Accelerated Magnetic Resonance Imaging Using Deep Learning Reconstruction

Jon Andre Ottesen

Thesis submitted for the degree of
Master in Biological and Medical Physics
60 credits

Department of Physics
Faculty of Mathematics and Natural Sciences

UNIVERSITY OF OSLO

Spring 2021

Accelerated Magnetic Resonance Imaging Using Deep Learning Reconstruction

Jon Andre Ottesen

2021 Jon Andre Ottesen

Accelerated Magnetic Resonance Imaging Using Deep Learning Reconstruction

http://www.duo.uio.no/

Printed: Reprosentralen, University of Oslo

Abstract

Reducing image acquisition times while maintaining adequate image quality is an important aim in diagnostic imaging, and in magnetic resonance imaging (MRI) in particular. This has led to a large body of research in the field referred to as accelerated acquisition methods whereby images are reconstructed from undersampled data while maintaining sufficient image quality. Recently, use of deep learning has emerged as a promising approach for reconstruction of sparse data and the topic of this Master's thesis was to investigate acceleration of MRI using modern deep learning methods to reconstruct accelerated images. Several approaches have been studied, varying from simple yet proficient baseline models, to more complex methods involving advanced reconstruction schemes and the inclusion of new architectural advances from the deep learning research community. Emphasis has been put on supplementing the different models and methods using the architectural recommendations of the deep learning research community.

Simulated 4x and 8x accelerated MRI scans were produced from fully sampled *k*-space data by removing phase encoding lines. The original fully sampled images were reconstructed from the undersampled images using deep learning networks trained on pairs of original and undersampled images on a known dataset. The learning process minimizes the mean absolute error and the structural similarity index measure (SSIM) of the network's output relative to the fully sampled image.

On the test set, the baseline U-Net model achieved SSIMs of 0.944 and 0.917 for the 4x and 8x accelerated images, respectively, thus outperforming the current gold-standard of sparse sampling methods with SSIMs of 0.900 and 0.870 for the same 4x and 8x acceleration factors, respectively. Without changing the method of reconstruction, the architectural recommendations from the deep learning community including skip connections, squeeze-and-excitation and length scaling improved the reconstruction baseline model to SSIMs of 0.947 and 0.920 for 4x and 8x acceleration, respectively.

The high-end reconstruction schemes involving direct *k*-space manipulation showed improved accuracy, achieving SSIMs of 0.954 and 0.931 for the 4x and 8x accelerated images, respectively. Further accuracy improvements were observed after the inclusion of model refinements including skip connections, squeeze-and-excitation, and a different activation function, resulting in SSIMs of 0.956 and 0.935 for the 4x and 8x accelerated images, respectively.

It is concluded that deep learning based reconstruction of undersampled MRI data can provide images with quality close to the fully sampled reference for a four-fold acceleration factor. The reconstruction results achieved were comparable to state-of-the-art methods reported in literature. Nonetheless, the proposed methods need to be validated in clinically relevant patient cohorts.

Acknowledgments

First, I would like to thank my main supervisor, Atle Bjørnerud for your excellent guidance and support. Thank you for both allowing me to figure things out at my own pace, whether slow of fast, but also making sure I stayed on track. Even if I ended up crashing the GPU mini-server a tad too much. Second, I would like to thank my co-supervisor, Inge Groote for some extremely detailed proofreading and the suggestion for this to be my Master's thesis. A big thanks to the both of you for the many valuable suggestions and comments on my draft, it really made a huge difference. The two of you have really been a big help and have helped me make sure everything went fine in the end. I would also like to thank the wonderful people at CRAI for the many good discussions and lunch talks, and Ivar for making sure the server is up running even after my many crashes.

Next, I would like to offer a big thanks some of my fellow physicist and good friends Frida, Nils, Simen, Anders, Aron and Kristina for reading through my thesis and offering some great advice and proofreading. Especially, I would like to thank Frida, Nils and Simen for all the hours spent proofreading my thesis.

Finally, a thank you to my family for their support. A big thanks to mom for always making sure everything is fine and picking me up in Oslo for the last years whenever I wanted to come visit. Thank you to both of my grandparents, whether it is making sure my refrigerator is always stacked full, going out to fish or just talking.

Contents

List of abbreviations

ADAM Adaptive moment estimation

BART Berkeley advanced reconstruction toolbox

BiFPN Bi-directional feature pyramid network

CNN Convolutional neural network

EPI Echo planar imaging

ETL Echo train length

FLAIR Fluid-attenuated inversion recovery

FOV Field-of-view

GRAPPA Generalized autocalibrating partial parallel acquisition

GRE Gradient echo

MAE Mean absolute error

MRI Magnetic resonance imaging

MS-SSIM Multiscale SSIM

MSE Mean squared error

NMSE Normalized mean squared error

PSNR Peak signal-to-noise ratio

ReLU Rectified linear unit

RF Radiofrequency

RSS Root-sum-of-squares

SE Spin echo

SENSE Sensitivity encoding

SiLU Sigmoid linear unit

SNR signal-to-noise ratio

SSIM Structural similarity index measure

VarNet Variational network

Chapter 1

Introduction

Vivere est cogitare

— Marcus Tullius Cicero, *Tusculanae Disputationes*

Fifty years ago, Paul Lauterbur had the idea to encode the spatial distribution of an object's property in all three dimensions by applying magnetic field gradients. Using this technique, he first imaged the water contents of water tubes [1, 2], and later in 1974 he imaged a pine branch, a clam and the thoracic cavity of a mouse [3]. Finally in 1977, Damadian and colleagues performed the first successful magnetic resonance imaging (MRI)-scan on a human [4].

By any standard, MRI is a powerful imaging modality, offering unprecedented degrees of soft tissue contrast and spatial resolution without exposure to ionizing radiation. MRI has become essential for clinical evaluation in regards to soft tissue, such as in neuroimaging and cardiac imaging. The biophysical principle of MRI dictates a trade-off between achievable image quality and required acquisition time. Typical acquisition times on modern MRI systems are in the range 1-10 minutes, depending on sequence type and indication. Excessively long acquisition times may result in unwanted features in the image (artifacts) due to patient motion and poor patient compliance. Hence, finding methods to reduce overall scan times while maintaining sufficient image quality has been a major challenge and research focus in MRI since the very beginning.

Accelerated MRI has been a long running avenue of research within the MRI research community. Most advances can be categorized as either or both hardware and sequence related, achieving accelerated imaging time by faster signal acquisition [5]. Despite this, fundamental physical constraints limit the rate of signal acquisition, to that end, significant effort has been made to reduce the total sampled signal while maintaining image quality [6]. However, the required amount of samples are directly proportional to the resolution and field-of-view (FOV) [7, 8]. Hence, by the Nyquist-Shannon theorem [9], reducing the total number of samples could introduce multiple aliasing artifacts such as wrap-around and truncation error [10, 11]. To counteract this, multiple acceleration techniques have been devised to exploit the inherent redundancy in the sampled signal such as partial Fourier [12], parallel imaging [6] and sparse sampling [13].

There has recently been a significant progress in the field of artificial neural networks, and especially in the subcategory of deep learning. This progress includes the various fields of image classification [14], object localization [15], segmentation [16] and super-resolution [17, 18]. From this rapid improvement in deep learning, multiple research groups have raised the question: Can deep learning be used to successfully reconstruct a clinically viable image from undersampled data, outperforming current paradigms? For this reason a surge of deep learning-based MRI reconstruction methods have been proposed [19, 20, 21, 22, 23]; the best of which require enormous hardware resources to train [23]. The initial results suggest the possibility for excellent reconstructive models within reach, outperforming the current state-of-the art sparse sampling methods at 4x and 8x acceleration, from both a numerical standpoint and by a working radiologists in a 2019 study, albeit still lacking in proper clinical viability [24]. However, the field of deep learning accelerated MRI is still very much in its early stages, but the need for high-end high-cost hardware makes front-end research in this field challenging for small sized research groups with limited hardware requirements.

To this end, this thesis has three goals:

1. To train a simple non-hardware intensive acceleration deep learning model;

2. To improve upon the model by applying fresh ideas from the deep learning community not yet used for deep learning accelerated MRI;

3. To apply these models in a more hardware-intensive reconstruction framework.

Chapter 2

Theory

Nobody knows why, but the only theories which work are the mathematical ones.

— Michael Holt, *Mathematics in Art*

2.1 Spin and magnetization

2.1.1 Spin

In quantum mechanics, elementary particles carry intrinsic angular momentum spin $\boldsymbol{S}$[1] and extrinsic angular momentum $\boldsymbol{L}$ [25, pg. 171]. Spin ($\boldsymbol{S}$) is an intrinsic quantity similar to the mass and charge, whereas angular momentum is dependent on the properties of the system. Unlike other intrinsic quantities such as the charge, spin is a vector quantity ($\boldsymbol{S}$). However, by the generalized uncertainty principle, only one of the Hermitian spin operators $\hat{S}_x$, $\hat{S}_y$ and $\hat{S}_z$ can be known at any given time, other measurements collapse the wave function and previous measurements are made obsolete [25, pg. 106, 111 and 171][2]. Nonetheless, the techniques used in MRI only require the knowledge of a single spin component, in addition to the size of the total spin $\hat{S}^2 = \hat{S_x}^2 + \hat{S_y}^2 + \hat{S_z}^2$ (which can be measured simultaneously with a spin component) [25, pg. 161 and 171].

The measurement by a Hermitian spin operator on some system (e.g., protons or nuclei) yields one of the possible spin eigenvalues. The system consequently collapses into the corresponding eigenstate. In general, measurements with Hermitian operators on some system will always return one of the possible eigenvalues corresponding to its eigenstates [25, pg. 106]. For a general spin eigenstate, the measurements are [25, pg. 171]

$$\begin{aligned}\hat{S}^2 \left|I, m_s\right\rangle &= \hbar^2 I(I+1) \left|I, m_s\right\rangle , \\ \hat{S}_z \left|I, m_s\right\rangle &= \hbar m_s \left|I, m_s\right\rangle ,\end{aligned}$$

[1] Bold typefont indicates vector notation

[2] This is because the spin operators $\hat{S}_x$, $\hat{S}_y$ and $\hat{S}_z$ do not commute, much like position and momentum.

where I and m_s are quantum numbers and given by

$$I = 0, \frac{1}{2}, 1, \frac{3}{2}, 2, \frac{5}{2}, 3, ..., \tag{2.1.1a}$$

$$m_s = -I, -I+1, ..., 0, ..., I-1, I. \tag{2.1.1b}$$

The quantum number I is the maximum possible spin in any direction and is often referred to as the spin of the particle [25, pg. 172]. The quantum number m_s is the spin magnitude along the measured axis.

2.1.2 Magnetic moment and net magnetization

The magnetic moment of a particle is proportional to the spin, and is given by

$$\boldsymbol{\mu} = \gamma \boldsymbol{S},$$

where γ is a proportionality constant called the *gyromagnetic* ratio [25, pg. 178]. The gyromagnetic ratio is uniquely identified for each nuclear isotope possessing a spin [7, pg. 7].

For a sample of N particles, the net magnetization is given by

$$\boldsymbol{M} = \sum_{i=1}^{N} \boldsymbol{\mu_i}, \tag{2.1.2}$$

that is, the sum of all the magnetic moments for every particle [7, pg. 7]. In a uniform system with no directional spin preference, the net magnetization is zero, as the direction of the spins are randomly distributed.

2.1.3 Particles in a magnetic field

A magnetic moment under the influence of an external magnetic field experiences a spin dependency in the energy. As such, the Hamiltonian is extended by a linear term on the form [25, pg. 179]

$$\hat{H} = -\hat{\boldsymbol{\mu}} \cdot \boldsymbol{B}, \tag{2.1.3}$$

where $\hat{\boldsymbol{\mu}} = \hat{S}_x \boldsymbol{e}_x + \hat{S}_y \boldsymbol{e}_y + \hat{S}_z \boldsymbol{e}_z$ and $\boldsymbol{e}_i$ is the unit vector in the i^{th} direction.

A simple but important case is for a time-independent magnetic field $\boldsymbol{B} = B_z \boldsymbol{e}_z$ directed along the z-axis, which simplifies equation (2.1.3) to

$$\hat{H} = -\gamma B_z \hat{S}_z. \tag{2.1.4}$$

The eigenvalues of the Hamiltonian in equation (2.1.4) when acting on spin eigenstates are given by

$$\hat{H} \left| I, m_s \right\rangle = -\gamma B_z \hat{S}_z \left| I, m_s \right\rangle = -\gamma B_z \hbar m_s \left| I, m_s \right\rangle.$$

The energy of a particle in an external magnetic field is therefore given by

$$E_{m_s} = -\gamma B_z \hbar m_s. \tag{2.1.5}$$

Equation (2.1.5) is often written as

$$E_{m_s} = -\hbar\omega_0 m_s, \tag{2.1.6}$$

where

$$\omega_0 = \gamma B_z \tag{2.1.7}$$

is known as the *Larmor frequency* [25, pg. 180]. As evident from equation (2.1.6), spins aligned parallel to the magnetic field have lower overall energy.

From equation (2.1.6), particles with spin-1/2 in a magnetic field have two possible quantized energy levels:

$$E_{1/2} = -\frac{\hbar\omega_0}{2} \quad \text{(parallel alignment),} \quad \text{and} \tag{2.1.8a}$$

$$E_{-1/2} = +\frac{\hbar\omega_0}{2} \quad \text{(anti-parallel alignment).} \tag{2.1.8b}$$

The difference in energy given by

$$\Delta E = E_{-1/2} - E_{1/2} = \hbar\omega_0,$$

and the split in energy levels is depicted in figure 2.1 for the spin-1/2-state.

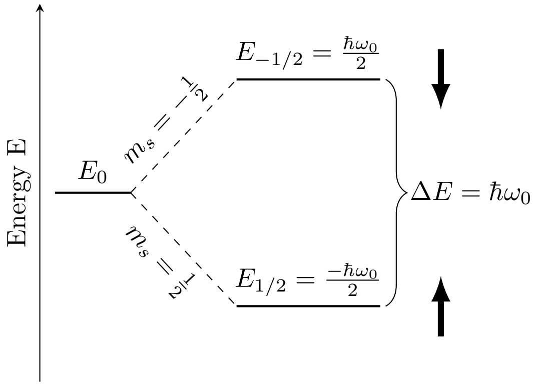

Energy E
E_0
$m_s = -\frac{1}{2}$
$E_{-1/2} = \frac{\hbar\omega_0}{2}$
$m_s = \frac{1}{2}$
$E_{1/2} = \frac{-\hbar\omega_0}{2}$
$\Delta E = \hbar\omega_0$

Figure 2.1: The split in energy for the spin-up/spin-down state placed in an external magnetic field.

2.1.4 Time evolution for the spin-1/2 system in a magnetic field

As the spin-1/2 system is important for MRI, it is necessary to know how it evolves in time. In particular, since MRI is based on measuring large ensembles of spins, it is necessary to know the time evolution of the spin expectation values.

Consider a semi-general spin-1/2-system given by

$$|\psi\rangle = \cos(\alpha/2)\,|\uparrow\rangle + \sin(\alpha/2)\,|\downarrow\rangle\,,$$

with $|\uparrow\rangle$ and $|\downarrow\rangle$ being the spin up and spin down eigenstates, respectively, and α a real number [25, pg. 179-180]. The time-dependent state is given by

$$|\psi(t)\rangle = \cos(\alpha/2)\,|\uparrow\rangle\, e^{i\omega_0 t/2} + \sin(\alpha/2)\,|\downarrow\rangle\, e^{-i\omega_0 t/2},$$

where the exponents are determined by the energies in equations (2.1.8a) and (2.1.8b) [25, pg. 179-180]. It can be shown that the time-dependent expectation values in the x, y and z directions are given by [25]

$$\langle S_z \rangle = \frac{\hbar}{2}\cos(\alpha), \tag{2.1.9a}$$

$$\langle S_x \rangle = \frac{\hbar}{2}\sin(\alpha)\cos(\omega_0 t), \tag{2.1.9b}$$

$$\langle S_y \rangle = \frac{\hbar}{2}\sin(\alpha)\cos(\omega_0 t). \tag{2.1.9c}$$

As shown in figure 2.2, equations (2.1.9a), (2.1.9b) and (2.1.9c) are analogous to a magnetic dipole precessing around the z-axis with angular frequency ω_0.

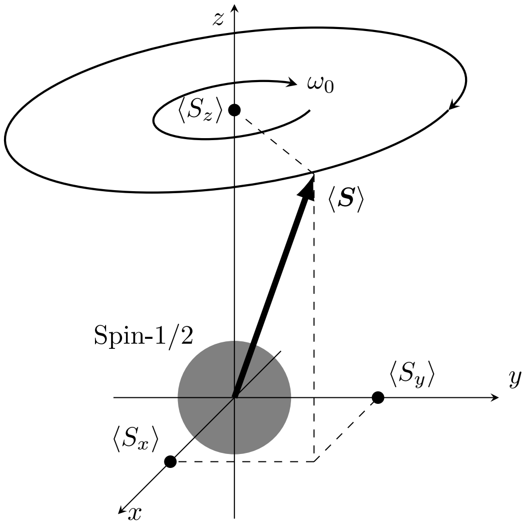


Figure 2.2: Precession of $\langle \boldsymbol{S} \rangle$ for a spin-1/2 system with angular frequency ω_0 in a uniform magnetic field.

2.1.5 Spin distribution of nuclei

For a system in thermal equilibrium, the probability distribution for the different energy states is given by

$$P_n = \frac{e^{-\frac{E_n}{k_B T}}}{\sum_{i=1}^{N} e^{-\frac{E_i}{k_B T}}}, \tag{2.1.10}$$

where k_B is the Boltzmann factor, T is the temperature in Kelvin, E_n is the energy for the n^{th} energy state and N is the number of energy states [7, pg. 7]. For some arbitrary spin I, the possible energy states are given in equation (2.1.6), and the corresponding probability distribution for some spin state is given by

$$P_{m_s} = \frac{e^{\frac{\hbar\omega_0 m_s}{k_B T}}}{\sum_{s=-I}^{I} e^{\frac{\hbar\omega_0 m_s}{k_B T}}}. \tag{2.1.11}$$

For the special case of spin-1/2, the fraction of spin-up/spin-down is given by

$$\frac{P_{1/2}}{P_{-1/2}} = e^{\frac{\hbar\omega_0}{k_B T}}. \tag{2.1.12}$$

As evident by equations (2.1.11) and (2.1.12), it is more likely for a nuclei to align itself parallel with the magnetic field rather than anti-parallel, and moreover, the probability increases with increasing magnetic field $\omega_0 = \gamma B_0$. For the perpendicular x-y plane there is no branching energy levels as depicted in figure 2.1. For that reason, the distribution in equation (2.1.10) dictates that all directions in the x-y plane are equally probable, as such, there will be no net magnetization.

2.1.6 The net magnetization

The net magnetization defined in equation (2.1.2) is not intuitive, and the sheer number of particles in human tissue makes it inapplicable. However, using the first order Taylor approximation, it can be shown that the net magnetization for a macroscopic sample is given by [7, pg. 8]

$$M_z \approx N_0 \frac{\hbar^2\gamma^2 I(I+1)}{3k_B T}\boldsymbol{B}_0. \tag{2.1.13}$$

Equation (2.1.13) has two important consequences:

1. The net magnetization in a sample is directly proportional to the magnetic field $\boldsymbol{B}_0$, and a larger magnetization generates a greater MRI signal [7, pg. 8].

2. The net magnetization is proportional to the square of the gyromagnetic ratio γ^2. In vivo, ^{1}H is the isotope with the highest gyromagnetic ratio, coupled with the abundance of water in human tissue, this makes protons the most detectable nuclei in MRI [7, pg. 8].

2.2 Magnetic resonance imaging

The entire premise of MRI is to measure the net magnetization of nuclei under the influence of an external magnetic field B_0 parallel to the z-axis, or explicitly: to detect a transverse signal in the x-y plane created from the B_0-field. However, without any external influence except a time-independent magnetic field B_0, no net transverse signal exists. To produce MR-images, the net magnetization must be excited into the transverse plane by a magnetic field in the transverse plane. Thereafter, position dependency is encoded into the transverse signal by additional position dependent magnetic gradients directed parallel to the z-axis, but dependent on x, y and z separately.

A general MRI setup with the coordinate axis, a magnetic field and a receiver coil is illustrated in figure 2.3. The receiver coil is used for recording the outgoing transverse signal.

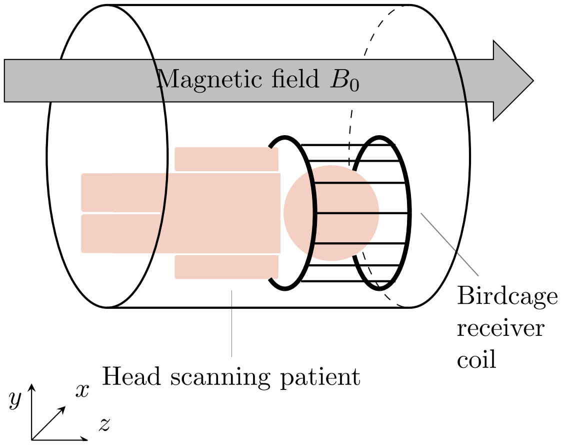


Figure 2.3: A simplified MRI head scan with coordinate axis, the magnetic field and an illustration of a birdcage receiver coil.

2.2.1 Spin magnetization as classical quantities

To compare quantum mechanical spin to classical spin is just an analogy. However, Ehrenfest's theorem guarantees that the spin expectation value $\langle \boldsymbol{S} \rangle$ evolves according to classical laws [25, pg. 180]. In MRI, large samples of spins are measured, as such, we are effectively measuring the expectation value of the spins and the magnetic moment. By this, the magnetic moment and spin can be discussed as would have been in a classical sense with a specific direction and movement.

2.2.2 The Bloch equation

The Bloch equation is one of the most fundamental equations in magnetic resonance, describing the change of magnetization under the influence of an external magnetic field. For the human body, the Bloch equation describes the internal magnetization within the body, and is given by [26]

$$\frac{d\boldsymbol{M}}{dt} = \gamma \boldsymbol{M} \times \boldsymbol{B}, \tag{2.2.1}$$

where γ is the gyromagnetic ratio (nuclei dependent), $\boldsymbol{M} = [M_x, M_y, M_z]$ is the magnetization vector and $\boldsymbol{B} = [B_x, B_y, B_z]$ denotes the magnetic field vector [8, pg. 55]. Equation (2.2.1) states that the magnetic moment experiences a torque perpendicular to both the magnetization and the magnetic field [26].

The Bloch equation (2.2.1) gives a complete description of the time-evolution of the magnetization under the influence of a magnetic field. However, it does not account for particle-particle interactions and interactions with the surrounding tissue: "Fortunately the coupling of the nuclear spins mutually and with the surrounding matter is weak, which allows a classical treatment on the basis of the Bloch equation extended with terms describing the relaxation in a phenomenological way" (Vlaardingerbroek [8, pg. 55]).

2.2.2.1 Rotating frame of reference

To simplify future calculations, the observed frame of reference in the Bloch equation (2.2.1) can be interchanged. From the laboratory frame of equation (2.2.1), i.e., the patient's non-rotating frame of reference, to a rotational frame of reference with angular velocity Ω around the z-axis. The Bloch equation in the rotational frame of reference is given by [7, pg. 10]

$$\frac{d\boldsymbol{M'}}{dt} = \gamma \boldsymbol{M'} \times \left(\boldsymbol{B'} - \frac{\Omega}{\gamma}\Omega \boldsymbol{e_z} \right).^{3} \tag{2.2.2}$$

Equation (2.2.2) can be simplified to

$$\frac{d\boldsymbol{M'}}{dt} = \gamma \boldsymbol{M'} \times \boldsymbol{B'}_{eff}, \tag{2.2.3}$$

where $\boldsymbol{B'}_{eff}$ is the effective magnetic field given by $\boldsymbol{B'}_{eff} = \boldsymbol{B'} - \frac{\Omega}{\gamma}\Omega\boldsymbol{e}_z$. [7, pg. 10]. Note that equation (2.2.3) implies a stationary magnetization in the rotating frame of reference at the Larmor frequency when $\Omega = \gamma B_0$.

2.2.3 External magnetic field

The magnetic field imposed during an MRI scan can be divided into three parts [8, pg. 56]:

1. The main magnetic field $B_0\boldsymbol{e}_z$ responsible for creating the net magnetization. When talking about MRI scanners, terms like 3 T or 1.5 T scanners refers to the strength (magnetic flux density in units of Tesla) of the B_0-field [8, pg. 56].
2. The magnetic field gradients $(\boldsymbol{G}(t) \cdot \boldsymbol{r})\boldsymbol{e}_z = (G_x x + G_y y + G_z z)\,\boldsymbol{e}_z$ responsible for position encoding [8, pg. 55-84].
3. The $\boldsymbol{B}_1$-field perpendicular to the B_0-field responsible for exciting the net magnetization into the transverse plane [7, pg. 33-34].

A treatment of the magnetic field using these three terms is an approximation and can lead to potential errors. There exists local inhomogeneities δB caused by the scanner-magnet, and correctional terms to the magnetic field gradients since magnetic field lines cannot be entirely linear [8, pg. 56 and 151].

A general expression for the external magnetic field in an MRI sequence can be written as

$$\boldsymbol{B}(\boldsymbol{r}, t) = B_0\boldsymbol{e}_z + (\boldsymbol{G}(t) \cdot \boldsymbol{r})\,\boldsymbol{e}_z + \boldsymbol{B}_1(t). \tag{2.2.4}$$

The time-dependency depicted in equation (2.2.4) for the magnetic gradients are on the form of trapezoid functions [8, pg. 61 and 67]. This time-dependency allows the fields to be turned off and on as pleased during a sequence, except with a small ramp-up period at the beginning and end. For simplicity, the trapezoid time-dependency can and will be approximated using rectangular functions.

[3] Primed variables ' denote the rotating frame of reference.

The time-dependency in the $\boldsymbol{B}_1(t)$-field is not solely from the rectangular function. In addition, the $\boldsymbol{B}_1(t)$-field is an oscillating magnetic field that can be activated when needed with a chosen range of angular frequencies $\Delta\omega$. This range of frequencies are related to the gradient properties and the desired spatial encoding of the signal [8, pg. 58-61, 7, pg. 33-34]. However, with the use of the rotational frame, this field can be viewed as time-independent when only consisting of a single frequency [7, pg. 33-34].

2.2.4 Magnetization in an external magnetic field

The magnetic field defined in equation (2.2.4) has an inherent time-dependency, which can be bypassed by choosing specific time intervals with the rectangular function approximation or using the rotational frame of reference. As such, it is of interest to study the time evolution of the internal magnetization in a time-independent magnetic field

$$\boldsymbol{B} = B_z(x,y,z)\boldsymbol{e}_z. \tag{2.2.5}$$

Consider the time-independent magnetic field defined in equation (2.2.5), the Bloch equation for the laboratory frame (2.2.1) is given by

$$\frac{d\boldsymbol{M}}{dt} = \gamma \begin{vmatrix} \boldsymbol{e}_x & \boldsymbol{e}_y & \boldsymbol{e}_z \\ M_x & M_y & M_z \\ 0 & 0 & B_z \end{vmatrix} = \gamma \left(M_y B_z \boldsymbol{e}_x - M_x B_z \boldsymbol{e}_y \right).$$

The time derivatives in component form are given by[4]

$$\begin{aligned} \dot{M}_x &= \gamma M_y B_z, \\ \dot{M}_y &= -\gamma M_x B_z, \\ \dot{M}_z &= 0. \end{aligned}$$

The solutions for this system of ordinary differential equations are given by

$$M_x = C(\boldsymbol{r})e^{i\gamma B_z t} + D(\boldsymbol{r})e^{-i\gamma B_z t}, \tag{2.2.7a}$$

$$M_y = iC(\boldsymbol{r})e^{i\gamma B_z t} - iD(\boldsymbol{r})e^{-i\gamma B_z t}, \tag{2.2.7b}$$

$$M_z = f(\boldsymbol{r}), \tag{2.2.7c}$$

with $C(\boldsymbol{r}), D(\boldsymbol{r})$ and $f(\boldsymbol{r})$ functions of x, y and z. These solutions are on the same form as equation (2.1.9a), (2.1.9b) and (2.1.9c), i.e., a magnetic dipole precessing around the z-axis as illustrated in figure 2.2.

The transverse magnetization detected during an MRI scan can be defined by [7, pg. 34, 8, pg. 57]

$$M_T = M_x + iM_y.$$

[4] Dotted variables denote time derivatives $\dot{x} = \frac{dx}{dt}$.

As equations (2.2.7a) and (2.2.7b) are general solutions for a time-independent magnetic field, the transverse magnetization is given by

$$\begin{aligned} M_T(t, \boldsymbol{r}) &= 2D(\boldsymbol{r})e^{-i\gamma B_z t} && (2.2.8a) \\ &= M_T(0, \boldsymbol{r})e^{-i\gamma B_z t} && (2.2.8b) \\ &= M_T(0, \boldsymbol{r})e^{-i\omega_0 t}, && (2.2.8c) \end{aligned}$$

where $M_T(0)$ is a constant determined by boundary conditions and ω_0 is the Larmor frequency as defined in equation (2.1.7). For a time-dependent field along the z-axis, the transverse magnetization is given by [8, pg. 58]

$$M_T(t, \boldsymbol{r}) = M_T(0, \boldsymbol{r})e^{-i\gamma \int_0^t B_z(\tau)d\tau}. \tag{2.2.9}$$

2.2.5 Slice selection

As stated in section 2.1.5 there is no net magnetization in the transverse plane, as there is no magnetic field in the transverse plane, and therefore no MRI signal. This problem is solved by an excitation pulse named the radiofrequency (RF) pulse, which is the $\boldsymbol{B}_1'(t)$-field from equation (2.2.4) [7, pg. 33-37]. When the RF-pulse is used in conjunction with the magnetic gradient G_z, slice selective imaging in MRI is achieved, generating a transverse magnetization from the longitudinal magnetization for specific desired ranges of z-values along a volume [7, pg. 33-34].

The addition of the magnetic gradients introduce a position dependency in the Larmor frequency ω_0, enforcing a position dependency in the angular frequency of the precession around the z-axis from equation (2.2.8c). As such, under the influence of the G_z-gradient, the Larmor frequency is given by

$$\omega_{G_z} = \gamma \left(B_0 + G_z \cdot z\right). \tag{2.2.10}$$

If the Larmor frequency is chosen as the angular frequency in the rotational frame of reference at $z = z_0$, the effective magnetic field defined in equation (2.2.3) is given by

$$\boldsymbol{B}_{eff}' = [B_1', 0, G_z \cdot (z - z_0)], \tag{2.2.11}$$

with the B_1'-field chosen along the x'-direction. Since the Bloch equation states that the magnetization vector experiences a force perpendicular to both the magnetic field and the magnetization, the magnetization is excited into the transverse plane (y' in this specific case). However, for $G_z \cdot |z - z_0| >> B_1$, the magnetic field is essentially parallel to the z-axis and no torque is experienced [7, pg. 33-34]. Exciting only the section at $z \approx z_0$ into the transverse plane [7, pg. 33-34].

The frequency of the RF-pulse contains a bandwidth $\Delta\omega$ of angular frequencies, exciting thin slices at $z = z_0$ with width

$$\Delta z = \frac{\Delta\omega}{\gamma G_z}, \tag{2.2.12}$$

where the width of the slices (slice thickness) can be adjusted by either adjusting the gradient strength or the frequency bandwidth [7, pg. 34]. The gradients used during the excitation process are often referred to as *slice selective gradients* [7, pg. 33].

The introduction of a gradient along the z-axis added a z-dependency in the magnetic field, introducing a phase shift for the spin population along the z-axis (as the frequency was higher for larger z-values). This phase shift causes a de-phasing of the spins along the z-axis, and can be counteracted with the same G_z-gradient with the opposite sign applied for half the time of the excitation pulse [8, pg. 60-61].

2.2.5.1 Flip angle

The angle the longitudinal magnetization M_z is excited into the transverse plane is referred to as the flip angle α_f. The chosen flip angle is sequence dependent and affects the final image contrast and total imaging time [7, pg. 11, pg. 48-65]. An illustration of how the net magnetization is affected by the RF-pulse is shown in figure 2.4 with $\alpha_f = 40°$. The excitation pulse lasts anywhere from 1.28 ms to 3.84 ms, with shorter pulses having less accurate slice profiling [27].

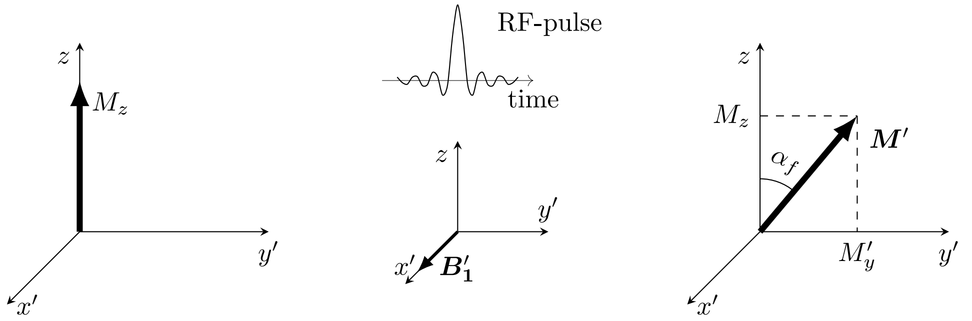


Figure 2.4: The RF-pulse exciting the longitudinal magnetization into the transverse plane with a flip angle of $\alpha_f = 40°$. The effective magnetic field is the same as in equation (2.2.11), which is why the transverse magnetization only has a transverse M_y' component.

2.2.6 Phase encoding

After a slice is excited at, e.g., $z = z_0$ into the transverse plane, the x and y positions can be encoded by the magnetic gradients G_x and G_y. For the excited slice, the internal magnetization is described by equation (2.2.9) for a time-dependent magnetic field. For a field with the two magnetic gradients (G_x and G_y) and the B_0-field, the transverse magnetization is given by

$$M_T(t, \boldsymbol{r}) = M_T(0, \boldsymbol{r}) e^{-i\omega_0 t - i\gamma \int_0^t G_x(\tau)\cdot x + G_y(\tau)\cdot y d\tau},$$

where the B_0-field is expressed with the Larmor frequency $\omega_0 = \gamma B_0$, and τ is a dummy integration variable. In the rotating frame of reference, the expression is simplified using the effective magnetic field (2.2.3) and is given by [7, pg. 37]

$$M_T'(t, \boldsymbol{r}) = M_T'(0, \boldsymbol{r}) e^{-i\gamma \int_0^t G_x(\tau)\cdot x + G_y(\tau)\cdot y d\tau}. \tag{2.2.13}$$

2.2.6.1 Phase shift

In the rotating frame of reference with angular frequency $\omega_0 = \gamma B_0$, the transverse magnetization M_T' is entirely static and non-rotating in the absence of additional magnetic fields. However, with non-zero gradients the magnetization experiences spatial frequencies given by [7, pg. 39, 8, pg. 67]

$$k_x = \gamma \int_0^t G_x(\tau) d\tau, \tag{2.2.14a}$$

$$k_y = \gamma \int_0^t G_y(\tau) d\tau. \tag{2.2.14b}$$

Notice how the spatial frequencies are dependent on both the gradient strength and the time the gradient is applied. Under the assumption of rectangular functioned gradients, the spatial frequencies are given by [8, pg. 67]

$$k_x = \gamma \int_0^t G_x(\tau) d\tau = \gamma G_x T_x, \tag{2.2.15a}$$

$$k_y = \gamma \int_0^t G_y(\tau) d\tau = \gamma G_y T_y, \tag{2.2.15b}$$

where $T_{x,y}$ is the time the gradients are applied between $[0, t]$. G_x and G_y are the gradient strengths when the gradient is turned on.

The spatial frequencies in equations (2.2.14a) and (2.2.14b) introduce a position dependent phase shift on the previously static transverse magnetization. This phase shift or phase angle is given by [7, pg 37-38]

$$\alpha_p(t, x, y) = \gamma \int_0^t G_x(\tau) d\tau \cdot x + \gamma \int_0^t G_y(\tau) d\tau \cdot y = k_x \cdot x + k_y \cdot y. \tag{2.2.16}$$

As neither of the gradients are enforced positive, the phase shift can be both positive and negative. The phase angle for a sample of 5×5 spins is demonstrated in figure 2.5 for zero and non-zero k_x and k_y with varying x' and y'. Notice how the de-phasing increases for increasing spatial coordinates, and once de-phased another gradient must be applied to re-phase the spins, as the frequency in the entire sample would otherwise be uniform.

2.2.7 k-space

The spatial frequencies k_x and k_y represent coordinates or frequencies in the Fourier domain (spatial frequency domain), also known as *k-space* in MRI. Using the k-space coordinates from equations (2.2.14a) and (2.2.14b), the transverse magnetization (2.2.13) can be rewritten as [7, pg. 37, 8, pg. 67]

$$M_T'(x, y, k_x, k_y) = M_T(0, x, y) e^{-i(k_x \cdot x + k_y \cdot y)}. \tag{2.2.17}$$

However, the transverse magnetization is position dependent, and the receiver coils that are responsible for detecting the transverse signal will detect signal from the entire excited sample or slice [7,

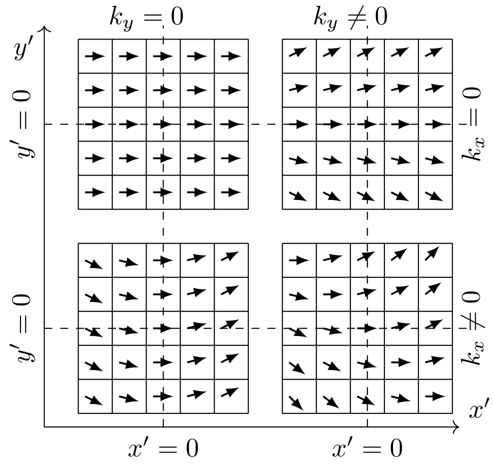


Figure 2.5: A visual representation of the phase angle α_p under the influence of constant spatial frequencies, but varying spatial coordinates x' and y'. This figure is inspired by figure 5.2 in [7].

pg. 38]. Hence, the total transverse signal detected by the receiver coils is the integrated position dependent transverse magnetization over a slice A and is given by [7, pg. 38, 8, pg 67]

$$S'_T(k_x, k_y) = \int\int_A M'_T(x, y, k_x, k_y)dxdy. \tag{2.2.18}$$

Note that the signal has no explicit time-dependency, only implicit in the spatial frequencies by k_x and k_y (see equations (2.2.14a) and (2.2.14b)). Furthermore, the detected signal is the spatial frequency distribution of the image [7, pg. 39].

A time-dependent signal can be decomposed into its frequency distribution by the Fourier transform. In a similar manner, the spatially dependent signal can be decomposed into its spatial frequency distribution by the Fourier transform

$$F(k_x, k_y) = \frac{1}{2\pi}\int\int_A f(x, y)e^{-i(k_x \cdot x + k_y \cdot y)}dxdy.$$

The inverse Fourier transform maps the frequency distribution back to its spatial distribution

$$f(x, y) = \frac{1}{2\pi}\int\int_A F(k_x, k_y)e^{i(k_x \cdot x + k_y \cdot y)}dk_x dk_y,$$

both using the angular frequency convention and the $\frac{1}{(2\pi)^{\frac{n}{2}}}$ convention where n is the integral dimension. As such, the initial transverse magnetization $M_T(0, \boldsymbol{r})$ can be recovered from the spatial distribution by the inverse Fourier transform [7, pg. 39, 8, pg. 68]

$$M_T(0, \boldsymbol{r}) = \frac{1}{2\pi}\int\int_A S'_T(k_x, k_y)e^{i(k_x \cdot x + k_y \cdot y)}dk_x dk_y. \tag{2.2.19}$$

By equation (2.2.19) the initial transverse magnetization as a function of position can be calculated when the frequency distribution is known. The absolute value of the transverse magnetization is directly proportional to the spin density [7, pg. 38]

$$\rho(x, y) \propto |M_T(0, x, y)|. \tag{2.2.20}$$

From this, MRI can be understood as a method where the magnetic gradients G_x and G_y are applied to some sample of tissue to map out the spatial frequency distribution. The sampled spatial frequency distribution is inverse Fourier transformed to the corresponding spatial distribution, i.e., the nucleic density for the imaged nuclei (often ^{1}H)).

Although equation (2.2.19) describes the relation between k-space and the spin density distribution, this is idealistic at best, and multiple assumptions have been made. First and foremost, the sampled signal is by no means continuous, and sampling the entire frequency domain is unrealistic. Instead, k-space is often sampled as a rectangular grid between $[-k_{x,max}, k_{x,max}]$ and $[-k_{y,max}, k_{y,max}]$ with $k_{x,max}$ and $k_{y,max}$ not necessarily the same [7, pg. 51]. The rectangular k-space grid is shown in figure 2.6 for a different number of samples along the k_x and k_y axis to illustrate the rectangular nature of k-space sampling.

Equation (2.2.19) is correct for the Bloch equation as defined in (2.2.3), but this definition does not hold for a non-ideal sample with particle-particle (spin-spin) interactions and local field inhomogeneities [8, pg. 85]. To account for this, the Bloch equation is extendable with additional terms as stated in subsection 2.2.2. However, any rigorous treatment of these errors is omitted in this thesis.

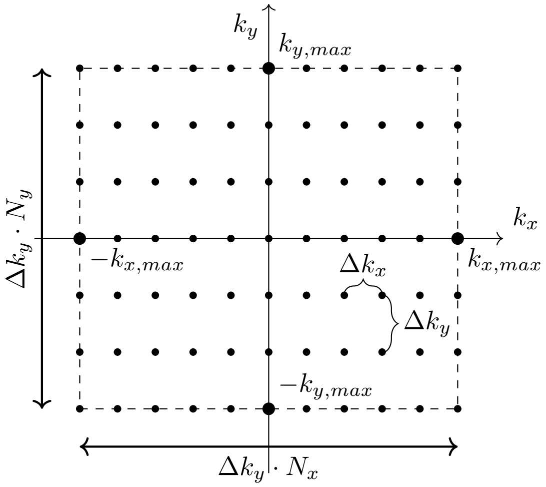


Figure 2.6: The sampled k-space grid between $[-k_{x,max}, k_{x,max}]$ and $[-k_{y,max}, k_{y,max}]$ where Δk_x and Δk_y are the sampling distance and $N_{x/y}$ is the number of sampled points along the given axis. This figure is inspired by figure 2.6 in [8] and figure 5.6 in [7].

2.2.8 Discrete sampling

MRI is not a continuous process. Time is constrained, and data sampling does not occur infinitely fast [7, pg.42]. The sampling rate and sampling time limits the resolution and FOV [7, pg. 42].

The spin density for a discretely sampled k-space distribution is given by the discrete Fourier transform

$$M'_T(0,x,y) \propto \sum_{k_x}\sum_{k_y} S'_T(k_x,k_y)e^{i(k_x\cdot x+k_y\cdot y)} \tag{2.2.21}$$

under the assumption of equidistant sampling as shown in figure 2.6. The proportionality stems from not using any conventional pre-factors.

Assuming rectangular functioned gradients as in equations (2.2.15a) and (2.2.15b), the largest k-value in k-space is given by [7, pg. 43, 8, pg. 68-71]

$$k_{x,max} = \gamma G_x \frac{t_{acq}}{2}, \tag{2.2.22}$$

where t_{acq} is the time required to traverse the distance from $-k_{x,max}$ to $k_{x,max}$ for a given gradient strength. When k-space is traversed from $-k_{x,max}$ to $k_{x,max}$, the signal is sampled with a rate of $f = \frac{1}{t_s}$, such that the distance between the sampled k-space points are given by [7, pg. 43-45, 8, pg. 44]

$$\Delta k_x = \gamma G_x t_s. \tag{2.2.23}$$

From equation (2.2.19) it can be seen that the exponential $\exp(ik_x \cdot x)$ is periodic with a wavelength of [8, pg. 68]

$$\lambda_{k_x} = \frac{2\pi}{k_x}. \tag{2.2.24}$$

This is the wavelength in which the spatial harmonics are decomposed [8, pg. 68]. From this, there are two extrema: The maximum spatial wavelength and the minimum spatial wavelength. The maximum spatial wavelength is given by combining equation (2.2.24) with the minimum spatial frequency (2.2.23), resulting in

$$\lambda_{x,max} = \frac{2\pi}{\Delta k_x} = \frac{2\pi}{\gamma G_x t_s}. \tag{2.2.25}$$

The minimum spatial wavelength is given by a combination of equation (2.2.24) with the maximum spatial frequency (2.2.22), giving

$$\lambda_{x,min} = \frac{2\pi}{2\cdot k_{x,max}} = \frac{2\pi}{\gamma G_x t_{acq}}. \tag{2.2.26}$$

The maximum spatial harmonic wavelength $\lambda_{x,max}$ is the maximum x-value or length in which an entire period of the exponential $\exp(ik_x \cdot x)$ from equation (2.2.19) elapses between two sampled data points Δk_x. The minimum spatial wavelength $\lambda_{x,min}$ is the minimum x-value or length in which an entire period of the exponential $\exp(ik_x \cdot x)$ from equation (2.2.19) elapses between $[-k_{x,max}, k_{x,max}]$. This relationship between the spatial frequency $k_{x,min/max}$ and the spatial wavelengths $\lambda_{x,max/min}$ is illustrated in figure 2.7 for a sinusoidal as a function of k_x where the x-variable is kept constant as either $\lambda_{x,min/max}$.

The importance of the spatial harmonics stems from their close relation to image space. The FOV is determined by the largest spatial harmonic (2.2.25) and is given by [7, pg. 43, 8, pg. 69]

$$\text{FOV}_x = \lambda_{x,max}. \tag{2.2.27}$$

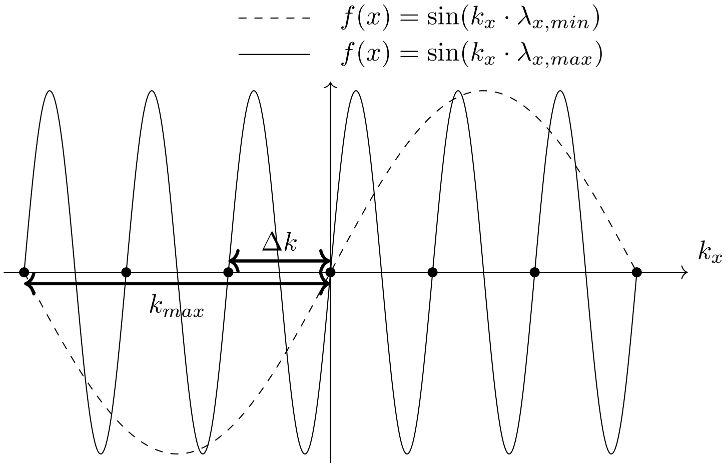


Figure 2.7: A plot of the imaginary sinusoidal component of equation (2.2.19) for two different spatial wavelengths $\lambda_{x,min}$ and $\lambda_{x,max}$ for $k_x \in [-k_{x,max}, k_{x,max}]$. The dots represent sampled k-space data points with a sampling distance of Δk.

The smallest spatial harmonic (2.2.26) determines the resolution and is given by [7, pg. 43]

$$\delta x = \lambda_{x,min}. \tag{2.2.28}$$

Under the assumption of rectangular gradients, the FOV is given by

$$\mathrm{FOV}_x = \frac{2\pi}{\gamma G_x t_s},$$

and the resolution is given by

$$\delta x = \frac{2\pi}{\gamma G_x t_{acq}}.$$

The spatial coordinates x and y are interchangeable, such that $\mathrm{FOV}_y = \lambda_{y,max}$ and $\delta_y = \lambda_{y,min}$.

2.2.9 T_1 and T_2 relaxation

After the excitation pulse the transverse magnetization M_T rapidly decays to zero while the longitudinal magnetization M_z gradually recovers [7, pg. 18]. The transverse decay is called T_2 relaxation, whereas the longitudinal recovery is called T_1 relaxation.

2.2.9.1 T_1 relaxation

If at any time $M_z \neq M_0$, namely, the magnetization is not equal to the equilibrium magnetization in the z-direction M_0, the longitudinal magnetization will approach its equilibrium value exponentially with the characteristic tissue dependent time T_1 [7, pg. 18-21, 26]. This relaxation process is called the longitudinal-relaxation or spin-lattice relaxation and can be modeled by the differential equation [7, pg. 18-21, 26]

$$\dot{M}_z = -\frac{(M_z - M_0)}{T_1}, \tag{2.2.29}$$

where M_0 is the longitudinal magnetization at equilibrium. The longitudinal relaxation is due to thermal perturbations and the fluctuating magnetic field exhibited by molecules with frequency at or near the Larmor frequency. Exciting or de-exciting neighboring spins, transforming their energy to thermal energy [8, pg. 18, 7, pg. 266-267, 26].

2.2.9.2 T_2 relaxation

The spin-spin relaxation responsible for the decay of the transverse magnetization M_T is characterized by the tissue dependent relaxation time T_2 [8, pg. 57, 7, pg. 18-19]. T_2 relaxation is caused by local microscopic field inhomogeneities introduced by various ”shielding effects” at the molecular level [7, pg. 18-19]. For a static water molecule, each proton experiences both the external magnetic field and the magnetic moment produced by other nuclei [8, pg. 266]. However, there is no static situation, and matter is in thermal agitation causing varying magnetic fields de-phasing neighboring protons [8, pg. 266].

In addition to the T_2 effects, the spins will also de-phase if there are bulk inhomogeneities within the B_0-field. These bulk inhomogeneities include local inhomogeneities δB caused by the magnet, and local inhomogeneities due to variations of susceptibility between tissue types and at interfaces between tissue and air [7, pg. 19]. These effects are combined with the T_2 relaxation time giving [7, pg. 19]

$$\frac{1}{T_2*} = \frac{1}{T_2} + \gamma \Delta B_0, \tag{2.2.30}$$

where ΔB_0 should be interpreted as the bulk inhomogeneity within a single image volume element (voxel) [7, pg. 19]. However, unlike the T_2 de-phasing caused by varied local magnetic fields by neighboring nuclei and atoms, the T_2* field inhomogeneity ΔB_0 can be considered a static effect for the duration of the MR signal generation process [7, pg. 55].

The decay of the transverse signal caused by the T_2 effects can be modeled by the differential equations [7, pg. 19]

$$\dot{M}_x = -\frac{M_x}{T_2} \tag{2.2.31a}$$

$$\dot{M}_y = -\frac{M_y}{T_2}, \tag{2.2.31b}$$

where T_2 is interchangeable with T_2*.

Since $T_2* < T_2$, the signal will deteriorate at a faster rate for T_2* relaxation. To prevent this, there exists sequence techniques where the static nature of ΔB_0 is exploited, reversing the static dephasing component and the signal then effectively decays with T_2 not T_2* [7, pg. 55].

2.2.9.3 Extended Bloch equation

The Bloch equations defined in (2.2.1) and (2.2.3) can be extended to account for the relaxation effects. In presence of relaxation the Bloch equation can be rewritten as [7, pg. 19-29]

$$\frac{d\boldsymbol{M'}}{dt} = \gamma \boldsymbol{M} \times \boldsymbol{B'_{eff}} - \boldsymbol{R}\left(\boldsymbol{M'} - \boldsymbol{M_0}\right),$$

where

$$\boldsymbol{R} = \begin{bmatrix} \frac{1}{T_2} & 0 & 0 \\ 0 & \frac{1}{T_2} & 0 \\ 0 & 0 & \frac{1}{T_1} \end{bmatrix} \quad \text{and} \quad \boldsymbol{M_0} = M_0 \boldsymbol{e_z}.$$

Under the assumption that the relaxation effects can be neglected during the RF-pulse, the longitudinal magnetization after excitation is given by [7, pg. 20]

$$M_z(t) = M_0 \left[1 - e^{-\frac{t}{T_1}}\right] + M_z(0)e^{-\frac{t}{T_1}}, \tag{2.2.32}$$

where $M_z(0)$ is the longitudinal magnetization after the RF-pulse. In the case of a perfect 90° RF-pulse the longitudinal magnetization after the excitation would be zero ($M_z(0) = 0$). Equation (2.2.32) is plotted in figure 2.8a for some arbitrary constants, and an RF-pulse pulse with a flip angle over 90° for two different T_1-values. Notice how the longitudinal magnetization is zero for different time points after the excitation pulse depending on the T_1-values.

The solution for the transverse magnetization with the relaxation effects is given by [8, pg. 58]

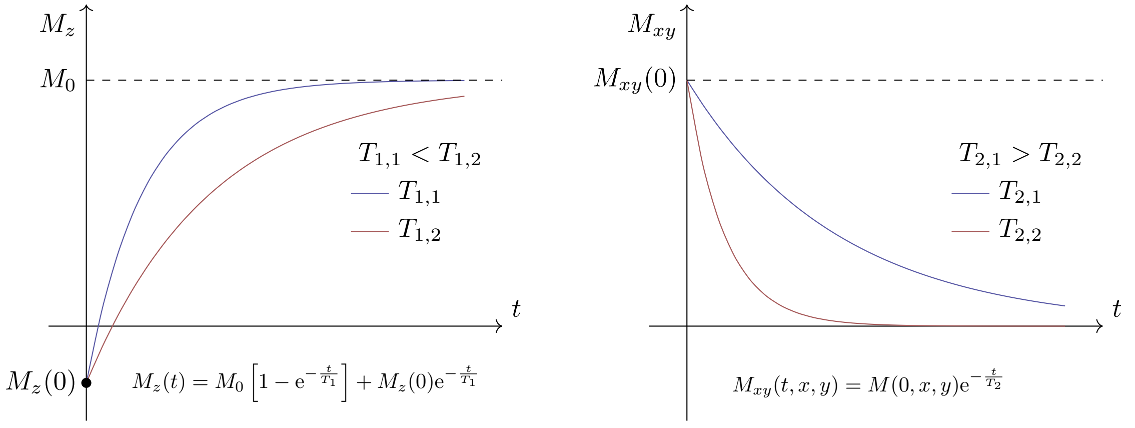
$$M_T(t, x, y) = M(0, x, y)e^{-i\gamma \int B_z(t)dt}e^{-\frac{t}{T_2}}, \tag{2.2.33}$$

where T_2 is interchangeable with T_2*. The transverse relaxation is shown in figure 2.8b for two different T_2-values. As evident from the figure, tissues with different T_2-values have different signal strengths depending on the time after excitation. Figure 2.8 shows both plots for the T_1 and T_2 relaxation.

2.2.10 Pulse sequences

In order to reconstruct the spin density from equation (2.2.20), the k-space representation of the imaged object must be properly sampled. The desired image resolution and FOV impose well defined constraints on the k-space sampling distance Δk and the number of sampled points $N_{x,y}$ as evident from equations (2.2.27) and (2.2.28). The process of applying gradients to satisfactory sample k-space within the resolution and FOV constraints is referred to as a *pulse sequence* [7, pg. 48].

A possible (albeit extremely time intensive) pulse sequence would be to sample a single k-space coordinate (k_x and k_y pair) for each RF-pulse. This is done by exciting the longitudinal magnetization into transverse plane by the RF-pulse. Thereafter, applying the gradients G_x and G_y to match some coordinate in k-space and sample said k-space coordinate. Then wait until the longitudinal magnetization has recovered and the transverse signal has decayed. This process is repeated



(a) The longitudinal relaxation from equation (2.2.32) for some arbitrary $M_z(0) < 0$ and two different relaxation values $T_{1,1}$ and $T_{1,2}$ with $T_{1,1} < T_{1,2}$.

(b) The exponential decay of the transverse magnetization from function (2.2.33) for two different relaxation values $T_{2,1}$ and $T_{2,2}$ with $T_{2,1} > T_{2,2}$.

Figure 2.8: The longitudinal T_1 relaxation and the transverse T_2 relaxation for some arbitrary constants.

$N_x \times N_y$ times with different combinations of gradients to fully sample the required amount of k-space. However, this pulse-sequence is extremely time consuming. For this reason, all modern practical pulse-sequences sample multiple k-space coordinates for each RF-pulse [7, pg. 48-49].

There exists a multitude of different pulse sequences, and the two main sequence types are the gradient echo (GRE) and spin echo (SE). In both sequences, an entire line in k-space is sampled for each RF-pulse. The GRE and SE sequences can be divided into additional subcategories depending on the order the gradients and RF-pulses are applied [7, pg. 49-53].

For simplicity and ease of argument, all magnetic gradients will be assumed to be simple time-dependent rectangular functions that can be turned off and on at will. Although, in a more practical setting the gradients are of trapezoidal nature.

2.2.10.1 Gradient echo

In a GRE sequence, an entire line of k-space is sampled for each excitation pulse. This k-space line is sampled for some given k_y-value while varying k_x starting at $-k_{x,max}$ to $k_{x,max}$. A GRE sequence is given in figure 2.9, and the corresponding k-space coordinate traversal is illustrated at the right side of the figure.

The GRE sequence starts with an excitation pulse as shown in figure 2.9, where the flip angle α_f is an adjustable parameter such that $\alpha_f \in [0°, 180°]$. During the excitation, the G_z-gradient is turned on to allow slice selection, and afterwards the G_z gradient switches polarity to ensure phase correction along the z-axis in the imaged object.

After the slice selection process, the position encoding gradients G_x and G_y are applied. The negative gradient G_x moves k_x from $k_x = 0$ to $k_x = -k_{x,max}$ along the k_x-axis, and the G_y

gradient introduces a controlled phase-shift such that $k_y \in [-k_{y,max}, k_{y,max}]$. This is illustrated in figure 2.9, where the multiple lines for the G_y-gradient correspond to a different degree of phase shift along the k_y-axis matching the k_y-lines from the k-space illustration. In figure 2.9 the initial phase shift from the G_y gradient is given by $k_y = -k_{y,max}$. With a phase shift of $k_y = -k_{y,max}$ the total phase angle from equation (2.2.16) is given by

$$\alpha_p = -k_{x,max} \cdot x - k_{y,max} \cdot y, \tag{2.2.34}$$

which corresponds to the bottom left corner of k-space in figure 2.9. After both the G_x and G_y gradients have introduced their respective phase shifts, the G_x-gradient is applied with opposite polarity. The second time the G_x gradient is applied it's for $t_{acq} \sim 0.3 - 20$ ms, during this time-interval the transverse magnetization is sampled for the different k_x-values with time-intervals of t_s for a total of N_x-samples on the traversed k-space line [27].

After a line in k-space is sampled, the entire process is repeated, starting with another excitation pulse for a different controlled phase shift. The total time between each pulse is categorized by the repetition time T_R. Note that the excitation pulse for this type of sequence needs to be applied N_y times. Hence, it is the resolution and FOV along the y-direction that is expensive in terms of imaging time, and the total imaging time is given by [7, pg. 50, 68]

$$T_{tot} = N_y \cdot T_R. \tag{2.2.35}$$

It is therefore common to have asymmetric sampled matrices (acquisition matrix) with higher acquisition along the k_x-direction [7, pg. 50]. To circumvent this, k-space is often zero-padded to achieve a quadratic matrix (reconstruction matrix), nonetheless, the resolution and FOV is still limited by the acquisition matrix [7, pg. 50].

Before continuing, some terminology is in order:

- Phase encoding gradient/direction is the k_y-axis as the G_y-gradient introduces a controlled phase shift for each sampled line in k-space.
- Frequency encoding gradient/direction is the k_x-axis as the G_x-gradient is applied during the sampling of the signal, thus changing the frequency during sampling.

By this terminology, it can be said that the phase encoding direction is undersampled, as the imaging time is proportional to the number of phase encoding steps (k_y-lines).

Notice in figure 2.9 that there are multiple dotted parts along the time-axis for the gradients and RF-pulse. This illustrates an unspecified amount of time, and by adjusting this time-period it is possible to adjust the image contrast and imaging time by changing T_R and T_E, as will be evident later.

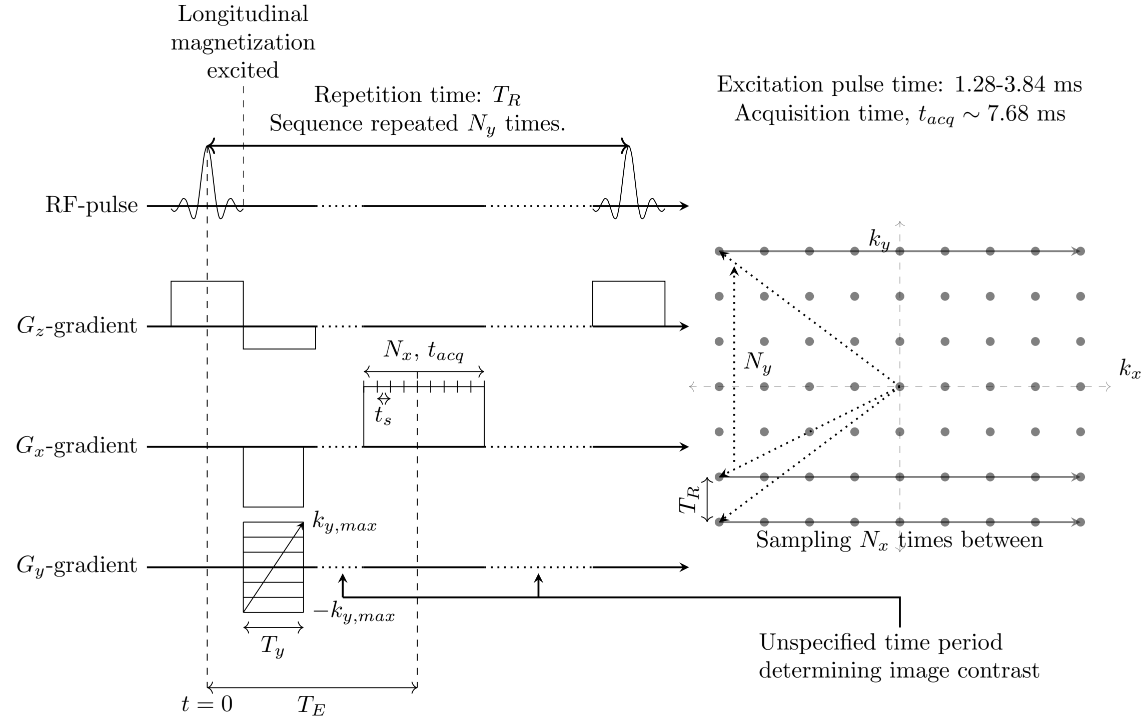

Longitudinal magnetization excited
Repetition time: T_R
Sequence repeated N_y times.
Excitation pulse time: 1.28-3.84 ms
Acquisition time, $t_{acq} \sim 7.68$ ms
RF-pulse
G_z-gradient
N_x, t_{acq}
t_s
G_x-gradient
$k_{y,max}$
G_y-gradient
$-k_{y,max}$
T_y
$t = 0$
T_E
k_y
N_y
k_x
T_R
Sampling N_x times between
Unspecified time period determining image contrast

Figure 2.9: A GRE sequence where a single line of k-space is sampled. The dotted lines in the k-space diagram on the right represent the initial phase shift from the G_x and G_y gradients, whereas the thick line along the k_x axis represents data sampling by the second G_x gradient. The left-hand side shows the sequence of the various gradients and the RF-pulse. This figure is inspired by figure 6.2 and 6.3 in [7].

2.2.10.2 Spin echo sequence

The SE sequence is similar to the GRE sequence, although slightly different. The SE sequence starts with a 90° excitation pulse in contrast to the varying flip angle in GRE sequence [7, pg. 52]. The 90° pulse is followed by the G_x and G_y gradients as in the GRE sequence, introducing their respective phase shifts. However, after these initial gradients, a second RF-pulse with a flip angle of 180° is applied. The 180° pulse switches the polarity of any prior introduced k-space phase such that $k_{x,y} \rightarrow -k_{x,y}$ $(-k_{x,y} \rightarrow k_{x,y})$ [7, pg. 52-53]. Afterwards, a line in k-space is sampled with a gradient of equal polarity as the initial gradient, because the 180° pulse switched the polarity. Any gradients following the 180° pulse does not have their polarity switched. The entire sequence is illustrated in figure 2.10; notice the dashed lines in the k-space diagram illustrating the switch in polarity.

The inclusion of the 180° pulse changes the exponential T_2* decay to T_2 decay if the center of the pulse is delivered at $T_E/2$ [7, pg. 55] by reasons not discussed here. For $T_2 \geq T_2*$, the transverse signal decays at a slower rate, ensuring greater signal during sampling. Much like the excitation pulse, only the G_z gradient can be active during the 180° pulse. As such, the initial frequency and phase encoding gradients must be applied before or after the 180° pulse before readout (signal sampling).

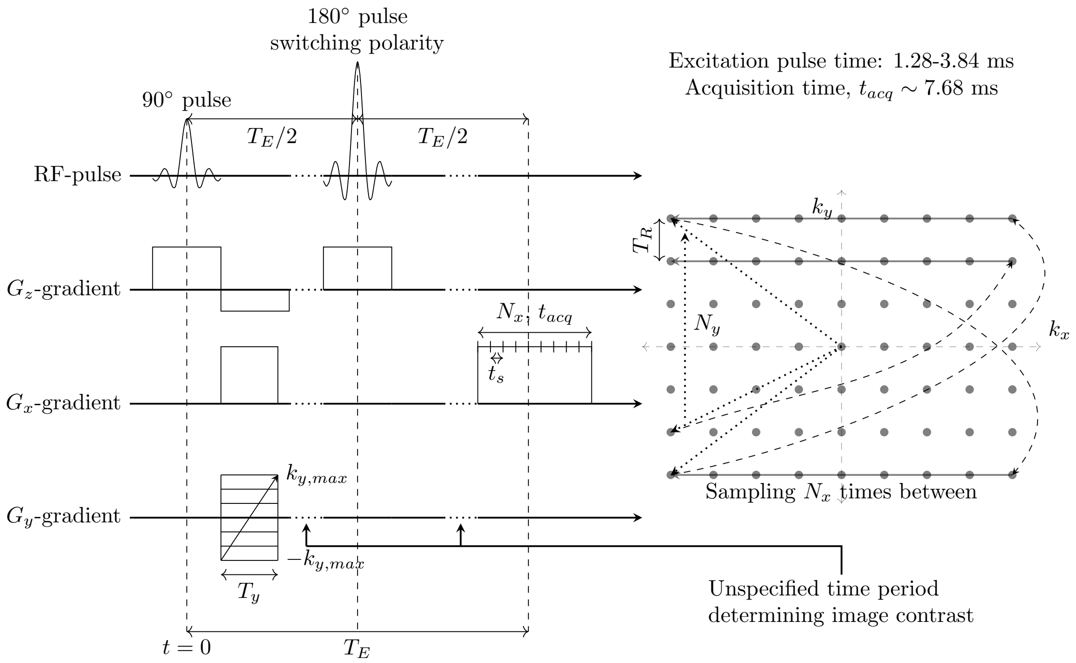


Figure 2.10: A SE sequence where a single line of k-space is sampled. The dotted lines in the k-space diagram on the right represent the initial phase shift from the G_x and G_y gradients, whereas the thick line along the k_x axis represents data sampling by the second G_x gradient. The dashed lines represent a switch in polarity caused by the 180° pulse. The left-hand side shows the sequence of the various gradients and the RF-pulse. This figure is inspired by figure 6.5 from [7] and figure 2.6 form [8].

2.2.11 3D acquisition

One approach used to image a 3D volume would be to excite different slices for different z-values, as such an entire volume could be built by stacking the different slices. However, this approach is limited by the signal-to-noise ratio (SNR) considerations, imperfections in slice profiling and the minimum achievable slice thickness [7, pg. 67]. These limitations can partially be overcome by extending the already established 2D Fourier imaging for a specific slice to a 3D Fourier volume imaging.

To achieve 3D imaging without slice selective excitation, the entire volume is excited without any magnetic gradients. Thereafter, the G_z is applied in a similar manner to the G_y gradient to create an additional phase shift in the k_z-direction [7, pg. 67]. The spin distribution from equation (2.2.19) is expanded upon by the inclusion of the k_z spatial frequency, and the 3D Fourier transform is given by [7, pg. 67]

$$M_T(0,x,y,z) = \frac{1}{(2\pi)^{3/2}} \int_{k_x} \int_{k_y} \int_{k_z} S'_T(k_x,k_y,k_z) e^{i(k_x \cdot x + k_y \cdot y + k_z \cdot z)}. \tag{2.2.36}$$

By this, the z-direction is constrained to the same FOV and resolution restrictions as the x and y directions, the FOV depends on the sampling distance Δk_z, and the resolution depends on the maximum k-space span $k_{z,max}$. In 3D imaging, the total imaging time is given by [7, pg. 68]

$$T_{acq} = T_R \cdot N_y \cdot N_z, \tag{2.2.37}$$

where N_z is the number of slices in the image volume. By today's standards, 3D-acquisition is a common modality since T_R can be made short for GRE sequences (< 5 ms), achieving reasonable imaging times around the five minute mark for volumetric imaging [7, pg. 68].

2.2.12 Image weighting

At this point, it has not been mentioned explicitly how T_E and T_R can be manipulated to achieve tissue dependent contrast by utilizing the tissue specific decay constants T_1, T_2 and T_2*. Tissue dependent weighting is used to circumvent the potential problem that the spin density alone is not a very sensitive contrast parameter in MRI, since the water contents of tissue are rather constant [7, pg. 54].

For ease of argument, assume a SE sequence with a flip angle of 90° and that the transverse magnetization is zero at T_R, i.e., before the next excitation pulse. After the initial RF-pulse with $\alpha_f = 90°$ the longitudinal magnetization recovers with T_1-relaxation, and at T_R the recovered magnetization is given by

$$M_z = M_0 \left[1 - e^{\frac{T_R}{T_1}}\right], \tag{2.2.38}$$

which is a slight alteration of equation (2.2.32). The transverse magnetization at $t = 0$, i.e., $M'_T(0,x,y)$ is proportional to the excited longitudinal magnetization and is given by

$$M'_T(0,x,y) = M(0,x,y) \left[1 - e^{\frac{T_R}{T_1}}\right]. \tag{2.2.39}$$

As the transverse magnetization decays with T_2/T_2*, the transverse magnetization is given by

$$M'_{t,T_2} = M(0,x,y) \left[1 - e^{\frac{T_R}{T_1}}\right] e^{-i(k_x \cdot x + k_y \cdot y)} e^{\frac{t}{T_2^{(*)}}}. \tag{2.2.40}$$

However, as the signal is recorded at $t = T_E$ and the sampling time is small, the transverse magnetization during sampling is approximately given by

$$M'_{T,T_2} \approx M(0,x,y) \left[1 - e^{\frac{T_R}{T_1}}\right] e^{\frac{T_E}{T_2^{(*)}}} e^{i(k_x \cdot x + k_y \cdot y)}. \tag{2.2.41}$$

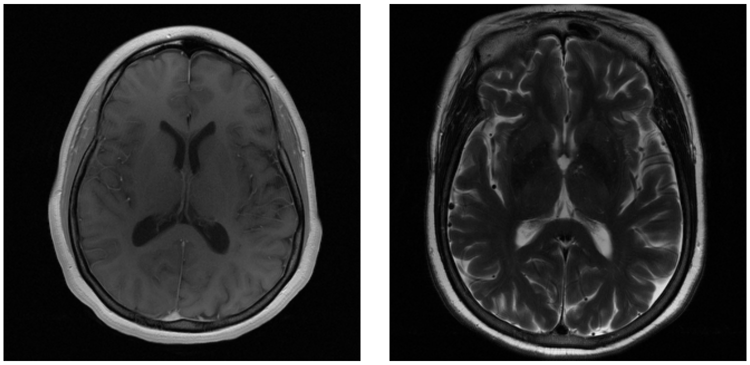
Looking back at equation (2.2.17), the T_1 and T_2 effects can be "absorbed" by the constant magnetization such that the transverse magnetization can be written as

$$M'_T = M(0,T_R,T_E,x,y) e^{-i(k_x \cdot x + k_y \cdot y)}. \tag{2.2.42}$$

Thus, as $M(0, T_R, T_E, x, y)$ is still time-independent, the Fourier transform is unchanged, and the magnetization signal is given by [7, pg. 57] (page 57)

$$SI \propto M(0, x, y) \left[1 - e^{\frac{T_R}{T_1}}\right] e^{\frac{T_E}{T_2^{(*)}}}. \tag{2.2.43}$$

From this, it can be understood that the recorded signal is the T_1 and T_2/T_2* weighted proton density, where T_R and T_E are chosen such that the exponential containing T_1 or T_2/T_2* is emphasized. Two differently weighted images are shown in figure 2.11, subfigure 2.11a emphasizes T_1 relaxation and subfigure 2.11b emphasizes T_2 relaxation.



(a) T_1-weighted image

(b) T_2-weighted image

Figure 2.11: Comparison of two different weighted images, emphasizing either T_1 or T_2 relaxation. The images are credited Knoll et al. [28].

For T_1 weighted images, T_E is chosen small to minimize the T_2/T_2* contribution with $T_R \sim T_1$, and T_1 in tissue varies between $T_1 = 300$ ms for fat to $T_1 = 3000$ for cerebrospinal fluid [7, pg. 72]. T_2 sequences are achieved by choosing $T_E \sim T_2$ with large T_R, i.e., $T_R/T_1 \approx 1$ to minimize T_1 contribution, and T_2 is typically around 100 ms [7, pg. 57]. This is the case for SE sequences, with the generalization to GRE sequences being somewhat more complicated with the addition of the flip angle variable. However, the general idea is unchanged that by adjusting T_R, T_E and α_f it is possible to emphasize either T_1 or T_2/T_2* decay in the final image.

2.2.13 Motion artifact

Motion artifact has been and still is the most common type of artifact in MRI [29]. This artifact is caused by patient motion in-between phase encoding steps creating inconsistencies between the various portions of k-space, resulting in blurring and/or ghosting along the phase encoding direction [29]. The k-space inconsistency violates the assumption of a stationary object during sampling made by the inverse Fourier transform [29]. Note that these artifacts are common along the phase encoding direction, not the frequency encoding direction, as the sampling of a frequency encoding line only lasts milliseconds [27], which is not enough time for movement inconsistencies. Conversely, the sampling of a phase encoding line takes the entirety of the scan to complete, increasing the chance of motion.

The most significant efforts to reduce motion artifacts are from the availability of faster imaging techniques, rather than more complicated motion correction schemes [29]. Shorter scan times reduce the chance of spontaneous patient movement due to patient discomfort or uncontrolled movement.

2.3 Accelerated imaging

2.3.1 Accelerated k-space trajectories

In both the GRE and SE sequences a single line of k-space is sampled for each T_R-interval. As such, the total scan time for a 2D image is proportional to the number of phase encoding lines (2.2.35). For a $T_R = 2500$ ms and $N_y = 256$ a single image would take almost 11 minutes to fully sample. However, the acquisition time can be significantly reduced by sampling multiple phase encoding lines in a single T_R-interval.

2.3.1.1 Fast Spin Echo

In a fast SE sequence, the imaging time is reduced by sampling multiple phase encoding lines for each excitation pulse. This is done by repeating the 180° excitation pulse N-times. For each 180° pulse a single line in k-space is sampled [7, pg. 89]. The total imaging time is therefore reduced to [7, pg. 89]

$$T_{tot} = \frac{N_y}{\text{ETL}} \cdot T_R, \tag{2.3.1}$$

where the echo train length (ETL) is the number of k-space lines sampled during a single T_R interval. In practice for 3D multi-slice acquisition, the time gain may not be as large as in equation (2.3.1), since the "dead-time" after sampling is used to sample other slices.

The acquisition of multiple k-space lines for a single excitation pulse causes blurring artifacts [7, pg. 91]. These blurring artifacts are caused by the T_2-decay of the signal, and increases for decreasing T_2 and increasing ETL, since the amplitude of the consecutive k_y lines are modulated by the exponential T_2-decay [7, pg. 91].

2.3.1.2 Echo Planar imaging

Similar to the fast SE, echo planar imaging (EPI) samples multiple lines in k-space during a single excitation pulse, with the main difference being the absence of the 180° pulses. EPI can be acquired in less than 100 ms on modern scanners [7, pg. 96]. However, this increased acquisition speed introduces many artifacts such as: extreme sensitivity to field inhomogeneities and significant T_2* blurring along the phase-encode-axis [7, pg. 97-100].

2.4 k-space undersampling

The MR image acquisition time is roughly proportional to the number of phase encoding steps. An overarching goal in MRI is to reduce this number without violating the Nyquist criterion and introducing aliasing artifacts in addition to lower SNR [30, 22]. These artifacts include wrap-around (foldover), blurring and Gibbs-ringing. To reduce scan time while minimizing aliasing artifacts, multiple techniques have been devised, this includes partial Fourier, parallel imaging and sparse sampling.

2.4.1 Undersampling artifacts

As evident by equations (2.2.27) and (2.2.28), the FOV and resolution along the phase encoding direction is inversely proportional to the maximum span in k-space k_{max} and the sampling distance Δk, respectively. Therefore, any decrease in k_{max} reduces the resolution along the phase encoding direction, and any increase in the sampling distance Δk decreases the FOV [6]. In general, there exists two extremes for undersampling artifacts: Decreasing k_{max} while keeping Δk constant (truncation artifact) or increasing Δk while keeping k_{max} constant (wrap-around artifact). Both of these artifacts are presented in figure 2.12 with the corresponding ground truth image, and the original k-space lines are interchanged with zeros to ensure a quadratic reconstruction matrix.

2.4.1.1 Wrap-around artifact

Wrap-around is an artifact in which an object that contributes to the MRI signal is located outside the FOV. The object not included in the FOV is then ”folded over” to the opposite side of the image [10, 11]. As such, any increase in the sampling distance Δk decreases the FOV (2.2.27) and can cause wrap-around artifacts, as MR-signal can be generated from tissue outside the current FOV. Different severities of wrap-around artifacts are shown in figures 2.12c and 2.12e, for both doubled and tripled sampling distance Δk along the phase encoding direction. Note how the resolution for the images in figures 2.12c and 2.12e are unchanged in comparison to the ground truth in figure 2.12a.

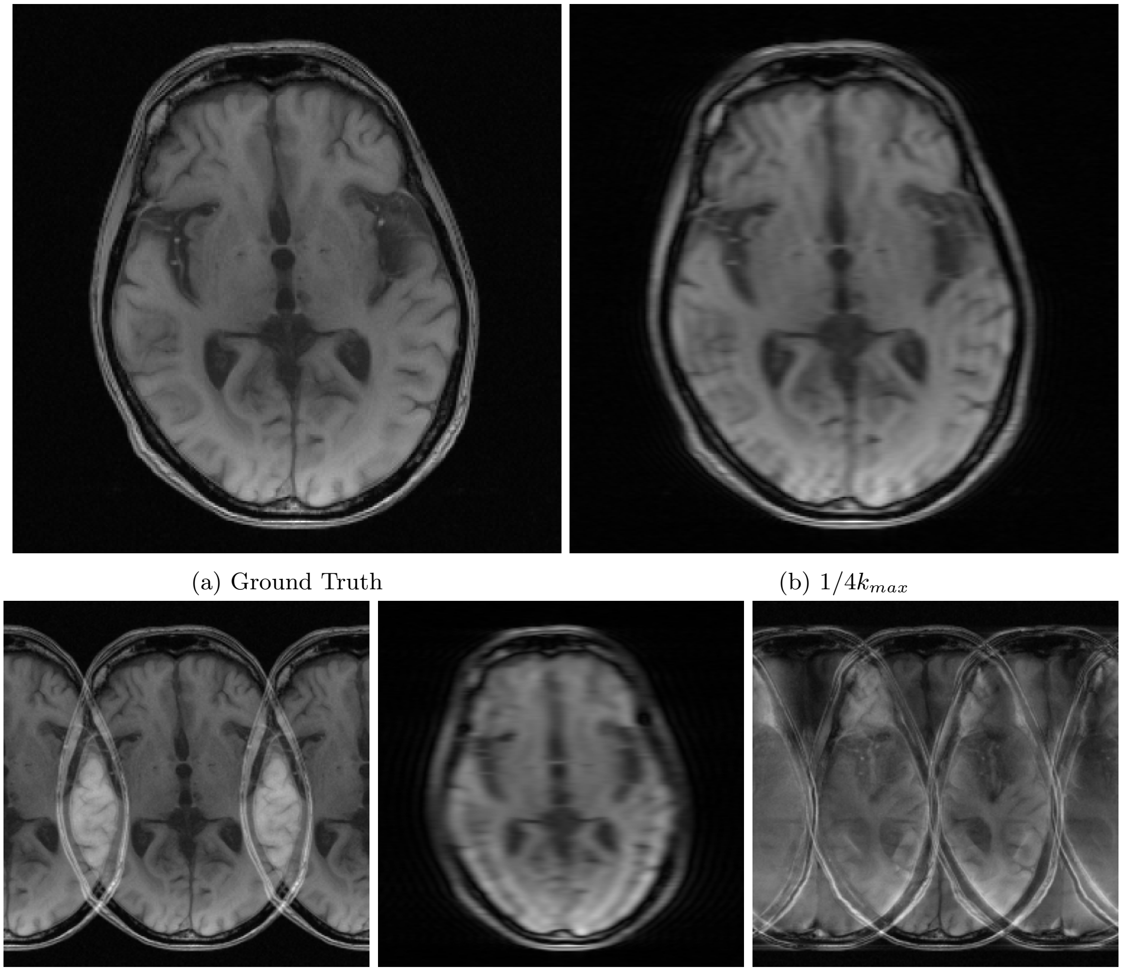
Wrap-around artifacts are caused by a violation of the Nyquist-Shannon theorem: “If a function f(t) contains no frequencies higher than W cps, it is completely determined by giving its ordinates at a series of points spaced 1/2W seconds apart” [9]. Although the theorem is often viewed in the time domain, it is a general consequence proved by use of the Fourier transform [9]. Thus, it holds in MRI, as the Fourier transform is the ”bridge” between the spatial frequencies and image space. Enforcing a limit on the maximum spatial length, such that the maximum spatial harmonic must have at max a single period passed for a sampling distance of Δk. This is illustrated with the solid-styled sine function in figure 2.7.

2.4.1.2 Truncation artifact and resolution reduction

Much like wrap-around artifacts, truncation artifacts and a reduced resolution are both consequences of undersampling. Wrap-around artifact is caused by increasing Δk. Truncation artifact and a reduced resolution are caused by decreasing k_{max}. The decrease in resolution and the truncation artifact are shown in figures 2.12b and 2.12d where k_{max} is $\frac{1}{4}$ and $\frac{1}{8}$ of the original k_{max}-value.

The truncation artifact and a reduced resolution becomes apparent for smaller k_{max}, however their causes differ. The resolution is determined by the smallest spatial harmonics (2.2.28), whereas truncation artifacts are caused by the lack of convergence of the Fourier integral at discontinuities [10]. The truncation artifacts describes the ringing effects that appear close to discontinuities (high contrast-interfaces) [10, 31]. The effects of truncation artifacts can be reduced by increasing the acquisition matrix, i.e., increasing k_{max} while maintaining the same FOV [11, 32].

To better understand the truncation error, a mathematical treatment of a truncated spatial fre-



(a) Ground Truth

(b) $1/4k_{max}$

(c) $2\Delta k$

(d) $1/8k_{max}$

(e) $3\Delta k$

Figure 2.12: Simulated k-space undersampling artifacts, this includes both the wrap-around artifact and truncation artifact for different severities. The removed k-space lines for artifact simulation are all set to zero to ensure a quadratic reconstruction matrix, and the undersampling is carried out along the k_x-axis. The k-space data is credited Knoll et al. [28].

quency spectrum in one dimension is in order. Consider a continuous truncated signal given by [31]

$$f_t(k) = f(k) \cdot g(k), \tag{2.4.1}$$

where $f(k)$ is the entire one-dimensional frequency spectrum, $g(k)$ is a rectangular function from $[-k_{max}, k_{max}]$ and $f_t(k)$ is the truncated frequency spectrum. The inverse Fourier transform of equation (2.4.1) is given by the convolution theorem [33]

$$\mathcal{F}^{-1}\left\{f_t(k)\right\} = \mathcal{F}^{-1}\left\{f(k)\right\} * \mathcal{F}^{-1}\left\{g(k)\right\},$$

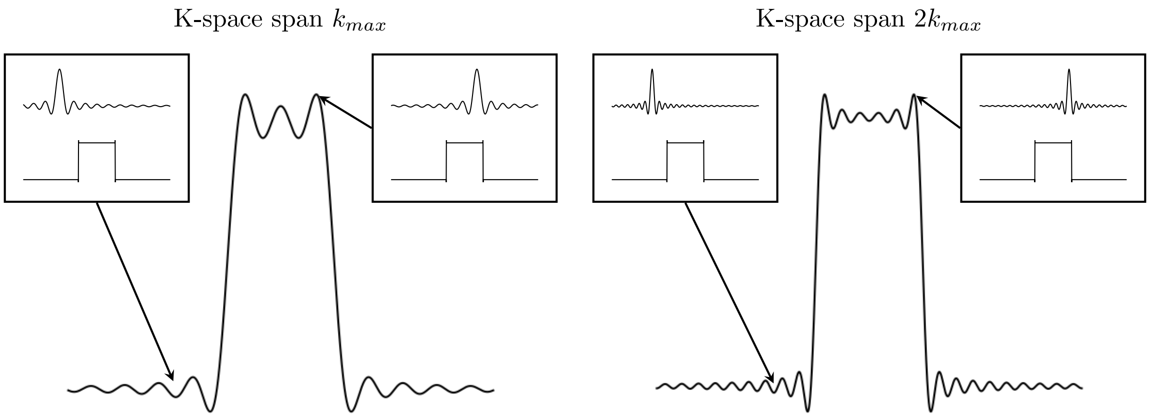


Figure 2.13: The convolution between a rectangular function and sinc function for two different k_{max}-values.

and written more explicitly

$$f_t(r) = f(r) * \mathcal{F}^{-1}\left\{g(k)\right\}, \tag{2.4.2}$$

where r is some spatial variable (could be x, y or z) and * is the convolution operator defined by $f(t)*g(t) = \int_{-\infty}^{\infty} f(\tau)g(t-\tau)d\tau$. The inverse Fourier transform of the rectangular function centered at origin from $[-k_{max}, k_{max}]$ is given by

$$\mathcal{F}^{-1}\left\{g(k)\right\} = \frac{1}{\sqrt{2\pi}}\,\mathrm{sinc}(rk_{max}).$$

As such, the truncated 1-D "image" in equation (2.4.2) is given by the proportionality

$$f_t(r) \propto \int_{FOV} f(\tau)\,\mathrm{sinc}((r-\tau)\,k_{max})d\tau, \tag{2.4.3}$$

with the convolution taken over the entire FOV of the imaged object since anything outside is zero.

Equation (2.4.3) states that each and every voxel in the FOV is a sinc weighted integral of the entire FOV centered at r. As the sinc function falls as $\frac{1}{k_{max}r}$, only signal from adjacent voxels will affect each other at any noticeable degree. From this it follows that for areas with similar adjacent tissue properties and hence small contrast differences, the truncation error is not visible. For areas near high contrast interfaces, the sinc-weighted neighbouring signals will greatly impact this part of the truncated image leading to truncation errors and the ringing artifact. The sinc weighting and the corresponding error is illustrated in figure 2.13 for a rectangular function, where it can be seen that the lobes on the sinc function determine whether the ringing represents a maximum or a minimum and how the k_{max} impacts the sinc function.

2.4.2 Partial Fourier

Partial Fourier algorithms are a series of methods that exploit Fourier conjugate symmetry under the assumption that the Fourier transformed signal $\rho(x, y)$ only has real components [12].[5] In the ideal case this assumption holds true, as the transverse magnetization is real since M_x and M_y both are real numbers. Hence, by the complex conjugate symmetry it can be shown that $S'_T(k_x, k_y) = \overline{S'_T}(-k_x, -k_y)$ [34]. As such, only about half the number of k-space lines along either the phase encoding or frequency encoding direction needs to be sampled, although some lines around origin may be sampled for a better defined ”peak” (i.e., low frequency area) [34].

A decrease in the number of phase encoding lines decreases the SNR, since the SNR depends on $\sqrt{N_y N_z}$ where N_y and N_z are the number of phase encoding lines in their respective direction [7, pg. 110-111].

In the ideal case $S'_T(k_x, k_y)$ would be real. However, motion and magnetic field inhomogeneity introduces a nonzero phase $\phi(x, y)$ to the image $\rho(x, y)$ [33]. To counteract this, multiple techniques like the Margosian and Cuppen-Pocs methods introduce some phase correctional factor $\psi(x, y)$ estimated from the symmetric low frequency area [33]. Because of this, the phase correctional methods require a bit more than 50% of the phase encoding lines sampled, with both correctional methods outperforming the standard non-corrected conjugate symmetry approach [12].

2.4.3 Parallel MRI and sensitivity encoding

In MRI, the sampled signal is recorded by the so-called receiver coils. These coils come in a variety of different shapes, and the coil design is a trade-off between tissue sensitivity and SNR [7, pg. 14]. Larger coils are sensitive to signal from a larger area but with reduced SNR, smaller coils are sensitive to a smaller area but with higher SNR [7, pg. 14]. For this reason, modern scanners combine multiple small coils in an independent channel wise manner to ensure better sensitivity, while keeping the enhanced SNR [7, pg. 15]. This gives rise to a series of separate independent images with localized sensitivities for only a small portion of the imaged object, and when combined, the entire object is imaged.

Parallel MRI is a series of techniques that utilize the inherent spatial sensitivity found in an array of multiple receiver coils [35]. Sensitivity encoding permits a reduction in the number of time consuming phase encoding steps N_y [6, 35]. This is done by increasing the distance between the sampled data points Δk without reducing k_{max}, preserving the spatial resolution (2.2.28), but decreasing the FOV (2.2.27) [6]. The general idea is illustrated in figure 2.14 for three receiver coils with box-like sensitivity profiles, each coil sampling 100% of their respective regions, and the final image is a combination of the three sub-images [35]. In this ideal case, the distance Δk could be tripled as each coil only samples signal from $\frac{1}{3}$ of the total FOV, reducing the imaging time by a factor of 3.

[5] Although $\rho(x, y)$ is assumed to be real, it is still a complex number. However it can be written on the form $a + ib$ where a and b are real.

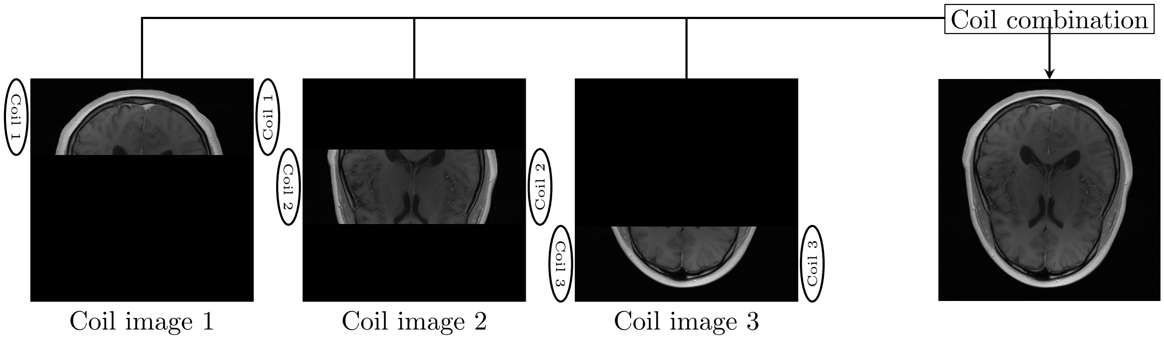


Figure 2.14: Three receiver coils with box like sensitivity profiles for $\frac{1}{3}$ of the total object and the corresponding coil combined image.

Although the example in figure 2.14 is illustrated in image-space, parallel MRI is not restricted to image space, with generalized autocalibrating partial parallel acquisition (GRAPPA) commonly being used in the clinical setting, seeking to reconstruct the missing k-space lines [36]. However, any further discussion will be limited to image-space and sensitivity encoding (SENSE) reconstruction.

SENSE reconstruction is a method that is applied in image domain and can be categorized as an "unfolding" algorithm [35]. The key idea behind SENSE reconstruction is that the wrap-around artifact occurs with different weights according to local coil sensitivity. Therefore, there exists some set of coil sensitivities C that can be used to "unfold" the "folded" image.

Consider the existence of a properly sampled image $\rho \in \mathbb{C}^{h\times w}$ and an undersampled image $\rho' \in \mathbb{C}^{h/R\times w}$ where h is the image height, w is the width and $h/R \leq h$ is the undersampled height. From ρ there exists a column vector $\hat{\boldsymbol{\rho}}^j \in \mathbb{C}^{r\times 1}$ with $\hat{\boldsymbol{\rho}}^j \subset \rho$ and $r \leq h$, where $\hat{\boldsymbol{\rho}}^j$ is a column vector consisting of the overlapping pixels from the j^{th} column from original image ρ on a folded pixel ρ'_{ij}. As such, the folded pixel is given by

$$\rho'_{ij} = \mathbf{1}\hat{\boldsymbol{\rho}}^j \tag{2.4.4}$$

where $\mathbf{1}$ is a row vector consisting of only ones. Hence, the folded pixel is a linear sum of the overlapping pixels from the original image. In the above terminology, the overlapping pixels are all from the same column (the j^{th} column); this equation assumes both equal coil sensitivity and a single coil. For a multicoil setup, the undersampled pixel is given by

$$\boldsymbol{I} = C\hat{\boldsymbol{\rho}}^j, \tag{2.4.5}$$

where $I \in \mathbb{C}^{n_c\times 1}$ is the overlapping pixel for all the n_c receiver coils, and $C \in \mathbb{R}^{n_c\times r}$ is the coil sensitivity for the different pixels that overlap for all the n_c coils [35]. This process is illustrated in the figure 2.15 for four coil elements and four overlapping pixels. From the previously established equations, it can be shown that

$$\tilde{\boldsymbol{\rho}} = \left(C^H\Psi^{-1}C\right)^{-1}C^H\Psi^{-1}\boldsymbol{I} \tag{2.4.6}$$

where $\tilde{\boldsymbol{\rho}}$ is the reconstructed estimate of $\hat{\boldsymbol{\rho}}^j$, this is the vector of overlapping pixels [37]. As such, the original pixels can be calculated, and the original image can be estimated. For this equation to be solvable, namely, to find C^{-1}, it is required that the number of coils are greater than the acceleration factor [37, 35]. Furthermore, the SNR for a SENSE reconstructed image is given by

$$\text{SNR}_{red} = \frac{\text{SNR}_{full}}{g\sqrt{R}} \tag{2.4.7}$$

where $g \geq 1$ and is called the geometry factor as it depends on coil geometry [6].

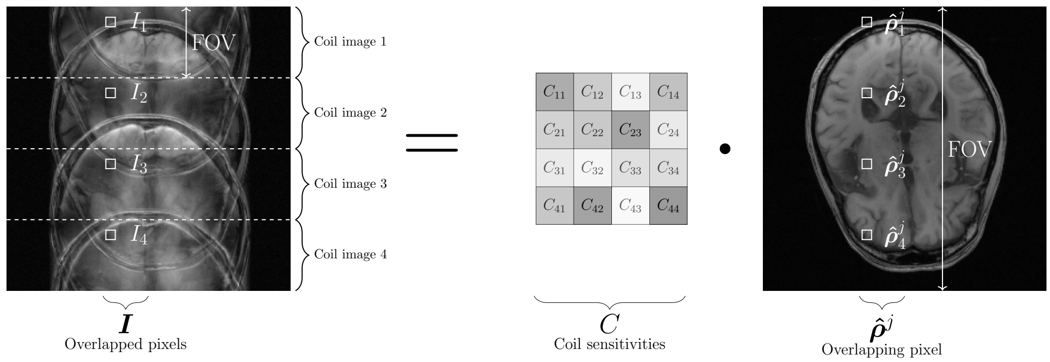

I_1
FOV
I_2
I_3
I_4
Coil image 1
Coil image 2
Coil image 3
Coil image 4
C_{11} C_{12} C_{13} C_{14}
C_{21} C_{22} C_{23} C_{24}
C_{31} C_{32} C_{33} C_{34}
C_{41} C_{42} C_{43} C_{44}
$\hat{\boldsymbol{\rho}}_1^j$
$\hat{\boldsymbol{\rho}}_2^j$
$\hat{\boldsymbol{\rho}}_3^j$
$\hat{\boldsymbol{\rho}}_4^j$
FOV
$\boldsymbol{I}$
Overlapped pixels
C
Coil sensitivities
$\hat{\boldsymbol{\rho}}^j$
Overlapping pixel

Figure 2.15: An illustration of the mathematics behind equation (2.4.5) for four coils and four overlapping pixels. This figure is inspired by figure 4 from [35] and the data is credited Knoll et al. [28].

2.4.4 Sparse sampling

Sparse sampling, also known as compressed sensing, is a reconstruction method based around the principle of exploiting the implicit sparsity in MR images to reduce the total sampled signal without any visible quality loss in the image [38]. Implicit sparsity means that the image has some sparse representation in a known fixed mathematical transform domain, and the angiogram is an example of a sparse representation in image domain, whereas other medical images may be too complex and not sparse in image domain [38].

The concept of exploiting sparsity was inspired by lossy compression of natural images with little to no loss in image quality [38, 39]. The idea behind the compression algorithms can be summarized as follows:

> Since the images we intend to acquire will be compressible, with most transform coefficients negligible or unimportant, is it really necessary to acquire all that data in the first place? Can we not simply measure the compressed in-formation directly from a small number of measurements, and still reconstruct the same image, which would arise from the fully sampled set? Furthermore, since MRI measures Fourier coefficients..., the question is whether it is possible to do the above by measuring only a subset of k-space. (Lustig, Donoho, and Pauly [38])

This is done by the minimization problem

$$\min_x \left(||S - \mathcal{F}x||_2^2 + \lambda||\Psi x||_1 \right), \tag{2.4.8}$$

where x is the reconstructed image, S is the subsampled measured k-space data, $\mathcal{F}$ is the Fourier transform, Ψ a sparsity transform and λ a regularization parameter [39].

Sparse sampling requires a randomly undersampled k-space along the phase encoding direction [38]. MR images with randomly undersampled k-space exhibit incoherent artifacts and incoherent aliasing in the transform domain can be recovered [38]. In contrast, in coherent equi-spaced aliasing (wrap-around) there exists an ambiguity, and it is therefore not possible to distinguish between the original signal and the replicas [38].

The sparse sampled reconstructed images have a preferred or equally preferred degree of image reconstruction quality in comparison to parallel imaging [40]. However, although sparse sampling shows great promise, most routine clinical MRI examinations are still based on Cartesian sampling, as the incoherent sampling criterion is challenging to fulfill [19].

2.5 Neural networks

Neural networks encompass computational models with use cases ranging from natural language processing [41] to image analysis [15] and finance [42]. Despite that the performance and utility of neural networks are impressive, the current ”cutting edge” research is highly complex, both in terms of methodology and design/architecture, with a complex set of subfields and techniques are proposed to increase the predictivity for an expanding amount of applications. Because of this, the study of neural networks stray far from the original linear framework resembling linear regression methods [43].

Rather than starting at a complex subfield of machine learning, many of the most important aspects of *deep learning* - and especially *deep neural networks* - can be discussed with ”minimal” complexity. To that end, the *the feedforward network* will be the main focus of this section, representing a general function such that $f : \mathbb{R}^n \to \mathbb{R}^m$. The network will be used to highlight the most important concepts and improvements that researchers have used years to develop and implement, creating the groundwork for the later discussion of convolutional neural network (CNN) and deep learning.

2.5.1 Feedforward neural network

In spite of their relative simplicity, feedforward neural networks are still an important cornerstone in deep learning based architectures: From ”AlexNet” achieving ground breaking result for image classification in 2012 [44] to the ”EfficientNet”-family having the current best classification scores on the famous ImageNet dataset [14] and speech recognition tasks [45].

The feedforward architecture used in those networks is called a *fully connected layer*. One such layer is a stack of multiple so-called *neurons*, where a single neuron is the weighted linear sum of n real-valued inputs acted upon by some function, the so-called *activation function* [46]. This implies that a single neuron is given by [47]

$$a_i = \sigma\left(W_i \boldsymbol{x} + b_i\right), \tag{2.5.1}$$

where $W_i \in \mathbb{R}^{1\times n}$ is n real valued weights, $\boldsymbol{x} \in \mathbb{R}^{n\times 1}$ is the n real valued inputs, b_i is some real valued scalar known as the bias and σ is a non-polynomial function called the activation function [46, 48]. Hence, each node can be seen as the weighted importance of the inputs, and a (non-polynomial) transformation σ [46]. For the general case of a stack of m-neurons in a single layer, the neuron column vector is given by [47]

$$\boldsymbol{a} = \sigma\left(W\boldsymbol{x} + \boldsymbol{b}\right), \tag{2.5.2}$$

with $\boldsymbol{a} \in \mathbb{R}^{m\times 1}$ being the next column vector, $\boldsymbol{b} \in \mathbb{R}^{m\times 1}$ is a bias column vector and $W \in \mathbb{R}^{m\times n}$ is the weight matrix for all the neurons, such that each row corresponds to the weights for a single neuron [46, 49].

The feedforward architecture is nothing more than an arbitrary amount of stacked fully connected layers, where the outputs of a single layer of neurons are treated as the inputs for the next. Mathematically this is given by

$$\hat{\boldsymbol{y}}(\boldsymbol{x}) = \sigma^L\left(W^L\sigma^{L-1}\left(W^{L-1}...\sigma^2\left(W^2\sigma^1\left(W^1\boldsymbol{x} + \boldsymbol{b^1}\right) + b^2\right)\boldsymbol{b^{L-1}}\right) + \boldsymbol{b^L}\right). \tag{2.5.3}$$

As evident by equation (2.5.3), the feedforward architecture can be viewed as an ensemble of neuron column vectors. Each column vector is the result of a matrix multiplication between the corresponding weights and the prior column vector [50]

$$\boldsymbol{a}^l = \sigma^l\left(W^l\boldsymbol{a}^{l-1} + \boldsymbol{b^l}\right), \tag{2.5.4}$$

starting from the input vector $\boldsymbol{x}$. This architecture is shown in figure 2.16 with a total of 4 layers. In the figure, the intermediate column vector or output for the i^{th} layer is denoted by $\boldsymbol{a}^i$, meaning that the first set of intermediate neurons are denoted by $\boldsymbol{a}^1$,[6] and the intermediate layers in-between the input and output columns are known as *hidden layers*. The number of layers is usually denoted as the depth of a neural network, and the number of neurons in a single layer denotes the width [51]. Hence, a deep network has many layers, whereas a wide network has many neurons per layer. By this, a wide network has few, but large, weight matrices, and deep networks have many but smaller weight matrices.

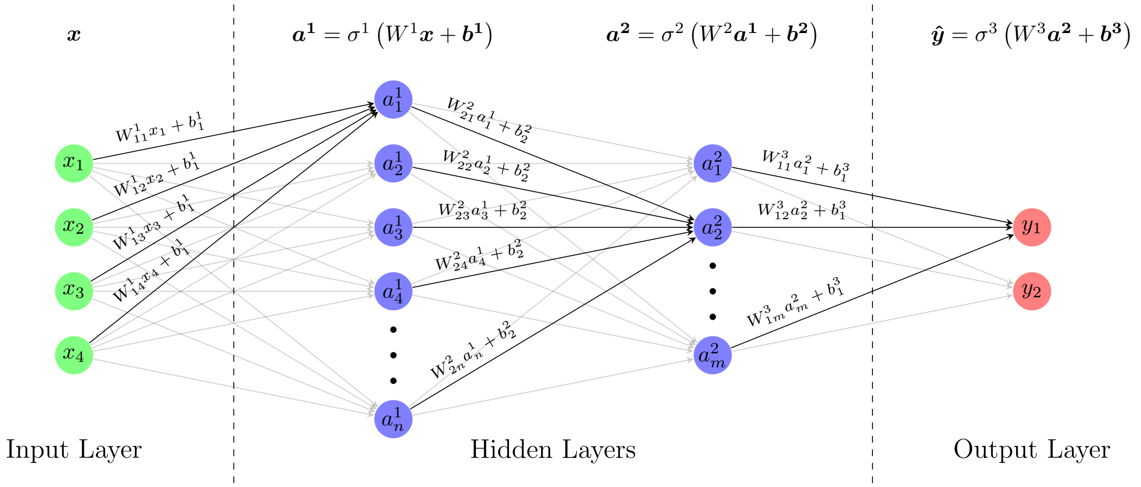


Figure 2.16: A simple feedforward architecture with two hidden layers with n and m nodes, respectively.

[6] The input $\boldsymbol{x}$ is the zero'eth layer, i.e., $\boldsymbol{a}^0 = \boldsymbol{x}$

2.5.2 Universal approximation theorem

The hidden layers shown in figure 2.16 are not enforced, however, they greatly increase the expressivity of neural networks [46], with George Cybenko proving in 1989 that a feedforward neural network can approximate any real-valued function [52]. More precisely, a feedforward neural network with a single hidden layer using the so-called sigmoid activation function can approximate any real valued function to a desired accuracy [52]. Furthermore, a feedforward neural network can approximate any continuous multi-input/multi-output function with arbitrary accuracy, as long as the activation function in the hidden layer is continuous and non-polynomial [46, 48].

The universal approximation theorem also holds for networks scaled in depth, as feedforward neural networks with constant width, but arbitrary depth are also universal approximators [53]. This further extends to both recurrent neural networks [54] and CNNs [55].

For infinite depth or width, the error ϵ between the true function and the approximation limits zero [55, 48]

$$\text{width/depth} \to \infty \quad \epsilon \to 0.$$

However, this says nothing about the rate of approximation, only its existence [48]. Nonetheless, it can be shown that a fully connected neural network with two hidden layers can unlike a single hidden layered network approximate a continuous function arbitrary well with a finite amount of neurons [48].[7]

2.5.3 Parameter optimization - training

In subsection 2.5.1 it was shown that for a simple feedforward network, the input $\boldsymbol{x}$ propagates through the network by a series of matrix multiplications and activation functions. This propagation from the initial input through the network is called *forward propagation* [51]. The output $\hat{\boldsymbol{y}}$ can then be compared to the *ground truth* $\boldsymbol{y}$ by use of the so-called *loss function* or *cost function*. The loss function is a means on which to measure the degree of error between the real values $\boldsymbol{y}$ and the output $\hat{\boldsymbol{y}}$. Examples of loss functions are the *l1* and *l2* functions, also known as the mean absolute error (MAE) and mean squared error (MSE) given by

$$C_{l1}(\hat{\boldsymbol{y}}, \boldsymbol{y}) = \text{mean}\left[|\hat{\boldsymbol{y}} - \boldsymbol{y}|\right] \tag{2.5.5a}$$

$$C_{l2}(\hat{\boldsymbol{y}}, \boldsymbol{y}) = \text{mean}\left[(\hat{\boldsymbol{y}} - \boldsymbol{y})^2\right], \tag{2.5.5b}$$

following the definitions in the open source machine learning framework known as PyTorch [56]. By the universal approximation theorem, there exists some configuration of weights W and biases b such that this error can become arbitrarily small. However, the universal approximation theorem only guarantees its existence, not the method in which the parameters can be optimized.

Optimization of the network parameters is often labeled as training. This training is the conjunct efforts between the so-called *backpropagation* algorithm followed by gradient descent methods [51].

For simplicity, and to ensure an intuitive understanding of the backpropagation algorithm, it will only be explained for the case of a feedforward neural network. Nonetheless, it can be generalized for other artificial neural networks such as recurrent neural network and CNN [43].

[7] More precisely, it can be shown that for a two hidden layered network, there exists no theoretical lower bound, unlike that of a single hidden layer [48].

2.5.3.1 Backpropagation

The backpropagation is an algorithm that computes the gradient updates for the gradient descent algorithms based on the loss function [46]

$$\frac{\partial C}{\partial W_{j,k}^{l}}, \quad \frac{\partial C}{\partial b_j^l}. \tag{2.5.6}$$

By this, the weight parameter $W_{j,k}^l$ and bias b_j^l can be updated by gradient descent methods [46, 47]

$$W_{j,k}^l = W_{j,k}^l - \eta \frac{\partial C}{\partial W_{j,k}^l}, \tag{2.5.7a}$$

$$b_j^l = b_j^l - \eta \frac{\partial C}{\partial b_j^l}, \tag{2.5.7b}$$

where η is known as the learning rate and is a configurable hyperparameter. For clarification, the weight $W_{j,k}^l$ is the weight for the l^{th} layer for the i^{th} neuron from the j^{th} input node (see figure 2.16). The overall goal of neutral networks can therefore be understood as a method to optimize the weights and biases to minimize the loss function.

As one might expect by the name, backpropagation is rather similar to forward propagation, only with adjusted expressions. The "input" for backpropagation is the derivative of the loss function with respect to the output of the neural network

$$\nabla_{\hat{\boldsymbol{y}}} C(\hat{\boldsymbol{y}}, \boldsymbol{y}) = \left[\frac{\partial C}{\partial \hat{y}_1}(a_1^L, y_1), ..., \frac{\partial C}{\partial \hat{y}_N}(a_N^L, y_N)\right].$$

For the MSE this corresponds to

$$\nabla_{\hat{\boldsymbol{y}}} C(\boldsymbol{y}, \hat{\boldsymbol{y}}) = 2\left(\hat{\boldsymbol{y}} - \boldsymbol{y}\right).$$

However, from this point on, any such derivatives will not be stated, as many deep learning libraries such as PyTorch handle automatic differentiation [56].

The first "layer" (or former) during backpropagation is given by [50]

$$\delta^L = \nabla_{\hat{\boldsymbol{y}}} C(\boldsymbol{y}, \hat{\boldsymbol{y}}) \odot \sigma^{L\prime}(\boldsymbol{z}^L), \tag{2.5.8}$$

where $\odot$ is the Hadamard product (element-wise multiplication) and

$$\boldsymbol{z^l} = W^l \boldsymbol{a^{l-1}} + b^l$$

is the input of the activation function from equation (2.5.4). The state from equation (2.5.8) propagates backwards by [50]

$$\delta^l = \left(\left(W^{l+1}\right)^T \delta^{l+1}\right) \odot \sigma^{l\prime}(\boldsymbol{z}^l). \tag{2.5.9}$$

The importance of δ^l stems from its close relation to the rate of change in equation (2.5.6), as the rate of change for the weights are given by [50]

$$\frac{\partial C}{\partial W_{j,k}^l} = a_k^{l-1} \delta_j^l \tag{2.5.10}$$

and for the biases [50]

$$\frac{\partial C}{\partial b_j^l} = \delta_j^l. \tag{2.5.11}$$

From this, it can be understood that the gradient updates propagate from the initial δ^L and backwards by equation (2.5.9). Each gradient update (2.5.6) for the corresponding weights and biases in a specific layer is given by the δ^l for that layer as seen by equations (2.5.10) and (2.5.11).

2.6 Deep convolutional networks

Deep learning is a methodology adapted in neural networks in which multi-layered computational models learn intricate structures using the backpropagation algorithm [57]. In the previously established framework of feedforward neural networks (2.5), deep learning is the methodology where the networks are scaled in depth by increasing the number of layers, instead of scaling in width. However, deep learning is not limited to the simple feedforward architecture, rather, one of the most widely used types deep networks is the so-called deep convolutional network [58]. Deep convolutional networks apply convolutional layers, and these layers are commonly seen in image analysis. The convolutional layer introduces shared weights, sparse interactions and equivariance into the network [59, 58].

Deep convolutional networks are by today's standards state-of-the-art in many types of image related analysis: image classification [14], image segmentation and object detection [15], sequence modeling [60] and AlphaZero [61]. However, this has not always been the case. In the early stages of neural networks, deep neural networks showed no empirical advantage over their shallower counterparts. They were difficult and hard to train without specific training regimes [49, 43], and this problem was dubbed the *vanishing gradient problem* [49, 43].

Even with the vanishing gradient problem, deep neural networks have prospered the last few years, with one of the main breakthroughs happening not further back than 2012 with the publication of "AlexNet", achieving groundbreaking results for image classification [44]. Since then, deep CNN has become the norm for image analysis, stacking hundreds of convolutional layers in the top performing deep CNNs, from AlexNets humble seven layers. Newer networks utilize simple but effective methods for combating the vanishing gradient problem.

2.6.1 Convolutional layer

The main building block of a CNN is the convolutional layer. The input images for a convolutional layer are often in the form of multidimensional arrays, known as tensors [59]. Much like the previously introduced fully connected layer (2.5.1), the input is acted upon by some set of weights W and added with biases $\boldsymbol{b}$. More concrete, for a conventional layer, the input image a^l[8] is convolved with a set of K kernels $W \in \{W_1, W_2, ..., W_K\}$ with biases $\boldsymbol{b} \in \{b_1, b_2, ..., b_K\}$ added [49]

$$z_k^l = W_k^l * a^{l-1} + b_k^l,$$

with z_k^l element-wise acted upon by the non-linear activation function [49]

$$a_k^l = \sigma^l \left(W_k^l * a^{l-1} + b_k^l \right). \tag{2.6.1}$$

[8] Using the same notation as previously used for a feedforward network.

Although equation (2.6.1) illustrates the fact that a convolutional layer is some input a_k^{l-1} acted upon by the weights W_k^l, it states nothing about the shape of the tensors, and how the convolution is carried out.

Consider an input tensor $a^{l-1} \in \mathbb{R}^{N \times C_{in} \times h \times w}$ consisting of N images, with C_{in} channels (RGB images have three channels), height h and width w. From this there exists a complementary set of weight tensors with shape $W^l \in \mathbb{R}^{C_{out} \times C_{in} \times k_y \times k_x}$ where C_{out} is the number of outgoing channels after the convolution, k_y is the kernel height and k_x is the kernel width. The output for the c^{th}-channel in n^{th}-image is given by

$$a_{n,c}^l = b_c^l + \sum_{k=0}^{C_{in}-1} W_{c,k}^l \star a_{n,k}^{l-1} \tag{2.6.2}$$

where $\star$ is the 2D cross-correlation operator and $\boldsymbol{b^l} \in \mathbb{R}^{C_{out}}$ is the bias [56]. The cross-correlation operation from equation (2.6.2) is illustrated in figure 2.17 for a single image with a one-channel input and a two-channel output, using a 3×3 kernel. The output pixels are the sum of the element-wise multiplication between the kernel and the input image, with the kernel following a sliding window approach. For each new output pixel, the kernel has moved either to the right or down, and the output height and width are both reduced by two, since the kernel cannot slide outside image bounds. The number of kernels used for a convolutional operation is directly proportional to the number of in-channels and out-channels, as such, width scaling for a CNN is done by increasing the number of channels, by this increasing the number of kernels and the number of parameters.

The size of the output - the feature map - along a given axis is given by [56]

$$h' = \frac{h - k_y - 1}{\text{stride}} + 1, \tag{2.6.3}$$

where *stride* is the number of pixels the kernel slides for each output pixel along said axis. A common operation is to use a 2×2 kernel with stride 2 to half the resolution, another common approach is to zero-pad the input for kernel sizes larger than 1, such that the resolution is unchanged. For a 3×3 convolution, the input image would be zero-padded along both the height and width, such that the height and width are given by $h = h + 2$ and $w = w + 2$. By equation (2.6.3), the output height and width would be equal to the original non-padded input shape.

From figure 2.17 and equation (2.6.2), it can be seen that the weights from the convolutional kernel $W_{c,k}^l$ is shared across the entire input, although the kernels for the different input channels vary. This differs from the fully connected layer, where a single weight is only used once as depicted in figure 2.16. This change does not affect the runtime of the forward propagation, only the necessary number of parameters [59]. Kernels smaller than the input image enables so-called sparse interactions [59] that allows the detection of small but meaningful features such as edges, reducing the number of required parameters, and decreasing the computational cost with smaller kernels [59]. Furthermore, this removes the need for separate detectors for objects appearing in multiple places in the same image, making the network equivariant in respect to translations [49].

2.6.2 Vanishing gradient

As evident from equations (2.5.8) and (2.5.9), it can be understood that the backpropagation algorithm can be written in same form as the forward propagation algorithm in equation (2.5.3).

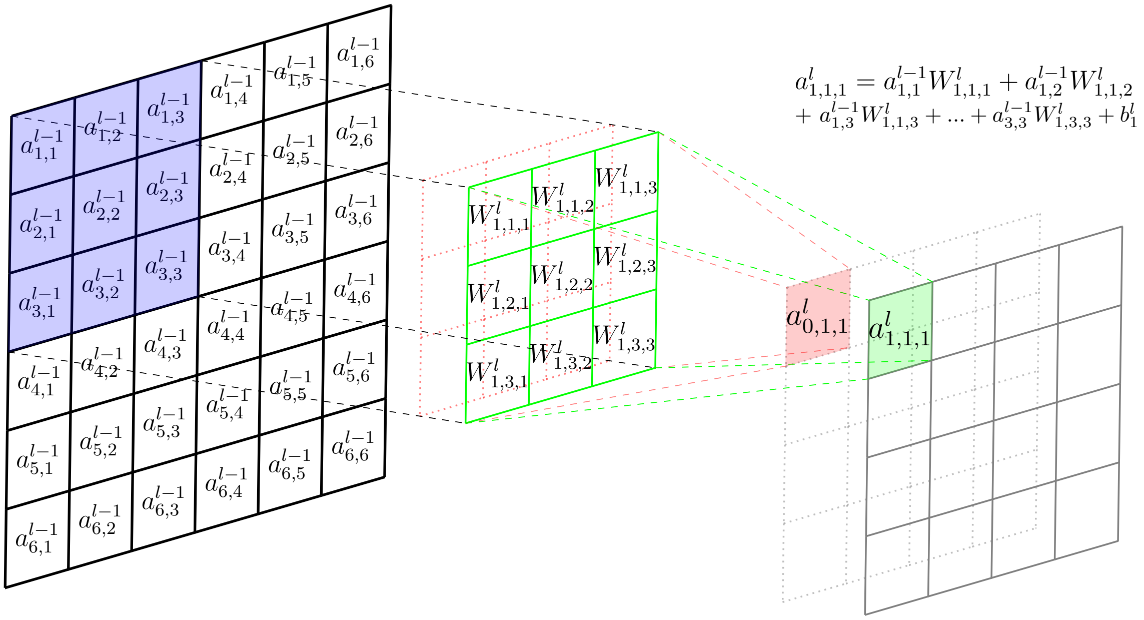

$a^{l-1}_{1,1}$ $a^{l-1}_{1,2}$ $a^{l-1}_{1,3}$ $a^{l-1}_{1,4}$ $a^{l-1}_{1,5}$ $a^{l-1}_{1,6}$
$a^{l-1}_{2,1}$ $a^{l-1}_{2,2}$ $a^{l-1}_{2,3}$ $a^{l-1}_{2,4}$ $a^{l-1}_{2,5}$ $a^{l-1}_{2,6}$
$a^{l-1}_{3,1}$ $a^{l-1}_{3,2}$ $a^{l-1}_{3,3}$ $a^{l-1}_{3,4}$ $a^{l-1}_{3,5}$ $a^{l-1}_{3,6}$
$a^{l-1}_{4,1}$ $a^{l-1}_{4,2}$ $a^{l-1}_{4,3}$ $a^{l-1}_{4,4}$ $a^{l-1}_{4,5}$ $a^{l-1}_{4,6}$
$a^{l-1}_{5,1}$ $a^{l-1}_{5,2}$ $a^{l-1}_{5,3}$ $a^{l-1}_{5,4}$ $a^{l-1}_{5,5}$ $a^{l-1}_{5,6}$
$a^{l-1}_{6,1}$ $a^{l-1}_{6,2}$ $a^{l-1}_{6,3}$ $a^{l-1}_{6,4}$ $a^{l-1}_{6,5}$ $a^{l-1}_{6,6}$
$W^l_{1,1,1}$ $W^l_{1,1,2}$ $W^l_{1,1,3}$
$W^l_{1,2,1}$ $W^l_{1,2,2}$ $W^l_{1,2,3}$
$W^l_{1,3,1}$ $W^l_{1,3,2}$ $W^l_{1,3,3}$
$a^l_{0,1,1}$
$a^l_{1,1,1}$
$a^l_{1,1,1} = a^{l-1}_{1,1}W^l_{1,1,1} + a^{l-1}_{1,2}W^l_{1,1,2}$
$+ a^{l-1}_{1,3}W^l_{1,1,3} + ... + a^{l-1}_{3,3}W^l_{1,3,3} + b^l_1$

Figure 2.17: A visual representation of a convolutional layer with a one input channel and a two output channels with a 3 × 3 kernel.

For the simplest case of a single neuron in all L layers including the output, the gradient update for the first bias is given by [62]

$$\frac{\partial C}{\partial b^1} = \sigma^{1'}(z^1) \times W^2 \times \sigma^{2'}(z^2) \times ... \times W^{L-1} \times \sigma^{L-1'}(z^{L-1}) \times W^L \times \sigma^{L'}(z^L) \times \frac{\partial C}{\partial a^L}. \tag{2.6.4}$$

From equation (2.6.4), it can be seen that the derivative of the activation function $\sigma^{l'}$ is repeated for each backpropagated layer. Hence, for the bias in the first layer, the derivative of the activation function is repeated L-times. For the historically popular sigmoid activation function given by

$$\text{sigmoid}(x) = \frac{1}{1 - e^{-x}}$$

the derivative is given by

$$\frac{d}{dx}\text{sigmoid}(x) = \text{sigmoid}(x)(1 - \text{sigmoid}(x)),$$

where the maximum value for the derivative is $\frac{1}{4}$ [62]. Since the derivative is less than 1, the numerical value and thus gradient update may decrease for each subsequent layer, reducing the overall numerical value for the parameter. Of course, this no rigorous proof, but rather an intuitive illustration that the numerical value and thus gradient updates decrease as the depth of the neural network increases. The vanishing gradient problem encompasses the idea that the gradient updates decrease for deeper neural networks.

The transition to the rectified linear unit (ReLU) activation function

$$\mathrm{ReLU}(x) = max(x, 0) \tag{2.6.5}$$

showed significant improved performance and training time over the sigmoid activation function [43, 57]. Note that the derivative of the ReLU function is either 0 or 1. Although the requirement of a non-polynomial activation function is satisfied for the sigmoid function (2.5.2), this does not guarantee the optimal activation function. Hence, a popular field of research within the deep learning community is to discover more optimized activation functions. However, even with this, the ReLU function is still highly viable, and a common benchmark for the more recent activation functions.

2.6.3 Normalization layer

Consider the c^{th} channel for the n^{th} input or image in an intermediate state tensor in a convolution neural network given by $a^l_{n,c}$. A common approach in deep learning is to standardize an intermediate state by [56]

$$a^l_{n,c} = \frac{a^l_{n,c} - \mu_{a^l_{n,c}}}{\sqrt{\mathrm{Var}\left(a^l_{n,c}\right) + \epsilon}} \cdot \gamma + \beta, \tag{2.6.6}$$

where $\mu_{a^l_{n,c}}$ is the mean of $a^l_{n,c}$, ϵ is added for numerical stability, γ and β are learnable parameters. This is the so-called instance normalization [63] shown to improve the performance and results for image generation. In this, every channel is independently standardized with a mean of zero and standard deviation of one, and the learnable parameters are often enforced to zero. Besides instance normalization, another common approach for intermediate state normalization is the so-called batch normalization, where the entire tensor a^l is standardized, and batch normalization has shown to improve convergence [64]. Since the normalization layers have shown improved performance they are standard in the current state-of-the-art deep convolution networks, and a convolutional layer is often paired together with a normalization layer in a deep learning model.

2.7 Metrics

2.7.1 Structural similarity index measure

The structural similarity index measure (SSIM) is an image quality assessment metric proposed in 2004 by Wang et al. [65]. This metric deviates from the common point-wise Minkowski error metrics to a more structure-based metric. The SSIM is based around the assumption that the human visual system is highly adept at extracting structural information, mimicking the image quality as perceived by the human eye [65]. A simple motivation for this is evident in figure 2.18 where the same image is plotted for almost identical normalized mean squared error (NMSE), but different SSIMs. In this figure, the same image has undergone multiple transformations such as Gaussian blurring, k-space undersampling, mean shift, a simple contrast enhancement and random Gaussian noise. For the SSIM metric, higher values indicate better image quality, in contrast to the NMSE where lower values indicate better image quality.

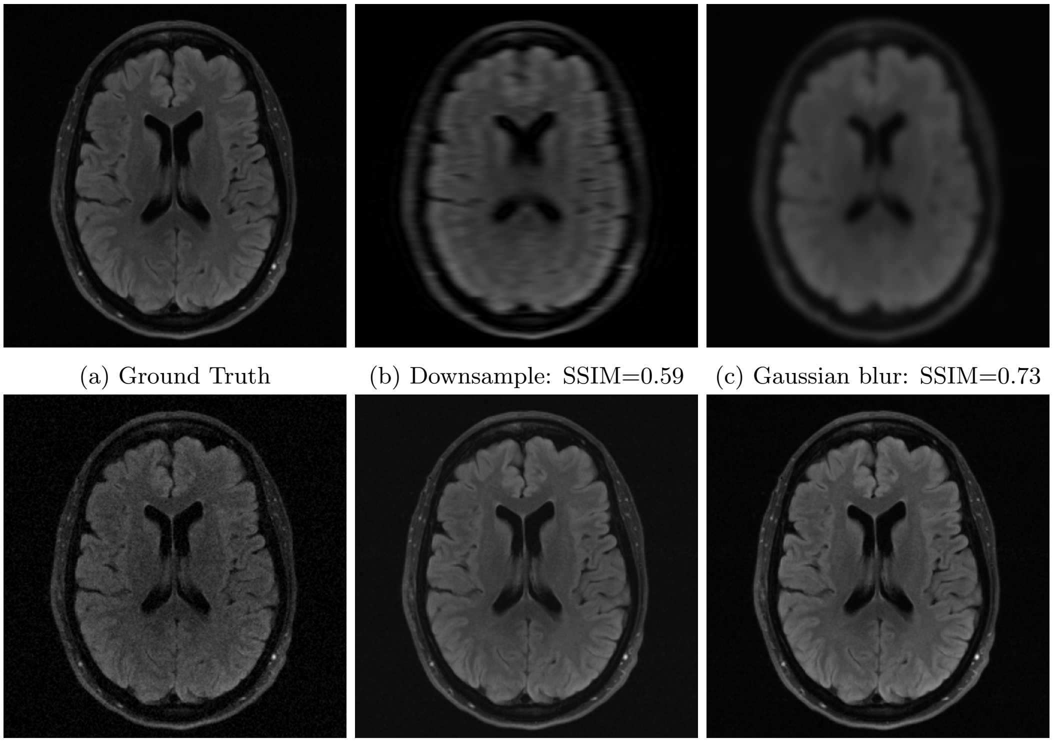

(a) Ground Truth
(b) Downsample: SSIM=0.59
(c) Gaussian blur: SSIM=0.73
(d) Random Noise: SSIM=0.59
(e) Mean shift: SSIM=0.91
(f) Contrast: SSIM=0.98

Figure 2.18: The SSIM for a single brain image undergone multiple transformations with almost identical NMSE (within a couple percent), and higher SSIM suggests better perceived image quality. The original k-space data credited Knoll et al. [28].

To achieve a structural based error metric, the SSIM incorporates three different components: luminance, contrast and structure

$$\mathrm{SSIM}(\boldsymbol{x},\boldsymbol{y}) = l(\boldsymbol{x},\boldsymbol{y})^{\alpha} c(\boldsymbol{x},\boldsymbol{y})^{\beta} s(\boldsymbol{x},\boldsymbol{y})^{\gamma}, \tag{2.7.1}$$

where $\alpha > 0$, $\beta > 0$ and $\gamma > 0$ [65]. The luminance between x and y are compared based on their respective means, such that the luminance is given by

$$l(\boldsymbol{x},\boldsymbol{y}) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1}, \tag{2.7.2}$$

where $\mu_{x/y}$ is the mean image intensity and C_1 is a constant included to avoid instability [65]. The C_1 constant is further defined by

$$C_1 = (k_1 L)^2, \tag{2.7.3}$$

where L is the *dynamic range* of the pixel values (e.g., the maximum image intensity difference), and k_1 a chosen parameter. The contrast measurement is modeled by the comparison of the standard deviation of the respective input signals

$$c(\boldsymbol{x}, \boldsymbol{y}) = \frac{2\sigma_x \sigma_y + C_2}{\sigma^2 + \sigma_y^2 + C_2}, \tag{2.7.4}$$

where C_2 is a added constant to avoid instability given by $C_1 = (k_2 L)^2$, $\sigma_{x/y}$ is the standard deviation and k_2 a chosen parameter [65]. Structural information is compared using the cross-correlation term [66], such that the structural component of the SSIM is given by [65]

$$s(\boldsymbol{x}, \boldsymbol{y}) = \frac{\sigma_{xy} + C_3}{\sigma_x \sigma_y + C_3}, \tag{2.7.5}$$

with C_3 being an added constant to avoid instability. Commonly, the terms $\alpha = \beta = \gamma = 1$, which simplifies the SSIM to

$$\text{SSIM}(\boldsymbol{x}, \boldsymbol{y}) = \frac{(2\mu_x \mu_y + C_1)(2\sigma_{xy} + C_2)}{\left(\mu_x^2 + \mu_y^2 + C_1\right)\left(\sigma_x^2 + \sigma_y^2 + C_2\right)}, \tag{2.7.6}$$

with $C_3 = C_2/2$ and the constants are commonly set to $k_1 = 0.01$ and $k_2 = 0.03$.

For image quality assessment, the SSIM should be applied locally rather than globally, following a sliding window approach, as image statistical features are highly non-stationary, and local image distortions may depend on local image statistics [65]. The SSIMs shown in figure 2.18 are the mean of all local SSIMs, using a sliding window with size 7 × 7. The sliding window approach is illustrated in figure 2.19, where the SSIMs are calculated for each cyan square. However, unlike the figure, it is more common to have overlapping and smaller windows than the illustration.

2.7.2 Normalized mean squared error

The NMSE is defined as

$$\text{NMSE}(\hat{y}, y) = \frac{||\hat{y} - y||_2^2}{||y||_2^2}, \tag{2.7.7}$$

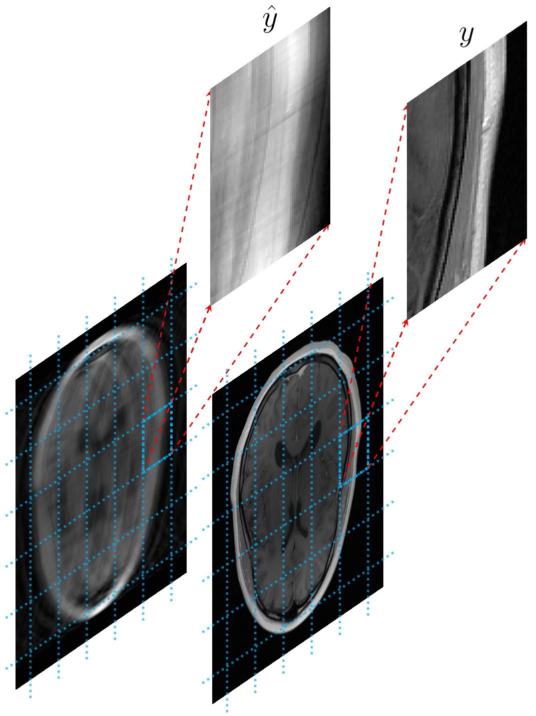
where $||x||_2^2$ is the squared Euclidean norm, $\hat{y}$ is the reconstructed image, y the ground truth and $(\hat{y} - y)$ applying element-wise subtraction [22]. The NMSE has a tendency to favor smoothness rather than sharpness.

2.7.3 Peak signal-to-noise ratio

The peak signal-to-noise ratio (PSNR) is given by

$$\text{PSNR}(\hat{y}, y) = 10 \log_{10} \left(\frac{\max(y)}{\text{MSE}(\hat{y}, y)} \right), \tag{2.7.8}$$

where y being the ground truth, $\hat{y}$ the reconstructed input, $\max(y)$ the maximum value in the ground truth and $\text{MSE}(\hat{y}, y)$ is the mean squared error $\frac{1}{N}\sum_i^N (\hat{y}_i - y_i)^2$ with N being the number of entries [22]. The PSNR does therefore represent the ratio between the maximum and noise/errors [22], with higher values indicating better reconstruction.



$$\mu_y = mean(y)$$

$$\mu_{\hat{y}} = mean(\hat{y})$$

$$\sigma_y^2 = mean(y^2) - \mu_y^2$$

$$\sigma_{\hat{y}}^2 = mean(\hat{y}^2) - \mu_{\hat{y}}^2$$

$$\sigma_{y\hat{y}} = mean(y\hat{y}) - \mu_{\hat{y}}\mu_y$$

$$\text{SSIM} = \frac{\left(2\mu_y\mu_{\hat{y}}+c_1\right)\left(2\sigma_{y\hat{y}}+c_2\right)}{\left(\mu_y^2+\mu_{\hat{y}}^2+c_1\right)\left(\sigma_y^2+\sigma_{\hat{y}}^2+c_2\right)}$$

Figure 2.19: An example of how the SSIM is calculated for a image using the sliding window approach, calculating the SSIM for local areas, rather than for the entire image. The original k-space is credited Knoll et al. [28].

Chapter 3

Methods

"Though this be madness, yet there is method in it."
— William Shapespeare, *Hamlet*

Research in deep learning - and neural networks in general - can be thought of as a multilayered process. First, large quantities of data must be acquired. In MRI, this translates to acquiring data from multiple scanners at different field strengths, with different contrasts and from multiple patients. However, this raw data may be inadequate for further analysis in and of itself; hence comes the second step in the multilayered process: The creation of a *deep learning pipeline*, being the automated set of processes from raw data to model training and finally model validation. The pipeline should enable users to control each and every step of the process, reducing the chance of errors that can affect the final performance of the model.

In this chapter, the deep learning pipeline is outlined. This includes the various transforms and augmentation techniques used on the data, the various acceleration factors and the downsampling scheme, hyperparameter fine tuning, model development, model improvements, and then to the final part of model validation, using a pre-created test dataset.

3.1 Introduction

The pipeline and model presented in the coming sections were programmed in the Python programming language [67]. To avoid overflowing the appendix, the codebase is kept separate, and can be found at the following GitHub repositories:

1. `https://github.com/JonOttesen/fMRI-models`
2. `https://github.com/JonOttesen/fMRI-training`.

The first repository contains the majority of the code. This includes everything from model scripts to the various training classes, with the intention of being an importable library. The second repository utilizes the first library to initialize the training and processing of the data.

3.1.1 MRI data

Acquiring a large enough training dataset for training MRI reconstruction is nigh impossible in the time given to finish a Master's project. As such, the dataset used for training, validation, testing and illustration is all credited the NYU fastMRI initiative [28, 22][1]. The fastMRI dataset includes raw k-space brain image sets totaling 4469 training volumes and 1378 validation volumes, all acquired at different field strengths (1.5 T and 3.0 T) and for different weighted images, such as T_1-weighted, T_2-weighted and fluid-attenuated inversion recovery (FLAIR). The validation set was randomly split in half, with 689 volumes in the training validation set and 689 volumes in the test set. The split of the original validation set ensures an independent test set after training for model evaluation besides a validation set used during training.

The exact data distribution of the fastMRI dataset is shown in table 3.1, where some of the T_1-weighted images were acquired after administration of a contrast agent. However, not all brain volumes are labeled, making a concrete estimate of the T_1-weighted distribution with and without contrast agent difficult, but the approximate distribution is $\sim \frac{1}{3}$ pre-contrast and $\sim \frac{2}{3}$ post-contrast [28].

Table 3.1: Overview of the dataset distribution for the different weighting schemes.

	T_2	T_1	FLAIR	Total
Training	2678	1447	344	4469
Validation	399	241	49	689
Test	416	215	58	689

3.2 Preprocessing

With all the formalities presented, this section revolves around the preprocessing of the raw k-space data provided by the fastMRI initiative, ensuring satisfactory model inputs and ground truth images.

3.2.1 Ground truth

As stated earlier, all the images provided by the fastMRI initiative are in form of raw k-space data. The volume data all have the following format:

$$V_i \in \mathbb{C}^{s \times c \times h \times w},$$

such that a single volume V_i contains s brain slices, with c coil elements, height h and width w; all of which are volume dependent, i.e., two volumes may have a different number of coil elements, image shape and number of slices.

As one might expect, there exists multiple possibilities regarding the ground truth. This all depends on what the model is supposed to reproduce. One such option might be the fully sampled k-space data. By this, the model would learn to reconstruct k-space data, which could be Fourier

[1] urlhttps://fastmri.med.nyu.edu/

transformed into image space (see equation (2.2.21)). However, common practice is to use the **magnitude image** as the ground truth [23, 19, 68, 22].

For the magnitude image, the j^{th} slice in the i^{th} volume $V_{i,j}$, k-space is zero-padded to a minimum size of 320×320 for all coil elements. This implies that the reconstruction matrix $\check{V}_{i,j}{}^{2}$ has 320 or higher phase and frequency encoding lines, albeit the outermost may be zero. As image space reconstruction is the norm, the inverse multidimensional Fourier transform is taken of the reconstruction matrix $\check{V}_{i,j}$

$$\check{I}_{i,j} = \mathcal{F}^{-1}(\check{V}_{i,j}). \tag{3.2.1}$$

The fully sampled slice image $\check{I}_{i,j}$ is the complex zero-padded image for all c coils, with shape $\check{I}_{i,j} \in \mathbb{C}^{c \times \check{h} \times \check{w}}$, where $\check{h}$ and $\check{w}$ are the height and width of the zero-padded image. This is the complex image for an arbitrary number of coils c, all with different sensitivities.

The magnitude image $\check{m}_{i,j}$ is given by the complex absolute and root-sum-of-squares (RSS) of $\check{I}_{i,j}$ such that

$$\check{m}_{i,j} = \sqrt{|\check{I}_{i,j,0}|^2 + |\check{I}_{i,j,1}|^2 + ... + |\check{I}_{i,j,c-1}|^2} \tag{3.2.2a}$$

$$= \sqrt{\sum_{k=0}^{c-1} |\check{I}_{i,j,k}|^2}. \tag{3.2.2b}$$

From this, the magnitude image $\check{m}_{i,j}$ is the square root of the complex absolute of the sum of squares of the coil elements.

The second to last step is to center crop the image height and width to exactly 320×320, and by this ensuring similar image size and removing any FOV oversampling.

Lastly, the final step is model dependent and not mandatory. In this step, the image is normalized, i.e., standardizing the image with a mean of zero, and a standard deviation of one, defining the standardized ground truth as

$$\text{GT} = \frac{\check{m}_{i,j}^{(crop)} - \text{mean}(\check{m}_{i,j}^{(crop)})}{\text{std}(\check{m}_{i,j}^{(crop)})}. \tag{3.2.3}$$

3.2.2 Input

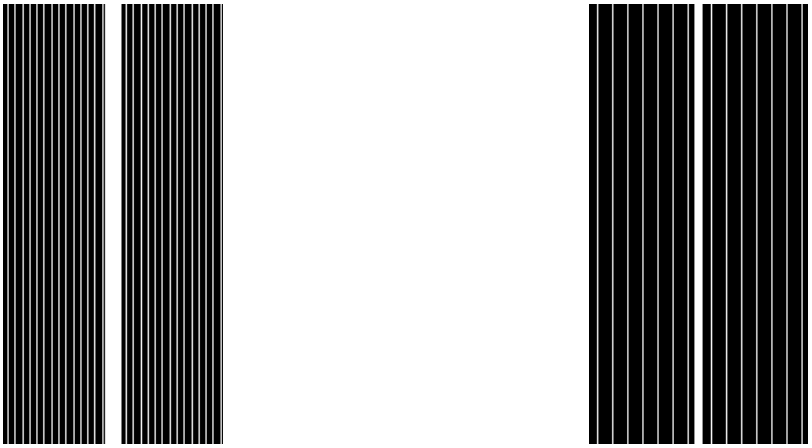
In subsection 2.4.1, it was established that by increasing the sampling distance Δk or decreasing the k-space span k_{max}, various image artifacts are introduced. These artifacts can be simulated by manipulation of the fully sampled k-space data, allowing for the simulation of under-sampled data and artifacts as would happen during an MRI scan. The first undertaking is therefore to simulate the downsampled model input.

As the imaging time is proportional to the number of phase encoding steps, simulated undersampling is carried out by masking some of the phase encoding lines, substituting the original values with zeros. This is done by the appliance of Boolean masks with equal shape as that of the image, and two such masks are shown in figure 3.1. The masks shown in figures 3.1a and 3.1b are the so-called

[2] Zero-padded variables are denoted by check.

equidistant masks, with acceleration factors of 4x and 8x, respectively, where the acceleration factor is defined as the inverse of the fraction of omitted lines. Furthermore, it should be commented that the center lines are kept to allow for the coil sensitivities to be estimated [68].

To ensure a small amount of data augmentation, the lone equidistant encoding lines in figure 3.1, can be shifted. This is done by initializing the very first line randomly between zero and the line spacing, shifting every subsequent line, resulting in a slightly different undersampled image.



(a) Equidistant mask for 4x acceleration with 0.08 center fraction

(b) Equidistant mask for 8x acceleration with 0.04 center fraction

Figure 3.1: Boolean equidistant masks with acceleration 4x and 8x for k-space undersampling, the black lines are the undersampled lines.

Before downsampling, k-space is zero-padded to the minimum size of 320×320, then inverse Fourier transformed into image space and cropped to the size of 320×320, by this eliminating any oversampling along the phase and frequency encoding direction. The data is subsequently Fourier-transformed back into k-space. The now cropped k-space data with size of 320×320 is downsampled along the phase encoding direction by the masks shown in figure 3.1.

The initial zero-padding before undersampling might seem strange, as the downsampling is done on a larger portion of k-space than what is sampled, possibly undersampling phase encoding lines that are already zero. The reason for this is because many of the volumes (V_i) from the fastMRI initiative dataset [28] are already pre-zero-padded, making it difficult to know the actual sampled portion of k-space.

After the data is downsampled, the remaining steps are identical to that of the ground truth image, except the image cropping, as this is already done. First; the slice is inverse Fourier-transformed (3.2.1). Second; the RSS and the complex absolute are calculated (3.2.2b). Finally; the slice is standardized with a mean of zero, and standard deviation of one (3.2.3).

The preprocessing described in subsections 3.2.1 and 3.2.2 is illustrated in figure 3.2 for some arbitrary slice from the training dataset [28]. For simplicity, the number of coils visualized is truncated to four, with only the real part of the dataset shown. Notice how the data in figure 3.2

is zero-padded before the zero-padding step. Furthermore, all the ending branches are designated a letter, which will later be used to connect the different inputs and ground truths to the different models.

3.2.2.1 Memory considerations

In deep learning, model training takes time and the memory on the graphics processing unit (GPU) is restricted. To minimize such constraints when using raw *k*-space data as a model input (as will be done later), any FOV oversampling in image domain is cropped and Fourier transformed back, only keeping the relevant image domain information in the *k*-space domain, potentially decreasing the input size with over 50%. With less memory bound to the input image, more memory can be allocated to the deep learning model, allowing for potential improvements requiring more memory. Note that the oversampling from the fastMRI dataset is not limited to the frequency encoding direction, rather, many samples are additionally oversampled along the phase encoding direction. Clinically the phase encoding direction is rarely oversampled, making it unrealistic to use oversampled data for model training.

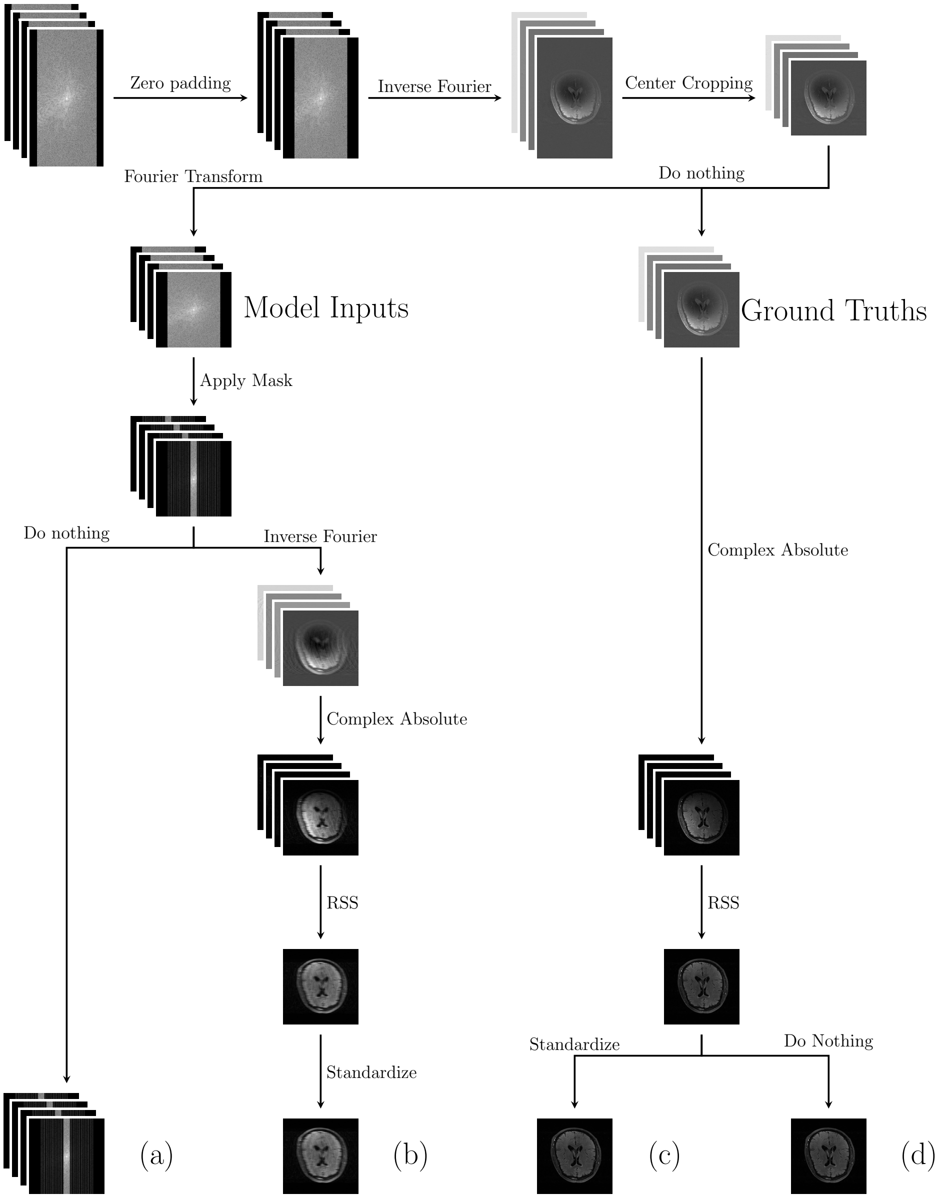


Figure 3.2: A visual representation of how the preprocessing is performed on real training data for a single slice (with four independent coil channels and only the real components) from the fastMRI dataset [28]. The letters at the end are used to enumerate the different inputs and ground truths, with (a) and (b) being inputs while (c) and (d) being ground truths.

3.3 Deep learning training

With the model input and ground truth readily defined in section 3.2, and the preprocessing shown in figure 3.2, the next step is model development and training.

3.3.1 Aims and goals

The main goals for the following sections of model training are:

1. To train a baseline model for MRI reconstruction using the downsampled magnitude input - magnitude output reconstruction scheme.
2. To utilize state-of-the-art architectural design from the deep learning research community, developing a highly customizable model for the downsampled magnitude input - magnitude output reconstruction scheme, outperforming the baseline model.
3. To introduce the newly developed model into the more hardware intensive state-of-the art variational network (VarNet) [68] that already achieves cutting edge MRI reconstruction, ideally achieving comparable results to the high-end reconstruction models.

To improve upon the results of high-end reconstruction models is infeasible because the high-end reconstruction models reported in literature are trained on extensive computational resources with multiple GPU's each having 32GB of memory. Instead, the main focus of the subsequent models is to achieve similar or comparable accuracy without the need for multiple high-end GPU's, by using the architectural recommendations from the deep learning community.

3.3.2 Models

In this thesis there were trained a total of four different models. The first two models follow a magnitude input - magnitude output training scheme. The first of these models and the baseline for the deep learning MRI reconstruction is the so-called U-Net architecture [16]. In the time since the U-Net model was first published in 2015, the deep learning community have made multiple discoveries regarding model architectures. The second model is therefore a modified U-Net model dubbed "Res-Udet", this model implements many of the newer discoveries from the deep learning community. These discoveries include squeeze-and-excitation [69], skip connections [70], a so-called bi-directional feature pyramid network (BiFPN) [15] and the sigmoid linear unit (SiLU) activation function [71].

The two last models follow the more complex VarNet [68]. The VarNet is an open source MRI reconstruction framework that achieves state-of-the art reconstruction images. The first VarNet trained followed the original version from literature, and it uses the U-Net model as deep learning architecture within the reconstruction framework. The second model trained is a modified VarNet that applies the Res-Udet model instead of the U-Net model.

3.3.3 Model training

3.3.3.1 Training iterations

When training deep learning models, it is common practice to iterate through the entire dataset (randomly or not) in so-called epochs, where one epoch is a single iteration through the entire dataset and training can continue for hundreds of epochs. This is not the case for MRI reconstruction; complex coil data is large with a noticeable transfer time, and the preprocessing is time consuming. These elements combined makes epoch-wise training time-wise unrealistic. Instead, the training was carried out in iterations, where one iteration loops over a randomly selected subset from the training data. The same applies for validation, as only a subset of the validation set was selected for each validation period.

The number of slices for a single training iteration was chosen to be 4000 whereas the number of validation slices was chosen to be 2000. Unless stated otherwise, every model was trained for a total of 250 iterations, this equals 12-13 epochs if epoch-based training was used.

3.3.3.2 Hyperparameters

PyTorch [56] (v.1.3.1, v.1.6 and v.1.7) was used for model development and automatic differentiation to calculate the gradient updates from equation (2.5.6). Adaptive moment estimation (ADAM) [72], a gradient-descent based algorithm was used for parameter optimization using the updates calculated by PyTorch in a similar manner as equations (2.5.7a) and (2.5.7b). The ADAM parameters chosen were the default PyTorch parameters with $\beta = (0.9, 0.999)$, $\epsilon = 10^{-8}$, a weight decay of zero, and *amsgrad* enabled for improved convergence [73]. The learning rate was chosen at $2 \cdot 10^{-3}$ after an initial grid search, with step-wise learning rate decay every 100^{th} iteration. By this, the learning rate decays with the factor $\gamma = 0.1$ every 100^{th} iteration, improving both optimization and generalizability [74]. The batch size used was 4, following the suggestions of Masters and Luschi [75], with the batch size enforced constant unless stated otherwise.

3.3.3.3 Loss function

The loss function to be minimized is a linear combination of the MAE ($l1$) and the SSIM (2.7.6) given by

$$\text{loss}(\hat{y}, y) = (1 - \text{SSIM}(\hat{y}, y)) + l1(\hat{y}, y), \tag{3.3.1}$$

following the approach taken by multiple research groups [23]. However, no measures were taken to ensure similar orders of magnitude between the SSIM and $l1$ loss. Furthermore, the SSIM was implemented with a Gaussian weighted kernel with a sliding window size of 7 and standard deviation of $\sigma = 1.5$, with the dynamic range, (L from equation (2.7.3)) set as the numerical difference between the minimum and maximum pixel intensity for the ground truth image. Note that this implementation differs from the fastMRI initiative implementation [28], instead following the Tensorflow implementation [76] using a Gaussian kernel. This was done to reduce the possibility of so-called blocking artifacts as suggested by the original authors of the SSIM [77].

A simplified summary of the process of training a neural network is shown in algorithm 1, where the main steps of network training is illustrated.

Algorithm 1: How to train Neural Networks

```
Result: A trained neural network
Initialize model
Set initial learning rate
Create loss function
for iterations do
    for input, truth in training do
        Preprocess the input and ground truth
        Predict reconstruction
        Calculate the loss between the ground truth and the prediction
        Update the weights and biases based on the loss
    end
    for input, truth in validation do
        Preprocess the input and ground truth
        Predict reconstruction
        Calculate loss between ground truth and prediction
    end
    if decay learning rate then
        Update learning rate
    else
        Keep learning rate unchanged
    end
end
```

3.4 U-Net

3.4.1 U-Net model

The U-Net model consists of two overarching paths. First, a contracting path, decreasing the image resolution while increasing the channel dimension. Second, an expansive path, decreasing the channel dimension while increasing the resolution. The slightly modified U-Net architecture used as a baseline is shown in figure 3.3. It consists of gray intermediate states with n channels and a series of operator arrows for the blue 3×3 convolutions (zero-padded convolutions) including instance normalizations and the ReLU activation function, the yellow/orange copy and concatenate operations, the red 2×2 maxpool layers, the green 2×2 bilinear upsampling layers and the cyan 1×1 convolution to the output.

3.4.2 Training

The U-Net model trained had $n = 128$ channels, scaling the model in width rather than depth. There were trained two different versions of this model, one for 4x and one for 8x accelerated equidistant downsampling. In total, two U-Net versions, one for each acceleration factor. The model input was the letter (b) and ground truth was the letter (c) from figure 3.2, the Boolean

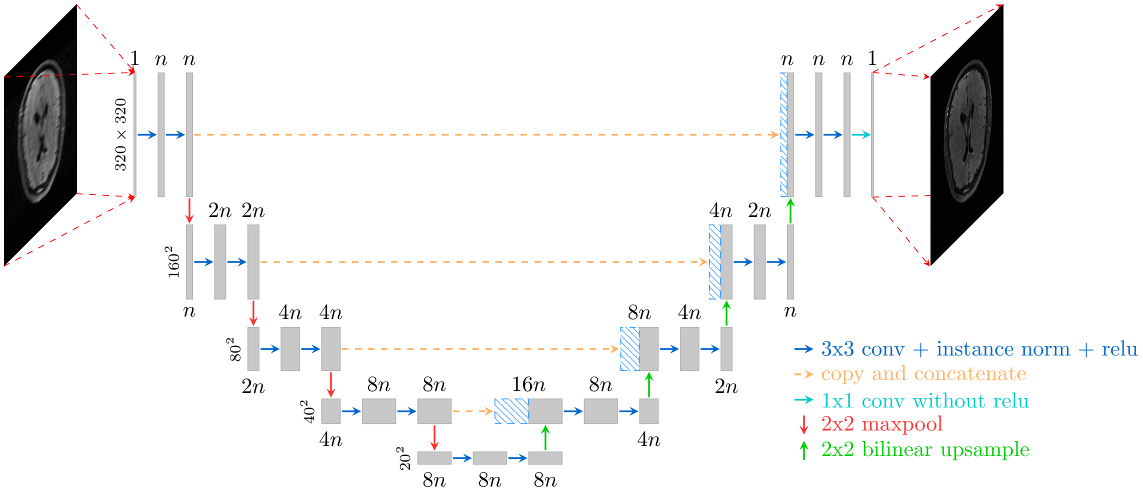


Figure 3.3: The baseline U-Net architecture, the arrows represent different operations used in the network, the grey blocks represent intermediate states, and n represents the number of channels in the intermediate states.

mask was interchanged for the different accelerations. The U-Net training lasted for about 3.5 days for both the 4x model and the 8x model on an Nvidia RTX2080 GPU, using approximately 8GB of memory.

3.5 Architectural improvements

In this section, the topic is the architectural additions from the deep learning community, these are *squeeze-and-excitation blocks* [69], *residual connections* [70] and BiFPN [15].

First, some terminology is in order; inspecting figure 3.3 - especially the blue arrow operators - notice that they represent a cascade of operations, not a single 3×3 convolutional layer. Such a series of operations are referred to as a *block*, with a block being a series of operations performed on some input. Hence, a block can range from an entire model, to a single convolutional operator. Most commonly, blocks are used to denote subunits of an overarching model that are very often repeated.

3.5.1 Residual connections and squeeze-excitation blocks

3.5.1.1 Squeeze-and-excitation

Squeeze-and-excitation is an architectural addition introduced by Hu, Shen, and Sun [69] meant to improve the representational power of the network. This is done by explicitly modelling the interdependency between channels by adaptive recalibration of channel-wise feature response [69].

The general idea is that some input of size $U \in \mathbb{R}^{C \times H \times W}$ is passed through a *squeeze* operation, reducing the spatial dimension to $U_c \in \mathbb{R}^{C\times 1\times 1}$ [69]. Notice that the shape of U_c can be thought of as a column vector, as required by the feedforward network (2.16). The squeezed input U_c is then put through an *excitation* operation consisting of two fully connected layers, where the final fully connected layer has the same number of neurons as there is channels. The activation function for this output is the sigmoid activation function, enforcing the elements in the output column vector to be between $[0, 1]$, this output is multiplied neuron-to-channel with original input U, weighting the channel importance.

In summary, the squeeze-and-excitation block performs learnable channel-based weighting determined by the input, showing improved accuracy with a very minor computational cost [69].

3.5.1.2 Residual connections/Skip connections

Residual connections, also known as *skip connections*, is a technique used in deep learning where a prior intermediate state is added to a latter intermediate state. There are a multitude of different proposed skip connection methods in literature. One of the most cited is the so-called residual connections, also known as short skip connections, proposed by He et al. to combat the vanishing gradient problem [70].

The previously introduced U-Net model already applies so-called *long skip connections*, located in the concatenating path in figure 3.3, with these connections having shown to help the recovery of the full spatial resolution [78]. Furthermore, the combination of short and long range skip connections have shown increased accuracy for segmentation tasks, helping gradient flow [78], and short skip connections are standard on the best performing image recognition network [14].

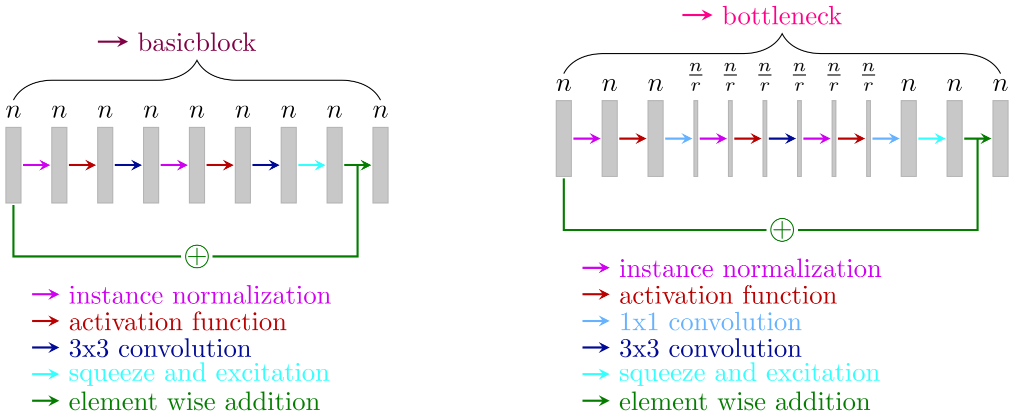
The short skip connections were added by the inclusion of the so-called *basicblock* and *bottleneck* block, first introduced in [70]. The implemented basicblock and bottlneck block followed the adjustments made by He et al. [79] rearranging the order of the activation function, normalization and the convolutional layer in the block [79].

The basicblock is shown in figure 3.4a, and consists of two normalization layers, two activation functions and two convolutional layers [79], where the last operation is the added squeeze-and-excitation process briefly mentioned above. The final green arrow represents element-wise addition between the input and the output of the squeeze-and-excitation operation. Note that the number of channels is kept constant throughout the entire block, allowing for element-wise addition in the last step.

The second addition from He et al. [79] is the so-called *bottleneck*, consisting of a 1×1 convolutional layer for channel reduction, a 3×3 conventional layer, and a final 1×1 conventional layer to restore the input channel dimension [70]. As evident, the number of channels is reduced in the middle, reducing the total number of parameters in the bottleneck, as a 3×3 convolution is parametrically more expensive than a 1×1 convolution. The bottleneck is illustrated in figure 3.4b, where r is the reduction in the number of channels in the middle of the bottleneck.

3.5.2 Bi-directional feature pyramid network

The BiFPN is an architecture proposed in [15], and it was shown that models including the BiFPN have improved accuracy [15]. Furthermore, there are positive results for improved accuracy in lung



(a) Basicblock

(b) Bottleneck

Figure 3.4: The basicblock and bottleneck block from He et al. [70], the implementation follows the re-ordered implementation by He et al. [79]. The arrows represents operations and the gray blocks are intermediate states with n-channels or n/r-channels.

segmentation with the BiFPN implemented in U-Net [80].

The BiFPN architecture is shown in figure 3.5, where the initial inputs correspond to the different resolutions in U-Net (3.3). It should be noted that figure 3.5 is highly simplified, omitting the normalization, activation and even the size of the conventional layers for readability. Furthermore, the upsampling and downsampling are in the form of convolutions to ensure correct channel dimension, a problem not encountered in the original BiFPN [15]. By this, it is more correct to call it a BiFPN-like architecture, as it is modified for the use in U-Net.

As evident from the U-Net architecture in figure 3.3, there exists an inherent one-way informational flow throughout the model, and the concatenating operators are unable to utilize all the extracted features further down the downsampled layers. The main goal of the BiFPN layer is to enable cross-scale feature fusion between layers [15], reducing the inherent one-way information flow in the U-Net architecture, and by this introducing a bi-directional informational flow, as seen in figure 3.5, allowing for multi-scale feature fusion by weighted additions of features extracted at different resolution levels in the network.

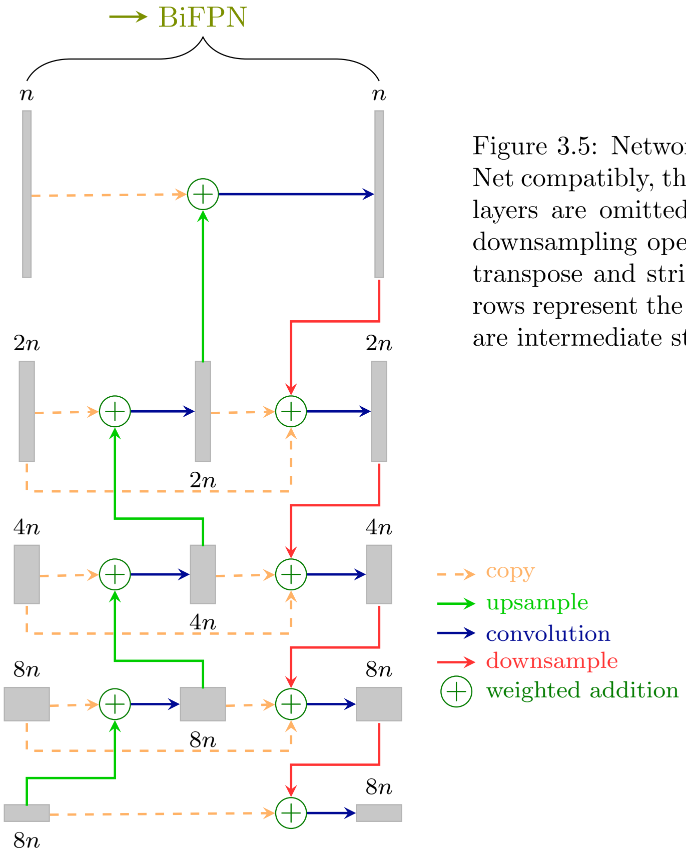


Figure 3.5: Network design of the modified BiFPN for U-Net compatibly, the activation functions and normalization layers are omitted for readability. The upsampling and downsampling operators in the BiFPN are in the form of transpose and strided convolutions, respectively. The arrows represent the different operations and the gray blocks are intermediate states with n-channels.

3.6 Res-Udet

The overall architecture, including the various architectural extensions to the standard U-Net model is outlined in figure 3.6, this model is dubbed ”Res-Udet”.

3.6.1 Res-Udet model

The activation function used in the Res-Udet model is the SiLU activation function given by

$$\mathrm{SiLU}(x) = \mathrm{sigmoid}(x) \cdot x. \tag{3.6.1}$$

Research suggests similar or improved accuracy in comparison to the widely popular ReLU activation function [81, 71].

The Res-Udet model is similar to the U-Net model, consisting of a contracting path and an expansive path, with the model shown in figure 3.6. It consists of the gray intermediate states with n channels and a series of operator arrows for the blue 3×3 convolutions (zero-padded convolutions) including

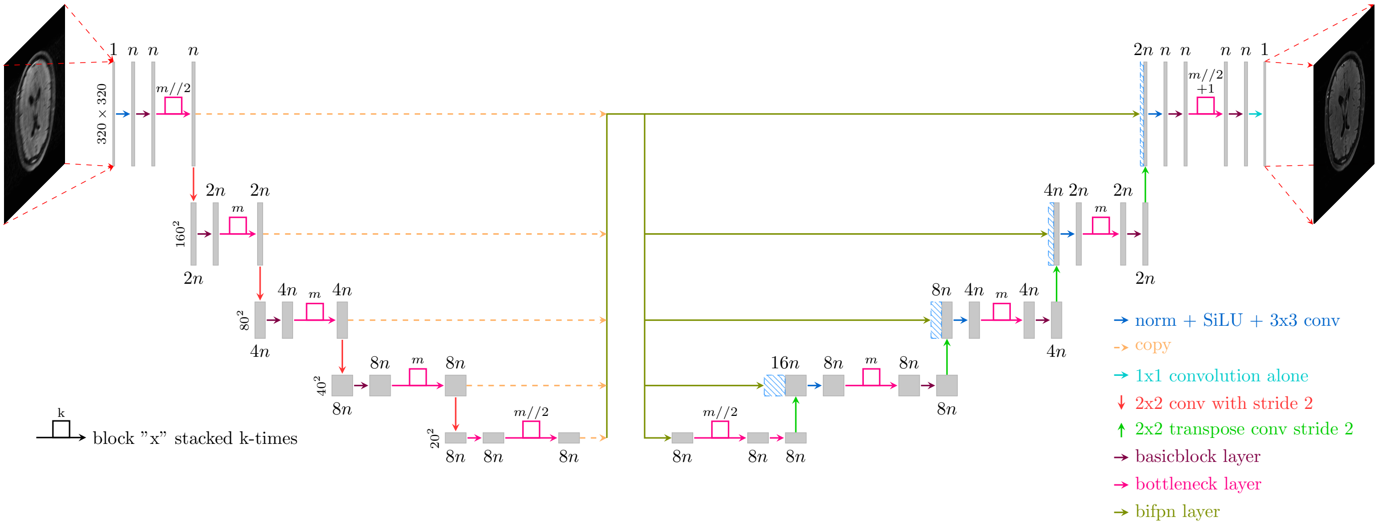


Figure 3.6: The Res-Udet model incorporating basicblocks, bottleneck blocks and the BiFPN layer, where n denotes the number of channels in the gray intermediate states, and the arrows represent the different operations. The looped or rectangular formed arrows represent a block repeated x-times, with the bottleneck block repeated m or $m//2$ times for each looped arrow, where $//$ is the modulus operator.

instance normalization and the SiLU activation function, the yellow/orange copy arrow, the cyan 1×1 convolution, the red strided 2×2 convolution, the green 2×2 transpose convolution, the burgundy basicblocks, the pink bottleneck blocks and the golden green BiFPN block.

3.6.2 Training the model

3.6.2.1 Configuration possibilities

The model depicted in figure 3.6 was trained in a similar manner to that of U-Net. Although the regime of training was the same as for the U-Net model, the number of free variables is higher, with the model in figure 3.6 having a total of three free variables:

1. n: the number of channels
2. m: the number of bottleneck layers
3. the number of BiFPN layers.

The main question to be investigated is then: what is the ideal combination of channels n, bottleneck layers m and BiFPN layers?

Fortunately, this question has already been somewhat answered by the Tensorflow research group, where the major find was that model accuracy scales with a linear increase in model width (number of channels), model depth (number of convolutional layers) and the input resolution (i.e., pixel count), with the accuracy quickly saturating when only scaling one variable [14, 15].

The model length was scaled by increasing the number of bottleneck layers, without massively increasing the computational load and the number of parameters. This stems from the fact that the 3×3 convolution in the bottleneck layer is reduced by a factor r as evident by figure 3.4b. Furthermore, the resolution of 320×320 was fixed, both to reduce the computational complexity of the model but also since MR-images are natively around a resolution of 256×256, potentially missing out on the improved accuracy as suggested by [14] for an increased resolution, but at the same time, the memory requirement for a larger image was reduced.

3.6.2.2 Model configuration

In total, there were trained four different Res-Udet models with different combinations of bottleneck layers and the BiFPN. The different model configurations are given in table 3.2, and all models had the same number of channels n. The input and ground truth images for these models were the same as for U-Net, the model input was the letter (b) and the ground truth was the letter (c) from figure 3.2.

Table 3.2: Overview of the different configurations of the Res-Udet model

Acceleration	channels n	bottlenecks m	BiFPN
4x acceleration	32	0	0
	32	4	0
	32	4	1
8x acceleration	32	4	1

The $m = 4$ models with the BiFPN layer were both trained on an RTX 3090 GPU, using approximately 12GB of memory, and had a total training time of 7 days. The remaining networks were trained on an RTX 2080, all using less than 11GB of memory, and was trained for 9 days. Note that the training time does not equate to GPU usage, and the majority of the training time consisted of data transfer from the hard drive to memory. The majority of the difference in training time between the U-Net and Res-Udet models were because of a difference in data transfer time between the systems used for training.

3.7 Variational network

3.7.1 Network explanation

VarNet is a parallel imaging based deep learning method, where the network itself learns the coil sensitivity estimation by utilizing the low spatial frequency center in the downsampling masks (3.1) [68]. For higher acceleration factors, traditional sensitivity calibration methods become inaccurate. This problem can potentially be alleviated by using a deep learning network for coil sensitivity prediction [68]. In conjunction to sensitivity prediction, the network itself follows an iterative approach, refining the output for each intermediate step [68], while at the same time re-introducing the original sampled k-space lines in each iteration.

The model input ((a) from figure 3.2) is first used to estimate the complex and real components of the coil sensitivities using a deep learning model (originally U-Net). The estimated coil sensitivities are used to estimate the complex coil-summed image by the so-called reduce operation [68]

$$R(\boldsymbol{x_1}, \boldsymbol{x_2}, ..., \boldsymbol{x_{Coils}}) = \sum_{i=1}^{\text{Coils}} \overline{S_i}\boldsymbol{x_i} \tag{3.7.1}$$

and an expand operation expanding a coil-summed to separate images for all coils [68]

$$E(\tilde{\boldsymbol{x}}) = (S_1\tilde{\boldsymbol{x}}, S_2\tilde{\boldsymbol{x}}, ..., S_{Coils}\tilde{\boldsymbol{x}}). \tag{3.7.2}$$

The VarNet consists of so-called *cascades*, and in a cascade both the reduce and expand operations are carried out. The input k-space is first inverse Fourier-transformed to image space and reduced by equation (3.7.1). The reduced image is used as an input into a deep learning model (originally U-Net), the model output is then expanded by equation (3.7.2) and Fourier-transformed back to k-space. In k-space, the original scanner sampled date are substituted into the newly predicted k-space with some set of weights. This k-space data is the output of a cascade and has the same shape as the original input. The output data from the previous cascade is used as the input for a new cascade, and the number of cascades in the VarNet is a chosen hyperparameter. Even if the cascades are repeated a given number of times, the deep learning parameters are **not** shared between cascades, although the sensitivity maps are. The final model output is the magnitude image on the output k-space from the last cascade.

3.7.2 Training the variational network

3.7.2.1 Hyperparameters

The VarNet from literature has shown state-of-the-art results, and was previously trained on 32 GPU's each with 32 GB memory [23]. To successfully train a VarNet based model with the improved architecture, the batch size was reduced from the previously used four to one, with research suggesting a too small batch size may affect the model performance [75].

The training of the VarNet followed many of the previously implemented schemes. The loss function and learning rate were kept unchanged, with the loss function following equation (3.3.1) and the learning rate was kept as $lr = 2 \cdot 10^{-3}$. However, the number of iterations was reduced to 200 (a single iteration took 1-3 hours) and the learning rate decay scheme was modified to decay every 150$^{\text{th}}$ iteration. Furthermore, the number of validation images was reduced to 1000 to reduce training time. The total number of training samples seen by the VarNets in this thesis was vastly reduced when compared to [23], with their network being trained for 50 epochs, this is about five times the number of iterations used in this thesis.

3.7.2.2 Model configurations

In the original VarNet, there were a total of 12 cascades, and the deep learning model in each cascade was the U-Net model with $n = 18$ channels. In this thesis, two versions of the VarNet were trained both with 12 cascades: one VarNet with the U-Net as the deep learning model, and one where the U-Net was interchanged with the Res-Udet. The U-Net version of the VarNet was scaled in width. This was possible because of the reduced memory requirement from the smaller k-space input from figure 3.2. The Res-Udet version of the variation network was also scaled in width, and was not scaled in length enforcing $m = 0$ additional bottleneck layers without the BiFPN layer. Both versions of the VarNet were scaled to have approximately equal number of trainable parameters. Note that the architectural improvements in the Res-Udet model used in the VarNet are the residual connections, squeeze-and-excitation and the SiLU activation function, not any additional length scaling or the BiFPN. The U-Net based VarNet was re-trained in this thesis to ensure a fair comparison between the original VarNet and the modified VarNet trained, enforcing similar hardware restrictions and the same number of iterations.

3.7.2.3 Training regime

The model input was changed from the magnitude images to the downsampled k-space data, (a) from figure 3.2, and the ground truths were the non-standardized (d) from figure 3.2. All trained VarNets were trained conjunct on both 4x and 8x acceleration simultaneously, with a 50/50 chance for either a 4x or 8x downsampled input. During validation, the number of 4x and 8x downsampled inputs were split equally, such that 500 of the input images were 4x downsampled and the remaining 500 images were 8x downsampled.

The original VarNet re-trained in this thesis was trained on a single Nvidia RTX2080 for ~ 7 days, while the modified version was trained on a single Nvidia V100 GPU with 32GB of memory for ~ 12 days. Both the original and the modified VarNet maxed out their respective GPU's memory, using ~ 10.5GB and ~ 25GB, respectively, for training. If the modified VarNet had applied length scaling, the training time would approximately double, even for a reduced network width.

3.8 Reconstruction evaluation

In order to evaluate the reconstruction capabilities of the different models, the independent test that originally was part of the validation set was used, with the ground truth images and inputs following the same procedure as shown in figure 3.2. It is important to note that this dataset was completely new for the model, and unlike the validation dataset, was not used to validate the accuracy during training.

Unlike during model training where a single slice from a volume was evaluated in loss function independently, during evaluation the entire volume was evaluated as one. The volumetric evaluation reduced the number of evaluation samples to the 689 volumes in the test dataset. However, it allowed for comparison between literature and this thesis. It should be emphasized 689 volumes corresponds to 10942 brain slices, reducing any potential bias caused by a too small test dataset.

3.8.1 Metrics

The metrics used for the volumetric evaluation were the SSIM (2.7.6), NMSE (2.7.7) and PSNR (2.7.8). However, as the SSIM evaluates similarities in spatial image structure, rather than similarities between individual pixels was acknowledged as the most important metric.

Note, the maximum value $\max(y)$ in the PSNR and the dynamic range in the SSIM (2.7.3) are chosen with respect to the entire volume, not individual slices as during training.

3.8.2 Sparse sampling

As a comparison to the deep learning reconstructed images, sparse sampling was used as an alternative reconstruction software. The sparse sampling method used was implemented by the Berkeley advanced reconstruction toolbox (BART) [82], making it possible to compare the current state-of-the-art methods with the deep learning reconstruction. For sparse sampling, the k-space data was randomly downsampled fulfilling the requirement set by sparse sampling, except a small center portion used for coil sensitivity estimation.

The implemented sparse sampling method followed the approach by Zbontar et al. [22] which used the method introduced by Uecker et al. [83]. The regularization parameter λ from equation (2.4.8) was chosen as $\lambda = 10^{-2}$ and the optimization algorithm ran for 200 iterations for each slice.

An important comment regarding the sparse sampling method is that the ground truth from figure 3.2 was **not** used for the reconstruction evaluation. The reason for this is because the BART does not use the RSS method for coil combination during reconstruction. Instead, the BART uses a parallel imaging based coil combination method that applies the coil sensitivities. Any comparison between these methods will therefore tend to be biased towards worse metrics since the methodology differs at a fundamental level. To emulate a somewhat correct scoring, the ground truth was set to be the reconstructed image using the full k-space data. Nonetheless, a direct comparison between the sparse sampling method scores and RSS reconstruction is inevitably biased as the noise characteristics differs for different reconstruction algorithms [84, 6], and the SSIM is sensitive to differences in noise as seen by figure 2.18. A direct metric comparison should therefore not be seen as absolute, and an image quality comparison is the more important measure of reconstruction quality.

3.8.3 Post-processing

Although the output images from the network were valid in and of themselves, some post-processing was necessary for cross research comparison and better handling of negative values in the NMSE and PSNR metrics. These negative values cause an overshoot in the NMSE and an undershoot in the PSNR, as the denominator in the NMSE and nominator in the PSNR don't account for negative values, reporting worse metrics than what is really the case.

The first issue addressed was the data over- and under-shoot caused by the inclusion of negative values. This issue was corrected by de-standardizing the reconstructed image, using the known mean and standard deviation of ground truth images. From this, the de-standardized output is given by

$$\hat{y}' = \frac{\hat{y} + \text{mean}(y)}{\text{std}(y)} \tag{3.8.1}$$

where y is the ground truth and $\hat{y}$ is the model output. The problem with negative values only appeared for magnitude reconstruction for the U-Net and Res-Udet models, since they used the standardized ground truth ((c) from figure 3.2) as the ground truth during training.

3.8.4 Statistical significance

Since multiple models are trained, it is of interest to measure whether the difference in performance is of statistical significance. Ideally, one would measure this by training the model multiple times using k-fold cross validation, and create a distribution of the mean test scores for each model. The mean test score distribution could then be used to test statistical significance using the Student's t-test. However, training the same model k-times (e.g 10 or 20) for all four models is unrealistic, since each model used anywhere from 3.5 - 12 days worth of training time. Instead, it was assumed that the statistical difference from the random parameter initialization was of no major significance. Furthermore, it was assumed that since the test dataset consisted of $\sim$ 10000 slices, it was large enough to not exhibit any noticeable degree of impact on the models.

To evaluate the statistical significance of the four models, the paired two sample Student's t-test [85] was used on the test dataset distribution for all three metrics. The Student's t-test was used to compare in-between models, e.g., compare the U-Net distribution to the Res-Udet distribution for all metrics, this is a total of six comparisons for each metric and acceleration factor. Common practice is to rejected the null hypothesis if the p-value is $p < 0.05$. However, since there is a total of six comparisons, the statistical significance required for a null hypothesis rejection was Bonferroni corrected to $p < \frac{0.05}{6}$. The paired two sample Student's t-test was calculated by using the Scipy implementation [86].

Chapter 4

Results

> *There are two possible outcomes: If the result confirms the hypothesis, then you've made a measurement. If the result is contrary to the hypothesis, then you've made a discovery*
>
> — Enrico Fermi

In this chapter, the results of the deep learning accelerated MRI are presented, both with respect to the training process itself, but also the independent test set. The presented results include the various reconstructed images with their respective ground truths and their numerical SSIMs for the different reconstruction models and the different sequence types.

4.1 Deep learning models

The training and validation loss for the U-Net and Res-Udet models for 4x and 8x acceleration are presented in figures 4.1 and 4.2, respectively, omitting the first 10 iterations. The saved model states are represented by $\star$ in their respective training plots. The Res-Udet training plot for 4x acceleration includes the validation loss for the BiFPN-less models from table 3.2 denoted with "x", only plotting the validation loss every fifth iteration.

The training and validation loss for original and modified VarNets are presented in figure 4.3. The loss of the original VarNet is denoted with "x", only plotting the loss every fifth iteration. The saved models from the VarNets are the state after the 200$^{\text{th}}$ iteration.

The SSIM, NMSE and PSNR for the reconstructed 4x and 8x accelerated images from the test dataset using the four reconstruction models and the sparse sampling method are given in table 4.1. Table 4.1 includes the number of trainable model parameters and the mean and standard deviation of the inference time for 1000 forward propagations for each model, where inputs for the VarNets had 20 coils.

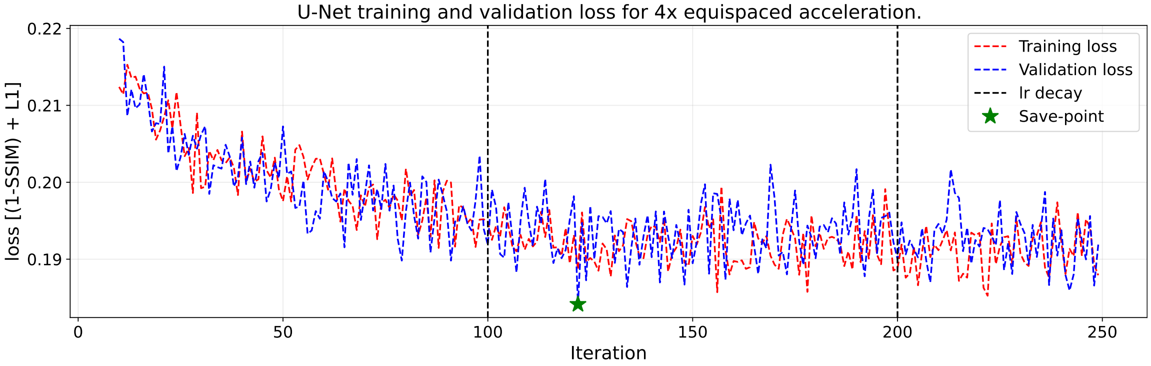
A violin plot of the SSIM distribution for the reconstructed T_1-weighted, T_2-weighted and FLAIR-weighted images with the U-Net, Res-Udet, the original VarNet and the modified VarNet models is shown in figure 4.4 for both acceleration factors. The violin plot for the entire dataset with all four models for both acceleration factors is presented in figure 4.5, and the y-axis is truncated to a maximum value of $y = 0.83$. The p-values from the Student's t-test for the metric distributions between the different models are given in table 4.2.

Four sets of reconstructed images using the four reconstruction networks on the FLAIR-weighted, T_1-weighted post-contrast, T_1-weighted pre-contrast and T_2-weighted images for 4x acceleration are shown in figures A.1, A.3, A.5 and A.7, respectively. The corresponding 8x accelerated images are shown in figures A.2, A.4, A.6 and A.8. These images have similar volume SSIM to that of the mean SSIMs from table 4.1. All these reconstructed images have color-coded areas of interest highlighted:

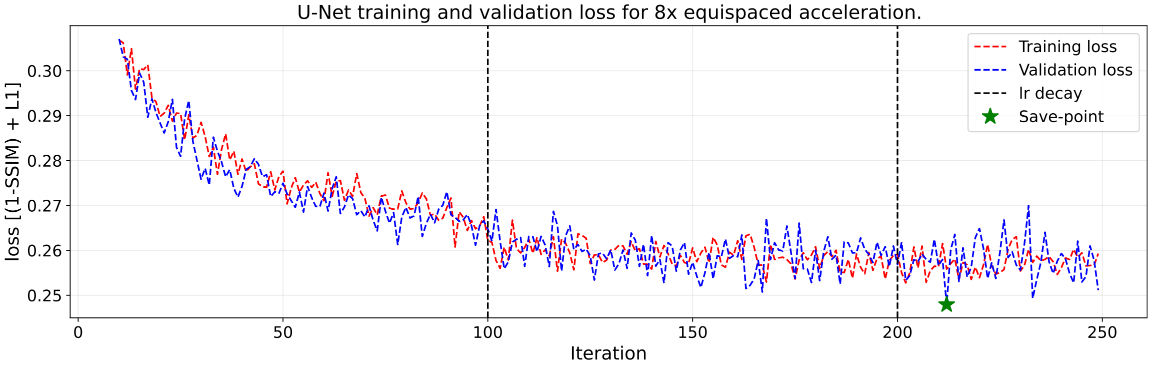
1. The color red highlights areas of interest with missing or un-satisfactory reconstruction.
2. The color blue highlights differences in reconstruction between the model pairs U-Net - Res-Udet and OGVarNet - VarNet.
3. The color green highlights differences in reconstruction between the magnitude input - magnitude output and the VarNets.

Lastly, the modified VarNet reconstructed T_2-weighted image with the worst SSIM is given in figure A.9.

The sparse sampling reconstructed FLAIR-weighted, T_1-weighted post-contrast, T_1-weighted pre-contrast and T_2-weighted images for 4x acceleration are given in figure A.10. These images are the same as those used for the deep learning accelerated reconstruction.

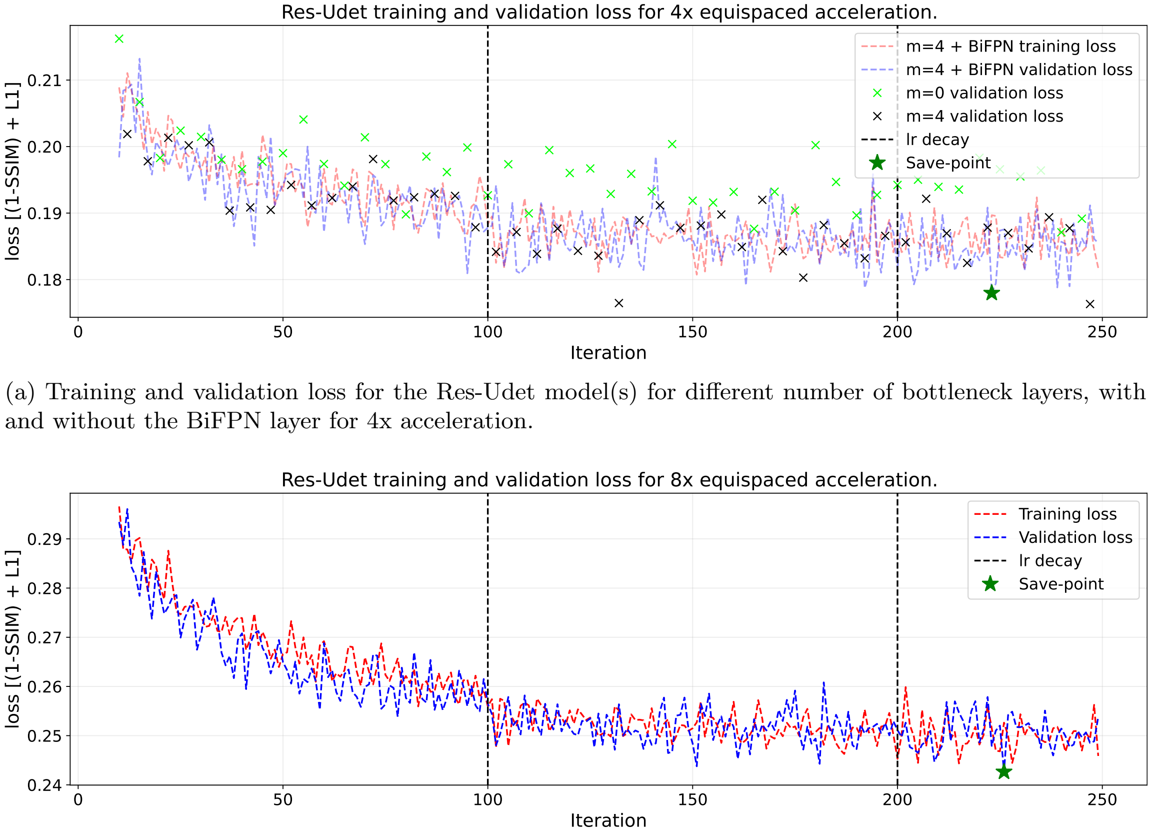


(a) Training and validation loss for the U-Net model with 4x acceleration.



(b) Training and validation loss for the U-Net model with 8x acceleration.

Figure 4.1: Training and validation loss for the U-Net model (see figure 3.6) with 4x and 8x acceleration for k-space undersampling with the equidistant subsampling filter (see figure 3.1). The training was for a total of 250 iterations with an initial learning rate of $2 \cdot 10^{-3}$ and step-wise learning rate decay every 100^{th} iteration. The loss function is the linear sum of the SSIM and the MAE.



(a) Training and validation loss for the Res-Udet model(s) for different number of bottleneck layers, with and without the BiFPN layer for 4x acceleration.

(b) Training and validation loss for the Res-Udet model with 8x acceleration.

Figure 4.2: Training and validation loss for the Res-Udet model (see figure 3.6) with 4x and 8x acceleration for k-space undersampling with the equidistant subsampling filter (see figure 3.1). The training was for a total of 250 iterations with an initial learning rate of $2 \cdot 10^{-3}$ and step-wise learning rate decay every 100th iteration. The loss function is the linear sum of the SSIM and the MAE.

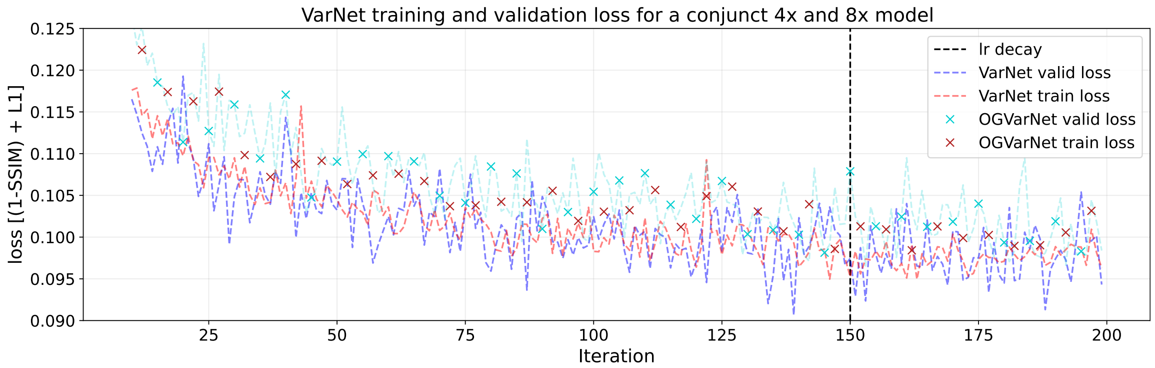
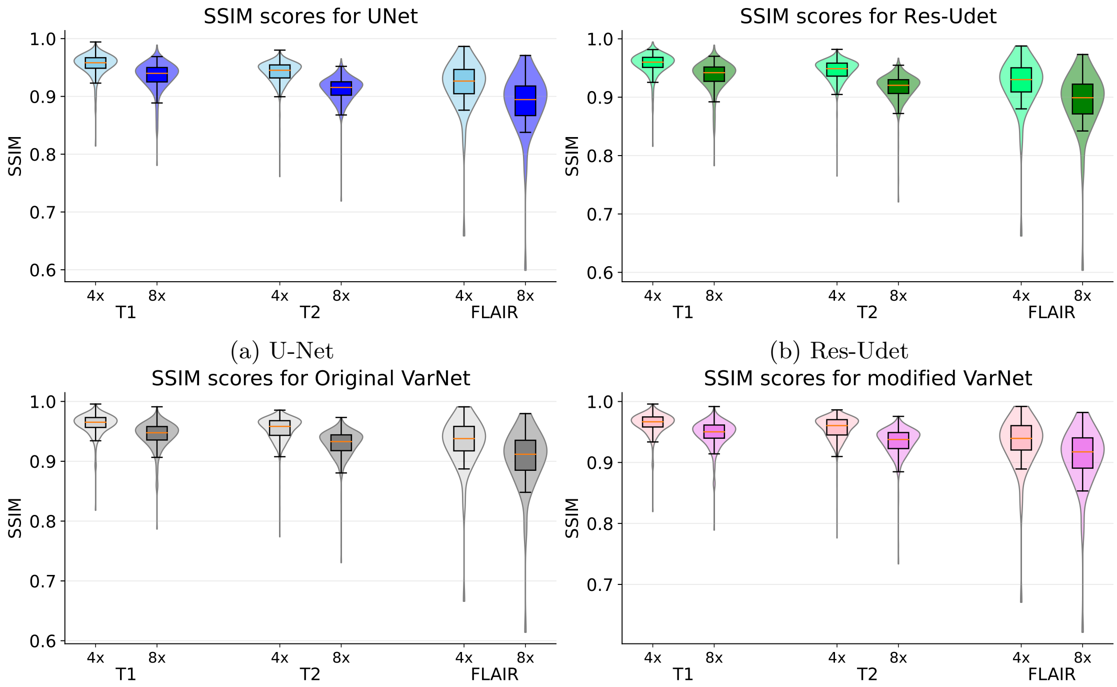


Figure 4.3: Training and validation loss for the original and modified VarNets for a conjunct 4x and 8x k-space undersampling during training. The training was for a total of 200 iterations with an initial learning rate of $2 \cdot 10^{-3}$ and step-wise learning rate decay every 150th iteration.



(a) U-Net

(b) Res-Udet

(c) Original VarNet

(d) Modified VarNet

Figure 4.4: The distribution of the SSIMs for the different sequence types for the 4x and 8x accelerated images from the test dataset using the U-Net, Res-Udet, original VarNet and the modifed VarNet models.

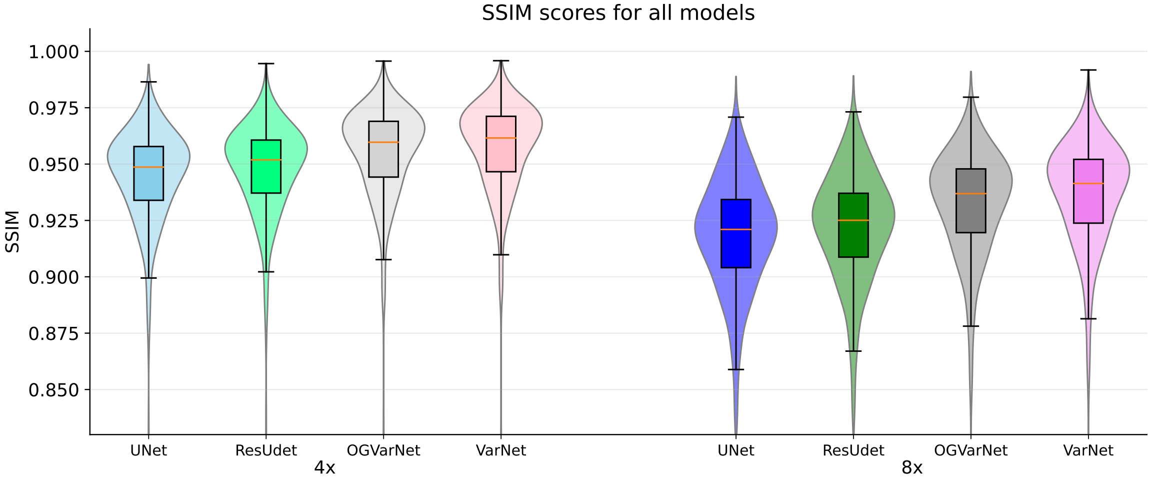


Figure 4.5: The SSIM distribution for the 4x and 8x reconstructed images from the entire test dataset for all four models across all three image modalities (T_1-weighted pre/post contrast, T_2-weighted and FLAIR-weighted).

Table 4.1: The SSIM, NMSE and PSNR scores for 4x and 8x accelerated reconstructed images on the test dataset using U-Net, Res-Udet, the original VarNet, the modified VarNet and sparse sampling. This includes the number of parameters and the mean and standard deviation of the inference time in milliseconds. The starred models * are all trained separately for 4x and 8x acceleration, whereas the non-starred models are trained conjunct on 4x and 8x acceleration simultaneously.

Models	Inference	Params.	Sequence	4x-acceleration			8x-acceleration		
	[ms]			SSIM	NMSE	PSNR	SSIM	NMSE	PSNR
U-Net*	60 ± 5	69.0M	T_1	0.955	0.0049	39.69	0.934	0.010	36.46
			T_2	0.941	0.0071	36.63	0.911	0.017	32.81
			FLAIR	0.920	0.0076	37.68	0.889	0.015	34.51
			Mean	0.944	0.0065	37.68	0.917	0.015	34.10
Res-Udet*	88 ± 2	11.9M	T_1	0.956	0.0046	40.03	0.936	0.0098	36.72
			T_2	0.945	0.0061	37.30	0.916	0.015	33.30
			FLAIR	0.924	0.0068	38.15	0.893	0.014	34.86
			Mean	0.947	0.0057	38.22	0.920	0.013	34.50
OGVarNet	150 ± 1	44.6M	T_1	0.962	0.0036	41.13	0.943	0.0079	37.60
			T_2	0.954	0.0045	38.76	0.928	0.011	34.56
			FLAIR	0.923	0.0059	38.88	0.903	0.012	35.72
			Mean	0.954	0.0044	39.51	0.931	0.010	35.61
VarNet	404 ± 1	45.7M	T_1	0.963	0.0032	41.65	0.946	0.0069	38.20
			T_2	0.956	0.0040	39.34	0.933	0.0098	35.30
			FLAIR	0.932	0.0054	39.34	0.908	0.010	36.32
			Mean	0.956	0.0039	40.06	0.935	0.0089	36.29
Sparse sampling	NA	NA	T_1	0.927	0.012	35.92	0.901	0.034	33.66
			T_2	0.890	0.025	31.25	0.862	0.033	30.47
			FLAIR	0.874	0.019	34.72	0.828	0.058	31.16
			Mean	0.900	0.021	33.00	0.870	0.036	31.52

Table 4.2: The p-values between the metric distributions of the test dataset for the different models on both 4x and 8x accelerations. The grey highlighted cells are all larger than the Bonferroni corrected statistical significance of $\frac{0.05}{6} \approx 8 \cdot 10^{-3}$.

Model 1	Model 2	4x acceleration			8x acceleration		
		SSIM	NMSE	PSNR	SSIM	NMSE	PSNR
U-Net	Res-Udet	$3.1 \cdot 10^{-2}$	$1.6 \cdot 10^{-10}$	$1.5 \cdot 10^{-5}$	$3.2 \cdot 10^{-2}$	$1.7 \cdot 10^{-7}$	$2.7 \cdot 10^{-3}$
U-Net	OGVarNet	$5.2 \cdot 10^{-14}$	$1.3 \cdot 10^{-64}$	$2.7 \cdot 10^{-44}$	$4.4 \cdot 10^{-17}$	$8.2 \cdot 10^{-65}$	$1.0 \cdot 10^{-28}$
U-Net	VarNet	$1.5 \cdot 10^{-19}$	$3.7 \cdot 10^{-96}$	$2.3 \cdot 10^{-69}$	$6.8 \cdot 10^{-28}$	$2.1 \cdot 10^{-112}$	$3.3 \cdot 10^{-56}$
Res-Udet	OGVarNet	$5.0 \cdot 10^{-8}$	$1.6 \cdot 10^{-31}$	$5.3 \cdot 10^{-24}$	$2.3 \cdot 10^{-10}$	$1.5 \cdot 10^{-36}$	$8.1 \cdot 10^{-17}$
Res-Udet	VarNet	$2.7 \cdot 10^{-12}$	$2.3 \cdot 10^{-57}$	$9.7 \cdot 10^{-45}$	$3.7 \cdot 10^{-19}$	$5.1 \cdot 10^{-79}$	$5.5 \cdot 10^{-40}$
OGVarNet	VarNet	$1.2 \cdot 10^{-1}$	$8.7 \cdot 10^{-6}$	$1.8 \cdot 10^{-5}$	$7.4 \cdot 10^{-3}$	$6.9 \cdot 10^{-13}$	$1.3 \cdot 10^{-7}$

Chapter 5

Discussion

You could not step twice into the same river.

— Heraclitus

The main aim of this chapter is to discuss the methods used and the results obtained for the different reconstruction approaches investigated. The results indicate a clear improvement from the sparse sampling methods to the deep learning approaches, and that an improved network architecture achieves improved reconstruction quality in the form of higher SSIM and lower NMSE.

5.1 Preprocessing

All models tested were subject to some preprocessing steps required for model compatibility. One limitation with respect to the preprocessing was the resolution enforcement of 320×320. This is a higher resolution than what is typical in clinical MRI, with common numbers of phase and frequency encoding steps for structural scans being in the range 192 to 256. However, the models trained in this thesis were never meant to be used in a clinical environment. Instead, the models were meant as a proof-of-concept, illustrating the reconstructive capabilities of deep learning. Clearly, any clinically accepted models should be trained on varying resolutions, ensuring a wide applicability and generalizability. Note that it is possible to simulate different resolutions with the fastMRI dataset by reducing the k_{max}-values in the phase and frequency encoding directions.

An important point of discussion is in regards to the method of coil combination. The RSS combined images exhibit a substantial amount of background noise [23], and the same background noise is treated as the ground truth. The background noise has a substantial influence on the SSIM of the reconstructed image. This effect is evident in figure 2.18d, and further emphasized by figure A.9 of the reconstructed T_2-weighted image with the lowest SSIM for all the reconstructed T_2-weighted images using the modified VarNet. In this figure, the reconstructed image is adequate in terms of pathology, however it does not reconstruct the noise profile. This has two consequences, first, the SSIM of some volume samples are inevitably biased towards lower metrics because of the noise in the ground truth images, even if the pathologies have been properly reconstructed. Second, since the reconstructions are penalized by the noise profile of the ground truths, the networks may learn or try to learn to replicate the noise profile when the effect of the noise in the loss is substantial when compared to the pathology. A possible solution to reduce the inherent image noise would be to apply other coil combination schemes, such as the adaptive combined reconstruction method proposed by Walsh, Gmitro, and Marcellin [87], a method that reduces noise in dark areas. However, an issue with this suggestion is the additional processing time because adaptive combined reconstruction is approximately ten times more time intensive than the RSS method [87]. Nonetheless, for magnitude reconstruction, the major time bottleneck is data transfer time, any additional computational complexity in the preprocessing step would only pose a small time penalty in comparison to data transfer time. However, for a model like the VarNet which performs the RSS coil combination method within the model itself (final step), the change in coil combination method may substantially affect both training time and model performance.

A possible solution to the noise sensitivity problem that would not impose any coil combination restrictions could be to apply some noise reduction or smoothing filter on the ground truth images, possibly at the cost of some loss in spatial resolution.

5.2 U-Net and Res-Udet

5.2.1 Training graphs

As evident by figures 4.1 and 4.2, neither the U-Net nor Res-Udet architectures exhibit any noticeable degree of overfitting, since their respective training and validation losses are not diverging. This is in agreement with previous findings for the U-Net architecture in regards to MRI reconstruction for the fastMRI dataset [22]. However, it cannot be guaranteed that a Res-Udet model with additional length or width scaling (i.e., increased number of parameters) would not overfit the data.

By studying figure 4.2a, the importance of depth scaling is evident, as the validation loss decreases with increasing depth. However, the possibility of depth saturation was not studied, although literature suggest that this may happen [14]. Furthermore, the training graphs indicate that the addition of the BiFPN layer provides no further accuracy improvements. In fact, the BiFPN-less Res-Udet model with $m = 4$ had a lower overall validation loss, suggesting a small accuracy penalty by the addition of the BiFPN layer. Nonetheless, as the validation slices are picked at random, this stochasticity may explain the differences in validation scores between the BiFPN and BiFPN-less models. Further studies are therefore needed to explore the possible advantages of using BiFPN layers in combination with the Res-Udet model architecture.

5.2.2 Reconstruction scores

From table 4.1 it can be concluded that the Res-Udet model exhibits improved accuracy altogether for both 4x and 8x acceleration when compared to the U-Net. From a numerical standpoint, this difference may seem small, but the SSIM listed underestimates the difference in the improved accuracy. This is because the maximum value for the SSIM is 1, any improvements close to this number will percentage-wise seem insignificant. Instead, it is more correct to compare NMSE-values when comparing numerical improvement. The Res-Udet model has a $\sim 12\%$ decrease in the NMSE when compared to the U-Net model for 4x acceleration ($\sim$ 15% for 8x acceleration). This implies that the architectural additions accounted for a numerical improvement of over 10%. Furthermore, if one were to compare the $1 - \text{SSIM}$ values, the Res-Udet model had a numerical improvement of $\sim$ 5%. These numerical improvements are all for a model with 12M parameters, this is less than 20% of the number of parameters in the trained U-Net model, but the effects of additional parameters in the Res-Udet model remains to be tested.

Although a difference of $\sim$ 5% and $\sim$ 10% seem small, findings by radiologists have suggested a close correlation between the SSIM, NMSE and PSNR and the radiologists evaluation of the pathology depicted [23]. As such, any increase in the SSIM may result in a better overall reconstructed image, potentially making deep learning accelerated MRI more viable in a clinical setting. This implies that future reconstruction schemes employing deep learning models have the potential for improved accuracy by using novel model designs from the deep learning community, achieving an appreciable reconstruction improvement with architectural enhancements alone. Nonetheless, this does not mean that the network developed during this thesis is the best approach. It only shows that for magnitude image reconstruction, employing depth scaling, residual connections, squeeze-and-excitation and the SiLU activation function improves upon the accuracy for the U-Net architecture.

A comparison between the SSIM of the deep learning reconstruction models and the sparse sampling method clearly indicate a significant improvement in reconstruction for the deep learning models over the sparse sampling approach. The Res-Udet and U-Net models have a mean SSIMs of 0.947 and 0.944, respectively, in comparison to a SSIM of 0.900 for the 4x accelerated sparse sampling. Not only are there a significant difference for the 4x accelerated SSIMs, the 8x accelerated deep learning models achieve better reconstruction quality than the 4x accelerated sparse sampling approach.

5.2.3 Image quality

5.2.3.1 Comparison to sparse sampling

From the reconstructed U-Net and Res-Udet images in figures A.1c and A.1d, we observe a clear image quality improvement in comparison to the sparse sampling reconstructed images in figure A.10. Furthermore, the 8x accelerated reconstructed images from U-Net and Res-Udet have improved image quality when compared to the 4x accelerated sparse sampling images. This statement holds true for all the reconstructed images for the different weighting schemes used in this thesis.

5.2.3.2 U-Net and Res-Udet

As seen from the Res-Udet and U-Net reconstructed images, it is evident that they are by no means perfect reconstructions of the ground truth, and exhibit reduced reconstruction quality for subtle- to minor-details in brain structures. This can for instance be seen from the red dashed ellipsoid in the 4x reconstructed FLAIR-weighted image in figure A.1, where some of the more subtle pathology is not reconstructed. The same can be seen from the large red ellipsoid in the 4x accelerated T_1-weighted post-contrast image in figure A.3, where the minor brain structures in the ellipsoid is smudged and not properly reconstructed. The lack of proper reconstruction of subtle- to minor-details in brain structure is further evident in the red dashed ellipsoid for the T_2-weighted image in figure A.7, where the U-Net and Res-Udet are incapable to completely reconstructing the white dots.

For the 8x accelerated images, the U-Net and Res-Udet models are incapable of reconstructing more minor- to moderate-details in brain structure. This is evident by the dashed circles and ellipsoids in the FLAIR-weighted reconstruction in figure A.2 and the T_1-weighted post-contrast reconstruction in figure A.4.

As discussed previously, both the U-Net and the Res-Udet may exhibit reduced reconstruction quality for subtle- to minor-details and minor- to moderate-details in brain structures for 4x and 8x acceleration, respectively. The Res-Udet model should by the improved SSIM, NMSE and PSNR values as seen from table 4.1 have improved reconstructive quality. This improvement is difficult to discern in a direct image comparison. However, it can be discerned by a close inspection. This difference is visible between the U-Net and Res-Udet in the blue dashed circles and ellipsoids in figures A.1, A.3 and A.8 among others. Nonetheless, for the untrained eye, it is difficult to discern any significant difference between the models. This suggest that a SSIM improvement of 0.003 ($\sim$ 5%) is barely visible with an untrained eye. Furthermore, a SSIM of 0.947 is too small too properly reconstruct all brain structures without any informational loss.

An interesting observation from the reconstructed brain images is that the networks act as smoothing filters, removing clearly visible noise and suppressing the Gibbs ringing artifact as evident from figure A.3. This suggests the possibility of training deep learning models for noise reduction and artifact correction. A future extension of thesis would be to test out the hypothesis of using deep learning for noise reduction, and possibly use the noise reduction network to combat the problem of background noise exhibited by some of the brain scans.

5.2.4 Comparison to literature

5.2.4.1 Differences in approach

The authors of the fastMRI dataset - Knoll et al. [28] - have much like previous authors of large public datasets been hosting a yearly competition, trying to push the gold standard for MRI reconstruction.[1] As the models from the fastMRI website are trained on the same dataset, the website will serve as a source of comparison. However, a direct comparison between metrics should be carried out with caution because the method of preprocessing differ slightly. In this thesis, any oversampling along the phase or frequency encoding directions are removed before downsampling. In the fastMRI approach, the data is downsampled before any oversampling along the phase and frequency directions are handled. Another possible difference is the test dataset used in this thesis, as it was originally part of the validation set before the initial split. Nonetheless, the final results should be comparable by default, as the results from this thesis and other research are based upon 4x or 8x acceleration.

5.2.4.2 Metric comparison

The achieved SSIMs of 0.947 and 0.920 for the Res-Udet model are inferior to the models with the best reconstruction metrics on the fastMRI dataset [23, 68]. One of the best performing deep

[1] `https://fastmri.org/`

learning accelerated models on the fastMRI dataset have SSIMs of 0.9640 and 0.952 for 4x and 8x acceleration, respectively. However, it should be stated that this method was trained separately for 4x and 8x acceleration on four Nvidia V100 GPU's, and has $\sim$ 200M parameters [88]. Although the magnitude reconstruction has shown improved results, the architectural improvements alone for the U-Net model was incapable of closing the gap between the simple magnitude input - magnitude output reconstruction and more complex reconstructive methods with greater GPU requirements.

The architectural improvements alone for the magnitude input - magnitude output reconstruction scheme was incapable of competing with the more complex methods of reconstruction. Nonetheless, Ding et al. [89] achieved a SSIM of 0.9533[2] for 4x acceleration with their residual U-Net model, a model almost identical to the Res-Udet model depicted in figure 3.6 with $m = 0$ and no BiFPN layer. Their approach used a different loss function and changed the model prediction to a correctional map applied by addition on the input, instead of predicting the ground truth. This suggests that slight modifications to the Res-Udet model prediction and an altered loss function might improve the reconstructions. The validation loss from figure 4.2a suggests that the addition of bottleneck layers and depth scaling could further improve upon their results.

5.3 Variational network

5.3.1 Reconstruction scores

By a strict comparison of metrics, the modified VarNet outperforms the original version at a third decimal level, much like Res-Udet and U-Net. The total SSIM difference between the models were 0.002 and 0.004 for 4x and 8x accelerations, respectively, and both models have approximately the same number of trainable parameters. The accuracy improvements resulted in a $\sim$ 10% improvement in the NMSE and a $\sim$ 5% improvement in the SSIM for both 4x and 8x acceleration. As such, the inclusion of the basicblock shows great promise as it results in an undeniable improvement in the reconstruction metrics. The same improvement is evident from the training and validation loss seen in figure 4.3, where the modified version converges to a lower overall minimum. This suggests that the previous U-Net architecture in the VarNet had some problems with vanishing gradients and un-optimized parameters. The interchange of deep learning architecture from U-Net to Res-Udet results in a substantial performance increase without any additional modifications to the network. This suggest that high-end reconstruction frameworks can gain an additional increase in accuracy by adopting research from the deep learning community, and by this achieve improved image quality and reconstruction capabilities without any additional framework modifications.

5.3.2 Image quality

5.3.2.1 Comparison with U-Net and Res-Udet

The improvement in image quality from U-Net and Res-Udet to the VarNets is apparent for each and every reconstructed image. A couple of examples are highlighted by the green circles/ellipsoids in figures A.3, A.5 and A.7 for 4x acceleration. In these figures, the VarNets are capable of reconstructing more minor details in brain structure in comparison the U-Net and Res-Udet.

[2] `https://fastmri.org/leaderboards/challenge/`

The difference in reconstruction quality between the VarNets and U-Net/Res-Udet is particularly noticeable for the 8x accelerated images, where the reconstructed images from the VarNets are close to the 4x accelerated reconstructed U-Net images. The accuracy improvement is particularly noticeable in the highlighted green circles/ellipsoids in figures A.2 and A.8, where the VarNets are capable of reconstructing more moderate details in brain structure in comparison to the U-Net and Res-Udet models. This suggests that a SSIM difference of ~ 0.01 is clearly noticeable for the untrained eye, whereas SSIM differences around 0.003 are difficult to discern.

5.3.2.2 VarNet and modified VarNet

It can be seen from the 4x reconstructed images that the VarNets exhibits reduced reconstruction capabilities for more subtle details in brain structure. This is evident from the red highlighted circles/ellipsoids in figures A.3 and A.5. This suggests that further reconstructive improvements are in order before 4x deep learning accelerated MRI is clinically viable. Furthermore, for the 8x accelerated reconstructions, the VarNets reconstructions exhibit reduced reconstruction quality for minor details in brain structure. This reduction in reconstruction quality for minor details in the 8x accelerated images are highlighted by the red circles/ellipsoids in figure A.2 and A.4.

Many of the points from the previous discussion regarding the visual improvement between the U-Net and Res-Udet models still hold. The numerical difference between the original VarNet and the architecturally modified version are somewhat minor (all things considered); because of this, it is difficult for the untrained eye to properly distinguish the two reconstructions. Nonetheless, there are minor differences between the two models in which one can see the slight improvement in metrics. Some of these minor differences are highlighted by the blue circles/ellipsoids in figures A.1, A.3 and A.6 for the two VarNets.

5.3.2.3 SSIM distributions

From the SSIM distribution graphs for the different sequences in figures 4.4a, 4.4b, 4.4c and 4.4d, it can be seen that the SSIMs are dependent on the sequence type. This is also evident from the different reconstructed images, with the T_1-weighted and T_2-weighted images being closer to the ground truth. There may be many reasons for this, but the most probable is the skewed distribution of the dataset, with less than 10% of the training dataset being FLAIR-weighted images as seen from table 3.1. As such, for clinical use, more FLAIR-weighted images should be available for training; otherwise, active measures must be taken to ensure a less skewed distribution. One such measure could be to vary the learning rate depending on the type of weighting used during training, e.g., the FLAIR-weighted images could have a learning rate of approximately ten times that of the T_2-weighted images (see table 3.1).

Lastly, as evident by the violin graph in figure 4.5, each and every improved model shift the mean slightly upwards whilst narrowing the distribution, ensuring a less varied reconstruction result. However, as the distribution is rather wide, the final reconstructed images may vary in quality. An important future milestone should therefore be to narrow down the distribution, ensuring consistent reconstruction quality. It should be said that this skewness may partly be caused by noise in the dataset as discussed previously and seen in figure A.9.

5.3.3 Comparison to literature

The original VarNet re-trained in this thesis has inferior reconstruction metrics in comparison to what is listed in literature. The VarNet in this thesis achieves SSIMs of 0.954 and 0.931 for 4x and 8x acceleration, respectively, whereas the VarNet in literature achieves SSIMs of 0.959 and 0.945 for 4x and 8x acceleration, respectively, for a non-conjunct 4x and 8x training [68]. This difference in metrics is to be expected, and may be justified by hardware restrictions, since the batch size is shown to affect the final performance of deep learning models [75]. Another reason for this SSIM difference may stem from the reduced training time in this thesis, with the model in literature trained for approximately five times longer. Nonetheless, the modified VarNet achieves SSIMs of 0.956 and 0.935 for 4x and 8x acceleration when trained conjunct on the two acceleration factors, closing in on the metrical difference by architectural improvements alone.

Besides the hardware restrictions, there is one more major difference that may affect the difference in metrics between this thesis and what is reported in literature. The loss function implemented in this thesis was the Gaussian weighted SSIM, whereas the VarNet in literature is trained on a non-Gaussian weighted SSIM [68]. However, research seem to suggest that the SSIM function is sub-optimal, and that the multiscale SSIM (MS-SSIM) [90] has the potential to achieve improved reconstruction in comparison to the SSIM [91].

5.3.4 Inference time

Unlike the U-Net and Res-Udet models, the inference time for the VarNets are somewhat substantial. The modified VarNet uses about 404 ms to completely reconstruct a twenty coils image on an RTX 2080 Ti GPU, with this reconstruction time a complete reconstruction of a volume with 200 slices would take approximately one and a half minute. Although not substantial when compared to a 15-minute scan, it raises the question whether there is a trade-off between reconstruction quality and inference time in the future for the potentially more complex models. Furthermore, if frequency oversampled data was used in the model inference, the inference time would inevitably increase, as the size of the input image increases.

5.4 Statistical Significance

The results summarized in table 4.2 imply that most of the differences measured between the model metrics are statistically significant, except the SSIM for 4x and 8x acceleration between the U-Net - Res-Udet and the SSIM for 4x acceleration between OGVarNet - VarNet. This suggests that the improved metrics are due to the differences between the models. Despite this, the possibility that the metrical improvements is due to stochasticity during parameter initialization or fetching of training samples cannot be excluded. In order to confirm that the metrical improvements are due to the architectural recommendations from the deep learning community, a method like the k-fold cross validation should have been used, and each model should have been re-trained multiple times, however this was not done because of time constraints. Nonetheless, metrical improvements of $\sim$ 10% or above for the NMSE and $\sim$ 5% for the SSIM in conjunction with the statistically significant differences in the distributions highly suggests that the improved metrics are because of the architectural additions.

It should be noted that the only difference when training the models was with regards to the model architecture, and all hyperparameters were fixed except for the reduced batch size and training time when training the VarNets. As such, the only difference that could affect model performance was the randomly initialized parameters and the stochasticity during training.

5.5 Clinical use

5.5.1 Clinical relevance

Faster MRI acquisition is important in order to decrease the chance of motion artifacts and to reduce imaging costs, but also to widen the applicability of MRI. Today, most MRI vendors market their scanners with the possibility of decreasing scan time with up to 50%. However, as seen in this thesis, the potential for larger acceleration factors using deep learning is a viable option, offering excellent image reconstruction for the 4x accelerated scans. This could potentially reduce scan time by an additional 50% over the sparse sampling methods used on modern scanners, even offering the unexpected surprise of noise suppression. It should be noted that from the results of this thesis, 8x accelerated MRI scans using deep learning is not clinically feasible. However, the 4x deep learning accelerated models are close to reconstructing adequate images.

A decrease in imaging time by an additional 50% could make MRI a feasible imaging modality for patient groups where extended scanning time is demanding. This can range from patients groups with claustrophobia, epilepsy, chronic pain and pediatric imaging where sedatives may be necessary to counteract patient movement and ensure viable images.

Deep learning accelerated MRI can potentially introduce acceleration factors not yet seen in any clinical setting with minimal penalty to image quality. Even though the 4x reconstructions were not perfect, this acceleration factor is still higher than clinical acceleration. A possible extension of this thesis would therefore be to test whether deep learning accelerated scans could outperform sparse sampling at clinical viable acceleration factors. Nonetheless, the deep learning accelerated scans have the potential to decrease scan time to new lows without any major decrease in image quality.

5.5.2 Clinical testing

A major issue with deep learning accelerated MRI is the need for large datasets from a variety of scanners at different field strengths and from multiple vendors. This is to ensure that the model is capable of adapting to differences in scans between different scanners as well as different pathologies. In addition, the differences in noise characteristic caused by the number of receiver coils and the method of coil combination may also affect the reconstruction of the model. Deep learning accelerated MRI is therefore not as simple as to employ the fastMRI trained models for clinical use, since small variations in the scan to be reconstructed may potentially break the models if the models are not trained to handle those differences.

In addition to the requirement of large and varied training data from different scanners, is the need for large scale clinical testing. However, this is not only expensive, but difficult to achieve. The software must be installed on a computer close to the scanner, radiologists must verity the image quality and the technique must be used on large patient cohorts. Furthermore, the patient group must be diverse to test the generalizability of the model, and by this ensure that no pathology is incorrectly reconstructed.

5.5.3 Potential problems

As mentioned earlier, the model must be trained on a heterogeneous data to ensure generalizability. If this not the case, it is possible for the reconstruction model to break down when exposed to "unexpected" data and not give a valid reconstruction.

Such a problem can be caused by a bias within the dataset, for example, lack of training data containing a given type of pathology. The model could then potentially learn to "cover-up" potential areas of interest since the model has not seen such pathology during training. This just emphasizes the need to train any model on sufficiently large and varied data, combined with extensive clinical testing. It should be noted that in this thesis, no measures were taken to ensure a non-biased training data, nonetheless the models were capable of reconstructing varied pathology (see figures A.1 and A.3).

Chapter 6

Conclusions

$S = k \log W$

— Inscription of Boltzmann's tombstone

Deep learning-accelerated MRI shows great promise with respect to the image reconstruction quality, and the baseline U-Net model outperformed the traditional sparse sampling method, both in terms of metrics, but also from a visual standpoint. Nonetheless, the 4x accelerated images were incapable of reconstructing more subtle- to minor-details in brain structure, whereas the 8x accelerated scans were only capable of reconstructing more moderate brain structures.

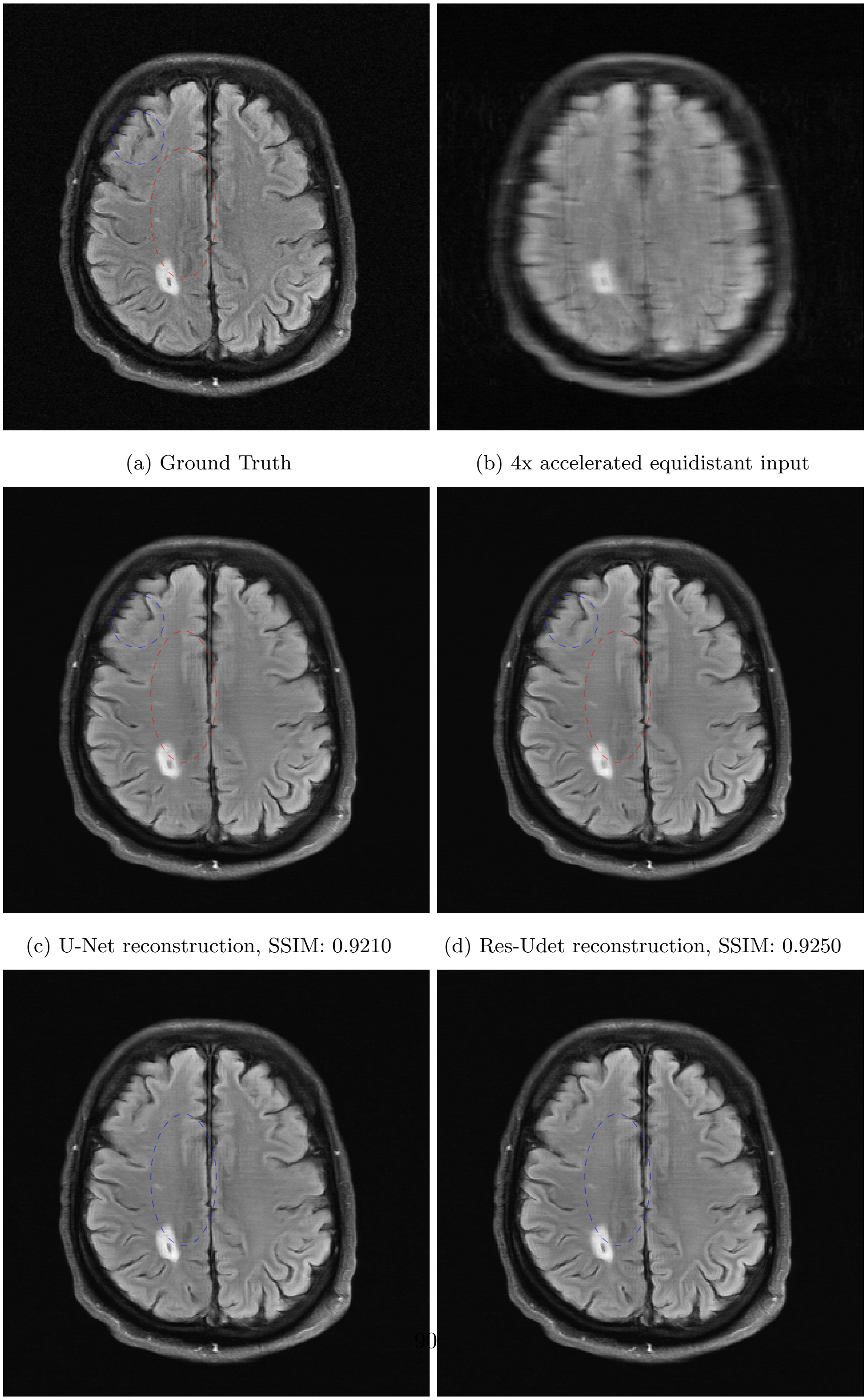
The architecturally improved U-Net model: Res-Udet, outperformed the baseline model, achieving a metrical improvement of $\sim 10\%$ and $\sim 5\%$ for the structural similarity index measure (SSIM) and normalized mean squared error (NMSE), respectively. However, the 4x reconstructed images were incapable of reconstructing subtle- to minor-details in brain structure, but a visual improvement from the baseline model could be discerned. This improvement indicates the importance of depth scaling, with decreased loss for increasing depth, agreeing with the assumption that an improved deep learning model improves the reconstruction quality. Furthermore, the findings suggest that the addition of the BiFPN architecture had no impact on the accuracy of the reconstruction, possibly imposing a small reconstruction loss.

The step from the U-Net and Res-Udet models to the VarNets netted a significant improvement in image reconstruction quality, this allowed for the reconstruction of minor details in brain structure for 4x accelerated brain scans, and minor- to moderate-details in brain structure for 8x acceleration. The interchange of deep learning models from U-Net to Res-Udet in the VarNet further improved the reconstruction metrics, achieving a $\sim 5\%$ and $\sim 10\%$ increase in the SSIM and NMSE, respectively, for 4x and 8x acceleration. The accuracy improvements from the Res-Udet model also suggests that the current state-of-the-art in MRI reconstruction could gain improved accuracy by following the recommendations of the deep learning community. The downside with an improved deep learning model is the increased complexity, and an increased inference time. The inference time for the modified VarNet increased from 150 ms to 404 ms with the same number of parameters, suggesting a possible trade-off between reconstruction time and reconstruction quality for more architecturally complex models.

Deep learning accelerated MRI has the potential to offer acceleration factors not yet used in a clinical setting, further decreasing scan time by additional 50% compared to today's gold standard sparse sampling. Although we may foresee a future where all MRI sequences can be acquired in 2 minutes or less, further improvements in deep learning architecture, model framework, large scale clinical testing and hardware resources are needed for this goal to become a clinical reality. In conclusion, the results of my Master's thesis as presented here, leaves me optimistic that the future of deep learning based MR image reconstruction is bright and will have a major clinical impact in the years to come.

Appendix A

Reconstructed images



(a) Ground Truth

(b) 4x accelerated equidistant input

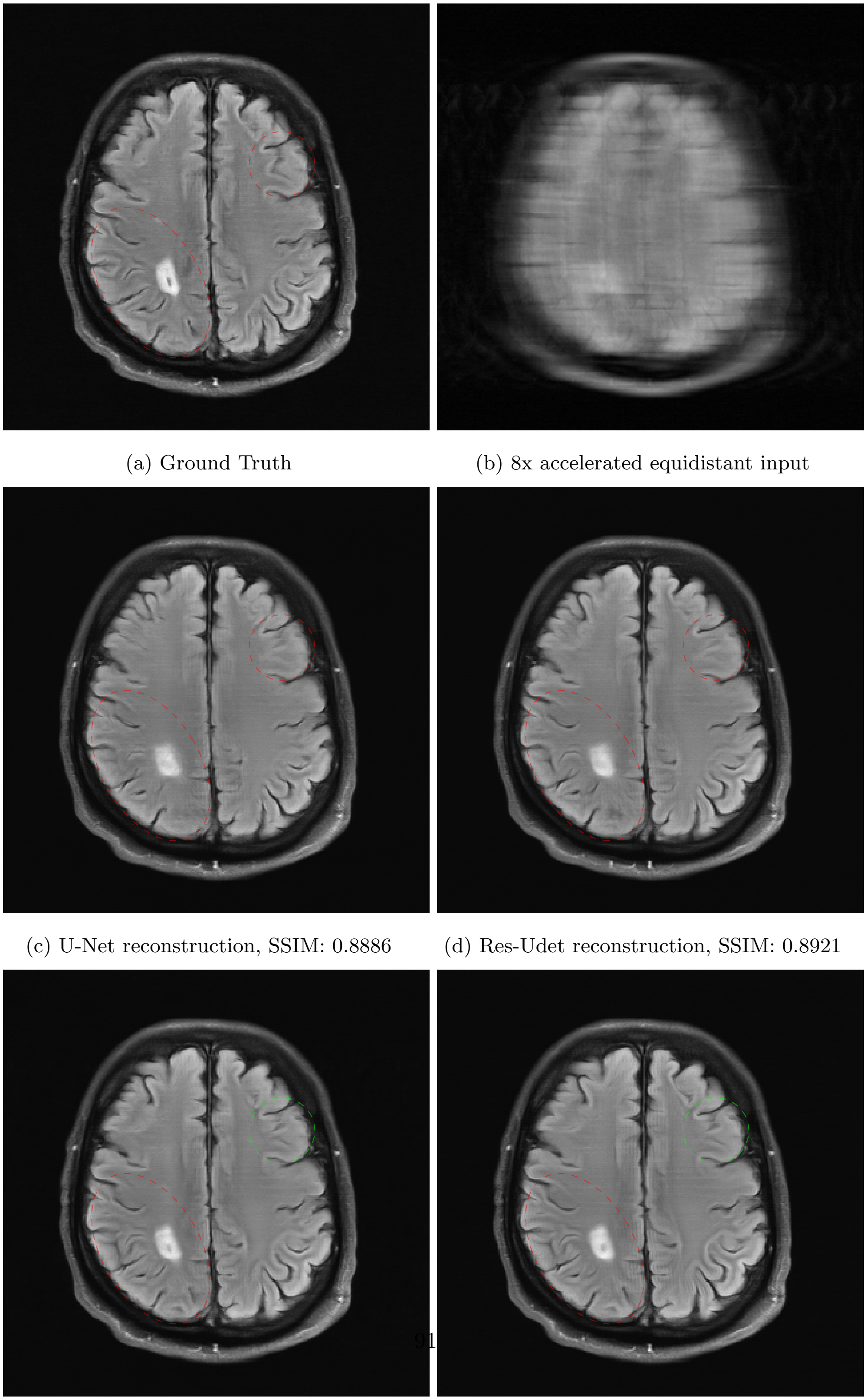
(c) U-Net reconstruction, SSIM: 0.9210

(d) Res-Udet reconstruction, SSIM: 0.9250

(e) Original VarNet reconstruction, SSIM: 0.9351

(f) Modified VarNet reconstruction, SSIM: 0.9381

Figure A.1: A reconstructed 4x accelerated FLAIR-weighted image from the fastMRI dataset [28], with SSIM close to that of the mean SSIM from table 4.1 for the different models.



(a) Ground Truth

(b) 8x accelerated equidistant input

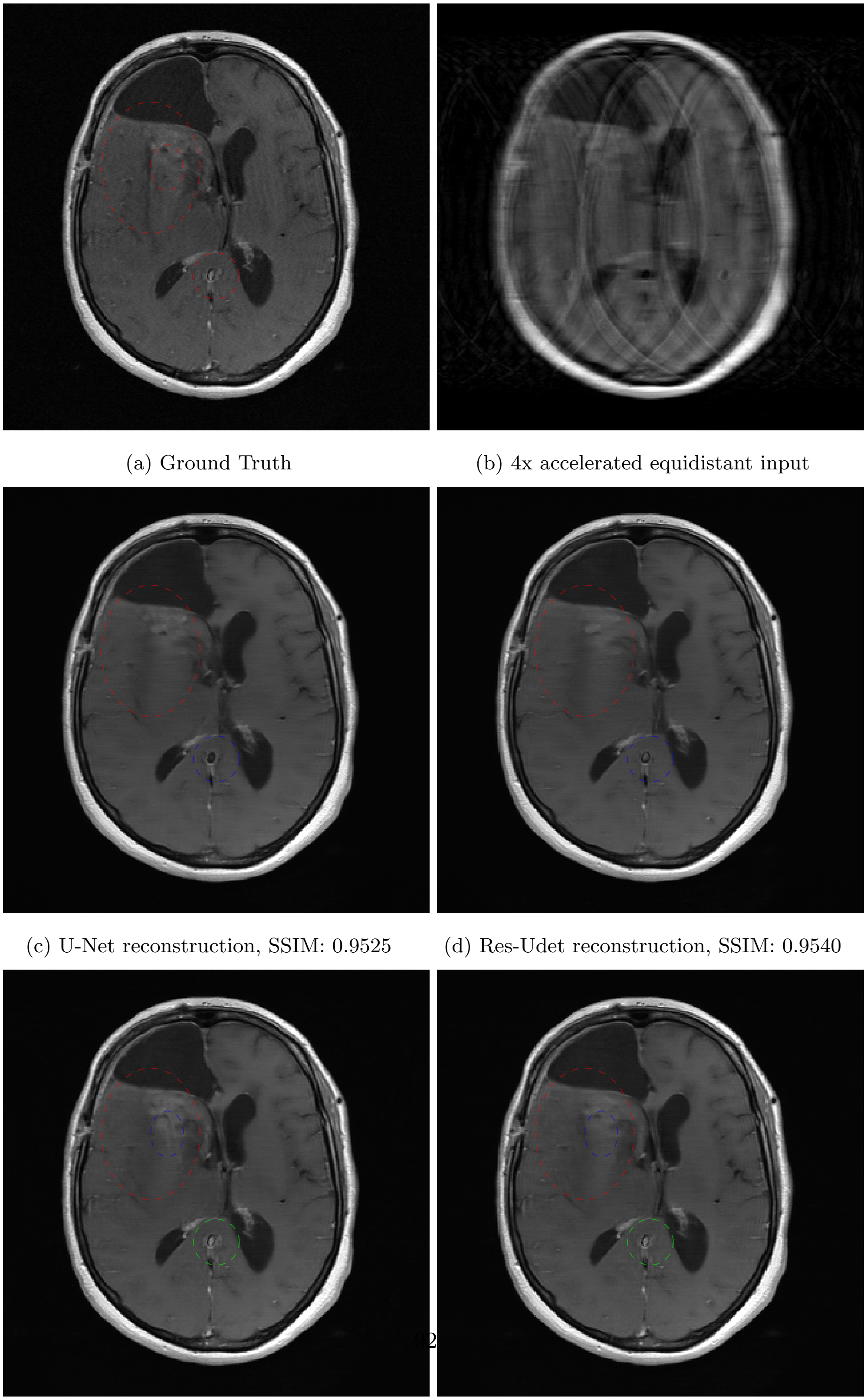
(c) U-Net reconstruction, SSIM: 0.8886

(d) Res-Udet reconstruction, SSIM: 0.8921

(e) Original VarNet reconstruction, SSIM: 0.9071

(f) Modified VarNet reconstruction, SSIM: 0.9125

Figure A.2: A reconstructed 8x accelerated FLAIR-weighted image from the fastMRI dataset [28], with SSIM close to that of the mean SSIM from table 4.1 for the different models.



(a) Ground Truth

(b) 4x accelerated equidistant input

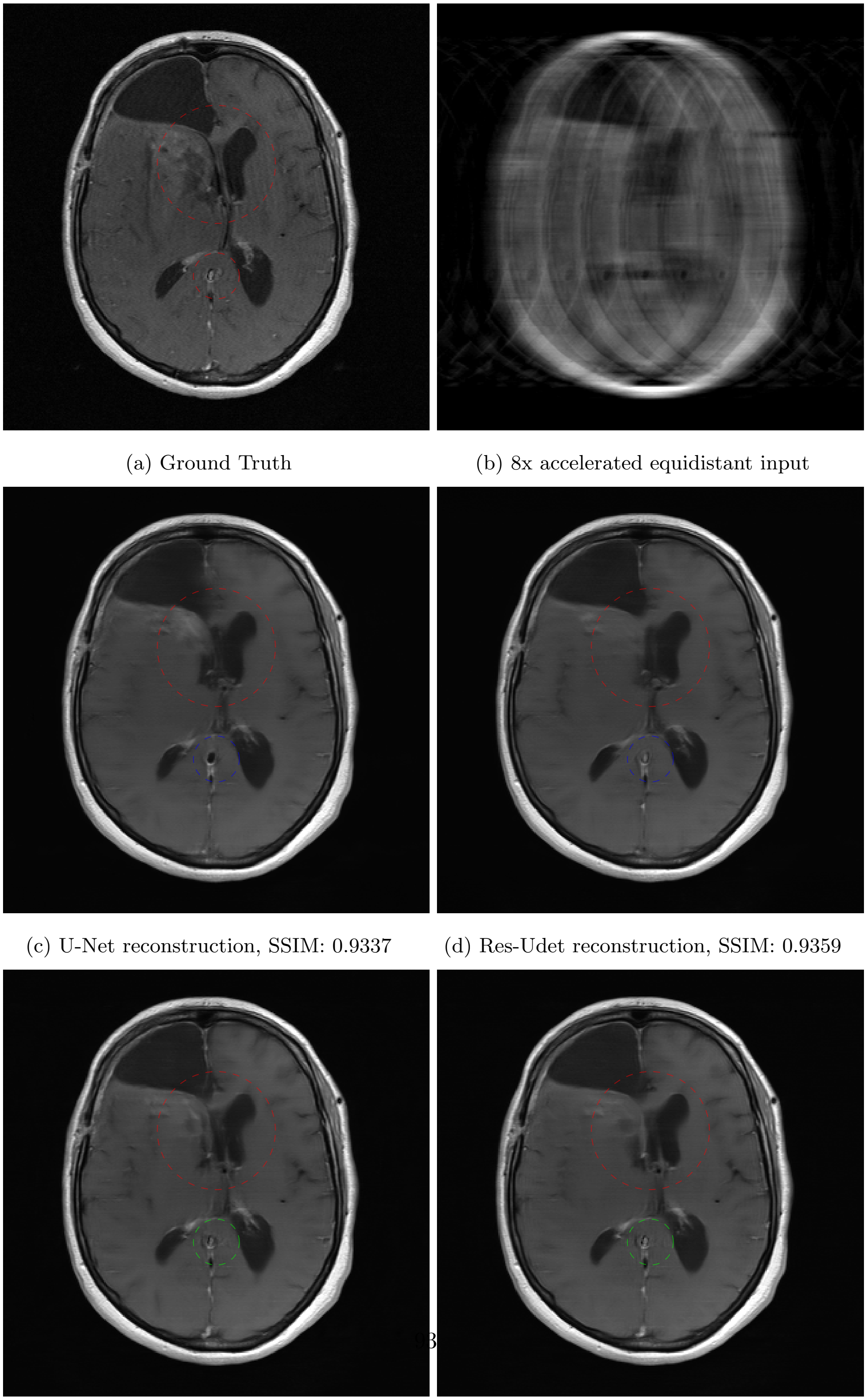
(c) U-Net reconstruction, SSIM: 0.9525

(d) Res-Udet reconstruction, SSIM: 0.9540

(e) Original VarNet reconstruction, SSIM: 0.9597

(f) Modified VarNet reconstruction, SSIM: 0.9615

Figure A.3: A reconstructed 4x accelerated T_1-weighted post-contrast image from the fastMRI dataset [28], with SSIM close to that of the mean SSIM from table 4.1 for the different models.



(a) Ground Truth

(b) 8x accelerated equidistant input

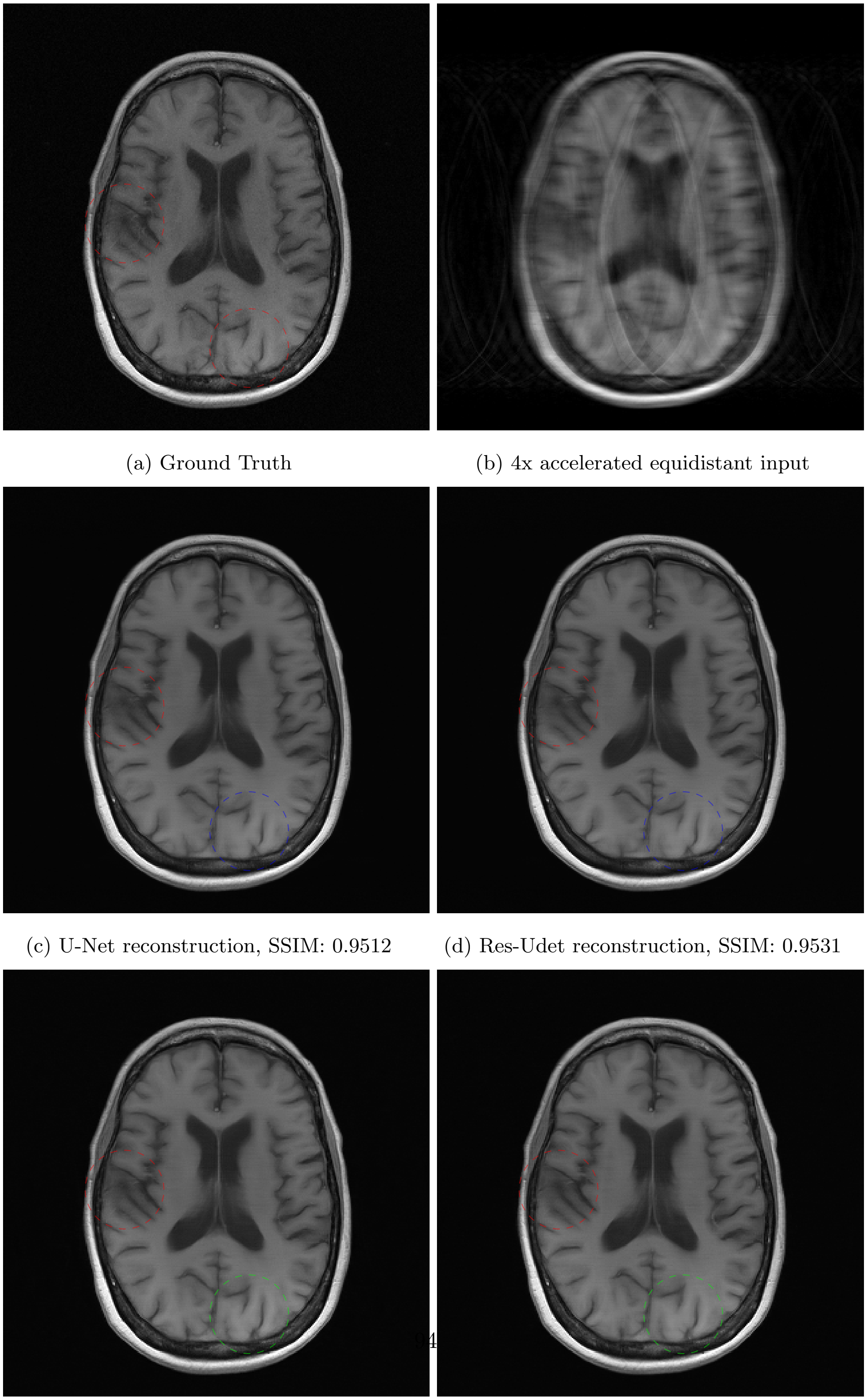
(c) U-Net reconstruction, SSIM: 0.9337

(d) Res-Udet reconstruction, SSIM: 0.9359

(e) Original VarNet reconstruction, SSIM: 0.9430

(f) Modified VarNet reconstruction, SSIM: 0.9471

Figure A.4: A reconstructed 8x accelerated T_1-weighted post-contrast image from the fastMRI dataset [28], with SSIM close to that of the mean SSIM from table 4.1 for the different models.



(a) Ground Truth

(b) 4x accelerated equidistant input

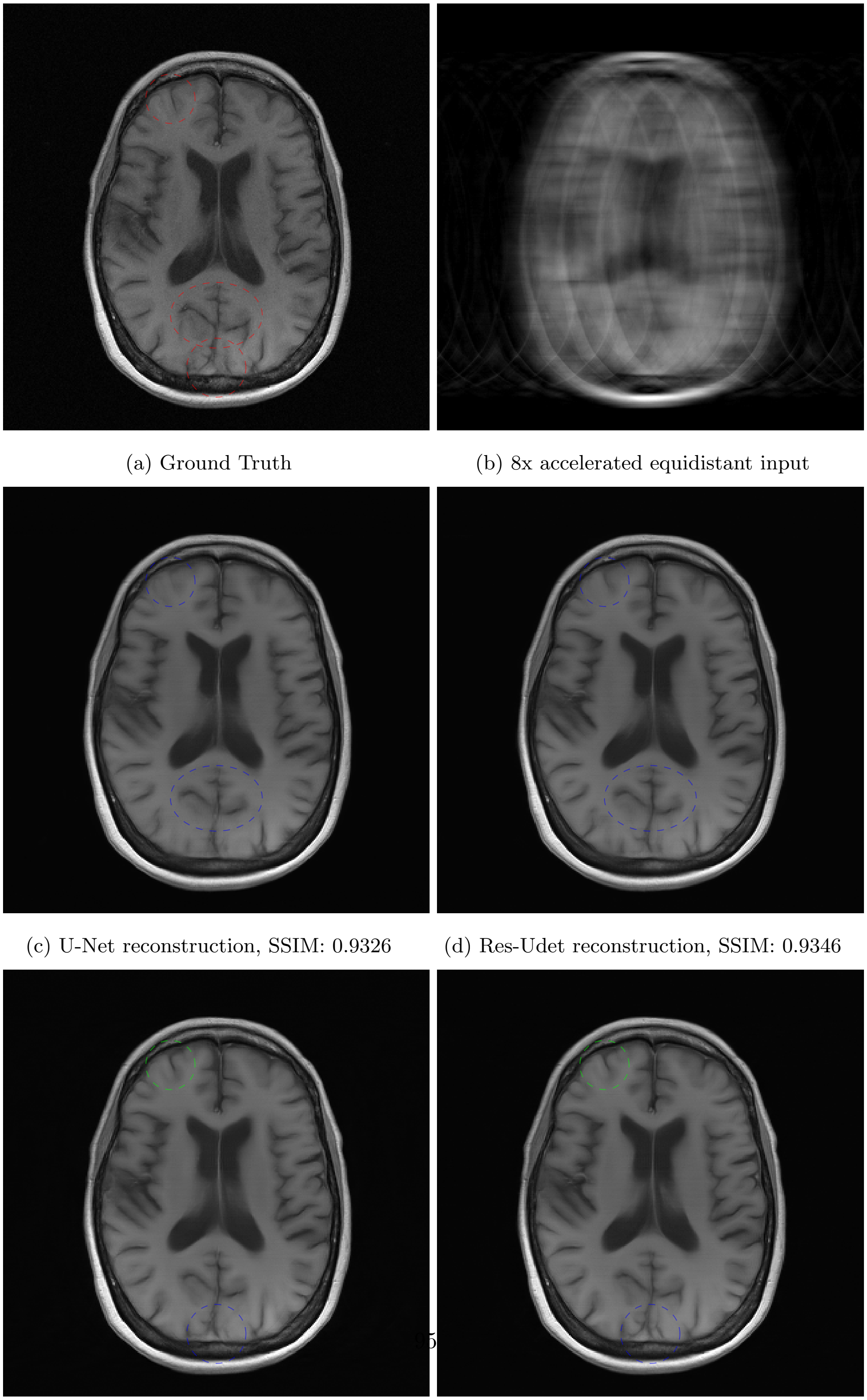
(c) U-Net reconstruction, SSIM: 0.9512

(d) Res-Udet reconstruction, SSIM: 0.9531

(e) Original VarNet reconstruction, SSIM: 0.9581

(f) Modified VarNet reconstruction, SSIM: 0.9595

Figure A.5: A reconstructed 4x accelerated T_1-weighted pre-contrast image from the fastMRI dataset [28], with SSIM close to that of the mean SSIM from table 4.1 for the different models.



(a) Ground Truth

(b) 8x accelerated equidistant input

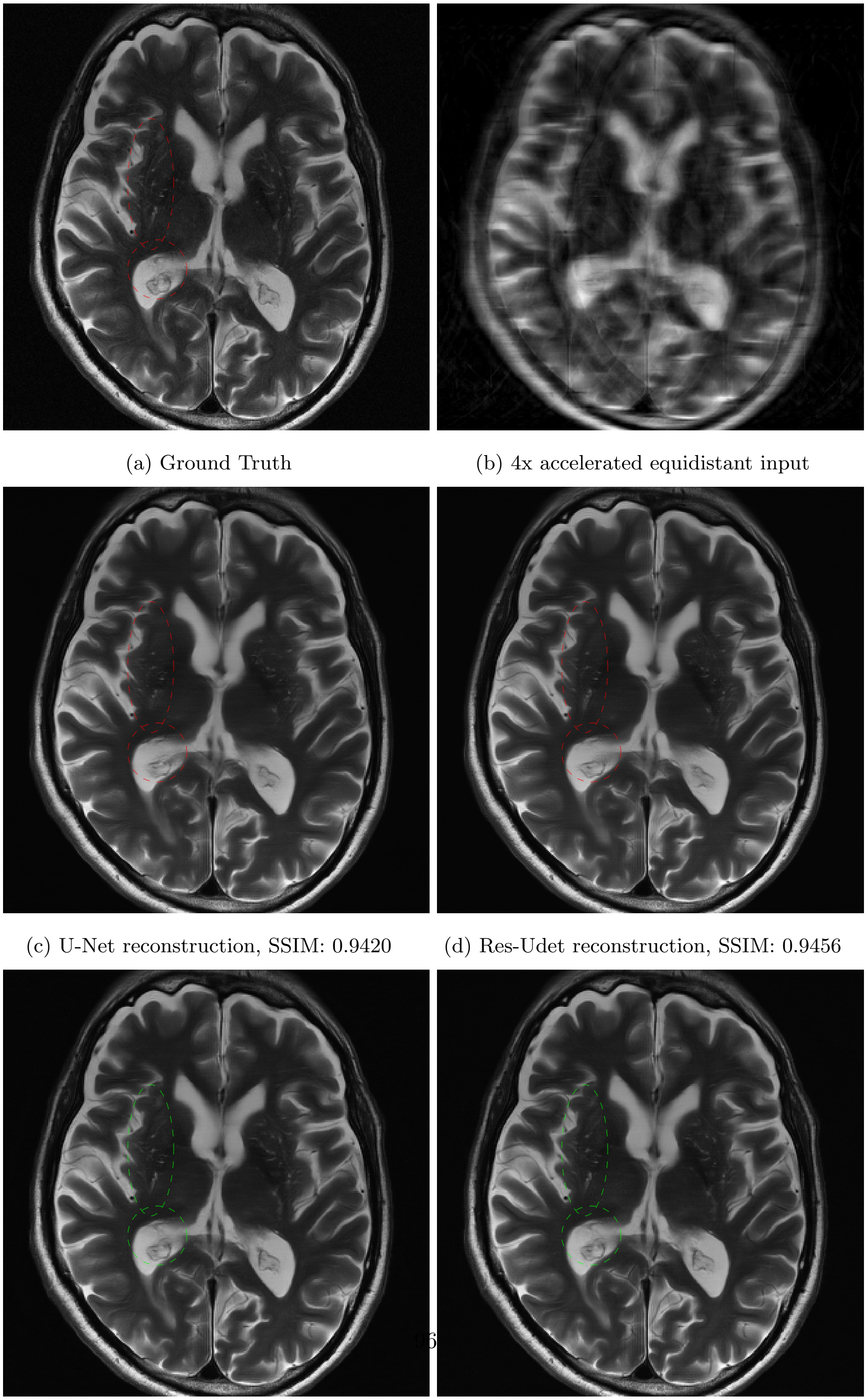
(c) U-Net reconstruction, SSIM: 0.9326

(d) Res-Udet reconstruction, SSIM: 0.9346

(e) Original VarNet reconstruction, SSIM: 0.9413

(f) Modified VarNet reconstruction, SSIM: 0.9442

Figure A.6: A reconstructed 8x accelerated T_1-weighted pre-contrast image from the fastMRI dataset [28], with SSIM close to that of the mean SSIM from table 4.1 for the different models.



(a) Ground Truth

(b) 4x accelerated equidistant input

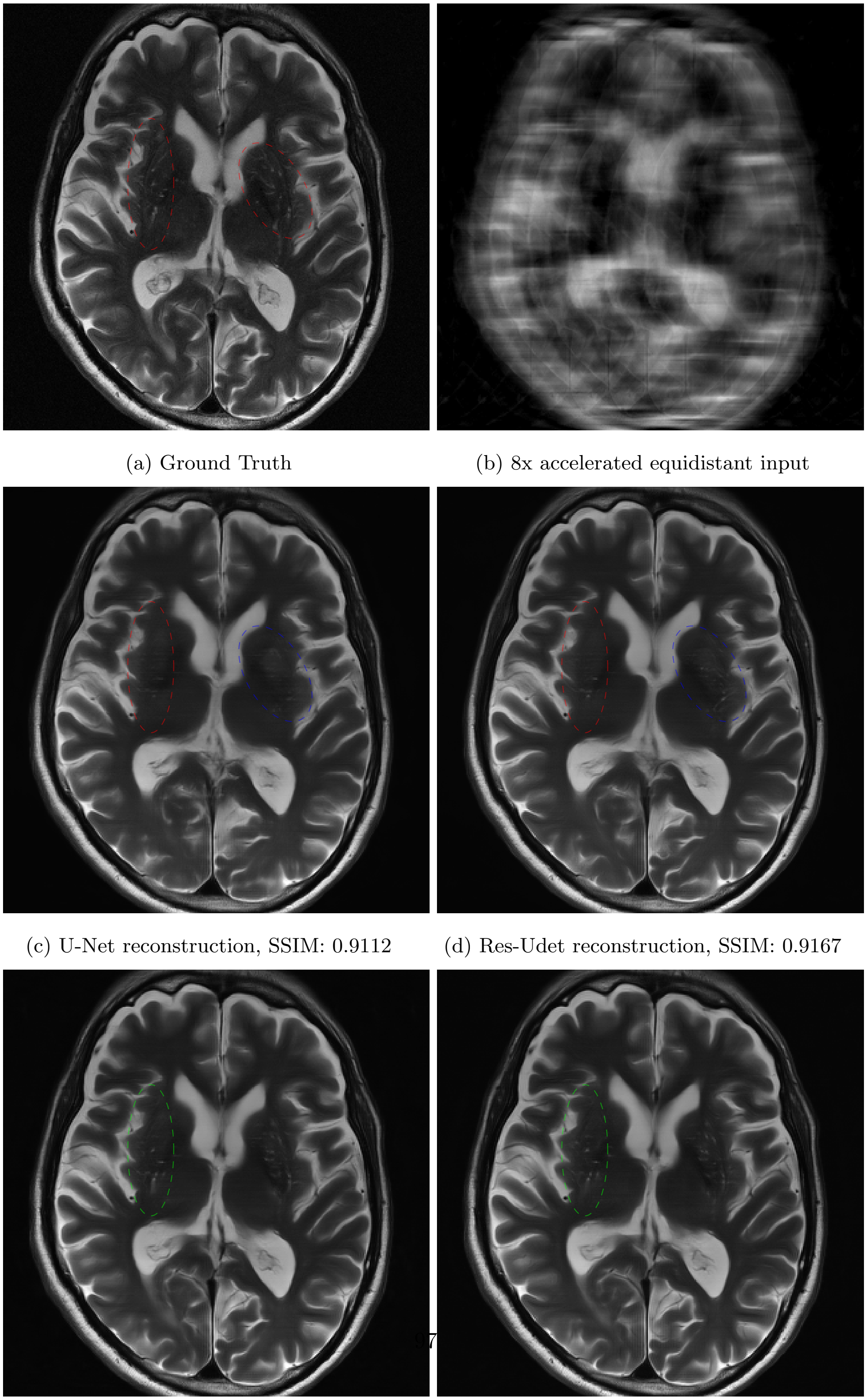
(c) U-Net reconstruction, SSIM: 0.9420

(d) Res-Udet reconstruction, SSIM: 0.9456

(e) Original VarNet reconstruction, SSIM: 0.9553

(f) Modified VarNet reconstruction, SSIM: 0.9571

Figure A.7: A reconstructed 4x accelerated T_2-weighted image from the fastMRI dataset [28], with SSIM close to that of the mean SSIM from table 4.1 for the different models.



(a) Ground Truth

(b) 8x accelerated equidistant input

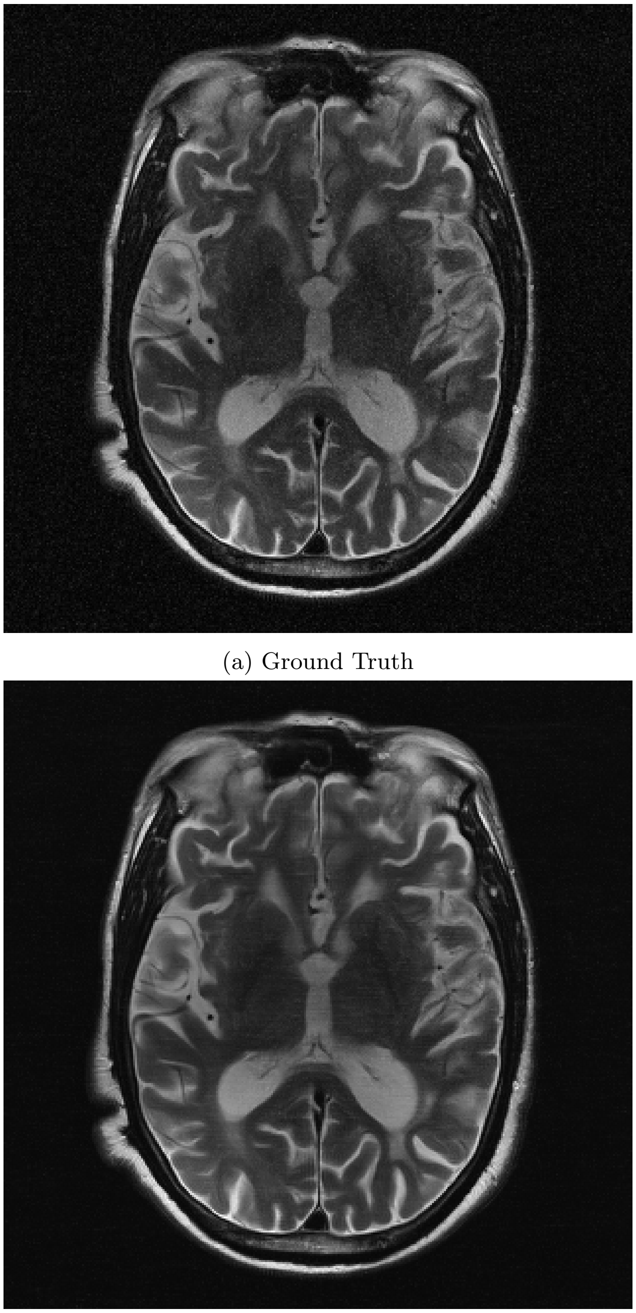
(c) U-Net reconstruction, SSIM: 0.9112

(d) Res-Udet reconstruction, SSIM: 0.9167

(e) Original VarNet reconstruction, SSIM: 0.9322

(f) Modified VarNet reconstruction, SSIM: 0.9367

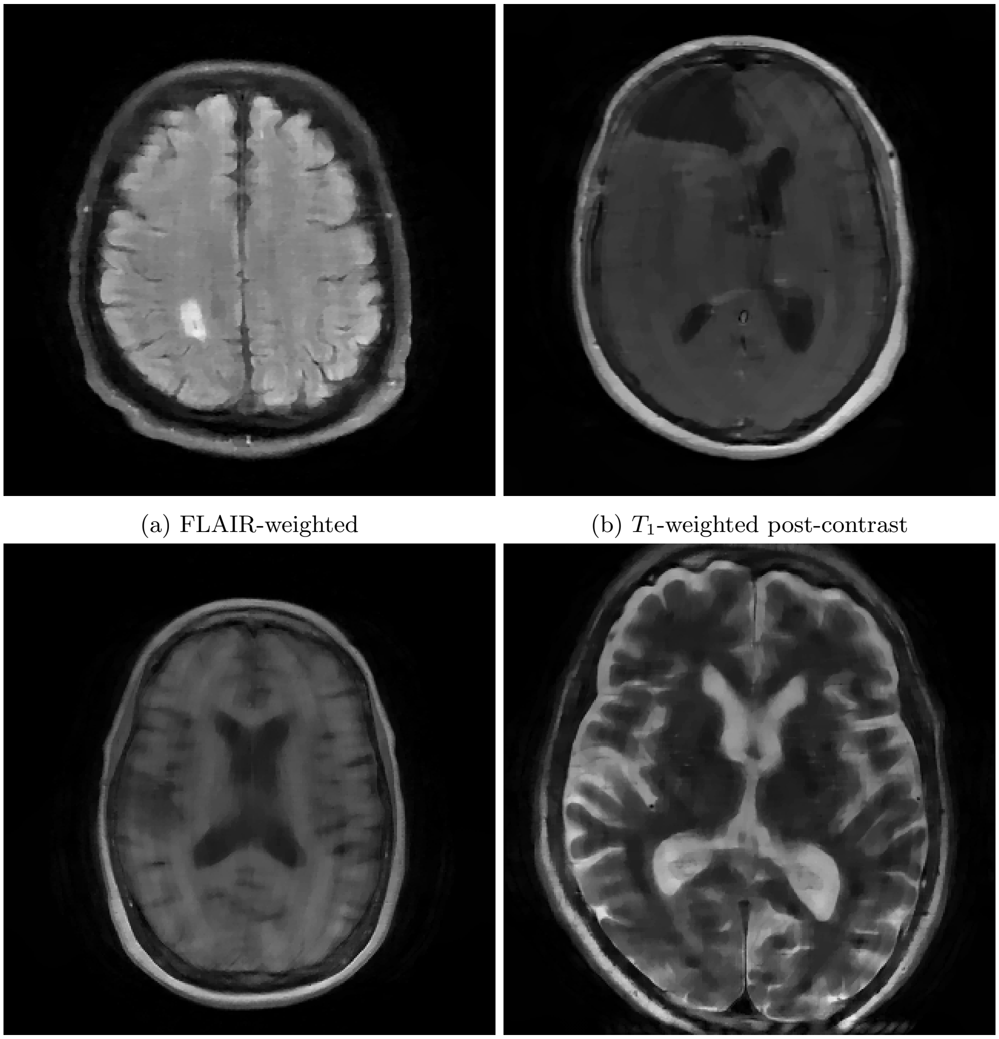
Figure A.8: A reconstructed 8x accelerated T_2-weighted image from the fastMRI dataset [28], with SSIM close to that of the mean SSIM from table 4.1 for the different models.



(a) Ground Truth

(b) Modified VarNet reconstruction, SSIM: 0.776

Figure A.9: The reconstructed and ground truth image of the T_2-weighted reconstruction with the worst SSIM for 4x acceleration using the modified VarNet.



(a) FLAIR-weighted

(b) T_1-weighted post-contrast

(c) T_1-weighted pre-contrast

(d) T_2-weighted

Figure A.10: The sparse sampling reconstructed 4x accelerated images for the four different sequence types FLAIR-weighted, T_1-weighted post-contrast, T_1-weighted pre-contrast and T_2-weighted.

Bibliography

[1] P.A. Rinck. “A short history of magnetic resonance imaging”. In: *Spectroscopy Europe* 20.1 (2008), pp. 7–10.

[2] P.C. Lauterbur. “Image Formation by Induced Local Interactions: Examples Employing Nuclear Magnetic Resonance”. In: *Nature* 242.5394 (1973), pp. 190–191.

[3] P.C. Lauterbur. “Magnetic resonance zeugmatography”. In: *Pure and Applied Chemistry* 40.1-2 (1974), pp. 149–157.

[4] R. Damadian, M. Goldsmith, and L. Minkoff. “NMR in cancer: XVI. FONAR image of the live human body”. In: *Physiological chemistry and physics* 9.1 (1977), pp. 97–100, 108.

[5] R.R. Edelman. “The History of MR Imaging as Seen through the Pages of Radiology”. In: *Radiology* 273.2S (2014), S181–S200.

[6] K.P. Pruessmann et al. “SENSE: Sensitivity encoding for fast MRI”. In: *Magnetic Resonance in Medicine* 42.5 (1999), pp. 952–962.

[7] A. Bjørnerud. *The Physics of Magnetic Resonance Imaging.* eng. Department of Physics. Univeristy of Oslo, 2020.

[8] M.T. Vlaardingerbroek. *Magnetic resonance imaging : theory and practice.* eng. 2nd, rev. and enl. ed. Berlin: Springer, 1999.

[9] C.E. Shannon. “Communication in the Presence of Noise”. In: *Proceedings of the IRE* 37.1 (1949), pp. 10–21.

[10] L. Arena, H.T. Morehouse, and J. Safir. “MR imaging artifacts that simulate disease: how to recognize and eliminate them.” In: *RadioGraphics* 15.6 (1995), pp. 1373–1394.

[11] J. Zhuo and R.P. Gullapalli. “MR Artifacts, Safety, and Quality Control”. In: *RadioGraphics* 26.1 (2006), pp. 275–297.

[12] G. McGibney et al. “Quantitative evaluation of several partial fourier reconstruction algorithms used in mri”. In: *Magnetic Resonance in Medicine* 30.1 (1993), pp. 51–59.

[13] D.L. Donoho. “Compressed sensing”. In: *IEEE Transactions on Information Theory* 52.4 (2006), pp. 1289–1306.

[14] Mingxing Tan and Quoc V. Le. *EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks.* 2020. arXiv: `1905.11946v5 [cs.LG]`.

[15] M. Tan, R. Pang, and Q.V. Le. “EfficientDet: Scalable and Efficient Object Detection”. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).* 2020, pp. 10778–10787.

[16] O. Ronneberger, P. Fischer, and T. Brox. “U-Net: Convolutional Networks for Biomedical Image Segmentation”. In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*. Ed. by N. Navab et al. Cham: Springer International Publishing, 2015, pp. 234–241.

[17] C. Dong et al. “Image Super-Resolution Using Deep Convolutional Networks”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38.2 (2016), pp. 295–307.

[18] J. Kim, J.K. Lee, and K.M. Lee. “Accurate Image Super-Resolution Using Very Deep Convolutional Networks”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 1646–1654.

[19] K. Hammernik et al. “Learning a variational network for reconstruction of accelerated MRI data”. In: *Magnetic Resonance in Medicine* 79.6 (2018), pp. 3055–3071.

[20] J. Schlemper et al. “A Deep Cascade of Convolutional Neural Networks for Dynamic MR Image Reconstruction”. In: *IEEE Transactions on Medical Imaging* 37.2 (2018), pp. 491–503.

[21] T. Eo et al. “KIKI-net: cross-domain convolutional neural networks for reconstructing undersampled magnetic resonance images”. In: *Magnetic Resonance in Medicine* 80.5 (2018), pp. 2188–2201.

[22] J. Zbontar et al. *fastMRI: An Open Dataset and Benchmarks for Accelerated MRI*. 2019. arXiv: `1811.08839v2 [cs.CV]`.

[23] M.J. Muckley et al. *State-of-the-Art Machine Learning MRI Reconstruction in 2020: Results of the Second fastMRI Challenge*. 2020. arXiv: `2012.06318v2 [eess.IV]`.

[24] K. Lønning et al. “Recurrent inference machines for reconstructing heterogeneous MRI data”. In: *Medical Image Analysis* 53 (2019), pp. 64–78.

[25] D.J. Griffiths. *Introduction to quantum mechanics*. eng. 2nd ed. Cambridge: Cambridge University Press, 2017.

[26] F. Bloch. “Nuclear Induction”. eng. In: *Physical review* 70.7-8 (1946), pp. 460–474.

[27] J. Graessner. “Bandwidth in MRI”. In: *Magnetom Flash* 2 (2013), pp. 3–8.

[28] F. Knoll et al. “fastMRI: A Publicly Available Raw k-Space and DICOM Dataset of Knee Images for Accelerated MR Image Reconstruction Using Machine Learning”. In: *Radiology: Artificial Intelligence* 2.1 (2020), e190007.

[29] M. Zaitsev, J. Maclaren, and M. Herbst. “Motion artifacts in MRI: A complex problem with many partial solutions”. In: *Journal of Magnetic Resonance Imaging* 42.4 (2015), pp. 887–901.

[30] C.M. Hyun et al. “Deep learning for undersampled MRI reconstruction”. In: *Physics in Medicine & Biology* 63.13 (2018), p. 135007.

[31] L.F. Czervionke et al. “Characteristic features of MR truncation artifacts”. In: *American Journal of Roentgenology* 151.6 (1988), pp. 1219–1228.

[32] S. Heiland. “From A as in Aliasing to Z as in Zipper: Artifacts in MRI”. In: *Clinical Neuroradiology* 18.1 (2008), pp. 25–36.

[33] Z.-P. Liang et al. “Constrained reconstruction methods in MR imaging”. In: *Reviews of Magnetic Resonance in Medicine* 4.2 (1992), pp. 67–185.

[34] D.A. Feinberg et al. "Halving MR imaging time by conjugation: demonstration at 3.5 kG." In: *Radiology* 161.2 (1986), pp. 527–531.

[35] M. Blaimer et al. "SMASH, SENSE, PILS, GRAPPA: how to choose the optimal method". eng. In: *Topics in Magnetic Resonance Imaging* 15.4 (2004), pp. 223–236.

[36] A. Deshmane et al. "Parallel MR imaging". In: *Journal of Magnetic Resonance Imaging* 36.1 (2012), pp. 55–72.

[37] D.J. Larkman and R.G. Nunes. "Parallel magnetic resonance imaging". In: *Physics in Medicine & Biology* 52.7 (2007), R15–R55.

[38] M. Lustig, D. Donoho, and J.M. Pauly. "Sparse MRI: The application of compressed sensing for rapid MR imaging". In: *Magnetic Resonance in Medicine* 58.6 (2007), pp. 1182–1195.

[39] O.N. Jaspan, R. Fleysher, and M.L. Lipton. "Compressed sensing MRI: a review of the clinical literature". In: *The British Journal of Radiology* 88.1056 (2015), p. 20150487.

[40] S.S. Vasanawala et al. "Improved Pediatric MR Imaging with Compressed Sensing". In: *Radiology* 256.2 (2010), pp. 607–616.

[41] J. Devlin et al. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.* 2019. arXiv: `1810.04805v2 [cs.CL]`.

[42] W. Huang et al. "NEURAL NETWORKS IN FINANCE AND ECONOMICS FORECASTING". In: *International Journal of Information Technology & Decision Making* 06.01 (2007), pp. 113–140.

[43] J. Schmidhuber. "Deep learning in neural networks: An overview". In: *Neural Networks* 61 (2015), pp. 85–117.

[44] A. Krizhevsky, I. Sutskever, and G. Hinton. "ImageNet classification with deep convolutional neural networks". In: *Communications of the ACM* 60.6 (2017), pp. 84–90.

[45] Dario Amodei et al. "Deep Speech 2 : End-to-End Speech Recognition in English and Mandarin". In: *Proceedings of The 33rd International Conference on Machine Learning.* Ed. by M.F. Balcan and K.Q. Weinberger. Vol. 48. Proceedings of Machine Learning Research. New York, New York, USA: PMLR, 2016, pp. 173–182.

[46] P. Mehta et al. "A high-bias, low-variance introduction to Machine Learning for physicists". In: *Physics Reports* 810 (2019), pp. 49–60.

[47] M. Nielsen. "Neural Networks and Deep Learning". In: Determination Press, 2015. Chap. Using neural nets to recognize handwritten digits.

[48] A. Pinkus. "Approximation theory of the MLP model in neural networks". In: *Acta Numerica* 8 (1999), pp. 143–195.

[49] G. Litjens et al. "A survey on deep learning in medical image analysis". In: *Medical Image Analysis* 42 (2017), pp. 60–88.

[50] M. Nielsen. "Neural Networks and Deep Learning". In: Determination Press, 2015. Chap. How the backpropagation algorithm works.

[51] I. Goodfellow, Y. Bengio, and A. Courville. "Deep Learning". In: MIT Press, 2016. Chap. Deep Feedforward Networks.

[52] G. Cybenko. "Approximation by superpositions of a sigmoidal function". eng. In: *Mathematics of Control, Signals, and Systems* 2.4 (1989), pp. 303–314.

[53] P. Kidger and T. Lyons. “Universal Approximation with Deep Narrow Networks”. In: *Proceedings of Thirty Third Conference on Learning Theory.* Ed. by Jacob Abernethy and Shivani Agarwal. Vol. 125. Proceedings of Machine Learning Research. PMLR, 2020, pp. 2306–2327.

[54] A.M. Schäfer and H.G. Zimmermann. “Recurrent Neural Networks Are Universal Approximators”. In: *Artificial Neural Networks – ICANN 2006.* Ed. by S.D. Kollias et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 632–640.

[55] D.-X. Zhou. “Universality of deep convolutional neural networks”. In: *Applied and Computational Harmonic Analysis* 48.2 (2020), pp. 787–794.

[56] A. Paszke et al. “PyTorch: An Imperative Style, High-Performance Deep Learning Library”. In: *Advances in Neural Information Processing Systems.* Ed. by H. Wallach et al. Vol. 32. Curran Associates, Inc., 2019, pp. 8024–8035.

[57] Y. Lecun, Y. Bengio, and G. Hinton. “Deep learning”. In: *Nature* 521.7553 (2015), pp. 436–444.

[58] M. Nielsen. “Neural Networks and Deep Learning”. In: Determination Press, 2015. Chap. Deep learning.

[59] I. Goodfellow, Y. Bengio, and A. Courville. “Deep Learning”. In: MIT Press, 2016. Chap. Convolutional Networks.

[60] S. Bai, J.Z. Kolter, and V. Koltun. *An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling.* 2018. arXiv: `1803.01271v2 [cs.LG]`.

[61] D. Silver et al. “A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play”. In: *Science* 362.6419 (2018), pp. 1140–1144.

[62] M. Nielsen. “Neural Networks and Deep Learning”. In: Determination Press, 2015. Chap. Why are deep neural networks hard to train?

[63] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. *Instance Normalization: The Missing Ingredient for Fast Stylization.* 2017. arXiv: `1607.08022v3 [cs.CV]`.

[64] Sergey Ioffe and Christian Szegedy. *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift.* 2015. arXiv: `1502.03167v3 [cs.LG]`.

[65] Z. Wang et al. “Image quality assessment: from error visibility to structural similarity”. In: *IEEE Transactions on Image Processing* 13.4 (2004), pp. 600–612.

[66] D.M. Rouse and S.S. Hemami. “Understanding and simplifying the structural similarity metric”. In: *2008 15th IEEE International Conference on Image Processing.* 2008, pp. 1188–1191.

[67] G. Van Rossum and F.L. Drake. *Python 3 Reference Manual.* Scotts Valley, CA: CreateSpace, 2009.

[68] A. Sriram et al. *End-to-End Variational Networks for Accelerated MRI Reconstruction.* 2020. arXiv: `2004.06688v2 [eess.IV]`.

[69] J. Hu, L. Shen, and G. Sun. “Squeeze-and-Excitation Networks”. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 2018, pp. 7132–7141.

[70] K. He et al. “Deep Residual Learning for Image Recognition”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).* 2016, pp. 770–778.

[71] P. Ramachandran, B. Zoph, and Q.V. Le. *Searching for Activation Functions.* 2017. arXiv: `1710.05941v2 [cs.NE]`.

[72] D.P. Kingma and J. Ba. *Adam: A Method for Stochastic Optimization.* 2017. arXiv: `1412.6980v9 [cs.LG]`.

[73] S.J. Reddi, S. Kale, and S. Kumar. *On the Convergence of Adam and Beyond.* 2019. arXiv: `1904.09237v1 [cs.LG]`.

[74] K. You et al. *How Does Learning Rate Decay Help Modern Neural Networks?* 2019. arXiv: `1908.01878v2 [cs.LG]`.

[75] Dominic Masters and Carlo Luschi. *Revisiting Small Batch Training for Deep Neural Networks.* 2018. arXiv: `1804.07612v1 [cs.LG]`.

[76] M. Abadi et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems.* 2015.

[77] Z. Wang et al. "Image quality assessment: from error visibility to structural similarity". In: *IEEE Transactions on Image Processing* 13.4 (2004), pp. 600–612.

[78] M. Drozdzal et al. "The Importance of Skip Connections in Biomedical Image Segmentation". In: *Deep Learning and Data Labeling for Medical Applications.* Ed. by G. Carneiro et al. Cham: Springer International Publishing, 2016, pp. 179–187.

[79] K. He et al. "Identity Mappings in Deep Residual Networks". In: *Computer Vision – ECCV 2016.* Ed. by B. Leibe et al. Cham: Springer International Publishing, 2016, pp. 630–645.

[80] V.K. Nikhil, A.B.P. Samson, and S.R.A. Chandra. *U-Det: A Modified U-Net architecture with bidirectional feature network for lung nodule segmentation.* 2020. arXiv: `2003.09293v1 [eess.IV]`.

[81] Dan Hendrycks and Kevin Gimpel. *Gaussian Error Linear Units (GELUs).* 2020. arXiv: `1606.08415v4 [cs.LG]`.

[82] M. Uecker et al. "Berkeley advanced reconstruction toolbox". In: *Proc. Intl. Soc. Mag. Reson. Med.* Vol. 23. 2486. 2015.

[83] M. Uecker et al. "ESPIRiT—an eigenvalue approach to autocalibrating parallel MRI: Where SENSE meets GRAPPA". In: *Magnetic Resonance in Medicine* 71.3 (2014), pp. 990–1001.

[84] O. Dietrich et al. "Influence of multichannel combination, parallel imaging and other reconstruction techniques on MRI noise characteristics". In: *Magnetic Resonance Imaging* 26.6 (2008), pp. 754–762.

[85] Student. "The Probable Error of a Mean". In: *Biometrika* 6.1 (1908), pp. 1–25.

[86] P. Virtanen et al. "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python". In: *Nature Methods* 17 (2020), pp. 261–272.

[87] D.O. Walsh, A.F. Gmitro, and M.W. Marcellin. "Adaptive reconstruction of phased array MR imagery". In: *Magnetic Resonance in Medicine* 43.5 (2000), pp. 682–690.

[88] Matthew J. Muckley et al. *Results of the 2020 fastMRI Challenge for Machine Learning MR Image Reconstruction.* 2021. arXiv: `2012.06318v3 [eess.IV]`.

[89] P.L.K. Ding et al. "Deep residual dense U-Net for resolution enhancement in accelerated MRI acquisition". In: *Medical Imaging 2019: Image Processing.* Ed. by E.D. Angelini and B.A. Landman. Vol. 10949. International Society for Optics and Photonics. SPIE, 2019, pp. 110–117.

[90] Z. Wang, E.P. Simoncelli, and A.C. Bovik. "Multiscale structural similarity for image quality assessment". In: *The Thrity-Seventh Asilomar Conference on Signals, Systems Computers, 2003*. Vol. 2. 2003, pp. 1398–1402.

[91] N. Pezzotti et al. *An Adaptive Intelligence Algorithm for Undersampled Knee MRI Reconstruction.* 2020. arXiv: `2004.07339v2 [eess.IV]`.