GrappaNet: Combining Parallel Imaging with Deep Learning for Multi-Coil MRI Reconstruction

A. Dithering as post-processing

The GrappaNet model was trained to optimize a linear combination of Structural Similarity [5] and L1 loss between the reconstruction and the ground truth image. SSIM and L1 loss are imperfect proxies for radiologists' visual perception; optimizing SSIM, L1 loss, or a linear combination of them can produce unnaturally smooth reconstructions even when preserving diagnostic content. We can enhance the perceived sharpness of the images by adding low levels of noise, that is, by dithering. As established by [3], filtered noise ("Perlin noise") is a good model for the synthesis of natural textures — natural-looking textures include some noise. Quoting [4], "the preservation of film grain noise can also help enhance the subjective perception of sharpness in images, known as acutance in photography, although it degrades the signal-to-noise ratio. The intentional inclusion of noise in processing digital audio, image, and video data is called dither."

To avoid obscuring dark areas of the reconstruction by adding too much noise, we adapt the level of noise to the brightness of the image around each pixel. Specifically, we first normalize the image we wish to dither by dividing each pixel by the maximum pixel intensity in the image; then we blur the normalized image with a median filter taking medians over patches 11 pixels high by 11 pixel wide, then take the square root of the value at each pixel of the blurred image, and finally add to the image being dithered centered Gaussian noise of standard deviation σ times the associated blurred pixel. We set $\sigma = 0.025$ for non-fat-suppressed images and $\sigma = 0.05$ for fat-suppressed images (which have a worse native SNR).

Examples of GrappaNet reconstructions with and without noise are shown in figures 1 and 2. The dithered images look more natural, especially the PDFS images with $8\times$ undersampling. The metrics reported in the main paper do not include this added noise.

B. Training with random masks to counter adversarial examples

Compressed sensing is the reconstruction of images to a resolution beyond what reconstruction via classical signal processing would permit for the amount of measurements actually made. In MRI, the measurements are taken in k-space, and the classical signal processing involves an inverse Fourier transform. Compressed sensing reconstructs to the same resolution as if using an inverse Fourier transform on more measurements than actually taken; compressed sensing must be nonlinear to succeed. When taking measurements in k-space at fixed locations, it is relatively straightforward to construct objects whose measurements at these fixed locations will result in reconstructions from compressed sensing that are horribly wrong: simply alter arbitrarily the objects in the parts of k-space in between those locations in k-space that are actually measured. Whether such so-called "adversarial" examples of objects being measured are worrisome depends on where the actual measurements are made and (especially) on the algorithm used for reconstruction.

If the algorithm used for reconstruction is trained on a set of examples with measurements always taken at the same locations in k-space, then the reconstruction is likely to be blind to properties of objects that depend on parts of k-space in between those actually measured. The adversarial examples can then hide horrible problems in between the parts of k-space that are actually measured; the algorithm for reconstruction trained on only fixed locations in k-space will have no hope of learning how the unmeasured parts of k-space contribute to the correct reconstruction. On the contrary, if the algorithm used for reconstruction is trained on examples with measurements taken at random locations in k-space (which is particularly advantageous if each example gets measured at several different random realizations of the sampling pattern), and the random locations cover all k-space (over enough random realizations), then the algorithm is likely to learn about all parts of k-space during training (here, "all" k-space refers to the sampling pattern used for conventional reconstruction via the inverse Fourier transform at full resolution). When taking measurements at random locations in k-space, the algorithm for reconstruction will probably detect at least a piece of any adversarial attempt to hide horrible artifacts in parts of k-space, and will learn how all relevant parts of k-space affect the correct reconstruction.

Therefore, a machine-learned algorithm for reconstruction should train on examples measured at randomized locations in k-space in order to avoid some adversarial ex-

Figure 1. Example reconstructions from the GrappaNet model with $4\times$ under-sampling with and without dithering. The top row shows PD images without fat suppression and the bottom row shows PDFS images.

amples, such as those constructed by [1]. Moreover, the measurements for the validation and testing sets must also be randomized, in the following subtle sense: the locations of the measurements in k-space must be stochastically independent of the object being imaged. Ideally the object will be deterministic and the locations of the measurements in k-space will be drawn randomly independently of the object. Thus, the object being imaged should not be constructed conditional on knowing the locations in k-space of the measurements being taken; an object in physical reality

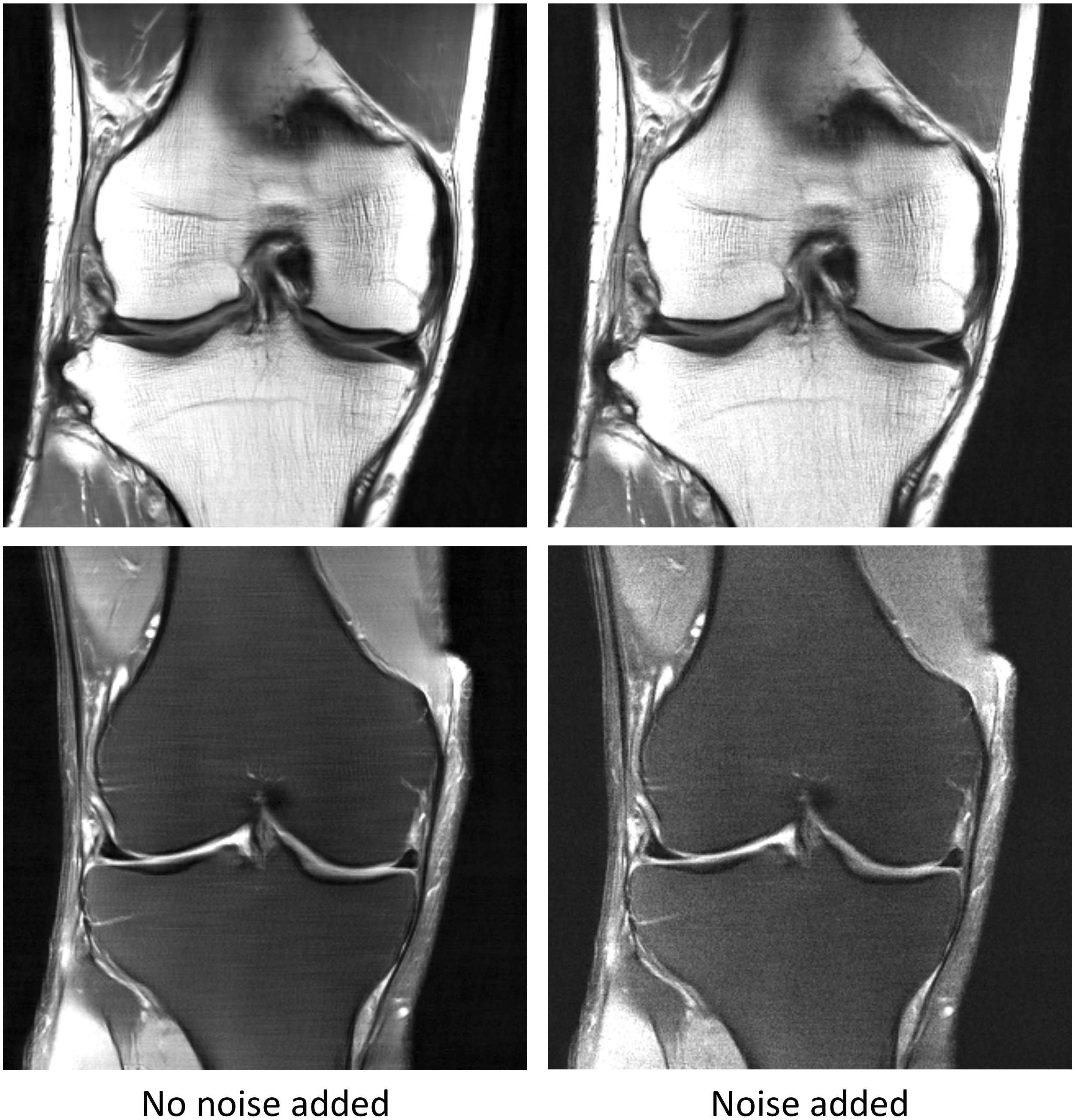

Figure 2. Example reconstructions from the GrappaNet model with $8\times$ under-sampling with and without dithering. The top row shows PD images without fat suppression and the bottom row shows PDFS images.

has no way of knowing where the measurements are being taken. The adversarial examples of [1] construct objects that depend on where the measurements are being taken, and so are inapplicable to the setting of randomized locations for the measurements. In practice, the same random locations in k-space can be used for multiple objects, provided that the objects being imaged cannot alter themselves based on knowing where the measurements are being taken, and provided that the training of any machine-learned reconstruction considers many different random locations in k-space (preferably covering all k-space over enough random realizations).

To summarize:

1. Measurements should be at randomized locations in k-space during training of machine-learned algorithms for reconstruction, such that the random locations cover all k-space (over enough random realizations), where "all" refers to the sampling pattern used for conventional reconstruction at full resolution via the inverse Fourier transform.
2. Measuring each object in the training set at multiple different random samples in k-space is ideal, constituting a kind of data augmentation that regularizes the reconstruction and improves generalization and robustness to adversarial examples.
3. The object being imaged in reality during validation and testing should be deterministic, with the random locations in k-space where measurements are taken being stochastically independent of the object.
4. When taking measurements at randomized locations in k-space, the object should not alter itself based on where measurements are made; adversarial examples are irrelevant when they are conditional on knowing the locations of the randomized measurements.
5. The same random locations in k-space can be used across the objects in the validation and testing sets (yet these locations must vary during training!).

Fortuitously, algorithms for reconstruction that obey the above conditions are also ideal for use in estimating errors via the bootstrap, as described by [2].

Regarding technologically reasonable sampling patterns, MRI works well taking measurements along the following lines:

1. radial lines in k-space, with the lines at random angles
2. parallel lines in k-space, with the lines at random offsets
3. equispaced parallel lines in k-space, with the overall offset chosen at random

In all cases, "random" means the same angle or offset for different objects being imaged in validation and testing sets, but with the angles or offsets varying at random during training and in the bootstrap or jackknife error estimation. So-called parallel imaging usually supplements the above measurements with some additional measurements of mainly low frequencies for autocalibration of sensitivity maps or of convolutional kernels for fusing contributions from multiple receiver coils, as discussed by [1] and others. The extra set of autocalibration measurements is merely a bonus, not requiring the same randomization as the other measurements.

C. Example Reconstructions

Additional example reconstructions picked at random from the validation set are shown in figures 3-6. In each case, the images on the left are the ground truths and the images on the right are the dithered reconstructions.

References

[1] Vegard Antun, Francesco Renna, Clarice Poon, Bed Adcock, and Anders C. Hansen. On instabilities of deep learning in image reconstruction — does AI come at a cost? Technical Report 1902.05300, arXiv, February 2019. Available at http://arxiv.org/abs/1902.05300. 2, 3, 4

[2] Aaron Defazio, Mark Tygert, Rachel Ward, and Jure Zbontar. Compressed sensing with a jackknife and a bootstrap. Technical Report fbooja, GitHub, 2019. Available at https://github.com/facebookresearch/fbooja. 4

[3] Ken Perlin. An image synthesizer. In *Proceedings of the Twelfth Annual Conference on Computer Graphics and Interactive Techniques*, pages 287–296. ACM, 1985. 1

[4] Tuan D. Pham. Noise-added texture analysis. In *CIARP 2016: Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, volume 10125 of *Lecture Notes in Computer Science*, pages 93–100. Springer, 2016. 1

[5] Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. 1

PD 4x

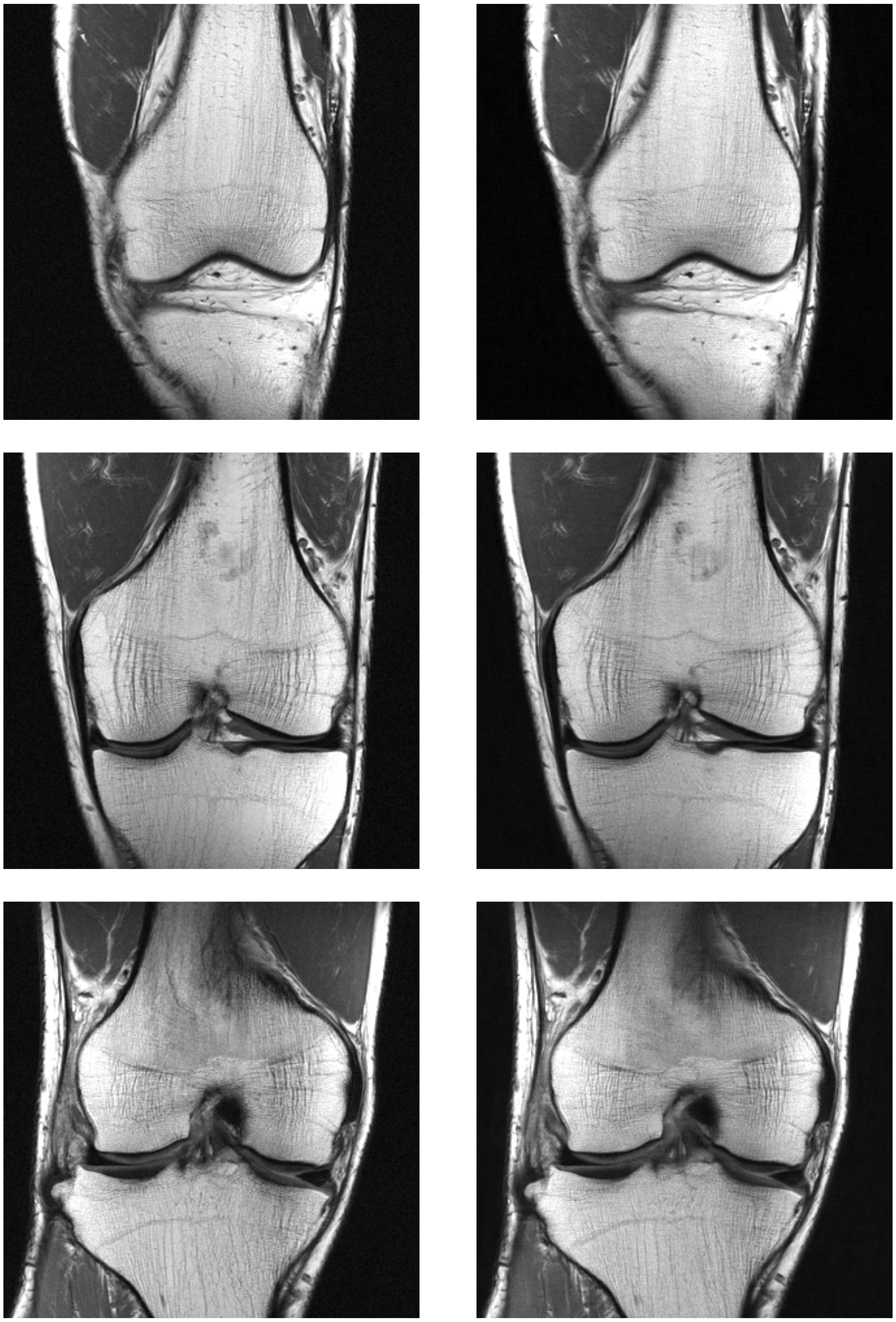

Figure 3. Proton Density with $4\times$ under-sampling.

PDFS 4x

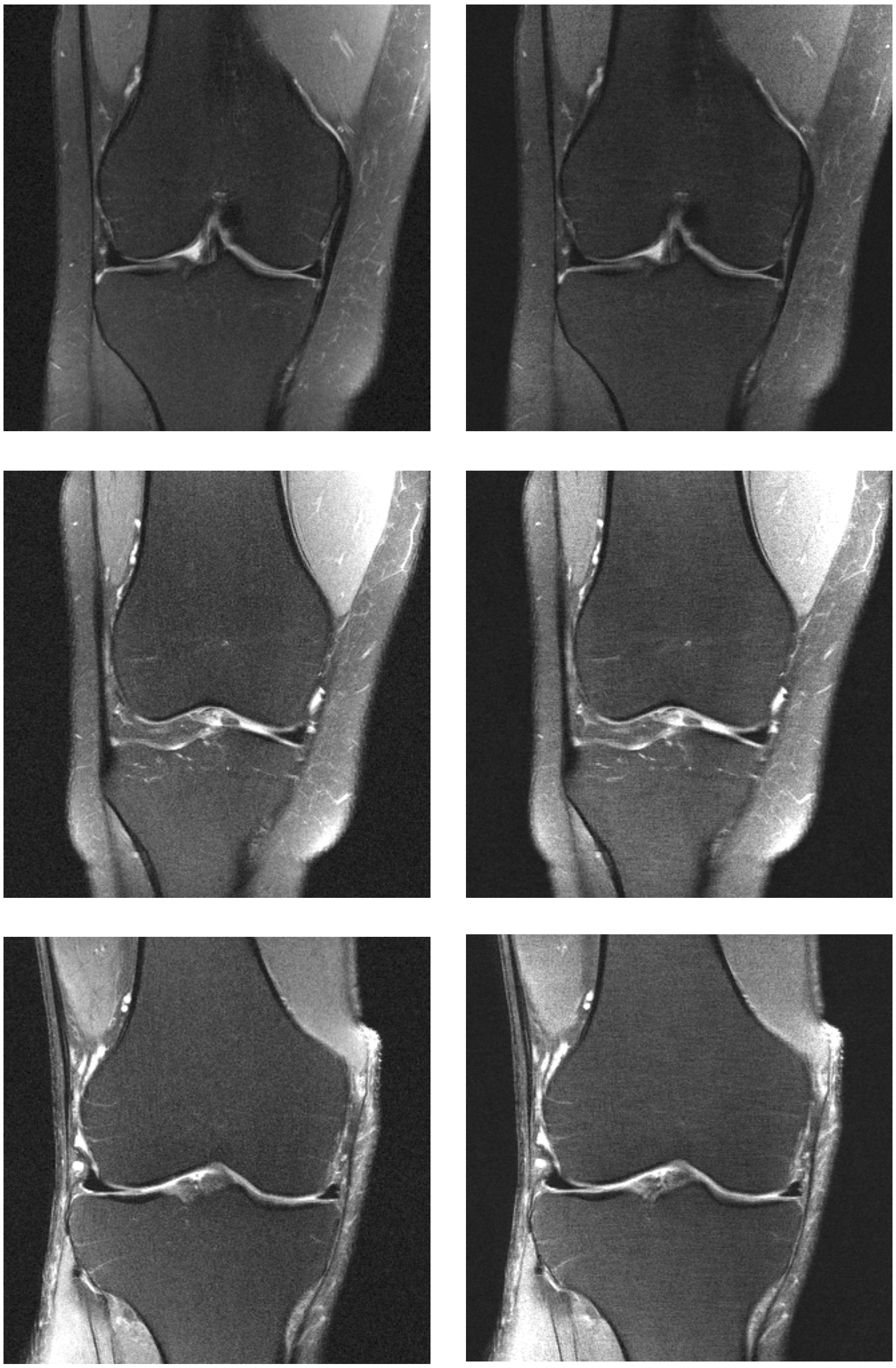

Figure 4. Proton Density with Fat Suppression with $4\times$ under-sampling.

PD 8x

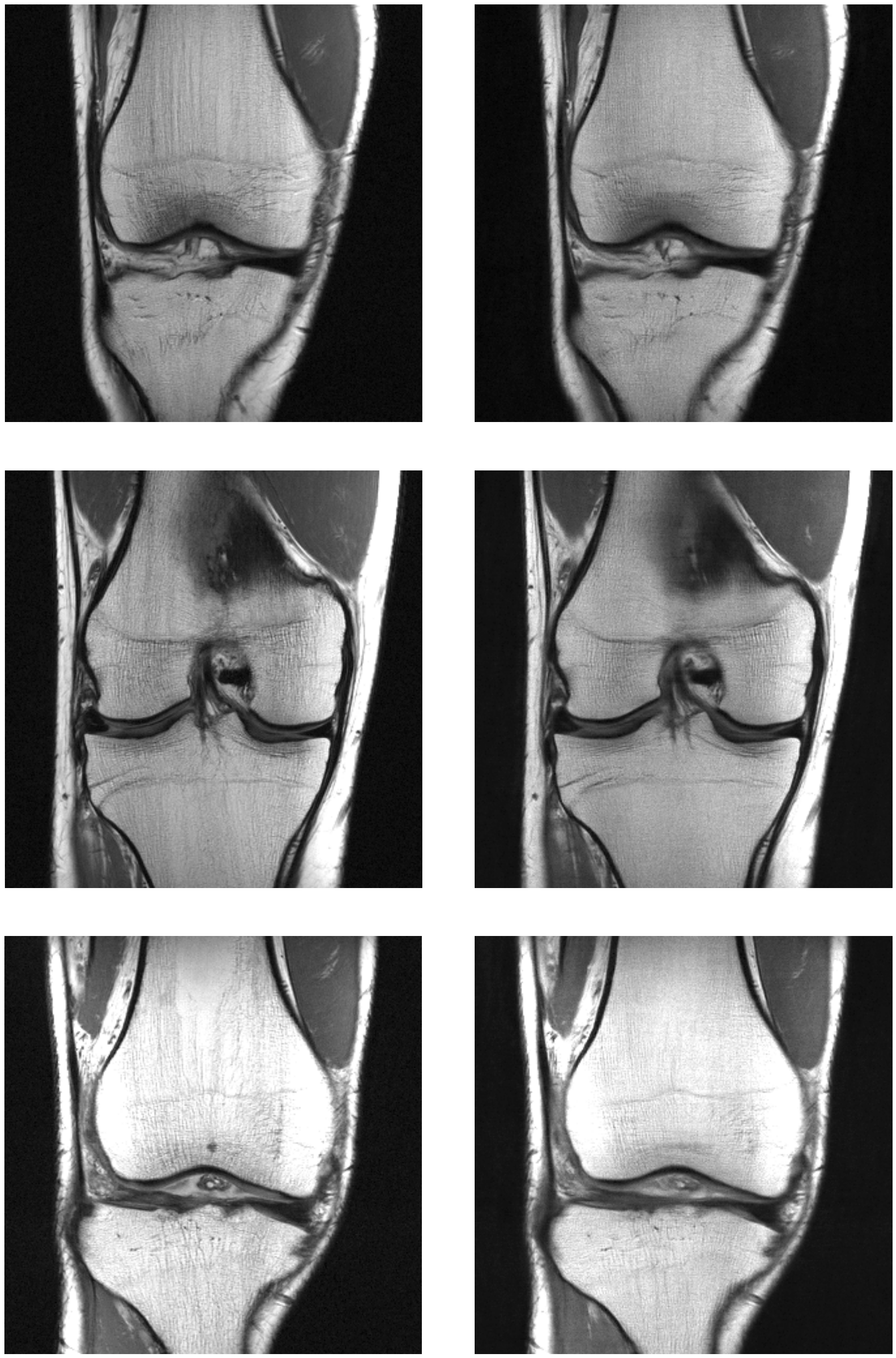

Figure 5. Proton Density with $8\times$ under-sampling.

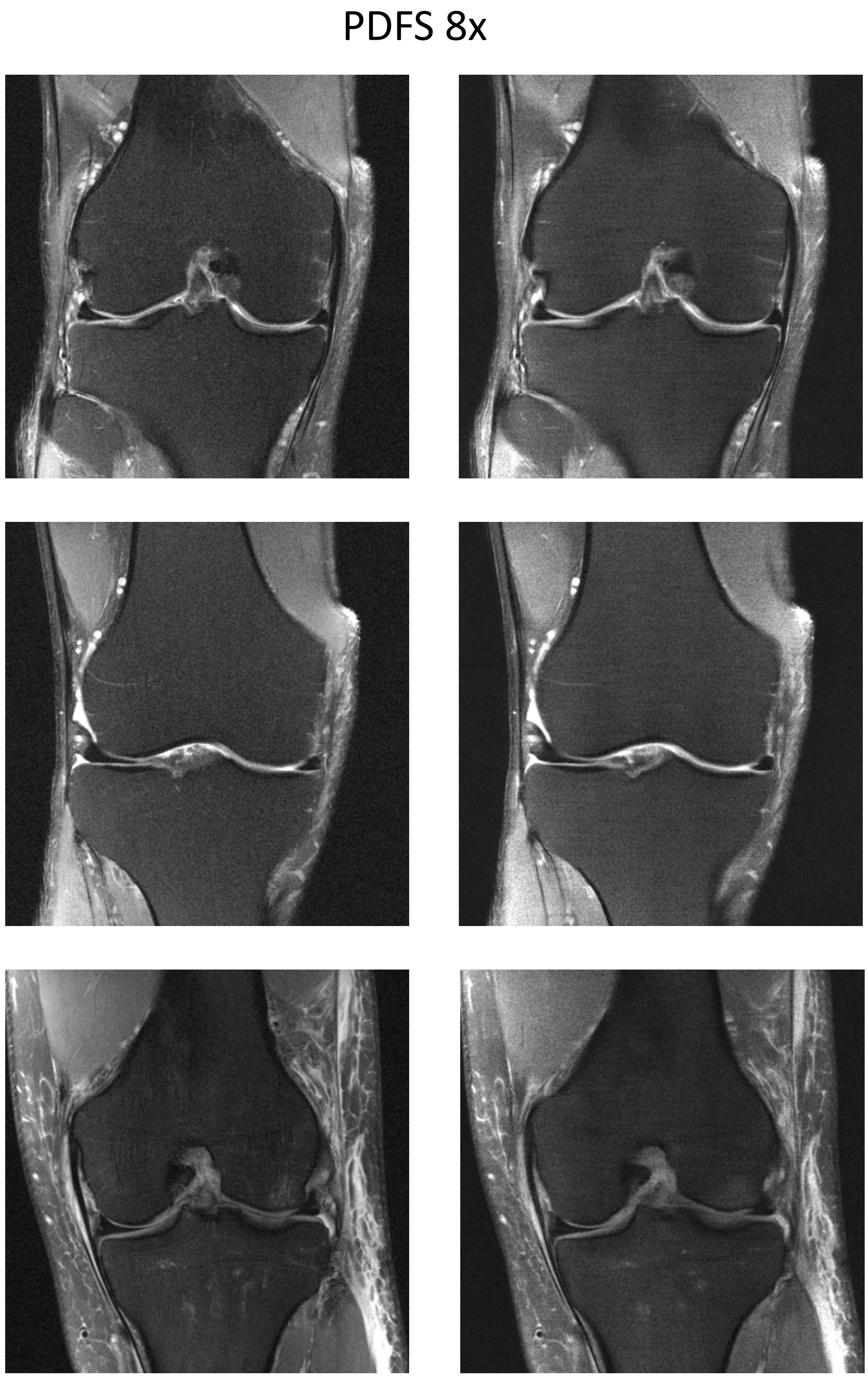

Figure 6. Proton Density with Fat Suppression with $8\times$ under-sampling.