CS230

Boosting MRI Image Quality with Deep Learning

Jordan Greenberg
Department of Computer Science
Stanford University
jgberg@stanford.edu

Avanika Narayan
Department of Computer Science
Stanford University
avanikan@stanford.edu

Alex Walczak
Department of Computer Science
Stanford University
awal@berkeley.edu

Abstract

MRI scans are highly effective in diagnosing an abundance of medical conditions. However, the standard processes for generating MR images from raw sensor inputs are timely and rely heavily on expert knowledge. Image reconstruction is difficult because knowledge of the exact inverse transform between the output image and raw sensor data is unknown a priori. Thus, traditional techniques produce low-quality output images in sub-optimal conditions (such as when the subject moves). Drawing inspiration from the AUTOMAP model [6], we have developed a neural network that learns the mapping between the sensor and image domain of MRI knee scans. The model yields higher quality images when compared with standard reconstruction methods applied to the same dataset, achieving a peak signal-to-noise ratio of 28.2.

1 Introduction

In this paper we explore the viability of an end-to-end deep learning framework for MRI image reconstruction that operates in the absence of expert knowledge. Traditional image reconstruction methods involve hand-crafted, sequential modular reconstruction chains composed of several signal processing stages that may include discrete transforms, data interpolation techniques, nonlinear optimization and various filtering mechanisms. This is a slow and computationally intensive process. As such, we propose a unified deep learning-based image reconstruction framework that can learn the reconstruction relationship between sensor and image domain without expert knowledge. Our algorithm takes as input a complex-valued k-space image, which represents the intermediate representation of the object in the sensor domain. A CNN is then used to output a reconstructed MRI image that maps to the the raw input sensor data.

2 Related work

CNNs for MRI reconstruction. The primary source used in this study is the state-of-the-art AUTOMAP image reconstruction model [6]. AUTOMAP uses a CNN to learn reconstruction transforms. The key strength of the model is its ability to produce accurate MR images, even reducing some reconstruction artifacts. One weakness is the model's massive FC (fully connected) layers [1]. As a result, AUTOMAP has limited ability to generalize to inputs of different sizes. Our work builds upon the primary architecture structure of the AUTOMAP model (same number of convolutional and deconvolutional layers). However, our model differs with regards to the domain of the input image (knee vs. brains scans), the total number of model parameters, and the dimensions of the input and output images. Schlemper *et al.* [3] discuss a deep cascade of CNN's for MRI reconstruction. This model's strengths lie in its ability to quickly reconstruct images (23ms) while also yielding error rates lower than state-of-the-art dictionary learning-based MRI (DLMRI) reconstruction techniques. A key insight of this model was to use undersampling masks to increase the speed of the reconstruction process. This undersampling step is what differentiates the model in [2] from our work as we do not perform any undersampling on the input image.

GANs for MRI reconstruction. There exists some literature [4] on the use of GAN based architectures, called DAGAN, for MRI reconstruction. The primary contribution of this implementation is the superior reconstruction quality when compared with conventional CS-MRI methods and the reduced processing time (5ms) which is suitable for real-time processing. This model differs significantly from our work which uses a CNN based approach. Also, there may be risk using GANs for medical images as they are known to hallucinate content.

Undersampled MRI reconstruction. CNNs have been used for undersampled MRI reconstruction. Using U-net, a CNN that specializes in biomedical image segmentation, it has been shown [2] that only 29% of k-space data is necessary to generate high quality reconstructions. A key contribution of this paper is the use of a k-space correction in addition to the deep learning framework. This study differs from our work in that 1) it uses the U-net 2) the input images are undersampled 3) a k-space correction is used along with the deep learning model.

Limitations of DL in image reconstruction. The state-of-the-art work has been performed in understanding the instabilities of deep learning for image reconstruction [1]. The novel contributions of the paper are in empirically proving the inherent instability of DAGAN and AUTOMAP models. Specifically how these models react to tiny perturbations in the input. This work provided guidance on how we could best improve the AUTOMAP model in order to reduce instability.

3 Dataset and Features

We used the fastMRI dataset [5] published by Facebook AI Research and NYU Langone Health. In total, the dataset contains includes 1,594 knee volumes for a total of 56,987 slices. All files are stored as HDF5 files. 90% of the data is used as the training set, with the remainder split evenly to create validation and test sets. The HDF5 files in test dataset include an undersampled k-space images and a mask which defines the undersampled cartesian k-space trajectory.

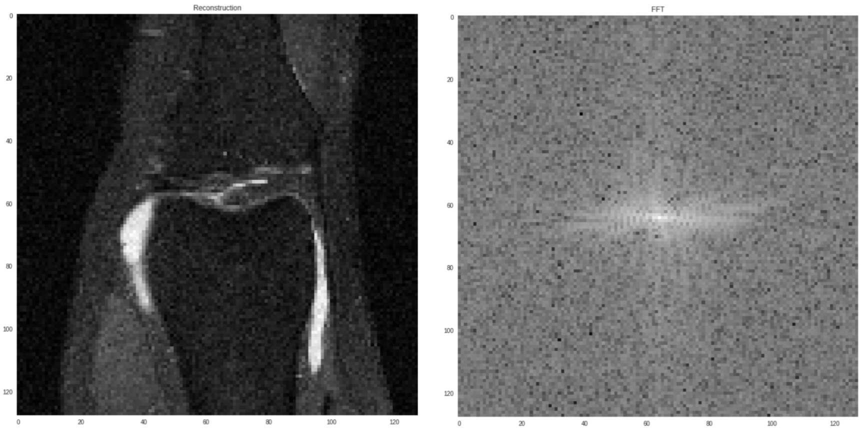


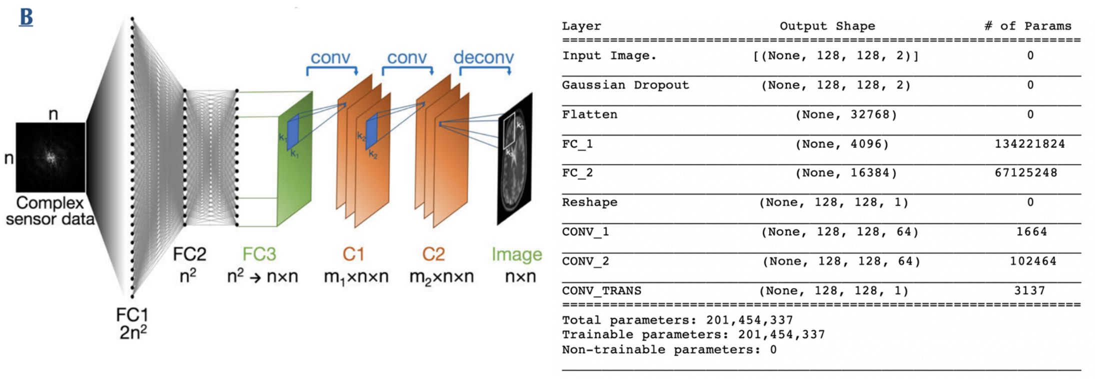
Figure 1: MRI knee scan (left) and its corresponding k-space magnitude image (right).

3.1 Data Preprocessing

Each of the HDF5 files in the dataset contains an MRI volume of k-space images of shape (640, 384). This was used to produce reconstructions (via inverse Fourier Transform) of the same shape, which we center-cropped to (320, 320) and then downsampled to (128, 128). In addition, we normalized the target images to the range [0, 1] and performed data augmentation on the training set by randomly cropping a tiling of the four reflections. The purpose of this step was to help our model become more robust to variances in data inputs such as translations. For each slice, we split the real and imaginary components of the complex-valued k-space data into separate channels. As a result, our k-space input batches have shape (N, 128, 128, 2) and our target reconstruction batches have shape (N, 128, 128, 1).

4 Methods

Our algorithm is based on an existing implementation called AUTOMAP (AUtomated TransfOrm by Manifold APproximation). The specifics of the model architecture are shown in *Figure 2*.



```
Layer                        Output Shape                # of Params
====================================================================
Input Image.              [(None, 128, 128, 2)]                0
____________________________________________________________________
Gaussian Dropout           (None, 128, 128, 2)                 0
____________________________________________________________________
Flatten                         (None, 32768)                  0
____________________________________________________________________
FC_1                            (None, 4096)               134221824
____________________________________________________________________
FC_2                            (None, 16384)              67125248
____________________________________________________________________
Reshape                    (None, 128, 128, 1)                 0
____________________________________________________________________
CONV_1                     (None, 128, 128, 64)               1664
____________________________________________________________________
CONV_2                     (None, 128, 128, 64)              102464
____________________________________________________________________
CONV_TRANS                 (None, 128, 128, 1)                3137
====================================================================
Total parameters: 201,454,337
Trainable parameters: 201,454,337
Non-trainable parameters: 0
____________________________________________________________________
```

Figure 2: AUTOMAP architecture (left) Input/output shapes of all layers of the model. (right).

Our model uses a modification of the AUTOMAP architecture with MSE loss. The original AUTOMAP architecture had 806 million parameters: over 4x the number of parameters in our model (201M). We note that this modification is a key contribution of this paper - adapting the AUTOMAP model such that it is computationally feasibly to train. As a quantitative evaluation metric, we used a peak signal-to-noise ratio: a function of the mean square error between a reconstruction and the expected image measured in decibels as shown, where MAX is the maximum pixel value: $PSNR(reconstruction, image) = 20 * log_{10}(MAX) - 10 * log_{10}(MSE)$.

5 Experiments/Training

With regards to hyperparameters, we set the batch size to 32 and the initial learning rate to 2e-4. This was the largest batch size that fit on a p3.2xlarge EC2 instance. The initial learning rate used was provided in the AUTOMAP paper (starting orders of magnitude higher or lower resulted in slower training). A key difference between our model and the AUTOMAP model was the optimizer; Zhu, B. et al. used RMSProp while we used Adam. This is due to the fact that Adam automatically sets the learning rates for each layer, requiring less tuning. We also observed faster training using a smaller L1 activity regularization constant (1e-6 instead of 1e-4 in AUTOMAP). Finally, we shrunk the first two FC layers of AUTOMAP, which led to the drastic decrease in model size and improvement in training speed (300 epochs of training lasted 12 hours).

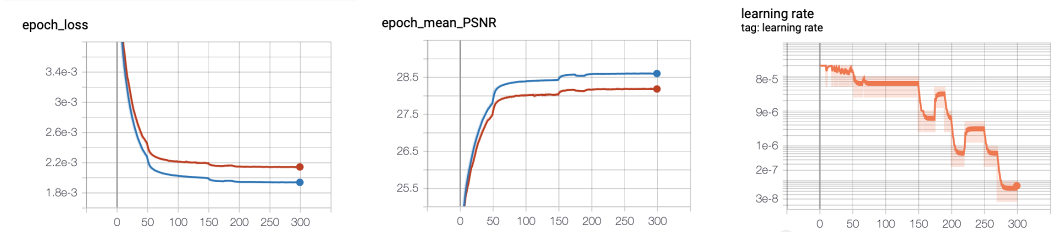


Figure 3: Loss and PSNR curves on training (blue) and validation (red) sets. Learning rate over time.

Significant time was spent on architecture selection - specifically, in choosing the placement and size of FC layers in our model. While FC layers improve the output resolution of the reconstructed images, they are computationally expensive. This lead to several model iterations in which we tested different positions and counts of FC layers in our model, as well as using fully convolutional architectures.

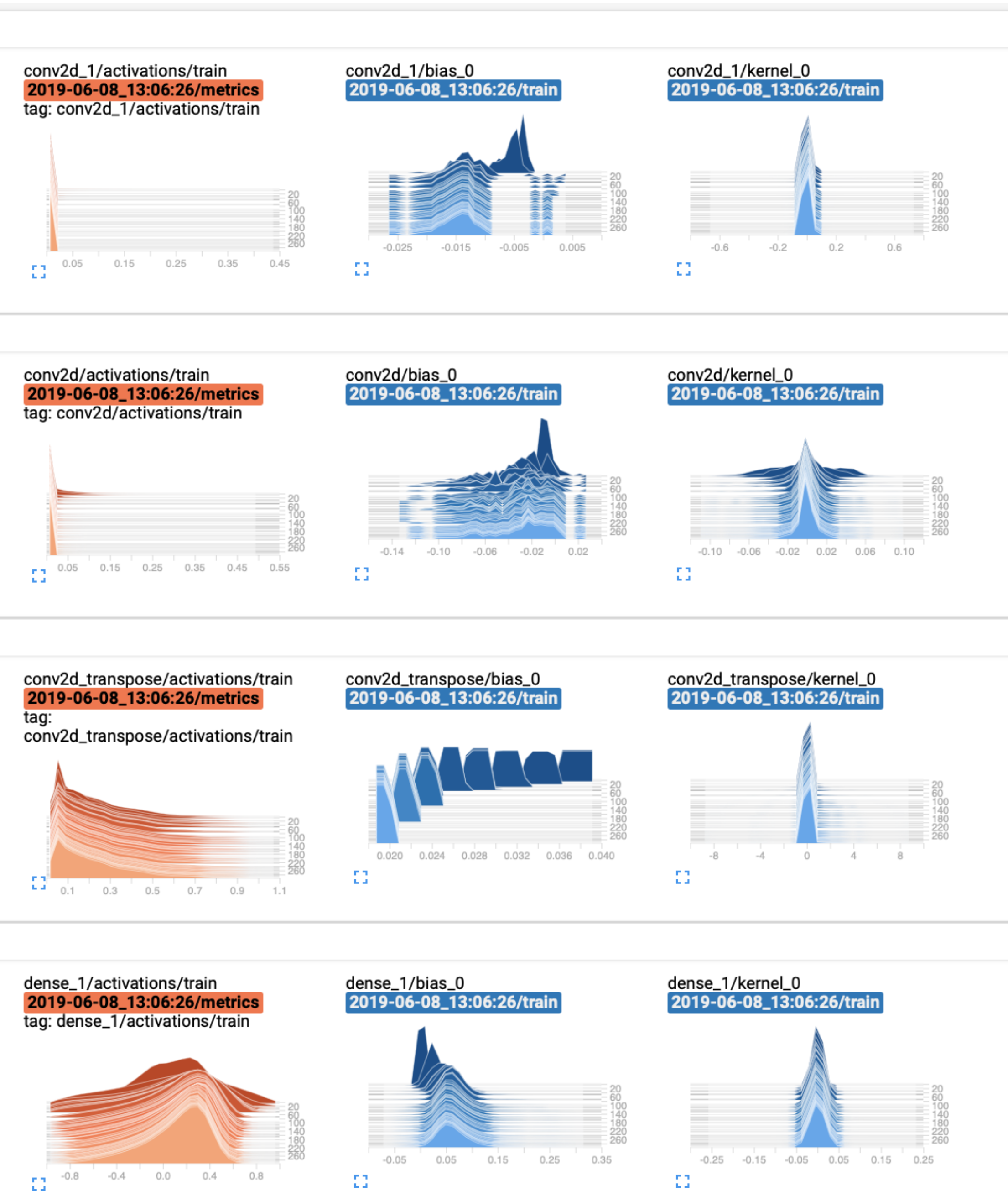

conv2d_1/activations/train
2019-06-08_13:06:26/metrics
tag: conv2d_1/activations/train
0.05 0.15 0.25 0.35 0.45
conv2d_1/bias_0
2019-06-08_13:06:26/train
-0.025 -0.015 -0.005 0.005
conv2d_1/kernel_0
2019-06-08_13:06:26/train
-0.6 -0.2 0.2 0.6
conv2d/activations/train
2019-06-08_13:06:26/metrics
tag: conv2d/activations/train
0.05 0.15 0.25 0.35 0.45 0.55
conv2d/bias_0
2019-06-08_13:06:26/train
-0.14 -0.10 -0.06 -0.02 0.02
conv2d/kernel_0
2019-06-08_13:06:26/train
-0.10 -0.06 -0.02 0.02 0.06 0.10
conv2d_transpose/activations/train
2019-06-08_13:06:26/metrics
tag: conv2d_transpose/activations/train
0.1 0.3 0.5 0.7 0.9 1.1
conv2d_transpose/bias_0
2019-06-08_13:06:26/train
0.020 0.024 0.028 0.032 0.036 0.040
conv2d_transpose/kernel_0
2019-06-08_13:06:26/train
-8 -4 0 4 8
dense_1/activations/train
2019-06-08_13:06:26/metrics
tag: dense_1/activations/train
-0.8 -0.4 0.0 0.4 0.8
dense_1/bias_0
2019-06-08_13:06:26/train
-0.05 0.05 0.15 0.25 0.35
dense_1/kernel_0
2019-06-08_13:06:26/train
-0.25 -0.15 -0.05 0.05 0.15 0.25
20 60 100 140 180 220 260

Figure 4: The weights of the large FC layers ("dense") did not change much during training.

6 Results/Discussion

The primary metric we used in assessing the accuracy of our model was peak signal-to-noise ratio (equation above). Our model achieved a PSNR of 28.2 on validation data, outperforming the standard total variational baseline (having PSNR 25.9) trained on the same dataset [5]. The paper for the AUTOMAP model did not release PSNR values for single-coil MRI data.

In *Figure 4*, we observe the distribution of weights and biases for almost all of the layers in our model. We found that the FC layers (labeled "dense") did not change much during training (only the biases shifted). However, without FC layers, we could not train a good model. We hypothesize that the differences in FC layer weights could have been too small to visualize. We also propose that the FC layers did not help much in training and theorize that the flattening of k-space data was what helped the most. After all, we created a fully convolutional network with more layers and just as many parameters, but it could not learn a mapping.

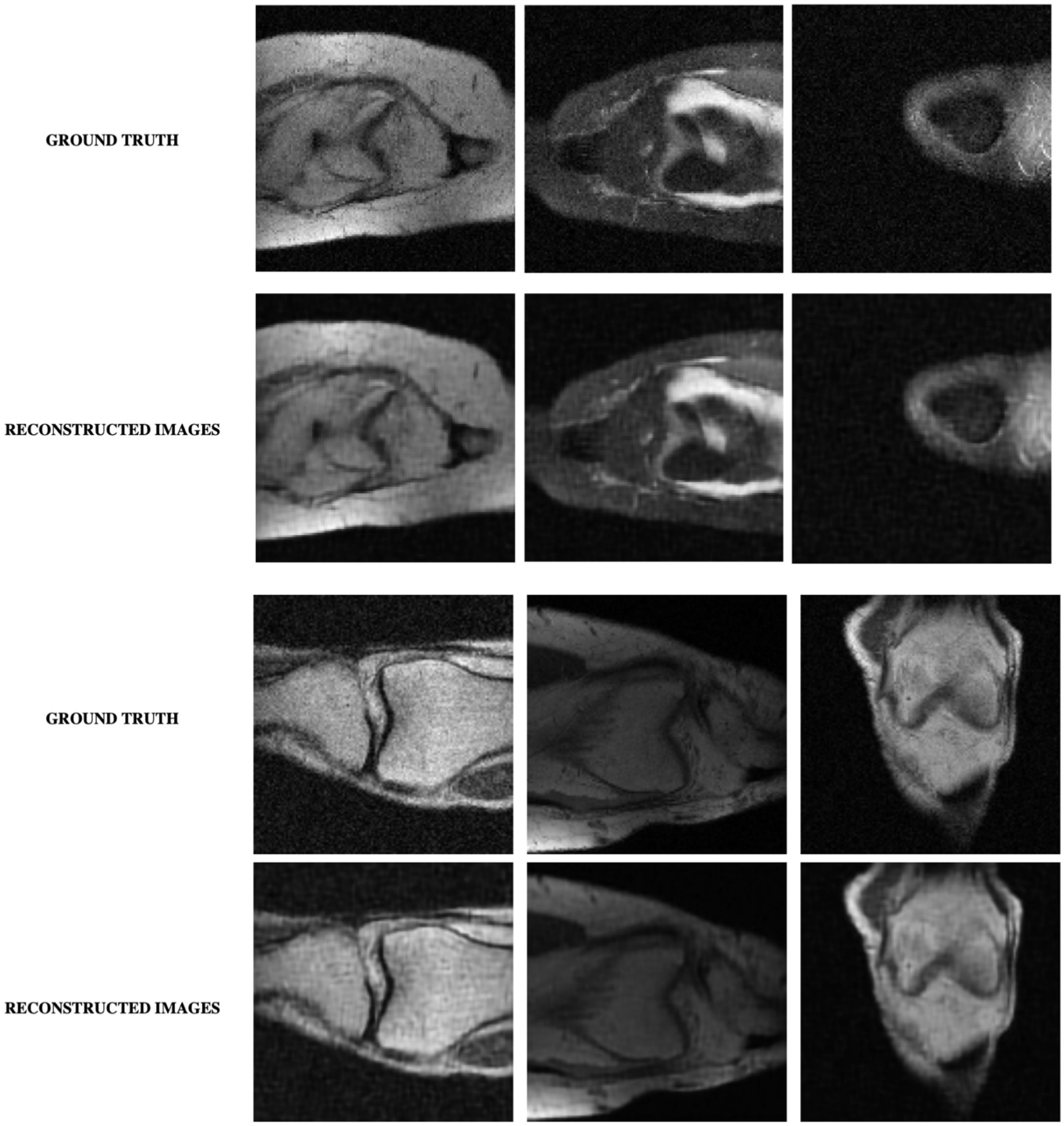


Figure 5: Our model's predictions versus ground truth on random test set images.

7 Conclusion/Future Work

Our model achieved a PSNR of 28.2 on validation data and 28.9 on training data outperforming the standard total variational baseline (PSNR 25.9) trained on the same dataset [5]. These results demonstrate the ability of our model to produce high quality image reconstructions. A qualitative analysis of our results indicates that the resolution of the reconstructed images can still be improved. Due to the high dimensionality of our inputs, we we cannot rely on FC layers in our architecture to improve resolution. Thus, we would need to experiment more with a fully convolutional architecture or adopt other architectures such as a modified U-net model or a GAN, though we would not want the model to hallucinate content in MRI data.

In the future, we hope to improve reconstruction quality by tweaking our model architecture. Moreover, we would like to test whether we can increase the generalizability of our model. Specifically, whether the weights of the trained model can be used via a transfer learning approach to quickly learn the reconstruction transforms for different classes of MRI scans (e.g., murine scans, human brains, etc.).

8 Contributions

All authors contributed equally to all parts of the project including model development, analysis and preparation of reports.

References

[1] Antun, Vegard, et al. "On instabilities of deep learning in image reconstruction-Does AI come at a cost?." *arXiv preprint arXiv*:1902.05300 (2019).

[2] Hyun, Chang Min, et al. "Deep learning for undersampled MRI reconstruction." *Physics in Medicine Biology* 63.13 (2018): 135007.

[3] Schlemper, Jo, et al. "A deep cascade of convolutional neural networks for MR image reconstruction." *International Conference on Information Processing in Medical Imaging*. Springer, Cham, 2017.

[4] Yang, Guang, et al. "DAGAN: deep de-aliasing generative adversarial networks for fast compressed sensing MRI reconstruction." *IEEE transactions on medical imaging* 37.6 (2017): 1310-1321.

[5] Zbontar, Jure, et al. "fastmri: An open dataset and benchmarks for accelerated mri." arXiv preprint arXiv:1811.08839 (2018).

[6] Zhu, B. et al. Image reconstruction by domain-transform manifold learning. *Nature* **555**, 487–492 (2018).