IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 18 July 2024, accepted 11 August 2024, date of publication 21 August 2024, date of current version 19 September 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3446791

RESEARCH ARTICLE

MRI Rician Noise Reduction Using Recurrent Convolutional Neural Networks

JAVIER GURROLA-RAMOS [1], TERESA ALARCON [2], (Member, IEEE), OSCAR DALMAU [1,2], AND JOSÉ V. MANJÓN [3]

[1]Department of Computer Science, Mathematics Research Center, Guanajuato 36023, Mexico
[2]Department of Computer Science and Engineering, Centro Universitario de los Valles, Ameca, Jalisco 46708, Mexico
[3]Medical Imaging Area of IBIME Research Group, ITACA Institute, Universidad Politécnica de Valencia, 46022 Valencia, Spain

Corresponding author: Oscar Dalmau (dalmau@cimat.mx)

This work was supported in part by Consejo Nacional de Ciencia y Tecnología (CONACYT), Mexico, under Grant 258033; and in part by the Project Laboratorio de Supercómputo del Bajío under Grant 300832.

ABSTRACT Magnetic resonance images are usually corrupted by noise during the acquisition process, which can affect the results of subsequent medical image analysis and diagnosis. This paper presents a denoising recurrent convolutional neural network for Brain MRI denoising. The proposed model consists of a one-level autoencoder architecture with a shortcut, in which the standard convolutional blocks are changed for a new recurrent convolutional denoising block. This block is based on the gated recurrent units combined with local residual learning, allowing us to filter the noisy image recursively. Additionally, we adopt global residual learning to directly estimate the corrupted image's noise instead of the noise-free image. The proposed model requires less computation than other models based on neural networks and experimentally outperforms state-of-the-art models on clinical brain MRI datasets, particularly for high noise levels.

INDEX TERMS Autoencoder, convolutional neural network, denoising, gated recurrent units, MRI denoising, recurrent convolutional neural network.

I. INTRODUCTION

Magnetic resonance imaging (MRI) is a non-invasive medical imaging technique based on magnetic field technology that obtains images of organs and tissues, which are used for monitoring, diagnosis and detection of different alterations. During the acquisition process, these images are corrupted by noise; this fact explains the distortion and loss of information. The statistical distribution of the noise depends on the number channels (single or multi-channel coils) and the reconstruction method (sum of square, root sum of square, spatial matched filter) for combining the data [1]. According to experimental results, see McVeigh et al. [2], the noise in both the k-space and the image domain is Gaussian with zero mean and equal variance in the real and imaginary parts. For the previous reason, a common assumption is to consider the real and imaginary components of the MR complex raw data corrupted by white additive Gaussian noise with the same variance in the real and imaginary parts. This assumption is valid for single and multi-coil acquisitions [3]. In case of single coil acquisitions, it is known that magnetic resonance magnitude images can be modeled with a Rician distribution [3], [4], [5]. That is, let $\boldsymbol{R}$ and $\boldsymbol{I}$ the real and imaginary parts of an MRI image, respectively, the magnitude of the noise-free MRI image $\boldsymbol{y}$ and its corresponding noisy image $\boldsymbol{x}$ are defined as follows:

$$\boldsymbol{y} = \sqrt{\boldsymbol{R}^2 + \boldsymbol{I}^2}, \tag{1}$$

$$\boldsymbol{x} = \sqrt{(\boldsymbol{R} + \eta_R)^2 + (\boldsymbol{I} + \eta_I)^2}, \tag{2}$$

where $\eta_R, \eta_I \sim \mathcal{N}(0, \sigma)$, i.e., the magnitude $\boldsymbol{x}$ of the complex MR image is described by a Rician distribution. The presence of noise affects the subsequent image processing tasks, such as image analysis and interpretation. Therefore, image denoising is a crucial step during image processing to improve image quality [6], allowing the analysis algorithms' better performance. For instance, Hua et al. [7] provide an analysis of different clustering algorithms for the

The associate editor coordinating the review of this manuscript and approving it for publication was Marco Giannelli.

segmentation of brain tissues. In this research the authors consider different image noise levels and confirm the need of the denoising step before the segmentation, in order to increase the precision of all clustering techniques for brain tissue detection. Prakash et al. in [8] propose the use of denoising networks to improve the performance of other DL-based image segmentation methods. A similar research is carried out in [9]. In this work, the authors assess the impact of different denoising techniques on classification task for medical images. The authors also show that the use of denoising methods yield a significant improvement of the classification and prediction results.

There are classical methods that address the denoising problem considering the self-similarity of patterns in the image [5], [10], [11]. The main idea of this class of methods is to reduce noise by averaging similar patterns. Majon et al. [5] propose a variant of non-local means filter considering the sparseness and self-similarity properties, i.e., the pre-filtered rotationally invariant non-local means (PRI-NLM). This method combines the discrete cosine transform and a modified version of the non-local means filter based on rotationally invariant similarity. Similarly, Maggioni et al. [10] propose a collaborative filtering by exploiting the local correlation between voxels and the non-local correlation between voxels (BM4D). Kong et al. [11] propose a collaborative filtering method based on the tensor decomposition framework (MNL-tSVD) and consider the self-similarity property and the 3D structure of magnetic resonance images. The mentioned techniques have been used successfully for denoising of volumetric MRI data because they considerably reduce and smooth the noise, achieving high PSNR. It is worth to mention two merits of these conventional techniques: its computational efficiency and they require less training data [12]. However they have a limitation of a fine tuning on specific data, lack adaptability to consider different kinds of noise and the extraction of features for different dataset is hard. For more details see [13].

More recently, deep learning techniques have emerged as a successful alternative for MRI denoising [14], [15], [16], [17]. Some of these models have assumed a residual autoencoder convolutional neural network in order to maintain structural details that are present in the noisy image. Ran et al. [15] propose a model that combines a residual autoencoder with the Wasserstein generative adversarial network (RED-WGAN), and for the training, they propose a combined loss function that includes the mean square error, a perceptual loss, and a discriminative loss. Dongsheng Jiang et al. [14] propose a multichannel denoising convolutional neural network (MCDnCNN) which is an extension of the Residual Convolutional Neural Network (DnCNNs) originally proposed by Zhang et al. [18] for natural image denoising.

While self-similarity methods and CNNs serve different purposes and operate at different levels of abstraction, they both contribute significantly to the field of image processing and analysis. Deciding between self-similarity methods and CNNs for denoising hinges on the specific characteristics of the noise, the desired level of detail preservation, available computational resources, and the availability of training data. Self-similarity methods perform exceptionally well in scenarios characterized by uniform noise and specific requirements for preserving texture. In contrast, CNNs provide a more generalizable approach capable of effectively managing varied noise patterns and achieving cutting-edge results given adequate training data and computational resources. Combining both approaches or tailoring the choice based on specific task requirements can lead to optimized solutions for denoising images in various practical applications, see a summary in Table 1, see also [19] and references therein.

In this paper, we present a denoising recurrent convolutional neural network (DRCnet) for Rician noise reduction in MR images. The main contributions of this paper are:

1) A Deep recurrent convolutional neural network that can be efficiently used for MRI denoising.
2) A new denoising block that integrates factorized convolutions in a gated recurrent neural network, i.e., the recurrent convolutional denoising block (RCDB).

We assess both the state-of-the-art methods and the proposed model for the task of removing Rician noise. The experiment setup includes two MRI datasets and the corresponding datasets corrupted by Rician noise with levels in the range $[1\%, 15\%]$. According to the experiments, the proposed method yields competitive results compared to state-of-the-art methods.

The rest of the paper is organized as follows: Section II describes the proposed architecture, including the recurrent convolutional denoising block and a brief study of parameters (ablation study); in Section III we present details of the training step, some experimental results, and a discussion; finally Section IV presents the conclusions.

II. PROPOSED METHOD

The proposed model is shown in Fig. 1. The aim of MRI denoising is to restore the original MR image $\boldsymbol{y}$ from the corresponding noisy image $\boldsymbol{x}$. The general idea is to remove or reduce the noise level of the noisy image $\boldsymbol{x}$ in order to achieve a high-quality estimation $\hat{\boldsymbol{y}}$. Therefore, this problem can be formulated as finding the parametric function $\mathcal{G}(\cdot;\Theta)$ such that $\hat{\boldsymbol{y}} = \mathcal{G}(\boldsymbol{x};\Theta)$, where $\hat{\boldsymbol{y}}$ is an estimation of the clean MR image $\boldsymbol{y}$, and Θ are unknowns parameters of the function.

In order to estimate the parameters of the function $\mathcal{G}(\cdot,\Theta)$, we can follow a standard machine learning technique. That is, given a training dataset $\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^{N}$ containing N pairs of noisy and clean images, respectively, we can estimate the parameters Θ of the function $\mathcal{G}(\cdot;\Theta)$ solving the following optimization problem:

$$\Theta^* = \arg\min_{\Theta} \frac{1}{N}\sum_{i=1}^{N} \mathcal{L}\left(\mathcal{G}(\boldsymbol{x}_i;\Theta), \boldsymbol{y}_i\right) + \frac{\lambda}{2}\Omega(\Theta), \tag{3}$$

TABLE 1. Comparing self-similarity methods and CNN-based methods.

	Self-similarity-based methods	CNN-based methods
Feature Representation	Self-similarity methods focus on capturing local patterns and textures explicitly within an image. Effective for moderate noise reduction and preserving texture details but may struggle with complex noise patterns.	CNNs learn hierarchical features automatically from data, capturing both local patterns and global context. Highly effective across various noise levels and types, capable of learning complex noise patterns and global features. They can be useful for tasks demanding high accuracy and generalization across diverse and complex datasets.
Computational Requirements	Require less data compared to CNNs. They rely on local patterns and similarities within the image itself rather than learning from a large dataset. Generally less computationally intensive, making them more feasible for real-time applications. However, these methods can be computationally intensive, especially for high-resolution images or complex patterns.	Typically require large amounts of data for training to effectively learn features and patterns. Require significant computational resources, particularly during training, but benefit from optimizations and parallel processing on GPUs.
Interpretability	Often more interpretable with explicit metrics like SSIM, that reflects the similarity or structural integrity of an image, but sensitive to parameter settings.	Given their layered structure and complex internal representations, these methods often lack direct interpretability. Require careful architectural design and hyperparameter tuning, which presents decision-making challenges.

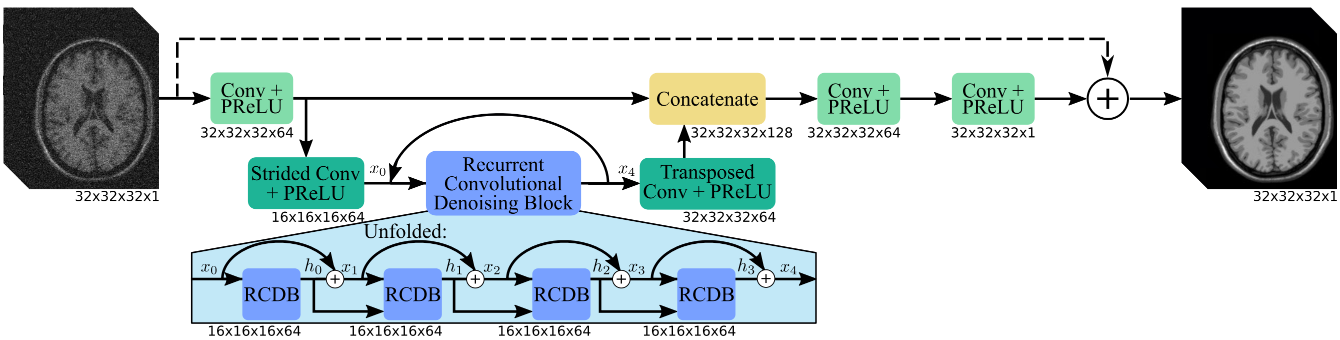


FIGURE 1. Architecture of the proposed model. The recurrent convolutional denoising block (DRCD) is shown in an unfolded form, see Fig. 2a.

where the first term is the fidelity term, the second term is the regularization term, $\mathcal{L}(\cdot,\cdot)$ is the loss/cost function and $\lambda > 0$ is a hyperparameter that controls the trade-off between fidelity and regularization terms. In particular, in this work we use ℓ_1-norm and ℓ_2-norm as fidelity and regularization terms, respectively. In the case of $\mathcal{L}(\cdot,\cdot)$, we use the ℓ_1-norm since it is known that ℓ_1-norm is a robust metric, this makes the model more robust and less prone to overfitting. This metric has also been used with success for natural image denoising [20]. On the other hand, for the regularizer $\Omega(\Theta)$ we use the ℓ_2-norm, that penalizes the model for having large weights.

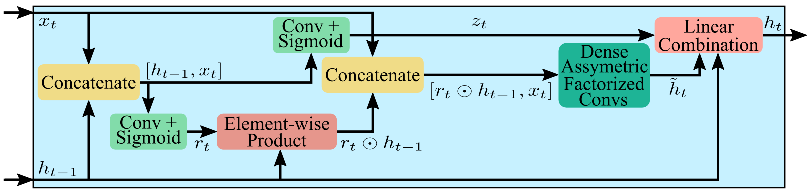
Since the purpose is to reconstruct the original image from the noisy image $\boldsymbol{x}$, and noting that $\boldsymbol{x}$ contains the main structures of $\boldsymbol{y}$, this suggests modeling the function $\mathcal{G}(\cdot;\Theta)$ as follows:

$$\mathcal{G}(\boldsymbol{x};\Theta) = \mathcal{F}(\boldsymbol{x};\Theta) + \boldsymbol{x}, \tag{4}$$

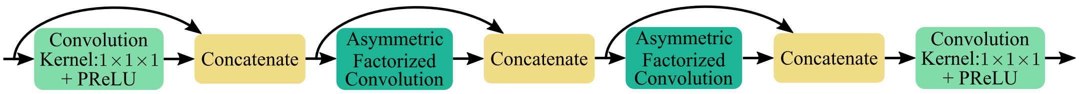
i.e., the above equation models the additive noise, through $\mathcal{F}(\boldsymbol{x};\Theta)$, while maintaining the observed information $\boldsymbol{x}$. This formulation, Eq. (4), is called Global Residual Learning (GRL).

The main structure of our proposal is therefore the function $\mathcal{F}(\boldsymbol{x};\Theta)$ and it consists of three blocks: one encoding block, one processing block, described in Sec. II-A, and one decoding block. Additionally, the encoding and decoding blocks are connected through a shortcut, Eq. (4).

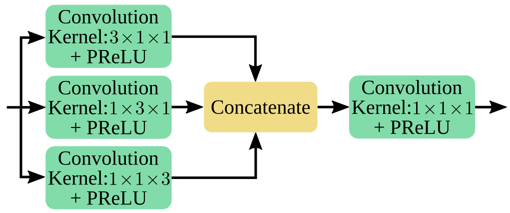
- Encoding block: The first block consists of a convolution with a kernel size $k = 3\times3\times3$, and it is used for feature extraction of the noisy brain volume $\boldsymbol{x}$. Afterward, a convolution with kernel size $k = 2\times2\times2$ and stride size of 2 is used to downsample the image, halving its spatial dimension, which consequently reduces the computational cost.
- Processing block: The main process is carried out during this stage (denoising block) and is applied to the downsampled image obtained from the previous block, see details in Sec. II-A. The idea of this process is to reduce recursively the noise, i.e., this module basically corresponds to a recurrent Convolution Denoising Block based on Gated recurrent units (GRUs).
- Decoding block: The processed image in the previous stage is upsampled using a transposed convolution with kernel size $k = 2\times2\times2$ and stride 2, and its result is concatenated to the image generated by input convolution. Finally, a convolution with kernel size $k = 1\times1\times1$ and another convolution with $k = 3\times3\times3$ are



(a) Recurrent convolutional denoising block (RCDB).



(b) Dense asymmetric factorized convolutions.



(c) Asymmetric factorized convolution.

FIGURE 2. **Main building blocks of the proposed model.**

performed to estimate the residual noise of the input image and perform the residual learning according to Eq. (4).

Each convolution of the proposed model generates 64 feature maps. These convolutions use Parametric Rectified Linear Unit (PReLU) or Sigmoid as activation function.

A. RECURRENT CONVOLUTIONAL DENOISING BLOCK

We consider the denoising operation as a sequence of image filtering to reduce the approximation error. Then, based on Eq. (4), we can estimate a filtered image $\boldsymbol{x}_{t+1}$ using the following recurrence

$$\boldsymbol{x}_{t+1} = \boldsymbol{x}_t + \boldsymbol{h}_t, \tag{5}$$

where $\boldsymbol{x}_t$ is the filtered image from the previous iteration, and $\boldsymbol{h}_t$ is the hidden state used to filter the image generated from the previously filtered image $\boldsymbol{x}_t$. This operation can be performed under a gated recurrent network scheme that we call Recurrent Convolution Denoising Block (RCDB). Unlike recurrent neural networks (RNNs) where every recurrence receives an input vector $\boldsymbol{x}_t$ at every time t, the RCDB generates the filtered image $\boldsymbol{x}_{t+1}$ using the hidden state $\boldsymbol{h}_t$ and the previously filtered image $\boldsymbol{x}_t$. The proposed RCDB is shown in Fig. 2a.

Typical gated recurrent units (GRU) operate using fully connected units, i.e., they learn parameter matrices and operate on vectors. However, the convolutions applied to MRI volumes are 4D tensors. Thus, applying the GRU to a vectorized volume can be computationally expensive; moreover, vectorized volumes do not take advantage of the local structure present in the feature maps. To overcome the previous drawbacks, we replace the fully connected operations in GRU with convolution operations. The RCDB is defined by the following equations:

$$z_t = \sigma\left(k_z * [\boldsymbol{h}_{t-1}, \boldsymbol{x}_t]\right), \tag{6}$$

$$\boldsymbol{r}_t = \sigma\left(k_r * [\boldsymbol{h}_{t-1}, \boldsymbol{x}_t]\right), \tag{7}$$

$$\tilde{\boldsymbol{h}}_t = g\left([\boldsymbol{r}_t \odot \boldsymbol{h}_{t-1}, \boldsymbol{x}_t]; \theta_g\right), \tag{8}$$

$$\boldsymbol{h}_t = (1 - z_t) \odot \boldsymbol{h}_{t-1} + z_t \odot \tilde{\boldsymbol{h}}_t, \tag{9}$$

$$\boldsymbol{x}_{t+1} = \boldsymbol{x}_t + \boldsymbol{h}_t. \tag{10}$$

Then, given the initial image $\boldsymbol{x}_0$, and the hidden state, $\boldsymbol{h}_{-1} = \boldsymbol{0}$, we can generate the sequence $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_{t+1}$ where $\boldsymbol{x}_{t+1}$ depends on $\boldsymbol{x}_t$ and the hidden state $\boldsymbol{h}_t$. Note that the hidden state $\boldsymbol{h}_t$ depends on $\boldsymbol{h}_{t-1}$ and $\boldsymbol{x}_t$.

The update gate z_t decides how much information from the previous hidden unit is passed to the future while the reset gate $\boldsymbol{r}_t$ allows forgetting the previous hidden state when the values of $\boldsymbol{r}_t$ are close to zero. On the other hand, k_z, k_r are the convolutional kernels of the convolutions in the RCDB. The symbol $*$ denotes the convolution operation, and $\boldsymbol{x}_t$ is the filtered image at time t. The function $g(\cdot; \theta_g)$ represents a set of dense asymmetric factorized convolutions described in detail below, see also Fig. 2b.

Although the computational cost is reduced through convolutions instead of full matrices, the number of learnable parameters and operations performed can be high when dealing with 3D images. For these reasons, we use factorized convolutions, Fig. 2c, based on the Inception module from the Inception-v2 model [21]. Instead of performing a convolution with kernel size $3 \times 3 \times 3$, three asymmetric convolutions with kernel sizes $3 \times 1 \times 1$, $1 \times 3 \times 1$, and $1 \times 1 \times 3$ are performed in parallel. Afterward, the resulting images are concatenated, and the number of feature maps is reduced using a $1 \times 1 \times$ 1 convolution. To illustrate the reduction of computational cost, consider an MRI volume of size $N_1 \times N_2 \times N_3$ with f_i feature maps and a 3D convolution with kernel size $k \times k \times k$ with f_i input feature maps and f_o feature maps that preserves the spatial dimension of the image. This type of convolution requires $N_1N_2N_3k^3f_if_o$ multiplications. On the other hand, using factorized 3D convolution, only $3N_1N_2N_3f_o(f_ik + f_o)$ multiplications are required. For $k = 3$, the asymmetric convolution is 60% cheaper than the 3D convolution. As in the rest of the model, every convolution generates 64 feature maps.

In addition to the proposed model for the case of MRI volumes, we present a simplified version to handle 3D images by estimating 2D slices individually. This 2D version, denoted as DRCnet-2D, uses standard 2D convolution, has less trainable parameters, and the asymmetric factorized convolution (See Fig. 2c) has only two convolutions with kernel size 3×1 and 1×3 at the beginning. One advantage of the 2D model over the 3D model is that it requires less memory; however, the time to estimate the complete volume is higher. The estimation of the $j-$th denoised slice is carried out taking the neighbor slices $\{j-2, j-1, j+1, j+2\}$ as additional input channels, as in the case of the MCDnCNN model.

B. ABLATION STUDY

Now, we compare the behavior of the DRCnet using different configurations without modifying the number of trainable parameters, which is 406.79 k. For the comparison, we consider the number of unfolded RCDB from 1 to 4 and the use of global residual learning (GRL). The results of the comparison are shown in Table 2. Based on this comparison, the model DRCnet used for the experimental section corresponds to the model with 4 unfolded RCDB and global residual learning. Since the RCDB requires 10.78G of multiplication-accumulation operations (MACs), this block represents 77% of the computation of the DRCnet. For this study we used the Hammersmith and Guys databases, see details of these databases in Section III-A

TABLE 2. Ablation study: multiplication-accumulation operations, execution time required to estimate a 64 × 64 × 64 volume and PSNR values of T1 images for 9% noise level.

Unfoldings	GRL	MACs	Time (ms) CPU / GPU	PSNR (dB) Hammersmith / Guys
1	✗	23.61G	525.80 / 18.61	34.79 / 33.48
1	✓	23.61G	530.31 / 19.82	34.88 / 33.53
2	✗	34.39G	681.86 / 26.51	35.43 / 33.86
2	✓	34.39G	748.58 / 28.44	35.47 / 33.90
3	✗	45.17G	894.55 / 33.42	35.65 / 34.01
3	✓	45.17G	937.16 / 34.10	35.66 / 34.02
4	✗	55.96G	1041.56 / 40.80	35.74 / 34.06
4	✓	55.96G	1189.33 / 42.10	35.75 / 34.08

III. EXPERIMENTS

In this section, we validate the performance of the proposed DRCnet model and its 2D version, i.e., the model DRCnet-2D, described in Sec. II-A. We compare both models with state-of-the-art MRI denoising algorithms, using classical and CNN-based denoising models, all available online. We conducted several experiments with clinical datasets, using T1, T2, and PD sequences.

TABLE 3. Number of trainable parameters, multiplication-accumulation operations, and execution time of the CNN-based models required to estimate a 64 × 64 × 64 volume.

Model	Trainable parameters	MACs	Time (ms) CPU / GPU
MCDnCNN	299.46 k	78.72 G	533.76 / 39.68
RED-WGAN	2.33 M	610.00 G	3434.18 / 98.15
DRCnet-2D	279.55 k	69.76 G	1488.64 / 192.64
DRCnet	406.79 k	55.96 G	1189.33 / 42.10

For the comparison, we consider the following classical methods for MRI denoising: BM4D [10], PRI-NLM3D [5], MNL-tSVD [11]. Additionally, we include well-known CNN-based denoising methods: MCDnCNN [14] and RED-WGAN [15]. For assessing the previous models, we use the peak signal-to-noise ratio (PSNR) average [22] and the structural similarity index (SSIM) average [23]. It is worth mentioning we have only included classical and CNN-based methods that have their source code publicly available by the authors. For the classical techniques, we used the default parameters provided by the authors. In the case of the CNNs, we trained them from scratch using the same datasets used for training the proposed model, see Sec. III-B.

A. DATASET

The clinical IXI dataset is available at https://brain-development.org/ixi-dataset/ and consists of images acquired from 3 different hospitals. For our experiments, we consider Hammersmith and Guy's subsets of the IXI dataset, which were acquired using a Philips 3T system and a Philips 1.5 T system, respectively. Please refer to the previously mentioned website for more details of the scanner parameters. This dataset was used to train the CNN-based models and test all the compared models.

B. EXPERIMENTAL SETTING FOR TRAINING

To prepare the training, validation, and testing data sets, we first split the Hammersmith and Guy data sets into three subsets: 80% for training, 10% for validation and 10% for testing respectively. The above corresponding subsets are then combined to obtain the final training, validation and testing datasets. Note that each dataset contains images from both the Hammersmith and Guy datasets. The final datasets are used for training and testing the CNN-based models. Each CNN-based model was trained independently for each MRI sequence, i.e., T1, T2, and PD.

During the training, we selected random patches of size $32 \times 32 \times 32$ for the 3D models, and patches of size 32×32 for the 2D models, generating 10320 samples per epoch. We obtained the corresponding noisy patch for each patch by adding random Rician noise (2) with noise levels in the range

TABLE 4. Results of different denoising methods on the IXI Hammersmith dataset.The best two results of PSNR (dB) and SSIM are highlighted in red and blue respectively.

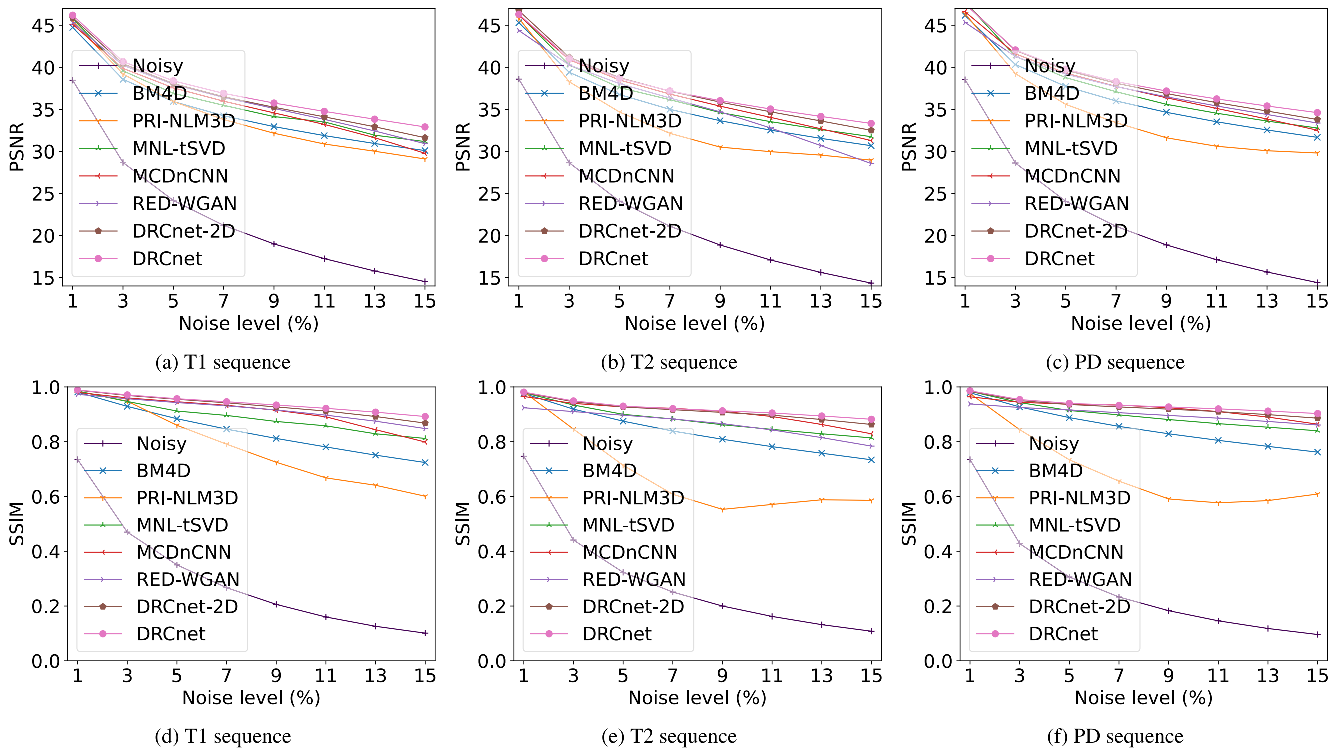
Method	Noise level 1% PSNR/SSIM	3% PSNR/SSIM	5% PSNR/SSIM	7% PSNR/SSIM	9% PSNR/SSIM	11% PSNR/SSIM	13% PSNR/SSIM	15% PSNR/SSIM
T1								
Noisy	38.46 / 0.735	28.68 / 0.470	24.17 / 0.3503	21.21 / 0.267	19.00 / 0.206	17.25 / 0.160	15.78 / 0.126	14.52 / 0.101
BM4D	44.74 / 0.981	38.57 / 0.929	35.93 / 0.884	34.24 / 0.846	32.95 / 0.812	31.88 / 0.781	30.94 / 0.751	30.10 / 0.724
PRI-NLM3D	45.50 / 0.984	39.21 / 0.948	35.93 / 0.860	33.79 / 0.791	32.16 / 0.725	30.88 / 0.668	30.02 / 0.641	29.10 / 0.601
MNL-tSVD	45.79 / 0.985	39.58 / 0.947	36.91 / 0.912	35.48 / 0.896	34.14 / 0.874	33.51 / 0.858	31.93 / 0.829	31.17 / 0.812
MCDnCNN	45.11 / 0.980	39.87 / 0.960	37.56 / 0.946	35.97 / 0.933	34.58 / 0.915	33.21 / 0.891	31.63 / 0.844	29.78 / 0.799
RED-WGAN	45.29 / 0.973	40.24 / 0.957	37.98 / 0.943	36.42 / 0.931	35.11 / 0.916	33.81 / 0.897	32.43 / 0.875	30.92 / 0.848
DRCnet-2D	45.90 / 0.988	40.42 / 0.969	38.07 / 0.955	36.50 / 0.942	35.25 / 0.927	34.11 / 0.912	32.93 / 0.892	31.63 / 0.868
DRCnet	46.19 / 0.989	40.68 / 0.971	38.39 / 0.957	36.90 / 0.946	35.75 / 0.934	34.76 / 0.922	33.83 / 0.908	32.91 / 0.892
T2								
Noisy	38.59 / 0.747	28.65 / 0.441	24.08 / 0.323	21.10 / 0.251	18.87 / 0.200	17.09 / 0.162	15.61 / 0.132	14.34 / 0.108
BM4D	45.30 / 0.976	39.41 / 0.919	36.76 / 0.875	35.00 / 0.839	33.65 / 0.809	32.53 / 0.782	31.55 / 0.758	30.68 / 0.734
PRI-NLM3D	45.82 / 0.982	38.28 / 0.846	34.68 / 0.714	32.15 / 0.608	30.51 / 0.553	29.97 / 0.571	29.56 / 0.588	28.96 / 0.586
MNL-tSVD	46.26 / 0.979	40.28 / 0.934	37.59 / 0.900	36.15 / 0.883	34.62 / 0.862	33.54 / 0.845	32.60 / 0.829	31.75 / 0.814
MCDnCNN	46.18 / 0.965	40.90 / 0.940	38.49 / 0.927	36.82 / 0.922	35.38 / 0.911	34.05 / 0.892	32.69 / 0.863	31.29 / 0.829
RED-WGAN	44.37 / 0.924	40.27 / 0.910	38.06 / 0.897	36.35 / 0.883	34.70 / 0.866	32.79 / 0.843	30.69 / 0.815	28.56 / 0.784
DRCnet-2D	46.73 / 0.979	41.20 / 0.946	38.77 / 0.927	37.16 / 0.917	35.86 / 0.907	34.71 / 0.897	33.62 / 0.881	32.51 / 0.863
DRCnet	46.30 / 0.981	41.00 / 0.949	38.69 / 0.930	37.19 / 0.921	36.03 / 0.913	35.04 / 0.905	34.16 / 0.894	33.33 / 0.882
PD								
Noisy	38.53 / 0.735	28.62 / 0.428	24.07 / 0.306	21.09 / 0.233	18.88 / 0.183	17.12 / 0.146	15.66 / 0.118	14.40 / 0.096
BM4D	46.17 / 0.979	40.33 / 0.927	37.73 / 0.888	36.00 / 0.856	34.65 / 0.829	33.52 / 0.805	32.55 / 0.783	31.67 / 0.762
PRI-NLM3D	46.37 / 0.976	39.25 / 0.844	35.59 / 0.734	33.42 / 0.656	31.62 / 0.591	30.61 / 0.577	30.08 / 0.585	29.81 / 0.609
MNL-tSVD	47.75 / 0.984	41.31 / 0.943	38.77 / 0.914	37.10 / 0.898	35.60 / 0.881	34.54 / 0.866	33.61 / 0.853	32.78 / 0.840
MCDnCNN	46.60 / 0.964	41.60 / 0.944	39.32 / 0.937	37.73 / 0.934	36.38 / 0.924	35.11 / 0.910	33.85 / 0.890	32.52 / 0.864
RED-WGAN	45.34 / 0.938	41.27 / 0.925	39.19 / 0.916	37.70 / 0.906	36.49 / 0.896	35.40 / 0.886	34.39 / 0.874	33.37 / 0.861
DRCnet-2D	47.60 / 0.984	42.01 / 0.950	39.66 / 0.936	38.11 / 0.927	36.88 / 0.919	35.80 / 0.910	34.80 / 0.899	33.79 / 0.886
DRCnet	47.51 / 0.986	42.06 / 0.954	39.77 / 0.940	38.30 / 0.933	37.18 / 0.927	36.24 / 0.920	35.40 / 0.912	34.61 / 0.903

TABLE 5. Results of different denoising methods on the IXI Guys dataset. The best two results of PSNR (dB) and SSIM are highlighted in red and blue respectively.

Method	Noise level 1% PSNR/SSIM	3% PSNR/SSIM	5% PSNR/SSIM	7% PSNR/SSIM	9% PSNR/SSIM	11% PSNR/SSIM	13% PSNR/SSIM	15% PSNR/SSIM
T1								
Noisy	38.49 / 0.746	28.73 / 0.513	24.21 / 0.416	21.25 / 0.346	19.05 / 0.290	17.29 / 0.244	15.83 / 0.207	14.58 / 0.175
BM4D	43.15 / 0.981	36.64 / 0.927	33.91 / 0.878	32.22 / 0.839	31.01 / 0.807	30.07 / 0.779	29.26 / 0.752	28.54 / 0.727
PRI-NLM3D	44.17 / 0.977	37.90 / 0.954	34.86 / 0.884	32.70 / 0.803	31.01 / 0.730	29.68 / 0.669	28.64 / 0.629	27.94 / 0.609
MNL-tSVD	44.87 / 0.986	38.21 / 0.946	35.51 / 0.909	34.00 / 0.894	32.45 / 0.870	31.63 / 0.853	30.78 / 0.836	30.00 / 0.817
MCDnCNN	43.50 / 0.976	38.25 / 0.959	35.86 / 0.947	34.34 / 0.936	33.15 / 0.922	32.16 / 0.908	31.31 / 0.897	30.48 / 0.879
RED-WGAN	43.87 / 0.966	38.55 / 0.953	36.27 / 0.942	34.78 / 0.931	33.65 / 0.921	32.70 / 0.910	31.87 / 0.898	31.08 / 0.885
DRCnet-2D	44.73 / 0.988	38.88 / 0.970	36.45 / 0.958	34.90 / 0.943	33.72 / 0.937	32.74 / 0.926	31.93 / 0.915	31.19 / 0.903
DRCnet	44.93 / 0.990	39.05 / 0.972	36.66 / 0.960	35.17 / 0.950	34.08 / 0.941	33.20 / 0.932	32.45 / 0.923	31.77 / 0.913
T2								
Noisy	38.44 / 0.731	28.62 / 0.458	24.08 / 0.355	21.11 / 0.289	18.89 / 0.240	17.11 / 0.201	15.64 / 0.169	14.37 / 0.143
BM4D	43.59 / 0.976	37.32 / 0.919	34.67 / 0.875	32.96 / 0.839	31.68 / 0.809	30.63 / 0.782	29.72 / 0.758	28.91 / 0.734
PRI-NLM3D	44.64 / 0.971	37.71 / 0.883	34.34 / 0.720	31.79 / 0.607	29.72 / 0.535	28.49 / 0.501	27.41 / 0.475	27.36 / 0.530
MNL-tSVD	45.28 / 0.983	39.59 / 0.940	36.36 / 0.902	34.84 / 0.885	33.28 / 0.864	32.25 / 0.846	31.36 / 0.829	30.57 / 0.814
MCDnCNN	45.01 / 0.963	39.69 / 0.946	37.24 / 0.935	35.55 / 0.922	34.15 / 0.905	32.96 / 0.886	31.67 / 0.850	30.37 / 0.805
RED-WGAN	43.26 / 0.926	38.97 / 0.914	36.77 / 0.903	35.11 / 0.891	33.67 / 0.876	32.23 / 0.859	30.57 / 0.837	28.70 / 0.811
DRCnet-2D	45.59 / 0.981	40.00 / 0.955	37.54 / 0.940	35.92 / 0.928	34.64 / 0.917	33.54 / 0.906	32.55 / 0.893	31.60 / 0.879
DRCnet	45.15 / 0.982	39.68 / 0.956	37.38 / 0.941	35.89 / 0.930	34.75 / 0.921	33.79 / 0.912	32.95 / 0.902	32.16 / 0.891
PD								
Noisy	38.40 / 0.724	28.61 / 0.454	24.09 / 0.351	21.12 / 0.285	18.91 / 0.236	17.15 / 0.198	15.69 / 0.167	14.44 / 0.142
BM4D	44.32 / 0.981	38.24 / 0.927	35.65 / 0.885	33.99 / 0.851	32.75 / 0.822	31.73 / 0.797	30.86 / 0.774	30.09 / 0.753
PRI-NLM3D	45.47 / 0.983	38.51 / 0.876	34.80 / 0.748	32.98 / 0.668	31.33 / 0.617	29.89 / 0.571	28.75 / 0.534	27.73 / 0.504
MNL-tSVD	46.01 / 0.985	39.81 / 0.946	37.36 / 0.916	35.80 / 0.902	34.30 / 0.883	33.31 / 0.868	32.46 / 0.854	31.71 / 0.841
MCDnCNN	45.59 / 0.968	40.34 / 0.953	37.93 / 0.943	36.34 / 0.932	35.10 / 0.922	33.97 / 0.906	32.87 / 0.885	31.82 / 0.857
RED-WGAN	44.42 / 0.944	39.97 / 0.933	37.82 / 0.923	36.36 / 0.914	35.22 / 0.905	34.24 / 0.896	33.31 / 0.884	32.39 / 0.871
DRCnet-2D	46.42 / 0.985	40.77 / 0.963	38.33 / 0.949	36.76 / 0.939	35.56 / 0.930	34.57 / 0.921	33.61 / 0.911	32.65 / 0.898
DRCnet	46.26 / 0.986	40.80 / 0.965	38.46 / 0.952	36.97 / 0.943	35.84 / 0.935	34.93 / 0.928	34.14 / 0.920	33.33 / 0.911

[1%, 15%]. Additionally, we applied flips and rotations in the different axes of the MRI patches for data augmentation.

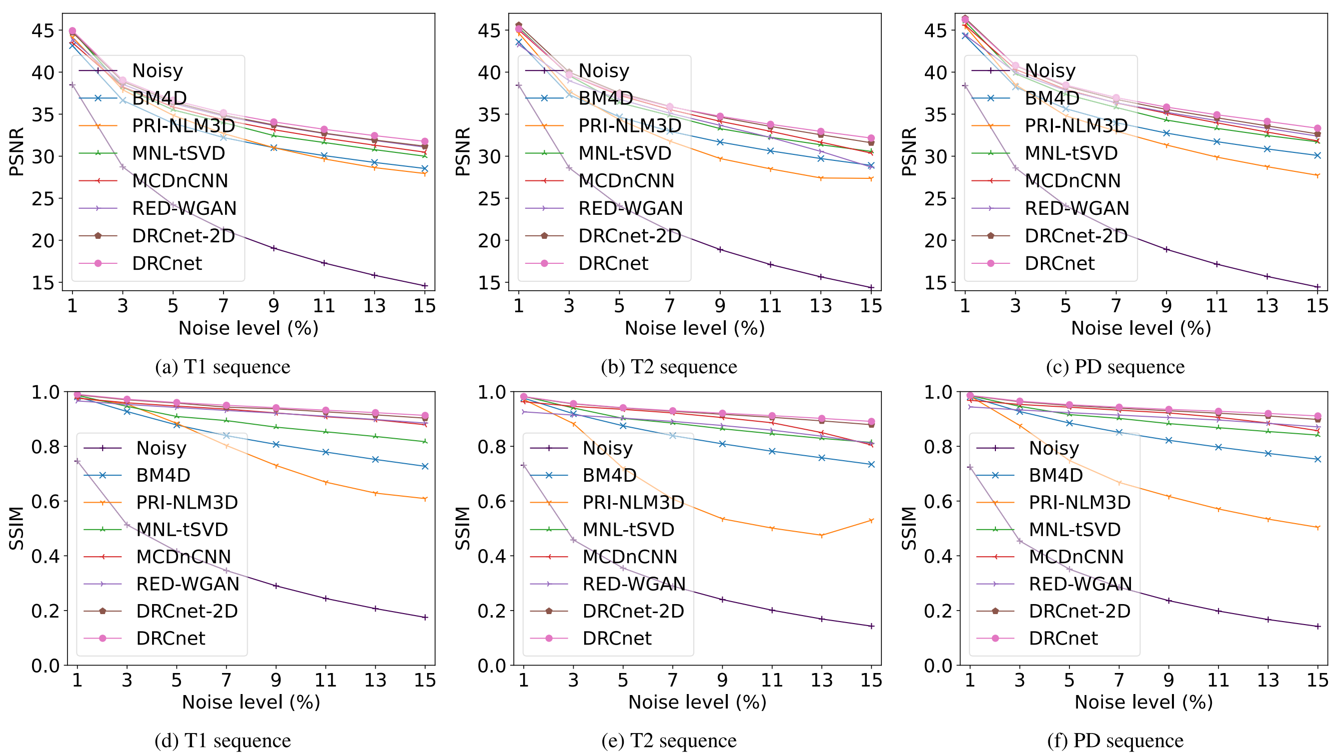
In order to optimize Eq. (3) we use the AdamW algorithm [24]. The parameters of the AdamW algorithm



(a) T1 sequence (b) T2 sequence (c) PD sequence

(d) T1 sequence (e) T2 sequence (f) PD sequence

FIGURE 3. Graphic illustration of the results in Table 4 for the IXI-Hammersmith dataset.



(a) T1 sequence (b) T2 sequence (c) PD sequence

(d) T1 sequence (e) T2 sequence (f) PD sequence

FIGURE 4. Graphic illustration of the results in Table 5 for the IXI-Guys dataset.

are $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$, and the regularization parameter in Eq. (3) $\lambda = 10^{-2}$. The initial learning rate is $\alpha_0 = 10^{-3}$, which is halved every 5 epochs.

The proposed model was trained with a batch size of 16 for 50 epochs. Our model was implemented in Python 3.6 using PyTorch framework. The training time was about

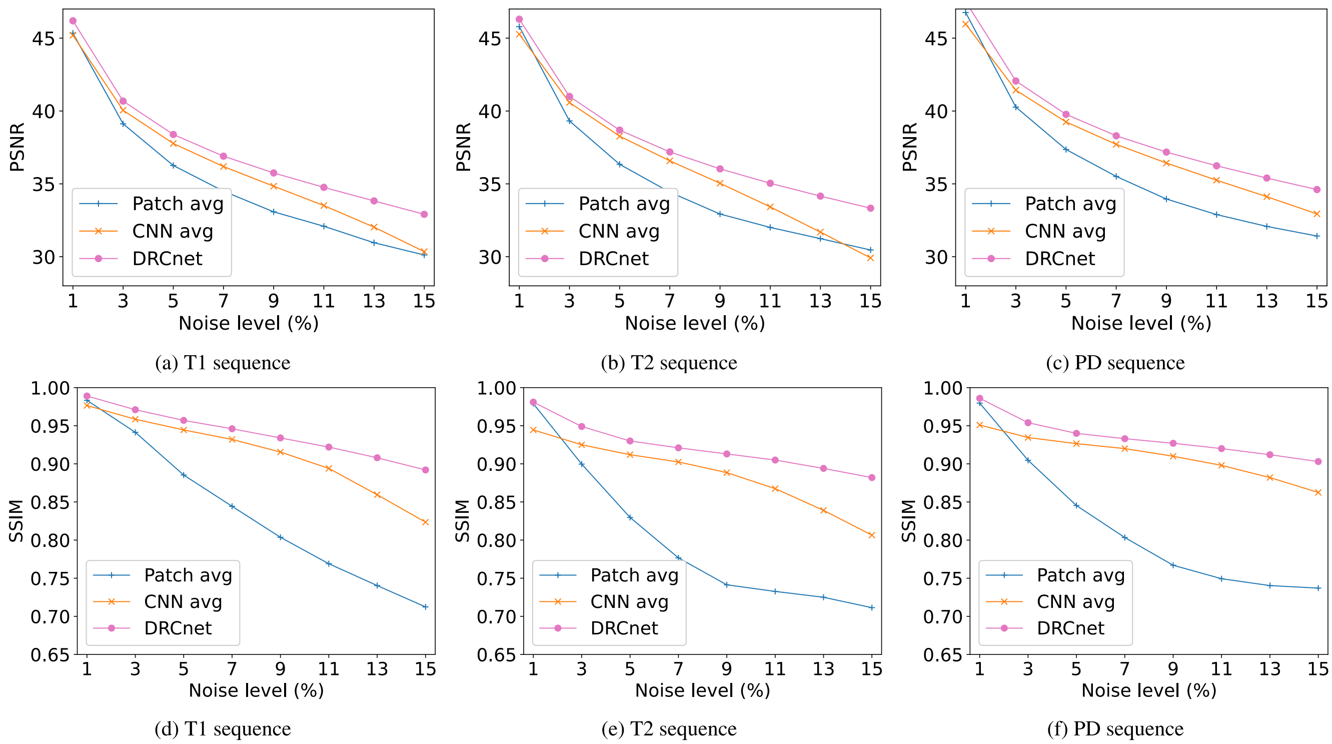


FIGURE 5. Comparison of average results between patch-based methods (Patch avg), CNN methods (CNN avg), and the DRCNet method for the IXI-Hammersmith dataset.

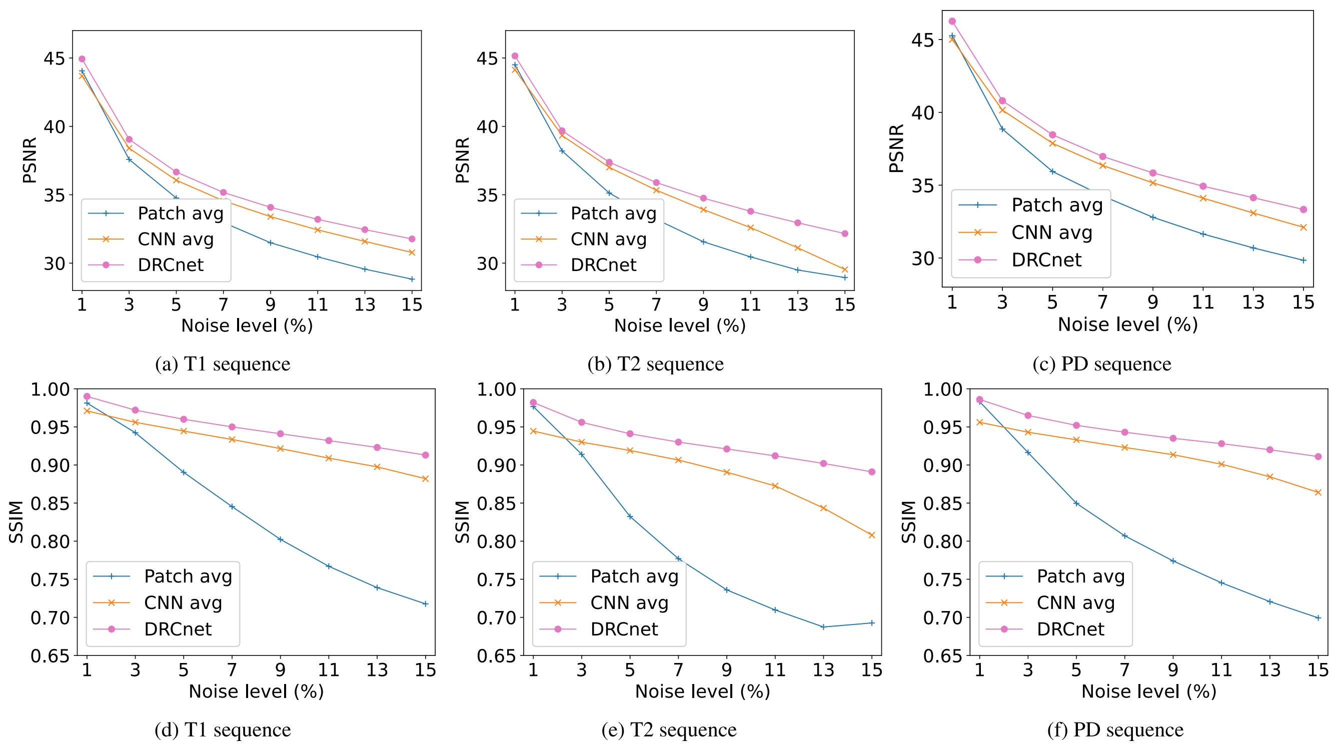


FIGURE 6. Comparison of average results between patch-based methods (Patch avg), CNN methods (CNN avg), and the DRCNet method for the IXI-Guys dataset.

20 hours in an Nvidia RTX Titan GPU. The source code, pretrained model, and dataset splits are available at GitHub (https://github.com/JavierGurrola/DRCnet).

C. RESULTS AND COMPARISONS

Tables 4 and 5 present comparative results on the IXI-Hammermith and IXI-Guy's datasets, respectively.

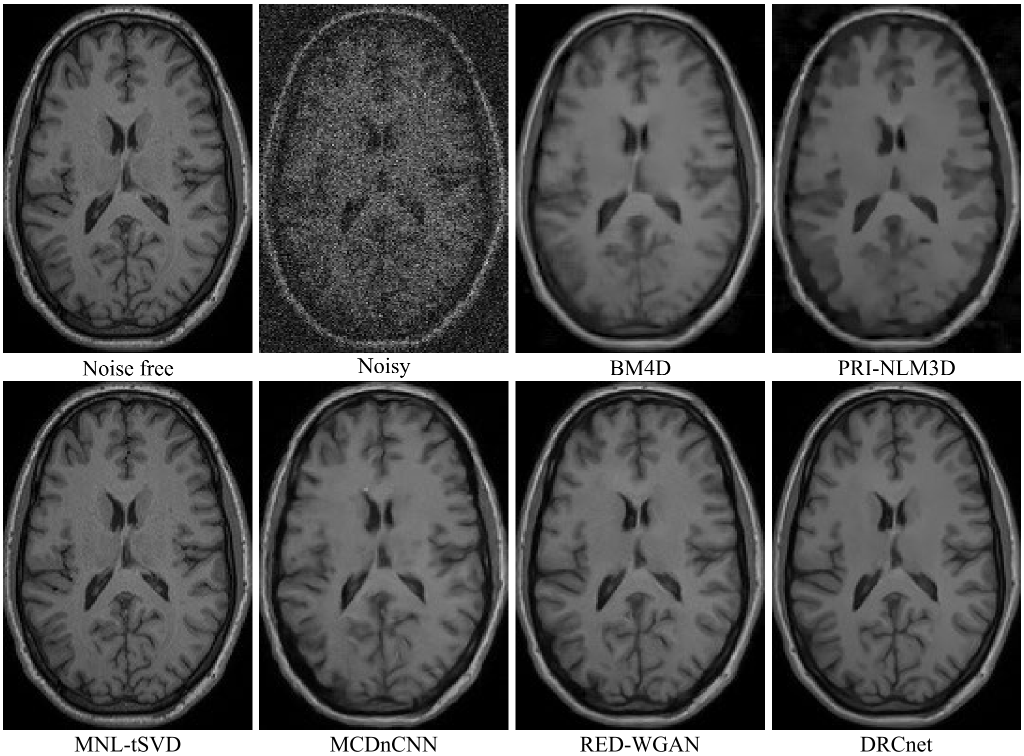


FIGURE 7. Illustration of a noise-free slice of a T1 sequence taken from an IXI-Guys image. The noisy image is obtained from the noise-free image with a noise level of 15%. The rest of the images correspond to the denoised results using different methods.

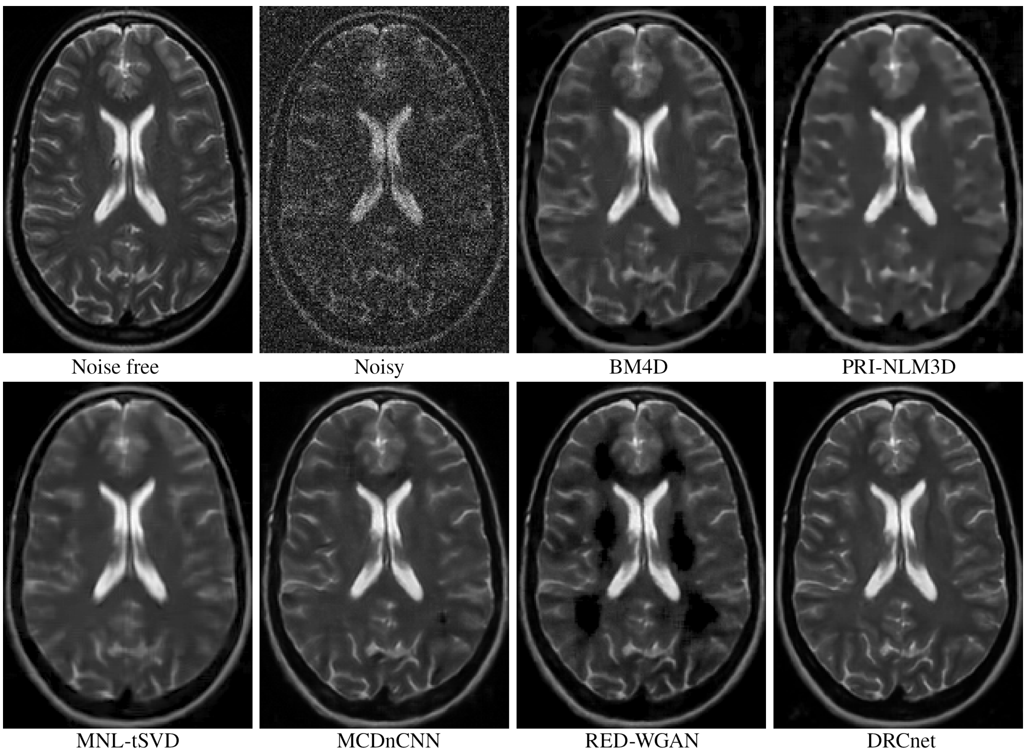


FIGURE 8. Illustration of a noise-free slice of a T2 sequence taken from an IXI-Hammersmith image. The noisy image is obtained from the noise-free image with a noise level of 15%. The rest of the images correspond to the denoised results using different methods.

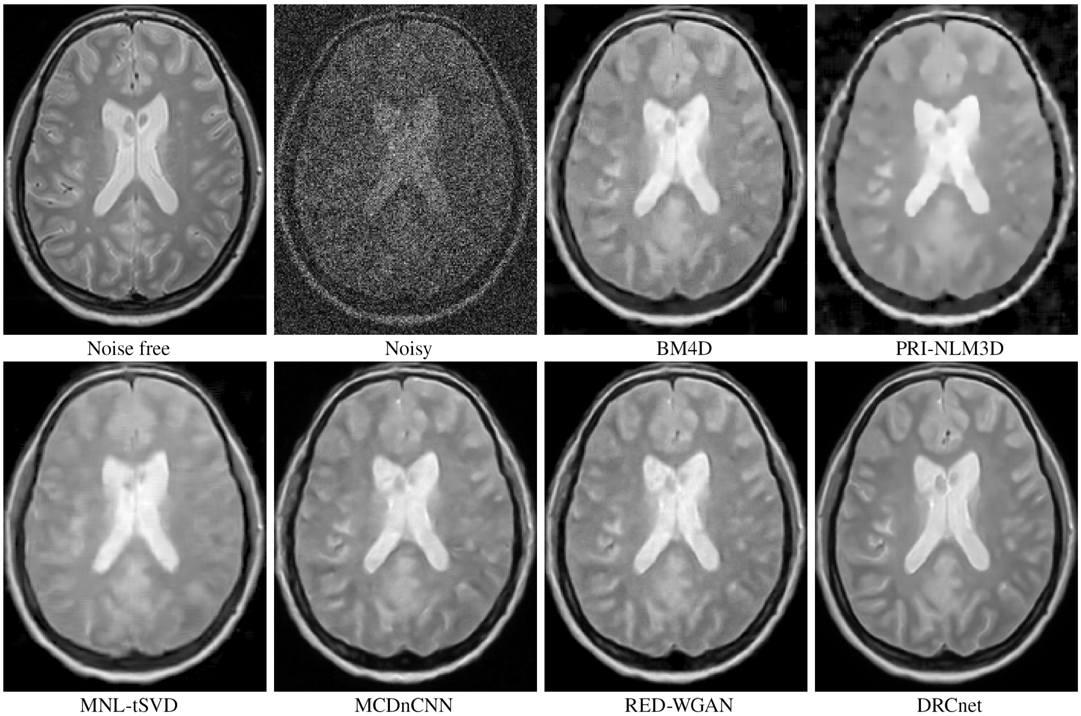


FIGURE 9. Illustration of a noise-free slice of a PD sequence taken from an IXI-Hammersmith image. The noisy image is obtained from the noise-free image with a noise level of 15%. The rest of the images correspond to the denoised results using different methods.

TABLE 6. Dice coefficients of brain WM and GM tissues of a T1 image under different noise conditions for denoised and raw images, Fig. 11.

Tissue/Noise	1 %	3%	5%	7%	9%	11%	13%	15%
Noise GM	0.9652	0.9029	0.8598	0.8285	0.8028	0.7762	0.7590	0.7360
Denoised GM	0.9706	0.9489	0.9345	0.9206	0.9102	0.8874	0.8779	0.8693
Noise WM	0.9643	0.9052	0.8633	0.8273	0.7914	0.7545	0.7444	0.7279
Denoised WM	0.9667	0.9470	0.9338	0.9219	0.9137	0.9017	0.8928	0.8842

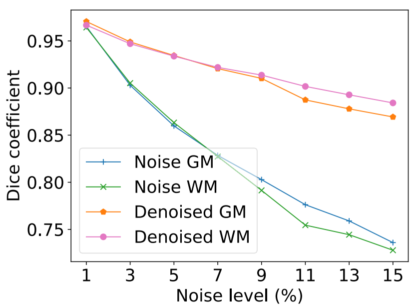


FIGURE 10. Dice coefficients of brain WM and GM tissues of a T1 image under different noise conditions for denoised and raw images, Fig. 11.

Figs. 3 and 4 show the graphical corresponding to the numerical results in Tables 4 and 5 respectively. It can be seen that the classical methods have a good performance for low noise levels, in particular for 1% and 3%, obtaining in some cases a better performance than some CNN-based models. However, when the level of noise increases, the CNN-based models have a better performance than the classical methods in general. Note the CNN-based methods are superior in both PSNR and SSIM metrics, although the MNL-tSVD method is very competitive. On the other hand, the other classical techniques notably reduce their performance while increasing the noise level. Observe that, the CNN-based methods have more stable behavior. Their performance does not decrease as drastically as the classical methods when the noise level is increased. Note that the two proposed models, in 3D and 2D, present the best performance in general, in the three modalities T1, T2, and PD, and the PSNR and SSIM metrics, showing stable results as the noise level increases. Figs. 5 and 6 depict a comparison between the average of patch-based methods BM4D, PRI-NLM3D, and MNL-tSVD (Patch avg), the CNN-based methods MCDnCNN and RED-WGAN (CNN avg), and the proposed method DRCnet. Average results are calculated from Tables 4 and 5. It can be observed again that on average the patch-based methods (Patch avg) have a good performance for low noise levels. However, when the level of noise increases, the performance on average of CNN-based models (CNN avg) is better than the patch-based methods. The improvement is better observed for the SSIM metric. It can also be seen that the DRCnet method achieves the best results in both metrics.

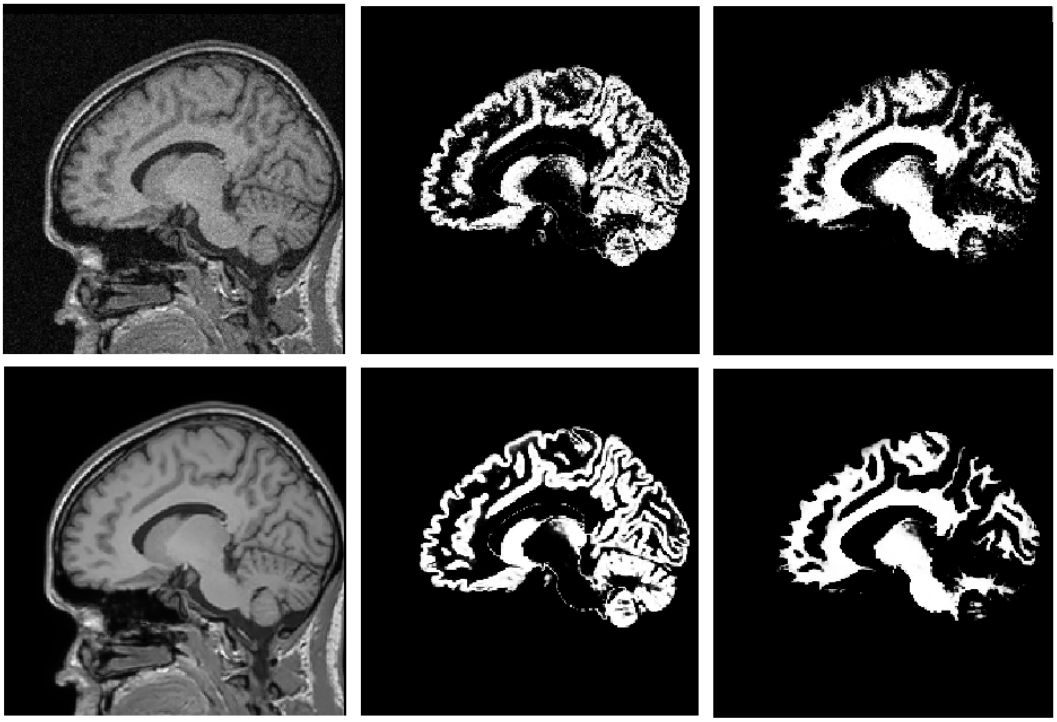


FIGURE 11. Visual example of the impact of denoising in the segmentation process. In the upper row, a noise T1 (3 %) and its corresponding tissue maps are shown. In the lower row, the same subject filtered with the proposed filter and the corresponding tissue maps are shown. As can be noticed the tissue probability maps are visually more consistent after the denoising process.

Note from Tables 4 and 5 that the numerical results of all methods are higher on the Hammersmith dataset than on the Guys dataset, i.e., all methods achieved better performance with the best quality data set, which were obtained in this case with a magnetic field strength of 3T. Figs. 7, 8 and 9 depict some visual results of the compared models applied to MRI images of T1, T2 and PD sequences from the IXI Guy and Hammersmith datasets. In all cases, the noisy image is obtained from the noise-free image with a noise level of 15%. Note that self-similarity-based methods tend to oversmooth images, whereas CNN-based methods better preserve details. We observe that the proposed method achieves excellent visual results, and is able of reducing noise while preserving details.

It is worth mentioning that the numerical outcomes of classical methods could be improved by tuning the hyperparameters for new databases. Nevertheless, tuning hyperparameters is not always an easy task and its complexity increases with number of hyperparameters, which could be a challenging and time-consuming task. On the other hand, the CNN-based methods presented here have been trained to reduce the Rician noise for each image sequence T1, T2, and PD. If the type of noise changes, for example in the case of multichannel coils, or the image sequence is different, then we need to retrain the models, which could be a limitation of these models, including our proposals. Another alternative to address these problems is to apply fine-tuning to existing models for a new type of noise. In the case of new MRI image sequences, we can also apply transfer learning. For this, we can use pretrained models on larger datasets so that we can improve denoising performance on medical image datasets, even using smaller datasets (few-shot learning). Transfer learning and few-shot learning have been recently used to improve generalization for different machine learning tasks in different domains [25], [26], [27].

D. COMPLEXITY OF CNN MODELS

Table 3 shows the number of trainable parameters of the compared CNN-based models and the proposed model. In the case of 2D model, we first compute the MACs and time results for 64×64 images and then we report the previous result multiplied by 64. That is, it is necessary to estimate 64 slices to compare all the models fairly. The results for 3D models correspond to $64 \times 64 \times 64$ volumes. Note that even though MCDnCNN has more parameters than the DRCnet-2D, it requires less execution time and MACs to estimate an image. This is due to the MCDnCNN model does not perform any recurrence during the estimation of the image. On the other hand, note that the DRCnet is only 45% larger than the DRCnet-2D, and it requires less time to estimate the test volume, considering that the DRCnet-2D needs to estimate 64 slices independently. We also measure the average inference time for an image of IXI dataset with size $256 \times 256 \times 150$. In this case, the average CPU time is

2908 ms and the average CUDA time is 1214 ms. One of the drawbacks of CNN-based models, including our approach, is that training and inference time can be high due to the large number of operations required.

E. IMPACT OF DENOISING ON SEGMENTATION

Image denoising is a very important preprocessing step in current MRI analysis tasks. It has a significant impact on MRI inhomogeneity correction (by reducing random dispersion on specific tissues), registration (reducing also multimodal intensity distribution dispersion) and segmentation for example. To highlight the importance of the proposed denoising method we evaluate the impact of the proposed method in a well-known segmentation pipeline. For that purpose, we selected a real low-noise MRI case from IXI dataset and different amounts of random noise (1% to 15%) were added to study its impact on the segmentation process. Finally, we compared the Dice coefficient [28] of the noisy and denoised versions compared to the original low-noise MRI. To segment the IXI MRI case, we used the well-known package SPM12 and obtained the masks for gray matter (GM) and White Matter (WM).

Table 6 summarizes the results. As can be noted, see also Fig. 10, the proposed denoising method improved the Dice coefficient for all noise levels and for both tissues. In Fig. 11, a visual example of the impact of the proposed method on the estimation of the tissue maps is shown.

IV. CONCLUSION

In this paper, we presented a recurrent convolutional neural network for brain MRI denoising. The model included global residual learning in order to estimate the noise of the corrupted image instead of the noise-free image directly. As the main component, we introduced a recurrent convolutional denoising block based on GRU, which allowed us to reduce the number of trainable parameters, achieving good performance. The denoising block was combined with local residual learning to filter the noisy image recursively. According to the conducted experiments with clinical brain MRI datasets, the proposed models obtained a more stable behavior and outperformed state-of-the-models for MRI denoising. This result was most notable for medium and high noise levels in the three studied modalities, T1, T2, and PD, compared with classical denoising methods. Overall, the CNN-based models excelled in image denoising tasks, demonstrating their capability to effectively remove noise while preserving important features and details in images.

REFERENCES

[1] O. Dietrich, J. G. Raya, S. B. Reeder, M. Ingrisch, M. F. Reiser, and S. O. Schoenberg, "Influence of multichannel combination, parallel imaging and other reconstruction techniques on MRI noise characteristics," *Magn. Reson. Imag.*, vol. 26, no. 6, pp. 754–762, Jul. 2008.

[2] E. R. McVeigh, R. M. Henkelman, and M. J. Bronskill, "Noise and filtration in magnetic resonance imaging," *Med. Phys.*, vol. 12, no. 5, pp. 586–591, Sep. 1985.

[3] H. Gudbjartsson and S. Patz, "The Rician distribution of noisy MRI data," *Magn. Reson. Med.*, vol. 34, no. 6, pp. 910–914, Dec. 1995.

[4] R. D. Nowak, "Wavelet-based Rician noise removal for magnetic resonance imaging," *IEEE Trans. Image Process.*, vol. 8, no. 10, pp. 1408–1419, Aug. 1999.

[5] J. V. Manjón, P. Coupé, A. Buades, D. Louis Collins, and M. Robles, "New methods for MRI denoising based on sparseness and self-similarity," *Med. Image Anal.*, vol. 16, no. 1, pp. 18–27, Jan. 2012.

[6] L. Fan, F. Zhang, H. Fan, and C. Zhang, "Brief review of image denoising techniques," *Vis. Comput. Ind., Biomed., Art*, vol. 2, no. 1, pp. 1–12, Dec. 2019.

[7] L. Hua, Y. Gu, X. Gu, J. Xue, and T. Ni, "A novel brain MRI image segmentation method using an improved multi-view fuzzy C-means clustering algorithm," *Frontiers Neurosci.*, vol. 15, pp. 675–685, Mar. 2021.

[8] M. Prakash, T.-O. Buchholz, M. Lalit, P. Tomancak, F. Jug, and A. Krull, "Leveraging self-supervised denoising for image segmentation," in *Proc. IEEE 17th Int. Symp. Biomed. Imag. (ISBI)*, Apr. 2020, pp. 428–432.

[9] M. K. Raczkowska, P. Koziol, S. Urbaniak-Wasik, C. Paluszkiewicz, W. M. Kwiatek, and T. P. Wrobel, "Influence of denoising on classification results in the context of hyperspectral data: High definition FT-IR imaging," *Analytica Chim. Acta*, vol. 1085, pp. 39–47, Nov. 2019.

[10] M. Maggioni, V. Katkovnik, K. Egiazarian, and A. Foi, "Nonlocal transform-domain filter for volumetric data denoising and reconstruction," *IEEE Trans. Image Process.*, vol. 22, no. 1, pp. 119–133, Jan. 2013.

[11] Z. Kong, L. Han, X. Liu, and X. Yang, "A new 4-D nonlocal transform-domain filter for 3-D magnetic resonance images denoising," *IEEE Trans. Med. Imag.*, vol. 37, no. 4, pp. 941–954, Apr. 2018.

[12] N. Nazir, A. Sarwar, and B. S. Saini, "Recent developments in denoising medical images using deep learning: An overview of models, techniques, and challenges," *Micron*, vol. 180, May 2024, Art. no. 103615.

[13] P. Kaur, G. Singh, and P. Kaur, "A review of denoising medical images using machine learning approaches," *Current Med. Imag. Rev.*, vol. 14, no. 5, pp. 675–685, Sep. 2018.

[14] 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," *Jpn. J. Radiol.*, vol. 36, no. 9, pp. 566–574, Sep. 2018.

[15] M. Ran, J. Hu, Y. Chen, H. Chen, H. Sun, J. Zhou, and Y. Zhang, "Denoising of 3D magnetic resonance images using a residual encoder–decoder Wasserstein generative adversarial network," *Med. Image Anal.*, vol. 55, pp. 165–180, Jul. 2019.

[16] Y. Zhu, X. Pan, J. Zhu, L. Li, and Y. Liu, "Denoising of magnetic resonance images with deep neural regularizer driven by image prior," in *Proc. IEEE 7th Int. Conf. Data Sci. Adv. Analytics (DSAA)*, Oct. 2020, pp. 255–263.

[17] R. Singh and L. Kaur, "Magnetic resonance image denoising using patchwise convolutional neural networks," in *Proc. 8th Int. Conf. Comput. Sustain. Global Develop. (INDIACom)*, Mar. 2021, pp. 652–657.

[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 Trans. Image Process.*, vol. 26, no. 7, pp. 3142–3155, Jul. 2017.

[19] M. Elad, B. Kawar, and G. Vaksman, "Image denoising: The deep learning revolution and beyond—A survey paper," *SIAM J. Imag. Sci.*, vol. 16, no. 3, pp. 1594–1654, Sep. 2023.

[20] J. Gurrola-Ramos, O. Dalmau, and T. E. Alarcón, "A residual dense U-Net neural network for image denoising," *IEEE Access*, vol. 9, pp. 31742–31754, 2021.

[21] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2818–2826.

[22] Q. Huynh-Thu and M. Ghanbari, "Scope of validity of PSNR in image/video quality assessment," *Electron. Lett.*, vol. 44, no. 13, p. 800, 2008.

[23] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[24] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *Proc. 7th Int. Conf. Learn. Represent.*, 2019, pp. 1–24.

[25] M. Kumar, V. Kumar, H. Glaude, C. de Lichy, A. Alok, and R. Gupta, "Protoda: Efficient transfer learning for few-shot intent classification," in *Proc. IEEE Spoken Lang. Technol. Workshop (SLT)*, Jan. 2021, pp. 966–972.

[26] T. T. Chungath, A. M. Nambiar, and A. Mittal, "Transfer learning and few-shot learning based deep neural network models for underwater sonar image classification with a few samples," *IEEE J. Ocean. Eng.*, vol. 49, no. 1, pp. 1–17, May 2023.
[27] H. Wei and L. Jiao, "A survey of few-shot image classification based on transfer learning," in *Proc. IEEE 6th Int. Conf. Pattern Recognit. Artif. Intell. (PRAI)*, Aug. 2023, pp. 461–469.
[28] A. P. Zijdenbos, B. M. Dawant, R. A. Margolin, and A. C. Palmer, "Morphometric analysis of white matter lesions in MR images: Method and validation," *IEEE Trans. Med. Imag.*, vol. 13, no. 4, pp. 716–724, Jul. 1994.

JAVIER GURROLA-RAMOS received the B.Eng. degree in computer engineer from UAA Aguascalientes, Mexico, in 2016, and the M.Sc. degree in computer science and industrial mathematics from the Mathematics Research Center (CIMAT), Guanajuato, Mexico, in 2018, where he is currently pursuing the Ph.D. degree in computer science. His research interests include machine learning, optimization, image processing, and pattern recognition.

TERESA ALARCON (Member, IEEE) received the Engineer degree in automated systems of management from Moscow Institute of Direction "Sergo Orchonikidze," in 1989, Russia, the master's degree in digital image processing from the "José Antonio Echeverría" Polytechnic Institute, Cuba, in 1999, and the Ph.D. degree in computer science from the Center for Mathematics Research, Guanajuato, Mexico, in 2007. She is currently an Associate Professor with the Computational Sciences and Engineering Department, University of Guadalajara's Valley Campus, Ameca, Jalisco. Her research interests include digital image processing, including filtering, segmentation, and pattern recognition.

OSCAR DALMAU received the B.Sc.Ed. degree in mathematics from ISP, Manzanillo, Cuba, in 1989, and the M.Sc. degree in computer science and industrial mathematics and the Ph.D. degree in computer science from the Mathematics Research Center (CIMAT), Guanajuato, Mexico, in 2004 and 2010, respectively. He is currently with CIMAT. His research interests include machine learning, optimization, image processing, and computer vision.

JOSÉ V. MANJÓN is currently an Associate Professor with the Universidad Politécnica de Valencia, Spain, where he leads the medical image processing area of the IBIME Research Group. He is one of the pioneers in patch-based medical image processing and has developed several state-of-the-art algorithms for MRI denoising, super-resolution, segmentation, and automatic diagnosis.