2021-08-01

Fast Magnetic Resonance Image Reconstruction With Deep Learning Using An Efficientnet Encoder

Tahsin Rahman
University of Texas at El Paso

FAST MAGNETIC RESONANCE IMAGE RECONSTRUCTION WITH DEEP LEARNING USING AN EFFICIENTNET ENCODER

TAHSIN RAHMAN

Master's Program in Electrical Engineering

APPROVED:

Sergio Cabrera, Ph.D., Chair

Ali Bilgin, Ph.D., Co-chair

Patricia Nava, Ph.D.

Michael Pokojovy, Ph.D.

Stephen L. Crites, Jr., Ph.D.
Dean of the Graduate School

Dedication

To my wife Sharmin Abdullah, for believing that I could actually do this.

FAST MAGNETIC RESONANCE IMAGE RECONSTRUCTION WITH DEEP LEARNING USING AN EFFICIENTNET ENCODER

by

TAHSIN RAHMAN

THESIS

Presented to the Faculty of the Graduate School of

The University of Texas at El Paso

in Partial Fulfillment

of the Requirements

for the Degree of

MASTER OF SCIENCE

Department of Electrical and Computer Engineering

THE UNIVERSITY OF TEXAS AT EL PASO

August 2021

Acknowledgements

First and foremost, I would like to thank Dr. Sergio Cabrera for his help and advice with this thesis and my graduate research. This work would not have been possible without his unwavering patience and support. I would like to extend my special regards to Dr. Ali Bilgin from the University of Arizona for helping me to navigate the intricate world of MR image processing as well as deep learning, and guiding my research toward new and exciting avenues in this field. His constant feedback was vital to the success of this work. Additionally, I would like to thank my committee members Dr. Patricia Nava and Dr. Michael Pokojovy for their guidance and feedback at various stages of my thesis work. Furthermore, I would like to acknowledge the researchers maintaining the Calgary-Campinas brain MR dataset used in this thesis, in particular Dr. Roberto Souza, for this kind responses to my many queries about the data. My final acknowledgement is to the faculty and staff of the Electrical and Computer Engineering Department at the University of Texas at El Paso who helped me directly or indirectly to make this work possible.

Abstract

This thesis aims to develop an efficient, deep network based method for Magnetic Resonance Imaging (MRI) acceleration through undersampled MR image reconstruction. Deep Neural Networks, particularly Deep Convolutional Networks, have been demonstrated to be highly effective in a wide variety of computer vision tasks, including MRI reconstruction. However, modern highly efficient encoder structures, such as the EfficientNet can potentially reduce reconstruction times further while improving reconstruction quality. To that end, we have developed a multi-channel U-Net MRI reconstruction network which uses an EfficientNet encoder and a custom asymmetric. The network was trained and tested using 5x undersampled multi-channel brain MR image data from the Calgary Campinas dataset and was found to outperform comparable traditional U-Net structures in terms of image quality metric analysis and basic visual comparison while achieving a four-fold reduction in inference time.

Table of Contents

List of Figures

Chapter 1: Introduction

Magnetic resonance imaging (MRI) is a medical imaging technique that uses magnetic field gradients to generate images of the organs in the body. It is considerably 'safer' than other comparable non-invasive imaging techniques such as X-rays, Computed Tomography (CT) or Positron-Emission Tomography (PET), all of which involve subjecting the patient to potentially harmful radiation.

MRI has a wide range of applications in medical diagnosis and is an invaluable tool for neurological, cardiovascular and musculoskeletal imaging. Aside from cost, the biggest downside of MR imaging is the long acquisition time. Since the patient is expected to lie perfectly still during the entire duration of the scan to get the best possible image, 30 minute plus scan times almost guarantee unwanted patient motion which causes motion artifacts in the image. Hardware and software improvements in the last 40 years have constantly sought to bring the acquisition time down, with Parallel MRI bringing in some significant accelerations.

Over the last decade, post processing techniques such as Compressed Sensing (CS), and more recently Deep Learning (DL), combined with Parallel Imaging have been proven to be able to accelerate MR Image acquisition by a factor of 8 [1] or more by reconstructing diagnostic quality images from input data sampled at sub-Nyquist rates. While sparsifying transform based CS methods for MRI reconstruction are more interpretable and can operate with only a small amount of data, they require lengthy computation due to the necessity of going through a large number of optimization iterations every time a new output needs to be calculated.

Deep Learning, one of the newest and most promising sub-fields of Machine Learning (ML), on the other hand, can produce outputs much faster than CS alternatives and open up exciting possibilities for real-time image processing, classification and even diagnosis. Over the

last couple of years, the release of large, high quality MRI datasets by medical institutes [2] and new research on improving the interpretability of Deep Networks [3] has sparked a renaissance in DL research for MRI reconstruction.

In this thesis, a U-Net based efficient deep reconstruction network is proposed for fast undersampled MRI reconstruction. The output of the network is compared with CS and standard U-Net based approaches in terms of widely used image comparison metrics. The organization of the remainder of the thesis is as follows:

In Chapter 2, we take a look at some fundamental DL concepts along with the relevance of DL in image processing tasks. In particular, we study the EfficientNet, a cutting-edge CNN architecture which uses depthwise convolutions in a highly optimized network structure.

In Chapter 3, some basic concepts of MRI imaging and MRI acceleration are explored. Different existing approaches to MRI reconstruction and acceleration are discussed along with the image quality metrics most widely used for MR image quality comparison.

In Chapter 4, we discuss our proposal for an Efficient U-Net optimized for MRI reconstruction and highlight two different approaches for building a U-Net using an EfficientNet encoder. The details of the dataset that we have used for our experiments are also discussed.

Chapter 5 details the experimental work that was performed along with our findings in terms of visual and metric based comparison of the different reconstruction approaches.

Chapter 6 contains concluding remarks and possible avenues of future work.

Chapter 2: Deep Learning and Efficient Architectures in Image Processing

Machine Learning and Deep Learning are becoming ubiquitous terms in nearly every data driven research area today. This entire field is concerned with developing elaborate algorithms that can ‘learn’ how to solve problems in a way similar to how humans can learn to become proficient at certain tasks. In their simplest form, ML algorithms, or ‘models’ as they are frequently called, can classify data or predict new target values once they have been adequately ‘trained’ to detect important features in similar datasets. Artificial Neural Networks (ANNs) take this one step further by having the model itself decide which features in the data are important along with how to detect them. This is particularly helpful for input data such as images for which manually handcrafting features is prohibitively difficult for anything but the simplest examples. Deep Learning is essentially the use of ‘deep’ ANNs where the depth is a measure of the number of layers of processing units, or neurons, inside the network. Deep ANNs, particularly Deep Convolutional Networks, are now at the forefront of performance in most ML tasks, especially in computer vision and image processing, where they are unparalleled [4].

In this chapter, we take a look at the basic building blocks of Deep Neural networks, the relevance of Deep Convolutional Networks in modern computer vision and image processing, as well as the current interest in building lightweight and computationally efficient deep networks. As a leading example of an efficient architecture, we study the EfficientNet [5] and take a look at current efforts to modify the network for regression tasks such a MRI reconstruction.

2.1 Neural Networks and Convolutional Neural Networks in Image Processing

2.1.1 Machine Learning and Supervised Learning

A big part of statistical analysis of data is to derive decisions or patterns based on a given set of data points. With the gradual rise in the volume and complexity of data over the last century, ML approaches for data analysis, where a ‘machine’ (essentially numerical algorithms or models) is tasked with analyzing data and deriving or learning decision-making capabilities to achieve certain tasks have become more and more popular. In essence, ML revolves around the development and study of methods that give computers the ability to solve problems by learning from past experience. The goal is to create mathematical models that can be trained to produce useful outputs when fed input data. Machine learning models are provided with training data and are tuned to produce accurate predictions for these data using an optimization algorithm. The learning process allows for generalization of the expertise of the model so that it may be used to deliver correct predictions for new, unseen data.

There are several kinds of ML, categorized according to how the models utilize their input data during training. In reinforcement learning, the model learns from its environment through trial and error while optimizing some objective function. In unsupervised learning, such as clustering analysis, the computer is tasked with uncovering patterns in the data without expert guidance. However, the most popular ML systems today belong to the class of supervised learning. Here, the computer is given a set of already labeled or annotated data, and asked to produce correct labels on new, previously unseen data sets based on the rules discovered in the labeled data set. Commonly used supervised ML algorithms include linear or logistic regression, random forest,

support vector machines, and ANNs, of which ANNs are recently experiencing a surge of popularity, particularly for image processing tasks.

2.1.2 Artificial Neural Networks

In traditional Machine Learning techniques, most of the applied features need to be identified by a domain expert in order to reduce the complexity of the data and make patterns more visible to learning algorithms for them to work. ANNs [6] [7] were introduced as a ML approach where this feature extraction step is completely performed by the learning algorithm. This significantly reduces or even eliminates the need of domain expertise and manual feature extraction.

In broad strokes, a neural network consists of a number of connected computational units, called neurons, arranged in layers. In the basic form of a neural network, which is the feedforward neural network as shown in Figure 1, there is an input layer where data enters the network, followed by one or more internal or 'hidden' layers transforming the data as it flows through, before ending at an output layer that produces the neural network's predictions. In a traditionally defined fully-connected neural network, each neuron in a particular layer will receive the output from each neuron in the preceding layer and its own output will be propagated to every neuron in the following layer.

The network is trained to output useful predictions by identifying patterns in a set of labeled training data, fed through the network while the outputs are compared with the actual labels by an objective function. During the training phase, the strength of each neuron, i.e. is the factor with which a neuron multiplies its input and is commonly referred to as its weight, is uncalibrated until the patterns identified by the network result in good predictions for the training data. Once the patterns are learned, the network can be used to make predictions on new, unseen data.

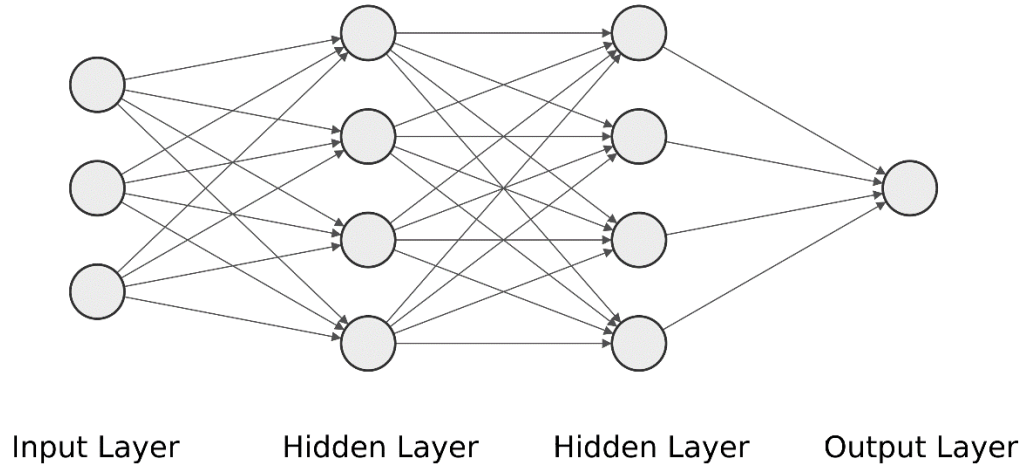


Figure 1: A Feedforward Neural Network

Each neuron inside a neural network takes a bias w_0 and a weight vector $w = (w_1, \ldots, w_n)$ as parameters to model a decision based on the input x using a non-linear activation function $h(x)$ (Eqn. 1).

$$\hat{f}(\boldsymbol{x}) = h(\boldsymbol{w}^T \boldsymbol{x} + w_0) \tag{1}$$

As classically defined, an activation function is chosen such that it is monotonic, bounded, and continuous. In this case, the maximum and the minimum can be interpreted as a decision for the one or the other class. Typical representatives for such activation functions in classical literature are the signum function $sign(x)$ resulting in Rosenblatt's perceptron, the sigmoid function (Eqn. 2) and the tangens hyperbolicus (Eqn. 3).

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{2}$$

$$tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{3}$$

Individual neurons, however, can only model linear decision boundaries, an issue widely known as the XOR problem. The problem can be overcome by using multiple neurons. With only a single layer of neurons, neural networks can approximate any continuous function $f(\boldsymbol{x})$ on a compact subset of $\mathbb{R}^n$ [8]. A single layer network is conveniently summarized as a linear combination of N individual neurons using combination weights v_i (Eqn. 4).

$$\hat{f}(\boldsymbol{x}) = \sum_{i=0}^{N-1} v_i h\left(\boldsymbol{w}_i^T \boldsymbol{x} + w_{0,i}\right) \tag{4}$$

All trainable parameters of this network can be summarized as $\boldsymbol{\theta}$ (Eqn. 5).

$$\boldsymbol{\theta} = \left(v_0, w_{0,0}, \boldsymbol{w}_0, \dots, v_N, w_{0,N}, \boldsymbol{w}_N\right) \tag{5}$$

The predictions of the network will only be valid for samples that are drawn from the same distribution as the compact set on which the network was trained. Therefore, an additional practical requirement for an approximation is that the training set is representative and future observations will be similar.

During the training phase, the training data set is processed, and meaningful features are extracted. To do so, a training data point (or, typically, a small batch of training points) is fed to the network, the outputs and local derivatives at each node are recorded, and the difference between the output prediction and the true label is measured by an objective function, such as mean absolute error (L_1 norm), mean squared error (squared L_2 norm), or cross-entropy loss, depending on the application. The derivative of the objective function with respect to the weights is calculated and used as a feedback signal. For networks with multiple hidden layers, the discrepancy is propagated backwards through the network and all the weights are updated to reduce the error. This is achieved using backward propagation, which calculates the gradient of the

objective function with respect to the weights in each node using the chain rule together with gradient descent.

Multiple training runs are performed with different initialization techniques in order to estimate a mean and a standard deviation for the model performance. Furthermore, it is very common to use typical regularization terms on parameters, such as L_1 and L_2 regularization or techniques such as Dropout. Dropout is an averaging method based on stochastic sampling of neural networks. By randomly removing neurons during training slightly different networks are used for each batch of training data, and the weights of the trained network are tuned based on optimization of multiple variations of the network.

In addition to the training data set, a validation set is used to safeguard against over-fitting. In contrast to the training set, the validation set is never used to directly update the parameter weights. Hence, the loss of the validation set allows an estimate for the error on unseen data. Hyper-parameter tuning, which refers to the typically user-defined parameters of the network such as learning rate and training batch size, has to be done on validation data before actual test data is employed. In principle, test data should only be looked at once architecture, parameters, and all other factors of influence are fixed. Only then the test data is to be used. Otherwise, repeated testing will lead to overly optimistic results and the system's performance will be over-estimated. During optimization, the loss on the training set will continuously fall. However, as the validation set is independent, the loss on the validation set will increase at some point in training. This is typically a good point to stop updating the model before it overfits to the training data.

2.1.3 Deep Neural Networks and Deep Learning

Typically, ANNs with more than one hidden layer of neurons are referred to as Deep Networks. While a neural network containing a single hidden layer can, in theory, approximate any function, many researchers, most notably Goodfellow et al. [9], empirically demonstrated that greater network depth, i.e., having multiple hidden layers, leads to better generalization for a wide variety of tasks. As the number of hidden layers in a network is increased, however, exponentially more resources in terms of memory and computation ability are required to perform all the operations required to train the network and more training data is required to obtain sufficient convergence of the weight values.

The rapid spread of the application of Deep Learning to virtually every applicable ML problem is partly due to the present widespread availability of GPU-based compute resources and partly due to the proliferation of the open-source software movement in the cutting-edge computing community. Computer vision challenges in particular have attracted the use of deep networks in recent decades due to the availability of datasets like the ImageNet which allow for the training of very deep networks. In case of medical imaging, privacy and patient confidentiality issues have historically restricted open access to the huge amounts of labelled data required to train deep networks. Thanks to the release of large, anonymized, medical imaging datasets in recent years, Deep Learning has suddenly become viable in this field.

Aside from requiring very large datasets, DL implementations tend to use novel, unbounded activation functions in the hidden layers, such as the immensely popular Rectified Linear Unit (ReLU) (Eqn. 6), the Leaky ReLU (LReLU) ($\alpha = 0.01$), or the Parametric ReLU (PReLU), in which α is a learnable parameter (Eqn 7).

$$ReLU(x) = \begin{cases} x \ \ if \ \ x > 0 \\ 0 \ \ otherwise, \end{cases} \tag{6}$$

$$PReLU(x)/LReLU(x) = \begin{cases} x \ \ if \ \ x > 0 \\ \alpha x \ \ otherwise, \end{cases} \tag{7}$$

More recently, automated search techniques have been used to discover new, more effective activation functions, such as the Swish [10] (Eqn. 8), which in its simplest form ($\beta = 1$) is a sigmoid-weighted linear unit.

$$Swish(x) = x\sigma(\beta x) \tag{8}$$

Contrary to the classical activation functions, many of the new activation functions are convex and have large areas with non-zero derivatives. As the computation of the gradient of deeper layers using the chain rule requires several multiplications of partial derivatives, the deeper the net, the more multiplications are required. If several elements along this chain are smaller than 1, the entire gradient decays exponentially with the number of layers. Hence, non-saturating derivatives are important to solve numerical issues, which were historically the reasons why vanishing gradients [11] did not allow training of networks that were too deep.

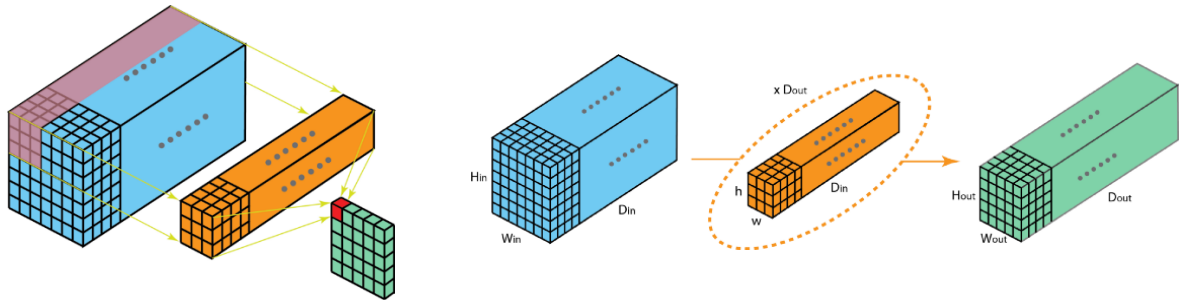
2.1.4 Convolutional Neural Networks

Convolutional Neural Networks or CNNs [12], employ the mathematical convolutional operation in order to more effectively process data that has a known, grid-like topology, such as image data which can be thought of as a 2D grid of pixels. Convolutional Networks are typically made to have sparse connections by making the convolution kernel much smaller than the feature map. As a result, each neuron in a layer is only affected by (or connected to) a small set of neighboring neurons in the previous layer, drastically reducing the memory footprint of the network. This is particularly important for input data such as images which might have thousands or millions of pixels being fed into the input layer. For most practical purposes, a deep, fully-connected network for large images would currently be prohibitively expensive in term of computational complexity and memory requirements.

CNNs that are applied to image data generally have a very similar basic structure. A typical CNN will have an input layer of neurons that is of the same shape as the input images in terms of resolution (height and width of the image), and the number of channels. Successive layers in the network generally increase the number of channels, which represent the number of feature planes (the depth of the layer) and reduce the spatial resolution. The depth of the network is a measure of the number of layers of neurons in the network.

Figure 2 shows how a typical 2D convolution operation works in a CNN. A parameterized 3D filter comprised of a stack of small 3x3 convolution kernels, which can have different values along the channel dimension, and the same number of channels as the input of the layer (D_{in}) is used to summarize the information across all channels in the receptive field into a single value. The filter is then moved along the spatial dimensions in order to produce a single channel of the output volume. The number of such filters to be used is equal to the required number of output

channels of that layer (D_{out}). Such convolutional layers in a CNN summarize the presence of local features in an input image. By having each filter share the exact same weights across the whole input domain, i.e., translational equivariance at each layer, the number of weights that need to be learned is drastically reduced.

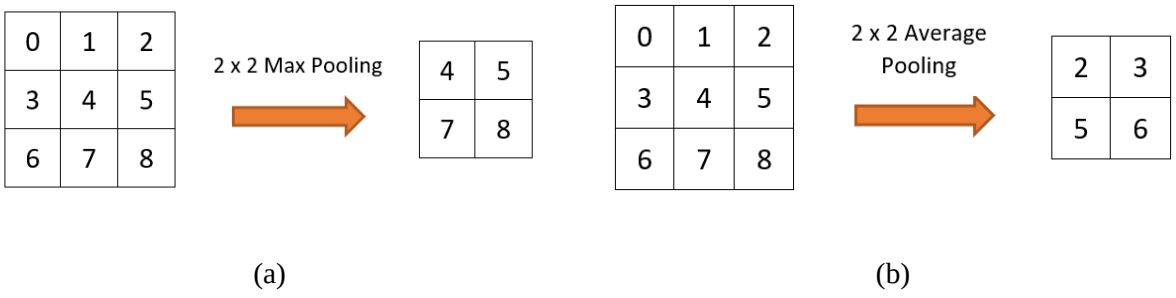


(a) (b)

Figure 2: 2-dimensional Convolution Operation[13]

Another important type of operation in a CNN structure is pooling. Pooling layers provide an approach to down-sampling feature maps by summarizing the presence of features in patches of the feature map. For images, typically areas of 2×2 or 3×3 are analyzed and summarized to a single value. Two common pooling methods are average pooling and max pooling that summarize the average value and the largest value respectively of a feature in the pooling region. The operations are summarized in Figure 3.

CNNs used for classification tasks can also have one or several fully connected layers right before the output in order to optimally classify the features extracted by the convolutional layers. It is possible, however, to replace the terminal fully connected layers with convolutional layers resulting in what is called a Fully Convolutional Network (FCN). FCN structures have been shown to be ideal for tasks such as semantic segmentation [14].



(a) (b)

Figure 3: 2D Max Pooling (a) and Average Pooling (b) Operations

Upsampling or Transposed Convolutions are used in some types of CNNs, particularly for image-to-image translation tasks such as super resolution, in order to increase the spatial size of the image, usually by powers of 2. An upsampling operation, which usually employs nearest neighbor or bilinear upsampling to the input, is typically followed by a convolution since the upsampling layer itself usually has no learnable parameters. A Transposed Convolution uses the transpose of a parameterized convolution kernel to generate the spatially higher dimensional output. Figure 4 shows an intuitive breakdown of the transposed convolution operation.

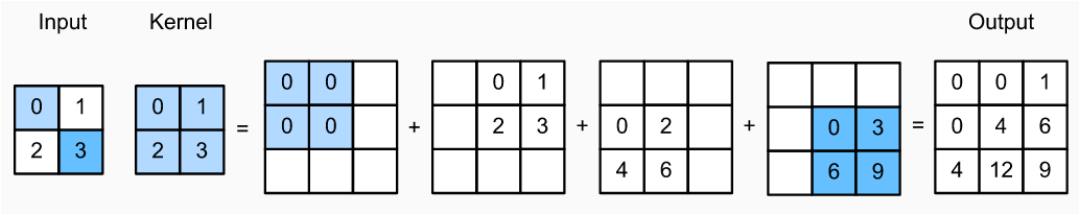


Figure 4: Transposed Convolution Operation [15]

Recently, a number of researchers have revealed the tremendous performance of CNNs on image related tasks. Spurred by the success of AlexNet [16] in the ImageNet ILSVRC challenge in 2012, researchers are now in a constant race to develop the best possible network for general image classification. The 2014 winner of the challenge, GoogleNet [17], proved the merits of the fully convolutional architecture and introduced the inception block resulting in a dramatic

reduction in the number of parameters of the network. The 2014 runner-up VGGNet [18], demonstrated the importance of network depth by having a deep but simple and homogeneous architecture that is still used as a baseline CNN for comparisons today. The ResNet architecture [19], which won the challenge in 2015, popularized the concept of adding skip connections to address the vanishing gradient problem of very deep networks as well as Batch Normalization [20]. In 2016, the ResNeXt architecture [21] incorporated into ResNet the Inception model idea of having branched paths within a block (known as the split-transform-merge strategy [17]). The SENet [22], which won the challenge in 2017, popularized the attention mechanism in mainstream image classification through the Squeeze-and-Excitation block (SE block) mechanism which first pools the input in the spatial dimensions and then squeezes the number of channels by a factor r to obtain a set of channel attention weights through fully connected layers.

Although the official challenge ended in 2017, the ImageNet dataset is still being used as the de facto standard for training and comparing modern CNN classification architectures, many of which are offshoots and improvements over past winning networks. Two of the most popular architectures in use today, ResNeSt [23] and EfficientNet [5], are modern adaptations of the ResNet and GoogleNet, respectively which can achieve state of the art test accuracy while massively improving on the time required for a trained network to produce an output (typically referred to as inference latency) by incorporating techniques such as split-attention and depthwise convolutions. A detailed look at the depthwise convolution technique and how it is used in the EfficientNet is provided later in this chapter when we review the EfficientNet architecture.

2.2 U-Net

CNNs have also been employed widely for image-to-image translation tasks. Different from those for image classification, a CNN in an image translation task outputs an image that has a one-to-one pixel correspondence with the input. One of the common uses of such CNNs is for image segmentation, where the output is the segmented region maps. The most popular architecture designed for this task is the fully convolutional U-net [24], and its 3D variant, the V-net [25], which for the first time introduced the Dice loss layer widely incorporated nowadays.

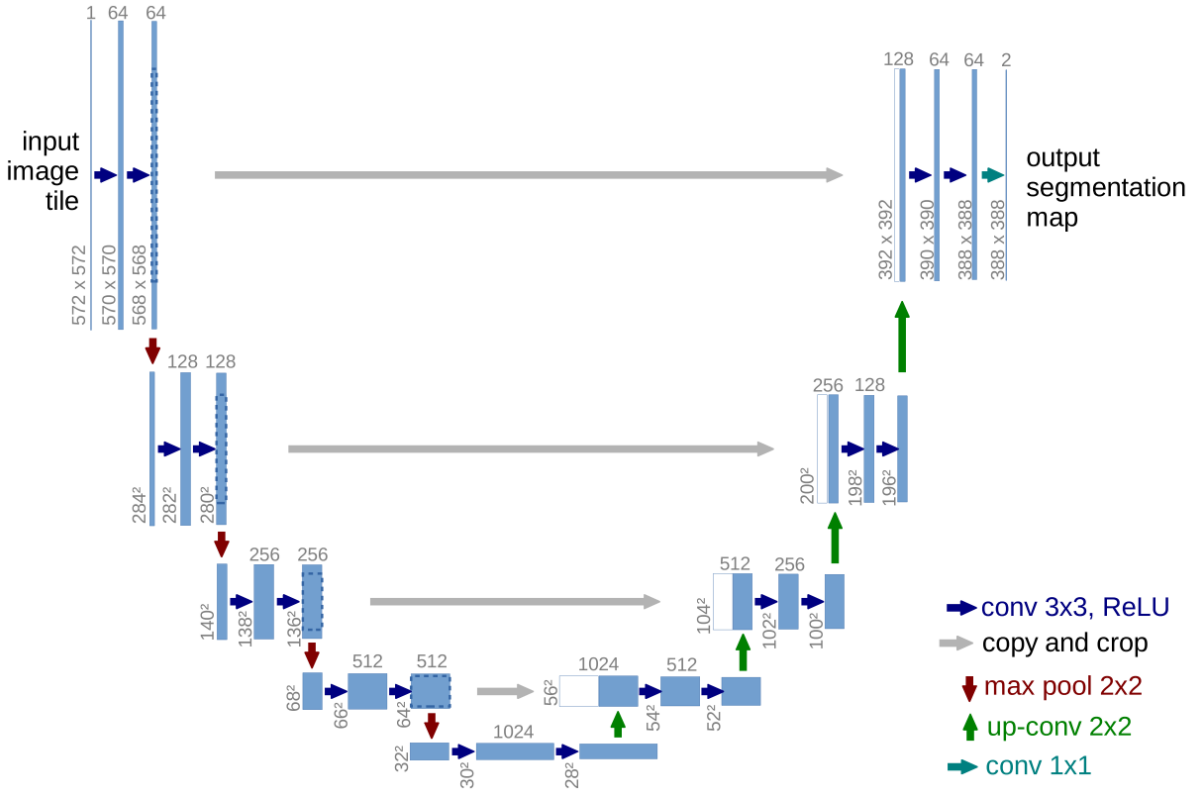


Figure 5: Ronneberger's U-Net [24]

The original structure of the U-Net architecture as proposed by Ronnerberger is depicted in Figure 5. A typical U-Net consists of a contracting or down-sampling path which follows the

typical encoder architecture of a convolutional network and an expansive or up-sampling path, often referred to as the decoder path. In the originally proposed configuration, the down-sampling path on the left consists of two 3x3 convolutions with ReLU activation, and a 2x2 max pooling operation to halve the spatial dimensions. At each down-sampling step the number of feature channels is doubled. Every step in the decoder path consists of an upsampling of the feature map to double the spatial resolution followed by a 2x2 convolution. The upsampled output is then concatenated with the correspondingly cropped feature map from the encoder path through a skip connection, followed by two 3x3 convolutions with ReLU activation. At the final layer, a 1x1 convolution is used to reduce the number of channels to the number of channels in the segmentation map without changing the spatial resolution.

Because of the multiple layers of convolutions at different spatial resolutions in both the encoder and decoder paths, U-net's and V-net's architectures allow effective calculation and combination of both local and global features. The skip connections from the high resolution encoder layers allow fine-grained details to be recovered more effectively [24].

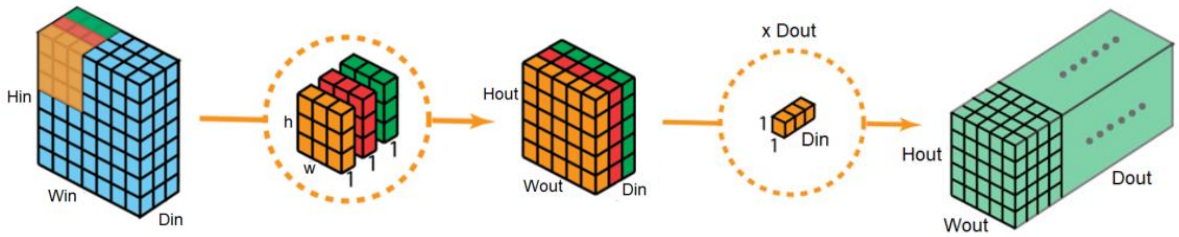
In recent years, many modifications have been made to this basic U-Net structure in order to improve its efficacy, starting from the use of skip connections within the layers and developing asymmetric U-Nets [26] with swappable or even multiple encoder paths [27], to connecting multiple U-Nets in series [28] or parallel [29] with dense skip connections [30] and attention mechanisms [26]. Originally used for image segmentation, the U-Net has now gained immense popularity in other image-to-image translation tasks, one of which is MR image reconstruction.

2.3 Efficient Architectures and EfficientNet

2.3.1 MobileNets

There has been rising interest in building small and efficient neural networks in the recent literature [31][32][33], primarily from the growing need to deploy neural networks in low powered edge devices, such as smartphones and IoT nodes. While the training of cutting-edge deep networks still requires the use of massively powerful and parallelized compute clusters, they can be designed to have very low inference latency. Model compression [34], [35], which exploits the fact that deep convolutional networks are often overparameterized, is a common approach to reduce model size in order to improve computational efficiency. Another approach is to directly design networks using computationally efficient blocks and operations. MobileNets [36], is a class of network architectures that allows a model developer to specifically choose a small network that matches the resource restrictions (latency, size) for their application. MobileNets primarily focus on optimizing for latency but also yield small networks.

At the core of the MobileNet architecture is the Depthwise Separable Convolution [37], a form of factorized convolution which factorizes a standard convolution into a depthwise convolution (Figure 6 (a)) and a 1×1 convolution called a pointwise convolution (Figure 6 (b)). In order to perform a depthwise convolution, the 3-dimensional convolution filter is split in the channel direction into a set of 2-dimensional filters each of which acts on a single channel of the input. The outputs for the depthwise convolutions are then stacked together and a 1x1 convolution filter, also with the same number of channels as the input, is used to combine the results of the depthwise convolutions into one channel of the output. The number of feature channels in the final output is determined by the number of filters used for the pointwise convolution. This factorization leads to drastically reduced computation complexity and model size [36].



(a) (b)

Figure 6: Depthwise Separable Convolution [13]

While the first proposed MobileNet was handcrafted, subsequent iterations have taken advantage of Network Architecture Search [38] and the idea has been expanded to other types of networks such as the MnasNet [39] which uses a factorized hierarchical search space that factorizes a CNN model into groups of unique operations called 'blocks', and then searches for the operations and connections per block separately. This acts to balance the diversity of layers and the size of the total search space and allowing for different layer architectures in different blocks.

2.3.2 EfficientNet

One of the latest approaches to build highly efficient networks is the EfficientNet [5]. It is the result of a multi-objective factorized neural architecture search similar to the one performed for MnasNet that optimizes both accuracy and FLOPS. Its main building block is the mobile inverted residual bottleneck MBConv [5], [40], to which squeeze-and-excitation (SE) optimization [22] is added.

The MBConv Block shown in Figure 7, which is the basic building block of the EfficientNet, consists of an initial expansion convolution which increases the number of input feature channels by a predefined factor e, followed by a Depthwise Separable Convolution, each of which is accompanied by a Batch Normalization layer and a Swish activation layer. Following the activation layer is the SE Block which serves as a channel attention module for the expanded

feature channels. In the SE block, average pooling is first performed in the spatial dimensions to reduce them to 1x1. Two 1x1 convolution layers are then used in succession, the first of which squeezes, or reduces the number of channels by a factor r, while the second expands the number of channels to the original value. The resultant set of channel weights are then multiplied with the input of the SE block and a final convolution (followed by Batch Normalization) is performed in order to produce the output of the MBConv block. For repeated MBConv blocks, in which case the number of input and output channels of the block remain the same, the input is added to the final output through a skip connection.

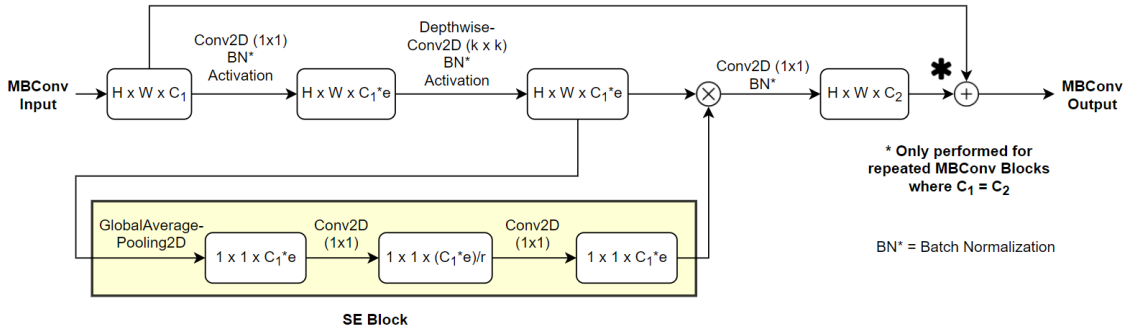


Figure 7: Diagram Showing the Internal Layers of the MBConv Block

What makes the EfficientNet so powerful is the way the MBConv blocks are combined and how the parameters of each block, such as the expansion ratios, output channel sizes and convolution kernel sizes, are tuned to achieve the best possible performance. The optimal set of parameter values were obtained as result of a network architecture optimization search focusing on efficiency and performance [5]. The network obtained by using the parameter values from the result of this search is called the EfficientNet B0 network. The structure of the EfficientNet B0 is listed in Figure 8.

Here, the number after MBConv denotes the expansion ratio for the expansion convolution and the values after 'k' denote the kernel size for the depthwise convolution. $\hat{C}_i$ is the number of output channels for any layer in that particular stage and $\hat{L}_i$ denotes how many times a layer in each stage is repeated. Instead of using pooling layers for spatial resolution reduction, the EfficientNet utilizes 2x2 strided depthwise convolutions where necessary.

This network can then be further scaled up using the compound scaling method proposed by the authors in order to create a class of EfficientNets from B0 to B7. This method uniformly scales the resolution, 'width' (layer depths) and 'depth' (number of layers) using a common compound coefficient φ.

Stage i	Operator $\hat{\mathcal{F}}_i$	Resolution $\hat{H}_i \times \hat{W}_i$	#Channels $\hat{C}_i$	#Layers $\hat{L}_i$
1	Conv3x3	224×224	32	1
2	MBConv1, k3x3	112×112	16	1
3	MBConv6, k3x3	112×112	24	2
4	MBConv6, k5x5	56×56	40	2
5	MBConv6, k3x3	28×28	80	3
6	MBConv6, k5x5	14×14	112	3
7	MBConv6, k5x5	14×14	192	4
8	MBConv6, k3x3	7×7	320	1
9	Conv1x1 & Pooling & FC	7×7	1280	1

Figure 8: Parameters and Structure of EfficientNet B0 [5]

2.3.3 Efficient U-Net for Image Processing

While the main focus of the EfficientNet architecture is to build computationally efficient networks, EfficientNets have also proven themselves to be very powerful CNN encoder structures capable of achieving state-of-the-art accuracy in classification tasks. Researchers have therefore recently started to harness the superb feature resolution capability of the EfficientNet for image-

to-image translation tasks by using the full EfficientNet (without the terminal fully-connected layers) as an encoder in U-Net structures. This Efficient U-Net structure has been successfully used for diverse tasks ranging from cloud image classification [41] to natural environment [42] and blood vessel segmentation [43]. The success of these approaches indicates that just like a regular U-Net, the Efficient U-Net can prove to be effective at regression tasks such as MRI reconstruction while being immensely more conservative in terms of resource usage. The improved inference latency of such a network would be an added benefit for MRI acceleration. The methodology behind developing an Efficient U-Net structure for MRI reconstruction is discussed in detail in Chapter 4.

Chapter 3: MR Imaging and Acceleration

In this chapter, seminal aspects of the MR imaging process relevant to our work are first discussed. The need and basic approach for MRI acceleration is then explained and a literature review of the current state of MRI acceleration using deep CNN based techniques is presented. Finally, the issue of quantitative image quality assessment and comparison in the context of MR images is discussed and relevant metrics explored.

3.1 MR Imaging Process

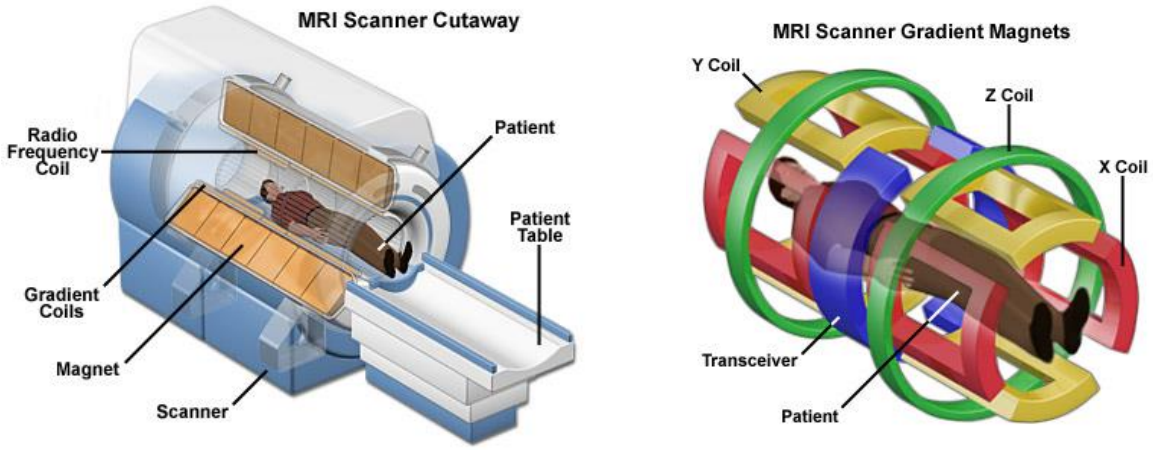


Figure 9: The Fundamental Components of an MRI [44]

In this section, we take a brief look into the MR data acquisition and image formation process. Detailed breakdowns of MR imaging principles and acquisition methods can be found in MRI textbooks, e.g., [45].

Proposed and developed by Paul Lauterbur in the 1970s [46], Magnetic Resonance Imaging is an indirect imaging process which works on biological organisms by creating a strong magnetic field around the body to align the spin axes of the hydrogen nuclei that are present throughout our bodies due to the abundance of water. During a clinical scan, a person is placed inside a large

cylindrical bore which houses all the imaging components including the main magnet which produces the strong homogeneous magnetic field. Additional RF energy, in the form of a sequence of spatially and temporally varying magnetic fields called a "pulse sequence", is then applied using an RF transceiver coil and a set of orthogonal gradient coils to make the aligned hydrogen protons resonate. This can be done along 2D "slices" or planes determined using the gradient coils or across an entire 3D volume at once. When the RF pulse is switched off, the excited protons return to their resting state causing an RF signal to be emitted. This signal is picked up as a set of frequency and phase measurements by the RF transceiver coil. The acquired signal corresponds to points on the multidimensional Fourier-space representation, commonly known as 'k-space', of an imaged body. An inverse Fourier transform is then performed on the acquired k-space data to generate grayscale magnitude images of the anatomy for evaluation by a medical expert.

For many clinical applications, the patient is also injected with a contrast agent with special magnetic properties, such as gadolinium, to intensify signals from certain regions of the anatomy or abnormalities such as tumors. Specific combinations of RF and gradient pulse intensities and timings, along with the usage of different contrast agents, form what are known as MRI sequences. Different MRI sequences are used for highlighting different types of tissue, fluids, or abnormalities, producing high temporal resolution sequences (e.g. in cardiac MRI videos), and even for neural activity tracking through functional MRI (fMRI).

3.2 Parallel Imaging

In parallel MR imaging, multiple receiver coils are used, each of which produces a separate k-space measurement matrix. Each of these matrices is different, since the view each coil provides of the imaged volume is modulated by the differential sensitivity that coil exhibits to MR signal arising from different regions. In other words, each coil measures Fourier components of the imaged volume multiplied by a complex-valued position-dependent coil sensitivity map. The measured k-space signal y_i for coil i in an array of n_c coils is given by

$$y_i = F(S_i m) + \text{noise}$$

Each coil is typically highly sensitive in one region, and its sensitivity falls off significantly in other regions. If the sensitivity maps are known, and the k-space sampling is full (i.e., satisfying the Nyquist sampling condition), then the set of linear relations between m and each y_i defines a linear system that is overdetermined by a factor of n_c. It may be inverted using a pseudoinverse operation to produce a reconstruction of m, as long as the linear system is full rank. The quality of this reconstruction will depend on the measurement noise, since the signal-to-noise ratio is poor in parts of the volume where the coil sensitivity is low.

For most practical purposes, the individual coil images are combined into a single magnitude image using the complex images and their respective sensitivity maps or through a more direct root-sum-of-squares combination of the individual magnitude images.

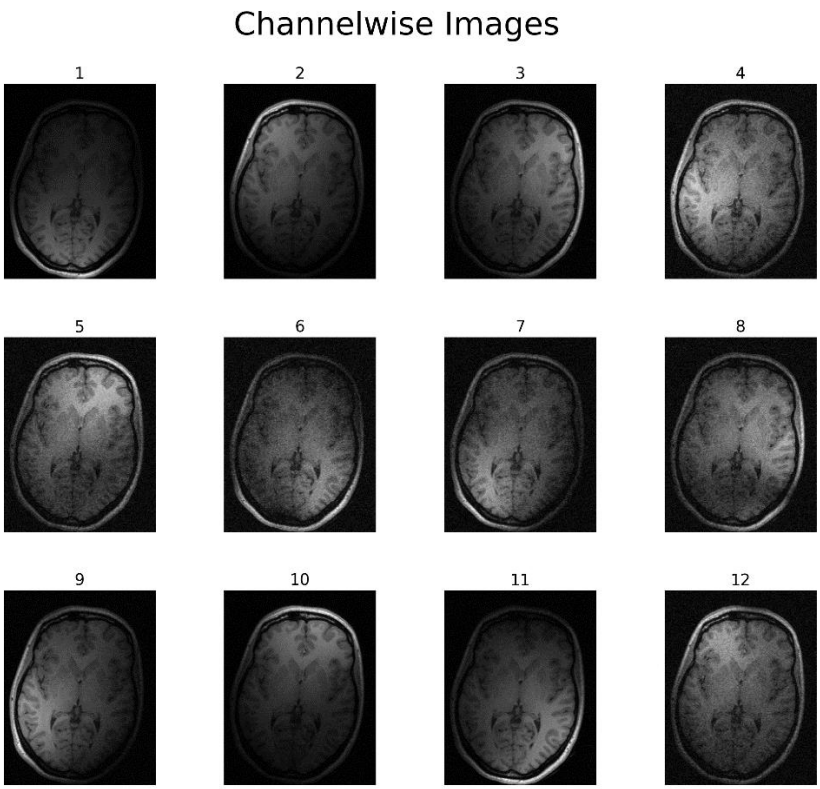


Figure 10: Parallel MRI - Individual Channel Images from a 12 Coil Acquisition

3.3 Sampling in MRI

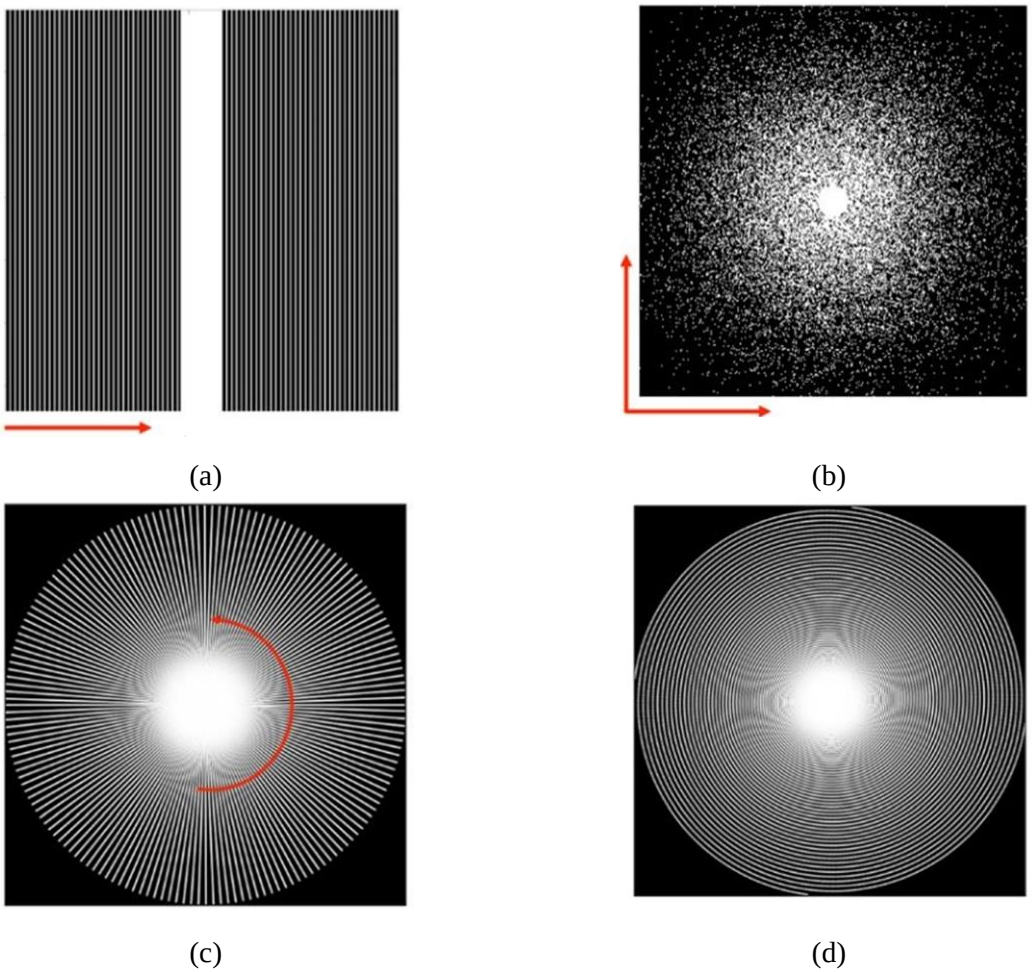


Figure 11: K-space Sampling Patterns (a) Cartesian 1D Uniform (b) Cartesian Poisson Disc (c) Radial (d) Spiral

A variety of different k-space trajectories may be used for MR image acquisition, as shown in Figure 11, with the Cartesian trajectory (sampling points that fall on a rectangular grid) being the most popular in clinical practice. Sampling in the Cartesian trajectory typically takes place along parallel lines of sampling points, but it is also possible to effectively sample random points on the 2D slices if the acquisition is being made in 3D. Some of the other possible trajectories include radial acquisitions, which are less susceptible to motion artifacts than Cartesian trajectories and spiral acquisitions which make the most efficient use of the MRI hardware and are used in

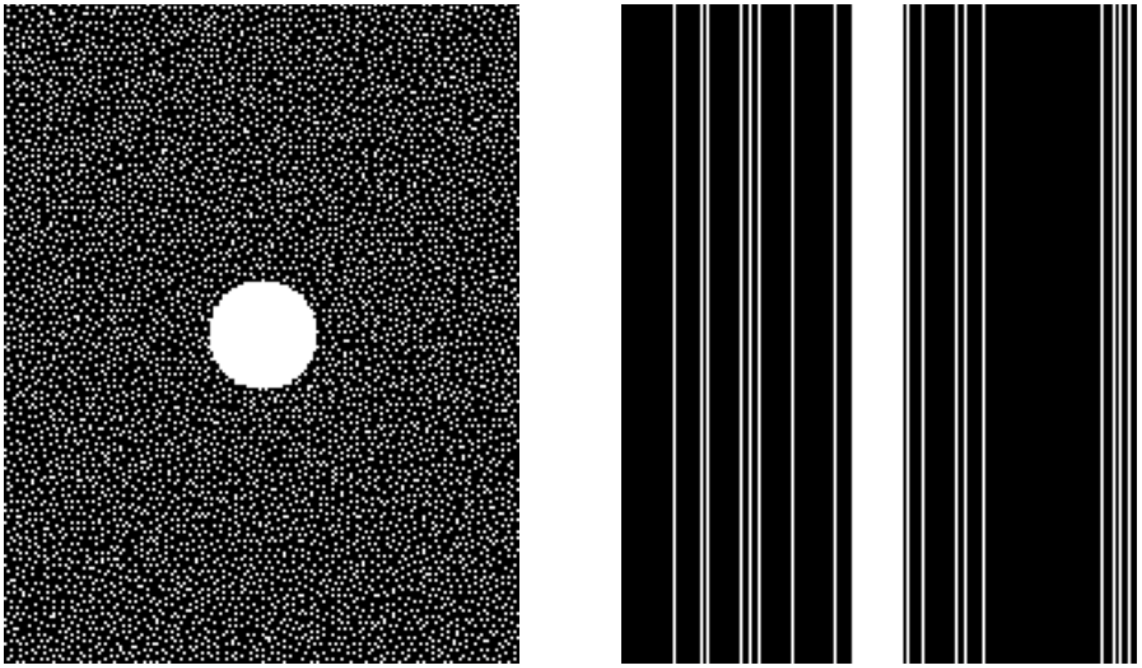
real-time and rapid imaging applications. Efficient reconstruction from such non-Cartesian trajectories requires using filtered back-projection or interpolation schemes (e.g., gridding).

3.4 MRI Acceleration Through Undersampled MRI Reconstruction

MRI is a comparatively slow imaging modality, limited traditionally by Nyquist sampling requirements. The long acquisition times for MR images results in increased motion artifacts in the images, decreased patient comfort, and increased cost of scans which restricts accessibility at a population level. Research on accelerating MRI data acquisition has therefore been an active area of study since the early days of MRI.

One possible approach to accelerating MRI data acquisition is to conduct k-space sampling at a rate below the Nyquist–Shannon sampling rate [47]. For this approach, the acceleration rate achieved is proportional to the undersampling ratio, i.e., acquiring one-fourth the number of data points compared to the fully sampled case (referred to as 4x undersampling) typically requires one-fourth the time, which results in 4x acceleration. As is true for most natural images, the power of the MRI data collected in the frequency domain (k-space) is highly concentrated toward the center of the k-space grid (low frequency components) and falls off toward the edges (high frequency components). However, only sampling the center of the k-space, produces images of lower spatial resolution that lacks the fine details which are often the key to making diagnostic inferences from an MR image. A more balanced approach is to fully sample a small section of the central region of the k-space and sparsely sample the outer regions. Various undersampling patterns that utilize this approach in different ways can be used to collect k-space data, depending on the acquisition type. For Cartesian 2D acquisitions, 1D undersampling with a certain number of fully sampled central lines, typically in the phase encoding direction, is commonly used, whereas Cartesian 3D

acquisitions can be undersampled using a 2D pattern, with a small fully sampled center and a Gaussian or Poisson distribution of sampling points elsewhere.



(a) (b)

Figure 12: Examples of 5x K-space Undersampling Patterns (a) 2D Poisson Disc with Fully Sampled Center of Radius 18 (b) 1D Random Sampling with Fully Sampled Center of Width 11

However, conventional linear reconstruction from the sub-Nyquist undersampled k-space using two-dimensional (2D) inverse Fourier transform (IFT) with zero filling can result in severe aliasing artifacts and noise in reconstructed images. Therefore, various reconstruction algorithms have been proposed to predict the missing k-space data and/or reduce the presence of image artifacts due to undersampling.

One such method is accelerated parallel imaging, in which each coil's k-space signal is undersampled. As long as the total number of measurements across all coils exceeds the number of image voxels to be reconstructed, an unregularized least squares solution can still be used,

leading to a theoretical n_c-fold speedup over fully-sampled single-coil imaging where n_c is the number of coils. Each extra coil effectively produces an additional “sensitivity-encoded" measurement of the volume, which augments the frequency and phase encoded measurements obtained from the sequential application of magnetic field gradients in the MR pulse sequence. Estimates of coil sensitivity patterns, required for inversion of the undersampled multi-coil linear system, may be generated from separate low-resolution calibration scans or derived directly from the k-space measurements by fully sampling a comparatively small central region of k-space, which corresponds to low spatial frequencies.

The image domain sensitivity encoding (SENSE) technique developed by Prussemann et al. [48] exploited the spatial diversity information from coil sensitivity maps for fast MRI acquisition and led to the development of algorithms such as Generalized Autocalibrating Partially Parallel Acquisitions (GRAPPA) by Griswold [49] in which missing k-space points are directly interpolated.

3.5 Compressed Sensing Reconstruction

In practice, the use of sub-sampling results in significant amplification of noise and the introduction of undersampling artifacts. CS, which is one of the most successful modern approaches for reconstructing high-quality artifact-free MR images from undersampled data, is a signal processing technique that focuses specifically on exploiting the 'sparsity' in signals, typically in a transform domain. Ever since the first demonstration of CS MRI by Lustig et al., [50], it has become an essential tool in modern MR imaging research.

MRI obeys two key requirements for successful application of CS. Firstly, MR images, like most medical imagery are naturally compressible by sparse coding in an appropriate transform domain (e.g., by wavelet transform). Some types of MR images, such as those obtained from MR angiography, are quite sparse even in the pixel representation as well. They can be made sparser by spatial finite differencing. More complex imagery, such as brain images, can be made sparse in more sophisticated domains, such as the wavelet domain. Secondly, MRI scanners naturally acquire encoded samples, rather than direct pixel samples (e.g., in spatial-frequency encoding).

In general, CS reconstruction tries to solve the following constrained optimization problem for the reconstructed complex image m:

$$m^* = \text{argmin } \|\Psi m\|_1 \quad [1]$$

$$\text{subject to } \|D\mathcal{F}m - y\|_2 < \epsilon$$

Here, Ψ is an appropriate basis which converts the image m into a sparse representation, D is the undersampling pattern, $\mathcal{F}$ represents the Fourier transform y is the measured k-space data and ϵ is the error threshold. The l_1 norm in the objective is what enforces the sparsity.

Compressed sensing theory exploits this sparsity to reconstruct the signal with good accuracy from relatively few measurements by a nonlinear procedure. As such, CS can be used to make accurate reconstructions from a small subset of k-space, rather than an entire k-space grid. For CS to be applied successfully, the MR images, in addition to being sparse in some domain, must have incoherent undersampling artifacts and should be reconstructed by a nonlinear method that enforces both sparsity of the image representation and consistency of the reconstruction with the acquired samples. The low coherence can be guaranteed through a random sampling of the k-space. Since most energy in MR imagery is concentrated close to the center of k-space and rapidly decays towards the periphery, uniform sampling yields poorer results compared to variable density sampling strategies with denser sampling toward the center.

However, as random sampling techniques may be subject to MRI hardware limitations, Hyun et al., [51] demonstrated that uniform subsampling along with some dense low-frequency sampling toward the center can deal with anomaly location uncertainty in the uniform sampling for a CS reconstruction approach.

The sparsity of a signal is closely related to signal redundancies as redundant signals can be easily converted to sparse signals using some transforms. Apart for coil redundancy, which forms the basis for acceleration through parallel imaging, two other major sources of redundancy have been investigated in compressed sensing MRI. First and foremost, it is the spatial domain redundancy which allows MR images to be sparsely represented in finite difference or wavelet transform domain. As a result, total variation (TV) [52], where finite-difference is used as the sparsifying transform and wavelet transform have been extensively used in most of the early CS MRI research approaches. Adding a Total Variation penalty to force the image to be sparse in finite-difference and Ψ, modifies the objective equation to:

$$m^* = \text{argmin } \text{TV}(m) \quad [2]$$

$$\text{subject to } \|D\mathcal{F}Sm - y\|_2 < \epsilon$$

where S defines the sensitivities of the coils in a multi-coil acquisition and $\text{TV}(m)$ can be expressed as:

$$\text{TV}(m) = \|\nabla m\|_1 \quad [3]$$

∇m here represents the gradient.

On the other hand, dynamic MR images such as cardiac cine, functional MRI, and MR parameter mapping have significant redundancy along the temporal dimension as well. For example, if the image is perfectly periodic, then temporal Fourier transform may be the optimal transform to sparsify the signal. However, in many dynamic MR problems, the temporal variations are dependent on the MR physics as well as specific motion of organs, so the analytic transform such as Fourier transform may not be an optimal solution, but more data-driven approaches such as PCA or dictionary learning are better options.

3.6 MRI Reconstruction Through Deep Learning

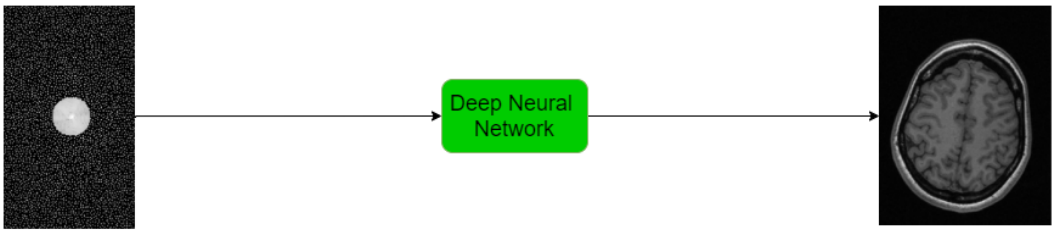
A large drawback of using PI and CS for MRI Acceleration is the time-consuming nature of the reconstruction problems that these methods are designed to solve. This is due in part to the fact that PI and CS approaches treat every examination and reconstruction task as a new, independent optimization problem. While reconstruction can be done offline, clinical scenarios ultimately require speed for the reconstruction of individual scans. Deep learning methods are useful because they perform the optimization over many training images prior to solving the reconstruction for any particular given image. They can take advantage of common features of

anatomy as well as the structure of undersampling artifacts that are present across the training images. With deep-learning reconstruction, the optimization process is then effectively decoupled from the time-sensitive image reconstruction process for each individual study. Thus, for a new scan, unlike CS reconstructions, which each require lengthy computation time, deep-learning-based models can complete the reconstruction in seconds. It is for these reasons that several groups are now investigating the use of deep-learning-based approaches to achieve accelerated, high-quality MRI reconstruction. Several techniques use DL to learn more effective regularization terms (prior information), and the approaches are derived from the concepts of GRAPPA, SENSE, and CS. Like PI, DL techniques for reconstruction of sparsely sampled data can be applied both in k-space and in image space.

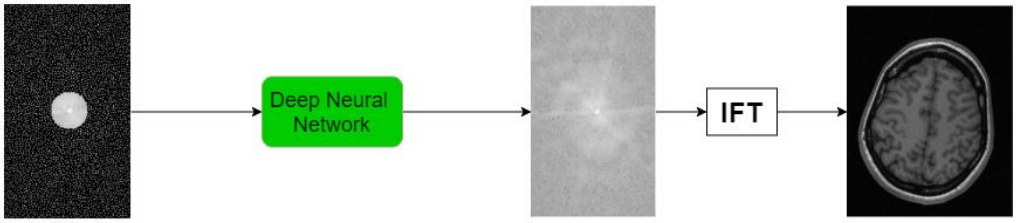
Current approaches to using DL for MRI Acceleration fall into two broad categories: Deep learning based MRI reconstruction, which utilizes complex multi-coil data (in k-space or image space) to generate an output image [28], and DL based MRI post-processing which aims to improve the quality of coil-combined images [53].

The first approach can be used to train powerful deep networks to exploit phase and coil information present in the raw data. However, such raw MRI datasets are still rare, and have to be deliberately acquired for research purposes since the extraction and curation of complex k-space data is not part of the regular MRI diagnostic process. The strength of the second approach lies in the more widespread availability of magnitude MRI data and the relative ease with which general DL based computer vision techniques and insights can be applied to magnitude images.

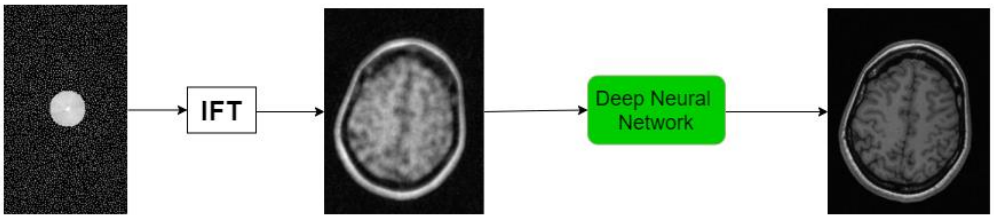
Now, depending on how the deep network is being used in the reconstruction process, DL based MRI reconstruction approaches can be further divided into four broad categories as shown in Figure 13



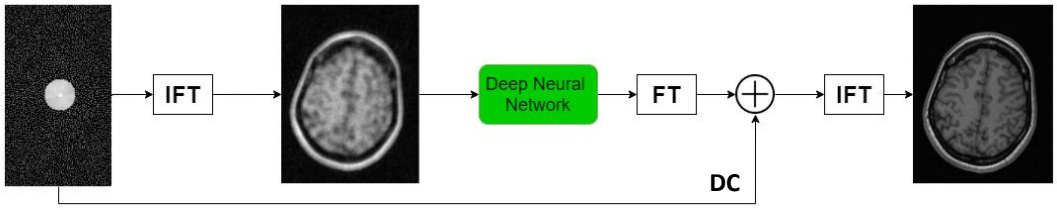
(a) End-to-end Reconstruction



(b) K-space Reconstruction



(c) Magnitude Image Domain Reconstruction



(d) Complex Image Domain reconstruction with Data Consistency (DC)

Figure 13: Different Approaches to Using Deep Neural Networks for MRI Reconstruction

The simplest approach (as shown in Figure 13(a)) is complete end-to-end recovery as proposed in the AUTOMAP system [54]. The undersampled complex k-space coil data is fed into a fully-connected network which produces the final output image. However, the excessive

computational requirements due to the large number of necessary fully-connected layers largely restrict the maximum size of the images that can be handled by this network. Since then, in an effort to reduce the resource requirements for the domain transform learning process, Schempler et al [55] have proposed a decomposed AUTOMAP and Eo et al., [56] have proposed an alternative approach which restricts the undersampling to 1D so that the fully connected layers only need to learn a 1D global transform.

The second approach (Figure 13(b)) is using the deep network for direct k-space interpolation. Akçakaya et al., [57] introduced the use of neural networks for k-space interpolation inspired by parallel imaging methods such as GRAPPA. Although a convolutional network is used in the process, it is a non-traditional implementation of deep networks as it is a scan-specific process that trains on ACS data. LORAKI, which uses a recurrent neural network, was subsequently developed by Kim et al [58] in an effort to translate the parallel imaging method AC-LORAKS into a nonlinear DL method. The first ground up DL k-space interpolation technique came from Han et al [59] who used a U-Net to implement a structured low-rank Hankel matrix approach. Recently, this approach was enhanced by Du et al., [60] through the addition of weight sharing, frequency attention and integrating information from slices in the spatial neighborhood into the reconstruction process.

Wang et al [61] proposed the usage of CNNs for MRI reconstruction in a more conventional way, i.e., in the image domain as shown in Figure 13(c). In their work, a 3-layer CNN is used to learn the mapping between the zero-filled aliased image and the ground truth image. This approach was then extended by Schlemper et al [62] into a cascade of CNNs interleaved with Data Consistency steps which restore known k-space values. Aside from CNNs, many other deep network architectures have been used for image domain MRI reconstruction including multi-layer

perceptrons [63], variational networks [64] [65], recurrent networks [66] [67], U-Nets [68] [51], and Generative Adversarial Networks (GANs) [69]. Enforcement of data consistency can be incorporated into this approach (Figure 13(d)) by replacing data from the originally acquired k-space into the reconstructed images.

These methods can also be combined to create hybrid methods where cascaded or parallel deep networks, each focusing on data in a different domain, are trained together on k-space and image data. Eo et al [70] and Souza et al [28] independently explored different combinations of cascaded k-space and image domain networks with interleaved data consistency steps and found the approach to be promising.

3.7 U-Nets for MRI Reconstruction

Encoder-decoder CNN architectures are one of the most popular approaches to solving inverse problems in imaging with deep networks. The encoder part spatially down-samples the input in several stages and acts as a compressive element which learns an abstract representation of the input image. The decoder part expands the abstract representation to form an output image through a number of up-sampling steps.

One issue with this approach is that a significant amount of detail from the input is lost as the spatial information is compressed by the encoder. This loss is mitigated in structures such as the U-Net through the use of skip-connections between encoder and decoder layers at the same network depth. The skip-connections act to reintroduce high resolution spatial features that may have been lost due to encoder compression.

U-Net, or architectures derived from the U-Net, have successfully been applied to many different inverse problems for imaging such as image denoising [71], and CT reconstruction [72]. In 2016, Jin et al. [72] first demonstrated the feasibility of using a residual U-Net for X-Ray CT and MRI reconstruction and highlighted the benefit of residual learning, an approach where the network only learns the difference between the input and the output, in the same study. Since then, many U-Net based approaches for MRI reconstruction have been proposed.

Inspired by the hybrid cascade of k-space and image domain CNN KIKI-net developed by Eo et al., [73], Souza et al proposed, and then improved upon a cascade of U-Nets for MRI reconstruction [74][75].

3.8 Image Quality Metrics (IQMs) for MRI Assessment

Today, the most widespread usage of MRI output images is for qualitative assessment by a radiologist. While automated segmentation and classification of MRI data is becoming commonplace in research settings, in practice, only a doctor or expert in MR image analysis can look at the test set output for a truly undersampled input sequence and provide the final verdict about the quality of reconstruction. However, the validation data may be compared quantitatively with the available ground truth using different Image Quality Metrics (IQMs) to compare the performance of two networks or to conduct hyperparameter tuning without the need for constant expert evaluation.

The simplest way to directly compare the similarity of two images is to determine their overall pixel-wise intensity difference through a metric such as the Mean Squared Error (MSE) or the Root Mean Squared Error (RMSE). For two images $A(x, y)$ and $B(x, y)$ of width w and height h, the MSE and RMSE are defined as follows:

$$\mathrm{MSE}(A, B) = \frac{1}{wh}\sum_{x=1}^{w}\sum_{y=1}^{h}[A(x, y) - B(x, y)]^2$$

$$\mathrm{RMSE}(A, B) = \sqrt{\mathrm{MSE}(A, B)}$$

Thanks to its computational simplicity, the squared error is widely used as a loss function in image reconstruction.

The Peak Signal to Noise Ratio (PSNR) metric is also widely used for quantitative image quality comparison. It is the ratio of peak signal power to average noise power (typically represented with the MSE) and is expressed as:

$$PSNR = 10 \cdot log_{10}\left(\frac{\|B\|_{\infty}^{2}}{MSE(A,B)}\right)$$

However, pixel-wise-difference-based metrics have been shown to not be reliable indicators of perceived image quality. The performance of objective IQMs with respect to how well they correlate with human perceptions of image quality (usually represented by Mean Opinion Scores or MOS) is a vigorously contentious area of research in computer vision that has seen numerous metrics being developed over the last few decades. In limited studies, metrics such as the Visual Information Fidelity (VIF), Feature Similarity Measure (FSIM), Noise Quality Measure (NQM) and High-Frequency Error Norm (HFEN) have been shown to correlate to MOS to various degrees depending on the type of image degradation, but their high computation costs along with a lack of universal consensus about their efficacy is preventing widespread adaptation of these metrics. Due to its relative simplicity among perceptual image quality metrics, the Structural Similarity Index Measure (SSIM), developed by Wang et al., in 2004 [76], is still the most widely used IQM for MR image quality comparison and evaluation.

SSIM compares the luminance, contrast, and structure of two images to provide a measure of the similarity of the images on a scale of 0 to 1. The SSIM for two images $A(x, y)$ and $B(x, y)$ is given by:

$$\text{SSIM}(A,B) = \frac{(2\mu_A\mu_B + C_1)(2\sigma_{AB} + C_2)}{(\mu_A^2 + \mu_B^2 + C_1)(\sigma_A^2 + \sigma_B^2 + C_2)}$$

Here, μ_A is the mean intensity of A, σ_A^2 is a variance of A and σ_{AB} is a covariance of A and B. C_1 and C_2 are two variables included to avoid instability when $\mu_A^2 + \mu_B^2$ is very close to zero. They are defined as $C_1 = (K_1L)^2$ and $C_2 = (K_2L)^2$ where L is the dynamic range of the pixel

intensities and K_1 and K_2 are small constants typically set to 0.01 and 0.03, respectively, for most standard implementations.

For most image quality assessment tasks, SSIM is usually calculated locally using a square window which moves pixel-by-pixel to generate a pixel-wise SSIM index map of the entire image. Depending on the implementation, this window size typically varies from 7 to 11 pixels. This SSIM index map is usually averaged over the entire image and the Mean SSIM (MSSIM) value used as a comparison metric for the images in question. The MSSIM for two images $A(x, y)$ and $B(x, y)$ each with dimensions w and h can be represented as:

$$\mathrm{MSSIM}(A, B) = \frac{1}{wh}\sum_{x}\sum_{y}\mathrm{SSIM}(x, y)$$

where $\mathrm{SSIM}(x, y)$ is the SSIM for window position (x, y).

Chapter 4: Developing an Efficient U-Net for MRI Acceleration

In this chapter, we investigate two methods of creating a U-Net with an EfficientNet encoder. The Calgary-Campinas multi-coil dataset that is used to train and evaluate the Efficient U-Nets is also explored, as well as the full reconstruction pipeline which delineates how raw MR k-space data is processed and reconstructed using a multi-channel Efficient U-Net.

4.1 Decoder Design for the Efficient U-Net

While a U-Net can be created by designing the encoder and decoder parts as a whole [28], a common approach is to take an existing CNN classifier architecture, remove any fully-connected layers, and use the resulting fully convolutional network as the encoder part of the U-Net [77]. The decoder half is then designed to either mirror the encoder (with upsampling layers instead of pooling) or designed iteratively through empirical evaluation of the full network. The latter approach often ends up producing asymmetric U-Nets [42].

4.1.1 Symmetric Decoder

Figure 14 shows a fully symmetric approach to the decoder design. In this approach, a decoder block is used for every unique MBConv Block in the encoder. Successive MBConv Blocks that have the same number of output channels are ignored. Depending on whether MBConv Block downsamples the input (pink encoder blocks in the diagram) or retains the input spatial dimension size (yellow encoder blocks), an UpBlock or a Feature Block is used as the next corresponding decoder block. The main difference between these two decoder blocks is that, as the name suggests, the UpBlock begins with an upsampling operation in the form of a 2D Transposed Convolution whereas the Feature Block begins with a regular 2D Convolution. Both blocks follow the initial operation with a concatenation of the feature maps with encoder feature maps obtained through the skip connection and two convolution operations. A skip connection is

provided from the first operation to the last convolution of the block to make the network robust against vanishing gradients. This approach allows us to extract the maximum number of unique feature maps from the encoder layers through the skip connections.

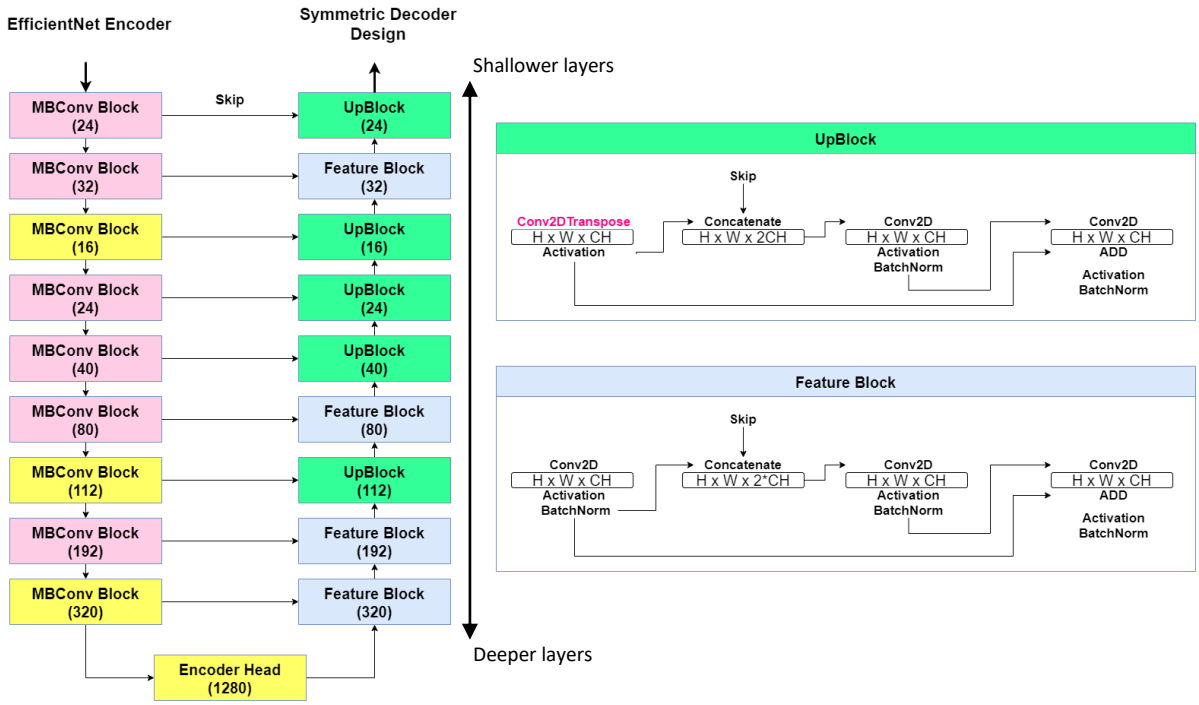


Figure 14: Efficient U-Net with Symmetric Decoder. The Encoder spatially downsamples the images while increasing the number of feature channels (in parenthesis) and the Decoder reverses the process to obtain an image of the original spatial dimensions.

4.1.2 Asymmetric Decoder

A more refined approach to the decoder design is one that takes into account the fact that Deep CNNs are very effective at quickly recovering low frequency information compared to high frequency information [78][79]. The deeper layers of the EfficientNet encoder, which was originally designed as a classifier, create feature maps from an input that is already heavily downsampled and so do not contribute as significantly to the recovery of high frequency information. It is therefore beneficial for the decoder to focus more on the feature maps from the

upper layers for detail recovery. A problem that arises at this point is that the upper layers of the EfficientNet encoder, by design, have relatively few feature channels compared to spatially similar layers in comparable networks. Deep CNN encoders typically double the number of feature channels every time the input is pooled or downsampled [28], but in case of the EfficientNet, the number of output feature channels at each depth level has been chosen from the result of the neural architecture search which has FLOPS efficiency as one of its priorities. Having a large number of feature maps at higher spatial resolutions results in convolution operations with large numbers of parameters and floating-point operations which results in the architecture search choosing small numbers of output channels instead.

In order to mitigate this deficiency of the encoder with respect to our purpose, we add a feature expansion step in the decoder blocks and exploit the high frequency information crossing over to the decoder through the skip connections. An expansion factor 'M' is used to expand the number of channels through successive convolutions within a decoder block after the upsampled image is concatenated with the skip input. An output convolution is then used to bring the number of channels down to make the decoder and skip features have the same number of channels in the next layer.

This approach, however, introduces a large number of new floating point operations into the network, reducing the overall efficiency. This issue can be mitigated by removing some redundant layers from the decoder. Since the EfficientNet is a very powerful encoder, we can expect the final compressed feature representation from the deepest point in the network, the 'Encoder Head', to very effectively convey high-level features to the decoder making some of the deeper skip connections and decoder layers redundant. This idea also applies to successive MBConv blocks that output feature maps at the same spatial resolution. If a skip connection is

already obtained from a deeper MBConv block within the encoder, skip connections from shallower blocks at the same spatial resolution might be ignored because the encoded features at that particular spatial resolution has already been processed by the decoder by that point.

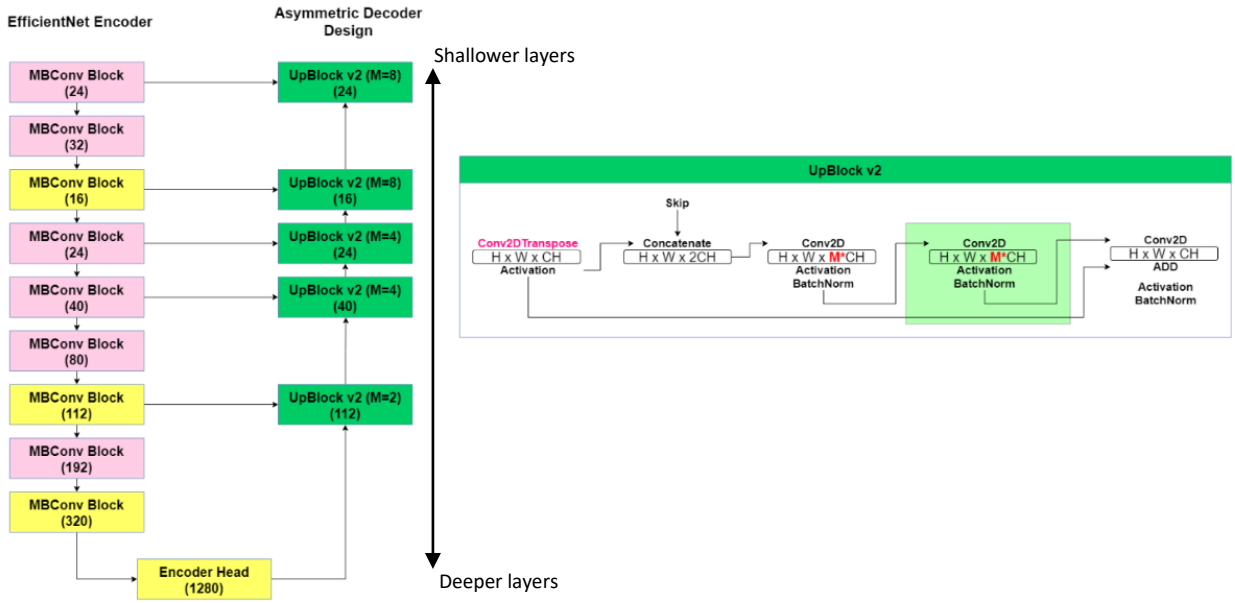


Figure 15: Efficient U-Net with Asymmetric Decoder

Figure 2 shows a complete Efficient U-Net using the asymmetric decoder design. The modified UpBlock (v2) has an extra convolution step with both internal convolution layers expanding the channels by a factor of 'M'. M is increased in the shallower layers to make more filter parameters available for the network to effectively learn low level features in the input.

4.2 Dataset Details

The MRI dataset used for the purposes of evaluating the Efficient U-Net is the 12-Channel Coil (k-space) brain dataset released as part of the Calgary-Campinas Public Dataset [80]. It

consists of 67 volumes of 3D, T1-weighted, gradient-recalled echo, 1 mm isotropic sagittal acquisitions collected on a clinical MR scanner. The acquisition and dataset details are as follows:

- Scanner: Discovery MR750; General Electric (GE) Healthcare, Waukesha, WI
- Subjects: Healthy subjects (age: 44.5 years ± 15.5 years; range: 20 years to 80 years)
- Acquisition parameters: TR/TE/TI = 6.3 ms/2.6 ms/650 ms and TR/TE/TI = 7.4 ms/3.1 ms/400 ms
- Average scan duration: ~341 seconds
- Slices: 170 to 180 contiguous 1.0-mm slices
- Field of view: 256 mm x 218 mm

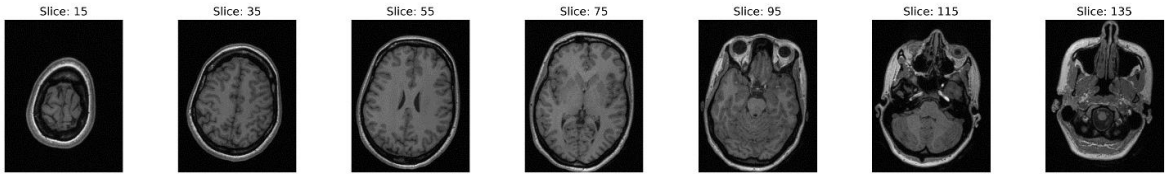


Figure 16: Sample Slices from the Calgary-Campinas Multi-coil Dataset

The acquisition matrix size for each channel was $Nx \times Ny \times Nz = 256 \times 218 \times [170,180]$. In the slice-encoded direction (kz), data was partially collected up to 85% of its matrix size and then zero filled to $Nz = [170,180]$. The inverse Fourier transform was applied by the scanner to the frequency-encoded direction and a hybrid $x - ky - kz$ dataset was saved. As a result, the k-space can be undersampled in the phase encoding and slice encoding directions to simulate 1D or 2D undersampled reconstruction problems.

Figure 3 shows some sample axial slices from a volume in the dataset. The data is stored as HDF5 volumes. The 12 complex k-space coil value arrays are stored as separate channels of 64-bit floating point real and imaginary components, resulting in 24 channels of data per slice.

4.3 Reconstruction Pipeline

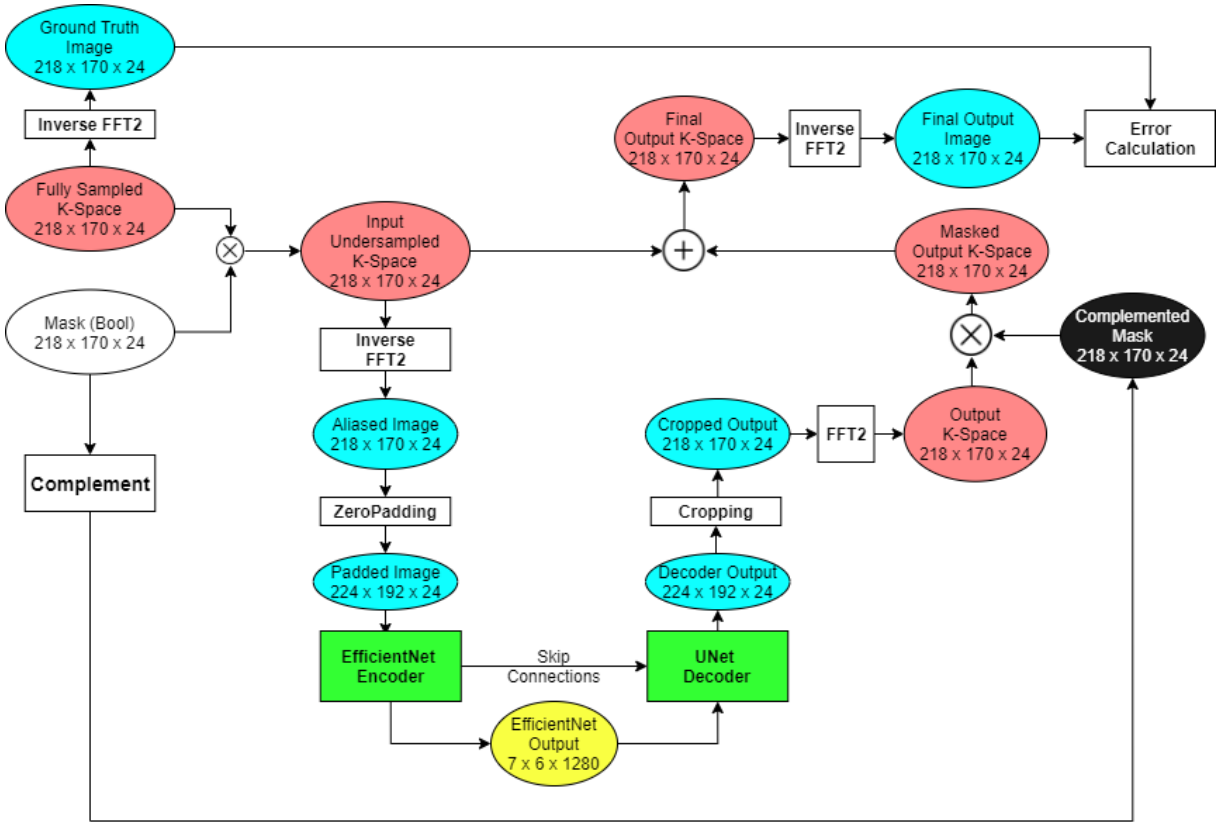


Figure 17: Efficient U-Net MRI Reconstruction Pipeline

The full reconstruction pipeline for the Efficient U-Net is shown in Figure 17. Red ovals represent data in the frequency domain (k-space) and blue ones represent data in the image domain. The EfficientNet encoder output is a set of compressed image feature maps represented by the yellow oval. Rectangular blocks represent operations with green rectangular blocks representing operations with learnable parameters (the core of the network).

The fully sampled k-space from the dataset is first masked with a 2D Poisson disc Boolean undersampling mask (Figure 18) to achieve 5x undersampled k-space. A 2D inverse Fourier transform is used to convert the k-space data to the ground truth image data. The complement of

the undersampling mask, the undersampled k-space, and the ground truth image is then passed into the network.

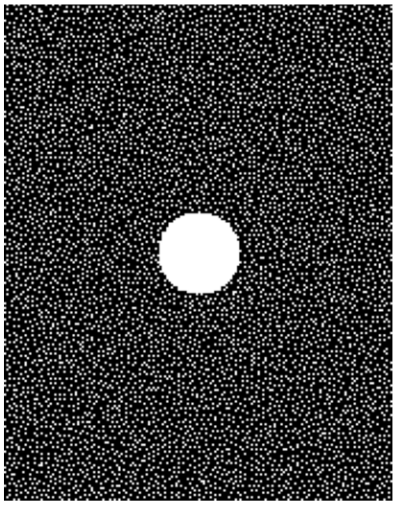


Figure 18: 2D Poisson Disc 5x Undersampling Mask (Center R = 18)

The deep network based learning process is performed entirely in the image domain. Inside the network, the undersampled k-space is converted to aliased images without any type of channel combination so that the coil-wise images (split into real and imaginary components) can be used as the input for the learnable section of the network. The images are then zero-padded in each spatial dimension just enough to make sure that they can be repeatedly downsampled by factors of 2 up to the deepest layer of the network and still maintain integer spatial dimension values.

The zero-padded images are then passed onto the first section of the Efficient U-Net, which is the standard EfficientNet Encoder. The output of the encoder, along with feature maps form shallower layers, are passed onto the custom decoder section which, through the use of UpBlocks, gradually reconstructs the image until it is back to the same spatial resolution and number of channels as the encoder input.

A data consistency step in the frequency domain is then performed on the decoder output. The decoder output, initially in the image domain, is cropped so that the spatial resolution matches the original data and then a Fourier transform is performed to convert it back to the k-space. This output k-space is then masked using the complement of the original undersampling mask to only retain points at which the input masked k-space did not have any data. The input undersampled k-space is then added to this complement-masked output k-space to generate the final output k-space which is converted to the image domain through an inverse Fourier transform. The reconstruction error between this final image domain output and the input ground truth image is calculated through a loss function and backpropagated through the network to update the filter weights in the encoder and decoder.

Chapter 5: Experiments and Results

5.1 Efficient U-Net Implementation Details

The details for the Efficient U-Net as implemented in this work are given below:

- All networks were implemented in TensorFlow using the Keras API on Ubuntu Linux workstation
- Learning Rate of 1×10^{-3} was used for the first 20 epochs and then reduced to 1×10^{-4} till convergence
- A batch size of 4 was used for all training sessions
- Adam optimizer was used with default parameters and Mean Squared Error loss
- Training set: 47 volumes from the Calgary Campinas Dataset
- Validation set: 19 volumes from the Calgary Campinas Dataset
- MRI Acceleration was simulated through 2D 5x Poisson Disc undersampling of the data with a fully sampled center of radius 18

5.2 Deep Learning Baseline – U-Net

The chosen DL baseline was a traditional multi-channel symmetric U-Net of depth 5 layers and 256 feature channels at the deepest layer with Data Consistency steps as implemented in our Efficient U-Net reconstruction pipeline. This baseline U-Net was trained with the same 12-channel complex image training and validation data, as well as the same set of hyperparameters as the Efficient U-Net. 2D convolutions were used in the encoder path along with ReLU activation and 2x2 MaxPooling. Upsampling2D was used for upsampling operations in the decoder path followed by 2D convolutions and ReLU activation.

5.3 Compressed Sensing Baseline – Total Variation Minimization

The BART toolkit, developed by Lustig et al. [81], was used to perform Compressed Sensing reconstruction on 5x Poisson Disc undersampled test images using a total variation regularizer. A grid search was used to find optimum values for regularization strength and number of iterations. Based on the results, a regularization strength of 0.001 was used for the final test data and the optimization was performed over 100 iterations.

5.4 ROI Evaluation of IQMs

The main goal of the neural networks used for MRI reconstruction is to accurately reconstruct anatomical detail. However, in most slices of MR images, there is an empty region around the anatomy since imaged anatomy is irregularly shaped and the data is typically saved as a multi-dimensional Cartesian array. This 'empty' region consists of random noise caused by artifacts and incoherent magnetic field measurements in the k-space. Therefore, it is difficult for CNN-based image-domain neural networks, which attempt to learn features in the form of visual structures, to learn how to faithfully generate these regions that are devoid of anatomy. It follows then, that for a more truthful evaluation of a deep network's ability to reconstruct coherent visual structures through an IQM such as the SSIM, performing the metric calculation over a region of interest (ROI) that mainly includes anatomical structures may be advantageous.

In order to do so, coarse ROI masks were generated for the test datasets by thresholding out regions where the signal value is less than the average noise value and then using morphological processing to fill gaps and generate binary masks that encompass any region with anatomy present. The SSIM and PSNR metrics shown in this work were evaluated over the generated ROI of the images to better isolate the performance of the reconstruction process in

structured regions in the images. Figure 19 shows some slices along with corresponding ROI masks.

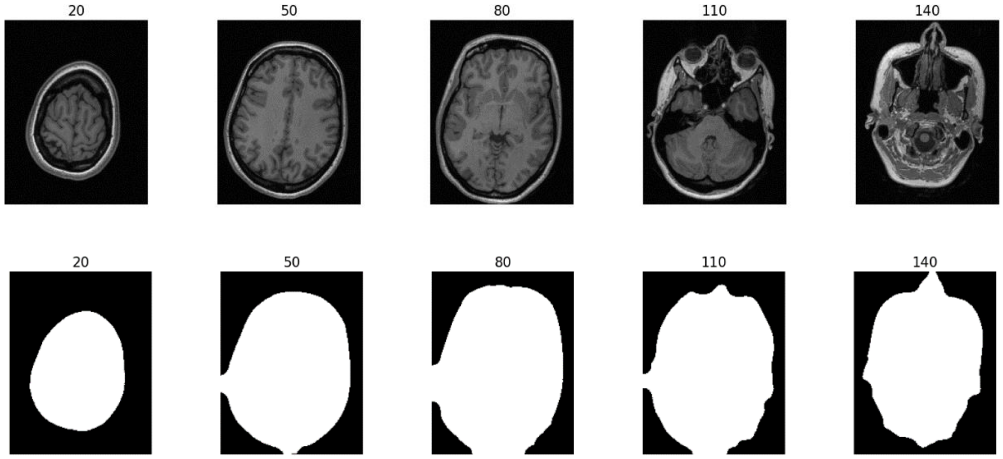


Figure 19: Sample Images and Corresponding ROI Masks. Slice Number Shown in Parentheses.

5.5 IQM Comparison

Table 1: MSSIM and PSNR Comparison of the Different Reconstruction Techniques

	MSSIM				**PSNR**			
	Min	**Max**	**Mean**	**Std.**	**Min**	**Max**	**Mean**	**Std.**
Zero-filled Reconstruction	0.5934	0.7821	0.6634	0.0373	11.5871	25.5846	17.2815	3.5456
CS Baseline	0.6113	0.7419	0.6429	0.0290	18.2295	29.7100	21.5590	2.3551
Baseline U-Net	0.8418	0.9350	0.8923	0.0221	23.7158	32.0591	28.9922	1.7102
Efficient U-Net 1 (Symmetric)	0.7873	0.9196	0.8322	0.0265	17.0242	29.0235	23.8891	2.0071
Efficient U-Net 2 (Asymmetric)	**0.8548**	**0.9383**	**0.9018**	**0.0207**	**24.4798**	**32.3598**	**29.7709**	**1.5690**

Table 1 shows the IQM values obtained from an analysis of the outputs of the various reconstruction techniques when subjected to the test data. Both the PSNR and Mean SSIM metrics

were calculated over the ROIs described in the previous section. It is clearly evident what while the symmetric approach to producing an Efficient U-Net struggles to reach the reconstruction quality of the baseline which uses regular convolutions, making the network asymmetric by adding more learnable parameters in the decoder at larger spatial resolution levels overcomes these issues and produces a network which outperforms the baseline and all other reconstruction methods tested while still being highly computationally efficient. The average reconstruction time for all three deep networks is shown in Table 2: Reconstruction Time Comparison where we can see that despite the small added penalty for introducing more parameters in the asymmetric network, it is still roughly 4 times faster at inference compared to the baseline.

Compared to the DL approaches, the CS total variation regularization technique, while managing to boost the PSNR of the input images compared to the zero-filled reconstruction, actually resulted in a reduction in the Mean SSIM as it failed to enhance structural detail.

Table 2: Reconstruction Time Comparison

	Baseline U-Net	**Efficient U-Net 1 (Symmetric)**	**Efficient U-Net 2 (Asymmetric)**
Average reconstruction time per slice (ms)	159	25	37

5.6 Visual Comparison of Results

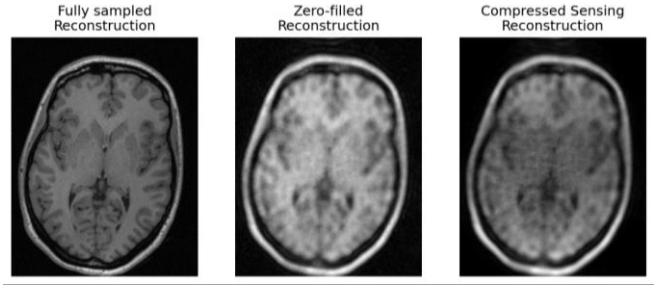


Figure 20: Compressed Sensing Reconstruction Results

From a visual analysis of the results, it is evident that simple compressed sensing reconstruction using total variation regularization struggles with MRI data undersampled at high acceleration factors. This finding is consistent with the views expressed in relevant literature and the IQM results that were obtained. In our tests, total variation regularization was able to 'denoise' the data to a certain extent and remove some artifacts which resulted in an increase in the PSNR, but from a visual comparison of the images and the MSSIM value of the reconstruction, we can see that this technique was not very effective at recovering structural detail within the anatomy that is lost as a result of undersampling.

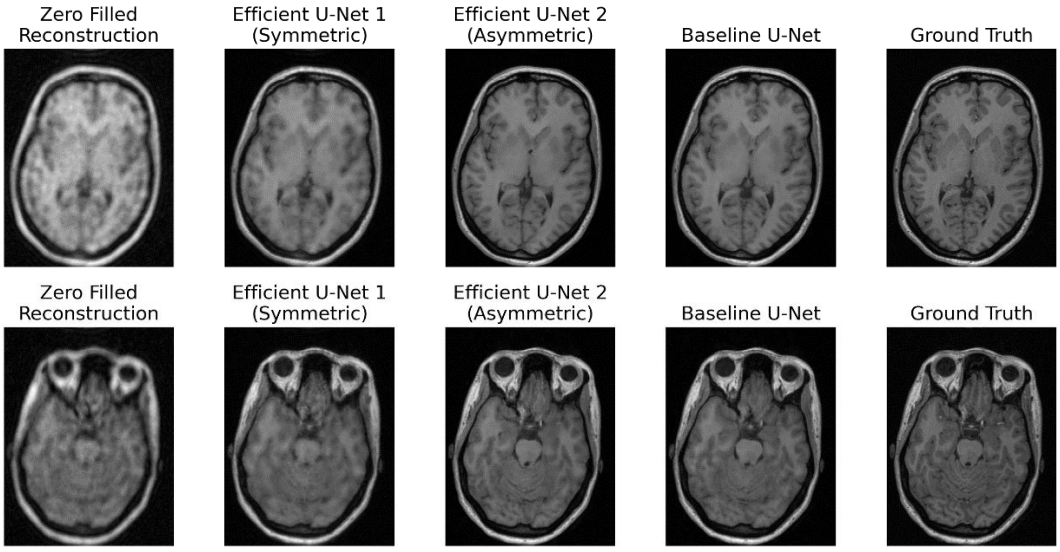


Figure 21: Deep Learning Reconstruction Results for Two Different Slices from the Test Set

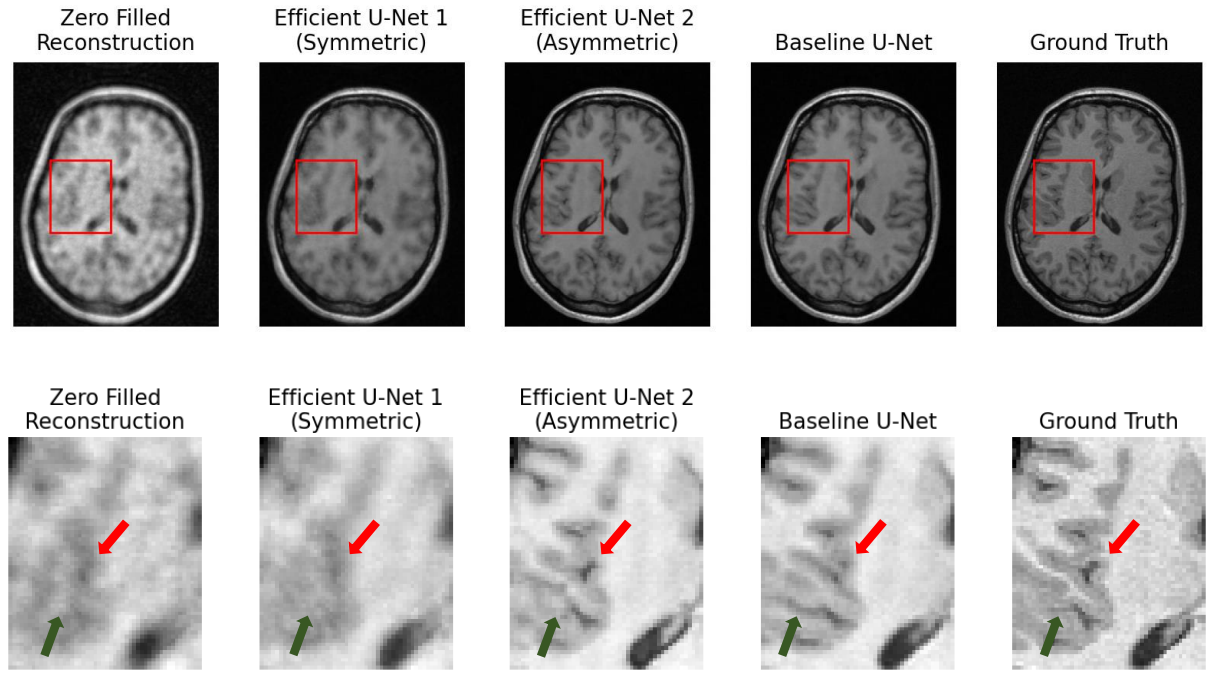


Figure 22: Detailed Look at Reconstruction Quality and Errors. The Second Row of Figures Show Zoomed-in Views of the Rectangular Region Highlighted in Red.

Figure 21 shows a comparison of the reconstruction quality of the deep networks for two different slices from the test set. As reflected by the IQMs, the asymmetric Efficient U-Net manages to produce results that are visually closer to the ground truth compared to the symmetric approach. However, the visual difference between the reconstruction quality of the asymmetric Efficient U-Net and the baseline U-Net is difficult to ascertain by casual inspection. Both networks manage to produce results that are strikingly similar to the ground truth.

A closer look at the images, as shown in Figure 22, reveals that both networks are prone to producing errors when reconstructing small structural details. The red arrow highlights a region where the Asymmetric Efficient U-Net successfully reproduced a structure but the baseline network could not. On the other hand, the green arrow highlights a region where a small structure was not properly reproduced by the Asymmetric Efficient U-Net. While the baseline network

managed to create that particular structure, it ‘hallucinated’ extra detail resulting in an overall unfaithful reproduction. The relevance of these tiny structures in the anatomy is beyond the scope of this thesis and belongs in the realm of subjective analysis by trained radiologists. However, the capability of truthfully reproducing tiny structural details is one the major goals of MRI acceleration strategies and it is encouraging to see the Efficient U-Net being on the conservative end of the spectrum when it comes to creating detail that was not present in the undersampled image.

Chapter 6: Conclusion and Future Work

6.1 Summary

The objective of this Thesis was to explore Deep Learning based undersampled MRI reconstruction for MRI acceleration. The highly effective U-Net structure was chosen as the basis for the deep network and was combined with the EfficientNet CNN which uses depthwise convolutions to construct a highly effective network which promises powerful feature learning capabilities while being greatly computationally efficient thanks to the use of depthwise convolutions.

Two approaches to the decoder design (symmetric and asymmetric) were tested and the asymmetric network, when compared with Total Variation Compressed Sensing and standard U-Net baselines and was found to generate higher quality results in terms of the SSIM and PSNR metrics while having greatly reduced inference time compared to a standard U-Net.

6.2 Proposal for Future Work

The complex nature of the MRI reconstruction problem along with the fast-paced nature of DL research leaves a wide variety of avenues for future work in this area. Newer versions of the EfficientNet can be incorporated into this framework along with efficient decoder designs to further increase reconstruction speed and accuracy, and the model could be trained and tested using a variety of other datasets and acceleration factors, as well as different, possibly learnable undersampling patterns to better study its learning and generalization capabilities. Further work can also be done on developing an in-depth understanding of feature extraction by deep networks for multi-channel MRI reconstruction to improve network reliability.

References

[1] R. Otazo, D. Kim, L. Axel, and D. K. Sodickson, "Combination of Compressed Sensing and Parallel Imaging for Highly Accelerated First-Pass Cardiac Perfusion MRI," *Magnetic resonance in medicine*, vol. 64, pp. 767–776, 2010, doi: 10.1002/mrm.22463.

[2] J. Zbontar *et al.*, "fastMRI: An Open Dataset and Benchmarks for Accelerated MRI," pp. 1–29, 2018, [Online]. Available: http://arxiv.org/abs/1811.08839

[3] G. Montavon, W. Samek, and K. Müller, "Methods for interpreting and understanding deep neural networks," *Digital Signal Processing*, vol. 73, pp. 1–15, 2018, doi: 10.1016/j.dsp.2017.10.011.

[4] A. Khan, A. Sohail, U. Zahoora, and A. S. Qureshi, "A survey of the recent architectures of deep convolutional neural networks," *Artificial Intelligence Review*, vol. 53, no. 8, pp. 5455–5516, 2020, doi: 10.1007/s10462-020-09825-6.

[5] M. Tan and Q. v. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," *36th International Conference on Machine Learning, ICML 2019*, vol. 2019-June, pp. 10691–10700, 2019.

[6] W. S. Mcculloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *Bulletin of Mathematical Biology*, vol. 52, no. l, pp. 99–115, 1943.

[7] Rosenblatt, "The Perceptron - A Perceiving and Recognizing Automaton," 1957.

[8] G. Cybenko, "Approximation by Superpositions of a Sigmoidal Function," *Mathematics of Control Signals, and Systems*, vol. 2, no. 4, pp. 303–314, 1989.

[9] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016. [Online]. Available: http://www.deeplearningbook.org

[10] B. Zoph and Q. v Le, "Searching for activation functions," *6th International Conference on Learning Representations, ICLR 2018 - Workshop Track Proceedings*, pp. 1–13, 2018.

[11] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 157–166, Mar. 1994, doi: 10.1109/72.279181.

[12] Y. LeCun, P. Haffner, L. Bottou, and Y. Bengio, "Object Recognition with Gradient-Based Learning," in *Lecture Notes in Computer Science*, no. 1681, 1999, pp. 319–345. doi: 10.1007/3-540-46805-6_19.

[13] K. Bai, "A Comprehensive Introduction to Different Types of Convolutions in Deep Learning," *Towards Data Science*, 2019. https://towardsdatascience.com/a-comprehensive-introduction-to-different-types-of-convolutions-in-deep-learning-669281e58215 (accessed May 25, 2021).

[14] E. Shelhamer, J. Long, and T. Darrell, "Fully Convolutional Networks for Semantic Segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 4, pp. 640–651, 2017, doi: 10.1109/TPAMI.2016.2572683.

[15] D. Mishra, "Transposed Convolution Demystified," 2020. https://towardsdatascience.com/transposed-convolution-demystified-84ca81b4baba

[16] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, May 2017, doi: 10.1145/3065386.

[17] C. Szegedy *et al.*, "Going deeper with convolutions," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 07-12-June, pp. 1–9, 2015, doi: 10.1109/CVPR.2015.7298594.

[18] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, pp. 1–14, 2015.

[19] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[20] S. Ioffe and C. Szegedy, "Batch Normalization : Accelerating Deep Network Training by Reducing Internal Covariate Shift," in *32nd International Conference on International Conference on Machine Learning*, 2015, pp. 448–456.

[21] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 5987–5995, 2017, doi: 10.1109/CVPR.2017.634.

[22] J. Hu, "Squeeze-and-Excitation_Networks_CVPR_2018_paper.pdf," *Cvpr*, pp. 7132–7141, 2018, [Online]. Available: http://openaccess.thecvf.com/content_cvpr_2018/html/Hu_Squeeze-and-Excitation_Networks_CVPR_2018_paper.html

[23] H. Zhang *et al.*, "ResNeSt: Split-Attention Networks," 2020.

[24] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9351, pp. 234–241, 2015, doi: 10.1007/978-3-319-24574-4_28.

[25] F. Milletari, N. Navab, and S. Ahmadi, "V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation Fausto," in *Fourth International Conference on 3D Vision (3DV)*, 2016, pp. 565–571.

[26] J. Zhang, H. Zhu, P. Wang, and X. Ling, “ATT Squeeze U-Net: A Lightweight Network for Forest Fire Detection and Recognition,” *IEEE Access*, vol. 9, pp. 10858–10870, 2021, doi: 10.1109/ACCESS.2021.3050628.

[27] W. Zhang *et al.*, “ME-Net: Multi-encoder net framework for brain tumor segmentation,” *International Journal of Imaging Systems and Technology*, no. November 2020, pp. 1–15, 2021, doi: 10.1002/ima.22571.

[28] R. Souza, M. Bento, N. Nogovitsyn, K. J. Chung, R. M. Lebel, and R. Frayne, “Dual-domain Cascade of U-nets for Multi-channel Magnetic Resonance Image Reconstruction,” pp. 1–13, 2019, [Online]. Available: http://arxiv.org/abs/1911.01458

[29] Y. Kong *et al.*, “Automated yeast cells segmentation and counting using a parallel U-Net based two-stage framework,” *OSA Continuum*, vol. 3, no. 4, p. 982, 2020, doi: 10.1364/osac.388082.

[30] Z. Zhou, M. M. Rahman Siddiquee, N. Tajbakhsh, and J. Liang, “UNet++: A Nested U-Net Architecture for Medical Image Segmentation BT - Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support,” vol. 11045, no. 2018, pp. 3–11, 2018, doi: 10.1007/978-3-030-00889-5.

[31] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, “SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size,” pp. 1–13, 2016, [Online]. Available: http://arxiv.org/abs/1602.07360

[32] M. Wang, B. Liu, and H. Foroosh, “Factorized Convolutional Neural Networks,” *Proceedings - 2017 IEEE International Conference on Computer Vision Workshops, ICCVW 2017*, vol. 2018-Janua, pp. 545–553, 2017, doi: 10.1109/ICCVW.2017.71.

[33] J. Wu, C. Leng, Y. Wang, Q. Hu, and J. Cheng, “Quantized convolutional neural networks for mobile devices,” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2016-Decem, pp. 4820–4828, 2016, doi: 10.1109/CVPR.2016.521.

[34] S. Han, H. Mao, and W. J. Dally, “Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding,” *4th International Conference on Learning Representations, ICLR 2016 - Conference Track Proceedings*, pp. 1–14, 2016.

[35] Y. He, J. Lin, Z. Liu, H. Wang, L. J. Li, and S. Han, “AMC: AutoML for model compression and acceleration on mobile devices,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11211 LNCS, pp. 815–832, 2018, doi: 10.1007/978-3-030-01234-2_48.

[36] A. G. Howard *et al.*, “MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications,” Apr. 16, 2017. http://arxiv.org/abs/1704.04861

[37] F. Chollet, “Xception: Deep learning with depthwise separable convolutions,” *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 1800–1807, 2017, doi: 10.1109/CVPR.2017.195.

[38] A. Howard, W. Wang, G. Chu, L. Chen, B. Chen, and M. Tan, “Searching for MobileNetV3 Accuracy vs MADDs vs model size,” *International Conference on Computer Vision*, pp. 1314–1324, 2019.

[39] M. Tan *et al.*, “Mnasnet: Platform-aware neural architecture search for mobile,” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2019-June, pp. 2815–2823, 2019, doi: 10.1109/CVPR.2019.00293.

[40] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. C. Chen, “MobileNetV2: Inverted residuals and linear bottlenecks,” *arXiv*, pp. 4510–4520, 2018.

[41] T. Ahmed and N. H. N. Sabab, “Classification and understanding of cloud structures via satellite images with EfficientUNet,” 2020, [Online]. Available: http://arxiv.org/abs/2009.12931

[42] B. Baheti, S. Innani, S. Gajre, and S. Talbar, “Eff-UNet: A Novel Architecture for Semantic Segmentation in Unstructured Environment,” in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Jun. 2020, pp. 1473–1481. doi: 10.1109/CVPRW50498.2020.00187.

[43] M. R. Mathews, S. M. Anzar, R. Kalesh Krishnan, and A. Panthakkan, “EfficientNet for retinal blood vessel segmentation,” *2020 3rd International Conference on Signal Processing and Information Security, ICSPIS 2020*, pp. 4–7, 2020, doi: 10.1109/ICSPIS51252.2020.9340135.

[44] K. Coyne, “MRI: A Guided Tour.” https://nationalmaglab.org/education/magnet-academy/learn-the-basics/stories/mri-a-guided-tour

[45] D. G. Nishimura, *Principles of Magnetic Resonance Imaging*. Stanford University, 1996.

[46] P. C. Lauterbur, “Image formation by induced local interactions,” *Nature*, vol. 242, no. 5394, pp. 190–191, 1973.

[47] K. F. Cheung and R. J. Marks., “Imaging sampling below the Nyquist density without aliasing,” *JOSA A*, vol. 7, no. 1, pp. 92–105, 1990.

[48] P. Boesiger, K. P. Pruessmann, M. Weiger, and M. B. Scheidegger, “SENSE: sensitivity encoding for fast MRI.,” *Magnetic Resonance in Medicine*, vol. 42, no. 5, pp. 952–962, 1999, [Online]. Available: http://eutils.ncbi.nlm.nih.gov/entrez/eutils/elink.fcgi?dbfrom=pubmed&id=10542355&retmode=ref&cmd=prlinks

[49] M. A. Griswold *et al.*, “Generalized autocalibrating partially parallel acquisitions (GRAPPA).,” *Magnetic Resonance in Medicine*, vol. 47, no. 6, pp. 1202–10, 2002, doi: 10.1002/mrm.10171.

[50] M. Lustig, D. Donoho, and J. M. Pauly, "Sparse MRI: The application of compressed sensing for rapid MR imaging," *Magnetic Resonance in Medicine*, vol. 58, no. 6, pp. 1182–1195, 2007, doi: 10.1002/mrm.21391.

[51] C. M. Hyun, H. P. Kim, S. M. Lee, S. Lee, and J. K. Seo, "Deep learning for undersampled MRI reconstruction," *Physics in Medicine and Biology*, vol. 63, no. 13, 2018, doi: 10.1088/1361-6560/aac71a.

[52] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D*, vol. 60, pp. 259–268, 1992, doi: 10.1016/0167-2789(92)90242-F.

[53] O. Shitrit and T. Riklin Raviv, "Accelerated Magnetic Resonance Imaging by Adversarial Neural Network," in *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, 2017. doi: 10.1007/978-3-319-67558-9_4.

[54] B. Zhu, J. Z. Liu, S. F. Cauley, B. R. Rosen, and M. S. Rosen, "Image reconstruction by domain-transform manifold learning," *Nature*, vol. 555, no. 7697, pp. 487–492, 2018, doi: 10.1038/nature25988.

[55] J. Schlemper *et al.*, "dAUTOMAP: decomposing AUTOMAP to achieve scalability and enhance performance," vol. 1, no. 1, pp. 1–5, 2019, [Online]. Available: http://arxiv.org/abs/1909.10995

[56] T. Eo, H. Shin, Y. Jun, T. Kim, and D. Hwang, "Accelerating Cartesian MRI by domain-transform manifold learning in phase-encoding direction," *Medical Image Analysis*, vol. 63, p. 101689, 2020, doi: 10.1016/j.media.2020.101689.

[57] M. Akçakaya, S. Moeller, S. Weingärtner, and K. Uğurbil, "Scan-specific robust artificial-neural-networks for k-space interpolation (RAKI) reconstruction: Database-free deep learning for fast imaging," *Magnetic Resonance in Medicine*, vol. 81, no. 1, pp. 439–453, 2019, doi: 10.1002/mrm.27420.

[58] T. H. Kim, P. Garg, and J. P. Haldar, "LORAKI: Autocalibrated Recurrent Neural Networks for Autoregressive MRI Reconstruction in k-Space," pp. 1–24, Apr. 2019, [Online]. Available: http://arxiv.org/abs/1904.09390

[59] Y. Han, L. Sunwoo, and J. C. Ye, "k-Space Deep Learning for Accelerated MRI," *IEEE Transactions on Medical Imaging*, 2019.

[60] T. Du *et al.*, "Adaptive convolutional neural networks for accelerating magnetic resonance imaging via k-space data interpolation," *Medical Image Analysis*, vol. 72, p. 102098, 2021, doi: 10.1016/j.media.2021.102098.

[61] S. Wang *et al.*, "Accelerating magnetic resonance imaging via deep learning," *Proceedings - International Symposium on Biomedical Imaging*, vol. 2016-June, pp. 514–517, 2016, doi: 10.1109/ISBI.2016.7493320.

[62] J. Schlemper, J. Caballero, J. v. Hajnal, A. Price, and D. Rueckert, "A deep cascade of convolutional neural networks for MR image reconstruction," *Lecture Notes in Computer*

Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 10265 LNCS, no. c, pp. 647–658, 2017, doi: 10.1007/978-3-319-59050-9_51.

[63] K. Kwon, D. Kim, and H. Park, "A parallel MR imaging method using multilayer perceptron:," *Medical Physics*, vol. 44, no. 12, pp. 6209–6224, 2017, doi: 10.1002/mp.12600.

[64] K. Hammernik *et al.*, "Learning a variational network for reconstruction of accelerated MRI data," *Magnetic Resonance in Medicine*, vol. 79, no. 6, pp. 3055–3071, 2018, doi: 10.1002/mrm.26977.

[65] J. Schlemper *et al.*, "Nonuniform Variational Network: Deep Learning for Accelerated Nonuniform MR Image Reconstruction," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11766 LNCS, pp. 57–64, 2019, doi: 10.1007/978-3-030-32248-9_7.

[66] C. Qin, J. Schlemper, J. Caballero, A. N. Price, and J. v Hajnal, "Convolutional Recurrent Neural Networks for Dynamic MR Image Reconstruction," pp. 1–11, 2017.

[67] K. Lønning, P. Putzky, and M. Welling, "Recurrent Inference Machines for Accelerated MRI Reconstruction," *Int. Conf. Medical Imaging with Deep Learning*, no. Midl 2018, pp. 1–11, 2018, [Online]. Available: https://www.narcis.nl/publication/RecordID/oai:pure.knaw.nl:publications%2F1b3afd87-4c94-4766-848f-b1f2bd154933

[68] D. Lee, J. Yoo, and J. C. Ye, "Deep artifact learning for compressed sensing and parallel MRI," 2017, [Online]. Available: http://arxiv.org/abs/1703.01120

[69] D. Narnhofer, K. Hammernik, F. Knoll, and T. Pock, "Inverse GANs for accelerated MRI reconstruction," no. September 2019, p. 45, 2019, doi: 10.1117/12.2527753.

[70] T. Eo, Y. Jun, T. Kim, J. Jang, H. J. Lee, and D. Hwang, "KIKI-net: cross-domain convolutional neural networks for reconstructing undersampled magnetic resonance images," *Magnetic Resonance in Medicine*, vol. 80, no. 5, pp. 2188–2201, 2018, doi: 10.1002/mrm.27201.

[71] X. Mao, C. Shen, and Y.-B. Yang, "Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections," *Advances in neural information processing systems*, vol. 29, pp. 2802–2810, 2016.

[72] K. H. Jin, M. T. McCann, E. Froustey, and M. Unser, "Deep Convolutional Neural Network for Inverse Problems in Imaging," *IEEE Transactions on Image Processing*, vol. 26, no. 9, pp. 4509–4522, Sep. 2017, doi: 10.1109/TIP.2017.2713099.

[73] T. Eo, Y. Jun, T. Kim, J. Jang, H. J. Lee, and D. Hwang, "KIKI-net: cross-domain convolutional neural networks for reconstructing undersampled magnetic resonance

images," *Magnetic Resonance in Medicine*, vol. 80, no. 5, pp. 2188–2201, 2018, doi: 10.1002/mrm.27201.

[74] R. Souza and R. Frayne, "A hybrid frequency-domain/image-domain deep network for magnetic resonance image reconstruction," *Proceedings - 32nd Conference on Graphics, Patterns and Images, SIBGRAPI 2019*, pp. 257–264, 2019, doi: 10.1109/SIBGRAPI.2019.00042.

[75] R. Souza, R. M. Ca, R. M. Lebel, R. Frayne, and R. Ca, "A Hybrid, Dual Domain, Cascade of Convolutional Neural Networks for Magnetic Resonance Image Reconstruction," *Proceedings of Machine Learning Research*, vol. 102, pp. 437–446, 2019.

[76] E. P. Simoncelli, H. R. Sheikh, A. C. Bovik, and Z. Wang, "Image quality assessment: From error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.

[77] V. Iglovikov and A. Shvets, "TernausNet: U-Net with VGG11 Encoder Pre-Trained on ImageNet for Image Segmentation," 2018, [Online]. Available: http://arxiv.org/abs/1801.05746

[78] N. Rahaman *et al.*, "On the spectral bias of neural networks," *36th International Conference on Machine Learning, ICML 2019*, vol. 2019-June, no. 1, pp. 9230–9239, 2019.

[79] Z. Q. J. Xu, Y. Zhang, and Y. Xiao, "Training behavior of deep neural network in frequency domain," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11953 LNCS, pp. 264–274, 2019, doi: 10.1007/978-3-030-36708-4_22.

[80] R. Souza *et al.*, "An open, multi-vendor, multi-field-strength brain MR dataset and analysis of publicly available skull stripping methods agreement," *NeuroImage*, vol. 170, pp. 482–494, 2018, doi: 10.1016/j.neuroimage.2017.08.021.

[81] M. Uecker *et al.*, "Software toolbox and programming library for compressed sensing and parallel imaging," 2013.

Vita

Tahsin Rahman was born in Dhaka, Bangladesh on November 11, 1990. He graduated from Maple Leaf International School in 2008 after completing O and A levels and pursued a bachelor's degree in Electronics and Telecommunication Engineering at North South University, also at Dhaka, Bangladesh.

Following the bachelor's degree, completed summa cum laude in 2016, Mr. Rahman continued to further his education by pursuing a doctoral degree in Electrical Engineering at the University of Texas at El Paso (UTEP). At UTEP, his interest in signal and image processing led him to do research with Dr. Sergio Cabrera on Magnetic Resonance (MR) image processing. In the course of pursuing his Ph.D., Mr. Rahman worked as a Graduate Teaching Assistant for the Digital Systems Design Lab as well as Assistant Instructor for Discrete Time Signals and Systems in the ECE Department at UTEP.

As part of the Research Competency section of his PhD Qualifying Examination, he completed a Master's Degree in Electrical Engineering in the summer of 2021 with a thesis on MRI acceleration through deep learning.