Learned Compression of High Dimensional MRI Datasets

Qingxi Meng[1], Elizabeth Cole[1], John Pauly[1], Shreyas Vasanawala[2]
[1]Department of Electrical Engineering
[2]Department of Radiology
Stanford University
{*qingxi, ekcole, pauly, vasanawala*}*@stanford.edu*

Abstract

In many applications, such as magnetic resonance imaging (MRI), multiple images are acquired to reduce the noise of the eventual reconstructed image. However, this leads to very high dimensional datasets which have redundant information across the various acquired images. In MRI, multiple images are acquired via multiple RF coil arrays in the scanner. Afterwards, coil compression is performed to convert the original set of coil images into a smaller set of virtual coil images to enable smaller datasets and faster computation time. However, traditional iterative coil compression methods are lossy and time-consuming. In this work, we propose a novel neural network-based coil compression method in pursuit of higher reconstruction accuracy and faster coil compression. Our learned compression method achieves up to 1.5x lower NRMSE and up to 10 times runtime speed compared to traditional methods on a benchmark test dataset. We also explore 2D Convolutional Encoder for larger dataset with 32 coils and 3D Convolutional Encoder for non-cartesian dataset, which both show better performances compared to traditional methods.

1. Introduction

In many imaging applications, such as magnetic resonance imaging (MRI), multiple images are acquired of approximately the same scene in order to reduce the noise of the eventual image. In MRI, this is done by using receiver arrays with multiple coil elements [5], enabling parallel imaging (PI) [2] acceleration. Taking these multiple coil images and combining them into one image in the eventual image reconstruction process results in higher signal to noise ratio (SNR). However,the large number of coils creates prohibitively large MRI datasets because these datasets contain multiple volumetric slices per patient and potentially a dynamic dimension. Additionally, a scan for a single patient typically contains multiple sequences (10-12 sequences per scan). The size of these datasets becomes increasingly problematic in terms of memory as well as infeasible computation time for reconstruction. Coil compression algorithms are effective in mitigating this problem by compressing data from many coils into fewer virtual coils.

We propose a neural network-based framework to learn the coil compression task of mapping from our original coil space to our latent virtual coil space. Ultimately, our goal is to learn the representation of such a mapping. Inputs to the network are the original coil images from the physical coil array. The inputs contain multiple volumetric slices per patient and potentially a dynamic dimension. Outputs of the network are a set of virtual coil images.

The coil compression algorithms aim to be effective in mitigating this problem by compressing data from many coils into fewer virtual coils. We can set the number of virtual coil images by setting the number of channels in the last layer of our network. For the baseline methods, we will use the traditional state-of-the-art coil compression algorithm SVD and GCC implemented in the MRI toolbox BART [13].

2. Related Work

There are two main traditional methods for coil compression. The oldest and simplest method is to take the singular value decomposition [2] (SVD) across the set of various coil images. However, this method is slower compared to more advanced techniques. A faster method is geometric coil compression [15] (GCC). GCC removes the correlations between various coils across the fully-sampled readout dimension, allowing for reduction in data. Although GCC is faster than SVD, both are still iterative method, which are computationally expensive.

Deep learning (DL) has been successful in a wide range of MRI applications, including reconstruction [4] and segmentation [1]. However, as far as we now, researchers have not tried using DL-based approach for coil compres-

The authors Elizabeth Cole, John Pauly, and Shreyas Vasanawala are not enrolled in CS 231N

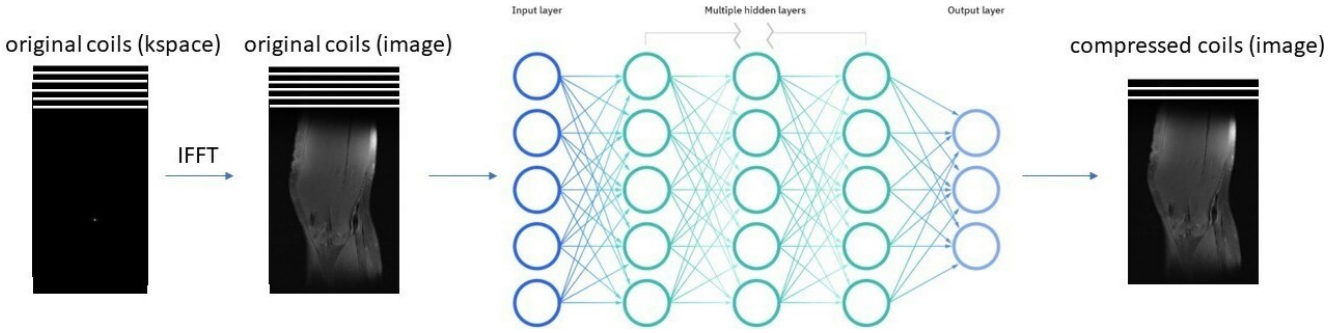

Figure 1. Framework overview of the rapid coil compression using neural networks. Our fully-connected neural network consists of 5 fully connected layers and it has 10 neurons per hidden layer. The number of channels in our input layer is equal to the number of original coils in our raw dataset. The number of channels in our output layer is equal to the number of virtual coils (the number of images in our set we wish to compress to).

sion. DL has also been used for general image compression. Specifically, the work [8] used the recurrent neural network to effectively perform lossy compression for general 2D image. Furthermore, the work [10] uses representation learning and generative model to perform lossless compression for images. Although those models perform very well on a single general image, it is shown that they do not fit in high dimensional MRI datasets [7] due to special 3D volumetric structure. In this work, we therefore propose and develop a neural network-based coil compression (NN-based CC) method to provide faster and more accurate compression on a set of images, rather than on the individual images themselves.

3. Data

In total,we train and test our mobels on three different datasets. The first dataset is the benchmark multi-coil knee dataset from fastMRI [14], which were acquired using a 15 channel knee coil array and a conventional Cartesian 2D TSE protocol on either 3T or 1.5 T scanners. Each subject volume contains slices of size 640x368 with 15 coils. There are 973 volumetric subjects with 34742 total slices during training and 56 volumetric subjects with 1959 slices during testing. For the pre-processing steps, we first apply the inverse fast Fourier transform (IFFT) to convert the original coil data from the frequency domain, otherwise known as k-space, to the image domain. We also subtract the mean value, computed on the training set, from each pixel. We also normalize the input images by the maximum value in each image.

The second dataset is provided from Stanford hospital using a 32 channel torso coil array and a conventional Cartesian 2D TSE protocol on scanners. Each subject volume contains slices of size 224x320 with 32 coils. There are 30 volumetric subjects with 1431 total slices during training and 3 volumetric subjects with 263 slices during testing. For the pre-processing steps, we first apply the inverse fast Fourier transform (IFFT) to convert the original coil data from the frequency domain, otherwise known as k-space, to the image domain. We also subtract the mean value, computed on the training set, from each pixel. We also resize the image in image domain to the same size.

Finally, we are provided with 3D non-cartesian cone dataset from Stanford hospital. Each subject volume contains slices of size 311x311x205 with 60 coils. There are 4 volumetric subjects during training and 1 volumetric subjects during testing. The non-cartesian data is stored in trajectory, coordinate and density formats. Thus, we preprocess the data by first applying nonuniform fast Fourier transform (NUFFT) to convert the data into a 3D volume. We also subtract the mean value, computed on the training set, from each pixel. We also normalize the input images by the maximum value in each image.

4. Methods

4.1. Baseline

For the baseline methods, we will use the singular value decomposition (SVD) and geometric coil compression (GCC). SVD is based on decomposing a matrix into two matrices and a vector containing scale factors called singular values. GCC improves over SVD by removing the correlations between various coils across the fully-sampled readout dimension, allowing for reduction in data. Although GCC is faster than SVD, both are still iterative method, which are computationally expensive. In addition, the compression performance of both SVD and GCC drops off when the ratio of the dimensionality of original coils to virtual coils is very high.

4.2. NN-based Coil Compression

We propose a neural network-based framework to learn the coil compression task of mapping from our original coil

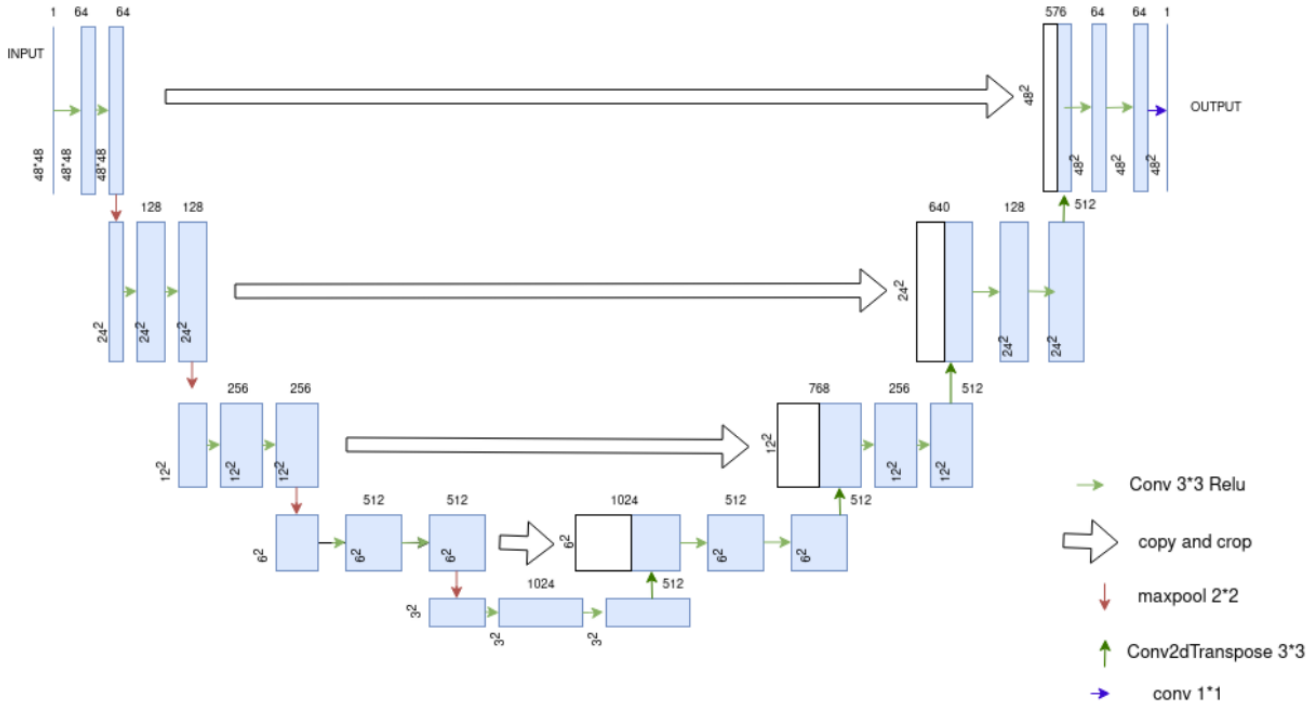

Figure 2. Framework overview of the rapid coil compression using UNet-like neural networks [12]. The number of channels in our input layer is equal to the number of original coils in our raw dataset. The number of channels in our output layer is equal to the number of virtual coils (the number of images in our set we wish to compress to).

space to our latent virtual coil space. Ultimately, our goal is to learn the representation of such a mapping. Inputs to the network are the original coil images from the physical coil array. Outputs of the network are a set of virtual coil images. We can set the number of virtual coil images by setting the number of channels in the last layer of our network.

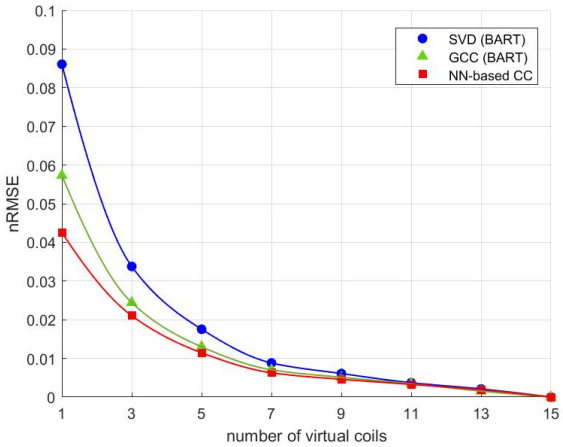

Figure 3. NRMSE vs. number of virtual coils for SVD, GCC and NN-based CC on the fastMRI test dataset.

We first apply the inverse fast Fourier transform (IFFT) to convert the original coil data from the frequency domain, otherwise known as k-space, to the image domain. Our raw data is in the frequency domain because MRI signal is collected in the sparse frequency domain, not the signal domain. We separate the volumetric scan of the patient into slices along the axial dimension. Then, we feed multi-coil images, slice-by-slice, into a fully connected neural network. In the output layer of the neural network, we generate a set of virtual coils with a lower dimensionality than the original set of coils. This dimensionality is defined by the number of channels in the last layer of our network. The framework of our model is shown in Figure 1. Specifically, we choose to use a fully-connected neural network for dataset with relatively fewer number of coils (e.g. 15 coils) because we found that it achieved very good results while model is also not over-complicated. We use a 6 layer fully-connected network with hidden units being 256, 64, 64, 32, and 32 while the output dimension is the number of virtual coils desired.

4.3. 2D CNN-based Coil Compression

For the non-cartesian datasets with large number of coils (e.g. 32 coils), we are inspired by the work UNet [9] and use a network consists of a 2-D DenseUNet [6] for efficiently extracting intra-slice features and a 3-D counterpart for hier-

archically aggregating volumetric contexts under the spirit of the auto-context algorithm for liver and tumor segmentation [10]. In our context, we use 3 convolutional layers for both with kernel size 5, and padding 2. We also use 3 convolutional transpose layers with kernel size 2 and stride 2 to gradually scale up the output size. The framework of our model is shown in Figure 1.

4.4. 3D CNN-based Coil Compression

For the non-cartesian datasets with large number of coils, we are inspired by the work faster R-CNN [3] and 3D CNN [11]. Unlike 2D-CNN architecture, which does not entirely examine the volumetric information in MRI images, we adapt our network structure to 3D convolutional layers that provide a detailed feature map exploiting the entire volumetric spatial information to incorporate both the local and global contextual information, which have been shown to perform well on 3D MRI datasets [11]. Specifically we use a network consists of 3D convolutional encoders and decoders [11]. We use three 3D convolutional layers to gradually scale down the feature size. We also use three 3D convolutional transpose layers to gradually scale up the feature size. The details about the 3D CNN are shown in Table 1, and the parameters are chosen inspired by the work [11]. One modification is that we added a fully-connected layer in-between the convolutional layers, and we found it will help the model to train better and easier.

4.5. Evaluation

We use the square root of sum-of-square (SSOS) to combine individual coil images into one image. The SSOS on an complex-valued image m with n coils is defined as:

$$SSOS(m) = \sqrt{\sum_{i=1}^{n}(|m_i|^2)} \tag{1}$$

where $|m_i|$ is the elementwise absolute value of each pixel in m.

We optimize the network parameters over the root mean square (RMS) loss between the SSOS image from the original coils and the SSOS image from the virtual coils.

5. Experiments

5.1. Experiment Setups

We train and test our models on three different datasets described in the Data section in details. Experiments were performed on both AWS server and a TITAN Xp GPU which had 64GB of RAM and 3TB of disk memory provided by Stanford Magnetic Resonance Systems Research Laboratory (MRSRL). We use normalized root-mean-square-error (NRMSE) between the original SSOS image and the compressed SSOS image to measure the compression accuracy. We compare our NN-based CC method with SVD and GCC, both implemented in BART [13], on the fastMRI dataset. The hyperparameters we choose for each model are shown in Table 2. As shown in Table 2, we choose the smaller learning rates for the more complex models (i.e. 2d-Conv Encoder and 3D-Conv Encoder) because we found that it helps the model to converge better. For the batch size, we choose batch size to be 1 for 3D non-cartesian dataset because each data input takes a huge amount of memory. For the optimizers, we found that SGD + Momentum with default parameters works slightly better for the 2D-Conv Encoder.

5.2. Experimental Design

We perform different experiments on different datasets. For the 15 coil datasets, we apply the NN-based coil compression for different number of coils ranging from 1 to 15. We also compare the our results with SVD and GCC implemented by BART [13] on one MRI slice (slice 1504). In addition, we also compare the magnitude of 5 virtual coils on a randomly chosen slice in the fastMRI knee training dataset for SVD, GCC and NN-based CC. For the 32 coil datasets, we perform the experiment to compress the datasets from 32 coils to 4 coils using SVD, fully-connected network, and the 2D CNN-based coil compression we proposed. For non-cartesian datasets with 60 coils, we perform the experiment to compress the datasets from 60 coil to 10 coils using SVD, the fully connected network, and the 3D CNN-based coil compression we proposed.

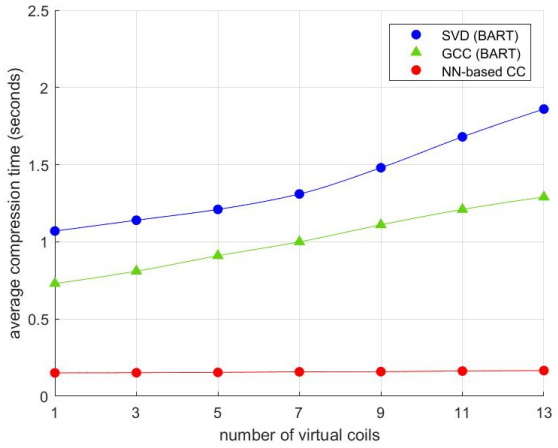

Figure 4. Average compression time vs. number of virtual coils for SVD, GCC and NN-based CC on the knee test dataset. NN-based CC was about 10 times faster than SVD and GCC. The compression time of our model was nearly the same for different numbers of virtual coils while the compression time for SVD and GCC increased as the number of virtual coils increased.

Layer	Type	Kernel Size	Number of Filters	Stride	Padding	FC Units
#1	3D-Conv+Max Pooling + BN	(2, 2, 2)	32	(1, 1, 1)	(2, 2, 2)	-
#2	3D-Conv+Max Pooling + BN	(2, 2, 2)	64	(1, 1, 1)	(2, 2, 2)	-
#3	3D-Conv+Max Pooling + BN	(2, 2, 2)	128	(1, 1, 1)	(2, 2, 2)	-
#4	FC (Softmax)	-	-	-	-	256
#5	3D-ConvTransposed+Max Pooling + BN	(2, 2, 2)	128	(2, 2, 2)	(2, 2, 2)	-
#6	3D-ConvTransposed+Max Pooling + BN	(2, 2, 2)	64	(2, 2, 2)	(2, 2, 2)	-
#7	3D-ConvTransposed+Max Pooling + BN	(2, 2, 2)	#output coils	(2, 2, 2)	(2, 2, 2)	-

Table 1. Model configurations

5.3. Results and Discussion

For the 15 coil datasets, compression accuracy, measured as the NRMSE between SSOS images from original coils and virtual coils, is shown in Figure 3 for different numbers of virtual coils and different coil compression methods. Our NN-based CC consistently achieved smaller NRMSE compared to SVD and GCC. Figure 4 shows the compression time for SVD, GCC and our NN-based CC. Our NN-based CC was much faster than the traditional methods across all numbers of virtual coils. In Figure 5, we also visualize the SSOS image from virtual coils and its difference from the original SSOS image for SVD, GCC and our NN-based CC. Information contained in each virtual coil compressed by SVD, GCC and our NN-based CC is shown in Figure 6. For the 32 coil datasets, we visualize the SSOS image from virtual coils from the SVD, fully-connected network and the 2D Convolutional Encoder. Information contained in each virtual coil compressed by SVD, fully-connected network and the 2D Convolutional Encoder is shown in Figure 7. For the non-cartesian datasets, in Table 3, we compare the nRMSE for SVD, GCC and the 3D Convolutional Encoder on the test dataset when we compress from 60 coils to 10 coils.

As shown in Figure 3, NN-based CC out-performs the traditional methods SVD and GCC. Also, it performs much better than the traditional methods when the dimensionality of the virtual coils is very small. Specifically, our method achieves more than 1.5x improvement in NRMSE over both SVD and GCC when the set of original coils is compressed to 1 virtual coil (i.e., a final composite image).

As shown in Figure 4, another main advantage of the NN-based CC is that the compression time is much faster than the traditional methods. Although DL models take a long time to train - in our case, approximately 8 hours - training only needs to be done once. For our task of coil compression, inference time using our method is significantly reduced compared to the processing time of traditional iterative methods.

As shown in Figure 5 and 6, we could observe that the SSOS image from our model had smaller compression loss compared to both SVD and GCC. Furthermore, we could also observe that the information for NN-based CC was spread across each virtual coil, so it was more robust to the noise.

As shown in Figure 7, we could observe that the SSOS image from our 2D-Conv Encoder had smaller compression loss and looked better compared to both SVD and fully-connected network. We could also observe that the the compressed image in frequency domain for the SVD had lots of noise, which demonstrated that DL-based compression methods are more robust to the noise in frequency domain.

As shown in Table 3, we also compare the performance

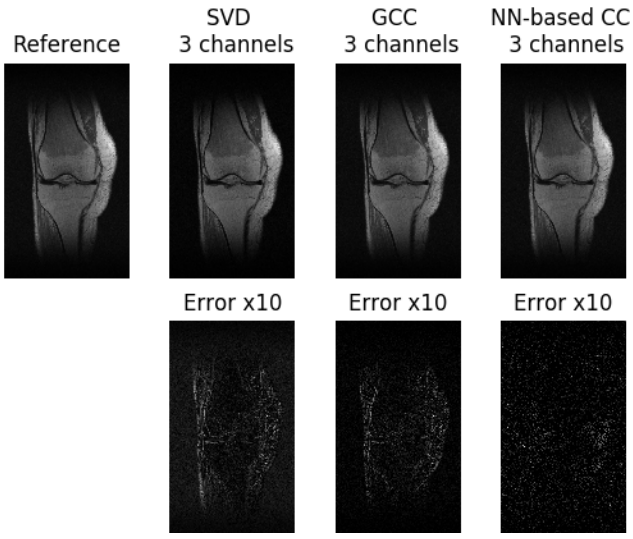

Figure 5. Comparison of SVD, GCC and NN-based CC. The original image (slice 1504 in the fastMRI knee training dataset) was used as the reference in the first column of the first row. The compression results of SVD, GCC and NN-based CC with 3 virtual coils were shown in the second, third and fourth column of the first row, respectively. Compression errors (10× in SVD, GCC and NN-based CC) were shown in the second row. The SSOS image from our model had smaller compression loss compared to both SVD and GCC.

Model Type	learning rate	batch size	optimizer	#epoches
NN-based CC (FC)	0.05	4	Adam	20
2D-Conv Encoder	0.001	4	SGD + Momentum	30
3D-Conv Encoder	0.00015	1	Adam	20

Table 2. Model hyperparameters

of our 3D-Conv Enocoder with SVD and GCC on the non-cartesian datasets. We could see that both the nRMSE and the average compression time for 3D-Conv Encoder is much better than SVD and GCC, which shows potential of our methods for various different types of datasets.

As shown in figure 8, we also compare the fully connected network's nRMSE loss curve with that of 2D-Conv encoder. We observe that the loss curve of 2D-Conv encoder is smoother and decrease faster. The reason may be that the 2-D DenseUNet [6] helps efficiently extracting intra-slice features and a 3-D counterpart for hierarchically aggregating volumetric contexts so that the model learns faster.

6. Conclusion

In this work, we propose a neural network-based framework to learn the representation of the coil compression task, where we wish to map from the space of a higher dimensional set of images to the space of a lower dimen-

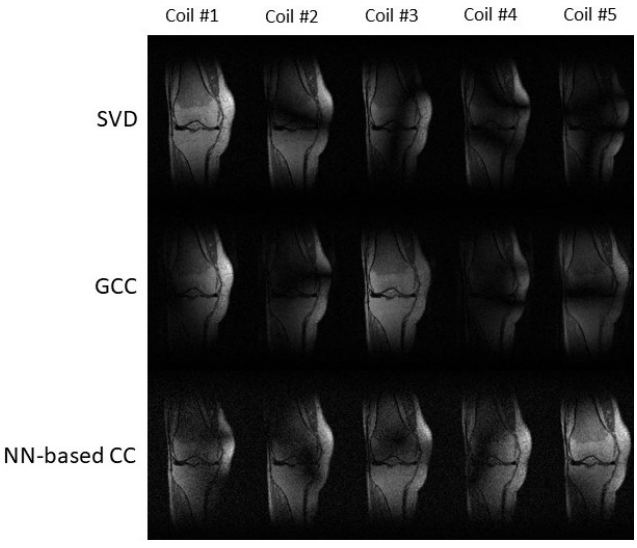

Figure 6. Magnitude of 5 virtual coils on a randomly chosen slice in the fastMRI knee training dataset for SVD (first row), GCC (second row) and NN-based CC (third row) respectively. For SVD and GCC, the information is concentrated in one or two virtual coils. For NN-based CC, the information was spread across each virtual coil, so it was more robust to the noise.

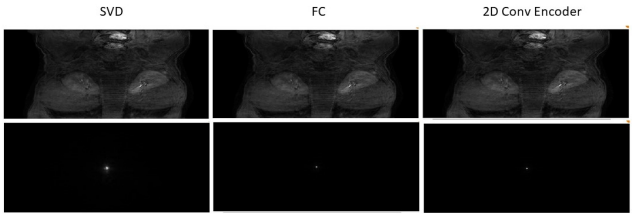

Figure 7. Comparison of SVD, NN-based CC, and 2D Conv Encoder on slice 54 in the 32 coil torse training dataset. The compression results of SVD, NN-based CC, and 2D Conv Encoder with 4 virtual coils were shown in the first, second and third column of the first row, respectively. Compressed image in frequency domain were shown in the second row.

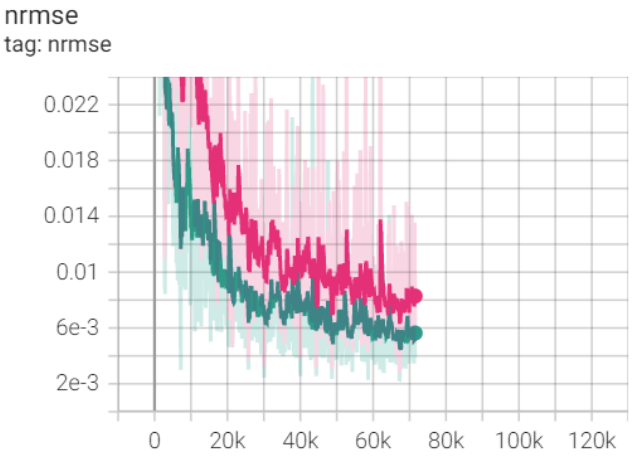

Figure 8. The nRMSE loss curve for the fully-connected neural network and 2D-Conv encoder

Model	nRMSE	Avg Compression Time (secs)
SVD	0.18	0.07
GCC	0.09	0.15
3D-Conv Encoder	0.05	0.001

Table 3. Comparison of performance of SVD, GCC, and 3D-Conv Encoder on non-cartesian 60 coil datasets

sion set of images. To our knowledge, this is the first time MRI coil compression has been learned. Our learned representation achieves a higher compression accuracy compared to the two traditional state of the art iterative methods. Our method also achieves up to 10 times faster compression speed.

One current large limitation is that a different model needs to be trained for every different dimensionality of virtual coils. For future work, we will explore generalizing one model to any dimensionality of virtual coils. Related to this model generalization is developing a model which is robust to dataset shift.

Another future work is to try data augmentation or transfer learning since our non-cartesian data is very limited. Another interesting direction to further explore is to train in the frequency domain instead of image domain. We tried training in frequency domain but the results become very bad mainly because the image is very sparse in the frequency domain. However, we could potentially design our network specifically for the sparse data as a future work.

7. Involvement of non-CS 231N contributors

Elizabeth Cole help discuss the methods, design and train the model. John Pauly and Shreyas Vasanawala help provide the datasets, the computing resources and advise on the project.

References

[1] Zeynettin Akkus, Alfiia Galimzianova, Assaf Hoogi, Daniel L Rubin, and Bradley J Erickson. Deep learning for brain mri segmentation: state of the art and future directions. *Journal of digital imaging*, 30(4):449–459, 2017. 1

[2] Martin Buehrer, Klaas P Pruessmann, Peter Boesiger, and Sebastian Kozerke. Array compression for mri with large coil arrays. *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine*, 57(6):1131–1139, 2007. 1

[3] Yu-Wei Chao, Sudheendra Vijayanarasimhan, Bryan Seybold, David A Ross, Jia Deng, and Rahul Sukthankar. Rethinking the faster r-cnn architecture for temporal action localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1130–1139, 2018. 4

[4] Feiyu Chen, Valentina Taviani, Itzik Malkiel, Joseph Y Cheng, Jonathan I Tamir, Jamil Shaikh, Stephanie T Chang, Christopher J Hardy, John M Pauly, and Shreyas S Vasanawala. Variable-density single-shot fast spin-echo mri with deep learning reconstruction by using variational networks. *Radiology*, 289(2):366–373, 2018. 1

[5] Chris D Constantinides, Ergin Atalar, and Elliot R McVeigh. Signal-to-noise measurements in magnitude images from nmr phased arrays. *Magnetic Resonance in Medicine*, 38(5):852–857, 1997. 1

[6] Steven Guan, Amir A Khan, Siddhartha Sikdar, and Parag V Chitnis. Fully dense unet for 2-d sparse photoacoustic tomography artifact removal. *IEEE journal of biomedical and health informatics*, 24(2):568–576, 2019. 3, 6

[7] Daniel T Huff, Amy J Weisman, and Robert Jeraj. Interpretation and visualization techniques for deep learning models in medical imaging. *Physics in Medicine & Biology*, 66(4):04TR01, 2021. 2

[8] Nick Johnston, Damien Vincent, David Minnen, Michele Covell, Saurabh Singh, Troy Chinen, Sung Jin Hwang, Joel Shor, and George Toderici. Improved lossy image compression with priming and spatially adaptive bit rates for recurrent networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4385–4393, 2018. 2

[9] Xiaomeng Li, Hao Chen, Xiaojuan Qi, Qi Dou, Chi-Wing Fu, and Pheng-Ann Heng. H-denseunet: hybrid densely connected unet for liver and tumor segmentation from ct volumes. *IEEE transactions on medical imaging*, 37(12):2663–2674, 2018. 3

[10] Kang Liu, Dong Liu, Li Li, Ning Yan, and Houqiang Li. Semantics-to-signal scalable image compression with learned revertible representations. *International Journal of Computer Vision*, 129(9):2605–2621, 2021. 2, 4

[11] Hiba Mzoughi, Ines Njeh, Ali Wali, Mohamed Ben Slima, Ahmed BenHamida, Chokri Mhiri, and Kharedine Ben Mahfoudhe. Deep multi-scale 3d convolutional neural network (cnn) for mri gliomas brain tumor classification. *Journal of Digital Imaging*, 33(4):903–915, 2020. 4

[12] Milan Tripathi. Facial image denoising using autoencoder and unet. *Heritage and Sustainable Development*, 3(2):89–96, 2021. 3

[13] Martin Uecker, Jonathan I Tamir, Frank Ong, and Michael Lustig. The bart toolbox for computational magnetic resonance imaging. In *Proc Intl Soc Magn Reson Med*, volume 24, 2016. 1, 4

[14] Jure Zbontar, Florian Knoll, Anuroop Sriram, Tullie Murrell, Zhengnan Huang, Matthew J Muckley, Aaron Defazio, Ruben Stern, Patricia Johnson, Mary Bruno, et al. fastmri: An open dataset and benchmarks for accelerated mri. *arXiv preprint arXiv:1811.08839*, 2018. 2

[15] Tao Zhang, John M Pauly, Shreyas S Vasanawala, and Michael Lustig. Coil compression for accelerated imaging with cartesian sampling. *Magnetic resonance in medicine*, 69(2):571–582, 2013. 1