Reconstructing MRI: K-space Imputation and Image Reconstruction (Project Report - CSC2541)

Duc Truong
MScAC, DCS
(1005565149)
University of Toronto
dhtruon3@cs.toronto.edu

Pulkit Mathur
MScAC, DCS
(1005483692)
University of Toronto
pulkit@cs.toronto.edu

Sumeet Ranka
MScAC, DCS
(1004945152)
University of Toronto
ranka47@cs.toronto.edu

Vaibhav Saxena
MScAC, DCS
(1004824639)
University of Toronto
vaibhav@cs.toronto.edu

Abstract

In magnetic resonance imaging (MRI), undersampling the k-space is widely adopted for acceleration of the process. However, there is a trade-off between the acquisition speed and the reconstructed image's quality. To address this challenge, we explore several novel machine learning frameworks that have the potential of constructing ill-posed MR images, caused by k-space undersampling, to accurate high quality images. To solve the problem, we developed our systems based on two different approaches: k-space imputation using De-noising Autoencoder; and image reconstruction using Generative Adversarial Networks. K-space imputation is a less explored process, and we present our analysis which brings out interesting challenges in this aspect of MRI reconstruction. We also implemented an end-to-end model, which combined both k-space imputation and image reconstruction to generate sharper MRI images from the blurry ones.

1 Introduction

The use of magnetic resonance imaging (MRI) is growing exponentially, because of its excellent anatomic and pathological details provided through the images. However, the long acquisition time in MRI, which generally exceeds 30 minutes, leads to low patient throughput, patient discomfort and noncompliance, artifacts from patient motion, and high examination costs [1]. As a consequence, decreasing the acquisition time is one of the major ongoing research goal since the advent of MRI in the 1970s. The goal can be achieved through both hardware developments (such as improved magnetic field gradients) and software advances (such as improved image reconstruction). In this project, we will focus on using machine learning approaches to optimize the MR image reconstruction process. However, we first need to understand the fundamentals of MRI.

MRI works by acquiring signals from the Hydrogen nuclei in the object under observation. The object to be scanned is placed in a strong magnetic field, which causes the spins of the hydrogen nuclei to align either parallel or anti-parallel to the field. Radio frequency (RF) pulse sequences are then applied to excite the Hydrogen nuclei in our cells. Receiver coils are simultaneously used to capture the electromagnetic signals which are reflected from the body part. These signals are stored in the form of k-space (Fourier space, or the raw data space of MRI). A point in the k-space contains specific frequency, phase (x-y coordinates) and signal intensity information (brightness). Inverse Fourier transformation is applied after k-space acquisition to derive the final image. Every pixel in the resultant image is the weighted sum of all the individual points in the k-space. The size of the k-space is the same as the size of the MR image. However, a point in the k-space does not correspond

Repository Links:

- https://github.com/ranka47/MRI_reconstruction
- https://github.com/vaibhavsaxena11/pGAN-MRI
- https://github.com/mathurp/end_to_end_MRI_reconstruction

to a point in the image matrix. The data at the center of the k-space, which has **low frequency value, contains most of the signal information and the contrast information** of the image, and the data along the outer edges, which has **higher frequency values, contains information about the edges and the boundaries.** Another important property of k-space is the existence of symmetrical data; according to this property only half of the k-space data will need to be collected and the remaining half of the k-space can be estimated mathematically by using complex conjugate synthesis. This technique can potentially reduce the MR data acquisition time, but it did not provide successful results because of low Signal-to-Noise ratio (SNR) values of the reconstructed image.

A feasible approach to speed up MRI data acquisition is to reduce the amount of the collected k-space data. This technique known as undersampling, however, can lead to the aliasing artifacts in the reconstructed images [2]. There are basically two approaches to tackle this problem: k-space imputation, and image reconstruction. In [3], the authors mentioned several classical Compressed Sensing (CS) approaches. However, it was reported that classical CS methods have multiple limitations regarding the computational efficiency and constructed image's quality.

In this paper, we propose to apply machine learning techniques to find an optimized reconstruction function, which can generate images closely resembling the ground truth. We start by breaking down the problem to its fundamentals in Section 2, description of the dataset we used in Section 3, and the evaluation metrics we used in Section 4. In Section 5, we develop a deep neural network model which focuses on the k-space imputation task. Besides that, inspired by the sharp, high texture quality images retrieved by Generative Adversarial Networks (GANs) in the other fields, we also developed our perceptual GAN (pGAN), in which we were inspired by the framework proposed in [4]. In pGAN, we adopt the U-net architecture with skip connections for the generator network, with a variety of losses giving a training signal to the generator, which we describe in Section 6. Finally, in Section 7, we present our model for end-to-end MRI reconstruction which contains a model for k-space imputation followed by image reconstruction. In Section 8, we summarize our quantitative results, and talk about some our challenges in Section 9. In Section 10, we talk about a possible future direction using Discriminator Rejection Sampling [5] for confidence quantification. We talk about the division of our work for this project in Section 11, and finally conclude in Section 12.

2 Breaking down the problem

It has been argued that the optimal reconstruction for an MRI image can be obtained only when the original measurements for the MRI are maintained. In other words, the 'unmasked' frequencies, or the frequencies that were measured, remain unchanged. Hence, we maintain two points to be kept in mind:

1. Reconstructed images should be **data-consistent**.
2. Reconstructed images should **appear plausible**.

Correspondingly, there are two approaches to MRI reconstruction that we experiment with:

1. **K-space imputation:** In this approach, we leave the original measured frequencies untouched. Theoretically, this is the only way we can obtain an 'optimal' reconstruction.
2. **Image reconstruction:** In this approach, we give ourselves the liberty to manipulate all the possible frequencies, both measured and un-measured.

Finally, we also experiment with combining both the approaches in our end-to-end setup, where we first try to impute the k-space frequencies, and use the imputed the k-space to construct an image using inverse fast Fourier transform (iFFT), an operation that converts constituent frequencies in the k-space to the original signal i.e. image, and use this image as input to an image reconstruction model.

3 Dataset description

For the analysis, the dataset collected by NYU Langone [1] is used. The anonymized imaging data comprises raw k-space data from more than 1,500 fully sampled knee MRIs obtained on 3 and 1.5 Tesla magnets and DICOM images from 10,000 clinical knee MRIs also obtained at 3 or 1.5 Tesla. The raw dataset, which is used as part of this project, includes data from two pulse sequences, yielding coronal proton density-weighted images with and without fat suppression. The sequence parameters

were as follows: echo train length 4, matrix size 320x320, in-plane resolution 0.5mmx0.5mm, slice thickness 3mm, no gap between slices. Originally collected using multi-coil methodology, the multi-coil k-Space data was converted to single-coil data using emulated single-coil (ESC) methodology. ESC computes a complex-valued linear combination of the responses from multiple coils, with the linear combination fitted to the ground-truth root-sum-of-squares reconstruction in the least-squares sense.

As part of this project, the focus was on the **single-coil** track of the dataset. The k-space data is divided in four parts: training, validation, test, and challenge. The first two contain the fully-sampled acquisitions with ground-truth images. The latter two contain only the undersampled k-space data. K-space data is a complex-valued matrix with height of 640 and varying width. Four information are required to describe the frequency component of an image: **direction, frequency, amplitude, and phase**. Each element in the matrix represents information for each individual sampled frequency. The spatial location in the matrix with respect to the center describes the direction and the individual spatial frequency component. The magnitude and the angle of the complex number stored represents the amplitude and the phase of the frequency signal, i.e. points in k-space near the center represent low frequency signals whereas those farther from the center represent high frequency signals. It is interesting to note that **low frequency signals make up the contrast** in the image, while **high frequency signals form the edges**. Figures 8, 9 and 10 in Appendix A describe the understanding of individual frequency information.

For our analysis, we used the original validation set as our test set, and split the original train set into new train and validation sets in a ratio of 7:3. The reason for doing this was that there are no target images in the original test set for us to find quantitative results on. After obtaining these sets, the undersampled k-space over our training and validation data are obtained by applying a mask function. Undersampling low frequency signals reduces the contrast of the image, whereas undersampling high frequencies reduces edges. The undersampled k-space data are used to construct blurry images as well, as per requirement of our models.

Table 1 describes the count statistics for each part of the data.

	Single-coil slices
Training	29427
Validation	5278
Test	7101

Table 1: Number of slices in each set

4 Evaluation Metrics

We use the following evaluation metrics for our models:

NMSE

The normalized mean squared-error between two images X and Y is given by

$$\text{NMSE} = \frac{\frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[X(i,j) - Y(i,j)]^2}{\frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}X(i,j)^2}. \tag{1}$$

SSIM

Structural similarity index (SSIM) is a perception-based metric used to measure the similarity between two images. It considers image degradation as perceived change in structural information, while also incorporating important perceptual phenomena. Structural information is the idea that the pixels have strong inter-dependencies especially when they are spatially close. These dependencies carry important information about the structure of the objects in the visual scene. The SSIM between two images x and y of same size is given by

$$\text{SSIM}(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_1)}, \tag{2}$$

where μ_x is the average of x, μ_y is the average of y, σ_x^2 is the variance of x, σ_y^2 is the variance of y, σ_{xy} is the covariance of x and y, $c_1 = (k_1 L)^2$ and $c_2 = (k_2 L)^2$ are denominator stabilizers, L is the dynamic range of pixel values ($k_1 = 0.01$ and $k_2 = 0.03$ by default).

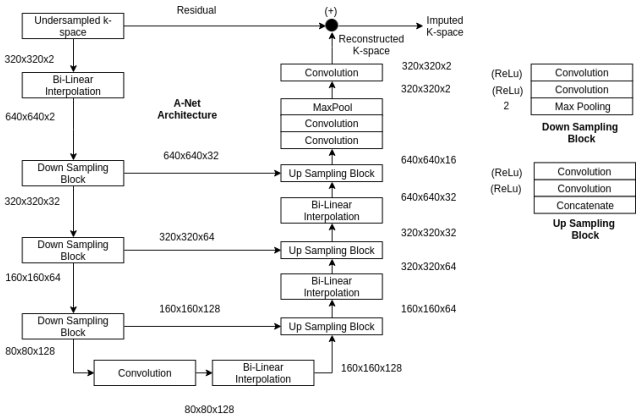


Figure 1: A-Net architecture for k-space imputation.

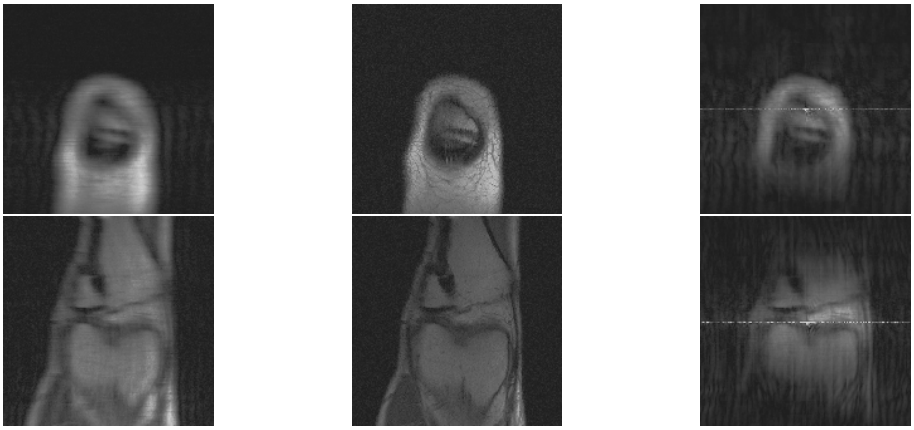


Figure 2: Results from A-Net architecture. The left-most column displays the zero imputed images, the middle column is the target images, and the right-most column displays the obtained images.

PSNR

Peak signal-to-noise ratio is the ratio between the maximum possible power of the signal and the power of the corrupting noise. It is expressed in a logarithmic scale, and is given by

$$\text{PSNR} = 10 \cdot \log_{10} \left(\frac{MAX_I^2}{MSE} \right), \tag{3}$$

where MAX_I is the maximum possible pixel value of the image, the squared value of which gives the signal power, and the MSE gives the noise in the signal.

5 K-space Imputation

5.1 Baseline

We take the most highly used method for k-space imputation in every machine learning based MRI reconstruction paper, which is **zero-filling reconstruction**. It is the substitution of zeroes for unmeasured data points. Zero filling processes can be very practical in everyday clinical usage by reducing scanning times without much loss in resolution or SNR, since zero filled points contain neither signal nor noise. Hence SNR is unaffected.

Using zero-filling reconstruction, we obtained an average NMSE of 0.0512, SSIM of 0.3887, and PSNR of 19.2445 over the test set.

5.2 A-Net: Denoising Autoencoder based U-Net

Denoising autoencoder (DAE) [6] is a technique applied in autoencoders to make them robust to partial corruption of the input data. The idea is to project the input onto a larger latent space and then

reconstruct the original input by penalizing on the loss dependent on the difference between the input and the output.

U-Net [7], originally designed for image segmentation, is one of the proposed baseline model for MRI reconstruction. Our aim was to **combine the advantages of both DAE and U-Net** to impute the masked values of k-space. The masked input k-space is bilinearly interpolated from 320x320x2 to 640x640x2 before being fed to the U-Net. The output from the U-Net, of shape 640x640x16, is then passed through a series of convolutional layers with kernel size of 1 to reduce the number of channels to 8 and then 2. After that, the size of the output is reduced to 320x320x2 using MaxPool layer.

The idea behind the architecture is that every element in the input, i.e. information for every frequency component, is projected to a larger latent space based on the local and global context. Each element is then expanded to a space of 2x2 and represented in increasing number of channels. As a result, an element is finally represented using 4 nodes in the lowest layer of A-Net. After up-scaling the input, the obtained latent representation is compressed back to the original input size. This leads to combining the information learned across multiple channels to get the reconstructed k-space. The imputed values from the masked positions are then combined with the unmasked value to get the final output.

The code can be found on `https://github.com/ranka47/MRI_reconstruction`.

5.3 Loss functions

We used two loss functions, combined using different weights:

1. **Image MSE**: Between target image and image obtained from imputed k-space; Weight = 0.1
2. **K-space MSE**: Between fully-sampled k-space and reconstructed k-space; Weight = 2.2

The weights were obtained through multiple runs of the code.

5.4 Experiments and Results

The model was trained on Google Cloud Platform with 4 virtual CPUs, and NVIDIA Tesla P100 GPU. The model was implemented using PyTorch v1.0, and CUDA v9.0. Fig. 11 (Appendix B) displays the change in loss over the iterations. Fig. 2 displays some of the obtained results. It can be noticed that the loss is noisy. This can be attributed to a small batch-size which we worked with due to the lack of resources. The batch size was taken as 6 making an epoch having a little more than 8000 iterations.

Our reconstructed images obtained from k-space imputation appear much sharper than the images obtained after zero-filling reconstruction. However, we observe a **'zipper artifact'** in our outputs, an artifact which occurs due to disturbances in the phase, which we were unable to remove even after extensive experiments where we gave higher weights to the phases of the imputed frequencies.

We ran our trained model on the test set, and achieved average NMSE, SSIM and PSNR of 0.1508, 0.534 and 24.54 respectively.

6 Image Reconstruction using a Perceptual GAN

Now, we discuss the second aspect of MRI reconstruction, as mentioned in Section 2, where we have the liberty to change measured frequencies in order to make the image look more plausible.

The entire code for this section, including the Jupyter notebook, can be found on `https://github.com/vaibhavsaxena11/pGAN-MRI`.

6.1 Loss functions

Let I be an image and K be the corresponding k-space for that image, obtained using fast Fourier transform (FFT). We use I and K for the target image and k-space, and $\hat{I}$ and $\hat{K}$ for the reconstructed image and k-space.

Image MSE

$$\mathcal{L}_{\text{img}} = \frac{1}{2} \left\| \hat{I} - I \right\|_2^2 \tag{4}$$

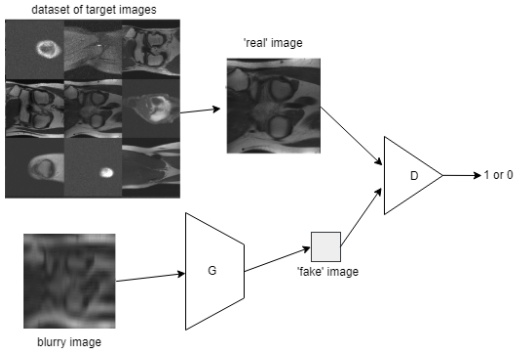


Figure 3: MRI image reconstruction framework using an adversarial objective.

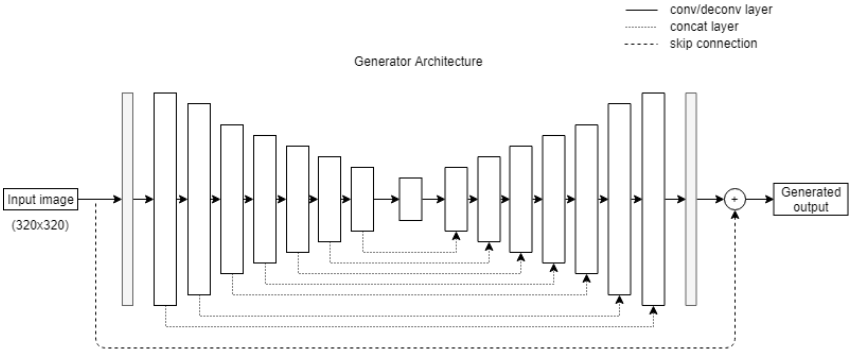


Figure 4: Generator network used for image reconstruction.

where $\hat{I} - I$ denotes a pixel-wise difference between the reconstructed and target image.

K-space MSE

$$\mathcal{L}_{\text{freq}} = \frac{1}{2} \left\| \hat{K} - K \right\|_2^2 \tag{5}$$

where $\hat{K} - K$ denotes the point-wise difference between the reconstructed and target image in the frequency domain.

Adversarial loss

$$\mathcal{L}_{\text{adv}} = -\log(D_{\theta_d}(G_{\theta_g}(I'))) \tag{6}$$

where D is a discriminator network whose output should be 1 for a target image, and 0 for a reconstructed image from the generator G, and I' is the blurry input image to the generator.

Perceptual loss

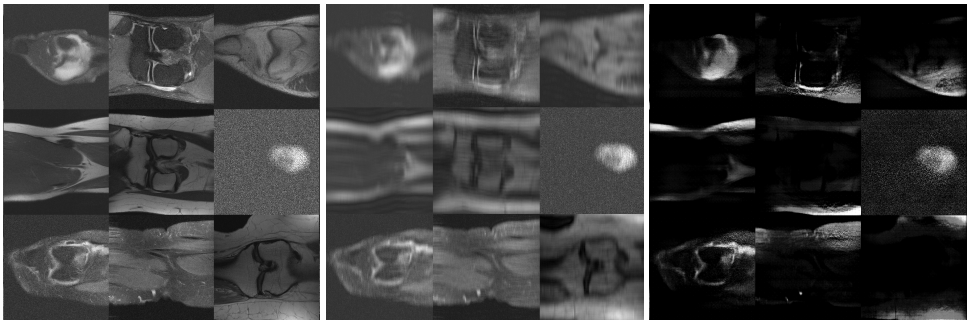
$$\mathcal{L}_{\text{perceptual}} = \frac{1}{2} \left\| VGG(\hat{I}) - VGG(I) \right\|_2^2 \tag{7}$$

We used this perceptual loss to account for perceptual similarity between the reconstructed image and target image [8]. This loss is based on the intermediate ReLU activation layers of the pre-trained 16 layer VGG network described in Simonyan and Zisserman [9]. Such a perceptual loss helps in identifying visually more convincing anatomical or pathological details in the image.

The total loss was calculated as a weighted sum of these losses, so that their magnitudes become approximately the same. We weighted $\mathcal{L}_{\text{img}}$ by 15, $\mathcal{L}_{\text{freq}}$ by 0.1, $\mathcal{L}_{\text{perceptual}}$ by 0.0025, and $\mathcal{L}_{\text{adv}}$ by 1.

6.2 Model architecture

We use a GAN framework for our image reconstruction pipeline, which we call perceptual GAN (or pGAN), where the input to the generator is a blurry image, and the output of the generator is fed into the discriminator as a 'fake' image. The target MRI is fed into the discriminator as an image



(a) Sample target images. (b) Sample blurry images. (c) Reconstructed images.

Figure 5: GAN-based image reconstruction. It is interesting to note that even though our GAN did not reconstruct the image completely, the partially reconstructed images from the p-GAN are **sharper** than the image constructed by zero-filling k-space.

from the 'real' distribution. This framework gives us the adversarial loss for the discriminator and the generator, and is illustrated in Fig. 3. We use 3 other losses as mentioned in Section 6.1.

We use a succession of 8 convolutional layers for the discriminator, and a U-Net architecture for the generator, with 8 convolutional layers for the encoding part and 8 deconvolutional layers for the decoding part. We also put a skip connection from the input to the output of the final layer of the model. This enabled the generator to learn a refinement over the input image which can be added to the input image to obtain the final output. The model architecture is illustrated in Fig. 4.

6.3 Experiments and Results

Our system was setup on Google Cloud Platform with 4 virtual CPUs, and NVIDIA Tesla P100 GPU. The model was implemented using Tensorflow v1.10.0, CUDNN v7.1.3, and CUDA v8.0.

To evaluate the performance of our proposed model, we did experiments on the knee MRI dataset as described in Section 3. We trained the entire networks for 10 epochs. We used Adam optimizer with an initial learning rate of 0.0001, with a decay of 0.5 every 2 epochs. Since there was GPU memory limitation, the batch size was only 16 images. During the first epoch, the generator loss decreased greatly from 2764 to around 120. However, for the epochs afterwards, we did not see any obvious progress on the generator. As shown in the loss plots in our jupyter notebook, starting from epoch 2, the discriminator loss stayed at 0 most of the time. As a result, the generator could hardly learn anything further, and its loss fluctuated around 120. This is a well-known problem of GANs models, called diminished gradient. This means that the discriminator got too successful that the generator gradient vanished and learned nothing. We then ran our trained model on the test set, and achieved average NMSE, SSIM and PSNR of 0.7862; 0.1365 and 23.1372 respectively. The generated images are illustrated in Fig. 5. The Jupyter notebook displaying our numerical results can be found at `https://github.com/vaibhavsaxena11/pGAN-MRI/blob/master/test_results/model_test.ipynb`

7 End-to-end Modeling (K-space imputation followed by Image reconstruction)

We present a hybrid framework, as illustrated in Fig. 6, which operates in both k-space and image domain. We start with the undersampled k-space, perform imputation over it, and then send the image constructed using the imputed k-space to an image reconstruction model to get the final output.

The entire code for this section, including the Jupyter notebook, can be found on `https://github.com/mathurp/end_to_end_MRI_reconstruction`.

7.1 Model architecture

The framework comprises of residual U-Net for **k-space imputation** and a vanilla U-Net for **image reconstruction**. These U-Nets are connected by an inverse fast Fourier transform (iFFT) operation which converts the residual U-Net's output k-space to image and passes it to a vanilla U-Net for image reconstruction. The model does not need to learn this domain transformation function (iFFT), which essentially reduces our model parameter complexity to O(n^2).

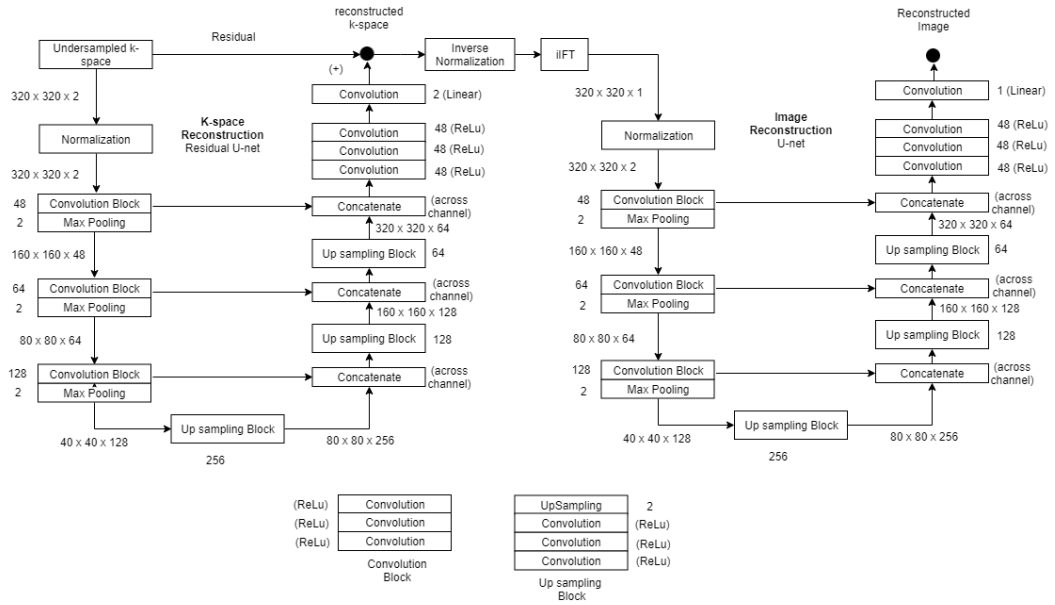


Figure 6: End to end architecture for k-space imputation and image reconstruction.

Each U-Net first down-samples the input by reducing the dimension by 8 times and increasing the number of channels to 256. During up-sampling, we use a transpose convolution instead of unmaxpooling which helps the model to learn during up-sampling as well. An end-to-end architecture like this helps to backpropagate both k-space and image loss which helps the model to learn faster. Moreover, it takes advantage of information presented in k-space and image domain, as opposed to other image domain only approaches [4].

7.2 Loss functions

We use a **MSE loss for k-space imputation** ($\mathcal{L}_{\text{fMSE}}$), and **MSE and SSIM losses for image reconstruction** ($\mathcal{L}_{\text{iMSE}}$ and $\mathcal{L}_{\text{iSSIM}}$). The total loss is calculated at the end of each U-Net by comparing the U-Net output with its corresponding target in k-space and image space correspondingly. The final loss is given by a weighted sum of the three losses:

$$\mathcal{L}_{\text{tot}} = \alpha\mathcal{L}_{\text{fMSE}} + \beta\mathcal{L}_{\text{iMSE}} + \gamma\mathcal{L}_{\text{iSSIM}} \tag{8}$$

We set $\alpha = 0.02, \beta = 0.80, \gamma = 0.18$ for our experiments.

7.3 Experiments and Results

Our system was setup on Google Cloud Platform with 4 virtual CPUs, and NVIDIA Tesla P100 GPU. The model was implemented using Tensorflow v1.10.0, CUDNN v7.1.3, and CUDA v8.0.

The entire network was trained for 40 epochs. We used the Adam optimizer with an initial learning rate of 0.0001 and decay of 10^{-7}. We used batch size of 16 comprising of under-sampled k-space data. To prevent over-fitting, we also used a L_2 regularizer at each convolution layer with weight 0.01.

In Fig. 7, we illustrate the result of our end-to-end model with MSE+SSIM loss. We can infer that our model was able to reduce the NMSE from 0.00785 to 0.00778, and increase SSIM from 0.51 to 0.57. The increase in SSIM was achieved as a result of the SSIM loss function being included in our model. Overall, the model was able to reconstruct fine details in the input blurred images, including reconstructing sharp edges and increasing the contrast. A detailed set of reconstructed test images can be found on `https://github.com/mathurp/end_to_end_MRI_reconstruction/blob/master/end-to-end-test.ipynb`

The end-to-end model with MSE loss gave an average NMSE of 0.0389, SSIM of 0.3981, and PSNR of 19.6499 over the entire test set. The end-to-end model with MSE+SSIM loss gave an average NMSE of 0.0335, SSIM of 0.4400, and PSNR of 19.9476 over the entire test set. We observe a clear decrease of NMSE and increase of PSNR when incorporating SSIM into our loss function.

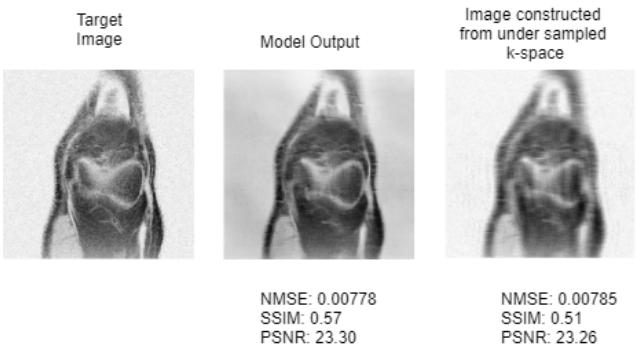


Figure 7: Sample test result for end-to-end reconstruction.

8 Summary of results

We tabulate the quantitative results for all our models and baseline, for the evaluation metrics described in Section 4, in Table 2.

	NMSE	SSIM	PSNR
Zero-filling reconstruction	0.0512	0.3887	19.2445
A-Net	0.1508	0.5340	24.5400
Perceptual GAN	0.7862	0.1365	23.1372
End to End Model (MSE loss)	0.0389	0.3981	19.6499
End to End Model (MSE + SSIM loss)	0.0335	0.4400	19.9476

Table 2: Comparison of results w.r.t. NMSE, SSIM and PSNR metrics.

Our best results over the test set w.r.t. SSIM and PSNR were obtained using A-Net, the k-space imputation model, and the second best results were obtained using the end-to-end model containing both k-space imputation and image reconstruction models and trained using both MSE and SSIM losses.

9 Limitations and Challenges

We mention some of the limitations of our work, which we feel could be improved in a future work.

1. We only worked with single-coil data for our experiments due to lack of computational resources.
2. As shown in a multitude of papers, GANs require a good pre-training for them to be able to perform well. For working with image datasets, most of the state-of-the-art GAN-based applications pre-train on the ImageNet dataset, which we were unable to do due to limited computational resources.
3. Our results are on the validation data with inputs to the models obtained by a mixture of 4x and 8x acceleration undersampling techniques. This subjects our models to a disadvantage when tested over only 4x acceleration undersampled k-space/images.
4. The fastMRI [1] paper suggests to use NMSE, SSIM, and PSNR as evaluation metrics, but does not give specific details on how to weight them. Our end-to-end model uses NMSE and SSIM as loss functions, but does not use PSNR, which is one of the limitations of our model.
5. Since the area of interest in a MRI image is much less as compared to the size of the entire image, it might be possible that a model learns to reconstruct the background and still get a low MSE. Thus, it was difficult to judge as to which metric makes the most sense w.r.t. reconstructing the area of interest.

10 Future works

In order for such a system to be implemented in a real-world clinical setting, the system should be interpretable, and more importantly, it should be able to tell when it is not confident about its results.

One method to for confidence prediction with GANs that we explored so far involves a theoretical technique called Discriminator Rejection Sampling [5] which gives a confidence measure for every image our generative model reconstructs in the form of an **acceptance probability**. Please refer Appendix C for more details on this method.

11 Division of work

Exploratory Data Analysis -

1. Assessing the effects of existing undersampling methods over the constructed image (Sumeet, Pulkit)
2. Exploring into the best possible undersampling technique, which will remain constant throughout future experiments (Duc, Vaibhav)

Setting up the baselines, and analysis of their results -

1. K-space interpolation: Constructing images with zero-filled k-space (Pulkit, Duc)
2. Image reconstruction: Implementing U-Net baseline, as per the guidelines mentioned by Facebook AI Research (Sumeet, Vaibhav)

Implementation of the proposed algorithms -

1. A-Net (Sumeet)
2. Perceptual-GAN (Vaibhav, Duc)
3. End-to-End model (Pulkit)

The final analysis of the models implemented, and the compilation of the results was a collective task.

12 Conclusion

In this work, we implemented multiple models based on the two approaches: k-space imputation, and image reconstruction. For k-space imputation, we analysed the performance of DAEs. We also implemented p-GAN, which demonstrated the image reconstruction task. And last but not least, an end-to-end model was developed, in which we demonstrated the combination of both k-space imputation and image reconstruction.

13 Acknowledgement

We would like to express our very great appreciation to Dr. Ben Fine for his valuable and constructive suggestions during the planning and development of this research work. His willingness to give his time so generously has been very much appreciated.

References

[1] Jure Zbontar et al. fastmri: An open dataset and benchmarks for accelerated mri. *arXiv preprint arXiv:1811.08839*, 2018.

[2] R. T. Seethamraju S. Y. Huang et al. Body mr imaging: Artifacts, k-space and solutions. *RadioImaging*, pages 1439–1460, 2015.

[3] Omur Afacan. K-space undersampling strategies for functional and cardiac mri: Achieving rapid acquisition while maintaining image quality. 2011.

[4] Guang Yang, Simiao Yu, Hao Dong, Gregory G. Slabaugh, Pier Luigi Dragotti, Xujiong Ye, Fangde Liu, Simon R. Arridge, Jennifer Keegan, Yike Guo, and David N. Firmin. DAGAN: deep de-aliasing generative adversarial networks for fast compressed sensing MRI reconstruction. *IEEE Trans. Med. Imaging*, 37(6):1310–1321, 2018.

[5] Samaneh Azadi, Catherine Olsson, Trevor Darrell, Ian Goodfellow, and Augustus Odena. Discriminator rejection sampling. *arXiv preprint arXiv:1810.06758*, 2018.

[6] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103. ACM, 2008.

[7] Olaf Ronneberger et al. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[8] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew P. Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, and Wenzhe Shi. Photo-realistic single image super-resolution using a generative adversarial network. *CoRR*, abs/1609.04802, 2016.

[9] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

Appendices

A Relevance of high and low frequency components

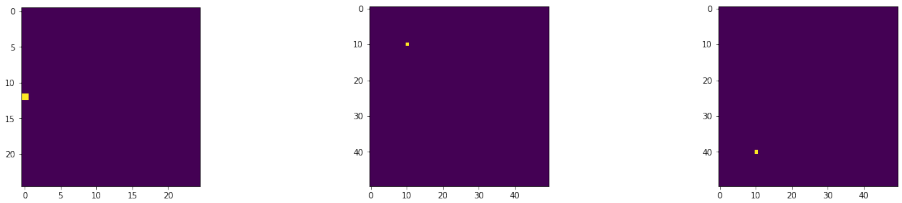


Figure 8: K-space plot for (a) left, (b) top left, (c) bottom left

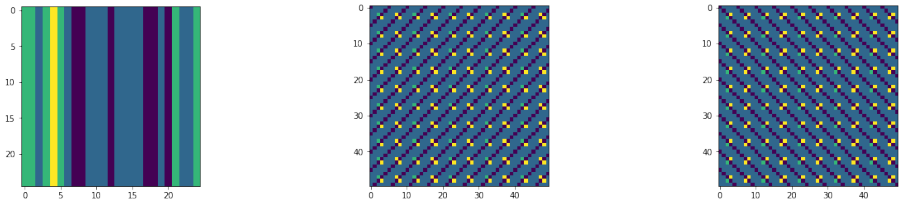
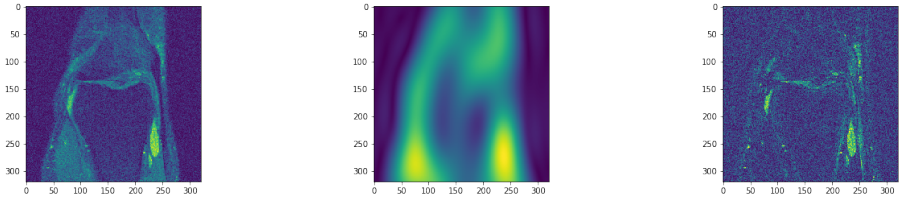


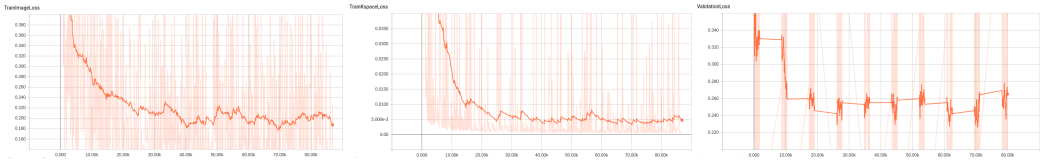
Figure 9: Image space plot for corresponding k-space



(a) Original Image (b) High frequencies removed (c) Low frequencies removed

Figure 10: Relevance of high and low frequency components to represent a high-resolution image

B Training plots for A-Net Architecture



(a) Training: Image Loss (b) Training: K-space Loss (c) Validation: Combined Loss

Figure 11: Graph of iterations vs loss.

C Future work: Confidence measure for image reconstruction systems

For a Generative Adversarial Network setting where $p_r(x)$ denotes the real data distribution and $p_g(x)$ denotes the generator's distribution, Azadi et al. [5] proposed a method for obtaining the ratio of densities $p_r(x)$ and $p_g(x)$ for a given sample x, from the optimal discriminator D^*. This ratio, which directly points to an acceptance probability of the generated image x, can also act as a confidence measure for the image.

Assume that the output of the discriminator is a sigmoid over the logits $\tilde{D}$, as given by

$$D(x) = \frac{1}{1 + e^{-\tilde{D}(x)}} \tag{9}$$

Also, the optimal discriminator D^*, which minimizes the loss for a particular distribution p_g of the generator G, takes the form as

$$D^*(x) = \frac{p_r(x)}{p_r(x) + p_g(x)} \tag{10}$$

Using (9) and (10), we have the following derivation:

$$\begin{aligned} D^*(x) = \frac{1}{1 + e^{-\tilde{D}^*(x)}} &= \frac{p_r(x)}{p_r(x) + p_g(x)} \\ \implies 1 + e^{-\tilde{D}^*(x)} &= \frac{p_r(x) + p_g(x)}{p_r(x)} \\ \implies p_r(x) e^{-\tilde{D}^*(x)} &= p_g(x) \\ \implies \frac{p_r(x)}{p_g(x)} &= e^{\tilde{D}^*(x)} \end{aligned} \tag{11}$$

Therefore, for a fixed generator distribution p_g, if we have the optimal discriminator D^* then we can obtain the ratio of p_r and p_g using (11). The ratio $\frac{p_r(x)}{p_g(x)}$, obtained for any generated image x, can be used as a **confidence measure for any reconstructed MRI image** obtained from the generator.