

An Investigation of Deep Learning Models for Image Enhancement in MRI

Evan M. R. Petrimoulx

April 5, 2025

Abstract

Since its invention, MRI has been an essential tool in imaging samples and diagnosing patients. While the results from an MRI scan are crucial in providing doctors and researchers clear images of **their** sample, it can take hours to gather the necessary k-space data to produce useful images. A rising technique to help combat the large scan times and image processing times is to implement deep learning models. These neural networks are capable of learning from the data in the frequency domain across multiple training scans and predict the images produced from new samples given a restricted k-space domain. Reducing k-space sampling shortens scan times but often degrades image quality, introducing aliasing and contrast loss. Teaching a neural network the trends and patterns present in the frequency domain for a sample can help generate images with close to the same image quality as that produced with a full frequency spectrum scan. Correctly training these models allows for the reconstruction of high-quality images from undersampled scans. Convolutional Neural Networks (CNNs), a common feedforward model used in image processing, are particularly effective for MRI enhancement and reconstruction, and are the standard model used in the field. This report investigates the different techniques and implementation strategies of these models in MRI, and discusses the advantages as well as the limitations of these models. Furthermore, this report provides examples of neural networks which showcase the capability of image reconstruction, as well as the issues present with them such as overfitting, which can cause hallucinations that are present in the final generated image and must be avoided.

Introduction

Magnetic Resonance Imaging (MRI) is an imaging technique that utilizes the resonant frequency of atoms, typically hydrogen, as well as strong external gradient magnetic fields to produce image slices of a subject. The process of recovering an image given a measurement of frequency represented as a signal is done by performing the inverse Fourier Transform to the measured signal.

$$\rho(x) = \mathcal{F}^{-1}\{S(k)\} = \int_{-\infty}^{\infty} S(k)e^{i2\pi kx} dk . \quad (1)$$

The problem lies within the collection of signal data in the frequency domain. If the entirety of the frequency domain is not sampled, then key data pertaining to the image reconstruction is lost. This typically leads to loss of contrast in the image, as well as decreased image sharpness, and can cause aliasing.

MRI relies on the relaxation time of isochromats tending back to equilibrium after being aligned with the external field. Because of this, rapid data acquisition is constrained by T_1 , T_2 , and T_2^* signal decay times. As a result, obtaining a full k-space dataset requires significant time, causing a negative relationship between image clarity and speed. Finding techniques that allow for less frequency domain to be required **would therefore** decrease scanning times, and maintain image quality. This would allow for hospitals to see more patients more frequently at a cheaper cost of diagnosis. One of these methods is the implementation of **neural networks**. Neural networks are a category of machine learning that is modelled after the human

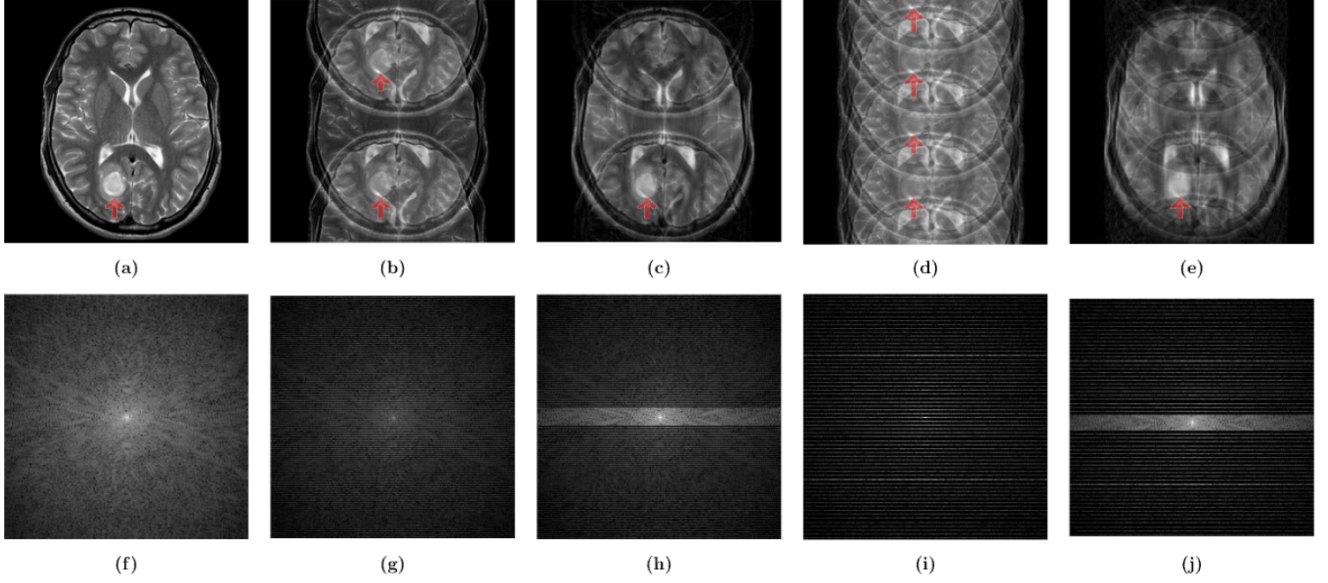


Figure 1: Image of a human brain from [1]. As the frequency domain data collection is reduced, the image quality becomes worse and aliasing occurs.

brain [2]. They are capable of complex pattern recognition and extrapolation which can be used to successfully construct a high quality image given a reduced amount of k-space data [3]. Successfully training a neural network to identify patterns in k-space data allows researchers to spend less time performing data acquisition in the frequency domain, while still retaining quality in the image domain.

Theory

In order to understand how neural networks work, it is helpful to start with a simpler model. Machine learning as a whole can roughly be separated into two categories, **regression** and **classification**. Classification problems deal with the identification of discrete data. This is common for solving problems such as image identification. Classification algorithms can identify things like handwriting, or

parts of the human body given an image. Classification is used to model discrete data. Regression is the opposite. Regression models in machine learning deal with identification of trends or continuous data. Regression models, sometimes referred to as regressors, are capable of predicting trends such as in price changes in the stock market, or rate of spread of disease. Both models require large amounts of data for training. In order for these models to learn, they must be given data involving many features (variables) and a known output. When the output of the function being analyzed is known, the development of the model is called supervised learning. These models then analyze the trends via the relationships between the features presented in the dataset, and find a complex equation that it deems worthy of accurately representing the data it was presented. For the problem of image enhancement in MRI, regressors are the correct tool for the problem since a continuous amount of data from the frequency domain is needed to reconstruct an image.

In the event that the data presented to the model is more complex than a linear relationship, the model will have a difficult time predicting an output[4]. To solve this issue and allow for the analysis of higher order correlations in data, the features can be processed through intermediate relationships called layers. For a single layer, the problem is linear and any simple regressor can predict an output. When there is more than one layer present in the model, the machine learning algorithm is referred to as a deep learning model. Neural Networks are a form of deep learning model where each feature in the data set is connected to other features in a web across multiple layers. The mathematical relationship between

a single neuron (a node within the layer of the neural network) in the network is given by the following equation [5]:

$$y = f \left(\sum_{i=1}^n w_i x_i + b \right) . \quad (2)$$

Where f is the activation function which is dependant on the weighting of each linear feature for each neuron w_i , the feature or variable itself x_i , and the bias term b , which tells the algorithm how important a value is to the pattern it is finding and is common to all weighting parameters. The summation sums over all of the features that are interconnected in the layer. Training a neural network requires adjusting the weights and biases of the nodes (neurons) to minimize a loss function via an optimization algorithm such as gradient descent.

A loss function is a function that takes two inputs and computes how similar they are to one another. The more similar the two values are, the more the model is able to predict given the trend it has discovered. The average loss over the entire dataset can tell us how correct our model is. It is given by the following formula:

$$\langle L_{\text{total}}(y, \hat{y}) \rangle = \frac{1}{N} \sum_{i=1}^N L(y, \hat{y}) . \quad (3)$$

Where there are N variables or features in the dataset and $L(y, \hat{y})$ is the loss function, y is our prediction value, and \hat{y} is the known correct value [6]. The loss function and its minimization are what makes the neural network capable of

learning nonlinear trends. This is one of the advantages of deep learning models compared to linear regression models such as ridge regression. There are many different possible loss functions that are dependant on the statistics of the machine learning model being used, but a common example is the least squares loss function. This has the typical definition for least squares fitting [6]

$$f_{\text{lsqf}} = (y - \hat{y})^2 . \quad (4)$$

The more layers that are present, the more complex the neural network model becomes. By learning to reconstruct undersampled k-space data, deep learning effectively approximates the missing frequency components. This allows it to find deeper connections between the signals in k-space, but has the risk of introducing a flaw if the model is too complex; overfitting.

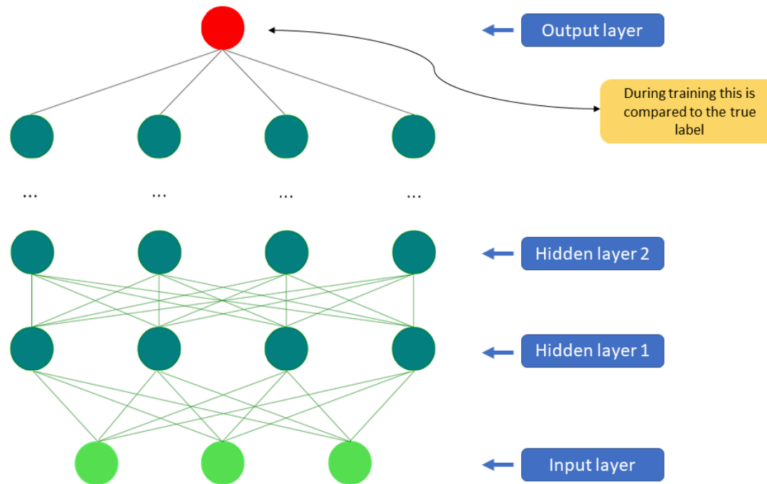


Figure 2: Image from [7] depicting the graphical representation of a deep learning neural network

In the event that a neural network contains too many layers, its ability to analyze data becomes increasingly complex. When this happens, the model starts to memorize specific patterns in the training data, rather than learning generalizable trends. In other words, the model learns too much from the dataset such that it starts to make predictions that are biased towards the data it is trained on. This is known as overfitting, and when applied to an image processing algorithm, can be referred to as hallucinations. When the model learns too deeply about its training data, it may start to predict patterns that are specific to only the training data, and not anything in nature or any real pattern. The model then tries to impose these features onto other problems when they shouldn't be there. A model trained on mice brains when presented with the **brain of a rat** may try to enhance the image of the rat brain by imposing features of the mice brains it studied onto the image; this is called a hallucination. Training a model effectively involves not only training it with a given dataset, but ensuring that the model does not become too complex and bias your output data. This is crucial in the implementation for MRI scans, since a biased model may hide important features of the image such as tumours that doctors or researchers are looking for. However, if the model is correctly trained and does not overfit, **it can be given** some frequency domain data that is undersampled **which** does not contain the entire spectrum, and the model will regenerate a correct image that can be analyzed, **as shown by** [1]. This can save time and memory usage on the computer when processing MRI scans, allowing for quicker scan times and faster image processing for less computational cost [1].

Another way to help prevent overfitting is by using Convolutional Neural Net-

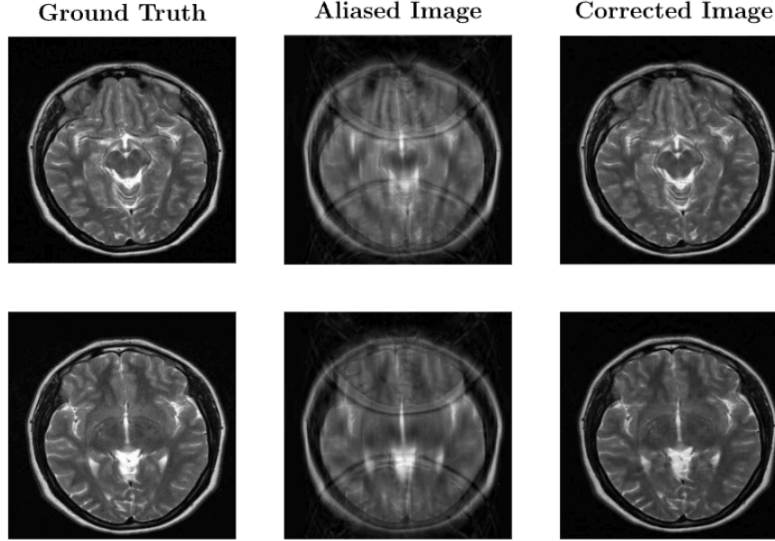


Figure 3: Example of MRI image reconstruction using a deep learning model, showing improved contrast and reduced aliasing [1].

works (CNNs). CNNs are a specific archetype of neural networks with a slightly different layering scheme. Since the **more layers** that are present in the network, the more complex the algorithm is, we can reduce the number of compared functions for each node in the layer to provide less complex data to the model. This keeps the nonlinear capabilities of the model by keeping many layers, but removes the likelihood of overfitting by supplying less data at each iteration. A convolutional neural network does this by forming two different types of layers; a convolution layer and a pooling layer.

The convolutional layer only compares a few nodes with **each other**, typically a small patch of 3×3 nodes [8]. This allows for each feature being compared in the smaller layer to hold more impact, and its correlation between the features to be more pronounced. This provides insight on the correlation between a few features, instead of trying to compare all features at once.

The pooling layer comes after the convolutional layer has been computed, and simplifies the number of parameters in the function. This is typically done by reducing the size of the feature matrix being passed in to the layer. A reduced system of feature equations to be solved means less computation time and less complexity, further reducing the chances of overtraining the model and causing hallucinations.

Methods

In order to begin training a CNN, a model must first be chosen. For this report, tensorflow was utilized as the main machine learning library in conjunction with scikit-learn. Tensorflow is a machine learning library built by Google, and contains many useful features that allow for simple setup of a CNN, such as MaxPooling2D, which allows for the pooling process of 2-dimensional data. This is particularly useful for when we pass 2D k-space data to the neural network. All models were trained on an Apple Macbook air M2 with 16Gb of unified memory.

First, the dataset was downloaded. This project uses the IXI dataset [9], which contains MRI image data from various hospitals in England. The dataset provides T_1 and T_2 weighted images, but the focus of this project will mainly be on T_1 weighted images. The datasets are quite large (around 11Gb each), so to limit the size and training time of the model, the data for T_1 weighted images was saved and stored in a compressed CSV called a parquet. Next, the data is converted to a dataframe in pandas for easy data manipulation. From here, the desired slices from the 581 different images present are stored in a separate data file for processing.

Five slices at a similar depth are gathered for each patient in order to increase the data available to the model, but many slices are discarded since the focus of this project is brain scans, not the entire skull. Discarding slices far away from the point of interest helps the model focus on learning brain images, and not become confused about a slice containing frequency data for a jaw, for example.

Now that the data has been gathered correctly, a second dataset is constructed identical to the first. Since all of the data processed so far is from the IXI dataset, there is no undersampled k-space data. Here, all photos are randomly sampled and have regions of high-frequency k-space removed from them. This random approach removes any bias from the user when creating undersampled images. The low frequency ranges are prioritized here since they contain the most important pieces of information. This artificial undersampled data serves as our testing data for the model.

Once the frequency data for the desired 2D slices is gathered, the real and imaginary parts of the k-space data are split up. This is necessary since most CNN models do not accept complex data. To work around this, the imaginary part of the data is stored as if it was a second real value, and then is recombined at the end of running the CNN with the real part of the data to successfully reconstruct the image. From here, many combinations of layers, iterations, batches, and epochs are tested as well as the intensity of the undersampling. A batch is a dataset that has been divided into smaller parts to be given to the network, and an epoch is the complete analysis of all of the batches at once. The images are then gathered and model accuracy is evaluated using the mean squared error, as well as expressing the

number of correct predictions versus incorrect predictions as a percentage. This process is repeated at different intensities for the model to test for underfitting, overfitting, and generalization. It is important to note here that the dataset only contains 581 images. Most neural networks require thousands of images to learn effectively, and the model may have reduced performance due to the lack of data. This is countered by having high learning rates (the amount of learned information passed from one layer to the next, expressed as a percentage of the total information gathered) from pixel to pixel in the neural network, with many iterations through the data.

Discussion and Results

After training many models on the artificially undersampled IXI dataset, it was found that the models were able to successfully learn from the frequency space data and recreate the MRI image scans. Two examples are shown displaying the predictive power of the CNN and its ability to reconstruct missing k-space data in figures 4 and 5. Given a medium intensity undersampled k-space, the frequency data was successfully reconstructed, leading to a reconstructed image space which resembles the original image of the patient without undersampled k-space data. The additional blurring to the image is due to the model's dataset being quite small, likely causing underfitting.

It can be shown in figures 6 and 8 that this approach is not perfect, but with a deeper model and more training time as well as a larger sample dataset the network shows promise. This is highlighted in figure 7, where a higher computation time

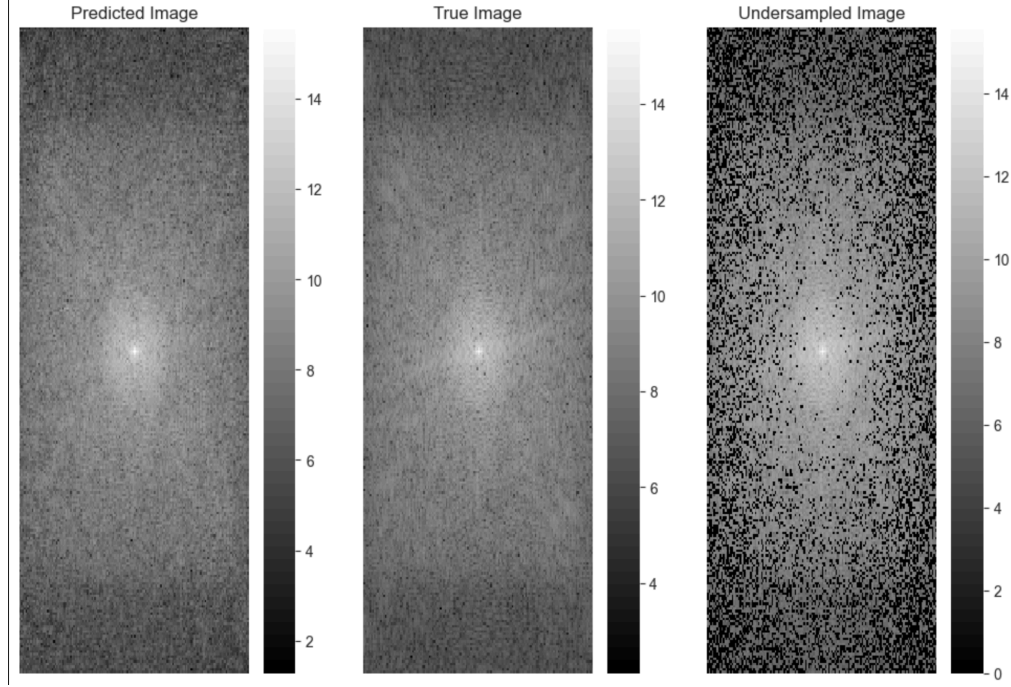


Figure 4: k-space reconstruction using a CNN for medium intensity undersampled k-space.

with a larger model was used for the same data as in image 6. The key purpose here is to show that the neural networks can successfully reconstruct k-space image trends accurately.

The power of this image processing technique works for any scan, as long as the model is properly trained and not overfitting. Research has shown that CNNs are capable of image enhancement and reconstruction for T_1 and T_2 weighted images for a variety of sizes and orientations [10]. These models will only grow in predictive power as more image data becomes available for study. A current issue in this area of research is that MRI scan datasets including k-space data is scarce, and models are typically trained on smaller datasets up to 500 images [10]. An increase in abundance of MRI k-space sampling data will provide the model more information to learn from, which may help it to more accurately produce correct and high

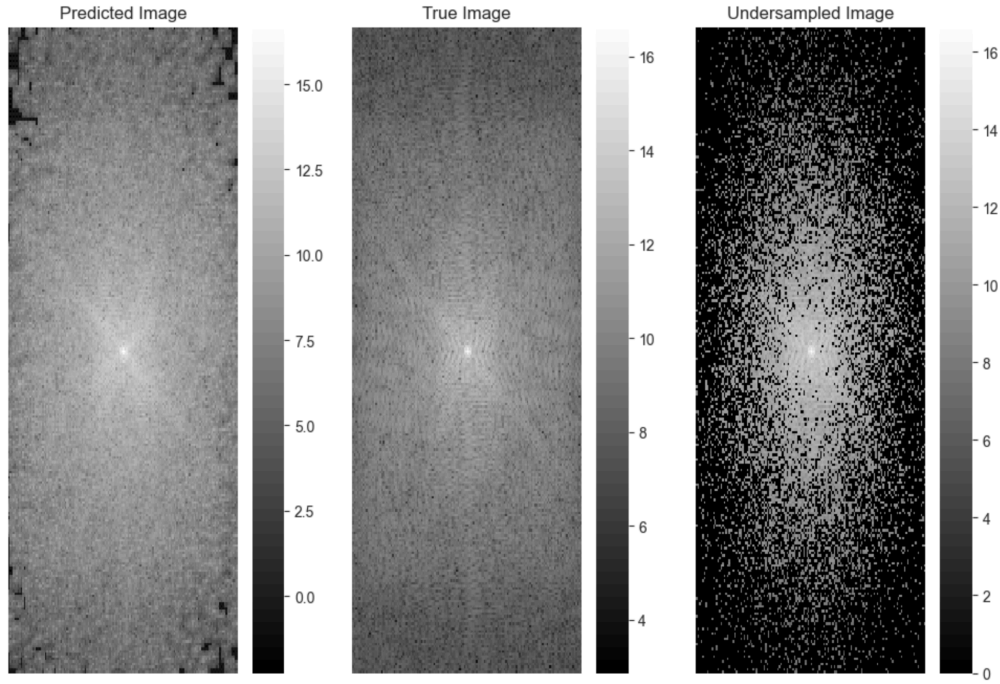


Figure 5: k-space reconstruction using a CNN for heavily undersampled k-space. quality images.

It should be discussed that the approach presented by the small model in this paper is nonstandard. Typically, k-space data is transmitted to image data first, before the CNN is trained on the data [10]. Both approaches have their benefits and drawbacks, but this approach is focused on reconstructing the frequency data. Prioritizing frequency data gives the model more information about phase and signal strength, while prioritizing image data prioritizes features of the image. These are not unrelated since they are linked via the **Fourier** transform depicted in equation (1), but the model will learn different skills while presented with different data. A future extension of this project would be to perform the FFT (**Fast-Fourier Transform**) first, then train the CNN on the true image data, and try to reconstruct its features from there. Another future extension of the project would be to

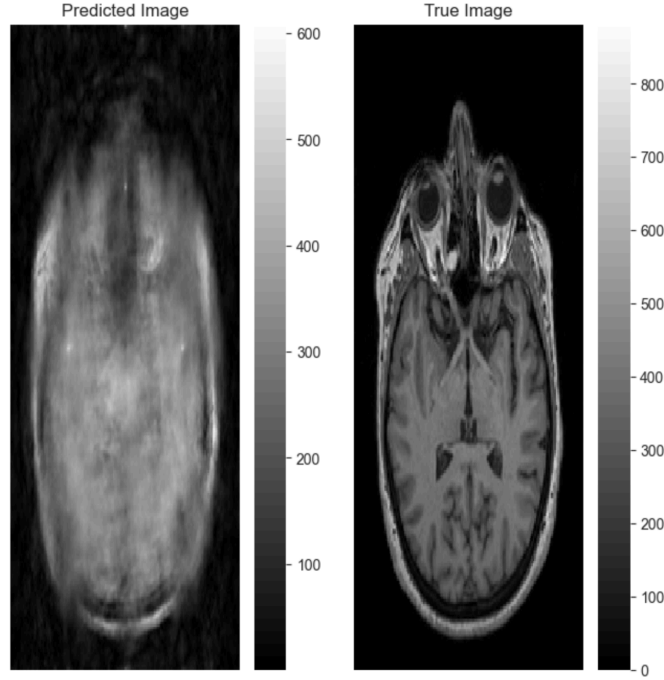


Figure 6: k-space reconstruction using a CNN for highly undersampled k-space.

analyze how the image reconstruction improves if given not only the T_1 -weighted images, but also the T_2 -weighted images and the PD weighted images. Providing a larger sample size will increase the likelihood that the model can better reconstruct frequency information. This comes at a heavy cost of computation time, computational resources, and memory. These CNN's on average took between 30 minutes to an hour to run on an Apple Macbook Air M2 with 16Gb of unified memory. Increasing the information in the dataset would stretch these calculation times much further. This can be circumvented by utilizing GPU programming, but that is a topic outside the scope of this report and can be investigated in a future study.

Additionally, these models have clinical limitations that need to be highlighted and discussed, such as their immediate application in clinical setting. Currently,

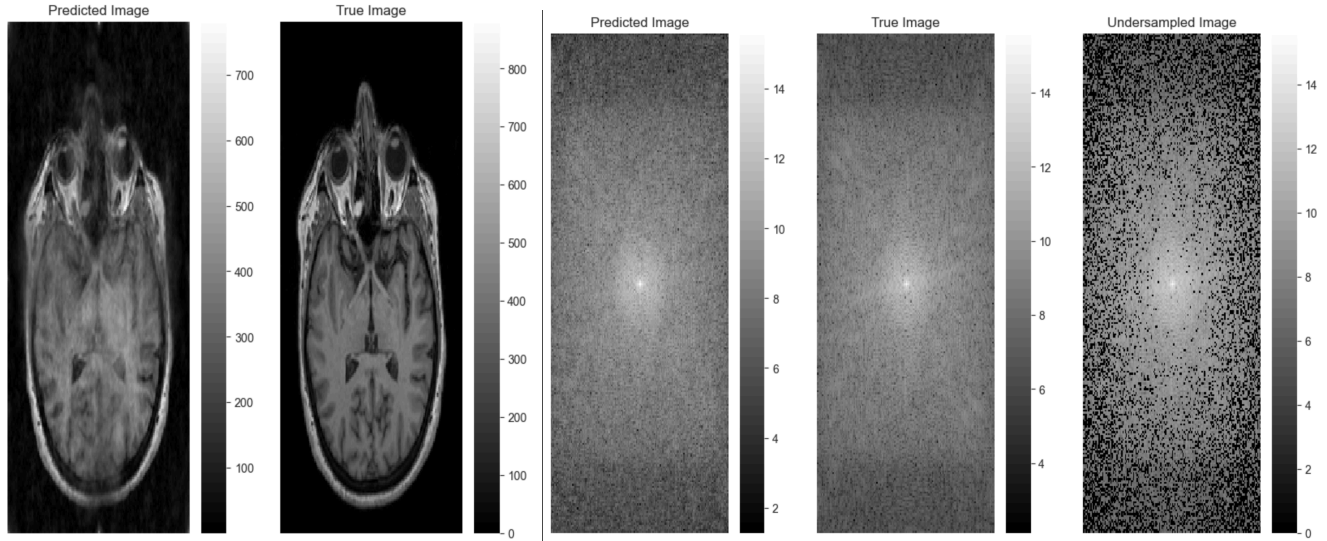


Figure 7: Image data of another sample with higher computation time and a larger CNN.

neural networks and other machine learning algorithms are not used in the clinical setting. This is due to several factors which include the small datasets the models are trained on as mentioned previously, as well as the repercussions of an overfit model. Highly complex models are extremely sensitive to overfitting, and if doctors rely on the image processing done by these models, there is a chance of misdiagnosis, which could have serious effects on patient care and health. Larger datasets and further careful training is required before such models can be implemented in a clinical setting. Having a larger dataset allows for a bigger sample size of images for models to learn from, which not only can reduce underfitting, but can help provide more reliable results. When training a CNN, it is important to consider generalization and overfitting in the model analysis. Without careful consideration of model complexity, image artifacts, blurring, and smoothing effects are likely to emerge.

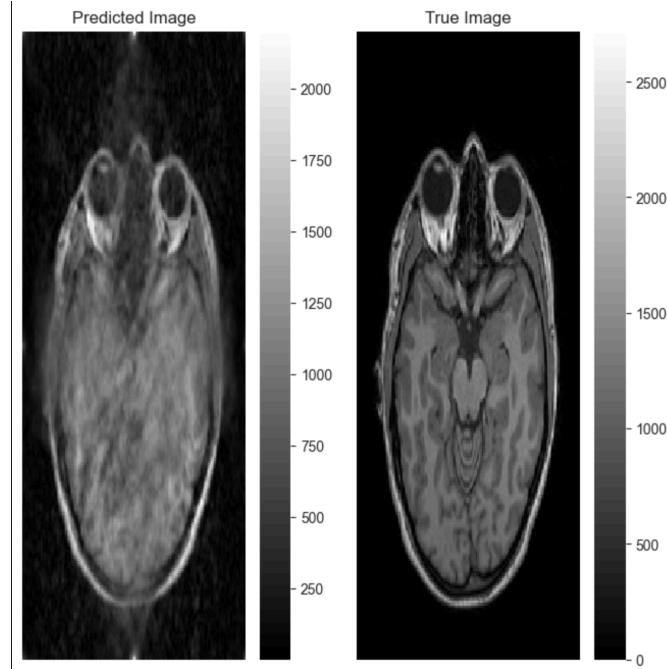


Figure 8: k-space reconstruction using a CNN for medium intensity undersampled k-space.

Conclusion

In conclusion it has been shown that neural networks can serve as a powerful tool for reducing the amount of frequency domain acquisition needed from MRI scans to produce meaningful images of a sample. By training a neural network to analyze the frequency domain data and predict the missing acquisition data for a soft tissue sample it is possible to reconstruct the image without aliasing and significant loss of contrast.

As the area of research grows, more complex and larger datasets will become available, allowing for further advanced training of the Concurrent Neural Networks. The application of CNNs in the medical field for MRI imaging has the power to reduce data acquisition times and allow for more patients to be treated

via MRI. This can help in the identification of cancerous cells and other tumours in the samples being studied and allow MRI to become a more accessible imaging technique to those in need.

Bibliography

- [1] 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, p. 135007, Jun. 2018. [Online]. Available: <https://iopscience.iop.org/article/10.1088/1361-6560/aac71a>
- [2] A. R. Jha and D. G. Pillai, 2021.
- [3] K. Hammernik, T. Klatzer, E. Kobler, M. P. Recht, D. K. Sodickson, T. Pock, and F. Knoll, “Learning a variational network for reconstruction of accelerated mri data,” *Magnetic Resonance in Medicine*, vol. 79, no. 6, pp. 3055–3071, Jun. 2018. [Online]. Available: <https://onlinelibrary.wiley.com/doi/10.1002/mrm.26977>
- [4] S. Raschka, *Machine Learning with PyTorch and Scikit-Learn: Develop machine learning and deep learning models with Python*, 1st ed. Birmingham: Packt Publishing Limited, 2022.
- [5] L. Gazit, *Mastering NLP from Foundations to LLMs: Apply advanced rule-based techniques to LLMs and solve real-world business problems using Python*, 1st ed. Birmingham: Packt Publishing Limited, 2024.

- [6] D. Hoyle, 15 Math Concepts Every Data Scientist Should Know: Understand and learn how to apply the math behind data science algorithms, 1st ed. Birmingham: Packt Publishing Limited, 2024.
- [7] A. S. Lundervold and A. Lundervold, “An overview of deep learning in medical imaging focusing on mri,” *Zeitschrift für Medizinische Physik*, vol. 29, no. 2, pp. 102–127, May 2019. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S0939388918301181>
- [8] J. Liu, Y. Pan, M. Li, Z. Chen, L. Tang, C. Lu, and J. Wang, “Applications of deep learning to mri images: A survey,” *Big Data Mining and Analytics*, vol. 1, no. 1, pp. 1–18, Mar. 2018. [Online]. Available: <https://ieeexplore.ieee.org/document/8268732/>
- [9] [Online]. Available: <https://brain-development.org/ixi-dataset/>
- [10] S. Wang, Z. Su, L. Ying, X. Peng, S. Zhu, F. Liang, D. Feng, and D. Liang, “Accelerating magnetic resonance imaging via deep learning,” in 2016 IEEE 13th International Symposium on Biomedical Imaging (ISBI). Prague, Czech Republic: IEEE, Apr. 2016, pp. 514–517. [Online]. Available: <http://ieeexplore.ieee.org/document/7493320/>