

UCSF

UC San Francisco Electronic Theses and Dissertations

Title

Comparing The Training Performance of a Deep Neural Network for Accelerated MRI Reconstruction Using Synthesized and Realistic k-space Data

Permalink

<https://escholarship.org/uc/item/1jv9t4q5>

Author

Kemisetti, Anil

Publication Date

2021

Peer reviewed|Thesis/dissertation

Comparing The Training Performance of a Deep Neural Network for Accelerated MRI Reconstruction Using Synthesized and Realistic k-space Data

by
Anil Kemiseti

THESIS

Submitted in partial satisfaction of the requirements for degree of
MASTER OF SCIENCE

in

Biomedical Imaging


in the

GRADUATE DIVISION

of the

UNIVERSITY OF CALIFORNIA, SAN FRANCISCO

Approved:


DocuSigned by:

1644A2CD853841E... _____ Peder Larson
Chair

DocuSigned by:

Valentina Pedoia _____ Valentina Pedoia

DocuSigned by:

Janine Lupo _____ Janine Lupo

DocuSigned by:

E4A4A067E4C8436... _____ Roland Krug

Committee Members

Acknowledgments

I appreciate the dedication, commitment, guidance, and aid of the following groups and individuals, without them I would not have been able to complete this Master's thesis. Thank you.

UCSF Master of Science in Biomedical Imaging Program:

Students and Faculty

UCSF Peder Larson's Lab:

Peder Larson, PhD

Andrew Leynes

UCSF Valentina Pedoia's Lab:

Valentina Pedoia, PhD

Francesco Caliva

Defense Committee:

Janine Lupo, PhD

Roland Kurg, PhD

UCSF Center for Intelligent Imaging:

Pablo Damasceno

Nvidia:

Ahmed Harouni, PhD

Comparing The Training Performance of a Deep Neural Network for Accelerated MRI Reconstruction Using Synthesized and Realistic k-space Data

Abstract

Anil Prabhakar Kannappa Kemiseti

Magnetic Resonance Imaging (MRI) is a powerful medical imaging modality used as a diagnostic tool. There is a steady rise in the use of imaging examinations¹. Trends from 2000 - 2016 showed that nearly 16 million to 21 million patients had enrolled annually in various US health care systems. The number of MRIs increased from 62 per 1000 to 139 per 1000 patients from 2000 to 2016. MR images are usually stored in Picture Archiving and Communication Systems(PACS) in Digital Imaging and Communication in Medicine(DICOM) format. The DICOM format includes a header and image data. In MRI, raw data obtained during the MR signal acquisition is measured in spatial-frequency space, or k-space. The raw k-space data is generally transformed into the anatomical images, and raw data is discarded and not transferred to the PACS. The abundant DICOM data has the potential to be used for training neural networks, as Deep Neural Network models depend on large and extensive training datasets. However, DICOM stores images only as magnitude and without the image phase, making it difficult to generate realistic k-space data. To leverage DICOM data for neural network training using k-space data, for example in MRI reconstruction networks, it is essential to understand the effect of missing image phase information.

My thesis attempts to compare a deep neural network's performance for accelerated MRI reconstruction comparing realistic k-space data to synthesized k-space data using

magnitude images as stored in DICOM objects. MRI offers a great deal of control to the user to acquire the data and reconstruct the clinical images, but suffers from relatively long acquisition times. Typical scan times are between 30 to 40 mins. Scan times go up to 60 mins if a contrast agent needs to be administered. Such long acquisition times are not only expensive but a cause of inconvenience to the subject as it is impossible to stay motionless in the bore during the whole duration. To reduce the scan time, accelerated acquisition and corresponding reconstruction methods are needed.

Methods like compressed sensing and parallel imaging are used to accelerate MRI acquisition. Compressed sensing achieves scan acceleration by overcoming the requirement of Nyquist sampling criteria. An undersampling pattern like the Poisson Disk pattern is typically used to acquire a random subset of data instead of the full k-space. The "sigpy.mri" python library's "Poisson" Application Programming Interface (API) was used to simulate this undersampling². This Python API generates a Variable-Density Poisson-Disc (VDPD) sampling pattern. Compressed Sensing theory mentions that image reconstruction would be possible using signals less than the number indicated by Nyquist as long as the k-space undersampling results in incoherent aliasing, and does not lead to structural aliasing³. This API uses a fully sampled calibration region at the center of the k-space in addition to the acceleration factor. This scheme leads to pseudo-random sampling and avoids structured aliasing artifacts.

After the image acquisition is complete, the reconstruction of the fully sampled k-space data or images with good Signal to Noise Ratio (SNR) must be performed. A deep-learning neural network was trained to perform the reconstruction of the retrospectively undersampled k-space data. The undersampled realistic k-space data's training performance is compared with that of the undersampled synthetic k-space data obtained from magnitude images, as they are stored in DICOM objects.

Our experiments have shown that the resulting images obtained from realistic k-space data

has consistently better initial training performance and faster convergence when compared to the images obtained from synthetic “DICOM” k-space from magnitude images.

However, it is also observed that after training enough epochs, the performance of the model trained using realistic data is comparable to that of the DICOM k-space data. The significance of this finding is in the fact that the abundantly available DICOM data can potentially be used to train a deep neural network for tasks involving k-space data such as MRI reconstruction.

Contents

1	Introduction	1
2	Methods	6
3	Results	18
4	Discussion	35
5	Conclusion	37
	Bibliography	38

List of Figures

2.1	Unet Network Architecture	9
2.2	Nvidia Clara Train Components	10
2.3	Retrospective Undersampling and Reconstruction Pipeline Using Raw k-space	14
2.4	Retrospective Undersampling and Reconstruction Pipeline For DICOM k-space	15
3.1	Sample Images from DICOM k-space reconstruction for All Runs	22
3.2	Images from original k-space reconstruction for All Runs	23
3.3	Histogram for using 10% of data in training	24
3.4	Histogram for using 20% of data in training	25
3.5	Histogram for using 30% of data in training	26
3.6	Histogram for using 40% of data in training	27
3.7	Histogram for using 50% of data in training	28
3.8	Histogram for using 60% of data in training	29
3.9	Histogram for using 100% of data in training	30
3.9	Learning Curves during training	32
3.10	Plot of MAE comparing Runs	33
3.11	Plot of SSIM comparing Runs	33

3.12 Plot of PSNR comparing Runs	34
--	----

List of Tables

3.1	Results of experiments with original k-space data for various amounts of training images	19
3.2	Results of experiments with DICOM k-space data for various amounts of training images	19

List of Acronyms

API Application Programming Interface.

BYOC Bring Your Own Component.

CNNs convolution neural nets.

CS Compressed Sensing.

DICOM Digital Imaging and Communication in Medicine.

GPU Graphics Processing Unit.

MAE Mean Absolute Error.

MRI Magnetic Resonance Imaging.

MSE Mean Squared Error.

NEMA National Electrical Manufacturers Association.

NSP Null Space Property.

PACS Picture Archiving and Communication Systems.

PSNR Peak Signal To Noise Ratio.

QSM Quantitative Susceptibility Mapping.

ReLU Rectified Linear Unit.

RF Radio Frequency.

RIP Restricted Isometry Property.

SNR Signal To Noise Ratio.

SSIM Structural Similarity.

SWI Susceptibility Weighted Imaging.

VDPD Variable Disk Poisson Disk.

VENC Velocity Encoding.

List of Algorithms

1	Variable Density Poisson Disk Sampling	16
2	Simulating “DICOM” k-space using DICOM images	17
3	Generating Masks	17
4	Applying Retrospective Undersampling VDPDS Mask	17

1. Introduction

DICOM, which stands for Digital Imaging and Communication in Medicine, is a standard developed by the National Electrical Manufacturers Association (NEMA) to store the clinical images. These images are generally stored in a Picture Archiving and Communication System (PACS). k-space is the complex-valued raw data in the frequency domain acquired during an MR Scan⁴. This data is processed to create the anatomical images needed for clinical use. These anatomical images are typically stored as magnitude images from the complex-valued images, as they are what are used for clinical interpretation. The raw k-space is almost always discarded to conserve the storage space. DICOM standard only supports these anatomical images⁵. If k-space is needed to be stored, it can be stored in the proprietary private fields of the DICOM metadata header or in other proprietary data formats. The phase images are used in applications like Quantitative Susceptibility Mapping (QSM), Susceptibility Weighted Imaging (SWI), Velocity Encoding (VENC), etc. Even for these phase images, the DICOM standard does not provide a standard specification, and private fields in the metadata headers are used. When it comes to MRI storing of k-space is not supported by the DICOM standard, and only stored in limited cases. Between 2000 and 2016, nearly 135 million imaging studies are conducted in the US and Canada alone¹. Out of these examinations, MRI examinations would be around one fifth. k-space data offers greater potential to conduct research, particularly when investigations include MRI reconstruction, but its availability is limited. It is imperative to find the utility of using the widely-available DICOM data for such research.

For acquiring an MRI image, a strong magnetic field is combined with radio frequency (RF) excitation applied to the subject body. A radio antenna listens for a signal and data is acquired in the frequency domain, or k-space. The number of samples needed to get a clinically viable anatomical image depends on the Nyquist frequency. This MR physics

makes the acquisition relatively slow. A typical MRI exam takes from 30 to 60mins, with individual sequences requiring up to 10 minutes. On the other hand, MR imaging offers a tremendous amount of flexibility because the pulse sequence used to can be designed for a variety of image contrasts and image orientations. There are various techniques to speed up the acquisition such as, using the pulse sequence to acquire multiple lines simultaneously and acquiring multiple slices simultaneously. On the other hand, parallel imaging tries to speed acquisition time up by reducing the number of phase encodes. Another important technique that takes advantage of undersampling is compressed sensing. All these techniques fall into a category that speeds up the acquisition by measuring fewer signals.

According to compressive sensing theory^{6,7}, a fully sampled signal can be reconstructed from incoherently undersampled signals, which must be sparse in a transform domain like the frequency domain. For example, MR images are typically sparse in the wavelet domain. There is a motivation⁶ to acquired an undersampled k-space to speed up MRI acquisition. These characteristics make Compressed Sensing(CS) an ideal candidate as MRI acquisition strategy.

CS reconstruction is an inverse problem. To explain, let us start with a fully sampled signal vector \mathcal{X} of length n . The core idea of compressed sensing is that sampling k signals in the vector \mathcal{X} are enough to reconstruct a fully sampled vector \mathcal{X} . The k sample vector, let us call it \mathcal{Y} , is the undersampled signal which is measured in k-sparse. A matrix, \mathbb{A} called compress sensing matrix transforms vector \mathcal{X} to vector \mathcal{Y} . Conversely, to reconstruct vector \mathcal{X} from \mathcal{Y} , a matrix inverse of \mathbb{A} is need to transform \mathcal{Y} .

$$\mathcal{Y} = \mathbb{A} \cdot \mathcal{X} \tag{1.1}$$

$$\mathcal{X} = \mathbb{A}^{-1} \cdot \mathcal{Y} \tag{1.2}$$

The linear equation system shown in equation 1.1, which this matrix is trying to solve is underdetermined. An underdetermined system would result in the possibility of having

more than one solution for the reconstruction of the fully sampled vector \mathcal{X} . To establish uniqueness in the reconstruction of \mathcal{X} , along with the requirement of incoherent sampling and sparsity in the transform domain for vector \mathcal{Y} , another requirement for CS is the Null Space Property (NSP) of compress sensing matrix \mathbb{A} . A linearly underdetermined system would result in a matrix \mathbb{A} , which has linearly dependent columns. The measure for this linear dependency is called the spark of a matrix. It is the smallest number of linearly dependent columns of a matrix. Null Space Property states that, for a measured $\mathcal{Y} \in \mathbb{R}^m$, there is not more than one signal vector $\mathcal{X} \in k$ such that $\mathcal{Y} = \mathbb{A} \cdot \mathcal{X}$, if and only if \mathbb{A} has the null space property of order $2k$. Finding the compressed sensing matrix \mathbb{A} is an np-hard problem. The solution is an approximation, and the requirement of spark greater than $2k$ bounds the reconstruction error and constraints the maximum compressibility in compressed sensing.

NSP can only help solve the compress sensing matrix when there is no noise in measuring the sparse signal vector \mathcal{Y} . The compress sensing matrix \mathbb{A} should satisfy Restricted Isometry Property (RIP) to make the reconstruction robust in the presence of noise. Isometry preserves the distance between the points after applying a transformation. RIP restricts the amount of change matrix \mathbb{A} is allowed on the original vector to get the sparse vector after transformation. This is achieved by ensuring that the norm-2 size of the resultant vectors $\mathbb{A} \cdot \mathcal{X}$ from transformation using matrix \mathbb{A} would not go very far from the norm-2 size of signal vectors \mathcal{X} due to noise.

The acquisition domain for MRI is k-space. In the k-space, the low spatial frequencies are close to the center and high spatial features towards the edges. Furthermore, data in the center of the kspace has typically larger amplitude when compared to data in the periphery of the kspace. Because of this nature of the kspace, a variable density undersampling random is preferred⁸. There are a variety of ways to perform non-uniform undersampling. It can be done just by random sampling or by jittered stratification. Both these techniques

suffer from sampling more in one region and not sampling in the other areas. This creates clusters and gaps even though it offers better coherence, but it does not provide uniformity. On the other hand, Poisson disk sampling provides incoherence and assurance of minimum distance between the samples. Imagine a disk of a minimum assured radius around each sample where other samples cannot exist. This is how it got the disk in its name. Low-frequency signals are typically higher in amplitude and thus have more information content. So, the lower frequency signals are to be sampled more than the higher frequencies. The radius of the disk is changed from the center of the kspace to the edge of the kspace. This is called variable density Poisson disk sampling⁹. The center of the kspace is sampled very densely. Often a square portion of the region is fully sampled near the kspace for Cartesian sampling.

When Poisson disk sampling is applied to kspace, the signal values at unsampled locations can be initially filled with zero values, or "Zero filling"¹⁰. Reconstructing an image with zero filling results in aliasing artifacts specific to the undersampling pattern. For compressed sensing, incoherent undersampling aliasing artifacts appear noise-like in the image. This is not additive noise and should be treated as leakage of energy from the neighboring non-zero signal values.

The UNet Deep Learning Network was developed for biomedical image segmentation tasks^{10,11}. This architecture can be applied to any image to image task. The network architecture is based on series of blocks in three sections forming a U shape. These three sections are a contracting section, a bottleneck section, and an expansion section. The contracting section is achieved by a series of blocks each having a max pooling layer which is followed by convolution layer and a rectified linear unit (ReLU). Max pooling layers provides the downsampling, which doubles the number of feature channels. The contraction section leads to a bottleneck section followed by an expansion section. The expansion section consists of an up-convolution layer followed by convolution layer and a

ReLU unit. This network also includes skip connection from the upsampling block to the down sampling block with similar resolution. Pixel-wise soft-max is computed in the final layer and combined with cross-entropy loss function to model the segmentation task. This architecture allows to weight the pixels at the border higher to achieve the separation of each pixel in a class need to identify the border for segmentation. So, UNet converts the image segmentation into a multi-class classification problem.

UNet architecture can also be used for MRI reconstruction which is a regression task. MR image reconstruction is an inverse problem¹². For this problem the input is a low-dimensional image and the output is a high-dimensional image. The network is similar to the segmentation but in the final layer it would be pixel-wise regression. The network is trained to model the output pixel value as a regression value of the input pixel value. The loss function is typically the Mean Absolute Error (MAE) of the predicted pixel value and the target value. U-Net uses convolution neural nets (CNNs), which are composed of layers of trainable parameters. CNN based architectures avoid handcrafting features needed for dictionary-based methods.

The downside of deep learning methods is they require large amounts of data to train the network with millions of parameters. With the limited quantity and large size of k-space data, its use in training deep neural network models is challenging to scale. On the other hand, DICOM data is abundantly available. So, this work aims to compare the performance of training of a undersampled reconstruction Unet neural net model when starting from k-space data versus using DICOM images to simulated k-space data.

2. Methods

Dataset

The training for all the experiments was performed using an open dataset called fastMRI¹². This dataset is one of the few publicly available datasets which provides k-space data. The FastMRI dataset offers data from different types of MRI acquisitions of knee and brain. It has raw multi-coil k-space data, which is unprocessed complex-valued multi-coil MR values. It also offers emulated single-coil k-space data, which is derived from the multi-coil acquisition. All the experiments for this work used this single-coil data. FastMRI has 973 volumes of the emulated single-coil knee data with 34742 slices of training data. Such a large number of slices helped train a deep neural network, which demands a large amount of training data. For all the experiments in this work, only the training dataset of the fastMRI dataset was taken and was split up into train, validation, and test for this project. The split was 80%, 10%, and 10% for training, validation, and testing respectively.

Experimental Setup

There are two sets of experiments for this work. Both these experiments start with the k-space data from the fastMRI single-coil training dataset. For the experiment mimicing the use of DICOM image data, new k-space data is simulated from the magnitude image generated from the original k-space data. This simulation is explained in algorithm 2. For this simulation, an inverse Fourier transform is applied to the k-space data to obtain the magnitude image. The simulated k-space is obtained by taking Fourier transform from the above image. From now on, we call this simulated k-space, or DICOM k-space. For the other set of experiments, the original k-space is used. As far as the training is concerned, the neural net input is the image generated after undersampling the original k-space or the DICOM k-space. The label is the image generated from the original k-space or the DICOM

k-space, respectively.

In each of the experiment, k-space is undersampled by applying a mask. This mask is generated using the "sigpy.mri.possion" python api². The algorithm used is explained in algorithm 1. The factor for the central area is chosen as 0.2 in both directions. An acceleration factor of 4 is chosen. These parameters are kept constant for all the experiments. This mask generation is shown in figures 2.4 and 2.3. These pipelines are used for all the experiments. The mask generated is a binary image that is made of ones and zeros. When this mask is applied to the k-space, it removes some of the k-space values and fills them with zeros. This retrospective undersampling represents an accelerated acquisition as the zero fillings represent the measurements that are not obtained while scanning. The magnitude image generated from undersampled k-space is the input to the network explained in section 2.4. The magnitude image generated from the full k-space (before applying the mask) is the ground truth or the label. Seven experiments were performed with 10%, 20%, 30%, 40%, 50%, 60% and 100% of the input images. All these 14 experiments were designed to compare the DICOM k-space training performance with original k-space data. Various sample sizes are chosen to understand convergence behavior and the metrics obtained with various amounts of input data. Only the input size was varied in all these experiments. The same test was used across all the experiments to make the comparison.

Preprocessing and Augmentation

There were 973 volumes of emulated single-coil knee data. The slice size varied from volume to volume. As explained in algorithm 3, a list of all the different slice sizes is obtained. For each size 1000, Poisson disk sampling masks were generated. Before applying the mask, slices from all the volumes are collected, and they are shuffled. k-space for the slices from the shuffled collection is picked, and randomly a mask is obtained from the 1000

mask for the selected slice size, and it is applied. The shuffling and picking of a mask randomly helped the training to see a different batch of undersampled images in every epoch. Thus this scheme contributed to data augmentation and allowed more samples beyond the actual input image size. This is explained in the algorithm 4. After undersampling, the center of magnitude image is cropped to a size of 320x320. This cropped image was the input for training.

Network Architecture

To compare the training performance using DICOM vs. original k-space data, a single domain network¹⁰ is chosen. Both the ground truth as well as the prediction are in the image domain. This reconstruction is modeled as an image to image task to perform pixel-wise regression to achieve reconstruction. In effect, the U-net model achieves the pseudo-inverse operation on the magnitude image of the undersampled k-space as described in equation 1.2. The U-net model has a series of blocks composed of convolution layers along with average-pooling, up-convolution, and ReLU layers. These blocks are composed of millions of trainable parameters. These blocks are organized into three sections¹¹ as shown in figure 2.1. The first among them is the contraction section. This layer uses the max-pooling to achieve the contraction. This contraction path acts as an encoder capturing the context of the image. This contraction section leads to the bottleneck section. This bottleneck section helps the network to achieve compression and learn the critical features in the image. An expansion section follows the bottleneck section. This section aids the network to localize the features learned in the contracting section. Another important component of the U-net architecture is the skip connection. These connections feed the output from the contracting section as an input to the expanding section. Skip connection assists in the localization of the critical features identified in the contraction path. All three sections together form a U-net, which is a fully convolutional neural network. For this work, a single U-net architecture explained above is chosen. In the contracting and

expansion path of this architecture, there are blocks with a filter size of 16,32,64,128, and 256. In the blocks, the convolution layer is followed by the instance normalization. In this case, instance normalization is chosen because it helps in noise-sensitive applications¹³. Another important consideration for neural network architecture is weight initialization. It aids in achieving faster convergence. A larger initialization leads to exploding gradients, and smaller initialization results in vanishing gradients. For the network shown in 2.1 he_normal initialization is chosen. “he_normalization” has shown to be giving better results for ReLU activation functions¹⁴. U-net achieves the contraction using pooling. An important consideration for pooling is to achieve translational invariance. Max pooling is preferred when choosing the brightest pixel is the goal. For example, in the case of detecting edges. Min pooling helps to choose the darkest pixel. On the other hand, the aim of our work is not to do object segmentation. Average pooling works better in the task like reconstruction as it aids in creating a smoother image.

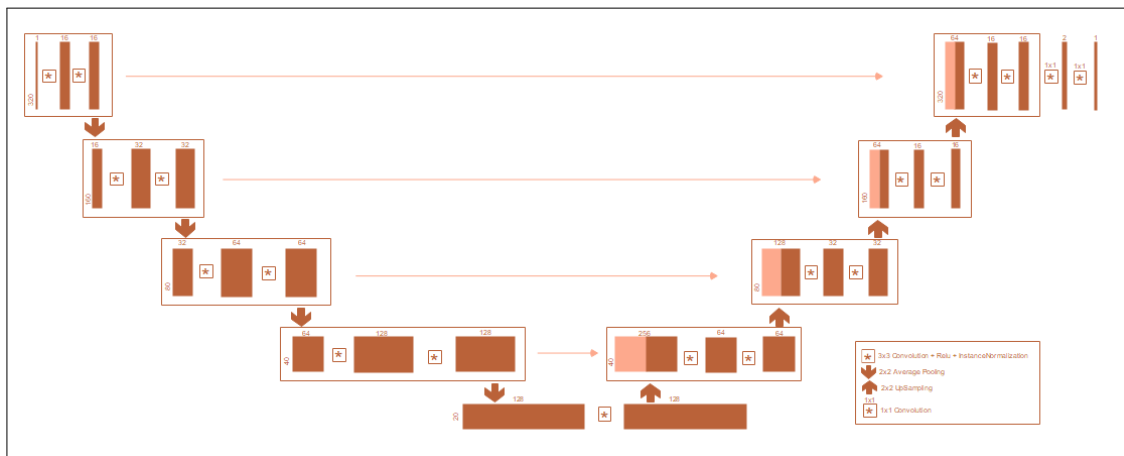


Figure 2.1: Unet Network Architecture

Training

The pipelines shown in figures 2.3 and 2.4 were developed using the Nvidia Clara Train framework (from here referred to simply as Clara). Clara is a python framework that provides an end-to-end workflow for all Deep Learning training needs involving Medical

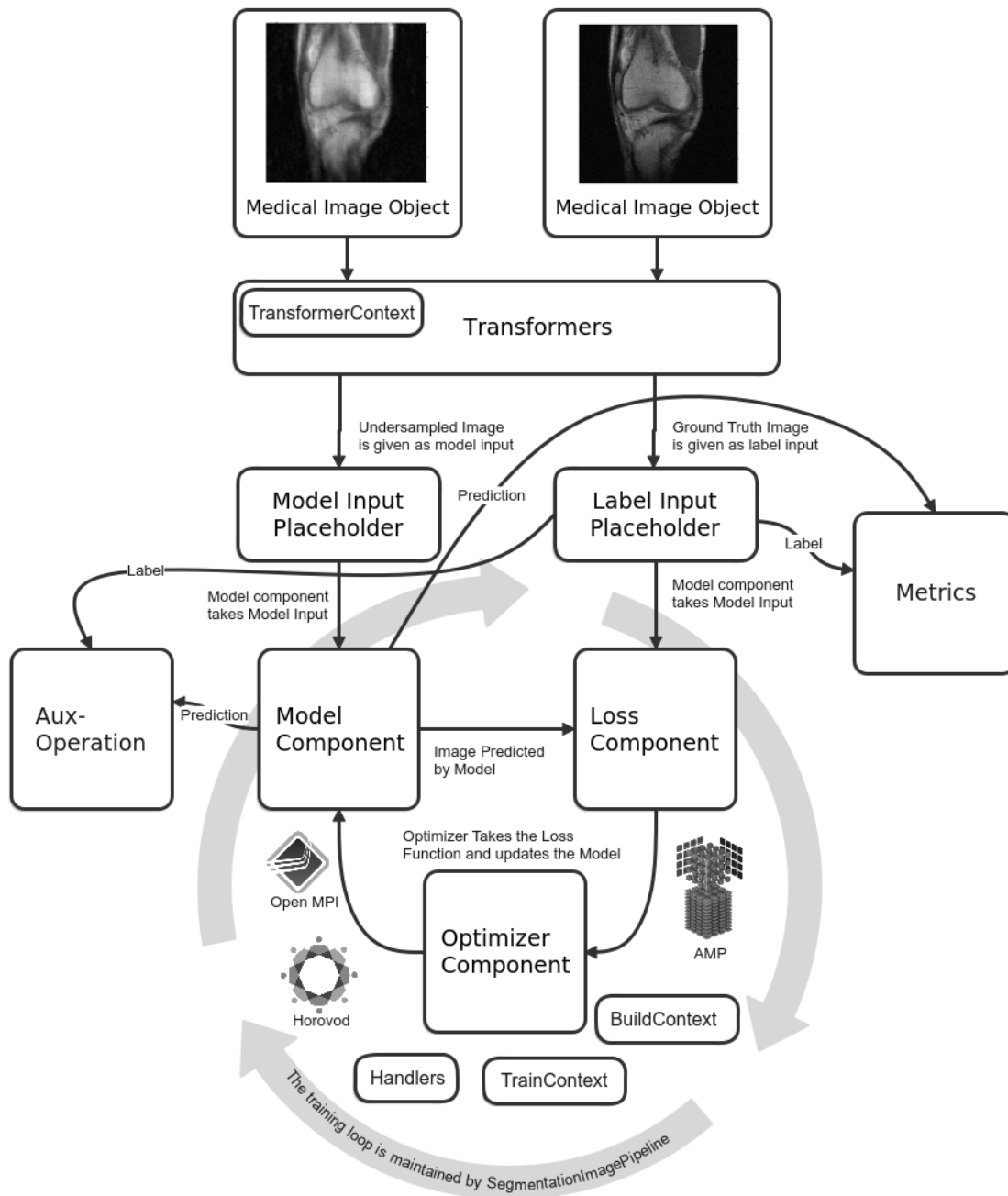


Figure 2.2: Nvidia Clara Train Components

Imaging. This framework is a component-based architecture. These components are described in the figure 2.2. All the image-related attributes are wrapped in a Python object called “medicalImageObject”. This object and the “TransformContext” object are passed

through the transformers. The function of the transformers is to prepare the image data. The input image is prepared and fed to the “model component” for training. The model generates a prediction image. The “loss component” takes the prediction from the model component and the “label input” and feeds it to the optimizer. The “optimizer component.” tries to optimize the model. Deep learning tries to use an iterative method of optimizing the objective function. As shown in the figure 2.2 there is a loop among the model component, loss component, and optimizer component. The SegmentationImagePipeline manages this loop. Other components which help in the training are context objects like “TransformContext”, “BuildContext,” “TrainContext,” and handlers. All these components are configured using a configuration file called “train_config.json”.

Clara also provides an option of using custom python objects called Bring Your Own Component (BYOC). For this work, all custom components were used. The configuration for this training includes an image that is center-cropped to a size of 320x320. The training for this model was supervised. A single pass through a full training dataset is called an epoch. There are thousands of slices, and they cannot be processed in a single iteration, so they are divided into batches. The batch size for all the experiments was 64. Bigger batch size has shown to have a better performance¹².

This training was performed on Nvidia DGX II¹⁵. This machine used Nvidia Telsa V100 Graphics Processing Unit (GPU) with 16 GPUs, with each GPU having a graphics memory of 32GB. Data parallelism was achieved using Horovod¹⁶ and Open MPI. Because only data parallelism was implemented, the batch size is limited to the size of each GPU. A batch size of 64 was the maximum supported by the GPU memory. All the experiments were conducted to have at least have a training of 200 epochs to arrive at a useful comparison of the training performance. A learning rate of 10^{-4} was chosen. Mean Absolute Error (MAE) Loss with l1 regularization with a scale of 5e-7 was used. The adam optimizer provided by Clara was used for this training.

Metrics

Two metrics were used to tune the training and also for the comparison of the training performance during validation. Those are Structural Similarity(SSIM) and Peak Signal To Noise Ratio (PSNR)¹⁷. The goal of the optimizer is to adjust the model to minimize the loss function. Validation is performed after a set number of epochs. The trained model is run through the entire validation dataset, and the above mention metrics are collected. These metrics provide a measure of the current model's quality. Clara keeps track of these validation metrics and stores the current best key metric and the model if it is better than the previous run. Training is continued till all the specified epochs are completed or a stopping value is reached for the key metric. For this project, SSIM was chosen as a key metric.

For a given test image g and a reference image f of $M \times N$ dimensions, the PSNR value is given by equation 2.1.

$$PSNR(f, g) = 10 \log_{10} \left(\frac{255^2}{MSE(f, g)} \right) \quad (2.1)$$

Where

$$MSE(f, g) = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N (f_{ij} - g_{ij})^2 \quad (2.2)$$

PSNR depends on means squared error. Both MSE and PSNR are not effective in discriminating the structural content in the image¹⁷. On the other hand, SSIM has three parts that measure the loss of correlation shown in equation 2.6, luminance distortion 2.4, and contrast distortion 2.5.

$$SSIM(f, g) = l(f, g) \cdot c(f, g) \cdot s(f, g) \quad (2.3)$$

Where

$$l(f, g) = \frac{2\mu_f\mu_g + C_1}{\mu_f^2 + \mu_g^2 + C_1} \quad (2.4)$$

$$c(f, g) = \frac{2\sigma_f\sigma_g + C_2}{\sigma_f^2 + \sigma_g^2 + C_2} \quad (2.5)$$

$$s(f, g) = \frac{\sigma_{fg} + C_3}{\sigma_f\sigma_g + C_3} \quad (2.6)$$

μ_f, μ_g are the local mean and σ_f, σ_g are the local standard deviation of the images. $l(f, g)$ measures the luminance closeness of the two images. The contrast distortion term, $c(f, g)$ gives an idea about the closeness of the contrast of the two images. The third term includes the cross covariance, σ_{fg} , and gives a measure of correlation coefficient. These three terms together closely correlates with the quality of perception of human visual system¹⁸. Hence SSIM was chosen as the key metric.

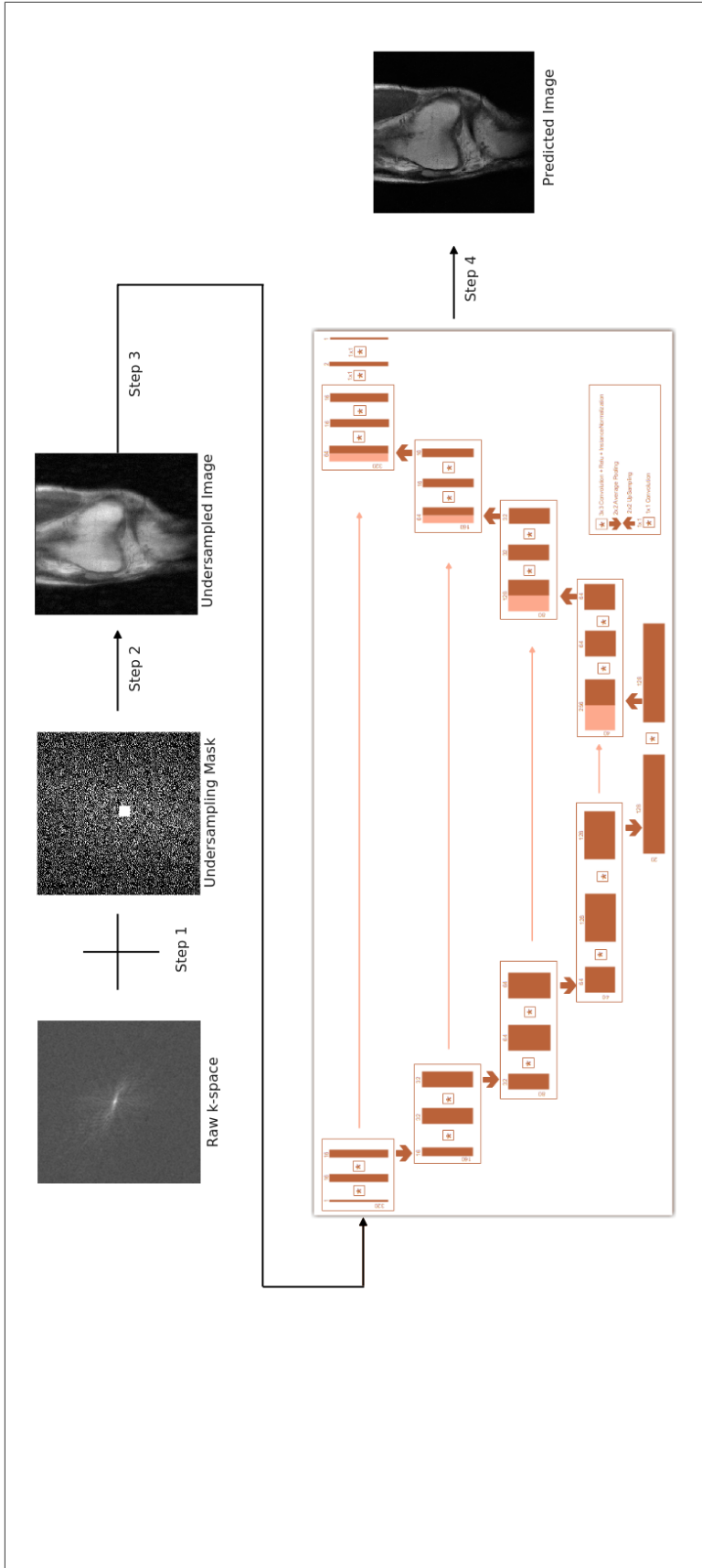


Figure 2.3: Retrospective Undersampling and Reconstruction Pipeline Using Raw k-space

Step 1- Undersampled k-space is generated by applying the variable density Poisson disk sampling mask to the original k-space; Step 2- Magnitude image is generated from the under-sampled k-space obtained in step 1; Step 3- The undersampled magnitude image is the input for the network. Step4- Trained model generates the images, which is the prediction.

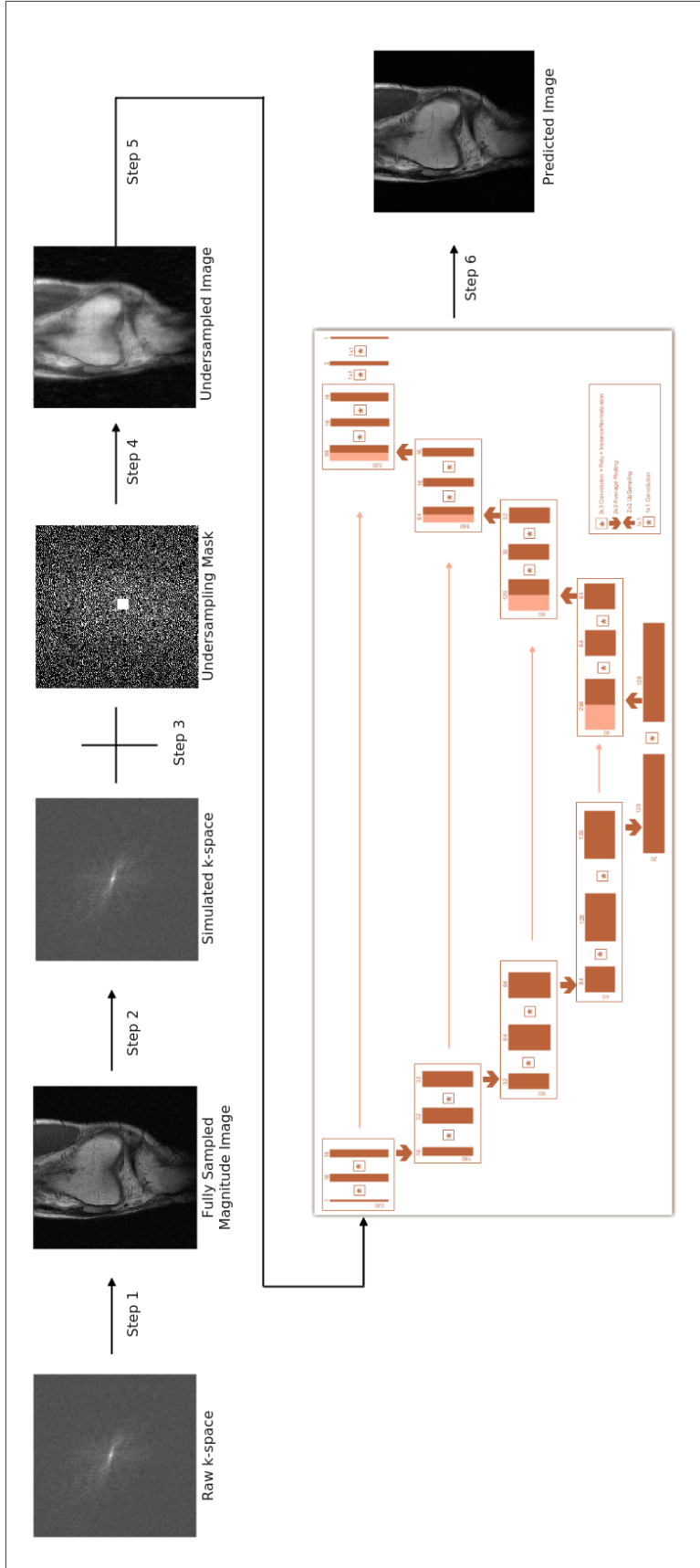


Figure 2.4: Retrospective Undersampling and Reconstruction Pipeline For DICOM k-space

Step 1- The original k-space is taken as an input and a magnitude image is generated. This image is the label or ground truth; Step 2- Simulated k-space is generated from the fully sampled magnitude image; Step 3- Undersampled k-space is generated by applying the variable density Poisson disk sampling mask to the simulated k-space; Step 4- Magnitude image is generated from the under-sampled k-space obtained in step 3; Step 5- The undersampled magnitude image is the input for the network. Step6- Trained model generates the images, which is the prediction.

Algorithm 1: Variable Density Poisson Disk Sampling

Function poisson(*imageShape*, *acceleration*, *calibrationRegion*, *maxAttempts*):

Initialization:

$x, y \leftarrow$ grid using the *imageShape*

$r \leftarrow$ for each cell in the grid using x, y

$s \leftarrow$ Initialize the scaling factor based on the distance from the calibration region to the maximum of image dimensions

while *Either binary search of the scaling factor is exhausted or desired acceleration is achieved* **do**

$scaledRadius \leftarrow (1 + s|r|)$

$mask \leftarrow$ generateMask(*imageShape*, *scaledRadius*, *calibrationRegion*, *maxAttempts*)

$actualAcceleration \leftarrow \frac{\prod_{imageShape\ dimensions}}{\sum_{mask}}$

if *Required Acceleration is achieved* **then**
 break from the loop to return the mask ;

else
 rescale the scale factor ;

return *mask*

Function generateMask(*imageShape*, *scaledRadius*, *calibrationRegion*, *maxAttempts*):

while *points need to be checked* **do**

$i \leftarrow$ select a random index from the grid; $px, py \leftarrow$ retrieve the x and y co-ordinate for i

while *selected index is checked against all the other indexes or the maxAttempts are exhausted* **do**

$v \leftarrow$ generate point randomly from *scaledRadius* and $2 \times scaledRadius$

$t \leftarrow$ generate a random angle

$qx, qy \leftarrow$ calculate the position based the *scaledRadius* and the t angle

if *point defined by qx and qy is outside the grid or close to other points* **then**

 discard the point ;

else
 Add to the mask ;

return *mask*

)

Variable Density Poisson Disk Sampling Algorithm²

Algorithm 2: Simulating “DICOM” k-space using DICOM images

Function `simulatingKSpace(data)`:
 Initialization:
 $data \leftarrow \text{inversefftshift}(data)$
 $data \leftarrow \text{inversefft}(data)$
 $data \leftarrow \text{fftshift}(data)$
 $magnitudeImage \leftarrow \text{absolute_value}(data)$
 $data \leftarrow \text{fftshift}(magnitudeImage)$
 $data \leftarrow \text{fft}(data)$
 $simulatedKSpace \leftarrow \text{inversefftshift}(data)$
 return $simulatedKSpace$

algorithm is for simulating k-space using magnitude images.

Algorithm 3: Generating Masks

Function `genMask(data)`:
 $maskSizes \leftarrow \text{IdentifyMaskSize}$
 $maskdictionary \leftarrow \text{GenerateMaskDictionary}(maskSizes)$
 return $maskdictionary$

Generating VDPDS masks

Algorithm 4: Applying Retrospective Undersampling VDPDS Mask

Function `applyMask(data)`:
 Initialization:
 $undersampledKSpace \leftarrow mask \times full_Kspace$
 return $undersampledKSpace$

Applying Retrospective Undersampling VDPDS Mask.

3. Results

Figure 3.1 and 3.2 show an example reconstruction of the undersampled knee images using DICOM k-space and original k-space data, respectively. The images portions labeled (a) to (f) shown in the figure 3.1 uses, 10% to 60% of training data, respectively. Predicted image intensities oscillate between too dark and too bright for the DICOM k-space data. In figure 3.2, the reconstruction from the original k-space data shows a steady improvement as the percentage of the training data increased from 10% to 60%. The highest quality reconstruction quality is observed in the portion of the figure 3.2 labeled (g). This experiment used 100% of the training data. From the portion labeled (e), it can be observed that 50% of the data also achieves good performance.

Table 3.1 summarizes the results from experiments using original k-space data. Table 3.2 summarizes the results from experiments using DICOM k-space data. These two tables provide a comparison of metrics described in the “Metrics section of Methods” to understand the neural network’s training performance for accelerated MRI reconstruction using the k-space to using DICOM only data. The data used for these experiments is described in the “Datasets section of Methods”. For each category, seven experiments are performed keeping all the parameters constant as described in the “Experimental Setup section of Methods”, only by changing the training data percentage.

The number of epochs are kept constant at 200 during the training described in the “Training Section of Methods”.

The efficiency of the k-space data in training can be further explained from the observations made in the histograms shown in figures from 3.3 - 3.9. Even with as little as 10% of training data the predicted image distributions as shown in portions of the figure 3.3 labeled (a) and (b) indicate that the predicted image distribution using the original k-space data is close to the label image’s distribution when compared to the predicted

Table 3.1: Results of experiments with original k-space data for various amounts of training images

Experiment Name	MAE	PSNR	SSIM
Original k-space Using 100% data	0.284*	28.96 *	0.628*
Original k-space Using 10% data	0.284	28.41	0.607
Original k-space Using 20% data	0.279	28.68	0.617
Original k-space Using 30% data	0.277	28.78	0.621
Original k-space Using 40% data	0.267	28.79	0.622
Original k-space Using 50% data	0.275	28.89	0.626
Original k-space Using 60% data	0.274*	28.97 *	0.628*

This table summarizes the metrics for the experiments conducted with original K-Space data. '*' values indicate the best performance. The network based on U-Net performed well after 60% of the training data was used.

Table 3.2: Results of experiments with DICOM k-space data for various amounts of training images

Experiment Name	MAE	PSNR	SSIM
DICOM k-space Using 100% data	0.293*	28.12 *	0.621*
DICOM k-space Using 10% data	0.381	25.19	0.538
DICOM k-space Using 20% data	0.313	27.42	0.6
DICOM k-space Using 30% data	0.297	28.46	0.615
DICOM k-space Using 40% data	0.327	24.79	0.582
DICOM k-space Using 50% data	0.31	25.78	0.596
DICOM k-space Using 60% data	0.297*	26.68 *	0.608*

This table summarizes the metrics for the experiments conducted with synthesized DICOM k-space data. '*' values indicate the best performance. The network based on U-Net started to perform well after 60% of the training data. But there was still scope of improvement. A comparable performance to using the original k-space data (Table 3.1) was seen only when using the complete training dataset.

distribution from synthesized DICOM k-space data. This is also seen in figures 3.4 with 20% data, figure 3.6 with 40% data, figure 3.7 with 50% data, and figure 3.8 with 60% training data. It is also seen in figure 3.9 with 100% training data.

Another important observation from the portions labeled (a), (b) and (c),(d) of histograms of image intensities shown in figures 3.3 to 3.9 is the spread of the distribution of the difference between the ground truth image intensities and the predicted image intensities.

This spread gradually reduces, indicating the improvement in the training’s prediction performance when using more training data. An important difference is observed in the image portion labeled (b) in the histogram images related to the synthesized DICOM k-space data. The learned intensities were often spread out. Even though there was an overall improvement in the prediction, this suggests the DICOM k-space data-based training is trying to learn the background. A strategy of segmenting¹⁹ the image from the background and only computing the loss function over the image for training might result in better performance.

Figure 3.9 shows the learning curves for all 14 experiments. Portion (a) and (b) of the figure 3.9 shows the MAE loss for both the training using raw data and using the DICOM data. Losses using the original k-space data range from 0.274 to 0.284, as shown in table 3.1, and from 0.293 to 0.381 using the DICOM k-space data, as shown in table 3.2. It indicates that the original k-space data is more efficient in training the network even at a smaller percentage of data than the DICOM k-space data. It can also be observed that loss values for 100% for both the experiments with values of 0.284 and 0.293 are close. It is an indication that even though using original k-space data has better training behavior, a comparable performance could be achieved even with synthesized k-space data from DICOM images given enough data and training. Both plots (a) and (b) show an asymptotic nature, indicating that both k-space data sources could achieve convergence. The two observations about performance and convergence are further confirmed in the subplot of (c), (d) for the metric PSNR, and the subplot of (e), (f) for the metric SSIM. The number of epochs are limited to 200, resulting in a different number of a total number of iterations. PSNR values when using the original k-space data ranges from 28.41 to 28.97 as shown in table 3.1 and it ranges from 25.19 to 28.12 using the DICOM k-space data as shown in table 3.2. Similarly, SSIM, using the original k-space data, range from 0.607 to 0.628, as shown in table 3.1 and it varies from 0.538 to 0.621 using the DICOM k-space data, as shown in table 3.2.

Finally, the training and metrics for all the experiments are summarized in the three plots shown in figures 3.10, 3.12, 3.11 showing the MAE loss, PSNR and SSIM. The observations made above are reinforced in the aggregate representation of the data in these figures. The training using original k-space data has a gradual improvement as the percentage of training data increases. This is represented using the solid line in all three plots. The training using DICOM k-space data is shown using a dotted line in all these plots. This training is not as efficient as the original k-space data but as the amount of training data increases and the training performance increases and the performance using DICOM synthesized k-space data is comparable to that of the original kspace data.

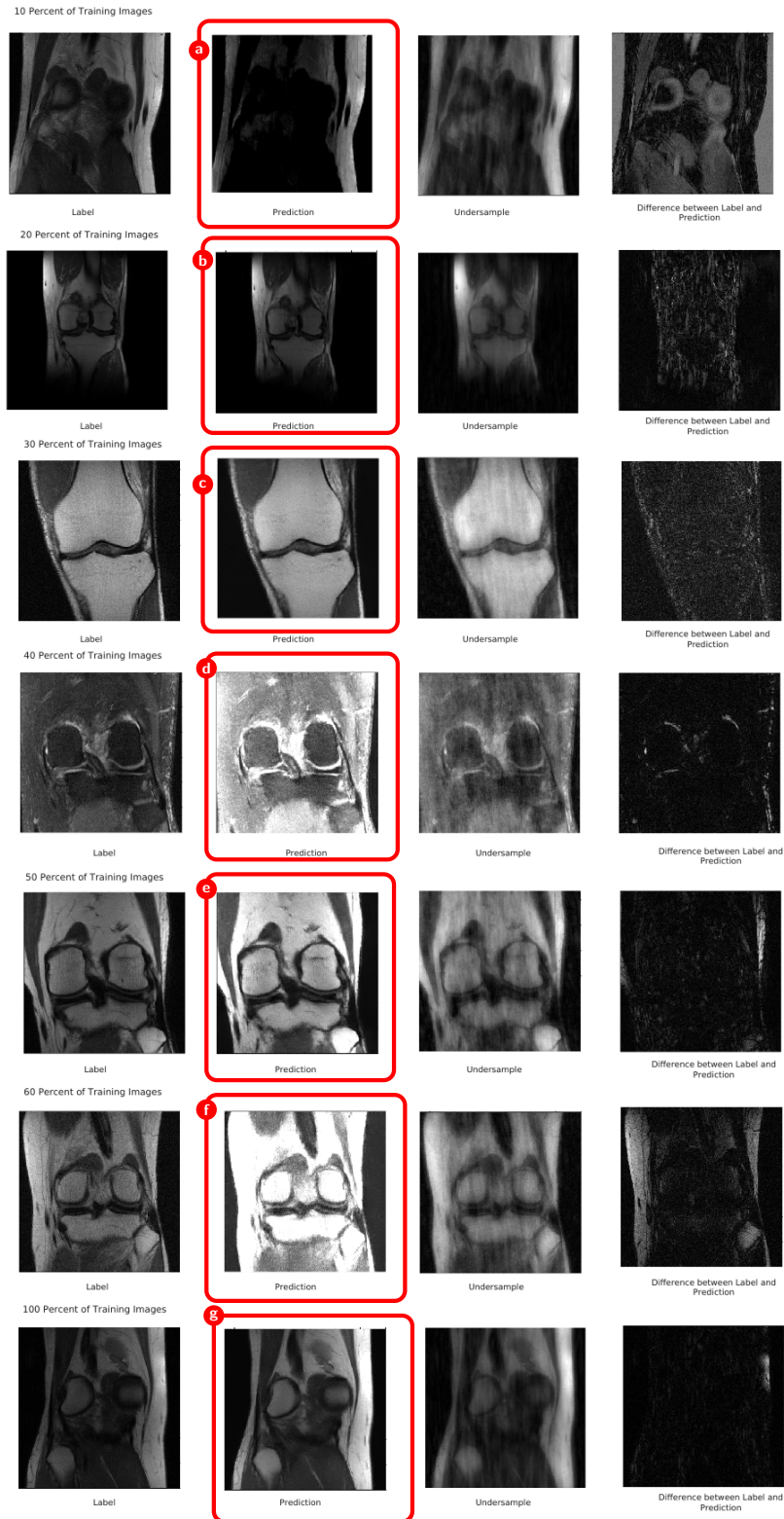


Figure 3.1: Sample Images from DICOM k-space reconstruction for All Runs

This figure shows example of the knee reconstruction following training using synthesized DICOM k-space data. Rows (a)-(f) show results from using increasing amounts of training data, from 10 to 60%, while row (g) used all of the training data.

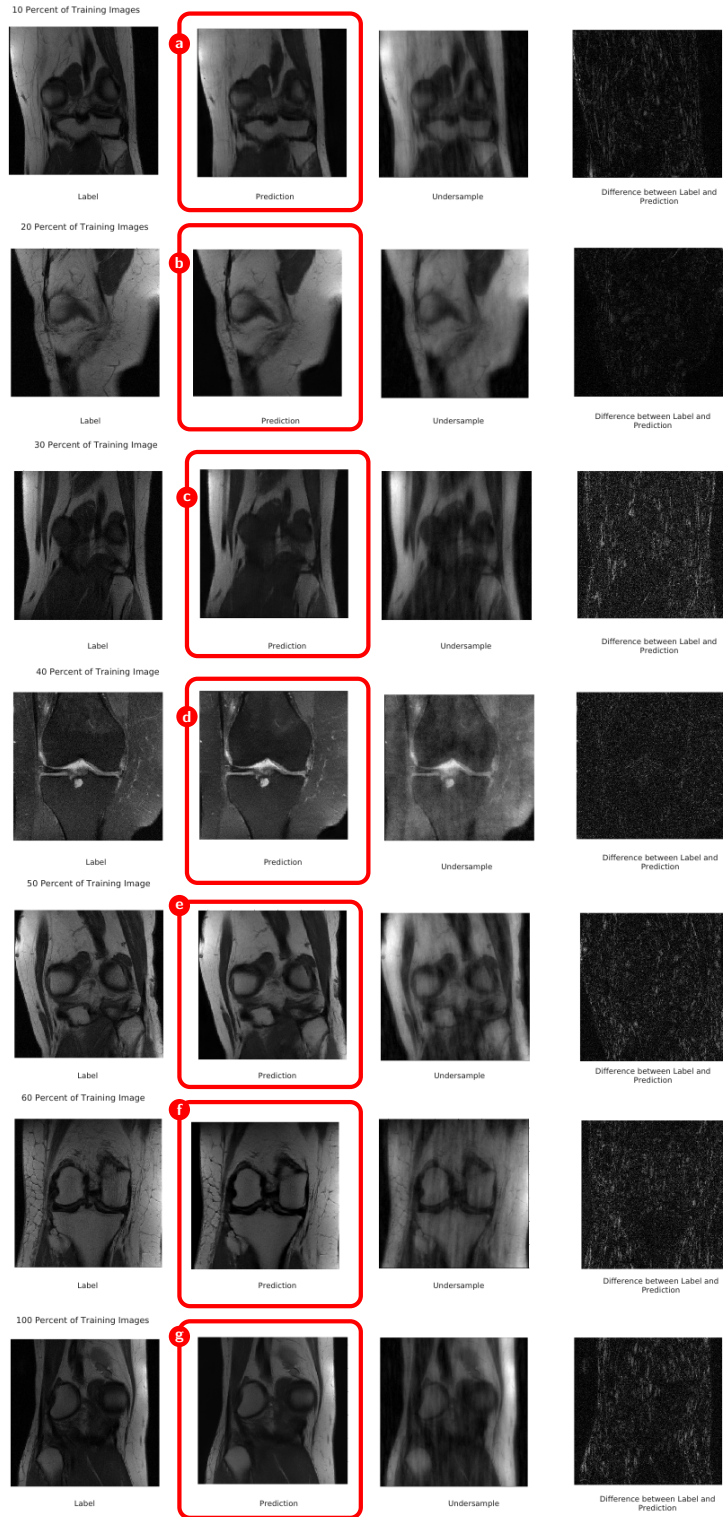


Figure 3.2: Images from original k-space reconstruction for All Runs

This figure shows example of the knee reconstruction captured during training using raw k-space data. Rows (a)-(f) show results from using increasing amounts of training data, from 10 to 60%, while row (g) used all of the training data.



Figure 3.3: Histogram for using 10% of data in training
 A comparison of histograms of the training images using 10% of training data for both original k-space and DICOM k-space data.



Figure 3.4: Histogram for using 20% of data in training
 A comparison of histograms of the training images using 20% of training data for both original k-space and DICOM k-space data.

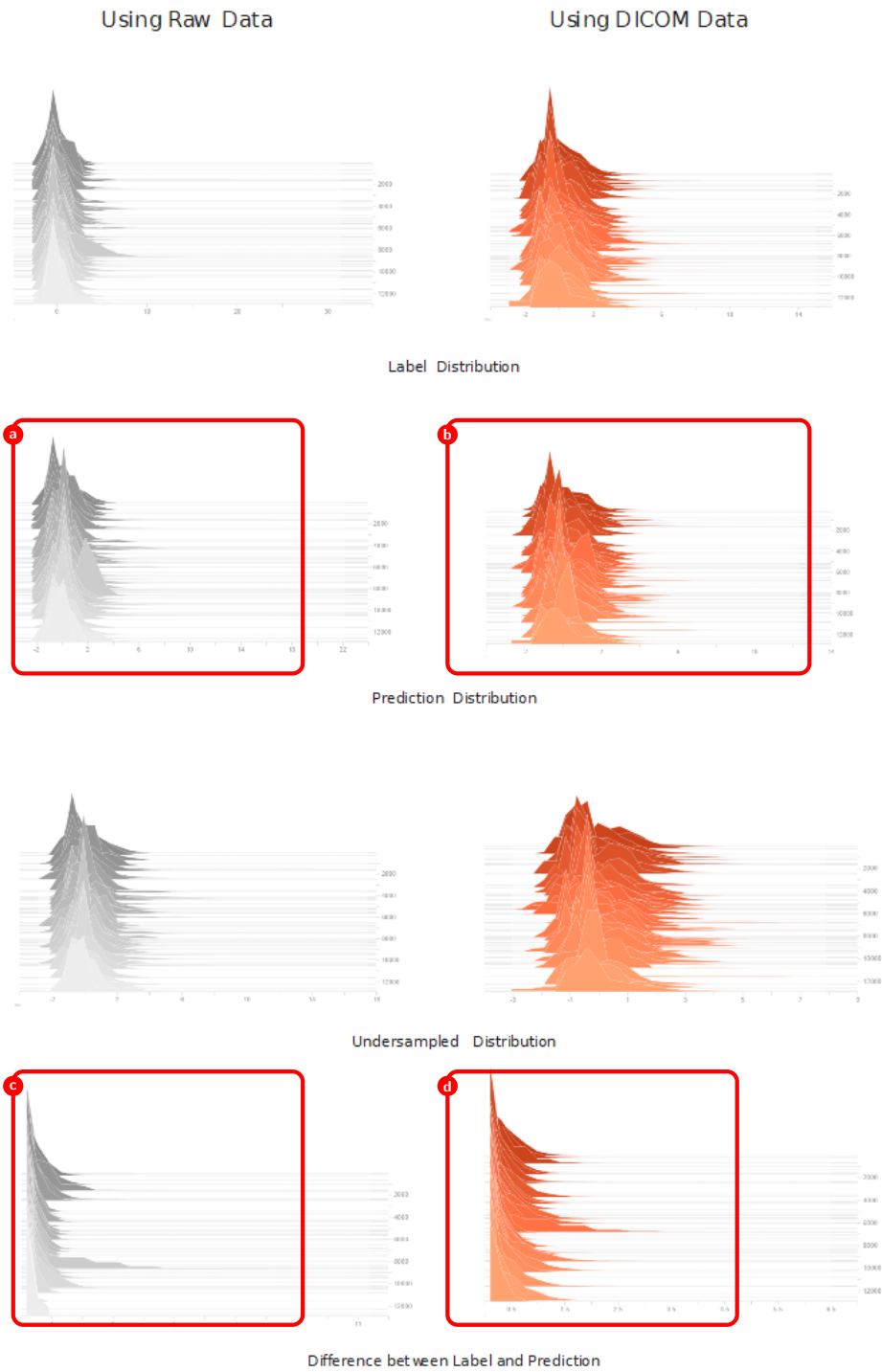


Figure 3.5: Histogram for using 30% of data in training
 A comparison of histograms of the training images using 30% of training data for both original k-space and DICOM k-space data.



Figure 3.6: Histogram for using 40% of data in training
 A comparison of histograms of the training images using 40% of training data for both original k-space and DICOM k-space data.



Figure 3.7: Histogram for using 50% of data in training
 A comparison of histograms of the training images using 50% of training data for both original k-space and DICOM k-space data.

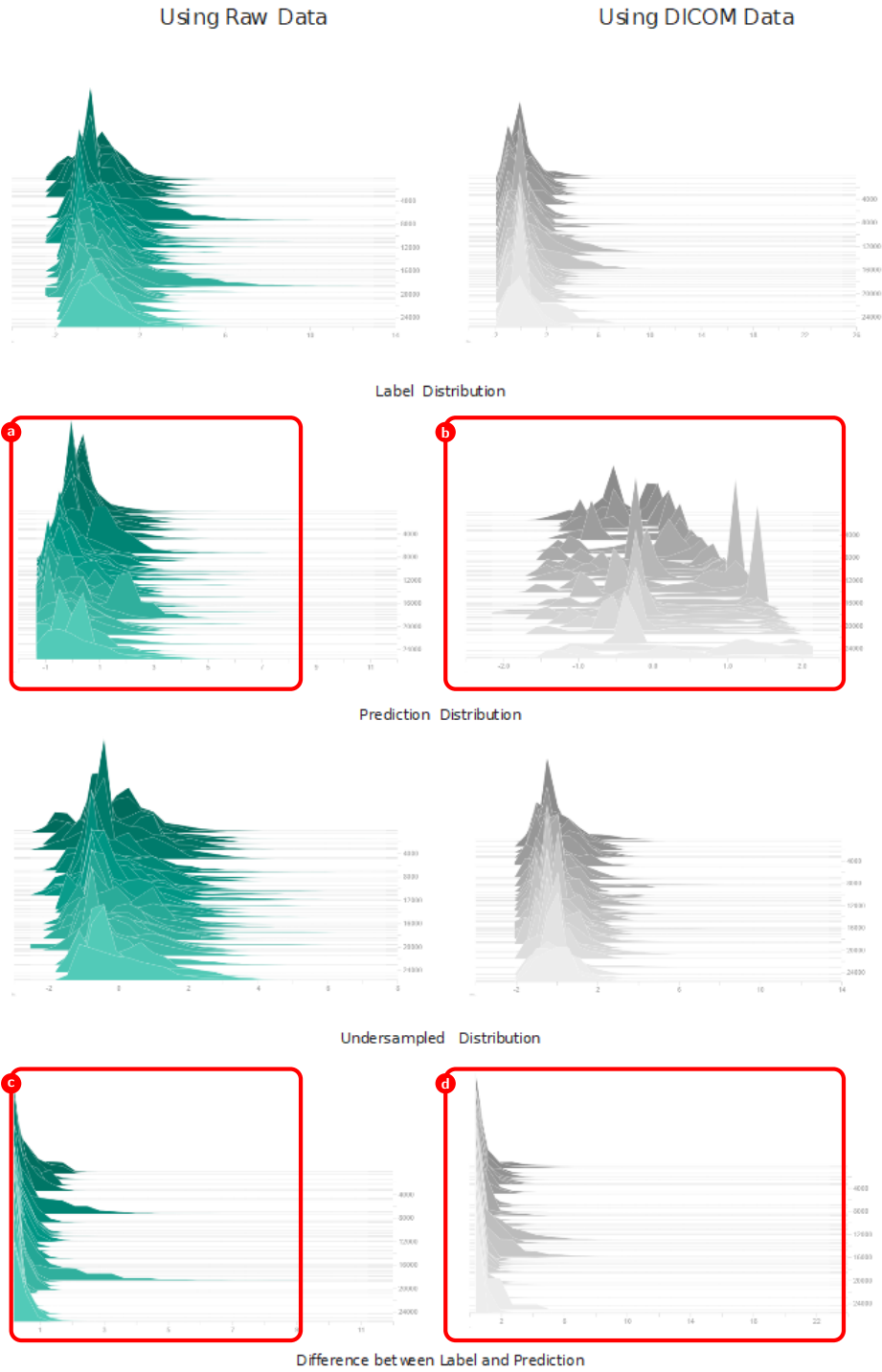
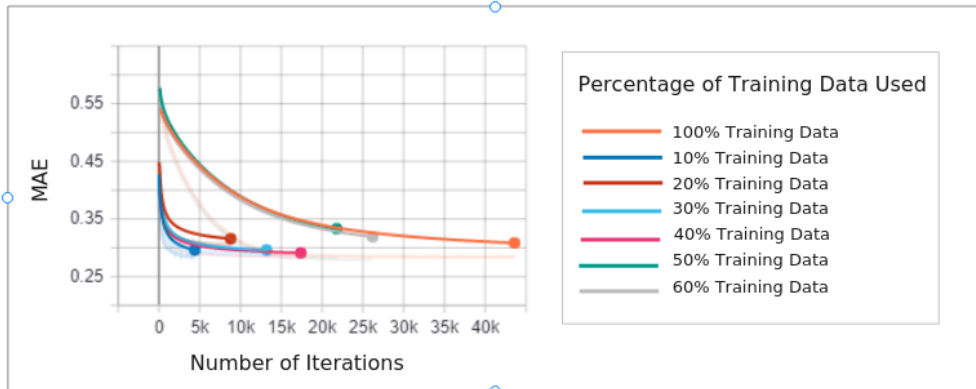


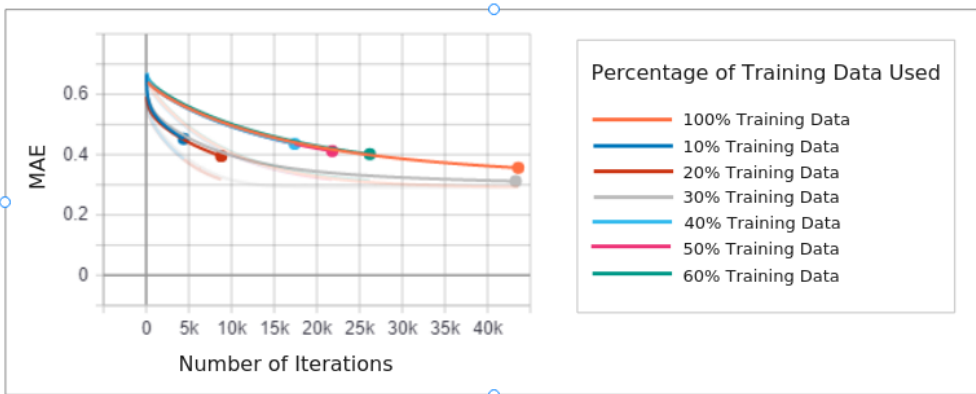
Figure 3.8: Histogram for using 60% of data in training
 A comparison of histograms of the training images using 60% of training data for both original k-space and DICOM k-space data.



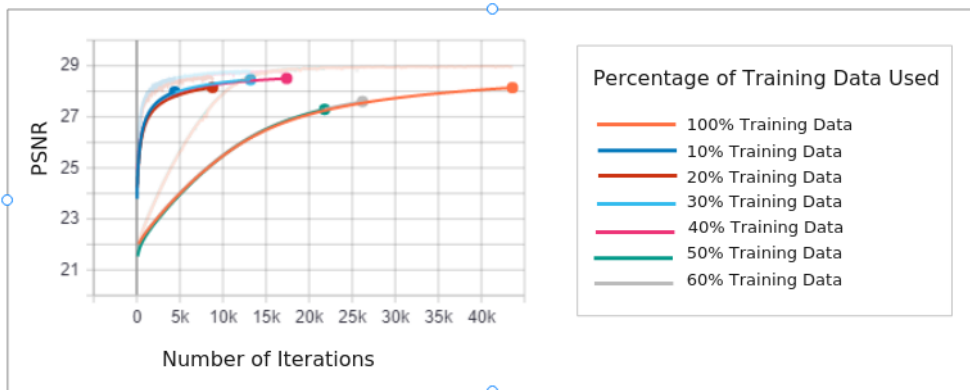
Figure 3.9: Histogram for using 100% of data in training
 A comparison of histograms of the training images using 100% of training data for both original k-space and DICOM k-space data.



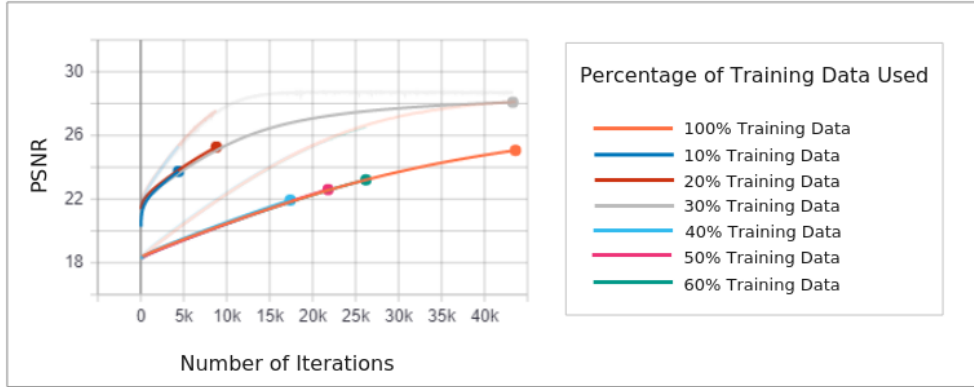
(a) MAE using the Original k-space Data



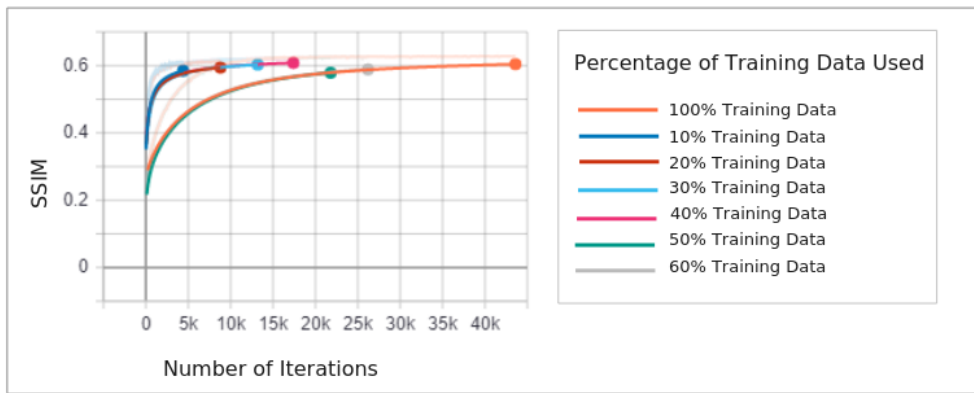
(b) MAE using the DICOM k-space Data



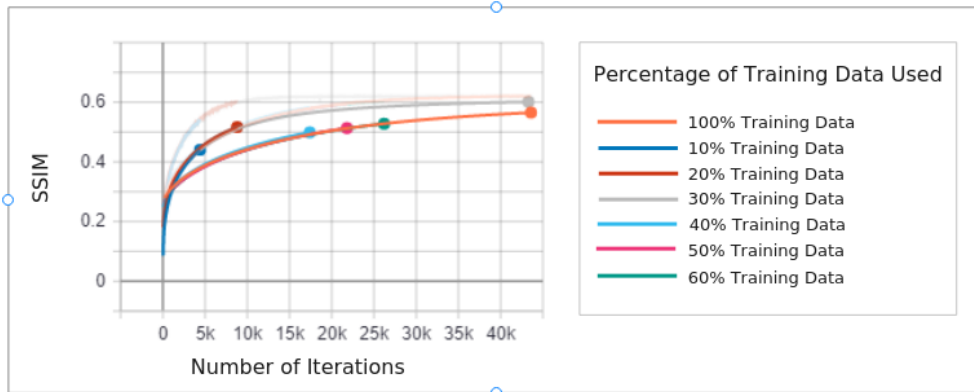
(c) PSNR using the Original k-space Data



(d) PSNR using the DICOM k-space Data



(e) SSIM using the Original k-space Data



(f) SSIM using the DICOM k-space Data

Figure 3.9: Learning Curves during training

This figure shows the learning curves during training. (a) and (b) shows the loss MAE during the training. (c) and (d) shows the metric PSNR (e) and (f) shows the metric SSIM. (a), (c) and (e) are related to the original k-space data. (b), (d) and (f) are related DICOM k-space data.

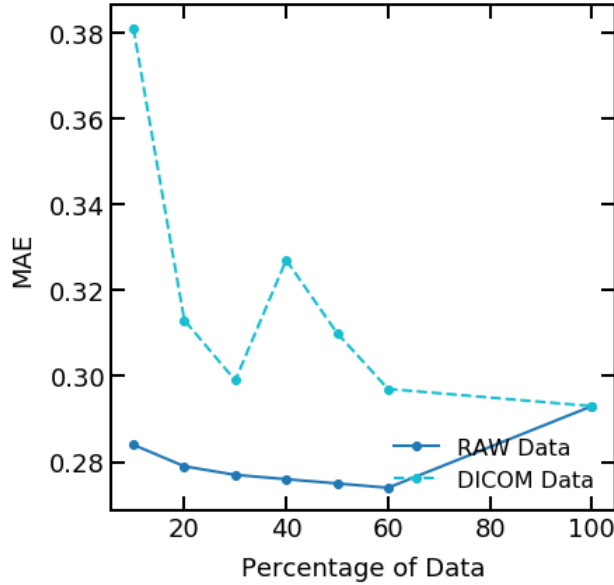


Figure 3.10: Plot of MAE comparing Runs

The plot above compares the training loss during inference using the models trained using Original k-space Data(solid line) and DICOM k-space data(dashed line). The loss is plotted against various percentages of training data used. The values for this plot is obtained from the tables 3.1 and 3.2

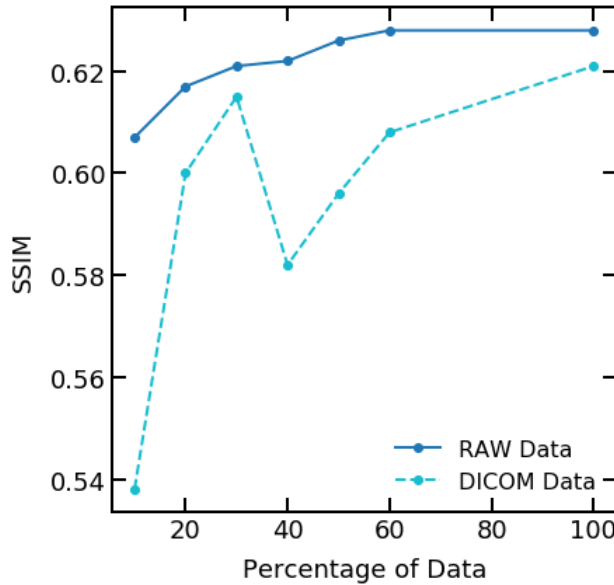


Figure 3.11: Plot of SSIM comparing Runs

The plot above compares the training loss during inference using the models trained using Original k-space Data(solid line) and DICOM k-space data(dashed line). The metric SSIM is plotted against various percentages of training data used. The values for this plot is obtained from the tables 3.1 and 3.2

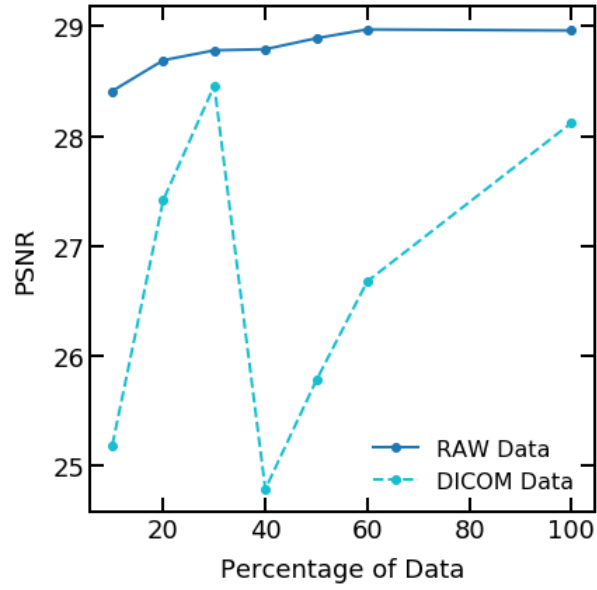


Figure 3.12: Plot of PSNR comparing Runs

The plot above compares the training loss during inference using the models trained using Original k-space Data(solid line) and DICOM k-space data(dashed line). The metric PSNR is plotted against various percentages of training data used. The values for this plot is obtained from the tables 3.1 and 3.2

4. Discussion

The work aims to compare a deep learning training network’s performance to perform image reconstruction of an undersampled MRI k-space data. For this task, a relatively simple network based on UNet architecture was chosen. It has comparable performance to the fastMRI challenge leaderboard for a similar network²⁰. The SSIM metric for Single-Coil knee for an acceleration of 4x on the leaderboard ranges from 0.78 to 0.60. SSIM for the presented network using original K-Space data is 0.63, and it is 0.62 using the DICOM k-space data. MRI DICOM data is abundant. But its utility is limited for projects involving MRI reconstruction because it has only the magnitude-valued image, unlike the complex-valued images available with the actual k-space data. Realistic k-space data (emulated single-coil data in fastMRI challenge) required less data and achieved faster convergence than synthesized (DICOM) k-space obtained from a magnitude image. The realistic k-space input achieved better SNR after reconstruction as well. However, synthesized k-space data from DICOM images did provide similar performance with more extended training duration and more training data.

MRI signal data is composed of real and imaginary parts. This noise associated with the real and imaginary part is assumed to Gaussian with zero mean. This noise is generally considered to be white noise. In the original k-space data, the Gaussian nature of the noise is preserved. The magnitude images are calculated using the square root of the sum of the square of the real and imaginary parts. This operation is non-linear and resulting noise in the no longer Gaussian in the resulting magnitude image. This change in the noise distribution is carried forward when the magnitude images (DICOM) are converted back to the frequency domain (k-space). Hence, there is a fundamental difference in the distribution of noise in original k-space data and synthesized or pseudo k-space data obtained using the magnitude image. The magnitude images have a Rician distribution, which is approximately a Gaussian distribution when the SNR is greater than 2^{21} . The

background noise is a special case of Rician distribution called Rayleigh distribution. It can be seen in the histograms in figures [3.3 - 3.9]. It was easy and efficient to reconstruct the undersampled magnitude image from original k-space even with as little as 10 percent of the data and 5000 iterations. In the case of the synthesized DICOM k-space data, the variation of the distribution between the undersampled magnitude image and the ground truth magnitude image was large, and training takes more data. From figure 3.8, we can observe that the distribution with even 60% of training data DICOM k-space does not achieve a good performance. The images shown in figure 3.1 indicate that the network does not learn the absolute intensities correctly and there are residual differences near the edges of the anatomy, even for using 60% of the training data. It can also be observed as the training data increased, and the number of iterations is more than 45,000, the performance is similar to the one obtained using the original k-space data as show in figures 3.9, 3.2. This establishes the potential utility of using DICOM images for creating synthetic k-space for retrospective undersampling and training networks for MRI reconstruction.

5. Conclusion

From figures 3.10, 3.11, 3.12, we can conclude that realistic k-space data offers a better training performance for undersampled MRI reconstruction compared to using synthesized k-space data from magnitude (DICOM) images^{3,22,23}. But DICOM data is available in abundance. It is important to explore its utility. In the current work, only one acceleration factor and one central sampling region is considered. Further experiments are needed with various permutations of acceleration factors and the amount of central region to get a full comparison of training MRI reconstruction networks using DICOM images. We have also used a simple U-Net architecture. This comparison has to be explored with other advanced network architectures like KIKI-net, Cascade-net, PD-net, and variational networks¹⁰. Nvidia Clara’s AutoML feature can be used to find the optimal network architecture for a chosen network. A certain selection of hyperparameters might help the training, particularly for the more challenging cases when using less synthesized k-space data. Furthermore, recent studies in image denoising and super resolution have pointed out the fact that U-net suffers from the degradation problem and may not be an ideal architecture to achieve the optimal performance w.r.t reconstruction. Variants like U-net with group normalization, Residual U-net and Dense U-net could be explored²⁴. U-net family of networks are primarily used for segmentation²⁵, some of these networks could be fashioned to work for reconstruction and explored for their performance with k-space synthesized from DICOM images.

Bibliography

- [1] Rebecca Smith-Bindman, Marilyn L. Kwan, Emily C. Marlow, Mary Kay Theis, Wesley Bolch, Stephanie Y. Cheng, Erin J. A. Bowles, James R. Duncan, Robert T. Greenlee, Lawrence H. Kushi, Jason D. Pole, Alanna K. Rahm, Natasha K. Stout, Sheila Weinmann, and Diana L. Miglioretti. Trends in Use of Medical Imaging in US Health Care Systems and in Ontario, Canada, 2000-2016. 322(9):843. ISSN 0098-7484. doi: 10.1001/jama.2019.11456. URL <https://jamanetwork.com/journals/jama/fullarticle/2749213>. iv, 1
- [2] Sigpy.mri.poisson — sigpy 0.1.21 documentation. URL <https://sigpy.readthedocs.io/en/latest/generated/sigpy.mri.poisson.html>. v, 7, 16
- [3] Kieren Grant Hollingsworth. Reducing acquisition time in clinical MRI by data undersampling and compressed sensing reconstruction. page 27. v, 37
- [4] R Mezrich. A perspective on K-space. 195(2):297–315. ISSN 0033-8419, 1527-1315. doi: 10.1148/radiology.195.2.7724743. URL <http://pubs.rsna.org/doi/10.1148/radiology.195.2.7724743>. 1
- [5] 11_10pu.pdf. URL http://dicom.nema.org/Dicom/2011/11_10pu.pdf. 1
- [6] Christoph Forman, Jens Wetzl, Carmel Hayes, and Michaela Schmidt. Compressed Sensing: A Paradigm Shift in MRI. page 6. 2
- [7] mahdi Khosravy, editor. *Compressive Sensing in Healthcare*. Elsevier. ISBN 978-0-12-821247-9. 2
- [8] Jasper Schoormans, Gustav J Strijkers, Anders C Hansen, Aart J Nederveen, and Bram F Coolen. Compressed sensing MRI with variable density averaging (CS-VDA) outperforms full sampling at low SNR. 65(4):045004. ISSN 1361-6560. doi:

10.1088/1361-6560/ab63b7. URL

<https://iopscience.iop.org/article/10.1088/1361-6560/ab63b7>. 3

- [9] Nicholas Dwork, Corey Baron, Ethan Johnson, Daniel O'Connor, John Pauly, and Peder Larson. *Fast Variable Density Poisson-Disc Sample Generation with Directional Variation*. 4
- [10] Zaccharie Ramzi, Philippe Ciuciu, and Jean-Luc Starck. Benchmarking MRI Reconstruction Neural Networks on Large Public Datasets. 10(5):1816. ISSN 2076-3417. doi: 10.3390/app10051816. URL <https://www.mdpi.com/2076-3417/10/5/1816>. 4, 8, 37
- [11] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. URL <http://arxiv.org/abs/1505.04597>. 4, 8
- [12] Jure Zbontar, Florian Knoll, Anuroop Sriram, Tullie Murrell, Zhengnan Huang, Matthew J. Muckley, Aaron Defazio, Ruben Stern, Patricia Johnson, Mary Bruno, Marc Parente, Krzysztof J. Geras, Joe Katsnelson, Hersh Chandarana, Zizhao Zhang, Michal Drozdal, Adriana Romero, Michael Rabbat, Pascal Vincent, Nafissa Yakubova, James Pinkerton, Duo Wang, Erich Owens, C. Lawrence Zitnick, Michael P. Recht, Daniel K. Sodickson, and Yvonne W. Lui. fastMRI: An Open Dataset and Benchmarks for Accelerated MRI. URL <http://arxiv.org/abs/1811.08839>. 5, 6, 11
- [13] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance Normalization: The Missing Ingredient for Fast Stylization. URL <http://arxiv.org/abs/1607.08022>. 9
- [14] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. page 8. 9
- [15] Dgx2-user-guide.pdf. URL <https://docs.nvidia.com/dgx/pdf/dgx2-user-guide.pdf>. 11

- [16] Alexander Sergeev and Mike Del Balso. Horovod: Fast and easy distributed deep learning in TensorFlow. URL <http://arxiv.org/abs/1802.05799>. 11
- [17] Alain Hore and Djemel Ziou. Image Quality Metrics: PSNR vs. SSIM. In *2010 20th International Conference on Pattern Recognition*, pages 2366–2369. IEEE. ISBN 978-1-4244-7542-1. doi: 10.1109/ICPR.2010.579. URL <http://ieeexplore.ieee.org/document/5596999/>. 12
- [18] Z. Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image Quality Assessment: From Error Visibility to Structural Similarity. 13(4):600–612. ISSN 1057-7149. doi: 10.1109/TIP.2003.819861. URL <http://ieeexplore.ieee.org/document/1284395/>. 13
- [19] Liyan Sun, Zhiwen Fan, Xinghao Ding, Yue Huang, and John Paisley. Region-of-interest undersampled MRI reconstruction: A deep convolutional neural network approach. 63:185–192. ISSN 0730725X. doi: 10.1016/j.mri.2019.07.010. URL <https://linkinghub.elsevier.com/retrieve/pii/S0730725X19301870>. 20
- [20] FastMRI. URL <https://fastmri.org/>. 35
- [21] HáKon Gudbjartsson and Samuel Patz. The rician distribution of noisy mri data. 34 (6):910–914. ISSN 07403194, 15222594. doi: 10.1002/mrm.1910340618. URL <http://doi.wiley.com/10.1002/mrm.1910340618>. 35
- [22] Martin Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mane, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viegas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin

- Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems. page 19. 37
- [23] James F. Glockner, Houchun H. Hu, David W. Stanley, Lisa Angelos, and Kevin King. Parallel MR Imaging: A User's Guide. 25(5):1279–1297. ISSN 0271-5333, 1527-1323. doi: 10.1148/rg.255045202. URL <http://pubs.rsna.org/doi/10.1148/rg.255045202>. 37
- [24] Pak Lun Kevin Ding, Zhiqiang Li, Yuxiang Zhou, and Baoxin Li. Deep Residual Dense U-Net for Resolution Enhancement in Accelerated MRI Acquisition. URL <http://arxiv.org/abs/2001.04488>. 37
- [25] Shawn Wang. ShawnBIT/UNet-family. URL <https://github.com/ShawnBIT/UNet-family>. 37

Publishing Agreement

It is the policy of the University to encourage open access and broad distribution of all theses, dissertations, and manuscripts. The Graduate Division will facilitate the distribution of UCSF theses, dissertations, and manuscripts to the UCSF Library for open access and distribution. UCSF will make such theses, dissertations, and manuscripts accessible to the public and will take reasonable steps to preserve these works in perpetuity.

I hereby grant the non-exclusive, perpetual right to The Regents of the University of California to reproduce, publicly display, distribute, preserve, and publish copies of my thesis, dissertation, or manuscript in any form or media, now existing or later derived, including access online for teaching, research, and public service purposes.

DocuSigned by:

Anil Kemisetti

6A1598152CD34A4...

Author Signature

3/10/2021

Date