The architecture of DnCNN network for blind noise. First layer has Conv and an AF. BN is added between Conv and AF in the middle layers 2–(d-1). To handle images with blind and spatially variant noises, patches with different noise levels are combined in a single batch. AF, activation function; BN, batch normalization; Conv, convolution; DnCNN, denoising convolutional neural network.

Adam[19] was used in order to minimize the value of the loss function and bring the network closer to the optimal state for updating the training parameters. We trained the DnCNN for 20 epochs. The initial learning rates were as follows: 1.0×10^{-3} in epochs 1 through 10 and 1.0×10^{-4} in epochs 11 through 20.

Swish

Swish can be described as the multiplication of the input x with the sigmoid function, as shown in Eq. [1], where β is a constant. Swish becomes a linear function $f(x) = x/2$ when $\beta = 0$ and becomes like the ReLU Function When $\beta \to \infty$.[15] Fig. 2 plots the profiles of Swish for β values from 0.1 to 4.0. The function looks like ReLU in the positive region, but has a bump shape in the negative region, i.e., decreases from 0 toward the negative region of the input signal and then increases again.

$$f(x) = \frac{x}{1+\exp(-\beta x)} \qquad [1]$$

In the present study, Swish was used as an AF in order to improve the denoising performance in the modified DnCNN. The refined method is hereinafter referred to as the denoising convolutional neural network with Swish (DnCNN[S]) in order to distinguish it from the conventional DnCNN.

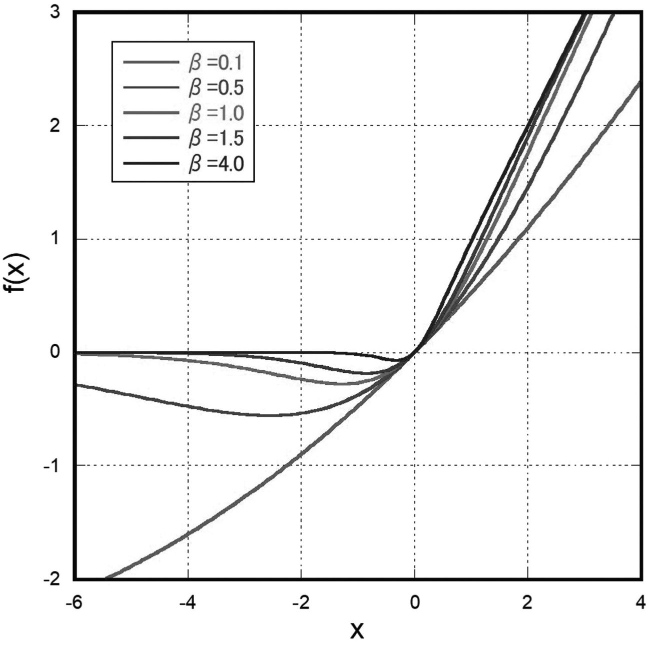


Fig. 2 Profiles of Swish varying the parameter.

Data preparation

The MR signal obtained in MRI is acquired as complex values that include noise. This noise is well modeled by a Gaussian probability density function in the real and imaginary parts of the complex signal. In the simulation experiments, Gaussian noise was added to the real and imaginary parts of the MRI signals calculated by MR image models. MRI images were then reconstructed by applying inverse Fourier transform to the noisy MRI signals. A reconstructed image with superimposed noise can be expressed as:

$$y = x + n \qquad [2]$$

where x denotes an image without spatial phase variation and noise and n denotes Gaussian noise. In the present study, since the images used to calculate MR signals were magnitude images, the imaginary part of reconstructed image y contained

Table 1 Imaging conditions of MR image data set used in the experiments

Data set	Imaging subjects	Imaging conditions	
Data set A	400 MR (T1W, T2W, PDW) images (18 person, 10 male and 8 females, ages from 26 to 40)	(IXI dataset)	
		MR scanner	Philips 1.5 T
		T1-weighted	TR = 9.813 ms, TE = 4.603 ms, NPE = 192, FA = 8 degrees
		T2-weighted	TR = 8178.34 ms, TE = 100 ms, N_{PE} = 187, echo train length = 16, Flip angle = 90 degrees
Data set B	900 MR (T1W, T2W, PDW) images (18-person, 10 male and 8 females, ages from 26 to 40)	PD-weighted	TR = 8178.34 ms, TE = 8 ms, N_{PE} = 187, echo train length = 16, Flip angle = 90 degrees
Data set C	1-person, male age 26	MR scanner	Canon Medical VantageTitan 3.0 T, 3D fast spin echo, TR = 3500 ms, TE = 352 ms, spatial resolution = 1.1 mm, flip angle = 90 degrees, slice thickness = 1.2 mm, slice spacing = 1.2 mm, 256 × 256 pixel image for x–y slices

FA, flip angle; NPE, number of phase encoding; PDW, PD-weighted; T1W, T1-weighted; T2W, T2-weighted.

only noise. Reconstructed images are often provided as the magnitude images, which are the absolute values of the complex images. Therefore, we attempted to denoise the magnitude of the real-part of the complex MRI image y. Noise level is defined as the ratio of $\mathrm{Re}(\sigma_n)/\max(\mathrm{Re}[x])$ and is expressed a percentage in the present paper, where σ_n represents the standard deviation of the added noise, and "Re" is the real part of the signal. The initial Gaussian noise n in image space is nonlinearly transformed, and noise in the image then becomes Rician-distributed noise. When the SNR is high, the Rician distribution can be conveniently approximated by a Gaussian distribution.[20]

We used the IXI dataset[21] including normal, healthy volunteer images for training the CNN and denoising tests. Strictly speaking, MR images contained in the IXI dataset contain a small amount of noise. We regarded these images as noise-free data. We used three datasets: A, B, and C. The imaging conditions of these datasets are listed in Table 1. Datasets A and B were used for CNN learning and testing and dataset C was used only for testing. In dataset A, 400 MR images contained in the IXI dataset were mixed in order to split the IXI dataset into four datasets, and three of the four datasets were used for training and one for testing. In dataset B, 900 MR images contained in the IXI dataset were mixed to split the IXI dataset into three datasets, and two of the three datasets were used for training and one for testing. In dataset C, healthy volunteer images obtained with 3T MRI were used. Informed consent was obtained from the volunteers.

Data augmentation, which is a technique to increase the amount of effective training data, was used to boost the robustness of the network. In the present study, data augmentation by random rotation and scale change was used on the training datasets. Random rotation in this case means a random selection and execution of any one of eight processes: (1) no change, (2) horizontal flipping, (3) 90° rotation, (4) 90° rotation and horizontal flipping, (5) 180° rotation, (6) 180° rotation and horizontal flipping, (7) 27° rotation, and (8) 270° rotation and horizontal flipping. Scale change converts the image scale by factors of 1.0, 1.0×0.9, $1.0 \times 0.9 \times 0.8$, and $1.0 \times 0.9 \times 0.8 \times 0.7$, and the converted images were then used to train for various object sizes.

Parallelized blind image denoising

In the training of the DnCNN-B network, images with several noise levels were broken down into small patches, and these patches were then batched in such a way that patches with different noise levels were combined in a single batch, as shown in Fig. 1.

If the noise fluctuates randomly about a zero mean and its statistics are independent of the position on the image, the averaging of acquired images will improve the SNR of images. In the accumulated images during averaging, the amount of noise increases in proportion to the square root of the number of accumulations, whereas the amplitude of image components increases in proportion to the accumulation. Therefore, the SNR of an averaged image is expected to improve in proportion to the square root of the number of averagings. Similar to this example, the SNR of an averaged image will improve if the 2D slice images have distributions that are similar to those of adjacent images, as in the cases of multi-slice 2D imaging or 3D Fourier transform imaging. In general, denoising of higher-SNR images contributes to smaller degradation of images and higher preservation of structures. The parallelized blind image denoising (referred to herein as ParBID) procedure consists of three steps. Figure 3 illustrates the scheme of the ParBID. The first step is improvement of the image SNR by linearly combining multi-slice images with given weights. Let the kth slice image be $r_k = \rho_k + \delta_k$, where ρ_k is a noise-free image and δ_k is the noise superimposed on the image ρ_k, then the linear

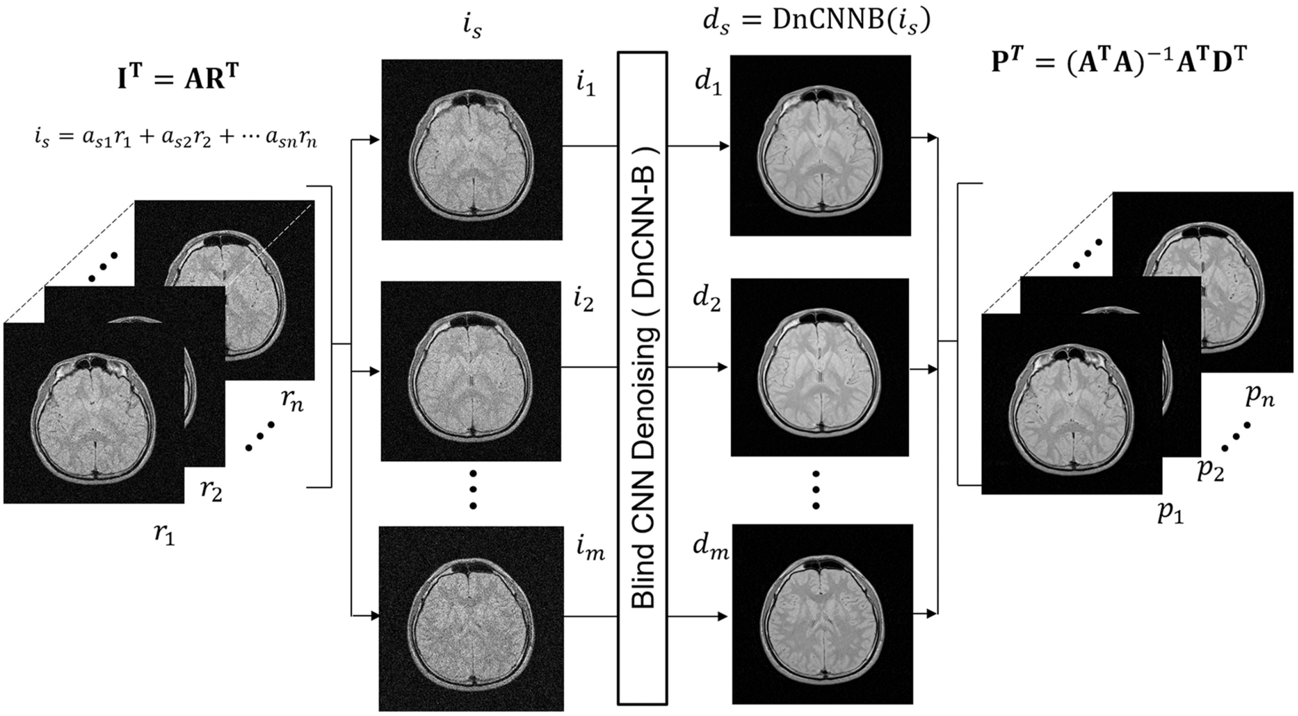

i_s
$d_s = \mathrm{DnCNNB}(i_s)$
$\mathbf{I^T = AR^T}$
$\mathbf{P}^{\boldsymbol{T}} = (\mathbf{A^T A})^{-1}\mathbf{A^T D^T}$
$i_s = a_{s1}r_1 + a_{s2}r_2 + \cdots a_{sn}r_n$
i_1
i_2
i_m
d_1
d_2
d_m
Blind CNN Denoising (DnCNN-B)
r_1
r_2
r_n
p_1
p_2
p_n

Fig. 3 Schematic of ParBID. The first step is the linear combination of adjacent images with given weights *a*. The second step is the blind DnCNN of the combined images. The third step is the separation of linearly combined images by solving linear equations. DnCNN, denoising convolutional neural network; ParBID, denoising convolutional neural network.

combinations of slice images i_s are written as follows using weight coefficients $a_{s,k}$:

$$i_s = \sum_k a_{s,k} r_k \qquad [3]$$

$$= \sum_k a_{s,k}\rho_k + \sum_k a_{s,k}\delta_k. \qquad [4]$$

Equation [3] can be made linear by varying weight coefficients $a_{s,k}$. By assembling noisy and combined images in vectors, the linear image combination can be rewritten in matrix notation:

$$\mathrm{I}^{\mathrm{T}} = \mathrm{AR}^{\mathrm{T}} \qquad [5]$$

where R = (r_1, r_2,…, r_n), I = (i_1, i_2,…, i_n), A is $n \times m$ size weight coefficient matrix, and T indicates the transverse of the vector. The linearly combined image i_s has a higher SNR than r_s, while blurring will occur due to the averaging of multi-slice images. The second step is blind denoising of the combined images:

$$d_{\mathrm{s}} = \mathrm{DnCNNB}(i_s) \qquad [6]$$

where DnCNNB(i_s) refers to DnCNN-B operation with Swish (hereinafter DnCNN-B refers to DnCNN-B operation with Swish), and d_s indicates the denoised image by the DnCNN-B. Since averaged image i_s has a higher SNR than r_s and the noise distribution on i_s varies according to weights $a_{s,k}$ in Eq. [4], the manner of noise removal is considered to vary in each denoising process of i_s in Eq. [6]. The third step is the separation of linearly combined images by solving linear equations. When the weighting matrix A is a full row rank, which makes $\mathrm{A^TA}$ a regular matrix, denoised images can be obtained by solving Eq. [7]. Blurring of images is canceled by this final step:

$$\mathrm{P} = (\mathrm{A^T A})^{-1}\,\mathrm{A^T D^T} \qquad [7]$$

where P and D are vectors of separated denoised image p_s and image d_s, respectively, and P = (p_1, p_2, …, p_n) and D = (d_1, d_2, …, d_m). Image sequence (p_1, p_2, …, p_n) is the output of ParBID, with which we are herein concerned. The ParBID experiments were carried out using from two to four slices in our examination. We refer to ParBID using *s* slices for averaging as "*s*-slice ParBID". Coefficients of linear combination $a_{s,k}$ for 2-slice to 4-slice ParBID used in Eq. [4] were determined by preliminary examination as (0.6, 0.4), (0.5, 0.3, 0.2), and (0.4, 0.3, 0.2, 0.1), and these values were used in a cyclical manner, for example, $i_1 = 0.5r_1 + 0.3r_2 + 0.2r_3$, $i_2 = 0.3r_1 + 0.2r_2 + 0.5r_3$, and $i_3 = 0.3r_1 + 0.2r_2 + 0.5r_3$ were used for 3-slice ParBID. ParBID is assumed to be used in a manner such that the window function that defines the slice sequence R used for the linear equation shown in Eq. [5] moves along the slice direction. In other words, target slice images to be denoised should be placed in the center of the slice sequence, except for the 2-slice ParBID.

Comparison of images

Testing was performed using the test data sets listed in Table 1. In order to evaluate the obtained images quantitatively, we used the peak SNR (PSNR) and the structural similarity index (SSIM)[22] value defined as follows:

$$\mathrm{PSNR} = 20\log_{10}\frac{\max[\rho_\mathrm{o}]}{\mathrm{RMSE}} \qquad [8]$$

in which root mean square error (RMSE) is the root mean square error of the remaining noise using the difference between the original image ρ_o and the denoised image. The SSIM is a quality metric used to measure the similarity between two images.[22] The SSIM is considered to be correlated with the quality perception of the human visual system and is defined as follows:

$$\mathrm{SSIM} = \left(\frac{2\mu_\mathrm{r}\mu_\mathrm{t}+C_1}{\mu_\mathrm{r}^2+\mu_\mathrm{t}^2+C_1}\right)\left(\frac{2\sigma_\mathrm{r}\sigma_\mathrm{t}+C_2}{\sigma_\mathrm{r}^2+\sigma_\mathrm{t}^2+C_2}\right)\left(\frac{\sigma_\mathrm{rt}+C_3}{\sigma_\mathrm{r}+\sigma_\mathrm{t}+C_3}\right) \qquad [9]$$

where *r* and *t* indicate the reference and test images, respectively. The positive constants C_1, C_2, and C_3 are used in order to avoid a null denominator and are defined as follows:

$$C_1 = (K_1L)^2,\ C_2 = (K_2L)^2,\ C_3 = \frac{C_2}{2} \qquad [10]$$

where *L* is the maximum pixel value, and K_1 and K_2 are small constants. We use $K_1 = 0.01$ and $K_2 = 0.03$ following the original paper reporting this index.[22]

First, comparison of the reconstructed images was performed in order to validate the effectiveness of Swish as an AF. In order to examine the appropriate Swish parameter β in the DnCNN architecture, denoising tests were executed varying the parameter β from 0.1 to 2.0. Dataset A was used for the learning and denoising tests. Denoising performances were compared in terms of the PSNR among the methods, i.e., the DnCNN(S), DnCNN with ReLU, BM3D, and WNNM at each noise level of 2.5%, 5.0%, and 7.5%.

In order to evaluate the denoising performance for blind noise removal with DnCNN-B, the PSNR and SSIM of denoised images were compared with the DnCNN(S) at noise levels of 2.5%, 5.0%, and 7.5%. Dataset B was used for learning and denoising tests. In the following, DnCNN-B was applied to images in which the noise level varied spatially and the denoising performances were compared with the DnCNN(S) trained with a single noise level as a control.

ParBID

Examinations of from 2-slice to 4-slice ParBID were performed at noise levels of 2.5%, 5.0%, and 7.5% using the PSNR of the DnCNN(S) as a control. Dataset B was used for learning and denoising tests. Improvements of the PSNR and SSIM were examined with reference to the number of slices used for ParBID. In order to evaluate the effectiveness of ParBID for experimentally obtained noisy images, ParBID was applied to noisy images acquired with 1.5 T MRI (Gyroscan Intera, Philips Medical Systems, Eindhoven, the Netherlands). Dataset B was used for learning and dataset C was used for denoising tests.

Statistical comparisons

Since denoising tests using Swish or ReLU could be performed for the same noisy MR images with the same noise distributions, a two-sided paired *t*-test was performed between Swish-based DnCNN and ReLU-based DnCNN at noise levels of 2.5%, 5.0%, and 7.5%. We assumed that the significance level was 5% and the mean PSNR difference, the standard deviation (s.t.d.) of the mean PSNR difference (s.t.d. of mean), the 95% confidence interval, the *t*-value, the degree of freedom (d.f.), and the *P*-value were calculated.

For the denoising evaluation, we used a computer equipped with an Intel Core i7-7700 (3.60 GHz) CPU (Intel, Santa Clara, CA, USA), 32 GB of memory, and an GeForce GTX 1080 Ti GPU (NVIDIA, Santa Clara, CA, USA) with 11 GB of memory. It took approximately 9.6 hours to train a 600-image dataset using the GPU. We used Visual Studio 2017 (Microsoft, Redmond, WA, USA), MATLAB R2017b (MathWorks, Natick, MA, USA) with MatConvNet package,[23] CUDA 9.0 (NVIDIA), and cuDNN v7.0.5 (NVIDIA) for the framework.

Results

Application of Swish to the DnCNN (DnCNN[S])

Table 2 shows the results of the obtained PSNRs and standard deviations with Swish parameter β = 0.1, 0.5, 1.0, 1.5, and 2.0. Table 2 also shows the PSNRs obtained with the DnCNN with ReLU, BM3D, and WNNM for comparison. The highest PSNR was obtained with Swish at β = 1.5 for all noise levels. Based on these results, we hereinafter used Swish with β = 1.5 for the AF of the DnCNN.

Figure 4 shows the box-and-whisker diagram of the PSNR difference between Swish-based DnCNN ($\beta = 1.5$) and the ReLU-based DnCNN at noise levels of 2.5%, 5.0%, and 7.5%. Higher PSNRs were obtained with Swish for 95 out of 100 images at a noise level 2.5%, and all 100 images show higher PSNRs at noise levels of 5.0% and 7.5%. Table 3 summarizes the results of two-sided paired *t*-tests at noise levels of 2.5%, 5.0%, and 7.5%. The mean PSNR difference, the standard deviation of the mean PSNR difference (s.t.d. of mean), the 95% confidence interval, the *t*-value, the degree of freedom (d.f.), and *P*-value are listed. The *P*-values of 5.41×10^{-6} for 2.5%, 1.65×10^{-19} for 5.0%, and 1.04×10^{-17} for 7.5% shown in Table 3 indicate a statistically significant superiority in PSNR improvement of our method.

Figure 5 shows the denoising results for a PDW image adding 5.0% noise. Subimages (a) and (b) show images before and after adding 5.0% noise, respectively, and subimages (c) through (f) show denoised images using Swish (β = 1.5), ReLU, WNNM, and BM3D, respectively. The time required for denoising one image in the DnCNN(S) was approximately 2s using CPU computing, which is similar to BM3D, whereas that for WNNM was approximately

Table 2 Comparison of PSNR in DnCNN with reference to the Swish parameter β at noise levels of 2.5%, 5.0% and 7.5%. PSNRs obtained in DnCNN with ReLU, WNNM, and BM3D are also examined for comparison. Bold figures indicate the maximum PSNR at each noise level

Noise level Method		2.5%	5.0%	7.5%
DnCNN	$B = 0.1$	36.50 ± 0.648	32.71 ± 0.711	30.59 ± 0.734
β: Swish	$B = 0.5$	36.79 ± 0.633	32.92 ± 0.701	30.82 ± 0.722
parameter	$B = 1.0$	36.82 ± 0.624	32.95 ± 0.694	30.84 ± 0.720
	$B = 1.5$	**36.86 ± 0.595**	**32.98 ± 0.683**	**30.87 ± 0.721**
	$B = 2.0$	36.85 ± 0.596	32.97 ± 0.685	30.83 ± 0.719
DnCNN (ReLU)		36.77 ± 0.635	32.88 ± 0.704	30.77 ± 0.726
BM3D		35.63 ± 0.621	31.98 ± 0.687	29.92 ± 0.702
WNNM		36.04 ± 0.626	32.41 ± 0.690	30.05 ± 0.711

BM3D, block-matching and 3D; DnCNN, denoising convolutional neural network; PSNR, peak SNR; ReLU, rectifier linear unit; WNNM, weighted nuclear norm minimization.

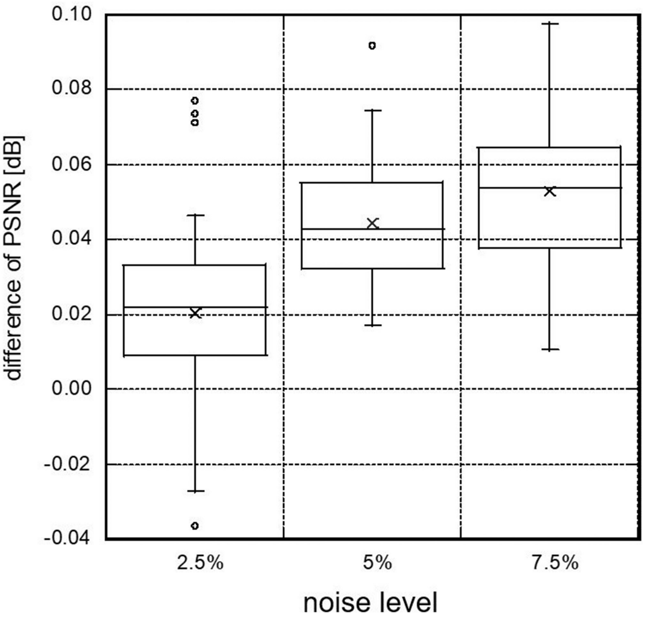


Fig. 4 Box-and-whisker diagram of the PSNR difference between DnCNN using Swish ($\beta = 1.5$) and ReLU at noise levels 2.5%, 5.0%, and 7.5%. DnCNN, denoising convolutional neural network; PSNR, peak SNR; ReLU, rectifier linear unit.

Table 3 Results of two-sided paired *t*-tests of PSNR difference between Swish-based DnCNN and ReLU-based DnCNN

Noise level	Mean difference	s.t.d. of mean	95% confidence interval		*t*-Value	d.f.	*P*-value Two-sided
			Lower	Upper			
2.5% pair	0.0204	0.0245	0.01491	0.02582	4.81	99	5.41e–6
5.0% pair	0.0444	0.0166	0.04812	0.04942	11.3	99	1.65e–19
7.5% pair	0.0528	0.0224	0.04778	0.05775	10.5	99	1.04e–17

d.f., degree of freedom; DnCNN, denoising convolutional neural network; ReLU, rectifier linear unit; s.t.d., standard deviation.

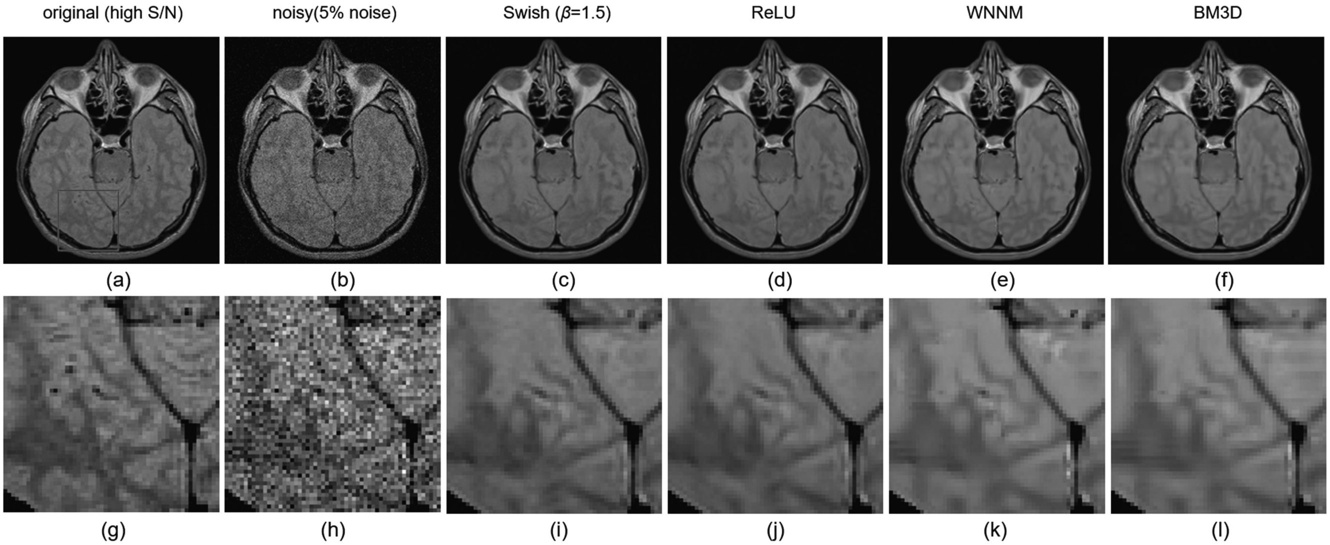


Fig. 5 Comparison of denoised images. Subimage (**a**) and (**b**) show images before and after adding 5% noise. Subimage (**c**) and (**d**) show denoised images by DnCNN using Swish ($\beta = 1.5$) and ReLU, respectively. Subimage (**e**) and (**f**) show denoised images by WNNM and BM3D. Enlarged images of the area inside the red box on subimage (**a**) are shown in subimages (**g**)–(**l**). Contrast of images is adjusted in the same condition in order to make it easier to see the differences between them. BM3D, block-matching and 3D; DnCNN, denoising convolutional neural network; ReLU, rectifier linear unit; WNNM, weighted nuclear norm minimization.

70s. Enlarged images of the area inside the red box on subimage (a) are shown in subimages (g) through (l). The contrast of images is adjusted to the same condition in order to make it easier to see the differences between the images. Comparing the DnCNN-based method using ReLU or Swish with the conventional WNNM and BM3D, the contrast of the images remained to a greater degree for the DnCNN-based method. Comparing the performances for AFs used in the DnCNN, denoised images by Swish retain the tissue structure more clearly, as compared to those by ReLU.

DnCNN-B

Figure 6A and 6B shows the results of the PSNR and SSIM evaluations of DnCNN-B for noise levels from 2.5% to 7.5%. The PSNR and SSIM characteristics of DnCNN-B versus noise levels are shown as red lines, and these values for the DnCNN(S) trained with fixed noise levels of 2.5%, 5.0%, and 7.5% are shown in the same graph for comparison. The DnCNN(S) trained with a fixed noise level shows the highest PSNRs at the noise level at which the network was trained. However, the performances are degraded significantly at other noise levels. In contrast, DnCNN-B shows the performance comparative to the PSNR and the SSIM of the DnCNN(S) trained with fixed noise levels of 2.5%, 5.0%, and 7.5%.

Denoising tests using an inhomogeneous-noise image are shown in Fig. 7. Figure 7A shows a map of the noise level in the FOV, where the noise levels are maintained at a certain level in a small segment and the noise levels were varied segment to segment from 2.5%, 5.0%, and 7.5%. Figure 7B and 7C shows images before and after adding spatially variant noise, and subimages (d) and (e) show the denoised images by the DnCNN-B and removed noise that is calculated by subtracting denoised image (d) from noisy image (c). The amount of removed noise shows a spatially variant distribution consistent with the noise levels shown in Fig. 7A. Figure 7F–7k is enlarged images of the region inside the red box shown in Fig. 7B. The images are the original image, an image containing spatially variant noise, the DnCNN-B image, and DnCNN(S) images trained with fixed noise levels of 7.5%, 5.0%, and 2.5%, respectively. The obtained images by the DnCNN(S) with 7.5%, 5.0%, and 2.5% noises exhibit adequate denoising for the region at which the trained and tested noise levels match. However, oversmoothing or residual noise appears where the trained and tested noise levels do not match, as shown in Fig. 7I–7k. In contrast, DnCNN-B shows an image preserving the fine structure of the subject and is comparable to the images of the DnCNN(S), with an adequate noise level. Figure 8 shows the PSNRs of each segment, segments A, B, and C, and the average of all segments. Similar to the homogeneous-noise evaluation of Fig. 6, DnCNN-B shows PSNRs comparative to the case in which the noise levels of the tested image and trained images match. These results indicate that blind denoising can appropriately remove inhomogeneous noise in the image space.

Application of ParBID

Let the sequential *s* slice images be ($\rho_1, \rho_2, \ldots, \rho_n$) in order, then improvements in the PSNR and the SSIM were varied depending on the position in the slice sequence. Table 4 summarizes the PSNR and the SSIM of each slice image obtained by ParBID at noise levels of 2.5%, 5.0%, and 7.5%. ParBID from 2-slice to 4-slice was evaluated using the PSNR of DnCNN-B as a control.

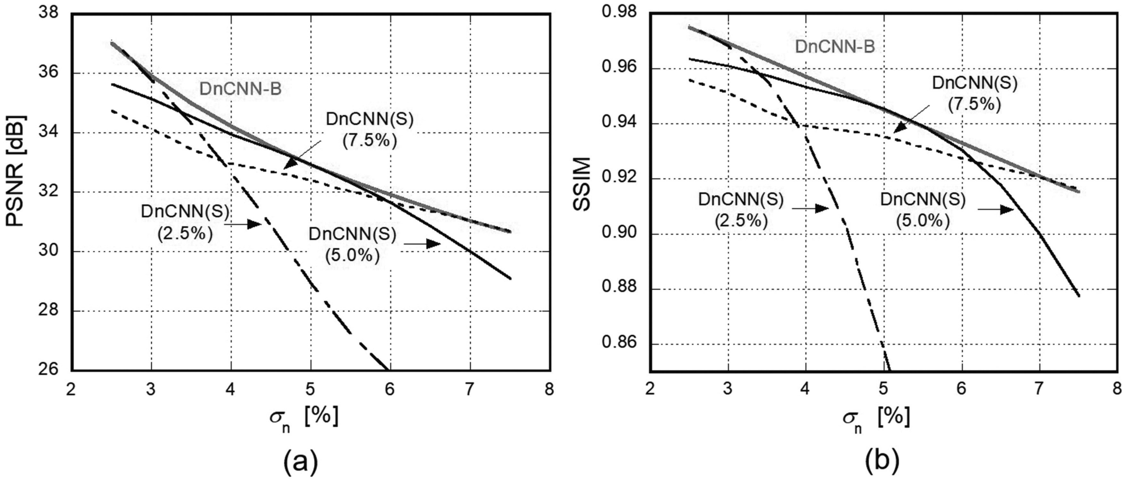


Fig. 6 PSNR and SSIM characteristics of DnCNN and DnCNN-B. Subimages (**a**) and (**b**) show PSNR and SSIM, respectively. DnCNN(S)s trained with fixed noise level show highest performances at the noise level where the network was trained, however, the performances are degraded significantly at other noise levels. DnCNN, denoising convolutional neural network; DnCNN-B, denoising convolutional neural network-blind; DnCNN(S), denoising convolutional neural network with Swish; PSNR, peak SNR; SSIM, structural similarity index.

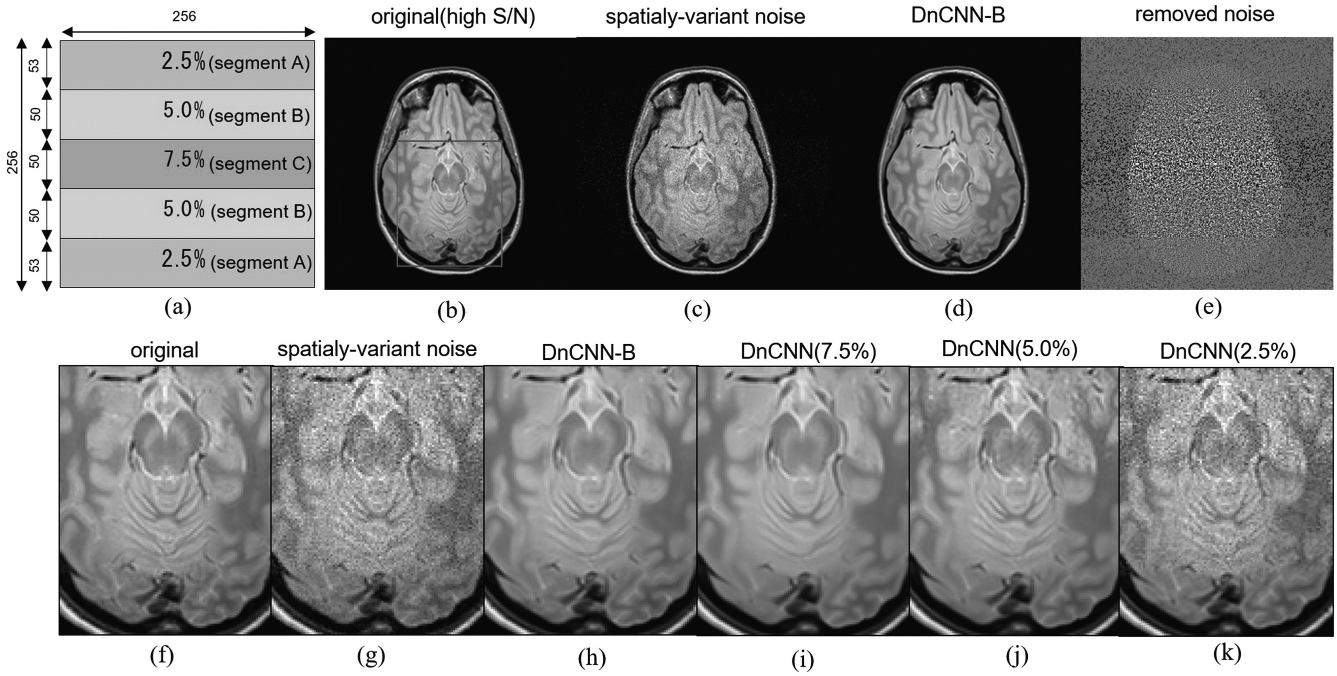


Fig. 7 Denoising results using inhomogeneous-noise image. Subimage (**a**) shows the noise map on the image space where noise levels are varied vertically 2.5%, 5.0% and 7.5%. Subimage (**b**) and (**c**) show images before and after adding inhomogeneous noise, and (**d**) and (**f**) show the denoised images by DnCNN-B and removed noise, respectively. Subimage (**f**)–(**l**) are enlarged images of the region surrounded by red box shown in (**b**) in original, spatially-variant noise, DnCNN-B, DnCNN(S) trained with fixed noise level 7.5%, 5.0%, and 2.5%, respectively.DnCNN, denoising convolutional neural network; DnCNN-B, denoising convolutional neural network-blind; DnCNN(S), denoising convolutional neural network with Swish.

Since almost the same PSNR and SSIM values were obtained at both end slices (ρ_1, ρ_3) in 3-slice ParBID, and at both end slices (ρ_1, ρ_4) and middle slices (ρ_2, ρ_3) in 4-slice ParBID, the averages of these values are listed in Table 4. The results show that improvements in the PSNR and the SSIM are greater at the middle slice, as compared to either end slice.

Figure 9 shows the results of the PSNR and SSIM evaluations of ParBID at each noise level. The highest PSNR and SSIM values obtained for the middle slice in each ParBID were used. The PSNR and the SSIM improve up to three slices compared to DnCNN-B as a control and then decrease for all noise levels. The improvement in the PSNR was

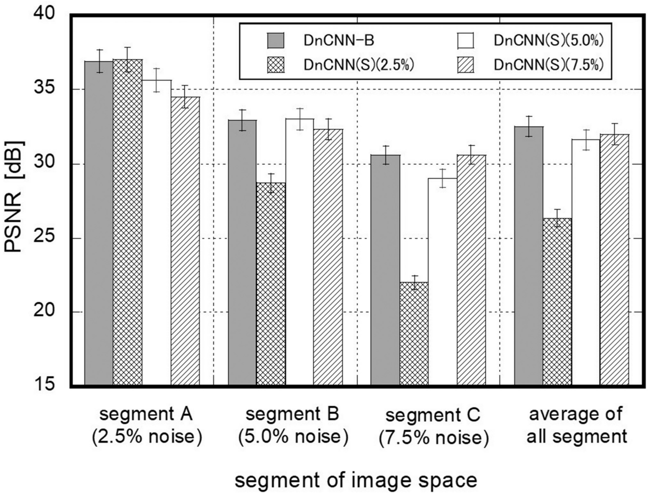


Fig. 8 Results of PSNR evaluation using inhomogeneous-noise image. PSNR were calculated on segment-by-segment basis as well as overall image space. DnCNN, denoising convolutional neural network; DnCNN-B, denoising convolutional neural network-blind; PSNR, peak SNR.

Table 4 PSNR and SSIM of denoised image using ParBID at noise levels of 2.5%, 5.0% and 7.5%

Noise level σ_n	DnCNN-B	ParBID				
	1-slice	2-slice	3-slice		4-slice	
		End	End	Middle	End	Middle
(a) PSNR						
2.5%	37.28 ± 0.598	37.75 ± 0.601	37.85 ± 0.604	**38.11 ± 0.610**	37.71 ± 0.601	37.73 ± 0.605
5.0%	33.05 ± 0.683	33.31 ± 0.689	33.28 ± 0.685	**33.52 ± 0.696**	33.32 ± 0.692	33.35 ± 0.692
7.5%	30.82 ± 0.721	30.92 ± 0.723	30.88 ± 0.730	**30.95 ± 0.736**	30.89 ± 0.732	30.91 ± 0.735
(b) SSIM						
2.5%	0.9790 ± 2.83e−3	0.9798 ± 2.85e−3	0.9797 ± 2.85e−3	**0.9804 ± 2.86e−3**	0.9801 ± 2.85e−3	0.9803 ± 2.86e−3
5.0%	0.9450 ± 4.61e−3	0.9471 ± 4.62e−3	0.9469 ± 4.62e−3	**0.9478 ± 4.64e−3**	0.9475 ± 4.62e−3	0.9477 ± 4.63e−3
7.5%	0.9140 ± 6.15e−3	0.9151 ± 6.15e−3	0.9155 ± 6.18e−3	**0.9165 ± 6.21e−3**	0.9158 ± 6.19e−3	0.9163 ± 6.20e−3

Let the sequential *s* slice images be (p_1, p_2, …, p_s) in order, end slices are (p_1, p_2), (p_1, p_3) and (p_1, p_4) for 2-, 3- and 4-slice ParBID, respectively, and middle slice images are p_2 and (p_2, p_3) for 3- and 4-slice ParBID, respectively. DnCNN, denoising convolutional neural network; DnCNN-B, denoising convolutional neural network-blind; ParBID, parallelized blind image denoising; PSNR, peak SNR; SSIM, structural similarity index.

greater for lower noise levels at 2.5%, as shown in Fig. 9A, and the improvement in the SSIM is greater for higher noise levels at 7.5%, as shown in Fig. 9F. Based on these results, the advantage of ParBID became evident because PSNRs obtained in ParBID improved compared to the control.

Figure 10 shows the denoised image with ParBID. The obtained PSNRs and SSIMs are given with the denoising method in subimages (e) through (i). Enlarged images of the region inside the red box shown in Fig. 10A are compared. Subimages (b) and (c) show the images before and after adding 5.0% noise. The linear combination of adjacent images is shown in subimage (d). Subimages (e) through (i) show the denoised images using WNNM, BM3D, single-slice DnCNN-B, and the 2- and 3-slice ParBID, respectively. Details of the subjects and image contrast are preserved to a greater degree in 3-slice ParBID, as compared to single-image DnCNN-B, as shown in the region indicated by the red dashed circle. The PSNR and SSIM of subimage (d) were improved compared to DnCNN-B image (g). Application of ParBID to an experimentally obtained MR image is shown in Fig. 11. Magnitude images (256 × 256) were used for testing. Subimage (a) is the target image, and subimages (b) and (c) are denoised images by BM3D and WNNM, respectively. Denoised images by DnCNN-B, 2-slice ParBID, and 3-slice ParBID are shown in subimages (d) through (f), respectively. Subimages (g) through (l) are enlarged images of (a)–(f), respectively, for the region surrounded by the red box in subimage (a). Denoised images (f) and (l) obtained by 3-slice ParBID clearly retain image contrast, as shown in the region indicated by the white arrow. Subimages (m) and (n) are the difference images between 3-slice ParBID (l) and DnCNN-B (j) and the

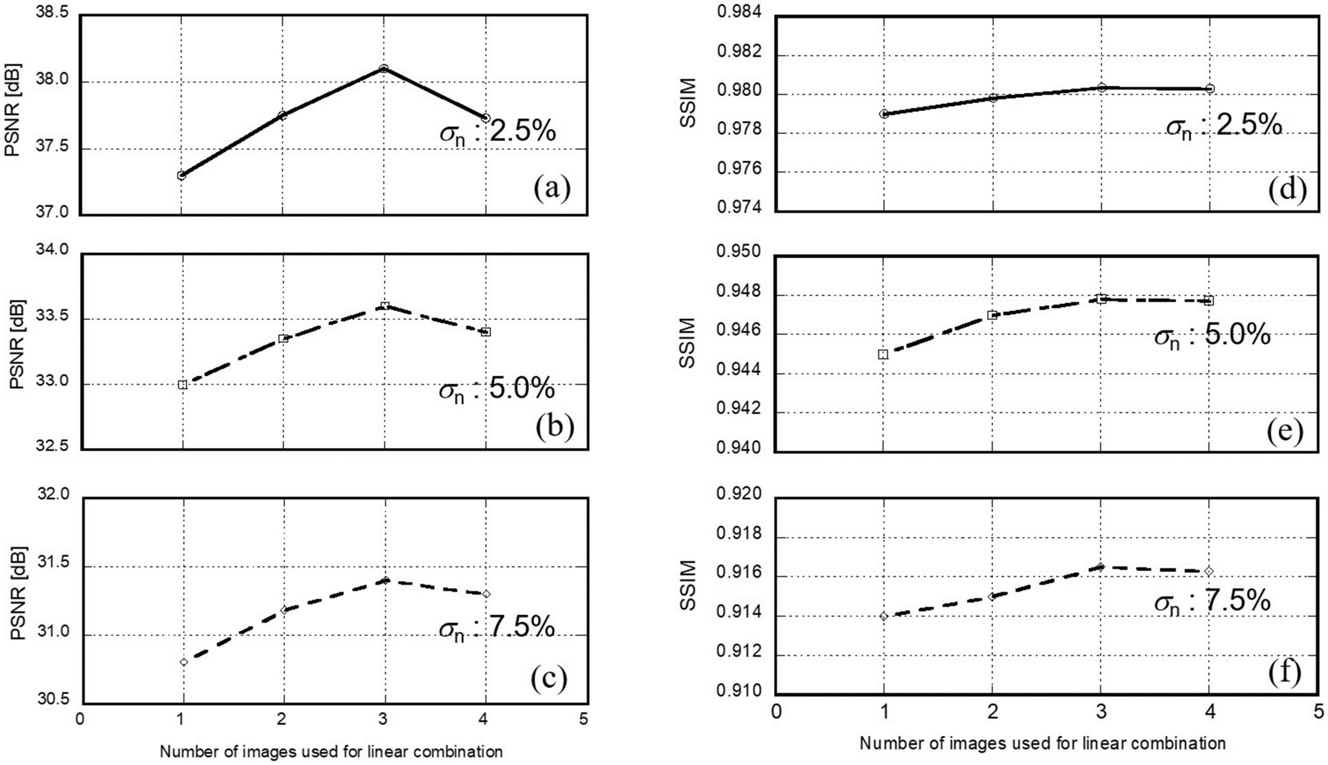


Fig. 9 Results of PSNR and SSIM evaluation with ParBID. PSNR results for 2.5%, 5.0% and 7.5% noise levels are shown in (**a**), (**b**), (**c**) and SSIM results for 2.5%, 5.0% and 7.5% noise levels are shown in (**d**), (**e**), (**f**) respectively with reference to number of images used for linear combination. ParBID, denoising convolutional neural network; PSNR, peak SNR; SSIM, structural similarity index.

difference image between 3-slice ParBID (l) and 2-slice ParBID (k), respectively. An outline of the image appears to have been extracted in subimages (m) and (n). These images indicate that details of images are preserved to a much higher degree in 3-slice ParBID.

Discussion

A preliminary study on the appropriate network depth d and the size of the receptive field was carried out while varying the input noise level between 2.5% and 7.5%. The highest performance was obtained for a receptive field size of 35 with network depth $d = 17$ for all SNRs. Therefore, we used these parameters in our examination. Next, we examined the validity of the number of epochs and learning rates used in training. The PSNR and the SSIM increase rapidly until 10 epochs, and the change in these values becomes smaller thereafter. Therefore, we used 20 epochs throughout the present paper.

In order to accelerate the training process of the CNN, several normalization techniques have been proposed. These normalizations alleviate the problem of internal covariance shift and can improve the efficiency of the backpropagation optimization methods significantly. Swish is a monotonically increasing function for a positive input, similar to ReLU, but has negative output in the region for a negative input. Klambauer et al. showed that if an AF has monotonically increasing response for a positive input and has negative response for a negative input, and if the input to the AF follows a Gaussian distribution with mean and variance floating around 0 and 1, then the mean and variance of the output tend to approach 0 and 1 for certain weights.[24] The bump-shaped function of Swish is properly scaled to be able to push the output towards zero mean statistically while having an approximately zero response to a large negative input. This self-normalizing property simultaneously addresses the problems of covariate shift and vanishing of the gradient between layers. We used Swish instead of ReLU in the DnCNN and applied the method to the denoising problem of MR images. It was shown that the PSNR and the SSIM are improved slightly for all noise levels from 2.5% to 7.5%. The denoised images shown in Fig. 5 reveal that image contrast remained to a greater degree in Swish. These results indicate that an AF having a bump shape with proper adjustment is more suitable for the DnCNN than ReLU, and hence has higher denoising performances.

In the training of DnCNN-B, we tried two approaches. First, we trained the CNN varying the noise level from batch to batch, where noise levels of small patches were the same in a single batch. Second, training of the CNN was executed by constructing several patches, including several noise levels as described

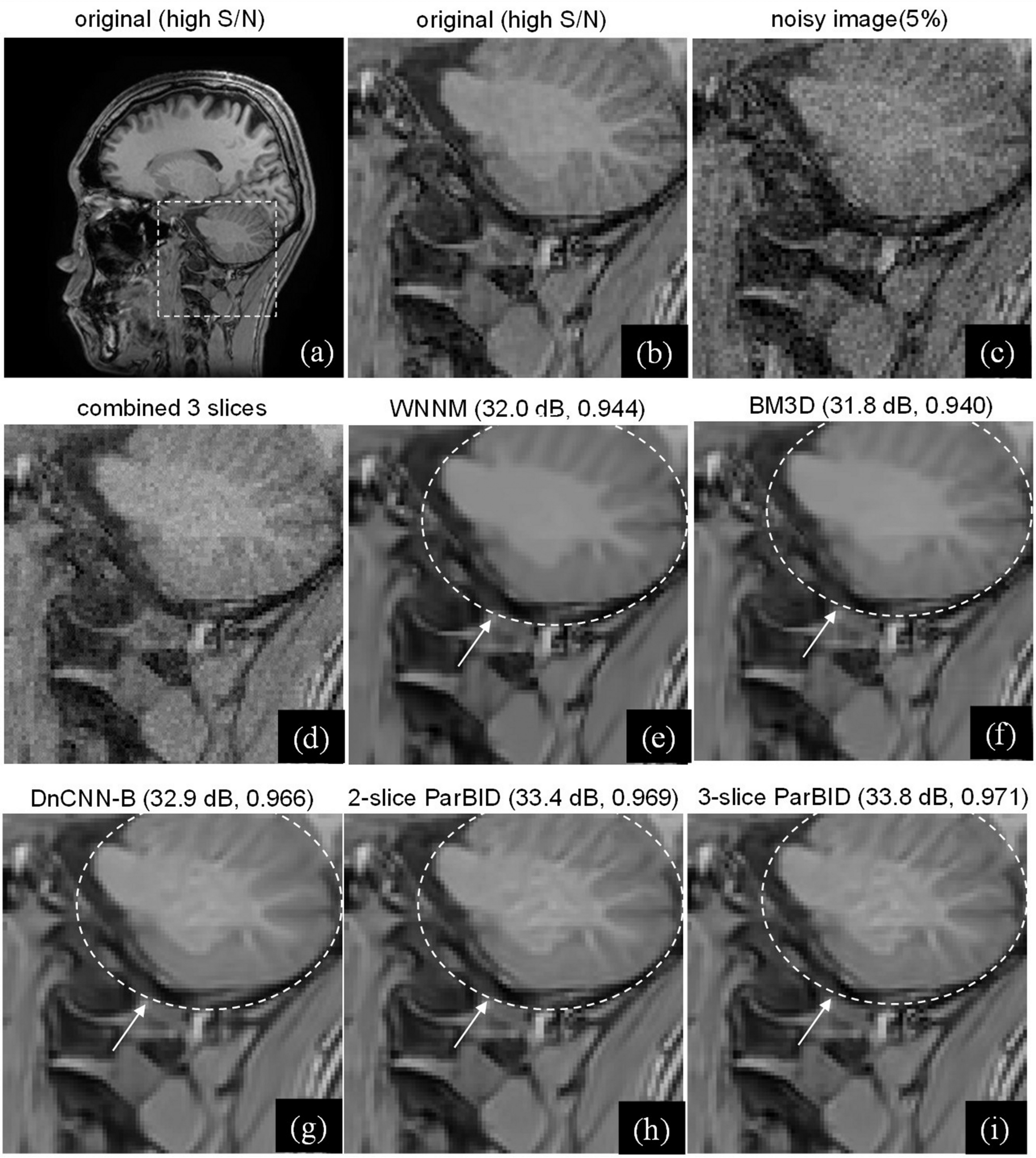


Fig. 10 Comparison of denoised images using ParBID and other methods. Subimages (**b**) and (**c**) show the images before and after adding 5.0% noise inside the box region of original high S/N image (**a**). Subimages (**d**)–(**i**) show the linear combination of adjacent 3 images and denoised images using WNNM, BM3D, single-slice DnCNN-B, and 2- and 3-slice ParBID, respectively. Obtained PSNRs and SSIMs are given with the denoising method in the subimages (**e**)–(**i**). BM3D, block-matching and 3D; DnCNN-B, denoising convolutional neural network-blind; ParBID, denoising convolutional neural network; PSNRs, peak SNRs; SSIMs, structural similarity indexes; WNNM, weighted nuclear norm minimization.

in Method section. A simulation study showed that blind denoising was successfully obtained by the latter method and not by the former training method. This result indicates that the noise level should be varied from patch to patch, so that the CNN does not learn a particular noise level.

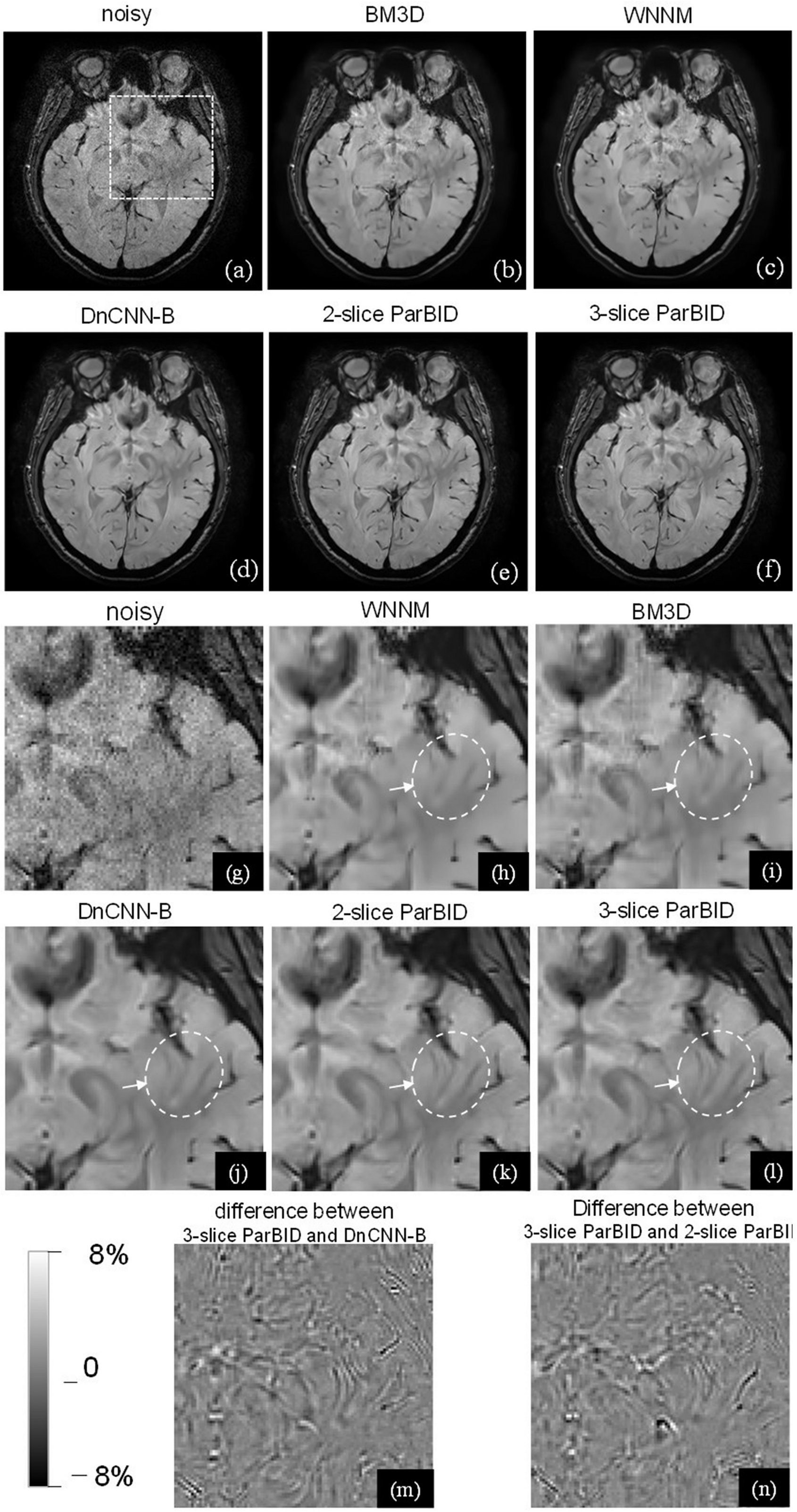


Fig. 11 Application of ParBID to noisy image obtained with MRI. Subimage (**a**) is the target noisy image. Denoised images by BM3D, WNNM, DnCNN-B, and 2- and 3-slice ParBID are shown in (**b**)–(**f**), respectively. Subimages (**g**)–(**l**) are the enlarged images of (**b**)–(**f**), respectively for the region inside the box in subimage (**a**). Subimages (**m**) and (**n**) are the difference image between 3-slice ParBID and DnCNN-B, and the image between 3-slice ParBID and 3-slice ParBID, respectively. BM3D, block-matching and 3D; DnCNN-B, denoising convolutional neural network-blind; ParBID, denoising convolutional neural network; WNNM, weighted nuclear norm minimization.

According to the results shown in Fig. 6, in which DnCNN-B was applied to several noise-level images, comparative PSNRs with the DnCNN(S) were obtained for each noise level. As shown in Fig. 6, the highest PSNRs were obtained in the DnCNN(S) trained with a single noise level, and comparative PSNRs were obtained in DnCNN-B. Figure 6 shows the feasibility and adaptivity of the DnCNN-B for images in which the noise level varied spatially. These results indicate the effectiveness of blind noise removal in MR images containing spatially variant noises.

It was shown in the examination of 3-slice ParBID that the PSNR improvements were slightly reduced at both end slices, p_1 and p_3, as compared to the middle slice, p_2. This can be attributed to the fact that the image similarity was reduced because there are two slice intervals between one end and the other end, like p_1–p_3 in 3-slice ParBID. In contrast, there is only one slice interval for both directions in the middle of the three slices p_2–p_1 and p_2–p_3. Therefore, greater PSNR improvements were obtained in the middle slice. Similar to 3-slice ParBID, greater PSNR improvements were obtained in middle slices in 4-slice ParBID. However, the improvements are slightly smaller than those of the 3-slice ParBID, as shown in Fig. 9. This is also attributed to the similarity of the image distribution. Consider the slice intervals from middle p_2 to other intervals in 4-slice ParBID. There is one slice interval between p_2–p_1 and p_2–p_3, and two slice intervals between p_2 and p_4. The similarity of the image distribution will be reduced when the slice interval is greater than one. Therefore, the PSNR of 4-slice ParBID becomes slightly smaller than that of the middle slice in 3-slice ParBID. These results also suggest that we should not divide the multi-slice images into several blocks and applying the ParBID on a block-by-block basis, since denoising was limited to images within a block that may not have adjacent images at the end of image blocks. Based on these results, we conclude that three-slice ParBID is the best in our examination, and the middle slice p_2 should be adopted in order to obtain the best PSNR improvement. The best slice number for ParBID may vary depending on the slice spacing. If the slice spacing is small, then the best number for combining slice images may vary and may be greater than 3-slice ParBID. The edges, contours, and contrast of images are restored by the application of ParBID, even though the improvements in the PSNR and the SSIM by 3-slice ParBID are rather small in the numerical evaluation. The algorithm of ParBID is also effective for other denoising methods as well, such as BM3D and WNNM, and will improve the denoising performances compared to single-slice denoising. However, DnCNN-B is effective for the MR images with unknown and space-variant noise levels.

Figure 9 shows that improvements in the PSNR are greater at a noise level of 2.5%, as compared to other noise levels. The PSNR is calculated using the difference between the denoised image and the ideal high S/N image, which is the image degradation due to the denoising process. If the amplitude variation of the imaging subject is partially compared to that of the noise, it is difficult to distinguish whether the pattern is derived from noise or the structure of the imaging subject in a single image. Let the image degradation due to single DnCNN be ΔD and let the restoration of the image by ParBID be Δd, then the SNRs by the DnCNN-B and ParBID can be roughly estimated as max $[\rho_o]/\Delta D$ and max $[\rho_o]/(\Delta D - \Delta d)$, respectively. The amount of ΔD increases with the level of superimposed noise and becomes much greater than Δd at 7.5%. Therefore, the PSNR improvements related to the difference of max $[\rho_o]/(\Delta D - \Delta d)$ and max $[\rho_o]/\Delta D$ will be small compared to smaller-noise cases, such as 5.0% or 2.5%. In contrast, with regard to the SSIM, there was no significant change between noise levels, as shown in Fig. 9B. According to the results shown in Fig. 11, in which DnCNN-B and ParBID are applied to an experimentally obtained MR image, the proposed method is effective and provides an improved contrast image while preserving the fine tissue structure.

There are some limitations to the present study. First, the DnCNN assumes Gaussian distribution noise with zero mean. When the SNR of the obtained image is small, the Rician distribution of magnitude images cannot be approximated by a Gaussian distribution. The experimental results showed that the proposed method is effective for images with 7.5% noise. In the case of denoising images containing much more noise, we can consider applying the DnCNN to real and imaginary parts of the complex MR image before calculating the magnitude images, since the noises are assumed to be more Gaussian. Second, since ParBID assumes the similarity of the distribution of MR images used for averaging, the effect of ParBID will be reduced when the similarity of adjacent sliced images is small, such as when the subject is moving or the slice interval is thick. Third, since ParBID assumes zero or very small correlation between the noises on adjacent sliced images, the benefit of ParBID will be suppressed when there are correlations between sliced images. Noise on the MR images may have correlations between sliced images depending on the imaging sequences or image reconstruction algorithm. For example, sliced 2D images obtained by the 3D Fourier transform imaging technique sometimes have small correlation between adjacent sliced images due to bandwidth limitations in signal acquisition or a kernel-based 3D filtering operation. In the present study, although the IXI dataset was regarded as noise-free data, but it contains a small amount of noise. Denoising performance may be further improved by using reduced noise MR images.

Conclusion

The results of the present study suggest that Swish can improve the denoising performance of the DnCNN compared to the ReLU function. Simulation studies showed that the DnCNN can remove noise from noisy images, where the noise level was blind and varied spatially on images. It was also suggested that parallelized image denoising using blind denoising has the possibility to further improve the denoising performance of the DnCNN using linear operations of images.

Acknowledgments

The present study was supported in part by JSPS KAKENHI 19K04423 and KAYAMORI Foundation of Informational Science Advancement. The authors would like to thank Canon Medical Systems Corp. for the use of clinical magnetic resonance images and brain-development org. for the use of the IXI Dataset.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

1. Rudin LI, Osher S, Fatemi E. Nonlinear total variation based noise removal algorithms. Physica D 1992; 60:259–268.
2. Bredies K, Kunisch K, Pock T. Total generalized variation. SIAM J Imaging Sci 2010; 3:492–526.
3. Gerig G, Kubler O, Kikinis R, et al. Nonlinear anisotropic filtering of MRI data. IEEE Trans Med Imaging 1992; 11:221–232.
4. Buades A, Bartomeu C, Morel JM. A non-local algorithm for image denoising, Proceedings of 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Diego, 2005; 2:60–65.
5. Dabov K, Foi A, Katkovnik V, et al. Image denoising by sparse 3-D transform-domain collaborative filtering. IEEE Trans Image Process 2007; 16:2080–2095.
6. Gu S, Xie Q, Meng D, et al. Weighted nuclear norm minimization and its applications to low level vision. Int J Comput Vis 2017; 121:183–208.
7. Knaus C, Zwicker M. Dual-domain image denoising. Proceedings of 2013 IEEE International Conference on Image Processing, Melbourne, 2013; 440–444.
8. He K, Zhang X, Ren S, et al. Deep residual learning for image recognition. Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, 2016; 770–778.
9. Zhang K, Zuo W, Chen Y, et al. Beyond a Gaussian denoiser: residual learning of deep CNN for image denoising. IEEE Trans Image Process 2017; 26:3142–3155.
10. Manjón JV, Coupé P. MRI denoising using deep learning. Patch-Based Techniques in Medical Imaging: 4th International Workshop, Patch-MI 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 20, 2018. Proceedings. doi: 10.1007/978-3-030-00500-9_2.
11. Jiang D, Dou W, Vosters L, Xu X, Sun Y, Tan T. Denoising of 3D magnetic resonance images with multi-channel residual learning of convolutional neural network. Jpn J Radiol 2018; 36:566–574.
12. Perona P, Malik J. Scale-space and edge detection using anisotropic diffusion. IEEE Trans Pattern Anal Mach Intell 1990; 12:629–639.
13. Latif G, Iskandar DA, Alghazo J, et al. Deep CNN based MR image denoising for tumor segmentation using watershed transform. Int J Eng Technol 2018; 7:37–42.
14. Isogawa K, Ida T, Shiodera T, et al. Deep shrinkage convolutional neural network for adaptive noise reduction. IEEE Signal Process Lett 2018; 25:224–228.
15. Ramachandran P, Zoph B, Le QV. Searching for activation functions. arXiv:1710.05941, 2017.
16. Kidoh M, Shinoda K, Kitajima M, et al. Deep learning based noise reduction for brain mr imaging: tests on phantoms and healthy volunteers. Magn Reson Med Sci 2020; 19:195–206.
17. Pruessmann KP, Weiger M, Scheidegger MB, Boesiger P. SENSE: sensitivity encoding for fast MRI. Magn Reson Med 1999; 42:952–962.
18. Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556, 2015.
19. Kingma DP, Ba JL. Adam: a method for stochastic optimization. arXiv:1412.6980, 2017.
20. Gudbjartsson H, Patz S. The Rician distribution of noisy MRI data. Magn Reson Med 1995; 34:910–914.
21. Imperial College London. IXI dataset. https://brain-development.org/ixi-dataset/. (Accessed: Apr 20, 2020)
22. Wang Z, Bovik AC, Sheikh HR, et al. Image quality assessment: from error visibility to structural similarity. IEEE Trans Image Process 2004; 13:600–612.
23. Copyright The MatConvNet Team. MatConvNet: CNNs for MATLAB. http://www.vlfeat.org/matconvnet/. (Accessed: Apr 20, 2020)
24. Klambauer G, Unterthiner T, Mayr A, et al. Self-normalizing neural networks. Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS), Long Beach, 2017; 972–981.