

## Research Article

# Research on Optimization Scheme for Blocking Artifacts after Patch-Based Medical Image Reconstruction

Yan Xu <sup>1,2</sup>, Shunbo Hu <sup>2</sup>, and Yuyue Du <sup>1</sup>

<sup>1</sup>College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao 266590, China

<sup>2</sup>College of Information Science and Engineering Linyi University, Shandong Province 276005, China

Correspondence should be addressed to Shunbo Hu; [hushunbo@lyu.edu.cn](mailto:hushunbo@lyu.edu.cn) and Yuyue Du; [yydu001@163.com](mailto:yydu001@163.com)

Received 9 June 2022; Accepted 12 July 2022; Published 31 July 2022

Academic Editor: Lin Lu

Copyright © 2022 Yan Xu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Due to limitations of computer resources, when utilizing a neural network to process an image with a high resolution, the typical processing approach is to slice the original image. However, because of the influence of zero-padding in the edge component during the convolution process, the central part of the patch often has more accurate feature information than the edge part, resulting in image blocking artifacts after patch stitching. We studied this problem in this paper and proposed a fusion method that assigns a weight to each pixel in a patch using a truncated Gaussian function as the weighting function. In this method, we used the weighting function to transform the Euclidean-distance between a point in the overlapping part and the central point of the patch where the point was located into a weight coefficient. With increasing distance, the value of the weight coefficient decreased. Finally, the reconstructed image was obtained by weighting. We employed the bias correction model to evaluate our method on the simulated database BrainWeb and the real dataset HCP (Human Connectome Project). The results show that the proposed method is capable of effectively removing blocking artifacts and obtaining a smoother bias field. To verify the effectiveness of our algorithm, we employed a denoising model to test it on the IXI-Guys human dataset. Qualitative and quantitative evaluations of both models show that the fusion method proposed in this paper can effectively remove blocking artifacts and demonstrates superior performance compared to five commonly available and state-of-the-art fusion methods.

## 1. Introduction

In recent years, with the development of deep learning, medical imaging has gradually become one of the most promising fields of artificial intelligence. Especially in the aspects of localization and detection, recognition and classification, lesion segmentation, registration, and fusion in medical images, deep learning algorithms have played a crucial role in assisting doctors to diagnose accurately and efficiently [1]. Medical images have the characteristics of large image pixels [2], when using neural networks to process them, due to the limitations of computer video cards and video memory, directly using the original image as an input will often lead to insufficient graphics memory. The usual processing method involves slicing the original image, reconstructing the image patches, and then splicing the image patches. In deep learning, patch-based training methods

have been widely used because they can save GPU memory, are not affected by insufficient training data, and can obtain better local performance than other methods [3]. However, there is also a grid artifact phenomenon at the edge of patches when they are spliced for image reconstruction as shown in Figure 1 [3], these artifacts are called image blocking artifacts.

To solve this problem, Yang et al. [4] and Hu et al. [5] obtained the final predicted deformation field by averaging the overlapping regions of patches when reconstructing the deformation field in medical image registration. In brain image registration using dual-supervised fully convolutional networks, as proposed by Fan et al. [6], only the deformation field in the central region of the patch is estimated, the input patch size is  $64 \times 64 \times 64$ , and the output DDF (dense displacement field) is  $24 \times 24 \times 24$ . The MINScnn model proposed by Müller and Kramer [7] used the average fusion

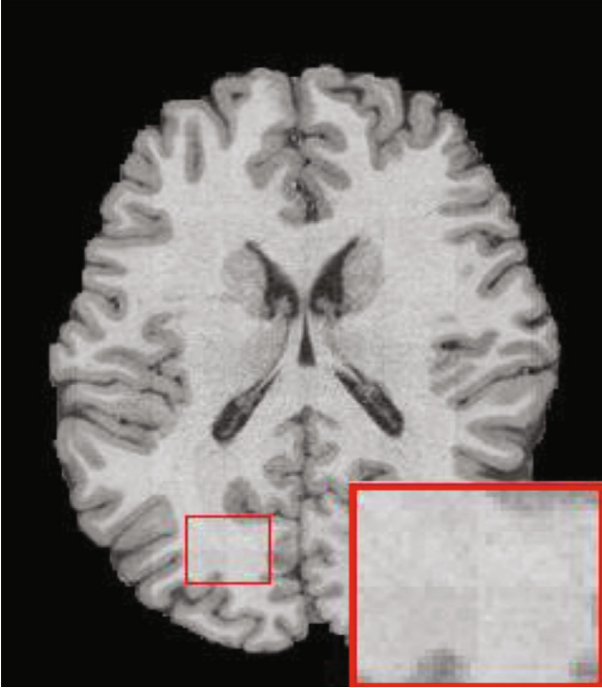


FIGURE 1: Image blocking artifacts after patch-based image reconstruction.

method for the two overlapping patches in the segmentation field. In the patchify method proposed by Wu, the overlapping region is covered with the next patch [8]. The EDW (Exponential-Distance-Weighted) method proposed by Wu et al. [9] uses an exponential function to convert the distance between the point of the overlapping area and the center point of the patch into a weight coefficient and reconstruct the predicted deformation field by weighting. This method achieved good results.

During image reconstruction, most of the above methods reduce image blocking artifacts to a certain extent by reducing the step size or by performing estimates for the central area of the patch, but blocking artifacts are still obvious after image reconstruction. Considering the uncertainty of patch edge region prediction, we propose to use the truncated Gaussian function as a weighting function to convert the Euclidean distance from the points in the overlapping regions to the center point of the patch into a weight coefficient. The value of the weight coefficient will decrease with increasing distance, reducing the predicted value in the patch edge region. Finally, the reconstructed image will be obtained by weighted fusion. For distance calculation, we choose Euclidean distance as the distance measure [9]. This method can be applied to any patch-based deep learning network without modifying the network structure and loss function. Compared with the image reconstruction methods used in the above literature, the patch-based fusion method in this paper shows better performance in both qualitative and quantitative evaluations.

The paper is organized as follows. In Section 2, we briefly review the popularly available methods for extracting

patches in deep learning. Section 3 describes the proposed patch fusion method in detail. In Section 4, we evaluate the proposed patch fusion algorithm on synthetic and real datasets and present the experimental results. At the same time, a comparison with other state-of-the-art methods is also given. Conclusions and future works are discussed in Section 5.

## 2. Common Methods for Extracting Patches

Nonoverlapping slicing is performed directly on the image, as shown in Figure 2(a). Assuming that the image size is  $512 \times 512$ , the image is sliced into four equal parts, and the size of each patch is  $256 \times 256$ . The patches are input to the network to obtain the reconstructed image patches, and then, they are stitched together. Because the boundary information of each patch is inconsistent after reconstruction, the stitched image has obvious blocking artifacts.

In overlapping slicing, the image is sliced into patches, so that there are overlapping areas between the patches, as shown in Figure 2(b). The red region is the region where the two patches overlap. Assuming that the image size is still  $512 \times 512$  and the red overlapping region is 8 pixels wide, then the size of the input patch is  $260 \times 260$ . After reconstruction, in situ pixel stitching is performed, and the red overlapping part is taken as the average value after stitching. This scheme has a better effect in removing the blocking artifact. However, since the average weighting method is adopted for the overlapping region, the weight of the pixels in the overlapping region is the same no matter how far away from the center point, so grid artifacts appear.

When padding is used around the image block, a patch is obtained as shown in block 1 in Figure 2(c). First, the padding is made around the image, the image is sliced after padding, and the cyan line is the central axis of the image. For example, in block 1, the orange part is the size of the original image block, and the surrounding red area consists of the expanded pixels. If the original image block is  $256 \times 256$  and the red area occupies a width of 8 pixels, the input patch size is  $272 \times 272$  ( $256 + 8 * 2$ ) for reconstruction. Then, the reconstructed image block is cut with a width of 8 pixels around to obtain a patch with a size of  $256 \times 256$ , and finally, it is stitched in sequence. For image block 1, the red part is removed after reconstruction. For image block 2, the blue part on the left needs to be cut off. This scheme is better than the previous two schemes in removing blocking artifacts, but it will be affected by stride. The smaller the stride is, the better the effect of removing blocking artifacts. However, the smaller stride, the greater the amount of calculations. Since the weight of the entire estimated central region is the same, the image blocking artifacts will still occur.

## 3. Methods and Theories

3.1. *One-Dimensional Gaussian Distribution.* The one-dimensional Gaussian function has the following form [10]:

$$f(x) = ae^{-(x-b)^2/2c^2}, \quad (1)$$

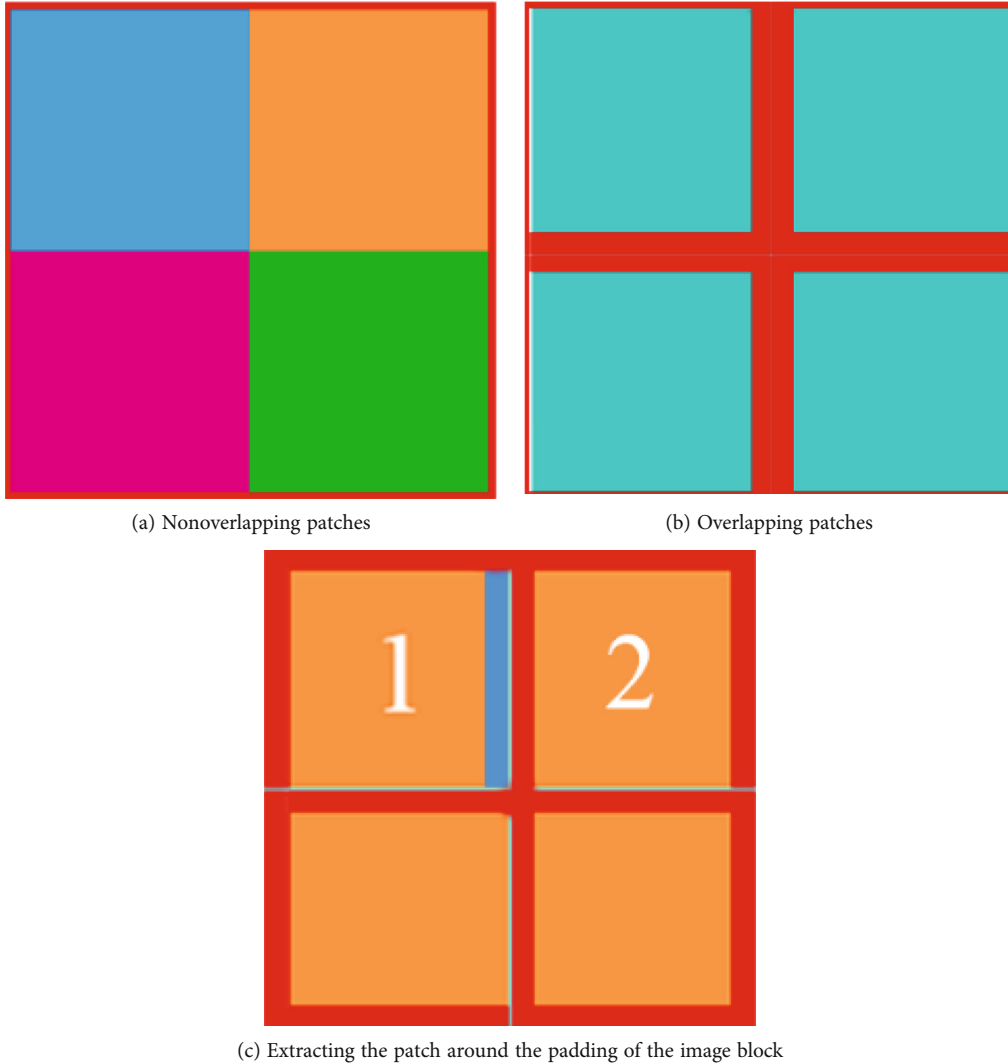


FIGURE 2: Three methods for extracting patches of images.

where  $a, b$ , and  $c$  are arbitrary real numbers. The Gaussian curve graph is a characteristically symmetrical “bell curve” shape, where  $a$  represents the height of the peak of the curve,  $b$  is the position of the center of the peak, and  $c$  is called the standard variance, which is a parameter used to control the width of the “bell.” Let  $a = 1, b = \mu$  (mean value) and  $c = \sigma$  (standard variance); the Gaussian distribution can be obtained as follows:

$$G(x) = e^{-(x-\mu)^2/2\sigma^2}. \tag{2}$$

Its function curve is shown in Figure 3.

The horizontal axis represents the possible values, the vertical axis represents the probability distribution density  $G(x)$ , the expected value  $\mu$  of the normal distribution determines the center of the curve, and the standard variance  $\sigma$  determines the magnitude of the distribution. In Figure 3, the settings are  $a = 1, b = \mu = 0$ . Hence, the peak heights and center positions of all curves are the same. With increas-

ing standard variance, the graph becomes wider, the distribution becomes dispersed, and the area of the graph becomes increasingly larger. The closer the value of  $x$  is to the center, the greater the value of  $G(x)$ . Therefore, we can use this feature of the Gaussian function by assigning the distance from the pixel’s location to the center point of the patch as the input  $x$  to the Gaussian function and calculating the weight coefficient of each pixel point through Gaussian function [11].

**3.2. The Proposed Patch Fusion Method.** In Figure 4, the blue part is patch 1,  $B$  is the central point of patch 1, the orange part is patch 2,  $C$  is the central point of patch 2, and the red part is the overlapping part of these two patches. It can be seen that pixel  $A$  is closer to the central point  $B$  in patch 1 in the overlapping area, so its predicted value in patch 1 is more accurate. To reduce the weight of the predicted value of the pixel located at the boundary, we convert the Euclidean distance between the pixel and the center of the patch to the weight coefficient of the

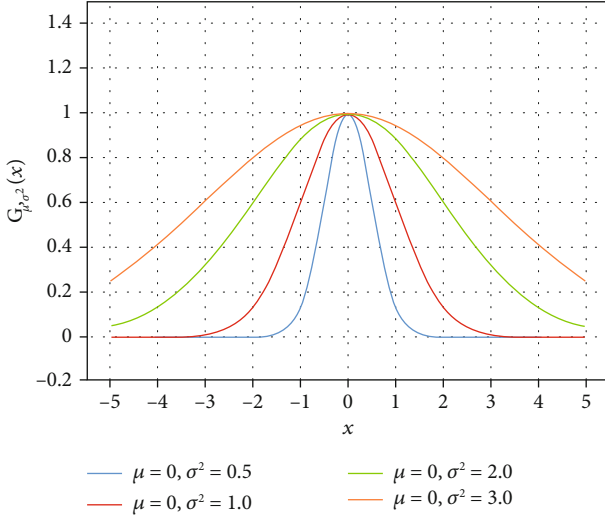


FIGURE 3: The Gaussian function curve ( $a = 1, b = \mu, c = \sigma$ ).

pixel through the Gaussian truncation function. The formula is defined as follows [11, 12]:

$$W(y) = \begin{cases} \frac{1}{z} e^{-|y|^2/2\sigma^2}, & \text{for } |y| \leq \rho, \\ 0, & \text{else,} \end{cases} \quad (3)$$

where  $\sigma$  represents the standard variance,  $z$  represents the normalization factor of the normalized Gaussian kernel, and  $\rho$  represents the radius to measure the size of the local region. Therefore, in Figure 4, assuming that the predicted value of point  $A$  in patch 1 is  $P_{AB}$  and the predicted value in patch 2 is  $P_{AC}$ , the final predicted value of point  $A$  is  $P_A = W(y1)P_{AB} + W(y2)P_{AC}$ .

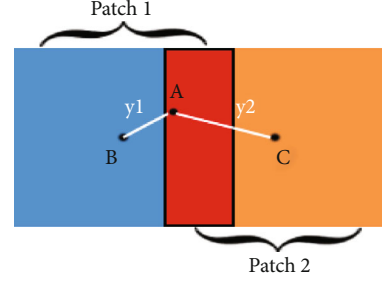


FIGURE 4: Schematic diagram of patch overlap. The red part is the overlapping area, and  $B$  and  $C$  are the center points of patch 1 and patch 2, respectively.  $A$  is a point in the overlapping area, and  $y1$  and  $y2$  are the distances from point  $A$  to the two center points.

We take a two-dimensional graph as an example. Assuming that the image size is  $M \times N$ , the patch size is  $h \times w$ , and the stride size is  $s$ ,  $n$  patches can be generated, and the value of  $n$  can be calculated by

$$n = \left\lfloor \frac{M - (h - s)}{s} \right\rfloor \left\lfloor \frac{N - (w - s)}{s} \right\rfloor. \quad (4)$$

In these  $n$  patches, the Gaussian distribution of the distance between each pixel point and the center point of the patch is the same, which is calculated by Formula (2). The Gaussian function value corresponding to the pixel point  $(i, j)$  in the  $k$ th patch can be recorded as  $G_k^{(i,j)}(y_k)$ , where  $y_k$  is the Euclidean distance from this point to the central point of the patch. Therefore, Formula (3) can be expressed by  $G_k^{(i,j)}(y_k)$ .

$$W(y_k) = \begin{cases} \frac{1}{z} G_k^{(i,j)}(y_k), & \text{for } |y_k| \leq \rho, \\ 0, & \text{else,} \end{cases} \quad (5)$$

$$m_i = \begin{cases} \left\lfloor \frac{i}{s} \right\rfloor + 1, & i < h, \\ \frac{h}{s}, & h \leq i \leq \left\lfloor \frac{M - (h - s)}{s} \right\rfloor s - 1, \\ \left\lfloor \frac{((\lfloor (M + s)/s \rfloor - 1)s - i)}{s} \right\rfloor, & \left\lfloor \frac{M - (h - s)}{s} \right\rfloor s - 1 < i < \left( \left\lfloor \frac{M + s}{s} \right\rfloor - 1 \right) s, \\ 0, & \left( \left\lfloor \frac{M + s}{s} \right\rfloor - 1 \right) s \leq i. \end{cases} \quad (6)$$

Assuming that the pixel point  $O(i, j)$  overlaps  $m$  times,  $m = m_i \times m_j$ ,  $i \in [0, M - 1]$ ,  $j \in [0, N - 1]$ , then  $m_i$  can be calculated by Formula (6) [9], and the edge part that cannot

be obtained by the patch is discarded. The value of  $m_j$  can be obtained by replacing  $M$  in Formula (6) with  $N$ . Assuming that the  $m$  predicted values of the pixel point are  $P^{(i,j)}$

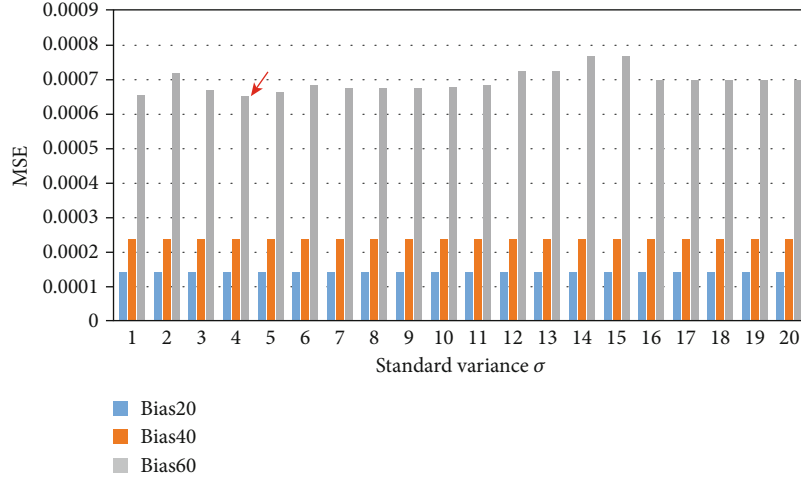


FIGURE 5: MSE values of different standard variance on different INU level datasets.

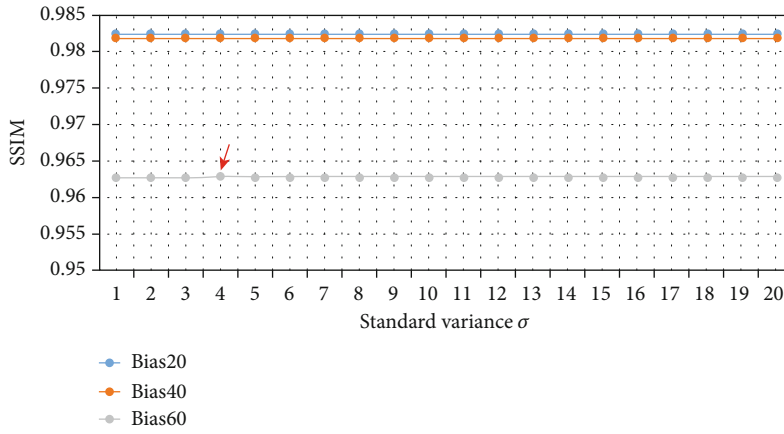


FIGURE 6: SSIM values of different standard variance on different INU level datasets.

$= \{p_1^{(i,j)}, p_2^{(i,j)}, \dots, p_m^{(i,j)}\}$  and the weight coefficients of the  $m$  predicted values calculated by Formula (5) are  $W^{(i,j)} = \{W_{y_1}^{(i,j)}, W_{y_2}^{(i,j)}, \dots, W_{y_m}^{(i,j)}\}$ , then the final predicted value of the pixel point  $O(i, j)$  is

$$\hat{P}^{(i,j)} = P^{(i,j)} \cdot W^{(i,j)} = \sum_{k=1}^m p_k^{(i,j)} W_{y_k}^{(i,j)}, \quad (7)$$

where  $\sum_{k=1}^m W_{y_k}^{(i,j)} = 1$ .

#### 4. Experimental Results and Analysis

To evaluate the effective fusion ability of the algorithm, we employed DN-RESnet (a deep convolutional neural network consisting of several residual blocks) [13] as the training model. The bias field removal experiment was carried out on the simulation database BrainWeb [14–15] and the real dataset HCP (Human Connectome Project) [16]. Meanwhile, we also used the DnCNN (De-noising Convolutional Neural Net-

work) denoising model [17] for validation on the IXI-Guys (<http://brain-development.org/ixidataset>) dataset [18].

##### 4.1. Validation of the Proposed Fusion Method in the Bias Removal Model

**4.1.1. Results Obtained on the Simulation Database BrainWeb.** In the Simulated Brain Database (SBD) [15], the size of each MRI brain image volume is  $181 \times 217 \times 181$  voxels with a resolution of  $1 \times 1 \times 1$  mm<sup>3</sup>. 2D MRI slices were extracted from the volumes for training and testing, with different intensity nonuniformity (INU) levels (20%, 40%, and 60%) on the axial plane, respectively. We used rotation, mirror imaging, translation, and other methods to expand the samples. Finally, 604 slices were obtained for training and 30 slices for testing from the volume of each INU level. In order to obtain fewer redundant regions, we cropped the edges of each slice, so the slice size becomes  $210 \times 180$ . In the training stage, we set the patch size to  $60 \times 60$ , and obtained 99 patches from each image by using a sliding window with a stride of  $15 \times 15$ . Finally, we obtained 59796 patches.

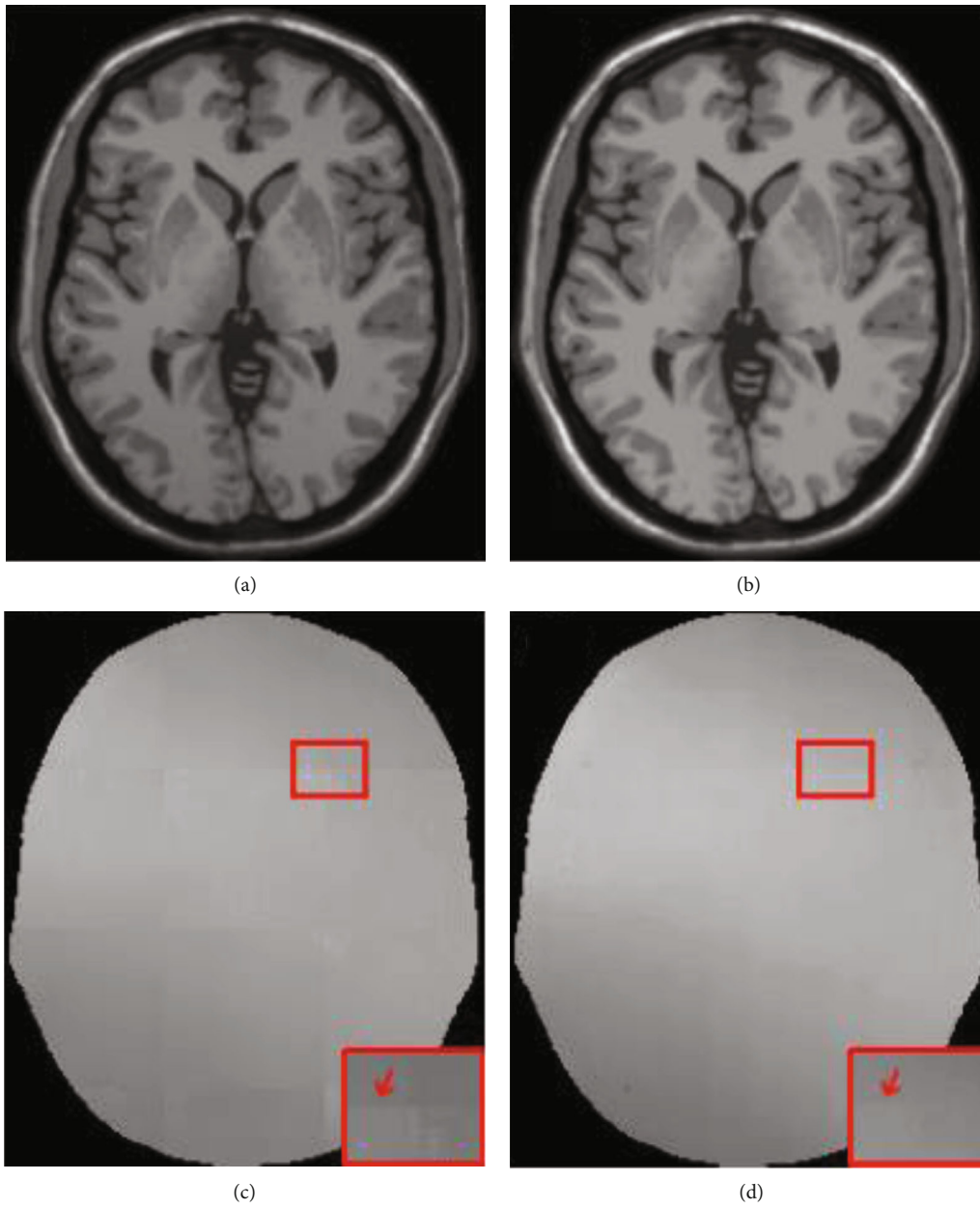


FIGURE 7: Continued.

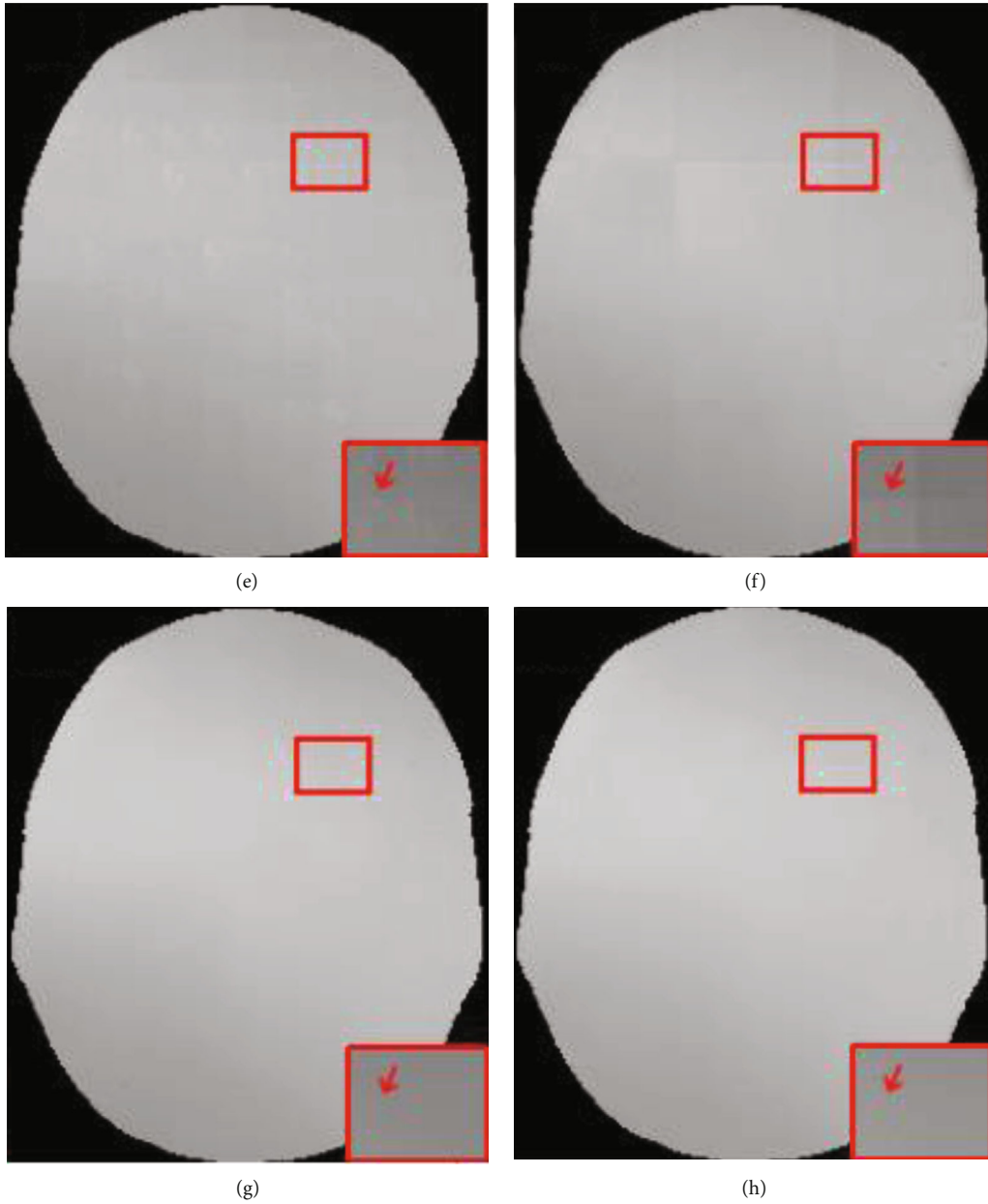


FIGURE 7: Contaminated image, real image, and bias field images obtained by different fusion methods from the BrainWeb database: (a) original image (INU = 40%, noise level = 0%); (b) real image; (c) NPS; (d) AAW; (e) MIScnn; (f) pathify; (g) EDW; (h) proposed method.

In our algorithm, we needed to set the value of the standard variance  $\sigma$ . We conducted experiments on datasets with different (INU) levels (20%, 40%, and 60%) setting the value of  $\sigma$  in the range of 1 to 20. The mean square error (MSE) and the structural similarity index (SSIM) were used as the quantitative evaluation criteria after image reconstruction. The results are shown in Figures 5 and 6.

It can be seen from Figures 5 and 6 that when the INU level is not high (20%, 40%), no matter what the value of the standard variance  $\sigma$  was, it had little effect on the MSE and SSIM after image reconstruction. However, when the INU level increased to 60%, MSE and SSIM values fluctuated for different values of  $\sigma$ . When  $\sigma = 4$ , MSE and SSIM showed better performance. At the same time, we also performed the

TABLE 1: MSE values between the real image and the corrected images obtained by different fusion methods.

Methods	n3-20	n0-40	n0-60
NPS	$13.60 \times 10^{-4}$	$2.139 \times 10^{-4}$	$8.228 \times 10^{-4}$
AAW [5]	$16.95 \times 10^{-4}$	$2.054 \times 10^{-4}$	$6.575 \times 10^{-4}$
MIScnn [7]	$48.57 \times 10^{-4}$	$2.108 \times 10^{-4}$	$7.079 \times 10^{-4}$
Pathify [8]	$13.53 \times 10^{-4}$	$2.110 \times 10^{-4}$	$7.985 \times 10^{-4}$
EDW [9]	$6.593 \times 10^{-4}$	$2.053 \times 10^{-4}$	$7.887 \times 10^{-4}$
Proposed	$4.193 \times 10^{-4}$	$2.051 \times 10^{-4}$	$6.212 \times 10^{-4}$

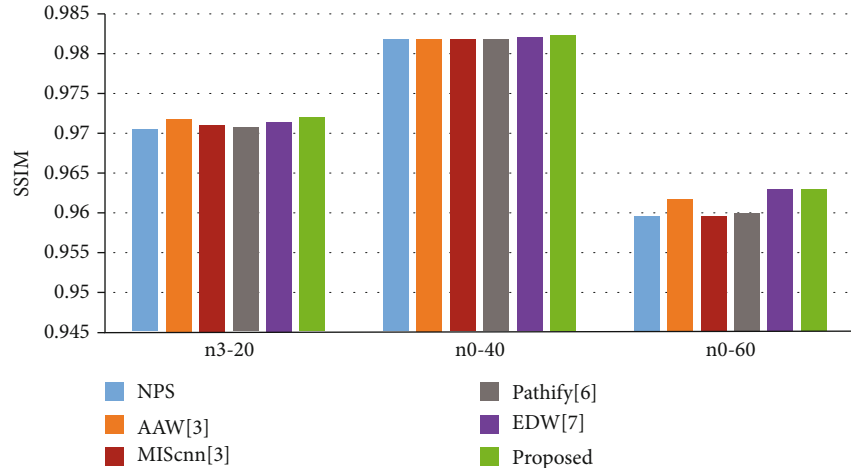


FIGURE 8: SSIM values between the real image and the corrected images obtained by different fusion methods.

same experiments on the real dataset HCP [16] and reached at consistent conclusions. Therefore, in the bias removal experiment of our algorithm, the standard variance  $\sigma$  was set to 4.

To evaluate the fusion ability of the proposed method after image reconstruction, we compared the proposed method with popularly available and state-of-the-art nonoverlapping patch splicing (NPS) method, arithmetic average weighted method (AAW) [5], MIScnn [7], Pathify [8], and Exponential-Distance-Weighted method (EDW) [9] on three datasets with different INU levels from BrainWeb. The dataset with INU = 20% and noise level = 3% was denoted as n3-20, the dataset with INU = 40% and noise level = 0% was denoted as n0-40, and the dataset with INU = 60% and noise level = 0% was denoted as n0-60. The experimental results are shown in Figure 7, Table 1, and Figure 8.

In Figure 7, we show the bias field images fused by different methods. The bias field images of NPS (Figure 7(c)) and Pathify (Figure 7(f)) demonstrate an obvious blocking artifact phenomenon, while the blocking artifact phenomena of AAW (Figure 7(d)) and MIScnn (Figure 7(e)) are greatly improved, but the seam line between patches are vaguely visible. The AAW method is greatly affected by the stride. The larger the stride is, the more obvious the blocking artifact phenomenon is. However, in Figure 7(e), the MIScnn method also shows the phenomenon of uneven bias field, indicating the inaccurate prediction of the edge region. EDW (Figure 7(g)) and our method (Figure 7(h)) consider the uncertainty of edges and assign different weights to the central region and the edge region, effectively eliminating the grid artifacts. From Table 1 and Figure 8, it is observed that on the n0-40 dataset, our method has little difference from the EDW method, but on the n3-20 and n3-60 datasets, our method shows better performance in the quantitative evaluation of MSE and SSIM than EDW, indicating that our method has better generalization ability.

*4.1.2. The Results from the Real Database HCP.* HCP [16] real human brain dataset contains T1-W and T2-W struc-

tural MR images of healthy adults from Siemens Skyra 3T scanner, along with the corresponding bias fields images. For detailed information, please refer to [19]. In the experiment, we randomly selected the T1-W structural images of 10 patients from the dataset. The size of each MRI brain image was  $260 \times 311 \times 260$  voxels, and the resolution was  $0.7 \times 0.7 \times 0.7$  mm<sup>3</sup>. We randomly selected 1000 slices from these 10 brain image volumes. To better train the model, we augmented the data with rotation and mirroring methods and finally obtained 3200 training samples and 200 testing samples. Similarly, to speed up the training process and obtain fewer redundant regions, we cropped the image edges of each sample with size  $300 \times 240$ . In the training stage, we set the patch size to  $60 \times 60$  and obtained 63 patches from each image by using a sliding window with a stride of  $30 \times 30$ . Finally, we obtained 201600 patches.

In Figure 9, we show the bias field images from HCP fused by different methods and the corrected images. It can be seen that the bias field images fused by the NPS, AAW, MIScnn, and Pathify methods have obvious seam lines at the junction between patches, which indicates that the prediction at the boundary region is not accurate. EDW and our proposed method can effectively reduce grid artifacts by assigning different weights to pixels in different regions. In the enlarged red box of the corrected image, it can be seen that there are no grid artifacts in Figures 9(g) and 9(h). Meanwhile, it can be seen from Figures 9(g) and 9(h) that the bias field obtained by our method is smoother, and the intensity of each tissue area of the corrected image looks more consistent, which is obvious in the white matter region of Figure 9(h). This conclusion is also verified in Table 2. Our method can obtain a lower MSE value and a higher SSIM value.

In the AAW method, the stride size has a great influence on the patch fusion effect. Therefore, on the HCP dataset, we verified the effect of different stride sizes on the fusion effect of AAW [5], EDW [9], and the method proposed in this paper. Table 3 lists the number of patches obtained on each image with different stride sizes and the fusion time of these three methods. At the training stage, we used a sliding window



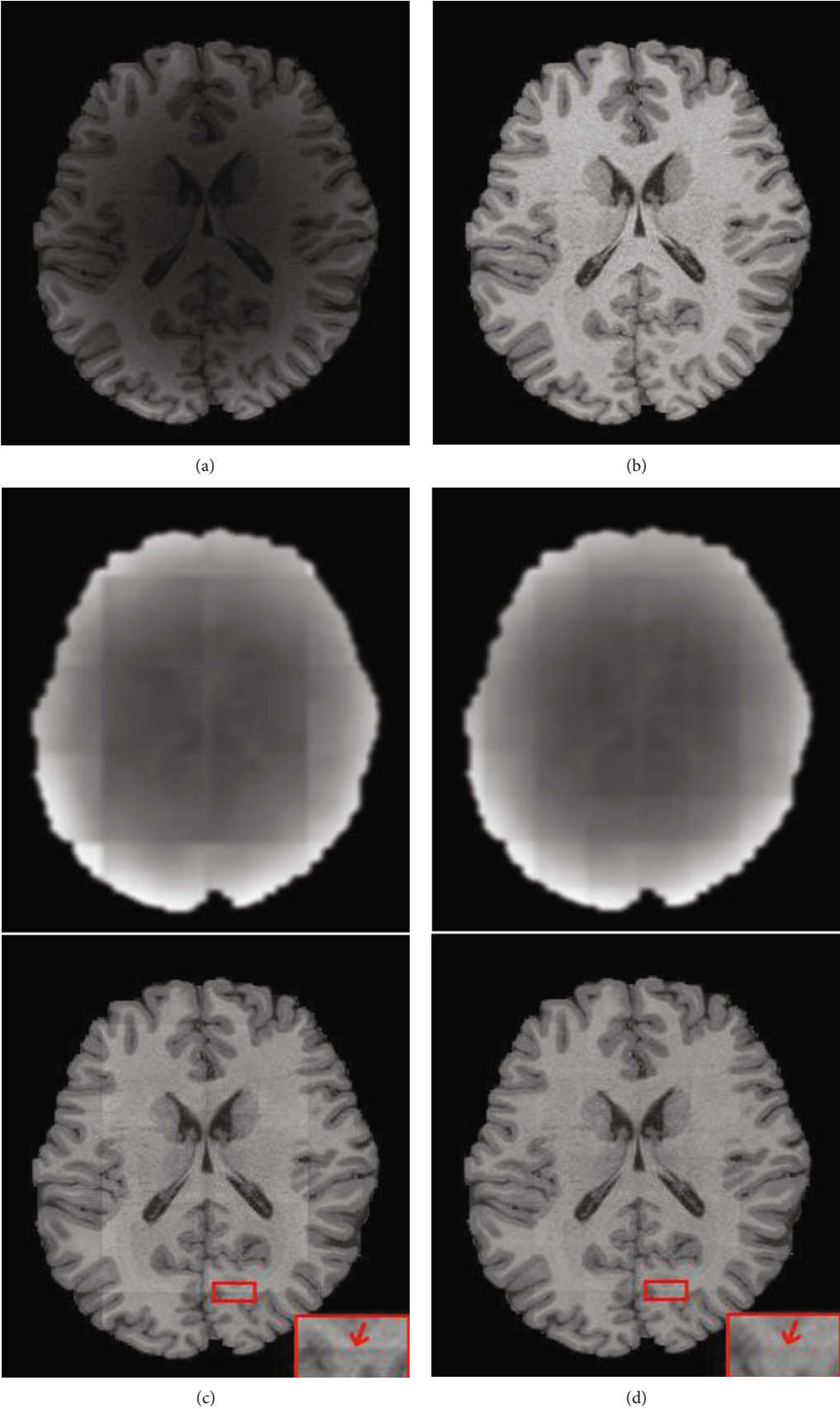


FIGURE 9: Continued.

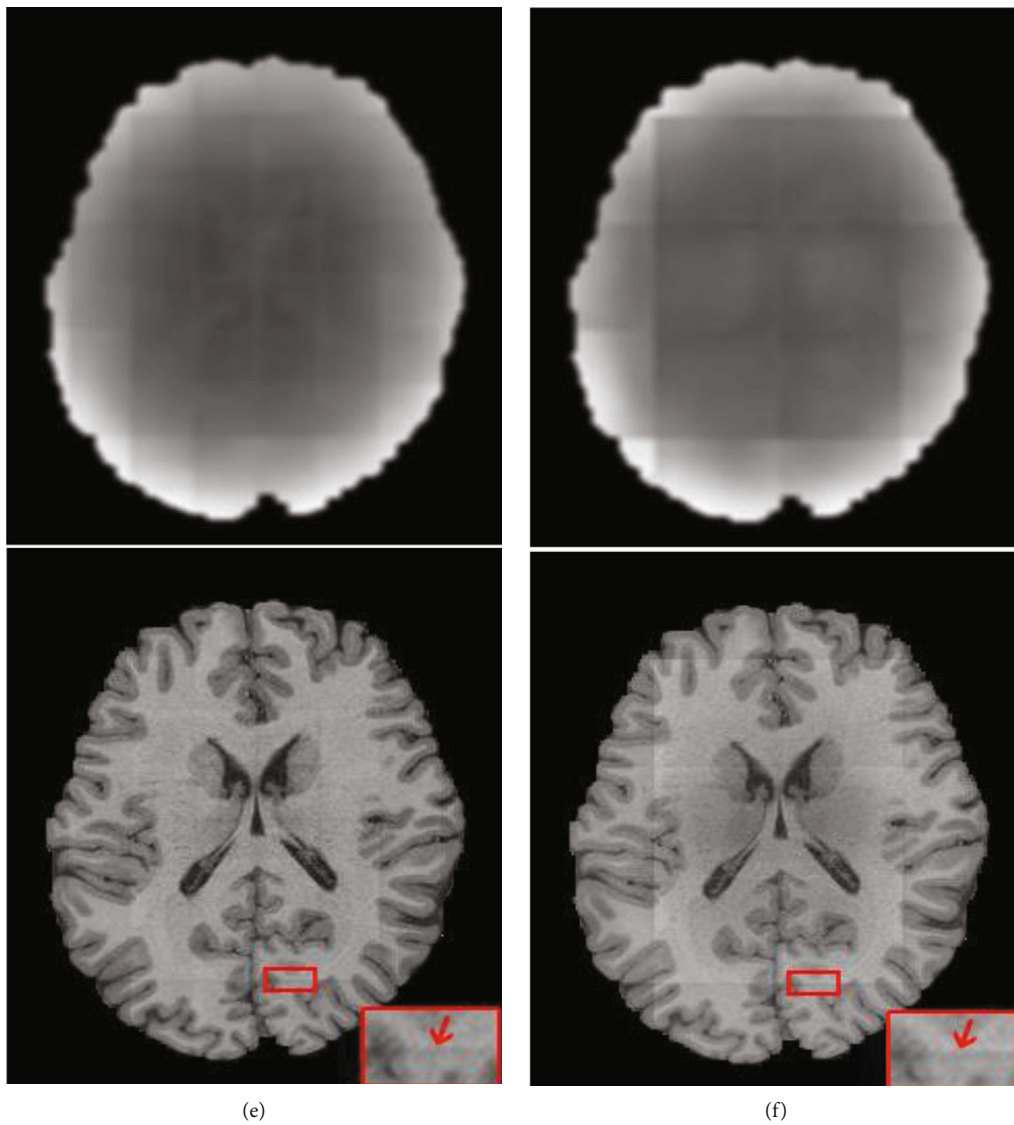


FIGURE 9: Continued.

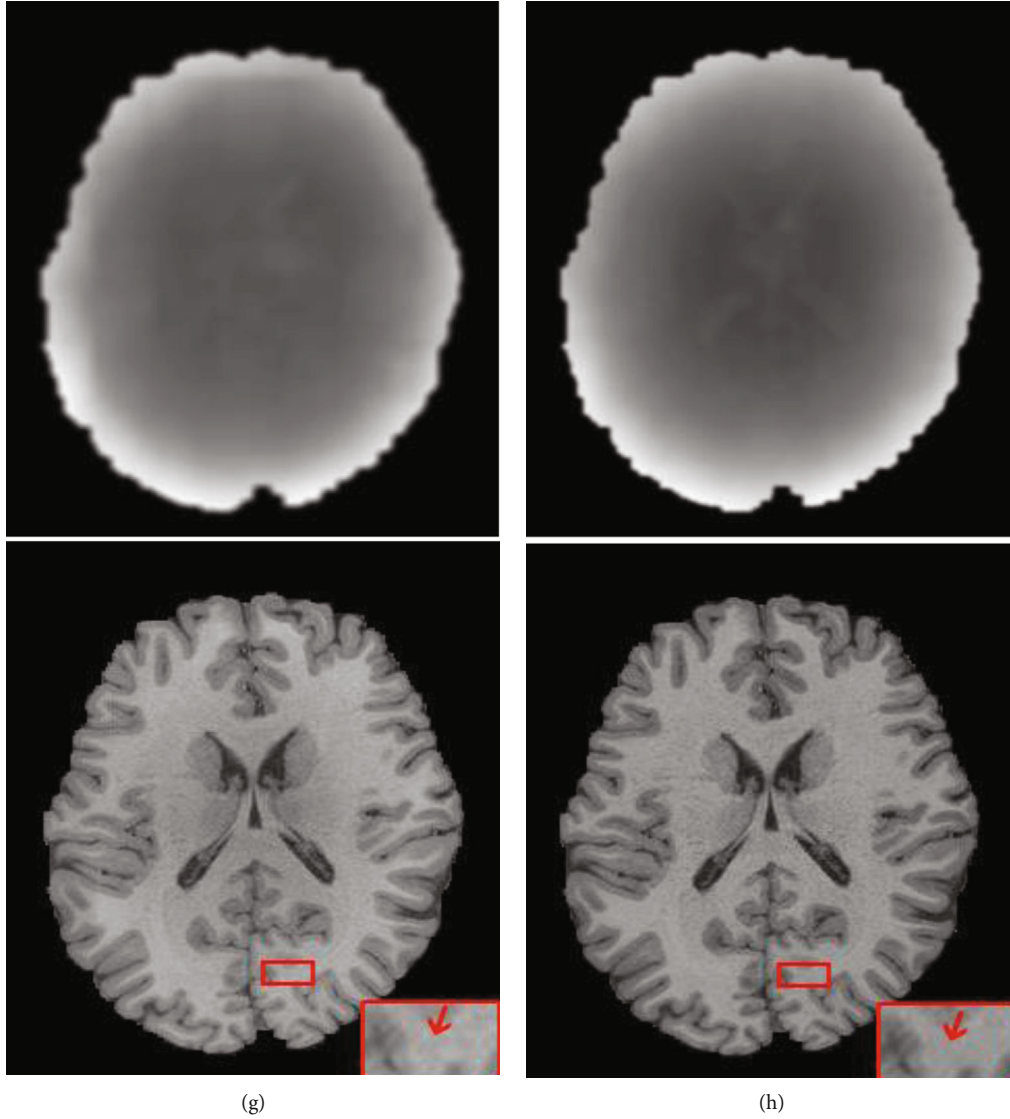


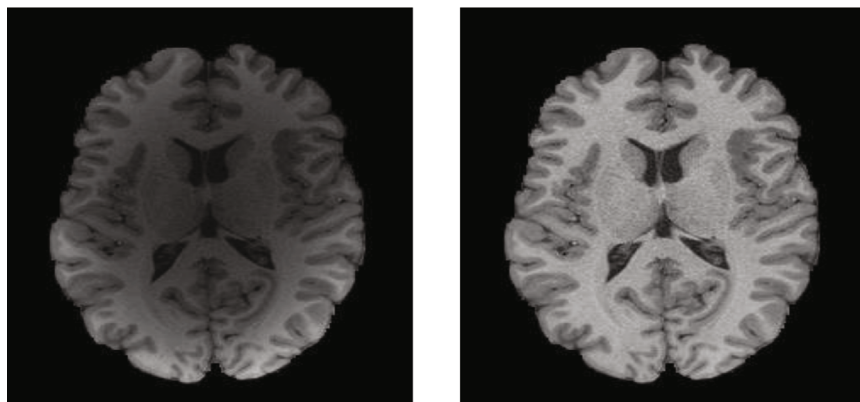
FIGURE 9: Bias field images obtained by different fusion methods on real HCP data and corresponding corrected images: (a) original image, (b) real image, (c) NPS, (d) AAW, (e) MIScnn, (f) pathify, (g) EDW, and (h) proposed method.

TABLE 2: The MSE values, SSIM values and synthesis time between the real image and the corrected image obtained by different fusion methods.

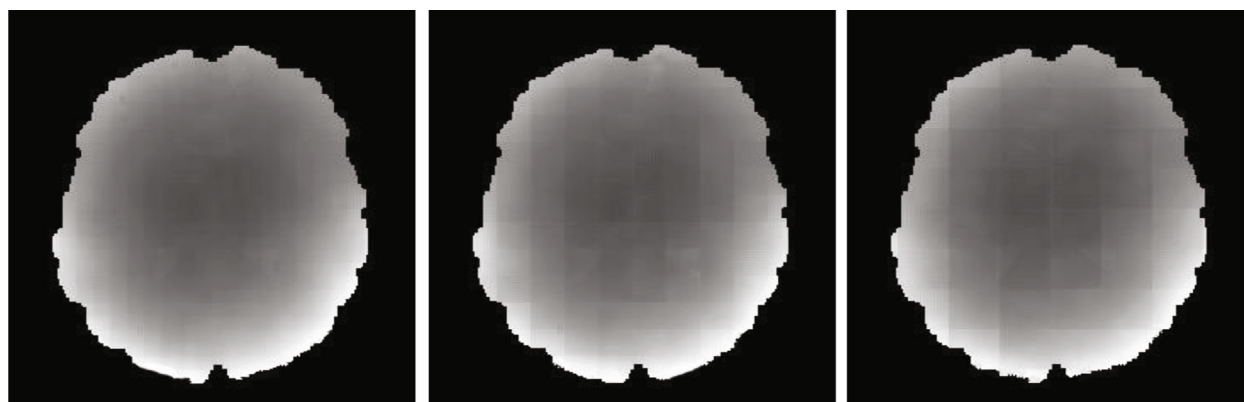
Method	NPS	AAW [5]	MIScnn [7]	Pathify [8]	EDW [9]	Proposed
MSE	$1.233 * 10^{-3}$	$1.130 * 10^{-3}$	$1.136 * 10^{-3}$	$1.468 * 10^{-3}$	$1.139 * 10^{-3}$	$1.104 * 10^{-3}$
SSIM	0.9882	0.9909	0.9913	0.9869	0.9936	0.9942
Time (s)	0.25	0.40	0.41	0.39	1.79	1.81

TABLE 3: The number of patches extracted from each image with different strides and the fusion time.

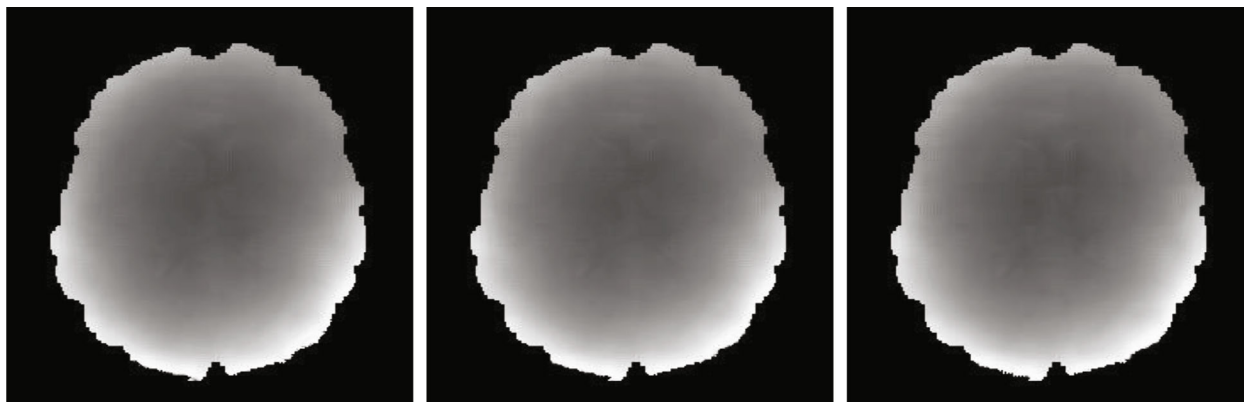
Stride	$10 \times 10$			$20 \times 20$			$30 \times 30$		
Patch number	475			130			63		
Fusion time (s)	AAW	EDW	Proposed	AAW	EDW	Proposed	AAW	EDW	Proposed
	1.77	6.16	6.20	0.61	2.44	2.45	0.37	1.79	1.81



(a)



(b)



(c)

FIGURE 10: Continued.

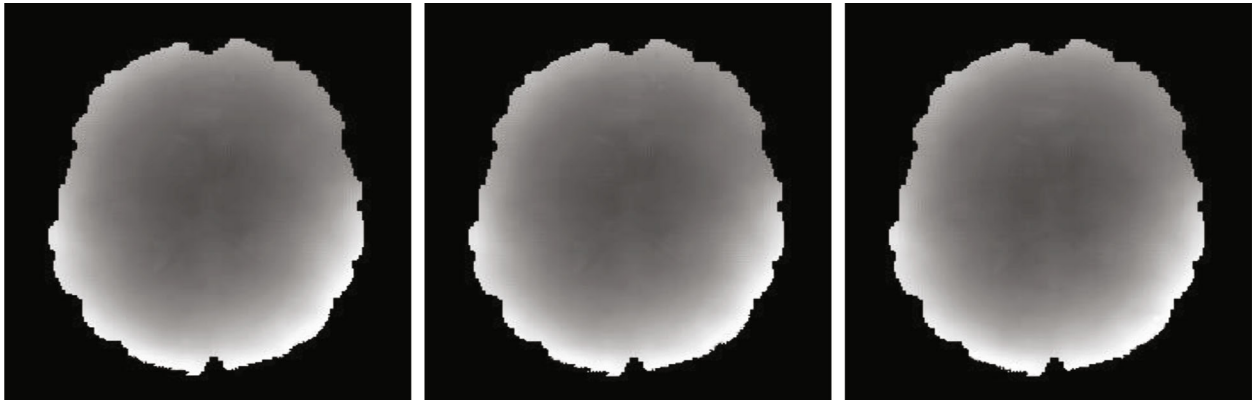


FIGURE 10: Bias field images obtained by fusing patches with AAW, EDW, and the proposed methods under different strides. (a) The original image and the real image. (b) The result of AAW. (c) The result of EDW. (d) The result of the proposed method. The step sizes from left to right are  $10 \times 10$ ,  $20 \times 20$ ,  $30 \times 30$ .

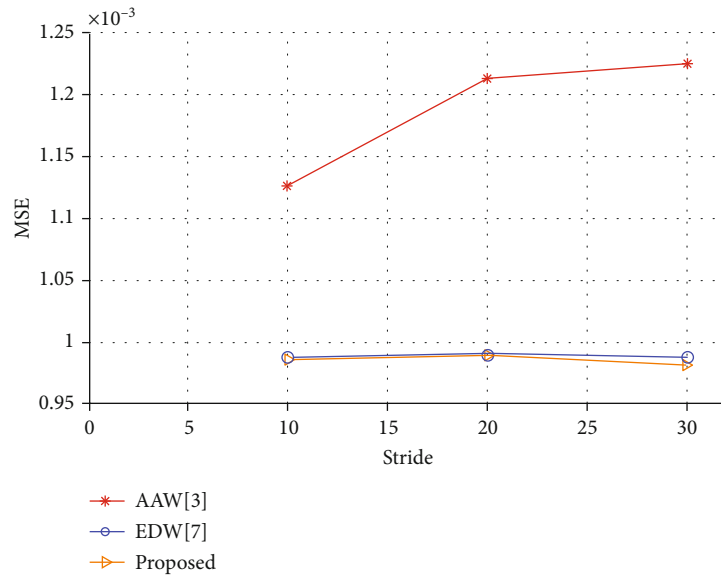


FIGURE 11: MSE values of AAW, EDW, and the proposed methods using different strides on the HCP dataset.

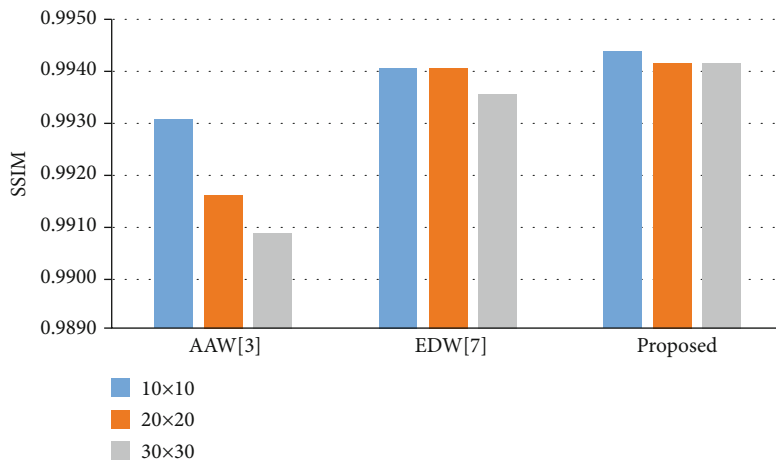


FIGURE 12: SSIM values of AAW, EDW, and the proposed methods using different strides on the HCP dataset.

TABLE 4: Average MSEs, PSNRs, and SSIMs of different fusion methods on the IXI-Guys dataset.

Method	NPS	AAW [5]	MIScnn [7]	Pathify [8]	EDW [9]	Proposed
MSE	$4.574 * 10^{-4}$	$3.034 * 10^{-4}$	$3.024 * 10^{-4}$	$3.064 * 10^{-4}$	$2.986 * 10^{-4}$	$2.971 * 10^{-4}$
PSNR	32.85	35.64	35.67	35.61	35.71	35.76
SSIM	0.9789	0.9833	0.9838	0.9835	0.9841	0.9843

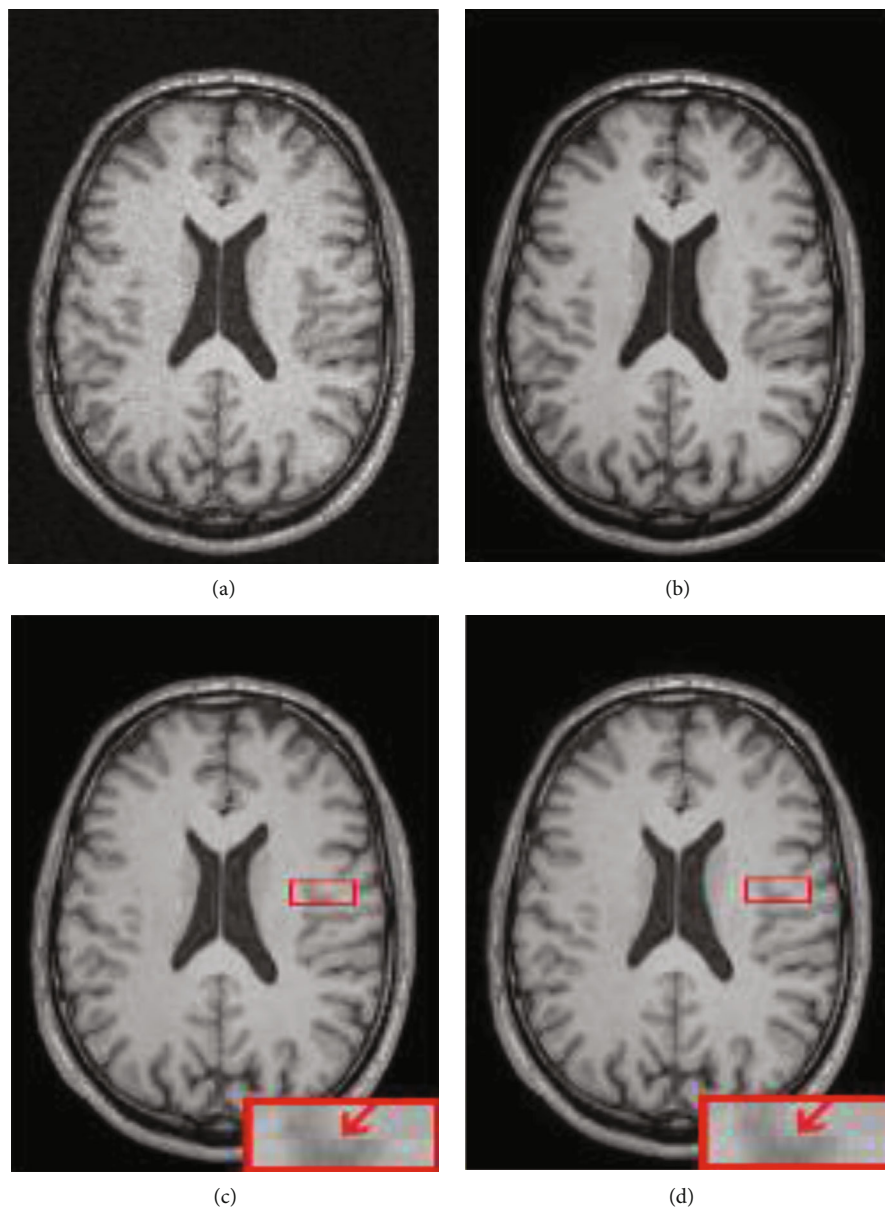


FIGURE 13: Continued.

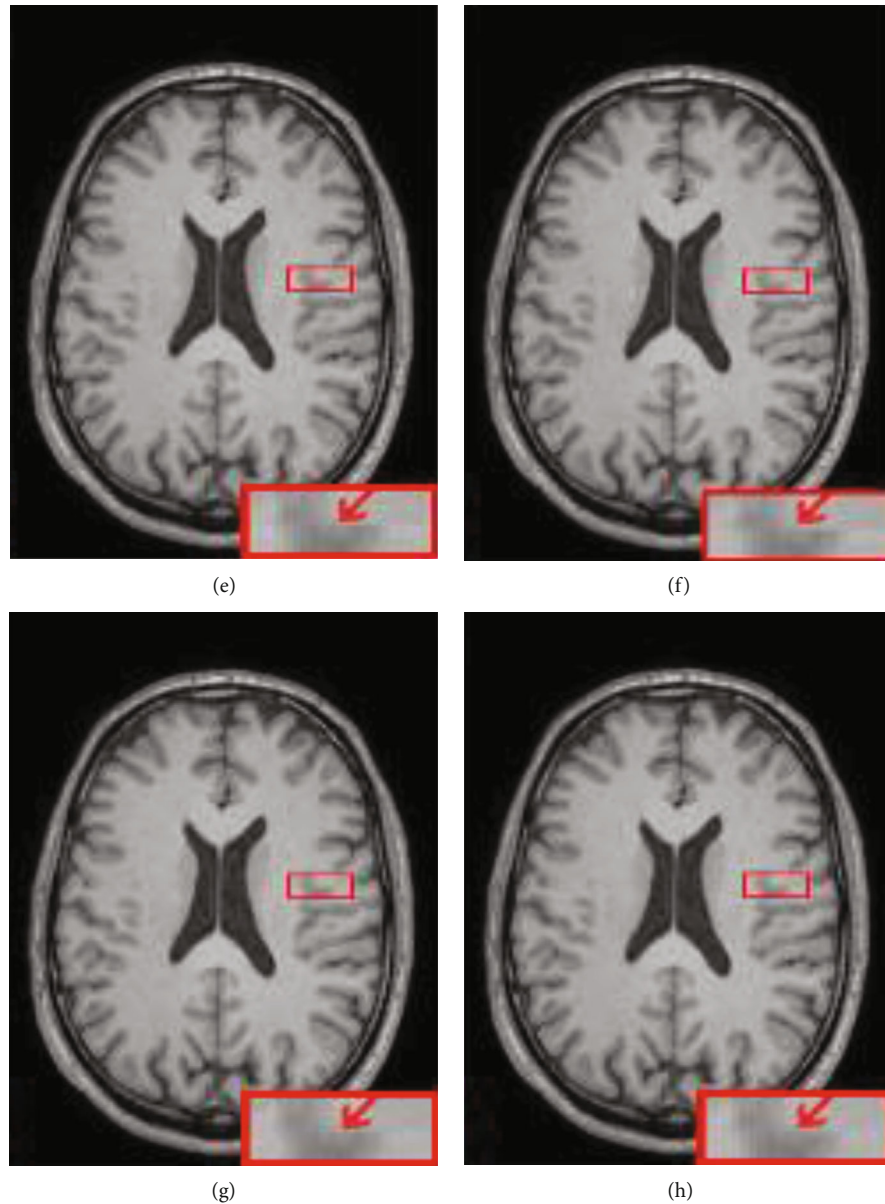


FIGURE 13: Images from the IXI-Guys dataset with 30% level Rician noise, real images, and denoised images obtained by different fusion methods: (a) noise image, (b) noise-free image, (c) NPS, (d) AAW, (e) MIScnn, (f) pathify, (g) EDW, and (h) proposed method.

with a stride of  $30 \times 30$  to obtain 63 patches from each image and finally obtained 201,600 patches. At the test phase, according to Equation (6), we used sliding windows of different strides to extract patches as shown in Table 3 and explored the performance of the proposed fusion method.

In Figure 10, we can see that the AAW method (Figure 10(b)) is greatly affected by the stride size. When the stride size is  $10 \times 10$ , the image blocking artifact is not obvious in the fused bias field image, but with increasing stride size, obvious grid artifacts appear in the fused image. The EDW method (Figure 10(c)) and our method (Figure 10(d)) are less affected by the stride size. From the line chart of the quantitative evaluation index MSE in Figure 11, we can see that the highest point of the AAW method fluctuates by approximately 9% relative to the lowest point, while the highest point of the EDW method and our method fluctuates by less than 1% rel-

ative to the lowest point %. In Figure 12, we show the change in SSIM value with the change in stride size. The SSIM value of the AAW method decreases obviously with increasing stride. Although the SSIM values of the EDW method and our method fluctuate with the change in stride, their amplitude is small. In addition, compared with the EDW method, our method resulted in a smoother bias field image. As seen from Figure 12, our method can obtain a higher SSIM value than the EDW method despite the INU level, regardless of stride size.

As seen in Table 3, the smaller the step size, the more the number of patches extracted from each image; hence, the fusion time increases accordingly. Although the AAW method can obtain a relatively smooth bias field image when the stride is  $10 \times 10$ , the fusion time difference is very small compared with the other two methods when the stride is  $30 \times 30$ . However, there are still blocking artifacts in the

images fused by the AAW method. Our method and the EDW method are less affected by the stride. Even when the stride is  $30 \times 30$ , the bias field image obtained does not appear to contain image blocking artifacts, having a lower MES value and higher SSIM value than the AAW method. Therefore, in the experiment on HCP database, to shorten the fusion time, we uniformly used a sliding window with a stride of  $30 \times 30$  to extract patches.

*4.2. Validation of the Proposed Fusion Method on the Denoising Model for 3D Brain MR Images.* To further evaluate the fusion performance of the proposed algorithm, we adopted the DnCNN denoising model [18] for validation on the IXI-Guys human dataset. We randomly selected 20 T1w brain images from the IXI-Guys dataset with an image size of  $256 \times 256 \times 150$  and voxel resolution of  $0.9375 \times 0.9375 \times 1.2 \text{ mm}^3$  [20, 21]. 12 images were randomly selected as the training set, 4 images were used for verification, and the other 4 images were used for testing. In this dataset, we manually added 30% Rician noise to simulate the noise image [22]. In order to better train the model, we cropped the image to  $240 \times 240 \times 150$ , used the mirroring method to augment the data, and finally obtained 36 training samples. For patch-based training, we set the patch size to  $60 \times 60 \times 60$  and used a sliding window with stride  $30 \times 30 \times 30$  to obtain 196 patches from each image. Finally, we obtained 7056 patches for training the 3D model. Our fusion algorithm is evaluated on the testing set based on three aspects: MSE, PSNR (peak signal-to-noise ratio), and SSIM. The results are shown in Table 4.

In Table 4, we quantitatively compared the image qualities of the proposed fusion method and the state-of-the-art fusion methods in terms of MSE, PSNR, and SSIM [23] by training the denoising model on the 3D IXI-Guys human dataset. As seen from Table 4, the fusion method proposed in this paper shows lower MSE and higher PSNR and SSIM than other methods, which further proves the effectiveness of our method. In addition, in the enlarged red box in Figure 13, it can be seen that the denoising images of NPS, AAW, MIScnn, and Pathify have obvious seam lines between patches, which further illustrates the difference in the prediction of boundary regions between adjacent patches. However, in Figures 13(g) and 13(h), this situation is much better, because both the EDW method and our method fully consider the uncertainty of the edge and assign different weights to the central and the edge regions to solve the problem. However, a small grid artifact appears faintly in Figure 13(g), while in Figure 13(h) obtained by our fusion method, there is no grid artifact and is smoother than EDW.

## 5. Discussion and Conclusions

In deep learning, when the resolution of the image to be processed is too large and the resources (such as video card and video memory) are limited, the image is divided into small patches for processing, and the image patches are reconstructed and then spliced. Since the zero-padding method is commonly used in deep learning networks to ensure the consistency of input and output sizes, this method can lead to uncertainty in edge prediction. After

image reconstruction, obvious blocking artifacts appear in the spliced image due to the inconsistency of the boundary information of each patch processed. To solve this problem, we have studied the most popularly available patch fusion methods and proposed a fusion method with a truncated Gaussian function as a weighting function to assign weights to each pixel in the patch. In this method, we used the weighting function to convert the Euclidean-distance between the overlapping point and the central point of the patch into a weight coefficient, considered the predicted pixel values in all patches, and reduced the weight of the predicted pixels at the boundary. Finally, we obtained the predicted pixel values through the weighted calculations.

We carried out experiments on the simulated database BrainWeb and the real dataset HCP using the bias removal model. After comparing the proposed method with the popularly available fusion methods, NPS, AAW, MIScnn, patchify, and EDW, our method can obtain a super seamless and smooth bias field image. As evidenced by quantitative analysis, our fusion method achieves lower MSE and greater SSIM on both simulated and real data, which is clearly superior to the other five methods. In addition, we also discussed the bias field images obtained by AAW, EDW, and our proposed method with different stride on the HCP dataset. To further demonstrate the robustness of our method, we conducted experiments by training a denoising model on the IXI-Guys human dataset. Experimental results show that our fusion method also performs better than other five methods in quantitative analysis of MSE, PSNR, and SSIM on 3D dataset. However, our method has two shortcomings: (1) the standard variance of Gaussian function  $\sigma$  needs to be determined. Different application backgrounds require different values of  $\sigma$ . For example, in the bias removal model, the best fusion performance is achieved when  $\sigma = 4$ , while on the 3D brain image denoising model, the best fusion performance is achieved when the value of  $\sigma$  is around 8; (2) while our approach for obtaining an artifact-free image under a wider sliding window takes about the same amount of time as the AAW method for obtaining the same quality image under a smaller stride, it still takes a lengthy period. In a future work, we hope to be able to adaptively select the hyperparameters of different scenes and further optimize the algorithm to shorten the running time.

In this paper, we introduced a truncated Gaussian function as a weighting function, which converted the Euclidean distance between each pixel in the patch and the center point into the weight coefficient of this pixel, to reduce the image blocking artifact in patch-based image reconstruction. We demonstrated that the method proposed in this paper has significant advantages over existing patch fusion methods. Additionally, our approach can be applied to any patch-based deep learning model, even when the model is already trained.

## Data Availability

All datasets used in this study are discussed in Section 4. They are publicly available and cited in the list of references.



## Conflicts of Interest

The authors declare no conflict of interest.

## Authors' Contributions

YX and SH are assigned to the conceptualization. SH is assigned to the methodology. YX and SH did the validation. SH and YD did the formal analysis. YX and SH curated the data. YX did the writing—original draft preparation. SH and YD did the writing—review and editing. SH and YD did the supervision. SH and YD acquired funding. All authors have read and agreed to the published version of the manuscript.

## Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 61771230, Grant 61903229 and Grant 61973180 and in part by the Natural Science Foundation of Shandong Province of China under Grant ZR2019BF004, Grant ZR2019PF005, Grant ZR2020QF011, and Grant ZR2021MF115.

## References

- [1] J. Lin, Y. Pang, L. Xu, and Z. Huang, "Research progress of medical image processing based on deep learning," *Life Science Instruments*, vol. 16, no. 4, pp. 45–54, 2018.
- [2] Q. Zhang, D. Zhao, and X. Chi, "Review for deep learning based on medical imaging diagnosis," *Computer Science*, vol. 44, no. 11A, pp. 1–7, 2017.
- [3] Y. Fu, Y. Lei, T. Wang, W. J. Curran, T. Liu, and X. Yang, "Deep learning in medical image registration: a review," *Physics in Medicine and Biology*, vol. 65, no. 20, pp. 1–32, 2020.
- [4] X. Yang, R. Kwitt, M. Styner, and M. Niethammer, "Quicksilver: fast predictive image registration - a deep learning approach," *NeuroImage*, vol. 158, pp. 378–396, 2017.
- [5] S. Hu, L. Zhang, G. Li, M. Liu, D. Fu, and W. Zhang, "Brain deformable registration using global and local label-driven deep regression learning in the first year of life," *IEEE Access*, vol. 8, pp. 25691–25705, 2019.
- [6] J. Fan, X. Cao, P. T. Yap, and D. Shen, "BIRNet: brain image registration using dual-supervised fully convolutional networks," *Medical Image Analysis*, vol. 54, pp. 193–206, 2019.
- [7] D. Müller and F. Kramer, "MIScnn: a framework for medical image segmentation with convolutional neural networks and deep learning," *BMC Medical Imaging*, vol. 21, no. 1, pp. 1–11, 2021.
- [8] W. Wu, "Patchify," <https://github.com/dovahcrow/patchify.py>.
- [9] L. Wu, S. Hu, and C. Liu, "Exponential-distance weights for reducing grid-like artifacts in patch-based medical image registration," *Sensors*, vol. 21, no. 21, p. 7112, 2021.
- [10] L. S. Athanasiou, D. I. Fotiadis, and L. K. Michalis, "Plaque Characterization Methods Using Optical Coherence Tomography," in *Atherosclerotic Plaque Characterization Methods Based on Coronary Imaging*, pp. 95–113, Academic Press, 2017.
- [11] J. Song and Z. Zhang, "Brain tissue segmentation and bias field correction of MR image based on spatially coherent FCM with nonlocal constraints," *Computational and Mathematical Methods in Medicine*, vol. 2019, Article ID 4762490, 13 pages, 2019.
- [12] C. Li, C. Xu, A. W. Anderson, and J. C. Gore, "MRI tissue classification and bias field estimation based on coherent local intensity clustering: a unified energy minimization framework," in *Proceedings of International Conference on Information Processing in Medical Imaging*, pp. 288–299, Springer-Verlag, Williamsburg, VA, USA, 2009.
- [13] H. Ren, M. El-Khamy, and J. Lee, "Dn-resnet: efficient deep residual network for image denoising," in *Asian Conference on Computer Vision*, pp. 215–230, Springer, Cham, 2018.
- [14] Y. Xu, S. Hu, and Y. Du, "Bias correction of multiple MRI images based on an improved nonparametric maximum likelihood method," *IEEE Access*, vol. 7, pp. 166762–166775, 2019.
- [15] C. A. Cocosco, V. Kollokian, R. K. Kwan, G. B. Pike, and A. C. Evans, "BrainWeb: online interface to a 3D MRI simulated brain database," *NeuroImage*, vol. 5, 1997.
- [16] D. C. Van Essen, S. M. Smith, D. M. Barch, T. E. J. Behrens, E. Yacoub, and K. Ugurbil, "The WU-Minn human connectome project: an overview," *NeuroImage*, vol. 80, pp. 62–79, 2013.
- [17] A. Hammers, C.-H. Chen, L. Lemieux et al., "Statistical neuroanatomy of the human inferior frontal gyrus and probabilistic atlas in a standard stereotaxic space," *Human Brain Mapping*, vol. 28, no. 1, pp. 34–48, 2007.
- [18] 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, no. 7, pp. 3142–3155, 2017.
- [19] J. Elam, "HCP Wu-minn, 900 Subjects Data Release: Reference Manual," 2015, <https://www.humanconnectome.org/study/hcp-young-adult/document/900-subjects-data-release>.
- [20] L. Wu, S. Hu, and C. Liu, "Denoising of 3D brain MR images with parallel residual learning of convolutional neural network using global and local feature extraction," *Computational Intelligence and Neuroscience*, vol. 2021, Article ID 5577956, 18 pages, 2021.
- [21] D. Jiang, W. Dou, L. Vosters, X. Xu, Y. Sun, and T. Tan, "Denoising of 3D magnetic resonance images with multi-channel residual learning of convolutional neural network," *Japanese Journal of Radiology*, vol. 36, no. 9, pp. 566–574, 2018.
- [22] M. Ran, J. Hu, Y. Chen et al., "Denoising of 3D magnetic resonance images using a residual encoder-decoder Wasserstein generative adversarial network," *Medical Image Analysis*, vol. 55, pp. 165–180, 2019.
- [23] J. Chen, J. Chen, H. Chao, and M. Yang, "Image blind denoising with generative adversarial network based noise modeling," in *Computer Vision and Pattern Recognition (CVPR), 2018 IEEE Conference on*, Salt Lake, 2018.