Conditional GAN for brain MRI denoising in the k-space

Michela Proietti[1] (Project author), Sofia Santilli[1] (Project author), Christian Napoli[1] (Professor), Stefano Giagu[1] (Professor) and Giorgio De Magistris[1] (Ph.D.)

[1] *Sapienza University of Rome, Italy*

Abstract

MRI is one of the most widely used imaging techniques, since it allows to collect information about organs and tissues in a completely non-invasive way. However, despite the recent advancements in MRI technology, the acquisition of high-resolution MR images requires a long time. For this reason, a lot of studies focused on finding a way to improve the quality of low-SNR images, by developing different denoising techniques. However, it would be ideal to remove noise before reconstructing the actual MR scan, using k-space data, while it is still affected by white noise with a Gaussian probability distribution. Therefore, this paper proposes a denoising Conditional Generative Adversarial Network (CGAN) that is trained on k-space data. The results are then compared with those of a simpler Denoising Convolutional Neural Network (DnCNN), which will serve as a baseline.

Keywords

Denoising CGAN, MRI, residual learning, k-space, Dn-ResNet, skip connections

1. Highlights

- Denoising brain MRI in the k-space, using different noise levels (low, high, blind).
- Using a Conditional Generative Adversarial Network with a Dn-ResNet as generator and a CNN as discriminator.
- Using skip connections in the generator to widen the network's receptive field in order to capture fine structures in the images.
- Making comparisons with the results obtained using a DnCNN as baseline.
- Achieved both quantitative and qualitative improvements.

2. Introduction

Magnetic Resonance Imaging (MRI) is a mechanism, developed in the 1970s, to encode spatial information into a Nuclear Magnetic Resonance signal using magnetic field gradients. This medical imaging technique is used in radiology to provide highly detailed images of tissues, organs and of physiological processes of the human body. It is widely used for medical diagnosis, staging and follow-up of a disease. Compared to other techniques, MRI can discriminate among soft-tissues, exploiting physical and biochemical properties (e.g. in the brain or abdomen). This ability to contrast between grey and white matter, together with MRI's high spatial resolution, makes this method one of the most common choices to analyse the anatomical structure and many conditions of the central nervous system, including demyelinating diseases, dementia, cerebrovascular disease, infectious diseases, Alzheimer's disease and epilepsy. Moreover, its noninvasiveness makes it less dangerous than other modalities, such as ultrasounds, Positron Emission Tomography (PET) or Computed Tomography (CT).

Nevertheless, although MRI has such good properties and over the recent years the resolution, signal-to-noise ratio and acquisition speed of MRI technology have been improved, the quality of this data is usually deteriorated by the presence of noise and artifacts. This is an issue, as noise in MRI can cause inaccurate diagnoses, putting patients at risk. However, noise is implicit in the acquisition process, since when taking an MRI of a living subject there are multiple noise sources, including noise from stochastic variation, several physiological processes, eddy currents, artifacts from the magnetic susceptibilities between neighboring tissues, rigid and nonrigid body motions, also due to the long acquisition time[1], and many others [1, 2]. Thermal noise is the main type of noise that is present in MR images, and it is caused by the body temperature and movement, and by the MRI instrumentation itself, such as RF coils, transmission lines and receiver circuits.

Because of the inevitability of noise in MR images, image

proietti.1739846@studenti.uniroma1.it (M. Proietti); santilli.1813509@studenti.uniroma1.it (S. Santilli); c.napoli@uniroma1.it (C. Napoli); stefano.giagu@uniroma1.it (S. Giagu); demagistris@diag.uniroma1.it (G. D. Magistris)
https://github.com/michelaproietti (M. Proietti); https://github.com/Sofia-Santilli/ (S. Santilli); https://sites.google.com/diag.uniroma1.it/napoli (C. Napoli); https://sites.google.com/a/uniroma1.it/stefanogiagu/ (S. Giagu); https://phd.uniroma1.it/web/DE-MAGISTRIS-GIORGIO_nP1645068.aspx (G. D. Magistris)

CEUR Workshop Proceedings (CEUR-WS.org)

[1] Usually an MR scan takes at least 20 minutes and can go on for an hour or more sometimes.

denoising is a fundamental step of image preprocessing, that is needed both in the case of experts' and automatic diagnoses. Many different techniques have been proposed in the literature in order to properly denoise MR images. Traditional techniques are generally based on using linear or nonlinear filters. Nonetheless, with the advent of deep learning, some models have managed to outperform traditional approaches and have been demonstrated to work well even in the presence of uncertain noise types.

In order to improve the quality of the denoised images, it would be useful to take some properties of the acquired signal into account. In fact, the raw data obtained during MRI acquisition are complex values that consist in the Fourier transform of a magnetization distribution of a volume of tissue, and this represents the so called k-space. By using a two dimensional inverse Fourier transform, this data is converted into magnitude, phase and frequency components, that more directly represent the physiological and morphological features of interest in the human body. Because of the linearity and orthogonality properties of the Fourier transform, noise in the k-space in the MR data from each coil is assumed to be an additive white noise with a Gaussian probability distribution, characterized by zero mean and equal variance in both real and imaginary parts. However, the denoising algorithms we have previously mentioned have been applied to the reconstructed magnitude images, since the phase information is usually discarded to avoid phase artifacts. Due to the fact that the magnitude of the MRI signal is computed by a non-linear operation, namely the square root of the sum of the squares of two independent Gaussian variables, the noise manifests itself with a different distribution of pixel intensities: in the case of a single channel acquisition, it is assumed to be a Rician distribution, while in a multichannel signal it is described by a noncentral Chi distribution. Since it is much more difficult to deal with these kinds of noise, it would be significantly more convenient to work with a Gaussian noise, performing denoising on k-space data.

For this reason, we present a denoising Conditional GAN that directly takes as input k-space data and its structure will be explained in details in the next sections.

3. Related Works

In general, there are two typical ways to reduce noise in images: acquisition-based noise reduction methods and post-acquisition methods. The first category of approaches consists in decreasing the spatial resolution or in acquiring the data several times and averaging the different acquisitions, thus increasing the acquisition time. On the contrary, post-acquisition methods denoise images after they have been completely acquired. In the following, we will focus on this second type of approaches, presenting the most widely used ones.

First of all, we present filtering methods, that consist in applying different types of filters to the images. Among the linear filtering approaches for Gaussian noise reduction proposed in 1985 by McVeigh et al. [3], we need to make a distinction between spatial and temporal filters. Spatial filtering consists in simply applying a convolution between an image and a filter. In this way, the variance is reduced but the signal-to-noise ratio remains unaltered, since both the signal and the noise are reduced by the same factor. This causes the blurring of sharp edges, bringing to a loss of spatial resolution. Differently, temporal filters are exploited to avoid aliasing, but in order to do this and to avoid the loss of signal at edges they need a proper selection of frequency response.

An alternative to linear filtering consists in using nonlinear filters. For instance, a method that significantly improved the image quality with respect to the results obtained through spatial filters is Anisotropic Diffusion Filtering (ADF), which was developed by Perona and Malik [4]. It is an iterative denoising process based on the resolution of second order partial differential equations (PDE). Specifically, smoothing is used for noise removal from homogeneous regions, while the process is stopped at boundaries, by selecting the local gradient strengths in different directions: the image is convolved only in the direction orthogonal to the gradient of the image, thus preserving and sharpening the edges. On the contrary, disadvantages in using ADF consist in the erase of small features and in the transformation of image statistics, which cause blocky staircase effect in the image. Successively, Tomasi and Manduchi [5] presented the Bilateral Filter, which is similar to ADF in the edge preservation and the smoothing effect, but that is a non-iterative alternative and doesn't involve PDF resolution. Instead, it consists in the combination of two filters, weighted by some similarities that depend on the properties of the chosen filters.

Later, Buades et al. [6] proposed Nonlocal Mean Filter (NLM) that generally removes small structures, mistakenly considered as noise, but is able to preserve large scale structures, by exploiting the redundancy of information. More specifically, the new value of each pixel i is computed as a weighted average of all the pixels in the image, $j \in I$, in which weights are not arbitrary but depend on the similarity of the intensity gray level vectors between the two considered pixels, i and j. Some examples of applications are those in [6], who used NLM for Gaussian noise removal in 2D natural images, and Manjon et al. [7], who applied the filter for denoising multispectral MR images. The main disadvantage of NLM is its computational burden due to the complexity of calculating the weight of the pixel. Coupe et al. [8, 9] proposed an optimized implementation for denoising 3D MRI, managing in decreasing the computational time up to the factor of 50.

Another type of denoising methods is that of Transform Domain approaches. One possibility is to perform a multiresolution analysis (MRA) of the Wavelet Transform, which consists in analysing the signal at different frequencies, giving different resolutions. Using the Wavelet transform allows to filter out most of the noise, while preserving the edges and fine details. Again, the main drawback is that small structures that are similar in size to the noise are eliminated, so the Wavelet-transform-based denoising methods are not suitable for describing the signals which have high dimensional singularities, such as edges. To address this problem, we could alternatively use the Curvelet Transform, which was proposed for denoising MR images by Ashmol et al. [10]. Moreover, the wavelet transform is useful in representing images containing smooth areas which are separated with edges. However, it cannot perform effectively when the edges are smooth curves. In this case, the Contourlet Transform, which is able to capture contours and fine details in the images, could instead be used.

A few works have been also proposed for MRI denoising through statistical approaches, that are based on the estimation of the noise variance in MR images. Some of the most common methods that fall in this category are those proposed by Sijbers et al. [11], who estimated the Rician noise level and performed signal reconstruction using the maximum likelihood (ML) approach, Aja-Fernandez et al. [12] who used the linear minimum mean square error (LMMSE) estimator for Rician noise, and Tisdall and Atkins [13], who proposed the phase error estimation scheme for MRI denoising.

Lately, many deep learning architectures have been proposed for noise and artifact reduction. In particular, Convolutional neural networks (CNN) are widely used for noise reduction, because their small receptive field is suited for removing Gaussian noise, which is incoherent and position-independent. A common architecture is that of Denoising convolutional neural networks (DnCNN), which learns residual components of images, that correspond to noise. Batch normalization is usually used to make the training procedure faster. Differently from Gaussian noise, artifacts are globally distributed, so networks with a larger receptive field are needed. For this reason, ResNet is commonly used, since it has a large receptive field that covers whole input images owing to a large number of layers. Also autoencoders are very useful for extracting significant information for high-dimensional datasets, since they compress the entire image information into a low-dimensional structure to effectively learn the underlying manifold. In fact, after Gondara [14] first developed an autoencoder-based filter for X-ray images, many studies have been proposed for CT, MRI and others [15, 16]. Finally, U-net has also been used, especially to extract artifacts. In fact, it has a large receptive field by utilizing multi-scale features. Moreover, it contains skip connections, that allow to capture localized features, such as fine structures of images. U-net has been also used as generator in Generative Adversarial Networks (GAN) [17], that have been proven to be very promising for noise and artifact removal.

4. Dataset

The proposed denoising approach has been applied to the FastMRI dataset [18]. It contains fully sampled k-space acquisitions of knees and brains, consisting of raw multicoil k-space data of unprocessed complex valued NMR measurements. The original purpose of this dataset is to be used by algorithms that perform image reconstruction of frequency subsampled data. However, it can be easily exploited for a denoising task, since the acquisition modality of this samples is characterised by an higher quality than the one commonly available in clinical practice. This leads to almost noise-free samples, that have been used as ground truth during training. Actually, some of the acquisitions can be slightly noisy, but they are a minority and the noise is negligible with respect to the noise successively added for training purposes. This imperfection also allows to improve robustness of the denoising network.

The FastMRI dataset is divided into training, validation and masked test sets. For the purposes of the work only brain MRI have been considered, and masked test sets were ignored. Each file in the training and validation sets is an HDF5 file, organized in two tensors. The first represents a multi-coil k-space data, with shape (number of slices, number of coils, coil height, coil width). The second is the corresponding reconstructed image, with shape (number of slices, height, width), computed as:

$$M_T^2 = \sum_{l=1}^{L} |M_l(x)|^2 \tag{1}$$

which corresponds to the root of the sum of square (SoS) of the single magnitude images reconstructed independently from each coil.

Because of memory limitations on the working platform, the training set was not available. Therefore, the validation set of fastMRI has been split: the 80% was used for the training dataset, the 12% for the validation dataset and the remaining 8% for the test dataset.

5. Data preparation and preprocessing

In order to reduce computational load and allow faster trainings, subsampling is performed, by randomly selecting a certain number of samples from the respective sets: 350 patients for training, 80 for validation and 80 for testing. For the same reason, only eight coils for each slice are considered, while the slices with a lower number of coils are completely discarded. Regarding the slices, from each HDF5 file, two thirds of the total number of slices are selected, thus preserving only the most significant ones. In fact, the slices from each HDF5 file consist in the cross sections at different levels of the brain and only the deepest ones contain relevant information. After subsampling, the real preprocessing is performed. Each coil is initially normalized with respect to the maximum value of the modulus of the coil itself. Then, the acquisition size is cropped to 220x114, mainly for two reasons. First, this allows to remove high frequency components, that mostly represent noise. Secondly, by reducing the size of the samples, we are able to reduce the space in memory that is needed for training.

With a 50% probability, we also perform data augmentation on samples in the training set. Specifically, with different probabilities, we pick one of the following transformations:

- Phase multiplication, performed as $S(x,y) = S(x,y) * e^{i\theta}$, with S being the k-space signal and $\theta \in [0, 2\pi]$;
- Horizontal flipping, computed as $S(x,y) \rightarrow S(-x,y)$;
- Vertical flipping, computed as $S(x,y) \rightarrow S(x,-y)$, and half of the times it is followed by horizontal flipping, thus getting $S(x,y) \rightarrow S(-x,-y)$;

Once the noise-free dataset is obtained, we generate corrupted samples by adding complex white noise in the frequency domain. The noise standard deviation in the k-space, σ_0, is chosen in order to have images with a SNR comprised between 2 and 5, when measured on the corresponding magnitude images. In a multiple coil system, the additive Gaussian noise model is still valid at the coil level, but because of the presence of a mutual inductance between the coils, there is a strong correlation in the additive noise term. For this reason, the noise is modelled as a multivariate Gaussian with a non-diagonal covariance matrix, in the form:

$$R = \begin{bmatrix} 1 & \rho_{12} & \dots & \rho_{1L} \\ \rho_{21} & 1 & \dots & \rho_{2L} \\ \dots & \dots & \dots & \dots \\ \rho_{L1} & \rho_{L2} & \dots & 1 \end{bmatrix} \quad (2)$$

where each value $\rho_{i,j}$ expresses the covariance between coils i and j and depends on the particular model of MRI scanner. In our work, we make an approximation using a circular geometry in which the correlation between coils is $\rho_{i,j} = 0.3$ if the coils are first neighbours, $\rho_{i,j} = 0.15$ if they are second neighbours and $\rho_{i,j} = 0.05$ otherwise. Having the correlation matrix, we compute the covariance matrix as:

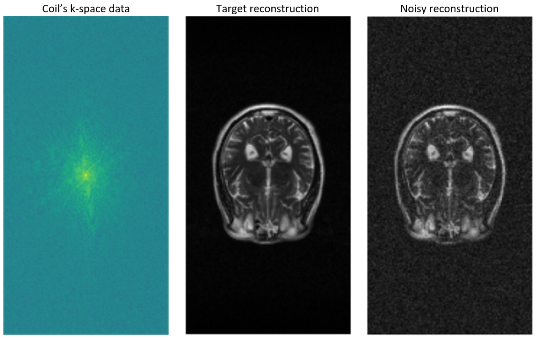

Figure 1: From left to right: k-space data of a single coil of the slice; reconstruction of the slice, starting from the target coils; reconstruction of the slice, starting from the noisy coils.

$$C = \sigma_0 \, R \quad (3)$$

Depending on the value chosen for the standard deviation, σ_0, we created three datasets, each characterised by a different level of noise, resulting in different values of PSNR and SSIM for the reconstructed images, which are reported in Table 1. PSNR and SSIM are the metrics that are used to evaluate the resulting denoised images, and they will be discussed in details in the following section. We want to highlight that the blind dataset is obtained by randomly picking for each sample a value for the standard deviation that is between those used to build the low-noise and high-noise datasets.

Table 1
Values of standard deviations used to create the different corrupted datasets, with the corresponding values of the metrics obtained for the reconstructed noisy images.

Noise level	σ_0	PSNR	SSIM
High	3 $\times 10^{-3}$	17.5 $\pm$5.3	0.24 $\pm$0.11
Low	1.5 $\times 10^{-3}$	24.3 $\pm$6.4	0.45 $\pm$0.28
Blind	1.5-3 $\times 10^{-3}$	20.5 $\pm$8.4	0.32 $\pm$0.29

Finally, both target and corrupted slices are linearly scaled in images with mean 0 and variance 1. Figure 1 shows the information content of a sample in the datasets.

6. Metrics

The performances of the denoising techniques have been evaluated based on two quantitative performance metrics and in terms of the visual quality of the image. Since the noise-free image is available, the Peak Signal-to-Noise Ratio (PSNR) can be used to quantify the quality of the magnitude image. PSNR is defined as:

$$PSNR = 20 \cdot \log_{10}\left(\frac{MAX_I}{\sqrt{MSE}}\right) \tag{4}$$

where MAX_I is the maximum value a pixel can assume, and it is expressed in decibel (dB). MSE is the Mean Squared Error that, for a pair of NxM real images, I and I*, is written as:

$$MSE = \frac{1}{N \cdot M}\sum_{i=1}^{N}\sum_{j=1}^{M}[I(i,j) - I*(i,j)]^2 \tag{5}$$

Here MSE is just the mean square euclidean distance in the pixel space computed between the target image and the denoised one.

The other metric that is used is the Structural Similarity Index Measure (SSIM), which measures the similarity between two images by exploiting the inter-dependencies among nearby pixels. SSIM is inherently able to evaluate structural properties of the objects in an image and is computed at different image locations by using a sliding window. The SSIM between the original image patch m and the patch $\hat{m}$ of the denoised image is computed by applying a sliding filter, and it is defined as:

$$SSIM(m,\hat{m}) = \frac{(2\mu_{\hat{m}}\mu_m + c_1)(2\sigma_{\hat{m}m} + c_2)}{(\mu_{\hat{m}}^{\;2} + \mu_m^2 + c_1)(\theta_{\hat{m}}^{\;2} + \theta_m^2 + c_2)} \tag{6}$$

where μ is the intensity mean of the patch, σ is the standard deviation, $\sigma_{\hat{m}m}$ is the covariance, and c_1, c_2 are small regularizing constants, that are set by default to 0.01 and 0.03. Additionally, the filter size is set to the default value of 11×11.

7. Model

Since GANs seem to be a very promising technique for removing noise and artifacts but they have not been used for denoising in the k-space, we decided to adopt them in our approach. A GAN consists of two separate networks: generator and discriminator. The generator is the network that creates the denoised images, whereas the discriminator acts as a classifier to determine whether the image given as input is a real noiseless image or a denoised one. The feedback from the discriminator contributes to updating the generator in order to create convincing denoised images that are able to better fool the discriminator. In particular, the developed model consists in a Conditional Generative Adversarial Network (CGAN), that exploits a Denoising Residual neural network (Dn-ResNet) as generator and a simple Convolutional neural Network (CNN) as discriminator. The

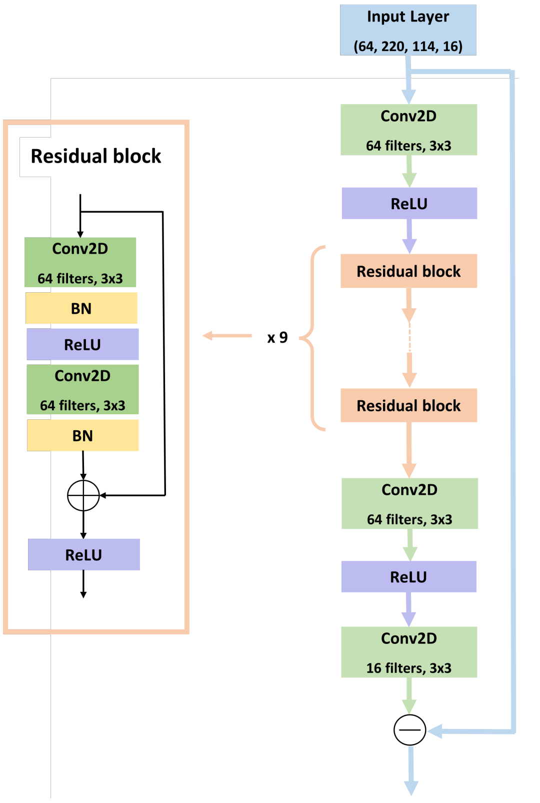

Figure 2: Structure of the generator, which is a Dn-ResNet. The input has shape (batch_size, image_height, image_width, n_channels).

Dn-ResNet is a deep convolutional neural network (CNN) that has to learn to estimate the noise content and remove it. The network's input is represented by the concatenation of the real and imaginary parts of the complex k-space data for each coil. Therefore, we have a total of 16 channels, two for each of the eight coils selected from a single slice. The generator's structure is shown on the right of Figure 2. Specifically, it consists of an initial Conv2D layer with 64 3×3 filters, 9 residual blocks (ResBlocks), a Conv2D followed by Batch Normalization and Relu, and a final Conv2D layer with 16 3×3 filters. Each residual block, shown on the left of Figure 2, is characterised by a skip connection that jumps two layers and allows to sum the input of the ResBlock to the output of the intermediate layers and ReLU is then applied to the resulting tensor. Each convolutional layer in the Residual block has 64 3×3 filters. Finally, a residual learning connection, that traverses the whole generator network, computes a subtraction between the noisy input of the Dn-ResNet and the output of the last Conv2D, which corresponds to the extracted noise. The exploitation of residual learning allows both to handle the vanishing gradient problem and to capture fine structures in the image. Furthermore, residual learning together with batch normalization are exploited to speed up the training process as well as boost the denoising performance. Since Gaussian noise is incoherent and position-independent, a small receptive field is usually preferred for noise reduction. For this aim, pooling, strides and dilation were avoided in the network. The loss function exploited in order to train the generator is the following:

$$L_g = MSE(S_y, S) + \beta \cdot MSE(Sos(S_y), M) + \gamma \cdot L_{dp} \tag{7}$$

$$L_{dp} = BCE(1, D(S_y)) \tag{8}$$

where S_y is the 16 channels output of the Dn-ResNet, S is the ground truth signal in the k-space, $Sos(S_y)$ is the reconstruction of the predicted slice, M is the ground truth reconstructed image. L_{dp} is the Binary Cross Entropy computed on the classification performed by the discriminator on S_y. In fact, if the discriminator manages in correctly classifying S_y as output of the generator, L_{dp} will be higher, thus increasing the generator loss; if the discriminator fails, it means that the generator is becoming good in denoising, so its loss decreases.

As discriminator, a simple CNN is exploited. The structure is shown in Figure 3. It consists in three convolutional layers, with respectively 64, 128 and 256 3×3 filters, all followed by a LeakyRelu activation function. It follows a flatten layer, a 0.4 dropout, a 64-units dense layer, a LeakyRelu function and a final single unit dense layer. The loss function used to train the discriminator is:

$$L_d = BCE(1, D(Sos(S))) + BCE(0, D(Sos(S_y))) \tag{9}$$

The Binary Cross Entropy is as small as higher is the ability of the discriminator in distinguishing if a reconstructed image is the target one or the denoised one. As specified before, we have implemented a Condidional GAN, which means that every time we give as input to the discriminator a target image or a denoised one, we concatenate to that the noisy reconstruction. In this way, while discriminating the denoised images, the discriminator takes into account also the original input given to the generator.

8. Experiments and Results

Overall, training a GAN is very hard because we need to simultaneously train the generator model and the discriminator model, without making one of the two improve too much with respect to the other. For this reason, most of the experiments were aimed at finding the best combination of hyperparameters that guaranteed a stable training.

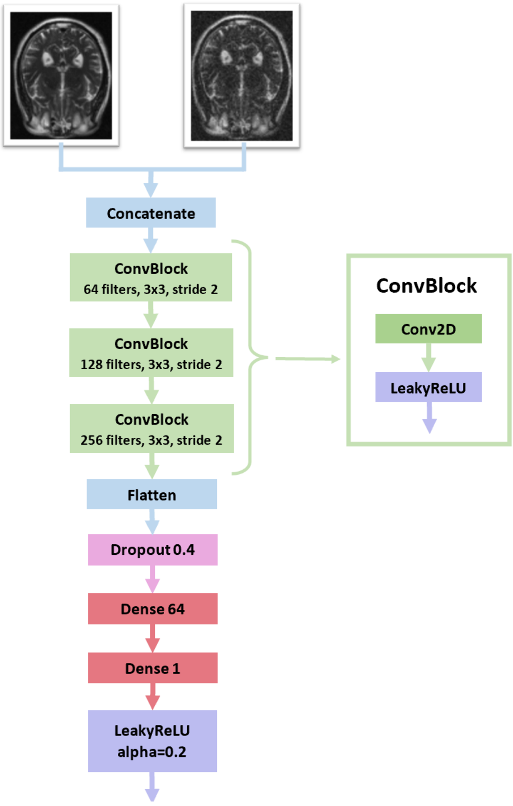

Figure 3: Structure of the discriminator. During training, we give it as input first the target image and then the denoised one. In both cases, the input is concatenated with the noisy reconstruction, so that the initial image given to the generator is taken into account during the discrimination.

In order to evaluate our model, we have used as baseline the results obtained by [19] using a simple DnCNN. Since their main focus was on knee MRIs and they did not report any numerical result concerning the performance of the network on brain MRIs, we first reproduced the DnCNN architecture and trained it on brain MRIs to get our baseline. The DnCNN consists of a Conv2D layer with ReLU activation function and 64 filters of size 3×3, other 15 Conv2D layers followed by batch normalization, and a final Conv2D layer with linear activation function, with a residual connection to subtract the extracted noise from the input. All convolutional layers have as kernel initializer *he_uniform*, which draws samples from a uniform distribution within $[-limit, limit]$, where $limit = \sqrt{6/fan\_in}$, with *fan_in* being the number of input units in the weight tensor. Our very first experiment consisted in using the DnCNN exactly as it was as generator of our GAN, to see if the adversarial training alone could benefit the denoising performance of the network. In particular, this was done by training the

Noise level	Architecture	PSNR	ΔPSNR	SSIM	ΔSSIM
High Noise	DnCNN (baseline)	30.4 ± 5.1	12.9 ± 0.4	0.77 ± 0.14	0.53 ± 0.06
	GAN - DnCNN	31.0 ± 5.3	13.5 ± 0.2	0.82 ± 0.11	0.58 ± 0.09
	CGAN - generator Dn-ResNet (Adam)	31.6 ± 5.4	14.1 ± 0.1	0.84 ± 0.09	0.60 ± 0.11
	CGAN - generator Dn-ResNet (RMSprop)	31.5 ± 6.1	14.0 ± 0.6	0.84 ± 0.11	0.60 ± 0.09
Low noise	DnCNN (baseline)	34.3 ± 6.0	10.0 ± 0.4	0.84 ± 0.10	0.39 ± 0.24
	GAN - DnCNN	33.6 ± 6.1	9.3 ± 0.5	0.83 ± 0.11	0.38 ± 0.15
	CGAN - generator Dn-ResNet (Adam)	35.0 ± 5.8	10.7 ± 0.2	0.85 ± 0.12	0.40 ± 0.22
	CGAN - generator Dn-ResNet (RMSprop)	35.0 ± 5.7	10.7 ± 0.3	0.89 ± 0.13	0.44 ± 0.23
Blind noise	DnCNN (baseline)	31.1 ± 7.1	10.6 ± 2.1	0.84 ± 0.11	0.52 ± 0.28
	GAN - DnCNN	29.9 ± 7.7	9.4 ± 0.8	0.67 ± 0.22	0.35 ± 0.17
	CGAN - generator Dn-ResNet (Adam)	32.3 ± 6.4	11.8 ± 2.8	0.84 ± 0.09	0.52 ± 0.3
	CGAN - generator Dn-ResNet (RMSprop)	32.9 ± 8.0	12.4 ± 1.2	0.86 ± 0.07	0.57 ± 0.32

Table 2

Results obtained from training the different architectures on the three datasets, using the training setups described in Section 8. The grey rows refer to the model that gave the best results on all the three datasets. The columns containing the Δ quantities refer to the improvement in the metrics values with respect to their values for the noisy datasets.

GAN using the Adam optimizer for both the generator and the discriminator, first for 300 epochs with a learning rate of $3 \cdot 10^{-3}$, and then until the validation loss decreased with learning rate $3 \cdot 10^{-4}$. The loss coefficient β was initialized at 0 and it was linearly increased from 0 to $6 \cdot 10^{3}$ between epochs 50 and 200. These were the same hyperparameters' values used in [19]. However, because of the difficulties related to the training of GANs, this did not bring to an improvement in the results. By looking at table 2, it is possible to notice that the results on the low-noise and blind-noise datasets are actually slightly lower than those obtained using the DnCNN. However, it must be highlighted that for the high-noise dataset there is already a big improvement, especially concerning the SSIM value, which goes from 0.77 to 0.82. This shows that the GAN is able to capture noise when it is more evident, but that it can do it better than the DnCNN.

Our next experiment consisted in adding skip connections to the DnCNN architecture, since most of the denoising GANs proposed in the literature have as generator U-Net, that as we specified in section 3 is able to capture localized features, such as fine structures, thanks to the presence of skip connections. While looking for the best training setup, we also introduced a condition in the discriminator learning procedure, by concatenating the noisy data to each input of the discriminator, thus obtaining a Conditional GAN. In this way, we have reached the architecture presented in the previous section. In these experiments, the CGAN was trained for 800 epochs, using a batch size of 64, and Adam optimizer with learning rate of $5 \cdot 10^{-3}$. However, while the generator is trained at each epoch, the discriminator is trained every six epochs, to give the generator the time to learn how to fool it. Additionally, since while training the CGAN the generator loss on the validation set arrived at a point in which it did not manage to further decrease, we have implemented a modified early stopping. More specifically, the patience has been set to 5 and if the validation loss of the generator does not decrease for that number of epochs, we first half the learning rate, and after 8 times that this has happened, we stop the training. Furthermore, we also store the minimum reached validation loss and if this loss remains above that value for 20 epochs, we use the same mechanism we have just described. Finally, the loss weights, β and γ, have been set respectively to the constant values of $6 \cdot 10^{3}$ and 1. Clearly, all the hyperparameters mentioned until now have been selected by performing hyperparameters tuning in each of the experiments. This training setup allowed us to obtain better results with respect to the baseline, that are reported in table 2. In this case, the improvements do not just concern the high-noise dataset, but all the three of them. Again, the highest improvement is obtained on the SSIM for the high-noise dataset, which is raised by 0.07 with respect to the baseline. The greatest gap in the PSNR, instead, is reported both for the high-noise and the blind datasets, for which we registered an increase of 1.2. Moreover, to demonstrate that the improvement was due to the combination of the new generator and the adversarial training, and not only to the introduction of skip-connections, we have also trained the generator as a network by itself, so showing that our CGAN was able to achieve a better performance. We do not report the obtained results, since they are comparable with the DnCNN used as baseline.

The last experiment that has been performed consisted in switching optimizer, by using RMSprop in the Conditional GAN. This change allowed to gain a slight further improvement in the results we had previously obtained with the Adam optimizer. In particular, the most signifi-

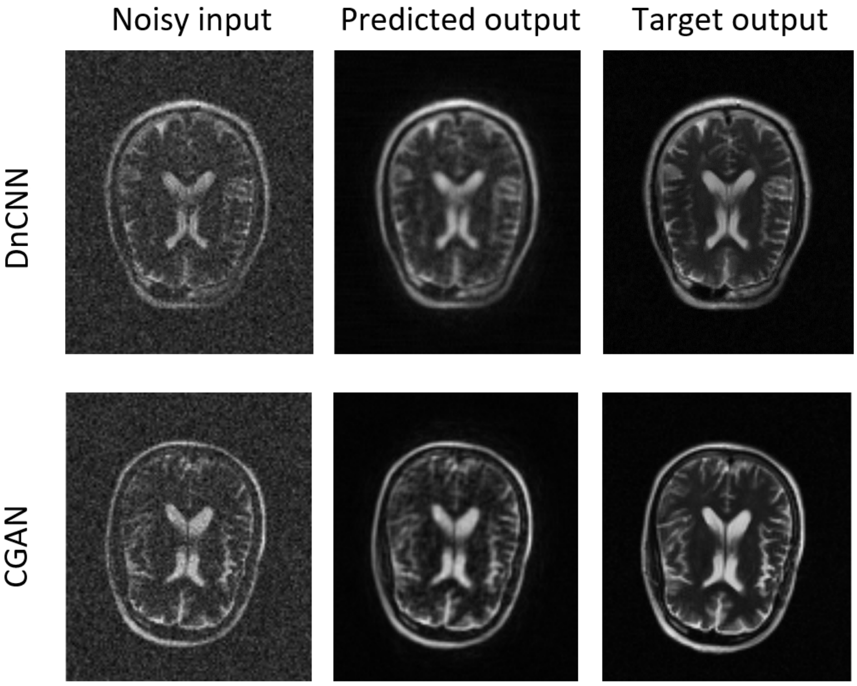

Figure 4: Results on similar images from the high-noise test dataset obtained using respectively the DnCNN and the CGAN trained using RMSprop.

cant improvements result in the low-noise dataset, where the SSIM is raised by 0.04, and in the blind-noise dataset, where PSNR is raised by 0.6 and SSIM by 0.02, all with respect to the architecture with Adam optimizer. Regarding the high-noise dataset, comparable results have been obtained with both optimizers. Since PSNR and SSIM are relevant but not necessarily indicative in the analysis of complex structures, also the visual quality of the denoised images must be taken into account. In Figures 4, 5, and 6, we have reported the noisy, target and denoised images obtained with the baseline DnCNN and our CGAN with RMSprop optimizer, in order to make some comparisons. By looking at the images, it can be noticed that denoising through the DnCNN brings to more blurred images. This represents a major problem for making diagnoses both by doctors and by automatic algorithms, because it makes it much harder to identify small structures of the brain, such as the hippocampus, or anomalies. Moreover, by worsening the definition of edges, it makes the images unsuitable to be used as input for automatic segmentation algorithms. In conclusion, the adversarial training brought improvements both in the quantitative metrics and in the quality of the denoised images.

9. Conclusions

In this work, we present a novel approach to brain MRI denoising. In particular, we highlight the importance of performing image denoising in the k-space, before reconstructing the actual magnitude image, and the lack of much prior research in this field. Moreover, since GANs have been demonstrated to be very promising in denoising tasks and they had not been previously used on k-space data, we propose a CGAN having a Dn-ResNet as generator and a CNN as discriminator. It is important to underline that the addition of skip connections to the generator allowed to have an increase in the values of the metrics, since they are able to capture localized features, such as fine structures. Furthermore, the concatenation of the noisy input of the generator to the input of the discriminator allowed to boost the performance, and to have an improvement on all the three datasets. Finally our CGAN also influences the visual quality of images, generating slightly more defined brain denoised images, with less blurred edges. Therefore, exploiting GANs for MRI denoising in the k-space seems to be a promising technique, which would be interesting to further explore, using different architectures for the generator and discriminator models.

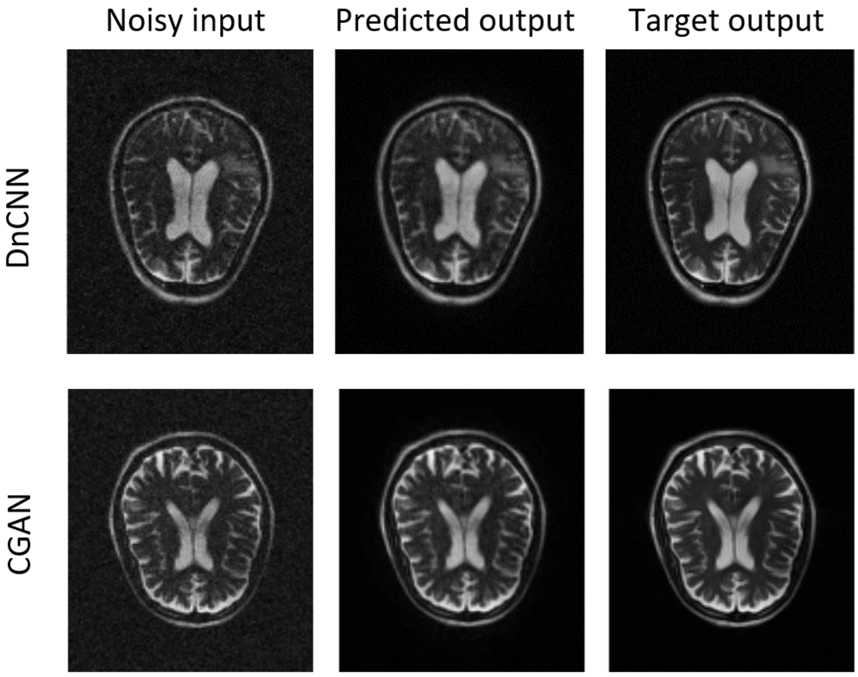

Figure 5: Results on similar images from the low-noise test dataset obtained using respectively the DnCNN and the CGAN trained using RMSprop.

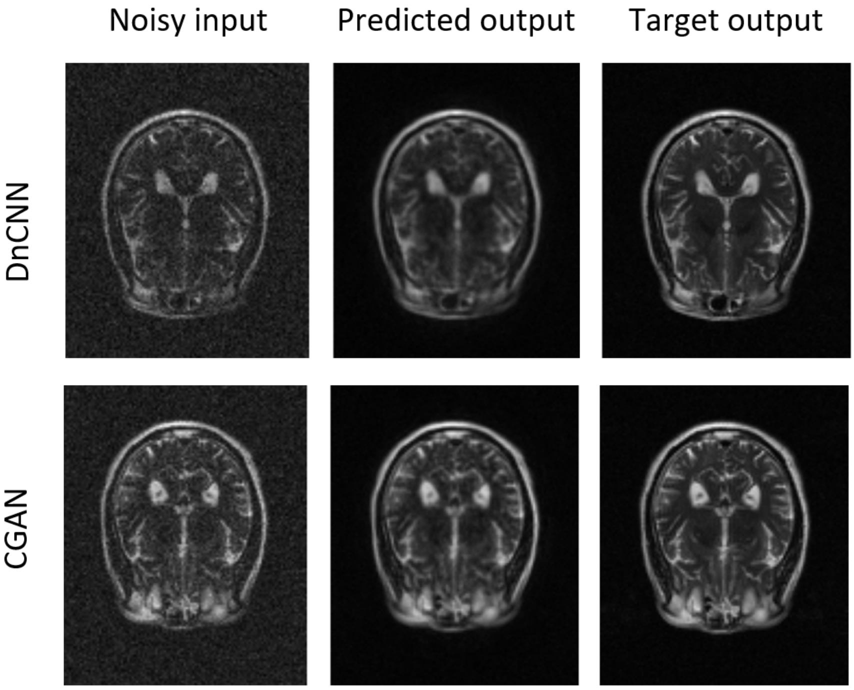

Figure 6: Results on similar images from the blind-noise test dataset obtained using respectively the DnCNN and the CGAN trained using RMSprop.

Acknowledgments

We would like to thank professor Christian Napoli, professor Stefano Giagu and Ph.D. Giorgio De Magistris for the advices and support received during this work.

References

[1] T. Redpath, Commentary signal-to-noise ratio in mri, The British Journal of Radiology 71 (1998) 704–707. doi:10.1259/bjr.71.847.9771379, pMID: 9771379.

[2] H. Zhu, Y. Li, J. Ibrahim, X. Shi, H. An, Y. Chen, W. Gao, W. Lin, D. Rowe, B. Peterson, Regression models for identifying noise sources in magnetic resonance images, Journal of the American Statistical Association 104 (2009) 623–637. doi:10.1198/jasa.2009.0029.

[3] E. McVeigh, R. Henkelman, M. Bronskill, Noise and filtration in magnetic resonance imaging, Medical physics 12 (1985) 586–591.

[4] P. Perona, J. Malik, Scale-space and edge detection using anisotropic diffusion, IEEE Transactions on pattern analysis and machine intelligence 12 (1990) 629–639.

[5] C. Tomasi, R. Manduchi, Bilateral filtering for gray and color images, in: Sixth international conference on computer vision (IEEE Cat. No. 98CH36271), IEEE, 1998, pp. 839–846.

[6] A. Buades, B. Coll, J.-M. Morel, A review of image denoising algorithms, with a new one, Multiscale modeling & simulation 4 (2005) 490–530.

[7] J. Manjon, M. Robles, N. Thacker, Multispectral mri de-noising using non-local means, in: Proc. MIUA, volume 7, Citeseer, 2007, pp. 41–45.

[8] P. Coupé, P. Yger, C. Barillot, Fast non local means denoising for 3d mr images, in: International conference on medical image computing and computer-assisted intervention, Springer, 2006, pp. 33–40.

[9] P. Coupé, P. Yger, S. Prima, P. Hellier, C. Kervrann, C. Barillot, An optimized blockwise nonlocal means denoising filter for 3-d magnetic resonance images, IEEE transactions on medical imaging 27 (2008) 425–441.

[10] V. Ashamol, G. Sreelekha, S. P S, Diffusion-based image denoising combining curvelet and wavelet, 2008, pp. 169 – 172. doi:10.1109/IWSSIP.2008.4604394.

[11] J. Sijbers, A. den Dekker, P. Scheunders, D. Van Dyck, Maximum-likelihood estimation of rician distribution parameters, IEEE Transactions on Medical Imaging 17 (1998) 357–361. doi:10.1109/42.712125.

[12] S. Aja-Fernandez, C. Alberola-Lopez, C.-F. Westin, Noise and signal estimation in magnitude mri and rician distributed images: A lmmse approach, IEEE Transactions on Image Processing 17 (2008) 1383–1398. doi:10.1109/TIP.2008.925382.

[13] D. Tisdall, M. Atkins, Mri denoising via phase error estimation, Proceedings of SPIE - The International Society for Optical Engineering 5747 (2005). doi:10.1117/12.595677.

[14] L. Gondara, Medical image denoising using convolutional denoising autoencoders, CoRR abs/1608.04667 (2016). URL: http://arxiv.org/abs/1608.04667. arXiv:1608.04667.

[15] M. Nishio, C. Nagashima, S. Hirabayashi, A. Ohnishi, K. Sasaki, T. Sagawa, M. Hamada, T. Yamashita, Convolutional auto-encoder for image denoising of ultra-low-dose ct, Heliyon 3 (2017).

[16] C. Bermúdez, A. J. Plassard, L. T. Davis, A. T. Newton, S. M. Resnick, B. A. Landman, Learning implicit brain mri manifolds with deep learning, in: Medical Imaging, 2018.

[17] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial networks, 2014. URL: https://arxiv.org/abs/1406.2661. doi:10.48550/ARXIV.1406.2661.

[18] J. Zbontar, F. Knoll, A. Sriram, T. Murrell, Z. Huang, M. J. Muckley, A. Defazio, R. Stern, P. Johnson, M. Bruno, et al., fastmri: An open dataset and benchmarks for accelerated mri, arXiv preprint arXiv:1811.08839 (2018).

[19] A. Ciardiello, Improvements and deep learning applications in 19F-NMR, Ph.D. thesis, Sapienza University of Rome, 2021.

[20] H. Ren, M. El-Khamy, J. Lee, Dn-resnet: Efficient deep residual network for image denoising, in: Asian Conference on Computer Vision, Springer, 2018, pp. 215–230.

[21] L. D. Tran, S. M. Nguyen, M. Arai, Gan-based noise model for denoising real images, in: Proceedings of the Asian Conference on Computer Vision, 2020.

[22] M. Mirza, S. Osindero, Conditional generative adversarial nets, arXiv preprint arXiv:1411.1784 (2014).