ESE 498 Capstone Design Project Proposal

Deep Learning for 3D Quantitative Brain MRI

Submitted to Professors Wang and the Department of Electrical and Systems Engineering

Date Submitted: June 1, 2023
Conducted: December 2022 – July 2023

Kerri Prinos
Department of Electrical and Systems Engineering
B.S./M.S. Electrical Engineering
k.prinos@wustl.edu

Advisors:

Dr. Ulugbek Kamilov
Computational Imaging Group
Department of Electrical and Systems Engineering, Department of Computer Science and Engineering
kamilov@wustl.edu

Yuyang Hu (Ph.D. candidate)
Computational Imaging Group
Department of Electrical and Systems Engineering

ABSTRACT

Deep learning algorithms, such as convolutional neural networks (CNNs), are powerful tools for accelerating brain magnetic resonance imaging (MRI) image reconstruction, greatly improving the imaging process for patients and healthcare workers. We aim to use the U-net architecture – a CNN architecture – to reconstruct: (1) a 2-D MRI brain image from a zero-filled image in the image domain, (2) a 2-D MRI brain image from a zero-filled image in k-space, and (3) a 3-D MRI brain image from a zero-filled image in k-space. For the 2-D reconstruction tasks, we will use the Brain Dataset from Aggarwal et al. [1-2], and for the 3-D reconstruction task, we will use the 3-D brain MRI images from the NYU Langone Health fastMRI initiative database [3-4]. We will assess the quality of the reconstructed images from each task using average PSNR and SSIM. Our preliminary study of the first task yielded average PSNR and SSIM values of 27.03 dB and 0.7562, respectively.

INTRODUCTION

Magnetic Resonance Imaging (MRI) is a valuable, non-invasive diagnostic tool in medicine. Brain MRI is advantageous over other imaging methods since it does not expose the patient to radiation, and it provides high-resolution images of the brain and surrounding tissues [5-7]. This is essential for monitoring and diagnosing diseases and disorders that affect the brain [5-7]. The downside to brain MRI is the lengthy acquisition time. A brain MRI can take anywhere from 30–90 minutes to complete, which makes MRI difficult to use in time-sensitive situations, as in the case of a stroke [7-8]. Motion artifacts from patient movement reduce image quality, so infants, patients with Parkinson's disease or claustrophobia often require general anesthesia for long scans, which poses its own risks [9-11]. While decreasing acquisition times improves patient comfort and reduces medical cost, it leads to severe degradation of image quality [10-11].

<u>Image Domain and k-space</u>

Raw MRI data is collected in k-space as an array of frequency- and phase-encoded spatial information [12]. Fig. 1 illustrates the relationship between the image domain and k-space. The image of the brain on the left is the result of the inverse Fourier transform of the k-space data. In the k-space representation on the right in Fig. 1, each pixel represents a spatial frequency [8,12]. The center of the k-space array contains the low spatial frequency components and the edges contain the high spatial frequency components [8,12].

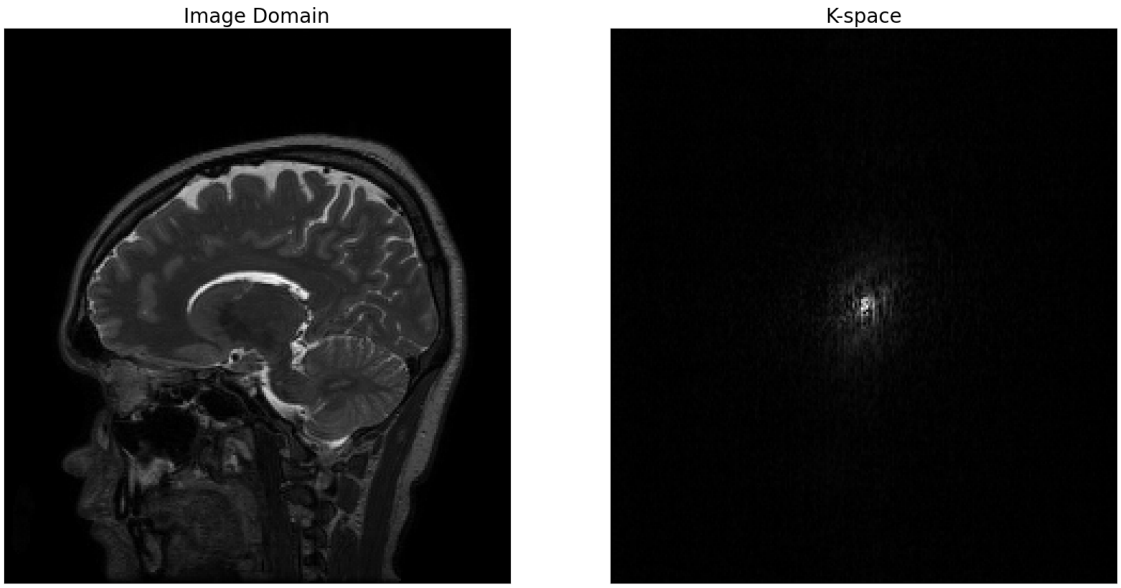

Figure 1. *Left:* A fully sampled brain MRI image in the image domain.
Right: The same image in k-space (the Fourier transform of the image domain).
Images obtained from the dataset [Agarwal et al.].

MRI acquisition can be accelerated by undersampling in k-space [8,13]. For example, by acquiring only one-sixth of the data points, you can achieve 6-fold acceleration [13]. The missing data points are filled in with 0's, a process known as zero-filling. Zhu et al. [14] found that zero-filling is always necessary in order to visualize structures – but not artifacts – in high resolution.

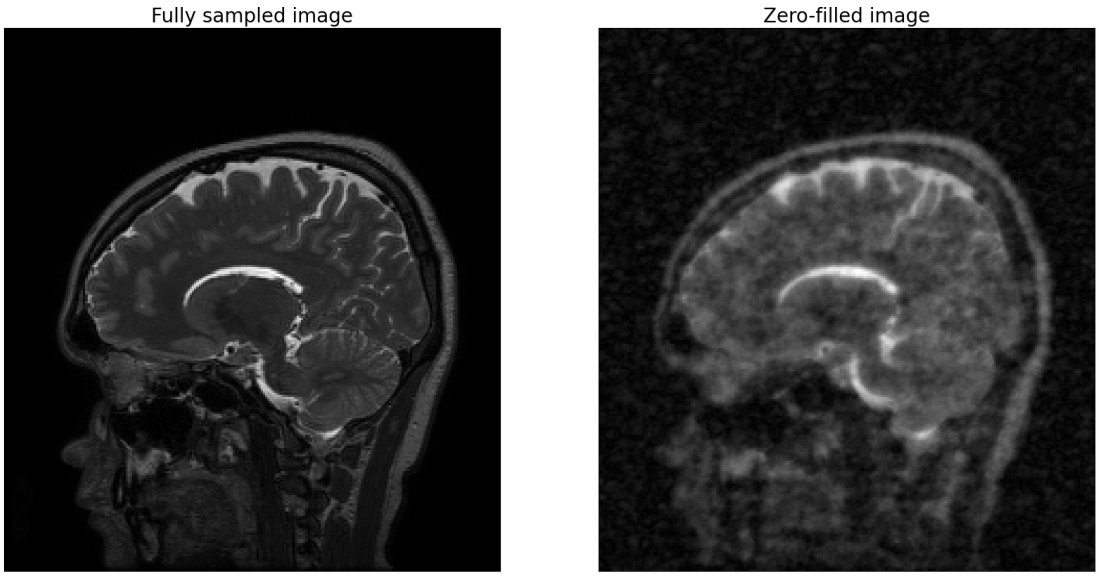

Figure 2. *Left:* A fully sampled brain MRI image
Right: A zero-filled image created using a random undersampling mask to do 6-fold acceleration. Images obtained from the dataset [1-2].

As seen in Fig. 2, zero-filled images are more distorted and contain more aliasing artifacts than fully-sampled images [8]. Previous works have demonstrated that deep-learning based methods, such as convolutional neural networks (CNNs), are powerful tools for reconstructing MRI images from zero-filled images in both the image domain [10] and directly in k-space [15].

Convolutional Neural Networks

CNNs are a type of deep learning algorithm that is used for image classification and object recognition tasks [16]. There is a feedforward process for feature extraction and classification and a backward path for training [16]. Fig. 3 illustrates typical CNN architecture for feedforward computation, which has three main types of layers: a convolutional layer, a pooling or subsampling layer, and a fully-connected layer [16].

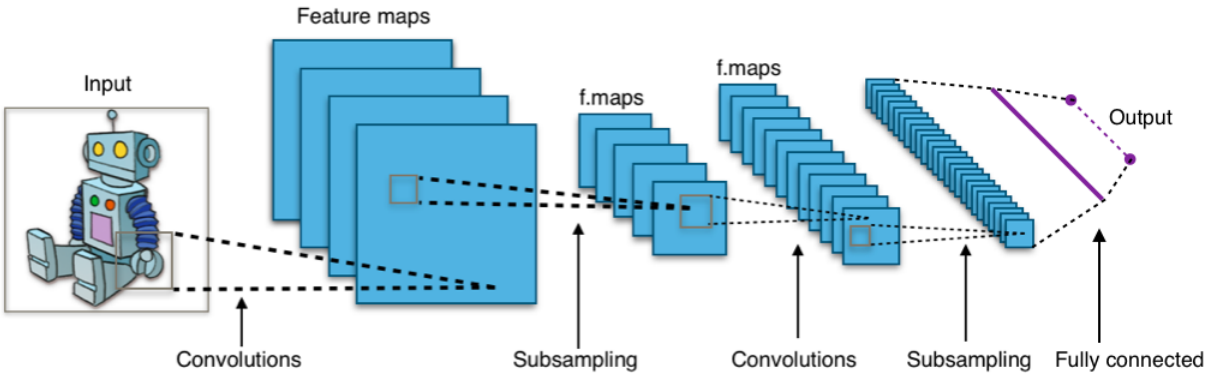

Figure 3. Overview of typical CNN architecture for feedforward computation [17]. Note the three main types of layers: a convolutional layer, a pooling or subsampling layer, and a fully connected layer. Feature maps are the result of the convolution of the input image and the feature detectors.

The feature maps are the result of the convolution of the input image and the feature detectors [16]. In Fig. 3, these feature detectors detect simple features, such as the outline of the robot or the different colors [16-17]. As the image progresses through the layers of CNN architectures, the layers become more complex [16-17]. For example, in Fig. 3, the network will recognize small groups of features, such as the edges and color of the robot's antenna or feet. Then it will identify larger elements, such as the head or torso, and eventually, it identifies the entire figure as a robot.

U-net Architecture for MRI image reconstruction

The U-net architecture is a convolutional network architecture that was developed by Ronneberger et al. [18] for biomedical segmentation applications. Ronneberger et al. [18] presented a U-net architecture consisting of two paths: a contracting path (left side) and an expansive path (right side), which gives it the characteristic U-shape as seen in Fig. 4 [18, Fig. 1].

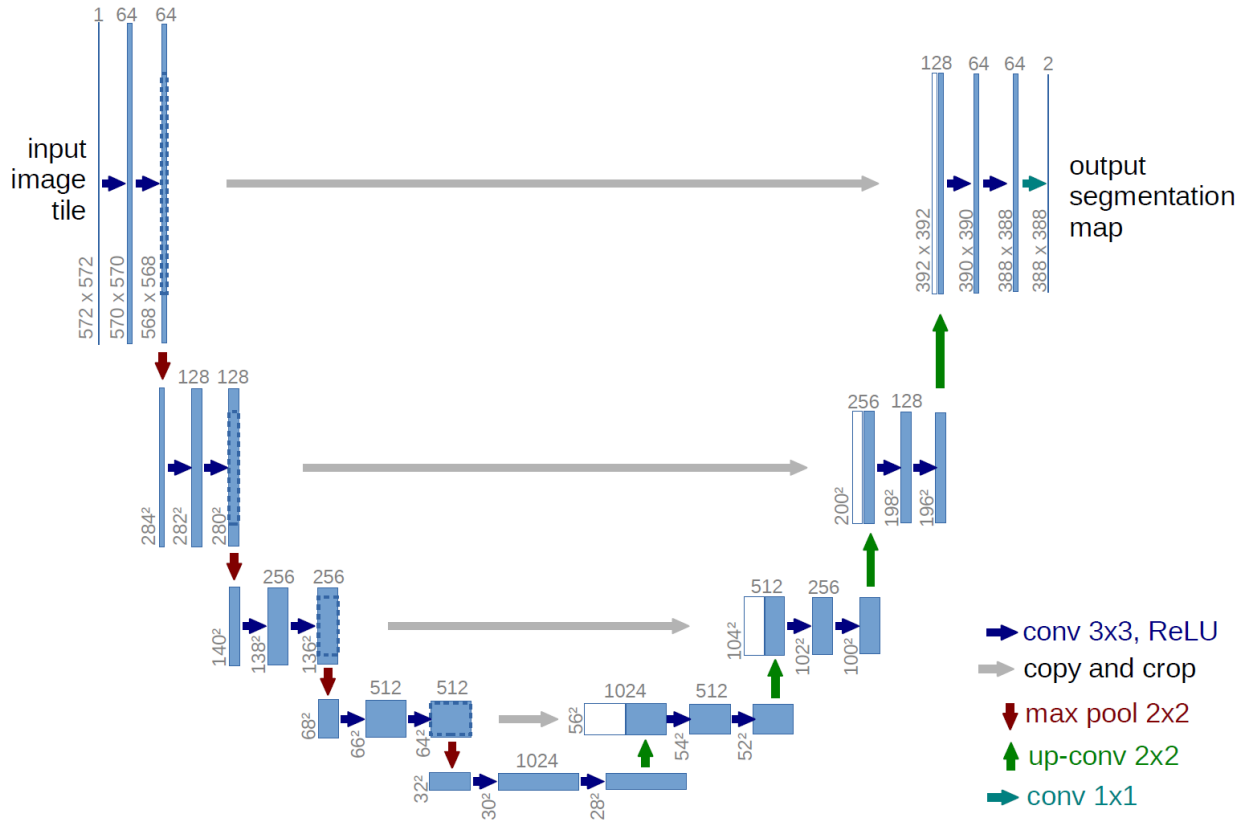

Figure 4. The original U-net architecture for biomedical image segmentation developed by Ronneberger et al. [18, Fig. 1]. Each blue box represents a multi-channel feature map. The number of channels are listed at the top of these boxes, and the x-y dimensions are listed to the side of the boxes. The two channels at the start and end are for the real and imaginary components of the data. The legend explains the color coded arrows that show convolution, max pool, and copy and crop operations.

The contracting path is similar to the example architecture of Fig. 3. Fig. 4 illustrates the sequence of the following operations: 3 x 3 unpadded convolutions, a rectified linear unit (ReLU) that helps the network to converge, and 2 x 2 max pooling operations with stride 2 for downsampling [18]. Ronneberger et al. [18] include a copy and crop operation after each set of two convolutions due to the loss of border pixels in every convolution. The final layer uses a 1x1 convolution to map the 64-component feature vector to the number of desired segmentation classes [18].

Jin et al. [10] first adapted the U-net architecture to perform image reconstruction tasks. Their modifications include the use of zero-padding to prevent the image size from decreasing with each convolution and replacing the last layer with a convolutional layer that reduces the 64-component feature vector to one output image [10]. Jin et al. [10] also replace the copy and crop operation with a copy and concatenate operation to improve model convergence in backpropagation [19]. The U-net model is effective for image reconstruction because the many

convolutions in both paths result in high-level local and global feature recognition [10,13,20]. Since its introduction for MRI image reconstruction, many researchers have achieved better results by modifying the basic U-net architecture [1, 4, 8-11,13,19-24].

3-D MRI image reconstruction

The next goal for researchers is using deep learning on 3-D MRI images. Sandino et al. [19] extend the work on Jin et al. using the U-net architecture for 3-D cardiac MRI images. Instead of directly inputting the 3-D data into the U-net model, Sandino et al. [19] separated the 3-D volume into a series of 2-D slices to be input into the model.

We aim to extend the work of Sandino et al. [19] and use deep learning to perform direct k-space interpolation for 3-D MRI images of the brain. That is, we want to perform deep learning on the 3-D image without breaking it into 2-D slices as done in previous works. To achieve this goal, we'll sequentially use deep learning to reconstruct: (1) a 2-D MRI brain image from a zero-filled image in the image domain, (2) a 2-D MRI brain image from a zero-filled image in k-space, and (3) a 3-D MRI brain image from a zero-filled image in k-space. Du et al. theorized that training CNNs in k-space to interpolate the missing k-space data points will achieve better performance than training in the image domain [15, 22]. This drives our ultimate goal of achieving 3-D image reconstruction directly in k-space. We will be using the basic U-net architecture for our three image reconstruction tasks, as it provides a baseline for comparing our results to those in existing literature.

PROPOSED TECHNICAL APPROACHES

MRI reconstruction can be described by an inverse problem. In (1), we define the inverse problem that recovers a ground truth image $\mathbf{x} \in \mathbb{C}^n$ from a noisy measurement $\mathbf{y} \in \mathbb{C}^n$ characterized by the linear model:

$$\mathbf{y} = \mathbf{PFx} + \mathbf{e}, \qquad (1)$$

where $\mathbf{F} \in \mathbb{C}^{n \times n}$ denotes the Fourier transform, $\mathbf{P} \in \mathbb{F}^{n \times n}$ is a sampling operator, and $\mathbf{e} \in \mathbb{C}^{n \times n}$ is a noise vector. Both arrays are random undersampling masks to do 6-fold acceleration. The zero-filled images are given by:

$$\hat{\mathbf{x}}_i = \{\mathbf{F}^{-1}\mathbf{y}_i\}_i^N \qquad (2)$$

where $\mathbf{F}^{-1}$ denotes the inverse Fourier transform and N is the total number of training samples.

2-D Reconstruction

Two MRI datasets will be used in this project. For 2-D reconstruction, we use the Brain Dataset from Aggarwal et al. [1-2]. The parallel MRI brain data of five human subjects was collected using 3D T2 CUBE sequences with Cartesian readouts using a 12-channel head coil. The dataset includes an array called tnrOrg, which contains the fully sampled MRI data in the image domain from four test subjects. trnOrg is a complex array with dimensions 256 x 232 x 360. There are 90 slices from each of the four subjects, and each slice is of spatial dimension 256 x 232.

trnOrg will serve as our ground truth $\{\mathbf{x}\}_i^N$ in the image domain. For our ground truth in k-space, we will use the Fourier transform of the trnOrg. We also use trnMask from the dataset, which is a random undersampling mask to do 6-fold acceleration. We use (1) to find the noisy measurement with trnMask as $\mathbf{P}$. $\mathbf{y}$ is our zero-filled image in k-space. We apply (2) to find the zero-filled images in the image domain. We will train the CNN f_θ by mapping zero-filled images (2) to their desired ground truth $\{\mathbf{x}\}_i^N$ in both the image domain and k-space. We will use a 80-10-10% split for the training, testing, and validation datasets.

All our code will be written in Python 3.10.11 on Google Colab. We will use the NVIDIA T4 TensorCore GPUs available on Google Colab for training and testing our model. Since we have chosen to use Pytorch version 2.0.1 optimized with CUDA version 11.8, we can easily use the PyTorch implementation of a U-net model from Zbontar et al. [18, 21]. We will instantiate the U-net model as follows:

```
# setting up unet model, loss function, and optimizer
model = Unet(in_chans=2,
       out_chans=2,
       chans = 32,
       num_pool_layers = 4,
       drop_prob  = 0.0)
```

We will use two channels to handle the real and imaginary components of the MRI data as in [24], and a batch size of 16. Figure 5 provides a detailed, visual representation of the U-net architecture.

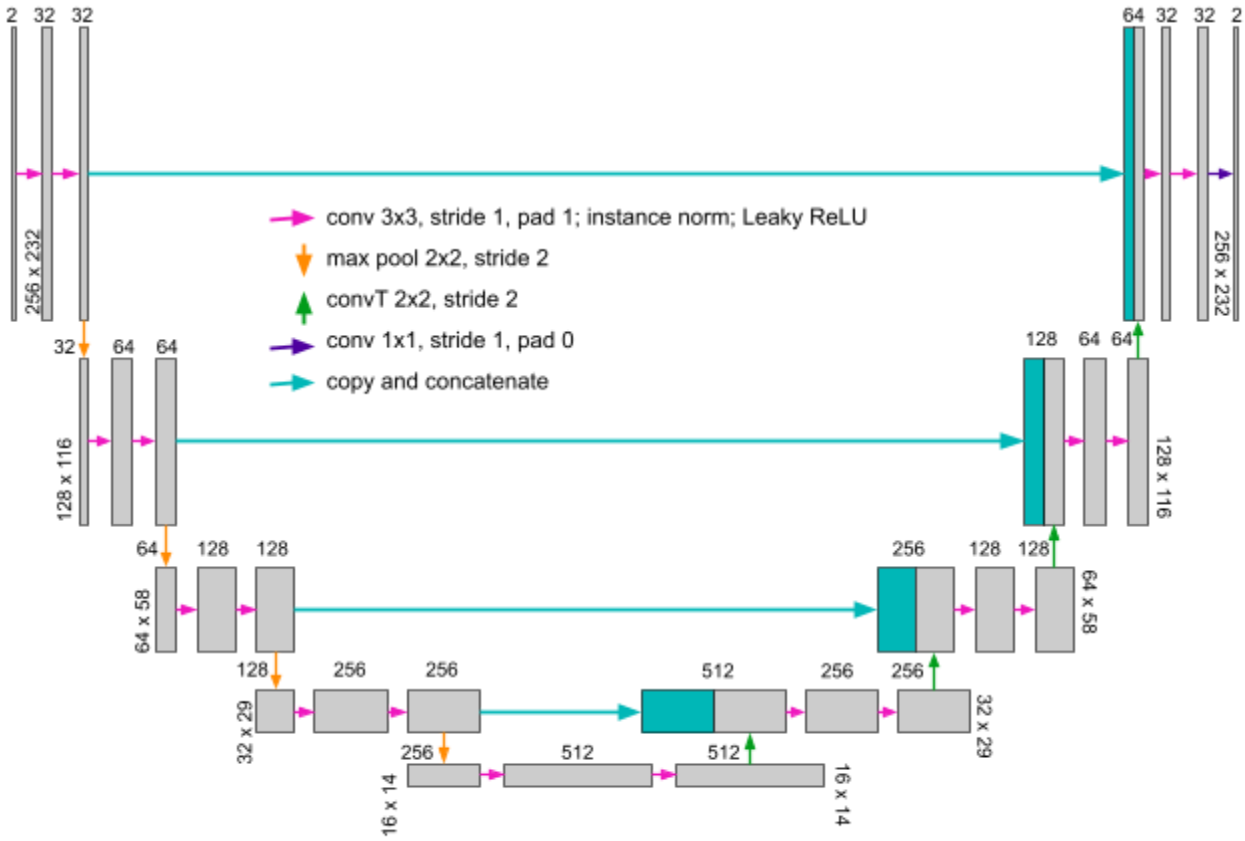

Figure 5. U-net architecture for 2-D reconstruction. The gray boxes represent multi-channel feature maps and the teal boxes represent the copied feature maps. The number of channels are listed at the top of these boxes, and the x-y dimensions are listed to the side of the boxes. The two channels at the start and end are for the real and imaginary components of the data. The legend explains the color coded arrows that show convolution, max pool, and copy and concatenate operations. This figure was adapted from: [10 Fig. 2; 18, Fig. 1; 21, Fig. 7;].

This U-net model has 7,756,418 parameters, which are all trainable parameters. The network will be trained for 200 epochs with Adam optimization, a learning rate of 0.0001, and the mean squared error (MSE) loss function defined in (3).

$$\frac{1}{N}\sum_{i}^{N}\|f_\theta(\hat{\mathbf{x}}_i)-\mathbf{x}_i\|_2^2 \quad . \qquad (3)$$

3-D Reconstruction

To our knowledge, we are the first to attempt to directly reconstruct a 3-D MRI image in k-space. That is, train a CNN using volumetric MRI data, rather than using 2-D slices of volumetric MRI data [19]. It is important that we are able to achieve results comparable to those

in literature for the 2-D reconstruction tasks in the image domain and k-space. Only then will we move on to the 3-D image reconstruction task.

We will be using the 3-D brain MRI images from the NYU Langone Health fastMRI initiative database for this task [3-4]. The NYU fastMRI investigators provided this data, to accelerate the goal of testing whether machine learning can aid medical image reconstruction [3-4]. The brain MRI contains 6,970 fully sampled brain MRIs, which were obtained using magnets with field strengths of 1.5 T and 3 T. The raw dataset includes axial images with T1, T2, and FLAIR pulse sequences. The investigators noted that the T1 weighted images include images with and without a contrast agent.

Although we will not discuss in depth the differences between these pulse sequences, it is worth noting that each pulse sequence causes certain features to be accentuated in the image. This is essential for accurate medical diagnosis. For example, in T1-weighted images, fatty tissue will appear bright, whereas in T2-weighted images, fatty tissue and water will appear bright [25]. Cerebrospinal fluid, a water-based fluid, will appear bright in T2-weighted images, and dark on T1-weighted images [25]. In FLAIR images, the cerebrospinal fluid will appear dark, but any abnormalities in the cerebrospinal fluid will remain bright [26]. Contrast agents are also useful for getting a clearer image of certain features of the brain.

The inverse problem posed in (1) is also valid for 3-D image reconstruction in k-space [9]. One potential approach is to directly change our 2-D U-net architecture to a 3-D U-net architecture. The operations in the 2-D U-net architecture in Fig. 5 would be changed from their 2-D to their 3-D versions, ex. 3-D instead of 2-D convolutions. Then, we would input the 3-D images in k-space, splitting real and imaginary components into two channels as before, and obtain the 3-D reconstructions also in k-space. The z-component of the data presents the biggest challenge in this task. Previous works have avoided the added complication the z-component presents by inputting 2-D slices of the 3-D images [19]. We will need to be careful in how we incorporate the z-component.

DATA TO BE COLLECTED AND PROCEDURE FOR ACQUIRING

For all three reconstruction tasks, we will be evaluating the performance using qualitative and quantitative metrics. An arbitrary image will be selected from ground truth from the testing set for a side by side comparison with its zero-filled image and reconstructed image. We will use the average peak signal-to-noise ratio (PSNR) and the average structural similarity index measure (SSIM) of the testing set to quantitatively assess the quality of the reconstructed image. SSIM and PSNR are widely used in MRI reconstruction research, so it will be beneficial to compare our values to those in literature.

We use the psnr() and ssim() functions from the metrics model from scikit-image, which is an open-source library for image processing in Python, to calculate the average PSNR and SSIM for the reconstructed test images. As the name suggests, the PSNR is the ratio of the maximum signal in the ground truth image to the mean-square error between the reconstructed and ground truth images. The PSNR in decibels can be calculated as in (4).

$$\mathrm{PSNR}(x, \hat{x}) = 10 \log_{10} \left(\frac{\mathrm{MAX}^2}{\frac{1}{N} \sum_{i=1}^{N} (x_i - \hat{x}_i)^2} \right), \quad (4)$$

where x is the ground truth image, $\hat{x}$ is the reconstructed image, MAX represents the maximum signal in the ground truth image, and N is the total number of pixels in the ground truth image. A higher PSNR value of 30-50 dB indicates better image reconstruction. Wang et al. developed the SSIM index to complement simpler metrics such as PSNR [27]. SSIM "compares local patterns of pixel intensities that have been normalized for luminance and contrast," so it better aligns with the way humans perceive image quality. The Structural Similarity Index (SSIM) between x and $\hat{x}$ is given by (5):

$$\mathrm{SSIM}(x, \hat{x}) = \frac{2\mu_x \mu_{\hat{x}} + C_1}{\mu_x^2 + \mu_{\hat{x}}^2 + C_1} \cdot \frac{2\sigma_{x\hat{x}} + C_2}{\sigma_x^2 + \sigma_{\hat{x}}^2 + C_2} \cdot \frac{\sigma_{x\hat{x}}}{\sigma_x \sigma_{\hat{x}} + C_3}, \quad (5)$$

where μ_x and $\mu_{\hat{x}}$ represent the means of x and $\hat{x}$, respectively. σ_x and $\sigma_{\hat{x}}$ denote the standard deviations, and $\sigma_{x\hat{x}}$ represents the covariance between x and $\hat{x}$. C_1, C_2, and C_3 are small constants for numerical stability. SSIM values range from 0 to 1. We want an SSIM value close to 1, which indicates that the reconstructed image is very similar to the ground truth image.

FEASIBILITY STUDY WITH PRELIMINARY RESULTS

The first task serves as an effective feasibility study. We use the PyTorch implementation of a U-net model from Zbontar et al. [21] to map the 2-D zero-filled MRI images in the image domain to their desired ground truth images, also in the image domain [18, 21]. The model is trained over 200 epochs using the MSE loss function and the Adam optimization algorithm. The duration of the training process was approximately 25 minutes. Figure 6 shows a qualitative comparison of a randomly selected ground truth image from the testing set and its corresponding zero-filled image and reconstructed image. Note, the brain tissue can be seen in more detail in the reconstructed image compared to the zero-filled image. This observation is reflected in the PSNR and SSIM values, as well.

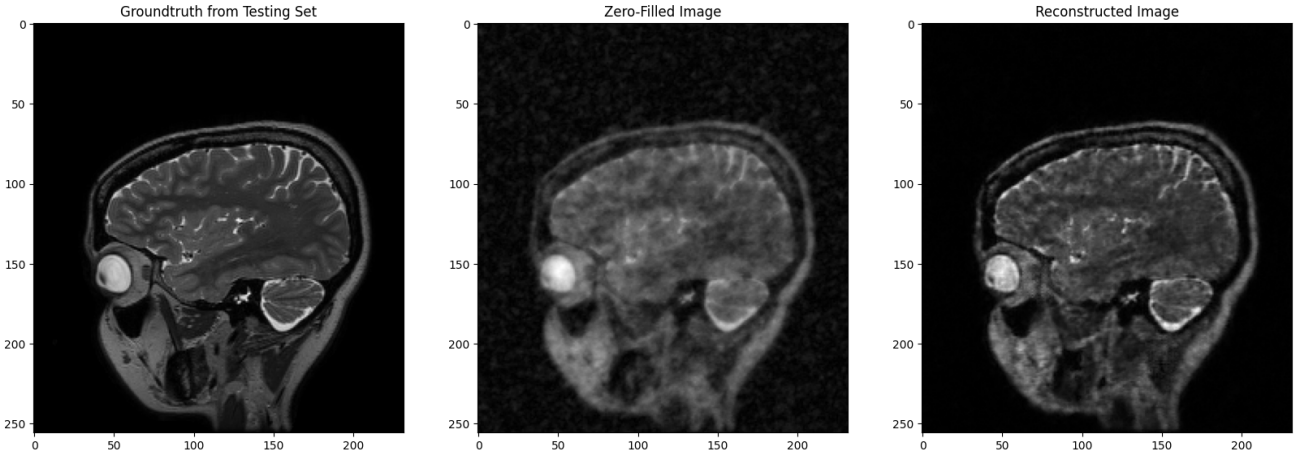

Figure 6. Side by side comparisons of the ground truth image from the testing set and its corresponding zero-filled image and reconstructed image.

Quantitatively comparing a randomly selected reconstructed image to its corresponding ground truth image, we obtain an average PSNR value of 27.03 dB and an average SSIM value of 0.7562. Both these values show improvement over the average PSNR value of 24.77 dB and the average SSIM value of 0.4860 for the corresponding zero-filled image compared to the ground truth image. The PSNR and SSIM values for the reconstructed image are comparable to those in the literature for reconstruction using basic U-Net models [8,9,11,13,21].

ANTICIPATED RESULTS

Our final results will be similar to those of the feasibility study. We will have a side by side visual comparison of the ground truth, zero-filled image, and the reconstructed image for each reconstruction task. The average PSNR and SSIM will provide a quantitative measure of the quality of our reconstructed images for each task.

Completing the second task will be a great achievement, as 2-D reconstruction in k-space has not been completed by our lab previously. We will be able to compare our results against previous works [1, 4, 8-11,13,20-24].

We are hoping to complete, or at the very least get preliminary results for the first task. Perhaps, we will only be able to comment on what methods did not work or the quality of the reconstruction will not be where we want it. We will be able to compare our results to Sandino et al., who performed 3-D reconstruction in k-space, using 2-D slices of volumetric data [19].

DELIVERABLES

The deliverables of this project include:

- The **Python code** for training the convolutional neural network used to complete each of the three tasks.
- The **average PSNR and SSIM values**, which provide a quantitative measure of success for each of the three tasks.

- The **reconstructed brain MRI images**, which provide a qualitative measure of success for each of the three tasks.
- A **web page** explaining the project and presenting the above deliverables.
- A **final report and powerpoint presentation** providing an in-depth explanation of the project, the methods, and the results achieved for each task.

TIMELINE FOR PROJECT COMPLETION

The Gantt chart (Fig. 7) illustrates the project schedule, including deadlines for the steps required to complete the deliverables. Table 1 details each task.

Table 1. Tasks to be completed and project schedule

	Task	Duration (days)	Start	Finish	Predecessors
1	Formal proposal	NA*	1/31/23	6/1/23	
2	Improving & fixing errors in task 2 code	18	6/1/23	6/18/23	
3	Update report with task 2 results	3	6/19/23	6/21/23	2
4	Research and testing for task 3	18	6/22/23	7/9/23	2,3
5	Improving & fixing errors in task 3 code	6	7/10/23	7/15/23	4
6	Update report with task 3 results	3	7/16/23	7/18/23	5
7	Final report draft	4	7/19/23	7/23/23	6
8	Final report update based on feedback	7	7/24/23	7/30/23	7
9	Final presentation	7	7/24/23	7/30/23	7
10	Final webpage	7	7/24/23	7/30/23	7

* progress interrupted due to medical condition

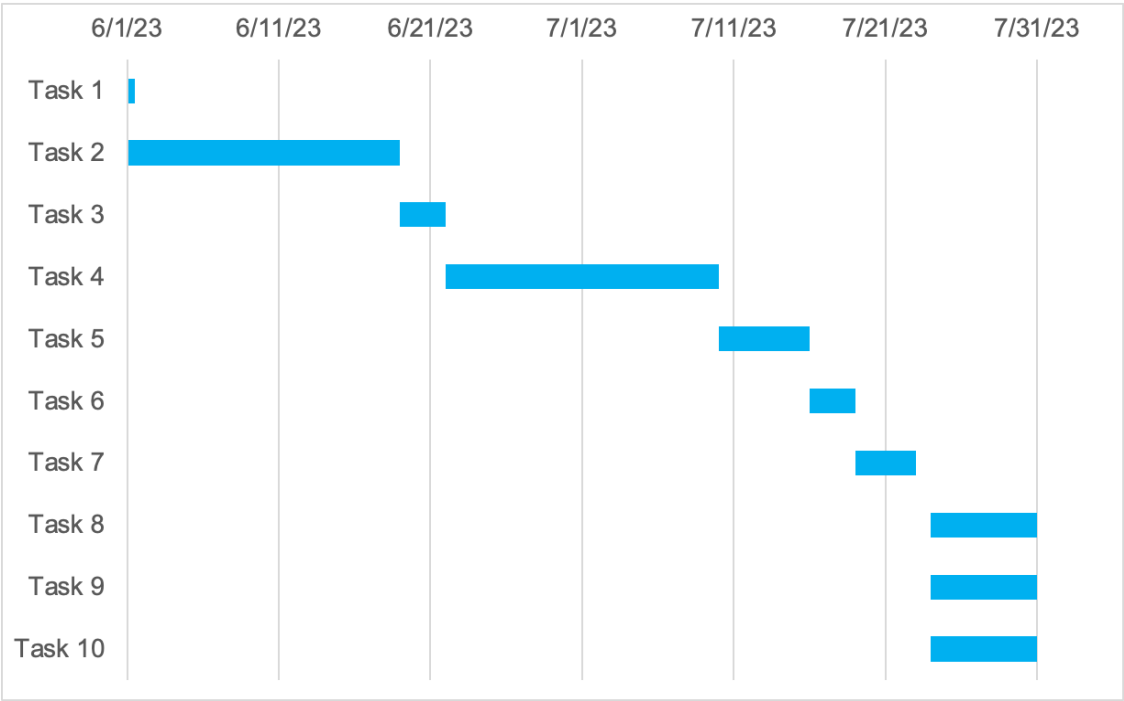

Figure 7. Gantt chart for anticipated timeline for project completion.

REFERENCES

[1] H.K. Aggarwal, M.P. Mani, and M. Jacob, "MoDL: Model Based Deep Learning Architecture for Inverse Problems," 2019. [Online serial]. Available: https://arxiv.org/abs/1712.02862. [Accessed Feb. 7, 2023].

[2] H.K. Aggarwal, M.P. Mani, and M. Jacob, "Brain Dataset," *Github: hkaggarwal/modl*, 2019. Available: https://drive.google.com/file/d/1qp-l9kJbRfQU1W5wCjOQZi7I3T6jwA37/view. [Accessed Dec. 2, 2022].

[3] F. Knoll, P.M. Johnson, D.K. Sodickson, M.P. Recht, and Y.W. Lui, "NYU fastMRI Dataset: Brain MRI," *NYU Langone Health: fastMRI Dataset*, 2020. Available: https://fastmri.med.nyu.edu. [Accessed May 24, 2023].

[4] F. Knoll et al., "fastMRI: A Publically Available Raw k-space and DICOM Dataset of Knee Images for Accelerated MR Image Reconstruction Using Machine Learning," *Radiology: Artificial Intelligence*, vol. 2, no. 1, 2020. [Online]. Available: RSNA, https://www.rsna.org. [Accessed May 24, 2023].

[5] Mayo Clinic Staff, "MRI." Mayo Clinic. https://www.mayoclinic.org/tests-procedures/mri/about/pac-20384768. [Accessed May 30, 2023].

[6] Mount Sinai, "Head MRI." Mount Sinai. https://www.mountsinai.org/health-library/tests/head-mri. [Accessed May 30, 2023].

[7] Cleveland Clinic, “Brain MRI.” Cleveland Clinic. https://my.clevelandclinic.org/health/diagnostics/22966-brain-mri. [Accessed May 30, 2023].

[8] M.B. Hossain, K.-C Kwon, R.K. Shinde, S.M. Imtiaz, and N. Kim, “A Hybrid Residual Attention Convolutional Neural Network for Compressed Sensing Magnetic Resonance Imaging Reconstruction,” *Diagnostics*, vol. 13, no. 7, pp. 1306, 2023. [Online]. Available: MDPI, https://www.mdpi.com. [Accessed May 15, 2023].

[9] Z. Ramzi, P. Ciuciu, and J. Starck, “Benchmarking MRI Reconstruction Neural Networks on Large Public Datasets, *Appl Sci*, vol. 10, no. 2, pp.1816, 2020. [Online]. Available: MDPI, https://www.mdpi.com. [Accessed May 15, 2023].

[10] K.H. Jin, M.T. McCann, E. Froustey, and M. Unser, “Deep Convolutional Neural Network for Inverse Problems in Imaging,” *IEEE Transactions on Image Processing*, vol. 26, no.9, pp. 4509-4522, 2017. [Online]. Available IEEE Xplore, https://ieeexplore.ieee.org/Xplore/home.jsp. [Accessed May 14, 2023].

[11] C.M. Hyun, H.P. Kim, S.M. Lee, S. Lee, and J.K. Seo, “Deep learning for undersampled MRI reconstruction,” *Physics in Medicine and Biology*, vol. 63, no.13, pp. 135007, 2018. [Online]. Available: IOPScience, https://iopscience.iop.org. [Accessed May 14, 2023].

[12] T.A. Gallagher, A.J. Nemeth, and L. Hacein-Bey, “An Introduction to the Fourier Transform: Relationship to MRI,” *American Journal of Roentgenology*, vol. 190, no. 5, 2008. [Online]. Available: American Journal of Roentgenology, https://www.ajronline.org. [Accessed May 30, 2023].

[13] T. Rahman, “Fast Magnetic Resonance Image Reconstruction With Deep Learning Using An Efficientnet Encoder,” 2021. [Online serial]. Available: https://scholarworks.utep.edu/cgi/viewcontent.cgi?article=4323&context=open_etd. [Accessed Feb. 7, 2023].

[14] X. Zhu, B. Tomanek, and J. Sharp, "A Pixel is an Artifact: On the Necessity of Zero-Filling in Fourier Imaging," *Concepts Magn Reson*, vol. 42A, no. 2, pp. 32-44, 2013. [Online]. Available: Wiley Online Library, https://onlinelibrary.wiley.com. [Accessed Feb. 2, 2023].

[15] T. Eo, Y. Jun, T. Kim, J. Jang, H. Lee, and D. Hwang, “KIKI-net: cross-domain convolutional neural networks for reconstructing undersampled magnetic resonance images,” *Magn Reson Med,* vol. 80, pp. 2188-2201, 2018. [Online]. Available Wiley Online Library, https://onlinelibrary.wiley.com. [Accessed Jan. 30, 2023].

[16] IBM, “Convolutional Neural Networks.” IBM. https://www.ibm.com/topics/convolutional-neural-networks. [Accessed June 1, 2023].

[17] Aphex34, “typical CNN architecture,” Wikimedia Commons. 2015. https://commons.wikimedia.org/wiki/File:Typical_cnn.png. [Accessed June 1, 2023].

[18] O. Ronneberger, P. Fischer, and T. Brox, “U-Net Convolutional Networks for Biomedical Segmentation,” 2015. [Online serial]. Available: https://arxiv.org/abs/1505.04597. [Accessed Feb. 7, 2023].

[19] C.M. Sandino, N. Dixit, J.Y. Cheng, and S.S. Vasanawala, “Deep convolutional neural networks for accelerated dynamic magnetic resonance imaging,” 2017. [Online serial]. Available: http://cs231n.stanford.edu/reports/2017/pdfs/513.pdf. [Accessed May 14, 2023].

[20] V. Ghodarti et al., “MR image reconstruction using deep learning: evaluation of network structure and loss functions,” *Quant Imaging Med Surg*, vol. 9, no. 9, pp.1516-1527, 2019. [Online]. Available: QIMS, https://qims.amegroups.com. [Accessed May 14, 2023].

[21] J. Zbontar et al., “fastMRI: An Open Dataset and Benchmarks for Accelerated MRI,” 2018. [Online serial]. Available: https://arxiv.org/abs/1811.08839. [Accessed May 15, 2023].

[22] T. Du, H. Zhang, Y. Li, H.K. Song, and Y. Fan, “Adaptive convolutional neural networks for k-space data interpolation in fast magnetic resonance imaging,” 2020. [Online serial]. Available: https://arxiv.org/abs/2006.01385. [Accessed Jan. 30, 2023].

[23] L. Xu, J. Xu, Q. Zheng, J. Yuan, and J. Liu, “A miniature U-net for *k*-space-based parallel magnetic resonance imaging reconstruction with a mixed loss function,” *Quant Imaging Med Surg,* vol. 12, no. 9, pp. 4390-4401, 2022. [Online]. Available PubMed Central, https://www.ncbi.nlm.nih.gov/pmc/. [Accessed Feb. 7, 2023].

[24] Y. Han, L. Sunwoo, and J.C. Ye, “*k*-Space Deep Learning for Accelerated MRI,” *IEEE Transactions on Medical Imaging*, vol. 39, no. 2, pp. 377-386, 2020. [Online]. Available: IEEE Xplore, https://ieeexplore.ieee.org/Xplore/home.jsp. [Accessed Feb. 7, 2023].

[25] D.C. Preston, “Magnetic Resonance Imaging (MRI) of the Brain and Spine: Basics.” Case Western Reserve University: Department of Neurology. 2016. https://case.edu/med/neurology/NR/MRI%20Basics.htm. [Accessed May 30, 2023].

[26] R. Bakshi, et al., “Fluid-Attenuated Inversion-Recovery MR Imaging in Acute and Subacute Cerebral Intraventricular Hemorrhage,” *American Journal of Neuroradiology*, vol. 20, no. 4, 1999. [Online]. Available: NCBI-NIH, https://www.ncbi.nlm.nih.gov. [Accessed May 29, 2023].

[27] Z. Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli, “Image quality assessment: from error visibility to structural similarity,” *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600-612, 2004. [Online]. Available: IEEE Xplore, https://ieeexplore.ieee.org/Xplore/home.jsp. [Accessed May 27, 2023].