

# Deep Unrolled Reconstruction Network for Highly Undersampled MRI images using Multiple Sets of Coil Sensitivity Maps

Antoine Cadiou<sup>1</sup>

Department of Image Reconstruction, Siemens Healthineers, Princeton, New-Jersey, United-States

<sup>1</sup>École Normale Supérieure Paris-Saclay

antoine.cadiou@ens-paris-saclay.fr

The concepts and information presented in this paper/presentation are based on research results that are not commercially available.

Future commercial availability cannot be guaranteed.

## Abstract

*Magnetic Resonance Imaging (MRI) is a method of acquisition that is used in the medical field. Acquiring an MRI image is a very time consuming process. Indeed, the principle is to sample the  $k$ -space (Fourier space) by acquiring magnetized signal through the body of the patient, then, to reconstruct this signal into the image-space (with a Fourier transform) so that the doctor can use it to make a diagnostic.*

*One can parallelize the acquisition process by using multiple coils which are physical plates that allow to capture a different signal depending on their position on the body of the patient (they are more sensitive to nearby areas). The quality of the resulting image depends very much on the  $k$ -space sampled beforehand. A complete acquisition of a volume in MRI can therefore take several tens of minutes, which brings many constraints for the patient (having to remain immobile for a very long time, which is not always possible for all patients) but also poses problems of image registration related to possible movements.*

*For all these reasons we have to accelerate the acquisition process (by a factor of 2, 3 or 4 for example), and this induces an under-sampled  $k$ -space with missing lines.*

*By observing the resulting image, one will have a very aliased image and its quality will be strongly degraded. The challenge is to develop a Deep Learning pipeline to reconstruct this under-sampled signal (to find the missing values in the  $k$ -space) so that the final image is of good quality and the acquisition time is considerably reduced. For this we will use a SENSE-like method based on an iterative model (ISTA) to reconstruct the under-sampled image. This method will require 2 sets of Coil Sensitivity Maps (CSM) to be given as priors, but another method will be explored to compute them in an end-to-end approach.*

## 1. Introduction

In magnetic resonance imaging, the data collected is not directly in image-space format. Instead, MRI data are collected in  $k$ -space. This Fourier space contains spatial frequency information as well as the phase of the collected signal.  $k$ -space data are produced using spatially varying magnetic field gradients superimposed on the main magnetic field of the MRI scanner. A coil positioned on the body of the patient captures the magnetized signal and represents this signal in the Fourier space ( $k$ -space, also called the signal captured by the coil). The inverse Fourier transform is then used to transform this  $k$ -space data into image space once the acquisition process is complete; hence one can obtain a coil-image. Usually, multiple coils are used at the same time to capture different parts of the body simultaneously, this helps to speed-up the process of acquisition (see subsection 2.3).

The acquisition process of an MRI is very slow, but once we have completely sampled the  $k$ -spaces (see subsection 2.1), then we proceed to the reconstruction which consists in doing the Inverse Fourier Transform (IFT or  $\mathcal{F}^{-1}$ ) to each of them and then combining the coil-images with a root-sum-square (RSS). This process allows us to obtain a fully-sampled image that does not present any subsampling artifacts (Figure 1).

When comes the question to speed up this very slow process, one can under-sample the  $k$ -space, i.e. omit some lines. Thus, by taking only one line out of 2, it accelerates by two the acquisition time. The counterpart of such strategy is that after proceeding to the naïve IFT+RSS reconstruction (by putting zeros at the unknown values in the  $k$ -space), the resulting image will be of lower quality, with a certain number of artifacts, especially aliasing (Figure 1).

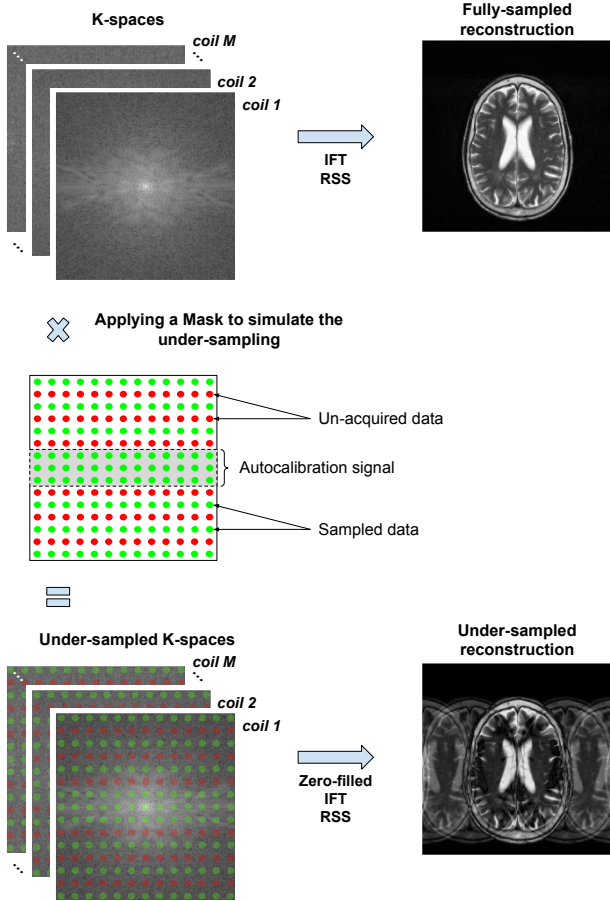


Figure 1. Reconstruction of fully-sampled coils ( $k$ -spaces). Simulation of an under-sampling by multiplying each coil by a mask. Reconstruction of under-sampled coils.

Reconstructing this is an inverse problem: one wants to go from these under-sampled coil-images to the fully-sampled coil-combined image.

Inverse problems are typically ill-posed, as opposed to the well-posed problems usually met in mathematical modeling. Of the three conditions for a well-posed problem suggested by Jacques Hadamard (existence, uniqueness, and stability of the solution) the condition of uniqueness is most often violated in MRI image reconstruction and denoising problematics.

So we wonder how to improve this result, and to do so, one must find a way to recover the missing values of the  $k$ -spaces that were set to zero. Many techniques can be considered for this problem, which has always been a very active subject. Historically, there are efficient methods such as GRAPPA [5], SENSE [10], ESPIRiT [13], DL-ESPIRiT [12] and wavelet transforms which adopt the

idea of iterative models using CSMs as priors to improve the reconstruction.

First, we will define the objects we manipulate (coils,  $k$ -space, CSMs, ...), then we will present different existing techniques.

In a second time, we will present our solution, our experimentations (data, pre/post processing, ...) and finally we will discuss our quantitative and qualitative results.

## 2. Background

### 2.1. K-Space and the MRI image

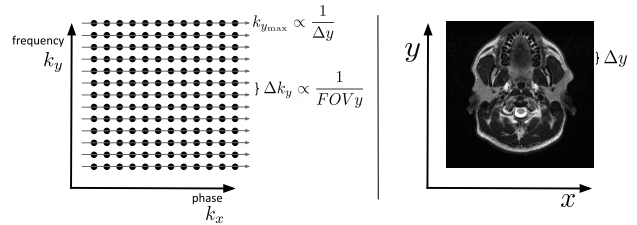


Figure 2. Left: Representation of a  $k$ -space Right: Image-Space obtained after having applied an IFT on the left image

As it is well explained in the paper [3], the spacing between  $k$ -space points in each direction is inversely proportional to the field-of-view (FOV) in that direction (Figure 2). For example, decreasing the spacing in the  $k_y$ -direction will result in an increase in the FOV in the  $y$ -direction of the image. The highest frequency collected in  $k$ -space ( $k_{x_{max}}$  or  $k_{y_{max}}$ ) is inversely proportional to the image resolution, or spacing between points in the image domain ( $\Delta x$  or  $\Delta y$ , respectively). Therefore, increasing  $k_{y_{max}}$  will result in decreasing  $\Delta y$  and increasing resolution in the  $y$ -direction of the image.

The FOV and resolution of an image can be manipulated by changing the number of points acquired in the corresponding  $k$ -space and the spacing between these points.

In general,  $k$ -space data are collected line-by-line in order to fill a grid of points (for a cartesian trajectory),  $k_x$  is defined as the read (or frequency-encoding) direction, and  $k_y$  is the phase-encoding direction (as illustrated in Figure 2). In a volumes (3D), there is an additional axis  $k_z$ , which corresponds to partition encoding (it's basically a second phase-encoding direction). The total acquisition time ( $T_A$ ) needed to acquire a 2D slice can be written as:

$$T_A = T_R \times N_{PE} \quad (1)$$

where  $T_R$  is the repetition time, or the time needed to acquire one line of  $k$ -space along the  $k_x$  direction, and

$N_{PE}$  is the number of phase-encoding lines in the  $k_y$  direction.  $T_R$  is helpful to determine the contrast in the image and  $N_{PE}$  to determine the resolution of the image in the phase-encoding direction, as described above.

There are 2 possible ways to reduce the acquisition time: or the  $k$ -space data is collected more quickly (reducing the  $T_R$  in Equation 1), or less data is collected in the  $k$ -space (reducing the  $N_{PE}$ ).

Focusing on the second method, a possible way to achieve this is to simply decrease  $k_{y_{max}}$  while keeping the same  $\Delta k_y$  spacing, because the resolution is proportional to  $1/k_{y_{max}}$  this option will lead to a reduction in image resolution.

If image resolution must be maintained for clinical assessment of the images (which is exactly our situation), another option is to remove some phase-encoding lines because removing lines implies increasing the  $\Delta k_y$  spacing, the result is a reduction in the FOV, which can lead to spatial aliasing.

This last idea is the one commonly used in this work: we remove some phase-encoding lines (in fact, they are not acquired). Hence, our work aims to reduce spatial aliasing artefacts due to this method.

Last detail to keep in mind: The center of the  $k$ -space corresponds to the low frequencies, they contain in particular the information of the contrast. The high frequencies contain the details/contours of the image. Thus, when we would like to estimate the CSMs later (c.f. subsection 4.5), it will be wise to focus on the central information of the  $k$ -space, which we will call the calibration region.

## 2.2. Parallel imaging methods

While it is possible to accelerate MRI data acquisition by collecting fewer phase-encoding lines in  $k$ -space, the resulting spatial aliasing must be removed before the images can be used for clinical purposes. Parallel imaging has been developed to address this issue.

$k$ -space data are under-sampled in the phase-encoding direction (and potentially also the partition-encoding direction in 3D imaging) to reduce the scan time. The acceleration factor,  $R$ , is defined as the ratio of the amount of  $k$ -space data required for a fully sampled image to the amount collected in an accelerated acquisition (if every other line in  $k$ -space is collected, the acquisition is accelerated by factor  $R = 2$ ). If the Nyquist criterion is not met (and the FOV becomes smaller than the object), the result is an aliased image.

## 2.3. Receiver Coil Arrays & Coil Sensitivity Maps (CSMs)

Data are acquired using an array of independent receiver channels instead of using a large homogeneous volume receive coil (Figure 3). Each receiver coil is more sensitive to the specific volume of tissue nearest to the coil, which means that the coils provide an additional source of spatial information for image reconstruction. The coil sensitivity describes how sensitive a given channel is to a specific point in space; this sensitivity is often dependent on the object in the receiver coil, and therefore can vary from patient to patient.

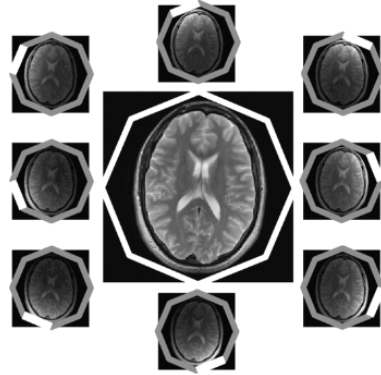


Figure 3. Representation of coil-images with their sensitivity depending on the location of the receiver coil. Figure from [3].

The individual receiver coils are arranged in an array such that the sensitivity profiles cover the desired FOV. When performing a scan with multiple coils, all of the images resulting from the multiple channels must be combined to form a single image. This can be performed using a root-sum-of-squares operation as explained previously, or by using a special algorithm which requires some knowledge of the individual coil sensitivities. Because parallel imaging relies on these coil sensitivity differences, acceleration can only practically take place in directions with coil sensitivity variations.

When a coil array is designed for a specific part of the body, great care is taken to construct a geometry that allows maximum acceleration in standard phase-encoding directions. Overlap between the different channels in the array, which reduces the amount of spatial information provided by the array, is also avoided.

Figure 4 shows that a coil-image is simply a result of multiplying the object with the coil sensitivity map associated to this coil.

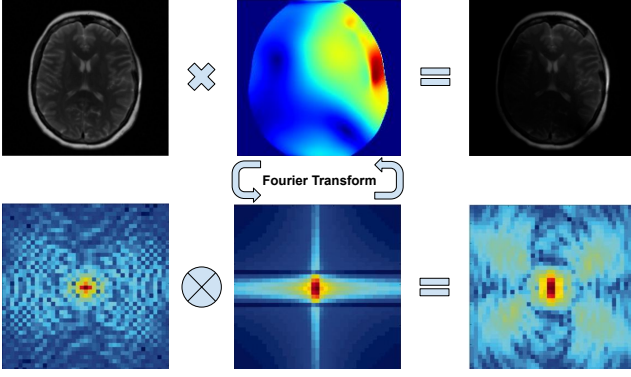


Figure 4. Image domain (top row) and Fourier space (bottom row). The object (left) is weighted, i.e. multiplied, by the sensitivity of the receiver coil (center) to yield the corresponding coil-image (right).

### 3. Related Works

Two different lines of reconstruction algorithms are in use today: Reconstruction algorithms based on explicit knowledge of the coil sensitivities such as SENSE and algorithms based on local kernels in  $k$ -space, which exploit the learned correlation between multiple channels in neighboring points in  $k$ -space, such as GRAPPA.

#### 3.1. Sensitivity Encoding (SENSE)

SENSE [10] is a reconstruction technique that operates in the image-space, which means that when we have our  $k$ -space under-sampled, we can do the inverse Fourier transform to get the under-sampled image-space that contain many artefacts. The idea is to work on these images directly to reconstruct them.

This technique does not require homogeneous and non-overlapping coil sensitivities, but the CSMs must be known.

SENSE poses the parallel imaging reconstruction as a linear inverse problem.

As we can see on Figure 4, to recover an image of the object, we need to have the coil-images and their associated CSM.

Let  $x$  be the coil-combined image in the image-space,  $S_i$  be the sensitivity map of the  $i^{th}$  coil ( $1 \leq i \leq N$  for  $N$  coils),

$\mathcal{F}$  be a Fourier operator,

and  $\mathcal{P}$  a mask that chooses only acquired locations in  $k$ -space (used to simulate the under-sampling). The received signal for the  $i^{th}$  coil can be written as

$$y_i = \mathcal{P}\mathcal{F}S_i x \quad (2)$$

When one knows the coil sensitivity maps  $S_i$ , or one can estimate them (c.f. subsection 4.5) precisely enough, then

the reconstruction is an inverse linear problem which can be solved with a least squares method directly or iteratively.

The only major drawback to the SENSE reconstruction is the need for an accurate coil sensitivity map. Errors in the CSM will cause artifacts in the form of residual aliasing in the reconstructed full FOV image. There are many factors that can cause the sensitivity profiles to be inaccurate such as the placement of the coils around the object or if the patient moves during the acquisition process.

#### 3.2. Generalized Autocalibrating Partially Parallel Acquisition (GRAPPA)

GRAPPA [5] is a very well known technique which allows to reconstruct each coil. The goal is to find the non-acquired values in each coil by interpolating with the acquired neighboring values. Once the missing lines have been restored, applying an inverse Fourier transform results in the final image. This is a technique that reconstructs in Fourier space ( $k$ -space), unlike SENSE which reconstructs in image-space.

The idea is to find a missing value of a  $k$ -space by looking in a 'frame' around it at the values that have been acquired.

As a reminder, we convolve the  $k$ -space with the result of the Fourier transform of the corresponding coil sensitivity map and this leads to a spreading of the data in the  $k$ -space (c.f. Figure 4). GRAPPA is based on this principle and suggests that the missing information of a  $k$ -space can be found by looking at the neighboring pixels and combining them together in an appropriate way (linear combination).

We can see on Figure 5 a diagram of the GRAPPA algorithm. For each missing point, we have to find a weight matrix that allows to find the information of the central point (the missing point) thanks to a linear combination of the neighboring values. We will call this weight matrix the reconstruction kernel  $g_{ri}$  ( $1 \leq i \leq N$ ,  $N$  be the number of coils, and  $r$  is an unsampled point)

Let  $R_r$  be the operation of choosing a block of  $k$ -space from all the coils around the position  $r$  of an unacquired value in  $k$ -space,

$P_r$  to represent local sampling patterns that choose only acquired samples from a block of  $k$ -space,

and  $y$  be our coil-images.

Finally, we can retrieve an unsampled point  $x$  in a coil-image  $i$  at location  $r$  as follow:

$$x_i(r) = (P_r R_r y)^T g_{ri} \quad (3)$$

The full  $k$ -space grid is then reconstructed by computing Equation 3 for all unsampled points in each coil-image.

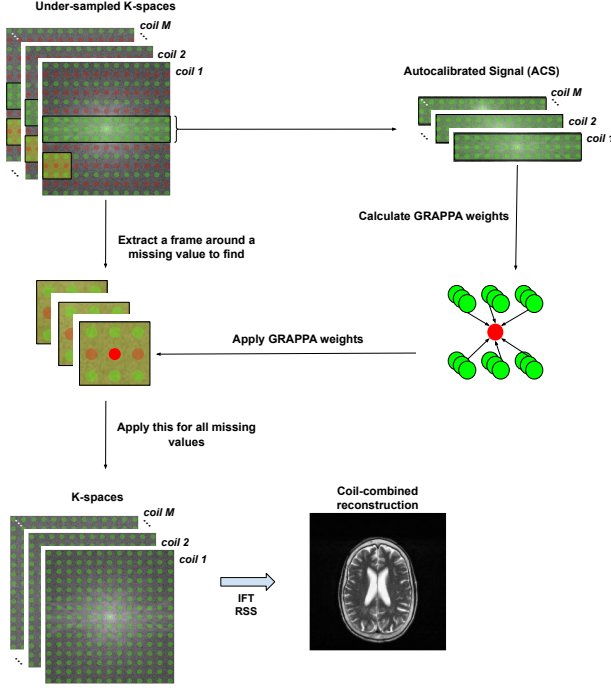


Figure 5. Overview of the GRAPPA algorithm.

These weights  $g_{ri}$  are determined using the central portion of the k-space (the first 24 rows generally) which is called the autocalibration signal (ACS). This region is fully-sampled, so we say that GRAPPA is an "autocalibrated" method.

To perform the calibration it is useful to construct a so called calibration matrix, denoted by  $A$ , from the ACS portion of the acquired data (c.f. Figure 6). It is constructed by sliding a window throughout the ACS data, taking each block  $(R_r y)^T$  inside the ACS region to be a row in the matrix. The columns of  $A$  are shifted versions of the ACS area, leading to a matrix structure known as Block-Hankel. This calibration matrix will be helpful to obtain ESPIRiT maps, as we will see in next section (subsection 3.3).

### 3.3. ESPIRiT: Sensitivity Maps as an Eigenvalue problem

A good option to analyze the calibration matrix  $A$  is to calculate its singular value decomposition (SVD):

$$A = U \Sigma V^H \quad (4)$$

The columns of the matrix  $V$  are a basis for the rows of  $A$ , and thus a basis for all the overlapping blocks in the calibration data.

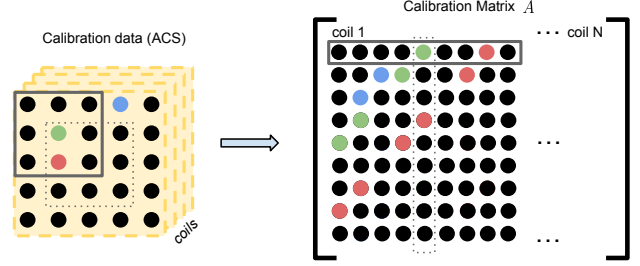


Figure 6. The calibration matrix  $A$  is constructed by sliding a window through the calibration data.

The paper [13] shows that thanks to the above decomposition, one can obtain the sensitivity profiles by following some few steps as depicted in Figure 7:

- First, select the number of eigenvalues to use (usually 1 or 2, depending on the number of sets of CSMs to extract). Then keep the eigenvectors corresponding to the selected eigenvalues.
- Then, for a given eigenvalue, apply a threshold on the signal energy so that it keeps the pixels of each eigenvector that are above this threshold.

Finally, the result is the ESPIRiT maps (or coil sensitivity maps) that can be used to train the solution.

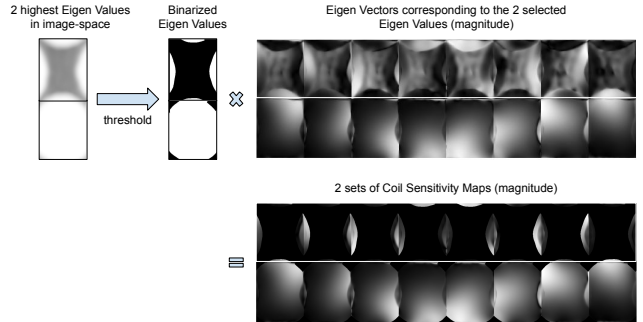


Figure 7. Steps to compute the ESPIRiT maps.

## 4. Method

### 4.1. Problem Definition

As we have just seen, SENSE defines the reconstruction problem as the Equation 2. This equation assumes that the 1st set of CSMs is known (it can be for example the ESPIRiT maps computed with the 1st eigenValue).  $S$  is then written in the following form:

$$S = [S^1, \dots, S^N]^T \quad (5)$$

With  $N$  the number of coil sensitivity maps in one set.

However, we can use several sets of CSMs, but this implies that the Equation 2 becomes:

$$y = \mathcal{PF} \sum_{i=1}^M S_i x_i \quad (6)$$

with  $M$  the number of sets of CSMs that we use (which directly refers to the number of eigenvalues that we keep when we want to extract the ESPIRiT maps for instance).

This Equation 6 can be rewritten in this form:

$$y = \mathcal{PF} E x \quad (7)$$

With:

$$E = \begin{bmatrix} S_1^1 & \dots & S_M^1 \\ \vdots & \ddots & \vdots \\ S_1^N & \dots & S_M^N \end{bmatrix}$$

$$x = [x_1, \dots, x_M]^T$$

Finally, we can solve the Equation 7 with a least squares method, but since it is an ill-posed problem we will use a regularizer which is a way to guide/constrain the reconstruction as we go along the iterations. The final problem that we want to optimize to reconstruct our image is the following:

$$\hat{x} = \arg \min_x \|y - \mathcal{PF} E x\|_2^2 + \lambda \mathcal{R}(x) \quad (8)$$

Let  $\mathcal{A}$  be the transformation such as  $\mathcal{A}(x) = \mathcal{PF} E x$  (Equation 7),

$\mathcal{A}^H = S^H \mathcal{F}^{-1} \mathcal{P}^H$  his conjugate,

$y$  the coil-images in the k-space and  $x$  the reconstructed image (in the image-space).

## 4.2. Iterative Reconstruction

So we want to solve the problem defined in Equation 8. We have a function of the following form that we wish to minimize:

$$\min f(x) + g(x)$$

With  $f(x) = \|y - \mathcal{A}(x)\|_2^2$  which is a smooth function,  $g(x) = \lambda \mathcal{R}(x)$  which is not smooth.

So we end up with a function which is not smooth, but which is convex (potentially nonstrongly convex).

First, let's focus on the function  $f$ . We can use the subgradient method to optimize this smooth function such that

$$x_{k+1} = x_k - \eta \nabla f(x_k) \quad (*)$$

$$= x_k - \eta (\mathcal{A}^H (\mathcal{A}(x_k) - y))$$

(\*) With  $\nabla f(x) = \mathcal{A}^H (\mathcal{A}(x) - y)$

This can be written in the proximal form as :

$$x_{k+1} = \arg \min_x f(x_k) + \nabla f(x_k)^T (x - x_k) + \frac{1}{2\eta} \|x - x_k\|_2^2$$

Now we want to minimize  $f + g$  with  $g$  non-smooth but being an 'easy prox operator'. So we use the proximal gradient formula:

$$x_{k+1} = \text{prox}_{\eta g}(x_k - \eta \nabla f(x_k))$$

$$\Rightarrow x_{k+1} = \arg \min_x \frac{\beta}{2} \|x - (x_k - \frac{1}{\beta} \nabla f(x_k))\|_2^2 + \lambda \mathcal{R}(x_k) \quad (*)$$

$$(*) \eta = \frac{1}{\beta} \text{ since } f \text{ is } \beta\text{-smooth} \quad (9)$$

The architecture selected to solve the above problem (Equation 9) is the Iterative Shrinkage/Thresholding Algorithm Model (ISTA) [19]

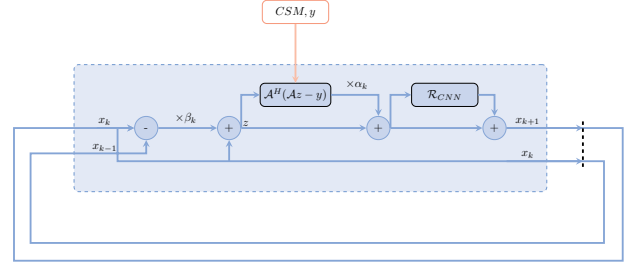


Figure 8. Iteration Block

An iteration block (Figure 8) consists of 3 phases (although some phases can be skipped at the beginning or at the end of the iterations, c.f. subsection 4.4).

- Extrapolation: use of a Nesterov momentum extrapolation to speed up the process; this allows to transform the ISTA into FISTA (Fast ISTA) [15].

$$z_k = (x_k - x_{k-1})\beta_k + x_k$$

- Gradient Update: in other words, we update the gradient according to the data-consistency.

It is about calculating the gradient to apply to  $x_k$  until convergence; it is calculated following Equation 9

If we use the extrapolation presented above, we update the gradient with this new formula:

$$x_{k+1} = \arg \min_x \frac{\beta}{2} \|x - (z_k - \frac{1}{\beta} \nabla f(z_k))\|_2^2 + \lambda \mathcal{R}(x)$$

Both  $\alpha_k = \frac{1}{\beta}$  and  $\beta_k$  (in the Nesterov momentum) are trainable parameters.

- Regularization: c.f subsection 4.3



### 4.3. Regularization Network

We can accompany the reconstruction algorithm with a denoising step, so that at each iteration of the process, we are able to denoise the image a little more, and thus, progressively improve the quality of the reconstructed image.

The development of deep learning has significantly improved the performance of image denoising techniques. The first methods [6] [16] used simple convolutional neural networks (CNN) but provided lower performance than conventional techniques such as Block-matching and 3D filtering (BM3D) [2]. Several years later, DnCNN [21] achieved a significant performance improvement using convolution, batch normalization, ReLU, and residual learning as a basic structure within a deep neural network. Then, RDN [23] was introduced with an architecture based on both residual learning and dense connections that fully exploit the hierarchical features from all the convolutional layers and achieves a considerable improvement in the performance of Gaussian noise image denoising. Most approaches assume known noise and thus require the training of a specific model for each noise level. FFDNet [22] was designed to handle unknown noise and multi-level noise with a single model using a noise map as an input to the network.

In our case, we use a denoising model called DIDN [18]. DIDN addresses several of the limitations of the aforementioned techniques by improving the U-Net [11] architecture for the image denoising task as shown in Figure 9. Its CNN architecture iteratively downscales and upscales the resolution of the feature maps with very large receptive fields and introduces several changes to the up-scaling and down-scaling layers. As the number of trainable parameters can be increased with its memory-efficient architecture, it can process a wide range of noise levels with a single model without requiring noise information and exhibits state-of-the-art performance.

It can be interesting to investigate this regularization network, trying to improve the current model which presents some defects, especially structural details as we can see on Figure 10. Thus, other works aiming at correcting these defaults and globally improving this pipeline have been carried out. We have compared various models of denoising: SCUNET [20] which is based on the attention mechanism as defined in the paper [14].

Finally, a last method of denoising is based on model diffusions. The denoising diffusion probabilistic models (DDPM) [17] are generative models like the Generative Adversarial Network (GAN) [4], the Variational Auto-Encoders (VAE) [7] or the Flow-based models [24]; that

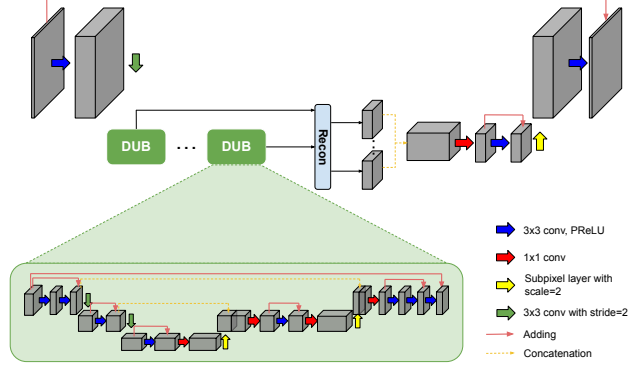


Figure 9. The architecture of the proposed DIDN from the official paper [18]

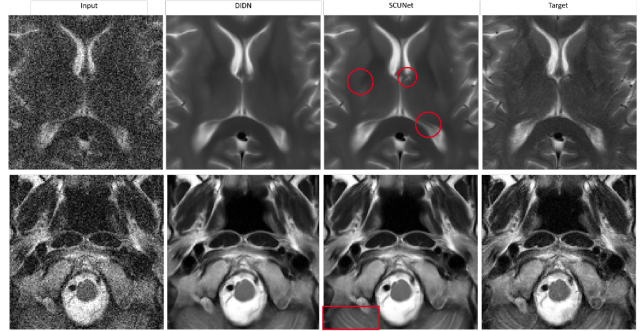


Figure 10. Comparison of DIDN and SCUNET for denoising

is to say that they try to learn a distribution, and then to reconstruct a new image by sampling this distribution. To do this, the DDPM contains 2 phases:

- Forward diffusion process: it consists, from an image  $x_0$  sampled from a real data distribution, to add iteratively a small gaussian noise (Equation 10). Thus, after  $T$  steps ( $T = 1000$  in the official paper), we end up with a completely noisy image in which we cannot distinguish any structure: we can compare this to the latent space  $z$  of the generative models, although in our case, the size of  $z$  is equal to the size of  $x_0$  (in the 'standard' generative models),  $|z| \ll |x_0|$

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1 - \beta_t}x_{t-1}, \beta_t) \quad (10)$$

A property of this Equation 10 is that we can calculate  $x_t$  directly from  $x_0$  using the following formula:

Let  $\alpha_t = 1 - \beta_t$  and  $\bar{\alpha}_t = \prod_{i=1}^T \alpha_i$

$$x_t = \sqrt{\alpha_t}x_{t-1} + \sqrt{1 - \alpha_t}z_{t-1} \quad (*)$$

$$= \sqrt{\alpha_t \alpha_{t-1}}x_{t-2} + \sqrt{1 - \alpha_t \alpha_{t-1}}\bar{z}_{t-2} \quad (**)$$

$= \dots$

$$= \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1 - \bar{\alpha}_t}z$$

$$q(x_t|x_0) \sim \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t}x_0, (1 - \bar{\alpha}_t)I) \quad (11)$$

(\*) where  $z_{t-1}, z_{t-2}, \dots \sim \mathcal{N}(0, I)$

(\*\*) where  $\bar{z}_{t-2}$  merges two Gaussians; the merged standard deviation is now  $\sqrt{1 - \alpha_t \alpha_{t-1}}$

- Reverse diffusion process: At the end of the forward process, we have  $z$  (or  $x_T$ ), and now we would like to sample an example from this  $z$ . This is the role of the reverse diffusion process. the goal is to iterate a noise removal function  $p_\theta(x_{t-1}|x_t)$  which refers in our case to a model such as U-NET [11]. It is therefore a very slow process because if  $T = 1000$ , then we need to make 1000 inferences with the U-NET to have a prediction.

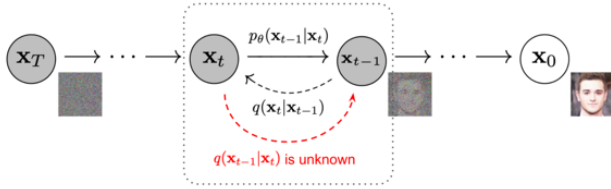


Figure 11. The Markov chain of forward (reverse) diffusion process of generating a sample by slowly adding (removing) noise. Figure from [17].

To reduce the time of the reverse process, and given that our input  $x$  already contains a very low noise (compared to the one in  $z$ ), we will use a Hijacking method to estimate the noise distribution in our input and start the reverse process at a very advanced stage (instead of doing 1000 iterations, we only do about ten (c.f. Figure 12).

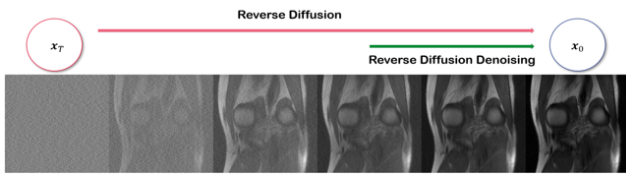


Figure 12. Hijacking of the diffusion process for the denoising task. Figure from [1].

On the Table 1 we can compare the performances of these different denoising models. They have all been trained with a loss  $L_1$ -complex (a loss  $L_1$  that takes into account the

complex property of images). Compute the  $L_1$  loss on the magnitude directly is not a good option because the noise to reduce follows a normal law on both the Real and Imaginary part, but for the magnitude the noise on the resulting image doesn't follow a normal distribution but a Rician distribution, and one wants to keep the normal distribution property (especially for diffusions models that are essentially based on this principle).

$$L_{1_{\text{magnitude}}} = \sum_{i=1}^n |y - \hat{y}|$$

$$L_{1_{\text{complex}}} = \sum_{i=1}^n \sqrt{\left(\Re(y) - \Re(\hat{y})\right)^2 + \left(\Im(y) - \Im(\hat{y})\right)^2} \quad (12)$$

For the next part of this report, we will set the regularizer to the DIDN model because it is still faster to train and to make the inference than the other mentioned models; however, the SCUNET could be a promising track to replace the DIDN.

#### 4.4. Pre and Post Iterations

In our pipeline, in addition to the blocks of iterations with the complete architecture (momentum + Gradient Update + regularizer), we distinguish the first iterations and the last ones that include some changes.

- Pre-Iterations: They employ trainable extrapolation; however, no regularization is applied, allowing the network to focus on the parallel imaging part of the reconstruction problem. Such design is motivated by the empirical finding that initial steps in the variational network focus on the signal recovery of missing data near the  $k$ -space center.

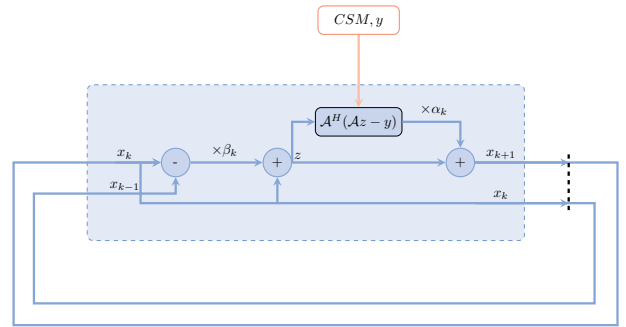


Figure 13. Pre-Iteration Block

- Post-Iterations are employing non-trainable extrapolation. They are used for further guarantees on the data consistency, which minimizes the risk of hallucination when adversarial training is applied



Model	# params (M)	Latency (s)	PSNR (magnitude)	SSIM (magnitude)
DIDN	9.97	0.02	33.99	0.7922
SCUNET	17.94	0.08	34.10	0.7737

Table 1. Comparison of performances for 2 denoising models (DDPM is not compared because it is a generative model so it may not be coherent to compute a PSNR or SSIM for the generated image). Models trained during 2000 epochs on 12138 images of size 128x128 with a batch size of 32, optimizer Adam and a Learning rate of  $1.10^{-4}$  (cosine scheduler)

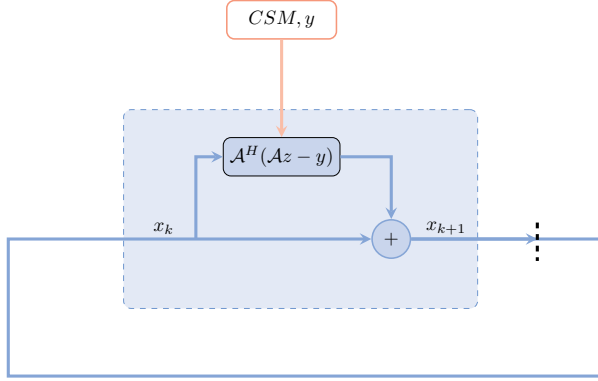


Figure 14. Post-Iteration Block

## 4.5. CSM Estimation

Initially, we have computed the CSMs offline for each of the slices in the dataset. The method originally used is the one described in the ESPIRiT subsection 3.3. We also used a new toolbox to compute these CSMs: BART. This method works in the same way as ESPIRiT, except that the threshold is computed automatically for each slice (it is not hard-coded for all slices).

So we obtain CSMs which are slightly different, and we will compare the quality of the reconstructions with the CSMs produced by these 2 methods.

### 4.5.1 End-to-end model

The problem with estimating CSMs with ESPIRiT as described above is that it takes a lot of time offline to generate the CSMs. It could be relevant to learn how to reconstruct these CSMs and to use these learned CSMs as input to the reconstruction model. For this, it would be necessary to train a complete pipeline (estimation of CSMs + reconstruction) in an end-to-end way. The back-propagation of the final loss would allow to update the weights of these 2 models and in-fine, to converge towards a solution. This would have a huge advantage on the time saving of the inference because we don't have to compute the CSMs offline anymore (this could take a lot of time: up to 2 minutes for a slice of size 1248x1998 for example).

Thus, one can think of a U-NET style solution but a new

architecture named DeepSet is introduced [9] that consists of multiple iterations of DIDNs Figure 15. This model has for role to generate the CSMs that one can then inject in the reconstruction model explained above. Ideally, the reconstructed output alone would allow to train the whole pipeline (csm estimation model + reconstruction model) with a L1Complex. However, the training of such a model is very time consuming; We have therefore divided the training of this pipeline in 3 steps:

- 1st step: Train the reconstruction model by itself, by using the CSMs computed offline as priors (CSMs from ESPIRiT maps or from BART Toolbox).
- 2nd step: Introduce the CSMs estimation model upstream, then freeze the pre-trained weights of the reconstruction model of step 1 in order to do the back-propagation only on the 1st module.
- 3rd step: Fine-tune the entire pipeline.

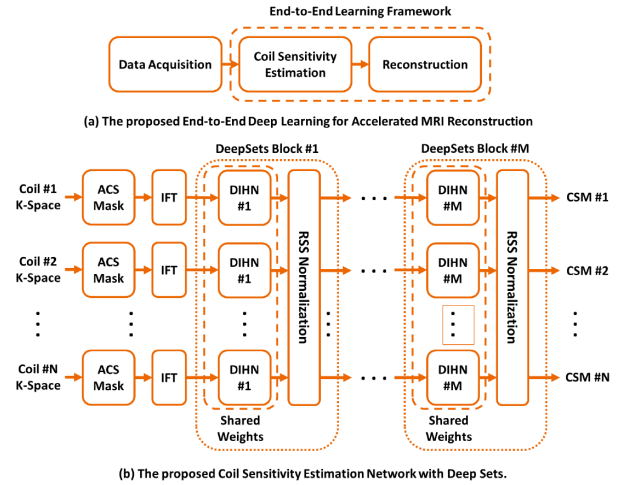


Figure 15. Overview of the end-to-end reconstruction pipeline (a) and architecture of the CSMs estimation network (b). Architecture from [9].

## 4.6. Dataset

The dataset we use is exclusively composed of brain MRIs and is divided into a training set and a validation set;

- Train-set: 7539 slices in 240 volumes.
- Validation-set: 1044 slices in 33 volumes,

For each of the slices, we have: its ground-truth fully-sampled, its 2 sets of CSMs (computed with the method presented in the subsection 4.5), its 8 coil-images in k-space, as well as its biasfield.

Since we would like to study the 2nd set of CSMs, we wanted to observe them; and we can see a strange phenomenon: most of the 2nd CSMs contain little or no information. One explanation for this phenomenon is that the brain data do not have a reduced FOV (the brain is completely represented in the image, it does not extend to the edges as other organs might), but generally the 2nd CSMs set allows to capture precisely this information that 'extends' beyond the field of view (in the case of a reduced FOV), as well as some motion/aliasing artifacts. In order to train the reconstruction model, it would be ideal to have a dataset with relevant 2nd CSMs, so we curated the dataset to keep only the examples that interest us.

### 4.6.1 Dataset Curation

For this, it was decided to look for each slice the magnitude of its 2nd CSMs and to binarize it according to a threshold of 0.001 (all pixels  $> 0.001$  are set to 1, otherwise 0). Then, we could compute the average content different from '0' ( $> 0.001$ ) in each slice, we can see the distribution obtained for the 2 sets of train/validation on Figure 16, and so we kept only the slices that had more than 5%. This procedure allows us to isolate 108 (24) slices in the train (validation) set. However, we still want to keep all the volumes in which these slices belong. Finally, after curation of the dataset, we end up with 59 and 15 volumes for the train and validation sets respectively.

### 4.6.2 Reduced FOV (rFOV) simulation

Starting from the fact that the 2nd CSMs are mostly empty because the MRIs do not have rFOV, and that following the data curation step we have removed many of the volumes that were initially present, it would be relevant to ask ourselves if we can eventually generate a dataset with rFOV. This would allow us to have relevant 2nd CSMs and thus our reconstruction model could learn efficiently from the 2nd set of CSMs (to generalize to other types of data than brain MRI). To do so, we thought about performing brain segmentation, then we can crop a part of the brain on each side and make sure that there is a folding of information on the

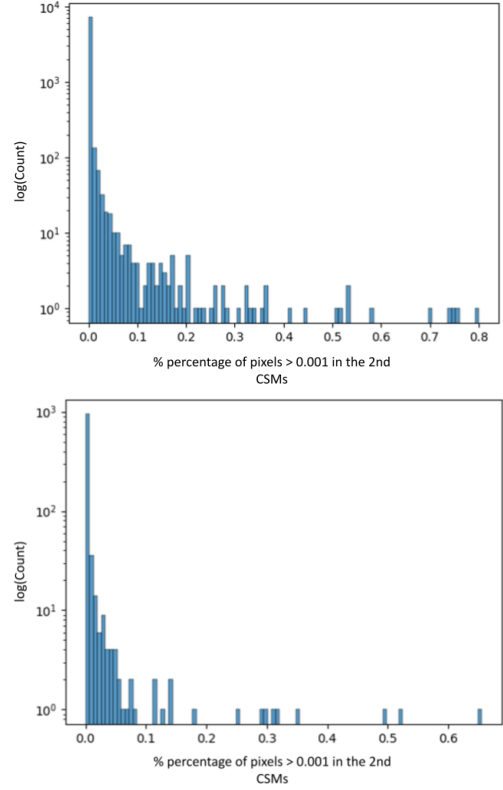


Figure 16. Distribution of percentage of content in the 2nd CSMs for each slice in train-set (left) and validation-set (right)

opposite side (see Figure 17). To segment the brains we apply morphological operations, such as erosions followed by dilations (an opening). This is to remove the small arefacts in the image while keeping the main structure of the brain. Once this is done, we get a mask of the brain pixels and we can then crop and fold the right and left sides according to a predefined threshold. We finally did not use this solution because we preferred to use the end-to-end model which estimated the CSMs rather than using this part because it would have been necessary to re-train the models and this requires time, and we had to define priorities.

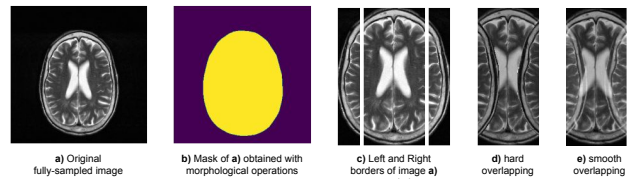


Figure 17. reduced FOV pipeline

### 4.6.3 Acceleration simulation

As a reminder, each slice is fully-sampled, i.e. we know the ground truth because the acquisition process has been carried out in its entirety (we have not accelerated the process). Since we want to be able to under-sample the data and then reconstruct them as similar as possible to their ground truth, we will have to find a way to artificially under-sample our slices in order to simulate an under-sampling. For this, we distinguish 2 cases: the case where we know the center of the k-space (usually the 24 lines in the center) in each coil-image, and the case where we only know them partially. Indeed, the center of the k-space plays an extremely important role because it is here that the low frequencies are located, and thus where are defined the biggest structures of the image (inversely, the finer structures are provided by the high frequencies). We will generate a mask to simulate the under-sampling. By multiplying this mask on each coil-image, we finally obtain our artificially under-sampled image which can be used as input in our model. A last notion to understand is the acceleration. We say that the acceleration factor  $n$  represents a mask in which we keep only 1 line out of  $n$ . The higher the acceleration factor is, the less we sampled our k-space and the worse the reconstructed zero-filled image will be (but the acquisition process gains in speed).

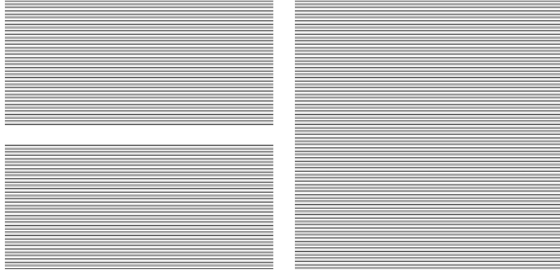


Figure 18. Masks to apply on fully-sampled  $k$ -space to simulate an under-sampling. Acceleration x2. With (without) center reference lines on left (right).

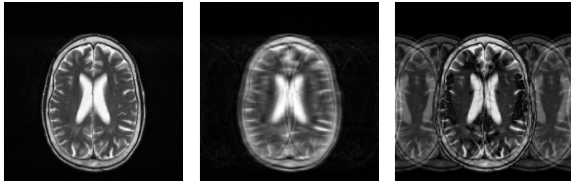


Figure 19. Left: Ground-truth fully-sampled  
Center: Zero-filled reconstruction after applying a mask with center reference lines  
Right: Zero-filled reconstruction after applying a mask without center reference lines

Our solution is trained only on under-sampled cases with center reference lines, with an Acceleration chosen between 1 and 4. The second case requires training the pipeline for this specific problem, with some adjustments notably for the estimation of the CSMs, and this is a possible direction for further works.

### 4.7. Metrics

We work mainly on complex data, which allows us to distinguish the metrics computed on the complex images from those computed on their magnitude.

#### 4.7.1 Peak Signal to Noise Ratio (PSNR)

It is a measure of distortion that quantifies the quality of a reconstructed image compared to the original image.

The PSNR represents the ratio between the maximum value that the desired signal (the target) can take and the power of the distortion (or error) between the target and the reconstructed image. This ratio is expressed in decibels. Peak signal-to-noise ratio is the most commonly used quality assessment technique for measuring the reconstruction quality of lossy image compression codecs. The signal is considered the original data and the noise is the error produced by compression or distortion. PSNR is a rough estimate of human perception of reconstruction quality in relation to compression codecs.

$$\begin{aligned} PSNR(y, \hat{y}) &= 20 \times \log_{10}(\max(y)) - 10 \times \log_{10}(MSE) \\ &= 10 \times \log_{10}\left(\frac{\max(y)^2}{MSE}\right) \end{aligned} \quad (13)$$

With  $MSE$  the Mean Square Error between the target  $y$  and the prediction  $\hat{y}$  :  $\mathbb{E}[(\hat{y} - y)^2]$

This metric allows us to directly assess the error between two images, however it has its limits, for example when we want to evaluate the quality of the structural information. We will therefore introduce a new metric: the SSIM.

#### 4.7.2 Structural Similarity Index Measure (SSIM)

The Human visual perception system is highly capable of identifying structural information from a scene and hence identifying the differences between the information extracted from a reference and a sample scene. Hence, a metric that replicates this behavior will perform better on tasks that involve differentiating between a sample and a reference image. The Structural Similarity Index (SSIM) metric extracts 3 key features from an image and the comparison between  $y$  and  $\hat{y}$  is performed by computing these 3 features : luminance, contrast and structure.

$$luminance(y, \hat{y}) = \frac{2\mu_y\mu_{\hat{y}} + c_1}{\mu_y^2 + \mu_{\hat{y}}^2 + c_1}$$

$$contrast(y, \hat{y}) = \frac{2\sigma_y\sigma_{\hat{y}} + c_2}{\sigma_y^2 + \sigma_{\hat{y}}^2 + c_2}$$

$$structure(y, \hat{y}) = \frac{2\sigma_{y\hat{y}} + c_3}{\sigma_y\sigma_{\hat{y}} + c_3}$$

With  $\mu_y$  the mean over all pixel values in the image  $y$ ,  
 $\sigma_y$  the standard deviation of all the pixel values,  
 $c_1$  and  $c_2$  two constants to ensure stability when the  
denominator becomes 0, which is equal to  $(k_1L)^2$  and  
 $(k_2L)^2$  respectively,  
 $L$  be the maximum value that  $y$  can take,  
 $k_1 = 0.01$  and  $k_2 = 0.03$ ,  
 $c_3 = c_2/2$

Finally, we can compute the the SSIM between 2 images:

$$\begin{aligned} SSIM(y, \hat{y}) &= luminance(y, \hat{y})^\alpha \times contrast(y, \hat{y})^\beta \\ &\quad \times structure(y, \hat{y})^\gamma \\ &= \frac{(2\mu_y\mu_{\hat{y}} + c_1)(2\sigma_{y\hat{y}} + c_2)}{(\mu_y^2 + \mu_{\hat{y}}^2 + c_1)(\sigma_y^2 + \sigma_{\hat{y}}^2 + c_2)} \quad (*) \end{aligned} \quad (14)$$

(\*) if  $\alpha, \beta$  and  $\gamma = 1$

It is important to understand how these 2 metrics work together, and why they are both important to assess the quality of a reconstruction. The PSNR may not be good enough, in fact, two images can have the same PSNR and look really different (for example, adding a blur to an image may not impact its PSNR but may impact the SSIM (c.f. Figure 20) Ideally, we would like to improve the two metrics for a given reconstruction.

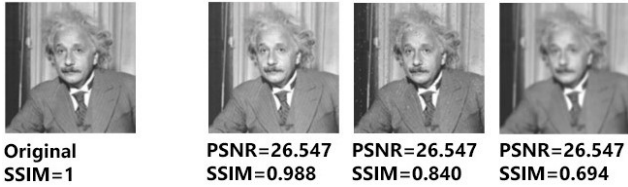


Figure 20. Comparison of PSNR and SSIM scores on a blurred image of Albert Einstein, Professor at Princeton University. Image from [8].

## 5. Results

We consider separately the experiments with CSMs calculated offline and those estimated by the model.

### 5.1. Reconstruction Models

We identify 3 models of reconstructions:

- Model trained with the 1st set of CSMs only (baseline).
- Model trained with the two sets of CSMs, and the loss is computed on the 1st prediction only (because since we use 2 sets of CSMs, we have 2 inputs, and hence 2 outputs).
- Model trained with the two sets of CSMs, and the loss is computed on the 2 predictions (fully-supervised).

Our 3 models have been trained for 1.5 million steps until their loss (L1 complex) converges. We use the PSNR and SSIM metrics to compare the models.

We also wanted to compare the scores for the two types of CSMs defined above: those obtained with the ESPIRiT method and those obtained with the BART Toolbox.

Table 2 allows us to hold 2 conclusions:

- For a given model, let's say number (1), we observe that the CSMs of the BART Toolbox obtain significantly higher results. The first conclusion is that BART Toolbox brings a real gain, globally, on the quality of the reconstructions. This is valid whatever the model.
- When we focus on ESPIRiT CSMs, we do not observe any improvement between the 3 models, whereas with BART Toolbox CSMs, there is a real interest in using a model that uses 2 sets of CSMs.

However, we cannot draw a conclusion as to which model is better than the other between (2) and (3). This is because the validation-set is large and the reconstructions are globally of the same quality (so when we average the PSNRs we do not have major differences). Thus, it is also good to remember that this work aims to remove small artifacts that appear for a few rare examples of the dataset, and that therefore making a general quantitative comparison is not the best solution: one should focus on those few images that show artifacts and make the qualitative and quantitative comparison on each of them to appreciate the different patterns. Thus, we can visualize some specific examples that contain artifacts.

In Figure 21, we can see the artifact that appears on the reconstruction (c), this artifact is due to the aliasing of the under-sampling that can be visualized on the image (b). This is particularly difficult to remove it since it is located relatively in the center of the image. Generally the aliasing artifacts at the center are the ones that persist the most because the coils sensitivity maps contain little information about the center of the image (they are more

Model	CSMs used	SSIM magnitude	PSNR magnitude	PSNR complex
(1) 1st set of CSMs	ESPIRiT	0.900	37.05	35.11
	BART Toolbox	0.952	37.57	35.72
(2) 2 sets of CSMs Loss on the 1st output	ESPIRiT	0.901	37.08	35.13
	BART Toolbox	0.954	37.74	35.89
(3) 2 sets of CSMs fully-supervised	ESPIRiT	0.900	37.04	35.10
	BART Toolbox	0.953	37.63	35.79

Table 2. Scores obtained on the validation set for the 3 different reconstruction models.

sensitive to edges). When we use ESPIRiT CSMs to reconstruct the image, with 1 (a) or 2 (b, c) sets of CSMs, the reconstructions contain few differences, the artifact persists. On the other hand, when we use the CSMs from BART Toolbox, in this case, the artifact is captured because the models with 2 sets of CSMs. Moreover, we can see a qualitative difference between the models whose loss is trained on the 1st reconstruction (g) or fully-supervised (h). These observations are also valid for Figure 2 which also contains an artifact in the center of the image. We note that the reconstruction of image (h) is of better quality than the one of (g) which is itself better than the 4 others. This allows us to propose the idea that the fully-supervised model trained on the 2 sets of CSMs of BART Toolbox is the one that obtains better results and therefore it is this one that we would like to use.

Finally, we can see in Figure 22 another example of reconstruction where the BART Toolbox CSMs obtain better results. But, the interesting thing in this example that differentiates it from the 2 previous ones, are the reconstructions (d) and (e) that show that even with ESPIRiT CSMs, the models using 2 sets of CSMs bring an improvement compared to the model (c) using only one set of CSMs.

## 5.2. End-to-end Models

The CSMs estimation model has an important role mainly in the acceleration of the process. Indeed, when we look at the computation time needed to obtain the CSMs, we realize that the offline method using the ESPIRiT maps takes a lot of time as we can see on the Table 3, this is caused by the computation of the SVD.

When we look at the actual CSMs predicted by the model (Figure 24, we notice that the results are very correct, they

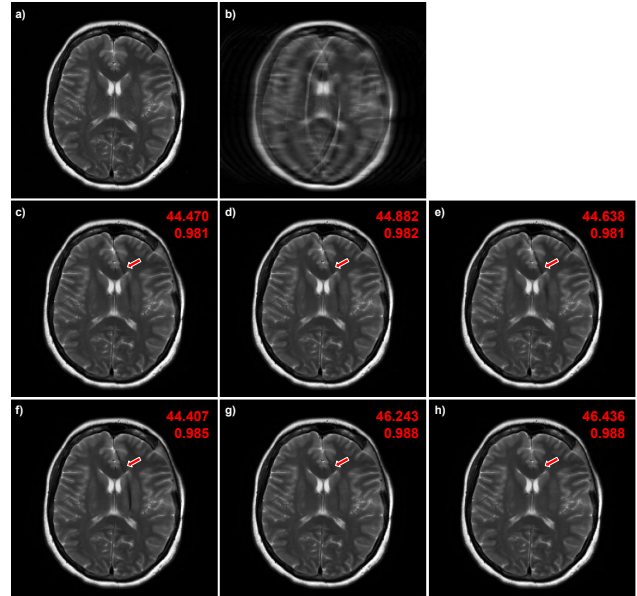


Figure 21. Example from a simulated under-sampled image of a brain with Aliasing artefacts. a) Ground Truth fully-sampled. b) Simulated under-sampled image -input of the pipeline- PAT 4 with 24 reference lines. c - e) Reconstructions using ESPIRiT CSMs with model 1 to 3 respectively. f - h) Reconstructions using BART Toolbox CSMs with model 1 to 3 respectively. Scores are PSNR first then SSIM.

make sense with those returned by the ESPIRiT method.

To obtain these results, we first trained this model in a supervised way, by calculating directly the error between the predicted CSMs and the ones of the ground truths (ESPIRiT method). Then, we will train the complete pipeline directly by computing the reconstruction loss at the output of the reconstruction model and by doing the back-propagation on



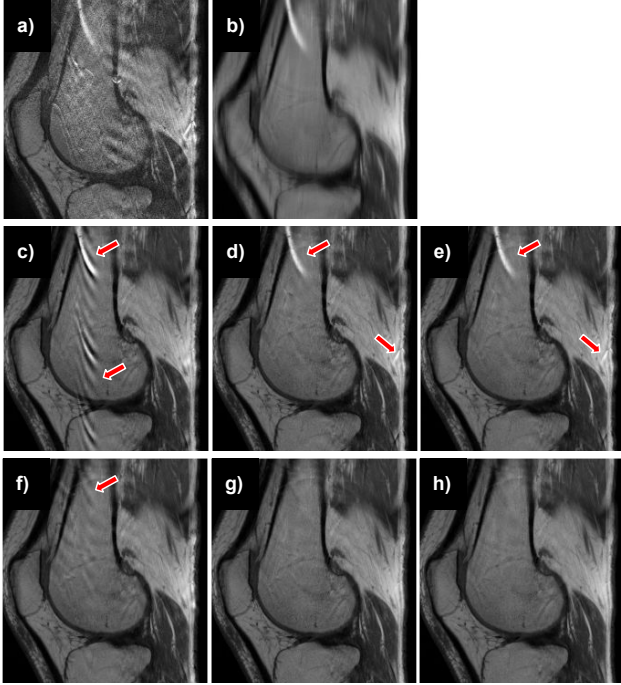


Figure 22. Example from an under-sampled case of a knee. a) SENSE Reconstruction. b) Under-sampled coil-combined image - input of the pipeline-. c - e) Reconstructions using ESPIRiT CSMs with model 1 to 3 respectively. f - h) Reconstructions using BART Toolbox CSMs with model 1 to 3 respectively.

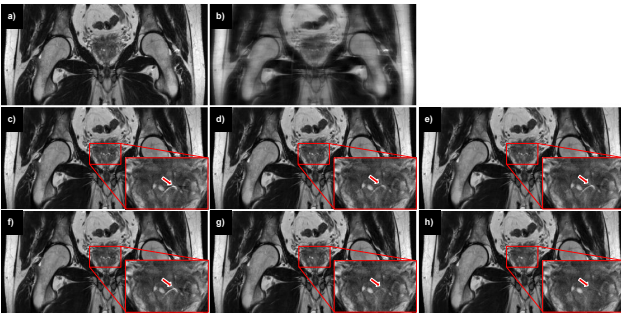


Figure 23. Example from a reducedFOV case of a pelvis. a) Ground Truth fully-sampled. b) Simulated under-sampled image - input of the pipeline-; PAT 3 with 24 reference lines. c - e) Reconstructions using ESPIRiT CSMs with model 1 to 3 respectively. f - h) Reconstructions using BART Toolbox CSMs with model 1 to 3 respectively.

the whole pipeline (CSMs estimation model and reconstruction model). This analysis is still in progress and I don't have any results at the time of the report submission (my internship is ending in 1 month, I will get the last results by then).

	ESPIRiT algorithm	Estimation model
slice	121.44	0.15
volume	$\sim 4860.40$	$\sim 5.91$

Table 3. Computational time (s) to obtain the CSMs. Size of the slice analyzed: 1248x1998, the volume used in the table is composed of 40 slices of the same size

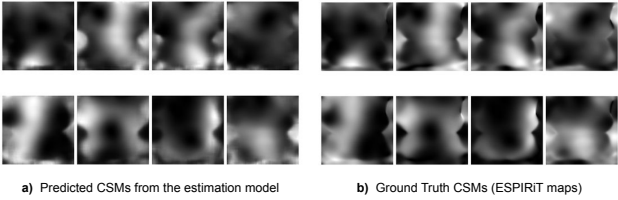


Figure 24. CSMs predicted by the estimation model compared to the ones obtained by the ESPIRiT method.

## 6. Conclusion

During this internship, I learned a lot about how an MRI works, the data acquisition process and the existing reconstruction methods in order to obtain a quality image allowing the doctor to make relevant diagnoses, and thus ultimately, to improve patient care.

I introduced the use of a second set of CSMs in the reconstruction pipeline, the motivation behind this was to improve some artifacts that persisted when using only one set of CSMs. Indeed, some artifacts due to the aliasing of the under-sampling (and thus of the acceleration of the acquisition process) persisted and the information needed to attenuate them was in the 2nd set of CSMs.

Moreover, as we noticed that the origin of the CSMs played an extremely important role in the reconstruction (those from BART Toolbox are more efficient than ESPIRiT maps), we were interested in an end-to-end model that would ideally generate relevant CSMs before using them for the MRI reconstruction model. This process saves a lot of time and can potentially further improve the results (work is still in progress).

We have been able to show quantitatively and qualitatively the efficiency of our solution while keeping in mind the initial problem which was to accelerate the acquisition process while improving the quality of the reconstructions.

## References

- [1] Hyungjin Chung, Eun Sun Lee, and Jong Chul Ye. MR Image Denoising and Super-Resolution Using Regularized Reverse Diffusion, Mar. 2022. arXiv:2203.12621 [cs, eess].
- [2] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Image Denoising by Sparse 3-D Transform-Domain Collaborative Filtering. *IEEE Transactions on Image Processing*, 16(8):2080–2095, Aug. 2007. Conference Name: IEEE Transactions on Image Processing.
- [3] Anagha Deshmane, Vikas Gulani, Mark Griswold, and Nicole Seiberlich. Parallel MR imaging. *Journal of magnetic resonance imaging : JMRI*, 36:55–72, July 2012.
- [4] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Networks, June 2014. arXiv:1406.2661 [cs, stat].
- [5] Mark A. Griswold, Peter M. Jakob, Robin M. Heidemann, Mathias Nittka, Vladimir Jellus, Jianmin Wang, Berthold Kiefer, and Axel Haase. Generalized autocalibrating partially parallel acquisitions (GRAPPA). *Magnetic Resonance in Medicine*, 47(6):1202–1210, June 2002.
- [6] Viren Jain and Hyunjune Seung. Natural Image Denoising with Convolutional Networks. pages 769–776, Jan. 2008.
- [7] Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes, May 2014. arXiv:1312.6114 [cs, stat].
- [8] Medium. A Quick Overview of Methods to Measure the Similarity Between Images, Mar. 2020.
- [9] Mahmoud Mostapha, Boris Mailhe, Simon Arberet, Dominik Nickel, and S. Nadar Mariappan. Coil sensitivity estimation with deep sets towards end-to-end accelerated mri reconstruction. *Abstract 1166, ISMRM and SMRT, Annual Meeting and Exhibition.*, May 2021.
- [10] K. P. Pruessmann, M. Weiger, M. B. Scheidegger, and P. Boesiger. SENSE: sensitivity encoding for fast MRI. *Magnetic Resonance in Medicine*, 42(5):952–962, Nov. 1999.
- [11] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation, May 2015. arXiv:1505.04597 [cs].
- [12] Christopher M. Sandino, Peng Lai, Shreyas S. Vasanawala, and Joseph Y. Cheng. Accelerating cardiac cine MRI using a deep learning-based ESPIRiT reconstruction. *Magnetic Resonance in Medicine*, 85(1):152–167, Jan. 2021.
- [13] Martin Uecker, Peng Lai, Mark Murphy, Patrick Virtue, Michael Elad, John Pauly, Shreyas Vasanawala, and Michael Lustig. ESPIRiT—an eigenvalue approach to autocalibrating parallel MRI: where SENSE meets GRAPPA. *Magnetic resonance in medicine : official journal of the Society of Magnetic Resonance in Medicine / Society of Magnetic Resonance in Medicine*, 71, Mar. 2014.
- [14] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention Is All You Need, Dec. 2017. arXiv:1706.03762 [cs].
- [15] Jinxi Xiang, Yonggui Dong, and Yunjie Yang. FISTA-Net: Learning A Fast Iterative Shrinkage Thresholding Network for Inverse Problems in Imaging. *IEEE Transactions on Medical Imaging*, 40(5):1329–1339, May 2021. arXiv:2008.02683 [physics].
- [16] Junyuan Xie, Linli Xu, and Enhong Chen. Image Denoising and Inpainting with Deep Neural Networks. *Advances in Neural Information Processing Systems*, 1, Jan. 2012.
- [17] Yutong Xie and Quanzheng Li. Measurement-conditioned Denoising Diffusion Probabilistic Model for Under-sampled Medical Image Reconstruction. Technical Report arXiv:2203.03623, arXiv, Mar. 2022. arXiv:2203.03623 [cs, eess] type: article.
- [18] Songhyun Yu, Bumjun Park, and Jechang Jeong. Deep Iterative Down-Up CNN for Image Denoising. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 2095–2103, Long Beach, CA, USA, June 2019. IEEE.
- [19] Jian Zhang and Bernard Ghanem. ISTA-Net: Interpretable Optimization-Inspired Deep Network for Image Compressive Sensing, June 2018. arXiv:1706.07929 [cs].
- [20] Kai Zhang, Yawei Li, Jingyun Liang, Jiezhang Cao, Yulun Zhang, Hao Tang, Radu Timofte, and Luc Van Gool. Practical Blind Denoising via Swin-Conv-UNet and Data Synthesis, Mar. 2022. arXiv:2203.13278 [cs, eess].
- [21] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising. *IEEE Transactions on Image Processing*, 26(7):3142–3155, July 2017. arXiv:1608.03981 [cs].
- [22] Kai Zhang, Wangmeng Zuo, and Lei Zhang. FFDNet: Toward a Fast and Flexible Solution for CNN based Image Denoising. *IEEE Transactions on Image Processing*, 27(9):4608–4622, Sept. 2018. arXiv:1710.04026 [cs].
- [23] Yulun Zhang, Yapeng Tian, Yu Kong, Bineng Zhong, and Yun Fu. Residual Dense Network for Image Restoration, Jan. 2020. arXiv:1812.10477 [cs].
- [24] Xingjian Zhen, Rudrasis Chakraborty, Liu Yang, and Vikas Singh. Flow-based Generative Models for Learning Manifold to Manifold Mappings, Mar. 2021. arXiv:2012.10013 [cs].